

Thesis submitted to the
UNIVERSITY OF MOHAMED BOUDIAF – MSILA
BUSINESS INCUBATOR SPACE- MSILA



FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE

In partial fulfillment of the requirements for the degree of

Masters in Computer Science

By

Messara, Norhane

Title of the thesis

**Intelligent E-commerce Platform for Small and
Home-based Businesses**

Under the supervision of

Mr. Bilal Lounnas

Composition of the jury

Mr. Abdelbaki Bouguerra	University of M'sila	President
Mr. Bilal Lounnas	University of M'sila	Reporter
Ms. Z. Maaref	University of M'sila	Examiner

June, 2026

Abstract

Hirftna (حرفتنا — "Our Craft") is a hybrid web marketplace for Algerian artisans and small home-based businesses. It combines a product marketplace, where sellers list handmade goods with images and price ranges, with a service workflow, where clients submit custom orders that go through negotiation, production, and delivery. The idea is straightforward: with handmade goods, the product and the service of making it cannot really be separated.

There is no shopping cart. A client describes what they want, sets a budget and deadline, and optionally attaches reference images. The seller accepts or rejects, works on the piece, sets a final price, and the client confirms completion. No direct messaging; the order flow handles everything. We also built a mutual rating system because artisans frequently deal with clients who place orders and then disappear. After each completed order, both sides rate each other publicly, so a seller can check a client's track record before agreeing to take on their work.

The backend runs on Node.js and Express.js, the frontend on React, and Supabase manages the database, authentication, and storage. The platform supports Arabic, English, and French with a full RTL (Right-to-Left) layout, an AI (Artificial Intelligence) chatbot powered by Google Gemini, two-factor authentication via email OTP (One-Time Password), seller verification, and a promotion system for featured placements. Hirftna was deployed as a working MVP (Minimum Viable Product) and is accessible online.

Keywords: hybrid marketplace; e-commerce; e-service; artisans; custom orders; Algeria

الملخص

«حرفتنا (Hirftna) هي منصة إلكترونية مختلطة مخصصة للحرفيين الجزائريين والمشاريع الصغيرة المنزلية. تجمع المنصة بين سوق للمنتجات، حيث يعرض البائعون سلعًا مصنوعة يدويًا مصحوبة بصور ونطاقات أسعار، وبين مسار عمل للخدمات، حيث يقدم العملاء طلبات مخصصة تمر بمراحل التفاوض والإنتاج والتسليم. الفكرة بسيطة: في حالة السلع المصنوعة يدويًا، لا يمكن الفصل فعليًا بين المنتج وخدمة صنعه.

لا توجد عربة تسوق. يصف العميل ما يريده، ويحدد الميزانية والموعد النهائي، ويرفق صورًا مرجعية بشكل اختياري. يقبل البائع أو يرفض الطلب، ويعمل على القطعة، ويحدد السعر النهائي، ويؤكد العميل إتمام العمل. لا توجد رسائل مباشرة، يتولى سير العمل الخاص بالطلب كل شيء. كما أنشأنا نظام تقييم متبادل لأن الحرفيين غالبًا ما يتعاملون مع عملاء يقدمون الطلبات ثم يختفون. بعد كل طلب مكتمل، يقوم الطرفان بتقييم بعضهما البعض علنًا، بحيث يمكن للبائع التحقق من سجل العميل قبل الموافقة على قبول عمله.

تعمل الخلفية على Node.js و Express.js ، بينما تعمل الواجهة الأمامية على React ، وتدير Supabase قاعدة البيانات والمصادقة والتخزين. تدعم المنصة ثلاث لغات العربية والإنجليزية والفرنسية مع تخطيط RTL كامل، وروبوت دردشة يعمل بالذكاء الاصطناعي مدعوم من Google Gemini ، ومصادقة ثنائية عبر كلمة مرور لمرة واحدة عبر البريد الإلكتروني، والتحقق من البائع، ونظام ترويج للمواقع المميزة. تم نشر Hirftna كنموذج أولي قابل للتشغيل ويمكن الوصول إليه عبر الإنترنت.

الكلمات المفتاحية: سوق هجين؛ التجارة الإلكترونية؛ الخدمة الإلكترونية؛ الحرفيون؛ الطلبات المخصصة؛

الجزائر.

Dedications

“To all those who stood by me, believed in me, and made this journey possible...

***To my father,** my pillar of strength and my greatest source of inspiration. Your sacrifices, your wisdom, and your unwavering belief in me have shaped the person I am today. This work is as much yours as it is mine.*

***To my mother,** the warmest heart I have ever known. Your prayers, your patience, and your endless love have been my shelter through every challenge. No achievement of mine could ever repay what you have given me.*

***To my brothers,** my lifelong companions, and my truest friends. You have filled my days with laughter, loyalty, and strength. Having you by my side is a blessing I never take for granted.*

***To my aunts,** for your kindness, your warm words of encouragement, and the love you have always shown me. You have been a second home to my heart, and I carry your support with me wherever I go.*

***To my grandparents,** whose wisdom, grace, and unconditional love have been the foundation of our family. You taught me the value of perseverance and humility. May God bless you and grant you health and long life.*

***To my friends,** who shared the late nights, the doubts, the laughter, and the small victories along the way. Your friendship made this journey lighter and far more meaningful.”*

Acknowledgments

First and foremost, I thank Allah for His mercy, guidance, and blessings throughout this journey, which gave us the strength to complete this work.

I would like to thank my supervisor, **Mr. Bilal LOUNNAS**, for his support, guidance, and availability throughout this project. His feedback helped me stay on track, especially during the early design phase when a lot of decisions had to be made quickly.

I sincerely thank the University Incubator of M'sila for their continuous support, guidance, and encouragement throughout this project. Their assistance greatly contributed to the success of this work.

I thank the Department of Computer Science at Mohamed Boudiaf University of M'sila for the opportunity to work on this practical project, which allowed me to apply my knowledge in a real context and gain valuable experience.

I sincerely thank my family and friends for their constant support, encouragement, and patience throughout this research project. Their belief in me helped me overcome challenges and complete this work.

Contents

List of figures	i
List of tables	i
Introduction	1
Chapter 1	3
Domain Analysis and Requirements	3
1.1.1. The Economic and Cultural Weight of Handicrafts.....	3
1.1.2. The Legal Framework (Law 18-05).....	5
1.4.1. Project Objectives	9
1.5.1. Concept of Hybridization in Hirftna	11
1.5.2. Core Hybrid Features	11
1.6.1. Identification of Actors	13
1.6.2. Functional Requirements	14
1.6.3. Non-Functional Requirements	18
Chapter 2	20
Design and Architecture	20
2.1. System Architecture.....	20
2.2. Backend Server	25
2.3. Frontend Application	41
2.4. Admin Dashboard	50
Conclusion.....	56
Chapter 3	57
System Implementation	57
3.1 Used Technologies	57
3.1.1. Development Environment.....	57
3.1.2. Development Stack.....	58
3.1.3. External Services.....	61
3.1.4. Deployment	62
3.2 Presentation of the Application	64
3.2.1 Public Pages	64
3.2.4 Admin Dashboard	73
3.2.5 AI chatbot.....	76
Conclusion.....	77
Chapter 4	78
Economic Study and Startup Project.....	78
4.1. Introduction	78
PART 1: IN-DEPTH MARKET STUDY	79

4.2.	Introduction and Methodology	79
4.3.	Project Presentation and Innovative Aspects	79
4.3.1.	Startup Identity.....	79
4.3.2.	Innovative Aspects.....	80
4.4.	PESTEL Analysis.....	80
4.5.	Target Market Analysis	82
4.5.1.	Market Sizing (TAM / SAM / SOM).....	82
4.5.2.	Market Growth.....	82
4.5.3.	Customer Segments and Personas.....	83
4.6.	Competitive Analysis	84
4.6.1.	Competitive Landscape.....	84
4.6.2.	Detailed Comparison.....	85
4.6.3.	Porter's Five Forces.....	85
4.6.4.	Unique Selling Proposition	86
4.7.	SWOT Analysis.....	86
4.8.	Primary Research — Field Interviews	87
4.8.1.	Methodology	87
4.8.2.	Key Findings.....	87
4.9.	Go-to-Market Strategy	88
4.9.1.	Pricing Strategy.....	88
4.9.2.	Distribution Channels and Acquisition	88
4.9.3.	Key Performance Indicators.....	89
4.9.4.	Growth Objectives	89
4.10.	Conclusion	89
	PART 2: BUSINESS MODEL AND DETAILED FINANCIAL STUDY	90
4.11.	Introduction.....	90
4.12.	Business Model Canvas	90
4.12.1.	Customer Segments.....	90
4.12.2.	Value Proposition.....	90
4.12.3.	Channels.....	90
4.12.4.	Customer Relationships	91
4.12.5.	Revenue Streams.....	91
4.12.6.	Key Resources.....	91
4.12.7.	Key Activities	91
4.12.8.	Key Partnerships	91
4.12.9.	Cost Structure.....	92
4.13.	Production and Organisation Plan.....	92

4.13.1. Service Delivery Process	92
4.13.2. Supply and Procurement	93
4.13.3. Workforce	94
4.13.4. Key Partnerships	94
4.14. Financial Projections.....	95
4.14.1. General Assumptions	95
4.14.2. Initial Investment Plan (CAPEX)	96
4.14.3. Revenue Projections — Three Scenarios	97
4.14.4. Projected Income Statement.....	99
4.14.5. Operating Expenses Breakdown	100
4.14.6. Cash Flow Projection	101
4.14.7. Required Capital Calculation	103
4.14.8. Financing Plan and Capital Structure.....	104
4.14.9. Profitability and Viability Indicators	104
4.15. Financial and Operational Risk Analysis.....	105
4.16. Intellectual Property and Legal Framework.....	106
4.16.1. Startup Labeling Process via startup.dz	106
4.16.2. Intellectual Property	107
4.16.3. Legal Structure.....	107
4.16.4. Administrative Formalities and Compliance	107
4.17. Development Roadmap (18–24 Months).....	108
Part 2 Conclusion	108
Chapter 4 Conclusion	109
References	111
APPENDICES	115

List of figures

Figure 1.1: Etsy product page.....	6
Figure 1.2: Instagram page of an Algerian artisan seller.....	7
Figure 1.3: The hybrid marketplace model adopted in Hirftna	12
Figure 1.4: Illustration of the Custom Order Lifecycle	15
Figure 2.1: System interaction overview of the Hirftna platform.	24
Figure 2.2 System containers	25
Figure 2.3: Backend server components	27
Figure 2.5: Entity-relationship diagram of the Hirftna database	34
Figure 2.6: Visitor use case diagram	44
Figure 2.7: Client use case diagram	45
Figure 2.8: Seller use case diagram	46
Figure 2.9: Registration and OTP verification process	47
Figure 2.10: Custom order request process	48
Figure 2.11: Managing orders and reviews process	50
Figure 2.15: Admin use case diagram	52
Figure 2.16: Display and manage users sequence diagram	53
Figure 2.17: Manage products sequence diagram	54
Figure 2.18: Manage categories sequence diagram.....	54
Figure 2.19: Manage promotions sequence diagram.....	55
Figure 2.20: View platform statistics sequence diagram.....	56
Figure 3.1: Deployment diagram.....	63
Figure 3.2: Home page	65
Figure 3.3: Product catalog page.....	66
Figure 3.4: Product details page	67
Figure 3.5: Seller profile page	68
Figure 3.6: Custom order request form	69
Figure 3.7: Client order management page	70
Figure 3.8: Wishlist page.....	70
Figure 3.9: Seller dashboard.....	71
Figure 3.10: Product management page	72
Figure 3.11: Seller order handling page	72
Figure 3.12: Promotion requests interface.....	73
Figure 3.13: Admin dashboard overview	74
Figure 3.14: User management interface.....	75
Figure 3.15: Category management interface.....	75
Figure 3.16: Promotion management interface.	76
Figure 3.17: Chatbot sample conversation.	76
Figure 4.1: Hirftna marketplace logo.	80

List of tables

Table 1.1: Product categories used in the Hirftna platform.....	4
Table 1.2: Comparative analysis of existing platforms versus Hirftna.	10
Table 1.3: Summary of the twelve core functional requirements of Hirftna.	18
Table 2.1: Technical stack used in the Hirftna platform.	23
Table 2.2: Attributes of the users table.....	28
Table 2.3: Attributes of the products table.	29
Table 2.4: Attributes of the orders table.	31
Table 2.5: Authentication API endpoints.	35
Table 2.6: Category API endpoints.	36
Table 2.7: Product API endpoints.	36
Table 2.9: Seller API endpoints.....	37
Table 2.10: Reviews and ratings API endpoints.....	38
Table 2.11: Supporting feature API endpoints.	38
Table 2.12: Administration API endpoints.....	39
Table 2.13: Public pages.....	42
Table 2.14: Authentication pages.	42
Table 2.15: Client pages (any logged-in user).....	43
Table 2.16: Seller pages.	43
Table 4.1: PESTEL analysis of the Algerian market for Hirftna.	81
Table 4.2: TAM, SAM, and SOM for Hirftna over the 5-year horizon.	82
Table 4.3: Persona summary.	84
Table 4.4: Competitive landscape map.	84
Table 4.5: Detailed comparison against principal competitors.	85
Table 4.6: Porter's Five Forces. Overall sector attractiveness: Medium.	86
Table 4.7: SWOT matrix grounded in the preceding analyses.....	87
Table 4.8: Key findings from the qualitative interviews (n=12 artisans).	88
Table 4.9: Growth objectives over the three horizons.....	89
Table 4.10: Service delivery process for a typical custom order on Hirftna.	92
Table 4.11: Main suppliers and how we pay them.	93
Table 4.12: Workforce evolution.	94
Table 4.13: Key partnerships and contributions.	95
Table 4.14: General financial assumptions used across all projections.....	96
Table 4.15: Initial investment plan (CAPEX) Year 1.	97
Table 4.16: Revenue projections under the three scenarios.	98
Table 4.17: Detailed revenue breakdown by stream.	99
Table 4.18: Five-year projected income statement.....	100
Table 4.19: Year 1 payroll breakdown.	101
Table 4.20: External operating charges over 5 years.....	101
Table 4.21: Year 1 monthly cash flow projection.	102
Table 4.22: Required capital calculation following the standard CNCSIIU format.....	103
Table 4.23: Funding plan and capital structure.	104
Table 4.24: Profitability indicators under the three scenarios.	105
Table 4.25: Financial and operational risk matrix.	106
Table 4.26: Development roadmap over the first 24 months.	108

Introduction

Most online marketplaces follow a simple model: fixed prices, a shopping cart, and instant checkout. That works for mass-produced goods, but not for handmade products. When an artisan makes something by hand, the price depends on the materials, size, detail, and deadline. Often, the item does not even exist yet; it gets made after the customer asks for it.

In Algeria, this mismatch is felt daily. The country has a deep craft tradition: pottery, textiles, silver jewelry, leatherwork, and woodcarving. and hundreds of thousands of artisans who depend on these skills for a living. Most of them sell through Facebook and Instagram. Social media gives them some reach, but no structure: no order tracking, no pricing tools, no reviews, no way for a buyer to verify whether a seller is reliable.

Problem and Objective

The problem has three sides. Sellers have no dedicated platform for how they actually work; everything happens through direct messages, which is hard to manage and impossible to scale. Buyers have no reliable way to find a trustworthy artisan or compare options. And there is a third side that rarely gets discussed: unreliable clients. It is common for someone to place a custom order, the artisan starts working (time, materials, effort), and then the client disappears. No payment, no pickup. The artisan is stuck with a piece made to someone else's specifications. None of the existing platforms has a mechanism to hold clients accountable.

We built Hirftna (حرفتنا, "Our Craft") to address all three. It is a hybrid marketplace that combines a product catalog, where artisans list their work with images, categories, and price ranges, with a service workflow that handles the custom order lifecycle: request, acceptance, production, final pricing, delivery, and mutual rating. We call it a hybrid because neither model alone would be enough. The catalog attracts buyers, but the order system is what makes the transaction actually work for made-to-order goods.

Motivation

Platforms like Etsy have proven that dedicated handmade marketplaces can succeed, but Etsy is built for Western markets. USD, Stripe, English. It is not realistic for an Algerian artisan. Locally, Ouedkniss treats a hand-embroidered dress the same as a used phone, and tools like Flexy offer basic storefronts without addressing variable pricing or the maker-buyer

relationship. None of these platforms deals with trust in both directions. That gap is what motivated this project.

Proposed Solution

The core of Hirftna is the custom order workflow. There is no shopping cart. A client describes what they want through a three-step form, including requirements, budget, delivery preferences, and contact details, and submits the request. The seller accepts or rejects (with a mandatory reason), works on the piece, sets a final price when it is ready, and the client confirms completion. After that, both sides rate each other. The client rates the seller and leaves a product review; the seller rates the client. Both reputations are public, which creates accountability on both sides, our direct answer to the fake-client problem.

The platform also includes bilingual Arabic/English support with full RTL layout, an AI chatbot powered by Gemini 2.5 Flash, two-factor authentication via email OTP, a seller verification system with trust badges, a promotion system for featured placements, and payment integration with Chargily (simulated for the MVP). The admin dashboard provides tools for managing users, products, categories, promotions, and platform statistics.

Methodology

We followed an incremental approach, building the backend first in 18 phases, each adding a complete feature, and wiring the frontend after the API (Application Programming Interface) was stable. The backend runs on Node.js v20 with Express.js v5; the frontend uses React 19 with Vite, and the database is PostgreSQL 15 managed through Supabase. The frontend is deployed on Vercel, the backend on Render, and the application is live at hirftna.vercel.app. [1]

Report Structure

This report has three chapters. Chapter 1 covers domain analysis: the handicraft sector, existing platforms, and the requirements we identified. Chapter 2 presents the system design: architecture, data model, UML (Unified Modeling Language) diagrams, and REST (REpresentational State Transfer) API. Chapter 3 covers implementation: technologies, deployment, and the user interfaces.

Chapter 1

Domain Analysis and Requirements

Before we could start designing the system, we needed to understand the environment it would operate in. Who are the users? What do they currently use? What is missing? This chapter goes through that groundwork. We start with the handicraft sector in Algeria and the legal rules around e-commerce, then review the existing platforms that artisans rely on today. From there, we identify the gaps those solutions leave open and explain the hybrid marketplace model we chose to address them. The chapter closes with the system requirements, the actors, the features, and the quality constraints that came out of this analysis and that will drive the design in Chapter 2.

1.1. Domain Context: Handicrafts and E-commerce in Algeria

To build a platform for Algerian artisans, we first need to understand two things: the craft sector itself, how big it is, what kinds of products it covers, and who the people behind it are, and the legal framework that governs online commerce in the country. This section covers both.

1.1.1. The Economic and Cultural Weight of Handicrafts

Traditional craftsmanship in Algeria is not just an economic activity; it is part of the country's cultural identity. A piece of Kabyle pottery or a hand-woven carpet is not just a product; it carries a history and a meaning that goes well beyond what someone pays for it. For many families, especially in rural areas, this kind of work is their primary source of income. A large portion of artisans are women working from home [2], which makes the craft sector one of the more important drivers of female employment in the country.

The numbers back this up. According to the Ministry of Tourism and Handicrafts, the crafts and trades sector employs close to one million people across Algeria. About 470,000 artisans are officially registered, and roughly 150,000 of them are women [3]. In 2022, the ministry reported that the sector contributed over 350 billion dinars to GDP [4], spread across more than 600 different craft specialties [5]. These are not niche numbers; this is a significant part of the economy.

On the regulatory side, Decree 96-01 laid out the conditions for registering as an artisan [6], and the Ministry of Tourism and Traditional Industry has been working for years to bring younger people into the field. More recently, initiatives like electronic registration for

artisans and the "auto-entrepreneur" status show that there is a real push toward digitizing the sector. Our project fits directly into that direction. [3]

Handicrafts categories

Algerian traditional crafts fall into a few major families. Pottery is probably the oldest, with roots going back to prehistoric times. Kabyle women in rural areas still produce terracotta pieces by hand using geometric patterns passed down through generations. Textiles cover everything from woven carpets to ceremonial garments like the chedda of Tlemcen, some of which have been recognized by UNESCO. Jewelry, particularly Kabyle and Tuareg silverwork that combines silver with coral, is what many people think of first when they hear "Algerian craftsmanship." [7] Leatherwork, woodcarving, and copper engraving round out the main categories. [2]

Table 1.1 lists the product categories we defined for the Hirftna platform. These are based on the most common craft types found across Algeria, and they map directly to the categories table in our database.

Category	Description
Pottery	Handmade terracotta and ceramic items such as plates, tagines, and decorative vases
Textiles	Woven carpets, traditional clothing, embroidered garments, and fabric-based crafts
Jewelry	Silver, coral, and handcrafted accessories including rings, bracelets, and necklaces
Leather goods	Handmade bags, belts, wallets, and traditional babouches
Paintings	Hand-painted artwork, calligraphy, and decorative wall pieces
Candles & Soap	Artisanal scented candles, handmade soaps, and natural skincare products
Food & Honey	Local honey, olive oil, traditional sweets, and homemade food products
Home décor	Handcrafted decorative items for the home, such as trays, mirrors, and wall hangings
Other	Any traditional craft that does not fall under the above categories

Table 1.1: Product categories used in the Hirftna platform.

1.1.2. The Legal Framework (Law 18-05)

Any e-commerce platform operating in Algeria needs to comply with Law 18-05, passed on May 10, 2018, which sets the rules for online commerce in the country [8] . The law covers several things: every seller must be registered as a commercial or craft activity, the seller's identity must be clearly visible to the consumer, buyers have a withdrawal period after purchase, and an electronic invoice must be generated for each transaction. Our platform does not implement every provision of this law at the MVP stage; for instance, the invoicing part is not built yet, but being aware of this framework matters for any future development of the project, especially around payment processing and legal seller verification.

1.2. State of the art

To understand where Hirftna fits, we studied the platforms that Algerian artisans and buyers currently rely on. We looked at global options, local marketplaces, and the social media tools that most sellers end up using by default. The goal was not to make an exhaustive catalog, but to identify what each solution does well and where it breaks down for the specific case of handmade, custom-made products.

Global Platforms

Etsy¹ is the most well-known global platform for artisanal products [9]. It does several things right: product pages are well-structured, sellers can tell their story and build a brand identity, and the review system gives buyers a reason to trust the platform. Figure 1.1 shows a typical Etsy product page with customization options and ratings.

¹ Etsy, Inc. *About Etsy*. Available at: <https://www.etsy.com/about> (Accessed June 2026).

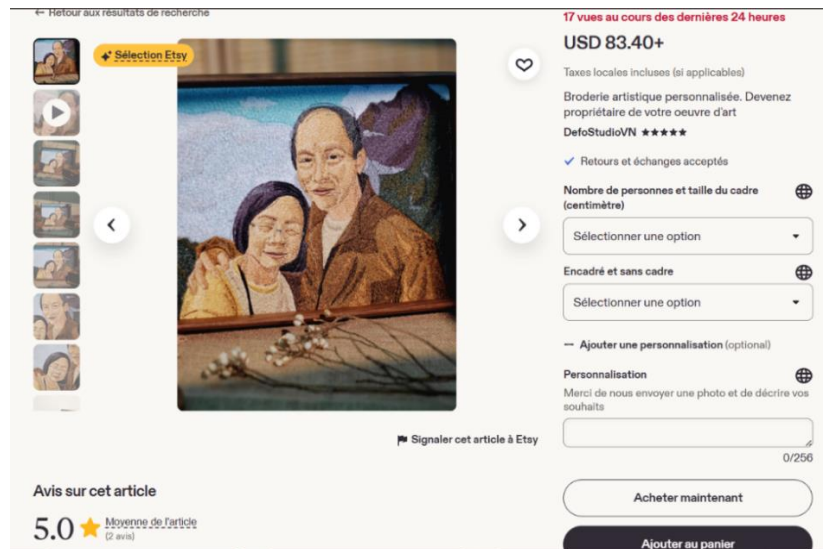


Figure 1.1: Etsy product page

The problem for Algerian users is that Etsy is built for global markets. It does not support local payment methods like CIB (Carte Interbancaire) or Edahabia cards; international shipping from Algeria is complicated and expensive, and the registration process assumes access to payment systems that most Algerian sellers do not have. So, despite being a good platform in general, Etsy is not a practical option for the market we are targeting.

General Local Platforms

Ouedkniss is by far the most popular online marketplace in Algeria. It is widely used and easy to access. But it is a general-purpose classifieds platform where people sell everything from used cars to apartments to electronics. Handmade products exist on Ouedkniss, but they get buried under thousands of unrelated listings. There is no concept of a custom order, no craft-specific categories that make sense to artisans, and communication between buyer and seller occurs entirely over phone calls, which are informal and unstructured. [10]

Specialized tools and communication channels

Then there are the platforms that most artisans actually end up using: Facebook and Instagram. These work well for one thing: getting visibility quickly. An artisan can post photos of their work, build a following, and take orders through direct messages. Figure 1.2 shows a real example of an Algerian artisan's Instagram page. [11]

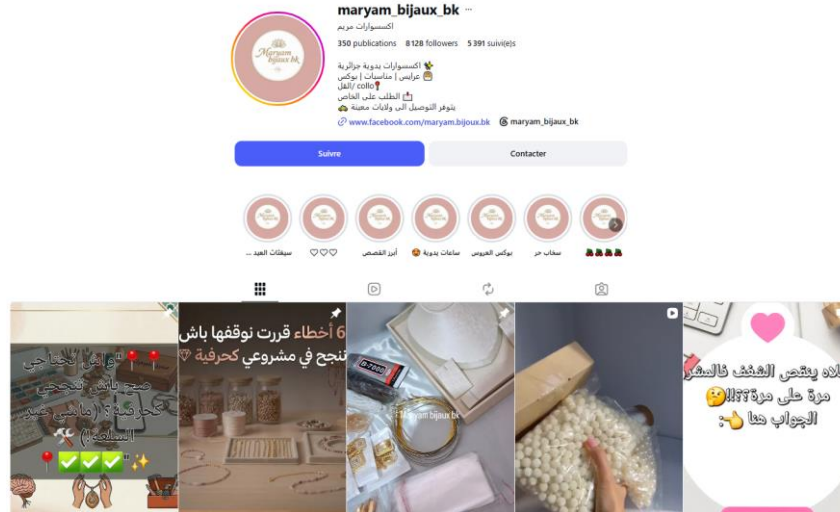


Figure1.2: Instagram page of an Algerian artisan seller.

But social media was not designed for commerce. There is no order management, no way to set up a proper product catalog with price ranges and categories, no review system that buyers can actually trust, and no mechanism to handle the back-and-forth that custom orders require. Everything happens in DMs, which is fine for one or two orders but breaks down quickly as the number of customers grows.

Other tools like Flexy offer basic online store creation, but they focus on the technical side of setting up a storefront without considering what makes artisanal commerce different, things like negotiable pricing, custom specifications, and the personal relationship between maker and buyer.

After looking at all of these, it was clear to us that nothing currently available meets the specific needs of Algerian artisans and their customers. Each option solves part of the problem, but none of them addresses the full picture. That observation is what pushed us toward building Hirftna.

1.3.Refined Problem Statement and Project Objectives

The platforms we studied each cover a piece of the puzzle, but none of them put it all together. Etsy [9] has the professionalism but ignores the Algerian context. Ouedkniss [10] is local but not specialized. Social media [11] gives visibility but no structure. In this section, we break down exactly where these solutions fall short, and then lay out the objectives we set for Hirftna based on those gaps.

1.4. Analysis of Current Gaps

Even though Facebook and Instagram are extremely popular among Algerian artisans, they remain surface-level solutions that do not address the real needs of either sellers or buyers. We identified four main shortcomings.

Gap 1- Informality and Lack of Structure: Most artisans use personal social media pages to showcase their work. This means there is no structured storefront that reflects the quality of what they make, no order management system to track requests from start to finish, and no organized way to communicate with customers about specifications, deadlines, and pricing. The entire selling process ends up being improvised, and that lack of structure hurts both the seller's efficiency and the buyer's confidence.

Gap 2- Pricing Rigidity: Traditional e-commerce platforms display a single fixed price for each product. But artisanal products do not work that way. The price of a handmade item depends on the materials the client wants, the size and level of detail, the degree of customization, and how urgently they need it. A fixed-price model creates a disconnect between how the product is actually made and how the platform tries to sell it. What artisans need is a way to show a price range and then agree on a final price with the client after understanding exactly what they want.

Gap 3- Trust Deficit on the Buyer Side: Trust is a basic requirement in any online transaction, but in the current landscape, buyers have very little to go on. There are no verified reviews, no seller verification mechanisms, and no detailed profiles that would help a buyer assess whether an artisan is reliable. Customers end up hesitating, and they spend a lot of time searching for a seller they feel they can trust.

Gap 4- Unreliable Clients (The Seller-Side Trust Problem): This gap is rarely discussed, but it comes up constantly when you talk to artisans. In the current informal system, it is common for a client to place an order for a custom piece, the artisan starts working, spending time, buying materials, sometimes days of labor, and then the client disappears. They never pick up the product, never pay, and the artisan is left with a finished item they cannot sell to anyone else because it was made to that specific client's specifications. This is not a rare occurrence; it is one of the biggest frustrations sellers mentioned during our research. None of the existing platforms has any mechanism to hold clients accountable, because they have no way to track client behavior or give sellers information about a buyer's reliability. This gap is what directly motivated us to implement a client rating system in Hirftna. Sellers rate clients after each

completed order, and that rating history is publicly visible. A seller can check a client's track record before deciding whether to accept their order.

1.4.1. Project Objectives

The four gaps above shaped the objectives we set for Hirftna. The overall goal is to build a professional platform where artisans can present their work properly, manage orders in a structured way, and have some assurance that the clients they work with are serious, while customers get a trustworthy space to discover local products and place orders with confidence. Here are the specific objectives we defined.

Changing the purchasing model (From Add-to-Cart to Request-to-Offer): Instead of the standard cart-based checkout, we implemented a custom order system. The client sends a request to the seller that includes their specifications, a budget range, a preferred deadline, and optional reference images. The seller reviews the request and decides whether to accept or reject it, providing a reason for rejection if they decline. This model fits artisanal work much better, because the details of the product often cannot be fully determined in advance, and the final result depends on a conversation between the maker and the buyer.

Flexible pricing through price ranges: Products on Hirftna display a minimum and maximum price instead of a fixed number. This gives the customer an approximate idea of cost, while giving the artisan the freedom to set a final price that reflects the actual work involved in that specific order.

Mutual accountability through two-way ratings: After a completed order, the client rates the seller and can leave a product review. But the seller also rates the client. Over time, both sides build a public reputation. A client who follows through on orders and behaves professionally will have a strong profile. A client who repeatedly abandons orders or wastes sellers' time will have a visible record that future sellers can check. This is our direct answer to the fake-client problem.

AI-powered assistance: We added a chatbot to the platform, powered by Google Gemini, that can help users while they browse. It answers questions about how the platform works, explains the ordering process, and helps clients find products, without requiring the seller to be available for every question.

Table 1.2 summarizes how Hirftna compares to the existing platforms we studied across the dimensions that matter most for artisanal commerce.

Criteria	Etsy (Global) [9]	Ouedkniss (Local) [10]	Social Media [11]	Hirftna (Our Project)
Market Focus	International Handmade	General Classifieds	General	Algerian Artisanat
Transaction Model	Instant Checkout	Manual Contact (Phone)	Direct Messages	Custom Order Request →Accept → Ready →Complete
Pricing Logic	Fixed Price	Negotiable / Hidden	Negotiable in DMs	Price Ranges (Min–Max) + Final Price by Seller
Seller Trust	Global Reviews	Basic Ratings	None	Verified Seller Profiles + Artisan Stories + Ratings
Client Trust	None	None	None	Client Rating System (sellers rate clients)
Local Payment	No (USD/Western)	Yes (DZD)	Cash / CCP (Compte Courant Postal)	Yes (DZD / Chargily)
AI Support	Generic Help	None	None	Integrated Craft Chatbot (Gemini)
Language	English	Arabic / French	Varies	Arabic (RTL) + English + French

Table 1.2: Comparative analysis of existing platforms versus Hirftna.

1.5. The Hybrid Marketplace Model

This section explains the hybrid model we adopted for Hirftna and why it fits handicrafts and custom-made products better than a traditional e-commerce approach.

In e-commerce, a "hybrid marketplace" usually refers to a platform that combines more than one sales model [12]. For example, Amazon operates both as a direct retailer and as a marketplace for third-party sellers. Other hybrids blend online and offline commerce into one system. The common thread is that these platforms do not limit themselves to a single, pure business model; they mix different approaches into something that works for their specific context.

In our case, we adapted the concept differently. Hirftna does not mix retail and marketplace roles. Instead, it combines a product-oriented marketplace with a service-oriented workflow for custom orders [12]. The platform has a product catalog where sellers list what they can make, with images, categories, ratings, and price ranges, which is the e-commerce side. But it also has a structured order process where clients submit requests, sellers negotiate and produce, and both sides track the order through a defined lifecycle that

is the e-service side. We call it hybrid because neither side would work alone: the catalog is what helps buyers discover products, but the service workflow is what makes the actual transaction possible for goods that are handmade and often made to order.

1.5.1. Concept of Hybridization in Hirftna

Most digital platforms fall into one of two categories: they are either product-oriented, like traditional online stores, or service-oriented, like freelance platforms or booking systems. We deliberately combined both in Hirftna because that is how artisanal commerce actually works in practice.

The product-focused side (e-commerce) is the catalog: Sellers list handmade goods, traditional clothing, pottery, jewelry, and so on, with images, categories, and indicators like ratings and reviews. This part works like any other online marketplace, and it is what gives buyers a way to browse and discover products.

The service-oriented side (e-service) is the custom order workflow: Since most of these products are made to order or heavily customized, the platform also manages the process of clarifying requirements, coordinating production, and tracking delivery. In this sense, the "product" is inseparable from the service of making it; the artisan is not just selling an item, they are translating the client's vision into a handcrafted piece through their skills.

By combining these two aspects, Hirftna can handle both straightforward orders where the client picks an existing design and more complex requests that involve customization, negotiation, and collaboration between the client and the artisan.

1.5.2. Core Hybrid Features

The hybrid nature of the platform comes down to three design decisions that distinguish it from standard e-commerce.

From fixed prices to price ranges: Instead of showing a single fixed price, each product on Hirftna displays a minimum and maximum price. This reflects the reality that the cost of a handmade item depends on the materials, the size, the level of detail, and sometimes how quickly the client needs it. The price range gives buyers a realistic estimate upfront, while the final price gets set by the seller later in the process, after they understand exactly what the client wants.

From "Add to Cart" to "Request Custom Order": The main call-to-action on product pages is not "Add to Cart"; it is "Request Custom Order." Clicking it opens a multi-step form where

the client describes what they need, sets a budget range, picks a deadline and delivery method, and can upload reference images. The result is a structured service request, not an instant purchase. This aligns with how service platforms work, where the details of the transaction need to be agreed upon before anything starts.

From anonymous storefronts to artisan stories: Instead of generic shop names and brand logos, the platform puts the artisan at the center. Every seller has a dedicated profile with their background, location, craft specialty, and a written story about their journey. This turns the transaction into something more personal; the buyer is not just ordering from a faceless store, they are connecting with the person who will actually make the product. In the craft sector, authenticity and trust are directly tied to the person behind the work, and the platform is designed to reflect that.

These three features give Hirftna its dual character: on the surface, it looks like a familiar online marketplace, but underneath, the way transactions actually happen is closer to a service platform built around customization, negotiation, and the relationship between the artisan and the client.

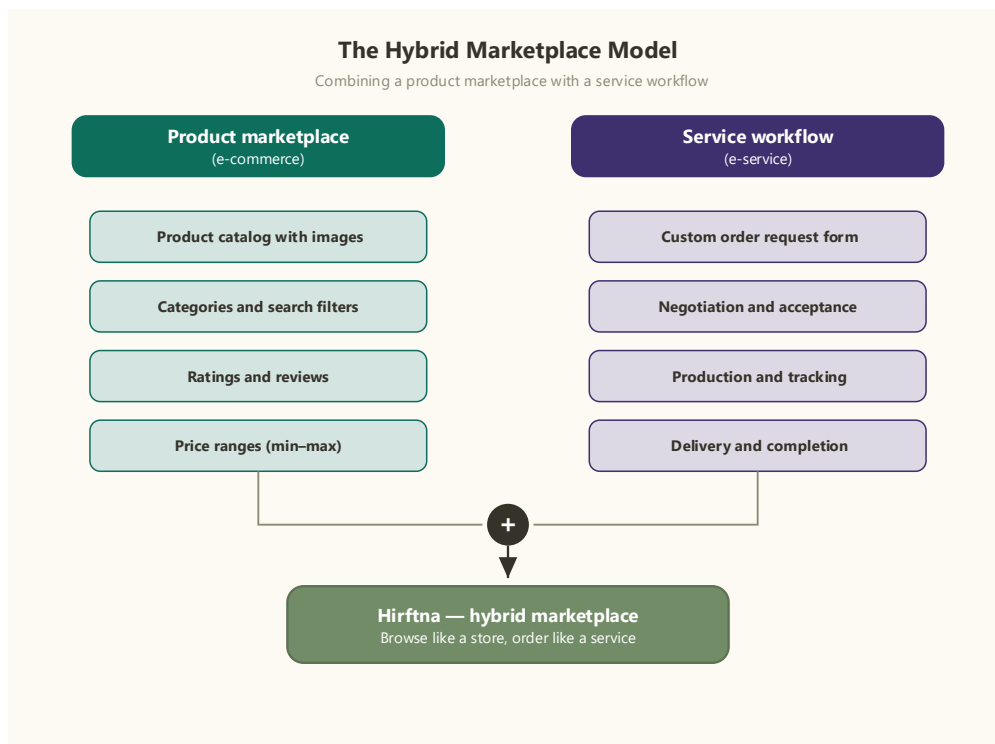


Figure 1.3: The hybrid marketplace model adopted in Hirftna

1.6. Overview of System Requirements

With the domain context, the market gaps, and the hybrid model all established, we can now define what the system actually needs to do. This section starts by identifying the actors who interact with the platform, then lists the functional requirements, the specific features and behaviors the system must support, followed by the non-functional requirements that define the quality standards and technical constraints we need to meet.

1.6.1. Identification of Actors

An actor represents a role that a user plays when interacting with the platform. Based on our analysis of the domain and the workflows described earlier, we identified four actors in the Hirftna system.

Visitor (unregistered user): Anyone browsing the platform without an account. Their goal is to discover artisans and products. They can browse the product catalog, search and filter by category and price, view product details and reviews, read seller profiles and artisan stories, and create an account either as a client or as a seller.

Client (registered buyer): A user with an account who wants to order a product. Their goal is to place custom orders and manage their shopping activity. They can submit custom order requests through the three-step form, track order status, manage a personal favorites list, submit ratings and reviews after completed orders, use the AI chatbot for assistance, and receive notifications about order updates.

Seller (artisan): A registered user who manages a digital shop. Their goal is to showcase their craft and fulfill custom orders. They can create and manage product listings with price ranges, respond to incoming orders by accepting or rejecting them, mark orders as ready and set the final price, update their shop profile and artisan story, rate clients after completed orders, view shop analytics, and request product promotions. Sellers can also place orders as buyers in other sellers' shops.

Administrator (platform moderator): The person responsible for the platform's integrity. Their goal is to maintain trust and content quality. They can verify or revoke seller accounts, manage users and change roles, moderate product listings, including force-deleting content, manage craft categories, review and act on promotion requests, and view platform-wide statistics. The admin role can only be assigned directly in the database; it is not accessible through the API for security reasons.

1.6.2. Functional Requirements

These requirements describe the specific features and behaviors the system must provide. Each one is tied back to either one of the gaps we identified in Section 1.3, the hybrid model described in Section 1.4, or a supporting need that emerged during the design process.

Authentication and account management: Users must be able to register and log in securely. Registration supports role selection; a user signs up as either a client or a seller, and each role has its own fields. Sellers, for instance, need to provide a shop name, location, and a description of their activity. The system includes two-factor authentication through email-based OTP codes during registration: after submitting the form, the user receives a six-digit code by email and has ten minutes and five attempts to verify it. The system also supports password reset through email with time-limited tokens.

Product management: Sellers need to create, edit, and delete product listings. Each product includes a name, description, images (with one cover image), a price range (minimum and maximum in dinars), an estimated number of days to complete, and a craft category. Sellers can also toggle product visibility without deleting the listing; this is useful when a seller wants to temporarily pause orders for a specific product.

Custom order lifecycle: This is the core of the platform and the most direct expression of the hybrid model. Instead of a shopping cart, the system uses a negotiation-based workflow that reflects how handmade products are actually ordered and produced:

- Step 1: The seller lists the products they can make, with images, a price range, and an estimated completion time.
- Step 2: The client browses the catalog, picks a product, and submits a custom order request. The form collects their requirements, a budget range (min and max in DA), a preferred deadline, the delivery method, the payment method, contact information, and optional reference images.
- Step 3: The seller receives a notification and reviews the request. They can accept it, which moves the order to "accepted" status, or reject it with a mandatory written reason, which moves it to "rejected" (a terminal state).
- Step 4: After finishing the work, the seller marks the order as "ready," sets the final price, and confirms the delivery method. The client gets notified.

- Step 5: The client reviews the final price and delivery details. If satisfied, they confirm the order as "completed."
- Step 6: After completion, both sides can rate each other. The client rates the seller and can leave a product review. The seller rates the client. This mutual rating is what builds accountability over time and directly addresses Gap 4.

Figure 1.4 below shows the activity diagram for this lifecycle, illustrating how the client, the system, and the seller interact at each step.

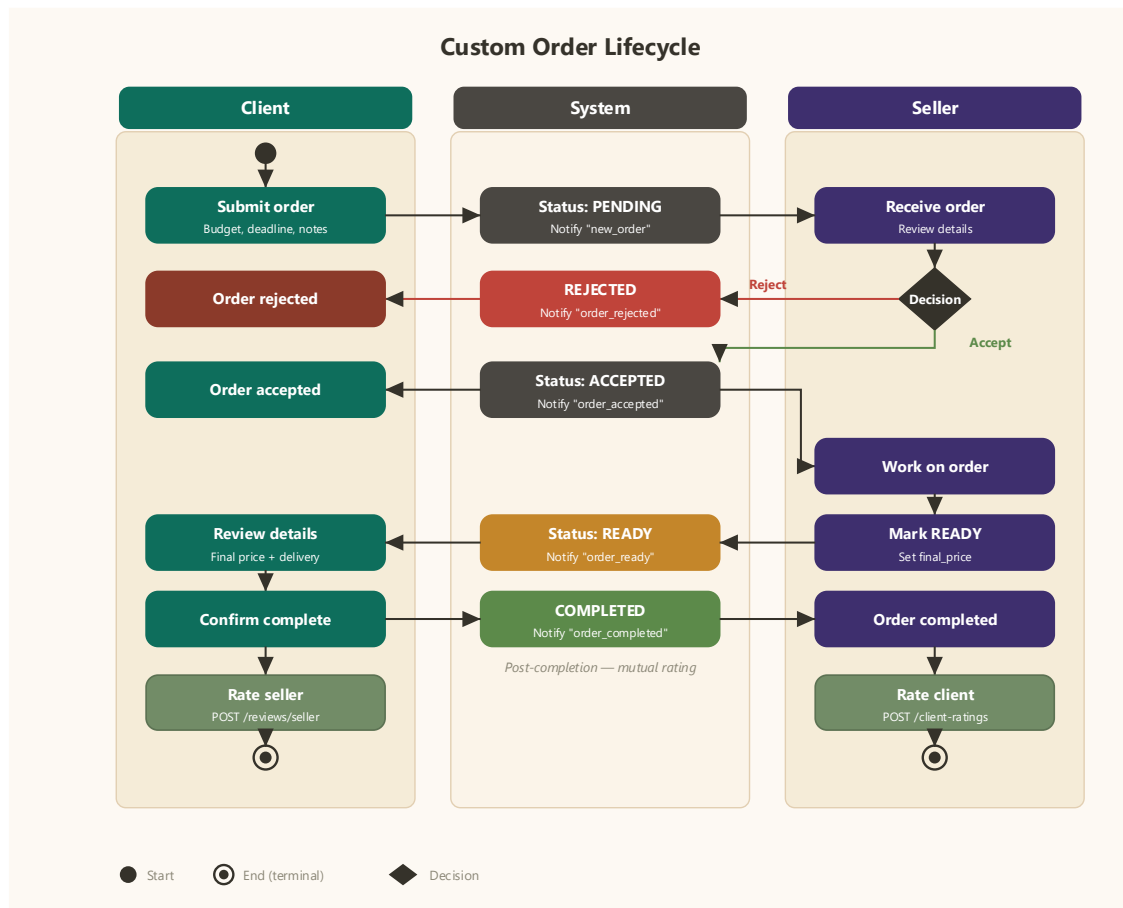


Figure 1.4: Illustration of the Custom Order Lifecycle

Seller profiles and artisan stories: Every seller has a public profile page that includes their shop name, location, a description of their work, and a written artisan story in Markdown format. This is designed to create a personal connection between the buyer and the maker, something that standard e-commerce platforms do not offer.

Seller verification: The platform supports two verification paths. The admin can manually verify a seller after checking their identity. On top of that, an automatic verification system checks whether a seller meets four criteria after each completed order: at least one active product, at least three completed orders, an average rating of

4.0 or higher, and a complete profile (shop name, description or bio, avatar, and location or city). Products from unverified sellers are hidden from the public catalog, which creates a natural incentive for new sellers to build up their track record. An admin override mechanism ensures that once an admin has explicitly set a seller's status, the automatic system cannot reverse that decision.

Wishlist: Any authenticated user, client, or seller can save products to a personal favorites list and remove them at any time. Since the platform does not use a cart model, the wishlist is how users keep track of products they might want to order later.

Review and rating system: After a completed order, the client can leave a product review (1 to 5 stars plus a comment) and rate the seller. The seller can also rate the client. This two-way system is one of the most important features of the platform, as it directly addresses the fake-client problem described in Gap 4. Sellers can check a client's rating history before accepting their order, and clients who consistently abandon orders or waste sellers' time will have a visible track record. Database triggers automatically recalculate the average rating for products and sellers whenever a review is added, updated, or deleted.

Notification system: Users receive notifications when important events happen in the order process: a new order arrives (for the seller), an order is accepted, rejected, marked as ready, or completed (for the client). The frontend polls for unread notifications at regular intervals so users stay informed without having to refresh the page constantly.

AI-powered chatbot: The platform includes a chatbot powered by Google Gemini 2.5 Flash. It helps users while they browse, explaining how the platform works, describing the ordering process, and helping clients find products. It is not a communication channel between buyers and sellers; that role belongs entirely to the structured order system. The chatbot is rate-limited to 20 requests per hour per user to stay within the free API tier.

Search and discovery: Users can search for products by keyword using PostgreSQL full-text search and filter results by category, price range, and rating. Pagination is applied to all list endpoints to keep the interface responsive as the catalog grows.

Promotion system: Sellers can request featured placement for their products in two locations: the hero banner on the homepage or the featured section on the browse page. They choose a duration (7, 14, or 30 days), submit the request, and the admin reviews it. If approved, the promotion runs for the requested period. If rejected, the admin provides a reason. This feature is designed as a future revenue source for the platform.

Administration and moderation: The admin has tools to manage users (search, filter by role, change roles, verify or revoke sellers), manage products (view all listings, including hidden ones, force-delete inappropriate content), manage craft categories (create, edit, delete), review promotion requests, and view platform statistics, including user counts, order volumes, revenue, and top-performing sellers and products.

To give a structured view of the functional requirements above, Table 1.1 below summarises the twelve core functional requirements identified for Hirftna, with their unique identifier, name, brief description, and the specific gap or design principle each one addresses. This table will serve as a reference throughout Chapters 2 and 3, where each functional requirement maps directly to the system design (use cases, sequence diagrams, database tables) and the implementation (API endpoints, frontend pages).

ID	Functional Requirement	Description	Related gap/principle
FR-01	Authentication & Account Management	Secure registration and login with role-based access (client/seller/admin), JWT-based session tokens, two-factor verification via OTP.	Gap 1: Informality
FR-02	Product Management	Sellers create, edit, and delete listings with images, price ranges, completion time, and category tags. No stock field — every product is made to order.	Hybrid model (price range)
FR-03	Custom Order Lifecycle	Six-step negotiation flow: submission → acceptance → ready → completion → mutual rating. A strict state machine is enforced in the backend.	Gap 3: No custom order workflow
FR-04	Seller Profiles & Artisan Stories	Public profile page per seller with shop name, location, Markdown-formatted artisan story, and aggregated rating.	Hybrid model (artisan story)
FR-05	Seller Verification	Two-path verification: manual admin verification and automatic check against the Algerian artisan registry. Verification is a trust badge, not a visibility gate.	Gap 2: No trust system
FR-06	Wishlist	Authenticated users (client or seller) can save products to a personal favourites list and remove them at any time.	User retention
FR-07	Review & Rating System	Two-way rating after each completed order: clients rate products and sellers; sellers rate clients.	Gap 4: Ghosting problem
FR-08	Notification System	Real-time notifications for order events (new request, accepted, rejected, ready, completed) and admin actions (verification, promotion approval).	User engagement

FR-09	AI-Powered Chatbot	Multilingual assistant powered by Google Gemini 2.5 Flash that explains the platform, guides users through the ordering process, and helps locate products.	Accessibility
FR-10	Search & Discovery	PostgreSQL full-text search with filters for category, price range, and rating; pagination on all list endpoints.	Catalogue navigation
FR-11	Promotion System	Sellers can request featured placement (hero banner or featured section) for 7, 14, or 30 days. Admin approves and activates each promotion.	Revenue model
FR-12	Administration & Moderation	Admin manages users (search, role change, verification, revocation), products (visibility, force-delete), categories, promotions, and platform statistics.	Platform integrity

Table 1.3: Summary of the twelve core functional requirements of Hirftna.

1.6.3. Non-Functional Requirements

Beyond what the system does, there are constraints on how well it needs to do it. These non-functional requirements define the quality standards and technical boundaries we set for the project.

Security: Data access is protected at two levels. At the application level, every API request to a protected route must include a valid JWT token, and middleware checks the user's role before allowing the operation. At the database level, Row Level Security (RLS) policies in PostgreSQL ensure that even if a bug in the application code lets a request through, the database itself blocks unauthorized access. For example, a seller can only modify their own products, and a client can only view their own orders.

Usability and accessibility: The interface follows a mobile-first design approach using TailwindCSS, since many users in Algeria primarily access the internet through their phones. The platform supports right-to-left (RTL) layout for Arabic, with automatic switching when the user changes language. All interactive elements, animations, and navigation adapt to the text direction.

Performance: Product images are stored in Supabase Storage with upload validation (file type, magic bytes, size limits) to keep storage efficient. The frontend implements loading skeletons and progressive rendering so users see content as it loads rather than staring at a blank page.

Reliability: The order lifecycle follows strict state transitions enforced in the backend: pending can only become accepted or rejected, accepted can only become ready, ready can only become completed, and both rejected and completed are terminal states. This prevents invalid transitions like jumping from pending to completed or modifying a closed order.

Scalability: The architecture is designed so that scaling does not require rewriting the application. The database, authentication, and storage are all on Supabase's cloud infrastructure, and the backend is stateless, with no server-side sessions, which means adding more server instances later would not require code changes.

Maintainability: The codebase follows a layered structure with a clear separation between routes, controllers, services, and validators. Each feature has its own files across these layers, making it straightforward for someone new to the project to find and modify the code related to a specific functionality.

Availability: The platform runs on Supabase for the database, Vercel for the frontend, and Render for the backend. This gives us automated deployments, managed uptime, and automatic SSL (Secure Sockets Layer). One known limitation of the MVP deployment is that the backend on Render's free tier spins down after 15 minutes of inactivity, which causes a cold-start delay of about 30 seconds on the first request after being idle.

Conclusion

In this chapter, we covered the background of the Hirftna project. We started with the economic weight of handicrafts in Algeria and the legal framework for e-commerce, then reviewed the platforms that artisans currently use and identified four gaps: the informality of social media selling, the rigidity of fixed-price models, the lack of trust mechanisms for buyers, and the problem of unreliable clients that no existing platform addresses. From there, we introduced the hybrid marketplace mode, combining e-commerce product discovery with an e-service custom order workflow, and showed why it fits artisanal commerce better than either model on its own. Finally, we defined the actors, functional requirements, and non-functional requirements that will guide the system design. In the next chapter, we move to the technical side: the architecture, the data model, and the UML diagrams that describe how the platform is built.

Chapter 2

Design and Architecture

The previous chapter was about understanding the problem, who the users are, what is missing in the market, and what the platform needs to do. This chapter is about figuring out how to build it.

We start with the system architecture: why we chose a client-server model, what technologies go into each layer, and how the different parts of the system talk to each other. Then we go into the data model, the database tables, the relationships between them, and the reasoning behind some of the less obvious schema decisions, like storing price ranges instead of fixed prices and enforcing order status transitions. After the data model, we present the REST API structure, the middleware and security layers, and how the backend is organized internally. On the frontend side, we describe the pages available to each user role, the use case diagrams for all four actors, the main workflows through flowcharts, and the sequence diagrams that trace key interactions step by step. By the end of this chapter, the design should be detailed enough to move directly into implementation, which is what Chapter 3 covers.

2.1. System Architecture

This section describes the technical foundation of the Hirftna platform, the architectural style we followed, the technologies we selected for each layer, and how the system's components are connected and communicate with each other.

2.1.1. Architectural Style

Hirftna follows a client-server architecture, which is a standard model for web applications [13]. The system is split into two independent parts: a client that sends requests and a server that processes them and returns responses. In our case, the client is a web browser running a React application, and the server is a Node.js/Express API that handles all the business logic and communicates with the database. The two sides interact over HTTP, and neither one needs to know how the other works internally. The client only cares about what the API returns, and the server only cares about what the request contains.

We went with this separation for a practical reason: it lets us work on the frontend and the backend independently. If we ever need to change the interface or build a mobile client, the API stays the same. That kind of decoupling is one of the main reasons client-server architectures are so widely used [14].

For the communication protocol between client and server, we adopted REST. REST was originally defined by Roy Fielding in his 2000 doctoral dissertation as a set of constraints for designing networked applications [15]. The core idea is that the server exposes resource products, orders, users, and so on through standard URLs, and the client interacts with them using HTTP methods: GET to read, POST to create, PUT to update, PATCH for partial modifications, and DELETE to remove. Each request is self-contained: the server does not store any session state between requests, so everything it needs to process a request must be included in the request itself [15].

In practice, this means our API uses a predictable URL structure: `/api/v1/products`, `/api/v1/orders/:id/status`, `/api/v1/sellers/me`, and so on. All responses follow a consistent JSON (JavaScript Object Notation) format, and HTTP status codes are used as expected: 200 for success, 201 for created, 400 for bad input, 401 for missing authentication, 403 for wrong role, 404 for not found. This consistency made the API much easier to test in Postman and to consume on the frontend side.

2.1.2. Technical Stack

We organized the system into three layers, each responsible for a different concern. This is sometimes called a three-tier architecture, where the presentation, logic, and data layers are handled separately [14]. The following describes what we used for each layer and why.

Presentation Layer (Frontend)

The user-facing side of the platform is built with React 19, using Vite as the build tool. We chose React because of its component-based structure, which made it natural to break the interface into reusable pieces: a product card, an order status badge, a navigation bar, a rating widget, each one handling its own logic and appearance. For styling, we used TailwindCSS, which gave us the flexibility to build a custom design system (warm cream backgrounds, sage green accents, rounded corners) without being constrained by a pre-built component library. Page navigation is handled by React

Router v7, and we added i18next for internationalization since the platform needs to support both Arabic and English with full right-to-left layout switching.

Logic Layer (Backend)

The backend runs on Node.js (v20) with Express.js (v5) as the web framework. We wrote everything in plain JavaScript, no TypeScript, to keep things simple and reduce the setup overhead for an MVP. Input validation is handled by Zod (v4), which lets us define strict schemas for every request and reject malformed data before it ever reaches the business logic. For logging, we use Winston, which writes structured logs with timestamps and severity levels. During development, those logs saved us a lot of time when something went wrong, and the error was not obvious from the API response alone.

Data Layer (Database and Services)

For the database, we chose Supabase, which is a managed PostgreSQL 15 instance with additional services built around it. Supabase gives us several things in one place: a relational database for all our data, an authentication system (Supabase Auth) that handles registration and JWT token generation, and a storage service for hosting product images and user avatars. It also provides a real-time engine, though in the current MVP, we ended up using polling on the frontend instead of WebSocket connections because it was simpler to get working reliably. We poll for unread notifications every two minutes.

Using Supabase saved us from setting up and managing each of these services independently, which would have been unrealistic given our one-month timeline.

Table 2.1 summarizes the full technical stack.

Layer	Technology	Role
Frontend	React 19 (Vite)	User interface and SPA routing
Frontend	TailwindCSS	Utility-first styling with custom design system
Frontend	React Router v7	Client-side page navigation and route guards
Frontend	i18next	Internationalization (AR/EN+ RTL switching)
Frontend	Axios	HTTP client with JWT interceptor and token refresh

Backend	Node.js v20	Server-side JavaScript runtime
Backend	Express.js v5	HTTP routing and middleware
Backend	Zod v4	Request validation with field-level errors
Backend	Winston	Structured logging
Backend	Nodemailer + SMTP	OTP emails and password reset links
Database	Supabase (PostgreSQL 15)	Relational data storage
Services	Supabase Auth	User authentication and JWT token management
Services	Supabase Storage	Image and file hosting (product-images, avatars)
Services	Gemini 2.5 Flash API	AI-powered chatbot
Services	Chargily	Algerian payment gateway (simulated in MVP)

Table 2.1: Technical stack used in the Hirftna platform.

2.1.3. System Interaction Overview

Figure 2.1 gives a high-level view of how the different parts of the Hirftna system are connected. On one side, we have the four actors from Chapter 1: visitors, clients, sellers, and the administrator, who all interact with the platform through the React frontend. The frontend communicates with the Express backend through a REST API, with every request carrying a JWT token for authentication when the user is logged in. The backend handles all the business logic and queries the Supabase cloud platform, which provides the PostgreSQL database, the authentication service, and the storage service.

Beyond these core components, the system depends on two external services. The Gemini API powers the AI chatbot that assists users while browsing. The SMTP server, accessed through Nodemailer, handles sending OTP codes during registration and password reset emails. Figure 2.1 illustrates this architecture and the communication paths between each component.

Figure 2.1 — System Interaction Overview (Context)

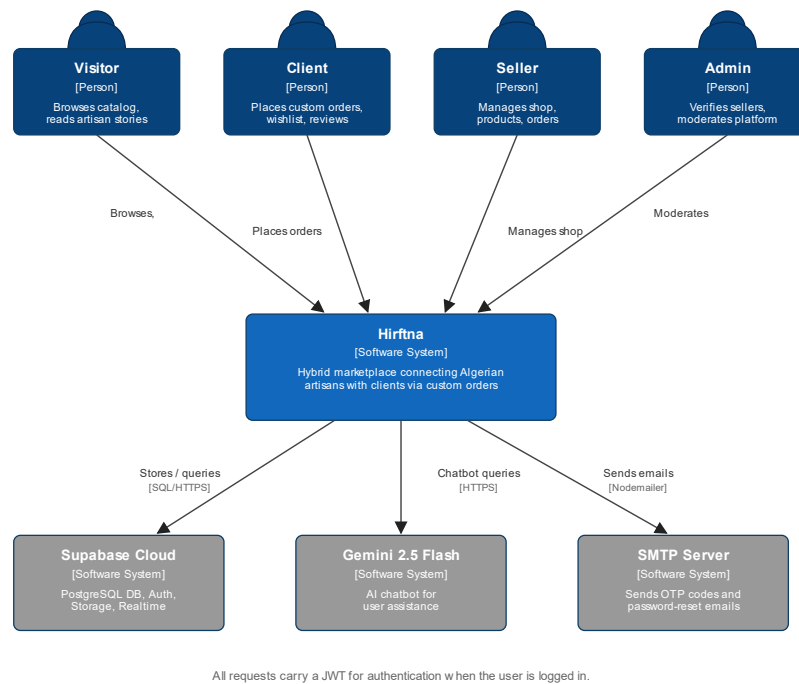


Figure 2.1: System interaction overview of the Hirftna platform.

2.1.4. System Containers

Figure 2.2 zooms in on what is inside the system boundary. While Figure 2.1 showed the platform as a single block, this diagram breaks it down into its main containers and how they communicate.

The first container is the end-user application, built with React and Vite. This is the interface that visitors, clients, and sellers all interact with. It handles everything from browsing the product catalog to submitting custom order requests and managing a seller's shop. The second container is the admin dashboard, which is technically part of the same React application but serves a completely different purpose. It is only accessible to administrators and provides tools for managing users, verifying sellers, moderating products, and monitoring platform activity. Both containers communicate with the backend through JSON over HTTPS (HyperText Transfer Protocol Secure).

The backend server, built with Node.js and Express.js, sits at the center of the architecture. It receives API requests from the frontend, validates the input, processes the business logic, and returns the appropriate responses. It also communicates with the

two external services: the Gemini API for chatbot queries and an SMTP server for sending OTP and password reset emails through Nodemailer.

At the data layer, the backend reads from and writes to a PostgreSQL 15 database hosted on Supabase. Beyond the database itself, Supabase provides three additional services: an authentication service for managing user accounts and generating JWT tokens, a storage service for hosting product images and user avatars in two public buckets (product-images and avatars), and a real-time engine that the platform architecture supports for pushing notifications, though in the current MVP we opted for frontend polling at two-minute intervals because it was simpler to implement and debug.

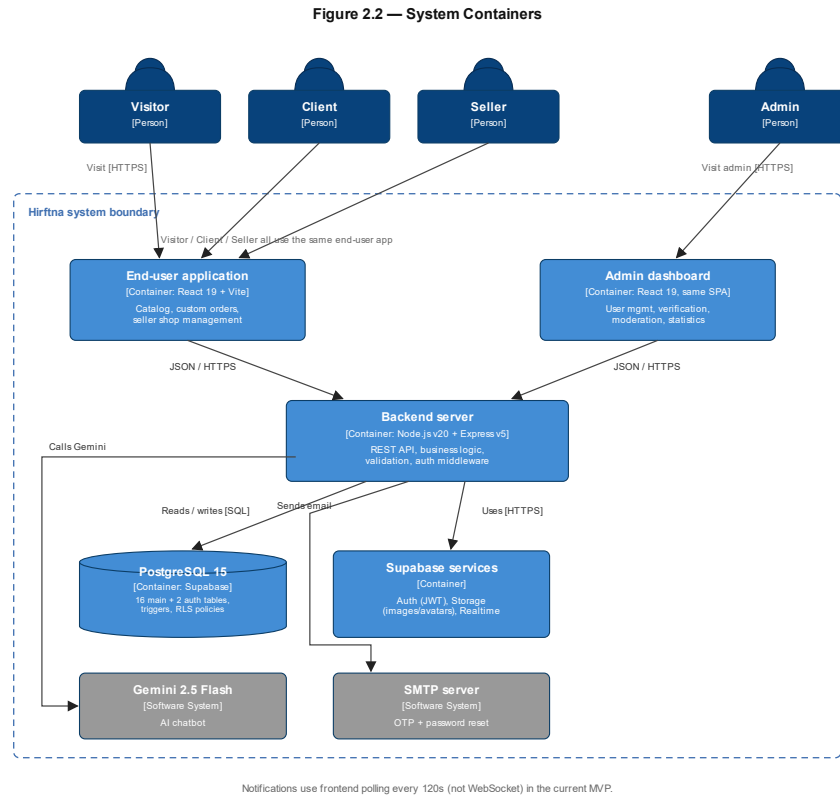


Figure 2.2 System containers

2.2. Backend Server

The backend is the layer that sits between the user interface and the database. Every action on the platform, whether it is a visitor browsing products, a client submitting a custom order, or an admin verifying a seller, goes through this server. This section describes how the backend is structured, how a request flows through it, and why we made some of the choices we did.

When a request comes in, the first thing it hits is the Router, which looks at the URL and HTTP method to determine which handler should deal with it. But the request does not go straight to the business logic; it first passes through a set of Middleware functions, each one doing a specific check. The authentication middleware verifies the JWT token to confirm the user is logged in. The role middleware checks whether that user has the right role for the action they are trying to perform, for example, only sellers can create products, and only admins can verify seller accounts. The validation middleware runs the request body against a Zod schema to catch missing or malformed fields before they cause problems further down. If any of these checks fail, the request gets rejected with a clear error message and never reaches the next step.

Only after all of that does the request reach the Controller. The controller does not contain business logic itself; its job is to call the right Service function, which is where the actual work happens. The service runs database queries, applies business rules (like checking whether a status transition is valid), and returns the result. The controller wraps that result into a standardized JSON response and sends it back to the frontend.

One decision worth mentioning is how we interact with the database. Some similar projects use a dedicated ORM like Prisma, but we went with the Supabase JavaScript client directly. Since we were already using Supabase for authentication, storage, and real-time features, it made more sense to use its built-in query builder instead of adding another dependency. It kept the setup simpler, which mattered given our timeline.

Figure 2.3 — Backend Server Components

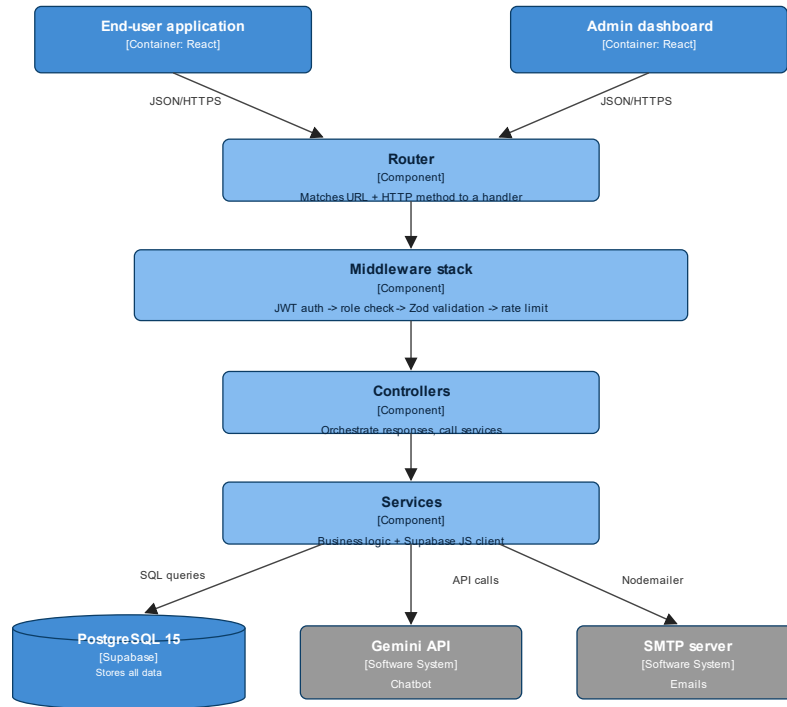


Figure 2.3: Backend server components

2.2.1. Data Model

Figure 2.4 shows the entity-relationship diagram of the Hirftna database. The schema consists of 16 main business tables plus 2 supplementary authentication tables (otp_sessions and password_reset_tokens), all hosted on PostgreSQL 15 through Supabase. Rather than going through every table with the same level of detail, we focus on the three main entity groups that are central to the platform: user-related entities, product and order entities, and the trust system. For each key table, we describe the attributes and explain the reasoning behind the less obvious design choices.

User-related entities

The users table is at the core of the whole schema. Each row mirrors an authentication record managed by Supabase Auth. There is a database trigger (on_auth_user_created) that automatically creates a user row whenever someone registers. The table captures the basic profile information shared by all users on the platform. Table 2.2 lists its attributes.

Attribute	Type	Description
id	UUID	The primary key mirrors the corresponding record in Supabase Auth
email	TEXT	The user's email address must be unique across the platform
full_name	TEXT	The user's display name
phone	TEXT	An optional contact phone number
avatar_url	TEXT	Link to the profile image stored in Supabase Storage
role	TEXT	Determines permissions can be client, seller, or admin
created_at	TIMESTAMPTZ	Account creation timestamp
updated_at	TIMESTAMPTZ	Last profile update timestamp

Table 2.2: Attributes of the users table.

The role field is the most important column in this table because it controls everything a user can and cannot do on the platform. A client can browse and place orders. A seller can do everything a client does, plus manage a shop and handle incoming orders. An admin has full access to moderation tools. The admin role can only be assigned directly in the database, never through the API. The endpoint for changing user roles only accepts "client" or "seller" as values.

When someone registers as a seller, a separate row is created in the sellers table, which extends their profile with shop-specific information. This includes the shop name, a short description and bio, a longer artisan story written in Markdown that appears on their public profile, and their location and city. The `is_verified` flag can be set by the admin after manually checking the seller's identity, or it can be updated automatically by the verification service when certain criteria are met. There is also an `admin_override` column when an admin explicitly sets a seller's verification status. This column gets set to a non-null value (TRUE or FALSE), which prevents the automatic system from overriding the admin's decision later. Fields like `avg_rating` and `total_sales` are computed automatically by database triggers whenever a new rating is submitted. We also added a `response_time_hours` field, which gives potential clients an idea of how fast a seller usually responds to incoming orders.

Two additional tables handle authentication features that go beyond what Supabase Auth provides on its own. The `otp_sessions` table stores everything needed during two-factor verification at registration: the hashed OTP code (SHA-256, never stored in plain text), how many attempts have been made (maximum 5), when the session expires (10-minute TTL), and a snapshot of the user data that gets committed to the database once verification succeeds. The session is identified by a UUID token, not by the email, so there is no way to enumerate users through the OTP endpoint. The `password_reset_tokens` table works similarly; it holds a temporary token linked to a user ID, set to expire after 15 minutes, and the token is deleted immediately after successful use, so it can only be used once.

Product and order entities

The `products` table is where sellers list the items they can make. What makes it different from a typical e-commerce product table is that there is no stock field, everything on Hirftna is made to order, so inventory tracking does not apply. Table 2.3 lists the main attributes.

Attribute	Type	Description
id	UUID	Primary key
seller_id	UUID	The foreign key links to the seller who listed this product
category_id	UUID	Foreign key linking to the product's craft category
name	TEXT	The product name
description	TEXT	A detailed description of the product
price	NUMERIC	A base reference price (legacy fallback)
price_min	NUMERIC	Lower bound of the price range in DA
price_max	NUMERIC	Upper bound of the price range in DA
completion_days	INTEGER	Estimated number of days to produce the item
avg_rating	NUMERIC	Average rating, recalculated automatically by a trigger
view_count	INTEGER	Number of times the product page has been viewed
is_active	BOOLEAN	Whether the product is visible to buyers
is_featured	BOOLEAN	Whether the product appears in the featured section
is_new	BOOLEAN	Whether the product is tagged as newly added
fts	TSVECTOR	Full-text search vector used by PostgreSQL's search engine

Table 2.3: Attributes of the `products` table.

The three fields that really set this table apart are `price_min`, `price_max`, and `completion_days`. Instead of a single fixed price, the platform displays a range, for example, "2,500 – 4,000 DA", which reflects the fact that the final cost of a handmade item depends on the materials, the size, the level of detail, and sometimes the urgency. The `completion_days` field gives clients a rough timeline before they even submit an order. These three fields are a direct implementation of the hybrid marketplace model from Chapter 1. There is also an `fts` column, which is a PostgreSQL `tsvector` that enables full-text search across product names and descriptions. Product images are stored in a separate `product_images` table, where the `position` field determines the display order. Position 0 is always the cover photo.

Regarding verification: the `is_verified` flag on sellers is used as a trust signal, not a visibility gate. All products from active sellers appear in the public catalog regardless of verification status. However, verified sellers display a trust badge on their profile and product cards, which gives buyers a quick visual cue about the seller's track record. The verification criteria (at least one active product, three completed orders, an average rating of 4.0 or higher, and a complete profile) are checked automatically after each completed order by a backend service. The admin can also verify or revoke a seller manually. The `admin_override` column ensures that once an admin has explicitly set a seller's status, the automatic system cannot reverse that decision.

The orders table is the most complex entity in the schema. Each row represents a custom order request and connects a client to a seller. Table 2.4 lists its attributes.

Attribute	Type	Description
id	UUID	Primary key
client_id	UUID	The user who placed the order
seller_id	UUID	The seller who receives the order
status	TEXT	Current lifecycle state: pending, accepted, rejected, ready, or completed
total_amount	NUMERIC	Calculated total for the order
final_price	NUMERIC	Set by the seller when marking the order as ready
delivery_type	TEXT	Chosen delivery method: fast, office_pickup, or hand_to_hand
payment_method	TEXT	Payment preference: card or cash_on_delivery

client_name	TEXT	Client's name for delivery purposes
client_phone	TEXT	Client's phone number
client_address	TEXT	Delivery address
custom_description	TEXT	Detailed description of what the client wants
notes	TEXT	Any additional notes from the client
budget_min	NUMERIC	Lower bound of the client's stated budget
budget_max	NUMERIC	Upper bound of the client's stated budget
deadline	DATE	The client's requested completion date
reference_images	TEXT[]	An array of image URLs uploaded as visual references
rejection_reason	TEXT	Required explanation when a seller rejects an order
is_custom	BOOLEAN	Whether this is a custom order (default: true)
ready_at	TIMESTAMPTZ	Timestamp when the seller marked the order as ready
completed_at	TIMESTAMPTZ	Timestamp when the client confirmed completion

Table 2.4: Attributes of the orders table.

A few of these fields need extra explanation: The `budget_min` and `budget_max` fields let the client communicate what they are willing to spend, but the actual price is decided by the seller later, which is what the `final_price` column is for. The seller fills it in when they mark the order as ready, and the client sees it before confirming completion. The `rejection_reason` field is not optional when the status changes to "rejected". The backend enforces this through validation because we wanted clients to always understand why their request was turned down, not just see a generic rejection. The `reference_images` field is a PostgreSQL text array that stores Supabase Storage URLs for any photos the client uploaded to illustrate what they have in mind. The status field is constrained to five values, and the backend enforces strict transitions between them: pending can only become accepted or rejected, accepted can only become ready, ready can only become completed, and both rejected and completed are terminal states. This is checked in the service layer through a `VALID_TRANSITIONS` constant before any update is applied.

Each order is connected to the product it refers to through the `order_items` table, which records the product ID, quantity, and a snapshot of the unit price at the time the order was placed.

Trust system entities

We set up three separate tables for ratings, which might seem like a lot, but each one covers a different relationship in the trust system. The `reviews` table handles product-level feedback: after a completed order, the client can leave a rating from 1 to 5 along with a written comment. A uniqueness constraint (`product_id`, `client_id`) makes sure a client can only review the same product once. The `ratings` table works at the seller level, also submitted by clients after a completed order, with a similar uniqueness constraint (`seller_id`, `client_id`). The `client_ratings` table goes in the opposite direction; it lets sellers rate the clients they worked with. This table is keyed by (`order_id`, `seller_id`), meaning a seller can only rate a client once per completed order, and ratings are only allowed when the order status is "completed."

This two-way system was one of our design goals from the beginning. It directly addresses the unreliable-client problem we described in Chapter 1: sellers can now check a client's rating history before deciding whether to accept their order, and clients who repeatedly ghost or abandon orders will have a visible record. Both the `reviews` and `ratings` tables have associated database triggers (`trg_product_avg_rating` and `trg_seller_avg_rating`) that automatically recalculate the average rating on the corresponding product or seller whenever a review is inserted, updated, or deleted.

Supporting entities

The remaining tables handle features that support the main workflow. The `wishlist` table lets any authenticated user, whether client or seller, bookmark products they might want to order later. The `notifications` table stores alerts about order status changes (`new_order`, `order_accepted`, `order_rejected`, `order_ready`, `order_completed`, and system notifications), and includes a meta JSONB field for structured data like the order ID or final price. The `categories` table holds the nine craft categories defined in Chapter 1, each with a name, a URL-friendly slug, and an optional icon. The `subscriptions` table manages seller plans; only the free tier is active at the MVP stage, but the structure is there for future upgrades; a database

trigger (`trg_seller_subscription`) automatically creates a free subscription row whenever a new seller profile is inserted. The `promotions` table handles featured product placements, with fields for the placement location (hero or browse), the requested duration in days, the current status (pending, active, expired, or rejected), and a rejection reason if the admin declines the request. Finally, the `browsing_events` table records user interactions like product views, with a nullable `user_id` that allows tracking anonymous visitors as well. This data could feed into analytics or a recommendation system in the future.

Database triggers

The database relies on four automatic triggers that run without any intervention from the application code. The first (`on_auth_user_created`) fires after a new user signs up through Supabase Auth and creates a matching row in the `public users` table. The second (`trg_product_avg_rating`) fires after any insert, update, or delete on the `reviews` table and recalculates the `avg_rating` field on the corresponding product. The third (`trg_seller_avg_rating`) does the same thing for seller ratings. The fourth (`trg_seller_subscription`) fires after a new seller profile is created and generates a default free-tier subscription entry for that seller.

Figure 2.5 — Entity-Relationship Diagram

16 main tables + 2 auth tables (otp_sessions, password_reset_tokens)

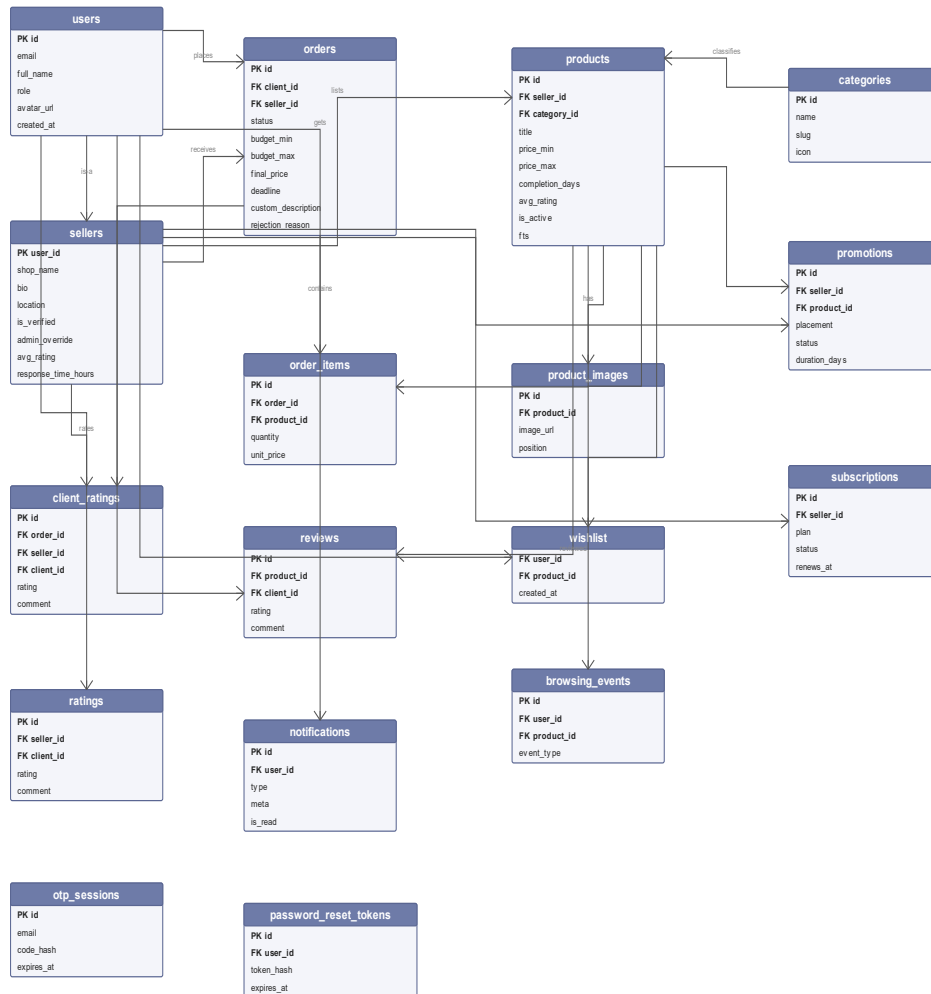


Figure 2.5: Entity-relationship diagram of the Hirftna database

Figure 2.5 below shows the complete relational structure of the Hirftna database in a single visual. Each rectangle represents a table, with its primary key marked PK and its foreign keys marked FK. The connecting lines show how the tables relate to one another: a single line ending in a small fork (the "crow's foot" notation) indicates the "many" side of a one-to-many relationship, while the simple bar ending indicates the "one" side. Reading the diagram from left to right, four logical clusters become visible: the user cluster (users, sellers, otp_sessions, password_reset_tokens) on the upper left, the catalog cluster (products, categories) in the center, the order cluster (orders, order_items) below it, and the trust cluster (reviews, seller_ratings, client_ratings) connected to both users and orders. Supporting tables (wishlist, notifications,

promotions, chatbot_messages) are attached to the user table because they all belong to a specific user account. The two-way arrows between the users table and the orders, reviews, and rating tables visualise the mutual rating system: a single user can appear both as a client and as a seller in different relationships, which is what makes the trust system symmetric.

2.2.2. API Endpoints

The backend exposes a RESTful API that the frontend uses for every action on the platform. All routes are prefixed with `/api/v1/`, and they follow a consistent pattern: the resource name first, then an optional ID or action suffix. The router module receives each incoming HTTP request, determines which middleware should process it, and forwards it to the appropriate controller. In the tables below, we present the endpoints organized by feature area. The "Access" column indicates whether the endpoint is public, requires authentication, or is restricted to a specific role.

Authentication: These endpoints handle the full authentication lifecycle, including account creation, login, token management, password reset, and two-factor verification.

Method	Endpoint	Result	Access
POST	/auth/register	Create a new account with role selection	Public
POST	/auth/login	Log in and receive a JWT token	Public
POST	/auth/verify-otp	Verify the OTP code sent during registration	Public
POST	/auth/logout	End the current session	Authenticated
GET	/auth/me	Get the current user's profile	Authenticated
PUT	/auth/me	Update profile information	Authenticated
POST	/auth/change-password	Change the current password	Authenticated
POST	/auth/refresh	Refresh an expired access token	Authenticated
POST	/auth/forgot-password	Send a password reset link by email	Public
POST	/auth/reset-password	Reset the password using the emailed token	Public

Table 2.5: Authentication API endpoints.

Categories: Categories organize products into craft types. They are publicly browsable, but only the admin can create, update, or delete them.

Method	Endpoint	Result	Access
GET	/categories	Get all craft categories	Public
GET	/categories/slug/:slug	Get a category by its URL slug	Public
GET	/categories/:id	Get a category by its ID	Public
POST	/categories	Create a new category	Admin
PUT	/categories/:id	Update an existing category	Admin
DELETE	/categories/:id	Delete a category	Admin

Table 2.6: Category API endpoints.

Products: Anyone can browse products, but only sellers can create and manage their own listings. Verified sellers display a trust badge on their listings. All active products appear in the public catalog regardless of the seller's verification status.

Method	Endpoint	Result	Access
GET	/products	Browse products with filters, search, and pagination	Public
GET	/products/:id	Get a product's full details, images, and seller info	Public
GET	/products/my-products	Get the seller's own product listings (including inactive)	Seller
POST	/products	Create a new product with images, a price range, and a category	Seller
PUT	/products/:id	Update an existing product (ownership checked)	Seller
DELETE	/products/:id	Delete a product listing (ownership checked)	Seller

Table 2.7: Product API endpoints.

Orders: These are the core of the custom order system. Access is role-scoped: clients create orders and confirm completion; sellers accept, reject, and mark orders as ready. The GET endpoints return different results depending on the user's role. Clients see their placed orders; sellers see incoming ones.

Method	Endpoint	Result	Access
POST	/orders	Submit a custom order request for a product	Public

GET	/orders	List orders (scoped by role: client sees own, seller sees incoming)	Public
GET	/orders/:id	Get the full details of a specific order (ownership enforced)	Public
PATCH	/orders/:id/status	Accept or reject an incoming order	Seller
PATCH	/orders/:id/ready	Mark the order as ready and set the final price	Seller
PATCH	/orders/:id/complete	Confirm completion after reviewing the final price	Client

Table 2.8: Order API endpoints.

Sellers: Seller profiles are publicly visible, allowing buyers to browse artisans and read their stories. Managing the shop itself, analytics, verification status, and profile editing requires the seller role.

Method	Endpoint	Result	Access
GET	/sellers	Browse sellers (verification badge shown)	Public
GET	/sellers/:id	Get a seller's public profile, products, and ratings	Public
GET	/sellers/me	Get the seller's own full profile	Seller
GET	/sellers/analytics	Get shop statistics: orders, revenue, top products	Seller
POST	/sellers	Create a new shop profile	Seller
PUT	/sellers/:id	Update shop name, description, story, and avatar (ownership checked)	Seller

Table 2.9: Seller API endpoints.

Reviews and Ratings: These endpoints manage the two-way trust system described in Chapter 1. Clients rate products and sellers after a completed order. Sellers rate clients after the same completed order. All ratings are tied to a specific order, and uniqueness constraints prevent duplicate entries.

Method	Endpoint	Result	Access
GET	/reviews/product/:id	Get reviews for a specific product	Public
GET	/reviews/seller/:id	Get ratings for a specific seller	Public
POST	/reviews/product	Write a review for a product (after completing the order)	Client
POST	/reviews/seller	Rate a seller (after a completed order)	Client
DELETE	/reviews/:id	Delete the user's own review	Client
POST	/client-ratings	Rate a client after a completed order	Seller
GET	/client-ratings/client/:id	Get ratings received by a specific client	Public

Table 2.10: Reviews and ratings API endpoints.

Supporting Features: These endpoints handle the wishlist, notifications, file uploads, and the AI chatbot. They are available to any authenticated user.

Method	Endpoint	Result	Access
GET	/wishlist	Get the user's list of saved products	Authenticated
POST	/wishlist	Add a product to the wishlist	Authenticated
GET	/wishlist/:product_id/check	Check if a product is already saved	Authenticated
DELETE	/wishlist/:product_id	Remove a product from the wishlist	Authenticated
GET	/notifications	Get notifications with pagination	Authenticated
GET	/notifications/unread-count	Get the count of unread notifications	Authenticated
PATCH	/notifications/:id/read	Mark a single notification as read	Authenticated
PATCH	/notifications/mark-all-read	Mark all notifications as read	Authenticated
DELETE	/notifications/:id	Delete a notification	Authenticated
POST	/uploads/images	Upload a single image to Supabase Storage	Authenticated
POST	/uploads/image	Upload up to 5 images at once	Authenticated
POST	/chatbot	Send a message to the AI assistant	Authenticated

Table 2.11: Supporting feature API endpoints.

Administration: All admin endpoints are restricted to users with the admin role, which can only be assigned directly in the database. The API endpoint for changing roles only accepts "client" or "seller" as values.

Method	Endpoint	Result	Access
GET	/admin/users	List all users with role filter, search, and pagination	Admin
GET	/admin/products	List all products, including inactive ones	Admin
GET	/admin/stats	Get platform statistics for users, orders, revenue, and top sellers	Admin
PATCH	/admin/sellers/:id/verify	Verify or revoke a seller's verification status	Admin
DELETE	/admin/products/:id	Force-delete any product from the platform	Admin
PATCH	/admin/users/:id/role	Change a user's role to client or seller	Admin
GET	/admin/promotions	List promotion requests (filterable by status)	Admin
PATCH	/admin/promotions/:id/activate	Approve a promotion (sets start and end dates)	Admin
PATCH	/admin/promotions/:id/reject	Reject a promotion with a reason	Admin

Table 2.12: Administration API endpoints.

2.2.3. Middleware and Security

Security was not something we bolted on at the end. It was built into the request pipeline from the start. In Express.js, a middleware function sits between the incoming request and the route handler, with access to both the request and response objects. It can either modify them and pass control to the next function with `next()`,

or stop the cycle by sending a response itself. The order matters: Express runs middleware sequentially, so a request that fails at one step never reaches the next [16].

Every request passes through a global stack before it reaches any route:

- **Helmet** sets security-related HTTP headers, for example, disabling X-Powered-By so the server does not advertise its framework.
- **CORS** restricts requests to allowed origins (localhost:5173 in development, only hirftna.vercel.app [1] in production).
- **HPP** blocks HTTP parameter pollution, where the same query parameter is sent multiple times to confuse the server.
- **Compression** applies gzip to responses to reduce bandwidth.
- **Body Parser** reads the JSON body with a 100KB limit to reject oversized payloads.
- **Morgan** logs each request with its method, URL, status code, and response time. This was the most useful one during development, since seeing every failed request in the terminal saved a lot of guessing.
- **Rate Limiter** caps requests per IP (Intellectual Property) across three tiers: 100 per 15 minutes globally, 10 per 15 minutes on authentication routes (a common brute-force target), and 20 per hour on the chatbot endpoint to stay within the Gemini free tier. [17]
- **JWT authentication** verifies the token from the Authorization header with Supabase Auth and attaches the user's identity to the request. An invalid token returns 401 [18]. One rule we kept throughout the project: we never trust user IDs sent in the request body. The identity always comes from the verified token.
- **Role checking** compares the user's role against what the route requires and returns 403 on a mismatch.
- **Input validation** runs the body against a Zod schema and returns a clear 400 with field-level errors if anything is missing or invalid, before the request ever reaches the business logic.

As a final safety net, we use Row Level Security (RLS) at the database level. RLS is a PostgreSQL feature that defines access policies directly on tables, controlling which rows a user can read or modify based on their identity. With Supabase helpers like `auth.uid()`, these become simple SQL (Structured Query Language) conditions such as "only allow updates where `seller_id` matches the current user." The benefit is defense in depth: if a bug ever lets a request slip past the middleware, the database itself still blocks the unauthorized operation. Admin actions use the Supabase service role key, which bypasses RLS but lives only as a backend environment variable and is never exposed to the frontend. Two-factor authentication and password reset are covered in Chapter 3. [19] [18]

2.3. Frontend Application

The frontend is the part of the system that users actually see and interact with. It is built as a single-page application (SPA) using React 19 with Vite as the build tool. In an SPA, the browser loads a single HTML file at the start, and after that, all navigation happens on the client side. React swaps components in and out based on the current route without reloading the entire page [20] This makes the experience feel faster compared to traditional websites, where every click triggers a full server round-trip.

We chose React because of its component-based approach: the interface is broken down into small, self-contained pieces, a product card, a navigation bar, an order status badge, a rating widget, each managing its own logic and rendering [21]. This made it easier to build things one piece at a time without worrying about breaking something else. React's virtual DOM also helps with performance: it calculates the minimum set of actual DOM changes needed before touching the page, which keeps things responsive even as the number of products and orders grows [22].

In this section, we describe the pages available to each user role, the use case diagrams for all four actors, the main workflows through flowcharts, and the sequence diagrams that trace the key interactions step by step.

2.3.1. Pages

The application is organized into pages, each mapped to a URL through React Router v7. Not all pages are accessible to everyone; some are public, some require the user to be logged in, and some are restricted to a specific role. We enforce this through route guards built into the router configuration: `RequireAuth` blocks unauthenticated users, `RequireNotAdmin` blocks admin users from regular user pages (they are redirected to the admin panel), `RequireSeller` checks for the seller role, `RequireAdmin` checks for the admin role, and `GuestOnly` prevents logged-in users from seeing pages like login or registration that they no longer need.

Table 2.13 lists the public pages accessible to everyone, including visitors without an account.

Route	Page	Purpose
/	HomePage	Hero section, featured categories, products, and sellers
/browse	BrowsePage	Search and filter products with pagination
/products/:id	ProductPage	Product details, image gallery, reviews, and order button
/sellers/:id	SellerPage	Seller profile, artisan story, products, and ratings
/404	NotFoundPage	Displayed when a route does not exist

Table 2.13: Public pages.

Table 2.14 shows the authentication pages, accessible only to users who are not logged in (`GuestOnly` guard).

Route	Page	Purpose
/login	LoginPage	Email and password login
/register	RegisterPage	Account creation with role selection and OTP verification
/forgot-password	ForgotPasswordPage	Request a password reset email
/reset-password	ResetPasswordPage	Set a new password using the emailed token

Table 2.14: Authentication pages.

Table 2.15 lists the pages available to any logged-in user (client or seller), protected by the `RequireNotAdmin` guard; admin users are redirected to the admin panel instead.

Route	Page	Purpose
/wishlist	WishlistPage	Saved products with remove option
/orders	OrdersPage	View placed orders and their current status
/profile	ProfilePage	View and edit personal information and avatar

Table 2.15: Client pages (any logged-in user).

Table 2.16 shows the seller-specific pages, protected by the RequireSeller guard.

Route	Page	Purpose
/seller/dashboard	SellerDashboard	Stats overview, recent orders, and quick actions
/seller/products	SellerProducts	Product CRUD (Create, Read, Update, Delete) with image upload and visibility toggle
/seller/orders	SellerOrdersPage	Incoming and outgoing orders
/seller/profile	SellerProfileEdit	Shop details, artisan story, and avatar upload
/seller/promotions	SellerPromotions	About the seller's promotion requests

Table 2.16: Seller pages.

2.3.2. Use Case Diagrams

Use case diagrams give a high-level picture of what each actor can do on the platform. Instead of cramming everything into a single crowded diagram, we present a separate diagram for each of the four actors. Each one is followed by a short description of the main use cases and how they relate to each other.

Visitor Use Cases

Figure 2.6 shows what a visitor using the platform without an account can do. Their interactions are limited to browsing and discovery: viewing the product catalog with search and filtering, opening individual product pages to see images and reviews, and reading seller profiles along with their artisan stories. The registration use case extends into role selection (client or seller) and, if the user chooses to register, includes OTP email verification.

Figure 2.6 — Visitor Use Case Diagram

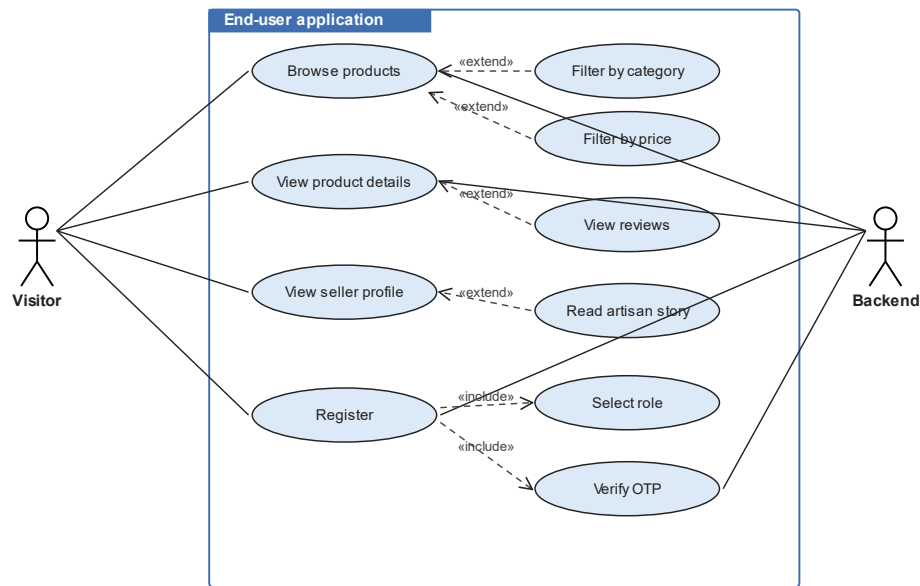


Figure 2.6: Visitor use case diagram.

Client Use Cases

Figure 2.7 shows the client's use cases. The client is a registered user who places custom orders and interacts with sellers through the structured order workflow. The "Submit custom order" use case is the most involved, as it includes specifying requirements, setting a budget range, choosing delivery and payment preferences, and uploading reference images. "Confirm completion" includes reviewing the final price set by the seller. After completion, the client can rate the seller and write a product review. The wishlist and chatbot are independent use cases available to any authenticated client.

Figure 2.7 — Client Use Case Diagram

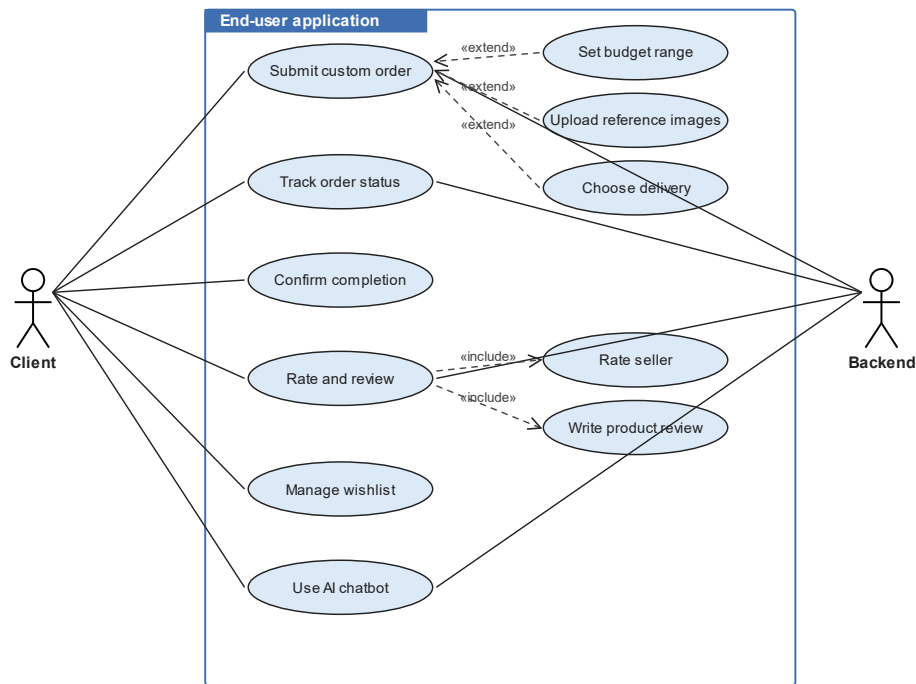


Figure 2.7: Client use case diagram

Seller use case diagram:

Figure 2.8 shows the seller's use cases. The seller has the most complex set of interactions because of their dual role. They manage their own shop but can also place orders as buyers for other sellers' products. "Manage products" includes creating, editing, deleting, and toggling the visibility of listings. "Respond to orders" covers reviewing client requirements and deciding whether to accept or reject (with a mandatory reason for rejection). "Mark order as ready" includes setting the final price and confirming the delivery method. After a completed order, the seller can rate the client. This is the use case that implements the accountability system from Gap 4. The seller also manages their shop profile, including the artisan story, and can view shop analytics and request promotions for their products.

Figure 2.8 — Seller Use Case Diagram

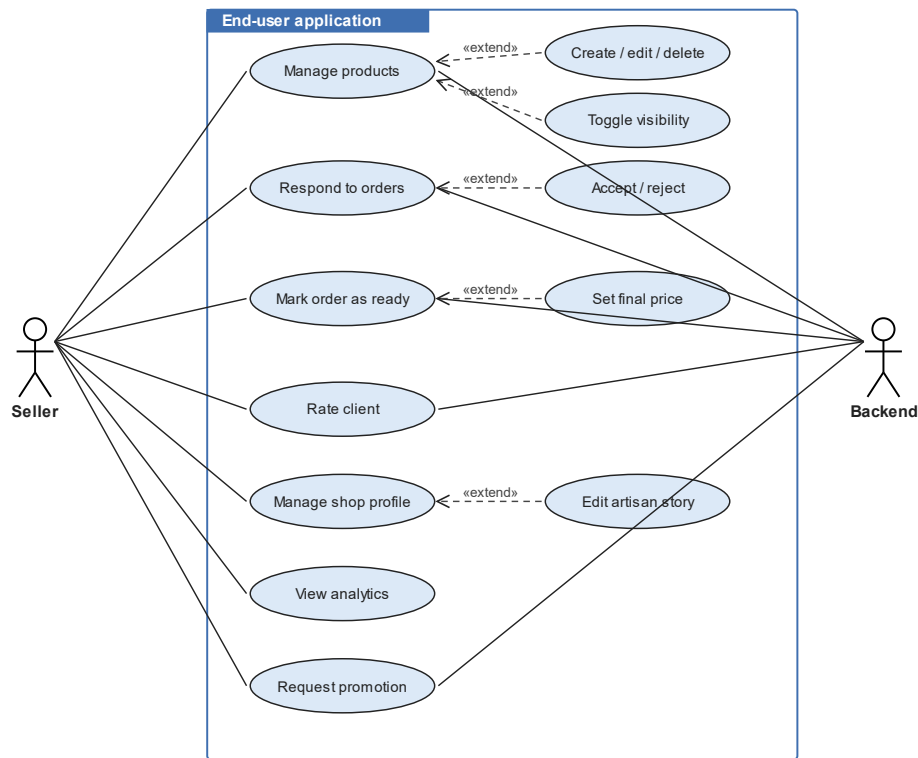


Figure 2.8: Seller use case diagram

2.3.3. Workflow Diagrams

The flowcharts in this section describe the main processes from the user's perspective. We chose three workflows that we consider the most important to visualize: creating an account with OTP verification, placing a custom order as a client, and handling an incoming order as a seller.

Registration and OTP Verification

Figure 2.9 shows the registration process. The visitor clicks "Register," selects their role (client or seller), and fills in the registration form. If they choose the seller role, additional fields appear for the shop name, location, and a short description. After submission, the backend sends a six-digit OTP code to the provided email. The user enters the code on the verification screen. If the code is correct and the session has not expired (10-minute TTL, 5 attempts maximum), the account is activated, and the user is redirected to the login page with a success message. If the code is wrong or the session expires, the user needs to start the registration process over.

Figure 2.9 — Registration and OTP Verification

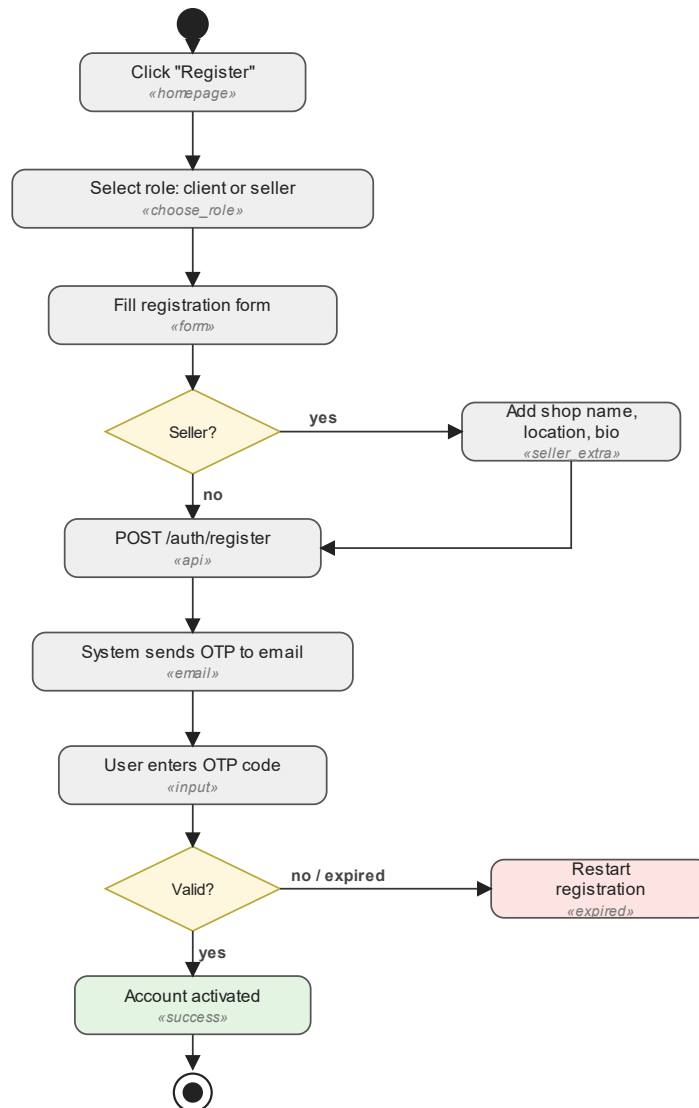


Figure 2.9: Registration and OTP verification process

Browsing and Requesting a Custom Order

Figure 2.10 shows the process a client follows when placing a custom order. The client browses the catalog, opens a product page, and clicks "Request Custom Order." A modal form opens, split into three steps.

In the first step, the client describes what they need, sets a budget range by entering a minimum and maximum amount in dinars, selects a deadline, and can optionally upload reference images to give the seller a visual idea. The second step asks for the delivery type (fast delivery, office pickup, or hand-to-hand) and the

payment method (card or cash on delivery). The third step collects the client's name, phone number, and delivery address. Each step runs its own validation before the client can advance. If a required field is empty or the input is invalid (for instance, the minimum budget is higher than the maximum), the form highlights the problem and blocks progression.

Once all three steps are complete and the client submits, the backend creates the order in the database with the status "pending" and sends a notification to the seller that a new request has arrived.

Figure 2.10 — Custom Order Request Process

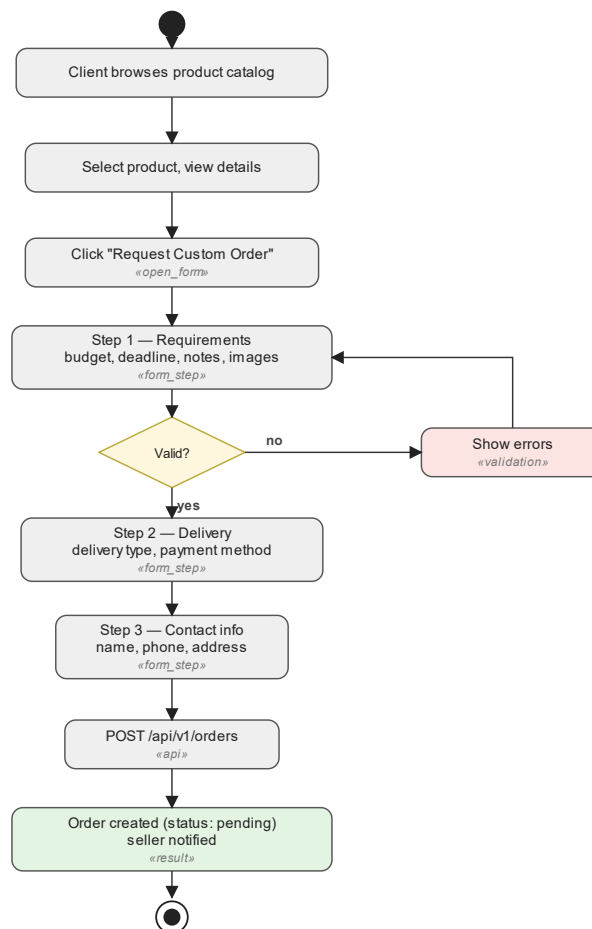


Figure 2.10: Custom order request process

Managing Orders and Reviews

Figure 2.11 shows the order management process from the seller's perspective. When a new order is submitted, the seller receives a notification. They open their

orders page and review the client's request, the budget range, deadline, notes, and reference images.

The seller makes a decision. If they accept, the order status changes to "accepted," and the client is notified. The seller works on the product, and when the work is finished, they mark the order as "ready." This step requires entering the final price and confirming the delivery method. The client receives another notification, opens the order to see the final price and delivery details, and if satisfied, confirms completion. The order moves to "completed."

If the seller rejects the order, they must write a reason that the backend validation does not accept a rejection without one. The order moves to "rejected," the client is notified with the seller's explanation, and the order becomes unmodifiable from that point. Both "rejected" and "completed" are terminal states.

After an order reaches "completed," both sides can rate each other. The client can leave a product review and a seller rating. The seller can rate the client. Neither is mandatory, but since both are publicly visible on the respective profiles, there is a practical incentive for both sides to participate, and for clients in particular, a good rating history makes future sellers more likely to accept their orders.

Figure 2.11 — Managing Orders and Reviews Process

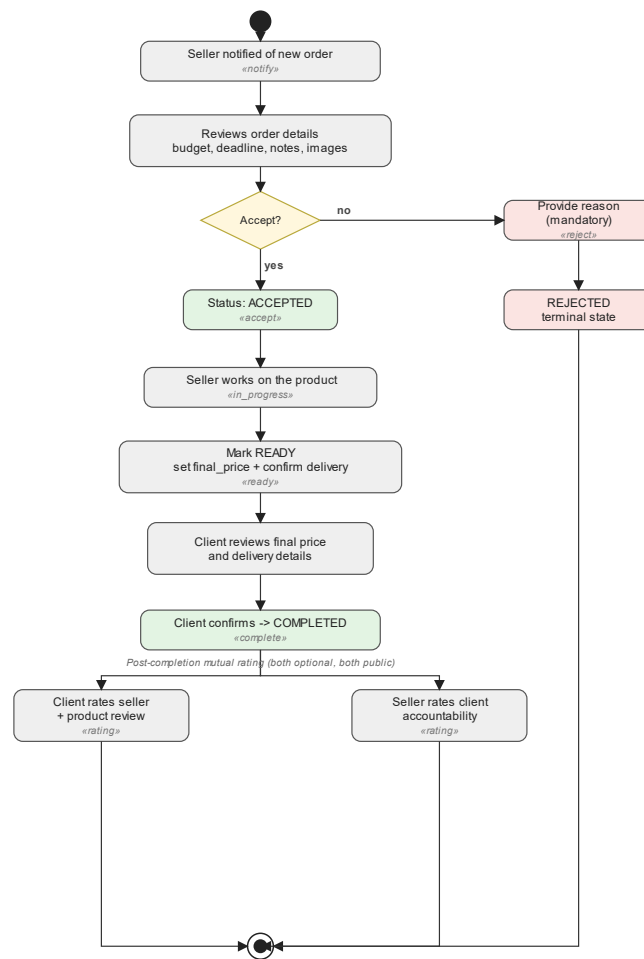


Figure 2.11: Managing orders and reviews process

2.4. Admin Dashboard

The admin dashboard is a separate section of the application reserved for platform management. It is built within the same React application but uses its own layout and is only accessible to users with the admin role. We made a deliberate decision not to allow the admin role to be assigned through the API; the endpoint for changing user roles only accepts "client" or "seller," so even if someone found a way past the other security layers, they could not grant themselves admin access.

2.4.1. Pages

The admin side of the application consists of five main pages, each focused on a different aspect of platform management. Table 2.17 lists them.

Route	Page	Description
/admin	AdminDashboard	Platform statistics, charts, top sellers, and products
/admin/users	AdminUsers	User list with role filter, search, and seller verification
/admin/products	AdminProducts	All products, including inactive ones, will be force-deleted
/admin/promotions	AdminPromotions	Promotion requests from sellers, with approve and reject actions
/admin/categories	AdminCategories	Full CRUD for craft categories with auto-slug generation

Table 2.17: Admin dashboard pages.

The statistics page gives the admin a quick overview of how the platform is doing, total users, products, orders, and revenue, along with charts showing the top five sellers by sales and the top five products by views. The users page lists every registered account with role filters, a search bar, and the ability to verify or revoke seller accounts and change user roles between client and seller. The products page shows all listings on the platform, including ones that sellers have hidden from public view, and lets the admin force-delete content that violates the platform's policies. The categories page provides full CRUD operations for craft categories. Each category has a name, a URL-friendly slug (auto-generated from the name), and an optional icon. The nine default categories were pre-filled when we set up the database, but the admin can create new ones or modify existing ones at any time. The promotions page lists all promotion requests submitted by sellers, filterable by status (pending, active, expired, rejected), and provides the tools to approve or reject each request.

2.4.2. Use Cases

Figure 2.15 shows the admin's use case diagram. The admin's interactions cover five areas. "Manage users" extends to viewing the user list, changing roles, and verifying or revoking sellers. "Manage products" includes viewing all listings and force-deleting products. "Manage categories" covers creating, editing, and deleting craft categories. "Manage promotions" extends into viewing requests, approving them (which sets the start and end dates based on the requested duration), and rejecting them with a reason. "View statistics" gives the admin access to platform-

wide metrics and performance data. All of these actions require authentication through the admin role.

Figure 2.15 — Admin Use Case Diagram

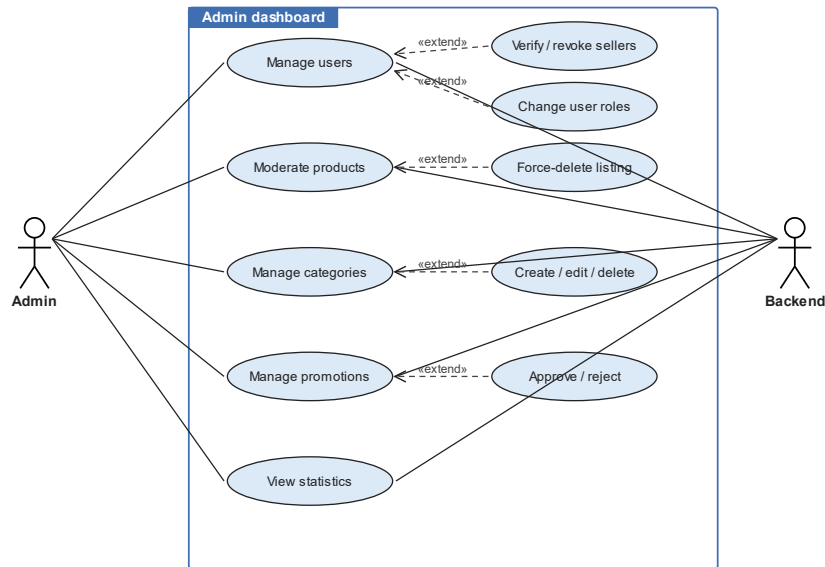


Figure 2.15: Admin use case diagram

2.4.3. Sequence Diagrams

The sequence diagrams in this section trace the interactions between the admin, the dashboard interface, the backend, and the database for the main administrative workflows.

Display and Manage Users

Figure 2.16 shows how user management works. When the admin opens the users page, the dashboard sends a GET /admin/users request to the backend, which queries the database with pagination and returns the user list. The admin can filter by role or search by name. When they click "Verify seller," the backend updates the seller's `is_verified` field to true, sets the `admin_override` column so the automatic verification system cannot reverse this decision, and creates a notification record for the seller informing them that their shop has been activated.

Figure 2.16 — Display and Manage Users (Sequence)

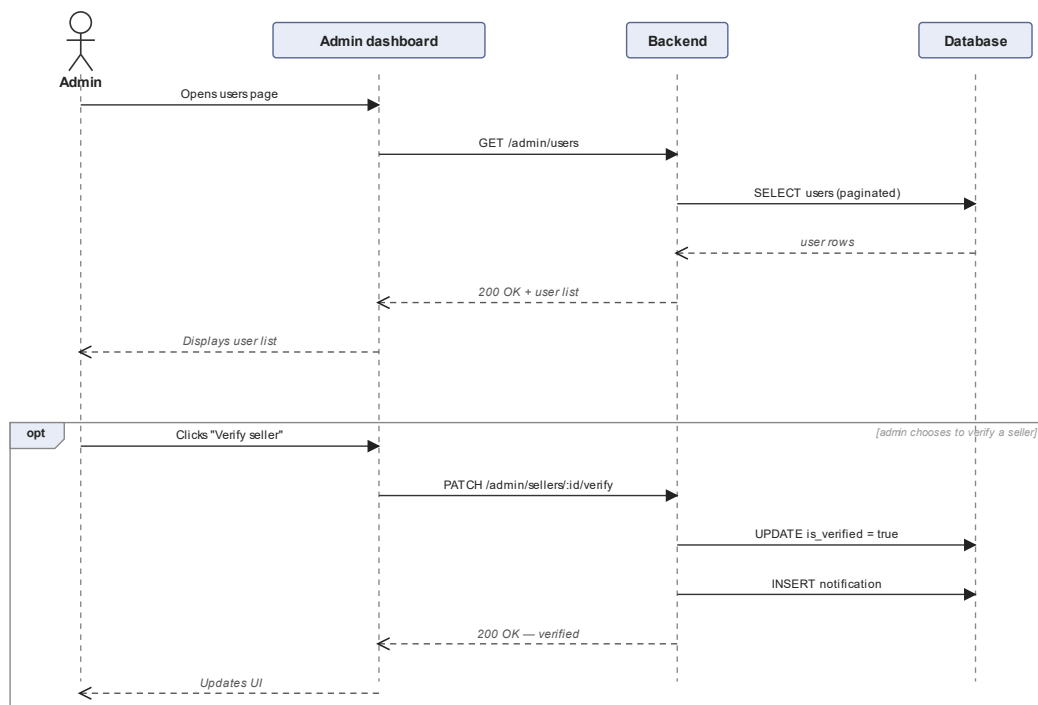


Figure 2.16: Display and manage users sequence diagram

Manage Products

Figure 2.17 shows the product management flow. The admin views all products on the platform, including inactive ones that sellers have hidden. When the admin force-deletes a product that violates the platform's rules, the dashboard sends a DELETE request to the backend, which removes the product from the database and returns a confirmation.

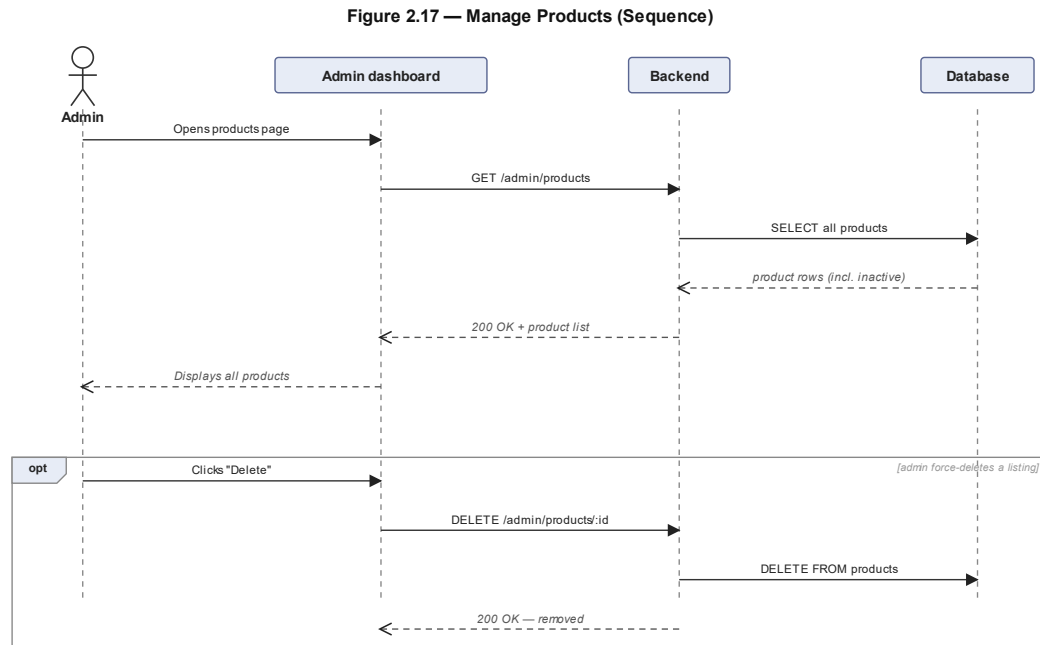


Figure 2.17: Manage products sequence diagram

Manage Categories

Figure 2.18 shows how the admin creates and edits craft categories. To create a new category, the admin fills in a form with the category name, the slug is auto-generated from the name, and the backend inserts a new row in the categories table. Editing works the same way: the admin updates the name or slug, and the backend applies the change. Categories can also be deleted, though this should be done carefully since products may still reference them.

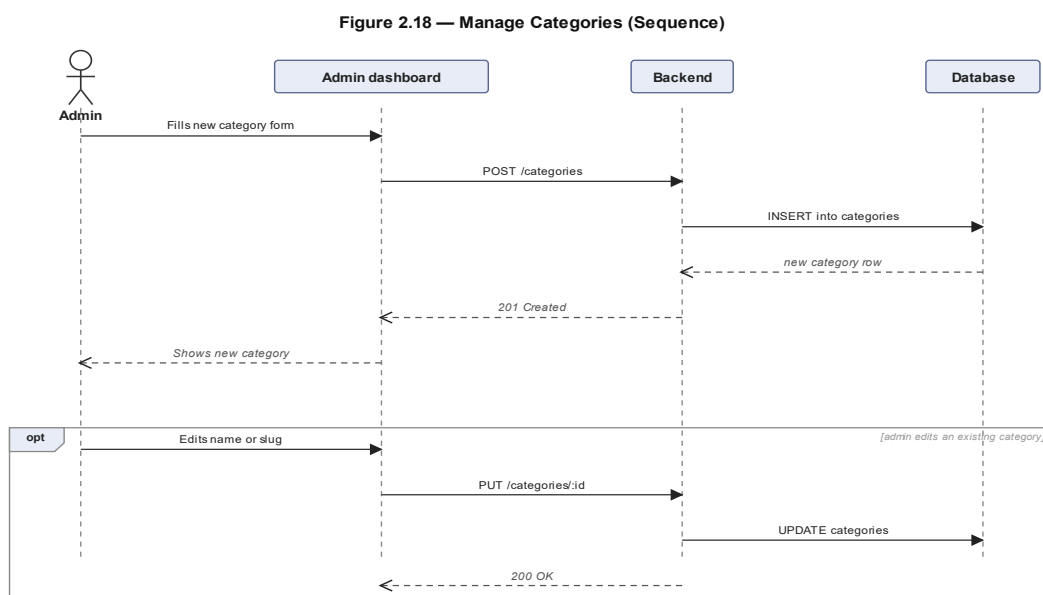


Figure 2.18: Manage categories sequence diagram

Manage Promotions

Figure 2.19 shows the promotion management workflow. The admin opens the promotions page and sees all requests, filterable by status. When they approve a request, the backend sets the start date to now and calculates the end date based on the number of days the seller requested (7, 14, or 30 days), marks the promotion as active, and sends a notification to the seller. If the admin rejects a request, they write a reason, and the seller receives a notification explaining why.

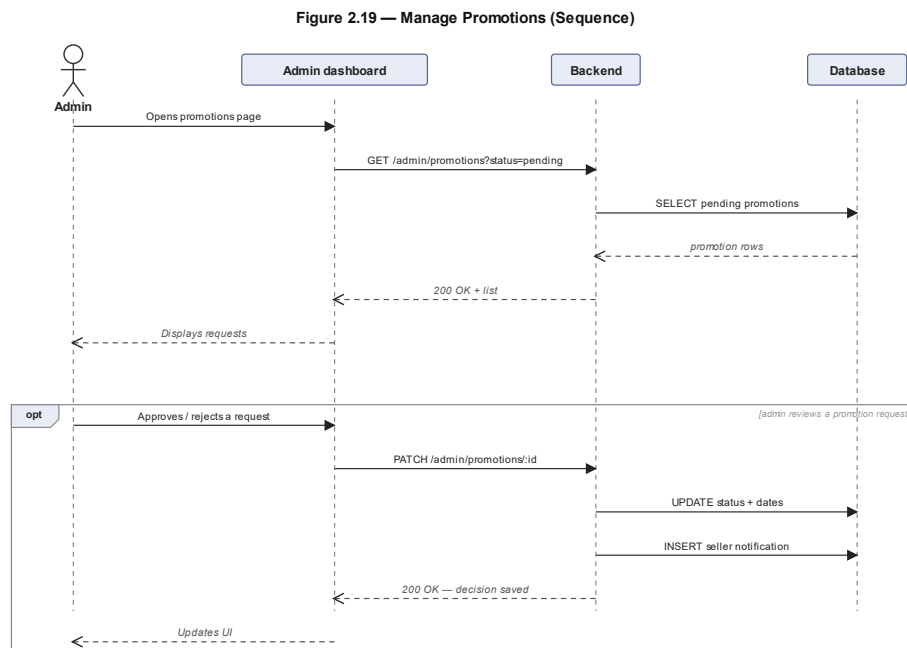


Figure 2.19: Manage promotions sequence diagram

View Platform Statistics

Figure 2.20 shows the statistics retrieval flow. When the admin opens the main dashboard page, the frontend sends a GET /admin/stats request. The backend runs several queries: counting total users, products, and orders, summing revenue from completed orders, and selecting the top five sellers and products by rating and views. The aggregated results are sent back to the dashboard, which displays them as stat cards, bar charts, and ranking tables.

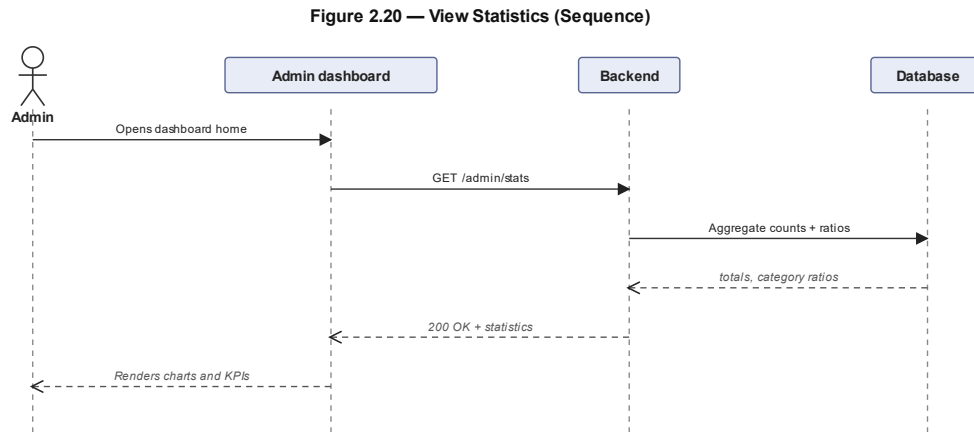


Figure 2.20: View platform statistics sequence diagram

Conclusion

This chapter presented the complete technical design of the Hirftna platform. We started with the architecture, a client-server model with a React frontend, an Express.js backend, and a Supabase-managed PostgreSQL database, and explained how the three layers communicate through a RESTful API. The data model covers 16 main tables plus 2 authentication-related tables, with database triggers that automate rating recalculations and user provisioning. The API is organized into 50+ endpoints across authentication, products, orders, sellers, reviews, client ratings, notifications, wishlist, uploads, chatbot, and administration. Security is enforced at multiple levels: middleware-based JWT authentication and role checking, Zod input validation on every request, and row-level security at the database level as a final safety net.

On the frontend side, we described the pages organized by role, the use case diagrams for all four actors, the main workflows through flowcharts, and the sequence diagrams that trace the key administrative interactions. The order lifecycle from pending through accepted, ready, and completed, with rejected as an alternative terminal state, is enforced through strict state transitions in the backend, and the mutual rating system that follows completion creates accountability on both sides of the transaction.

In the next chapter, we move from design to implementation: the development tools, the external services, how we deployed the application, and the actual user interfaces of the platform.

Chapter 3

System Implementation

This chapter covers the practical side of building the Hirftna platform. We start by going over the tools and technologies we used during development, the environment, the backend and frontend stacks, the external services, and how we deployed everything. After that, we present the application's main interfaces through screenshots, organized by user role. The point is to show how the requirements and design from the earlier chapters turned into something that actually works.

3.1 Used Technologies

This section presents the different technologies we used to build the Hirftna system. We cover the development environment first, then the backend and frontend stacks, and finally the external services and deployment.

3.1.1. Development Environment

Our development environment consists of the following tools:

- **Visual Studio Code (VS Code):** A free, open-source, cross-platform code editor made by Microsoft, available on Windows, macOS, and Linux [23]. It combines a lightweight editing experience with features like IntelliSense autocompletion, a built-in terminal, Git integration, and a debugger [24]. We added extensions for JavaScript linting, Tailwind CSS class suggestions, and REST API testing, which ended up saving us a lot of repetitive work.
- **Postman:** An API platform for building, testing, and documenting APIs [25]. We tested every backend endpoint in Postman before wiring it to the frontend. Our requests were grouped into collections by feature: authentication, products, orders, reviews, and admin, so we could easily re-run them after changes. This was especially useful for the custom order flow, where we had to make sure that invalid status transitions (for example, jumping from "pending" straight to "completed") were properly rejected.

- **Git:** A free, open-source distributed version control system used to track code changes across projects of all sizes [26] . We used it throughout the entire project for all version control.
- **GitHub:** A platform for hosting and collaborating on software projects using Git [27]. Our repository lived on GitHub, which gave us a commit history we could go back through whenever we needed to trace a bug or revert a change. We did not use a complicated branching strategy; with a small team, a simple workflow was more than enough. GitHub also became part of our deployment, since both Vercel and Render redeploy automatically whenever we push to the main branch.
- **StarUML:** StarUML is a sophisticated software modeler aimed at supporting agile and concise modeling. We used it to do the project conception [28].

3.1.2. Development Stack

Backend Stack

The Hirftna backend is a REST API responsible for authentication, product management, the custom order lifecycle, reviews, notifications, and the rest of the platform's business logic. Here are the technologies it is built on:

- **Node.js (v20 LTS):** A free, open-source JavaScript runtime environment that runs on Windows, macOS, and Linux [29]. We used it as the base for our server. Going with Node.js meant we could write JavaScript on both sides of the project, frontend and backend, which kept things simpler for us. Its non-blocking I/O model also fits well with what our API does most of the time: querying the database and calling external services.
- **Express.js (v5):** A web application framework for Node.js, known for being minimal and flexible [16]. It gave us a clean way to set up routes, plug in middleware for things like authentication and validation, and organize the codebase into controllers, services, and route files. Version 5 also handles errors thrown inside async route handlers on its own, so we did not have to wrap every controller in a try/catch.

- **PostgreSQL (via Supabase):** All application data is stored in a PostgreSQL 15 database managed through Supabase [18]. Supabase is an open-source platform built around PostgreSQL that bundles authentication, file storage, real-time subscriptions, and auto-generated APIs into one service. Using it meant we got a proper relational database with foreign keys, constraints, triggers, and row-level security, but we did not have to worry about server setup or maintenance. We describe Supabase's other roles in Section 3.1.3.
- **Zod (v4):** A schema declaration and validation library [30]. Every POST, PUT, and PATCH request that hits our API goes through a Zod schema first. We defined the expected shape of each request body, query, and URL parameter in one place. If something does not match, the request gets rejected with a clear, field-level error message before any service code runs.
- **Winston:** A logging library for Node.js [31]. We used it to write structured logs with timestamps and severity levels (info, warn, error). During development, this made it much easier to figure out what the server was doing when something went wrong, especially for errors that did not surface in the API response. We also configured it to redact sensitive fields like passwords and tokens before anything gets written to the log files.
- **JWT (JSON Web Tokens):** An open standard (RFC 7519) for compact tokens used to verify identity and exchange information securely [32]. After login, Supabase issues a JWT that the frontend attaches to every request. On the backend, our auth middleware checks this token on protected routes and pulls the user's identity from it. We never rely on user IDs coming from the client side.
- **Nodemailer:** A Node.js module for sending emails over SMTP [33]. We use it in two places: sending OTP codes by email during registration (when two-factor authentication is enabled) and sending password reset links. The reset tokens are 32-byte hex strings that expire after 15 minutes.

Beyond these, the backend uses a few smaller libraries that are worth a mention: Helmet for secure HTTP headers, express-rate-limit for per-IP and per-user request limits, Multer together with the file-type library for validating uploaded

images by their actual magic bytes (not just the file extension), and Sharp for resizing those images before they go to storage.

Frontend Stack

The frontend is a single-page application that talks to the backend over HTTP. It was built with:

- **React 19:** A JavaScript library for building user interfaces with components and hooks. Every page on the platform, the product catalog, the custom order form, the seller dashboard, and the admin panel is built with React using functional components only. Breaking the UI (User Interface) into components (product cards, status badges, rating modals, and so on) helped us reuse things across pages and keep each piece's logic self-contained. [34]
- **Vite:** A frontend build tool built for speed [35]. We picked it over Create React App mainly because hot module replacement is noticeably faster, which matters a lot when you are constantly tweaking the UI. The production builds it produces are also well-optimized without needing extra configuration, and the output is a plain `dist/` folder of static files, which is exactly what our host expects.
- **Tailwind CSS:** A utility-first CSS framework [36]. Rather than writing separate stylesheets, we applied styles directly through utility classes in our JSX, things like `bg-cream-50`, `text-warm-800`, and `rounded-xl`. This kept everything consistent with the design system we defined (the warm cream and sage green palette from Chapter 2). Because the platform is Arabic-first, we used Tailwind's logical properties (`ps`, `pe`, `start`, `end`) instead of left/right ones, so the whole layout flips correctly when the user switches to Arabic. For patterns that came up often, like buttons or cards, we created custom component classes in our main CSS file to avoid repeating long class strings everywhere.
- **React Router (v7):** A routing library for React [37]. We used its `createBrowserRouter` setup to handle navigation and set up route guards based on the user's authentication status and role. Public routes (the catalog, seller profiles) are open to everyone. Protected routes are split further: some need any logged-in user, some need the seller role specifically, and the admin

section requires the admin role. There is also a guard that does the opposite; it stops an admin account from opening the regular shopping pages, since the two interfaces are not meant to mix.

- **i18next:** An internationalization framework for handling translations and language detection [38]. Our platform supports Arabic and English. When the user switches to Arabic, the entire layout flips to right-to-left. All text in the interface goes through i18next, so adding another language in the future would just mean creating a new translation file.
- **Axios:** A promise-based HTTP client [39]. We created a centralized API module (`api.js`) that wraps all backend calls. It handles attaching the JWT to requests, normalizing response formats, refreshing the token automatically when a 401 comes back, and catching errors in a consistent way. No component makes raw HTTP calls; everything goes through this module.

3.1.3. External Services

The platform also depends on a few external services for things we could not realistically build ourselves in the time we had.

- **Supabase:** The most important external service in our setup. We used four parts of it. First, the PostgreSQL database, where all our data lives across 16 main tables (plus two extra tables for OTP sessions and password-reset tokens) with foreign keys, constraints, and triggers, for example, a trigger that recalculates a seller's average rating whenever a new review is submitted. Second, Supabase Auth for handling registration and login: when a user signs up, Supabase creates a record in its internal `auth.users` table and issues a JWT, and we create a matching row in our own `users` table. Third, Supabase Storage for hosting product images and user avatars in two public buckets. Fourth, Supabase Realtime, which we planned to use for pushing notifications to users; in the current version, we went with polling at a fixed interval instead, because it was simpler to get working reliably for the MVP [18].
- **Google Gemini API:** The chatbot feature runs on the **Gemini 2.5 Flash** model from Google's AI platform [17]. It works as an assistant: it can answer questions about how the platform works, explain the ordering process, or

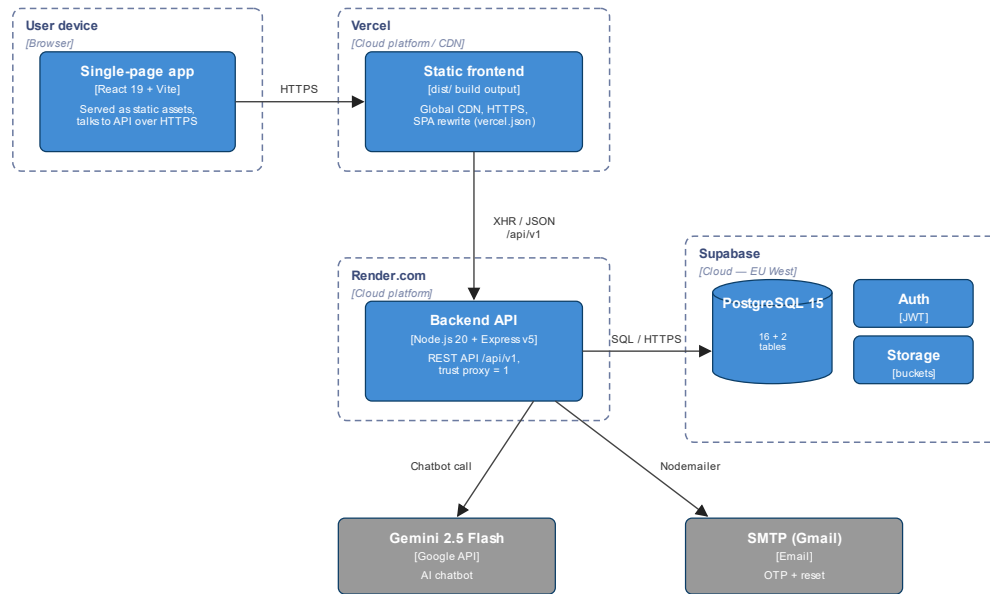
help someone find a product. It is not a communication channel between buyers and sellers; that part is handled entirely by the structured order system. The backend receives the user's message plus a short conversation history, converts the format for Gemini, and returns the response. We set a rate limit of 20 requests per hour per user to stay within the free tier, and the chatbot only appears for users who are logged in.

- **Chargily:** An Algerian payment gateway that supports local payment methods like CIB and Edahabia cards [40]. Since the platform is designed for the Algerian market, international gateways like Stripe are not an option (Stripe does not operate in Algeria). For the MVP, we built the full payment interface on the frontend. The user picks a method and goes through the payment step during order completion, but the actual charge is simulated rather than processed live. The Chargily keys are read from environment variables, so switching from the simulated flow to real processing is mostly a matter of supplying those keys and wiring up the webhook. We made this choice to keep the scope under control while still having the order and payment flow ready end-to-end.
- **SMTP (Gmail):** Email delivery for the OTP codes and password-reset links described above runs over a standard SMTP connection through Nodemailer. During development, we used a Gmail account with an app password, which is enough for the volume an MVP sees.

3.1.4. Deployment

Hirftna runs across three managed cloud services, each handling the part of the system it is best suited for. The frontend is served from Vercel, the backend API runs on Render, and the database, authentication, and storage all live on Supabase. Figure 3.1 shows how these pieces connect in production.

Figure 3.1 — Deployment Diagram



All traffic over HTTPS. Frontend holds no secrets; the service-role key lives only on Render.
 Render free tier spins down after 15 min idle (~30 s cold start); Vercel CDN has no cold start.

Figure 3.1: Deployment diagram

- **Frontend — Vercel:** We deployed the React application on Vercel, a platform for hosting frontend apps with a global CDN and automatic CI/CD [41]. It is connected to our GitHub repository, so every push to the main branch triggers a new build (`npm run build`) and redeploys the `dist/` output automatically. Because the app is a single-page application, we added a small `vercel.json` file with a rewrite rule that sends every path back to `index.html`; without it, refreshing the page on a deep link like `/orders` would return a 404. Vercel serves static files from a CDN, so there is no cold start and the site loads quickly. The application is accessible at hirftna.vercel.app [1].
- **Backend — Render:** The Express API is deployed separately on Render, a cloud platform for running web services [42]. Like Vercel, it redeploys on every push to the main branch, starting the server with `node src/server.js`. One detail that mattered here: on Render, the app sits behind a proxy, so we had to enable `trust proxy` in Express. Without it, every request looks like it comes from the same proxy IP, which would make our per-user rate limiting collapse into one shared bucket for everyone. We are on Render's free tier, which spins the service down after about 15 minutes of inactivity, so the very

first request after an idle period takes roughly 30 seconds while the server wakes back up. After that, responses are fast. The API is reachable at **hirftna-backend.onrender.com**, under the `/api/v1` prefix.

- **Database and services — Supabase:** The PostgreSQL database, authentication, and file storage are all hosted on Supabase's cloud in the EU West region, so there was nothing for us to deploy or maintain on that side. We set up the schema and triggers directly in the Supabase SQL editor, ran our three migration files in order, and created the two public storage buckets. The admin account is the one thing that cannot be created through the API on purpose. We set `role = 'admin'` directly in the database, which removes any chance of someone escalating their own privileges through a request [18].

Splitting the frontend and backend across two hosts instead of one was a deliberate choice. It lets each side scale and redeploy on its own, keeps the static site fast on a CDN, and means a backend cold start never blocks the page from loading; the interface shows up immediately, and only the first data request has to wait.

3.2 Presentation of the Application

In this section, we walk through the platform as the different users actually see it. The interfaces are grouped by role: the public pages that any visitor can reach, the client pages, the seller pages, and finally the admin dashboard. For each one, we show a screenshot and explain what the user can do there.

3.2.1 Public Pages

These are the pages available to anyone, with or without an account.

Home page

The landing page introduces the platform with a split-screen hero section, a search bar, and a list of craft categories. From here, a visitor can jump straight into browsing or follow a category into the catalog

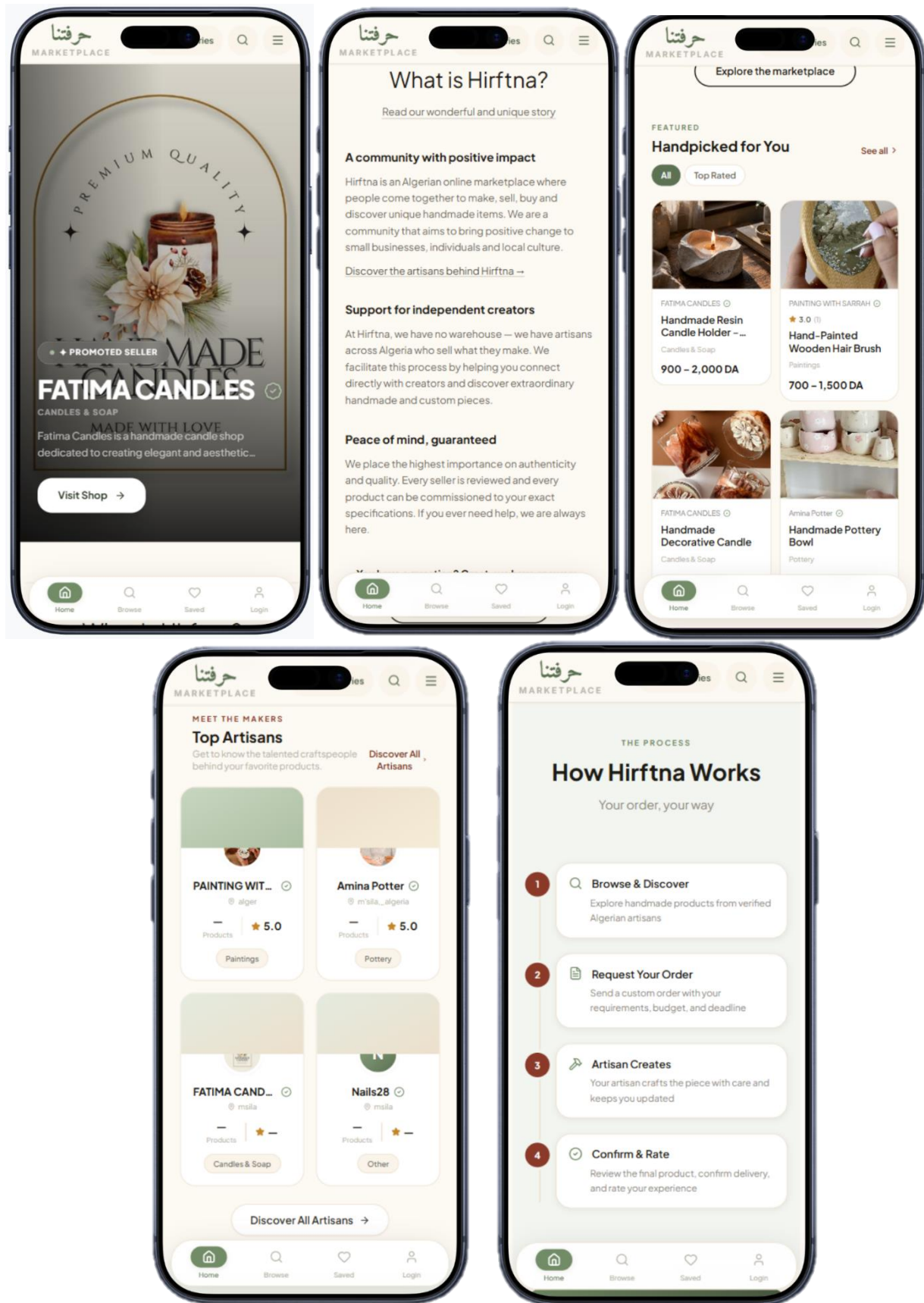


Figure 3.2: Home page

Product catalog

This page lists all active products from sellers, with filtering by category and price, and full-text search over titles and descriptions. Each card shows the price range, the estimated completion time, and the seller's rating.

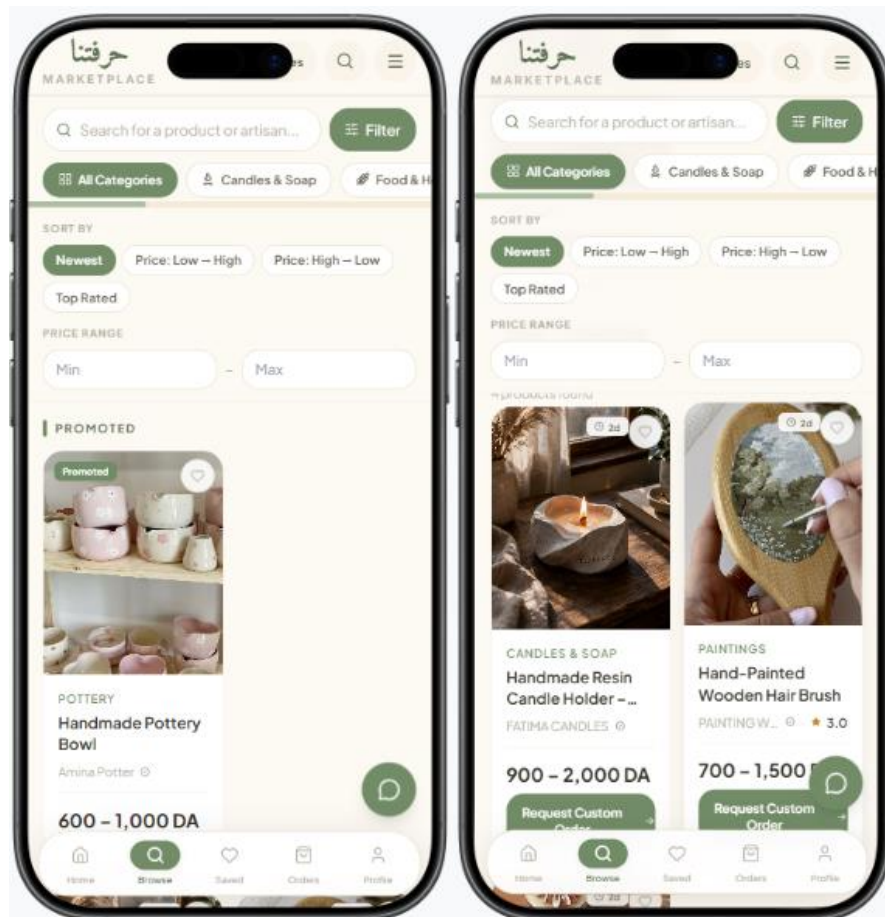


Figure 3.3: Product catalog page.

Product details

This page shows everything about a single product. the image gallery, the price range, the completion time, the seller's profile link, and the reviews left by previous clients. The main action here is the "Request Custom Order" button.

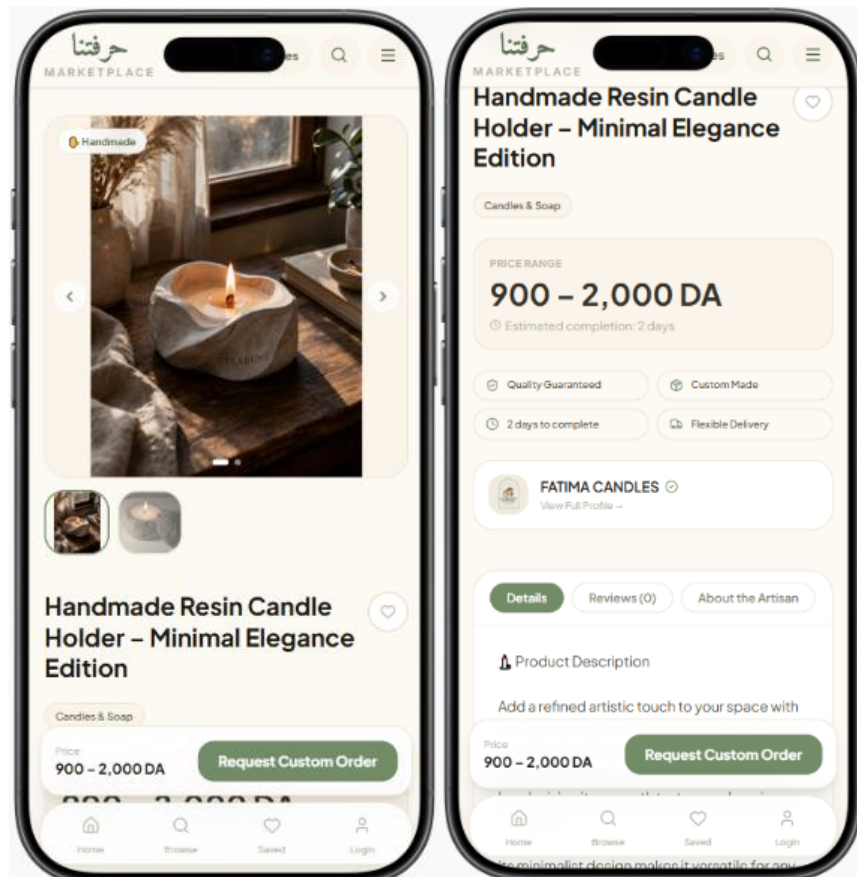


Figure 3.4: Product details page

Seller profile

Each seller has a public profile showing their shop name, their artisan story, their verification badge, their average rating, and the products they offer.

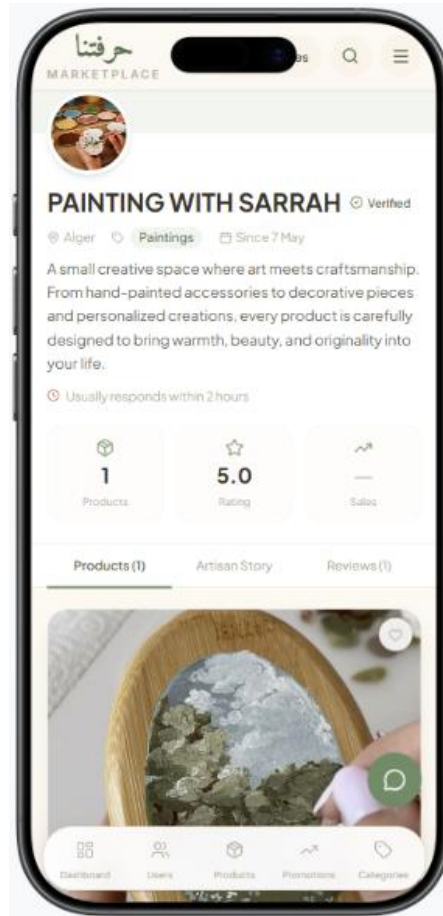


Figure 3.5: Seller profile page

3.2.2 Client Pages

These pages are available once a user signs in with a client account.

Custom order form

The three-step form where a client turns a product into an actual order: first, the requirements (budget range, deadline, notes, reference images), then the delivery and payment options, and finally their contact details.

The figure displays three sequential screenshots of a mobile application interface for a 'Custom Order Request' form, specifically for a 'Handmade Resin Candle Holder'.

Screen 1 (Details): The first screen shows the 'Details' step (1 of 3). It includes a progress bar at the top. The main heading is 'Tell the artisan what you want' with a subtext 'Be as specific as possible — colors, size, material'. Below this is a text input field with a placeholder example: 'e.g. I'd like a hand-painted ceramic vase, 30cm tall, in turquoise with Berber geometric patterns...'. There is a character count '0/500'. Below the text field is a 'Budget Range' section with 'Optional' and two input fields labeled 'DA Min' and 'DA Max'. Below that is a 'Deadline' section with 'optional — must be a future date' and a date input field 'mm/dd/yyyy'. At the bottom is a 'Reference Images' section with 'optional' and a small image upload icon. A green 'Next >' button is at the bottom.

Screen 2 (Delivery): The second screen shows the 'Delivery' step (2 of 3). The heading is 'How do you want to receive it?' with subtext 'Choose your delivery and payment preferences'. It features two sections: 'Delivery Method' and 'Payment Method'. Each section has three radio button options. Under 'Delivery Method': 'Hand to Hand' (Direct delivery from artisan), 'Office Pickup' (Pick up at our partner office), and 'Fast Delivery' (Express shipping to your door). Under 'Payment Method': 'Cash on Delivery' (Pay when you receive) and 'Card Payment' (Pay online securely). A green 'Next >' button is at the bottom.

Screen 3 (Contact): The third screen shows the 'Contact' step (3 of 3). The heading is 'Your contact details' with subtext 'The artisan needs this to reach you'. It includes three input fields: 'Full Name' (with value 'Norhane MESSARA'), 'Phone Number' (with value '+213555123454'), and 'Delivery Address' (with value 'M'sila 1200 LOGEMENTS'). A green 'Send Order Request' button is at the bottom.

Figure 3.6: Custom order request form

My orders

This page lists the client's orders grouped by status: pending, accepted, ready, completed, rejected, so they can follow each one through its lifecycle, confirm completion when the item is ready, and rate the seller afterward.

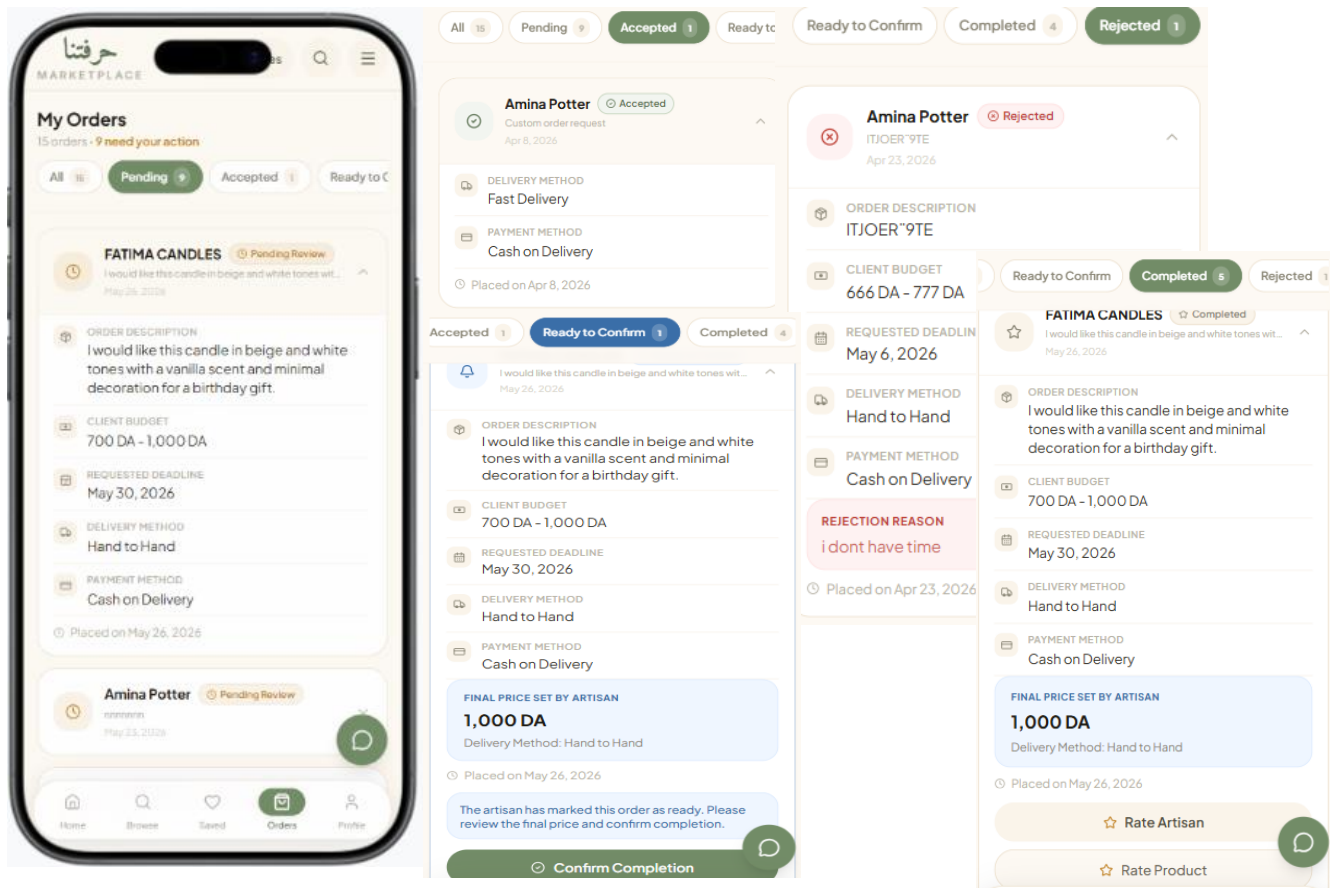


Figure 3.7: Client order management page

Wishlist

The products a client has saved to come back to later.

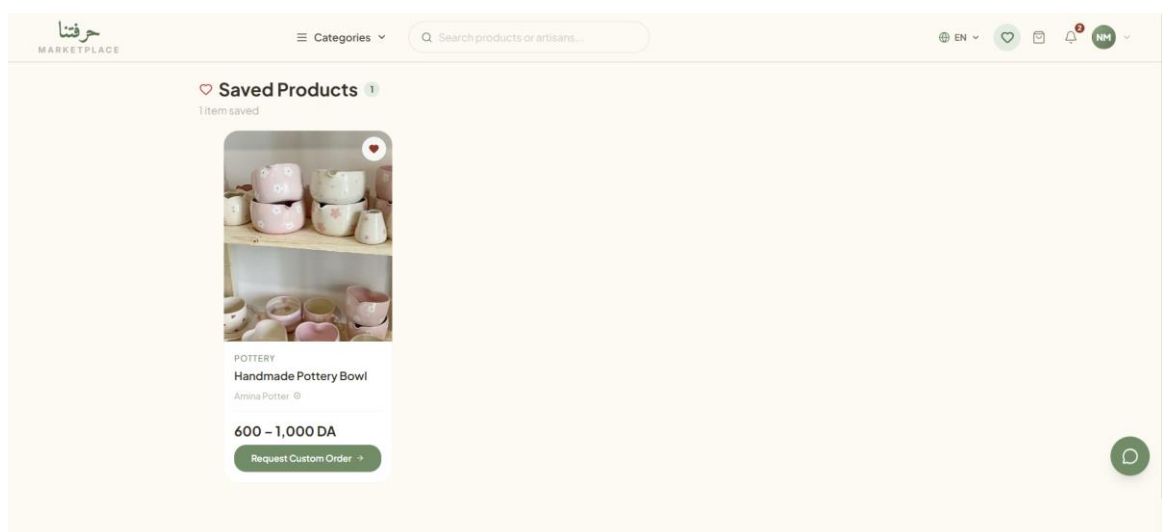


Figure 3.8: Wishlist page

3.2.3 Seller Pages

These pages are available to users with a seller account.

Seller dashboard

The seller's home base shows their incoming orders, basic analytics, and their verification status. From here, they manage everything related to their shop.

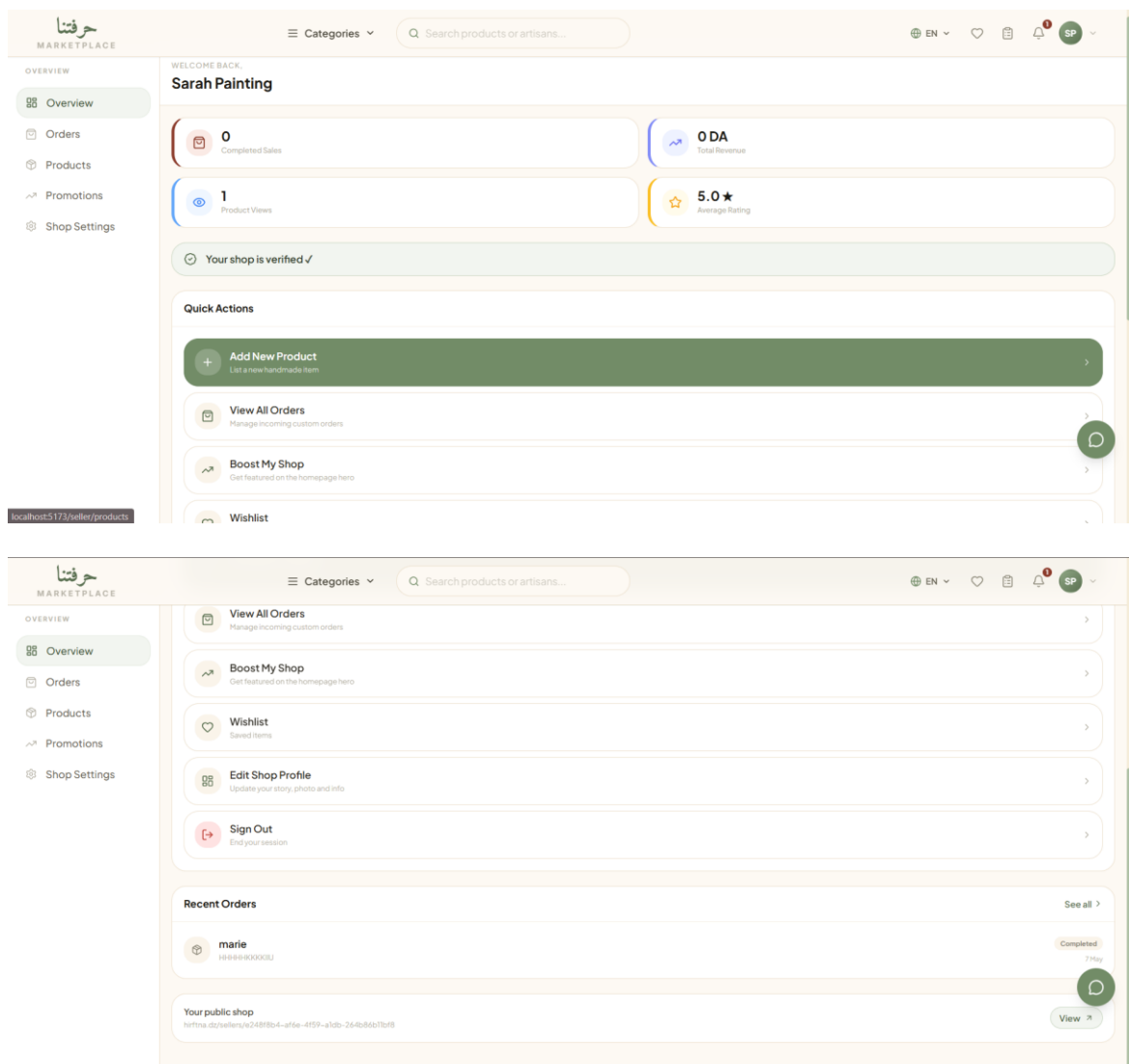


Figure 3.9: Seller dashboard

Product management

Where sellers create, edit, and delete their product listings, upload images, and toggle a product's visibility.

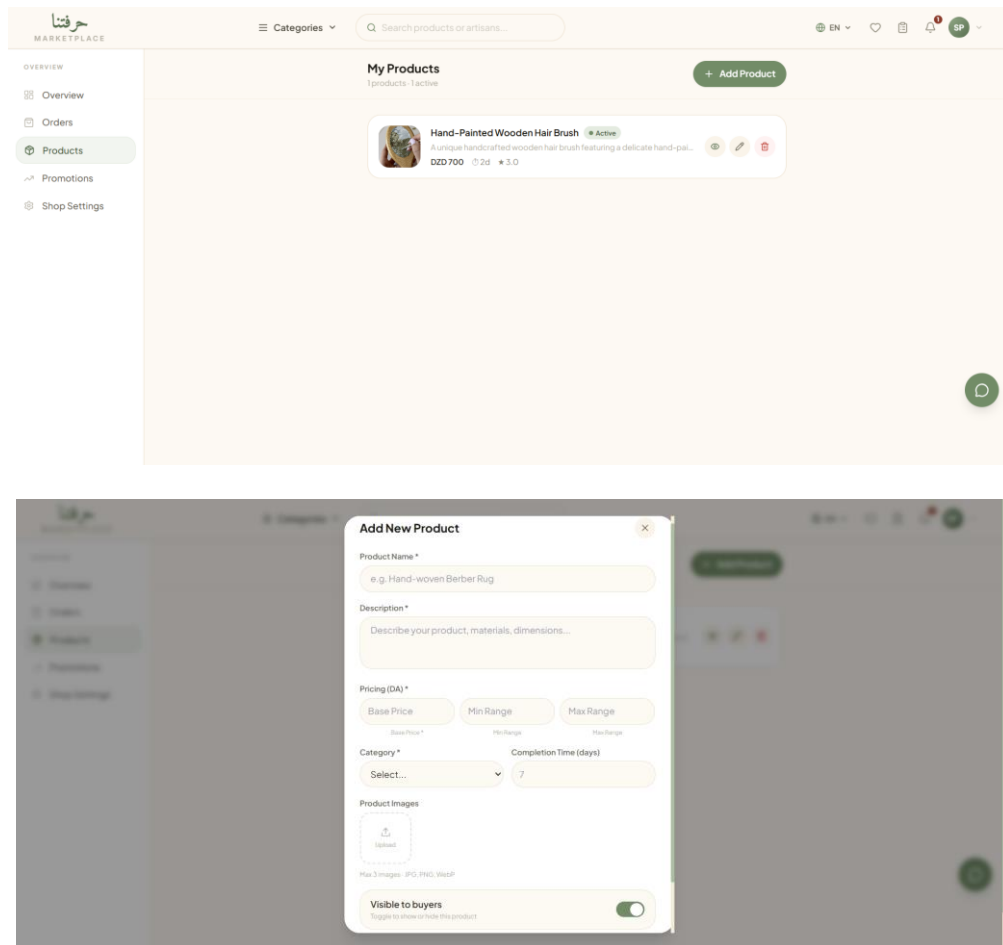


Figure 3.10: Product management page

Order handling

This is where a seller responds to incoming orders: accept or reject them (with a reason), mark an accepted order as ready while setting the final price, and rate the client once the order is complete.

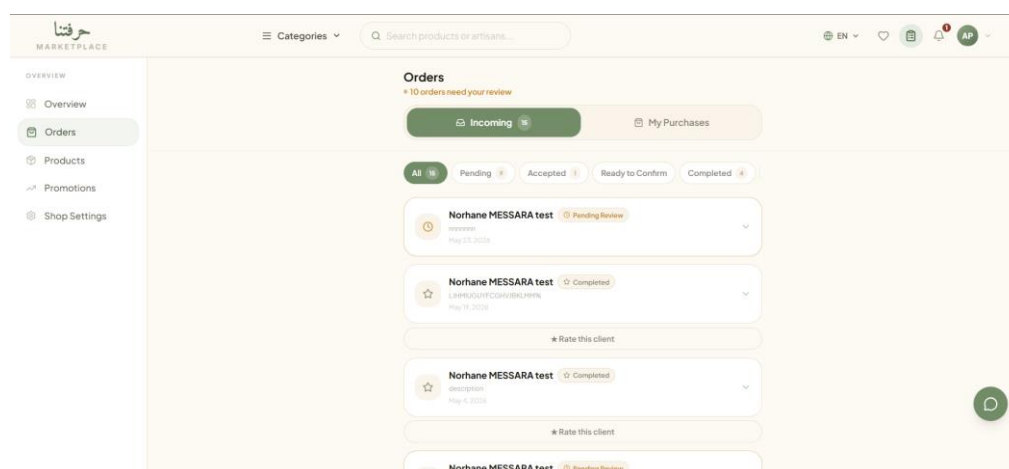


Figure 3.11: Seller order handling page

Promotion request page

This is where a seller asks to have one of their products featured. They pick a product, choose where it should appear (the homepage hero section or the featured grid on the browse page), and for how long (7, 14, or 30 days), then send the request to the admin for review. The page also shows what happened to their earlier requests, whether each one is still pending, approved, or rejected.

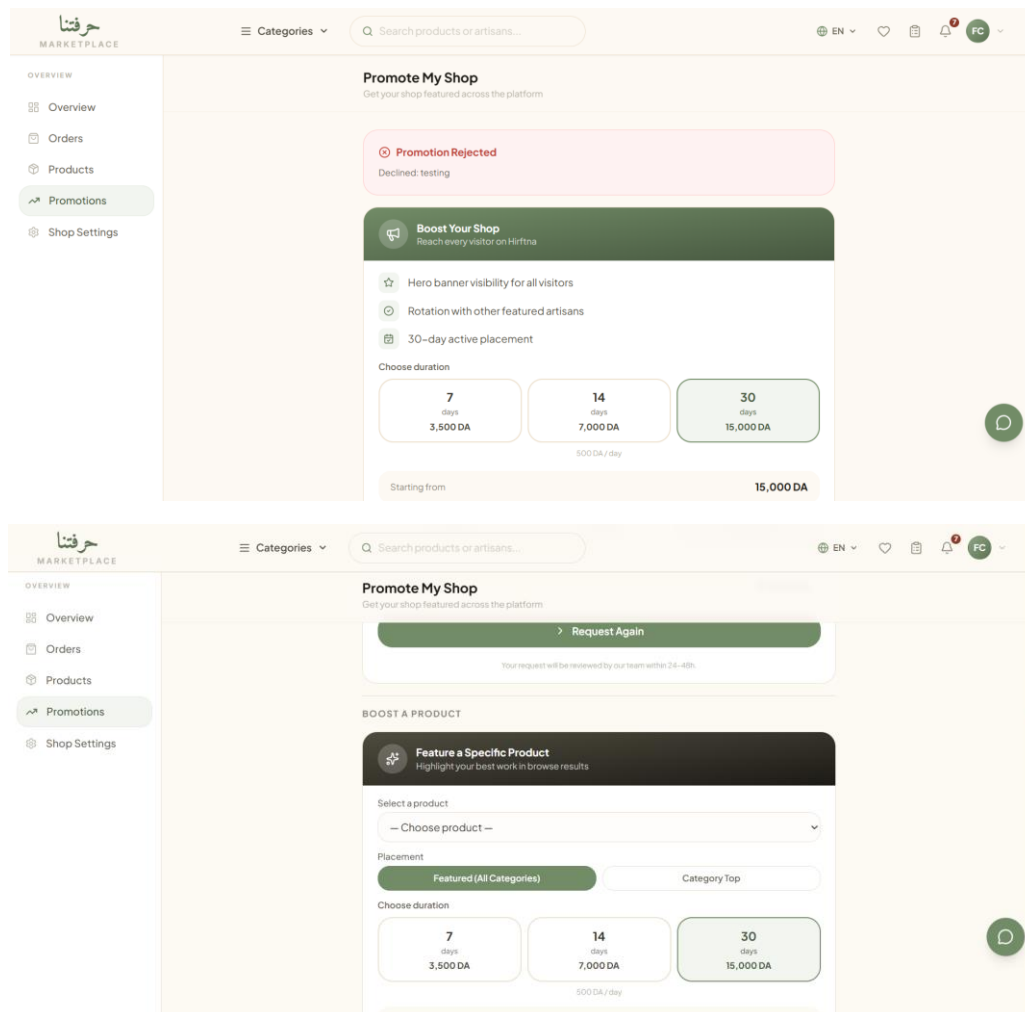


Figure 3.12: Promotion requests interface.

3.2.4 Admin Dashboard

The admin dashboard is a separate area for managing the platform. It is only reachable by an account whose role is set to admin in the database.

Overview

The landing page of the dashboard, with summary statistics: total users, products, and orders, plus category breakdowns and the most active sellers.

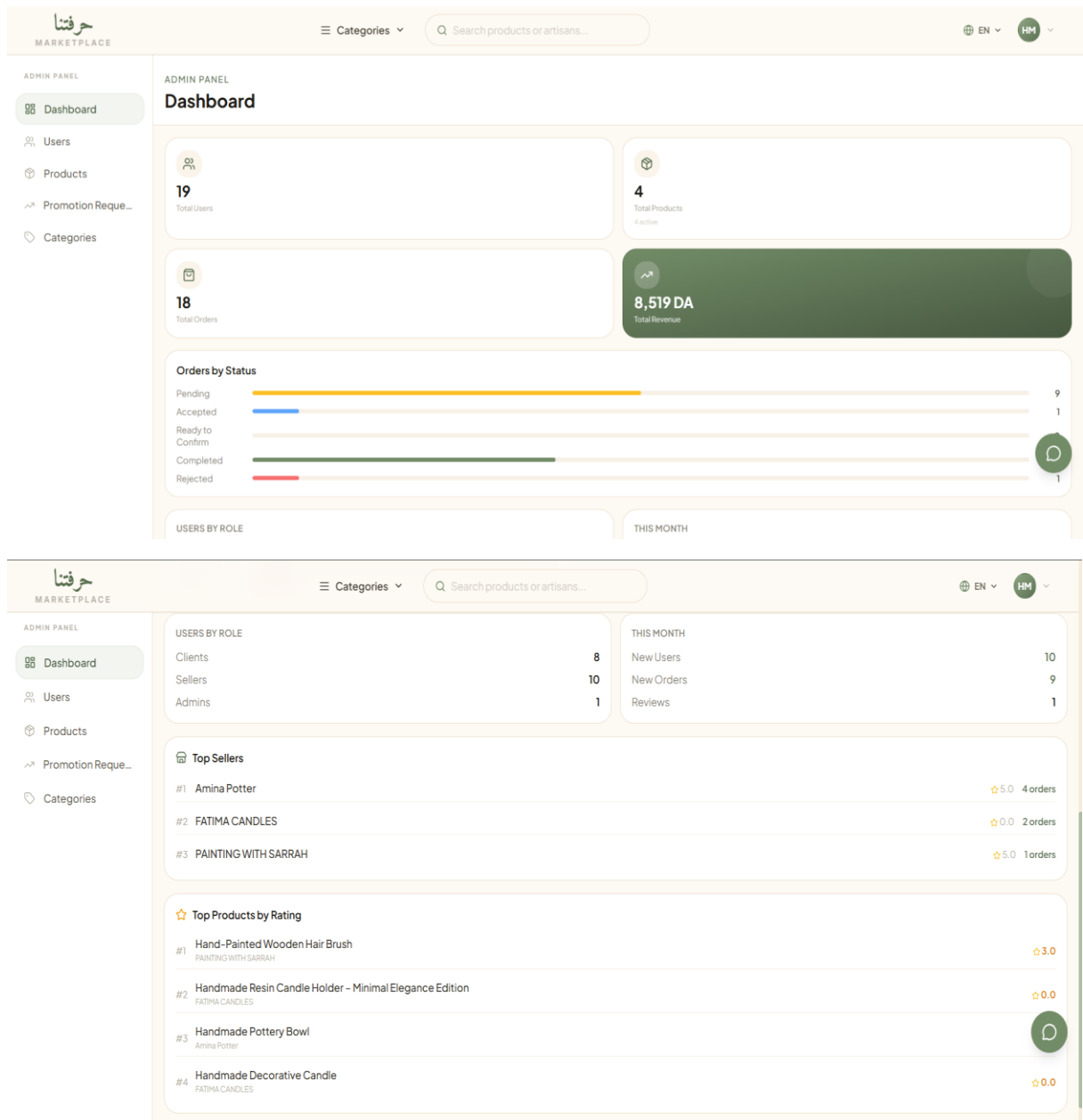


Figure 3.13: Admin dashboard overview

User and seller management

Where the admin views all users, verifies or revokes sellers, and adjusts roles. Seller verification is one of the key actions here; it awards the trust badge that appears on the seller's profile and product cards, giving buyers a quick way to identify established artisans.

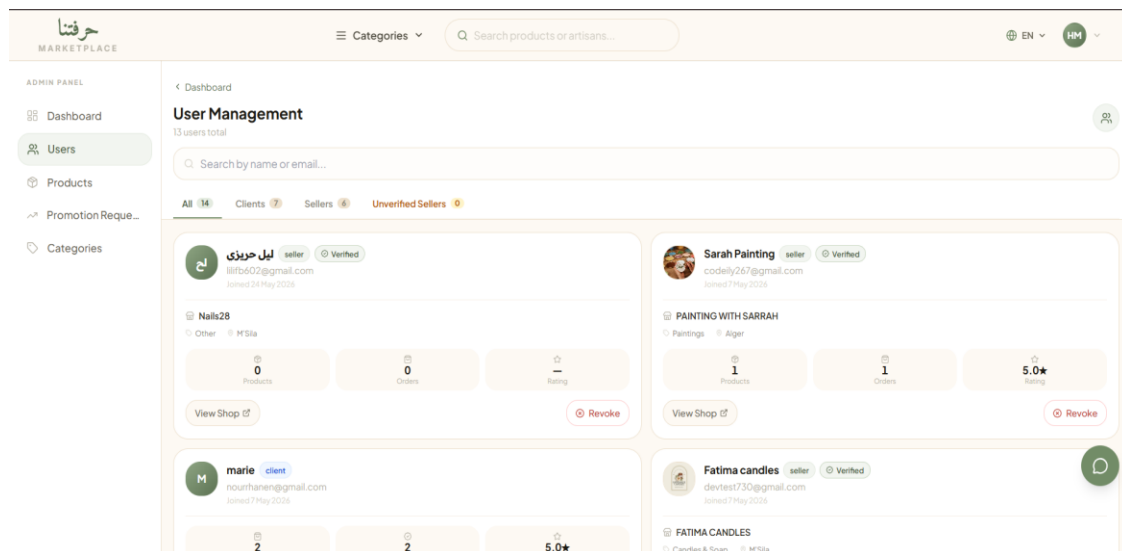


Figure 3.14: User management interface

Category and product moderation

The admin manages the list of craft categories (create, edit, and delete) and can force-delete any product listing on the platform.

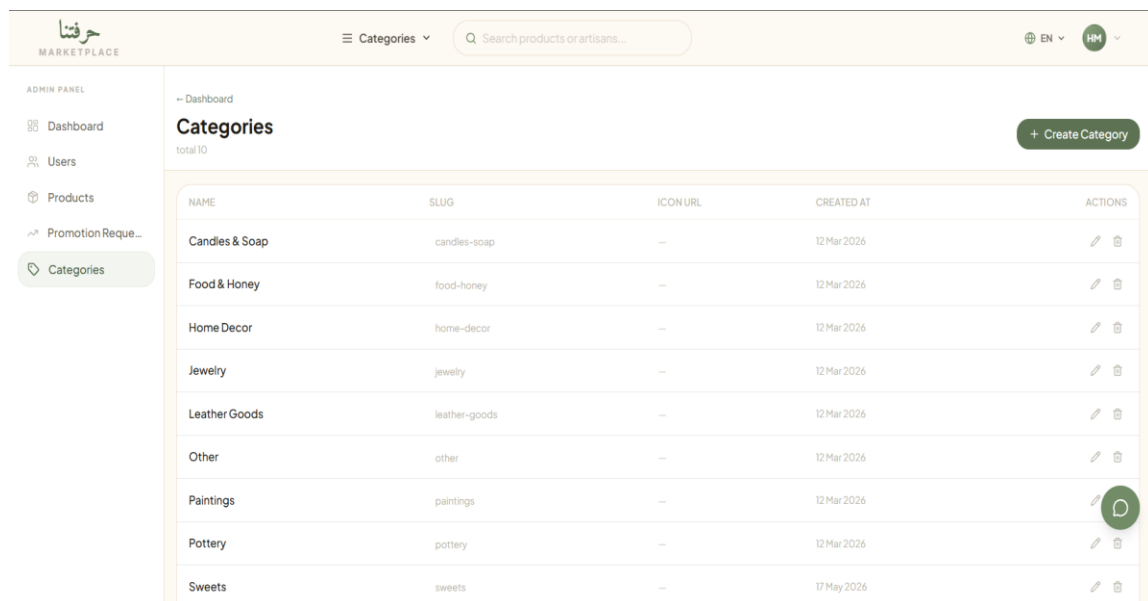


Figure 3.15: Category management interface

Promotion management

Sellers can request to have a product promoted in the hero section or higher in the browse results. This page is where the admin reviews those requests and approves or rejects them.

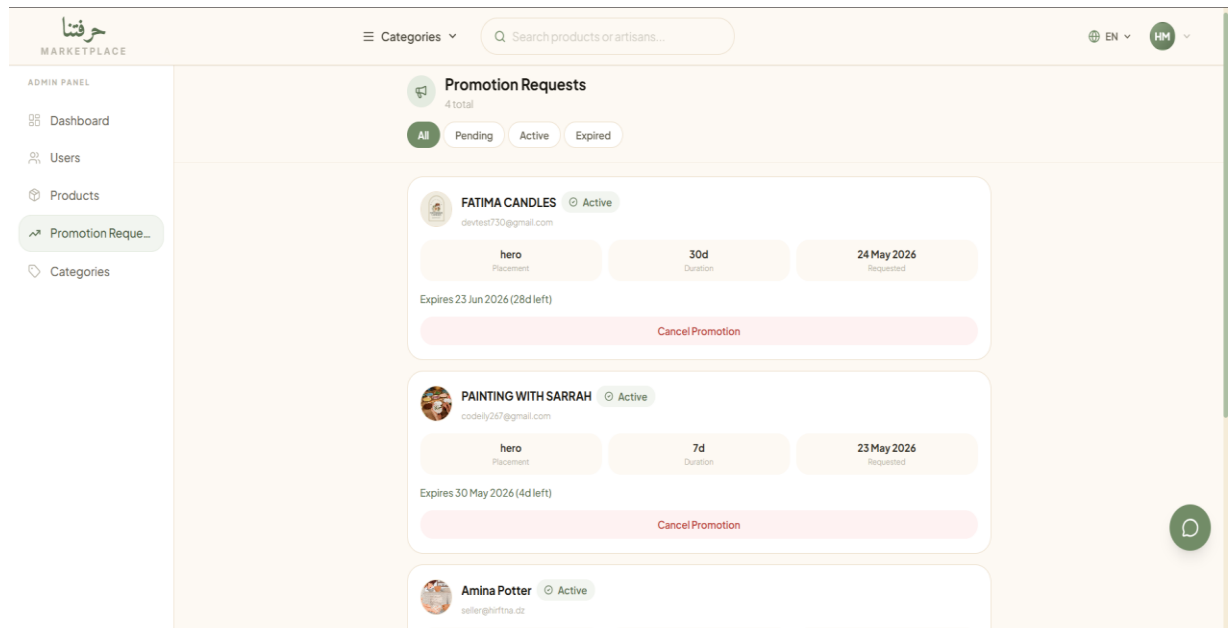


Figure 3.16: Promotion management interface.

3.2.5 AI chatbot

Hirftna assistant powered by Google Gemini, it answers questions about how the platform works and helps users find products in Arabic and English. This feature is only available to registered users. Figure 3.17 shows a sample conversation.

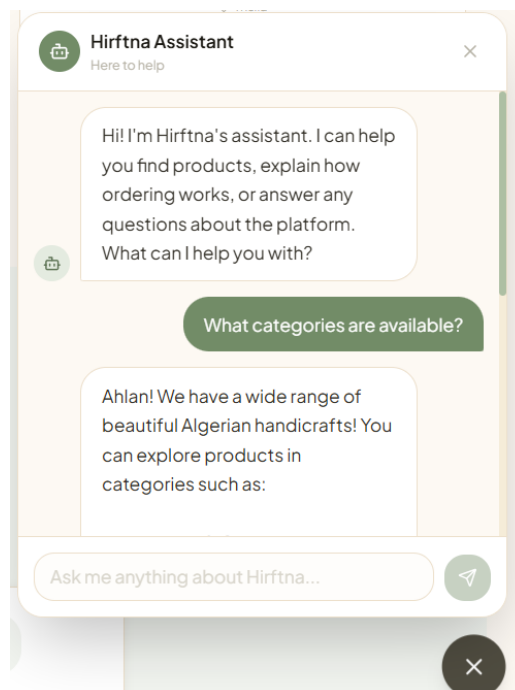


Figure 3.17: Chatbot sample conversation.

Conclusion

This chapter showed how the design from the previous chapters became a working platform. We went through the development environment, the backend and frontend stacks, and the external services we leaned on. Supabase for the database, auth, and storage, Gemini for the chatbot, and Chargily for payments. We also explained our deployment setup, where the frontend runs on Vercel, the backend on Render, and the data layer on Supabase, all redeploying automatically from GitHub. Finally, we presented the main interfaces of the application across the four user roles. With the system built and deployed, the next and final chapter steps back to look at what we achieved, the limitations we are aware of, and the directions we would take the project if we continued working on it.

Chapter 4

Economic Study and Startup Project

4.1. Introduction

The first three chapters of this report described how we designed and built Hirftna as a working software product. This chapter takes the project one step further: we look at Hirftna as a business that has to operate in the real Algerian market, and we test whether the model holds up over a five-year horizon.

The chapter is organised into two parts. The first part is an in-depth market study. We start with the PESTEL (Political, Economic, Social, Technological, Environmental, Legal) context, size the addressable market using TAM (Total Addressable Market)-SAM (Serviceable Addressable Market)-SOM (Serviceable Obtainable Market), build customer personas, map the competitive landscape, run a SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis, share what we learned from a small field study with Algerian artisans, and finish with the go-to-market strategy. The second part is the business model and a detailed financial study. We present the Business Model Canvas, the production and organisation plan, the five-year financial projections (income statement, treasury, balance sheet, profitability indicators), the risk matrix, the legal framework, and the 18-month roadmap.

We built everything around the framework of Ministerial Decree 1275 of 27 September 2022, which created the "One Diploma, One Startup" mechanism for student-led projects in Algeria. This means we treat the Decree's advantages, four-year tax holiday, ANADE (Agence Nationale d'Appui et de Développement de l'Entrepreneuriat) financing, and two-step labeling via startup.dz as real components of the business model, not vague benefits. The four official appendices expected by the CNCSIU (Comité National de Coordination Sectoriel pour l'Innovation et l'Incubation Universitaire) evaluation committee (balance sheet, projected income statement, treasury plan, and Business Model Canvas) appear at the end of the chapter. [43] [44]

PART 1: IN-DEPTH MARKET STUDY

4.2. Introduction and Methodology

Our analysis combines secondary sources (World Bank country data for Algeria, Industry reports on Algerian e-commerce, the ASJP (Algerian Scientific Journals Platform) scientific platform for the Decree 1275 legal framework) with primary qualitative research: twelve semi-structured interviews with Algerian artisans (eight from M'sila and Biskra, four from Algiers) plus informal feedback from potential buyers in our networks. [45] [46] [47] [48] [44]

4.3. Project Presentation and Innovative Aspects

4.3.1. Startup Identity

Commercial name: Hirftna Marketplace (Arabic: حرفتنا). The visual identity uses the Arabic word حرفتنا above the word "MARKETPLACE" in a clean typography with sage green accents (see logo in Figure 4.1). The mission is to give Algerian artisans a professional digital channel that fits how they actually work, and to give buyers a trusted way to discover handcrafted Algerian products. The vision over the next five years is to become the reference platform for Algerian artisanal commerce, both inside the country and for the diaspora.

Activity field: digital marketplace platform (services, e-commerce). Development stage: deployed MVP, accessible at hirftna.vercel.app with backend on Render and database on Supabase. The project is currently led by MESSARA Norhane as the sole founder, who designed and developed the MVP independently. The operational team structure planned at incorporation includes the founder taking the CEO role, a CTO to be recruited for platform engineering and scaling, and a Marketing and Communications Lead to be brought in for user acquisition. The financial projections in Part 2 reflect this full three-person team from Year 1, in line with the standard practice of presenting the intended operational team for the projection horizon. The team operates from the University of M'sila incubator under the Decree 1275 framework.



Figure 4.1: Hirftna marketplace logo.

4.3.2. Innovative Aspects

Hirftna's innovation rests on the Custom Order System described in Chapter 1, which replaces unstructured Instagram DM (Direct Message) negotiation with a formal flow where every step (submission, acceptance, readiness, completion, and rating) is tracked. Layered on top of that is the mutual rating system, where the buyer rates the seller and the seller rates the buyer, which addresses a problem unique to the Algerian market: the lack of protection for artisans against unreliable clients. The third pillar is the hybrid commerce model with COD (Cash on Delivery) and hand-to-hand delivery, which reflects the reality of Algerian transactions rather than imposing a Western e-commerce template that would not work here

4.4. PESTEL Analysis

PESTEL gives us a structured view of the external environment in which Hirftna will operate. We go through each dimension with figures sourced from official Algerian and international references.

Dimension	Key factors in the Algerian context	Impact	Intensity
Political	Strong government pushes for entrepreneurship through Decree 1275 [43] [44] and ANADE financing. Push to digitize the artisan sector via electronic registration under Decree 96-01 [49].	Positive	High
Economic	GDP growth reached 3.8% in 2025, with the World Bank projecting 3.5% in 2026 and 3.3% in 2027 [45]. Inflation eased significantly from 4.0% in 2024 to 1.7% during the first nine months of 2025, supported by stable food prices and a stable exchange rate [45]. Unemployment is near 11-12%. The Algerian e-commerce market reached approximately 1.7 billion USD in 2025 with 15-	Positive	High

Dimension	Key factors in the Algerian context	Impact	Intensity
	20% annual growth [47], with industry estimates valuing the sector at 1.9 billion USD and nearly 200,000 active online merchants [48]. UNCTAD's August 2025 eTrade Readiness Assessment reports that 13 million online transactions were processed in 2024, and that online businesses grew at an average annual rate of 92% between 2020 and 2024 [50].		
Sociocultural	Population at 48.0 million in 2026 (47.4 million in 2025), with a median age of 29.8 years and 29.9% under 15 years old. Strong cultural attachment to artisanal products. Growing acceptance of online shopping. Algerian diaspora of around 7 million abroad. [51]	Positive	High
Technological	Internet penetration at 76.9% in 2025 (36.2 million users) [52]. Mobile penetration is at 116%, with 91.4% on broadband (3G/4G/5G) [52] Chargily payment gateway has been operational since 2021; UNCTAD reports that electronic payments tripled between 2021 and 2024. [50]	Positive	High
Environmental	Growing interest in sustainable consumption among urban youth. Handicrafts naturally fit the slow-commerce narrative. No strong regulatory pressure yet on packaging or carbon footprint for small operators.	Positive	Medium
Legal	Law 18-05 on e-commerce [8] and Law 18-07 on personal data protection [53] have been in force since 2018. SAS legal form was introduced in 2022 (Loi 22-09) specifically for startups. Some regulatory ambiguity around COD remains.	Mixed	Medium

Table 4.1: PESTEL analysis of the Algerian market for Hirftna.

The overall picture is favorable. The macro context supports digital commerce, the government actively backs young founders through Decree 1275 and ANADE, and the demographic profile (young, mobile-first, increasingly comfortable online) matches what a marketplace needs. The two areas that need monitoring are the legal framework around cash transactions and the maturity of the local payment infrastructure.

4.5. Target Market Analysis

4.5.1. Market Sizing (TAM / SAM / SOM)

We size the addressable market in three layers, following the standard framework. The numbers below are conservative and grounded in published figures from the Algerian Ministry of Tourism and Handicrafts and from open e-commerce sources.

Level	Definition	Size	Source
TAM	Total Algerian handicraft economy across all channels.	≈ 450–460 billion DZD (Algerian Dinar)	World Bank Algeria Economic Update 2025 [45]
SAM	Digitally addressable handicraft demand	≈ 110 – 120 billion DZD	25% of TAM, consistent with internet penetration [52] and e-commerce growth [47]
SOM (Year 5)	Realistic market share for Hirftna	≈ 1.5–2 billion DZD GMV (Gross Merchandise Volume)/year	Startup penetration benchmark (1–2% SaaS (Software as a Service) marketplaces)

Table 4.2: TAM, SAM, and SOM for Hirftna over the 5-year horizon.

The relationship between SOM (GMV captured on the platform) and our actual revenue matters. Hirftna does not take a commission per sale, so revenue depends on paid promotions and tiered subscriptions. With 3,000 sellers in Year 5, our model projects 43.5 million DZD in revenue, representing around 2–3% of the total GMV that flows through the platform. That ratio is what the model is really sizing.

We acknowledge that the SOM estimate of 1.5–2% of SAM is itself uncertain. The Algerian e-commerce market is still in its early stages, making traditional adoption curves less predictive in emerging digital environments. Therefore, a cautious, penetration-based approach has been adopted.

4.5.2. Market Growth

Two trends matter for us. The first is the broader Algerian e-commerce trajectory: the Algerian e-commerce market reached approximately 1.7–1.9 billion USD in 2025 with 15–20% annual growth, and 200,000 active online merchants. UNCTAD's eTrade

Readiness Assessment (August 2025) confirms an average annual growth of 92% in online businesses between 2020 and 2024, with 13 million online transactions processed in 2024. B2C (Business-to-Consumer) e-commerce accounted for 0.8% of GDP in 2023, indicating significant room for growth. The second trend is the digitization of the supply side: the Ministry of Tourism and Handicrafts has been pushing electronic registration of artisans for several years, steadily growing the pool of formally registered sellers. Both trends move in our favor and slowly enough that a focused player has time to establish a position. [47] [48] [50] [49]

4.5.3. Customer Segments and Personas

Hirftna is a two-sided marketplace, so we built personas on both sides. Two personas describe the supply side, and one describes the demand side. The diaspora segment is identified as a secondary target activated from Year 3

Persona 1: Amel, the home-based artisan

Amel is 32, lives in Tlemcen, is home-based, and specialises in chedda-style embroidery. Monthly income from artisan work ranges from 35,000 to 50,000 DZD. Currently sells through Instagram DMs with around 500 followers. Main pain: clients ghosting her after she has started a piece. Willing to pay 1,000-2,500 DZD/month once the platform proves it brings consistent buyers.

Persona 2: Karim, the workshop owner

Karim is 45 and runs a small leather workshop in Ghardaïa with two apprentices. Monthly revenue 200,000-400,000 DZD. Sells through a physical shop and regional fairs. Main pain: cannot reach customers outside his region without closing the workshop. Willing to pay 2,000-5,000 DZD/month for a steady flow of mid-priced orders.

Persona 3: Sofiane, the urban buyer

Sofiane is 28, lives in Algiers, engineer earning 90,000-120,000 DZD/month. Buys handicrafts three to four times per year (mostly gifts), typical purchase 3,000-15,000 DZD. Prefers Cash on Delivery for first purchases, then CIB for trusted sellers. Free for him to use Hirftna; pays the artisans directly for products.

Persona	Side	Age / Location	Monthly income (DZD)	Willingness to pay
Amel: home artisan	Supply	32 / Tlemcen	35,000–50,000	1,000–2,500 DZD/month
Karim: workshop owner	Supply	45 / Ghardaïa	200,000–400,000	2,000–5,000 DZD/month
Sofiane: urban buyer	Demand	28 / Algiers	90,000–120,000	Free for the buyer; pays artisans directly

Table 4.3: Persona summary.

4.6. Competitive Analysis

4.6.1. Competitive Landscape

We grouped the players we compete against into three categories. The space is fragmented, and there is no clear leader, which is part of why we believe Hirftna can establish a defensible position.

Category	Players	Why we compete
Direct competitors	Warcha, AMADHAL Store	Warcha is a discovery-oriented platform that showcases Algerian creators. AMADHAL Store operates more as a curated boutique. Neither offers a structured custom-order workflow nor a mutual rating system. [54]
Indirect competitors	Ouedkniss, Jumia Algeria, Selltic	General-purpose marketplaces. Craft items get buried under unrelated listings. Communication remains informal and phone-based.
Substitutes	Facebook, Instagram, physical souqs, craft fairs	What most artisans currently use. Free and familiar, which makes them the hardest to displace. Their weakness is the same as their strength: unstructured. [11]

Table 4.4: Competitive landscape map.

Local artisan-focused platforms in Algeria are still emerging. Most are discovery- or boutique-oriented rather than full transaction marketplaces. This is part of why Hirftna's structured order workflow and mutual rating system are real differentiators. Looking at this map, we noticed that the players we compete most directly against are

not the platforms, but the habits. Instagram DMs and Facebook Marketplace are what artisans currently use, and changing a habit is harder than displacing a competitor.

4.6.2. Detailed Comparison

Criterion	Hirftna	Warcha	Ouedkniss	Facebook / Instagram
Pricing model	Promotions + freemium subscriptions, no commission	Listing-based	Free listings + paid boosts	Free
Custom order workflow	Yes — structured negotiation, budget range, final price	No	No	Informal via DM
Mutual rating system	Yes — both sides rate each other	No	No	No
Online payment (Chargily)	Yes — CIB and Edahabia	Partial	No	No
COD + hand-to-hand	Yalidine, ZR Express, and Maystro integrated	Limited	Phone-based, cash	Cash-based
BaridiMob manual payment	Yes (subscription verification)	No	No	No
Arabic-first (RTL)	Yes — full RTL	Partial	Bilingual	Bilingual

Table 4.5: Detailed comparison against principal competitors.

4.6.3. Porter's Five Forces

Porter's framework gives us a structural read of how attractive the Algerian artisan marketplace sector actually is.

Force	Assessment	Reasoning
Rivalry among existing competitors	Low–Medium	Few direct competitors. Substitutes (Facebook, Instagram) are unstructured. The space is underserved rather than saturated.
Threat of new entrants	Medium	Technical barriers are modest, but building trust and seller critical mass takes a minimum of 18-24 months.

Force	Assessment	Reasoning
Bargaining power of suppliers	Low	Artisans are many, dispersed, and not organized. No single seller can dictate terms.
Bargaining power of customers	Medium	Buyers have free alternatives. They switch if trust and selection are not delivered.
Threat of substitutes	High	Facebook and Instagram are free, familiar, and entrenched. This represents the principal structural challenge for the project.

Table 4.6: Porter's Five Forces. Overall sector attractiveness: Medium.

4.6.4. Unique Selling Proposition

Hirftna's USP (Unique Selling Proposition) combines the structured custom-order workflow (which replaces unstructured DM negotiation with a formal flow) and the mutual rating system (which protects artisans from unreliable clients). On top of these, the platform supports both Chargily online payment and Yalidine COD with hand-to-hand delivery, matching how Algerian commerce actually works. Each of these would take a competitor real-time and engineering to replicate, which is what makes them defensible.

4.7. SWOT Analysis

The SWOT below synthesises what came out of the PESTEL, the competitive analysis, and our internal assessment.

STRENGTHS	WEAKNESSES
MVP is already deployed and accessible online. The mutual rating system is unique in the Algerian market. Structured custom-order workflow purpose-built for handmade. Arabic-first interface with full RTL support. Founder with full-stack engineering capability; founding team of three to be completed at incorporation across engineering and marketing.	New brand with no commercial track record. Loss-making Year 1 due to the no-commission revenue model. Dependence on international cloud providers. A three-person team cannot cover every function. Limited marketing budget against entrenched substitutes. Chicken-and-egg problem at launch.

OPPORTUNITIES	THREATS
Algerian e-commerce reached 1.7-1.9 B USD in 2025 with 15-20% annual growth and 200,000 active online merchants [47] [48]. Decree 1275 institutional support (4-year tax holiday, ANADE financing). The diaspora segment (~7 million abroad) is largely untapped. The government pushes to digitize the artisan sector [49]. High mobile and broadband penetration [52]. Cultural pride in Algerian handicrafts; recent UNESCO recognitions.	Facebook and Instagram are free and entrenched substitutes. Cash culture: ~75% of transactions outside traceable rails. Regulatory uncertainty around e-commerce and digital tax compliance. Currency risk on international cloud payments. Inflation eased to 1.7% in the first 9 months of 2025 from 4.0% in 2024, reducing immediate pressure on purchasing power but warranting monitoring [45]. Possible new entrants once the model is validated.

Table 4.7: SWOT matrix grounded in the preceding analyses.

4.8. Primary Research — Field Interviews

We ran twelve semi-structured interviews with Algerian artisans during the project. The sample is small and qualitative on purpose: at this stage, we wanted depth rather than statistical generalisation. A larger quantitative survey is planned for the post-launch phase, once we can sample from real users of the platform.

4.8.1. Methodology

We interviewed twelve artisans: eight from M'sila and Biskra and four from Algiers. Eleven were women, and one was a man. We used a semi-structured guide covering five themes: current sales channels, recurring pain points, attitudes towards online commerce, willingness to pay for tooling, and reactions to the Hirftna concept (presented as a mockup of the deployed platform). The full interview guide is in Appendix F.

4.8.2. Key Findings

The interviews confirmed three of our design assumptions and challenged one.

Finding	Frequency	Implication for Hirftna
Currently sells via Instagram or Facebook DMs	11 of 12 artisans	Confirms our main substitute hypothesis. We need to give them a reason to switch.

Finding	Frequency	Implication for Hirftna
Has been ghosted by a client in the last 30 days	9 of 12 artisans	Validates the mutual rating system. This is the single most cited pain point.
Willing to pay 800-2,500 DZD/month for a platform that brings real orders	7 of 12 artisans	Supports the freemium subscription pricing planned for Year 2.
Considers Cash on Delivery essential, online payment secondary	12 of 12 artisans	Confirms our decision to support both COD and online payment.
Trusts online payment for the first transaction with a new buyer	4 of 12 artisans	Challenges to commission-based models. Supports our no-commission decision.

Table 4.8: Key findings from the qualitative interviews (n=12 artisans).

Two design decisions came directly from these findings: keeping the mutual rating system central in the platform UX (User Experience), and removing per-sale commission from the revenue model in favour of promotions and subscriptions paid upfront. Conducting these interviews changed how I designed the rating system. Before the interviews, I assumed buyers would be the most demanding side; the interviews showed me that artisans need protection more than buyers do.

4.9. Go-to-Market Strategy

4.9.1. Pricing Strategy

We chose a freemium model with two paid layers. The free tier lets any artisan sign up and list up to five products with no monthly fee. The paid layers are paid promotions (active from Year 1, available to any seller, with prices of 3,500 / 7,000 / 15,000 DZD for 7 / 14 / 30 days on either Featured Product or Shop Boost placements) and tiered subscriptions introduced from Year 2 (Standard at 800 DZD/month and Pro at 2,000 DZD/month). We deliberately do not take a commission per sale. The detailed rationale is in section 4.14.3.

4.9.2. Distribution Channels and Acquisition

Hirftna is direct-to-user. There are no resellers between the platform and either side of the marketplace. Acquisition runs across four channels: organic social media (Instagram and Facebook pages featuring real artisans and their work), influencer

partnerships in the Algerian artisanat space, direct outreach to artisan cooperatives in M'sila and Biskra during the first six months, and presence at artisanal events such as the annual Salon de l'Artisanat in Algiers.

4.9.3. Key Performance Indicators

We track three primary KPIs. Customer Acquisition Cost (CAC) is projected at approximately 1,600 DZD per seller in Year 1 (240,000 DZD marketing budget for 150 sellers). Customer Lifetime Value (LTV) is estimated at 22,000 DZD per seller over a two-year retention. The LTV-to-CAC ratio sits around 14:1, which is healthy.

4.9.4. Growth Objectives

Horizon	Target	Revenue target
Short term (0–12 months)	150 active sellers, MVP fully validated, first paid promotions sold	855,000 DZD
Medium term (1–3 years)	900 active sellers, freemium subscriptions launched, presence across major wilayas	10.5 million DZD
Long term (3–5 years)	3,000 active sellers, diaspora segment activated, preparation for Maghreb expansion	43.5 million DZD

Table 4.9: Growth objectives over the three horizons.

4.10. Conclusion

The market study confirms a real and accessible opportunity. The Algerian handicraft sector represents roughly 450–460 billion DZD per year, with industry estimates valuing the sector at 1.9 billion USD and nearly 200,000 active online merchants, and the competitive landscape is fragmented enough to leave room for a focused player. Field interviews validated our two key design choices: the mutual rating system and the no-commission revenue model. Part 2 turns this opportunity into a five-year business and financial plan.

PART 2: BUSINESS MODEL AND DETAILED FINANCIAL STUDY

4.11. Introduction

This second part takes the market opportunity validated in Part 1 and turns it into a five-year business and financial plan. Hirftna is modelled on incorporation as an SAS. All amounts are in DZD. Key assumptions: IBS (Impôt sur les Bénéfices des Sociétés) at the standard 23% SME rate, except during the four years of the Decree 1275 Startup Label exemption (0% IBS, 0% TVA (Taxe sur la Valeur Ajoutée) on core services); inflation at 4% per year; 15% discount rate for NPV (Net Present Value); 2.5% terminal growth rate. The full model is in our Excel workbook. See below Appendices. [55]

4.12. Business Model Canvas

The complete Business Model Canvas is provided as a separate Excel file (see Appendix D). Below, we present a summary of each of the nine blocks.

4.12.1. Customer Segments

Two sides, three priority segments. On the supply side: home-based artisans (Persona Amel) and workshop owners (Persona Karim). On the demand side: Algerian buyers (Persona Sofiane). Priority ranking is home-based artisans first (the largest pool, biggest pain point we solve), then buyers (who pull the supply side forward), then workshop owners (higher revenue per seller but harder to onboard).

4.12.2. Value Proposition

For home-based artisans, Hirftna provides a professional sales channel that filters out unreliable clients through a mutual rating system, replacing the chaos of Instagram DMs. For workshop owners, Hirftna extends their reach beyond their region without requiring them to travel to fairs. For urban buyers, our project offers trust and selection that informal channels cannot match. The shared differentiator is the structured custom-order workflow combined with mutual rating.

4.12.3. Channels

The web platform (hirftna.vercel.app) [1] is the main point of entry. Customer acquisition relies on Instagram and Facebook pages, influencer partnerships, direct outreach to artisan cooperatives, and physical presence at events. Estimated CAC per

seller is around 1,600 DZD in Year 1, with a conversion rate from first visit to active seller of approximately 8%.

4.12.4. Customer Relationships

Self-service via the platform for most interactions. Personalized onboarding for the first hundred artisans, provided by the founder and later supported by the marketing and customer success team. AI chatbot (powered by Google Gemini) for 24/7 assistance. Private Telegram channel for the seller community to facilitate communication, share updates, and gather feedback.

4.12.5. Revenue Streams

Two streams over the five-year plan. In Year 1, revenue comes only from paid promotions: Featured Product and Shop Boost placements at 3,500 / 7,000 / 15,000 DZD for 7 / 14 / 30 days. From Year 2, we add tiered subscriptions (Free tier capped at 5 products, Standard at 800 DZD/month, Pro at 2,000 DZD/month). Premium subscriptions are paid through BaridiMob or CCP transfer, with manual admin verification (see section 4.13.1). No commission on sales.

4.12.6. Key Resources

The platform itself (proprietary code on GitHub). The founding team of three engineers. The growing seller community. Cloud infrastructure on Supabase, Vercel, and Render.

4.12.7. Key Activities

Platform development and maintenance, artisan onboarding and verification, content moderation and dispute resolution, marketing and user acquisition, community management, and customer support in Arabic, English, and French.

4.12.8. Key Partnerships

Logistics partners: Yalidine Express, ZR Express, and Maystro Delivery for desk-to-desk and desk-to-home Cash on Delivery. Payment partners: Chargily for online payment via CIB and Edahabia, and Algérie Poste for BaridiMob and CCP transfers. Institutional partners: the University of M'sila incubator (CATI), CNCSIU, ANADE for financing, and the Algerian Startup Fund (ASF). Sector

partner: the National Chamber of Handicrafts (CNAM). Technology partners: Supabase, Vercel, Render, and Google Gemini.

4.12.9. Cost Structure

Fixed costs dominate, especially in Year 1: payroll for the three-person founding team (2.42 million DZD/year including employer social charges), marketing (240,000 DZD), and a long tail of smaller items. Variable costs are minimal and tied mostly to payment processing (Chargily, around 2.5%) and cloud bandwidth overages from Year 2. Rent is zero because the start-up's head office is family-owned, saving approximately 240,000 DZD/year compared to a co-working space.

4.13. Production and Organisation Plan

4.13.1. Service Delivery Process

Hirftna is a software platform rather than a physical product, so the "production process" is the service flow that delivers value to users. A typical custom order goes through six sequential steps from seller registration to mutual rating.

Step	Activity	What happens
1	Seller registration and verification	Artisan signs up, completes profile, submits proof of artisan registration under Decree 96-01. Hirftna verifies and activates the account.
2	Product listing	Seller creates product listings with photos, category, price range, and craft technique. Up to 5 products on the Free tier.
3	Custom order initiation	Buyer browses, finds a product, and submits a custom order with budget range, deadline, and reference images.
4	Negotiation and acceptance	Seller accepts or proposes a counter-offer. Final price agreed within the platform flow.
5	Production and delivery	Seller produces the piece, posts progress updates, then ships via Yalidine, ZR Express, or Maystro, or arranges hand-to-hand delivery.
6	Completion and mutual rating	Buyer confirms receipt. Both sides rate each other publicly. Rating becomes part of the user's permanent profile.

Table 4.10: Service delivery process for a typical custom order on Hirftna.

For premium-tier subscriptions launched in Year 2 (Standard 800 DZD/month, Pro 2,000 DZD/month), the payment flow matches how Algerian commerce actually works. The artisan selects the tier in the dashboard, transfers the monthly fee via

BaridiMob or CCP to the platform's Algérie Poste account, then uploads the receipt screenshot. The admin panel manually verifies the transaction reference and activates the premium features within 24 hours. We chose manual verification deliberately: most Algerian users still prefer to keep card details off online platforms. From Year 3, we plan to add an automated Chargily flow alongside the BaridiMob option once user trust in online payment grows.

4.13.2. Supply and Procurement

Hirftna's "supply" is the set of cloud services, third-party APIs, and external tools that the platform depends on. We selected each on three criteria: availability in Algeria, free tier sufficient for the first year, and a clear upgrade path.

Supplier	What do we use it for	Payment method
Supabase	PostgreSQL database, authentication, file storage	Credit card via the founder's international debit card initially; ASF covers from Y2 once Startup Label is granted
Vercel	Frontend hosting (React 19 + Vite SPA)	Same as above
Render	Backend hosting (Node/Express)	Same as above
Chargily	CIB and Edahabia online payment processing	Algerian dinar, B2B (Business-to-Business) invoice
BaridiMob / CCP (Algérie Poste)	Manual transfers for premium subscriptions	Direct receipt to the platform CCP account
Google Gemini API	AI chatbot for user assistance	Pay-as-you-go, prepaid via Wise
Yalidine, ZR Express, Maystro	Delivery and Cash on Delivery logistics	B2B invoicing with the carrier

Table 4.11: Main suppliers and how we pay them.

Foreign currency payments are handled through two channels. In Year 1, the founders cover modest international payments through personal international debit cards or through services like Wise and Paysera, which are accessible to Algerian residents. Once the Startup Label is granted (expected end of Year 1), the Algerian Startup Fund explicitly allows coverage of overseas digital operational expenses for labeled startups, removing the foreign-currency constraint from Year 2 onwards.

4.13.3. Workforce

Year 1 starts with the three-person founding team (the founder as CEO, plus the recruited CTO and Marketing Lead). We add one new employee each year following a deliberate sequence: marketing first because user acquisition is the early bottleneck, then customer support as the seller base grows, then dedicated engineers as the platform scales.

Position	Y1	Y2	Y3	Y4	Y5	Monthly base (DZD)
CEO — General management	1	1	1	1	1	60,000
CTO — Engineering	1	1	1	1	1	60,000
Marketing & Communications Lead	1	1	1	1	1	40,000
Marketing Assistant	0	1	1	1	1	25,000
Customer Support Specialist	0	0	1	1	1	25,000
Junior Developer	0	0	0	1	1	35,000
Senior Developer	0	0	0	0	1	50,000
Total headcount	3	4	5	6	7	—

Table 4.12: Workforce evolution.

4.13.4. Key Partnerships

Hirftna depends on six partnership categories that together make the operation possible.

Partner type	Specific partners	Role
Institutional — Decree 1275	University of M'sila Incubator (CATI), CNCSIIU, Ministry of Higher Education	Mentoring, access to the Startup Label process, support in connecting with ASF and ANADE.
Public financing	Algerian Startup Fund (ASF), ANADE	ASF provides seed financing once the Startup Label is granted; ANADE provides the standard bank-loan-plus-interest-free package for young entrepreneurs.

Partner type	Specific partners	Role
Sector institutions	Chambre Nationale de l'Artisanat (CNAM), Ministry of Tourism and Handicrafts	Validation of artisan registration; potential co-marketing on national artisan campaigns.
Logistics	Yalidine Express, ZR Express, Maystro Delivery	Desk-to-desk and desk-to-home Cash on Delivery. Hirftna integrates shipping label printing through the carrier APIs.
Payment	Chargily, BaridiMob / Algérie Poste	Chargily handles CIB and Edahabia card payments; BaridiMob and CCP transfers serve the manual subscription verification flow.
Technology	Supabase, Vercel, Render, Google Gemini	Infrastructure and AI services on a pay-as-you-go basis with free tiers used during Year 1.

Table 4.13: Key partnerships and contributions.

4.14. Financial Projections

4.14.1. General Assumptions

Before any figure, we state the assumptions that underpin the projections. They were chosen to be defensible against scrutiny rather than to produce flattering outputs. The Decree 1275 Startup Label tax exemption (0% IBS for four years from incorporation) is applied because Hirftna qualifies for the label as a student-led innovative project. Salaries for the founding team are set at moderate Year 1 levels (60,000 DZD/month for technical leadership and 40,000 DZD/month for marketing), reflecting the willingness of founders to accept below-market compensation during the validation phase in exchange for equity stakes, a common practice among Algerian early-stage startups.

Assumption	Value	Source / Rationale
Currency	DZD throughout	All revenues and costs are local.
Annual sales growth (sellers)	150 → 3,000 over 5 years	Moderate trajectory; below 2% market penetration by Year 5.

Assumption	Value	Source / Rationale
Inflation assumption	3% per year	conservative forward-looking estimate; Algerian CPI inflation eased from 4.0% in 2024 to 1.7% over the first 9 months of 2025, with the World Bank projecting moderate stability in 2026-2027 [45].
IBS corporate tax rate	0% Y1-Y4, 23% Y5	Decree 1275 Startup Label exemption for 4 years; standard SME rate after.
TVA on core services	0% during the label period	Decree 1275 Startup Label exemption.
TAP (Professional Activity Tax) turnover tax	Abolished	Removed in recent Algerian budget laws.
Employer social charges	26% on top of gross	Standard CNAS (Caisse Nationale d'Assurances Sociales) employer contribution.
Discount rate (NPV)	15%	Conservative-to-realistic range for early-stage Algerian startups.
Terminal growth rate	2.5%	Long-run growth assumption for the Algerian digital sector.
DSO (Days Sales Outstanding) (client payment delay)	3 days	Chargily settlement for promotion and subscription payments.
DPO (Days Payable Outstanding) (supplier payment delay)	10 days	Weighted average of immediate-pay cloud and 30-day local B2B.
DIO (Days Inventory Outstanding) (inventory days)	0 days	Pure marketplace, no inventory held.

Table 4.14: General financial assumptions used across all projections.

4.14.2. Initial Investment Plan (CAPEX)

Hirftna is asset-light. Year 1 CAPEX (Capital Expenditure) totals 969,000 DZD across ten items. The rest of the funding raised goes into operating expenses and working capital.

#	Item	Qty	Unit price (DZD)	Total (DZD)	Depreciation
1	Developer-grade laptops	3	110,000	330,000	5 years
2	External 24-inch monitors	3	22,000	66,000	5 years
3	Office furniture (desk + chair)	3	80,000	240,000	5 years
4	Test smartphones (Android + iOS)	2	50,000	100,000	5 years
5	Domain name + 5-year SSL	1	8,000	8,000	5 years
6	Trademark registration (INAPI)	1	25,000	25,000	5 years
7	SAS legal incorporation (notary+CNRC+NIF)	1	90,000	90,000	5 years
8	Brand identity design	1	50,000	50,000	5 years
9	Product photography equipment	1	35,000	35,000	5 years
10	Workstation peripherals	1	25,000	25,000	5 years
	TOTAL Year 1 CAPEX			969,000	

Table 4.15: Initial investment plan (CAPEX) Year 1.

For Years 2 through 5, replacement and growth CAPEX is modelled at 150,000 / 200,000 / 250,000 / 300,000 DZD, respectively. The 90,000 DZD for SAS incorporation reflects the cost of drafting flexible bylaws compatible with the Decree 1275 framework. The full CAPEX is depreciated linearly over five years.

4.14.3. Revenue Projections — Three Scenarios

We modelled three scenarios to test the robustness of the project. The Realistic scenario is the base case and feeds directly into the P&L. Pessimistic halves seller acquisition and conversion; Optimistic adds about 60% to the Realistic case. A per-sale commission was deliberately excluded from the model. Because around 75% of Algerian transactions happen through Cash on Delivery or hand-to-hand exchanges that do not pass through the platform, making any commission projection unenforceable in practice [45].

Scenario	Year 1	Year 2	Year 3	Year 4	Year 5
Pessimistic — active sellers	100	250	500	900	1,400
Pessimistic — Revenue (DZD)	570,000	2,232,000	5,238,000	11,556,000	21,780,000
Realistic — active sellers	150	400	900	1,800	3,000
Realistic — Revenue (DZD)	855,000	4,464,000	10,476,000	23,112,000	43,560,000
Optimistic — active sellers	200	600	1,500	3,200	5,000
Optimistic — Revenue (DZD)	1,140,000	6,696,000	17,460,000	41,088,000	72,600,000

Table 4.16: Revenue projections under the three scenarios.

The Pessimistic scenario still reaches break-even by Year 3 and turns profitable from Year 4. The Optimistic scenario would put us close to break-even already in Year 2. The rest of this section uses the Realistic scenario unless otherwise stated.

Revenue breakdown by stream (Detailed Revenue Breakdown)

The table below decomposes the Realistic scenario into its four underlying revenue streams. Year 1 revenue comes entirely from promotion sales; the two subscription tiers launch in Year 2 once the platform has built enough trust with the seller community.

Stream / Metric	Year 1	Year 2	Year 3	Year 4	Year 5
Featured Product — Quantity (promos sold)	60	200	500	1,200	2,500
Featured Product — Avg price HT (DZD)	5,700	5,700	5,700	5,700	5,700
Featured Product — Revenue (DZD)	342,000	1,140,000	2,850,000	6,840,000	14,250,000
Shop Boost — Quantity (promos sold)	90	200	500	1,200	2,500
Shop Boost — Avg price HT (DZD)	5,700	5,700	5,700	5,700	5,700
Shop Boost — Revenue (DZD)	513,000	1,140,000	2,850,000	6,840,000	14,250,000
Standard Subscription — User-months	0	1,920	4,800	10,800	18,000
Standard Subscription — Price (DZD/month)	—	800	800	800	800

Stream / Metric	Year 1	Year 2	Year 3	Year 4	Year 5
Standard Subscription — Revenue (DZD)	0	1,536,000	3,840,000	8,640,000	14,400,000
Pro Subscription — User-months	0	324	468	396	330
Pro Subscription — Price (DZD/month)	—	2,000	2,000	2,000	2,000
Pro Subscription — Revenue (DZD)	0	648,000	936,000	792,000	660,000
TOTAL REVENUE (DZD)	855,000	4,464,000	10,476,000	23,112,000	43,560,000
Year-over-year growth rate	—	+422%	+135%	+121%	+88%

Table 4.17: Detailed revenue breakdown by stream.

Reading the table: promotions stay the dominant stream in Year 5 revenue overall because the 5,700 DZD average price reflects a weighted mix of 7-day, 14-day, and 30-day packages, and the volume scales linearly with the seller base. Standard subscriptions become the second-largest line by Year 3 because the 800 DZD/month price aligns with Persona Amel's willingness-to-pay range identified in our field interviews. The Pro tier remains deliberately smaller: it targets workshop owners like Persona Karim, who are fewer in number but pay more per month.

4.14.4. Projected Income Statement

The five-year P&L below comes from the Excel financial model. The IBS line reflects the Decree 1275 Startup Label tax exemption: 0% for Years 1 through 4 (label period), 23% for Year 5 (after the label expires). All amounts in DZD.

Line item	Year 1	Year 2	Year 3	Year 4	Year 5
Revenue	855,000	4,464,000	10,476,000	23,112,000	43,560,000
Direct variable costs	(51,375)	(251,600)	(591,900)	(1,237,800)	(2,189,000)
Gross margin	803,625	4,212,400	9,884,100	21,874,200	41,371,000
External charges	(589,000)	(1,097,000)	(1,877,000)	(2,954,000)	(4,197,000)
Payroll (incl. social charges)	(2,419,200)	(2,797,200)	(3,175,200)	(3,704,400)	(4,460,400)
EBITDA (Earnings Before Interest, Taxes,	(2,204,575)	318,200	4,831,900	15,215,800	32,713,600

Line item	Year 1	Year 2	Year 3	Year 4	Year 5
Depreciation, and Amortization)					
Depreciation	(193,800)	(223,800)	(263,800)	(313,800)	(373,800)
EBIT (Earnings Before Interest and Taxes)	(2,398,375)	133,400	4,607,100	14,941,000	32,339,800
IBS (0% — Startup Label active)	—	—	—	—	(7,438,154)
NET RESULT	(2,398,375)	133,400	4,607,100	14,941,000	24,901,646
EBITDA margin	-257.9%	7.1%	46.1%	65.8%	75.1%

Table 4.18: Five-year projected income statement.

Year 1 closes with a net loss of approximately 2.4 million DZD, fully covered by the 4 million DZD of funding raised. The platform turns profitable in Year 2, and net profit grows to 24.9 million DZD by Year 5. The EBITDA margin improvement from negative 258% in Year 1 to positive 75% by Year 5 reflects the operating leverage typical of marketplace platforms: once the fixed costs are covered, additional revenue flows mostly to the bottom line. Cumulative net profit over the five years reaches approximately 42.1 million DZD.

4.14.5. Operating Expenses Breakdown

Direct variable costs

These move proportionally with revenue. They include Chargily transaction fees (around 2.5% of revenue processed online), SMS and email notification services, and cloud bandwidth overages once we exceed the free tiers of Supabase and Vercel. They stay between 5% and 6% of revenue across the five years.

Payroll

Payroll is the biggest cost driver in Year 1, accounting for 2.42 million DZD against a revenue of only 855,000 DZD. The three-person founding team's total annual cost is structured as follows.

Role	Monthly gross	Annual gross	Employer charges 26%	Annual cost
CEO	60,000	720,000	187,200	907,200

Role	Monthly gross	Annual gross	Employer charges 26%	Annual cost
CTO	60,000	720,000	187,200	907,200
Marketing Lead	40,000	480,000	124,800	604,800
TOTAL Year 1	160,000	1,920,000	499,200	2,419,200

Table 4.19: Year 1 payroll breakdown.

External charges (overhead)

Item	Y1	Y2	Y3	Y4	Y5
Marketing and acquisition	240,000	600,000	1,200,000	2,000,000	3,000,000
Cloud hosting (Supabase, Vercel, Render)	11,000	83,000	120,000	180,000	250,000
Subcontracting (freelances)	24,000	60,000	120,000	240,000	360,000
Accountant	60,000	84,000	120,000	144,000	144,000
Legal (lawyer + notary)	60,000	25,000	25,000	25,000	25,000
Telecom and internet	48,000	54,000	60,000	60,000	60,000
Other (utilities, insurance, supplies)	146,000	191,000	232,000	305,000	358,000
Office rent	0	0	0	0	0
TOTAL External charges	589,000	1,097,000	1,877,000	2,954,000	4,197,000

Table 4.20: External operating charges over 5 years.

Note: the office rent is 0, as it is family-owned.

4.14.6. Cash Flow Projection

Cash flow is the most critical document for a startup because it tells us exactly when we might run out of money. We present the monthly view for Year 1 (where cash is tightest) and the annual view for Years 2-5.

Year 1 monthly cash flow (DZD)

Month	Receipts	Disbursements	Monthly balance	Cumulative
M0 (funding inflow)	4,000,000	(969,000)	3,031,000	3,031,000

Month	Receipts	Disbursements	Monthly balance	Cumulative
M1	0	(254,076)	(254,076)	2,776,924
M2	5,000	(254,076)	(249,076)	2,527,848
M3	20,000	(254,076)	(234,076)	2,293,772
M4	35,000	(254,076)	(219,076)	2,074,696
M5	50,000	(254,076)	(204,076)	1,870,620
M6	65,000	(254,076)	(189,076)	1,681,544
M7	80,000	(254,076)	(174,076)	1,507,468
M8	95,000	(254,076)	(159,076)	1,348,392
M9	110,000	(254,076)	(144,076)	1,204,316
M10	125,000	(254,076)	(129,076)	1,075,240
M11	135,000	(254,076)	(119,076)	956,164
M12	135,000	(254,076)	(119,076)	837,088
Year 1 totals	4,855,000	(4,017,912)	837,088	837,088

Table 4.21: Year 1 monthly cash flow projection.

The cumulative column shows the actual bank account trajectory, combining the M0 financing inflow (4,000,000 DZD from founders' equity, Prime Startup 1275, and ANADE) with the monthly operational activity. The closing position at M12 lands at approximately +837,088 DZD, which confirms that the 4 million DZD financing envelope is sized adequately to absorb the 2,398,375 DZD net loss of Year 1 with a comfortable safety margin before revenue grows in Year 2. The flat monthly disbursement of 254,076 DZD is the annual operating cost (payroll, external charges, and direct variable costs) spread evenly across the twelve months, net of the small working capital benefit from the 3-day client settlement and 10-day supplier payment terms. The formal three-section cash flow statement (operating, investing, and financing flows separated in the standard Algerian SCF format) is provided in Appendix C.

From Year 2 onwards, operating cash flow turns positive, and the cash position grows steadily: it reaches approximately 20.5 million DZD by the end of Year 4 and approximately 45.4 million DZD by the end of Year 5.

4.14.7. Required Capital Calculation

Before presenting the financing plan, we justify the total funding target by computing the required capital using the standard formula expected by the CNCSIU evaluation grid. The formula adds three components: the initial investment (CAPEX), the working capital needed to cover operating expenses during the cash-negative period before revenue ramps up, and a safety margin to absorb unforeseen costs [56].

Component	Value (DZD)	How it is computed
Investment costs (CAPEX Year 1)	969,000	Sum of the 10 CAPEX items from section 4.14.2.
Annual operating costs	3,008,200	External charges (589,000) + Payroll incl. social charges (2,419,200).
Monthly operating costs	250,683	Annual operating costs / 12.
Coverage period	7 months	Months of OPEX (Operating Expenditure) coverage are needed before revenue covers fixed costs (break-even is around month 18).
Working capital required	1,754,783	Monthly OPEX $\times$ 7 months' coverage.
Safety margin rate	8%	Applied to (CAPEX + Working capital) to cover unforeseen costs.
Safety margin amount	217,903	$8\% \times (969,000 + 1,754,783)$.
TOTAL REQUIRED CAPITAL	2,941,686	CAPEX + Working capital + Safety margin.
Rounded financing target	4,000,000	Rounded up to 4,000,000 DZD to provide a safety cushion above the calculated minimum of 2,941,686 DZD; the envelope remains well within the ANADE financing ceiling for student-led projects.

Table 4.22: Required capital calculation following the standard CNCSIU format.

The arithmetic justifies a minimum required capital of approximately 2.94 million DZD, rounded up to 4 million DZD to provide a safety cushion absorbed by the ANADE financing envelope. Reading the result, looking at the breakdown, of every 100 DZD of capital raised, approximately 24 DZD goes into CAPEX (laptops, smartphones, legal incorporation, trademark, brand identity), 44 DZD goes into

operating expenses during the seven-month coverage period, 5 DZD is held as a safety reserve, and 27 DZD remains as an additional liquidity buffer. The capital structure described in the next section maps this requirement to three concrete funding sources.

4.14.8. Financing Plan and Capital Structure

Total funding raised: 4 million DZD across three sources. We deliberately avoided outside equity in Year 1 to retain full operational control during the validation phase. The financing plan integrates the Decree 1275 mechanisms (Prime Startup 1275, ANADE) that are specifically designed for projects like ours.

Source	Amount (DZD)	Share	Conditions	Timeline
Founders' personal contribution	1,000,000	25%	Equity, fully paid at SAS incorporation	M0
PRIME Startup 1275	500,000	12.5%	Non-refundable grant for labelled startups [43] [44].	M1–M3
ANADE financing	2,500,000	62.5%	Mixed: ~700k bank loan + ~300k interest-free; 5-yr repayment; 1-yr grace	M3–M6
TOTAL	4,000,000	100%	—	—

Table 4.23: Funding plan and capital structure.

Note on the funding structure: the decision to keep the founding team's contribution at 33% rather than 51% was a small thing on paper, but it mattered to us internally. It signals that the founders are financially committed and personally invested in the success of the company without making the SAS structure feel like a closed family company.

4.14.9. Profitability and Viability Indicators

We computed the standard set of indicators across the three scenarios. The values incorporate the Decree 1275 Startup Label tax exemption.

Indicator	Realistic	Pessimistic	Optimistic	Comment
Break-even point	Year 2 ($\approx$ mid-Year 2)	Year 3 ($\approx$ month 30)	Year 2 ($\approx$ month 14)	Revenue needed to cover all fixed costs.

Indicator	Realistic	Pessimistic	Optimistic	Comment
Contribution margin rate	≈ 94%	≈ 94%	≈ 94%	Variable costs are minimal, typical for a digital marketplace.
NPV @ 15% discount	≈ 118 million DZD	≈ 45 million DZD	≈ 205 million DZD	Includes 5-year cash flows + terminal value at 2.5% growth.
IRR (Internal Rate of Return)	≈ 78%	≈ 38%	≈ 115%	All scenarios exceed the 15% cost of capital.
Payback period	≈ 29 months	≈ 44 months	≈ 24 months	Time to recover the initial 3 M DZD investment.
5-year cumulative ROI	≈ 14×	≈ 7×	≈ 24×	Cumulative net profit divided by initial investment.

Table 4.24: Profitability indicators under the three scenarios.

Every indicator is positive in all three scenarios. Even in the Pessimistic case, IRR at 38% comfortably exceeds our 15% cost of capital, NPV stays strongly positive, and the project recovers the initial investment within four years. These indicators establish the financial viability of Hirftna across all three scenarios.

4.15. Financial and Operational Risk Analysis

No financial plan is complete without an honest discussion of what could go wrong. We identified six principal risks and rated each on likelihood and impact.

Risk	Likelihood	Impact	Criticality	Mitigation
Slower seller acquisition than projected	Medium	High	High	Conservative funding mix gives approximately 16-18 months' runway until operational break-even in Year 2. Direct cooperative partnerships as a fallback.
Low subscription conversion in Year 2	Medium	Medium	Medium	Year 1 is dedicated to trust-building before any paid feature is mandatory. A hard 5-product cap on the Free tier creates natural upgrade pressure.

Risk	Likelihood	Impact	Criticality	Mitigation
Cash flow shortfall in months 10–12 of Year 1	Medium	High	High	Active monthly cash monitoring. Reserve of ~470k DZD held untouched (8% safety margin + rounding headroom from the 3 M DZD funding envelope). ANADE's second tranche can be requested if needed.
Cloud cost overruns	Low	Medium	Low	Budget buffer in Year 5 (250k DZD vs ~120k expected). Migration path to self-hosted PostgreSQL kept open.
Regulatory change on Cash on Delivery	Low	High	Medium	The platform already supports both COD and online payment. Mix can shift to Chargily without changing the model.
Dependency on a single logistics partner	Low	Medium	Low	Three logistics partners integrated (Yalidine, ZR Express, Maystro). Hand-to-hand delivery supported.

Table 4.25: Financial and operational risk matrix.

4.16. Intellectual Property and Legal Framework

4.16.1. Startup Labeling Process via startup.dz

The Algerian Decree 1275 framework defines a two-step labeling process managed through the national platform startup.dz. We will follow this path.

Step 1: Label Projet Innovant. This is the pre-incorporation label, obtained before the thesis defense. The application requires a pitch deck, a SWOT analysis (provided in section 4.7), and a short video demonstration of the MVP. The label gives access to the incubator infrastructure: free domiciliation, mentoring, and workshops. It also puts the project in line for the post-defense Startup Label.

Step 2: Label Startup. Granted, after graduation, once the SAS is incorporated. This label activates the four-year tax holiday (0% IBS and 0% TVA on the platform's core services), eligibility for ASF financing, and the institutional support of the National

Coordinating Committee (CNCSIU) [56]. The post-defense incorporation is the formal step that activates these benefits.

4.16.2. Intellectual Property

Two protection measures are planned for the project, and one is deliberately set aside. The Hirftna trademark (name and Arabic logo حرفتنا) will be filed with INAPI (Institut National Algérien de la Propriété Industrielle), securing exclusive commercial rights on the brand identity in Algeria. The platform source code will be deposited with ONDA (Office National des Droits d'Auteur et Droits Voisins) under copyright law, providing legal proof of ownership of the original codebase. Patent protection is not pursued at this stage because the innovations in Hirftna are mainly UX, workflow design (the Custom Order System), and the mutual rating system, none of which qualify as patentable technical inventions under Algerian law.

4.16.3. Legal Structure

The company will be incorporated as an SAS (Société par Actions Simplifiée) under Algerian law. The SAS form is chosen over SARL (Société à Responsabilité Limitée) because the SAS was specifically introduced into the Algerian Commercial Code by Loi 22-09 to meet the needs of startups and to work hand in hand with Decree 1275 and the Algerian Startup Fund. The bylaws give the President (CEO) operational authority, while major decisions (capital increases, equity transfers, new shareholders) require unanimous agreement among the founding shareholders. Share capital is set at 500,000 DZD, fully paid at incorporation, divided among the three founders according to the equity distribution defined in the founding agreement. [57]

4.16.4. Administrative Formalities and Compliance

The company will register with the CNRC (Centre National du Registre du Commerce) for the commercial registry, obtain the NIF (Numéro d'Identification Fiscale) for tax identification, and the NIS (Numéro d'Identification Statistique) for the statistical office, as well as CASNOS (Caisse Nationale de Sécurité Sociale des Non-Salariés) social security registration for the founder-managers. Two laws directly apply to the platform's activities: Law 18-05 on Electronic Commerce, which governs platform operators and consumer rights, and Law 18-07 on Personal Data Protection, which establishes the framework for handling user data. The platform's authentication

system (with two-factor authentication), encryption of stored data, role-based access through Supabase Row Level Security, and explicit consent flows on sign-up are all designed to comply with these two laws. [8] [49] [53]

4.17. Development Roadmap (18–24 Months)

The roadmap below covers the first 24 months after the official launch. Each milestone has a clear success indicator.

Period	Milestone	Owner	Success indicator
M0–M3	MVP polish and stress-testing; onboarding of first beta artisans	CEO + CTO	30 beta artisans onboarded; core flows stable in production
M3–M6	SAS incorporation; soft launch in M'sila and Algiers	Full team	Legal entity registered; 100 active sellers
M6–M12	Scale to 150 sellers; iterate UX based on real usage; first paid promotion conversions	Marketing Lead	150 active sellers; 855k DZD revenue achieved
M12–M18	Launch tiered subscriptions; hire a Marketing Assistant	CEO + Marketing	30% paid conversion rate; 400 active sellers
M18–M24	Geographic expansion to all major wilayas; first ASF Series A discussions	CEO + CTO	900 active sellers in progress; Year 3 trajectory confirmed

Table 4.26: Development roadmap over the first 24 months.

Part 2 Conclusion

With 4 million DZD of initial funding, Hirftna absorbs a Year 1 loss of approximately 2.4 million DZD, turns profitable in Year 2, and accumulates approximately 45 million DZD of cash by the end of Year 5. NPV at a 15% discount rate reaches around 118 million DZD. The model rests on assumptions that are realistic for the Algerian market: a revenue model adapted to how Algerian commerce actually works, a conservative subscription conversion curve, a careful year-by-year hiring plan, and the Decree 1275 mechanisms (four-year tax exemption, ANADE financing) treated as concrete elements of the plan, not vague benefits.

Chapter 4 Conclusion

Taken together, the two parts indicate that Hirftna is a viable Algerian startup under Decree 1275. The single most important decision in the whole project is to drop the per-sale commission model that international marketplaces use and to replace it with promotions and freemium subscriptions paid upfront, with manual BaridiMob verification during the early subscription phase. This choice reflects the dominance of Cash on Delivery and hand-to-hand exchanges in Algerian commerce, and it makes every dinar of projected revenue something we can actually collect. The next concrete step is to finalize the SAS incorporation, begin onboarding the first hundred artisans during the incubation phase at the University of M'sila, and progressively turn the financial plan presented here into operational reality.

Conclusion

This report documented the full development and economic study of our hybrid marketplace for Algerian artisans, Hirftna. The first three chapters covered the software engineering side: business analysis, where we studied the domain and defined the requirements; system design, where we modeled the architecture, database, and user interactions using C4 and UML diagrams; and implementation and deployment, where we built and launched the platform using Node.js v20, Express.js v5, React 19, and Supabase. Chapter 4 then turned the project into a startup case under the Decree 1275 framework, combining a detailed market study (PESTEL, TAM/SAM/SOM, personas, competitive analysis, SWOT, and field interviews with Algerian artisans) with a five-year business model and financial plan that confirmed the project's viability under a 4 million DZD seed funding envelope.

The result is a working MVP that lets artisans list handmade products and handle custom orders through a structured lifecycle, while clients can browse, place orders, and rate sellers, and sellers can rate clients back through a mutual accountability system. The core features are in place: the custom order flow, bilingual Arabic/English/French support with full RTL layout, an AI chatbot powered by Gemini 2.5 Flash, two-factor authentication (functional in the test environment, production deployment pending due to hosting constraints), seller verification, and a promotion system. Several directions for future work remain: integrating real payment processing through Chargily, adding an optional messaging channel for complex orders, introducing subcategories to improve product discovery as the catalog grows, and automating admin workflows such as promotion activation and expiry. The platform is deployed and accessible online, and although there is still a lot to grow, the foundation is in place: a marketplace where the order itself structures the conversation, and where both sides of the transaction are held accountable.

References

- [1] Hirftna Marketplace, [Online]. Available: <https://hirftna.vercel.app/>.
- [2] Gassa, A., & Bengeutaf, A., "The role of Algerian traditional industries and crafts in," 04 2024. [Online]. Available: <https://asjp.cerist.dz/en/downArticle/351/9/4/253543>. [Accessed 2026].
- [3] Ministry of Tourism and Traditional Industry, "Legislation and Regulations of the Craft Industry," [Online]. Available: <https://www.mta.gov.dz/legislation-and-regulations-of-the-craft-industry/?lang=en>. [Accessed 2026].
- [4] M. Bouam, "The contribution of crafts to GDP exceeded 350 billion DA in 2022," [Online]. Available: <https://embrussels.mfa.gov.dz/en/news-and-press-releases/the-contribution-of-crafts-to-gdp-exceeded-350-billion-da-in-2022>.
- [5] Gassa, A., & Bengeutaf, A., "Support policies for Developing Traditional Industries and Crafts in Algeria," 06 2024. [Online]. Available: <https://asjp.cerist.dz/en/downArticle/57/11/1/252333>. [Accessed 2026].
- [6] Legal Doctrine, "Nouvelle plateforme d'inscription numérique en faveur des artisans algériens," 06 juin 2022. [Online]. Available: <https://legal-doctrine.com/edition/Nouvelle-plateforme-dinscription-num%C3%A9rique-en-faveur-des-artisans-alg%C3%A9riens>. [Accessed 2026].
- [7] "Chedda Tlemcénienne /Patrimoine immatériel de l'humanité," 19 12 2022. [Online]. Available: <https://almanach-dz.com/>. [Accessed 2026].
- [8] République Algérienne Démocratique et Populaire, "Loi n° 18-05 du 10 mai 2018 relative au commerce électronique," 10 may 2018. [Online]. Available: https://www.mpt.gov.dz/wp-content/uploads/2023/11/Loi-n%C2%B0-18-05-du-10-mai-2018.-fr_0_0.pdf. [Accessed 2026].
- [9] Etsy, Inc, "Etsy — Shop for handmade, vintage, custom, and unique gifts," [Online]. Available: <https://www.etsy.com/>. [Accessed 2026].
- [10] Ouedkniss SARL, "Ouedkniss — Algeria's leading online classifieds platform," [Online]. Available: <https://www.ouedkniss.com/>. [Accessed 2026].
- [11] R. Malika, "Integrating Social Media Into E-commerce: Prospects And Obstacles In The Algerian Market," 20 12 2025. [Online]. Available: <https://asjp.cerist.dz/en/article/282089>. [Accessed 2026].
- [12] "Product-Service Hybrids," [Online]. Available: <https://ixdf.org/literature/topics/product-service-hybrids>. [Accessed 2026].

- [13] GeeksforGeeks, "Client-Server Architecture – System Design.," 10 Feb 2026. [Online]. Available: <https://www.geeksforgeeks.org/client-server-model/>. [Accessed 2026].
- [14] Coursera, "What Is Client-Server Architecture?," 26 Nov 2025. [Online]. Available: <https://www.coursera.org/articles/client-server-architecture>. [Accessed 2026].
- [15] Fielding, R. T., "Architectural Styles and the Design of Network-Based Software Architectures," 2000. [Online]. Available: https://ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. [Accessed 2026].
- [16] Express.js, "express.js," [Online]. Available: <https://expressjs.com/en/>. [Accessed 2026].
- [17] "Gemini API," [Online]. Available: <https://ai.google.dev>. [Accessed 2026].
- [18] "Supabase Documentation," [Online]. Available: <https://supabase.com/docs>. [Accessed 2026].
- [19] "Postgresql," [Online]. Available: <https://www.postgresql.org/docs/>. [Accessed 2026].
- [20] IJRPR, "React SPA Features: Component-Based Architecture and Virtual DOM.," [Online]. Available: <https://ijrpr.com/uploads/V6ISSUE5/IJRPR45613.pdf>.
- [21] GeeksforGeeks, "Component Architecture in React: Atomic Design Pattern.," [Online]. Available: <https://www.geeksforgeeks.org/reactjs/react-component-based-architecture/>. [Accessed 2026].
- [22] R. Documentation, "Components and Props," [Online]. Available: <https://react.dev>. [Accessed 2026].
- [23] "Visual Studio Code," [Online]. Available: <https://code.visualstudio.com/>. [Accessed 2026].
- [24] V. C. Documentation, "Why Did We Build Visual Studio Code?," [Online]. Available: code.visualstudio.com/docs. [Accessed 2026].
- [25] "Postman API Platform," [Online]. Available: <https://postman.com>. [Accessed 2026].
- [26] "Git," [Online]. Available: <https://git-scm.com>. [Accessed 2026].
- [27] "GitHub," [Online]. Available: <https://github.com/about>. [Accessed 2026].
- [28] starUMLDocumentation, "starUML Documentation," [Online]. Available: <https://docs.staruml.io/>. [Accessed 2026].

- [29] "Node.js," [Online]. Available: <https://nodejs.org>. [Accessed 2026].
- [30] "Zod," [Online]. Available: <https://zod.dev/>. [Accessed 2026].
- [31] "Winston," [Online]. Available: <https://github.com/winstonjs/winston>. [Accessed 2026].
- [32] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," [Online]. Available: <https://www.rfc-editor.org/info/rfc7519>. [Accessed 2026].
- [33] "Nodemailer," [Online]. Available: <https://nodemailer.com>. [Accessed 2026].
- [34] "React.js," [Online]. Available: <https://react.dev/>. [Accessed 2026].
- [35] "Vite," [Online]. Available: <https://vite.dev/>. [Accessed 2026].
- [36] "Tailwind CSS," [Online]. Available: <https://tailwindcss.com/>. [Accessed 2026].
- [37] "React Router," [Online]. Available: <https://reactrouter.com>. [Accessed 2026].
- [38] "i18next," [Online]. Available: i18next.com. [Accessed 2026].
- [39] "Axios," [Online]. Available: <https://axios-http.com>. [Accessed 2026].
- [40] "Chargily," [Online]. Available: <https://chargily.com>. [Accessed 2026].
- [41] "Vercel," [Online]. Available: <https://vercel.com/>. [Accessed 2026].
- [42] Render Cloud Services, "Render," cloud application hosting for developers, 2024. [Online]. Available: <https://render.com/>. [Accessed 2026].
- [43] République Algérienne, "'Arrêté ministériel n° 1275 du 27 septembre 2022 portant création du dispositif 'Un Diplôme, Une Startup / Un Brevet',", "Ministère de l'Enseignement Supérieur et de la Recherche Scientifique, 2022.
- [44] ASJP (CERIST), "Arrêté ministériel n° 1275 : Création de valeurs," 2024. [Online]. Available: <https://asjp.cerist.dz/en/downArticle/360/22/1/222404..> [Accessed 2026].
- [45] World Bank Group, "Algeria Economic Update — Fall 2025: Responding to Climate Challenges and Supporting Sustainable Development. Washington, D.C.: World Bank Group," 4 December 2025. [Online]. Available: <https://www.worldbank.org/en/news/press-release/2025/12/04/algeria-responding-to-climate-challenges-and-supporting-sustainable-development>. [Accessed 2026].
- [46] World Bank Group, "'Algeria Economic Update — Spring 2024," World Bank Group, Washington DC, 2024. [Online]. Available: <https://openknowledge.worldbank.org/entities/publication/17f58d02-e6e3-4e08-81eb-401c9e06d897>. [Accessed 2026].

- [47] ECDB (E-Commerce Database), "Algeria E-Commerce Market Report 2025," 2025. [Online]. Available: <https://ecdb.com/markets/dz/all>. [Accessed 2026].
- [48] The North Africa Journal, "Algeria's E-Commerce Growth Signals New Phase for Regional Trade," 2025. [Online]. Available: <https://north-africa.com/algerias-e-commerce-growth-signals-new-phase-for-regional-trade/>. [Accessed 2026].
- [49] République Algérienne, "Décret législatif n° 96-01 du 10 janvier 1996 fixant les règles régissant l'activité de l'artisanat et des métiers," *Journal Officiel*, 1996.
- [50] UNCTAD (United Nations Conference on Trade and Development), "Algeria: eTrade Readiness Assessment. Geneva, August 2025," [Online]. Available: <https://unctad.org/publication/algeria-etrade-readiness-assessment>. [Accessed 2026].
- [51] United Nations Population Fund (UNFPA) / World Population Prospects 2024 Revision, "Algeria — Demographic Dashboard 2025," [Online]. Available: <https://www.unfpa.org/data/world-population/DZ>. [Accessed 2026].
- [52] Selma R, "Internet mobile en Algérie : Taux de pénétration de 116% en 2025," 2025. [Online]. Available: <https://lesenjeuxeco.dz/internet-mobile-en-algerie-taux-de-penetration-de-116-en-2025/>. [Accessed 2026].
- [53] République Algérienne Démocratique et Populaire, "Loi n° 18-07 du 10 juin 2018 relative à la protection des personnes physiques dans le traitement des données à caractère personnel," *Journal Officiel*, 2018.
- [54] Warcha, "Warcha," [Online]. Available: <http://warcha-dz.com/>. [Accessed 2026].
- [55] Bank, Trading Economics / World, "Algeria — Inflation, Consumer Prices (annual %)," 2024. [Online]. Available: <https://tradingeconomics.com/algeria/inflation-consumer-prices-annual-percent-wb-data.html>. [Accessed 2026].
- [56] CNCSIU, "Guide de Projet pour l'obtention du Certificat de Startup dans le cadre de l'Arrêté Ministériel 1275," Ministère de l'Enseignement Supérieur, 2022.
- [57] Journal Officiel de la République Algérienne, "Loi n° 22-09(SAS dans le code de commerce)," 5 mai 2022. [Online]. Available: https://droit.mjustice.gov.dz/sites/default/files/Loi_n%C2%B0_22-09.pdf. [Accessed 2026].

APPENDICES

The standard appendices below correspond to the documents the CNCSIIU evaluation grid expects: the startup balance sheet, the projected income statement, the treasury statement, and the Business Model Canvas. Each financial appendix points to the corresponding tab in the Excel workbook (Hirftna_Financial_Model.xlsx), where the complete tables, formulas, and yearly detail can be inspected. One additional academic appendix, the field interview guide, follows. All financial figures across the appendices have been computed using the Excel financial model attached to this thesis.

Appendix A: Startup Balance Sheet (Bilan de Startup)

The five-year projected balance sheet of Hirftna sits in the Excel financial model under the tab B.4 Bilan. The structure follows what the CNCSIIU evaluation grid expects: assets on one side, equity and liabilities on the other, with a verification row at the bottom that checks the balance equilibrium for every year. The Check row returns 0 across all five years.

The headline figures: total equity moves from approximately –898,000 DZD at the end of Year 1 (the 2.40 million DZD startup loss exceeds the 1.5 million DZD equity portion of the funding raised) to approximately 43.6 million DZD by Year 5, which is the result of accumulated profits during the four-year tax exemption window granted by Decree 1275.

File reference: Hirftna_Financial_Model.xlsx, tab B.4 Bilan. The full table lists fixed assets, receivables, cash, share capital (founders' equity plus the Prime Startup 1275 grant), long-term debt (ANADE financing), retained earnings, net income for the year, and trade payables.

Appendix B: Projected Income Statement (Compte de Résultat Prévisionnel)

The full five-year projected income statement, in the Algerian *Système Comptable Financier* (SCF) format, is in the Excel financial model. The summary version — all the key lines from Production de l'exercice down to Net Result — already appears as Table 4.18 in section 4.14.4.

File reference: Hirftna_Financial_Model.xlsx, tab B.1. P&L. The IBS line applies the Decree 1275 Startup Label exemption: 0% for Years 1 through 4 (label active), 23% in Year 5. Year 1 closes with a net loss of 2,398,375 DZD. From Year 2, the company is profitable, and the cumulative net result across the five years reaches approximately 42.1 million DZD.

Appendix C: Treasury / Cash Flow Statement (TFT)

The full cash flow statement in the standard Algerian SCF format (operating, investment, and financing flows separated) is provided in the Excel financial model. The monthly Year 1 view, which shows how the 4 million DZD funding envelope absorbs the startup loss month by month, already appears as Table 4.21 in the chapter body (section 4.14.6).

File reference: Hirftna_Financial_Model.xlsx, tab B.2. TFT. The closing trésorerie position grows from approximately +837,000 DZD at the end of Year 1 (after the 4 million DZD funding envelope absorbs the startup loss) to approximately 45.4 million DZD by the end of Year 5. The tab also includes the NPV computation at a 15% discount rate and the terminal value calculation at a 2.5% long-term growth rate.

Appendix D: Business Model Canvas (BMC)

The complete Business Model Canvas is provided as a separate Excel file (Hirftna_BMC.xlsx). The textual summary below covers each of the nine standard blocks.

Block	Content for Hirftna
Key Partners	Yalidine, ZR Express, Maystro (logistics) Chargily, BaridiMob / Algérie Poste (payment) Supabase, Vercel, Render (cloud) University of M'sila Incubator, CNCSIIU, ANADE, ASF (institutional) CNAM (Chambre Nationale de l'Artisanat et des Métiers), Ministry of Tourism (sector)
Key Activities	Platform development and maintenance Artisan onboarding and verification Content moderation and dispute resolution Marketing and user acquisition Community management Customer support (Arabic, English, and French)
Value Propositions	For artisans: a professional digital sales channel with mutual rating, no commission per sale, replacing chaotic Instagram DMs. For buyers: trusted access to authentic Algerian handicrafts with structured custom orders and COD support.

Block	Content for Hirftna
Customer Relationships	Self-service via the platform Personalized onboarding for the first 100 artisans AI chatbot (Gemini) for 24/7 assistance Private Telegram community for sellers
Customer Segments	Supply: home-based artisans (Persona Amel), workshop owners (Persona Karim) Demand: urban Algerian buyers (Persona Sofiane), diaspora segment (activation from Year 3)
Key Resources	Proprietary platform code Founding team (3 engineers) Growing seller community
Channels	Web platform (hirftna.vercel.app) Instagram and Facebook (organic + influencer partnerships) Direct outreach to artisan cooperatives Artisan salons and events (Salon de l'Artisanat)
Cost Structure	Fixed: payroll (2.42 M DZD Y1), marketing (240k DZD Y1), cloud (11k Y1), legal (60k Y1). Variable (~5-6% of revenue): Chargily fees, SMS/email, cloud overage. CAPEX Y1: 774k DZD. Rent: 0 DZD (free incubator domiciliation).
Revenue Streams	Year 1: Promotion fees only (3,500 / 7,000 / 15,000 DZD for 7/14/30 days). Year 2+: Promotions + tiered subscriptions (Free 5 products, Standard 800 DZD/mo via BaridiMob, Pro 2,000 DZD/mo). No commission per sale.

Appendix E: Field Interview Guide

The semi-structured interview guide used during the primary research (section 4.8) is reproduced below. Interviews were conducted in Arabic with notes taken in French. Twelve artisans participated: eight from M'sila and Biskra regions, four from Algiers.

Section 1 — Profile: Could you describe your craft activity? How long have you been doing it? Approximate monthly revenue from your artisan work?

Section 2 — Current sales channels: Where do you currently sell (physical shop, fairs, Instagram, Facebook, other)? What works in your current setup? What does not?

Section 3 — Recurring pain points: Have you ever had a client place an order and then disappear? How often does this happen? How do you handle several parallel orders at the same time? Have you ever had a dispute with a client over price or delivery?

Section 4 — Attitudes toward online commerce: Do you currently use online payment (CIB / Edahabia)? Why or why not? Have you used Yalidine COD? Would you trust an Algerian platform with your business?

Section 5 — Willingness to pay: If a platform brought you steady orders and protected you from unreliable clients, would you pay a monthly fee for it? Maximum acceptable amount? Preference between fixed subscription, per-sale commission, or paid promotion when you need visibility?

Section 6 — Reactions to the Hirftna concept (mockup shown): What is your first impression? Which features feel most useful? Which feels unnecessary? Would you sign up today?

Excel files links

Hirftna_Financial_Model.xlsx link:

https://docs.google.com/spreadsheets/d/1UxOonfJkZ_v1KQIEAao_uwJ7HPNGDpoo/edit?usp=drive_link&oid=100717235444263749812&rtpof=true&sd=true

Hirftna_BMC.xlsx link:

https://docs.google.com/spreadsheets/d/1mf3TECKi19AENFaLvOgWalte-LXS5XTf/edit?usp=drive_link&oid=100717235444263749812&rtpof=true&sd=true