

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

MOHAMED BOUDIAF UNIVERSITY - M'SILA

FACULTY: Mathematics and Computer  
Science

DEPARTMENT: Computer Science

N° : .....



DOMAIN: Mathematics and Computer  
Science

FIELD: Computer Science

OPTION: Information Systems and  
Software Engineering

**A Dissertation Submitted in Partial Fulfilment of the  
Requirements for the Degree of Academic Master**

**By:**

Amroune Adem Mahmoud

**SUBJECT**

**A Community Platform for Designers: Fostering  
Collaboration and Creativity**

**Board of Examiners:**

|                        |                      |            |
|------------------------|----------------------|------------|
| Dr. Mohamed BAHACHE    | University of M'sila | President  |
| Dr. Mahmoud BRAHIMI    | University of M'sila | Supervisor |
| Dr. Abdelouahab KETTAF | University of M'sila | Examiner   |

**Academic year: 2023 /2024**



## **DEDICATION**

*“I want to say Alhamdulillah, I want to thank my beautiful parents for everything they did for me and the effort they put in, and for supporting and dedicating their lives for us, so that we could become who we are. I extend my thanks to my sisters for their efforts, and I extend my thanks to my family, and to all my friends and anyone who has helped and supported me until this moment, and it has been a beautiful experience. We do not forget our people in Palestine and Gaza.”*

## **Acknowledgments**

*I would like to thank Dr. Mahmoud Brahimi for his efforts in helping me with this thesis*

*I thank a committee consisting of Dr. Bahache Mohamed and Kettaf Abdelouahab*

*For their efforts and experience*

*Also thanks to the friends and all the people who contributed support and help, and all the classmates and teachers*

## Table of content

|   |    |
|---|----|
| List of Figures .....   | vi |
| Introduction .....  | 1  |
| Chapter I. System Analysis .....                              | 2  |
| 1. Introduction .....   | 2  |
| 2. Presentation of the project.....                           | 2  |
| 2.1. Objectives of the project .....                          | 2  |
| 2.2. Project Features .....                                   | 2  |
| 2.1. Contribution of the project .....                        | 2  |
| 3. Structure of the project.....                              | 3  |
| 3.1. Individuals.....   | 3  |
| 3.2. Requirements.....  | 3  |
| 4. Community Building and Engagement in Design Platforms..... | 4  |
| 4.1. Importance of Community in the Design Field.....         | 4  |
| 4.2. Strategies for Building an Engaged Community.....        | 4  |
| 4.3. Enhancing User Interaction on Our Platform.....          | 4  |
| 5. Monetization Strategies for Design Platforms .....         | 5  |
| 5.1. Overview of Monetization Models .....                    | 5  |
| 5.2. Implementing E-Commerce on Design Platforms .....        | 5  |
| 5.3. Revenue Generation on Our Platform.....                  | 5  |
| 6. Competitor Analysis in the Design Platform Market.....     | 5  |
| 6.1. Examples of Real-World Competitors .....                 | 6  |
| 6.1.1. Behance .....  | 6  |
| 6.1.2. Dribbble.....  | 6  |
| 6.2. Problems and Challenges .....                            | 6  |
| 6.3. Value of Our Platform.....                               | 6  |
| 7. Advanced Web Technologies and Development .....            | 7  |

|                                   |   |    |
|-----------------------------------|---|----|
| 7.1.                              | What is a Web Application?.....                               | 7  |
| 7.2.                              | Architectural Foundations of Web Applications.....            | 7  |
| 7.3.                              | Components of Web Applications Architecture .....             | 7  |
| 7.3.1.                            | Modern Web Application Architecture.....                      | 9  |
| 7.4.                              | The difference between a web application and a website? ..... | 10 |
| 7.5.                              | Web Application Models .....                                  | 11 |
| 7.5.1.                            | Static Web Applications.....                                  | 11 |
| 7.5.2.                            | Dynamic Web Applications .....                                | 11 |
| 7.5.3.                            | E-Commerce Web Applications.....                              | 11 |
| 7.5.4.                            | Content Management System.....                                | 11 |
| 7.5.5.                            | Single-Page Applications .....                                | 12 |
| 7.5.6.                            | Multi-Page Applications .....                                 | 12 |
| 7.5.7.                            | Progressive Web Applications .....                            | 12 |
| 7.6.                              | The Web Application Development Lifecycle.....                | 12 |
| 7.6.1.                            | Planning and Discovery .....                                  | 12 |
| 7.6.2.                            | Design.....   | 12 |
| 7.6.3.                            | Development .....   | 13 |
| 7.6.4.                            | Testing.....  | 13 |
| 7.6.5.                            | Deployment .....  | 13 |
| 7.6.6.                            | Maintenance and Updates .....                                 | 13 |
| 8.                                | Tailoring our Platform for the Algerian Market.....           | 13 |
| 8.1.                              | Addressing the Unique Needs of Algerian Designers.....        | 13 |
| 8.2.                              | Empowering Algerian Designers .....                           | 14 |
| 8.3.                              | Bridging the Gap in Digital Resources .....                   | 14 |
| .9                                | Conclusion.....   | 14 |
| Chapter II. Design Analysis ..... |   | 14 |
| 1.                                | Introduction .....  | 14 |

|                                  |  |    |
|----------------------------------|--|----|
| 2.                               | What is Unified Modeling Language (UML) .....            | 14 |
| 3.                               | Diagrams .....   | 14 |
| 3.1.                             | Use Case Diagram .....                                   | 14 |
| 3.2.                             | Class Diagram .....                                      | 15 |
| 3.3.                             | Sequence Diagram.....                                    | 16 |
| 3.3.1.                           | Create an account .....                                  | 16 |
| 3.3.2.                           | Login User Diagram.....                                  | 18 |
| 3.3.3.                           | Upload User New Desing Diagram.....                      | 18 |
| 3.3.4.                           | Upload New Assets Diagram .....                          | 20 |
| 3.3.5.                           | Add Job Diagram .....                                    | 21 |
| 3.3.6.                           | Add Service Diagram .....                                | 22 |
| 4.                               | Conclusion.....  | 23 |
| Chapter III. Implementation..... |  | 24 |
| 1.                               | Introduction .....                                       | 24 |
| 2.                               | Integrated development environment.....                  | 24 |
| 2.1.                             | Technologies and Programming Languages .....             | 24 |
| 2.1.1.                           | JavaScript .....   | 24 |
| 2.1.2.                           | Html.....  | 24 |
| 2.1.3.                           | Css .....  | 25 |
| 2.2.                             | Database Management System.....                          | 25 |
| 2.2.1.                           | Appwrite.....  | 25 |
| 2.2.2.                           | Visuals at Appwrite .....                                | 26 |
| 2.3.                             | Frameworks and Libraries.....                            | 28 |
| 2.3.1.                           | Next Js .....  | 28 |
| 2.3.2.                           | React Js.....  | 28 |
| 2.3.3.                           | Tailwind Css.....  | 29 |
| 2.3.4.                           | GSAP (GreenSock Animation Platform, is a powerful) ..... | 29 |

|        |   |    |
|--------|---|----|
| 2.4.   | Development Tools .....                     | 29 |
| 2.4.1. | Visual Studio Code.....                     | 29 |
| 2.4.2. | Postman .....                               | 30 |
| 2.4.3. | Spline.....                                 | 30 |
| 2.4.4. | Figma.....                                  | 30 |
| 2.4.5. | Lucidchart.....                             | 31 |
| 3.     | Desing Ui/Ux.....                           | 31 |
| 3.1.   | Inspiration.....                            | 31 |
| 3.2.   | Design the platform logo and identity ..... | 32 |
| 3.3.   | Design Standers.....                        | 32 |
| 3.4.   | Platform Desing UI/UX .....                 | 33 |
| 4.     | Presentation of Platform Interfaces .....   | 33 |
| 4.1.   | Landing Page.....                           | 33 |
| 4.2.   | Registration Page.....                      | 35 |
| 4.3.   | Login Page.....                             | 35 |
| 4.4.   | Home Page .....                             | 36 |
| 4.4.1. | Upload New Desing .....                     | 36 |
| 4.4.2. | Presentation of Assets .....                | 38 |
| 4.5.   | Assets Page.....                            | 39 |
| 4.5.1. | Upload New Assets .....                     | 40 |
| 4.5.2. | Presentation of Assets .....                | 42 |
| 4.6.   | Jobs.....                                   | 43 |
| 4.6.1. | Add New Job.....                            | 44 |
| 4.6.2. | Presentation Job.....                       | 45 |
| 4.7.   | Services .....                              | 45 |
| 4.7.1. | Presentation Service .....                  | 46 |
| 4.8.   | Profile Page .....                          | 46 |

|                         |    |
|-------------------------|----|
| 4.9. Edit Profile ..... | 47 |
| 5. Conclusion.....      | 47 |
| Conclusion.....         | 48 |
| References .....        | 50 |

## List of Figures

|  |                                    |
|--|------------------------------------|
| Figure. 1: Web Application Architecture [4] .....                      | 9                                  |
| Figure. 2: Web Application Architecture Modern [4].....                | 10                                 |
| Figure. 3: Difference between a web application and a website [5] .... | <b>Erreur ! Signet non défini.</b> |
| Figure.4: Use Case Diagram .....                                       | 15                                 |
| Figure.5: Class Diagram.....   | 16                                 |
| Figure. 6: Sequence Diagram, User registration .....                   | 17                                 |
| Figure.7: Sequence Diagram, Login User .....                           | 18                                 |
| Figure. 8: Sequence Diagram, Upload new Design .....                   | 19                                 |
| Figure. 9 : Sequence Diagram, Upload New Assets .....                  | 20                                 |
| Figure. 10: Sequence Diagram, Add Job.....                             | 21                                 |
| Figure. 11: Sequence Diagram, Add Service .....                        | 22                                 |
| Figure. 12: JavaScript Logo .....                                      | 24                                 |
| Figure.13: Html Logo.....  | 25                                 |
| Figure.14: Css Logo .....  | 25                                 |
| Figure.15: Appwrite Logo.....  | 26                                 |
| Figure.16: Appwrite Console .....                                      | 26                                 |
| Figure.17: Appwrite Console Database .....                             | 27                                 |
| Figure.18: Appwrite Console Database Table .....                       | 27                                 |
| Figure.19: Next Js Logo .....  | 28                                 |
| Figure.20: React Js Logo.....  | 28                                 |
| Figure.21: Tailwind Css Logo.....                                      | 29                                 |
| Figure.22: GSAP Logo.....  | 29                                 |
| Figure.23: Visual Studio Code Logo.....                                | 29                                 |
| Figure.24: Postman Logo .....  | 30                                 |
| Figure.25: Spline Logo.....  | 30                                 |
| Figure.26: Figma Logo.....   | 30                                 |
| Figure.27: Lucidchart Logo.....  | 31                                 |
| Figure.28: Figma Tool, Inspiration .....                               | 31                                 |
| Figure.29: Figma Tool, Desing Platform Logo.....                       | 32                                 |
| Figure.30: Figma Tool, Design Standers .....                           | 32                                 |
| Figure.31: Figma Tool, Platform Design .....                           | 33                                 |
| Figure.32: Landing Page .....  | 34                                 |

|  |    |
|--|----|
| Figure.33: Registration Page .....             | 35 |
| Figure.34: Login page .....                    | 35 |
| Figure.35: Home Page.....                      | 36 |
| Figure.36: Upload New Desing Section.....      | 37 |
| Figure.37: Add Designs.....                    | 37 |
| Figure.38: Add Information of New Design ..... | 38 |
| Figure.40: Presentation of Design .....        | 39 |
| Figure.41: Assets Page .....                   | 40 |
| Figure.42: Upload New Assets Section.....      | 40 |
| Figure.43: Add Examples of Assets .....        | 41 |
| Figure.44: Add Assets Information .....        | 42 |
| Figure.45: Presentation Assets .....           | 43 |
| Figure.46: Jobs Offers .....                   | 44 |
| Figure.47: Add New Job Form.....               | 44 |
| Figure.48: Presentation Job Details.....       | 45 |
| Figure.49: Services.....                       | 45 |
| Figure.50: Presentation Service.....           | 46 |
| Figure.51: Profile Page.....                   | 46 |
| Figure.52: Edit Profile.....                   | 47 |

## List Of Tables

|  |    |
|--|----|
| Table 1: Difference between a web application and a website [5]..... | 11 |
|--|----|

## Introduction

With the evolution of the Internet from simple static pages to complex, multi-tasking platforms, especially following the Web 2.0 revolution, it has become an indispensable tool for communication and collaboration, enabling the exchange of experiences and cooperative task completion.

In this digitally enriched context, our project aims to develop and create a website for the Algerian creative community to advance the field of design. This platform will allow users to showcase their skills through visual clips, search for creative talents, and communicate with other design professionals. It aspires to be a multi-purpose hub for designers, enabling them to publish their work, sell their digital assets, and serve as a valuable source of inspiration and exposure to contemporary designs. Additionally, it provides an opportunity for designers to publish their portfolios.

To analyse, design, and complete this project, we have structured this thesis into three chapters as follows:

**Chapter One:** This chapter addresses the theoretical aspects of the subject, reviewing the general concepts related to the process of publishing creative designs, selling digital assets, and sharing works online.

**Chapter Two:** This chapter focuses on modelling the project and defining the overall conceptual structure of the application using the Unified Modelling Language (UML).

**Chapter Three:** This chapter presents the technical aspects of the project, reviewing the completed outputs with detailed explanations and evaluations.

Finally, we conclude this work with a summary and analysis of the results, along with suggestions for future development.

## **Chapter I. System Analysis**

### **1. Introduction**

This chapter delves into the study and analysis of the system, focusing on its fundamental concepts. It explains the processes and stages necessary to achieve the main objectives of software development, identifies the users and target segments of the program, and discusses the benefits and impact it has on society.

### **2. Presentation of the project**

The following section presents the main objectives, features, and contributions of our proposed project.

#### **2.1. Objectives of the project**

- Showcase and publish designs created by designers.
- Create a community where designers can share their experiences.
- Serves as source of inspiration for designers and provide them with the latest modern design ideas.
- Allow designers to publish educational publications, ideas, and tricks owned by design experts.
- Act as a valuable resource containing contemporary and trendy design ideas.

#### **2.2. Project Features**

- Enable users to upload, sell, or share designed digital assets for free.
- Allow designers and users to publish their job openings and portfolios.
- Provide a platform for designers to offer their services.
- Support both freelance and full-time work opportunities for designers.

#### **2.3. Contribution of the project**

- Contribute to the advancement and innovation in the field of design.
- Facilitate learning and professional development in design and creativity.

- Serve as a valuable source of visual inspiration and creative ideas.
- Foster the formation of a supportive community of designers.
- Promote the growth of a local design community within the country.
- Provide a local alternative for designers to share and promote their work.

### 3. Structure of the project

Here is a set of actors along with some technical requirements related to our proposal.

#### 3.1. Individuals

- **Platform team:** developers responsible for the website, its management, maintenance, and development.
- **Designers:** Platform users publish their designs and services and sell digital assets.

#### 3.2. Requirements

- Upload designs to multiple snapshots.
- The platform contains several categories, creative fields (logo design, web design, brand design, ...).
- You can add a description and explanation to the design post.
- Designs are also uploaded in the form of videos and animation.
- The designer can upload assets of several types (lines, vectors, templates).
- The designer can sell assets or for free.
- Digital assets can be in multiple formats (.PSD, .SVG, .PNG, ...).
- Digital assets are downloadable, without delay.
- The designer can upload examples of digital assets and create a description.
- The designer sets the price of the digital assets or makes them free to use.
- The designer reserves the right to decide whether the digital Assets are for personal or commercial purposes.
- Designers or users can publish jobs.
- Even companies can publish jobs that benefit designers.

- The designer has the right to determine its availability.
- Determine availability for self-employment, full-time work, or both.
- Ordinary users can hire designers.
- Ordinary users can hire designers.
- Digital assets are displayed with the price or tag for free.
- The designer can present his experience and years of work.

## **4. Community Building and Engagement in Design Platforms**

In the dynamic landscape of design, community plays a pivotal role in shaping creativity and innovation. This section delves into the significance of community in the design field, exploring how collaboration and engagement propel the industry forward.

### **4.1. Importance of Community in the Design Field**

The design field thrives on collaboration and community engagement. A strong community enables designers to share ideas, provide feedback, and inspire one another, fostering innovation and creativity.

### **4.2. Strategies for Building an Engaged Community**

Effective community building involves strategies such as regular interaction, user-generated content, events, and projects. Implementing these strategies can significantly enhance user engagement and retention.

### **4.3. Enhancing User Interaction on Our Platform**

Our platform enhances user interaction through features such as personalized portfolios, showcasing user projects, and providing a marketplace for selling digital assets. These elements foster a sense of community by allowing users to present their work, receive feedback through ratings, and engage in transactions that benefit both buyers and sellers. Additionally, our platform

supports the creation of user Portfolio that highlight their skills and achievements, making it easier for members to connect based on shared interests and expertise.

## **5. Monetization Strategies for Design Platforms**

Monetization lies at the heart of sustaining design platforms, ensuring their viability and growth. This section delves into the various monetization models available, emphasizing the importance of selecting the right approach to align with platform goals and audience needs.

### **5.1. Overview of Monetization Models**

Monetization is crucial for the sustainability of design platforms. Various models exist, including subscription services, premium memberships, transaction fees, advertising, and freemium models. Each approach has its advantages and potential drawbacks, making it essential to choose the right mix based on the target audience and platform objectives. Subscription services offer a steady revenue stream, while premium memberships provide additional features for a fee. Transaction fees generate income from sales conducted on the platform, and advertising can be lucrative if the platform attracts high traffic volumes.

### **5.2. Implementing E-Commerce on Design Platforms**

Integrating e-commerce functionalities is a powerful way to monetize design platforms. This includes setting up an online marketplace where designers can sell their digital assets such as templates, illustrations, fonts, and other creative resources. Secure payment gateways are essential to ensure smooth transactions. Additionally, implementing commission-based fees on each sale provides a sustainable revenue model while incentivizing designers to actively engage with the platform. The e-commerce component should be seamlessly integrated into the platform's UI/UX to ensure ease of use and high conversion rates.

### **5.3. Revenue Generation on Our Platform**

Our platform adopts a hybrid monetization model to ensure sustainable revenue generation while providing value to users. Additionally, we implement a commission-based model for transactions occurring within our marketplace, allowing designers to sell their digital assets and receive a majority share of the proceeds. This diversified approach ensures multiple revenue streams, enhancing the platform's financial stability and ability to invest in further development and user experience improvements.

## **6. Competitor Analysis in the Design Platform Market**

Analysing competitors is a crucial step in understanding the landscape and opportunities within the design platform market. This section provides insights into key competitors, their strengths, weaknesses, and how our platform aims to differentiate itself.

## **6.1. Examples of Real-World Competitors**

In the realm of design platforms, several well-established competitors have set high standards for community engagement, portfolio presentation, and monetization. Two prominent examples are Behance and Dribbble.

### **6.1.1. Behance**

Behance, a platform owned by Adobe, allows designers to showcase their work, gain exposure, and connect with potential clients and collaborators. It integrates seamlessly with Adobe's suite of creative tools, making it a convenient choice for users within the Adobe ecosystem. Behance emphasizes high-quality visuals and detailed project presentations, attracting a professional audience.

### **6.1.2. Dribbble**

Dribbble is another leading platform that serves as a community for designers to share small screenshots of their work, known as "shots." It focuses on fostering interaction and feedback among designers through likes and comments. Dribbble also offers job listings, making it a valuable resource for freelance and full-time design opportunities.

## **6.2. Problems and Challenges**

Despite their popularity, platforms like Behance and Dribbble face several challenges :

- **Oversaturation:** With millions of users, it can be difficult for new or less well-known designers to gain visibility and stand out from the crowd.
- **Limited Monetization Options:** While both platforms offer ways to showcase and promote work, they provide limited direct monetization options for designers beyond exposure and job listings.
- **User Interface Complexity:** Some users find the interfaces of these platforms to be complex and not user-friendly, especially for those who are new to the design field or less tech-savvy.

## **6.3. Value of Our Platform**

Our platform seeks to address these issues and provide enhanced value through several key features:

- **Direct Monetization Opportunities:** We offer robust E-commerce capabilities, allowing designers to sell their digital assets directly on the platform.
- **User-Friendly Interface:** With a focus on simplicity and ease of use, our platform's user interface is designed to be intuitive and accessible, reducing the learning curve for new users and enhancing overall user satisfaction.
- our platform is designed to cover all categories of users, including people who are willing to invest money in buy assets digital, paid services,

## **7. Advanced Web Technologies and Development**

This section investigates into the field of advanced web technologies and development practices, exploring the foundational concepts and architectural frameworks that support modern web applications.

### **7.1. What is a Web Application?**

In simple terms, A Web App is like a special computer program that you can use on the internet without having to download it onto your computer. It is different from regular apps that you need to install. You can open your web browser (Chrome and Safari) and start using the app right there online and accessed from a remote server via web browser. Web applications are made for interaction, allowing users to send and consume data between the browser and the web server. This interaction can be as simple as logging in to an account, or as complex as making a payment with your credit card. [1][2]

### **7.2. Architectural Foundations of Web Applications**

This section explores the foundational principles and components essential for designing and developing robust, scalable, and efficient Web application

### **7.3. Components of Web Applications Architecture**

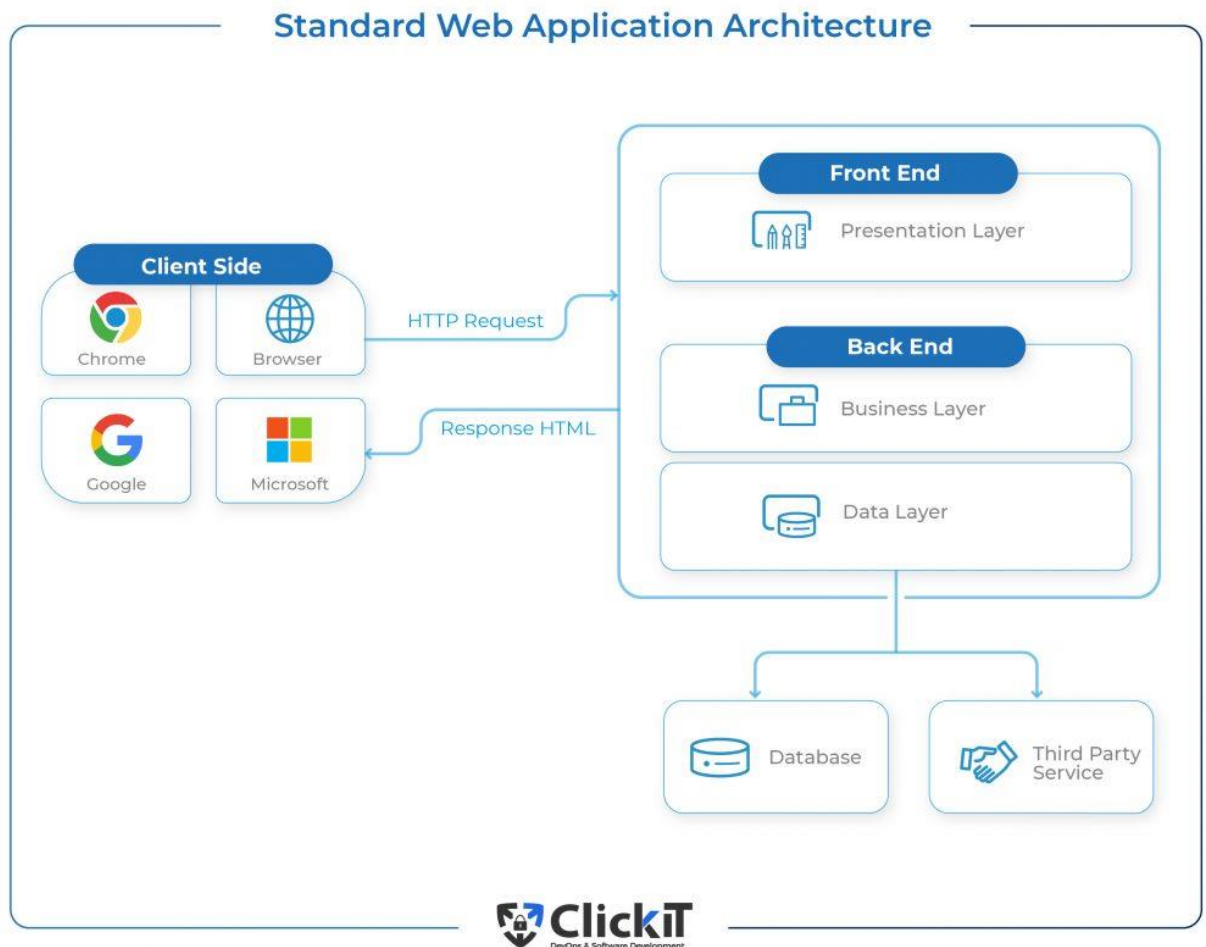
Web application architecture Is the structural Framework That guides the design and development of a web application. It involves a set of Principles and best practices That determine how different parts of the application work together to deliver its intended functions.

In simpler terms, it's like the blueprint that organizes the frontend, (what users see), backend, (server and business logic), and database, ensuring they interact smoothly.

The main objective is to create a unified and scalable system that meets the application's functional needs while also considering factors like performance, scalability, maintainability, and security.[3]

The primary objectives of web application architecture revolve around ensuring the efficiency, reliability, and scalability of a digital system.[3]

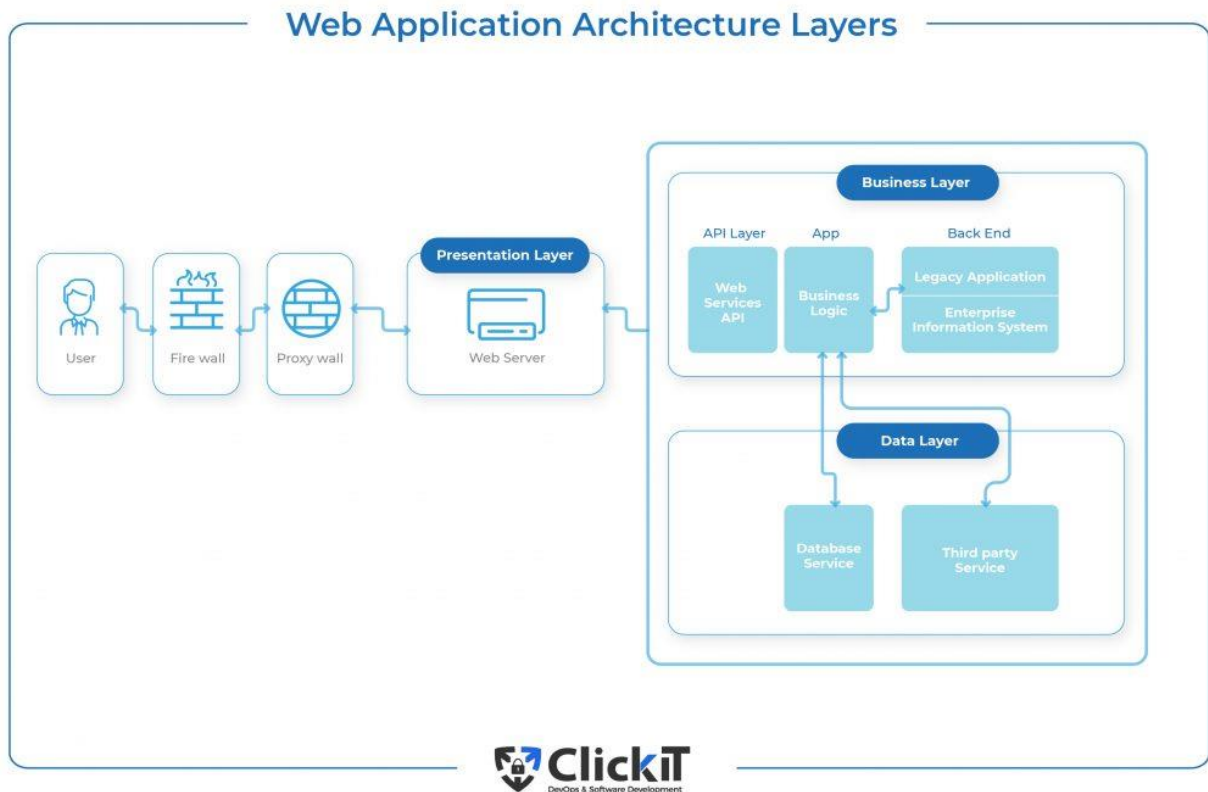
- It aims to provide a seamless and intuitive user experience by organizing the frontend components, such as interfaces and interactions, in a user-friendly manner.[3]
- The architecture addresses backend functionality, focusing on creating a robust and efficient server that handles app logic, processes data, and communicates effectively with the frontend.[3]
- Scalability is another crucial goal, aiming to design a system that can handle growth in users or data volume without compromising performance.[3]
- Security considerations are woven into the web application architecture to safeguard sensitive information and protect against potential threats.[3]



**Figure. 1: Web Application Architecture [4]**

### 7.3.1. Modern Web Application Architecture

Building a layered modern web application architecture diagram helps you identify each component's role and easily make changes to the corresponding layer without affecting the overall application. It enables you to write, debug, manage, and re-use code easily ensures scalability, maintainability, and security.[4]



**Figure. 2: Web Application Architecture Modern [4]**

#### 7.4. The difference between a web application and a website?

While both web applications and websites are accessed through web browsers, they serve different purposes.

| Website  | Web Application  |
|--|--|
| Website basically contains static content.                                   | Web application is designed for interaction with users.                                      |
| The user of website only can read the content of website but not manipulate. | The user of web application can read the content of web application and manipulate the data. |
| It's easy to develop since it requires only the basic web technologies.      | The web application site should be precompiled before deployment                             |

|  |  |
|--|--|
| The function of website is simple.                       | The function of a web application is quite complex.                              |
| Web site is not interactive for users.                   | Web applications are interactive for users.                                      |
| The browser capabilities involved with web site is high. | The browser capabilities involved with a web application is high.                |
| Integration is simpler for web site.                     | Integration is complex for web application because of its complex functionality. |
| In web site authentication is not necessary.             | Web application mostly requires authentication                                   |
| Example: Breaking News, Aktu website, etc.               | Example: Amazon, Facebook, etc.  |

**Table 1: Difference between a web application and a website [5]**

## 7.5. Web Application Models

This section explores various models of web applications, each catering to specific functionalities and user experiences. From static to dynamic, e-commerce to content management, and single page to progressive web applications.

### 7.5.1. Static Web Applications

Static web applications do not indulge in any server-side generation of HTML pages. Rather, these apps run on the client side within the browser and do not interact directly with the users.[6]

### 7.5.2. Dynamic Web Applications

This type of web app is designed to deliver a dynamic user experience by using server-side rendering of HTML pages. [6]

### 7.5.3. E-Commerce Web Applications

E-Commerce web app development companies are helping businesses to create robust web apps that meet the extensive demands of the evolving eCommerce market. [6]

### 7.5.4. Content Management System

CMS web applications have transformed how businesses would create and manage content. The process has become faster, simpler, and more intuitive. [6]

### **7.5.5. Single-Page Applications**

Single-page applications have gained immense popularity for their fast page load concept, where the browser loads all the pages all at once. It does not require reloading pages to view or access new information. [6]

### **7.5.6. Multi-Page Applications**

Unlike single-page applications, this type of web application includes multiple pages relating to multiple sets of information. The browser reloads the website to access different pages existing on the web application. [6]

### **7.5.7. Progressive Web Applications**

PWAs exhibit a native app-like experience despite being built in the web architecture. These apps are installable and provide an impeccable user experience. [6]

## **7.6. The Web Application Development Lifecycle**

Creating a web application is a highly organized, comprehensive process where the main goal is to develop a product that meets your business needs. In most cases, it consists of several stages.[7]

When it comes to creating the best web applications, the dedicated web developers start with gathering the project goals, requirements, and more. Then move on to design and development, after which comes the testing and deployment.[8]

### **7.6.1. Planning and Discovery**

- Objective Definition: Clearly define the purpose and goals of the web application.
- User Persona Development: Understand the target audience to tailor the user experience.[9]

### **7.6.2. Design**

- Wireframing and Prototyping: Create visual representations of the application's layout and functionalities.
- Design the user interface: interface (UI) and user experience (UX) of the web application. [9]

### 7.6.3. Development

- Design Database: Design and develop a database schema based on the application requirements.[9]
- Front-End Development: Front-end development is the art and science of crafting the visual and interactive elements of a website or web application. .[9]
- Back-End Development: Build the server, database, and application logic using frameworks like Node.js, Django, or Ruby on Rails. .[9]
- Integration: Connect the front-end and back-end components to ensure seamless communication. .[9]

### 7.6.4. Testing

- Unit Testing: Verify individual components for functionality. .[9]
- Integration Testing: Ensure smooth interaction between integrated components. .[9]
- User Acceptance Testing (UAT): Validate the application against user expectations. .[9]

### 7.6.5. Deployment

- Hosting: Choose a suitable hosting platform like AWS, Heroku, or Azure. .[9]
- Domain Configuration: Set up the domain and configure DNS settings. .[9]

### 7.6.6. Maintenance and Updates

- Monitoring: Implement tools for continuous monitoring of application performance. .[9]
- Bug Fixes and Updates: Regularly update and patch the application to enhance security and functionality. .[9]

## 8. Tailoring our Platform for the Algerian Market

This section introduces the strategic adaptations and initiatives undertaken to customize our platform for the unique dynamics of the Algerian market. We explore how our platform aims to empower and elevate the local design community.

### 8.1. Addressing the Unique Needs of Algerian Designers

The local Algerian market currently lacks platforms dedicated to designers and the design community. Our platform aims to fill this gap by providing a space where Algerian designers can

publish, showcase, and sell their work. This platform will serve as a central hub for design talent in Algeria, promoting local creativity and innovation.

## **8.2. Empowering Algerian Designers**

Our platform empowers Algerian designers by offering tools and features that cater to their specific needs. From building professional portfolios to finding freelance jobs. By highlighting local talent, we aim to foster a vibrant design community within Algeria.

## **8.3. Bridging the Gap in Digital Resources**

In a market devoid of similar platforms, our website offers a much-needed resource for Algerian designers. This knowledge-sharing aspect is essential for the development of design skills and expertise in the region.

## **9. Conclusion**

In conclusion, the development and launch of our web platform for designers in Algeria represent a significant step forward in supporting and nurturing local creative talent. The platform addresses a critical gap in the Algerian market, providing a dedicated space for designers to showcase their work, connect with potential clients, and access valuable resources. By leveraging advanced web technologies and adhering to best practices in web application development, we have created a robust and user-friendly environment that meets the unique needs of the Algerian design community.

## Chapter II. Design Analysis

### 1. Introduction

This chapter covers the basic components of software system design using architectural diagrams and database design. These plans give developers a better perspective and understanding of the platform for effective targeting.

### 2. What is Unified Modeling Language (UML)

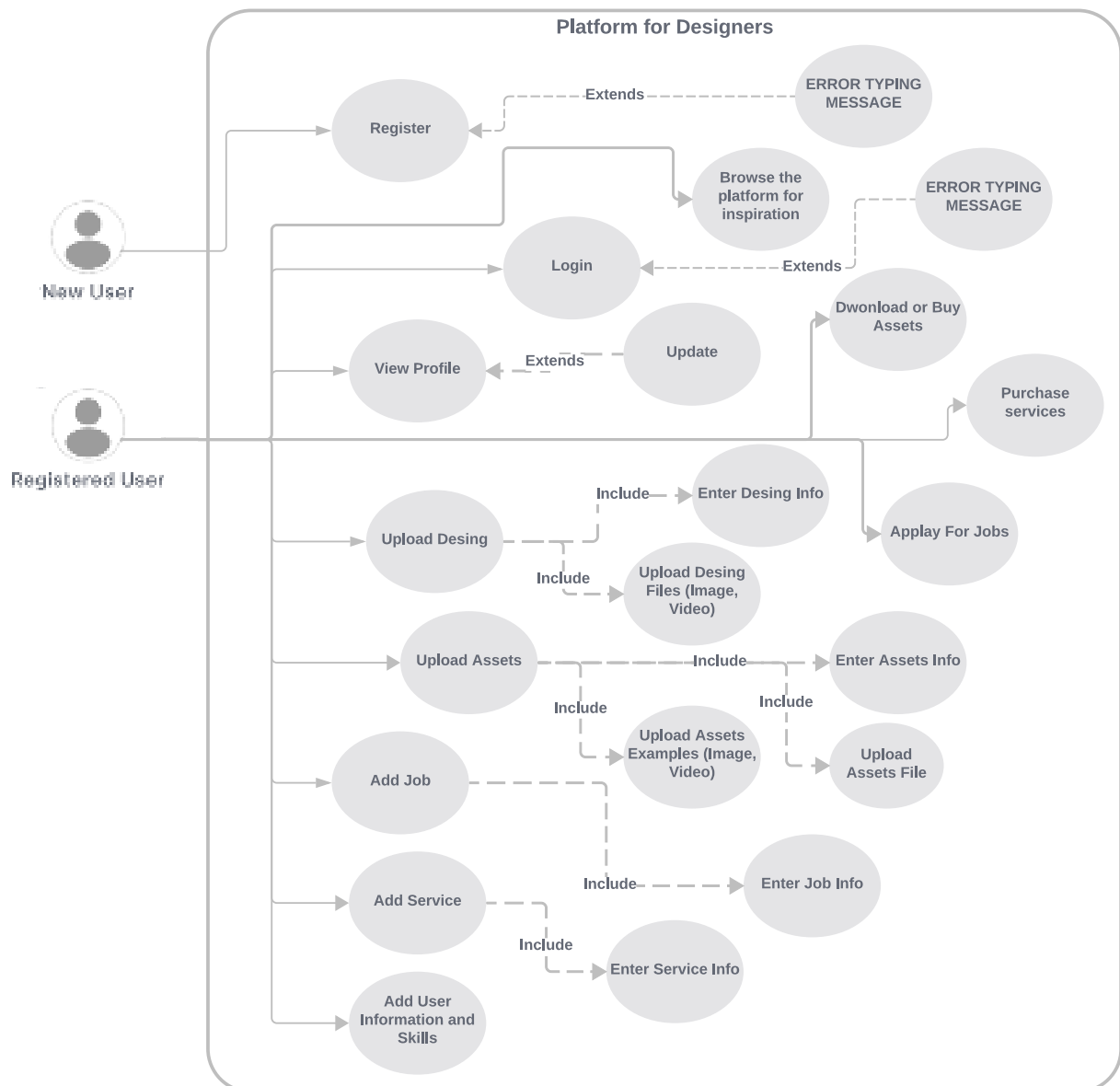
The Unified Modeling Language (UML) was created to forge a common, semantically and syntactically rich visual modeling language for the architecture, design, and implementation of complex software systems both structurally and behaviorally. UML has applications beyond software development, such as process flow in manufacturing.[10]

UML is not a programming language but there are tools that can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design.[10]

### 3. Diagrams

#### 3.1. Use Case Diagram

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss. [11]. figure (4).



**Figure.4: Use Case Diagram**

### 3.2. Class Diagram

A class diagram is a static UML diagram type that describes the structure of a system by showing its classes, attributes, operations, and relationships among objects.

Class diagrams are used for visualizing, describing, and documenting different aspects of a system so that you can construct executable code for software applications. [12]

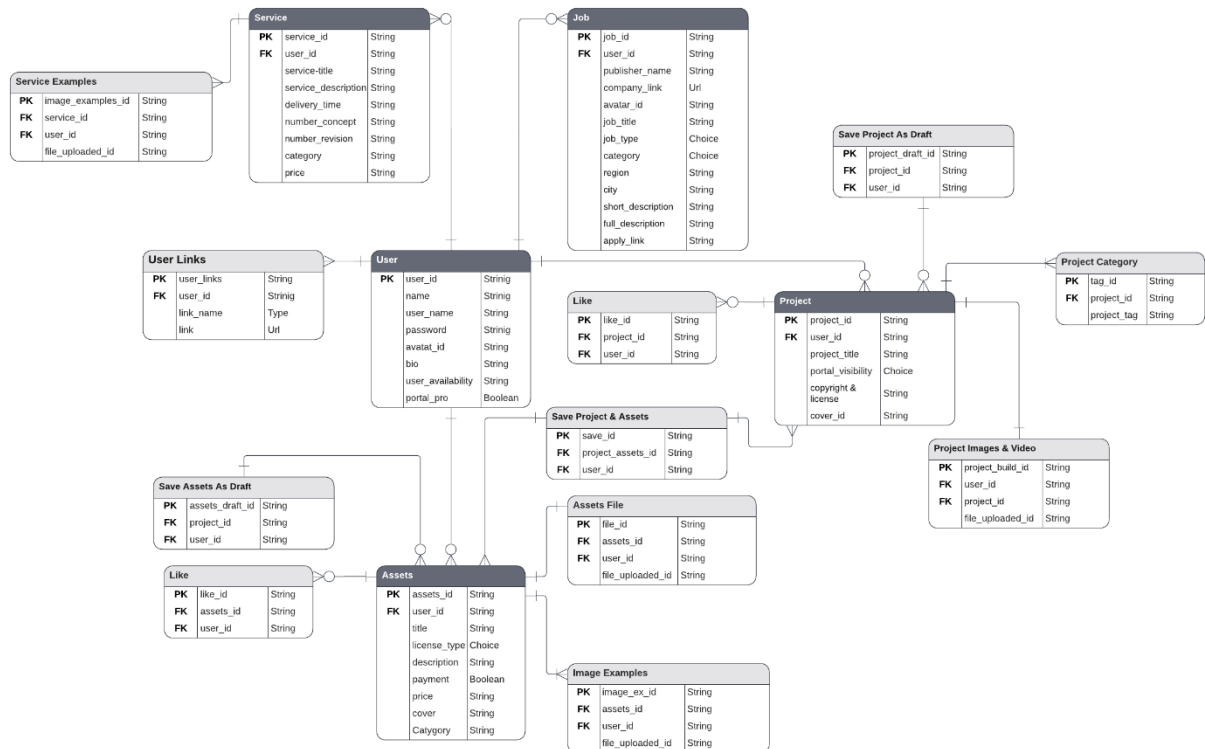


Figure.5: Class Diagram

### 3.3. Sequence Diagram

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.[13]

#### 3.3.1. Create an account

In this diagram below figure [6], it represents how the user registers on the platform and the actions that follow from that and a responses in case errors occur.

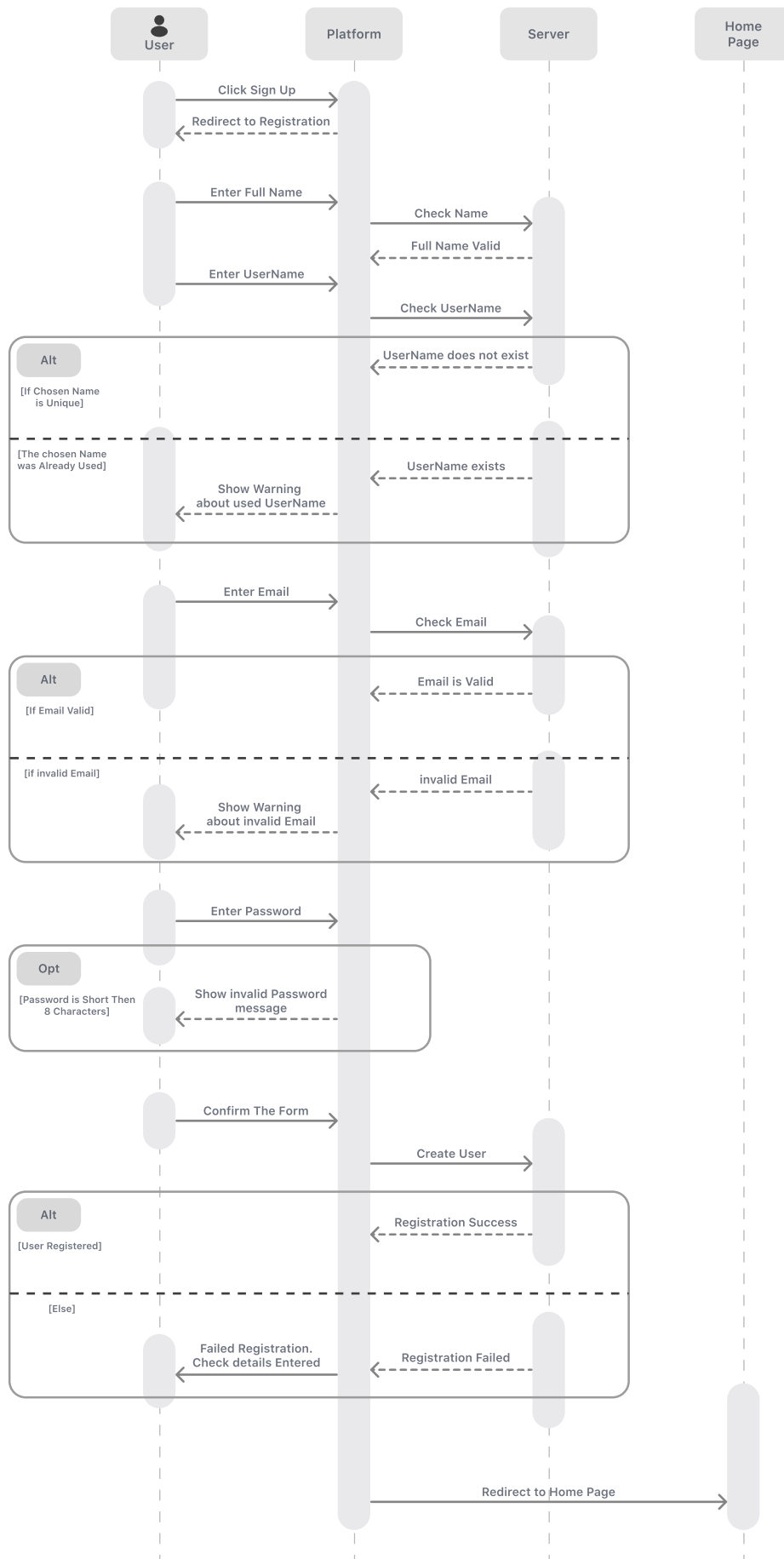


Figure. 6: Sequence Diagram, User registration

### 3.3.2. Login User Diagram

In this diagram below Figure [7], it represents how to log in to the platform and the actions that result from that and a responses in case errors occur.

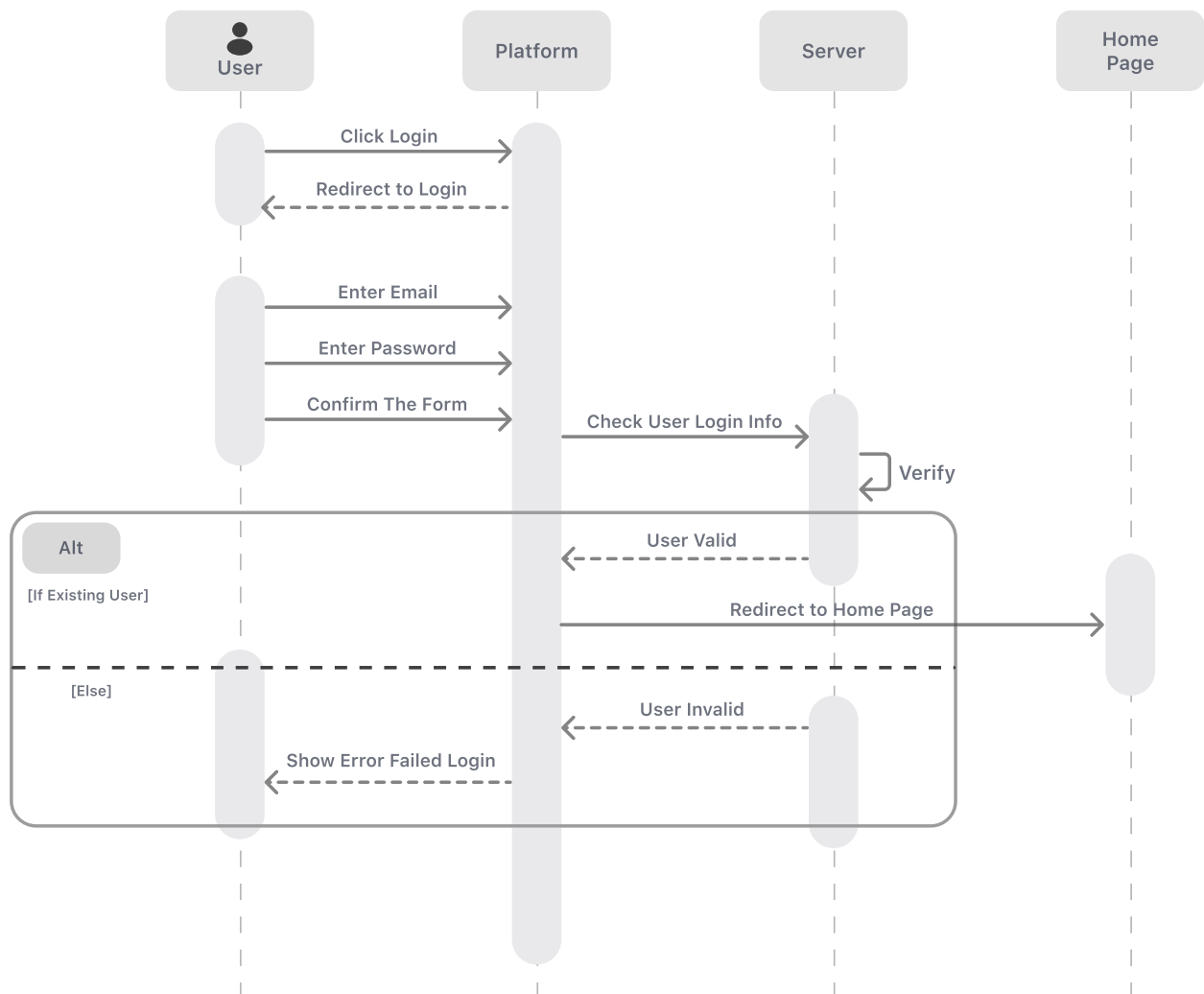


Figure.7: Sequence Diagram, Login User

### 3.3.3. Upload User New Desing Diagram

In this diagram below Figure [8], it represents how the user uploads a new design and the actions that follow in terms of image conditions and responses in case errors occur.

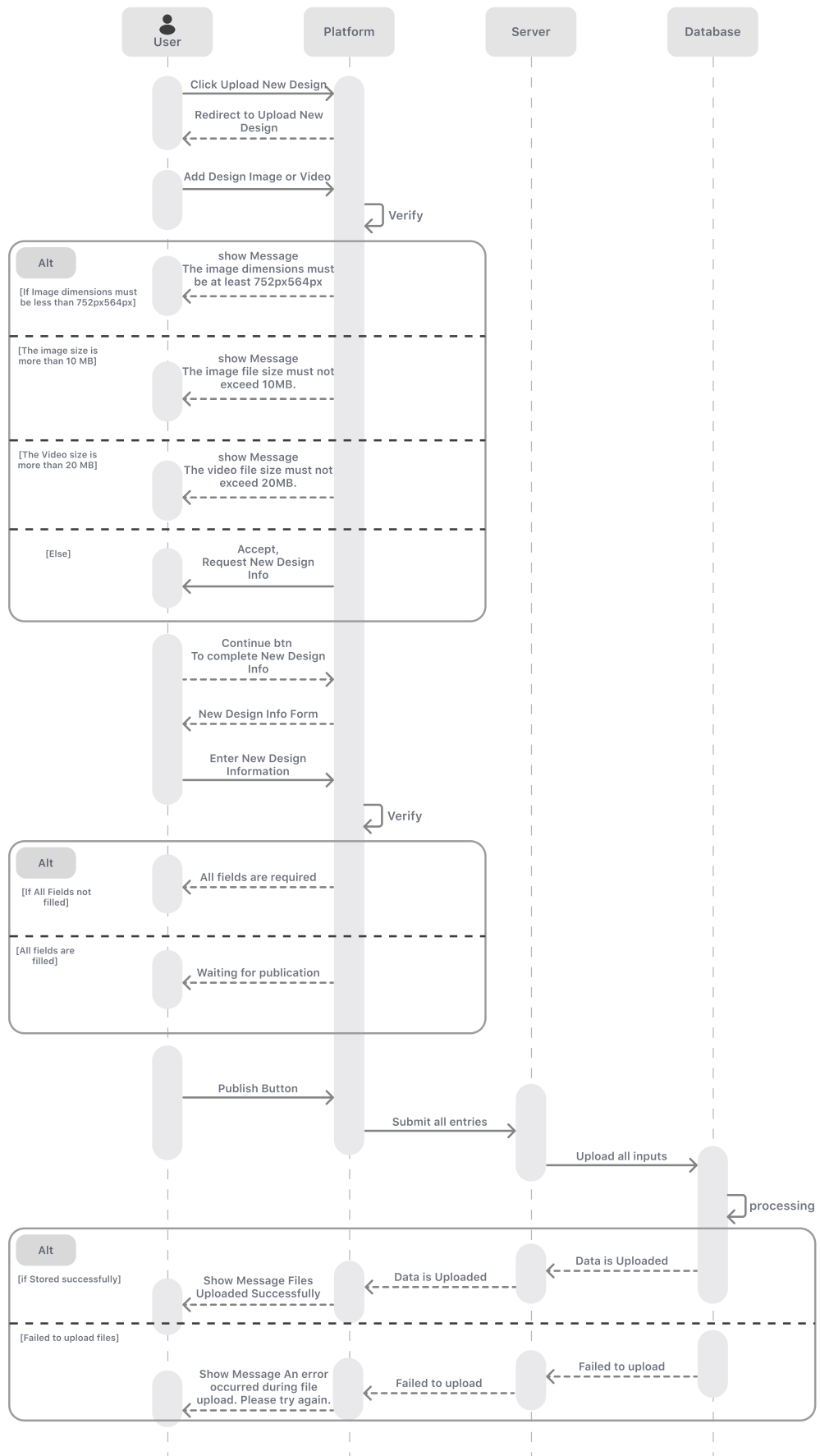


Figure. 8: Sequence Diagram, Upload new Design

### 3.3.4. Upload New Assets Diagram

In this diagram below Figure [9], it represents how the user uploads a new asset and the actions that follow in terms of image conditions, adding price information and responses in case errors occur.

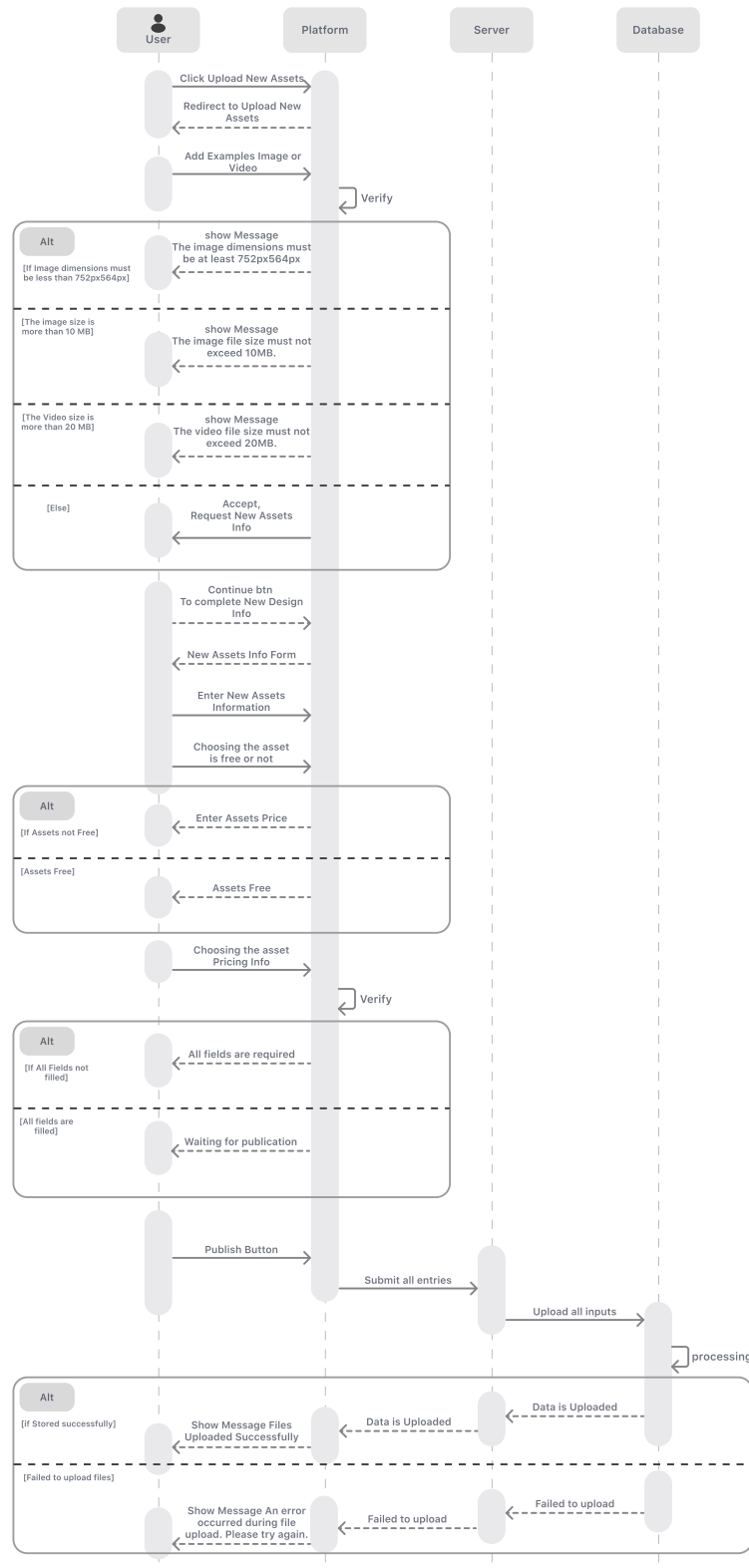


Figure. 9 : Sequence Diagram, Upload New Assets

### 3.3.5. Add Job Diagram

In this diagram below Figure [10], it represents how the user uploads a function, the actions that follow, and a responses in case errors occur.

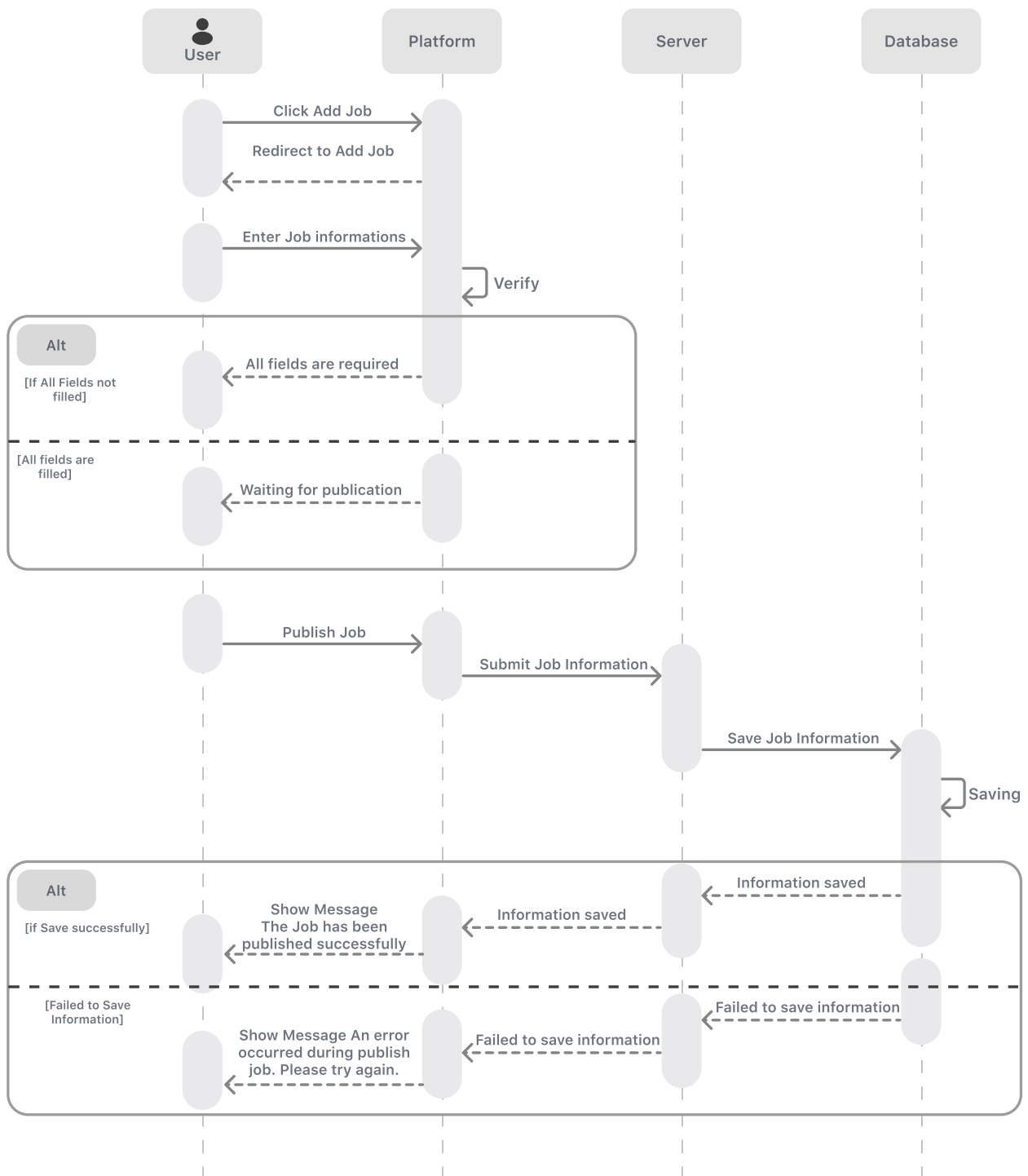


Figure. 10: Sequence Diagram, Add Job

### 3.3.6. Add Service Diagram

In this diagram below Figure [11], it represents how the user uploads a service and the actions that result from that and a display in case errors occur.

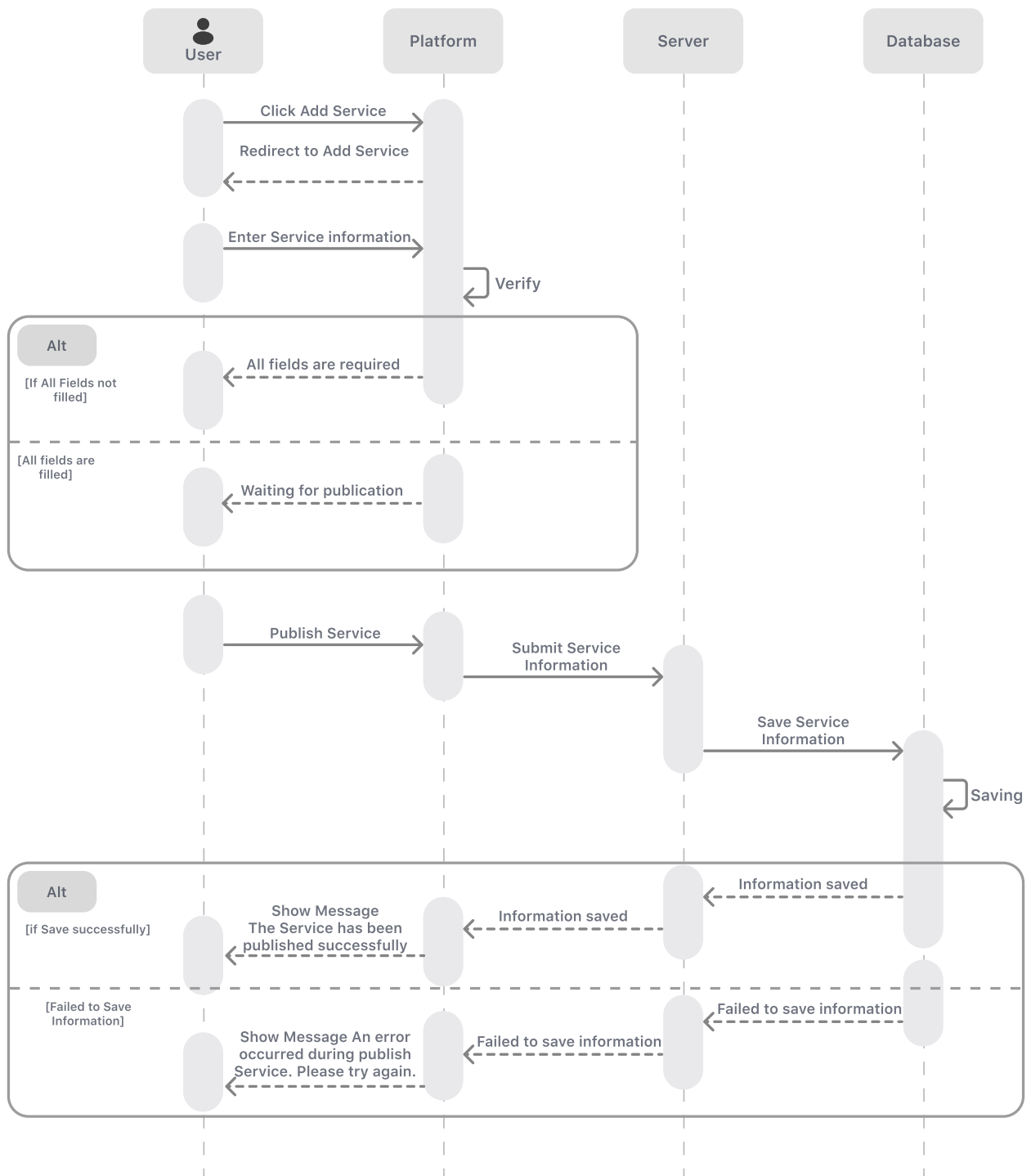


Figure. 11: Sequence Diagram, Add Service

## **4. Conclusion**

In conclusion, the use of visual modeling language is one of the languages described in this chapter, UML. It helps in planning and designing complex systems by creating diagrams that show how the different parts of the system work together. The use of UML by developers effectively helps to avoid its risks, which It makes their projects more success.

## Chapter III. Implementation

### 1. Introduction

In this chapter, we will discuss the implementation part and present all aspects. This is part of the development and the most important technologies contributing to the creation of the platform, with snapshots from inside the platform.

### 2. Integrated development environment

#### 2.1. Technologies and Programming Languages

##### 2.1.1. JavaScript

JavaScript is a scripting language that enables you to create dynamically updating content, that allows you to implement complex features on web pages control multimedia, animate images, and pretty much everything else.[14]



Figure. 12: JavaScript Logo

##### 2.1.2. Html

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).[15]



**Figure.13: Html Logo**

### **2.1.3. Css**

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.[16]



**Figure.14: Css Logo**

## **2.2. Database Management System**

### **2.2.1. Appwrite**

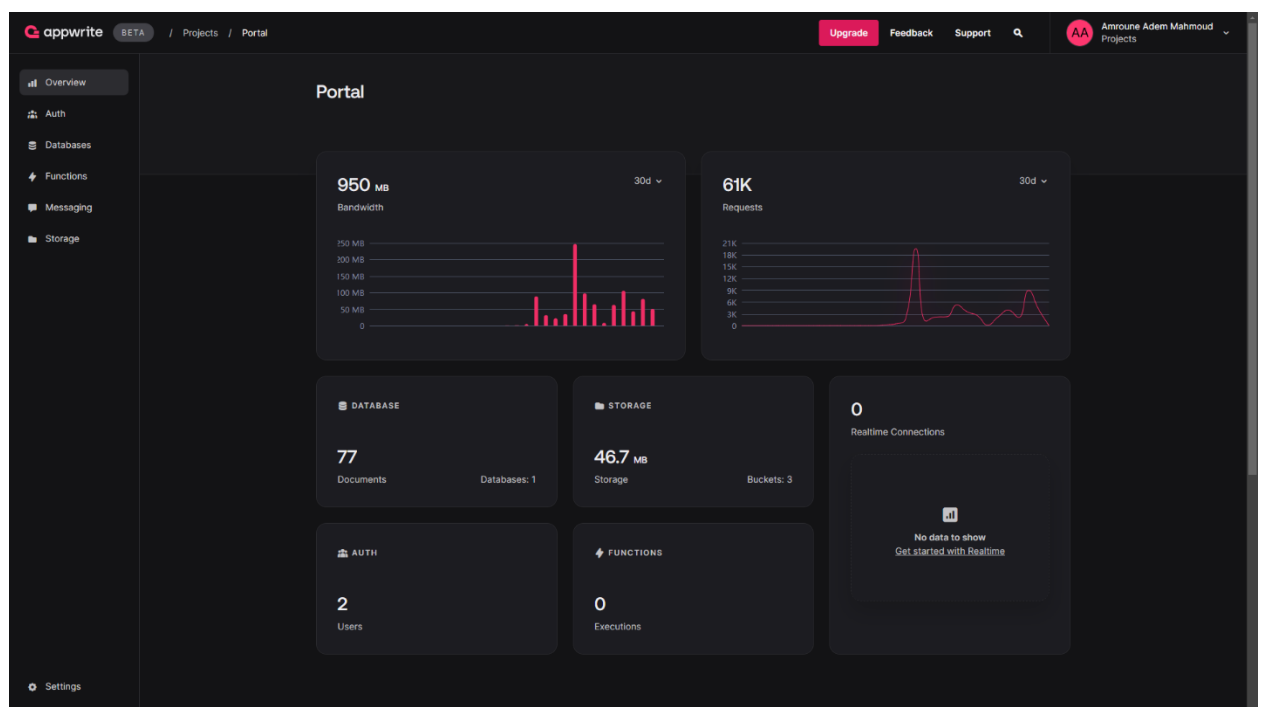
Appwrite is an end-to-end backend server for Web, Mobile, Native, or Backend apps packaged as a set of Docker microservices. Appwrite abstracts the complexity and repetitiveness required to build a modern backend API from scratch and allows you to build secure apps faster.

Using Appwrite, you can easily integrate your app with user authentication and multiple sign-in methods, a database for storing and querying users and team data, storage and file management, image manipulation, Cloud Functions, and more services.[17]

**Figure.15: Appwrite Logo**

### 2.2.2. Visuals at Appwrite

Appwrite Console is the Graphical User Interface that developers interact with when accessing their Appwrite instance in the web browser. Designed to be accessible to developers of all skill levels. Complex workflows will provide more guidance for newer developers and promote good technical decisions.[18]

**Figure.16: Appwrite Console**

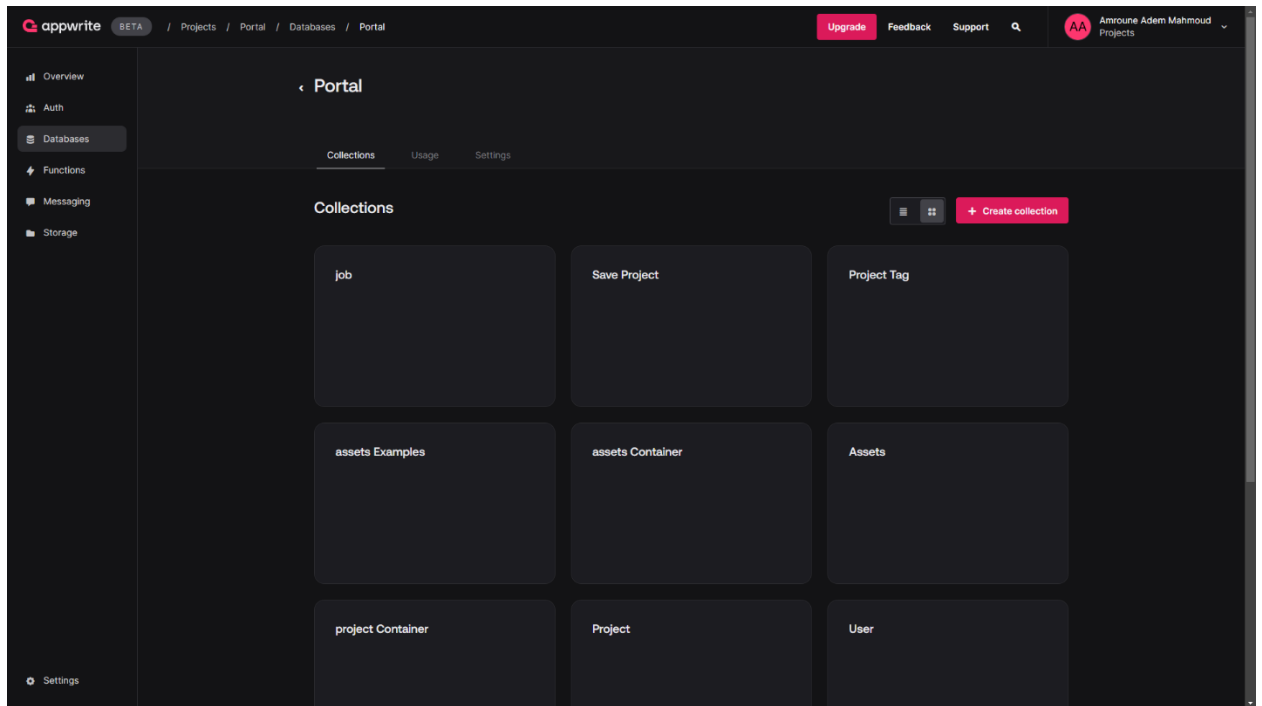


Figure.17: Appwrite Console Database

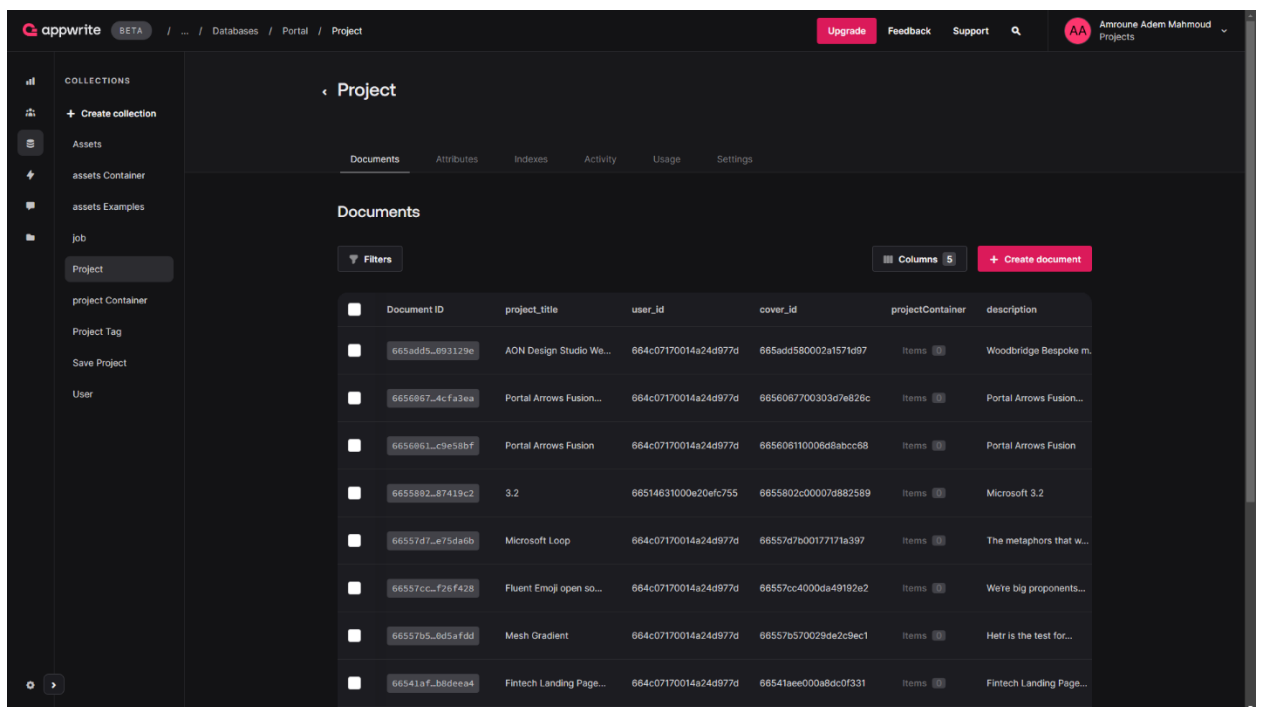


Figure.18: Appwrite Console Database Table

## 2.3. Frameworks and Libraries

### 2.3.1. Next Js

Next.js is a front-end framework that builds on the capabilities of React, a versatile JavaScript library used for crafting engaging user interfaces. What sets Next.js apart is its ability to simplify complex aspects of web development by providing a comprehensive suite of tools and features.

Another distinguishing attribute of Next.js is static generation. This feature pre-generates files during build time, allowing them to be served ahead of time without needing to wait for data fetching.[19]

Next.js stands out by seamlessly blending frontend and backend capabilities. Its dual nature becomes evident in scenarios where traditional distinctions between frontend and backend become less defined.[20]



Figure.19: Next Js Logo

### 2.3.2. React Js

React is a framework that employs Webpack to automatically compile React, JSX, and ES6 code while handling CSS file prefixes. React is a JavaScript-based UI development library. Although React is a library rather than a language, it is widely used in web development. The library first appeared in May 2013 and is now one of the most commonly used frontend libraries for web development.[21]



Figure.20: React Js Logo

### 2.3.3. Tailwind Css

Tailwind CSS is described as ‘A Utility-first CSS Framework that packed with classes.’ This means that Tailwind is a CSS framework that prioritizes the use of utility classes.[22]



Figure.21: Tailwind Css Logo

### 2.3.4. GSAP (GreenSock Animation Platform, is a powerful)

GSAP, or GreenSock Animation Platform, is a powerful JavaScript animation library that is used by web developers and designers to create interactive and engaging websites. GSAP is known for its versatility, speed, and ease of use, making it a popular choice among developers worldwide.[23]



Figure.22: GSAP Logo

## 2.4. Development Tools

### 2.4.1. Visual Studio Code

Visual Studio Code is one of the popular text editors used by professionals and recommended to new coders. Being one of Microsoft’s renowned software, VS code is free to use, open-source, and compatible with Windows, Linux, and macOS.[24]



Figure.23: Visual Studio Code Logo

### 2.4.2. Postman

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.[25]



Figure.24: Postman Logo

### 2.4.3. Spline

Spline is an easy to use 3D design tool geared for any designer regardless of their 3D experience. It's simpler to learn than full featured 3D apps—such as Cinema 4D or Blender—because it doesn't bog you down with loads and loads of settings and features. Best of all, it is browser-based and FREE,[26]



Figure.25: Spline Logo

### 2.4.4. Figma

Figma is a powerful design tool that helps you to create anything: websites, applications, logos, and much more.[27]



Figure.26: Figma Logo

### 2.4.5. Lucidchart

Lucidchart is a web-based diagramming application that allows users to visually collaborate on drawing, revising and sharing charts and diagrams, and improve processes, systems, and organizational structures.[28]



Figure.27: Lucidchart Logo

## 3. Desing Ui/Ux

In this section, we present the design for all parts of the platform's user interface, including setting the times and Forms of inputs and components using the Figma tool.

### 3.1. Inspiration

In this snapshot at the below of Figure [28], it represents the stage of inspiration and taking a look at competing sites

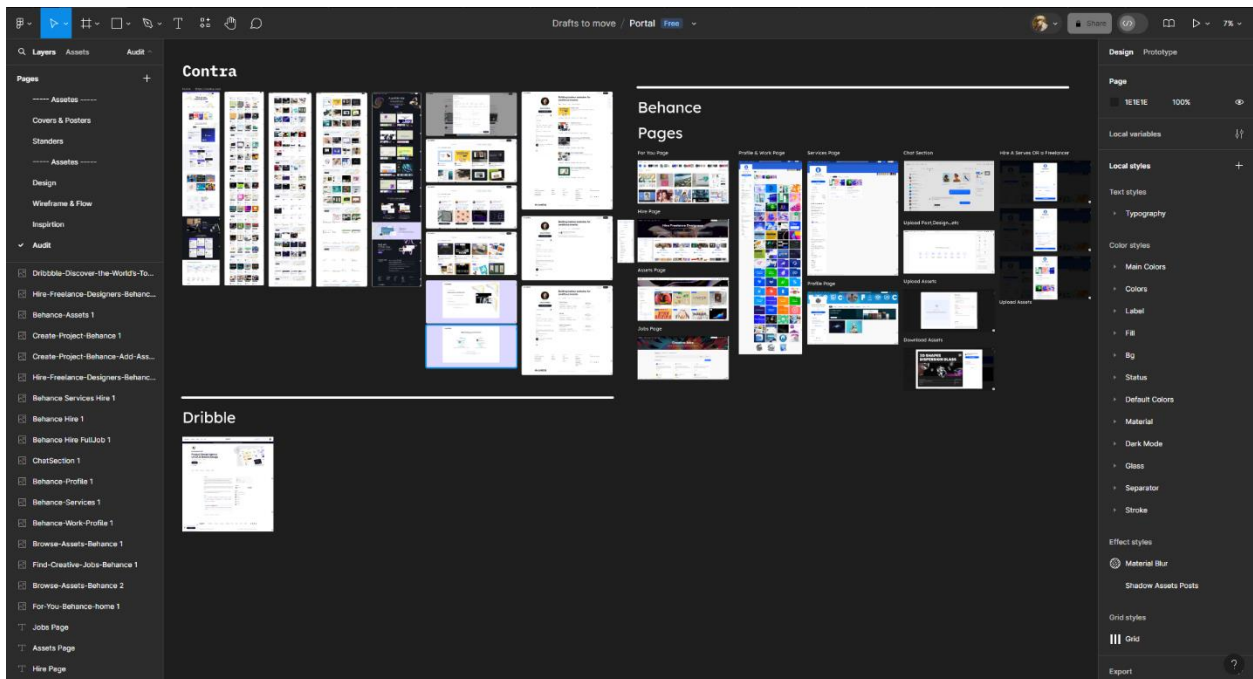


Figure.28: Figma Tool, Inspiration

### 3.2. Design the platform logo and identity

In this snapshot at the below of Figure [29], it represents the design of the platform's logo and identity

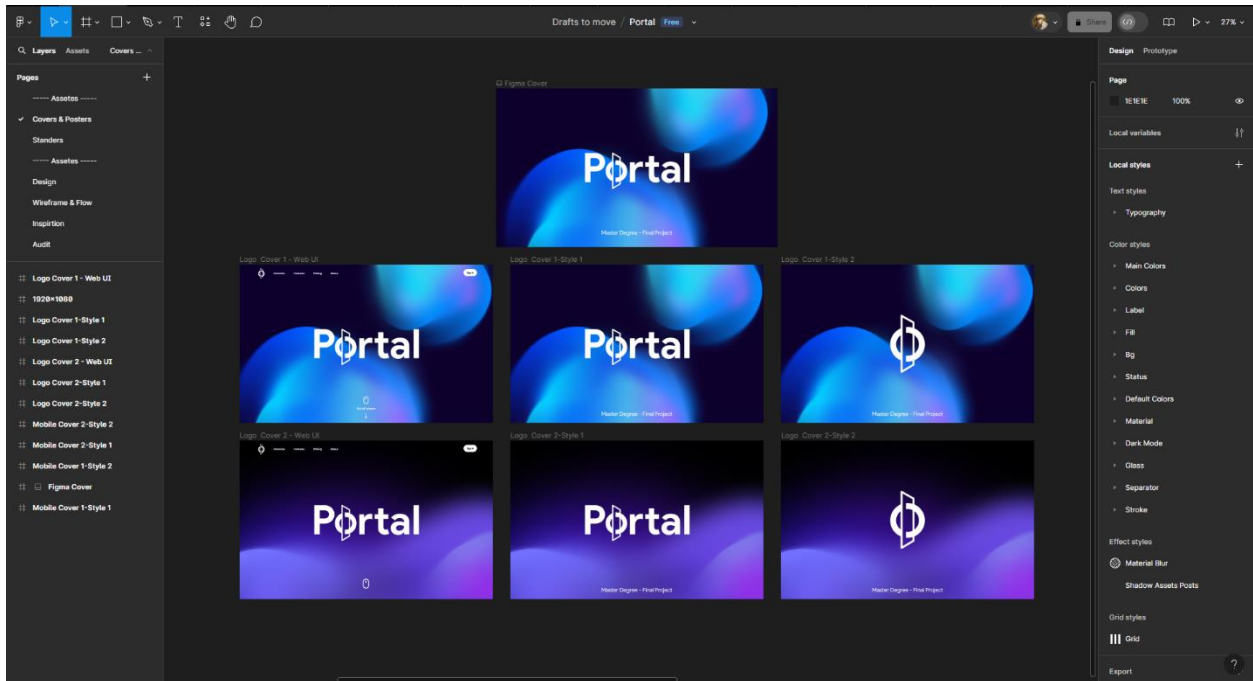


Figure.29: Figma Tool, Desing Platform Logo

### 3.3. Design Standers

In this snapshot at the below of Figure [30], it represents of designing standards, input shapes, colors, and components.

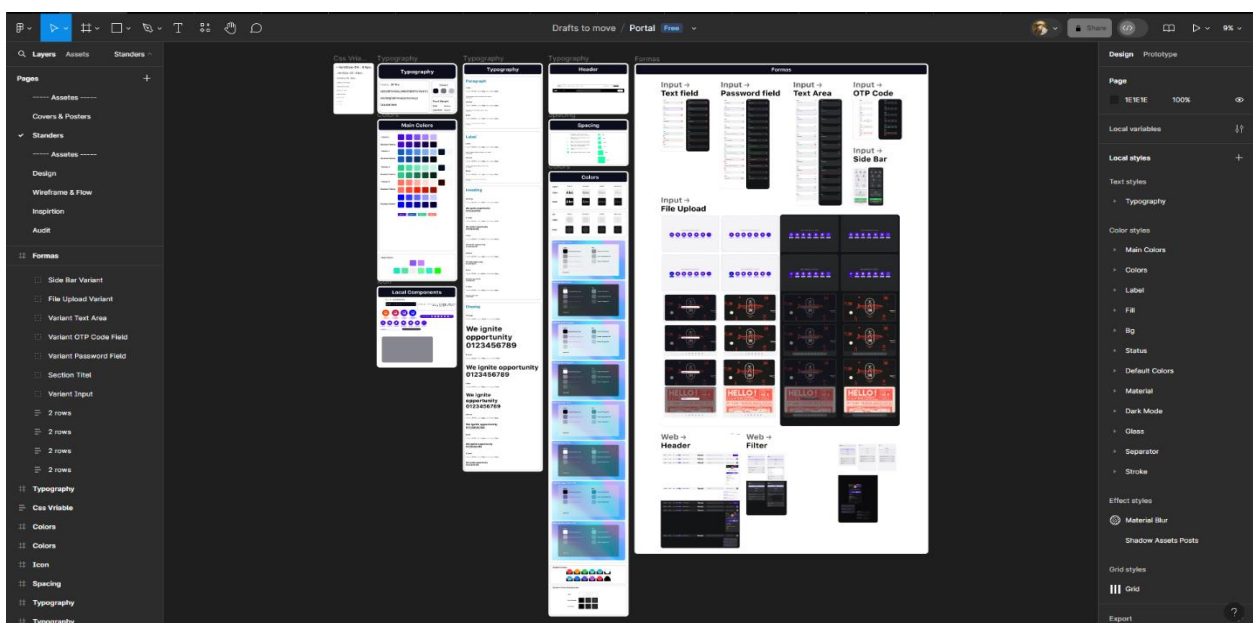


Figure.30: Figma Tool, Design Standers

### 3.4. Platform Desing UI/UX

In this snapshot at the bottom of Figure [31], it represents the design phase of all user pages and interfaces and what the platform looks like

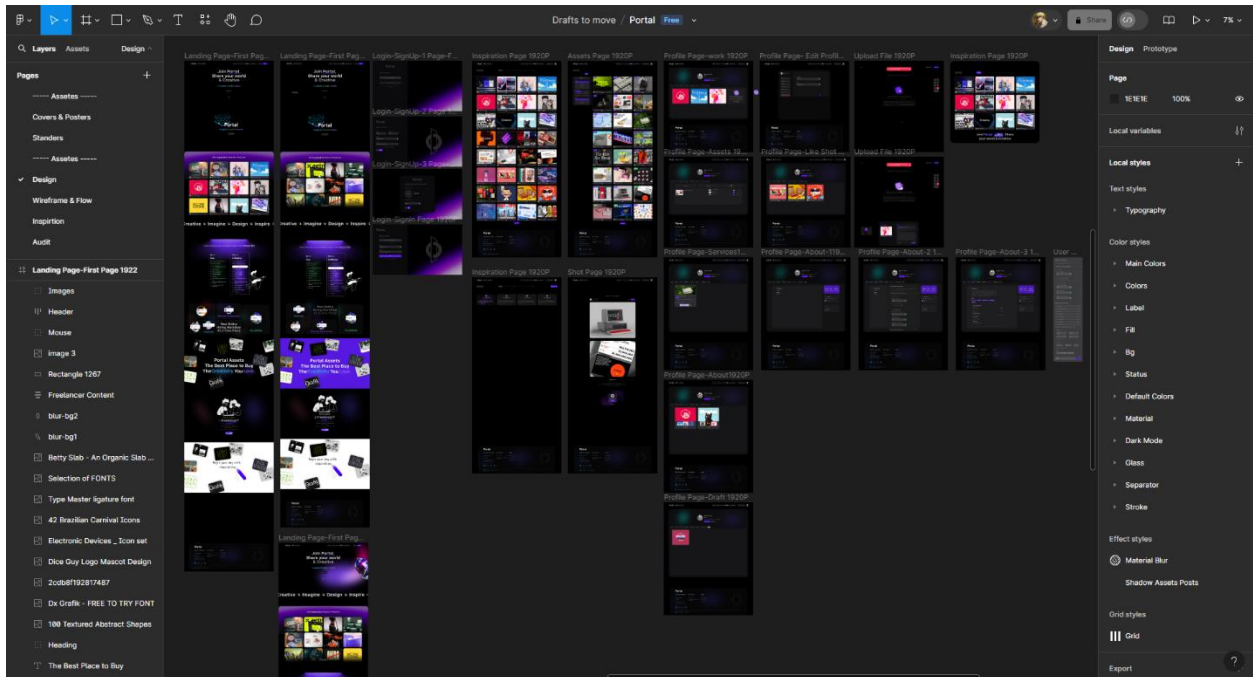


Figure.31: Figma Tool, Platform Design

## 4. Presentation of Platform Interfaces

### 4.1. Landing Page

The first page is an introduction page about the platform and a presentation of the features Figure (32).

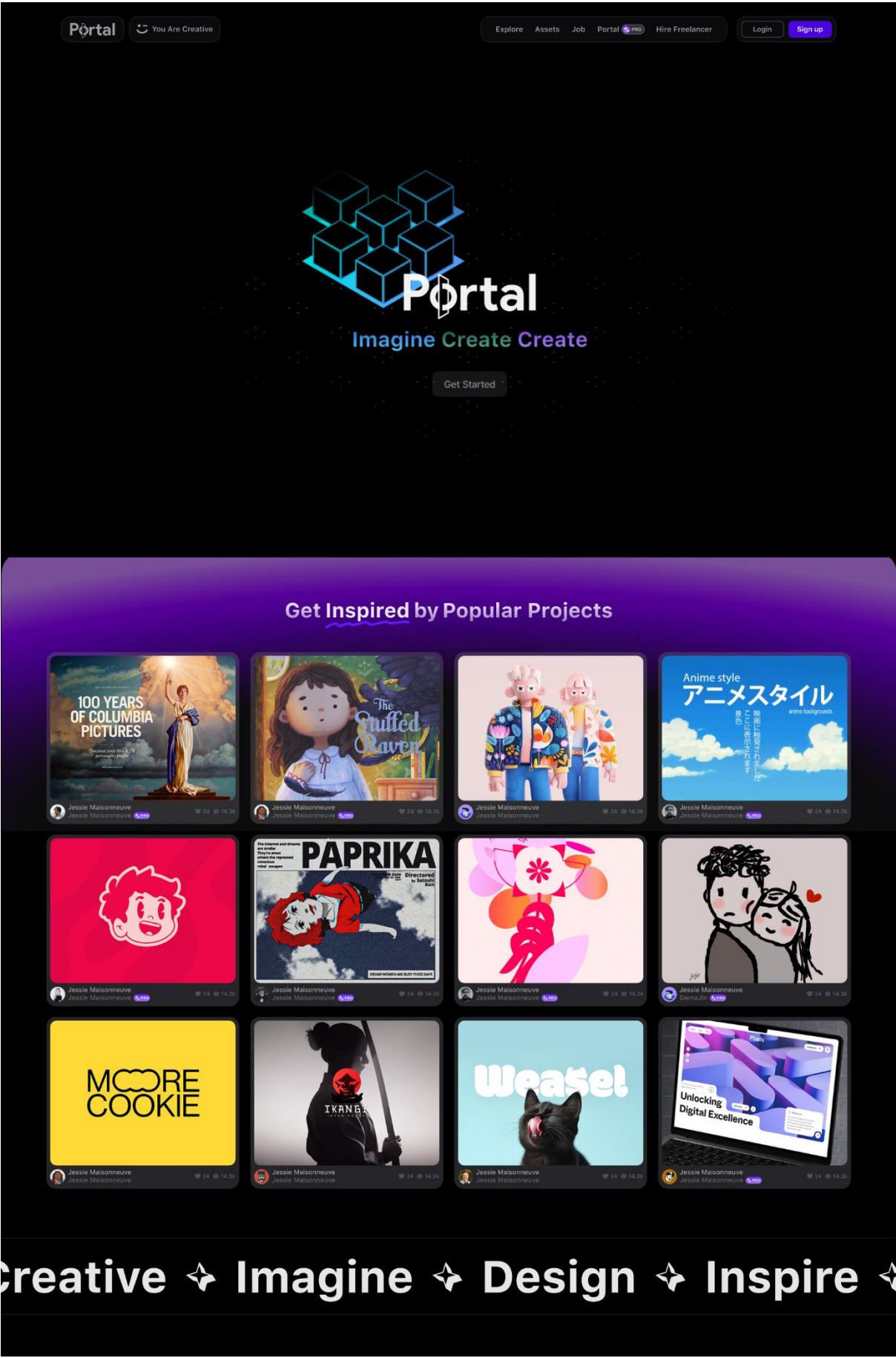


Figure.32: Landing Page

## 4.2. Registration Page

The registration page through which a new user can be registered, Figure (33).

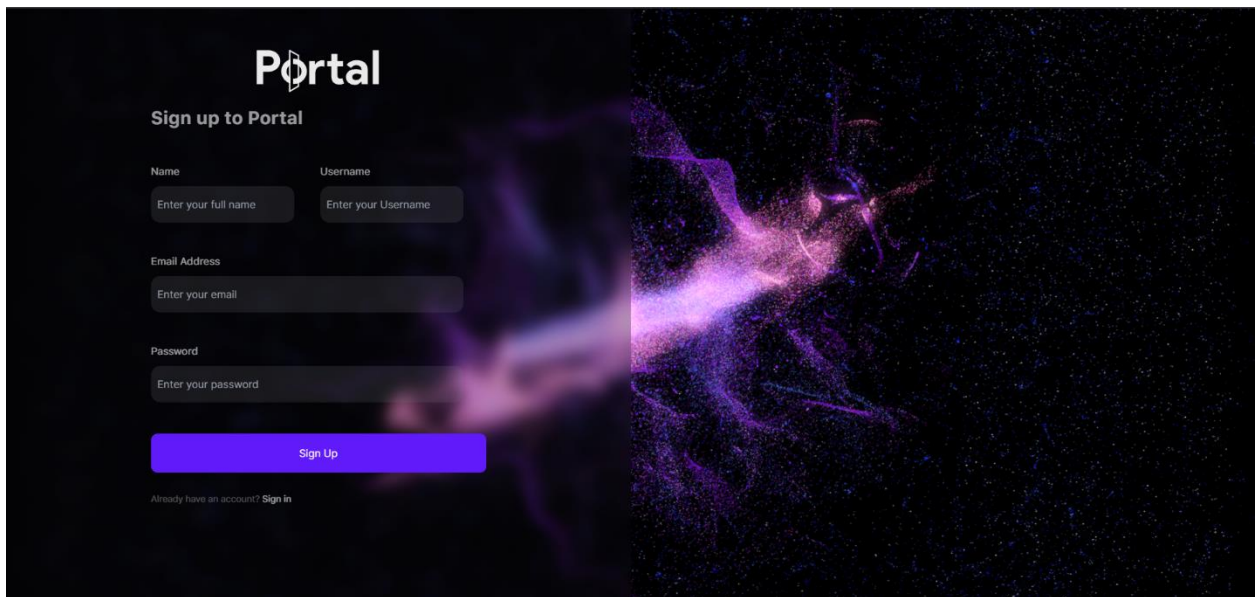


Figure.33: Registration Page

## 4.3. Login Page

The registration page, through which the user logs in to the platform, Figure (34).

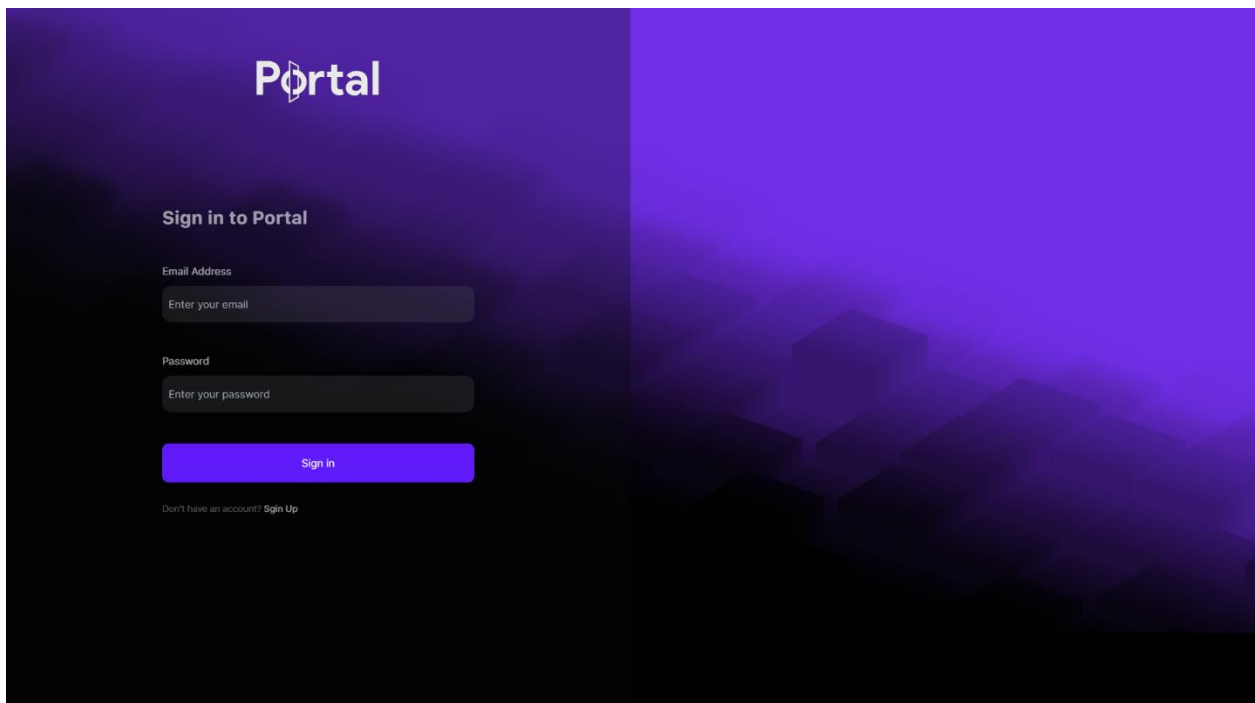


Figure.34: Login page

## 4.4. Home Page

It is the Main Page, displaying the designs published on the platform and filtering by categories

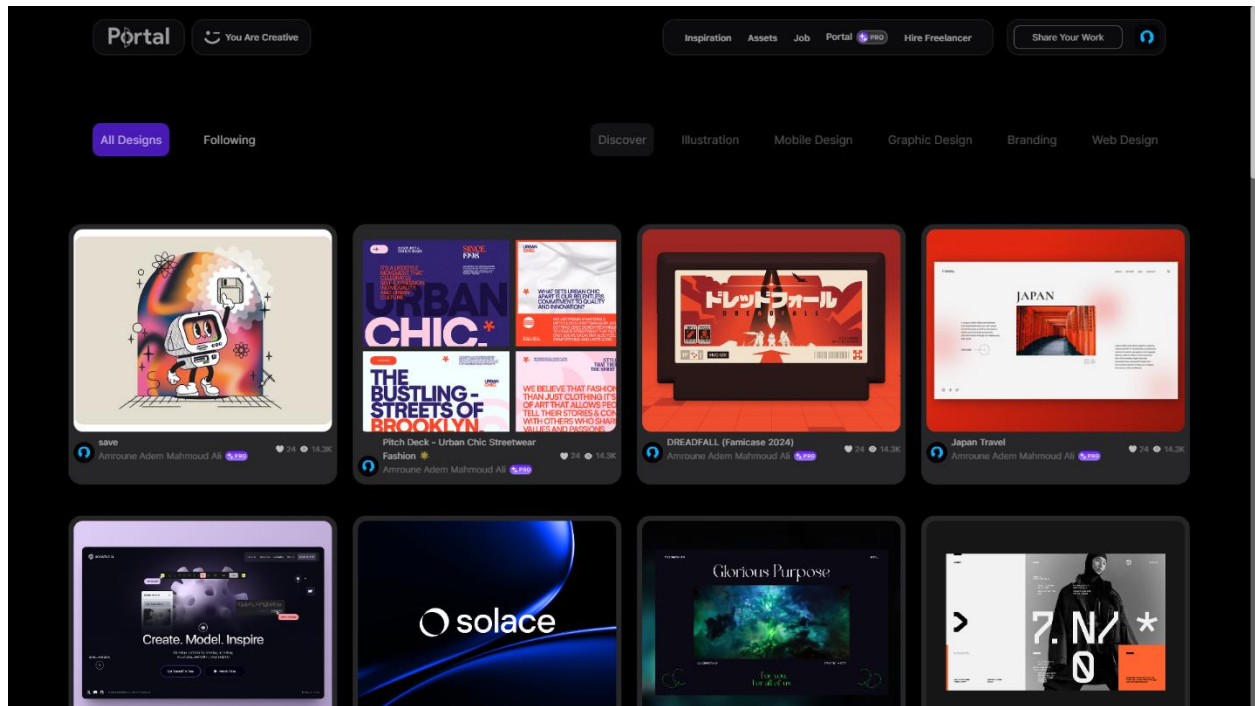


Figure.35: Home Page

### 4.4.1. Upload New Desing

It is the page for uploading a new design. It allows uploading a design, an image or a video, and multiple designs together with the ability to control the order, and enter design information at the end. Figure (36)(37)(38)

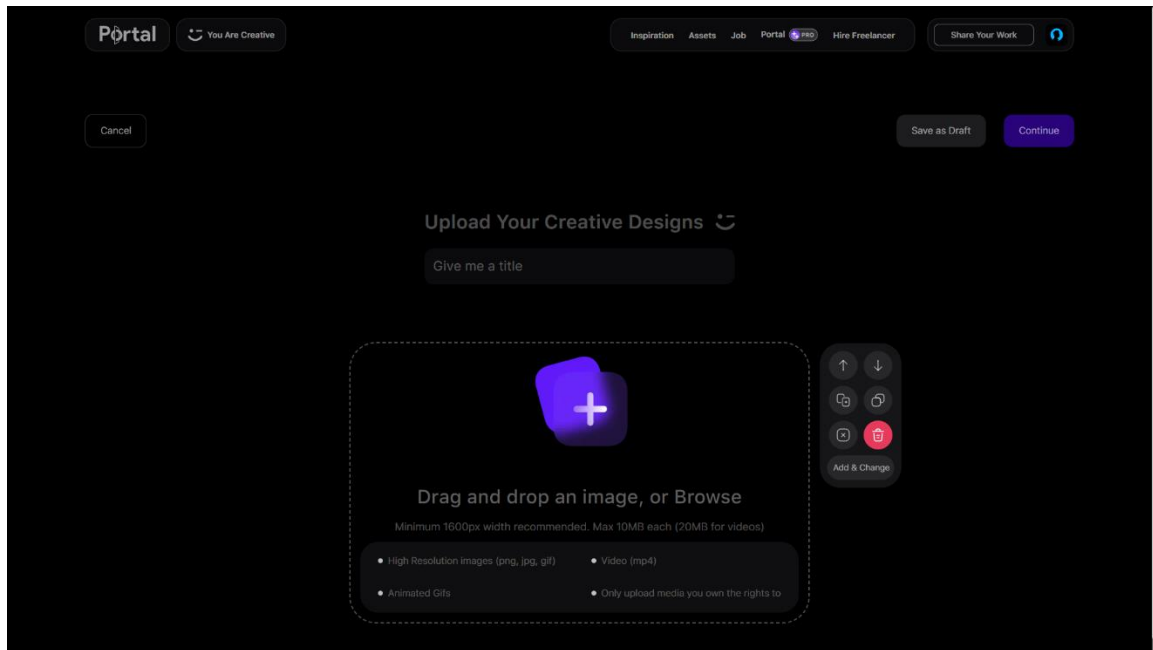


Figure.36: Upload New Desing Section

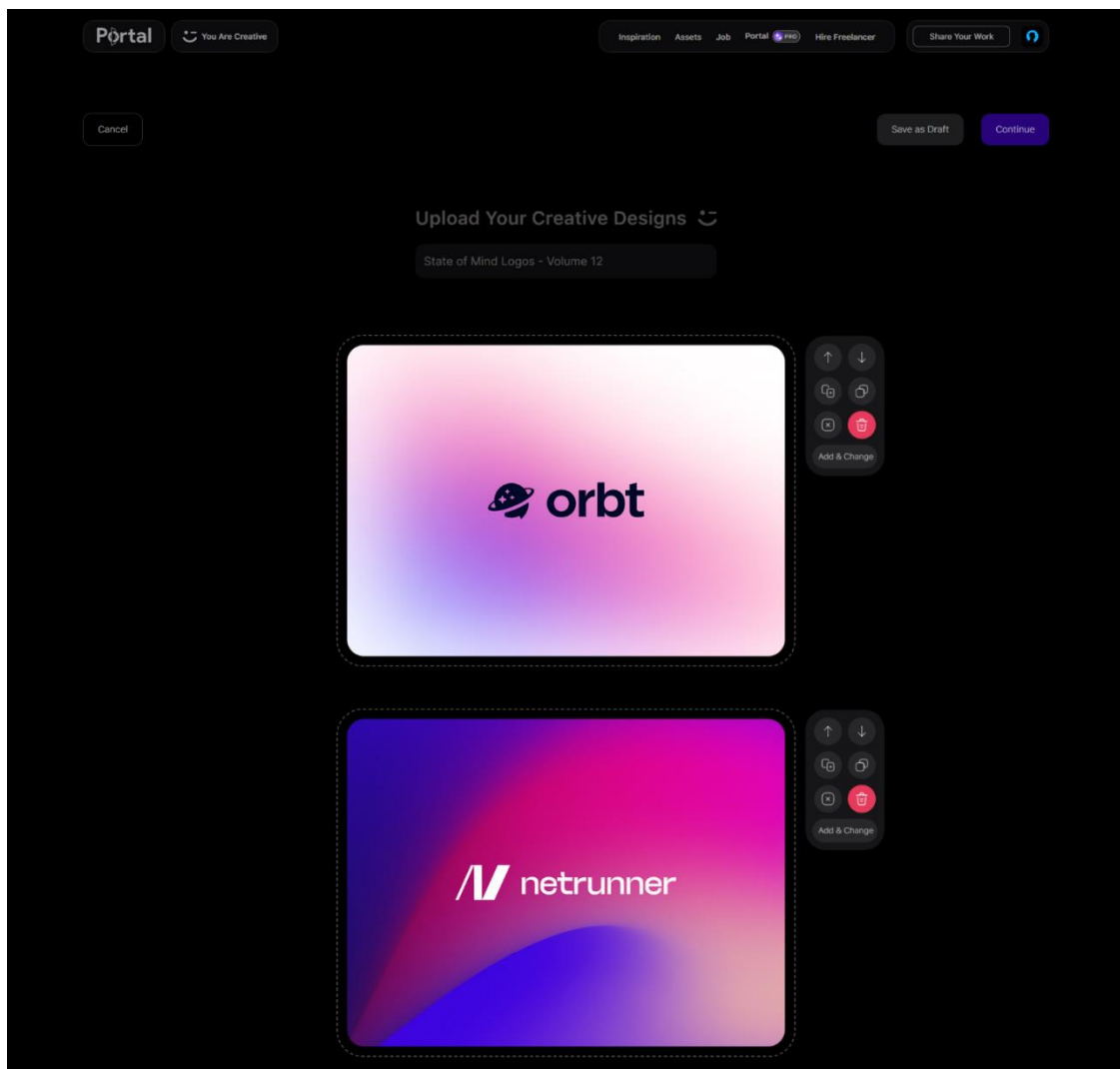


Figure.37: Add Designs

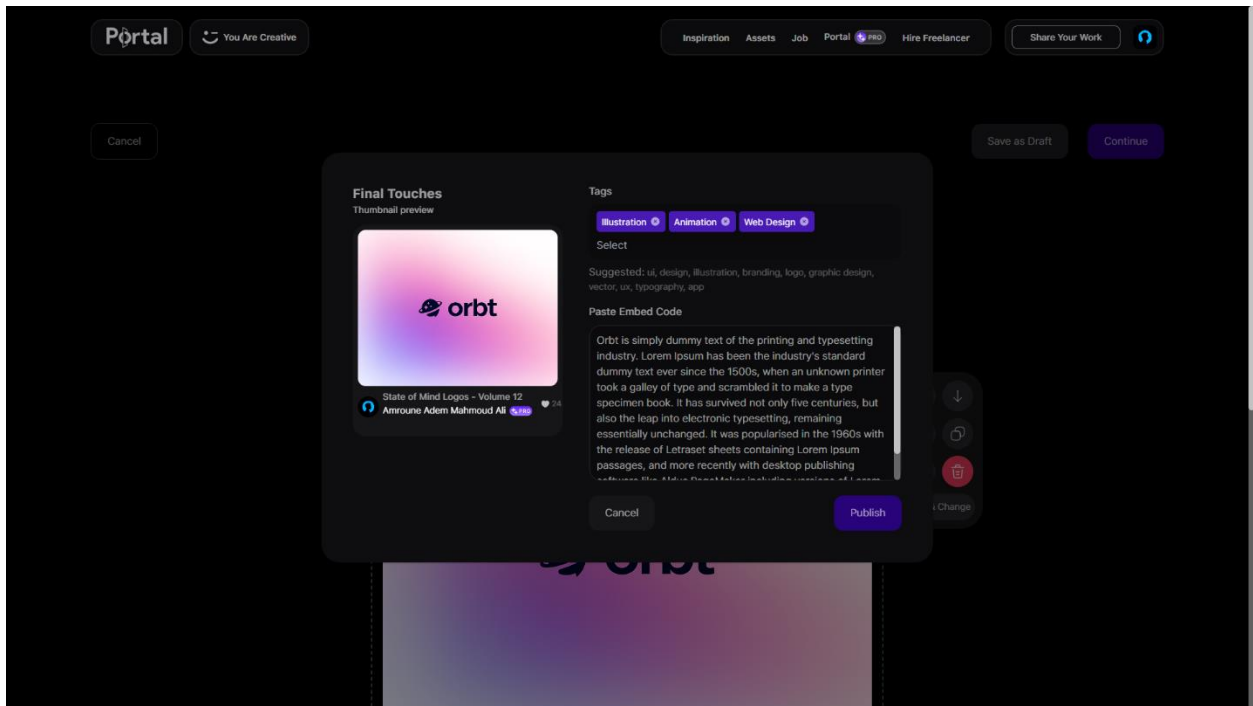
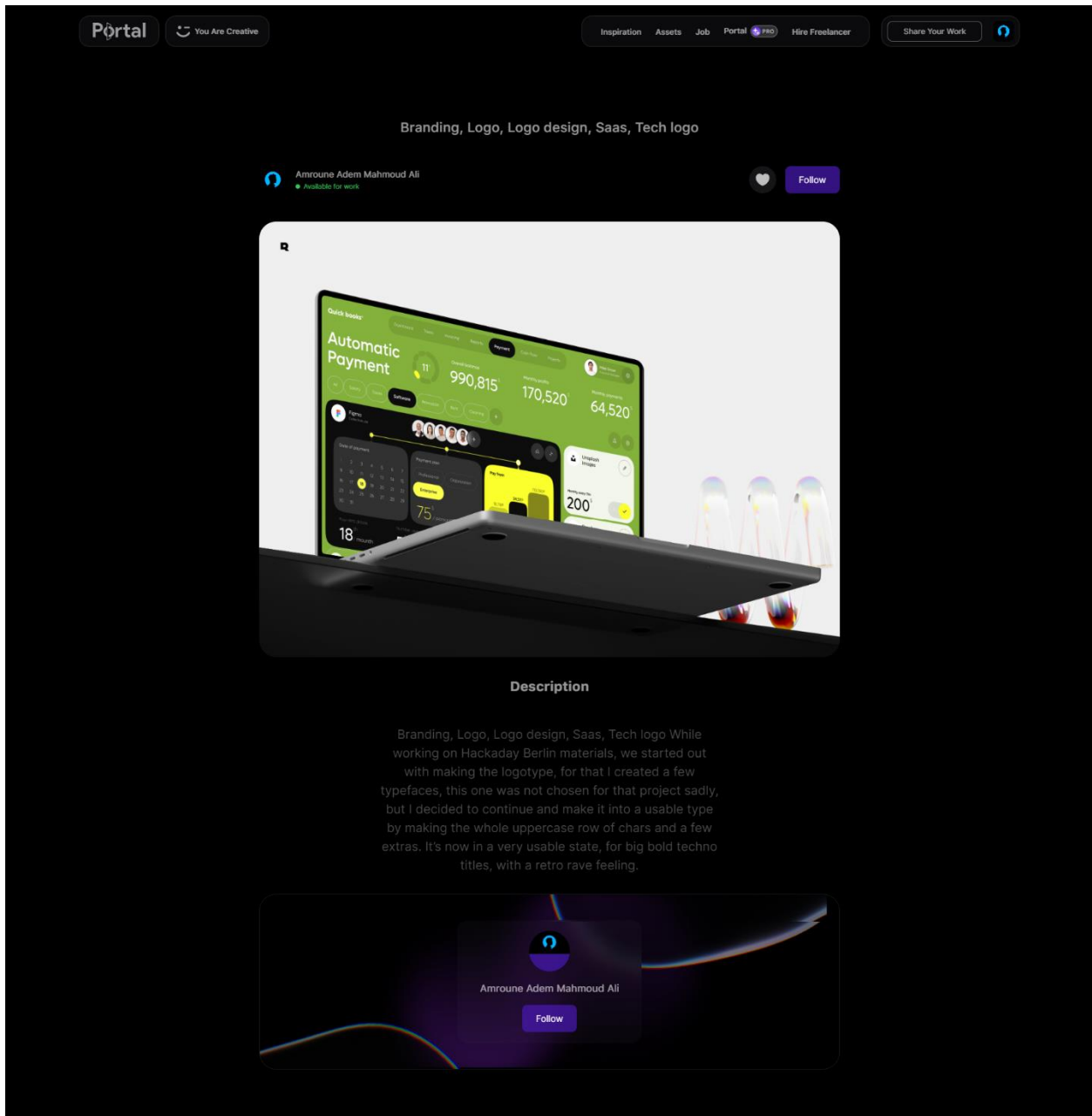


Figure.38: Add Information of New Design

#### 4.4.2. Presentation of Assets

On this page see the design, its contents and its description. Figure (39),



**Figure.40: Presentation of Design**

#### 4.5. Assets Page

It is a digital assets page, displaying the assets published on the platform, with filtering by categories, and with a price or free offer. Figure (41)

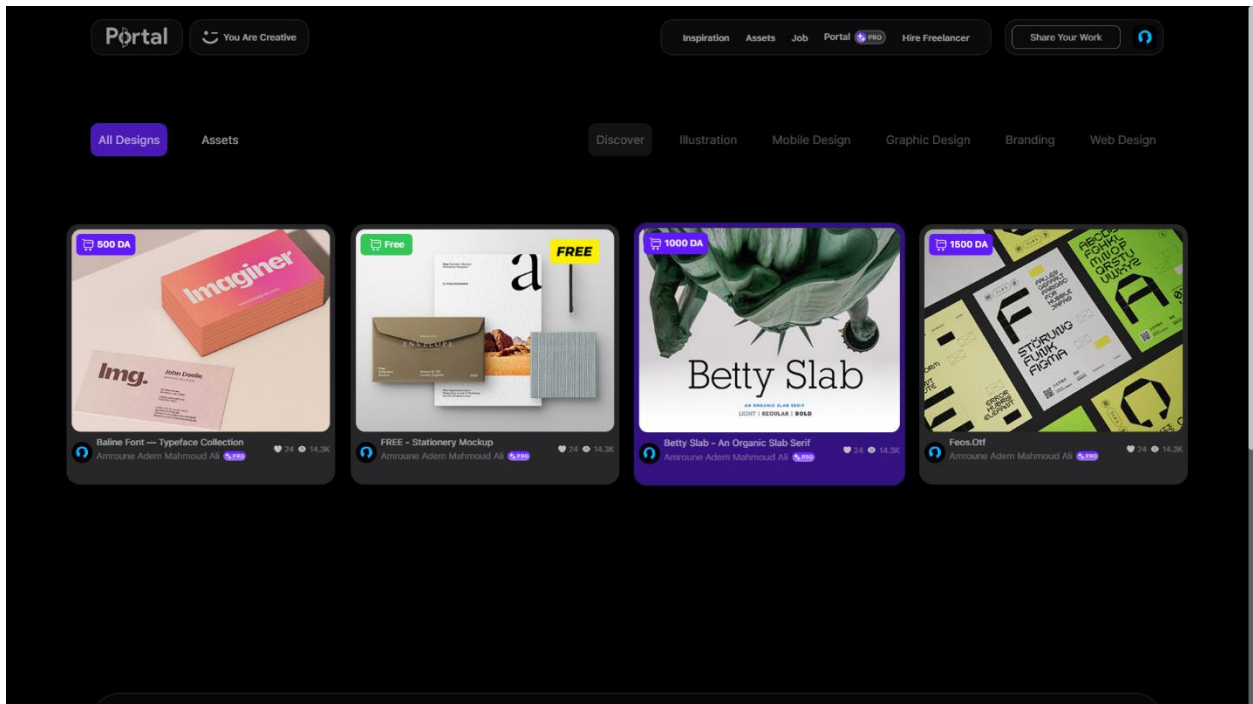


Figure.41: Assets Page

#### 4.5.1. Upload New Assets

It is the page for uploading a new asset. It allows uploading assets and placing examples of the assets, an image or video, and multiple examples together with the ability to control the order, and enter design information at the end. Figure (42)(43)(44).

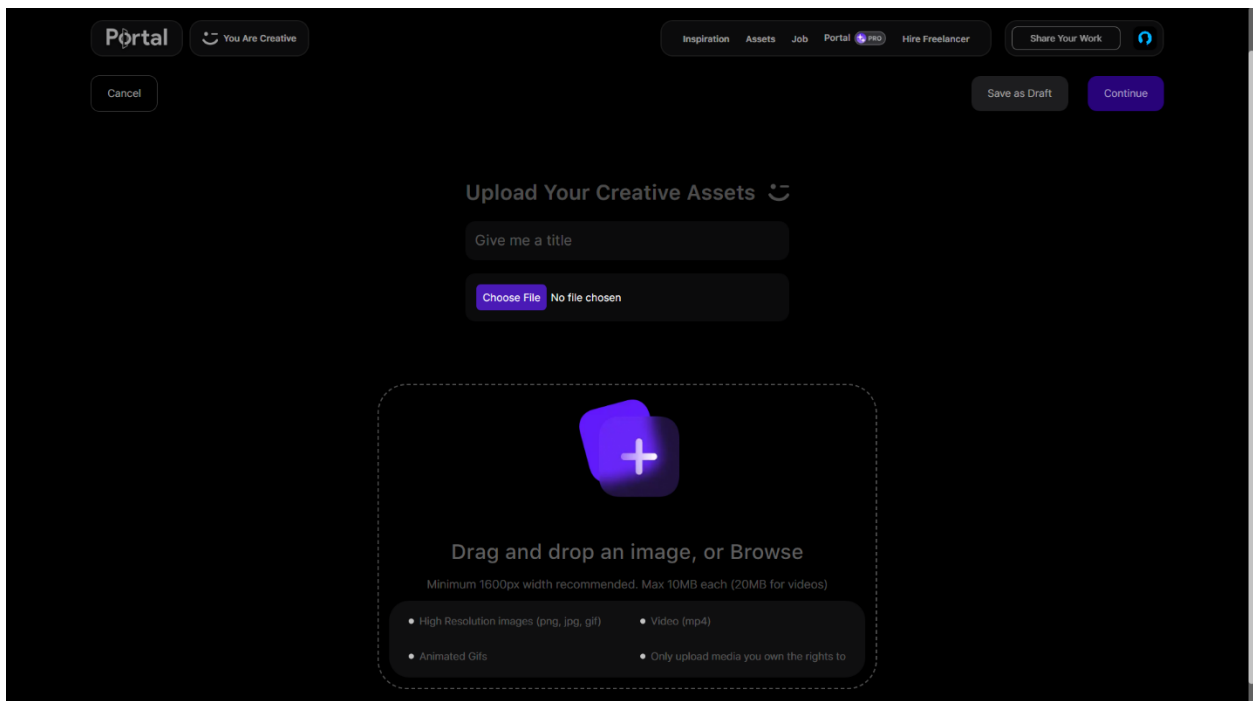


Figure.42: Upload New Assets Section

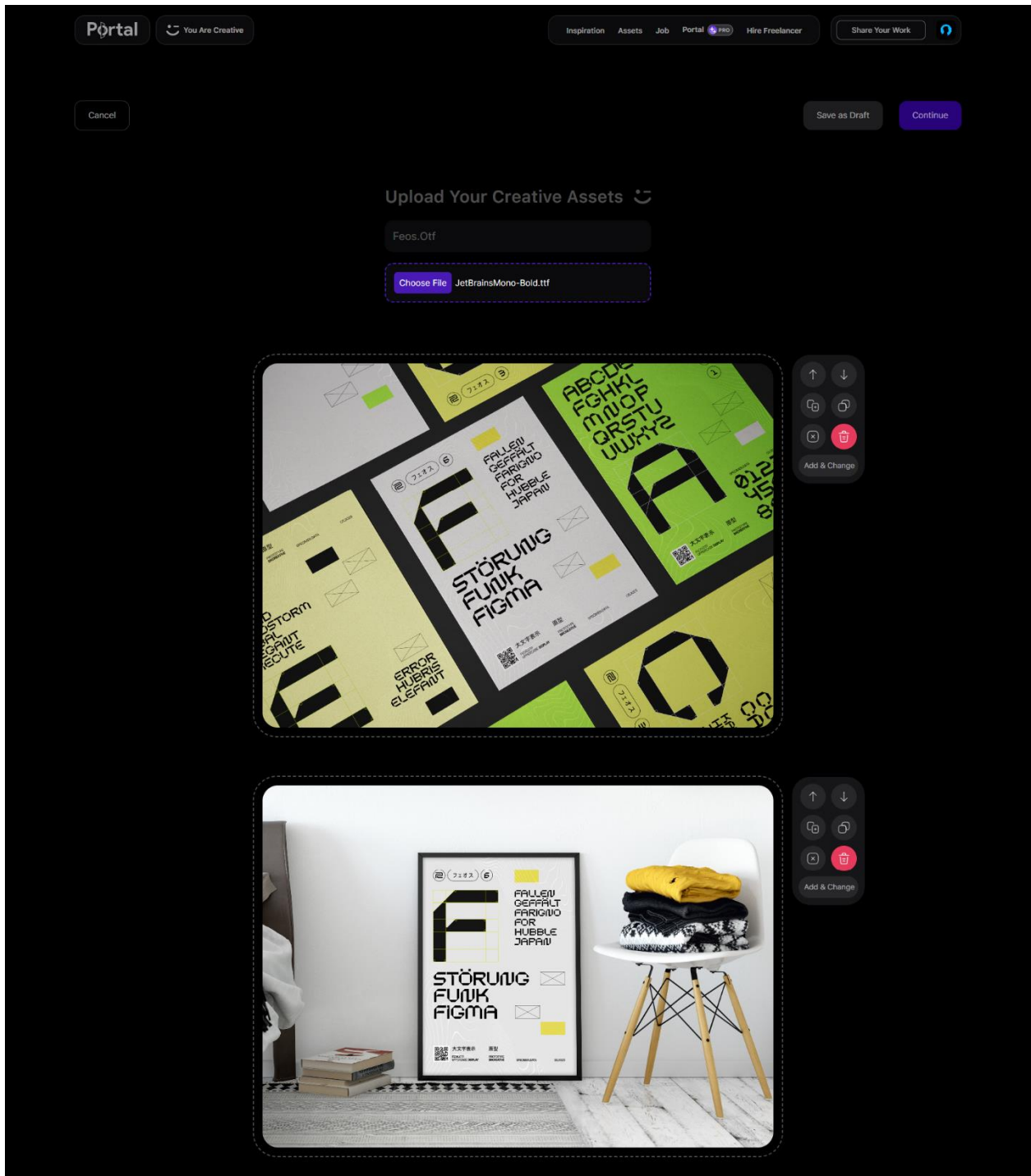
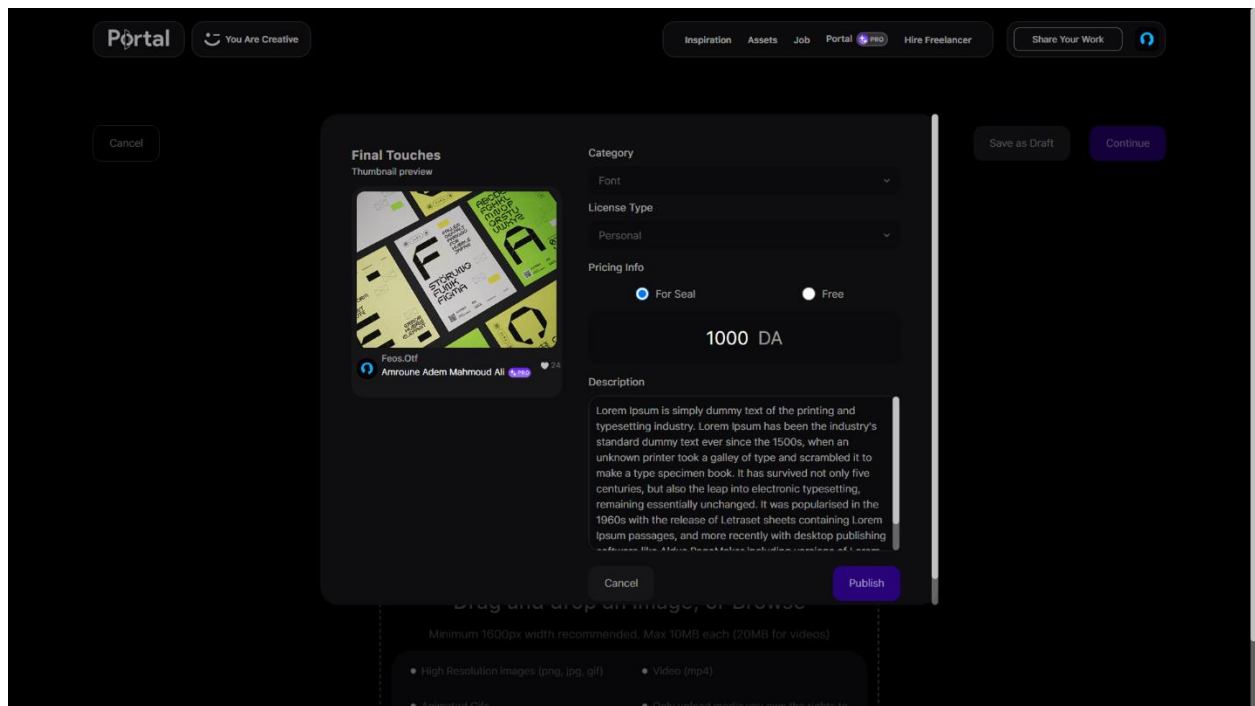


Figure.43: Add Examples of Assets



**Figure.44: Add Assets Information**

#### 4.5.2. Presentation of Assets

On this page, you can see examples of assets that can be downloaded for free or purchased, and their contents and description Figure (45).

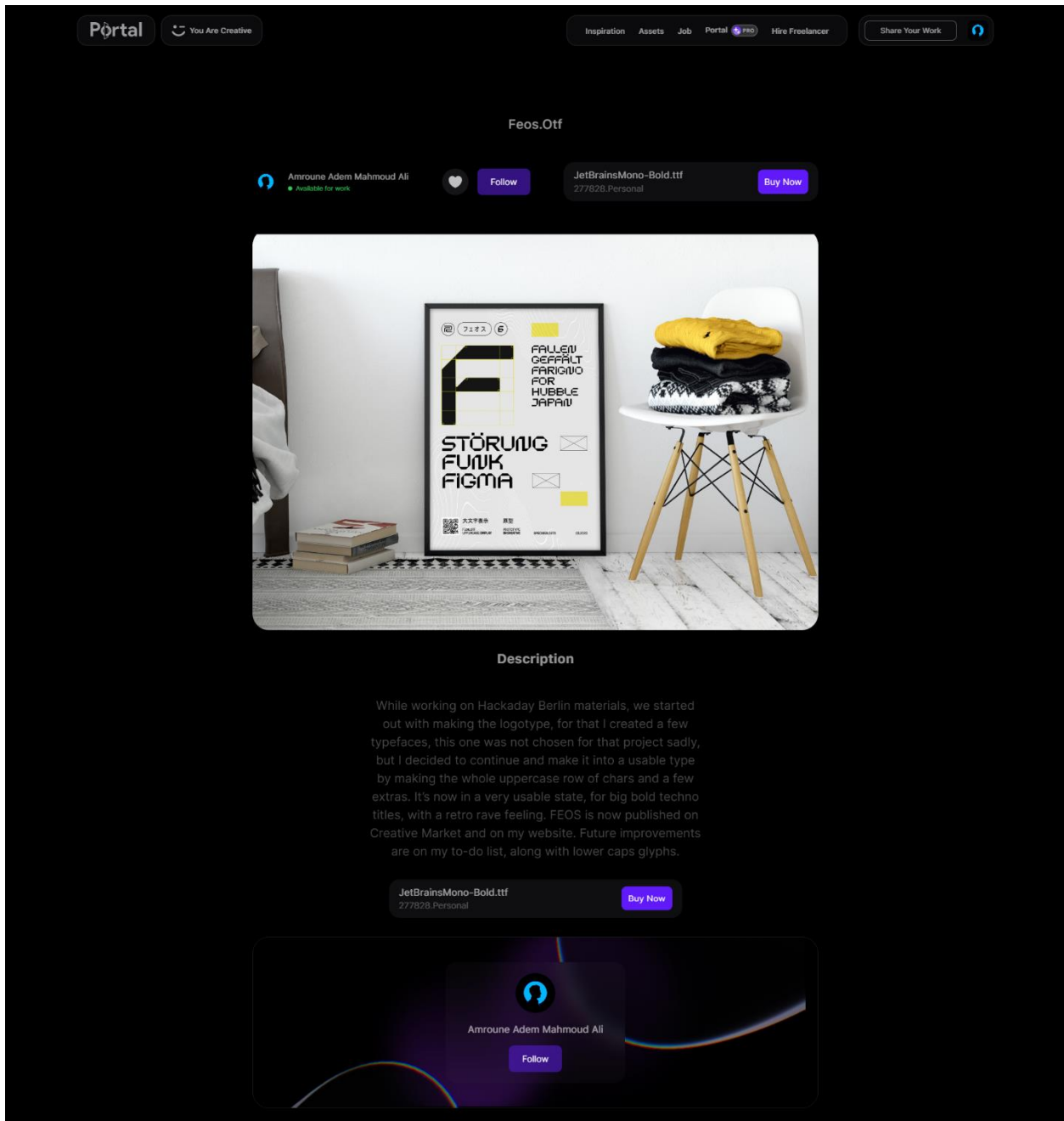
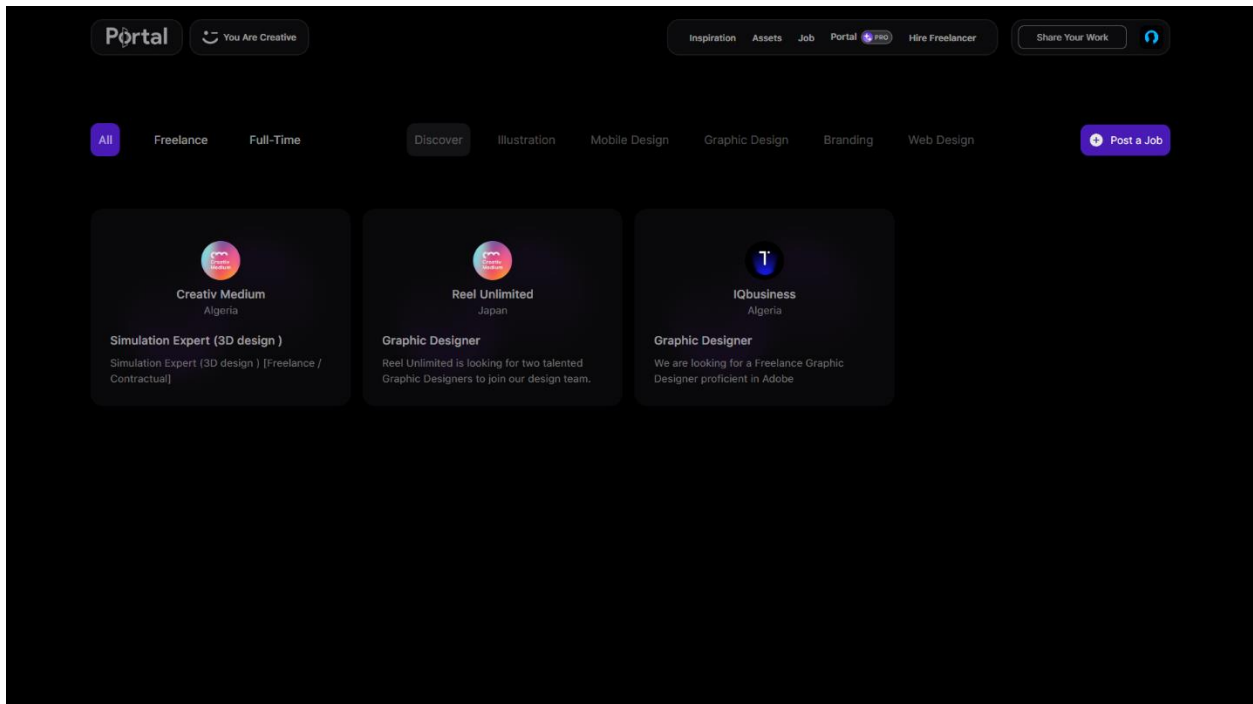


Figure.45: Presentation Assets

#### 4.6. Jobs

On this page, jobs are displayed by freelancers, companies, and professionals, with filtering by categories, type, and job Figure (46).



Fiureg.46: Jobs Offers

#### 4.6.1. Add New Job

In this Modal, you enter job information and determine if you are an Individual person or a professional (company). Figure (47)

Figure.47: Add New Job Form

## 4.6.2. Presentation Job

On this page you can view the job details, description, and Apply link. Figure (48).

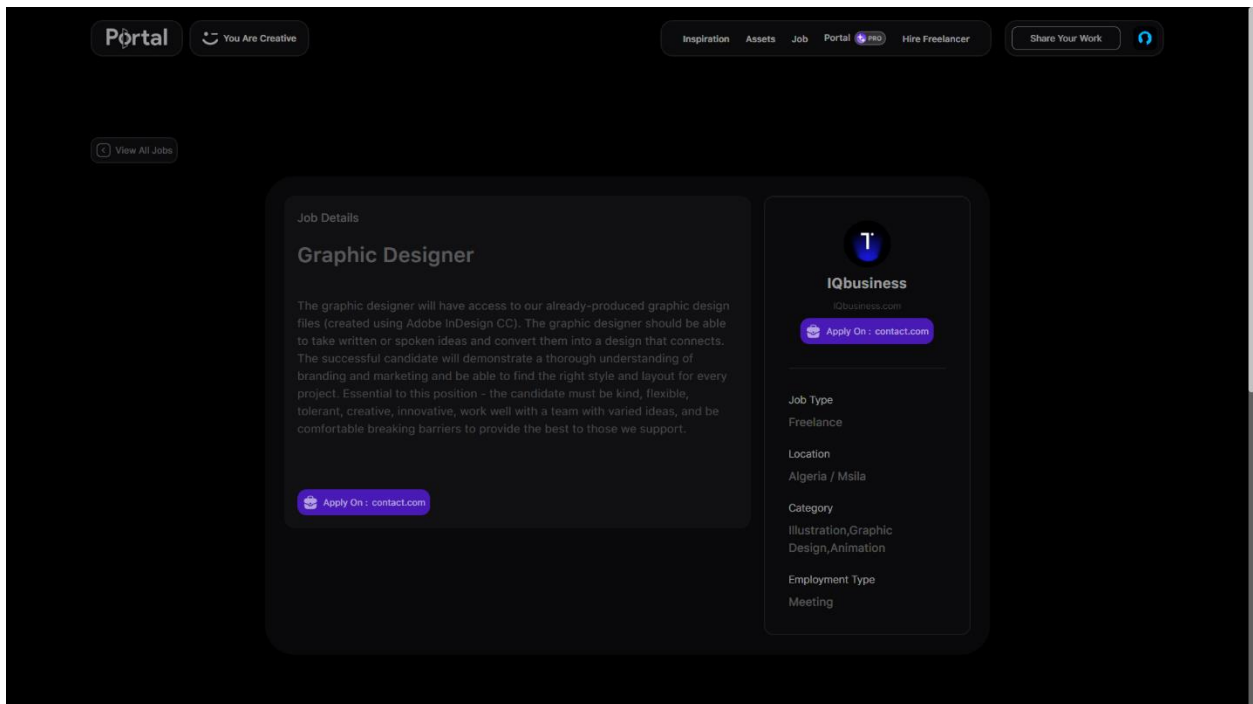


Figure.48: Presentation Job Details

## 4.7. Services

It is a page dedicated to services. The services are displayed, detailed, and the price of the service. Figure (49)

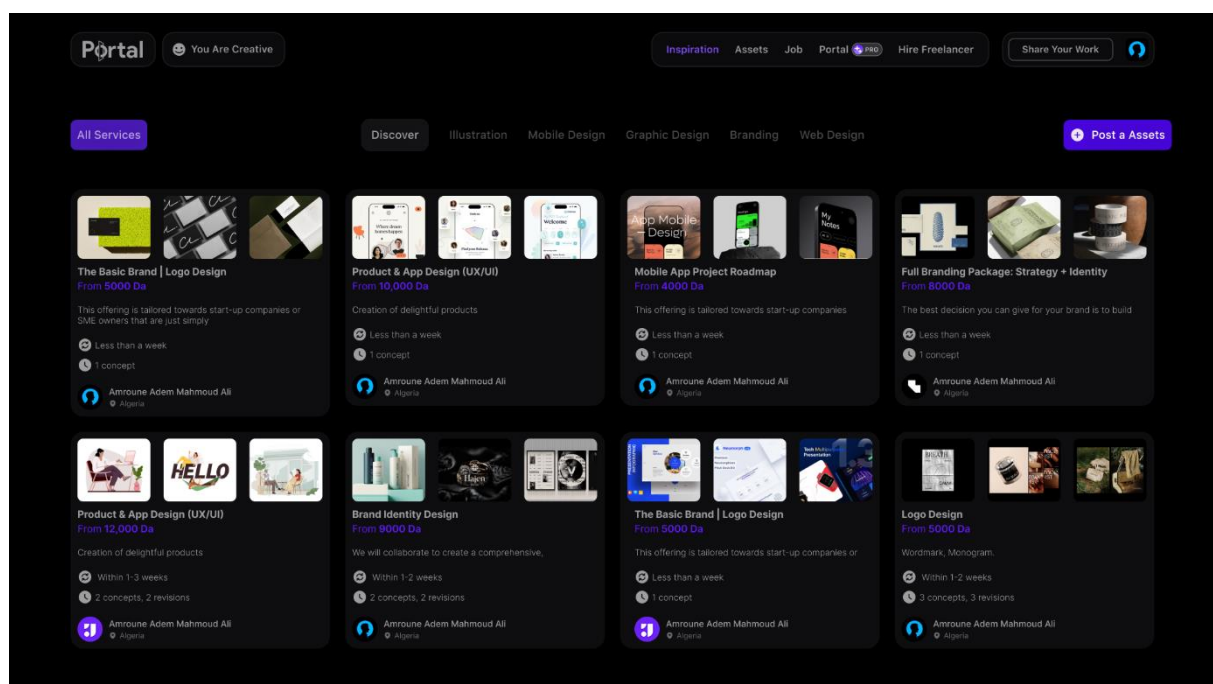


Figure.49: Services

### 4.7.1. Presentation Service

It is the service display page. The service is displayed, detailed, the price of the service, the designs are designed by the service publisher, and a Apply button. Figure (50).

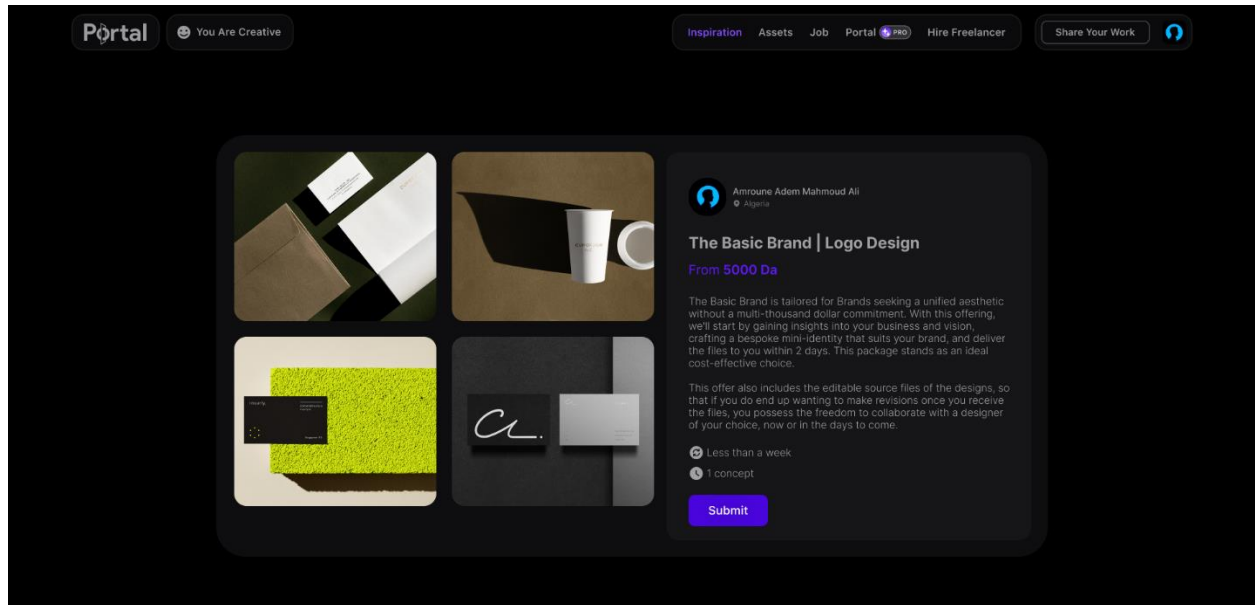


Figure.50: Presentation Service

### 4.8. Profile Page

It is a page for profile, where users can see their personal information's, designs, digital assets, services, and also saved designs, that have been liked and About Me.

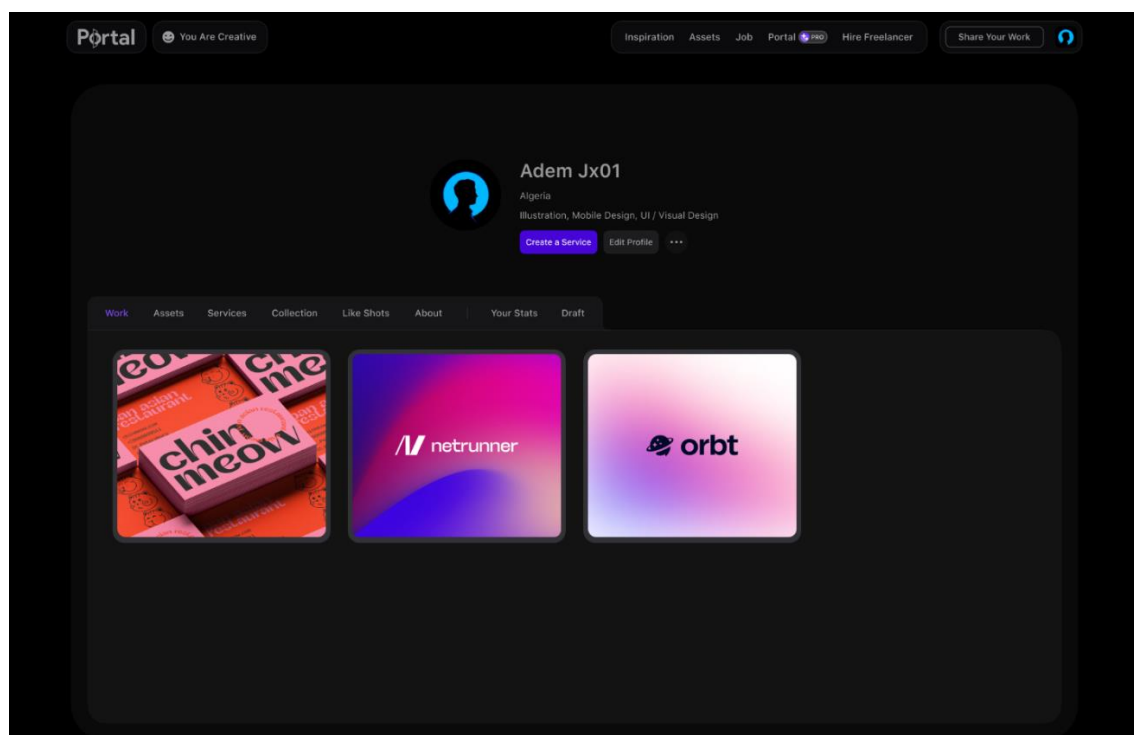


Figure.51: Profile Page

## 4.9. Edit Profile

It is a page for modifying the profile, as it allows the user to modify his account, reset the password, add social media accounts, and also delete the account. Figure (52)

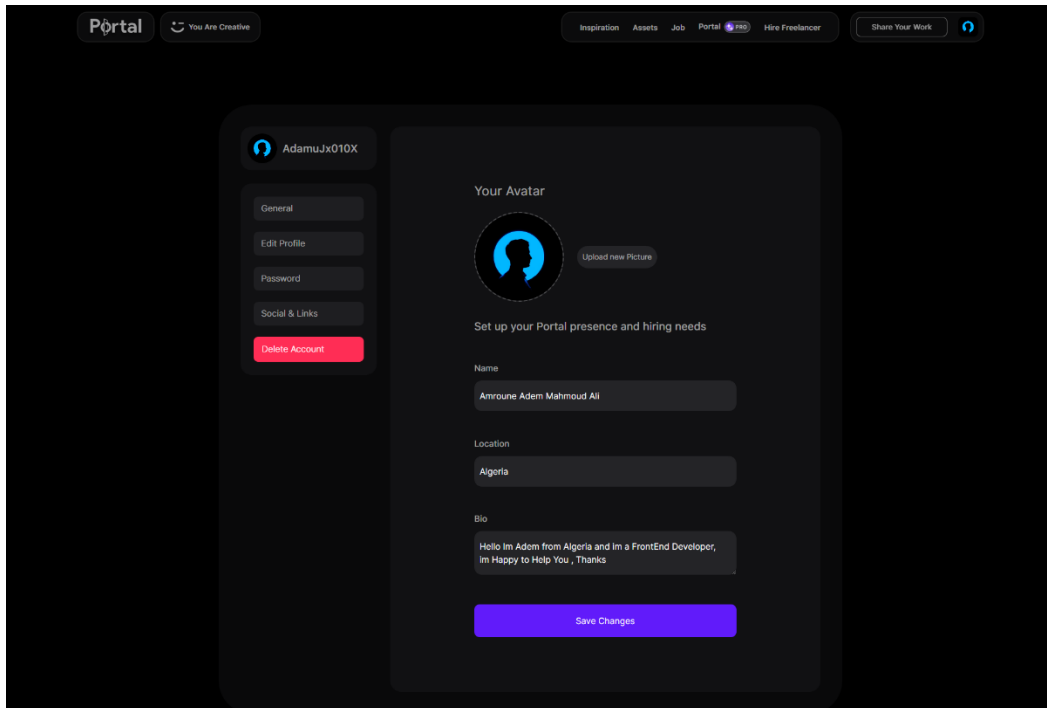


Figure.52: Edit Profile

## 5. Conclusion

In this chapter, we discussed the stages of the platform's development from design to implementation, knowledge of all the technologies that were used, and the most important snapshots of the platform's interface that explain how it works.

## Conclusion

This work introduces a groundbreaking concept aimed at revolutionizing the field of design and designers in Algeria, particularly within the local market. The idea involves creating a platform specifically tailored for designers, providing them with the opportunity to showcase their designs, share experiences, discover creative talents, and engage with the creative community. This platform serves as a multifaceted tool, allowing designers to exhibit their work, sell digital assets, post job opportunities, and access a wealth of inspiration and contemporary designs, it as a portfolio for designers.

In order to bring this idea to fruition and develop the platform, a comprehensive study of the system, its requirements, characteristics, and analysis is imperative. This involves defining the project's structure, extracting necessary requirements and rules for platform usage, building a community of designers, enhancing user interaction through additional features, and devising strategies to generate income through various avenues. By adopting a diverse approach, we aim to establish multiple revenue streams while also learning from competitors' mistakes and shortcomings to refine our platform.

The next phase involves designing the system and presenting proposed structures that illustrate how the platform operates in different scenarios. Utilizing renowned architectural and structuring techniques such as UML enables us to gain a deeper understanding of the system's functionality.

Upon completion of the system analysis, architecture, and design, we proceed to implement the platform using key techniques outlined in the third chapter. We presented the most important interfaces of the platform and explained how it works.

Looking ahead to future developments, we are excited about implementing several innovative ideas and features. One of our key objectives is to broaden the range of designs that can be uploaded, including the incorporation of interactive 3D elements and electronic payment options. Additionally, we are exploring the possibility of creating a portfolio website where designers can showcase their work.

Furthermore, enhancing the platform's performance by increasing chat messaging capabilities and integrating artificial intelligence will not only enhance the overall user experience but also streamline processes, ultimately saving time for designers. These advancements are aimed at elevating our platform to new heights and providing our users with cutting-edge tools and features.

The experience was truly unique, providing us with valuable insights and knowledge that were previously unknown to us. We approached our work with unwavering perseverance and diligence, resulting in a truly enriching and beautiful experience.

## References

- [1] GeeksforGeeks. (2024, January 12). *What is Web App?* GeeksforGeeks. <https://www.geeksforgeeks.org/what-is-web-app/>
- [2] Codecademy. (n.d.). *What is a Web App?* Codecademy. <https://www.codecademy.com/article/what-is-a-web-app>
- [3] Petrenko, V., & Litslink. (2024, April 29). *Understanding contemporary web application architecture: key components, best practices, and beyond.* Litslink. <https://litslink.com/blog/web-application-architecture>
- [4] William, & William. (2024, April 26). *Web Application Architecture: The Latest Guide 2024.* ClickIT. <https://www.clickittech.com/devops/web-application-architecture/amp/>
- [5] GeeksforGeeks. (2022, January 27). *Difference Between Web application and Website.* GeeksforGeeks. <https://www.geeksforgeeks.org/difference-between-web-application-and-website/>
- [6] Rexettit. (2023, September 2). *Types of web applications - Rexettit - Medium.* Medium. <https://medium.com/@rexettit/types-of-web-applications-f13265a3e289>
- [7] Adam. (2023, September 27). *The Web Application Development Process Flow: 6 Key steps.* Blurify. <https://blurify.com/blog/the-web-application-development-process/>
- [8] Shah, N. (2023, November 7). *Web Application development Process: Step-by-Step Guide.* WPWeb Infotech. <https://wpwebinfotech.com/blog/web-application-development-process/>
- [9] Savaniya, N. (2023, December 4). *Web application development: process, tools, & examples.* Medium. <https://medium.com/@niteshsavaniyaofficial/web-application-development-process-tools-examples-39c41e4ca98b>
- [10] *What is Unified Modeling Language.* (n.d.). Lucidchart. <https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language>
- [11] *UML Use Case Diagram Tutorial.* (n.d.). Lucidchart. <https://www.lucidchart.com/pages/uml-use-case-diagram>
- [12] Creately. (2021, February 3). *UML Class Diagram Tool.* <https://creately.com/diagram-type/class-diagram/>
- [13] *What is Sequence Diagram?* (n.d.). <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
- [14] *What is JavaScript? - Learn web development | MDN.* (2024, January 24). MDN Web Docs. [https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript)
- [15] *HTML: HyperText Markup Language | MDN.* (2024, April 25). MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [16] *CSS: Cascading Style Sheets | MDN.* (2024, March 5). MDN Web Docs. <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [17] Appwrite. (n.d.). *GitHub - appwrite/appwrite: Your backend, minus the hassle.* GitHub. <https://github.com/appwrite/appwrite>

- [18] Appwrite. (n.d.-b). GitHub - appwrite/console: The Console that makes Appwrite tick from the browser 📄. GitHub. <https://github.com/appwrite/console>
- [19] *Next.js overview*. (2024, January 17). Sanity.io. <https://www.sanity.io/glossary/next-js>
- [20] Boilerplate, N. S. (2024, January 20). Is Next.js a frontend or backend framework? - Next.js SAAS boilerplate - medium. *Medium*. <https://medium.com/@nextjs-saas-boilerplate/is-next-js-a-frontend-or-backend-framework-01c385342aa9>
- [21] Deshpande, C. (2023, October 5). *The best guide to know what is react*. Simplilearn.com. <https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>
- [22] Budiman, M. A. (2023, September 9). A Comprehensive Introduction to Tailwind CSS - Muhammad Alif Budiman - Medium. *Medium*. <https://medium.com/@alifm2101/a-comprehensive-introduction-to-tailwind-css-36bc9cb81a1c>
- [23] K, N. (2023, April 19). *Introduction to Web Animations with GSAP*. DEV Community. <https://dev.to/topboyasante/introduction-to-web-animations-with-gsap-645#:~:text=GSAP%2C%20or%20GreenSock%20Animation%20Platform,popular%20choice%20among%20developers%20worldwide.>
- [24] Tuama, D. Ó. (2023, October 23). *What is VS Code? (Visual Studio Code) - Code Institute IE*. Code Institute IE. <https://codeinstitute.net/global/blog/what-is-vs-code/>
- [25] *What is Postman? Postman API Platform*. (n.d.). Postman API Platform. <https://www.postman.com/product/what-is-postman/>
- [26] *3D Design with Spline*. (n.d.-b). <https://www.schoolofmotion.com/blog/3d-design-with-spline>
- [27] Twarog, A. (2021, June 21). *What is Figma? A Design Crash Course [2021 Tutorial]*. freeCodeCamp.org. <https://www.freecodecamp.org/news/figma-crash-course/>
- [28] Wikipedia contributors. (2024, January 7). *Lucidchart*. Wikipedia. <https://en.wikipedia.org/wiki/Lucidchart>

## Abstract

With the evolution of the Internet from static pages to complex, multi-tasking platforms post-Web 2.0, it has become a crucial tool for communication and collaboration. Our project aims to develop a website for the Algerian creative community to enhance design skills. This platform will enable users to showcase their skills, search for talents, and communicate with other professionals, serving as a hub for publishing work, selling digital assets, and gaining inspiration. Additionally, the platform will provide opportunities for professional development and enhance communication, ensuring an interactive and vibrant space that encourages Algerian designers to thrive.

**Key words:** Web 2.0, Digital Platform, Design, Algerian Creative Community

---

### ملخص:

مع تطور الإنترنت من صفحات ثابتة إلى منصات معقدة ومتعددة المهام بعد ثورة الويب 2.0، أصبحت أداة حيوية للتواصل والتعاون. يهدف مشروعنا إلى تطوير موقع إلكتروني للمجتمع الإبداعي الجزائري، لتعزيز مهارات التصميم. ستتمكن هذه المنصة المستخدمين من عرض مهاراتهم، والبحث عن المواهب، والتواصل مع المحترفين الآخرين، لتكون بمثابة مركز لنشر الأعمال وبيع الأصول الرقمية والحصول على الإلهام. بالإضافة إلى ذلك، ستوفر المنصة فرصاً للتطوير المهني وتعزيز التواصل مما يضمن مساحة تفاعلية وحيوية تدفع للمصممين الجزائريين للازدهار.

**الكلمات المفتاحية:** الويب 2.0، منصة رقمية، التصميم، المجتمع الإبداعي الجزائري

---

## Résumé

Avec l'évolution d'Internet, passant de pages statiques à des plateformes complexes et multitâches après la révolution du Web 2.0, il est devenu un outil crucial pour la communication et la collaboration. Notre projet vise à développer un site web pour la communauté créative algérienne afin d'améliorer les compétences en design. Cette plateforme permettra aux utilisateurs de présenter leurs compétences, de rechercher des talents et de communiquer avec d'autres professionnels, servant de hub pour publier des œuvres, vendre des actifs numériques et trouver de l'inspiration. De plus, la plateforme offrira des opportunités de développement professionnel et améliorera la communication, garantissant un espace interactif et dynamique qui encourage les designers algériens à prospérer.

**Mots-clés :** Web 2.0, Plateforme numérique, Design, Communauté créative algérienne