  
 **[JopShop]**

Group members:

|  |  |
| --- | --- |
| Name | ID |
| Abdelrahman Khaled Abdo | 32101003011 |
| Ibrahem Mohammad Abu Othman | 32101003013 |

**Supervisor**

**[Dr.Dheeb Albashish]   
 Graduation project Report SUBMITTED IN FULFILMENT FOR THE DEGREE OF Bachelor of Computer Information System**

**Computer Information System Department, Prince Abdullah bin Ghazi Faculty of Information and Communication Technology,**

**Al-Balqa Applied University, Salt, Jordan**

**2025**

# **Acknowledgement**

First, we want to thank the Dean of the College of Prince Abdullah bin Ghazi ***Dr. Osama Dorgham***, Deputy Dean ***Dr.Dheeb Al-Bashish*** and teaching staff there.

Second, our thanks and appreciation to ***Dr. Dheeb Al-Bashish,*** project supervisor,

who gave us a hand to build our project systematically, until it finished.

Finally, our heartfelt gratitude extends to all individuals who’s contributed directly or indirectly to the successful completion of this project. Their support, guidance, and encouragement were invaluable in shaping the outcome and enhancing our learning experience. We deeply appreciate their involvement and assistance throughout this journey.

# **Abstract**

This project presents a comprehensive digital platform that merges two primary components: a freelance services marketplace and an online store. The freelance section allows individuals and businesses to offer and request a wide range of professional services, providing opportunities for flexible employment, remote collaboration, and skill-based networking. Clients can easily find qualified freelancers to meet their project needs, while freelancers can showcase their expertise and secure work opportunities.

Parallel to this, the integrated store enables users to sell and purchase a variety of products, creating a vibrant marketplace that caters to different consumer needs. Sellers can easily manage their products, while buyers enjoy a streamlined shopping experience with secure transactions and responsive customer support.

By combining freelancing and e-commerce in one unified platform, the project aims to meet the evolving demands of the digital economy. It offers users a convenient and efficient way to access both services and goods without the need to switch between separate platforms. The system is designed with a strong emphasis on user experience, security, scalability, and future adaptability, ensuring that it can grow alongside the users' needs. This innovative approach fosters an interconnected community where collaboration, entrepreneurship, and commerce thrive side by side.

Contents

[**Acknowledgement** I](#_Toc155793282)

[**Abstract** II](#_Toc155793283)

////………………………………………………………………………………………………………………………………………….

[**Chapter one: Introduction** 1](#_Toc155793284)

[**1.1 Project motivation** 1](#_Toc155793285)

[**1.2 Problem statement** 1](#_Toc155793286)

[**1.3 Project Aim and Objective** 1](#_Toc155793287)

[**1.4 Project scope:** 3](#_Toc155793288)

[**1.5 Project hardware and software requirements:** 3](#_Toc155793289)

[**1.6 Project limitation** 8](#_Toc155793290)

[**1.7 Project expected output** 8](#_Toc155793291)

[**Chapter Two: Related Existing Systems** 9](#_Toc155793292)

[**2.1 Introduction** 9](#_Toc155793293)

[**2.2 Existing Systems** 9](#_Toc155793294)

[**2.3 Over all problems of existing Systems** 10](#_Toc155793295)

[**2.4 Over all solution approach** 10](#_Toc155793296)

[**2.5 Summary** 10](#_Toc155793297)

[**Chapter Three: System Requirements Engineering and Analysis** 11](#_Toc155793298)

[**3.1 Introduction** 11](#_Toc155793299)

[**3.2 Feasibility Study** 11](#_Toc155793300)

[**3.2.1 Technical Feasibility** 11](#_Toc155793301)

[**3.2.2 Operational feasibility** 12](#_Toc155793302)

[**3.2.3 Economic feasibility to the system** 12](#_Toc155793303)

[**3.4 Targeted users** 13](#_Toc155793304)

[**3.5 Functional Requirements Definition and specification** 14](#_Toc155793305)

[**3.6 Nonfunctional Requirements:** 16](#_Toc155793306)

[**3.7 Summary:** 16](#_Toc155793307)

[**Chapter Four: System Design** 17](#_Toc155793308)

[**4.1 Introduction:** 17](#_Toc155793309)

[**4.2 UML Use Case Diagram:** 17](#_Toc155793310)

[**ER diagram (ERD):** 21](#_Toc155793311)

[**Chapter Five: System Implementation** 22](#_Toc155793312)

[**5.1 Introduction** 22](#_Toc155793313)

[**5.2 Database Implementation** 22](#_Toc155793314)

[**5.3 Graphical User Interface Implementation** 22](#_Toc155793315)

[**Chapter Six: System Testing and Installation** 31](#_Toc155793316)

[**6.1 Introduction** 31](#_Toc155793317)

[**6.2 Heuristic Evaluation** 31](#_Toc155793318)

[**6.3 Pre-Evaluation Procedures** 33](#_Toc155793319)

[**6.4 Evaluation Procedures** 33](#_Toc155793320)

[**6.5 Post-Evaluation Procedures** 35](#_Toc155793321)

[**6.6 black box testing** 36](#_Toc155793322)

[**6.7 White Box Testing** 37](#_Toc155793323)

[**6.8 Automation Testing** 37](#_Toc155793324)

[**6.9 GUI Testing** 38](#_Toc155793325)

[**6.10 System Installation** 39](#_Toc155793326)

[**6.11 Summary** 40](#_Toc155793327)

[**Chapter seven: Project Conclusion & Future Work** 41](#_Toc155793328)

[**7.1 Introduction** 41](#_Toc155793329)

[**7.2 Overall Weaknesses** 41](#_Toc155793330)

[**7.3 Overall strengths** 41](#_Toc155793331)

[**7.4 Future Works** 42](#_Toc155793332)

[**7.5 Summary** 42](#_Toc155793333)

[**7.6 References and Appendix** 42](#_Toc155793334)

# **Chapter one: Introduction**

## **1.1 Project motivation**

## In today's rapidly evolving digital world, individuals and businesses are increasingly seeking flexible and efficient ways to both offer services and sell products online. Traditional freelance platforms and e-commerce stores operate separately, forcing users to manage multiple accounts, navigate different systems, and divide their efforts across various platforms. This separation leads to inefficiencies, added costs, and a fragmented user experience.

## The motivation behind this project stems from the need to bridge this gap by providing a unified platform that integrates freelance services and an online marketplace. By doing so, users can seamlessly switch between offering services and selling products, maximizing their opportunities for income generation and business growth. Freelancers can expand their work scope by not only offering services but also selling digital or physical products related to their expertise. Similarly, entrepreneurs can easily find skilled freelancers to support their business operations without leaving the platform.

## Furthermore, the project seeks to empower a wide range of users — from independent professionals to small businesses — by offering them a flexible, scalable, and secure environment to succeed in the digital economy. This combined platform fosters community engagement, encourages entrepreneurship, and simplifies the overall online business experience, making it more accessible and efficient for everyone.

## **1.2 Problem statement**

## Despite the growing demand for online services and e-commerce, most existing platforms treat freelance marketplaces and online stores as entirely separate ecosystems. This fragmentation forces users to maintain multiple accounts, manage different payment systems, and split their activities across several platforms, leading to inefficiency, higher operational costs, and a fragmented user experience.

## Freelancers who wish to offer both their services and related products must navigate two separate platforms, limiting their ability to fully leverage their skills and assets in one place. Similarly, businesses looking to hire talent and purchase goods face unnecessary complexity and wasted time switching between service platforms and online stores.

## There is currently a lack of an integrated solution that allows users to seamlessly offer services and sell products within a single, user-friendly platform. Without such integration, users miss out on cross-selling opportunities, unified brand management, and streamlined operations, ultimately hindering their growth and success in the digital economyThis project aims to solve this gap by developing a combined freelance and store platform that offers a cohesive, efficient, and scalable solution for individuals and businesses to operate, grow, and collaborate digitally — all in one place.

## **1.3 Project Aim and Objective**

To develop an integrated digital platform that combines freelance services and an online store, providing users with a unified environment to offer services, sell products, and manage transactions efficiently.

**Project Objectives:**

* Create a user-friendly platform that supports both service-based and product-based activities.
* Enable users to seamlessly switch between freelancing and selling within the same account.
* Implement secure and reliable transaction and payment systems.
* Foster a digital community that encourages collaboration, entrepreneurship, and growth.
* Ensure the platform is scalable, adaptable, and optimized for future needs.

**The project objectives include**:

* To design and develop an integrated digital platform that simultaneously supports freelance service offerings and product sales within a unified environment.
* To enable users to manage their services and products through a single, centralized account to enhance operational efficiency and user satisfaction.
* To create an intuitive, accessible, and responsive user interface that meets modern usability standards across multiple devices.
* To implement robust security measures for user authentication, payment processing, data protection, and transaction management.
* To facilitate seamless communication and collaboration between freelancers, vendors, and clients through built-in messaging and notification systems.
* To ensure the platform architecture is scalable and flexible, allowing for future feature enhancements and increased user base growth.
* To support entrepreneurial initiatives by providing tools and resources that encourage innovation, business development, and community engagement.
* To establish effective administrative controls for platform governance, including user management, content moderation, and financial oversight.

## **1.4 Project scope:**

## The scope of this project encompasses the design, development, and deployment of a digital platform that integrates both freelance service offerings and an online marketplace for product sales. The platform will allow users to register, create profiles, offer freelance services, list products for sale, and engage in secure transactions through a unified system.

## Key features to be developed include user authentication and account management, service posting and browsing, product listing and purchasing functionalities, integrated payment processing, order and service tracking, real-time messaging between users, and an administrative dashboard for platform management.

## The platform will be accessible via web browsers and optimized for both desktop and mobile devices to ensure a broad reach. Security, scalability, and user experience will be prioritized to support a growing user base and evolving market needs.

## However, the scope does not cover the development of native mobile applications at this stage (such as standalone Android/iOS apps) or advanced AI-based recommendation engines, which may be considered for future expansions.

## **1.5 Project software requirements:**

|  |  |
| --- | --- |
| **Software** | **Requirement** |
| OS | Windows, Mac, Linux |
| Development tool | Visual Studio , Figma, MySQL server |
| Programming languages and frameworks | JS, C#  Bootstrap, MVC |
| Database | MySQL serve |

Table (1.1) For software requirement for the Exposed site

**JavaScript (JS):**

JavaScript is one of the most essential technologies in web development. It is a high-level, interpreted language that runs in the browser and powers dynamic, interactive behaviour on web pages.  
From simple DOM manipulation to complex single-page applications (SPAs), JavaScript allows developers to handle user interactions, fetch data from servers asynchronously (using AJAX or Fetch API), and control multimedia or animations.

Modern JavaScript supports **ES6+ features** such as arrow functions, classes, template literals, restructuring, modules, and promises, which enhance code readability and performance.  
Additionally, tools like **Webpack**, **Babel**, and **npm** help in managing and bundling JavaScript projects efficiently.

It is also the foundation for powerful frameworks and libraries like **React**, **Angular**, and **Vue.js**, which enable building complex front-end applications.

📚 **Further Reading:**

* [JavaScript Documentation – MDN](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
* [JavaScript.info – Comprehensive Tutorial](https://javascript.info)

**C#:**

C# is a modern, type-safe, and object-oriented programming language developed by Microsoft. It is one of the core languages used in the **.NET ecosystem**, and is ideal for building a wide range of applications, including:

* Web applications (with ASP.NET)
* Desktop applications (using WinForms or WPF)
* Mobile apps (via Xamarin or MAUI)
* Game development (with Unity)
* APIs and microservices

C# combines the power of C++ with the simplicity of Visual Basic. It features advanced capabilities like **generics**, **delegates**, **events**, **LINQ**, **asynchronous programming (async/await)**, and **garbage collection**, which contribute to its robustness.

Its integration with the .NET platform allows for seamless development and deployment across different operating systems with .NET Core/.NET 6+.

📚 **Further Reading:**

* [Official C# Documentation](https://learn.microsoft.com/en-us/dotnet/csharp/)
* [C# Tutorials – Microsoft Learn](https://learn.microsoft.com/en-us/training/paths/csharp-first-steps/)

**Bootstrap:**

Bootstrap is a free, open-source CSS framework developed by Twitter to streamline front-end development. It offers:

* A responsive grid system
* Predefined UI components (buttons, modals, cards, alerts, forms, etc.)
* CSS utilities for typography, spacing, flex, shadows, and more
* Built-in JavaScript plugins (e.g., carousels, tooltips, dropdowns)

With Bootstrap, developers can create mobile-first, responsive designs without writing custom CSS from scratch. It ensures consistent design across pages and browsers.  
It supports customization through SASS variables and can be extended with custom themes or third-party templates.

Bootstrap 5 removed jQuery dependency, making it lighter and better suited for integration with modern JavaScript frameworks.

📚 **Further Reading:**

* [Bootstrap Documentation](https://getbootstrap.com)
* [Bootstrap Examples – Start Bootstrap](https://startbootstrap.com/)

**MVC (Model-View-Controller):**

MVC is a widely adopted architectural pattern that separates concerns in application development. It divides the application into:

* **Model**: Manages data and business logic, communicates with the database.
* **View**: The UI layer that presents data to the user and collects input.
* **Controller**: Handles user input, updates the model, and determines which view to display.

Using MVC leads to **clean, maintainable, and testable** code, as logic and presentation are separated. It’s perfect for team development since frontend and backend developers can work independently.

In **ASP.NET MVC**, developers can build robust web applications using C#, Razor views, routing, validation, and dependency injection.  
MVC is ideal for applications requiring clear separation of business rules and UI, and it also supports RESTful service creation natively.

**Advantages of MVC include:**

* Scalability and flexibility
* Reusability of components
* Better unit testing and debugging
* Clear organization and modular codebase

📚 **Further Reading:**

* [ASP.NET MVC Overview – Microsoft Docs](https://learn.microsoft.com/en-us/aspnet/mvc/)
* MVC Explained – FreeCodeCamp

MySQL Server:

MySQL is one of the world’s most popular open-source relational database management systems. It is used by companies of all sizes to manage structured data and power dynamic websites.

MySQL supports:

* **SQL (Structured Query Language)** for querying and managing data
* **ACID compliance** for reliable transactions
* **Stored procedures, triggers, views, and indexing**
* **Replication and backup features** for high availability
* **User roles and permissions** for access control

MySQL is highly scalable and integrates with many platforms and languages like PHP, Python, Java, and .NET. It is often the database of choice for content management systems like **WordPress**, **Joomla**, and **Drupal**.

In enterprise environments, MySQL is often used with:

* **MySQL Workbench** for visual database design
* **phpMyAdmin** for easy web-based administration
* **Clustered architectures** for high availability and performance

Its performance, security, and ease of use make it an ideal solution for both small web projects and large-scale enterprise applications.

📚 **Further Reading:**

* [MySQL Documentation](https://dev.mysql.com/doc/)
* MySQL Tutorial – W3Schools
* [MySQL Workbench Guide](https://dev.mysql.com/doc/workbench/en/)

## **1.6 Project limitation**

This Project will be used in our university /departments for the mean time.

## **1.7 Project expected output**

The expected output of this project is the successful development and deployment of a fully functional digital platform that integrates freelance service offerings and product sales in one cohesive system. The platform will provide the following key outputs:

1. **User Registration and Account Management:**
   * A seamless user registration process, allowing users to create accounts and manage their profiles.
   * Functionality to separate roles between freelancers, clients, and store owners, with role-based permissions.
2. **Freelance Service Marketplace:**
   * A feature to post and browse freelance services across various categories.
   * A booking system for clients to hire freelancers based on their services, skills, and availability.
3. **Online Store Integration:**
   * A fully integrated e-commerce system where sellers can list products and buyers can browse, purchase, and pay securely.
   * Integration of product inventory management, shopping cart, and order tracking features.
4. **Communication Tools:**
   * Built-in messaging system to allow real-time communication between freelancers, clients, and sellers.
   * Notification system for users to receive updates on project status, product shipments, and payment confirmations.
5. **Admin Dashboard:**
   * An administrative interface for platform management, including user management, content moderation, and financial oversight.
   * Tools to monitor platform activity, track sales, service bookings, and manage user reports.
6. **Scalability and Future Expansion:**
   * A platform architecture designed to be scalable, allowing for the addition of new features such as mobile app integration, AI-powered recommendations, and expanded product categories in the future.
7. **Security and Data Privacy:**
   * Secure user data management, including encryption for personal and transaction data.
   * Compliance with standard data privacy regulations to ensure the safety and confidentiality of user information.

**In summary**, the expected output is a unified platform that successfully blends freelancing services with e-commerce capabilities, offering a seamless and efficient user experience for all parties involved. The platform will be secure, scalable, and user-friendly, ensuring both service providers and product sellers can thrive in a digital ecosystem.

|  |  |  |
| --- | --- | --- |
| **Rule ID** | **Rule** | **Participation** |
| **1** | System analyses and specified the requirements (hardware, software) | All team members |
| **2** | Specify the requirement for each type of user (User, Admin) | All team members |
| **3** | design the website and choose the color, font, images, logo etc. | All team members |
| **4** | Implementation (front end) | Abdelrahman |
| **5** | Draw the data base and connect the different pages with the local host | Ibrahem |
| **6** | Test the website | All team members |
| **7** | Writing documentation | All team members |
| **8** | Prepare for the presentation | All team members |

**1.8 Project schedule**

Table (1.2) For the teamwork scheduling

# **Chapter Two: Related Existing Systems**

## **2.1 Introduction**

This chapter includes a brief about the related existing system similar to the same idea that we have in our project, also it includes overall about the problems inside the existing systems and what is the purposed solutions to solve those problems.

## **2.2 Existing Systems**

* **Amazon**

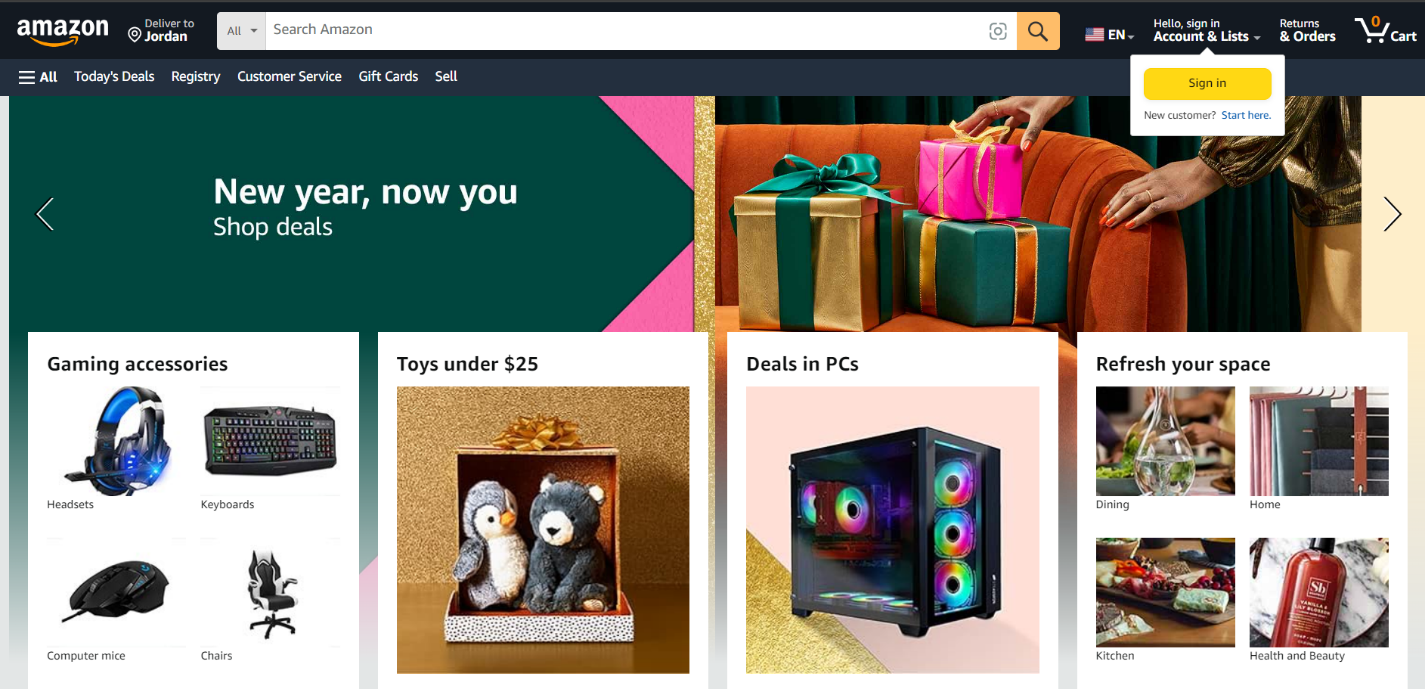
****Amazon, the unparalleled giant in the world of e-commerce, offers an unparalleled shopping experience. Boasting a vast array of products spanning numerous categories, from electronics and fashion to home essentials and beyond, Amazon is a one-stop destination for shoppers worldwide. With its user-friendly interface and intuitive search functionality, finding desired items is effortless. Moreover, the platform's robust customer review system empowers buyers to make informed decisions, ensuring satisfaction with their purchases. Amazon's commitment to exceptional customer service, swift delivery options, and a seamless checkout process further solidifies its position as the go-to online marketplace for millions, redefining the landscape of modern retail through innovation and customer-centricity.

Figure 2.1: Amazon Page

* **Upwork**

**Upwork** is one of the leading freelance platforms, connecting businesses with skilled professionals from around the world. It allows freelancers to create profiles, bid on projects, and collaborate with clients across a variety of industries, including web development, design, and writing. Upwork offers features such as escrow payment systems, time tracking, and dispute resolution, ensuring secure transactions and efficient work processes. However, it mainly focuses on freelance services and does not integrate e-commerce functionalities, unlike the proposed project, which aims to combine freelancing services with product sales in a single unified platform.

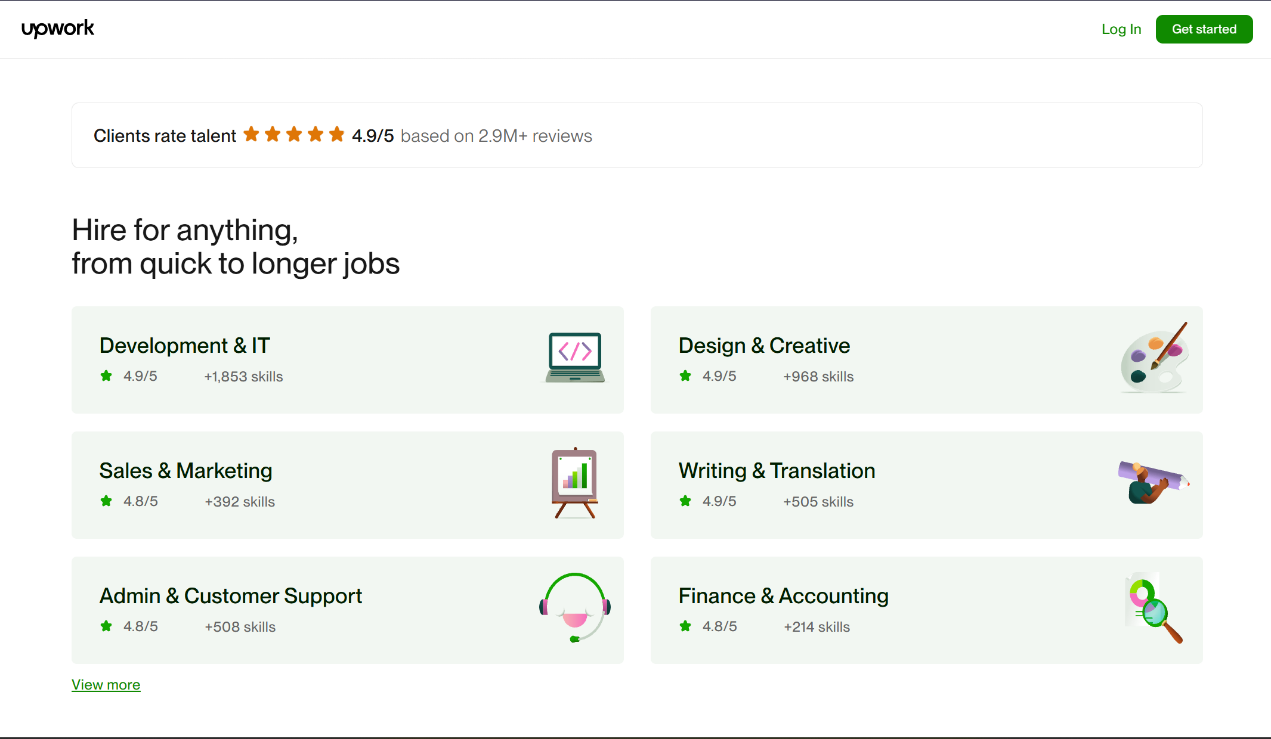


Figure 2.2: Upwork Page

## **2.3 Over all problems of existing Systems**

Amazon has faced criticism regarding antitrust issues and its market dominance. Some experts and regulators argue that Amazon's size and influence in the e-commerce industry might stifle competition and harm smaller businesses. Concerns have been raised about Amazon's practices, such as using its platform to promote its own products over those of third-party sellers or potentially engaging in anti-competitive behavior that limits consumer choice. Critics suggest that Amazon's significant market power could lead to unfair advantages, making it difficult for smaller companies to thrive in the market.

Existing systems like Upwork face several issues, including a fragmented user experience, where freelancing and e-commerce services are separated across different platforms. High transaction fees reduce earnings for freelancers, and limited payment options hinder global transactions. Overcrowded marketplaces make it difficult for new freelancers to stand out, while most platforms focus only on services, leaving out integrated product sales. Additionally, scalability issues arise as platforms grow, affecting performance during high traffic periods.

## **2.4 Over all solution approach**

So, in conclusion we decided to make a website of what people really need and require, especially in our country where we don’t have such services.

## **2.5 Summary**

Finally, this website will be a useful tool for the companies, In which we provide the company it’s breached data and we showed some other existing systems, and clarified how we can be different and better and how the users will benefit from it.

# **Chapter Three: System Requirements Engineering and Analysis**

## **3.1 Introduction**

Requirements Engineering (RE) is defined as the process of establishing the services that the user requires from the system and the constraints that need to be adhered to by the user.

While Requirements Analysis(RA) is the process of determining the requirements of the user and add them to the system and modifying the system in case there was any modification to the site.

## **3.2 Feasibility Study**

Feasibility study determines whether the project is worth doing. Feasibility study is performed to determine the feasibility of a project. The concept and recommendations of such study will be used as sound basis for deciding to proceed, Post pone or cancel the project. In the conduct of feasibility study, We will usually consider following inter-related type of feasibility. They are:

### **3.2.1 Technical Feasibility**

This part is concerned with specifying the hardware and software that is needed to achieve the user requirements, it can be ran on any platform with an internet connection and a web browser or mobile app, To connect to a web server or a local database that is needed to keep the information and use it for the website or mobile app, Also new features might be added and maintained in the future

### **3.2.2 Operational feasibility**

The system will be easy to use as the user interface is GUI based. The system is easy to use so no specific skills will be required to use the system. New users will find it easy to use. So, the project will be operationally feasible.

### **3.2.3 Economic feasibility to the system**

* The economic feasibility of this system is supported by its potential to combine freelance services and e-commerce functionalities within a single platform, addressing the inefficiencies present in current systems. By offering both services and product sales, the platform can attract a wide user base, leading to diversified revenue streams.
* The system's cost-effectiveness comes from its ability to reduce overhead by consolidating operations into one platform, reducing the need for multiple service providers or infrastructure. Transaction fees, although necessary for revenue generation, can be optimized to remain competitive, ensuring both freelancers and clients find value in using the platform. Additionally, the scalability of the platform ensures that as user demand grows, the system can expand without significantly increasing operational costs, making it a sustainable long-term investment.

1. Software:

|  |  |
| --- | --- |
| **Software** | **Specification** |
| Microsoft office | Word, PowerPoint, |
| Visual studio | IDE for ASP.NET |
| VS Code | Code editor for front |
| SQL Server Management studio | For Managing DB |
| Chrome | To run the project |

Table 3.1: Software Tools

## **3.4 Targeted users**

The targeted users of this system include a diverse range of individuals and businesses seeking both freelance services and e-commerce solutions.

1. **Freelancers:**  
   Independent professionals across various industries such as web development, design, writing, marketing, and more, looking to offer their services and connect with potential clients globally.
2. **Small and Medium Enterprises (SMEs):**  
   Businesses in need of flexible workforce solutions and seeking to hire freelancers for specific projects while also requiring a platform to sell products or services online.
3. **Consumers:**  
   Individuals interested in purchasing products from the integrated online store, whether it’s physical goods or digital products, and **those seeking services from freelancers.**
4. **Entrepreneurs and Startups:**  
   New businesses looking for an efficient way to access both freelance services and a marketplace to launch their products, combining both needs in a single platform.

The system aims to cater to users who want a comprehensive, easy-to-use solution for both professional services and e-commerce, reducing the need for multiple platforms.

## **3.5 Functional Requirements Definition and specification**

In this section we will review the functional requirements of the system:

|  |  |  |
| --- | --- | --- |
| **Targeted User** | **Group** | **Requirement Description** |
| Admin | G1 | Create an account. |
| Admin | G1 | Logging in to admin dashboard. |
| Admin | G1 | Edit the profile. |
| Admin | G1 | Manage the user’s accounts. |
| Admin | G1 | Adding a product. |
| Admin | G1 | Deleting a product. |
| Admin | G1 | Update product information. |
| Admin | G1 | Adding users. |
| Admin | G1 | Deleting users. |
| Admin | G1 | Update user’s information. |
| Admin | G1 | Show The Sales-Reports |
| Admin | G1 | Receive Email From The Visitors And Other People |
| Admin | G2 | Logging in to the admin dashboard. |
| Admin | G2 | Adding a product. |
| Admin | G2 | Deleting a product. |
| Admin | G2 | Update product information. |
| Admin | G3 | Logging in to the admin dashboard. |
| Admin | G3 | Show orders |
| Admin | G3 | Show product |
| Visitor | G4 | Create an account. |
| Visitor | G4 | View product. |
| Visitor | G4 | Search product. |
| Visitor | G4 | Add to cart. |
| Visitor | G4 | View/Update cart. |
| Customer | G5 | Logging in. |
| Customer | G5 | View product. |
| Customer | G5 | Add to cart. |
| Customer | G5 | View/Update cart. |
| Customer | G5 | Add to favourite. |
| Customer | G5 | Checkout. |
| Customer | G5 | Search product. |
| Customer | G5 | Logging out. |

Table 3.2: Functional Requirements

## **3.6 Nonfunctional Requirements:**

|  |  |
| --- | --- |
| **Non-Functional Requirements** | **Description** |
| **Usability** | It means ease of use and the website shouldn’t be complex.  We design our website to use it in an effortless way. |
| **Testability** | The website should be able to make testing for its component (software). |
| **Security** | The website should be secure, because it shall allow only the authorized users to access his data. |
| **Maintainability** | The website should be able to update. If we want to add new features on the future. |
| **Performance** | The website shall be designed to be the most accurate possible. (The system should allow users to use the website any time of day). |
| **Flexibility** | The system should be able to be updated easily to suit the changing demands. |

Table 3.3: Nonfunctional Requirements

## **3.7 Summary:**

System Requirements Engineering and Analysis involves the process of gathering, defining, and documenting the functional and non-functional requirements of a system. It is a crucial phase in software development that ensures the system meets user needs and business objectives. The process includes activities such as requirement elicitation, validation, and specification, often through interviews, surveys, and use cases. Non-functional requirements, like performance, security, and scalability, are also analysed to ensure the system's robustness. Effective requirements engineering helps to avoid scope creep, reduce development costs, and ensure that the final product aligns with stakeholders’ expectations.

# **Chapter Four: System Design**

## **4.1 Introduction:**

This chapter includes many important figures that describe our website process, it will include context diagram, data flow diagram (DFD), entity relation diagram (ERD), use cases diagrams, sequences diagrams.

## **4.2 UML Use Case Diagram:**

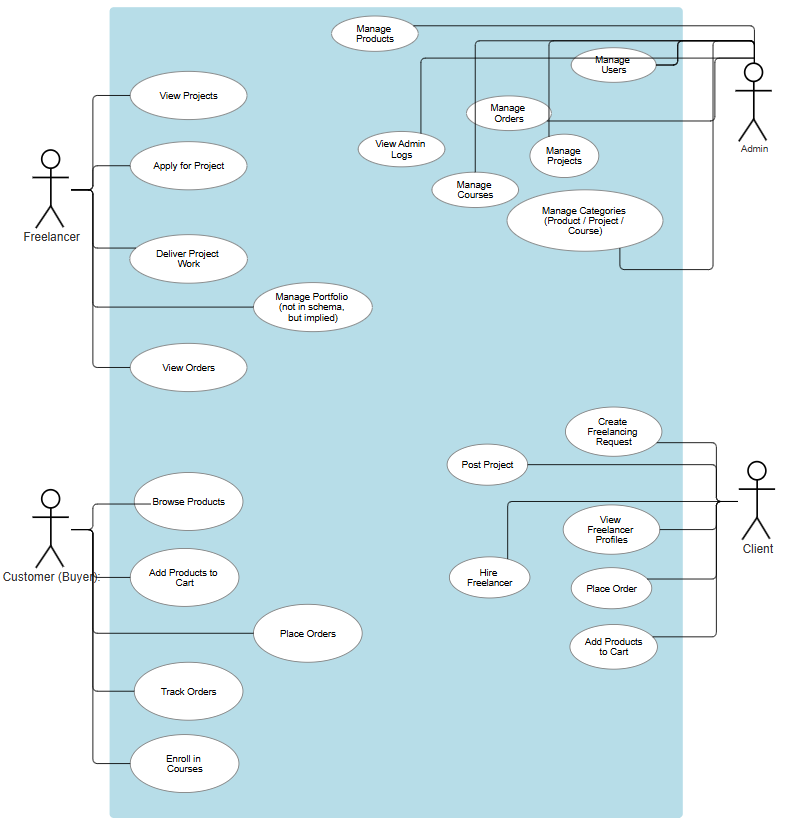
****

Figure 4.1**:** Use case of admin and Customer/Visitor.

**4.3 UML Sequence:**

* **For the Customer/Visitor:**

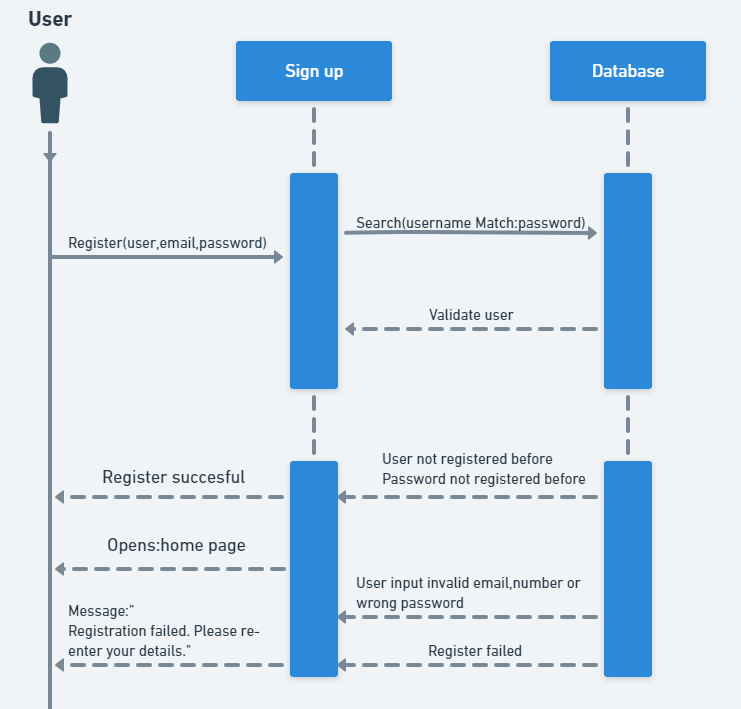
1.Sign up

Figure 4.2: signup

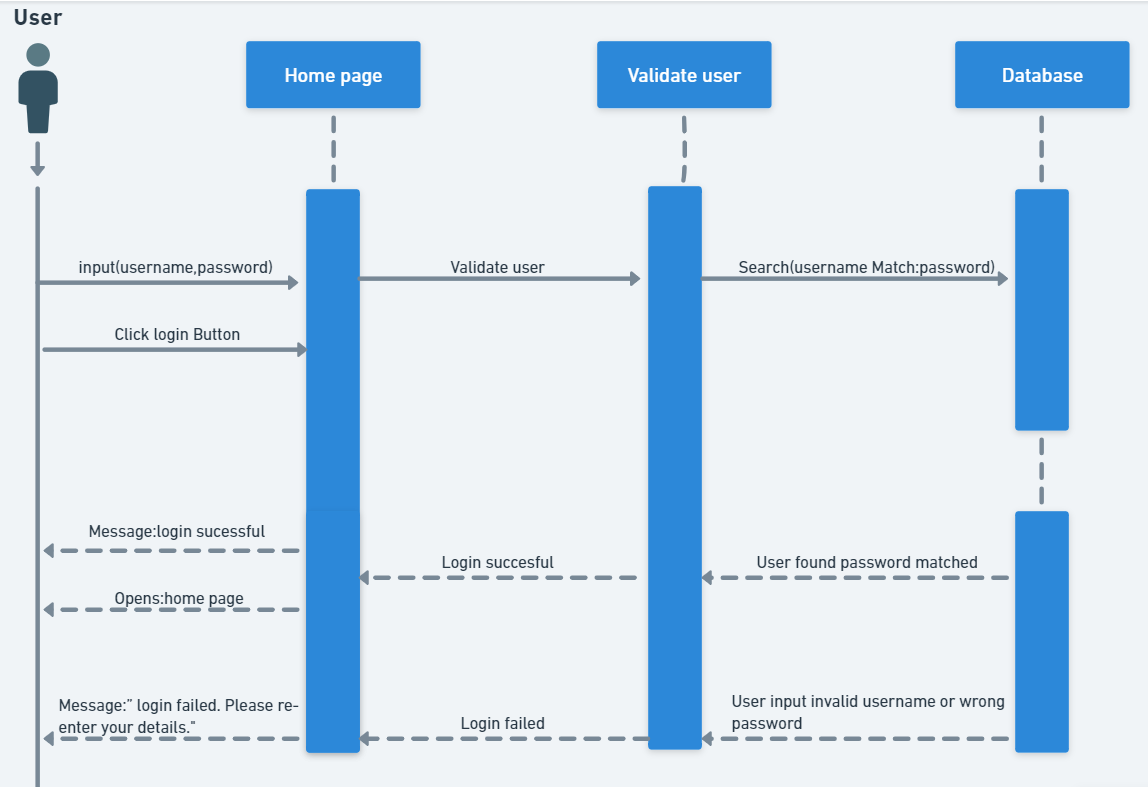
2.Login

Figure 4.3: login

A diagram of a website

Description automatically generated3.Logout

Figure 4.4: logout

A diagram of a product

Description automatically generated4.Search Bar

Figure 4.5: Search Bar

5.About page

A diagram of a website

Description automatically generated

Figure 4.6: About page

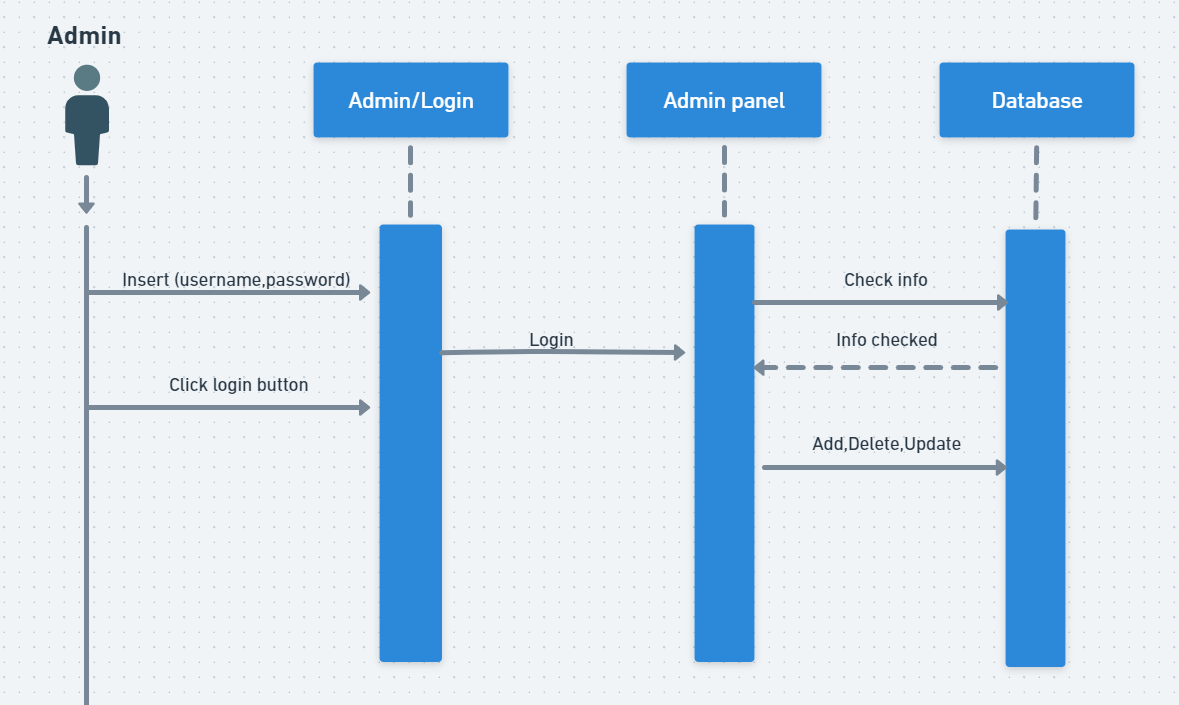
* **For the Admin:**

Figure 4.7: Admin

## **ER diagram (ERD):**

A screenshot of a computer

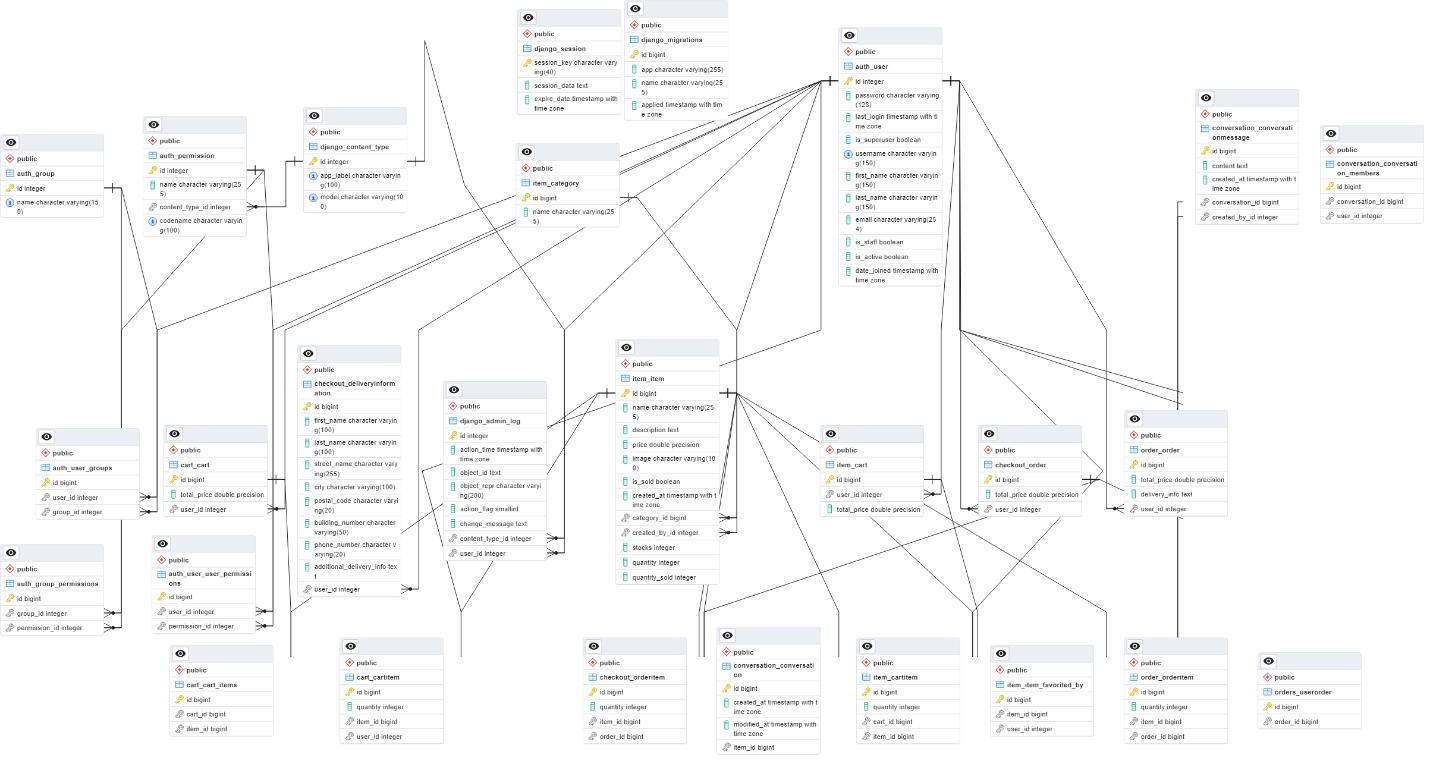
Description automatically generated

Figure 4.8: ERD

# **Chapter Five: System Implementation**

## **5.1 Introduction**

In this chapter, the system implementation is discussed. In section 5.2, we will discuss database implementation while, graphical user interface implementation is presented in section 5.3

## **5.2 Database Implementation**

Database implementation using MySQL involves designing a relational database to store and manage the system's data efficiently.

Key steps include:

1. **Database Design:** Defining tables and relationships based on system requirements.
2. **Table Creation:** Creating tables with appropriate data types and keys (primary, foreign) for data integrity.
3. **Normalization:** Reducing redundancy by normalizing the database into related tables.
4. **Stored Procedures and Triggers:** Implementing procedures and triggers to automate tasks and improve performance.
5. **SQL Queries and Indexing:** Writing efficient SQL queries and applying indexing for faster data retrieval.
6. **Security and Backup:** Implementing security measures and regular backups to protect data.

Using MySQL ensures a reliable, secure, and scalable database system for managing data and transactions efficiently.

.

## **5.3 Graphical User Interface Implementation**

This section offers a comprehensive preview of both the website and mobile app pages, emphasizing their respective functionalities, interactive elements, media content, and thorough functionality testing. Additionally, compatibility across various browsers for the website and multiple platforms (iOS, Android) for the mobile app is evaluated to ensure a seamless user experience across different devices and operating systems.

A sign up form with white squares

Description automatically generated

A black square with white squares and white text

Description automatically generatedFigure 5.1: Registration.

Figure 5.2: Login.

A screenshot of a computer

Description automatically generated

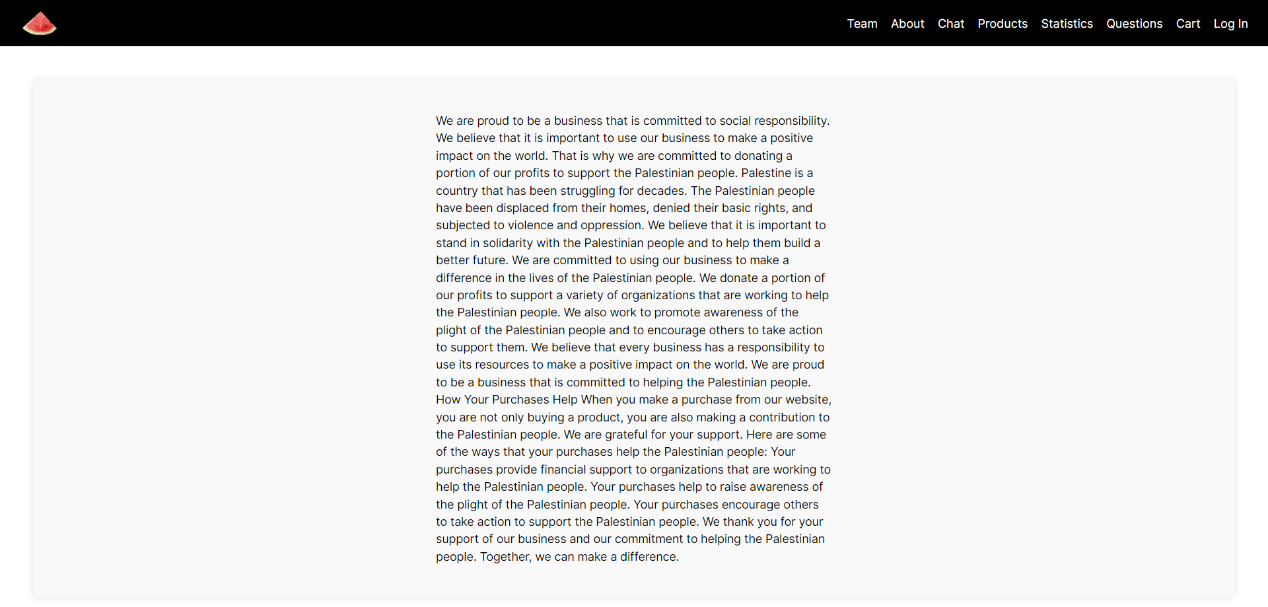
Figure 5.3: Main page.

Figure 5.4: About page

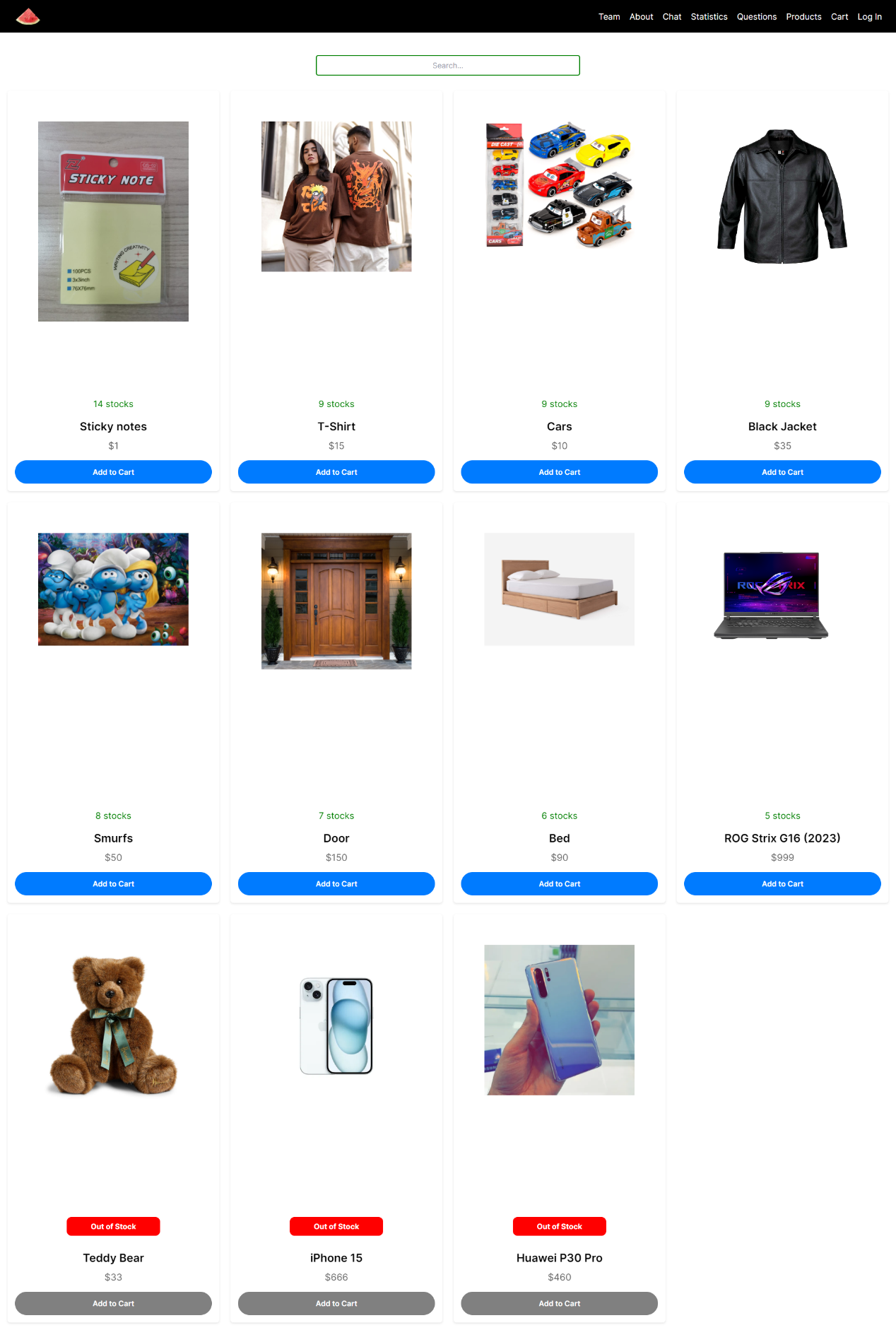


Figure 5.5: Products Page.

A screenshot of a computer

Description automatically generated

Figure 5.6: Team Page.

A screenshot of a computer

Description automatically generated

Figure 5.7: Statistics Page.

A screenshot of a chat

Description automatically generated

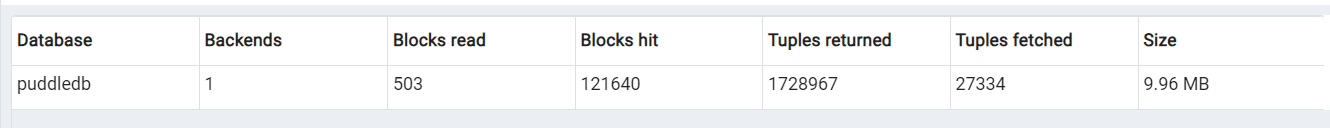
Figure 5.8: Questions.

Figure 5.9: Database Data.

A screenshot of a computer

Description automatically generated

Figure 5.10: Database Statistics.

**A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated Mobile:**

Figure 5.11 & Figure 5.12: Products Page.

**A screenshot of a login form

Description automatically generatedA screenshot of a login form

Description automatically generated** Figure 5.13: Login page. Figure 5.14: Signup page

**A screenshot of a phone

Description automatically generated**

Figure 5.15: Best Seller Pager.

# **Chapter Six: System Testing and Installation**

## **6.1 Introduction**

System Testing and Installation are critical phases in the software development life cycle, ensuring that the system functions as intended and is ready for deployment. **System Testing** involves evaluating the entire system to verify that all components work together seamlessly, meet the specified requirements, and function under various conditions. It includes different testing techniques, such as functional testing, performance testing, security testing, and user acceptance testing, to identify and resolve potential issues before the system is live.

**System Installation** refers to the process of deploying the system on the target hardware or environment. This step involves configuring the system, ensuring that all necessary components (such as databases, software libraries, and network configurations) are set up correctly, and verifying that the system is operational in a production environment. Proper installation is crucial for ensuring smooth system performance and providing users with a stable and reliable platform.

Together, testing and installation ensure that the system is fully functional, secure, and ready for real-world use, ultimately leading to a successful product launch.

## **6.2 Heuristic Evaluation**

Heuristic evaluation is usability experts review your user interface such as lists, combo box, and database connection and compare it against accepted usability principles. This evaluation was performed on our web- based website and mobile app by Two Information Technology (IT) experts who basically examined the interface and judged its compliance with recognized usability principles known as heuristics.

Kindly find the actual Heuristic Evaluation forms of our website included in Appendix D.

Furthermore, Table 6.1 lists the heuristics of usability evaluation and their descriptions.

Table 6.1: Lists of Heuristics for Usability Evaluation and their Descriptions

|  |  |  |
| --- | --- | --- |
| **Numbering Scheme** | **Heuristics** | Description |
| **H1** | **Visibility of system status** | Keep users informed about every step: from item selection to checkout.  Instant updates on stock and delivery status for a seamless shopping experience.  Clear notifications ensure you're always in the know while navigating our site. |
| **H2** | **Match between system and the real world** | The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. |
| **H3** | **User control and freedom** | Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. |
| **H4** | **Consistency and standards** | Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions. |
| **H5** | **Error prevention** | Even better than a good error message is a careful design that prevents a problem from occurring in the first place. |
| **H6** | **Recognition rather than recall** | Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. |
| **H7** | **Flexibility and efficiency of use** | Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. |
| **H8** | **Aesthetic and minimalist design** | Dialogues should not contain information that is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. |
| **H9** | **Help users recognize, diagnose, and recover from errors** | Error messages should be expressed in plain language (no codes), precisely indicating the problem, and constructively suggesting a solution. |
| **H10** | **Help and documentation** | Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. |

## **6.3 Pre-Evaluation Procedures**

The participants were asked to come to a meeting to test the website and in the meeting the participants were taught about the Exposed website, how does it work and what it does.

The participants were given instructions of what tasks they need to do and how they’re going to do it and were told to note any problem that occurred to later discus it and were told that the time they take to do the tasks is recorded.

## **6.4 black box testing**

* Black box testing involves testing a system with no prior knowledge of its internal workings.
* A tester provides an input, and observes the output generated by the system under test.
* Black box testing is a powerful testing technique because it exercises a system end to end.

1. Login test case for user when the user login to website.

Table 6.5: login test case

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Description | Input | Out put |
| 1 | **valid input** in login form:   1. valid username 2. password | Username: Moahmmad554  Password: \*\*\*\*\*\*\*\*\*\* | Users can enter the website freely and browse its content. |
| 2 | User enter Invalid username | Email: Mohammad33@  Password: \*\*\*\*\*\*\*\*\*\* | User can’t login to system “username or password not correct” |
| 3 | User enter incorrect password | Username: Moahmmad554  Password: \*\*\*\*\* | User can’t login to system “username or password not correct” |

2. Create an account test case for the user.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case** | **Description** | **Input** | **Out put** |
| 1 | **valid input** in create account form:   1. username 2. email 3. password 4. confirm password | Username: mohammad55  Email:mohammad2024@gmail.com  Password:\*\*\*\*\*\*\*\*\*\*  confirm password: \*\*\*\*\*\*\*\*\*\* | A new account has been created for the user. |
| 2 | User enter different confirm password | Username: mohammad55  Email:mohammad2024@gmail.com  Password:\*\*\*\*\*\*\*\*\*\*  confirm password: \*\*\*\* | The confirmed password does not match the original one entered. |

Table 6.6: Signup test case

## **6.5 White Box Testing**

White Box Testing, also known as Clear box testing, open box testing, transparent box testing, Code-based testing, or Glass box testing, is a software testing approach where the internal structure, design, and code of software are scrutinized. This method aims to verify the flow of input and output, enhance design, improve usability, and fortify security. Unlike Black Box Testing, which evaluates the software from an external or end-user perspective, White Box Testing involves testers having visibility into the code and delves into the internal workings of an application.

One specific technique within White Box Testing is Control Flow Testing. This method concentrates on determining the sequence of execution of statements or instructions within a program's control structure. The objective is to develop test cases that navigate through the control structure of the program. Testers select a specific segment of a larger program to establish the testing path, frequently applied in unit testing. Test cases are often visualized and represented by the control graph of the program.

## **6.6 Automation Testing**

Test Automation is a software testing technique that involves the utilization of specialized automated testing software tools to execute a suite of test cases. In contrast, Manual Testing is conducted either by a human operating a computer or mobile device, carefully executing the designated test steps. Both computers and mobile devices serve as platforms for executing manual testing procedures, with testers interacting directly with the interface to evaluate software functionality and performance.

## **6.7 GUI Testing**

GUI Testing is a software testing type that focuses on assessing the Graphical User Interface elements within the software application. Its primary goal is to ensure that the functionalities of the software align with the specified requirements by examining screens and controls, such as menus, buttons, icons, and other visual components. Some aspects tested during GUI testing include:

* Verification of element size, position, width, and height.
* Evaluation of displayed error messages.
* Assessment of font readability.
* Checking the alignment of texts, icons, buttons, and other elements.
* Examination of font colors and contrast for readability.
* Testing the colors used for error and warning messages.
* Spelling checks within the interface.
* Evaluation of the interface's visual appeal.
* Assessing the alignment of headings and textual elements.
* Testing the responsiveness of screens in various resolutions by zooming in and out (e.g., 640 x 480, 600 x 800, etc.).
* **6.8 Summary**

Heuristic and Cooperative Evaluations were applied on Exposed website, each of these evaluations has its own method in assessing the overall usability of the website.

# **Chapter seven: Project Conclusion & Future Work**

## **7.1 Introduction**

As we concluded this project, we understand there is no student's perfect project, but we humbly hope our project is good enough for Balqa Applied University(BAU). We did our best using the acquired IT knowledge and the guidance and the patience of our project advisor.

In this chapter, we will present the overall weakness of the system in section 7.2, while in section 7.3 the overall strengths of the system are discussed, future work is illustrated in section 7.4 and finally, chapter summary is presented in section 7.5.

## **7.2** **Overall Weaknesses**

The website’s database is in a server so the user is required a constant connection to the internet, and this might affect the speed of the website so the user might experience some delay if they have a slow internet connection.

But still we tried to do our best to make the website as good as possible.

## **7.3** **Overall strengths**

The project successfully integrates freelance service offerings with e-commerce functionality into a unified platform, providing users with a seamless experience for both work and product transactions. The system’s strength lies in its ability to combine these two core functionalities, eliminating the need for users to manage multiple platforms.

The platform’s intuitive user interface, secure payment system, and scalable architecture ensure that it can meet the evolving needs of freelancers, businesses, and consumers alike. The use of MySQL for robust database management guarantees efficient data handling and integrity, while the system’s focus on performance optimization through features like time tracking and real-time communication enhances user satisfaction.

## **7.4** **Future Works**

We intend to implement VISA payment. implement an Arabic version to help more people to find what they want more easily And in the future we intend to send a notification to the user’s email whenever a breach occurs.

* **Mobile Application Integration:** Develop dedicated mobile applications for iOS and Android to provide users with an optimized experience on mobile devices.
* **AI Integration:** Implement AI-based recommendation engines to suggest relevant services and products to users.
* **Global Expansion:** Expand payment gateway integrations and support for multiple currencies to facilitate global transactions.
* **Advanced Analytics:** Incorporate advanced analytics for freelancers and businesses to track performance metrics and growth.

## **7.5 Summary**

In this chapter, we went through the overall weakness, the overall strengths of the system in addition to, future work to enhance system performance.

**7.6 References and Appendix  
1)ChatGPT**

[**https://chatgpt.com/**](https://chatgpt.com/)

**2) Google :**

[[**https://www.google.jo/**](https://www.google.jo/)](https://www.google.jo/)