

Tribhuvan University

Faculty of Humanities and Social Sciences

Linkonomy: “Website to Hire Professionals”

Project Report

Submitted to

Department of Computer Application

National College of Computer Studies

*In partial fulfillment of the requirements for Bachelor’s in Computer Application*

Submitted By:

Ronak Maharjan

Pranav Sthapit

Kritan Shahi

Under the Supervision of

Sachita Maharjan



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**National College of Computer Studies**

**SUPERVISOR’S RECOMMENDATION**

We hereby recommend that this project prepared under our supervision by **Sachita Maharjan** entitled “**Linkonomy : Website to Hire Professionals**” in partial fulfillment of the requirements for a degree of Bachelor’s in Computer Application is recommended for the final evaluation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SIGNATURE**

**Sachita Maharjan**

**SUPERVISOR**

**Faculty Member**

**Department of Computer Application**

**National College of Computer Studies**

# Abstract

Linkonomy is a service-oriented platform designed to connect users with skilled professionals across various fields in a convenient and efficient manner. The website allows individuals to browse through a curated list of professionals, view their profiles, and hire them based on their needs. Linkonomy supports multiple features including user-friendly navigation, search and filtering options, and professional ratings to help users make informed decisions. Built using React with TypeScript , the platform ensures a responsive and dynamic user experience. By streamlining the hiring process, Linkonomy empowers users to find the right professionals quickly and securely

# Acknowledgement

We would like to express our heartfelt gratitude to everyone who supported us throughout the development of our project, **Linkonomy: Website to Hire Professionals**. We are especially thankful to our project supervisor for their consistent guidance, valuable insights, and encouragement, which greatly contributed to the success of this project. We also extend our thanks to our faculty members and peers for their constructive feedback and technical assistance. Lastly, we are grateful to our families and friends for their continuous support and motivation, which played a crucial role in helping us complete this project as a team.

# Table of Contents

[Abstract ii](#_Toc200988726)

[Acknowledgement iii](#_Toc200988727)

[Table of Contents iv](#_Toc200988728)

[List of Figure vi](#_Toc200988729)

[List of Tables vii](#_Toc200988730)

[Chapter 1: Introduction 1](#_Toc200988731)

[1.1. Introduction 1](#_Toc200988732)

[1.2. Problem Statement 1](#_Toc200988733)

[1.3. Objectives 1](#_Toc200988734)

[1.4. Scope and Limitation 2](#_Toc200988735)

[1.5. Report Organization 3](#_Toc200988736)

[1.5.1. Introduction 3](#_Toc200988737)

[1.5.2. Background Study 3](#_Toc200988738)

[1.5.3. System Analysis and Design 3](#_Toc200988739)

[1.5.4. Implementation and Testing 3](#_Toc200988740)

[1.5.5. Conclusion and Future Recommendations 3](#_Toc200988741)

[Chapter 2: Background Study 4](#_Toc200988742)

[2.1. Background Study 4](#_Toc200988743)

[2.1.1. Study of existing system 4](#_Toc200988744)

[2.1.2. Literature Review 6](#_Toc200988745)

[Chapter 3: System Analysis and Design 8](#_Toc200988746)

[3.1. System Analysis 8](#_Toc200988747)

[3.1.1. Requirement Identification 9](#_Toc200988748)

[3.1.1.1. Functional Requirements 9](#_Toc200988749)

[3.1.1.2. Non-Functional Requirements 11](#_Toc200988750)

[3.1.2. Feasibility Study 12](#_Toc200988751)

[3.1.2.1. Technical Feasibility 12](#_Toc200988752)

[3.1.2.2. Operational Feasibility 12](#_Toc200988753)

[3.1.2.3. Economic Feasibility 13](#_Toc200988754)

[3.1.3. Data Modeling 14](#_Toc200988755)

[3.1.4. Process Modeling 16](#_Toc200988756)

[3.1.6. Sequence Diagram 18](#_Toc200988757)

[3.1.7. Activity Diagram 19](#_Toc200988758)

[3.2. System Design 20](#_Toc200988759)

[3.2.1. Architectural Design 20](#_Toc200988760)

[3.2.2. Database Schema Design 21](#_Toc200988761)

[Chapter 4: Implementation and Testing 24](#_Toc200988762)

[4.1. Implementation 24](#_Toc200988763)

[4.1.1. Tools used: 24](#_Toc200988764)

[4.1.2. Implementation Detail of Module: 24](#_Toc200988765)

[4.2. Testing 24](#_Toc200988766)

[4.2.1. Test Case 25](#_Toc200988767)

[Chapter 5: Conclusion and Future Recommendation 30](#_Toc200988768)

[5.1. Lesson Learnt/Outcome 30](#_Toc200988769)

[5.2. Conclusion 30](#_Toc200988770)

[5.3. Future Recommendations 30](#_Toc200988771)

[References 31](#_Toc200988772)

# List of Figure

[Figure 1: Prototyping Model 1](#_Toc200871670)

[Figure 2: Use Case Diagram of Linkonomy 1](#_Toc200871671)

[Figure 3: E-R Diagram of Linkonomy 1](#_Toc200871672)

[Figure 4: Level 1 Data Flow Diagram of Linkonomy 1](#_Toc200871673)

[Figure 5: Sequence Diagram of Linkonomy 1](#_Toc200871674)

[Figure 6: Activity Diagram of Linkonomy 1](#_Toc200871675)

[Figure 7: Three-Tire Architecture Diagram of Linkonomy 1](#_Toc200871676)

[Figure 8: Database Schema of Linkonomy 1](#_Toc200871677)

# List of Tables

[Table 1: Test Case 1 1](#_Toc200836953)

[Table 2: Test Case 2 1](#_Toc200836954)

[Table 3: Test Case 3 1](#_Toc200836955)

[Table 4: Test case 4 1](#_Toc200836956)

[Table 5: Test Case 5 1](#_Toc200836957)

# Chapter 1: Introduction

## 1.1. Introduction

The internet has significantly transformed how services are accessed and delivered. With the rise of online platforms, individuals can now connect with skilled professionals and service providers without the limitations of time or location. This trend has reshaped the traditional service industry by offering faster, more transparent, and more efficient ways to hire professionals.

**Linkonomy** is an online platform created to bridge the gap between users and skilled professionals across various domains. It allows users to browse through a curated list of professionals, view their profiles, and hire them based on their specific needs. Built using TypeScript and React, Linkonomy offers a responsive and intuitive interface with features like search filters, user reviews, and secure login systems. The platform simplifies the process of finding and hiring reliable professionals by enabling users to explore available service providers from the comfort of their homes.

## 1.2. Problem Statement

Many individuals struggle to find qualified professionals for tasks such as graphic design, tutoring, tech support, home repairs, and more. Traditional methods like personal referrals, flyers, or word-of-mouth are often unreliable, time-consuming, and limited by geographic constraints. Moreover, users often lack access to detailed information about professionals, such as their experience, ratings, or portfolio, making it difficult to make informed decisions. There is also the challenge of trust and security when dealing with unknown individuals for professional services.

The aim of the **Linkonomy** platform is to address these challenges by providing a centralized, user-friendly, and secure system where users can easily find and hire verified professionals.

## 1.3. Objectives

The main objectives of **Linkonomy** are:

* To provide users with a reliable platform to browse, compare, and hire professionals from various fields.
* To enable professionals to showcase their profiles, skills, and past work to attract potential clients.
* To ensure a secure and streamlined experience through features such as user registration, profile ratings, and service categorization.
* To improve accessibility by allowing users to search and filter professionals based on expertise.

## 1.4. Scope and Limitation

**Scope of Linkonomy:**

* **Diverse Professional Categories:** The platform allows listing and browsing of professionals from various sectors such as IT, education, home services, design, and more.
* **User-Friendly Interface:** Designed using TypeScript and React, the site offers a smooth and responsive browsing experience.
* **Filtering and Search Options:** Users can apply filters to find professionals based on category, experience level, and ratings.
* **Secure Account Management:** Registered users can create profiles, log in, view and connect with professionals directly.
* **Professional Profiles:** Each professional has a dedicated profile showcasing their skills, experience, and client reviews.

**Limitations of Linkonomy:**

* **Limited Geographic Reach:** Initially, the platform may only support certain regions, affecting both users and professionals outside the coverage area.
* **No Built-in Payment Integration (Initial Phase):** In the early stages, users may have to rely on external communication for payment and hiring processes.
* **Verification Challenges:** Although the platform may allow professionals to register freely, fully verifying the legitimacy of each profile may be time-consuming.
* **Internet Dependency:** Like most web-based platforms, Linkonomy’s functionality is dependent on stable internet access, which may hinder users in low-connectivity areas.

## 

## 1.5. Report Organization

### 1.5.1. Introduction

This chapter introduces the project, the problem it aims to solve, and outlines the objectives, scope, and limitations of the proposed solution.

### 1.5.2. Background Study

This chapter explores the current methods and platforms used for hiring professionals. It includes a review of similar systems and their limitations, which inspired the development of Linkonomy.

### 1.5.3. System Analysis and Design

This chapter covers the project's requirements collecting, feasibility assessment, and design. Diagrams, functionality analysis, a technique for obtaining requirements, and a process model are all included.

### 1.5.4. Implementation and Testing

Here, the tools, technologies, and methodologies used for developing the project are discussed. It also explains the testing procedures followed to ensure the system’s reliability and performance.

### 1.5.5. Conclusion and Future Recommendations

This chapter summarizes the key takeaways from the project and outlines suggestions for future improvements, such as adding in-app payments, advanced filtering, and expanding the geographical reach.

# 

# Chapter 2: Background Study

## 2.1. Background Study

This chapter outlines the research and study conducted to understand existing professional service platforms, relevant technologies, and methodologies used to develop **Linkonomy**, a website for hiring professionals. Our focus was to analyze current digital solutions that connect users with skilled individuals and identify key features that contribute to a seamless and reliable hiring experience. The background study includes an overview of established platforms, technical frameworks, and security standards, with the goal of building a feature-rich and user-friendly professional services portal.

Through our study, we explored platforms like **Upwork**, **Fiverr**, and **UrbanClap**, as well as development tools such as **React**, **TypeScript**, and **PortgreSQL**. We also examined how these platforms handle user trust, profile verification, and real-time availability. The findings of this study guided us in selecting technologies and designing features that are relevant to our platform's objective of connecting users with trusted professionals.

### 2.1.1. Study of existing system

To understand how professional service platforms operate, we studied and analyzed various existing systems, with **Upwork** and **UrbanClap (now Urban Company)** being key references for the development of Linkonomy.

**Upwork** is one of the world’s largest freelancing platforms, offering businesses and individuals access to a wide network of remote professionals across various fields. It provides features like profile creation, client reviews, job categorization, and secure contracts. One of its standout characteristics is the implementation of a **robust verification and rating system** that ensures trust between clients and freelancers. Furthermore, Upwork supports real-time chat, milestone-based payments, and detailed filtering options, which we found essential for a seamless user experience [1].

On the other hand, **UrbanClap (Urban Company)** is a regionally popular platform offering home-based services like cleaning, beauty, appliance repair, etc. It focuses on location-based services and allows users to schedule appointments with vetted professionals. The strength of UrbanClap lies in its simplified UI/UX design and its real-time availability check, both of which we found useful for designing Linkonomy’s layout and scheduling logic [2].

The features we adopted from these platforms include:

* **Search and filtering options** for faster navigation.
* **Professional profile pages** with experience, ratings, and portfolios.
* **User registration system** for secure login and service history tracking.

These studies provided a strong foundation for determining the must-have features of Linkonomy and guided us in creating a platform that prioritizes **trust**, **ease of use**, and **professional discovery**.

### 2.1.2. Literature Review

In our literature review, we examined various sources related to the use of **React**, **TypeScript**, and **PostgreSQL** as part of the tech stack for building scalable and secure web applications. Our research focused on the capabilities, advantages, and integration of these technologies in the development of a professional hiring platform like *Linkonomy*.

**React** is an open-source JavaScript library maintained by Facebook, widely used for building interactive user interfaces. It offers a component-based architecture and virtual DOM, which improves rendering performance and enables code reusability. According to *React Official Documentation* [3], React’s efficient update mechanism ensures high performance even in complex UIs, making it ideal for applications that require fast interactions and dynamic content such as user profiles and listings.

**TypeScript**, a statically typed superset of JavaScript, enhances code reliability and developer experience. As stated by Microsoft [4], TypeScript allows developers to catch errors during development rather than at runtime, which helps improve code quality. It is widely adopted in large-scale applications due to its support for type checking, object-oriented programming, and tooling integration with modern editors.

For data storage and retrieval, we have chosen **PostgreSQL**, a powerful open-source relational database system. PostgreSQL supports ACID compliance, complex queries, JSON support, and indexing mechanisms, making it suitable for handling structured data such as user profiles, service categories, bookings, and reviews. According to the PostgreSQL documentation [5], its reliability and robustness make it a top choice for enterprise-level applications. Unlike NoSQL alternatives, PostgreSQL supports strict data integrity and relationships through foreign keys, which is essential for maintaining connections between users, services, and transactions.

In comparison to platforms like **Upwork** and **Fiverr**, which are leading services in the freelance and professional hiring domain, *Linkonomy* aims to implement similar trust mechanisms, such as verified profiles and review systems, while offering a simplified and user-friendly experience. Research from online reviews highlights the importance of clear categorization, easy search/filter features, and real-time notifications, all of which we aim to include in our system.

In conclusion, our literature review highlights the importance of using reliable front-end technologies (React + TypeScript) and robust back-end solutions (PostgreSQL) to build a seamless, secure, and scalable web application. These tools, combined with modern UI/UX practices and proper data modeling, form the technical foundation for the successful implementation of *Linkonomy*.

# Chapter 3: System Analysis and Design

## 3.1. System Analysis

For the development of the Linkonomy platform, the Prototyping Model was adopted as the software development methodology. This model emphasizes iterative development and user feedback, which proved highly effective for a user-centric platform like Linkonomy. In the initial phase, a basic prototype of the user interface was designed to visualize core functionalities such as job posting, and application submission. This allowed early validation of the design and usability which made it easier to identify missing requirements

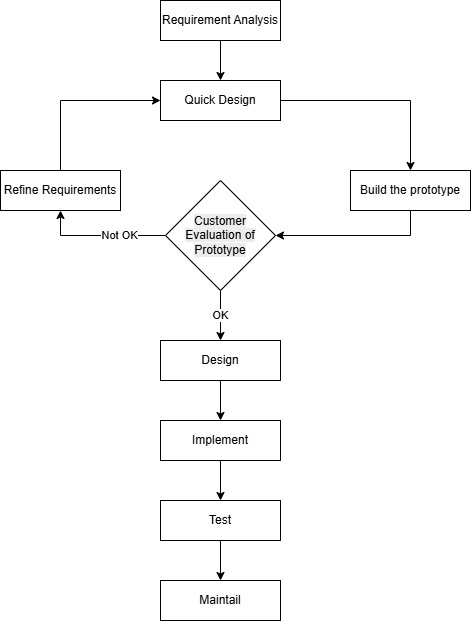


Figure : Prototyping Model

The iterative nature of the Prototyping Model ensured that feedback collected from initial reviews could be incorporated into the subsequent versions of the system. Each prototype refinement included enhancements to user registration, job application handling, and administrative controls. This approach minimized the chances of requirement misunderstanding and enabled more efficient communication between the development team and the end users. Ultimately, the Prototyping Model facilitated the creation of a platform that was both functionally accurate and user-friendly, aligning well with the objectives of Linkonomy.

### 3.1.1. Requirement Identification

Requirements of the system are identified through personal research of visiting various ecommerce sites.

#### 3.1.1.1. Functional Requirements

Functional requirements for Linkonomy :

1. User Registration and Login **-** Users (clients and professionals) should be able to create an account and log in securely to access and utilize the features of the platform.
2. Profile Management **-** Professionals should be able to create and manage detailed profiles including skills, experience, portfolio, availability, and pricing. Clients should be able to edit their profiles and manage preferences.
3. Search and Filter Professionals **-** Clients should be able to browse and search for professionals based on categories, skills, ratings and price.
4. Job Posting and Management **-** Clients should be able to post job requirements or service requests with details like description, budget, timeline, and preferred skills. They should also be able to manage or cancel these postings.
5. Hiring Process **-** Can review applicants and hire the most suitable professional based on profiles, ratings, and reviews.
6. Responsive Design **-** The platform should be responsive and fully compatible across all modern devices and screen sizes (desktop, tablet, and mobile).
7. Security and Privacy **-** The platform should implement authentication, encryption, and data protection measures to ensure the security and privacy of all users’ personal and financial data.
8. User Dashboard **-** Both clients and professionals should have a dedicated dashboard to track jobs, payments, applications, messages, and notifications.

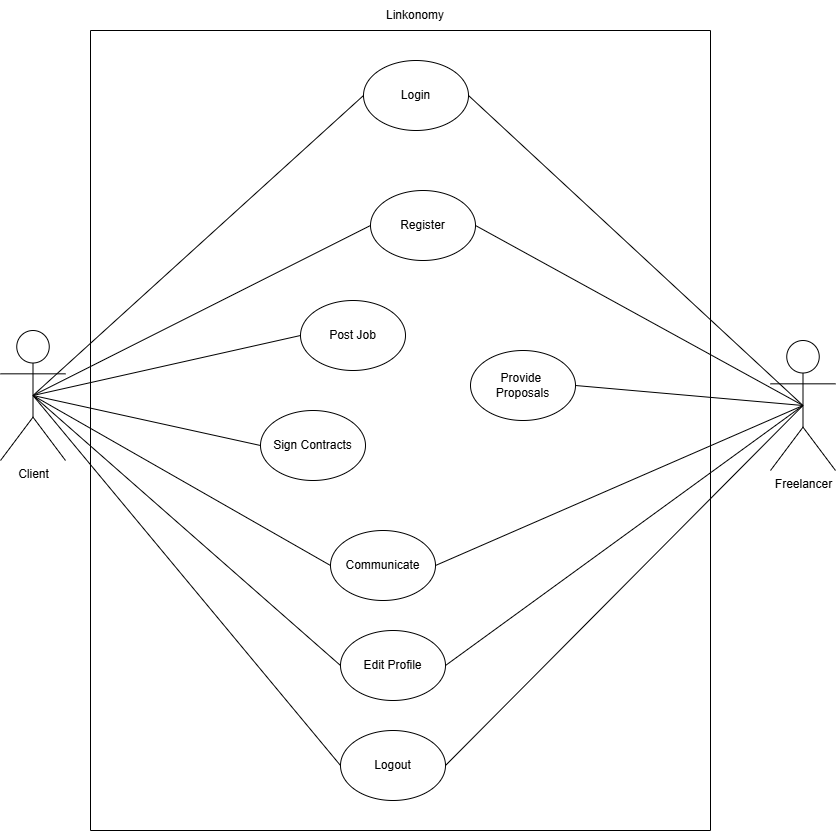


Figure : Use Case Diagram of Linkonomy

The use case diagram for Kapi e-commerce platform represents the functionalities and user interactions of the system. It consists of actors such as Customers and Administrators and use cases that describe the different actions that can be performed by the actors, such as searching for products, adding products to cart, checking out, making payments, viewing order history, registering, logging in, and managing copies. The use cases are connected to the actors through relationships, such as a customer being able to search and purchase products, an administrator being able to manage the inventory of copies. The diagram also includes boundaries that define the scope of the system, and extension points to indicate that a specific use case can have additional functionality in certain conditions. The use case diagram effectively conveys the different interactions and functionalities provided by the system and helps to understand the system's behavior and user interactions.

#### 3.1.1.2. Non-Functional Requirements

Non-Functional requirements for Linkonomy :

1. Performance - The platform should be capable of handling multiple concurrent users, especially during peak hours when professionals and clients are interacting frequently.
2. Scalability - The system should be scalable to accommodate future growth in user base, job postings, and service categories.
3. Availability - The platform should maintain high availability, ensuring minimal downtime and reliable access to users around the clock.
4. Compliance - The system must adhere to legal and regulatory requirements related to data privacy, employment, and digital transactions.
5. Usability - The interface should be user-friendly and intuitive for both clients and professionals with minimal learning curve.
6. Mobile Compatibility - The platform must be fully responsive and optimized for use on smartphones and tablets.
7. User Experience - The platform should provide a smooth and efficient user experience, including fast load times, simple navigation, and clear visual design.

### 3.1.2. Feasibility Study

A feasibility study evaluates whether the proposed Linkonomy platform is practical and viable in terms of technology, operations, and finances. It helps stakeholders determine whether the project should proceed and identifies any potential risks or alternative approaches.

#### 3.1.2.1. Technical Feasibility

Technical feasibility examines whether the project can be successfully developed using existing technology and resources. For Linkonomy, the platform is designed to connect clients with professional service providers via a web interface.

The platform will be built using modern web technologies such as TypeScript and React for the frontend, and PostgreSQL for the backend database. PostgreSQL offers robust data handling capabilities, supports ACID compliance, and is scalable to meet the needs of the platform.

The hosting and deployment environment will support common server configurations such as Node.js runtime, PostgreSQL database server, and Nginx or Apache as the web server. Integration of secure payment gateways such as Khalti or eSewa will also be supported.

The development team has adequate experience in full-stack development, UI/UX design, and secure API integration, which ensures a high level of confidence in successfully building and maintaining the platform.

Therefore, the Linkonomy platform has strong technical feasibility and can be implemented with the current technology stack and available resources.

#### 3.1.2.2. Operational Feasibility

Operational feasibility evaluates whether the platform can be adopted and integrated into real-world use by the target audience.

The platform targets clients looking to hire professionals across various service categories (e.g., graphic design, tutoring, plumbing), as well as skilled professionals seeking work opportunities. The service is expected to fill a significant gap in the market by providing a centralized platform for such interactions.

The operational flow of posting a job, applying, hiring, payment, and reviewing will be streamlined through intuitive workflows and real-time communication features. The platform can be integrated with external calendar APIs and payment systems for scheduling and transactions.

A dedicated support and development team will manage platform operations, monitor for issues, and push updates based on user feedback and market trends. Customer support channels like FAQs, live chat, and ticketing will ensure that operational problems are addressed promptly.

Given its alignment with user needs and support infrastructure, Linkonomy has strong operational feasibility.

#### 3.1.2.3. Economic Feasibility

Economic feasibility examines whether the benefits of the project justify its costs.

The Linkonomy platform is designed to be cost-effective by utilizing open-source tools and frameworks. While development may require investment in custom features and secure infrastructure, the overall cost is controlled through efficient technology choices and streamlined architecture.

Initial costs will include domain registration, web hosting, secure payment gateway integration, and development effort. Ongoing operational costs include server maintenance, customer support, and promotional activities.

Revenue will be generated via platform commissions on transactions, featured listings for professionals, and potential subscription models for premium services.

Considering the demand for an organized platform to hire professionals and the scalability of the business model, the revenue generated is expected to outweigh the initial and operational costs.

Hence, Linkonomy is economically feasible and is expected to be financially sustainable in the long run.

### 3.1.3. Data Modeling

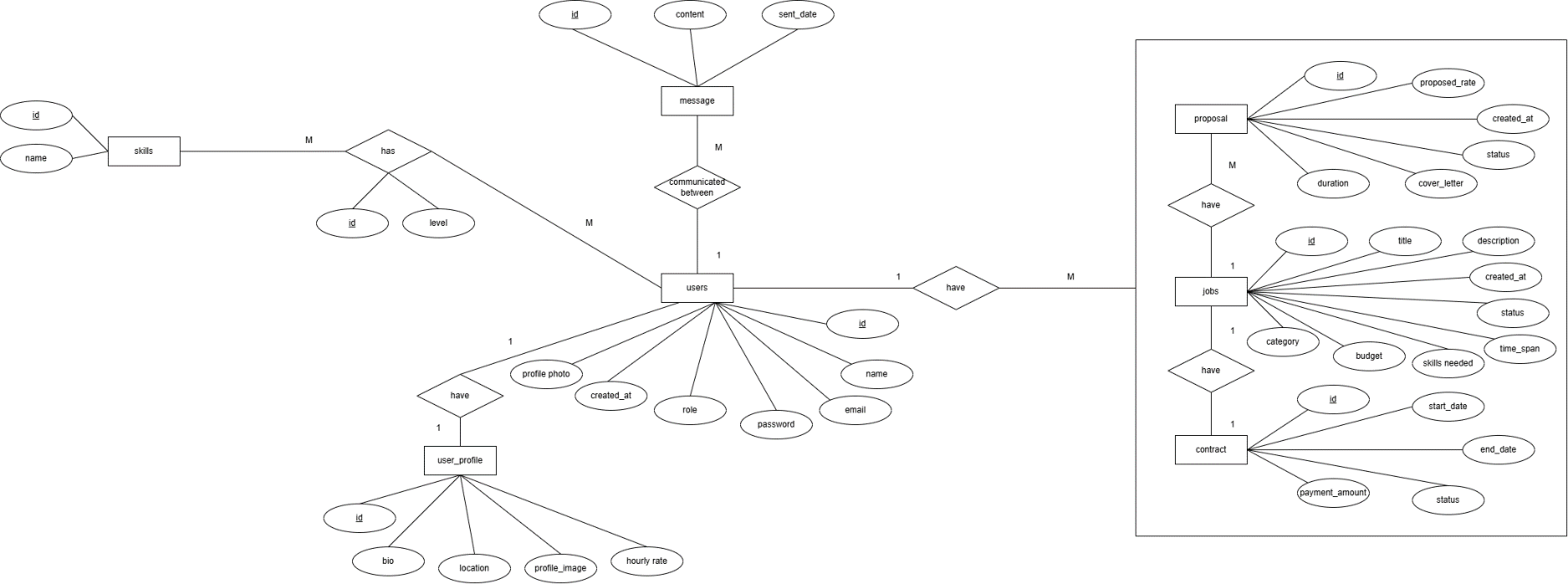


Figure : E-R Diagram of Linkonomy

The Entity-Relationship (ER) Diagram of Linkonomy presents a comprehensive overview of the entities and their interrelations that make up the core functionality of the platform. The key entities identified are: User, Professional, JobPost, Application, Review, and Category.

* User: Represents any individual who registers on the platform, either as a client or a professional.
* Professional: A specialized role associated with a user that offers services on the platform. Professionals can create and update their service profiles.
* JobPost: Represents a task or requirement posted by a client. Each JobPost is linked to one user (the client) and can receive multiple applications.
* Application: Denotes the proposal submitted by a professional in response to a JobPost. It includes status indicators such as pending, accepted, or rejected.
* Review: Represents client feedback for a completed job. Each review is linked to one JobPost and one Professional.
* Category: Used to classify JobPosts and Professional services into organized service types.

Relationships:

* A user can create multiple job posts.
* A job post can receive multiple applications from different professionals.
* Each application is submitted by one professional.
* A review is submitted by a client after job completion and is associated with a specific professional and job post.
* Professionals and job posts are associated with categories.
* Foreign key constraints are established to maintain data integrity, ensuring smooth and secure data linkage across the platform.

Foreign key constraints are established to maintain data integrity, ensuring smooth and secure data linkage across the platform.

### 3.1.4. Process Modeling

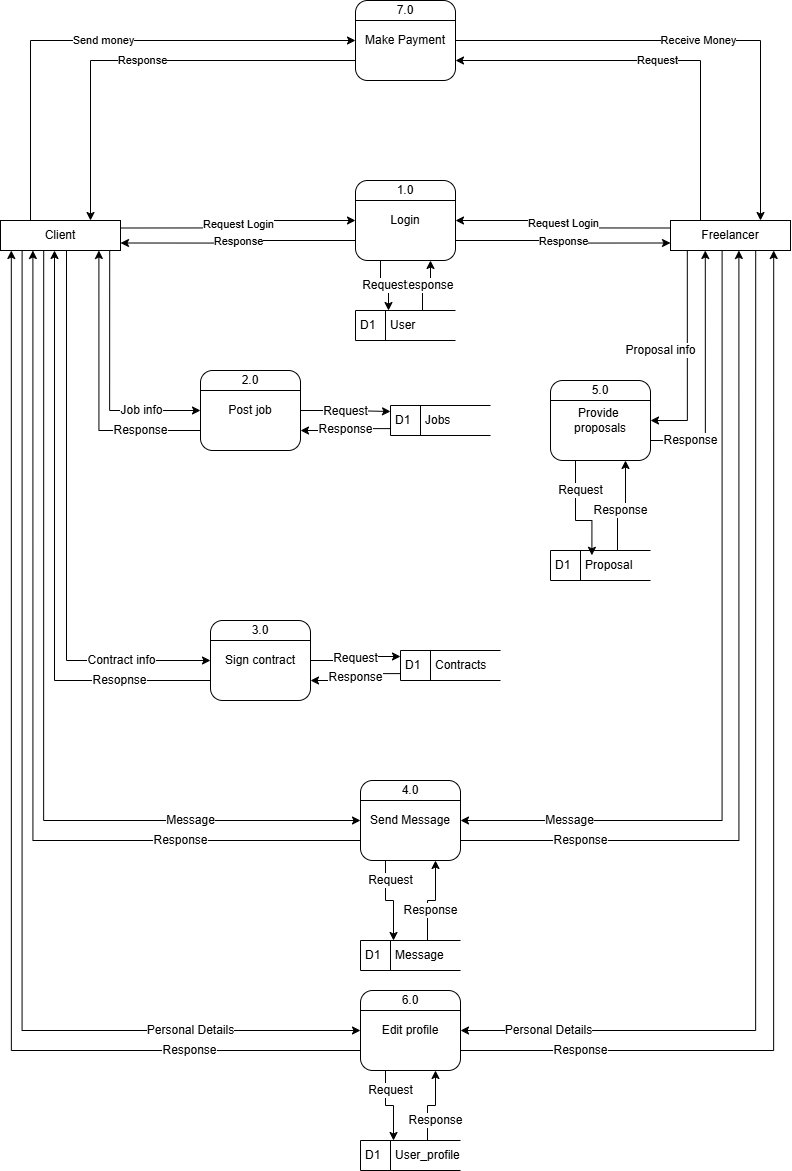


Figure : Level 1 Data Flow Diagram of Linkonomy

The Level 1 Data Flow Diagram (DFD) provides a high-level visualization of how data moves through the Linkonomy system. It outlines the essential processes, external entities, and data stores.

External Entities:

* Client/User: Initiates actions like registration, job posting, and hiring.
* Professional: Views and applies for job posts.
* Admin: Manages platform-wide operations like user management, category updates, and handling disputes.

Processes:

* User Registration and Login: Manages authentication and stores user credentials.
* Job Posting: Allows clients to create job listings.
* Professional Application Submission: Enables professionals to apply for posted jobs.
* Job Review: Lets clients submit reviews after service completion.

Data Stores:

* User Data: Stores credentials, roles, and profiles.
* Job Posts: Stores all job listings.
* Applications: Contains application data submitted by professionals.
* Reviews: Stores feedback and ratings.

The DFD highlights the flow of information between processes and data stores, providing a clear understanding of system functionality.

### 3.1.6. Sequence Diagram

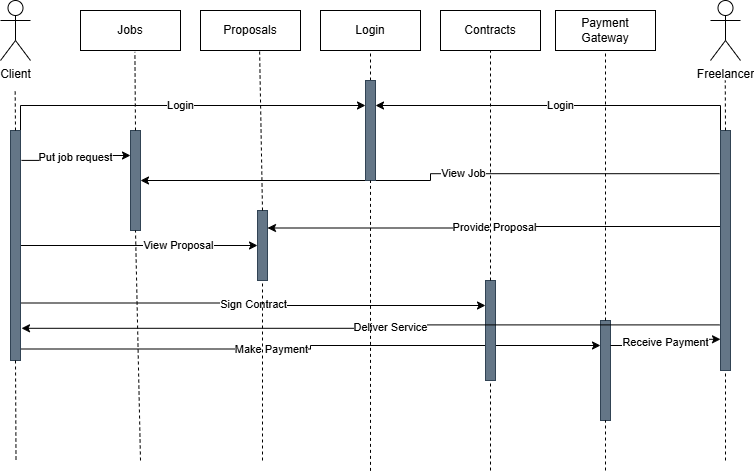


Figure : Sequence Diagram of Linkonomy

The sequence diagram for Linkonomy illustrates the chronological interaction between the system components: Client, Professional, Linkonomy System, and Admin.

Flow of Interaction:

1. The client logs in and posts a new job requirement.
2. The system saves the job post and makes it available for professionals.
3. Interested professionals log in and apply to the job.
4. The client views the list of applications and selects a professional.
5. The system updates the application status and sends notifications to both parties.
6. After the job is completed, the client provides a review.
7. Admin oversees the entire process and handles exceptions or disputes if raised.

This diagram captures the real-time communication and the sequence of operations that support the job hiring workflow on Linkonomy.

### 3.1.7. Activity Diagram

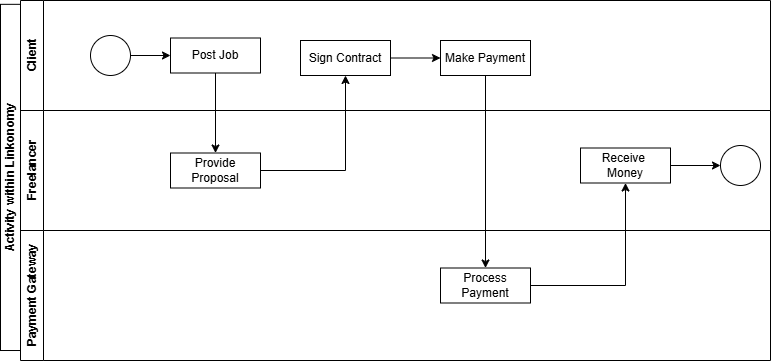


Figure : Activity Diagram of Linkonomy

The activity diagram represents the dynamic workflow involved in hiring a professional using the Linkonomy platform. It maps out all major user actions and system responses in a step-by-step manner.

Key Activities:

* User Login: Begins the process for both clients and professionals.
* Client Actions: Post job > Wait for applications > Review and hire professional > Confirm job completion > Submit review.
* Professional Actions: Browse jobs > Submit application > Wait for response > Deliver service.
* Admin Actions: Monitor platform activity, manage users, handle reported issues.

Decision Points:

* Whether the job post is complete before submission.
* Whether the professional application meets client requirements.
* Whether a job is marked as completed for review submission.

The activity diagram provides a comprehensive look at how different users interact with the system, helping stakeholders understand operational flow and decision-making paths within Linkonomy.

## 3.2. System Design

### 3.2.1. Architectural Design

The three-tier architecture has been used for this project. This architecture is a client- server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

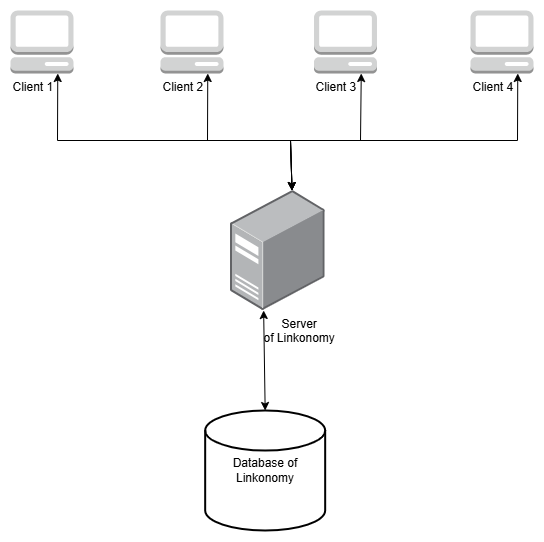


Figure : Three-Tire Architecture Diagram of Linkonomy

Three-Tier Architecture has the following layers.

* **Presentation Layer:** This is the topmost layer and serves as the interface between the user and the system. Built using React and TypeScript, it includes the web pages and user interface components that allow clients and professionals to navigate, post jobs, apply, and communicate. It collects input from the user and displays processed results received from the application layer.
* **Application Layer:** Also known as the business logic or middle tier, this layer handles all the logic and computation required by the system. It is responsible for processing user input, performing validations, managing user sessions, job applications, notifications, and routing data to and from the database. Technologies like Node.js and Express are used here.
* **Data Layer:** This layer consists of the PostgreSQL database server. It is responsible for storing persistent data such as user credentials, professional profiles, job posts, applications, reviews, and categories. It handles query execution, indexing, and ensures secure access to stored data.

This architecture provides several advantages such as modularity, scalability, and maintainability. It also enhances security by restricting direct access to the database and centralizing the application logic in a manageable layer.

### 3.2.2. Database Schema Design

The database schema diagram represents the relational structure of the freelancing web application. It captures the key entities, their attributes, and relationships essential to managing users, jobs, proposals, and contracts within the platform.

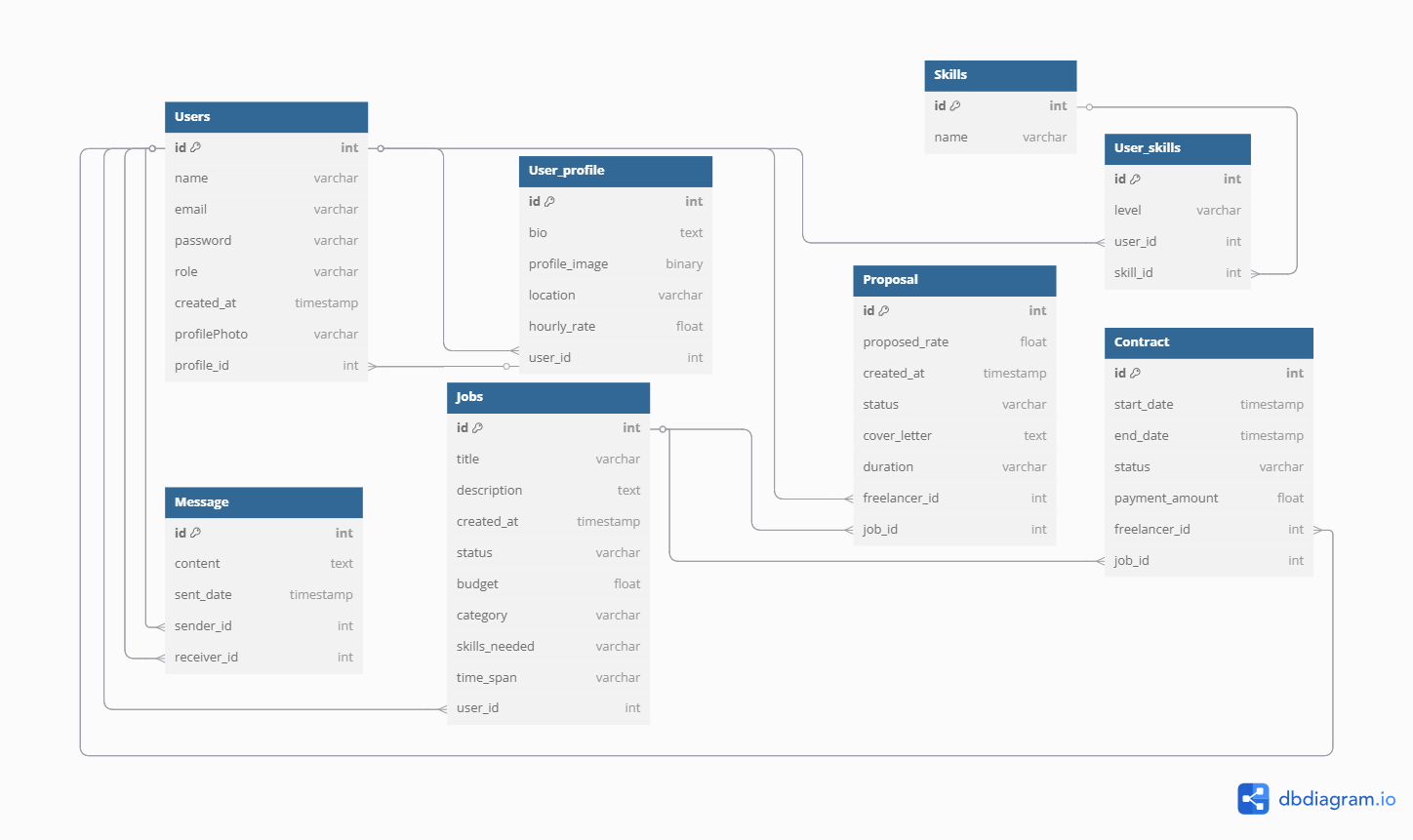


Figure : Database Schema of Linkonomy

**Key Entities and Relationships**

1. **Users**

* Central table storing user information including name, email, password, role, and creation timestamp.
* Users can have different roles: client, freelancer, admin, or both.

1. **User Profile**

* One-to-one relationship with Users.
* Stores additional freelancer-specific data such as bio, location, hourly rate, and profile image.

1. **Skills & User Skills**

* Skills defines available skill names.
* User\_skills is a many-to-many join table linking users to skills with an associated proficiency level (beginner, intermediate, advance).

1. **Messages**

* Enables direct communication between users.
* Each message has a sender and receiver (both referencing Users).

1. **Jobs**

* Created by clients to describe tasks or projects.
* Includes title, description, budget, required skills, and status (open, in\_progress, completed).

1. **Proposals**

* Freelancers apply for jobs by submitting proposals.
* Contains proposed rate, duration, cover letter, and proposal status (pending, accepted, rejected).
* A unique constraint ensures a freelancer can submit only one proposal per job.

1. **Contracts**

* Formed once a proposal is accepted.
* Ties a freelancer to a job and includes payment amount, status (active, completed, cancelled), and work duration.
* Each job can only have one contract.

1. **Reviews**

* Submitted after contract completion.
* Users (typically clients) can rate and comment on the freelancer’s performance.

**Normalization & Integrity**

* The schema follows 3rd Normal Form (3NF) to reduce redundancy.
* Strong use of foreign keys ensures referential integrity between related tables.

This schema provides a robust foundation for managing a freelancing platform, supporting scalable user interactions, job processing, and secure contract workflows.

# Chapter 4: Implementation and Testing

## 4.1. Implementation

### 4.1.1. Tools used:

* **Frontend :** Next.js, Styled Component, React
* **Backend :** Express, Prisma, Node.js, PostgreSQL
* **CASE Tools :** VS-Code, GitHub, draw.io, dbdiagram.io, Ms. Word, Ms. PowerPoint

### 4.1.2. Implementation Detail of Module:

The implementation of the proposed system, Linkonomy, was carried out using modern web technologies with a focus on performance, scalability, and user experience. The frontend of the platform was developed using **React** with **TypeScript**, allowing for a component-based structure, strong type safety, and reusable UI elements. This helped in maintaining a modular codebase and improving overall development efficiency.

For the backend, **Node.js** was used to manage server-side operations, including API routing, authentication handling, and communication with the database. The platform utilizes **PostgreSQL** as the database, chosen for its robustness, support for relational data, and ACID compliance — making it suitable for managing user information, job listings, applications, and reviews.

Additionally, a responsive and mobile-friendly design was ensured using CSS libraries and component-based UI logic. The platform allows clients to browse professional profiles, post job requirements, review applications, and hire professionals, while service providers can apply to jobs, manage profiles, and track application statuses.

## 4.2. Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system.

### 4.2.1. Test Case

**Project Name:** Linkonomy

|  |
| --- |
| **Test Case ID:** 1  **Test Title:** Verify login with valid username and password.  **Test Designed Date:** 14 June, 2025 |

|  |
| --- |
| **Pre-conditions:** User has valid username and password.  **Dependencies:** None |

Table : Test Case 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| Navigate to login page, enter valid credentials, and click login | Email: [client@example.com](mailto:client@example.com)  Password: password123 | User should be redirected to their dashboard | User is navigated to the dashboard successfully | Pass |
| Enter invalid credentials and attempt login | Email: [wrong@example.com](mailto:wrong@example.com)  Password: wrong123 | Login should be rejected with an error message | Login rejected with an error message | Pass |

|  |
| --- |
| **Post-condition:**  User is either redirected to dashboard or shown an error, depending on input. |

|  |
| --- |
| **Test Case ID:** 2  **Test Title:**  Post a new job  **Test Designed Date:** 14 June, 2025 |

|  |
| --- |
| **Pre-conditions:** Client must be logged in as client  **Dependencies:** Job form should be functional |

Table : Test Case 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| Navigate to Post Job page, fill required fields, and Post | Title:Web Design  Description: Designer Need with 3 years of experience  Budget: $50  Kwywords: Frontend  Skills Needed: CSS,HTML, React,etc  Duration:2 weeks | Job post should be created successfull | Job post created and listed in dashboard | Pass |

|  |
| --- |
| **Post-condition:**  New job is saved and shown under the client’s posts. |

|  |
| --- |
| **Test Case ID:** 3  **Test Title:**  Apply for a job as a freelancer  **Test Designed Date:** 14 June, 2025 |

|  |
| --- |
| **Pre-conditions:** Professional must be logged in as freelancer and a job must be available  **Dependencies:** Job list should be functional |

Table : Test Case 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| Open job details, click apply, submit proposal | Proposal: “I can do it.”  Duration: 1weeks  Rate: $60 | Application should be submitted successfully | Application is submitted and visible in dashboard | Pass |

|  |
| --- |
| **Post-condition:**  Application is linked to the selected job and visible to the client. |

# 

|  |
| --- |
| **Test Case ID:** 4  **Test Title:**  Hire a professional for a posted job  **Test Designed Date:** 14 June, 2025 |

|  |
| --- |
| **Pre-conditions:** Client must be logged in and have received applications  **Dependencies:** At least one application exists |

Table : Test case 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| Make contract of Proposal | Sign Contract | Contract is visible to both | Professional is successfully hired | Pass |

|  |
| --- |
| **Post-condition:**  The job status is updated and the professional is hired |

# 

|  |
| --- |
| **Test Case ID:** 5  **Test Title:**  Make Payment  **Test Designed Date:** 14 June, 2025 |

|  |
| --- |
| **Pre-conditions:** e-sewa account needed  **Dependencies:** Contract should exist |

Table : Test Case 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected**  **Result** | **Actual**  **Result** | **Status**  **(Pass/Fail)** |
| Freelancer is being paid | Id : 9806800001  Password: Nepal@123 | Payment Successful | Payment Successful | Pass |

|  |
| --- |
| **Post-condition:**  The Payment is successfully completed |

# Chapter 5: Conclusion and Future Recommendation

## 5.1. Lesson Learnt/Outcome

From the development of this project, we gained hands-on experience in planning, designing, and implementing a full-stack web application using modern technologies such as React and TypeScript. We learned how to work with frontend components, manage data using PostgreSQL, and structure a scalable and user-friendly system. We also became familiar with key aspects of project documentation and the design of system diagrams such as use case, ER, activity, and sequence diagrams. This project gave us valuable insight into how service-based platforms operate and how to solve real-world problems through web development.

## 5.2. Conclusion

Linkonomy is a professional service platform developed to bridge the gap between clients and skilled professionals. It is built using React and TypeScript, providing a fast, secure, and responsive user experience. The platform allows users to post jobs, browse professional profiles, apply for services, and manage reviews all within a centralized system. With its focus on usability, trust, and real-time interaction, Linkonomy aims to offer a reliable alternative to traditional service hiring methods. This project also demonstrates how modern frontend frameworks and relational databases can be used to build a feature-rich and intuitive web application. Several technical diagrams, including use case, ER, sequence, and activity diagrams, were incorporated to illustrate the system's structure and behavior.

## 5.3. Future Recommendations

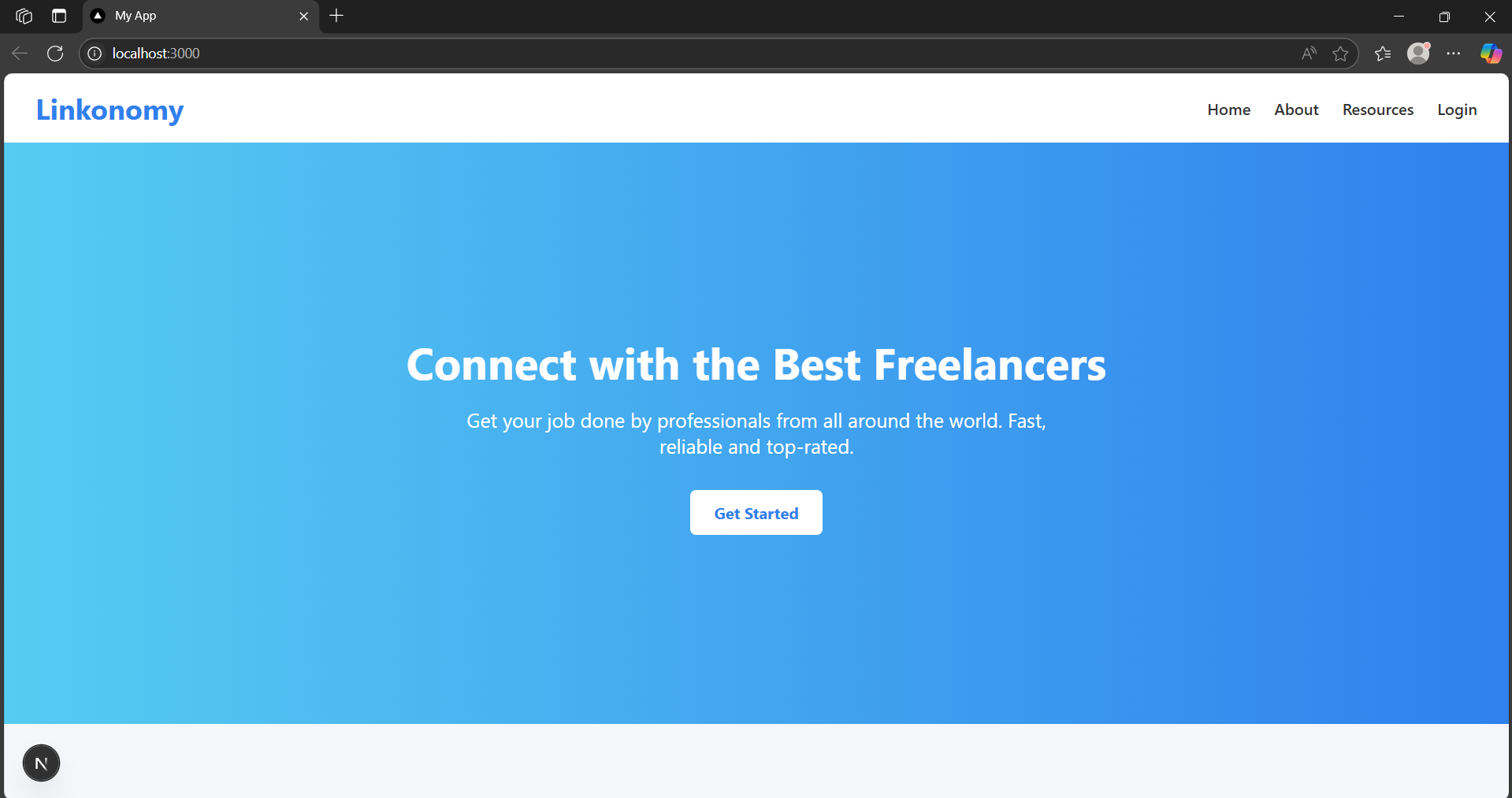
The system can be enhanced and extended in the future with the following improvements:

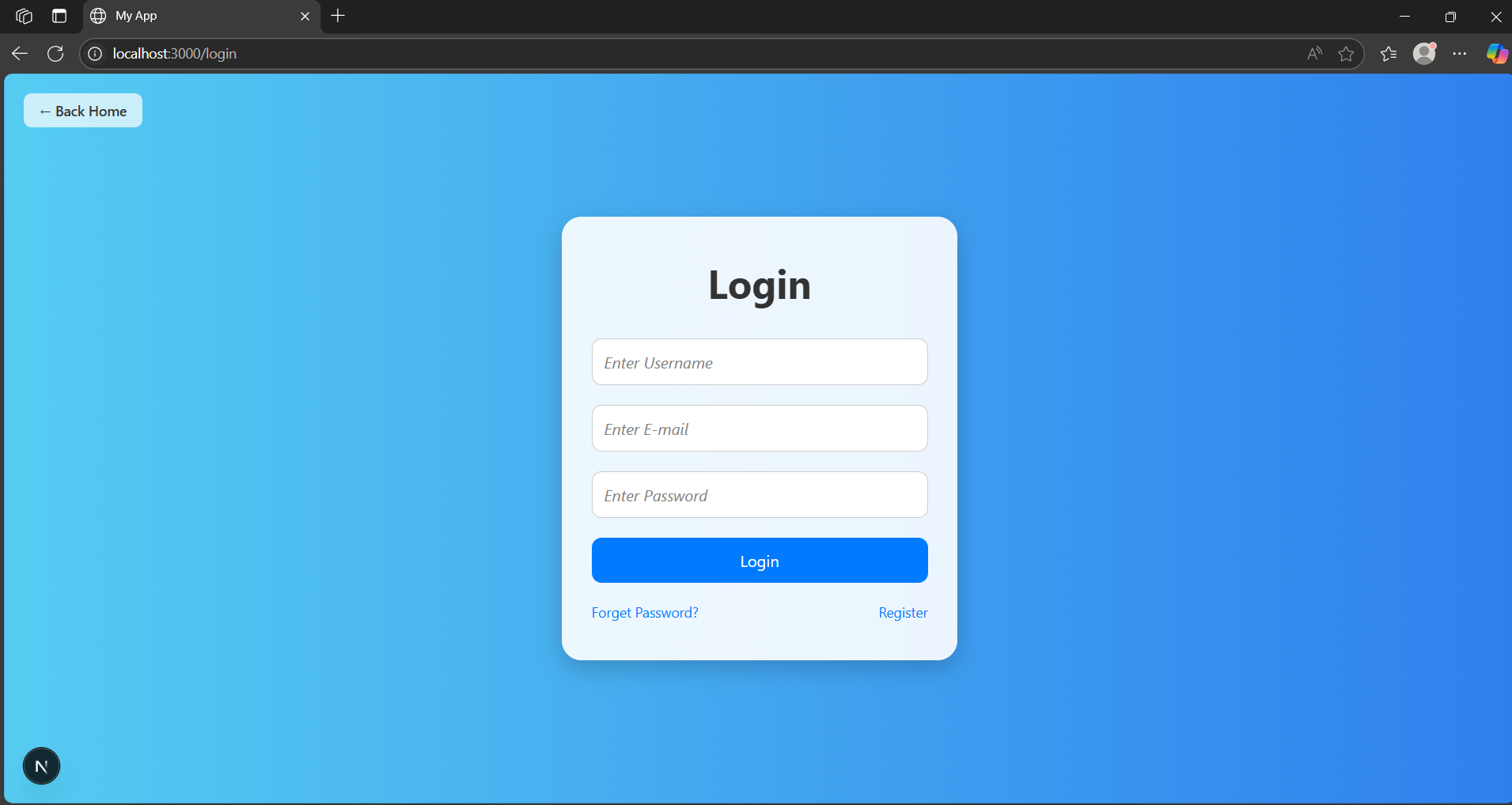
* Improve UI/UX design to make the platform more engaging and accessible.
* Build a dedicated mobile application to reach a wider user base.
* Include advanced filters and AI-based suggestions for better matching.
* Optimize performance and loading speed for users with limited internet access.
* Add a “Saved Professionals” or “Favorite Listings” feature for better user convenience.

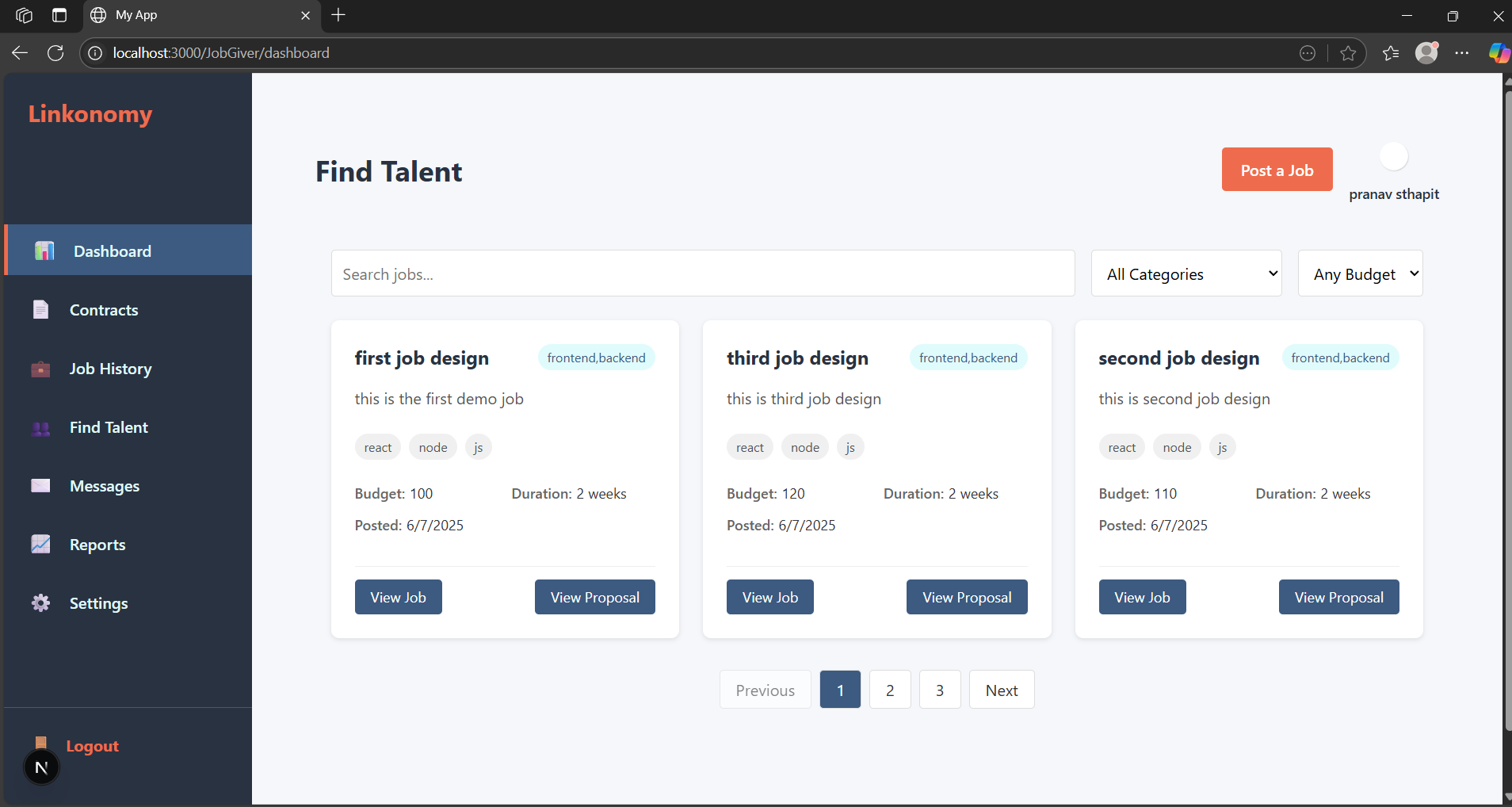
# References

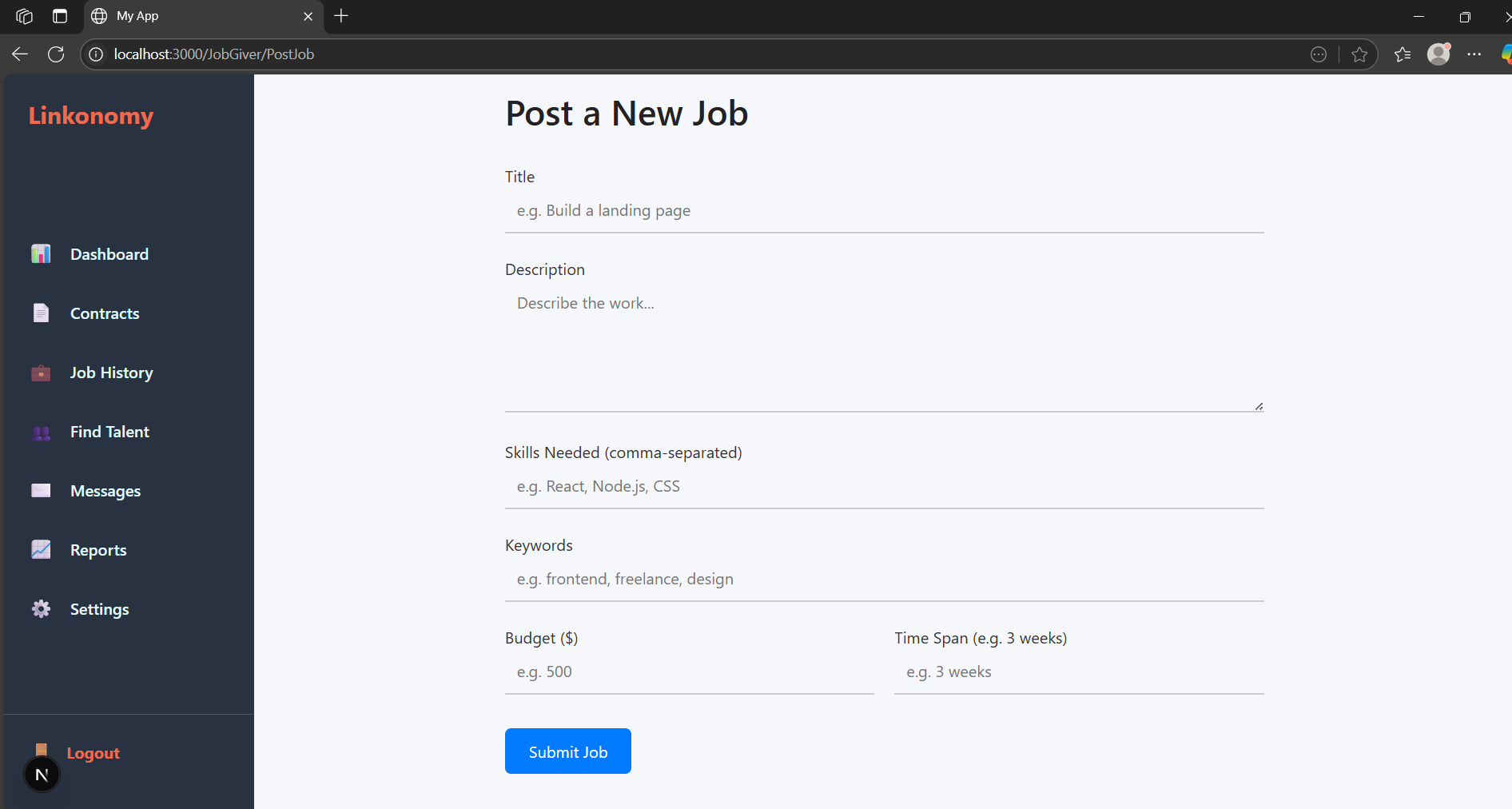
|  |  |
| --- | --- |
| [1] | "Upwork Inc., 'How It Works-Upwork," [Online]. Available: https://www.upwork.com/i/how-it-works/. [Accessed 12 May 2025]. |
| [2] | "Urban Company, "Urban Company - Get Expert Professional Services at Home,"," [Online]. Available: https://www.urbancompany.com/. [Accessed 14 May 2025]. |
| [3] | "React, "React – A JavaScript library for building user interfaces," Meta," [Online]. Available: https://react.dev/. [Accessed 14 May 2025]. |
| [4] | "Microsoft, "TypeScript: JavaScript With Syntax For Types,"," [Online]. Available: https://www.typescriptlang.org/. [Accessed 14 May 2025]. |
| [5] | "Google Firebase, "Firebase Documentation,"," [Online]. Available: https://firebase.google.com/docs. [Accessed 12 May 2025]. |

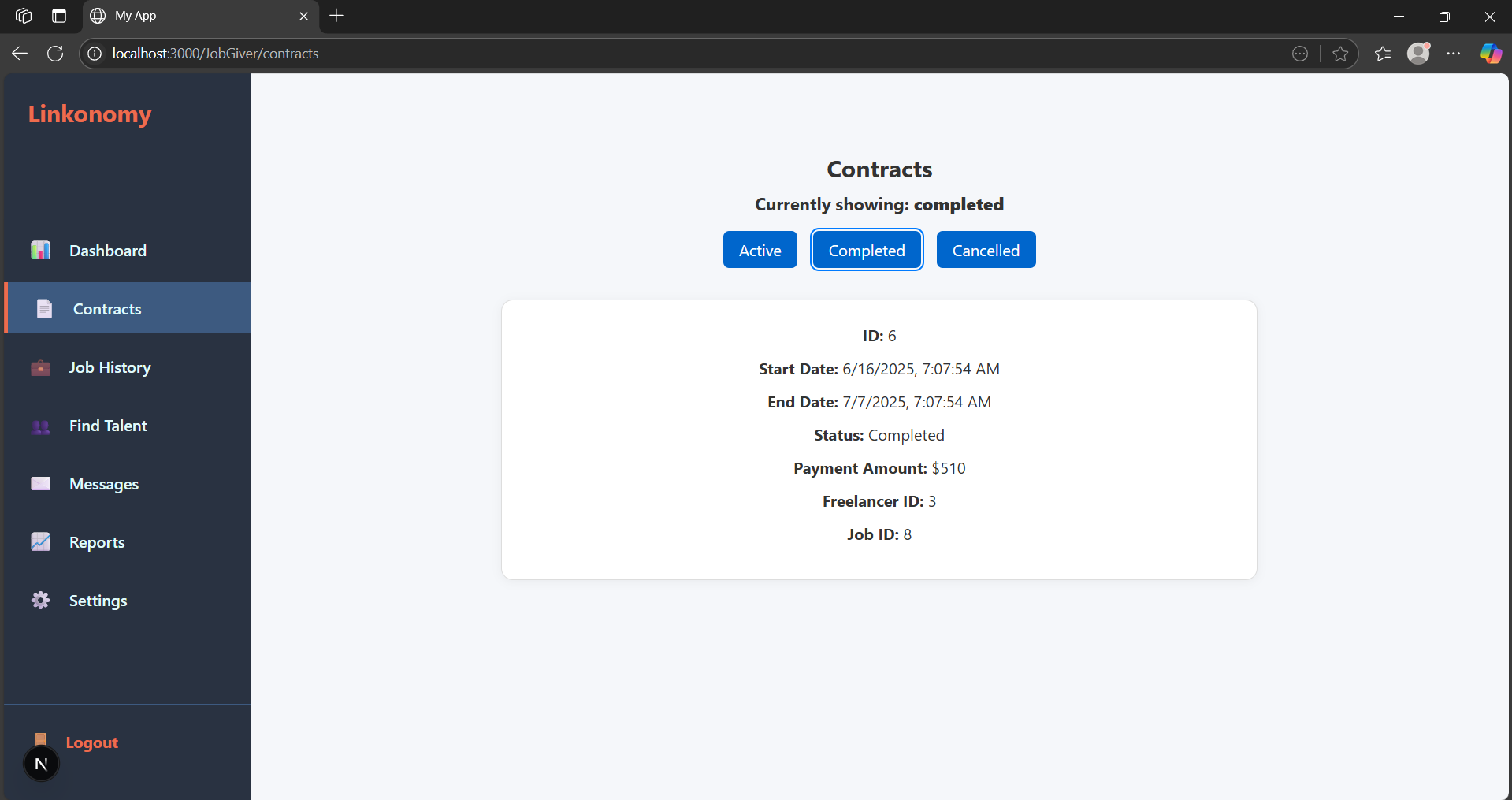
**Appendices**

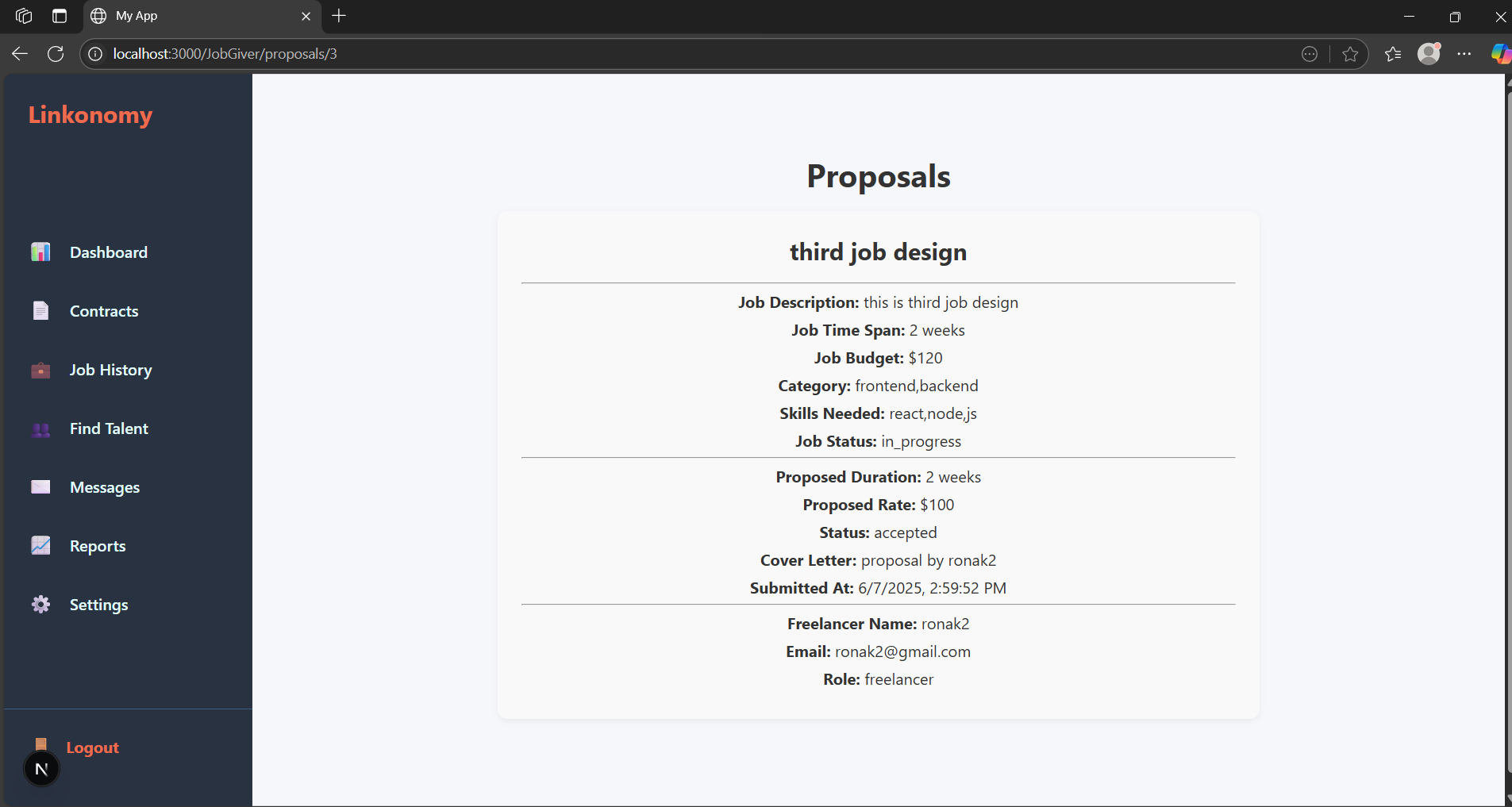
****

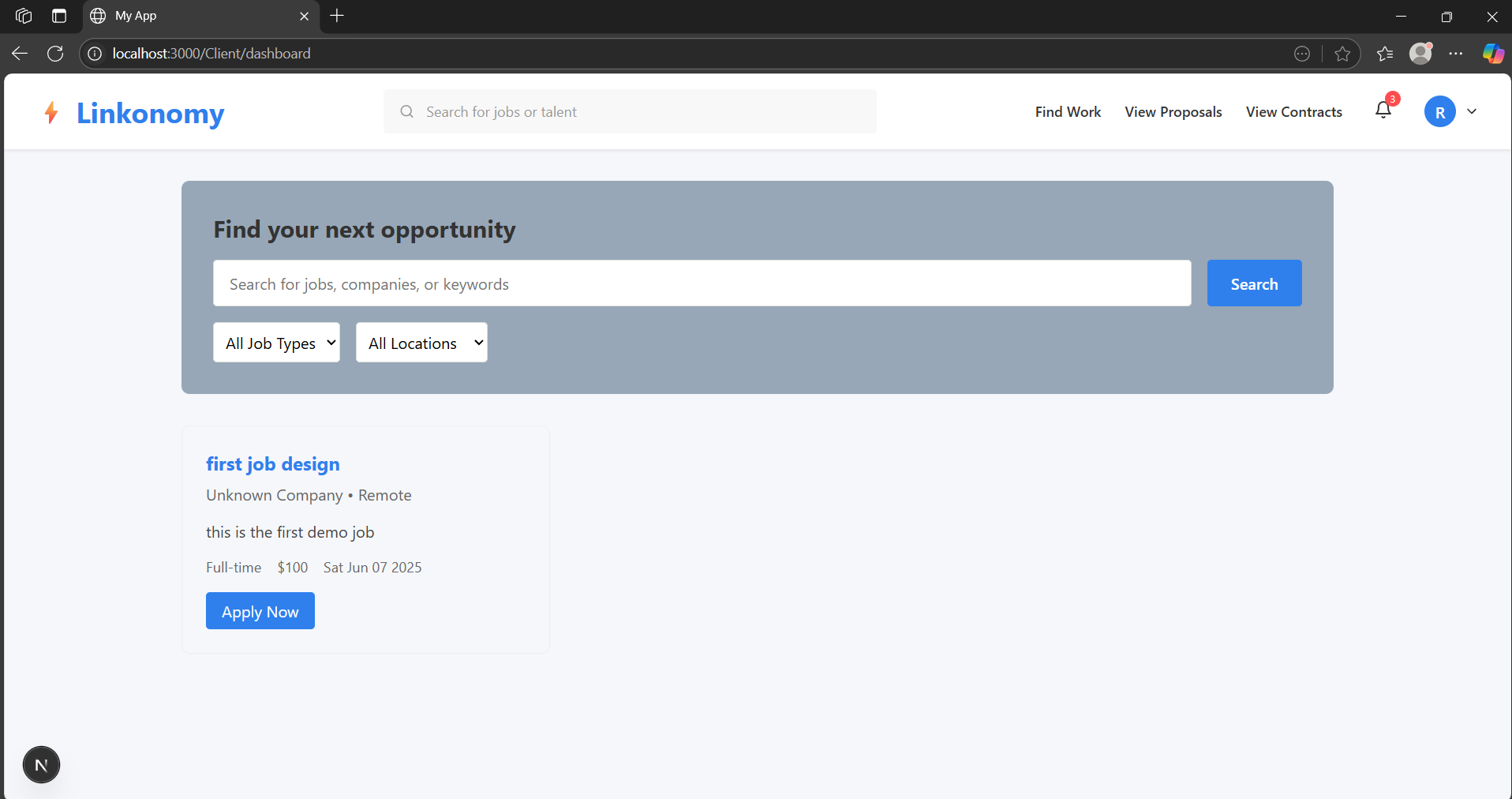
****

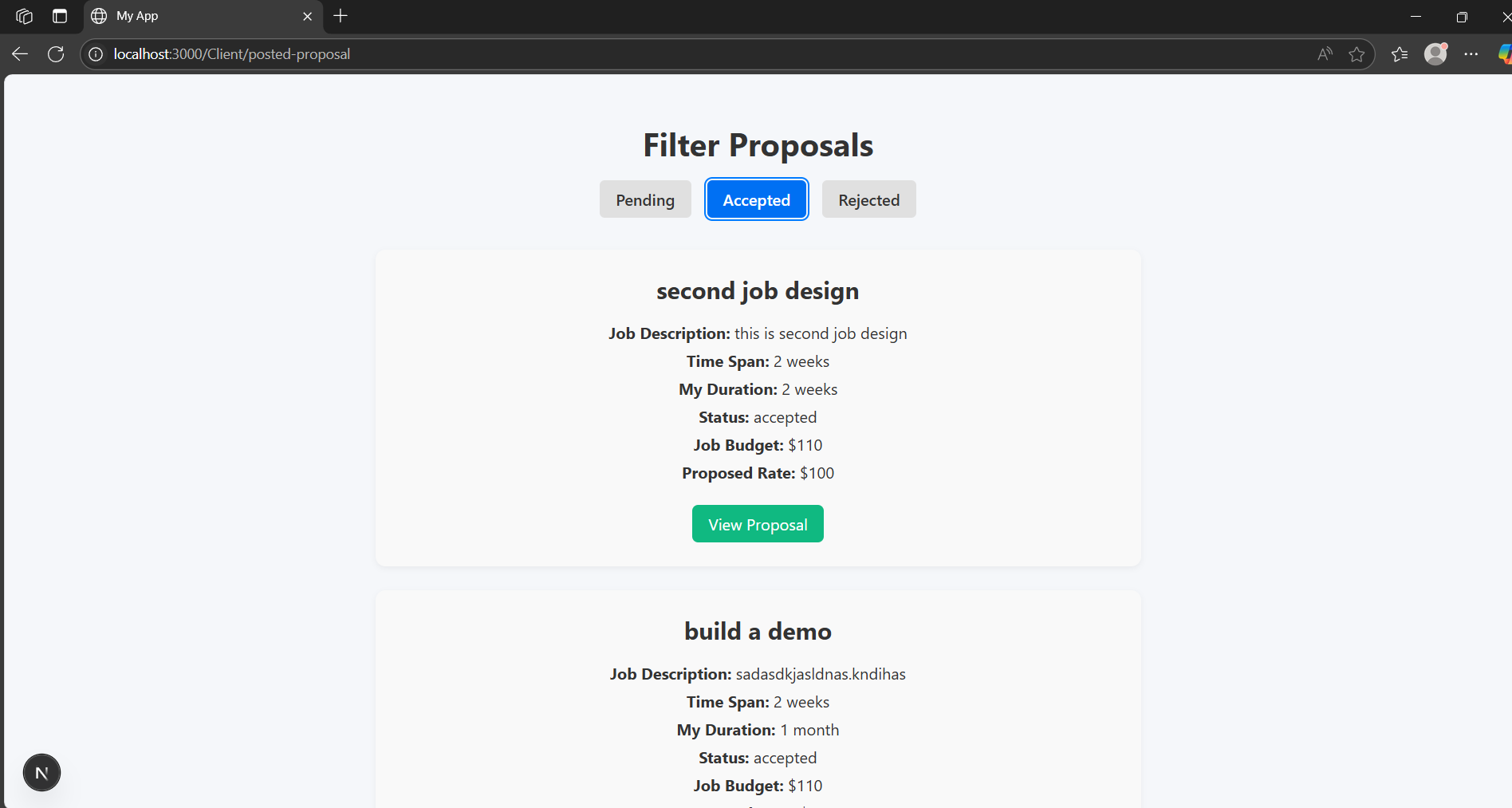
****

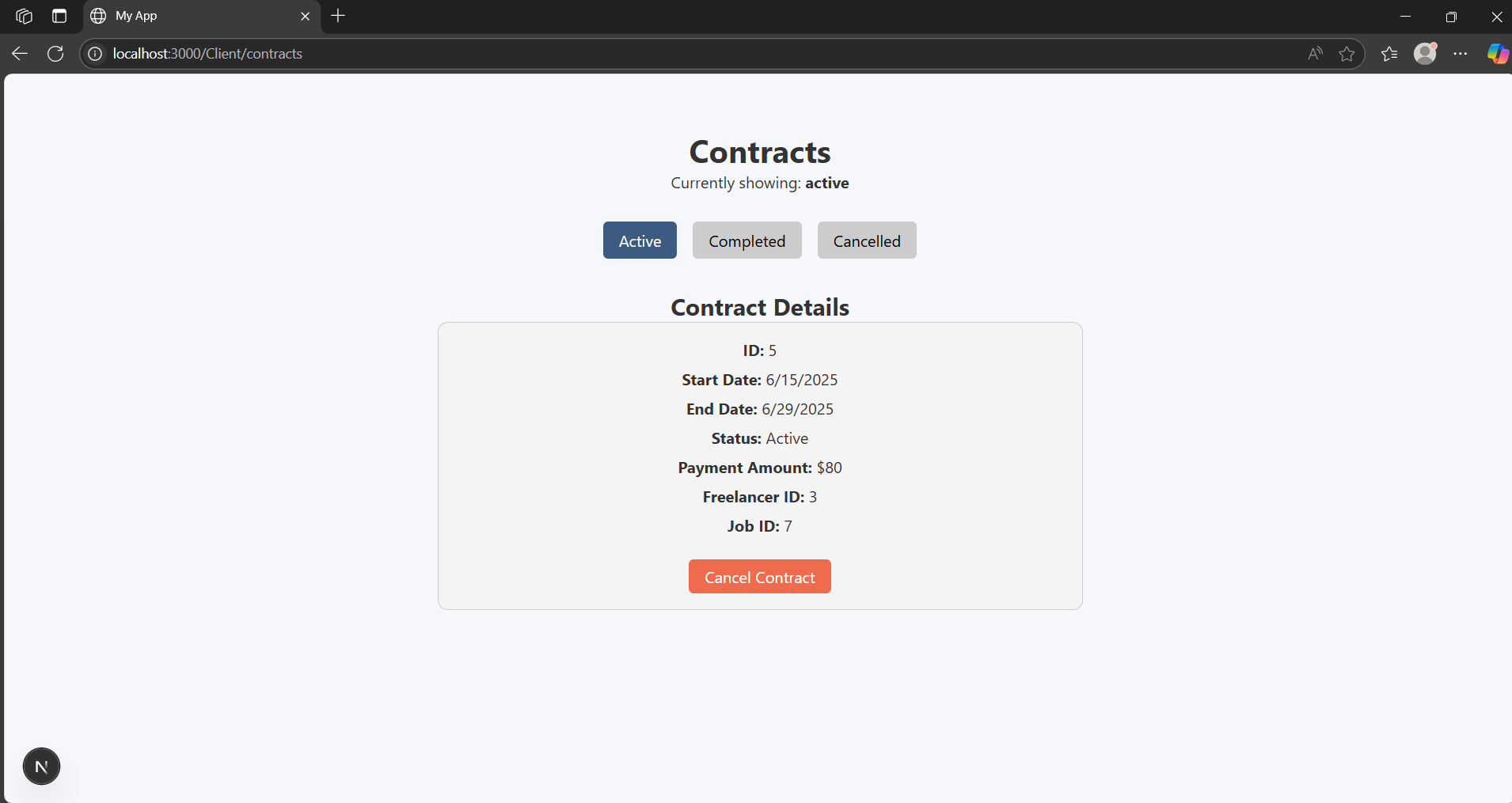
****

****

****

****

****

****