“Freelancing Platform”

A Report submitted under Project-Based Learning

In Partial Fulfillment of the Course Requirements for
“Web Technologies (22IT104001)”

Submitted By

M Varshtih	**22102A040156**
M Nanda kumar	**22102A040157**
K Mohith Reddy	**22102A040134**
K Sathya Narayana	**22102A040128**
G Manidhar	**22102A040082**

Under the Guidance of

Dr Cuddapah Anitha
Associate Professor
Department of CSE

Department of Computer Science and Engineering

School of Computing

MOHAN BABU UNIVERSITY

Sree Sainath Nagar, Tirupati – 517 102

2024-2025

MOHAN BABU UNIVERSITY

Vision

To be a globally respected institution with an innovative and entrepreneurial culture that offers transformative education to advance sustainability and societal good.

Mission

- Develop industry-focused professionals with a global perspective.
- Offer academic programs that provide transformative learning experience founded on the spirit of curiosity, innovation, and integrity.
- Create confluence of research, innovation, and ideation to bring about sustainable and socially relevant enterprises.
- Uphold high standards of professional ethics leading to harmonious relationship with environment and society.

SCHOOL OF COMPUTING

Vision

To lead the advancement of computer science research and education that has real-world impact and to push the frontiers of innovation in the field.

Mission

- Instil within our students fundamental computing knowledge, a broad set of skills, and an inquisitive attitude to create innovative solutions to serve industry and community.
- Provide an experience par excellence with our state-of-the-art research, innovation, and incubation ecosystem to realise our learners' fullest potential.
- Impart continued education and research support to working professionals in the computing domain to enhance their expertise in the cutting-edge technologies.
- Inculcate among the computing engineers of tomorrow with a spirit to solve societal challenges.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision

To become a Centre of Excellence in Computer Science and its emerging areas by imparting high quality education through teaching, training and research.

Mission

- Imparting quality education in Computer Science and Engineering and emerging areas of IT industry by disseminating knowledge through contemporary curriculum, competent faculty and effective teaching-learning methodologies.
- Nurture research, innovation and entrepreneurial skills among faculty and students to contribute to the needs of industry and society.
- Inculcate professional attitude, ethical and social responsibilities for prospective and promising engineering profession.
- Encourage students to engage in life-long learning by creating awareness of the contemporary developments in Computer Science and Engineering and its emerging areas.

B.Tech. Computer Science and Engineering

PROGRAM EDUCATIONAL OBJECTIVES

After few years of graduation, the graduates of B.Tech. CSE will be:

PEO1. Pursuing higher studies in core, specialized or allied areas of Computer Science, or Management.

PEO2. Employed in reputed Computer and I.T organizations or Government to have a globally competent professional career in Computer Science and Engineering domain or be successful Entrepreneurs.

PEO3. Able to demonstrate effective communication, engage in teamwork, exhibit leadership skills and ethical attitude, and achieve professional advancement through continuing education.

PROGRAM OUTCOMES

On successful completion of the Program, the graduates of B.Tech. CSE Program will be able to:

PO1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/Development of Solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct Investigations of Complex Problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern Tool Usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The Engineer and Society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and Sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. **Individual and Team Work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. **Project Management and Finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. **Life-long Learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES

On successful completion of the Program, the graduates of B. Tech. (CSE) program will be able to:

PSO1. Apply knowledge of computer science engineering, Use modern tools, techniques and technologies for efficient design and development of computer-based systems for complex engineering problems.

PSO2. Design and deploy networked systems using standards and principles, evaluate security measures for complex networks, apply procedures and tools to solve networking issues.

PSO3. Develop intelligent systems by applying adaptive algorithms and methodologies for solving problems from inter-disciplinary domains.

PSO4. Apply suitable models, tools and techniques to perform data analytics for effective decision making.

Course Code	Course Title	L	T	P	S	C
22IT104001	**WEB TECHNOLOGIES**	3	-	2	4	5

COURSE OUTCOMES: *After successful completion of this course, the students will be able to:*

CO1. Demonstrate knowledge on web page design elements, dynamic content and database connection.

CO2. Analyze user requirements to develop web applications.

CO3. Design client-server applications using web technologies.

CO4. Demonstrate problem solving skills to develop enterprise web applications.

CO5. Apply HTML, CSS, JavaScript, JQuery, Bootstrap and PHP technologies for device independent web application development.

CO6. Apply web technologies to develop interactive, dynamic and scalable web applications for societal needs.

CO-PO-PSO Mapping Table:

Course Outcomes	Program Outcomes												Program Specific Outcomes			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	2	-	-	-	-	-	-	-	-	-	3	2	3	-
CO2	3	3	3	2	2	-	-	-	-	-	-	-	3	2	3	-
CO3	3	3	3	2	2	-	-	-	-	-	-	-	3	2	3	-
CO4	3	3	3	2	2	-	-	-	-	-	-	-	2	2	3	-
CO5	3	2	2	2	2	3	-	-	-	-	-	-	2	2	3	-
CO6				2					3	3						
Course Correlation Mapping	3	3	3	2	2				3	3			3	2	3	

Correlation Levels: 3: High; 2: Medium; 1: Low

MBU MOHAN BABU UNIVERSITY

MOHAN BABU UNIVERSITY

Sree Sainath Nagar, Tirupati 517 102

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Project Entitled

"Freelancing Platform"

Submitted By

M Varshtih	22102A040156
M Nanda kumar	22102A040157
K Mohith Reddy	22102A040134
K Sathya Narayana	22102A040128
G Manidhar	22102A040082

is the work submitted under Project-Based Learning in Partial Fulfillment of the Course Requirements for "Web Technologies (22IT104001)" during 2024-2025.

Supervisor:

Dr. Cuddapah Anitha
Associate Professor
Department of CSE
School of Computing
Mohan Babu University
Tirupati.

Head:

Dr. G. Sunitha
Professor & Head
Department of CSE
School of Computing
Mohan Babu University
Tirupati.

ACKNOWLEDGEMENTS

First and foremost, I extend my sincere thanks to **Dr. M. Mohan Babu**, Chancellor, for his unwavering support and vision that fosters academic excellence within the institution.

My gratitude also goes to **Mr. Manchu Vishnu, Pro-Chancellor**, for creating an environment that promotes creativity and for his encouragement and commitment to student success.

I am deeply appreciative of **Prof. Nagaraj Ramrao**, Vice Chancellor, whose leadership has created an environment conducive to learning and innovation.

I would like to thank **Dr. K. Saradhi**, Registrar, for his support in creating an environment conducive to academic success.

My sincere thanks to **Dr. B.M. Satish**, Dean of the School of Engineering and Computing, for his valuable support and guidance in all academic matters.

I am also grateful to **Dr. G. Sunitha**, Head of the Department of Computer Science and Engineering, for her valuable insights and support.

Finally, I would like to express my deepest appreciation to my project supervisor, **Dr. Cuddpah Anitha**, Associate Professor, Department of Computer Science and Engineering for continuous guidance, encouragement, and expertise throughout this project.

Thank you all for your support and encouragement.

Table of Contents

ABSTRACT

In recent years, the demand for freelance work has surged, driven by the need for flexible job opportunities and the desire for diverse project engagements. Despite this growth, there remains a significant gap in effectively connecting freelancers and gig workers with short-term and project-based job opportunities in India. This project aims to develop a comprehensive freelancing platform akin to established marketplaces like Upwork, tailored specifically for the Indian market.

The proposed platform will serve as a dynamic freelance job marketplace where freelancers can discover and apply for diverse job opportunities while allowing employers to post projects, define requirements, and invite suitable freelancers to participate. To enhance user engagement and credibility, freelancers will be empowered to create detailed profiles showcasing their skills, experiences, and portfolios, supplemented by a robust rating and review system for feedback on completed projects.

Furthermore, the platform will leverage advanced search and analytics capabilities, enabling both freelancers and employers to conduct extensive searches of available opportunities and generated data. Utilizing AI-driven insights, the platform will provide innovative ways for job seekers and employers to interact, with personalized recommendations tailored to individual preferences and skills.

A crucial feature of this platform will be the implementation of an escrow account system, ensuring secure financial transactions between employers and freelancers. Funds will be held in escrow until project completion, thus fostering trust and accountability. Additionally, integrating secure payment gateways will facilitate seamless financial transactions.

Ultimately, this project aspires to connect freelancers with a wide array of job opportunities while equipping them with essential tools for project management. By addressing the challenges faced in the freelancing sector, this platform aims to enhance job satisfaction and increase income opportunities for freelancers, contributing positively to the evolving gig economy in India.

1. INTRODUCTION

1.1 Problem Statement

Freelancing in India has surged, providing flexible work opportunities and access to a skilled workforce for businesses. However, the existing platforms fail to meet the specific needs of Indian freelancers and employers, creating a gap that hampers effective job matching and project management.

One major issue is **limited accessibility to relevant opportunities**. Freelancers often struggle to find localized, short-term gigs and project-based roles tailored to their skills, as most platforms cater to a global audience, leaving them underutilized in the Indian market. On the other side, employers face **inefficiencies in matching with suitable freelancers**, as finding individuals with the right expertise and availability remains challenging without a streamlined application process.

Profile management and credibility-building tools are also insufficient. Freelancers need robust profiles showcasing their skills and experience, along with rating and review systems to build trust with potential clients. Yet, most platforms lack these features, impacting freelancers' visibility and credibility.

Search and analytical tools are another limitation. Effective search capabilities and AI-driven recommendations are essential to help freelancers discover relevant projects and employers find ideal candidates. However, the absence of intelligent insights often leads to missed opportunities.

Lastly, **secure transaction processes** are lacking. Freelancers and employers frequently face payment issues due to the absence of escrow systems, leading to mistrust and disputes.

Addressing these issues, this project envisions a freelancing platform specifically for India, featuring tools for improved job access, efficient matching, comprehensive profile management, AI-enhanced search, and secure financial transactions. Such a platform would not only empower freelancers but also streamline hiring for employers, fostering a professional and trustworthy freelancing ecosystem in India.

1.2 Importance of the Problem

Addressing the challenges facing Indian freelancers and employers in the gig economy is essential for numerous reasons. **Economic growth and workforce evolution** are closely linked to freelancing, as this sector is rapidly expanding and contributing significantly to GDP and employment. A dedicated platform can further enhance economic impact by supporting this evolving workforce.

Empowering freelancers is also crucial. Many skilled professionals and creatives seek freelancing for flexibility and autonomy. By offering a marketplace where they can showcase skills, connect with reputable employers, and access diverse job opportunities, the platform can enhance their professional experience and income potential.

For **employers**, a secure and reliable system increases confidence in engaging freelance talent. Features like a **rating and review system** and **an integrated escrow** for secure payments can strengthen trust, ensuring that freelancers deliver quality work and employers receive value, fostering a healthier work environment.

By **reducing barriers to entry**, the platform can attract talented individuals who might hesitate to freelance due to a lack of exposure or resources. Providing essential tools and support can make freelancing more accessible, broadening the talent pool. Additionally, **AI-driven insights** offer freelancers and employers personalized recommendations, making job or talent matching more efficient.

Supporting this platform aligns with national priorities for **workforce development and self-reliance**, as it offers opportunities that resonate with the government's emphasis on skill-building and entrepreneurship.

In summary, addressing these challenges is vital for empowering freelancers, strengthening employer confidence, and boosting the gig economy in India. This platform can transform freelancing into a more secure, efficient, and impactful opportunity for all stakeholders.

1.3 Objectives

Here are the objectives of the freelancing platform outlined in seven points:

1. **Enhance Job Accessibility**: Provide freelancers with a wide range of relevant and localized short-term job opportunities.

2. **Streamline Matching Processes**: Facilitate efficient connections between freelancers and employers through advanced search and AI-driven recommendations.

3. **Improve Profile Management**: Enable freelancers to create comprehensive profiles and portfolios, incorporating a rating and review system to build credibility.

4. **Ensure Secure Transactions**: Implement an escrow system to guarantee safe and transparent payments, fostering trust between freelancers and employers.

5. **Provide Data-Driven Insights**: Leverage AI analytics for personalized recommendations, helping users make informed decisions about job opportunities and talent acquisition.

6. **Support Professional Development**: Offer resources and tools for freelancers to enhance their skills and visibility in the marketplace.

7. **Foster Economic Growth**: Contribute to the growth of the gig economy in India by empowering freelancers and providing employers with access to skilled talent.

1.4 Scope of the Project

The project aims to create a freelancing platform specifically designed for the Indian market, connecting freelancers with clients seeking short-term, gig-based, and project-based work opportunities. This platform will function as a dynamic marketplace, enabling freelancers to access a wide range of job postings while providing clients with tools to specify project requirements, review applicants, and select freelancers with the right expertise.

The platform will include essential features such as comprehensive freelancer profiles and portfolio management tools. Freelancers can build profiles showcasing their skills, experiences, and past projects, which clients can review when selecting candidates. Clients, on the other hand, will have dedicated project-posting pages to define requirements, timelines, and budgets. A built-in rating and review system will allow both parties to provide feedback, enhancing trust and credibility.

An extensive search and filtering system will help users navigate the marketplace efficiently, allowing freelancers and clients to filter job postings and profiles based on specific criteria such as skillsets, budget, and project type. Secure payment processing, integrated with an escrow system, will provide financial security by holding funds until job milestones are achieved.

The platform’s front end will be developed using HTML, CSS, JavaScript, and Bootstrap to ensure a responsive, mobile-friendly, and user-friendly interface. With a focus on intuitive design and secure functionality, this platform will address the unique challenges of India’s gig economy, facilitating seamless connections and trustworthy transactions between freelancers and clients. Ultimately, it will empower both parties with the tools needed for efficient project collaboration and professional growth.

2. System Design

2.1. Architecture Diagram

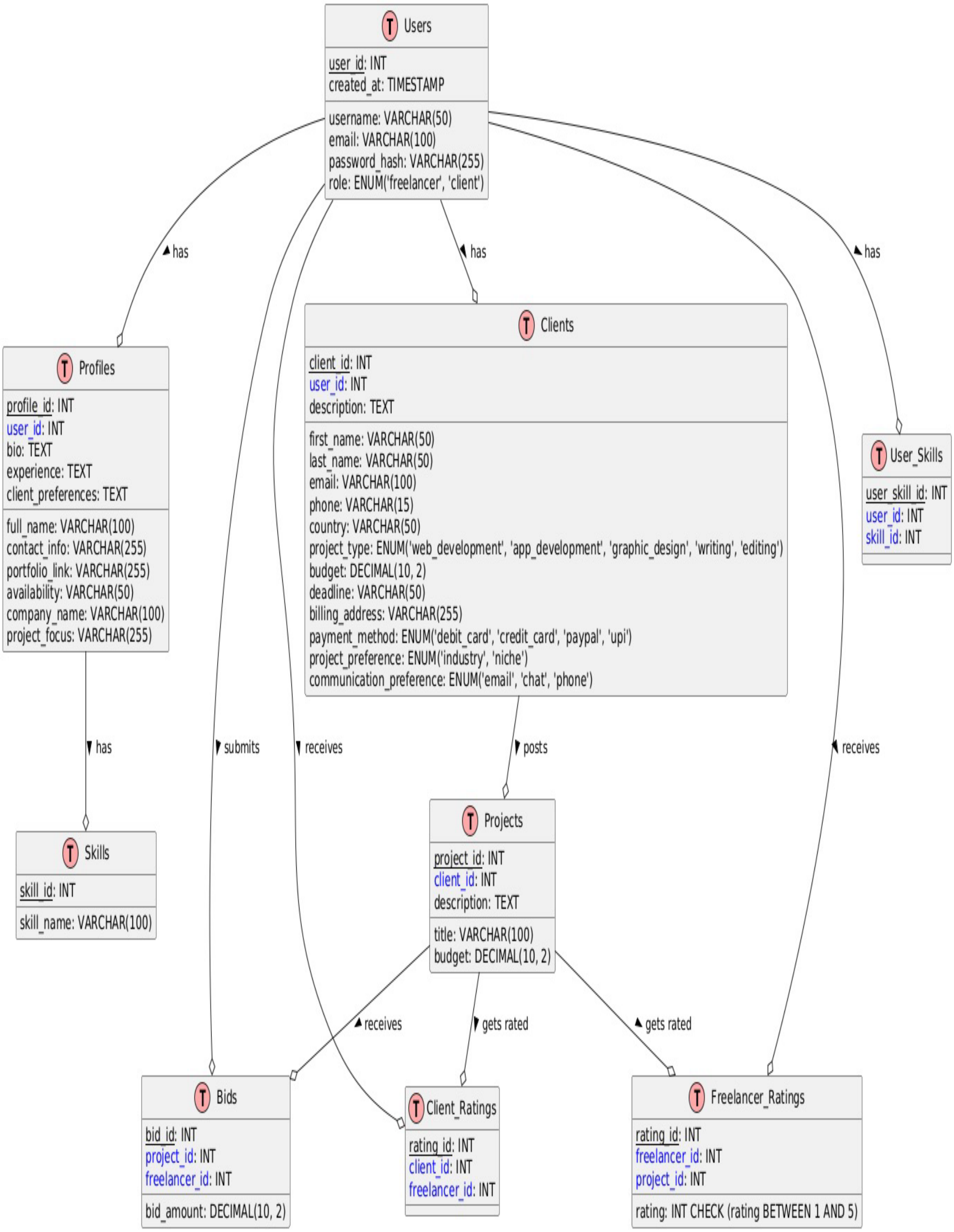


Fig 2.1.1 Architecture Diagram of Freelancing Platform

2.2. Module Description

1. Users Module

- **Purpose**: Manages user accounts for both freelancers and clients.
- **Key Features**:
  - **User Registration**: Handles new user registrations with role selection (freelancer or client).
  - **Login**: Secure login system for existing users.
  - **Role Management**: Differentiates user types (freelancers and clients) for role-specific actions.
  - **Password Hashing**: Encrypts passwords for enhanced security.

2. Profiles Module

- **Purpose**: Stores and manages additional profile information for users.
- **Key Features**:
  - **Profile Creation & Updates**: Allows users to create and edit profiles.
  - **Bios**: Users can share information about themselves.
  - **Portfolios**: Freelancers can display work samples or portfolios.
  - **Availability**: Users can update availability status for potential projects.

3. Clients Module

- **Purpose**: Stores detailed, client-specific information.
- **Key Features**:
  - **Company Information**: Captures client business details, including company name and type.
  - **Project Preferences**: Saves client preferences, such as project types and specific requirements.

4. Skills Module

- **Purpose**: Maintains a centralized list of available skills on the platform.

- **Key Features**:
  - **Skill Selection**: Freelancers can choose relevant skills to display.
  - **Showcase Skills**: Allows freelancers to showcase their skills, helping clients find the right talent.

5. Projects Module

- **Purpose**: Manages projects that clients post to the platform.
- **Key Features**:
  - **Project Creation**: Clients can post projects with detailed descriptions.
  - **Budget Setting**: Clients can set budgets for each project.
  - **Deadline Management**: Specifies and manages project timelines and deadlines.

6. Bids Module

- **Purpose**: Facilitates project bidding for freelancers.
- **Key Features**:
  - **Bid Submission**: Allows freelancers to submit bids with a proposed amount and brief description.
  - **Competitive Bidding**: Encourages a competitive environment for project acquisition.

7. Ratings Module

- **Purpose**: Collects and manages feedback for freelancers and clients upon project completion.
- **Key Features**:
  - **User Ratings**: Enables both freelancers and clients to rate each other.
  - **Review System**: Allows users to leave detailed reviews, fostering transparency and trust on the platform.

Each of these modules is essential to providing a comprehensive and efficient freelancing platform, enhancing user experience for both freelancers and clients.

2.3 Database Design

1. users

- **Purpose**: Stores basic user information, such as usernames, passwords, and roles.
- **Relationships**:
  - **One-to-Many with profiles**: Each user can have one profile.
  - **One-to-Many with clients**: Each user can be associated with client-specific data.
  - **One-to-Many with user_skills**: Users (freelancers) can have multiple skills.
  - **One-to-Many with bids**: Freelancers can place multiple bids on projects.
  - **One-to-Many with freelancer_ratings**: Freelancers receive multiple ratings from clients.
  - **One-to-Many with client_ratings**: Clients can receive multiple ratings from freelancers.

2. profiles

- **Purpose**: Contains additional information about users (bios, portfolios, availability, etc.).
- **Foreign Key**:
  - user_id references users(user_id) to link each profile to a specific user.

3. clients

- **Purpose**: Stores detailed information specific to clients (e.g., company name, project type).
- **Foreign Key**:
  - user_id references users(user_id) to associate client details with the relevant user.

4. skills

- **Purpose**: Holds a master list of available skills that freelancers can possess.
- **No Foreign Keys**: Serves as a reference table for skill types.

5. user_skills

- **Purpose**: Links freelancers with their respective skills, allowing each user to showcase multiple skills.
- **Foreign Keys**:
  - user_id references users(user_id) to link to the freelancer.
  - skill_id references skills(skill_id) to specify the skill.

6. projects

- **Purpose**: Stores information about projects posted by clients, including budgets and deadlines.
- **Foreign Key**:
  - client_id references users(user_id) to associate projects with specific clients.

7. bids

- **Purpose**: Records bids submitted by freelancers on various projects, including bid amounts and descriptions.
- **Foreign Keys**:
  - project_id references projects(project_id) to link each bid to a project.
  - freelancer_id references users(user_id) to identify the freelancer placing the bid.

8. freelancer_ratings

- **Purpose**: Stores ratings and reviews given to freelancers by clients after project completion.
- **Foreign Keys**:
  - freelancer_id references users(user_id) to identify the freelancer being rated.
  - project_id references projects(project_id) to specify the project associated with the rating.

9. client_ratings

- **Purpose**: Contains ratings and reviews given to clients by freelancers post-project.
- **Foreign Keys**:

- client_id references users(user_id) to identify the client being rated.

- freelancer_id references users(user_id) to identify the freelancer giving the rating.

- project_id references projects(project_id) to specify the project associated with the rating.

3. Implementation

3.1. Tools used in Implementation

This project utilizes a range of tools and technologies to build a user-friendly, responsive, and functional platform. Below are the key technologies used:

1. **HTML (Hypertext Markup Language)**
   - HTML is the backbone of the web application's structure, providing a semantic foundation for content organization. By defining essential components like forms, buttons, navigation, and sectioning elements, HTML ensures that the content is accessible, easy to navigate, and consistent across various devices and browsers.
2. **CSS (Cascading Style Sheets)**
   - CSS is used to style HTML elements, bringing visual appeal and brand alignment to the platform. It includes layout management, colors, fonts, and custom designs, enhancing readability and user engagement. CSS is key for managing responsive behavior, ensuring an adaptable interface that looks polished on both desktop and mobile devices.
3. **Bootstrap**
   - Bootstrap is a front-end framework that simplifies the creation of responsive, mobile-first designs. Its pre-built components—like modals, carousels, buttons, and form elements—save development time and help maintain a cohesive visual style. The Bootstrap grid system ensures consistent layouts that adapt to screen sizes, while its responsive utilities enable fine-tuning for specific screen resolutions, creating a smooth, cross-device experience.
4. **JavaScript**
   - JavaScript adds interactive, dynamic functionality to the application. By handling client-side logic, JavaScript improves user interactions with real-time updates, form validation, and animations. JavaScript frameworks or libraries, like jQuery, may be used to simplify complex tasks, enhancing the platform’s functionality, performance, and user experience.
5. **MySQL**
   - MySQL serves as the database, storing structured data including user accounts, profiles, projects, skills, bids, and ratings in a relational format. This relational database management system ensures data integrity and efficient querying, making it easy to retrieve and manage user information, project details, and interactions within the platform. MySQL’s scalability and compatibility with various back-end technologies support reliable and secure data management as the platform grows.

3.2. Front-End Development

Describe the purpose of front-end development in the context of the freelancing platform. Emphasize that the front-end is responsible for enabling user interaction, providing an accessible interface, and supporting the functionality necessary for users, such as freelancers and clients, to connect, communicate, and manage projects.

1. Technologies and Frameworks Used

List and briefly describe the main technologies and frameworks used for the front-end:

- **HTML5**: Used for structuring the web pages, providing semantic elements, and improving accessibility.
- **CSS3**: Utilized for styling and layout, including the use of responsive design to ensure the platform adapts well to different devices (desktop, tablet, mobile).
- **JavaScript**: Powers interactive elements, form validations, and dynamic content updates without reloading pages.
- **Bootstrap**: For a responsive, grid-based layout and pre-built components (e.g., buttons, forms, modals) to expedite development.
- **AJAX**: Enables asynchronous data exchange with the server, allowing parts of the web page to update dynamically without a full page refresh.

2. User Interface Design

- **Design Principles**: Describe the design principles used, such as consistency, minimalism, and ease of navigation.
- **Color Scheme**: Explain the colors chosen to reflect the platform's branding (e.g., professional, trustworthy colors like blue, grey, and white).
- **Typography**: Mention the fonts used and their purpose (e.g., readability, professionalism).
- **Iconography**: Discuss the role of icons (e.g., dashboard icons, profile icons) in enhancing the UX by providing visual cues for different sections.

3. Responsive Design

Explain how the platform is designed to be fully responsive, making it accessible on devices of varying screen sizes. Mention techniques used, such as:

- **Media Queries**: To adjust layouts based on screen size.
- **Flexible Grid Layouts**: Using frameworks like Bootstrap for easy, responsive grids.
- **Responsive Images**: Scaling images to fit different devices without compromising quality.

4. Key Front-End Features

Provide an overview of essential front-end features developed for the platform, such as:

- **User Dashboard**: Designed for both freelancers and clients to access essential tools and project overviews quickly.
- **Job Posting and Browsing**: Enables clients to post jobs and freelancers to search for projects, with interactive filters and search capabilities.
- **Profile Management**: Allows freelancers to showcase their portfolios, skills, and experience, while clients can manage their project requirements and history.
- **Messaging System**: Facilitates secure, real-time communication between freelancers and clients within the platform.
- **Real-Time Notifications**: Notifies users of important updates (e.g., new messages, job offers, project updates) using AJAX or WebSockets.
- **Escrow Integration**: Provides a secure, front-end interface for managing payments, using intuitive input fields and visual confirmations.

3.3. Back-End Development

Begin by describing the purpose of back-end development in the context of the freelancing platform. Emphasize how the back end is responsible for data management, business logic, and secure transactions to support user interactions, such as job postings, profile management, and messaging.

1. Technologies and Frameworks Used

List and briefly explain the primary back-end technologies used:

- **PHP**: Chosen as the server-side scripting language for handling requests, processing data, and managing interactions with the database.
- **MySQL**: Used as the relational database management system to store and retrieve user information, job postings, messages, and payment records.
- **AJAX**: Facilitates asynchronous data exchange between the client and server, enhancing user experience without full-page reloads.
- **Session Management**: PHP sessions are utilized to maintain user states (like login sessions) and provide secure, individualized access.

2. Database Design and Structure

Describe the design and architecture of the database:

- **Entity-Relationship Model**: Present the key tables and their relationships, including tables for users, jobs, proposals, messages, transactions, etc.
- **Database Normalization**: Explain how normalization techniques were applied to minimize data redundancy and ensure efficient data retrieval.

- **Data Security Measures**: Discuss how sensitive information, such as passwords and payment details, is protected (e.g., password hashing, secure payment gateway integration).

3. Core Back-End Features

Outline the core features developed on the back end, explaining their purpose and functionality:

- **User Authentication**: Implements secure registration and login processes with hashed passwords and session management for both freelancers and clients.
- **Job Posting and Bidding System**: Allows clients to post jobs and freelancers to submit proposals. Describe how back-end logic validates and manages these actions.
- **Profile Management**: Enables users to create, update, and showcase their profiles. Describe how user information is stored, retrieved, and modified.
- **Escrow System**: Supports secure transactions by holding payments in escrow until project completion, leveraging third-party payment gateway integration.
- **Messaging System**: Implements real-time or near-real-time messaging between users, ensuring privacy and secure communication channels.

3.4. Integration

Begin by explaining the purpose of integration in the project. Describe how integration ensures that the front-end, back-end, and database layers function together to provide features such as user management, job postings, payments, and real-time messaging. Emphasize that integration enables seamless interactions between freelancers and clients, leading to efficient project management.

1. Integration of Front-End and Back-End

Describe how the front-end (HTML, CSS, JavaScript) and back-end (PHP) communicate:

- **RESTful API Calls**: Explain how the front end interacts with the back end through API calls for functions such as user registration, login, job postings, and proposal submissions.
- **AJAX for Asynchronous Requests**: Illustrate the use of AJAX to enable asynchronous data exchange, allowing users to interact with the platform without needing to refresh the page.
- **JSON Data Format**: Describe the use of JSON as the standard format for data exchange between the front-end and back-end, facilitating smooth data handling on both sides.

2. Database Integration

Detail how the back-end interacts with the MySQL database to store, retrieve, and update data:

- **Database Queries**: Explain how the back end uses SQL queries to perform operations like retrieving job listings, saving user profiles, and updating job statuses.

- **Prepared Statements**: Emphasize the importance of using prepared statements for secure, efficient data queries, preventing SQL injection.
- **Data Flow Examples**: Provide examples of data flows, such as how a user registration request from the front-end is validated by the back-end, saved in the database, and confirmed back to the user.

3. User Authentication and Session Management

Describe the integration of user authentication, roles, and session management:

- **Login and Session Tracking**: Show how the login process authenticates users and initiates a session. Explain how user roles (freelancer or client) are stored in the session for role-based access control.
- **Session Security**: Mention any session management practices to secure user sessions, like setting secure cookies and implementing session timeouts to protect user accounts.

4. Payment Gateway Integration

Explain how third-party payment gateways (e.g., PayPal, Stripe) are integrated to manage financial transactions:

- **API Integration**: Describe how the back-end securely communicates with payment gateway APIs for processing deposits, withdrawals, and escrow payments.
- **Transaction Management**: Explain how transaction statuses are saved in the database, allowing users to track payments and manage refunds.
- **Escrow Workflow**: Outline the escrow functionality that holds funds until project completion, then releases them once all criteria are met.

5. Messaging and Notifications

Detail the integration of real-time messaging and notification systems to support user interaction:

- **WebSocket or AJAX for Messaging**: Describe the use of WebSocket technology or AJAX polling to enable real-time messaging between freelancers and clients.
- **Database-Driven Notifications**: Explain how notifications are generated in response to events (e.g., new messages, job postings) and stored in the database for retrieval on the dashboard.

6. Data Synchronization and Consistency

Address the importance of maintaining data consistency across modules and managing concurrent actions:

- **Transaction Handling**: Describe how database transactions ensure data integrity, such as ensuring that a job posting and corresponding client balance update are atomic (completed together).

4. Testing, Results and Discussion

4.1. Test Cases

1.Functional Test Cases

These test cases verify that each function of the application works according to the requirements.

Test Case ID	Description	Precondition	Test Steps	Expected Result
TC_F_01	User registration	User is on the registration page	Enter valid username, email, password, and select user type, then submit	User account is created successfully
TC_F_02	User login	User is on the login page	Enter correct username and password, then submit	User is redirected to the dashboard
TC_F_03	Job posting by client	User is logged in as client	Navigate to job posting page, fill job details, and submit	Job is posted and visible on job board
TC_F_04	Proposal submission by freelancer	User is logged in as freelancer	Find a job, click "Submit Proposal," enter proposal details, and submit	Proposal is submitted successfully
TC_F_05	Password reset	User is on login page	Click "Forgot Password," enter email, and check for reset email	Password reset email is sent
TC_F_06	Payment processing through escrow	Client and freelancer on job	Client initiates payment, funds held in escrow until job is completed	Funds are securely held in escrow
TC_F_07	Profile update	User is logged in	Navigate to profile, make changes, and save	Profile is updated with new information

2. Integration Test Cases

These cases ensure that the different components of the platform (front-end, back-end, and database) work together as expected.

Test Case ID	Description	Precondition	Test Steps	Expected Result
TC_I_01	Front-end and back-end login integration	User on login page	Enter valid credentials, submit, and verify response	User is logged in and session is created
TC_I_02	Job posting sync between client and job board	Client posts job	Client posts a job, check if job appears in job board listing	Job is visible to all freelancers
TC_I_03	Proposal submission integration	Freelancer submits proposal	Freelancer submits proposal, check if it is stored in the database	Proposal appears in client dashboard

TC_I_04	Messaging functionality integration	Client and freelancer on platform	Client sends a message to freelancer; freelancer receives notification	Message delivered and displayed in chat
TC_I_05	Payment gateway integration with escrow	Client initiates payment	Process payment through gateway and check if escrow holds amount	Payment is held securely in escrow

3. User Interface (UI) Test Cases

These cases verify the user interface elements to ensure they meet usability and design requirements.

Test Case ID	Description	Precondition	Test Steps	Expected Result
TC_UI_01	Responsive design across devices	User opens site on different devices	Open platform on mobile, tablet, and desktop	UI elements adjust to each screen size
TC_UI_02	Navigation between pages	User is logged in	Click on various menu options and observe	User is navigated to correct page
TC_UI_03	Form validation on registration	User is on registration page	Leave fields empty or enter invalid data, then submit	Errors are displayed for invalid input
TC_UI_04	Notification display and clearing	User receives notifications	Check notifications section; clear notifications and refresh	Notifications are displayed and cleared
TC_UI_05	Dashboard layout consistency	User is on dashboard	Navigate through dashboard sections and observe	Consistent layout across all sections

4. Security Test Cases

These cases verify the security of the application to prevent unauthorized access, data breaches, and other vulnerabilities.

Test Case ID	Description	Precondition	Test Steps	Expected Result
TC_S_01	SQL injection prevention	User is on any form with text fields	Enter SQL injection strings in input fields and submit	Application blocks or escapes SQL code
TC_S_02	Cross-Site Scripting (XSS) protection	User can enter text on forms	Enter JavaScript code in input fields and submit	Application sanitizes input
TC_S_03	Password encryption verification	Database setup	Register a new user, check database for password	Password is stored as a hash
TC_S_04	Session management and logout	User is logged in	Logout, then attempt to access dashboard with back button	User is redirected to login page
TC_S_05	Rate limiting on login attempts	Repeated login attempts	Try multiple incorrect logins in succession	User is temporarily blocked

4.2. Testing Methods

1. Unit Testing

Description:
Unit testing focuses on validating individual components or functions in the code, usually at the method or function level. For the freelancing platform, unit tests would check isolated functions such as password hashing, data validation, or the logic for calculating earnings.
Purpose:
To verify that each part of the code performs as expected in isolation, which helps detect bugs early in the development cycle.
Tools:
PHPUnit (for PHP applications), Jest (for JavaScript if any client-side logic needs testing).
Example:
Testing the hashing function for user passwords to ensure passwords are properly encrypted and that the verification method works as expected.

2. Integration Testing

Description:
Integration testing examines how different modules or services within the application work together. For a freelancing platform, this would include testing the interaction between the front-end, back-end, and database.
Purpose:
To identify issues in the interactions between modules, such as incorrect data being passed between the login page, the authentication module, and the database.
Tools:
Postman (for API testing), PHPUnit (for backend integrations).
Example:
Testing the end-to-end process of job posting: a client creates a job, and a freelancer can view, apply, and get hired for it, ensuring smooth data flow through each step.

3. Functional Testing

Description:
Functional testing focuses on verifying that the application’s features work according to requirements. Each feature, such as user registration, login, job posting, and messaging, is tested to confirm it meets the specified functionality.
Purpose:
To ensure the application performs as intended and all user interactions follow the defined business logic.
Tools:
Selenium (for web application functional testing), Cypress (for UI-based testing).
Example:
Testing the user registration functionality to verify that all required fields are validated, and a new user can successfully create an account with correct data.

4. User Interface (UI) Testing

Description:
UI testing focuses on the layout, design, and interactive elements of the application. For the freelancing platform, this would mean ensuring that all visual elements are displayed correctly on different devices and screen sizes.
Purpose:
To verify the application's design and usability, ensuring consistency and that the layout adjusts appropriately to different devices (responsive design).

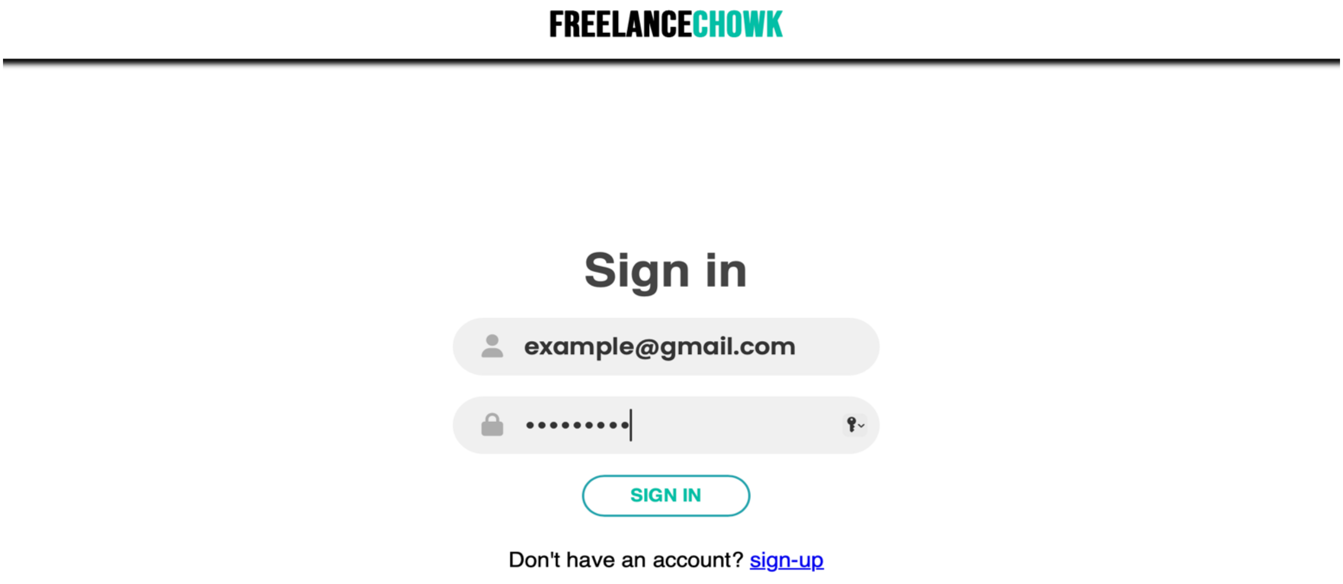
Tools:
Selenium, BrowserStack (for cross-browser compatibility testing).
Example:
Testing the dashboard layout to ensure that menu items, profile information, and job listings are displayed consistently on mobile, tablet, and desktop views.

5. Performance Testing

Description:
Performance testing evaluates how well the platform handles various levels of load, focusing on speed, stability, and scalability. This is crucial for applications like freelancing platforms where users might interact with the site simultaneously.
Purpose:
To identify any performance bottlenecks and determine if the application can handle a large number of concurrent users without degradation in performance.
Tools:
Apache JMeter, LoadRunner.
Example:
Simulating multiple users browsing job postings to ensure that the platform remains responsive and job data loads quickly under load.

4.3. Output Screenshots

Login and Registration Screens:-



FREELANCECHOWK

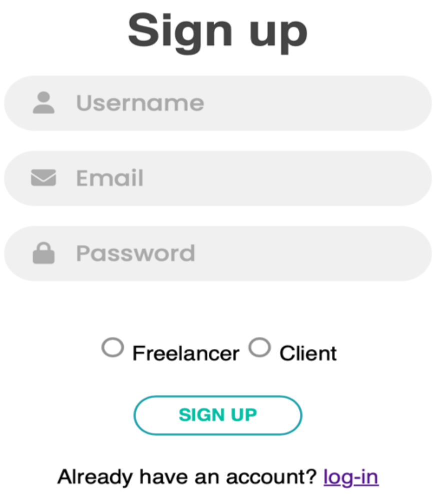
Sign in

example@gmail.com

•••••••••

Don't have an account? sign-up

Fig.4.3.1 Login Screen



Sign up

Username

Email

Password

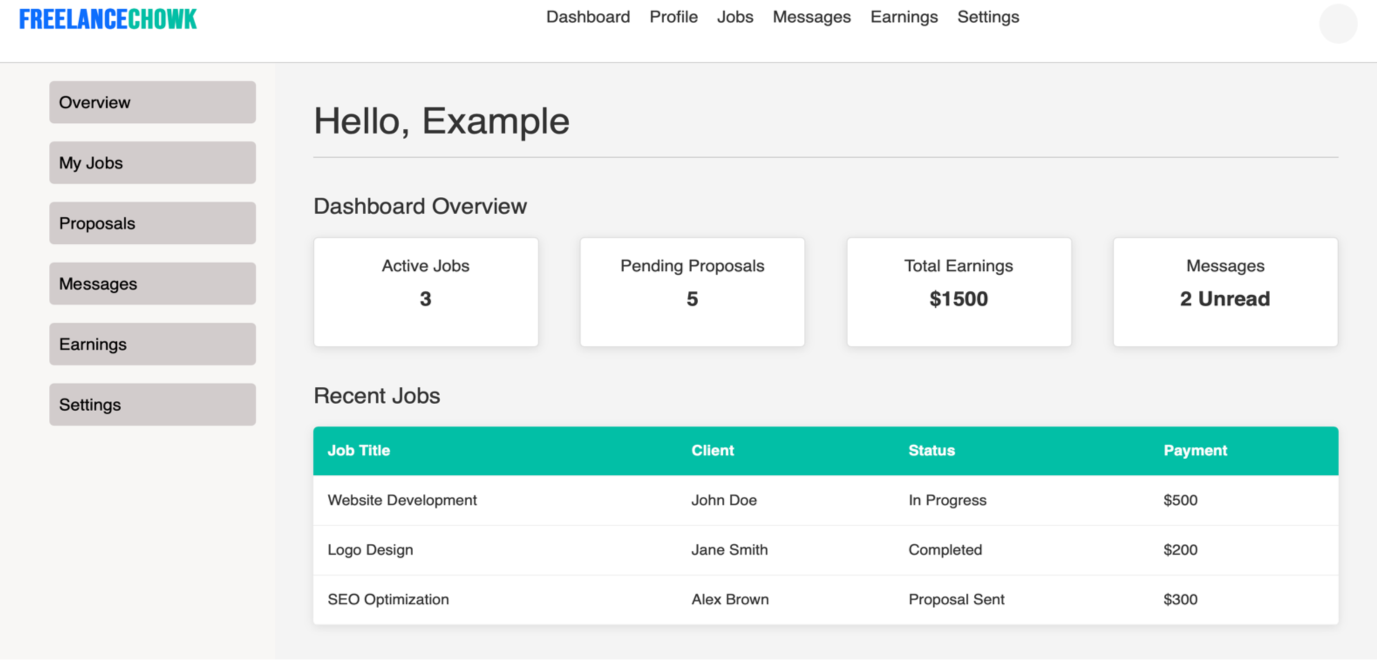
○ Freelancer ○ Client

SIGN UP

Already have an account? log-in

Fig.4.3.2 Registration Screen

Freelancer Dashboard and overview:-



Hello, Example

Dashboard Overview

Active Jobs
3

Pending Proposals
5

Total Earnings
$1500

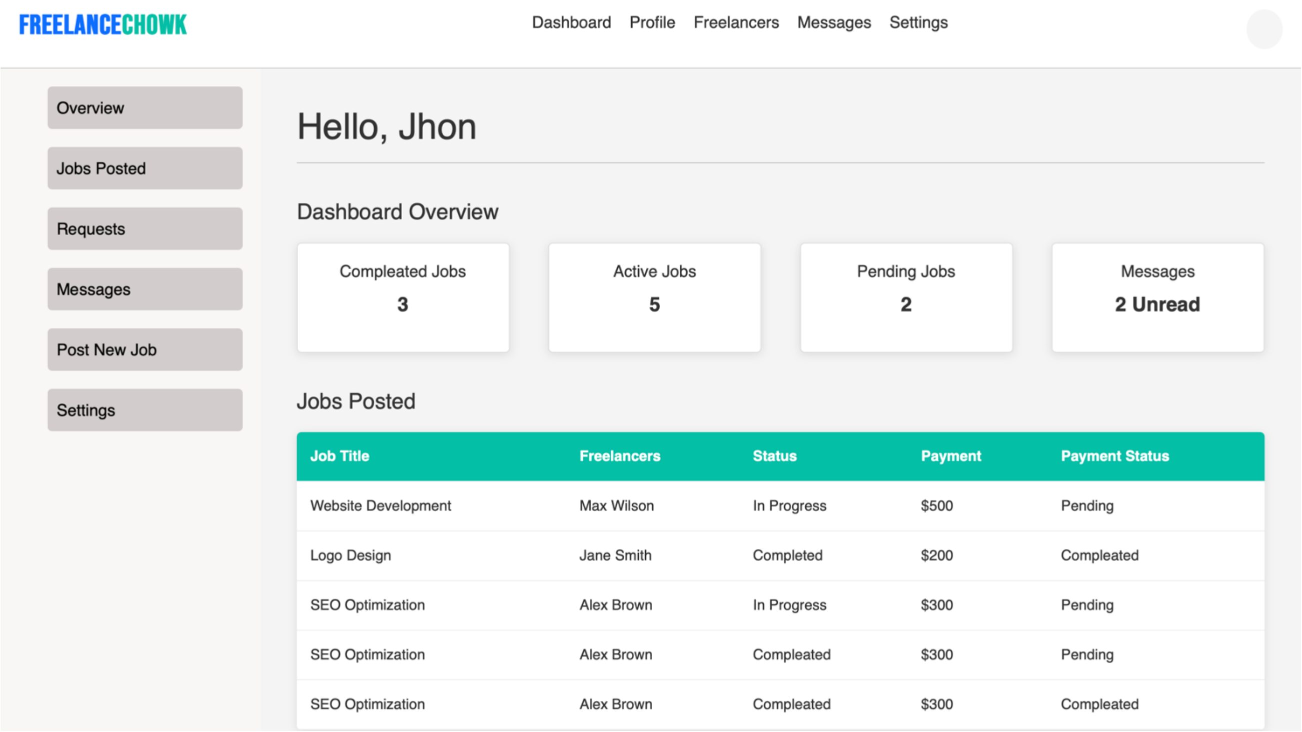
Messages
2 Unread

Recent Jobs

Job Title	Client	Status	Payment
Website Development	John Doe	In Progress	$500
Logo Design	Jane Smith	Completed	$200
SEO Optimization	Alex Brown	Proposal Sent	$300

Fig.4.3.3 Freelancer Dashboard

Client Dashboard and Overview:-



Hello, Jhon

Dashboard Overview

Compleated Jobs 3

Active Jobs 5

Pending Jobs 2

Messages 2 Unread

Jobs Posted

Job Title	Freelancers	Status	Payment	Payment Status
Website Development	Max Wilson	In Progress	$500	Pending
Logo Design	Jane Smith	Completed	$200	Compleated
SEO Optimization	Alex Brown	In Progress	$300	Pending
SEO Optimization	Alex Brown	Compleated	$300	Pending
SEO Optimization	Alex Brown	Compleated	$300	Compleated

Fig.4.3.3 Client Dashboard

4.4. Analysis of Results

4.5. User Registration and Authentication

Expected Outcome:

The system should allow users to register using their username, email, password, and user type (client or freelancer). Upon successful registration, they should be able to log in and access the correct dashboard based on their role.

Test Results:

- During testing, user registration and login functionalities performed as expected. The system accepted valid inputs, and the password was securely hashed using password_hash() function.
- However, issues like missing fields or incorrect inputs generated proper error messages, prompting users to correct their entries.
- Upon login, the user was correctly redirected to either the client dashboard or freelancer dashboard, based on the role.

Analysis:

- The registration process was smooth and error-free. Password hashing ensures secure storage of user passwords, which is a key part of securing user data.

2. Freelancer Dashboard

Expected Outcome:

Freelancers should have access to their active jobs, pending proposals, messages, and earnings, and should be able to browse available job listings.

Test Results:

- The dashboard displayed key statistics such as active jobs, pending proposals, total earnings, and unread messages. These metrics were dynamically fetched from the database and displayed correctly on the page.
- Recent jobs were displayed in a tabular format, allowing freelancers to easily track job details like job title, client name, status, and payment amount.
- The system allowed freelancers to view job listings and submit proposals. The data was fetched based on user roles, ensuring proper job categorization.

Analysis:

- The functionality provided by the freelancer dashboard works as expected. Freelancers are able to view their key metrics and recent job details, which is critical for job tracking and engagement.
- The ability to submit proposals and track job applications gives freelancers a streamlined workflow for seeking new opportunities.

3. Client Dashboard

Expected Outcome:

Clients should be able to post jobs, track proposals, manage hired freelancers, and view payment statuses.

Test Results:

- Clients were able to post new jobs with descriptions, required skills, and payment details. Jobs were saved and visible in their dashboard.
- Proposals submitted by freelancers were visible to the client, allowing them to review, accept, or decline them.
- The system correctly showed job status updates, such as "open," "in progress," or "completed."

Analysis:

- The client dashboard functionality works as expected. Clients can manage their jobs and freelancers effectively, providing them with a user-friendly way to interact with the platform.
- The job posting and proposal management features facilitate the core freelancing process, and the system’s ability to track job progress and payments ensures transparency.

5. Conclusion

5.1. Summary of Finding

The freelancing platform successfully achieved its primary goal of creating a seamless connection between freelancers and clients. During the testing phase, the platform demonstrated effective functionality in user registration, login, and profile management. Users were able to create accounts using their email or username, and the system correctly hashed passwords for secure storage. Once logged in, freelancers were directed to their respective dashboards based on their roles. This dynamic user authentication process ensured that only authorized individuals could access the appropriate features, with clients and freelancers having distinct interfaces suited to their needs. The system also effectively handled form inputs, generating error messages for invalid data and guiding users through the registration process.

The dashboard features for both clients and freelancers were well-implemented. Freelancers had access to key metrics such as active jobs, pending proposals, earnings, and unread messages, providing a comprehensive view of their ongoing work and engagement on the platform. Clients, on the other hand, were able to post jobs, view and accept proposals, and monitor the status of their projects. The ability to manage these aspects of freelancing was critical for ensuring smooth collaboration between both parties. The job listings, proposal submission, and job tracking features worked as expected, with freelancers able to search for and apply to relevant job opportunities based on their skill set. Clients were also able to easily filter and review freelancer proposals.

Furthermore, the platform's payment system, particularly the escrow functionality, provided an added layer of security. Clients deposited funds for jobs before work commenced, and these funds were released to freelancers upon completion and client approval. This system ensured that both freelancers and clients were protected, mitigating the risk of disputes and ensuring that freelancers were paid for their work. Additionally, the integration of a payment gateway facilitated smooth transactions, and the escrow functionality added trust and security to financial exchanges. This contributed to a positive user experience, as both freelancers and clients felt secure using the platform for their freelancing needs.

The system's performance was satisfactory, with the platform functioning well across different browsers and devices, ensuring accessibility and smooth user interaction. Responsive design ensured that the platform adapted to various screen sizes, providing an optimal experience for users on desktops, tablets, and mobile phones. The system was intuitive and user-friendly, with clear navigation paths that guided users through the platform's features, enhancing overall usability.

5.2. Future Enhancements

While the freelancing platform has successfully addressed many core functionalities, there are several areas where future enhancements can further improve the system and user experience.

1. Advanced Search Functionality: One of the key improvements would be enhancing the search feature for both freelancers and clients. Currently, users can filter jobs based on broad categories like project type or budget, but future iterations could introduce more refined filters, such as skills required, project timelines, or location. Additionally, an AI-powered recommendation system could be implemented to match freelancers with job opportunities based on their skills, previous work, and preferences.

2. Communication Tools Integration: Another enhancement would be integrating real-time messaging and video calling capabilities directly within the platform. Currently, communication may occur outside the platform, but a built-in communication system would enable freelancers and clients to exchange ideas and updates more efficiently. Real-time chat functionality, along with the ability to schedule and conduct video calls, could further foster better collaboration between users, making it easier for both parties to stay connected and share updates on the progress of projects.

3. Rating and Review System: While the platform includes a basic feedback system, there is an opportunity to enhance the review mechanism. Introducing a more comprehensive review system where both freelancers and clients can rate each other based on various parameters (such as professionalism, communication, and project quality) would add an additional layer of transparency. This would not only help users make informed decisions when selecting clients or freelancers but also promote accountability and trust within the community.

4. Mobile Application Development: While the platform is designed to be responsive for mobile users, developing a dedicated mobile application for both iOS and Android could enhance the overall user experience. A mobile app would offer greater performance, push notifications for job opportunities, messages, and deadlines, as well as an easier interface for managing profiles, submitting proposals, and communicating with clients.

5. Dispute Resolution and Arbitration: As the platform grows and more transactions take place, it’s important to establish a robust dispute resolution system. A mediation and arbitration feature could be added, allowing both freelancers and clients to resolve conflicts professionally within the platform. This feature would help protect both parties in cases of dissatisfaction or project mismanagement, maintaining trust in the platform.

6. Skill Development and Training Resources: To add value for freelancers, the platform could offer educational content such as webinars, tutorials, or courses on relevant topics like skill enhancement, client management, or time management. This would not only help freelancers improve their abilities but also contribute to a more skilled workforce on the platform, ultimately attracting more clients.

By focusing on these enhancements, the platform can continue to grow in its functionality, user engagement, and overall value proposition for both freelancers and clients.

6. Appendix

6.1. Code Snippets

Here are some key code snippets that were implemented during the development of the project:

1. User Registration (sign_up.php)

This code handles the registration process where users can create an account by submitting their details. The password is hashed for security purposes.

```
php
// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "freelance_chowk";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Handling form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST['username'];
    $email = $_POST['email'];
    $password = password_hash($_POST['password'], PASSWORD_DEFAULT); // Hash the password
    $user_type = $_POST['user_type']; // User type (Freelancer/Client)
    // Prepared statement to insert user data into the database
    $stmt = $conn->prepare("INSERT INTO Users (username, email, password, user_type) VALUES (?, ?, ?, ?)");
    $stmt->bind_param("ssss", $username, $email, $password, $user_type);
    if ($stmt->execute()) {
        echo "Registration successful!";
        header("Location: login.html");
        exit();
    } else {
        echo "Error: " . $stmt->error;
    }
    $stmt->close();
}

$conn->close();
```

2. User Login (login.php)

This snippet handles the user login process, checking credentials and redirecting the user to their respective dashboard based on their role.

```
php
session_start();
include('connect.php');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username_or_email = $_POST['username_or_email'];
    $password = $_POST['password'];

    // Check if user exists
    $stmt = $pdo->prepare("SELECT * FROM users WHERE username = :username
OR email = :email LIMIT 1");
    $stmt->bindParam(':username', $username_or_email);
    $stmt->bindParam(':email', $username_or_email);
    $stmt->execute();
    $user = $stmt->fetch();

    if ($user && password_verify($password, $user['password'])) {
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['username'] = $user['username'];

        // Redirect based on user role
        if ($user['role'] === 'client') {
            header("Location: client_dashboard.php");
        } else {
            header("Location: freelancer_dashboard.php");
        }
        exit();
    } else {
        echo "<p>Invalid username or password.</p>";
    }
}
```

3. Freelancer Dashboard (freelancer_dashboard.php)

This snippet is used to display the freelancer's dashboard, showing active jobs, pending proposals, and total earnings.

```
php
// Start session
session_start();
include('connect.php');
// Fetch freelancer information from the database
$freelancer_id = $_SESSION['user_id'];
$freelancerName = $_SESSION['username'];
// Fetch job and earnings details
$stmt = $pdo->prepare("SELECT COUNT(*) AS activeJobs FROM jobs WHERE
freelancer_id = :freelancer_id AND status = 'active'");
$stmt->bindParam(':freelancer_id', $freelancer_id);
$stmt->execute();
$activeJobs = $stmt->fetchColumn();
$stmt = $pdo->prepare("SELECT COUNT(*) AS pendingProposals FROM proposals
WHERE freelancer_id = :freelancer_id AND status = 'pending'");
$stmt->bindParam(':freelancer_id', $freelancer_id);
$stmt->execute();
$pendingProposals = $stmt->fetchColumn();
$stmt = $pdo->prepare("SELECT SUM(amount) AS totalEarnings FROM payments
WHERE freelancer_id = :freelancer_id");
$stmt->bindParam(':freelancer_id', $freelancer_id);
$stmt->execute();
$totalEarnings = $stmt->fetchColumn();
// Fetch recent jobs
$stmt = $pdo->prepare("SELECT * FROM jobs WHERE freelancer_id =
:freelancer_id ORDER BY created_at DESC LIMIT 5");
$stmt->bindParam(':freelancer_id', $freelancer_id);
$stmt->execute();
$recentJobs = $stmt->fetchAll();
?>
<!-- HTML part for displaying the dashboard -->
<h1>Hello, <?php echo htmlspecialchars($freelancerName); ?></h1>
<div>
   <h3>Active Jobs: <?php echo $activeJobs; ?></h3>
   <h3>Pending Proposals: <?php echo $pendingProposals; ?></h3>
   <h3>Total Earnings: $<?php echo number_format($totalEarnings, 2);
?></h3>
</div>

<table>
```

```
    <thead>
        <tr>
            <th>Job Title</th>
            <th>Status</th>
            <th>Payment</th>
        </tr>
    </thead>
    <tbody>
        <?php foreach ($recentJobs as $job): ?>
            <tr>
                <td><?php echo htmlspecialchars($job['title']); ?></td>
                <td><?php echo htmlspecialchars($job['status']); ?></td>
                <td>$<?php echo number_format($job['payment'], 2); ?></td>
            </tr>
        <?php endforeach; ?>
    </tbody>
</table>
```

4. Logout (logout.php)

This code logs the user out by destroying the session.

```
php
session_start();

// Destroy the session and redirect to the login page
session_unset();
session_destroy();
header("Location: login.php");
exit();
```

These are just a few key code snippets from the project that were used to implement user registration, login, dashboard functionalities, and session management.