

Project Report Format

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. IDEATION PHASE

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

4. PROJECT DESIGN

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

7. RESULTS

7.1 Output Screenshots

8. ADVANTAGES & DISADVANTAGES

9. CONCLUSION

10. FUTURE SCOPE

11. APPENDIX

Source Code(if any)

Dataset Link

GitHub & Project Demo Link

1.INTRODUCTION

1.1 Project Overview:

FreelanceFinder is a MERN-stack based web application designed to connect clients and freelancers through a centralized digital platform. The system allows users to register, log in securely, and access role-based dashboards. Clients can post projects, review proposals, and monitor work progress efficiently. Freelancers can browse available projects, submit proposals, and upload completed work through the platform. The application integrates project management, communication, and submission tracking into a single workflow. It uses React.js for the frontend interface, Node.js and Express.js for backend logic, and MongoDB for data storage. Secure authentication and authorization ensure safe user access. The platform aims to simplify freelancing processes and improve collaboration between users. Its modular architecture supports scalability and future enhancements. Overall, FreelanceFinder provides a user-friendly and efficient solution for managing online freelance projects.

1.2 Purpose:

The purpose of the FreelanceFinder project is to provide a centralized platform that simplifies the interaction between freelancers and clients. It aims to make project posting, proposal management, and work submission more efficient through a single web application. The system helps freelancers easily find suitable opportunities and manage their tasks effectively. Clients can monitor project progress and communicate with freelancers in a structured way. The platform focuses on improving user experience through secure authentication and role-based dashboards. It reduces manual effort by organizing project workflows digitally. The project also demonstrates practical implementation of the MERN stack in real-world application development. Overall, the goal is to create a scalable and user-friendly freelancing environment that enhances collaboration and productivity.

Key objectives of the system include:

- To develop a centralized MERN-stack platform that connects freelancers and clients in a single system.
- To enable secure user registration, login, and role-based access for different types of users.
- To simplify project posting, proposal submission, and work tracking through structured workflows.
- To provide an intuitive dashboard that improves user experience and productivity.
- To ensure secure data handling using authentication and authorization mechanisms.
- To reduce manual communication gaps by integrating project management features.
- To build a scalable architecture that supports future enhancements and growth.
- To demonstrate practical full-stack web development using React.js, Node.js, Express.js, and MongoDB

2. IDEATION PHASE

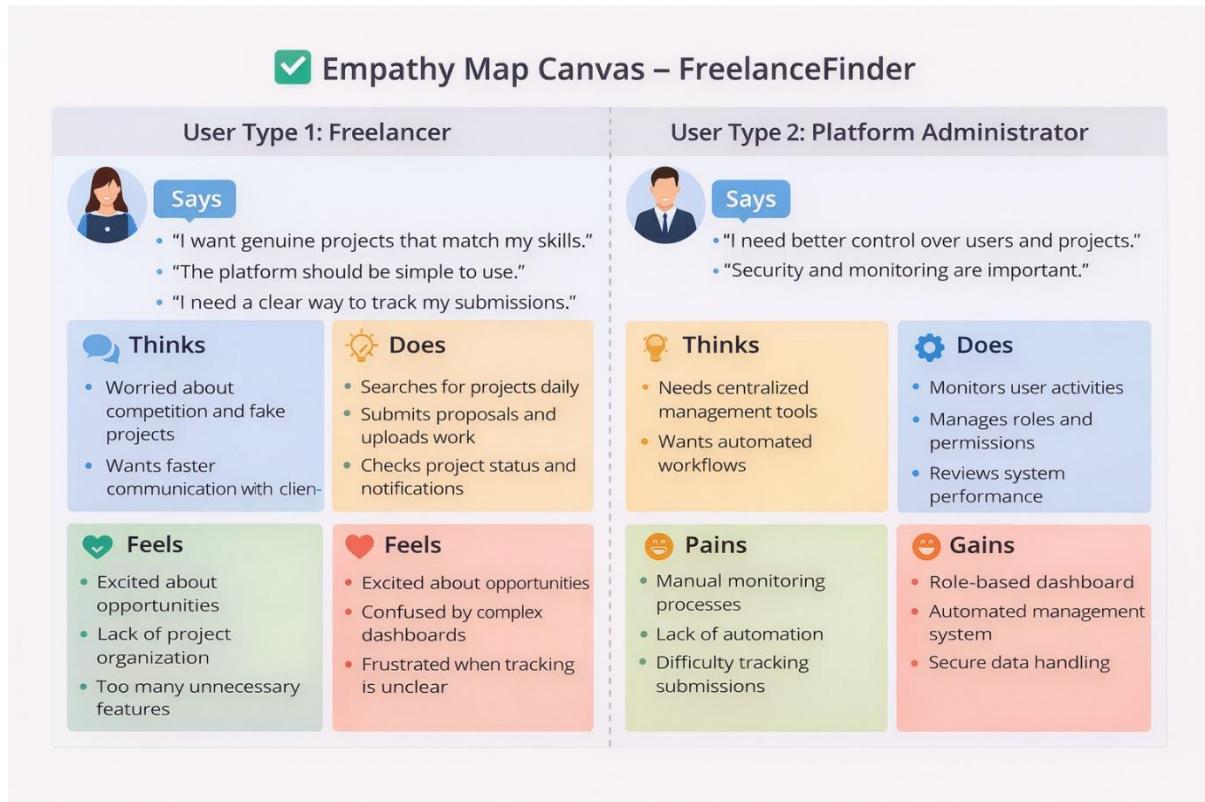
2.1 Problem Statement:

The freelancing process is often fragmented across multiple platforms, making it difficult for clients and freelancers to manage projects efficiently. Freelancers struggle to find reliable opportunities and track their submissions, while clients face challenges in monitoring proposals and communication. Existing systems can be complex and lack a simple, centralized workflow. There is a need for a secure and user-friendly platform that integrates project posting, proposal management, and work submission in one place. The FreelanceFinder project aims to solve these issues by providing a streamlined and organized freelancing environment.

| Problem Statement (PS) | I am (Customer) | I'm trying to | But | Because | Which makes me feel |
|-----------------------------|--|--|--|---|---|
| Problem Statement -1 | A freelancer looking for reliable projects online. | Find suitable work opportunities , submit proposals, and manage my tasks efficiently in one place. | Many existing platforms are complex, crowded, or not beginner-friendly for students and new freelancers. | They include unnecessary features, high competition , and lack simple project tracking tools. | Frustrated, confused, and less confident about finding genuine opportunities. |
| Problem Statement-2 | A platform administrator or system manager. | Maintain secure user access, manage project data, and monitor system activities efficiently. | Manual monitoring and lack of centralized control make it difficult to manage users and submissions. | Traditional systems lack automated workflows and structured role-based management. | Stressed about maintaining platform security and performance. |

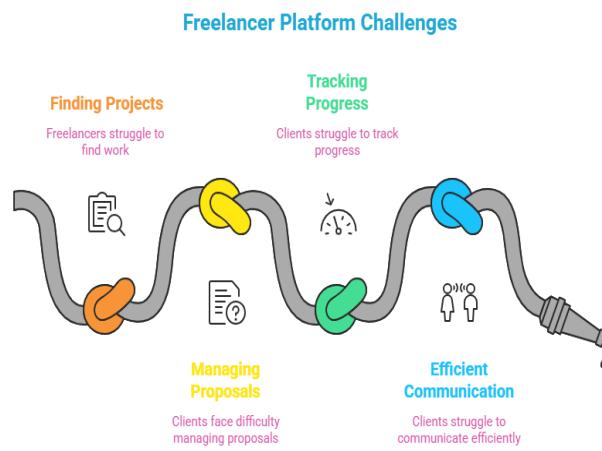
2.2 Empathy Map:

User: - Freelancer and Platform Administrator

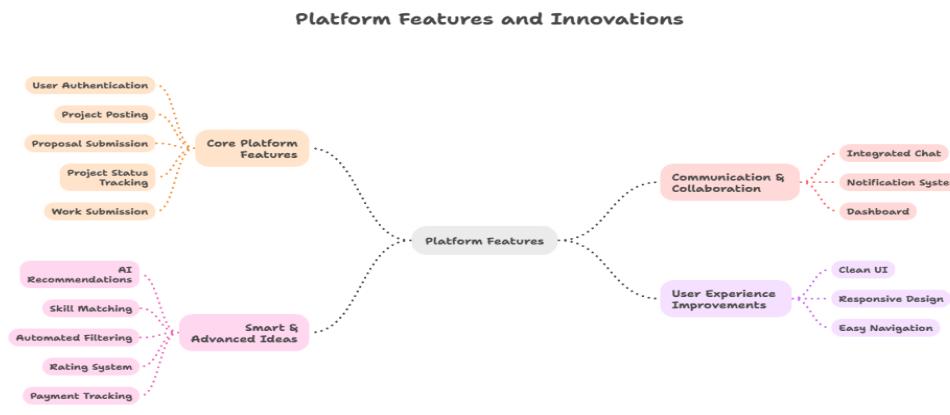


2.2 Brainstorm & Idea Prioritization:-

Step-1: Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization

Project Features



3. REQUIREMENT ANALYSIS

3.2 Solution Requirements

Functional Requirements:

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|--------|-------------------------------|--|
| FR-1 | User Registration | Registration through Form Registration through Gmail Registration through LinkedIn |

| | | |
|-------------|-------------------------------|--|
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | Project & Proposal Management | Client can post projects Freelancer can browse projects Freelancer can submit proposals Client can accept/reject proposals |
| FR-4 | Work Submission & Tracking | Freelancer uploads completed work Client views submissions System updates submission status |
| FR-5 | Communication & Notifications | Messaging between client and freelancer Notification alerts for updates |
| FR-6 | Admin Management | Admin manages users Admin monitors projects and submissions |

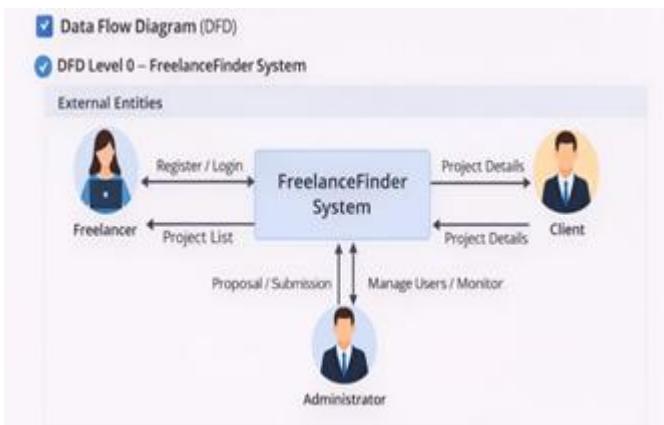
Non-functional Requirements:

| NFR No. | Non-Functional Requirement | Description |
|---------|----------------------------|--|
| NFR-1 | Usability | The system provides a simple, responsive, and user-friendly interface for freelancers, clients, and administrators. |
| NFR-2 | Security | Role-based authentication, secure login, and protected API endpoints ensure data privacy and system safety. |
| NFR-3 | Reliability | The platform maintains stable performance and prevents data loss during project or submission handling. |
| NFR-4 | Performance | Fast loading dashboards, optimized database queries, and efficient API responses improve user experience. |
| NFR-5 | Availability | The system is accessible anytime with minimal downtime for continuous project management. |
| NFR-6 | Scalability | The MERN architecture supports increasing users, projects, and submissions without performance degradation. |

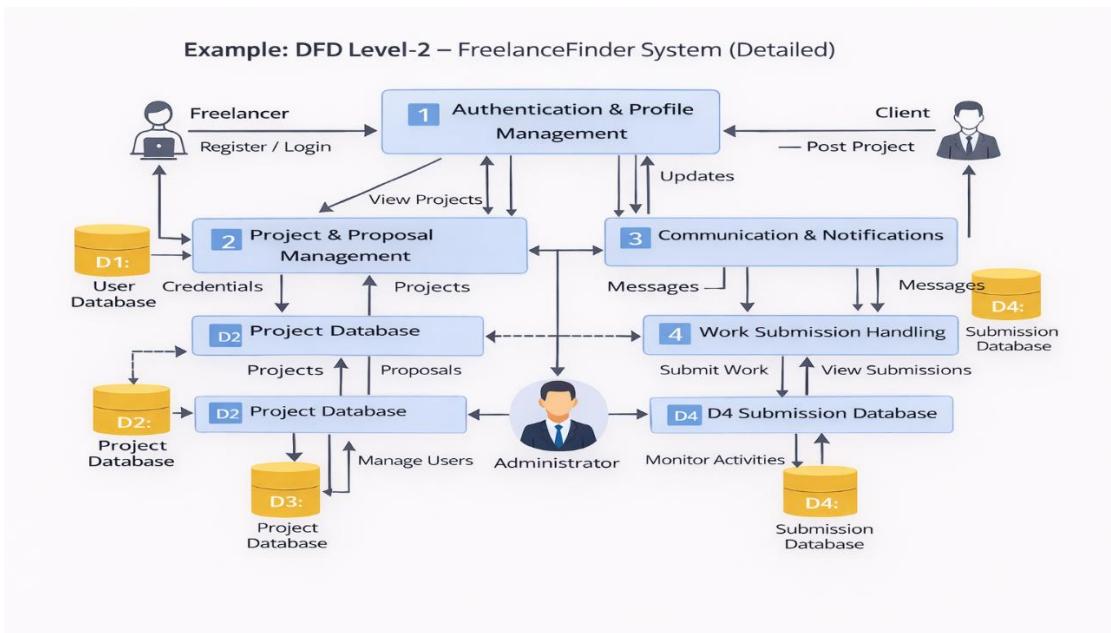
3.2 Data Flow Diagram: -

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

DFD Level-1



DFD Level-2



3.3 Technology Stack: -

FreelanceFinder platform follows a 3-Tier MERN Architecture consisting of a React-based frontend, Node.js/Express backend APIs, and MongoDB database. The user interface interacts with backend services through REST APIs. Application logic handles authentication, project posting, proposals, messaging, and submission tracking. Data is stored in MongoDB collections, and file uploads are handled through server storage. The system can be deployed locally or on cloud infrastructure with scalable server configuration.

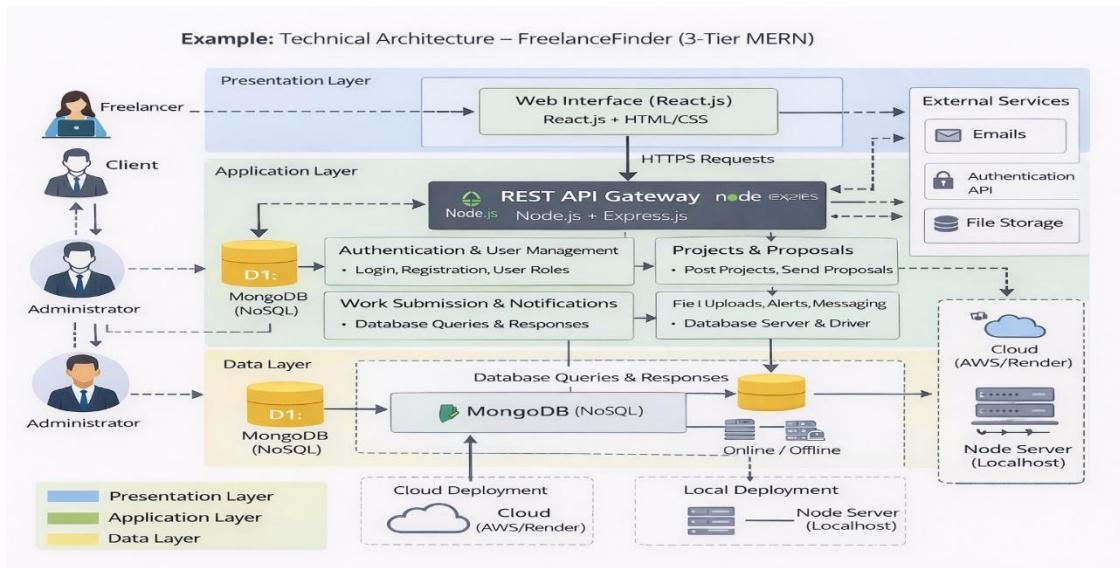


Table-1: Components & Technologies:

| S.No | Component | Description | Technology |
|------|------------------------|---|---------------------------------|
| 1 | User Interface | Web-based dashboard for freelancers, clients, and admin | React.js, HTML, CSS, JavaScript |
| 2 | Application Logic-1 | Authentication & User Management | Node.js, Express.js |
| 3 | Application Logic-2 | Project & Proposal Management | Node.js REST APIs |
| 4 | Application Logic-3 | Work Submission & Notifications | Express.js Middleware |
| 5 | Database | Stores users, projects, proposals, submissions | MongoDB (NoSQL) |
| 6 | Cloud Database | Optional hosted DB | MongoDB Atlas |
| 7 | File Storage | Stores uploaded work files | Local Storage / Cloud Storage |
| 8 | External API-1 | Email notifications | Nodemailer / SMTP Service |
| 9 | External API-2 | Authentication services | JWT Authentication |
| 10 | Machine Learning Model | Not Applicable (Optional future enhancement) | — |

| | | | |
|----|------------------------------------|------------------------|---|
| 11 | Infrastructure (Server / Cloud) | Application deployment | Localhost, Node Server, Cloud (AWS/Render) |
|----|------------------------------------|------------------------|---|

Table-2: Application Characteristics:

| S.No | Characteristics | Description | Technology |
|------|--------------------------|--|--|
| 1 | Open-Source Frameworks | MERN stack open-source technologies | React.js, Node.js, Express.js, MongoDB |
| 2 | Security Implementations | Secure login, token authentication, protected routes | JWT, Password Hashing, Role-based Access |
| 3 | Scalable Architecture | Modular backend APIs with separate frontend | 3-Tier MERN Architecture |
| 4 | Availability | Continuous system access via web deployment | Cloud Hosting / Node Server |
| 5 | Performance | Optimized API calls and database queries | REST APIs, MongoDB Indexing |

4. PROJECT DESIGN

Problem – Solution Fit Template:

The FreelanceFinder platform addresses the difficulties faced by freelancers and clients in managing projects through complex and unorganized systems. Users often struggle with communication gaps, tracking proposals, and monitoring submissions efficiently. The proposed MERN-based solution provides a centralized platform with secure authentication, project management, and role-based dashboards. By simplifying workflows and integrating all features into a single system, the solution improves usability, enhances collaboration, and ensures better control for administrators.

◆ Target Problem

- Difficulty in finding and managing freelance projects
- Lack of centralized communication and tracking
- Manual monitoring and inefficient workflows

◆ Proposed Solution

- MERN-based freelancing platform
- Integrated dashboards for Freelancer, Client, and Admin
- Secure authentication and real-time updates

◆ Expected Benefits

- Faster project collaboration
- Improved user experience
- Better platform control and scalability

4.2 Proposed Solution :-

| S.No. | Parameter | Description |
|-------|---|--|
| 1 | Problem Statement (Problem to be solved) | Freelancers and clients face difficulties using complex and scattered platforms for project management, communication, and work tracking. There is a need for a centralized system that simplifies project posting, proposal handling, and submission monitoring. |
| 2 | Idea / Solution Description | FreelanceFinder is a MERN-stack based freelancing platform that connects freelancers and clients through role-based dashboards. The system allows users to register, post projects, submit proposals, upload work, and communicate in real time within a single web application. |
| 3 | Novelty / Uniqueness | The platform integrates project management, proposal tracking, and work submission into one simplified workflow designed for students and beginner freelancers. It focuses on usability, secure authentication, and structured dashboards for different roles. |
| 4 | Social Impact / Customer Satisfaction | The system helps freelancers discover opportunities easily while allowing clients to manage tasks efficiently. It promotes digital collaboration, improves productivity, and creates a reliable freelancing environment that enhances user satisfaction. |
| 5 | Business Model (Revenue Model) | Possible revenue models include commission on project completion, premium subscriptions for advanced features, featured project listings, and enterprise-level admin tools. |
| 6 | Scalability of the Solution | Built using MERN architecture, the system supports scalable APIs, cloud deployment, and modular expansion. It can handle increasing users, projects, and submissions without significant performance issues. |

4.3 Solution Architecture :-

Solution Architecture:

The FreelanceFinder platform follows a 3-tier web architecture that connects business requirements with technology solutions. The system is designed to provide a centralized freelancing environment where freelancers, clients, and administrators interact through secure web interfaces.

● Frontend (Presentation Layer):

Built using React.js, HTML, CSS, and JavaScript. Provides role-based dashboards for freelancers, clients, and administrators to register, post projects, submit proposals, and track work.

● Backend (Application Layer):

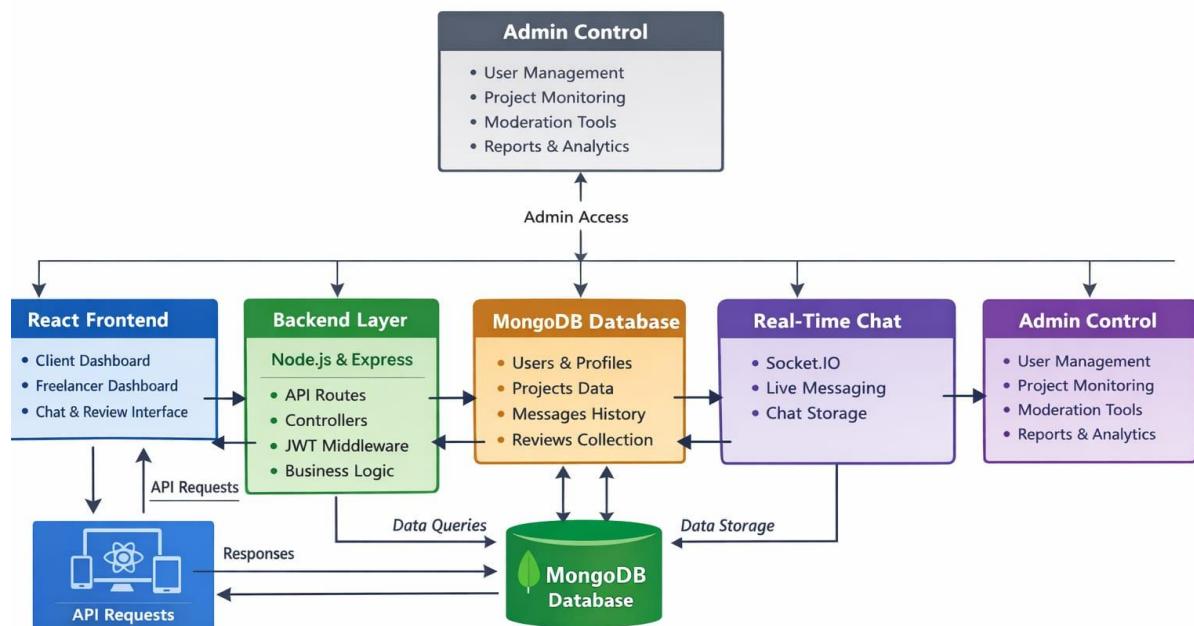
Developed with Node.js and Express.js. Handles authentication, project management, proposal processing, communication, and work submission through REST APIs.

● Database (Data Layer):

Uses MongoDB to store user profiles, project details, proposals, and submission data securely.

Solution Architecture Diagram:

Fig 1 :- System Architecture



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning : -

Product Backlog & Sprint Planning (ShopSmart)

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
|----------|-------------------------------|-------------------|--|--------------|----------|---------------|
| Sprint-1 | Registration | USN-1 | As a user, I can register using email and password | 2 | High | Jyothsna Devi |
| Sprint-1 | Registration | USN-2 | As a user, I receive confirmation after registration | 1 | High | Jyothsna Devi |
| Sprint-1 | Registration | USN-3 | As a user, I can register through Gmail | 2 | Medium | Jyothsna Devi |
| Sprint-1 | Login | USN-4 | As a user, I can log into the application | 1 | High | Jyothsna Devi |
| Sprint-1 | Dashboard | USN-5 | As a user, I can view my role-based dashboard | 3 | High | Jyothsna Devi |
| Sprint-2 | Project Management | USN-6 | As a client, I can post a project | 3 | High | Jyothsna Devi |
| Sprint-2 | Project Management | USN-7 | As a freelancer, I can browse projects | 2 | High | Jyothsna Devi |
| Sprint-2 | Proposal System | USN-8 | As a freelancer, I can submit proposals | 3 | High | Jyothsna Devi |
| Sprint-2 | Proposal System | USN-9 | As a client, I can accept or reject proposals | 3 | High | Jyothsna Devi |
| Sprint-2 | Work Submission | USN-10 | As a freelancer, I can upload completed work | 3 | High | Jyothsna Devi |
| Sprint-2 | Dashboard | USN-11 | As a client, I can view submissions | 3 | High | Jyothsna Devi |

Sprint Tracker (Velocity Table)

| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
|----------|--------------------|----------|-------------------|---------------------------|---|------------------------------|
| Sprint-1 | 14 | 5 Days | 05 Feb 2026 | 09 Feb 2026 | 14 | 09 Feb 2026 |
| Sprint-2 | 22 | 5 Days | 10 Feb 2026 | 14 Feb 2026 | 22 | 14 Feb 2026 |
| Sprint-3 | 18 | 5 Days | 15 Feb 2026 | 19 Feb 2026 | — | — |
| Sprint-4 | 18 | 5 Days | 20 Feb 2026 | 24 Feb 2026 | — | — |

Velocity Calculation

Velocity represents how much work the team completes in one sprint.

For the FreelanceFinder project:

- Sprint Duration = 5 Days
- Team Velocity = 18 Story Points per Sprint

● Average Velocity (AV) per Day

Formula:

$$\text{Average Velocity (AV)} = \text{Story Points per Sprint} \div \text{Sprint Duration}$$

$$AV = 18 \div 5$$

$$\text{Average Velocity} = 3.6 \text{ Story Points per Day}$$

Burndown Chart :

A Burndown Chart is a graphical representation that shows the remaining work versus time during a sprint. It helps track progress and ensures that tasks are completed within the planned schedule.

- ◆ Burndown Chart for FreelanceFinder

The X-axis represents the sprint timeline (Day 1 to Day 5).

The Y-axis represents remaining story points.

At the start of the sprint, the total story points are at the highest level.

As tasks such as registration, project posting, proposals, and submissions are completed, the remaining work decreases gradually.

By the final sprint day, the line ideally reaches zero, showing successful completion of planned tasks.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Test Scenarios & Results

| Test Case ID | Scenario (What to test) | Test Steps (How to test) | Expected Result | Actual Result | Pass/Fail |
|--------------|------------------------------|--|--|---|-----------|
| FT-01 | User Registration Validation | Enter valid and invalid email/password during signup | Valid registration succeeds, errors shown for invalid inputs | User account created successfully and validation messages displayed correctly | Pass |
| FT-02 | Login Authentication | Enter correct and incorrect login credentials | Dashboard opens for valid login, error for wrong details | Login successful and dashboard loaded without errors | Pass |
| FT-03 | Project Posting | Client fills project form and submits | Project stored and visible in project list | Project created and displayed successfully in project page | Pass |
| FT-04 | Proposal Submission | Freelancer submits proposal for a project | Proposal saved and visible to client | Proposal submitted and visible in client dashboard | Pass |
| FT-05 | Work File Upload | Freelancer uploads completed work file | File uploads successfully and appears in client dashboard | File uploaded successfully and displayed in submissions section | Pass |
| FT-06 | Role-Based Access Control | Login as freelancer and client separately | Each role sees correct dashboard features | Correct dashboards displayed for freelancer and client roles | Pass |
| PT-01 | API Response Time Test | Perform login or project fetch API call | Response should be | API response received within | Pass |

| | | | | | |
|--------------|------------------------------|---|---|--|------|
| | | | under 3 seconds | acceptable time ($\approx 1\text{--}2$ sec) | |
| PT-02 | Server Load Test | Open multiple dashboards simultaneously | System should remain stable without crashes | Application handled multiple requests without errors | Pass |
| PT-03 | File Upload Performance Test | Upload multiple submission files | Upload completes without errors or delays | Multiple files uploaded successfully without crash | Pass |

Testing Environment:

URL/Location: <http://localhost:3000>

Credentials: Test user login details

Test Cases:

| Test Case ID | Test Scenario | Test Steps | Expected Result | Actual Result | Pass/Fail |
|--------------|-----------------|---------------------------------------|----------------------|--------------------------------------|-----------|
| TC-001 | User Login | Enter valid credentials → Click Login | Dashboard opens | Dashboard loaded successfully | Pass |
| TC-002 | Post Project | Fill project form → Submit | Project created | Project visible in list | Pass |
| TC-003 | Submit Proposal | Freelancer submits proposal | Client sees proposal | Proposal visible in client dashboard | Pass |
| TC-004 | Upload Work | Upload submission file | File saved | File displayed in submission page | Pass |

Bug Tracking:

| Bug ID | Bug Description | Steps to reproduce | Severity | Status | Additional feedback |
|--------|---------------------------------------|--------------------|----------|--------|----------------------------|
| BG-001 | No critical bugs found during testing | — | Low | Closed | System working as expected |

7. RESULTS

7.1 Output Screenshots

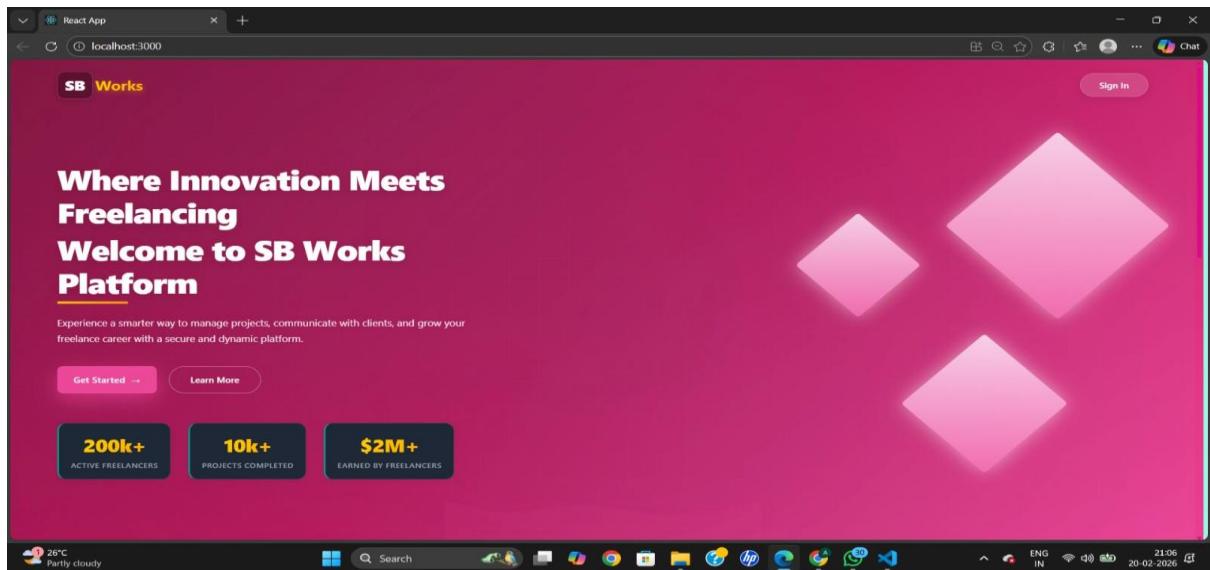


Fig : User Interface

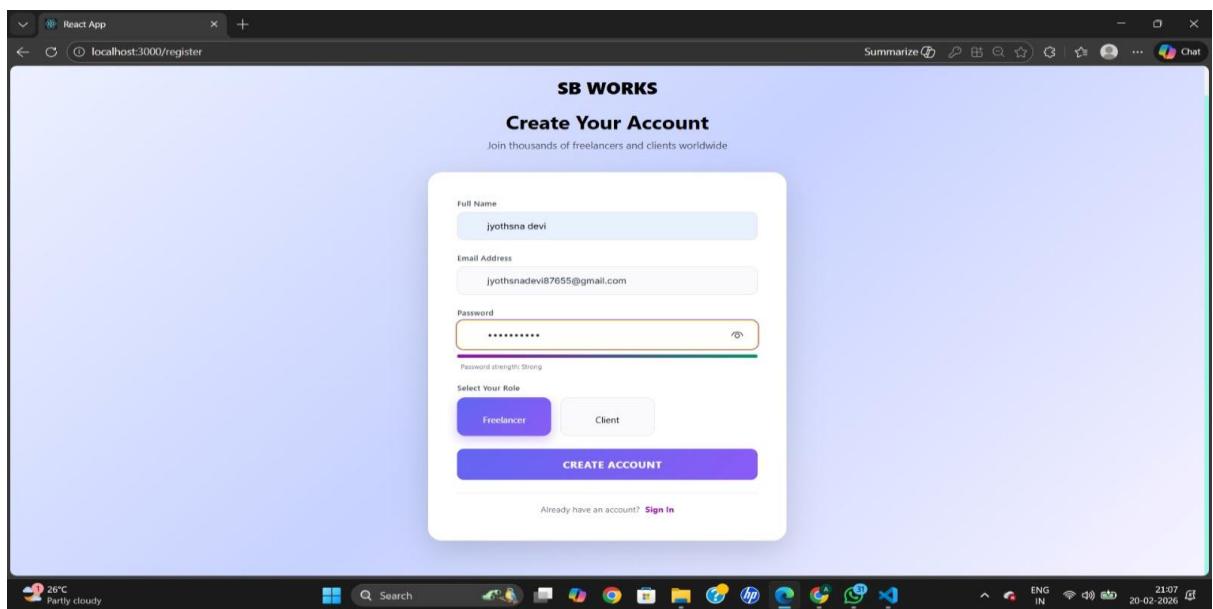


Fig : Register Page

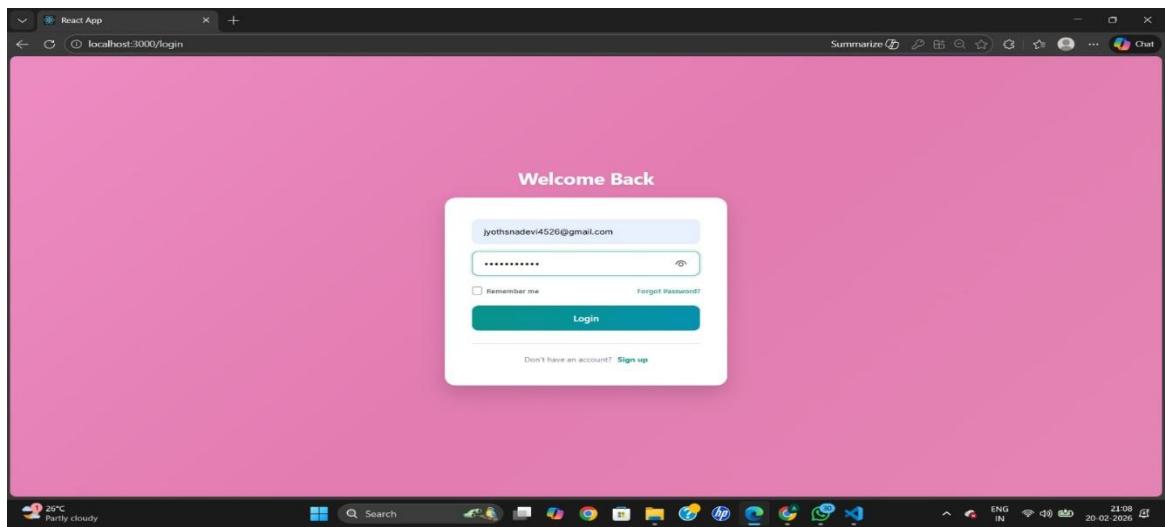


Fig : Login Page

Fig : Freelancer Dashboard

Fig : Client Dashboard

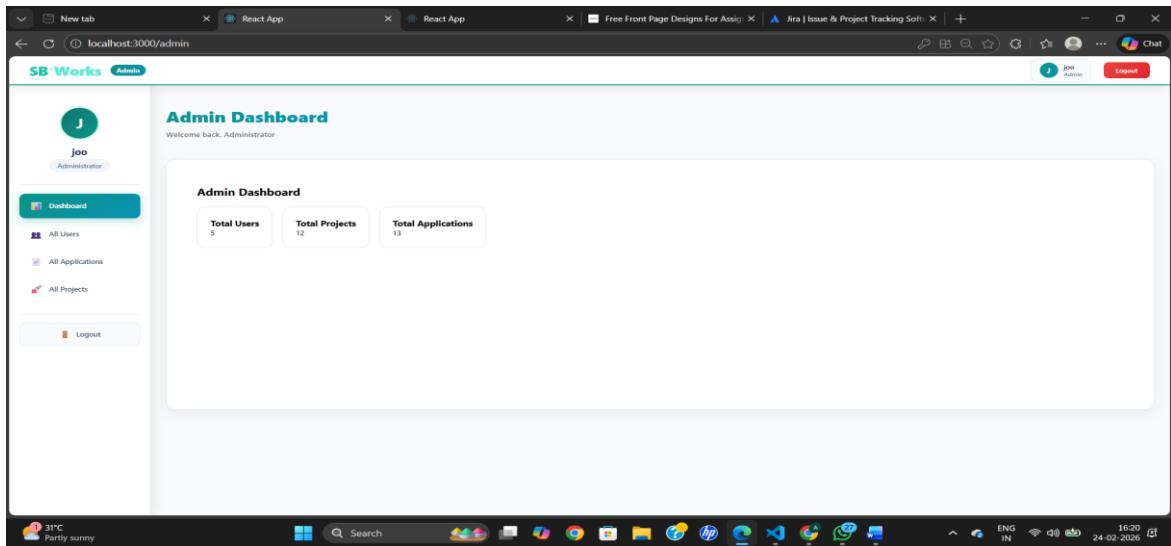


Fig : Admin dashboard

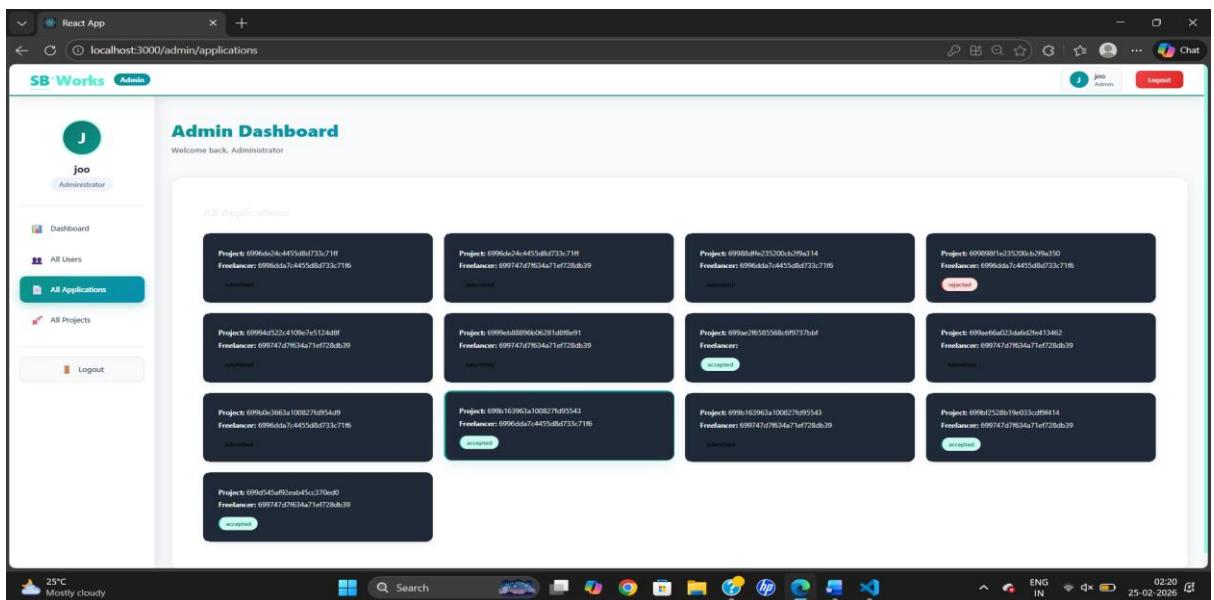


Fig : Project Information in Admin page

The screenshot shows the Admin Dashboard of a web application. On the left, there's a sidebar with a profile icon (J), the name 'joo', and the role 'Administrator'. Below it are buttons for 'Dashboard', 'All Users' (which is selected and highlighted in teal), 'All Applications', 'All Projects', and 'Logout'. The main content area is titled 'Admin Dashboard' and 'Welcome back, Administrator'. It features a table titled 'All Users' with columns for NAME, EMAIL, and ROLE. The data is as follows:

| NAME | EMAIL | ROLE |
|---------|---------------------------------|------------|
| jyooo | anunaram233@gmail.com | Freelancer |
| Ammu | tataqudilurghbhwa1138@gmail.com | Freelancer |
| joo | anunaram2003@gmail.com | Admin |
| ammu | bhwantatapud3@gmail.com | Client |
| karthik | karthiklowun1438@gmail.com | Freelancer |

The bottom of the screen shows a Windows taskbar with various icons and system status.

Fig : Users and Client Information in Admin page

The screenshot shows the Client Working interface. On the left, there's a sidebar with 'Panel', 'Project', 'Connections', and 'Projects' (which is selected and highlighted in teal). The main content area has a dark background. At the top right, there's a message bubble from a client named 'ammu' with the message 'hai'. Below it is a text input field with placeholder 'Type message...' and a 'Send ➤' button. Further down, there's a section for 'Freelancer Submission' with a status message 'Status: approved'. There's also a 'Rate Freelancer' section where a 5-star rating is selected, and a feedback input field with placeholder 'Write feedback...'. A 'Submit Rating' button is at the bottom of this section. The bottom of the screen shows a Windows taskbar with various icons and system status.

Fig : Real -time chat (Client & Freelancer) And Review by client

8. ADVANTAGES & DISADVANTAGES

Advantages of Freelancing

- Provides flexible work opportunities for freelancers from different locations.
- Helps clients quickly find skilled professionals for their projects.
- Centralized platform makes project management and communication easier.
- Role-based dashboards improve task tracking and productivity.
- Secure login and structured workflow reduce confusion and manual effort.
- Saves time by integrating project posting, proposals, and submissions in one system.

Disadvantages of the ShopSmart Project

- High competition among freelancers may make project selection challenging.
- Lack of face-to-face communication can sometimes cause misunderstandings.
- Freelancers may face inconsistent income depending on project availability.
- Clients need to carefully review proposals to choose the right candidate.
- System performance depends on stable internet and platform reliability.

9. CONCLUSION

The FreelanceFinder project was developed to create a centralized and user-friendly freelancing platform that simplifies the way clients and freelancers interact with each other. During the development process, we focused on building a system that allows users to register securely, manage projects, submit proposals, and track work progress through structured dashboards. By using the MERN stack, we were able to understand how frontend, backend, and database technologies work together to create a complete web application. This project not only helped in improving technical skills such as React development, API creation, and MongoDB data handling, but also enhanced problem-solving, planning, and testing abilities. One of the main achievements of this system is that it combines multiple freelancing activities into a single platform, reducing confusion and improving workflow efficiency. Throughout the phases of ideation, requirement analysis, design, development, and testing, we learned the importance of user-centric design and secure implementation. Although the current version provides essential features, there is still scope for future improvements such as advanced search, payment integration, and real-time notifications. Overall, the FreelanceFinder project demonstrates how modern web technologies can be used to build scalable, practical, and efficient solutions for real-world freelancing challenges while providing valuable learning experience in full-stack development.

10. FUTURE SCOPE

Although the FreelanceFinder platform successfully implements core freelancing functionalities, several improvements can enhance scalability, security, and user experience in future versions.

- Payment gateway integration for secure online transactions.
- Real-time notification system for messages, project updates, and reviews.
- Advanced analytics dashboard for admin monitoring and platform insights.
- AI-based freelancer recommendation based on skills and ratings.
- Video call feature between client and freelancer for live discussions.
- Secure token storage using HTTP-only cookies instead of localStorage.
- Performance optimization through caching and pagination.
- Mobile application version using React Native or Flutter.

11. APPENDIX

My Project Source code Files are available at :

https://drive.google.com/drive/folders/15gVc2thCpXiq06EWhkKiOYjwuIvqBA5z?usp=drive_link

My project Demo Video link is available at :

https://drive.google.com/drive/folders/1fdxppY3AmzaJhRJSIxANBAxCdstc5Uqe?usp=drive_link

Github Resopitory Link :

https://github.com/jyothsna2617/Freelancing_Sbworks_Platform_Design