
PROJECT REPORT

ON

FreelanceFinder: Discovering Opportunities, Unlocking Potential

BY

Kata Bharath(Team Leader)

Athuru Davidraj(Team Member)

Shaik Davoodh(Team Member)

Raja Dhanasekhar (Team Member)

1. INTRODUCTION

1.1 Project Overview

FreelanceFinder is a MERN-stack web application that bridges the gap between freelancers and clients. The platform enables freelancers to register, view projects, apply for jobs, and communicate through secure chat. Clients can post projects, review applications, and manage work submissions. Admins monitor the overall system.

1.2 Purpose

The main goal is to build an efficient and user-friendly freelance management platform that streamlines hiring, communication, and collaboration, using secure, scalable technologies.

2. IDEATION PHASE

2.1 Problem Statement

Freelancers and clients often struggle to find each other on a reliable, secure, and efficient platform. Communication issues, lack of transparency, and inefficient workflows create barriers to collaboration.

2.2 Empathy Map Canvas

Users: Freelancers, Clients, Admin

- **Think & Feel:** Seek efficiency, clarity, ease of use
- **See:** Clean UI, relevant project listings
- **Say & Do:** Communicate through chat, post/apply for jobs
- **Pain Points:** Delayed responses, security concerns, lack of platform reliability
- **Gain:** Smooth onboarding, real-time messaging, secure payments

2.3 Brainstorming

- Secure login system
- Real-time chat
- Project posting & application tracking
- Admin dashboard for monitoring
- User-friendly design with React components
- MongoDB for scalable storage

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. Client posts a project
2. Freelancer browses and applies
3. Both communicate via chat
4. Client selects a freelancer
5. Work submitted and reviewed
6. Admin supports dispute resolution

3.2 Solution Requirement

- User Authentication
- CRUD operations for users, projects, and applications
- Real-time messaging
- Admin monitoring and moderation tools

3.3 Data Flow Diagram

Client ↔ Server (Node.js + Express) ↔ MongoDB

Data flows via RESTful APIs, with CRUD operations for each module.

3.4 Technology Stack

- Frontend: React.js, Material UI, Bootstrap
 - Backend: Node.js, Express.js
 - Database: MongoDB with Mongoose
 - Other Libraries: Axios, bcrypt, cors
-

4. PROJECT DESIGN

4.1 Problem-Solution Fit

SB Works resolves the lack of centralized freelance platforms with real-time interaction, simplified application processes, and role-based access.

4.2 Proposed Solution

The platform provides:

- Project posting, application, and communication features
- Secure access for different users (freelancers, clients, admin)
- A modern and responsive UI

4.3 Solution Architecture

- React frontend communicates via Axios
- REST APIs managed by Express.js
- MongoDB stores users, projects, chats, and applications
- JWT for authentication
- Admin tools integrated in backend

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Milestone 1: Project Setup

- **Set up client/server directories**
- **Install necessary packages: React, Express, Mongoose, etc.**

Milestone 2: Backend Development

- **Configure MongoDB**
- **Create models: User, Freelancer, Project, Application, Chat**
- **Setup routes & authentication**

Milestone 3: Database Development

- **MongoDB setup (Atlas/local)**
- **Define collections**
- **Connect DB to server**

Milestone 4: Frontend Development

- **UI Components: Forms, Lists, Cards**
- **Navigation with React Router**
- **API integration using Axios**

Milestone 5: Implementation

- **Final testing**
- **Bug fixes**
- **UI optimization**

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- **Tested API response times**
- **Verified concurrent chat handling**
- **Validated MongoDB query efficiency**

- Frontend responsiveness ensured across screen sizes

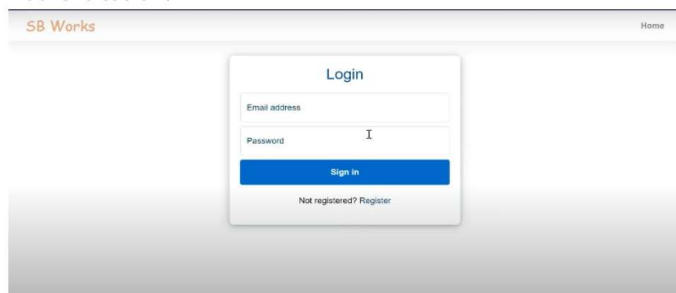
7. RESULTS

7.1 Output Screenshots

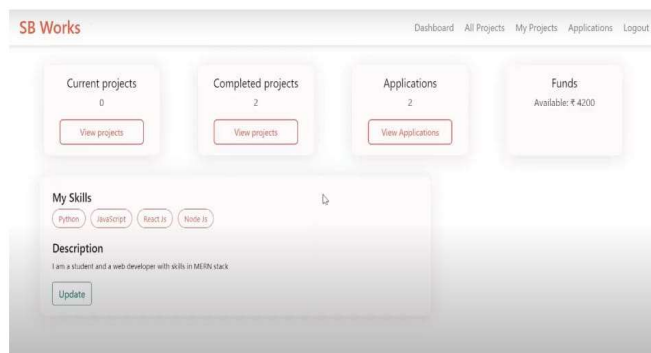
Landing page:



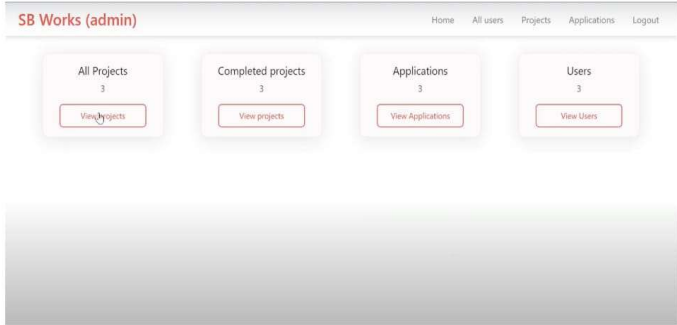
Authentication:



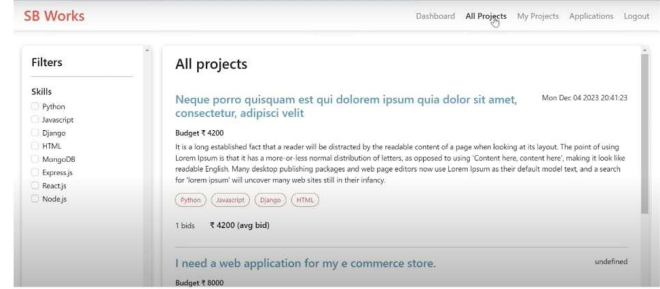
Freelancer dashboard:



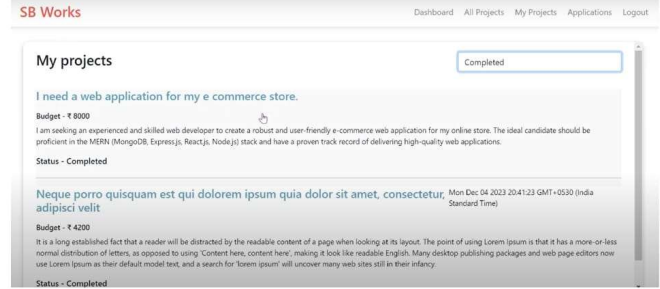
Admin dashboard:



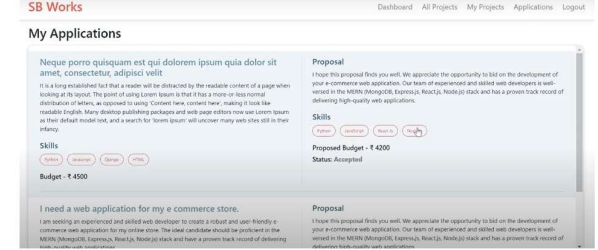
All projects:



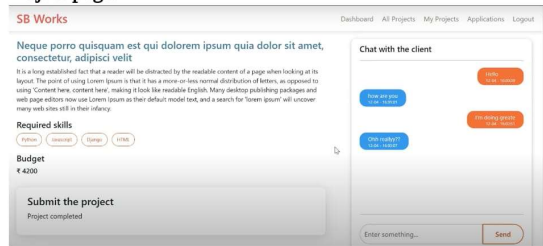
Freelance projects:



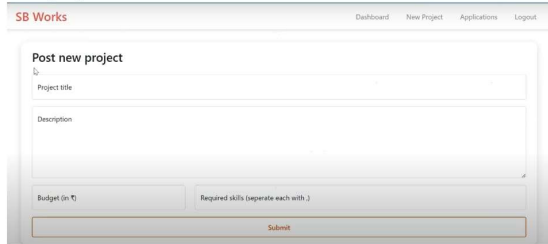
Applications:



Project page:



New project:



8. ADVANTAGES & DISADVANTAGES

Advantages

- Real-time communication
- Modern UI with intuitive layout
- Secure user authentication
- Role-based access for clients, freelancers, and admin

Disadvantages

- Limited offline functionality
- Dependency on internet and server uptime

9. CONCLUSION

The Freelancer Project Management System effectively facilitates collaboration between clients and freelancers. Backend components like Express.js and MongoDB allow fast, organized data handling. Frontend built with React ensures a responsive UI. The project is scalable and ready for real-world deployment.

10. FUTURE SCOPE

- Add payment gateway integration
 - Include project rating & review system
 - Build a mobile version
 - AI-based recommendation engine for project-freelancer matching
 - Notification and email service integration
-

11. APPENDIX

- GitHub Link: <https://github.com/Bharathkata/FreelanceFinder-Discovering-Opportunities-Unlocking-Potential>
- Video Demo: <https://drive.google.com/file/d/1hNkd-AEFYXQBHpqNgZnCOi7F-qtbQzeD/view?usp=sharing>