FREELANCING APPLICATION USING MERN STACK
SB Works

A Project Report

Submitted in partial fulfillment of the requirements for the
SmartInternz Virtual Internship Program

Submitted By
Shubham Revananath Salunke
Savitribai Phule Pune University (SPPU)

Project Details
Project Title: **Freelancing Application MERN**
Duration: 30 Days

Mentor Details
Mentor: Syed

Technology Stack
Frontend: React.js | Backend: Node.js, Express.js | Database: MongoDB

Submitted To
SmartInternz

Academic Year
2024 – 2025

Phase 1: Ideation Phase

1.1 Introduction

The ideation phase marks the initial stage of the SB Works project, where the fundamental concept of the freelancing platform was conceived. This phase focuses on identifying realworld challenges faced by freelancers and clients and proposing a digital solution that bridges the gap between skilled professionals and project owners. The idea was to build a reliable, secure, and user-friendly platform using modern web technologies.

1.2 Background of the Project

With the rapid growth of the gig economy, freelancing has become a preferred career choice for many professionals. However, existing platforms often involve complex interfaces, delayed communication, and trust issues between clients and freelancers. SB Works was ideated to overcome these limitations by providing a transparent system where users can easily collaborate, communicate, and complete projects efficiently.

1.3 Problem Identification

During the ideation process, several problems were identified:

1. Difficulty for freelancers to find suitable projects
2. Lack of transparency in client–freelancer communication
3. Inefficient project tracking and submission processes
4. Limited opportunities for beginners to build credibility

These challenges highlighted the need for a structured freelancing platform.

1.4 Proposed Solution

SB Works proposes a MERN stack-based freelancing application that allows clients to post projects and freelancers to bid on them. The platform integrates user authentication, project management, communication tools, and feedback mechanisms to ensure smooth collaboration.

Admin oversight ensures platform integrity and security.

1.5 Benefits of the Proposed System

1. Simplified project posting and bidding process
2. Secure authentication and role-based access
3. Real-time communication between users
4. Transparent review and feedback system
5. Scalable architecture for future enhancements

1.6 Scope of the Project

The scope of SB Works includes developing a full-stack web application that supports user registration, project management, freelancer bidding, work submission, and admin monitoring. The project focuses on core freelancing functionalities and can be extended with payment gateways and advanced analytics in the future.

Phase 2: Requirement Analysis

2.1 Introduction

The Requirement Analysis phase focuses on identifying, analyzing, and documenting the functional and non-functional requirements of the SB Works freelancing application. This phase ensures that the system is developed according to user expectations and business needs. Clear requirements help avoid ambiguities during development and provide a strong foundation for system design.

2.2 Stakeholder Identification

The primary stakeholders involved in the SB Works platform are:

1. **Clients**: Users who post projects and hire freelancers
2. **Freelancers**: Professionals who bid on projects and deliver work
3. **Admin**: Responsible for monitoring users and maintaining platform integrity
4. **System Developers**: Design and develop the application
5. **End Users**: Individuals who interact with the system daily

Each stakeholder has unique requirements that influence system functionality.

2.3 Functional Requirements

Functional requirements describe what the system should do.

2.3.1 User Authentication

1. Users must be able to register and log in securely
2. Role-based access for clients, freelancers, and admin
3. Secure password storage and session management

2.3.2 Client Module Requirements

1. Clients should be able to post new projects
2. View freelancer profiles and proposals

3. Select freelancers for projects
4. Communicate with freelancers through chat
5. Provide feedback and ratings after project completion

2.3.3 Freelancer Module Requirements

1. Freelancers should be able to browse available projects
2. Submit proposals and bids
3. Upload project deliverables
4. Track project status ● Receive feedback and ratings

2.3.4 Admin Module Requirements

1. Admin should be able to manage users
2. Monitor project activities
3. Handle disputes between clients and freelancers
4. Ensure platform security and reliability

2.4 Non-Functional Requirements

Non-functional requirements define system quality attributes.

2.4.1 Performance

1. The system should respond to user requests efficiently
2. Support multiple users simultaneously

2.4.2 Security

1. Secure authentication using JWT
2. Data encryption for sensitive information
3. Protection against unauthorized access

2.4.3 Scalability

1. The application should support future growth
2. MongoDB should handle increasing data volumes

2.4.4 Usability

1. User-friendly interface
2. Easy navigation for all user roles

2.4.5 Reliability

1. Minimal downtime
2. Consistent data availability

2.5 Hardware Requirements

1. System with minimum 4 GB RAM
2. Internet connectivity
3. Server capable of hosting Node.js applications

2.6 Software Requirements

1. Operating System: Windows / Linux
2. Frontend: React.js
3. Backend: Node.js, Express.js
4. Database: MongoDB
5. Tools: GitHub, Postman, VS Code

2.7 Assumptions and Constraints

1. Users have basic internet access
2. Payment integration is outside the current scope

3.1 Introduction

The Project Planning Phase defines the roadmap for the successful execution of the SB Works freelancing application. This phase focuses on organizing tasks, allocating resources, defining timelines, and identifying potential risks. Effective planning ensures that the project is completed within the specified duration and meets the required objectives.

3.2 Project Scope Definition

The scope of the SB Works project includes the design and development of a MERN stackbased freelancing platform. The system supports user authentication, project posting, bidding, communication, and project submission functionalities. Features such as payment gateway integration and advanced analytics are considered out of scope for the current implementation.

3.3 Project Timeline

The project is planned for a duration of **30 days**, divided into structured phases:

1. **Week 1**: Ideation and Requirement Analysis
2. **Week 2**: System Design and Database Modeling
3. **Week 3**: Backend and Frontend Development
4. **Week 4**: Testing, Documentation, and Deployment

This timeline ensures systematic progress and timely completion.

3.4 Resource Planning

The resources required for the project include:

1. Development tools such as VS Code and GitHub
2. MERN stack technologies
3. Cloud or local server for deployment
4. Documentation tools provided by SmartInternz

3.5 Role and Responsibility Allocation

1. **Frontend Developer**: Design and implement user interface using React.js
2. **Backend Developer**: Develop REST APIs and server logic using Node.js and Express.js
3. **Database Administrator**: Manage MongoDB schema and data storage
4. **Admin**: Monitor platform usage and user activities
5. **Mentor**: Provide guidance and review project progress

3.6 Risk Management

Potential risks identified during planning include:

1. Delays in development due to technical challenges
2. Integration issues between frontend and backend
3. Data security vulnerabilities
4. Time constraints

Mitigation strategies involve regular testing, mentor reviews, and version control using GitHub.

3.7 Milestones

Key project milestones include:

1. Completion of requirement analysis
2. Finalization of system design
3. Development of core modules
4. Successful testing and documentation
5. Final project submission

3.8 Quality Assurance Planning

Quality assurance is ensured through: ● Code reviews

1. Functional testing of modules
2. Performance testing

Phase 4: Project Design Phase

4.1 Introduction

The Project Design Phase focuses on transforming the requirements into a structured system design for the SB Works freelancing application. This phase defines the overall architecture, database structure, module interactions, and user interface design. A well-defined design ensures scalability, maintainability, and efficient system performance.

4.2 System Architecture Design

SB Works follows a **client-server architecture** where the frontend acts as the client and the backend acts as the server. The frontend is responsible for handling user interactions and displaying content, while the backend processes business logic, manages authentication, and interacts with the database.

Architectural Components:

1. **Client Layer**: React.js-based user interface
2. **Server Layer**: Node.js and Express.js REST APIs
3. **Database Layer**: MongoDB for data persistence
4. **Authentication Layer**: JWT-based authentication

This layered architecture ensures separation of concerns and secure data flow.

4.3 Data Flow Design

The data flow begins with user interactions on the frontend. Requests are sent to the backend using Axios in the form of RESTful API calls. The backend validates requests, processes data, and communicates with MongoDB. The response is then sent back to the frontend and rendered to the user.

This process ensures real-time updates and seamless user experience.

4.4 Database Design

The database is designed using MongoDB collections to store structured and unstructured data efficiently.

Key Collections:

1. **Users**: Stores client, freelancer, and admin details

2. **Projects**: Contains project postings by clients
3. **Proposals**: Stores freelancer bids and proposals
4. **Submissions**: Maintains uploaded project deliverables
5. **Reviews**: Stores feedback and ratings

Each collection is linked using unique identifiers to maintain data integrity.

4.5 Entity Relationship Overview

The relationship between entities is as follows:

1. One client can post multiple projects
2. One project can receive multiple proposals
3. One freelancer can submit proposals for multiple projects
4. Each project can have one final submission
5. Reviews are associated with completed projects

This structure supports efficient querying and data consistency.

4.6 Module Design

The application is divided into modular components:

User Authentication Module

Handles registration, login, and role-based access control.

Client Module

Allows clients to post projects, review proposals, hire freelancers, and provide feedback.

Freelancer Module

Enables freelancers to browse projects, submit bids, upload work, and track project status.

Admin Module

Ensures platform security, manages users, and resolves disputes.

Communication Module

Facilitates real-time messaging between clients and freelancers.

4.7 User Interface Design

The user interface is designed using React.js with Bootstrap and Material UI to ensure responsiveness and usability. The UI focuses on:

1. Simple navigation
2. Clear call-to-action buttons
3. Dashboard-based layouts
4. Mobile responsiveness

4.8 Security Design

Security is a critical aspect of the design:

1. JWT tokens for secure authentication
2. Role-based authorization
3. Input validation to prevent injection attacks
4. Secure API endpoints

4.9 Design Constraints

1. Dependency on third-party libraries
2. Limited scope for payment gateway integration
3. Browser compatibility considerations

4.10 Design Benefits

1. Scalable architecture
2. Modular design for easy maintenance
3. Enhanced security and data integrity

4. Improved user experience

Phase 5: Development & Implementation

Phase

5.1 Introduction

The Development and Implementation Phase focuses on converting the system design into a fully functional SB Works freelancing application. This phase involves coding the frontend and backend components, integrating APIs, implementing database operations, and ensuring secure user authentication. The MERN stack is used to achieve efficient and scalable development.

5.2 Development Environment Setup

The development environment was set up using the following tools:

1. **Visual Studio Code** for coding
2. **Node.js** runtime environment
3. **MongoDB** for database management
4. **GitHub** for version control
5. **Postman** for API testing

This setup ensured a smooth development workflow.

5.3 Frontend Development

The frontend was developed using **React.js**, focusing on component-based architecture.
Reusable components were created for forms, dashboards, and navigation.

Key Frontend Features:

1. User registration and login interfaces
2. Client and freelancer dashboards
3. Project listing and proposal submission pages
4. Integrated chat interface
5. Responsive design using Bootstrap and Material UI

5.4 Backend Development

The backend was implemented using **Node.js** and **Express.js**. RESTful APIs were developed to handle

various operations.

Backend Responsibilities:

1. User authentication and authorization
2. CRUD operations for projects and proposals
3. Handling file uploads for submissions
4. Managing reviews and ratings
5. Admin monitoring functionalities

JWT was used for secure authentication.

5.5 Database Implementation

MongoDB was used to store application data. Mongoose schemas were created for users, projects, proposals, and submissions. Indexing and validation ensured data consistency and performance.

5.6 API Integration

Frontend components interacted with backend APIs using Axios. Each user action triggered an API request, and the backend returned appropriate responses. Error handling mechanisms were implemented to manage failed requests.

5.7 Version Control and Collaboration

GitHub was used for version control, enabling:

1. Code backup
2. Collaboration among team members
3. Tracking of changes
4. Easy rollback in case of errors

5.8 Challenges Faced

1. Managing state across components
2. Handling asynchronous API calls

3. Ensuring secure authentication
4. Integrating frontend and backend seamlessly

These challenges were resolved through testing, debugging, and mentor guidance.

5.9 Implementation Outcome

At the end of this phase, the SB Works application was fully functional with core freelancing features implemented and ready for testing.

Phase 6: Performance Testing Phase

6.1 Introduction

The Performance Testing Phase focuses on evaluating the efficiency, reliability, and stability of the SB Works freelancing application. Testing ensures that the system performs well under normal and peak usage conditions. This phase validates whether the developed application meets the non-functional requirements defined during the requirement analysis phase.

6.2 Testing Objectives

The primary objectives of performance testing are:

1. To verify system responsiveness
2. To ensure smooth functionality under multiple user requests
3. To identify and fix performance bottlenecks
4. To validate system reliability and stability

6.3 Types of Testing Performed

6.3.1 Functional Testing

Functional testing was conducted to verify that each module performs as expected.

1. User registration and login
2. Project posting and bidding
3. Proposal submission and approval
4. Work submission and feedback
5. Admin monitoring functionalities

6.3.2 Integration Testing

Integration testing ensured proper communication between frontend and backend components.

1. API request and response validation
2. Database connectivity checks
3. Authentication flow verification

6.3.3 Performance Testing

Performance testing evaluated:

1. Page load time
2. API response time

3. Database query performance
4. Concurrent user handling capability

6.3.4 Security Testing

Security testing focused on:

1. Authentication validation
2. Role-based access control
3. Protection against unauthorized access
4. Secure data handling using JWT

6.4 Testing Tools Used

1. **Postman** for API testing
2. **Browser Developer Tools** for performance analysis
3. **MongoDB Compass** for database verification
4. **Manual Testing** for UI and usability validation

6.5 Test Environment

Testing was performed in a controlled environment with:

1. Stable internet connectivity
2. Local and cloud-based servers ● Multiple user roles simulated simultaneously

6.6 Observations and Results

1. The application responded efficiently to user actions
2. APIs handled multiple requests without significant delay

3. Authentication and authorization worked securely
4. No major performance issues were observed

6.7 Issues Identified and Resolutions

Minor issues such as UI alignment and delayed API responses were identified and resolved through code optimization and debugging.

6.8 Testing Outcome

The SB Works application successfully passed all testing scenarios and met performance expectations, making it ready for deployment and final evaluation.

Phase 7: Results & Conclusion Phase

7.1 Introduction

The Results and Conclusion Phase summarizes the overall outcome of the SB Works freelancing application. This phase evaluates the final system performance, highlights achieved objectives, and reflects on the learning experience gained during the project execution.

7.2 Results Obtained

The SB Works application was successfully developed and deployed as a fully functional MERN stack-based freelancing platform. The system supports all core functionalities including user authentication, project posting, bidding, work submission, and feedback mechanisms.

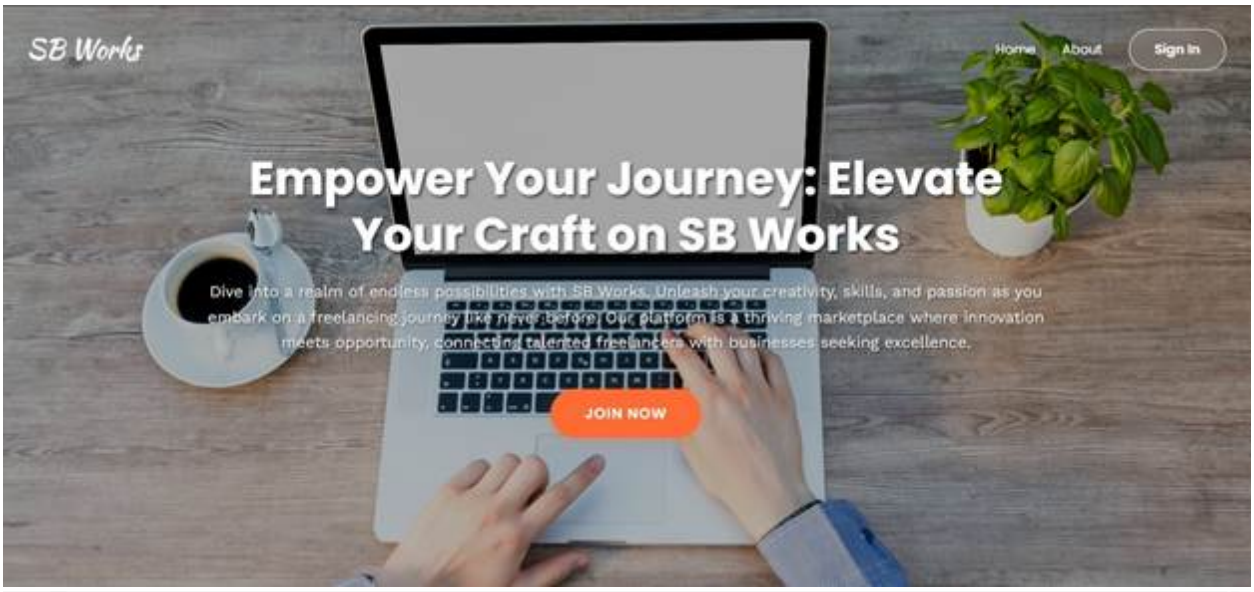
Key Results:

1. Smooth user registration and login functionality
2. Efficient project posting and proposal management
3. Seamless communication between clients and freelancers
4. Secure authentication and role-based access control
5. Admin monitoring to ensure platform integrity

The application performed reliably under normal usage conditions.

7.3 Output Screenshots

The following outputs were observed during execution:



Register

Username
ram

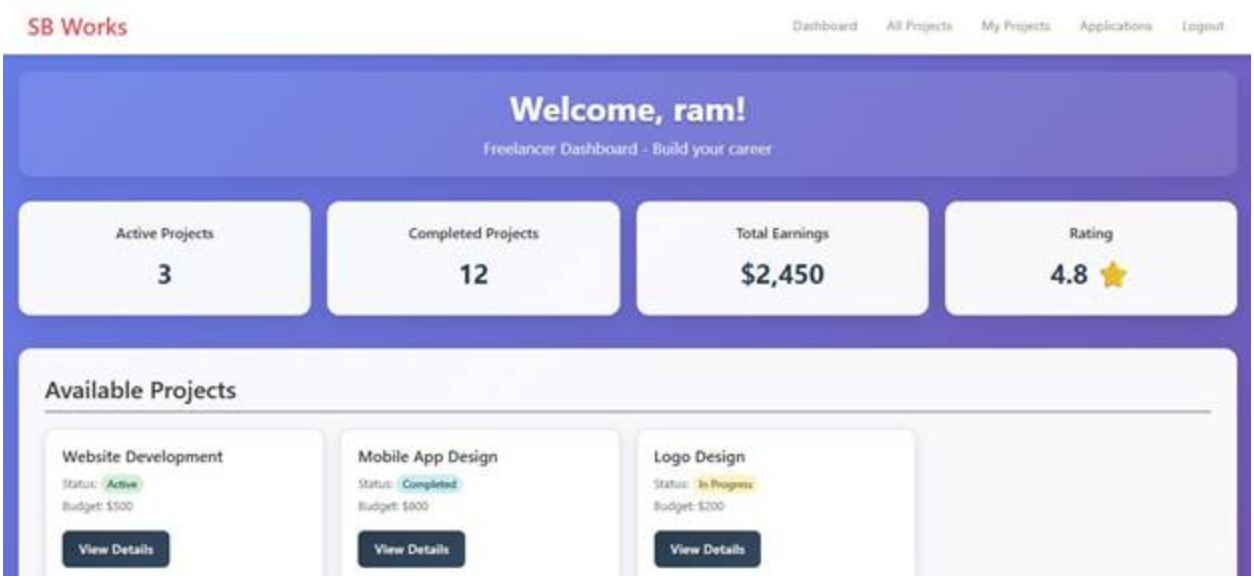
Email address
r@gmail.com

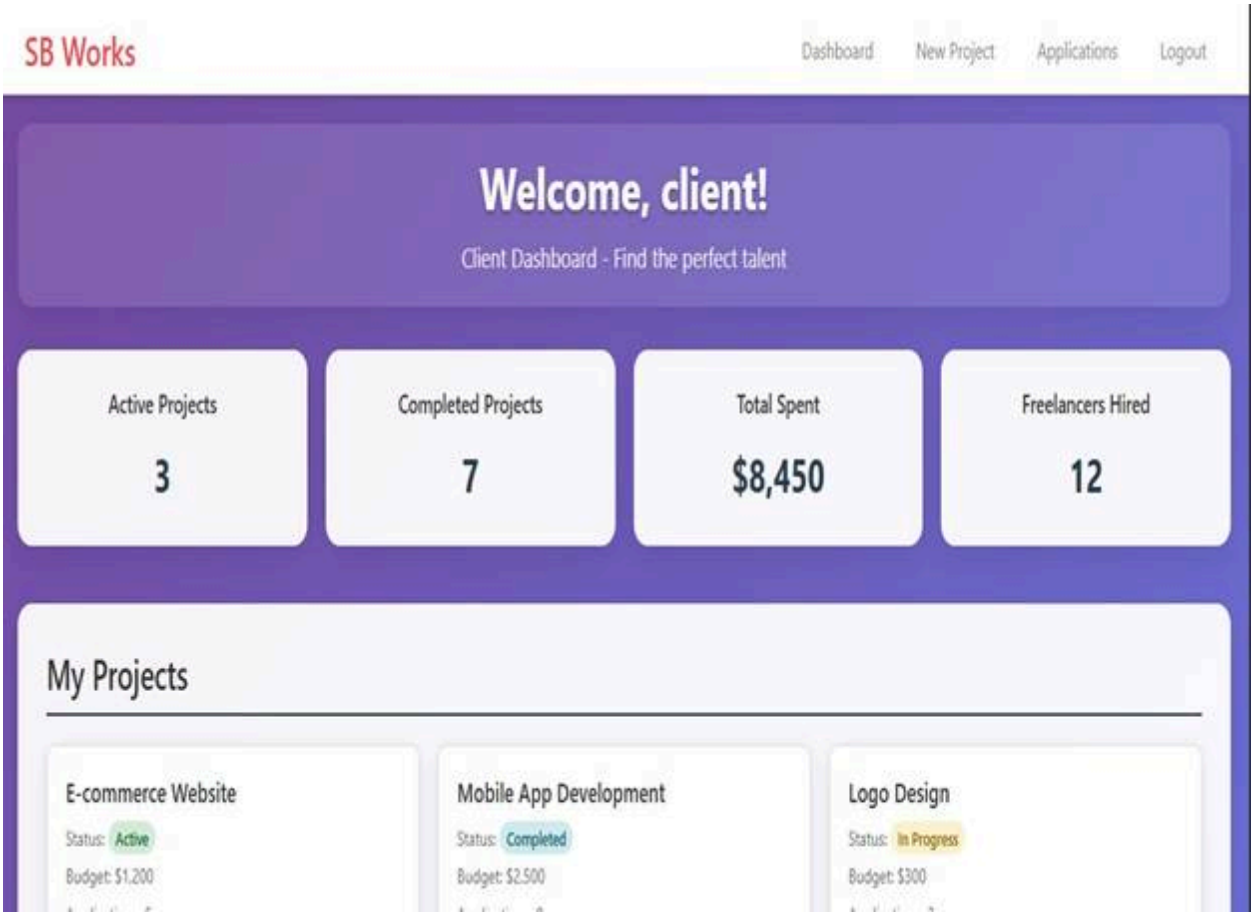
Password
•••

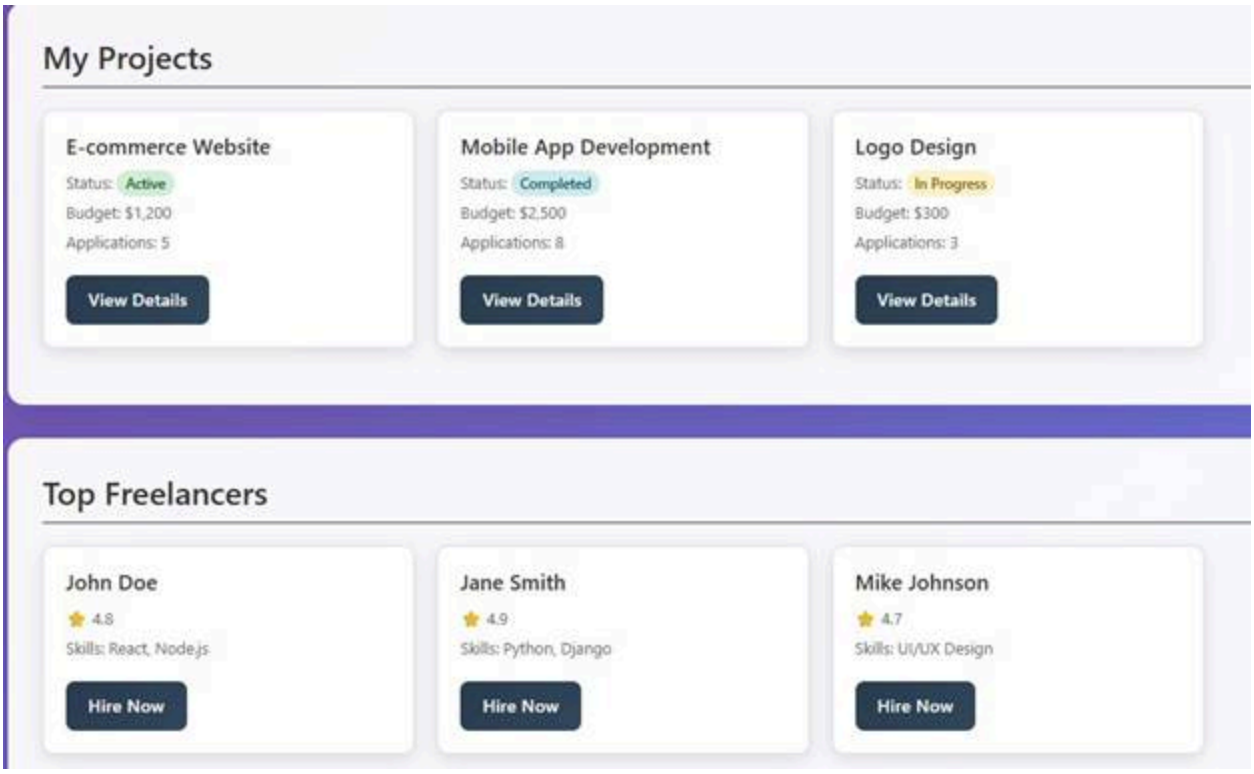
Freelancer

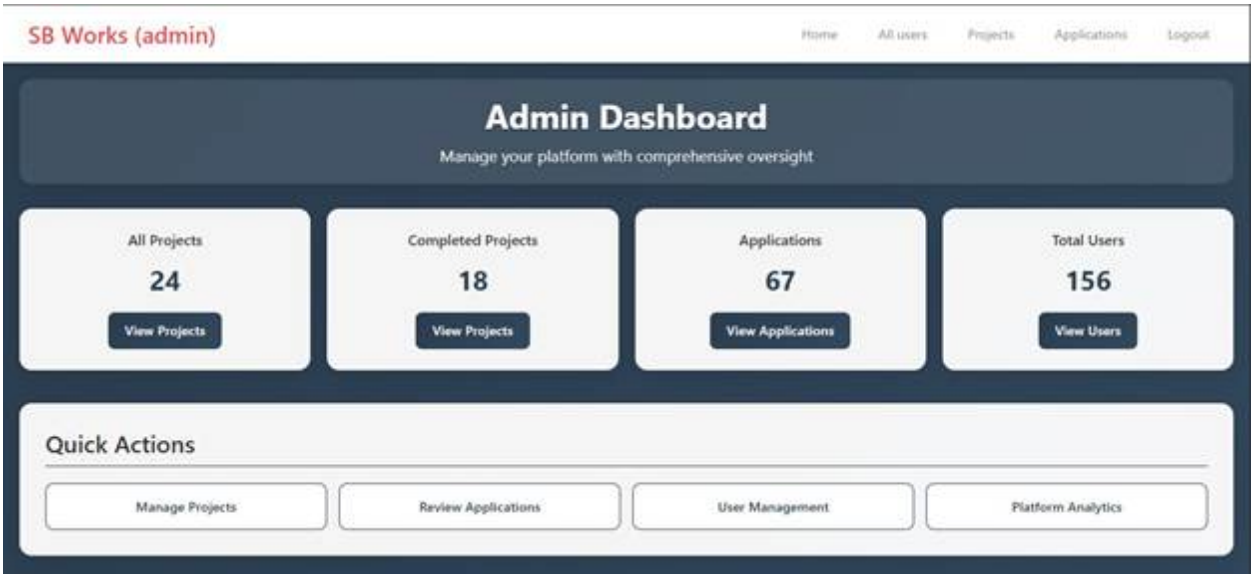
Sign up

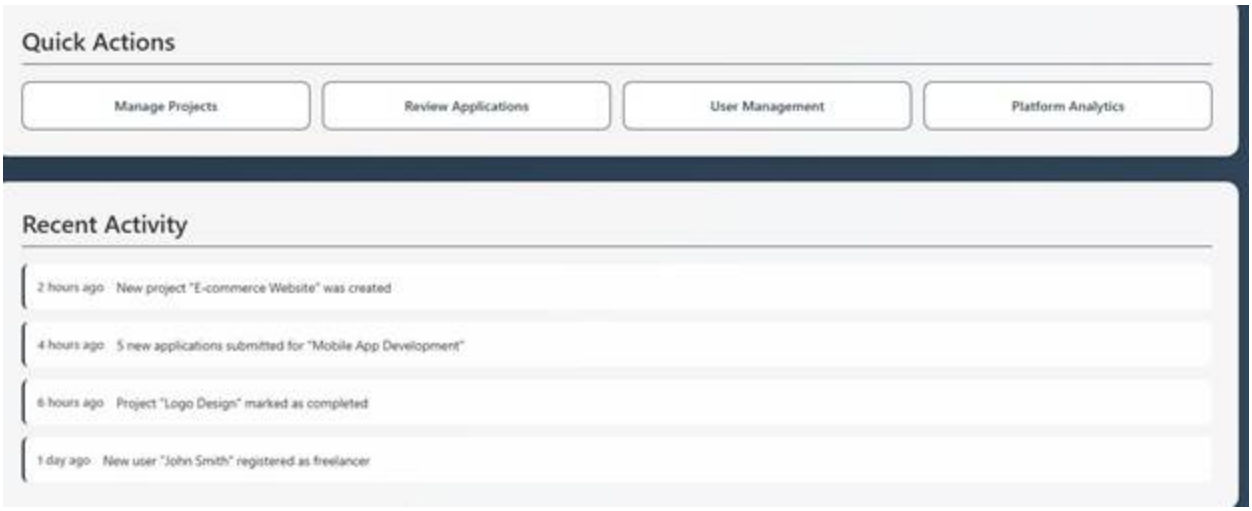
Already registered? Login


SB Works
Dashboard
All Projects
My Projects
Applications
Logout
Welcome, ram!
Freelancer Dashboard - Build your career
Active Projects
3
Completed Projects
12
Total Earnings
$2,450
Rating
4.8
Available Projects
Website Development
Status: Active
Budget: $500
View Details
Mobile App Design
Status: Completed
Budget: $800
View Details
Logo Design
Status: In Progress
Budget: $200
View Details


SB Works
Dashboard
New Project
Applications
Logout
Welcome, client!
Client Dashboard - Find the perfect talent
Active Projects
3
Completed Projects
7
Total Spent
$8,450
Freelancers Hired
12
My Projects
E-commerce Website
Status: Active
Budget: $1,200
Mobile App Development
Status: Completed
Budget: $2,500
Logo Design
Status: In Progress
Budget: $300


My Projects
E-commerce Website
Status: Active
Budget: $1,200
Applications: 5
View Details
Mobile App Development
Status: Completed
Budget: $2,500
Applications: 8
View Details
Logo Design
Status: In Progress
Budget: $300
Applications: 3
View Details
Top Freelancers
John Doe
4.8
Skills: React, Node.js
Hire Now
Jane Smith
4.9
Skills: Python, Django
Hire Now
Mike Johnson
4.7
Skills: UI/UX Design
Hire Now


SB Works (admin)
Home
All users
Projects
Applications
Logout
Admin Dashboard
Manage your platform with comprehensive oversight
All Projects
24
View Projects
Completed Projects
18
View Projects
Applications
67
View Applications
Total Users
156
View Users
Quick Actions
Manage Projects
Review Applications
User Management
Platform Analytics



7.4 Learning Outcomes

Through the development of SB Works, the following skills and knowledge were gained:

1. Full-stack development using the MERN stack
2. RESTful API development and integration
3. Database modeling with MongoDB
4. Secure authentication using JWT
5. Performance testing and debugging
6. Version control using GitHub

7.5 Limitations

1. Payment gateway integration is not implemented
2. Real-time chat can be enhanced using WebSockets
3. Advanced analytics and reporting are not included

7.6 Future Enhancements

1. Integration of online payment systems
2. Real-time notifications
3. AI-based freelancer recommendation system
4. Enhanced admin analytics dashboard

7.7 Conclusion

The SB Works Freelancing Application successfully fulfills its objectives by providing a secure, efficient, and user-friendly platform for freelancers and clients. The project demonstrates practical application of MERN stack technologies and follows a structured phase-wise development approach. This project has enhanced technical skills and provided valuable realworld development experience.