

Documentation for Freelancing

Application MERN

Table of contents

1. Introduction
2. System Requirements
3. Prerequisite
4. Architecture
5. ER Diagram
6. Project Structure
7. Application Flow
8. Project Setup and Configuration
9. Project Implementation and Execution
10. Conclusion

1.Introduction:

In this project, we develop a SB Works. It's an innovative freelancing platform designed to streamline the connection between clients and skilled freelancers across various industries. The platform offers an intuitive interface that enables clients to post projects, ranging from creative tasks to technical assignments, while freelancers can easily browse and bid on opportunities that match their expertise. With a focus on transparency, efficiency, and security, SB Works fosters a collaborative environment for both parties, ensuring smooth communication, reliable transactions, and high-quality results. This report explores the key features, technical architecture, and user experience of SB Works, highlighting how it is revolutionizing the freelancing landscape.

2.System Requirements:

To run the Freelance Application successfully, the following system requirements must be met

Hardware

Processor: 2 GHz or higher

RAM: 4 GB minimum (8 GB recommended)

Storage: 1 GB of free space for application and database storage

Software:

Operating System: Windows, macOS, or Linux

MongoDB: Version 4.x or higher

Node.js: Version 14.x or higher

npm: Version 6.x or higher

Web Browser: Latest version of Chrome, Firefox, or Safari

Development Tools:

IDE/Editor: Visual Studio Code or any preferred code editor, Postman for API testing

3.Prerequisite:

Before you start implementing the Freelance application, ensure the following prerequisites:

MongoDB Setup: Ensure that MongoDB is installed and running on your local machine or on a cloud service (e.g., MongoDB Atlas).

Node.js and npm: Install Node.js and npm (node package manager) for setting up the application server.

Version Control: Git should be installed for version control and collaboration.

Knowledge of JavaScript/Node.js: The application will use JavaScript with Node.js for the backend.

4.Architecture:

SB Works follows a client-server architecture designed for scalability, security, and efficiency. The platform is divided into two main components: the frontend and the backend, which communicate via RESTful APIs.

1.Frontend:

Built with React.js for dynamic, responsive user interfaces. Utilizes Bootstrap and Material UI for visually appealing and mobile-friendly design. Axios is used for API requests, ensuring smooth data flow between the client and server. Supports real-time features, such as messaging and notifications, through WebSockets.

2.Backend:

Developed with Express.js, providing a robust framework for handling HTTP requests and managing server-side logic. MongoDB is used as the database, offering a flexible and scalable solution for storing user data, projects, bids, and communications. Node.js powers the backend, ensuring fast, non-blocking operations.

3.Admin Layer:

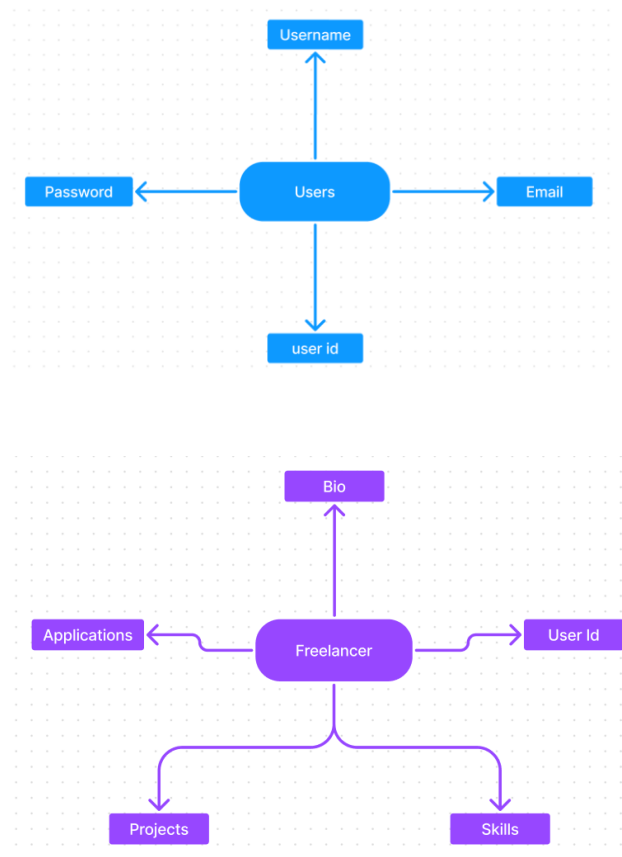
An admin interface oversees user authentication, content moderation, and transaction security, maintaining platform integrity. Admins manage project posting, freelancer selection, and financial transactions to ensure a safe and reliable environment.

4.Cloud Hosting and Deployment:

The platform is deployed on scalable cloud services like AWS or Google Cloud, ensuring high availability and performance. CI/CD pipelines are used for automated deployment and testing, ensuring efficient updates and version control.

This architecture ensures SB Works is secure, scalable, and able to handle real-time communication between clients and freelancers, providing an efficient freelancing experience.

5.ER Diagram:



SB Works connects clients with skilled freelancers through a user-friendly platform. Clients can post projects with details and browse freelancer profiles to find the perfect match. Freelancers can submit proposals, collaborate with clients through secure chat, and securely submit work for review and payment. An admin team ensures quality and communication, making SB Works a go-to platform for both clients and freelancers.

6.Project Structure:

SBWorks/

|

|— client/

Frontend (React)

	— public/	Public assets
	— index.html	Main HTML file
	— favicon.ico	Favicon
	— src/	Source code for frontend
	— assets/	Images, icons, and other static assets
	— components/	Reusable UI components
	— pages/	React components for different pages
	— services/	API service functions (Axios calls)
	— context/	Context API for state management (if using)
	— App.js	Main app component
	— index.js	Entry point for React app
	— .env	Environment variables for the frontend
	— package.json	Frontend dependencies & scripts
	— webpack.config.js	Webpack configuration for bundling
	— server/	Backend (Node.js/Express)
	— controllers/	Controllers for handling requests
	— authController.js	User authentication logic
	— projectController.js	Logic for project posting, bidding, etc.
	— userController.js	Logic for user management (clients & freelancers)
	— models/	MongoDB models (Schemas)
	— User.js	User model schema
	— Project.js	Project model schema
	— Bid.js	Bid model schema
	— Feedback.js	Feedback model schema

— routes/	API routes
— authRoutes.js	Authentication routes
— projectRoutes.js	Routes for managing projects
— userRoutes.js	Routes for managing users
— bidRoutes.js	Routes for bids and freelancer selection
— utils/	Utility functions and helpers
— authMiddleware.js	Middleware for user authentication
— uploadMiddleware.js	File upload handling (e.g., portfolio images)
— validation.js	Input validation logic
— config/	Configuration files
— db.js	MongoDB connection configuration
— server.js	Express server configuration
— .env	Environment variables for the backend
— .env	Backend environment variables
— package.json	Backend dependencies & scripts
— server.js	Main server entry point
— shared/	Shared code or libraries
— emailService.js	Email sending functionality (e.g., notifications)
— notificationService.js	Real-time messaging and notifications
— .gitignore	Files/folders to ignore in Git
— README.md	Project documentation
— package.json	Root level dependencies & scripts

8.Project Setup and Configuration:

1.Folder setup:

Now, firstly create the folders for frontend and backend to write the respective code and install the essential libraries.

- Client folders.
- Server folders

2.Installation of required tools:

1. Open the frontend folder to install necessary tools

For frontend, we use:

- React
- Bootstrap
- Material UI
- Axios
- react-bootstrap

2. Open the backend folder to install necessary tools

For backend, we use:

- Express Js
- Node JS
- MongoDB
- Mongoose
- Cors
- Bcrypt

3.Clone the Repository:

git clone <https://github.com/Hariragavan00/Freelancing-Website/tree/master>

4. Navigate into the cloned repository directory:

```
cd Freelance-Website
```

5.Install the required dependencies:

```
cd client
npm install
cd ..
cd server
npm instal
```

6.Start the Development Server:

- To start the development server, execute the following command:
npm start
- The SB Works app will be accessible at <http://localhost:3000>

9.Project Implementation and Execution:

The SB Works platform was developed using a full-stack approach with React for the frontend, Node.js/Express for the backend, and MongoDB for database management. Key features include secure user authentication using JWT, real-time messaging with Socket.io, and an intuitive user interface for both clients and freelancers. The backend API supports CRUD operations for projects, bids, and user profiles, while the frontend communicates with the API via Axios. Real-time communication between clients and freelancers is facilitated by WebSocket for instant updates on project discussions and bids. The platform was deployed using Docker for containerization and CI/CD pipelines for automated testing and deployment to cloud services like AWS and Netlify. The system is built to scale efficiently, ensuring smooth performance as the user base grows.

10.Conclusion:

The SB Works platform successfully connects freelancers and clients through an intuitive, secure, and scalable system. By leveraging modern technologies like React, Node.js, Express, and MongoDB, the platform provides seamless project management, real-time communication, and secure transactions. The implementation prioritizes user experience and system performance, while cloud deployment and CI/CD integration ensure efficient scaling. Overall, SB Works offers a robust solution for the freelancing ecosystem, enhancing collaboration and productivity for both clients and freelancers.