Full Stack Development with MERN

Project Documentation format

1. Introduction

- **Project Title:** Freelancing Application MERN
- **Team Members:** Lisha Reddy Alluri

2. Project Overview

- **Purpose:** The purpose of this project is to build a full-stack freelancing platform called SB Works that connects clients with skilled freelancers. The platform enables clients to post projects and freelancers to bid, work, and submit deliverables in a streamlined, secure, and collaborative environment.
- **Features:**
- User roles: Client, Freelancer, and Admin
- Project posting and bidding system
- User authentication and authorization
- Profile management and feedback system
- Real-time messaging and notifications
- Admin dashboard for monitoring and management

3. Architecture

- **Frontend:**
- ➢ Built using React.js
- ➢ Component-based architecture
- ➢ Uses React Router for navigation
- ➢ API communication handled using Axios
- ➢ State managed using useState and useEffect

- **Backend: (Node.js + Express.js)**
- ➢ Built on **Express.js** framework
- ➢ RESTful API architecture
- ➢ Handles routing, authentication, and business logic
- ➢ Uses **JWT** for secure user sessions

- **Database:**
- ➢ Uses MongoDB Atlas for cloud-hosted NoSQL database
- ➢ Data models include User, Project, Bid
- ➢ Mongoose is used for schema definitions and database interactions

4. Setup Instructions

- **Prerequisites**: Node.js , MongoDB Atlas account or local MongoDB , Git , VS code.
- **Installation:** Step-by-step :

1. Clone the repository:

git clone https://github.com/yourusername/freelancing-application-mern-sbworks.git
cd freelancing-application-mern-sbworks

2. Install server dependencies:

```
cd server
npm install
```

3. Create .env file inside the server folder:

```
PORT=5000
MONGO_URI=your_mongo_connection_string
JWT_SECRET=your_jwt_secret
```

4. Install client dependencies:

```
cd ../client
npm install
```

5. Folder Structure

- **Client:**
- ├── public/
- ├── src/
- │ ├── components/
- │ ├── pages/
- │ ├── services/
- │ ├── App.js
- │ └── index.js
- **Server:**
- ├── controllers/
- ├── models/
- ├── routes/
- ├── middleware/
- ├── server.js
- └── .env

6. Running the Application

- Provide commands to start the frontend and backend servers locally.
  - **Frontend:** cd client
    npm start
  - **Backend:** cd server
  - npm run dev

7. API Documentation

Endpoint	Method	Description
/api/auth/register	POST	Register a new user
/api/auth/login	POST	Login and return JWT
/api/projects	POST	Create new project (Client only)
/api/projects	GET	Get all projects
/api/bids/:projectId	POST	Place bid (Freelancer)

8. Authentication

- ➢ Implemented using JWT (JSON Web Tokens)
- ➢ Tokens are issued on login and stored in local storage
- ➢ Middleware protects routes to ensure role-based access (Client, Freelancer, Admin)

9. User Interface :

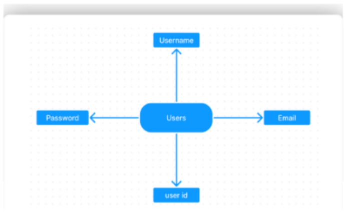
The user interface is built using React.js with a clean and intuitive design that supports multiple user roles: Client, Freelancer, and Admin. The application uses a responsive layout with a role-based navigation bar that dynamically updates based on the logged-in user's role.

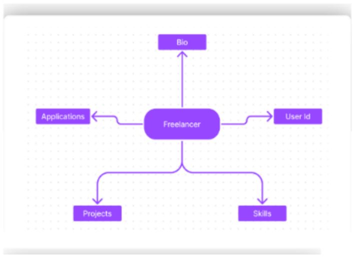
Clients can log in and access a dashboard where they can post new projects through a guided form, view submitted bids, and manage ongoing collaborations. Freelancers have access to a personalized dashboard that displays available projects, allows them to place bids, and track the status of their work. Admins can monitor all user activities, approve or block users, and ensure the overall integrity of the platform.

Core UI components include login and registration forms, dynamic dashboards, project cards, bid submission forms, feedback modals, and notification toasts. Each page is optimized for user experience with clear labels, action buttons, and smooth navigation.

Key pages:

- Login/Register Page – for secure sign-in and account creation
- Client Dashboard – project management and freelancer communication
- Freelancer Dashboard – list of available projects and bid history
- Project Bidding Form – form for freelancers to place project bids
- Admin Panel – tools for user monitoring and platform control

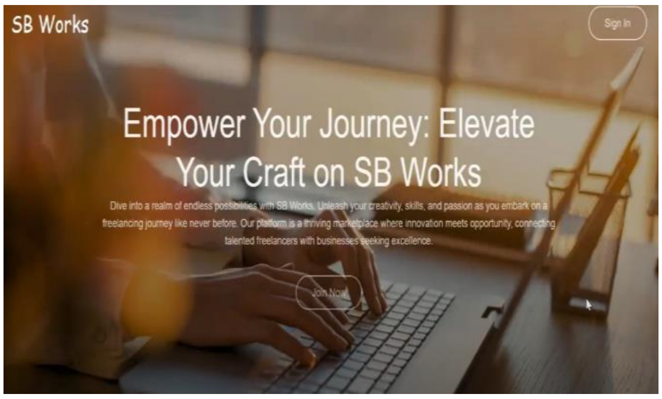




10. Testing

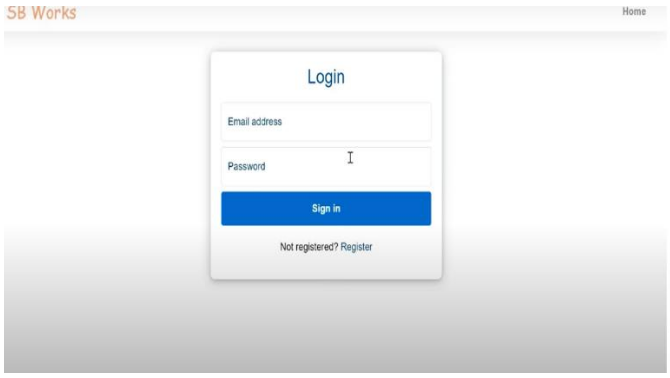
- ➢ Manual testing via Postman for API
- ➢ Basic frontend testing using React Developer Tools
- ➢ Future enhancements may include unit tests using **Jest** and **React Testing Library**

11. Screenshots or Demo

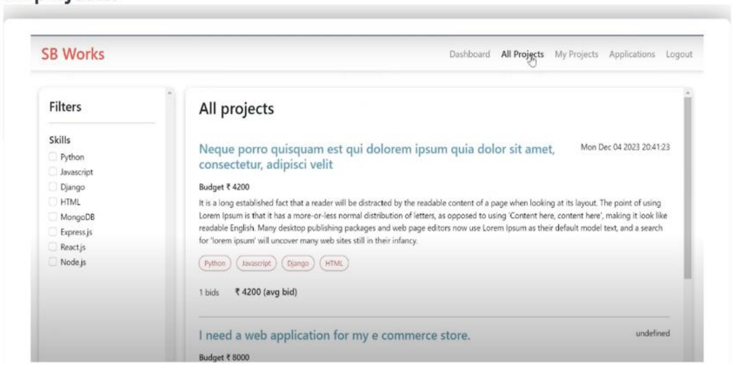


Empower Your Journey: Elevate Your Craft on SB Works

Dive into a realm of endless possibilities with SB Works. Unleash your creativity, skills, and passion as you embark on a freelancing journey like never before. Our platform is a thriving marketplace where innovation meets opportunity, connecting talented freelancers with businesses seeking excellence.



All projects:



All projects

Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit

Mon Dec 04 2023 20:41:23

Budget ₹ 4200

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy.

Python Javascript Django HTML

1 bids ₹ 4200 (avg bid)

I need a web application for my e commerce store.

undefined

Budget ₹ 8000

12. Known Issues

- ➢ UI may need optimization for mobile devices
- ➢ Bid validation logic needs stricter checks
- ➢ Notifications system may have occasional delays

13. Future Enhancements

- Payment integration (e.g., Razorpay, Stripe)
- Chat system using Socket.IO
- Rating and review system
- Admin analytics dashboard
- Email notifications for updates