

Full Stack Development with MERN Project

Documentation

1. Introduction

Project Title : FreelanceFinder-Discovering-Opportunities-Unlocking-Potential
Team ID : LTVIP2025TMID54850
Team Members :

Ungata Kalyani (Team Leader)-Backend Developer

Teku Brenham (Member 2)-Database & Testing

Tirumalasetti Thanmayee (Member 3)- Frontend Developer

Toleti Yasasvi Sai Sree (Member 4)-Full Stack Developer

2. Project Overview

Purpose: FreelanceFinder is a MERN-based freelancing platform where clients can post projects and freelancers can bid on them. It simplifies remote work hiring by connecting talented freelancers with project owners through a seamless and secure platform.

Features:

- User registration and JWT-based authentication (Client/Freelancer).
- Role-based dashboard views and access.
- Profile management with portfolio for freelancers.
- Real-time bid tracking and notifications.
- Admin panel for user/project management.

3. Architecture

Frontend(React.js):

- Built with React and React Router for navigation.
- Axios used for API calls.
- Role-based routing and component rendering.
- Axios is used for making HTTP requests to the backend APIs.
- Uses Bootstrap/Tailwind CSS for a responsive, clean interface.

Backend(Node.js+Express.js):

- RESTful API architecture.
- Clients can bookmark freelancers they like.
- Controller-service-repository structure for scalability.

- JWT for authentication and protected routes.
- Freelancers can upload proposals or sample work while bidding.
- Bcrypt.js for password encryption.

Database(MongoDB):

- Users, Projects, and Bids stored in MongoDB.
- Mongoose used for schema modeling and validation

Relationships:

- Users (Clients or Freelancers).
- Projects linked to Client.
- Bids linked to Projects and Freelancers

4. Setup Instructions

Prerequisites:

- Node.js (v18+ recommended)
- MongoDB (local or Atlas)
- Git

Installation:

```
# Clone the repository
git clone https://github.com/yourusername/booknest.git
cd booknest

# Install server dependencies
cd server
npm install

# Install client dependencies
cd ../client
npm install

# Set up environment variables
# In server/.env:
PORT=5000

MONGO_URI=mongodb+srv://freelance_user:MySecurePass123@cluster0
.abcd1.mongodb.net/freelancefinder?retryWrites=true&w=majority

JWT_SECRET=myVeryStrongSecretKey123! @#456
```

5. Folder Structure:

Client:

```
client/
├── public/
│   └── index.html
├── src/
│   ├── components/    # Navbar, Footer, FreelancerCard, etc.
│   ├── pages/         # Home, Login, Register, Dashboard
│   ├── App.js
│   ├── index.js
│   └── api/           # Axios service modules
```

Server:

```
server/
├── controllers/    # User, Project, Bid controllers
├── models/         # Mongoose models
├── routes/         # Auth, Project, Bid, User routes
├── middleware/     # JWT auth middleware
├── server.js       # Server entry point
└── config/         # MongoDB connection setup
```

6. Running the Application

- Provide commands to start the frontend and backend servers locally.

Frontend: `cd client`

`npm start`

Backend: `cd server`

`npm start`

7. API Documentation

Base URL: `http://localhost:5000/api`

Authentication:

- POST /api/auth/register:Register a new user (Client or Freelancer).
- POST /api/auth/login:Log in a registered user and return a JWT token.

Users:

- GET /api/users/:id:Fetch the profile details of a user by their ID.
- PUT /api/users/:id:Update a user's profile information.

Projects:

- POST /api/projects:Post a new freelance project (Client only).
- GET /api/projects:Retrieve all posted projects (for freelancers to browse and bid on).
- GET /api/projects/:id:Get full details of a specific project by its ID.

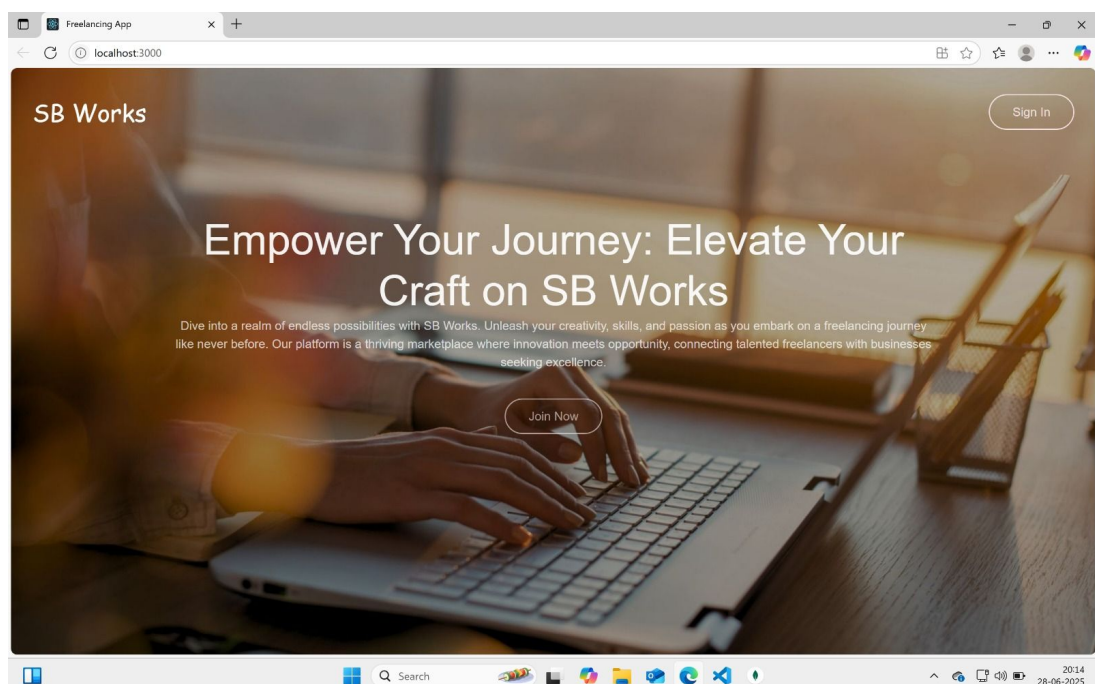
Bids:

- POST /api/bids:Place a new bid on a project (Freelancer only).
- GET /api/bids/project/:projectId:Get all bids placed on a specific project.
- GET /api/bids/user/:userId:Retrieve all bids placed by a specific freelancer.

8. Authentication

- JWT token generated on login, stored in localStorage.
- Backend verifies JWT on protected routes.
- Passwords are hashed with bcryptjs before storage in MongoDB.
- Middleware checks role and access level.
- Passwords hashed with bcryptjs.

9. User Interface



localhost:3000/auth/register

SB Works

Home

Register

Username
s/90

Email address
sree@gmail.com

Password

Freelancer

Sign up

Already registered? Login

Freelancing App

localhost:3000/freelancer

SB Works

Dashboard All Projects My Projects Applications Logout

localhost:3000 says
User data updated

Current projects
0
View projects

Completed projects
0
View projects

Applications
0
View Applications

Funds
Available: ₹ 0

My Skills

java ,Angularjs,Nodejs,Html,Css

Description

I am web developer

Update

Freelancing App

localhost:3000/auth/register

SB Works

Home

Register

Username
lucky

Email address
lucky1@gmail.com

Password

Client

Sign up

Already registered? Login

Freelancing App

localhost:3000/new-project

SB Works

Dashboard New Project Applications Logout

localhost:3000 says
new project added!

Post new project

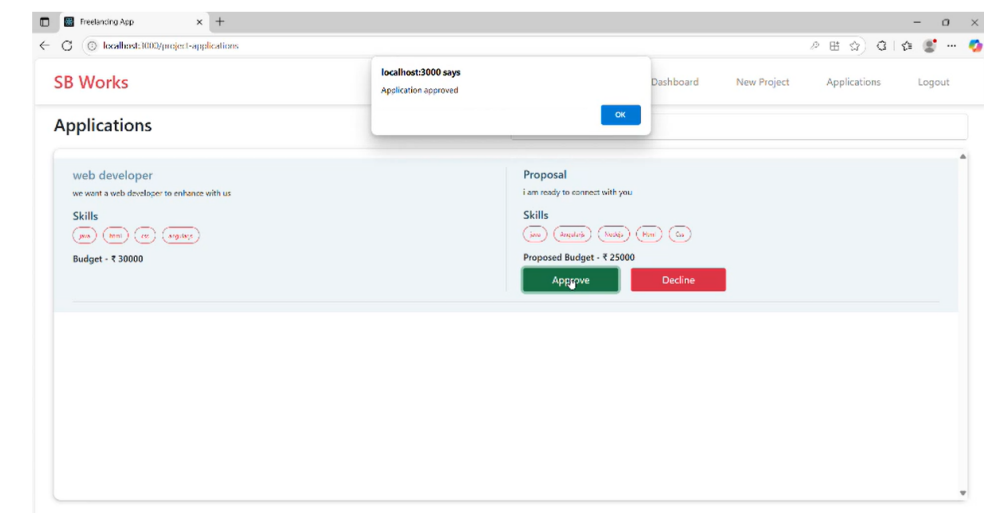
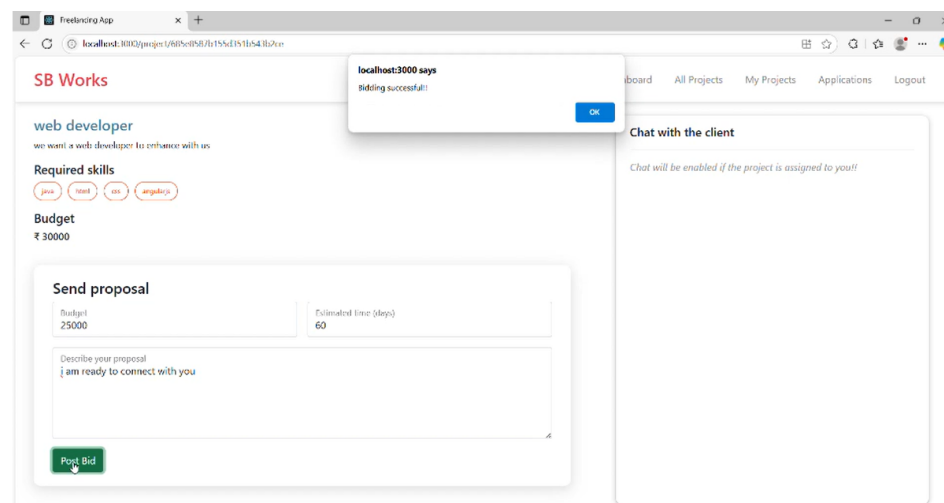
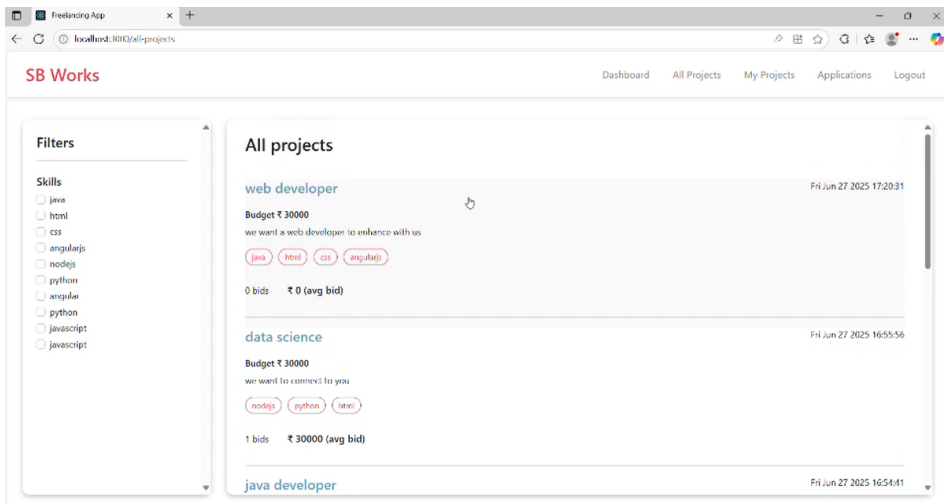
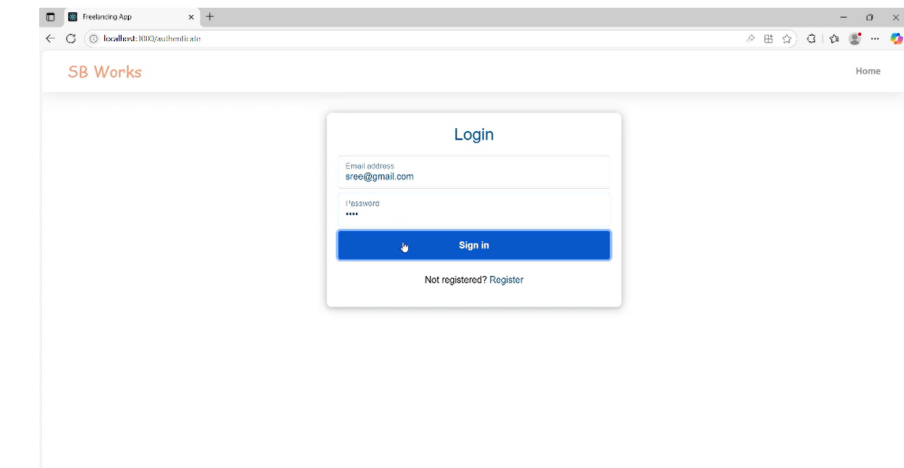
Project title
web developer

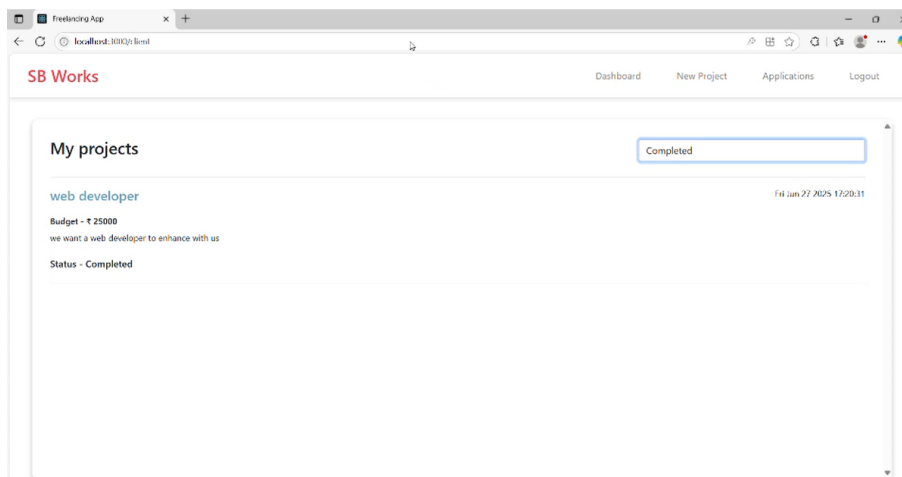
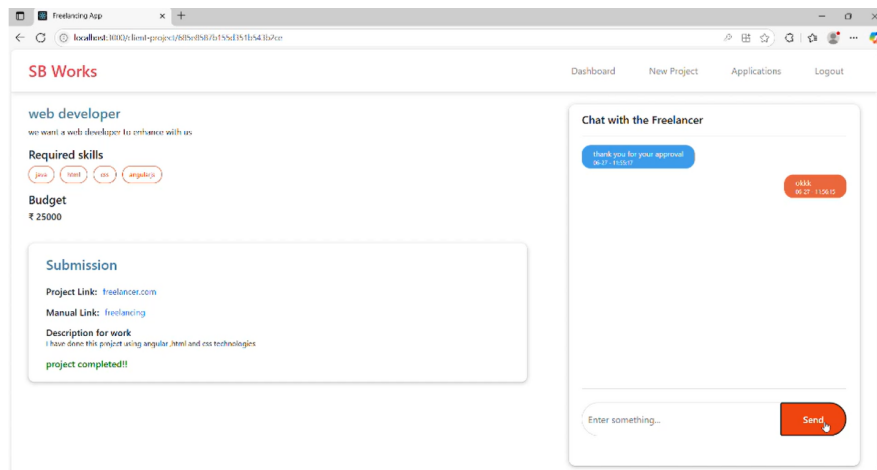
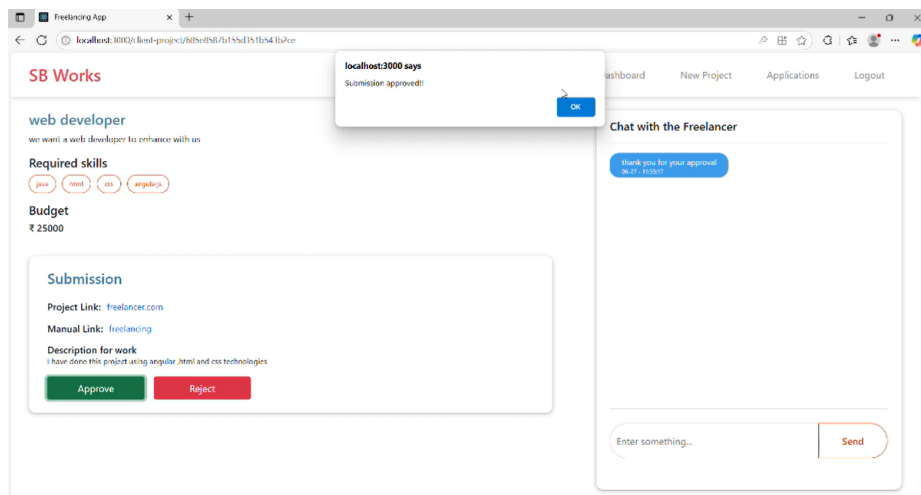
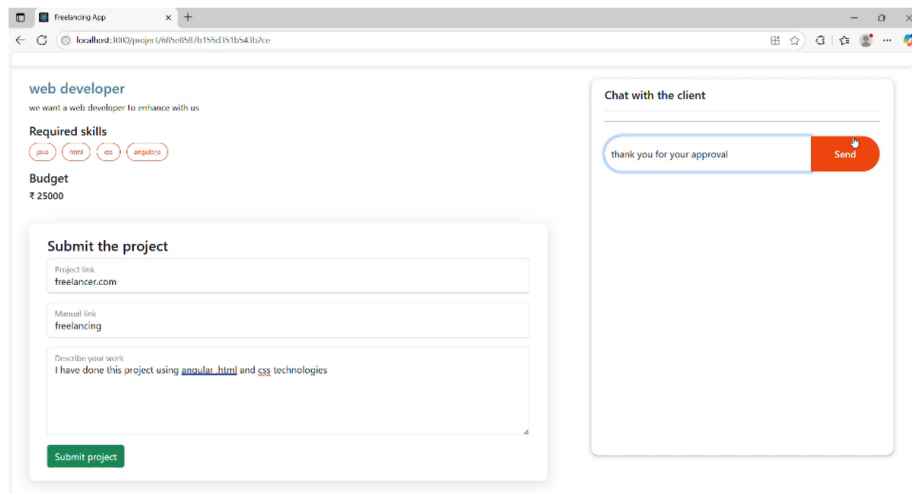
Description
we want a web developer to enhance with us

Budget (in ₹)
30000

Required skills (separate each with ,)
java,html,css,angularjs

Submit





10. Testing

- Postman for API testing.
- Manual testing of registration, project posting, bidding.
- Optional: Jest & React Testing Library.

11.Demo link

[click me](#)

12.Known Issues

- No payment gateway yet.
- Limited error handling.
- No email/SMS notifications.
- Admin dashboard under construction.

13. Future Enhancements

- Payment integration (Stripe/Razorpay).
- Ratings and reviews.
- Email notifications.
- Real-time chat system.
- Admin analytics.
- CI/CD pipeline.