Full Stack Development with MERN - Project Documentation

1. Introduction

Project Title: Freelance Finder

Team Members:

- Muzeeb Ali Khan
- Patheti Balaji
- Yeripilli Anand
- Bollareddy Pranoy Raj **Team ID:** LTVIP2025TMID43202

---

2. Project Overview

Purpose: Freelance Finder is an end-to-end platform designed to connect freelancers with potential clients across various domains. The platform bridges the gap between service providers and seekers by offering seamless onboarding, search, booking, and collaboration features. It empowers freelancers to showcase their portfolios and enables users to hire verified talent. The aim is to simplify project outsourcing while promoting skill-based work culture.

Features:

- User authentication for freelancers and clients
- Profile and portfolio management for freelancers
- Project posting and bidding system
- Search and filter functionalities by skill, location, and rating
- Admin panel to manage users and listings
- Real-time chat system for project discussions

---

3. Architecture

Frontend: Developed using React.js with component-based architecture, React Router for navigation, and responsive design via CSS modules.

Backend: Built using Node.js and Express.js. Provides RESTful API endpoints and uses controllers for business logic separation.

Database: MongoDB with Mongoose for schema definitions and object modeling. Includes collections for Users, Projects, Bids, and Messages.

---

4. Setup Instructions

Prerequisites:

- Node.js >= 14.x
- MongoDB (Atlas or Local)
- Git

Installation:

1. Clone the repository:

```
git clone [your-repo-url]
```

1. Navigate to project root:

```
cd freelance-finder
```

1. Install dependencies:

```
npm install
```

1. Create `.env` file with required keys.

Environment Variables:

```
MONGO_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret
PORT=5000
```

---

5. Folder Structure

Client:

```
client/
├── src/
│   ├── components/       # Reusable UI components
│   ├── pages/            # Main pages (Home, Login, Dashboard)
│   └── App.js            # Routing and global layout Server:
```

```
server/
├── models/              # Mongoose schemas
├── routes/              # API route definitions
├── controllers/         # Logic for handling routes
├── middleware/          # Auth and error middleware
└── server.js            # App entry point
```

---

6. Running the Application

Frontend:

```
cd client
npm start
```

Backend:

```
cd server
npm start
```

---

7. API Documentation

- `POST /api/register` – Register a new freelancer or client
- `POST /api/login` – Login and get JWT token
- `GET /api/freelancers` – Fetch list of freelancers
- `POST /api/projects` – Post a new freelance project
- `POST /api/bid` – Submit bid for a project
- `GET /api/projects/:id` – Get project details by ID

8. Authentication

JWT-based authentication system:

- Tokens issued on login
- Middleware validates token before accessing protected routes
- Role-based access control (Freelancer / Client / Admin)

---

9. User Interface

Screenshots are not included here but the UI supports:

- Modern, responsive layouts
- Dashboards for both freelancers and clients
- Form-based interactions with validations

---

10. Testing

- Manual testing with Postman for API routes
- Frontend tested with browser dev tools
- Future plan: Add Jest and Cypress for automated tests

---

11. Demo Link

TheDemovideo

12. Known Issues

- Notification system is not implemented
- Limited mobile optimization on certain screens
- Profile editing is missing image crop feature

---

13. Future Enhancements

- In-app payment gateway integration
- Notification alerts via email/SMS
- Multi-language support
- Advanced search filters
- Freelancer rating and feedback system

---

14. Sample Code: Project Posting Endpoint

```
// POST /api/projects
router.post('/projects', async (req, res) => {
  const { userId, title, description, budget } = req.body;
  try {
    const project = new Project({ userId, title, description, budget });
    await project.save();
    res.status(201).send('Project posted successfully');
  } catch (err) {
    res.status(500).send('Error posting project');
  }
});
```

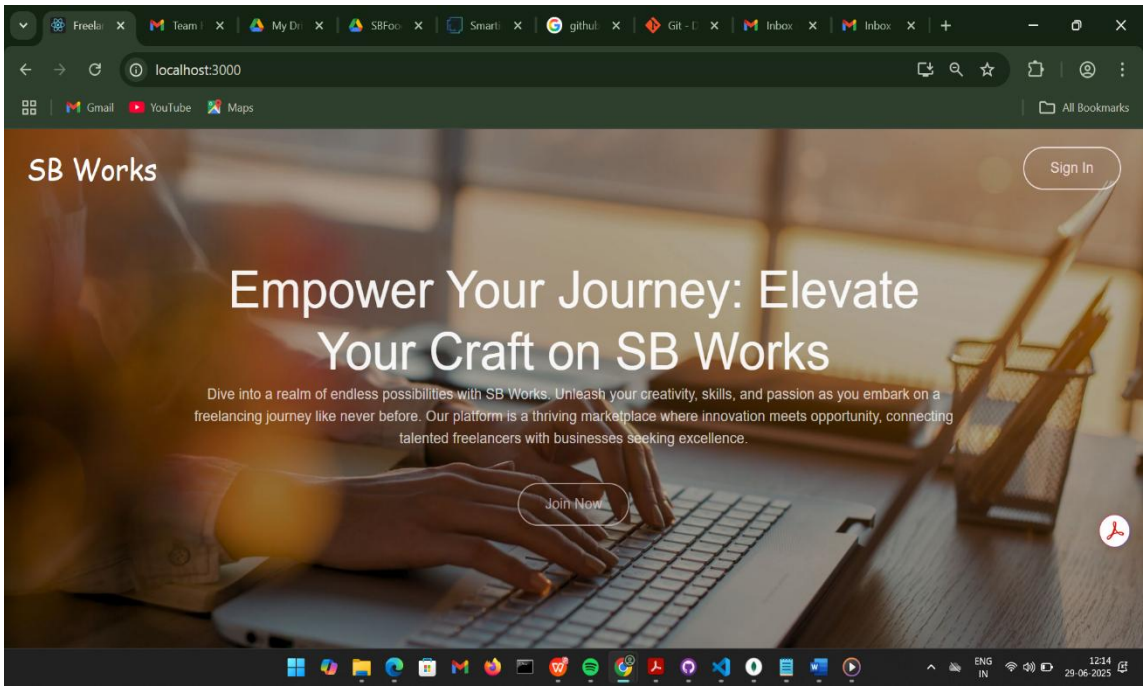
---

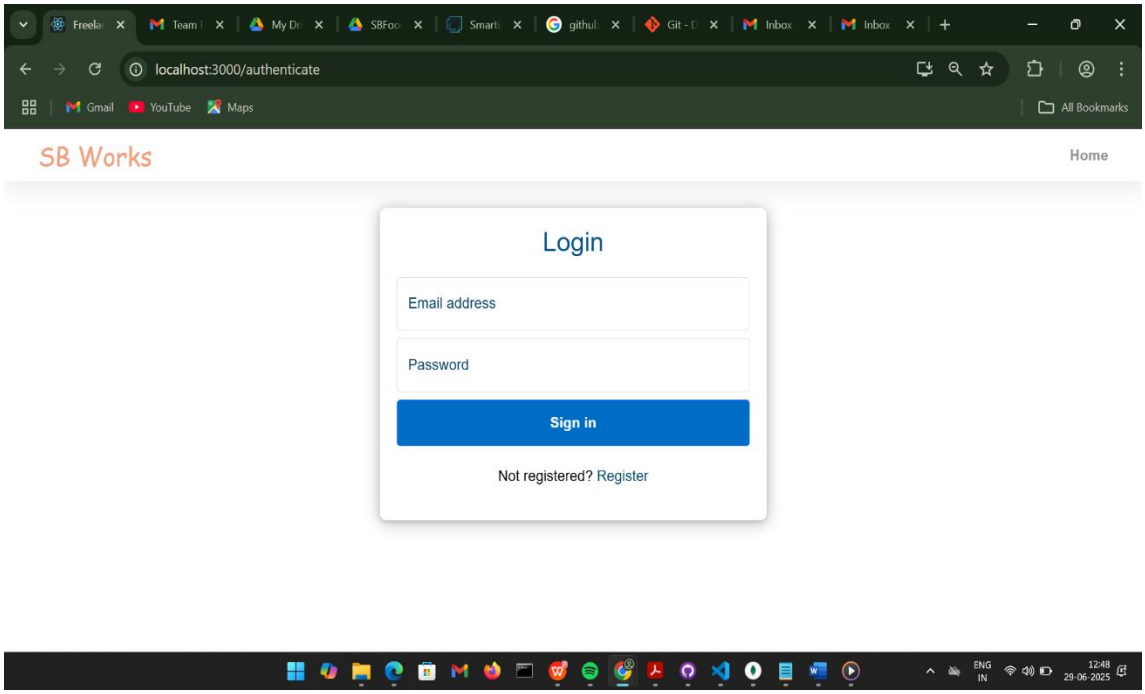
15. Sample Code: Mongoose Project Schema

```
const mongoose = require('mongoose');

const projectSchema = new mongoose.Schema({
  userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
  title: String,
  description: String,
  budget: Number,
  createdAt: { type: Date, default: Date.now }
});

module.exports = mongoose.model('Project', projectSchema);
```

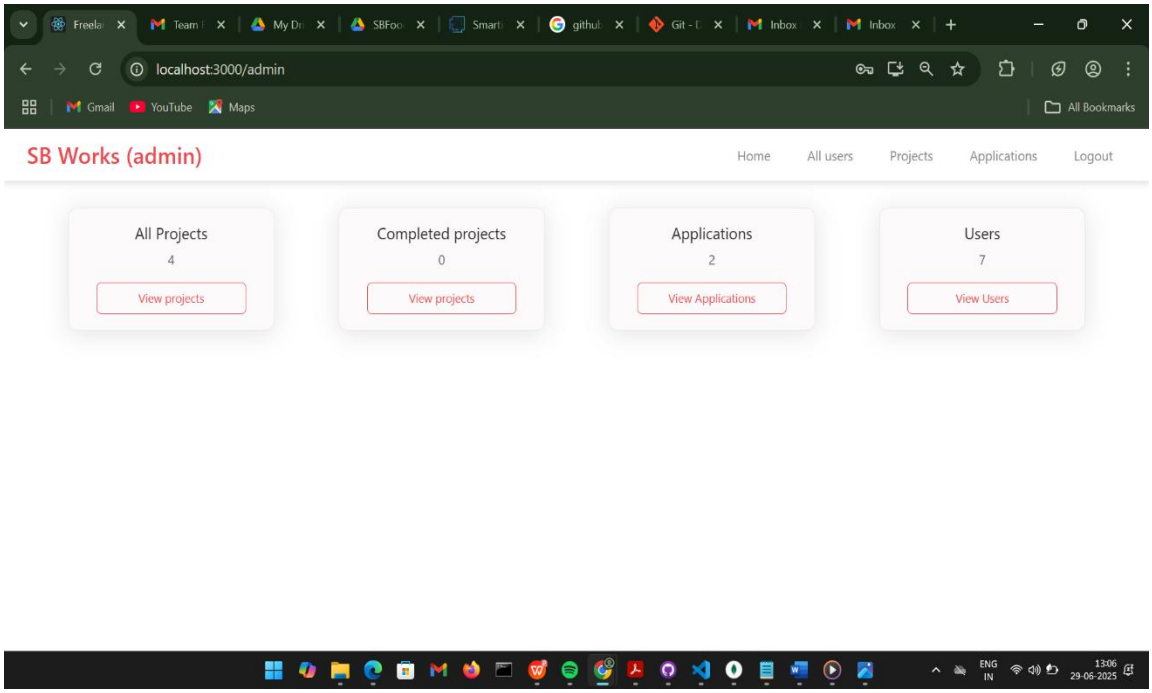
16.Demo pictures

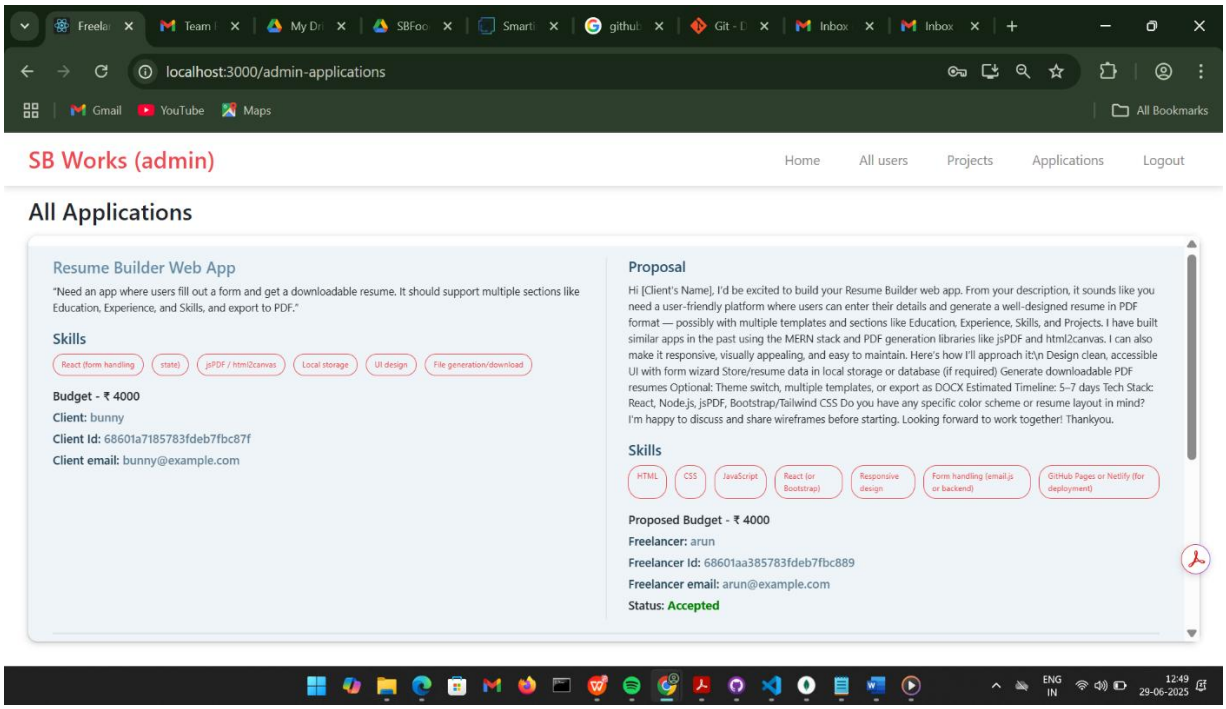
- **Web Interface**



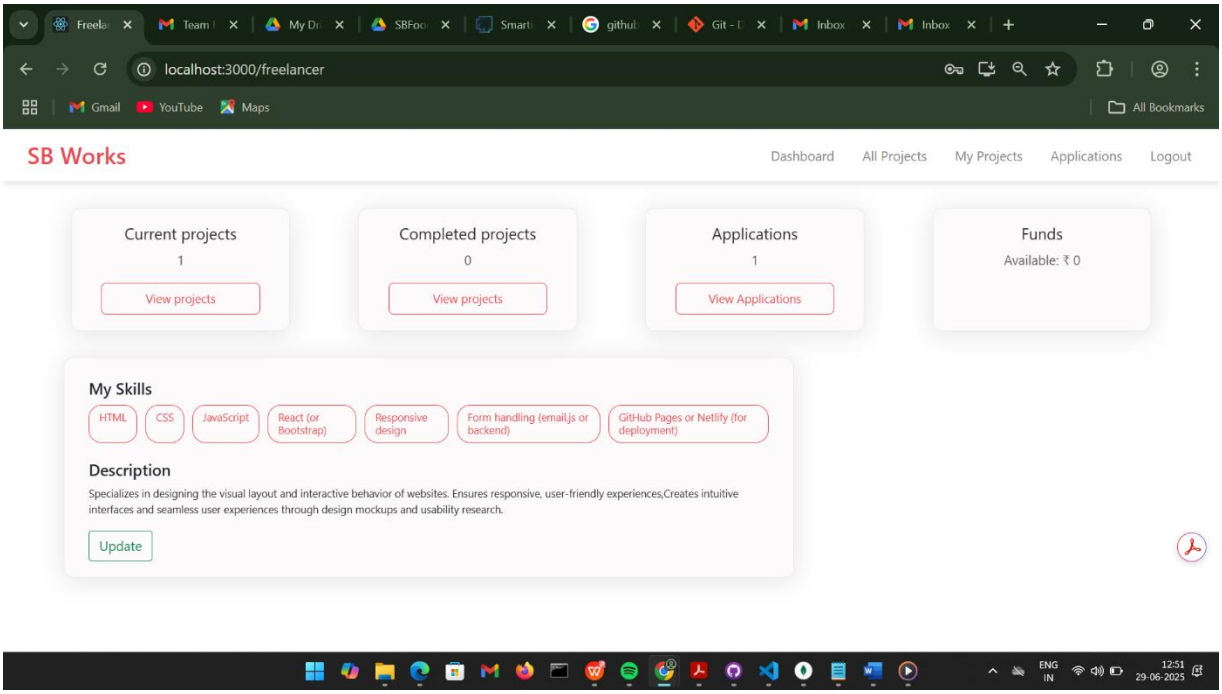


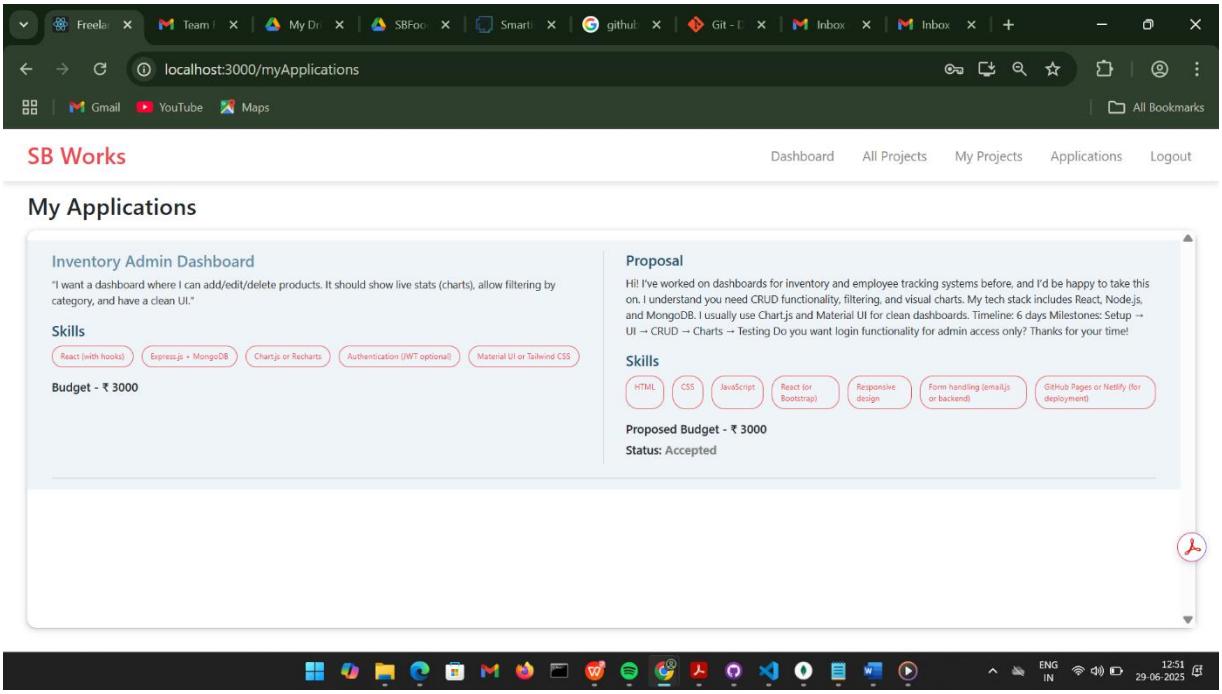
- **Admin' Interface**

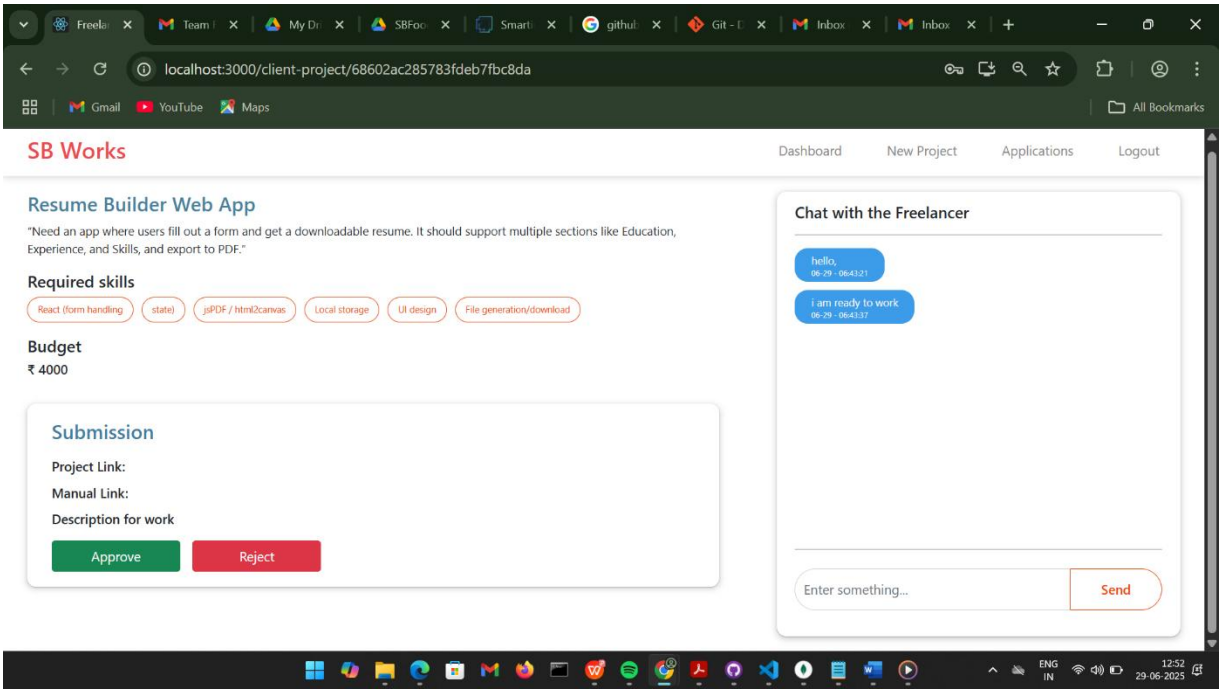




- **Freelancer and Client's Interface**




localhost:3000/myApplications
Gmail
YouTube
Maps
All Bookmarks
SB Works
Dashboard
All Projects
My Projects
Applications
Logout
My Applications
Inventory Admin Dashboard
"I want a dashboard where I can add/edit/delete products. It should show live stats (charts), allow filtering by category, and have a clean UI."
Skills
React (with hooks)
Express.js + MongoDB
Chart.js or Recharts
Authentication (JWT optional)
Material UI or Tailwind CSS
Budget - ₹ 3000
Proposal
Hi! I've worked on dashboards for inventory and employee tracking systems before, and I'd be happy to take this on. I understand you need CRUD functionality, filtering, and visual charts. My tech stack includes React, Node.js, and MongoDB. I usually use Chart.js and Material UI for clean dashboards. Timeline: 6 days Milestones: Setup → UI → CRUD → Charts → Testing Do you want login functionality for admin access only? Thanks for your time!
Skills
HTML
CSS
JavaScript
React (or Bootstrap)
Responsive design
Form handling (email.js or backend)
GitHub Pages or Netlify (for deployment)
Proposed Budget - ₹ 3000
Status: Accepted


localhost:3000/client-project/68602ac285783fdeb7fbc8da
Gmail
YouTube
Maps
All Bookmarks
SB Works
Dashboard
New Project
Applications
Logout
Resume Builder Web App
"Need an app where users fill out a form and get a downloadable resume. It should support multiple sections like Education, Experience, and Skills, and export to PDF."
Required skills
React (form handling)
state)
jsPDF / html2canvas
Local storage
UI design
File generation/download
Budget
₹ 4000
Submission
Project Link:
Manual Link:
Description for work
Approve
Reject
Chat with the Freelancer
hello,
06-29 - 06:43:21
i am ready to work
06-29 - 06:43:37
Enter something...
Send