**🚀 FreelanceFinder: Discovering Opportunities, Unlocking Potential 🔓**

**SHORT TERM INTERNSHIP PROGRAMME - By SmartBridge**

Category : Full Stack Development

Team ID: LTVIP2025TMID55501

Team Size : 4

Team Leader : Team Leader :  Tadikonda Tejanikhil

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE,GUDLAVALLERU

Team member : Sriramadasu Manikumar Chari

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE , GUDLAVALLERU

Team member : Sravya Chinthala

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE , GUDLAVALLERU

Team member : Sudabathula Jahnavi

SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE, GUDLAVALLERU

### **1. Introduction**

* **Project Title**: FreelanceFinder – Discovering Opportunities, Unlocking Potential

### **2. Project Overview**

* **Purpose**:  
   FreelanceFinder is a web application that bridges the gap between freelancers and clients by allowing clients to post freelance projects and freelancers to find and apply for them. It fosters collaboration, boosts visibility, and creates job opportunities in a user-friendly way.
* **Features**:
  + User Authentication (JWT-based)
  + Client/Freelancer dashboards
  + Project posting and application system
  + Real-time updates and messaging
  + Admin control panel
  + Responsive design and modular UI

### **3. Architecture**

* **Frontend**:
  + Built with React.js
  + React Router for navigation
  + Tailored components for dashboards, profile, project posting, messaging
* **Backend**:
  + Node.js with Express.js
  + REST APIs for authentication, posting, messaging, user management
* **Database**:
  + MongoDB via Mongoose
  + Schemas for users, projects, applications, messages

### **4. Setup Instructions**

* **Prerequisites**:
  + Node.js & npm
  + MongoDB
* **Installation**:
  + Clone the GitHub repository
  + Install dependencies:
    - npm install in /client and /server
  + Create .env files:
    - MongoDB URI
    - JWT Secret

### **5. Folder Structure**

* **Client**:
  + /src: Components, Pages, Services, Routes
  + /assets: Styles and media
* **Server**:
  + /routes: API endpoints
  + /controllers: Business logic
  + /models: MongoDB schemas

### **6. Running the Application**

* **Frontend**:  
   cd client && npm start
* **Backend**:  
   cd server && npm start

### **7. API Documentation**

* POST /api/auth/register – User Registration
* POST /api/auth/login – Login & get token
* POST /api/projects – Client posts a project
* GET /api/projects – List all projects
* POST /api/apply/:projectId – Freelancer applies
* POST /api/message – Send a message

### **8. Authentication**

* JWT for secure login
* Tokens stored in localStorage
* Protected routes with middleware

### **9. User Interface**

* Responsive layout for web and mobile
* Simple UI for freelancers and clients
* Functional dashboards and clean navigation
* Tailwind CSS for styling

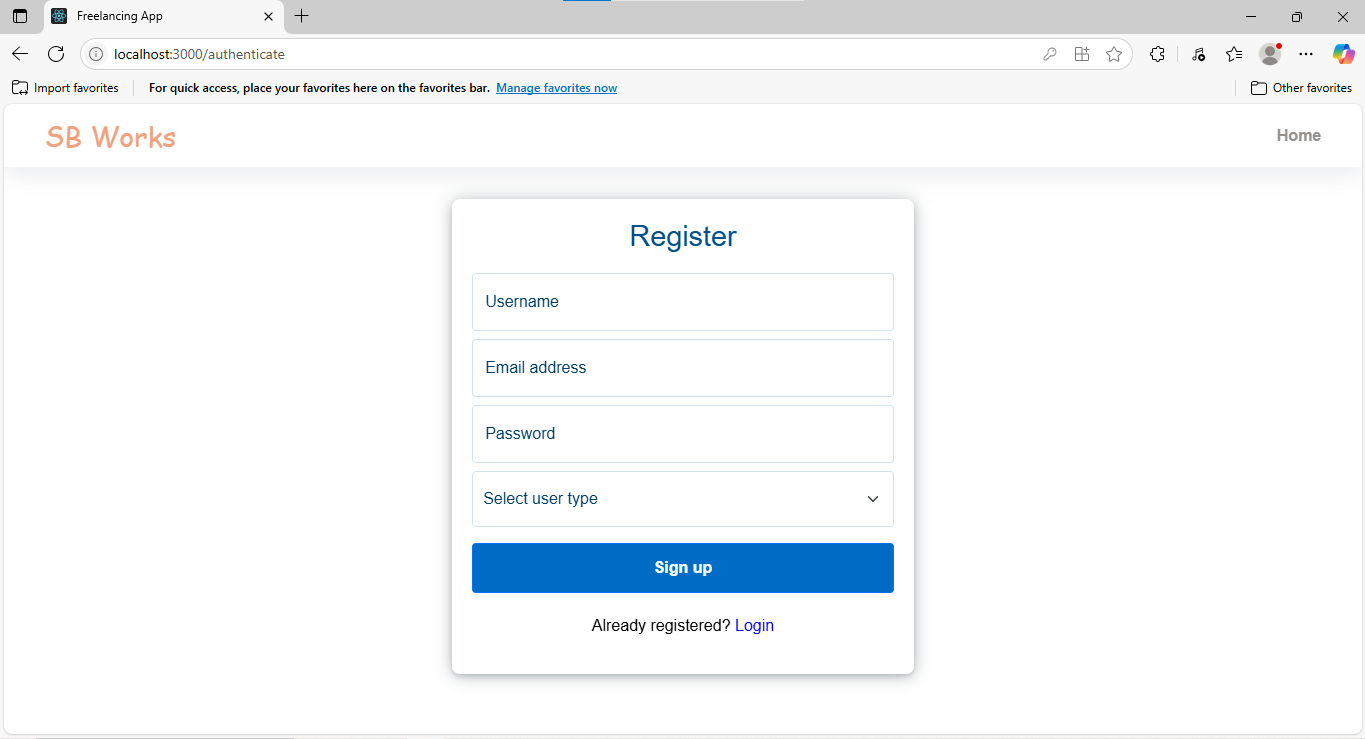
### **10. Testing**

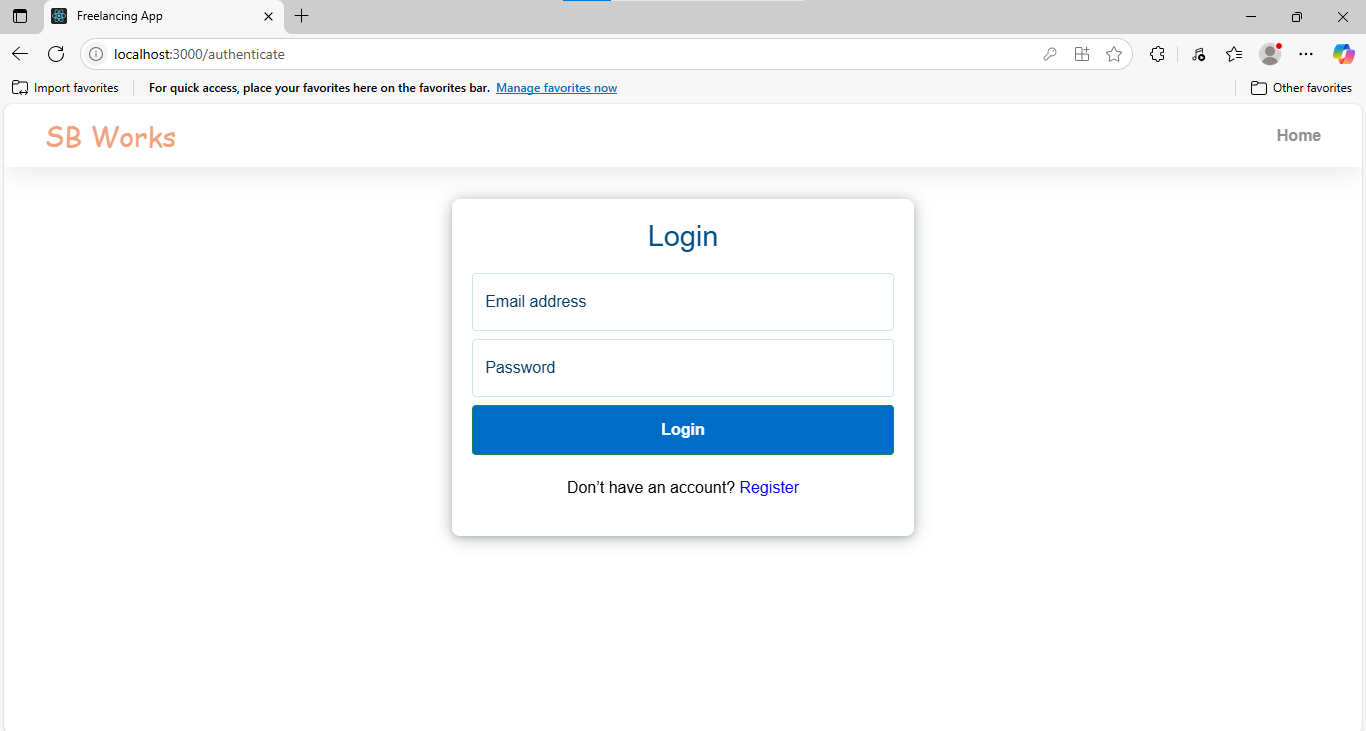
* Manual testing
* Postman for API endpoint verification
* Console logging and error handling

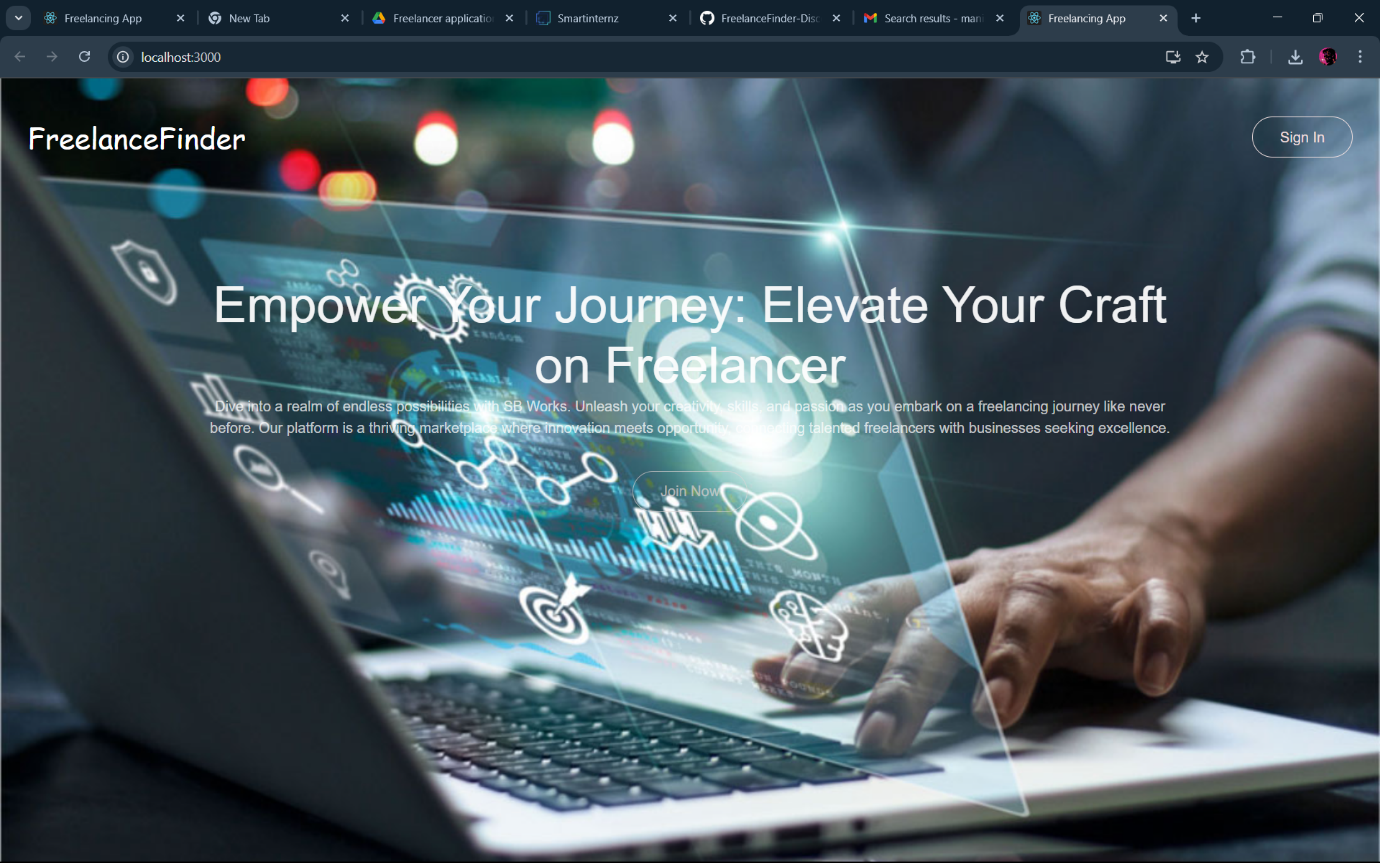
#### **11.🔧 Technologies & Frameworks Used**

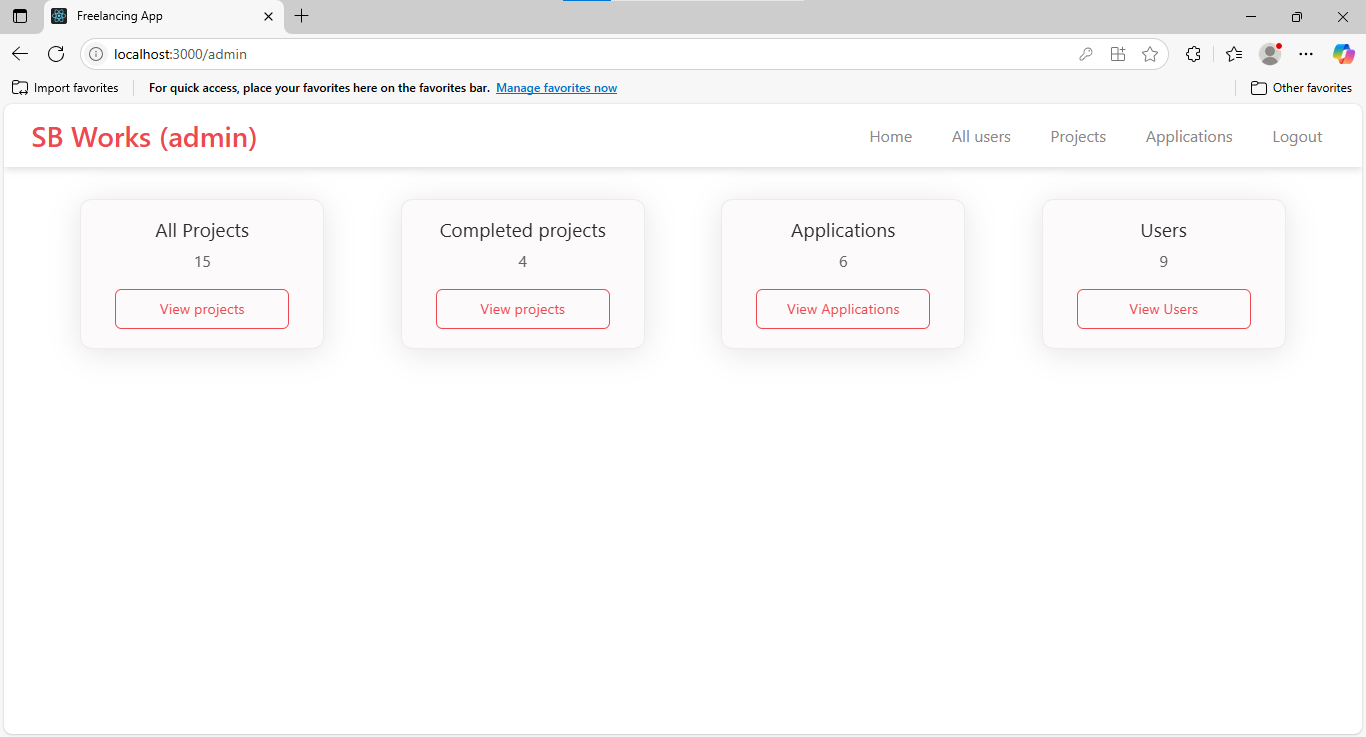
* **React.js Documentation**  
   <https://react.dev/> Official documentation for building the frontend UI of FreelanceFinder.
* **Node.js Documentation**  
   <https://nodejs.org/en/docs> Server-side JavaScript runtime environment used for backend development.
* **Express.js Documentation**  
   <https://expressjs.com/> Web framework for creating RESTful APIs and routing logic.
* **MongoDB Documentation**  
   <https://www.mongodb.com/docs/> NoSQL database used for storing user, project, and application data.
* **Mongoose.js Documentation**  
   <https://mongoosejs.com/docs/> ODM library for MongoDB used to define models and interact with the database.

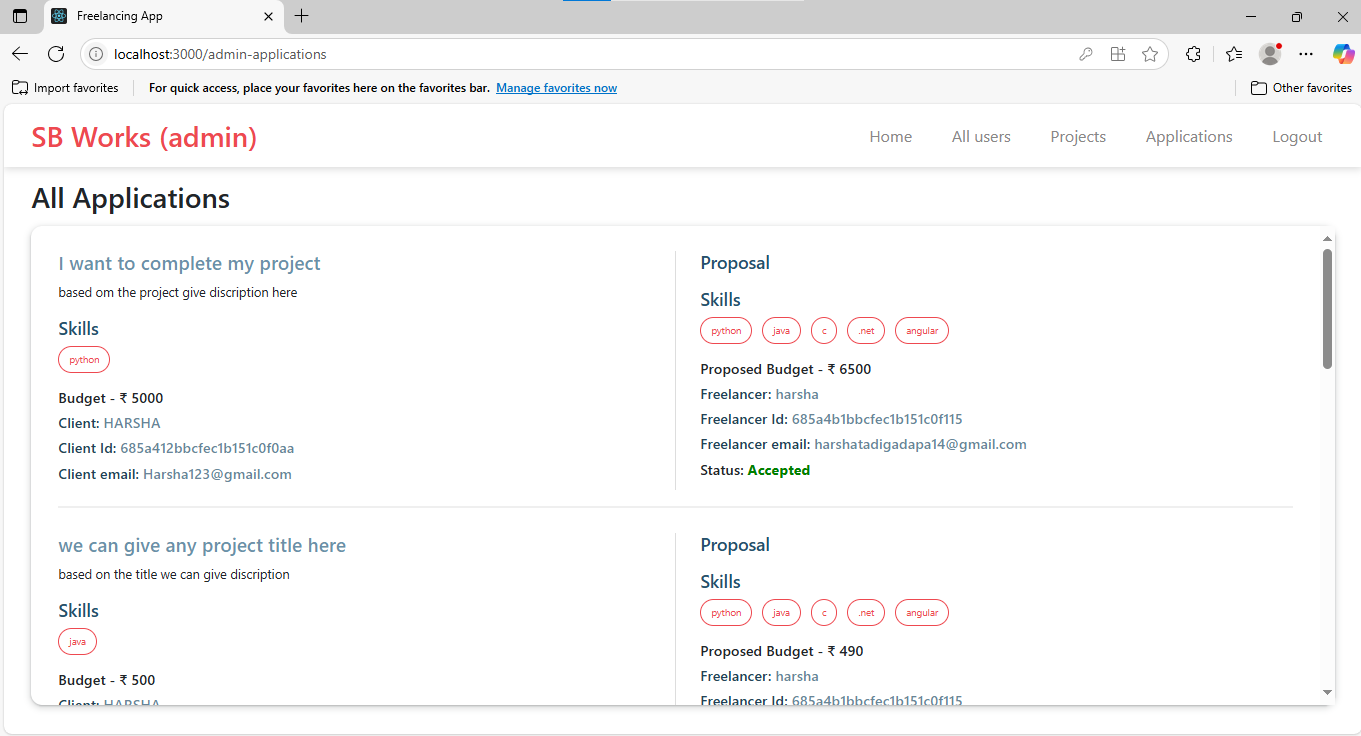
### **12. Screenshots / Demo**



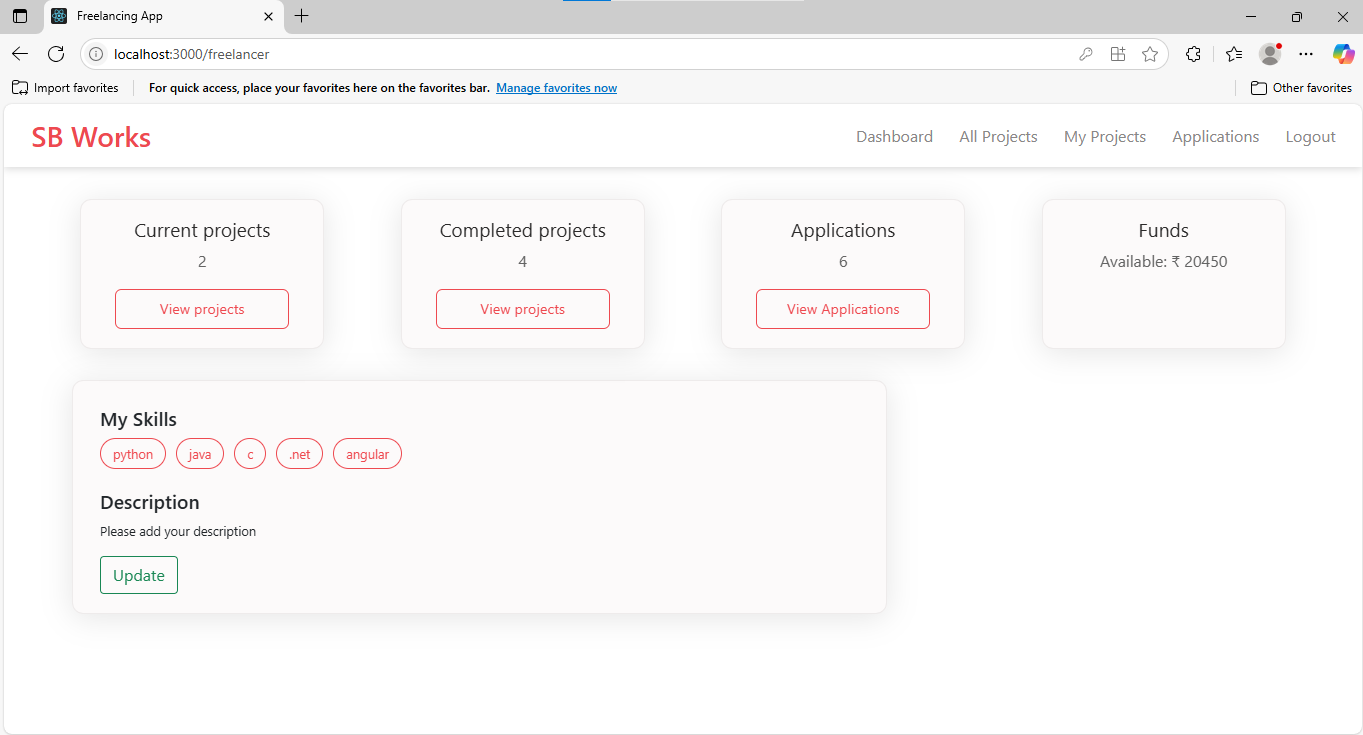


****





### 



### **Demo video link**

[https://drive.google.com/drive/folders/1GYK6uMv5pfE\_faOWAQJoAKWrCSg51YvB? usp=sharing](https://drive.google.com/drive/folders/1GYK6uMv5pfE_faOWAQJoAKWrCSg51YvB?%20%20%20%20%20%20%20usp=sharing)

### **13. Known Issues**

* UI improvements needed
* No real-time chat (currently basic messaging)
* Minimal form validations

### **14. Future Enhancements**

* Real-time messaging with Socket.io
* Skill-based project recommendations
* Email notifications
* Enhanced Admin analytics dashboard

**THANK YOU**