**SB Works – Freelance Finder**

*Full Stack MERN Freelance Platform*

**1. Introduction**

**Project Title:** SB Works – Freelance Finder

**Team Members:**

* **Nidamanuri Naga Lavanya :** Frontend UI design with React, routing, state management, API integration, role-based dashboards, toast notifications, styling with Bootstrap/Material UI, animations, and frontend testing.
* **Puligadda Lakshmi Manasa:** Backend API development, JWT authentication, role-based routing, database schema design, project assignment logic, environment setup, and backend testing.

**2. Project Overview**

SB Works is a comprehensive freelancing platform connecting clients with skilled freelancers. Clients can post projects, freelancers can bid or get assigned, and everyone benefits from a streamlined workflow—from project posting to delivery and review.

**Key Features:**

* Role-based user system: Admin, Client, Freelancer
* Client dashboard to post/manage projects
* Freelancer dashboard to view and update assigned tasks
* Secure JWT-based authentication and authorization
* Real-time toast notifications for user feedback
* Responsive, clean UI built with Bootstrap and Material UI
* Future-ready architecture for chat, ratings, and payments

**3. Architecture**

**Frontend**

* Built using **React** and **TypeScript**
* Uses **React Router** for navigation
* **Axios** for REST API communication
* **Auth Context** manages login state and role-based access
* Styled with Bootstrap and Material UI components

**Backend**

* **Node.js** with **Express.js** RESTful API server
* JWT authentication middleware protects routes
* Controllers separate business logic (e.g., assign freelancer, manage projects)
* Routes organized by functionality and role

**Database**

* **MongoDB** (via Mongoose) stores users, projects, assignments
* Schemas enforce data structure with validation
* Users include roles and relations to projects

**4. Setup Instructions**

**Prerequisites**

* Node.js
* MongoDB
* Git
* Code Editor (VS Code )

**Installation Process**

**1. Frontend Setup**

* Open a terminal and navigate to the frontend folder:

cd SmartInternz\_Freelancer\_Platform/client

* Install dependencies:

npm install

* Start the React development server:

npm start

* The frontend will be accessible at:  
   http://localhost:3000

**2. Backend Setup**

* Open another terminal and navigate to the backend folder:

cd SmartInternz\_Freelancer\_Platform/server

* Install dependencies:

npm install

* Create a .env file inside the server folder with the following environment variables:

PORT=5000

MONGO\_URI=your\_mongodb\_connection\_string

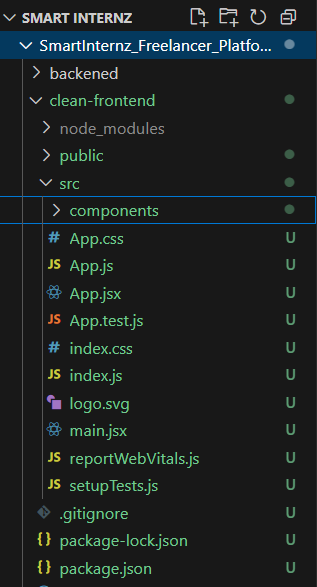
JWT\_SECRET=your\_jwt\_secret\_key

* Start the backend server:

npm start

* The backend API will be running at:  
   http://localhost:5000

**5. Folder Structure**



**6. Running the Application**

1. Start backend server:

🡪 cd backend

🡪node server.js

1. Start frontend server:

🡪cd clean-frontened

🡪npm start

Access the app in your browser: <http://localhost:3000>

**7. Authentication and Authorization**

* User sessions managed by **JWT** tokens
* Tokens stored in browser local Storage after login
* Backend validates tokens on protected routes
* Role-based access:
  + **Admin:** Full access to users/projects
  + **Client:** Create/manage projects
  + **Freelancer:** View assigned projects only

**8. User Interface**

**Admin Dashboard**

* User and project management
* Assign projects to freelancers

**Client Dashboard**

* Post new projects
* Track project status

**Freelancer Dashboard**

* View assigned projects
* Update project status

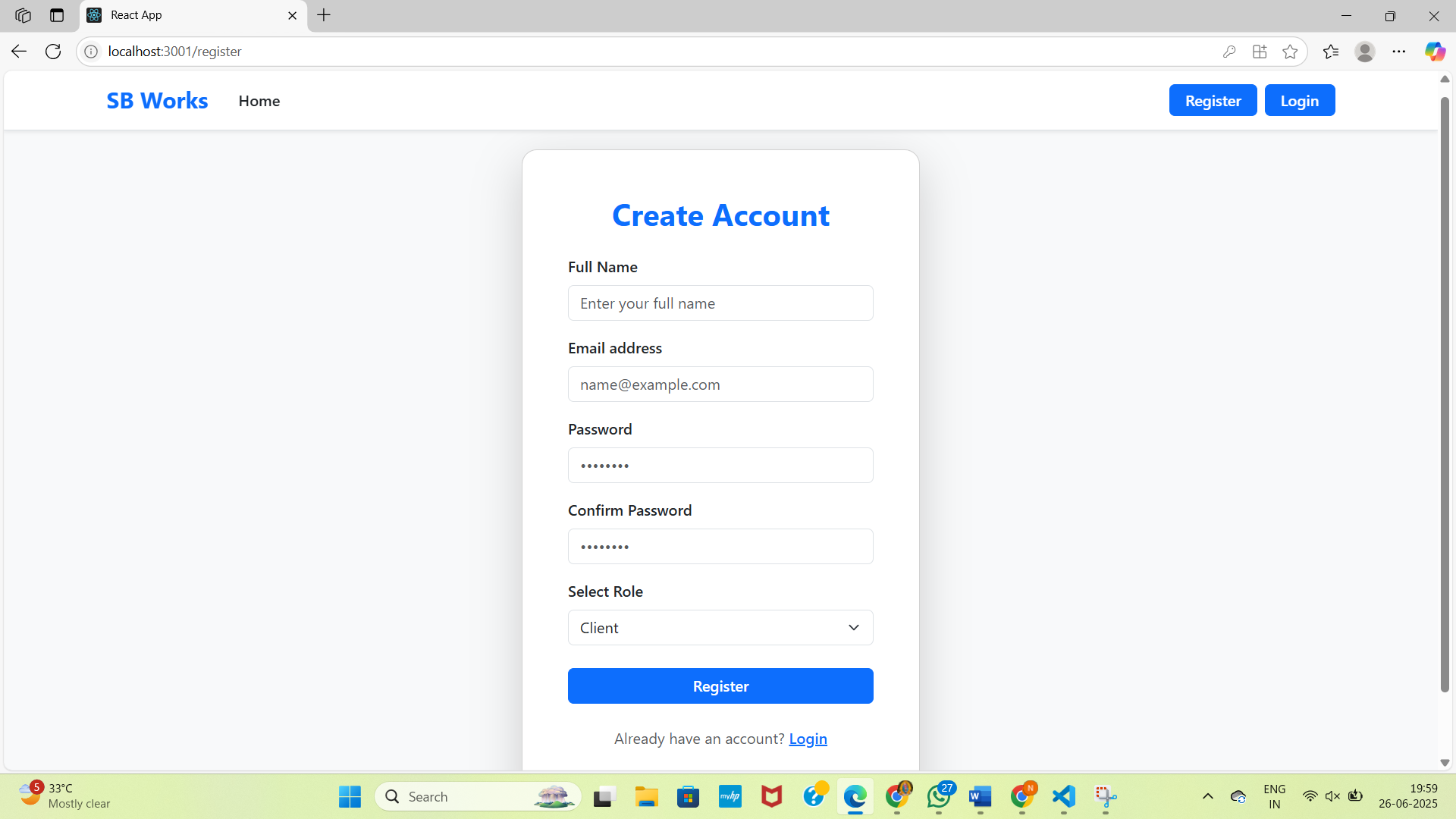
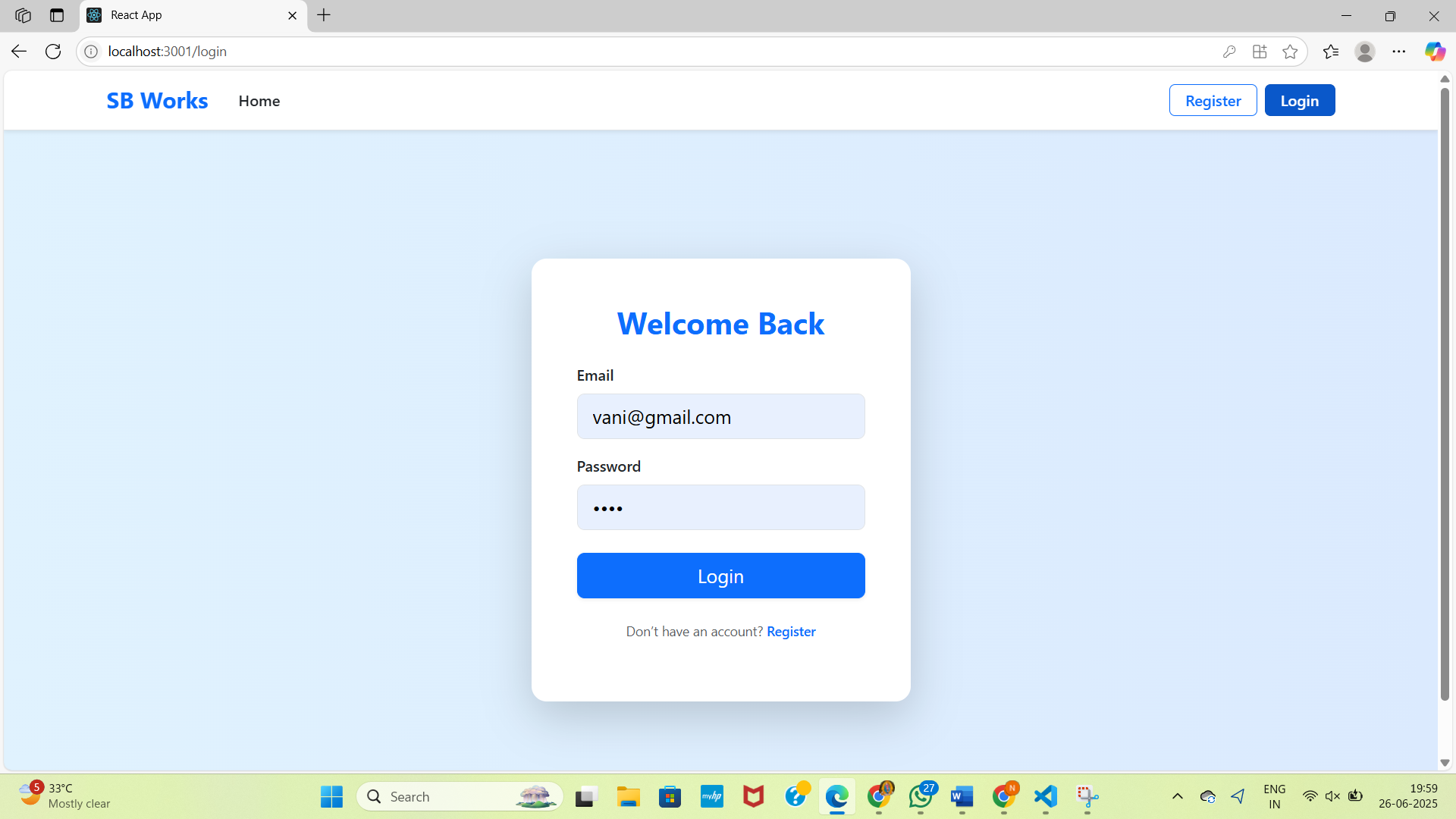
**UI Highlights:**

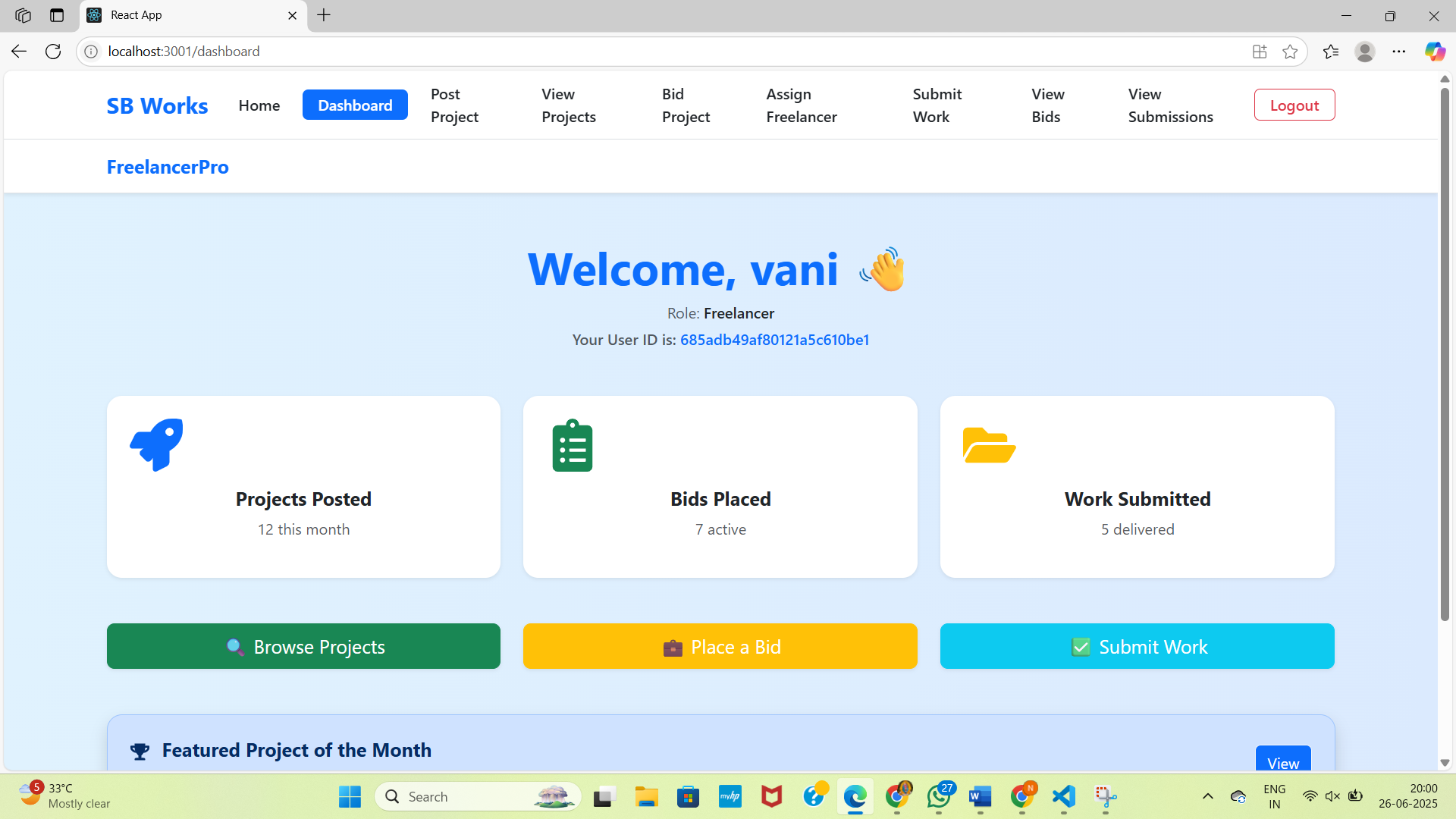
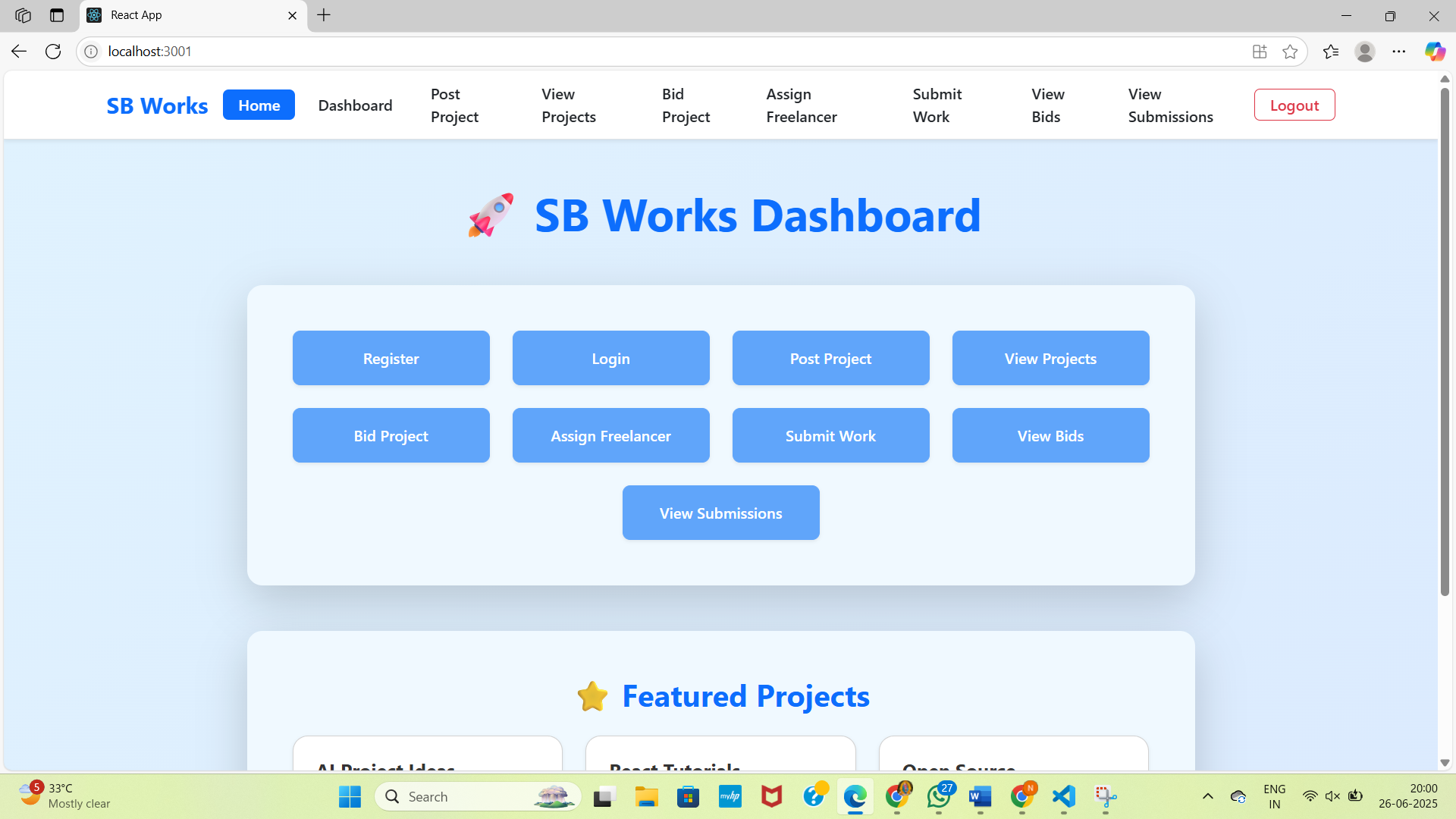
* Responsive design for desktop and mobile
* Bootstrap + Material UI styling
* Toast notifications for user feedback
* Smooth animations with Framer Motion

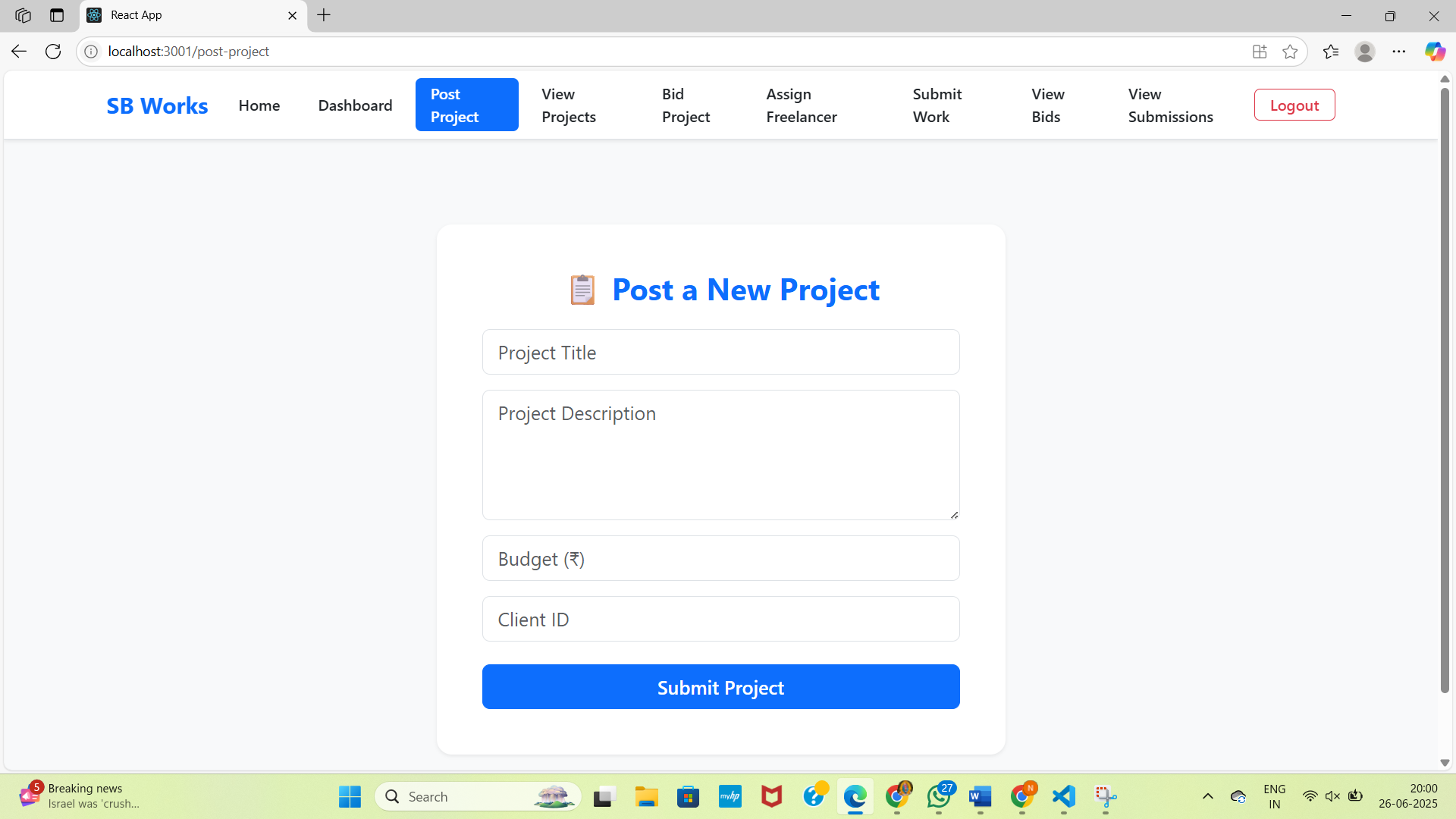
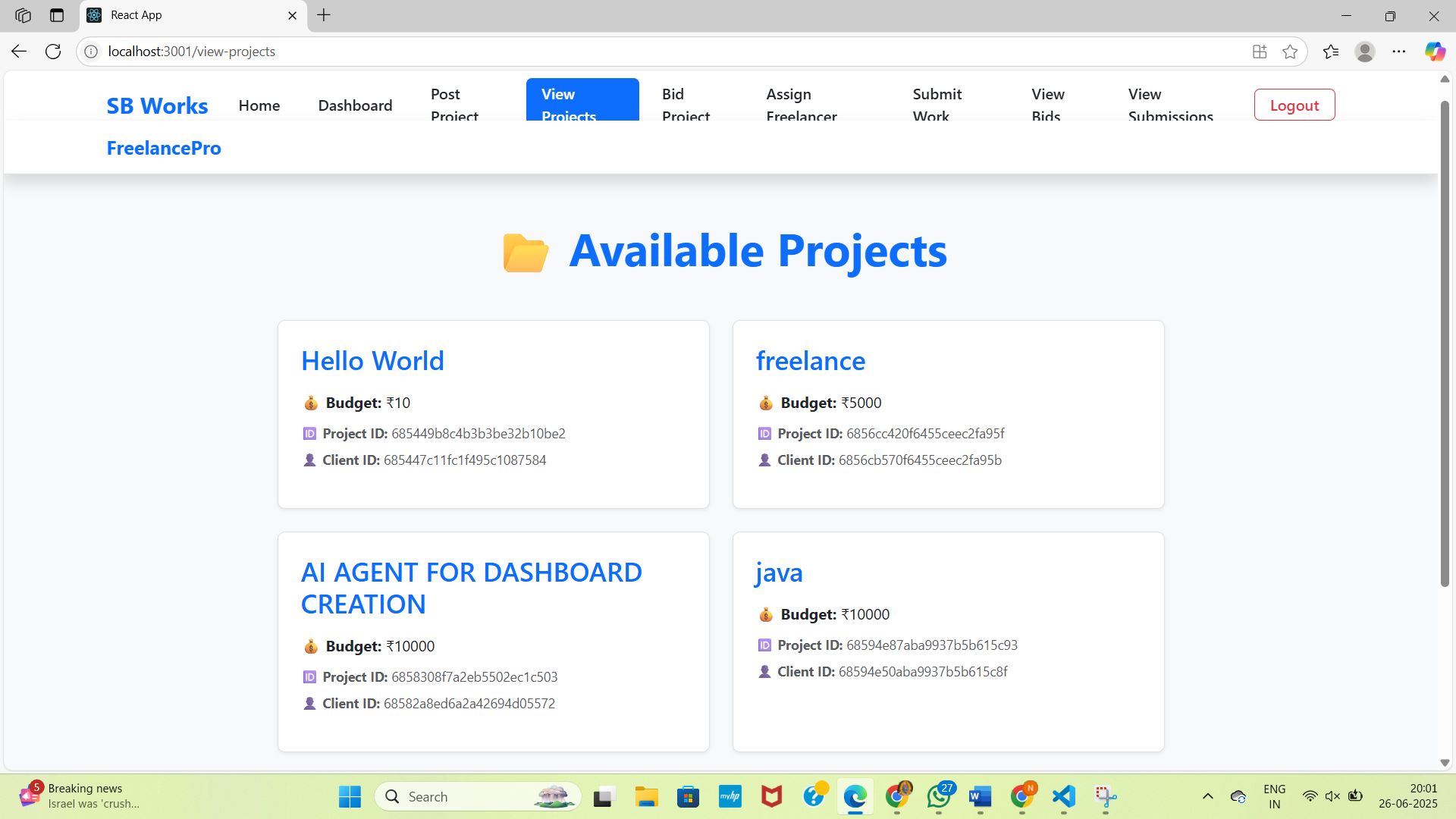
**10. Testing**

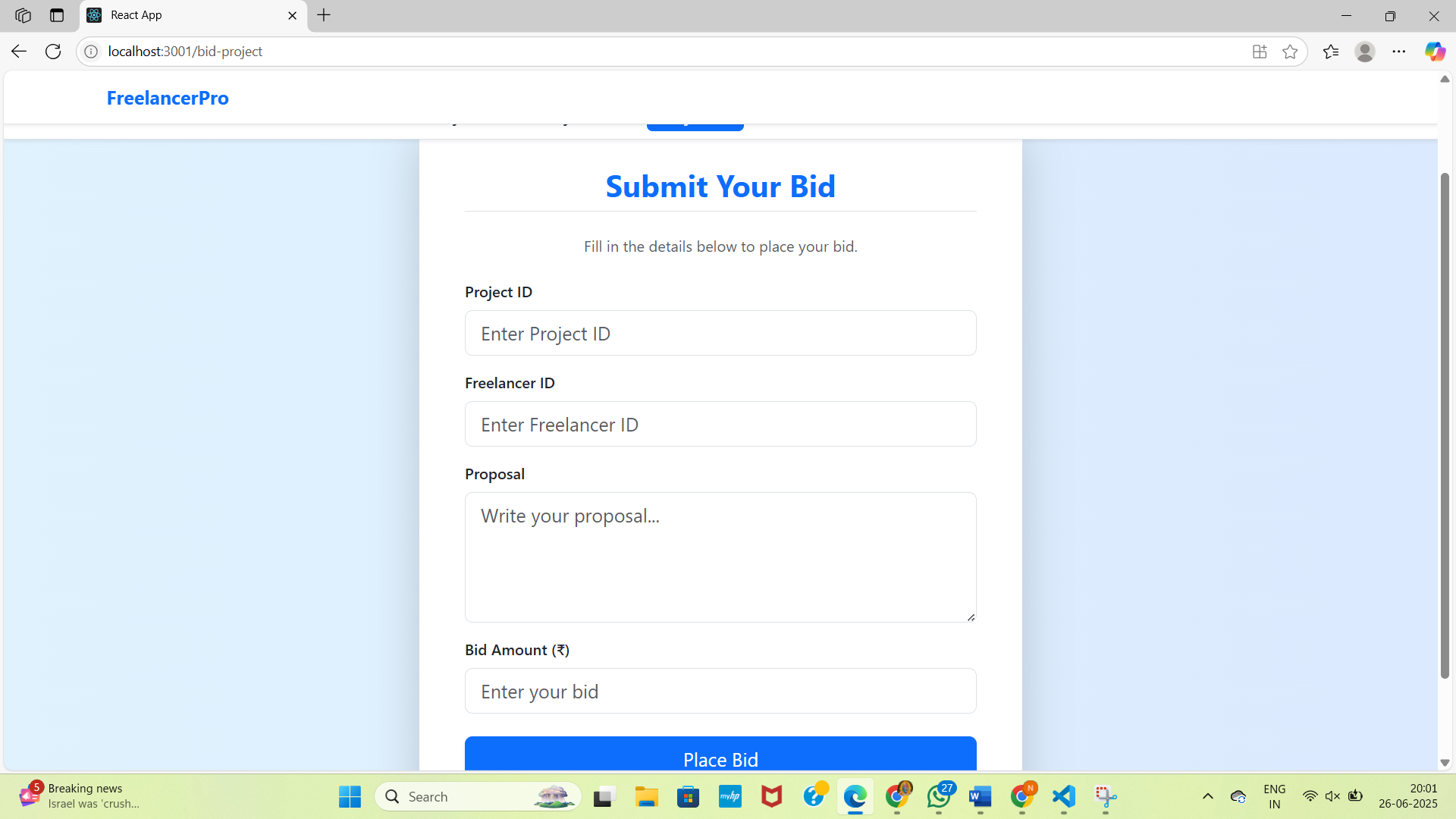
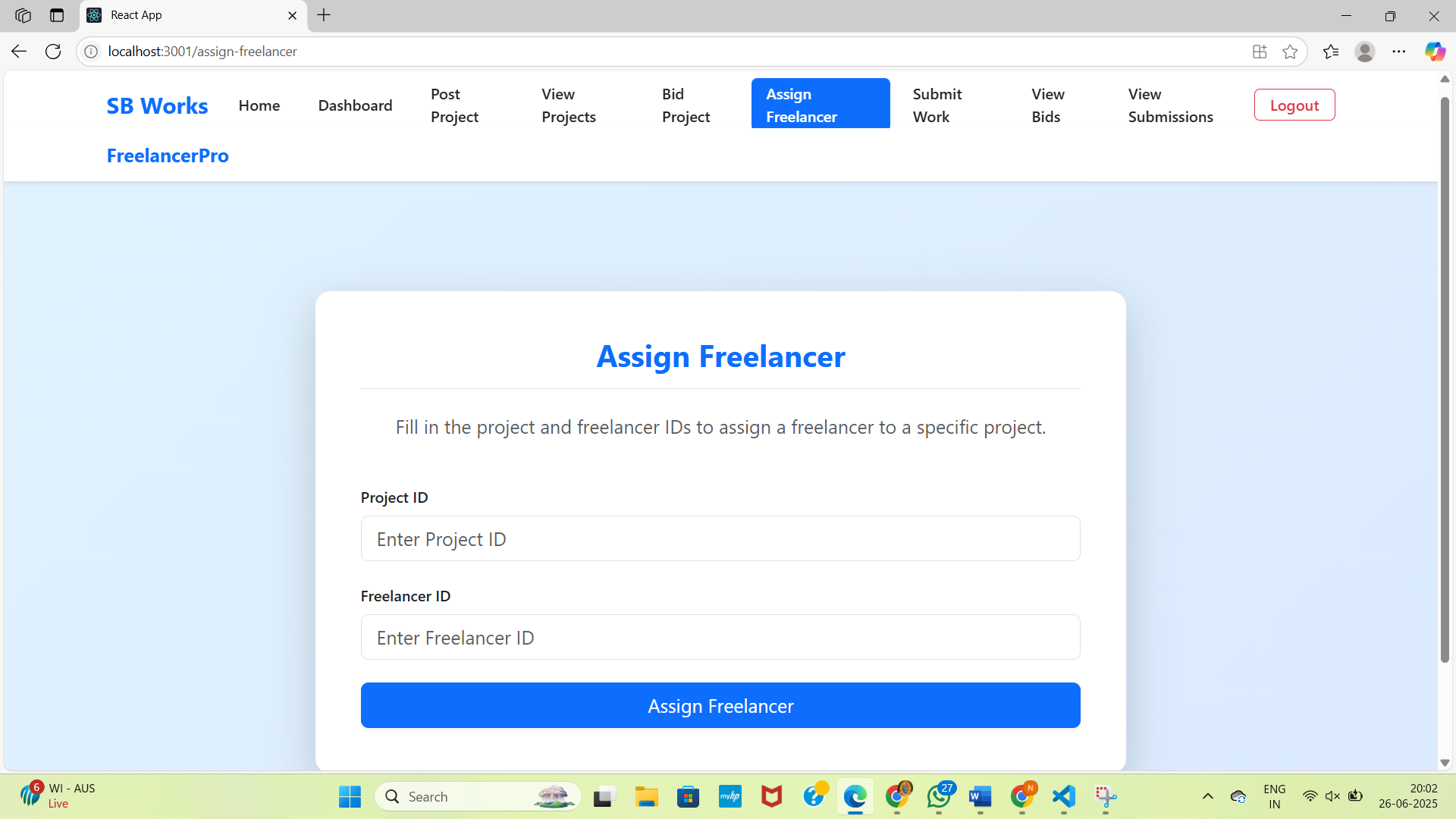
* Manual testing of login flows and role restrictions
* API testing with Postman
* Planned addition: Frontend unit testing with Jest

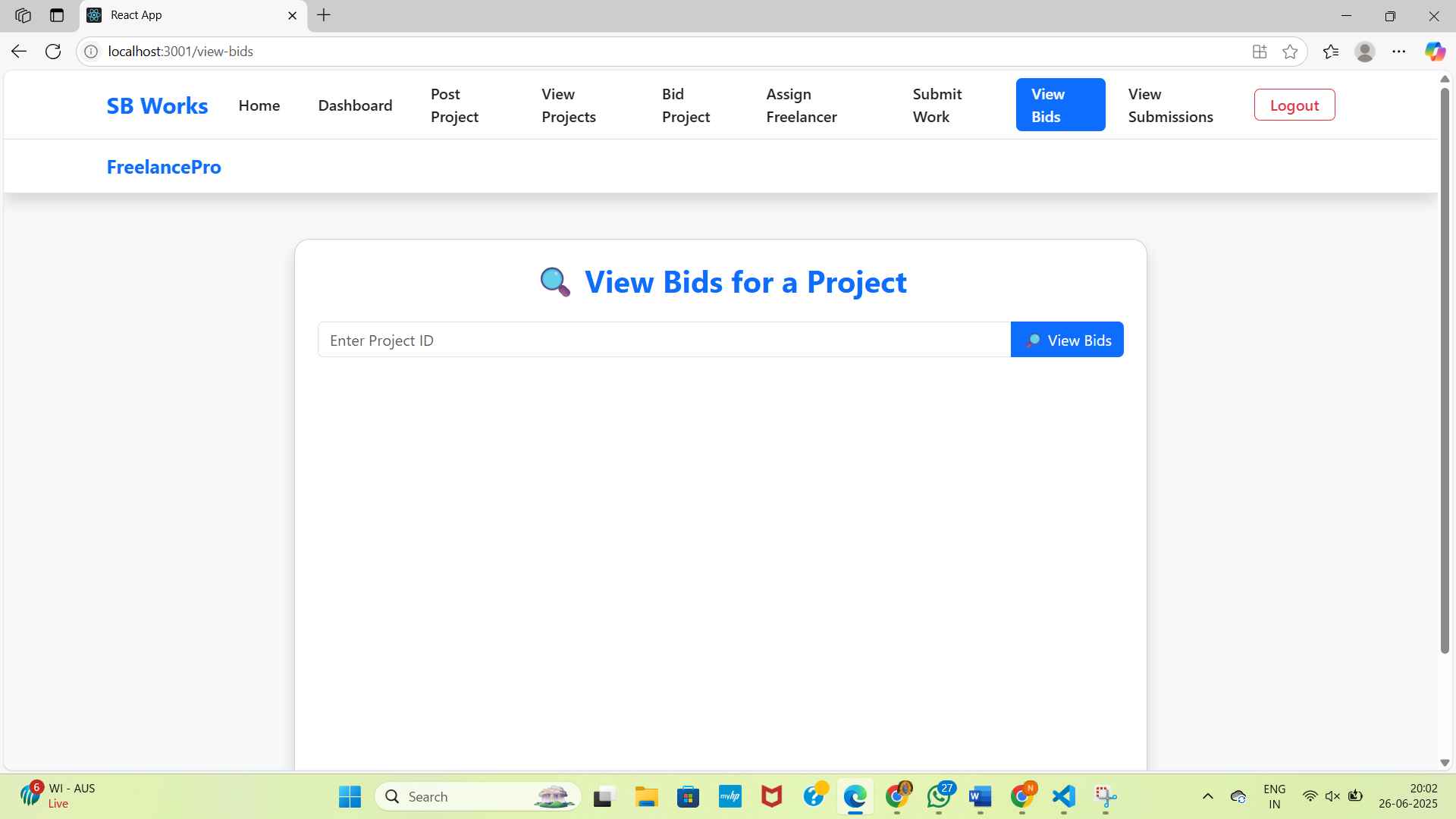
**11. Screenshots**



**12. Known Issues**

* No confirmation modals on assignment actions
* Basic input validation (email/password) needs improvement
* Project lists lack pagination
* Chat and notification systems are future tasks

**13. Future Enhancements**

* Real-time chat between clients and freelancers
* Ratings and reviews system after project completion
* Notification system for assignments and status changes
* Mobile app using React Native
* Payment gateway integration for transactions
* Analytics dashboard for project performance