**​ ​1.** Introduction

​**Project Title:** FitFlex

​**Team ID:**

​**Team Leader**: **MADAVAN N** [**madhavann577@gmail.com**](mailto:madhavann577@gmail.com)

​**Team Member : MADAVAN N** [**madhavann57@gmail.com**](mailto:madhavann57@gmail.com)

**CHANDRU V** [**chandruv8932@gmail.com**](mailto:chandruv8932@gmail.com)

**DHANUSH K** [**cocdhanush31@gmail.com**](mailto:cocdhanush31@gmail.com)

**PRAVEEN M** [**praveencj50@gmail.com**](mailto:praveencj50@gmail.com)

​2. Project Overview

The purpose of this project is to connect clients and freelancers through project postings, bidding, and real-time communication, a system similar to the one described for "SB Works". The key features include:

* ​Project posting and bidding
* ​Secure chat system
* ​Feedback and review system
* ​Admin control panel

​**3. Architecture**

​Frontend: The frontend is built with React.js and styled with Bootstrap and Material UI.

​Backend: The backend uses Node.js and Express.js to manage server logic and API endpoints.

​Database: MongoDB is used to store user data, project information, applications, and chat messages.

​4. Setup Instructions

* ​Prerequisites
* ​Node.js
* ​MongoDB
* ​Git
* ​React.js
* ​Express.js
* ​Mongoose
* ​Visual Studio Code
* ​Installation Steps
* ​Clone the repository using Git clone.
* ​Navigate to the client directory and install dependencies with npm install.
* ​Navigate to the server directory and install dependencies with npm install.

​5. Folder Structure

​The project has a two-part folder structure:

​SB-Works/

​client/ (React frontend)

​\_components/

​\_pages/

​server/ (Node.js backend)

​\_routes/

​\_models/

​\_controllers/

​6. Running the Application

​Frontend: Navigate to the client directory and run npm start.

​Backend: Navigate to the server directory and run npm start.

​Access: The application is accessible by visiting http://localhost:3000.

​7. API Documentation

​**User:**

​/api/user/register

​/api/user/login

​**Projects:**

​/api/projects/create

​/api/projects/:id

​Applications: /api/apply

​**Chats:**

​/api/chat/send

​/api/chat/:userld

​8. Authentication

The system uses JWT-based authentication for secure login and middleware to protect private routes.

​

​The user interface includes the following pages:

​Landing Page

​Freelancer Dashboard

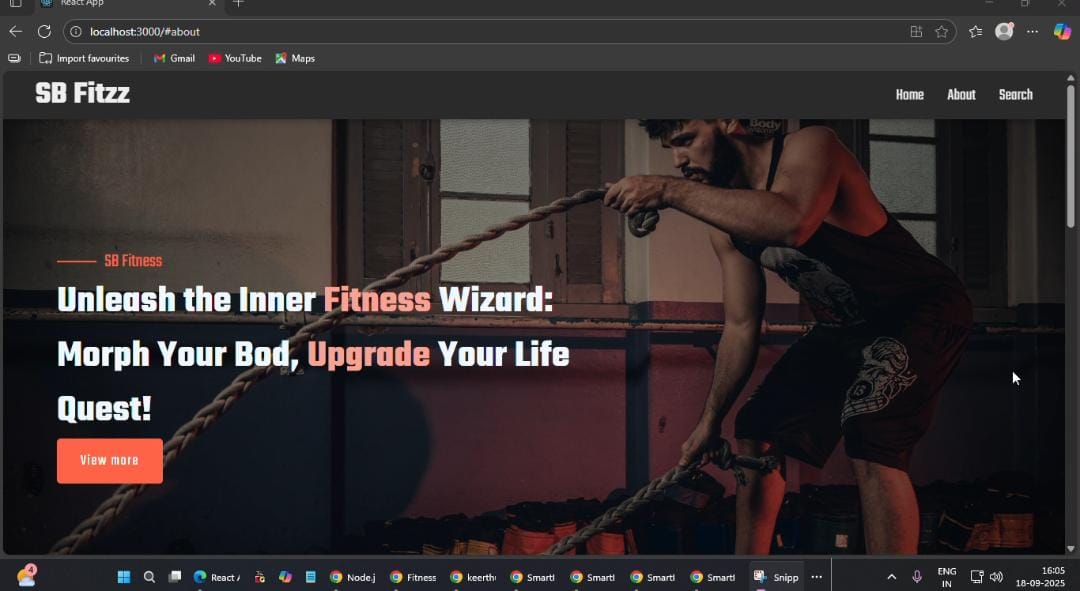
​Admin Panel

​Project Details Page

​10. Testing

​Testing is performed manually during milestones using tools such as Postman and Chrome Dev Tools.

​11. Screenshots

​

​12. Known Issues

* ​ **Lag on Older Devices Animation**
* Animations in the UI may not run smoothly on older or lower-performance hardware. Users with less capable devices may experience visible lag.
* **Progress Tracking Performance Issues**
* As data grows (e.g. many past workouts, large logs), the progress tracker feature may require optimization. Large data entries slow it down.

​13. Future Enhancements

* ​UI/UX: smoother animations, accessibility, customizable themes.
* Tracking: AI workout suggestions, gamification, detailed analytics, export/share progress.
* Personalization: multiple profiles, nutrition logging, wearable integration.
* Performance: cloud database, offline-first sync, faster caching.
* Community: group challenges, forums, live sessions.
* Security: encryption, OAuth login, user data control.
* Maintenance: more automated tests, CI/CD pipeline, error monitoring.