**Project Documentation**

**1. Introduction**

Project Title**: FitFlex Your Personal Fitness Companion**

Team ID: [NM2025TMID42716]

Team Leader: [ SWETHA M & [mswethaswetha330@gmail.com](mailto:mswethaswetha330@gmail.com) ]

Team Members: [ SARANYA D & [saranya16112006@gmail.com](mailto:saranya16112006@gmail.com) ]

[ NISHADAVI B & [mnisha9965@gmail.com](mailto:mnisha9965@gmail.com) ]

[ SAKTHI NILA M & [sakthinilasakthinilam@gmail.com](mailto:sakthinilasakthinilam@gmail.com) ]

[NISHA R & [nishasurya8807@gmail.com](mailto:nishasurya8807@gmail.com)

**2. Project Overview**

Purpose:

FitFlex is a personalized fitness companion application that helps users track workouts, monitor diet, set fitness goals, and stay motivated with AI-driven recommendations.

**Features:**

1.User registration and profile management

2.Personalized workout and diet plans

3.Activity and calorie tracking

4.Progress visualization with charts and analytics

5.Community & social sharing features

6.AI-powered recommendations based on fitness goals

7.Admin panel for content and user management

**3. Architecture**

**-**Frontend: React.js with Tailwind CSS and Material UI

**-** Backend: Node.js with Express.js handling APIs and business logic

**-** Database: MongoDB for storing user profiles, workouts, and progress logs

**4. Setup Instructions**

1.Prerequisites:

2.Node.js

3.MongoDB

4.Git

5.React.js

6.Express.js

7.Visual Studio Code

**Installation Steps**:

# Clone the repository

git clone <repo-link>

# Install client dependencies

cd client

npm install

# Install server dependencies

cd ../server

npm install

**5. Folder Structure**

FitFlex/

│-- client/ # React frontend

│ ├── components/

│ ├── pages/

│ └── assets/

│

│-- server/ # Node.js backend

│ ├── routes/

│ ├── models/

│ ├── controllers/

│ └── middleware/

**6. Running the Application**

**Frontend:**

cd client

npm start

**Backend:**

cd server

npm start

Access:

Visit: http://localhost:3000

**7. API Documentation**

User:

POST /api/user/register

POST /api/user/login

Workouts:

POST /api/workouts/create

GET /api/workouts/:id

Diet Plans:

POST /api/diet/create

GET /api/diet/:id

Progress Tracking:

POST /api/progress/log

GET /api/progress/:userId

**8. Authentication**

JWT-based authentication for user login and registration

Middleware to protect private

routes

**9. User Interface**

Landing Page – overview and signup/login

Dashboard – personalized fitness plan and daily summary

Workout Tracker – track exercises and calories burned

Diet Tracker – log meals and calorie intake

Progress Charts – visualize weight, calories, and goals

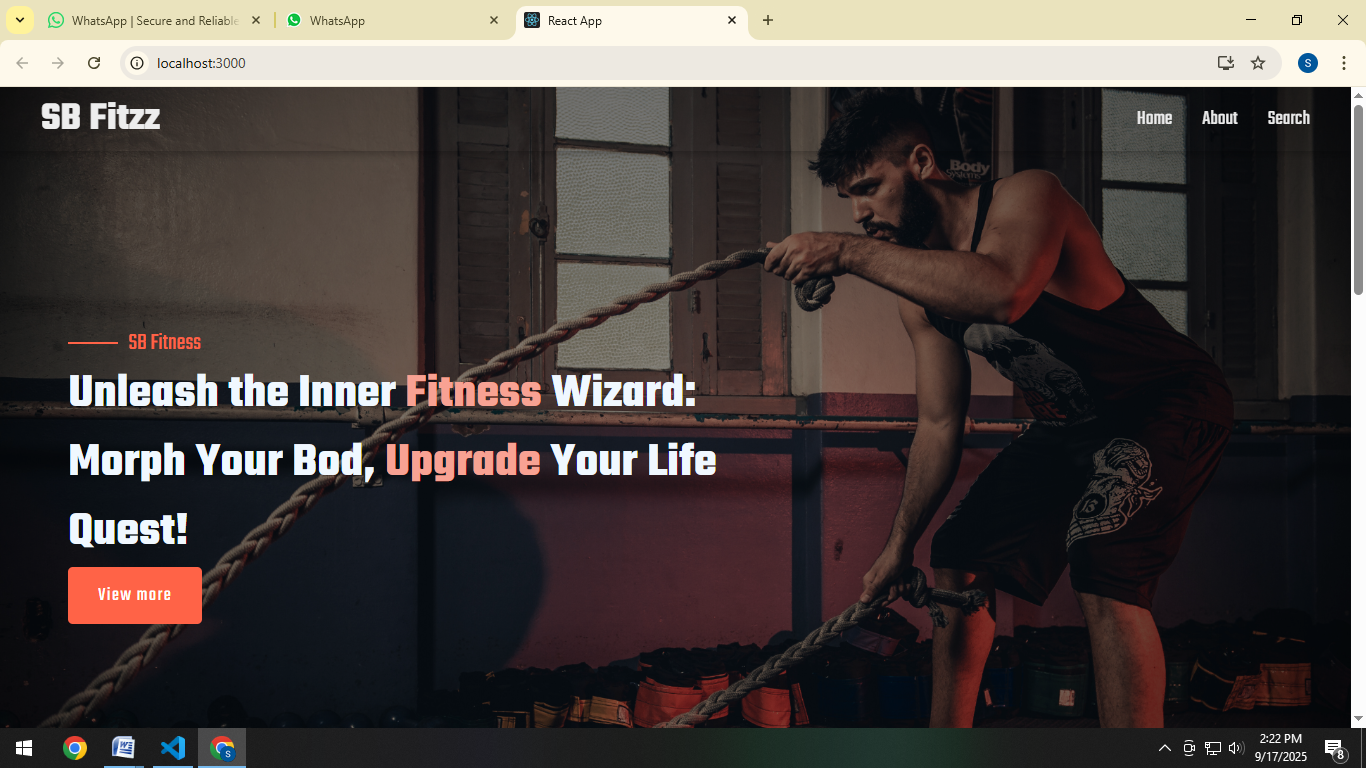
Admin Panel – manage users, workouts, and diets

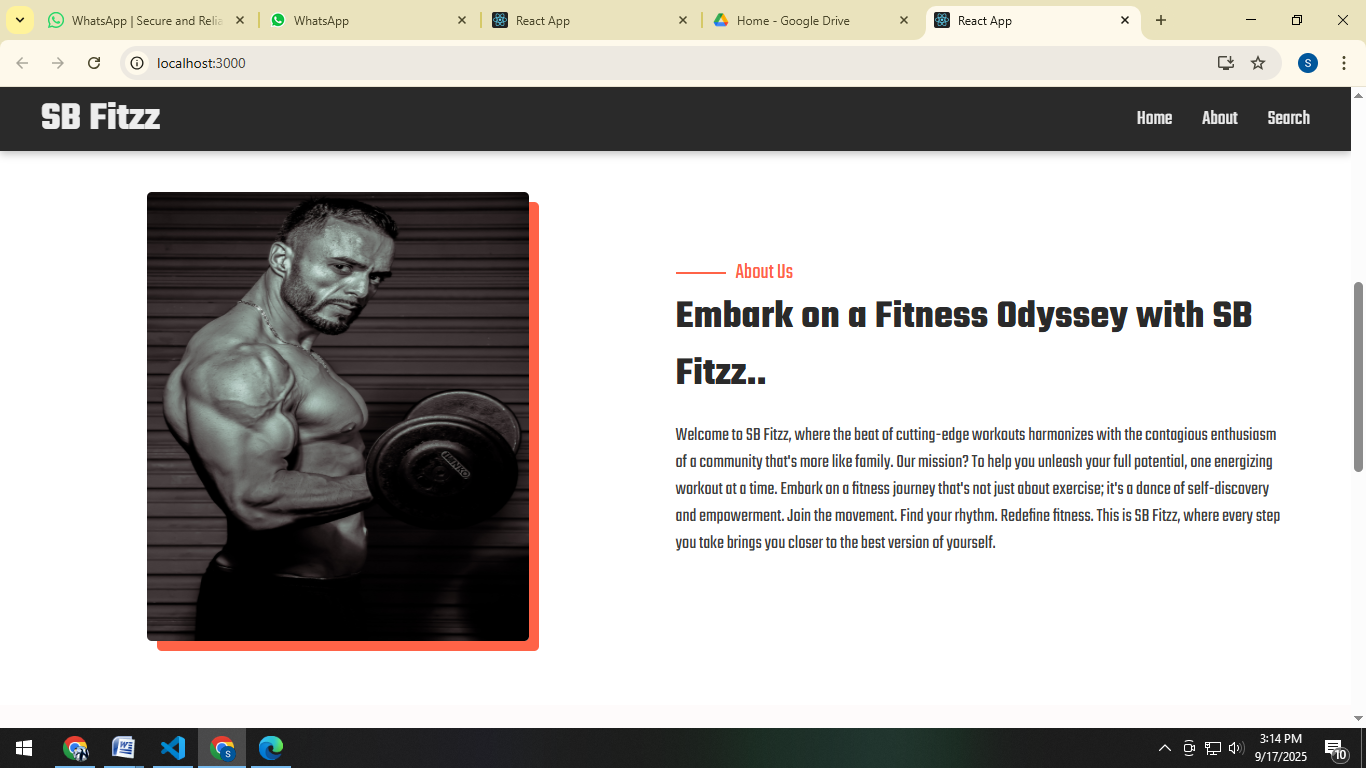
**10. Testing**

Manual testing during development

Tools: Postman, Chrome Dev Tools, Jest for unit testing

**11. Screenshots or Demo**





**12. Known Issues**

Limited support for offline mode

Requires manual input for some food items

**13. Future Enhancements**

Integration with wearable devices (smartwatches, fitness bands)

AI chatbot for instant fitness guidance

Voice-assisted workout mode

Enhanced gamification features (badges, leaderboards)