

Project Documentation

FITFLEX: YOUR PERSONAL FITNESS COMPANION

1. Introduction:

Project Title: Fitness – Your Personal Fitness Companion

Team ID: NM2025TMID37641

Team Leader: Preetha.D & preethadavid1029@gmail.com]

Team Members: Monika S & monikaajith04@gmail.com

Team Members: Priyadharshini E & vanithapriya36@gmail.com]

Team Members: Vasantharani P & krishnaraj052@gmail.com

Team Members: Sivaranjini R & sivaranjinirajendran35@gmail.com]

2. Project Overview

Purpose:

Fitness is a personal fitness companion app designed to help users track workouts, monitor diet plans, log progress, and stay motivated through personalized recommendations and analytics.

Features:

- User registration & login
- Personalized workout plans
- Meal and calorie tracking
- Progress tracking (weight, BMI, workout stats)

Notifications & reminders

Social/community features (optional future add-on)

Admin dashboard for managing content and users

3. Architecture:

Frontend: React.js with Bootstrap and Material UI

Backend: Node.js and Express.js for server logic and APIs

Database: MongoDB for storing user profiles, workouts, meals, progress logs

Authentication: JWT for secure login and protected routes

4. Setup Instructions

Prerequisites:

- Node.js
- MongoDB
- Git
- React.js
- Express.js
- Mongoose
- Visual Studio Code

Installation Steps:

```
# Clone the repository
```

```
git clone <repo-url>
```

```
# Install client dependencies
```

```
cd client
```

```
npm install
```

```
# Install server dependencies
```

```
cd ../server
```

```
npm install
```

5. Folder Structure

```
Fitness-App/
```

```
| -- client/          # React frontend
```

```
| | -- components/
```

```
| | -- pages/
```

```
|
```

```
| -- server/         # Node.js backend
```

```
| | -- routes/
```

```
| | -- models/
```

```
| | -- controllers/
```

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

Meals & Nutrition:

POST /api/meals/add

GET /api/meals/:userId

Progress:

POST /api/progress/log

GET /api/progress/:userId

8. Authentication

JWT-based authentication for secure login

Middleware to protect private routes (workout logs, meal logs, progress tracking)

9. User Interface

Landing Page (App intro, sign-up/login)

User Dashboard (overview of workouts, meals, progress)

Workout Page (exercise plans, logs)

Meal & Nutrition Page

Progress Tracking Page (graphs & analytics)

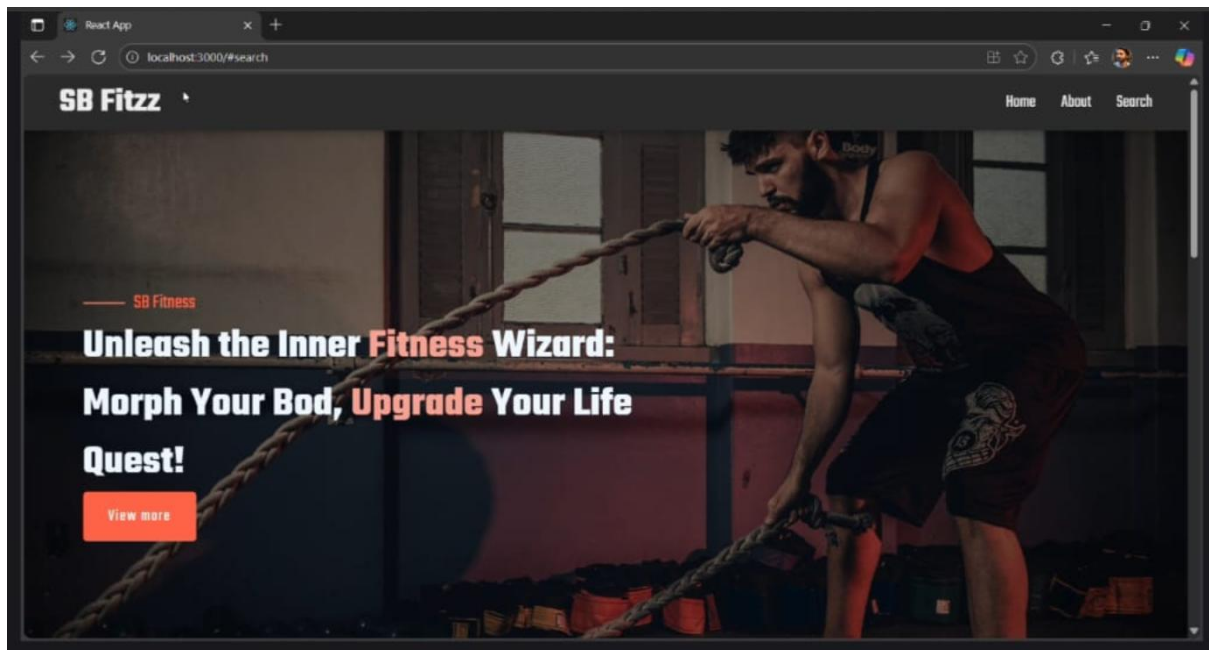
Admin Panel

10. Testing

Manual testing at milestones

Tools: Postman (API testing), Chrome DevTools (frontend debugging)

11. Screenshots or Demo



12. Known Issues

Pending integration with wearable devices (fitness bands, smartwatches)

Limited offline support

13. Future Enhancements

- AI-based personalized workout & meal suggestions
- Integration with fitness trackers (Fitbit, Apple Watch)

- Community forum for user interaction
- Gamification features (badges, challenges)