# Frontend Development with React.js

Project Documentation for Fitness App

## Introduction

Project Title: FitFlex : Your Personal Fitness companion

Team leader:LUBNAA FIRDOWS

Team members:

NOORUNISHA S

SUFFIYA A

## Project Overview

Purpose: The Fitness App is designed to help users track workouts, monitor progress, and stay motivated through personalized fitness plans.

Features:  
- User authentication (login/signup)  
- Dashboard with progress tracking  
- Workout plans and exercise library  
- Nutrition tracking  
- Responsive design

## Architecture

Component Structure: Major components include Navbar, Dashboard, WorkoutList, WorkoutDetail, Profile, and Authentication components.

State Management: The app uses Context API for global state management, along with local component states where appropriate.

Routing: Implemented using React Router DOM for navigation between pages.

## Setup Instructions

Prerequisites:  
- Node.js (v16 or higher)  
- npm or yarn package manager

Installation:  
1. Clone the repository.  
2. Navigate to the project directory.  
3. Run 'npm install' to install dependencies.  
4. Configure environment variables if required.  
5. Run 'npm start' to start the development server.

## Folder Structure

Client:  
- src/components: Reusable UI components  
- src/pages: Application pages  
- src/assets: Images, fonts, and static files  
- src/context: Context API providers  
- src/utils: Helper functions and custom hooks

## Running the Application

To run the frontend locally:  
- Navigate to the client directory  
- Run 'npm start'

## Component Documentation

Key Components:  
- Navbar: Provides navigation across the app  
- Dashboard: Displays user progress and quick actions  
- WorkoutList: Lists available workouts  
- WorkoutDetail: Shows detailed workout instructions  
- Profile: Displays and updates user information  
- Auth Components: Handles login/signup/logout

Reusable Components:  
- Button, Input, Modal, Card

## State Management

Global State: Managed via Context API (UserContext, WorkoutContext, etc.).  
Local State: Used within components for UI interactions (e.g., toggling modals).

## User Interface

The app features a clean and responsive UI with dashboards, charts, and forms to ensure smooth interaction across devices.

## Styling

CSS Frameworks/Libraries: Tailwind CSS is used for styling.  
Theming: Custom themes are implemented for light and dark modes.

## Testing

Testing Strategy: Components are tested using Jest and React Testing Library.  
Code Coverage: Achieved through automated unit and integration tests.

## Screenshots or Demo

Screenshots and demo links can be added here to showcase application features.

## Known Issues

Currently, some responsive issues exist on smaller devices. Performance optimization is also ongoing.

## Future Enhancements

- Integration with wearable devices for real-time tracking  
- Advanced analytics dashboard  
- Social features (friends, challenges)  
- Offline mode support