- Fit App Documentation
  - Overview
  - Table of Contents
  - Features
    - 🏠 Dashboard & Home
    - 🤖 AI Coach (Neeed FIT AI)
    - 🏋️ Workout Management
    - 🥗 Diet & Nutrition
    - 👥 Social Features
    - 📊 Health Reports
    - 🔐 User Management
  - Technical Architecture
    - Frontend Framework
    - Backend & Database
    - Key Libraries & Integrations
  - Component Structure
    - Core Components (/src/components/)
      - UI Components
      - Specialized Components
      - Utility Components
    - Feature Components (/src/Feature/)
      - Home Dashboard (/Feature/Home/)
      - AI Coach (/Feature/AiCoach/)
      - Diet Management (/Feature/MyDiet/)
      - User Onboarding (/Feature/Splash/)
    - Pages (/src/app/)
      - Main Application Pages
      - Dynamic Routes
  - Key Packages & Dependencies
    - Core Framework
    - Authentication & Database
    - AI & Machine Learning
    - UI & Styling
    - Data Management
    - Specialized Features
  - Database Schema
    - Collections

        - users
        - workoutPlans
        - weight
        - diet_AI
    - Development Guide
        - Project Structure
        - Key Development Patterns
            - State Management
            - Authentication Flow
            - Data Fetching
        - Getting Started
        - Contributing Guidelines
    - Architecture Decisions
        - Why Next.js 14?
        - Why Firebase?
        - Why Clerk?

Fit App Documentation

Overview

Fit App is a comprehensive fitness and health tracking application built with Next.js 14. It provides users with personalized workout plans, diet tracking, AI coaching, and health monitoring features.

Table of Contents

- Features
- Technical Architecture
- Component Structure
- Key Packages & Dependencies
- Database Schema
- Development Guide

Features

🏠 Dashboard & Home

- **Personal fitness dashboard** with BMI tracking, calorie calculations, and daily goals
- **Health metrics visualization** with progress tracking
- **Quick access to all app features** through an intuitive interface

🤖 AI Coach (Neeed FIT AI)

- **AI-powered workout planning** using Google's Generative AI
- **Interactive chat interface** for personalized fitness guidance
- **Real-time workout recommendations** based on user preferences

🏋️ Workout Management

- **Custom workout plan creation** with exercise selection
- **Saved workout plans** with detailed exercise instructions
- **Exercise execution tracking** with real-time progress monitoring
- **Set and reps tracking** with rest timers
- **Body part visualization** for targeted workouts

🥗 Diet & Nutrition

- **Diet plan creation and tracking**
- **Macro nutrient monitoring** (proteins, carbs, fats)
- **Food image analysis** for automatic nutrition detection
- **Weekly meal planning** with calendar view
- **Calorie tracking** and goal setting

👥 Social Features

- **Instructor enrollment** and client management

- **Mentor profiles** and coaching services
- **Client progress tracking** for trainers

📊 Health Reports

- **Comprehensive health analytics**
- **Progress visualization** with charts and graphs
- **Historical data tracking**
- **BMI and weight progression**

🔐 User Management

- **Secure authentication** via Clerk
- **User profile management**
- **Onboarding flow** with personal metrics setup
- **Role-based access** (user, coach, admin)

Technical Architecture

Frontend Framework

- **Next.js 14** with App Router
- **React 18** with modern hooks and patterns
- **Tailwind CSS** for styling with custom design system
- **Framer Motion** for smooth animations

Backend & Database

- **Firebase Firestore** for data storage
- **Supabase** for additional database operations
- **Clerk** for authentication and user management

Key Libraries & Integrations

- **Google Generative AI** for AI coaching features
- **Material Tailwind** for UI components
- **React Query** for state management and caching
- **Razorpay** for payment processing
- **React Webcam** for image capture

Component Structure

Core Components (`/src/components/`)

UI Components

- **Button/** - Various button styles (Regular, Pill, Footer, Custom)
- **Card/** - Reusable card components for different content types
- **InputCs/** - Form input components with validation
- **Navbar/** - Navigation components (Floating, Standard)
- **Sidebar/** - App navigation sidebar with user menu

Specialized Components

- **BlurryBlob/** - Animated background elements
- **WeightScale/** - Interactive weight selection component
- **StopWatch/** - Timer component for workout tracking
- **ExerciseCard/** - Exercise display and interaction
- **WorkoutChat/** - AI chat interface component

Utility Components

- **SecureComponent/** - Authentication wrapper
- **Toast/** - Notification system
- **ProgressBar/** - Visual progress indicators
- **FileUpload/** - Image and file upload handling

Feature Components (`/src/Feature/`)

Home Dashboard (`/Feature/Home/`)

- **Dashboard.jsx** - Main dashboard with health metrics
- **BMICard.jsx** - BMI calculation and display
- **CaloriesCard.jsx** - Daily calorie tracking

AI Coach (`/Feature/AiCoach/`)

- **ExerciseAiCard.jsx** - AI-generated exercise recommendations
- **MealPlan.jsx** - AI-powered meal planning

Diet Management (`/Feature/MyDiet/`)

- **MacroTracker.jsx** - Macro nutrient tracking interface
- **ImageAnalysisModal.jsx** - Food image analysis popup
- **WeeklyCalendar.jsx** - Meal planning calendar
- **PlannedMeal.jsx** - Individual meal display

User Onboarding (`/Feature/Splash/`)

- **Splash.jsx** - Welcome screen
- **GenderSelection.jsx** - Gender selection interface
- **HeightSelection.jsx** - Height input component
- **WeightSelection.jsx** - Weight input component
- **ActivityLevel.jsx** - Activity level selection
- **AgeSelection.jsx** - Age input component

Pages (`/src/app/`)

Main Application Pages

- **/** - Home dashboard
- **/AICoach/** - AI coaching interface
- **/MyDiet/** - Diet tracking and planning
- **/SavedPlan/** - Workout plan management
- **/profile/** - User profile settings
- **/healthReport/** - Health analytics

Dynamic Routes

- **/SavedPlan/[plan]/** - Individual workout plan execution
- **/diets/[dietId]/** - Specific diet plan details
- **/createPlanPage/** - Workout plan creation wizard

Key Packages & Dependencies

Core Framework

```
{
  "next": "14.2.11",
  "react": "^18",
  "react-dom": "^18"
}
```

Authentication & Database

```
{
  "@clerk/nextjs": "^5.7.2",
  "firebase": "^11.4.0",
  "@supabase/supabase-js": "^2.45.5"
}
```

AI & Machine Learning

```
{
  "@google/generative-ai": "^0.24.1"
}
```

UI & Styling

```
{
  "tailwindcss": "^3.4.11",
  "@material-tailwind/react": "^2.1.10",
  "framer-motion": "^11.11.11",
```

```
  "bootstrap": "^5.3.3",
  "react-bootstrap": "^2.10.4"
}
```

Data Management

```
{
  "@tanstack/react-query": "^5.56.2",
  "axios": "^1.7.7",
  "lodash": "^4.17.21"
}
```

Specialized Features

```
{
  "react-webcam": "^7.2.0",
  "react-calendar": "^5.1.0",
  "recharts": "^2.15.1",
  "razorpay": "^2.9.6",
  "next-pwa": "^5.6.0"
}
```

Database Schema

Collections

users

- User profile information
- Personal metrics (height, weight, age, gender)
- Activity level and fitness goals
- Onboarding completion status

workoutPlans

- Custom workout plans created by users

- Exercise lists with sets, reps, and instructions
- Plan metadata (name, description, difficulty)

weight

- Weight tracking history
- Timestamp-based entries
- User association via userIdCl

diet_AI

- AI-generated diet plans
- Meal recommendations and nutritional information
- User-specific dietary preferences

Development Guide

Project Structure

```
/src
├── app/                 # Next.js 14 App Router pages
├── components/          # Reusable UI components
├── Feature/            # Feature-specific components
├── context/            # React Context providers
├── hooks/              # Custom React hooks
├── service/            # API service functions
├── utils/              # Utility functions
├── config/             # Configuration files
└── firebase/           # Firebase configuration
```

Key Development Patterns

State Management

- **Global Context** for app-wide state (user data, authentication)
- **React Query** for server state management and caching
- **Local state** for component-specific data

Authentication Flow

- **Clerk** handles user authentication
- **SecureComponent** wrapper protects routes
- **Role-based access** for different user types

Data Fetching

- **Firebase** for real-time data operations
- **Custom hooks** for data fetching logic
- **Error handling** and loading states

Getting Started

1. **Install dependencies:**

```
npm install
```

2. **Set up environment variables:**
    - Clerk authentication keys
    - Firebase configuration
    - Google AI API key
    - Razorpay credentials
3. **Run development server:**

```
npm run dev
```

4. **Build for production:**

```
npm run build
```

Contributing Guidelines

1. **Component Creation:**
    - Follow existing naming conventions
    - Use TypeScript for type safety
    - Implement proper error boundaries
    - Add loading states for async operations
2. **Styling:**
    - Use Tailwind CSS classes
    - Follow the existing design system
    - Ensure responsive design
3. **State Management:**
    - Use context for global state
    - Implement proper data validation
    - Handle error states gracefully
4. **Testing:**
    - Write unit tests for utility functions
    - Test component interactions
    - Validate API integrations

Architecture Decisions

Why Next.js 14?

- **App Router** for modern routing patterns
- **Server-side rendering** for better performance
- **Built-in optimization** for production builds
- **API routes** for backend functionality

Why Firebase?

- **Real-time database** for live updates
- **Scalable cloud infrastructure**

- **Easy integration** with authentication
- **Offline support** capabilities

Why Clerk?

- **Complete authentication solution**
- **Social login integration**
- **User management dashboard**
- **Role-based access control**

This documentation provides a comprehensive overview of the Fit App structure and functionality. For specific implementation details, refer to the individual component files and their inline documentation.