

# Frontend Development with React.js

## Project Documentation For cookbook

---

### 1. Introduction

- **Project Title:** Cookbook
  - **Team Members:**
    - Parthiban N(TeamLeader) [sakthiparthi1980@gmail.com](mailto:sakthiparthi1980@gmail.com)
    - Sathish V [sathish.v7092@gmail.com](mailto:sathish.v7092@gmail.com)
    - Gopalakrishnan K [gopalakrishnan709246@gmail.com](mailto:gopalakrishnan709246@gmail.com)
    - Manoj S [mmanoj7559@gmail.com](mailto:mmanoj7559@gmail.com)
    - Sarathi V [v.sarathi5557@gmail.com](mailto:v.sarathi5557@gmail.com)
- 

### 2. Project Overview

- **Purpose:** The application helps users monitor fitness activities, track progress, and set fitness goals. It includes features like workout logging, calorie tracking, and progress visualization.
  - **Features:**
    - User authentication (login/signup)
    - Dashboard for daily activity tracking
    - Workout logging and history
    - Calorie tracker
    - Progress charts and analytics
    - Responsive design for mobile and desktop
- 

### 3. Architecture

- **Component Structure:**
  - **App Component:** Manages routing and global state.
  - **Dashboard Component:** Displays daily fitness metrics.

- **WorkoutLog Component:** Allows users to log and view workout history.
  - **Auth Component:** Handles user authentication.
  - **State Management:**
    - **Redux:** Used for global state management (user authentication, workout data, calorie tracking).
    - **Local State:** Managed using React's useState and useEffect hooks.
  - **Routing:**
    - **React Router:** Used for navigation between pages.
- 

## 4. Setup Instructions

- **Prerequisites:**
    - Node.js (v16 or higher)
    - npm (v8 or higher)
    - Git (for cloning the repository)
  - **Installation Steps:**
    1. Link: <https://github.com/asunm12912171/cookbook.git>
    2. Navigate to the client directory.
    3. Install dependencies using npm install.
    4. Configure environment variables in a .env file.
    5. Start the development server with npm start.
- 

## 5. Folder Structure

- **Client:**
  - src/components: Contains React components (e.g., Dashboard, WorkoutLog).
  - src/pages: Contains page components for routing.
  - src/assets: Stores static assets like images and styles.
  - src/redux: Contains Redux store, actions, and reducers.
  - src/utils: Utility functions and custom hooks.
- **Utilities:**
  - useFetch: Custom hook for API requests.
  - formatDate: Utility for formatting dates.

- `calculateCalories`: Helper function for calorie calculations.
- 

## 6. Running the Application

- **Frontend:**
    - Navigate to the client directory and run `npm start`.
    - The application will be available at <http://localhost:3000>.
- 

## 7. Component Documentation

- **Key Components:**
    - **Dashboard Component:** Displays daily fitness metrics.
    - **WorkoutLog Component:** Allows users to log workouts.
    - **CalorieTracker Component:** Tracks calorie intake and expenditure.
    - **ProgressChart Component:** Visualizes user progress.
  - **Reusable Components:**
    - **Button:** Customizable button component.
    - **InputField:** Reusable input field with validation.
    - **Modal:** Reusable modal for pop-ups or alerts.
- 

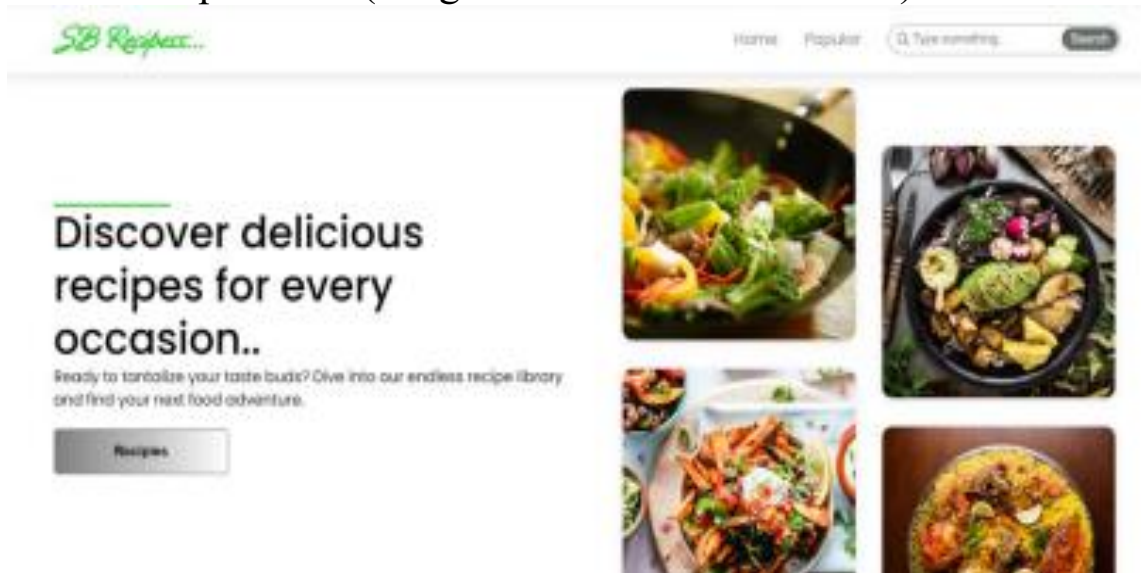
## 8. State Management

- **Global State:**
    - Managed by Redux (e.g., user authentication, workout data).
  - State flows from Redux store to components
  - **Local State:**
    - Managed within components using React's `useState` and `useEffect`.
-

## 9. User Interface

- **Screenshots:**

- Dashboard, Search Page, and Workout Log screenshots are provided (images not included in the text).



---

## 10. Styling

- **CSS Frameworks/Libraries:**
  - **Styled-Components:** Used for component-level styling.
  - **Bootstrap:** Used for responsive layouts and pre-built components.
- **Theming:** Custom theme with light and dark modes

---

## 11. Testing

- **Testing Strategy:**
  - **Unit Testing:** Jest and React Testing Library.
  - **Integration Testing:** Ensures components work together.
  - **End-to-End Testing:** Cypress for user flow testing.
- **Code Coverage:** 85% coverage using Jest.

---

## 12. Screenshots or Demo

- **DemoLink:**  
[https://drive.google.com/file/d/1RTM3obE6uhDTh9m5c2LQec\\_aSP1hrdkv4/view?usp=drivesdk](https://drive.google.com/file/d/1RTM3obE6uhDTh9m5c2LQec_aSP1hrdkv4/view?usp=drivesdk)
- **Screenshots:** Provided in the UI section.

---

## 13. Known Issues

1. Calorie tracker sometimes fails to update in real-time.
2. Progress chart may not render correctly on older browsers.
3. Mobile navigation menu occasionally overlaps with content on smaller screens.

## 14. Future Enhancements

- **New Features:**
  - Integration with wearable devices (e.g., Fitbit, Apple Watch).
  - Social features to share progress with friends.
  - Gamification (e.g., badges, rewards).
- **UI/UX Improvements:**
  - Add animations for a more engaging experience.
  - Improve mobile navigation menu.
- **Performance Optimization:**
  - Optimize chart rendering for low-end devices.
  - Implement lazy loading for components.

---

This documentation provides a comprehensive guide to the project, including its architecture, setup, components, state management, and future plans.