1/21/2025

Proof of Concept

PoC – Calorie Counter

Geethapriyan S
23CS049

Proof of Concept (PoC) Document for Calorie Counter

1. Project Writeup

The Calorie Counter is a web-based application designed to help users log their daily meals and calculate calorie intake efficiently. By providing an intuitive interface, the platform simplifies meal tracking and supports users in achieving their health and fitness goals. With built-in features such as meal logging, real-time calorie computation, and historical tracking, the app empowers users to make informed dietary decisions.

Key Features:

- **Meal Logging**: Users can input meals, select ingredients, and specify portion sizes.
- **Calorie Calculation**: Automatic calorie calculation based on user input and a preloaded nutritional database.
- **Personalized Goals**: Users can set daily calorie targets based on their fitness objectives.
- **Meal Insights**: Displays macro and micronutrient breakdowns.
- **User Authentication**: Secure user accounts to store and retrieve personalized data.

The Calorie Counter is designed to streamline the meal tracking process while providing valuable insights into users' dietary habits, helping them maintain a healthy lifestyle.

---

2. Components of the Project

Frontend

Framework: React.js

Description:

- Provides a dynamic and interactive user interface for logging meals and viewing calorie data.
- Displays calorie and nutritional information in an organized and visually appealing manner.

Libraries Used:

- **React Router**: For seamless navigation between pages such as the meal log, dashboard, and history views.
- **Chart.js**: Generates visualizations for trends in calorie consumption and nutrient breakdowns.

Backend

Framework: Node.js & Express

Description:

- Manages API endpoints for meal logging, calorie computation, and user data retrieval.
- Processes and validates user inputs.
- Integrates with the nutritional database for accurate calorie calculations.

APIs:

- **Meal Logging API**:
  - ✓ POST /log-meal: Logs a new meal entry.
  - ✓ GET /meals: Retrieves a user's logged meals.
- **Calorie Calculation API**:
  - ✓ GET /calories: Fetches calorie data based on the ingredients and portion sizes.
- **User Management API**:
  - ✓ POST /register: Handles user registration.
  - ✓ POST /login: Authenticates users.

User Authentication:

- **MongoDB**: Manages user sign-ups, logins, and secure session storage.

Database

Database: MongoDB

Description:

- **Collections:**
  - ✓ **Users:** Stores user profile information and preferences.
  - ✓ **Meals:** Contains logged meals, ingredients, and timestamps.
- **Key Features:**
  - ✓ Tracks user-specific data efficiently.

- ✓ Supports querying for historical calorie trends.

Hosting Platform

Hosting Services:

- **Frontend**: Hosted on Vercel for static deployment.
- **Backend**: Render for dynamic API management.
- **Database**: MongoDB Atlas for cloud-based database services.

---

3. Frontend Components

Components Overview:

1. **Home.jsx**:
   - ✓ Introduction to the application with options to log meals or view dashboard insights.
2. **MealLog.jsx**:
   - ✓ Interface for logging meals by selecting ingredients, portion sizes, and meal types (e.g., breakfast, lunch).
3. **Dashboard.jsx**:
   - ✓ Displays daily calorie consumption, macro breakdown, and progress towards daily goals.
4. **History.jsx**:
   - ✓ Visualizes historical calorie data with charts for trends analysis.
5. **LoginRegister.jsx**:
   - ✓ Handles user authentication for secure data access.

---

4. Backend Components

API Endpoints:

- **Authentication:**
  - ✓ /signup: Registers a new user and initializes their profile.
  - ✓ /login: Authenticates user credentials.

- **Meal Management:**
  - ✓ /log-meal: Logs a meal with associated details.
  - ✓ /meals: Retrieves logged meals for a specific user.
- **Nutritional Data:**
  - ✓ /calories: Computes calorie data for provided ingredients.
- **History:**
  - ✓ /history: Retrieves historical meal data for trends analysis.

Real-Time Features:

- Updates calorie calculations and dashboard metrics dynamically as users log meals.

---

5. Database Components

- **Users Collection:**
  - ✓ user_id: Unique identifier for each user.
  - ✓ email: User's email address.
  - ✓ password: Hashed password for secure authentication.
  - ✓ daily_goal: User-defined daily calorie target.
- **Meals Collection:**
  - ✓ meal_id: Unique identifier for each meal entry.
  - ✓ user_id: Links meal to a specific user.
  - ✓ meal_type: Type of meal (e.g., breakfast, lunch).
  - ✓ ingredients: List of ingredients with portion sizes.

---

6. Hosting Platforms

- **Frontend Hosting:**
  - ✓ Vercel for serving the static React application.
- **Backend Hosting:**
  - ✓ Heroku or AWS for scalable API hosting.
- **Database Hosting:**

- ✓ MongoDB Atlas for secure, cloud-based database management.

---

7. Flow Diagram of the Project

User Flow:

1. **Sign In/Register:** Users create an account or log in.
2. **Set Goals:** Users define their daily calorie targets.
3. **Log Meals:** Users log meals by selecting ingredients and portion sizes.
4. **View Dashboard:** Users monitor daily progress and macro breakdowns.
5. **Adjust Goals:** Users refine their targets based on progress.

Summary

The Calorie Counter is a robust and user-friendly platform designed to simplify meal tracking and calorie management. With real-time calculations, insightful analytics, and historical data tracking, it supports users in achieving their dietary goals. This PoC demonstrates the feasibility of implementing the app with modern web technologies and cloud-based services, ensuring scalability and user satisfaction.