

# Short Overview: Steps to Build the Calorie Tracker Project

## 1. Project Planning

- **Define Features:**
    - User Profiles: Age, weight, height, activity level for calculating calorie targets.
    - Calorie Logging: Users log food and calories manually.
    - Trends and Predictions: Show daily/weekly trends and forecast future calorie needs.
- 

## 2. Technology Stack

- **Frontend:** React.js (UI development with interactive charts and forms).
  - **Backend:** Node.js (Express) or Python (Flask/Django) for APIs and logic.
  - **Database:** SQLite or PostgreSQL to store user and food log data.
  - **Data Analysis:** Python with Pandas, NumPy, Matplotlib, and Prophet for trends and forecasting.
  - **APIs:** Edamam or Nutritionix for automated calorie lookups.
  - **Deployment:** React on Vercel/Netlify, Backend on Heroku/AWS, Database on Heroku Postgres.
- 

## 3. Steps to Develop

1. **Frontend Development (React):**
  - Design user interface: Pages for profile setup, logging food, and viewing trends.
  - Use `axios` for API calls to backend.
  - Integrate Chart.js or Recharts for visualizations.
2. **Backend Development:**
  - Build RESTful APIs for user profiles, calorie logging, trends, and predictions.
  - Use Python scripts for time series forecasting and integrate them with backend endpoints.
3. **Database Setup:**
  - Create tables for users, food logs, and predictions.
  - Use SQLite for local development and PostgreSQL for deployment.
4. **Data Analysis:**
  - Implement scripts to calculate trends and forecast using ARIMA/Prophet.
  - Visualize trends using Matplotlib/Seaborn.
5. **API Integration:**
  - Connect to Edamam/Nutritionix APIs for automated food calorie lookup.
  - Fetch and store calorie details in the database.

## 6. Deployment:

- Deploy the frontend on Vercel/Netlify.
  - Deploy the backend and database on Heroku/AWS.
- 

## 4. Features Overview

- **Basic Features:**
    - User Profiles: Personal details for BMR-based calorie targets.
    - Manual Calorie Logging: Add foods and calories via form.
  - **Advanced Features:**
    - Food Database API: Auto-fetch calorie data for common foods.
    - Forecasting: Predict calorie trends for the next 7 days using time series analysis.
  - **Additional Features:**
    - Meal Suggestions: Recommend meals based on remaining calories.
    - Charts: Visualize trends using line/bar graphs.
    - Scaling Up: Add water tracking, macros, and exercise logs.
- 

## 5. Tools and Skills Required

- **Frontend:** React, Chart.js, Axios.
- **Backend:** Node.js (Express) or Python (Flask).
- **Data Analysis:** Python (Pandas, NumPy, Matplotlib, Prophet).
- **Database:** SQLite/PostgreSQL.
- **API Integration:** Edamam/Nutritionix.
- **Deployment:** Vercel/Netlify, Heroku.

# TOOLS

## Frontend Development (React):

- Use **VS Code**. It's lightweight, powerful, and widely used for React projects.
- Extensions like Prettier and React snippets can speed up your work.

## Backend Development (Node.js or Python):

- Stick to **VS Code** for backend coding, whether you're using Python (Flask/Django) or Node.js (Express).

## Data Analysis and Visualization (Python):

- Start with **Jupyter Notebook** for prototyping your data analysis scripts (e.g., trends, ARIMA/Prophet models).
- Once your analysis logic is ready, integrate it into the backend using VS Code.

## Testing APIs:

- Use **Postman** for testing backend endpoints during development.

## Database:

- If you use **SQLite**, manage it with **DB Browser for SQLite** for a simple GUI to view and debug your database.
- If you use **PostgreSQL**, **pgAdmin** is great for managing it but optional during initial development.

## Deployment:

- Use **Vercel** or **Netlify** for deploying your React frontend.
- Use **Heroku** or **Railway** for deploying your backend and database.

# FIGMA COMPONENTS DETAILS

For a basic structure, you can focus on the following essential pages:

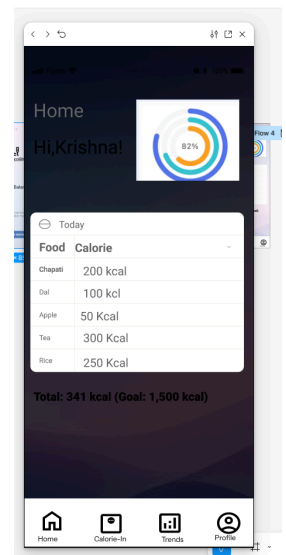
1. **Home Page**
  - Login/Signup options
  - Brief intro or CTA to start tracking calories
2. **User Profile Page**
  - Form to input basic details (age, weight, height, activity level) for calorie target calculation
3. **Calorie Logging Page**
  - Simple form to log food items and calories manually or via API lookup
4. **Trends Page**
  - Basic visualization of daily/weekly calorie intake trends (graphs)
5. **Settings Page** (Optional)
  - Basic options to update profile or log out

# Detailed Analysis of every module in Project

## Home Page

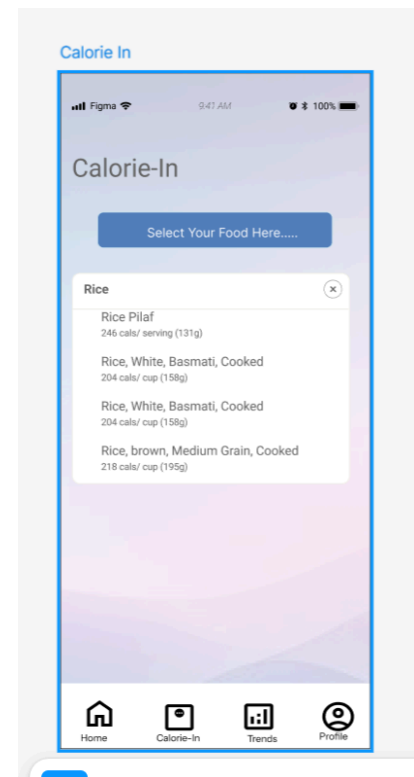
The homepage of the project must have

1. A card must be displaying all my consumed food with their calories of Today.
2. Total calories consumed must be shown in the home page
3. Insert a progress tracker- Show a progress bar at the top of the page, indicating how many calories have been consumed vs. the daily goal.
4. The progress bar should update dynamically as the user logs new food entries.
5. The display should be clear and concise, with easy-to-read font sizes and colors.



## Calorie-In Page

1. There is a button saying “ select your food “
2. After clicking on the button there must be a search bar to search For the food.
3. The data to be searched should come from an API and from the data we select food.
4. The selected data must be saved in two pages
  1. It must be shown in home page
  2. The selected data must be saved in the backend of trends pageWhere we will perform analysis on the selected data



## Trends Page

1. There is a button of showing trends , it has drop down to select which analysis u want to see, either TODAY , THIS WEEK,THIS MONTH
2. After user selects what analysis user wants to see , it can see the charts.
3. User can also download the charts.



## Profile Page

The **Profile Page** provides user information.. The page includes fields such as the user's name, ID (username or handle), email address, phone number, current weight, height, and target weight.

There are two buttons.

- **Edit:** Allow user to update weight height and target weight..
- **Log Out:** Provides a secure option for users to log out of their accounts

## Here is the workflow of the profile page

### Profile Page:

- Displays user details such as weight, height, and target weight.
- Includes an **Edit** button for users to update their information, which navigates to the **Detail Page**.

### {sub page } Detail Page:

- Allows users to input or modify their height, weight, gender, and date of birth.
- Provides two options:
  - **Calculate BMI and Weight:** Redirects to the **BMI Calculation Page** with results.
  - **Skip Calculation:** Proceeds without BMI analysis.

### BMI Calculation Page:

- Displays the user's BMI, weight status (e.g., overweight), and overweight value (if applicable).
- Offers an option to proceed by setting a target weight and timeline for achieving fitness goals.

