Amrita Vishwa Vidyapeetham
Centre for Excellence in Computational Engineering and Networking
Amrita School of AI, Coimbatore

# Bite2Burn: Nutritional Insight and Fitness Tracking Application

**Prepared By Team 11:**

J. Ajay Surya – CB.EN.U4AIE21018
V. Mohitha – CB.EN.U4AIE21032
M. Prasanna Teja – CB.EN.U4AIE21035
P. Sai Ravula – CB.EN.U4AIE21041

**Supervised by:**
Dr. Sachin Kumar S

An End Semester Project submitted to the CEN department as part of course evaluations of **"21AIE457 –Full Stack Development"** for B. tech in **Artificial Intelligence Engineering**.

# Table of Contents

## Introduction

In an era where health and fitness have become integral to modern lifestyles, the ability to make informed dietary decisions plays a crucial role in achieving personal wellness goals. However, navigating the complexities of calorie counting, meal planning, and nutritional intake can be overwhelming, especially in the context of diverse cuisines such as Indian food, which is rich in variety and cultural significance. While calorie estimation apps exist, most fail to account for the unique composition of Indian foods and often lack personalization tailored to individual health profiles.

To address these gaps, **Bite2Burn** aims to revolutionize the way individuals approach their dietary habits. Powered by advanced AI techniques, the application leverages the **Indian Food Composition Tables (IFCT)** published by the National Institute of Nutrition. This resource ensures that calorie information is precise, culturally relevant, and rooted in scientific data. Bite2Burn goes beyond simple calorie tracking, offering users **personalized meal suggestions, calorie consumption targets, and tailored exercise plans**, fostering a holistic approach to health management.

## Problem Statement

This project focuses on developing **Bite2Burn**: An AI powered application that provides accurate calorie counting, personalized suggestions on meals, and targeted nutritional guidance for improved health outcomes.

## Motivation

The motivation behind developing Bite2Burn stems from several factors:

- **Addressing Health Challenges:**
  With rising cases of obesity, diabetes, and cardiovascular diseases, effective dietary management is more important than ever. A significant contributor to these health challenges is the lack of awareness regarding calorie intake and its balance with physical activity.

- **Cultural Relevance:**
  Existing calorie estimation tools are often Western-centric and fail to account for the nuanced diversity of Indian cuisine, which includes unique preparation methods, ingredients, and portion sizes. This gap necessitates a solution specifically tailored to the Indian dietary context.

- **Empowering Personalization:**
  Each individual has unique nutritional needs based on factors like age, gender, activity level, and health goals. Bite2Burn addresses this by providing **personalized recommendations** that empower users to take charge of their health in a way that aligns with their lifestyle and personality.

# Challenges

1) Inaccurate Self-Reported Data: Users often underestimate or overestimate their calorie intake, leading to inaccurate tracking and ineffective weight management strategies.
2) Lack of Personalization: Generic dietary advice fails to account for individual needs, preferences, and dietary restrictions, reducing user engagement and limiting the effectiveness of recommendations.

# Objective

Develop an AI-powered system for accurate calorie estimation: This system will utilize the Indian Composition Food Tables pdf published by the National Institute of Nutrition, for providing accurate calorie content information about general foods. It will also provide users with personalized suggestion on how many calories to consume based on their personality, and how to burn them too!

# Dataset Description

## Indian Food Composition Table (IFCT)

- Indian Food Composition Tables (IFCT) 2017 is the major source of food composition data in India, generated, developed, managed and maintained by the National institute of Nutrition (ICMR).
- The uniqueness of the IFCT is that all nutrient values presented here have been derived from comprehensive national food sampling followed by analysis and contains only analytical data.
- Foods with common characteristics have been placed together and arranged in groups (Table1).

- All foods have been categorized into 20 food groups and the number in parenthesis indicates the total number of foods present in each group.
- A total of 528 foods have been analyzed for more than 150 parameters and

**Table 1. Food groups in the IFCT.**

| Code | Food groups | No. of food entries |
|---|---|---|
| A | Cereals and Millets | 24 |
| B | Grain Legumes | 25 |
| C | Green Leafy Vegetables | 34 |
| D | Other Vegetables | 78 |
| E | Fruits | 68 |
| F | Roots and Tubers | 19 |
| G | Condiments and Spices | 33 |
| H | Nuts and Oil Seeds | 21 |
| I | Sugars | 2 |
| J | Mushrooms | 4 |
| K | Miscellaneous Foods | 2 |
| L | Milk and Milk Products | 4 |
| M | Egg and Egg Products | 15 |
| N | Poultry | 19 |
| O | Animal Meat | 63 |
| P | Marine Fish | 92 |
| Q | Marine Shellfish | 8 |
| R | Marine Mollusks | 7 |
| S | Fresh Water Fish and Shellfish | 10 |
| T | Edible Oils and Fats | 9 |

presented under different nutrient component parameter.

## Description of Table1

- The table 1 consists of food items falling into the category of Cereals and Millets with food code starting with A.
- The alphabetic food character indicates the food group and the numeric character(s) represent a record position within a category.
- The values of six separate regions namely East, Northeast, North, West, South and Central India obtained from compositing adequate number of similar foods collected from within the region was tabulated.
- Food energy is expressed in Kilo joules (KJ)

## Additional Information

All foods are expressed per 100g edible portion. Generally, the values have been expressed to a constant number of decimal places for each nutrient.

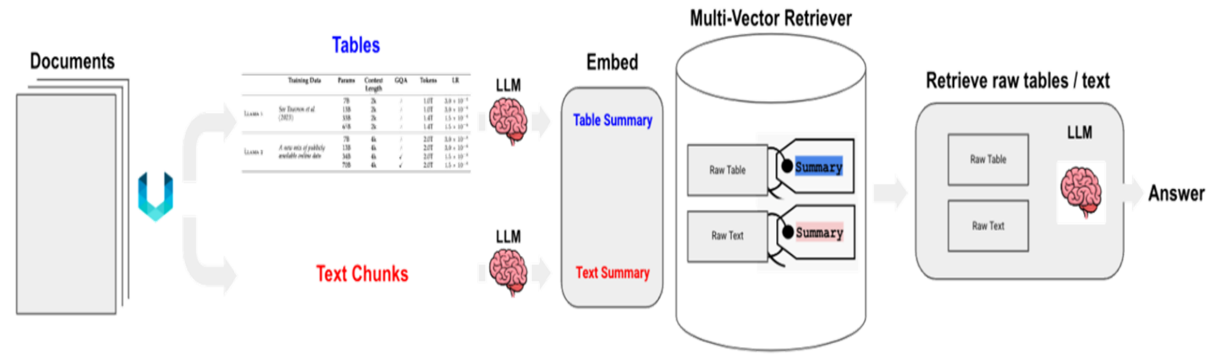
All blank spaces in the nutrient tables represents below detectable limit of the particular method used for analysis.

Regional foods may have a different nutrient profile that may differ from national average and hence individuals seeking local specific nutrient data are encouraged to obtain region specific data from NIN.

| Food Code | Food Name | No.of Regions | Moisture | Protein | Ash | Total Fat | Dietary Fibre | | | Carbohydrate | Energy (KJ) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | Total | Insoluble | Soluble | | |
| **A** | **CEREALS AND MILLETS** | | | | | | | | | | |
| A001 | **Amaranth seed, black** *(Amaranthus cruentus)* | 1 | 9.89 | 14.59 | 2.78 | 5.74 | 7.02 | 5.76 | 1.26 | 59.98 | 1490 |
| *A002* | **Amaranth seed, pale brown (Amaranthus)** | 6 | 9.200004 | 13.2700034 | 3.05±0.30 | 5.56±0.33 | 7.47±0.09 | 5.80±0.17 | 1.67±0.21 | 61.46±0.60 | 1489±10 |
| A003 | **Bajra** *(Pennisetum typhoideum)* | 6 | 8.970006 | 10.9600026 | 1.37±0.17 | 5.43±0.64 | 11.49±0.62 | 9.14±0.58 | 2.34±0.42 | 61.78±0.85 | 1456±18 |
| A004 | **Barley** *(Hordeum vulgare)* | 6 | 9.7700038 | 10.9400051 | 1.06±0.22 | 1.30±0.20 | 15.64±0.64 | 9.98±0.62 | 5.66±0.68 | 61.29±0.77 | 1321±19 |
| A005 | **Jowar** *(Sorghum vulgare)* | 6 | 9.0100077 | 9.9700043 | 1.39±0.34 | 1.73±0.31 | 10.22±0.49 | 8.49±0.40 | 1.73±0.40 | 67.68±1.03 | 1398±13 |
| A006 | **Maize, dry** *(Zea mays)* | 6 | 9.2600055 | 8.8000049 | 1.17±0.16 | 3.77±0.48 | 12.24±0.93 | 11.29±0.85 | 0.94±0.18 | 64.77±1.58 | 1398±25 |
| A007 | **Maize, tender, local** *(Zea mays)* | 6 | 68.2900052 | 3.5700042 | 0.38±0.04 | 1.40±0.30 | 3.67±0.26 | 3.23±0.23 | 0.43±0.07 | 22.69±0.94 | 502±7 |
| A008 | **Maize, tender, sweet** *(Zea mays)* | 4 | 74.4000071 | 4.1600041 | 0.36±0.06 | 1.35±0.07 | 3.30±0.51 | 2.71±0.53 | 0.59±0.11 | 16.42±0.89 | 405±14 |
| A009 | **Quinoa** *(Chenopodium quinoa)* | 1 | 10.43 | 13.11 | 2.65 | 5.50 | 14.66 | 10.21 | 4.46 | 53.65 | 1374 |
| A010 | **Ragi** *(Eleusine coracana)* | 5 | 10.8900061 | 7.1600063 | 2.04±0.34 | 1.92±0.14 | 11.18±1.14 | 9.51±0.65 | 1.67±0.55 | 66.82±0.73 | 1342±10 |

# Methodology

## Backend Architecture




## A Detailed Report

The system leverages a combination of Large Language Models (LLMs), vector databases, and a Retrieval Augmented Generation (RAG) pipeline to provide accurate and efficient responses.

### I. Data Ingestion and Preprocessing:

- **Input Data**: The system begins with a semi-structured Indian food composition table PDF as input. This PDF contains nutritional information organized in tabular format.
- **PDF Chunking**: The initial step involves splitting the large PDF document into smaller, manageable text chunks. This improves processing efficiency and allows for parallel processing of individual chunks.
- **Text and Table Extraction**: The chunked data is then processed to identify and separate textual content from tabular data. This stage involves Optical Character Recognition (OCR) to convert scanned pages into searchable text, followed by YOLO algorithm to detect and delineate tables based on formatting cues (e.g., rows, columns, borders)
- **Data Categorization**: The extracted data is categorized into two streams: text chunks and tables. This separation is crucial because different processing methods will be applied to each.

## II. Summary Generation:

- **LLM Summarization:** The Mistral 7B Instruct model (Hugging Face transformer model) is employed to generate summaries for both the extracted text chunks and tables. This process condenses the information, making it easier for subsequent stages to process.
- **Summary Linking:** The generated summaries are linked back to their corresponding raw text chunks and tables. This crucial step maintains the relationship between the original data and its condensed representation. A unique identifier system is likely in place to ensure proper mapping.

## III. Embedding and Vector Database Storage:

- **Vector Embedding Generation:** The raw text chunks, tables, and their corresponding summaries are converted into vector embeddings using the FastEmbed library. Vector embeddings represent textual data as numerical vectors in a high-dimensional space, where semantically similar data points are closer together.
- **ChromaDB Storage:** The generated vector embeddings are stored in ChromaDB, a multi-vector retrieval database. This database allows for efficient similarity search, enabling fast retrieval of relevant information based on user queries.

## IV. Query Processing and Answer Generation:

- **Query Embedding:** When a user submits a query, it's first converted into a vector embedding using the same FastEmbed library used in the previous step, ensuring consistent representation.
- **Similarity Search:** The query embedding is compared against the vector embeddings stored in ChromaDB. The database efficiently returns the most similar embeddings (i.e., the text chunks, tables, and summaries that are most relevant to the query) based on calculated similarity scores (cosine similarity). This is the core of the RAG pipeline.
- **Context Retrieval:** The top-scoring embeddings (along with their associated raw text and tables) are retrieved and provided as context to the next stage.
- **Llama 3.1 Response Generation:** Finally, the retrieved context is passed to the Llama 3.1 LLM (another Hugging Face model) to generate a concise and accurate answer to the user's query. The LLM uses the contextual information to formulate a coherent and informative response.

## Technology Stack

### I. Large Language Models (LLMs):

- **Mistral 7B Instruct (Hugging Face)**: This powerful LLM is used for generating concise and informative summaries of both textual and tabular nutritional data extracted from the input PDFs. Its instruction-following capabilities ensure high-quality summaries tailored to the context.
- **Llama 3.1 (ollama):** This LLM serves as the core of the question-answering system. It takes the user's query and the retrieved context (raw text, tables, and summaries) to formulate accurate and comprehensive answers. The choice of Llama 3.1 likely reflects its strengths in generating natural and informative text.

### II. Vector Database:

- **ChromaDB**: This is a crucial component for efficient similarity search. ChromaDB stores the vector embeddings generated from the nutritional data, enabling rapid retrieval of the most relevant context for each user query. We chose ChromaDB as it is a scalable and performant vector database suited for the application's needs.

### III. Embedding Generation:

- **FastEmbed**: This library is responsible for converting the raw textual and tabular data (along with generated summaries) into vector embeddings. FastEmbed's efficiency is critical for timely processing of large datasets.

### IV. Document Processing:

- **Unstructured**: This library plays a key role in handling the initial semi-structured data (the nutritional PDFs). Unstructured is used for the extraction of both textual content and tabular data from the input PDFs. This choice reflects the need for a robust library capable of handling variations in PDF formatting.

### V. Programming Language:

- **Python**: Python is the selected programming language for this project. This choice is common for machine learning projects due to the extensive availability of libraries like LangChain, Unstructured, and FastEmbed.

## Frontend Architecture

### I. Overall Structure:

The frontend is structured as a single-page application (SPA) built with React. This approach ensures a smooth and responsive user experience. The application is divided into several key components:

- **Navigation Bar**: A fixed top navigation bar provides access to core functionalities: Calorie Calculator, Steps2Calories calculator, and a chat interface for interaction with the AI-powered nutrition guide.
- **Chat Interface**: The central component of the application is the chat interface, where users can interact with the AI using natural language. It displays the conversation history, offers suggestions, and shows a loading indicator during query processing.
- **Calorie Calculator**: This component provides a dedicated tool for users to calculate their daily calorie needs based on demographic information (age, sex, height, weight) and activity levels.
- **Steps to Calories Calculator**: This component allows users to estimate calorie expenditure based on the number of steps taken, weight, height, and walking pace. The result provides a detailed breakdown of calorie burned per step, total calories burned, and an estimate of the time spent walking.
- **React:** This JavaScript library forms the foundation of the frontend application, handling user interface rendering, component management, and state updates.
- **Lucide-React:** This library provides the icons used in the application, including the flame, send button, menu icon, calculator icon, and footprints icon.
- **axios:** This library is used for asynchronous communication with the backend API. It handles sending user queries and receiving responses from the server.
- **recharts:** This charting library is used specifically within the Calorie Calculator to render the interactive pie chart illustrating macronutrient distribution.

### II. Styling and Design:

The application uses a consistent color scheme primarily focused on purples and whites. This scheme contributes to a visually appealing and modern look. The layout uses Tailwind CSS for responsive design and efficient styling.

## Frontend – Backend Integration

### I. Backend (Flask) API Endpoint:

The Flask backend exposes a single API endpoint, /query, to handle requests from the frontend. This endpoint is the entry point for all user queries.

- **Request Reception:** The /query endpoint receives POST requests from the frontend. The request body includes the user's query as a JSON payload.
- **Query Processing:** The received query is processed using the LLM + RAG pipeline.
- **Response Transmission:** The generated response from the LLM is formatted as a JSON object (likely containing an answer key) and sent back to the frontend as an HTTP response.

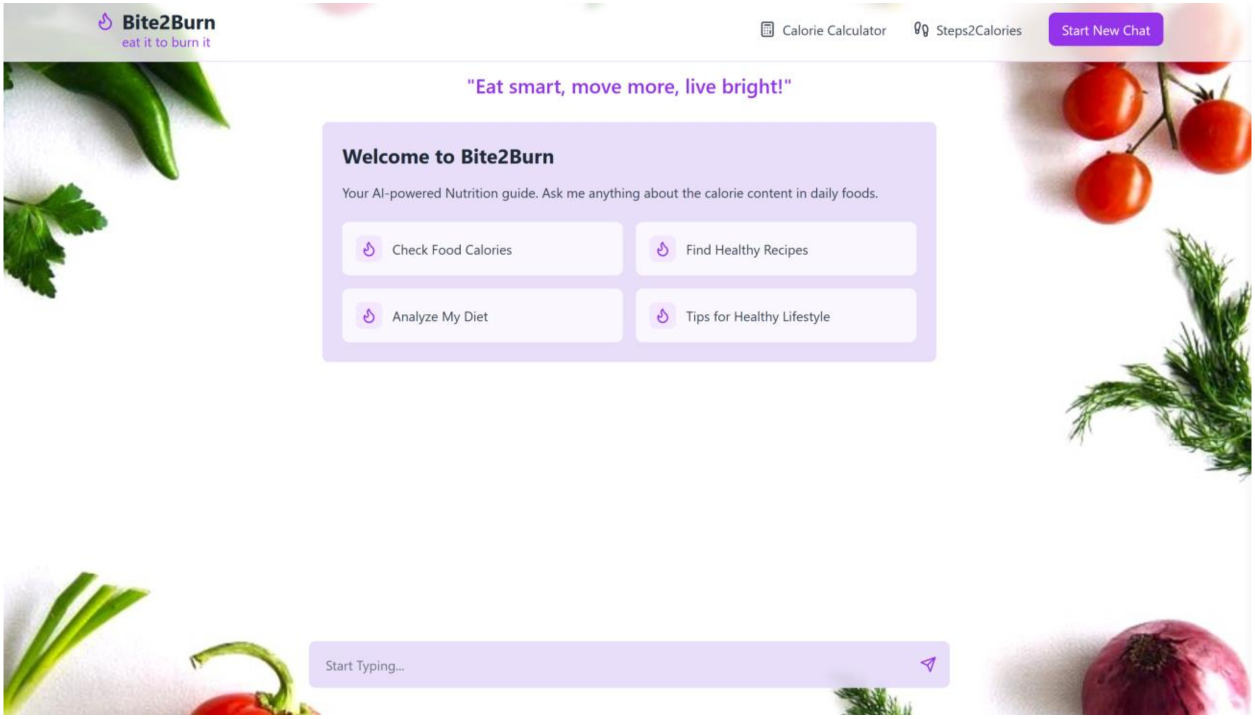
### II. Frontend (React) Communication:

The React frontend utilizes the Axios library to communicate with the Flask backend:

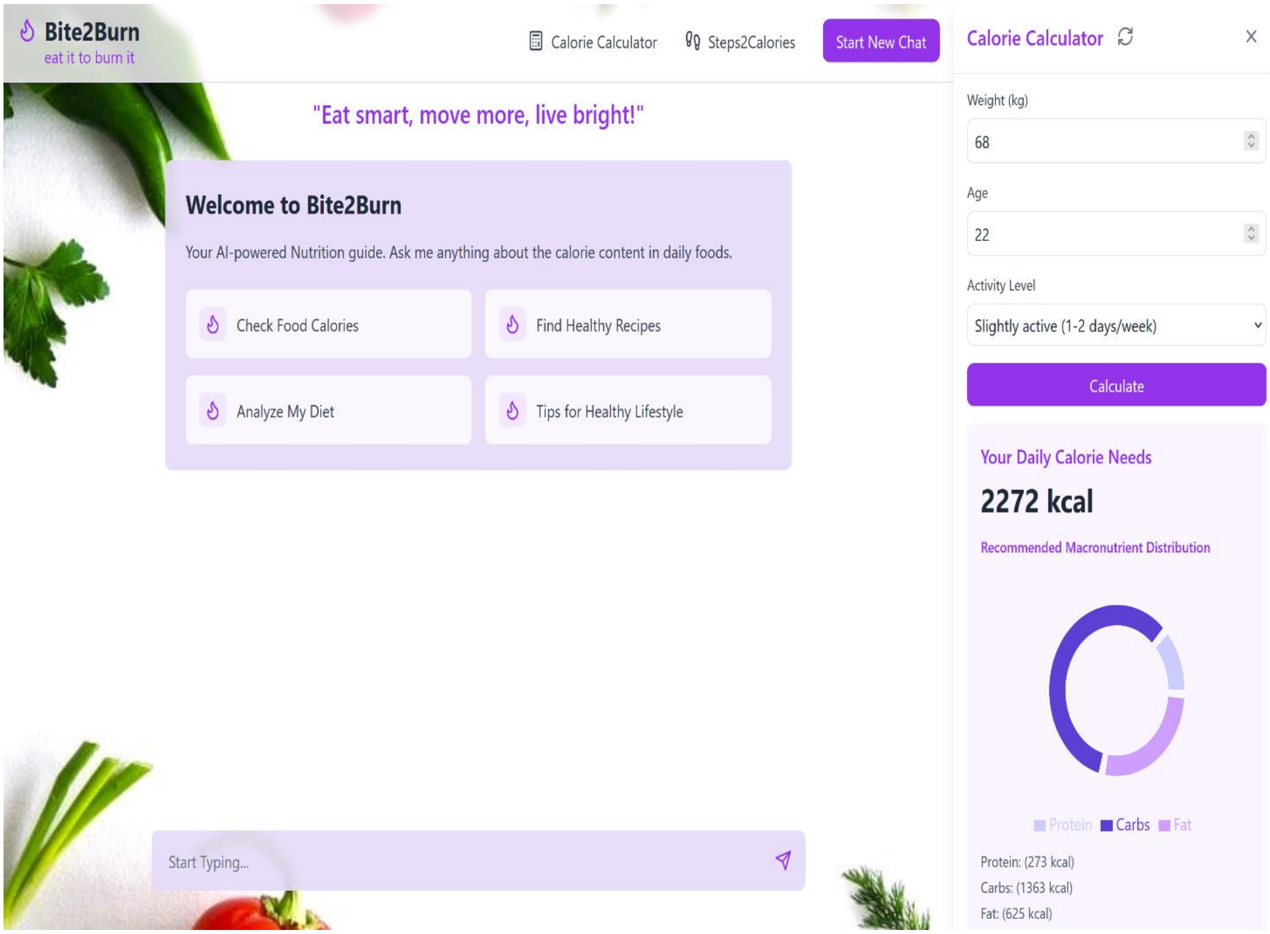
- **Request Initiation:** When a user submits a query through the chat interface, the React component triggers an Axios POST request.
- **Request Payload:** The POST request includes the user's query in the request body, similar to the JSON format expected by the /query endpoint.
- **Response Handling:** The Axios request returns a promise. The React component handles the promise:
  - **Success**: If the request is successful, the received JSON response (containing the LLM's answer) is processed and displayed in the chat interface.
  - **Error:** In case of an error, an appropriate message (such as "Sorry, I encountered an error...") is presented to the user.

# Application Overview

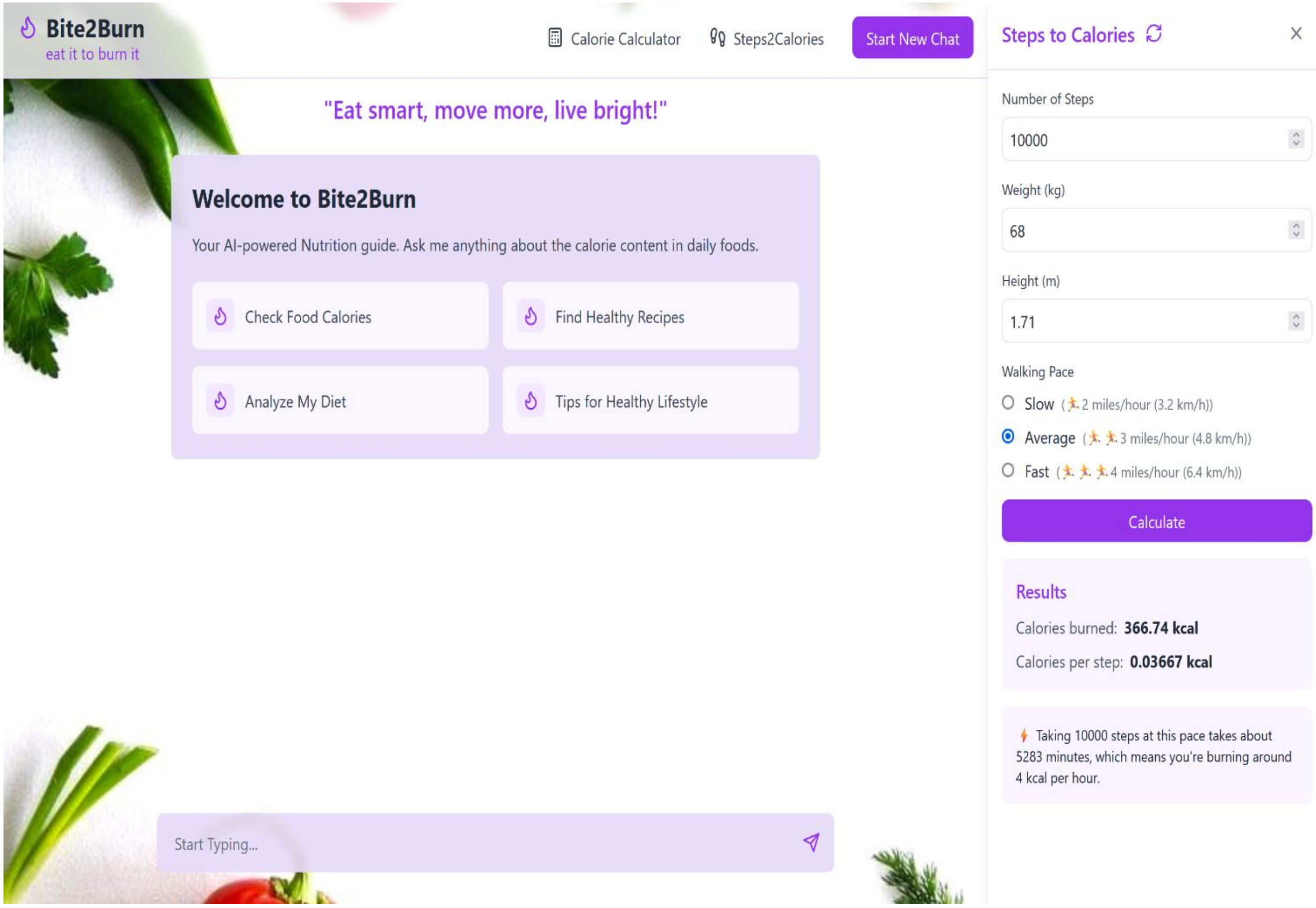
## Start page



## Calorie Calculator



## Steps to Calories Calculator



# Conclusion & Future Works

Bite2Burn is poised to transform the way individuals approach their health and nutrition, particularly in the Indian context. By leveraging AI and the **Indian Food Composition Tables (IFCT)**, the application ensures precise calorie estimation and provides personalized dietary recommendations tailored to users' unique health profiles. Its dual focus on calorie consumption and expenditure fosters a comprehensive understanding of nutrition and fitness, making healthy living accessible and achievable for users. This project demonstrates how technology can bridge cultural and informational gaps to empower individuals with actionable insights for improved health outcomes.

While Bite2Burn lays a strong foundation, there are several avenues for enhancing the system in the future these include: Expansion of Dataset, Incorporating Additional Health Metrics, including all the tables in IFCT for application, Advanced Personalization including giving recommendations to diabetic and heart patients on what foods to consume.