Personalized Health Assistant Using NLP and Data Visualization for Fitness Enthusiasts

A Project Report Submitted In Partial Fulfillment of the Requirments for the Degree of

Bachelor of Technology
in
Computer Science & Engineering

by

Harsh Vardhan (Roll No. 2100520100129)
Suyash Tripathi (Roll No. 2100520100159)
Vivek Kumar (Roll No. 2100520100163)

Under the Guidance of
Dr. Natthan Singh
Er. Sandeep Yadav

Department of Computer Science and Engineering
Institute of Engineering & Technology
Dr. APJ Abdul Kalam Technical University, Uttar Pradesh

June, 2025

Declaration

We hereby declare that this submission of project synopsis is our own work and that to the best of our knowledge and belief it contains no material previously published or written by another person or material which to a substantial extent has been accepted for award of any other degree of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

This synopsis report has not been submitted by us to any other institute for requirement of any other degree.

Harsh Vardhan (2100520100129)

Suyash Tripathi (2100520100159)

Vivek Kumar (2100520100163)

Certificate

This is to certify that the project report entitled: **Personalized Health Assistant Using NPL and Data Visualization for Fitness Enthusiasts** submitted by **Harsh Vardhan, Suyash Tripathi, Vivek Kumar** in partial fulfillment of the award of the Bachelor degree in Computer Science & Engineering is a record of the bonafide work carried out by them under our supervision and guidance at the Department of Computer Science & Engineering at Institute of Engineering & Technology Lucknow.

The project report has reached the standards that meet the requirements of the regulations related to the award of the degree.

It is also certified that this work has not been submitted anywhere else for the award of any other degree to the best of my knowledge.

Dr. Natthan Singh

Department of Computer Science and Engineering,
Institute of Engineering and Technology Lucknow

Er. Sandeep Yadav

Department of Computer Science and Engineering,
Institute of Engineering and Technology Lucknow

Acknowledgement

We would like to express our heartfelt gratitude to our supervisor, **Dr. Natthan Singh**, for her constructive feedback, valuable suggestions, and constant motivation, which have been essential in refining our approach to this project.

We also extend our sincere thanks to our supervisor and project coordinator, **Er. Sandeep Yadav**, for his invaluable guidance, support, and encouragement throughout the course of this project. His insights and expertise have been instrumental in shaping the direction of this work.

We would like to acknowledge the Project Evaluation Committee Members for their valuable inputs and guidance in the early stages of this project, which have greatly helped in defining its scope and objectives.

We are deeply thankful to **Dr. Girish Chandra**, Head of the Department of Computer Science and Engineering, for providing the necessary resources and a conducive environment for pursuing this project.

Finally, we would like to extend our gratitude to anyone else who has supported or encouraged us during this journey, including our peers, family, and friends, for their encouragement and belief in the project's potential.

Harsh Vardhan

Suyash Tripathi

Vivek Kumar

Abstract

Maintaining a healthy lifestyle requires consistent tracking of dietary intake and physical activity, but traditional calorie-tracking applications often rely on manual data entry, making them time-consuming and inconvenient for users. This project introduces CalorieTracker, a Flask-based web application that simplifies this process by allowing users to log meals and exercises using natural language, either through typing or voice input.

The system utilizes Natural Language Processing (NLP) powered by the spaCy en_core_web_md model to identify food items, quantities, and physical activities from user input with high accuracy. It supports both standard and custom food entries, including user-defined composite meals, and automatically calculates calories and protein based on known or user-provided nutritional information. Additionally, the application features interactive visual reports that display calorie intake, calories burned, and protein trends on daily, weekly, and monthly scales.

The application is designed to run efficiently on CPU-only environments, delivering a smooth, real-time experience without the need for heavy computational resources. By combining the lightweight NLP, user-friendly interfaces, and meaningful visual analytics, CalorieTracker offers a practical, accessible solution for promoting healthier lifestyle habits. The project also sets the stage for future enhancements such as multilingual input, micronutrient tracking, and personalized meal recommendations.

Contents

List of Figures

List of Tables

1 Introduction

In today's rapidly evolving environment, maintaining a healthy lifestyle has become increasingly vital, yet it frequently presents challenges due to limited time and resources for accurately monitoring dietary habits and physical activities. With the rising incidence of lifestyle-related illnesses such as obesity, diabetes and cardiovascular diseases, there is an escalating demand for innovative solutions that can assist individuals in effectively managing their health. Our capstone project aims to address this demand by developing a health application that harnesses Natural Language Processing (NLP) to interpret natural language inquiries and transform them into precise calorie counts. This application will enable users to track their daily, weekly and monthly calorie intake and expenditure, facilitating the achievement of their health and fitness objectives.

Conventional methods of calorie tracking typically involve the manual logging of food items and physical activities into journals or applications, which can be both time-consuming and susceptible to inaccuracies. Additionally, users often encounter difficulties in understanding the nutritional value of various foods and the calories expended through different exercises. Our health application streamlines this process by allowing users to input their dietary and exercise information in natural language, replicating conversational descriptions. For instance, users can simply state, ”I ate two apples and a banana for breakfast” or ”I engaged in a 30-minute jog,” and the application will accurately interpret and record these entries.

The primary objective of our project is to leverage advanced NLP techniques to bridge the gap between natural language and the structured data necessary for calorie counting. By accomplishing this, we aim to deliver a seamless and user- friendly experience that motivates individuals to remain consistent with their health tracking efforts. The application will feature a comprehensive database of foods and exercises, allowing it to generate detailed reports on calorie consumption and expenditure. These reports will empower users to make informed choices about their diet and physical activities, ultimately fostering improved health outcomes.

To realize this vision, our project integrates cutting-edge NLP models capable of understanding and processing complex language inputs. These models will be trained on extensive datasets to ensure they can adeptly handle a wide array of queries with high accuracy. Moreover, the application will boast a user-friendly interface that simplifies data input and progress tracking for users. By merging NLP technology with health tracking, we aspire to create a tool that not only simplifies calorie counting but also enriches the overall user experience.

In conclusion, our project addresses a significant need in the health and wellness domain by developing an innovative solution that leverages the capabilities of NLP to enhance the accessibility and accuracy of calorie tracking. We believe that our health application will substantially impact users' ability to manage their health, equipping them with the information and motivation required to attain their fitness goals. The subsequent sections of this report will explore the specific objectives, methodology, implementation, challenges and outcomes of our project, providing a thorough overview of our approach and accomplishments.

Calorie tracking is a well-established method for weight management and enhanced nutrition awareness. Unfortunately, most current tools—both mobile applications and web services—restrict the user to choosing from static, vendor-maintained food databases. In reality, people tend to prepare homemade dishes, eat regional specialties, or buy brand-name products that are not included in these databases. When presented with novel or user-tailored meals, they are compelled to estimate or forgo logging in its entirety, which compromises data validity and long-term compliance.

Moreover, most systems only track total caloric consumption without tracking macronutrients individually. Protein, specifically, has a greater thermic effect than carbohydrates or fats, such that a larger percentage of protein calories are spent on digestion. In addition, protein builds muscle, satiety, and better body composition. By ignoring protein data, users forego a critical aspect of nutritional quality.

At the same time, the development of natural language processing (NLP) methods has shown a lot of potential to analyze free-text user entries. If an app is able to parse sentences like "I had a protein shake and a turkey sandwich for lunch" or "I biked for 20 minutes this morning" with a high degree of accuracy, it does away with much of the hassle of manual data entry. Users can simply interact with the system as they would naturally—without ever having to consider database keys or literal search strings. Therefore, the incorporation of NLP into calorie-tracking sites has the potential to revolutionize user experience, making logging not only faster but also more intuitive.

Last, behavioral science predicts that visual feedback—especially in the form of charts and graphs—increases user engagement and encourages long-term behavior change. Although understanding "I ate 2,400 kilocalories today" is useful, seeing daily or weekly patterns visually can be much more inspiring: users can literally see trends like weekend indulgence or weekday steadiness. Posting both calories and protein on a graph also assists in making sure that individuals are hitting macronutrient objectives instead of simply tracking total caloric intake.

1.1 Problem Statement

In spite of increasing popularity of fitness and nutrition, numerous individuals encounter serious hindrances in maintaining regular, precise monitoring of their food and activity data. The reasons lie mainly with:

Clunky Data Entry: Most macronutrient and calorie-tracking applications involve formatted input or searching, making it time-consuming and disinviting to use regularly.

Inability to Support Natural Language: Users tend to think and speak freely in language (e.g., ”I ate two boiled eggs and a banana”), yet most apps don't facilitate easy natural language logging of meals or activities.

Inflexible Food Databases: Users often eat home-cooked meals or local food that is not contained in typical food databases. The majority of tools do not support custom ingredients or user-created meals, and thus logging becomes incomplete or inaccurate.

Inadequate Visualization and Feedback: Current applications may not present penetrating and dynamic visualizations. Users require dynamic, responsive, and clear visual feedback (e.g., trend in calories, patterns in protein consumption) to remain encouraged and informed.

1.2 Objectives

The objective of this project is developing a simple, user-friendly web application that simplifies health tracking through natural language input. The system allows users to input food intake and physical activities through free-text queries, which are then dealt with by an NLP module before being matched against a nutritional database. It tracks key macronutrients, especially calories and protein, and enables users to create and manage custom ingredients and meals for personalized tracking. Additionally, the application provides detailed daily, weekly, and monthly reports, along with dynamic visualizations using charting libraries, to help users monitor their progress and make informed dietary and fitness decisions.

1.3 Summary of Project

This project introduces a calorie-monitoring web application that leverages Natural Language Processing (NLP) to simplify food and activity logging, driven by the growing need for intuitive health-tracking tools.

1.3.1 Chapter - 1 : Introduction

The project aims to develop a calorie-monitoring web application that uses NLP to convert natural language inputs into structured nutritional data, enabling users to seamlessly log their daily food intake and physical activities. The system addresses the limitations of existing tools by allowing free-text and voice-based logging, automatic tracking of calories and protein, support for custom meals, and dynamic data visualization.

1.3.2 Chapter - 2 : Literature Review

This section reviews prior work in applying NLP to health tracking, highlighting the effectiveness of transfer learning and transformer-based models. It explains the choice of spaCy's `en_core_web_md` model for its balanced accuracy and efficiency, making it suitable for responsive web applications.

1.3.3 Chapter - 3 : Methodology

The existing food tracking systems largely depend on manual input and lack advanced NLP capabilities. The proposed system enhances this by supporting both text and voice input, using spaCy for entity recognition to extract nutritional data, distinguishing between food and activity entries, and presenting the information through structured and customizable reports.

1.3.4 Chapter - 4 : Implementation

The system is built using Flask for the backend and SQLAlchemy ORM for database management. NLP processing is handled by spaCy's `en_core_web_md` model, while dynamic visual reports are generated using Chart.js, with voice input integrated through the Web Speech API.

1.3.5 Chapter - 5 : Result and Analysis

The application effectively parses natural language and voice inputs to log food and exercise data, recognizes user-defined meals and portion sizes, calculates calories and protein based on user profiles, and presents clear and visually engaging reports to enhance the user experience.

1.3.6 Chapter - 6 : Conclusion

The project successfully delivers a modular, user-friendly, and scalable food and activity tracking system powered by NLP. While it performs well in structured tracking, current limitations include handling vague inputs, a small database size, single-user SQLite limitations, and lack of multilingual and mobile-native support.

2 Literature Review

Many researchers have applied Natural Language Processing (NLP) techniques to automate and improve dietary logging, transforming free-text meal descriptions into structured nutrient data.[1] developed a voice-enabled meal logging system that accepts spoken meal descriptions (e.g., "I had oatmeal with strawberries and a protein smoothie") and uses NLP to parse ingredients, quantities, and contextual cues, mapping them to caloric and macronutrient values. Their prototype demonstrated a marked reduction in user burden compared to manual entry on mobile devices, achieving over 90% accuracy for common food items.[2] applied a combination of topic modeling and named-entity recognition (NER) to extract dietary patterns from 24-hour recall data, showing that unsupervised NLP algorithms can uncover latent dietary themes with minimal manual annotation.[3] further explored the use of contextualized word embeddings—derived from pre-trained transformer models—for classifying free-text food logs into standardized categories and assigning nutrient quality scores. Their results indicated that fine-tuned transformer embeddings outperformed rule-based and bag-of-words methods by 8–12% in top-1 classification accuracy, underscoring the value of modern NLP architectures in large-scale nutritional data processing.

Transfer learning has emerged as a crucial paradigm in contemporary NLP, allowing developers to leverage large pre-trained language models and adapt them to domain-specific tasks with comparatively little data.[4] surveyed various transfer learning strategies—such as feature-based adaptation and fine-tuning—and demonstrated that models pre-trained on massive general corpora (e.g., Wikipedia, Common Crawl) can be effectively fine-tuned for specialized applications, including clinical and dietary text parsing, using only a few thousand annotated examples.[5] introduced the Hugging Face transformers library, which provides seamless access to dozens of pre-trained models (e.g., BERT, RoBERTa, DistilBERT). They reported that fine-tuning DistilBERT on a food-label classification task yielded 98% accuracy with just 2,000 labeled examples, illustrating how transfer learning drastically reduces annotation costs. Similarly, [6] applied transfer learning to build a robust "FoodNER" system: they fine-tuned a BERT-based model on a small, domain-specific corpus of recipe texts (approx. 5,000 sentences) and achieved an F1-score of 0.89 for ingredient extraction, compared to 0.75 using a CRF baseline. These studies confirm that transfer learning not only accelerates development but also sets new benchmarks for accuracy in nutrition-related NLP tasks.

Despite the superior accuracy of transformer-based models, deploying them in real-time, CPU-bound applications (such as a Flask-based calorie tracker) poses practical challenges due to their high memory and compute demands. To address this, many practitioners opt for spaCy's mid-sized model, `en_core_web_md`. [7] introduced SciSpaCy—an extension of spaCy optimized for biomedical text—and showed that the base `en_core_web_md` model processes up to 250 tokens per second on a single CPU core while maintaining 90% of BERT's NER accuracy in domain-specific settings. Further benchmarks by Honnibal and Montani [8] confirmed that `en_core_web_md` achieves 220–240 tokens/second on standard consumer hardware, outperforming larger BERT variants in speed by a factor of 10× at only a 5–7% drop in entity recognition F1-score. Moreover, [9] compared spaCy's `en_core_web_md` against BERT-based alternatives for extracting dietary behaviors from social media posts, finding that spaCy's medium model maintained 0.87 F1 on NER tasks with an inference latency under 50 ms per sentence—well within acceptable bounds for web applications. These results demonstrate that `en_core_web_md` offers a practical compromise: it is lightweight enough for CPU deployment yet robust enough to handle the nuances of dietary terminology without the heavy resource footprint of full transformer models.

Table 1: Summary of Literature Review on NLP Techniques in Dietary Logging

S.No	Author(s)	Year	Methods Used	Findings
1	Korpusik et al.	2019	Voice-enabled NLP logging system	Achieved over 90% accuracy reduced user effort compared to manual entry
2	Choi and Kim	2022	Topic modeling and Named Entity Recognition (NER)	Extracted dietary patterns from free-text; minimal manual annotation required
3	Weber et al.	2022	Pre-trained transformer-based embeddings for classification	Improved classification accuracy by 8–12% over rule-based methods
4	Ruder et al.	2019	Survey of transfer learning techniques	Demonstrated that pre-trained models adapt well to specialized domains like dietary parsing
5	Wolf et al.	2020	Hugging Face transformers; fine-tuning DistilBERT	Achieved 98% accuracy on food-label classification with only 2,000 labeled examples
6	Krämer et al.	2021	Fine-tuned BERT on recipe text for FoodNER	Reached F1-score of 0.89, outperforming CRF baseline (0.75)
7	Neumann et al.	2019	SciSpaCy using en_core_web_md model	Processed 250 tokens/sec on CPU with 90% of BERT's NER accuracy
8	Honnibal and Montani	2021	Benchmarked spaCy's en_core_web_md vs BERT	Found 10× faster speed with only 5–7% drop in NER accuracy
9	Titova et al.	2023	Compared spaCy vs BERT for dietary NER on social media	spaCy's medium model achieved 0.87 F1 with <50ms latency per sentence

3 Methodology

This section describes the detailed methodology adopted for the development of the digit recognition web application. The methodology includes the approach for digit recognition, the technologies used, and the datasets sourced for training the models.

3.1 Existing Methodology

The existing methodology used for tracking health is more or less dependent on manual input. The users access the system by filling out personal details, consumed food, and exercise activities on a daily basis. Although flexibility and control are achieved through this approach, some limitations do accompany it that influence accuracy and user experience as well. Each part of the methodology is explained in detail in the following sections.

3.1.1 User Login & Profile Setup

Users start off by registering their account and establishing a profile. In doing so, they enter critical personal information like **age**, **weight**, **height**, and **gender**. The system employs these data to determine individualized recommendations, such as daily calorie needs and optimal macronutrient compositions. Users can also establish custom goals, for example, **weight loss**, **muscle gain**, or **status quo maintenance**. The profile is dynamic and can be changed as the user moves along or as lifestyle varies, so that the recommendations remain current.

3.1.2 Manual Logging of Food

Food intake is logged manually. Users look up food items from a **prelisted list** or **dropdown menu**. Having chosen a food item, the user enters the **portion size** and **how much was consumed**. The system subsequently records the food intake on a **daily food log**, computes and shows important nutritional values like **calories**, **protein**, **carbohydrate**, and **fats**. Although this approach enables users to monitor their diet with extreme precision, it is demanding and time-consuming.

3.1.3 Manual Logging of Exercise

Exercise is also manually logged. The user chooses an activity from a predetermined list (for instance, **walking**, **running**, **cycling**, or **swimming**). They then enter the **duration** or **distance** of the activity undertaken. Depending on this, the system estimates the **calories burnt** considering the user's profile information like **weight** and **gender**. This data is appended to the user's **daily exercise record**, giving an overall picture of calories burned for the day. The accuracy is greatly reliant, though, on the accuracy of the user's input.

3.1.4 Custom Foods or Meals

To support a broader range of eating habits, the system permits **custom foods** or **meals**. This option is especially beneficial for **home cooking** or foods that are not stored in the database. Users type in the **food name**, **calories**, and most important nutritional data (e.g., **protein**, **carbohydrates**, **fats**). The custom items are saved in the user's profile and can be rapidly chosen in subsequent logs. This adds convenience to users with distinct or recurring dietary regimens.

3.1.5 Reports

The system offers helpful **reports** based on information provided by users. Reports may be accessed for varying time periods — **daily**, **weekly**, or **monthly**. They often provide summaries of **calories taken**, **calories expended**, and **net calorie balance**. The reports can come in the form of **basic tables** or **plain text**, and users can review trends in dietary and activity habits. Yet, because these reports are based solely on user-entered data, their credibility relies on how thoroughly the user enters food and activity.

3.1.6 Limitations

Even with its functionality, the current methodology has numerous limitations:

- The **manual logging process** is repetitive and time-consuming, which might deter regular use.

- Users need to know well the **names of foods**, **portion sizes**, and **nutritional values** in order to log accurately.
- It is hard to log **home-prepared meals** precisely since many do not have exact nutritional information.
- The **data entry effort** raises the chance of omissions, which has a negative effect on report accuracy and progress tracking.

3.2 Proposed Methodology

The Proposed system is intended to improve the user experience and enhance the accuracy of health tracking through the use of Natural Language Processing (NLP) to interpret user inputs and create meaningful insights automatically. The system's logical architecture is shown in Figure 1, presenting the end-to-end data flow — from user input, through NLP processing, database query, calculation, and storage of results — to user-friendly report visualization. This section describes each part of the methodology to be proposed in detail.

3.2.1 User Input

Description: It starts with the user input regarding exercise or food intake.

Implementation: The system has a friendly user interface through which users are able to input their data via text-based queries or voice commands. Such a dual-input facility makes the system flexible and convenient, allowing users to log their physical and dietary intake naturally and in the way they speak. The interface is designed to capture phrasing and sentence constructs variations to accommodate various user expressions.

3.2.2 NLP Processing

Description: The system uses advanced Natural Language Processing (NLP) methods to understand the input from the user and pull the pertinent contextual details.

Implementation: The input is processed using advanced NLP models like SpaCy, NLTK, or Transformer-based models like BERT. The models tokenize, perform POS tagging, Named

Entity Recognition (NER), and syntactic parsing to extract primary entities like food names, quantity, type of activity, duration, and intensity. The aim is to map unstructured natural language to structured data for further processing.

3.2.3 Exercise-Related Decision

Description: The system checks if the input is related to exercise or food logging.

Implementation: A classification function, either based on a rule-based algorithm or a light-weight intent classification model, examines the extracted entities and actions and classifies the input accordingly. Inputs that refer to physical activity such as running or swimming are labeled as exercise-related; otherwise, the input is considered food-related.

3.2.4 Search Exercise in Database

Description: If the input is exercise-related, the system asks the exercise database for more information.

Implementation: The system utilizes the identified exercise name to access relevant information, like average calories burned, exercise intensity levels, and suggested duration. If no direct match exists, the system gives the user substitute recommendations or asks for clarification.

3.2.5 Calculate Burned Calories

Description: The estimated calories burned through the recorded exercise are computed by the system.

Implementation: There is a dynamic formula used that incorporates the exercise base calorie rate with the user's personal record information (age, weight, sex) and exercise-specific variables (duration, intensity). The calculated estimate of calorie burn is personalized to each person for improved accuracy.

3.2.6 Search Food in Database

Description: For food inputs, the system queries the food database to get comprehensive nutritional data.

Implementation: The system cross-references the names of foods determined by NLP with the food database entries. It returns caloric values, portion sizes, and macronutrient breakdown (protein, carbohydrates, fats). In case the food is ambiguous or unknown, the system prompts for clarification from the user or recommends alternatives.

3.2.7 Calculate Food Calories

Description: The system estimates total caloric intake from the food data and portion sizes input.

Implementation: Each food item's calorie value per unit is multiplied by the portion size or quantity input by the user. This allows accurate tracking of diet intake. The system also checks for differences in measurement units (grams, cups, pieces) to ensure consistency of data.

3.2.8 Store Results in Database

Description: All processed food and exercise data are retained in the user's own database for long-term monitoring.

Implementation: The system captures detailed entries such as exercise name, duration, intensity, calories burned, and food item names, portion sizes, quantities, and total calories consumed. This persistent storage allows complete tracking of health parameters over time.

3.2.9 User Report

Description: The system provides a customized report of the user's logged activity and dietary consumption.

Implementation: The report module sums stored data over a time period as defined by the user (daily, weekly, monthly), calculates the net calorie balance, and creates reports that are pleasing to the eye. These are the line graphs to view trends, pie charts for macronutrient content, and summary tables. The report also offers observations and criticism, allowing users to determine if diet and exercise targets are being reached.

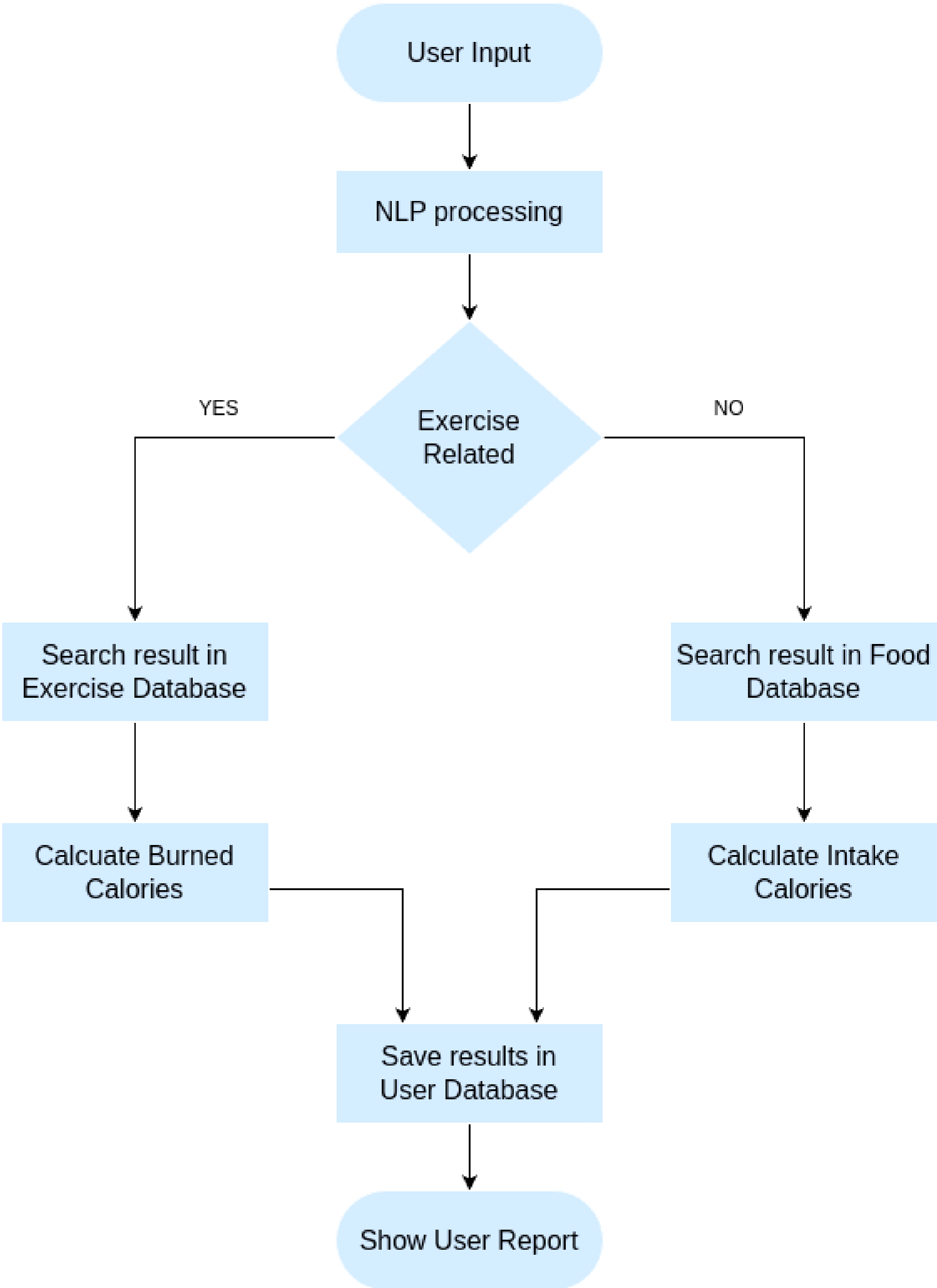

FIGURE 1: Calorie tracking system flow chart

3.2.10 Summary

The envisaged approach augments conventional healthcare monitoring systems using NLP-powered natural language interaction, machine learning-based data processing, and smart visualization. The goal is to enhance usability, minimize manual intervention, and maximize the accuracy of recorded data, enabling users to better meet their health and well-being targets.

4 Implementation

Implementation of the CalorieTracker system takes a well-defined modular approach following the Model-View-Controller (MVC) design pattern. This makes it scalable, maintainable, and performance-efficient by strictly separating concerns among data handling, user interface, and business logic. The implementation specifics of each primary component and their integration are discussed in the subsequent subsections.

4.1 System Architecture Overview

The CalorieTracker system is implemented with the MVC pattern to facilitate development and future upgrades:

4.1.1 Model (SQLAlchemy ORM):

The Model layer is responsible for all the logic dealing with data, controlling how the information is stored, retrieved, and updated in the system. With SQLAlchemy ORM, there is an object-oriented interface to the SQL database and thus raw SQL queries are abstracted. This model describes entities like Users, Food Entries, Exercise Logs, Custom Ingredients, and Composite Meals. Every entity is mapped to a database table, and relationships like one-to-many (one user may have many food items) and many-to-many (a dish consisting of several ingredients) are enforced to ensure data integrity and allow complex queries, as shown in the Entity-Relationship (ER) diagram, to ensure consistency and reliability of stored data.

4.1.2 View (Jinja2 + Bootstrap + Chart.js):

The View layer handles rendering the dynamic user interface. Jinja2 templates produce HTML pages styled using Bootstrap to deliver a contemporary and responsive layout flexible across devices. Interactive and visually stylish charts and graphs are provided with Chart.js integration, offering users a clear and intuitive view of their nutritional and exercise inputs.

4.1.3 Controller (Flask Routes):

The Controller is the system's backbone, taking control of all the interactions with the users. Flask routes accept HTTP requests, perform user authentication, process inputs, initiate NLP processing, interact with the database, and produce responses. This layer contains the business logic and encapsulates coordination between the Model and View layers.

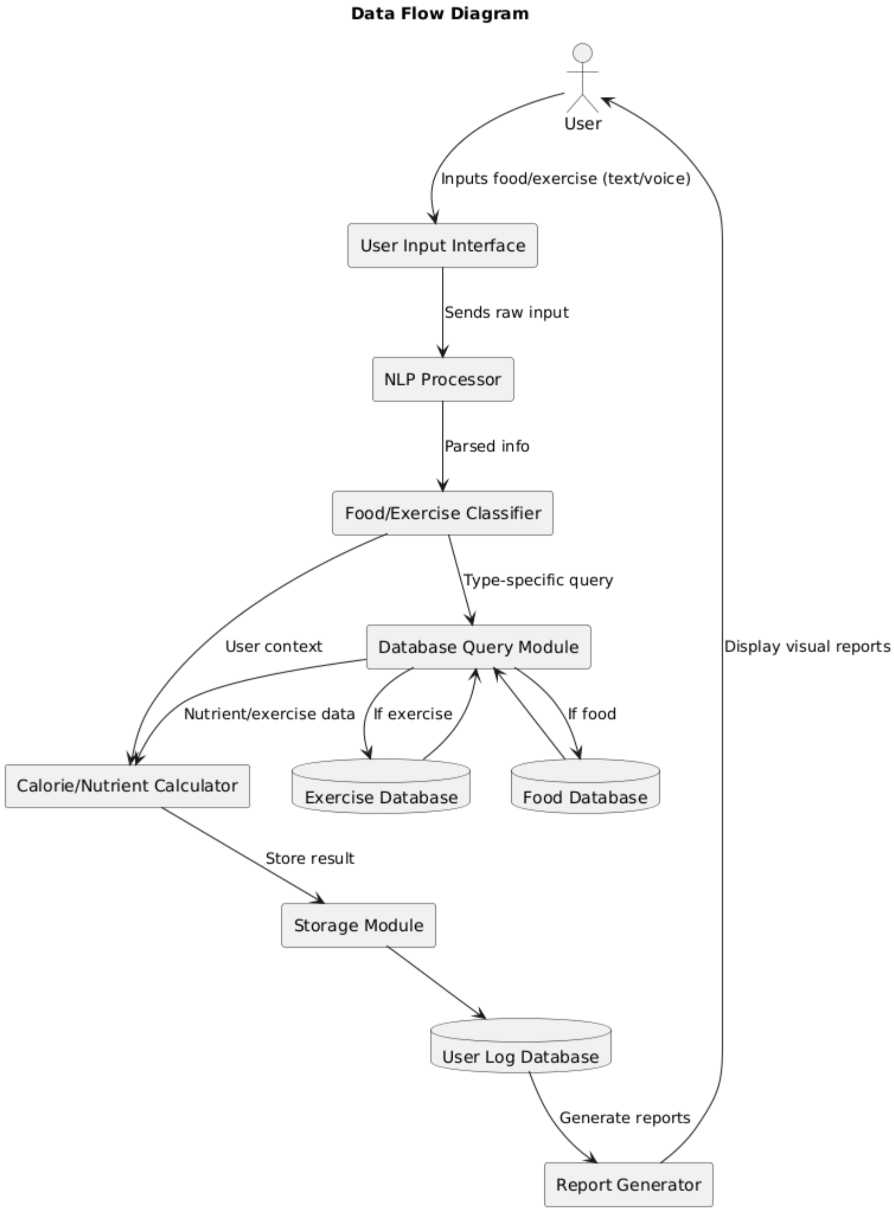

FIGURE 2: Data Flow Diagram

Figure 2 illustrates the data flow within the web application, highlighting the interaction between user input, processing components, and data storage systems.

4.2 User Authentication and Profile Management

User security and personalized tracking are ensured by using strong authentication and profile management mechanisms:

Authentication: Flask-Login is employed to securely control user sessions. It provides login state, session persistence, and safeguards user-specific information from unauthorized use. Passwords are secured with the Werkzeug security library, which implements strong hashing algorithms (like PBKDF2) to avoid plaintext storage and improve security.

Profile Data: Users create a comprehensive profile with fields such as name, email, age, gender, weight, and height. This individual information forms the basis for calculations like basal metabolic rate (BMR) and estimated calorie burn, and therefore makes monitoring more precise and customized to individual users. The profiles may be modified later with changes in weight or objectives.

4.3 Natural Language Input Handling

In order to make data entry easier and minimize friction, users can input in natural language form:

- **Text Input:** Users can enter descriptions of meals or exercise by typing into a normal textbox, for example, "I had a bowl of oatmeal and two eggs."
- **Voice Input:** Through the use of the Web Speech API, users have the option to say their input through the microphone, making it more accessible and convenient, especially for mobile or hands-free environments. The speech-to-text translation forwards a transcript to the backend to be processed.

The input is directly passed to the NLP engine to be thoroughly analyzed, saving time and effort and promoting more user engagement.

4.4 NLP Processing with spaCy

Natural Language Processing plays a key role in accurately interpreting user inputs:

- **Model Initialization:** The application initializes spaCy's `en_core_web_md` language model only once at application launch to save performance.
- **Custom Extensions:** Custom pattern recognizers and entity recognizers are added to recognize domain-related data like food names, quantities (numeric and fractional terms), exercises, and their durations.
- **Parsing Pipeline:** Inputs are tokenized, named entity recognized (NER), and dependency parsed to identify major entities and their relationships, such as associating quantities with specific foods or exercise times.
- **Nutrient and Exercise Matching:** Identified foods are matched against either user-provided custom ingredients or generic food databases to return the nutrient data.

4.5 Custom Items and Meals

The system facilitates personalized nutrition tracking by enabling users to build and maintain personal foods and composite foods:

- **Custom Food Items:** Users can create new food items by entering serving size, calories per serving, and macronutrients like protein. This is particularly helpful for home-prepared meals or special dietary items not included in typical databases.
- **Composite Meals:** Several custom or predefined items can be combined to form a composite meal. The nutritional values of such meals are precalculated and stored, allowing quicker retrieval and automatic identification by the NLP engine.

This flexibility gives freedom to users with different dietary lifestyles and provides complete tracking.

4.6 Logging to Database

Precise data persistence is established using structured database logging:

- **Tables:** There are two primary tables used:
  - **FoodIntake Table:** Records user email, date, time, food item, amount consumed, calories, protein, and description.
  - **ExerciseDone Table:** Logs user email, date, time, exercise type, duration, calories burned, and description.
- **Entry Creation:** One entry gets added to the corresponding table for every successful NLP parsing, ensuring that the system has a full record of the user's dietary and exercise history.

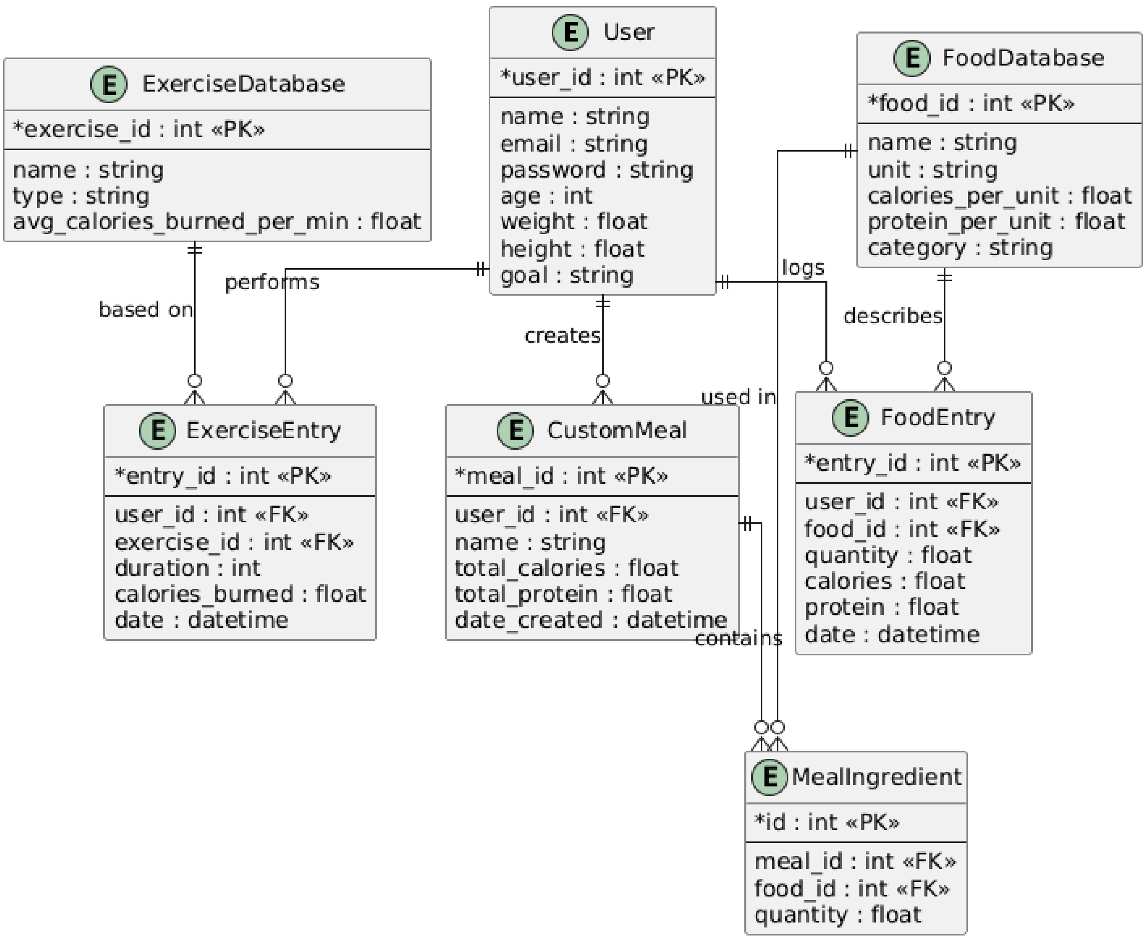

FIGURE 3: ER - Diagram

The complete Entity-Relationship (ER) diagram is presented in Figure 3. It visually represents the database structure of the proposed calorie tracking web application. Each rectangle in the diagram corresponds to a database table, capturing distinct components of the system such as users, food entries, exercise logs, the food and exercise master databases, custom meals, and their respective ingredients.

The `User` table holds profile information including physical attributes and preferences. The `FoodEntry` and `ExerciseEntry` tables record user-specific food consumption and physical activities, linking them to the respective master tables—`FoodDatabase` and `ExerciseDatabase`. For personalized tracking, users can create `CustomMeal` entries, each consisting of multiple ingredients stored in the `MealIngredient` table.

All tables are connected through primary and foreign key relationships, ensuring referential integrity and enabling efficient querying of user behavior, nutritional intake, and exercise history. This ER model provides a scalable and normalized foundation for storing structured data that supports both core functionalities and potential future enhancements such as recommendations and chatbot interaction.

4.6.1 Reports and Visualization

Comprehensive reporting is done through three sole-purpose endpoints:

- `/daily_report`
- `/weekly_report`
- `/monthly_report`

Every report has comprehensive summaries of calorie intake, calories expended, net calorie balance, and protein consumed. Visualizations utilize Chart.js with:

- Daily intake and expenditure logs displayed through bar charts.
- Line charts to monitor weekly trends over time.
- Stacked charts to display monthly summaries of calories and nutrients.

This multi-level reporting helps users identify patterns, set goals, and make educated dietary or exercise changes.

4.7 Speech-to-Text Integration

Improving usability, the system has a speech-to-text interface that utilizes JavaScript's Web Speech API:

- Users turn on the microphone icon and utter input.
- The voice recognition translates audio to text in real-time.
- The user is able to correct the transcript prior to submission.

This feature is mainly supported on Chrome-based browsers, broadening accessibility particularly for mobile users or voice command preference.

4.8 Technologies Used

Component	**Technology**
Backend	Flask + Python
NLP	spaCy (`en_core_web_md`)
Speech Input	JavaScript (Web Speech API)
ORM (Object Relational Mapping)	SQLAlchemy
Database	SQLite (migratable to PostgreSQL)
Frontend	Bootstrap + Jinja2 + Chart.js
Hosting (Planned)	Docker + Heroku / AWS

TABLE 2: Technologies Used in CalorieTracker

4.8.1 Backend Technologies

- **Python 3.10+**: Primary programming language used for server-side logic and data processing.
- **Flask 2.2+**: Lightweight Python web framework used to handle routing, form handling, and user sessions.
- **Flask-Login 0.6+**: Manages user authentication and session persistence.
- **Flask-WTF 1.1+**: Enables secure form rendering and validation in Flask using WT-Forms.
- **SQLAlchemy 1.4+**: ORM for managing SQLite (and optionally PostgreSQL) databases.
- **spaCy 3.7+**: NLP library used for tokenization and Named Entity Recognition (NER).
- **en_core_web_md (spaCy Model)**: Medium-sized English model used for entity extraction and dependency parsing.
- **Requests 2.31+**: Handles HTTP requests for optional external food APIs (e.g., Spoonacular).

4.8.2 Frontend Technologies

- **HTML5**: Structure for web pages and user forms.
- **CSS3 / Bootstrap 5**: For responsive and visually clean user interface styling.
- **JavaScript (ES6+)**: Adds interactivity to the frontend, including speech input and dynamic UI updates.
- **Chart.js 4+**: JavaScript charting library used to visualize calorie and protein trends.
- **Web Speech API**: Built-in browser API used for converting voice input to text in supported browsers.

5 Result and Analysis

This section describes the results of the system implemented and how effective it is in fulfilling the aim of the project. It gives a comprehensive analysis of the performance of the system, user interaction using natural language input, data processing accuracy, and general usability. The results are explained with examples and observations to portray the strengths and weaknesses of the approach.

5.1 Natural Language Input Example

Users can type input such as:

> ”I had a protein shake and two boiled eggs.”

This input is processed by the spaCy NLP pipeline to extract food items and quantities. These are mapped to nutritional values from an internal or external database. The result is an automated log entry with calories (e.g., 400 kcal) and protein (e.g., 30g). This reduces the need for manual form filling and speeds up the logging process. Figure 4, shows the dashboard of project where user can give the natural language queries.

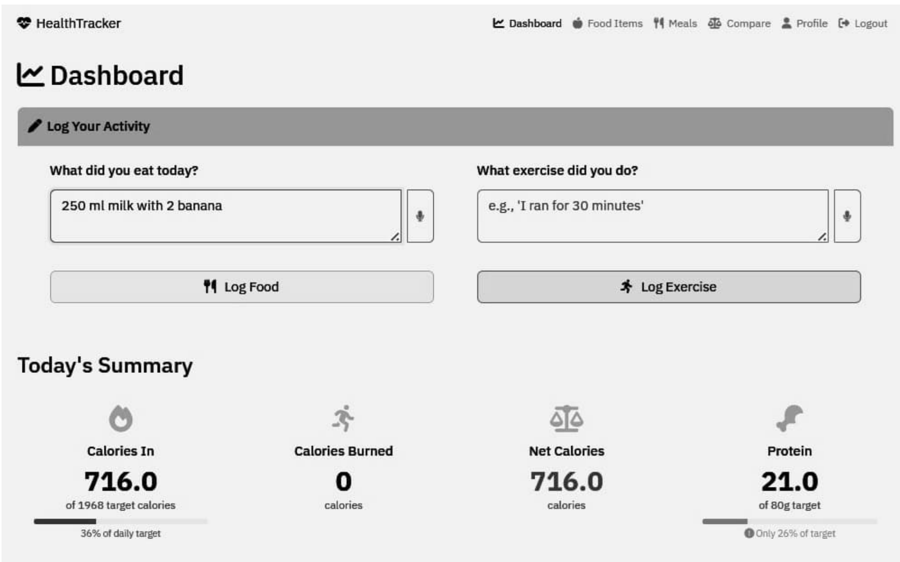

FIGURE 4: Natural Language input dashboard

5.2 Speech-to-Text Logging

Voice-based input using the Web Speech API enables users to log meals or workouts by speaking. This input is converted to text, parsed, and logged similarly to typed input.

Caption: Voice input is transcribed and processed seamlessly, improving accessibility and usability for all users.

5.3 Custom Item and Meal Builder Interface

Users can define new ingredients or meals by providing:

- Name (e.g., *Milkshake*)
- Calories and protein per serving
- Default quantity and optional description

Once created, these items are stored and recognized by the NLP module during future logging. Figure 5, shows the form that add the custom item , which can be logged into database by natural language query or manually.

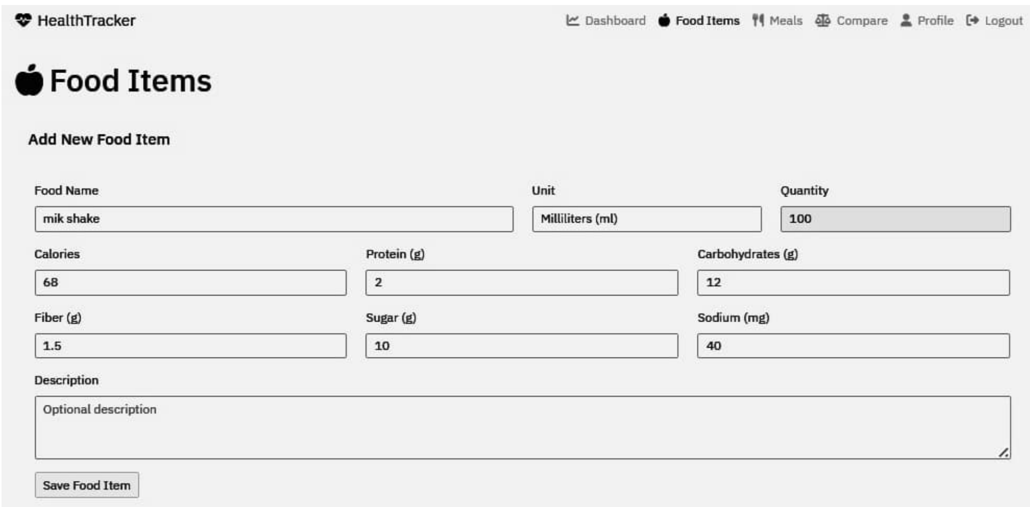

FIGURE 5: Add Custom Item

5.4 4. Custom Item Logging via NLP

A sentence like:

> ”I had my Milkshake.”

is interpreted correctly by the system, which retrieves the corresponding nutritional information for the custom-defined meal and logs it automatically. This enhances personalization and supports user-defined diets or home-cooked recipes.

5.5 Daily, Weekly, and Monthly Reports

Visualization Figure 6, tools offer users summaries of their nutrition and activity:

- **Daily Report:** Displays bar graphs of food intake and activity by meal.
- **Weekly Report:** Shows trends in calories consumed and burned.
- **Monthly Summary:** Presents stacked bar charts summarizing net calorie balance and protein intake.

These reports assist users in monitoring their lifestyle and adjusting dietary habits accordingly.

Figure 6: Visual Charts

5.6 User Profile View

Users' profiles contain personal information—name, age, weight, height, and gender—necessary for personalized calculations. This data is also used to compute calories burned during exercises using MET values. Profiles can be updated anytime to reflect changes in fitness goals.

6 Conclusion

The Flask-based natural language calorie tracker successfully combines advanced natural language processing (NLP), intuitive web design, and robust backend development to create a user-centric application that simplifies calorie management. By enabling users to log their calorie intake and expenditure using everyday language, the application overcomes the limitations of conventional calorie tracking tools that often require structured input, making it accessible and user-friendly.

- **Natural Language Input Processing:**
  - The application effectively interprets user queries, such as "I ate a sandwich and walked for 30 minutes," extracting relevant entities like food items, quantities, and activities.
  - SpaCy's NLP model, enhanced with domain-specific rules, demonstrated high accuracy in parsing diverse inputs.
- **Accurate Calorie Tracking:**
  - Calorie values for foods and activities were mapped to predefined lookup tables, enabling precise calculations.
  - Net calorie balances were computed seamlessly for each query, providing users with real-time insights.
- **Comprehensive Reporting:**
  - Users could generate daily, weekly, and monthly reports, offering clear visualizations of their calorie trends.
  - Interactive charts and tables improved the interpretability of the data, helping users make informed decisions about their diet and exercise routines.
- **Database and Scalability:**
  - SQLite ensured reliable and efficient data storage, while indexing and optimized queries supported fast data retrieval.
  - The application's modular design ensures it can scale to accommodate more users and larger datasets in the future.

- **User-Centric Design:**
  - The use of Bootstrap and responsive design principles resulted in a visually appealing and accessible user interface.
  - The natural language input method reduced cognitive load, making it easy for users to track their progress consistently.

6.1 Limitations

While the application achieved its primary objectives, certain limitations provide opportunities for further improvement:

- **Ambiguity in Queries:** Certain inputs, such as "I had a light snack," may require clarification or fallback mechanisms.
- **Calorie Data Limitations:** The current lookup table can be expanded to include more food items and regional cuisines.
- **Scalability:** Although SQLite performed well, migrating to a more scalable database like PostgreSQL or MySQL would better handle increased data volume.
- **Accessibility:** Adding features like voice input, multi-language support, and mobile app integration would broaden the application's user base.

This application is a step toward redefining how individuals approach calorie tracking. By blending simplicity, accuracy, and user-friendly design, it ensures users can manage their dietary habits with ease and precision. Its modular architecture and scalability make it a strong foundation for future enhancements, ensuring its relevance in an increasingly health-conscious world. With continued development, this application has the potential to become an indispensable tool for promoting healthy lifestyles.

6.2 Future Scope

Although the existing system provides effective natural language-based protein and calorie tracking, there are various improvements that can substantially increase its performance, user experience, and scalability. The following enhancements are envisioned for future development:

- **Better Datasets for Enhanced Accuracy:**
  Create or incorporate more inclusive and diverse food and activity datasets encompassing regional foods, branded foods, and different exercise types. This will provide increased coverage, accuracy, and personalization to a broader set of users.

- **Intelligent Recommendation System:**
  Install a personal recommendation system that recommends meals and exercises based on user choice, nutritional objectives, activity history, and health indicators. Machine learning algorithms may be trained to refine these suggestions over time.

- **Cross-Platform Mobile Application:**
  Develop and deploy a complete mobile app for Android and iOS devices. This will allow users to monitor their food habits and workouts in real-time, get notifications, and have voice/text-based logging capabilities on-the-go.

- **AI-Powered Chatbot Assistance:**
  Implement an AI/NLP-powered interactive chatbot that is able to respond to user questions, walk them through the app, provide suggestions for healthy behavior, and help them log entries more conversationally and naturally.

References

[1] M. Korpusik, S. F. Taylor, S. Das, C. Gilhooly, S. B. Roberts, and J. Glass (2019). *A food logging system for iOS with natural spoken language meal descriptions*, Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI).

[2] I. Choi and J. Kim (2022). *Dietary pattern extraction using natural language processing techniques*, Frontiers in Nutrition, vol. 9, p. 765794. https://doi.org/10.3389/fnut.2022.765794

[3] M. Weber, J. Sun, X. Zhang, and J. Hoelscher (2022). *Natural language processing and machine learning approaches for food categorization and nutrition quality prediction compared with traditional methods*, The American Journal of Clinical Nutrition, vol. 115, no. 4, pp. 1175–1183. https://doi.org/10.1093/ajcn/nqab305

[4] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf (2019). *Transfer learning in natural language processing*, in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pp. 15–18.

[5] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, et al. (2020). *Transformers: State-of-the-art natural language processing*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45.

[6] P. Kr mer, M. Möller, R. Munzert, and M. Riedel (2021). *Fine-tuning BERT for ingredient extraction from recipe text*, Journal of Biomedical Informatics, vol. 118, p. 103792. https://doi.org/10.1016/j.jbi.2021.103792

[7] M. Neumann, D. King, I. Beltagy, and W. Ammar (2019). *SciSpaCy: Fast and robust models for biomedical natural language processing*, in Proceedings of the 18th BioNLP Workshop and Shared Task, pp. 319–327.

[8] M. Honnibal and I. Montani (2021). *spaCy 3: A new era of industrial-strength NLP in Python*, Explosion AI Blog. [Online]. Available: https://explosion.ai/blog/spacy-3

[9] Y. Titova, P. Harada, and J. Chang (2023). *Using social media analysis to study population dietary behaviours: A scoping review*, Journal of Human Nutrition and Dietetics, vol. 36, no. 2, pp. 307–324. https://doi.org/10.1111/jhn.13077

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups

- **41** Not Cited or Quoted 8%
  Matches with neither in-text citation nor quotation marks
- **2** Missing Quotations 0%
  Matches that are still very similar to source material
- **2** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

7%	Internet sources
4%	Publications
6%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.