

A3: Build your own Language App

1. How does the App work?

I developed a web application using Flask, a micro web framework for Python, offering a nice interface that caters to health-conscious individuals and those seeking interactive responses via a chatbot. In the following section I will provide a detailed exploration of the app's functionalities, underlying technologies, and user interactions.

Application Structure and Environment Setup

The application leverages the power of Flask to manage web requests and render HTML templates dynamically. An important aspect of the setup includes the use of an environment variable to securely store and retrieve an API key necessary for the OpenAI API interactions.

Core Functionalities

1. Calorie Calculator:

- a. **Interface:** The homepage of the application presents a user-friendly form where individuals can input their age, gender, height, weight, and activity level.

The screenshot shows the 'CS50 Fitness' logo and navigation links 'Calorie Calculator' and 'Chat bot'. Below is a heading 'Welcome to CS50 Fitness' and a sub-heading 'By providing your actual parameters (in metric units), you'll be able to know how many calories you need in order to lose, gain or maintain your weight and achieve to your fitness goals!'. The form contains five input fields: 'Age' (a dropdown menu with 'Enter your age'), 'Gender' (a text input with 'Select your gender'), 'Height' (a text input with 'Enter your height (in cm)'), 'Weight' (a text input with 'Enter your weight (in kg)'), and 'Activity' (a text input with 'What's your daily physical activity'). A blue 'Calculate' button is positioned below the 'Activity' field.

- b. **Functionality:** Upon form submission via a POST request, the application calculates the Total Daily Energy Expenditure (TDEE) based on the Harris-Benedict equation, which adjusts basal metabolic rate (BMR) according to the user's activity level. The calculated TDEE helps users understand how many calories they need to maintain their current weight, which is crucial for making informed dietary choices.

The screenshot shows the 'CS50 Fitness' logo and navigation links. Below is a table with two columns: 'Weight goal' and 'Calories per day'. The table lists five goals: 'Maintain weight' (2813 Kcal/day), 'Mild weight gain' (3063 Kcal/day), 'Mild weight loss' (2563 Kcal/day), 'Weight gain' (3313 Kcal/day), and 'Weight loss' (2313 Kcal/day). Below the table is a section titled 'Caloric Recommendations' with five bullet points: 'By adjusting your daily calorie intake to 2813 Kcal, you will maintain your current weight.', 'By increasing your daily calorie intake to 3063 Kcal, you can expect to gain around 0.25kg per week.', 'By decreasing your daily calorie intake to 2563 Kcal, you can expect to lose around 0.25kg per week.', 'By increasing your daily calorie intake to 3313 Kcal, you can expect to gain around 0.5kg per week.', and 'By decreasing your daily calorie intake to 2313 Kcal, you can expect to lose around 0.5kg per week.' At the bottom is a disclaimer: 'Disclaimer: While this calculator can provide a rough estimate of daily calorie needs, it does have limitations. It does not take into account individual differences in body composition, genetics, and hormone levels, all of which can influence an individual's metabolism and calorie needs. Additionally, it assumes that all individuals within a gender and age group have similar physical activity levels, which may not always be the case. Therefore, it is important to use the Harris-Benedict equation as a starting point and adjust calorie intake based on individual needs and goals.'

| Weight goal | Calories per day |
|------------------|------------------|
| Maintain weight | 2813 Kcal/day |
| Mild weight gain | 3063 Kcal/day |
| Mild weight loss | 2563 Kcal/day |
| Weight gain | 3313 Kcal/day |
| Weight loss | 2313 Kcal/day |

Caloric Recommendations

- By adjusting your daily calorie intake to 2813 Kcal, you will maintain your current weight.
- By increasing your daily calorie intake to 3063 Kcal, you can expect to gain around 0.25kg per week.
- By decreasing your daily calorie intake to 2563 Kcal, you can expect to lose around 0.25kg per week.
- By increasing your daily calorie intake to 3313 Kcal, you can expect to gain around 0.5kg per week.
- By decreasing your daily calorie intake to 2313 Kcal, you can expect to lose around 0.5kg per week.

Disclaimer: While this calculator can provide a rough estimate of daily calorie needs, it does have limitations. It does not take into account individual differences in body composition, genetics, and hormone levels, all of which can influence an individual's metabolism and calorie needs. Additionally, it assumes that all individuals within a gender and age group have similar physical activity levels, which may not always be the case. Therefore, it is important to use the Harris-Benedict equation as a starting point and adjust calorie intake based on individual needs and goals.

- c. **Feedback:** If a user submits invalid data (e.g., negative numbers), the application uses a helper function called 'apology' to generate error messages, thus enhancing the user experience by providing clear guidance on input expectations.

2. Chat Bot

- a. **Interface:** A separate page equipped with a form allows users to ask questions.

- b. **Functionality:** On form submission, the application leverages the OpenAI API to process the user's question and generate a response. The interaction mimics a real-time chat environment, making it engaging for users to receive AI-powered responses.

- c. **User Interaction:** This feature effectively simulates a conversational agent, providing users with a platform for obtaining instant responses to their queries, which can range from general knowledge to specific advice.

- d. **LLM description:** The chat bot uses the OpenAI "text-davinci-002" model to generate responses to user queries. This model is called with specific parameters aimed at producing precise and relevant outputs. The temperature is set to 0.2 to ensure focused responses, while the response length can extend up to 1024 tokens. It also uses a top_p value of 1, indicating that the model considers the full range of possible continuations based on probability, and both frequency and presence penalties are set to 0, allowing for natural repetition and relevance in the generated text. This configuration ensures that the responses are coherent, contextually appropriate, and directly responsive to the user's input.

Web Page Templates and Styling

The application uses Flask's templating engine to serve HTML content dynamically. The templates ensure a consistent layout across different pages, enhancing the visual and functional coherence of the application. Bootstrap is employed to style the forms and results, ensuring the interface is responsive and accessible on various devices.

Linguistical importance of the App

The development of the described web application integrates a NLP technology, specifically leveraging OpenAI's language model "text-davinci-002", to create a chatbot feature. This integration not only demonstrates the practical application of NLP in interactive web environments but also shows the linguistic significance of using advanced language models in real-world applications. Below, I'll explore why this app is linguistically meaningful in the context of NLP and language models.

Advanced Language Model Usage

The application uses OpenAI's language models, which are among the leading technologies in NLP nowadays. These models have been trained on diverse datasets, enabling them to understand and generate human-like text. By incorporating such a model, the app facilitates sophisticated interactions that can interpret user inputs with a high degree of linguistic nuance. Also, the language model's ability to process natural language input and generate coherent, contextually appropriate responses demonstrates its robust understanding of language syntax, semantics, and pragmatics. This capability is important in applications where user engagement depends on the system's ability to communicate rightly.

Interactive Language-Based Features

The chatbot feature enhances the app's utility and user engagement by providing a platform for real-time, language-based interaction. Users can pose questions in natural language and receive answers generated by the language model. This interaction is linguistically significant as it not only showcases the model's response generation capabilities but also its ability to handle a wide range of inquiries from straightforward factual questions to more complex queries requiring understanding of context.

Linguistic adaptability and personalization

The use of a language model allows the chatbot to adapt its responses based on the user's input style and complexity. This adaptability can make the chatbot more effective in communicating with users of varying linguistic abilities, enhancing the personalization of the interaction.