Back End REST API

Author: Andrew DeLeon

Health at Hand utilizes a REST API, coded in the Flask framework in Python. The REST API receives calls from front-end points of the application. Data is transmitted to and from the database as a JSON object. For the MySQL database utilized in the application, PYMYSQL is utilized. The REST API also makes API calls to Nutrition IX, for food and nutritional data tracked in the food log for calorie tracking. Due to issues with the SIS server hosted by Pitt, the program was unable to be deployed and hosted by the University.

In terms of hosting, do not host the application on any Enterprise Web Infrastructure (EWI), as Pitt’s server proved to run into issues upon deployment for this reason.

The remaining documentation will go into detail of each function of the REST API operating on the backend of Health at Hand.

**REST API Calls**

**Endpoint:** Login (/login)

**Methods Utilized:** GET

**Description:** The login function received two arguments, *email* and *password.* The REST API makes a call to the database and returns a 200 success if the user exists. If not, there will be a 400 error message given to the user. The front end will also receive the health coach ID number associated with the user upon the user being registered and found.

**Endpoint:** Dashboard (/dashboard)

**Methods Utilized:** GET

**Description:** The dashboard gathers the userID and the date. Using this information, the server will return the remaining calories and total calories from the database for the user for the day. It will also return information for the weekly exercise duration, both remaining and total minutes.

**Endpoint:** Meal Notes (/mealnotes)

**Methods Utilized:** GET, POST, PUT

**Description:** The meal notes endpoint gathers the user ID and date to display the current food diary note for the day to the user. The endpoint also offers the opportunity for the user to add a new food diary note, as well as update it at any point due to error or typo.

**Endpoint:** Exercise Diary (/exernotes)

**Methods Utilized:** GET, POST, PUT

**Description:** *See Meal Notes*

**Endpoint:** Reports (/report)

**Methods Utilized:** GET, POST, PUT

**Description:** The function for reports gathers all weight and date information for the user to display as a chart on the front end. New weight inserts can be added at any time under the POST method. If any updates should occur to weight, the POST handles any updates to previous weights inserted.

**Endpoint:** Email (/email)

**Methods Utilized:** POST

**Description:** The email endpoint stores all records of messages sent from the user to the health coach into the database. Once the user hits send, the endpoint gathers the contents of the email to store to the database.

**Endpoint:** Meal Log (/meallog)

**Methods Utilized:** GET, POST, PUT, DELETE

**Description:** Incorporates API calls to Nutrition IX. The GET method will gather all meal information for the user, along with any foods attached to that meal. Users can add new meals, which will hit the POST method. They can also update and delete existing meals using the PUT and DELETE functions. Data stored will also include the calories of each food, along with the meal, based on serving size entered.

**Endpoint:** Exercise Log (/exerlog)

**Methods Utilized:** GET, POST, PUT, DELETE

**Description:** Similar to meal log, exercise log utilizes the Nutrition IX API for workout regimens performed. The server will gather all existing logs for the day, along with the duration and intensity for said regimen. Users can update or delete existing entries, with the front end hitting either the PUT or DELETE methods respectively.

**NutritionIX API Calls**

**(**Imported from nutritionix.py)

**Endpoint:** Search (/search)

**Methods Utilized:** GET

**Description:** With the parameters being food searched, the REST API will connect to the NutritionIX API and return two categories of search results; Branded & Common. Branded foods include objects such as “Giant Eagle Honey”, where as common will return unbranded, common results such as “Honey.” Users can easily find exactly the object they are searching for and add their respective food to their meal.

**Endpoint:** Nutrition (/nut)

**Methods Utilized:** GET

**Description:** The nutrition endpoint works hand in hand with the search endpoint on the front end application of Health at Hand. For uncommon foods, the nutrition endpoint **must be utilized**. As unbranded (common) foods do not return with caloric information, the nutrition endpoint will supply this to the front end for calorie calculation.

**Endpoint:** Exercise (/exercise)

**Methods Utilized:** GET

**Description:** To guide the user for available workout regimens, the exercise search will return existing workouts from Nutrition IX. Users can then select a workout, and work through the applications intuitive front end design to select both the intensity level and duration of workout.