[CMSC 455] Assignment #2: REST APIs

Due: End of Day, September 26th, 2024

Objective: The objective of this assignment is to create a microservices application for grocery shopping. Students will develop two Flask microservices, "Product Service" and "Cart Service," with specific endpoints. They will also deploy both services on the Render platform. This assignment will help students gain practical experience in microservices architecture, API development with Flask, and cloud deployment.

Task Overview:

- *Service 1: Product Service*

  Create a Flask application named "Product Service" that serves as the first microservice. Implement the following endpoints:

  `/products` (GET): Retrieve a list of available grocery products, including their names, prices, and quantities in stock.
  `/products/product_id` (GET): Get details about a specific product by its unique ID.
  `/products` (POST): Allow the addition of new grocery products to the inventory with information such as name, price, and quantity.

- *Service 2: Cart Service*

  Develop another Flask application named "Cart Service" serving as the second microservice. Create the following endpoints, which interact with the "Product Service" microservice:

  `/cart/{user_id}` (GET): Retrieve the current contents of a user's shopping cart, including product names, quantities, and total prices.
  `/cart/{user_id}/add/{product_id}` (POST): Add a specified quantity of a product to the user's cart.
  `/cart/{user_id}/remove/{product_id}` (POST): Remove a specified quantity of a product from the user's cart.

- *Communication:*

  1. Establish communication between the "Cart Service" and the "Product Service" using HTTP requests (e.g., GET, POST).
  2. Ensure that the "Cart Service" can successfully retrieve product information from the "Product Service" and manage the user's shopping cart. Deployment on Render:
  3. Sign up for an account on the Render platform (https://render.com/).
  4. Deploy both the "Product Service" and "Cart Service" on Render.
  5. Configure any necessary environment variables or settings for your applications on Render.

- *Hints:*

1. Get the "Product Service" working and test all of its end points using curl (or similar), then get the "Cart Service" working, then try to deploy. Getting ahead of yourself in this assignment can lead to difficult problems to debug (e.g. the service is not responding, is it the code or I didn't deploy it properly?)

2. Your don't have to use a database in this assignment, but it would be nice if you used SQLite via flask_sqlalchemy as we demoed in class.

Submit: Submission is via GitHub. To accept the assignment, visit `https://classroom.github.com/a/hyNkYuwz` and create your personal repository. We will grade your submissions directly on GitHub. There will be no submissions on Canvas. List the URLs in the README.md file (in the base directory of the repo) used to contact both of your services on Render.