

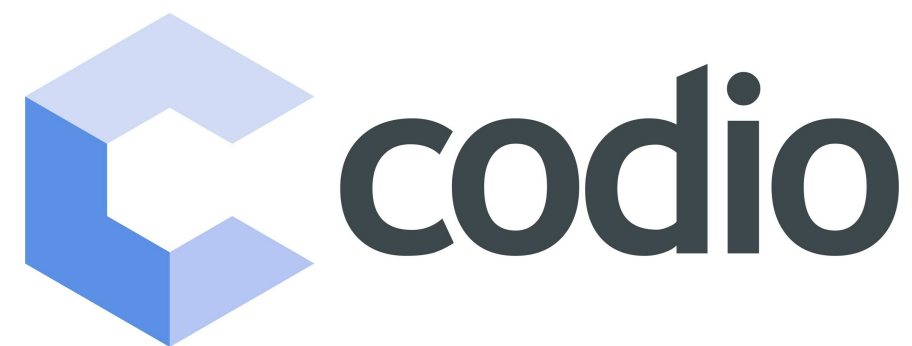
E-Commerce Cloud Project

Problem Being Solved

- Create a realistic E-Commerce cloud application that can enable selling goods online with an intuitive experience.
- Employ Flask on Codio to provide a simple and repeatable environment for developing, debugging, and viewing the application.
- Ensure the application can work on a number of browsers and conditions and the final AWS output is correct in each case.

Methods

- Flask was used to host the application which enabled a simple backend to facilitate the site interaction.
- DynamoDB on AWS is used as the final write destination for order placement.
- The app runs on the Codio system which exposes a browser interface for clients to talk to.



Features

- The site features a user registration and login page with input validation and error messages.
- After signing in users are able to place items into a cart from an inventory selection.
- They can then view the contents of their cart along with item information and send the order where it could be processed.

Challenges

- Jinja enables Python data from Flask to be dynamically queried and embedded in the HTML. This is a powerful feature but requires a careful reading of the documentation on the syntax.
- Unit testing enables ongoing verification of methods through the API interface, but the code should be refactored as much as possible so the critical logic in the browser code path is verified as well through these tests.
- During partial feature development it is tempting to want to use Curl from the terminal then test only the implemented browser features. However, similar to someone opening multiple page,s the cookies aren't shared alternatives were pursued.

Architecture Summary

- Flask is run as part of Codio's hosted linux environment.
- The Client's browser can then access the page generated by this and send requests and responses back to the Flask server.
- The AWS DynamoDB service is used to write to by the Flask application after the order has been gathered and is ready to be placed.

