Gilbert Polo

COP-4045 Python programming

Final Project

Python version 3.5.6

macOS X Sierra

IDE Xcode 8.3.3

IDE Python IDLE for Debug

Python Web Framework Flask

IBM Cloud Lite Account (BlueMix)

Introduction

This final python project is based off of the book’s sample shopping cart object oriented design. The project design and code is comprised of three different tiers; database, business, and web. For the database tier, Sqlite3 is used as the database. Python Flask is used as the web framework, which is used to provide the UI interface.

Database Tier

This tier is responsible for the design and creation of the Sqlite3 database. It also has a python module db.py that contains static functions to read and write information from and to the Sqlite3 database.

Business Tier

This tier contains the objected oriented design of the shopping cart program. Each python class represents part of the application such as the products available. The client’s shopping cart and the current items in the shopping cart.

Web Tier

This tier contains the python Flask modules that is used to create the users UI experience.

Project Folder Structure

* FinalProject
  + shopping
    - db
    - shopping
    - web
    - test

Application Flow.

1. Create database tables and populate products table
2. The user begins by starting a browser and pulling up the available products.
3. Products are selected and added to their shopping cart.
4. Using python flask, the cart’s primary id is stored in the user’s cookie.
5. The user can continue adding more products to their shopping cart, or remove items from their cart.
6. Finally the order is submitted for final processing and the whole session begins again from step 2.

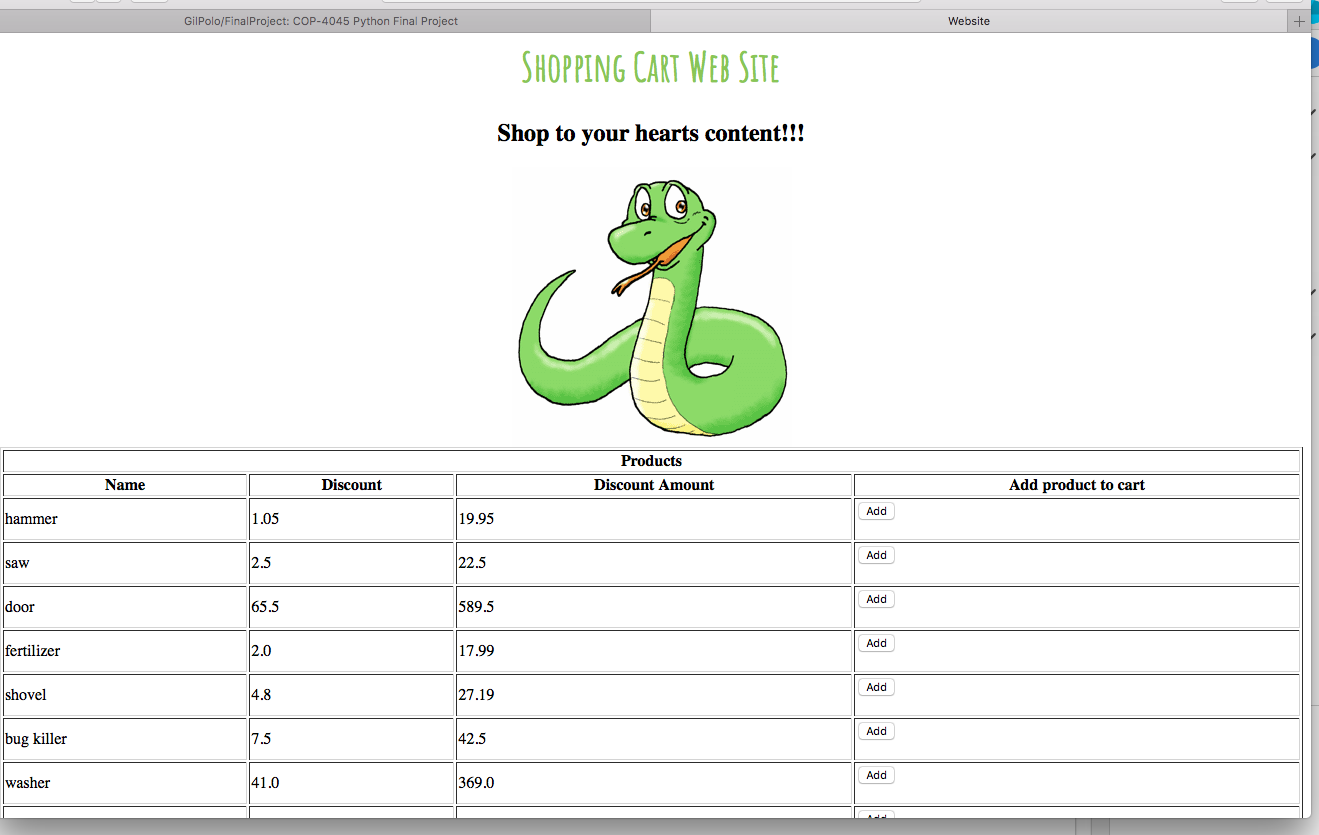
Challenges Encountered.

Numerous challenges were encountered developing this project, too many to mention. However, I can say that the bulk of the challenges were with Python Flask itself. Learning how to create a web application took time and there was re-design because of this. It seemed like once I got thru one hurdle, there was another waiting to be overcome. The handling of adding and removing products from a cart were a challenge, at the same time keeping all of this information in sync with the database was also a challenge. But with a lot of debugging I overcame these hurdles. The deployment of a testing module benefited me greatly because I was able to test both the business and database tiers independently of the web module. Without this it would have been to much to test and debug all three tiers at the same time.

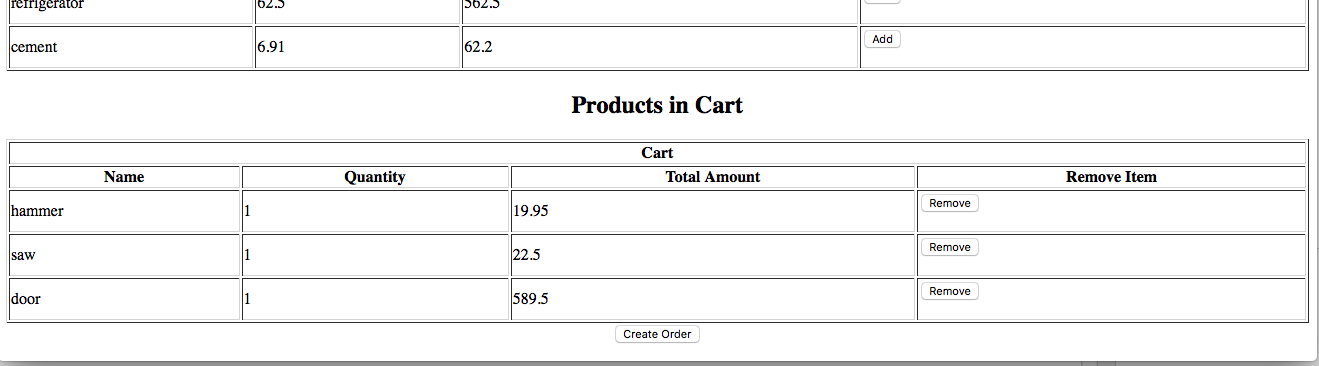
Cloud challenges.

Wow!!! This was very challenging. Documentation is what can you say IBM documentation ☺. BlueMix is now IBM Cloud so whatever documentation there is its on BlueMix. I got burned by downloading IBM Cloud CLI which is somehow not compatible with an IBM Cloud Lite account. Had to download Cloud Foundry CLI (cf) for mac, then things started to work until finally got my application running on the cloud.

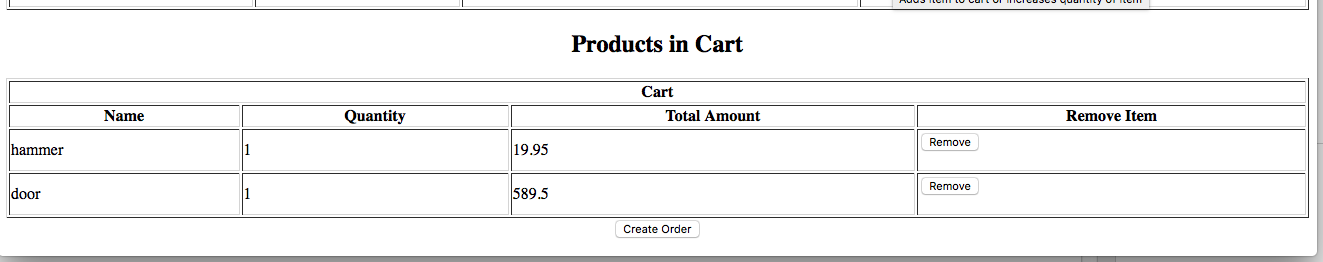
Screenshots.



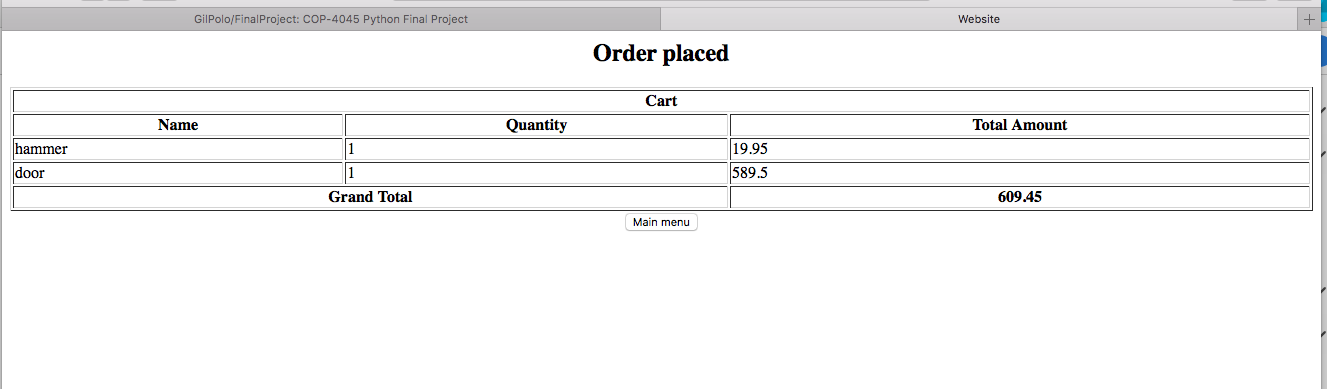
Screenshot after adding three products.



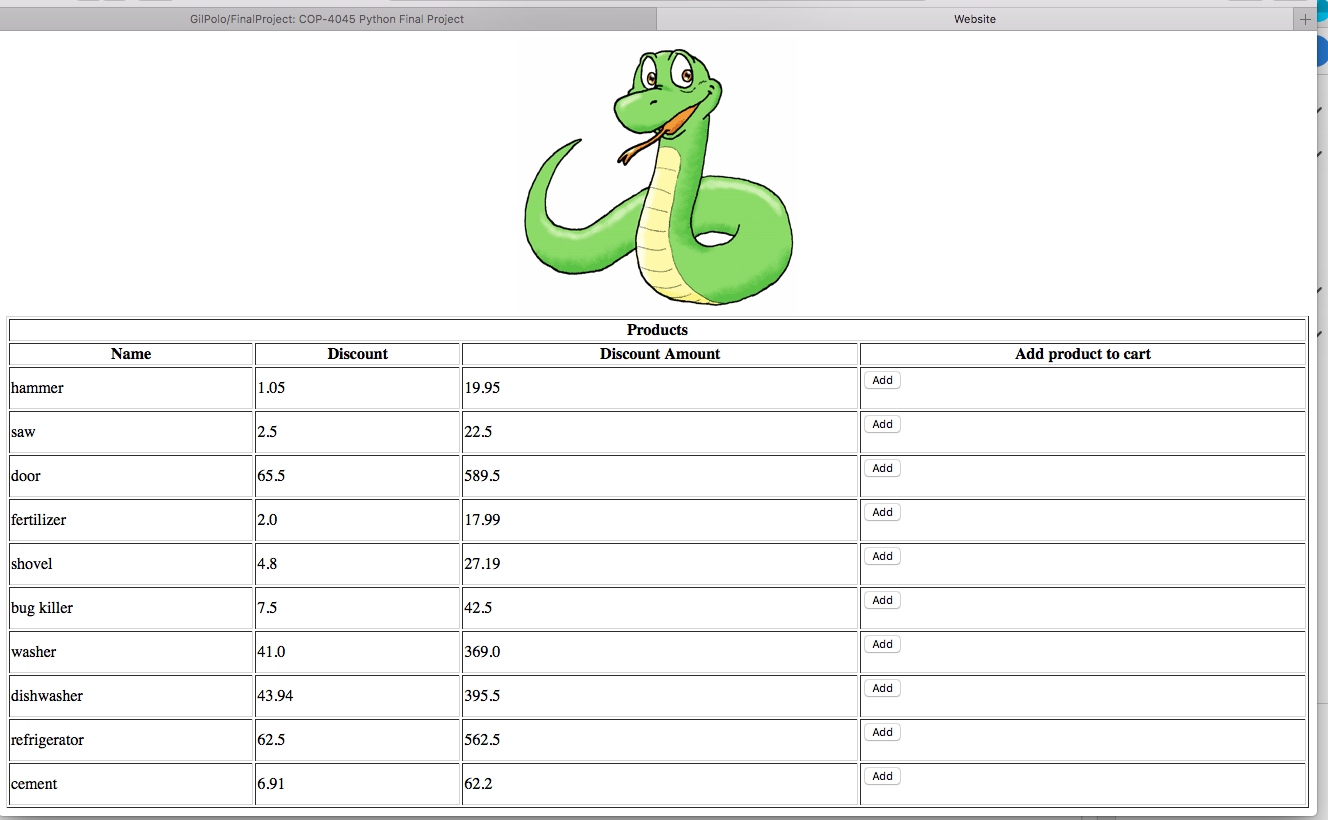
Screenshot after removing “saw” product from cart.



Screenshot of cart submitted for fulfillment.



Screenshot of hitting “Main menu” on prior web page. This shows the main menu after an order, which allows the user to start the whole process again.



Future Work.

Finally, I believe that future work to the project should concentrate on making the application more “bullet-proof”. i.e. handling of all possible exceptions. Also, using python logging facilities to log more information. This would allow for more debugging and performance tuning.

Git.

Project can be found on GIT: <https://github.com/GilPolo/FinalProject>

IBM Cloud URL

<https://gpolocop4045finalproject-talkative-platypus.mybluemix.net/>

Running the application.

* Initialize database
  + go to directory shopping/db
  + sqlite3 \_products.db < sql.txt
* change directory to root of directory in other words parent of the “shopping” directory
* execute python -m shopping.web.main