Manpreet Bahl

Inventory Management System Project Report

The overall architecture of the inventory manage system is straightforward. There's a frontend component which is composed of a checkout page, checkin page, and a request page. The checkout page allows the user to select items from an inventory table, indicate how much of the selected items they wish to checkout, and then allow them to checkout the items. The checkin page has an identical user interaction as the checkout page but the checkin page performs a checkin rather than a checkout. The request page allows the user to submit a form for an item that does not exist in the inventory and their contact information. The form data is emailed to the inventory administrator who can take any appropriate actions. The backend of the system is written in Python Flask which is a web framework. It's similar to Express in terms of the code structure with routes tied to a functions that handle the data processing, but with Flask, the server can also utilize other Python dependencies to provide functionality.

At the present state, the inventory management system meets the core functionalities that were made during the project pitch. However, I do plan to continue working on this system and add more functionality; some of the features are described in the stretch goals of the project pitch. This can also be evident in the current state of the application as well. For example, I plan to have a login page added to the system, thus the frontend has a login button in the navigation bar. The database also contains a table containing all login information but hasn't been utilized yet. I plan to implement this feature correctly as possible in that login is secure and doesn't have a high chance of compromising user information. This also brings to me to a concern I have for the system at the moment. There is some sensitive information that needs to be hardcoded into the backend in order for the application to work. In particular, the email login credentials are hardcoded since the server needs to authenticate into the email account before being able to send an email. The database credentials also are hardcoded for now, and I plan on doing research into how to effectively obtain credentials securely for the server to use.

Lastly, the application itself looks plain and simple and I wanted to add some cosmetics to it to make it more visually appealing. However, while I was working on this project, I focused more on the functionality and wanting to get it done as correctly as I can. Overall, I am satisfied with the end outcome of the application at the present state, but I also know that there is a lot of improvements and functionality that I can add to the system to make it even better.