A diagram of a company

Description automatically generated

GENERAL INFORMATION

This Database project was built as a web application utilizing flask and SQLite. I used the in-class example as the base for setting up the login system and created the rest of the application off of it. For the sake of documentation, I will explain the basics of how it works. On start up the program sets up flask as well as creates a key to be used for session cookie hashing. The program then calls the sys\_init() function which initializes the database if it doesn’t exist as well as insert some basic data into it if that doesn't exist either. The user is then sent to the login page, the input is sent back to project.py using post command in script.js, and the login(username, password) function is called which checks the parameters against the accounts stored in the db. On a success the user data is recorded to the session and script.js redirects to the /library route. From there the role of the user if checked and if they are a librarian they are sent to their respective page and if they are a patron they are sent to their own page. Each respective page is filled with operations only they can perform. Once the relevant information is filled and the corresponding button pressed, the js (libScript.js for librarian page and patScript.js for patron) sends a json to the /route with the relevant information and what function is to operate. The /data route calls the relevant function from dbmanip.py and the result is sent back to the js and displayed on the page. When the user no longer wishes to continue, the can click the logout link which calls the /logout route to end the session.

DETAILS

RBAC: For the Role based access control I have created an if statement on log in that redirects to a librarian specific or patron specific page. This limits patrons and librarians to the functions provided to them on their respective pages.

DATA SECURITY: Data security is achieved through utilizing password hashing and parameterizing the data that is passed into any query or database alteration.

CONSIDERATIONS

* I decided to make the librarian and patrons part of the same table users, as their relevant attributes are identical besides their role which can be distinguished by an attribute.
* I have placed a unique restriction on item titles and user usernames. This was mainly done to prevent duplicates of the dummy data from being added to the tables.
* I found no real purpose in separating different formats into different tables, so I made a simple “format” attribute for an items table
* I ensured that a patron could not check out an already checked out item by adding a check to see if the borrower attribute is NULL
* The view all items and search items functions for both the librarian and patron operate off of the same py function as they are trying to accomplish the same thing
* I made sure to send feedback information for the add and checkout operations to let the user know the process has successfully gone through