<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login Page</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<div class="login-container">

<h1>Login</h1>

<form action="login.php" method="POST">

<label for="username">Username:</label><br>

<input type="text" id="username" name="username" required><br>

<label for="password">Password:</label><br>

<input type="password" id="password" name="password" required><br>

<button type="submit">Login</button>

</form>

</div>

</body>

</html>

\*\*\*\*

<?php

$name = $\_POST['name'];

$email = $\_POST['email'];

// Call Python script with arguments

$result = exec("python script.py $name $email");

echo "Python script output: $result";

?>

\*\*\*\*\*\*

from flask import Flask, request, render\_template, jsonify

import sqlite3

app = Flask(\_name\_)

DATABASE = 'database.db'

def init\_db():

with sqlite3.connect(DATABASE) as db:

cursor = db.cursor()

cursor.execute('''

CREATE TABLE IF NOT EXISTS users (

id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT NOT NULL,

email TEXT NOT NULL

)

''')

db.commit()

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/users', methods=['GET', 'POST'])

def users():

if request.method == 'GET':

print("GET request received for /users")

db = get\_db()

cursor = db.execute('SELECT \* FROM users')

users = cursor.fetchall()

db.close()

return jsonify(users)

elif request.method == 'POST':

data = request.json

name = data.get('name')

email = data.get('email')

print("POST request received for /users")

print(f"Name: {name}, Email: {email}")

db = get\_db()

db.execute('INSERT INTO users (name, email) VALUES (?, ?)', (name, email))

db.commit()

db.close()

return jsonify({'message': 'User added successfully'})

def get\_db():

db = sqlite3.connect(DATABASE)

return db

if \_name\_ == '\_main\_':

init\_db() # Initialize the database

app.run(debug=True)

\*\*\*\*\*\*\*

class Library:

def \_\_init\_\_(self, name):

self.name = name

self.books = {}

def add\_book(self, book\_id, title, author, quantity):

if book\_id in self.books:

self.books[book\_id]['quantity'] += quantity

else:

self.books[book\_id] = {'title': title, 'author': author, 'quantity': quantity}

def remove\_book(self, book\_id, quantity=1):

if book\_id in self.books:

if self.books[book\_id]['quantity'] <= quantity:

del self.books[book\_id]

else:

self.books[book\_id]['quantity'] -= quantity

else:

print("Book not found.")

def display\_books(self):

print("\nLibrary Catalog:")

for book\_id, details in self.books.items():

print(f"ID: {book\_id}, Title: {details['title']}, Author: {details['author']}, Quantity: {details['quantity']}")

def borrow\_book(self, book\_id):

if book\_id in self.books:

if self.books[book\_id]['quantity'] > 0:

self.books[book\_id]['quantity'] -= 1

print("Book borrowed successfully.")

else:

print("Book not available.")

else:

print("Book not found.")

def return\_book(self, book\_id):

if book\_id in self.books:

self.books[book\_id]['quantity'] += 1

print("Book returned successfully.")

else:

print("Book not found.")

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

library = Library("My Library")

library.add\_book(1, "Python Programming", "John Doe", 5)

library.add\_book(2, "Java Basics", "Jane Smith", 3)

library.add\_book(3, "Data Science for Beginners", "Alice Johnson", 2)

library.display\_books()

library.borrow\_book(1)

library.borrow\_book(2)

library.borrow\_book(1)

library.display\_books()

library.return\_book(1)

library.display\_books()