Creating a **Cafe Management System** as a mini-project can showcase your skills effectively. Here's a simplified version of the project using Python, HTML, CSS, JavaScript, and MySQL.

**Project Overview**

1. **Backend**: Python (Flask framework)
2. **Frontend**: HTML, CSS, JavaScript
3. **Database**: MySQL
4. **Functionality**:
   * Add and view menu items.
   * Place customer orders.
   * View order history.

**Step 1: Set Up the Environment**

1. Install Python dependencies:
2. pip install flask flask-mysql
3. Set up MySQL:
   * Create a database cafe\_db.
   * Run the following SQL script:
   * CREATE TABLE menu (
   * id INT AUTO\_INCREMENT PRIMARY KEY,
   * item\_name VARCHAR(50),
   * price DECIMAL(10, 2)
   * );
   * CREATE TABLE orders (
   * id INT AUTO\_INCREMENT PRIMARY KEY,
   * item\_name VARCHAR(50),
   * quantity INT,
   * total\_price DECIMAL(10, 2)
   * );

**Step 2: Backend Code (app.py)**

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

import mysql.connector

app = Flask(\_\_name\_\_)

# MySQL Connection

db = mysql.connector.connect(

host="localhost",

user="root",

password="your\_password",

database="cafe\_db"

)

cursor = db.cursor()

# Home Route

@app.route('/')

def home():

cursor.execute("SELECT \* FROM menu")

menu = cursor.fetchall()

return render\_template("index.html", menu=menu)

# Add Menu Item

@app.route('/add\_menu', methods=['POST'])

def add\_menu():

item\_name = request.form['item\_name']

price = request.form['price']

cursor.execute("INSERT INTO menu (item\_name, price) VALUES (%s, %s)", (item\_name, price))

db.commit()

return redirect('/')

# Place Order

@app.route('/place\_order', methods=['POST'])

def place\_order():

item\_name = request.form['item\_name']

quantity = int(request.form['quantity'])

cursor.execute("SELECT price FROM menu WHERE item\_name = %s", (item\_name,))

price = cursor.fetchone()[0]

total\_price = price \* quantity

cursor.execute("INSERT INTO orders (item\_name, quantity, total\_price) VALUES (%s, %s, %s)", (item\_name, quantity, total\_price))

db.commit()

return redirect('/orders')

# View Orders

@app.route('/orders')

def orders():

cursor.execute("SELECT \* FROM orders")

orders = cursor.fetchall()

return render\_template("orders.html", orders=orders)

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

app.run(debug=True)

**Step 3: Frontend Code**

**HTML (templates/index.html)**

<!DOCTYPE html>

<html>

<head>

<title>Cafe Management</title>

<link rel="stylesheet" href="/static/style.css">

</head>

<body>

<h1>Cafe Menu</h1>

<table>

<tr>

<th>Item</th>

<th>Price</th>

</tr>

{% for item in menu %}

<tr>

<td>{{ item[1] }}</td>

<td>{{ item[2] }}</td>

</tr>

{% endfor %}

</table>

<h2>Add Menu Item</h2>

<form action="/add\_menu" method="POST">

<input type="text" name="item\_name" placeholder="Item Name" required>

<input type="number" name="price" placeholder="Price" step="0.01" required>

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

</form>

<h2>Place Order</h2>

<form action="/place\_order" method="POST">

<input type="text" name="item\_name" placeholder="Item Name" required>

<input type="number" name="quantity" placeholder="Quantity" required>

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

</form>

<a href="/orders">View Orders</a>

</body>

</html>

**HTML (templates/orders.html)**

<!DOCTYPE html>

<html>

<head>

<title>Order History</title>

<link rel="stylesheet" href="/static/style.css">

</head>

<body>

<h1>Order History</h1>

<table>

<tr>

<th>Item</th>

<th>Quantity</th>

<th>Total Price</th>

</tr>

{% for order in orders %}

<tr>

<td>{{ order[1] }}</td>

<td>{{ order[2] }}</td>

<td>{{ order[3] }}</td>

</tr>

{% endfor %}

</table>

<a href="/">Back to Home</a>

</body>

</html>

**CSS (static/style.css)**

body {

font-family: Arial, sans-serif;

margin: 20px;

}

h1, h2 {

color: #333;

}

table {

width: 100%;

border-collapse: collapse;

margin-bottom: 20px;

}

table, th, td {

border: 1px solid #ddd;

}

th, td {

padding: 10px;

text-align: left;

}

form {

margin-bottom: 20px;

}

form input, form button {

padding: 10px;

margin-right: 10px;

}

a {

display: inline-block;

margin-top: 20px;

color: #0066cc;

text-decoration: none;

}

**How to Run**

1. Save the Python file as app.py.
2. Create the necessary directories: templates/ for HTML and static/ for CSS.
3. Start the Flask server:
4. python app.py
5. Open the app in your browser at http://127.0.0.1:5000/.

**Future Enhancements**

* Add user authentication.
* Implement a dashboard for daily sales reporting.
* Enhance UI with frameworks like Bootstrap.

Let me know if you need any further guidance!

Let’s enhance the **Cafe Management System** with the requested features: **user authentication**, a **daily sales dashboard**, and an improved UI using **Bootstrap**.

**1. User Authentication**

**Update the MySQL Database**

Add a users table for storing login credentials:

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE,

password VARCHAR(255)

);

Add an admin user (hashed password using bcrypt):

import bcrypt

# Hash password

password = "admin123"

hashed\_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())

# Insert admin user

cursor.execute("INSERT INTO users (username, password) VALUES (%s, %s)", ("admin", hashed\_password))

db.commit()

**Install Required Libraries**

pip install bcrypt flask-session

**Update app.py**

from flask import Flask, render\_template, request, redirect, session, url\_for

import bcrypt

from flask\_session import Session

app = Flask(\_\_name\_\_)

# Session Configuration

app.secret\_key = "your\_secret\_key"

app.config["SESSION\_TYPE"] = "filesystem"

Session(app)

# Login Route

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

def login():

if request.method == 'POST':

username = request.form['username']

password = request.form['password']

cursor.execute("SELECT \* FROM users WHERE username = %s", (username,))

user = cursor.fetchone()

if user and bcrypt.checkpw(password.encode('utf-8'), user[2].encode('utf-8')):

session['user'] = username

return redirect('/')

else:

return render\_template('login.html', error="Invalid credentials")

return render\_template('login.html')

# Logout Route

@app.route('/logout')

def logout():

session.pop('user', None)

return redirect('/login')

# Protect Routes

@app.before\_request

def require\_login():

allowed\_routes = ['login']

if request.endpoint not in allowed\_routes and 'user' not in session:

return redirect('/login')

**Login Template (templates/login.html)**

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

<h2 class="text-center">Cafe Management - Login</h2>

{% if error %}

<div class="alert alert-danger">{{ error }}</div>

{% endif %}

<form method="POST" class="mt-3">

<div class="mb-3">

<label for="username" class="form-label">Username</label>

<input type="text" name="username" class="form-control" id="username" required>

</div>

<div class="mb-3">

<label for="password" class="form-label">Password</label>

<input type="password" name="password" class="form-control" id="password" required>

</div>

<button type="submit" class="btn btn-primary w-100">Login</button>

</form>

</div>

</body>

</html>

**2. Daily Sales Dashboard**

**Backend Changes**

Add a route to fetch daily sales data:

@app.route('/dashboard')

def dashboard():

cursor.execute("SELECT item\_name, SUM(quantity), SUM(total\_price) FROM orders GROUP BY item\_name")

sales = cursor.fetchall()

return render\_template("dashboard.html", sales=sales)

**Dashboard Template (templates/dashboard.html)**

<!DOCTYPE html>

<html>

<head>

<title>Daily Sales Dashboard</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

<h2>Daily Sales Dashboard</h2>

<table class="table table-bordered mt-3">

<thead>

<tr>

<th>Item</th>

<th>Quantity Sold</th>

<th>Total Sales</th>

</tr>

</thead>

<tbody>

{% for sale in sales %}

<tr>

<td>{{ sale[0] }}</td>

<td>{{ sale[1] }}</td>

<td>₹{{ sale[2] }}</td>

</tr>

{% endfor %}

</tbody>

</table>

<a href="/" class="btn btn-primary">Back to Home</a>

</div>

</body>

</html>

**3. Enhance UI with Bootstrap**

**Update the Base Template (e.g., templates/index.html)**

Use Bootstrap classes for better styling:

<!DOCTYPE html>

<html>

<head>

<title>Cafe Management</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

</head>

<body>

<div class="container mt-5">

<h1 class="text-center">Cafe Menu</h1>

<table class="table table-striped mt-3">

<thead>

<tr>

<th>Item</th>

<th>Price</th>

</tr>

</thead>

<tbody>

{% for item in menu %}

<tr>

<td>{{ item[1] }}</td>

<td>₹{{ item[2] }}</td>

</tr>

{% endfor %}

</tbody>

</table>

<h2>Add Menu Item</h2>

<form action="/add\_menu" method="POST" class="mt-3">

<div class="mb-3">

<input type="text" name="item\_name" class="form-control" placeholder="Item Name" required>

</div>

<div class="mb-3">

<input type="number" name="price" class="form-control" placeholder="Price" step="0.01" required>

</div>

<button type="submit" class="btn btn-primary w-100">Add</button>

</form>

<h2>Place Order</h2>

<form action="/place\_order" method="POST" class="mt-3">

<div class="mb-3">

<input type="text" name="item\_name" class="form-control" placeholder="Item Name" required>

</div>

<div class="mb-3">

<input type="number" name="quantity" class="form-control" placeholder="Quantity" required>

</div>

<button type="submit" class="btn btn-success w-100">Order</button>

</form>

<a href="/orders" class="btn btn-info mt-3">View Orders</a>

<a href="/dashboard" class="btn btn-warning mt-3">View Dashboard</a>

</div>

</body>

</html>

**Final Notes**

* Test the project thoroughly to ensure all features work.
* Consider deploying it on a cloud platform like **Heroku** or **Render** for accessibility.

Let me know if you need help with deployment or further modifications!