Cafe project

Maktab 52

Project overview:

Technical Requirements:

- Python language
- Flask : Web server
- Git: Version control
- PostgreSql: Database
- HTML, CSS, JS : Front-end
- Bootstrap CSS framework

Description:

Create a Cafe website project with two major functionalities:

A. **Landing page**:

Public page of the cafe website that contains some sub-pages for customers view:

1. **Home:** Designed to attract customers. Must have keypoint information about the cafe and have good Graphic design
2. **About:** This section presents full information about the cafe, website, staff, ...
3. **Contact us:** contains Geographical information, Address, Phone number and other ways to communicate with the cafe.
   Also have a form to interact with users from the website.
   (optional) Add map view into the contact page.
4. **Menu**: List of products and foods to be served by the Cafe.
   The Menu section must contain:
   - **Products** along with their **prices**
   - **Order** button, to order.
5. **Orders list:** Show list of customer orders along with the status of them.
   e.g: Waiting, Cooking, Served , ...

B. **Cashier Panel**:

This panel is designed for **Cashiers**, to manage customer orders.

Cashier Panel must have features like:

1. **Dashboard:** provides shortcuts and important information for the cashier. Like: Received messages, Recent Orders, ...
2. **Orders:** List of orders along with their status and action buttons. You can change the status of orders by doing actions on them like Send them to the kitchen, Serve them and ... that effects on the status of orders.

   We can have several sub-pages for every status of orders like:

   - **New orders**: List of new orders from customers.
   - **Cooking orders**: List of orders that are sent to the kitchen.
   - **Served orders**: List of recent orders that are served to the customer.
   - **Deleted Orders**: List of rejected orders by the cashier.
   - **Paid Orders**: List of paid orders.
   - **Archive:** a complete history of customer orders.

   *** Use bootstrap modals to see details of orders**

   *** Each row of orders must have action buttons to change the status of the order.**

3. **Tables:** Map of the cafe tables along with their current orders. The cashier can see the order list of each table by clicking on them and doing some actions on them.

   He/She can also export **receipts** for each table from this page.
4. **Menu Items:** Manage Items(foods) that are served by the cafe. The cashier can Add, Edit or Delete every product from this page.

- **Add Items:** This section allows the cashier to add new items (foods) to the menu by their details like Name, Price, Category, Description, Discount(Optional) , ...
- **Edit Items:** Change products prices, set discount for each product, add description for them and ...
- **Delete Items:** Delete Items from the menu.

5. **Receipts:** List of exported receipts from the system.
   The cashier should have access to see, report and manage receipts.
   - **Receipt:** receipt details page.
     - **(optional):** implement **@media print**
   - **Paid Receipts:** paid receipts by customers.
   - **Archive:** History of receipts.
   - ...

6. **Charts (Optional):** Statistical charts containing information like:
   - Number of orders / hour
   - Number of orders / week days
   - Order Categories / hour
   - Order Categories / week days
   - ...

Phase 1#:

In this phase you should analyze your Business first, Then you can initialize your project by designing structures, basics and architecture.

Tasks

1. Git Configuration:

First, Create a **private** Git repository on Github.com.
Git name: Maktab52_Cafe_project.
Invite your Teachers to collaborate.

Teachers github username:

- **mohammadT77**
- **amirch72**
- **ArminNaCl**
- **shahinAbolqasemi**

* Make sure that your source code is Git integrated to prevent data loss.
* Benefit from GIT **branching.**

2. ERD:

You should store data using Relational Databases management systems into **PostgreSQl.**
In the first step, you should analyze your business entities and Design your **ERD.**

Cafe project Entities:

- **Users:**
  - Id
  - First name
  - Last name
  - Phone number
  - Email
  - Password to login with
  - Extra information such as birthday
  - ...
- **Tables:**
  - Id
  - Table number
  - Cafe space position
  - ...
- **MenuItems:**
  - Id
  - Name
  - Price
  - Category
  - Discount (Optional)
  - Serving time period (Optional)
  - Estimated cooking time (Optional)
  - ...
- **Orders:**
  - Id
  - Table
  - Menu Items (food)
  - Number
  - Status

    - Timestamp
    - ...
- *** Receipts:**
    - Id
    - Orders
    - Total price
    - Final price (with Discount)
    - Timestamp,
    - ...
- ....

Note:

- Analyze your project and extend your ERD fitting the business requirements.
- Extract attributes from entities and make them a standalone Entity as your project needs.
- **Receipt** entities have a **One-to-Many** relation with **Order** entities.
- **Category Tree (Optional):** You can create a hierarchical category database structure, using self-relations in SQL databases. Read about it.

After analyzing the database and designing ERD, implement them into **PostgreSQL Database**.

3. OOP Models:

Design your system OOP models that fit your database structure.
Each model in this project is a python class that represents a Database entity (table).

Considering database entities, you should develop Python classes (models) fitting the entities to

After developing models, Use **psycopg2** to store and fetch data from the database.
(Optional) Develop and Implement a **DatabaseManager** using psycopg to mapping Models on Database entities and providing four basic **CRUD** (Create, Read, Update, Delete) actions.

4. Front-End Design:

- First, Design all Web-pages using only HTML, CSS, Bootstrap.
  See several cafe templates like this and get inspired by them.
  Finally develop your own templates using only HTML, CSS and Bootstrap.
- Notice that the Landing page must be well-designed and customer attracting.
- The cashier admin page must be panel formed like this.
- All web pages must be **Fully-Responsive.**
- Use Bootstrap modals and other utilities (Tooltip, Toast, Inputs, ...)
- **(Optional)** Design a bilingual (English , Persian) landing page.

Submission:

- The deadline for submission of each phase of the project is **one week.**
- Git commit time, considered as submission time.
- all members of the group must have **equal responsibility** in developing the project since each member is graded individually and according to their cooperation in the team.