**Name:** Kowsalya K

**Roll\_No:** 21i324

**Department:** Information Technology

**College:** PSG College of Technology

**ICE CREAM PARLOR APPLICATION DOCUMENTATION**

**1. INTRODUCTION**

This Ice Cream Parlor Application is a Python-based solution that allows the management of seasonal ice cream flavors, ingredients, customer suggestions, and shopping carts using an SQLite3 database. Users can interact with the application to add, update, and manage various records in the system, including products and inventory.

**2. PROJECT STRUCTURE**

The project consists of the following files:

* **main.py**: The main script containing the user interface and interaction logic.
* **database.py**: Handles database connections and table creation.
* **models.py**: Defines the models for managing ice cream flavors, ingredients, customer suggestions, and cart items.
* **requirements.txt**: Lists the required Python packages for the project.
* **Docker file**: Defines the steps to create a Docker container for running the application.
* **ReadMe.md**: Provides the setup and run instructions.

**3. APPLICATION WORKFLOW**

**3.1 Database Connection and Setup**

When the application starts, it checks if the SQLite database ice\_cream.db exists. If not, it is created, and the following tables are initialized:

1. **flavors**: Stores information about ice cream flavors (e.g., name, description, seasonal status).
2. **ingredients**: Stores ingredients available for making ice cream (e.g., name, stock level).
3. **suggestions**: Stores customer suggestions regarding flavors and allergy concerns.
4. **cart**: Tracks items added to the cart for purchase (e.g., product ID, quantity).

**3.2 User Interface**

The application is text-based and prompts the user with options to interact with the system.

**Menu options**:

1. **Add Flavor**:
   * User can add a new ice cream flavor by entering its name, description, and seasonal status (yes/no).
2. **Add Ingredient**:
   * User can add new ingredients (like milk, sugar) to the inventory and specify the stock level.
3. **Add Customer Suggestion**:
   * Users can suggest a new flavor and report any allergy concerns related to it.
4. **Add to Cart**:
   * Users can add a specific product (flavor) to their cart with a given quantity.
5. **Exit**:
   * Exit the application.

Each user action is followed by a confirmation message. If the user selects an invalid option, the application will prompt them again.

**4. CODE EXPLANATION**

**4.1 Database Functions (database.py)**

The database module contains functions for creating a connection to the SQLite database and creating the necessary tables if they do not already exist.

**Key Functions**:

* **create\_connection()**: Establishes a connection to the SQLite database (ice\_cream.db).
* **create\_tables()**: Creates the required tables for the application (flavors, ingredients, suggestions, cart).

**4.2 Model Classes (models.py)**

The model classes define the operations that interact with the database to insert new records.

**Key Classes**:

1. **Flavor**:
   * Adds new flavors to the flavors table.
   * **Method**: add\_flavor(name, description, is\_seasonal)
2. **Ingredient**:
   * Adds ingredients to the ingredients table.
   * **Method**: add\_ingredient(name, stock\_level)
3. **Suggestion**:
   * Adds customer suggestions and allergy concerns to the suggestions table.
   * **Method**: add\_suggestion(flavor\_name, allergy\_concerns)
4. **Cart**:
   * Adds items to the cart in the cart table.
   * **Method**: add\_to\_cart(product\_id, quantity)

Each method executes an SQL INSERT statement to add records to the appropriate table.

**4.3 Main Application (main.py)**

The main script serves as the entry point to the application. It displays the menu options, accepts user input, and invokes the appropriate methods from the models.py to manipulate the database.

**Key Steps**:

* Displays the menu for the user to choose actions.
* Calls corresponding methods (from models.py) to perform actions like adding flavors, ingredients, etc.
* Continues to loop, allowing multiple actions until the user chooses to exit.

**5. EXAMPLE WORKFLOW**

**Step 1: Add a Flavor**

* User selects option 1 (Add Flavor).
* The application prompts the user to enter the flavor's name, description, and whether it is seasonal.
* After the input, the flavor is saved in the flavors table.

**Step 2: Add an Ingredient**

* User selects option 2 (Add Ingredient).
* The user is asked for the ingredient's name and stock level.
* The ingredient is saved in the ingredients table.

**Step 3: Add a Customer Suggestion**

* User selects option 3 (Add Customer Suggestion).
* The application prompts for the suggested flavor and any allergy concerns.
* The suggestion is added to the suggestions table.

**Step 4: Add to Cart**

* User selects option 4 (Add to Cart).
* The user enters the product ID and quantity.
* The item is added to the cart table.

**Step 5: Exit**

* User selects option 5 (Exit).
* The application ends.

**6. RUNNING THE APPLICATION**

1. **Clone or Download the Repository**:
   * Clone the repository or download the application files to your local machine.
2. **Set Up the Environment**:
   * Install Python 3.x (if not already installed).
   * Create a virtual environment and install dependencies using the command:
3. **Run the Application**:
   * Execute the Python script:
4. **Access the Application**:
   * Follow the on-screen instructions to interact with the system.

**7. CONCLUSION**

The Ice Cream Parlor Application is a simple yet effective system for managing ice cream flavors, ingredients, and customer interactions using an SQLite database. This documentation serves as a guide to understanding the structure, functionality, and usage of the application.

**8. FUTURE IMPROVEMENTS**

* Implementing more advanced inventory management.
* Adding user authentication and cart persistence.
* Providing a graphical user interface (GUI) for better user experience.