CAPSTONE PROJECT-1

PHARMACY DATABASE PROJECT

Here are the step-by-step instructions to implement and run the provided Python script for managing medicine details in a pharmacy system:

Step-by-Step Instructions

1.Set Up Your Environment:

* Ensure you have Python installed on your system.

* Install the necessary libraries, such as pymysql-connector-python, if not already installed. You can install it using:

```
pip install pymysql-connector-python
```

2.Establish Database Connection:

* Before running the script, ensure you have a MySQL database set up with a table named Medicine. The table should have columns for medicine_name, category, price, and stock.

* Establish a connection to your MySQL database. Add the following code at the beginning of your script to connect to the database:

```
import pymysql.connector

connection = pymysql.connector.connect(
  host="your_host",
  user="your_username",
  password="your_password",
  database="your_database"
)

cursor = connection.cursor()
```

3.Define Functions:

- Copy the provided functions into your script. These functions include:
- add_medicine(): To add new medicine details.
- display_medicine(): To display all medicine details.
- list_medicine_by_category(): To list medicines by category.
- count_medicine(): To count the total number of medicines.

4.Create the Main Menu:

* Implement the main menu function to provide options for the user to interact with the system:

```
def main():
  while True:
    print("\nApollo Pharmacy System")
    print("1. Add medicine details")
    print("2. Display medicine details")
    print("3. List medicine by category")
    print("4. Count the number of medicines")
    print("5. Exit")
    choice = input("Enter your choice: ")

    if choice == '1':
      add_medicine()
    elif choice == '2':
      display_medicine()
    elif choice == '3':
      list_medicine_by_category()
    elif choice == '4':
      count_medicine()
    elif choice == '5':
      break
    else:
      print("Invalid choice! Please try again.")
```

5.Run the Script:

* Ensure the script includes the following lines to start the main menu and close the database connection when done:

Python

```
if __name__ == "__main__":
    main()
    # Closing the connection
    connection.close()
```

6.Execute the Script:

* Run the script in your Python environment. You can do this by saving the script to a file (e.g., pharmacy_system.py) and running it from the command line:

```
python pharmacy_system.py
```

7.Interact with the System:

* Follow the on-screen prompts to add, display, list, and count medicine details.

Example Usage

- Adding Medicine:
- Enter Medicine_Name: Paracetamol
- Enter the Category: Painkiller
- Enter the price: 50
- Enter the stock: 100
- Medicine added successfully!

- Displaying Medicines:

```
Medicine_Name: Paracetamol, Category: Painkiller, price: 50, stock: 100
```

- Listing Medicines by Category:

  ```
  Enter category Name: Painkiller

  Medicine_Name: Paracetamol, Category: Painkiller, price: 50, stock: 100
  ```

- Counting Medicines:

  ```
  Total number of medicines: 1
  ```