# dav-clg-logo.pngMRA DAV Public School

# (BY PASS ROAD, Solan)

# HIMACHAL PRADESH

****

**INFORMATICS PRACTICES**

**Class-XII**

PROJECT REPORT

PROJECT BY:

NAME: MANN UPADHYAY

ROLL NO: 25

**CERTIFICATE**

This is to certify that this Project Report on Hospital Management System using Python and MySQL is submitted by Mann Upadhyay (XII-Alpha) (R.No.25) to the Computer Department of MRA DAV PUBLIC SCHOOL, SOLAN, Himachal Pradesh, carried out by her/him towards partial completion of Practical Exam for class XII during academic year 2023-2024.

# **Internal⠀Examiner External⠀Examiner H.O.D PRINCIPAL**

**AKNOWLEDMENT**

I would want to convey my gratitude to everyone who has assisted me in finishing my assignment successfully.

First and first, I want to thank CBSE from the bottom of my heart for giving me such a wonderful opportunity to develop a project and learn more about this fascinating subject.

Second, I want to express my gratitude to our school’s principal, Ms. Masooma Singha, for providing us with the inspiration and amazing support we needed to finish the project.

Thirdly, I would like to thank Mr. Rajesh, my IP teacher, who guided me through every step of the Project Report preparation.

Finally, I’d want to thank everyone who has helped me along the way, including my teachers, parents, and whose support has made this effort possible

**STUDENT PROFILE**

|  |  |  |
| --- | --- | --- |
| **Name:** | **:** | Mann Upadhyay |
| **Date of Birth:** | **:** | 28 August 2006 |
| **Father Name:** | **:** | Mr. Rohit |
| **Mother Name:** | **:** | Mrs. Chandan Dhar |
| **Class:** | **:** | 12th |
| **Section:** | **:** | Alpha |
| **Roll No:** | **:** | 25 |
| **Address:** | **:** | C/O Sh. Deepak Thakur, Near Rajeshwari Temple, Kandaghat, Distt- Solan, Himachal Pradesh, 173215 |
| **Hobbies:** | **:** | Drawing Sketches, Coding and Listening Music |

**PROJECT REPORT ON HOSPITAL MANAGEMENT SYSTEM USING PYTHON AND MYSQL**

Table of Contents

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topic** | **Page No** |
| 1 | General Introduction of the Project | 6 |
| 2 | Hardware and Software Requirements | 7 |
| 3 | Python Code for the Project | 8-21 |
| 4 | MySQL used for the Project | 22-28 |
| 5 | Output of the Project | 29-36 |
| 6 | Further scopes of the Project | 37 |
| 7 | Bibliography | 38 |

### GENERAL INTRODUCTION OF THE PROJECT

### Hospital Management project is developed using python and

Mysql. In this project there are three categories, that can use this project:

* **Admin** : Can create new tables, updated records, enter records, browse the data, search using wildcard.
* **Doctor**: Can browse records, update records
* **Patient**: Can buy the medicine, check his/her disease history, browse his/her detail.



### INTRODUCTION TO PYTHON & MYSQL

### HARDWARE REQUIREMENT SOFTWARE REQUIREMENT

#### Python is a programming language that lets you work more quickly and integrate your systems more effectively.

Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,  
built on top of the [Python](https://www.python.org/) programming language.

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

MySQL is the world’s most popular open source database

This project is developed on following hardware and software requirements:

A) Hardware Requirements:

* Processor: Intel(R) Core(TM) i3 CPU

@2,40GHz

* Installed RAM: 4.00 GB
* System type: 64-bit operating system
* Operating System: MS Windows 10

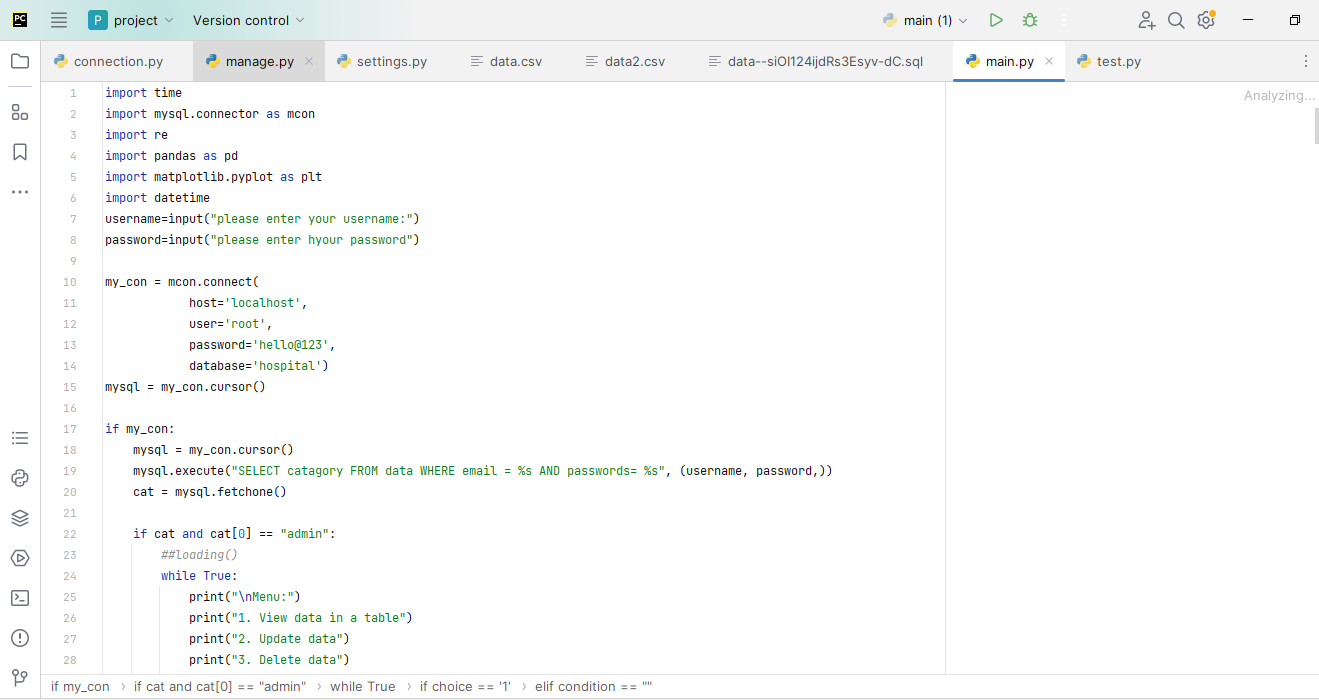
B) Software Requirements:

* PyCharm
* MySQL workbench and phpmyadmin

**PYTHON CODE FOR PROJECT**



**Figure 1: PyCharm tool is used for developing the application**

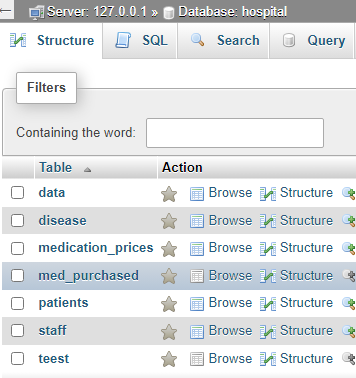


**Figure 2: Code in python in PyCharm**

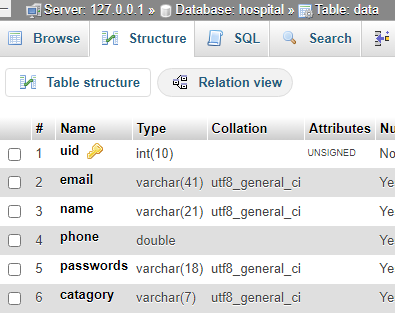
**Python Code:**

import time  
import mysql.connector as mcon  
import re  
import pandas as pd  
import matplotlib.pyplot as plt  
import datetime  
username=input("please enter your username:")  
password=input("please enter hyour password")  
  
my\_con = mcon.connect(  
 host='localhost',  
 user='root',  
 password='',  
 database='hospital')  
mysql = my\_con.cursor()  
  
if my\_con:  
 mysql = my\_con.cursor()  
 mysql.execute("SELECT catagory FROM data WHERE email = %s AND passwords= %s", (username, password,))  
 cat = mysql.fetchone()  
  
 if cat and cat[0] == "admin":  
 *##loading()* while True:  
 print("\nMenu:")  
 print("1. View data in a table")  
 print("2. Update data")  
 print("3. Delete data")  
 print("4. Create a new table")  
 print("5. Delete a table")  
 print("6. Alter a table")  
 print("7. Search for wildcard character")  
 print("8. Graphs")  
 print("9. Exit")  
  
 choice = input("Enter your choice: ")  
  
 if choice == '1':  
 mysql.execute("SHOW TABLES")  
 tables = mysql.fetchall()  
  
 print("Available Tables:")  
 for i, table in enumerate(tables, start=1):  
 print(f"{i}. {table[0]}")  
  
  
 table\_choice = int(input("Select a table (enter the number): "))  
 if 1 <= table\_choice <= len(tables):  
 tbname = tables[table\_choice - 1][0]  
 else:  
 print("Invalid choice. Please select a valid table.")  
  
 mysql.execute(f"SHOW COLUMNS FROM {tbname}")  
 columns = mysql.fetchall()  
  
 print(f"Columns in {tbname}:")  
 for i, column in enumerate(columns, start=1):  
 print(f"{i}. {column[0]}")  
 clminp = input("Enter the names of the columns you want (comma-separated): ")  
 if clminp=="":  
 clminp="\*"  
 condition = input("Enter the condition (e.g., 'column\_name = value'): ")  
 order\_direction = input("Enter 'ASC' for ascending or 'DESC' for descending: ")  
 order\_direction = order\_direction.upper()  
 ordcl= columns[0]  
 ordcl=ordcl[0]  
  
 if order\_direction not in ["ASC", "DESC"]:  
 order\_direction = "ASC"  
  
 if condition == "" and clminp == "":  
 sql\_query = "SELECT \* FROM " + tbname + " ORDER BY "+ordcl +" "+ order\_direction + ";"  
 elif condition == "":  
 sql\_query = "SELECT " + clminp + " FROM " + tbname + " ORDER BY "+ordcl +" "+ order\_direction + ";"  
 elif clminp == "":  
 sql\_query = "SELECT \* FROM " + tbname + " WHERE " + condition + " ORDER BY "+ordcl +" "+ order\_direction + ";"  
 else:  
 sql\_query = "SELECT " + clminp + " FROM " + tbname + " WHERE " + condition + " ORDER BY "+ordcl +" "+ order\_direction + ";"  
  
 print(sql\_query)  
  
 try:  
 mysql.execute(sql\_query)  
 result = mysql.fetchall()  
  
 if result:  
 print("Selected Data:")  
 for row in result:  
 print(row)  
 else:  
 print("No data matching the condition.")  
 except mcon.Error as err:  
 print(f"Error executing SELECT query: {err}")  
  
  
  
 elif choice == '2':  
 mysql.execute("SHOW TABLES")  
 tables = mysql.fetchall()  
  
 print("Available Tables:")  
 for i, table in enumerate(tables, start=1):  
 print(f"{i}. {table[0]}")  
  
 table\_choice = int(input("Select a table (enter the number): "))  
 if 1 <= table\_choice <= len(tables):  
 tbname = tables[table\_choice - 1][0]  
 else:  
 print("Invalid choice. Please select a valid table.")  
 condition = input("Enter the condition (e.g., 'column\_name = value'): ")  
 new\_data = input("Enter the new data (e.g., 'column\_name = new\_value'): ")  
  
 *# Construct the SQL query* sql\_query = "UPDATE "+ tbname +" SET "+ new\_data +" WHERE "+ condition +";"  
  
 try:  
 mysql.execute(sql\_query)  
 mysql.connection.commit()  
 print("Data updated successfully.")  
  
 except mcon.Error as err:  
 print(f"Error updating data: {err}")  
  
 elif choice == '3':  
 *##loading()* mysql.execute("SHOW TABLES")  
 tables = mysql.fetchall()  
  
 print("Available Tables:")  
 for i, table in enumerate(tables, start=1):  
 print(f"{i}. {table[0]}")  
  
 table\_choice = int(input("Select a table (enter the number): "))  
 if 1 <= table\_choice <= len(tables):  
 tbname = tables[table\_choice - 1][0]  
 else:  
 print("Invalid choice. Please select a valid table.")  
 condition = input("Enter the condition (e.g., 'column\_name = value'): ")  
 sql\_query = "DELETE FROM "+tbname+" WHERE "+condition+";"  
  
 try:  
 mysql.execute(sql\_query)  
 mysql.connection.commit()  
 print("Data deleted successfully.")  
  
 except mcon.Error as err:  
 print(f"Error deleting data: {err}")  
  
 elif choice == '4':  
  
 tbname = input("Enter the name of the new table: ")  
 num\_columns = int(input("Enter the number of columns: "))  
  
 columns = []  
 for i in range(num\_columns):  
 column\_name = input(f"Enter name for column {i + 1}: ")  
 column\_type = input(f"Enter data type for column {i + 1}: ")  
 columns.append(f"{column\_name} {column\_type}")  
  
 *# Construct the SQL query* sql\_query = f"CREATE TABLE {tbname} ({', '.join(columns)});"  
  
 try:  
 mysql.execute(sql\_query)  
 print(f"Table '{tbname}' created successfully.")  
  
 except mcon.Error as err:  
 print(f"Error creating table: {err}")  
 elif choice == '5':  
 table\_name = input("Enter the name of the table to delete: ")  
  
 *# Construct the SQL query* sql\_query = f"DROP TABLE {table\_name};"  
  
 try:  
 mysql.execute(sql\_query)  
 print(f"Table '{table\_name}' deleted successfully.")  
  
 except mcon.Error as err:  
 print(f"Error deleting table: {err}")  
  
 elif choice == '6':  
 mysql.execute("SHOW TABLES")  
 tables = mysql.fetchall()  
  
 print("Available Tables:")  
 for i, table in enumerate(tables, start=1):  
 print(f"{i}. {table[0]}")  
  
 table\_choice = int(input("Select a table (enter the number): "))  
 if 1 <= table\_choice <= len(tables):  
 tbname = tables[table\_choice - 1][0]  
 else:  
 print("Invalid choice. Please select a valid table.")  
  
 print("Available options for altering the table:")  
 print("1. edit column")  
 print("2. Delete column")  
 option = input("Enter your choice: ")  
  
 if option == '1':  
 column\_name = input("Enter the name of the new column: ")  
 column\_type = input("Enter data type for the new column: ")  
  
 *# Construct the SQL query* sql\_query = f"ALTER TABLE {tbname} ADD COLUMN {column\_name} {column\_type};"  
  
 elif option == '2':  
 mysql.execute(f"SHOW COLUMNS FROM {tbname}")  
 columns = mysql.fetchall()  
 print(columns)  
 old\_column\_name = input("Enter the name of the column to modify: ")  
 new\_column\_name = input("Enter the new name for the column: ")  
 new\_column\_type = input("Enter the new data type for the column: ")  
  
 *# Construct the SQL query* sql\_query = f"ALTER TABLE {tbname} CHANGE COLUMN {old\_column\_name} {new\_column\_name} {new\_column\_type};"  
  
 elif option == '3':  
 column\_name = input("Enter the name of the column to delete: ")  
  
 *# Construct the SQL query* sql\_query = f"ALTER TABLE {tbname} DROP COLUMN {column\_name};"  
  
 else:  
 print("Invalid option.")  
  
  
 try:  
 mysql.execute(sql\_query)  
 print(f"Table '{tbname}' altered successfully.")  
  
 except mcon.Error as err:  
 print(f"Error altering table: {err}")  
  
  
 elif choice == '7':  
 wildcard = input("Enter the character you want to search for: ")  
 mysql.execute("SHOW TABLES")  
 tables = [table[0] for table in mysql.fetchall()]  
  
 for tbname in tables:  
 *# Get a list of all columns in the table.* mysql.execute(f"SHOW COLUMNS FROM {tbname}")  
 columns = [column[0] for column in mysql.fetchall()]  
  
 for column in columns:  
 *# Construct and execute a query to search for the wildcard character in the column.* query = f"SELECT \* FROM {tbname} WHERE {column} LIKE %s"  
 params = (f"%{wildcard}%",) *# Add '%' before and after the wildcard character.* mysql.execute(query, params)  
 results = mysql.fetchall()  
  
 if results:  
 print(f"Table: {tbname}, Column: {column}")  
 print("Matching Rows:")  
 for row in results:  
 print(row)  
 elif choice=='8':  
  
 print("Possible graphs:"  
 "1. Comparison of prices of medicines"  
 "2. Observed diseases"  
 "3. Exit")  
 choice = int(input("Enter your choice (1/2/3): "))  
  
 if choice == 1:  
 mysql.execute("SELECT NAME, Price FROM medication\_prices")  
 data = mysql.fetchall()  
 if data:  
 df = pd.DataFrame(data, columns=["Medication", "Price"])  
 df.plot.bar(x="Medication", y="Price", title="Comparison of Medication Prices")  
 plt.show()  
 else:  
 print("No data found for medication prices.")  
  
 elif choice == 2:  
 while True:  
 lst = []  
 mysql.execute("select diseases from patients")  
 y = mysql.fetchall()  
 df = pd.DataFrame({"": y})  
 df.columns = ["diseases"]  
 print(df)  
 x = 0  
 while len(df) != x:  
 y = df.iloc[x, 0]  
 input\_tuple = y  
 *# Extract the string from the tuple* input\_str = input\_tuple[0]  
  
 *# Split the string into a list of substrings using ',' as the delimiter* integers\_as\_strings = input\_str.split(',')  
  
 *# Convert each substring to an integer and print it* for num\_str in integers\_as\_strings:  
 lst.append(int(num\_str))  
 x += 1  
 df = pd.DataFrame({"dis": lst})  
  
 df = df['dis'].value\_counts().reset\_index()  
 mysql.execute("SELECT scientific\_name FROM disease LIMIT 10;")  
 namedis = mysql.fetchall()  
 *# Step 8: Plot the disease names and their counts* print(df)  
 plt.bar(df['dis'], df['count'])  
 plt.ylim(300)  
  
 *# Show the plot* plt.show()  
 elif choice == 3:  
 pass  
 else:  
 print("it appears the data you entered is wrong,kindly re-enter it")  
 elif choice == '9':  
 my\_con.close()  
 print("Exiting the program.")  
 break  
 else:  
 print("it appears the data you entered is wrong,kindly re-enter it")  
  
  
 elif cat and cat[0] == "doctors":  
 while True:  
 print("\nMenu:")  
 print("1. View data in a table")  
 print("2. Update data")  
 print("3. Exit")  
  
 choice = int(input("Enter your choice: "))  
  
 if choice == '1':  
 mysql.execute("SHOW TABLES")  
 tables = mysql.fetchall()  
  
 print("Available Tables:")  
 for i, table in enumerate(tables, start=1):  
 print(f"{i}. {table[0]}")  
  
 table\_choice = int(input("Select a table (enter the number): "))  
 if 1 <= table\_choice <= len(tables):  
 tbname = tables[table\_choice - 1][0]  
 else:  
 print("Invalid choice. Please select a valid table.")  
  
 mysql.execute(f"SHOW COLUMNS FROM {tbname}")  
 columns = mysql.fetchall()  
  
 print(f"Columns in {tbname}:")  
 for i, column in enumerate(columns, start=1):  
 print(f"{i}. {column[0]}")  
 clminp = input("Enter the names of the columns you want (comma-separated): ")  
 if clminp == "":  
 clminp = "\*"  
 condition = input("Enter the condition (e.g., 'column\_name = value'): ")  
 order\_direction = input("Enter 'ASC' for ascending or 'DESC' for descending: ")  
 order\_direction = order\_direction.upper()  
 if order\_direction not in ["ASC", "DESC"]:  
 order\_direction = "ASC"  
 if condition == "" and clminp == "":  
 sql\_query = "SELECT \* FROM " + tbname + " ORDER BY " + order\_direction + ";"  
 elif condition == "":  
 sql\_query = "SELECT " + clminp + " FROM " + tbname + "+ ORDER BY " + order\_direction + ";"  
 elif clminp == "":  
 sql\_query = "SELECT \* FROM " + tbname + "WHERE" + condition + " ORDER BY " + order\_direction + ";"  
 else:  
 sql\_query = "SELECT " + clminp + " FROM " + tbname + " WHERE " + condition + " ORDER BY " + order\_direction + ";"  
 print(sql\_query)  
  
 try:  
 mysql.execute(sql\_query)  
 result = mysql.fetchall()  
  
 if result:  
 print("Selected Data:")  
 for row in result:  
 print(row)  
 else:  
 print("No data matching the condition.")  
  
 except mcon.Error as err:  
 print(f"Error executing SELECT query: {err}")  
  
  
 elif choice == '2':  
 mysql.execute("SHOW TABLES")  
 tables = mysql.fetchall()  
  
 print("Available Tables:")  
 for i, table in enumerate(tables, start=1):  
 print(f"{i}. {table[0]}")  
  
 table\_choice = int(input("Select a table (enter the number): "))  
 if 1 <= table\_choice <= len(tables):  
 tbname = tables[table\_choice - 1][0]  
 else:  
 print("Invalid choice. Please select a valid table.")  
 condition = input("Enter the condition (e.g., 'column\_name = value'): ")  
 new\_data = input("Enter the new data (e.g., 'column\_name = new\_value'): ")  
  
 *# Construct the SQL query* sql\_query = "UPDATE " + tbname + " SET " + new\_data + " WHERE " + condition + ";"  
  
 try:  
 mysql.execute(sql\_query)  
 mysql.connection.commit()  
 print("Data updated successfully.")  
  
 except mcon.Error as err:  
 print(f"Error updating data: {err}")  
  
 elif choice == '3':  
 my\_con.close()  
 print("Exiting the program.")  
 break  
 else:  
 print("it appears the data you entered is wrong,kindly re-enter it")  
  
 elif cat and cat[0] == "user":  
 while True:  
 print("\nMenu:")  
 print("1.view personal data")  
 print("2.view diseases encountered by now")  
 print("3.purchase medicines")  
 print("4.exit")  
  
 choice = int(input("Enter your choice: "))  
 if choice == 1:  
  
 try:  
 sql = "SELECT \* FROM patients WHERE email = %s"  
 params = (username,)  
 mysql.execute(sql, params)  
 result = mysql.fetchall()  
  
 if result:  
 print("Selected Data:")  
 for row in result:  
 print(row)  
 else:  
 print("No data matching the condition.")  
  
 except mcon.Error as err:  
 print(f"Error executing SELECT query: {err}")  
  
  
  
 if choice == 2:  
 *##loading()* mysql.execute("SELECT diseases FROM patients WHERE email = %s", (username,))  
  
 *# Fetch the results* myresult = mysql.fetchall()  
  
 *# Print the diagnosed diseases and recommended medicines* print("You have been diagnosed with the following diseases (with the names of recommended medicines):")  
 for row in myresult:  
 diseases = row[0].split(",") *# Split diseases if they are comma-separated* for disease in diseases:  
 disease = disease.strip() *# Remove leading/trailing spaces (like trim)* disease\_id = f"d\_{disease.replace(' ', '\_')}" *# Create the disease ID* disease\_id = disease\_id.replace(",", "") *# Remove commas from disease ID  
 # Execute a query to fetch details of the disease and recommended medicines* mysql.execute("SELECT \* FROM disease WHERE disease\_id = %s", (disease\_id,))  
 disease\_info = mysql.fetchall()  
  
 *# Print disease information* if disease\_info:  
 print(  
 f"Disease: {disease\_info[0][1]}") *# f for strings and [0][1]Means first row 3rd coloumn* print(  
 f"Recommended Medicines: {disease\_info[0][2], disease\_info[0][3], disease\_info[0][4]}")  
 print()  
  
  
 if choice == 3:  
 print("Available medicines:")  
 mysql.execute("SELECT \* FROM medication\_prices")  
 med = mysql.fetchall()  
 for item in med:  
 print(f"ID: {item[3]}, Name: {item[0]}, Price: ${item[1]}, Quantity: {item[2]}")  
 mysql.execute("SELECT user\_id FROM patients WHERE email = %s", (username,))  
 userid = mysql.fetchall()  
 userid= userid[0]  
 userid = userid[0]  
 item\_id = int(input("Enter the ID of the med you want to buy: "))  
 quantity = int(input("Enter the quantity you want to buy: "))  
 dt = datetime.datetime.now()  
 mysql.execute("SELECT name, price, qty FROM medication\_prices WHERE med\_id = %s", (item\_id,))  
 item = mysql.fetchone()  
  
  
 if item:  
 item\_name, item\_price, item\_quantity = item  
 if item\_quantity >= quantity:  
 total\_cost = item\_price \* quantity  
 print(f"Item: {item\_name}, Quantity: {quantity}, Total Cost: ${total\_cost}")  
  
 confirm = input("Confirm purchase (yes/no): ").strip().lower()  
 if confirm == "yes":  
 *# Deduct the purchased quantity from the item's quantity* mysql.execute("UPDATE medication\_prices SET qty = qty - %s WHERE med\_id = %s",(quantity, item\_id))  
 print(userid, dt, item\_id, quantity)  
  
 sql = "INSERT INTO med\_purchased (uid, date, med\_id, qty\_pur) VALUES (%s, %s, %s, %s)"  
  
  
 params = (userid, dt, item\_id, quantity)  
  
  
 mysql.execute(sql, params)  
 print("Purchase successful!")  
 else:  
 print("Purchase canceled.")  
 else:  
 print("Insufficient quantity available.")  
 else:  
 print("Item not found.")  
  
 elif choice == 4:  
 pass  
 else:  
 print("it appears the data you entered is wrong,kindly re-enter it")  
  
 else:  
 print("It seems that an unexpected error has occurred. Please inform the staff about it.")

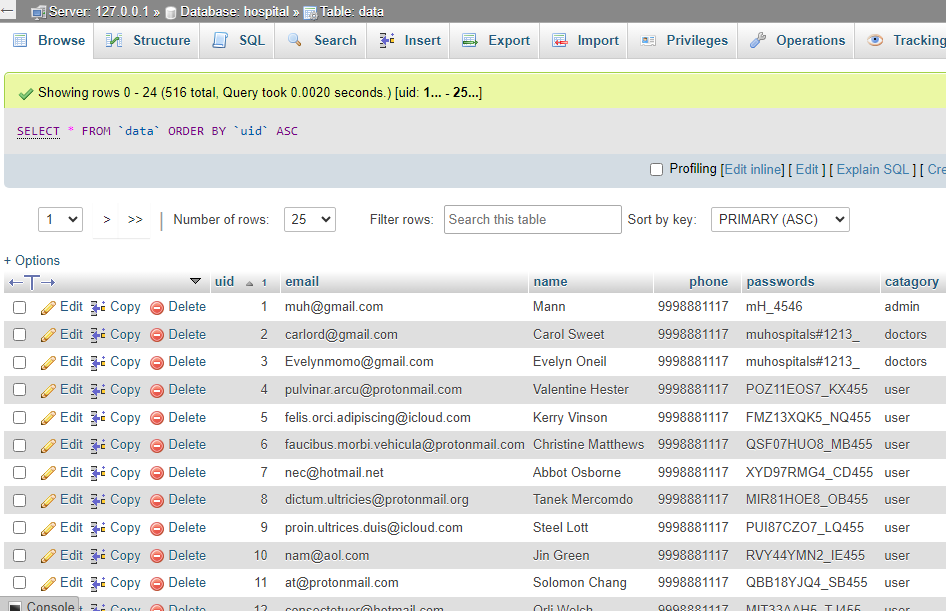
**MySQL USED FOR THE PROJECT**

****

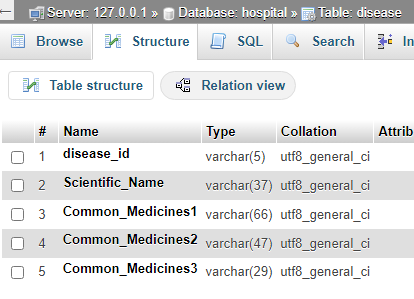
**Figure 3 : Created Database with Name Hospital**

****

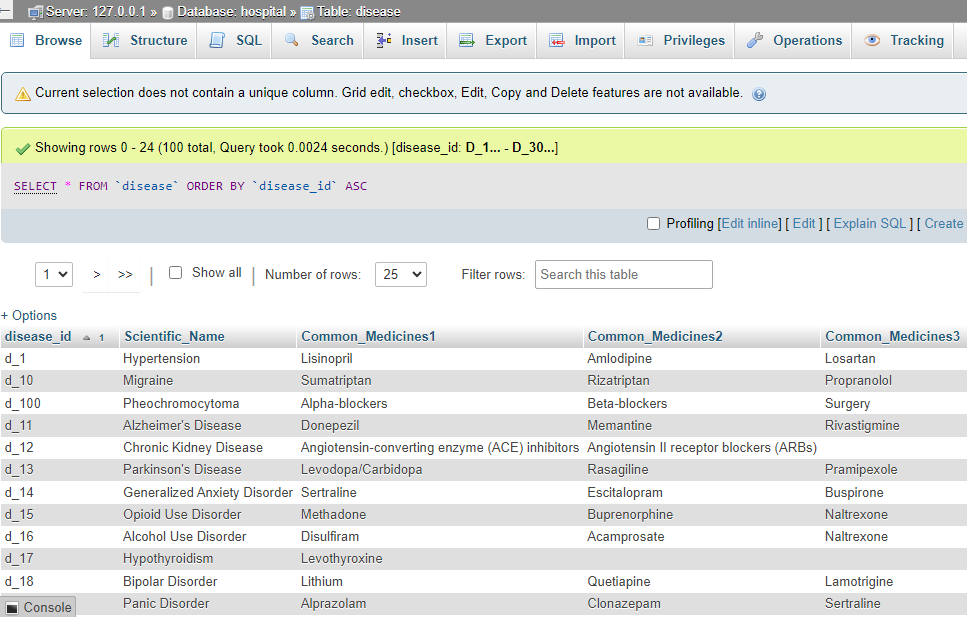
**Figure 4 : Created Table with Name data**

****

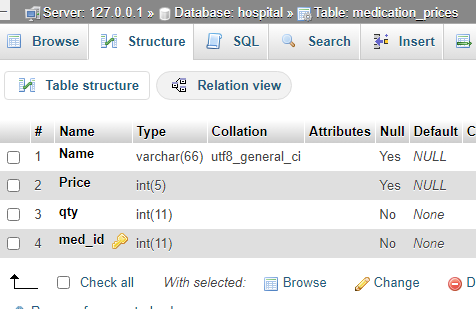
**Figure 5 : Inserted rows in table data**

****

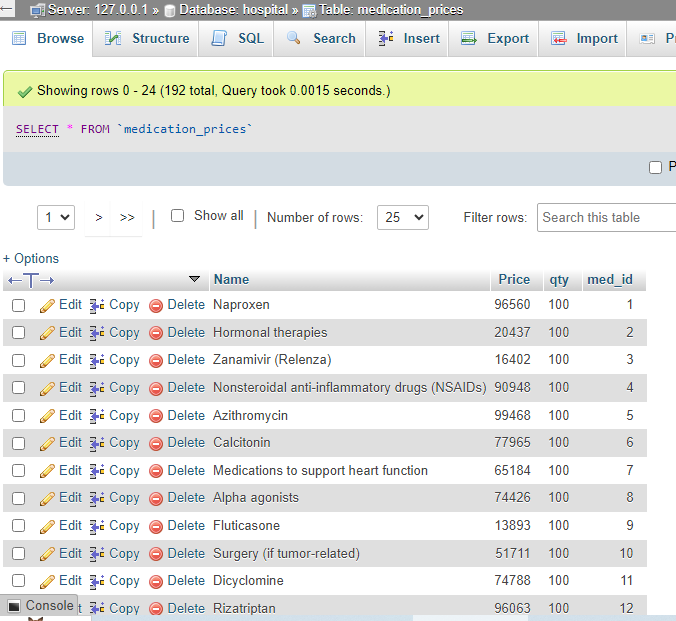
**Figure 6 : Created Table with Name disease**

****

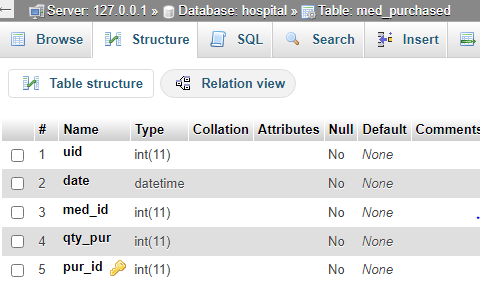
**Figure 7 : Inserted rows in table disease**

****

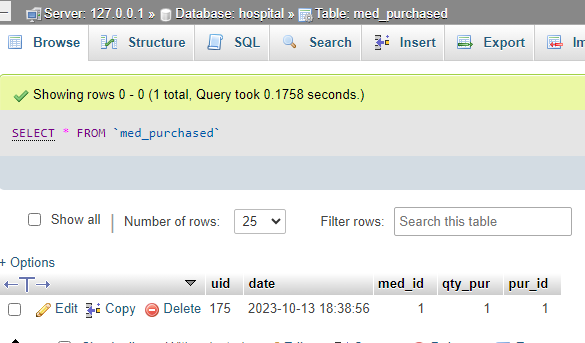
**Figure 8 : Created Table with Name medication\_prices**

****

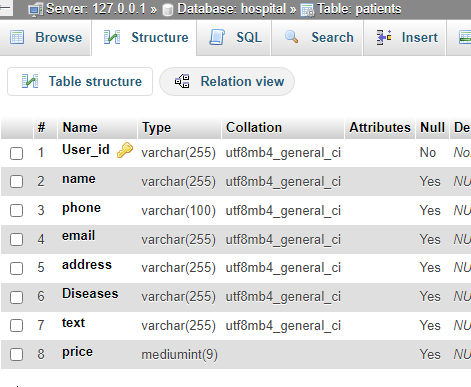
**Figure 9 : Inserted rows in table medication\_prices**

****

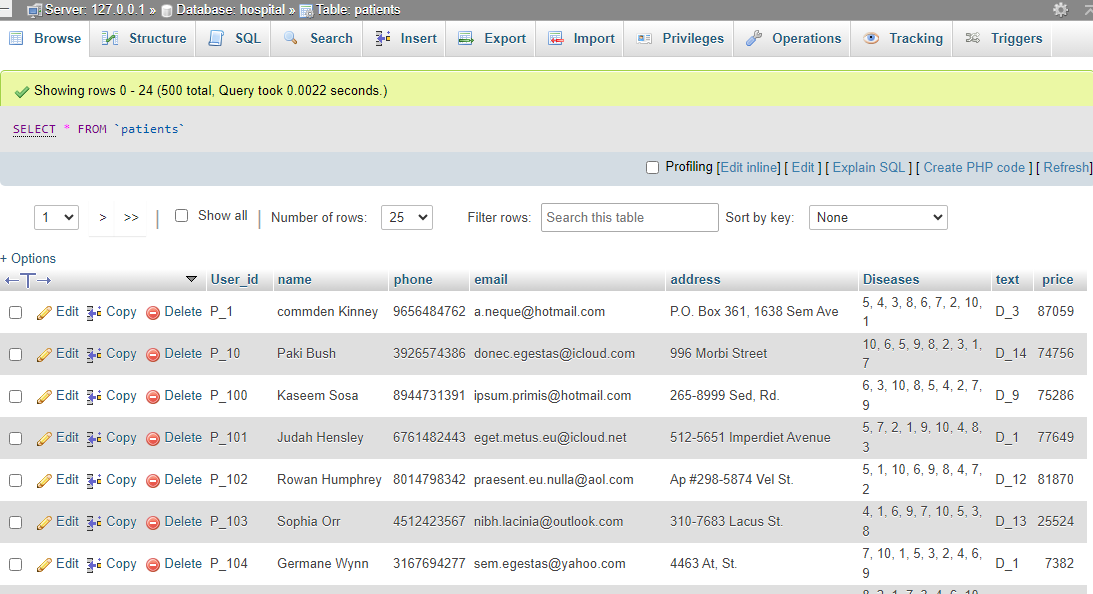
**Figure 10 : Created Table with Name med\_purchased**

****

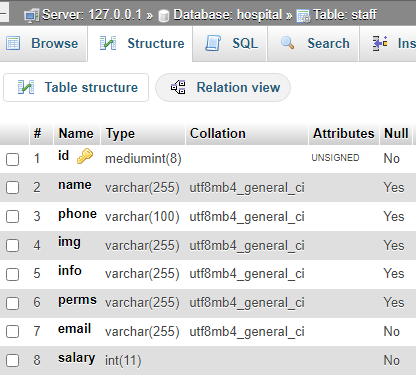
**Figure 11 : Inserted rows in table med\_purchased**

****

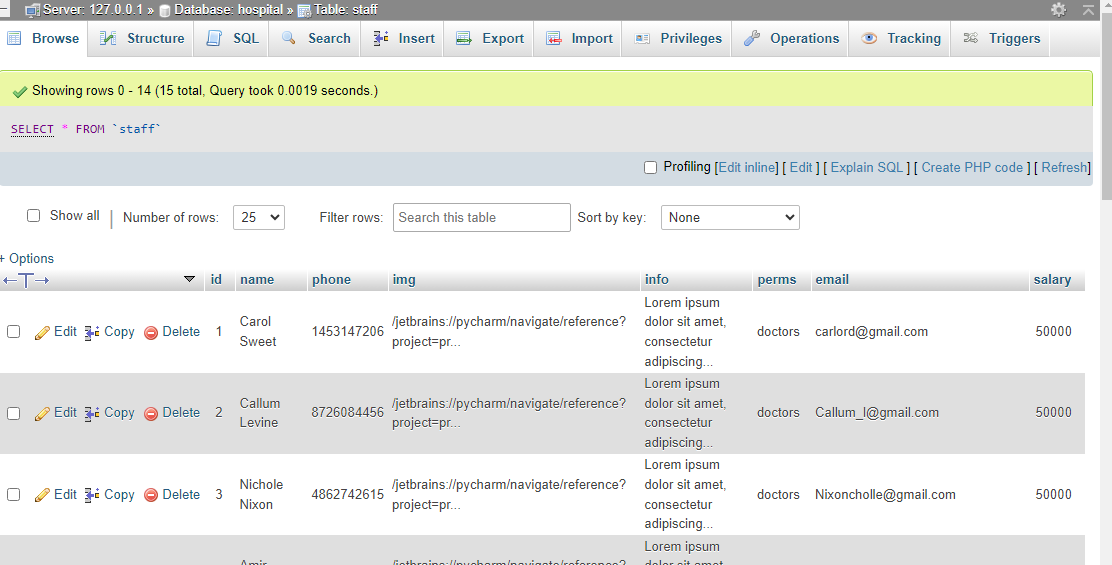
**Figure 12 : Created Table with Name patients**

****

**Figure 13 : Inserted rows in table patients**

****

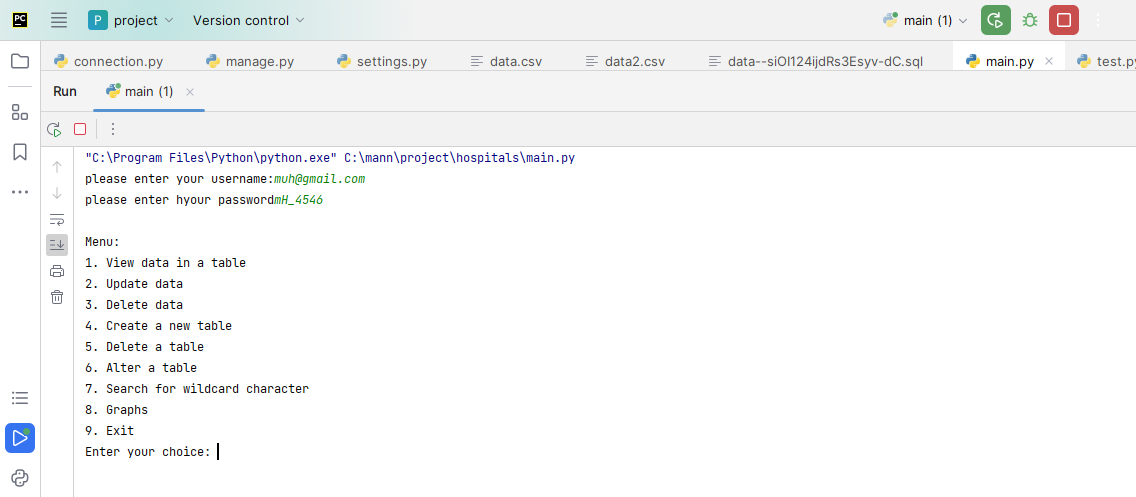
**Figure 14 : Created Table with Name staff**

****

**Figure 15 : Inserted rows in table staff**

**Output of the project:**

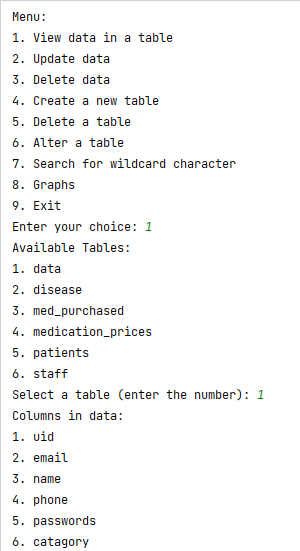
**ADMIN PANEL**

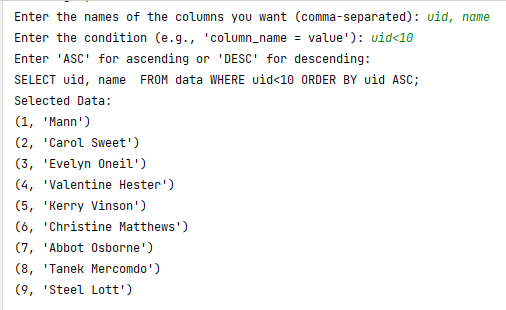


**Figure 16: Execution of the project**

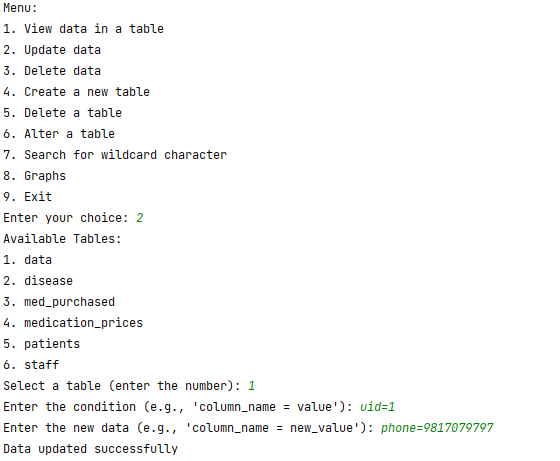


**LOGIN AS ADMIN:**

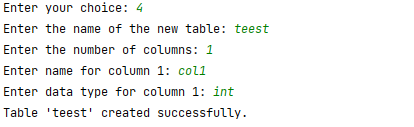




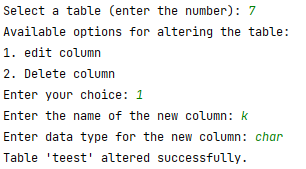
**SELECT DATA FROM TABLE:**



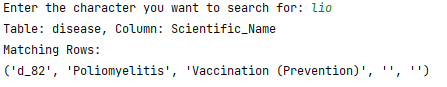
**UDATE TABLE:**



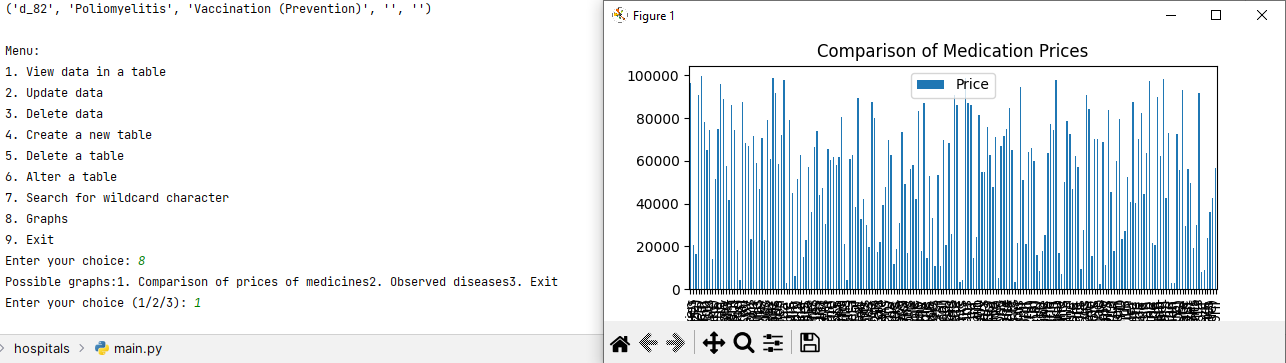
**CREATE TABLE:**



**ALTER:**

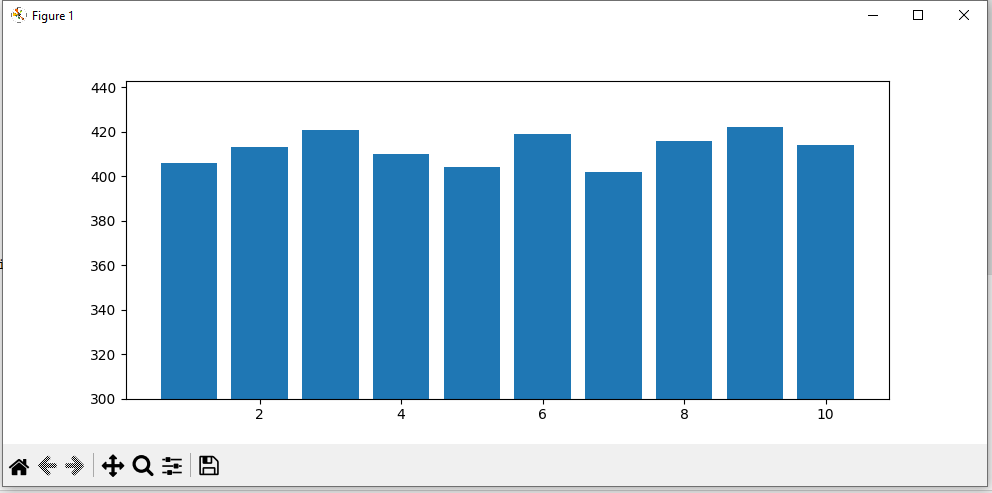


**WILDCARD:**



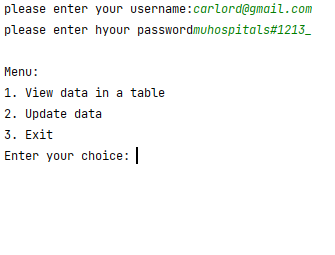
**Figure 5 : Comparison of Medication Prices**

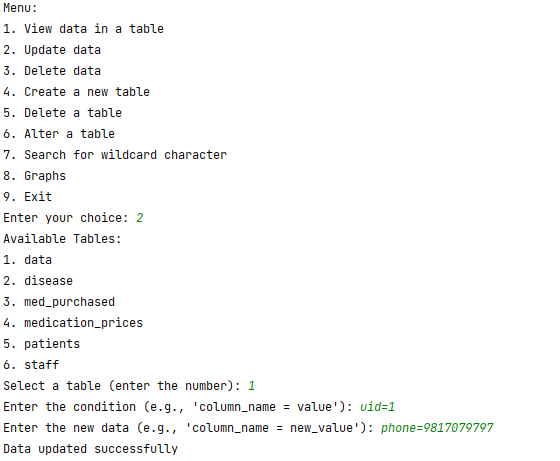
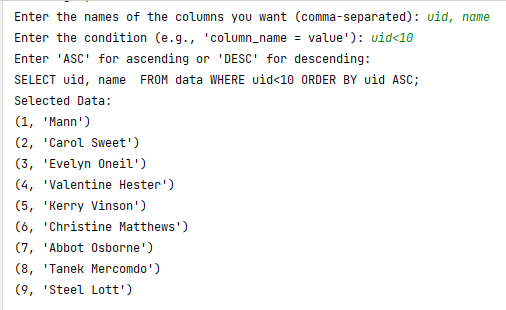




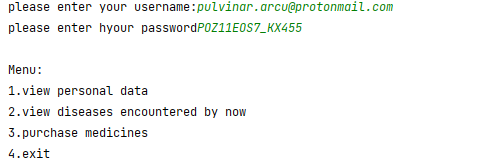
**Figure 5 : Comparison of observed diseases**

**LOGIN AS DOCTOR:**



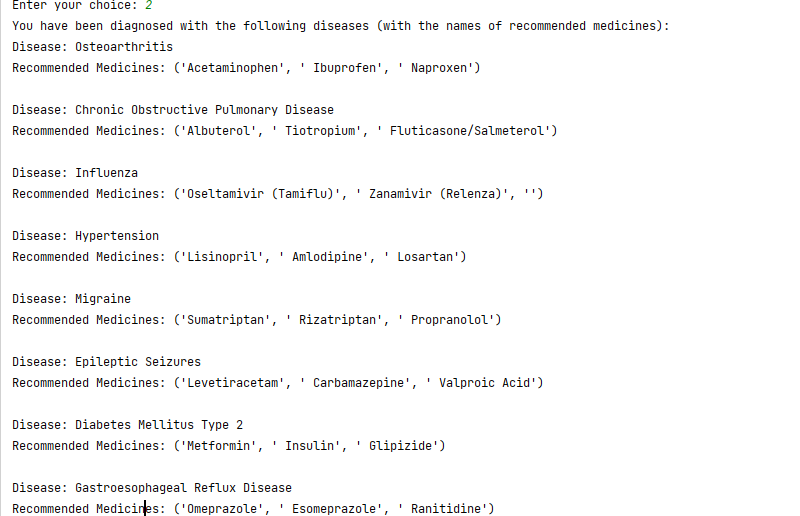


**LOGIN AS PATIENT:**

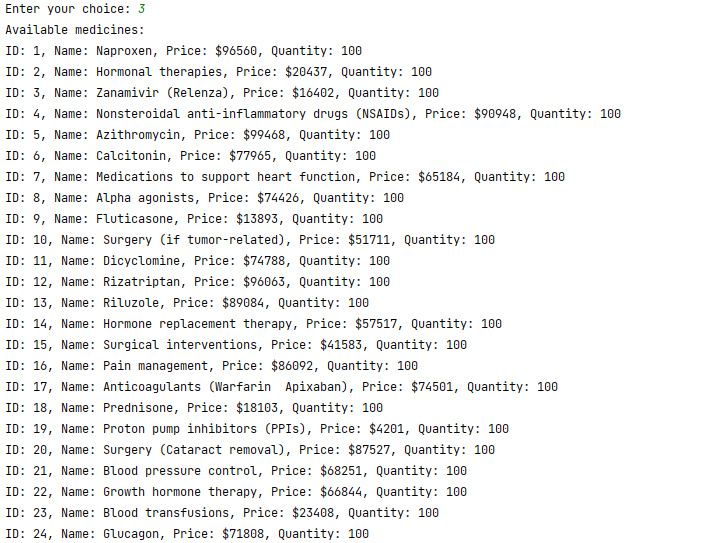
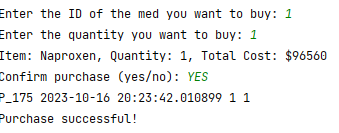


\

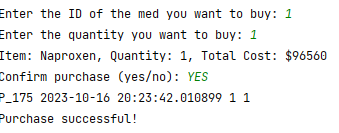
**VIEW OWN DATA:**



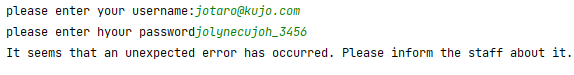
**DISEASES HISTORY:**



**PURCHASE MEDICINES:**



**ENTERING WRONG EMAIL OR PASSWORD:**



### FURTHER SCOPES OF THE PROJECT

### This project can be provided with the GUI functionality. Report facility can also be implemented. Some security implementation like encryption of the password can be incorporated in the project.

### BIBLIOGRAPHY

* <https://www.w3schools.com/>
* <https://ncert.nic.in/>
* <https://techterms.com/>
* <https://www.geeksforgeeks.org/>
* A book on Informatics Practices by Sumita Arora, Dhanpat Rai Publication
* <https://stackoverflow.com/>
* <https://bugs.python.org/file47781/Tutorial_EDIT.pdf>