INFORMATICS PRACTICES PROJECT

2021-22

Bal Bhawan School School

Shamla Hills,Bhopal

Topic :

E-COMMERCE WEBSITE

Database management system using Python-MySQL connector

Project made by:

Mohammad Maasir
(Class XII Science)

Guided by:

Mr. Abhishek Shrivasta &
Mrs. Sudha Nair

Certificate

this is to certify that **MOHAMMAD MAASIR** of class **12th SCIENCE** of **BAL BHAWAN SCHOOL**, Shamla Hills, Bhopal has completed his informatics practices project for the session 2021-22 on the topic **"E-COMMERCE WEBSITE DATABASE MANAGEMENT SYSTEM"** using Python – MySQL RDBMS technologies.

I CERTIFY THAT THE PROJECT IS IN ACCORDANCEWITH THE GUIDELINES ISSUED BY

CENTRAL BOARD OF SECONDARY EDUCATION,INDIA.

External	**Internal**	**Principal**

Message

It is a sign of creative activity and cooperation, that our students, guided and inspired by the teachers are preparing themselves not only for a bright future but also to become a successful person of character for the nation.

Iam delighted to know that Mohammad Maasir of class XII–Science has completed his project on the topic “E-Commerce Website database management system” within the stipulated time period.

Mrs.Humaira Arif
(Principal, Bal Bhawan School, Bhoplal)

Acknowledgement

I owe a great many thanks to a great many people who helped and supported me during the making of this project. My deepest thanks to my teachers, **Mr. Abhishek Shrivastava** & **Mrs. Sudha Nair**, the guides of the project for guiding and correcting various mistakes with attention and care. They have taken pain to go through the project and make necessary correction as and when needed. I express my thanks to the respected principal **Mrs. Humaira Arif** of Bal Bhavan School, for extending her support. Thanks, and appreciation to all the helpful people for their support. I also extend my heartfelt thanks to my family & well-wishers for their effective encouragement, guidance, and patience throughout the project.

Contents

Introduction to Python

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modifyfiles.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this project. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this project Python will be written in the default python IDLE(Integrated Development and Learning Environment). It is possible to write Python in an Integrated Development Environment, such as Virtual Studio Code, Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using white space, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

MySQL

MySQL is freely available open source Relational Database management System (RDBMS) that uses Structured Query Language (SQL). It is downloadable from site www.mysql.org In a MySQL database, information is stored in tables. A single MySQL database can contain many tables at once and store thousands of individual records. MySQL provides you with a rich set of features that that support a secure environment for storing, maintaining, and accessing data. MySQL is a fast, reliable, scalable alternative to many of the commercial RDBMS available.

MySQL Database System

My SQL database system refers to the combination of a MySQL server instance and a MySQL database. MySQL operates using client/server architecture in which the server runs on the machine containing the database and clients connect to the server over a network. MySQL is a multi-user database system, meaning several users can access the database simultaneously. The sever (MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients. Clients are programs that connect to the database server and issue queries in a pre-specified format. MySQL is compatible with the standards-based SQL (Structured Query Language). The client program may contact the server programmatically (meaning a program call the server during execution) or manually.

Classification of SQL Statements

SQL provides many different types of commands used for different purposes. SQL commands can be divided into following categories :-

(please turn over)

(i)Data Definition Language (DDL)Commands:

Commands that allows you to perform tasks related to data definition. Example-

- ✓ Creating, altering and dropping
- ✓ Granting and revoking privileges and roles
- ✓ Maintenance commands

(ii)Data Manipulation Language (DML) Commands:

Commands that allow you to perform tasks related to data manipulation. Example:

retrieval, insertion, deletion and modification of data stored in database

(iii)Transaction Control Language (TCL) Commands:

Commands that allow you to manage and control the transaction (a transaction is one complete unit of work involving many steps) for example:

- ✓ making changes todatabase, permanent
- ✓ undoing changes todatabase, permanent
- ✓ creating save points
- ✓ setting properties for current transactions.

Brief History About MySQL

- MySQL was created and is supported by MySQL LAB, a company based in Sweden. This company is now a subsidiary of Sun Microsystems, which holds the copyright to most of the codebase. On April 20TH, 2009 Oracle database, announced a deal to acquire Sun Microsystems.
- The chief inventor of MySQL was Michael Widenius (a.k.a. Monty). MySQL has been named after Monty's daughter My.
- The logo of MySQL, the dolphin, is named as "Sakila".

Front-End and Back-End

FRONT END

The project uses python as its front end, that is it is the part where user interact. The front end basically includes input facilities through which the user interacts. Apart from the user interaction, the front end also contains database objects that form a layer between the user and BACK END.

BACK END

MySQL is used as a back end. It handles all database accesses through one or more servers. A Server is a special computer that is responsible for processing requests send to it. It processes all the data and query passed to it and sends result back to the front end where it is displayed to the user

Hardware And Software Requirements

Minimum Hardware Specification of the machine to be used :-

1. P-1 Processor
2. 32MB RAM
3. 1.4inch color monitor
4. 2.1GB HDD
5. 233 MHZ CPU Clock Speed

Minimum Software Specification of the machine to be used :-

(tools & platforms used)

Python 3.6 and above(I am using python3.10)

MySQL version 8.0 is used in this project.
This project uses GUI provided by Python and MySQL which is a RDBMS.
(additionally I have also used Microsoft Visual Studio Code as code editor.)

ENTITY RELATIONSHIP MODEL FOR MY E- COMMERCE WEBSITE

Description About My E-Commerce Website :

This is a website with Registered buyers, and sellers use to do their respective tasks. Furthermore, this website also keeps records of Registered Delivery Mans. All the three category of people mentioned here have to make an account on the website, with distinct user-id, and create a password for their account. Furthermore, they have to submit their **mobile number, email address and pin code(city)** while making the account.

Any customer can order products from the website and give ratings & reviews about the products or items. They can also give ratings about the items, sellers and deliverymen.

Apart from ratings and reviews, the users are free to complain about any issue related to their orders.

Dates and time of all important events will be recorded.

A user which is found not active for 2 years should be given warning notification (on Email/SMS) that his/her account would be removed from the database if he/she remains inactive for 6 more months.

The name of this imaginary website is “Eshop”

Entity Relationship Model :

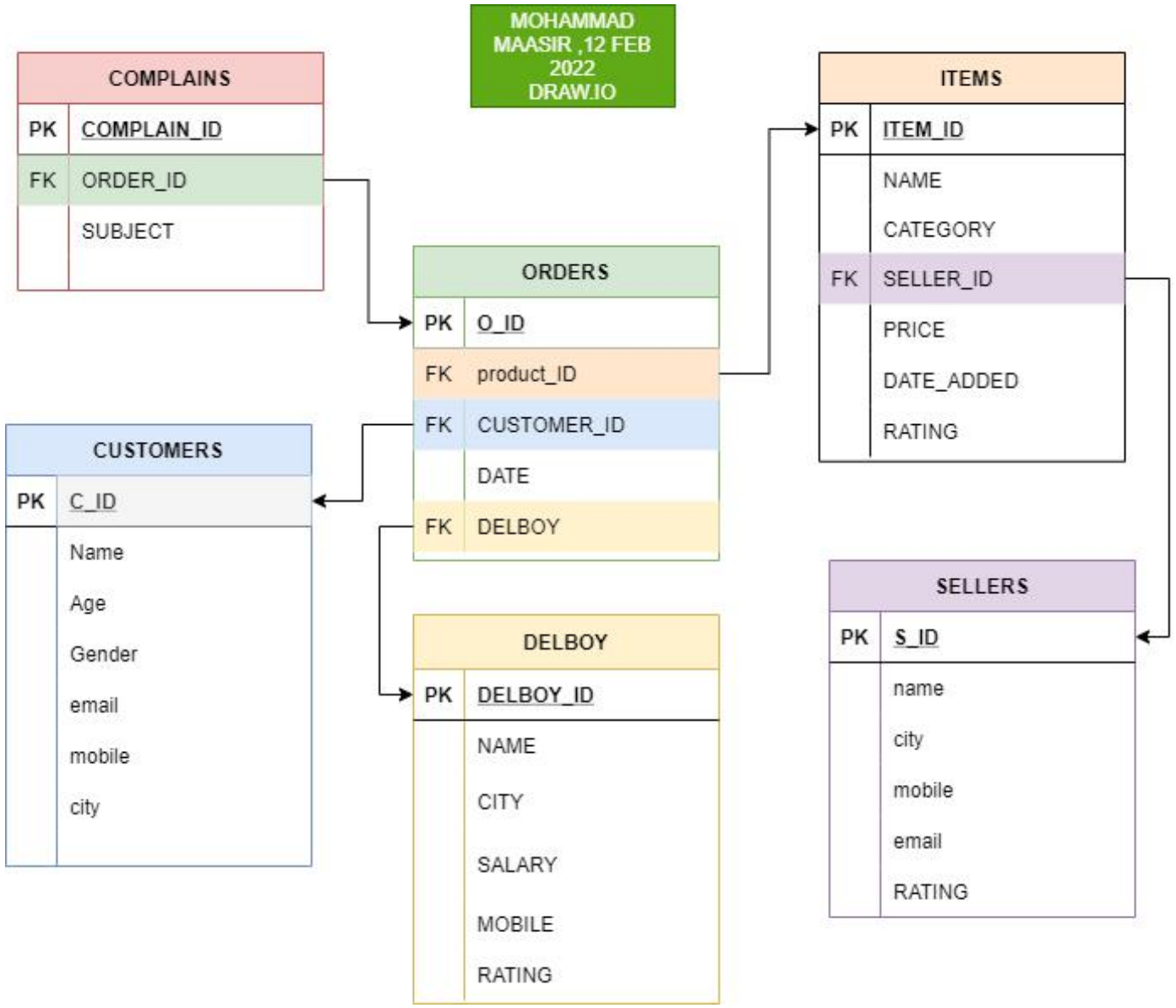
The ER model will define the database which stores information about:-

1. Customer (consumer) details
2. Seller details
3. Item details
4. Record of all orders
5. Delivery boy details
6. Ratings of: items, seller and delivery boys,which are done by the customers, will be stored in respective tables.no need to create separate table
7. Complaints done by the costumers on a particular order regarding delivery boy or item. Will be stored in a single table.

(six tables in total)

Reference for abbreviations	
Abbreviation	***meaning***
C_ID	Customer ID (unique identity)
O_ID	Order ID (unique identity)
S_ID	Seller ID (unique identity)
DELBOY	Delivery Boy

ER DIAGRAM :



Simplified Entity Relation diagram for the relational database. The tables in the database will be created according this diagram.

Diagram created by me, using draw.io

This diagram is also available on my GitHub Page : github.com/maasir554

Direct Link :
https://github.com/maasir554/maasir554/blob/main/Untitled%20Diagram.drawio

FRONT END CODING

PYTHON(and SQL connector), Manually Highlighted

Please refer to last pages of this document for viewing code highlighted by GitHub+VS Code .

```
#importing the connector :-
import mysql.connector as c
#defining the database :-
#Note: the name of database is Eshop. you have to create database Eshop
#to run this program.
database = c.connect(host = "localhost", user = 'root', password =
"password", database = 'eshop')
#defining cursor(it will be used to excute SQL commands) :-
cursor = database.cursor(buffered=True)
```

```
#making the tables in SQL database using python commands
print('You are connected to the server eshop successfully.')
cursor.execute('CREATE TABLE IF NOT EXISTS items(item_id INT(4) PRIMARY
KEY NOT NULL,item_name VARCHAR(50),category VARCHAR(30),seller_id
INT(4) NOT NULL,price INT(8) NOT NULL,date_added DATE,rating
FLOAT(2,1))' )
cursor.execute('CREATE TABLE IF NOT EXISTS customers(customer_id INT(4)
PRIMARY KEY NOT NULL,customer_name VARCHAR(30) NOT NULL,customer_age
INT(3),gender ENUM("male","female","others"),email VARCHAR(30),mobile
VARCHAR(12),city VARCHAR(20))')
cursor.execute('CREATE TABLE IF NOT EXISTS delboy(delboy_id INT(4)
PRIMARY KEY NOT NULL,name VARCHAR(30) NOT NULL,city VARCHAR(20),mobile
VARCHAR(12),salary INT(4),rating float(2,1))')
cursor.execute('CREATE TABLE IF NOT EXISTS sellers(seller_id INT(4)
PRIMARY KEY NOT NULL,name VARCHAR(30) NOT NULL,city VARCHAR(20),mobile
VARCHAR(12),email VARCHAR(30),rating float(2,1))')
cursor.execute('CREATE TABLE IF NOT EXISTS orders(order_id INT(4)
PRIMARY KEY NOT NULL,item_id INT(4) NOT NULL,customer_id INT(4) NOT
NULL,date_placed DATE,delboy_ID INT(4) NOT NULL)')
cursor.execute('CREATE TABLE IF NOT EXISTS complains(complain_id INT(4)
PRIMARY KEY NOT NULL,order_id INT(4) NOT NULL,subject VARCHAR(20) NOT
NULL)')
```

```
#MAKING THE MENU(USER INTERFACE) FOR FRONT END
def menu():
    print("--- menu opened ---")
    a = int(input("to add data please type : 1 \nto delete data please type : 2
\nto exit this menu, please type 3 \nResponse : "))
   if (a==1):
        b= int(input("type number : \n'1' to add customer data \n'2' to add
seller data \n'3' to add item data \n'4' to add order data \n'5' to add delboy
data \n'6' to add complain. \nResponse : "))
        if (b==1):
            add_customer()
        elif (b==2):
            add_seller()
        elif (b==3):
            add_item()
        elif (b==4):
            add_order()
        elif (b==5):
            add_delboy()
        elif (b==6):
            add_complain()
        else:
            print('please enter a valid response')
 elif (a==2):
        b= int(input("type number : \n'1' to delete customer data \n'2' to
delete seller data \n'3' to delete item data \n'4' to delete order data \n'5' to
delete delboy data \n'6' to delete complain \nResponse : "))
        if (b==1):
            delete_customer()
        elif (b==2):
            delete_seller()
        elif (b==3):
            delete_item()
        elif (b==4):
            delete_order()
        elif (b==5):
            delete_delboy()
        elif (b==6):
            delete_complain()
        else:
            print('please enter a valid response')
            menu()
```

```
#DEFINING FUNCTIONS FOR FEEDING DATA IN THE TABLES OF THE SERVER
def add_customer():
    print("add_customer() function started.")
    c_id = input('enter customer ID  : ')
    name = input('enter customer name : ')
    age = input("enter customer age : ")
    gender = input("enter gender of customer : ")
    email = input('enter customer email ID : ')
    mobile = input('enter customer mobile number : ')
    city = input('enter customer city : ')
    data = (c_id,name,age,gender,email,mobile,city)
    statement = "INSERT INTO customers VALUES(%s,%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table customer successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_customer()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. toa call back
use menu())")
def add_seller():
    print("add_seller() function started.")
    s_id = input('enter seller ID  : ')
    name = input('enter seller name : ')
    city = input('enter seller city : ')
    mobile = input('enter seller mobile number : ')
    email = input('enter seller email ID : ')
    rating = input('enter the rating of seller : ')
    data = (s_id,name,city,mobile,email,rating)
    statement = "INSERT INTO sellers VALUES(%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table sellers successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_seller()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back
         to call back use menu())")
```

```
def add_item():
    print("add_item() function started.")
    item_id = input('enter item ID  : ')
    name = input('enter item name : ')
    cat = input('enter category of item : ')
    sid = input('enter seller_id of item seller : ')
    price = input('enter price of item : ')
    date_added = input('enter date added : ')
    rating = input('enter rating of item : ')
    data = (item_id,name,cat,sid,price,date_added,rating)
    statement = "INSERT INTO items VALUES(%s,%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table items successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_item()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call
back use menu())")

def add_order():
    print("add_order() function started.")
    oid = input('enter order id : ')
    iid = input('enter item id : ')
    cid = input('enter customer id : ')
    dp = input('enter date placed : ')
    dbid = input('enter delivery boy id : ')
    data = (oid,iid,cid,dp,dbid)
    statement = 'insert into orders values(%s,%s,%s,%s,%s)'
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to the table ORDERS successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_order()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call
          back use menu())")
```

```
def add_delboy():
    print("add_delboy() function started.")
    did = input('enter delboy ID  : ')
    name = input('enter delboy name : ')
    city = input('enter delboy city : ')
    mobile = input('enter delboy mobile number : ')
    salary = input('enter delboy salary : ')
    rating = input('enter the rating of delboy : ')
    data = (did,name,city,mobile,salary,rating)
    statement = "INSERT INTO delboy VALUES(%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table delboy successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_delboy()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call
back use menu())")

def add_complain():
    print("add_complain() function started.")
    cid = input("enter complain ID : ")
    oid = input("enter order id : ")
    sub = input("enter the subject of complain : ")
    data = (cid,oid,sub)
    statement = "INSERT INTO COMPLAINS VALUES(%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print("data inserted to the table COMPLAINS successfully!")
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_complain()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. toa call
back use menu())")
```

```
#DEFINING FUNCTIONS TO DELETE DATA FROM TABLES OF DATABASE :-
def delete_customer():
    print("delete_customer() function started.")
    cid = input('enter customer_id : ')
    cursor.execute('DELETE FROM customers WHERE customer_id=%s',(cid,))
    database.commit()
    print('deleted the customer data of customer_id : ',cid,'from table customers
successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_customer()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back use
menu())")
def delete_seller():
    print("delete_seller() function started.")
    sid = input('enter seller_id : ')
    cursor.execute('DELETE FROM sellers WHERE seller_id=%s',(sid,))
    database.commit()
    print('deleted the seller data of seller_id : ',sid,' from table sellers
successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_seller()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use
menu())")
def delete_item():
    print("delete_item() function started.")
    iid = input('enter item_id : ')
    cursor.execute('DELETE FROM items WHERE item_id=%s',(iid,))
    database.commit()
    print('deleted the item data of item_id : ',iid,' from table items
successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_item()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use
menu())")
```

```
def delete_order():
    print("delete_order() function started.")
    x = input('enter order_id : ')
    cursor.execute('DELETE FROM orders WHERE order_id=%s',(x,))
    database.commit()
    print('deleted the item data of order_id : ',x,' from table orders
successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_order()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call
back, use menu())")
    def delete_delboy():
    print("delete_delboy() function started.")
    x = input('enter delboy_id : ')
    cursor.execute('DELETE FROM delboy WHERE delboy_id=%s',(x,))
    database.commit()
    print('deleted the data of delboy_id ',x,' from table delboy successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_delboy()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call
back, use menu())")
    def delete_complain():
    print("delete_complain() function started.")
    x = input('enter complain_id : ')
    cursor.execute('DELETE FROM complains WHERE complain_id=%s',(x,))
    database.commit()
    print('deleted the data of complain_id ',x,' from table complains
successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_complain()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call
back, use menu())")
```

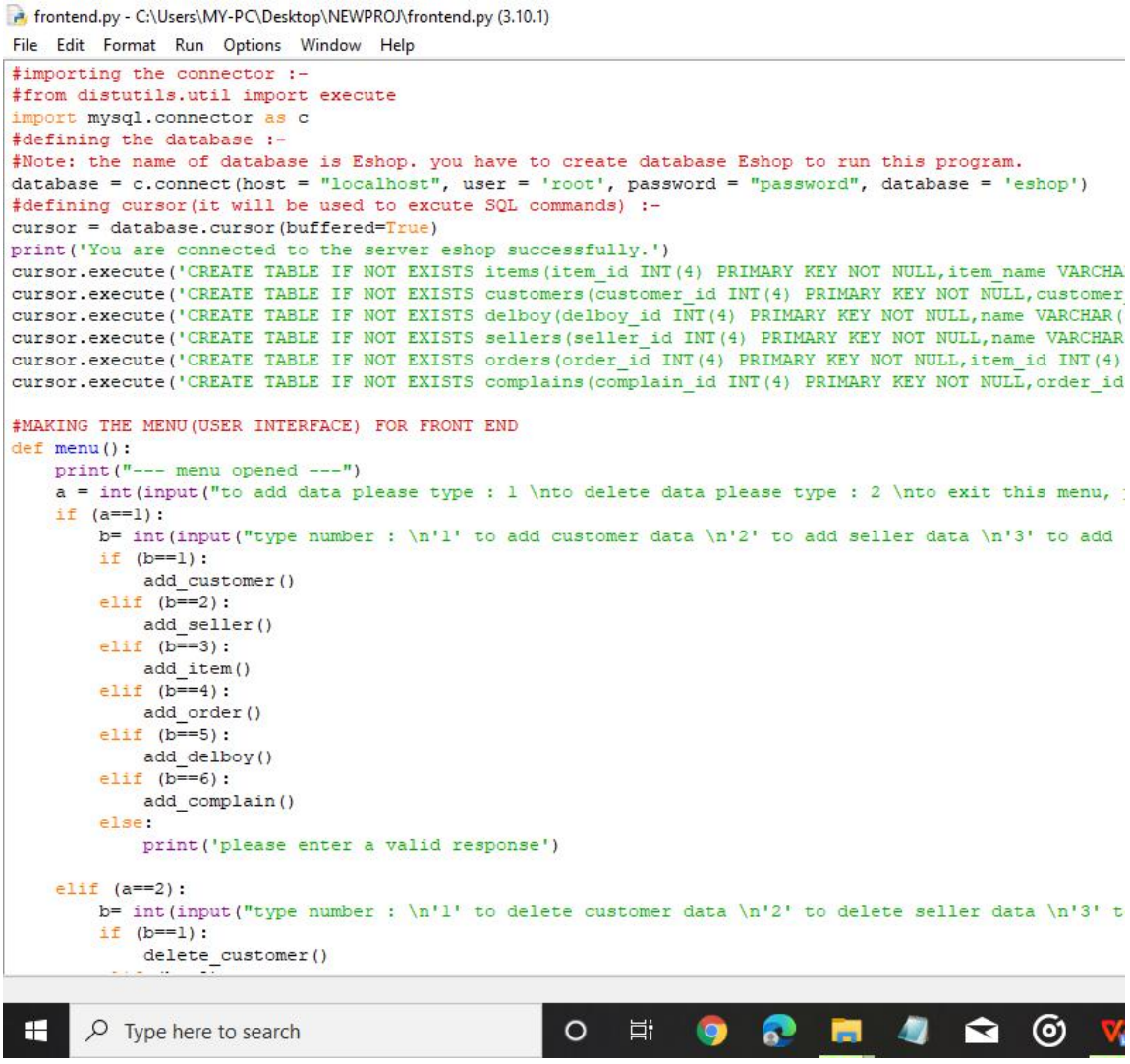
```
#FINALLY, SERVING THE PROGRAM TO USER : -

print(“ WELCOME TO THE DATABASE MANAGEMENT USER INTERFACE OF ESHOP")
print("(created by Mohammad Maasir @ date 13th Feb,2022, as a school
project.)")
menu()
```

NOTE

The above boxes of code are NOT the screenshot,These are copied from my file frontend.py and pasted here,To dispay the text clearly. Original file’s screenshots are pasted below for reference.

ORIGINAL FILE SCREENSHOTS



frontend.py - C:\Users\MY-PC\Desktop\NEWPROJ\frontend.py (3.10.1)

File Edit Format Run Options Window Help

```
            print('please enter a valid response')

    elif (a==2):
        b= int(input("type number : \n'1' to delete customer data \n'2' to delete seller data \n'3' to delete item data \n
        if (b==1):
            delete_customer()
        elif (b==2):
            delete_seller()
        elif (b==3):
            delete_item()
        elif (b==4):
            delete_order()
        elif (b==5):
            delete_delboy()
        elif (b==6):
            delete_complain()
        else:
            print('please enter a valid response')
            menu()

    elif(a==3):
        print('Thank You! for using this program, Have a nice day! \n you can now exit or call functions manually. \nlist
    else:
        print('please enter a valid response')
        menu()

#DEFINING FUNCTIONS FOR FEEDING DATA IN THE TABLES OF THE SERVER

def add_customer():
    print("add_customer() function started.")
    c_id = input('enter customer ID  : ')
    name = input('enter customer name : ')
    age = input("enter customer age : ")
    gender = input("enter gender of customer : ")
    email = input('enter customer email ID : ')
    mobile = input('enter customer mobile number : ')
    city = input('enter customer city : ')
    data = (c_id,name,age,gender,email,mobile,city)
    statement = "INSERT INTO customers VALUES(%s,%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
```

frontend.py - C:\Users\MY-PC\Desktop\NEWPROJ\frontend.py (3.10.1)

File Edit Format Run Options Window Help

```
    database.commit()
    print('data inserted to table customer successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_customer()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. toa call back use menu())")

def add_seller():
    print("add_seller() function started.")
    s_id = input('enter seller ID  : ')
    name = input('enter seller name : ')
    city = input('enter seller city : ')
    mobile = input('enter seller mobile number : ')
    email = input('enter seller email ID : ')
    rating = input('enter the rating of seller : ')
    data = (s_id,name,city,mobile,email,rating)
    statement = "INSERT INTO sellers VALUES(%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table sellers successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_seller()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back use menu())")

def add_item():
    print("add_item() function started.")
    item_id = input('enter item ID  : ')
    name = input('enter item name : ')
    cat = input('enter category of item : ')
    sid = input('enter seller_id of item seller : ')
    price = input('enter price of item : ')
    date_added = input('enter date added : ')
    rating = input('enter rating of item : ')
    data = (item_id,name,cat,sid,price,date_added,rating)
```

frontend.py - C:\Users\MY-PC\Desktop\NEWPROJ\frontend.py (3.10.1)

File Edit Format Run Options Window Help

```
    data = (did,name,city,mobile,salary,rating)
    statement = "INSERT INTO delboy VALUES(%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table delboy successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_delboy()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back use menu())")

def add_complain():
    print("add_complain() function started.")
    cid = input("enter complain ID : ")
    oid = input("enter order id : ")
    sub = input("enter the subject of complain : ")
    data = (cid,oid,sub)
    statement = "INSERT INTO COMPLAINS VALUES(%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print("data inserted to the table COMPLAINS successfully!")
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_complain()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. toa call back use menu())")

#DEFINING FUNCTIONS TO DELETE DATA FROM TABLES OF DATABASE :-
def delete_customer():
    print("delete_customer() function started.")
    cid = input('enter customer_id : ')
    cursor.execute('DELETE FROM customers WHERE customer_id=%s',(cid,))
    database.commit()
    print('deleted the customer data of customer_id : ',cid,'from table customers successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_customer()
```

frontend.py - C:\Users\MY-PC\Desktop\NEWPROJ\frontend.py (3.10.1)

File Edit Format Run Options Window Help

```
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back use menu())")

def delete_seller():
    print("delete_seller() function started.")
    sid = input('enter seller_id : ')
    cursor.execute('DELETE FROM sellers WHERE seller_id=%s',(sid,))
    database.commit()
    print('deleted the seller data of seller_id : ',sid,' from table sellers successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_seller()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use menu())")

def delete_item():
    print("delete_item() function started.")
    iid = input('enter item_id : ')
    cursor.execute('DELETE FROM items WHERE item_id=%s',(iid,))
    database.commit()
    print('deleted the item data of item_id : ',iid,' from table items successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_item()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use menu())")

def delete_order():
    print("delete_order() function started.")
    x = input('enter order_id : ')
    cursor.execute('DELETE FROM orders WHERE order_id=%s',(x,))
    database.commit()
    print('deleted the item data of order_id : ',x,' from table orders successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
```

frontend.py - C:\Users\MY-PC\Desktop\NEWPROJ\frontend.py (3.10.1)

File Edit Format Run Options Window Help

```
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_order()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use menu())")

def delete_delboy():
    print("delete_delboy() function started.")
    x = input('enter delboy_id : ')
    cursor.execute('DELETE FROM delboy WHERE delboy_id=%s',(x,))
    database.commit()
    print('deleted the data of delboy_id ',x,' from table delboy successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_delboy()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use menu())")

def delete_complain():
    print("delete_complain() function started.")
    x = input('enter complain_id : ')
    cursor.execute('DELETE FROM complains WHERE complain_id=%s',(x,))
    database.commit()
    print('deleted the data of complain_id ',x,' from table complains successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_complain()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use menu())")

#FINALLY, SERVING THE PROGRAM TO USER : -

print("* * * * * * * WELCOME TO THE DATABASE MANAGEMENT USER INTERFACE OF ESHOP * * * * * * *")
print("(created by Mohammad Maasir @ date 13th Feb,2022, as a school project.)")
menu()
```

frontend.py - C:\Users\MY-PC\Desktop\NEWPROJ\frontend.py (3.10.1)

File Edit Format Run Options Window Help

```
    if (b=='y'):
        delete_delboy()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use menu())")

def delete_complain():
    print("delete_complain() function started.")
    x = input('enter complain_id : ')
    cursor.execute('DELETE FROM complains WHERE complain_id=%s',(x,))
    database.commit()
    print('deleted the data of complain_id ',x,' from table complains successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_complain()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use menu())")

#FINALLY, SERVING THE PROGRAM TO USER : -

print("* * * * * * * WELCOME TO THE DATABASE MANAGEMENT USER INTERFACE OF ESHOP * * * * * * *")
print("(created by Mohammad Maasir @ date 13th Feb,2022, as a school project.)")
menu()
"""
although this program have a menu, we can also call the functions defined here to
add or delete data when this program is running in the IDLE shell, manually (when you are outside menu i.e. exit the menu.)

|

********************************************* END OF THE CODE ******************************************************
"""
```

ENTERING DATA TO THE SERVER

(Using the above code to feed/insert data in the tables of the server)

For customers :-

IDLE Shell 3.10.1

File Edit Shell Debug Options Window Help

```
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec  6 2021, 19:10:37) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\MY-PC\Desktop\NEWPROJ\frontend.py =============
You are connected to the server eshop successfully.
* * * * * * * WELCOME TO THE DATABASE MANAGEMENT USER INTERFACE OF ESHOP * * * * * * *
(created by Mohammad Maasir @ date 13th Feb,2022, as a school project.)
--- menu opened ---
to add data please type : 1
to delete data please type : 2
to exit this menu, please type 3
Response : 1
type number :
'1' to add customer data
'2' to add seller data
'3' to add item data
'4' to add order data
'5' to add delboy data
'6' to add complain.
Response : 1
add_customer() function started.
enter customer ID  : 1
enter customer name : Aamir Khan
enter customer age : 42
enter gender of customer : male
enter customer email ID : amirk136@gmail.com
enter customer mobile number : 6985894532
enter customer city : mumbai
data inserted to table customer successfully!
do you want ot add more data to THIS table? (y/n) : y
add_customer() function started.
enter customer ID  : 2
enter customer name : Rajpal Yadav
enter customer age : 35
enter gender of customer : male
enter customer email ID : rajpal235@yahoo.com
enter customer mobile number : 9856421978
enter customer city : mumbai
data inserted to table customer successfully!
do you want ot add more data to THIS table? (y/n) : y
add_customer() function started.
```

```
enter customer name : Pankaj Tripathi
enter customer age : 38
enter gender of customer : male
enter customer email ID : tripath.pankaj@outlook.com
enter customer mobile number : 6985458798
enter customer city : mumbai
data inserted to table customer successfully!
do you want ot add more data to THIS table? (y/n) : y
add_customer() function started.
```

```
enter customer ID  : 5
enter customer name : Lara Croft
enter customer age : 34
enter gender of customer : female
enter customer email ID : croft01lara@outlook.co
enter customer mobile number : 3369845621
enter customer city : york
data inserted to table customer successfully!
do you want ot add more data to THIS table? (y/n
add_customer() function started.
enter customer ID  : 6
enter customer name : Heng Fuji
enter customer age : 14
enter gender of customer : male
enter customer email ID : heng3346@yahoo.com
enter customer mobile number : 0120360152
enter customer city : tokyo
data inserted to table customer successfully!
do you want ot add more data to THIS table? (y/n
--- menu opened ---
to add data please type : 1
to delete data please type : 2
to exit this menu, please type 3
Response :
```

For sellers :-

```
enter seller ID  : 4
enter seller name : Xiomi
enter seller city : Bejing
enter seller mobile number : 6589785421
enter seller email ID : support.mi.ac.in
enter the rating of seller : 4.8
data inserted to table sellers successfully!
do you want ot add more data to THIS table? (y/n) :
```

For items

```
--- menu opened ---
to add data please type : 1
to delete data please type : 2
to exit this menu, please type 3
Response : 1
type number :
'1' to add customer data
'2' to add seller data
'3' to add item data
'4' to add order data
'5' to add delboy data
'6' to add complain.
Response : 3
add_item() function started.
enter item ID  : 2
enter item name : Redmi Note 9 Pro
enter category of item : Mobile
enter seller_id of item seller : 4
enter price of item : 13999
enter date added : 2022-01-13
enter rating of item : 4.9
data inserted to table items successfully!
do you want ot add more data to THIS table? (y/n) : y
add_item() function started.
enter item ID  : 3
enter item name : 12th Boards PYQ Book
enter category of item : Books
enter seller_id of item seller : 2
enter price of item : 599
enter date added : 2022-02-05
enter rating of item : 4.8
data inserted to table items successfully!
do you want ot add more data to THIS table? (y/n) : n
--- menu opened ---
```

For Orders placed

```
--- menu opened ---
to add data please type : 1
to delete data please type : 2
to exit this menu, please type 3
Response : 1
type number :
'1' to add customer data
'2' to add seller data
'3' to add item data
'4' to add order data
'5' to add delboy data
'6' to add complain.
Response : 4
add_order() function started.
enter order id : 1
enter item id : 1
enter customer id : 2
enter date placed : 2022-01-23
enter delivery boy id : 3
data inserted to the table ORDERS successfully!
```

For delivery boys

```
add_delboy() function started.
enter delboy ID  : 3
enter delboy name : Kalvin
enter delboy city : canada
enter delboy mobile number : 2398564201
enter delboy salary : 8900
enter the rating of delboy : 4.9
data inserted to table delboy successfully!
do you want ot add more data to THIS table? (y/n) : y
add delboy() function started.
```

For complains

```
add_complain() function started.
enter complain ID : 1
enter order id : 3
enter the subject of complain : late delivery
data inserted to the table COMPLAINS successfully!
do you want ot add more data to THIS table? (y/n) : y
add_complain() function started.
enter complain ID : 2
enter order id : 2
enter the subject of complain : damaged item
data inserted to the table COMPLAINS successfully!
do you want ot add more data to THIS table? (y/n) : y
add_complain() function started.
enter complain ID : 3
```

OUTPUT FOR THE COMMANDS

```
mysql> select * from customers;
+-------------+----------------+--------------+--------+-----------------------------+------------+------------+
| customer_id | customer_name  | customer_age | gender | email                       | mobile     | city       |
+-------------+----------------+--------------+--------+-----------------------------+------------+------------+
|           1 | Aamir Khan     |           42 | male   | amirk136@gmail.com          | 6985894532 | mumbai     |
|           2 | Rajpal Yadav   |           35 | male   | rajpal235@yahoo.com         | 9856421978 | mumbai     |
|           3 | Pankaj Tripathi|           38 | male   | tripath.pankaj@outlook.com  | 6985458798 | mumbai     |
|           4 | Julie Martin   |           24 | female | juliemartin445@gmail.com    | 21023654   | california |
|           5 | Lara Croft     |           34 | female | croft01lara@outlook.com     | 3369845621 | york       |
|           6 | Heng Fuji      |           14 | male   | heng3346@yahoo.com          | 0120360152 | tokyo      |
|           7 | Lakshmi Anand  |           25 | female | anand.lakshmi2680@gmail.com | 7000598315 | delhi      |
|           8 | Sapna Yadav    |           34 | female | sapna346@outlook.com        | 7000598465 | delhi      |
|           9 | Balvinder Singh|           24 | male   | singh12balvinder@gmail.com  | 5894563245 | canada     |
+-------------+----------------+--------------+--------+-----------------------------+------------+------------+
9 rows in set (0.00 sec)
```

```
mysql> select * from sellers;
+-----------+--------------+------------+------------+-------------------------+--------+
| seller_id | name         | city       | mobile     | email                   | rating |
+-----------+--------------+------------+------------+-------------------------+--------+
|         1 | Canon Inc.   | California | 5698521230 | support@canon.ac.in     |    4.5 |
|         2 | Arihant Pub. | meerut     | 9875642139 | support@arihant.co.in   |    4.0 |
|         3 | Sony Inc.    | Bejing     | 974123659  | support@sony.com        |    4.2 |
|         4 | Xiomi        | Bejing     | 6589785421 | support.mi.ac.in        |    4.8 |
+-----------+--------------+------------+------------+-------------------------+--------+
4 rows in set (0.00 sec)
```

```
mysql> select * from items;
+---------+----------------------------+-------------+-----------+-------+------------+--------+
| item_id | item_name                  | category    | seller_id | price | date_added | rating |
+---------+----------------------------+-------------+-----------+-------+------------+--------+
|       1 | Canon EOS 1300D DSLR camera| Electronics |         1 | 34999 | 2021-11-23 |    4.6 |
|       2 | Redmi Note 9 Pro           | Mobile      |         4 | 13999 | 2022-01-13 |    4.9 |
|       3 | 12th Boards PYQ Book       | Books       |         2 |   599 | 2022-02-05 |    4.8 |
+---------+----------------------------+-------------+-----------+-------+------------+--------+
3 rows in set (0.00 sec)

mysql>
```

```
mysql> select * from orders;
+----------+---------+-------------+-------------+-----------+
| order_id | item_id | customer_id | date_placed | delboy_ID |
+----------+---------+-------------+-------------+-----------+
|        1 |       1 |           2 | 2022-01-23  |         3 |
|        2 |       3 |           3 | 2022-02-09  |         3 |
|        3 |       3 |           8 | 2021-12-09  |         2 |
+----------+---------+-------------+-------------+-----------+
3 rows in set (0.00 sec)
```

```
mysql> select * from delboy;
+-----------+--------+-----------+------------+--------+--------+
| delboy_id | name   | city      | mobile     | salary | rating |
+-----------+--------+-----------+------------+--------+--------+
|         1 | jole   | california | 3698564589 |   4599 |    4.5 |
|         2 | Heron  | mumbai     | 6547859865 |   1200 |    3.5 |
|         3 | Kalvin | canada     | 2398564201 |   8900 |    4.9 |
|         4 | Henry  | delhi      | 5698564236 |  15999 |    5.0 |
|         5 | louis  | california | 16000502   |  15000 |    5.0 |
+-----------+--------+-----------+------------+--------+--------+
```

```
mysql> select * from complains;
+-------------+----------+-----------------+
| complain_id | order_id | subject         |
+-------------+----------+-----------------+
|           1 |        3 | late delivery   |
|           2 |        2 | damaged item    |
|           3 |        1 | wrong product.  |
+-------------+----------+-----------------+
3 rows in set (0.00 sec)
```

DEMONSTRATION FOR DELETING

```
...
--- menu opened ---
to add data please type : 1
to delete data please type : 2
to exit this menu, please type 3
Response : 2
type number :
'1' to delete customer data
'2' to delete seller data
'3' to delete item data
'4' to delete order data
'5' to delete delboy data
'6' to delete complain
Response : 5
delete_delboy() function started.
enter delboy_id : 5
deleted the data of delboy_id  5  from table delboy successfully!
do you want ot delete more data to THIS table? (y/n) : n
--- menu opened ---
to add data please type : 1
to delete data please type : 2
to exit this menu, please type 3
Response : 3
Thank You! for using this program, Have a nice day!
 you can now exit or call functions manually.
list of the functions(prefix add_ OR delete_ before them) :
customer
seller
item
order
delboy
complain
--- menu closed ---
>>>
```

```
mysql> select * from delboy;
+-----------+--------+------------+------------+--------+--------+
| delboy_id | name   | city       | mobile     | salary | rating |
+-----------+--------+------------+------------+--------+--------+
|         1 | jole   | california | 3698564589 |   4599 |    4.5 |
|         2 | Heron  | mumbai     | 6547859865 |   1200 |    3.5 |
|         3 | Kalvin | canada     | 2398564201 |   8900 |    4.9 |
|         4 | Henry  | delhi      | 5698564236 |  15999 |    5.0 |
+-----------+--------+------------+------------+--------+--------+
4 rows in set (0.00 sec)
```

GRAPHS RELATED TO MY PROJECT

NOTE :

The data I am using in the graphs in the following sections is related to the website management system. But this data is NOT related to any real life survey.

Code for Graphs(using Python's Matplotlib)

```
import matplotlib.pyplot as plt
import numpy as np
#graph 1
x1 = [2000,2005,2010,2015,2020]
y1= [15,18,19,22,35]
plt.plot(x1,y1,marker='o',color = 'violet')
plt.xlabel("Year")
plt.ylabel("number of users (in Thousands)")
plt.show()
#graph 2
arr1 = [48,60,2]
labels1 = ['males','females','others']
colors = ['yellow','hotpink','grey']
plt.pie(arr1,labels = labels1,colors=colors)
plt.legend(title = 'Genderwise Proportion of number of customers as of
the year 2021 :')
plt.show()
#graph 3
x2 = ['Grocery','Wardrobe','Electronics','Books','Furniture']
y2 = [24,57,107,68,20]
plt.bar(x2,y2, color="lightgreen")
plt.xlabel("Category of items")
plt.ylabel("Profit made by the website in the year 2020")
plt.show()
#graph 4
x3 = ['Peter England','Canon Inc.','Sony Inc.','Xiomi','Apple
Inc.','Samsung']
y3 = [2.7,4.6,3.5,4.5,4.8,2.2]
plt.barh(x3,y3,color="pink")
plt.ylabel("Popular Seller(Brand) Names")
plt.xlabel("Mean Customer Rating out of 5.0")
plt.show()
#graph 5
x4 = ['Grocery','Wardrobe','Electronics','Books','Furniture']
y4 = [5056,3269,25699,1509,31220]
plt.bar(x4,y4,color = 'orange')
plt.xlabel('Category of Item : ')
plt.ylabel('Average Product Retail Price in Rupees : ')
plt.show()
```

```
#graph 6
arr2 =
[18,20,25,23,54,16,12,19,13,23,25,24,26,26,24,23,21,33,32,33,35,41,48,4
8,41,49,45,48,47,27,55,56,45,44,41,43,42,41,42,42,41,22,21,21,21,21,23,
23,22,22,21,21,]
bins = [0,5,10,15,20,25,30,35,40,45,50,55,60,65,70,70]
plt.hist(arr2,bins=bins,color = 'skyblue')
plt.xlabel('age groups')
plt.ylabel('number of customers(in thousands)')
plt.show()
```

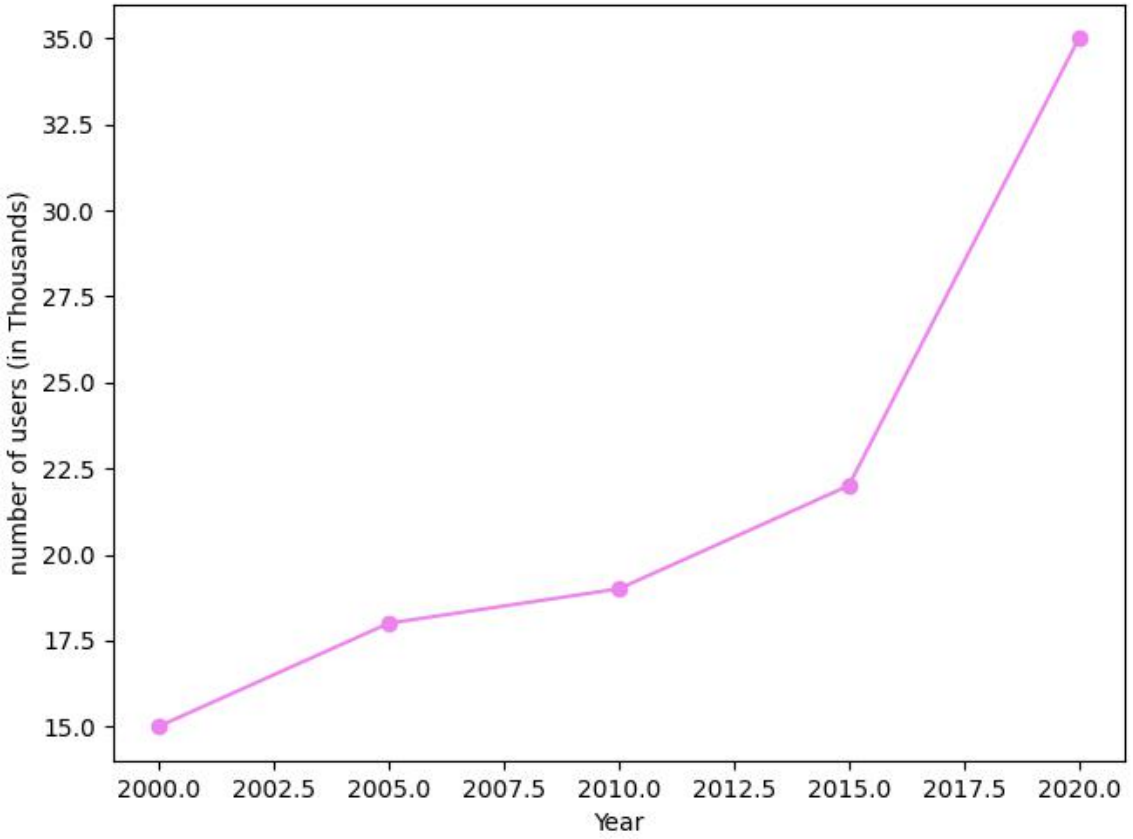
```
#graph 7 : customer proportion(location wise)
x5 = ['Delhi','Mumbai','California','Canada','Washington
D.C.','U.A.E','Bejing','Tokyo','others']
y5 = [10,12,15,8,12,9,17,14,18]
explode = [0.1,0.05,0.20,0.15,0.09,0.1,0.2,0.1,0.05]
plt.pie(y5,labels=x5,explode=explode,shadow = True,colors =
['pink','lightblue','lightgreen','hotpink','crimson','purple','violet',
'skyblue','grey'],autopct='%1.1f%%')
plt.legend(title = 'Proportion of customers location
wise',bbox_to_anchor=(1,1))
plt.show()
```

```
#graph 8 profit of company v/s year
x6 =
[2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,
2014,2015,2016,2017,2018,2019,2020,2021,2022,]
y6=[0.1,0.8,1,1.5,-2.4,3.1,07.5,4.7,14,5.0,1.0,12,-
0.5,1.0,2.7,5.0,15.0,30.5,28,20,31,35,12]
plt.plot(x6,y6,marker = 'o',color = 'green')
plt.xlabel("Financial Year :")
plt.ylabel("Profit made by our website : ")
plt.show()
```

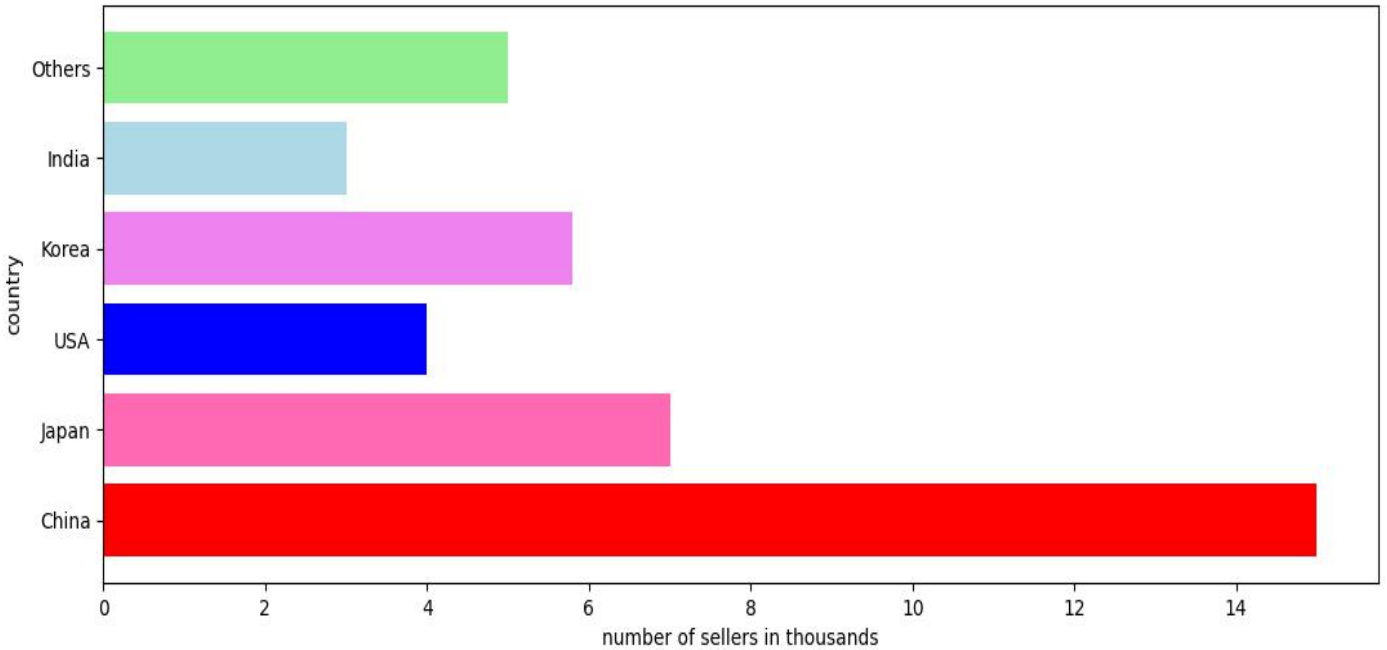
```
#graph 9 number of sellers and location
country = ['China','Japan','USA','Korea','India','Others']
seller = [15,7,4,5.8,3,5]
colors = ["red",'hotpink','Blue','violet','lightblue','lightgreen']
plt.barh(country,seller,color=colors)
plt.ylabel("country")
plt.xlabel("number of sellers in thousands")
plt.show()
```

Output for Graph Code:

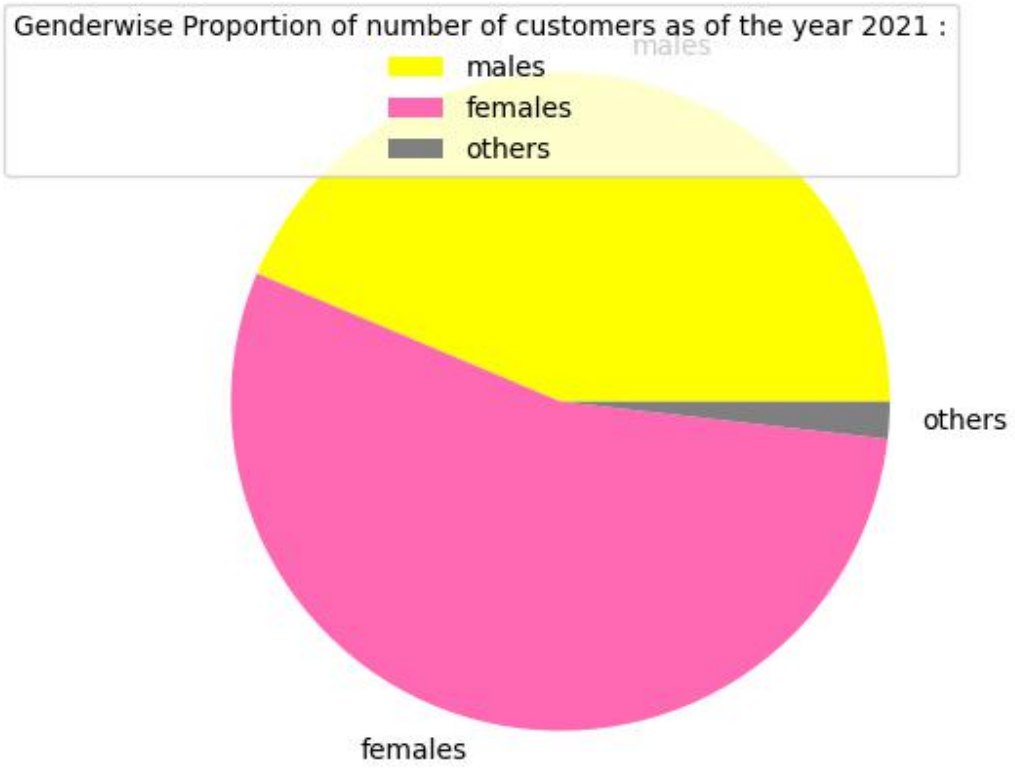
Total number of users v/s year



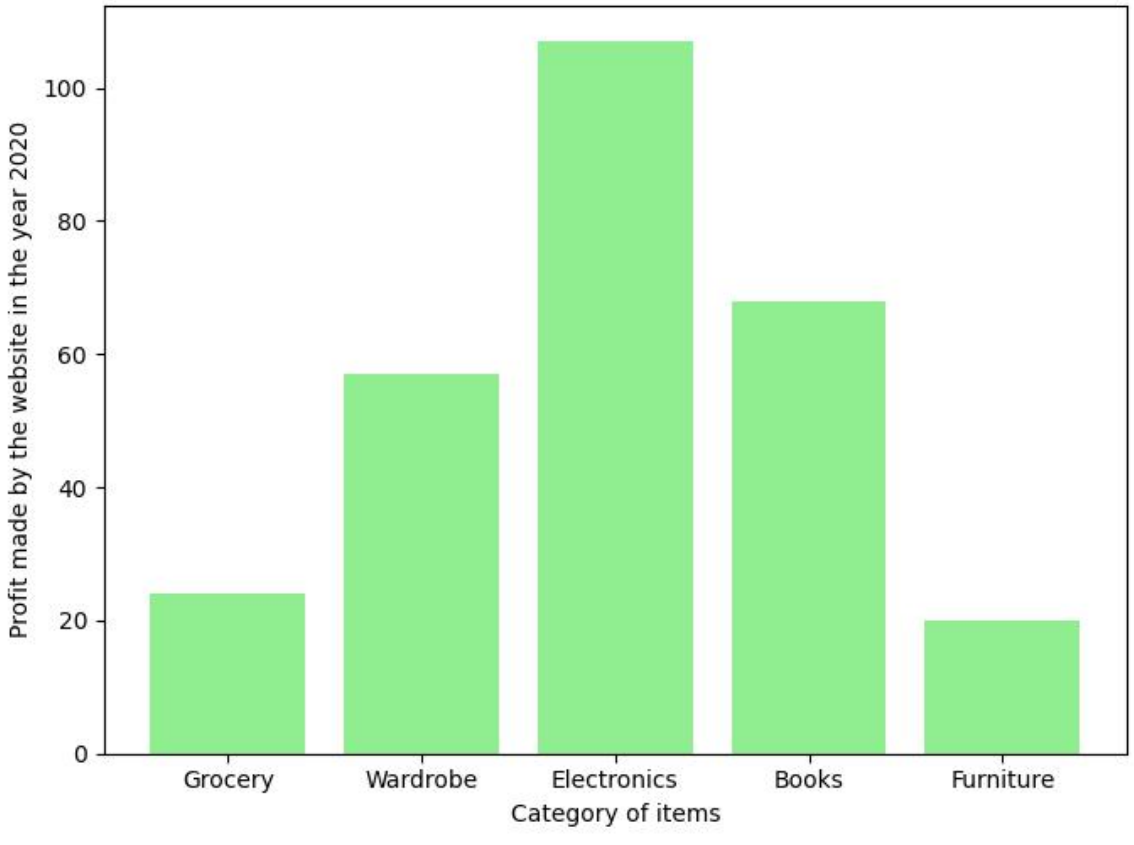
Number of sellers according to countries



Customer proportion(Gender wise)

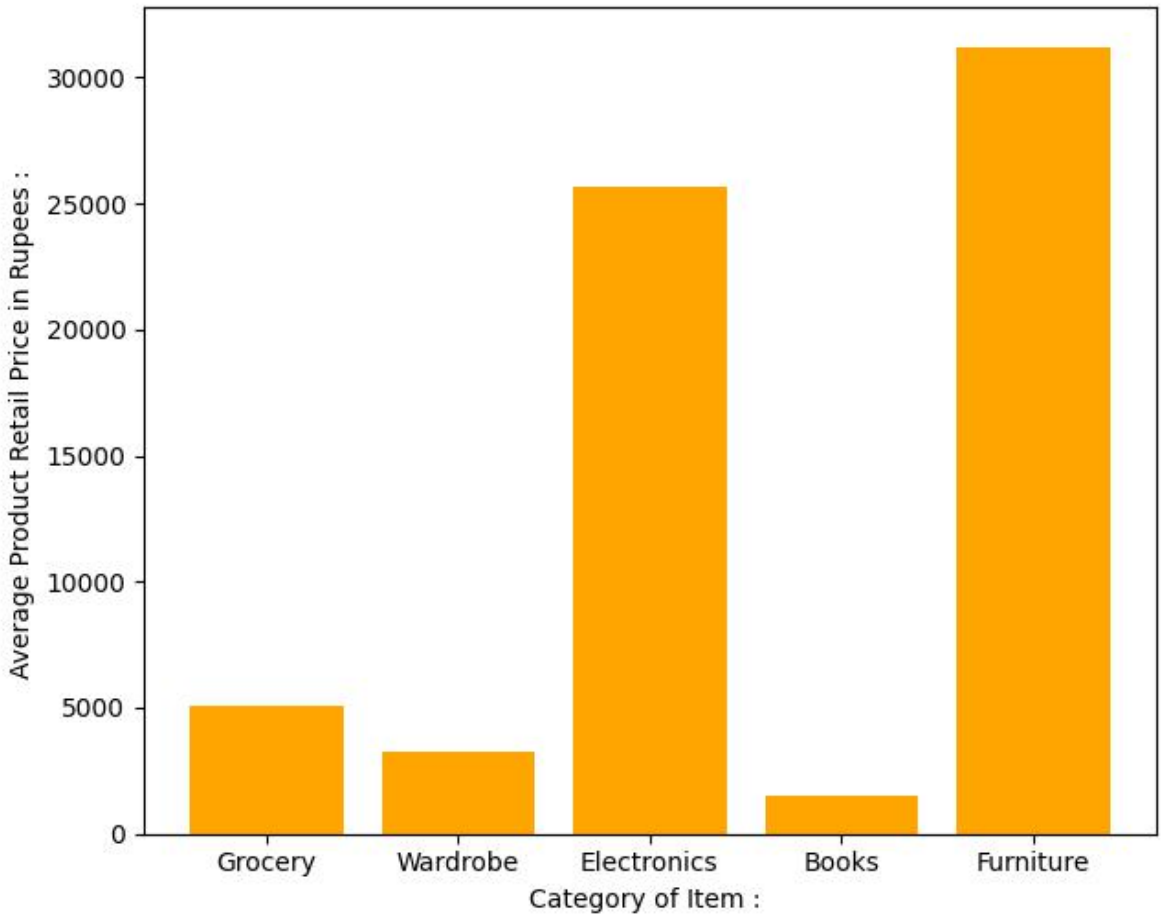


Profit made by website(in lakhs) v/s Category of Product

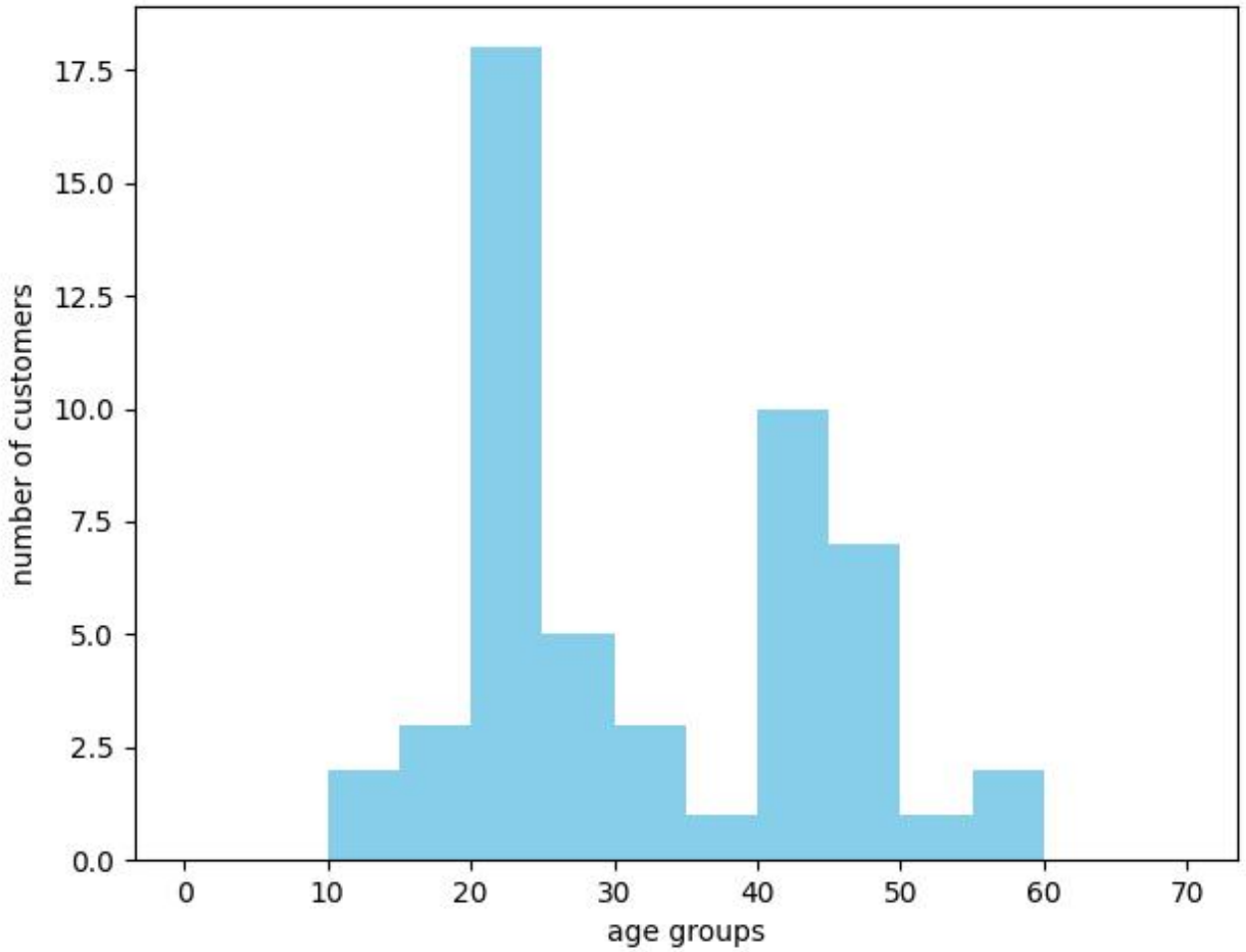


Some popular sellers v/s Mean customer rating

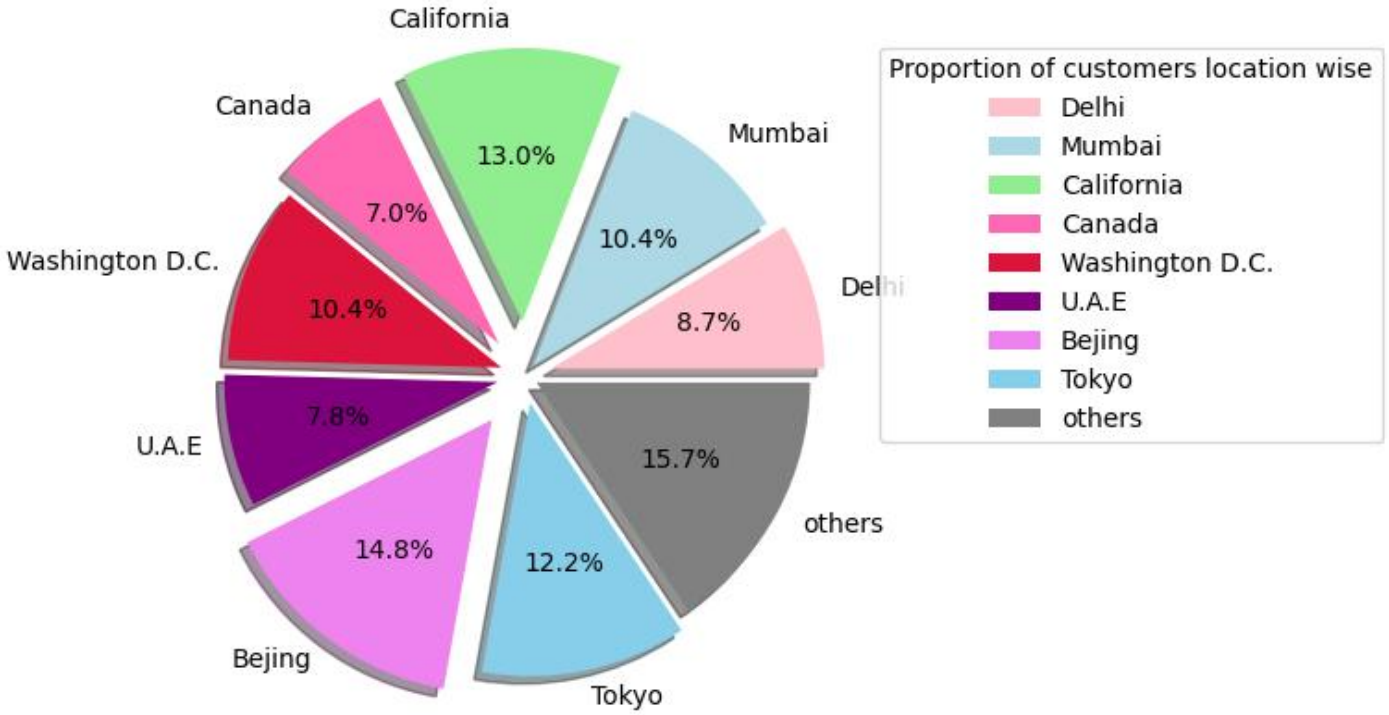
Avg. Product retail price v/s Category of products



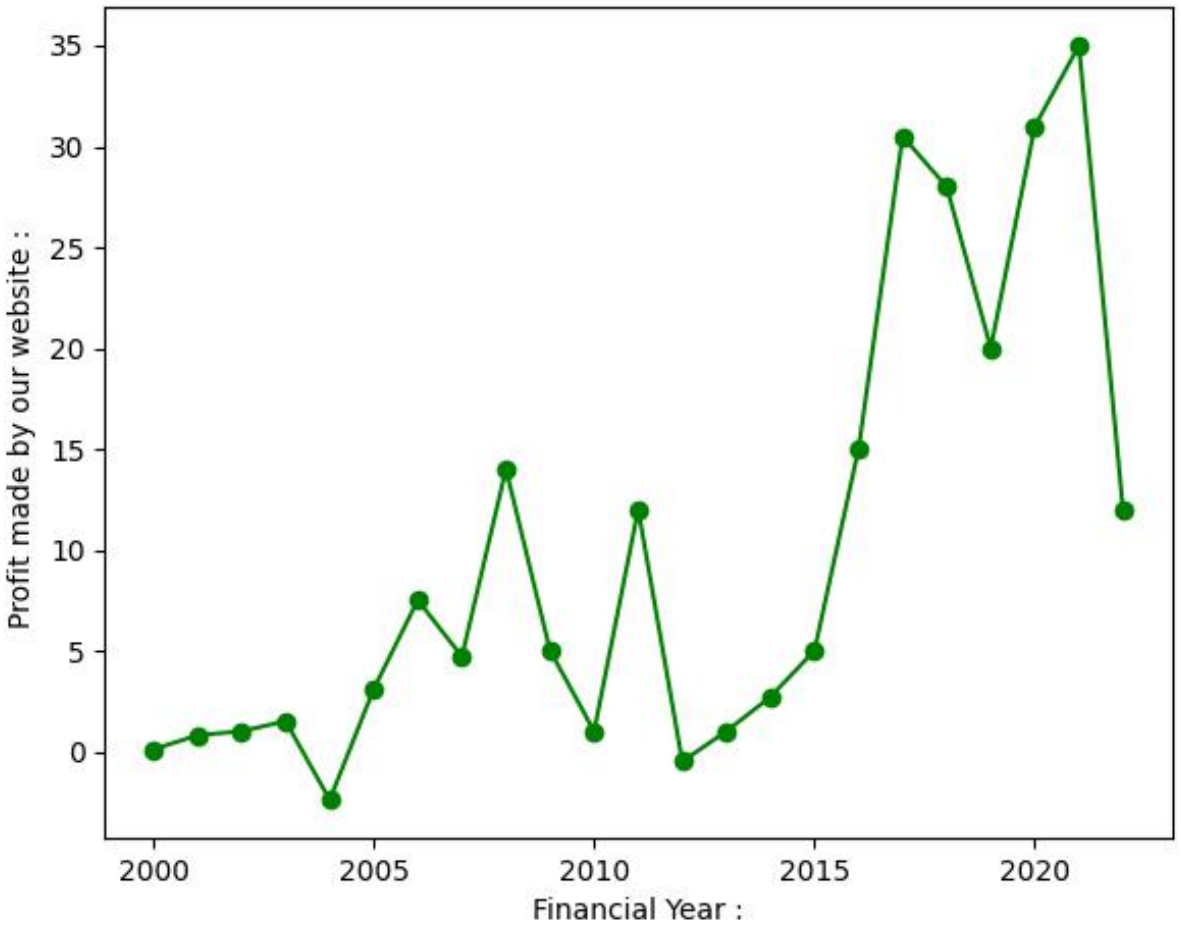
histogram of no. of users v/s age



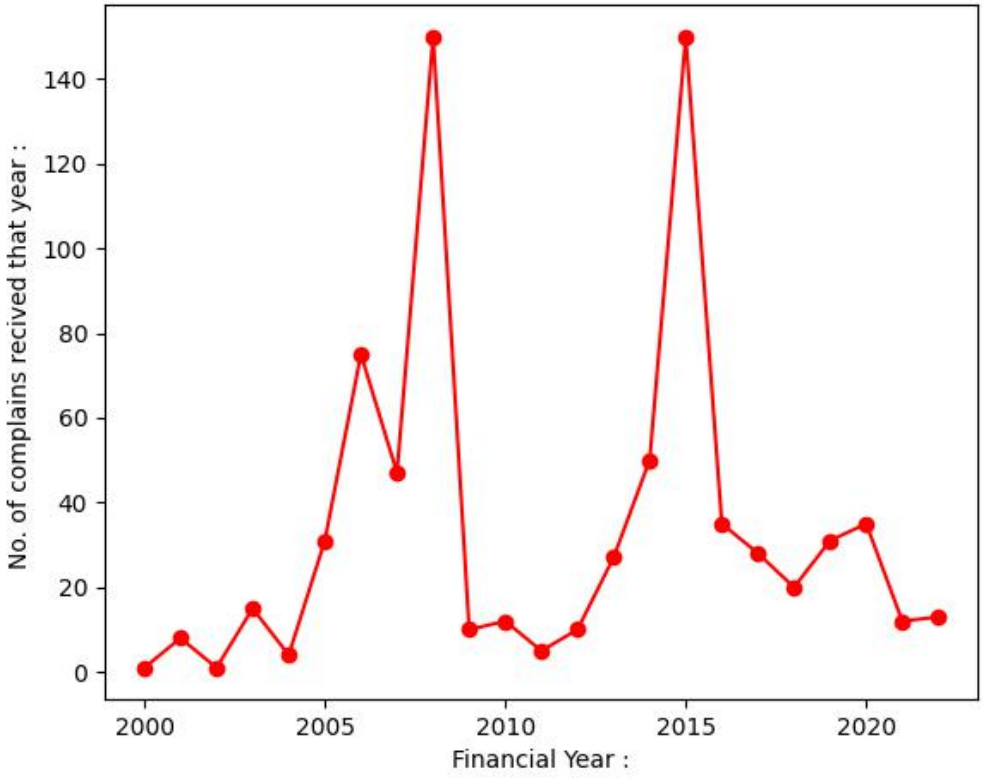
Proportion of customers location wise



Profit made(in thousand rupees) v/s year



Number of complains received



HILIGHTED CODE FROM VS CODE

```
#importing the connector :-
#from distutils.util import execute
import mysql.connector as c
#defining the database :-
#Note: the name of database is Eshop. you have to create database Eshop to run this
program.
database = c.connect(host = "localhost", user = 'root', password = "password",
database = 'eshop')
#defining cursor(it will be used to excute SQL commands) :-
cursor = database.cursor(buffered=True)
print('You are connected to the server eshop successfully.')
cursor.execute('CREATE TABLE IF NOT EXISTS items(item_id INT(4) PRIMARY KEY NOT
NULL,item_name VARCHAR(50),category VARCHAR(30),seller_id INT(4) NOT NULL,price
INT(8) NOT NULL,date_added DATE,rating FLOAT(2,1))' )
cursor.execute('CREATE TABLE IF NOT EXISTS customers(customer_id INT(4) PRIMARY KEY
NOT NULL,customer_name VARCHAR(30) NOT NULL,customer_age INT(3),gender
ENUM("male","female","others"),email VARCHAR(30),mobile VARCHAR(12),city
VARCHAR(20))')
cursor.execute('CREATE TABLE IF NOT EXISTS delboy(delboy_id INT(4) PRIMARY KEY NOT
NULL,name VARCHAR(30) NOT NULL,city VARCHAR(20),mobile VARCHAR(12),salary
INT(4),rating float(2,1))')
cursor.execute('CREATE TABLE IF NOT EXISTS sellers(seller_id INT(4) PRIMARY KEY NOT
NULL,name VARCHAR(30) NOT NULL,city VARCHAR(20),mobile VARCHAR(12),email
VARCHAR(30),rating float(2,1))')
cursor.execute('CREATE TABLE IF NOT EXISTS orders(order_id INT(4) PRIMARY KEY NOT
NULL,item_id INT(4) NOT NULL,customer_id INT(4) NOT NULL,date_placed DATE,delboy_ID
INT(4) NOT NULL)')
cursor.execute('CREATE TABLE IF NOT EXISTS complains(complain_id INT(4) PRIMARY KEY
NOT NULL,order_id INT(4) NOT NULL,subject VARCHAR(20) NOT NULL)')

#MAKING THE MENU(USER INTERFACE) FOR FRONT END
def menu():
    print("--- menu opened ---")
    a = int(input("to add data please type : 1 \nto delete data please type : 2 \nto
exit this menu, please type 3 \nResponse : "))
    if (a==1):
        b= int(input("type number : \n'1' to add customer data \n'2' to add seller
data \n'3' to add item data \n'4' to add order data \n'5' to add delboy data \n'6'
to add complain. \nResponse : "))
        if (b==1):
            add_customer()
        elif (b==2):
            add_seller()
        elif (b==3):
            add_item()
        elif (b==4):
            add_order()
        elif (b==5):
            add_delboy()
        elif (b==6):
            add_complain()
        else:
```

```
    elif (a==2):
            b= int(input("type number : \n'1' to delete customer data \n'2'
    to delete seller data \n'3' to delete item data \n'4' to delete order
    data \n'5' to delete delboy data \n'6' to delete complain \nResponse :
    "))
            if (b==1):
                delete_customer()
            elif (b==2):
                delete_seller()
            elif (b==3):
                delete_item()
            elif (b==4):
                delete_order()
            elif (b==5):
                delete_delboy()
            elif (b==6):
                delete_complain()
            else:
                print('please enter a valid response')
                menu()

        elif(a==3):
            print('Thank You! for using this program, Have a nice day! \n
    you can now exit or call functions manually. \nlist of the
    functions(prefix add_ OR delete_ before them) :
    \ncustomer\nseller\nitem\norder\ndelboy\ncomplain\n--- menu closed ---
    ')
        else:
            print('please enter a valid response')
            menu()

    #DEFINING FUNCTIONS FOR FEEDING DATA IN THE TABLES OF THE SERVER


    def add_customer():
        print("add_customer() function started.")
        c_id = input('enter customer ID  : ')
        name = input('enter customer name : ')
        age = input("enter customer age : ")
        gender = input("enter gender of customer : ")
        email = input('enter customer email ID : ')
        mobile = input('enter customer mobile number : ')
        city = input('enter customer city : ')
        data = (c_id,name,age,gender,email,mobile,city)
        statement = "INSERT INTO customers VALUES(%s,%s,%s,%s,%s,%s,%s)"
        cursor.execute(statement,data)
        database.commit()
        print('data inserted to table customer successfully!')
        b = input("do you want ot add more data to THIS table? (y/n) : ")
        if (b=='y'):
            add_customer()
        elif (b=='n'):
            menu()
        else:
            print("please enter a valid response. (you exited the menu. to
```

```
def add_seller():
    print("add_seller() function started.")
    s_id = input('enter seller ID  : ')
    name = input('enter seller name : ')
    city = input('enter seller city : ')
    mobile = input('enter seller mobile number : ')
    email = input('enter seller email ID : ')
    rating = input('enter the rating of seller : ')
    data = (s_id,name,city,mobile,email,rating)
    statement = "INSERT INTO sellers VALUES(%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table sellers successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_seller()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu.
to call back use menu())")

def add_item():
    print("add_item() function started.")
    item_id = input('enter item ID  : ')
    name = input('enter item name : ')
    cat = input('enter category of item : ')
    sid = input('enter seller_id of item seller : ')
    price = input('enter price of item : ')
    date_added = input('enter date added : ')
    rating = input('enter rating of item : ')
    data = (item_id,name,cat,sid,price,date_added,rating)
    statement = "INSERT INTO items VALUES(%s,%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table items successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_item()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu.
to call back use menu())")
```

```
def add_delboy():
    print("add_delboy() function started.")
    did = input('enter delboy ID  : ')
    name = input('enter delboy name : ')
    city = input('enter delboy city : ')
    mobile = input('enter delboy mobile number : ')
    salary = input('enter delboy salary : ')
    rating = input('enter the rating of delboy : ')
    data = (did,name,city,mobile,salary,rating)
    statement = "INSERT INTO delboy VALUES(%s,%s,%s,%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print('data inserted to table delboy successfully!')
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_delboy()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to
call back use menu())")

def add_complain():
    print("add_complain() function started.")
    cid = input("enter complain ID : ")
    oid = input("enter order id : ")
    sub = input("enter the subject of complain : ")
    data = (cid,oid,sub)
    statement = "INSERT INTO COMPLAINS VALUES(%s,%s,%s)"
    cursor.execute(statement,data)
    database.commit()
    print("data inserted to the table COMPLAINS successfully!")
    b = input("do you want ot add more data to THIS table? (y/n) : ")
    if (b=='y'):
        add_complain()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. toa
call back use menu())")
```

```
#DEFINING FUNCTIONS TO DELETE DATA FROM TABLES OF DATABASE :-
def delete_customer():
    print("delete_customer() function started.")
    cid = input('enter customer_id : ')
    cursor.execute('DELETE FROM customers WHERE customer_id=%s',(cid,))
    database.commit()
    print('deleted the customer data of customer_id : ',cid,'from table
customers successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) :
")
    if (b=='y'):
        delete_customer()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to
call back use menu())")

def delete_seller():
    print("delete_seller() function started.")
    sid = input('enter seller_id : ')
    cursor.execute('DELETE FROM sellers WHERE seller_id=%s',(sid,))
    database.commit()
    print('deleted the seller data of seller_id : ',sid,' from table
sellers successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) :
")
    if (b=='y'):
        delete_seller()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to
call back, use menu())")

def delete_item():
    print("delete_item() function started.")
    iid = input('enter item_id : ')
    cursor.execute('DELETE FROM items WHERE item_id=%s',(iid,))
    database.commit()
    print('deleted the item data of item_id : ',iid,' from table items
successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) :
")
    if (b=='y'):
        delete_item()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to
call back, use menu())")
```

```
def delete_order():
    print("delete_order() function started.")
    x = input('enter order_id : ')
    cursor.execute('DELETE FROM orders WHERE order_id=%s',(x,))
    database.commit()
    print('deleted the item data of order_id : ',x,' from table orders successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_order()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use
menu())")

def delete_delboy():
    print("delete_delboy() function started.")
    x = input('enter delboy_id : ')
    cursor.execute('DELETE FROM delboy WHERE delboy_id=%s',(x,))
    database.commit()
    print('deleted the data of delboy_id ',x,' from table delboy successfully!')
    b = input("do you want ot delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_delboy()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use
menu())")

def delete_complain():
    print("delete_complain() function started.")
    x = input('enter complain_id : ')
    cursor.execute('DELETE FROM complains WHERE complain_id=%s',(x,))
    database.commit()
    print('deleted the data of complain_id ',x,' from table complains successfully!')
    b = input("do you want to delete more data to THIS table? (y/n) : ")
    if (b=='y'):
        delete_complain()
    elif (b=='n'):
        menu()
    else:
        print("please enter a valid response. (you exited the menu. to call back, use
menu())")

#FINALLY, SERVING THE PROGRAM TO USER : -
print("* * * * * * * WELCOME TO THE DATABASE MANAGEMENT USER INTERFACE OF ESHOP * * * *
* * *")
print("(created by Mohammad Maasir @ date 13th Feb,2022, as a school project.)")
menu()
"""
although this program have a menu, we can also call the functions defined here to
add or delete data when this program is running in the IDLE shell, manually (when you
are outside menu i.e. exit the menu.)
************************* END OF THE CODE *********************************
"""
```

BIBLIOGRAPHY

1. https://www.mysqltutorial.org/
2. https://dev.mysql.com/doc/refman/5.6/en/
3. https://www.geeksforgeeks.org/
4. https://www.w3schools.com/sql/
5. https://www.w3schools.com/python/
6. https://app.diagrams.net/ (or draw.io , used for ER Diagram)
7. https://www.youtube.com/c/CodeWithHarry
8. https://codewithharry.com/
9. https://www.youtube.com/c/TechWithTim
10. https://matplotlib.org/stable/tutorials/introductory/pyplot.html
11. https://code.visualstudio.com/
12. NCERT textbooks class 11th and 12th : https://ncert.nic.in/textbook.php
13. Sumita Arora class 12th

Git Hub Page Link

This project is available on my GitHub page. link:
https://github.com/maasir554/maasir554/blob/main/frontend.py

For viewing more projects made by me, refer to my GitHub page:
https://github.com/maasir554/

Conclusion

In this project, a SQL database was created, and connected to python successfully.
Various operations such as insertion of values, deletion of values and data analysis was done using Python-MySQL connector, and Matplotlib.

Thank You

(End of this document)