GLOBAL SCHOOL

INVESTIGATORY PROJECT
ON
HOTEL MANAGEMENT

Submitted for

ALL INDIA SENIOR SECONDARY

CERTIFICATE EXAMINATION

By

Miss. Shifa Shameemah

Register No:

SUBMITTED TO THE

DEPARTMENT OF COMPUTER

SCIENCE

GLOBAL SCHOOL

New S.no.165/28, Thirumagal Nagar, Rajakilpakkam, Chennai – 600073

SUBJECT: Computer Science

INVESTIGATORY PROJECT

Certified to be the bonafide Project work done by

---------- of Standard of Grade 12

during the academic year 2023 - 2024

REGISTER No: _____

PRINCIPAL **TEACHER IN CHARGE**

Submitted for the practical examination held on

January/February 2024 at

TRILEAVES GLOBAL School, Chennai – 73

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I would like to extend my sincerest thanks to my teacher, Mrs. Ashwathy Sumesh for her continuous support and guidance throughout the entire duration of the project. Thanks to her, the project was successfully completed without further complications.

I would also like to express my gratitude to our Correspondent Mrs. Parameswari S., our Principal Mrs. Geethika Sivaramakrishnan, and Vice Principal Mrs. Monica R who made this project possible with their extensive support and patience.

Last but not least, I would like to thank my cousins Mohammed Aneef and Abrar Ahmed, and my best friend Supriya Rao for helping me with this project, Without them it would have been a huge deal for me to complete this project. They really helped me by guiding me in this project, and with many other procedures that were involved in this project.

Index

Introduction

Python:

Python is a high-level, versatile, dynamically typed programming language known for its simplicity, readability, and versatility. Created by Guido van Rossum and first released in 1991, Python has become one of the most popular and widely used languages in the world. The design philosophy emphasizes code readability and clean, easy-to-understand syntax; which makes it ideal for both beginners and experienced developers. Python is a popular programming language. It is a general-purpose application. It has been designed with an emphasis on the readability of the code, and its syntax allows users to express their content in fewer lines. Python is a programming language that allows you to work quickly and collaborate more efficiently. It is designed for web development, software development, math, and writing. Python's syntax allows developers to program in fewer lines than other programming languages. Python runs on an interpreter; this means that the code can be run as soon as it is written. This means that prototyping can be very fast. Python can be ported to methods, object-oriented, or functions.

MySQL:

MySQL is a widely used open-source relational database management system (RDBMS) known for its speed, reliability, and ease of use. It plays an important role in the management and storage of design information, making it the technology used for a variety of software applications, from small projects to large enterprises. MySQL is a relational database management system. A database is a standalone application that stores data. We use a relational data management system (RDMS) to store and manage large amounts of data. It is called a relational database because all data is stored in different tables and relationships. It is cross-platform. It is popular among all databases and is known for its popular data in use today. MySQL itself is a very powerful service. It does most of the work on the most expensive and powerful packages. MySQL is very fast and works well even with large files. Knowledge management is the process of receiving, storing, organizing, and

managing information created and collected by an organization. The information management system consists of a combination of different activities that share the goal of ensuring that the information in the company's system is accurate, valid, and usable.

Learning Objective

The objective of this project is to effectively manage the databases and files in a Hotel Management system with the use of two platforms: Python and MySQL. This project helps with the purpose of organizing and segregating information easily with the mentioned platforms. The supporting features of this project to make data storage more accessible is simply done by:

- Registering the Customer's ID in the Hotel
- Looking for the Customer ID and the details entered
- Updating the Customer's ID in the Hotel
- Deleting the Customer's ID in the Hotel
- Room booking details of the Customer
- Buffet booking details of the Customer
- Viewing all the booking details of the Customer
- Exiting the program

Aim

- To understand how bookings are done in Hotel Management using Python and MySQL
- To understand how the records and details of the Customers are stored in the system
- To make booking more accessible by understanding the procedure of Hotel Management
- To use databases effectively and efficiently
- To understand the role of programming language and relational database management systems in terms of Hotel Management

Software and Hardware Requirements

Software:

- Python IDLE
- Web browser
- Python Shell 3.6.9
- MySQL Workbench 3.0 CE

Hardware:

- A computer system
- Laptop
- 8.00 GB of RAM
- Wifi Router

About Python

- Python is a general-purpose programming language. The design concept emphasizes the readability of the code through the use of key indentation.
- Developed by Guido van Rossum and started as an extension language for ABC programming in the 1980s, it was first released as Python 0.9.0 in 1991.
- Python 2.0 was released in 2000 and brought new features like name recognition. , usage counting, and Unicode support. Released in 2008, Python 3.0 is a major overhaul that has not been replicated with earlier versions.
- Python interpreters are available for many operating systems. An international community of programmers develops and maintains a free, open-source application, Python.
- The Python Software Foundation is a non-profit organization that manages and directs resources for the development of Python.
- **Readability:** Python's syntax uses indentation (space) to identify blocks of code, making it visually clean and easy to understand. This reduces errors by promoting policy consistency and readability.

- **High-level languages:** Python abstracts many low-level concepts, making it efficient and allowing developers to focus on solving problems rather than managing memory or hardware.
- **Interpretation:** Python is an interpretation language that does not require compilation steps. You can make the development process faster by running Python code directly.
- **Cross-platform:** Python, Windows, macOS, Linux, etc. It can be used on many platforms, including, and is easily portable.
- **System library:** Python comes with an excellent library that provides models and functions for many tasks such as data management, networking, and data management. This saves development time by reducing the need for external libraries.
- **Dynamic typing:** Python uses dynamic typing, which allows variables to change their data type at runtime. This change can simplify the code but should be handled with care to avoid errors.
- Python's features include:
    - Red uses multiple languages
    - Has dynamic typing
    - Python programs are generally smaller than other programming languages such as Java.
    - The biggest strength of Python is a huge collection of standard libraries that can be used for the following:
        - Machine Learning
        - GUI Applications (like Kivy, Tkinter, PyQt, etc.)

About MySQL

- MySQL is a widely used open-source relational database management system (RDBMS) known for its speed, reliability, and ease of use. It plays an important role in the management and storage of design information, making it the technology used for a variety of software applications, from small projects to large enterprises. Relational databases store data in separate tables instead of putting all the data in one big building. The database system is organized into physical files for speed. The data structure provides a flexible environment by containing objects such as data, comments, rows, and columns. You can set different association rules between different data, such

as one-to-one, one-to-many, custom, required, or optional, and the "pointers" of tables. These data management policies ensure that with well-structured data, your app will never find inconsistent, duplicate, orphaned, closed, or missing data.

- MySQL is fast, reliable, scalable, and easy to use. It was originally designed for high-speed processing of large files and has been used in harsh production environments for many years.
- MySQL has the following features:
    - **Ease of use:** Developers can install MySQL in minutes and the database is easy to manage.
    - **Reliability:** MySQL is one of the most mature and widely used databases. It has been tested for more than 25 years in many cases, including many of the world's largest companies. Organizations rely on MySQL to run business-critical applications for its reliability
    - **Scalability:** MySQL scales to meet application access needs. MySQL's native replication architecture allows organizations like Facebook to scale applications to support millions of users.
    - **Performance:** MySQL HeatWave is faster and cheaper than other database services, as verified by many industry standards.
    - **Many Features:** MySQL provides a full set of native, integrated tech technologies for high availability and disaster recovery. Clients for negotiating key business practices and high-level service contracts
    - **Security:** Information security must be protected and comply with business and government regulations; MySQL Enterprise Edition provides the best security, including authentication/authorization. transparent data encryption, auditing, data masking, and database firewall.
    - **Flexibility:** MySQL data storage gives users the greatest flexibility in building traditional SQL and non-SQL database applications. Developers can mix and match data relationships within the same database and application.

Programs:

MySQL Code:

```
use BeachBay_Hotel;
show tables;
Select*from Customer_details;
Select*from Room;
Select*from Buffet;
Select*from Bills;
```

Python Code:

#This is the Final Project call for CS Hotel Management Project:

#CS project class 12 Shifa Shameemah

#Name: Shifa Shameemah

#Class: 12

#Project: Hotel Management

#Connecting Python and MySQL

import mysql.connector

data=mysql.connector.connect(host='localhost',user='root',password='bossy',database="BeachBay_Hotel")

point=data.cursor()

#point.execute("CREATE DATABASE BeachBay_Hotel")

#point.execute("SHOW DATABASES")

#point.execute("CREATE TABLE Customer_details (Customer_id INT PRIMARY KEY, Customer_Name VARCHAR(225), Customer_Address VARCHAR(225), Customer_Age INT, Customer_Phone VARCHAR(225), Customer_Mail VARCHAR(225))")

#point.execute("CREATE TABLE Room (Customer_id INT PRIMARY KEY, Room_type VARCHAR(10), Type_full VARCHAR(50),Price_room

```
INT,Price_service INT, Num_nigths INT, Total_price INT)")
#point.execute("CREATE TABLE Buffet (Customer_id INT PRIMARY KEY, Meal_time VARCHAR(10), Num_Nights INT, Buffet_Price INT, Table_Price INT, Total_Price INT)")
#point.execute("CREATE TABLE Bills (Customer_id INT, Room_Total INT, Buffet_Total INT, Total_Amount INT)")

#Declaring Global Variables
Name=""
Customer_id=""
Address=""
Age=""
Phone=""
Mail=""
GetOne=""
GetAll=""
ch=""
Records=""
SQL=""
Customer=""
sb=1
db=2
sbl=3
dbl=4
sdb=5

#Helper Functions
def Customer_exists(Customer_id):
    GetOne = "SELECT * FROM Customer_details WHERE Customer_id=%s"
    point.execute(GetOne, [Customer_id])
    Records = point.fetchall()
```

```
    return len(Records) != 0

def Room_exists(Customer_id):
    GetRoom = "SELECT * FROM Room WHERE Customer_id=%s"
    point.execute(GetRoom,[Customer_id])
    Rooms=point.fetchall()
    return len(Rooms)!=0

def BuffetBooking_exists(Customer_id):
    GetTable = "SELECT * FROM Buffet WHERE Customer_id=%s"
    point.execute(GetTable,[Customer_id])
    Tables=point.fetchall()
    return len(Tables) !=0

def Bills_exists(Customer_id):
    GetBill = "SELECT * FROM Bills WHERE Customer_id=%s"
    point.execute(GetBill,[Customer_id])
    Bills=point.fetchall()
    return len(Bills) !=0

#Entering Customer Details into Customer_details
def Customer_details():

    Customer_id=int(input("Enter a customer ID:\n"))
    if (Customer_exists(Customer_id))==True:
        print('Customer ID already exists. Pleasse try again.')
        Customer_details()
    Name=input("Enter your Name:\n")
    Address=input("Enter your Address:\n")
    Age=int(input("Enter your Age:\n"))
```

```
    Phone=input("Enter your Contact Number:\n")
    Mail=input("Enter your E-Mail ID:\n")
    Details=[Customer_id,Name,Address,Age,Phone,Mail]
    SQL="INSERT INTO Customer_details (Customer_id, Customer_Name,
Customer_Address, Customer_Age, Customer_Phone,
Customer_Mail)VALUES(%s,%s,%s,%s,%s,%s)"
    point.execute(SQL,Details)
    data.commit()
    print("Data Entered Sucessfully!")
    main_menu()


#Updating the Customer Details
def Update_details():
    Customer_id=int(input("Enter a customer ID:\n"))
    if (Customer_exists(Customer_id))==False:
        print('Customer ID not found. Try again.')
        Update_details()
    else:
        Name1=input("Enter the new name you wish to update:\n")
        Address1=input("Enter the new address you wish to update:\n")
        Age1=int(input("Enter the new age you wish to update:\n"))
        Phone1=int(input("Enter the new phone number you wish to update:\n"))
        Mail1=input("Enter the e-mail ID you wish to update:\n")
        Details=[Name1, Address1, Age1, Phone1, Mail1,Customer_id]
        Check="UPDATE Customer_details SET Customer_Name=%s,
Customer_Address=%s, Customer_Age=%s, Customer_Phone=%s,
Customer_Mail=%s WHERE Customer_id=%s"
        point.execute(Check,Details)
        data.commit()
        main_menu()
```

```
#Deleting the Data of Customer
def Delete_data():
    Customer_id=int(input("Enter a customer ID:\n"))
    if (Customer_exists(Customer_id))==False:
        print("Customer ID not found. Try again.")
        Delete_data()
    Customer=(Customer_id,)
    delete="DELETE FROM Customer_details WHERE Customer_id=%s;"
    delete_rooms="DELETE FROM Room WHERE Customer_id=%s;"
    delete_bill="DELETE FROM Bills WHERE Customer_id=%s;"
    delete_buffet="DELETE FROM Buffet WHERE Customer_id=%s"
    ch=input("Are you sure you want to delete your information? Yes/No:\n")
    if ch=="Yes" or ch=='yes':
      point.execute(delete_rooms, Customer)
      point.execute(delete, Customer)
      point.execute(delete_bill, Customer)
      point.execute(delete_buffet, Customer)
      data.commit()
      print("Data successfully deleted. :)")
    else:
        print("Cancelled.")
    main_menu()

#To Book a Room
def book_room():
    Customer_id=int(input("Enter a customer ID:\n"))
    if (Customer_exists(Customer_id))==False:
        print("Customer ID not found. Try again.")
        book_room()
```

```
    if (Room_exists(Customer_id))==True:
        print('You already have a room!')
        main_menu()

    RoomType= input("Enter the room type. Type the character near the room: \n"
                    "S - Single bed, Price 1000/Night. \n"
                    "D - Double bed, Price 1300/Night. \n"
                    "SL - Single bed large, Price 2200/Night. \n"
                    "DL - Double bed large, Price 2500/Night. \n"
                    "DSL - Double bed + 1 single bed, Price 3000/Night.\n\n"
                    "NOTE: Room service costs Rs5000/night, regardless of room
booked.\n")

    if RoomType.upper()=='S':
        price=1000
        Type='Single bed'
    elif RoomType.upper()=='D':
        price=1300
        Type='Double bed'
    elif RoomType.upper()=='SL':
        price=2200
        Type=' Single bed Large'
    elif RoomType.upper()=='DL':
        price= 2500
        Type='Double bed Large'
    elif RoomType.upper()=='DSL':
        price=3000
        Type="Double bed + 1 Single bed"
    else:
```

```
        print("Wrong Type has been entered. Please try again.")
        book_room()
    Num_nights=int(input("Enter the number of nights for your stay at the
BeachBay hotel:\n"))

    if Num_nights<=0:
        print("Invalid Input! Please try again")
        book_room()

    Room_price = price*Num_nights
    Room_service_price= 500*Num_nights
    Total_price= Room_price + Room_service_price

    SQL = "INSERT INTO Bills (Customer_id, Room_Total, Buffet_Total,
Total_Amount) VALUES (%s, %s, 0, %s)"
    point.execute(SQL, (Customer_id, Room_price, Total_price))
    data.commit()



    print("\n" + "-" * 70 + "\n")
    print("Booking Summary:")
    print("\n" + "-" * 70 + "\n")
    print("| {:<30} | {:<30} |".format("BOOKINGS", "CUSTOMER'S BOOKING
TYPE"))
    print("| {:<30} | {:<30} |".format("-" * 30, "-" * 30))
    print("| {:<30} | {:<30} |".format("Type of Room", Type))
    print("| {:<30} | {:<30} |".format("Room Price", Room_price))
    print("| {:<30} | {:<30} |".format("Room Service Price", Room_service_price))
    print("| {:<30} | {:<30} |".format("Total Bill Amount", Total_price))
    print("\n" + "-" * 70 + "\n")
```

```
    ch= input ("We hope you are satisfied with the bill, would you like to like to confirm your booking Y/N:\n")
    if ch in "Yy":
        SQL="INSERT INTO Room (Customer_id,Room_type, Type_full, Price_room, Price_service, Num_nigths, Total_price) VALUES(%s,%s,%s,%s,%s,%s,%s)"

Val=[Customer_id,RoomType.upper(),Type,Room_price,Room_service_price,Num_nights,Total_price]
        point.execute(SQL,Val)
        data.commit()
        print("Room Data Entered  Successfully! :)")
        print("Returing to main menu.....")
        main_menu()
    else:
        print("Booking cancelled")
        main_menu()


def Buffet_booking():
    Customer_id=int(input("Enter a customer ID:\n"))
    if Customer_exists(Customer_id)== False:
        print("Customer ID not found. Try again.")
        Buffet_booking()


    if (BuffetBooking_exists(Customer_id))==True:
        print('You already have a table!')
        main_menu()



    Meal=input("Enter the meal timing you want to place your buffet table:\n"
               "B - Breakfast, Timings from 6:00 A.M. - 10:00 A.M., Price- 1300 /per day, \n"
```

```
            "L - Lunch, Timings from 12:00 P.M. - 3:00 P.M., Price- 1500 /per
day, \n"
            "D - Dinner, Timings from 6:00 P.M. - 10:00 P.M.\n"
            "Please Enter the Buffet timings you prefer, B - Breakfast, L - Lunch,
D - Dinner: \n"
            "Price- 2000 /per night \n\nNOTE: Table costs Rs1000/table for
booking.\n")

    if Meal.upper()=='B':
        cost = 1300
        Time = 'Breakfast'
    elif Meal.upper()=='L':
        cost = 1500
        Time = 'Lunch'
    elif Meal.upper()=='D':
        cost = 2000
        Time = 'Dinner'
    elif Meal.upper()=='BL':
        cost = 1300 + 1500
        Time = 'Breakfast and Lunch'
    elif Meal.upper()=='BLD':
        cost = 1300 + 1500 + 2000
        Time = 'Breakfast, Lunch and Dinner'
    else:
        print("Wrong Time has been entered. Please try again.")
        Buffet_booking()

    Nights=int(input('Enter the number of Nights your stay in BeachBay Hotel:\n'))

    if Nights<=0:
        print("Invalid input. Try again.")
```

```
        Buffet_booking()


    Buffet_price = cost*Nights
    Table_price =  1000*Nights
    Buffet_Total_price= Buffet_price+Table_price


    if Bills_exists(Customer_id):
        UpdateBill = "UPDATE Bills SET Buffet_Total = %s WHERE Customer_id
= %s"
        point.execute(UpdateBill, (Buffet_Total_price, Customer_id))
        data.commit()


    print("\n" + "-" * 70 + "\n")
    print("Buffet Booking Summary:")
    print("\n" + "-" * 70 + "\n")
    print("| {:<30} | {:<30} |".format("BUFFET DETAILS", "CUSTOMER
BOOKINGS"))
    print("| {:<30} | {:<30} |".format("-" * 30, "-" * 30))
    print("| {:<30} | {:<30} |".format("Time of Meal", Time))
    print("| {:<30} | {:<30} |".format("Buffet Price", Buffet_price))
    print("| {:<30} | {:<30} |".format("Table Price", Table_price))
    print("| {:<30} | {:<30} |".format("Total Buffet Bill", Buffet_Total_price))
    print("\n" + "-" * 70 + "\n")


    ch=input("Are you satisfied with the bill and would like to confirm your
booking? Y/N:\n")
    if ch in "Yy":
        SQL="INSERT INTO
Buffet(Customer_id,Meal_time,Num_Nights,Buffet_price,Table_price,Total_pric
e) VALUES(%s,%s,%s,%s,%s,%s)"

Var=[Customer_id,Meal.upper(),Nights,Buffet_price,Table_price,Buffet_Total_pr
```

```
ice]
        point.execute(SQL,Var)
        data.commit()

        print("Buffet Data Entered Successfully! :)")
        print("Returning to main menu.....")
        main_menu()
    else:
        print("Buffet Booking cancelled")
        main_menu()

def Details_ofCustomer():
    Customer_id=int(input("Enter a customer ID:\n"))
    if Customer_exists(Customer_id):
        GetOne = "SELECT * FROM Customer_details WHERE Customer_id=%s"
        point.execute(GetOne, [Customer_id])
        Records = point.fetchall()
        if len(Records) > 0:
            print("\n" + "-" * 70 + "\n")
            print("Customer Information for Customer ID:", Customer_id)
            print("\n" + "-" * 70 + "\n")
            print("| {:<15} | {:<40} |".format("DETAILS", "CUSTOMER
INFORMATION"))
            print("| {:<15} | {:<40} |".format("-" * 15, "-" * 40))
            print("| {:<15} | {:<40} |".format("Customer ID", Records[0][0]))
            print("| {:<15} | {:<40} |".format("Name", Records[0][1]))
            print("| {:<15} | {:<40} |".format("Address", Records[0][2]))
            print("| {:<15} | {:<40} |".format("Age", Records[0][3]))
            print("| {:<15} | {:<40} |".format("Phone", Records[0][4]))
            print("| {:<15} | {:<40} |".format("Email", Records[0][5]))
```

```
if Room_exists(Customer_id):
    print("| {:<15} | {:<40} |".format("Room Booking",
                                       "You Have a Room Booking Done! "))
    print("| {:<15} | {:<40} |".format(" ",
                                       "To view Press V"))
    print("| {:<15} | {:<40} |".format(" ",
                                       "as mentioned in the main menu"))
else:
    print("| {:<15} | {:<40} |".format("Room Booking",
                                       "You Don't have a Room Booking,"))
    print("| {:<15} | {:<40} |".format(" ",
                                       "To make a Room Booking Press B"))
    print("| {:<15} | {:<40} |".format(" ",
                                       "as mentioned in the main menu"))

# Check if the customer has a buffet booking
if BuffetBooking_exists(Customer_id):
    print("| {:<15} | {:<40} |".format("Buffet Booking",
                                       "You Have a Buffet Booking Done!"))
    print("| {:<15} | {:<40} |".format(" ",
                                       "To view Press V"))
    print("| {:<15} | {:<40} |".format(" ",
                                       "as mentioned in the main menu"))
else:
    print("| {:<15} | {:<40} |".format("Buffet Booking",
                                       "You Don't have a Buffet Booking,"))
    print("| {:<15} | {:<40} |".format(" ",
                                       "To make a Buffet Booking Press M"))
    print("| {:<15} | {:<40} |".format(" ",
                                       "as mentioned in the main menu"))
```

```
            print("\n" + "-" * 70 + "\n")
        else:
            print('Customer Details Not Found!')
            Details_ofCustomer()
    else:
        print('Customer Details Not Found!')
        Details_ofCustomer()

def View_booking():

    Customer_id = int(input("Enter a customer ID:\n"))
    if Customer_exists(Customer_id) == False:
        print("Customer ID not found. Please make sure you have made a room
booking and try again.")
        book_room()

    GetRoom = "SELECT * FROM Room WHERE Customer_id=%s"
    point.execute(GetRoom, [Customer_id])
    Rooms = point.fetchall()
    GetBuffet = "SELECT * FROM Buffet WHERE Customer_id=%s"
    point.execute(GetBuffet, [Customer_id])
    Tables = point.fetchall()
    GetOne = "SELECT * FROM Customer_details WHERE Customer_id=%s"
    point.execute(GetOne, [Customer_id])
    Records = point.fetchall()
    GetBill = "SELECT * FROM Bills WHERE Customer_id=%s"
    point.execute(GetBill, [Customer_id])
    Bills = point.fetchall()
```

```
    if Room_exists(Customer_id) == False:
        print("You don't have a room booking!")
        print("Going back to the main menu")
        main_menu()
    if BuffetBooking_exists(Customer_id) == False:
        print("You don't have a Buffet booking!")
        booknow = input("Do you wish to make a Buffet booking? Y/N:")
        if booknow.upper() == 'Y':
            Buffet_booking()
        else:
            room_table = "| {:<30} | {:<30} |\n".format("BOOKINGS",
"CUSTOMER'S BOOKING TYPE") + "| {:<30} | {:<30} |\n".format("-" * 30, "-"
* 30) + "| {:<30} | {:<30} |\n".format("Type of Room", Rooms[0][2]) + "| {:<30} |
{:<30} |\n".format("Room Price", Rooms[0][3]) + "| {:<30} | {:<30}
|\n".format("Room Service Price", Rooms[0][4]) + "| {:<30} | {:<30}
|\n".format("Total Bill Amount", Rooms[0][6]) + "\n" + "-" * 60 + "\n"
            print(room_table)
            main_menu()


    Room_data = Rooms[0]
    Customer_data = Records[0]
    Buffet_data = Tables[0]
    Bill_data = Bills[0]
    Total_Bill = Room_data[6] + Buffet_data[5]


    print("\n" + "-" * 70 + "\n")
    print("Booking Data for Customer ID:", Customer_id)
    print("\n" + "-" * 70 + "\n")
    print("| {:<25} | {:<25} |".format("Details", "Customer Data"))
    print("| {:<25} | {:<25} |".format("-" * 25, "-" * 25))
    print("| {:<25} | {:<25} |".format("Customer Name", Customer_data[1]))
    print("| {:<25} | {:<25} |".format("Customer Phone", Customer_data[4]))
```

```
    print("| {:<25} | {:<25} |".format("Customer Email", Customer_data[5]))
    print("| {:<25} | {:<25} |".format("Type of Room", Room_data[2]))
    print("| {:<25} | {:<25} |".format("Room Price", Room_data[3]))
    print("| {:<25} | {:<25} |".format("Room Service Price", Room_data[4]))
    print("| {:<25} | {:<25} |".format("Total Room Bill", Room_data[6]))
    print("| {:<25} | {:<25} |".format("Days of Buffet", Buffet_data[2]))
    print("| {:<25} | {:<25} |".format("Meal Time", Buffet_data[1]))
    print("| {:<25} | {:<25} |".format("Buffet Price", Buffet_data[3]))
    print("| {:<25} | {:<25} |".format("Table Price", Buffet_data[4]))
    print("| {:<25} | {:<25} |".format("Total Buffet Bill", Buffet_data[5]))
    print("| {:<25} | {:<25} |".format("Total Billing", Total_Bill))
    print("\n" + "-" * 70 + "\n")
    print(' ')
    if Bills_exists(Customer_id) and len(Bills) > 0:
        Total_Bill = Room_data[6] + Buffet_data[5]
        print("\n" + "-" * 70 + "\n")
        print("| {:<25} | {:<25} |".format("Total Room Bill", Room_data[6]))
        print("| {:<25} | {:<25} |".format("Total Buffet Bill", Buffet_data[5]))
        print("| {:<25} | {:<25} |".format("Overall Total", Total_Bill))
        print("\n" + "-" * 70 + "\n")
        print(' ')
        main_menu()
    else:
        print("Billing details not found for this customer.")
        print("Going back to the main menu")
        main_menu()
    main_menu()

def main_menu():
    print("---------------*WELCOME TO BeachBay HOTELS*---------------")
```

```
    print("----------        *MAIN MENU*        ----------")
    var="yes"
    while var.lower()=='yes':
        ch=input("To Create a New Customer Account, Press N\n"
                 "To look for your details, Press S \n"
                 "To Update Existing Customer Account, Press U \n"
                 "To Delete Customer Account, Press D \n"
                 "To Book a Room, Press B \n"
                 "To View Your Booking Details, Press V \n"
                 "NOTE: TO VIEW YOUR BOOKING DETAILS YOU MUST HAVE
A ROOM BOOKING \n"
                 "To Make Your Buffet Booking Details, Press M \n"
                 "To Exit Menu, Press E \n")
        if ch.upper()=="N":
            Customer_details()
        elif ch.upper()=="S":
            Details_ofCustomer()
        elif ch.upper()=="U":
            Update_details()
        elif ch.upper()=="D":
            Delete_data()
        elif ch.upper()=="B":
            book_room()
        elif ch.upper()=="V":
            View_booking()
        elif ch.upper()=="M":
            Buffet_booking()
        elif ch.upper()=="E":
            print("Thank you for choosing  BeachBay Hotel, Visit us again!")
            exit()
```

```
        else:
            print("Something is Invalid, Redirecting to main menu.")
            main_menu()
        var=input('Do you want to execute again? "Yes/No":')
    else:
        print("Thank you so much for choosing BeachBay Hotel! Hope you visit us soon for the next vacation")


main_menu()
```

Output:

Creating New Customer Account:

- Creating a new Customer ID from scratch and the following output:

```
===== RESTART: C:\Users\newni\OneDrive\Desktop\X I 1\Final Project call.py =====
---------------*WELCOME TO BeachBay HOTELS*---------------
---------------        *MAIN MENU*          ----------------
To Create a New Customer Account, Press N
To look for your details, Press S
To Update Existing Customer Account, Press U
To Delete Customer Account, Press D
To Book a Room, Press B
To View Your Booking Details, Press V
NOTE: TO VIEW YOUR BOOKING DETAILS YOU MUST HAVE A ROOM BOOKING
To Make Your Buffet Booking Details, Press M
To Exit Menu, Press E
N
Enter a customer ID:
857423
Enter your Name:
Nari
Enter your Address:
Aderson 78452 Willpark
Enter your Age:
23
Enter your Contact Number:
8574965485
Enter your E-Mail ID:
nari@gmail.com
Data Entered Sucessfully!
```

- Creating a new Customer ID from scratch; where Customer ID already exists and the following output:

```
N
Enter a customer ID:
857423
Customer ID already exists. Pleasse try again.
Enter a customer ID:
478125
Enter your Name:
Hannah
Enter your Address:
Olive 78542 New Edison
Enter your Age:
25
Enter your Contact Number:
7480051267
Enter your E-Mail ID:
hannah123@gmail.com
Data Entered Sucessfully!
```

Looking for Customer Details:

- Looking for Customer Details and the following output:
  - When room and buffet booking is not done:

```
s
Enter a customer ID:
875412

------------------------------------------------------------------------

Customer Information for Customer ID: 875412

------------------------------------------------------------------------

| DETAILS          | CUSTOMER INFORMATION                        |
| ---------------- | ------------------------------------------- |
| Customer ID      | 875412                                      |
| Name             | Safi                                        |
| Address          | Ceacers 98574 Tailboard                     |
| Age              | 32                                          |
| Phone            | 458796325                                   |
| Email            | sforsafi@gmail.com                          |
| Room Booking     | You Don't have a Room Booking,              |
|                  | To make a Room Booking Press B              |
|                  | as mentioned in the main menu               |
| Buffet Booking   | You Don't have a Buffet Booking,            |
|                  | To make a Buffet Booking Press M            |
|                  | as mentioned in the main menu               |

------------------------------------------------------------------------
```

- When the room is done:

```
s
Enter a customer ID:
857423

------------------------------------------------------------------------

Customer Information for Customer ID: 857423

------------------------------------------------------------------------

| DETAILS          | CUSTOMER INFORMATION                        |
| ---------------- | ------------------------------------------- |
| Customer ID      | 857423                                      |
| Name             | Nari                                        |
| Address          | Aderson 78452 Willpark                      |
| Age              | 23                                          |
| Phone            | 8574965485                                  |
| Email            | nari@gmail.com                              |
| Room Booking     | You Have a Room Booking Done!               |
|                  | To view Press V                             |
|                  | as mentioned in the main menu               |
| Buffet Booking   | You Don't have a Buffet Booking,            |
|                  | To make a Buffet Booking Press M            |
|                  | as mentioned in the main menu               |

------------------------------------------------------------------------
```

- When the buffet is done:

```
s
Enter a customer ID:
478125

------------------------------------------------------------------------

Customer Information for Customer ID: 478125

------------------------------------------------------------------------

| DETAILS          | CUSTOMER INFORMATION                        |
| ---------------- | ------------------------------------------- |
| Customer ID      | 478125                                      |
| Name             | Hannah                                      |
| Address          | Olive 78542 New Edison                      |
| Age              | 25                                          |
| Phone            | 7480051267                                  |
| Email            | hannah123@gmail.com                         |
| Room Booking     | You Don't have a Room Booking,              |
|                  | To make a Room Booking Press B              |
|                  | as mentioned in the main menu               |
| Buffet Booking   | You Have a Buffet Booking Done!             |
|                  | To view Press V                             |
|                  | as mentioned in the main menu               |

------------------------------------------------------------------------
```

- When both are done:

```
S
Enter a customer ID:
101906

------------------------------------------------------------------------

Customer Information for Customer ID: 101906

------------------------------------------------------------------------

| DETAILS          | CUSTOMER INFORMATION                          |
| ---------------- | --------------------------------------------- |
| Customer ID      | 101906                                        |
| Name             | Shifa                                         |
| Address          | Ajman 53062 Pure Gold                         |
| Age              | 16                                            |
| Phone            | 5974320019                                    |
| Email            | its_shxyz@gmail.com                           |
| Room Booking     | You Have a Room Booking Done!                 |
|                  | To view Press V                               |
|                  | as mentioned in the main menu                 |
| Buffet Booking   | You Have a Buffet Booking Done!               |
|                  | To view Press V                               |
|                  | as mentioned in the main menu                 |

------------------------------------------------------------------------
```

Updating Customer Details:

- Updating Customer Details and the following output:

```
U
Enter a customer ID:
101010
Enter the new name you wish to update:
Jake
Enter the new address you wish to update:
Sam 78523 Oldroad
Enter the new age you wish to update:
25
Enter the new phone number you wish to update:
8574196325
Enter the e-mail ID you wish to update:
jake@gmail.com
```

- Table in MySQL before Updation:

Customer_id	Customer_Name	Customer_Address	Customer_Age	Customer_Phone	Customer_Mail
101010	Rae	Edison 45874 Manila	25	8741496850	raiee@gmail.com
101906	Shifa	Ajman 53062 Pure Gold	16	5974320019	its_shxyz@gmail.com
121504	Basco	Louisiana 42178 Wisconsin	18	7012152111	basco_BNC@gmail.com
199726	Lisa	Pranpriya 60324 Lalisa	26	3019972616	lisamanobal@gmail.com
201063	Mincheol	Manila 85476 Han River	32	7854966330	mincheoljin@gmail.com
214999	Sunmi	Heroine 54781 Gashina	31	1992220076	miyayeah@gmail.com
235222	Jiya	Shariah 50505 Bhutina	17	0503615523	hellooimposter@gmail.com

- Table in MySQL after Updation:

Customer_id	Customer_Name	Customer_Address	Customer_Age	Customer_Phone	Customer_Mail
101010	Jake	Sam 78523 Oldroad	25	8574196325	jake@gmail.com
101906	Shifa	Ajman 53062 Pure Gold	16	5974320019	its_shxyz@gmail.com
121504	Basco	Louisiana 42178 Wisconsin	18	7012152111	basco_BNC@gmail.com
199726	Lisa	Pranpriya 60324 Lalisa	26	3019972616	lisamanobal@gmail.com
201063	Mincheol	Manila 85476 Han River	32	7854966330	mincheoljin@gmail.com
214999	Sunmi	Heroine 54781 Gashina	31	1992220076	miyayeah@gmail.com
235222	Jiva	Shariah 50505 Bhutina	17	0503615523	helloogimposter@gmail.com

Delete Customer Account:

- Deleting Customer Account and the following output in Python:
  - When Customer doesn't want to delete:

```
D
Enter a customer ID:
101010
Are you sure you want to delete your information? Yes/No:
No
Cancelled.
```

  - When Customer wants to delete:

```
D
Enter a customer ID:
999999
Are you sure you want to delete your information? Yes/No:
Yes
Data successfully deleted. :)
```

Output generated in MySQL:

- Tables Customer_Details, Room, Buffet, and Bill Before Deletion:
  - Customer_Details:

Customer_id	Customer_Name	Customer_Address	Customer_Age	Customer_Phone	Customer_Mail
875412	Safi	Ceacers 98574 Tailboard	32	458796325	sforsafi@gmail.com
963258	Eli	Hostel 47852 Konopelski	19	4578120022	headofhosteleli@gmail.com
968745	Logan	Leejeans 85274 Willson	55	9856700331	loganlee@gmail.com
987500	Sally	Hostel 24301 Gangdong	22	7580014522	sallypark@gmail.com
999999	Danny	Cicada 40204 Hasegawa	26	87453256	daniel@gmail.com

  - Room:

Customer_id	Room_type	Type_full	Price_room	Price_service	Num_nigths	Total_price
782354	SL	Single bed Large	6600	1500	3	8100
855744	DL	Double bed Large	7500	1500	3	9000
856932	S	Single bed	2000	1000	2	3000
857423	SL	Single bed Large	4400	1000	2	5400
963258	DL	Double bed Large	12500	2500	5	15000
999999	DL	Double bed Large	7500	1500	3	9000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Buffet:

Customer_id	Meal_time	Num_Nights	Buffet_Price	Table_Price	Total_Price
782354	BLD	3	14400	3000	17400
855744	BLD	3	14400	3000	17400
856932	BL	2	5600	2000	7600
857423	BLD	2	9600	2000	11600
963258	BLD	5	24000	5000	29000
999999	BLD	3	14400	3000	17400
NULL	NULL	NULL	NULL	NULL	NULL

- Bills:

Customer_id	Room_Total	Buffet_Total	Total_Amount
987500	8800	0	10800
385241	2000	4600	3000
478563	1000	0	1500
547896	11000	0	13500
857423	4400	11600	5400
548963	4400	7600	5400
999999	7500	17400	9000

Output generated in MySQL:

- Tables Customer_Details, Room, Buffet, and Bill After Deletion:
  - Customer_Details:

Customer_id	Customer_Name	Customer_Address	Customer_Age	Customer_Phone	Customer_Mail
857423	Nari	Aderson 78452 Willpark	23	8574965485	nari@gmail.com
875412	Safi	Ceacers 98574 Tailboard	32	458796325	sforsafi@gmail.com
963258	Eli	Hostel 47852 Konopelski	19	4578120022	headofhosteleli@gmail.com
968745	Logan	Leejeans 85274 Willson	55	9856700331	loganlee@gmail.com
987500	Sally	Hostel 24301 Gangdong	22	7580014522	sallypark@gmail.com
NULL	NULL	NULL	NULL	NULL	NULL

  - Room:

Customer_id	Room_type	Type_full	Price_room	Price_service	Num_nigths	Total_price
741258	DL	Double bed Large	7500	1500	3	9000
782354	SL	Single bed Large	6600	1500	3	8100
855744	DL	Double bed Large	7500	1500	3	9000
856932	S	Single bed	2000	1000	2	3000
857423	SL	Single bed Large	4400	1000	2	5400
963258	DL	Double bed Large	12500	2500	5	15000
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- Buffet:

Customer_id	Meal_time	Num_Nights	Buffet_Price	Table_Price	Total_Price
741258	BLD	3	14400	3000	17400
782354	BLD	3	14400	3000	17400
855744	BLD	3	14400	3000	17400
856932	BL	2	5600	2000	7600
857423	BLD	2	9600	2000	11600
963258	BLD	5	24000	5000	29000
NULL	NULL	NULL	NULL	NULL	NULL

- Bills:

Customer_id	Room_Total	Buffet_Total	Total_Amount
201063	7500	17400	9000
987500	8800	0	10800
385241	2000	4600	3000
478563	1000	0	1500
547896	11000	0	13500
857423	4400	11600	5400
548963	4400	7600	5400

Booking room and viewing bill:

- Customer room booking; when the customer is satisfied with the total bill amount and the following output:

```
B
Enter a customer ID:
857423
Enter the room type. Type the character near the room:
S - Single bed, Price 1000/Night.
D - Double bed, Price 1300/Night.
SL - Single bed large, Price 2200/Night.
DL - Double bed large, Price 2500/Night.
DSL - Double bed + 1 single bed, Price 3000/Night.

NOTE: Room service costs Rs5000/night, regardless of room booked.
SL
Enter the number of nights for your stay at the BeachBay hotel:
2

------------------------------------------------------------------------

Booking Summary:

------------------------------------------------------------------------

| BOOKINGS                    | CUSTOMER'S BOOKING TYPE        |
| --------------------------- | ------------------------------ |
| Type of Room                |  Single bed Large              |
| Room Price                  | 4400                           |
| Room Service Price          | 1000                           |
| Total Bill Amount           | 5400                           |

------------------------------------------------------------------------

We hope you are satisfied with the bill, would you like to like to confirm your booking Y/N:
Y
Room Data Entered  Successfully! :)
Returing to main menu.....
```

- Customer room booking; when the customer is not satisfied with the total bill amount and the following output:

```
B
Enter a customer ID:
548963
Enter the room type. Type the character near the room:
S - Single bed, Price 1000/Night.
D - Double bed, Price 1300/Night.
SL - Single bed large, Price 2200/Night.
DL - Double bed large, Price 2500/Night.
DSL - Double bed + 1 single bed, Price 3000/Night.

NOTE: Room service costs Rs5000/night, regardless of room booked.
SL
Enter the number of nights for your stay at the BeachBay hotel:
2

------------------------------------------------------------------------

Booking Summary:

------------------------------------------------------------------------

| BOOKINGS                    | CUSTOMER'S BOOKING TYPE        |
| --------------------------- | ------------------------------ |
| Type of Room                |  Single bed Large              |
| Room Price                  | 4400                           |
| Room Service Price          | 1000                           |
| Total Bill Amount           | 5400                           |

------------------------------------------------------------------------

We hope you are satisfied with the bill, would you like to like to confirm your booking Y/N:
N
Booking cancelled
```

- Customer room booking; when the customer already has a room booking and the following output:

```
B
Enter a customer ID:
101906
You already have a room!
```

Viewing Booking Details:

- Viewing customer booking details; when the customer has both Room and Buffet booked and the following output:

```
v
Enter a customer ID:
101906

------------------------------------------------------------------------

Booking Data for Customer ID: 101906

------------------------------------------------------------------------

| Details                  | Customer Data             |
| ------------------------ | ------------------------- |
| Customer Name            | Shifa                     |
| Customer Phone           | 5974320019                |
| Customer Email           | its_shxyz@gmail.com       |
| Type of Room             | Single bed                |
| Room Price               | 4000                      |
| Room Service Price       | 2000                      |
| Total Room Bill          | 6000                      |
| Days of Buffet           | 4                         |
| Meal Time                | B                         |
| Buffet Price             | 5200                      |
| Table Price              | 4000                      |
| Total Buffet Bill        | 9200                      |
| Total Billing            | 15200                     |

------------------------------------------------------------------------



------------------------------------------------------------------------

| Total Room Bill          | 6000                      |
| Total Buffet Bill        | 9200                      |
| Overall Total            | 15200                     |

------------------------------------------------------------------------
```

- Viewing customer booking details; when the customer only has a room booking and the following output:
    - When the customer doesn't want to book a buffet:

```
V
Enter a customer ID:
857423
You don't have a Buffet booking!
Do you wish to make a Buffet booking? Y/N:N
| BOOKINGS                    | CUSTOMER'S BOOKING TYPE        |
| --------------------------- | ------------------------------ |
| Type of Room                |  Single bed Large              |
| Room Price                  | 4400                           |
| Room Service Price          | 1000                           |
| Total Bill Amount           | 5400                           |

------------------------------------------------------------
```

    - When the customer wants to book a buffet:

```
V
Enter a customer ID:
857423
You don't have a Buffet booking!
Do you wish to make a Buffet booking? Y/N:Y
Enter a customer ID:
857423
Enter the meal timing you want to place your buffet table:
B - Breakfast, Timings from 6:00 A.M. - 10:00 A.M., Price- 1300 /per day,
L - Lunch, Timings from 12:00 P.M. - 3:00 P.M., Price- 1500 /per day,
D - Dinner, Timings from 6:00 P.M. - 10:00 P.M.
Please Enter the Buffet timings you prefer, B - Breakfast, L - Lunch, D - Dinner:
Price- 2000 /per night

NOTE: Table costs Rs1000/table for booking.
BLD
Enter the number of Nights your stay in BeachBay Hotel:
2

------------------------------------------------------------------------

Buffet Booking Summary:

------------------------------------------------------------------------

| BUFFET DETAILS                | CUSTOMER BOOKINGS              |
| ----------------------------- | ------------------------------ |
| Time of Meal                  | Breakfast, Lunch and Dinner    |
| Buffet Price                  | 9600                           |
| Table Price                   | 2000                           |
| Total Buffet Bill             | 11600                          |

------------------------------------------------------------------------

Are you satisfied with the bill and would like to confirm your booking? Y/N:
Y
Buffet Data Entered Successfully! :)
```

- Viewing customer booking details; when the customer doesn't have a room booking and the following output:

```
V
Enter a customer ID:
478125
You don't have a room booking!
Going back to the main menu
----------------*WELCOME TO BeachBay HOTELS*----------------
----------------          *MAIN MENU*          ----------------
To Create a New Customer Account, Press N
To look for your details, Press S
To Update Existing Customer Account, Press U
To Delete Customer Account, Press D
To Book a Room, Press B
To View Your Booking Details, Press V
NOTE: TO VIEW YOUR BOOKING DETAILS YOU MUST HAVE A ROOM BOO
To Make Your Buffet Booking Details, Press M
To Exit Menu, Press E
```

Booking Buffet:

- Buffet booking; when the customer is satisfied with the total bill amount and the following output:

```
M
Enter a customer ID:
478125
Enter the meal timing you want to place your buffet table:
B - Breakfast, Timings from 6:00 A.M. - 10:00 A.M., Price- 1300 /per day,
L - Lunch, Timings from 12:00 P.M. - 3:00 P.M., Price- 1500 /per day,
D - Dinner, Timings from 6:00 P.M. - 10:00 P.M.
Please Enter the Buffet timings you prefer, B - Breakfast, L - Lunch, D - Dinner:
Price- 2000 /per night

NOTE: Table costs Rs1000/table for booking.
BLD
Enter the number of Nights your stay in BeachBay Hotel:
2

------------------------------------------------------------------------

Buffet Booking Summary:

------------------------------------------------------------------------

| BUFFET DETAILS                | CUSTOMER BOOKINGS                |
| ----------------------------- | -------------------------------- |
| Time of Meal                  | Breakfast, Lunch and Dinner      |
| Buffet Price                  | 9600                             |
| Table Price                   | 2000                             |
| Total Buffet Bill             | 11600                            |

------------------------------------------------------------------------

Are you satisfied with the bill and would like to confirm your booking? Y/N:
Y
Buffet Data Entered Successfully! :)
Returning to main menu.....
```

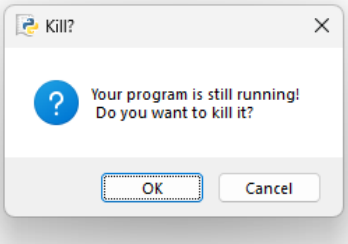
- Buffet booking; when the customer is not satisfied with the total bill amount and the following output:

```
M
Enter a customer ID:
548963
Enter the meal timing you want to place your buffet table:
B - Breakfast, Timings from 6:00 A.M. - 10:00 A.M., Price- 1300 /per day,
L - Lunch, Timings from 12:00 P.M. - 3:00 P.M., Price- 1500 /per day,
D - Dinner, Timings from 6:00 P.M. - 10:00 P.M.
Please Enter the Buffet timings you prefer, B - Breakfast, L - Lunch, D - Dinner:
Price- 2000 /per night

NOTE: Table costs Rs1000/table for booking.
BL
Enter the number of Nights your stay in BeachBay Hotel:
2

------------------------------------------------------------------------

Buffet Booking Summary:

------------------------------------------------------------------------

| BUFFET DETAILS                | CUSTOMER BOOKINGS               |
| ----------------------------- | ------------------------------- |
| Time of Meal                  | Breakfast and Lunch             |
| Buffet Price                  | 5600                            |
| Table Price                   | 2000                            |
| Total Buffet Bill             | 7600                            |

------------------------------------------------------------------------

Are you satisfied with the bill and would like to confirm your booking? Y/N:
N
Buffet Booking cancelled
```

Exiting code:

- Exiting the code and the following output:

```
---------------*WELCOME TO BeachBay HOTELS*---------------
---------------       *MAIN MENU*         ----------------
To Create a New Customer Account, Press N
To look for your details, Press S
To Update Existing Customer Account, Press U
To Delete Customer Account, Press D
To Book a Room, Press B
To View Your Booking Details, Press V
NOTE: TO VIEW YOUR BOOKING DETAILS YOU MUST HAVE A ROOM BOOKING
To Make Your Buffet Booking Details, Press M
To Exit Menu, Press E
E
Thank you for choosing  BeachBay Hotel, Visit us again!
```



Advantages and Disadvantages

Advantages:

- Quick and easy to access.
- Data is stored in a secure database.
- User friendly.
- Comprehensible and modifiable source code (Python source code)
- Data storage is as per the customer's wish.

Disadvantages:

- MySQL database is secure only if the central server is secure.
- Information is temporary.
- If the source code fails, no changes can be made to the output.

Future Enhancement:

- More features like Restaurant Bills.
- The code can be simple.
- You can create larger and more efficient databases using MySQL
- Python modules such as Tinkinter can be used to make the output more attractive.

Bibliography

- Sumita Arora Class 12 computer science textbook
- https://www.w3schools.com/sql
- https://www.plus2net.com/python
- https://www.geeksforgeeks.org/