ALL SAINT'S CHURCH SR.SEC. SCHOOL

M.I. ROAD , JAIPUR

A PROJECT REPORT ON

HOTEL MANAGEMENT SYSTEM

SUBJECT: INFORMATICS PRACTICES (065)

Session: 2023-2024

SUBMITTED BY

Inshaa Naz

SUBMITTED TO-

Mrs.Sharon Hiskiel

CERTIFICATE

This is to certify that Inshaa Naz of class XII SCIENCE has successfully completed the project on the topic Hotel Management System, In Partial fulfilment of the requirement for the AISSCE Partal Examination of the subject code Informatics Practices(065).

The project work reported here is as per the guidelines of CBSE for AISSCE Practical Examination and it s done under the supervision Mrs. Sharon Hiskiel, PGT COMPUTER. The project work, carried out by us is not a form of any other project work.

Internal Examiner **Principal**

External Examiner **School Seal**

ACKNOWLEDGMENT

We would like to express our special thanks to our teacher Mrs. Sharon Hiskiel for mentoring us throughout this project work. I also thank our respected principal Mrs. Shabnam Haque for her motivation and guidance throughout the year.

Our project is titled as "Hotel Management System" and it has enabled us to do a lot of research and We came to Know about so many new things in software design and development.

Also, We would also like to thank our parents who motivated and supported us during our work.

Inshaa Naz

XII IP

INDEX

PYTHON INTRODUCTION

Python is a general purpose, dynamic, high-level, and interpreted programming language. Python is a high level language. It is a free and open source language. It is an interpreted language, as python programs are executed by an interpreter. Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science. Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called cleaning the data.

MySQL INTRODUCTION

The “Hotel Management System” created by us is based on PYTHON AND MYSQL.

Its an automation of the existing system which enables its user to perform few operations pertaining to management of Hotel as listed below.

The Project Enables its user to:

1.) Add new Guest, new Staff and new Room records.

2.) Delete Guest, Staff and Room records.

3.) Update Guest, Staff and Room records.

4.) View Guest, Staff and Room records from the Database.

System Requirements

----------|Hardware Requirements |----------

Device name	DESKTOP-LEQ
Processor	Intel(R) Core(TM) i5-6000U CPU @ 2.40GHz 2.50 GHz
Installed RAM	8.00 GB (7.88 GB usable)
Device ID	A1FF6-D0-41EE-A3A4
Product ID	00330-50419-68709-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	Touch support with 10 touch points

INTRODUCTION OF PROJECT

PROJECT TITLE-(HOTEL MANAGEMENT)

DBMS: MySQL

Host : localhost

User: root
Password: root
Database: HOTEL

Table Structure: As per the Screenshot given below:

Screenshots OF PROJECT

Hotel table has following Schema

Guest table has following Schema

```
mysql> Desc Guests;
+--------+--------------+------+-----+---------+----------------+
| Field  | Type         | Null | Key | Default | Extra          |
+--------+--------------+------+-----+---------+----------------+
| Id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| name   | varchar(255) | YES  |     | NULL    |                |
| rno    | varchar(255) | YES  |     | NULL    |                |
| typ    | varchar(255) | YES  |     | NULL    |                |
| loc    | varchar(255) | YES  |     | NULL    |                |
| guest  | int(11)      | YES  |     | NULL    |                |
| rent   | int(11)      | YES  |     | NULL    |                |
| status | varchar(255) | YES  |     | NULL    |                |
+--------+--------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```

Staff table has following Schema

```
mysql> Desc staff;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| Id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| name     | varchar(255) | YES  |     | NULL    |                |
| position | varchar(255) | YES  |     | NULL    |                |
| salary   | int(11)      | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

Room table has following Schema

```
mysql> Desc roomservice;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| service_id     | int(11)      | NO   | PRI | NULL    | auto_increment |
| guest_id       | int(11)      | YES  |     | NULL    |                |
| service_name   | varchar(255) | YES  |     | NULL    |                |
| service_charge | int(11)      | YES  |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

USER OUTPUT

GUEST MODULE DETAILS:

```
Hotel Management System Menu:
1. Guest Operations
2. Staff Operations
3. Room Service Operations
4. Exit
Enter your choice (1-4): 1

Guest Operations:
1. Add Guest
2. View All Guests
3. Update Guest
4. Delete Guest
5. Back to Main Menu
Enter your choice (1-5):
```

STAFF MODULE DETAILS:

```
Hotel Management System Menu:
1. Guest Operations
2. Staff Operations
3. Room Service Operations
4. Exit
Enter your choice (1-4): 2

Staff Operations:
1. Add Staff
2. View All Staff
3. Update Staff
4. Delete Staff
5. Back to Main Menu
Enter your choice (1-5):
```

ROOM MODULE DETAILS:

```
Hotel Management System Menu:
1. Guest Operations
2. Staff Operations
3. Room Service Operations
4. Exit
Enter your choice (1-4): 3

Room Service Operations:
1. Add Room Service
2. View All Room Services
3. Update Room Service
4. Delete Room Service
5. Back to Main Menu
Enter your choice (1-5):
```

EXIT MODULE DETAILS:

```
Hotel Management System Menu:
1. Guest Operations
2. Staff Operations
3. Room Service Operations
4. Exit
Enter your choice (1-4): 4
Exiting the system. Goodbye!
```

CHARTS

Chart: GuestID vs Salary

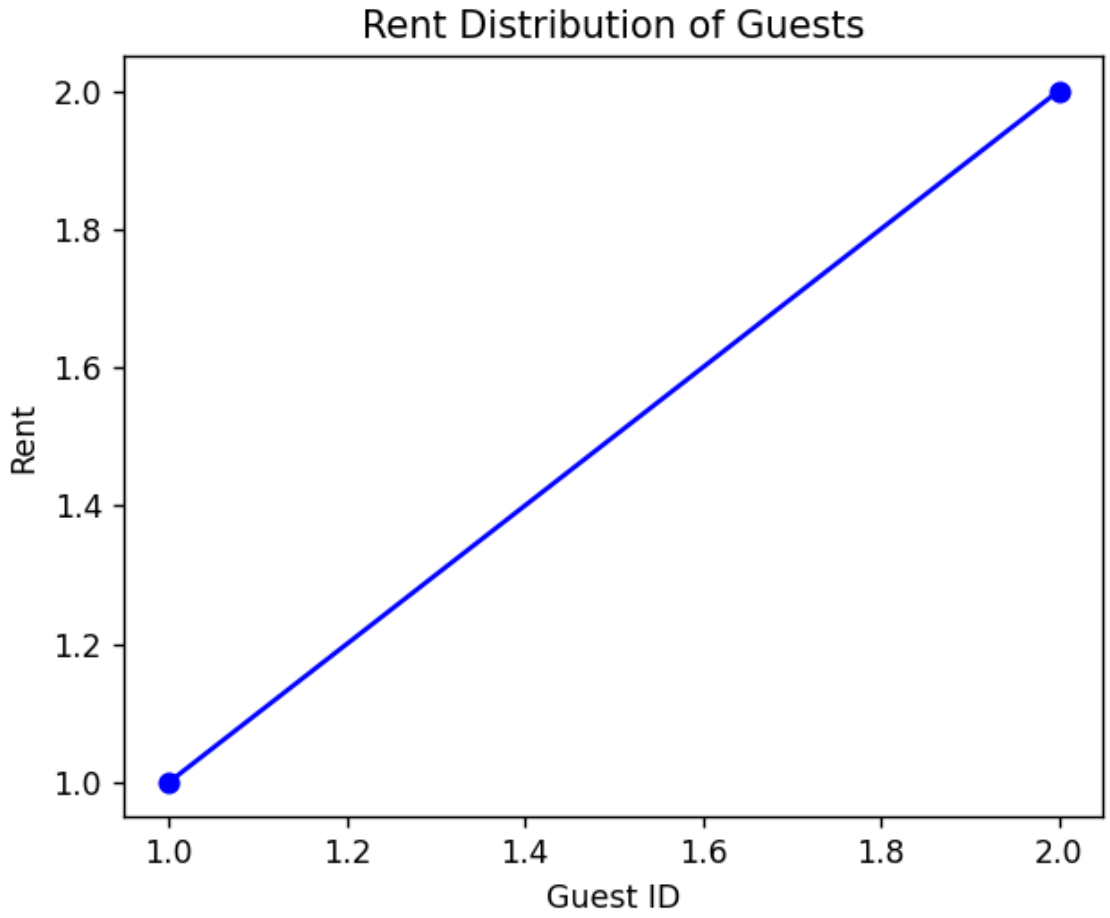


Chart: GuestName vs Rent

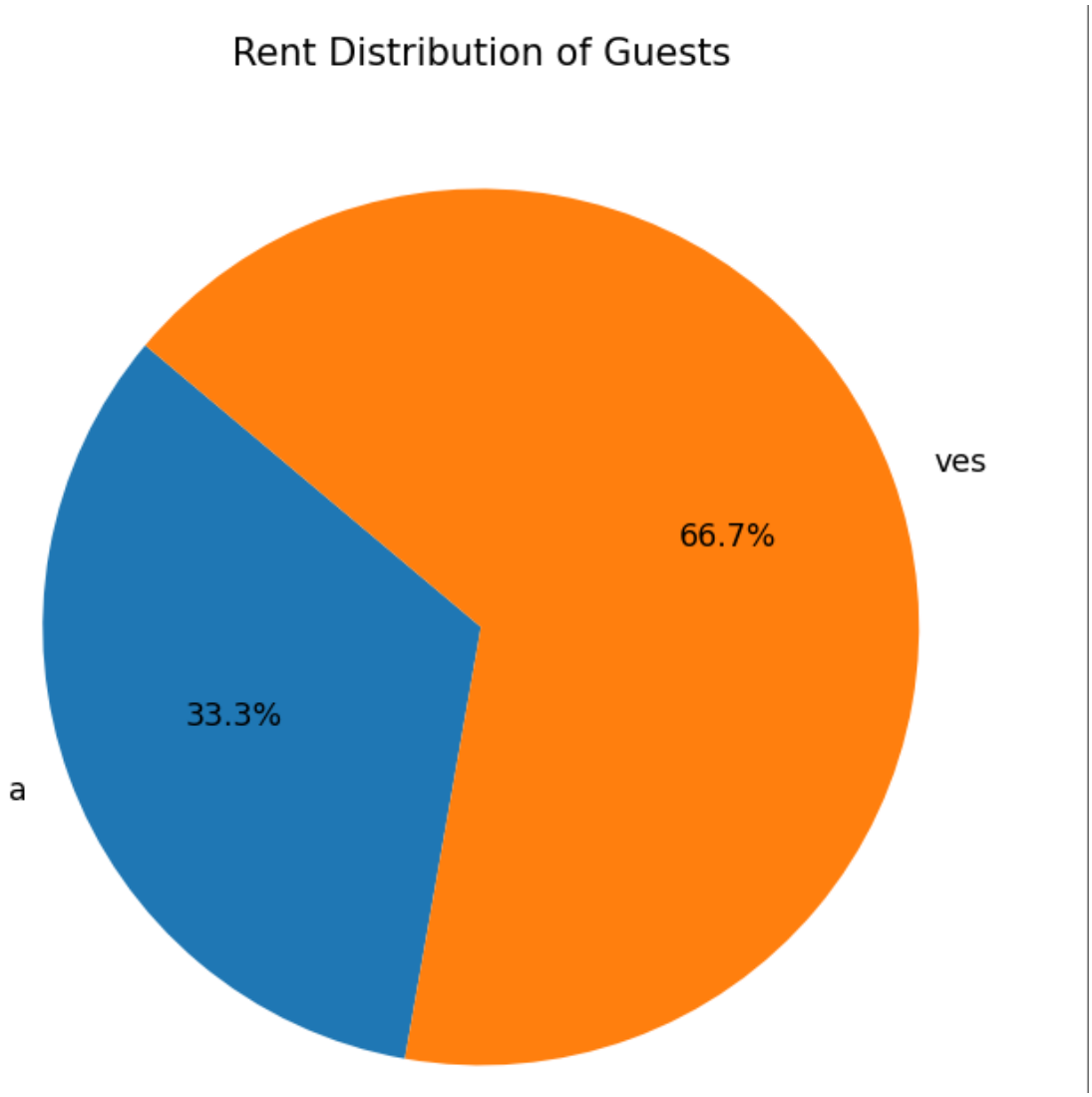


Chart: StaffName vs Salary

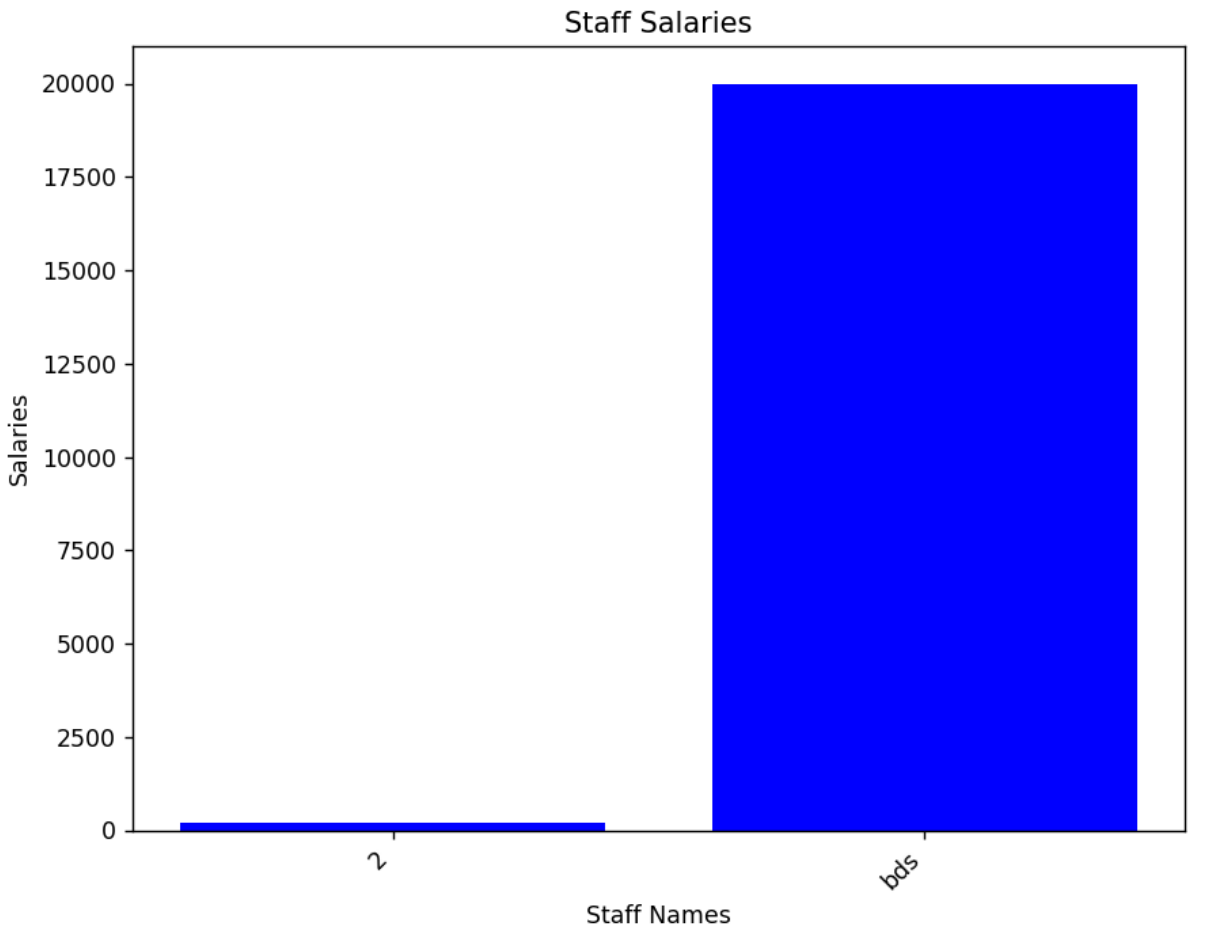


Chart: StaffName vs Staff salary

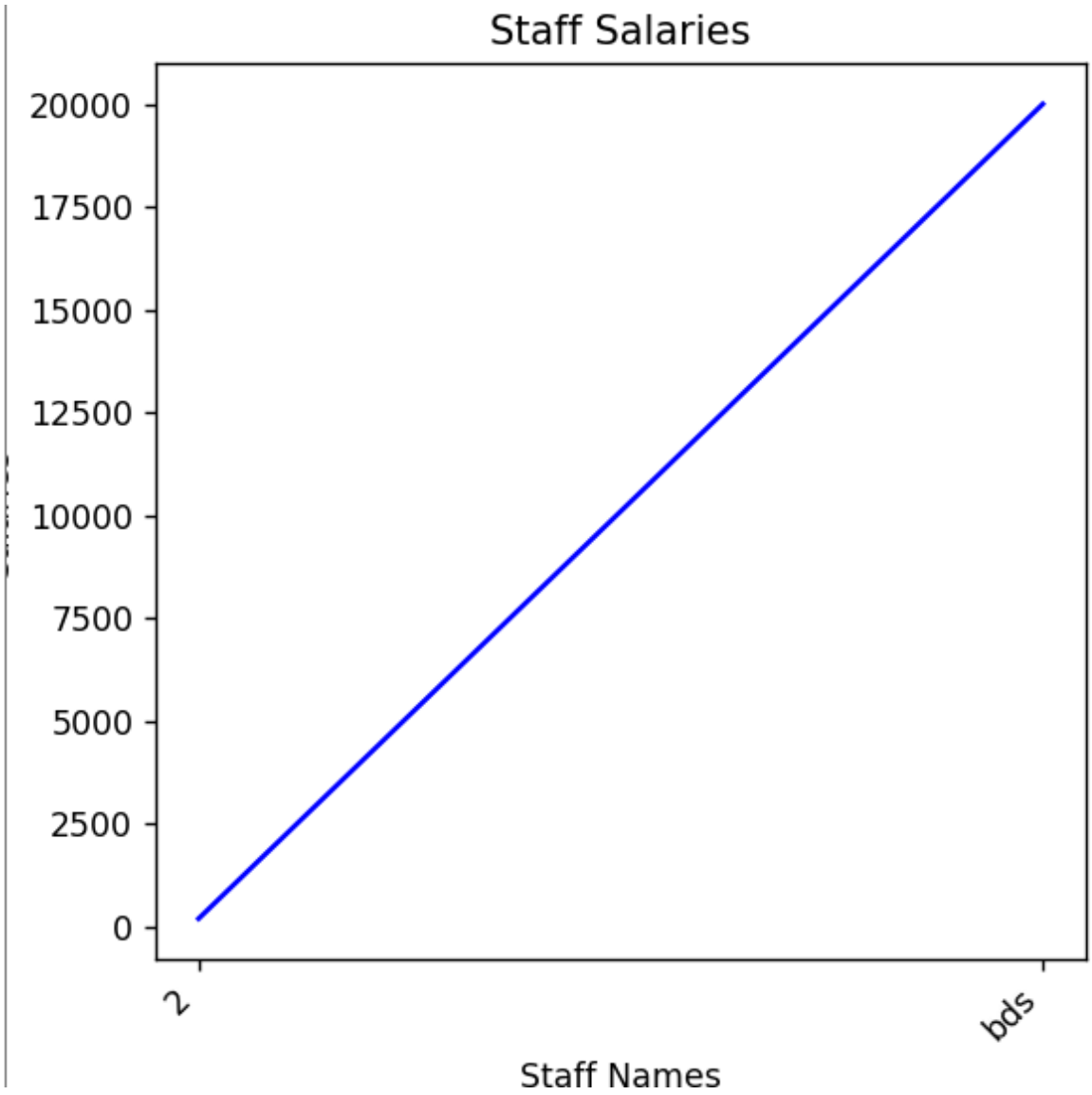


Chart: GuestName vs RoomCharges

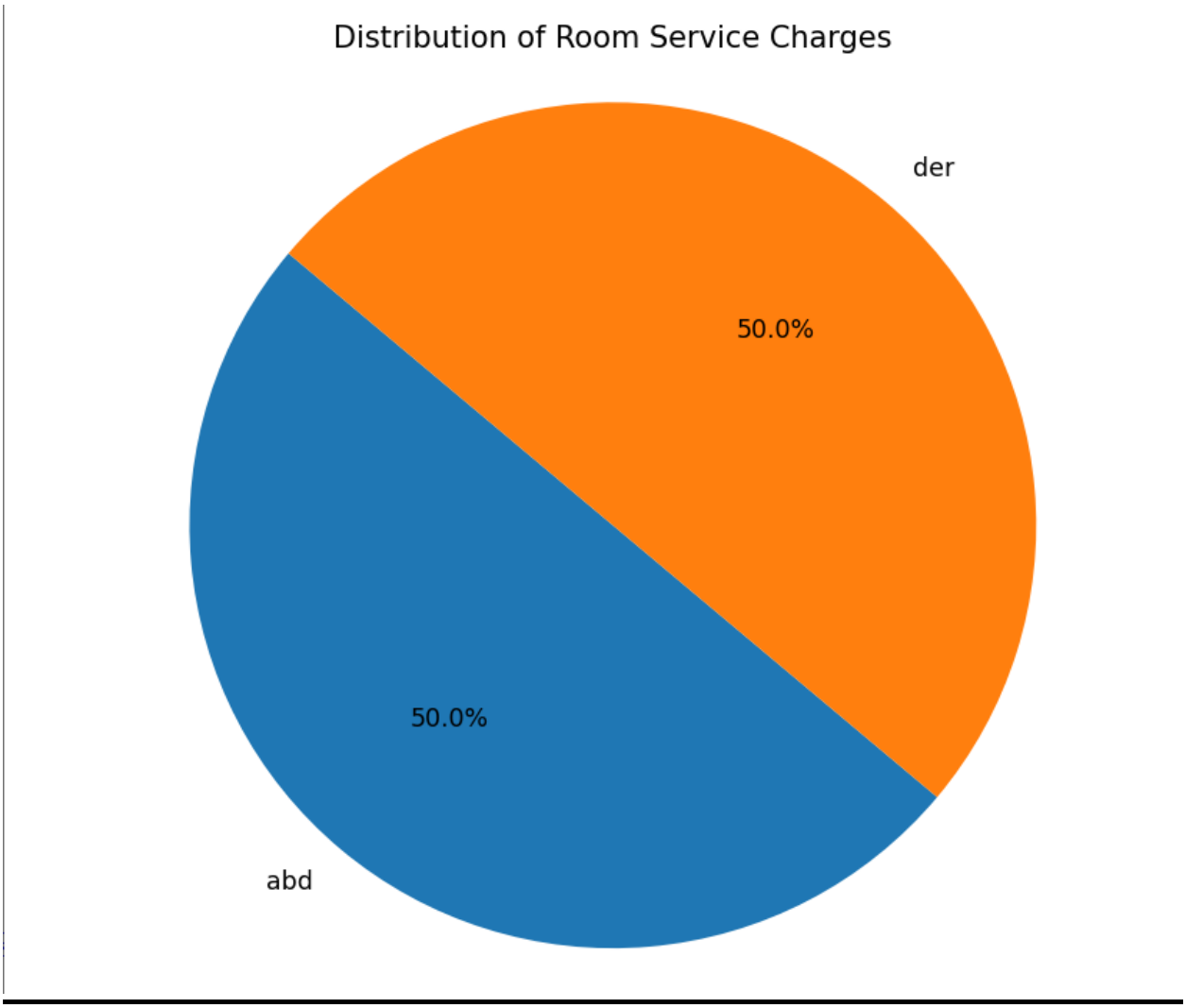
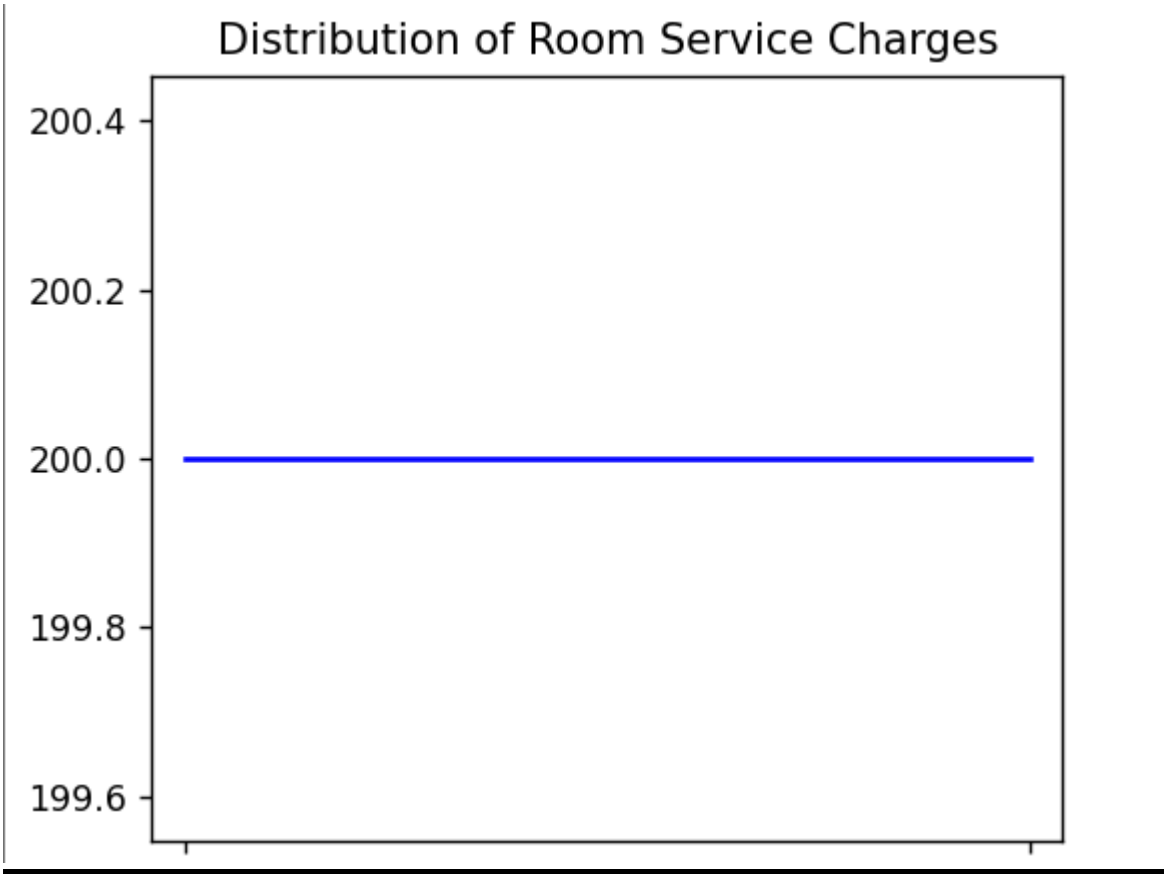


Chart: Guest vs Service Charges



USER OUTPUT(SOURCE CODE)

```
import mysql.connector
import matplotlib.pyplot as plt
# Connecting to the MySQL server
mydb = mysql.connector.connect(
    host="localhost",
    user='root',
    password='root')


# Creating a database
def create_database():
    cursor = mydb.cursor()
    cursor.execute('CREATE DATABASE IF NOT EXISTS Hotel')
    cursor.execute('USE Hotel')
create_database()
# Creating tables if not exists
def create_tables():
    cursor = mydb.cursor()
    # Guests table
    cursor.execute('CREATE TABLE IF NOT EXISTS Guests (Id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), rno VARCHAR(255), typ VARCHAR(255), loc VARCHAR(255), guest INT, rent INT, status VARCHAR(255))')
    # Staff table
    cursor.execute('CREATE TABLE IF NOT EXISTS Staff (Id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), position VARCHAR(255), salary INT)')
    # RoomService table
    cursor.execute('CREATE TABLE IF NOT EXISTS RoomService (service_id INT AUTO_INCREMENT PRIMARY KEY, guest_id INT, service_name VARCHAR(255), service_charge INT)')
    mydb.commit()
```

```
# Function to add a new guest
def add_guest():
    name = input("Enter Guest Name: ")
    rno = input("Enter Room No.: ")
    typ = input("Enter Room type: ")
    loc = input("Enter Location details: ")
    guest = int(input("Enter maximum number of guests: "))
    rent = int(input("Enter Per Day Charges: "))
    status = input("Vacant/Empty: ")
    cursor = mydb.cursor()
    sql = "INSERT INTO Guests (name, rno, typ, loc, guest, rent, status) VALUES (%s, %s, %s, %s, %s, %s, %s)"
    val = (name, rno, typ, loc, guest, rent, status)
    cursor.execute(sql, val)
    mydb.commit()
    print("Guest added successfully.")


# Function to view all guests
def view_guests():
    cursor = mydb.cursor()
    cursor.execute("SELECT * FROM Guests")
    result = cursor.fetchall()
    print("Press (r) to see the Record")
    print("Press (l) to see in the form of line graph")
    print("Press (p) to see in the form of pie graph")
    ch = input("Enter your choice: ")
    if ch=='r':
        for row in result:
            print(row)
    elif ch=='l':
```

```
        # Adding a line chart
        plt.figure(figsize=(8, 6))
        plt.plot([row[0] for row in result], [row[5] for row in result], marker='o', linestyle='-', color='b')
        plt.xlabel('Guest ID')
        plt.ylabel('Rent')
        plt.title('Rent Distribution of Guests')
        plt.show()
    elif ch=='p':
        # Adding a pie chart
        guest_names = [row[1] for row in result]
        rents = [row[5] for row in result]
        plt.figure(figsize=(8, 8))
        plt.pie(rents, labels=guest_names, autopct='%1.1f%%', startangle=140)
        plt.title('Rent Distribution of Guests')
        plt.axis('equal')
        plt.show()
# Function to update guest details
def update_guest():
    guest_id = int(input("Enter Guest ID to update: "))
    name = input("Enter Guest Name: ")
    rno = input("Enter Room No.: ")
    typ = input("Enter Room type: ")
    loc = input("Enter Location details: ")
    guest = int(input("Enter maximum number of guests: "))
    rent = int(input("Enter Per Day Charges: "))
    status = input("Vacant/Empty: ")
cursor = mydb.cursor()
```

```
    sql = "UPDATE Guests SET name = %s, rno = %s, typ = %s, loc = %s, guest = %s, rent = %s, status = %s
WHERE Id = %s"

    val = (name, rno, typ, loc, guest, rent, status, guest_id)

    cursor.execute(sql, val)

    mydb.commit()

    print("Guest details updated successfully.")


# Function to delete guest

def delete_guest():

    guest_id = int(input("Enter Guest ID to delete: "))

    cursor = mydb.cursor()

    sql = "DELETE FROM Guests WHERE Id = %s"

    val = (guest_id,)

    cursor.execute(sql, val)

    mydb.commit()

    print("Guest deleted successfully.")


# Function to add staff

def add_staff():

    name = input("Enter Staff Name: ")

    position = input("Enter Staff Position: ")

    salary = int(input("Enter Staff Salary: "))

    cursor = mydb.cursor()


    sql = "INSERT INTO Staff (name, position, salary) VALUES (%s, %s, %s)"

    val = (name, position, salary)

    cursor.execute(sql, val)

    mydb.commit()

    print("Staff added successfully.")
```

```
# Function to view all staff
def view_staff():
    cursor = mydb.cursor()
    cursor.execute("SELECT * FROM Staff")
    result = cursor.fetchall()
    print("Press (r) to see the Record")
    print("Press (b) to see in the form of bar graph")
    print("Press (l) to see in the form of line graph")
    ch = input("Enter your choice: ")
    if ch=='r':
        for row in result:
            print(row)
    elif ch=='b':
        # Adding a bar chart for staff salaries
        plt.figure(figsize=(8, 6))
        staff_names = [row[1] for row in result]
        staff_salaries = [row[3] for row in result]
        plt.bar(staff_names, staff_salaries, color='blue')

        plt.xlabel('Staff Names')
        plt.ylabel('Salaries')
        plt.title('Staff Salaries')
        plt.xticks(rotation=45, ha="right")
        plt.show()
    elif ch=='l':
        # Adding a line chart for staff salaries
        plt.figure(figsize=(8, 6))
        staff_names = [row[1] for row in result]
```

```
        staff_salaries = [row[3] for row in result]
        plt.plot(staff_names, staff_salaries, color='blue')
        plt.xlabel('Staff Names')
        plt.ylabel('Salaries')
        plt.title('Staff Salaries')
        plt.xticks(rotation=45, ha="right")
        plt.show()
# Function to update staff details
def update_staff():
    staff_id = int(input("Enter Staff ID to update: "))
    name = input("Enter Staff Name: ")
    position = input("Enter Staff Position: ")
    salary = int(input("Enter Staff Salary: "))
    cursor = mydb.cursor()
    sql = "UPDATE Staff SET name = %s, position = %s, salary = %s WHERE Id = %s"

    val = (name, position, salary, staff_id)
    cursor.execute(sql, val)
    mydb.commit()
    print("Staff details updated successfully.")

# Function to delete staff
def delete_staff():
    staff_id = int(input("Enter Staff ID to delete: "))
    cursor = mydb.cursor()
    sql = "DELETE FROM Staff WHERE Id = %s"
    val = (staff_id,)
    cursor.execute(sql, val)
```

```
    mydb.commit()
    print("Staff deleted successfully.")
# Function to add room service
def add_room_service():
    guest_id = int(input("Enter Guest ID: "))
    service_name = input("Enter Service Name: ")
    service_charge = int(input("Enter Service Charge: "))
    cursor = mydb.cursor()
    sql = "INSERT INTO RoomService (guest_id, service_name, service_charge) VALUES (%s, %s, %s)"
    val = (guest_id, service_name, service_charge)
    cursor.execute(sql, val)


    mydb.commit()
    print("Room service added successfully.")
# Function to view all room services
def view_room_service():
    cursor = mydb.cursor()
    cursor.execute("SELECT * FROM RoomService")
    result = cursor.fetchall()
    print("Press (r) to see the Record")
    print("Press (p) to see in the form of graph")
    print("Press (l) to see in the form of graph")
    ch = input("Enter your choice: ")
    if ch=='r':
        for row in result:
            print(row)
    elif ch=='l':
        # Adding pie chart
```

```
        service_names = [row[2] for row in result]
        service_charges = [row[3] for row in result]
        plt.figure(figsize=(8, 8))
        plt.plot(service_names, service_charges, color='blue')
        plt.title('Distribution of Room Service Charges')
        plt.axis('equal')
        plt.show()
    elif ch=='p':
        # Adding pie chart
        service_names = [row[2] for row in result]
        service_charges = [row[3] for row in result]
        plt.figure(figsize=(8, 8))
        plt.pie(service_charges, labels=service_names, autopct='%1.1f%%', startangle=140)
        plt.title('Distribution of Room Service Charges')
        plt.axis('equal')
        plt.show()
# Function to update room service details
def update_room_service():
    service_id = int(input("Enter Service ID to update: "))
    guest_id = int(input("Enter Guest ID: "))
    service_name = input("Enter Service Name: ")
    service_charge = int(input("Enter Service Charge: "))
    cursor = mydb.cursor()
    sql = "UPDATE RoomService SET guest_id = %s, service_name = %s, service_charge = %s WHERE service_id =
%s"
    val = (guest_id, service_name, service_charge, service_id)
    cursor.execute(sql, val)
    mydb.commit()
    print("Room service details updated successfully.")
```

```
# Function to delete room service
def delete_room_service():
    service_id = int(input("Enter Service ID to delete: "))
    cursor = mydb.cursor()
    sql = "DELETE FROM RoomService WHERE service_id = %s"
    val = (service_id,)
    cursor.execute(sql, val)
    mydb.commit()
    print("Room service deleted successfully.")
# Get the user's choice
def get_choice():
    while True:
        create_tables()
        print("\nHotel Management System Menu:")
        print("1. Guest Operations")
        print("2. Staff Operations")
        print("3. Room Service Operations")
        print("4. Exit")
        choice = input("Enter your choice (1-4): ")
        if choice == '1':
            guest_operations()
        elif choice == '2':
            staff_operations()
        elif choice == '3':
            room_service_operations()
        elif choice == '4':
            print("Exiting the system. Goodbye!")
```

```
            break
        else:
            print("Invalid choice. Please enter a valid option.")
# Guest operations menu
def guest_operations():
    while True:
        print("\nGuest Operations:")
        print("1. Add Guest")
        print("2. View All Guests")
        print("3. Update Guest")
        print("4. Delete Guest")
        print("5. Back to Main Menu")
        choice = input("Enter your choice (1-5): ")
        if choice == '1':
            add_guest()
        elif choice == '2':
            view_guests()
        elif choice == '3':
            update_guest()
        elif choice == '4':
            delete_guest()
        elif choice == '5':
            break
        else:
            print("Invalid choice. Please enter a valid option.")
# Staff operations menu
def staff_operations():
    while True:
```

```
        print("\nStaff Operations:")
        print("1. Add Staff")
        print("2. View All Staff")
        print("3. Update Staff")
        print("4. Delete Staff")
        print("5. Back to Main Menu")
        choice = input("Enter your choice (1-5): ")
        if choice == '1':
            add_staff()
        elif choice == '2':
            view_staff()
        elif choice == '3':
            update_staff()
        elif choice == '4':
            delete_staff()
        elif choice == '5':
            break
        else:
            print("Invalid choice. Please enter a valid option.")
# Room Service operations menu
def room_service_operations():
    while True:
        print("\nRoom Service Operations:")
        print("1. Add Room Service")
        print("2. View All Room Services")
        print("3. Update Room Service")
        print("4. Delete Room Service")
        print("5. Back to Main Menu")
```

```
        choice = input("Enter your choice (1-5): ")
        if choice == '1':
            add_room_service()
        elif choice == '2':
            view_room_service()
        elif choice == '3':
            update_room_service()
        elif choice == '4':
            delete_room_service()
        elif choice == '5':
            break
        else:s
            print("Invalid choice. Please enter a valid option.")
get_choice()
# Disconnecting from the MySQL server
mydb.close()
```

CONCLUSION

The Hotel Management System project is proposed to effectively understand the work of a hotel based organization and its workflow. It empowers users to manage hotel management system information very well.It mainly focused on maintaining information on Hotel and compensation .We have used Data Frame as they are effective in presenting the information in a concise manner.Data Visualization tool (Matplotlib) has been used for graphical representation of Data.So that completes our discussion for Hotel Management System Project Report with complete documentation (PDF).

Future Scope of Project

This project can be used in the hotel after adding some more useful modules in the project for which hotel are providing services . Almost care and back-up Procedures must be implementation established to ensure 100% successful implementation of the computerized hotel system. In case of system failure, the organization should be in a position to process the transaction with another organization or if the worst comes to the worst, it should be in a position to complete it manually. Scope of the improvement now a day's hotel is providing Many other facilities, this project can be improved with the improvement in the Hotels.

BIBLIOGRAPHY

- **GOOGLE**
- **www.wikipedia.com**
- **www.geeksforgeeks.org**
- **NCERT**
- **KIPS**
- **SUMITA ARORA**
- **PREETI ARORA**