DAYANANDA SAGAR UNIVERSITY

MINI PROJECT REPORT ON
"TOURIST INN RESERVATION SYSTEM"

SUBMMITED BY:

Pranav SS	**ENG20DS0030**
Gagana Malleshachari	**ENG20DS0016**
Rishank Gautam	**ENG20DS0035**

of

BACHELOR OF TECHNOLOGY

in

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

at

DAYANANDA SAGAR UNIVERSITY
SCHOOL OF ENGINEERING, BANGALORE-560068

3rd SEMESTER
(Course Code: 20CS2304)

DATABASE MANAGEMENT SYSTEM

DAYANANDA SAGAR UNIVERSITY

CERTIFICATE

This is to certify that the Mini project report entitled **"TOURIST INN RESERVATION SYSTEM"** being submitted by **PRANAV S S** to Department of Computer Science and Engineering, Dayananda Sagar University, Bangalore, for the 3rd Semester B.Tech CSE(DS) of this University during the academic year 2021-2022.

Date: 14-12-2021

Signature of the Faculty InCharge

Signature of the Chairman

DAYANANDA SAGAR UNIVERSITY

CERTIFICATE

This is to certify that the Mini project report entitled **"TOURIST INN RESERVATION SYSTEM"** being submitted by **GAGANA MALLESHACHARI** to Department of Computer Science and Engineering, Dayananda Sagar University, Bangalore, for the 3rd Semester B.Tech CSE(DS) of this University during the academic year 2021-2022.

Date: 14-12-2021

Signature of the Faculty InCharge

Signature of the Chairman

DAYANANDA SAGAR UNIVERSITY

CERTIFICATE

This is to certify that the Mini project report entitled **"TOURIST INN RESERVATION SYSTEM"** being submitted by **RISHANK GAUTAM** to Department of Computer Science and Engineering, Dayananda Sagar University, Bangalore, for the 3rd Semester B.Tech CSE(DS) of this University during the academic year 2021-2022.

Date: 14-12-2021

Signature of the Faculty InCharge

Signature of the Chairman

CONTENTS

INTRODUCTION

In this project an attempt is made to design a computer system for an **INN** that makes the management of recording user details, internet usage and billing much easier. The objective of this software is to maintain the details of users and their bookings. The Software powered by Streamlit and python assures clear and efficient services to the agency. This easy-to-operate system helps to access and modify user details, and provides an efficient billing facility. The software is designed to provide Reliable and error free information. The database is driven by My SQL thus providing portability.

This Database Design Project for **Tourist Inn Reservation System** is to reduce extensive paperwork in currently used system which is manual. It is a user-friendly. The owner and manager of the tourist inn can access this system to monitor all guest entering at the inn through the reports file. They can reserve a room, can book also and can add another day to stay in

PROBLEM STATEMENT

The purpose of the project is to automate Tourist Inn Reservation System. The software must include provisions to keep user details. It should be able search for a vacant room. Issue bills and check-in and check-out.

ABSTRACT

Computers have become a way of life for today's high society. Many aspects of modern life that we have come to accept as commonplace would not be possible if there were no computers. Today computers are used extensively in many areas of business, industry, science, education etc.

The major advantage of computers is its speed that makes it able to give some useful information very quickly. This speed also opens new approaches to problem solving and data processing. Another feature is its accuracy. Though the computers do only what is instructed at every instant, these instructions are taken into account and accurate information is produced. Computer can hold data and instruction in an electronic representation in internal memory and this data can be retrieved at any time

The project entitled "Inn Management System" is a software package, which can be used in Inn's for managing the visitor's details efficiently. It is indeed necessary to store the valid information of the customer who books a room in the inn. The system being used, will store data of check-in and check-out.

The main objective of the entire activity is to automate the process of day-to-day activities of hotel like: 1. Room activities, 2. Admission of a New Customer, 3. Assign a room according to customer's demand, 4. Checkout of a computer and releasing the room 5. Finally compute the bill etc... No data duplication No Paper Work Required. Time Efficient.

This system is developed in Python language with My SQL as backend. The system is a menu driven one. User-friendly menus will help both the administrator and the clients to work on it without any operational difficulty.

SYSTEM REQUIREMENTS

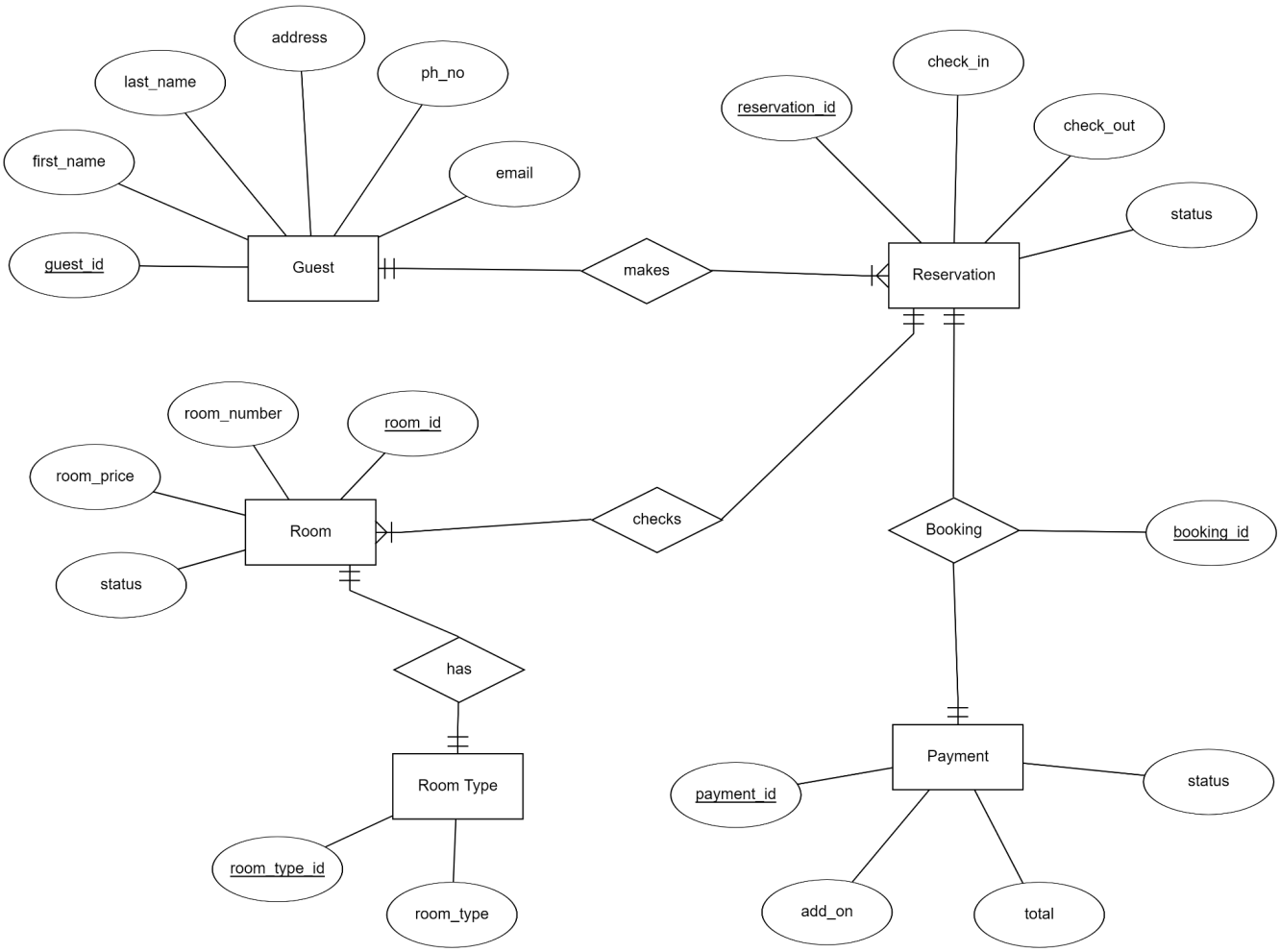
- **HARDWARE CONFIGURATION:**

Processor	-	Core i3 6th Gen
Memory	-	4GB RAM
Keyboard	-	105 Keys
Monitor	-	CRT or LCD
Mouse	-	USB mouse or PS/2 mouse

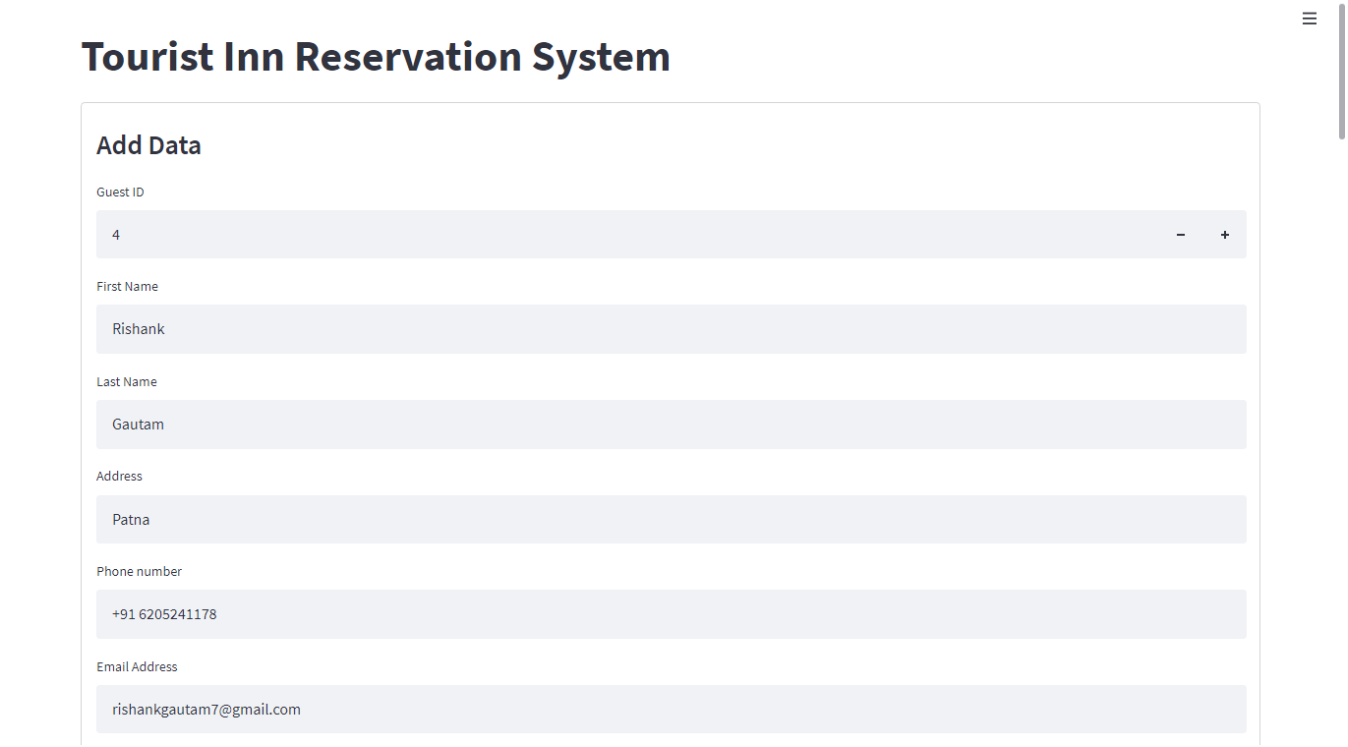
- **SOFTWARE SPECIFICATION:**

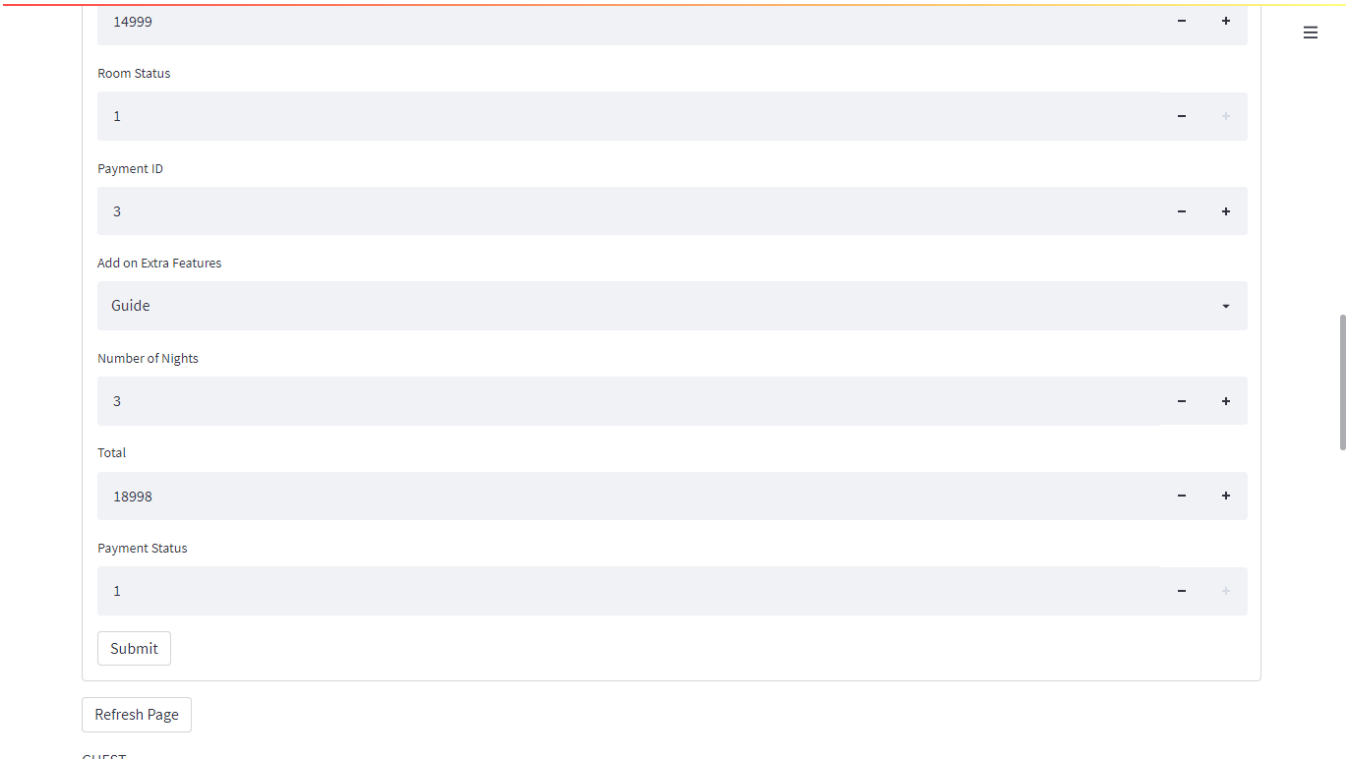
Operating System	:	Windows XP/8/10,
Front End	:	Streamlit
Back End	:	Python/MySQL
Documentation	:	MS Word

E-R DIAGRAM:


address
last_name
ph_no
first_name
email
guest_id
Guest
makes
reservation_id
check_in
check_out
status
Reservation
room_number
room_id
room_price
Room
checks
status
Booking
booking_id
has
Room Type
room_type_id
room_type
payment_id
Payment
status
add_on
total

SCREENSHORTS





Refresh Page

GUEST

	Guest ID	First Name	Last Name	Address	Phone number	Email
0	1	Pranav	S S	Bengaluru	+91 9986170312	pranavsindhanuru@gmail.com
1	2	Gagana	Malleshachari	Bengaluru	+91 7975680916	gaganamallesh2002@gmail.com
2	3	Rishank	Gautam	Patna	+91 6205241178	rishankgautam7@gmail.com

RESERVATION

	Reservation ID	Guest ID	Room Type ID	Check IN	Check OUT	Reservation Status
0	1	1	1	13/12/21	15/12/21	1
1	2	2	2	14/12/21	17/12/21	1
2	3	3	3	20/12/21	23/12/21	1

BOOKING

	Booking Id	Room Type ID	Guest_ID	Check IN	Check OUT
0	1	1	1	13/12/21	15/12/21
1	2	2	2	14/12/21	17/12/21
2	3	3	3	20/12/21	23/12/21

ROOM TYPE

	Room Type ID	Room Type
0	1	REGULAR
1	2	SMALL
2	3	LARGE

BOOKING

	Booking Id	Room Type ID	Guest_ID	Check IN	Check OUT
0	1	1	1	13/12/21	15/12/21
1	2	2	2	14/12/21	17/12/21
2	3	3	3	20/12/21	23/12/21

ROOM TYPE

	Room Type ID	Room Type
0	1	REGULAR
1	2	SMALL
2	3	LARGE

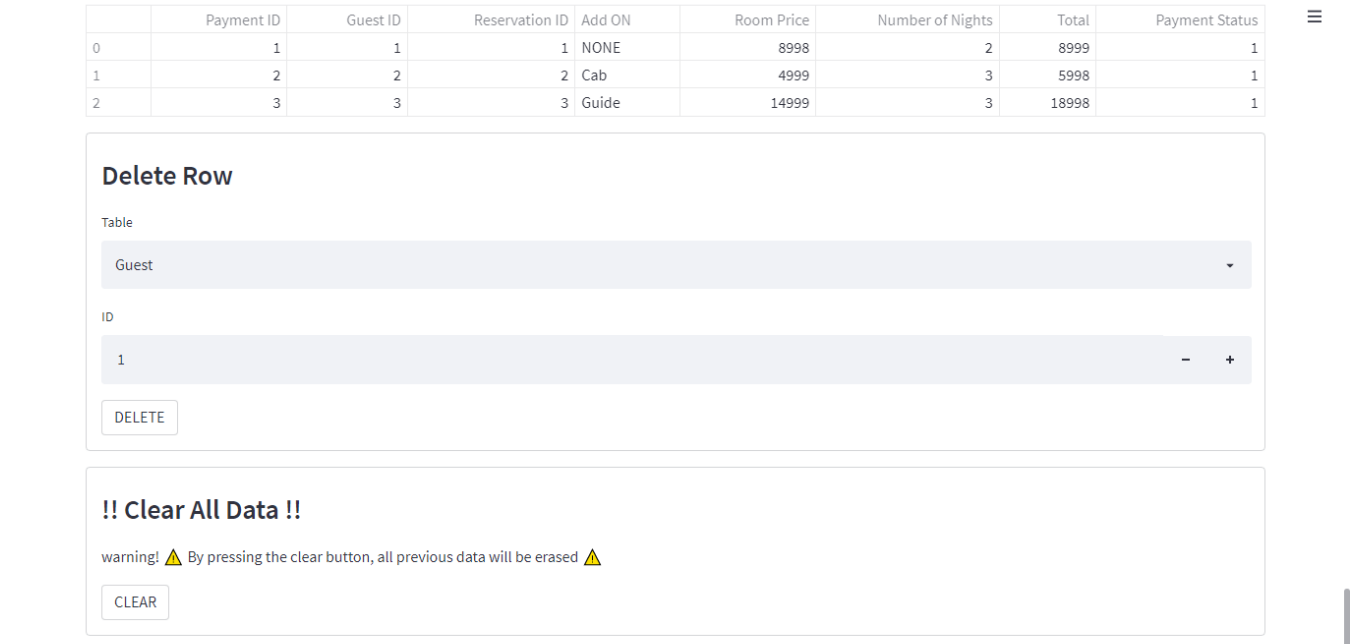
ROOM

	Room ID	Room Number	Room Type ID	Room Price	Room Status
0	1	10	1	8998	1
1	2	205	2	4999	1
2	3	6	3	14999	1

PAYMENTS

	Payment ID	Guest ID	Reservation ID	Add ON	Room Price	Number of Nights	Total	Payment Status
0	1	1	1	NONE	8998	2	8999	1
1	2	2	2	Cab	4999	3	5998	1
2	3	3	3	Guide	14999	3	18998	1

Delete Row



	Payment ID	Guest ID	Reservation ID	Add ON	Room Price	Number of Nights	Total	Payment Status
0	1	1	1	NONE	8998	2	8999	1
1	2	2	2	Cab	4999	3	5998	1
2	3	3	3	Guide	14999	3	18998	1

CODE

```
import mysql.connector
import streamlit as st
import pandas as pd
st.set_page_config(layout="wide")

db = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="root",
    database="dbms_mini_project"
    )

mycursor = db.cursor()

mycursor.execute("CREATE TABLE guest (Guest_id int, First_name varchar(50),
Last_name varchar(50), Address varchar(100), Phone_number varchar(30),
Email_address varchar(50))")
mycursor.execute("CREATE TABLE reservation (Reservation_id int, Guest_id int,
Room_type_id int, Check_in varchar(50), Check_out varchar(50),
Reservation_status int)")
```

```
mycursor.execute("CREATE TABLE booking (Booking_id int, Room_Type_id int,
Guest_id int, Check_in varchar(50), Check_out varchar(50))")
mycursor.execute("CREATE TABLE room_type (Room_type_id int, Room_type
varchar(50))")
mycursor.execute("CREATE TABLE room (Room_id int, Room_number int, Room_type_Id
int, Room_price int, Room_status int)")
mycursor.execute("CREATE TABLE payment (Payment_id int, Guest_id int,
Reservation_id int, Add_On varchar(50), Room_price int, Number_of_nights int,
Total int, Payment_status int)")

mycursor.execute("SELECT * FROM guest")
guest = pd.DataFrame(mycursor.fetchall(), columns=['Guest ID', 'First Name',
'Last Name', 'Address', 'Phone number', 'Email'])

mycursor.execute("SELECT * FROM reservation")
reservation = pd.DataFrame(mycursor.fetchall(), columns=['Reservation ID',
'Guest ID', 'Room Type ID', 'Check IN', 'Check OUT', 'Reservation Status'])

mycursor.execute("SELECT * FROM booking")
booking = pd.DataFrame(mycursor.fetchall(), columns=['Booking Id', 'Room Type
ID','Guest_ID', 'Check IN', 'Check OUT'])

mycursor.execute("SELECT * FROM room_type")
room_type = pd.DataFrame(mycursor.fetchall(), columns=['Room Type ID', 'Room
Type'])

mycursor.execute("SELECT * FROM room")
room = pd.DataFrame(mycursor.fetchall(), columns=['Room ID', 'Room Number',
'Room Type ID', 'Room Price', 'Room Status'])

mycursor.execute("SELECT * FROM payment")
payment = pd.DataFrame(mycursor.fetchall(), columns=['Payment ID', 'Guest ID',
'Reservation ID', 'Add ON', 'Room Price', 'Number of Nights', 'Total', 'Payment
Status'])

header = st.container()

with header:
    st.title('Tourist Inn Reservation System')
```

```
with st.form(key='datainput'):
    st.subheader('Add Data')
    guest_id = st.number_input('Guest ID', step=1, min_value=0)
    first_name = st.text_input('First Name')
    last_name = st.text_input('Last Name')
    address = st.text_input('Address')
    ph_no = st.text_input('Phone number')
    email = st.text_input('Email Address')
    reservation_id = st.number_input('Reservation ID', step=1, min_value=0)
    room_type_id = st.number_input('Room Type ID', step=1, min_value=0)
    check_in = st.text_input('Check In')
    check_out = st.text_input('Check Out')
    reservation_status = st.number_input('Reservation Status', step=1,
min_value=0, max_value=1)
    booking_id = st.number_input('Booking ID', step=1, min_value=0)
    room_id = st.number_input('Room ID', step=1, min_value=0)
    room_type_ = st.selectbox('Room Type', ['MINI', 'SMALL', 'REGULAR', 'BIG',
'LARGE'])
    room_number = st.number_input('Room Number', step=1, min_value=0)
    room_price = st.number_input('Room Price', step=1, min_value=0)
    room_status = st.number_input('Room Status', step=1, min_value=0,
max_value=1)
    payment_id = st.number_input('Payment ID', step=1, min_value=0)
    add_on = st.selectbox('Add on Extra Features', ['NONE', 'Pool', 'Spa',
'Guide', 'Cab'])
    number_of_nights = st.number_input('Number of Nights', step=1, min_value=0)
    total = st.number_input('Total', step=1, min_value=0)
    payment_status = st.number_input('Payment Status', step=1, min_value=0,
max_value=1)
    submit_button = st.form_submit_button()

if submit_button:
    with st.spinner('Running ....'):
        if guest_id != 0:
            mycursor.execute("INSERT INTO guest VALUES (%s, %s, %s, %s, %s,
%s)", (guest_id, first_name, last_name, address, ph_no, email))
        if reservation_id != 0:
            mycursor.execute("INSERT INTO reservation VALUES (%s, %s, %s, %s,
%s, %s);", (reservation_id, guest_id, room_type_id, check_in, check_out,
reservation_status))
```

```
        if booking_id != 0:
            mycursor.execute("INSERT INTO booking VALUES (%s, %s, %s, %s,
%s);", (booking_id, room_type_id, guest_id, check_in, check_out))
        if room_type_id != 0:
            mycursor.execute("INSERT INTO room_type VALUES (%s, %s);",
(room_type_id, room_type_))
        if room_id != 0:
            mycursor.execute("INSERT INTO room VALUES (%s, %s, %s, %s, %s);",
(room_id, room_number, room_type_id, room_price, room_status))
        if payment_id != 0:
            mycursor.execute("INSERT INTO payment VALUES (%s, %s, %s, %s, %s,
%s, %s, %s);", (payment_id, guest_id, reservation_id, add_on, room_price,
number_of_nights, total, payment_status))
        db.commit()
    st.success('Done!')

st.button('Refresh Page')

st.write('GUEST')
guest

st.write('RESERVATION')
reservation

st.write('BOOKING')
booking

st.write('ROOM TYPE')
room_type

st.write('ROOM')
room

st.write('PAYMENTS')
payment

with st.form('deletedata'):
    st.subheader('Delete Row')
    table = st.selectbox('Table', ['Guest', 'Reservation', 'Booking', 'Room
Type', 'Room', 'Payment'])
```

```
    str_x = table.lower().replace(' ', '_')
    str_x_id = str_x + '_id'
    id = st.number_input('ID', step=1, min_value=0)
    delete_button = st.form_submit_button('DELETE')

if delete_button:
    try:
        mycursor.execute(f"DELETE FROM {str_x} WHERE {str_x_id} = '{id}'")
        db.commit()
    except:
        print('Item doesn\'t exist')

with st.form(key='cleardata'):
    st.subheader('!! Clear All Data !!')
    st.write('warning! ⚠️ By pressing the clear button, all previous data will
be erased ⚠️')
    clear_button = st.form_submit_button('CLEAR')

if clear_button:
    mycursor.execute("TRUNCATE guest")
    mycursor.execute("TRUNCATE reservation")
    mycursor.execute("TRUNCATE booking")
    mycursor.execute("TRUNCATE room_type")
    mycursor.execute("TRUNCATE room")
    mycursor.execute("TRUNCATE payment")
```

REFERENCES

- https://itsourcecode.com/free-projects/database-design-projects/database-design-project-for-tourist-inn-reservation-system/