TABLE OF CONTENTS

PROJECT REPORT

INTRODUCTION:

The hotel booking system project is developed for smoothly running and managing hotels. First, we gathered information about the process from several hotels managed by both computerized and manual systems. After a detailed analysis of the data collected, the hotel management system project was developed. This project was developed on A MySQL server. The hotel booking system project will be helpful in reserving hotel rooms and reducing the workload of employees and customers searching for a hotel room for their appropriate choice. The under-designed students of 2nd year :

Roshan Avinash -211210055

Satya Prakash Mahour -211210060

Have completed a project on the Database of the Hotel Booking System, which cites all bookings of hotel rooms from different countries from your hand.

CASE STUDY:

Data Base stores the information of every BOOKING on our hotel chain website. Here we as Admin can view all the users who registered on our website, view all the bookings, and update and drop all the entries of the hotels and their certain rooms. The main motto to create this database RS HOTELS is to make it the customer easy to reserve rooms from various hotels with different features from different places to their destination. Here we ask the customer some basic questions to fill in the entries and connect them with us.

THE BASIC OBJECTIVES OF THE PROPOSED SYSTEM

1. To enable online booking via the internet.
2. To enable automated data entry methods.
3. Ensure efficient and reliable communication within the hotel.
4. Avoid data entry errors by the use of input masks.
5. Enable easy authorized modification of data.
6. Enforce security measures to avoid unauthorized access to guest records.
7. Enable fast and easy retrieval of guest records and data for fast reference activities.

Functional Requirements

- The system supports customers booking and able to modify them
- Customers can search based on hotel, apartment, inns (ex. Radisson, Singapore)
- When a customer search for hotels, apartment, and the search result must contain hotel or apartment information (Address, Ratings, and Price) and also its availability within choosing check in and check out date.
- Customers able to cancel their booking from their account.
- Staffs able to edit customers booking information (updating check in, check out, room preferences, bed preferences and also cancelling booking).
- Customers can book online and pay with credit or debit card.
- The system must send booking confirmation email after successful payment.
- Customers can write reviews about hotels and apartment and also rate them.
- Customers able to check their booking status from their individual account.
- Customers can send feedback or call the company for booking purposes.
- Customers can check for latest promotion or deal.

HARDWARE REQUIREMENTS

Assuming that a typical system offers hundreds of entries . The volume of the information to be handled is thus about millions of characters. Further the whole information has to be processed. All this suggest that the minimum hardware requirements should be :

- Operating System: Windows 7/8/8.1/10
- Memory (RAM): 1 GB of RAM required.
- Hard Disk Space: 200 MB of free space required.
- Processor: Intel Pentium 4 or later.
- Cache : 512KB

SOFTWARE REQUIREMENTS

In developing a project, selecting an appropriate DBMS Software and a platform is of primary importance. With many software options available, a developer has to consider the various features and functionalities and ease of handling the software; keeping an account of such things, we decided to use Bootstrap Studio for designing the front end.

The front end has been developed using HTML, CSS and JS. MySQL has been used as a back-end query language. PHP has been chosen as a scripting language. The server chosen is the localhost which would be hosting the website on the machine itself. The following software is required :

- Web Browser (Chrome/Edge/Firefox, etc.)
- Graphics Accelerator (Nvidia or ATI or GL Server.)
- XAMPP/WAMPP or any other web server which can host
- MySQL

CUSTOMER:

Here we add all the entries made by the customer such as username,email_id, password,ph_no, and other personal details for verification from the REGISTRATION table.

HOTEL:

WE ADD A LIST OF ALL HOTELS AND THEIR DATA IN THIS TABLE LIKE NO. OF ROOMS AVAILABLE AND HOTEL LOCATION.

ROOMS:

We enter all the rooms of all the hotels which a made partnerships with us along with their rating prices compared to other hotels and reviews given by other customers.

ROOM RATES:

When the room rates are high for the given hotels, we show the customers the standard prices and reduced special prices by using a reference code from our website of hotel chains.

ROOM_TYPES:

We display all the rooms of the room type filtered by the customer of their choice. Showing them the price range and availability of rooms in each hotel in a given period of time.

BOOKINGS:

Here I as an admin can view all the requests from the registered users in a need of hotel rooms of their choice from verified hotels.

PAYMENT:

After the user registers on our website as a customer and books a room for themselves then he can pay the total amount. We use this table to check and give the user a reminder if there is a due in the bill while reserving the room.

INVOICE:

After payment completion, we ensure to send a final bill to both the customer as well as the hotel to which the customer has reserved. This table helps us to store all the payment history of our customers.

BOOKING STATUS:

If the filter shows no rooms of given hotel then it displays n/a which means not available and if the room is verified by the admin through payment and booking then we send an invoice slip to the customer showing a signal of confirmation.

ER DIAGRAM:

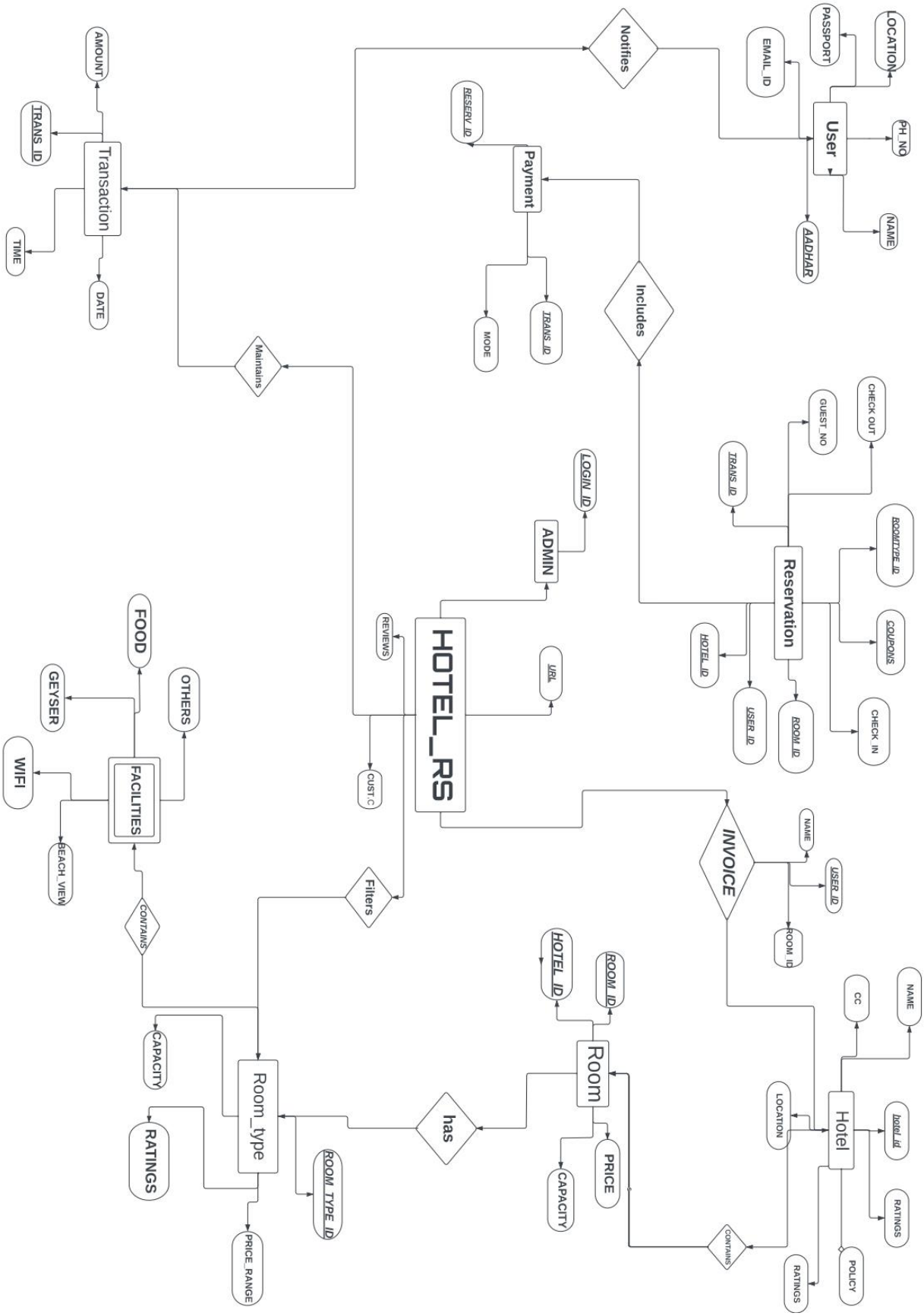


Figure 1 After checking loopholes, we made a final ER diagram

Final ER DIAGRAM:

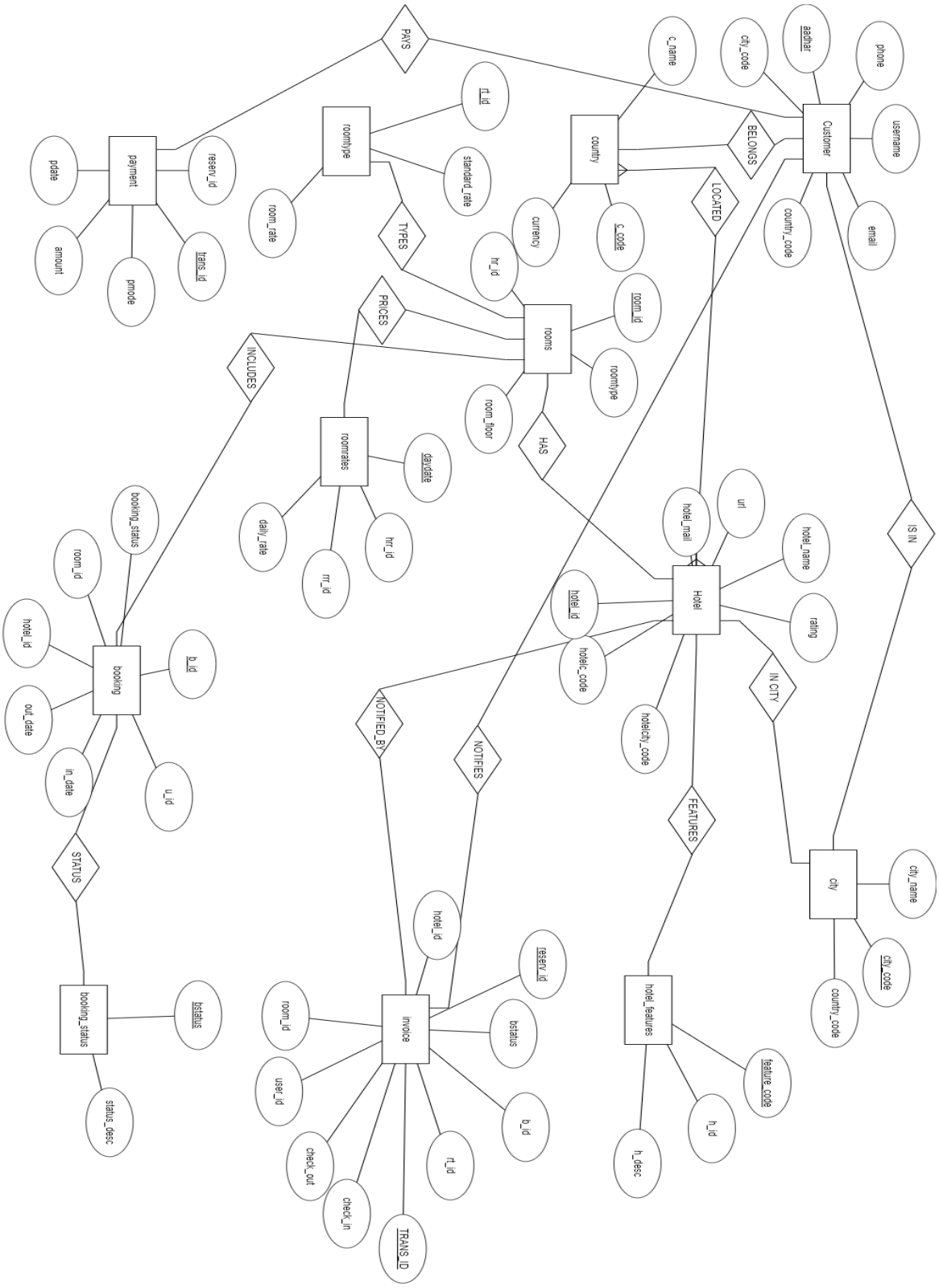


Figure 2 This is an ER diagram of our hotel booking system.

RELATIONAL MODEL:

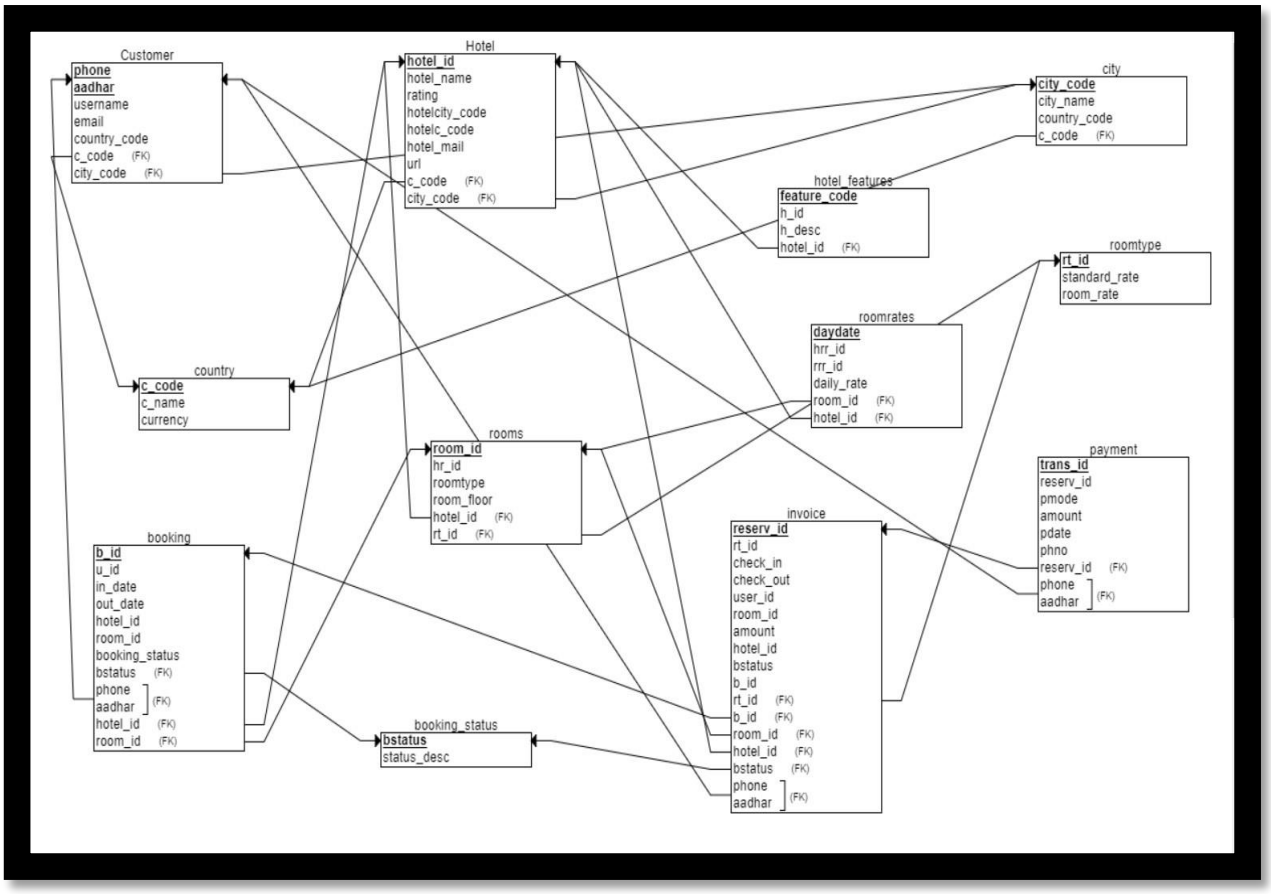


Figure 3 This is an ER->R Model of our Hotel Booking System.

CREATING TABLES

QUERIES TO CREATE THE RELATIONS AND POPULATE THE DATABASE:

```
CREATE TABLE CUSTOMER (
PHONE                                   INT         NOT NULL,
USERNAME                                VARCHAR(30)  NOT NULL,
EMAIL                                   VARCHAR(30)  NOT NULL,
COUNTRY_CODE                            VARCHAR(20)  NOT NULL,
AADHAR                                  INT      NOT NULL,
UPI                                     VARCHAR(20)  NOT NULL,
PRIMARY KEY(AADHAR,UPI));
```

```
CREATE TABLE HOTEL(
HOTEL_NAME       VARCHAR(30)          NOT NULL,
RATINGS              FLOAT
HOTELCITY_CODE   VARCHAR(20)         NOT NULL,
HOTELC_CODE    VARCHAR(20)           NOT NULL,
HOTEL_ID          VARCHAR(20)            NOT NULL,
HOTEL_MAIL     VARCHAR(30)           NOT NULL,
URL           VARCHAR (50)                NOT NULL,
PRIMARY KEY(HOTEL_ID));
```

```
CREATE TABLE HOTELFEATURES (
FEATURE_CODE VARCHAR(20) NOT NULL,
H_ID VARCHAR(20) NOT NULL,
HDESCP VARCHAR(100) NOT NULL,
PRIMARY KEY(FEATURE_CODE),
FOREIGN KEY (H_ID) REFERENCES HOTEL(HOTEL_ID));
```

```
CREATE TABLE CITY(
CITY_NAME  VARCHAR(30) NOT NULL,
CITY_CODE  INT     NOT NULL,
COUNTRY_CODE    INT      NOT NULL,
PRIMARY KEY (CITY_CODE),
FOREIGN KEY (COUNTRY_CODE) REFERENCES COUNTRY(C_CODE));
```

```
CREATE TABLE COUNTRY(
C_NAME  VARCHAR(30) NOT NULL,
C_CODE  VARCHAR(30)   NOT NULL,
CURRENCY   VARCHAR(30)   NOT NULL,
PRIMARY KEY (C_CODE));
```

```
CREATE TABLE ROOMS(
ROOM_ID  VARCHAR(10)  NOT NULL,
HR_ID                          VARCHAR(20)          NOT NULL,
ROOMTYPE VARCHAR(20)           NOT NULL,
ROOM_FLOOR                     INT,
PRIMARY KEY(ROOM_ID),
FOREIGN KEY (HR_ID) REFERENCES HOTEL(HOTEL_ID),
FOREIGN KEY(ROOMTYPE) REFERENCES ROOMTYPES(RT_ID));
```

```
CREATE TABLE ROOMRATES(
DAYDATE  DATE NOT NULL,
```

```
HRR_ID  VARCHAR(30) NOT NULL,
RRR_ID  VARCHAR(20) NOT NULL,
DAILYRATE  FLOAT  NOT NULL,
PRIMARY KEY (DAYDATE),
FOREIGN KEY (HRR_ID) REFERENCES HOTEL(HOTEL_ID),
FOREIGN KEY (RRR_ID) REFERENCES ROOMS(ROOM_ID))
```

```
CREATE TABLE ROOMTYPES(
RT_ID VARCHAR(20) NOT NULL,
STANDARDRATE VARCHAR(20)          NOT NULL,
ROOMRATE   VARCHAR(20)            NOT NULL,
PRIMARY KEY(RT_ID));
```

```
CREATE TABLE BOOKINGS
(B_ID                             VARCHAR(20)       NOT NULL,
U_ID                              INT NOT NULL,
INDATE   DATE                     NOT NULL,
OUTDATE                           DATE   NOT NULL,
HOTEL_ID                          VARCHAR(20)       NOT NULL,
ROOM_ID                           VARCHAR(20)       NOT NULL,
BOOKINGSTATUS  VARCHAR(20) NOT NULL,
PRIMARY KEY(B_ID),
FOREIGN KEY (U_ID) REFERENCES CUSTOMER(AADHAR),
FOREIGN KEY (ROOM_ID) REFERENCES ROOMS(ROOM_ID),
FOREIGN KEY (HOTEL_ID) REFERENCES HOTEL(HOTEL_ID),
FOREIGN KEY (BOOKINGSTATUS) REFERENCES BOOKINGSTATUS(BSTATUS))
```

```
CREATE TABLE BOOKINGSTATUS(
BSTATUS VARCHAR(20) NOT NULL,
STATUSDESCP  VARCHAR(200) NOT NULL,
PRIMARY KEY(BSTATUS));
```

```
---------------------------------------------------------------------------
CREATE TABLE INVOICE (
RESERV_ID  VARCHAR(30) NOT NULL,
RT_ID                                   VARCHAR(20)         NOT NULL,
CHECK_IN                                DATENOT NULL,
CHECK_OUT  DATE NOT NULL,
USER_ID                                 INT    NOT NULL,
ROOM_ID                                 VARCHAR(20)         NOT NULL,
AMOUNT                                      VARCHAR(20)  NOT NULL,
HOTEL_ID                                VARCHAR(20)          NOT NULL,
BSTATUS VARCHAR(30) NOT NULL,
BID VARCHAR(20) NOT NULL,
PRIMARY KEY(RESERV_ID),
FOREIGN KEY (USER_ID) REFERENCES CUSTOMER(AADHAR),
FOREIGN KEY (ROOM_ID) REFERENCES ROOMS(ROOM_ID),
FOREIGN KEY (BID) REFERENCES BOOKINGS(B_ID),
FOREIGN KEY (HOTEL_ID) REFERENCES HOTEL(HOTEL_ID),
FOREIGN KEY (BSTATUS) REFERENCES BOOKINGSTATUS(BSTATUS),
FOREIGN KEY (ROOM_ID) REFERENCES ROOMTYPES(RT_ID));

---------------------------------------------------------------------------
CREATE TABLE PAYMENT (
RESERV_ID VARCHAR(30) NOT NULL,
TRANS_ID  VARCHAR(30) NOT NULL,
PMODE    VARCHAR(10) NOT NULL,
AMOUNT   VARCHAR(10) NOT NULL,
PDATE DATE NOT NULL,
UPI VARCHAR(20) NOT NULL,

PRIMARY KEY(TRANS_ID),
FOREIGN KEY (UPI) REFERENCES CUSTOMER(UPI),
```

FOREIGN KEY(RESERV_ID) REFERENCES INVOICE(RESERV_ID)

);

SELECT*FROM PAYMENT;

INSERTION:

```
SET FOREIGN_KEY_CHECKS=0;

INSERT INTO rooms VALUES("01A","S","RS0001","13");
INSERT INTO rooms VALUES("01B","A","RS0002","14");
INSERT INTO rooms VALUES("02A","B","RS0003","15");
INSERT INTO rooms VALUES("02B","T","RS0004","16");
INSERT INTO rooms VALUES("03A","S","RS0005","17");
```

```
104 •   INSERT INTO city VALUES("DELHI","01","91");
105 •   INSERT INTO city VALUES("CALCUTTA","03","91");
106 •   INSERT INTO city VALUES("CHENNAI","02","91");
107 •   INSERT INTO city VALUES("QUWAIT","05","3");
108 •   INSERT INTO city VALUES("NEW JERSI","04","1");
109
110
111 •   SELECT *FROM city;
112
```

Result Grid | Filter Rows: | Edit: | Export/Import:

CITY_NAME	CITY_CODE	COUNTRY_CODE
DELHI	1	91
CHENNAI	2	91
CALCUTTA	3	91
NEW JERSI	4	1
QUWAIT	5	3
NULL	NULL	NULL

```
117 •   INSERT INTO hotelfeatures VALUES("RS01","AB1","WIFI");
118 •   INSERT INTO hotelfeatures VALUES("RS03","XY2","WIFI");
119 •   INSERT INTO hotelfeatures VALUES("RS04","RZ3","WIFI");
120 •   INSERT INTO hotelfeatures VALUES("RS05","BD4","WIFI");
121 •   INSERT INTO hotelfeatures VALUES("RS06","CE9","WIFI");
122
123 •   INSERT INTO hotelfeatures VALUES("RS02","AB1","GEYSER");
124 •   INSERT INTO hotelfeatures VALUES("RS07","XY2","GEYSER");
125 •   INSERT INTO hotelfeatures VALUES("RS08","RZ3","GEYSER");
126 •   INSERT INTO hotelfeatures VALUES("RS09","BD4","GEYSER");
127 •   INSERT INTO hotelfeatures VALUES("RS10","CE9","GEYSER");
128
129 •   SELECT *FROM hotelfeatures;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

FEATURE_CODE	H_ID	HDESCP
RS01	AB1	WIFI
RS02	AB1	GEYSER
RS03	XY2	WIFI
RS04	RZ3	WIFI
RS05	BD4	WIFI
RS06	CE9	WIFI
RS07	XY2	GEYSER
RS08	RZ3	GEYSER
RS09	BD4	GEYSER
RS10	CE9	GEYSER
NULL	NULL	NULL

```
153 •   INSERT INTO bookingstatus VALUES("CONFIRM","CONFIRMED");
154 •   INSERT INTO bookingstatus VALUES("N/A","NOT AVAILABLE");
155 •   INSERT INTO bookingstatus VALUES("AA","AVAILABLE");
156 •   INSERT INTO bookingstatus VALUES("FP","FEE PAY");
157
158 •   SELECT *FROM bookingstatus;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

BSTATUS	STATUSDESCP
AA	AVAILABLE
CONFIRM	CONFIRMED
FP	FEE PAY
N/A	NOT AVAILABLE
NULL	NULL

```
201 •  INSERT INTO invoice VALUES("R001","RT001","2022-12-02","2022-12-03","1234","01A","5000","AB1","CONFIRM","BS01");
202 •  INSERT INTO invoice VALUES("00000","0000","2022-12-00","2022-12-00","3215","00","0","AB0","N/A","BS25");
203
204 •  SELECT *FROM invoice;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RESERV_ID	RT_ID	CHECK_IN	CHECK_OUT	USER_ID	ROOM_ID	AMOUNT	HOTEL_ID	BSTATUS	BID
00000	0000	2022-12-00	2022-12-00	3215	00	0	AB0	N/A	BS25
R001	RT001	2022-12-02	2022-12-03	1234	01A	5000	AB1	CONFIRM	BS01
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
132 •  INSERT INTO hotel VALUES("TAJ HOTEL","5","01","91","AB1","TAJ@GMAIL.COM","WWW.TAJ.COM");
133 •  INSERT INTO hotel VALUES("RAJ HOTEL","4.5","01","91","RZ3","RAJ@GMAIL.COM","WWW.RAJ.COM");
134 •  INSERT INTO hotel VALUES("OBEROI HOTEL","4.2","01","91","CE9","OBEROI@GMAIL.COM","WWW.OBEROI.COM");
135 •  INSERT INTO hotel VALUES("OPERA HOTEL","3.7","04-","1","XY2","OPERA@GMAIL.COM","WWW.OPERA.COM");
136 •  INSERT INTO hotel VALUES("HAWELI HOTEL","4.8","05","3","BD4","HAWELI@GMAIL.COM","WWW.HAWELI.COM");
137
138 •  SELECT *FROM hotel;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

HOTEL_NAME	RATINGS	HOTELCITY_CODE	HOTELC_CODE	HOTEL_ID	HOTEL_MAIL	URL
TAJ HOTEL	5	01	91	AB1	TAJ@GMAIL.COM	WWW.TAJ.COM
HAWELI HOTEL	4.8	05	3	BD4	HAWELI@GMAIL.COM	WWW.HAWELI.COM
OBEROI HOTEL	4.2	01	91	CE9	OBEROI@GMAIL.COM	WWW.OBEROI.COM
RAJ HOTEL	4.5	01	91	RZ3	RAJ@GMAIL.COM	WWW.RAJ.COM
OPERA HOTEL	3.7	04-	1	XY2	OPERA@GMAIL.COM	WWW.OPERA.COM
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
162
163 •  INSERT INTO payment VALUES("R001","T001","DEBIT_C","5000","2022-12-01","UPI@9192929395");
164 •  INSERT INTO payment VALUES("R002","T005","CREDIT_C","10000","2022-12-25","UPI@6565656");
165 •  INSERT INTO payment VALUES("R003","T008","CASH","2000","2022-05-04","UPI@91991999515");
166 •  INSERT INTO payment VALUES("R004","T020","PHONPE","6000","2022-06-20","UPI@164646546");
167
168 •  SELECT *FROM payment;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RESERV_ID	TRANS_ID	PMODE	AMOUNT	PDATE	UPI
R001	T001	DEBIT_C	5000	2022-12-01	UPI@9192929395
R002	T005	CREDIT_C	10000	2022-12-25	UPI@6565656
R003	T008	CASH	2000	2022-05-04	UPI@91991999515
R004	T020	PHONPE	6000	2022-06-20	UPI@164646546
NULL	NULL	NULL	NULL	NULL	NULL

```
142 •  INSERT INTO roomtypes VALUES("RT001",1200,1000);
143 •  INSERT INTO roomtypes VALUES("RT002",5000,6000);
144 •  INSERT INTO roomtypes VALUES("RT003",8000,6000);
145 •  INSERT INTO roomtypes VALUES("RT004",10000,5000);
146 •  INSERT INTO roomtypes VALUES("RT005",5000,3000);
147
148 •  SELECT *FROM roomtypes;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

RT_ID	STANDARDRATE	ROOMRATE
RT001	1200	1000
RT002	5000	6000
RT003	8000	6000
RT004	10000	5000
RT005	5000	3000
NULL	NULL	NULL

roomtypes 7 ×

```
178
179 •  alter table customer modify phone varchar(10);
180
181
182 •  INSERT INTO customer VALUES("9874563555","U1234","U123@GMAIL.COM","91","784562589","UPI@9192929395");
183 •  INSERT INTO customer VALUES("7531596482","U3215","U3215@GMAIL.COM","1","689546321","UPI@6565656");
184 •  INSERT INTO customer VALUES("1592586458","U3210","U3210@GMAIL.COM","86","98954638","UPI@91991999515");
185 •  INSERT INTO customer VALUES("9535764852","U5988","U5988@GMAIL.COM","3","389546352","UPI@164646546");
186
187
188
189 •  SELECT *FROM customer;
190
191
192
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

phone	USERNAME	EMAIL	COUNTRY_CODE	AADHAR	UPI
1592586458	U3210	U3210@GMAIL.COM	86	98954638	UPI@91991999515
9535764852	U5988	U5988@GMAIL.COM	3	389546352	UPI@164646546
7531596482	U3215	U3215@GMAIL.COM	1	689546321	UPI@6565656
9874563555	U1234	U123@GMAIL.COM	91	784562589	UPI@9192929395
NULL	NULL	NULL	NULL	NULL	NULL

```
196
197 •  INSERT INTO bookings VALUES("B001","1234","2022-12-02","2022-12-03","AB1","01A","CONFIRM");
198 •  INSERT INTO bookings VALUES("B002","3215","2022-12-05","2022-12-06","XY2","01A","CONFIRM");
199 •  INSERT INTO bookings VALUES("B003","5988","2022-12-24","2022-12-25","AB1","02A","CONFIRM");
200 •  INSERT INTO bookings VALUES("B004","3210","2022-12-25","2022-12-00","AB1","000","N/A");
201
202 •  SELECT *FROM bookings;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

B_ID	U_ID	INDATE	OUTDATE	HOTEL_ID	ROOM_ID	BOOKINGSTATUS
B001	1234	2022-12-02	2022-12-03	AB1	01A	CONFIRM
B002	3215	2022-12-05	2022-12-06	XY2	01A	CONFIRM
B003	5988	2022-12-24	2022-12-25	AB1	02A	CONFIRM
B004	3210	2022-12-25	2022-12-00	AB1	000	N/A
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
204 •  INSERT INTO roomrates VALUES("2022-12-02","AB1","01A","5000");
205 •  INSERT INTO roomrates VALUES("2022-12-05","XY2","01A","3000");
206
207 •  SELECT *FROM roomrates;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

DAYDATE	HRR_ID	RRR_ID	DAILYRATE
2022-12-02	AB1	01A	5000
2022-12-05	XY2	01A	3000
NULL	NULL	NULL	NULL

SOME BASIC QUERIES.

FOR ADMIN{

DISPLAY AMOUNTS PAID ON DEC 25TH OF 2022 WITH TRANS_ID.

```
226 •  SELECT amount,trans_id FROM payment WHERE pdate="2022-12-25";
227
228
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

amount	trans_id
10000	T005

DISPLAY ALL THE HOTEL_IDS WHICH ARE CONFIRMED IN THE BOOKINGS.

```
220 •  SELECT hotel_id FROM bookings WHERE bookingstatus="CONFIRM";
221
222
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hotel_id
AB1
XY2
AB1

DISPLAY ALL THE USERS REGISTERED IN UR WEBSITE.

```
33 •    SELECT*FROM project.register;
34
```

Result Grid | Filter Rows: | Export:

name	email	password	contact
SATYA	SATYA@GMAIL.COM	NITDELHI	75315964
Roshan	roshan@GMAIL.COM	nitd1234	8569745215
Roshan	roshan@gmail.com	jhjh	8965741256

QUERIES OF CUSTOMER

{

WHAT IS THE COUNTRY_CODE OF DELHI CITY?

```
222 •    SELECT country_code FROM city WHERE city_name="DELHI";
223
224
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

country_code
91

SHOW THE CURRENCY OF ANY COUNTRY.

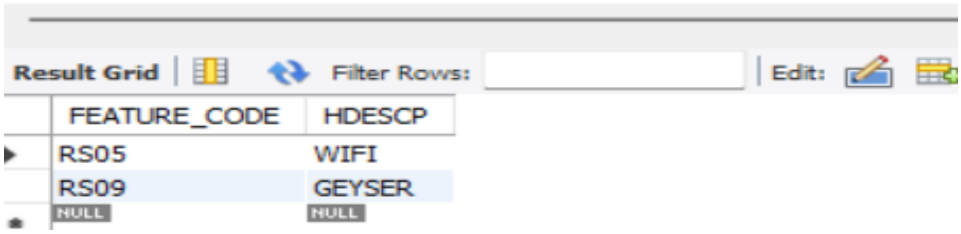
```
224 •    SELECT currency FROM country WHERE c_name="INDIA";
225
226
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

currency
RS

SHOW THE HOTEL FEATURES OF HOTEL TAJ WITH HOTEL ID='BD4'.

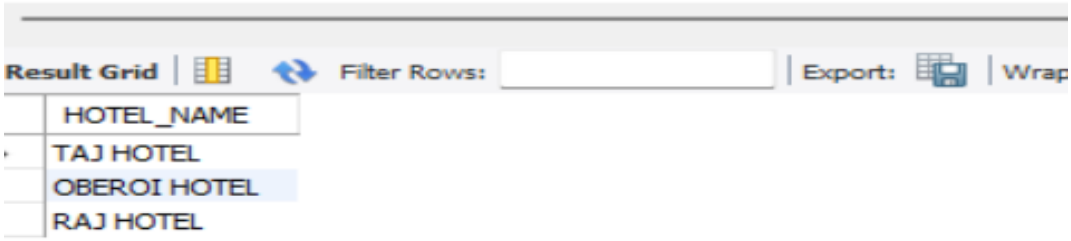
```
228
229 •    SELECT FEATURE_CODE,HDESCP
230      FROM project.hotelfeatures
231      WHERE H_ID='BD4';
232
```



FEATURE_CODE	HDESCP
RS05	WIFI
RS09	GEYSER
NULL	NULL

SHOW ALL THE HOTELS AVAILABLE IN INDIA.

```
228
229 •    SELECT HOTEL_NAME
230      FROM project.hotel
231      WHERE HOTELC_CODE='91';
232
```



HOTEL_NAME
TAJ HOTEL
OBEROI HOTEL
RAJ HOTEL

DISPLAY WHICH HOTEL PROVIDES ROOMS FOR >2000/-.

```
9  •   SELECT HOTEL_NAME,DAILYRATE
0      FROM project.roomrates,project.hotel
1      WHERE HRR_ID=HOTEL_ID AND DAILYRATE>2000;
2
```

sult Grid | Filter Rows: | Export: | Wrap Cell

HOTEL_NAME	DAILYRATE
TAJ HOTEL	5000
OPERA HOTEL	3000

HOW MANY HOTELS HAVE ROOMS OF TYPE RS0001?

```
229  •   SELECT COUNT(HOTEL_ID),ROOMTYPE
230      FROM  project.hotel,project.rooms
231      WHERE ROOMTYPE='RS0001';
```

Result Grid | Filter Rows: | Export: | Wra

COUNT(HOTEL_ID)	ROOMTYPE
5	RS0001

DB CONNECTIVITY.

Many people know from their own experience that it's not easy to install an Apache web server and it gets harder if you want to add MariaDB, PHP and Perl. The goal of XAMPP is to build an easy to install distribution for developers to get into the world of Apache. To make it convenient for developers. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

Prerequisites

Before going through XAMPP tutorial in-depth, you must have a fundamental knowledge of web development languages like HTML, and PHP.

FRONT END:

Figure 4 This is the home page of our website

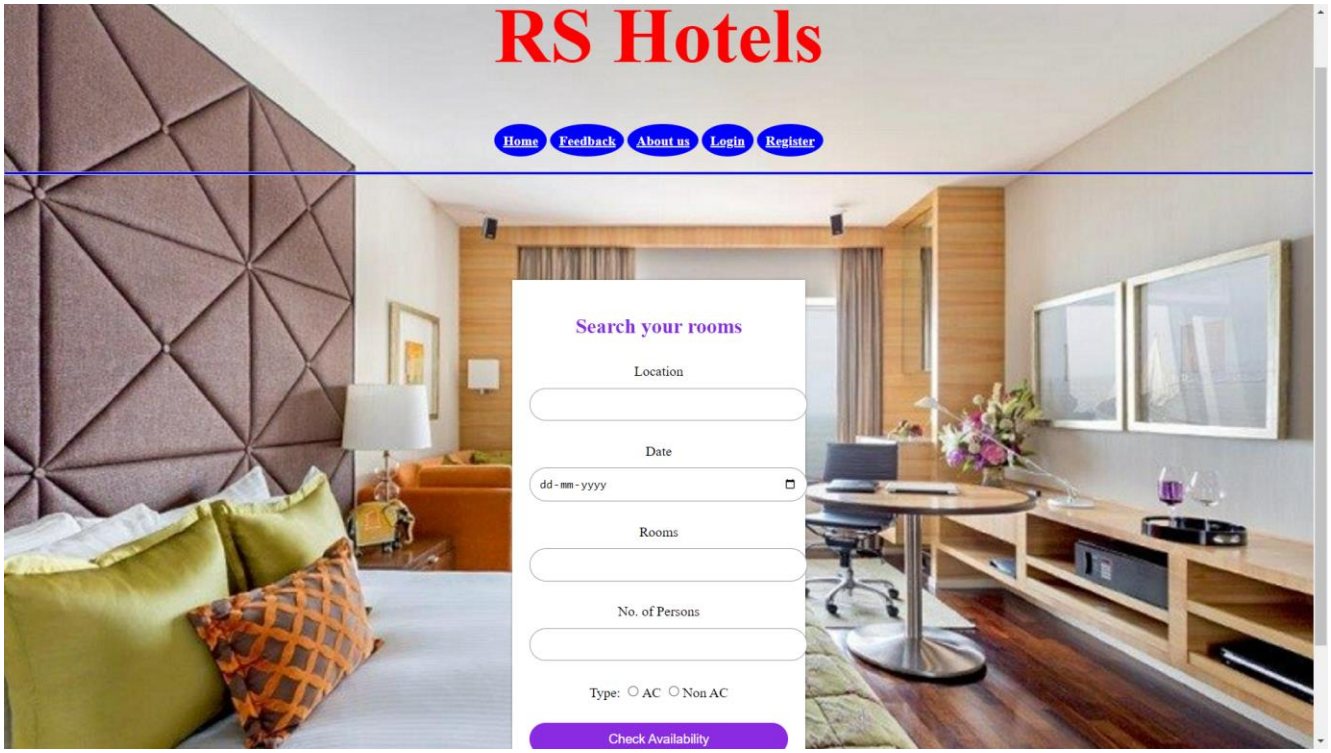


Figure 5 This is opened when the customer wants to book the rooms.

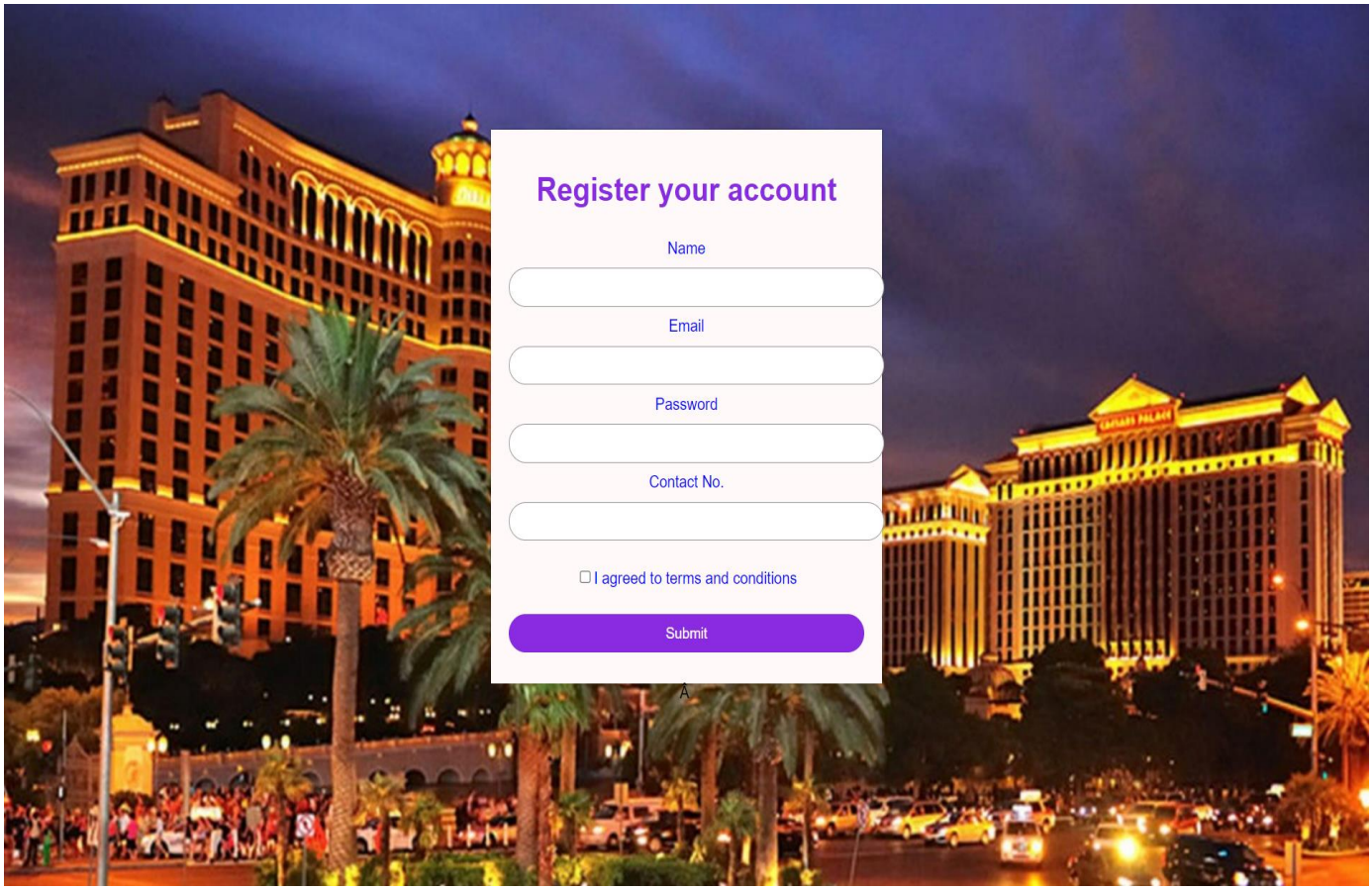


Figure 6 After that before entering payment we ask the customer to register their account on our website. So that on their next visit we can notify them whenever there is a room availability.

CONNECTING FRONT END AND BACK END USING PHP:

connection.php

C: > xampp > htdocs > project > connection.php

```php
<?php
    $name = $_POST['name'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $contact = $_POST['contact'];

    // Database connection
    $conn = new mysqli('localhost','root','201948','project');
    if($conn->connect_error){
        echo "$conn->connect_error";
        die("Connection Failed : ". $conn->connect_error);
    } else {
        $stmt = $conn->prepare("insert into register(name, email, password, contact) values(?, ?, ?, ?)");
        $stmt->bind_param("ssss",$name, $email, $password, $contact);
        $execval = $stmt->execute();
        echo $execval;
        echo "Registration successfully...";
        $stmt->close();
        $conn->close();
    }
?>
```

Figure 7 This PHP file is used to take the values entered by the customer when registering their account on our website.

THANK YOU