uOttawa

Faculté de génie
Faculty of Engineering

CSI 2132 Project Deliverable 2 Report
eHotel Booking System Database

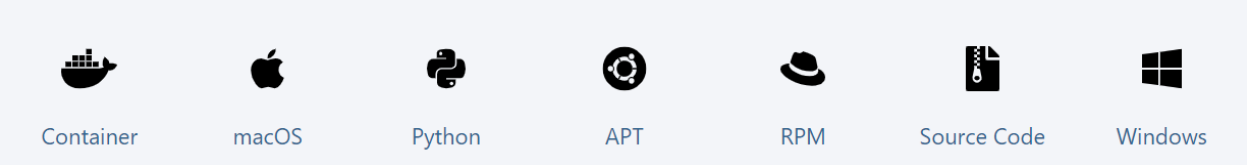
Group Member: Saffat Aziz (8708623)
Group Member: Nabil Ali (300067998)
Group Member: Anthony Polak (300119082)
Group Number: 14
Date: Sunday, April 3, 2021
Course Code: CSI 2132
Course Instructor: Paula Branco

1.0 Introduction:

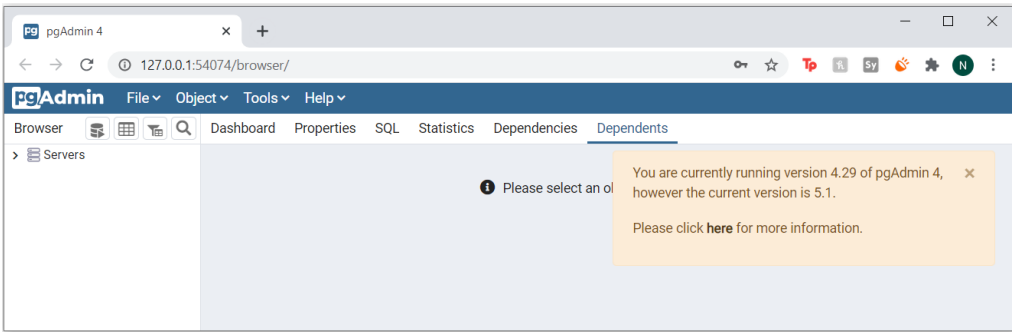
The premise of this project is to demonstrate the ability to create and manage the backend database of a large scale enterprise. The project assumes a collaboration between 6 hotel chains, each one with hotels in more than 20 locations. To address this collaboration we are asked to design and implement a scalable relational database system for all the hotel chains, their respective locations, employee records, customer entries and their room information. We do so by using the SQL developer pgAdmin 4 served by the University of Ottawa PostgresQL server.

2.0 DBMS Description

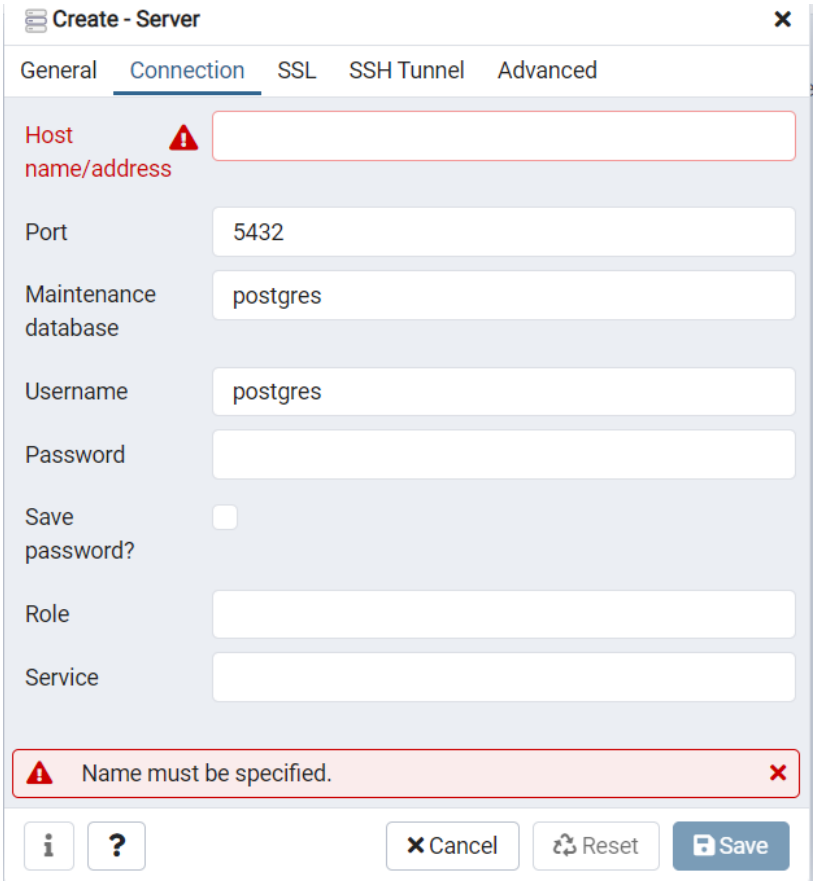
The database management system used to create the eHotel database is pgAdmin 4. pgAdmin 4 is a powerful PostgreSQL management tool, and is even open source so that anyone can use the platform free of cost and can be downloaded and used on Windows, MacOS, and other operating systems via https://www.pgadmin.org/download/.



Once you have downloaded the correct version for your operating system, complete the installation by following all prompts. You can now begin to set up your database! To open up pgAdmin, locate and open the pgAdmin application on your desktop. pgAdmin should now open up as a tab on your default web browser.



On the left is a column for your servers. Right click on ‘Servers’ → Click ‘Create’→ Click ‘Server...’, and now a ‘Create - Server’ window should open up. To get started with creating a server, set a Name in the General tab, and then click on the Connection tab.



Here, you can set the host name / address of your server, the port to use, the maintenance database, as well as the username and password for your database.

In our project, we used uOttawa’s web0.eecs.uottawa.ca as the host, 15432 as the port, and our uOttawa login and password to set up our server.

Once this step is complete, click the ‘Save’ button and your server is now set up!

3.0 DDLs for Database Creation

In terms of Data Description Languages, the only language used was the PostgreSQL database language. This SQL variation works perfectly within pgAdmin on the connected University of Ottawa PostgreSQL Database.

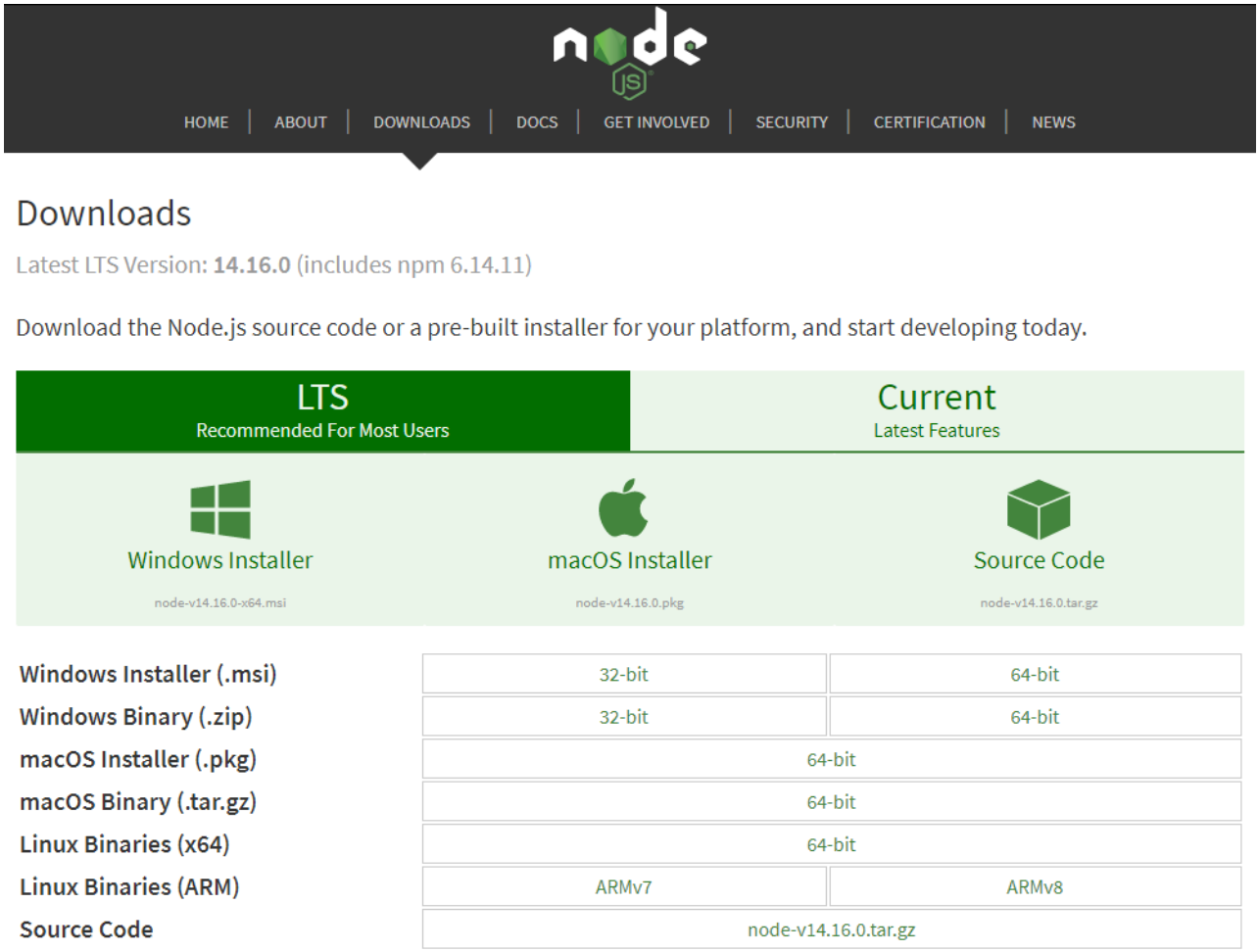
There are two files found within the Project Submission folder which allow for the successful database object creation and data insertion for the initial database state. The files are “eHotel.sql” and “insertValues.sql”. The 1st file is intended to create all of the tables, relations and objects within the database and the 2nd file is intended to populate the data into those tables.

4.0 Technologies and Programming Languages Used

The tech stack used to create this project includes technologies such as postgresQL, Express, React and Nodejs, also commonly known as the PERN stack. The Model View Controller (MVC) architectural design paradigm was followed to structure the project.

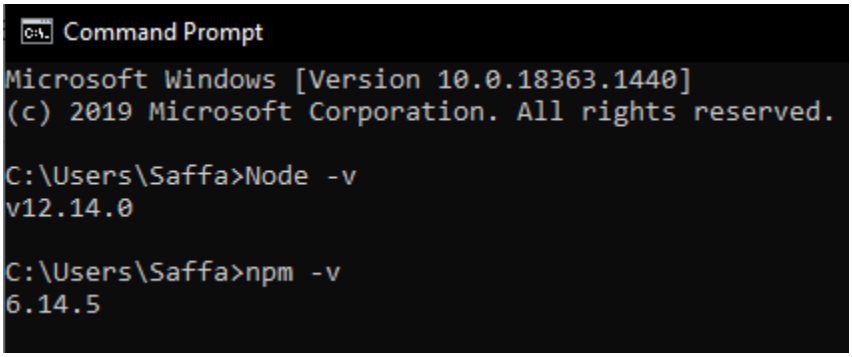
As mentioned above, postgresql (through pgadmin) was primarily used to create and manage the backend database. React was used to create the ‘View’ layer of the MVC architecture. Nodejs and Expressjs were used to create the server that communicates with the database to send the data to and from the database. This acts as the ‘Controller’ layer of the MVC architecture.

React is a component based javascript framework that uses HTML and CSS as the markup language and javascript as the underlying programming language. Nodejs is a javascript-based runtime environment and Expressjs is a backend web application framework for Nodejs. Express was used to build the server and REST API endpoints within the Node environment. The Server created using Nodejs and Express runs in localhost:3000 in parallel to the frontend React application, which runs on localhost:4545. Both these servers have to be running concurrently to run the application.
In order to successfully run the project in your local machine, there are a few dependencies that need to be installed. Please make sure that the most recent version of NodeJs and NPM is installed. This can be done by visiting the official Node website and selecting the appropriate installer for your computer.



After the installation, you can verify the inclusion of these dependencies by opening the cmd/terminal and typing the following commands. It should return the software version that has been installed. Once these two packages are installed, the project can then be installed and run in the local machine.

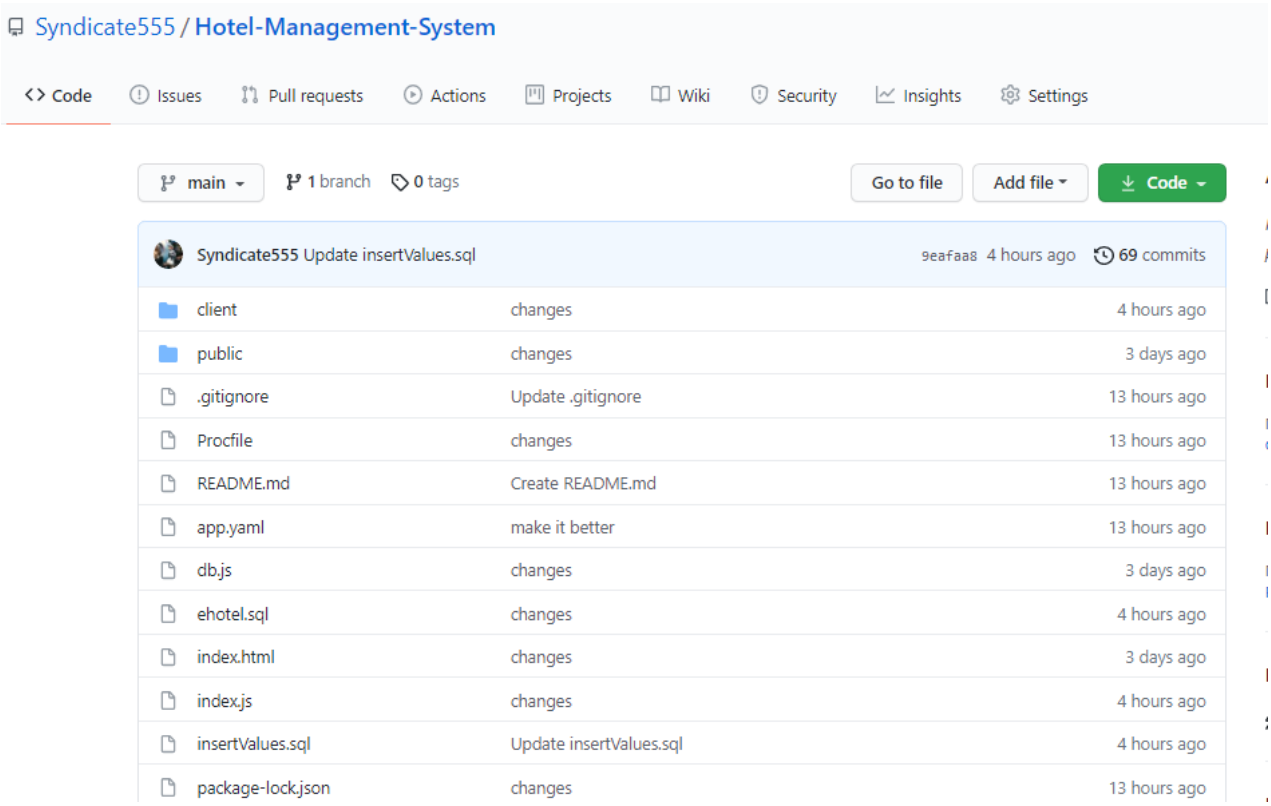
- Node -V
- Npm -V



5.0 Project Installation Guide

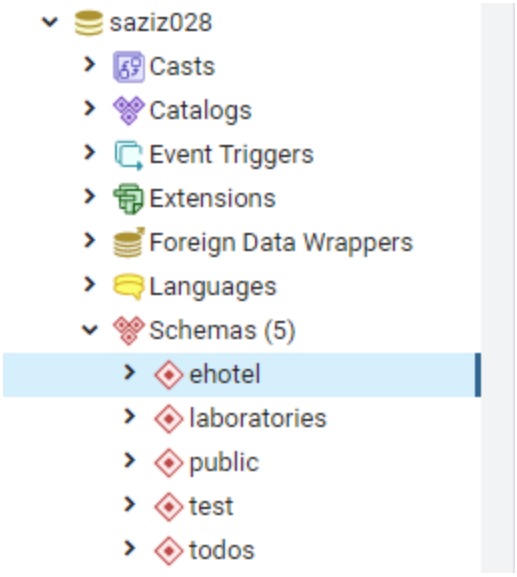
The FIles for this project can be accessed either by unzipping the project submission file or by cloning the following github repository.

Github repository - https://github.com/Syndicate555/Hotel-Management-System



In order to begin the installation process, please make sure that you have access to the University of Ottawa PostgresQL Server using pgadmin as described earlier in section 2. Within the project submission folder there will be 2 files with .sql extension. eHotel.sql contains all the sql commands to create the tables for the project and the insertValues.sql file contains all the insert commands to populate the database accordingly.

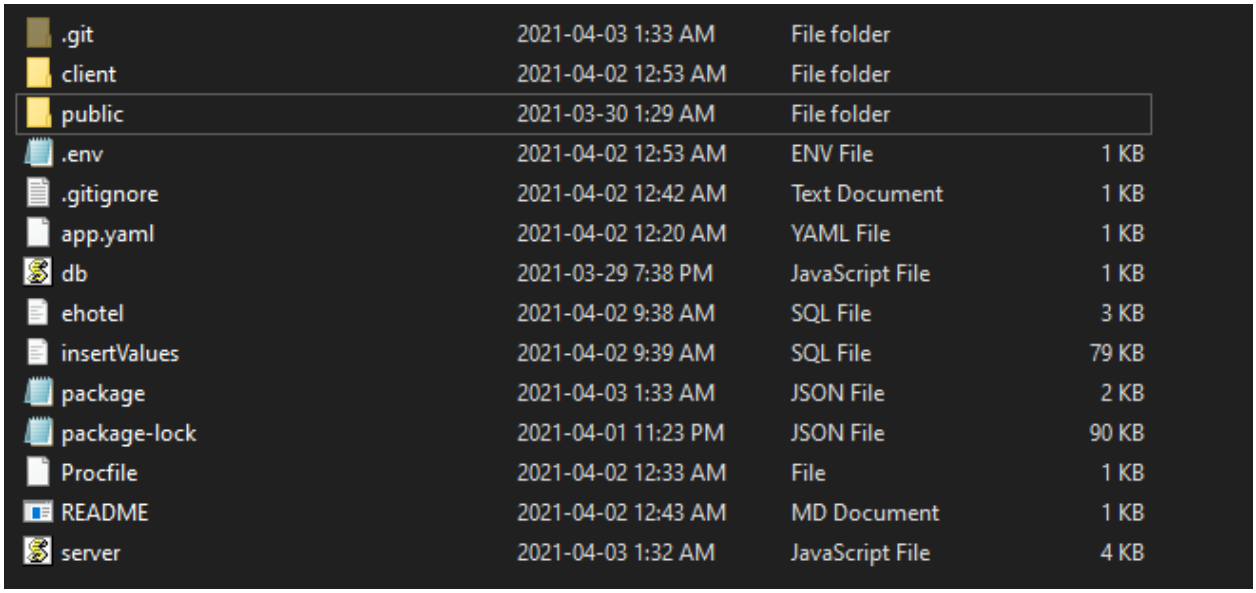
First, to set up the tables and other database parameters in PostgreSQL server, open pgAdmin. Within pgAdmin, inside your PostgreSQL database, right-click on the Schemas' tab and click 'New Schema...'. Within the pop-up window type the name of the schema as "ehotel" and then click 'OK'. The ehotel schema should now be visible under the 'Schemas' tab. If it is not, then simply refresh the database using the refresh button on the top and it should appear.



As can be seen in the above screenshot, the ‘ehotel’ schema has been created. Next step includes opening the Query tool by right clicking on the ehotel schema. This will open the main interface that deals with SQL commands to manipulate the database. We first copy all the contents of the ehotel.sql file into the query tool. This will create 5 tables that we will be using for this project. They are (‘customer’, ‘employee’, ‘hotel_chain’, ‘hotel’ and ‘room’). If they do not initially show up, then refresh the database.

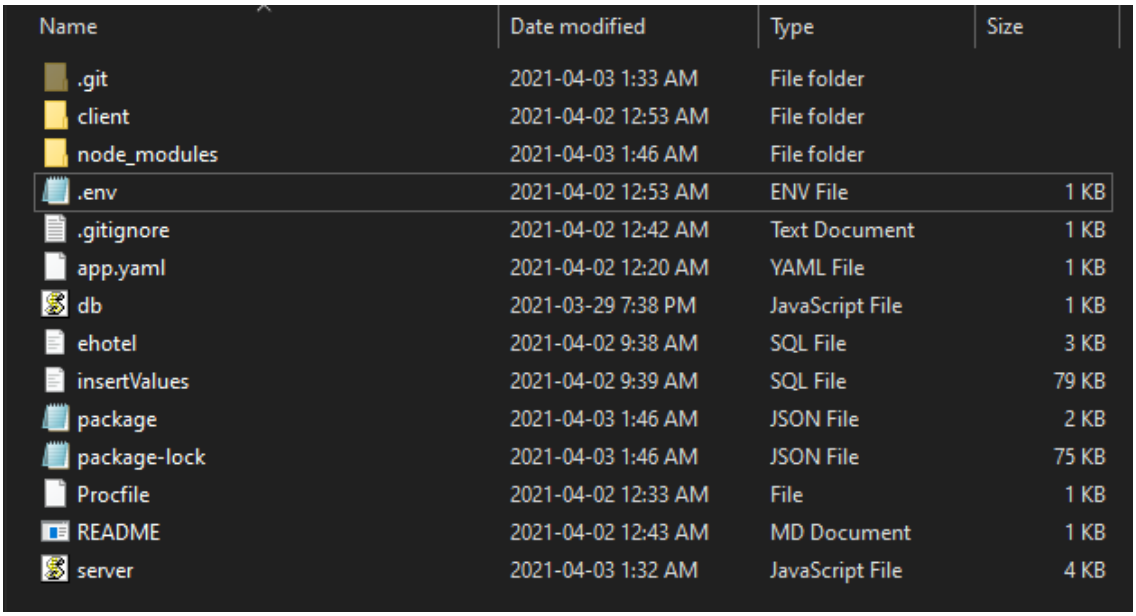
Now that the tables and relations of the database have been created, it is time to insert the data. To do this open a Query tool once again. Go to the “insertValues.sql” file within the Project Submission folder. Open this file and copy all of its contents. Go back to the Query tool and paste all of the contents into the Query tab, then run the Query. The Query should execute successfully if all of the steps above have been executed correctly. The Query tool may now be closed. All the tables in the database should now be populated with mock data.
To verify that the data has been inserted, right-click on any table and click ‘View/Edit Data > All Rows’ and you will see a window pop up containing all of the inserted data. This concludes the set-up of the PostgreSQL database.

Here are all project files required for the installation. All that is required to install the project is to run the ‘npm install’ command in the root directory and ‘npm install’ again in the client directory. The root directory contains all the server files. The main file is server.js which contains all the API endpoints of the project that sends/receives data from the postgresql database. After running the ‘npm install’ command on the terminal in both the root and client directories, a new folder named node_modules will be created. This will contain all the dependencies to run the project. The client folder contains all the frontend react files for the project



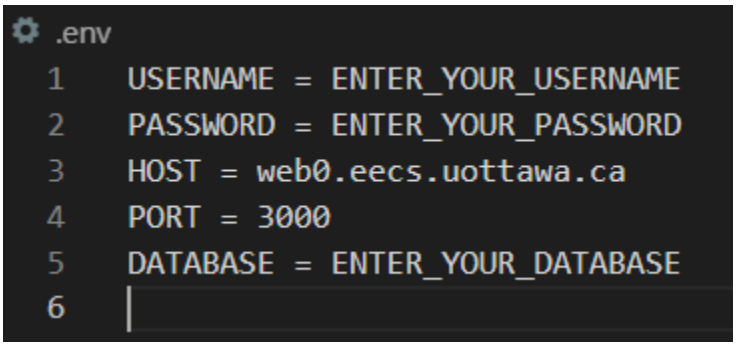
.git	2021-04-03 1:33 AM	File folder	
client	2021-04-02 12:53 AM	File folder	
public	2021-03-30 1:29 AM	File folder	
.env	2021-04-02 12:53 AM	ENV File	1 KB
.gitignore	2021-04-02 12:42 AM	Text Document	1 KB
app.yaml	2021-04-02 12:20 AM	YAML File	1 KB
db	2021-03-29 7:38 PM	JavaScript File	1 KB
ehotel	2021-04-02 9:38 AM	SQL File	3 KB
insertValues	2021-04-02 9:39 AM	SQL File	79 KB
package	2021-04-03 1:33 AM	JSON File	2 KB
package-lock	2021-04-01 11:23 PM	JSON File	90 KB
Procfile	2021-04-02 12:33 AM	File	1 KB
README	2021-04-02 12:43 AM	MD Document	1 KB
server	2021-04-03 1:32 AM	JavaScript File	4 KB

After the dependencies have been installed, this is what the folder contents will look like. The new node_modules folder has been created. The same command needs to be run inside the client folder and another node_modules is required there.



Name	Date modified	Type	Size
.git	2021-04-03 1:33 AM	File folder	
client	2021-04-02 12:53 AM	File folder	
node_modules	2021-04-03 1:46 AM	File folder	
.env	2021-04-02 12:53 AM	ENV File	1 KB
.gitignore	2021-04-02 12:42 AM	Text Document	1 KB
app.yaml	2021-04-02 12:20 AM	YAML File	1 KB
db	2021-03-29 7:38 PM	JavaScript File	1 KB
ehotel	2021-04-02 9:38 AM	SQL File	3 KB
insertValues	2021-04-02 9:39 AM	SQL File	79 KB
package	2021-04-03 1:46 AM	JSON File	2 KB
package-lock	2021-04-03 1:46 AM	JSON File	75 KB
Procfile	2021-04-02 12:33 AM	File	1 KB
README	2021-04-02 12:43 AM	MD Document	1 KB
server	2021-04-03 1:32 AM	JavaScript File	4 KB

Lastly, After both the frontend and the server dependencies have been installed, you need to enter your personal user credentials in the .env file in order to successfully connect to your pgadmin database. This can be achieved by doing the following. Open .env file → replace the placeholder credentials with your own unique credentials. For example : set the username to your own username, the password to your own pgadmin database password and lastly the database name to the specific database in pgadmin that you have access to. Then save the file and close it.



After you have entered your own credentials in the .env file. run the ‘npm run dev’ command in the root directory of the project (where the server.js file is). This should start the server in localhost:3000 and the react app in localhost:4545. This will automatically open a browser tab and start the web-based application ‘eHotel’.

6.0 Project Walk-through

As mentioned earlier, after a successful installation of the project, the user can then run the application using the ‘npm run dev’ command in the root directory of the project. This is the landing page of the ‘ehotel’ application that the user is taken to when they first start the app. It contains all the necessary routes described in the project guidelines for deliverable 2 (Q7)

Welcome to the eHotel Directory

Please select an option

Customer Online Booking

Employee Room Lookup

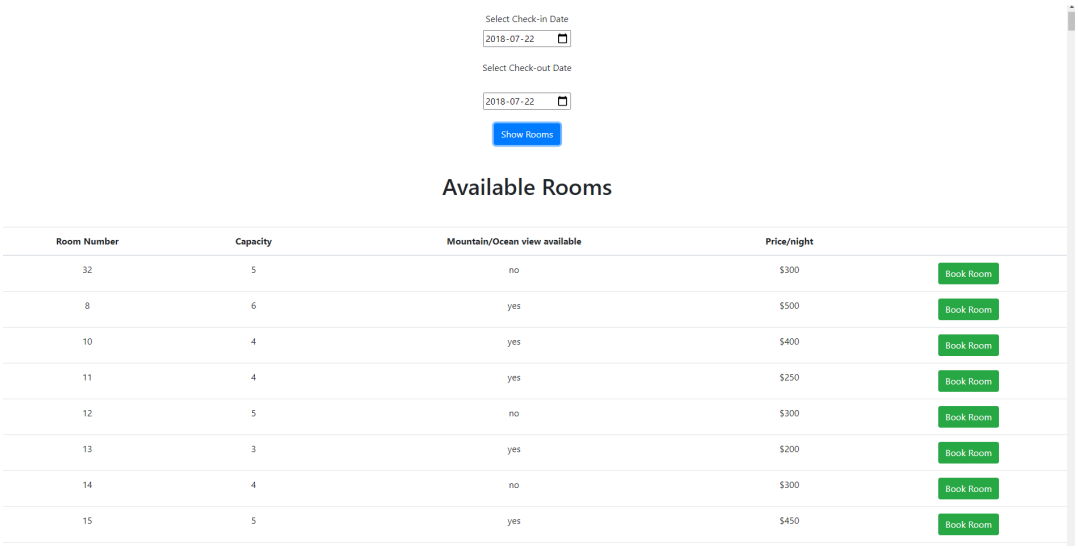
Confirm Customer Booking

Customer Registration - Walk In

Enter Customer Payment

Created by Saffat Aziz

Below is the page the user gets redirected to when they click on ‘Customer Online Booking’. Here they can pick a date range and all the available rooms will be displayed for them to book.



Available or Booked Rooms

Room Number	Capacity	Status	
5	5	booked	
54	4	booked	
32	5	available	Rent room for customer
8	6	available	Rent room for customer
10	4	available	Rent room for customer
11	4	available	Rent room for customer
12	5	available	Rent room for customer
13	3	available	Rent room for customer
14	4	available	Rent room for customer
15	5	available	Rent room for customer

The above is a screenshot of the employee lookup page, where they can view all the available or booked rooms. They can also book a room for a walk-in customer who shows up to the hotel without a prior booking. To register a walk-in customer, the following information has to be filled in.

Customer Registration - Walk In

First Name | Last Name

Your Business Email

Your Sin Number

Room Number

City | Country

Check In Date 2018-07-22

Check In Date 2018-07-22

Confirm Registration

Employees can then enter the customer payment information during the renting confirmation .

Customer Payment Confirmation

Customer Name

Room Number

Amout CAD

Confirm payment for room

Here are the Database tables for this project.

saziz028/saziz028@saffat aziz

Query Editor | Query History

```
1  select * from ehotel.hotel_chain
```

Data Output | Explain | Messages | Notifications

	name [PK] character varying (50)	street_name character varying (50)	unit_number integer	city character varying (20)	province_state character varying (20)	postal_code character varying (20)	country character varying (20)	email_address character varying (50)	phone_number character varying (20)	total_hotels integer
1	Ritz	Rideau Street	16	Ottawa	Ontario	K1N 5K0	Canada	Ritz@gmail.com	289-516-8765	8
2	Mariot	Sam Street	30	New York	New York	5HG 674	United States	Mariot@gmail.com	675-876-9042	8
3	Hotel Plaza	Williams Street	27	London	England	4HY 786	Europe	HotelPlaza@gmail.com	456-789-0001	8
4	Night Inn	Wallaby Street	8	Kansas City	Kansas	3JR 978	United States	NightInn@gmail.com	234-456-8765	8
5	Best Western	Ottawa Street	5	Vancouver	British Columbia	8PH 453	Canada	BestWestern@gmail.com	123-678-9023	8
6	Andaz	325 Dalhousie Street	5	Ottawa	Ontario	M3A 2G1	Canada	andaz123@gmail.com	123-456-7890	8

saziz028/saziz028@saffat aziz

Query Editor | Query History

```
1  select * from ehotel.hotel
2
```

Data Output | Explain | Messages | Notifications

	hotel_id [PK] integer	name character varying (50)	num_booked integer	num_available integer	rating integer	email_address character varying (50)	phone_number character varying (20)	street_name character varying (50)	unit_number integer	city character varying (20)	province_state character varying (20)	postal_code character varying (20)
1	1	Ritz	0	40	3	Ritz1@gmail.com	567-789-0098	Mike Street	78	Hamilton	Ontario	6GH 756
2	2	Ritz	0	23	4	Ritz2@gmail.com	456-675-9876	Appleby Street	90	Quebec City	Quebec	5JE 345
3	3	Ritz	0	78	2	Ritz3@gmail.com	234-567-8901	Hamilton Street	8	Calgary	Alberta	6TH 243
4	4	Ritz	0	90	5	Ritz4@gmail.com	235-467-9801	January Street	658	Buffalo	New York	9JK 856
5	5	Ritz	0	56	3	Ritz5@gmail.com	435-785-9076	Elk Street	7806	Rio de Janeiro	Brazil	2NM 345
6	6	Ritz	0	77	3	Ritz6@gmail.com	124-567-7865	Milk Street	754	Hamilton	Ontario	3FD 543
7	7	Ritz	0	89	4	Ritz7@gmail.com	089-756-4657	Middle Street	234	Montreal	Quebec	4FG 231
8	8	Ritz	0	100	4	Ritz8@gmail.com	234-765-9087	Main Street	123	Phoenix	Arizona	7HJ 901
9	9	Mariot	0	40	3	mariot1@gmail.com	123-456-7896	Jack Street	21	Toronto	Ontario	K1N 256
10	10	Mariot	0	40	2	mariot2@gmail.com	321-789-2431	Pharmacy Avenue	2	Montreal	Qubec	L16 543
11	11	Mariot	0	40	3	mariot3@gmail.comm	643-219-9999	Danforth Street	51	Toronto	Ontario	M1L 2N5
12	12	Mariot	0	40	3	mariot4@gmail.com	647-456-8976	Nowhere Street	31	Vancouver	British Columbia	321 4N5
13	13	Mariot	0	40	3	mariot5@gmail.com	613-222-7777	Somewhere Street	5	New York City	New York	41N 58Z
14	14	Mariot	0	40	3	mariot6@gmail.com	456-231-5645	Further Land	21	Qubec City	Qubec	N56 43K
15	15	Mariot	0	40	3	mariot7@gmail.com	819-2314-4326	James Street	91	Ottawa	Ontario	K1N 2SL
16	16	Mariot	0	40	3	mariot8@gmail.com	521-999-2365	Eglinton Avenue	56	Waterloo	Ontario	456 3K7
17	17	Hotel Plaza	0	45	5	HotelPlaza1@gmail.com	905-786-8976	Elm Street	98	Boston	Massachussets	6HT 764
18	18	Hotel Plaza	0	63	4	HotelPlaza2@gmail.com	456-908-7654	Applewide Street	107	Quebec City	Quebec	3FG 543
19	19	Hotel Plaza	0	102	3	HotelPlaza3@gmail.com	345-765-0987	Mighty Street	8054	Calgary	Alberta	9JK 754
20	20	Hotel Plaza	0	77	2	HotelPlaza4@gmail.com	213-435-6576	89th Street	654	Hamilton	Ontario	3RD 546
21	21	Hotel Plaza	0	86	4	HotelPlaza5@gmail.com	907-456-9987	Lake Street	365	Kitchener	Ontario	4FT 674

saziz028/saziz028@saffat aziz

Query Editor | Query History

```
1  select * from ehotel.room
2
```

Data Output | Explain | Messages | Notifications

	room_num [PK] integer	hotel_id integer	status character varying (20)	capacity integer	price integer	view character varying (20)	extendable character varying (20)	start_date date	end_date date	pending_balance boolean
1	2	1	rented	5	300	no	no	[null]	[null]	true
2	333	34	rented	3	200	yes	no	2021-03-31	2021-03-31	false
3	49	5	rented	5	450	no	yes	2018-07-02	2018-12-25	true
4	5	1	booked	5	450	yes	no	[null]	[null]	false
5	7	1	rented	4	250	yes	yes	2018-07-11	2018-10-18	true
6	54	6	booked	4	300	no	yes	[null]	[null]	false
7	90	9	rented	4	400	yes	no	2018-07-06	2018-07-20	false
8	32	4	available	5	300	no	no	[null]	[null]	false
9	8	1	available	6	500	yes	no	[null]	[null]	false
10	3	1	rented	3	200	yes	no	[null]	[null]	false
11	10	1	available	4	400	yes	no	[null]	[null]	false
12	11	2	available	4	250	yes	no	[null]	[null]	false
13	12	2	available	5	300	no	no	[null]	[null]	false
14	13	2	available	3	200	yes	no	[null]	[null]	false
15	14	2	available	4	300	no	yes	[null]	[null]	false
16	15	2	available	5	450	yes	no	[null]	[null]	false
17	16	2	available	2	150	yes	no	[null]	[null]	false
18	17	2	available	4	250	yes	yes	[null]	[null]	false
19	18	2	available	6	500	yes	no	[null]	[null]	false
20	19	2	available	5	450	no	yes	[null]	[null]	false
21	6	1	rented	2	150	yes	no	2018-07-22	2019-01-18	false

saziz028/saziz028@saffat aziz

Query Editor | Query History

```
1  select * from ehotel.employee
2
```

Data Output | Explain | Messages | Notifications

	ssn_sin [PK] integer	username character varying (20)	password character varying (20)	hotel_id integer	fname character varying (20)	lname character varying (20)	position_role character varying (50)	street_name character varying (20)	unit_number integer	city character varying (20)	province_state character varying (20)
1	12345678	Frank01	462102	1	Frank	Underwood	Manager	Rockline Avenue	5	Ottawa	Ontario
2	32165497	Harry01	1honEJ	1	Harry	Henry	Chef	Craigton Drive	41	Toronto	Ontario
3	98745612	Karen01	MWiXnY	1	Karen	Page	House Keeping	Borugh Drive	23	Montreal	Qubec
4	95135720	Rick01	zFxeWJ	2	Rick	Henry	Manager	Clearence Street	23	Ottawa	Ontario
5	15935785	Jack01	jGZDvO	2	Jack	Dcruze	Chef	Maryville Road	98	Vancouver	British Columbia
6	76134985	Saimoon01	zFxeWJ	2	Saimoon	Azad	House Keeping	Eglinton E	18	Kingston	Ontario
7	96543654	John1	kNk0Y1	3	John	Stark	Manager	Winterfell	6	Toronto	Ontario
8	95463658	Arya01	aEjjZw	3	Arya	Lannister	Chef	Gerard Street	26	Montreal	Qubec
9	36549219	Tom01	f4fYGN	3	Tom	Whatever	House Keeping	Warden Ave	36	Ottawa	Ontario
10	98796341	Kevin01	9XGKVW	4	Kevin	Potter	Manager	Danforth	5	Toronto	Ontario
11	14774163	Ginny01	s8NChn	4	Ginny	Wisely	Chef	Peterborough	10	Ottawa	Ontario
12	85225864	Severous01	o6BSml	4	Severous	Snape	House Keepeing	Hells Kitchen	15	New York City	New York
13	91537543	Ronnald01	Kv8cAG	5	Ronnald	Wisely	Manager	Monster Park	20	Ottawa	Ontario
14	19523762	Christopher01	ZCNifT	5	Christopher	Peterson	Chef	Nevermind	34	Toronto	Ontario
15	60238961	Gautham01	NFRFnu	5	Gautham	Sundararajan	House Keepeing	Bockline Avenue	6	Qubec City	Qubec
16	20315895	Elma01	bjYY3J	6	Elma	Imam	Manager	Princess Castle	30	Montreal	Qubec
17	63698524	Terence01	Yenmyx	6	Terence	Maby	Chef	Lees Avenue	90	Ottawa	Ontario
18	96587452	Simon01	yMaKmr	6	Simon	Zawardo	House Keeping	Wherever	21	Toronto	Ontario
19	32266549	Lemon01	EupXdF	7	Lemon	Padjnd	Manager	Dairy	100	Qubec City	Qubec
20	16985632	Kobban01	4V2rbZ	7	Kobban	shbmdjds	Chef	Layz	369	Ottawa	Ontario
21	69874522	Aftab01	l0CVoa	7	Aftab	Ahmed	House Keeping	Cheese	965	Toronto	Ontario

saziz028/saziz028@saffat aziz

Query Editor | Query History

```
1  select * from ehotel.customer
2
```

Data Output | Explain | Messages | Notifications

	ssn_sin [PK] integer	password character varying (255)	fname character varying (255)	lname character varying (255)	registration_date character varying (255)
1	9302618	Autumn01	Saffat	Aziz	23
2	3434	[null]	Gandu	Amin	22
3	199	[null]	Hasan	Gandu	2021/03/30
4	232323	[null]	Hubba	Bubba	2021-03-30
5	332233	[null]	Saffat	Aziz	2021-04-01
6	1	[null]	Hubba	Bubba	2021-03-30
7	2	[null]	Alex	rod	2021-03-30
8	454545	[null]	Masuma	Hamid	2021-04-02
9	3	[null]	Mark	Cuban	2021-03-30
10	4	[null]	Mark	Hamill	2021-03-30
11	5	[null]	Mandy	Aziz	2021-03-30
12	12	[null]	Hasan	Shahzad	2021-04-02
13	42	[null]	Hasan	Shahzad	2021-04-02
14	999999	[null]	Hasan	Shahzad	2021-04-02
15	321	[null]	Raiyan	Aziz	2021-04-02
16	4321	[null]	gandu	aziz	2021/04/02
17	44322212	[null]	saffat	aziz	2021-04-03
18	102618	Autumn01	Hasan	Shahzad	23
19	32618	Autumn01	Taha	Shahzad	23
20	40218	Autumn01	Adam	Shahzad	23
21	50418	Autumn01	Teia	Police	23

Deliverable 2 Answers

4) The database has been implemented using the code included in the ehotel.sql file. Here is a screenshot of some of the SQL code used to implement all the tables. We had to make a few minor changes to the E-R diagram to accommodate the implementation.

ehotel.sql

```
12  CREATE TABLE HOTEL_CHAIN(
13  NAME VARCHAR(50) NOT NULL,
14  STREET_NAME VARCHAR(50),
15  UNIT_NUMBER INT,
16  CITY VARCHAR(20),
17  PROVINCE_STATE VARCHAR(20),
18  POSTAL_CODE VARCHAR(20),
19  COUNTRY VARCHAR(20),
20  EMAIL_ADDRESS VARCHAR(50),
21  PHONE_NUMBER VARCHAR(20),
22  TOTAL_HOTELS INT,
23  CONSTRAINT PK_HOTEL_CHAIN PRIMARY KEY(NAME)
24  );
25  -- ************************************
26  -- Hotel Table Creation
27  -- ************************************
28  /*Create Hotel Table*/
29  CREATE TABLE HOTEL(
30  HOTEL_ID INT NOT NULL,
31  NAME VARCHAR(50),
32  NUM_BOOKED INT,
33  NUM_AVAILABLE INT,
34  RATING INT,
35  EMAIL_ADDRESS VARCHAR(50),
36  PHONE_NUMBER VARCHAR(20),
37  STREET_NAME VARCHAR(50),
38  UNIT_NUMBER INT,
39  CITY VARCHAR(20),
40  PROVINCE_STATE VARCHAR(20),
41  POSTAL_CODE VARCHAR(20),
42  COUNTRY VARCHAR(20),
43  HOTEL_ROOMS INT,
44  CONSTRAINT PK_HOTEL PRIMARY KEY(HOTEL_ID),
45  CONSTRAINT FK_HOTEL_NAME FOREIGN KEY (NAME) REFERENCES HOTEL_CHAIN (NAME) MATCH SIMPLE,
46  CONSTRAINT HOTEL_RATING_CHECK CHECK (RATING >=1 AND RATING <= 5)
```

5) The constraints included in the table are shown in the below screenshot. All the remaining constraints can be found in the ehotel.sql file.

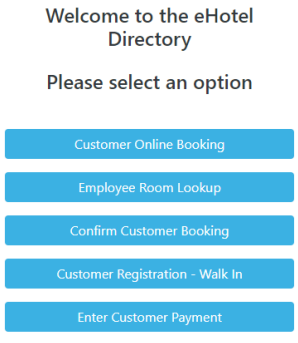
```
CREATE TABLE HOTEL(
HOTEL_ID INT NOT NULL,
NAME VARCHAR(50),
NUM_BOOKED INT,
NUM_AVAILABLE INT,
RATING INT,
EMAIL_ADDRESS VARCHAR(50),
PHONE_NUMBER VARCHAR(20),
STREET_NAME VARCHAR(50),
UNIT_NUMBER INT,
CITY VARCHAR(20),
PROVINCE_STATE VARCHAR(20),
POSTAL_CODE VARCHAR(20),
COUNTRY VARCHAR(20),
HOTEL_ROOMS INT,
CONSTRAINT PK_HOTEL PRIMARY KEY(HOTEL_ID),
CONSTRAINT FK_HOTEL_NAME FOREIGN KEY (NAME) REFERENCES HOTEL_CHAIN (NAME) MATCH SIMPLE,
CONSTRAINT HOTEL_RATING_CHECK CHECK (RATING >=1 AND RATING <= 5)
);
```

```
CREATE TABLE ROOM(
ROOM_NUM INT NOT NULL,
HOTEL_ID INT NOT NULL,
STATUS VARCHAR(20),
CAPACITY INT,
PRICE INT,
VIEW VARCHAR(20),
EXTENDABLE VARCHAR(20),
START_DATE DATE,
END_DATE DATE,
PENDING_BALANCE BOOLEAN,
CONSTRAINT PK_ROOM PRIMARY KEY(ROOM_NUM),
CONSTRAINT STATUS_CHECK CHECK (STATUS IN ('available','booked','rented')),
CONSTRAINT VIEW_CHECK CHECK (VIEW IN ('yes','no')),
CONSTRAINT EXTENDABLE_CHECK CHECK (EXTENDABLE IN ('yes','no'))
);
```

6) All the tables have been populated with diverse data. The insert commands can be found in the insertValues.sql file. Here is a screenshot of some of the sql commands.

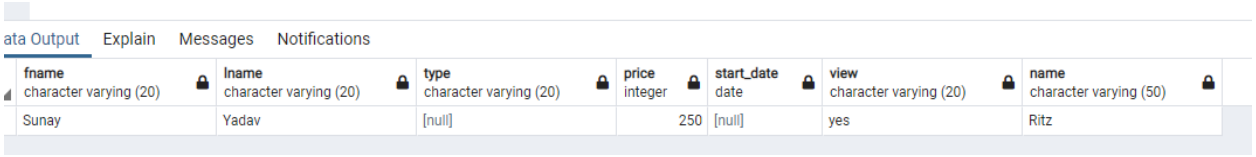
```
INSERT INTO ROOM VALUES (485, 48,  'available',5, 450, 'yes', 'no', null, null, false);
INSERT INTO ROOM VALUES (486, 48,  'available',2, 150, 'yes', 'no', null, null, false);
INSERT INTO ROOM VALUES (487, 48,  'available',4, 250, 'yes', 'yes', null, null, false);
INSERT INTO ROOM VALUES (488, 48,  'available',6, 500, 'yes', 'no', null, null, false);
INSERT INTO ROOM VALUES (489, 48,  'available',5, 450, 'no', 'yes', null, null, false);
INSERT INTO ROOM VALUES (490, 48,  'available',4, 400, 'yes', 'no', null, null, false);

-- ****************************
-- Inserting Values Hotel_Chain
-- ****************************

INSERT INTO HOTEL_CHAIN VALUES ('Ritz','Rideau Street',16,'Ottawa','Ontario','K1N 5K0', 'Canada', 'Ritz@gmail.com','289-516-8765',8);
INSERT INTO HOTEL_CHAIN VALUES ('Mariot','Sam Street',30,'New York','New York','5HG 674', 'United States', 'Mariot@gmail.com','675-876-9042',8);
INSERT INTO HOTEL_CHAIN VALUES ('Hotel Plaza','Williams Street',27,'London','England','4HY 786', 'Europe', 'HotelPlaza@gmail.com','456-789-0001',
INSERT INTO HOTEL_CHAIN VALUES ('Night Inn','Wallaby Street',8,'Kansas City','Kansas','3JR 978', 'United States', 'NightInn@gmail.com','234-456-8
INSERT INTO HOTEL_CHAIN VALUES ('Best Western','Ottawa Street',5,'Vancouver','British Columbia','8PH 453', 'Canada', 'BestWestern@gmail.com','123
INSERT INTO HOTEL_CHAIN VALUES ('Andaz','325 Dalhousie Street',5,'Ottawa','Ontario','M3A 2G1', 'Canada', 'andaz123@gmail.com','123-456-7890',8);

-- ******************************************
-- Inserting Values Hotel -> 8 Hotels per Chain
-- ******************************************

INSERT INTO HOTEL VALUES (1, 'Ritz', 0, 40, 3, 'Ritz1@gmail.com', '567-789-0098', 'Mike Street', 78, 'Hamilton', 'Ontario', '6GH 756', 'Canada',
INSERT INTO HOTEL VALUES (2, 'Ritz', 0, 23, 4, 'Ritz2@gmail.com', '456-675-9876', 'Appleby Street', 90, 'Quebec City', 'Quebec', '5JE 345', 'Cana
INSERT INTO HOTEL VALUES (3, 'Ritz', 0, 78, 2, 'Ritz3@gmail.com', '234-567-8901', 'Hamilton Street', 8, 'Calgary', 'Alberta', '6TH 243', 'Canada'
INSERT INTO HOTEL VALUES (4, 'Ritz', 0, 90, 5, 'Ritz4@gmail.com', '235-467-9801', 'January Street', 658, 'Buffalo', 'New York', '9JK 856', 'Unite
INSERT INTO HOTEL VALUES (5, 'Ritz', 0, 56, 3, 'Ritz5@gmail.com', '435-785-9076', 'Elk Street', 7806, 'Rio de Janeiro', 'Brazil', '2NM 345', 'Bra
INSERT INTO HOTEL VALUES (6, 'Ritz', 0, 77, 3, 'Ritz6@gmail.com', '124-567-7865', 'Milk Street', 754, 'Hamilton', 'Ontario', '3FD 543', 'Canada',
INSERT INTO HOTEL VALUES (7, 'Ritz', 0, 89, 4, 'Ritz7@gmail.com', '089-756-4657', 'Middle Street', 234, 'Montreal', 'Quebec', '4FG 231', 'Canada'
INSERT INTO HOTEL VALUES (8, 'Ritz', 0, 100, 4, 'Ritz8@gmail.com', '234-765-9087', 'Main Street', 123, 'Phoenix', 'Arizona', '7HJ 901', 'United S

```

7) As mentioned above, the frontend application to facilitate the user side interaction was created using the React (javascript) framework .The database administrator uses pgadmin (SQL developer) to insert/delete/update all information related to customers, employees, hotels and rooms. The online customers and employees will use the following web application to interact with the database.



8) Here are the list of queries
PLEASE NOTE, DUE TO FORMATTING ON DOCS SOME ‘ ‘ QUOTATIONS WILL BE DIFFERENT FROM WHAT WORKS ON PGADMIN4 SO DON’T FORGET TO CHANGE THEM OR THE QUERY WON’T WORK WHEN YOU TEST THEM AS TA’S

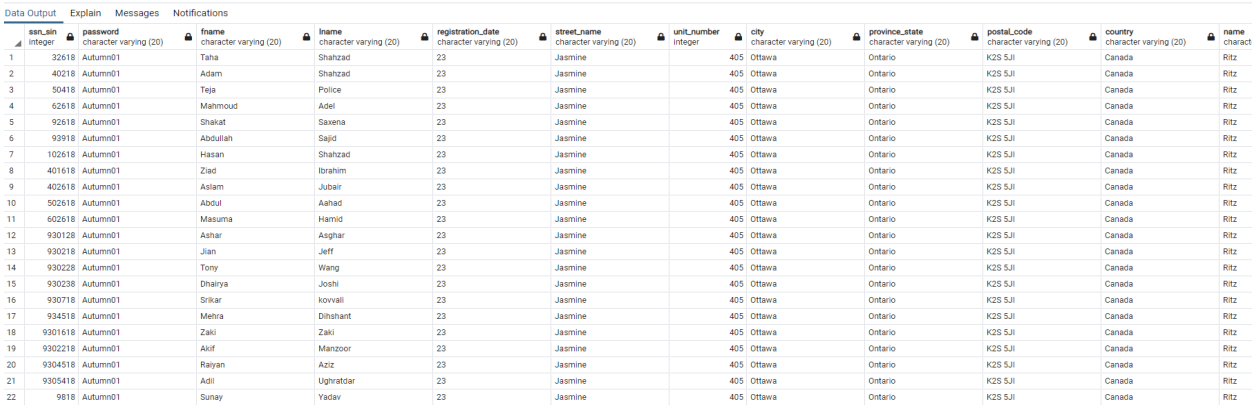
1. SELECT customer.fname, customer.lname, room.type, room.price, room.start_date, room.view, hotel.name
FROM customer, room, hotel
WHERE customer.room_ID = room.room_ID AND room.hotel_ID = hotel.hotel_ID AND room.status = ‘rented’
ORDER BY room.price ASC, room.start_date DESC;



ata Output | Explain | Messages | Notifications

fname character varying (20)	lname character varying (20)	type character varying (20)	price integer	start_date date	view character varying (20)	name character varying (50)
Sunay	Yadav	[null]	250	[null]	yes	Ritz

2. CREATE VIEW CustomerListView as
SELECT customer.SSN_SIN, customer.password, customer.fname, customer.lname, customer.registration_date, customer.street_name, customer.unit_number, customer.city, customer.province_state, customer.postal_code, customer.country, hotel.name
FROM customer, room, hotel
WHERE customer.room_ID = room.room_ID AND room.hotel_ID = hotel.hotel_ID
ORDER BY hotel.name;



Data Output | Explain | Messages | Notifications

	ssn_sin integer	password character varying (20)	fname character varying (20)	lname character varying (20)	registration_date character varying (20)	street_name character varying (20)	unit_number integer	city character varying (20)	province_state character varying (20)	postal_code character varying (20)	country character varying (20)	name charact
1	32618	Autumn01	Taha	Shahzad	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
2	40218	Autumn01	Adam	Shahzad	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
3	50418	Autumn01	Teja	Police	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
4	62618	Autumn01	Mahmoud	Adel	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
5	92618	Autumn01	Shakat	Saxena	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
6	93918	Autumn01	Abdullah	Sajid	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
7	102618	Autumn01	Hasan	Shahzad	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
8	401618	Autumn01	Ziad	Ibrahim	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
9	402618	Autumn01	Aslam	Jubair	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
10	502618	Autumn01	Abdul	Aahad	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
11	602618	Autumn01	Masuma	Hamid	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
12	930128	Autumn01	Ashar	Asghar	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
13	930218	Autumn01	Jian	Jeff	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
14	930228	Autumn01	Tony	Wang	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
15	930238	Autumn01	Dhairya	Joshi	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
16	930718	Autumn01	Srikar	kovvali	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
17	934518	Autumn01	Mehra	Dihshant	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
18	9301618	Autumn01	Zaki	Zaki	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
19	9302218	Autumn01	Akif	Manzoor	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
20	9304518	Autumn01	Raiyan	Aziz	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
21	9305418	Autumn01	Adil	Ughratdar	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz
22	9818	Autumn01	Sunay	Yadav	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	Ritz

3.
```
SELECT *
FROM Room
WHERE price =
        (SELECT MIN(price) FROM Room);
```

Data Output | Explain | Messages | Notifications

	room_id [PK] integer	hotel_id integer	status character varying (20)	capacity integer	price integer	view character varying (20)	extendable character varying (20)	start_date date	end_date date	pending_balance boolean	type character varying (20)
26	256	26	available	2	150	yes	no	[null]	[null]	false	[null]
27	266	27	available	2	150	yes	no	[null]	[null]	false	[null]
28	276	28	available	2	150	yes	no	[null]	[null]	false	[null]
29	286	29	available	2	150	yes	no	[null]	[null]	false	[null]
30	296	30	available	2	150	yes	no	[null]	[null]	false	[null]
31	306	31	available	2	150	yes	no	[null]	[null]	false	[null]
32	316	32	available	2	150	yes	no	[null]	[null]	false	[null]
33	326	33	available	2	150	yes	no	[null]	[null]	false	[null]
34	336	34	available	2	150	yes	no	[null]	[null]	false	[null]
35	346	35	available	2	150	yes	no	[null]	[null]	false	[null]
36	356	36	available	2	150	yes	no	[null]	[null]	false	[null]
37	366	37	available	2	150	yes	no	[null]	[null]	false	[null]
38	376	38	available	2	150	yes	no	[null]	[null]	false	[null]
39	386	38	available	2	150	yes	no	[null]	[null]	false	[null]
40	396	39	available	2	150	yes	no	[null]	[null]	false	[null]
41	406	40	available	2	150	yes	no	[null]	[null]	false	[null]
42	416	41	available	2	150	yes	no	[null]	[null]	false	[null]
43	426	42	available	2	150	yes	no	[null]	[null]	false	[null]
44	436	43	available	2	150	yes	no	[null]	[null]	false	[null]
45	446	44	available	2	150	yes	no	[null]	[null]	false	[null]
46	456	45	available	2	150	yes	no	[null]	[null]	false	[null]
47	466	46	available	2	150	yes	no	[null]	[null]	false	[null]
48	476	47	available	2	150	yes	no	[null]	[null]	false	[null]
49	486	48	available	2	150	yes	no	[null]	[null]	false	[null]

4.
```
SELECT *
FROM Room, hotel
WHERE room.room_id = hotel.hotel_id AND hotel.city = 'Ottawa'
ORDER BY hotel.rating ASC, room.price ASC;
```
Note: The entire query result was too long to screenshot (too many columns) but i included the rating and price in the screenshot to show the ordering.

	status character varying (20)	capacity integer	price integer	view character varying (20)	extendable character varying (20)	start_date date	end_date date	pending_balance boolean	type character varying (20)	hotel_id integer	name character varying (50)	num_booked integer	num_available integer	rating integer
5	available	2	150	yes	no	[null]	[null]	false	[null]	46	Andaz	0	40	3
5	available	5	300	no	no	[null]	[null]	false	[null]	42	Andaz	0	40	3
5	available	4	300	no	yes	[null]	[null]	false	[null]	44	Andaz	0	40	3
2	available	5	450	yes	no	[null]	[null]	false	[null]	15	Mariot	0	40	3
4	available	6	500	yes	no	[null]	[null]	false	[null]	38	Best Western	0	40	3

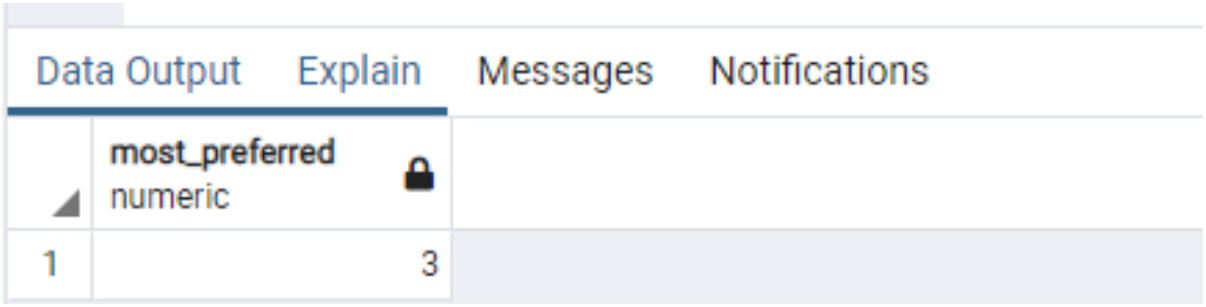
5.
```
SELECT *
FROM Room
WHERE room.status = 'rented' AND room.start_date = '2021-04-03';
```

	room_id [PK] integer	hotel_id integer	status character varying (20)	capacity integer	price integer	view character varying (20)	extendable character varying (20)	start_date date	end_date date	pending_balance boolean	type character varying (20)
1	1	1	rented	4	250	yes	no	2021-04-03	[null]	false	[null]

6. UPDATE Customer
SET phone = '519-234-5678'
WHERE fname = 'Sunay' AND lname = 'Yadav';

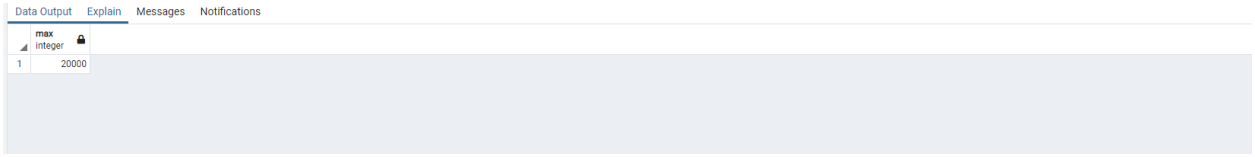
	ssn_sin [PK] integer	password character varying (20)	fname character varying (20)	lname character varying (20)	registration_date character varying (20)	street_name character varying (20)	unit_number integer	city character varyi	province_state character varying (20)	postal_code character varying (2	country character varyin	room_id integer	phone character varying (20)
1	9818	Autumn01	Sunay	Yadav	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	1	519-234-5678
2	32618	Autumn01	Taha	Shahzad	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	2	[null]
3	40218	Autumn01	Adam	Shahzad	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	3	[null]
4	50418	Autumn01	Teja	Police	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	4	[null]
5	62618	Autumn01	Mahmoud	Adel	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	5	[null]
6	92618	Autumn01	Shakat	Saxena	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	6	[null]
7	93918	Autumn01	Abdullah	Sajid	23	Jasmine	405	Ottawa	Ontario	K2S 5JI	Canada	7	[null]

7. SELECT FLOOR(AVG(preferred_rating)) AS most_preferred FROM customer;



8.

SELECT MAX(Employee.salary)
FROM Employee WHERE Employee.salary NOT IN (
SELECT MAX(Employee.salary) FROM Employee);



Member Contributions for Deliverable 2

Member Name	Contribution
Saffat Aziz	● Database Creation (Q4) ● Database Constraints (Q5) ● Populating the Database (Q6) ● Creating the Frontend React application (Q7) ● Project report (Parts 1,3,4,5 & 6)
Nabil Ali	● Updated relational model (Q4) ● Populating the Database (Q6) ● SQL queries (Q8) ● Project report (Part 2)
Anthony Polak	● SQL queries (Q8) ● Tested queries for Q8 (Q8), added photos and input