CSI 2132 Course Project

**a. DBMS and Programming Languages Used:**

The application was developed using the **PERN** stack, which includes:

PostgreSQL: A powerful, open-source relational database management system (DBMS).

Express.js: A Node.js web application framework used for building the backend server and APIs.

React.js: A JavaScript library for building user interfaces, used for the frontend.

Node.js: A JavaScript runtime environment used for building server-side applications.

**b. Installation Guide:**

Below are the specific steps to guide someone through installing the application:

1. Install Node.js:

Visit the Node.js website and download the installer for your operating system.

Follow the installation instructions provided on the website to install Node.js

2. Install PostgreSQL:

Visit the PostgreSQL website and download the installer for your operating system.

Follow the installation instructions provided on the website to install PostgreSQL.

During installation, remember the username, password, and port number used for PostgreSQL, as you'll need them later.

3. Clone the Repository:

Clone the repository containing the PERN stack application from the version control system where it's hosted (e.g., GitHub). In a terminal window, navigate to C:\...\HotelApp

4. Setup the Database:

Launch postgres and run the command `\i ./backend/database.sql`

This will setup all the necessary database functionalities.

6. Install Dependencies:

Navigate to the project directory in the terminal.

Run the command npm install to install the required dependencies for both the frontend and backend.

7. Start the Backend Server:

Navigate to the backend directory in the terminal.

Run the command npm start to start the Express.js server.

The backend server should now be running on a specified port (e.g., 5000).

8. Start the Frontend Server:

Navigate to the frontend directory in the terminal.

Run the command npm start to start the React development server.

The frontend server should now be running, and the application should automatically open in your default web browser.

9. Interact with the Application:

You can now interact with the application through the web browser.

Test different features and functionalities to ensure that the application is working as expected.

By following these steps, users should be able to install and run the PERN stack application on their local machines.

**c. DDL’s that Create Database**

This is the first half of our main .sql script to be executed before starting the project.  
(It can also be found in the github repository at HotelApp/backend/database.sql)

CREATE DATABASE hotelapp;

\c hotelapp

CREATE TABLE HotelChains (

ChainID SERIAL PRIMARY KEY,

ChainName VARCHAR(255),

HQAddress VARCHAR(255),

NumHotels INT,

ContactEmail VARCHAR(255),

ContactPhone VARCHAR(20)

);

CREATE TABLE Hotels (

HotelID SERIAL PRIMARY KEY,

ChainID INT,

HotelName VARCHAR(255),

Category INT,

HotelAddress VARCHAR(255),

NumRooms INT,

ContactEmail VARCHAR(255),

ContactPhone VARCHAR(20),

FOREIGN KEY (ChainID) REFERENCES HotelChains(ChainID)

);

CREATE TABLE Employee (

EID SERIAL PRIMARY KEY,

HotelID INT,

FullName VARCHAR(255),

Address VARCHAR(255),

SSN VARCHAR(11),

Role VARCHAR(100),

FOREIGN KEY (HotelID) REFERENCES Hotels(HotelID)

);

CREATE TABLE Customers (

CustomerID SERIAL PRIMARY KEY,

FullName VARCHAR(255),

Address VARCHAR(255),

IDType VARCHAR(50),

RegistrationDate DATE

);

CREATE TABLE Rooms (

RoomID SERIAL PRIMARY KEY,

HotelID INT,

RoomNumber INT,

PricePerNight DECIMAL(10, 2),

Capacity INT,

Seaview BOOLEAN,

MountainView BOOLEAN,

Extendable BOOLEAN,

MaintenanceNotes TEXT,

FOREIGN KEY (HotelID) REFERENCES Hotels(HotelID)

);

CREATE TABLE Bookings (

BookingID SERIAL PRIMARY KEY,

CustomerID INT,

RoomID INT,

CheckinDate DATE,

CheckoutDate DATE,

BookingDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID)

);

CREATE TABLE Rentings (

RentingID SERIAL PRIMARY KEY,

CustomerID INT,

RoomID INT,

EmployeeID INT,

CheckinDate DATE,

CheckoutDate DATE,

BookingDate DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),

FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID),

FOREIGN KEY (EmployeeID) REFERENCES Employee(EID)

);

CREATE TABLE Amenities (

AmenityID SERIAL PRIMARY KEY,

AmenityName VARCHAR(255)

);

CREATE TABLE RoomAmenities (

RoomID INT,

AmenityID INT,

PRIMARY KEY (RoomID, AmenityID),

FOREIGN KEY (RoomID) REFERENCES Rooms(RoomID),

FOREIGN KEY (AmenityID) REFERENCES Amenities(AmenityID)

);