MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Department of Computer Science and Engineering

Artificial Intelligence and Machine Learning

Mini Project Report
of
Database Systems Lab (CSE 2262)
by

DRUMIL HARESH VED
210962006
1
Section: A

PUNITH
210962012
4
Section: A

Department of Computer Science and Engineering
Manipal Institute of Technology, Manipal.
April 2023

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Manipal
08/05/2023

CERTIFICATE

This is to certify that the project titled **MiniProject Title** is a record of the bonafide work done by **Drumil Haresh Ved (Reg. No.210962006) Punith (Reg. No.210962012)** submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech.) in COMPUTER SCIENCE & ENGINEERING of Manipal Institute of Technology, Manipal, Karnataka, (A Constituent Institute of Manipal Academy of Higher Education), during the academic year 2022-2023.

Name and Signature of Examiners:

1. **Dr.Dinesh Acharaya, CSE Dept.**

TABLE OF CONTENTS

Abstract:

The objective of this project is to develop a hotel rating and listing website as a database system. The website aims to provide users with comprehensive information about hotels, including ratings and reviews, to help them make informed decisions when booking accommodations. The system will allow users to search for hotels based on various criteria, such as location, price range, amenities, and user ratings. Users will also have the ability to rate and review hotels based on their own experiences. By developing a reliable and user-friendly hotel rating and listing website, this project aims to enhance the user experience and simplify the process of finding suitable accommodations. It will serve as a valuable resource for travellers seeking trustworthy and comprehensive information about hotels, thereby facilitating informed decision-making and improving overall customer Satisfaction.

Chapter 1: Introduction

The proposed website would provide consumers with a one-stop solution for researching, comparing, and evaluating hotels based on a variety of factors. The platform will provide accurate and complete facts about hotels, including facilities, price, and location, by using a strong database system. Users will be able to search for hotels based on particular choices such as location, price range, and preferred facilities, allowing them to discover lodgings that meet their requirements.The inclusion of user-generated ratings and reviews is an important part of the project. Users will be able to rate and review hotels based on their personal experiences, offering vital information that might help other travellers make decisions. The website intends to improve transparency and dependability by offering a trustworthy forum for user evaluations, minimising the issue of biassed or outdated information that is commonly seen on existing hotel rating and listing websites.

This project is to improve the user experience and provide a useful resource for travellers by building a hotel rating and listing website as a complete database system. By providing factual information, user-generated reviews, and a simple search interface, the platform will enable consumers to make educated judgements. With the ability to ease the process of locating suitable rooms, the website will help to improve consumer happiness and confidence in the hotel decision process.

Chapter 2: PROBLEM STATEMENT & OBJECTIVES

Problem Statement:

With the ever-increasing number of hotels worldwide, finding the right accommodation can be a challenging task for travellers. The availability of online booking platforms has simplified the process, but users often struggle to make confident choices due to the lack of reliable and comprehensive information about hotels. The existing hotel rating and listing websites often suffer from issues such as outdated information, biassed reviews, and limited search capabilities. These problems create frustration for users and diminish the trustworthiness of the platforms. Moreover, the lack of a centralised and well-structured database system makes it difficult for users to compare and evaluate hotels effectively.

Functional Requirements:

1. Connecting the website front end to backend
2. Connecting the backend to front end
3. Pulling out all the data from database
4. Cleaning the data, and presenting only relevant parts of the data

Data Requirements:

1. Collecting the data of hotels which includes all the restaurant names, location , price range.
2. Getting the menu available in restaurants and their prices
3. Getting the data from the users like the rating for restaurants , review and rating for menu and restaurants

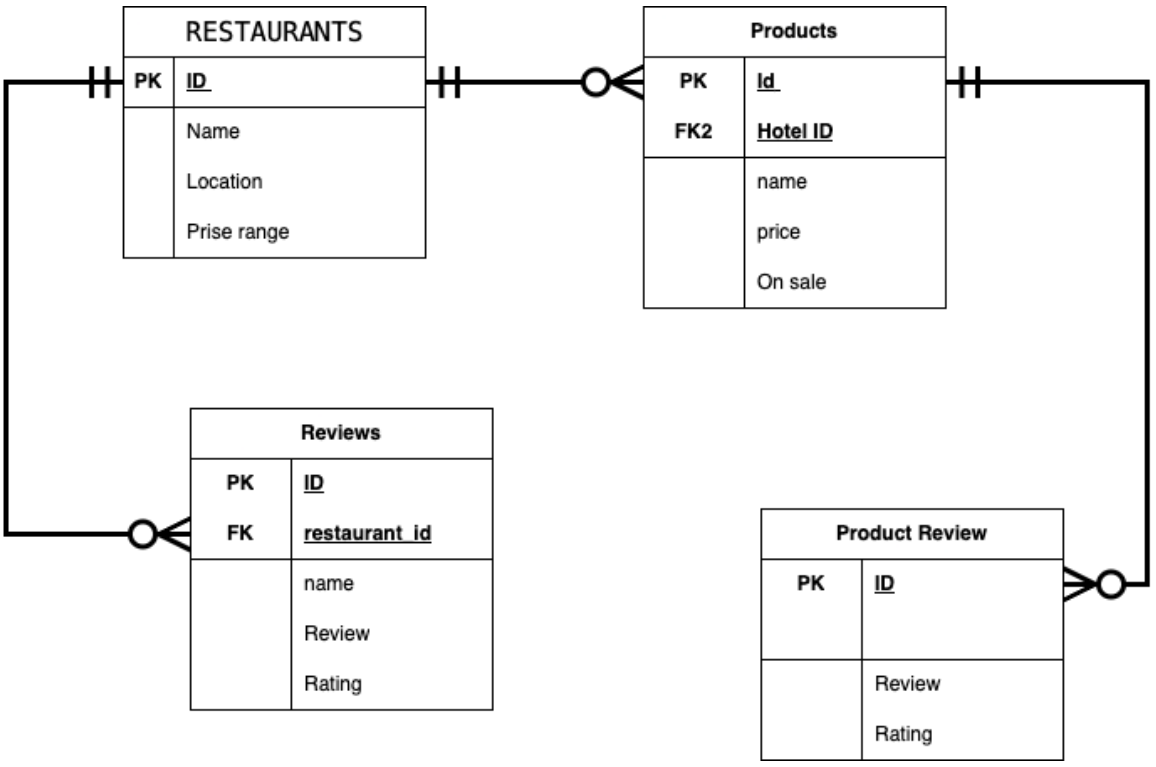
Objective:

This project's goal is to create a hotel rating and listing website as a database system. The website seeks to alleviate the difficulties that travellers encounter in locating credible and comprehensive hotel information. The project's precise goals are as follows:

1. Provide Accurate and Up-to-Date Information: The website will provide users with accurate and complete information on hotels, including amenities, pricing, and location. The platform strives to ensure that the information presented is current and trustworthy by using a strong database infrastructure.
2. Allow for User Ratings and Reviews: Users will be able to rate and evaluate hotels based on their own personal experiences. This tool seeks to give helpful information for other travellers while also increasing openness in the hotel selection process.
3. Improve User Decision-Making: The website will have advanced search features that will allow customers to filter and categorise hotels based on their preferences. The platform's goal is to enable users to make educated decisions about lodgings by offering comprehensive search results and user-generated reviews.

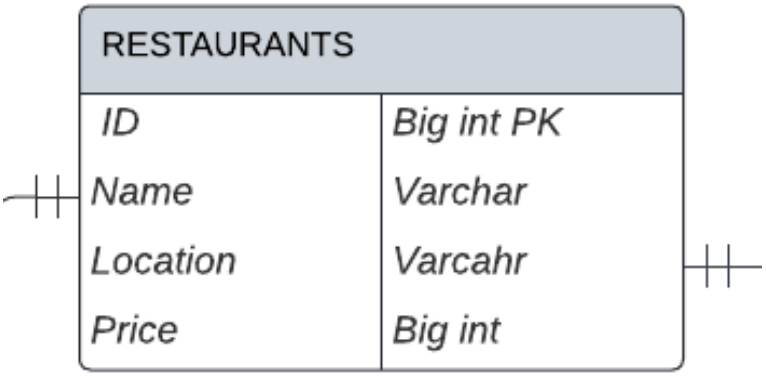
Chapter 3: Methodology

ER Diagram:



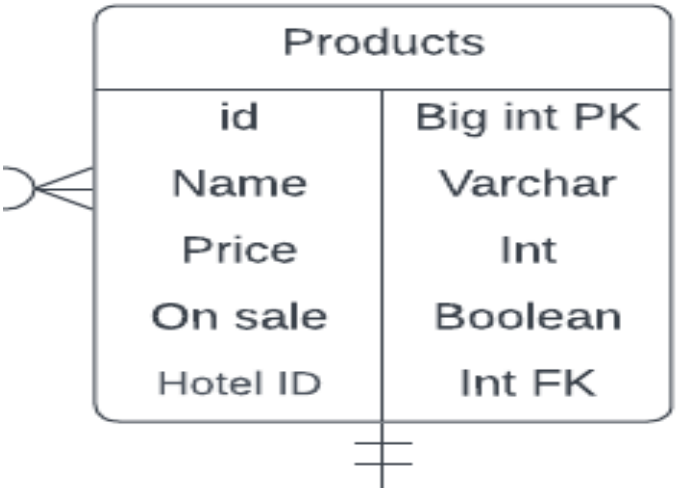
ER Diagram of the Restaurant's review system.

Relational Tables:




```
yelp=# select * from restaurants;
 id |       name        | location | price_range
----+-------------------+----------+-------------
  1 | vikram`s Falooda  | Udupi    |           2
  4 | Abishek`s Dosa    | Manglore |           3
  5 | Punith`s Sweets   | Manipal  |           2
(3 rows)
```

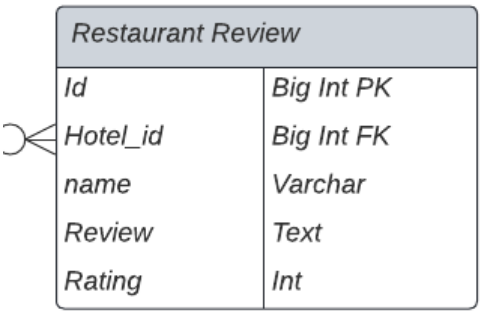
Restaurants Table:
id BIGSERIAL PRIMARY KEY,
name VARCHAR(50)
location VARCHAR(50)
price_range INT




```
yelp=# SELECT * From products;
 id |        name         | on_sale | hotel_id | price |                    description
----+---------------------+---------+----------+-------+-----------------------------------------------------
  2 | Mango falooda       | t       |        4 |   200 | made with fresh milk and mix of mangos, banans, apple
  3 | Mango falooda       | t       |        5 |   200 | made with fresh milk and mix of mangos, banans, apple
  4 | Banana Orio falooda | t       |        1 |   200 | made with fresh milk and mix of mangos, banans, apple
  5 | Banana Orio falooda | t       |        5 |   200 | made with fresh milk and mix of mangos, banans, apple
  6 | Banana Orio falooda | t       |        4 |   200 | made with fresh milk and mix of mangos, banans, apple
(5 rows)
```

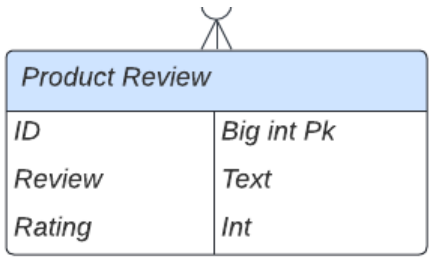
Products Table:

id BIGSERIAL Foreign key -> PRODUCT_REVIEW(product_id) ,
name VARCHAR(50)
price INT
on_sale boolean
hotel_id INT Foreign key -> restaurants(id)



```
UPDATE 6
[yelp=# SELECT * FROM  product_review;
 product_id |       review        | price
------------+---------------------+--------
          1 | good mango faluda   | 300.00
          2 | good mango faluda   | 300.00
          3 | good mango faluda   | 300.00
          4 | good mango faluda   | 300.00
          5 | good mango faluda   | 300.00
          6 | good mango faluda   | 300.00
(6 rows)
```

Restaurants Review:
id BIGSERIAL PRIMARY KEY,
restaurant_id BIGINT Foreign key -> restaurants(id),
name VARCHAR(50)
review TEXT
rating INT




```
;
 id | restaurant_id |     name     |   review    | rating
----+---------------+--------------+-------------+--------
  1 |             1 | vikram sood  | good food   |      4
  2 |             4 | vikram jia   | good food   |      4
  3 |             5 | john         | good food   |      4
  4 |             5 | harris       | good food   |      4
  8 |             4 | punith       | great place |      3
  9 |             4 | kishore      | great dosa  |      3
(6 rows)
```

Product Review:
product_id bigint PRIMARY KEY
review VARCHAR(50)
Rating Int

Necessary DDL Commands in Postgresql:

Restaurant Table :

```
-- this table holds all the hotels


CREATE TABLE RESTAURANTS(
   id BIGSERIAL PRIMARY KEY,
 -- SERAIAL HERE WILL AUTOMATICALLY GIVE A ID TO EVERY NEW RESTAURANT
   name VARCHAR(50) NOT NULL,
   location VARCHAR(50)  NOT NULL,
   price_range INT  NOT NULL CHECK(price_range >=1 and price_range <=5)
);
```

Products Table:

```

CREATE TABLE products(
   id  BIGSERIAL REFERENCES PRODUCT_REVIEW(product_id) ,
   name VARCHAR(50) NOT NULL,
   price INT NOT NULL,
   on_sale boolean NOT NULL ,
   hotel_id INT REFERENCES restaurants(id)
);
```

Product Review Table :

```
CREATE TABLE reviews (
   id BIGSERIAL NOT NULL PRIMARY KEY,
   restaurant_id BIGINT NOT NULL REFERENCES restaurants(id),
   name VARCHAR(50) NOT NULL,
   review TEXT NOT NULL,
   rating INT NOT NULL check(rating >= 1 and rating <= 5
   ));
```

PRODUCT_REVIEW Table:

```
CREATE TABLE PRODUCT_REVIEW(
   product_id bigint PRIMARY KEY,
   review VARCHAR(50),
   Rating Int
);
```

Database Connection:

```
const { Pool, Client } = require('pg')
// pools will use environment variables
// for connection information
const pool = new Pool()
pool.query('SELECT NOW()', (err, res) => {
 console.log(err, res)
 pool.end()
})
// you can also use async/await
const res = await pool.query('SELECT NOW()')
await pool.end()
// clients will also use environment variables
// for connection information
const client = new Client()
await client.connect()
const res = await client.query('SELECT NOW()')
await client.end()

const pool = new Pool();
module.exports = {
 query: (text, params) => pool.query(text, params),
};
```

.env file

```
PORT=3001

PGUSER=postgres
PGHOST=localhost
PGPASSWORD=password123
PGDATABASE=yelp
PGPORT=5432
```

Other SQL Queries:

1. This query made from the back end to the data base will return restaurant_id , count of total number of reviews , round off the average rating while making a join on tables Restaurants and Review

```
const restaurantRatingsData = await db.query(
     "select * from restaurants left join (select restaurant_id, COUNT(*),
TRUNC(AVG(rating),1) as average_rating from reviews group by restaurant_id) reviews
on restaurants.id = reviews.restaurant_id;"
   );
```

2. This query is to insert **Triggerand and PL SQL** a table for the trigger

```
CREATE table log_product_changes(
id  BIGSERIAL ,
name VARCHAR(50) NOT NULL,
price INT NOT NULL,
on_sale boolean NOT NULL ,
hotel_id INT,
action VARCHAR(3) ,
date_of_change TIMESTAMP DEFAULT now()
);


CREATE OR REPLACE FUNCTION log_product_changes_update()
 RETURNS TRIGGER AS $$
BEGIN
 -- Perform the INSERT operation here
 INSERT INTO log_product_changes (id, name, price, on_sale, hotel_id, action)
 VALUES (OLD.id, OLD.name, OLD.price, OLD.on_sale, OLD.hotel_id, 'upd');
  -- Return the modified row for the UPDATE operation to proceed
 RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION log_product_changes_delete()
 RETURNS TRIGGER AS $$
BEGIN
 -- Perform the INSERT operation here
 INSERT INTO log_product_changes (id, name, price, on_sale, hotel_id, action)
 VALUES (OLD.id, OLD.name, OLD.price, OLD.on_sale, OLD.hotel_id, 'del');
  -- Return the modified row for the UPDATE operation to proceed
 RETURN NEW;
END;
$$ LANGUAGE plpgsql;
-- Create the trigger
CREATE TRIGGER log_change_update
BEFORE UPDATE ON products
```

```
FOR EACH ROW
EXECUTE FUNCTION log_product_changes_update();

CREATE or replace TRIGGER log_change_delete
BEFORE delete ON products
FOR EACH ROW
EXECUTE FUNCTION log_product_changes_delete();
```

log_product_changes_update() and log_product_changes_delete() are the two procedures which will be called by log_change_update and log_change_delete triggers for updating and deleting the respective rows when updated and deleted respectively.

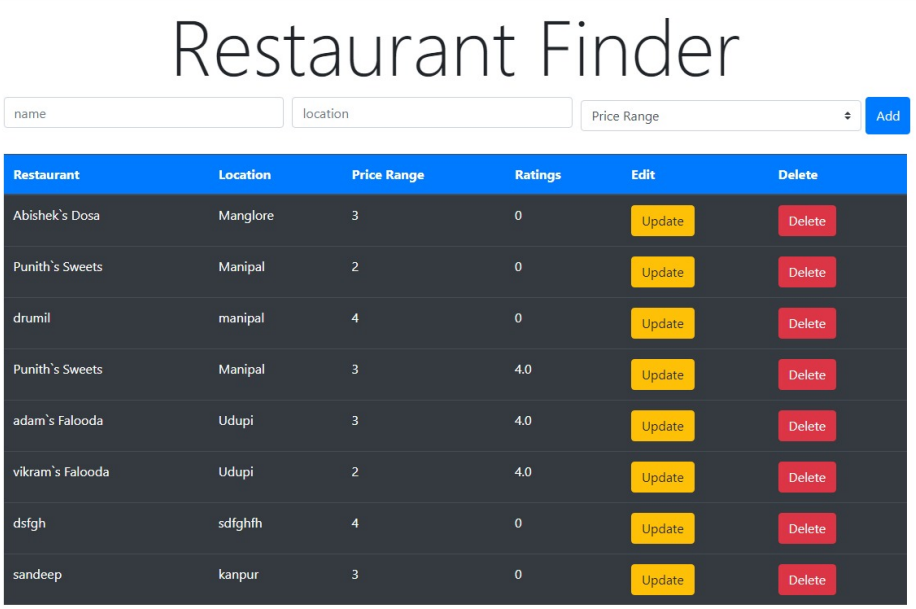
3. This query will return restaurant details like id , name , location , price range , and make a join with review table to return the respective reviews for the same table

```
SELECT  RESTAURANTS.id, RESTAURANTS.name, RESTAURANTS.location,
RESTAURANTS.price_range , reviews.rating FROM RESTAURANTS , reviews WHERE
RESTAURANTS.id = reviews.restaurant_id and RESTAURANTS.id = 1;
```

4. This query will retrieve product details like id , name, review , description form products and corresponding reviews from review table.

```
SELECT products.name, PRODUCTS.DESCRIPTION, products.price, reviews.rating from
products,reviews WHERE PRODUCTS.id = reviews.id and products.hotel_id = 2;
```

Chapter 4: Results and Snapshots



Restaurant	Location	Price Range	Ratings	Edit	Delete
Abishek`s Dosa	Manglore	3	0	Update	Delete
Punith`s Sweets	Manipal	2	0	Update	Delete
drumil	manipal	4	0	Update	Delete
Punith`s Sweets	Manipal	3	4.0	Update	Delete
adam`s Falooda	Udupi	3	4.0	Update	Delete
vikram`s Falooda	Udupi	2	4.0	Update	Delete
dsfgh	sdfghfh	4	0	Update	Delete
sandeep	kanpur	3	0	Update	Delete

Update Restaurant

Name

adam`s Falooda

Location

Udupi

Price Range

3

Submit

Punith`s Sweets

☆☆☆☆☆ (0)

vikram jia ★★★★☆	abhisex ★★★★★	wer ★★★☆☆
good food	wow	yfgvbhjnmk

Name

name

Rating

Rating

Review

Submit

Product	Price	Reviews	Edit	Delete
Mango falooda	₹ 400	0 reviews	Update	Delete
Mango falooda	₹ 400	0 reviews	Update	Delete
Banana Orio falooda	₹ 550	0 reviews	Update	Delete

Result:

The aforesaid project will result in a fully working hotel rating and listing website that also functions as a comprehensive database system. The website will deliver the following findings and benefits:

1. User-Friendly Interface: The website will have an intuitive and user-friendly interface, allowing users to effortlessly navigate and search for hotels. The user-centric features and simplified design will improve the overall user experience.

2. Users will get access to accurate and up-to-date information on hotels, such as facilities, pricing, and location. The database system will guarantee that the information presented is current and constantly updated.

3. User Ratings and Reviews: Based on their personal experiences, users will be able to rate and evaluate hotels on the site. This tool will give vital information for other travellers, allowing them to make more educated selections when selecting restaurants

Overall, the project will provide a functioning and user-friendly hotel rating and listing website with credible information, user-generated ratings, and extensive search capabilities. It will be a great resource for travellers, streamlining the process of discovering acceptable rooms and improving the whole hotel booking experience.

Chapter 5 : Conclusion

Finally, the hotel rating and listing website project attempts to overcome the difficulties that travellers encounter in locating credible and complete information on hotels. The project's goal is to improve the hotel choosing process by establishing a database system-driven platform that provides consumers with reliable information, user-generated reviews, and advanced search functions.

The development of a fully functional website will achieve the project's objectives, which include providing up-to-date information, allowing user ratings and reviews, improving user decision-making, facilitating secure booking and reservation, and ensuring administrative control and moderation.

In conclusion, the hotel rating and listing online project will be a vital resource for travellers, increasing the whole experience of locating and reserving Restaurants. The project's outcomes will lead to increased customer satisfaction, decision-making confidence, and a trusted environment for customers looking for accurate hotel information.

When completed, the project will result in a user-friendly website that streamlines the process of finding suitable dining. Users will benefit from accurate and trustworthy information, user-generated insights, and an easy-to-use interface that helps them to make educated decisions. Users will save time and effort by using the website's extensive search functions to filter and arrange hotels depending on their preferences.

Chapter 6 : Limitation and Future Work

Limitation:

1. This website can not be sure if the given review and information is correct or not.
2. We do not yet have any mechanism for photos
3. People will not trust such a website easily

Future Work:

We can add the following functionality :

1. Seach
2. Book reservation
3. Room Booking
4. Hate speech remover

.

Chapter 7 : References

REFERENCES:

1. Official documentation of Postgersql

   https://www.postgresql.org/files/documentation/pdf/15/postgresql-15-US.pdf
2. ER diagram : https://www.youtube.com/watch?v=QpdhBUYk7Kk
3. ER diagram : https://www.youtube.com/watch?v=lAtCySGDD48
4. Postgresql Queries: https://www.postgresqltutorial.com/
5. Node module for postgres connection : https://node-postgres.com/