Report for Final Project

Title of the Project: Hotel Reviews on Traveliers.com

Group No - 20

Name of the students: Sourabh Kenjale (kenjale@wisc.edu),
Jordan Hundt (jhundt@wisc.edu)

Drafted by - Sourabh Kenjale
Edited by - Jordan Hundt

I. **Introduction**:
Hotel Reviews on Traveliers.com is a private website where you can search for hotels' data in Europe, see their reviews, get their information, add a review on a hotel that exists in the website database, see their average scores as given by reviewers and search for tags on hotels as given by reviewers. This website suggests data about hotels on the basis of nationality preference. For Ex: Users can search hotels liked by the people of America, Germany, etc. This website is intended to serve as a design model which can be used to search hotelsand see their reviews.

Motivation:
We decided to make a project about hotel reviews for three main important reasons:
1) There is ample data about hotel reviews on sites like kaggle, with our dataset having more than 515K rows (we still decided to use only 30000 for better management and time saving) and more than 10 columns, which satisfies the minimum requirement for the project.
2) Hotel reviewing sites are quite relevant today and everybody has used hotel reviewing websites to get reviews about hotels they would be visiting. Thus, we wanted to create something useful out of it.
3) The dataset was quite clear to understand and it took us less time to create an ER model from the given dataset.

Application:
The Traveliers.com website is created on html, css and php. The database is created in mysql on our local computer and we decided to use php (by XAMPP) because it easily connects to the mysql database and is pretty easy to debug as well. The UI component is completely made by us from scratch. The coding section was the main part of the project which took quite a bit of time for us to complete but we managed to get it done. There were plenty of tutorials available for us to refer to, so it wasn't that difficult to integrate the UI component in the application. The application works quite simple: there is a main home page and there are relevant buttons that take you to the next step: add information in order to get what you desire. For Ex: there is a button called "search reviews". On clicking, you get transported to a new site that asks you for the hotel name and the type of review (good, bad or all kinds) you wish to get the reviews on. On entering valid hotel name and the type, all the relevant data gets displayed. On entering invalid input, the application prompts the user to enter correct input and you can go back to entering the information again. On entering valid input, the UI executes the relevant query and the information in the database is executed accordingly.

Organization and Teamwork:
(Note: Our group initially contained 3 members but near the end of the 5th week, one of our group members (Lordphone Wen) told us he had dropped the course which we had no idea about and we had to recuperate the time lost.)

Our organization goals :

Week	Work
3	Start drafting the project ideas and finalize ER model (same for everyone)
4	Working on the project (Everybody would work on the same section)
5	Working on project
6	Working on Interface + project
7	Working on interface + project + Testing
8	Final Presentation and wrap up

We were mostly up to date with the deadlines we had set at the end of each week, but most of the important work was done in the last 3 weeks of the course when we started coding for the UI and integrated the database with the UI. We had planned to distribute the work evenly among all three. Each of us had to contribute towards finding the data set, creating the ER diagram, filling the dataset and creating the UI component equally - that's what was planned.

Below is the list of the work done to the person involved in completing the work in entirety.

Finding the DataSet: Sourabh Kenjale
Drafting Checkpoint 1: Sourabh Kenjale, with suggestions from Jordan Hundt.
Drafting Checkpoint 2: Sourabh Kenjale with suggestions from Jordan Hundt.
Filling in the data tables: Sourabh Kenjale
Checkpoint 3: Sourabh Kenjale
Checkpoint 4: Sourabh Kenjale
Creating the UI component with all relevant code: Sourabh Kenjale
Testing: Sourabh Kenjale
Drafting the final presentation and report: Sourabh Kenjale

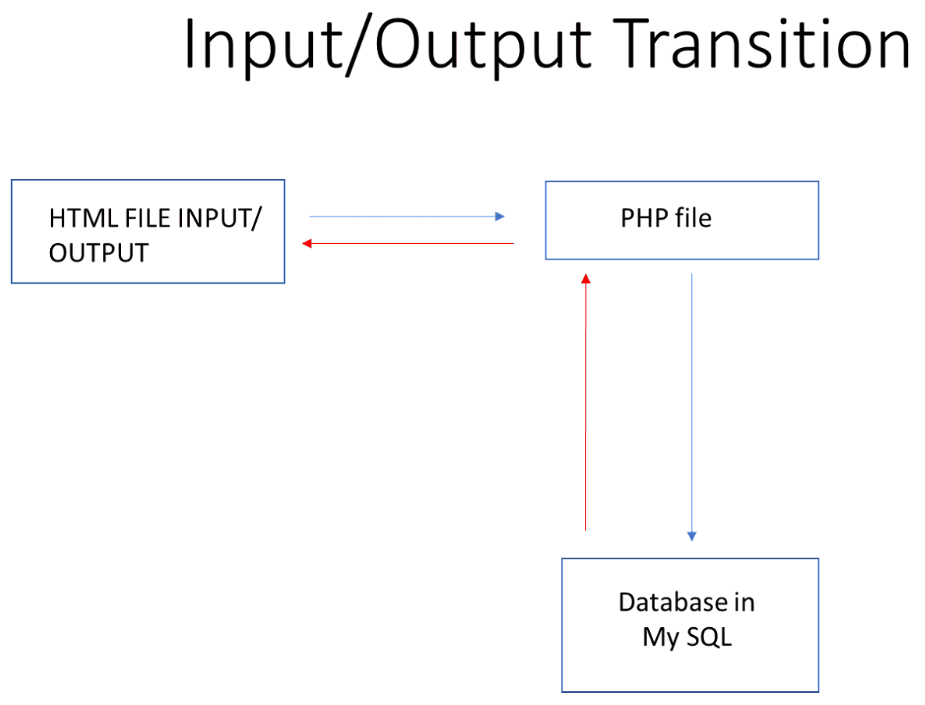
Reasons why our Database project can be considered in tandem with the project expectations:

1) We were only a team of two (one team member left suddenly without informing) with a major part of the project done by only one person, so it definitely delivers considering this one man effort. We showed efficient time management skills and got everything done well before the deadline.
2) We have managed to create a user friendly website that takes in user input and it is very efficient in guiding the user to get what he/she wants.
3) Our database is quite easy to understand with all of our relations either in BCNF or 3NF. Our database design is also concise and reflects well in the front end of the project.
4) We meet the minimum requirements for this project for the number of rows and columns

II. **2.1 System Architecture**:

The dataset (about Hotel reviews) is taken from Kaggle link below:
https://www.kaggle.com/datasets/jiashenliu/515k-hotel-reviews-data-in-europe.
The database tool used in order to input the information is mysql. The UI component is made using XAMPP, HTML,CSS, PHP. The coding part for the UI is entirely done by us. We were considering many other options for making a UI front end for our project but we finally decided to use php because it integrates well with mysql and the external appearance of the website is just like what you would see in a professional hotel reviewing website. We could have used Java/Python but we wanted to create a more user friendly interface so PHP was the best choice for us.
Link - https://www.apachefriends.org

Below is a diagram that shows the flow of the input/output in the project



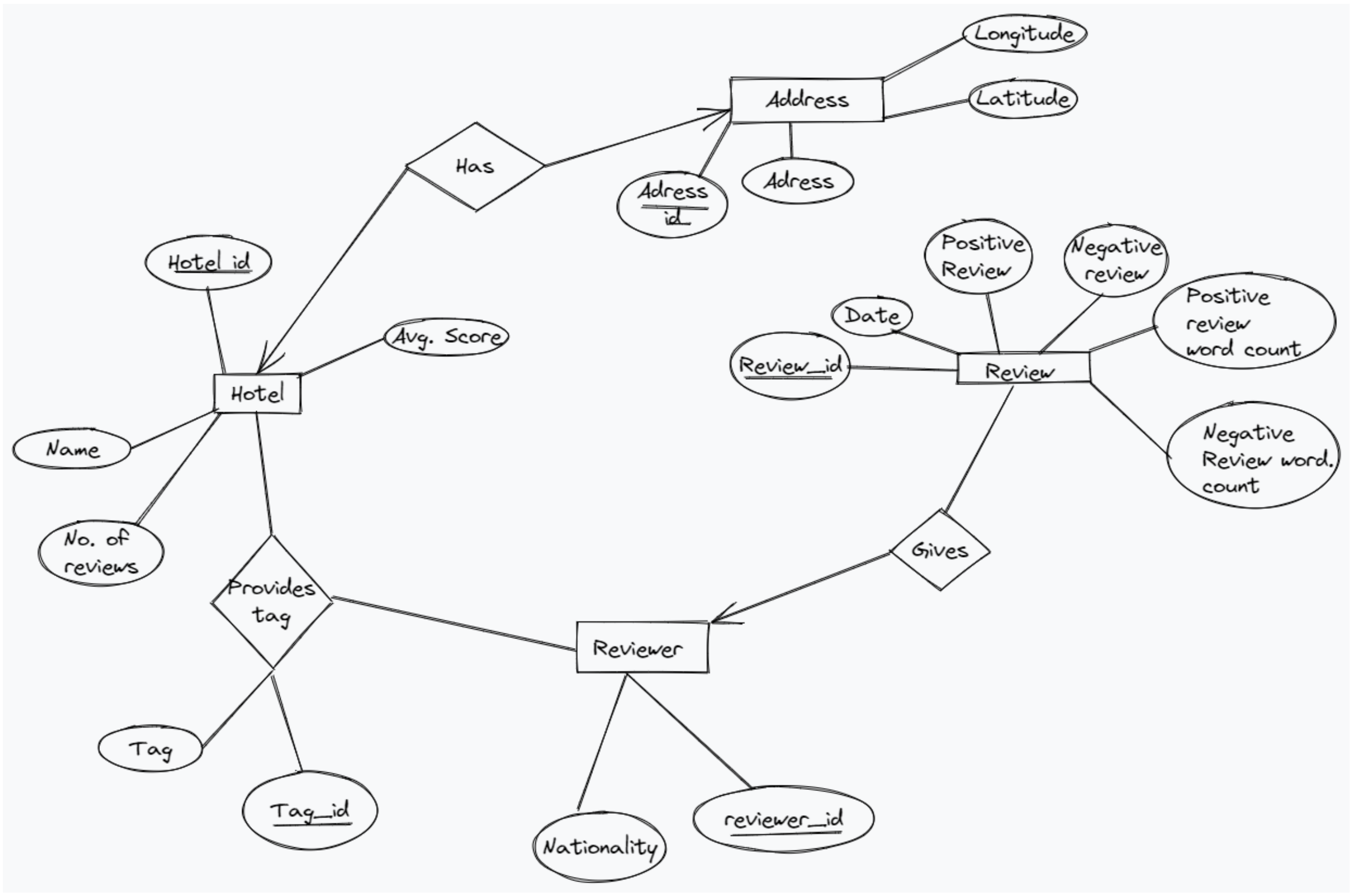
2.2 **Dataset Description**:

The selected dataset has almost 17 attributes including the hotel name, address, latitude, longitude, average score, number of valid reviews, and a tag. We also have the reviewer nationality, total no. of reviews given, the type of review (+ve or -ve), the date of the review, total positive review word count, total negative review word count, and the score.

In our project, we have used 14 out of these 17 attributes.

Used columns – (Hotel_Address, date, avg_score, hotel name, reviewer nationality, neg_review, neg_review word count, total_number of reviews, pos_review, pos_review word count, score, tags, latitude, and longitude).

2.3 **Entity Relationship Diagram**



2.4 **Relational model**

Entity Sets:

Hotel(Name - String, Avg Score - Float, no.of.reviews - Integer, No of Reviews - Integer, hotel_id - Integer)

Address(address: String, address_id: Integer (auto increment), latitude - String, longitude: String)

Reviewer(nationality: String, reviewer_Id : Integer(auto increment)

Review(review_id: Integer (auto increment), date:String, positive_review: String, pos_word_count: Integer, negative_review: String, neg_word_count: Integer, reviewer_id: foreign key from Reviewer, hotel_id: foreign key from Hotel)

Relations:

Has(hotel_id - Integer, address_id: Integer)
Gives(review_id - Integer, reviewer_id: Integer)
Provides Tag(tag_id: Integer)

Relation description :

1) Reviewer and Review : A reviewer gives a review. It is a one to many relation as one reviewer can give many reviews but a review can be given by only one reviewer.
A reviewer in this sense is a person from a nationality.

2) Hotel and Review - A hotel gets reviews. Review has a hotel_id from Hotel acting as a foreign key to better sort data on basis of the hotel_id.

3) Reviewer and Hotel - In the dataset a reviewer has provided a tag to a hotel. There is a many to many relationship. A reviewer can give many tags to a hotel and a hotel can have many tags from different reviewers.

4) Hotel and Address: A hotel has one and only one address. Each hotel has a unique address and thus they share a one to one relationship. Thus, each hotel will have only one address and one address will point to only one hotel.

Database Tables:

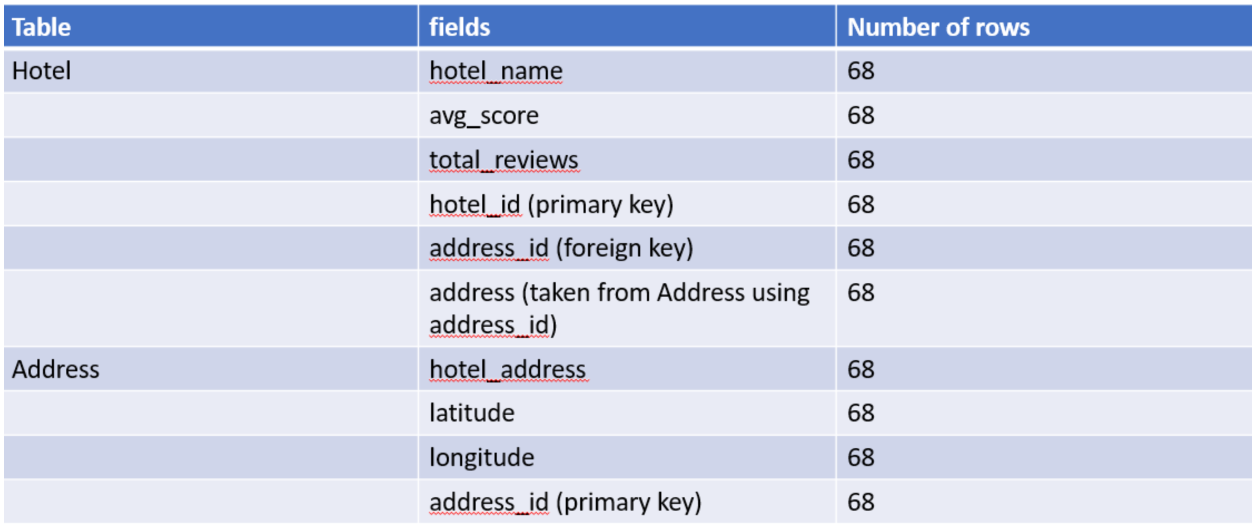


Table	fields	Number of rows
Hotel	hotel_name	68
	avg_score	68
	total_reviews	68
	hotel_id (primary key)	68
	address_id (foreign key)	68
	address (taken from Address using address_id)	68
Address	hotel_address	68
	latitude	68
	longitude	68
	address_id (primary key)	68

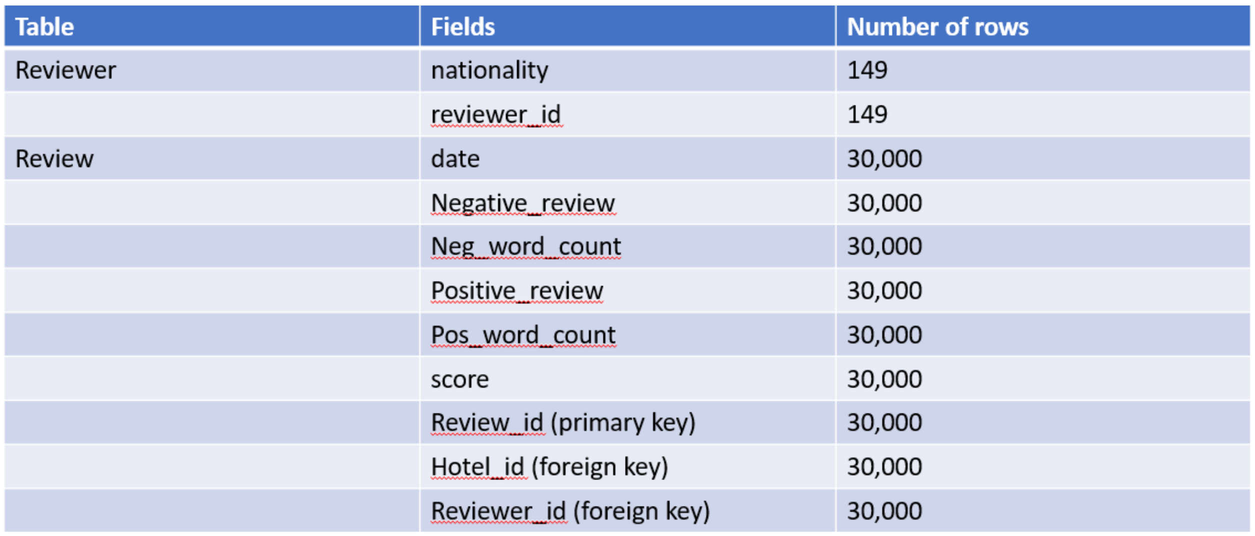


Table	Fields	Number of rows
Reviewer	nationality	149
	reviewer_id	149
Review	date	30,000
	Negative_review	30,000
	Neg_word_count	30,000
	Positive_review	30,000
	Pos_word_count	30,000
	score	30,000
	Review_id (primary key)	30,000
	Hotel_id (foreign key)	30,000
	Reviewer_id (foreign key)	30,000

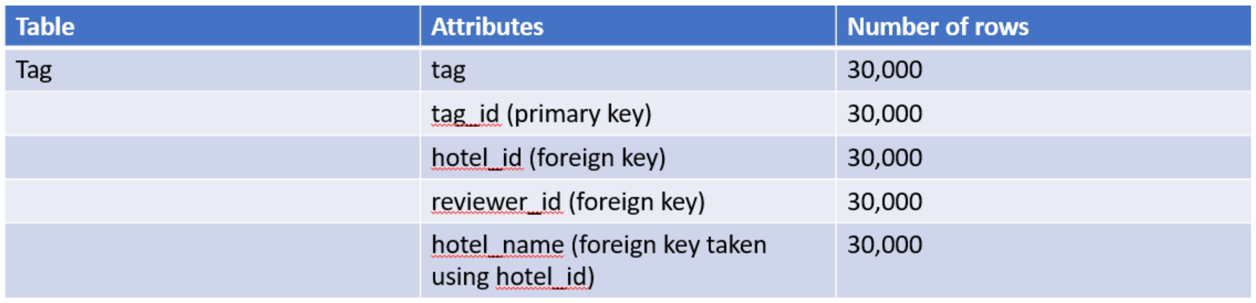


Table	Attributes	Number of rows
Tag	tag	30,000
	tag_id (primary key)	30,000
	hotel_id (foreign key)	30,000
	reviewer_id (foreign key)	30,000
	hotel_name (foreign key taken using hotel_id)	30,000

Functional Dependencies:

- Hotel (hotel_id ->(hotel_name, avg_score, total_reviews, address_id), address_id->address)
- Address(address_id->(address, latitude, longitude))
- Tag(tag_id->(tag, hotel_id, reviewer_id), hotel_id->(hotel_name))
- Reviewer(reviewer_id->(nationality))
- Review(review_id->(date, positive_review, pos_word_count, negative_review, neg_word_count, score, hotel_id, reviewer_id))

As we see from the functional dependencies, all the relations are in BCNF becasue in each, left side is superkey. An id is what can uniquely identify each unique row in every entity, thus it is the super key

2.5 **Implementation and Prototype Description:**

The data from the data set is divided into respective entities. As is visible from the ER diagram, there are 4 main independent entities: hotel, address, Review and Reviewer. In the database, there are different tables for each of these data sets and their relevant attributes as given in the ER diagram.
The Hotel entity is an independent entity representing a unique hotel and the Address entity is also an independent entity representing a unique address for each hotel. The Review entity is also an independent entity representing each unique review given by a reviewer. The Reviewer entity is essentially a list of all unique countries people have given reviews from. The idea is to group reviews on basis of nationality and suggest hotels based on preferences of a particular nationality. Since in our data set there was no name or contact given about a reviewer but only the nationality, we

decided to go with this approach. The tag is something that can exist independently but cannot be classified as an independent entity because it will be accessed by the hotel_id and the reviewer_id. So, we decided to make it a part of a relation. Tags can be sorted on basis of the hotel_id or reviewer_id or both. Thus, users can search for tags given to hotels and can also set a filter on reviews given by the people of a particular nationality.

Troubleshooting and Errors:

=> Before we even began inserting the tables in, we had to make some changes to the ER model because as we learned new concepts in MySQL through the lab, we could make a better ER digram. Our original ER diagram did not have an Address entity and was included in the Hotel entity but then we decided to separate these two and create a new entity called Address. For the relation provides tag, we made many amendments regarding the primary key but then we ultimately decided to make a unique tag id for each new tag.

=> While implementing our database, as we had 515K rows, loading all the data in Excel and sorting it to get unique values was very difficult and extremely time consuming. So in order to tackle this, for the purpose of our project, we decided to reduce the number of rows to 30K and then populated our database accordingly.

=> While coding for the UI part, we decided to use php as it integrate well with MySQL. The major problem was that when I installed XAMPP server for installing php, it had two programs needed to run which were Apache and MySQL. I think I faced a bug as whenever I tried to open my database on the local machine and establish a connection, it never got connected. When I tried to stop MySQL in task manager and restart again, it automatically got killed after 5s. After a whole day of troubleshooting and trying out different combinations, everything finally worked the way it should.

2.6 **Evaluation and Testing**:

The testing portion of this project is done through the UI directly. We have rigorously tested for each component in our project. We checked multiple aspects of the UI interface by entering information and seeing if the UI works as it is supposed to.

For Ex: In the add review component, users can enter a review about a particular hotel they have visited. They also have to specify the hotel name, the type of review (either positive review or negative review) and their nationality. For invalid input, we display a proper error message prompting the user to enter correct information. On entering an invalid hotel_name and then clicking add review, the UI displays to the user that the hotel name does not exist in the database. On entering a too generic name wherein that name could be part of the name of many hotels, the UI tells the user to enter a more specific name. When a review gets inserted successfully, we can check in the database for the last query in the Review entity, and we see that the result is exactly what was entered by the user. Similarly, we can test for search hotels, suggest hotels, find tags, etc by entering valid input and checking the result, and then entering invalid input and checking the error message.

Examples of queries used for the project

// Searching for reviews

```
1) select Review.positive_review, Hotel.hotel_name from Review, Hotel where
   Review.pos_word_count != 0 and Review.hotel_id in (select hotel_id from Hotel
   where Hotel.hotel_name LIKE “%Arena%”) and Review.hotel_id = Hotel.hotel_id;
```

// Executing an insert query; inserting a review

```
2) insert into Review(date, negative_review, neg_word_count,
   positive_review, pos_word_count, score, hotel_id, reviewer_id)
   values (“08/02/2022,”Bad food”,2,”No Positive”,0,5.0,2,4);
```

// The website suggests Hotels based on nationality preference; hotels ordered in descending order of their scores

```
3) select hotel_name, address,avg_score,total_reviews from Hotel
   where Hotel.hotel_id in (select distinct hotel_id from Review
   where Review.pos_word_count < Review.neg_word_count and
   Review.reviewer_id = (select Reviewer.reviewer_id from
   Reviewer where Reviewer.nationality LIKE "%China%")) ORDER BY
   Hotel.avg_score DESC, Hotel.total_reviews DESC;
```

// Searching for tags associated with specific hotels

```
4) select tag,hotel_name from Tag where Tag.hotel_id in (select
   Hotel.hotel_id from Hotel where Hotel.hotel_name LIKE "%London
   Paddington%");
```

// Searching for hotels based on address

```
5) select hotel_name,avg_score,total_reviews,address from Hotel
   where hotel_name LIKE "%K K%";
```

// Getting hotel information

```
6) select * from Hotel where Hotel.address_id in (select
   Address.address_id from Address where Address.hotel_address
   LIKE "%Paris%");
```

III. **Conclusion**:

This project was an interesting learning experience for me as this is the first time I have made a project about databases and made a UI component that interacts with the database. The most interesting part was making the ER model, drafting into the relational model and creating the UI component to take user input. Searching information through queries was also quite efficient and getting the information was very easy. I feel that the most uninteresting part of the project was writing the code for the UI. This is the first time I used PHP and HTML together with MYSQL to make the UI part of the project but I got to learn a lot while going through online tutorials and getting things ready. Database concepts like normalization and the connection between different entities helped me organize my database and get the required information.

Talking about the future outcomes for this project, we can definitely see a use of this project in behavioural analysis of customers. We can collect data of people from specific nationalities and figure out a general curve of how these people behave and what are their specific preferences. This would surely help in the hospitality industry as many hotels wants designs and infrastructure that would attract the most number of customers. A majority of the hotel reviews websites do not explicitly ask for survey type data but I believe collecting this information is very useful in research.

Hotel review websites are quite commonplace but we want to take it an edge further. We want to make this website bigger and much better than what it is at its primordial stage so that people can take advantage of the hotel reviews and we can get research related data.

IV. **References**:

https://www.w3schools.com/php/ - PHP tutorial
https://www.tutorialspoint.com/php/index.htm - PHP tutorial