

TRAVEL JOURNAL

SRAVANI KOPPALA (Team Leader)

SNEHAL GAIKWAD

VICHETNA GADDAM

*Prepared in partial fulfilment of
Database Systems Course (CS687)*

Under the Guidance of

*Dr. Ramazan Aygun
Associate Professor, Computer Science Department
The University of Alabama in Huntsville.*

ACKNOWLEDGEMENT

Firstly, we would like to thank the Department of Computer Science, The University of Alabama in Huntsville for giving us this opportunity of implementing and gaining practical knowledge as a part of the Data Base Systems course in our curriculum.

We would like to express our sincere thanks to Dr. Ramazan Aygun, Department of Computer Science, The University of Alabama in Huntsville for offering us a project in the field of our interest and guiding us. We thank you Professor for giving us the opportunity to carry out a project in this esteemed institution.

Table of Contents

Section 1. Informal Description of the Database Application	4
Section 2. System Requirements	5
Section 3. Project Plan	6
Section 4. Responsibilities of each group member	7
Section 5. Formal description of the database	8
Section 5.1. Components of the conceptual data model.	
Section 5.1.1. Strong Entities.	
Section 5.1.2. Weak Entities.	
Section 5.1.3. Relationships	
Section 6. ER or EER diagram	11
Section 7. Mapping ER/EER to relational or object database schema	13
Section 8. Relational or Object Database Schema	16
Section 8.1. "Create TABLE" statements	
Section 8.2. Tables in Tabular format	
Section 9. Data Acquisition	20
Section 10. Description of Implementation	21
Section 11. Issues	22
Section 11.1. General Issues	
Section 11.2 Specific Issues	
Section 12. User Interface	24
Section 13. Bonus	33
Section 14. Conclusion & Future Development	35
Section 15. Appendix	
Section 15.1 Meeting Minutes	36
Section 15.2 References	41

1. DESCRIPTION

Travel Journal is a web application which is useful to save memories in the form of a journal especially of your vacations. It also has other features like saving a diary entry for the days you would like to write memories of a particular day in the personal diary page. Also, you can create a To-Do list of your activities that you wish to have a track of!

In order to start using the application, a user should first register and login to utilize the features of this application. A Registered user can post Trips, Diary Entries and add a task in To Do List as well. To meet these requirements, we maintain a few important tables like Users table, Trips table, Attractions table, Personal Diary table and a To Do List table.

Our Project can be divided into 3 parts for explanation:

1. Trips: A registered user can post a trip s/he's visited. The trip record can include a photo of the place, with a name of the place, City, Country, type of that place, date, his/her best moment/memory. After adding trips, s/he can view a list of all his trips where s/he can edit or delete the trips as needed.
2. Personal Diary: A registered user can create a diary entry in his/her profile. S/he can add a date and write the story of that day in the text field. S/he can view these entries once added and edit or delete them.
3. To Do List: A registered user can add a task in his/her profile. S/he can add a Date and Task name. S/he can view the list of tasks that were added and can edit or delete them.

2. SYSTEM REQUIREMENTS

1. **System Environment:** Web application
2. **Operating System:** Windows 10, 64-bit
3. **Programming Environment and Languages:**
 - a. Programming language: PHP (Back-end)
 - b. Web designing: Html, Bootstrap
 - c. Database: MYSQL
 - d. Server: Wamp Server.

3. PROJECT PLAN

1. Topic submission with the names of group members
Expected end date: January 27th, 2020
Actual date of submission: January 27th, 2020
2. Description of database including a project plan
Expected end date: February 2nd, 2020
Actual date of submission: February 10th, 2020
3. Data Acquisition
Expected end date: February 19th, 2020
Actual end date: March 11th, 2020
4. Formal description of Database
Expected end date: March 11th, 2020
Actual end date: March 18th, 2020
5. Develop functionalities(interfaces) and DB
Actual end date: April 4th,2020
6. Enhance or Test and checking of all functions of application
Actual end date: April 8th, 2020
7. Report
Expected end date: April 15th, 2020
Actual date of submission: April 15th, 2020
8. Demo
Expected end date: April 15th, 2020

4. RESPONSIBILITIES OF EACH GROUP MEMBER

Responsibilities shared	Name of individuals
Data Acquisition	Snehal Gaikwad, Vichetna Gaddam
Creation and maintenance of database	Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam
ER and Schema Diagrams	Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam
Designing the web interface and Database Connection to Front-end	Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam
Application Testing	Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam
Project Report	Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam
Tracking of Minutes of Meeting	Vichetna Gaddam

5. FORMAL DESCRIPTION OF DATABASE

Our project Travel Journal has 5 entities. Their attributes are described as below:

5.1 Components of conceptual data model:

5.1.1 Strong Entities:

1. USER:

- **User_id:** It is a unique id for each User of INT data type, and it is the primary key which is auto incremented.
- **Fname:** It is the first name for each User of VARCHAR data type.
- **Lname:** It is the last name for each User of VARCHAR data type.
- **Gender:** It is the gender of each User and it is of CHAR data type.
- **DOB:** It is the birth date of the User and it is of DATE data type.
- **Email:** It is the email-id of the User and it is of VARCHAR data type and is unique.
- **Password:** Password of the User for corresponding email id and it is of VARCHAR data type.
- **Address:** It is the address of the User and it is of VARCHAR data type.
- **Zipcode:** It is the zip code of the address given by User and it is of INT data type.

2. TRIPS:

- **Trip_id:** It is a unique id for each Trip of INT datatype, and it is the Primary Key which is auto incremented.
- **Photo:** It is the image of the place the user visited, and it is of LONGTEXT datatype. This can be null.
- **Story:** It is the best moment, or a memory of the trip and it is of VARCHAR datatype.
- **Date:** The date of the visited trip and it is of DATE datatype.

3. ATTRACTIONS:

- **Attraction_id:** It is the unique id of the attraction and it is of INT data type which is also auto incremented.

- Place name: It is the name of place and it is of VARCHAR data type.
- City: It is the name of the city and it is of VARCHAR data type.
- Country: It is the name of the country and it is of VARCHAR data type.
- Location_Type: It is the type of the location and it is of VARCHAR data type.

4. PERSONAL_DIARY:

- Entry_id: It is a unique id for each diary entry, and it is of INT data type and it is the Primary Key which is also auto incremented.
- Date: It is the date when the entry for the diary is made and is of DATE data type.
- Diary_Entry: It is the text entry and is of TEXT data type.

5. TO_DO_LIST:

- Task_id: It is unique id for each task in the list, and it is of INT data type and it is the Primary Key which is also auto incremented.
- Date: It is the date when the entry for the diary is made and is of DATE data type.
- Task_Name: It is the name of the task and it is of TEXT data type.

5.1.2 Weak Entities:

There are no weak entities in this application.

5.1.2 Relationships:

1. Creates relationship is used to bind the User table and the To_do_list table. Here, the cardinality is 1 to n i.e. One User can create n number of tasks in a To_do_list. The participation is partial, as it is not mandatory for users to create or have a To_do_list.



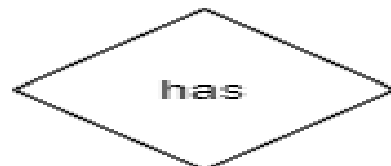
2. Adds relationship is used to bind the User table and the Personal_diary table. Here, the cardinality is 1 to n i.e. One User can add n number of entries in a Personal Diary. The participation is partial, as it is not mandatory for users to add or have a Personal Diary.



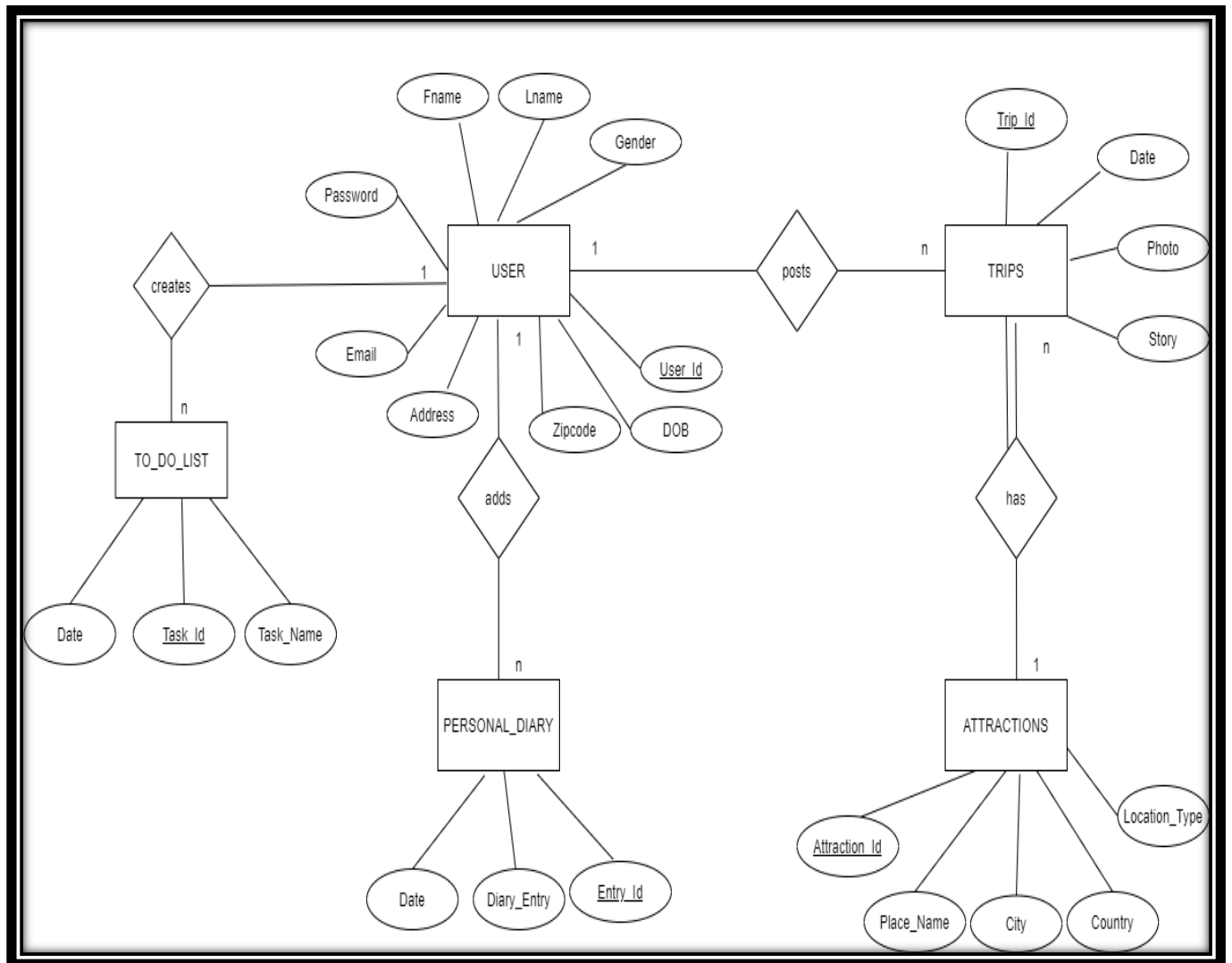
3. Posts relationship is used to bind the User table and the Trips table. Here, the cardinality is 1 to n i.e. One User can post n number of Trips. The participation is partial, as it is not mandatory for users to post or have a Trip.



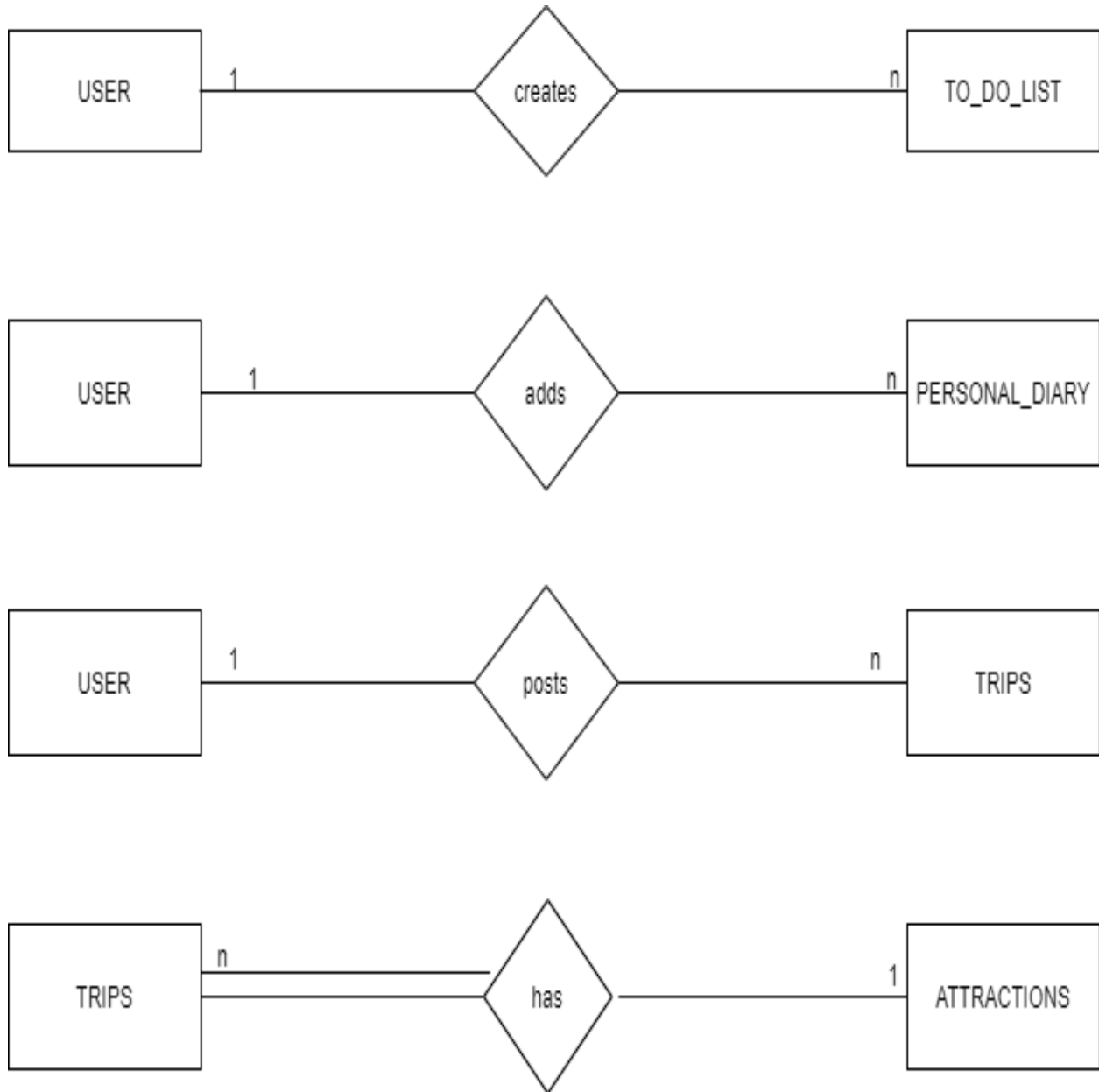
4. Has relationship is used to bind the Trips table and the Attraction table. Here, the cardinality is n to 1 i.e. One Attraction can have n number of Trips. The participation is total, as it is mandatory for trips to have at least one Attraction.



6. ER DIAGRAM



7. MAPPING ER TO RELATIONAL SCHEMA

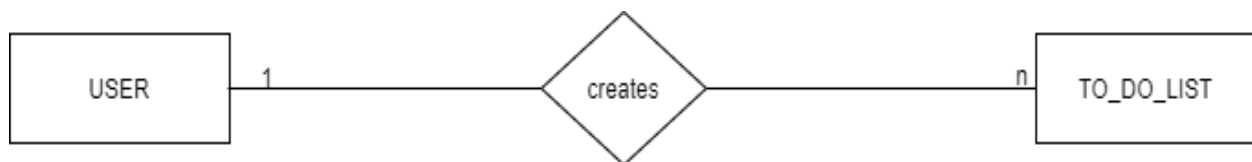


One To Many (1 To N) Relationship Mapping Into Tables:

1. As per 1 to N mapping, include the user's primary key into the To_do_list table as foreign key i.e, User_id is the foreign key in the To Do table. One User can create or post any number of To Do Lists. Hence, the relations can be stated as

User: (User_id, Password, Fname, Lname, Email, Gender, DOB, Address, Zipcode)

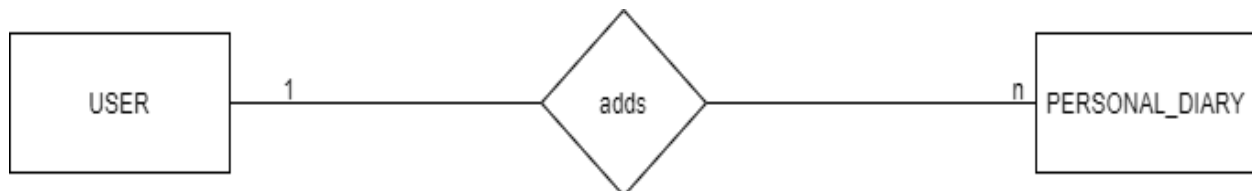
To_Do_List: (Task_id, Date, Task_Name, User_id)



2. As per 1 to N mapping, include user's primary key into personal_diary table as foreign key i.e, user_id is the foreign key in Personal_diary table. One User can create or post any number of Diaries. Hence the relations can be stated as

User: (User_id, Password, Fname, Lname, Email, Gender, DOB, Address, Zip-code)

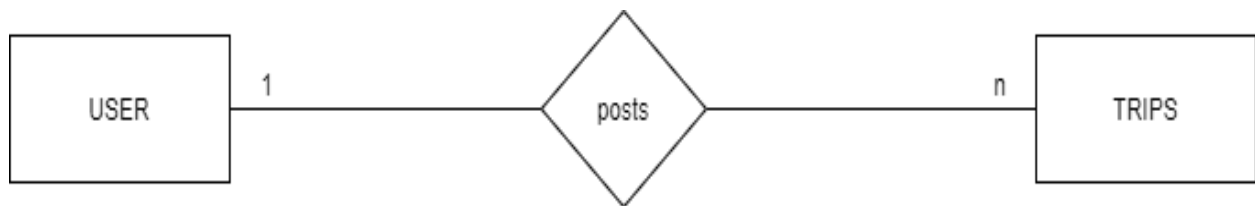
Personal_diary: (Entry_id, Date, Diary_Entry, User_id)



3. As per 1 to N mapping, include User's primary key into Trips table as foreign key i.e, user_id is the foreign key in trips table. One User can create or post any number of Trips. Hence the relations can be stated as

User: (User_id, Password, Fname, Lname, Email, Gender, DOB, Address, Zip-code)

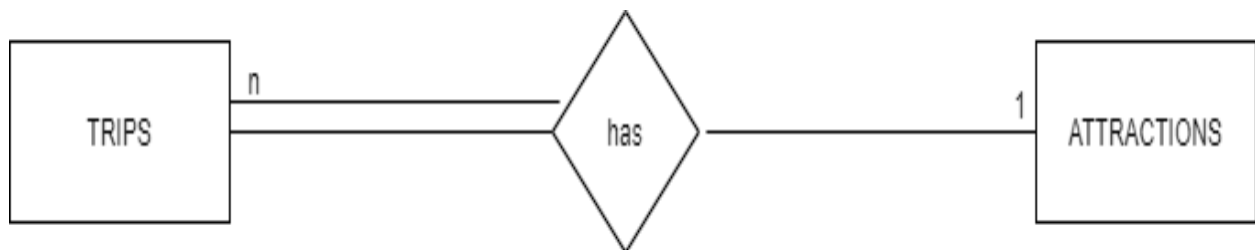
Trips: (Trip_id, Photo, Date, Memory, User_id)



4. As per 1 to N mapping, include Attractions primary key into trips table as foreign key i.e., Attraction_id is the foreign key in trips table. Hence the relations can be stated as

Trips: (Trip_id, Date, Photo, Memory, User_id, Attraction_id)

Attractions: (Attraction_id, place_name, City, Country, Location_type)



Entity Relationship (ER) Design Explanation:

The ER diagram in the above section clearly outlines the entities and their attributes. Total participations and partial participations have also been shown in the figure. The primary keys of each entity have been underlined. Cardinalities have been mentioned using their standard form (1-N).

ER (Entity Relation) diagram of our project helps to analyze the logical structure of database. Initially we have USER entity with USER_ID as primary key which can maintain and perform operations on this application like:

- Creating new trips, personal diaries and To Do lists.
- Updating/Editing the already existing trips, personal diaries and To Do Lists.
- Deleting entries of trips, Diaries and To Do lists.

USER entity has the information related to the user with attributes User_id(Unique), Password, Fname, Lname, Gender, Email, DOB, Address, Zip code. This entity is bound to TRIPS, PERSONAL_DIARY and TO_DO_LIST entities with POSTS, ADDs, CREATES relationship respectively which shows that the User can post Trips, add Diary entries and create tasks in To Do lists on his profile. TRIPS entity has details about the user's trips with attributes Trip_id(Unique), Date, Photo, Story. PERSONAL_DIARY has Entry_id(Unique), Date, Diary_Entry as its main attributes. To_do_list has Task_id(Unique), Date, Task_Name as its main attributes.

TRIPS entity is bound to ATTRACTIONS entity with HAS relationship which shows that Trips Contain Attractions of a place having attraction_id in the trips table as foreign key. Attraction entity contains attributes like Attraction_id, City, Country, Place_Name, Location_Type.










8. RELATIONAL DATABASE SCHEMA

8.1 Create Table Statements:

1. USERS

```
CREATE TABLE users (  
  user_id int(11) NOT NULL AUTO_INCREMENT,  
  password varchar(255) NOT NULL,  
  Fname varchar(15) NOT NULL,  
  Lname varchar(15) NOT NULL,  
  Sex char(1) NOT NULL,  
  Email varchar(30) NOT NULL UNIQUE,  
  DOB date NOT NULL,  
  Address varchar(40) DEFAULT NULL,  
  Zipcode int(5) DEFAULT NULL,  
  PRIMARY KEY (user_id),  
);
```





Alter TABLE USERS ADD CHECK (sex IN ('M', 'F'));

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 user_id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 password	VARCHAR(255)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Fname	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Lname	VARCHAR(15)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Sex	CHAR(1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Email	VARCHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 DOB	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Address	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 Zipcode	INT(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

2. PERSONAL_DIARY





```
CREATE TABLE personal_diary (  
  entry_id int(5) NOT NULL AUTO_INCREMENT,  
  date date NOT NULL,  
  diary_entry text NOT NULL,  
  userid int(11) NOT NULL,
```


PRIMARY KEY (entry_id),
 FOREIGN KEY (userid) REFERENCES USERS(user_id) ON DELETE
 SET NULL ON UPDATE CASCADE);

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 entry_id	INT(5)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 diary_entry	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 userid	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	






3. TO_DO_LIST

CREATE TABLE to_do_list (
 task_id int(10) NOT NULL AUTO_INCREMENT,
 user_id int(11) DEFAULT NULL,
 date date NOT NULL,
 task_name text NOT NULL,
 PRIMARY KEY (task_id),
 FOREIGN KEY (user_id) REFERENCES USERS(user_id) ON DELETE
 SET NULL ON UPDATE CASCADE);

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 task_id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 user_id	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 task_name	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	







4. ATTRACTIONS

CREATE TABLE attractions (
 attraction_id int(11) NOT NULL AUTO_INCREMENT,
 place_name varchar(40) NOT NULL,
 city varchar(11) NOT NULL,
 country varchar(11) NOT NULL,
 location_type varchar(20) NOT NULL,
 PRIMARY KEY (attraction_id));

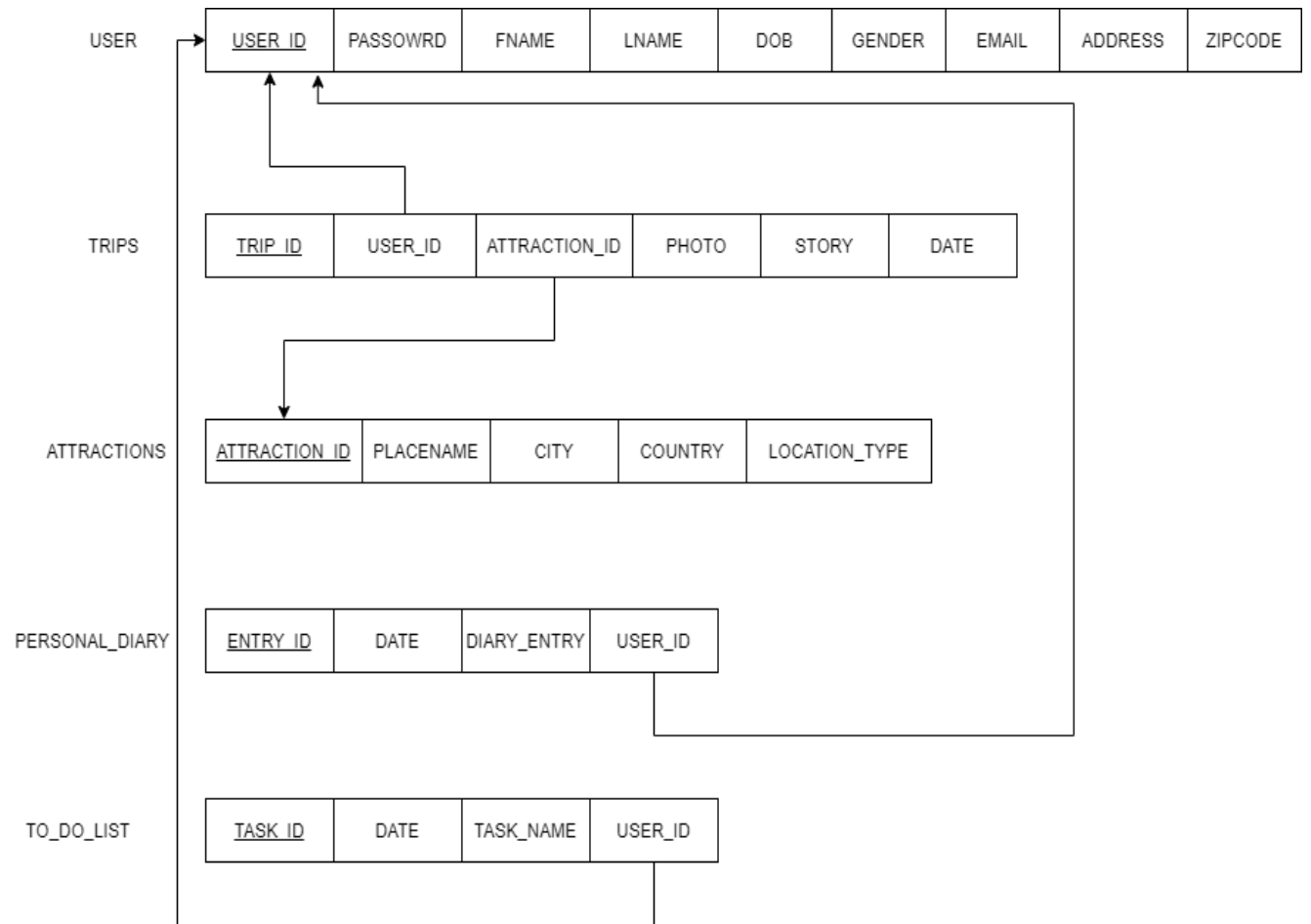
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 attraction_id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 place_name	VARCHAR(40)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 city	VARCHAR(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 country	VARCHAR(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 location_type	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

5. TRIPS

```
CREATE TABLE IF NOT EXISTS trips (
  trip_id int(11) NOT NULL AUTO_INCREMENT,
  user_id int(11) NOT NULL,
  attraction_id int(11) NOT NULL,
  date date NOT NULL,
  story text NOT NULL,
  photo longtext,
  PRIMARY KEY (trip_id),
  FOREIGN KEY (userid) REFERENCES USERS(user_id) ON DELETE
  SET NULL ON UPDATE CASCADE,
  FOREIGN KEY (attraction_id) REFERENCES ATTRACTIONS(user_id)
  ON DELETE SET NULL ON UPDATE CASCADE );
```

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 trip_id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 user_id	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 attraction_id	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 story	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 photo	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

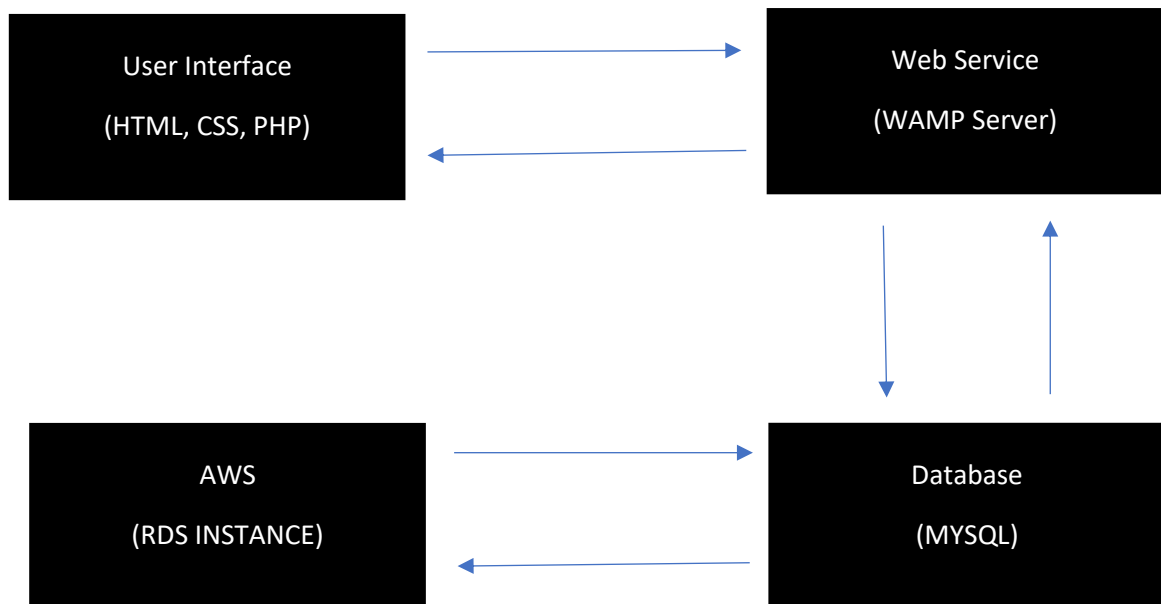
8.2 Relational Schema Diagram:



9. DATA ACQUISTION

1. We generated around 1000 records for the USERS table from a website with URL “<http://www.generatedata.com/>”. This website generates random data. We downloaded the data in csv form and made changes to make it compatible with table in our databases then we populated data into table using import csv option.
2. We got around 4600 records for the ATTRACTIONS table from a website named [Kaggle](https://www.kaggle.com/vitaliymalcev/russian-touris-attractions)(<https://www.kaggle.com/vitaliymalcev/russian-touris-attractions>).

10. DESCRIPTION OF IMPLEMENTATION



Roles of Each Components:

The user interface uses the basic HTML(integrated with Bootstrap and Font Awesome for CSS) to interact with the user. The user logs in and selects the required action (namely trips or personal diary or to do list). Once the user clicks any button or an action is submitted (Ex: Submit), the web service identifies the request and runs the PHP code. The PHP code runs the respective insert/update/delete statements on the database. Here, the Database server being used is MYSQL workbench which returns the requested results to the web service and web service (WAMP) in turn sends to the client. Also, the SQL Work Bench is connected to the AWS RDS instance using an end point given to the AWS Educate Account. So, this can be used to view the database hosted on AWS.

11. ISSUES CONSIDERED

11.1 General Issues:

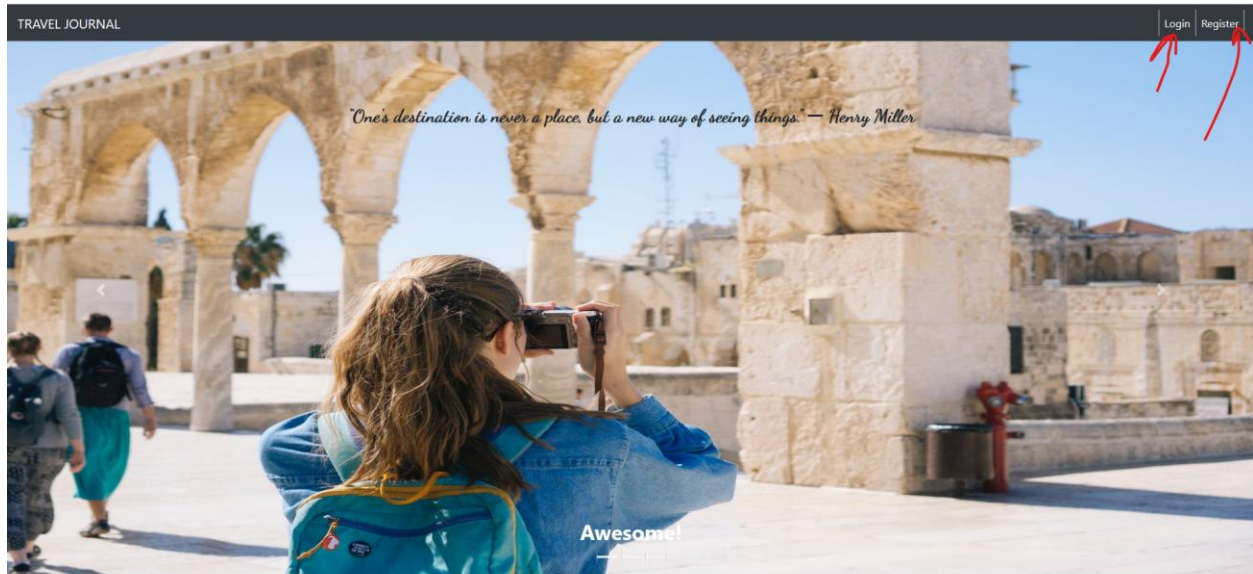
1. Is the application a single-user or multi-user application?
Answer: This is a multi-user application.
2. Is the application read-only (only retrieval) or read-write application?
Answer: It is a read-write application because user can insert, edit and delete records in the database from the user-interface itself.
3. How is concurrency handled?
Answer: Yes, concurrency is handled using sessions.
4. Is indexing or fast retrieval an important part of your application?
Answer: No. Not related to indexing but, we have seen that compared to the SQL database in WAMP, AWS database works fast.
5. What are the steps taken if the system or a transaction fails?
Answer: If the system fails, the user will be notified about the same and that the database is not able to connect.
6. Is the mobility of application a good feature? How about deploying front end app on an Android, iOS, or Windows 10 system? Consider whether the database will be stored at the client or at a server.
Answer: Yes. As of now, this application can be deployed on any of the windows platforms and in future can be converted to a mobile application.
7. Is your application data stored on a cloud database?
Answer: Yes, we are using AWS RDS instance.
8. What is the size of your database? Are you able to populate your database with realistic data?
Answer: AWS allocates around 100TB of storage and currently the DB has used 0.2TB. Yes, we can populate realistic data.
9. Do you get proper use of available APIs or SDKs made available by the companies?
Answer: We have used Bootstrap and Fontawesome APIs for styling.

11.2 Specific issues:

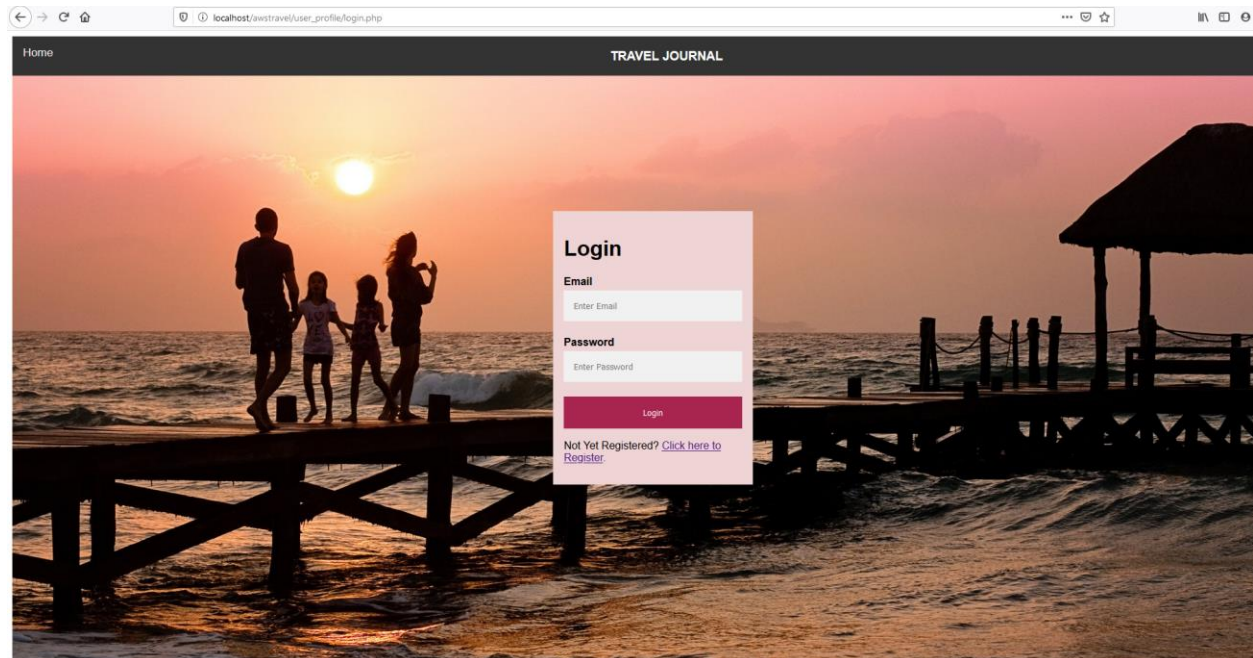
1. All the html tags and PHP tags were using `_GET` method. While adding an image, we didn't understand why the method was not running and was not going inside the if statement. Then, **Vichetna** suggested that instead we can use a `_POST` method and hence converted the html method to post and changed all the PHP tags as well. This magically worked and saved our lives!
2. When the SQL database, which was local to WAMP, was moved to cloud in AWS, **Sravani** was able to connect on her laptop but **Snehal** and **Vichetna** were not able to connect. **Sravani**, then figured that access was not given publicly to the instance and hence changed the same.
3. While working on either the update page or delete page for any of the tables, we needed the unique ID which has to be sent to the update/delete page from the current web page. **Snehal** is good with PHP and she suggested that we can use echo and send the ID in the href tag of html and retrieve it using `_GET` method in the next page.
4. We found it hard to make many to many relationships on attractions and trips table, so we all agreed upon making one to many relationship by creating a foreign key of `attraction_id` in the trips table and resolved the issue.

12. USER-INTERFACE SNAPSHOTS

The Home Page:



The Login Page:



The registration page:

← → ↻ 🏠 | 📍 localhost/awstravel/user_profile/Registration1.php | ⋮ 🛡️ ☆

Home TRAVEL JOURNAL

Create Your Account

First Name
Your name..

Last Name
Your last name..

Gender ☐ Male ☐ Female

E-Mail
Your email-id..

Date Of Birth
mm / dd / yyyy

Enter password
Your password

Confirm password
Re-type password

Address
Enter your address

Zipcode
Enter your zipcode

SUBMIT


User's Home Page:

← → ↻ 🏠 | 📍 localhost/awstravel/homepageAfterLogin.php | ⋮ 🛡️ ☆ | 📄 📅 📌 ☰

Home Trips Diary To-Do-List | Update Profile Logout


Welcome to Online Journal Sravani!

Travel




Add a Travel Post!

Diary



Add a Diary Entry!

To-Do



Add a To-Do Task!

Add a Trip Page:

localhost/awstravel/tripjournal/insert_travel_entry.php

Home Trips Diary To-Do-List Update Profile Logout

Create a Trip entry

Type of Location: Museum, Beach etc.

mm / dd / yyyy

Name of the place

City

Country

No file selected.

Write your story..!

View Trips Page:


localhost/awstravel/tripjournal/view_trips.php

Home Trips Diary To-Do-List Update Profile Logout

Travel Journal Entries


Trip to Hyderabad, India.
Restaurant: Bawarchi.
2020-04-06
Awesome Biryani with Double Masala.

Trip to Destin, Florida.
Beach: Boardwalk.



2020-03-30
White sand beach.

Trip to Hyderabad, India.
Restaurant: Bawarchi.



Update a Trip Page:

Update Trip

Restaurant

04/06/2020

Bawarchi

Hyderabad

India

Awesome Biryani with Double Masala.

Update Trip

Add a Diary Entry Page:

Create a new entry in your Personal Diary!

mm/dd/yyyy

Enter your Diary notes.

Add Task

View Diary Entries Page:

My Daily Journal Entries

Today was a big day for me. I had a presentation in my Big Data class and it went very well. I wanted to perform better than before because I blanked out for a minute in the last one

2020-04-01

+ Add Diary Entry!

Update a Diary Entry Page:

Updating Diary Entry

localhost/awstravel/PersonalDiary/update_diary_entry.php?id=5

Home Trips Diary To-Do List

Update Profile Logout

Update entry in your Personal Diary!

04 / 01 / 2020

Today was a big day for me. I had a presentation in my Big Data class and it went very well. I wanted to perform better than before because I blanked out for a minute in the last one.

Update

Add a Task Page:

Navigation Bar

localhost/awstravel/todoList/insert_tasks_entry.php

Home Trips Diary To-Do List

Update Profile Logout

Create a new entry in your To-Do List!

mm / dd / yyyy

Enter your Task...

Add Task

View To-Do-List Page:

navigation bar

localhost/awstravel/todoList/To-Do-List.php

Home Trips Diary To-Do List

Update Profile Logout

My To-Do List!

✓ Do Laundry 2020-04-01

Edit Delete

+ Add Task!

Update a Task Page:

localhost/awstravel/todoList/update_task.php?id=7

Home Trips Diary To-Do List

Update Profile Logout

Update To-Do Task

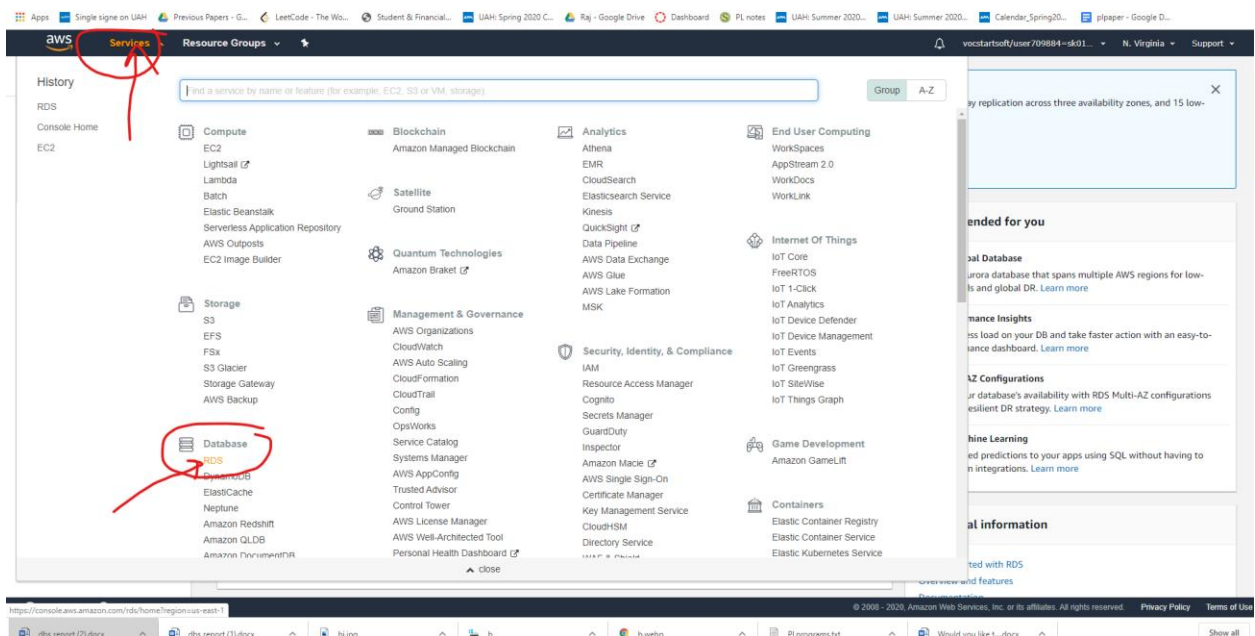
04 / 01 / 2020

Do Laundry

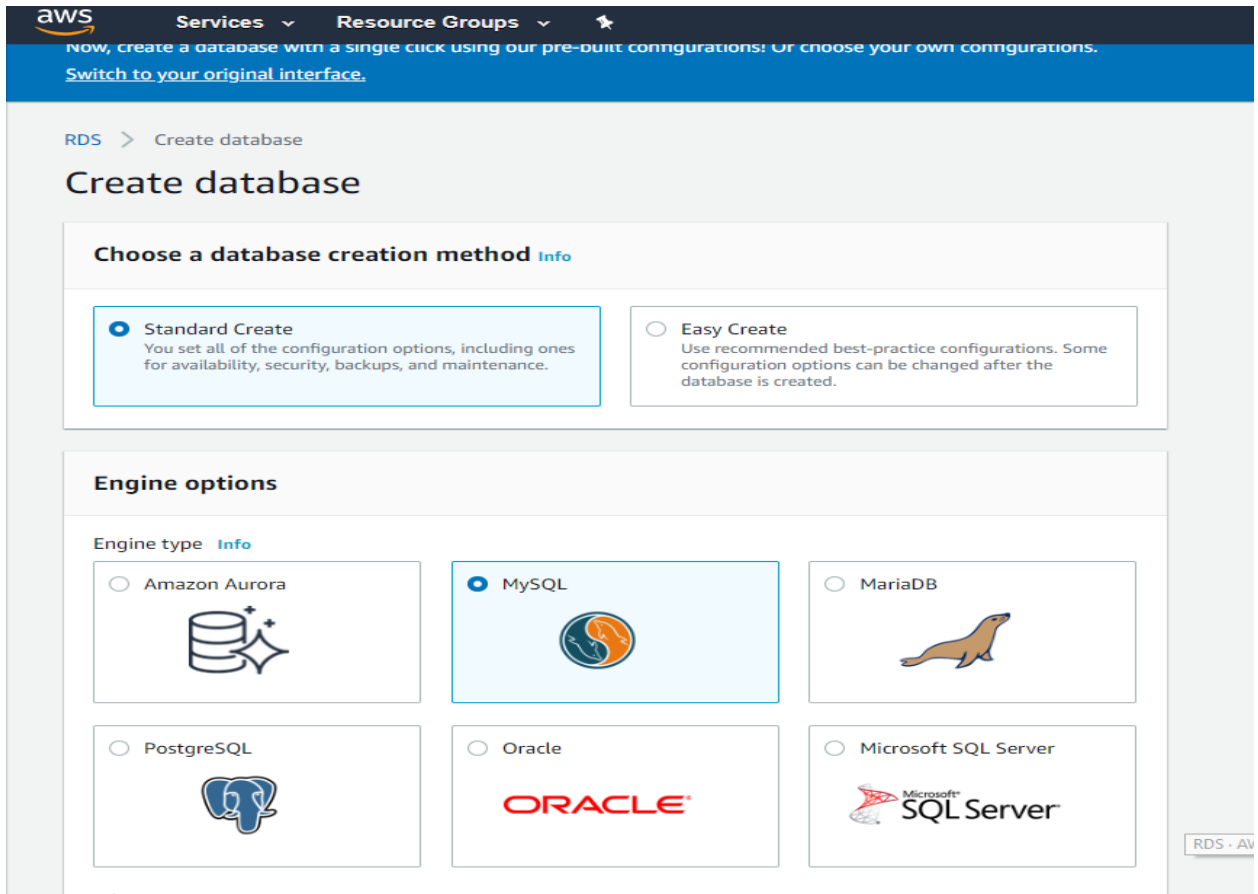
Update Task

13. SECTION BONUS

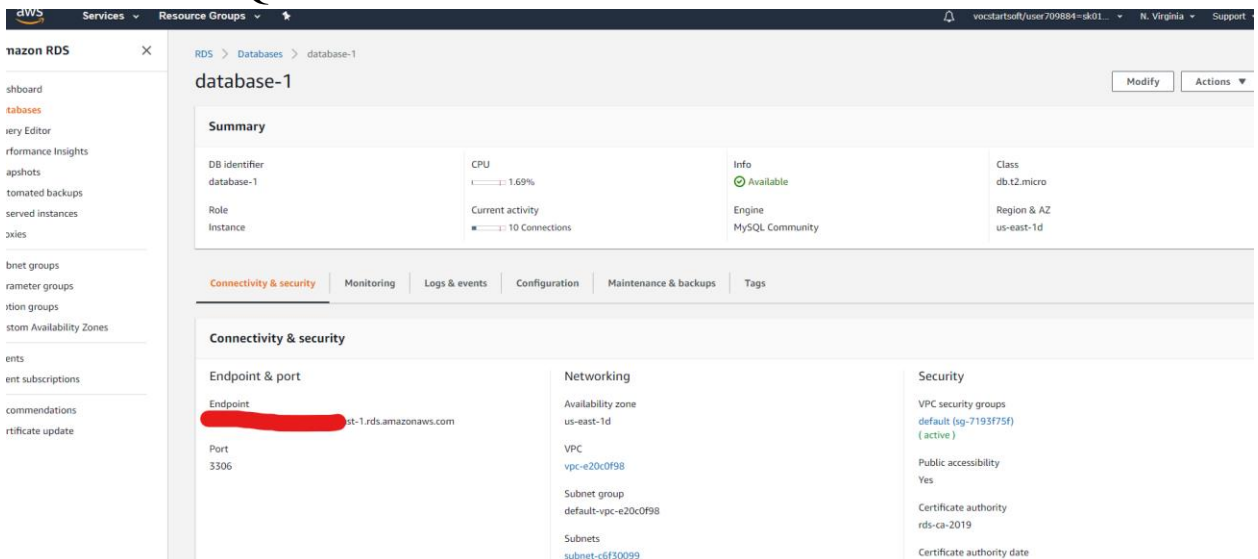
- We have done our bonus part by moving our database to AWS cloud and currently our DB instance is hosted on cloud and can be accessed by any number of users at the same time.
- Amazon Relational Database Service (or Amazon RDS) is a distributed relational database service by Amazon Web Services (AWS). It is a web service running in the cloud.
- Following are the steps to create the same. We have created a DB instance of MYSQL, after logging in with UAH email, in the Services → Database section → RDS instance of the AWS.



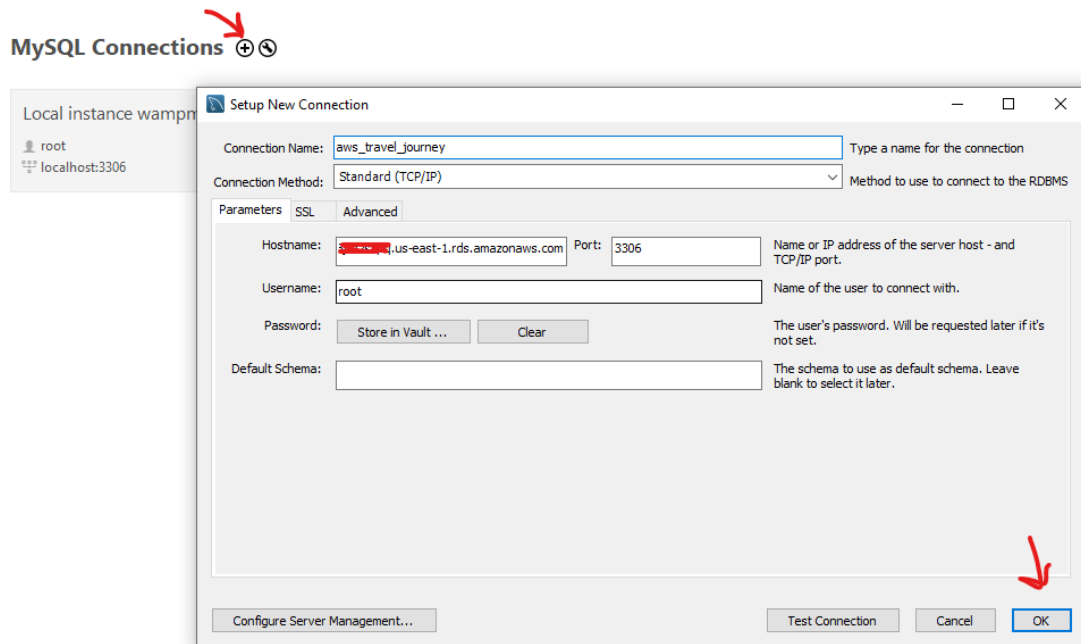
- Select Create Database and MySQL server, select all the default fields and click on Create instance.



- The following end-point, username, password has to be stored and can be used in the SQL workbench to connect and view the database.



- Make a new connection in the SQL work bench, enter the endpoint in the hostname field, give a name to the connection, enter username and password in the “store in vault” button and click on OK.



- Now, after connecting use the exported SQL file from WAMP and use the same as script to import the whole DB to AWS and start using.

14. CONCLUSION & FUTURE DEVELOPMENT

This application allows a User to post, edit and delete his Trips, Personal Diary and To Do List. User can view his list of trips, Diaries and Task lists.

In future, we can add many things to our application like:

1. Adding an option of video and audio clips. SQL may not support this, and we can move to Mongo DB for that.
2. Adding maps instead of city and country names.
3. Sharing memories with friends(other users).
4. Tagging Friends in posts.
5. Going live from a location and posting it on their profile.
6. Commenting on a friend's post if tagged.

15. APPENDIX

15.1 MEETING MINUTES

Meeting 1:

Date & Time: January 22, 2020: 12:45 pm

Location: Tech Hall

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 30 minutes

Topic: Topic Discussion.

Detailed Information: Vichetna and Sravani decided Snehal can also join our team and we started discussing different ideas like Travel Journal Application, Online Shopping, Social Networking site etc. Decided to do the bonus part.

Meeting 2 :

Date & Time: January 27, 2020: 12:45 pm

Location: Tech Hall

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 45 minutes

Topic: Detailed discussion about selection of the project.

Detailed Information: We narrowed down our topic to Travel journal and decided that we would use either use NoSQL or MongoDB as the bonus part for the backend and we submitted project name and names of group members to Dr.Aygun. Front-end, still thinking about React JS. Need to explore these topics and update them in the next meeting.

Meeting 3:

Date & Time: February 6, 2020: 11 am

Location: Tech Hall

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 20 minutes

Topic: Discussing Front end and back end

Detailed Information: Discussed the ideas the travel journal would be having. We fixed on getting a Bonus and decided on using the Mongo DB. For the Web Interface, we decided to do React JS.

Meeting 4:

Date & Time: February 10th, 10:30AM

Location: Tech Hall

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 20 minutes.

Topic: Project description

Detailed Information: We consolidated on our project description and briefly noted our goals and ideas and listed out the technologies we would be using and emailed them to Dr. Aygun. We decided on meeting on 12th to discuss about the Data Acquisition part.

Meeting 5:

Date & Time: February 12th, 1:30PM

Location: Tech Hall

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 2 Hours

Topic: Data Acquisition

Detailed Information: Initially we faced a little difficulty in finding the data sets as we did not find the exact desired data at one place. Finally, after consulting Dr. Aygun we gathered data from random generators for the users table and found a Kaggle dataset for the attractions table.

Meeting 6:

Date & Time: February 27th, 1:30PM

Location: Tech Hall

Attendees: Sravani Koppala, Snehal Gaikwad (Vichetna didn't attend as she was in India)

Duration: 2 Hours

Topic: MongoDB.

Detailed Information: We had to start from basics, as none of us had any prior knowledge on MongoDB. We decided to use MongoDB compass and MongoDB atlas for cloud and created accounts in Atlas and created a database schema.

Meeting 7:

Date & Time: March 5th, 1:30PM

Location: Tech Hall

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 1 Hour

Topic: Formal description of database.

Detailed Information: Identified all the entities and relations that are being used in the project and developed the ER model from the inputs provided in the class.

Meeting 8:

Date & Time: March 9th, 4:00PM

Location: Laurelwood Apartments

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 3 Hours

Topic: Creation of tables in database (MongoDB).

Detailed Information: We started creating tables according to our ER diagram. Initially it was difficult for us but after referring to a few examples, we created a few sample tables and thought of connecting to the front end in the next meeting.

Meeting 9:

Date & Time: March 14th, 2:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 1 Hour 50 minutes

Topic: Front end and Back end

Detailed Information: We met online on Google hangouts as we are not supposed to have face to face meetings due to COVID-19. We could not understand the backend nodeJS to write the logic as per our application and connect the same to

React JS as all 3 of us were amateurs and we were running out of time. We decided that PHP was easier and all of us knew SQL better and hence fixed on changing our DB from MongoDB to MySQL. Also, it was difficult for us to work on an entirely new database in less time and that too without meeting each other. So, we decided to continue the project implementation in php and SQL.

Meeting 10:

Date & Time: March 17th, 12:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 1 Hour 20 minutes

Topic: Discussion on bonus part and SQL tables

Detailed Information: We discussed moving our data to cloud for the bonus part if we finish implementing our project before the deadline. We showed each other our progress on creation of tables and logic.

Meeting 11:

Date & Time: March 19th, 2:00PM

Location: Overlook Apartments

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 2 Hours

Topic: Clubbing individual works

Detailed Information: We combined the logics of all individual codes and made a functional database and decided to use BootStrap for styling.

Meeting 12:

Date & Time: March 27th, 10:00AM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 2 Hours

Topic: Front-end

Detailed Information: We worked on the project environment setup, installing the web servers wamp as needed and testing with sample code to connect with sample database.

Meeting 13:

Date & Time: April 2nd, 1:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 2 hours 20 Minutes

Topic: Implementing the front-end

Detailed Information: We finally began implementing the front-end design by designing the server pages and made it run on local host. We combined our individual works on the pages and the Front-end was successfully implemented.

Meeting 14:

Date & Time: April 4th, 3:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 2 hours 45 Minutes

Topic: Resolved the problems faced in Implementation.

Detailed Information: As we tried connecting, we encountered many problems, we worked together in performing the database operations like updating, deleting necessary functionalities in our project and almost finished working on these functions with minor changes left over in the project.

Meeting 15:

Date & Time: April 6th, 6:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 5 Hours

Topic: Designing web pages and testing

Detailed Information: We started designing the webpages with background pictures, fonts etc., Then we tested the working of page functionalities and checked

for tuple insertion/deletion/updating. We added more tuples and tried the above functions and got good results. We started to work on final report.

Meeting 16:

Date & Time: April 8th, 3:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 2 Hours

Topic: Bonus Part

Detailed Information: We decided to move our database to cloud and informed Dr.Aygun about the same. We did the bonus part by creating an Account in AWS and running the SQL scripts and changing the connections to AWS in the PHP files and started working on final report by dividing work.

Meeting 17:

Date & Time: April 10th, 5:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 3 Hours

Topic: Final Report

Detailed Information: We finished documentation and reviewed the report. After a discussion, we changed a spelling mistakes in ER diagram. We changed the cardinality between Trips and Attractions tables, and we indicated total participation of attractions in trips.

Meeting 18 :

Date & Time: April 13th, 12:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 1 Hour

Topic: Project Demo

Detailed Information: We discussed how the final demo should look like and practiced for same.

Meeting 19 :

Date & Time: April 14th, 6:00PM

Location: Online (Hangouts)

Attendees: Sravani Koppala, Snehal Gaikwad, Vichetna Gaddam

Duration: 40minutes

Topic: Final Review

Detailed Information: We reviewed our project and report by checking if everything is working fine and did a final rehearsal.

15.2. REFERENCES

1. Lecture notes by Dr. Ramazan Aygun at UAH.
2. Textbook: Fundamentals of Database Systems, 7th Edition by Elamsri and Navathe.
3. Bootstrap: “<https://getbootstrap.com/docs/3.4/getting-started/>”
4. Fontawesome: “<https://fontawesome.com/>”
5. Inserting photo: “<https://makitweb.com/upload-and-store-an-image-in-the-database-with-php/>”
6. AWS cloud: “<https://console.aws.amazon.com/rds/home?region=us-east-1#>”