RatatUTD

Team 2: Second Is Best

Nadeeba Atiqui, David Tepeneu, Viet Vu, Harper Wood, Kacie Yee

CS 4347.003
Database Systems
Professor Jalal Omer
1 December 2024

Table of Contents

Introduction

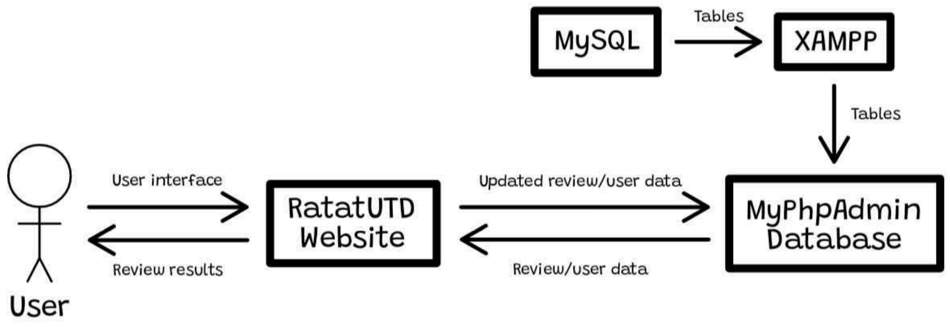
Provide a brief description of the project and the section organization of this report.

For our project, we created RatatUTD, a restaurant review website specifically targeted at UTD students. Using RatatUTD, you can log in, leave reviews, and see reviews of all other restaurants in the area made by other users. All the data displayed on RatatUTD is based on our MySQL server, which is hosted locally. In this report, we will be discussing the system requirements, the conceptual design of the database, a description of the schema, our functional dependencies, how to install our database, and how we designed our user application interface.

System Requirements

Give the context diagram (system architecture diagram) of the database system. List the interface requirements of the system (or each subsystem). List the functional and non-functional requirements of the database system.

Our context diagram (system architecture diagram) is as follows:



Where the user only directly interacts with the website, and the website directly speaks to our database. Our database is initially created with SQL statements which are fed in through a middle entity to help our SQL statements communicate with our database.

To maintain and/or use this system, we require the interfaces:

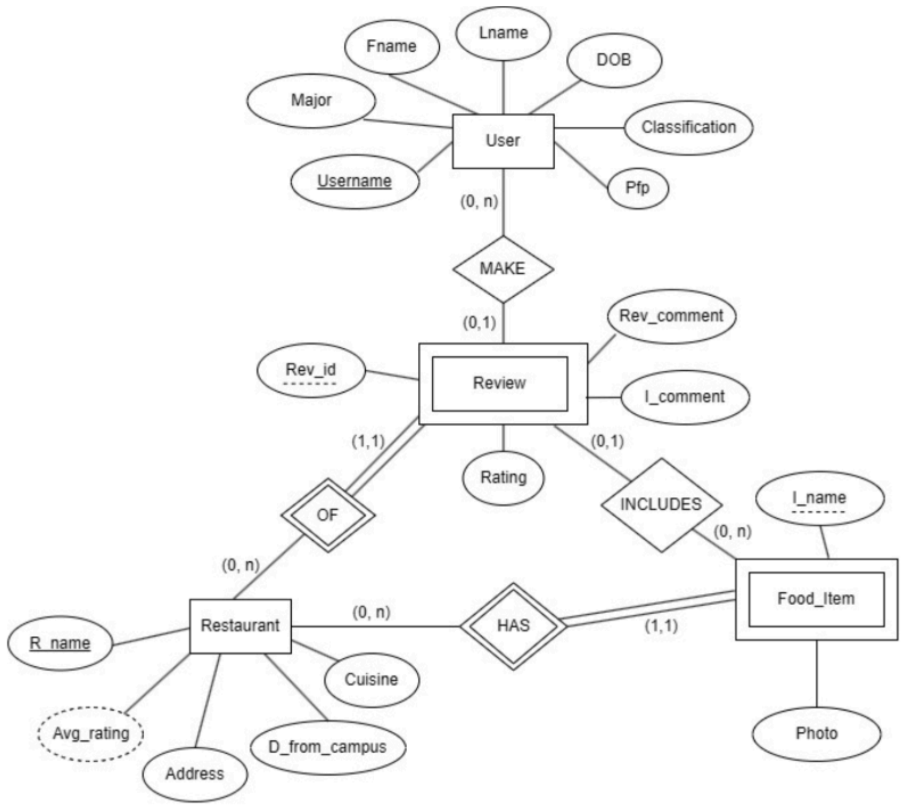
- GitHub - To pull all of our project files to recreate RatatUTD
- Web pages - To interact with our database through a user-friendly interface
- Any text editor - To view and manipulate the HTML code that produces our website
- MyPhpAdmin - To hold the initial and updated data for our database
- XAMPP - To connect MyPhpAdmin to MySQL
- MySQL - To run our CREATE TABLE and LOAD TABLE statements database creation

RatatUTD's functional requirements include creating new data and inputting it into the database, reading data from the database, updating the database's data, deleting from the database, user authentication, and displaying all of the data and any changes onto a front page for the user to comfortably view and interact with. In terms of our non-functional requirements, the development of RatatUTD focused on performance to ensure that users can view their data in a timely manner, scalability to allow for our database to hold an increasing amount of both user and review data, security to prevent common breaches such as SQL injections, and usability to make it so that the user has an easy experience navigating through our web application.

Conceptual Design of the Database

The complete Entity-Relationship (ER) model of your database. The data dictionary and business rules (i.e. constraints) of your ER model.

RatatUTD's full Entity-Relationship model requires four entities interacting like so:



And this is the full data dictionary for this Entity-Relationship model with its business rules:

Table	**Attribute**	**Data type**	**Key info.**	**Constraints**	**Description**
User	Username	VARCHAR (20)	Primary	NOT NULL, UNIQUE	A user's log in identifier
User	Major	VARCHAR (40)	-	-	A user's field of study

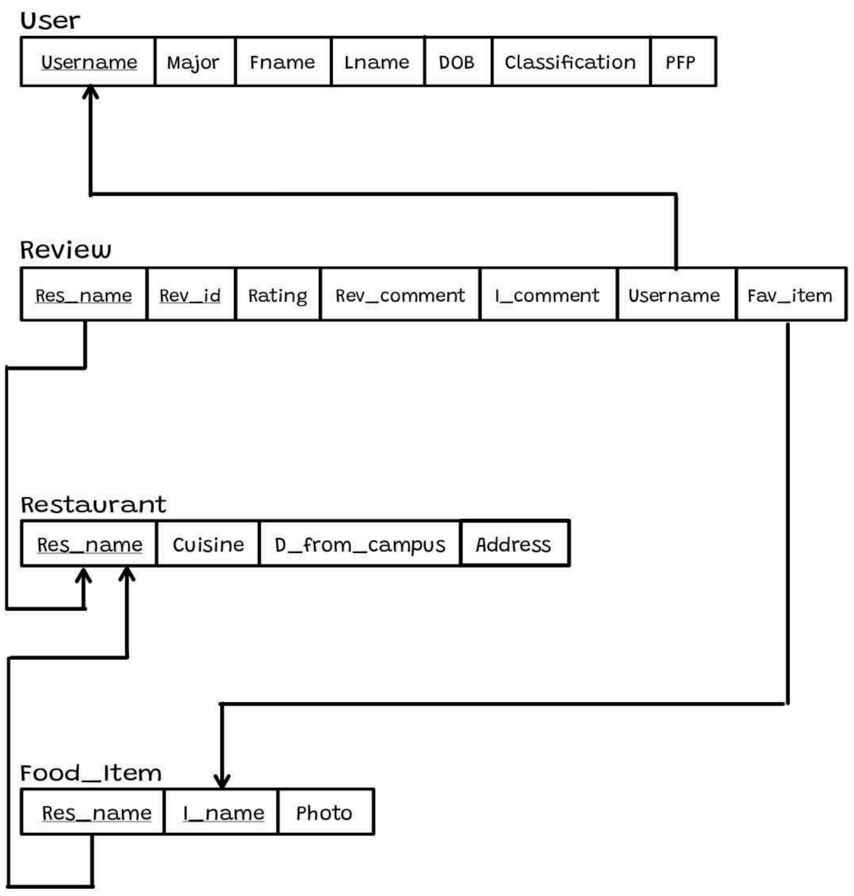
User	Fname	VARCHAR (20)	-	-	A user’s first name
User	Lname	VARCHAR (20)	-	-	A user’s last name
User	DOB	DATE	-	NOT NULL	A user’s date of birth
User	Classification	VARCHAR (10)	-	-	A user’s college classification
User	PFP	VARCHAR (200)	-	NOT NULL, DEFAULT VALUE	A user’s profile picture
Review	Res_name	VARCHAR (50)	Primary, Foreign	NOT NULL, REFERENCES Restaurant(Res_name)	A review’s corresponding restaurant
Review	Rev_id	INT	Primary	NOT NULL, AUTO_INCREMENT	A review’s identification number
Review	Rating	SMALLINT	-	NOT NULL, RANGE(1-5)	A review’s numerical rating
Review	Rev_comment	VARCHAR (500)	-	-	A review’s written rating
Review	I_comment	VARCHAR (500)	-	-	A review’s written rating for an item
Review	Username	VARCHAR (20)	Foreign	DEFAULT VALUE, REFERENCES User(Username)	A review’s corresponding user
Review	Fav_item	VARCHAR (20)	Foreign	REFERENCES Food_item(I_	A review’s chosen favorite item

				name)	
Restaurant	Res_name	VARCHAR (50)	Primary	NOT NULL	A restaurant's name
Restaurant	Cuisine	VARCHAR (20)	-	-	A restaurant's food type
Restaurant	D_from_campus	FLOAT	-	CAN BE NULL, DERIVED	A restaurant's distance from UTD
Restaurant	Address	VARCHAR (100)	-	-	A restaurant's location
Food_item	Res_name	VARCHAR (50)	Primary, Foreign	NOT NULL, REFERENCES Restaurant(Res_name)	An item's corresponding restaurant
Food_item	I_name	VARCHAR (30)	Primary	NOT NULL	An item's name
Food_item	Photo	VARCHAR (200)	-	-	An item's picture

Logical Database Schema

Give the schema of the database which is restructured and translated from the ER diagram presented in the section "Conceptual Design of the Database". Show the schema with appropriate referential constraints. Give the SQL statements used to construct the schema. List the expected database operations and estimated data volumes.

This is the schema for our database that is constructed from our ER diagram:



This schema is translated into our actual database with the following SQL statements:

```
CREATE TABLE User (
    Username VARCHAR(20) PRIMARY KEY NOT NULL,
    Major VARCHAR(40),
    Fname VARCHAR(20) NOT NULL,
    Lname VARCHAR(20),
    DOB DATE NOT NULL,
    Classification VARCHAR(10),
    PFP VARCHAR(200) NOT NULL DEFAULT 'link_to_silhouette_image'
);

CREATE TABLE Restaurant (
      Res_name VARCHAR(50) PRIMARY KEY NOT NULL,
      Cuisine VARCHAR(20),
      D_from_campus VARCHAR(20),
      Address VARCHAR(100)
);
```

```
CREATE TABLE Food_item (
      Res_name VARCHAR(50) NOT NULL,
      I_name VARCHAR(30) NOT NULL,
      Photo VARCHAR(200),
      PRIMARY KEY (Res_name, I_name),
      FOREIGN KEY (Res_name) REFERENCES Restaurant (Res_name)
      ON DELETE CASCADE
);

CREATE TABLE Review (
      Res_name VARCHAR(50) NOT NULL,
      Rev_id INT NOT NULL,
      Rating SMALLINT CHECK (Rating BETWEEN 1 AND 5),
      Rev_comment VARCHAR(500),
      I_comment VARCHAR(500),
      Username VARCHAR(20),
      Fav_item VARCHAR(30),
      PRIMARY KEY (Res_name, Rev_id),
      FOREIGN KEY (Res_name) REFERENCES Restaurant (Res_name)
      ON DELETE CASCADE,
      FOREIGN KEY (Username) REFERENCES User (Username)
      ON DELETE CASCADE,
      FOREIGN KEY (Res_name, Fav_item) REFERENCES Food_item (Res_name, I_name)
      ON DELETE CASCADE
);
```

We expect our database operations to include creating new users, retrieving user data, updating user details, deleting a user from the database, creating a new review, displaying reviews for users to read, and deleting a review. For future goals, we would also calculate the highest scored restaurants or use the gathered data for another way to help users better choose what restaurant they may want to try out. As our database is geared towards UTD students in particular, we expect the volume for this database to at most hold every student at UTD, with an average of 2 reviews per student as our application gains popularity.

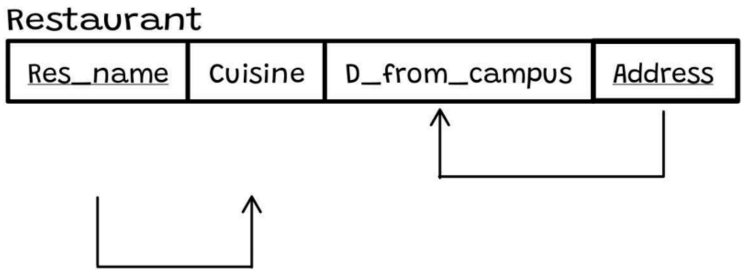
Functional Dependencies and Database Normalization

Identify and analyze the functional dependencies for each relation presented in the section "Logical Database Schema". Show the normalization process and normalized tables for each relation to 3NF (if applicable). Give the SQL statements for constructing the normalized table (if applicable).

As is required for the primary key of any relation, the primary key of each of our relations determines all other attributes of the relation. Additionally, the Address attribute of the Restaurant relation determines D_from_campus (Address → D_from_campus). The relations have no composite or multivalued attributes, and there are no nested relations, so they are in First Normal Form (1NF). The relations are in Second Normal Form (2NF) because they are in 1NF and there are no instances of a proper subset of a key determining a none-prime attribute. They are also in Third Normal Form (3NF) because they are in 2NF and have no transitive Functional Dependencies (FDs). Therefore, we performed no normalization and implemented the relations as they are illustrated above.

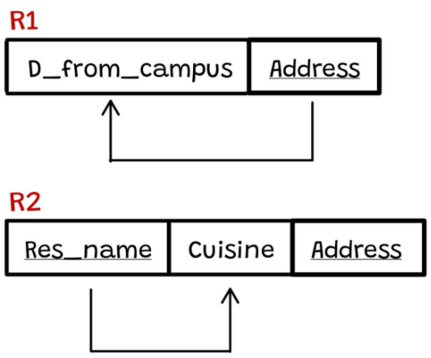
However, in order to demonstrate knowledge of database normalization, we created an altered version of the Restaurant relation where Res_name and Address together form the primary key. This version of the relation would allow a restaurant to have multiple locations. This was not allowed in the version of the database design which we implemented, but is useful to demonstrate normalization. This version has two FDs: Res_name → Cuisine; Address → D_from_campus.

This relation still has no composite attributes, multivalued attributes, or nested relations, so it is in 1NF. The Restaurant relation has key {Res_name, Address} and two FDs:
FD1: Address → D_from_campus
FD2: Res_name → Cuisine



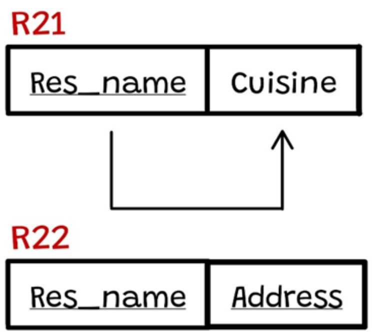
Examining the FD1, Address → D_from_campus, we see that Address is a proper subset of a key and D_from_campus is a none-prime attribute. Therefore, FD1 fails and the Restaurant relation is not in 2NF.

Decomposing this relation, we get:
R1 = (Address)+ = {Address, D_from_campus} with FD1: Address D_from_campus
R2 = (R – (Address)+) U Address = {Res_name, Cuisine, Address} with FD2: Res_name Cuisine



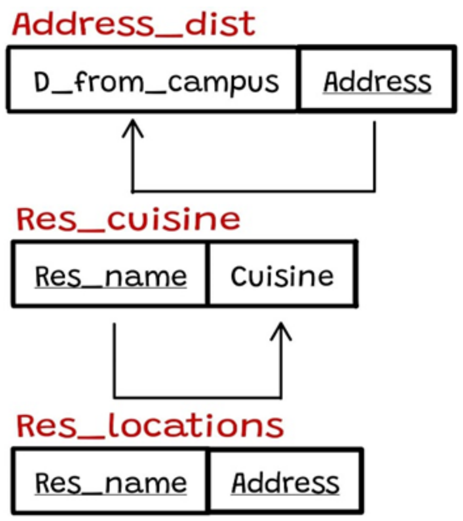
Examining R1 with FD1, we see that Address is not a proper subset of a key, it is a complete key, so FD1 does not fail and R1 is in 2NF.
Examining R2 with FD2, we see that Res_name is a proper subset of a key and Cuisine is a none-prime attribute, so FD2 fails and R2 is not in 2NF.

Decomposing this relation, we get
R21 = (Res_name)+ = {Res_name, Cuisine} with FD2: Res_name Cuisine
R22 = (R – (Res_name)+) U Res_name = {Res_name, Address}



Examining R21 with FD2, we see that Res_name is not a proper subset of a key, it is a complete key, so FD2 does not fail and R21 is in 2NF.
R22 has no FDs, so it is in 2NF as well.

The result is 3 relations:
R1 = Address_dist = {Address, D_from_campus} with FD1: Address D_from_campus
R21 = Res_cuisine = {Res_name, Cuisine} with FD2: Res_name Cuisine
R22 = Res_locations = {Res_name, Address}



All the relations are in 2NF, and since they have no transitive FDs, they are all in 3NF as well.

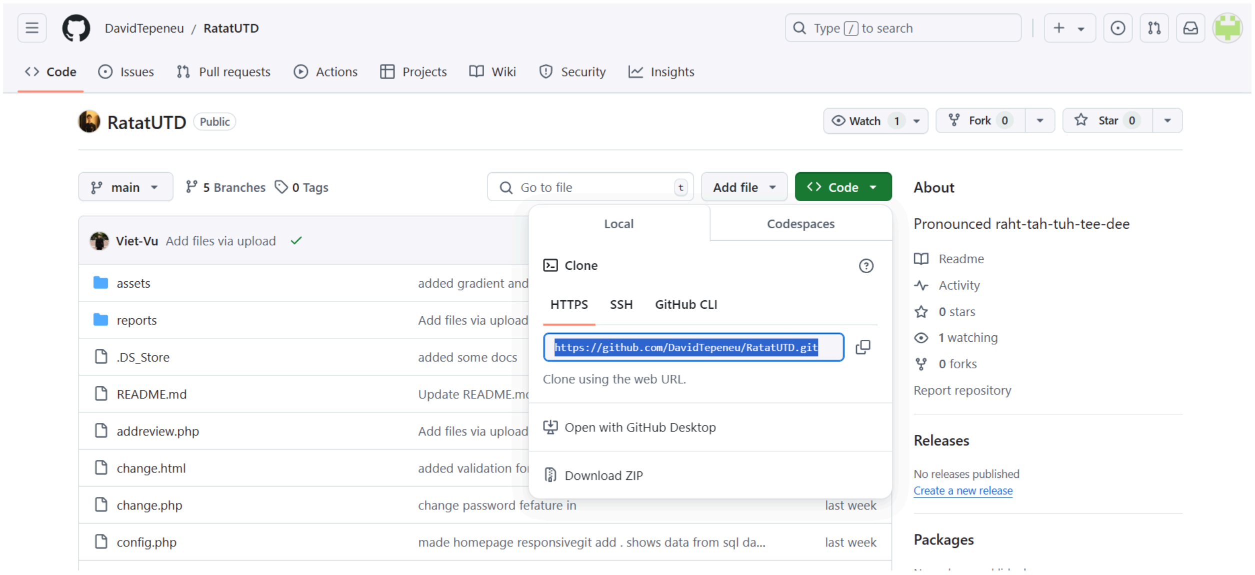
Since this was not the version of the Restaurant relation that we implemented for our project and was only meant to demonstrate the normalization process, we did not implement these 3 relations. The Restaurant relation we implemented was already in 3NF. However, if one wanted to implement these three relations, the SQL create table statements would be as follows:

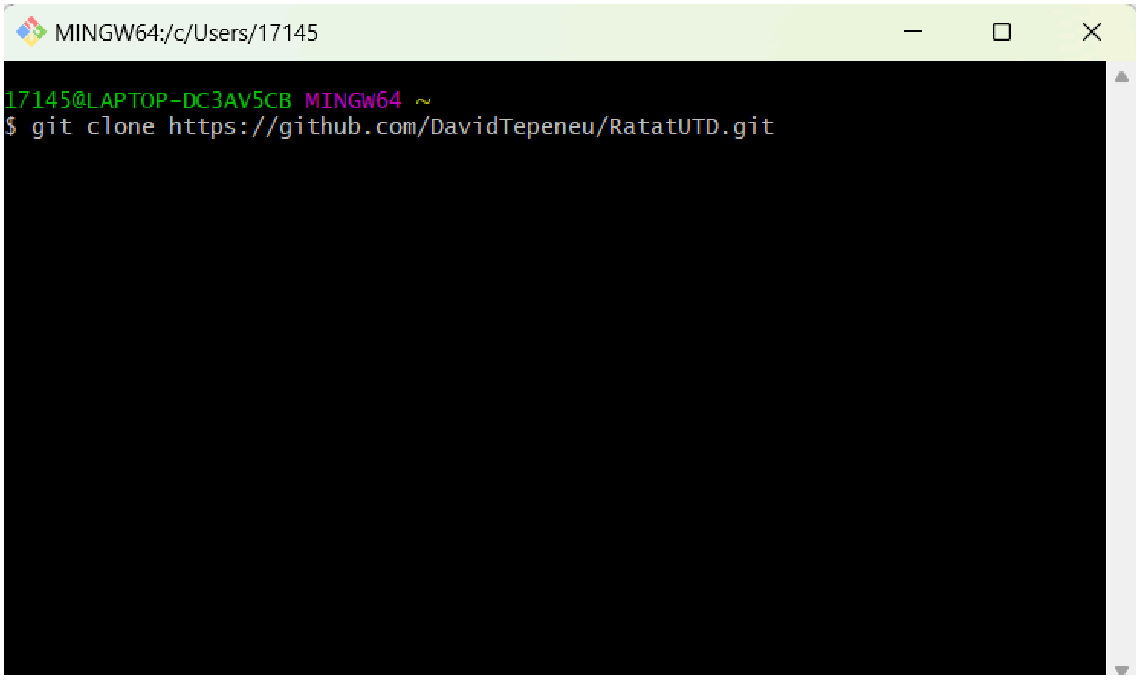
```
CREATE TABLE Address_dist (
      Address             VARCHAR(100)      NOT NULL,
      D_from_campus       VARCHAR(20),
      PRIMARY KEY (Address)
);

CREATE TABLE Res_cuisine (
      Res_name            VARCHAR(50)       NOT NULL,
      Cuisine             VARCHAR(20),
      PRIMARY KEY (Res_name)
);

CREATE TABLE Address_dist (
      Res_name            VARCHAR(50)       NOT NULL,
      Address             VARCHAR(100)      NOT NULL,
      PRIMARY KEY (Res_name, Address)
);
```

The Database System

Give a brief description of how to install and invoke your system. Provide the "screen dumps" showing how to use your system step by step.

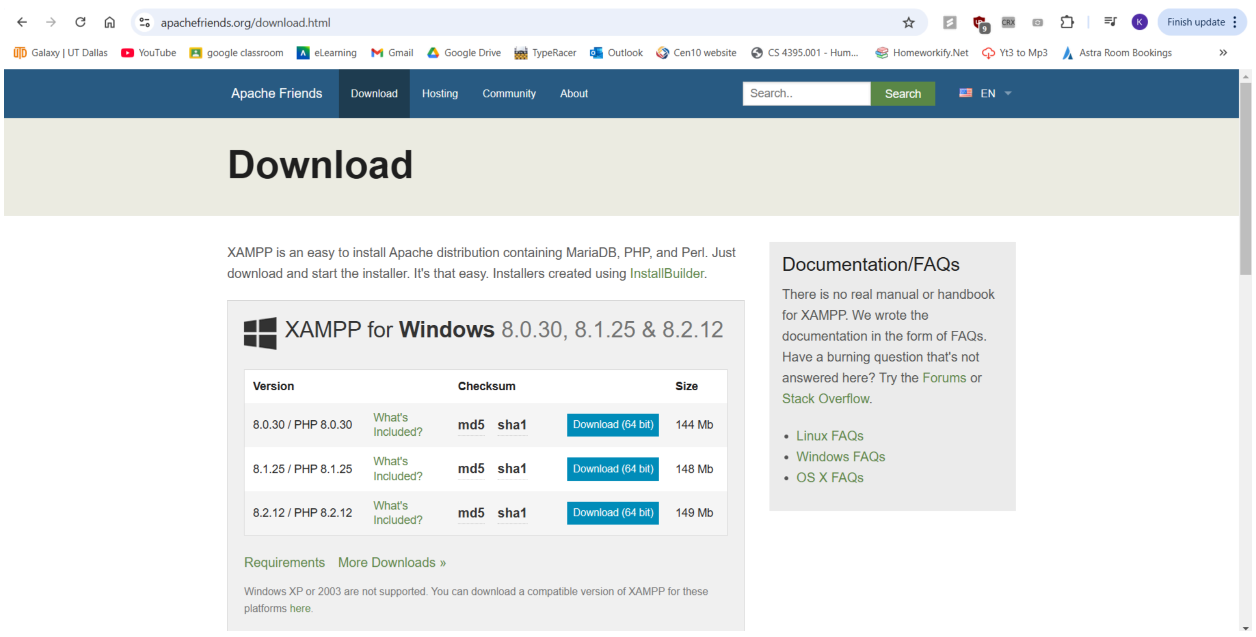
First, head to our repository hosted on Github, and using the link pictured below, clone our repository onto your local machine.





```
17145@LAPTOP-DC3AV5CB MINGW64 ~
$ git clone https://github.com/DavidTepeneu/RatatUTD.git
```

Now, you should have our code on your local machine. Next, we need to set up our database. You will need to download XAMPP at the linked website.



Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy. Installers created using InstallBuilder.

XAMPP for **Windows** 8.0.30, 8.1.25 & 8.2.12

Version		Checksum		Size
8.0.30 / PHP 8.0.30	What's Included?	md5 sha1	Download (64 bit)	144 Mb
8.1.25 / PHP 8.1.25	What's Included?	md5 sha1	Download (64 bit)	148 Mb
8.2.12 / PHP 8.2.12	What's Included?	md5 sha1	Download (64 bit)	149 Mb

Requirements More Downloads »

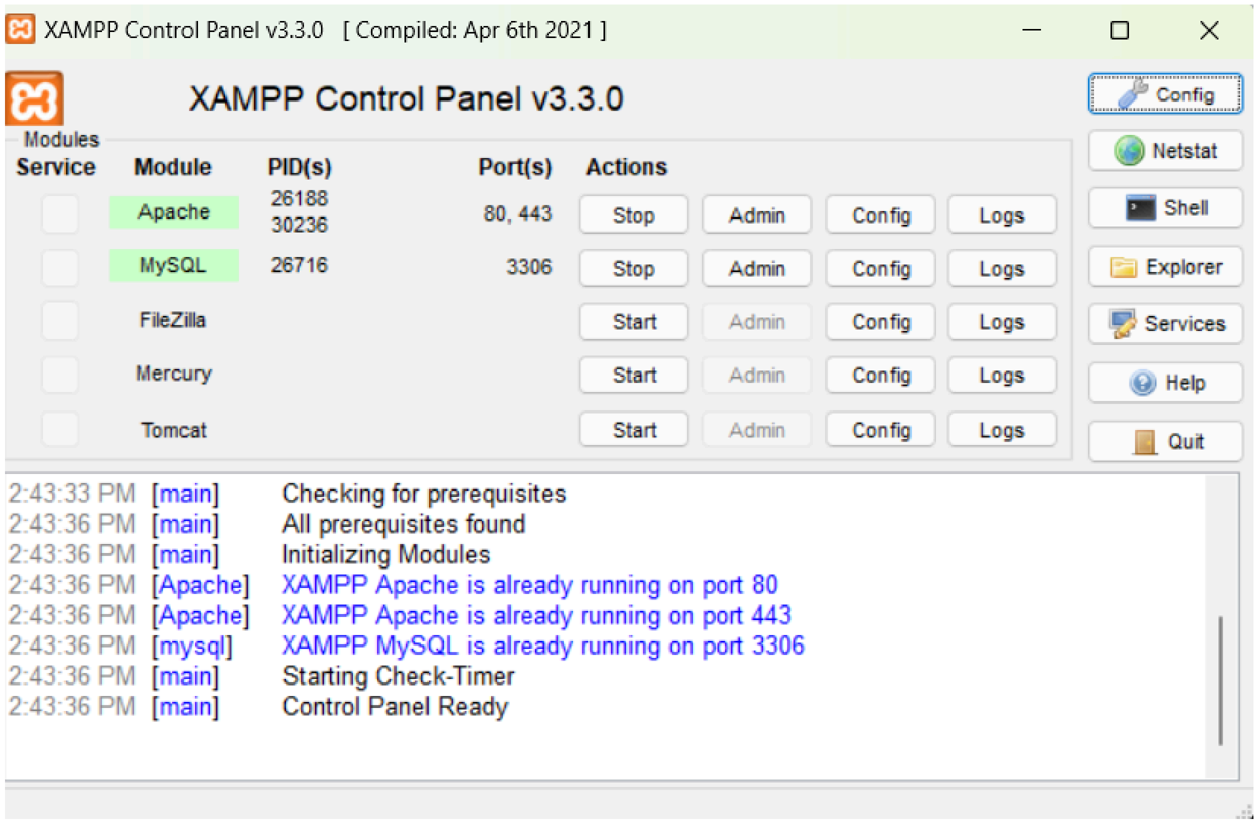
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs

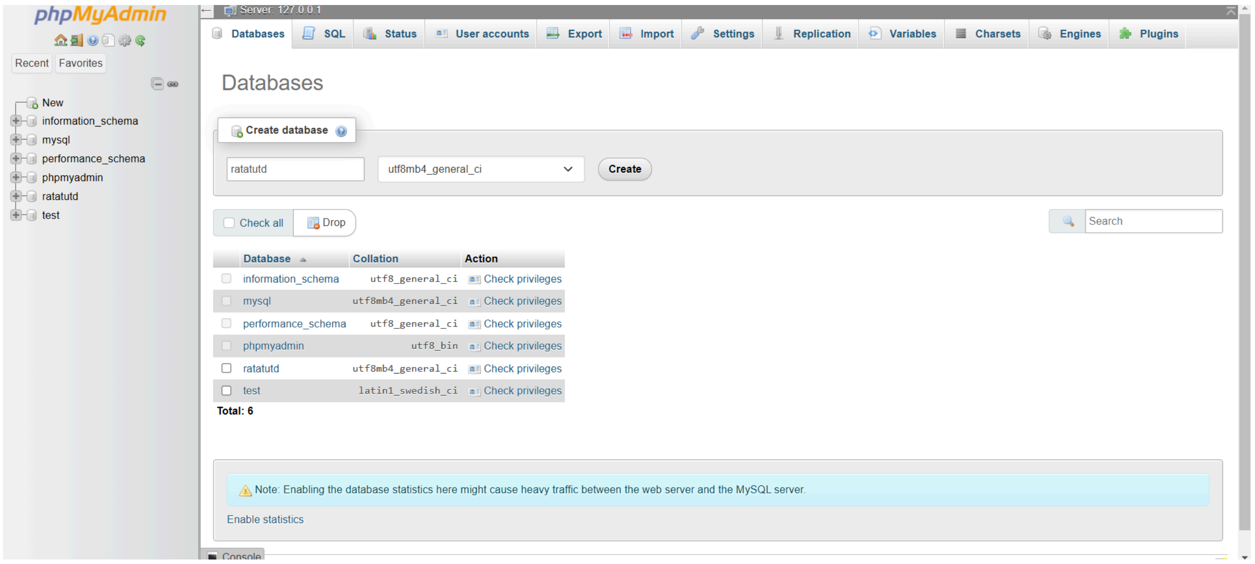
Select the appropriate version of XAMPP and follow all download instructions until you have the XAMPP Control Panel open on your device. Then, click the Start button next to Apache, followed by the Start button next to MySQL. Once both have been successfully started, your control panel should look like this:


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Config
Netstat
Shell
Explorer
Services
Help
Quit

Modules Service	Module	PID(s)	Port(s)	Actions			
	Apache	26188 30236	80, 443	Stop	Admin	Config	Logs
	MySQL	26716	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
2:43:33 PM  [main]     Checking for prerequisites
2:43:36 PM  [main]     All prerequisites found
2:43:36 PM  [main]     Initializing Modules
2:43:36 PM  [Apache]   XAMPP Apache is already running on port 80
2:43:36 PM  [Apache]   XAMPP Apache is already running on port 443
2:43:36 PM  [mysql]    XAMPP MySQL is already running on port 3306
2:43:36 PM  [main]     Starting Check-Timer
2:43:36 PM  [main]     Control Panel Ready
```

Now, click on Admin for MySQL, which brings you to phpmyadmin. Here, go to the Databases tab, and create a new database called “ratatutd”.



Databases

Create database

ratatutd | utf8mb4_general_ci | Create

Check all | Drop | Search

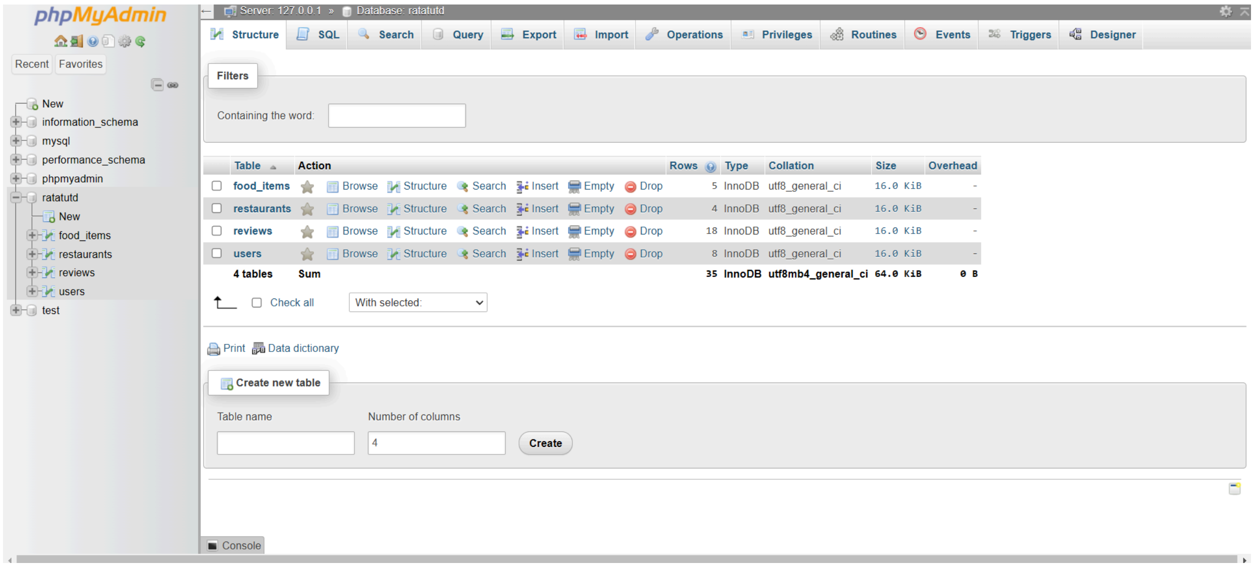
Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
ratatutd	utf8mb4_general_ci	Check privileges
test	latin1_swedish_ci	Check privileges

Total: 6

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

Enable statistics

Once you do, ratatutd should appear as an option in the hierarchy on the left. Click on it, then click on the Import tab. From there, upload all of our .csv files as included in our Appendix to create our tables. After you do so, your database should look like this:



Filters

Containing the word:

Table	Action						Rows	Type	Collation	Size	Overhead
food_items	Browse	Structure	Search	Insert	Empty	Drop	5	InnoDB	utf8_general_ci	16.0 KiB	-
restaurants	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8_general_ci	16.0 KiB	-
reviews	Browse	Structure	Search	Insert	Empty	Drop	18	InnoDB	utf8_general_ci	16.0 KiB	-
users	Browse	Structure	Search	Insert	Empty	Drop	8	InnoDB	utf8_general_ci	16.0 KiB	-
4 tables	Sum						35	InnoDB	utf8mb4_general_ci	64.0 KiB	0 B

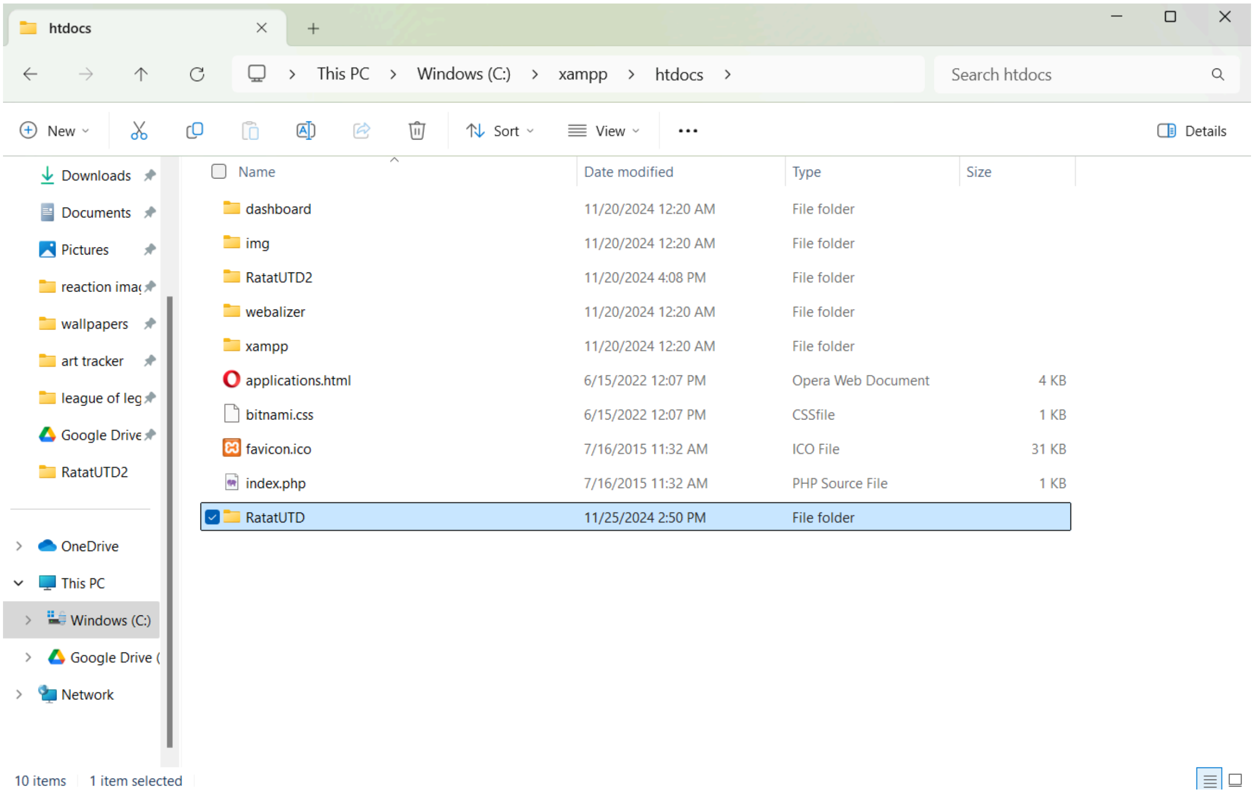
Check all | With selected:

Print | Data dictionary

Create new table

Table name | Number of columns: 4 | Create

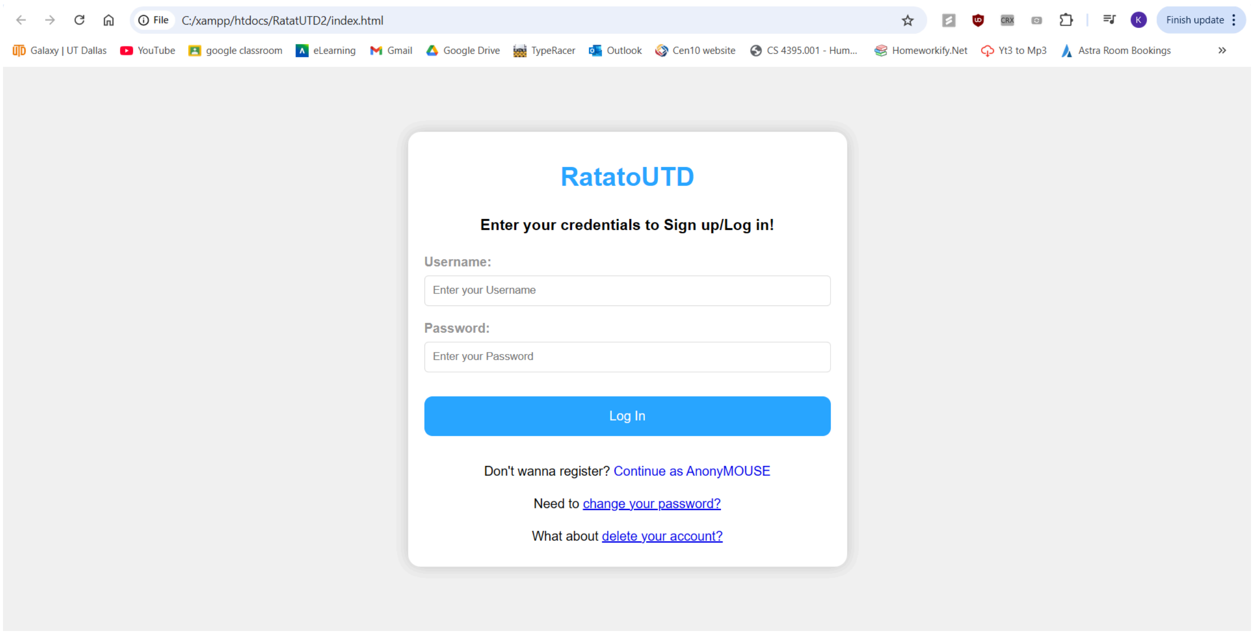
Now, our MySQL database is set up and ready to use on our app. To open our webpage, find the folder where you cloned our Git repository into, and copy paste it into the htdocs folder, which is in the xampp folder which comes with your XAMPP download. Once you do, the folder should look like this:



Name	Date modified	Type	Size
dashboard	11/20/2024 12:20 AM	File folder	
img	11/20/2024 12:20 AM	File folder	
RatatUTD2	11/20/2024 4:08 PM	File folder	
webalizer	11/20/2024 12:20 AM	File folder	
xampp	11/20/2024 12:20 AM	File folder	
applications.html	6/15/2022 12:07 PM	Opera Web Document	4 KB
bitnami.css	6/15/2022 12:07 PM	CSSfile	1 KB
favicon.ico	7/16/2015 11:32 AM	ICO File	31 KB
index.php	7/16/2015 11:32 AM	PHP Source File	1 KB
RatatUTD	11/25/2024 2:50 PM	File folder	

10 items 1 item selected

Once you have copied our folder into htdocs, click into the folder and locate index.html. Once you do, right-click index.html and open it in your browser of choice (personally, we prefer Google Chrome), which will successfully open our app on your device.



RatatoUTD

Enter your credentials to Sign up/Log in!

Username: Enter your Username

Password: Enter your Password

Log In

Don't wanna register? Continue as AnonyMOUSE

Need to change your password?

What about delete your account?

Congrats, you have now successfully set up RatatUTD on your device. Happy reviewing!

Suggestions on Database Tuning (optional)

Give suggestions on tuning your database in terms of index structures, database design, and/or queries.

New users creating accounts and current users creating reviews are extremely common transactions in our database application, so the insert operation is performed often. This means our database would be slowed by the addition of indices, as the indices would have to be maintained with all of the insert operations as well as updates and deletions. Additionally, we perform relatively few select queries with where clause conditions. This means we have no real reason to favor any additional indices. For this reason, we do not suggest adding any additional indices to our database.

Our application does not currently utilize range queries, and it does not display data in any particular order. For this reason, hash tables are a more efficient physical design for our database than B+ trees.

Note that this means, when writing queries for our database as a casual user, range queries will be somewhat slower to execute.

User Application Interface

Describe how you build the system user interface and how users use your system.

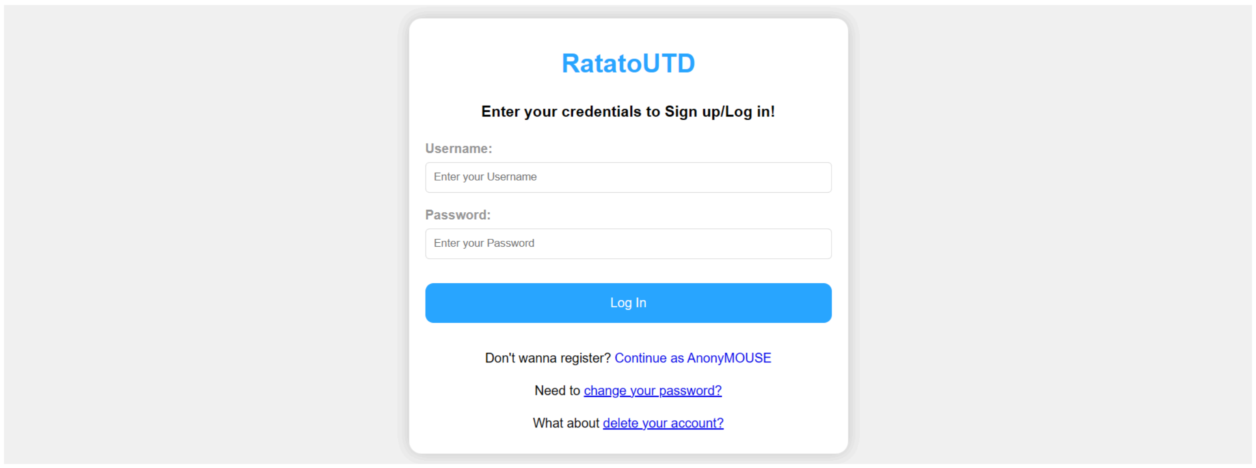
The user application interface has 3 main components: login/sign-in page, homepage, and add review modal.

The system was built using native html, css, javascript, and php. We utilized traditional web page design concepts such as href tags for navigation, javascript for text validation, and form tags for data transfer/communication with our DB.

Github was used for version control so that we could all contribute simultaneously.

Xampp was used to access phpmyadmin and host it locally.

Login/Sign-In


RatatoUTD

Enter your credentials to Sign up/Log in!

Username:

Enter your Username

Password:

Enter your Password

Log In

Don't wanna register? Continue as AnonyMOUSE

Need to change your password?

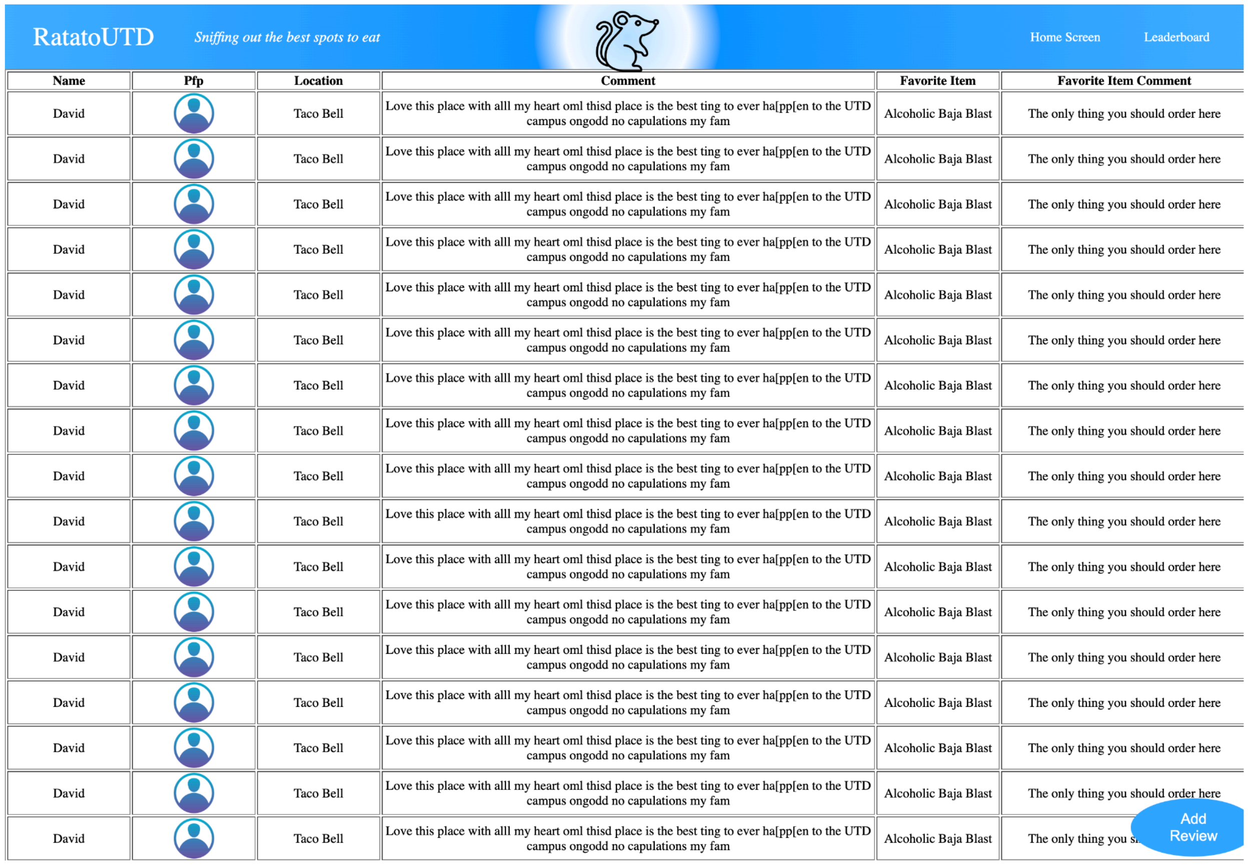
What about delete your account?

This page allows the user to do a number of things.

1) Create a new account and then log in with those credentials (inserts user into DB)
2) Login with a pre-existing account (checks if user exists in DB then signs them in)
3) Continue to the homepage without an account (bypass signing in/remain anonymous)
4) Change password for your account (update user password in DB)
5) Delete their account (deletes user record from DB)

This simple design highlights all the CRUD concepts that would need to be utilized for any user account (more on this in the future work section).

Homepage



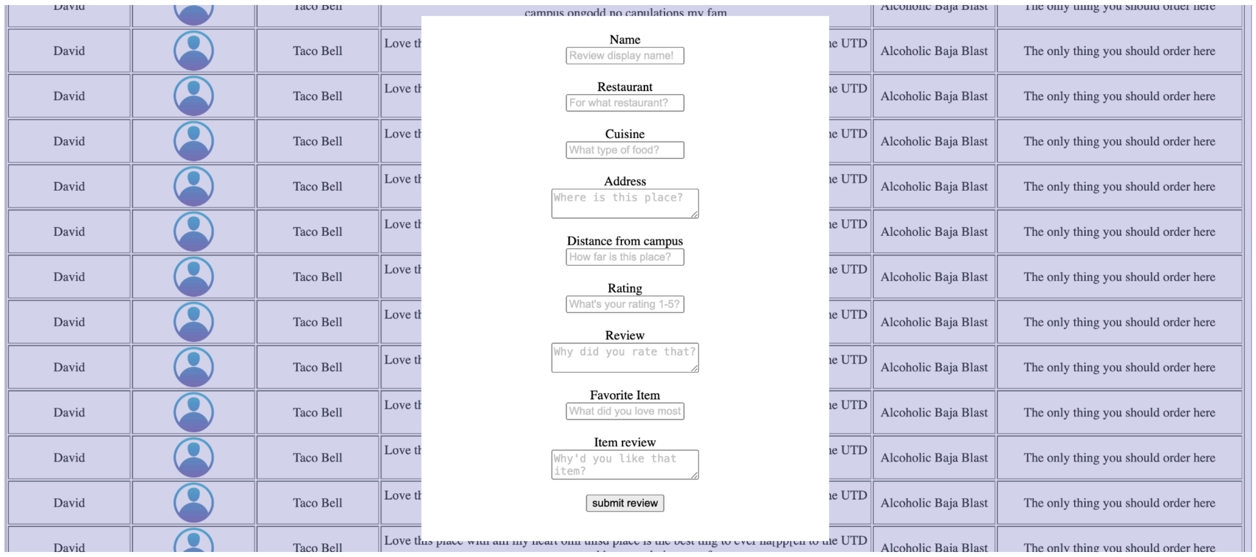
Name	Pfp	Location	Comment	Favorite Item	Favorite Item Comment
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David	👤	Taco Bell	Love this place with alll my heart oml thisd place is the best ting to ever ha[pp[en to the UTD campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you s

**mock data for demonstrative purposes*

This page contains some navigational components and loads all reviews stored in our DB. Users can interact with the following components:

1) Homepage button navigates back to this page
2) Leaderboard button takes us to a relatively empty podium page, this was a stretch goal.
3) Add review button reveals a popup that the user can interact with to leave a review.

Add Review Popup



David		Taco Bell	campus ongodd no capulations my fam	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love th ne UTD	Alcoholic Baja Blast	The only thing you should order here
David		Taco Bell	Love this place with all my heart omn thsd place is the best thig to ever ha[pp[en to the UTD	Alcoholic Baja Blast	The only thing you should order here

Name
Review display name!

Restaurant
For what restaurant?

Cuisine
What type of food?

Address
Where is this place?

Distance from campus
How far is this place?

Rating
What's your rating 1-5?

Review
Why did you rate that?

Favorite Item
What did you love most

Item review
Why'd you like that item?

submit review

Here the user is prompted to type up their review and then submit it when done. No other interactions are possible from this popup. On either action the popup is closed and the user is returned to the homepage.

This was built using javascript to hide and reveal the popup form wrapped in a div. Event listeners mainly.

Conclusions and Future Work

Give a conclusion or your feedback about this project. Provide a brief description of possible future work.

Evidently, implementing databases into web applications can prove to be a substantial challenge, especially when dealing with the hurdles of new technologies (XAMPP controller) and mixed operating systems (macOS vs Windows) in how their environment setup can vary dramatically. Once all our systems were up and running, the CRUD features essential for our website were not all too difficult to implement but we learned not to underestimate the learning curve that accompanies full stack projects. As this was our first DB implemented system for many of us, a lot was learned in a short span of time and we only wish that we could have utilized a more agile approach instead of the big bang rollout we had to induce for the presentation in front of the TA. That being said, this project was immensely beneficial for both our professional and academic development.

In the future there are several features that we would like to incorporate into our project.

1) The leaderboards page would eventually derive the top 3 restaurants and show them in a podium style ranking.
2) Forgot to add the close button on the add review popup, this would be an action item.
3) Users would be able to retrieve their passwords if forgotten by adding another level of authentication, often referred to as 2FA, such as their email or phone number.
4) Users would be allowed to modify their review after submission, but only when signed in to the account that posted it, much like modern comment threads in youtube or X (formerly twitter)
5) Taking our DB off of local and hosting it on some server elsewhere
6) Take our webpage hosting off of github pages and onto a free service like vercel or render

References

List the references or books used for this project.

Aside from the material presented to us in class, we utilized the following resources:

Xampp documentation: https://docs.joomla.org/XAMPP

Html + Css + Javascript:

1) https://www.w3schools.com/
2) Numerous Stack overflow threads

Video tutorials on xampp + php:

1) https://youtu.be/Q4-ZxSFTikU?si=YWc7SdD0tuttKKuJ
2) https://youtu.be/WYufSGgaCZ8?si=fSjeGEWoIWyEsr_X
3) https://youtu.be/J0L2T1D1eRY?si=0r-3fI9rL6PiXZvb
4) https://youtu.be/O0ttkT9Uc6k?si=Ys0ybFeQiMU7dTR9

Appendix

An appendix gives the zip file containing the work products (including demo slides, final report, SQL scripts, and source code of the project). The zip file must have the following directories for all the teams: /doc (contain all documents and presentation slides) /project (contain all source code, test code, data, web pages, SQL scripts, library, and executable files) a README file (describing how to install and use your program).

View zip file attached with this report submission on eLearning.