ITCS 6160 Project - Spring 2017

**Restaurants/Cafe/Pub- Info & Reviews**

Group 4:

* Kasey Eljoundi
* Sai Kiran Mankena
* Varchasvi Verma
* Amulya Kamshetty
* Jasleen Arora
* Pooja Reddy Nathala

Table of Contents:

Phase I

1. Fact finding techniques and Information Gathering… ….……….......3

1.1 Project Requirement…………………………….…….….....3

* 1. Scope of Work………………………………….………......4
  2. Business Rules………………………………….……….….5
  3. Constraints……………………………………….………....6
  4. EER Diagram…………………………………….…….…...7

Phase II

1. ER Model and Relational Database

2.1 EERD narrative, features.…………………………………....8

2.2 Tables…………………….………………………………......9

2.3 Relational Database……………………………………….....11

2.4 Data Implementation in SQL.………………………………..11

2.4 Triggers……………………………………………………....12

2.5 Stored Procedures……………………………………………13

Phase III

1. User Interface and PHP Implementation

3.1 Query Optimization Techniques………………………….….14

3.3 Screenshots of System in Action…………………………….15

3.5 Future Scope…………………………………………………17

Team leader and Other Positions……………………….………………..18

Contact Information……………………………………………………...18

Method of Communication & File sharing medium……………………..19

Project Plan………………………………………………………………19

Project Requirements:

Create a prototype of an application for Customers, Food bloggers, and Restaurant owners which will aid them to keep track of Restaurants, Breweries, Cafes in Charlotte. The application will allow the users to get all the relevant information about restaurants including ratings & reviews given by others which will help them making a choice. The application should also facilitate Businesses to add their restaurant/brewery to the application, modify or delete their information if needed.

Objective for this application is to be one-stop shop for browsing restaurants, search for reviews, add reviews, ratings etc. The idea is to provide convenient and efficient service to the customers as well as Business Owners. User can browse through information using various filter options like zip code, cuisine, restaurant type, ratings, number of reviews etc. Using this application, food bloggers, customers and Restaurant owners will be able to access all the information online and this also ensures that they stay connected and updated throughout the process.

Scope of Work:

The scope of project is to create a user interface application that allows users to search for a restaurant/cafe/pub in Charlotte, look for reviews and other related information, add reviews. This Application would also allow business owners to list their business and provide more details on their services and facilities.

Our scope of work is using a database that contains all the information regarding the restaurants in Charlotte that customer would like to know including name, address, zip code, neighborhood, cuisine, ratings, reviews, contact info etc. We will create user specific & restaurant specific logins so that users can browse, add and modify reviews. Restaurants/Cafe/Pub can update their information on application if required via admin. We would also incorporate retrieve and delete feature to keep database up-to-date.

This web-application will provide all the information of Restaurants/Cafe/Pub on a single platform and will give user access to rate & review. These points can be summed up in below points:

1. Keep track of Restaurants, Breweries, Cafes, in Charlotte.
2. Create user specific and business specific logins.
3. User can add new reviews or update his/her old review.
4. Business can update their information if needed like updating address if restaurant has moved to new location through admin.
5. User can browse information by zip code, neighborhood, cuisine, ratings, price range, number of reviews (customer should be able to filter using a single attribute like cuisine or multiple attribute like American cuisine with 4-star rating within $$ price range).
6. Delete the listing if restaurant is permanently closed.

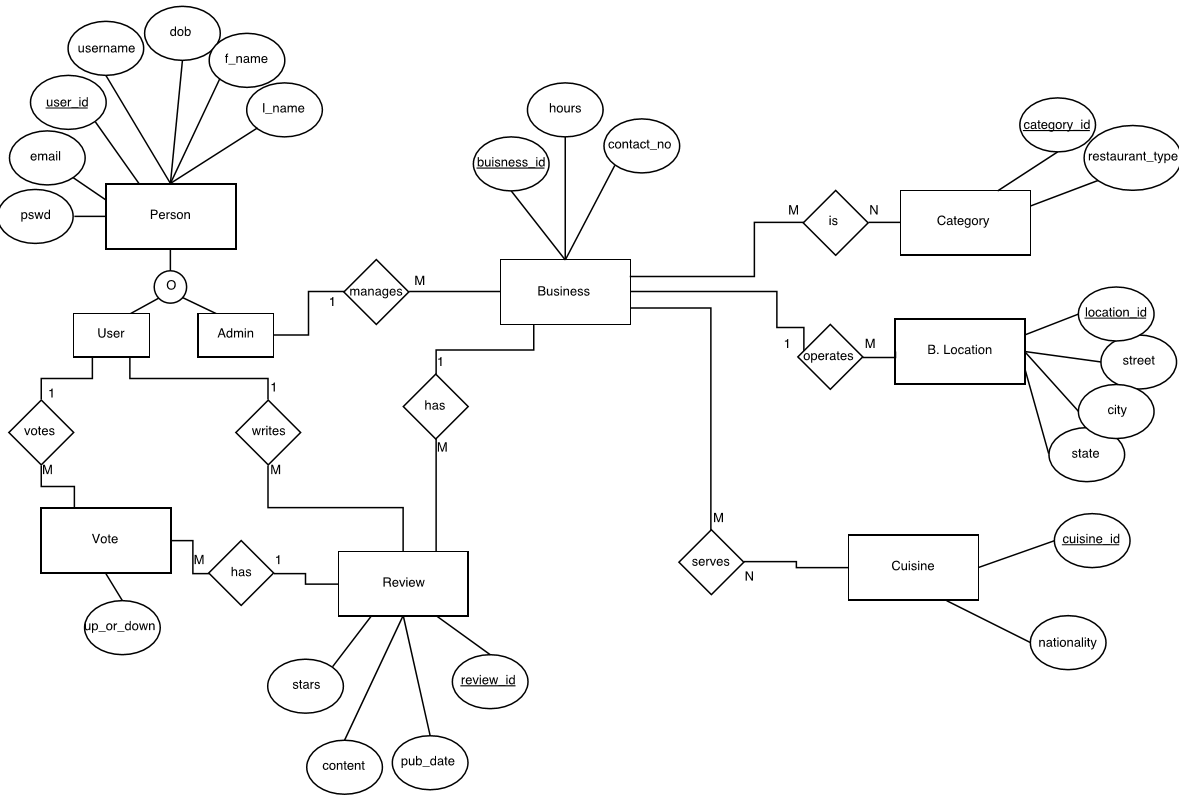
Business Rules:

1. An admin can manage one to many businesses.
2. A business can have one to many locations.
3. A business can be in one or more categories (Restaurant, Cafe, Pub, Brewery, etc.)
4. A business can serve one to many cuisines.
5. A person attribute should be defined using super-sub classification for user & admin and whether the participation is disjoint or overlapping should be indicated. A business can have 0 to many reviews.
6. A user can vote each review only once.
7. A business can be reviewed by multiple users. Also, a single user can review multiple businesses. Hence, many to many relationships.

Constraints:

1. Concrete credentials for person user\_id, email and password.
2. No two persons can have the same user\_id.
3. A User can only vote a review either up or down
4. A User can give the review in the form of stars, or content or both.
5. Only the registered users can vote on or write the review.
6. A person who views the review can be notified of Registration.
7. Only the admin has full access to the system.
8. Only the registered user has the opportunity to delete or edit his review.
9. The view for other user’s review is user preference.
10. The system need to aggregate the ratings for an item in a business and provide an updated average user rating for a particular item in a particular business.
11. A user can only write the review in English language and give the rating up to 5 stars.
12. The aggregate user rating will be notified to the admin for every month.

EER Diagram:



EERD Narrative & Features:

The EERD above depicts the Entity Relationships in the Restaurant/Cafe/Pub - Info & Review system. Every person in the system is stored in person table. A person can be User or Admin. Only Admin and User can login into the system. Type descriptor whether the participation is disjoint or overlapping should be indicated. Participation is between Admin and User is overlapping. An admin manages the business. Admin can add/update new business entry in the system. A Business can be either be a café/pub/restaurant or a combination of these categories. A business can have one to many locations. A business can serve one and only one cuisine and there can be many menu items for a business. One admin can manage one or many businesses. A business can have 0 to many reviews. When a user wants to vote, or write a review, system makes an entry in the review table. A user can then vote or write a review for one or many businesses, and many users can write review or vote for one business. This many-to-many relationship is split into two one-to-many relationships using the Review & Vote tables.

***Features:***

* **Tables are in 3rd Normal Form.**
* **Implementation of Generalization/Specialization:**

General entity Person is specialized into 2 entities - User and Admin with overlapping discriminator.

* **Resolving Many to Many relationship using two one to many relationships:**

The many to many relationship between User and Business entities is resolved by using Vote and Review entities, which has one to many relationships.

Tables:

We are using 10 tables which are listed below along with their data type.

|  |  |
| --- | --- |
| business |  |
| business\_id | int(10) |
| business\_name | varchar(45) |
| user\_id | int(11) |
| hours | varchar(100) |
| contact\_number | varchar(45) |
|  |  |

|  |  |
| --- | --- |
| business\_category\_relation |  |
| business\_id | int(10) |
| category\_id | Varchar(100) |
|  |  |

|  |  |
| --- | --- |
| business\_cuisine\_relation |  |
| business\_id | int(10) |
| cuisine\_id | int(10) |
|  |  |

|  |  |
| --- | --- |
| category |  |
| category\_id | int(10) |
| restaurant\_type | varchar(45) |
|  |  |

|  |  |
| --- | --- |
| business\_location |  |
| business\_location\_id | int(10) |
| business\_id | int(11) |
| street | varchar(45) |
| city | varchar(45) |
| state | varchar(2) |
| zip | varchar(5) |

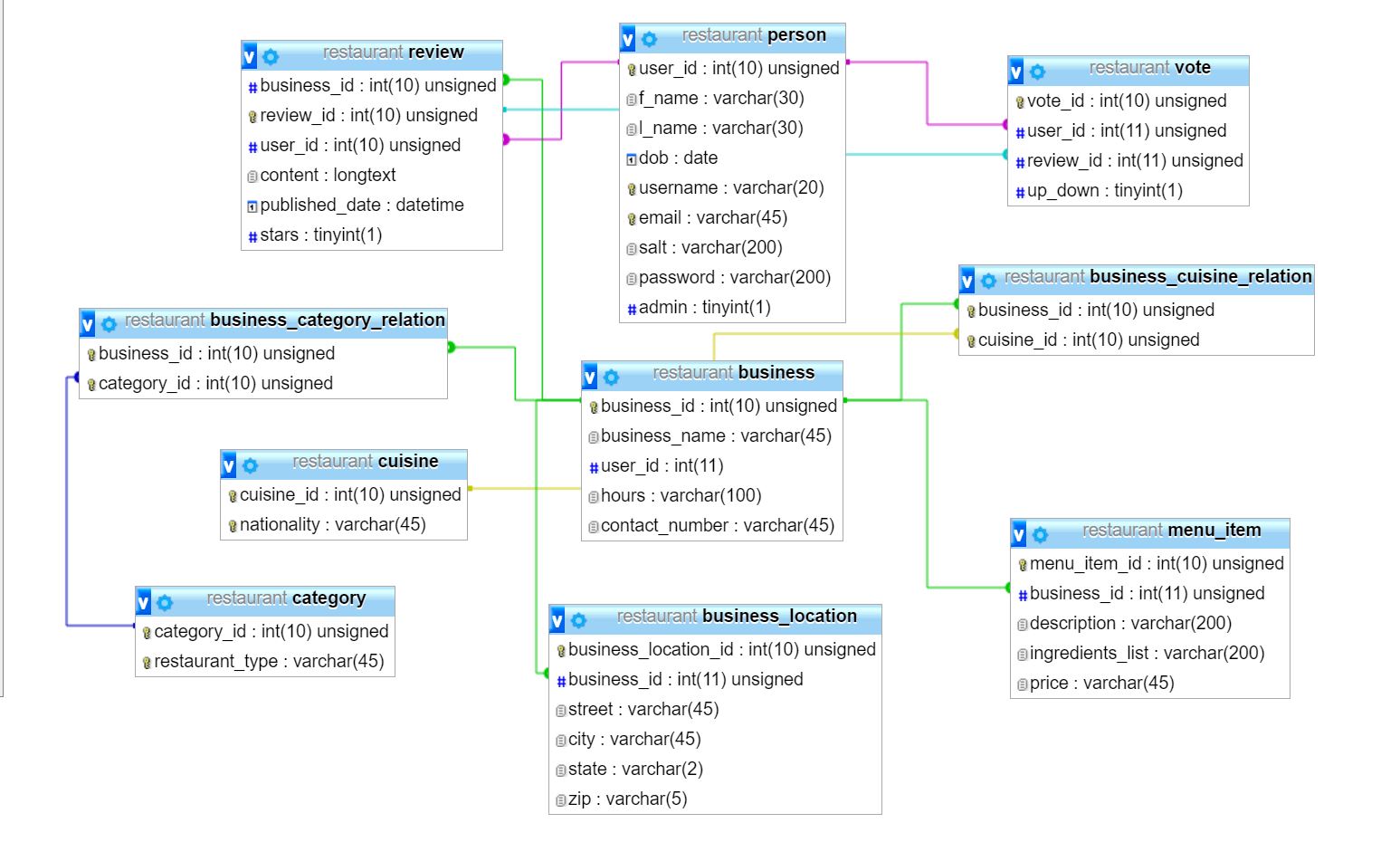
|  |  |
| --- | --- |
| vote |  |
| vote\_id | int(10) |
| user\_id | int(11) |
| review\_id | int(11) |
| up\_down | tinyint(1) |

|  |  |
| --- | --- |
| cuisine |  |
| cuisine\_id | int(10) |
| nationality | varchar(45) |

|  |  |
| --- | --- |
| person |  |
| user\_id | int(10) |
| f\_name | varchar(30) |
| l\_name | varchar(30) |
| dob | date |
| username  email  password | varchar(20)  varchar(45)  varchar(200) |
| admin | tinyint(1) |

|  |  |
| --- | --- |
| review |  |
| business\_id | int(10) |
| review\_id | int(10) |
| user\_id | int(10) |
| content  published\_date  stars | longtext  datetime  tinyint(1) |

Relational Database:



Database Implementation:

The following are the languages confined within the system to enable development and implementation.

* Apache web server to host the web application.
* Cross platform browser to visit the web application.
* Dreamweaver (IDE).
* MySQL for database development, implementation and manage.
* PHP for server side scripting
* Html5 and the CSS to develop the front-end UI design and add style to the web page.

Add/ Update/ Delete Operations:

ADD:

* Reviews are added when user visits a restaurant/Cafe.
* New users are added in person table.
* When a review is made, a new data will be added to the “reviews” table

UPDATE:

* If a review is updated, then all the changes updates will be made at review table.
* Admin can add/update/delete businesses.
* Users details can also be updated.

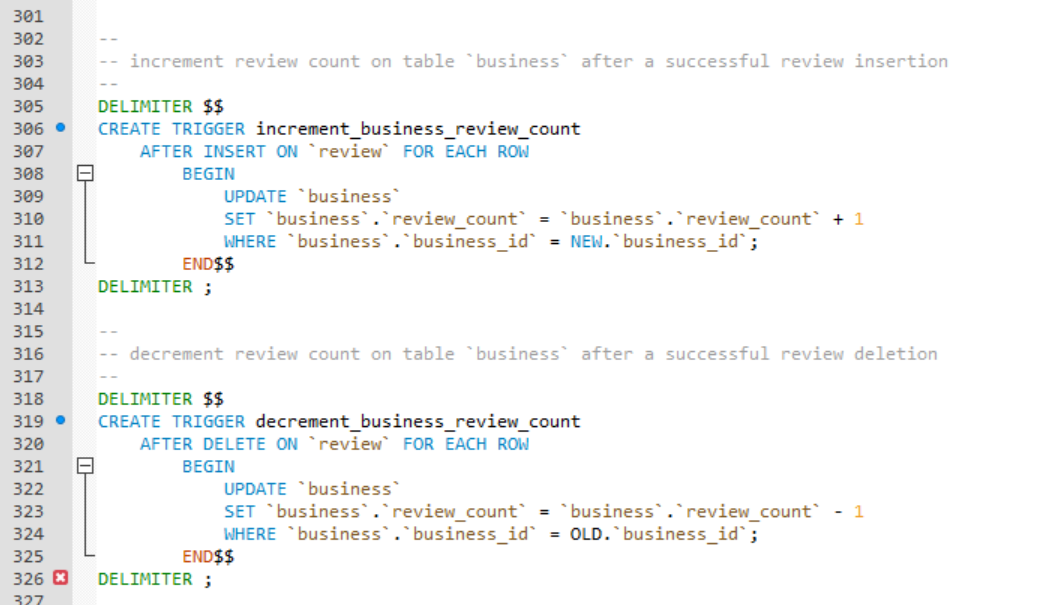
DELETE:

* Admin has right to delete businesses.

Triggers:

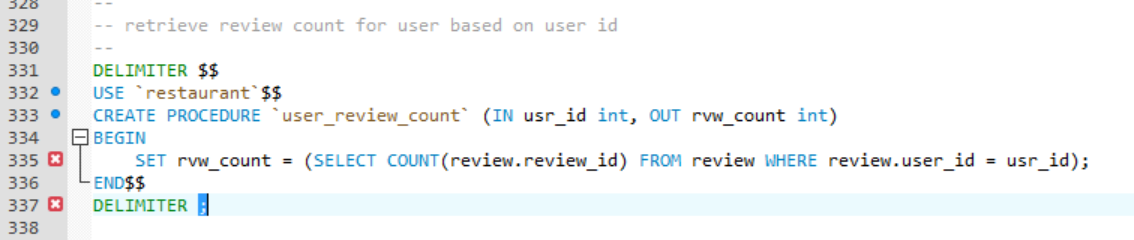
A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server. DML triggers execute when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

Since firing a Trigger and executing the triggering event is an expensive process, we have used very few trigger and tried to use constraints instead of trigger while implementing wherever we can. In this system, we have created trigger to keep a count on the no. of reviews. Thus, a trigger is written to increment or decrement the review count when any INSERT/ DELETE/ UPDATE operations are to be made in reviews.



Stored Procedure:

Stored procedure implemented return the number of posts written by a user given the user\_id.



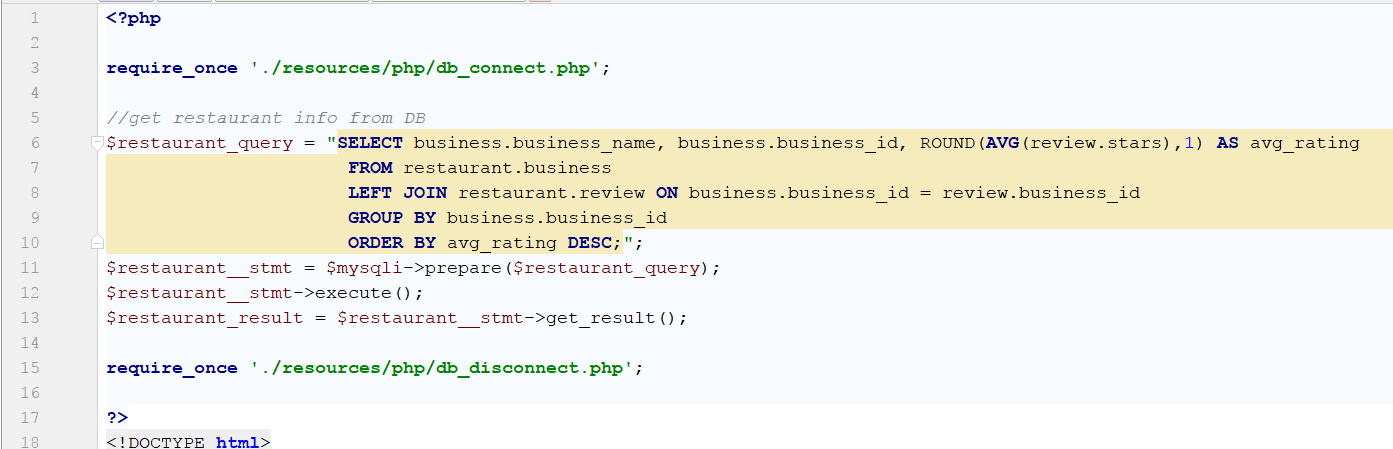
Query Optimization Techniques:

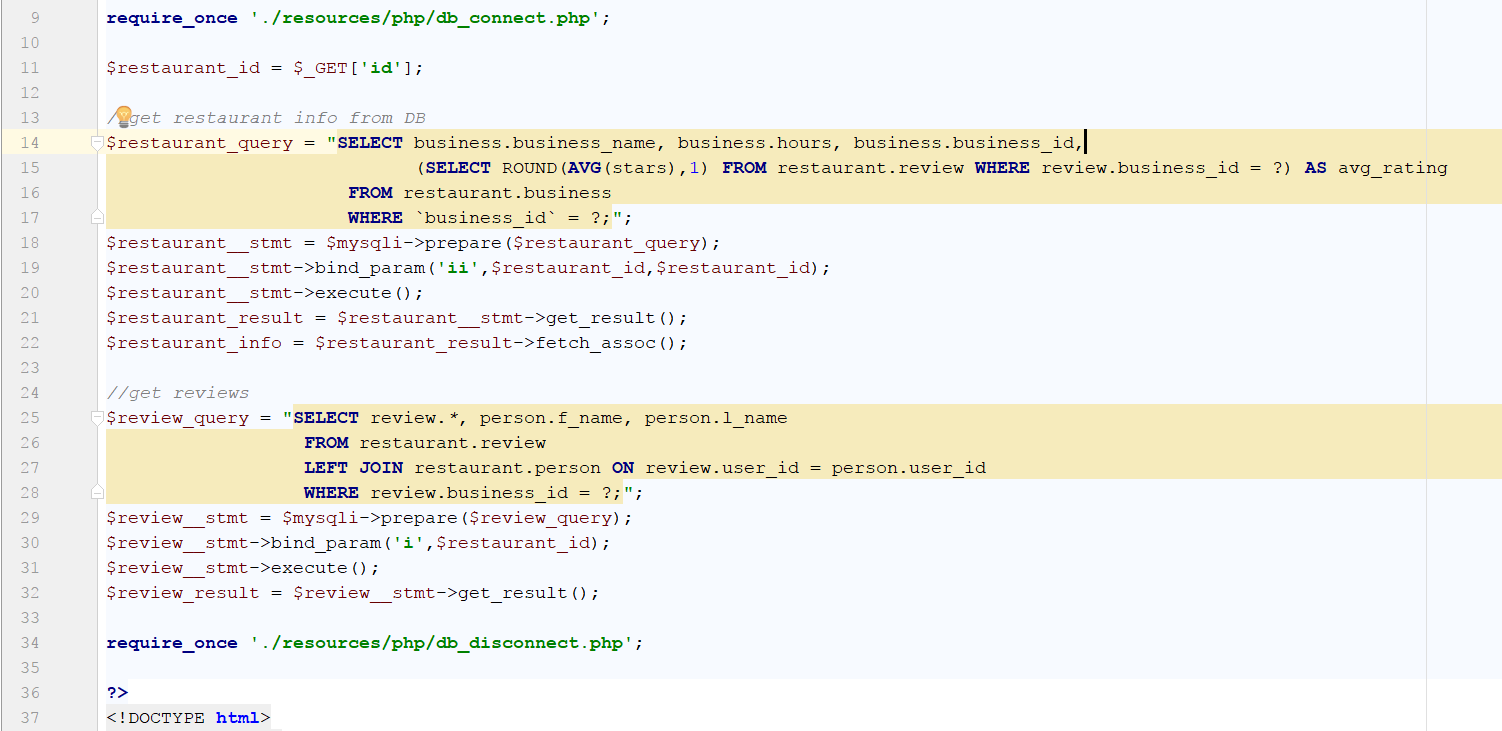
INDEXES:

Since every primary key is already an index along with unique columns and given our tables are not that complex, we didn’t feel the need of putting additional indexing techniques.

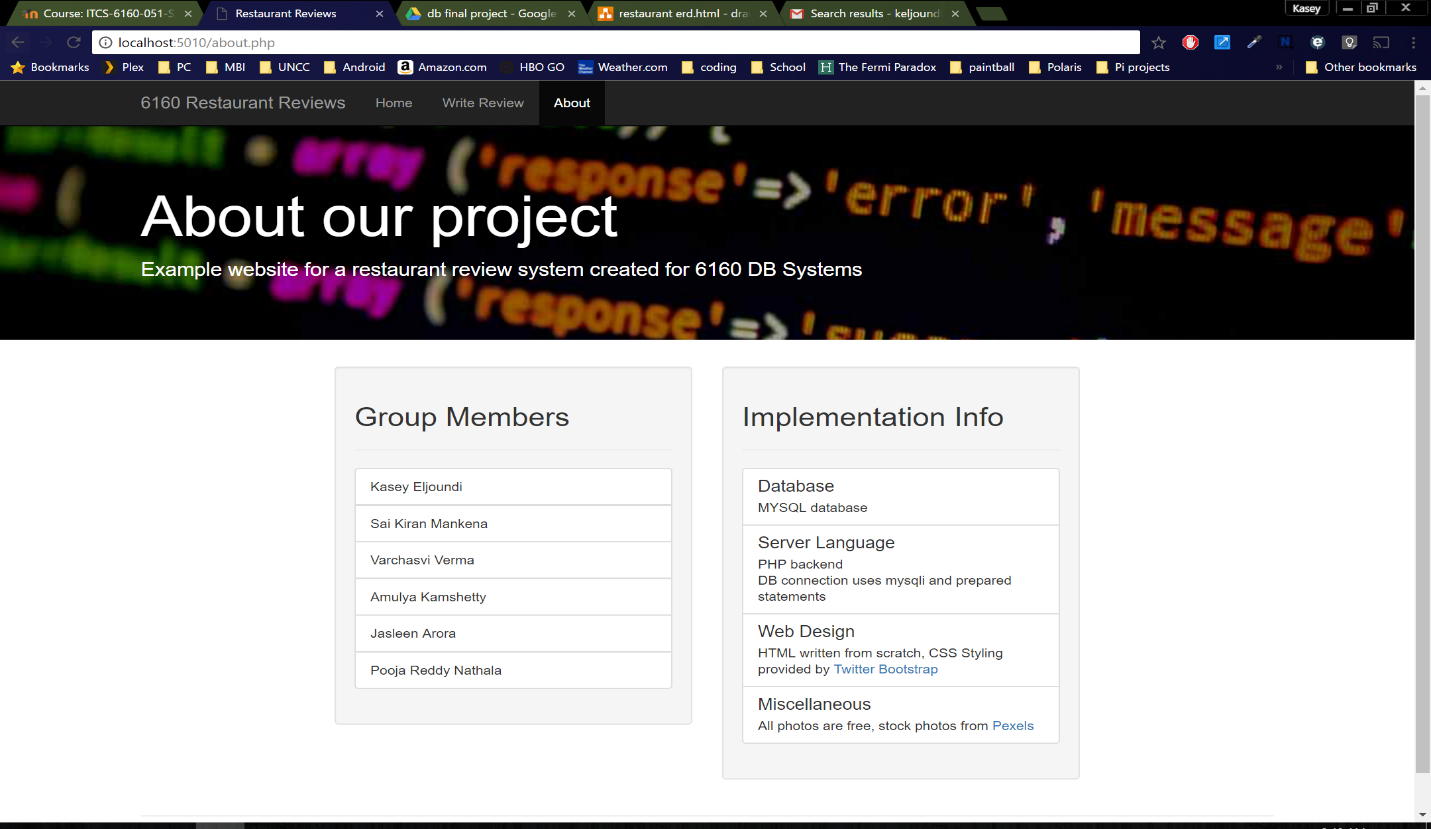
VIEWS:

We have implemented views to fetch related data of a restaurant from different tables (Business, Review and Person table) by joining them virtually. Below are the screenshots of implementing the joins in our UI:

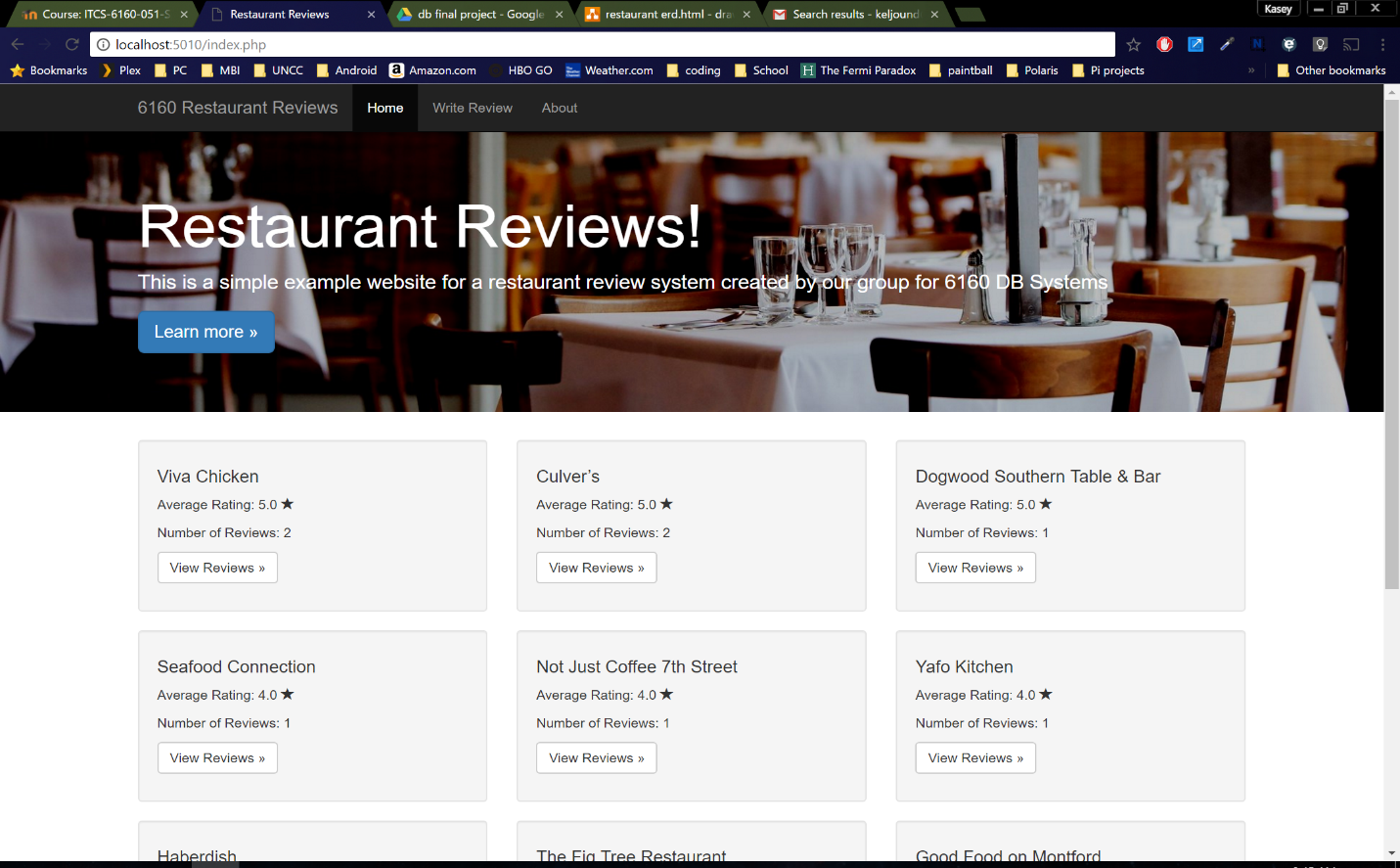




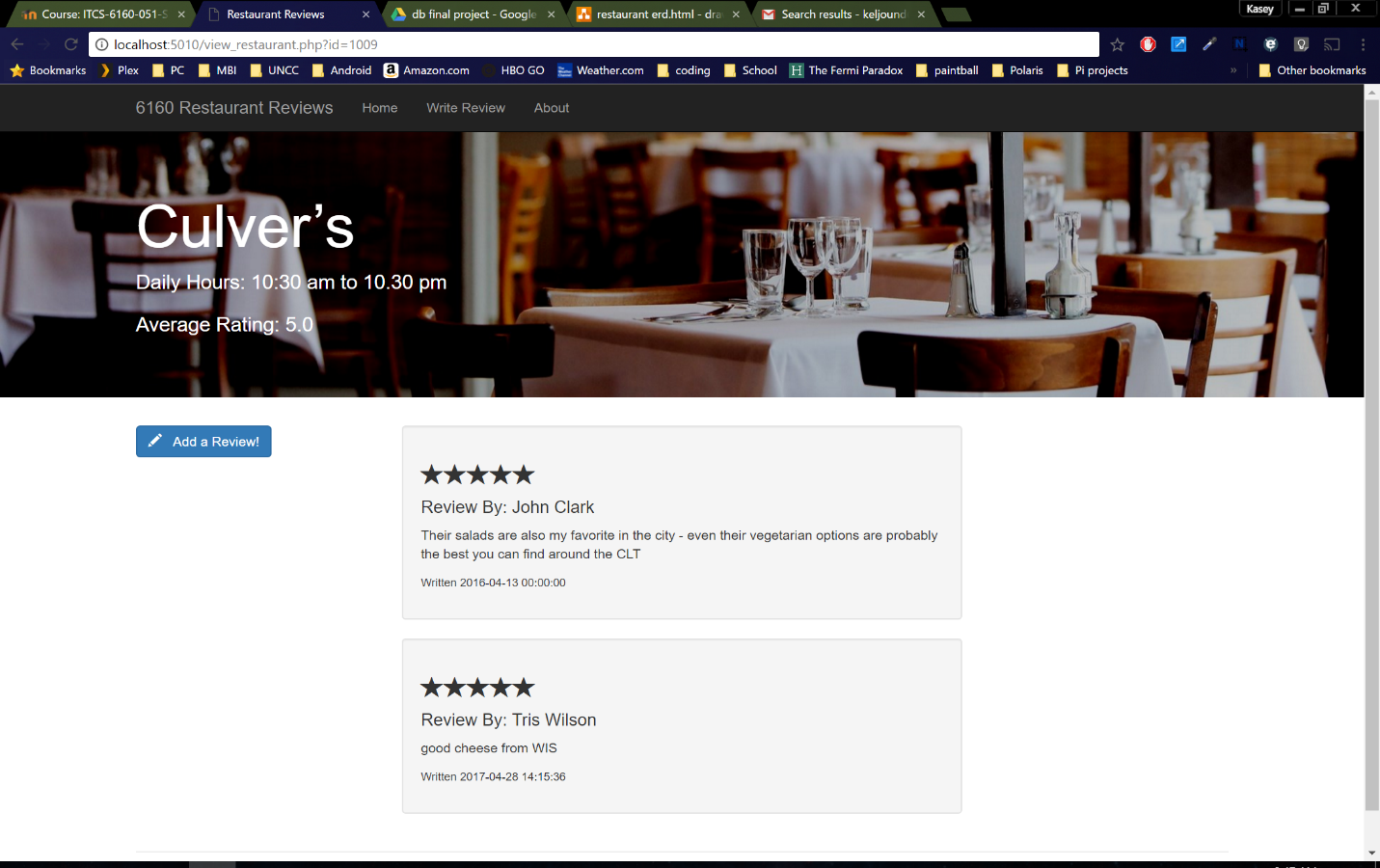
UI Snapshots:



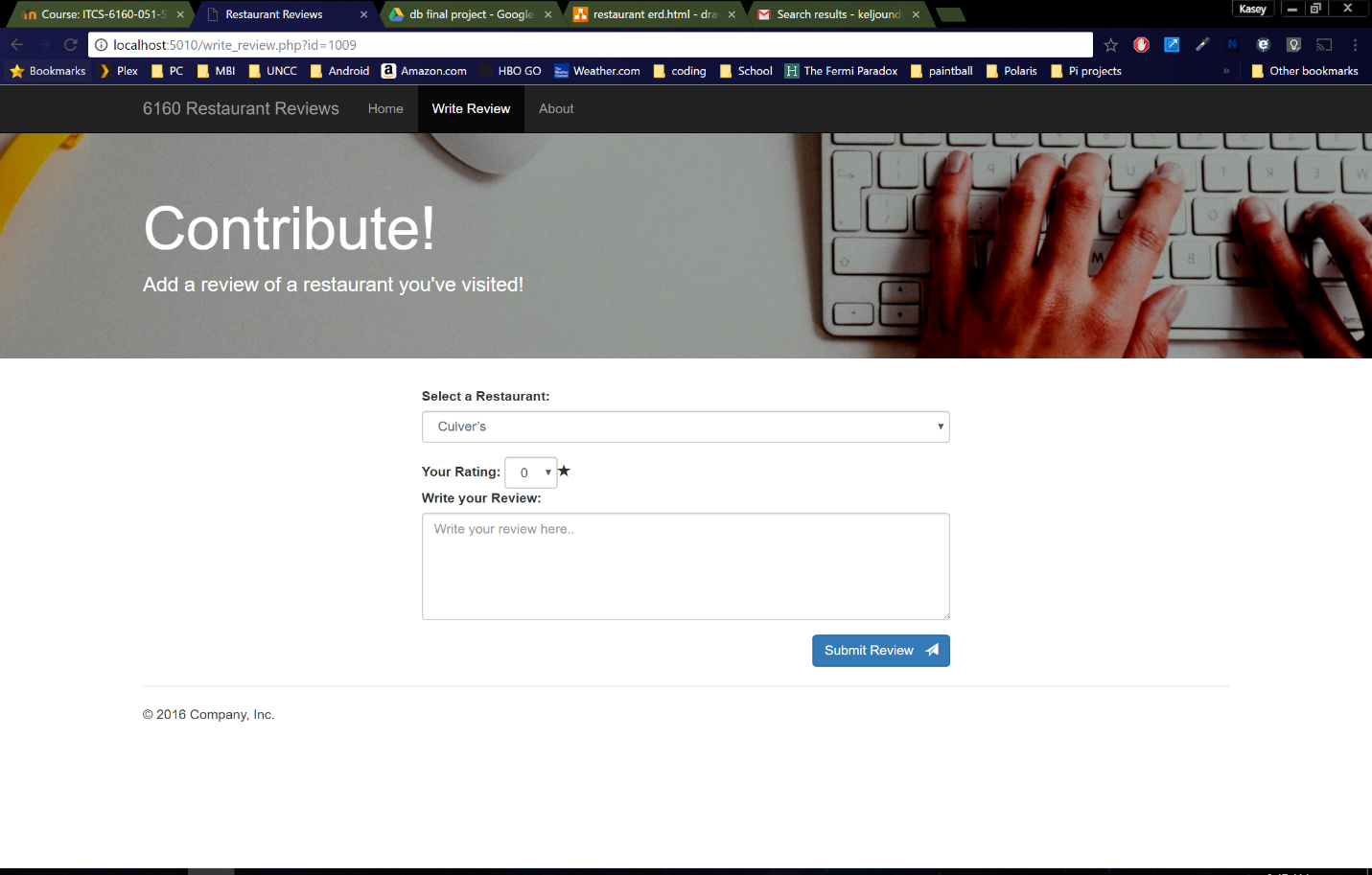
Home Page:



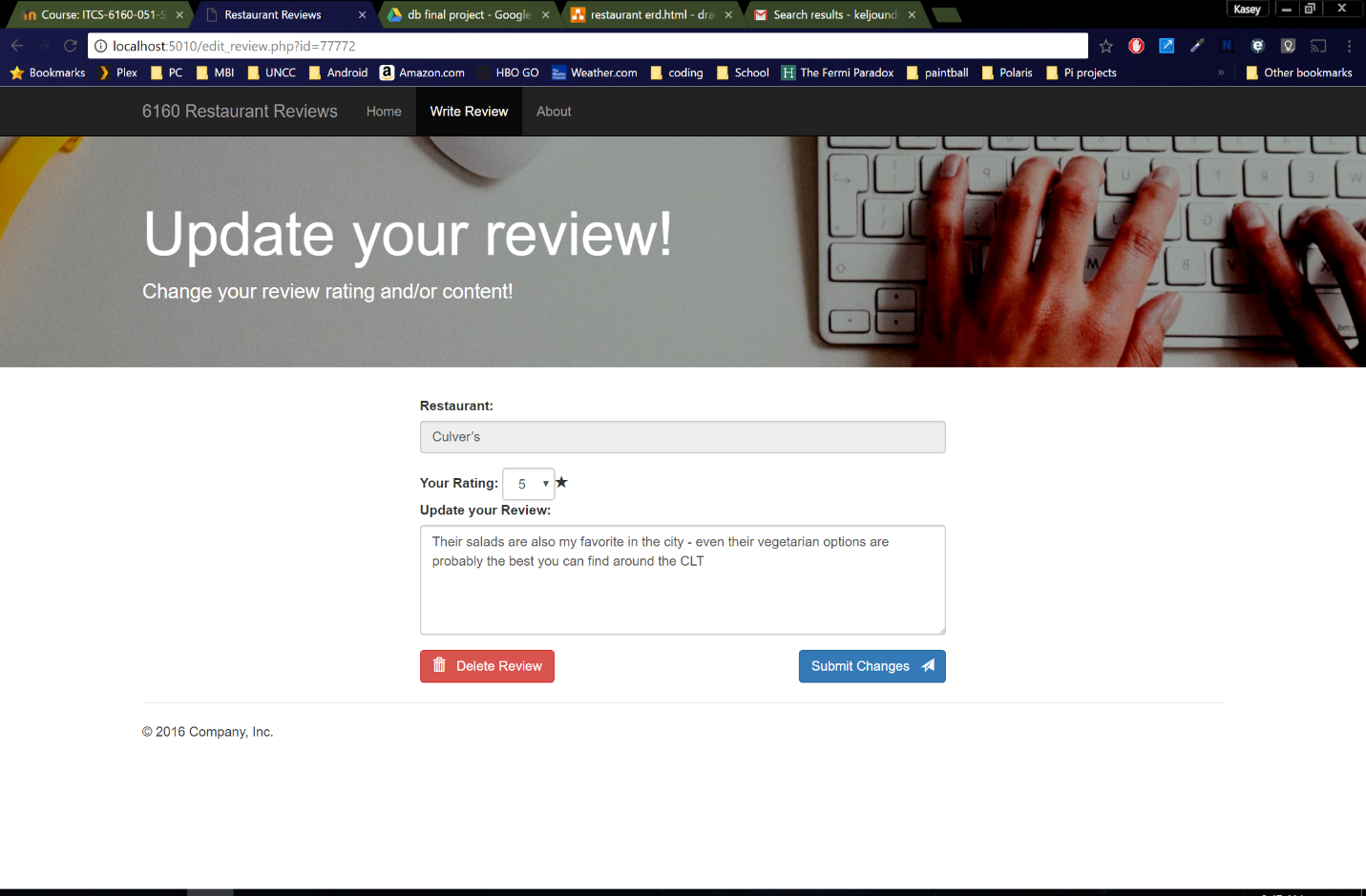
Restaurant Page:



Add review Page:



Update Review Page:



Future Scope:

* Additional features:
  + Restaurants can add feedback on user review.
  + User can make a reservation through the application.
  + Listing which of the restaurants provide order delivery or order pickup.
* Security: Considering sensitive personal data related of the User, the database would implement measures that make the whole system secure.
* Mobile App: The current system can be viewed on any Web Browser. But a stand-alone Mobile application would help access to the database system from mobile in a more efficient way.

Team Leader and Other Positions:

|  |  |  |
| --- | --- | --- |
| S.No. | Name | Role |
| 1 | Kasey Eljoundi | Lead web Developer |
| 2 | Sai Kiran Mankena | Lead SQL Programmer |
| 3 | Varchasvi Verma | Project Leader, Co-Report Writer |
| 4 | Amulya Kamshetty | Project Co-Leader, Co-Report Writer |
| 5 | Jasleen Arora | Co-lead SQL Programmer |
| 6 | Pooja Reddy Nathala | Project Supervisor |

Team Contact Information:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No. | Name | Email Address | Phone No. |
| 1 | Kasey Eljoundi | keljound@uncc.edu | 704-819-6263 |
| 2 | Sai Kiran Mankena | smankena@uncc.edu | 704-756-3175 |
| 3 | Varchasvi Verma | vverma4@uncc.edu | 602-702-3889 |
| 4 | Amulya Kamshetty | akamshet@uncc.edu | 470-685-6557 |
| 5 | Jasleen Arora | jarora2@uncc.edu | 980-585-6185 |
| 6 | Pooja Reddy Nathala | pnathala@uncc.edu | 980-345-9321 |

Method of Communication & File sharing medium:

* Method of Communication: WhatsApp Group, Email
* File sharing medium: Google Drive

Project Plan:

|  |  |
| --- | --- |
| Tasks | Date |
| EERD creation and Decision on technology | March 31, 2017 |
| Project Part I Report Submission | April 2, 2017 |
| Database Implementation into SQL | April 8, 2017 |
| Schema development | April 10, 2017 |
| Project Part II Report Submission | April 16, 2017 |
| GUI development & Integration | April 23, 2017 |
| Testing & Final Checks | April 28, 2017 |
| Presentation | April 29, 2017 |
| Project Part III Report Submission | May 2, 2017 |