

**Group: 2**

**Team Members :**

Travis Seevers

David Chan

Brandon Withington

**URL: <http://flip1.engr.oregonstate.edu:42077/>**

## **Project Report**

### **1. Executive Summary:**

#### **a. Summary of Feedback:**

- Webpage design was bland and didn't fit the project theme
- Couldn't click on the battles displayed in the homepage
- Some words/pictures were not aligned correctly
- Functionality for creating an account/login didn't work
- Newly created battles weren't displayed on the homepage
- Nice layout and navigation design for the project
- Some relationships in the table needed to be clarified/modified to make it more clear
- Not all tables were involved in a query
- Didn't have multi-table queries
- Search functionality didn't work
- No validation checks in place

#### **b. Changes Made:**

- Added validation checks for registering account, creating a battle and trying to access the admin page
- Changed the color them for the webpage to make it more appealing and fitting for the project theme
- Added the search functionality
- Battles on the homepage navigate to their own individual battles
- Newly created battles are also displayed on the homepage
- Cleaned up the relationships between tables in our database by adding more tables to clarify their relationship and remove ambiguity
- Included multi-table queries within the project
- Added the functionality for registering an account and logging in with the registered account
- Included all table in some sort of query

**c. Contribution/Roles:**

- Travis:
  - Implemented the initial server infrastructure, search functionality, login, registration, and account functionality, and base website infrastructure. As well as contributed to all reports.
- Brandon:
  - Implemented functionality for adding, inserting and deleting comments from battles. Worked on CSS and general layout stylization.
- David:
  - Functionality for adding and deleting admins, starting a battle, displaying and selecting homepage battles. Contributed to all of the reports.

**2. Project and Database Outlines:**

## Overview

Creature Colosseum is a web application at which users can specify which particular groups of critters go against another group of critters and vote the victor via a voting system. Our users can specify both the number and size of the battling critters. Other users will be able to see the battle and vote on which group would win. Users would have to create an account to vote and or submit battles to the web application. This application would seek to entertain people and serve as a time filler.

## Rules & Requirements

**Requirements:**

- Users must log on to interact with the site
  - Interactions:
    - Voting
    - Creating/submitting a battle
    - Creating a comment that appears in the database
    - Deleting comments from Battles
  - Account must have a unique email and screen name, and a secure password
- Users can search for particular battles
- Each critter in each battle should have a size.

- Users can vote the victor of a battle, which will be stored as a continuous tally of who is winning currently, with no time frame for the end of the battle.
- Users do not need to log on to see who is winning in a battle, however they cannot vote if not logged in.

### **“Business” Rules :**

- Passwords will be hashed and stored as integers. At no time will a plaintext password be stored on the servers
- One account gets one vote per battle
- A user can both create battles and comments
- One account can leave as many comments per battle as they wish
- Administrators are identified through their unique screen names, which are checked when trying to access admin features of the site.
- Each battle must have only 2 critters. The quantity, size, and species fields must be populated, but the fields don't have to be unique.
- Users can only like or dislike a comment or battle once.

### **Entities and Attributes :**

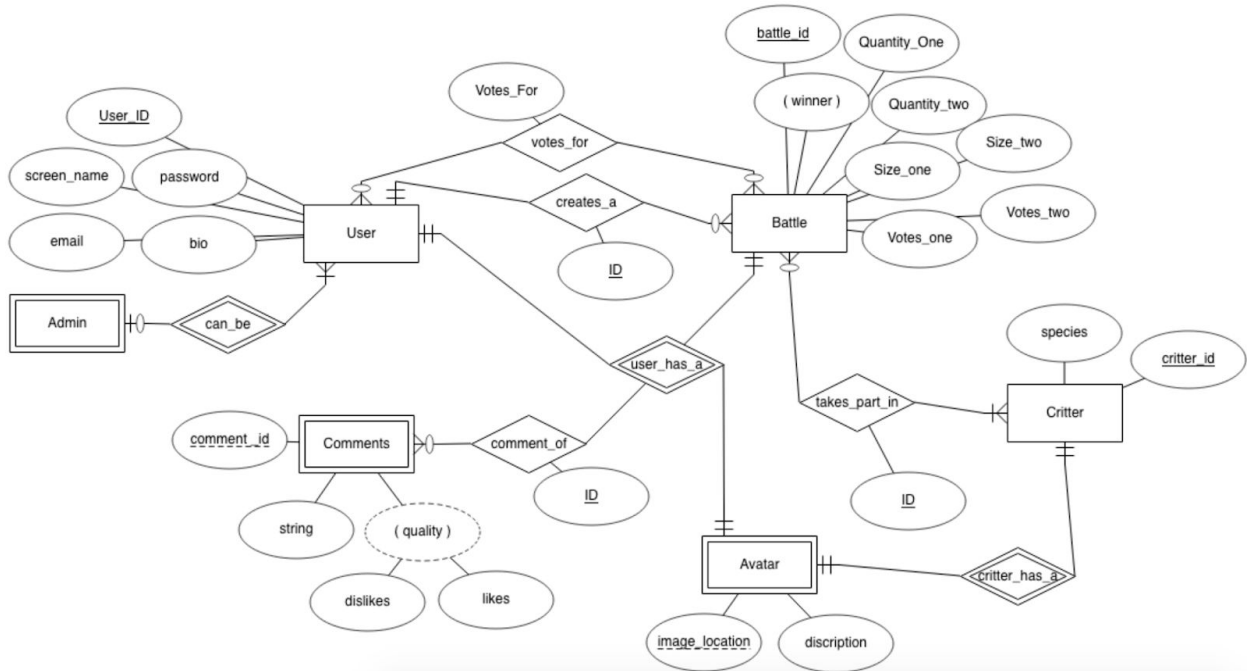
- User: The people who use our web application
  - Email -- This will be unique to the user account, disallowing people to make multiple accounts with one email.
  - Screen name -- This will be unique to the user account.
  - Password to log on to the web application -- composed of numbers and letters.
  - Bio -- user's description that will appear on their profile.
  - User\_id -- Unique user identifier
- Critter: A creature that a user will specify to participate in a battle
  - Species -- What does this critter or user look like?
  - Critter\_id -- Unique critter identifier.
- Battle: Store the information about the two critter battling, their sizes, quantities, species, and respected votes
  - Unique battle identifier to help identify and search for particular battles.
  - Stores the winner between two critters. It keeps track of the vote of both critters.
  - Critter size -- critter size will be categorized as a string value.
  - Critter quantity
    - will be a string value that indicates the number of a particular critter
  - Made\_by: contains the user id that created the battle
- Comments: Allow the users to voice their opinions on battles. Contains:
  - String of the user's comment
  - Likes: Stores the likes of the comment

- Dislikes: Stores the dislikes of the comment
- Comment Id : Unique comment identifier
- Administrator/Moderator: the superusers of the site, have the power to remove admins and add admins to the sites.
  - Admin id: Stores the user id
- Avatar: Image displayed for either a user or a critter. Contains:
  - URL of where the image is being hosted
  - Text description of what the image is/name

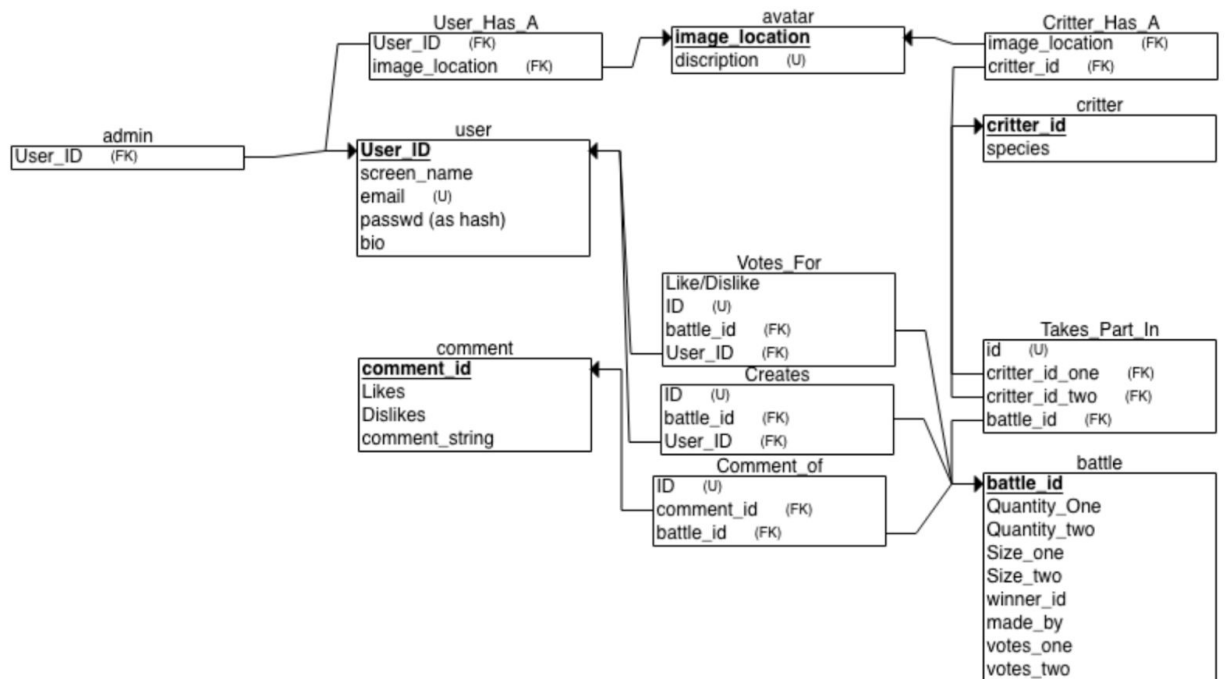
#### **Potential Relationships (some with attributes):**

- **Takes\_part\_in:** Must have two critters. Many to many relationship between two critters and a battle.
- **Votes\_for :** many-to-many relationship between the user and battle. Many users vote on many battles.
- **Can\_be :** many to one relationship between User and Admin. Not all users are administrators, but all administrators are users. Only a few are chosen.
- **Comment\_of:** one-to-many relationship between Battle and Comments.
- **Critter\_has\_a :** one-to-one relationship between critter and avatar. A Critter must have an avatar and they can only have one avatar.
- **User\_has\_a:** one-to-one relationship between user and avatar. A User must have an avatar and they can only have one avatar.
- **Creates\_a :** one to an optional many. This Describes the relationship between the User and Battle as a User can create zero or many battles.
- **Comment\_of:** optional many to optional one relationship between Comments and Battle since a Battle may have 0 or more comments but a Comment may only be in one Battle.

### 3. ER Diagram



### 4. Schema




## 5. UI Screenshots with Informative Titles and SQL Queries


### CREATE/INSERT/ADD NEW Start a Battle Page

**Critter Colosseum**

Home Account View Battles Start a Battle Battle Search... Search!



Enter the critter's name  
Enter your critter's size  
Enter how many critter there  
Supply an animal image link



Enter the critter's name  
Enter your critter's size  
Enter how many critter there  
Supply an animal image link

Publish!

```
"INSERT INTO Battle (Quantity_one, Quantity_two, Size_one, Size_two)
VALUES(?,?,?,?)"
```

```
"INSERT INTO Critter (Species) VALUES(?)"
```

```
INSERT INTO Takes_part_in (critter_one, critter_two, battle) VALUES(?,?,?)"
```

```
"INSERT INTO Critter_has_a (critter, av_id) VALUES (?,?)"
```

```
"INSERT INTO Creates_a (battle, usr) VALUES (?,?)"
```

## READ/DISPLAY Battles page



"SELECT T.battle as id, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_one) AS Name1, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_two) AS Name2 FROM Takes\_part\_in T ORDER BY T.id DESC"

"SELECT T.battle as id, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_one) AS Name1, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_two) AS Name2 FROM Takes\_part\_in T"

## UPDATE/INSERT/ADD NEW/READ View Battles Page

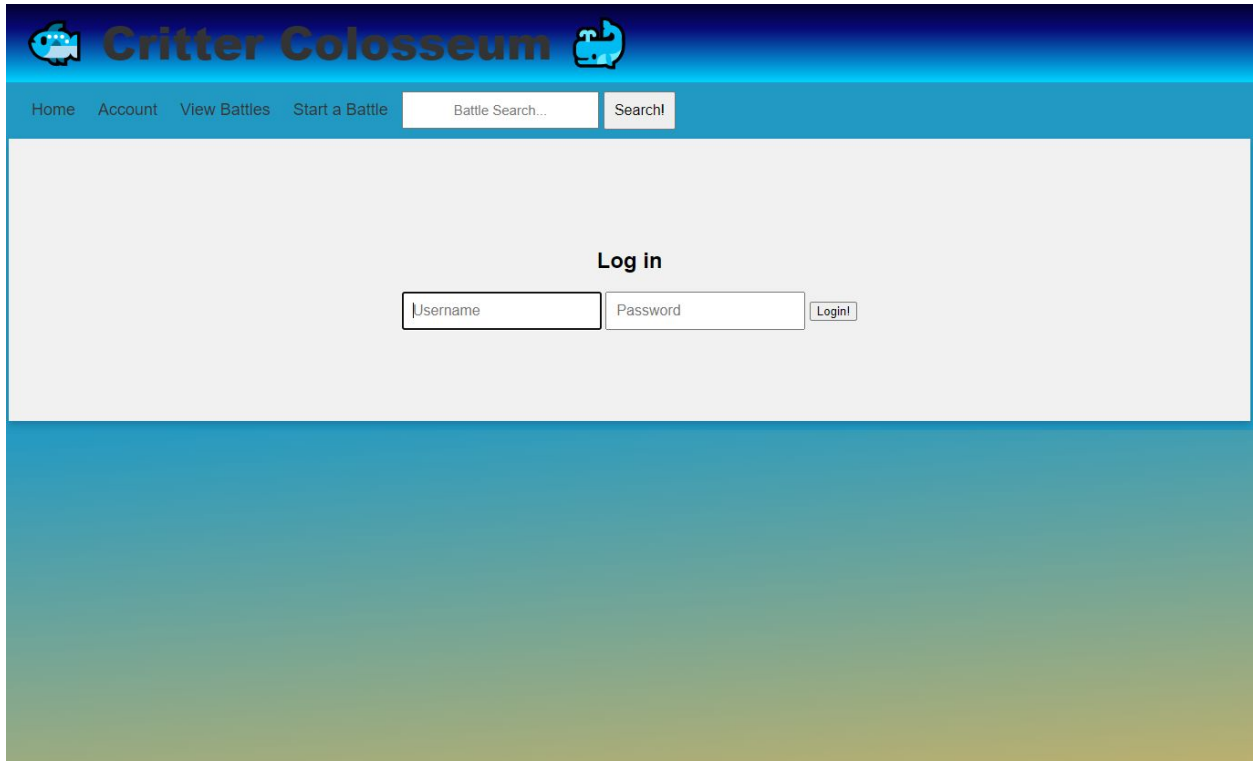
The screenshot shows the 'Critter Colosseum' web application. At the top, there's a navigation bar with links: Home, Account, View Battles, Start a Battle, and a search bar labeled 'Battle Search...' with a 'Search!' button. Below the navigation bar, the main content area is divided into two columns. The left column features a small image of a salamander and a text box containing: 'Critter one name : Salamander', 'Critter one size : 3 ft', 'Critter one quantity : 2', and 'Votes for critter one :'. Below this is a 'Vote!' button. The right column features a small image of an otter and a text box containing: 'Critter one name : Otter', 'Critter one size : 24 in', 'Critter one quantity : 3', and 'Votes for critter one :'. Below this is a 'Vote!' button. At the bottom of the main content area, there's a 'Comment Section' header. Below this, there's a 'Comment' label, a text input field, and a 'Submit' button. The background of the page is a blue gradient.

"SELECT \* FROM Comment\_of F INNER JOIN Comments C on F.comment\_id = C.comment\_id WHERE F.battle = (SELECT battle\_id FROM Battle ORDER BY battle\_id DESC LIMIT 1)"

"SELECT B.battle\_id, B.Quantity\_one, B.Quantity\_two, B.Size\_one, B.Size\_two, B.votes\_one, B.votes\_two, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_one) AS Name1, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_two) AS Name2 FROM Takes\_part\_in T, Battle B WHERE T.battle = (SELECT D.battle\_id FROM Battle D ORDER BY D.battle\_id DESC LIMIT 1) ORDER BY B.battle\_id DESC LIMIT 1"



## READ Login Page



**Critter Colosseum**

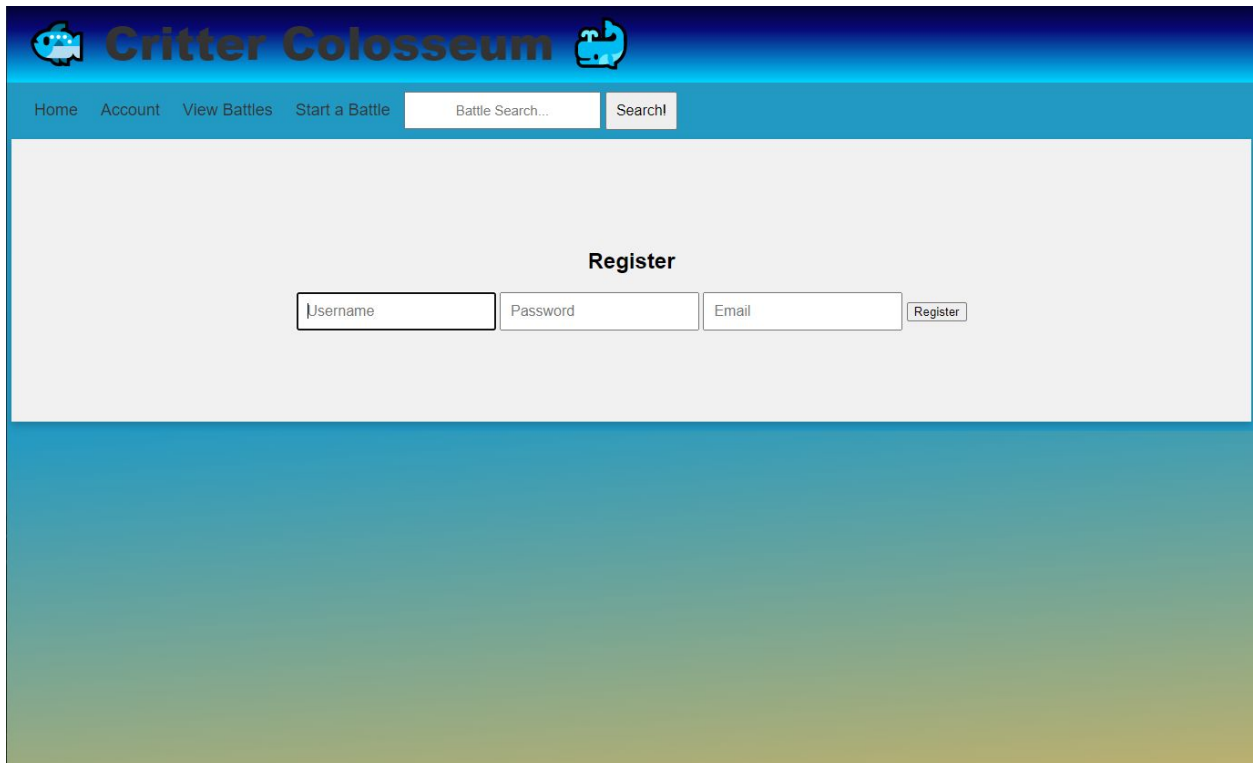
Home Account View Battles Start a Battle Battle Search... Search!

**Log in**

Username Password Login!

"SELECT User\_id FROM Usr WHERE screen\_name=? AND password=?"

## INSERT/ADD NEW/CREATE Register Page



The screenshot shows the 'Register' page of the 'Critter Colosseum' website. The header features the site's logo and navigation links: Home, Account, View Battles, and Start a Battle. A search bar is also present. The main content area is a light gray box with the title 'Register' centered at the top. Below the title are three input fields labeled 'Username', 'Password', and 'Email', followed by a 'Register' button. The background of the page is a blue-to-green gradient.

"INSERT INTO Usr (screen\_name, email, password, bio) VALUES (?, ?, ?, ?)"

"INSERT INTO Avatar (description, image\_location) VALUES (?, ?)"

"INSERT INTO User\_has\_a (usr, av\_id) VALUES (?, ?)"

## DELETE/READ/INSERT/DISPLAY Admin Page



The screenshot shows the 'Admin Page' of the 'Critter Colosseum' application. The page has a blue header with the site logo and name. Below the header is a navigation bar with links: Home, Account, View Battles, Start a Battle, and a search bar labeled 'Battle Search...' with a 'Search!' button. The main content area is divided into two sections. The first section, titled 'Admin List', contains a table with two rows of admin data. Each row has columns for 'Screen Name', 'Email', and a 'Delete' button. The second section, titled 'Add Admin', contains a form with a label 'Screen Name List:', a dropdown menu with 'Wilbur' selected, and a 'Submit' button.

Admin List		
Screen Name: James	Email: james@google.net	Delete
Screen Name: Hamantha	Email: ham@for.face	Delete

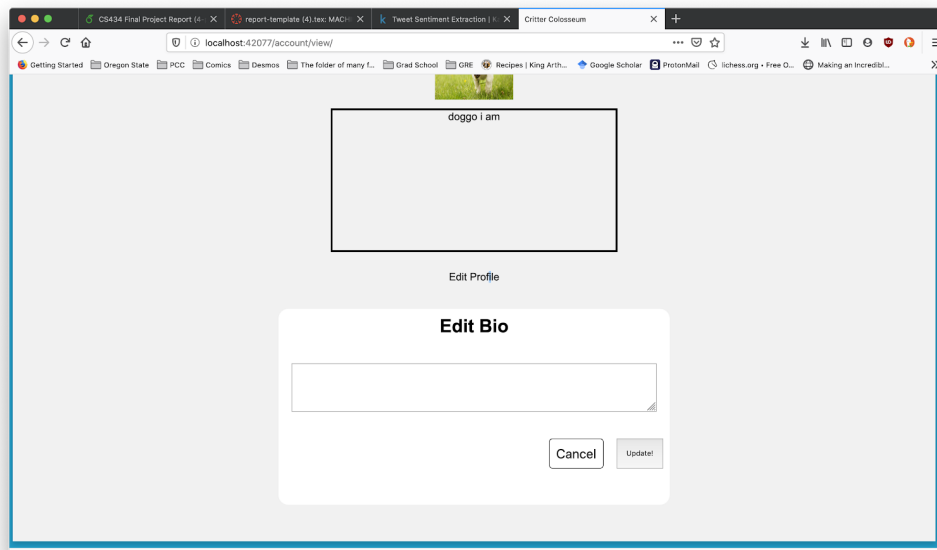
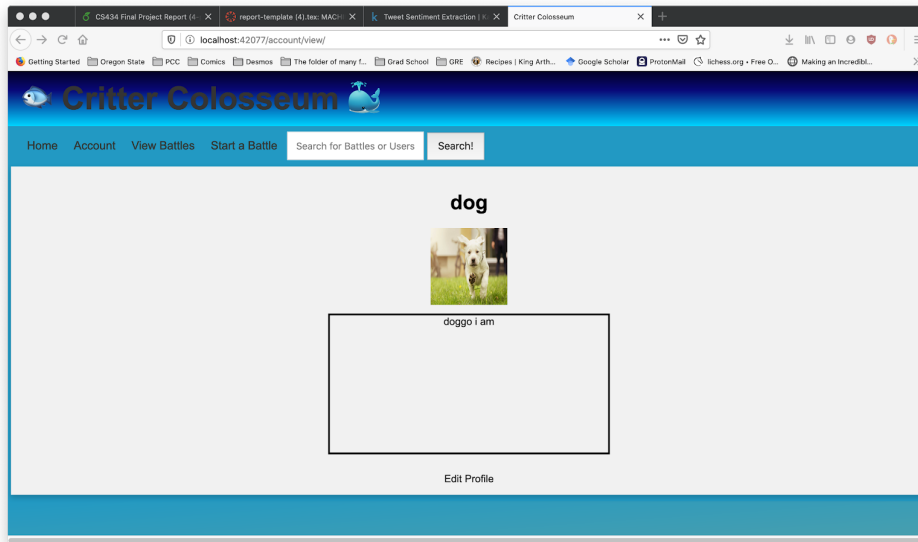
Add Admin	
Screen Name List:	<input type="text" value="Wilbur"/> <input type="button" value="Submit"/>

"SELECT \* FROM Usr U WHERE U.User\_id NOT IN (SELECT admin\_id FROM Admin)"

"SELECT admin\_id FROM Admin WHERE admin\_id = ?"

"SELECT \* FROM Usr INNER JOIN Admin ON admin\_id = User\_id"

## READ/BROWSE/DISPLAY & UPDATE: Account Page

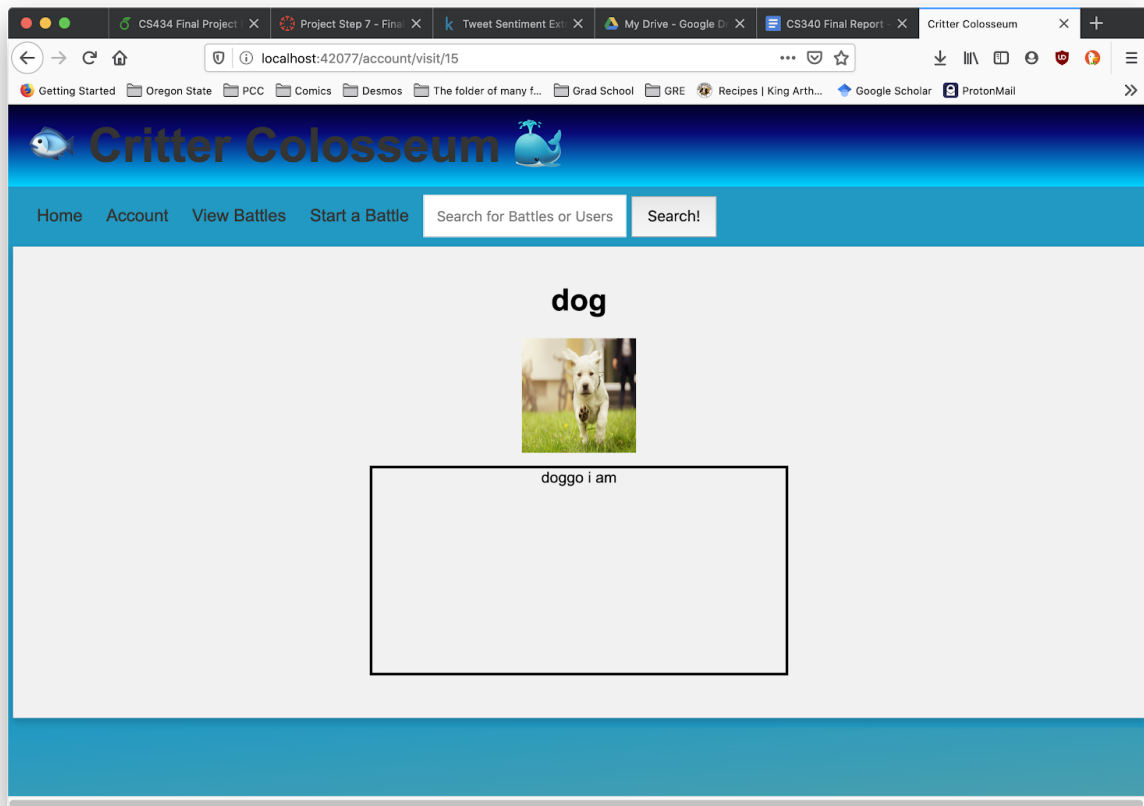


"SELECT screen\_name as sc, bio FROM Usr WHERE User\_id=?"

"SELECT Avatar.image\_location as LNK FROM Avatar JOIN User\_has\_a ON User\_has\_a.av\_id=Avatar.av\_id WHERE User\_has\_a.usr=?"

"UPDATE Usr SET bio=? WHERE User\_id=?"

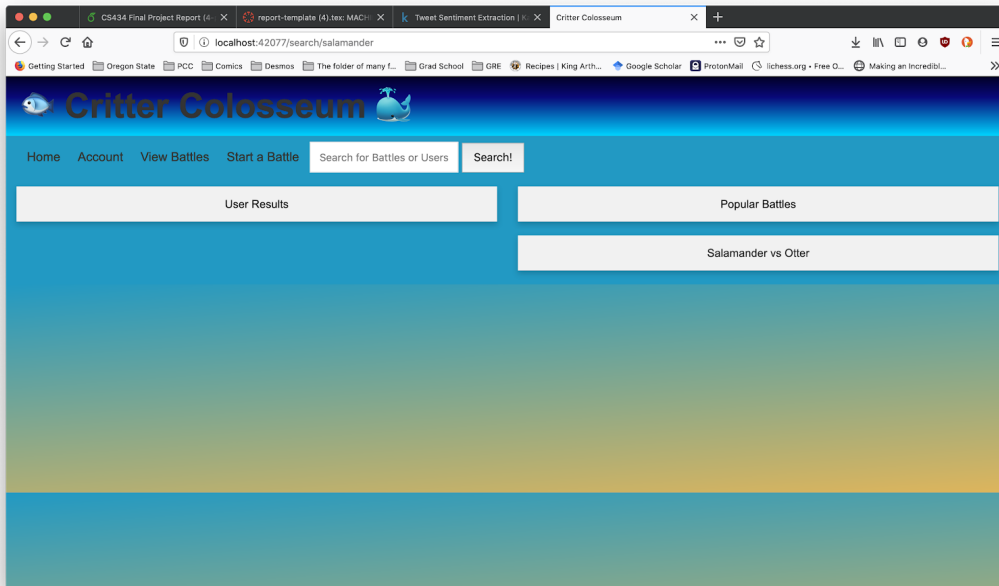
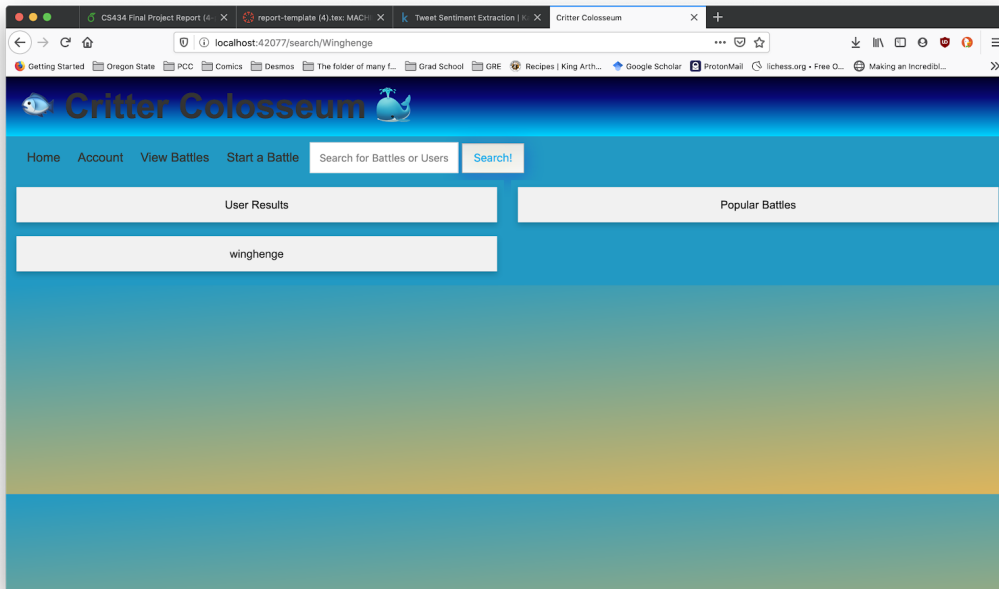
## READ/BROWSE/DISPLAY: View Other Account Page



**"SELECT screen\_name as sc, bio FROM Usr WHERE User\_id=?"**

**"SELECT Avatar.image\_location as LNK FROM Avatar JOIN User\_has\_a ON User\_has\_a.av\_id=Avatar.av\_id WHERE User\_has\_a.usr=?"**

## READ/BROWSE/DISPLAY: Search Page (Can search for a user or a battle critter)



"SELECT User\_id as id, screen\_name as SC FROM Usr"

"SELECT T.battle as id, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_one) AS Name1, (SELECT species FROM Critter C WHERE C.critter\_id = T.critter\_two) AS Name2 FROM Takes\_part\_in T ORDER BY T.id"

## 6. SQL - Triggers/Procedure/Functions/Views

- Trigger for the table Votes\_for:
  - This trigger updates the Battle table depending on the value that was inserted into the 'liked' column for the Votes\_for table. If the value for 'liked' was 0 then the 'votes\_one' value for the corresponding 'battle' will be incremented by 1. Otherwise, if the 'liked' value was 1 then the 'votes\_two' value would be incremented by 1 for the corresponding 'battle'.

### Additional Notes:

Account information for an admin that is already setup in the database/sql code

- Screen name: James
- Password: password123

Account information for a normal user thats already in the database

- Screen name: winghenge
- Password: abcdefg

Note: more users can be added through the registration page