# Jim’s Gyms

**Group 45:**

**Silas Dermer, Kali Pulanco, Matthew Mcmahon, Palmer Snapp**

**web.engr.oregonstate.edu/~mcmahoma/cs340/final_project/index.php**

## 1. Introduction

This application stores information for a fictional chain of rock climbing gyms. It has which gym has what route setters (workers that construct the routes), what routes a gym has, which routes have what holds, what setters have worked on which routes, and details about each of those entities. Employees would use this database to manage their gyms and routes, because they can update what holds are in use, update where a setter is working, and delete routes when they get taken down.

## 2. Detailed Functionality & Requirements

Requirements:

- A list of all gyms can be displayed.
- A list of all routes at a gym can be displayed, including its setter and total hold count.
- A list of the counts of routes of each difficulty can be displayed for a gym.
- A list of all setters employed at a gym can be displayed, including the total gyms they are employed at.
- A list of all routes set by a setter can be displayed, including its gym and total hold count.
- A list of all routes set by a setter at a specific gym can be displayed, including its total hold count.
- A list of statistics for each gym can be displayed, only showing a gym’s id, number of routes, and average difficulty of its routes.
- The user can add a new gym.
- The user can change a setter’s name.
- The user can delete a gym.
- The user can delete all routes at a gym that were created on a specific date.

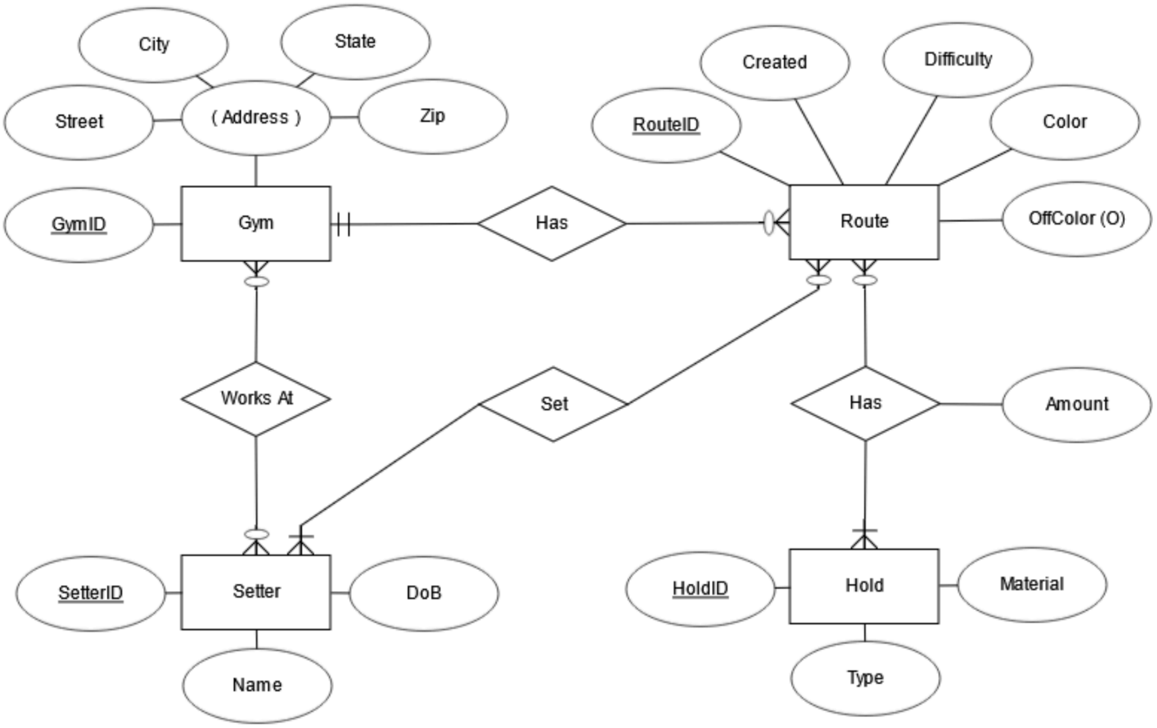
Business Rules:

- A Gym must at least have 20 routes and no more than 200.
- A Gym must have at least one route of each difficulty (up to V10).
- A Route must at least have 2 holds (types) and no more than 30 holds (types).
- A Setter can change a Route more than once (the specific holds in a route).

## 3. Database Design

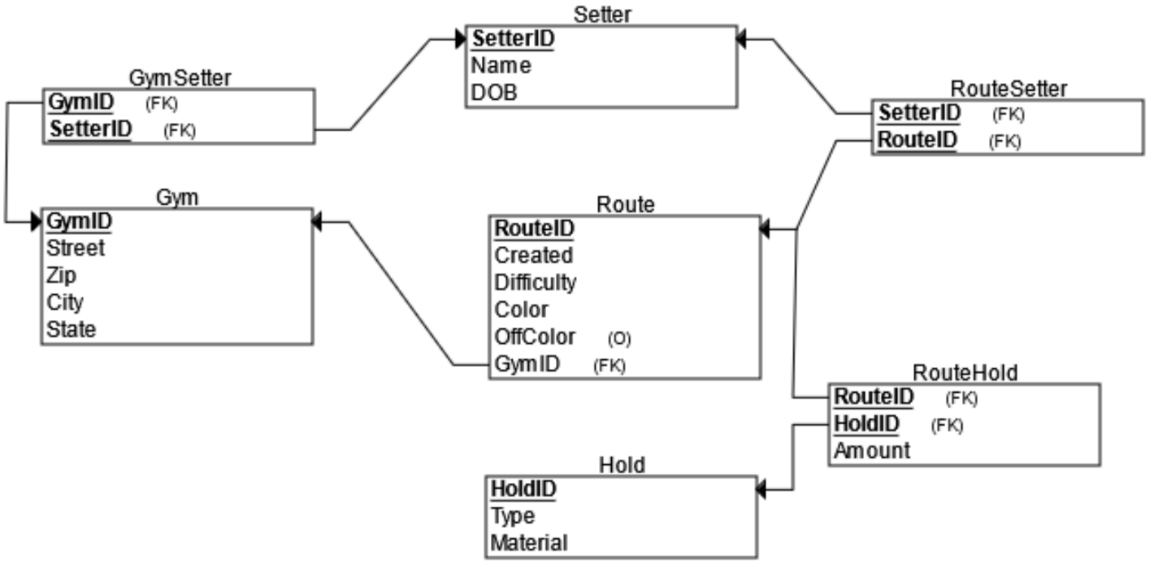
- ***ER Diagram of Database – ERD Plus***

  This diagram should capture all of the entities and relationships of your database.



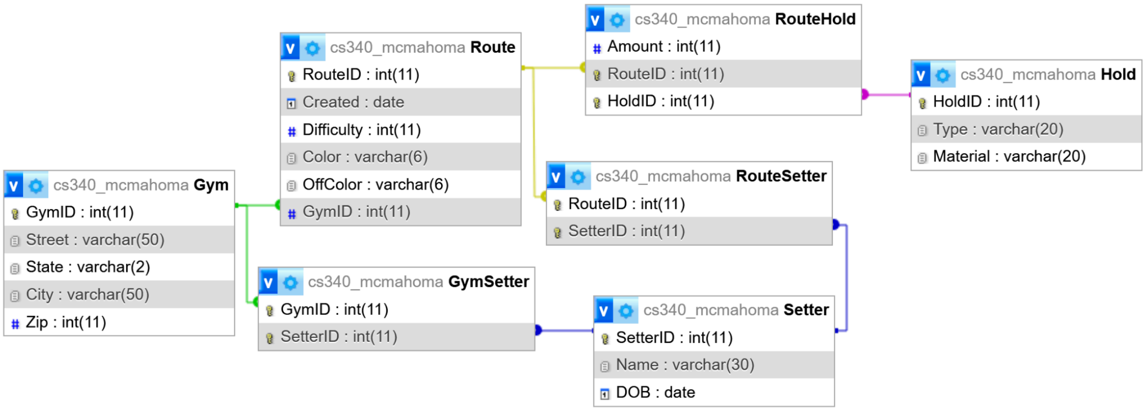
- ***Relation Schema - ERDPlus***

  Show relations, attributes, functional dependencies, primary and foreign keys.



- ***Database Structure - phpMyAdmin***

  Use the Designer tool to show the schema of the tables in your database. This should include the primary keys, foreign keys and types of all attributes.



cs340_mcmahoma **Gym**
GymID : int(11)
Street : varchar(50)
State : varchar(2)
City : varchar(50)
Zip : int(11)

cs340_mcmahoma **Route**
RouteID : int(11)
Created : date
Difficulty : int(11)
Color : varchar(6)
OffColor : varchar(6)
GymID : int(11)

cs340_mcmahoma **RouteHold**
Amount : int(11)
RouteID : int(11)
HoldID : int(11)

cs340_mcmahoma **Hold**
HoldID : int(11)
Type : varchar(20)
Material : varchar(20)

cs340_mcmahoma **RouteSetter**
RouteID : int(11)
SetterID : int(11)

cs340_mcmahoma **GymSetter**
GymID : int(11)
SetterID : int(11)

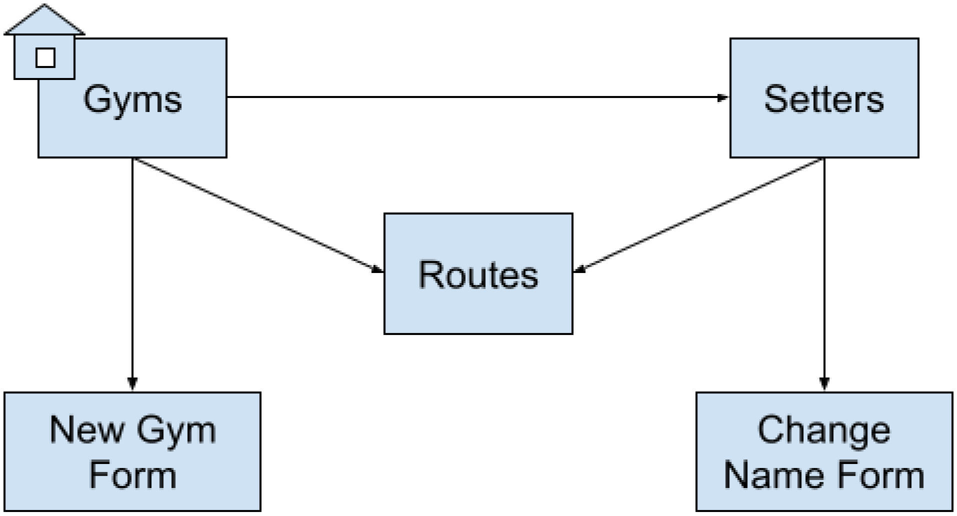
cs340_mcmahoma **Setter**
SetterID : int(11)
Name : varchar(30)
DOB : date

## 4. Website Design

Discuss the design of your website.

- ***Website Layout***

  This diagram should briefly describe the webpages in your website and show how they are connected.

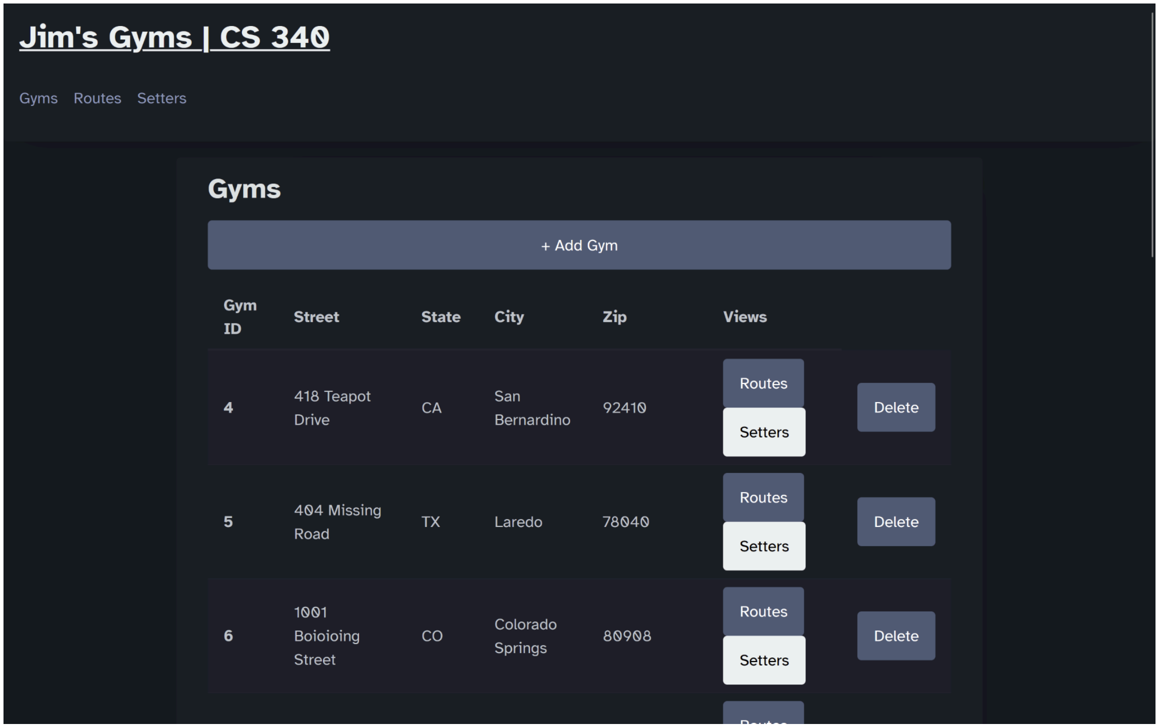


- ***User Interface & Instructions***

  Describe the functionalities of your system in the form of screenshots or wireframes of your user interfaces.

  o Describe the purpose of each page/form/etc.
  o Where and how can the user perform each of the CRUD operations.
  o Provide any instructions needed to use the website

Main page:
Includes directs to Gyms, Routes, and Setters pages in the header.
Gyms list includes Add Gym button, redirecting to the add new gym form.
Each gym has its information and buttons for viewing the gyms Routes and the Setters at that gym. There is also a Delete button to remove the gym from the database.

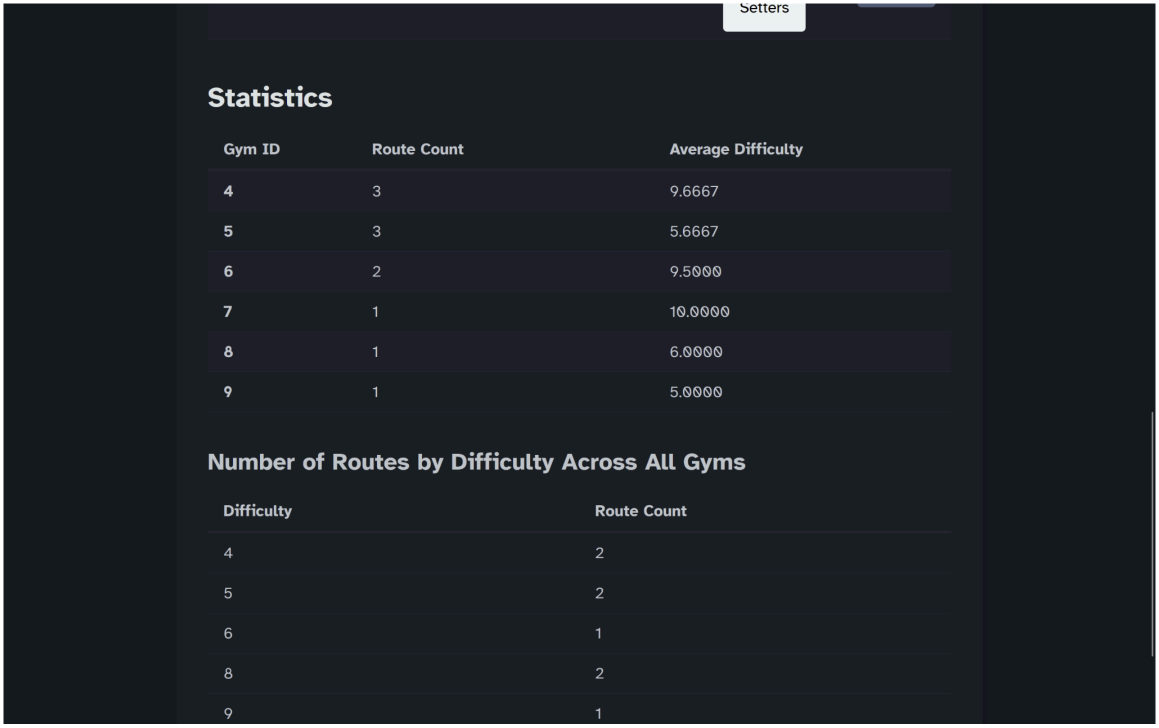


Jim's Gyms | CS 340

Gyms

| Gym ID | Street | State | City | Zip | Views | |
|---|---|---|---|---|---|---|
| 4 | 418 Teapot Drive | CA | San Bernardino | 92410 | Routes<br>Setters | Delete |
| 5 | 404 Missing Road | TX | Laredo | 78040 | Routes<br>Setters | Delete |
| 6 | 1001 Boioioing Street | CO | Colorado Springs | 80908 | Routes<br>Setters | Delete |

Below the gyms list, there is a table of gym statistics which includes information about the number of routes each gym has, and the average difficulty of all the routes the gym has.
Below that, the table with the counts of the number of routes counted by the difficulty over all the gyms.



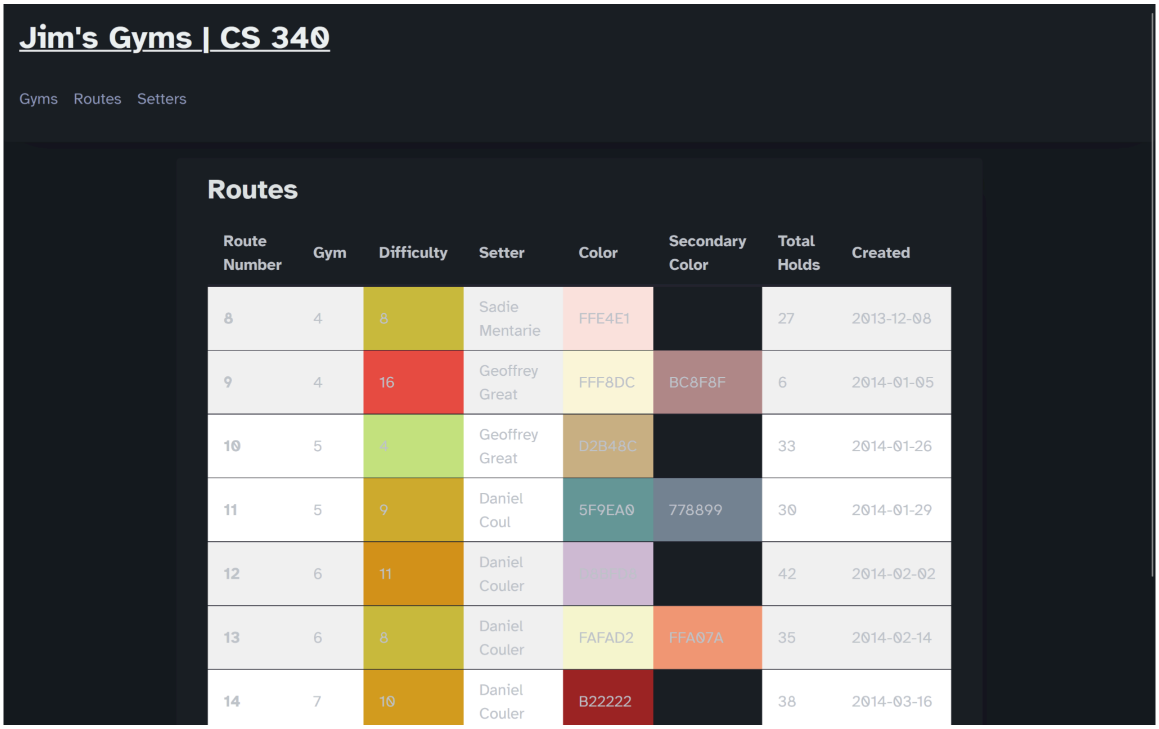
## Statistics

| Gym ID | Route Count | Average Difficulty |
|---|---|---|
| **4** | 3 | 9.6667 |
| **5** | 3 | 5.6667 |
| **6** | 2 | 9.5000 |
| **7** | 1 | 10.0000 |
| **8** | 1 | 6.0000 |
| **9** | 1 | 5.0000 |

## Number of Routes by Difficulty Across All Gyms

| Difficulty | Route Count |
|---|---|
| 4 | 2 |
| 5 | 2 |
| 6 | 1 |
| 8 | 2 |
| 9 | 1 |

Routes:
On this page, the header includes the same redirects as the main page.
There is also a table of routes which includes route information.



# Jim's Gyms | CS 340

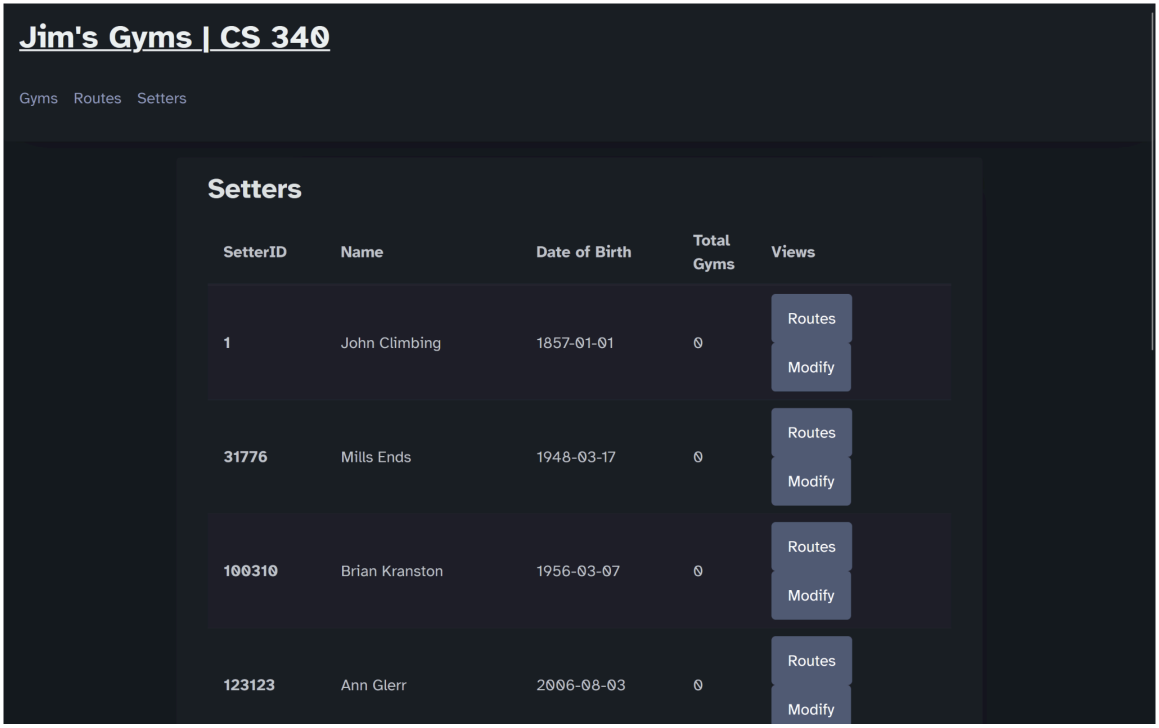
## Routes

| Route Number | Gym | Difficulty | Setter | Color | Secondary Color | Total Holds | Created |
|---|---|---|---|---|---|---|---|
| 8 | 4 | 8 | Sadie Mentarie | FFE4E1 | | 27 | 2013-12-08 |
| 9 | 4 | 16 | Geoffrey Great | FFF8DC | BC8F8F | 6 | 2014-01-05 |
| 10 | 5 | 4 | Geoffrey Great | D2B48C | | 33 | 2014-01-26 |
| 11 | 5 | 9 | Daniel Coul | 5F9EA0 | 778899 | 30 | 2014-01-29 |
| 12 | 6 | 11 | Daniel Couler | D8BFD8 | | 42 | 2014-02-02 |
| 13 | 6 | 8 | Daniel Couler | FAFAD2 | FFA07A | 35 | 2014-02-14 |
| 14 | 7 | 10 | Daniel Couler | B22222 | | 38 | 2014-03-16 |

Setters:

On this page, there is a table of each of the setter's information, as well as a Routes and Modify button.

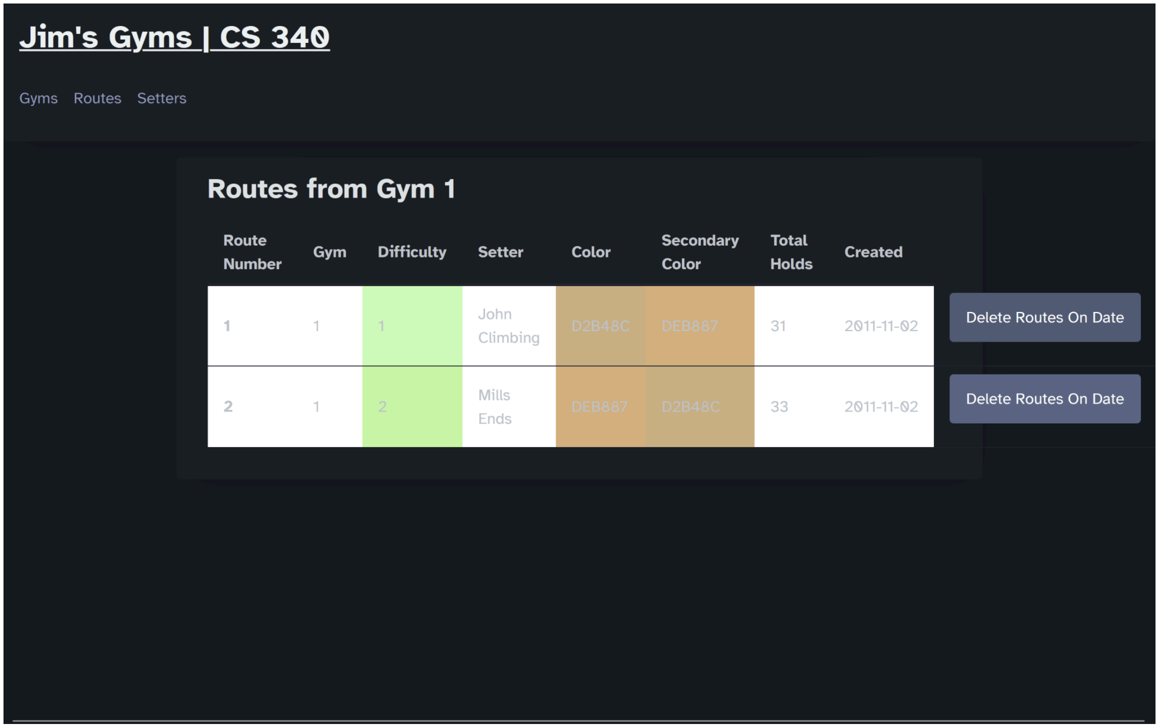
The Routes button redirects to a table of routes created by the setter, and the Modify button redirects to the name change form.



Setters

| SetterID | Name | Date of Birth | Total Gyms | Views |
|---|---|---|---|---|
| 1 | John Climbing | 1857-01-01 | 0 | Routes<br>Modify |
| 31776 | Mills Ends | 1948-03-17 | 0 | Routes<br>Modify |
| 100310 | Brian Kranston | 1956-03-07 | 0 | Routes<br>Modify |
| 123123 | Ann Glerr | 2006-08-03 | 0 | Routes<br>Modify |

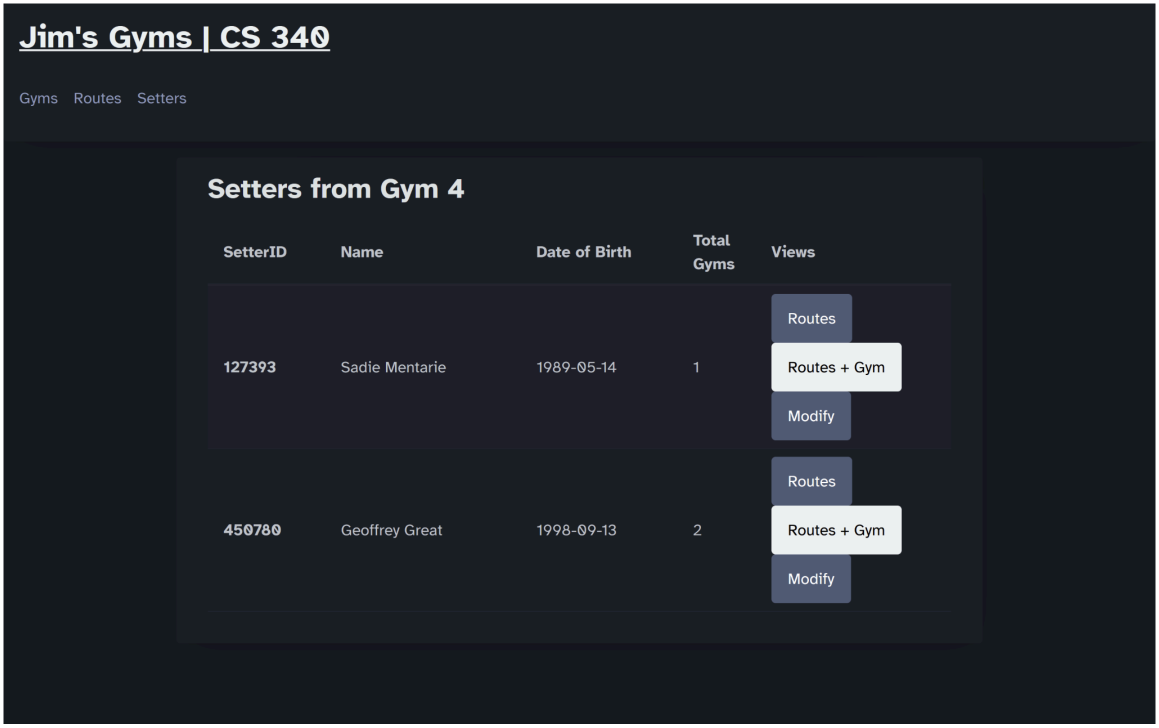
Routes from a Gym:
This page displays a table of routes that are set at a given gym.
This also includes a delete routes section that deletes routes by a user inputted date. If any routes match the inputted date in this gym, then it gets removed from the database.



## Routes from Gym 1

| Route Number | Gym | Difficulty | Setter | Color | Secondary Color | Total Holds | Created | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | John Climbing | D2B48C | DEB887 | 31 | 2011-11-02 | Delete Routes On Date |
| 2 | 1 | 2 | Mills Ends | DEB887 | D2B48C | 33 | 2011-11-02 | Delete Routes On Date |

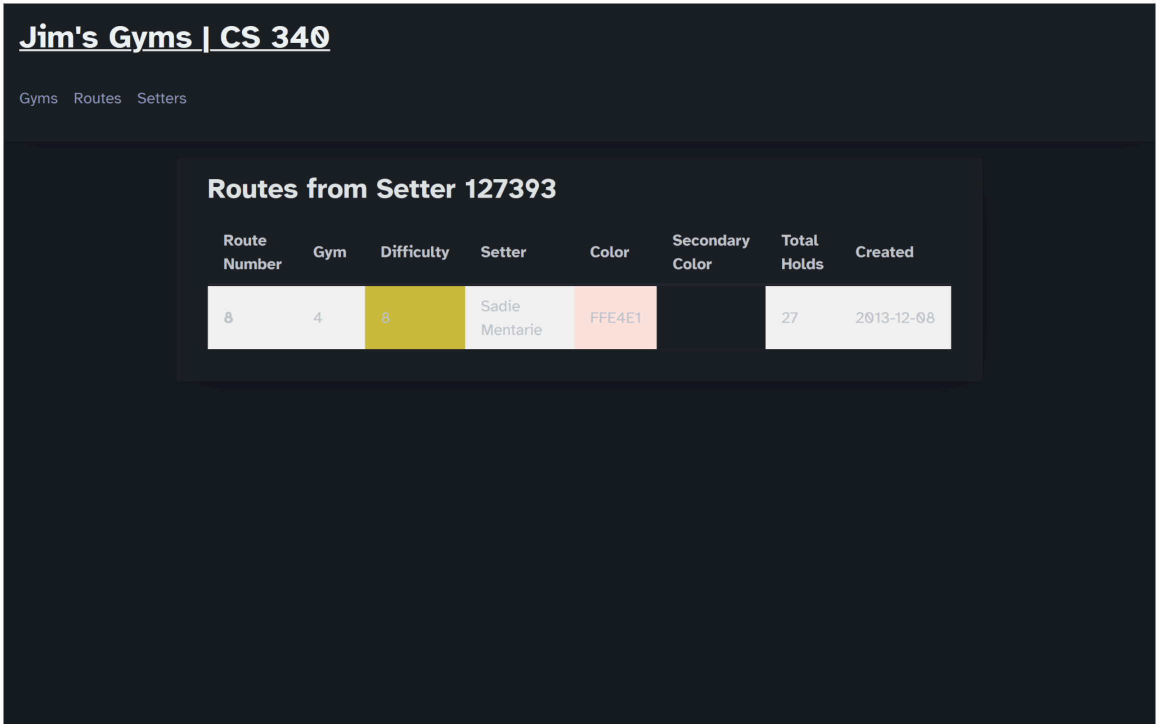
Setters from a Gym:
This page displays a list of setters from a selected gym.
Each setter has its information and a button to view the Routes made by a setter, the Routes and the Gym, and the name change form.



Setters from Gym 4

| SetterID | Name | Date of Birth | Total Gyms | Views |
|---|---|---|---|---|
| **127393** | Sadie Mentarie | 1989-05-14 | 1 | Routes<br>Routes + Gym<br>Modify |
| **450780** | Geoffrey Great | 1998-09-13 | 2 | Routes<br>Routes + Gym<br>Modify |

Routes from a Setter:
This page includes a table of routes set by a specified setter.

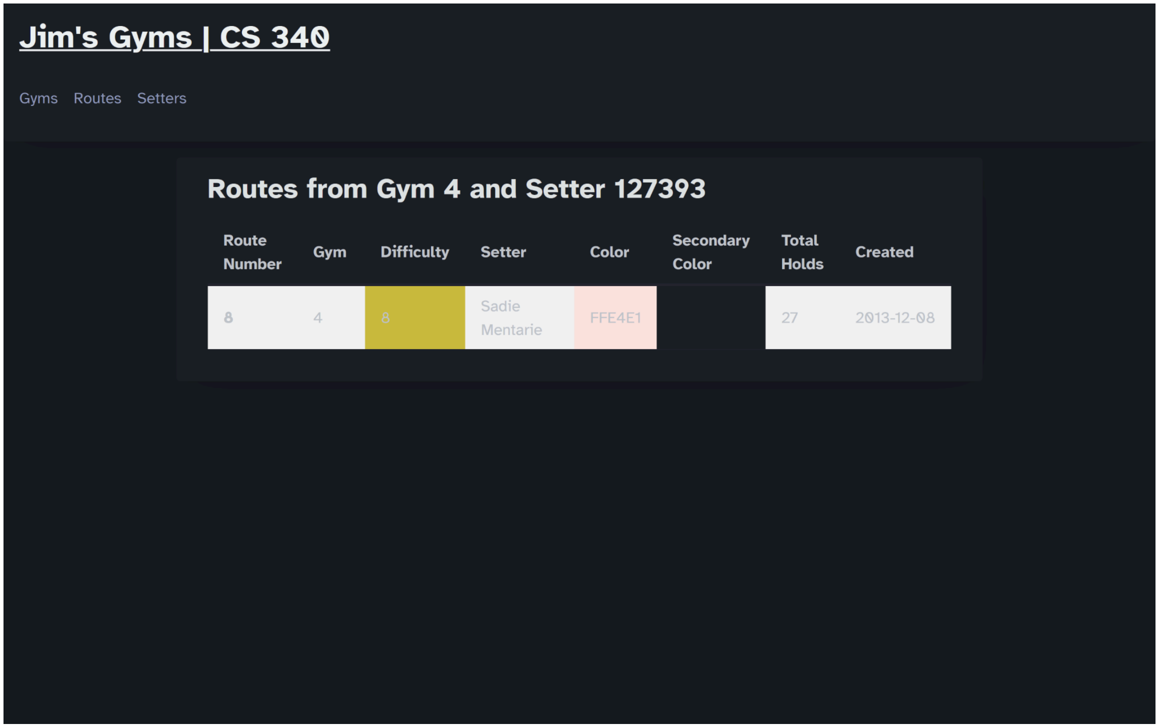


Jim's Gyms | CS 340

## Routes from Setter 127393

| Route Number | Gym | Difficulty | Setter | Color | Secondary Color | Total Holds | Created |
|---|---|---|---|---|---|---|---|
| 8 | 4 | 8 | Sadie Mentarie | FFE4E1 |  | 27 | 2013-12-08 |

Routes at a Gym set by a Setter:
This page shows all routes at a specified gym set by a specified setter.

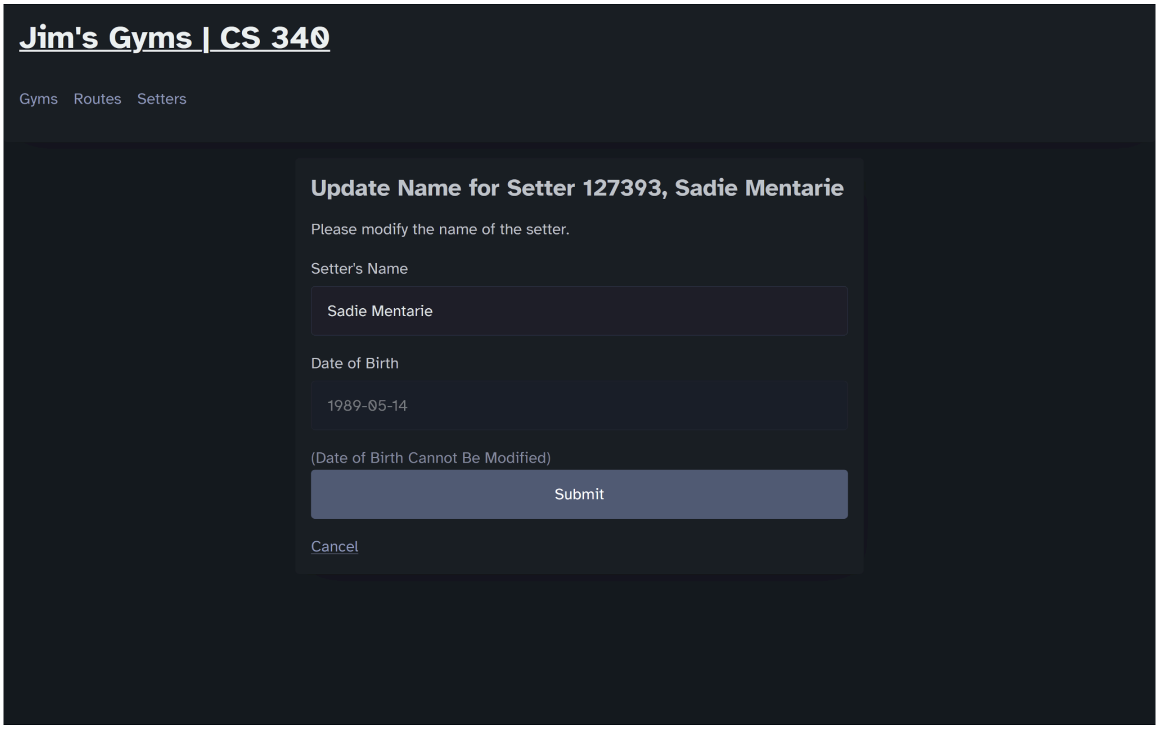


Jim's Gyms | CS 340

## Routes from Gym 4 and Setter 127393

| Route Number | Gym | Difficulty | Setter | Color | Secondary Color | Total Holds | Created |
|---|---|---|---|---|---|---|---|
| 8 | 4 | 8 | Sadie Mentarie | FFE4E1 |  | 27 | 2013-12-08 |

Name change form:
This page contains a form where you can change the name of a setter. The DOB option is there but is locked and cannot be changed. Clicking submit will change the setter’s name in the database.



Jim's Gyms | CS 340

Update Name for Setter 127393, Sadie Mentarie

Please modify the name of the setter.

Setter's Name

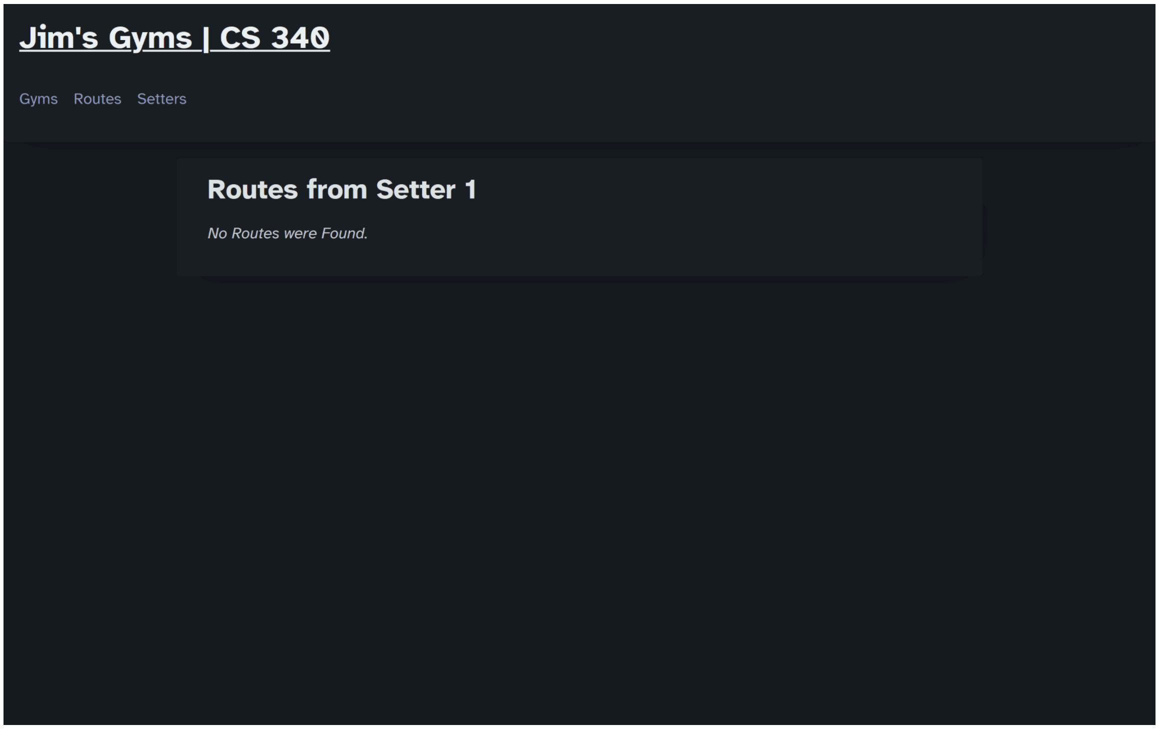
Sadie Mentarie

Date of Birth

1989-05-14

(Date of Birth Cannot Be Modified)

Routes from a Setter:
This page contains a list of routes set by a specified setter. If there are no routes, it lets the viewer know that no routes were found.

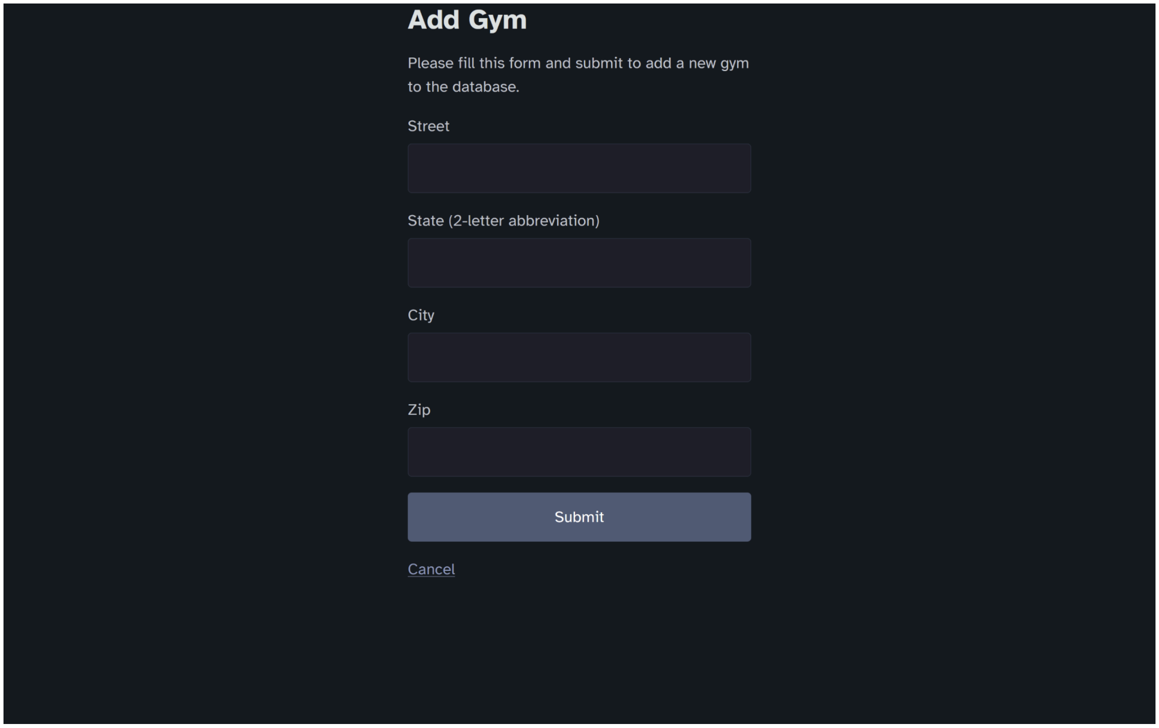


Jim's Gyms | CS 340

Routes from Setter 1

*No Routes were Found.*

Add gym form:
This page allows the user to add a new gym to the list of gyms. All information about the gym must be inputted into the form for the submit button to work. Once the submit button is clicked, it will redirect the user to the main page with the new list of gyms, including the new one.



## Add Gym

Please fill this form and submit to add a new gym to the database.

Street

State (2-letter abbreviation)

City

Zip

## 5. Application Implementation

- Describe your use of HTML/PHP/CSS/JavaScript/Node.js
    - The use of HTMLand PHP is largely based on the provided workplace project example, refitted for the purposes of this project.
    - We use the URL for queries and showing different displays, as well as specified buttons depending on the header queries (Navigating to the setters page from a specific gym vs from the navigation bar is the only instance of this).
    - Our CSS is a combination of the Atkinson Hyperlegible font face (https://fonts.google.com/specimen/Atkinson+Hyperlegible) and the PicoCSS minimalist css framework (https://picocss.com/).
    - We have no external javascript, but instead using embedded javascript for specified background colors as seen in the Routes page.
    - **Node.js is not used in this implementation.**
- Discuss your SQL queries
    - The main three pages use static SELECT queries to display all of the desired information and are only updated on refreshes.
        - SELECT #1 joins the routes to a single setter through its relationship table and gets the setter's name, the ids of the routes they set, and the date these routes were created.
        - SELECT #2 joins the routes to each gym and counts the number of routes for each gym id.
        - SELECT #3 joins the routes to each gym and averages the difficulties of all routes in each gym.
    - The pages branch off into separate pages for INSERT, DELETE, and UPDATE functions.
        - Our INSERT adds a new gym to the database using user input in a form.
        - Our UPDATE changes a setter's name using user input in a form.
        - Our DELETE deletes a gym from the database by clicking on a button on the gym's entry in the gyms table.
- Discuss your functions/procedures/triggers
    - Our first procedure deletes all routes in a gym that were created on a given date.
    - Our second procedure returns the number of routes of each difficulty across all gyms in the database. The procedure is run on the gym page to create the "Number of Routes by Difficulty Across All Gyms" table.
    - 
- Provide the URL for your Web App.
    - web.engr.oregonstate.edu/~mcmahoma/cs340/final_project/index.php

## 6. Future Work & Lessons Learned

What additional functionalities do you plan to implement beyond this class? What challenges have you run into during the design, implementation, and testing of your application? How did you address these issues? What would you do differently next time?

If there is any additional functionality we would add in the future, it would probably be adding new routes to gyms, adding and deleting setters, and adding and deleting individual routes. These would include attributing setters to the routes, and employing setters at the gyms. During our design phase, we weren't sure how we would represent our holds table in our website. Holds are related to routes and nothing else, and it would be a lot of unnecessary clutter to put them all on the website. In the end, we decided to represent them by counting the number of holds for each route. We ran into troubles with implementing the functions/procedures/triggers just because we didn't have a good understanding of them yet from the course materials. We ended up figuring it out, but it took more time to understand than the other parts of the project. Next time we would try to have a better understanding of them beforehand. A challenge we ran into during testing was that adding and deleting things using our website would change our database. We had to make a backup of our database and restore it every time we did any testing of our add/update/delete functionality. Next time, maybe we could research easier methods of protecting the database during testing.

## *Appendix – Team Report* *If you worked in a team summarize the division of labor.*

Kali - Developed a major portion of the website.
Matthew - Worked on implementing SQL queries for the website and wrote a large portion of the project grading sheet.
Palmer - Wrote a large portion of the project report. Significant contributions to the structural design of the website.
Silas - Worked on implementing SQL queries for the website. Significant contributions to the structural design of the website.
All members contributed to the grading sheet and project report.