CPSC 4620 Final Report

Benjamin Hargett - bharget - C56059977

Jacob Hurd - jhurd - C19777168

2017-12-02

1 Problem Statement and Motivation

It is obvious that people like to track things. With regards to fitness trackers alone, the most popular apps have a combined sum of over 50 million monthly users according to statista.com. People also love to know what other people are doing. As of 2017, 81% of Americans have a social media account according to the same source. With Range Finder, we want to expand the list of what people can keep track of while still allowing them to stay in touch with nature and their friends. There is not an adequate source for people to track the mountains that they have climbed, and the trackers that do exist do not have a social media component to allow people to connect with others who are passionate about the same thing. RangeFinder aims to fix that. Some people devote their lives to climbing mountains and with the sheer number of mountains in the world, it would be nearly impossible to track them for a serious climber. RangeFinder will be a valuable tool for both passionate mountain climbers and new-comers.

2 Web Portal Design

Our project, RangeFinder, is a Social Media web portal application that is designed to allow users to search a database of mountain peaks in North America as well as track the peaks that they have climbed. This involves a user registering for an account on our site which will then allow them to query the database based off of mountain name, country, state, elevation, latitude and longitude. Once logged in the user is able to mark a mountain as climbed as well as leave reviews and a rating for their trip. Users are also able to search for other user accounts and follow users to see information about their friends trips.

There is be an administrative component where a user with admin permissions can login and then edit, update, backup, or delete information in the database. Administrators can also restore a previous backup with the click of a button. Passwords will be encrypted to ensure the integrity and safety of the users information.

3 Database Population

Our dataset is contained in a table named mountains and consists of over 64,000 mountains across North America. This data was gathered from openstreetmap.org. We downloaded their map data for North America in XML format and then used their open-source tool, osmosis, to scan the XML for any data point that was labeled as a mountain peak. We then developed our own ruby script to parse this XML data into a JSON file. The script took the name, elevation, latitude, and longitude from the XML data entry and reverse-geocoded the coordinates to determine the country and state that the mountain is in. We then developed another ruby script to use this newly created JSON file to create a mountains table in our database and populate it with the data.

Users are a seperate table from mountains and are not created outside of the site. Upon deployment, the table for users will have a single entry for an admin user. All other users are created by registering for an account on the website interface. The username for admin is ”admin” and the password is ”password”. Logging in as admin will allow you to do anything a normal user could do plus admin privileges.

All relationships, comments, and ratings are also empty upon deployment. They are created by a user that has registered and logged in. We will however, create a few sample users to display the features.

4 Assumptions about Data

The data-set that we are using to populate our database is rather large so there are a few implicit assumptions that are made to allow users to accurately use the information from our site. We are assuming that the data-set is an accurate representation of mountains in North America. We are assuming that there are no duplicates in our data-set and we are assuming that no fields in the data are null. We have taken steps to try to eliminate any inconsistencies in our data-set, such as removing duplicates, but since the data-set is so large and it comes from an open-source organization with thousands of contributors, it is difficult to easily check the entire set for all assumptions made above.

There are also a few implicit assumptions that are made for the system to properly create a user that can navigate through and use the system properly. The system is designed in a way that forces users to enter in all the necessary information before the account can be created; however, we are assuming that all the information that the user enters is accurate information that reflects that individual. Email and phone validation are not implemented so we are assuming that they entered their information accurately.

Comments also have a few implicit assumptions that relate to Users. We assume that users are logged in when attempting to leave comments. Currently, the website is designed in a way that should not allow users to leave comments unless they are logged in. This design constraint is in place to avoid the creation of null fields within the comments table. We are also assuming that the User enters in accurate information about the Mountain in the comments field to ensure the integrity of the site.

There are also some assumptions about mountain_ratings. It is assumed that a user is logged in for them to be able to create a mountain_rating. It is also assumed that there will not be multiple rows in the table that have the same user_id and mountain_id. There are checks in place to prevent all of these faults from happening.

The assumptions regarding mountain_users are similar to those regarding mountain_ratings. It is assumed that a user is logged in for them to create a mountain_user. It is also assumed that there will not be multiple rows in the table with the same user_id and mountain_id. There are also checks in place to prevent this from happening.

For the Relationship Table, it is assumed that both the follower_id and followed_id in a column represent valid user ids within the Users Tables.

5 Sample Data and DB Diagrams

Mountains								
id	name	state	country	latitude	longitude	elevation	created_at	update_at
1	Kelly's Mountain	Nova Scotia	Canada	46.2519258	-60.5281258	240	09/19/17	09/24/17
2	Eliza Rock	Washington	United States of America	48.6437357	-122.5793553	0	09/19/17	09/24/17
3	Mount Whitney	California	United States of America	33.1088014	-117.1551691	527	09/19/17	09/24/17
4	Juniper Butte	California	United States of America	41.7914044	-121.4509942	1287	09/19/17	09/24/17
5	Wildcat Peak	California	United States of America	37.9192307	-122.2621766	370	09/19/17	09/24/17

Users												
id	username	password_hash	first_name	last_name	age	telephone	email	address	created_at	updated_at	admin	biography
1	jhurd	$2y$10$nOwe.BTD.YCQPC...	Jacob	Hurd	20	123-123-1234	jake@email.com	Some where	09/19/17	09/24/17	true	NULL

Comments				
id	content	created_at	mountain_id	user_id
1	Sample Content...	2017-10-31 20:14:06	1	1
2	Another Sample Content...	2017-11-31 20:14:06	1	300

Mountain_Ratings	
mountain_id	user_id
1	1
1	300

Mountain_Users		
cretated_at	mountain_id	user_id
2017-10-31 20:14:06	1	1
2017-11-18 20:14:06	1	300

Relationships		
created_at	followed_id	follower_id
2017-10-31 20:14:06	1	300
2017-11-18 20:14:06	300	1

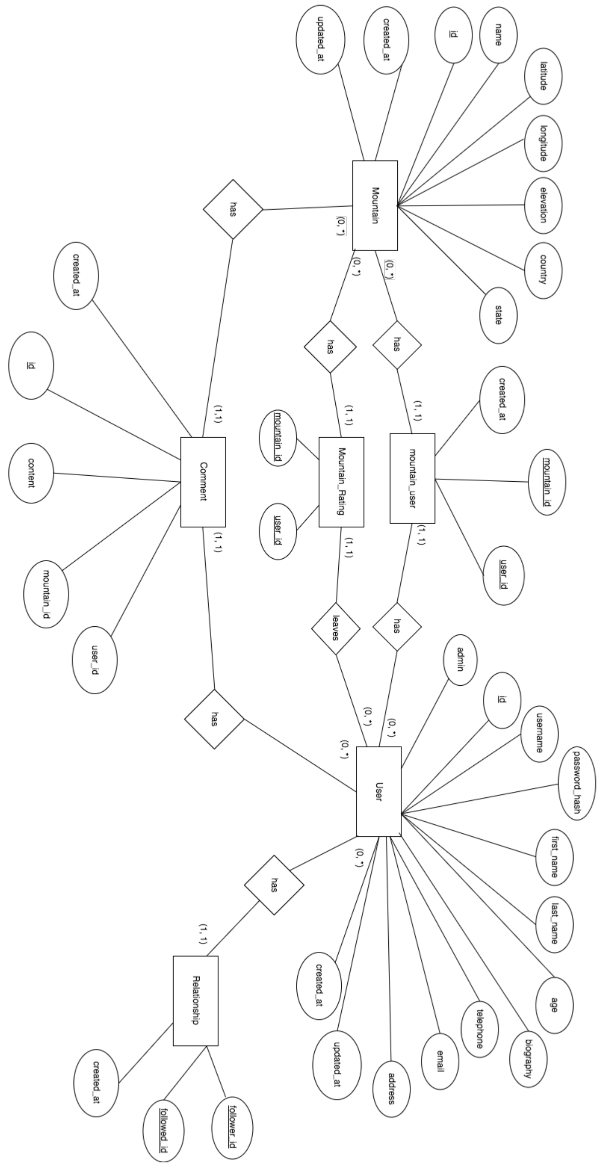


Figure 1: ER Diagram

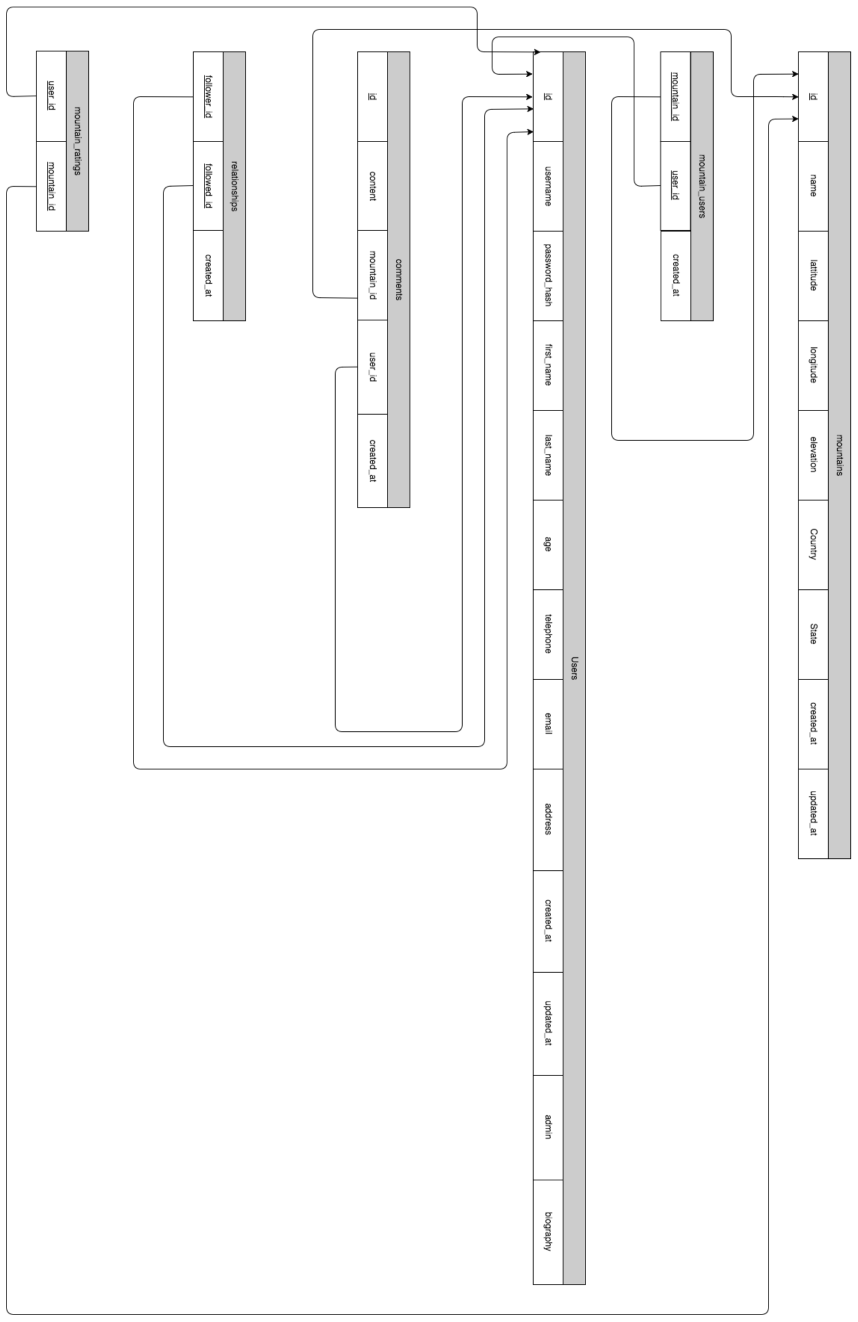


Figure 2: Relational Schema

6 SQL Queries

6.1 Logging in and registering users

- SELECT id, username, password_hash FROM users WHERE username = ? LIMIT 1
- SELECT password_hash FROM users WHERE id = ? LIMIT 1
- SELECT id FROM users where username = ? LIMIT 1
- INSERT INTO users (username, password_hash, first_name, last_name, age, telephone, email, address) VALUES (?, ?, ?, ?, ?, ?, ?, ?)
- SELECT * FROM users WHERE username LIKE ? OR first_name LIKE ? OR last_name LIKE ?

6.2 Searching for mountains

- SELECT * FROM mountains WHERE name like % ? %
- SELECT * FROM mountains WHERE state = ?
- SELECT * FROM mountains WHERE country = ?
- SELECT * FROM mountains WHERE latitude like ? %
- SELECT * FROM mountains WHERE longitude like ? %
- SELECT * FROM mountains WHERE elevation ¡= ?
- SELECT * FROM mountains WHERE elevation ¿= ?
- SELECT * FROM mountains where id = ? LIMIT 1
- INSERT INTO mountain_users (user_id, mountain_id, created_at) VALUES (?, ?, ?)

6.3 Get Mountains with Most Ratings

- SELECT count(mountains.id), mountains.name, mountains.id FROM mountains INNER JOIN mountain_ratings ON mountains.id = mountain_ratings.mountain_id GROUP BY (mountains.id) ORDER BY count(mountains.id) DESC

6.4 Generate User Feed

- SELECT mountains.id AS mountain_id, mountains.name, mountain_users.created_at, users.username, users.id AS user_id FROM relationships INNER JOIN mountain_users ON relationships.followed_id = mountain_users.user_id INNER JOIN users ON relationships.followed_id = users.id INNER JOIN mountains ON mountain_users.mountain_id = mountains.id WHERE follower_id = ? ORDER BY mountain_users.created_at DESC;

6.5 Get Mountains Associated with User

- SELECT * FROM mountain_users INNER JOIN mountains ON mountain_users.mountain_id = mountains.id WHERE mountain_users.user_id = ? ORDER BY created_at DESC

6.6 Get Mountain Ratings associated with User

- SELECT * FROM mountain_ratings INNER JOIN mountains ON mountain_ratings.mountain_id = mountains.id WHERE mountain_ratings.user_id = ? ORDER BY created_at DESC

6.7 Get the mountains who have the most climbers

- SELECT count(mountains.id), mountains.name, mountains.id FROM mountains INNER JOIN mountain_users ON mountains.id = mountain_users.mountain_id GROUP BY (mountains.id) ORDER BY count(mountains.id) DESC

6.8 Searching for Users

- SELECT * FROM users where id = ? LIMIT 1
- SELECT * FROM users where username = ?

6.9 Queries for Comments

- SELECT * FROM comments INNER JOIN users ON comments.user_id = ? WHERE mountain_id = ? ORDER BY created_at DESC
- INSERT INTO comments (content, created_at, mountain_id, user_id) VALUES (?, ?, ?, ?)

6.10 Queries for Mountain Ratings

- SELECT COUNT(*) FROM mountain_ratings WHERE mountain_id = ?
- SELECT * FROM mountain_ratings WHERE mountain_id = ?
- SELECT * FROM mountain_ratings WHERE mountain_id = ? AND user_id = ?
- INSERT INTO mountain_ratings (user_id, mountain_id, created_at) VALUES (?, ?, ?)
- SELECT COUNT(*) FROM mountain_ratings WHERE user_id = :id
- SELECT id, username, created_at FROM mountain_ratings INNER JOIN users ON mountain_ratings.user_id = users.id WHERE mountain_id = ? ORDER BY created_at DESC

6.11 Queries for Mountain Users

- SELECT COUNT(*) FROM mountain_users WHERE mountain_id = ?
- SELECT * FROM mountain_users WHERE mountain_id = ?
- SELECT * FROM mountain_users WHERE mountain_id = ? AND user_id = ?

- INSERT INTO mountain_users (use_id, mountain_id, created_at) VALUES (?, ?, ?)
- SELECT COUNT(*) FROM mountain_users WHERE user_id = ?
- SELECT id, username, created_at FROM mountain_users INNER JOIN users ON mountain_users.user_id = users.id WHERE mountain_id = ? ORDER BY created_at DESC

6.12 Queries for Relationships

- SELECT * FROM relationships INNER JOIN users ON relationships.follower_id = users.id WHERE followed_id = ?
- SELECT * FROM relationships INNER JOIN users ON relationships.followed_id = users.id WHERE follower_id = ?
- SELECT COUNT(*) FROM relationships WHERE follower_id = ?
- SELECT COUNT(*) FROM relationships WHERE followed_id = ?
- SELECT * FROM relationships WHERE follower_id = ? AND followed_id = ?

6.13 Deleting a Mountain

- DELETE FROM comments WHERE mountain_id = ?;
- DELETE FROM mountain_ratings WHERE mountain_id = ?;
- DELETE FROM mountain_users WHERE mountain_id = ?;
- DELETE FROM mountains WHERE id = ?

6.14 Liking a Mountain

- INSERT INTO mountain_ratings (user_id, mountain_id, created_at) VALUES (?, ?, ?)

6.15 Updating a Mountain

- UPDATE mountains SET name = ?, state = ?, country = ?, elevation = ?, latitude = ?, longitude = ? WHERE id = ?

6.16 Create a New Comment

- INSERT INTO comments (content, created_at, mountain_id, user_id) VALUES (?, ?, ?, ?)

6.17 Deleting a User

- DELETE FROM comments WHERE user_id = ?;
- DELETE FROM mountain_ratings WHERE user_id = ?;
- DELETE FROM mountain_users WHERE user_id = ?;
- DELETE FROM relationships WHERE follower_id = :id OR followed_id = ?;
- DELETE FROM users WHERE id = ?

6.18 Create a Relationship

- INSERT INTO relationships (follower_id, followed_id, created_at) VALUES (?, ?, ?)

6.19 Destroy a Relationship

- DELETE FROM relationships WHERE followed_id = ? AND follower_id = ?

6.20 Update a User

- UPDATE users SET first_name = ?, last_name = ?, email = ?, age = ?, telephone = ?, address = ?, biography = ?, admin = ? WHERE id = ?

7 Web Portal Interface Description

The users interaction with the portal interface begins with a login screen. The home page of the site is a prompt for the user to enter their username and password to login to Range Finder. If they do not have an account, there is an option to register for an account. If the user is already logged in, then instead of displaying a login prompt, a dashboard will be displayed with pertinent information for that user such as their most recent climbs as well as recent climbs by users that they are following. It will also display the sites most Popular Mountains.

Once logged in, on their account page they will be able to view all of the mountains that they have marked as climbed. They will also have the option to perform basic database queries which will give them the ability to specify the name, an elevation range, state, country, and/or coordinates. A list of results will be displayed and the user can select a mountain which will bring up a page with the pertinent information regarding that mountain. There will be an option to mark the mountain as climbed and/or liked, as well as an option to leave a comment. If the user chooses to leave a comment, then they will be prompted to enter text for the body of their comment. From the logged-in users account page, they will have the ability to view the list of people that they follow as well as the list of people that follow them. They will also have the option to search the database of users by entering their first name, last name, or username. A list of results will be displayed and the logged-in user can select a user they wish to view. A page will display that users publicly view-able information such as their name and username as well as their recent climbs and climber stats. If the user looks at their own page, or the My Account page, they will have the ability to edit their information. The logged-in user will then be able to follow or un-follow that user.

If an administrative user logs in to RangeFinder, they will be taken to a dashboard very similar to the normal user dashboard. They will be able to search the database for mountains similar to a normal user, but they will have an additional option to edit or delete that mountains information when viewing the mountains page. They go through a similar process with users; they are able to search for users but with the additional option to edit/delete them. If editing a mountain or user, they will be prompted with a form where they can input the updated information. In addition to the features listed above, administrators will have the ability to go to a DB Backup page from the Navigation Bar where they can create new backups, restore from old backups, delete backups, and download backups of the database.

8 Project Requirements

8.1 User Account Management

8.1.1 Basic Functions

We have created a login module, registration module, and and edit module for the users to create and manage accounts on RangeFinder (Figure 3). When on the homepage of the site, if the user is not logged in, there will be a prompt for a username and password if they wish to sign in and an option to register if they dont have an account. To test the registration and login module, you can fill out the registration form and then use the new account to login from the homepage (Figure 4). To test the edit module, login to RangeFinder and on the navigation bar there will be an account dropdown option. Under this option select My Account and the will be an edit option next to the users information. Selecting this will bring up a form to allow you to edit and update your account information (Figure 5). In addition to editing their account information, the user can also select to edit their biography to add a description about themselves on their account page (Figure 6).

8.1.2 Advanced Features

We implemented our registration form in a way that does not allow it to submit with any blank fields. It also checks the password entered by the user and verifies that it is at least 6 characters long (Figure 7).

8.2 Data Management

8.2.1 Basic Functions

There are a lot of features in RangeFinder that fall under the data management category. The users can search and view lists of mountains and users. This can be tested by going to the search dropdown in the navigation bar and selecting which you would like to search for. If mountains are selected, the user will be brought to a form where they can query the table of mountains based off what they enter for the mountains attributes (Figure 8). The same happens if the user chooses to search for other users; they will be brought a form where they can query the table of users based off first name, last name, or username.

Once the user searches for the information they desire, they will be brought to a page to view the results of their query (Figure 9). They can then scroll through the pages of results and select one that they wish to view. This will direct them to a page to display the user or mountains information depending on what they searched (Figure 10). If the user is an administrative user they will have the option to edit or delete each mountain and user if they wish when viewing their respective information page.

When viewing other users account page, the logged in user will also have the ability to follow or unfollow a user(Figure 11). The creates or deletes entries in the relationships table.

When viewing a mountains information page, the logged in user will be able to click a button to mark the mountain as climbed as well as like the mountain (Figure 10). They can also view a list of all the other users who have liked and/or climbed the given mountain by following the link under each of the respective buttons listing out many users have like/climbed it. The user will also have the ability to read and leave comments for the mountain they are viewing(Figure 12). To leave a comment, the user must be logged in and select the pencil icon next the comments title. This will bring up a form to submit a new comment.

8.2.2 Advanced Feature

For an advanced feature regarding data management, we had any page that lists results of a large query automatically split into pages so that the user can easily navigate a large number of results (Figure 13).

8.3 Database Administration

8.3.1 Basic Functions

For the basic functions dealing with database administration we have included an administrator account with the following credentials:

- Username: admin
- Password: password

The admin user will have the ability to search for user accounts similarly to any other user, but they will also be able to edit/update or delete the user when viewing another users information page.

In addition to account administration, the admin user will have access to the admin tab on the navigation bar where they can select DB Backup. This will direct them to a page where they can view the database backups stored on the server which will be listed in order from newest backup to oldest (Figure 14). There will be a button to create a new backup which will save an sql dump of the database on the server. There will also be an option to delete older backups; this will cut down on wasted server space. The administrator can click the name of the sql dump to download it.

8.3.2 Advanced Feature

There is an option to restore the database based off of one of the database backups. The easiest way to see that this works is to make a new backup, edit/delete user or mountain records, then go back and click restore for the backup most recently created. This will restore the database to what it was when that backup was created.

9 Limitations

The only technical limitation that we are aware of with our current implementation is that it is not possible to return our entire table of mountains in one query due to limitations on the Clemson webapp server. There is an error that is thrown stating that the allotted memory has been used. This is understandable though due to the fact that our database contains over sixty thousand records. When testing on local servers, this error is not thrown so it must have to do with the webapp server configuration and could be fixed in the future if it is ever hosted on another server where we could increase the allotted memory.

Even though there are no more know technical limitations known that would currently hinder the websites use, if there was more time to work on the project, it would be ideal to reimplement the website using a PHP web framework. This would allow us to cut down on code duplication and improve the general efficiency of the website as a whole.

10 Figures (Screenshots)

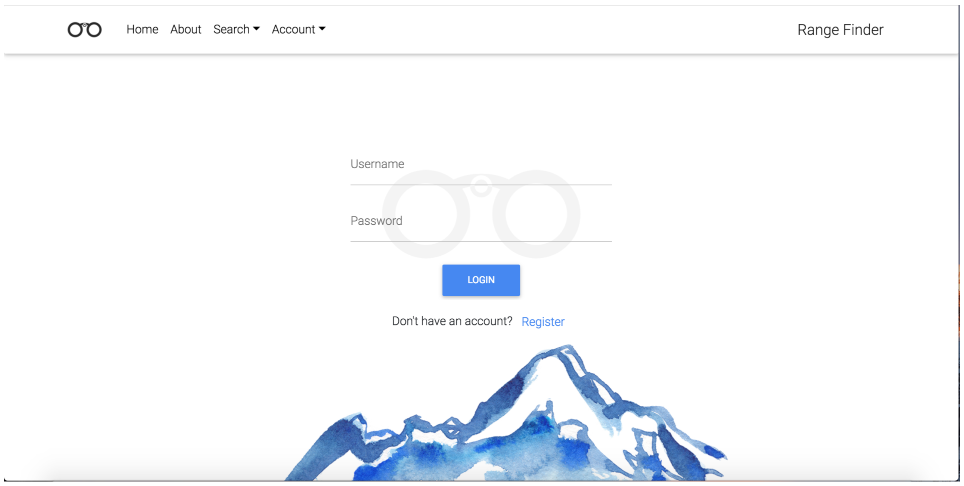
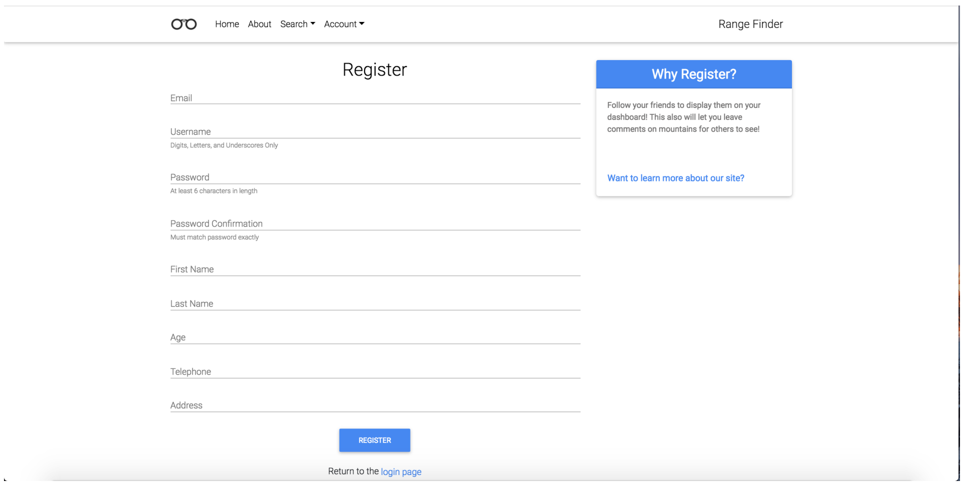
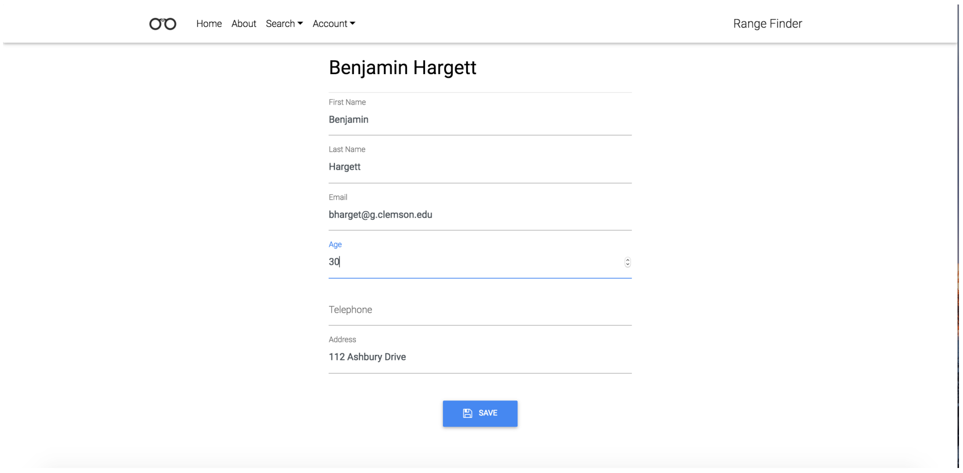


Figure 3: Login Page



Follow your friends to display them on your dashboard! This also will let you leave comments on mountains for others to see!

Figure 4: Register Page



Benjamin Hargett

First Name
Benjamin

Last Name
Hargett

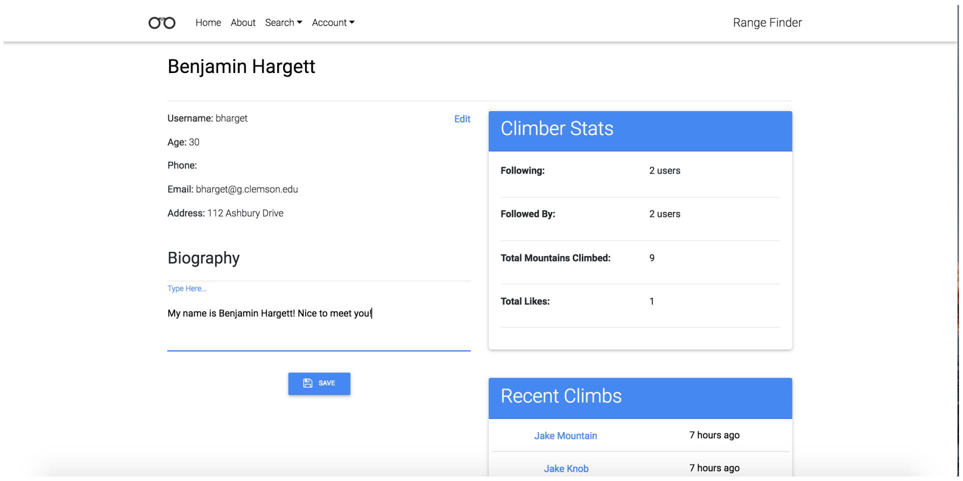
Email
bharget@g.clemson.edu

Age
30

Telephone

Address
112 Ashbury Drive

Figure 5: Update Account



Benjamin Hargett

Username: bharget

Age: 30

Phone:

Email: bharget@g.clemson.edu

Address: 112 Ashbury Drive

Biography

Type Here...

My name is Benjamin Hargett! Nice to meet you!

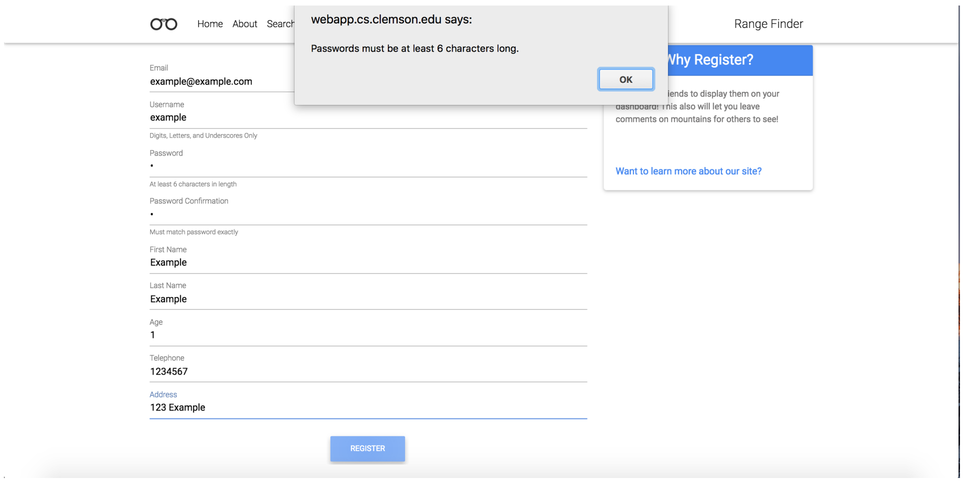
Climber Stats

Following:	2 users
Followed By:	2 users
Total Mountains Climbed:	9
Total Likes:	1

Recent Climbs

Jake Mountain	7 hours ago
Jake Knob	7 hours ago

Figure 6: Update Account - Biography



webapp.cs.clemson.edu says:

Passwords must be at least 6 characters long.

iends to display them on your dasnboard! This also will let you leave comments on mountains for others to see!

Want to learn more about our site?

Email
example@example.com

Username
example

Digits, Letters, and Underscores Only

Password
•

At least 6 characters in length

Password Confirmation
•

Must match password exactly

First Name
Example

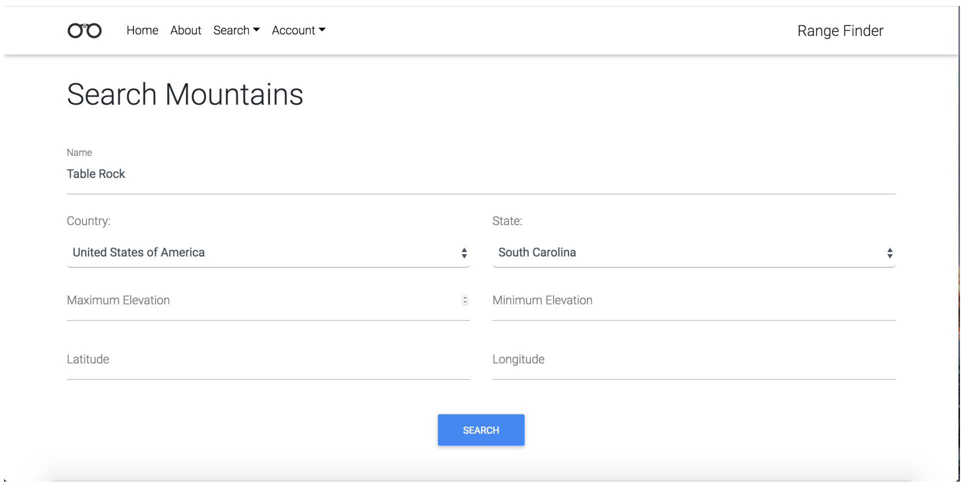
Last Name
Example

Age
1

Telephone
1234567

Address
123 Example

Figure 7: Password Restriction



Search Mountains

Name
Table Rock

Country:
United States of America

State:
South Carolina

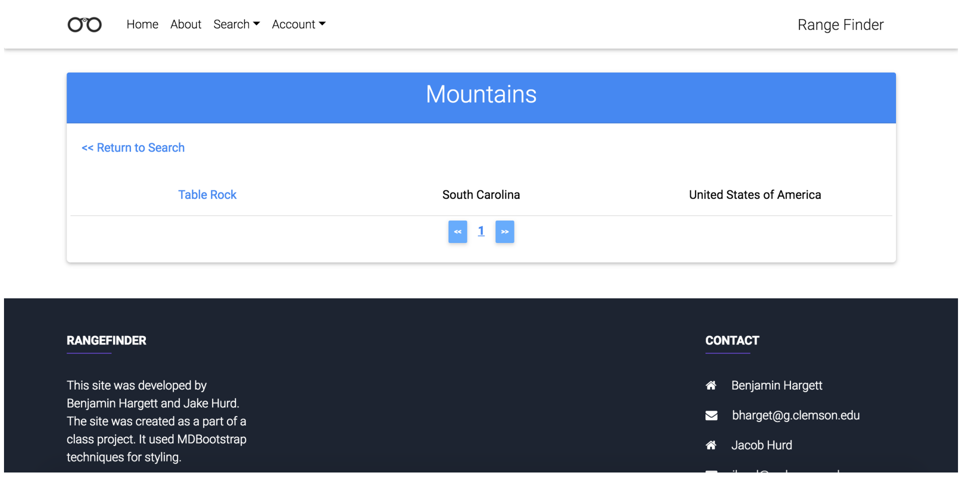
Maximum Elevation

Minimum Elevation

Latitude

Longitude

Figure 8: Search for Mountains



Mountains

<< Return to Search

Table Rock	South Carolina	United States of America

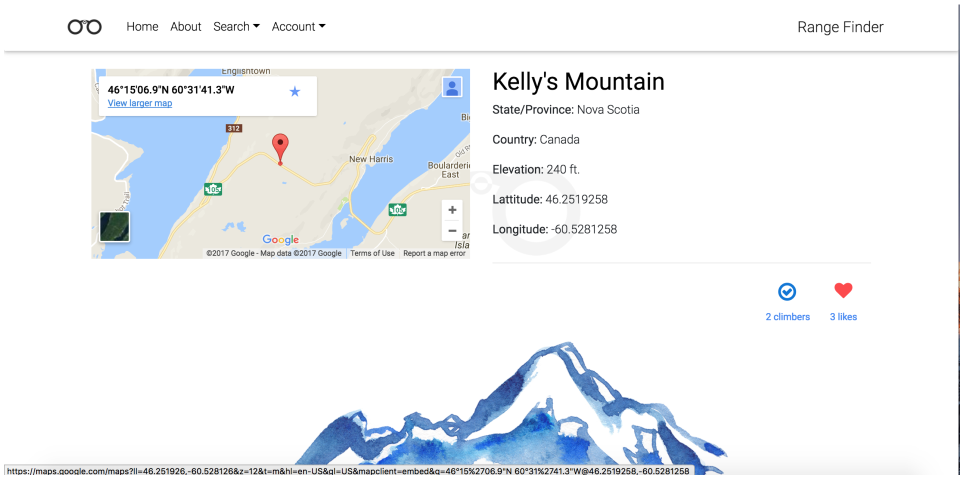
RANGEFINDER

This site was developed by Benjamin Hargett and Jake Hurd. The site was created as a part of a class project. It used MDBootstrap techniques for styling.

CONTACT

Benjamin Hargett
bharget@g.clemson.edu
Jacob Hurd

Figure 9: Mountain Search Results



Kelly's Mountain

State/Province: Nova Scotia

Country: Canada

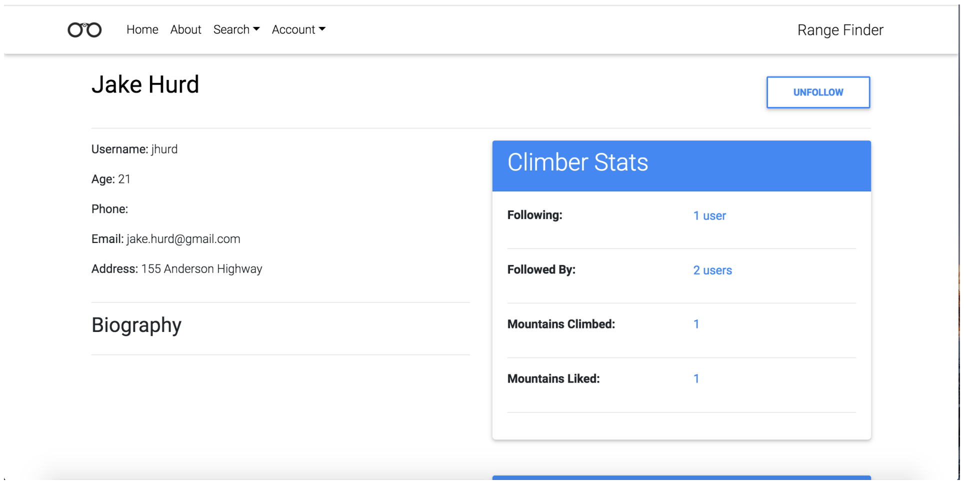
Elevation: 240 ft.

Lattitude: 46.2519258

Longitude: -60.5281258

https://maps.google.com/maps?ll=46.251926,-60.528126&z=12&t=m&hl=en-US&gl=US&mapclient=embed&q=46°15%2706.9"N 60°31%2741.3"W@46.2519258,-60.5281258

Figure 10: Mountain Information



Jake Hurd

Username: jhurd

Age: 21

Phone:

Email: jake.hurd@gmail.com

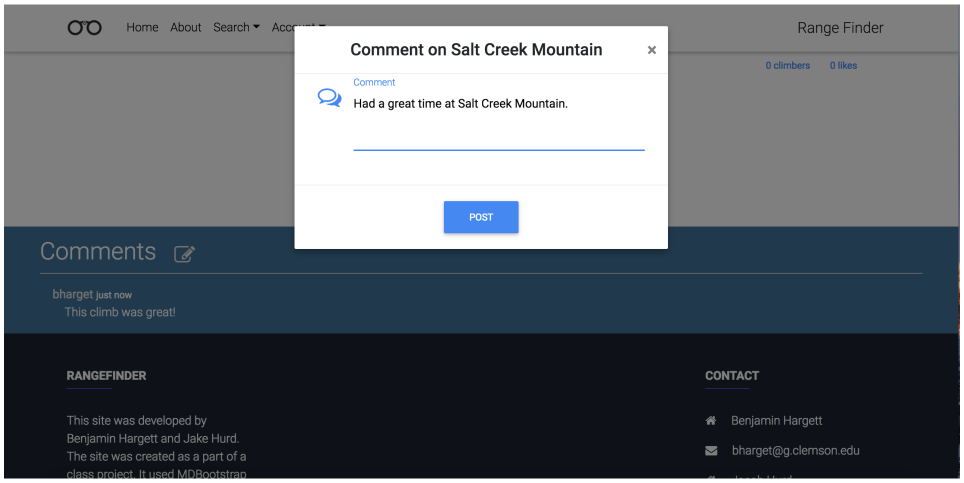
Address: 155 Anderson Highway

Biography

Climber Stats

Following:	1 user
Followed By:	2 users
Mountains Climbed:	1
Mountains Liked:	1

Figure 11: Follow/Unfollow



Comment on Salt Creek Mountain

Comment: Had a great time at Salt Creek Mountain.

Comments

bharget just now
This climb was great!

RANGEFINDER

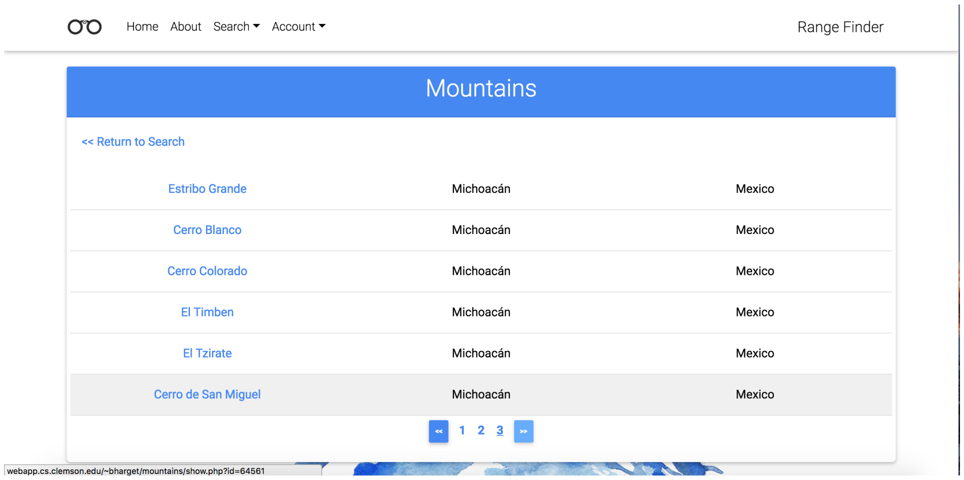
This site was developed by Benjamin Hargett and Jake Hurd. The site was created as a part of a class project. It used MDBootstrap

CONTACT

Benjamin Hargett

bharget@g.clemson.edu

Figure 12: Post a New Comment



Mountains

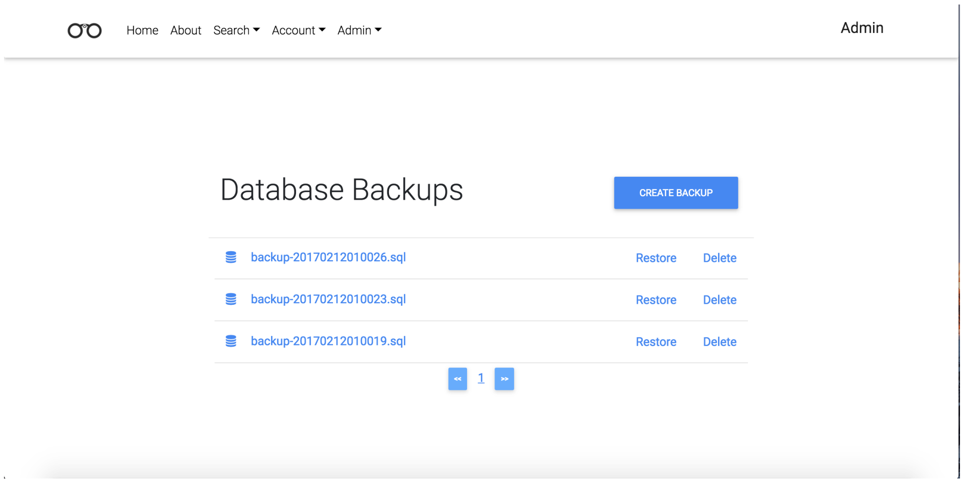
<< Return to Search

Estribo Grande	Michoacán	Mexico
Cerro Blanco	Michoacán	Mexico
Cerro Colorado	Michoacán	Mexico
El Timben	Michoacán	Mexico
El Tzirate	Michoacán	Mexico
Cerro de San Miguel	Michoacán	Mexico

« 1 2 3 »

webapp.cs.clemson.edu/~bharget/mountains/show.php?id=64561

Figure 13: Pagination



Database Backups

CREATE BACKUP

backup-20170212010026.sql	Restore	Delete
backup-20170212010023.sql	Restore	Delete
backup-20170212010019.sql	Restore	Delete

« 1 »

Figure 14: DB Backup Page

11 Contributions

Jake Hurd and Benjamin Hargett both worked on this project together throughout the course of the semester. Throughout that time, there have been no issues to report about team members not fulfilling their responsibilities with regards to the project. Although both Benjamin Hargett and Jake Hurd both contributed to different areas of the project, Benjamin was more responsible for the front end design of the web application where as Jake Hurd was mainly responsible for the back end of the site. That being said, both Benjamin and Jake worked together on many different problems that arose from the project as a whole.

Both members also were equally involved in the database design phase. The two members came up with the idea for the site and designed the layout together.

12 Links to Source Code

- https://goo.gl/reA2XA

or

- https://drive.google.com/uc?export=downloadid=1zrUleHb0FGbDZNcmLbOE6leymFyRzsqa