Project Report

Title: Acendify

Members:

Eleanor Blair: elblair

Brittany Lewis: brle0846

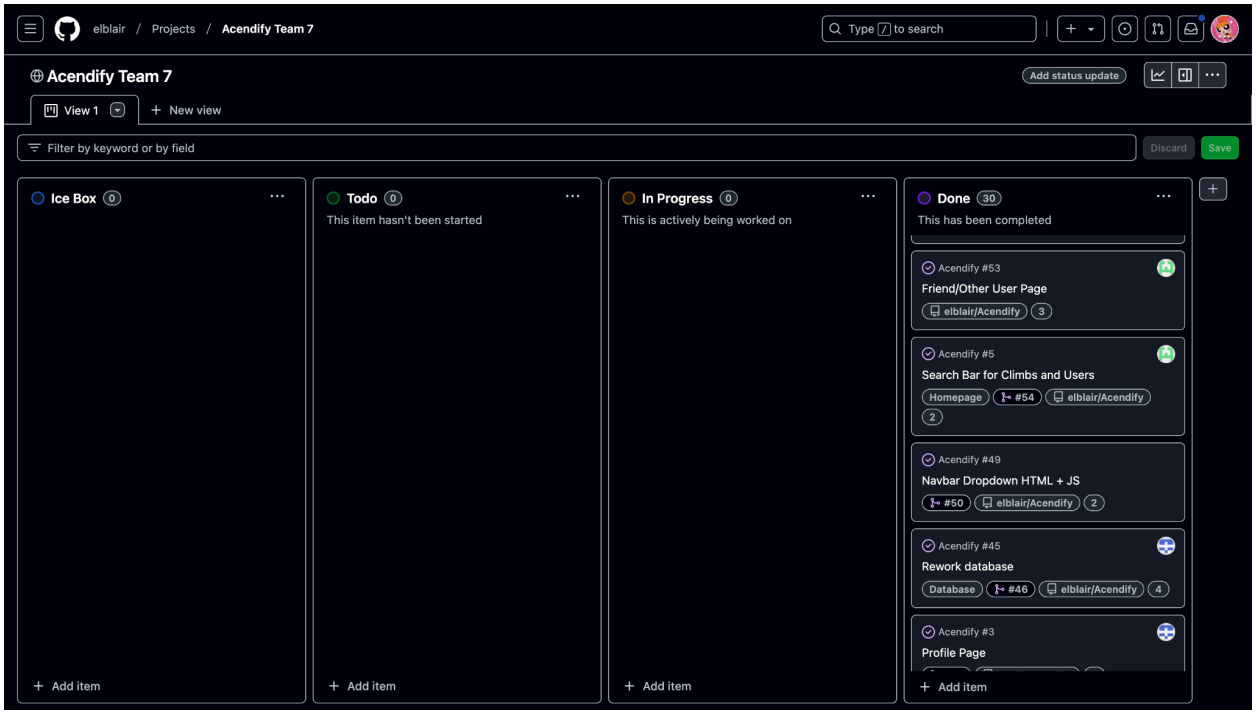
Owen White: owwh304/owhite7128

Adam Gebben: adge7706

Project Description:

Acendify is your go-to platform for logging and discovering climbing routes. Whether you're tracking your own ascents, exploring new climbs, or learning from the experiences of others, Acendify provides a centralized hub for climbers to document their progress, share insights, and find inspiration for their next adventure. It's not just a tool—it's a resource to elevate your climbing journey. When going on our website, the user starts on the register page and has the option to register an account that gets stored in a users database. Then, they can login and they are directed to our main home page that displays a feed of climbs. The user can add climbs or log ascents, and they will show up on the feed once they are logged. The user can also go into their profile and update various fields such as their full name, their age, their height, and their wingspan. There is a profile page that will store and display the updated information, along with the date that the user created their account in order for users to see how far they've come over time. Users also have the ability to click on other user's profiles and add them as a friend, as well as unadd them. Once they are friends, that user's information will be displayed on the main feed as well.

Project Tracker:



https://github.com/users/elblair/projects/1

Video:

https://github.com/elblair/Acendify/blob/main/milestone_submissions/demo.mkv

VCS:

https://github.com/elblair/Acendify

Contributions:

Every member contributed to this final project:)

Link to commits: https://github.com/elblair/Acendify/graphs/contributors

Brittany Lewis

For Acendify I had two main jobs, the first was coding the initial design for the home page, including a prototype for infinite scroll, and setting up the hbs pages that we assumed were going to be needed for this project. The initial hbs' were friends, home, map, profile, login, register. I reused some of the work we had done in previous labs to give us a head start on these. And then my other main

contribution was the initial setup for the search bar. I had done the initial design and setup of search until it was working, however it was updated later by Owen.

Eleanor Blair

I focused on structuring and front-end development. I created our GitHub repository, set up the code skeleton with essential pages, partials, and Docker files, and filled partials (except nav.hbs) with data from past labs, adapting them to our project. I handled most of the CSS, initially styling pages individually before Adam helped centralize it into a universal stylesheet. I developed the front-end for the user settings page, while Adam connected it to the database. Additionally, I worked on implementing a profile picture feature for user settings, though it wasn't used due to time constraints.

Adam Gebben

My contributions were to build out key features, namely design and population of the database and all features interfacing with the database. Some key features I created were the dynamically created cards on the homepage feed referencing the ascents table, the followers functionality, and many of the routes that interfaced with our postgreSQL database, like the user profile page and the user setting page, which queries the users table for information. I also used my experience with git and development to assist my fellow group members when they ran into issues, be it version control or javascript.

Owen White

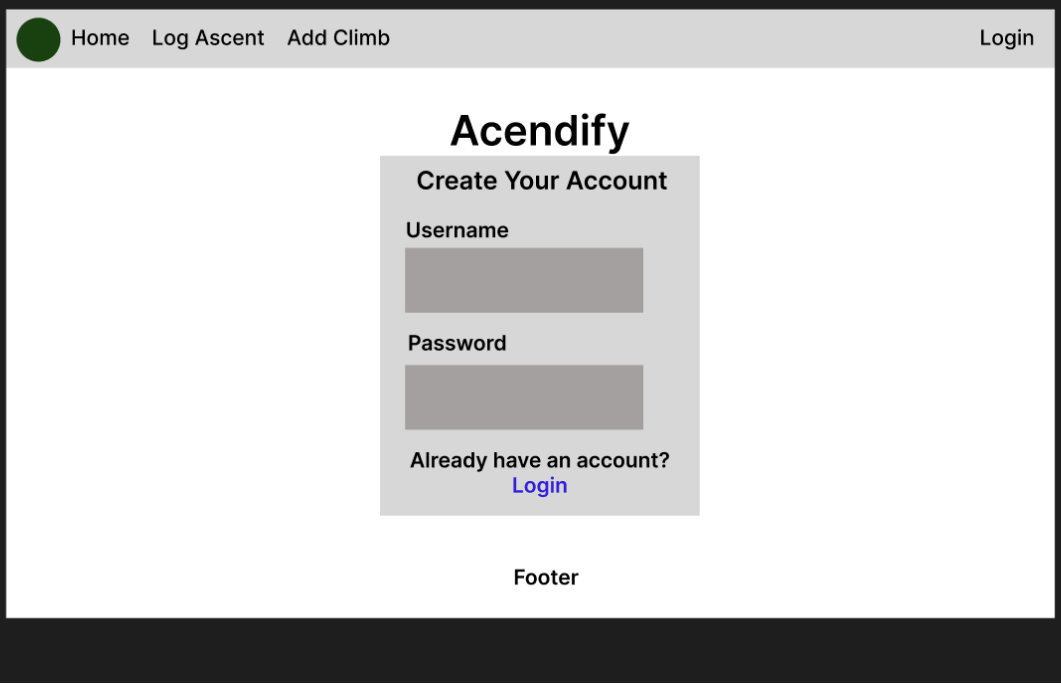
I was primarily focused on aiding my group members with the back end required for their front end to work. I helped Brittany out with the search bar and I also contributed by figuring out how to get infinite scrolling to work, so that it does not load all of the content at once. I was also tasked with the deployment of the website, as I have a server at home that was capable of hosting the website. I worked minorly on the html+css to get some of the big picture styling ideas together.

Case Diagram:

Wireframes:

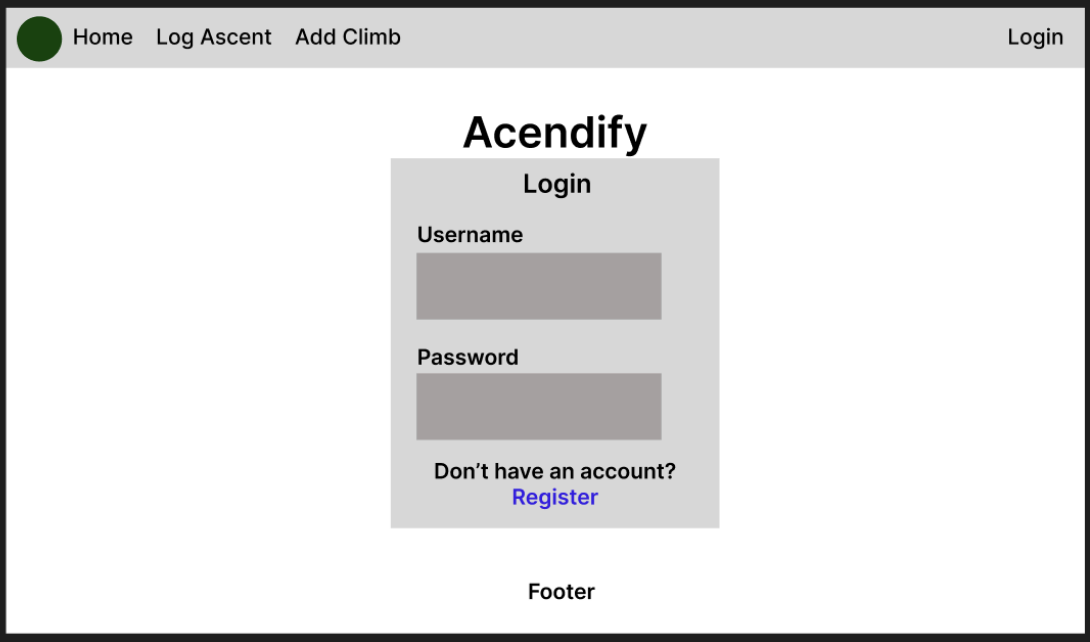
Register

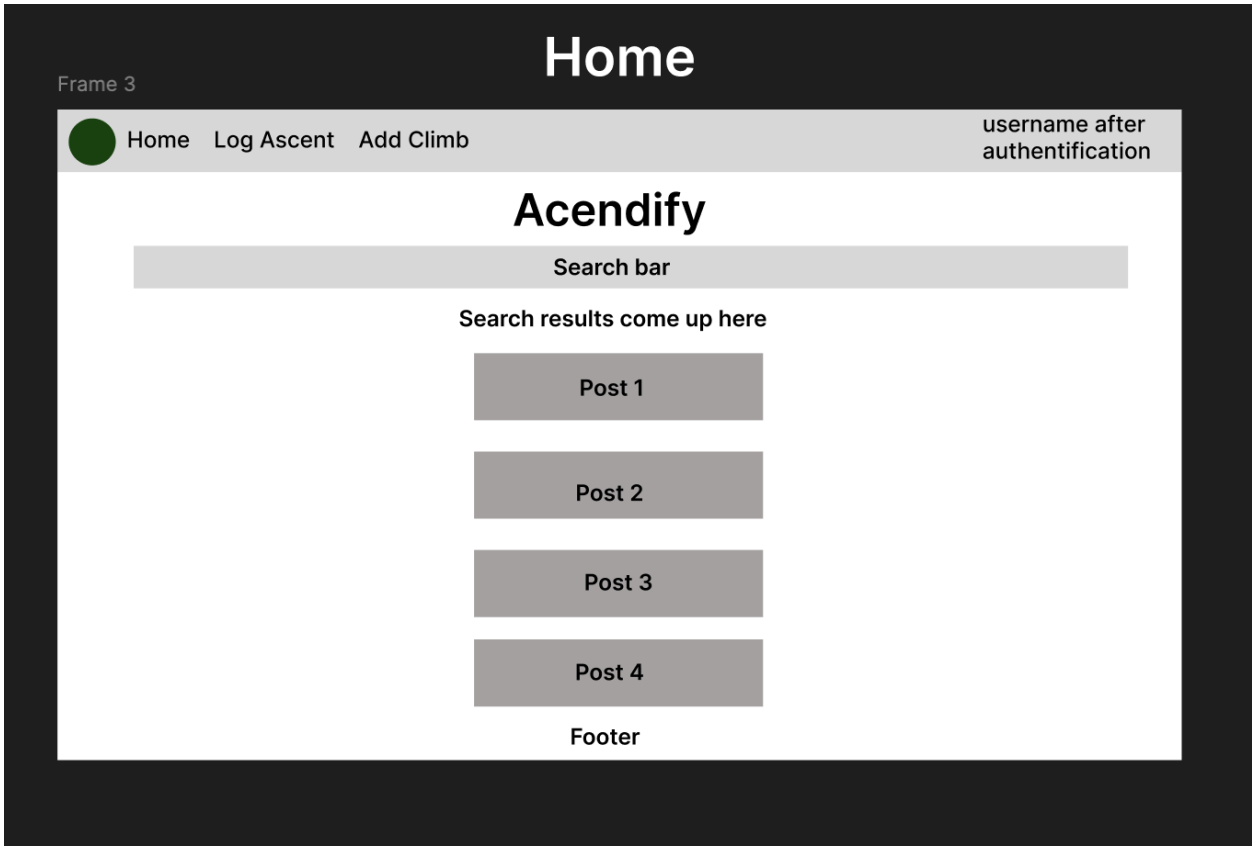
Frame 1



Login

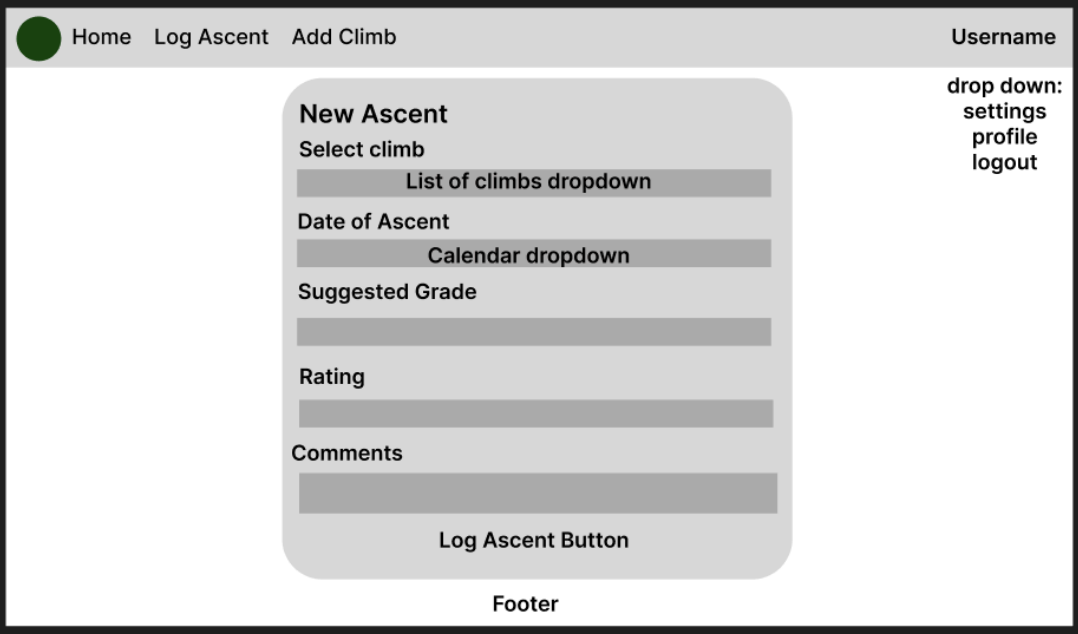
Frame 2




Home
Frame 3
Home
Log Ascent
Add Climb
username after authentification
Acendify
Search bar
Search results come up here
Post 1
Post 2
Post 3
Post 4
Footer

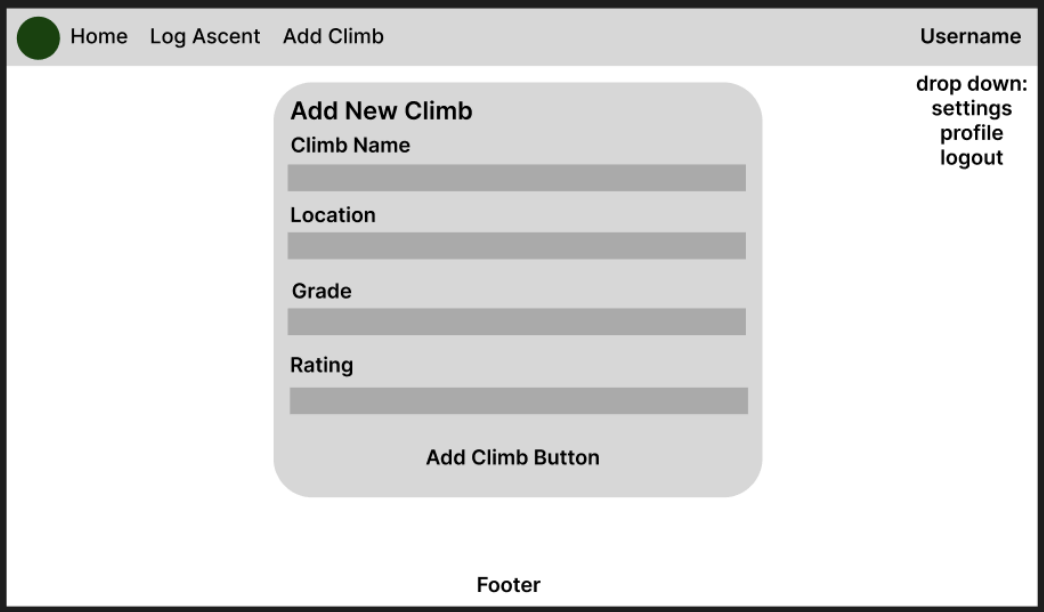
Frame 4

Log Ascent



Frame 6

Add Climb



Profile Page

Frame 8

Home Log Ascent Add Climb Username

Full Name
Username: _____
Age: _____
Height: _____
Span: _____
Account Created: _____

Follow Button

User's ascents are displayed here

Footer

Followers

Frame 5

Home Log Ascent Add Climb Username

Followers

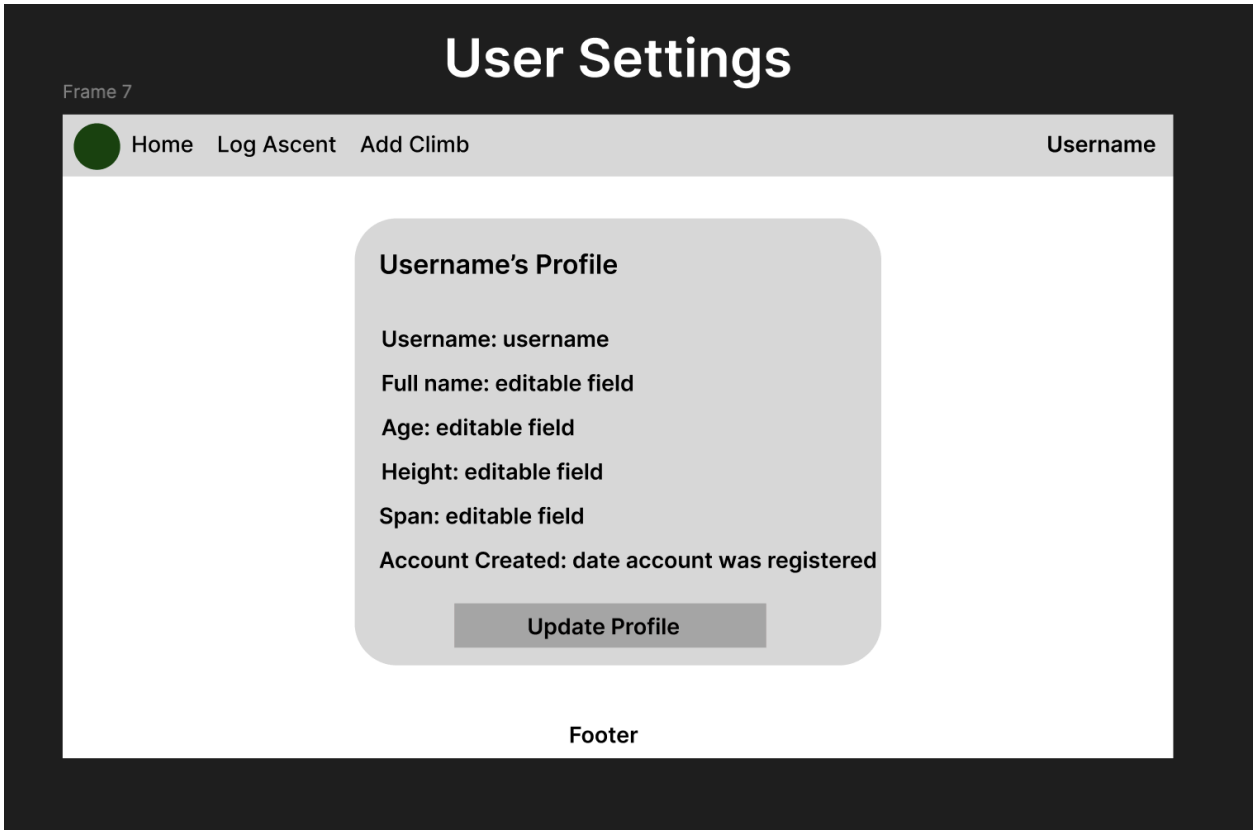
Following

Full name (username) unfollow
Full name (username) unfollow
...

Followers

Full name (username) unfollow
Full name (username) unfollow
...

Footer



Test Results:

For our user tests, we had a user attempt the following:

1. Register and log into the site
2. Add a climb
3. Add an ascent
4. Follow a user from the feed

Our findings were that our user (A roommate of one of the group members) could easily navigate the login and register pages, which is to be expected as they follow the standard username/password format. Our user was also able to add climbs and ascents easily, although the relation of climbs to ascents to users was not as obvious as we initially thought it would be. When our user attempted to follow a user from the feed, it was clear that given more time and a wider scope, reworking the 'follow' feature and having a toggle follow button on the cards in the feed (rather than just being able to add follows, and then

click into the followers page to remove follows) would have been more intuitive for the user. Our user was able to navigate through the intended actions we expected, but due to the setup of some features, an outside perspective showed us that in the future it would be good to rework follows page, and to maybe have more informational text on the add climb/ascent pages to better explain the relationship between each object users could add.

Results to our test spec can be found on the readme of our github page:
https://github.com/elblair/Acendify

Deployment:

Link: https://acendify.ddns.net/home

This website was deployed on bare metal hardware that resides on the rack in my living room. Since I run multiple web servers from my house, it is first run through a web proxy called Nginx proxy manager which transfers the requests to the specified server. It is deployed through the docker environment. The process of deploying the server is as follows: first pull the newest code from the github repository, then build the docker containers for the web server and host the container for the postgres db, this is using docker compose. Then I obtained a DNS address from No-IP for free that would handle dynamic DNS in case my home IP changed. Once I had this DNS name I added a configuration to the proxy server to forward all incoming traffic that requested that DNS name to the Raspberry Pi that is serving the data. I also used the built in SSL features to get a certificate so that the site could use HTTPS.

We chose to not use Render for deploying our application as we had some trouble with the CI/CD aspect, it was not pulling the right commits and not when we wanted it to. This led to it crashing and not working properly for us, so we decided to move to bare metal deployment.