Richard Wang
Sprint 4 reflection

What I did

For sprint 5, I implemented:

The leaderboard system which shows the top 100 all time top users by total experience, and a tabular display showing top daily, weekly, monthly users by experience gain.

The achievement system which gives the user cosmetic achievements upon milestones like first 5 habits completed, or a high habit streak. Using the Django Messages framework, the user is notified when they get an achievement. Then the achievement is shown on the user's profile.

The ability to sort the habit list by completeness (completed, skipped, or incomplete). Habits marked important are always shown first.

Daily login bonus and streak, which grants the user exp upon logging in for the first time of the day. A 7 day streak grants a greater exp bonus, while not logging in for more than a day resets the streak.

Daily spin minigame bonus, which lets the user to spin a wheel for a random small amount of exp.

Reformatted the navigation bar to have a user drop down on the top right, letting the user see some stats, go to their profile, or log out.

Redesigned the profile page, showing achievements and habit statistics. Also created a user profile page, letting anyone see another user's profile. Added a privacy toggle, which blocks non friends from seeing each other's detailed information by default. Also added quick links to copy or let the user share their profile on social media.

Improved the friend feature by adding friend requests, making the friend system a two way request. Also added searching users and viewing their profile.

Improved the site user interface throughout with Bootstrap and Font Awesome icons.

Deployed the project with Amazon EC2 and set up a valid SSL certificate.

Richard Wang
Sprint 4 reflection

Challenges

I initially wanted to use a deployment service like Render or Firebase, but decided to use a EC2 virtual machine. This way I could have more flexibility and control with what I could do with it. The first step was to mirror the repo, but because the history contained large video recordings, it was annoying to do. I did not want to commit the large videos without encoding/compression, but I didn't commit them first, I only pulled, compressed, and pushed them later. Now the files are on the history and we are not allowed to edit Git history, and Git Large File Share costs money. So instead I just used SCP, which was less than ideal but works.

Then I had to install all dependencies, set up the virtual environment, and serve static files. Next I used Gunicorn as the WSGI server and Nginx as the proxy. This was good as it was now online, but it was HTTP and accessible via only IP. Fortunately my parents have a domain already owned, and with that I could obtain a valid SSL certificate with AWS Certificate Manager and serve the site with secure HTTPS.

What I learned

I think by choosing the harder route and deploying via the EC2 instance, I learned a lot more than I would have if I used a platform. I got more hands on experience with server setup, dependencies, configuring the environment. It took a lot more time but I got a better understanding of what I was doing and why I was doing each step.

What I could improve

Not solely for this sprint, but in general, I think I really underestimated how much work went into a website for frontend design. Even though I worked really hard, it feels like our project doesn't look as good as a modern website, which is to be expected since it's only a class project, and we are building everything from scratch, and not with templates. If I were to redo this project, I would really research what kind of tools are at my disposal so I don't need to do a lot of redundant work.

Overall I really enjoyed this project and feel like I learned a lot, while also understanding I have so much more to learn about web development.