

Fitly Application Documentation

Description

Fitly is a fitness application that can be used to log their workouts into an entry. Users can add fitness goals and see if they met it post workout, and they can see the overall summary of the total time spent and their lifting stats. Users also can log multiple workouts in a day and can edit them anytime at ease. This is level 1 implementation for my Senior Design Project. By the end of April 2026, the app will have many more functionalities like a chatbot, meal plan tab and a body silhouette to map the muscles, a workout would be targeting.

Interviewing

Interview 1: I interviewed Jennifer who goes to the gym regularly and asked her what functionalities she would want to see in a logging application. She said she uses my fitness pal to track her food and workouts so some of the same functionalities would be beneficial.

Interview 2: I interviewed Matt and he said he doesn't to use any apps because he prefers something simple and straightforward so that he doesn't spend quite a lot of time figuring out the application.

Interview 3: I interviewed my friend Jazzy who doesn't work out regularly and she said she would prefer a simplistic app like a diary of sorts that she can write all about her day and her workouts.

Sketching

I drew my initial sketches and implemented my app based on it. One functionality that doesn't look the same is today's date not being in the left pane. I couldn't figure the positions out so I kept the date centered with the welcome message.

DATE	log today	Previous Logs
<p>Welcome User</p> <p>Active since —</p> <p>Log your entry</p> <div>Log</div>		

DATE	log today	Previous Logs
<p>Activity</p> <div> <div> <div>5 Cardio</div> <div>4</div> <div>3</div> <div>2</div> <div>1</div> </div> <div> <div>Duration</div> <div>Number</div> <div>Short text</div> </div> </div> <p>Write your thoughts</p> <div>logText</div> <p>Mood</p> <div> <div>😊</div> <div>😐</div> <div>😞</div> </div>		

DATE	log today	Previous Logs
September 19 2025	<div> <div>Mood</div> <div>Fitness</div> <div>Duration</div> </div> <div>Entry</div>	<div> <div>Mood</div> <div>Fitness</div> <div>Duration</div> </div> <div> <div> <div>August 25 2025</div> <div> <div>Mood</div> <div>Fitness</div> <div>Duration</div> </div> <div>Entry</div> </div> <div>Edit button</div> </div>

Feedback

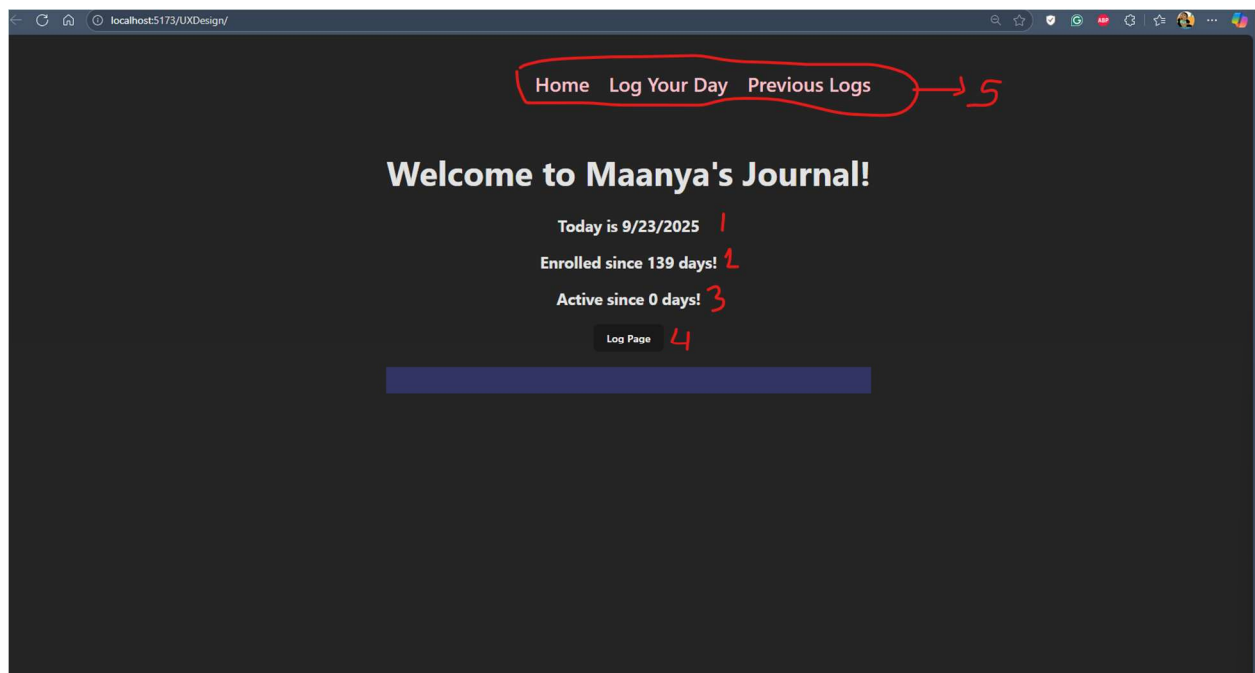
Feedback 1: After I showed my app to Jen, she said it's cute but she'd still use myfitnesspal. I am hoping that she will like it when I develop this project further and have will have more functionality beyond the sketches.

Interview 2: Matt liked how simple and straightforward my app was and he was excited to see the final outcome.

Interview 3: When I showed Jazzy the sketches, she was a bit concerned and then she suggested some color schemes because she is a communications major. She suggested I keep the app in neutral colors but I find them boring so I implemented my own color scheme.

Interface

Features and Controls

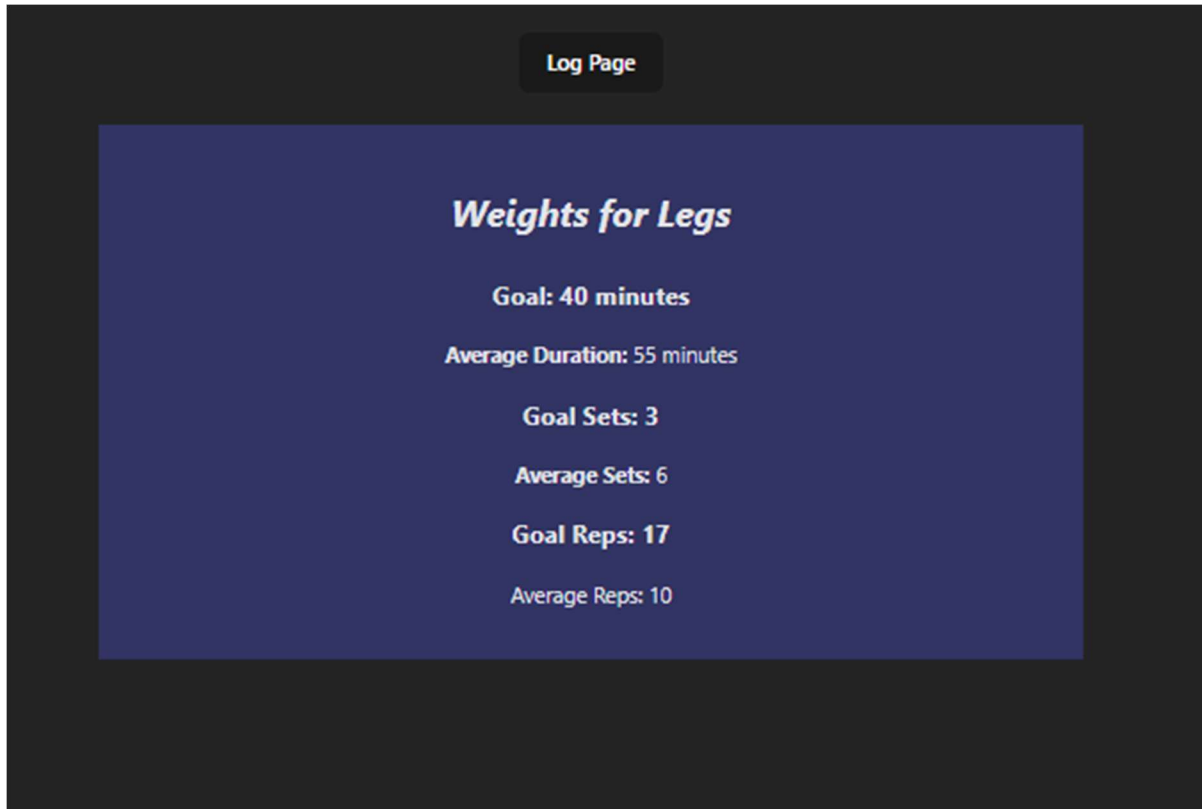


Things that are displayed on the homescreen

1. Today's date is today's date
2. Enrolled Date shows the number of days since you enrolled into the application
3. Active Date shows how long you have been logging for
4. Log Page Button takes you straight to Log Your Day page

5. Navigate through Nav Bar to go to Log Your Day to log or view entries or go to Previous Logs to view or edit your logs

Summary Overview



Contains the summary report of the averages of the log metrics the user may have inputted for every activity.

[Home](#) [Log Your Day](#) [Previous Logs](#)

Enter Your Activity

Pre-Workout Goals

What's the goal duration?

If lifting? How much weight?

If lifting? How many goal sets?

If lifting? How many goal reps?

After Workout Stats

How long did you work out for?

How much weight did you lift?

How many sets?

How many reps/set?

How do you feel after your workout?

Additional Notes

Your entry

The fields to log your entry consists of workout date, activity type (you can select a predetermined list or make your own), goal metrics (time, weight, sets, reps), actual workout results (time, weight, sets, reps), mood, and additional notes. Once you're done filling out the fields, you can hit submit to display the entries and update the summary report in the home page.

[Home](#) [Log Your Day](#) [Previous Logs](#)

Your previous logs

Entry 1

Date - 2025-09-23

Activity - Weights for Legs

Mood - Happy

Goal Duration: 40

Duration: 55

Goal Weight: 40

Actual Weight: 50

Goal Sets: 3

Actual Sets: 6

Goal Reps: 17

Actual Reps: 10

Extra Notes

Edit

When you want to edit, you can click the edit button

[Home](#) [Log Your Day](#) [Previous Logs](#)

Your previous logs

Edit Date

09 / 23 / 2025

Edit Activity

Weights for Legs

Edit goal duration

40

Edit weights

40

Edit goal sets

3

Edit goal reps

17

After Workout Stats

Edit workout duration?

55

Edit weights

50

Edit sets

6

Edit reps/set

10

How did you feel after your workout?

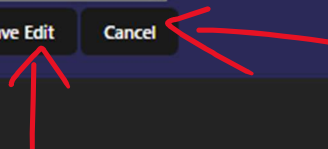
Happy

Additional Notes

Additional Notes

Save Edit

Cancel



You can edit all the fields and hit submit to update the log entries or you can cancel it and pretend it never happened

Implementation

Libraries

I started working on this project on September 15th after my initial interviews. It was straightforward to write the logic for the logging component, but I couldn't get it to route to the specific "Log Your Day" Tab on the navigation pane. I looked through the Svelte tutorial but most of the answers were towards Svelte kit, but the project didn't specify anything about Svelte kit, so I asked ChatGPT, and she told me to install spa-router and then I set up my pages accordingly. So for libraries, I installed svelte-spa-router for routes and im using logs as a writable store from svelte/stores.

Variables and Functions

I had my name as the name variable and I set the join in date for '05-08-2025' and got the current date to calculate the total days, the user has been enrolled for. I also set welcomemsg to true so when its false the welcome message and overview stats don't transfer through when changing tabs and only the routes for the navbar is shown and logs state is transferred through. This makes sure that when its false, the message doesn't repeat over to the other tabs.

Then, for the overview summary, I found out that if you use \$, it creates a reactive state so that activitySummary function updates when there are new entries in logs. It loops through each entry and groups them by activity type and then calculates totals such as total time, goal time, goalsets, sets, goalreps and reps for the averages so user can see the workout stats metricised. It also counts how many unique days have activity records for the active count.

Navbar

Basic navbar that has a page for Home, Logs and PreviousLogs. I wanted the date to be on the far left but it wouldn't format properly.

Log Your Day Tab

Log entry variables

Logs is imported from the stores and then I set up local variables like date, time, goal, activity, mood, notes, goalweight, weight, goalsets, goalreps, sets, and reps. These are initialized to empty strings or zeros so when user inputs the entry, the fields get updated without any errors. The listactivities array holds some workout activities that can be selected from a dropdown or the user has an option to not log any of it and choose whatever activity they want.

The function handleSubmit() creates a object for entries and then calls logs.update(...) to add this entry into the existing list of logs if there is any. Then it resets the fields so the user can start again.

Log entry format

The layout has Pre Workout Logs and then After Workout Logs that the user can put in the stats to see if they have achieved, exceeded or failed to reach their goals.

Once submitted logs are displayed below in a colored box and will also be displayed in PreviousLogs tab. The \$ in logs makes it react automatically and updates without needing extra code. Each entry shows the date, activity, mood, goal vs actual duration, weights, sets, reps, and additional notes.

Previous Logs Tab

Previous Log entry variables

Logs is imported from the stores and then I set up local variables like date, time, goal, activity, mood, notes, goalweight, weight, goalsets, goalreps, sets, and reps incase the user wants to edit it. These are initialized to empty strings or zeros so when user edits the entry, the fields get updated without any errors. The editEntry object holds a temp copy of the log values so the original data in store is not mutated directly. The editLog(ei) function takes the index of the entry to edit and copies its values into editEntry.

The saveEdit() function then updates the store by creating a new array of logs replacing only the entry at editIndex with the edited values, and returning the updated list. The cancelEdit() function pretends this never happened (easiest to implement).

Previous Log entry format

Logs are displayed below in a colored box. The layout has Pre Workout Logs and then After Workout Logs that the user can edit by clicking on the edit button.

Use of AI

I used AI to understand the underlying mechanics of svelte like the spread operator `{... }` and changing my string input into numeric through `+string`, etc. The strengths of Chat GPT are immense and there were at times, she picked up what I was trying to do and gave me the full code and even though it was tempting to use it, I really wanted to learn Svelte and master this application myself. There was also Copilot embedded in VS code that helped me fill out repetitive items like inputs, etc.

Future work

Like I mentioned earlier, this project will be a template for me to work on my Senior Design, Fitly. I will need to implement a BMI body silhouette page in the home screen and move the overview tabs to another tab called "Summary Report" I will include a lot more graphics and visuals compared to the plain text I have right now. On top of that, there will be more tabs for meal plans, workout tutorials and a better designed page for logging in their workouts. Lastly, there will also be a chatbot that can be utilized to give the user workout and meal plans in an instant through curated data, I would train it on, in case the user did not want to scroll through and find it themselves.

Include a 2-3 minute demo video, showing your interface in action.

<https://www.loom.com/share/d4e45ebd33ad49a9827b56c4b3b9586d?sid=b1b9b50c-7558-41ce-b1ef-f8f891c51537>

Include a link to your source code on github and a link to the publicly hosted application.

[mangopudding03/UXDesign at master](https://github.com/mangopudding03/UXDesign)

<https://mangopudding03.github.io/UXDesign/>