Title: FinLog

Who: Aditya Dhar (github: adidhar), Claire Ricca (github: claire-ricca42), Fred Zorgdrager (github: FredZorg), Juan Pleitez (github: JdP01), Michelle Lisowski (github: michelleLisowski)

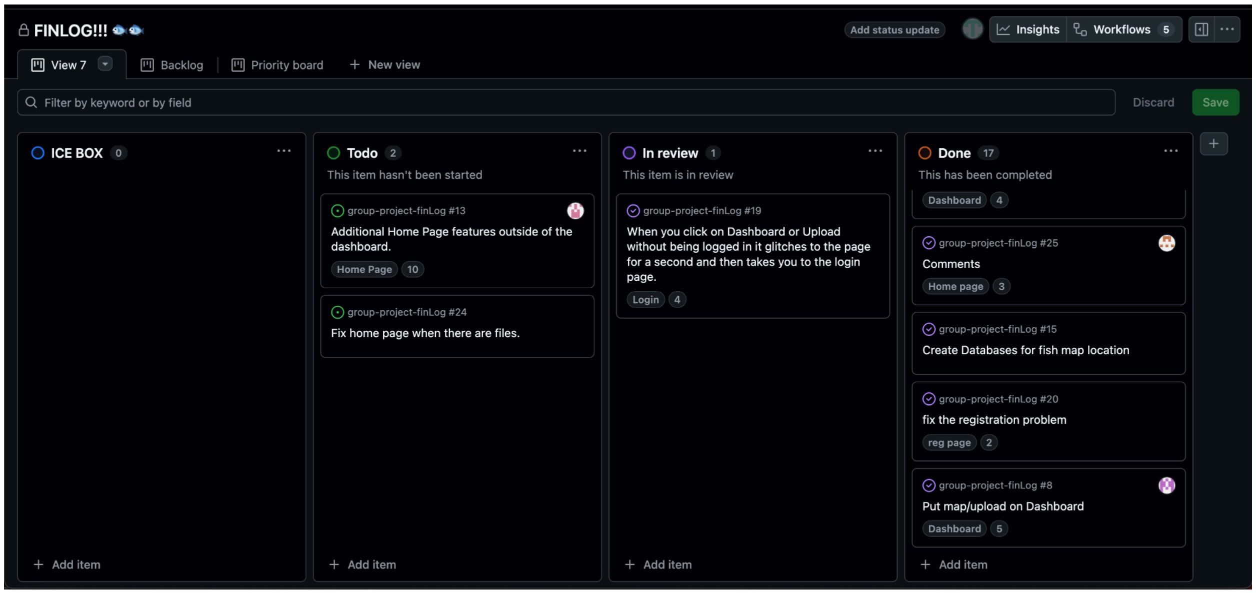
Project Description:

FinLog is a web-based fishing log that lets users record and analyze their catches, tracking species, weight, length, and location. There is an optional AI feature that identifies fish from photos and provides facts or habitat details. Designed for casual anglers, guides, and tournament fishers alike, FinLog helps you remember details, learn what works where and when, and improve your results on the water. Its clean interface makes logging fast so you can get back to fishing as soon as possible! There is a map feature so that users can see where they've caught their fish in the past and see other users' pins as well. Users can also comment on other users' posts in a social media fashion to help uplift each other and support each others' fish. Additionally, for the social media aspect, there are moderator and admin roles that ensure that there are no posts that are not appropriate for the platform.

VCS:
https://github.com/CU-CSCI3308-Fall2025/group-project-finLog

Our repo is organized slightly differently than specified, but we discussed with our TA about what would be most effective for us and we agreed on the structure that we used.

Project Tracker:
https://github.com/orgs/CU-CSCI3308-Fall2025/projects/7



ICE BOX 0

Todo 2
This item hasn't been started

group-project-finLog #13
Additional Home Page features outside of the dashboard.
Home Page 10

group-project-finLog #24
Fix home page when there are files.

In review 1
This item is in review

group-project-finLog #19
When you click on Dashboard or Upload without being logged in it glitches to the page for a second and then takes you to the login page.
Login 4

Done 17
This has been completed

Dashboard 4

group-project-finLog #25
Comments
Home page 3

group-project-finLog #15
Create Databases for fish map location

group-project-finLog #20
fix the registration problem
reg page 2

group-project-finLog #8
Put map/upload on Dashboard
Dashboard 5

For our dashboard, we assigned specific tasks to each other in Github during our in person meetings so there would be no confusion about who would take on which task.

Video:

https://youtu.be/czKJzb1EKQ0

Contributions:

Aditya:

For this project, I implemented the admin dashboard with role-based access control, allowing moderators to approve/flag posts while restricting deletion to admins through backend middleware authentication. I then built the likes feature across the full stack, creating the database schema, API endpoints for like/unlike toggling, and frontend JavaScript to update heart icons and counts in real-time. Finally, I enhanced the home feed filtering system by fixing the "Trending" filter to sort by comment count, and adding a "Size" filter for catches by weight.

Claire:
For this project I worked on consolidating and organizing our website format, as well as adding dynamic post functionality. This allowed for users to see other users' posts in real time, and their own posts were added to the home page when uploaded. I created the data structure for posts, built backend routes to create and retrieve posts on both the home page and the user's dashboard, and added a "feed" route to return posts from all users, so they would be visible by anyone registered in our database. On the frontend, I updated the home page to dynamically render posts using JavaScript and SQL to handle retrieving and storing data to and from the database.

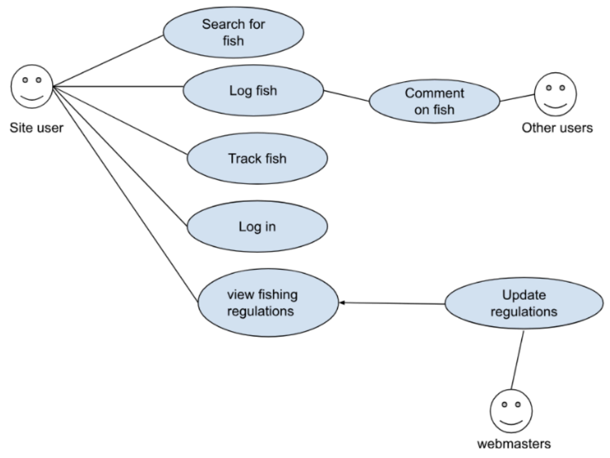
Fred:
While working on this project there were a few main areas that I contributed to. The primary area that I spearheaded was getting our project working on render, this took up a decent amount of time as I had to change some of our infrastructure to allow for the project to run in render. I also worked a lot on the UI building out the overall structure for the home page and making it so that all users could access all of the different areas while still having a nav bar on the side. Also on the home page I created the search functionality that allowed users to search fish and post titles. I spent the rest of my time adding smaller features like users being able to delete their own posts and debugging code like when we had a large linking issue which made it impossible to login or navigate the page or when our Database was not populating itself with the correct data.

Juan:
On the project, I started my contribution on setting up the first version of the leaflet.js framework to have a live map displaying users' live catch locations. I set up the first working version of the database for the most basic information such as users metadata and media posts, and before the team continued to grow it to its current state. My next contributions involved refactoring the frontend to update the UI and overall website theme and feel. Finally, I implemented a working

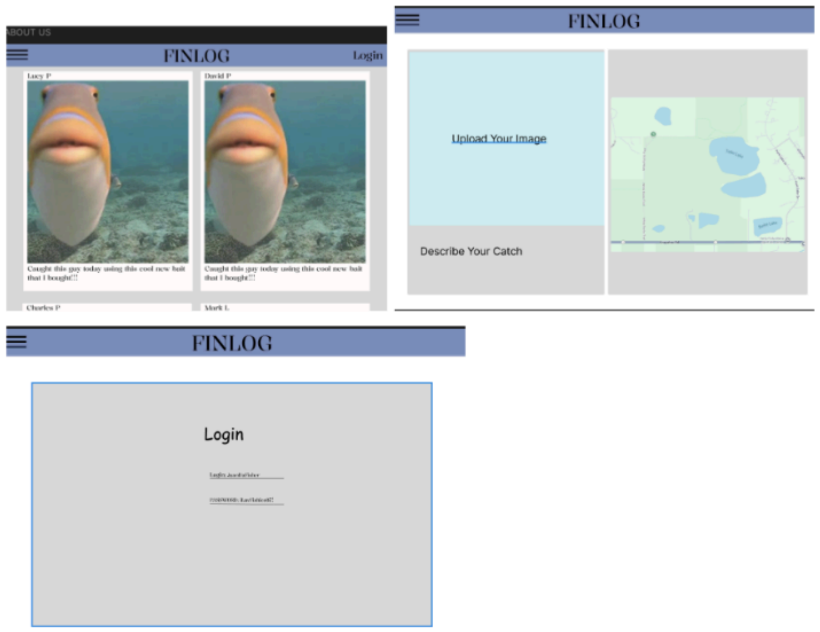
api call for google gemini to inform users about their catch and help moderate posting non-site related content.

Michelle:
For this project, I mainly contributed to the Home and Upload pages. I helped to make the upload page functional, as well as made the comments database and comment functionality for the Home page. I worked on these things through SQL to make the databases and HTML and JS for the functionality of the comments. I made tests in chai for our endpoints, and helped to coordinate our team meeting times every week as well as helped to make sure everyone was on the same page in terms of development. I also took notes in the weekly team meetings and defined goals for each week of development.

Use Case Diagram:



Wireframes:


ABOUT US
FINLOG
Login
Lucy P
David P
Caught this guy today using this cool new bait that I bought!!!
Caught this guy today using this cool new bait that I bought!!!
Charles P
Mark L
FINLOG
Upload Your Image
Describe Your Catch
FINLOG
Login

These are the wireframes from our initial conceptualization of the website. The upper left is the home page, the upper right is the map/upload page, which we ended up splitting into their own pages, and the bottom image is the login page.

Test results:

For Lab 10, we chose to test a User case where they upload a catch.
In the test, we had different features being tested than what we ended up implementing, but the general testing remains the same. We replaced the “Count” feature with “Weight” and the Date feature to be automated instead of manually inputted. The “Count” feature ended up being manually implemented on the dashboard, and the users can only upload one fish at a time. For dashboard tests, we exercised the user-facing flows (log in, open dashboard, apply species and date filters, and favorite an item) while accounting for a couple of implementation changes: the original “Count” field was swapped out for a “Weight” attribute, dates are produced/handled automatically rather than entered by hand, and users may only upload one fish image at a time with any manual count now done on the dashboard UI. The test validated that the dashboard loads, filtering and date scoping return the expected results, favorites persist across refreshes, and profile edits show up on the user’s sightings. To test admin features, we focused on elevated actions like flagging, deleting, and approving posts. We verified role changes persist in the database, admin edits immediately reflect in sighting details, deletions return 404 on subsequent GETs, and non-admin access to /admin is blocked with the appropriate status. These scenarios were executed under the same product differences (Weight instead of Count, automated dates, single-image uploads) and behaved as expected. Our tests were run locally (via docker compose up serving the Node app on port 3000 and Postgres). We used Chai for API endpoint testing like in Lab 10. Overall, the runs passed: endpoints returned expected HTTP statuses, filtered listings matched criteria, admin actions were effective, and uploads were stored under the expected paths and in the correct databases. Any minor differences were limited to the planned feature substitutions (Weight and Locations) and the single-upload constraint.

Deployment: https://finlog-7lhk.onrender.com/