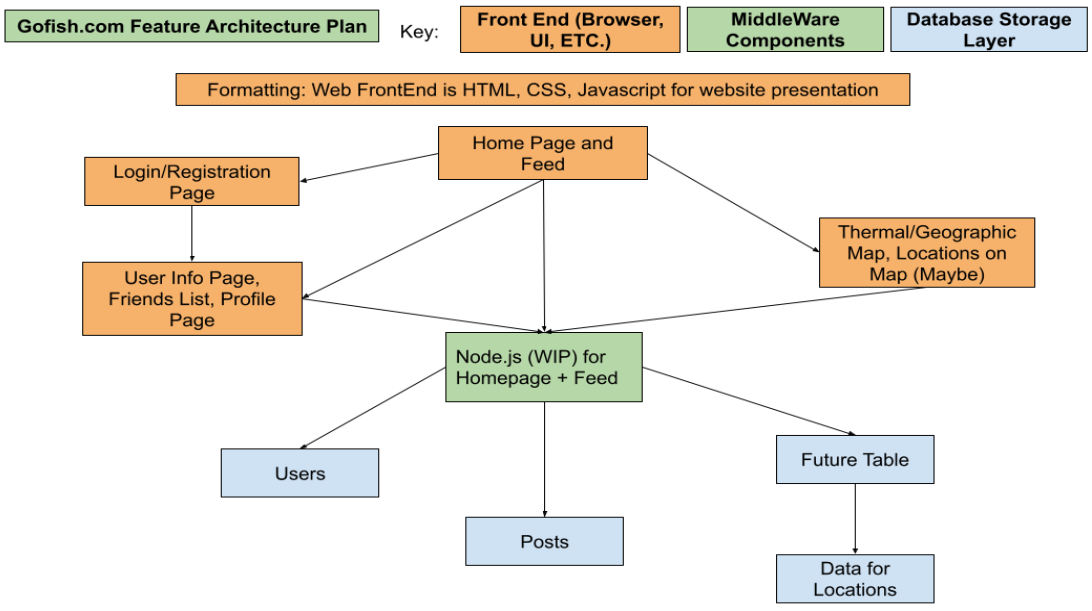
Team 017-04
Project Milestone #4
3/20/2022

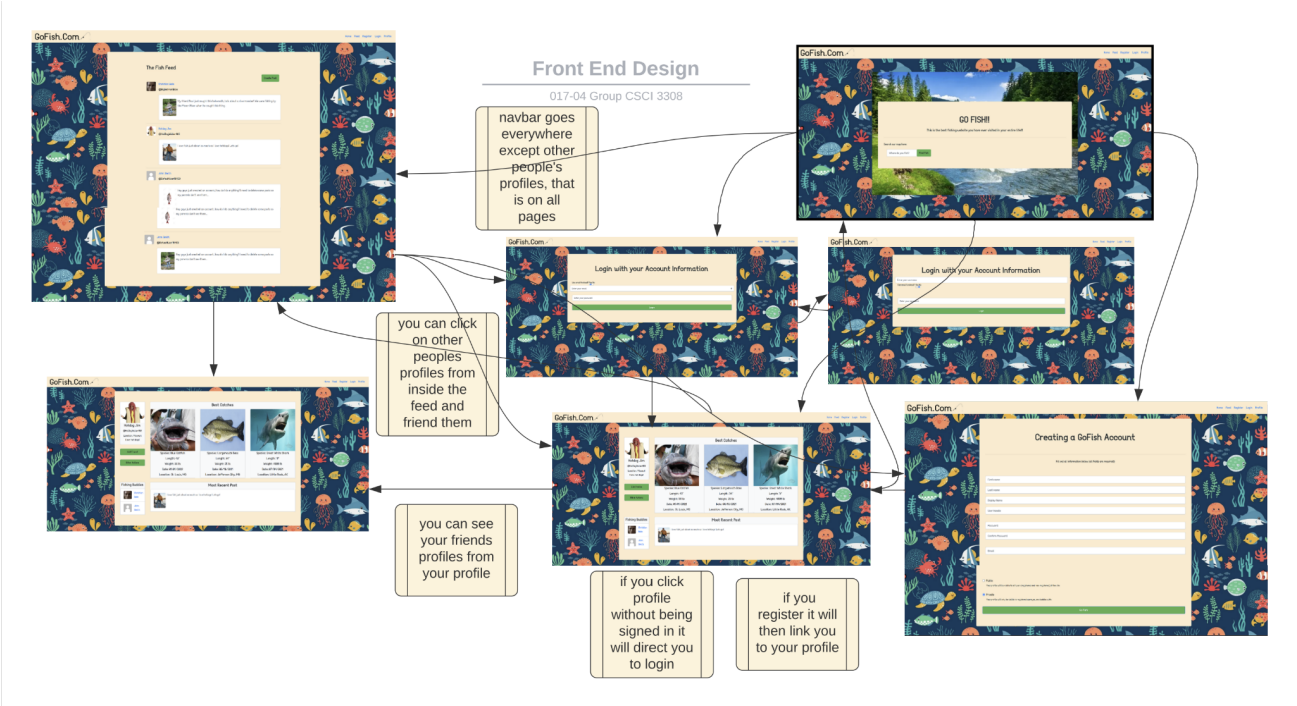
Revised List of Features (Ordered by Priority)

- Profile Creation/Log-In
  - Users can register a profile for the website. With a profile they can follow other registered users and access other features of the site.
- Fish Log
  - Users with a registered profile can log the fish they catch, including data such as length, species, and where it was caught. They can choose to publicly display some or all of their caught fish and can highlight their best catch.
- Text Posts
  - Registered users can make text posts akin to a traditional social media site to discuss fishing-related topics with other users.
- Feed
  - Any user can see posts from a specific location and can save specific posts to their liked page.
- Friend List (Fishing Buddies)
  - A user can use a friend code to add other users as “Fishing Buddies”. Each user can find a list of their “Fishing Buddies” in their user page and use it to link to their friend’s user pages.
- Best Catches
  - On their userpage, a user can see a list of the best catches they reported.
- Fish Map
  - Map available to both registered and unregistered users aggregating publicly available fish location data recorded on the site. The idea is to provide users with an idea of fishing “hot spots”

Architecture Diagram



Front End Diagram



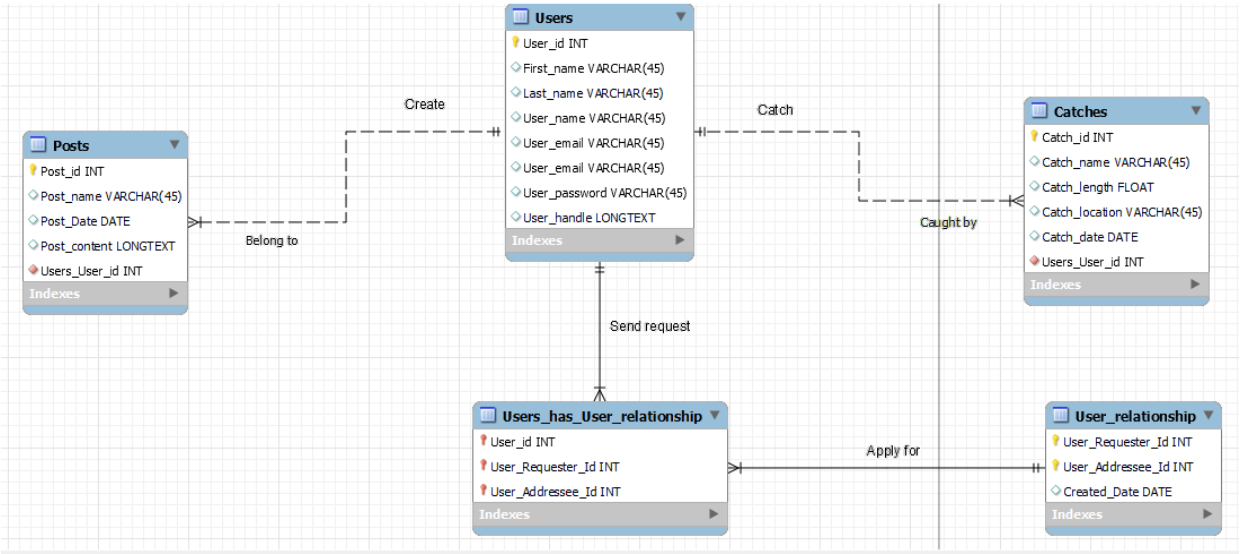
To see this in more detail, you can look at the html files All_Project_Code/views in the latest version of the commit, of which these are screenshots, or view a larger version of the image here: https://drive.google.com/file/d/11Ydv2nW1s9zOTcUBSb64StQpkFNDSWPA/view?usp=sharing

These are essentially what was shown in the Midterm Demo, but with buttons added for adding a friend and “other actions” (like adding a catch) on the user profile page and a button to create a post on the feed page. In addition, this design assumes we are not able to get a dynamic design for the navbar working.

Web Service Design

We are planning on utilizing a Google Maps API in order to be able to generate a map that we could overlay a heatmap type function over and utilize. This is still a work in progress so we have not gone further than identifying this API as something we may potentially use.

Database Design



This database design will use PostgreSQL. Passwords stored will be hashed (and salted, if that can be figured out) as a means to maintain security. Users logging in will be authenticated by comparing the hash of the entered password to the stored hash - the application will not store or transmit plaintext passwords between the server and the front-end.

Challenges

One potential challenge may be in implementing our fish map. This feature is outside the scope of our main website concept (as a social media site for people who like fishing) which means it is the least crucial of our features to implement. In addition, it is going to require the use of an outside API (specifically that for Google Maps). If we find that we don't have the time to implement this feature, we will likely just drop it in order to work on our social media functions instead.

One other challenge will be in converting the html we currently have written for the website into ejs templates and retaining or adding functionality. While ejs would be very useful to the project (in particular for making maintaining our code easier, since we could section off the repeating bits like the navbar), the group has generally found this concept and the associated lab to be one of the more difficult ones this semester.if we find that the conversion is not progressing, we could decide to not use ejs and instead use more limited NodeJS with the html we have created for each website page. However, this would come with the drawback that the project would be more difficult to maintain, especially in terms of a consistent styling.

A final challenge we have to face is in the design of the navbar. Currently, we are hoping for our navbar to change depending on whether or not a user is logged in - if they aren't, then the navbar will have buttons to register or log in. If they are, then the navbar will have buttons to link to their profile and a log-out button. The main challenge here is figuring out how exactly we will

be able to determine if a user is logged in or not, likely by using the NodeJS module “session” (which should track the user with cookies), which is not something we are too familiar with. In addition, doing thisIf this becomes too difficult, we could maintain a static navbar design and have an error page for users not logged in if they clicked on the profile page. That, of course, would certainly lead to a less than ideal user experience and would be best to avoid if possible.

Individual Contributions

Most recent commits link:

- Spencer - I have been working on converting the log-in page to ejs templates as well as establishing some of the partials we will need in converting the site to ejs, such as the header and the navbar. My second to last commit added a folder for the ejs files and started to transfer the log-in page to that format: https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-04/commit/36d07f844cadfafc5c3eb23ce6f3a2e42d4bdfbb. My last commit added one line to the server.js file that I forgot to commit in the previous one: https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-04/commit/ef319ad2c2838272cffd04f73f7b365e34cb20e1
- Yuhe - I worked mostly about the database setup and organization. We discussed what features our website might need and I am trying to create appropriate way to create tables and the columns. https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-04/commit/ceaabed7bd79bf15ccbeaa5798c58f33bf0a43e7
- Matthew - I most recently committed the initial input and formatting for the registration page in addition to basic input error prevention such as matching password confirm password, email format checking, and vulgar language check for username and userhandle. https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-04/tree/321a9792320fede6c533adcc986d2e5afaac13ea
- Victoria - I worked on cohesiveness on the front end html and css, formatting and design of all of the pages, and the front end design diagram in this milestone! I added new places for features that will be implemented, such as adding a friend, editing your profile, and creating a post! The new pages are 2home2.html, 2login2.html, 2me2.html, 2profile2.html, 2register2.html, and 2feed2.html! https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-04/commit/e0c92089992740d9d5359e7b5123da909381bae5
- Caelus - I worked on the user page a bit adding some architecture for adding friends and catches features we plan on implementing. I also made an ejs file for the user page with a basic outline to add catches and friends when the Node.js features are ready. https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-04/commit/41ffb1f97e222cad8d88d5c02b0a20a7a52449f4
- John - I have been working on the registration page. Currently just going through some tutorials online to work out how to do this.

https://github.com/cub-csci-3308-spring-2022/csci-3308-spring22-017-04/commit/b8e74757fb8a1543c54da2cb3904cc57db013036

We opted to use a collaborative google doc to continuously write in new features/explain what we are doing during every meeting. We just recently switched roles to make sure everyone was able to participate in different parts of the development process. **This was our current standing as of 3/16/2022**:

- **NodeJS (Team)**
  - **We need to set it up (Most important feature ATM).**
  - **Convert HTML to ejs**
- CSS Features (Victoria)
  - Need Overall CSS file to apply a template/format all the webpages to align
  - Decide on format look and aesthetics
  - Dropdown menu
- Database Features (Yuhe)
  - Need all tables established
  - Add columns to posts table
    - User id to connect what user made post
    - Column for content of a post
  - Add data table for registration info that will be used with login info
  - Add table for catches
    - Image column to store picture of catch
    - Name of fish
    - Length of fish
    - Location fish was caught
  - Add columns to users
    - Profile picture
    - Array of user id's of buddies
  -
- User Profile Page Features (Caelus)
  - Friends List
    - Add friends by friend code
    - Will be linked to users page
  - Display posts under page
    - Contains images, text, contains link to post itselfs
  - Needs users table (established), "post_content", table for friends list, catches table, etc.

- Go-FishFeed (Matthew)
  - Consistent background
  - Display posts
    - Posts contain image, text, user profile picture, user profile handle, user's actual name

        - Post links back to the user who posted it
    - Display posts under page
        - Contains images, text, contains link to post itselfs
    - User information table? Pulls posts from table to upload to display on page

- Gofishhome (Spencer)
    - Heatmap (maybe)
    - User search?
    - Reformat to be more of a landing page to direct users to log-in/sign-up
- Login (Spencer)
    - Fix navbar alignment
    - Connect login page with database
        - Store user information into the database, username, password, name, email, etc…
- Registration (John)
    - Database table for registration inputs
    - Add email check functionality better than just looking for “@”
    - Receive user input into a database
    - Add additional filter protection to inputs
- Userprofile Settings/Change (Victoria)
    - Feature to add catches
        - Image, text, etc…
    - View your friends list
        - Add and remove friend
    - Database for profile