Butterfly Longevity Tracker

Weekly Report 1 (1/1/25 - 1/30/25)

Project Information

- Project Title: Global Butterfly Longevity Tracker
- Group Number: sdmay25-03
- Client: Nathan Brockman
- Advisor: Maruf Ahamed
- Team Members:
    - Alex Herting - Full-stack Engineer
    - Andrew Ahrenkiel - Full-stack Engineer
    - Charles Dougherty - Frontend Developer
    - Jaret Van Zee - Backend Engineer
    - Carter Awbrey - Visionary

Summary

In this time period we have been focusing on getting screens done that were not included in our first prototype. Our prototype 1 was successful, and we have been working on getting the rest of the screens integrated into that design. The frontend design and screen creation aspect is taking up most of the time as we are accounting for every screen size (i.e., phones, computers, laptops, ipads, etc.).

For the backend we haven't added any additional features during this time period. Instead, we have worked on planning features for the upcoming semester and discussed changes to our app's design given the feedback we received by the faculty review panel presentation relating to our database plan.

Accomplishments

- Alex
    - Continued work on making the UI responsive through css and html code changes
    - Code reviewed for Andrew on his coding changes prior to merging
    - Created new screens that were not included in prototype 1
    - Collaborated on brainstorming ideas on how we can set up the tagging system in the frontend for new clients.
- Andrew
    - Continued work on making the UI responsive through css and html code changes
    - Code reviewed for Alex on his coding changes prior to merging
    - Created new screens that were not included in Prototype one
    - Collaborated on brainstorming ideas on how we can set up the tagging system in the frontend for new clients.
    - Created Weekly Availability Calendar to find best meeting times with client and advisor for all team members
- Charles
    - I began developing a report display for butterfly data
    - I started working on importing previous data from the old database to the new
    - Revamping CSS for better readability and future use
- Jaret
    - Review of current code and progress
    - Updating overall environment
    - Research on necessary technologies
- Carter
    - Created Meeting Notes
    - Researched AWS services that we need to use in the future.
    - Looked into transitioning to a relational database

Pending Issues

Front End

- Create the rest of the screens not included in prototype 1
- Set up site navigation
- Introduce responsive designs to account for different screen sizes

Back End

- Configure SSL encryption
- Finish Implement JWT authentication
    - Decide Whether to use Oauth
    - Fingerprinting Database Actions
    - Permissions System
- Unit Tests
- API Tests
- Optimize Database Queries For Better Performance
- Improve Input Validation
- Create API documentation with “Spring REST Docs”
- Consider switching from MongoDB -> PostgreSQL
- Configure long-term hosting site (AWS)
    - Discuss the expected costs of hosting on AWS

Individual Contributions

NAME	Individual Contributions	Hours this week	HOURS cumulative
Alex Herting	Frontend Responsiveness, Frontend Screen Creation, Code Reviews, Communication with Client and Advisor	15	56
Andrew Ahrenkiel	Frontend Views, Frontend Styling and Linking, Code Reviews, Meeting Availability	10	52
Charles Dougherty	Frontend report display, importing old database, CSS revamping	5	42
Jaret Van Zee	Reviewing back-end code, establishing communication, fixing code environment.	4	40
Carter Awbrey	Project Planning, DB research	8	56

Future Plans

For the front end, we will create the base design and functionality for the remaining screens not included in Prototype 1. We will eventually integrate those screens into our prototype 1 design and have a seamless site navigation system. The next step will be to connect each screen to the back end as needed. Frontend Linking will be the first task after the implementation of each view, meaning seamless navigation between screens will be implemented. Backend requests will then be implemented to views that need them.

For the backend, we will finalize our decision on what type of database will work best, then proceed to build optimized queries to retrieve data efficiently. After the database is configured and the queries are built, we will move on to ensuring that the communication between the user, server, and database is encrypted. The long-term hosting site (AWS) must be configured to host the database and site. Unit and API tests will be written and performed to make sure that all functions and requests are functioning as expected. API docs will also be developed to expedite the front-end linking and provide documentation for future development.