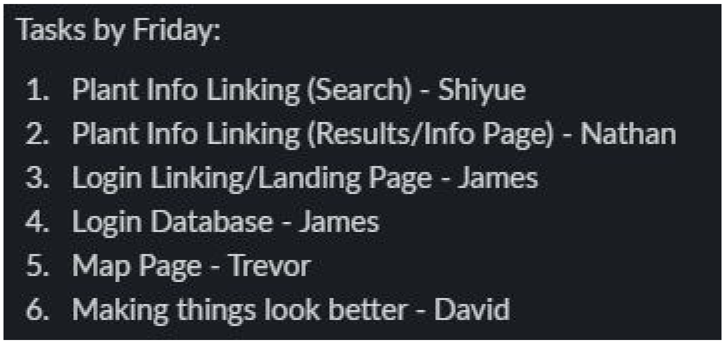
Team 105-7: The Botanizer's
Nathan Henault
Trevor Lana
David Gao
James Burke
Shiyue Zhang

Botanizer Project Milestone 7

- Project Title: The Botanizer
- Team
  - Nathan Henault
  - Trevor Lana
  - David Gao
  - James Burke
  - Shiyue Zhang
- Project Description
  - The Botanizer is a web-based application that allows users to search for and access information about care for various plants. Each page in the app features a navigation bar with links to the main features of the application. Users can search for specific plants by name and features, or fill out a plant survey to be given the closest match based on desired qualities. Users can also visit the FAQ page for answers to common questions about plant care, or visit the Plant Map page for interactive information about the national flowers of several countries. The landing page for the project features a login feature that allows users to create and sign into accounts. The application is run by a node.js server which loads the pages, and links information from the postgreSQL database backend to the client. The plant results page loads each plant's images and information dynamically using this application structure. The interface style is provided by Bootstrap CSS, and Javascript alongside the JQuery Javascript library are frequently used to alter and initialize page layouts. Overall The Botanizer is designed to have a pleasing, user friendly experience centered around plant care needs.
- Project Tracker
  - Gantt Chart: At the beginning of the project, the group created a Gantt Chart to highlight the general design flow of the project. The chart was included in our Milestone 2 submission which can be found here: https://github.com/nhenault/CSCI_3308_Team_105-7_Milestone_Submissions/blob/master/ProjectMilestone2_105-7.pdf

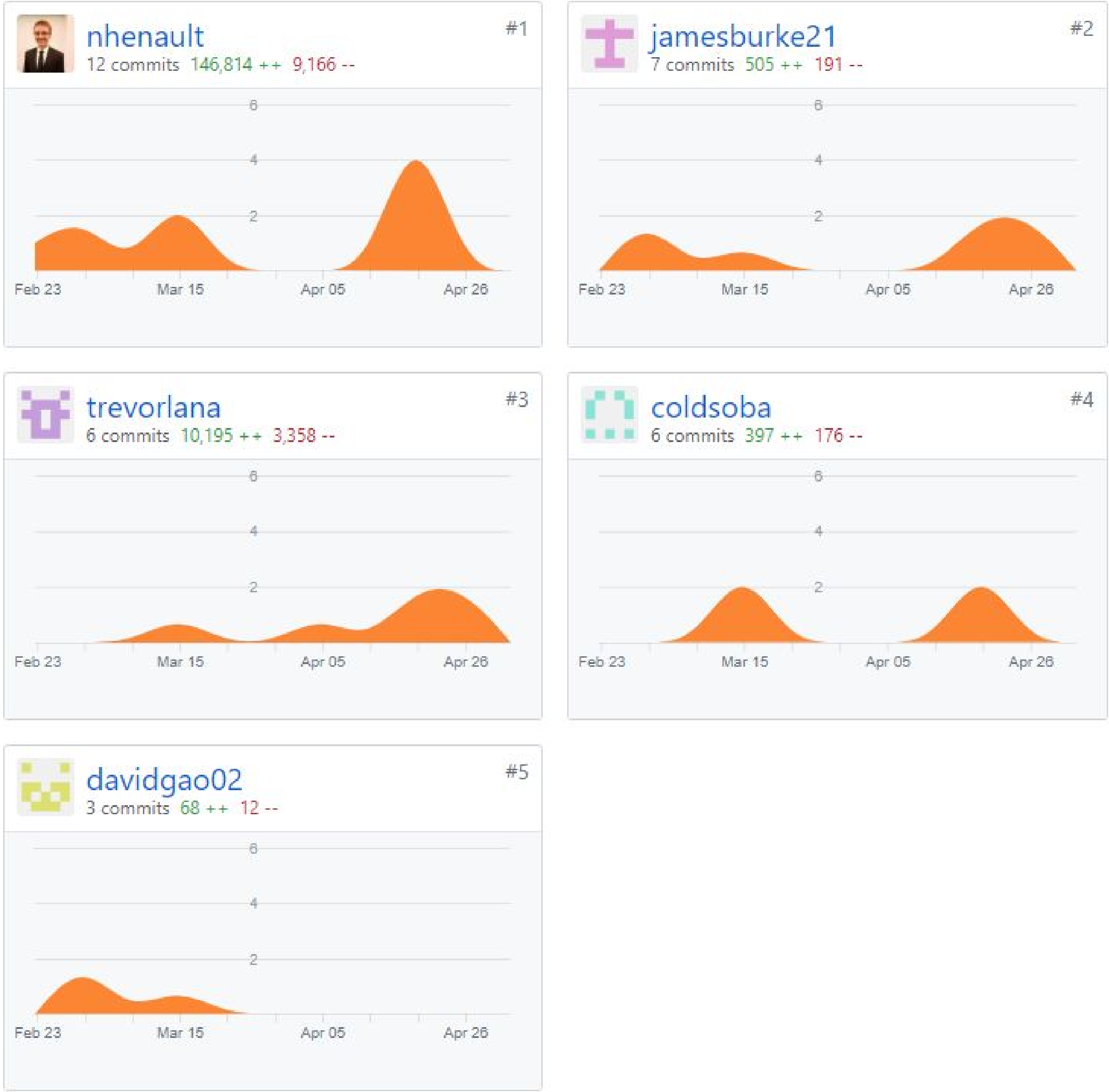
    - Slack/Zoom: As we progressed through the semester, we held weekly meetings where we discussed what progress we had made and what still needed to be completed. These meetings began in person and transitioned to Slack and Zoom during the Coronavirus quarantine. Meeting notes were captured in our team Slack channel, and a list of responsibilities for the week were written down each week. An example of one of these lists can be seen in the screenshot below.



Tasks by Friday:

1. Plant Info Linking (Search) - Shiyue
2. Plant Info Linking (Results/Info Page) - Nathan
3. Login Linking/Landing Page - James
4. Login Database - James
5. Map Page - Trevor
6. Making things look better - David

- VCS
    - GitHub: Our team used two public GitHub repositories for our project, one for milestone submissions and one for the actual project code. Links to both repositories can be found below.
        - Code: https://github.com/nhenault/CSCI_3308_Team_105-7_Project_Code
        - Milestones: https://github.com/nhenault/CSCI_3308_Team_105-7_Milestone_Submissions
    - To test the functionality of our software, a list of test cases was designed for project milestone 5. This test case document can be found here: https://github.com/nhenault/CSCI_3308_Team_105-7_Milestone_Submissions/blob/master/ProjectMilestone5_105-7.pdf
- Contributions
    - Commit contributions



nhenault #1
12 commits 146,814 ++ 9,166 --

jamesburke21 #2
7 commits 505 ++ 191 --

trevorlana #3
6 commits 10,195 ++ 3,358 --

coldsoba #4
6 commits 397 ++ 176 --

davidgao02 #5
3 commits 68 ++ 12 --

- **Nathan Henault (nhenault):** I was responsible for the initial Front-End development of the plant search page. This involved using HTML with bootstrap formatting to create a search form with optional filters, and an empty table which would be populated by EJS during the linking phase when a user enters a search. During the linking phase of the project, I was in charge of completing the linking for the Plant Information/Results page. This is the page which is loaded when the user selects a plant from their search results. The page contains fields for the plant name, an image of the plant, and the plant's care information. Unfortunately, we were unable to test this linking because the search linking was never completed so we are unable to generate results which the user must select in order to trigger the Plant Information linking and page generation.

  - **James Burke (jamesburke21):** I was responsible for the the plant info page, which is loads care information as well as an image once a user selects a plant from the search page. For this, I used HTML, CSS, and bootstrap. I also wrote the frontend linking and backend for the login/registration landing page. This included using javascript, SQL, and using Postgres. I ended up writing a separate javascript file in order to handle the pop-up modal that appears when a user clicks the registration button.
  - **Trevor Lana (trevorlana):** I was responsible for creating the plant information database which holds information for the various plants that we featured in the project, as well as relative file paths to images of the respective plants to be used on the plant info page. Later on I added several different habanero plants to the database to be used to test search functionality. This database was shared on github by creating a database dumb with pg_dump, then pushing that dump file to our repo. Later on in the project I wrote the Plant Map page, which makes use of a public domain SVG map of the world. Several countries are then highlighted and given on click properties to display information about the given country's national flower. This is accomplished using a JQuery document.ready function which then loops through a JSON object to change tags in the SVG by id. At the end of our project I fixed the main portion of our linking so that the project could be run and loaded via node.js, and each page would be linked to and load properly with its corresponding app.get request.
  - **Shiyue Zhang (coldsoba):** Near the beginning of the project, I wrote the HTML page for the plant matching survey. I used HTML, CSS, & Bootstrap to create a form that would allow users to input plant preferences. I also used Javascript to do linking for the Search page, which enables users to search for plants with specific qualities in the database.
  - **David Gao (davidgao02):** At the beginning of the project I wrote the html for the FAQ page. Later on, I was responsible for making changes and formatting almost all the html pages. I used HTML, CSS, and Bootstrap. These changes include changes to formatting and aesthetics. These changes affected the frontend of almost all the features. Near the end of the project I also spent time trying to debug parts of the code. During the course of the project I worked on a separate branch that was merged to the master branch and then deleted. Also my work caused a merge conflict that we ended up resolving by just having another team member copy my code and commit for me. Due to the separate branch being deleted and the merge conflict, a lot of my commits can't be viewed.
- Deployment Instructions
  - This app is deployed by running a server.js file locally. The user must download the entire repository, then navigate to it in a terminal. The database must also be

    loaded and active in a separate terminal. From the main project directory, the command ‘node server.js’ will load the application on ‘localhost:3000’. Please reference more detailed instructions on how to deploy the app in the ‘How to Use’ section of the README file in the main directory.

- Project functionality video
    - After milestone 6, we made some improvements to the project, allowing us to launch it from a local server rather than viewing the source files directly. A video of our improvements and the current functionality of the project can be seen by following the link below.
    - https://vimeo.com/414584668