Botanizer Project Details

Project Features:

1. Search Input
    a. Website has a search bar where users can input the plant they would like to find and view a list of results.
    b. Functional Requirements: User must be able to input a search term and see a list of potential matches.
    c. Non-Functional Requirements: Have a searchable database linked to the search bar UI element. When users input something on the search bar, the database is scanned and returns all potential matches.
2. Search Output
    a. Displays a list of matching plants from the search input.
    b. Functional Requirements: After input is given to the search bar, a list of potential matches are displayed below the search bar.
    c. Non-Functional Requirements: Returns all matching names from the searchable database linked to the search bar UI element.
3. Results Page
    a. The page which loads when a plant is selected. Displays all information pulled from the database including the plant name, a picture of the plant, and its care information.
    b. Functional Requirements: After selecting a plant from the search output, the results page must load with all information for the selected plant.
    c. Non-Functional Requirements: After the user selects a plant, pull all information about that plant from the database and display it in the results page.
4. Navigation Bar
    a. Top of the website has a navigation bar with drop down menus to navigate to different pages of the site.
    b. Functional Requirements: Every page header contains a bar at the top of the webpage. This bar contains menus/links to navigate to other pages of the website.
    c. Non-Functional Requirements: Each page starts with the same header which uses HTML divs, dropdowns, and anchors to create a top menu which is identical on every page. Users can click these links to redirect to pages within the website.
5. Plant matching survey
    a. A survey that tells the user what plant is most suited for them.
    b. Functional Requirements: The user can take a survey about their personal preferences/growing environment such as humidity, temperature, and difficulty.

    c. Non-Functional Requirements: Based on user responses, searches the database and returns the plant/plants with the most matching care requirements.

6. Plant Troubleshooting Pages
    a. Various pages to help users identify potential problems with their plants such as water, light, or nutrient deficiencies and pest infestations.
    b. Functional Requirements: Contains a menu page with links to a variety of categorized growing problems for ease of navigation.
    c. Non-Functional Requirements: Design a main page with links to many potential issues, each link leads to a separate page which contains pictures and information about the problem and how to resolve it.

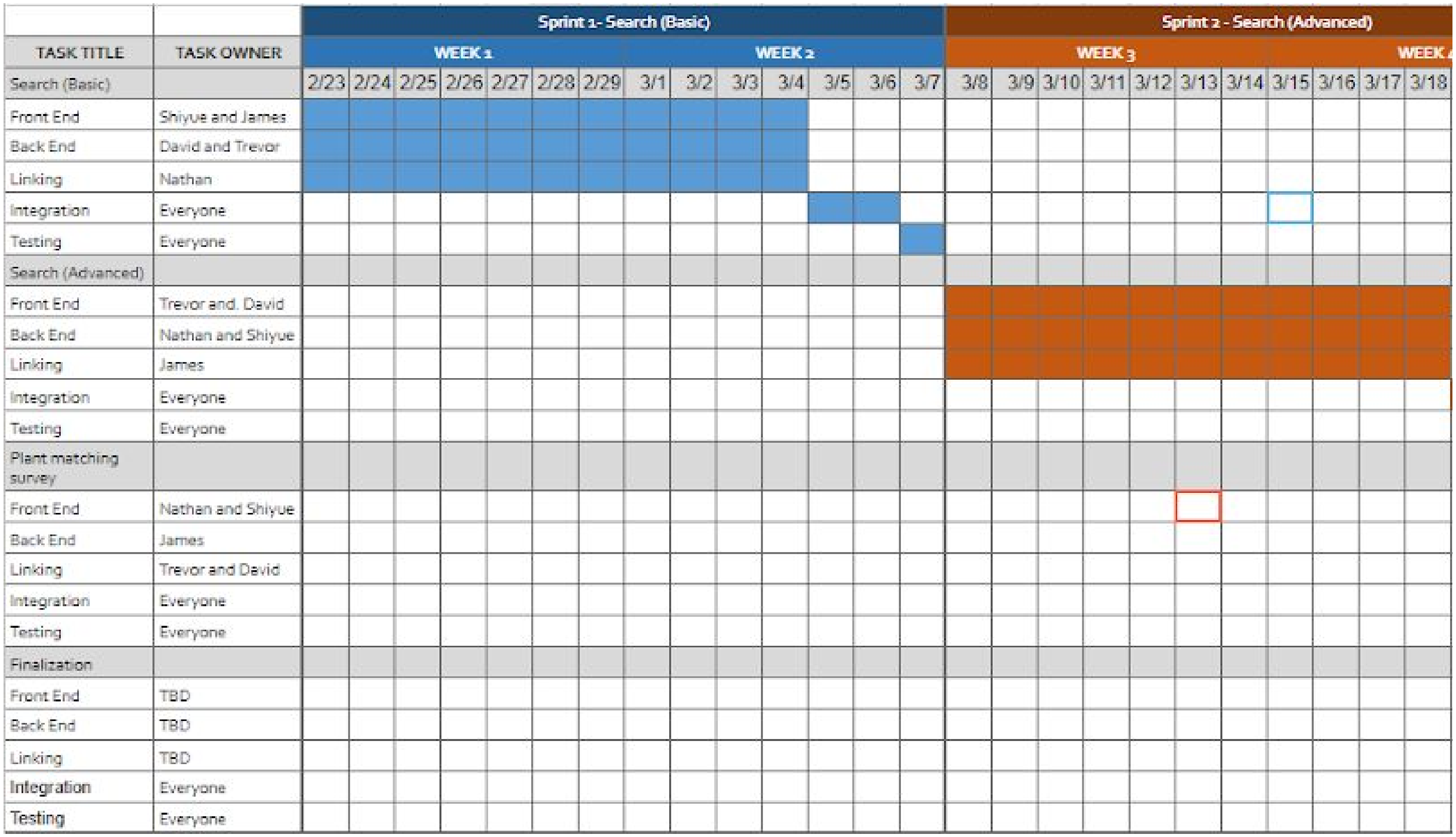
Project Plan:

- Management Tool:
- Management method:
- Sequence of Sprints
    - Sprint 1 - Search (Basic)
        - What will be developed in this sprint
            - Search Page/Navigation bar (front end)
            - Organized Database (back end)
            - Ability to search database with user input (linking)
        - Front End: Create an HTML page with Bootstrap formatting which contains a search bar and a table to list search matches (search page). Create a navigation bar header which will be applied and expanded on in all future sprints.
            - James and Shiyue
        - Back End: Create a framework for a database using PostgreSQL which contains plant names along with care information.
            - Trevor and David
        - Linking: When a search input is entered, scan database for matching names and return a list of full or partial matches to display in the results table.
            - Nathan
        - Time Frame: 2/23 - 3/7
    - Sprint 2 - Search (Advanced)
        - What will be developed in this sprint
            - Results page (front end)
            - Picture database (back end)
            - Ability to pull information for a single plant to the results page (linking)

        - Front End: Create an HTML page with Bootstrap formatting which contains plant name, space for an image, and a table of care information (results page). Update navigation bar as necessary.
            - David and Trevor
        - Back End: Create a separate database or add to the existing PostgreSQL database to add images to the information associated with each plant.
            - Shiyue and Nathan
        - Linking: When a plant is selected on the search page and redirected to the results page, the information associated with the selected plant is loaded into the results page.
            - James
        - Time Frame: 3/8 - 3/21
    - Sprint 3 - Plant Matching Survey
        - What will be developed in this sprint
            - Survey page (front end)
            - More entries (back end)
            - Ability to search care requirements in database (linking)
        - Front End: Create a survey page where users answer multiple choice questions about the care conditions they are looking for in a plant. Update navigation bar to include survey page.
            - Nathan and Shiyue
        - Back End: Continue to add more plants and information to the database.
            - James
        - Linking: When the user submits their survey results, scan the database for matching care information and return a list of the plants with the most matching care requirements.
            - Trevor and David
        - Time Frame: 3/22 - 4/4
    - Sprint 4 - Finalization (work done in this sprint will be dependant on prior success)
        - What will be developed in this sprint
            - Any previously mentioned features which were not finished/finalized
            - Plant Troubleshooting Pages (time permitting)
        - Front End: Finalize all prior features, then develop troubleshooting pages if there is enough time. Update navigation bar header as pages are added.
            - To be determined (based on amount of work left)
        - Back End: Continue to add more plants and information to the database.
            - To be determined (based on amount of work left)

- Linking: Finalize previous features.
    - To be determined (based on amount of work left)
- Time Frame: 4/5 - 4/18

			Sprint 3 - Plant Matching Survey														SPRINT 4 - FINALIZATION													
4			WEEK 5							WEEK 6							WEEK 7							WEEK 8						
3/19	3/20	3/21	3/22	3/23	3/24	3/25	3/26	3/27	3/28	3/29	3/30	3/31	4/1	4/2	4/3	4/4	4/5	4/6	4/7	4/8	4/9	4/10	4/11	4/12	4/13	4/14	4/15	4/16	4/17	4/18
🟧	🟧																													
		🟧																												
			▩	▩	▩	▩	▩	▩	▩	▩	▩	▩	▩																	
			▩	▩	▩	▩	▩	▩	▩	▩	▩	▩	▩																	
			▩	▩	▩	▩	▩	▩	▩	▩	▩	▩	▩																	
														▩	▩															
																▩														
																	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨			
																	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨			
																	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨	🟨			
																												🟨	🟨	
																														🟨



		Sprint 1- Search (Basic)														Sprint 2 - Search (Advanced)										
TASK TITLE	TASK OWNER	WEEK 1							WEEK 2							WEEK 3							WEEK			
Search (Basic)		2/23	2/24	2/25	2/26	2/27	2/28	2/29	3/1	3/2	3/3	3/4	3/5	3/6	3/7	3/8	3/9	3/10	3/11	3/12	3/13	3/14	3/15	3/16	3/17	3/18
Front End	Shiyue and James	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦														
Back End	David and Trevor	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦														
Linking	Nathan	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦														
Integration	Everyone												🟦	🟦												
Testing	Everyone														🟦											
Search (Advanced)																										
Front End	Trevor and. David															🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Back End	Nathan and Shiyue															🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Linking	James															🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧
Integration	Everyone																									
Testing	Everyone																									
Plant matching survey																										
Front End	Nathan and Shiyue																									
Back End	James																									
Linking	Trevor and David																									
Integration	Everyone																									
Testing	Everyone																									
Finalization																										
Front End	TBD																									
Back End	TBD																									
Linking	TBD																									
Integration	Everyone																									
Testing	Everyone																									