



TREE
HUGGER
CO.

Final Report

COSC4P02
SOFTWARE ENGINEERING II
WINTER 2024
NASER EZZATI-JIVAN
April 28th, 2024
Brock University

Alexandre Reuillon
ar16sv@brocku.ca
6197834

Balkaran Sidhu
bs19xt@brocku.ca
6837322

Lance Brown
lb17sy@brocku.ca
6445944

Manroop Singh Rakhra
mr19mq@brocku.ca
6857551

Marylina Aka Beyeh
lb20sm@brocku.ca
7056260

Nicole Vojvodić Chang
du20lm@brocku.ca
7201841

Parneet Gill
pg19uy@brocku.ca
6854640

Vishistha Sharma
vs19rq@brocku.ca
6855944

Table of Contents

1. Overview	03
a. Goals	03
b. Features	04
<hr/>	
2. User Manual	05
a. Accessing the web app	05
b. Creating an account and signing up	05
c. Home Page	06
d. Choose a file or open camera	06
e. Search plants	07
f. Plant List	08
g. User Account	09
h. Garden	11
i. Environments	13
j. Settings and Logging out	15
<hr/>	
3. Sprints and meetings	16
a. Meetings	16
b. Sprints	17
Sprint 1	18
Sprint 2	19
Sprint 3	20
<hr/>	
4. Testing	20
a. Unit Testing	20

b. Accuracy Testing for API	22
<hr/>	
5. Security Features	22
<hr/>	
6. Github and our Contributions	23
7. Issues	29
8. Links	30

1. Overview

In this document, we will offer a walkthrough on how to operate our plant identification web application as well as a comprehensive report on our development process throughout this term. The user manual will guide users on navigating the features within the web application. For our project, we implemented an Agile Scrum approach that is both incremental and iterative. The Agile Scrum methodology encompasses iterative development within brief cycles called sprints. This enables adaptability to evolving requirements; thus, it can be beneficial for our project where user feedback and new plant species may influence features and priorities.

a. Goals

Our project aims to create a dynamic application that revolutionizes plant identification while fostering a deeper connection with nature. By seamlessly blending technology with botanical knowledge, we envision an immersive experience that inspires users to explore the rich diversity of plant life around them. Through this application, users will not only gain proficiency in identifying various plant species but will also be encouraged to actively engage with their natural surroundings, promoting outdoor exploration and

environmental conservation. What makes our project distinguishable from other plant identification applications is user experience enhancement. Our application implements a reward system and an interactive 3d model of garden. Integrating a reward system adds a gamified element that can increase user engagement and motivation. The interactive 3D model of a garden further enhances the user experience by providing a visually engaging and immersive way for users to explore and interact with plants. This combination of features not only helps users identify plants but also makes the process enjoyable and interactive.

b. Features

- Sign-in and sign-up page with ‘forgot password’ functionality.
- Splash screen with transition and our logo.
- Account recovery feature that allows users to recover their account by using their email to reset their password.
- Homepage with all necessary functionalities, links, and important pages.
- Upload image function so that the user can upload their image from their computer.
- A comprehensive list of plants with an image, common name, and scientific name for each plant.
- Open camera function that allows the user to capture photos of their plant.
- Search functionality that allows users to search through a database of plants based on the user’s keywords.
- A user account page that contains the user’s information and bio. This page allows the user to see their achievements and recently identified plants. It also has wallpapers to use as a background or homescreen for the user’s smartphone or computer.
- A garden page that shows the user's collection of identified plant species with a default background.
- Environments page that shows an interactive 3D model of a garden.

- A settings page that allows the users to customize and manage their account such as age, name, contact, and bio.
- Logout feature to log out of the user's account.

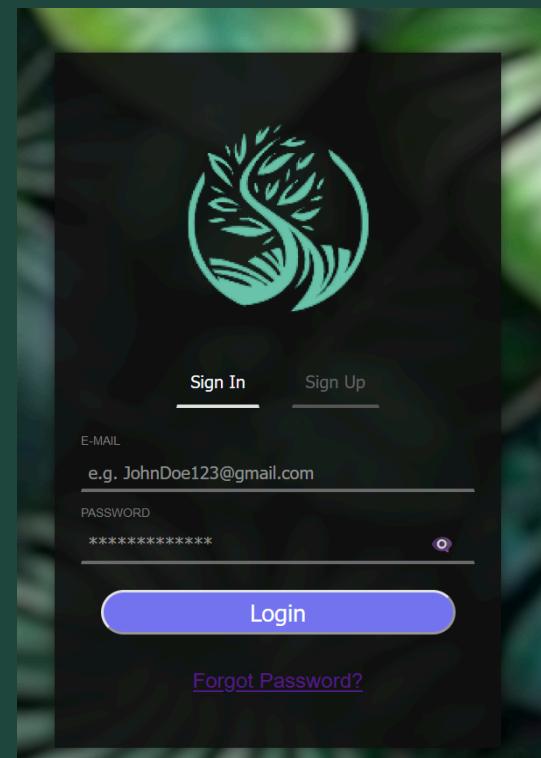
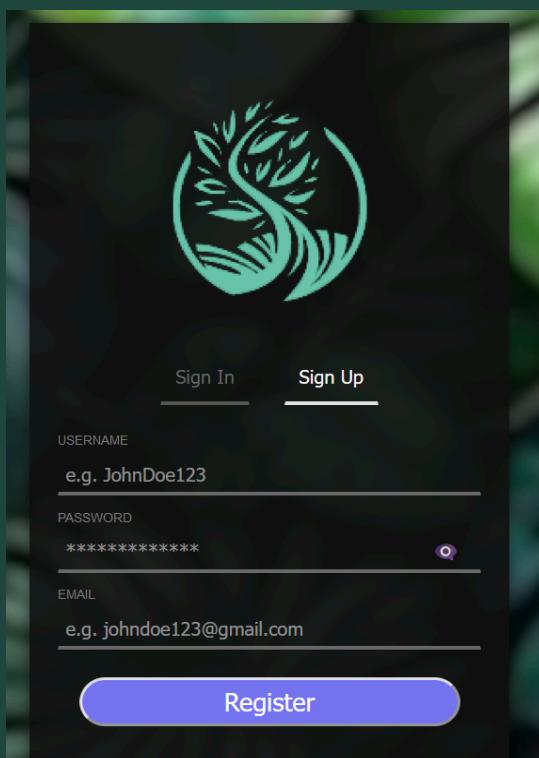
2. User Manual

a. Accessing the web app

This web application is available at <https://plantsource.org/>. This link will redirect you to the sign in and sign up page.

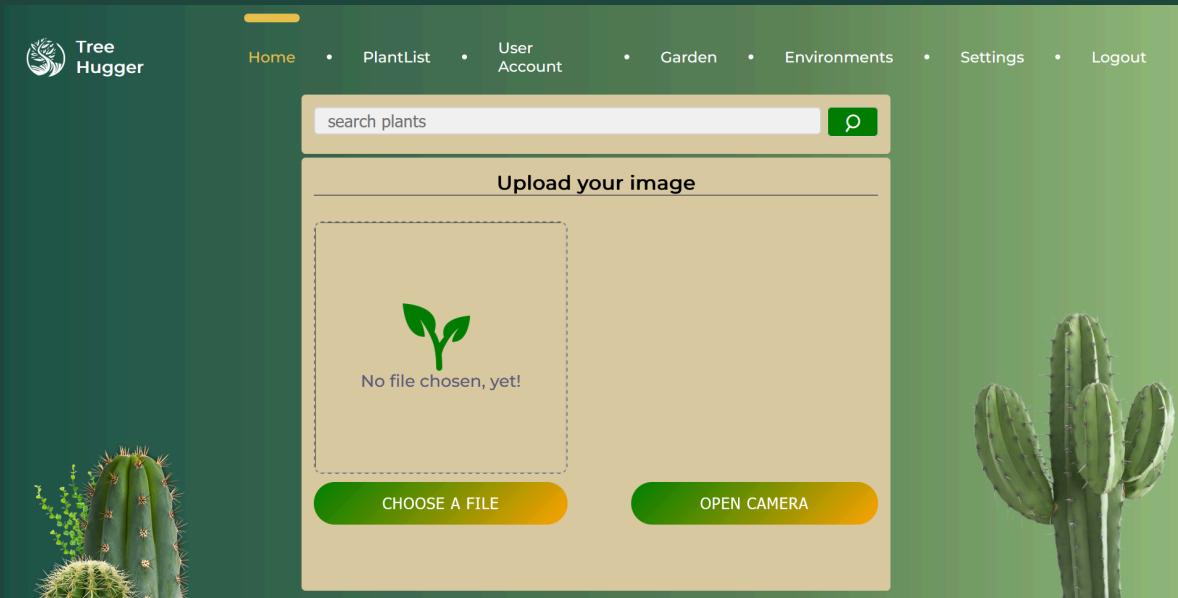
b. Creating an account and signing up

Before accessing the home page, an account must be created. Create an account by selecting the 'sign up' tab. Enter your username, password, and email, then click register. A message will appear stating that you have successfully created an account. Select the 'sign in' tab and enter your email and password, then click 'Login'.



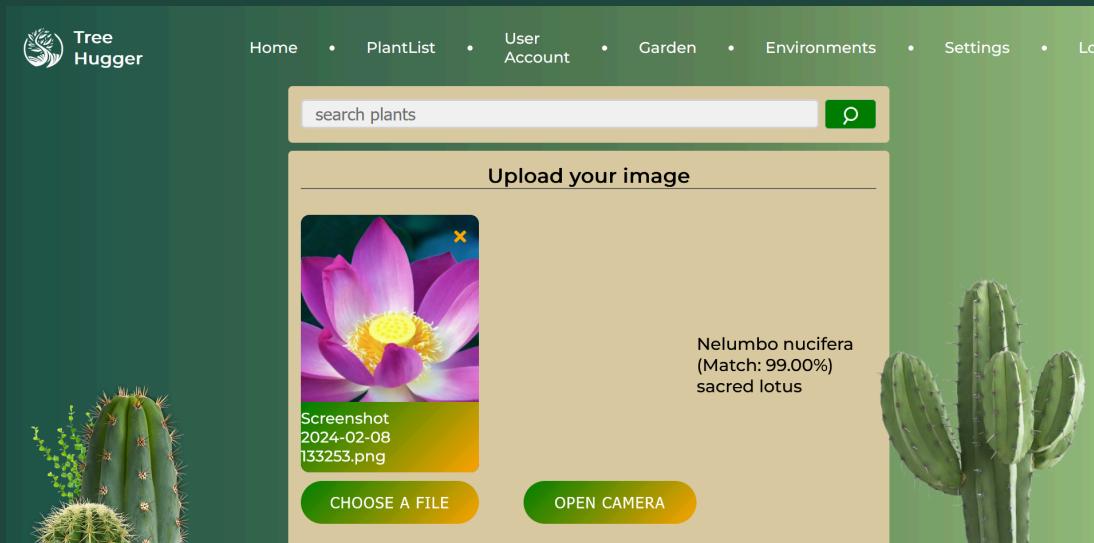
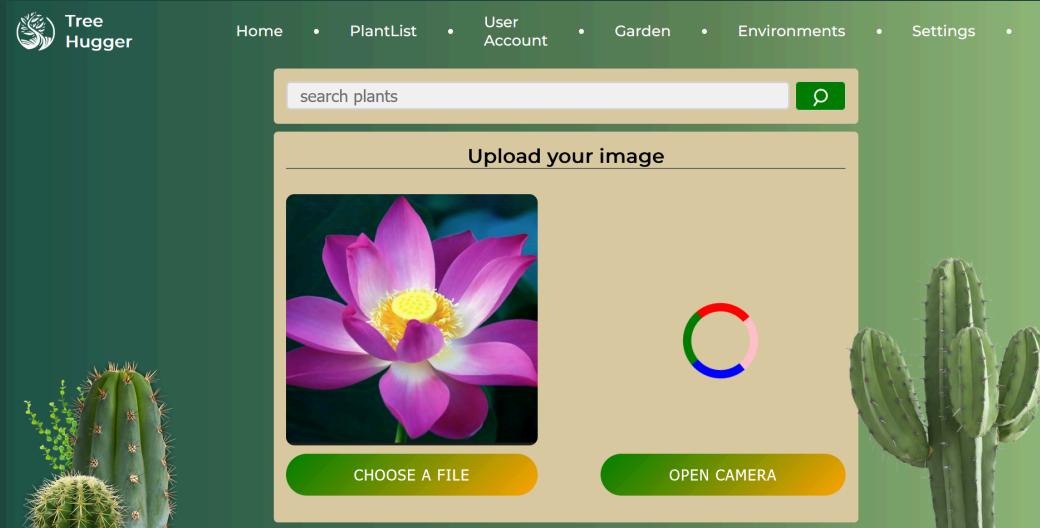
c. Home page

The landing page was designed to be as seamless as possible for the users. We opted for a sleek, vibrant design, balancing simplicity with vividness. At a glance, users grasp the webpage's essence seamlessly. Every button is appropriately labeled, enhancing ease of navigation.



d. Choose a file or open camera

The user has the option to either upload their images via their computer's file explorer or allow the application to access their camera to capture snapshots of the plants they wish to identify. When utilizing the camera feature, make sure that the captured photos have as much clarity as possible to ensure accurate identification of the desired plants. When the user uploads their photo, a loading circle appears and disappears once it is finished identifying. Also, when the user hovers over the photo preview box, the file name appears at the bottom of the photo preview box screen along with an 'x' on the top right corner. The 'x' can be clicked on so that the photo disappears and the user can choose another photo to upload.



e. Search plants

The search bar component efficiently filters a list of plant names according to the user's input keywords, streamlining the process of finding specific plants. The search bar component is versatile because it allows users to search for partial words or even full sentences; thus, enhancing the flexibility and ease of finding desired plant names. The search functionality retrieves the desired plant information using an API, called Perenual, and promptly presents it to the user.



Home PlantList My Account Garden Settings

[SEARCH PLANT](#)

SHOWING RESULTS FOR: "fir"

FOUND A TOTAL OF 30 RESULTS

[PREVIOUS PAGE](#) [NEXT PAGE](#)

IMAGE	COMMON NAME	SCIENTIFIC NAME
	European Silver Fir	<i>Abies alba</i>

f. Plant list

The plant list page hosts an extensive array of plant species. Each entry includes details such as common names, scientific classifications, and accompanying images, functioning as a user-friendly encyclopedia. The Perenual API facilitates the retrieval and presentation of this plant information. The user is able to navigate to the next or previous page to further explore different species of plants.

LIST OF PLANT SPECIES:

[PREVIOUS PAGE](#) [NEXT PAGE](#)

IMAGE	COMMON NAME	SCIENTIFIC NAME
	European Silver Fir	<i>Abies alba</i>
	Pyramidalis Silver Fir	<i>Abies alba 'Pyramidalis'</i>

g. User Account

The user account page is a personalized hub housing the user's profile information and biography. Here, users can view their accomplishments and recently identified plants. Additionally, the page offers a selection of wallpapers suitable for use as backgrounds or home screens on smartphones or computers. Under the "About" section, the user's information is displayed, including their name , contact email, location, and age. There's also an "About Me" section which provides additional details about the user (bio). The "Achievement" section encourages users to upload a certain number of plants to unlock special backgrounds or medals. In the "Recently Identified" section, there's a list of plants that have been recently identified by the user, along with the date of identification. Lastly, the "Garden" section includes different types of biomes such as "Garden," "Meadow," and "Forest." These offer a selection of wallpapers suitable for use as backgrounds or home screens on smartphones or computers.





stonecoldsteveaustin

- [Home](#)
- [About](#)
- [Achievement](#)
- [Plant List](#)
- [Garden](#)
- [Settings](#)

About



User Info:

> Name:	Steve Austin	> Age:	54
> Contact:	stone@gmail.com		
> City:	America		

About Me

Stone Cold Steve Austin.



Achievement

Upload 10 plants to unlock a special achievement background or medal



[Unlock Achievement](#)



Recently Identified

Plant Name

sacred lotus
sacred lotus
sacred lotus
sacred lotus
sacred lotus
sacred lotus
Pinecone Cap
Pinecone Cap
Pinecone Cap
Pinecone Cap

Date Identified

04/28/2024
04/28/2024
04/28/2024
04/28/2024
04/28/2024
04/28/2024
04/28/2024
04/28/2024
04/28/2024
04/28/2024

Garden

[ALL](#) [GARDEN](#) [MEADOW](#) [FOREST](#)



h. Garden

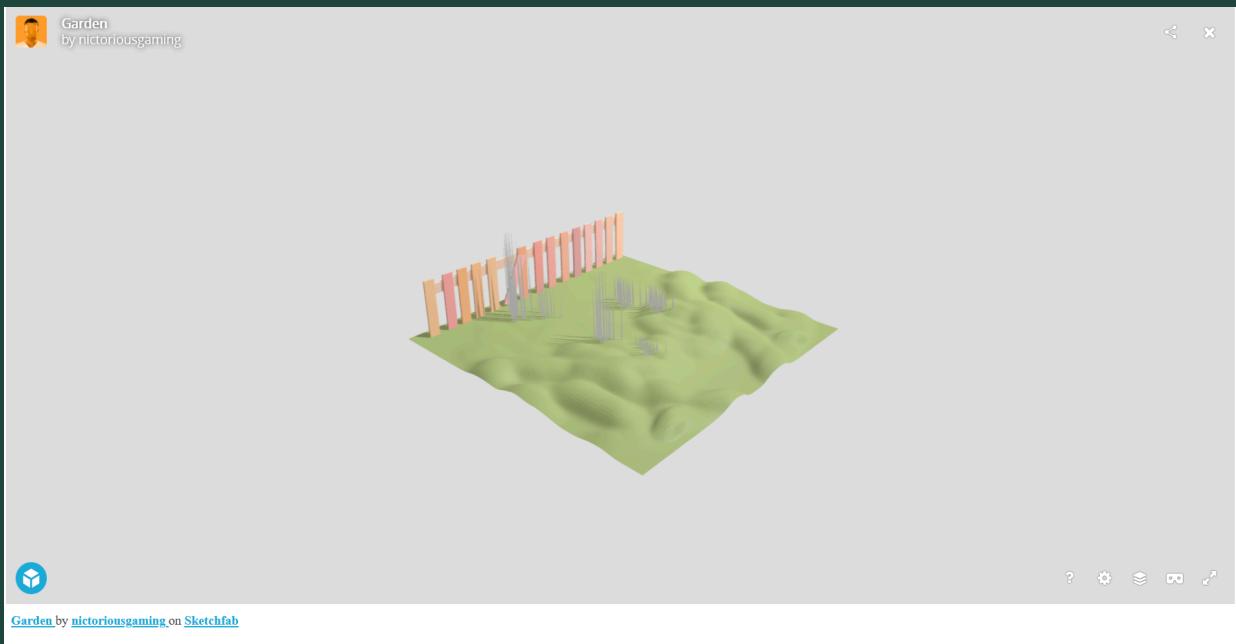
The garden page offers a 2D representation of the user's catalog of identified plant species, set against a default background. Users can explore their collection, which may include multiple entries for each plant species, drawn from a fixed list of plants. This page functions akin to a trophy room, where each new species identified contributes to

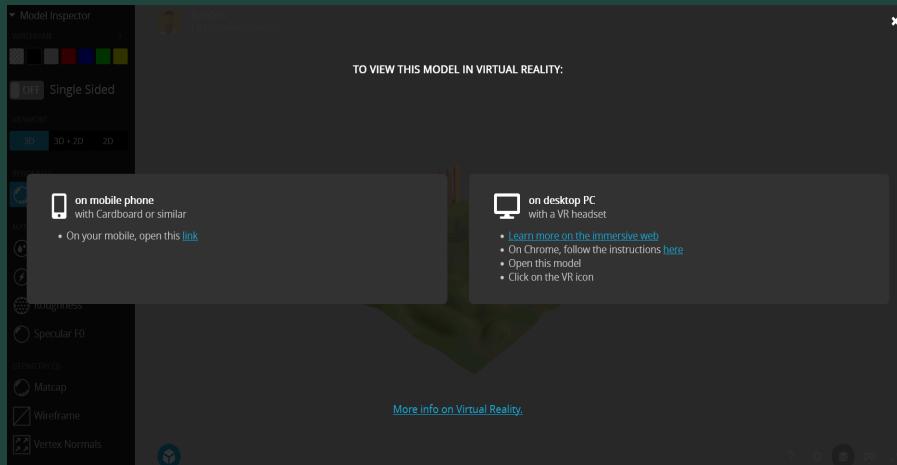
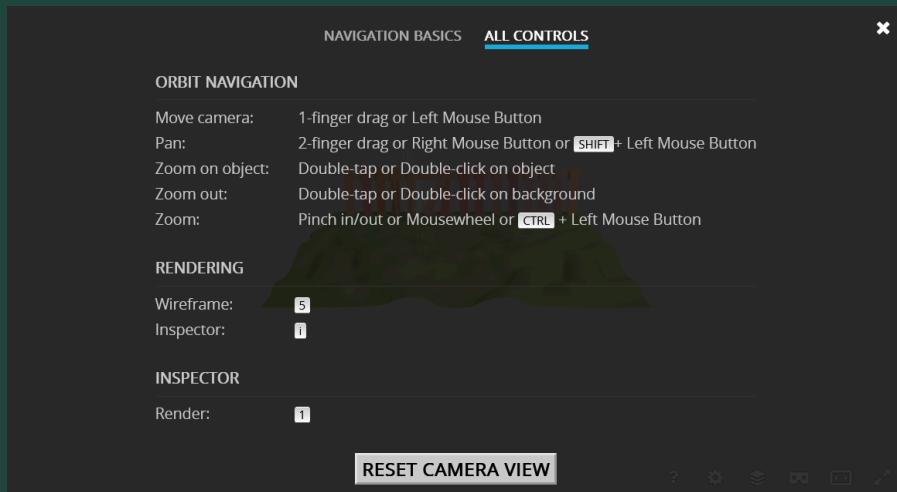
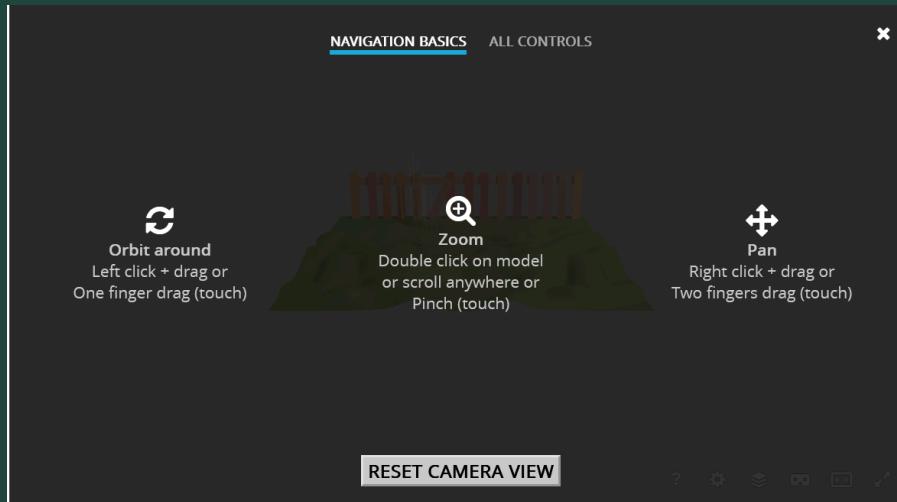
the evolving garden landscape, with plants “growing” and appearing as they are added to the collection. The user can select the collected plant and a pop up window appears, displaying the common name, the date identified, a description, and a photo of the plant.

The screenshot displays a mobile application interface. At the top, a navigation bar includes links for Home, PlantList, My Account, Garden (which is the active tab), and Settings. Below the navigation bar is a scenic background image of a green hill under a dramatic, cloudy sky. In the foreground, there is a grid of small, dark cards with white icons representing different plants. Six of these cards are labeled "sacred lotus", and one is labeled "Pinecone Cap". A larger, semi-transparent pop-up window is centered over the grid. The pop-up has a teal header bar with the text "sacred lotus" and a close button ("x"). The main body of the pop-up contains the text "This is a plant" and "Discovery Date: 04/28/2024". Below this text is a large, detailed grayscale photograph of a sacred lotus flower, showing its characteristic whorl structure. The entire application interface is set against a dark, textured background.

i. Environments

A 3D garden model was created for the Environments page utilizing the Blender 3D modeling tool. Through smooth integration with Sketchfab, we successfully embedded the immersive 3D model into our web application to enhance the user experience and allow for dynamic exploration of the garden environment. Users are able to orbit, zoom, pan, render, and view the model in VR mode. These functionalities are located at the bottom right corner of the interface. To return to the home page, users can simply click the "Go back one" arrow button or press the "Alt + Left Arrow" key combination on their keyboard. Below are screenshots detailing the step-by-step process for navigating and interacting with the 3D model.





j. Settings and logging out

The settings page enables users to customize and manage their account details, including their age, name, location, contact information, and biography. The updated information will display on the ‘My Account’ page. The user is given the option to set their account public or private, receive email updates, and allow a two factor authentication. Selecting the logout function will sign the user out of their account and automatically redirect them to the login page.

Account Details

Name: Steve

Contact: stone@gmail.com

Age: 54

Country: America

Account Bio:
Stone Cold Steve Austin.

UPDATE

Account Settings

Language: English

Account Visibility: Public

Receive Email Updates:

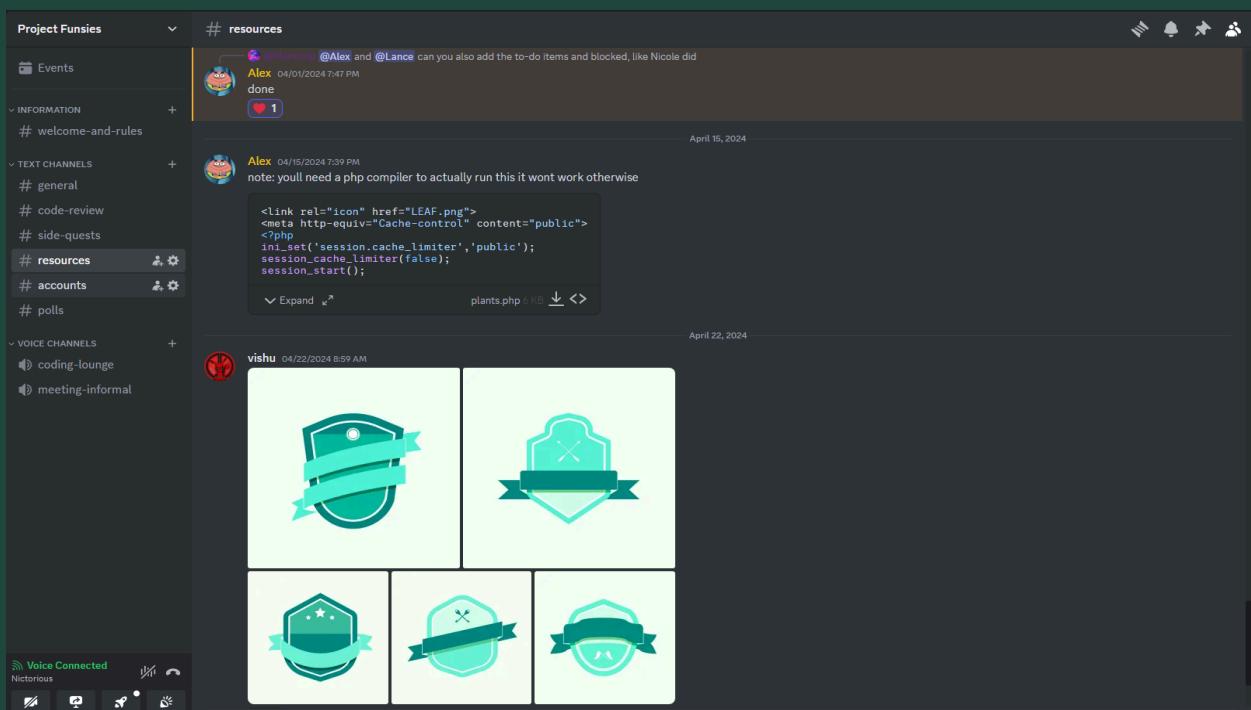
Two Factor Authentication:

Apply

3. Sprints and meetings

a. Meetings

The team meetings were being held on Discord voice channels. They catered to fit around everyone's schedules. The selection of Discord is predicated upon its user-friendly interface and array of practical features, which significantly contribute to the effective organization of information. Moreover, it serves as the primary platform for conducting all sprint meetings and pertinent update sessions. Leveraging its screen-sharing capabilities, we are able to seamlessly monitor the progress of all team members' work.



This was made streamlined by the use of a scheduling software (<https://www.when2meet.com/>) where everyone is able to mark their respective availability for the chosen days.

Scrum Meeting-2

To invite people to this event, you can [email them](#), send them a [Facebook message](#), or just direct them to <https://www.when2meet.com/?23632067-iDrUL>.

Your Time Zone: America/Toronto

Sign In

Your Name:

Password (optional):

Name/Password are only for this event.
New to this event? Make up a password.
Returning? Use the same name/password.

Group's Availability

0/6 Available 5/6 Available
Mouseover the Calendar to See Who Is Available

Feb 12 Mon

10:00 AM
11:00 AM
12:00 PM
1:00 PM
2:00 PM
3:00 PM
4:00 PM
5:00 PM
6:00 PM
7:00 PM
8:00 PM
9:00 PM
10:00 PM
11:00 PM

All the meetings were recorded and stored in a google drive for members to refer back to, or in the rare case someone could not attend the meeting due to legal excuses.

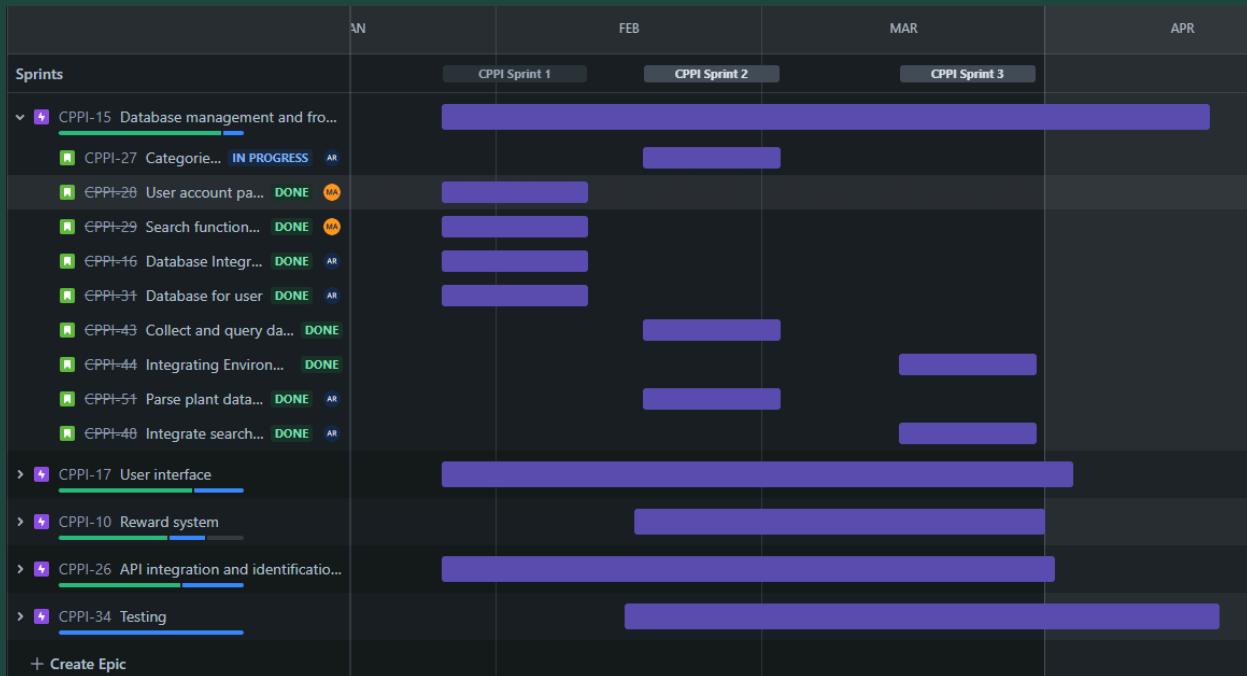
Name	Owner	Last modified b...	File size	⋮
Meeting-1 (Jan-22).mkv	me	2:10 PM	101.8 MB	
Meeting-2 (Feb-12).mkv	me	2:10 PM	174.5 MB	
Meeting-2 Part-2(Feb-13).mkv	me	2:10 PM	104.7 MB	
Meeting-3(Mar-1).mkv	me	2:10 PM	217.8 MB	
Meeting-4(Apr-15).mkv	me	2:12 PM	90.4 MB	

b. Sprints

We conducted three official sprints, each spanning a duration of two weeks, with the overarching objective of ensuring comprehensive awareness among team members regarding the progress achieved by various sub-teams. These sprints facilitated an environment conducive to brainstorming and ideation, fostering the exploration of novel concepts and solutions, irrespective of their integration into the final working prototype. Emphasis was placed on fostering an inclusive culture wherein all team members were

encouraged to express their insights and apprehensions, both within and beyond formal meeting settings.

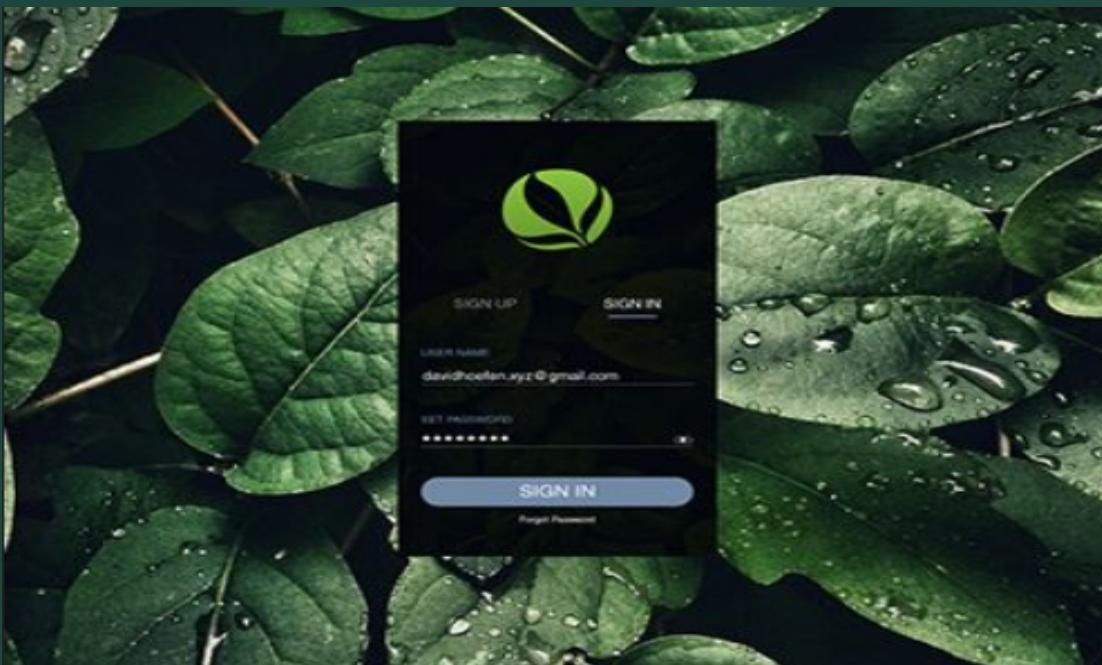
In addition to the three official sprints, we implemented unofficial "work at your own pace" sprints to accommodate periods of academic commitments, such as reading weeks, midterms, and final exams. These supplementary sprints were strategically scheduled to optimize productivity while acknowledging the exigencies of academic obligations.



Sprint 1 (26 Jan- 10 Feb)

We initiated the first sprint with seamless progress. For the first sprint, our team managed to produce a working 'skeleton' code, mock-ups for certain pages(for e.g. the login page mockup attached below) with important features, and a functioning user interface. Once we've crafted the essential functionalities, we proceed to fine-tune them and incorporate any essential additions into our backlogs for upcoming sprints. Our process involved prioritizing user stories and tasks. The highest priority was given to the

most crucial features and making sure the base of the web app is strong for all the other features to be built upon it.



Sprint 2 (17 Feb- 2 Mar)

In the following sprint, we have transferred all outstanding backlogs from the previous iteration to ensure continuity while adding more tasks along the way. Our principal objective remains the timely completion of all tasks. This juncture serves as a pivotal moment for evaluating our progress and determining whether any significant adjustments to the project's trajectory are warranted.

The screenshot shows the Jira Software interface for the 'COSC4P02 project: plant identifier' backlog. The left sidebar includes sections for Planning (Timeline, Backlog, Board, Goals, Issues, Add view), Development (Code, Project pages, Add shortcut, Project settings), and a note about being in a team-managed project. The main area displays a backlog of user stories (CPPI-32 to CPPI-37) categorized by type (User Interface, Database Management, API Integration and ID, Reward System) and status (Done, In Progress, To Do). Each item has a small icon, a title, and a list of labels (e.g., M, PG, AR, L, V, BS).

Type	Status	Labels
User Interface	DONE	L
User Interface	DONE	
User Interface	IN PROGRESS	PG
User Interface	IN PROGRESS	PG
User Interface	IN PROGRESS	L
User Interface	IN PROGRESS	L
DATABASE MANAGEMENT	IN PROGRESS	AR
DATABASE MANAGEMENT	DONE	MA
DATABASE MANAGEMENT	IN PROGRESS	MA
DATABASE MANAGEMENT	DONE	AR
API INTEGRATION AND ID	IN PROGRESS	BS
API INTEGRATION AND ID	IN PROGRESS	BS
API INTEGRATION AND ID	IN PROGRESS	V
Reward System	TO DO	V
Reward System	IN PROGRESS	PG

Sprint 3 (16 Mar- 30 Mar)

For the conclusive phase of our project sprint, our primary objective was to ensure the seamless integration of all essential website components. Given the imminent final exam period, we made a deliberate decision to prioritize judicious resource allocation. Consequently, certain features, such as the diverse environment models and the achievements section, were temporarily deferred to future iterations. This decision was informed by the intricate nature of the integration process and the necessity to maintain focus on immediate project goals.

4. Testing

a. Unit Testing- Unit tests are essential to validate the functionality of the JavaScript functions in isolation. For the JavaScript in our application, we used jest to create tests for the functions like identifyPlant, checking file uploads and functions in the login page (attached below is a snippet of the code used, can be found in the repository)

```

9   function setupFileReaderMock() {
10     class MockFileReader {
11       onload = null;
12       readAsDataURL(blob) {
13         if (this.onload) {
14           this.onload({ target: { result: 'data:image/jpeg;base64,' + btoa('fake data') } });
15         }
16       }
17       addEventListener(event, handler) {
18         this.onload = handler;
19       }
20     }
21     global.FileReader = MockFileReader;
22   }
23
24 describe('Plant Identification Page', () => {
25   beforeEach(() => {
26     document.documentElement.innerHTML = html.toString();
27     setupFileReaderMock();
28     require('./script'); // Load your script after the DOM setup
29     document.getElementById("loadingIndicator").style.display = 'none';
30   });
31
32   afterEach(() => {
33     jest.resetModules();
34     fetch.resetMocks();
35   });
36 });

```

Output-

```

.js:122:21)
      at jestAdapter (C:\Users\Balkaran\Documents\myproject\node_modules\jest-junit\index.js:122:21)
      at runTestInternal (C:\Users\Balkaran\Documents\myproject\node_modules\jest-junit\index.js:122:21)
      at runTest (C:\Users\Balkaran\Documents\myproject\node_modules\jest-junit\index.js:122:21)

39 |   })
40 |   .catch(error => {
> 41 |     console.error('Error fetching image:', error);
        ^
42 |     document.getElementById("identificationMessage").innerHTML = 'An error occurred while fetching the image. Please try again later.';
43 |     document.getElementById("loadingIndicator").style.display = 'none';
44 |     document.getElementById("retryButton").style.display = 'block';

      at error (script.js:41:17)

PASS  ./script.test.js
  Plant Identification Page
    ✓ shows loading indicator during plant identification (56 ms)
    ✓ handles API error correctly (36 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        1.239 s, estimated 2 s
Ran all test suites.

```

b. Accuracy Testing for API

For this testing phase, we meticulously uploaded images of 50 distinct plants sourced from both the internet and personal camera rolls. Out of this dataset, the API successfully identified 44 plant images, resulting in an accuracy rate of 88%. This pivotal assessment ensures the seamless functionality and flawless integration of the API, without any erroneous identifications, thus affirming its reliability and efficacy in our project.

5. Security Features-

1. Implemented HTTPS Protocol for Secure Website Connection
 - a. Introduced the HTTPS protocol to establish secure communication between users and the website.
 - b. Enabled end-to-end encryption to safeguard sensitive data transmitted over the network.
 - c. Enhanced user privacy and data integrity by ensuring secure data transmission and protection against eavesdropping or tampering.
2. Utilized Salted Hashing for Password Storage:
 - a. Implemented salted hashing techniques to securely store user passwords in the database.
 - b. Salted hashing involves appending a unique salt value to each password before hashing, significantly enhancing security.
 - c. Mitigated the risk of data leaks and unauthorized access by storing hashed passwords instead of plaintext, thereby preventing exposure of sensitive user credentials.
3. Implemented Input Sanitization to Prevent SQL Injection Attacks:
 - a. Implemented robust input validation mechanisms to parse input fields correctly and prevent SQL injection attacks.
 - b. Ensured that user-supplied data is sanitized and validated before processing to mitigate the risk of malicious SQL injection attempts.

- c. By sanitizing input fields, effectively neutralized potential exploits that could compromise the integrity of the database and expose sensitive information.

6. Github and our Contributions

a. Alexandre Reuillion (6197834) - Developer

1. Developed and integrated backend functionalities with the database:
 - a. Established user account database with secure storage of passwords.
 - b. Implemented user account creation and login authentication.
 - c. Created an identification page for viewing identified plants and potential analytics.
 - d. Utilized REST API to display a plant list with images and search functionality.
2. Set up server infrastructure for website hosting:
 - a. Configured a versatile web server supporting HTML/CSS, JavaScript, PHP, and SQL.
 - b. Ensured live SQL functionality for database operations.
3. Hosted website with Cloudflare Tunnels:
 - a. Enabled HTTPS secure connections.
 - b. Established the website domain: plantsource.com.
 - c. Integrated login page and various other pages into the hosted website.

b. Balkaran Sidhu (6837322) - Developer

1. Created a plant identification page integrating the plant.id API:
 - a. Implemented an upload image button for users to upload images from their system.
 - b. Added a camera button for users to capture pictures directly.

- c. Utilized hover functionality to enhance interactivity and provided clear instructions.
 - d. Included a retry button for users to upload different images.
2. Utilized the plant.id API for accurate plant identification:
 - a. Researched and selected the plant.id API capable of identifying over 33,000 plants.
 - b. Converted URL images to form-data to meet API input requirements.
 - c. Integrated the API key for sending requests to plant.id and displaying identified plant information.
 3. Resolved plant.id API authorization key issue for plant specifications display.
 4. Developed a responsive interface using HTML and CSS for enhanced user experience.
 5. Implemented camera and preview function
 - a. Improved image upload feature to immediately display uploaded images as previews.
 - b. Enabled the users to capture and submit images for identification without pre-saving.
 - c. Implemented error handling to manage issues related to image fetching and camera access.

c. Lance Brown (6445944) - Developer

1. Created the login page
 - a. Provides the user with the ability to log in or sign up
 - b. Created a server side program that handled login and sign up requests (**Not used in final build**)
 - i. Included sanitization for user input data
2. Created the garden page

- a. Created Javascript code that allows the user to view a dynamically created list of all their scanned plants
 - b. Created a default photo for plants that do not have their photo available
 - c. When icons for the plant are clicked they will produce a popup window that displays information about the plant as well as the full sized plant image
3. Created the settings page
 - a. Provides the user with preferential options regarding privacy, account security, notifications, and language

d. Manroop Singh Rakhra (6857551) - Scrum Master

1. Team Organization and Communication:
 - a. Organized team meetings ensuring majority attendance and conducted makeup sessions for absent members.
 - b. Recorded meetings using OBS studios for review or reference.
 - c. Communicated any conceptual doubts or issues to TAs or professors on behalf of the team.
2. Progress Reporting:
 - a. Compiled progress reports, ensuring comprehensive coverage of team activities.
 - b. Worked on the final report with Nicole, containing the cumulative progress of the project, along with a user guide, individual contributions and other important information.
3. Decision Making and Task Delegation:
 - a. Distributed tasks based on team strengths to expedite progress and minimize obstacles.
 - b. Encouraged team learning and research when encountering unfamiliar topics.
4. Collaborations:

- a. Collaborated with the Product Owner to establish web-app design and architecture. Provided input on research processes and design visions.
- b. Worked alongside Balkaran to address API authorization issues.

e. Marylina Aka Beyeh (7056260) - Developer

1. User Account and Profile Management:
 - a. Display user's basic information and provide navigation to profile sections.
 - b. About Me Page: Enable users to update basic details and profile picture, with an option to save changes.
 - c. Achievement Page: Showcase user's achievements, unlocking new features via JavaScript based on progress.
 - d. Delete Account Page: Provide warning before account deletion, with backend request for profile removal upon confirmation.
 - e. Environment Showcase: Highlight various plant environments, unlocking backgrounds as achievements are earned, with accessible links.
2. Enhanced aesthetics and responsiveness of the achievements page for improved user experience.
3. Implemented backend function to unlock reward systems upon user action.

f. Nicole Vojvodić Chang(7201841) - Product Owner and Developer

1. Structured sprints and backlogs for the development team:
 - a. Tailored tasks to each member's individual strengths and weaknesses.

- b. Ensured efficient task allocation and maximized productivity within the team.
2. Defined the Home Page Layout:
 - a. Considered the placement of elements such as the search bar and image uploader.
 - b. Used HTML, JavaScript, and CSS to create a visually appealing and responsive layout.
3. Implemented frontend search functionality into the home page layout:
 - a. Used HTML and CSS.
 - b. Utilized JavaScript to handle user interactions with the search bar, such as capturing user input.
4. Created and Implemented the image uploader functionality for the home page layout:
 - a. Allowed users to upload images from their devices.
 - b. Used HTML, JavaScript, and CSS to style the image uploader component.
 - c. Implemented JavaScript and HTML code to handle image upload events, such as selecting an image file and initiating the upload process.
5. Displayed Uploaded Images and results:
 - a. Used HTML, JavaScript, and CSS to dynamically update the home page content with the uploaded image and information about the image, ensuring a seamless user experience.
6. Implemented a 3D model of a garden to the web app:
 - a. Created a 3D garden for the Environments page utilizing the Blender 3D modeling tool.
 - b. Integrated and embedded the 3D model with Sketchfab.

g. Parneet Singh Gill (6854640) - Developer

1. Developed a splash screen using Figma animation and JavaScript, activating it for a designated duration before redirecting users to the HomePage.
2. Established the HomePage with the following features:
 - a. Provided instructions for capturing suitable images to enhance result accuracy.
 - b. Implemented JavaScript functions for users to upload images from their systems and capture images using the camera function.
3. Constructed backend infrastructure using Django (**not used in the final build**):
 - a. Created a Django application facilitating frontend interaction and data storage in a SQLite3 database.
 - b. Implemented user registration functionality, restricted application access to registered users, and developed pages for user registration.
 - c. Developed a Django admin interface to monitor database updates and user interactions.
4. Integrated Nicole's UI with Balkaran's Plant.id API into the final product:
 - a. Incorporated Balkaran's API for plant identification and integrated Open Camera functionality with the Plant.id API.
 - b. Made final adjustments to UI and JavaScript functions for the HomePage, reflecting the current version on the hosted website.
5. Improved the camera functionality and UX
 - a. Upon user capturing an image, a function converts it to a URL. The URL serves as input for the API, generating results. Results from the API are projected on the homepage for user view
 - b. Implemented a feature where clicking the "Open Camera" button toggles to a "Capture" button, allowing users to take a picture. Upon capture, the result is displayed, and the button reverts to "Open Camera" for continuous input.

h. Vishistha Sharma (6855944) - Developer

1. Achievement Page:
 - a. Showcase user's achievements, unlocking new features via JavaScript based on progress
2. Worked on the database page, parsing it into a usable interface
3. Collaborated with Balkaran to create test cases and checking for breakpoints
4. Wallpaper & Badges Rewards System(**not completely included in final build**):
 - a. Users receive wallpapers upon completing achievements.
 - b. A pop-up notification displays when a new background is unlocked.
 - c. Integrated with an achievement tracking system.
 - d. Combined with a background system for a more rewarding user experience.
5. Educational Resources:
 - a. Curated plant care guides and articles.
 - b. Accessible within the application interface.

7. Issues

- a. Scheduling Conflicts- Coordinating meetings and work sessions becomes difficult when team members have conflicting class schedules, part-time jobs, or other commitments.
- b. Task Allocation: Task distribution challenges arise when team members have conflicting schedules, making it difficult to quantify contributions like debugging or research assistance. Despite resolving this issue independently, discrepancies may occur when certain contributions, such as development work, pair programming, do not integrate into the final draft.

- c. Integration Challenges: Aligning individual contributions and integrating them into the project can be problematic if team members work on separate tasks at different times.
- d. Problems with software: For our specific case, the “Forgot Password” being finicky and not reliable, could cause some misunderstanding

8. Links

- Github repository:
<https://github.com/Nicole-Vojvodich-Chang/COSC4P02-PROJECT-PLANT-IDEN-TIFIER/tree/main>
- Link to the web application: <https://plantsource.org/>
- Sketchfab:
<https://sketchfab.com/3d-models/garden-174597df78204afb94137e16d735402f>
- Jira:
<https://cosc4p02plantidproject.atlassian.net/jira/software/projects/CPPI/boards/1/backlog>