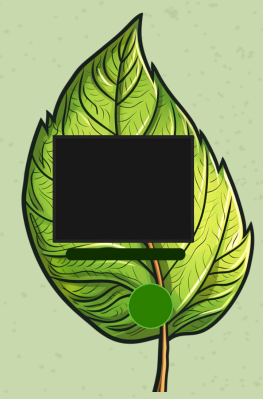
LeafDex

Milestone 1



By Chandan Marle, Nikolay Ostroukhov, Omar Natour, Samuel Guzman Hernandez

LeafDex
Project Overview

LeafDex is an app designed to allow users to take photos of plants they see for identification purposes. Features include...

- Plant Photo Identification
- Mobile and Desktop Interfaces
- Personal record of plants identified
- Map to show locations of previous plants
- Plant information
- Github Repo Link
- Github Issues Link

Team Members

Niko

Niko has been responsible for setting up the UI framework

Chandan

Chandan has been working on react functionality between components

Sami

Sami developed and contributed to the UI components

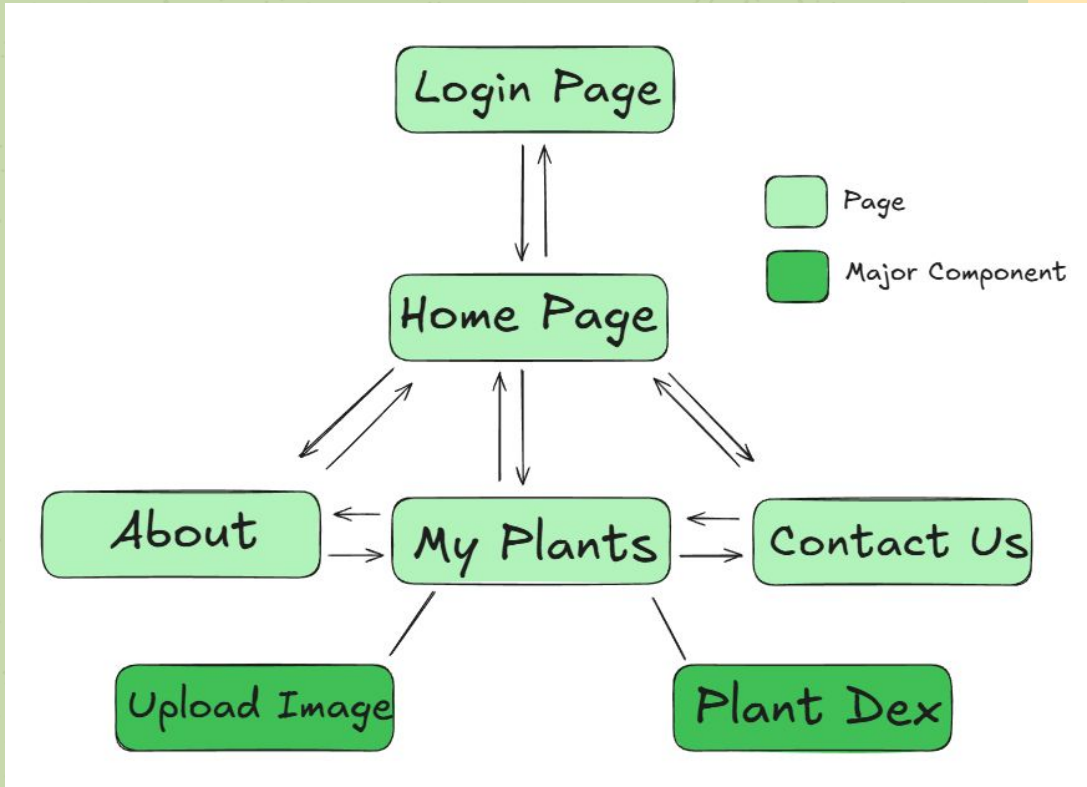
Omar

Omar has helped with sprint organization and code structure.

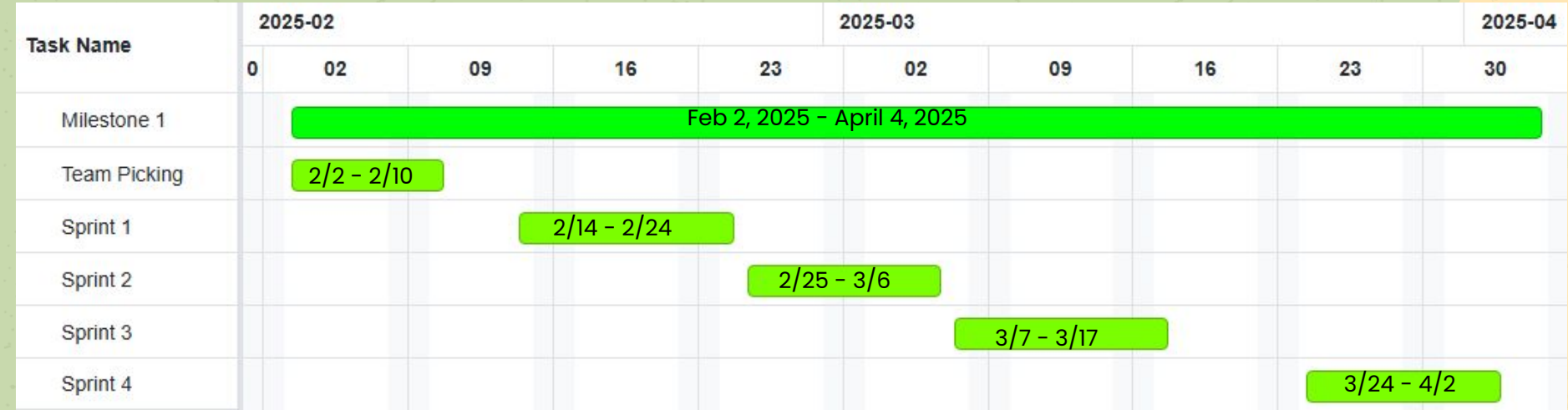
Software Architecture Overview

UI Layout Flow

Here you can see described the front end control flow described in a diagram. Light green blocks represent separate pages, while dark green blocks represent the major components in those pages



Historical Development Timeline



Design Guidelines

For LeafDex We've decided to implement Tailwind CSS and daisyUI.

For color choices we picked the daisyUI forest theme, to match represent the foliage our users will be interacting with.

We've chosen a dark theme for lower eye strain for our users.

Style Guideline Documentation

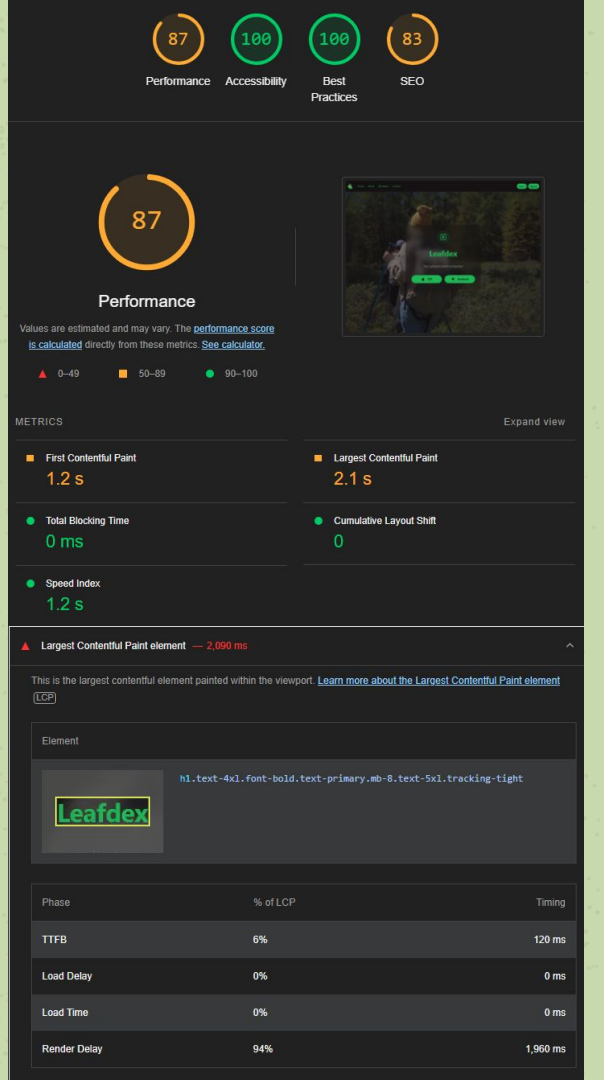
Performance Considerations

Lighthouse Analysis

87 overall score for Lighthouse score. The biggest detriment would be text compression, which we can look into for the future.

Contentful Paint

One of the biggest delays when rendering the page is our text leafdex. We wonder however if this is being wrongly attributed to the text and not the video file that plays. More investigation needed.



Niko's Contributions for Milestone 1

Assigned Work Summary

1. PR #25 Establish CSS framework (TailwindCSS + DaisyUI)
2. PR #31 Implement Desktop UI using CSS frameworkPR
3. PR #32 About, Contact Pages, and an Update to IdentifiedPlants Page

Niko's Contributions for Milestone 1

Assigned Work Summary

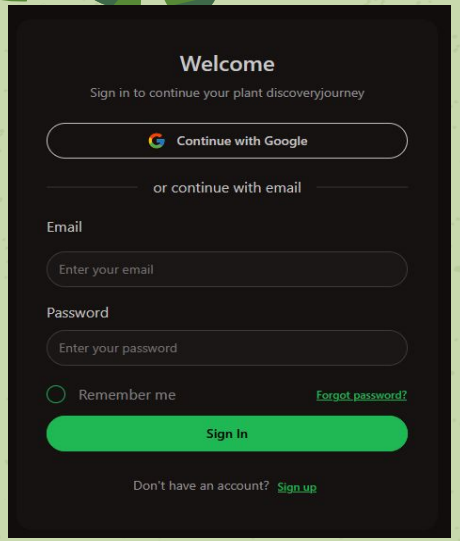
Team Role: UI/UX Design
My main contribution was to the design language and framework of the UI. With the team's approval, we decided to use TailwindCSS. I made the decision to use TailwindCSS plugin daisyUI to easily theme our components for consistency.

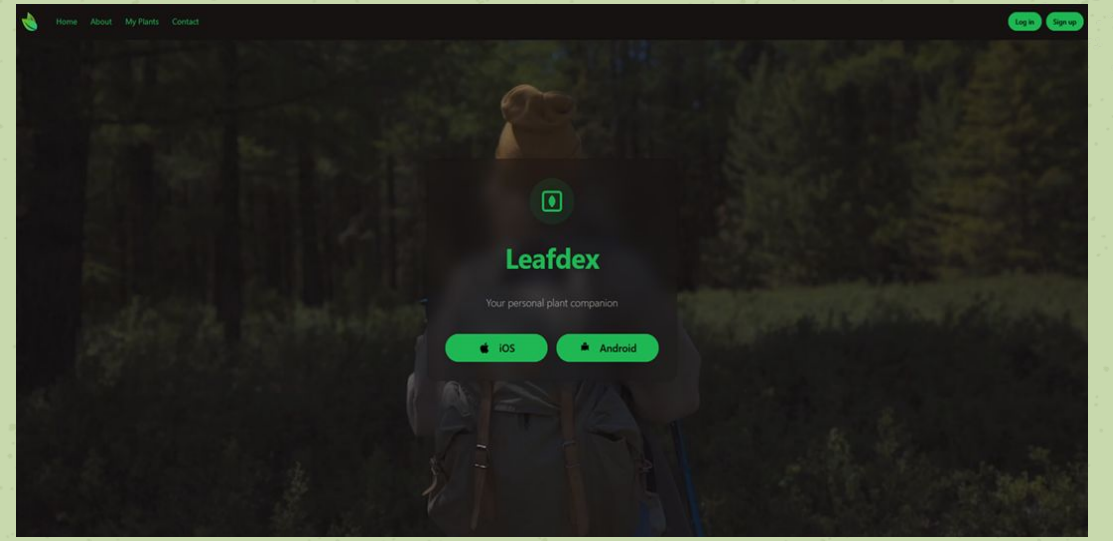
Here are my biggest contributions to the milestone:

1. PR #25 Establish CSS framework (TailwindCSS + DaisyUI)
2. PR #31 Implement Desktop UI using CSS framework
3. PR #32 About, Contact Pages, and an Update to IdentifiedPlants Page

Niko's Contributions for Milestone 1

Code & UI Explanation





I implemented the landing, about, contact, and my plants page. I also redesigned the login and sign up pages to maintain a consistent look and feel. I made the decision to use the daisyUI Tailwind plugin as it came with convenient theme support.

Niko's Contributions for Milestone 1

Challenges and Insights

CSS frameworks make CSS really convenient to use once you wrap your head around them. I didn't face too many obstacles but it did take some time to get a feel for how the UI should be structured.

Niko's Contributions for Milestone 1

Future Improvements & Next Steps

There is always room for improvement. I'd like to go through and clean up some of the UI code as well as polish up some elements, like the lack of padding between the page title and the nav bar. I'd also like to go through and make sure css rules are consistency applied throughout.

Our next step is to implement the mobile UI, and once the desktop and mobile UI's are in a good state, we'll continue to the backend side of things.

Chandan Marle - Work Accomplished

Issue 22	Create upload button that puts images into React Context
Issue 23	Create a list component that displays saved images
Issue 34	Create Card for items in info list
Issue 37	Make context images persistent through local storage
Commits/PR	08c89270e80fa6fcfbf873ea06089cc962b07156
	0872a269ff0bd1a1e4e50f39c2623023e201412d

Chandan - Code Impact

My best work is from two files, upload.tsx and uploadContext.tsx

uploadContext.tsx

This file defines the system that holds context of user info uploaded. It is responsible for managing the state of the uploaded images.

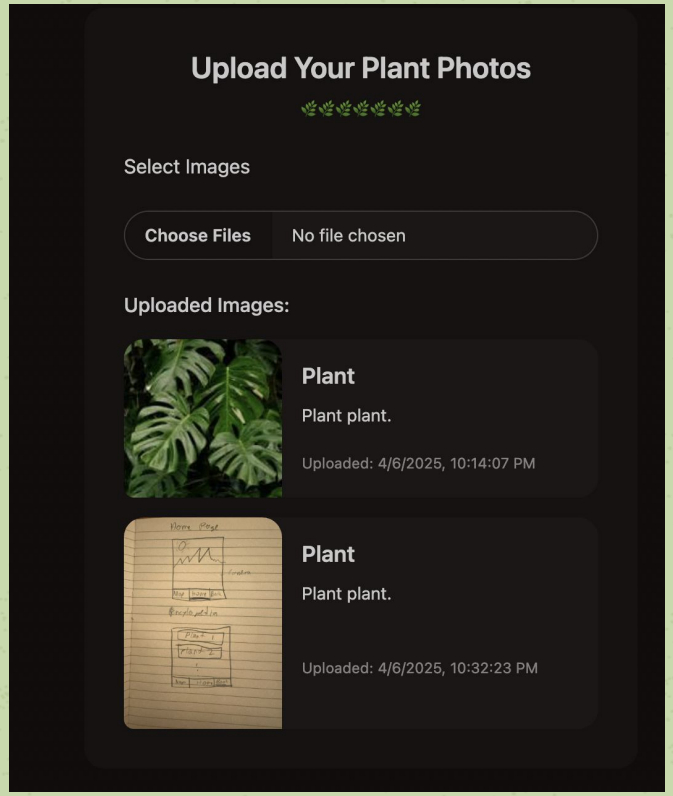
This file also contains local storage usage for persistent storage. The ContextWrapper can be used to convey the image state in any jsx.

upload.tsx

Upload holds the functionality behind the react UI component that allows the user to input images and the UI that outputs the info list.

This component also uses the DaisyUI framework laid out in the rest of the project for better user experience.

Chandan - UI


Upload Your Plant Photos
Select Images
Choose Files
No file chosen
Uploaded Images:
Plant
Plant plant.
Uploaded: 4/6/2025, 10:14:07 PM
Plant
Plant plant.
Uploaded: 4/6/2025, 10:32:23 PM

Chandan - Lessons Learned

Communicate issues earlier, I spent a long time on figuring out how to do react context. It did work in the end but I did spend too much time on it when I should have asked for help.

Git issues, I was running into authentication problems with my sftp and ssh connections to the repo, this ended up taking up a large portion of my time.

Time management, I miscalculated how much running into blockers and difficulties would cause in delays, in the future, time delays should be accounted for in the process.

Chandan - Future Steps

Work with the team on better time management and resource allocation.

Flesh out bugs and errors in pre existing code.

Begin working on server side microservices to make the app functional.

Samuel's Contributions for Milestone 1

Code & UI Explanation

1. Commit ID#b4cb3b5 First version of the log in page UI done (CSS)
2. Commit ID# f92885 Added video to the log in page
3. Commit ID# 9092e7f Navigation bar done to IdentifiedPlants Page

Samuel's Issues

Main Page overall Layout- Web page

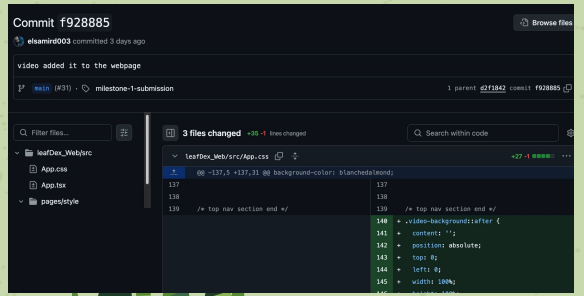
Signup and login button

Front end create account section

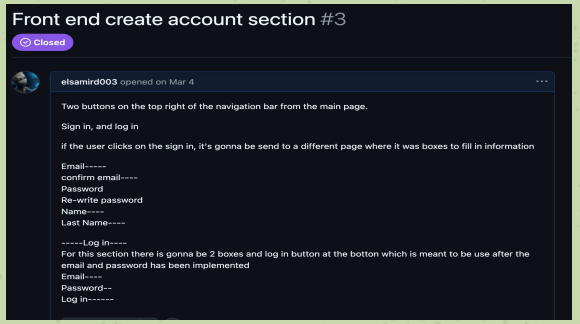
EC2 account for hosting our services

Assigned Work Summary (Samuel)

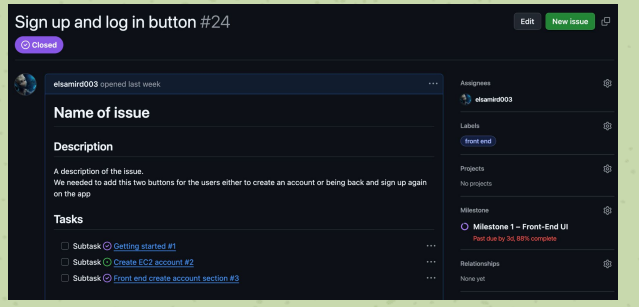
Video added



Front end create account section



Sign up and log in button

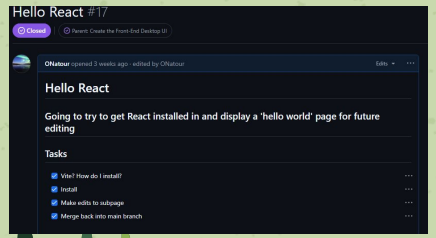


Designing the UI for the web page

Worked on the designing the overall website.

Assigned Work Summary (Omar)

Installing React



Info Ledger

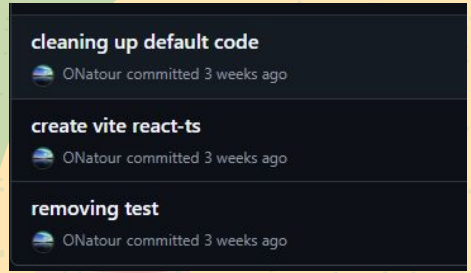
Documentation

Team Slides

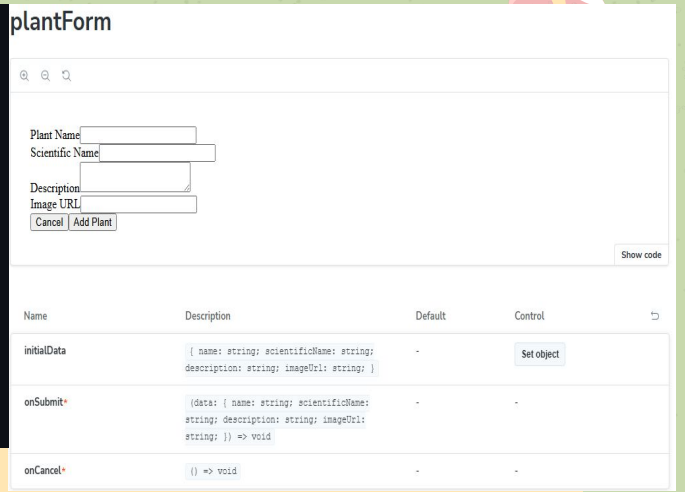
Made significant Headway in writing team slides including Diagrams

Demonstrations (Omar)

Logos




```
import type { Meta, StoryObj } from '@storybook/react';
import { fn } from '@storybook/test';

import PlantCard from '../components/ui/PlantCard.tsx';

const meta: Meta<typeof PlantCard> = {
    title: 'Example / plantcard',
    component: PlantCard,
    tags: ['autodocs'],
};

export default meta;

type Story = StoryObj<typeof PlantCard>;

export const Story: Story = {

};
```



Installing React

I initially installed react-vite for the web ui for our application.

Documentation (in- Progress)

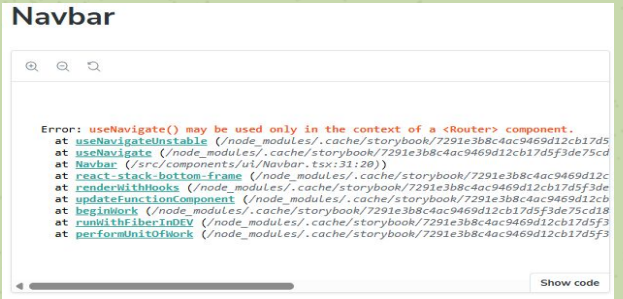
Using Storybook for auto documentation purposes.

Challenges & Insights (Omar)

A big issue that I've had during this Milestone was workload. Having lots of other classes in parallel, and procrastinating more than I'd like to resulted in some difficulty for me too.

Documentation Errors:

Using Storybook to generate documentation has given me some errors.



Info Ledger:

- I struggled understanding the useContext syntax in react for this component. This has been a delay for me, and the reason I was unable to complete the Info Ledger