Project Summary

Project Summary

By: Cody Cuento

Overview

The **Butterfly Tracker** project was developed as part of our Software Engineering course with the goal of creating a system for **monitoring and analyzing butterfly activity in outdoor environments**. The project combines a web-based software platform with data collected from field-deployed camera systems, providing researchers and users with a centralized way to view butterfly sightings, examine migration patterns, and manage monitoring devices.

Our team approached the project as a full software engineering effort rather than simply building an individual application. Throughout development, we worked through the major stages of the software development lifecycle, including **requirements gathering, system design, implementation, testing, integration, and documentation**. We also worked collaboratively using Github, dividing development responsibilities across the team while regularly integrating and reviewing each other's work.

Problem and Motivation

Monitoring butterfly populations can provide useful information about **migration patterns, population changes, habitat conditions, and broader environmental trends**. However, collecting this information manually can require significant time and effort, particularly when observations need to be gathered across multiple locations.

Our project explored how software and automated monitoring systems could make this process more accessible and scalable. The system was designed around the idea that cameras deployed in the field could collect butterfly observations that would later be processed and displayed through a centralized web application.

The primary problem our project aimed to address was:

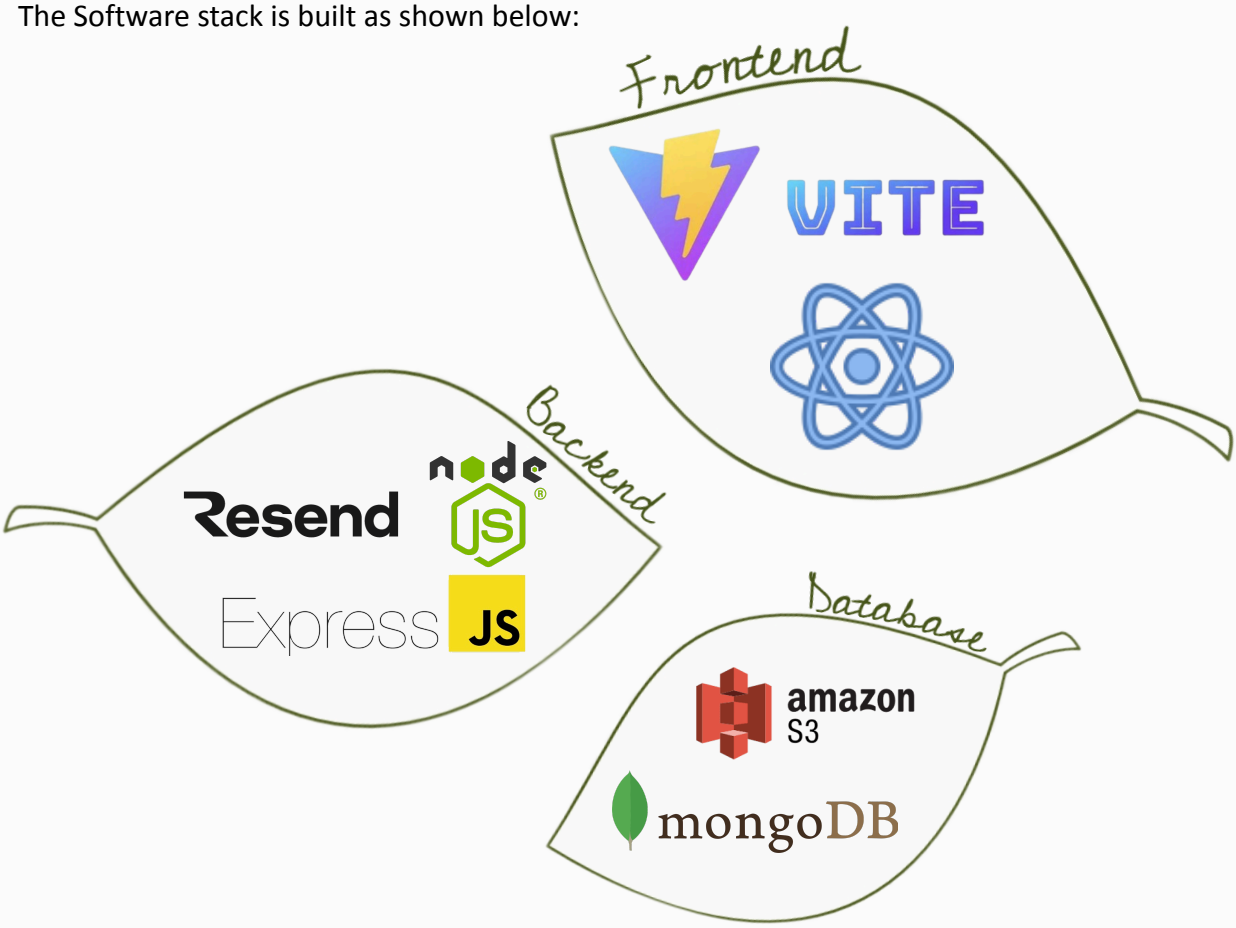
> **Monarch butterfly populations have declined by over 95% since the 1980s. This indicates ecosystem stress and extinction threats, while researchers conduct manual field surveys to gather data.**

Our intended users included **researchers, conservationalists, educators, and other interested parties such as students, wildlife enthusiasts, and more**, who could use the platform to **read and understand monarch butterfly population and migration data for conservation efforts.**

System Overview

The Butterfly Tracker consists of several components that work together to collect, manage, and visualize butterfly observation data.

The Software stack is built as shown below:



A major feature of the application is its visualization of butterfly activity. The platform uses **Google Maps API and Recharts** to display information through **heatmaps and graph visualizations**. These visualizations allow users to interpret large collections of observations more easily than they could from raw data alone.

Development Process

Our team followed **iterative development** during the term. Work was organized into **milestones and issues**, with team members responsible for different areas of the application. We used **Github Projects** to track development and **Git** for version control.

During each development cycle, we **distributed tasks, set deadlines, and reviewed each others' work to maintain a similar work balance** among the team. This process allowed us to continuously evaluate the state of the application and adjust our priorities as requirements changed.

One important aspect of development was coordinating work across different parts of the system. Features frequently depended on multiple components, such as the frontend interface, API endpoints, authentication state, and stored data. As a result, communication and integration were important throughout the project.

Some of the major technical challenges we encountered included **uneven workloads**, **absent team members**, and **competing academic responsibilities**. For example, **as a team we would communicate our availability throughout the week** so that we can plan our responsibilities accordingly. These challenges required us to reconsider parts of our original design and make changes as the project evolved.

Testing and Quality Assurance

Testing was incorporated into the development process to verify that individual components and larger application workflows behaved as expected. We used **Vitest + React Testing Library** to test features including **authentication, loading states, and maps functionality**.

Our testing included **unit, component, integration, and end-to-end** tests for functionality such as **authentication, camera management, maps, analytics, and user profile functionality**. We also used **coverage reports** to identify areas that required additional validation.

Testing revealed several issues during development, including **an issue with our filter, sort, and search system for the database, where we found our queries would turn up the wrong data**. Resolving these problems helped improve both the reliability of the application and our understanding of how changes in one component could affect other areas of the system.

Final Outcome

By the end of the project, our team produced a functional prototype capable of **storing, reading, writing, and viewing recorded data by a network of cameras**. Users can **build a camera**, **view camera/sighting data**, and **contribute to the mission at Valley Eco**, while the system provides **maps, analytics, and tools to query the database**.

Although the project reached **a complete working prototype** there are several areas that could be expanded with additional development time. Future improvements could include **support for networking**, **a better researcher application process**, and **improvements to the camera build**. A larger deployment could also explore **the same system for other species, scaling the project to any observable species**.

Overall, the Butterfly Tracker project provided experience developing a **multi-component software system within a team environment**. Beyond implementing individual features, the project required us to consider software architecture, requirements, testing, version control, integration, and communication throughout the development lifecycle. The resulting system demonstrates how software engineering techniques can be applied to a real-world environmental monitoring problem.

Poster Presentation

The Kaleidoscope (CSE-330) Project Partner: Valley Eco

Mentors:
Britne Clifton
Reyn Akiona

Monarch Butterfly Automated Detection Camera

THE PROBLEM

Since the 1980s, the monarch butterfly population has dropped by **more than 95%**. Despite this, monitoring methods are limited to slow **manual surveys**. Developing more efficient tracking could help prevent their **extinction.**

SOFTWARE STACK

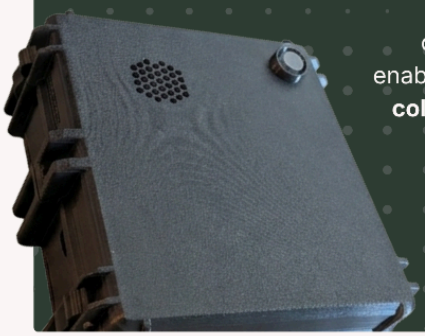
Frontend: VITE

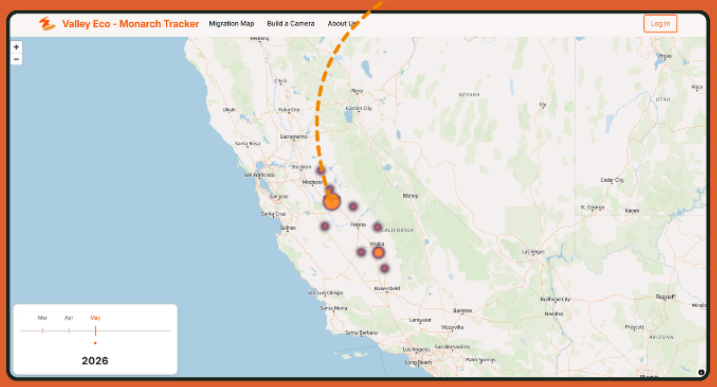
Backend: node JS, Express JS

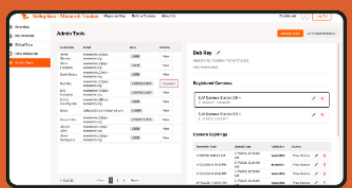
Database: mongoDB, amazon S3

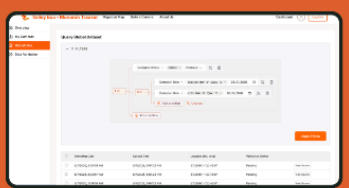
OUR SOLUTION

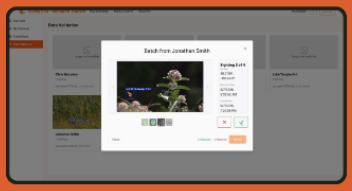
Our team developed a scalable, **automated monarch butterfly detection camera** system complemented by a **web app** that enables researchers to collaboratively **collect, host, aggregate, and query sighting data.**

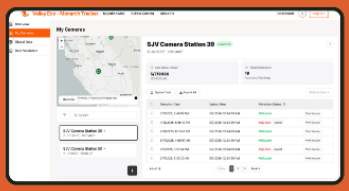












Awards

Awards

Lakireddy Award - University of California, Merced

- Our project was entered into a school wide engineering exposition for software engineers called **Innovate to Grow**.
- Our team was crowned the overall winner of the whole exposition.
- Over 30 teams comprised of 4-5 students participated
    - Each team had a different project to display, based on the respective needs of their own clients
- 5 awards were given out, one for each of the 4 tracks and an overall winner of the exposition.

Image of my winning team (from left to right: Ethan Carr, Mario Velez, Sharvin Manjrekar, Cody Cuento, Ellis Hampton)