

Implementing a New Web Application Interface
and Butterfly Data Visualization Tools
for the Beecology Project
Major Qualifying Project



A Major Qualifying Project submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the Degree of Bachelor of Science in
Computer Science and the Degree of Bachelor of Science in Data Science

Submitted to:

Professor Carolina Ruiz
Department of Computer Science
Data Science Program
Professor Elizabeth F. Ryder
Department of Biology and Biotechnology
Bioinformatics and Computational Biology Program

Submitted By:

Joanna Hu	William Dufault
Bernhard Nordemann	Mattheus Faria
Megan Sin	Steven Frisch
Aria Yan	Colin Fyock

Date:

March 22, 2024

This report represents the work of eight WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review.

Authorship

Chapter and Section Number	Primary Authors	Primary Editors
1.0 Introduction	Colin Fyock, Joanna Hu, Megan Sin	William Dufault, Mattheus Faria, Stevie Frisch
2.1 The Beecology Project	Mattheus Faria	Aria Yan
2.2 Pollinator Population Concerns	Joanna Hu	William Dufault, Megan Sin, Aria Yan
2.2.1 Native and Invasive Plant-Pollinator Interactions	Megan Sin	Megan Sin
2.2.2 Bee and Butterfly Populations	Joanna Hu, Aria Yan	Megan Sin
2.2.3 Prevention and Restoration	Joanna Hu, Megan Sin	Megan Sin
2.3 Importance of Butterflies in the Ecosystem	Megan Sin	Megan Sin
2.4 Butterfly Taxonomy	Joanna Hu	Megan Sin
2.5 Aspects of Citizen Science Projects	Megan Sin	Megan Sin
2.5.1 Participation in Citizen Science Projects	Megan Sin	Megan Sin
2.6 Impact of Data Visualizations	Joanna Hu	Megan Sin, Aria Yan
2.6.1 Data Visualizations Within eBird	Joanna Hu, Megan Sin	Megan Sin, Aria Yan
2.7 User Interface and User Experience Design	Stevie Frisch	Mattheus Faria, Colin Fyock
2.7.1 User Interface and Design Within iNaturalist and Other Web Applications	Stevie Frisch	Mattheus Faria
2.8 User Engagement and Motivations	Megan Sin	Megan Sin, Aria Yan
3.1 Overview of Development Process	William Dufault	Mattheus Faria, Colin Fyock
3.2 Initial Plans to Upgrade From Angular 8 to 16	William Dufault	Mattheus Faria, Colin Fyock

3.2.1 Rationale for Angular Upgrade	William Dufault	Mattheus Faria, Colin Fyock
3.2.2 Angular Upgrade Process	William Dufault	Mattheus Faria, Colin Fyock
3.2.3 Reverting to Angular 8	William Dufault	Mattheus Faria, Colin Fyock
3.3 User Experience Design and Prototyping	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.3.1 Why Do We Prototype?	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.3.2 Figma Prototyping	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.3.3 Mobile and Desktop Layout Improvements	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.3.4 Converting From Figma Into Production	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.3.5 Revisiting Figma For Iterations	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.3.6 Designing for Accessibility in Mind	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.4 Implementation of Visual Changes	William Dufault	Mattheus Faria, Colin Fyock
3.4.1 Bulma Implementation & CSS Difficulties	Stevie Frisch	William Dufault, Mattheus Faria, Colin Fyock
3.4.2 Streamlining the User Experience with a Navigation Redesign	William Dufault, Colin Fyock	Mattheus Faria

3.4.3 Additions to the B-Dex	Mattheus Faria	William Dufault, Colin Fyock
3.4.4 Redesigning the Log View Page	William Dufault, Stevie Frisch	Mattheus Faria, Colin Fyock
3.4.5 Redesigning the Log Creation Pages	William Dufault	Mattheus Faria, Colin Fyock
3.4.6 Redesigning the Accounts Tab	Colin Fyock	William Dufault, Mattheus Faria
3.4.7 Redesigning the Tutorial	Colin Fyock	William Dufault, Mattheus Faria
3.4.8 Documentation Additions	Mattheus Faria, Colin Fyock	William Dufault
4.1 Extending Bee Visualizations to Butterflies	Megan Sin	Bernhard Nordemann
4.2 Creating Butterfly Data Selection	Bernhard Nordemann, Megan Sin	Aria Yan
4.3 Summary Statistics	Aria Yan	Aria Yan
4.4 Separating Bee and Butterfly Data Analysis	Megan Sin	Bernhard Nordemann
4.5 Common Features to Butterfly Visualization Tools	Bernhard Nordemann, Megan Sin	Megan Sin
4.5.1 Butterfly Filter Dropdown Component	Bernhard Nordemann	Megan Sin
4.5.2 Butterfly Diversity Toggle Component	Joanna Hu, Megan Sin	Megan Sin
4.5.3 Butterfly Family, Subfamily, Species Global Coloring	Bernhard Nordemann	Megan Sin
4.6 Descriptions for Each Visualization Tool	Megan Sin	Aria Yan
4.6.1 Bar Chart Comparison Tool	Joanna Hu, Aria Yan	Megan Sin
4.6.2 Diversity Heatmap Tool	Joanna Hu, Aria Yan	Megan Sin

4.6.3 Location Map Tool	Joanna Hu, Megan Sin	Bernhard Nordemann, Megan Sin
4.6.4 Season Chart Tool	Joanna Hu, Aria Yan	Megan Sin
4.6.5 Summary Heatmap Tool	Megan Sin, Aria Yan	Aria Yan
4.6.6 Floral Network Tool	Bernhard Nordemann	Megan Sin
5.1 Final Thoughts	Matheus Faria, Joanna Hu	William Dufault, Colin Fyock
5.2 Challenges Faced	Matheus Faria, Joanna Hu	William Dufault, Colin Fyock
5.3 Future Work	Matheus Faria, Colin Fyock	William Dufault
5.3.1 Additional Bee-ID Web Application Implementations	Matheus Faria, Colin Fyock	William Dufault
5.3.2 Improvements to Butterfly Visualizations	Joanna Hu	Megan Sin, Aria Yan
5.3.3 Upgrading to the Latest Angular Version	William Dufault, Matheus Faria	Colin Fyock
5.3.4 Migrating Beecology to Amazon Web Services	Matheus Faria	William Dufault, Colin Fyock

Table of Contents

Abstract	11
Acknowledgements	12
1. Introduction	13
2. Background	15
2.1 The Beecology Project	15
2.2 Pollinator Population Concerns	16
2.2.1 Native and Invasive Plant-Pollinator Interactions	17
2.2.2 Bee and Butterfly Populations	18
2.2.3 Prevention and Restoration	19
2.3 Importance of Butterflies in the Ecosystem	19
2.4 Butterfly Taxonomy	20
2.5 Aspects of Citizen Science Projects	22
2.5.1 Participation in Citizen Science Projects	23
2.6 Impact of Data Visualizations	24
2.6.1 Data Visualizations Within eBird	25
2.7 User Interface and User Experience Design	26
2.7.1 User Interface and Design Within iNaturalist and Other Web Applications	27
2.7.2 User Engagement and Motivations	28
3. Methods and Results: Web Application Development	30
3.1 Overview of Development Process	30
3.2 Initial Plans to Upgrade From Angular 8 to 16	31
3.2.1 Rationale for Angular Upgrade	31
3.2.2 Angular Upgrade Process	32
3.2.3 Reverting to Angular 8	33
3.3 User Experience Design and Prototyping	34
3.3.1 Why Do We Prototype?	35
3.3.2 Figma Prototyping	35
3.3.3 Mobile and Desktop Layout Improvements	36
3.3.4 Converting From Figma Into Production	41
3.3.5 Revisiting Figma For Iterations	43
3.3.6 Designing for Accessibility in Mind	43
3.4 Implementation of Visual Changes	44
3.4.1 Bulma Implementation & CSS Difficulties	44
3.4.2 Streamlining the User Experience with a Navigation Redesign	45
3.4.3 Additions to the B-Dex	48
3.4.4 Redesigning the Log View Page	51
3.4.5 Redesigning the Log Creation Pages	52

3.4.5.1 Redesigning the Primary Log Creation View	53
3.4.5.2 Redesigning the Bee Identification Page	54
3.4.5.3 Redesigning the iNaturalist Flower Page	56
3.4.5.4 Redesigning the Manual Butterfly View	59
3.4.6 Redesigning the Accounts Tab	61
3.4.7 Redesigning the Tutorial	64
3.4.8 Documentation Additions	67
4. Methods and Results: Data Visualization	70
4.1 Extending Bee Visualizations to Butterflies	71
4.2 Creating Butterfly Data Selection	73
4.3 Summary Statistics	76
4.3.1 Butterfly Summary Statistics	77
4.4 Separating Bee and Butterfly Data Analysis	79
4.5 Common Features to Butterfly Visualization Tools	80
4.5.1 Butterfly Filter Dropdown Component	80
4.5.2 Butterfly Diversity Toggle Component	81
4.5.3 Butterfly Family, Subfamily, Species Global Coloring	82
4.6 Descriptions for Each Visualization Tool	83
4.6.1 Bar Chart Comparison Tool	83
Bee Bar Chart Comparison Tool	83
Butterfly Bar Chart Comparison Tool	84
4.6.2 Diversity Heatmap Tool	85
Bee Diversity Heatmap Tool	85
Butterfly Diversity Heatmap Tool	86
4.6.3 Location Map Tool	88
Bee Location Map Tool	88
Butterfly Location Map Tool	89
4.6.4 Season Chart Tool	90
Bee Season Chart Tool	90
Butterfly Season Chart Tool	91
4.6.5 Summary Heatmap Tool	95
Bee Summary Heatmap Tool	95
Butterfly Summary Heatmap Tool	95
4.6.6 Floral Network Tool	96
Bee Floral Network Tool	96
Butterfly Floral Network Tool	97
5. Conclusion	100
5.1 Final Thoughts	100
5.2 Challenges Faced	101
5.3 Future Work	102
5.3.1 Additional Bee-ID Web Application Implementations	102

5.3.2 Future Improvements to Butterfly Visualizations	108
5.3.3 Upgrading to the Latest Angular Version	109
5.3.4 Migrating Beecology to Amazon Web Services	110
References	112

Table of Figures

Figure 1. Diagram of butterfly taxonomy of relevant species.	21
Table 1. Citizen science projects related to pollinators.	23
Figure 2. The user interface of Figma and an example of how an early prototype of the Beecology web app title page was created.	34
Figure 3. Version 5 of the web app as shown on a wide screen.	37
Figure 4. An early prototype of the desktop view of the log view page for version 6.	38
Figure 5. A late prototype of the desktop view of the manual butterfly page for version 6.	39
Figure 6. A late prototype of the mobile view of the bee log creation page for version 6.	40
Figure 7. Desktop and mobile prototypes of the B-Dex page for version 6.	41
Figure 8. The version 5 B-Dex (top) compared with the version 6 prototype using Figma (left) and its production counterpart using Bulma (right).	42
Figure 9. The home screen of the web application for version 5 (above) compared to the production home screen of the web app for version 6 on desktop (below).	47
Figure 10. B-Dex Butterfly View Comparison.	49
Figure 11. The bee/butterfly toggle that is used on the B-Dex page in version 6.	50
Figure 12. The new search feature that was added to the B-Dex page in version 6.	51
Figure 13. The production bee log view page in version 6.	52
Figure 14. The primary log creation view in version 5 (top) compared to version 6 (bottom).	54
Figure 15. The bee identification page in version 5 (top) compared against the page in version 6 (bottom).	56
Figure 16. iNaturalist flower identification in version 5 (top) compared against the same page in version 6 (bottom).	58
Figure 17. The butterfly manual identification page in version 5 (top) compared against the page in version 6 (bottom).	60
Figure 18. Example of the popup of the selected video frame on version 5 of the web app.	61
Figure 19. The account page in version 5 (top) compared against the account page in version 6 (bottom).	63
Figure 20. The tutorial page in version 5 of the web application.	65
Figure 21. The version 6 tutorial page on mobile (top) and desktop (bottom).	66
Figure 22. 'Bee Data Selection' Page.	72
Figure 23. 'Data Analysis' page with the six visualization tools.	72
Figure 24. Searchable checklist tree dropdown component for butterfly species.	74
Figure 25. Previous iteration of the 'Data Selection' page.	75
Figure 26. New iteration of the 'Bee Data Selection' page.	75
Figure 27. Bee Summary Statistics showing species, gender, behavior, and tongue length.	77
Figure 28. Bee Summary Statistics showing flower color, flower shape, and decade.	77
Figure 29. Butterfly Summary Statistics showing species, behavior, and tongue length.	79
Figure 30. Butterfly Summary Statistics showing flower color, flower shape, and decade.	79
Figure 31. Dropdown Filter Component.	81
Figure 32. Searchable Dropdown Filter Component.	81

Figure 33. Diversity Toggle Component.	82
Figure 34. Bee Bar Chart Comparison Tool.	84
Figure 35. Butterfly Bar Chart Comparison Tool.	84
Figure 36. Bee Diversity Heatmap Tool.	86
Figure 37. Butterfly Diversity Heatmap Tool Selection Tab.	87
Figure 38. Butterfly Diversity Heatmap Tool Stacked Bar Tab.	87
Figure 39. Selection and Stacked Bar Chart Toggle.	88
Figure 40. Bee Location Map Tool.	89
Figure 41. Butterfly Location Map Tool.	90
Figure 42. Bee Season Chart Tool.	91
Figure 43. Example of Season Chart at the family level.	92
Figure 44. Example of Season Chart at the subfamily level.	92
Figure 45. Example of Season Chart at the species level.	93
Figure 46. Threshold selection component.	93
Figure 47. Example of Season Chart with greater than or equal to 15 observations.	94
Figure 48. Example of Season Chart with less than 15 observations.	94
Figure 49. Bee Summary Heatmap Tool.	95
Figure 50. Butterfly Summary Heatmap Tool.	96
Figure 51. Bee Floral Network Tool.	97
Figure 52. Butterfly Floral Network Visualization.	98
Figure 53. Butterfly Floral Network Visualization on Hover.	98
Figure 54. An early Figma prototype for the medals and level features for version 6.	105
Figure 55. An early Figma prototype for the landing page and user spotlight in version 6.	107

Abstract

The Beecology Project is dedicated to tracking pollinator populations through the use of web tools that allow citizen scientists to collect and analyze bee and butterfly data. This MQP revamped the interface of the web application to support mobile, tablet, and desktop devices through a rigorous prototyping process. The previous version (version 5) of the web application resembled a mobile phone on all devices. Our new version (version 6) implements a new navigation menu to improve the user experience on widescreen devices. Additionally, this upgrade allows easier identification of pollinators, further supporting Beecology's mission. Altogether, the new interface greatly improves the user experience by enhancing the layout and making navigation easier for users.

The Beecology Project is a citizen science project that aims to monitor pollinator populations and their floral interactions. Web-based visualization tools allow users to visualize where pollinators are located, how many different species are observed, and which flowers they interact with. In this project, we expanded on existing bee visualization tools and developed new visualization tools tailored for analyzing butterfly observations using Angular, TypeScript, JavaScript, and D3.js. We also created pages for butterfly data selection and the new visualizations. Key features we implemented across all the butterfly tools include hierarchical filters, diversity toggles, and consistent color mapping, enabling users to effectively explore and analyze the diverse number of butterfly species.

Acknowledgements

We would like to acknowledge the contributions made to this project by Oliver Yasuna through his extensive knowledge surrounding the web application, Grace Phillips for her aid in testing and adding flower images, the WPI Academic & Research Computing (ARCUnix) team through their GitLab expertise, Professor Robert Gegear for his insight into the trends of the Beecology user base, and, most importantly, our advisors Professor Elizabeth Ryder and Professor Carolina Ruiz for their support and timely feedback throughout the project.

1. Introduction

Pollinators are responsible for the existence of around 35% of the world's food crops and about 75% of the world's flowering plants (Lawrence, 2022). They are vital to nearly every ecosystem in each inhabited continent. In recent years, population declines have been observed for many common pollinators. For example, the monarch butterfly saw a decline of over 80% in the last decade (Semmens, 2009). The alarming decline of these pollinators has led to more investigation into their foraging habits. Creating a record that connects plants to pollinators by keeping track of which plants support which pollinators over time allows scientists to observe the species with the greatest decline. Tracking the connection between pollinators and the plants they pollinate can give scientists important information about the stability of habitats critical to conservation and restoration efforts (Grace, 2016). Sustaining biodiversity is crucial in maintaining an ecosystem that is resilient to environmental pressures.

To assist in this effort, the Beecology project aims to create a database containing ecological data on pollinators, specifically bumblebees and butterflies, as well as their preferred plant species. The Beecology project is a collaborative effort of UMass Dartmouth professor Robert Gegear, WPI professors Liz Ryder and Carolina Ruiz, and their many students. Professor Gegear is an expert ecologist and spends a significant amount of time collecting and analyzing pollinator and plant data as well as training users on the Beecology web app and the analysis and visualization tools, both citizen scientists and students. The Beecology web application allows citizen scientists to submit photos and videos of bees and butterflies in their natural habitat to the Beecology database. These submissions track the species of the bee or butterfly, the location it was seen, and the flower it was found on. Each of these aspects are important tracking

mechanisms to determine how different species of bees and butterflies are doing in their ecosystem.

The primary goals of this Major Qualifying Project (MQP) were to overhaul the user interface of the web application to include a desktop version and to extend Beecology's data visualization tools to support recently added butterfly data. The Beecology web application lacked a desktop interface in its version 5 release, instead opting for a fixed mobile view on all devices. The update to version 6 in the web application was a success, as we were able to implement completely new interfaces for mobile, tablet, and desktop, resulting in a better user experience. Specifically, the redesigned pages in the application regarding data collection and interpretation saw greatly improved accessibility and functionality by utilizing the additional space on wider devices. Additionally, a previous project expanded the Beecology project to butterflies by allowing users to submit butterfly observations (Pecorale & Poulos, 2022). As a result, there needed to be a way to visualize this new data in a similar format as the bee visualization tools. It was important for the new butterfly visualization tools to maintain a consistent design with the bee visualization tools to ensure users would not have to learn how to use completely new tools. However, a significant challenge emerged as there are many more butterfly species in New England than bumblebee species (121 versus 11). As we expanded on the existing bee visualization tools, we incorporated features into the new butterfly visualization tools, such as a dropdown filter selection and diversity toggle. The addition of these features aimed to address the challenge of effectively visualizing the numerous butterfly species while ensuring the tools remain intuitive and user-friendly.

2. Background

2.1 The Beecology Project

Founded in 2016, the Beecology Project aims to educate students and unite citizen scientists to learn about and analyze the declining pollinator population. Through the use of a web application, users can take pictures and videos of different species of bees and butterflies to be identified, as well as the plant species they are visiting, and submit these observations to an actively growing database of classifications and locations. This pollinator data can be viewed in an index of pollinators and used to create various datasets to be visualized and analyzed using several tools created by the Beecology Project team. Beecology is also used in a high-school curriculum called the Bio-CS Bridge, which was designed to teach students to connect computer science and biology concepts in a classroom setting. The Bio-CS Bridge puts an emphasis on how computer science can be used to approach important biological questions, such as how best to support the declining pollinator population, and uses the domain provided by biology for computer scientists to design software to collect, manipulate, and analyze data to address those questions.

The project initially started off as an Android application that recorded bumblebee classifications, identified the species of bumblebee and plant, and stored them in a database to be viewed later. The application itself was designed to be used offline, where users could locally store their records in logs saved onto their phones to be later submitted to the database when connected to the internet. Later, teams working on the project would add the capabilities to analyze and visualize the data collected by the citizen scientists. In 2019, the Beecology team created the Bio-CS Bridge and promoted the Beecology Project to high school computer science and biology teachers in Massachusetts. The initial proposal of The Bio-CS Bridge included the

development of a Bio-CS integrated curriculum, which provided students with an interdisciplinary approach to solving complex real-world problems, mainly addressing the decline in pollinator populations (Ryder et al., 2020). The Bio-CS Bridge curriculum teaches students about the connection between biology and computer science by highlighting how software and data analysis tools can help to answer complex biological questions and how biology, as a subject, is a bountiful source of data that enables computer scientists to collect and analyze the many facets of the biological spectrum.

Further development since the creation of the Beecology web application included adding more visualization and analytical capabilities and introducing the ability to identify and add butterflies to the database. The previous Beecology MQP report ended by stating that the next steps in the project would be the addition and upgrade of data visualizations for butterfly data, as well as the expansion of data analysis tools (Strano, 2023).

2.2 Pollinator Population Concerns

Pollinators play a key role in the ecosystem and reproduction of plants. Native pollinators are typically keystone species that are critical to the maintenance of functioning ecosystems, and also play an important role in agriculture. There are many types of pollinators, like birds, bats, bees, and butterflies. Among these types of animals, many species are native to different areas of the world. For example, Bumblebees (genus *Bombus*) are common pollinators native to North America (Inouye, n.d.), while honeybees, brought over from Europe, are invasive in North America (Irwin & Mawdsley, n.d.).

There is abundant evidence highlighting the transformation and decline of pollinator communities. Various factors related to land use and management, including urbanization,

deforestation, pollution, climate change, diseases, and the introduction of exotic species, have a significant impact on pollinators by altering both non-living and living elements (LeBuhn et al., 2021). Changes in the landscape or heightened pollution levels can directly affect pollinator habitats and available resources. Moreover, the transmission of diseases among pollinators or the presence of non-native species can further exacerbate the negative impact on their populations.

2.2.1 Native and Invasive Plant-Pollinator Interactions

To fully comprehend ecosystem stability and biodiversity conservation, it is essential to understand the dynamics of native and invasive plant-pollinator interactions. Co-flowering plants, which bloom or flower around the same time, often share pollinators because these pollinators visit all of the different plant species within a community (McKinney & Goodell, 2011). Because of this, non-native plants can appear to be well-integrated into native plant-pollinator ecosystems. As local pollinators also visit non-native plants, this can occasionally decrease pollinator visits to native plants. Changes in pollinator visitation to native plants can alter the amount and quality of pollen delivered to native plant stigmas, potentially influencing pollinator populations' dynamics and composition and their foraging behaviors towards indigenous plants (McKinney & Goodell, 2011). Over time, these interactions between invasive and native plants for pollinators may prompt either divergence (competition) or convergence (facilitation) of floral characteristics like flowering timing, floral structure, and reward production timing among native species (McKinney & Goodell, 2011).

2.2.2 Bee and Butterfly Populations

Based on 119 studies on the effects of climate change on bumblebees and butterflies, species are emerging from hibernation an average of 17 days earlier than they used to and mismatching with floral resources (Whipple et al., 2023). Bumblebees and butterflies also saw a 25% decrease in their range distribution, the geographical area where these species can be found (Whipple et al., 2023). Declines for different species are often not uniform. For species native to New England, *Bombus terricola* and *Bombus affinis* have disappeared from many sites where they have previously been observed (Thorp & Shepherd, 2005). In a survey conducted over the course of two years, it is apparent that some species such as *Bombus affinis*, *Bombus ferdigus*, *Bombus terricola*, and *Bombus vagans* have declined significantly in abundance. In contrast, species like *Bombus bimaculatus* and *Bombus impatiens* have increased proportionally. Between the 1970s and the 2000s, Eastern North American bumblebees have declined in species richness, evenness, and diversity.

When these species emerge early from winter hibernation and cannot find the flowers they need for food or reproduction, this can disrupt pollination and affect the population of these species over the years. According to Whipple et al. (2023), the distribution of 214 bumblebee and butterfly species has experienced a range decline over the past years. Nearly 26% of North American bumblebee species are threatened with various degrees of extinction risk (Cameron & Sadd, 2020). Most critically threatened is the *Bombus affinis* species, which was placed on the US Fish and Wildlife Service's endangered species list in 2017 (Cameron & Sadd, 2020). Within Beecology's own data, only one observation of *Bombus affinis* has occurred from 1996 to 2004. Therefore, there is an urgent need for prevention measures to preserve bumblebee and butterfly populations as they play a crucial role in the world's ecosystem.

2.2.3 Prevention and Restoration

If prevention measures are not implemented to combat the decline of bumblebee and butterfly species, the world will face an inevitable crisis. Multiple large-scale restoration efforts have emerged globally to counteract ecosystem degradation and biodiversity loss (Moreno-Mateos et al., 2020). However, restoration often needs to be improved to offset the loss. To enact policies to help these species, we must educate more people about the problem and help scientists with their research on bumblebee and butterfly distributions and behaviors (Pecorale et al., 2022).

Understanding not just species distribution but pollinator biology and ecosystem interactions is vital. Examining relationships between pollinators and native plants can inform resilient, climate-adaptive habitat restoration as individual pollinator species can have varying plant preferences. The Beecology project aims to build such knowledge across New England by engaging citizen scientists in data collection focused uniquely on ecological interactions—a frequently overlooked but essential kind of data for conservation. Though small now, this growing open database helps educate the public about pollinators while advancing research needed to preserve biodiversity and fragile beneficial relationships.

2.3 Importance of Butterflies in the Ecosystem

Butterflies play a vital role in the ecosystem. They are pollinators and also serve as indicators of how healthy an ecosystem is. Butterflies have a clear taxonomy and their biology and life history are well-defined. Their physiological tolerances, habitat, temperature, and light requirements can be quantified and correlated with changes in ecosystem conditions (Ghazanfar

et al., 2016). Butterflies and plants have a mutually beneficial relationship as butterflies eat nectar from flowers and carry pollen as they do so. Many butterfly species, across all families, migrate and spread pollen among plants that are geographically apart from one another (Chowdhury et al., 2021). Butterflies are known for their long-distance flight capabilities, especially during migration events. Monarch butterflies are famous for their migratory journeys that span thousands of miles. This pollen migration can lead to genetic variety in plant species and improve plants' chances of survival against various diseases. Thus, plant species diversity and distribution closely correlate to butterfly species (Shrestha et al., 2020). Butterflies also serve as a food source for other species like birds, reptiles, and amphibians (Ghazanfar et al., 2016).

2.4 Butterfly Taxonomy

Taxonomy is the science of naming, describing, and classifying organisms. This includes all plants, animals, and microorganisms of the world. Taxonomists identify, describe, and arrange species into different classifications based on morphological, behavioral, genetic, and biochemical observations (What is taxonomy?, 2010). Butterflies and moths are members of a taxonomic insect group called Lepidoptera, also known as Order Lepidoptera (Butterfly and moth taxonomy, n.d.). Within this order, the different butterflies and moths are placed into many subfamilies and species (Figure 1).

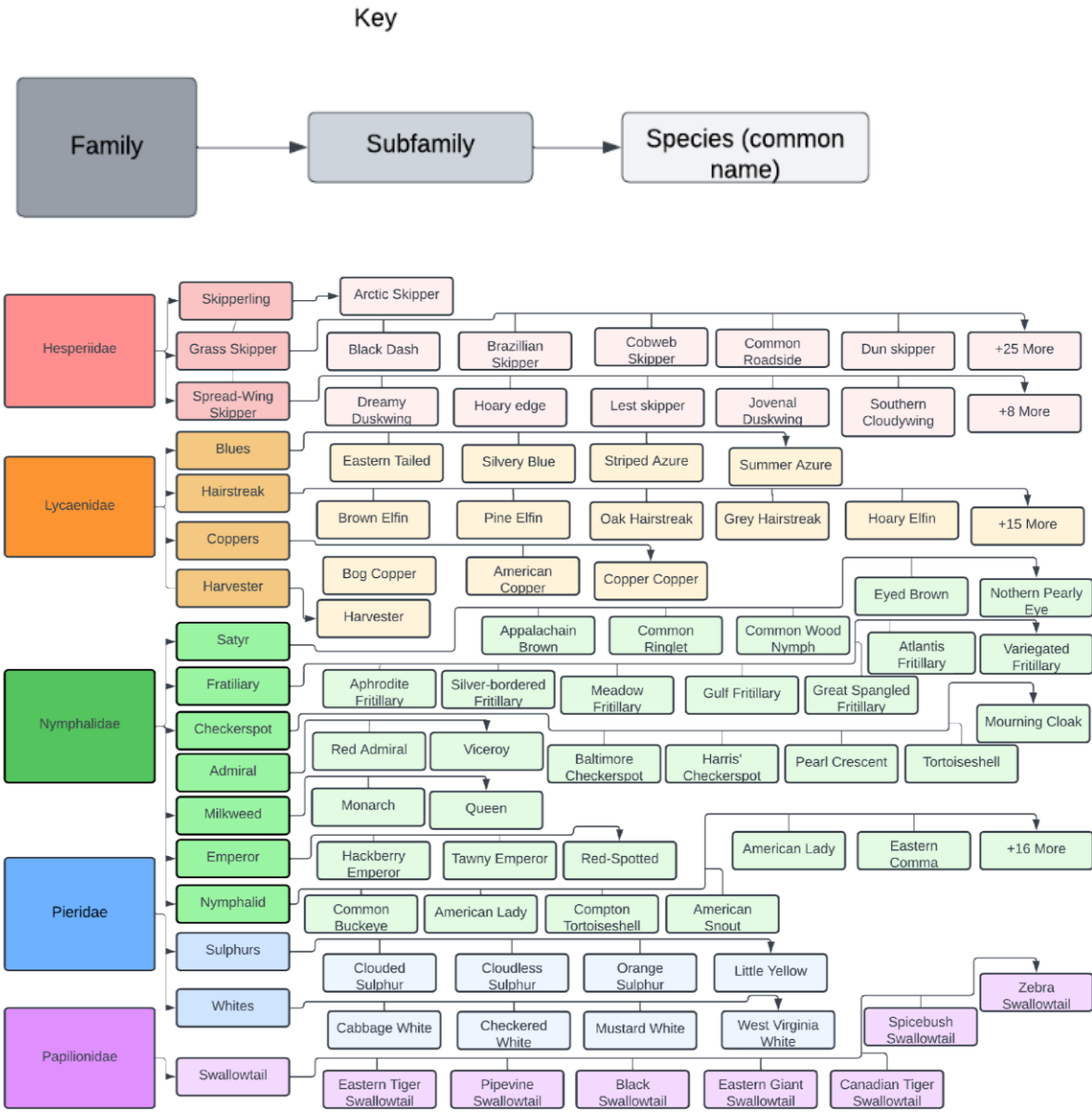


Figure 1. Diagram of butterfly taxonomy of relevant species. Each family listed belongs to the order Lepidoptera (Strano, 2023).

Within the current Beecology database, which focuses on pollinators native to the Northeast U.S., there are only 11 bumblebee species, all members of the genus *Bombus*. However, there are 121 butterfly species (Cech & Tudor, 2007). This makes it difficult to translate the bumblebee data visualizations directly to butterflies, as there are 11 times more

butterfly species than bumblebees. Therefore, another intuitive way of managing and representing the butterfly data is needed.

2.5 Aspects of Citizen Science Projects

More and more ecology projects involve citizens who volunteer to collect and/or process data (Silvertown, 2009). This is commonly known as citizen science—scientific research conducted with participation from the general public. Some examples of citizen science projects relating to bee data can be seen in Table 1. Being able to crowdsource data is highly valuable to scientists behind the citizen science project’s research and anyone curious about the topic. For the everyday user, citizen science projects can provide easy access to information in real time. As for scientists, they are able to create new hypotheses and guide future research initiatives based on a vast amount of data. Data is a vital tool for science and conservation.

Project Name	Summary
The Bee Conservancy (TBC)	TBC’s mission is to protect all bee species. They have a Community Service Project on the iNaturalist application (https://www.inaturalist.org/projects/the-bee-conservancy-community-science-projects). Users can participate by snapping a photo of their observations, identifying the species using iNaturalist’s technology, recording when and where they saw it, and uploading it to share with the community. There is a map of observations displaying where observations were found, photos of recent observations, as well as user leaderboards for different challenges.

<p>Planet Bee Foundation: The ZomBee Watch Program</p>	<p>This citizen science project tracks the spread of <i>Apocephalus borealis</i>, an invasive fly species that kills honeybees and native bees. They also have supplementary material for teachers, like virtual STEM lessons that guide students through the scientific method to collect and input data. This is one of four Planet Bee Citizen Science Projects. To participate in these projects, users must fill out an interest form on their website (www.planetbee.org/citizen-science).</p>
<p>Bumble Bee Watch</p>	<p>Bumble Bee Watch is a community science project with the goal of tracking and conserving North America’s bumble bees (https://www.bumblebeewatch.org/). Users can upload photos of bumble bees to start a virtual collection; identify the bumble bees in their photos and have their identifications verified by experts; help locate rare or endangered populations of bumble bees; learn about bumble bees, their ecology, and ongoing conservation efforts; and connect with other community scientists. Users can also explore a list with all of the observations and also visualize the data on a heat map.</p>
<p>Mass Butterfly Club (North American Butterfly Association)</p>	<p>North American Butterfly Association is an international wildlife conservation nonprofit working toward the goal of creating a world where butterflies thrive by shifting the paradigm from collecting butterflies to observing them. Users can submit butterfly sightings and include date, location, and species data (https://sightings.naba.org/).</p>

Table 1. Citizen science projects related to pollinators.

2.5.1 Participation in Citizen Science Projects

Most citizen science projects need enough volunteers over an extended period to be successful (Luna et al., 2018). Citizen science projects have been around for decades, but with the widely accessible technical tools we have today, such as smartphones and computers, it is now easier than ever for citizens to gather and report data. Mobile and web applications to support environment and biodiversity monitoring are often used to record the presence and location of native and invasive species and to identify patterns in species behavior (Luna et al., 2018). The success of a citizen science project that uses a mobile or web application will be

determined by the application's usability from the user's perspective. Additional attributes of a successful project include its effectiveness in achieving research goals, the capacity to engage and communicate with the public, and its ability to sustain engagement for an adequate duration (Luna et al., 2018).

The design and development of mobile and web interfaces should simplify data collection, invite involvement from a diverse group of individuals, and ideally increase scientific knowledge (Luna et al., 2018). Crowdsourced contributions through a website interface can include entering and uploading observation data that was recorded on a smartphone and processing and analyzing data. Mobile citizen science projects often provide a website and a smartphone application (Luna et al., 2018). If smartphone application development is not possible, it would be beneficial to implement a mobile-friendly version of the website.

Considering the European Citizen Science Association's (ECSA) 10 Principles of Citizen Science, an important principle to highlight is Principle Four, which states that citizen scientists should receive feedback on how their data is being used and what the research, policy, or societal outcomes are (ECSA, 2015). Another principle to note is Principle Seven, which states that given that there are no privacy concerns, project data, and metadata should be made publicly available. The results should also be published in an open-access format. It is vital that data and its respective analysis are presented in a format that is easy for participants to navigate and understand. One method to deliver this functionality is through interactive data visualizations.

2.6 Impact of Data Visualizations

With all collected data, there needs to be a way to manage and visualize it in order to bring meaning and create an understanding of the data. Data visualization means making graphic

displays of data by combining points, lines, a coordinate system, numbers, symbols, words, shading, and color (Tufte, 2001). The main goal of these visualizations is to represent data and statistics in a way that people can interpret and understand. Data visualizations are used universally to represent data in a way that is simpler to understand compared to looking at just the raw data. Graphics typically are useful for data cleaning, exploring data structure, detecting outliers and unusual groups, identifying trends and clusters, spotting local patterns, evaluating modeling output, and presenting results (Unwin, 2020). Regarding biology, these types of information can be useful in research, revealing trends, discovering new hypotheses, and more for scientists and those simply interested in the specific topic.

2.6.1 Data Visualizations Within eBird

Often, the most effective way to describe, explore, and summarize a large set of numbers or data is to look at pictures of that information (Tufte, 2001). With the current advancements in technology, data visualizations have become more powerful as people can now process much larger sets of data and process data faster. eBird is a web-based community that helps birders (people interested in birds) and scientists learn, research, and collect data on birds across North America. eBird's mission is to harness the power of everyday birders to understand bird distribution and abundance better and identify the factors that influence bird distribution patterns (Sullivan et al., 2009). With all this data from across the continent, data visualizations are useful tools to picture all the collected records together in order to discover patterns and other statistical information about the data. Data visualization and analysis tools featured on the eBird website (www.ebird.org) are accessed by more than one million unique visitors annually (Lagoze, 2014). These visualizations include trend maps displaying abundance trends of select bird species

within a certain location, migration pattern maps, and range maps designed to show species distribution and how far species are expected to travel. These visualizations focus on locations and the species and do not feature any information about interactions these species have with other things in their ecosystems.

EBird provides birders with an easy way to access real-time information about birds they are interested in. For scientists, eBird provides valuable bird occurrence information for more than 180,000 locations in an organized and accessible format, which helps raise questions that stimulate research and suggest ideas (Sullivan et al., 2009; Unwin, 2020). The data can also be used to test and improve bird species' distributional models required to determine which areas should receive conservation attention first and how to manage certain species (Sullivan et al., 2009). As a result, data visualization tools are useful to scientists and biologists for their work and for people who are simply interested in the topic. The ability to create, manage, and manipulate vast, real-time data resources influences how people study biology and conduct conservation research and planning (Sullivan et al., 2009).

2.7 User Interface and User Experience Design

The terms User Interface (UI) and User Experience (UX) are often used interchangeably but hold distinct meanings and cater to different aspects of product design. UI primarily deals with the visual elements, layouts, and interactive components of a product or software. The concept focuses mainly on the tactile experience of a user, or how they interact on screen using buttons, sliders, icons, links, and other design elements. The primary goal of having a good UI is to ensure the interface is intuitive, aesthetically pleasing, and responsive to user commands (Harold, 1990).

UX takes a deeper dive into the overall experience a user has with a product or service. Beyond the visual and interactive elements, UX encompasses the emotions, perceptions, and responses a user might have. This includes understanding user needs and preferences, ensuring ease of use, and evaluating the overall satisfaction a user may have after interacting with a product. The UX of an ecological app goes beyond capturing sightings. The first experience a user has with an app is a crucial part to retaining engagement and interest (Kim, 2019). An educational onboarding process that explains not only how to use the app, but also why tracking these insects is important, is a straightforward technique that encourages user interest. Furthermore, the UX should consider the varying contexts in which the app may be used: in the field with limited connectivity, during inclement weather, or by users who may not have specialized ecological training. Features like offline data logging, or gamified elements that make the data collection process more engaging can enhance the overall UX (Cheng et al., 2019). Additionally, the ability to track data visualizations allow users to see their contributions on a larger scale, which can add a layer of gratification and purpose, fostering continued engagement. Together, both the UI and UX form the shared keystones of a user-centric design, ensuring both seamless interaction and enjoyable experience for a broad range of users.

2.7.1 User Interface and Design Within iNaturalist and Other Web Applications

The UI of an application is an important factor in its success. iNaturalist, an application devoted to identifying different species of plants and animals, excels in these domains, effectively catering to a diverse audience that spans from amateur naturalists to professional scientists (www.inaturalist.org). The app's UI stands out for its intuitive design, featuring simple menus, clear icons, and straightforward labeling. An intuitive UI design simplifies the process of

entering complex data, such as species identification and geolocation, making it accessible to users with varying levels of expertise. In UX, iNaturalist elevates data collection by fostering community engagement, primarily by its platform for crowdsourcing identification of species. When a user uploads an observation, other users can confirm or correct the author's identification. Furthermore, the application is designed to function well under various conditions, including offline scenarios, acknowledging the often unpredictable nature of fieldwork.

The sidebar on YouTube's website is a critical element for navigation that significantly contributes to an intuitive UX (www.youtube.com). It serves as a navigational anchor, offering quick and easy access to a variety of categories and videos, and presents them clearly on the screen. This organized layout reduces cognitive load, allowing users to avoid distractions and remain focused within the app (Schnotz et al., 2007). It also enables users to swiftly transition between different types of content without having to return to the homepage or conduct a new search.

2.7.2 User Engagement and Motivations

One of the greatest challenges in citizen science projects is the motivation for retained user engagement and participation. Since participants are volunteering their time and effort to contribute to the project's success, project initiators have to care for their volunteers and ensure that the project 'gives back' to maintain motivation. Additionally, long-term initiatives with few or no user rewards are more likely to face issues with user retention and attracting new users (Sullivan et al., 2009). Incorporating motivational factors into the design of participants' task flow can enhance sustainable engagement. Motivational categories encompass a variety of factors, including the desire to learn about science, altruism in contributing to the project's

overarching goal, social proof indicating the value of one's actions, extrinsic motivations based on rewards, and intrinsic motivations rooted in participants' personal interest and enjoyment in the project's subject matter (Luna et al., 2018).

One study looked into the motivations for initial and continued engagement in citizen science projects, comparing intrinsic and extrinsic motivations. The results suggested that members with intrinsic motives are more likely to participate frequently and continue to do so for longer periods of time (Aristeidou et al., 2017). With that said, extrinsic motivations still drive participation.

Extrinsic rewards can increase attention to the research being done, particularly in the context of 'gamification,' which is the process of making tasks more like games by employing scoring and a competition aspect (Tinati et al., 2017). Another method is providing a forum to encourage users to have discussions and interact around project tasks (Eveleigh et al., 2014). These techniques focus on sustaining participation rather than attracting new participants. It is important to be cautious when using extrinsic motivations within citizen science as they face the danger of users trying to 'cheat' to earn more rewards. Research norms do not align with gaming norms. Although it may be immoral or wrong to cheat in a game, cheating or faking data entries is in the gray area of scientific misconduct (Kreitmair & Magnus, 2019). For example, it is common for players who are playing video games to post solutions or "cheat codes" for certain levels or problems online. This allows other players to skip levels or bypass difficult obstacles without actually solving them through their own efforts. While applying this practice to gamified citizen science projects, it would not be possible to hold someone who did this responsible for scientific misconduct.

3. Methods and Results: Web Application Development

This MQP has drawn inspiration from a significant number of elements in both iNaturalist and Youtube, along with other similarly designed web applications, to create a UX that is built for efficiency and usability while limiting clutter on screen with unnecessary graphics. Given that the broad target users of the Beecology Project application are ecologists and amateur nature observers, both groups with large age ranges, the interface needed to be intuitive and straightforward. Clear icons representing different navigational pages, for example, needed to be easily recognizable, with color schemes that are both visually pleasing and accessible to users with color blindness (Jefferson et al., 2007). The ease with which users can enter data directly affects the quantity and quality of the data collected. A well designed UI for a tool such as this could mean the difference between a sporadically used app and a powerful, community driven tool for ecological research.

3.1 Overview of Development Process

When deciding what to focus this iteration of the project on, we first assessed the version 5 of the Beecology web application in its functional state. After initial testing, it was clear that the frontend design needed the most attention. We scheduled a meeting with our advisors and proposed the idea of a total redesign of the application, which they agreed with. Upgrading the Angular version of the project was one of the first decisions we made during the early stages of our MQP (see section 3.2 below). After struggling with various package conflicts, TypeScript errors, and other inconsistencies, we decided to discard that idea and shift strictly towards frontend development. We started this by prototyping the newly designed pages of the app in Figma (see section 3.3 below), where our primary focus was to improve the space for desktop

users. After finishing the first draft of these prototypes, we began to implement them into the new version of the application (see section 3.3.4 below). During this stage, we found designs that were either not intuitive or not optimal. To fix this, we revisited our Figma prototypes, and re-implemented them with more thoughtful upgrades (see section 3.3.5 below). After we felt our changes to the prototypes satisfied our objectives, we completed development on the frontend of the project (3.4). The project was not easily hosted on a live server, as we ran into problems with networking and runners, preventing the website from being accessible to others (5.2). However, once runners were finally working, we were able to test a broader user base, and perform small, finalizing tweaks to the project, allowing for a polished, finished product.

3.2 Initial Plans to Upgrade From Angular 8 to 16

In the early stages of modernizing the Beecology web application, we noticed that the live website was using Angular 8, which was released in May 2019. The stable version of Angular, as of the start of this MQP in August 2023, was Angular 16. Therefore, following a discussion with our advisors, we decided that updating the app to the latest version would be one of our first priorities. However, after attempting to update the project, we were forced to roll back to Angular 8.

3.2.1 Rationale for Angular Upgrade

The two main benefits of upgrading Angular versions included improving application security and resolving many of the project's TypeScript errors. It is imperative that software is kept up-to-date, not only because of new features and functionality, but because it provides users with improved security due to new versions implementing patches that solve previous security

issues (Cybersecurity and Infrastructure Security Agency, n.d.). An application that remains on the same version may expose its users to security threats and vulnerabilities that do not exist in later, updated versions.

Another reason to upgrade was to resolve the TypeScript errors in the project, most of which resulted from variables lacking types or type mismatches. While this may have been acceptable when Angular 8 was released, newer versions of TypeScript have stricter guidelines, which a large portion of the code did not follow. Consequently, we decided that if we could upgrade the project to Angular 16 and resolve most, or all, of the TypeScript errors at once, we could focus entirely on redesigning the interface of the application.

3.2.2 Angular Upgrade Process

When we began upgrading the Angular version of the app, we started by attempting to simply update the packages inside of the project to their most up-to-date versions, but this led to further issues with conflicting packages and peer dependencies. After several hours of switching package versions and replacing broken packages with newer alternatives, we concluded that this approach was not viable.

Our next idea was to use Angular's Update Guide Tool and to follow the recommended steps to update our project, but this also did not work (update.angular.io). We tried following the steps outlined in all three application complexities, but all of them faced issues when upgrading to Angular 10. After several hours, we decided that this would not be a feasible way to update the app.

In the end, the only option we had left was to create a blank Angular 16 project and add in all of the project's components from version 5. We knew that both Angular 8 and Angular 16

had similar project structures and identical component structures, so we assumed it would be possible, but the transition would prove to be difficult. In total, there were 40 combined pages, components, and services, and all of the TypeScript needed to be updated to comply with modern TypeScript rules. In addition to resolving the errors, several of the key packages that the project used for log creation and displaying the location were not supported past Angular 9, so those would have to be replaced. We also had to add several polyfills to the project to ensure the project was compatible with modern browsers. In the end, the project was fully functional on Angular 16, besides a few features that would have to be replaced due to the previously mentioned unsupported packages and some minor styling that was not supported in Angular 16.

3.2.3 Reverting to Angular 8

After spending nearly all of winter break trying to upgrade the project, we started C term by continuing development and trying to catch up to our Gantt chart, which we were two weeks behind on. Despite the project running in Angular 16, attempts to redesign the log creation and B-Dex pages revealed several errors that we were unable to resolve, preventing us from making meaningful progress. Although we were able to make progress on the home and account pages, the large majority of our time was spent debugging the TypeScript code on the rest of the website. This caused development on the other pages to come to a complete standstill with no clear path forward. Following a meeting with our advisors, we mutually agreed that the best course of action was to revert to Angular 8 to prioritize UI development rather than debugging the TypeScript code.

3.3 User Experience Design and Prototyping

Prototyping is a crucial first step in the design process. The primary prototyping software used in this project was Figma, a cloud-based design tool that facilitates collaborative interface design, prototyping, and code generation in real-time. As seen in Figure 2 below, Figma allows users to leverage numerous features, including responsive prototyping, automatic layouts, and scalable designing.

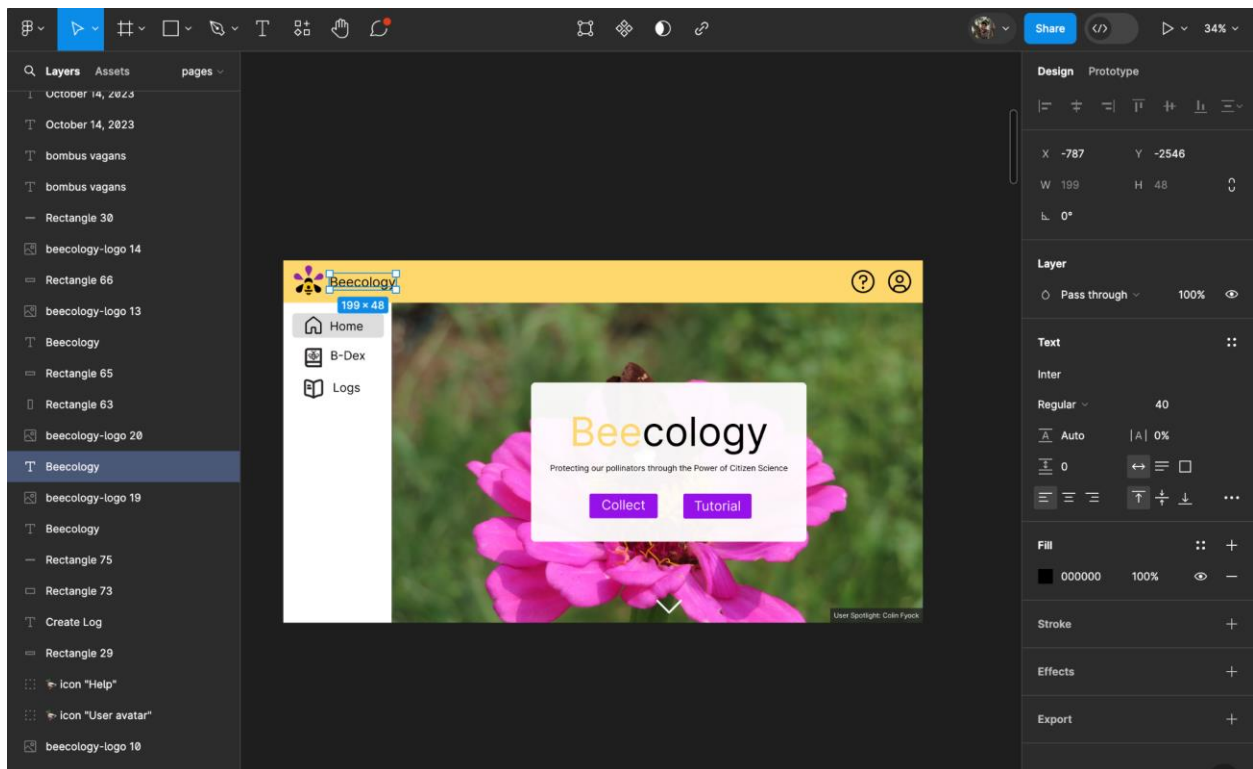


Figure 2. The user interface of Figma and an example of how an early prototype of the Beecology web app title page was created.

The platform also supports real-time collaboration, allowing team members to work simultaneously on the Beecology project together, with instant feedback. This streamlined the prototyping process, allowing rapid iterations and flexible options for user testing, with on-the-fly adjustments. Additionally, Figma allows for both mobile and desktop canvas frames, where prototyping for both platforms can be made and compared before production. Though Figma

served as a solid starting point, the design process continued into code, where continued refinement through Angular and SCSS became the primary process.

3.3.1 Why Do We Prototype?

Prototyping is a crucial phase in the design process, offering a way to test ideas and refine solutions efficiently. It helps identify potential problems early, encourages innovation, and facilitates stakeholder feedback through tangible models (Berglund, 2013). This approach minimizes risks and enhances the likelihood of developing successful products or services that align with user needs. The purposes of prototyping are broad, combining the evaluation of design concepts, improvement of user experience, and attempting to assess user problems that may be crucial to spearheading proper design practice. Additionally, prototypes act as effective communication tools by translating concepts into tangible forms for project participants. Since the goal of our project was to redesign the frontend of the application to make for a better user experience, we decided a rigorous prototyping process was crucial. This allowed us to improve the testing phase of the development process, by performing short tests before development, so that glaring issues could be addressed before additional resources were spent during production.

3.3.2 Figma Prototyping

Figma was an important tool during our design process, and allowed us to intentionally create and refine medium-fidelity prototypes of our application. This step was critical before transitioning into the coding phase, where we could examine and ensure that each of our designs touched on all of our goals before it was sent into production. The collaborative capabilities within Figma allowed us to create a shared storyboard, where every part of the prototype was fully accessible to the team. Each of these panels underwent individual review by our team,

adhering to our established design process protocols. This process consisted of identifying and addressing potential usability issues, enhancing aesthetic appeal, and ensuring consistency across the application. Figma's dynamic prototyping tools helped us simulate user interactions and refine design elements in real time. This review process is what allowed us to transform our initial concepts into user-centered design solutions ready for development.

3.3.3 Mobile and Desktop Layout Improvements

One of the primary goals of our project was to enhance the user experience of the web application. The reason for this decision came down to multiple factors, however, one of the most influential was clear – the interface of version 5 of the web app was not optimized for desktop users as shown in Figure 3.

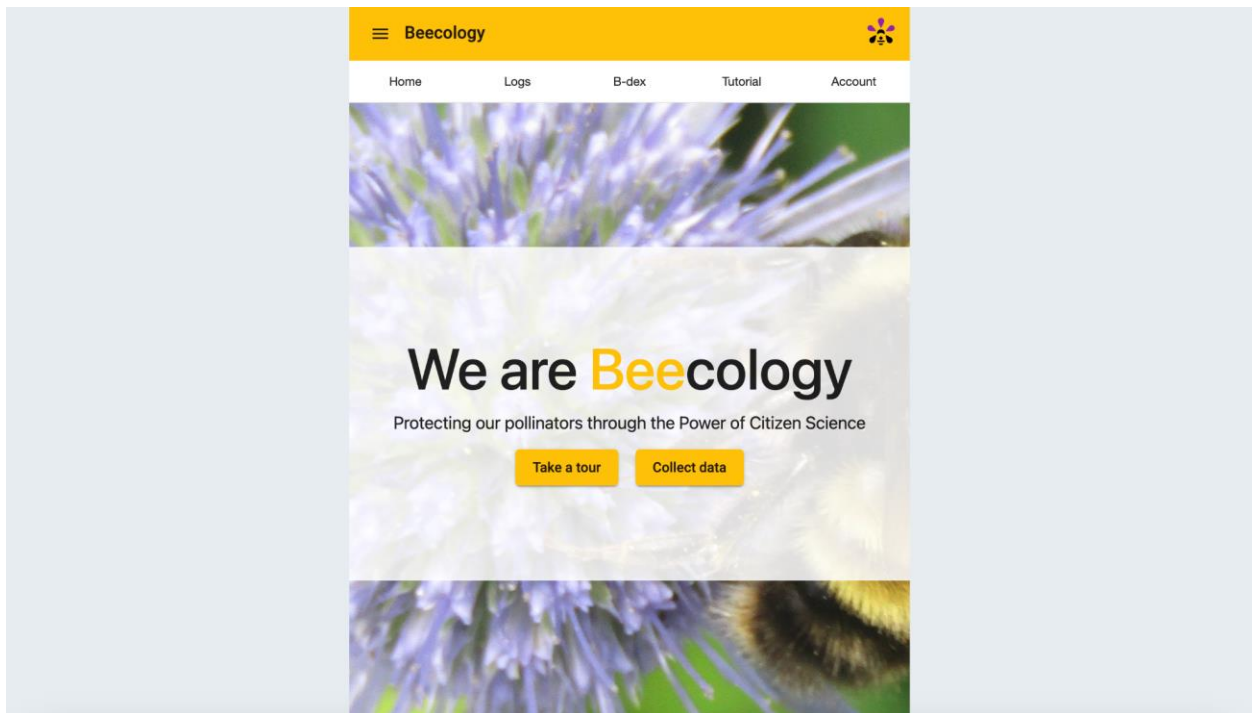


Figure 3. Version 5 of the web app as shown on a wide screen. Note the large margins on both sides of the page.

Professor Gegear, our client, informed us that over the past few years, the user base of the app has shifted from a combination of mobile and desktop to mostly desktop. This gave us a primary goal for the layout of the new prototypes: to expand version 5 of the app into a more functional, landscape presentation to accommodate the increased number of desktop users. An early prototype of this layout can be seen below in Figure 4.

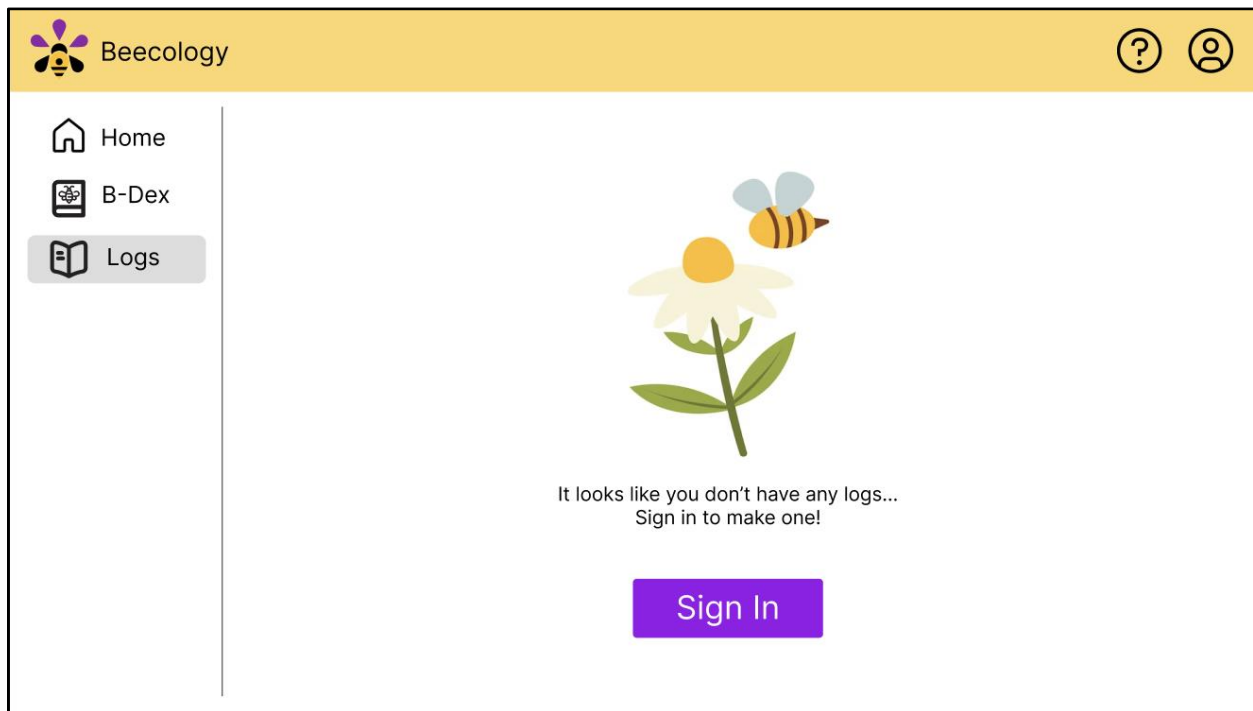


Figure 4. An early prototype of the desktop view of the log view page for version 6.

One of our original prototypes of a landscape layout had a header along the top of the webpage, and a burger menu for settings and more information. Our primary goal with this design was to save screen real estate, allowing for images and text within the main window to be more legible, and take up more space. Though this idea looked promising in early stages, we decided against it, as the navigational features on the page did not seem nearly as legible as other

navigational ideas. From there, we considered other navigational options, including carousels, hamburger menus, and toolbars (Figure 5).

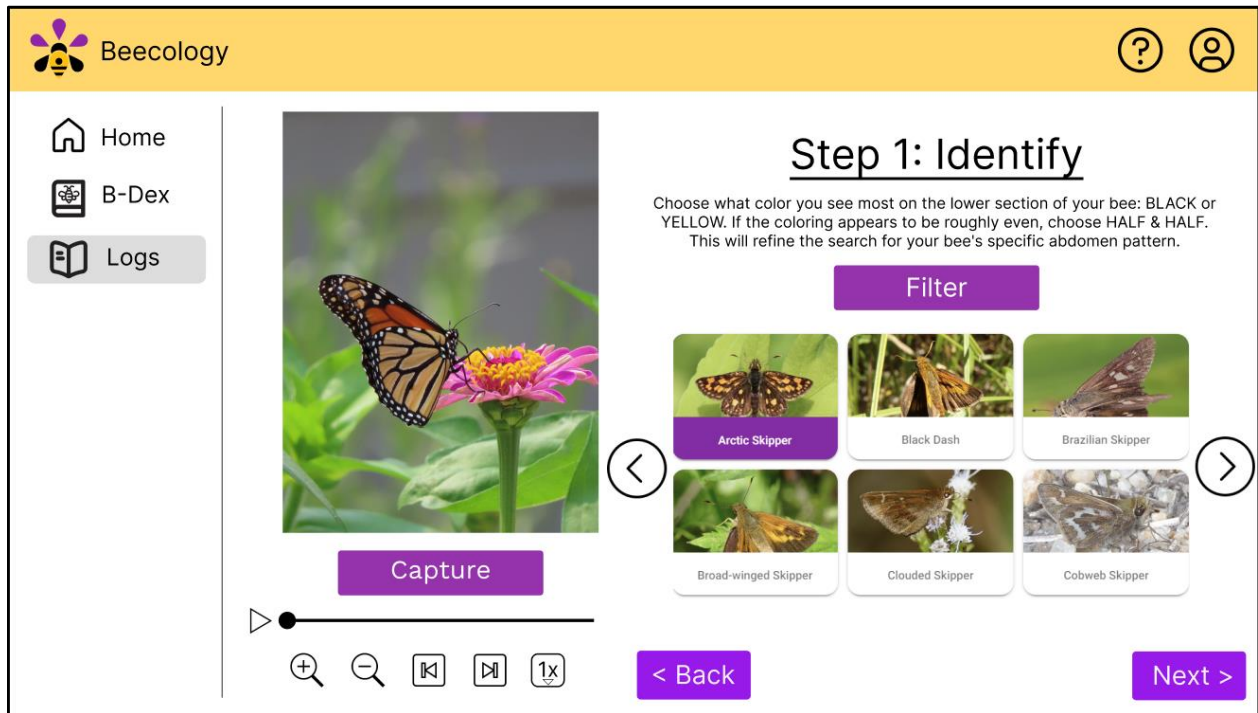


Figure 5. A late prototype of the desktop view of the manual butterfly page for version 6.

In our later stage prototype for app navigation, we settled on a sidebar, which resided on the left side of the screen. Unlike some of the other navigational options, this sidebar stayed permanently on the page, where the user could access navigation with ease at any time. The primary drawback of the sidebar was the width of the screen it occupied on thinner devices, such as tablets and mobile devices. We aimed to fix those issues with a different design for mobile and tablet views (Figure 6).

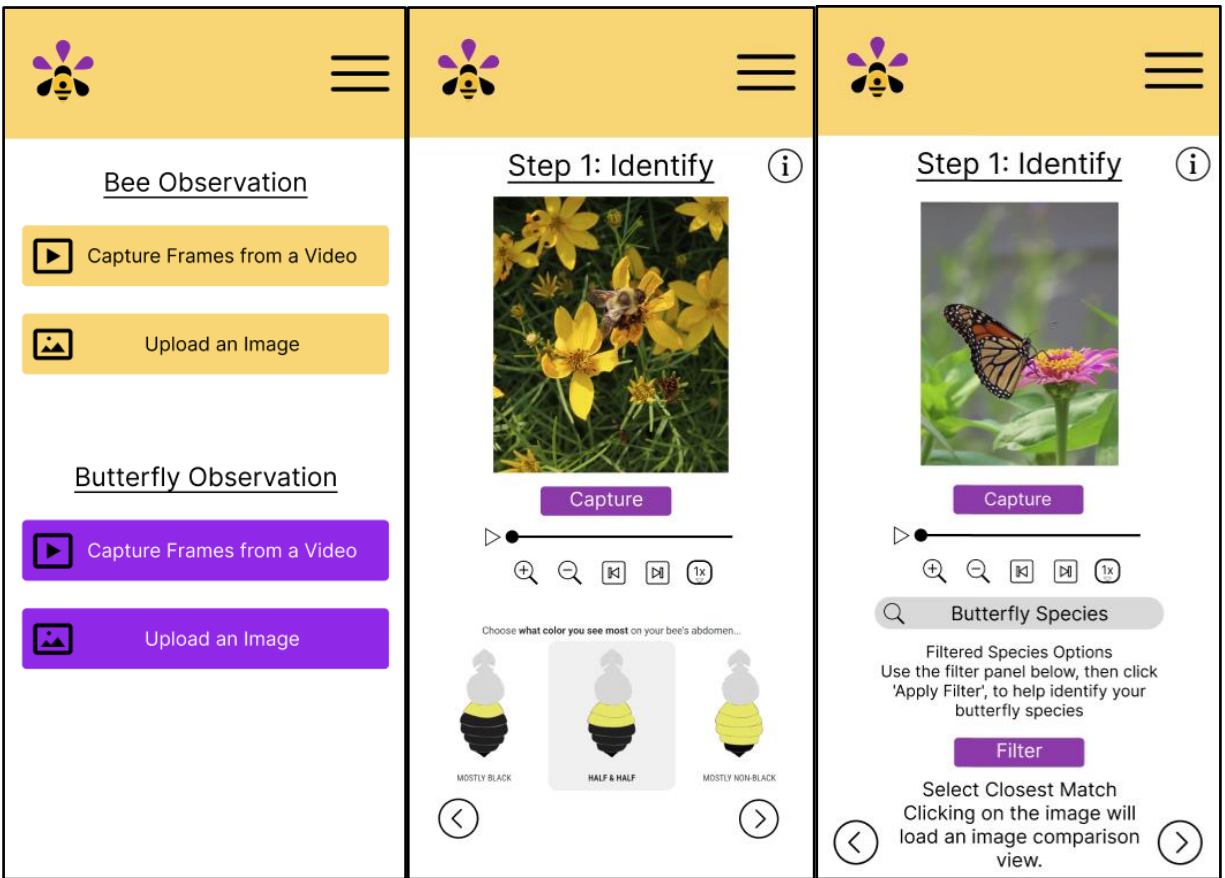


Figure 6. A late prototype of the mobile view of the bee log creation page for version 6.

For these, our initial prototype, which stayed in the final product, was to hide this sidebar within a hamburger menu that only appeared for widths below the tablet threshold. With this addition, we were able to prototype an adaptive navigation design that better utilized screen real estate for various sized displays.

Along with navigation, we also prototyped mobile views of all primary desktop views of pages within the web app as seen in Figure 7. The primary goal with these mobile views was to provide a glance into how we might improve each of the desktop pages for slimmer pages. Though none of these made it directly into production, they did show that each of the pages within the updated frontend could be adapted for mobile screens.

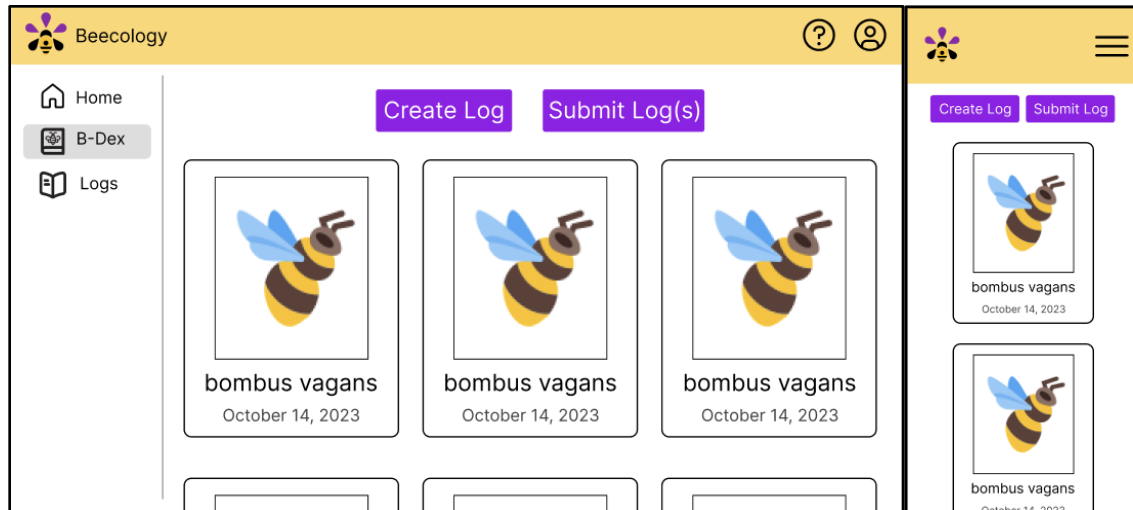


Figure 7. Desktop and mobile prototypes of the B-Dex page for version 6.

3.3.4 Converting From Figma Into Production

Converting these prototyped pages from Figma into production was difficult, as we had to find parallels between our prototype designs, and pre-built graphic designs. Instead of coding all of our elements by hand, we decided that using a CSS framework was a better strategy, as it allowed us to use pre-built graphics that could be adaptive and easily modifiable. Version 5 of the project used Bootstrap as its primary framework, which had confusing documentation and a large learning curve; thus, we decided to shift the project away from Bootstrap. After searching various frontend CSS frameworks, we decided on Bulma, as its design language was very similar to our prototypes. Due to the large number of changes to be made in this project, along with concern for clarity to future developers coding this project, we aimed to create our final designs in production with as few modifications to existing designs in Bulma as possible. Thus, we moved forward with Bulma, removing any references to Bootstrap in the process. Figure 8 shows a comparison between version 5 of the B-Dex, the Figma prototype, and the production version.

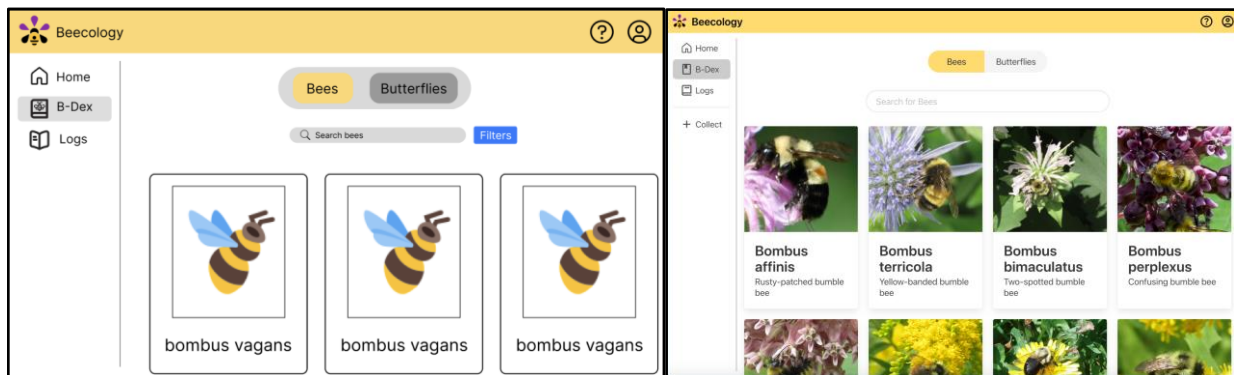
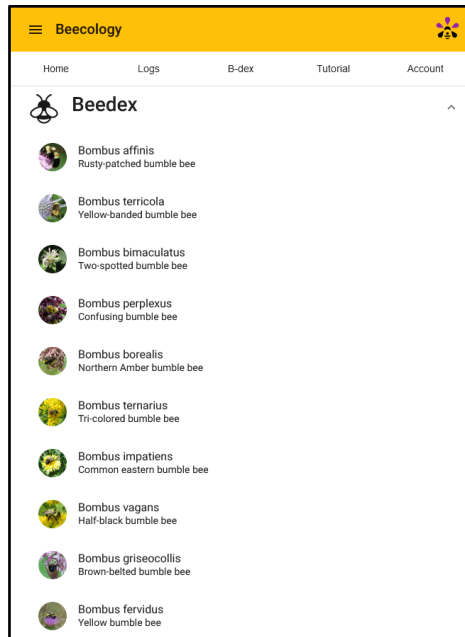


Figure 8. The version 5 B-Dex (top) compared with the version 6 prototype using Figma (left) and its production counterpart using Bulma (right).

The B-Dex page prototyped in Figma looked similar, though not identical, after we created and pushed it to production. In the example in Figure 8, the production view used the Bulma element ‘Card’, which differs in its shape and size from the element we designed in Figma, among other attributes. The primary benefit of using an element from Bulma without heavily modifying it is a significant improvement to production speed. Since Bulma elements are adaptive to nearly every possible screen size (widescreen, tablet, mobile), testing and modifying for different viewports was less necessary. Due to this adaptive nature of Bulma, continuing to

use its elements in as many places as possible allowed us to decrease the necessary work time for conversion from our prototype into production, while still retaining all the features necessary for proper app function.

3.3.5 Revisiting Figma For Iterations

One notable moment in our prototyping process was when working on the log view page. Our original implementation of this page scrolled awkwardly, and the overall experience with editing fields was unintuitive. To fix this, we revisited the Figma file, and came up with a design that met the updated usability goals we realized after initial testing. With the log view page, our primary design critique was the requirement to avoid having to scroll on desktop, which we solved by condensing the tables containing the log information. Other similar tweaks had the same process, where we revisited Figma and adjusted the page design before reimplementing it into production. This was the approach we took to fix issues like home page visibility, page formatting, and mobile views.

3.3.6 Designing for Accessibility in Mind

In refining our application dedicated to tracking bee and butterfly sightings, we prioritized looks and accessibility, aligning our design modifications with the Web Content Accessibility Guidelines (WCAG, 2023). Recognizing the diverse range of users, including those with visual impairments, we implemented several key changes. Firstly, we opted for bolder text across the application, ensuring that information is easily readable for all users, including those without the ability to discern lower contrast. This not only enhanced readability but also contributed to the overall clarity and visual impact of the content. Additionally, we increased the

size of the buttons found on the home and logs pages. Larger buttons not only make the app more navigable for users with motor difficulties but also improve the overall user experience by making interactions more comfortable and error-free. As such, we also simplified the UI in places where there might be an oversaturation of visual elements. Version 5 of the application was in a ratio that forced the cramming of elements, making it difficult for the user to understand what they were looking for. Lastly, the color scheme of our application underwent a slight, yet thoughtful revision. We selected colors that provide sufficient contrast between each other and are easily distinguishable by users with different types of color vision deficiencies, such as Deuteranopia, Tritanopia, and Protanopia. These adjustments, grounded in WCAG principles, not only made our application more accessible but also emphasized our commitment to creating a user-friendly and inclusive interface for all users.

3.4 Implementation of Visual Changes

After storyboarding our redesigned pages and planning out the different visual changes we were going to make, we began the development phase of our project. This included adding and customizing Bulma, creating new Angular components, and restyling most pages on the web app.

3.4.1 Bulma Implementation & CSS Difficulties

When implementing Bulma, we began by styling a few pages using some of Bulma's preset components without any customization to ensure that Bulma matched our vision for the website. Apart from a couple pages or classes that would need tweaking, Bulma accurately matched our design aesthetic we had in mind when prototyping the new UI.

Once we had settled on using Bulma, we attempted to follow the steps on the official Bulma documentation website to install Bulma using node. However, we were met with many errors saying that some of Bulma's dependencies had conflicting versions with packages already in our project despite installing its legacy peer dependencies. Without being able to install Bulma via node, we would not be able to customize any of the preset styles or colors, which would be very important for code readability and maintainability throughout the project. This issue persisted for several weeks before we went through and tested all the different package combinations until we found one that did not conflict, which can be found in the project's package file. After finding a working combination, we were successful in customizing the preset classes and changing Bulma's default theme colors, giving us much more control over how the UI looked.

Another issue that we faced when implementing Bulma included conflicting styles with Bootstrap, another styling framework that a prior team had added to the project. This created inconsistencies with what styles were being applied to different elements of the web app since Bulma and Bootstrap use very similar naming conventions for their CSS classes used to style elements. To solve this problem, we removed all references to Bootstrap in favor of Bulma, as Bootstrap was rarely used throughout the project.

3.4.2 Streamlining the User Experience with a Navigation Redesign

Version 5 of the web application displayed a mobile view regardless of device width. This was helpful when developing it originally, as early versions of the Beecology web app were intended for use on Android phones. However, Professor Gegear informed us that the user base shifted from primarily submitting logs on a mobile phone to submitting logs on a desktop

computer, it made sense to implement a new web application view that better conformed with these changes. Following other cosmetic changes to the web app, we changed the display of the app on desktop by removing the margins on the left and right sides of the page (Figure 9).

The margins are what made version 5 of the desktop page look like a mobile app; changing it from a margin view to a full width view allowed the app to look like a fully functional desktop app. We also changed the previously existing headers to better conform with a desktop view. We did this by implementing a combination of header and sidebar to allow the user to navigate around the app. Since we are using the Angular framework, a framework made by Google, we decided to take after the standard UI design practices that Google follows in several of their websites like YouTube, the Angular documentation, Google Drive, etc. The newly added sidebar is arranged, top to bottom, Home, B-Dex, Logs, and Collect. The natural reading order for English users is top to bottom and left to right, so the most important and most used aspects should be at the top left of the page. The items on the sidebar are placed there because they are the most used aspects of the website; most users will start on the homepage, and either click Collect Data in the middle of the screen or click the easily accessible Collect button on the bottom of the sidebar.

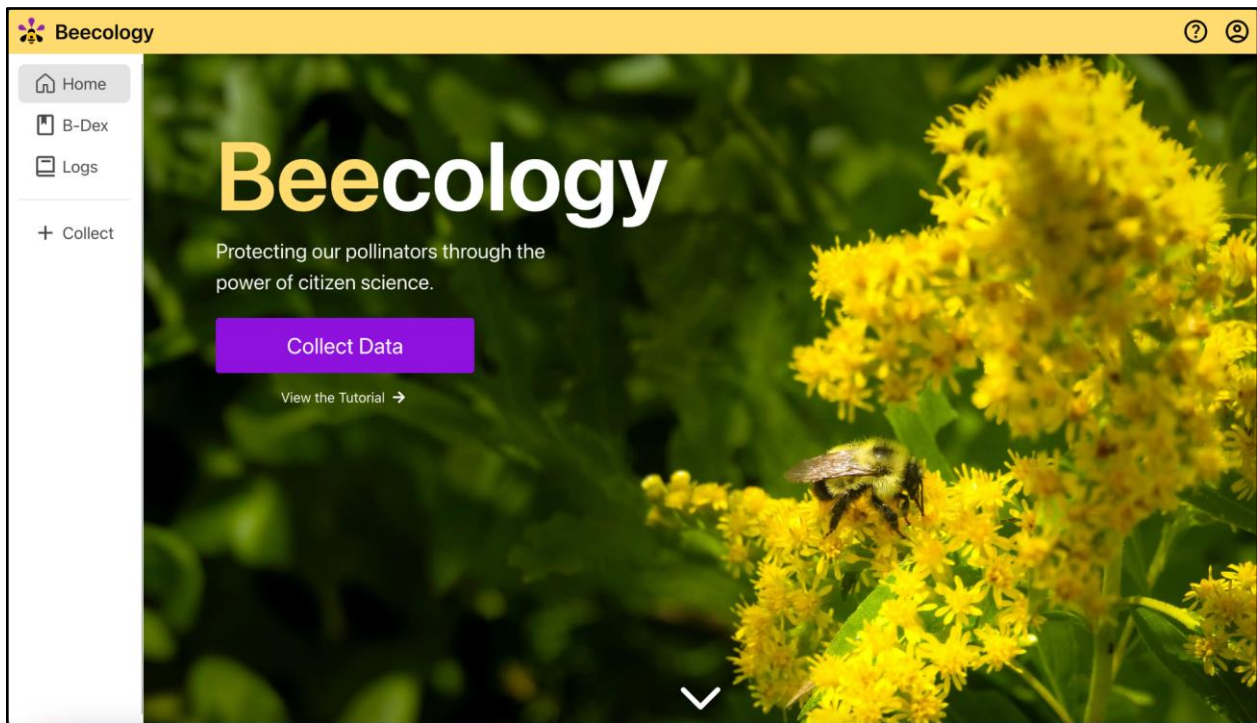
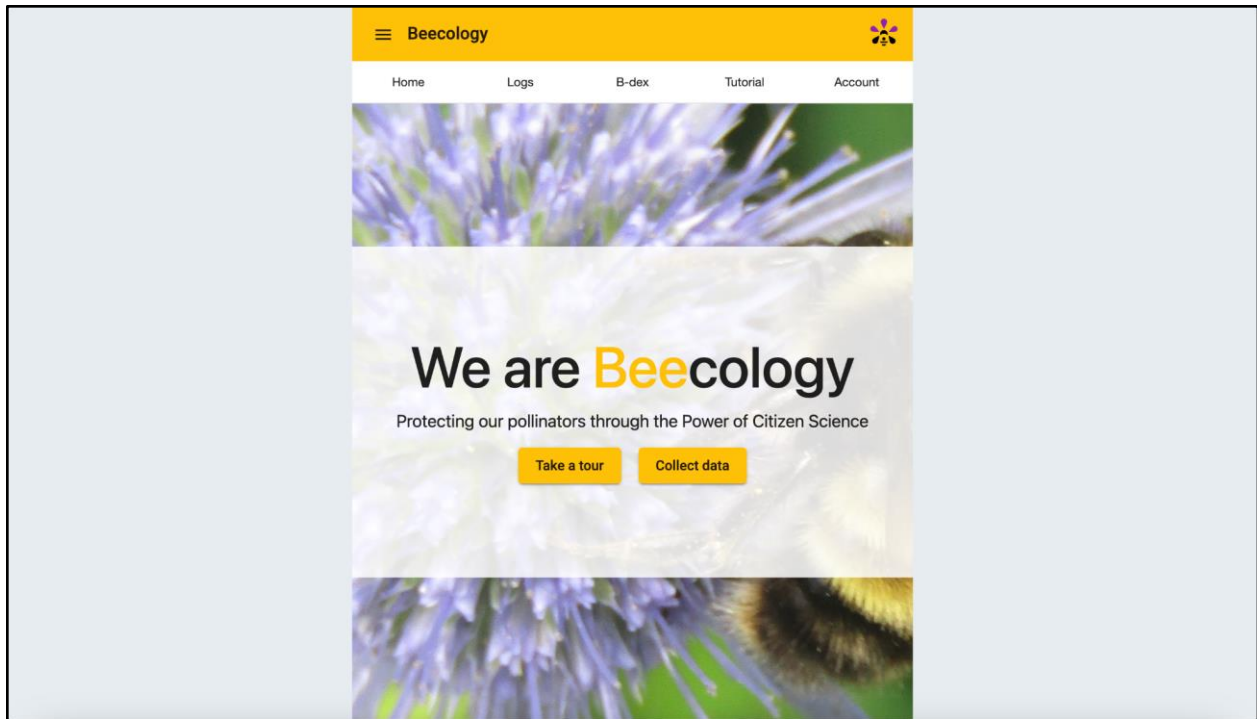


Figure 9. The home screen of the web application for version 5 (above) compared to the production home screen of the web app for version 6 on desktop (below).

The less used buttons, Tutorial and Account (the question mark and person icons seen in the top right of Figure 9 above), were moved to the header as they will not be used as frequently by users. Users only have to sign in once to use the web app and then do not return to the sign in unless they have more than one account. The tutorial page will only be viewed by new users, and can be accessed on the front page of the website if needed. Once the user visits the tutorial, they would understand the web app and would not need to view it again. The home page was also redesigned to show off a larger image taken from the *Bombus vagans* entry in the B-Dex. This could also be changed by later groups to implement a user spotlight, which is further described in 5.3.3 in the Future Work section.

3.4.3 Additions to the B-Dex

Version 5 of the B-Dex page features two drop-down menus for both bees and butterflies (Figure 10). Once the user clicks on either drop-down, it expands to show wide cards for each of the bees and butterflies. As the Beecology web application places a heavy emphasis on images and video frames, the small size of every image on this page makes identifying a bee difficult. Additionally, the drop-down for the bees is above the butterfly menu and clicking on the bees pushes all the butterflies under the selection of bees. With this expanded menu, a user would either have to close the bee menu or scroll down to the butterflies.

The first problem we addressed was the card view. Rather than keeping the full-width cards for each individual bee, we opted for a taller card that features a larger image as the main part of the card with the scientific and common names as captions. To keep the aspect ratio of the images as close as possible to their original ratios, we placed every card in rows of four.

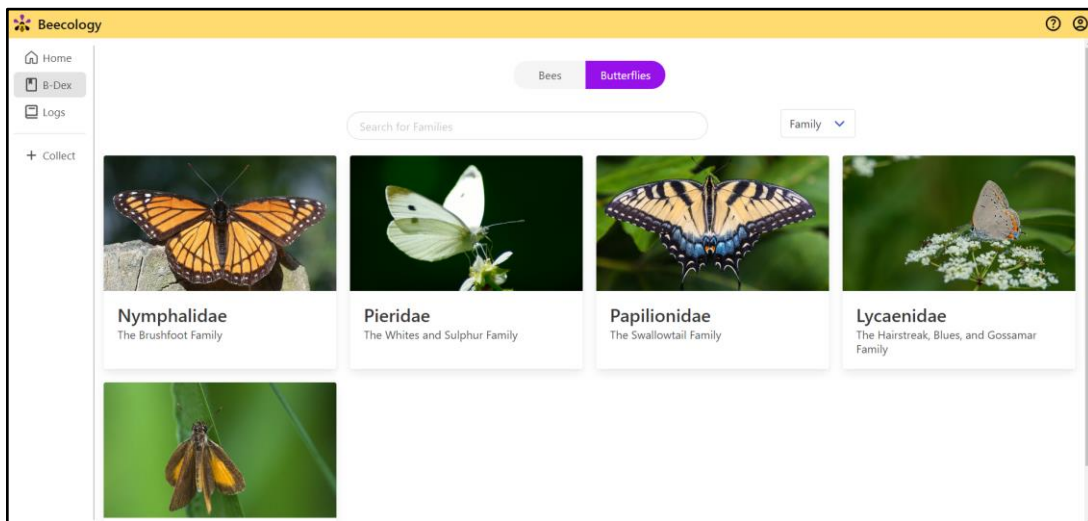
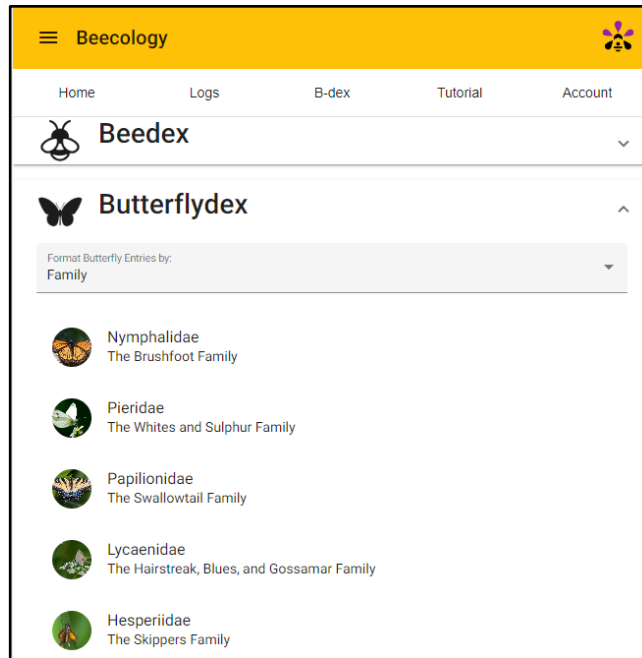


Figure 10. B-Dex Butterfly View Comparison. Version 5 of the butterfly B-Dex (top) compared with version 6 of the butterfly B-Dex (bottom). Note that version 5 uses drop down menus, while version 6 uses buttons to display the information. Version 6 utilizes the extra space to display much larger photos.

The number of cards per row also decreases as the window width decreases. This allows the image to be the primary focus of each B-Dex entry, while keeping the same information from version 5. Enlarging the image in the B-Dex view allows users to more easily distinguish

between families and species without having to memorize names or click on an entry to clearly see the image.

The change in card view did not address the problem of the bee drop-down pushing the butterfly entries down. To solve this, we replaced the clickable drop-down buttons with a set of toggleable buttons at the top of the screen (Figure 11). By clicking either of the options, a user can switch from displaying bees to displaying butterflies and vice-versa.



Figure 11. The bee/butterfly toggle that is used on the B-Dex page in version 6.

In addition to the visual changes we made while retaining the old functionality of the B-Dex, we also added a new feature, the search bar, as shown in Figure 12. By filtering the entire list of entries against the user's inputted "search", the list of shown entries updates and refreshes the Angular component to display the entries that fit the search. The search input compares the input to the entire string containing a bee or butterflies scientific name, so searching for any string of letters will display entries with that input anywhere in the name. For example, searching "affinis" will display the *Bombus affinis* bee species. The search is also stored in each page's respective search field, meaning switching from butterfly families to species will keep the last search before switching entry views.

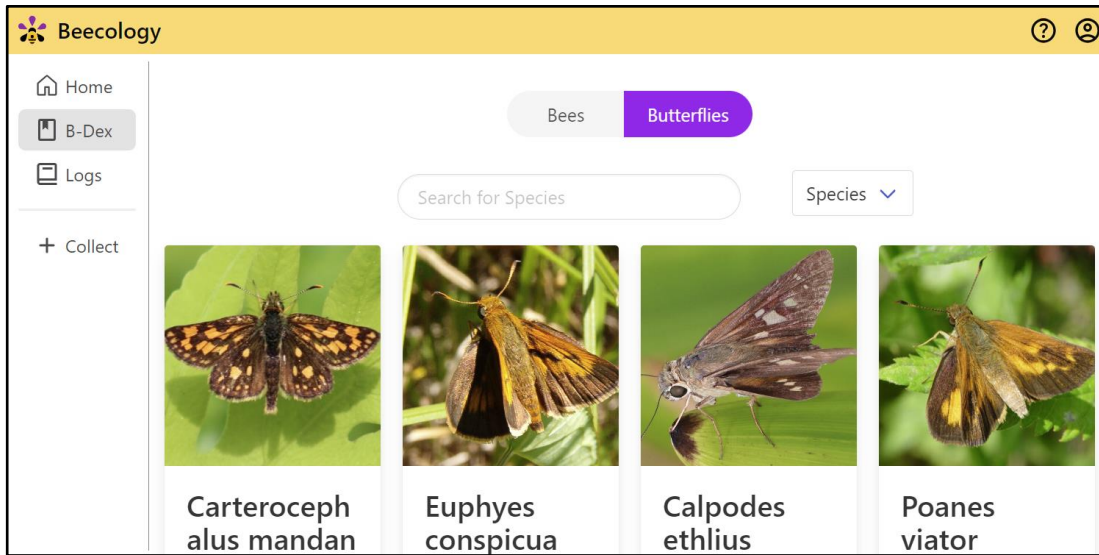


Figure 12. *The new search feature that was added to the B-Dex page in version 6. It also contains a family/species dropdown as an additional sorting option next to the search bar.*

3.4.4 Redesigning the Log View Page

The 'Log View' page allows the user to review their previously submitted logs, along with adding previously unknown information on the logs, such as location, or flower (Figure 13). One of the minor improvements to this page was an API call that allowed us to retrieve the common name of the bee species, allowing us to display that within the log details. Additionally, we made small changes to the layout of the page, condensing the information to be viewable at all times, without needing to scroll.

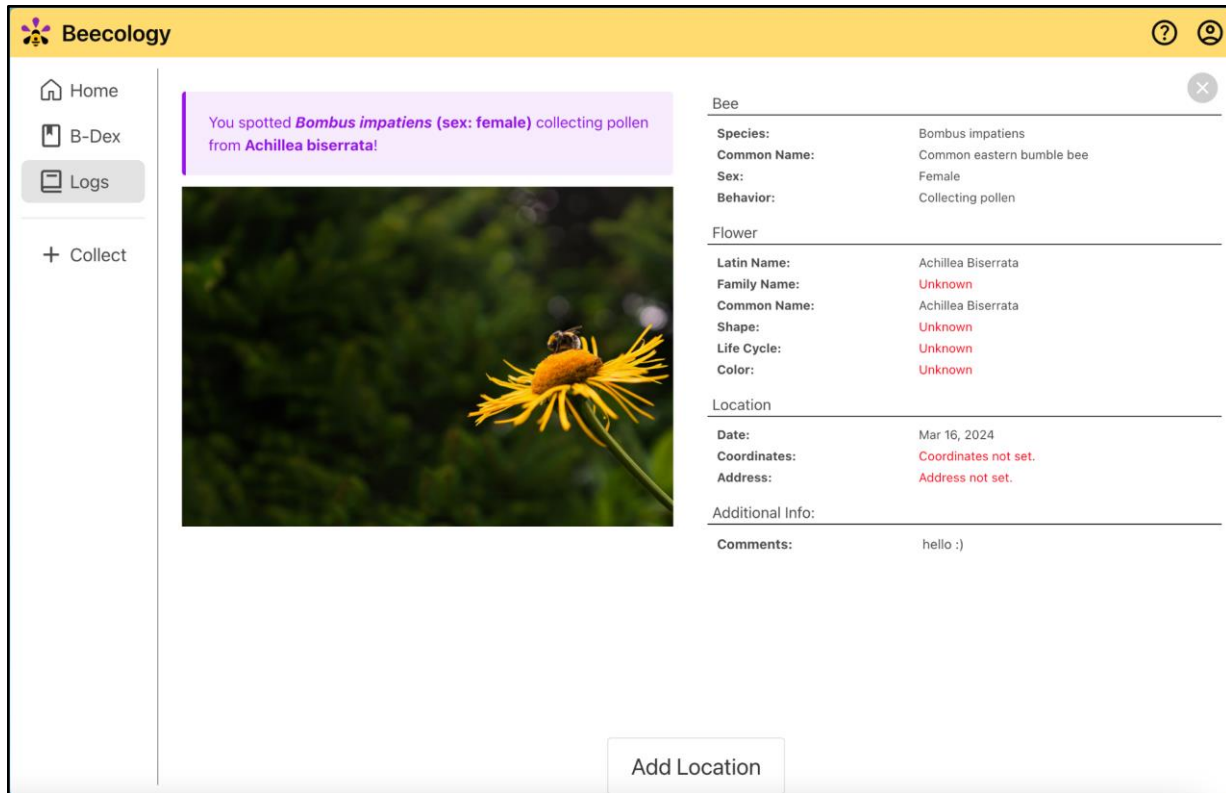


Figure 13. The production bee log view page in version 6. This page displays an overview of the information a user has collected before they decide to submit their log.

3.4.5 Redesigning the Log Creation Pages

Log creation is the core feature that allows data collection in the Beecology web app, so we had to be very intentional with our redesign of this section. When creating a log, a user must upload an image or video of a bee or butterfly, identify the species and its behavior, identify the flower, enter the location of the sighting, and optionally leave a comment. Our new design had to keep all of the functionality and simplicity of version 5 while laying the information out in a way that made good use of the extra space on the desktop view.

3.4.5.1 Redesigning the Primary Log Creation View

In version 5 of the log creation view, the image or video was displayed at the top of the page with any options or buttons directly below it. This worked well for most log creation pages, but for pages with many options, users would have to scroll down to see them all, causing the image or video to go off-screen. This was a major issue, since the user would then have to scroll down to read an option, then scroll back up to check the image and repeat this process for each available option. To resolve this issue, we added a fixed pane on the left side that is always visible. All options are then displayed on the right side, which scroll vertically for pages where there is too much content to display on the screen at once. We also made the image or video take up roughly half of the available width, since this will allow the user to more easily observe the different parts of their bee or butterfly, making the identification steps of log creation easier. Furthermore, we added a tall divider down the center to clearly distinguish between the different sections of each page. Each of these changes are displayed below in Figure 14.

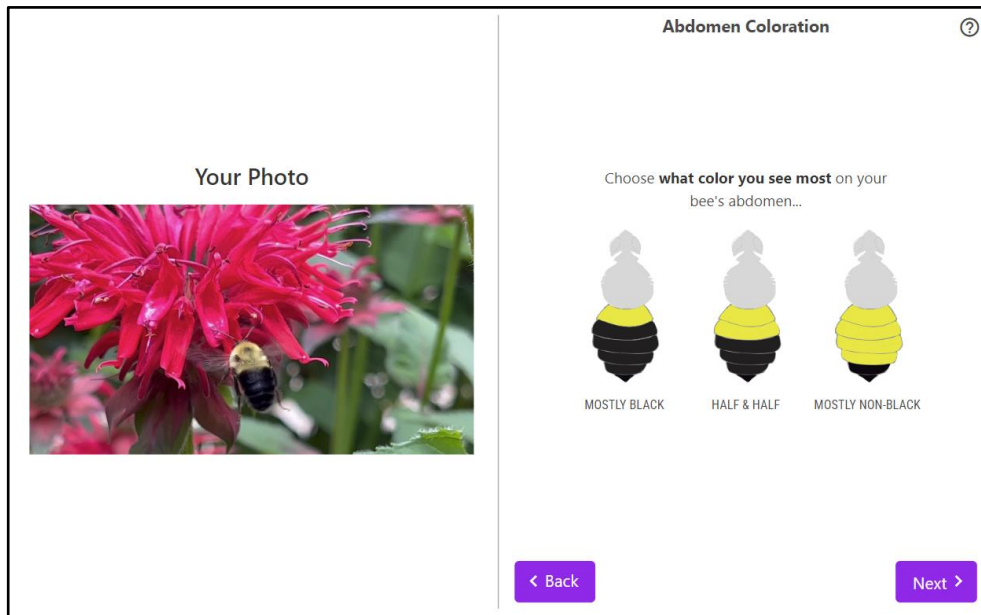
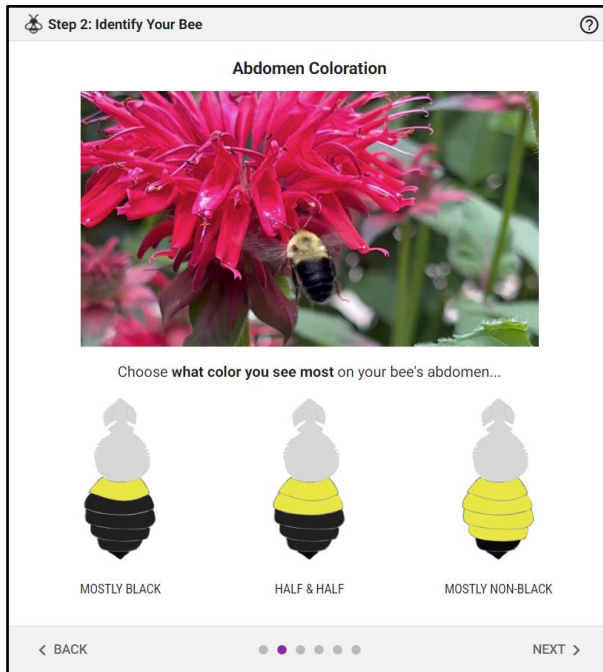


Figure 14. The primary log creation view in version 5 (top) compared to version 6 (bottom). This is an example of the log creation process for a bee image.

3.4.5.2 Redesigning the Bee Identification Page

As seen in Figure 15, in version 5 of the bee identification page, the user's bee image is displayed as the last element on the page with the "B-Dex Entry" section empty. In the code of

this page, there was a commented out Angular component that was supposed to display the image, but it was not working due to an issue with retrieving images from the database. In our redesign of this page, we kept the side-by-side layout to make better use of the extra space on wider devices, and re-added the image to the “B-Dex Entry” section so that the user can quickly compare their image or video with an example image of their bee species to make sure what they entered was correct. Increasing the size of the user’s image was especially important for this page because it is crucial that data being submitted to the database is accurate. Making the user’s image on this page larger and adding back the example B-Dex image for reference helps the user to double-check that their identification is correct before moving forward.

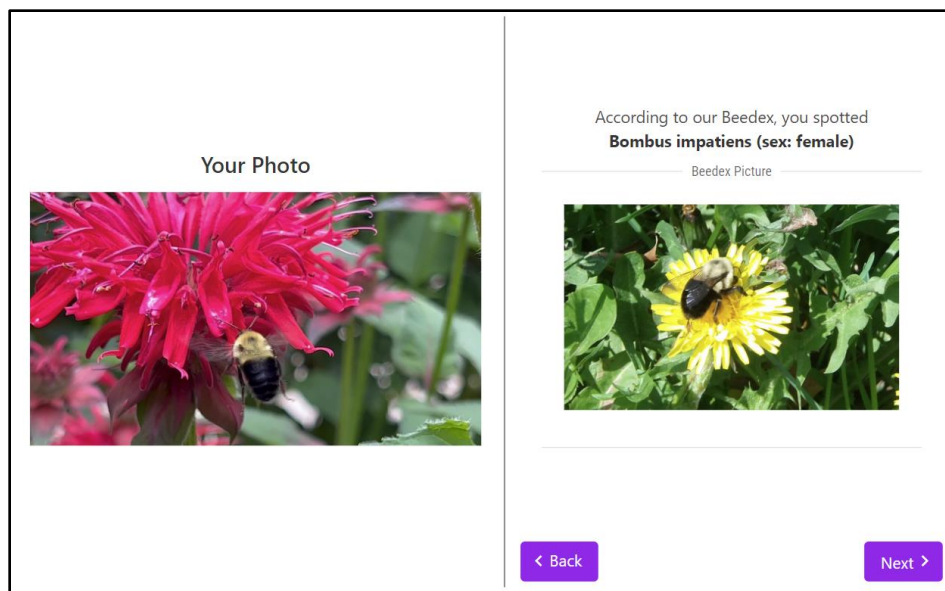
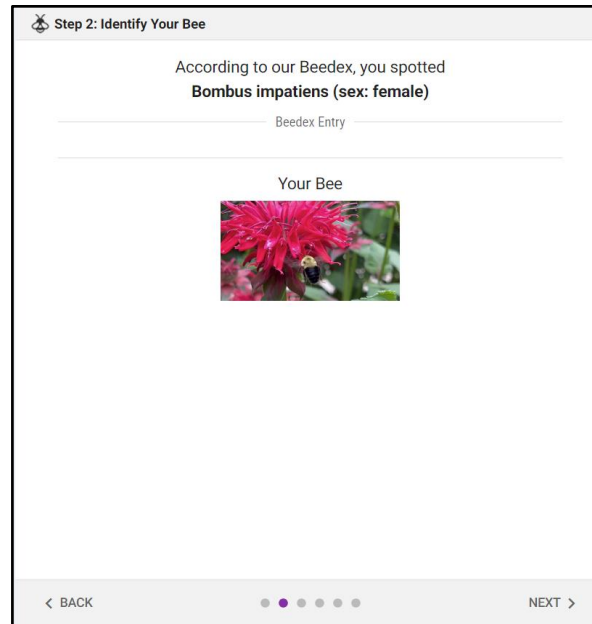


Figure 15. The bee identification page in version 5 (top) compared against the page in version 6 (bottom). Note that version 5 lacked a reference image from the B-Dex to help the user confirm their identification.

3.4.5.3 Redesigning the iNaturalist Flower Page

As seen in Figure 16, in version 5 of this page, the iNaturalist recommendation buttons are not visible when first opening the page due to the vertical video taking up the entire height of

the window. Furthermore, flower recommendations are displayed in rows of three, continuing to expand the page downward as more recommendations are given, exacerbating the space issue further. In the redesign of this page, the old cards were replaced with Bulma cards, and the flower images, which were previously off-center, were centered. We also decided to list all of the flower recommendations in a window that scrolled horizontally, which solved the issue of having additional recommendations extend the pane. Lastly, we moved the “Unknown” option to display after all of the iNaturalist recommendations, per our advisors request.

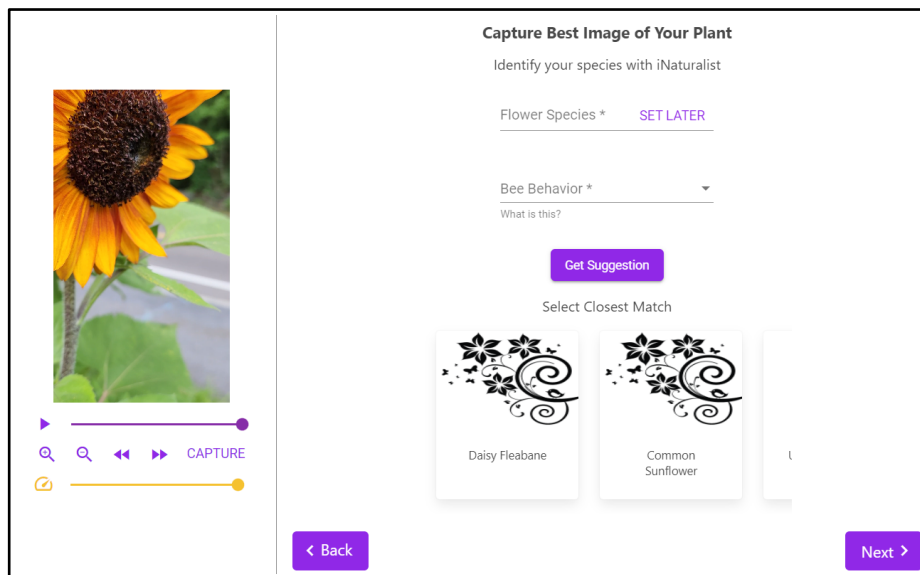
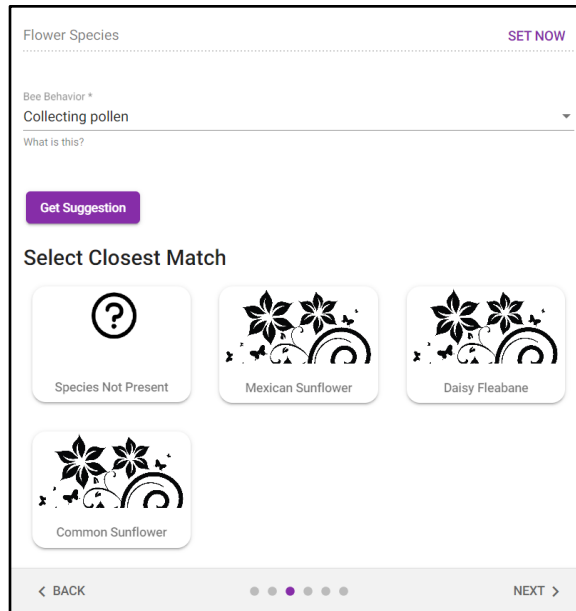


Figure 16. iNaturalist flower identification in version 5 (top) compared against the same page in version 6 (bottom). This page allows the user to select a video frame of a flower and get a suggestion of what the flower is from iNaturalist.

3.4.5.4 Redesigning the Manual Butterfly View

In version 5 of this page, the filter options dropdown menu and filtered butterflies are not visible when first opening the page because the height of the user-uploaded media takes up the entire screen. Moreover, there is a larger card beneath the six smaller cards. We also replaced the old Angular cards with responsive Bulma cards, as shown in Figure 17.

To address the issue of the height of the page being too tall, forcing the user to repeatedly scroll up and down, version 5 of this page would display the current captured frame in the top corner. This blocked off roughly one quarter of the display and prevented the user from clicking any buttons underneath it until it was dragged out of the way, as shown in Figure 18. Version 6 of this page addresses this issue by keeping the video visible at all times on the left, preventing the need to block off part of the filter options, as seen in Figure 18. However, due to the width restrictions on mobile devices, we kept the vertical layout from version 5 for mobile interfaces in version 6.

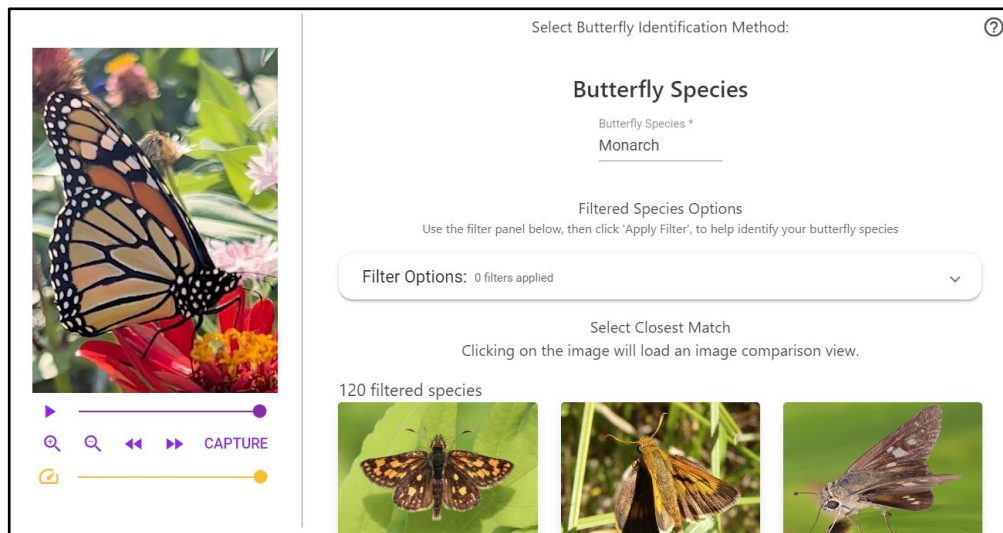
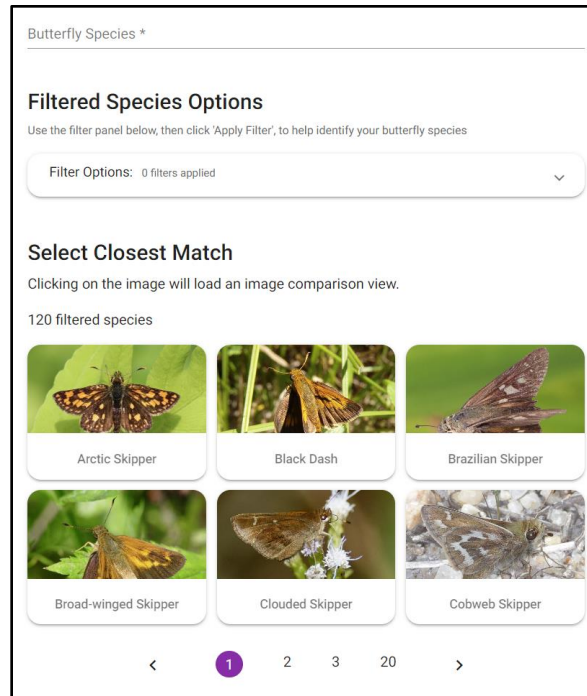


Figure 17. The butterfly manual identification page in version 5 (top) compared against the page in version 6 (bottom). Note the split view that allows the user to view their video while identifying the butterfly species.

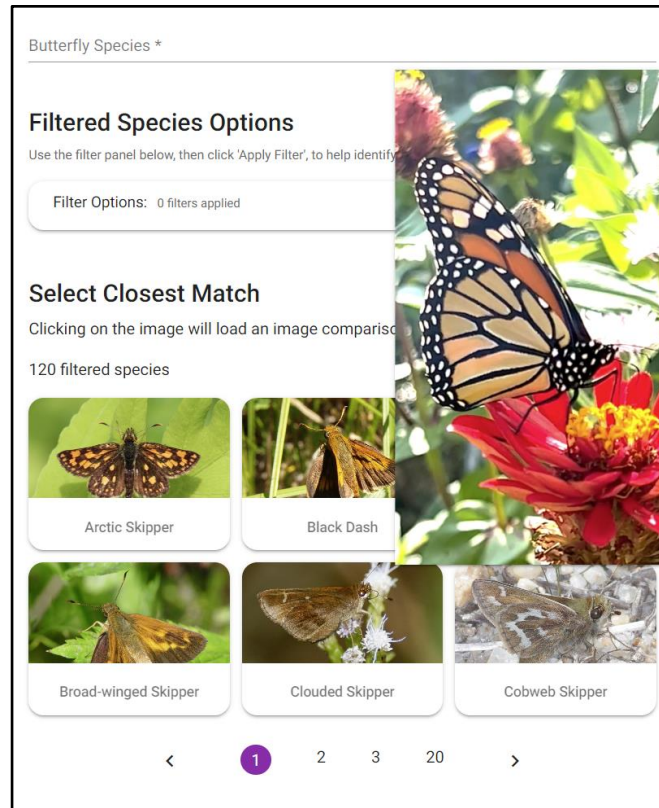


Figure 18. Example of the popup of the selected video frame on version 5 of the web app. Note that the popup covers a significant portion of the screen, preventing the user from accessing any options underneath until the user scrolls or the popup is moved.

3.4.6 Redesigning the Accounts Tab

The core issue with the existing accounts page in the web application was that it did not provide much information to the user. Version 5 of the design simply told the user who they were signed in as, a sign out button, and a Beecology disclaimer at the bottom. In addition to this, more information about the user could be accessed in the sidebar hamburger menu at the top left corner of the page. This menu showed the user all the same information as the accounts page, but additionally showed them the email they used to sign in along with their profile picture if they were signed in using Google. We consolidated these two separate pages so that all the information is now in the accounts page, which now displays the user's profile picture, their

email, their name, a disclaimer, and a logout button at the bottom. Additional space was intentionally left between the disclaimer and the logout button in case future groups want to implement a level or medal system as discussed in 5.3.3. The pages leading up to the accounts page, like the login, sign up, and forgot password pages were also adjusted. We used Bulma's buttons and input fields to better match the rest of the site. All of these changes can be seen in Figure 19.

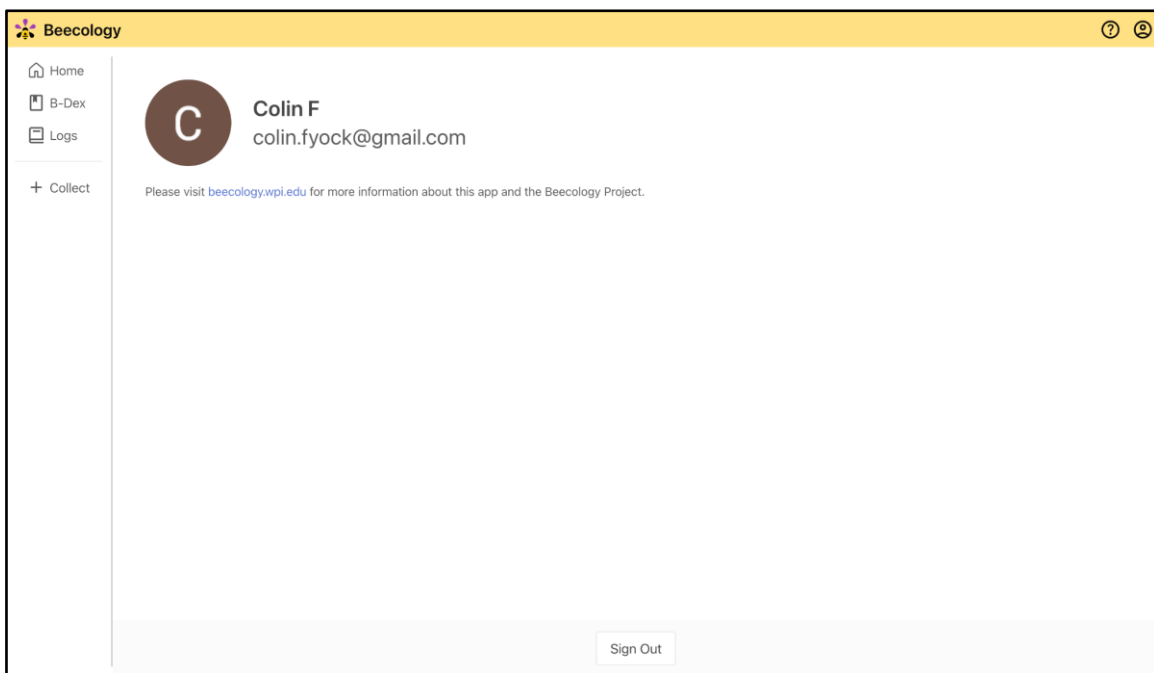
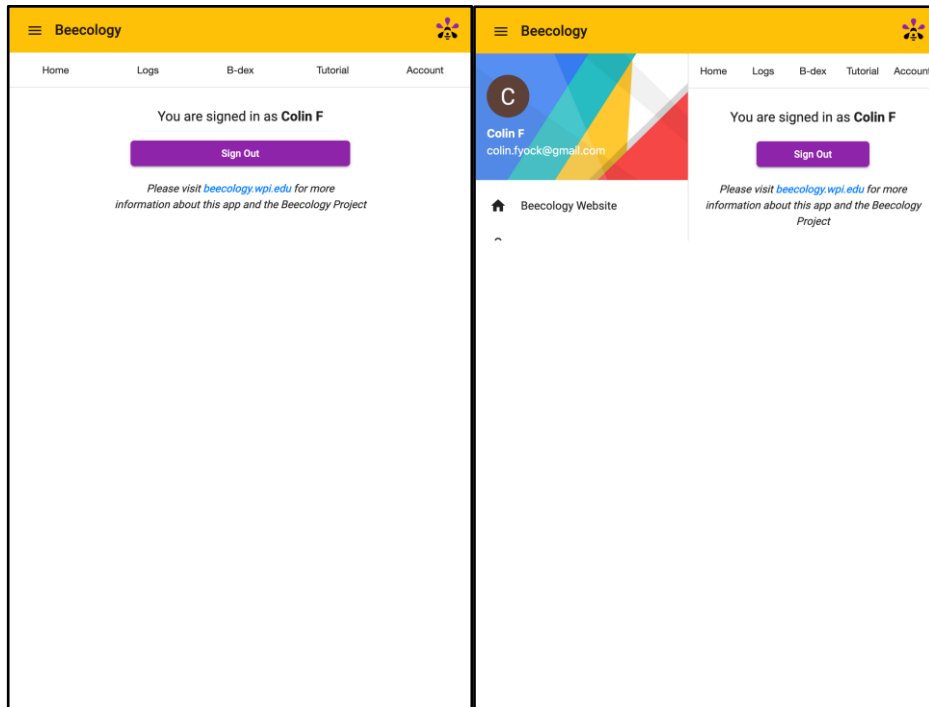


Figure 19. The account page in version 5 (top) compared against the account page in version 6 (bottom). Note that the user information in version 5 of the app is split between the sidebar and account page, which is now consolidated in the account page in version 6.

3.4.7 Redesigning the Tutorial

In version 5 of the web application, the tutorial page broke into various sections that could be expanded by the user to reveal the information in each section. This could sometimes be cumbersome for the user, as to get to certain sections they would have to expand multiple tabs to see the information they were looking for. A good example of this is wanting to know how to identify a butterfly. To get to this information, the user would expand the ‘Step 2: Identify Your Bee or Butterfly’ tab, then expand the ‘Identify a Butterfly’ tab, then expand the ‘Automatic Butterfly Identification’ tab. We aimed to solve this problem, and to allow the user to get to any part in the tutorial within two taps on mobile, or one click on desktop. On mobile, we utilized a table of contents that the user can tap to go to any section they see; if the user would like to search through the tutorial without the table of contents, they can just scroll down. Additionally, on mobile, once the user has scrolled past the table of contents, a small purple arrow button appears at the bottom right hand corner that allows the user to scroll back up to the table of contents to get to a new section. This prevents the issue of expanding multiple dropdowns by allowing the user to navigate to the table of contents whenever they need to, and then navigating to any section they desire. The desktop table of contents is even easier to use, as it is always locked to the right side of the screen even while scrolling. This allows the user to skip to any section they like with a single click. We also implemented images and gifs for the tutorial page that reflect the new design of the website. Each of the aforementioned changes are displayed below in Figures 20 and 21.

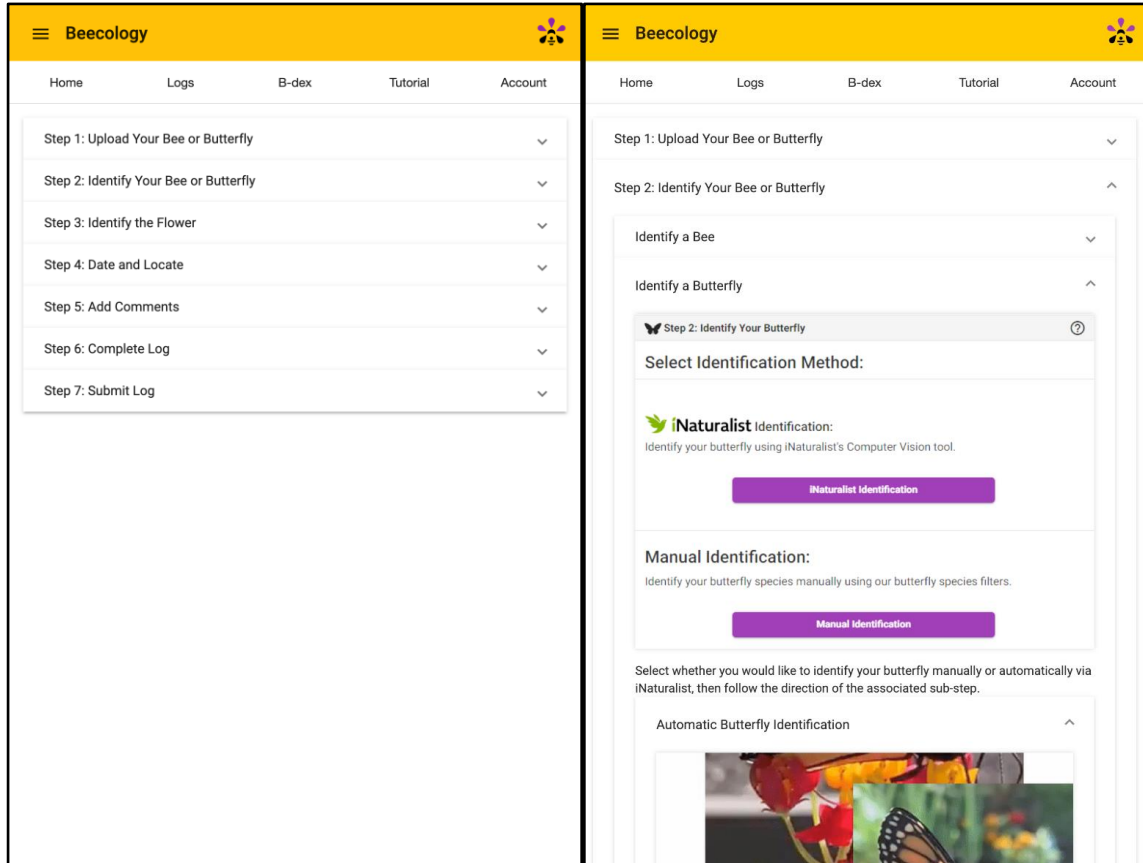


Figure 20. The tutorial page in version 5 of the web application. Note that it would take the user multiple clicks to narrow down a section of the tutorial.

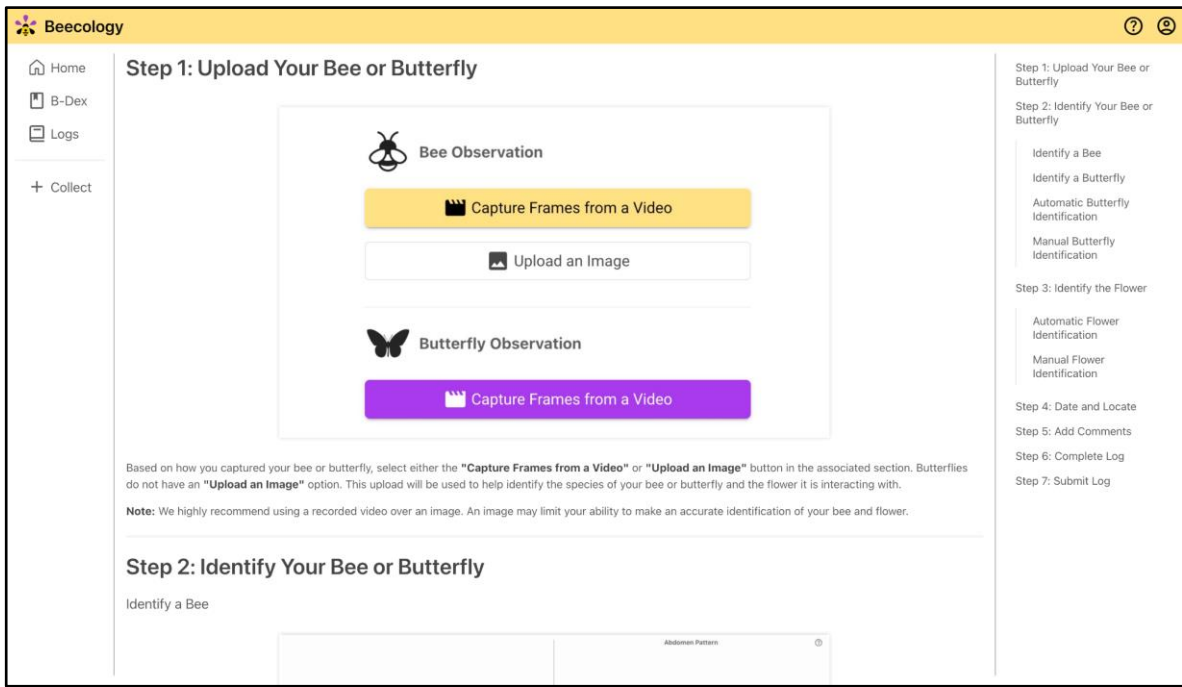
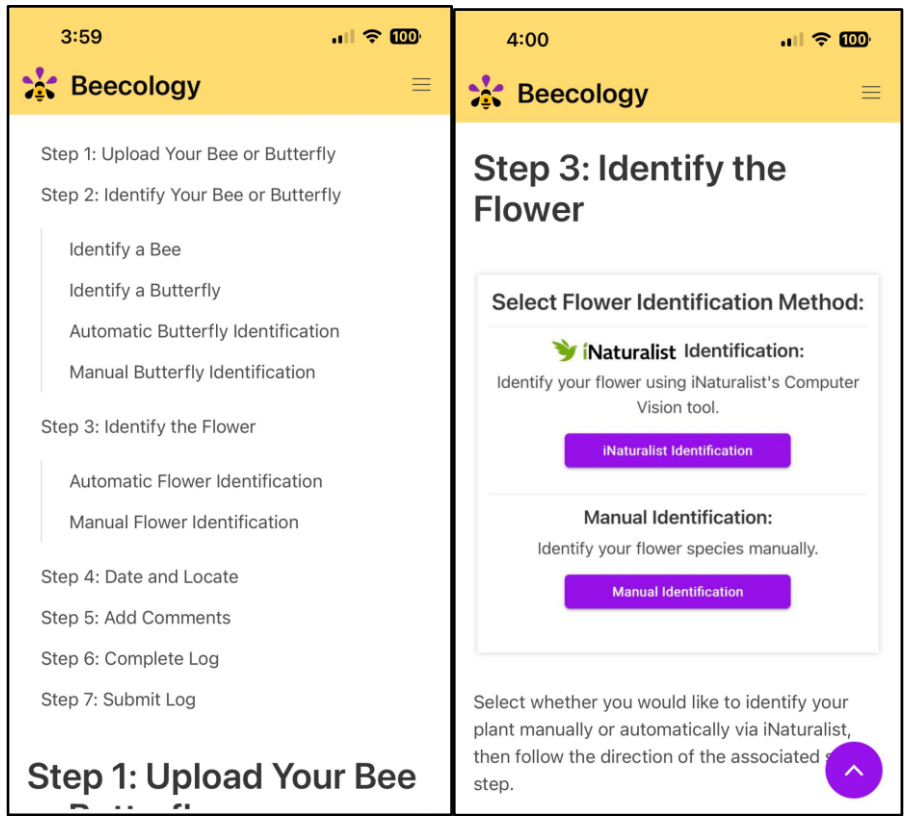


Figure 21. The version 6 tutorial page on mobile (top) and desktop (bottom). Note that navigating both the mobile and desktop pages require fewer clicks than in version 5.

3.4.8 Documentation Additions

Prior additions to the web application lacked significant documentation or direction on how features were implemented or utilized. The core problems of the documentation in the existing project were the documentation surrounding the database, the wiki, and the source code. The database was mentioned in the wiki in the Beecology-Docs repository, but all information was shallow. Accessing the database was also mentioned in the wiki. It brought the user through steps up to accessing the PostgreSQL database, but did not tell them how to look through the databases in the server or look at any tables or values within those tables. The wiki also only included each of the databases in the server, but none of the tables, rows, or columns that made up each of the databases, leading to users having to go through each database to find what they are looking for. In addition, the location where user-submitted media, like a bee or butterfly video, is stored is never mentioned in the documentation. Along with previous issues with the wiki, the overall organization of the wiki lacked cohesive flow between different pages. Standard wikis are set up to flow between each aspect of setting up a new development environment. The wiki pages did not flow like this, but instead were a wide array of pages that described each aspect of the Beecology project without much additional information or natural progression between them. This made setting up a new development environment and general navigation of the wiki very difficult. In addition, the source code of the web application contained large swaths of code without any comments or documentation. Nearly every TypeScript file in the web application contained several functions, sometimes named in an unhelpful fashion, that had no documentation on what they did. This leads to confusion when a new team enters the project, as they cannot decipher what each function is used for or where each function is utilized in the

frontend. The comments that did exist did not contribute to the understanding of the functions, but were used by previous teams to discuss future or unrelated implementations.

When redesigning the UI and web application, we greatly improved documentation surrounding the database, the organization of the wiki, and the source code of the files. To start, we rearranged the order of the wiki to flow between the different aspects of setting up the different pages in the Beecology Project. We also created new pages for the wiki that better explain various portions of the Beecology project and its codebase. These new pages were ‘Web Links,’ ‘Getting Started,’ ‘Beecology Setup Next Steps,’ ‘Bio-CS Bridge Setup Next Steps,’ ‘Before Coding,’ ‘Tutorials,’ ‘Databases: Detail,’ ‘Repository Explanations,’ ‘Bee ID Repo,’ and ‘Git/Pipeline Info.’

To keep a logical flow between wiki pages, the first page we created was ‘Web Links,’ which is a page dedicated to housing several Beecology-related links, such as the repositories for the major projects as well as their production and development website links. By viewing this wiki first, project viewers can get acquainted with the projects they will be working on through a simple project hub that includes descriptions of the respective projects, which is an important step to take before getting started with the Beecology projects.

After gaining a basic understanding of the projects, the next step would be to set up the respective project environment. To help with this project, the new ‘Getting Started’ page includes various tutorials on downloading required dependencies, as well as setting up the project in a development environment. There are different steps for concluding this process between the Beecology and the Bio-CS Bridge projects, so two separate pages were created with the project-specific steps.

Prior to the wik update, tools and resources for learning the relevant code frameworks Beecology uses were scattered throughout various project folders. Consequently, we compiled a list of resources on the wiki that are best viewed before beginning any coding. To start, almost every major project is coded using the JavaScript library, Angular, so the wiki page ‘Tutorials’ compiles a list of useful hyperlinks to the Angular documentation, all of which are key functionalities in the Beecology projects. Furthermore, version 6 of the Bee-ID web application uses the CSS framework, Bulma, and the Data Visualization application uses the JavaScript visualization library, D3.js, so links to their respective documentations are provided as well. Under the wiki page, ‘APIs’, we provide links to the Beecology APIs as well as basic information regarding the APIs that one might need to understand the backend of Beecology.

The previous wiki included a top-level summary of the databases of Beecology. However, this basic information was either not complete or not helpful in understanding the databases, so we made some additions and updates. The ‘Databases: Overview’ page provides an overall summary of the different databases and users. Most of the information in this page prior was complete, so we left it untouched. ‘The Databases: Accessing’ page was edited to include additional information on accessing the database via SSH on the Beecology server. The original steps stopped at the point of accessing PostgreSQL, but did not go any further. The new steps include how to connect to the various databases available along with how to index through those databases and see the entries within them. ‘The Databases: Detail’ page was added and contains information on the beecology2 database, which is the currently used database in the frontend web application. Prior to this update, there was no information in the wiki on the various tables in the beecology2 database. Previously, the only way to get this information would be to SSH into the server and look through each table manually.

Other than a basic explanation of each repository, the previous wiki did not go in-depth with the code within the different repositories. As reading through every component in an Angular project can be time-consuming and confusing, the addition of the ‘Repository Explanations’ page aims to provide beginner-level information about the parts within each repository. As of March 2024, the only repository added to this page goes over the Bee-ID 23-24 repository. This page includes a summary of every Angular component, page, pipeline, and service found in the code. With the understanding of Angular gained from the tutorials, a future developer would use this page to get an idea of the different components in this repository before and during their coding process.

The wiki also includes other pages with explanations that might be helpful to Beecology users, but not inherently required. This includes the ‘Git/Pipeline Info’ page, a detailed explanation of how GitLab pipelines and runners work and how GitLab creates and updates the public websites after pushing to specific repositories, as well as the ‘Servers’ pages found in the last wiki. The ‘Servers’ pages include general server, firewall, and SSH information, with the ‘SSH’ page including a tutorial on using SSH to connect to the Beecology servers.

4. Methods and Results: Data Visualization

From the data visualization aspect, one of the major goals of this MQP was to extend the current web application so both bumblebee and butterfly enthusiasts can explore up-to-date data collected by others across New England. Beecology also includes information about pollinator behavior and pollinator-plant interactions. With the new visualizations in place, users can monitor and study observed butterfly populations. Users can analyze the observations based on

families and subfamilies of butterflies, not just species. The visualizations displayed were all chosen to fulfill different research needs for Professor Gegear and other biology researchers.

To expand upon the web application, and add new pages, our team used Angular, which is a JavaScript web application framework made by Google, as well as the Angular Material UI libraries for creating uniform components on the webpage. The Angular framework is what the entire web application uses, and it pulls all the parts of the application together into one cohesive app. We also used TypeScript (developed by Microsoft), which extends JavaScript by adding static typing with type annotations .

To produce each visualization we used a combination of basic JavaScript and TypeScript, as well as the D3.js visualization library. This library is used to create custom dynamic and interactive visualizations. To create each visualization, we first retrieve the most current butterfly observation data from the Beecology API endpoint, and parse this data into a usable form for each visualization. We then render the visualization dynamically based on this information, so our visualizations are always as up-to-date as possible.

4.1 Extending Bee Visualizations to Butterflies

The Beecology bumblebee visualizations are hosted on a different website than the web app. Users must first navigate to the ‘Data Selection’ page from the header. Users can then filter the existing bee data based on bee species, bee gender, bee behavior, flower color, flower shape, flower name, month, start year, and location (Figure 22). Once the data are filtered, some summary bar charts are generated . Then, the user can either click the ‘Analyze Data’ button or the ‘Data Analysis’ header to view six different visualization tools to give the users different insights on the bees (Figure 23). These tools allow the user to examine bee species number and

diversity at different locations over both seasons and years, as well as their flower species preferences.

The screenshot shows a 'Bee Data Selection' page with four rows of filter options. The first row contains 'Bee Name', 'Bee Gender', and 'Bee Behavior'. The second row contains 'Flower Color', 'Flower Shape', and 'Flower Name'. The third row contains 'Month', 'Start Year', and 'End Year'. The fourth row contains a checkbox labeled 'Filter by location?'.

Figure 22. 'Bee Data Selection' Page.

The screenshot shows the 'Data Analysis' page with a yellow header and a navigation bar. The main content area displays six visualization tools in a grid. Each tool has a thumbnail image, a title, a brief description, and a 'START ANALYSIS' button. The tools are: Bar Chart Comparison Tool (Bee and flower trait analysis), Diversity Heatmap Tool (Bee species diversity), Location Map Tool (Map of bee observations by species), Season Chart Tool (Chart of bee species by month), Summary Heatmap Tool (Chart of observations by month and year), and Floral Network Tool (Graph of bee and flower associations).

Figure 23. 'Data Analysis' page with the six visualization tools.

As mentioned earlier, there are many more butterfly species than bee species (121 versus 11). As a result, to create visualizations for butterflies, we took inspiration from the bee

visualization designs, but also had to allow for changes that enable the user to better explore the available butterfly observation data. We created a new visualization page to distinguish between the use cases for the bee and butterfly datasets. The data analysis tools on this page exclusively visualize the selected butterfly dataset, while the previous visualization page is dedicated solely to the selected bee dataset. Additionally, we implemented butterfly common names as suggested by Professor Gegear, to make it easier for users to identify and interact with the data.

4.2 Creating Butterfly Data Selection

The data analysis page we inherited already had a data selection page specific to bees. To ensure that newly developed butterfly observation visualizations only included the relevant data pertaining to butterflies, we created a new page for butterfly data selection. The layout of the page is consistent with the bee data selection page to provide a coherent experience.

In addition to the features the bee selection page has, the Butterfly Data Selection page includes a searchable checklist tree dropdown that allows the user to search for different species, and select species by family and subfamily (Figure 24). If the user checks the box to select a family, all of the butterfly species in that family will be selected. The same occurs for subfamilies. This enables the user to quickly select groups of butterflies by taxonomic ranking to include or exclude in any visualization. The butterfly species' common name is shown first, and the scientific name is shown second in parenthesis. This was not implemented for the bee selection page as there are only eleven species.

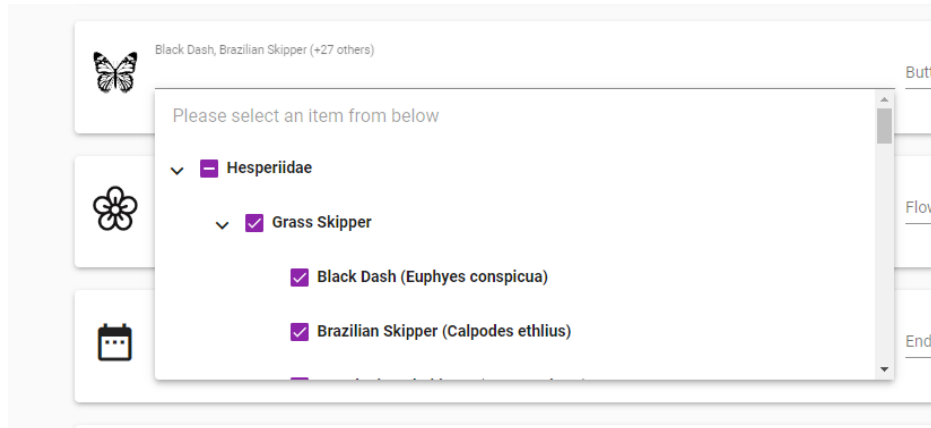


Figure 24. Searchable checklist tree dropdown component for butterfly species. Each species is binned in its respective family and subfamily dropdown.

We also moved the ‘Data Sets’ selection bar from the visualization selection page to the data selection page, to reinforce a regular flow of selecting data, and then visualizing the selected data. Previously, the user had to flip back and forth between these pages to effectively select a data set. This update can be seen below when comparing the previous and current page iterations in Figures 25 and 26. We wanted to separate the bee data selection and butterfly data selection with their respective visualizations, so we ensured that it was clear how to get from a bee or butterfly data selection page to the corresponding visualization page.

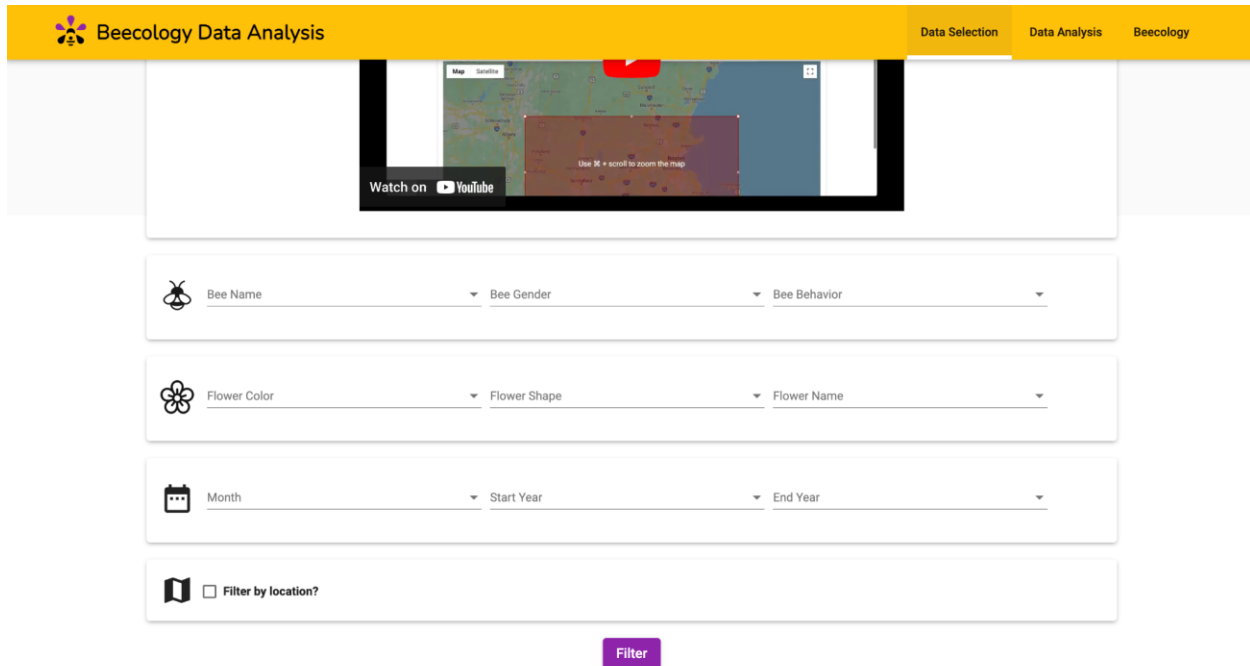


Figure 25. Previous iteration of the 'Data Selection' page. Users could only select bumblebee data on this page. The 'Data Sets' selection bar is not visible on this page.

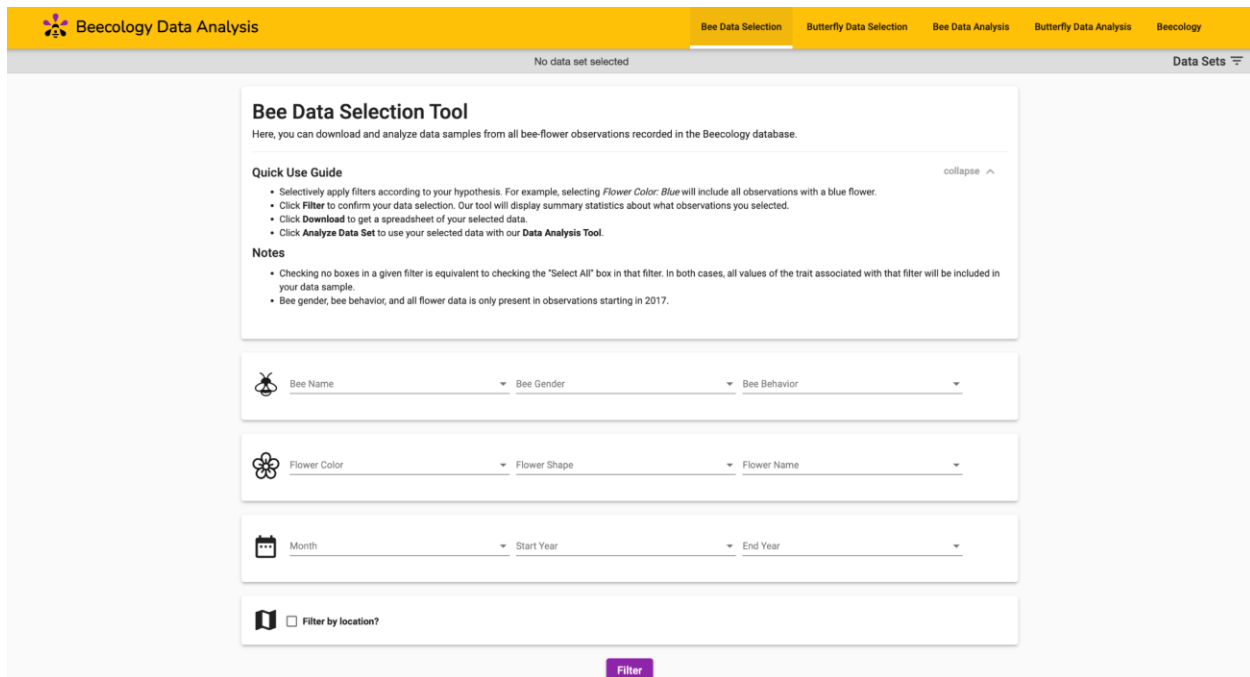


Figure 26. New iteration of the 'Bee Data Selection' page. The 'Data Sets' selection bar was added underneath the navigation bar. This design change is consistent between both the 'Bee Data Selection' and 'Butterfly Data Selection' pages.

4.3 Summary Statistics

Once the user clicks on the “Filter” button, a set of bar charts that summarize the selected data appear (Figures 27 and 28). These bar charts show a simple count of observations separated by a specific feature. The specific features are species, gender, behavior, tongue length, flower color, flower shape, and decade that the observation was recorded. All bar charts except for the one visualizing decade has the bars sorted and visualized from most to least observations. The bee species bar chart has a different color for each bee, as does the flower bar chart. The flower bar chart’s color is specific to the color the bar is visualizing.

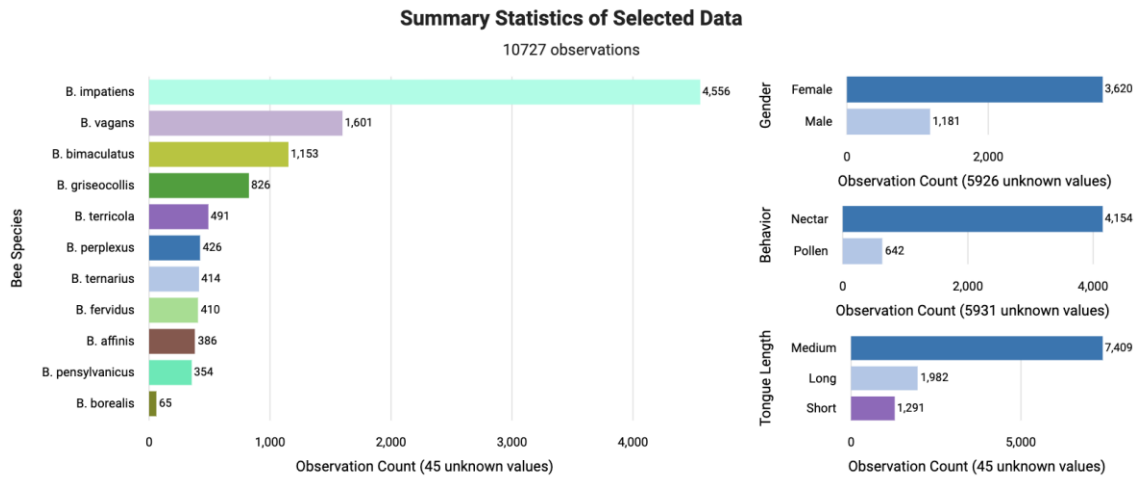


Figure 27. Bee Summary Statistics showing species, gender, behavior, and tongue length.

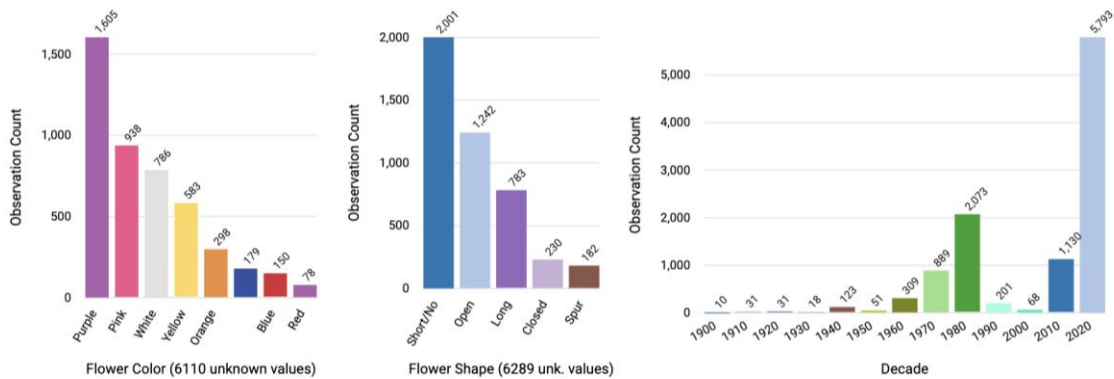


Figure 28. Bee Summary Statistics showing flower color, flower shape, and decade.

4.3.1 Butterfly Summary Statistics

Similarly, we also created a set of bar chart visualizations for the summary statistics for butterflies (Figures 29 and 30). This involved creating a new API endpoint called /summary-stats-butterfly that gathered the data in the backend and grouped the data depending on the features we wanted to display. We also changed the functions used to gather and structure the

data to suit our needs for the butterfly graphs. First, for the butterfly species bar chart, we changed the species' name to common names to fulfill Professor Gegear's request. The bar chart that displays all observations of butterfly species has a scrollbar that expands as more species of butterflies are observed, ensuring that each row has a minimum of 25 pixels. This is not necessary in the bee visualizations, as there are only so many bee species. Additionally, where there is a tie in the count of observations, the butterfly species are displayed in alphabetical order. Next, we removed the gender bar chart. All the records for gender in our database are listed as "unknown" as it is difficult to accurately classify the gender of butterflies using only a video or photo. We also fixed a bug where flowers that had no color listed were shown in the bar chart with no label. Currently, the bar charts for tongue length and decade are overly simplistic, as there is only one value represented. However, we expect this to change as more data is gathered with Beecology so we left them as is.

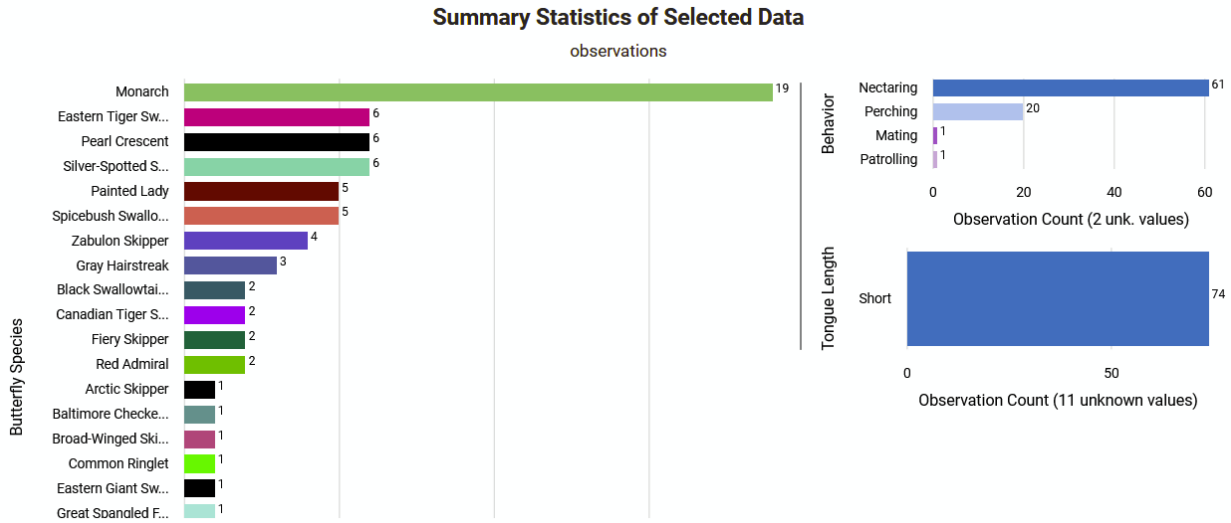


Figure 29. Butterfly Summary Statistics showing species, behavior, and tongue length. The species chart will continue to expand as more observations of butterfly species are gathered.

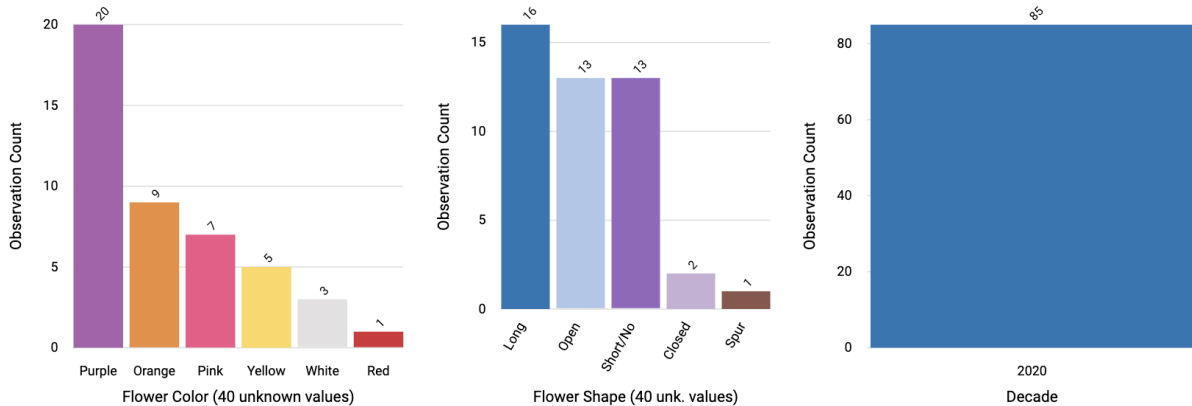


Figure 30. Butterfly Summary Statistics showing flower color, flower shape, and decade.

4.4 Separating Bee and Butterfly Data Analysis

We created a new additional page to handle butterfly datasets, separating the visualization tools from the ones visualizing bee datasets. Following the structure of the existing data analysis page, all six tools that are on this page visualize the selected butterfly dataset. We

will first discuss the common features we have implemented in all of the new butterfly tools, and then discuss each tool separately.

4.5 Common Features to Butterfly Visualization Tools

4.5.1 Butterfly Filter Dropdown Component

Since there are many more butterfly species than bee species, there needs to be an easier way to filter the visualizations without having to list out over 100 species on the side of the visualizations for users to select. We decided to develop a hierarchical dropdown filter to display the species in an organized manner where species are grouped into their subfamily and family, as seen in Figure 31. This component was incorporated into all of our data analysis tools pertaining to butterfly observations. We created two versions of this component. One version is always visible and is expandable depending on the depth of the tree; it is used in most of the data visualization pages. The other version expands when clicked and is searchable; it is used in the data selection page and the floral network page (Figure 32).

- Hesperidae
 - > Grass Skipper
 - Skipperling
 - Arctic Skipper
 - > Spread-wing Skipper
- > Lycaenidae
- > Nymphalidae
- > Papilionidae
- > Pieridae

Figure 31. Dropdown Filter Component.

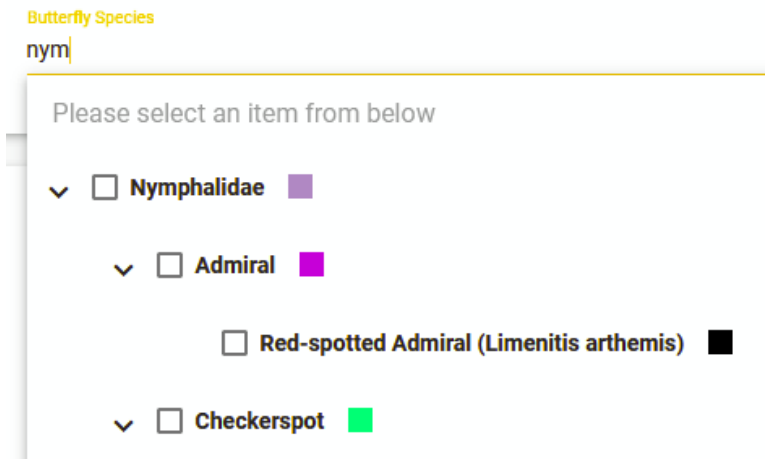


Figure 32. Searchable Dropdown Filter Component. Note that typing 'nym' brings up the family 'Nymphalidae,' as well as any of its sub-families or species

4.5.2 Butterfly Diversity Toggle Component

We wanted to allow users to customize visualization granularity, grouping the numerous butterfly observations into the species level, subfamily level, or family level. Due to the large number of species, there could be an overwhelming amount of information in the visualizations

that is not beneficial to the user. To solve this issue, we created a way to group observations by family, subfamily, and species. By allowing users to choose between levels of diversity, they can customize what they want their visualization to look like. This method makes it effortless for users to explore similarities and differences across species, subfamilies, and families. A visual of the diversity toggle component is seen in Figure 33 below.

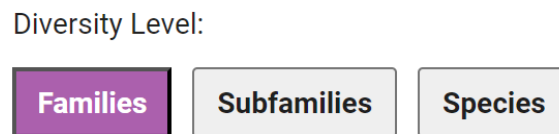


Figure 33. Diversity Toggle Component.

4.5.3 Butterfly Family, Subfamily, Species Global Coloring

We wanted to ensure a consistent look of the butterfly visualizations across every visualization. To do so, we created a consistent color mapping for each butterfly family, subfamily, and species and used this mapping across all visualizations. Users can easily recognize the relevant family, subfamily, and species when they switch from one visualization to another. Consistent coloring ties the visualizations together in a way that enables the user to record data from multiple visualizations and keep them coherent with each other.

We tried our best to consider using only colors that are accessible to all; however, because there are over 100 species, we would need over 100 different colors, and limiting the usage of various colors would have made many species' colors look the same. We ended up using an online tool that helps to create a number of unique colors that are as different from each other as possible on the color spectrum, and then we manually assigned a color to each species, subfamilies, and families so that the colors are uniform across all visualizations. Therefore, with

the time given for our project, we were unable to find a better way to display the colors consistently across visualizations while designing for accessibility. This could be a future project to tackle for future groups.

4.6 Descriptions for Each Visualization Tool

For each tool, we first describe the existing tool for bee visualizations, which we did not alter. Then we describe how we implemented a similar tool for butterflies, taking care to preserve features for consistency across pollinators, while adding features needed to incorporate the more complex taxonomy of butterfly species. Additionally, we ensured that the tools were intuitive and the designs were consistent between the bee and butterfly visualizations, providing a cohesive experience for users.

4.6.1 Bar Chart Comparison Tool

Bee Bar Chart Comparison Tool

The bar chart comparison tool compares the number of observations of certain species or flower traits across a range of time, elevation, longitude, or latitude. Traits are selected from a simple dropdown, while the range is selected from a horizontal scrollbar with adjustable width. The legend colors change depending on the selected trait. The height of each bar in the bar chart represents the number of observations of that particular trait over the selected range.

For bees, the possible traits include Bee Species, Bee Gender, Bee Behavior, Tongue Length, Flower Color, and Flower Shape. For example, in Figure 34, the first bar represents 116 observations of the *Bombus affinis* species from 1967 to 1977.

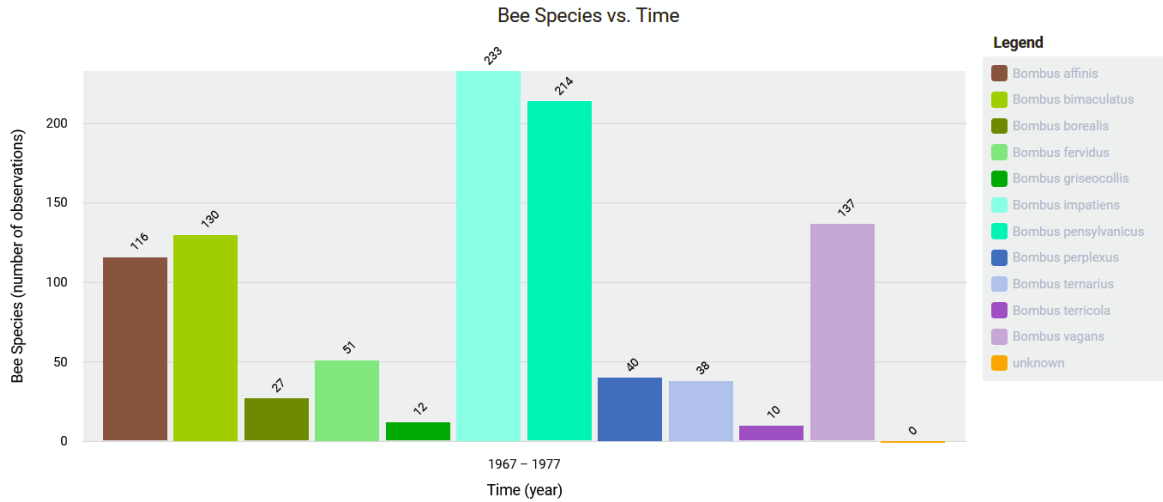


Figure 34. Bee Bar Chart Comparison Tool.

Butterfly Bar Chart Comparison Tool

The butterfly bar chart comparison tool has the same functionality as the bee bar chart comparison tool, but is filtered by slightly different traits. The current traits for butterflies include Butterfly Family, Butterfly Species, Butterfly Behavior, Flower Color, and Flower Shape. A visual of the butterfly bar chart comparison tool can be seen below in Figure 35.

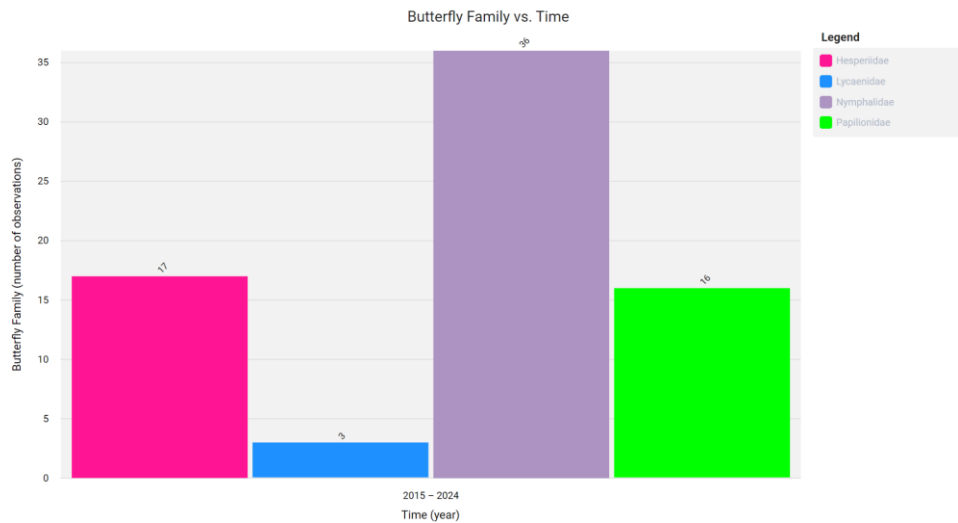


Figure 35. Butterfly Bar Chart Comparison Tool.

4.6.2 Diversity Heatmap Tool

Bee Diversity Heatmap Tool

The diversity heat map tool compares the diversity of species based on geographical location. The data can be filtered by the month or year of each observation. The default map generated is centered on Massachusetts, with observations grouped in squares of about 8 kilometers by 8 kilometers, and is powered by Google Maps API. To group observations in bigger or smaller squares, the user can zoom out or in the map, respectively. Each square is also transparent and colored from yellow to orange to red, with yellow representing low diversity and red representing high diversity. The default diversity index used is the Shannon index. However, the Simpson index or the raw species count can also be used to calculate diversity. Users can also click on a specific square on the heatmap. Once a square is selected, a chart appears on the right, showing the breakdown of the diversity and the number of observations counted in the selected area. This chart also has a legend informing the user of the exact species and count observed in the square.

For bees, there is no filter selection option for this tool. All species chosen on the Bee Data Selection Page are shown in the map as seen in Figure 36.

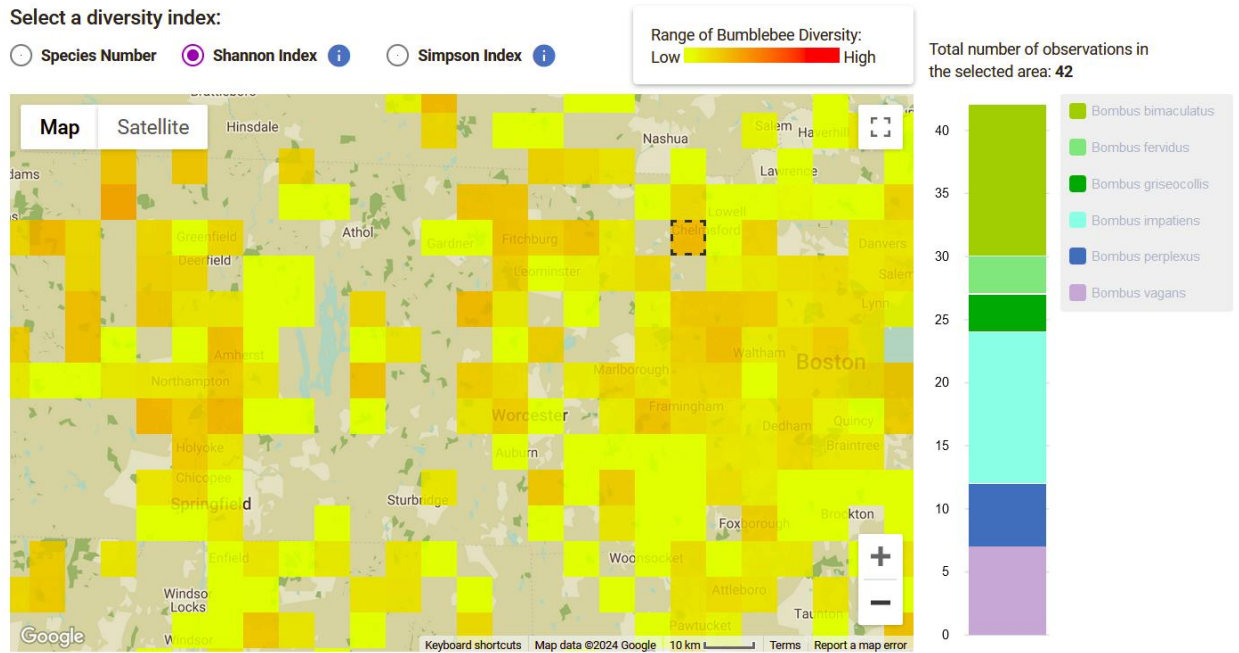


Figure 36. Bee Diversity Heatmap Tool.

Butterfly Diversity Heatmap Tool

For the butterfly diversity heatmap, we decided to add a filter using the drop-down selection component and the diversity toggle to make the charts simple to use and view (Figure 37).

Additionally, the diversity chart was slightly different than the other pages as it contained a stacked bar chart on the side, which ended up taking a decent amount of space (Figure 38).

Adding a drop-down component next to it would make the map too small to use comfortably by a user. We also tried placing the drop-down on top of the map, but this created large white spaces on the sides that were not as visually appealing. In the end, since a user would not need to view the drop-down and the bar chart at the same time, we decided to use a toggle feature as seen in Figure 39 to switch between the drop-down component and the stacked bar chart. This way, we were able to compact these two components to utilize the same space on the page and create a more visually appealing page.

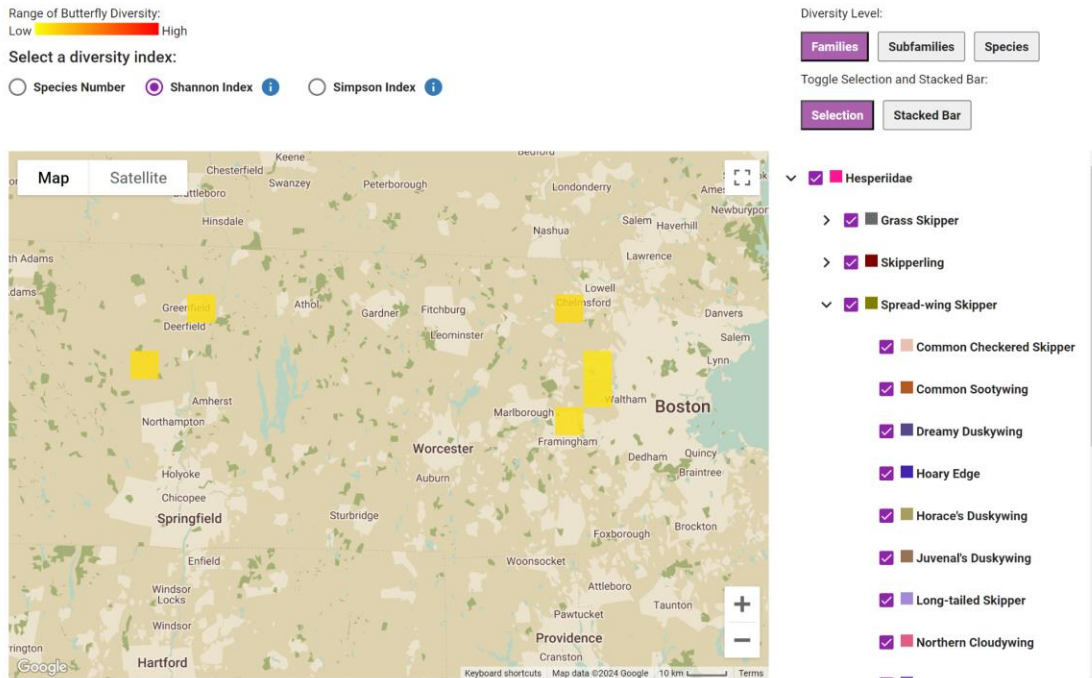


Figure 37. Butterfly Diversity Heatmap Tool Selection Tab.

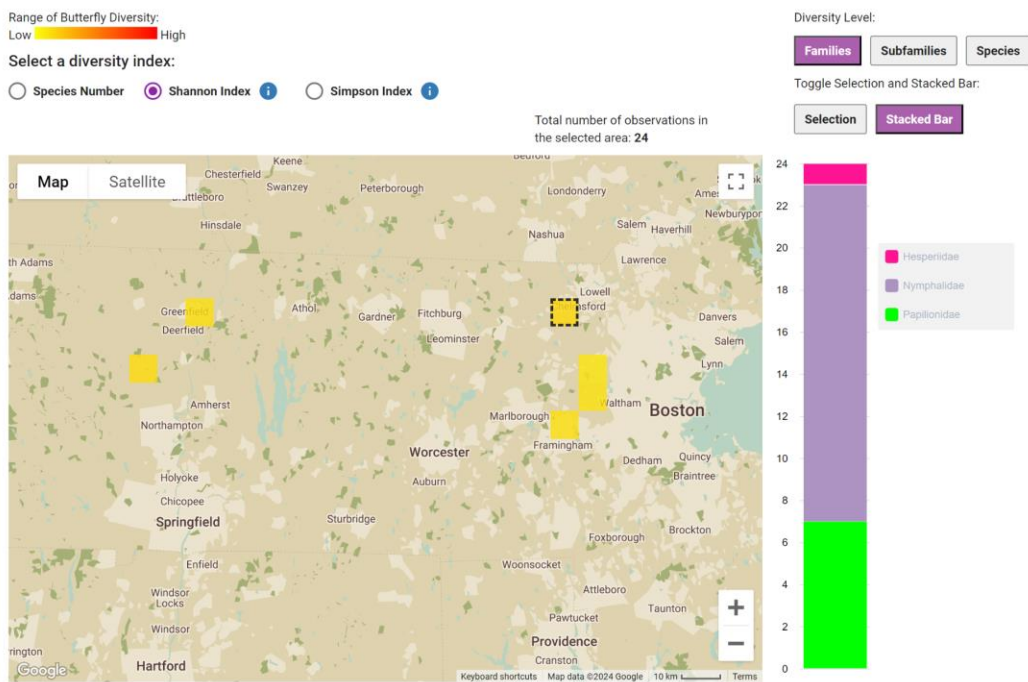


Figure 38. Butterfly Diversity Heatmap Tool Stacked Bar Tab.

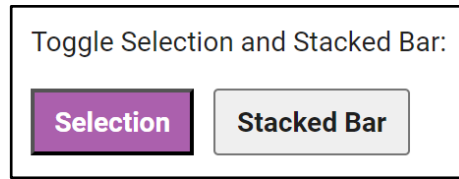


Figure 39. Selection and Stacked Bar Chart Toggle.

4.6.3 Location Map Tool

Bee Location Map Tool

The location map tool can also compare diversity and observations across different locations. Similar to the diversity heatmap tool, it uses Google Maps API to group observations into doughnuts on the map. The doughnuts are modified pie charts with colors representing each bee species and the total count of the doughnut in the middle. There is a filter on the right with every bee species which also acts as a color legend. Users can select and deselect which species to include in the visualization (Figure 40).

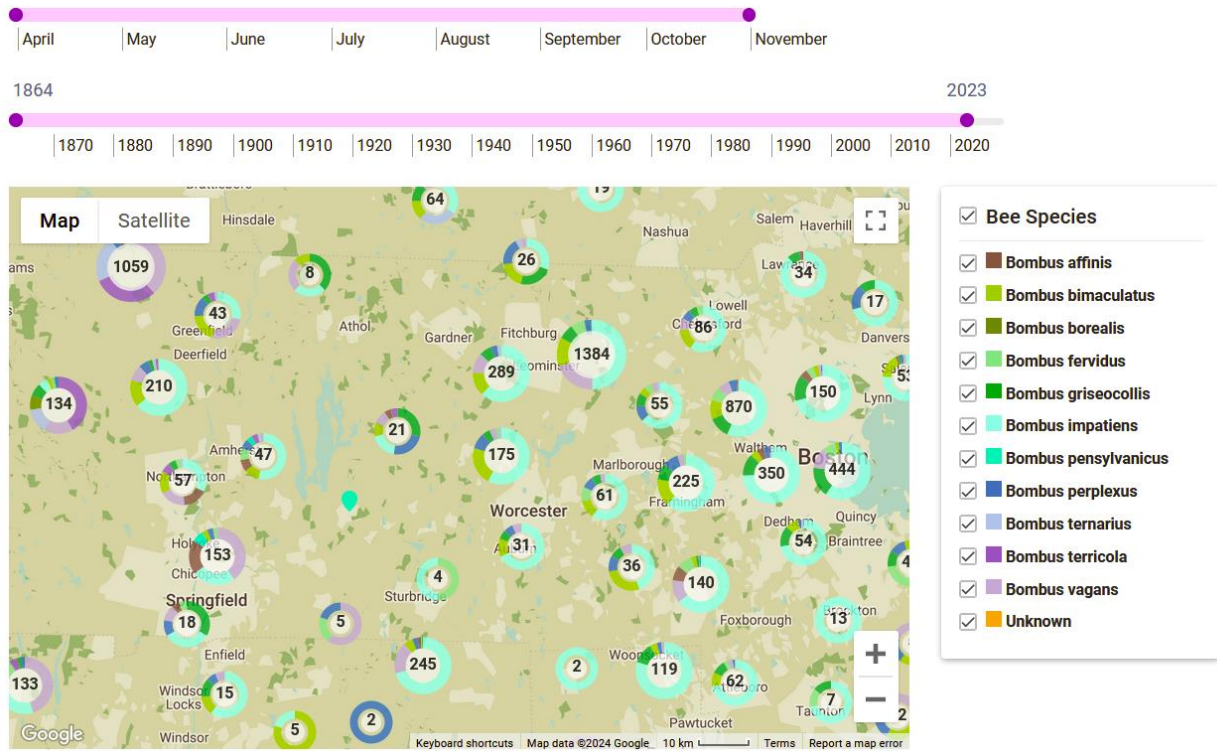


Figure 40. Bee Location Map Tool.

Butterfly Location Map Tool

In order to extend the current bee location map tool to butterflies, we added the butterfly filter dropdown component in order to select specific species, subfamilies, and families to view on the location map. We also added the diversity toggle component in order to group and visualize the butterfly species by family, subfamily, or species (Figure 41).

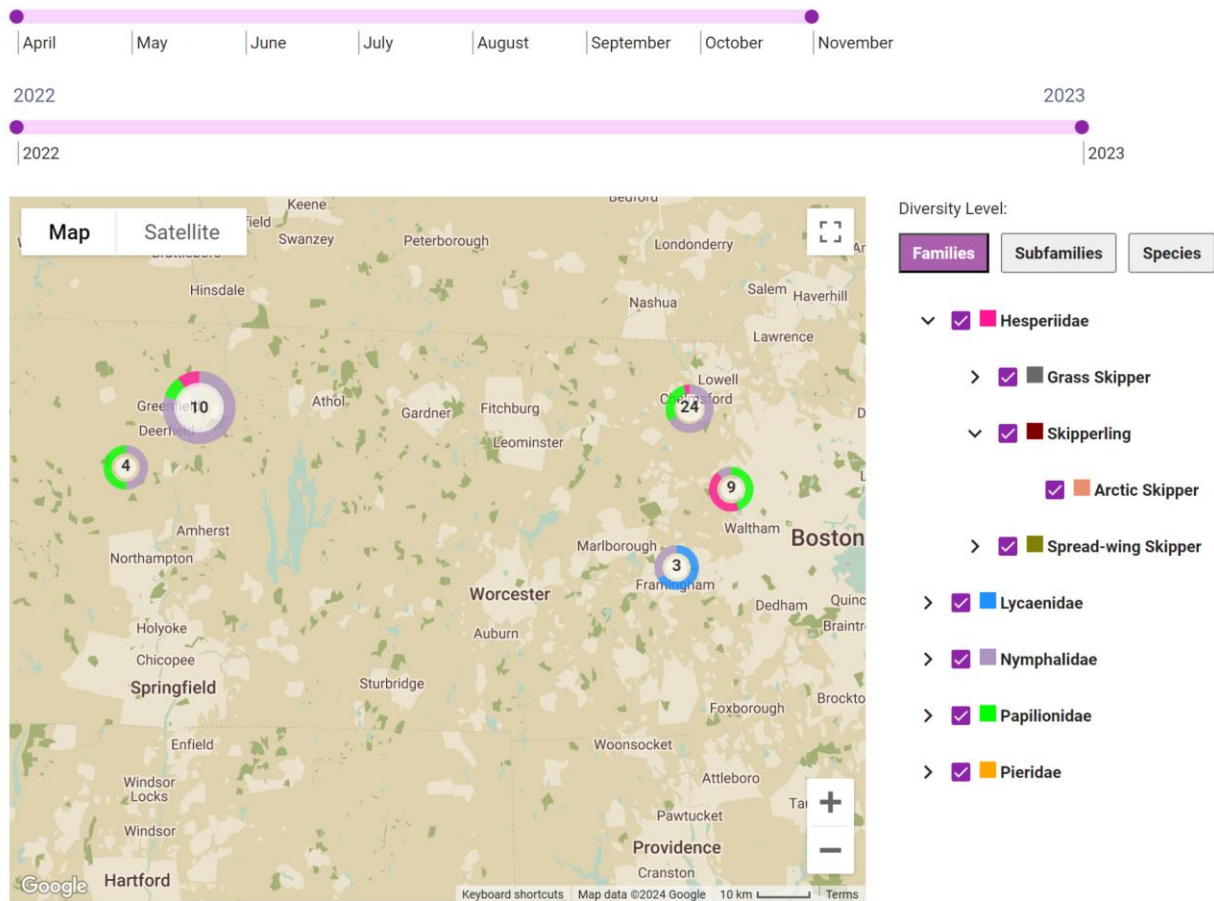


Figure 41. Butterfly Location Map Tool.

4.6.4 Season Chart Tool

Bee Season Chart Tool

The season chart tool shows how the observations of each species change throughout the year's seasons (Figure 42). To show when pollinators are active, it groups the observations by month from April to November. The data used for this visualization can be filtered by the range of years displayed, the range of months displayed, and the gender of the observation.

Additionally, the graph has four different kinds of scales for the line graph: absolute, logarithmic, relative to species, and relative to month. The absolute scale uses a linear y-axis for

the raw count of each species. The logarithmic scale uses a logarithmic y-axis for the raw count of each species. The relative to species scale uses the y-axis to represent the proportion or percentage of the total species count rather than the absolute number of species. This approach allows for a clearer comparison of the distribution or frequency of different species in a given context by considering their relative proportions. The relative to month scale shows the percentage of each species' observations for each month.

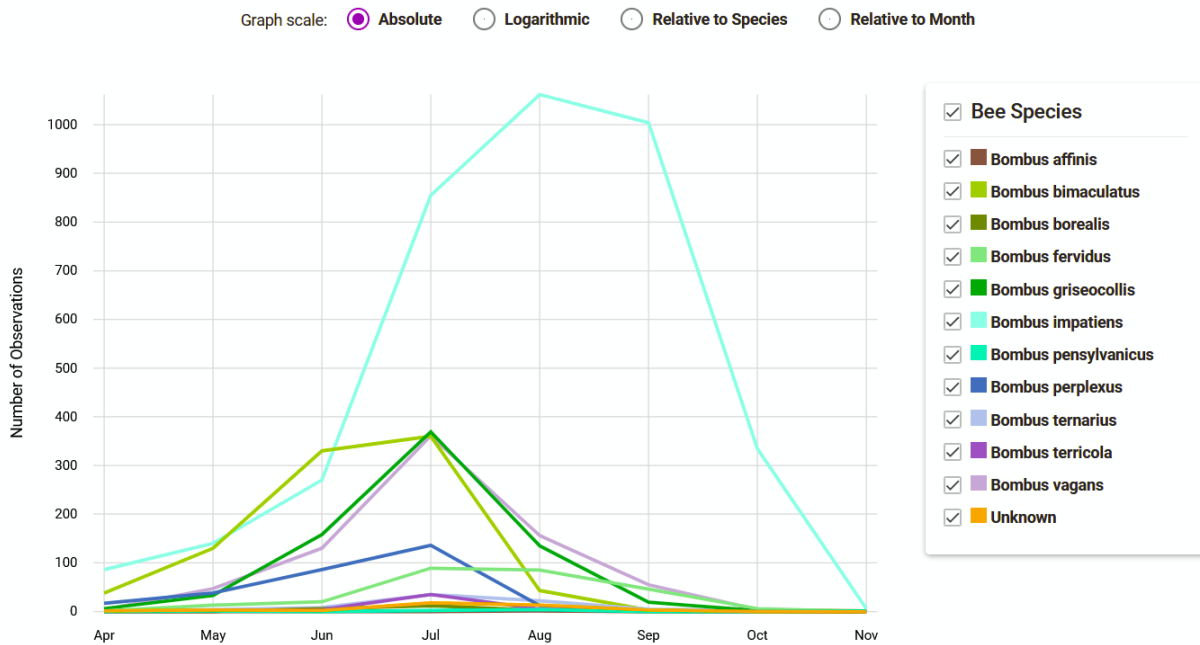


Figure 42. Bee Season Chart Tool.

Butterfly Season Chart Tool

Since there are so many species of butterflies, we added a diversity toggle that would group butterfly species by family, subfamily, or species. This prevents the lines from being too cluttered if all the butterfly species were represented. The Figures 43, 44, and 45 below display the difference in charts when the butterflies are grouped at different levels.

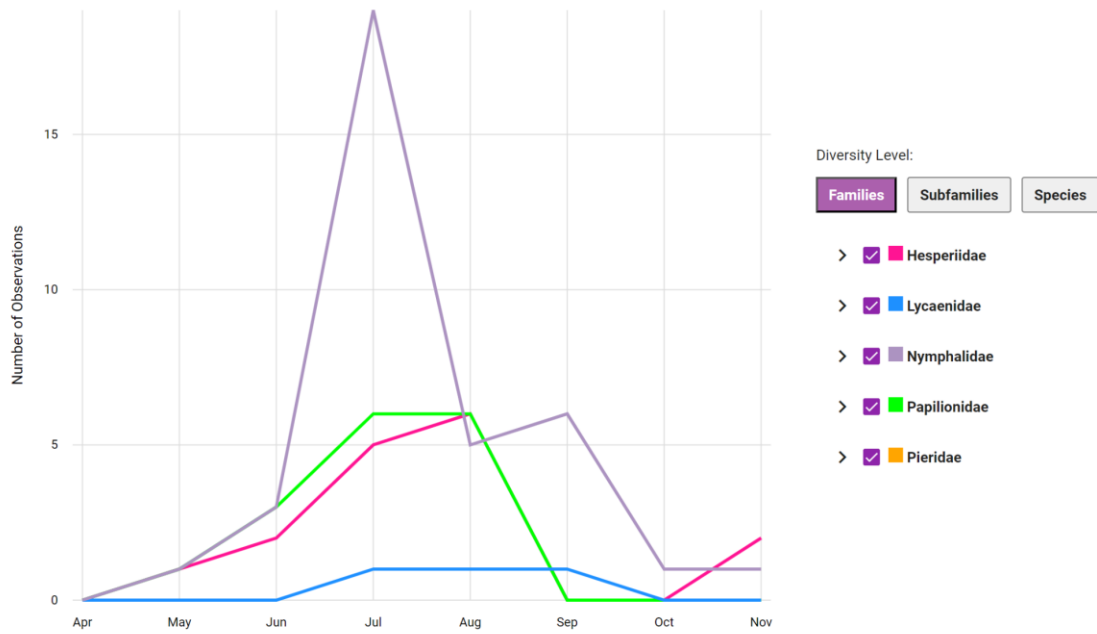


Figure 43. Example of Season Chart at the family level.

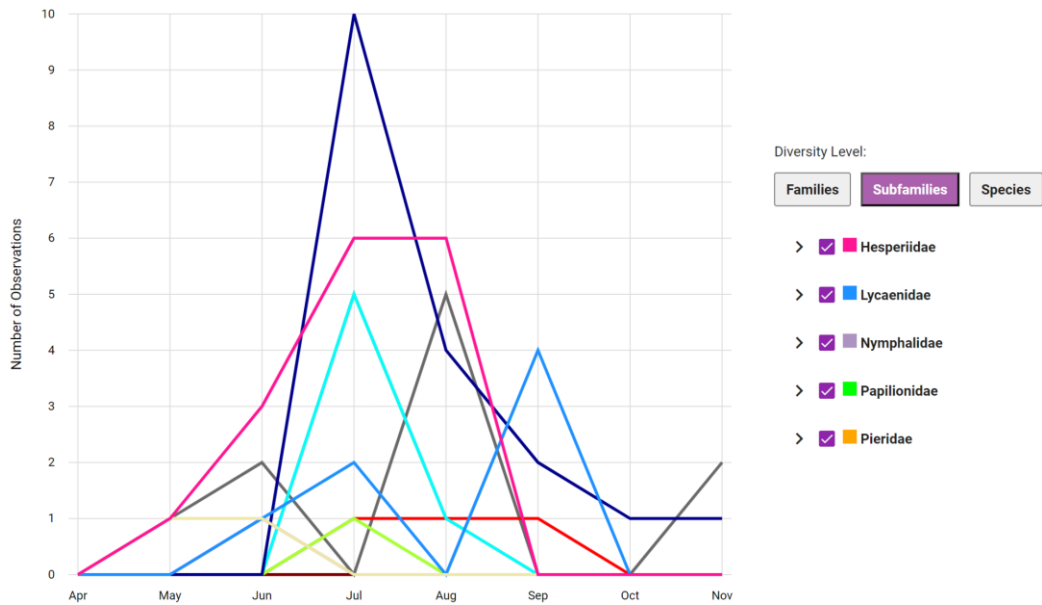


Figure 44. Example of Season Chart at the subfamily level.

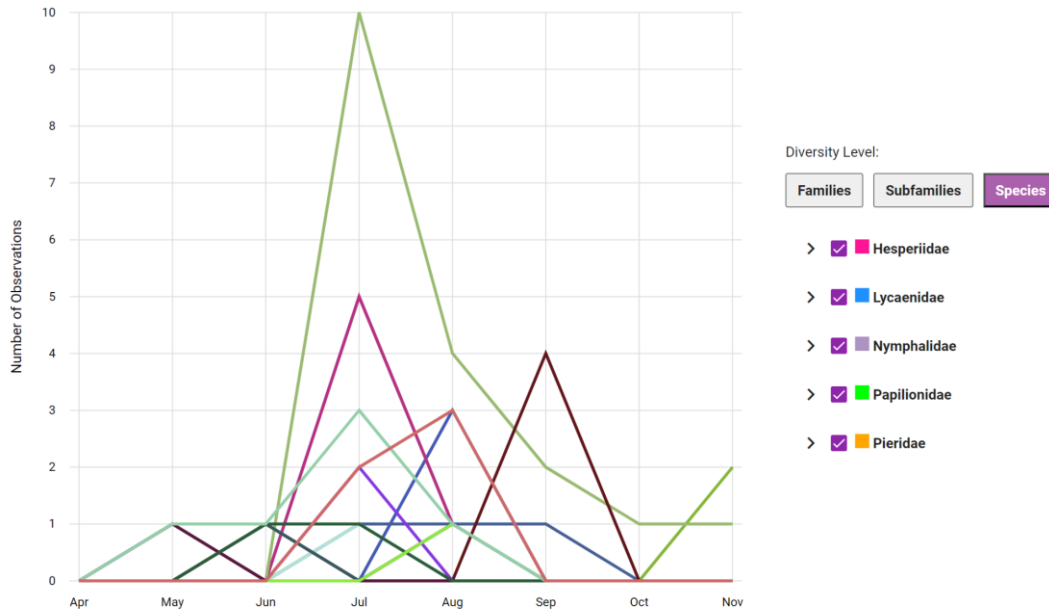


Figure 45. Example of Season Chart at the species level.

For the butterfly season chart tool, we added a drop-down component to filter the data and visuals. Since the butterfly features are relatively new, we do not have as many observations in our database, which results in a lot of species with zero data. In order to combat this, we added a minimum records feature to avoid showing families/subfamilies/species with less than a certain number of observations. As seen in Figure 46, the default threshold is one, so any species with less than one observation will be filtered out of the visualizations automatically.

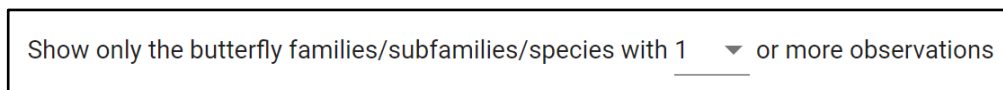


Figure 46. Threshold selection component.

Additionally, with the small number of observations, there was an issue with the season chart's y-axis labels repeating numbers when the maximum number was less than eight observations. This was fixed by making the graph increment by one if the maximum

observations for a month is less than 15, and if the maximum number of observations for a month is greater than or equal to 15, then it will start to increment each step by five. A comparison between the y-axis of these two scenarios is seen in Figure 47 and Figure 48.

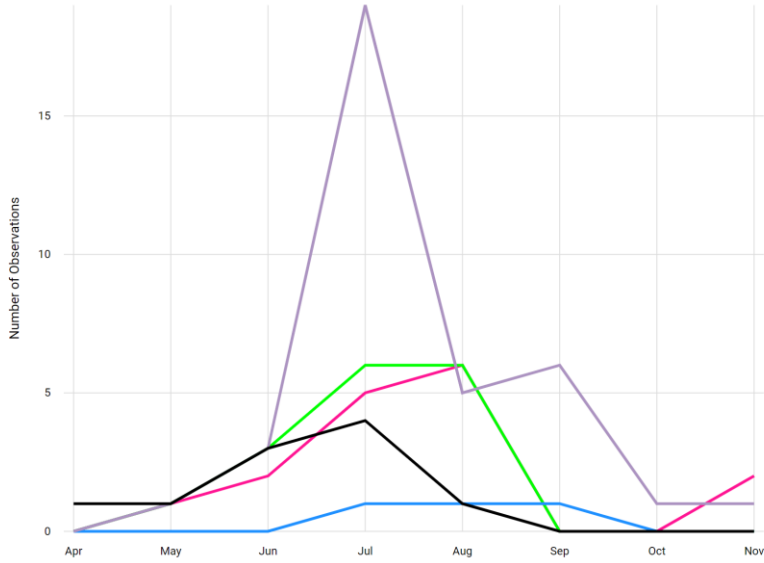


Figure 47. Example of Season Chart with greater than or equal to 15 observations.

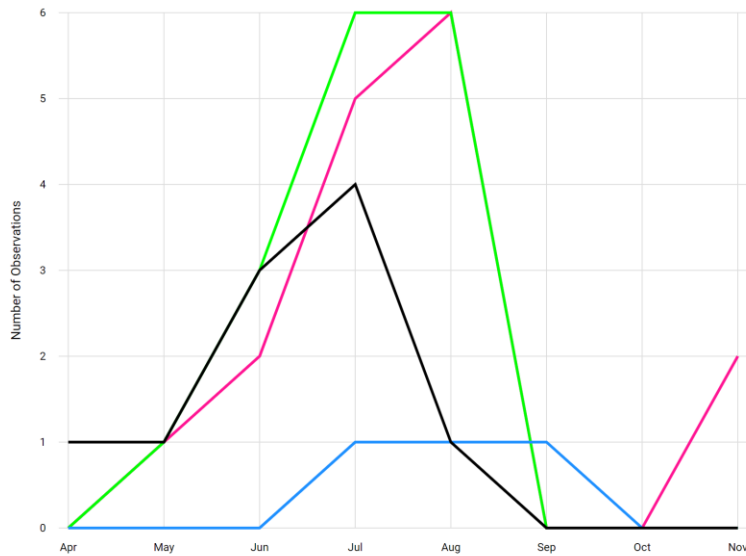


Figure 48. Example of Season Chart with less than 15 observations.

4.6.5 Summary Heatmap Tool

Bee Summary Heatmap Tool

The summary heatmap tool visualizes the number of observations made each month over all the years of Beecology data (Figure 49). The data used for this visualization can be filtered by the range of years displayed. It can also be filtered by species by selecting checkboxes on the right. The y-axis is the full range of months from January to December and the x-axis are the years the user has chosen to be displayed. Each tile on the chart has a color from yellow to red that represents the number of observations proportional to the maximum number of observations. If there are no observations, the square is colored gray. On the left, there is a colorbar from yellow to red that indicates the maximum and minimum number of observations for each month.

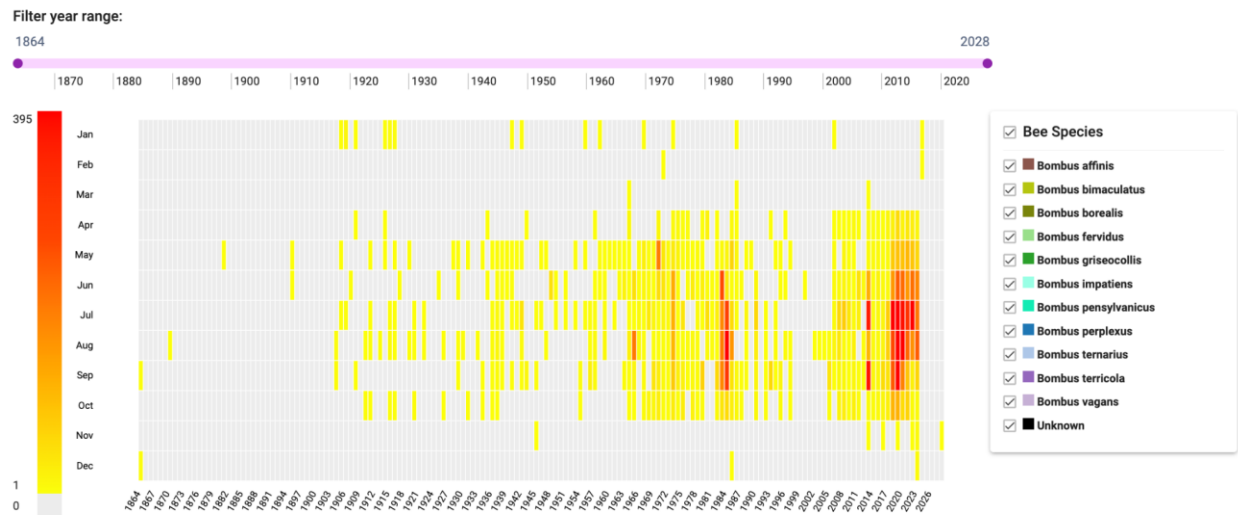


Figure 49. Bee Summary Heatmap Tool.

Butterfly Summary Heatmap Tool

Since the Summary Heatmap tool does not visualize on the species level, we did not have to alter the tool's initial design and functionality to fit butterfly data. We added the butterfly filter

dropdown component to allow selection of which specific species, subfamilies, and families to include in the visualization's observation counts (Figure 50).



Figure 50. Butterfly Summary Heatmap Tool.

4.6.6 Floral Network Tool

Bee Floral Network Tool

The Floral Network tool visualizes observations paired with their respective flowers (Figure 51). The number of observations is shown as a number next to the observation bar and expressed in the height of the bar for both bees and flowers. When hovering over one of these bars, the user can view the species name of either the flower or the bee, as well as the number of observations of that species expressed as both a number and a percentage of the total. These observations can be filtered by species, behavior, tongue length, and gender via a dropdown.

They can also be filtered by flower shape and color via a dropdown. Lastly, they can be filtered by date of observation as well via a sliding bar at the top of the visualization.

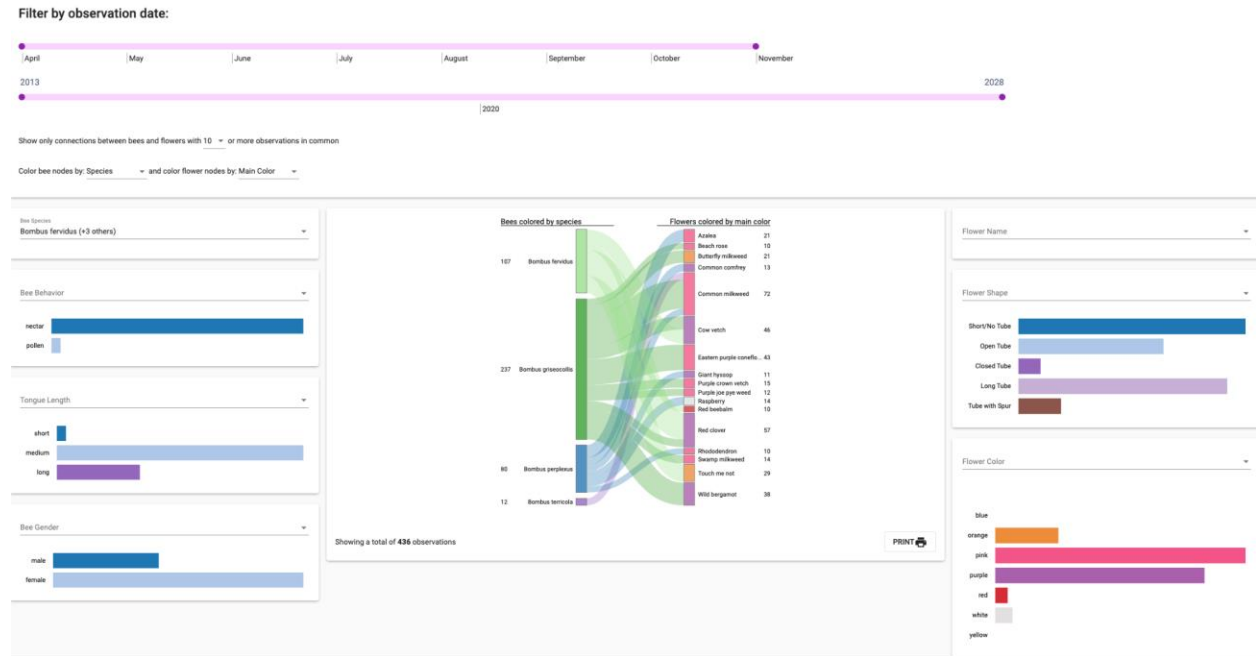


Figure 51. Bee Floral Network Tool. Four bee species are selected (*Bombus fervidus*, *Bombus griseocollis*, *Bombus perplexus*, *Bombus terrestris*).

Butterfly Floral Network Tool

To create the butterfly floral network tool we took inspiration from the existing bee tool, and implemented changes to show more information about the butterfly, such as family, subfamily and species in an intuitive way.

The Butterfly Floral Network Tool is used to show butterfly observations paired with which flower they were observed on. For each observation, we show the family, subfamily, and species of the butterfly, and connect the observation to a flower using a colored trail based on the set butterfly species' color. We also show the family and subfamily colors on each observation, sorting alphabetically first by family, then by subfamily, and then by species to give the user a view of how these factors come into play (Figure 52).

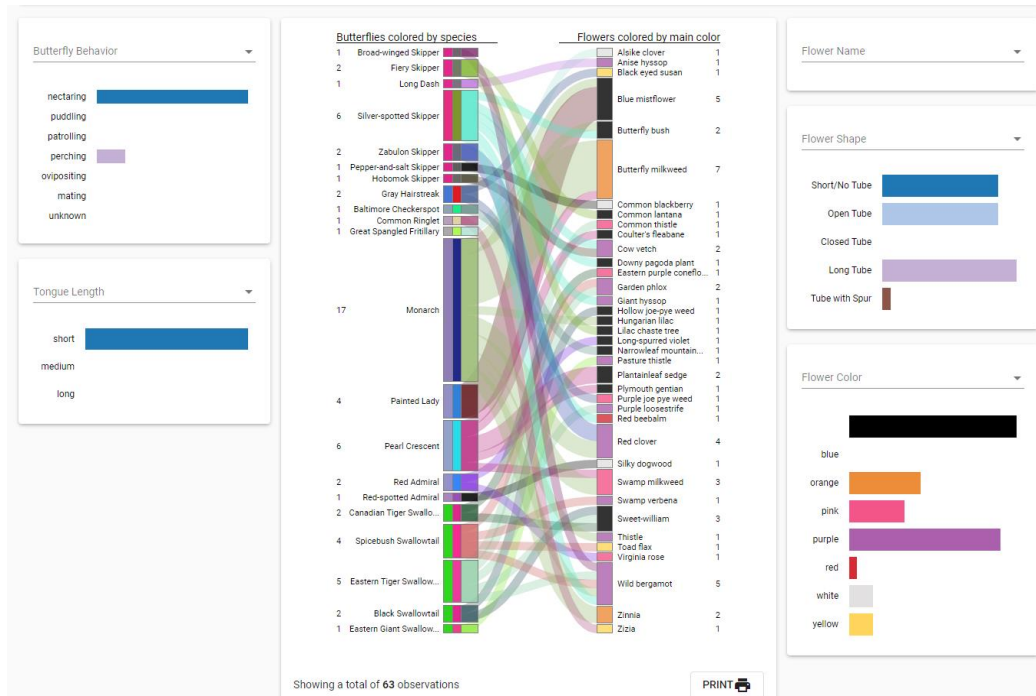


Figure 52. Butterfly Floral Network Visualization.

On hover of each butterfly observation, you can view the family, subfamily, and species, as well as the number of observations expressed as the actual number, and as a percentage of the total (Figure 53). There are also options to filter based on observation time.

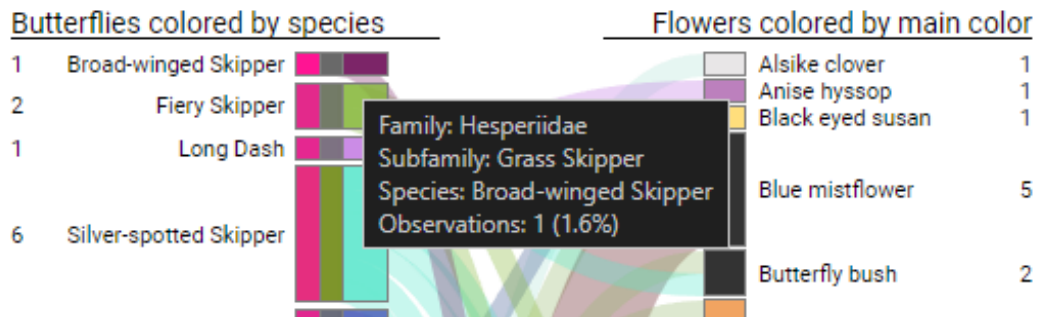


Figure 53. Butterfly Floral Network Visualization on Hover.

There is functionality to filter the butterflies in the visualization by family, subfamily, species, butterfly behavior, and butterfly tongue length. To filter the flower observations, the user is able to filter based on flower species, shape, and color.

5. Conclusion

5.1 Final Thoughts

Our new frontend design of the web application refreshed and updated the site, helping bring improved functionality for larger devices. We were also successful in improving upon the UI design in making a new desktop-focused interface. This new UI improved upon the issues we saw in the mobile-only web application from version 5. In order to keep up with the growing number of desktop users, we created a desktop interface for the web application that utilizes the entire width of the screen instead of a mobile view on desktop. This allowed us to increase the detail with which we can display the various bees and butterflies along with identifying information regarding them. By adding additional width to the viewport, both images and graphics have been expanded, giving app users a refreshing and more engaging experience. Additionally, these larger photos allow for much easier identification of species when creating logs, improving practicality of the app when recording data.

The expansion of data visualization tools to include butterflies allowed users to visualize butterfly data that they were previously unable to see, in an engaging and interactive way, which should in turn motivate greater collection efforts and promote awareness of how these insects behave in the wild. We were able to develop six visualization tools inspired by the existing bee visualization tools, customized to visualize butterfly data. We also developed new data selection and data analysis pages for butterflies, which are separate from the bees, and enable the user to engage with the butterfly data. These tools incorporate features that enable users to thoroughly explore and analyze the diverse butterfly species and their populations.

5.2 Challenges Faced

One of the biggest challenges our entire team faced at the beginning of the project was setting up our local environments to run the code. We found that the provided documentation was not very accurate or clear, which resulted in hours of work spent on initial setup. Furthermore, there was not much documentation provided for the actual front and backend of the project. This also consumed a considerable amount of time as we had to figure out the structure and design of the project on our own.

Another significant obstacle we faced during the end of the project was creating live demonstrations or updates that reflected all the changes we made. As most of the development of the project was done locally, pushing live project code was appended to the end of our agenda. Once we got to a reasonable stage to tackle this, we quickly found that working with GitLab was yielding near constant errors. The root cause of these issues was the inactive state of GitLab's runners. In order to run a GitLab pipeline, GitLab makes use of live servers called "runners" that will take a repository's custom pipeline and run the various stages described in its code. Two important stages in this process are building the live code, and deploying it to its proper location. When we began the process of working with live code, all of GitLab's runners were inactive. Throughout the course of a month, a long email chain with WPI's ARCUnix team would solve the initial issue and activate the runners. However, assigning the runners to their proper repositories also proved to be an issue. Through the use of two general sets of runners, GitLab uses a Bio-CS Bridge and Beecology runner to deploy to different locations. After gaining higher permissions in the Beecology GitLab and unlocking the runners from their projects, we switched around runner repository locations to get the proper runners deploying websites and APIs to the proper servers. In order to keep a higher standard of documentation, we added a section to the

Beecology documentation repository detailing all the information we learned throughout this process.

5.3 Future Work

Throughout the duration of the project, the combination of various recommendations from the team and the professors as well as numerous complications that came up during development lead to a final list of ideas for future work.

5.3.1 Additional Bee-ID Web Application Implementations

Along with the UI design revisions we made for the web application, we proposed various new features that could be added. These features included quality of life improvements to the systems of version 5 of the web application. These included adding flower images to the iNaturalist results when identifying a flower and additional filters for searching through the B-Dex. Other features included new user engagement features that would help us retain users so they would continue to submit bee and butterfly logs. Some of these features were a medal/award system, a level system, and a user spotlight feature. Each of these were proposed as additional features our team could add once we were finished with redesigning the web application, but we ended up not having enough time to implement them.

The first quality of life feature we intended to implement was adding flower images to the flower identification iNaturalist results. When the user identifies a butterfly, they have the option to identify the flower the butterfly is on. This can either be done using manual identification or automatic identification through iNaturalist. If they select automatic identification, iNaturalist gives the user a selection of the closest-matching flowers it identified.

However, these flower names do not have any pictures accompanying them and are blank portraits that can be filled in. A future team could implement this feature by utilizing the flower photos we have already in Google Drive folder titled Plant picture jpgs. This drive contains various common flowers that a user would see in the New England area. The only problem is that the names of these flowers would need to be cleaned and formatted to be usable in the web application. The current names vary between species, some are capitalized, some are not, some have hyphens in the flower names and some do not. These names would have to be standardized in order to call the correct names in the web application; ideally they would match with the strings that iNaturalist provides in the API. Additionally, the group implementing these flowers would have to store them on the Beecology server and make calls to them through their URL. This could be done in a similar fashion to how the B-Dex works currently, as it calls to the beeres/images/B-Dex folder to get bee images. You could then implement these two features to dynamically get the iNaturalist suggestions and tack them on to the end of a URL to display the images. This is an oversimplified solution to the problem presented, but should be a good start on how a future team could solve this problem.

A second quality of life feature that was proposed was more filters for the B-Dex. The current filters that exist are for butterflies, and they are sorting between the family of butterflies and displaying each individual butterfly species. This dropdown could be expanded to better help users sort for specific bees or butterflies. A potential filter that could be implemented would be an active months filter for bees and butterflies. The active months categorization exists for both bees and butterflies, and may help users better narrow down a species that they saw if they remember the month they saw it in. Another filter that could be added would be a filter for the dorsal colors of a butterfly. This only applies to butterflies, but would better differentiate species,

as most butterflies have a main color that people think of when they see a butterfly. These features could be implemented into the web application by repurposing the code for the family/species selector in the butterfly B-Dex, but expanding it to include new filters like the aforementioned dorsal color and active months.

One user engagement feature for the web application that was proposed was the medal system, as shown in Figure 54. The medal system would reward users for doing certain tasks in the web application. An example of this would be a user submitting their first bee photo. Once doing this they would be notified that they earned a medal called ‘Just Winging It’ or a similar bee/butterfly pun. Each of these medals would be listed in the accounts page under the user’s email and username. The system is primarily intended to encourage users to submit more bee and butterfly photos to the database, and to give users a sense of accomplishment when they unlock a medal. We theorized that a potential downside of this system would be that users could submit photos for the sake of getting medals instead of for the sake of citizen science. A future team would have to investigate potential positives and negatives to gamification and user engagement features in similar apps to understand the full picture. If a future team wanted to implement this feature, we would recommend them to place the list of medals under the user’s information on the accounts page. We intentionally left additional space under the new accounts page to leave space for a medal or level system. The future team would have to make additional edits to the user table in the beecology2 database in order to store the medals that each user had earned. They would then need to make an additional database call when retrieving user information for the accounts page in order to display medals earned.

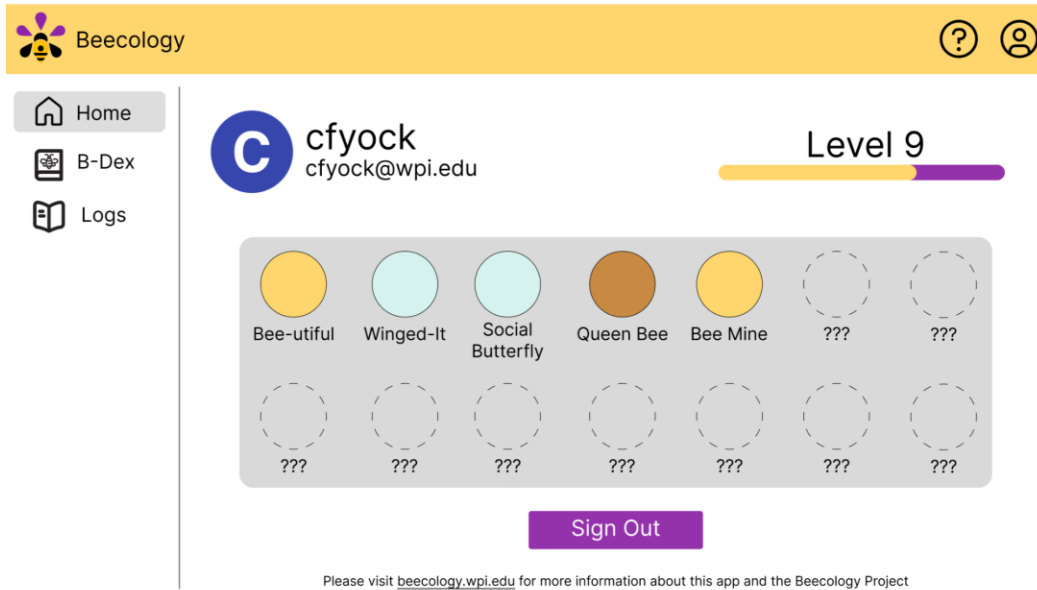


Figure 54. An early Figma prototype for the medals and level features for version 6.

Another potential user engagement feature for the web application that was proposed was the level system. Similar to how the medal system functions, the user would earn some experience after each submission of a bee or butterfly log. This experience would go to the user's level until it increased. This system could also be implemented in tandem with the medals system to give users medals each time they level up. This system has very similar positives and negatives as the medals system. The implementation would also be similar, as a future team would implement it by adding to the existing user table in the beecology2 database. They would then have to call the database for the user's level, and update the database if any changes to the level take place.

A third potential user engagement feature for the web application was daily or weekly streaks. These streaks would reward people for submitting a bee or butterfly log every day or week, depending on how the system is implemented. This method of user engagement is very popular in other apps like Snapchat, Duolingo, and habit-tracking apps. This is the most time-

tested method of user engagement, but might create bad habits for users, if it encourages them to incorrectly report the date of an observation, for example. Potential upsides include encouraging new users to submit more photos, and getting more data for the Beecology project as a whole. A future team could implement this feature by tracking the days between user submissions, by adding additional data to the user table in the beecology2 database. This would also require an additional call when loading the user's information in the accounts page. The streaks could then be displayed on the accounts page as the number of days or weeks the user has submitted a log in a row.

Another user engagement feature that we discussed adding was some form of user spotlight. The idea was that when a user submits a bee or butterfly log, they could choose whether or not to enter their submission in a pseudo-random raffle to be showcased on the landing page of the web application with their name in the bottom corner for credit, as seen in Figure 55. This selection process would be pseudo-random because we would need to make sure that any picture shown on the front of the web app would have to be of high quality and have the subject of the photo not be blocked off by the title and buttons. After talking with Professor Gegear, we decided not to implement this feature because he wanted it to be on the main Beecology website, which we would not have time to change. We also had the idea of a "Bug of the Month", which we eventually decided against because it would encourage users to submit logs for that creature even if they did not find it themselves. In other words, it would increase user engagement at the cost of possibly increasing the number of inaccurate submissions, which we decided was not worthwhile, since the main goal of Beecology is to collect accurate data. Furthermore, during winter months, it would not make sense to include a "Bug of the Month" since there would not be any bees or butterflies out during those months.

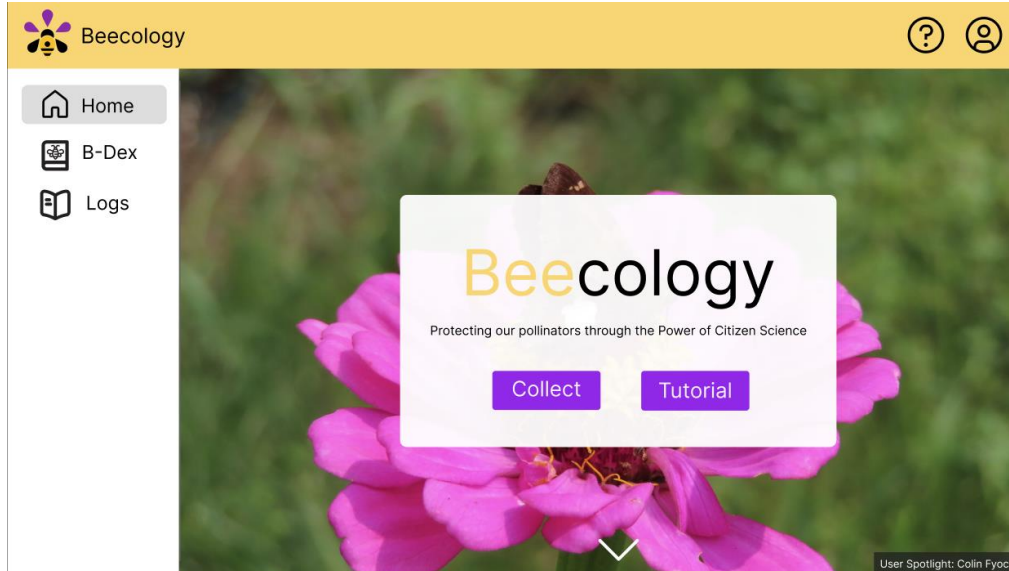


Figure 55. An early Figma prototype for the landing page and user spotlight in version 6. Note the bottom right user spotlight text showing that the background image of this page was taken by the user, Colin Fyock.

A final user engagement feature, which was suggested by Professor Gegear, is a group log sharing feature. This feature is similar to the already existing feature which allows users to copy the URL of submitted logs and give other users the link. This group feature would allow a user to create a group and invite other Beecology users. This group would then have a page that shows each of the logs each member in the group had submitted. This would allow groups of users to share logs with each other, and show off the bees and butterflies they have found. This idea could be implemented in tandem with the previously mentioned social features. For instance, a group could have its own level or its own badge achievements that would encourage a group's users to work together towards a common goal. A future team could implement this feature by adding additional information to the database. A team would first have to create a groups table in the beecology2 database that stores a group's name, id, and its members. They would additionally have to edit the existing users table to include the groups that they are in.

They would then need to make a new Angular component that displays the groups interface. This interface could be made up of a search bar, or an input for a group's id. It would also allow a user to create a new group and see the current groups they are in. This group feature could be implemented in many different ways, but the previous description is one potential path a team could take.

5.3.2 Future Improvements to Butterfly Visualizations

As mentioned before, currently, the color coding for each butterfly species, subfamily, and family is manually assigned so that the colors can be uniform across all visualizations. However, the colors currently chosen do not accommodate those with visual impairments. If there can be a better way to color-code the butterfly species, subfamilies, and families, it would make the visualizations more user-friendly and accommodating to all users.

Additionally, since butterfly visualizations are a new feature to the website, it would be beneficial to ask users for feedback on the new features. For example, creating a feedback form asking if the visualization tools are easy to use and if there are any bugs within the tools. Having user feedback can help to identify problems that were overlooked and help to make the data analysis experience better for our users.

Furthermore, adding more visualizations including flowers, such as a geographical display of which butterflies were seen with which flowers on a map could be beneficial to some research hypotheses, and would be making more complete use of all of the data at our disposal that is collected by users.

5.3.3 Upgrading to the Latest Angular Version

As was mentioned in Section 4.2, we were able to upgrade the project to Angular 16, with the caveat that there were several errors that we were unable to debug. The project compiled and everything worked as expected from version 5 of the app except for a few missing features on the log creation and B-Dex pages. Section 4.2.1 describes the importance of keeping software up-to-date and some of the risks of delaying updating, which is why we recommend that a future team prioritize upgrading the application to the latest Angular version.

While there may be other options for how to update the application from Angular 8, we recommend first trying to use the Angular Update Guide Tool since that is the official guide that Angular recommends. If that is unsuccessful, we recommend creating a new Angular project in the newest version and importing all of the old project files. Then, we would recommend going through the different TypeScript files and installing all of the packages used and adding polyfills where necessary. Once the project builds, we recommend going through all of the pages to find the packages producing errors, and finding alternatives for those. One package, in particular, that caused a lot of errors was `ngu-carousel`. It has sparse and inaccurate documentation and produced several seemingly unsolvable errors. We would advise future teams to look into an alternative carousel package or to modify the layout of the pages using it. The Google Maps package used, `agm-maps`, would also need to be replaced since it is not compatible with newer versions of Angular. This is a simplified roadmap and the entire process would easily take 50 or more hours if starting from scratch, which is why we saved our failed attempt at upgrading to Angular 16 in a GitLab repository, available under the Beecology GitLab.

5.3.4 Migrating Beecology to Amazon Web Services

During the course of our project, the idea of migrating the GitLab Beecology repositories and general code deployments to Amazon Web Services (AWS) was raised as a potential future project. Since migrating the entire project to AWS would be a large undertaking, we decided to sideline the idea while describing the pros and cons of a potential migration.

AWS is a cloud-based platform that offers a plethora of services that all work in tandem to operate large-scale projects. AWS services include API creation and hosting, database management, code and repository storage, server-based hosting, and many more helpful tools. With the right knowledge of these services, a project such as Beecology can be solely hosted using AWS. The current Beecology space is also home to various disconnected components. The server, API, and database are all in different locations, only being joined together to host the completed, built code in each respective GitLab repository. Through the use of AWS, all of these components could be migrated into different web services and, with proper documentation, could be easily accessed by future peers in one centralized platform. The proper learning curve would shift from learning a large amount of separate skills to access Beecology components to learning how AWS connects its services to create Beecology. The most important benefit to this migration would be the separation from WPI-related services. Currently, Beecology requires WPI-hosted servers and databases to function, meaning WPI has a large stake in the future development of Beecology. By migrating to AWS, the services required to run Beecology would be run by Beecology administrators alone.

Moving onto the drawbacks of a migration, AWS is a specialized platform that requires a baseline understanding of its features and how they connect together in order to properly run an application space. If any prospective developer were to begin work on Beecology, they would

first have to go through the roadblock of learning what AWS is, how it works, and how Beecology uses it. Assuming a developer does not have insights into the basics of AWS, the learning curve of understanding what Beecology uses in each service is steep. With the new documentation of Beecology that our team developed, the process of beginning to work on any projects should be simpler. Switching to AWS will either require extensive training or documentation. Another issue would be the sharp increase in price. The prices involved in running a full-stack application are paid at different rates for each service, meaning the more services that we implement to get Beecology running on AWS, the larger the overall price would get. While WPI is currently funding Beecology-related storage and computation costs, the Beecology Project would need to assume those costs if Beecology were to move to AWS.. From a help-desk service standpoint, switching to AWS would also include abandoning reliance on WPI's ARCUnix team in favor of AWS' customer service. This would be a major drawback, as WPI's ARCUnix team has a relatively quick response time and is able to provide specific insights, which AWS' customer service would lack in comparison. Furthermore, the WPI ARCUnix team is dedicated to helping WPI professors and students and can be reached in person if necessary. The AWS customer service, on the other hand, is less available and less reachable than WPI's staff. The final issue would be the time it would take to complete the migration. While it could be left up to the project's yearly assistants, the task will most likely involve a student-based project group such as ours spearheading the conversion. Assuming the Beecology team assembles a group of students well-versed in AWS, the migration planning, application, quality control, and error testing would take the better part of a full MQP. While for the students, it might be a worthwhile project, the Beecology application would see no discernible change in terms of its functionality.

References

- Aristeidou, M., Scanlon, E., & Sharples, M. (2017). Profiles of engagement in online communities of citizen science participation. *Computers in Human Behavior*, 74, 246-256.
- Butterfly and moth taxonomy. (n.d.). *Butterfly and Moth Taxonomy | Butterflies and Moths of North America*.
[https://www.butterfliesandmoths.org/taxonomy#:~:text=Butterflies%20and%20moths%20are%20members,Lepidopteran%20taxonomy\)%20are%20not%20static](https://www.butterfliesandmoths.org/taxonomy#:~:text=Butterflies%20and%20moths%20are%20members,Lepidopteran%20taxonomy)%20are%20not%20static).
- Berglund, A., & Leifer, L. (2013). Why we prototype! An international comparison of the linkage between embedded knowledge and objective learning. *Engineering Education*, 8(1), 2-15.
- Caldwell, B., Cooper, M., Reid, L. G., Vanderheiden, G., Chisholm, W., Slatin, J., & White, J. (2008). Web content accessibility guidelines (WCAG) 2.0. WWW Consortium (W3C), 290, 1-34.
- Cameron, S. A., & Sadd, B. M. (2020). Global trends in bumble bee health. *Annual review of entomology*, 65, 209-232.
- Cech, R. & Tudor, G. (2007). *Butterflies of the East Coast: An Observer's Guide* 978-0-69109-055-9
- Cheng, V. W. S., Davenport, T., Johnson, D., Vella, K., & Hickie, I. B. (2019). Gamification in apps and technologies for improving mental health and well-being: systematic review. *JMIR mental health*, 6(6), e13717.
- Chowdhury, S., Fuller, R. A., Dingle, H., Chapman, J. W., & Zalucki, M. P. (2021). Migration in butterflies: a global overview. *Biological Reviews*, 96(4), 1462-1483.

- Cybersecurity and Infrastructure Security Agency. (n.d.). Update Software. Cybersecurity and Infrastructure Security Agency. Retrieved 2 23, 2024, from <https://www.cisa.gov/secure-our-world/update-software>
- ECSA (European Citizen Science Association). (2015). Ten Principles of Citizen Science. <https://doi.org/10.17605/OSF.IO/XPR2N>.
- Eveleigh, A., Jennett, C., Blandford, A., Brohan, P., & Cox, A. L. (2014, April). Designing for dabblers and deterring drop-outs in citizen science. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.
- Ghazanfar, M., Malik, M. F., Hussain, M., Iqbal, R., & Younas, M. (2016). Butterflies and their contribution in ecosystem: A review. *Journal of Entomology and Zoology Studies*, 4(2), 115-118.
- Grace, J. (2016, January 15). Biodiversity Critical to Maintaining Healthy Ecosystems. United States Geological Survey. Retrieved March 15, 2024, from <https://www.usgs.gov/news/biodiversity-critical-maintaining-healthy-ecosystems>
- Inouye, D. (n.d.). Bumblebees (Bombus spp.) - Pollinators. USDA Forest Service. <https://www.fs.usda.gov/wildflowers/pollinators/pollinator-of-the-month/bumblebees.shtml>
- Irwin, E., & Mawdsley, J. (n.d.). How many species of native bees are in the United States? | U.S. Geological Survey. USGS.gov. <https://www.usgs.gov/faqs/how-many-species-native-bees-are-united-states>
- Jefferson, L., & Harvey, R. (2007, April). An interface to support color blind computer users. In *Proceedings of the SIGCHI conference on Human factors in computing systems*.

- Kim, S. (2019). Why should GUI designers create visual novelties?: The mechanisms of first impression formation in user experience. *Archives of Design Research*, 32(2), 31-43.
- Kreitmair, K. V., & Magnus, D. C. (2019). Citizen science and gamification. *Hastings center report*, 49(2), 40-46.
- Lagoze, C. (2014). eBird: curating citizen science data for use by diverse communities.
- Lawrence, M. (2022, June 10). Protecting Pollinators Critical to Food Production. National Institute of Food and Agriculture. <https://www.nifa.usda.gov/about-nifa/blogs/protecting-pollinators-critical-food-production>
- LeBuhn, G., Luna, J. (2021). Pollinator decline: what do we know about the drivers of solitary bee declines?. *Current Opinion in Insect Science*. Volume 46. Pages 106-111.
<https://doi.org/10.1016/j.cois.2021.05.004>.
- Luna, S., Gold, M., Albert, A., Ceccaroni, L., Claramunt, B., Danylo, O., ... & Sturm, U. (2018). Developing mobile applications for environmental and biodiversity citizen science: considerations and recommendations. *Multimedia tools and applications for environmental & biodiversity informatics*, 9-30.
- McKinney, A. M., & Goodell, K. (2011). Plant–pollinator interactions between an invasive and native plant vary between sites with different flowering phenology. *Plant Ecology*, 212, 1025-1035.
- Moreno-Mateos, D., Alberdi, A., Morriën, E., van der Putten, W. H., Rodríguez-Uña, A., & Montoya, D. (2020). The long-term restoration of ecosystem complexity. *Nature Ecology & Evolution*, 4(5), 676-685.
- Pecorale, A., & Poulos, J. (2022). Expanding Beecology Project’s Web Application for Butterfly Observations. *Major Qualifying Project. Worcester Polytechnic Institute*.

- Ryder, E. F., Ruiz, C., Weaver, S., & Gegeer, R. J. (2020). Choosing Your Own Adventure: Engaging the New Learning Society through Integrative Curriculum Design. *EPiC Series in Education Science*, 3, 188–199.
- Schnotz, W., & Kürschner, C. (2007). A reconsideration of cognitive load theory. *Educational psychology review*, 19, 469-508.
- Semmens, B. X., Semmens, D. J., Thogmartin, W. E., Wiederholt, R., López-Hoffman, L., Diffendorfer, J. E., Pleasants, J. M., Oberhauser, K. S., & Taylor, O. R. (2016). Quasi-extinction risk and population targets for the Eastern, migratory population of monarch butterflies (*Danaus plexippus*). *Scientific reports*, 6, 23265.
<https://doi.org/10.1038/srep23265>
- Shrestha, B. R., Timsina, B., Münzbergová, Z., Dostálek, T., Gaudel, P., Basnet, T. B., & Rokaya, M. B. (2020). Butterfly-plant interactions and body size patterns along an elevational gradient in the Manang region of central Nepal. *Journal of Mountain Science*, 17(5), 1115-1127.
- Silvertown, J. (2009). A new dawn for citizen science. *Trends in ecology & evolution*, 24(9), 467-471.
- Stein, L. (2008). Towards a cyberinfrastructure for the biological sciences: progress, visions and challenges. *Nature Reviews Genetics*, 9, 678-688
- Stout, J. C., & Tiedeken, E. J. (2017). Direct interactions between invasive plants and native pollinators: evidence, impacts and approaches. *Functional Ecology*, 31(1), 38-46.
- Strano, S. (2023). Computational and Data Science Tools for Environmental Science: Butterflies. *Major Qualifying Project. Worcester Polytechnic Institute.*

- Sullivan, B. L., Wood, C. L., Iliff, M. J., Bonney, R. E., Fink, D., & Kelling, S. (2009). eBird: A citizen-based bird observation network in the biological sciences. *Biological conservation*, 142(10), 2282-2292.
- 'The Beecology Project'. <https://beecology.wpi.edu/>.
- Thimbleby, H. (1990). *User interface design*. ACM.
- Thorp, R. W., & Shepherd, M. D. (2005). Profile: Subgenus *Bombus*. Red list of pollinator insects of North America, 457-480.
- Tinati, R., Luczak-Roesch, M., Simperl, E., & Hall, W. (2017). An investigation of player motivations in Eyewire, a gamified citizen science project. *Computers in Human Behavior*, 73, 527-540.
- Tufte, E. (2001). The visual display of quantitative information (2nd ed.) *Cheshire, CT: Graphics Press*.
- Unwin, A. (2020). Why Is Data Visualization Important? What Is Important in Data Visualization? *Harvard Data Science Review*, 2(1).
- <https://doi.org/10.1162/99608f92.8ae4d525>
- What is taxonomy?. (2010). *Convention on Biological Diversity*.
- <https://www.cbd.int/gti/taxonomy.shtml>
- Whipple, S., Bowser, G. (2023). The Buzz around Biodiversity Decline: Detecting Pollinator Shifts using a Systematic Review. *ISCIENCE*. doi: <https://doi.org/10.1016/j.isci.2023.108101>.