

Expanding Beecology Project's Web Application for Butterfly Observations

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

By:

Aidan Pecorale
Jared Poulos

Project Advisors:

Professor Carolina Ruiz
Department of Computer Science
Bioinformatics and Computational Biology Program

Professor Elizabeth F. Ryder
Department of Biology and Biotechnology
Bioinformatics and Computational Biology Program

Date:

April 28th, 2022

This report represents the work of two 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. For more information about the projects program at WPI, please see:

<https://www.wpi.edu/Academics/Projects>.

Table of Contents

Abstract	iii
Chapter 1: Introduction	1
Chapter 2: Background	3
2.1 The 'Beecology Project'	3
2.1.1 Pollinator Decline.....	3
2.1.2 The 'Beecology Project'	4
2.1.3 Bumblebee Data Collection and Analysis	4
2.1.4 Bio-CS Bridge	5
2.2 Butterflies	6
2.2.1 Butterflies as Pollinators	7
2.2.2 Butterfly Behaviors	7
2.2.3 Butterfly Approachability	8
2.3 Butterfly Identification	8
2.3.1 Taxonomy and Manual Identification	8
2.4 iNaturalist.....	14
2.4.1 The iNaturalist Computer Vision Application Programming Interface (API).....	14
2.4.2 Bias Concerns	15
Chapter 3: Methodology and Results	16
3.1 Project Goals.....	16
3.2 Webapp Design and Modifications	16
3.2.1 Overview of the Preexisting Webapp and Log Creation Flow	16
3.2.2 Extensions to the Webapp Log Creation Flow	21
3.2.3 Server API Modifications	26
3.2.4 Database Expansion	27
3.3 Species Identification using iNaturalist	30
3.3.1 Flower Identification	30
3.3.2 Butterfly Identification	31
3.4 Creating Manual Butterfly Identification.....	32
3.5 Introducing Butterfly Terminology and Tutorial.....	35
Chapter 4: Discussion.....	37
4.1 Review Identification Methods	37
4.2 Final Thoughts on the Project.....	38

4.3 Project Challenges	39
4.4 Future Work.....	39
4.4.1 Adding More Species	40
4.4.2 Creating More Data Visualization Tools	41
4.4.3 Comparing iNaturalist and Manual Identification	41
References.....	43
Appendix A: Butterfly Identification Flowchart.....	45

List of Figures

Figure 1: Lifecycle of a Butterfly	6
Figure 2: Anatomy of a Butterfly and Detailed Anatomy of Butterfly Wings	9
Figure 3: Butterfly from <i>Papilionidae</i> Family.....	10
Figure 4: Rock Grayling (<i>Hipparchia hermione</i>).....	11
Figure 5: Antennae Morphology.....	12
Figure 6: Taxonomy of Butterfly Families	13
Figure 7: Pre-Existing Webapp Flow	17
Figure 8: Pages for Pre-Existing Log Creation Flow.....	18
Figure 9: Extensions to Log Creation Flow.....	23
Figure 10: Pre-Existing Database Schema.....	28
Figure 11: Updated database schema.....	29
Figure 12: Image of iNaturalist Flower Identification Page	31
Figure 13: Section of Flowchart for Butterfly Identification.....	33
Figure 14: Manual Identification Filter Page and Filter Results.....	34
Figure 15: Examples of Popover Text Feature Identification Guides	36

List of Tables

Table 1: Log, LogRecord, and Record Fields for Bees and Butterflies.....	23
--	----

Abstract

The Beecology Project tracks pollinator populations using a webapp where citizen scientists can upload and submit videos of bumblebee-flower observations and identify the bumblebee and flower species. We extended the Beecology database, server, and webapp to include butterfly-flower observations. We developed a novel method for manually identifying butterfly species and added 120 butterfly species to the database. We also added functionality for automated butterfly and flower identification using the iNaturalist's computer vision API.

Chapter 1: Introduction

Pollinators are a critically important part of the ecosystem, as they are responsible for playing a role in the reproductive success of more than 75% of wild cultivated plant species (Thomann et al., 2013). Unfortunately, the last decade has seen a decline in the number of pollinators due to climate change and increased use of pesticides in agriculture (Potts et al., 2010). As researchers are observing pollinators more than ever, the Beecology Project focuses primarily on tracking and storing observations from the *Bombus* genera more commonly known as bumblebees. Utilizing citizen-collected data the Beecology database functions as a source for analyzing bee-flower interactions. The Beecology Project also serves as a resource for the Bio-CS Bridge, a National Science Foundation funded project that is creating a high school curriculum to teach students computational tools to analyze biological data (Bio-CS Bridge, 2021).

In an effort to expand the number of pollinators included, increase observations, and improve approachability for students and citizen scientists on the Beecology Project, we expanded the Beecology Project to include butterfly observations as well. Often known only for their beauty, butterflies are important pollinators for many plant species. Unlike bees which have specialized features for pollen transfer, butterflies pick up pollen on their bodies and wings when they sip nectar from a flower (Ghazanfar et al., 2016). Many butterflies migrate as many as 3,000 miles, which allows for cross pollination over large distances and increases flower biodiversity (Ghazanfar et al., 2016). The presence of butterflies can be an indicator of a healthy ecosystem, but like bees, butterflies' natural habitats are dwindling, and the effects of climate change and pesticide use are taking a toll on their population.

This project overhaul required expanding the existing Beecology database, server and web application as well as finding new ways for users to identify their butterflies entries. Utilizing the preexisting database and server we were able to add tables and communication functions for storing butterfly family/species information and butterfly-flower interactions. After expanding the applications database and user interface we needed to determine a way for users to identify their butterflies. We determined the best approach would be a combination of an automatic computer vision identification system, utilizing a third-party species identification API in combination with a manual identification process. The API we selected is from a site called iNaturalist which gathers data from citizen scientists about a large number of plant and animal species, and is capable of identifying those species (Loarie, 2021). The iNaturalist API utilizes a machine vision model to return the closest species matches based on the user's inputted photos. The manual identification procedure was developed by identifying and filtering each species by specific features. After adding both of these subcomponents to the butterfly log creation we needed to stylize and modify some other subcomponents of the webapp to account for the new data types. Lastly, the new webapp was tested and published, so citizen scientists could begin uploading observations.

Chapter 2: Background

2.1 The 'Beecology Project'

2.1.1 Pollinator Decline

The last decade has seen many scientific papers about pollinator decline, some focused on studying the causes of decline and others looking at ways to combat it (Wenzel et al., 2020). There is evidence that suggests the planet is going through its “6th mass extinction, losing between 1 - 10 % of biodiversity per decade, mostly as a result of habitat loss, pest invasion (exotics), pollution, over harvesting and disease” (Klusser & Peduzzi, 2007). One aspect that is particularly dangerous for humanity is the disappearance of pollinators and other species that are critical in the reproduction of plants, as they have drastic impacts on the native ecosystem and food web. Outside of affecting natural habitats, pollinators are vital to agriculture, contributing to the pollination of more than 70% of global crops or 87% of the wild flowering plants (Wenzel et al., 2020).

One important pollinator is the bumblebee. One of the reasons bumblebees and other bees are so effective at pollination is their hairy body, which attracts pollen through electrostatic forces, allowing them to effectively carry and spread pollen better than any other animals (Westerkamp, 1996). The number of bees is declining rapidly as a result of growing urbanization (habitat loss) and pesticides usage/ chemical residue (Klusser & Peduzzi, 2007). This decline results in a vicious cycle that causes loss of suitable habitats because of the decreased pollination; this in turn contributes to habitat loss which causes the pollinator numbers to decline again (Potts et al., 2010). Without measures being put in place to help preserve these habitats and a decreased use of the pesticides and chemicals that cause the collapse in the first place, colonies of bees have little chance of recovering to the numbers they were before.

While the effects of pollinator decline can be devastating, in order to enact policies to help these species it is crucial that we educate more people about the problem. More recently, pollinator decline research has shifted from just focusing on pollinators to include pollinator systems – pollinators and the plants they rely on for food as well as pollinate (Vanbergen, 2013). As a result of the lack of information on pollinator-plant interactions and a desire to educate more individuals about the problem The Beecology Project was formed.

2.1.2 The ‘Beecology Project’

Established in 2016, the Beecology Project was created in response to the ongoing pollinator decline across New England. In an effort to educate students and the local population on the declining bumblebee population, the Beecology Project recruits citizen scientists to record wild bumblebee-flower interactions and submit them through its mobile web application. Before our MQP, the Beecology Project was mainly focused on bumblebee-flower interactions found in Massachusetts but hopes to expand to more regions and pollinators. Aside from being a place to submit bumblebee-flower interactions, the web application consists of other components for bumblebee identification (the Beedex), data analysis, and visualization as well as simulation tools (Beecology Project, 2021). The Beecology Project is also closely partnered with the Bio-CS Bridge, a team developing a curriculum designed to encourage students to use computational tools to analyze biological data, using biology as a gateway to learn computer science concepts (Bio-CS Bridge, 2021).

2.1.3 Bumblebee Data Collection and Analysis

When a user records a bee-flower observation on the Beecology app, they provide details including location, time/date, species of bee, species of flower, behavior and more. This data is

collated on the Beecology Project's database with more than 9,300 other observations to-date, including some historical data on bee populations from as far back as the late 19th century. This quantity of data and level of detail is useful for analyzing and visualizing data by groups and finding trends over time. The site allows a user to download data from the database utilizing a variety of filters for dates, bumblebee species and behavior. The user can then analyze this data using several analysis tools: the Location Map Tool lets a user visualize bee density locations plotted on a map, the Floral Network Tool plots relationships between flower name and color to bee species interactions, the Season Chart Tool plots seasonal relationships between bee species, and more. The Beecology Project also has tools for simulations where users can experiment and observe some of the impacts bumblebees have on an ecosystem as well as model the effects invasive plants may have on an ecosystem.

2.1.4 Bio-CS Bridge

The Bio-CS Bridge is a project funded by the National Science Foundation developed by a multidisciplinary team of university professors, graduate and undergraduate students, and high school teachers and students. The Bio-CS Bridge seeks to develop a high school curriculum that teaches students to apply computer science concepts and technologies to biology problems. The curriculum involves both the students and the teachers in gathering real world data and building computer programs to analyze and draw conclusions from that data. The program also demonstrates the efficacy of a team of biology and computer science specialists in blending their areas of interest into a shared curriculum (Bio-CS Bridge, 2021).

2.2 Butterflies

Order Lepidoptera is the second largest order of insects behind only Coleoptera (beetles) totaling more than 180,000 species between butterflies and moths worldwide (Ghazanfar et al., 2016). Butterflies and moths start their lives hatching from eggs as caterpillars (larva) and feed on leaves (Figure 1). When a caterpillar has consumed enough nutrients it forms a chrysalis (pupa), which undergoes complete metamorphosis to emerge as a butterfly (Cinici, 2013). Most butterflies live for 2 to 4 weeks, although some migrating species (such as the Monarchs) can live for months (Feng, 2021). As adults, butterflies are responsible for feeding, mating, and laying eggs to continue their fragile life cycle. Moths are often a little more resilient as some species of adult moths are able to survive for as much as 10 to 12 months where they can carry out the same functions (Feng, 2021). Butterflies and moths are most commonly found in the summer months, but some species are visible in early spring and fall as well. Locations with warmer weather may also see butterflies emerging as late as the winter (Feng, 2021).

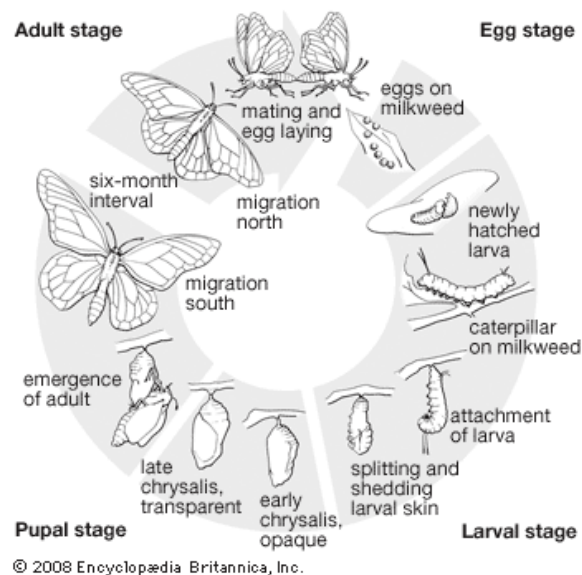


Figure 1: Lifecycle of a Butterfly. Taken from Encyclopædia Britannica.

2.2.1 Butterflies as Pollinators

Butterflies primarily feed on nectar-producing flowers and rotting fruits. In the process of feeding on nectar, butterflies pick up pollen on their bodies (primarily legs and head) and wings (Ghazanfar et al., 2016). The resulting interaction leads to a symbiosis where butterflies can feed in exchange for helping the plants reproduce through pollination. Despite lacking specialized body architecture of bees for holding and carrying pollen, butterflies serve as the primary pollinators for *Brassica* family, legumes, macadamia nuts, mint, peacock flowers, and many more (Hooks & Espíndola, 2017).

2.2.2 Butterfly Behaviors

While our main focus is observing butterflies' role as pollinators, butterflies have a few common behaviors not necessarily related to pollination: puddling, nectaring, patrolling, mating, and ovipositing. Puddling is a behavior where butterflies sit on wet or damp ground and drink the water in search of salts and other minerals for nutrition (Cech & Tudor, 2007). Nectaring is the action where butterflies use their proboscis to suck nectar from flowers; during this action pollen is being transferred to the butterfly's body and legs as it lands on the flowers. Repeated nectaring on the same plant species will result in pollen transfer that aids the plants in reproduction (Prasifka et al., 2018). Similar to bumblebees some butterfly species display strong preferences for what flower species they nectar on (Cech & Tudor, 2007). Patrolling can be closely related to finding food and resources while also looking for a mating partner. Mating and ovipositing are both part of the butterfly's reproduction cycle (Cech & Tudor, 2007). Similar to nectaring, many species of butterflies are often very particular about what types of plants they will lay their eggs on. For example, the Karner Blue butterfly will only feed and lay eggs on the endangered wild

blue lupine leaves; because of this delicate relationship between ecosystems and species it is important to work to protect the ecosystems as well (Candeias, 2021). Because these relationships are often not obvious it is important that we study the interactions between the pollinators and plants together as a pollination system rather than focusing on just one or the other.

2.2.3 Butterfly Approachability

While expanding the Beecology Project to more types of pollinators was always a goal, butterflies were chosen in part because they can be a more approachable entry point for younger citizen-scientists. Butterflies also have the advantage of often being easier to identify at a distance due to their larger size, bright colors and distinctive wings. Despite bumblebees' low rates of stinging humans, younger users and people with severe allergies can be discouraged from contributing to the application. By expanding to butterflies, a less threatening insect, we can better include these user groups and increase the number of observations on the app. Additionally, more observations can help improve pollinator research and be used to expand the food web complexity for the Bio-CS Bridge.

2.3 Butterfly Identification

2.3.1 Taxonomy and Manual Identification

In order to be able to identify butterfly species, citizen scientists and students need to understand the basics of butterfly anatomy and terminology (Figure 2). Like other insects, butterflies' central body structures include a head, thorax, and abdomen (Cech & Tudor, 2007). Features like antennae and proboscis extend from the head and can be used for identification of

some butterfly families, but most commonly butterflies are identified by their color, patterns and wing shape. Butterfly wings are four separate sections composed of 2 forewings and 2 hindwings with one of each on each side. Butterfly wings are composed of two chitinous layers covered in scales and hair (Cech & Tudor, 2007). The dorsal (top) side and ventral (bottom) side of the wings often have different colors and patterns used for attracting mates and hiding from or scaring predators. For the purpose of identification, a lot can be told from the wing shape and size, or from the presence of tails as shown on the hindwing in Figure 2. We will examine some of these features later as they relate to specific families and critical points in manual identification.

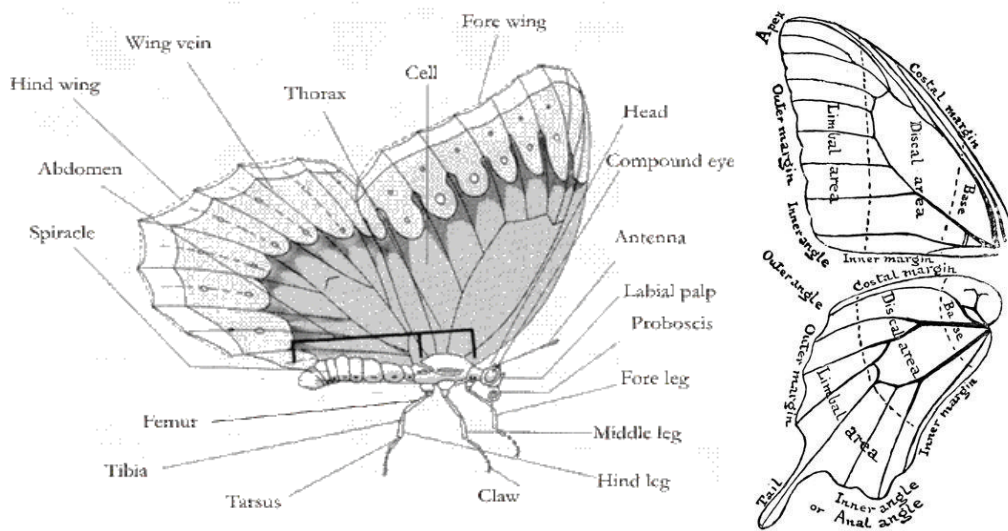


Figure 2: Anatomy of a Butterfly (Left) and Detailed Anatomy of Butterfly Wings (Right).
 Images taken from Sky Meadows and Woodland Honey.

Characteristics of the seven families of butterflies are described in “Butterflies of the East Coast” by Cech and Tudor (2007) and summarized below.

Papilionidae:

Papilionidae, also called Swallowtails, is one of the most easily recognizable butterfly families. Species in this family include some of the largest butterflies, with wingspans ranging from 3.5 to 6.5 inches, and have distinct tails on their hindwings (Figure 3). There are only six species of Swallowtails in Massachusetts: Black, Canadian Tiger, Eastern Tiger, Giant, Pipevine and Spicebush Swallowtail.



Figure 3: Butterfly from *Papilionidae* Family. Note distinct tails protruding from its hindwing. Image taken from the Beecology Website.

Pieridae:

The *Pieridae* family has two main sub-families, Whites and Sulphurs; with simple color palettes, these species are easy to distinguish from more complex families. They are often unassuming with small black markings on the tips of yellow, white, or yellow-green wings. While only six species of *Pieridae* reside in Massachusetts, they are one of the most common butterflies to see.

Nymphalidae:

The *Nymphalidae* family contains a wide variety of butterfly species. Many members contain oranges, browns, and blacks, but these colors are not exclusive to the family.

While Monarchs are one of the most recognizable butterflies in the *Nymphalidae* family, the family hosts other equally beautiful species, from the Speckled Fritillaries to the unique Commas. Another distinctive feature of the *Nymphalidae* family is their legs, often called brushfeet; the two front legs of *Nymphalidae* butterflies are not used for standing but can be found tucked up against the head or thorax of the butterfly as seen in Figure 4. Approximately forty species of *Nymphalidae* butterflies exist in Massachusetts.



Figure 4: Rock Grayling (*Hipparchia hermione*). Note reduced forelegs. Galicia, Spain.
Photograph by Raymond JC Cannon.

Riodinidae:

Riodinidae, also called Metalmarks, are often distinguished by their small metallic looking spots on their wings. Massachusetts is not home to any *Riodinidae* as they typically reside in warmer climates.

Lycaenidae:

Lycaenidae, also called Gossamer-winged butterflies, is the second largest family of butterflies. The family is comprised of seven subfamilies including, Blues, Hairstreaks, Coppers and Harvesters. Small in size, some subfamilies have distinguishing features like small tails on their hindwings or distinct patterns such as buckeyes.

Hesperiidae:

Hesperiidae, often called Skippers, can sometimes be seen as an outcast group that lies somewhere between “true” butterflies and moths. The name ‘Skippers’ is derived from their erratic movements usually consisting of short flights between low lying plants. Though they are often dull and brown colored, another distinguishing feature of Skippers is their hooked antennae shown in Figure 5. There are 46 species of Skippers in Massachusetts.

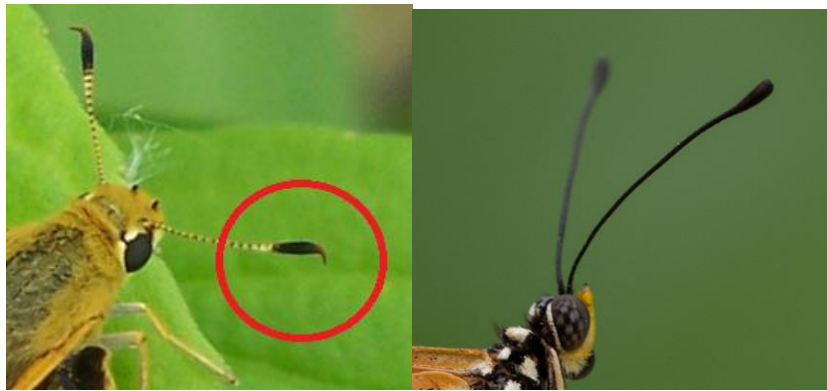


Figure 5: Antennae Morphology. Left, example of a hooked antenna from the *Hesperiidae* Family) Right, example of clubbed antennae. Images taken from the Beecology Project and Butterflies of Singapore.

Moths:

Moths, often considered just nocturnal butterflies, have other distinguishing features as well. Unlike “true” butterflies, moths’ antennas are bushy and hairy rather than slim and clubbed. Moths also have an extra piece of anatomy called the frenulum, which fastens the forewing and hindwing together so they work in unison during flight (Wildlife Insight, 2022). While the number of moth species is unknown for Massachusetts, moths make up around 90 percent of the Order Lepidoptera, to butterflies 10 percent, so using

the number of butterfly species we can estimate that more than 1000 species of moths reside in the state.

While specific information on when and where these families split from each other is largely unknown, some studies have tried to determine their relationship utilizing DNA and RNA sequencing (Kim et al., 2010). One of the most widely supported taxonomy relationships is the (*Hesperiidae* + (*Papilionidae* + (*Pieridae* + (*Nymphalidae* + (*Lycaenidae* + *Riodinidae*)))))) as shown in Figure 6.

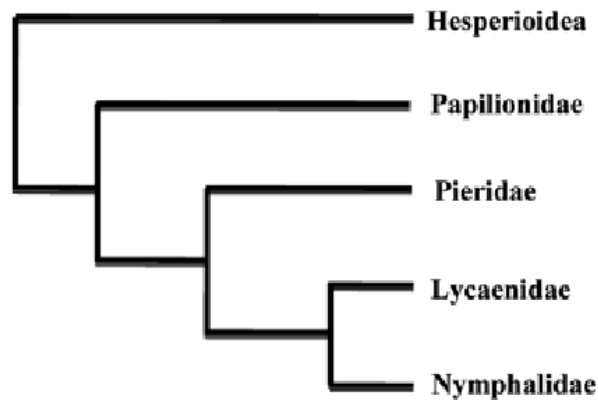


Figure 6: Taxonomy of Butterfly Families. (Kim et al., 2010).

While identifying any single species takes some practice, using the shape, color, pattern and other features to narrow down a species into a family or subfamily is achievable for anyone willing to try. This of course is not the only option for identifying butterflies as advancements in computer vision and machine learning have made computers a viable option for species identification. We will explore this idea more in the next section on the iNaturalist API and its abilities.

2.4 iNaturalist

The iNaturalist website and application were created to allow individuals to identify plant and animal species while also compiling that data for researchers' use. The site began as a Master's final project at UC Berkeley in 2008, becoming an LLC in 2011 and eventually becoming a joint initiative of the California Academy of Sciences and the National Geographic Society in 2017. The iNaturalist community has grown to over a million users identifying species and creating research-quality data (Loarie, 2021).

Initially, users would submit photos of a species and if they could not identify the species, the image would be shared with other members of the community for manual identification. This process took an average of 18 days to inform the user of their findings; thus, a better process was needed. The iNaturalist team developed an automated solution using machine learning (Shepard, 2022). The iNaturalist team utilized TensorFlow, an open-source machine vision library based on neural networks to develop their model (Goldsborough, 2016). The iNaturalist neural network was initially trained on images of approximately 10,000 species for which 20 distinct, high quality iNaturalist observation photos of that species were available. The machine vision algorithm is continuously being updated as more iNaturalist users submit high quality photos (Shepard, 2022).

2.4.1 The iNaturalist Computer Vision Application Programming Interface (API)

In addition to their own site, the iNaturalist Computer Vision API is available for integration into approved applications via RapidAPI (C. Seltzer, personal communication, July 27, 2021). After receiving a photo, the API is capable of returning the top ten closest matches for the species. API calls can also be 'seeded' to only look for matches within a certain taxonomic

branch, such as the Order Lepidoptera or the Kingdom Plantae. This feature is helpful for identifying either a butterfly or plant when both species are focused in one image.

2.4.2 Bias Concerns

One concern about relying on the iNaturalist API for species identification is that the iNaturalist identification model may be biased by the data used to train it. The uneven distribution of data means that a computer vision model is biased towards the set of species with an acceptable number of professional photos to train on. The Computer Vision API will attempt to make an identification whether or not it can correctly identify the subject of the image, so there is a likelihood of false positives if a species of butterfly or plant has too few observations in the iNaturalist database. The computer vision model can be used to correctly identify the species in about 85% of all unidentified iNaturalist observations, even though it is only trained to detect 10,000 out of the 2 million species iNaturalist users have observed (Shepard, 2022). This is due to biases in which species are recorded by iNaturalist users. An analysis of the approximately 74 million observations recorded before July 2021 found several indications that the data was not representative of reality (Di Cecco et al, 2021). The application and site are primarily used in North America and between the months of May and September, however, the count of submitted observations soar at specific iNaturalist ‘challenge’ events. Furthermore, there is a disproportionate amount of data from developed and deciduous areas of the United States, and environments such as grasslands and shrublands are underrepresented. For example, a total of 38% of observations in the United States in the iNaturalist database were in developed areas, which make up only 5% of the country’s land area (Di Cecco et al, 2021).

Chapter 3: Methodology and Results

3.1 Project Goals

The goal of this project was to expand the Beecology Web Application to include butterfly-flower observations. This objective required adding to and adapting various aspects of the existing application framework as well as exploring options for butterfly identification and user education. On the backend of the application, we had to introduce tables to store butterfly data on the database, make changes to the server to send and receive new butterfly classes of data. On the front end of the application, we had to educate the users about butterflies, create a method for users to be able to identify their butterflies using either iNaturalist Computer Vision API or a manual identification key, and finally make the app pages where users submit their observations.

3.2 Webapp Design and Modifications

3.2.1 Overview of the Preexisting Webapp and Log Creation Flow

The Beecology web application serves as an interface for citizen scientists to upload their bee observations. Since we utilized the existing framework for uploading butterfly observations it is important to understand the preexisting process for bee observations. For a summary of the flow and images of the old pages, Figure 7 shows each step in the old flow and Figure 8 shows each page in the process. Upon clicking ‘Collect Data’ to begin the gathering process, the user is given the choice to upload either a video or an image of the bee they observed. Once they have uploaded their visual data, they are taken through a flow in which they must indicate the color

pattern on their bee’s abdomen, thorax, and head. If they uploaded a video, they also must capture a frame displaying each of these. From the combination of those three patterns, the app determines the species and sex of the bee. Next the user is brought to a screen where they use a dropdown to manually select the species of flower the bee is interacting with and whether the bee is gathering pollen or nectar. Again, a frame of the flower must be captured if the user uploaded a video. The next page allows the user to select the date of their observation and its location. The user is then given the option to provide comments on their observation if they choose, and then are shown the completed log that they have created. At this point, the log may be missing either the flower species or the location if the user selected ‘SET LATER’ for those fields earlier in the process.

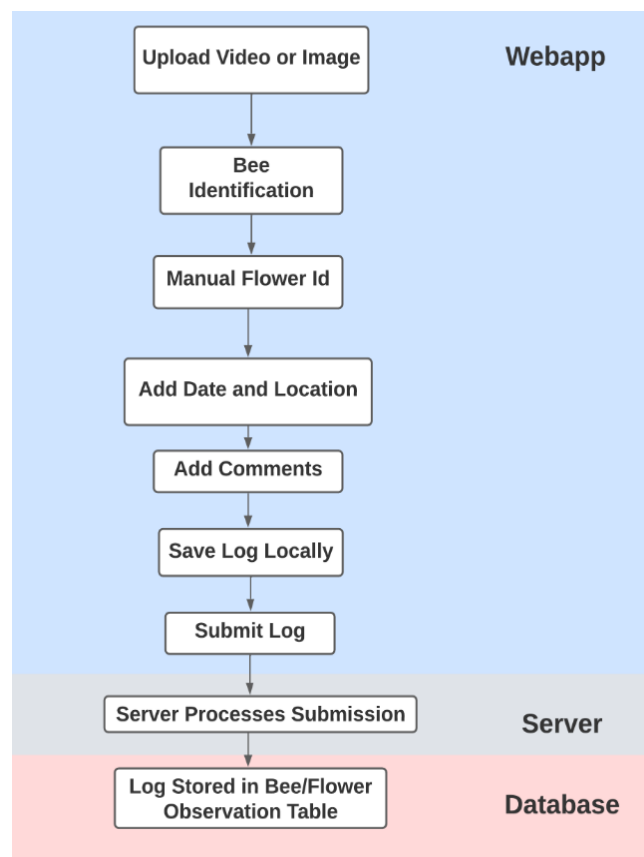
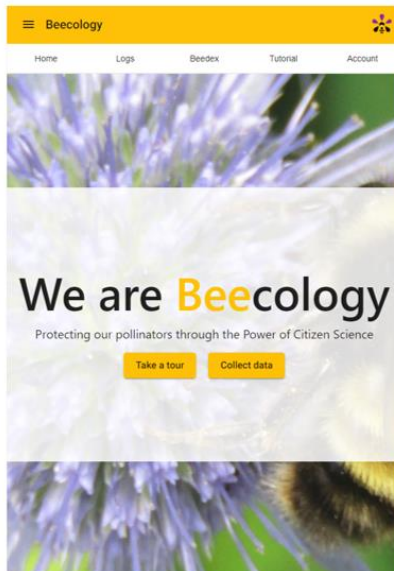
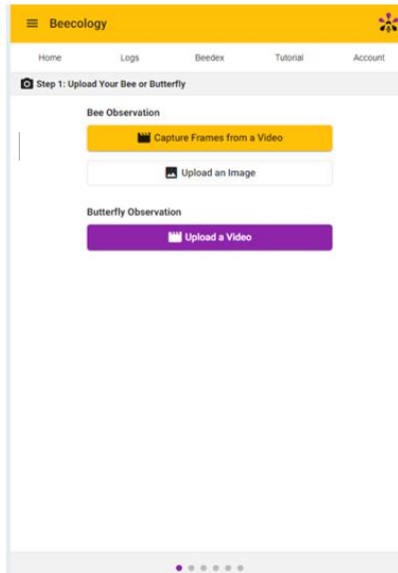


Figure 7: Pre-Existing Webapp Flow. Flow depicts how a user will move through the webapp creating a bee-flower observation and submit to the database.



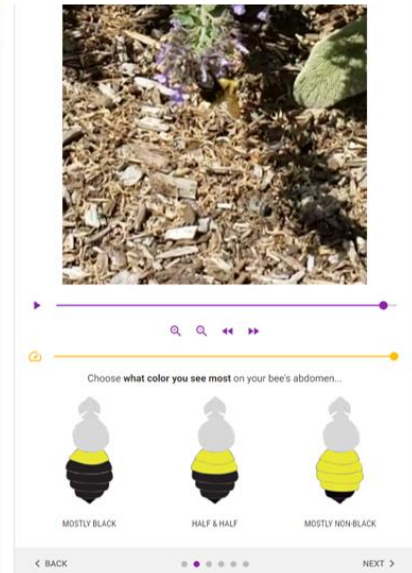
app-landing-page

1. User clicks 'Collect data' on the landing page



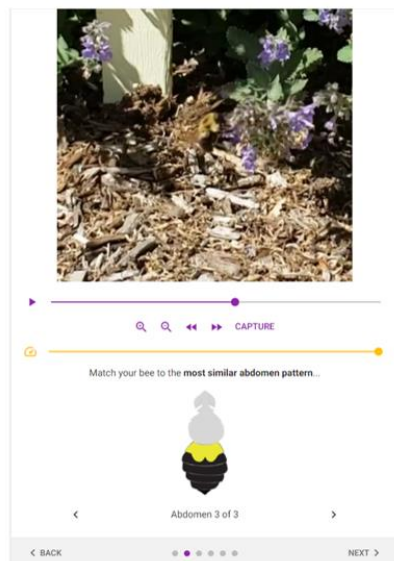
app-home-source-upload

2. User uploads their bee image or video. Each are stored as image data.



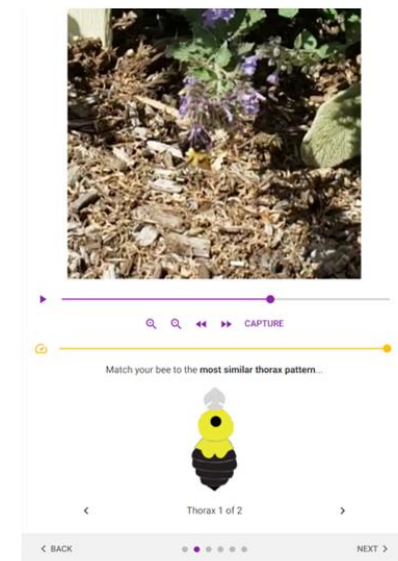
app-home-bee-id-guided (beeStepId = 0)

3. User determines which category of abdomen the bee has. This will determine the pattern choices for the next step.



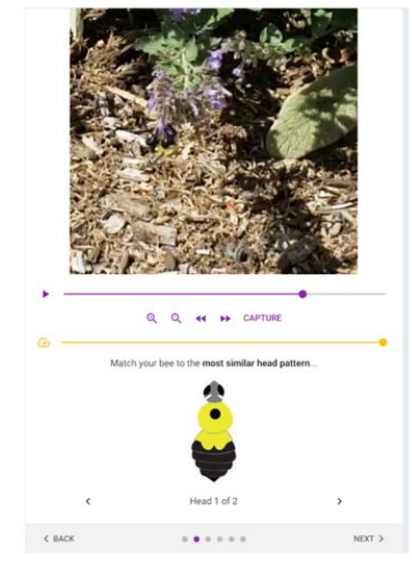
app-home-bee-id-guided (beeStepId = 1)

4. User determines which abdomen pattern the bee has. This choice is stored in a BeeLog, and the frame captured is stored in the image data.



app-home-bee-id-guided (beeStepId = 2)

5. User determines which thorax pattern the bee has. This choice is stored in a BeeLog, and the frame captured is stored in the image data.

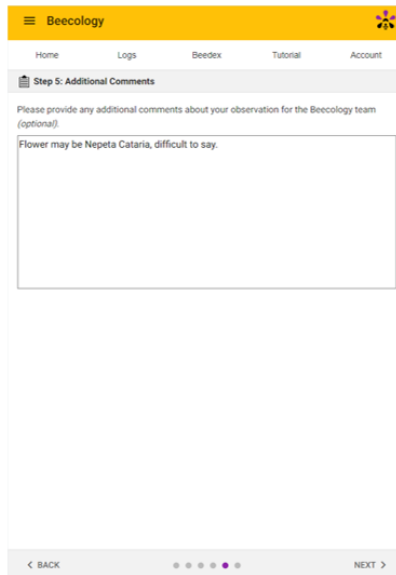


app-home-bee-id-guided (beeStepId = 3)

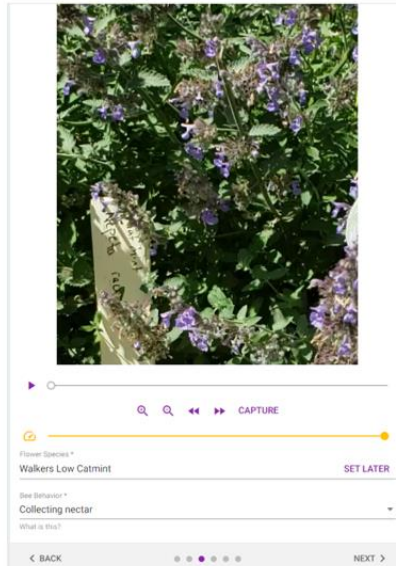
6. User determines which head pattern the bee has. This choice is stored in a BeeLog, and the frame captured is stored in the image data.



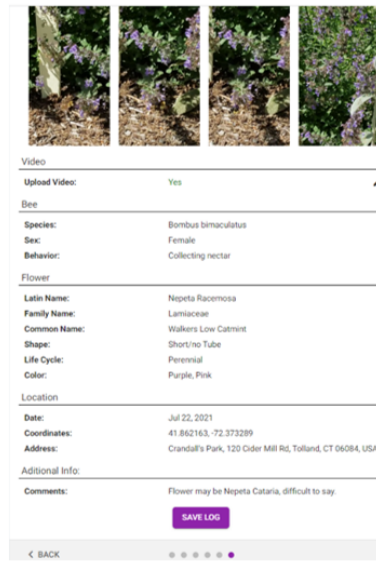
app-home-bee-id-guided (beeStepId = 4)
 7. Webapp determines the species and sex of the bee from the three patterns that the user chose. The species name is stored in the BeeLog as well.



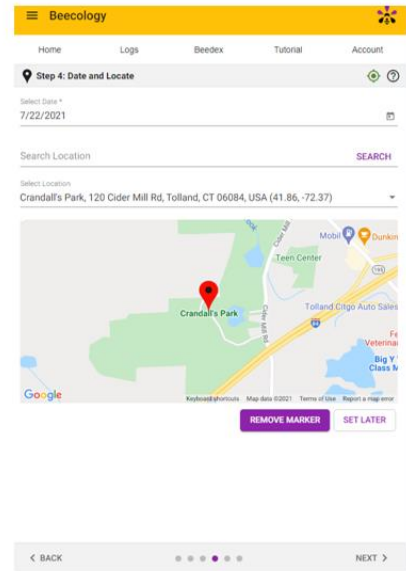
app-home-user-comment
 10. User enters any comments on the observation. These are stored in the BeeLog as well.



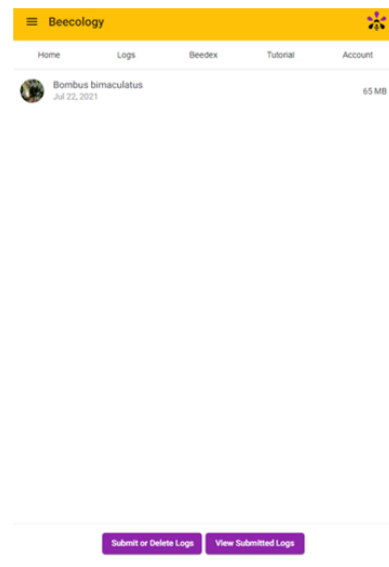
app-home-flower-id-expert
 8. User enters the flower species and the bee's behavior, both of which are stored in the BeeLog. They also capture a frame of the flower stored in the image data.



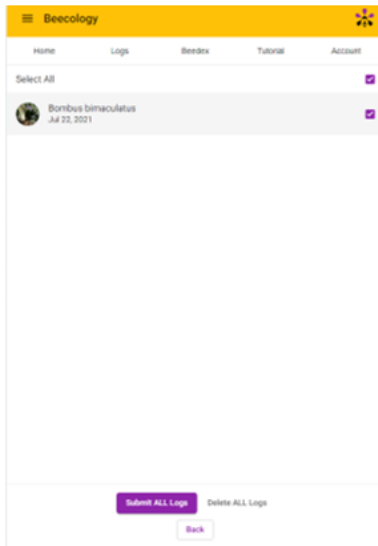
app-home-summary
 11. The user is prompted on whether to upload their video or just the images captured. After this prompt, all image data is added to the media field in the BeeLog object. When 'Save Log' is clicked the BeeLog is saved locally using PouchDB.



app-home-date-location
 9. User enters the date location where they made the observation. These are stored in the BeeLog with the location being stored as longitude and latitude.

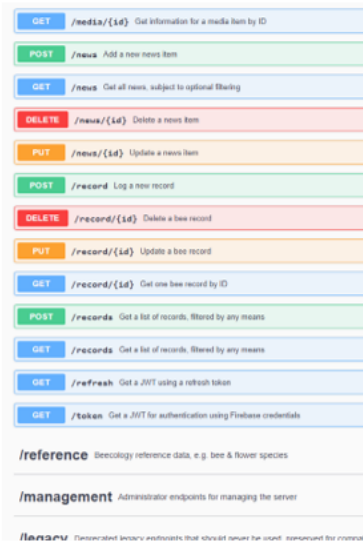


app-logs
 12. User is brought to the logs page. They click Submit or Delete Logs to activate the selection mode.



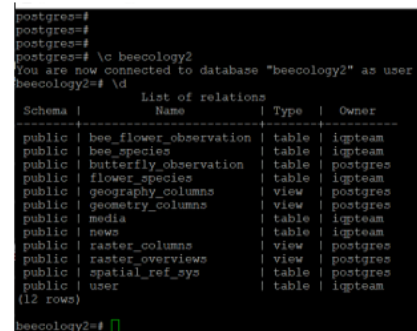
app-logs

13. User can check any logs they have created and then click 'Submit Logs'. To send their stored log to the API server with a POST request.



API Server

14. The API server processes the POST request and converts the data sent along in the body into the format of the database tables for bee_flower_observation and media. It generates unique ids and ensures the media entries have a foreign key that links them to the observation they are associated with.



Postgres Database

15. The observation and associated media are now recorded in the database.

Figure 8: Pages for Pre-Existing Log Creation Flow. Screen shots from webapp going through creating a bee-flower observation. Taken from the Beecology Project Website and database.

These locally saved logs can be submitted from the 'app-logs' page once they have a flower species and a location set. This submission sends the logs to the API server, which passes the log along to the 'bee_flower_observation' table in the database. As part of the submission, the webapp derives additional fields, such as the elevation of the location, before uploading to the server.

On a programming level, After the user clicks on the 'Collect data' button, they are brought to the app-home angular component. For the remainder of the data entering process, the user will click the 'Next' button at the bottom right of the application which will progress through several other angular subcomponents within app-home. The existing infrastructure stored bee data within an object called a 'BeeLog' while being filled out. The Beelog is handled by an

instance of an object called `BeelogService`, an attribute of `app-home` which allows for the subcomponents to easily add information to the log. When the log is completed and the ‘Save Log’ button is clicked, `BeelogService` converts the `Beelog` to another object format called `BeeLogRecord` and calls on an instance of another object called `DatabaseService` to save the record to a local `PouchDB` database.

On the `app-logs` page, all of the user’s `BeeLogRecords` in the local database are loaded and displayed as a list of cards denoted by the bee species and date of the observation. When a user selects a record to submit, the `BeeLogRecord` is converted to another object format called ‘`BeeRecord`’, has additional derived fields added to it, and is then sent to the API server.

3.2.2 Extensions to the Webapp Log Creation Flow

The new butterfly log creation functionality was based heavily on the bee log process, though its implementation required significant changes to several steps, and the addition of a few steps, as shown in Figure 9. The process of updating the log creation aspect of `webapp` was primarily about translating existing bee functionality to work for butterflies and adding the ability to switch between which type of observation was being recorded or submitted. We created objects for `ButterflyLog`, `ButterflyLogRecord`, and `ButterflyRecord` modeled after `BeeLog`, `BeeLogRecord`, and `BeeRecord` with species-specific fields replaced, as shown in Table 1. `BeelogService` was updated to allow it to handle both a `ButterflyLog` and a `BeeLog`, with an attribute that determines which of the two logs is in use. `App-home` and each of its submodules were also modified to fill the `ButterflyLog` if `BeelogService` was set to `butterfly`. The final submodule in that flow, `app-home-summary`, has a submodule of its own called `app-log-summary` which was also modified to be able to display a `ButterflyLog`’s fields in addition to the existing `BeeLog` functionality. The `DatabaseService` almost doubled in size, as a version of each

of the existing functions had to be added to create and update a second PouchDB database that would handle objects of the ButterflyLogRecord type. The functionality to load records into app-logs was updated to also add all ButterflyLogRecords from the local butterfly database in an identical fashion to the BeeLogRecords directly below them. In a similar manner to with BeeRecords, ButterflyRecords can be generated and submitted with DatabaseService.

The major changes made to the flow of the web application involved the addition of pages for butterfly identification, and a choice of the existing flower identification or the new iNaturalist API aided identification. The butterfly identification pages were utilized in contrast to the guided bee pattern selection to fill out the “ButterflyLog” fields. In both the butterfly and flower identification steps, users were also provided the option to select iNaturalist id as a quicker and easier way to fill in the logs. This feature was not added to bee identification because of the small number of species and the effectiveness and ease of existing identification. The log pages, database service, and PouchDB database were all expanded to handle and submit butterfly records. Lastly, other aspects of the application, such as the tutorials and “Beedex” (now “B-dex”) - an index of bee species - were expanded to include information on the new butterflies. Frontend color scheme and design on the new pages were kept as close as possible to the preexisting application.

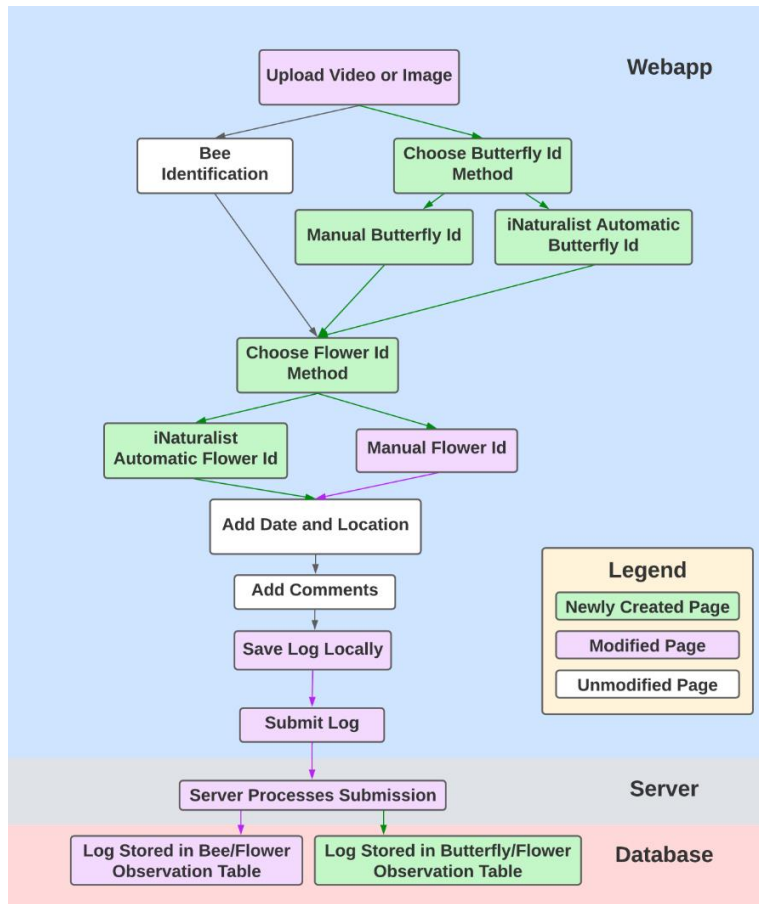


Figure 9: Extensions to Log Creation Flow. Visual representation of webpages that were modified and added to the flow of log creation and submission.

Table 1: Log, LogRecord, and Record Fields for Bees and Butterflies. Tables shows comparison of bee and butterfly data fields for each type of Log or Record.

Uploading Observation		
BeeLog Only	Shared Fields	ButterflyLog Only
_gender: string	_id: number	_wingSize: string
_cartoon: string	_picturesPath: string[]	_wingShape: string
	_isVideo: boolean	_antennae: string
	_date: Date	_dorsalPattern: string[]
	_flowerName: string	_ventralPattern: string[]

	_flowerFamilyName: string	_dorsalPrimaryColor: string[]
	_flowerCommonName: string	_dorsalSecondaryColor: string[]
	_flowerShape: string	_ventralPrimaryColor: string[]
	_flowerLifeCycle: string	_ventralSecondaryColor: string[]
	_flowerColor: string	_tailsLog: string
	_time: number	_speciesGuessArrayLog: string[]
	_videoPath: string	_speciesTopGuess: string
	_city: string	
	_location: string	
	_hasLocation: boolean	
	_appVersion: string	
	_userComments: string	
	_flowerSpeciesGuessArrayLog: string[]	
	_flowerSpeciesTopGuess: string	

Storing in Local Database and Viewing in Logs

BeeLogRecord Only	Shared Fields	ButterflyLogRecord Only
cabdomen: string	time: Date	dorsalPrimaryColor: string[]
cthорax: string	loc: string	dorsalSecondaryColor: string[]
chead: string	cityname: string	ventralPrimaryColor: string[]
gender: string	fname: string	ventralSecondaryColor: string[]
cartoon: string	ffname: string	dorsalPattern: string[]
	fcname: string	ventralPattern: string[]

	fshape: string	antennae: string
	lifecycle: string	wingShape: string
	fcolor: string	tails: string
	beename/butterflyname: string	wingSize: string
	beedictid/butterflyid: string	speciesTopGuess: string
	beebehavior/butterflybehavior: number	speciesGuessArray: string[]
	recordpicpath: string[]	
	recordvidpath: string[]	
	recordpicindex: string[][]	
	hasLocation: boolean	
	appVersion: string	
	userComments: string	
Submitting to the Server		
BeeRecord Only	Shared Fields	ButterflyRecord Only
abdomen_coloration: string	id: string	dorsalPrimaryColor: string[]
thorax_coloration: string	bee_species_id/butterfly_species_id: string	dorsalSecondaryColor: string[]
head_coloration: string	bee_species/butterfly_species: BeeSpecies/ButterflySpecies	ventralPrimaryColor: string[]
gender: enum	flower_species_id: string	ventralSecondaryColor: string[]
	flower_species: FlowerSpecies	dorsalPattern: string[]
	media: string[]	ventralPattern: string[]
	behavior/butterfly_behavior: enum/enum	antennae: enum

	time: string	wing_shape: enum
	submitted: string	tails: enum
	location: GISCoordinate	wing_size: enum
	elevation: number	species_top_guess: string
	closest_city: string	species_guess_array: string[]
	media: string[]	
	how_submitted: enum	
	user_comments: string	

3.2.3 Server API Modifications

The server API which is located on one of Worcester Polytechnic Institute’s servers listens for requests made to the webapp and primarily functions to host the web application online. The server also listens for requests the webapp makes when it needs to transfer data between the webapp and the database or vice versa. Information that is too large to be stored on the web application is stored in the database and the server API acts as the bridge between the two. The first thing that needed to be modified in the server API was the list of the database models for each of the stored tables. This encompassed adding models for each of the new database tables: butterfly species and butterfly-flower observations. We also added enumerated variable types for attributes with a limited number of choices. For example, a butterflyBehavior variable may only have values of ‘Nectaring’, ‘Puddling’, ‘Patrolling’, ‘Perching’, ‘Ovipositing’, ‘Mating’, or ‘Unknown’. Next, functions like getBees, which is called by the webapp with optional filters to return a list of bee species from the database to populate the Beedex, needed to be replicated for butterflies. A similar function called getButterflies was created utilizing new

filters which would allow us to filter butterflies using their families, wing colors, wing size, wing shapes, and other features. These additional filters were useful in sorting and accessing butterfly options for the users performing their manual identification of butterflies so they could filter by specific features as they progressed. The last major change to the server API was related to handling and submitting the new butterfly logs. Once logs are submitted from the webapp the server API checks that they are in the proper format and submits them to the database where they can later be retrieved for data analysis.

3.2.4 Database Expansion

The Beecology database stores more than just users' bee observations. The preexisting schema for the database includes tables for storing information on bee species, flower species, list of users, media elements and most importantly the bee/flower observations (Figure 10).

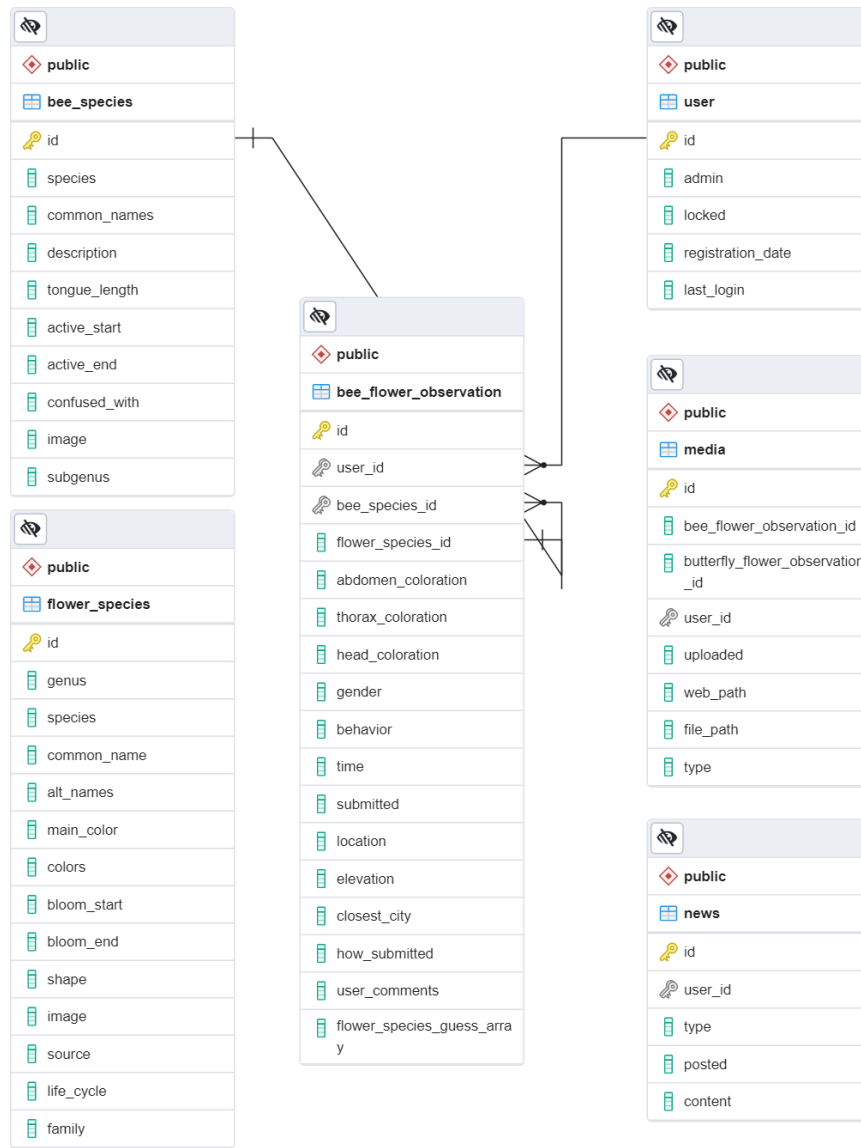


Figure 10: Pre-Existing Database Schema.

The bee species table holds general information about each bee species including their descriptions, common names, active start/end, image and list of bees they are commonly confused with. The bee-flower observation table holds all the geographical data as well as the recorded species and flower information for each observation. These data points were chosen so researchers could filter and pull information from the table for data analysis. To achieve the same clarity and ability to filter data as needed with butterflies, we determined what characteristics

should be recorded and stored for later use. The new “butterfly_flower_observation” table was created to include all the same geographical and flower observation data as well as identifying features of the butterfly which are used in the manual identification process. These features were the primary and secondary colors and patterns of both the dorsal and ventral sides of the wings, as well as the shape of the antenna and wings, the wing size, and whether or not the butterfly has tails. A “butterfly_species” table was also added to the schema to store general information on all the butterfly species in Massachusetts. This complete list of species is utilized in the webapp for displaying the “butterflydex” as well as filtering available options when identifying the species. The new schema can be seen below in Figure 11.

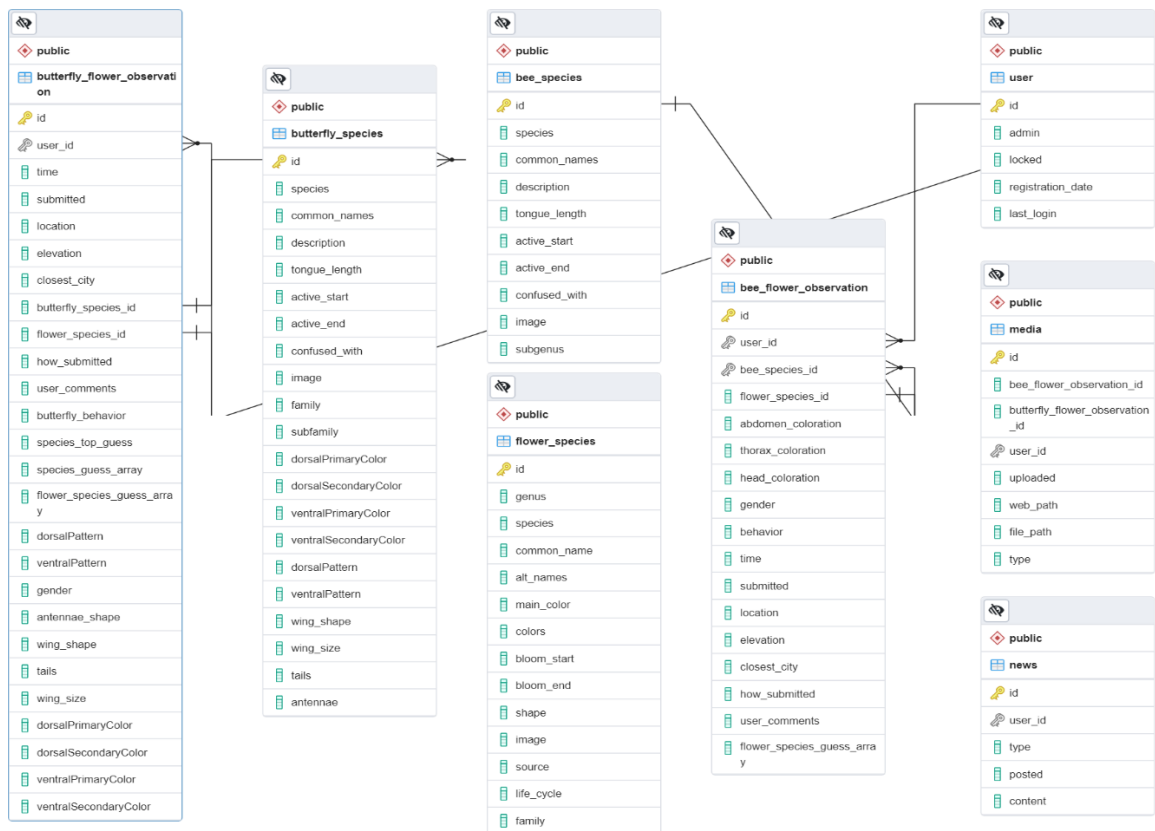


Figure 11: Updated database schema.

3.3 Species Identification using iNaturalist

3.3.1 Flower Identification

The preexisting framework for flower identification was a drop-down box used to select the flower from a list with no help in identifying it. The new system prompts users with a choice between using the existing flower identification or the new vision API. When the users capture an image of their plant and click “Get Suggestion” as shown in Figure 12, the image is condensed and sent by XML Request to the RapidAPI. We filtered these requests with the `taxon_id` for kingdom Plantae, which returned a list of the 10 closest matches for that plant. The top three options are presented to the user to select from. Any plant species returned by iNaturalist that is not already in our database is automatically added, this dynamically expands our list of flower species to include anything that iNaturalist returns.

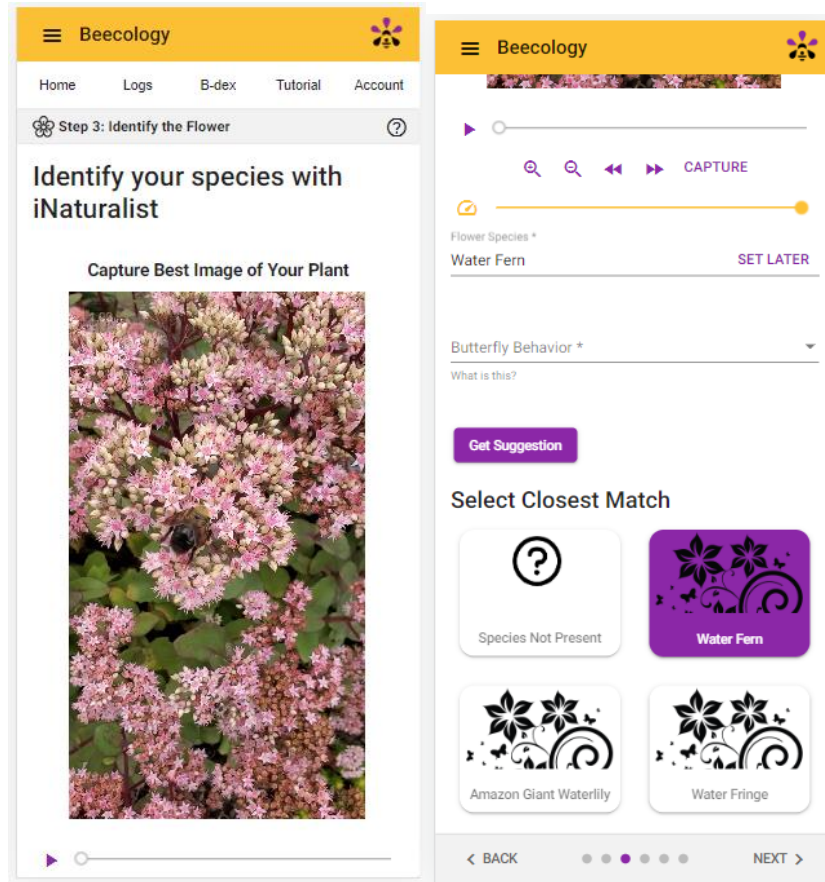


Figure 12: Screenshot of iNaturalist Flower Identification Page.

3.3.2 Butterfly Identification

As we had already begun implementing this feature for flower species identification, we decided to also implement it as an option for butterfly identification. The process is nearly identical to the one for flowers but uses a different `taxon_id` to filter out responses outside the Order Lepidoptera. We also filtered out species that are not present in our database to avoid showing the user options of butterflies that are not present in Massachusetts or species of moths. New species identified by iNaturalist are not added to our database as we need additional features for butterfly species that could not be filled in automatically. We do store the complete list of results with the observation so our domain experts can review them in the future. A

domain expert reviews all bee-flower and butterfly-flower observation submissions to ensure accuracy of identifications and fill in missing data.

3.4 Creating Manual Butterfly Identification

Although the iNaturalist Vision API provides a suitable and quick way to identify a butterfly's species based on an image, we wanted users to be able to contribute to the identification as well. This alternative method of identification will benefit the users by educating them more on butterfly biology and will provide another source of ids that could be used to compare with iNaturalist results to assess the accuracy of the identification. For this to be possible a significant amount of research had to be done into the unique features of each butterfly family as well as more detailed analysis on the specific species present in Massachusetts. Unlike the case for bumblebees, we determined using templates to represent all 120 butterfly species found in Massachusetts would provide too many options for the users.

The first identification method we attempted was utilizing a flowchart to draw the user through different questions to land on their butterfly species or family. A preliminary version of this flow chart shown leads the users through questions including wing size, wing shape, color and antennae shape; these were the same features we were recording for butterfly species and butterfly observations in the database. Initial testing of these flowcharts was promising, but some questions regarding shape were ambiguous and users could end up following the wrong path. Further modifications to the flowchart model were made which included picture examples at certain questions, some reordering of questions asked, and escape questions to redirect users who may have answered certain questions incorrectly (Figure 13).

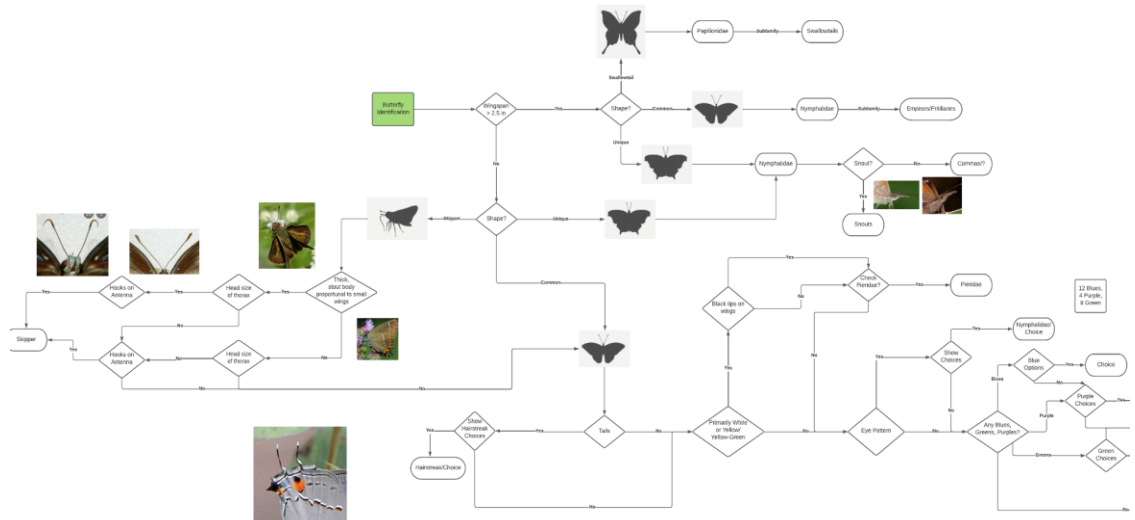


Figure 13: Section of Flowchart for Butterfly Identification. Full flowchart included in Appendix A.

Despite having a little more success with this approach, we came to two important realizations. First, some butterfly species would still have a few remaining options that users would need to scroll through to pick the closest match. Second, if we assigned tags to each of the butterflies with the correct answer to the questions, we could eliminate the need for a flowchart and having to ask questions in a specific order. This second realization is what led us to our final method for manual identification of butterflies, which involved tagging butterfly features and allowed users to filter by features. Each butterfly species was tagged with their dorsal colors, ventral colors, wing size, wing shape, antennae, patterns and presence of tails. In the app, users chose filters to restrict available species options. Because the list of options was available at all times, the users could fill out what fields they knew and were most confident about before scrolling through the options to find the closest matches. From informal testing of users, the ability to add/remove/change filters easily led to increased successful identification results compared to the flowchart method. Additionally, as users became more comfortable with the

available filters and began associating the types of results with their choices, they were able to find the correct matches more quickly, contributing to their mastery of butterfly identification.

As mentioned before, to use this implementation we had to upload and manually tag all of the butterflies in Massachusetts with the correct labels before putting them in the database. Once this was done, implementation on the webapp was simple; we gave the users a dropdown form for filtering options, then filtered the butterfly results by their responses (Figure 14). At the bottom of the page, we included a list of potential matches, limiting the results to the first few matches as to not overwhelm the user with options but provide the ability to scroll through more if desired (Figure 14).

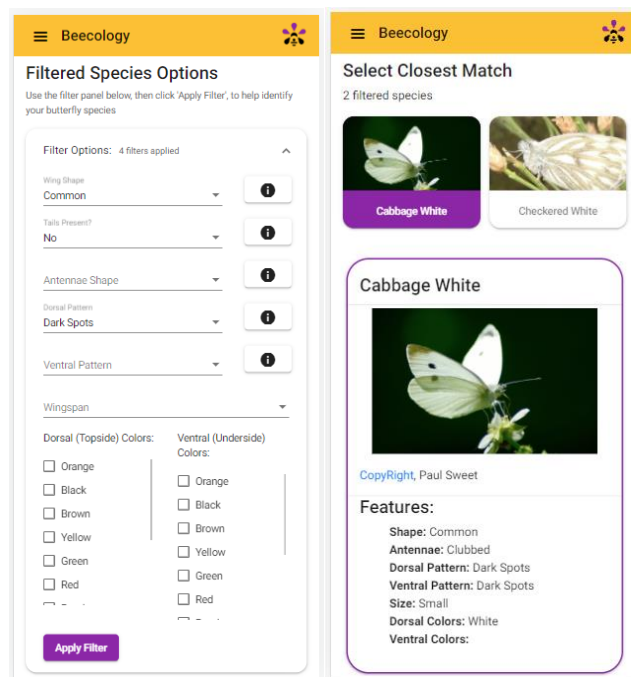


Figure 14: Manual Identification Filter Page and Filter Results.

3.5 Introducing Butterfly Terminology and Tutorial

The last major hurdle with expanding the webapp was educating the users about the new features and butterfly terminology. Because we added a large number of new species to the database, we needed to educate users on the new species. The index of bee species, aptly named the Beedex, was modified and expanded to include all the new butterflies. With options to sort the butterflies by families or individually, users can access the records to learn more about each of the species or families.

Since we are asking users to identify their species with the butterfly manual identification filters, it was important that we added details and examples to the filters to help guide people as they identify a butterfly. Next to each of the filters is an info button that opens a small window which provides picture examples and descriptions for each of the choices. An example shown in Figure 15, shows guides for identifying butterflies' shapes, antennas and patterns. Additionally, video sources for each of the available butterfly behaviors were added to the website to help users identify butterfly behaviors.

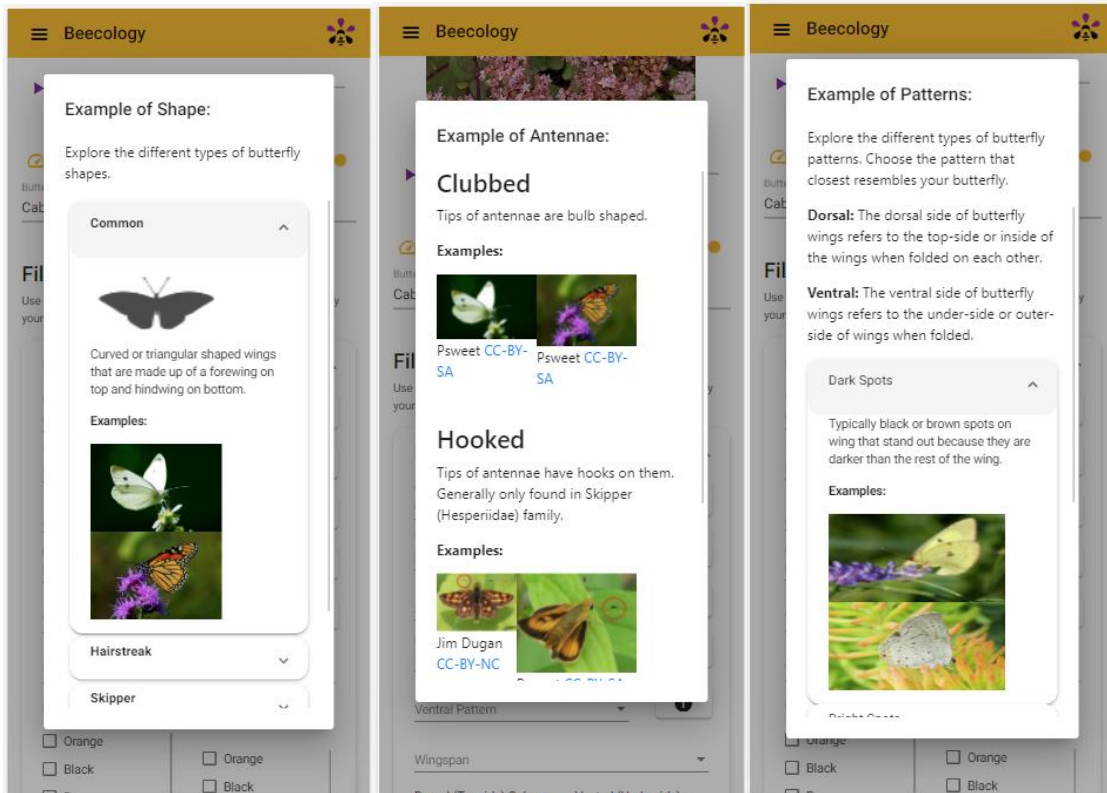


Figure 15: Examples of Popover Text Feature Identification Guides.

In addition to the resources added for butterfly identification, changes needed to be made to the existing tutorials for identification. Since the original tutorial only accounted for the bee identification and submission, we needed to add steps for butterfly identification and submission, and to educate users on how to use the iNaturalist identification for butterfly or flowers alike. Despite trying to maintain a simple user interface it is good to educate the user about its new features during large updates. In order to effectively prompt the user to look at the tutorial when a new version of the webapp is released, the app displays a pop-up message providing information about the new app version and a link to the new tutorial.

Chapter 4: Discussion

4.1 Review Identification Methods

A significant part of the project was creating the new functionality for manual butterfly identification and automatic iNaturalist butterfly and flower identification. The process of creating the manual identification began with a flowchart style similar to how bee identification uses the abdomen, thorax, and head patterns to determine the species. This approach did not quite work on the scale of 120 species, so we instead went with a system of filtering down the database list. This system allowed the user to get closer to a solution without knowing all of the traits of the butterfly, and was more forgiving of mistakes, since an incorrect filter can be removed or changed without resetting the rest. The new procedures for butterfly identification as detailed above will likely become a standard for any new species added. The robust design of the manual identification system can be adapted for different types of pollinators based on their distinctive features, as well as for a wider set of butterflies with the existing features. In the case of iNaturalist identification, instead of creating our own process, we implemented an existing one. This project involved communicating with third party software via RapidAPI, and ensuring we were processing both the image to send and the response we received properly. We also needed to contact the people from iNaturalist, properly credit them by displaying their logo, and deliberate over how much we wanted to pay for API calls, with a monthly limit on free calls and several priced options for more identifications. It is also likely that iNaturalist identification will become a standard option for future pollinator and plant expansions as it is both easy to implement and will work for many types of species.

The manual identification has better educational value for users and is free from any potential biases of the iNaturalist system, but the automatic system is quicker and easier to get a suggestion from. It is worth noting that iNaturalist is likely to be more viable for plant identification because of the vast amount of plant species that need to be accounted for, but also that it is less likely to be able to distinguish small species such as bees. Each system has its strengths and weaknesses, and both are valued additions to the application.

4.2 Final Thoughts on the Project

Ultimately, the proposed expansion of the Beecology Project's web application to include butterflies was a successful endeavor. Information pertaining to submitting butterfly-flower observations is available in the tutorial, accessible from the webapp's home page, and information about the individual butterfly families and species is available in the B-dex. As discussed above many of the design choices were kept in line with the preexisting web application. Additionally, while other variations of database restructuring were considered, the final version utilized the same framework that existed before the project. If the webapp were to be expanded again in the future to include more types of pollinators we would recommend that the team uses a similar approach. First, identify the key features of the pollinator and expand the database to hold its members. Next, expand and test the server with new data types. Finally, work on the web application log creation and submission. More static pages like the B-dex, or tutorials can be modified at any point, though it may be easier to modify toward the end.

The resulting expansion of the Beecology Project will be significant for the future of data collection in the application. With several "butterfly hunters" joining the project we expect an influx of observations. The resulting larger community of citizen scientists means more valuable

data on pollinator decline and their interactions with the ecosystem will be tracked and available for data analysis.

4.3 Project Challenges

One persistent challenge throughout the project was the learning curve required to work on an existing web application. Because we were making such large changes to the database and server, we did not want to risk working on the production environment as we were making frequent changes to the code that could cause large amounts of downtime. This required us to run the servers and database on our local machines which we had some difficulty setting up because of the lack of documentation for doing so. The other main challenge of working on an existing webapp was not being familiar with the deep intricacies of the existing system. Luckily former students who had worked on the application were able to help guide us from time to time, and our familiarity with the webapp's inner working grew as we continued working on the project. To mitigate some of these issues for future projects, we worked with another student, Oliver Yasuna, to add more documentation to the existing framework. He created a new module on the project's GitLab, the location where all the code is stored, and we worked to fill in the gaps regarding our new code and other useful information that was previously stored in random files or direct messages.

4.4 Future Work

While the implementation of butterflies and iNaturalist into the web application were successful, both aspects can be improved upon in the future. The most obvious expansion would be to include more bee/butterfly and plant species and expand the project's reach to more parts of

the world. Another logical expansion of the Beecology Project would be to extend the existing bee data visualizations (and create more) for the new butterfly dataset. One last side project that might be interesting to conduct is a comparison between the butterfly manual identification and the iNaturalist identification correctness.

4.4.1 Adding More Species

Currently, adding more species to the webapp for butterflies and bees is a very different procedure. Because of the butterfly's manual identification filtering method and iNaturalist's capability to identify butterfly species automatically, adding new butterfly species would only require an admin to add the new species to the database with the appropriate fields filled in. The current identification method for bumblebees utilizes a flowchart for selecting between bee patterns. The limited number of bumblebees makes this a viable option at the current moment, but if the Beecology Project wanted to support more bumblebee species or other bee families this approach would quickly grow cumbersome and lead to problems. One approach would be to rely on automatic identification using iNaturalist for further observations, which would eliminate this bottleneck. Another option would be to re-design the manual bee identification in a similar manner to the butterfly one, allowing it to be more easily expanded for new species. A few static pages of the application may need to be modified to accept new bee families as the current system is only set up for bumblebees, but those should be relatively easy to modify in comparison to adding a new type of pollinator assuming the database does not need to be modified to store more attributes.

As for adding other types of pollinators, the system would not look too different to the procedure we took for expansion. Hopefully the work done in this project setting up iNaturalist identification and code documentation will make the processes smoother for the next expansion.

4.4.2 Creating More Data Visualization Tools

The Beecology Project has a wide set of existing data visualizations tools for bee-flower observations, that need to be expanded for the new butterfly-flower observations. The preexisting visualizations focus mostly on tracking number of bees, locations, seasonality and bee-flower relationships. By adapting these tools for butterflies the Beecology Project can aim to track and visualize their life cycles and identify which types of flowers are the most vital in sustaining their populations. In addition to formatting the data to work for the existing visualizations, additional filters need to be added to accommodate the butterfly data so individuals can sort by families or other significant features. Other possible ideas for visualization tools specific to butterfly observations might be visualizations specific to butterflies that are mating or oviposition which could be useful for getting a better understanding of butterflies' life cycles and what plants may be critical in their reproduction.

4.4.3 Comparing iNaturalist and Manual Identification

It is difficult to predict which method, “Manual id” or “iNaturalist id”, people will use most for butterfly identification, but regardless the data that is collected from each of those observations can be useful in improving the other method. With a significant enough number of observations, a future project could assess the effectiveness and accuracy of our manual butterfly identification method versus the iNaturalist vision API. While it is possible most people will just use iNaturalist identifications for their butterfly observations, if it is seen that iNaturalist is continuously misidentifying some species the procedure can be modified to help guide the users into picking the correct choice. Likewise, if a group of species is commonly misidentified in the

manual id the app can be changed to warn users or suggest that they use iNaturalist to help distinguish between the group. Currently, regardless of which species was picked by the user, all of iNaturalists results are appended to the log which allows for future comparisons to be made assessing users decision making.

References

- Beecology Project. (2021). Beecology Project Home. <https://beecology.wpi.edu/>
- Bio-CS Bridge. (2021). Bio-CS Bridge Home. <https://biocsbridge.wpi.edu/>
- Britannica. (2013). Natural History. <https://www.britannica.com/animal/lepidopteran/Natural-history>
- Butterflies of Singapore. (2022). The Butterfly Antennae
<https://butterflycircle.blogspot.com/2017/10/the-butterfly-antennae.html>
- Candeias, M. (2021). In Defense of Plants: An Exploration into the Wonder of Plants. Mango Publishing Group. 978-1-64250-453-8
- Cannon, R. (2020). Forelegs And Four Legs in Butterflies.
<https://rcannon992.com/2020/03/25/forelegs-and-four-legs-in-butterflies/comment-page-1/>
- Cech, R. & Tudor, G. (2007). Butterflies of the East Coast: An Observer's Guide 978-0-69109-055-9
- Cinici, A. (2013). From caterpillar to butterfly: A window for looking into students' ideas about life cycle and life forms of insects. *Journal of Biological Education*, 47(2), 84-95.
- Di Cecco, G. J., Barve, V., Belitz, M. W., Stucky, B. J., Guralnick, R. P., & Hurlbert, A. H. (2021). Observing the observers: How participants contribute data to iNaturalist and implications for biodiversity science. *BioScience*, 71(11), 1179-1188.
- Feng, M. (2021). How Long Do Butterflies Live? By Species. Marco Feng.
<https://www.marcofeng.com/usa/how-long-do-butterflies-live>
- 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.
- Goldsborough, P. (2016). A tour of tensorflow. arXiv preprint arXiv:1610.01178.
- Hooks, C., & Espíndola, A. (2017). Moths, butterflies, and pollination. *Maryland Agronomy News*. Retrieved April 26, 2022, from
<http://blog.umd.edu/agronomynews/2020/06/05/moths-butterflies-and-pollination/>
- Kim, M.I., Wan, X., Kim, M.J., Jeong, H.C., Ahn, N., Kim, K., Han, Y.S., & Kim, I. (2010). Phylogenetic relationships of true butterflies (Lepidoptera: Papilionoidea) inferred from COI, 16S rRNA and EF-1 α sequences. *Molecules and Cells*, 30, 409-425.

- Kluser, S., & Peduzzi, P. (2007). Global pollinator decline: a literature review.
- Loarie, S. (2021). About iNaturalist. <https://www.inaturalist.org/pages/about>
- Potts, S. G., Biesmeijer, J. C., Kremen, C., Neumann, P., Schweiger, O., & Kunin, W. E. (2010). Global pollinator declines: trends, impacts and drivers. *Trends in ecology & evolution*, 25(6), 345-353.
- Prasifka, J. R., Mallinger, R. E., Portlas, Z. M., Hulke, B. S., Fugate, K. K., Paradis, T., ... & Carter, C. J. (2018). Using nectar-related traits to enhance crop-pollinator interactions. *Frontiers in plant science*, 9, 812.
- Shepard, A. (2022). iNaturalist Computer Vision Explorations. iNaturalist. https://www.inaturalist.org/pages/computer_vision_demo#:~:text=Vision%20Explorations%20%20%20%20iNaturalist-.iNaturalist%20Computer%20Vision%20Explorations,tell%20you%20what%20you%20saw
- Skymeadows. (2015). Butterfly Anatomy. <https://www.skymeadows.info/butterfly-anatomy>
- Thomann, M., Imbert, E., Devaux, C., & Cheptou, P. O. (2013). Flowering plants under global pollinator decline. *Trends in plant science*, 18(7), 353-359.
- Vanbergen, A. J., & Initiative, T. I. P. (2013). Threats to an ecosystem service: pressures on pollinators. *Frontiers in Ecology and the Environment*, 11(5), 251-259.
- Wenzel, A., Grass, I., Belavadi, V. V., & Tschardtke, T. (2020). How urbanization is driving pollinator diversity and pollination—A systematic review. *Biological Conservation*, 241, 108321.
- Westerkamp, C. H. (1996). Pollen in bee-flower relations some considerations on melittophily. *Botanica Acta*, 109(4), 325-332.
- Wildlife Insights. (2022). Moth Anatomy. Wildlife Insights. <http://www.wildlifeinsight.com/guide-to-british-moths/moth-anatomy/>
- Woodland Honey. (2012). Honey Diaries. <https://woodlandhoney.tumblr.com/post/30188401951>

