PROJECT REPORT

Artificial Intelligence

Digital Naturalist - AI Enabled tool for Biodiversity Researchers

Team Members:

Chiranjeevi M P - 2019103013

Shreya Ananth - 2019103580

Aditya Ramachandran - 2019103502

Vasudha E - 2019103073

Team ID: PNT2022TMID35185

1. INTRODUCTION

1.1 Project Overview:

In this project, a web application is created using flask which uses a deep learning model, trained on different species of birds, flowers and mammals. The application helps naturalists in identifying different species of plants, animals and birds on the go thereby making the learning process simpler.

1.2 Purpose:

Naturalists find it difficult to get information on different species of flora and fauna while venturing into woods. This application will serve as an alternative to traditional methods like guidebooks thereby leveraging the reach of information.

2. LITERATURE SURVEY

2.1 Existing problem:

People belonging to the fields of research, trekking, hiking are in constant search of information about flora and fauna for uses such as medicine, photography, and environmental protection. Thus they need a reliable, fast and simpler method to recognize the species which they encounter.

2.2 References:

1. Nguyen, H., Maclagan, S. J., Nguyen, T. D., Nguyen, T., Flemons, P., Andrews, K., ... & Phung, D. (2017, October). Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In 2017 IEEE international conference on data science and advanced Analytics (DSAA) (pp. 40-49). IEEE.

2. Huang, Y. P., & Basanta, H. (2021). Recognition of Endemic Bird Species Using Deep Learning Models. IEEE Access, 9, 102975-102984.

3. Gogul, I., & Kumar, V. S. (2017, March). Flower species recognition system using convolution neural networks and transfer learning. In 2017 fourth international conference on signal processing, communication and networking (ICSCN) (pp. 1-6). IEEE.

4. de Arruda, M. D. S., Spadon, G., Rodrigues, J. F., Gonçalves, W. N., & Machado, B. B. (2018, July). Recognition of endangered pantanal animal species using deep learning methods. In 2018 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

5. Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., Sudirman, S., & Liu, W. (2019). PestNet: An end-to-end deep learning approach for large-scale multi-class pest detection and classification. IEEE Access, 7, 45301-45312.

6. Zhang, H., He, G., Peng, J., Kuang, Z., & Fan, J. (2018, April). Deep learning of path-based tree classifiers for large-scale plant species identification. In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) (pp. 25-30). IEEE.

7. Jeon, W. S., & Rhee, S. Y. (2017). Plant leaf recognition using a convolution neural network. International Journal of Fuzzy Logic and Intelligent Systems, 17(1), 26-34.

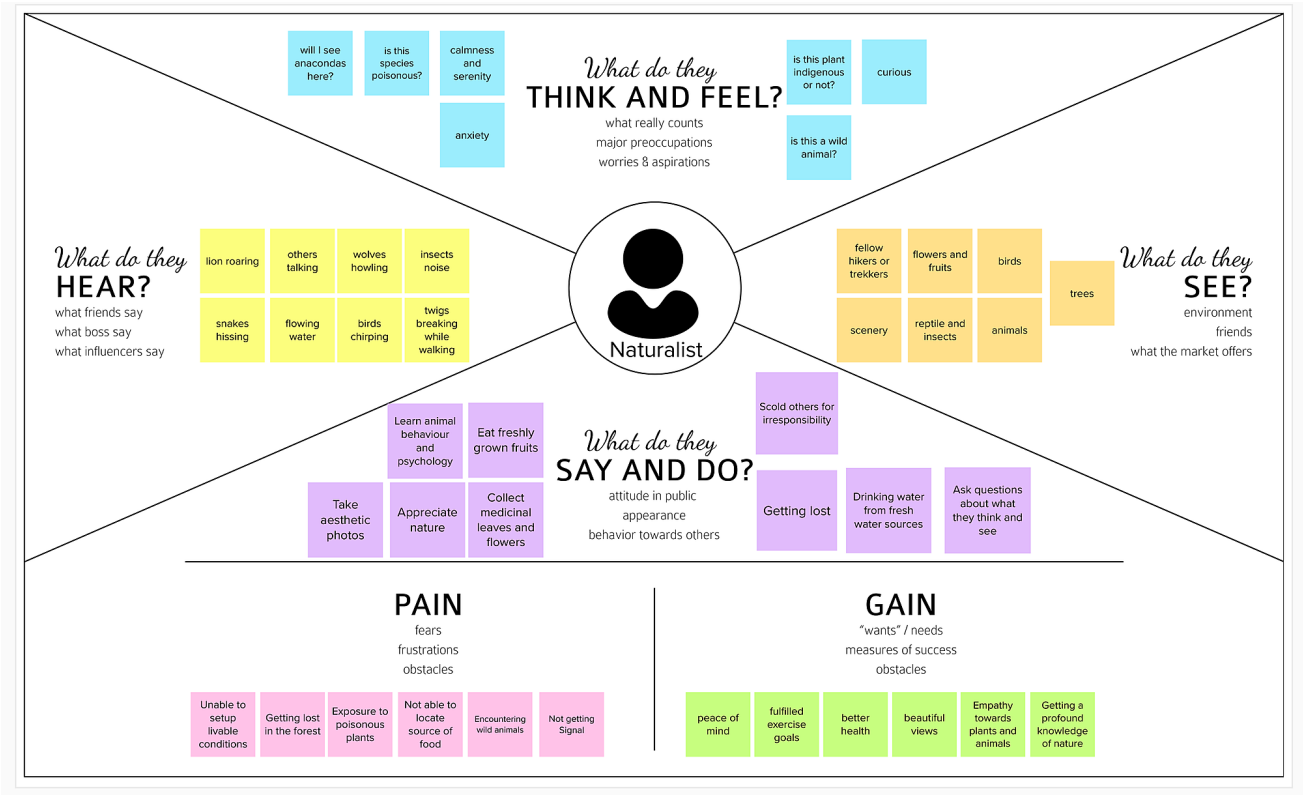
8. Huixian, J. (2020). The analysis of plants image recognition based on deep learning and artificial neural network. IEEE Access, 8, 68828-68841.

2.3 Problem Statement Definition:

Recognition of biological species of plants and animals in different terrains using image processing and computer vision techniques.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:

Template

Brainstorm & idea prioritization

Digital Naturalist - AI Enabled tool for Biodiversity Researchers

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Share template feedback

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM STATEMENT

How might we help people identify the biological species of the plants and animals around them?

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Chiranjeevi

A Local Guide

Trees are tagged with their information

Flora-Fauna Information Crash Course

Send in a team, where each member is a specific domain specialist

Shreya Ananth

Travel archives and encyclopedia

Information boards

Attribute and description based search on google

Prepare before hand by watching travel vlogs

Vasudha

Ask native people

Guess species based on geographic location

Accessing state government tourism portal to know native species

Test for contact poisoning by placing a piece of plant near the skin.

Aditya

Image based species prediction

Guess the species type based on climate

Be aware of strong, unpleasant, musty odour, as they are indication of poisonous plants.

Find out the type of plant by noticing its venation and flower type.

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Human guidance

- A Local Guide
- Send in a team, where each member is a specific domain specialist
- Prepare before hand by watching travel vlogs
- Ask native people
- Test for contact poisoning by placing a piece of plant near the skin.

Intelligent search

- Attribute and description based search on google
- Image based species prediction

Recorded information

- Travel archives and encyclopedia
- Information boards
- Trees are tagged with their information
- Flora-Fauna Information Crash Course
- Accessing state government tourism portal to know native species

Educational guess

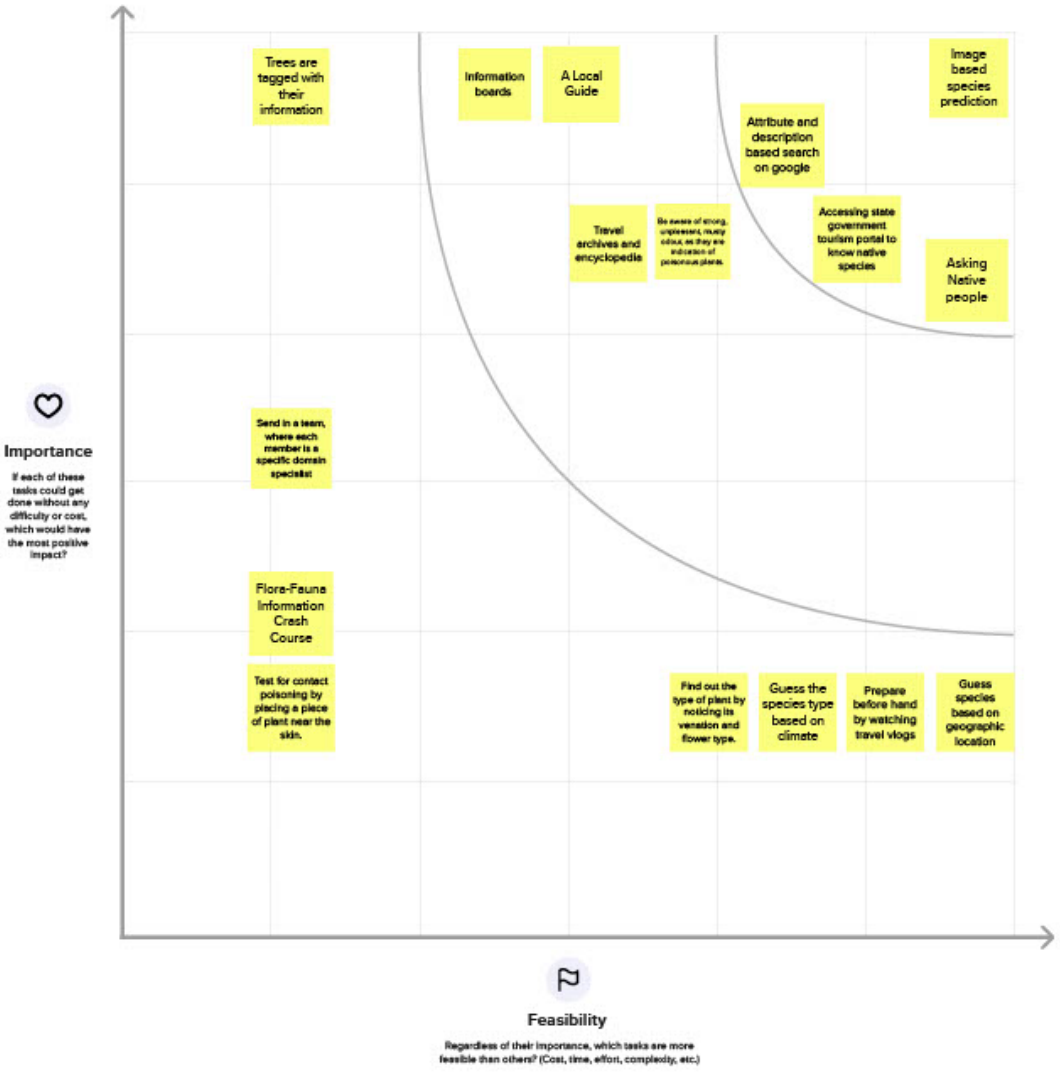
- Guess species based on geographic location
- Guess the species type based on climate
- Find out the type of plant by noticing its venation and flower type.
- Be aware of strong, unpleasant, musty odour, as they are indication of poisonous plants.

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	● There is a need for a way to analyze and identify the type of living beings in a particular environment a person finds themselves in, so that they can gain knowledge about different species. ● There should be a recognition software that is able to recognize the species in any given angle.
2.	Idea / Solution description	● The aim is to develop a recognition software using the concept of supervised learning that takes in the image of various species as the input and provides the species name as output.
3.	Novelty / Uniqueness	● Unlike the other open source solutions available, this application not only classifies an image as either plant or animal but also tells about the individual species name. ● There are also some solutions available which either work only for one class of species, i.e, either plants or animals.
4.	Social Impact / Customer Satisfaction	● Creates a set of model citizens who are aware of the various species in their surroundings prompting them to be more environmentally conscious. ● Creates a way to identify the indigenous and endangered species so that people can spread awareness about them and protect those species.
5.	Business Model (Revenue Model)	**Business to Consumer model** ● The solution is a reliable recognition software planned to be created as an application with which the consumers can identify the type of living beings in a particular environment. ● It follows a non-monetary revenue model where the consumers aren't asked to pay any fee but when they use the software for recognition purposes the image they provide is stored in the database and used for future training

6.	Scalability of the Solution	• This project is focused on recognizing a limited number of species of each category. • In future, this project can be extended to recognise many other species with the help of a carefully crafted dataset. • This project can be extended to provide more detailed information about each instance of a living being like places where they are commonly found, eating habits, etc.

3.4 Problem Solution fit:

Project Design Phase-I - Solution Fit

Project Title: Digital Naturalist - AI Enabled tool for Biodiversity Researchers **Team ID:** PNT2022TMID35185

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- Botanists
- Wildlife-photographers
- Trekkers
- Ornithologists
- Hikers
- Naturalists
- Mountaineer
- Backpackers

6. CUSTOMER CONSTRAINTS CC

- Inability to store massive amounts of jargons and information about flora & fauna in mind
- Unavailability of an one-stop solution for different species of both flora and fauna

5. AVAILABLE SOLUTIONS AS

- Accessing state government tourism portal to know about native species
- Asking native people
- Travel archives and Encyclopedia
- Flora-fauna information crash course

Explore AS, differentiat

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- There is a need for a way to analyze and identify the type of living beings in a particular environment a person finds themselves in, so that they can gain knowledge about different species.
- There should be a recognition software that is able to recognize the species in any given angle.

9. PROBLEM ROOT CAUSE RC

- Unavailability of information regarding the local flora and fauna
- There is very less knowledge about the animals and plants living in the locality

7. BEHAVIOUR BE

- In their free time they browse through various sources to gain knowledge about the local flora and fauna
- Whenever they need help onsite, access the online resources to clarify their doubts regarding the encountered species

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

- Unable to determine whether a particular species of plant is poisonous or not while camping
- Having trouble specifying the class of animals such as herbivore, carnivore, omnivore

10. YOUR SOLUTION SL

The aim is to develop a recognition software using the concept of supervised learning that takes in the image of various species as the input and provides the species name as output.

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE

- Whenever they need help onsite, access the online resources to clarify their doubts regarding the encountered species

8.2 OFFLINE

- In their free time they browse through various sources to gain knowledge about the local flora and fauna

Identify strong TR & EM

4. EMOTIONS: BEFORE / AFTER EM

- Before:
  - Frustrated,
  - lost,
  - & confused
- After:
  - Relieved,
  - enlightened,
  - & confident

4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Upload an image/Live capturing	Choose the image from file explorer Camera enabled devices support live capturing
FR-4	Search	Recognition of species from the uploaded photo Result display
FR-5	User history	Search results are stored

4.2 Non-Functional requirements:

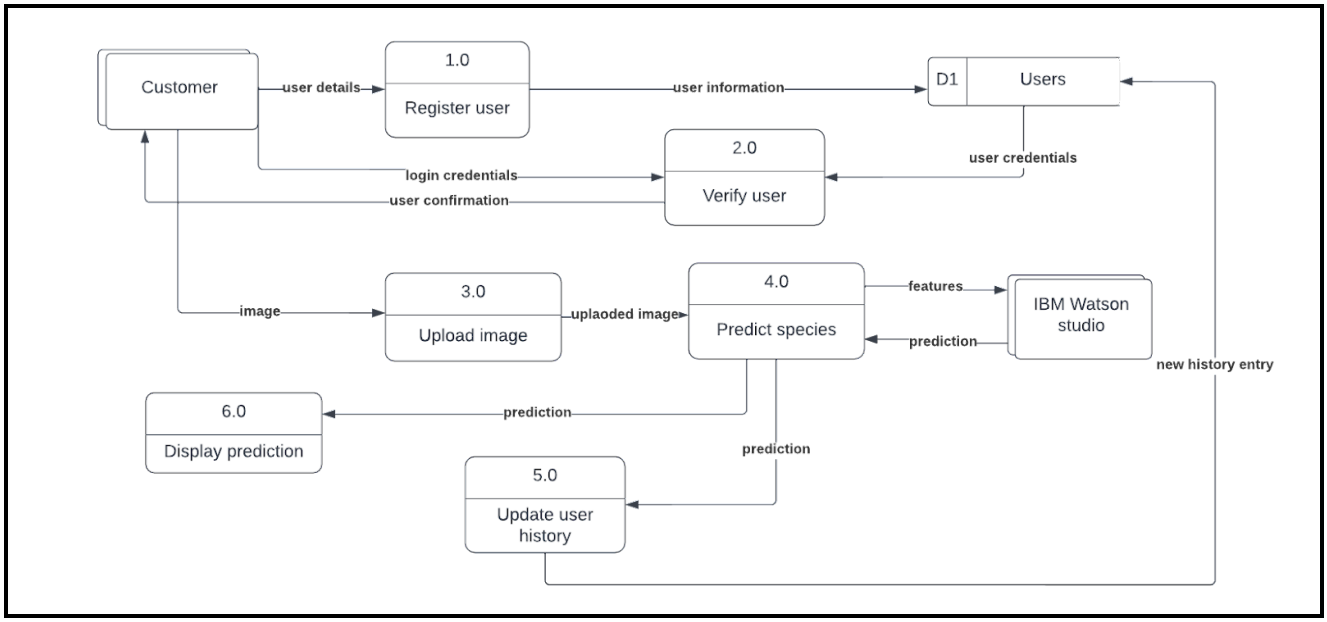
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Effectiveness, efficiency and overall satisfaction of the user while interacting with our application. It should be a GUI interface which should be easy to use.
NFR-2	**Security**	Authentication, authorization, encryption of the application provided by IBM cloud.
NFR-3	**Reliability**	The system must perform without failure for most of the time.
NFR-4	**Performance**	How the application is functioning and how responsive the application is to the end-users depending on the performance of IBM cloud platform.

NFR-5	**Availability**	Without near 100% availability, application reliability and the user satisfaction will affect the solution.
NFR-6	**Scalability**	Capacity of the application to handle growth, especially in handling more users.

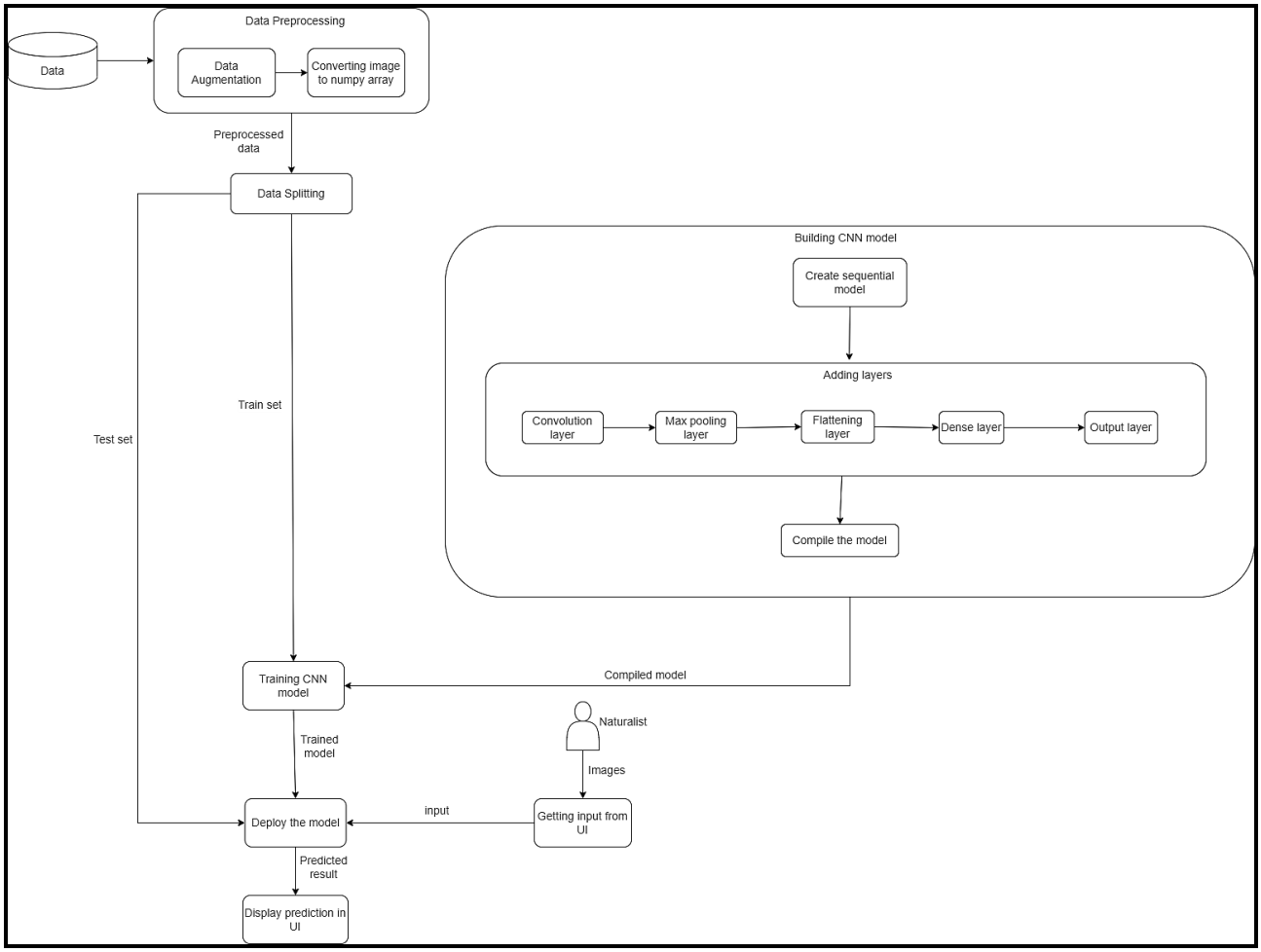
5. PROJECT DESIGN

5.1 Data Flow Diagrams:

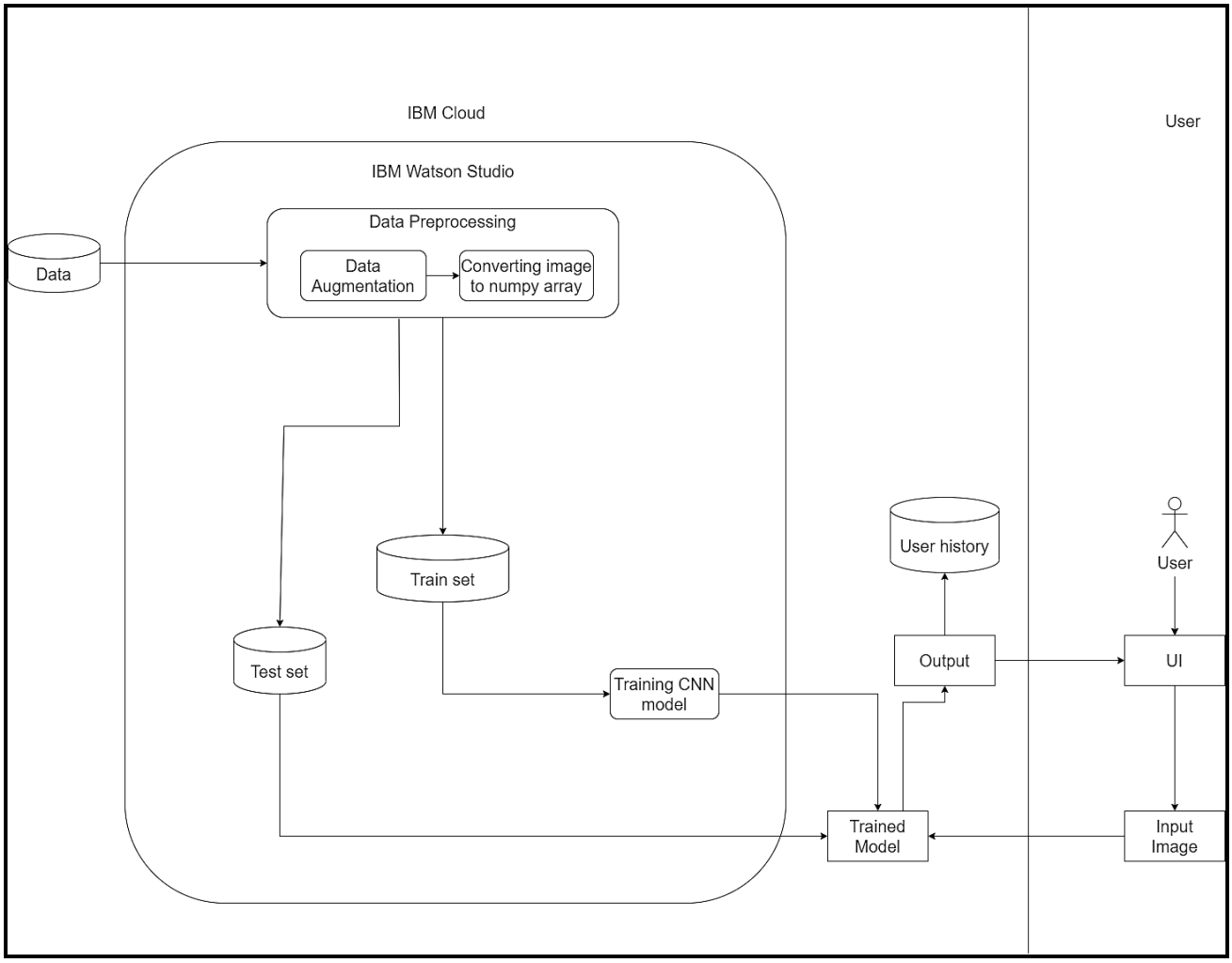


5.2 Solution & Technical Architecture:

Solution Architecture:



Technology Architecture:



5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	User Registration	USN-1	As a user, I can register for the application by entering my email, password and other details	I can access my account / dashboard	Medium	Sprint-2
Customer	User Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-3
Customer	Login	USN-3	As a user, I can log into the application by entering email & password	I can log in successfully and see my dashboard	Medium	Sprint-2
Customer (Web user)	Upload image	USN-4	As a user, I can upload the image of the species to identify it	A dialogue box stating successful upload	High	Sprint-1
Customer (Web user)	Predict species	USN-5	As a user, I can use this feature to predict the species	The name of species is displayed	High	Sprint-1
Customer (Web user)	User History	USN-6	As a user, I can check the search history	A list of the searched species names is displayed	High	Sprint-3

Custom er (Web user)	User Logout	USN-7	As a use, I want to be able to logout	I am able to successfully logout and the login page is displayed	Low	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-2	User Registration	USN-1	As a user, I can register for the application by entering my email, password and other details.	3	Medium	1. Vasudha E 2. Chiran Jeevi MP
Sprint-3	User Confirmation	USN-2	As a user, I will receive a confirmation email once I have registered for the application.	3	Medium	1. Vasudha E 2. Shreya Ananth

Sprint-2	Login	USN-3	As a user, I can log into the application by entering email & password.	3	Medium	1. Vasudha E 2. Aditya Ramachandran
Sprint-1	Upload image	USN-4	As a user, I can upload the image of the species to identify it.	2	High	1. Shreya Ananth 2. Chiran Jeevi MP
Sprint-1	Predict species	USN-5	As a user, I can use this feature to predict the species.	5	High	1. Shreya Ananth 2. Aditya Ramachandran
Sprint-3	User History	USN-6	As a user, I can check the search history.	4	High	1. Aditya Ramachandran 2. Chiran Jeevi MP
Sprint-3	User Logout	USN-7	As a user, I want to be able to logout.	1	Low	1. Chiran Jeevi MP

Sprint-4	Application testing	USN-8	The project is tested to guarantee that the system is successfully built and meets all the requirements.	3	Medium	1. Vasudha E 2. Chiran Jeevi MP
Sprint-4	Deployment	USN-9	The system is deployed on the IBM cloud. It is made available for use.	3	High	1. Aditya Ramachandran 2. Shreya Ananth

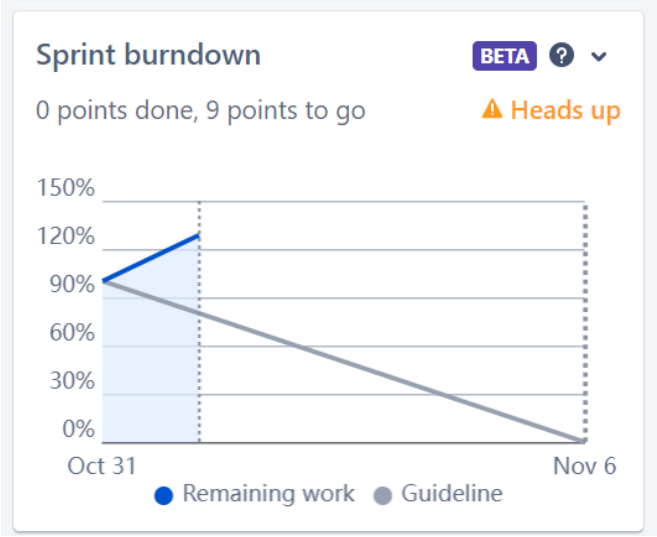
6.2 Sprint Delivery Schedule:

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	7	6 Days	24 Oct 2022	29 Oct 2022	7	29 Oct 2022
Sprint-2	6	6 Days	31 Oct 2022	05 Nov 2022	6	05 Nov 2022
Sprint-3	8	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022

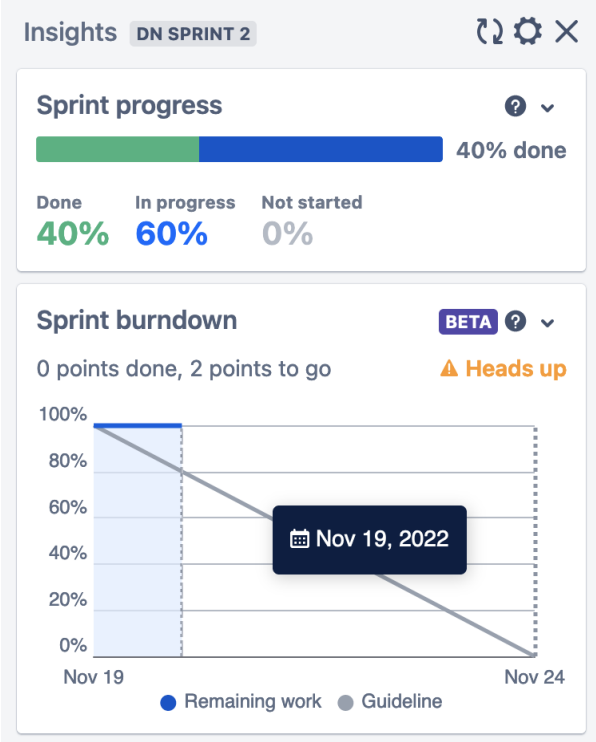
Sprint-4	6	6 Days	14 Nov 2022	19 Nov 2022	6	19 Nov 2022

6.3 Reports from JIRA:

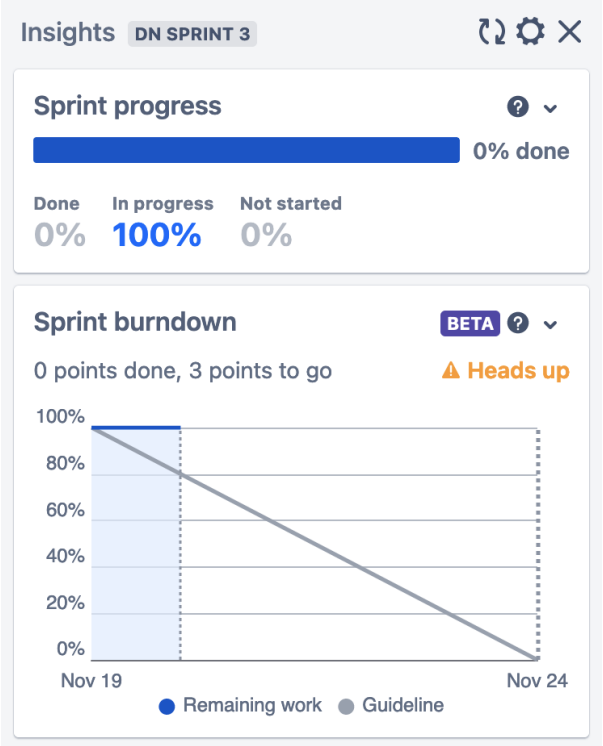
Sprint 1:



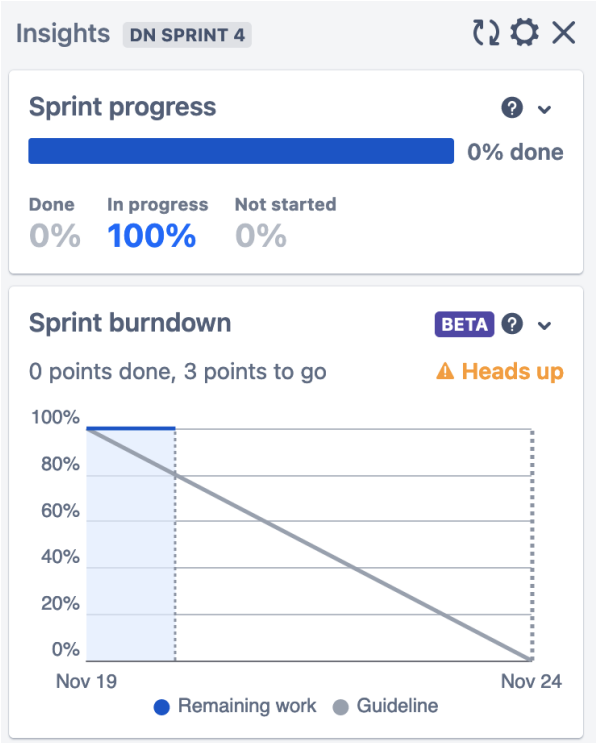
Sprint 2:



Sprint 3:



Sprint 4:



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1 (Prediction of species):

The user uploads an image of the species which will then be predicted for its name using a trained deep learning model.

Code:

Creating the model

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense,Dropout
from tensorflow.keras import regularizers
import tensorflow, keras
```

```
model = Sequential()
model.add(Convolution2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(300, activation='relu', kernel_regularizer=regularizers.L2(0.001)))
model.add(Dense(150, activation='relu', kernel_regularizer=regularizers.L2(0.001)))
model.add(Dense(75, activation='relu', kernel_regularizer=regularizers.L2(0.001)))
model.add(Dense(6, activation='softmax'))
```

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
model.summary()
```

Fitting the Model

```
from keras.callbacks import EarlyStopping, ReduceLROnPlateau
early_stopping = EarlyStopping(monitor='val_accuracy',patience=5)
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy',patience=5,factor=0.5,min_lr=0.00001)

callback = [reduce_lr,early_stopping]
```

```
model.fit(x_train, steps_per_epoch=len(x_train), epochs=30, validation_data=x_val, validation_steps=len(x_val))
```

Testing the model

```
a = model.evaluate_generator(x_test, len(x_test))
```

```
print(f'Error: {a[0]}')
print(f'Accuracy: {a[1]}')
```

```
Error: 2.347017765045166
Accuracy: 0.7232704162597656
```

```
from tensorflow.keras.preprocessing import image
```

```
%pylab inline
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

classes = {'Corpse Flower': 0,
 'Great Indian Bustard Bird': 1,
 'Lady Slipper Orchid Flower': 2,
 'Pangolin Mammal': 3,
 'Senenca White Deer Mammal': 4,
 'Spoon Billed Sandpiper Bird': 5}
k = 0
n = 0
for cat in listdir('DN/'):
    if cat == '.DS_Store':
      continue;

    for direc in listdir('DN/' + cat):
        if direc == '.DS_Store':
          continue;
        img = image.load_img('DN/' + cat + '/' + direc, target_size = (64,64))
        img = image.img_to_array(img)
        img = np.expand_dims(img, axis = 0)
        pred = np.argmax(model.predict(img), axis = 1)

        print(cat, pred, classes[cat])
        if pred == classes[cat]:
          k += 1

        #img = mpimg.imread('DN/' + cat + '/' + direc)
        #imgplot = plt.imshow(img)
        #plt.show()
        n += 1
print("Accuracy: ", k/n)
```

7.2 Feature 2 (Search history - storing the predictions):

The predicted image is then stored in the MySQL database along with the prediction. The stored contents are retrieved from the database to be rendered on HTML/CSS template as a carousel of images and predictions [as captions].

Flask Code:

```
@app.route('/showimage')
def showimage():
    if 'loggedin' in session:
        img_filename = request.args['filename']
        model = load_model('final_model.h5')
        classes = ['Corpse Flower','Great Indian Bustard Bird','Lady Slipper Orchid Flower',
                   'Pangolin Mammal','Senenca White Deer Mammal','Spoon Billed Sandpiper Bird']

        img = image.load_img(os.path.join(app.config['UPLOAD_FOLDER'], img_filename), target_size=(64, 64))
        img = image.img_to_array(img)
        img = np.expand_dims(img, axis=0)
        pred = np.argmax(model.predict(img), axis=1)
        print(pred)
        print('Prediction: ', classes[pred[0]])
        cursor = mysql.connection.cursor()
        fil = open(os.path.join(app.config['UPLOAD_FOLDER'], img_filename), 'rb').read()
        # We must encode the file to get base64 string
        fil = base64.b64encode(fil)
        cursor.execute('INSERT INTO imagetable (photo,username,pred) VALUES (%s,%s,%s)',
                       (fil, session['username'], classes[pred[0]]))
        mysql.connection.commit()
        return render_template(
            'showimage.html', uploaded_image=os.path.join(app.config['UPLOAD_FOLDER'], img_filename), prediction=classes[pred[0]])
    return redirect(url_for('login'))
```

```
@app.route('/history')
def history():
    cursor = mysql.connection.cursor()
    cursor.execute(
        'SELECT photo,pred FROM IMAGETABLE WHERE username = %s ORDER BY date_entry DESC', (session['username'],))
    data = cursor.fetchall()
    if len(data) == 0:
        return render_template('errorpage.html', message="No History found!!")
    images = []
    preds = []
    for item in data:
        (image,pred) = item
        binary_data = base64.b64decode(image)
        im = io.BytesIO(binary_data)
        encoded_img_data = base64.b64encode(im.getvalue())
        images.append(encoded_img_data.decode('utf-8'))
        preds.append(pred)
    return render_template('history.html', photos = images, preds = preds)
```

Template:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>History</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
  <style>

  .carousel-inner img {
    margin-top: 8%;
    display: block;
    margin-left: auto;
    margin-right: auto;
    height: 500px;
    width: auto;
  }

  .carousel-control-next,
  .carousel-control-prev {
    filter: invert(100%);
  }

  </style>
</head>
```

```
<body class="loggedin">
  <nav class="navtop">
    <div>
      <h1>Digital Naturalist</h1>
      <a href="{{ url_for('home') }}"><i class="fas fa-home"></i>Home</a>
      <a href="{{ url_for('profile') }}"><i class="fas fa-user-circle"></i>Profile</a>
      <a href="{{ url_for('history') }}"><i class="fa fa-history"></i>History</a>
      <a href="{{ url_for('logout') }}"><i class="fas fa-sign-out-alt"></i>Logout</a>
    </div>
  </nav>

<div style="padding-bottom: 10px; padding-left: 10px; padding-right: 10px;">
<div id="demo" class="carousel slide" data-ride="carousel">
  <!-- Indicators -->
  <ul class="carousel-indicators">
    <li data-target="#demo" data-slide-to="0" class="active"></li>
    {% for i in range(1,photos|length) %}
    <li data-target="#demo" data-slide-to="{{ i }}"></li>
    {% endfor %}
  </ul>

  <!-- The slideshow -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="{{ photos[0] }}" alt="LA">
      <div class="carousel-caption">
        <p>Predicted as {{ preds[0] }}</p>
      </div>
    </div>
    {% for i in range(1,photos|length) %}
     <div class="carousel-item">
       <img src="{{ photos[i] }}" alt="CHICAGO">
       <div class="carousel-caption">
         <p>Predicted as {{ preds[i] }}</p>
       </div>
     </div>
     {% endfor %}
   </div>

   <!-- Left and right controls -->
   <a class="carousel-control-prev" href="#demo" data-slide="prev">
     <span class="carousel-control-prev-icon"></span>
   </a>
   <a class="carousel-control-next" href="#demo" data-slide="next">
     <span class="carousel-control-next-icon"></span>
   </a>
 </div>
 </div>

 </body>
 </html>
```

7.3 Database Schema (if Applicable):

Schema name: ibmproj

1. SQL query to store the registered users:

```
CREATE DATABASE IF NOT EXISTS `ibmproj` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
USE `ibmproj`;
CREATE TABLE IF NOT EXISTS `accounts` (
	`id` int(11) NOT NULL AUTO_INCREMENT,
	`username` varchar(50) NOT NULL,
	`password` varchar(255) NOT NULL,
	`email` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;
```

2. SQL query to create a table to store the uploaded and predicted images:

```
USE `ibmproj`;
CREATE TABLE IF NOT EXISTS imagetable (
  id int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL,
  photo blob NOT NULL,
  pred VARCHAR(50),
  date_entry TIMESTAMP DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

8. TESTING

8.1 Test Cases:

User Test Cases

Feature Type	Component	Test Scenario	Steps to Execute	Test Data	Expected Result	Actual Result	Status
Functional	Login page	Verify that user is able to login	1) Enter URL and click go 2) Enter valid username and password	Username: shreya Password: 1234	If entered credentials are valid, the user is navigated to the home page, else an error message is displayed	Working as expected	Pass
Functional	Registration page	Verify that user is able to register	1) Enter URL and click go 2) Enter valid username, email password	Username: shreya_ananth email: bhvpadshrey@gmail.com. Password: 1234	If entered credentials are valid, the user is registered, else an error message is displayed	Working as expected	Pass
Functional	Registration page	Verify that user gets confirmation mail	1) Enter URL and click go 2) Enter valid username, email password	Username: shreya_ananth email: bhvpadshrey@gmail.com. Password: 1234	If entered credentials are valid, the user is registered, and gets a confirmation email. Else an error message is displayed and an email isn't sent	Working as expected	Pass
Functional	Home Page	Verify that user is able to get prediction results	1) Click browse button 2) Select the file 3) Click upload 4) Click predict	Any jpeg or png file in the system	If it was submitted properly, the name of the species appears below the image	Working as expected	Pass
Functional	User history page	Verify user is able to access the search history	1) Click the history button on the menu bar	-	If history exists, the images are displayed in a carousel, else the user is navigated to an error page	Working as expected	Pass
UI	User history page	Verify user is able to browse the search history	1) Click the history button on the menu bar	-	The user should be able to browse the carousel using the arrows	Working as expected	Pass
Function	Menu Bar	Verify that user is able to logout	1) Click the logout button	-	User should be navigated back to the login page	Working as expected	Pass
Functional	Home Page	Verify that user is able to upload pictures	1) Click browse button 2) Select the file 3) Click upload	Any jpeg or png file in the system	If it got uploaded, the name of the file appears	Working as expected	Pass

8.2 User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Digital Naturalist project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Low Severity	Medium Severity	High Severity	Subtotal of bugs
By UI	1	2	2	5
By Functionality	0	2	2	4
Duplicate	0	4	7	11
External	0	0	0	0
Fixed	1	4	4	9
Not Reproduced	0	0	0	0
Skipped	0	0	0	0
Won't Fix	0	0	0	0
Totals	2	16	15	29

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
User Interface	1	0	0	1
Flask Application	5	0	0	5
Exception Reporting	1	0	0	1
Final Report Output	1	0	0	1

9. RESULTS

9.1 Performance Metrics:

Accuracy: 0.7232704162597656

10. ADVANTAGES & DISADVANTAGES

Advantages:

- Reduces manual work of searching
- Provides accurate results even for rare species
- Stores the predicted results which could be referred later

Disadvantages:

- Need for a stable internet connection
- Images being uploaded should have good clarity

11. CONCLUSION

The application developed using flask serves as a robust tool for recognition of different species of flora and fauna on the go. An additional feature of user confirmation through email provides an easier way for users to sign in to the application and storage of prediction results under user history helps them to access their searches faster. Being able to identify the flora and fauna around us often leads to an interest in protecting wild spaces, and collecting and sharing information about the species we see on our travels is very useful for conservation groups like NCC.

12. FUTURE SCOPE

- This project is focused on recognizing a limited number of species of each category.
- In future, this project can be extended to recognise many other species with the help of a carefully crafted dataset.
- This project can be extended to provide more detailed information about each instance of a living being like places where they are commonly found, eating habits, etc.

13. APPENDIX

Source Code:

IBM-Project-29907-1660133639/Final Deliverables/Final-Code at main · IBM-EPBL/IBM-Project-29907-1660133639 (github.com)

GitHub & Project Demo Link:

Github repository: IBM-EPBL/IBM-Project-29907-1660133639: Digital Naturalist - AI Enabled tool for Biodiversity Researchers (github.com)

Project Demo Link:

Demo Link