Project Documentation

Team ID- 592716

1.Introduction

1.1 Project Overview

The FelinAI Taxonomy Classification project is a cutting-edge initiative that harnesses the power of artificial intelligence and machine learning to automate the classification and categorization of Felis (cat) species. It is designed to provide an accurate and efficient means of identifying Felis species based on diverse data inputs, including images and textual descriptions. The project's core objectives encompass several critical aspects, such as the development of highly accurate machine learning models for species classification, the curation of a comprehensive dataset through effective data collection and preprocessing, the creation of an intuitive and user-friendly interface for interaction, and the implementation of robust security measures to ensure data privacy and system integrity. One of the project's key features is its commitment to scalability, with a cloud-based architecture that can seamlessly adapt to increasing workloads and user interactions.

The FelinAI Taxonomy Classification project aims to serve a diverse range of beneficiaries, including zoologists, researchers, and wildlife enthusiasts who seek to efficiently and accurately identify Felis species. applications, making it an accessible and indispensable tool for both professionals and enthusiasts interested in taxonomy.

1.2 Purpose

The primary purpose of the FelinAI Taxonomy Classification project is to leverage artificial intelligence and machine learning to automate the precise classification and categorization of Felis (cat) species. By utilizing AI models, the project aims to provide an accurate and efficient tool for researchers, zoologists, wildlife enthusiasts, and developers to identify and understand Felis species, whether it be through images, textual descriptions, or a combination of data sources. The project's core purpose is to simplify and enhance the process of Felis taxonomy classification, contributing to the fields of biology, wildlife conservation, and education, while also offering practical applications for hobbyists and developers seeking to work with Felis species data. Moreover, the project's focus on scalability and security ensures its adaptability to growing user demands and the safeguarding of sensitive information. Ultimately, the FelinAI project serves as a valuable resource for automating the taxonomy classification of Felis species while promoting ongoing improvements through user feedback and AI refinement.

2. Literature Survey

2.1 Existing Problem

The existing problem addressed by the FelinAI Taxonomy Classification project is the manual and time-consuming process of categorizing and classifying Felis (cat) species. Traditionally, this task is performed by zoologists, researchers, and wildlife enthusiasts through visual inspection and reference to extensive species descriptions. This method is not only labor-intensive but also prone to errors and subjectivity. The existing problem lies in the need for an accurate and efficient automated solution that can handle a vast amount of Felis taxonomy data from various sources, including images and textual descriptions. It also involves addressing the limitations of current systems that lack scalability and security measures to protect sensitive data.

2.2 References

References from A revised taxonomy of the Felidae: The final report of the Cat Classification Task Force of the IUCN Cat Specialist Group.
Felines of the world: discoveries in taxonomic classification and history.
Examining the research taxonomy of artificial intelligence, deep learning & machine learning —a bibliometric analysis

In their extensive work on Felidae taxonomy, the 'A revised taxonomy of the Felidae: The final report of the Cat Classification Task Force of the IUCN Cat Specialist Group' offers a comprehensive reclassification of Felidae species, providing invaluable insights into the classification of various Felis species. Similarly, 'Felines of the world: discoveries in taxonomic classification and history' contributes to our understanding of the historical aspects of Felidae classification. These references have been pivotal in shaping our knowledge of Felis taxonomy.

In the realm of artificial intelligence and machine learning, the study 'Examining the research taxonomy of artificial intelligence, deep learning & machine learning —a bibliometric analysis' conducts a bibliometric analysis, shedding light on the evolving landscape of AI, deep learning, and machine learning research. This research underscores the dynamic nature of the field and the need for up-to-date and evolving taxonomies.

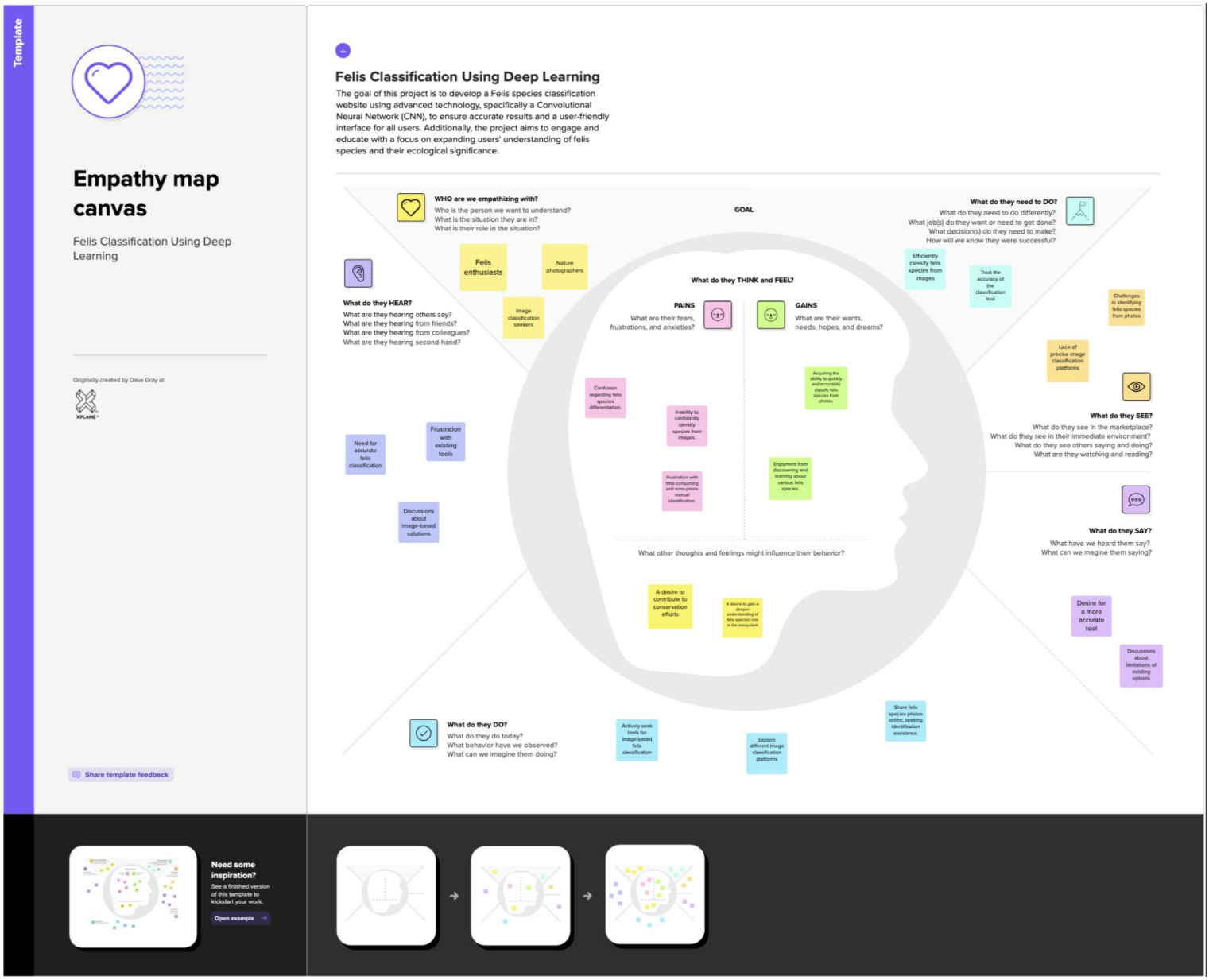
2.3 Problem Statement

The problem statement for the FelinAI Taxonomy Classification project is to create an advanced AI and machine learning system capable of automatically classifying and categorizing Felis species with a high degree of accuracy. This system should be able to handle diverse data inputs, including images and text, and provide users with an efficient means of identifying and understanding Felis species.

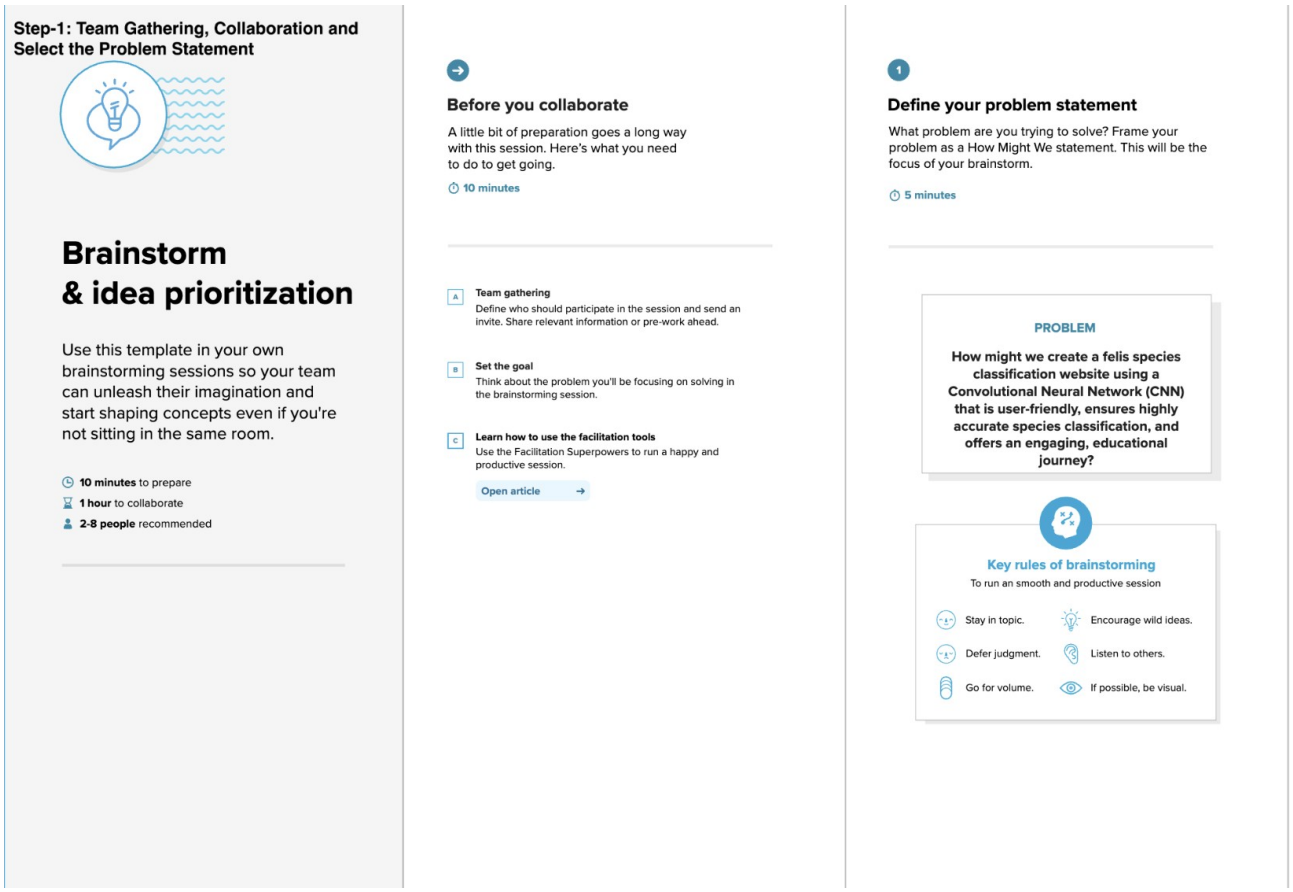
The key challenge is to develop machine learning models that can effectively process and classify this data, while ensuring data security and scalability to accommodate increasing workloads. The project aims to alleviate the burden of manual classification and provide a practical and accessible tool for a wide range of users, from researchers and zoologists to wildlife enthusiasts and developers, ultimately streamlining the Felis taxonomy classification process.

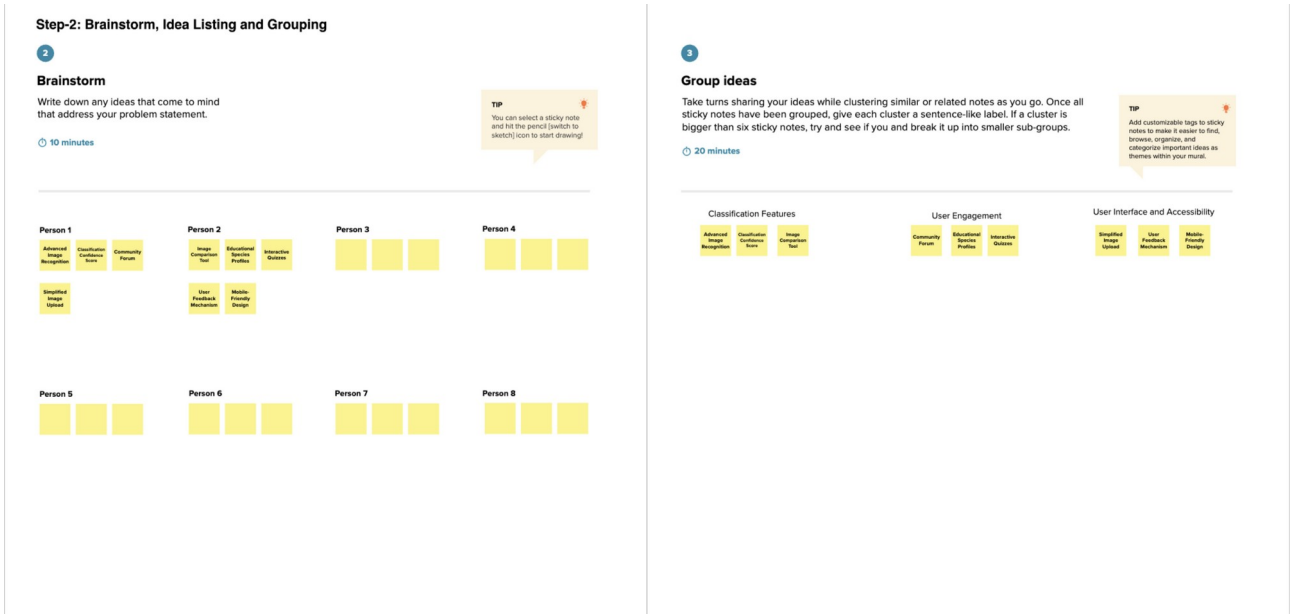
3. Ideation & Proposed Solution

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming





Step-3: Idea Prioritization

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

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

4. Requirement Analysis

4.1 Functional Requirements

Functional requirements describe what the system should do, specifying its intended behaviour, features, and capabilities. They are often detailed in terms of user interactions and system responses.

- Use Cases: Defining specific use cases that outline the interactions between users and the system. For example, user registration, data classification, and report generation.
- User Stories: Describing the system's features and functionalities from the perspective of end-users. User stories provide a more user-centric view of requirements.
- Input and Output: Specifying the data inputs required by the system and the expected outputs or results for each function. For instance, what information is needed for Felis species classification, and what results are returned to the user?
- Data Processing: Detailing how data will be processed and manipulated within the system. For FelinAI, this includes data collection, feature extraction, model training, and post-processing.
- Interactions: Defining how different system components and modules interact with each other. For example, how the user interface communicates with the machine learning models.

- Error Handling: Specifying how the system should handle errors, exceptions, and unexpected behaviour. This includes defining error messages and how the system should respond to invalid input.

4.2 Non-Functional Requirements:

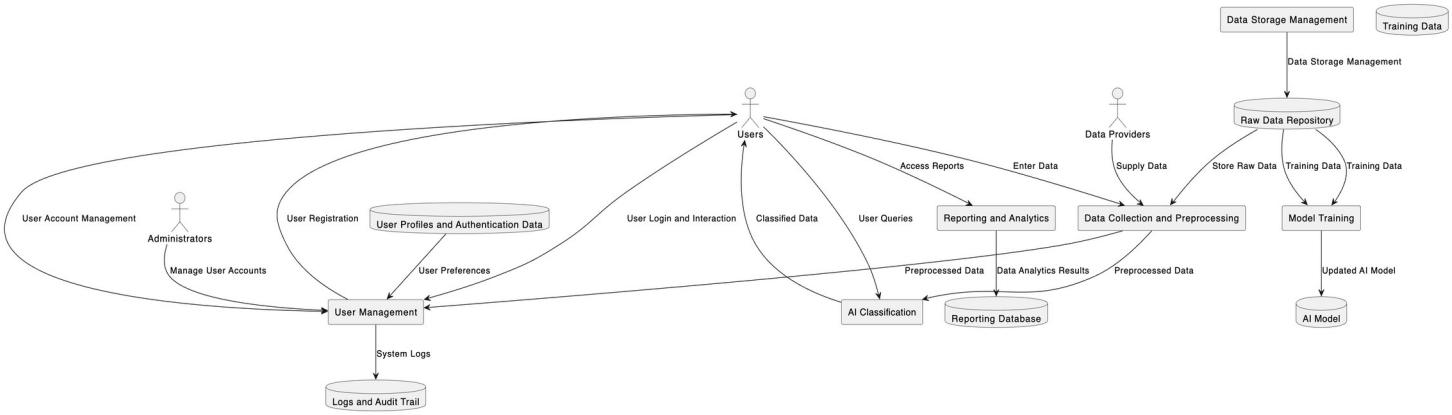
Non-functional requirements focus on the qualities and characteristics of the system, ensuring that it performs effectively, securely, and reliably.

- Performance: Defining performance requirements such as response times, throughput, and resource utilization. For FelinAI, this might include specifying that image classification should take no longer than a certain number of seconds.
- Scalability: Describing how the system can scale to accommodate increased loads and data volumes. This is crucial for a project like FelinAI that may experience fluctuations in usage.
- Security: Specifying the security measures required to protect user data, ensure data privacy, and prevent unauthorized access. For example, encryption standards and access controls.
- Availability: Defining the system's availability requirements, including uptime, redundancy, and failover mechanisms. FelinAI should be available to users with minimal downtime.
- Usability: Detailing the user interface's ease of use, accessibility, and user experience considerations. Ensure the system is intuitive and user-friendly.

- Regulatory Compliance: Addressing any industry or legal regulations that the system must comply with, such as data protection laws for user data handling.
- Reliability: Specifying how the system should minimize errors, failures, and data loss. This could include backup and recovery strategies for critical data.
- Maintainability: Describing how the system should be maintained, updated, and extended. This ensures that future improvements and changes can be made efficiently.
- Documentation: Including documentation requirements for developers, users, and administrators, facilitating system understanding and maintenance.
- Testing and Quality Assurance: Defining testing criteria and quality assurance processes to ensure that the system meets the defined requirements and standards.

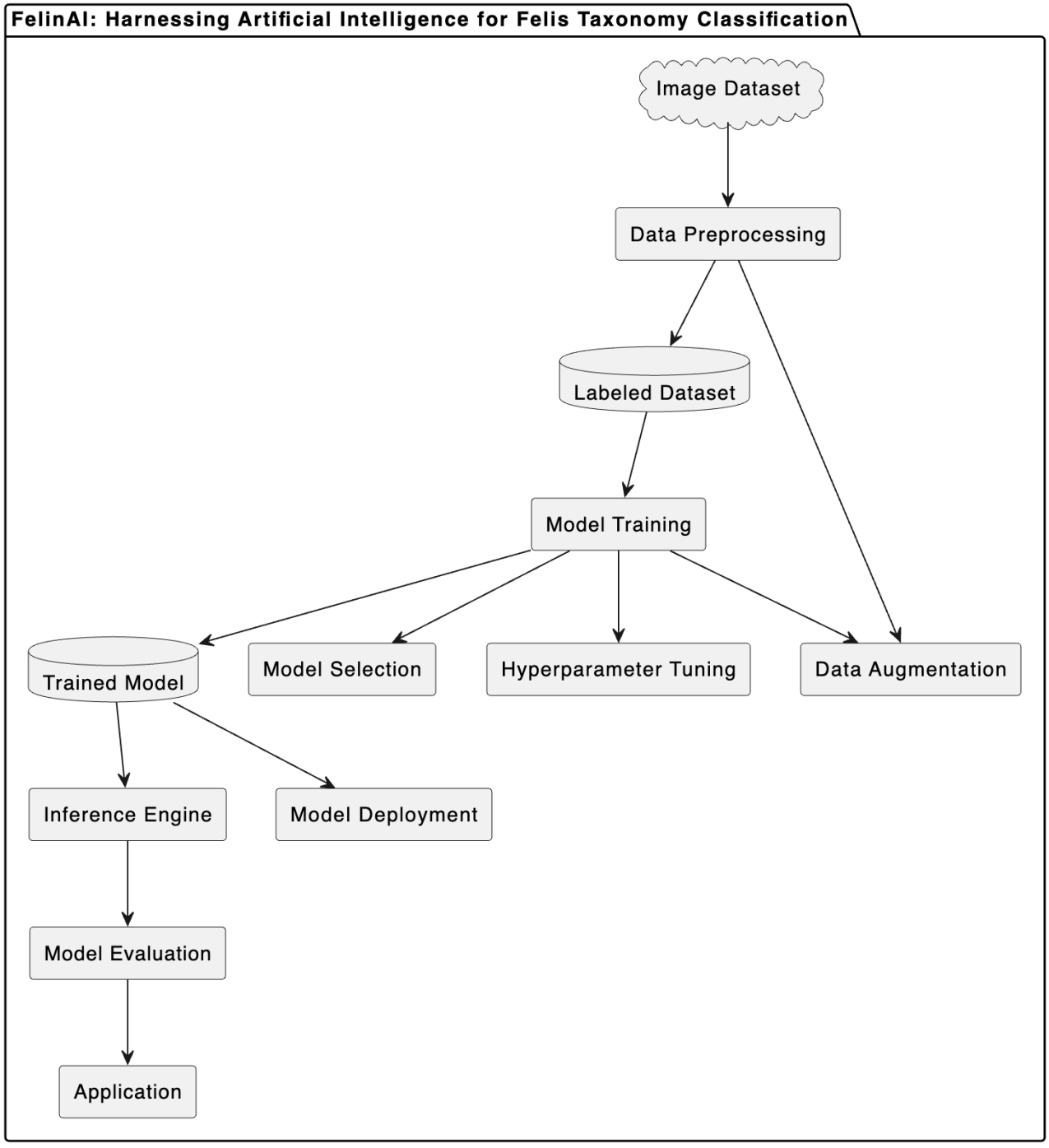
5. Project Design

5.1 Data Flow Diagram and User Stories



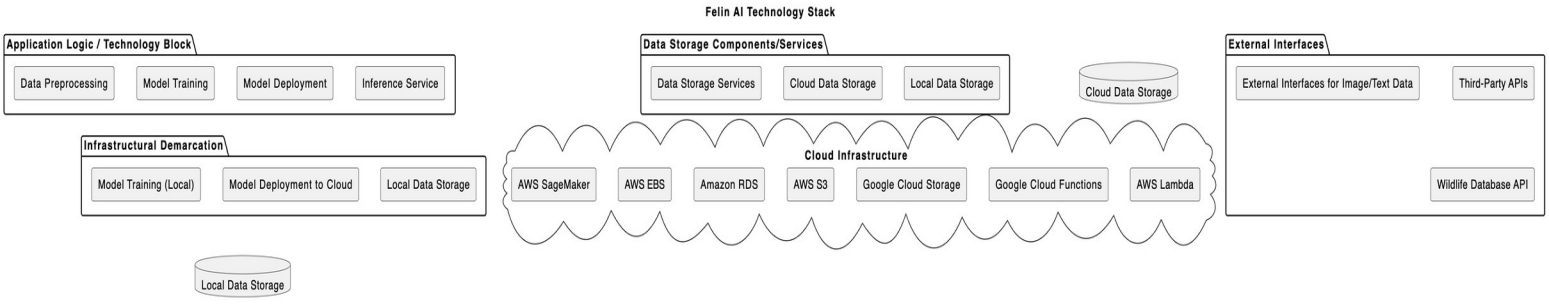
User Type	Function Requirement	User Story Number	User Story	Acceptance Criteria	Priority	Release
Administrator	User Management	USN-1	As an administrator, I can add, edit, or delete user accounts and assign roles.	I can access the user management panel, create new user accounts with roles	High	Sprint-3
Data Provider	Data Review Request	USN-2	As a data provider, I can request a review of my submitted data.	I can select specific data entries for review and provide explanations for the review request	Medium	Sprint-2
Data Reviewer	Data Review	USN-3	As a data reviewer, I can review and validate the accuracy of Felis taxonomy data.	I can access the data for review and approve or reject the data based on accuracy and completeness.	High	Sprint-2
Taxonomist	Feedback	USN-4	As a taxonomist, I can provide feedback on the accuracy and quality of classification results.	I can submit feedback with comments and ratings.	Low	Sprint-4
Researcher	Search	USN-5	As a researcher, I can search for Felis taxonomy information using keywords or criteria.	I can enter search terms and filter criteria.	High	Sprint-1
Data Analyst	Data Export	USN-6	As a data analyst, I can export Felis taxonomy classification data for further analysis.	I can select specific datasets or classifications for export.	Medium	Sprint-3

5.2 Solution Architecture



6. Project Planning and Scheduling

6.1 Technical Architecture



The technical architecture of the FelinAI Taxonomy Classification project is a comprehensive framework designed to automate the classification of Felis species using advanced artificial intelligence and machine learning techniques. It encompasses a series of interconnected layers that begin with data collection and preprocessing, where raw Felis taxonomy data is curated and prepared. Subsequently, feature extraction processes extract pertinent features from the data, which is then utilized for training machine learning models. These models are at the heart of the system, employing extensive datasets to accurately classify Felis species. To ensure accessibility and scalability, the trained models are deployed to a cloud-based environment, where they can be accessed through both a user-friendly web interface and an API, allowing developers to integrate FelinAI into their applications. Post-processing and visualization steps refine classification results, and robust security implementations safeguard user data and system integrity.

6.2 Sprint Planning and Estimation

Sprint planning is a crucial step in managing the FelinAI project effectively. It helps us set the direction for the upcoming sprint and allocate tasks based on priority and capacity.

- Product Backlog Review: The Product Owner begins by presenting the product backlog. This backlog comprises user stories and tasks related to improving Felin AI's taxonomy classification, user interface, and security features.
- Sprint Goal: The team collectively defines the sprint goal, which serves as the overarching objective for the upcoming sprint. This goal aligns our efforts and provides a clear purpose for the work ahead, such as improving classification accuracy or enhancing security.
- User Story Selection: The team, which includes data scientists, developers, UI/UX designers, and other relevant members, collaboratively selects user stories from the backlog. Our selection is based on the priority of the tasks and our assessment of their feasibility.
- Task Breakdown: For each user story, we break down the work into smaller tasks or sub-tasks. This detailed breakdown helps us understand the specific work involved, such as data collection, model fine-tuning, or UI enhancements.

- Estimation: We use story points as our estimation unit to assess the effort required for each task. Estimation helps us determine the complexity of tasks and ensure we don't overcommit during the sprint.
- Definition of Done (DoD): We establish a clear definition of "done" for each task. This includes setting acceptance criteria and conditions that must be met for a task to be considered complete.
- Capacity Assessment: The team evaluates its capacity for the sprint, considering factors such as team size, member availability, and any external events or holidays that might affect work hours.
- Commitment: Based on our estimation and capacity assessment, we make a commitment to complete a set of tasks during the sprint. These committed tasks form the sprint backlog.

Estimation Methodology in FelinAI:
In the FelinAI project, we use story points as our primary estimation unit. Story points help us gauge the relative complexity of tasks and allow for more accurate planning. Here's how we implement estimation:

- Team Collaboration: Estimation is a team effort. During estimation meetings, team members from various roles, including data scientists, developers, and UI/UX designers, come together to assess the complexity of tasks.

- Comparative Estimation: We compare tasks to a reference task, often referred to as a "baseline" task. For instance, if our baseline task is assigned 1 story point, a task estimated as 3 story points should be about three times more complex.
- Factors Consideration: We take into account various factors when estimating tasks. For example, the complexity of fine-tuning machine learning models, the volume of data to collect, the intricacy of UI improvements, or the implementation of security features.
- Open Discussion: Estimation meetings encourage open discussion and clarification. Team members share their perspectives on what makes a task more or less complex, leading to a shared understanding of the work involved.
- Consensus Building: We aim for consensus during the estimation process. If there's disagreement on a task's estimated size, we discuss the factors contributing to our estimates until we reach a shared understanding.
- Historical Data: We continuously collect data on the actual time and effort required to complete tasks with different story point estimates. This historical data informs our future estimations, making them more accurate.
- Velocity Metric: We use the velocity metric to track our team's productivity and capacity. By analysing the story points of completed tasks in previous sprints, we can set realistic expectations for future commitments.

6.3 Sprint Delivery Schedule

A sprint delivery schedule is crucial for ensuring that the work planned for the sprint is completed efficiently and delivered on time. In the context of the FelinAI project, we maintain a structured approach to manage sprint delivery:

- Sprint Kickoff: We begin the sprint with a kickoff meeting where we review the sprint goal, the selected user stories and tasks, and the definition of done for each task. We reconfirm our commitment to the sprint's scope.
- Task Assignment: During the kickoff meeting, we assign tasks to individual team members based on their expertise and availability. Each of us takes responsibility for specific tasks within the sprint backlog.
- Daily Stand-up Meetings: Throughout the sprint, we conduct daily stand-up meetings to track progress. We share updates on the tasks we are working on, discuss any challenges, and identify potential roadblocks. This ensures that tasks stay on track.
- Task Tracking: We use a project management tool or board (e.g., Kanban or Scrum board) to visually track the progress of tasks. Tasks move through various stages (e.g., "To Do," "In Progress," "Testing," "Done") as we work on and complete them.

- Continuous Collaboration: We maintain open communication and collaboration. If any of us face obstacles or require assistance, we can reach out to others for support. Collaboration fosters efficiency and problem-solving.
- Testing and Quality Assurance: As tasks are completed, they undergo testing and quality assurance to ensure that they meet the predefined acceptance criteria and the definition of done.
- Incremental Delivery: Throughout the sprint, we focus on incremental delivery. This means that as individual tasks are completed, they are potentially shippable increments of the product. This allows for flexibility in delivering valuable features early in the sprint.
- Sprint Review: At the end of the sprint, we conduct a sprint review meeting to showcase the work completed during the sprint. This is an opportunity for stakeholders to provide feedback and validate the delivered features.
- Sprint Retrospective: Following the sprint review, we hold a retrospective meeting to reflect on our accomplishments, discuss what went well, and identify areas for improvement in our sprint delivery process.

7. Coding and Solutioning

7.1 Feature 1

The predict feature of our website is a pivotal component that empowers users with the ability to identify felis species swiftly and accurately. Leveraging a cutting-edge Convolutional Neural Network (CNN), this feature ensures a high level of precision in classifying images, aiding wildlife enthusiasts, researchers, and the curious in their pursuit of species recognition. Users can effortlessly upload images, and within moments, the system provides a definitive classification, thus eliminating the need for complex manuals or extensive research. With the predict feature, we bring a user-friendly, accessible, and informative tool to the forefront, enriching the experience of felis species exploration for all.

```python
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.applications import VGG16
from tensorflow.keras.optimizers import Adam
from keras.callbacks import EarlyStopping

# Define your class names
class_names = ['african-wildcat', 'blackfoot-cat', 'chinese-mountain-cat', 'domestic-cat', 'european-wildcat',
               'jungle-cat', 'sand-cat']

train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.3
)

training_set = train_datagen.flow_from_directory(
    'dataset-1',
    target_size=(227, 227),
    batch_size=32,
    subset='training',
    class_mode='categorical',
    seed=64
)

test_set = train_datagen.flow_from_directory(
    'dataset-1',
    target_size=(227, 227),
    batch_size=32,
    subset='validation',
    class_mode='categorical',
    seed=64
)

vgg_model = Sequential()
vgg = VGG16(include_top=False, weights='imagenet', input_shape=(227, 227, 3))

vgg.trainable = True
for layer in vgg.layers[:15]:
    layer.trainable = False

vgg_model.add(vgg)
vgg_model.add(GlobalAveragePooling2D())
vgg_model.add(Dense(units=512, activation='relu'))
vgg_model.add(Dropout(0.5))  # Increase dropout rate
vgg_model.add(Dense(units=len(class_names), activation='softmax'))  # Adjust output units

import keras
opt = Adam(learning_rate=0.0001)  # Adjust learning rate
vgg_model.compile(optimizer=opt, loss=keras.losses.categorical_crossentropy, metrics=['accuracy'])

# Implement early stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)  # Increase patience

vgg_model.fit(training_set, batch_size=8, epochs=50, validation_data=test_set, callbacks=[early_stopping])
```

```python
def load_model():
    class_names = [
        "african-wildcat",
        "blackfoot-cat",
        "chinese-mountain-cat",
        "domestic-cat",
        "european-wildcat",
        "jungle-cat",
        "sand-cat",
    ]

    global model
    model = Sequential()

    vgg = VGG16(include_top=False, weights="imagenet", input_shape=(227, 227, 3))

    vgg.trainable = True
    for layer in vgg.layers[:15]:
        layer.trainable = False

    model.add(vgg)
    model.add(GlobalAveragePooling2D())
    model.add(Dense(units=512, activation="relu"))
    model.add(Dropout(0.5))
    model.add(Dense(units=len(class_names), activation="softmax"))

    opt = Adam(learning_rate=0.0001)
    model.compile(
        optimizer=opt, loss=keras.losses.categorical_crossentropy, metrics=["accuracy"]
    )

    model.load_weights("./weights.h5")


load_model()
```

```python
def process_image(file_path):
    class_names = [
        "african-wildcat",
        "blackfoot-cat",
        "chinese-mountain-cat",
        "domestic-cat",
        "european-wildcat",
        "jungle-cat",
        "sand-cat",
    ]

    img = load_img(file_path)
    img = img.resize((227, 227))
    x = img_to_array(img)
    x = np.expand_dims(x, axis=0)
    y_pred = model.predict(x)
    class_idx = np.argmax(y_pred, axis=1)[0]
    class_name = class_names[class_idx]
    result = class_name.split("-")
    result = result[0].capitalize() + " " + result[1].capitalize()
    return result
```

```python
@app.route("/predict", methods=["GET", "POST"])
def predict():
    file_path = "./static/images/file.jpg"
    if request.method == "POST":
        file = request.files["image"]
        if file:
            file.save(file_path)
            result = process_image((file_path))

            return render_template("predict.html", result=result)

    if os.path.exists(file_path):
        os.remove(file_path)
    return render_template("predict.html")
```

7.2 Feature 2

In addition to the predict feature, our website offers a comprehensive "Discover" section. This feature serves as an educational resource, providing users with detailed information on various felis species. Each species is presented with its unique characteristics, habitat details, conservation status, and physical attributes, allowing users to delve deeper into the world of felis species. Whether you're a passionate feline enthusiast, a budding researcher, or simply intrigued by these fascinating creatures, the Discover feature fosters a deeper understanding of felis species and their ecological significance. It aims to educate, engage, and inspire users in their exploration of the diverse felis species inhabiting our planet.

```
<!doctype html>
<html lang="en">
    <head>
        <title></title>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />

        <link
            href="{{ url_for('static', filename='css/styles.css') }}"
            rel="stylesheet"
        />
        <link
            href="{{ url_for('static', filename='css/discover.css') }}"
            rel="stylesheet"
        />
    </head>
    <body>_
        {% include 'navbar.html' %}
        <div class="discover">
            <h1>Meet Our Model's Recognised Felis Species</h1>

            <div class="species">
                <div class="species-info">
                    <h3>Domestic Cat</h3>

                    <p class="italics">Felis catus</p>
                    <p>
                        Domestic cats, scientifically referred to as Felis
                        catus, are beloved companions known for their diverse
                        breeds, colors, and personalities. They have a long
                        history of living alongside humans, combining
                        independence with affection.
                    </p>
                    <p>
                        <span class="bold">Habitat: </span>
                        Domestic cats live in a wide range of environments
                        worldwide, from urban areas to rural settings.
                    </p>
                    <p>
                        <span class="bold"> Size: </span>
                        Domestic cats come in various sizes, depending on their
                        breed. <br />

                        <span class="bold">Coat: </span>
                        The coat of domestic cats varies greatly, with a wide
                        range of colors, patterns, and lengths. <br />
                        <span class="bold"> Ears: </span>
                        Ears can be upright or folded, depending on the breed.
                    </p>

                    <p>
                        <span class="bold">Conservation Status:</span>
                        Not applicable (due to their domestication)
                    </p>
                </div>
                <div class="species-img">
                    <img
                        src="{{ url_for('static', filename='images/domestic_cat.jpg')}}"
                        alt="Species 1"
                    />
                </div>
            </div>
            <div class="species">
                <div class="species-info">
                    <h3>African Wildcat</h3>
```

```
            <p>
                The African Wildcat, scientifically known as Felis
                lybica, is a small, slender cat with a sandy coat
                adorned with subtle stripes and spots. These skilled
                hunters are known for their adaptability, allowing them
                to thrive in diverse African habitats.
            </p>
            <p>
                <span class="bold">Habitat: </span>
                African Wildcats are versatile, inhabiting various
                environments across Africa, from savannas and deserts to
                forests.
            </p>
            <p>
                <span class="bold"> Size: </span>
                African Wildcats are small to medium-sized cats, with
                males typically larger than females.

                <span class="bold">Coat: </span>
                They have a short, sandy-colored coat adorned with faint
                stripes and spots.
                <span class="bold"> Ears: </span>
                Their ears are relatively short compared to some other
                wildcats.
            </p>

            <p>
                <span class="bold">Conservation Status:</span>
                Least Concern
            </p>
        </div>
        <div class="species-img">
            <img
                src="{{ url_for('static', filename='images/african_wildcat.jpg')}}"
                alt="Species 1"
            />
        </div>
    </div>
    <div class="species">
        <div class="species-info">
            <h3>Blackfoot Cat</h3>

            <p class="italics">Felis lybica</p>
            <p>
                The African Wildcat, scientifically known as Felis
                lybica, is a small, slender cat with a sandy coat
                adorned with subtle stripes and spots. These skilled
                hunters are known for their adaptability, allowing them
                to thrive in diverse African habitats.
            </p>
            <p>
                <span class="bold">Habitat: </span>
                African Wildcats are versatile, inhabiting various
                environments across Africa, from savannas and deserts to
                forests.
            </p>
            <p>
                <span class="bold"> Size: </span>
                African Wildcats are small to medium-sized cats, with
                males typically larger than females.

                <span class="bold">Coat: </span>
                They have a short, sandy-colored coat adorned with faint
                stripes and spots.
```

```
            </p>

            <p>
                <span class="bold">Conservation Status:</span>
                Near Threatened
            </p>
        </div>
        <div class="species-img">
            <img
                src="{{ url_for('static', filename='images/blackfooted_cat.jpg')}}"
                alt="Species 1"
            />
        </div>
    </div>
    <div class="species">
        <div class="species-info">
            <h3>Chinese Mountain Cat</h3>

            <p class="italics">Felis bieti</p>
            <p>
                The Chinese Mountain Cat, or Felis bieti, thrives in the
                challenging mountainous environments of China. Their
                dense fur and unique appearance help them adapt to harsh
                conditions. They are solitary and elusive hunters.
            </p>
            <p>
                <span class="bold">Habitat: </span>
                Chinese Mountain Cats are native to high-altitude
                regions of China.
            </p>
            <p>
                <span class="bold"> Size: </span>
                Chinese Mountain Cats are small cats, with stocky
                bodies.

                <span class="bold">Coat: </span>
                Their fur is thick and soft, often grayish or pale brown
                with dark spots.
                <span class="bold"> Ears: </span>
                They have distinctive, rounded ears with small tufts of
                hair.
            </p>

            <p>
                <span class="bold">Conservation Status:</span>
                Vulnerable
            </p>
        </div>
        <div class="species-img">
            <img
                src="{{ url_for('static', filename='images/chinese_mountain_cat.jpg')}}"
                alt="Species 1"
            />
        </div>
    </div>
    <div class="species">
        <div class="species-info">
            <h3>European Wild Cat</h3>

            <p class="italics">Felis silvestris silvestris</p>
            <p>
                The European Wildcat, or Felis silvestris silvestris,
                closely resembles domestic cats but retains its wild
                instincts. They are known for their solitary nature and
                excellent hunting skills.
```

```
            <p>
                <span class="bold">Habitat: </span>
                European Wildcats inhabit forests and woodlands in
                Europe.
            </p>
            <p>
                <span class="bold"> Size: </span>
                European Wildcats closely resemble domestic cats in
                size.

                <span class="bold">Coat: </span>
                Their fur is dense and often grayish with a striped
                pattern.
                <span class="bold"> Ears: </span>
                They have typical cat ears with a small amount of
                tufting.
            </p>

            <p>
                <span class="bold">Conservation Status:</span>
                Least Concern
            </p>
        </div>
        <div class="species-img">
            <img
                src="{{ url_for('static', filename='images/european_wildcat.jpg')}}"
                alt="Species 1"
            />
        </div>
    </div>
    <div class="species">
        <div class="species-info">
            <h3>Jungle Cat</h3>

            <p class="italics">Felis chaus</p>
            <p>
                The Jungle Cat, scientifically known as Felis chaus, is
                a medium-sized wildcat known for its agility and hunting
                prowess. They are skilled hunters that adapt to their
                diverse habitats.
            </p>
            <p>
                <span class="bold">Habitat: </span>
                Jungle Cats thrive in wetlands, grassy areas, and
                forests in regions including Asia and the Middle East.
            </p>
            <p>
                <span class="bold"> Size: </span>
                Jungle Cats are medium-sized cats with a slender build.

                <span class="bold">Coat: </span>
                Their coat is usually tawny or grayish with dark spots
                and stripes.
                <span class="bold"> Ears: </span>
                They have relatively long and pointed ears.
            </p>

            <p>
                <span class="bold">Conservation Status:</span>
                Least Concern
            </p>
        </div>
        <div class="species-img">
            <img
                src="{{ url_for('static', filename='images/jungle_cat.jpg')}}"
```

FelinAI Discover Predict Contact FAQ

A Modern Approach to Cat Classification: Embracing AI for Precision and Insight

Try

FelinAI Discover Predict Contact FAQ

About FelinAI

At FelinAI, we are passionate about cats and cutting-edge technology. Our journey began with a shared love for feline companions and a vision to merge our enthusiasm with the power of artificial intelligence. Our mission is simple yet profound: to enhance the understanding of cat species through advanced AI classification.

Our team has meticulously trained our AI system to recognize and classify seven specific cat breeds, including the African Wildcat, Blackfoot Cat, Chinese Mountain Cat, Domestic Cats, European Wildcat, Jungle Cat, and Sand Cat. While we celebrate our current achievements, we're committed to expanding our breed recognition capabilities in the future.

Join us in this exciting venture where the world of cats converges with the realm of AI. Whether you're a cat enthusiast, a dedicated researcher, or simply curious about

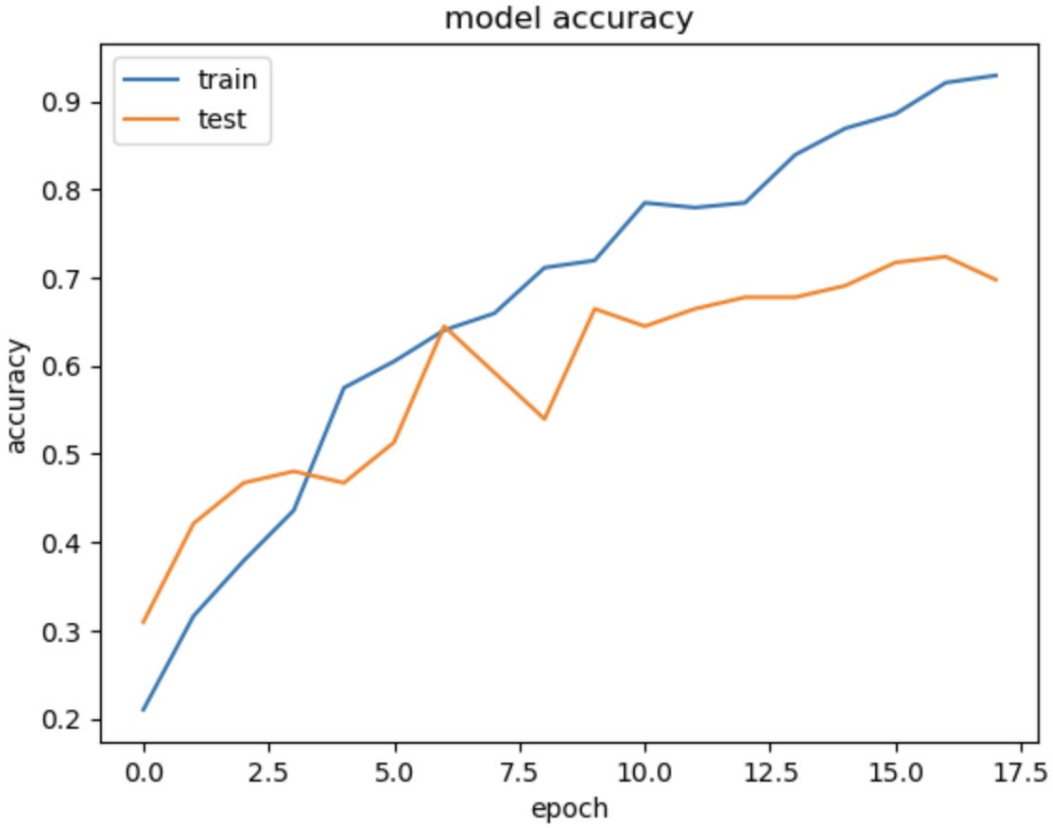
8. Performance Testing

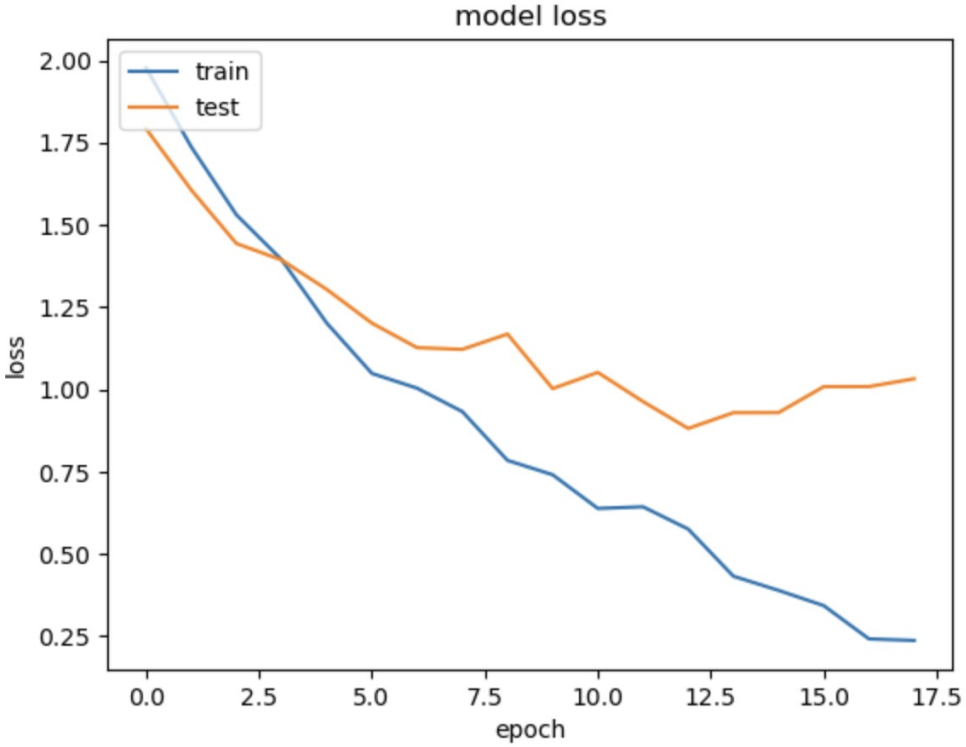
8.1 Performance Metrics

S.No	Parameter	Values	Screenshot

1.	Model Summary		`vgg_model.summary()`
			`Model: "sequential_1"`
			`_____`
			`Layer (type) Output Shape Param #`
			`=====`
			`vgg16 (Functional) (None, 7, 7, 512) 14714688`
			`global_average_pooling2d_1 (GlobalAveragePooling2D) (None, 512) 0`
			`dense_2 (Dense) (None, 512) 262656`
			`dropout_1 (Dropout) (None, 512) 0`
			`dense_3 (Dense) (None, 7) 3591`
			`=====`
			`Total params: 14980935 (57.15 MB)`
			`Trainable params: 7345671 (28.02 MB)`
			`Non-trainable params: 7635264 (29.13 MB)`
			`_____`

2.	Accuracy	Training Accuracy
		
		Validation Accuracy
			`Epoch 11/50`
			`12/12 [==========] - 113s 10s/step - loss: 0.6692 - accuracy: 0.7302 - val_loss: 1.0636 - val_accuracy: 0.6118`
			`Epoch 12/50`
			`12/12 [==========] - 119s 10s/step - loss: 0.5372 - accuracy: 0.8093 - val_loss: 1.0393 - val_accuracy: 0.6776`
			`Epoch 13/50`
			`12/12 [==========] - 111s 9s/step - loss: 0.4993 - accuracy: 0.8256 - val_loss: 0.9891 - val_accuracy: 0.6776`
			`Epoch 14/50`
			`12/12 [==========] - 104s 9s/step - loss: 0.4266 - accuracy: 0.8774 - val_loss: 0.9555 - val_accuracy: 0.6908`
			`Epoch 15/50`
			`12/12 [==========] - 110s 10s/step - loss: 0.3613 - accuracy: 0.9019 - val_loss: 1.0150 - val_accuracy: 0.7105`
			`Epoch 16/50`
			`12/12 [==========] - 105s 9s/step - loss: 0.2981 - accuracy: 0.9074 - val_loss: 1.0401 - val_accuracy: 0.7434`
			`Epoch 17/50`
			`12/12 [==========] - 101s 9s/step - loss: 0.2324 - accuracy: 0.9183 - val_loss: 1.0407 - val_accuracy: 0.7368`
			`Epoch 18/50`
			`12/12 [==========] - 100s 8s/step - loss: 0.2719 - accuracy: 0.9210 - val_loss: 0.9512 - val_accuracy: 0.7303`
			`Epoch 19/50`
			`12/12 [==========] - 95s 8s/step - loss: 0.2421 - accuracy: 0.9183 - val_loss: 0.9829 - val_accuracy: 0.7500`
			`Epoch 20/50`
			`12/12 [==========] - 101s 9s/step - loss: 0.2351 - accuracy: 0.9128 - val_loss: 1.0870 - val_accuracy: 0.6974`
			`Epoch 21/50`
			`12/12 [==========] - 98s 8s/step - loss: 0.2330 - accuracy: 0.9264 - val_loss: 1.0552 - val_accuracy: 0.6776`
			`Epoch 22/50`
			`12/12 [==========] - 100s 8s/step - loss: 0.1531 - accuracy: 0.9564 - val_loss: 1.1341 - val_accuracy: 0.7303`
			`Epoch 23/50`
			`12/12 [==========] - 93s 8s/step - loss: 0.1180 - accuracy: 0.9591 - val_loss: 1.1339 - val_accuracy: 0.7237`
			
			`vgg_model.evaluate(test_set)`
			`5/5 [==========] - 17s 3s/step - loss: 0.9079 - accuracy: 0.7368`
			`[0.9079163074493408, 0.7368420958518982]`


model accuracy
train
test
accuracy
epoch
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


model loss
train
test
loss
epoch
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

10. Advantages and Disadvantages

Advantages:

- Accurate Species Identification: The Felis app provides highly accurate species identification through advanced technology, making it a valuable resource for researchers and enthusiasts.
- User-Friendly Design: The app's user-friendly interface ensures easy navigation, making it accessible to a wide range of users, from beginners to experts.
- Educational Value: With informative content and interactive features, the app serves as an educational tool, deepening users' understanding of felis species and their ecological significance.
- Engaging Content: Engaging multimedia content, such as images and descriptions, keeps users actively involved and interested in exploring different felis species.
- Convenient Access: The app's availability on various devices and platforms ensures that users can access felis species information wherever and whenever they need it.
- User Support: The app might offer responsive customer support to address user queries and concerns promptly.
- Regular Updates: The app may receive periodic updates, enhancing its functionality and ensuring it stays relevant and accurate.
- Continuous Improvement: User feedback may drive ongoing app enhancements, ensuring its evolution and better performance.

Disadvantages:

- Limited Species Coverage: The app might not cover all felis species, leaving out some rare or less-known breeds.
- Storage Space: High-quality images and data can take up significant storage space on users' devices.
- Hardware Requirements: Running the app may demand a device with adequate processing power, limiting access for older or less powerful devices.
- Internet Dependency: Some features may require a stable internet connection, making offline use challenging.
- Privacy Concerns: The collection of user data for various purposes can raise privacy and consent issues.
- Data Security: The storage and transmission of sensitive data could be susceptible to security breaches.
- Limited Interactivity: The app may lack advanced interactive features, potentially reducing user engagement.
- Compatibility Issues: It may not function optimally on all devices and operating systems, leading to compatibility problems.
- Resource-Intensive: The app could consume considerable system resources, potentially slowing down the user's device.

11. Conclusion

In the ever-evolving landscape of technology, our Felis species classification project stands as a testament to the remarkable fusion of science and nature. The objective was clear from the outset: to create a user-friendly website that accurately classifies felis species while simultaneously engaging and educating users. The journey began with a problem statement that underscored the need for a reliable platform for species identification. This guided us to employ advanced technology, particularly a Convolutional Neural Network (CNN), to achieve highly accurate results.

One of the project's foremost advantages is its ability to provide immediate and precise species classification, thereby assisting both wildlife enthusiasts and researchers in their endeavors. However, like any endeavor, it comes with its own set of challenges. The intricacies of machine learning, data management, and the need for continued updates are some of the notable disadvantages.

Looking ahead, the future scope of this project is vast. It includes expanding the range of identified felis species, enhancing machine learning algorithms, and exploring collaborative opportunities with the scientific community. As we move forward, we strive to develop a holistic understanding of felis species and their ecological significance.

12. Future Scope

- Additional Species Inclusion: Expand the app's database to include more felis species, increasing its coverage and accuracy.
- Machine Learning Enhancements: Incorporate advanced machine learning models for species identification, potentially improving accuracy.
- Custom Image Upload: Allow users to upload images of unidentified felis species, contributing to a crowdsourced database for classification.
- Localization: Implement localization features to make the app available in multiple languages, increasing its accessibility and reach.
- Conservation Information: Include educational content on felis species conservation status, threats, and ways to support conservation efforts.
- Augmented Reality (AR): Develop an AR feature for immersive species identification, offering an interactive and engaging experience.
- Interactive Maps: Integrate maps that show the distribution of felis species, allowing users to explore their habitats.
- Offline Content Enrichment: Enhance offline capabilities by providing comprehensive offline content, including images and descriptions.

- Collaborations: Partner with wildlife conservation organizations and researchers to access their data and contribute to ongoing felis species studies.

- Educational Modules: Create educational modules or quizzes within the app to enhance users' knowledge and engagement.

13. Appendix

Source Code: https://github.com/smartinternz02/SI-GuidedProject-600569-1697465238/tree/main/Project_Development_Phase

Github: https://github.com/smartinternz02/SI-GuidedProject-600569-1697465238

Project Demo Link: https://drive.google.com/file/d/1ecXJRu_1F_nkv6XZ_VJiLFArxo1v9VjC/view?usp=sharing