KINGDOM OF SAUDI ARABIA | JAZAN UNIVERSITY

COLLEGE OF ENGINEERING & COMPUTER SCIENCE

Graduation project title

Space Exploration by using AI

Project Supervisor

Ms. Shabana Parveen

Department of Computer Science

STUDENT	ID
Shahad Abdulrahman	202006701
Marwa Ali	202007410

APPROVAL FOR BINDING OF GRADUATION PROJECT REPORT

GRADUATION PROJECT (495 COMP- 3)

GROUP (CS) : 2324

SESSION 2023-2024 / Third Semester

To be filled in by student

Student name(s)	:	Shahad Abdulrahman	Marwa Ali
University ID(s).	:	202006701	202007410
Project title	:	Space Exploration by using AI	

To be filled in by supervisor*

This is to certify that the project submitted and presented by the above mentioned student(s) is

☐ **COMPLETE & ACCEPTED** ☐ **INCOMPLETE** ☐ **NOT ACCEPTED**

and the draft of graduation project report has been corrected from all content flaws, typing errors and language mistakes.

Supervisor name	:	Ms.Shabana Parveen
Signature	:	
Date	:	

***NOTE**: The student may proceed with Ring binding ONLY after this section has been completed. The student shall submit three **copies** of Ring bound draft report for Graduation Project to the panel of examiners on the day of final presentation.

For Departmental Office use only

Date received	:	
Signatures availability	:	

Acknowledgement

The submission of this dissertation work provided us with a great opportunity to look back and thank to all those who have been directly or indirectly instrumental in successful completion of the project.

This all became possible by the Grace of Almighty “ALLAH” who gave us the strength and ideas for implementing the project. We are indebted to a number of people who have provided us with priceless guidance and support.

We are grateful to the Dean, Vice Dean and Head of the Department , College of Engineering and Computer Science for their kind support and guidance.

In this section of our graduation project we would like to express our great appreciation to the faculty advisor Dr. Shabana Parveen for her willingness to help us in each step of our graduation project and by giving her time so generously has been very much appreciated.

We also want to express our thanks to the faculty members and teachers at Jazan university for being very helpful and working alongside with us to help us achieve our dreams by graduating and starting our successful career paths.

We also would like to thank the college of Engineering and computer for the opportunity they gave to us to successfully build a bright future and for having faith and confidence in our capability to let us undertake this graduation project.

Abstract :

Space exploration generates vast amounts of data in the form of high-resolution images of celestial bodies. Manual analysis and classification of these images are time-consuming and challenging tasks. This project proposes the use of artificial intelligence (AI) models to automate image classification and provide valuable information about celestial bodies.

The project aims to develop an AI model capable of accurately classifying celestial body images and enabling efficient analysis of astronomical data. By leveraging AI techniques such as deep learning and image recognition algorithms, the model will streamline the analysis of astronomical images and provide insights into the nature and characteristics of celestial bodies more efficiently.

The potential applications of this project are vast. Researchers, enthusiasts, and educators will benefit from the automated image classification, as it will significantly reduce the time and effort required for manual analysis. The AI model will enable the analysis of larger volumes of data in a shorter time, facilitating more comprehensive observations and discoveries.

Moreover, the AI model can enhance the accuracy and consistency of image classification. It will be trained on a diverse dataset, allowing it to learn to recognize subtle patterns and features that may be difficult for human observers to identify. This will result in more precise categorization of celestial bodies and provide valuable insights into their composition, structure, and behavior.

The automation of image classification through AI models will also foster broader engagement in the field of space exploration. Accessible and interactive information about celestial bodies can be provided to the general public, enabling a deeper understanding and appreciation of the wonders of the universe.

In conclusion, the proposed AI model for space exploration has the potential to revolutionize the field by automating image classification and analysis. It will contribute to a deeper understanding of celestial bodies, facilitate scientific research, and inspire a wider audience to engage with the wonders of the universe.

Table of Contents

1.INTRODUCTION........8

1.1 Introduction........8

1.2Problem Background........9

1.3Problem Definition / Problem Statement........10

1.4Project Goals and Objectives........11

1.5 Project Scope........12

1.6Work Breakdown Structure & Gantt Chart........13

2. SYSTEM ANALYSIS........14

2.1 Development Methodology........14

2.2 User and System Requirements........15

2.2.1 Functional Requirements........15

2.2.2 Non-Functional Requirements........15

2.3 System Analysis Models........17

2.3.1 Use Case Diagram........17

2.3.2 Sequential Diagram........18

2.3.3 Activity Diagram........19

3.SYSTEM DESIGN........20

3.1 System Architecture........20

3.2 Data Flow Diagram........20

3.3 Class Diagram........23

4. SYSTEM IMPLEMENTATION........24

4.1 Required Hardware & Software........24

4.2 System Implementation........24

4.3Sample Code........25

5. SYSTEM TESTING........47

5.1 Unit Testing........47

5.2 Integration Testing 47

5.3 Testing reports……………………………………………………..48

6. SYSTEM DEMONSTRATION 48

6.1System Screens Snapshots 48

7. CONCLUSION…………………………………………………...58

7.1 Summary……………………………………………………...58

7.2 Limitations and Future Work 59

References:…………………………………………………………..60

Table of Figuer

FIGURE1 WBS -- 13
FIGURE2 GANTT CHART -- 14
FIGURE3 USE CASE DIAGRAM --- 17
FIGURE4 SEQUENTIAL DIAGRAM -- 18
FIGURE5 ACTIVITY DIAGRAM--- 19
FIGURE6 SYSTEM ARCHITECTURE -- 20
FIGURE7 DATA FLOW DIAGRAM LEVEL 0 -- 20
FIGURE8 DATA FLOW DIAGRAM LEVEL 1 -- 21
FIGURE9 DATA FLOW DIAGRAM LEVEL 2 -- 22
FIGURE10 CLASS DIAGRAM -- 23
FIGURE12 LINK -- 48
FIGURE13 WEBSIT-- 49
FIGURE14 RESULT OF LINK -- 50
FIGURE15 SEARCH BY FILE -- 50
FIGURE16 RESULT FROM FILE --- 51
FIGURE17 ANOTHER OUTPUT-- 51
FIGURE18 ANOTHER OUTPUT-- 52
FIGURE19 OUTPUT-- 52
FIGURE20 OUTPUT-- 53
FIGURE21 OUTPUT-- 53
FIGURE22 OUTPUT-- 54
FIGURE23 OUTPUT-- 54
FIGURE24 OUTPUT-- 55
FIGURE25 OUTPUT-- 55
FIGURE26 OUTPUT-- 56
FIGURE27 OUTPUT-- 56
FIGURE28 DATA SET -- 57

1. INTRODUCTION

1.1 Introduction

Space exploration has always been a captivating endeavor, and advancements in technology have allowed us to capture high-resolution images of celestial bodies. However, analyzing and classifying these images manually can be a time-consuming and challenging task due to the vast amount of data and the complex nature of celestial bodies. In this research project, we propose the use of artificial intelligence (AI) models to automate the image classification process and provide valuable information about celestial bodies.

The purpose of this project is to develop an AI model that can accurately classify images of celestial bodies and provide relevant data and descriptions for each image. By leveraging AI techniques, we aim to streamline the analysis of astronomical images and enable researchers and enthusiasts to gain insights into the nature and characteristics of celestial bodies more efficiently.

Space exploration gives rise to humongous amounts of data that cannot be analyzed through human intelligence. Through analyzing and deriving the meaning of the data, AI can change the trajectory of space exploration

Classifying celestial bodies involves identifying and categorizing various objects such as stars, galaxies, nebulae, and planets. Traditionally, these classifications have been performed manually by experts in the field, which is a labor-intensive and subjective process. By utilizing AI models, we can automate this process and potentially improve the accuracy and consistency of classifications.

The research question that guides this project is: "Can an AI model accurately classify images of celestial bodies and provide informative descriptions?" This question serves as the foundation for exploring AI techniques such as deep learning and image recognition algorithms to develop a model capable of analyzing and classifying celestial body images.

To achieve this goal, the project will employ a large dataset of labeled celestial body images to train the AI model. The model will learn to recognize patterns, features, and characteristics specific to different celestial bodies. Once trained, the model can be used to classify new images and provide relevant information about the identified celestial bodies, such as their names, types, distances, and other pertinent data.

The potential applications of this project are vast. The developed AI model can be integrated into websites or astronomy-related platforms, allowing users to upload celestial body images and obtain instant classifications and information. It can assist astronomers in their research by automating the initial analysis of collected images, potentially leading to new discoveries or insights. Additionally, it can serve as an educational tool, providing accessible and interactive information about celestial bodies to students, educators, and the general public.

1.2 Problem Background

1. Vast Amount of Data: With advancements in telescopes and space probes, we have access to an ever-increasing volume of high-resolution images of celestial bodies. Analyzing this massive amount of data manually is time-consuming and impractical.

2. Subjectivity and Human Error: Human experts' subjective interpretation and potential errors in classifying celestial body images can lead to inconsistencies and inaccuracies in the analysis.

3. Efficiency and Scalability: The manual classification process is not scalable, especially when dealing with large datasets. Automating this process can significantly enhance efficiency and enable researchers to process and analyze data at a much faster rate.

4. Complex and Diverse Celestial Bodies: Celestial bodies exhibit a wide range of characteristics and features, making their accurate classification and analysis a challenging task. AI-based image classification can help identify subtle patterns and features that might be difficult for human observers to detect.

5. Challenges in AI Technology: Despite advancements in artificial intelligence and machine learning, there are several challenges in automated image classification. AI algorithms may struggle to accurately identify complex patterns or distinguish subtle differences in astronomical images. Overcoming these challenges and improving the

accuracy and reliability of AI-based classification systems remains an ongoing problem.

6. Potential Limitations in Scientific Exploration: While AI-based image classification offers promising opportunities for scientific exploration, there are potential limitations to consider. Solely relying on automated analysis may overlook unique or unexpected phenomena that human experts could identify. Additionally, interpreting AI-generated results requires careful validation and verification to ensure their scientific validity. Balancing the benefits and limitations of AI technology in scientific exploration is an ongoing problem that researchers need to address.

1.3 Problem Definition / Problem Statement

Proposed Software Solution:

1. AI-Based Image Classification: The software solution aims to develop an AI model capable of classifying celestial body images accurately. This involves training the AI model using a diverse dataset of labeled images to recognize patterns and features specific to different celestial bodies.

2. Automated Data Extraction: The software will extract relevant data about each classified celestial body, such as its name, type, distance from Earth, and other pertinent information. This automated data extraction eliminates the need for manual data collection and provides valuable information for further analysis.

3. Integration with Websites and Platforms: The software solution can be integrated into websites or astronomy-related platforms, allowing users to upload celestial body images and obtain instant classifications and information. This integration enhances accessibility and enables enthusiasts, researchers, and educators to benefit from the automated analysis.

4. Real-Time Classification: The software solution provides real-time image classification, allowing users to receive instant results and classifications as soon as they upload the celestial body images. This ensures quick and efficient analysis without delays.

5. User-Friendly Interface: The software solution features a user-friendly interface that is easy to navigate and interact with. Users can effortlessly upload images, view classifications, and access extracted data in a clear and intuitive manner, making it accessible to users with varying levels of technical expertise.

6. Performance Optimization: The software solution includes performance optimization techniques to ensure efficient and fast image classification and data extraction. By leveraging algorithms and parallel processing, the solution can handle large datasets and perform computations quickly, providing users with timely results and reducing processing time. This optimization enhances the overall user experience and enables the software to handle high volumes of image uploads with ease.

1.4 Project Goals and Objectives

The primary goal of the "Space Exploration using AI Model" project is to develop an AI-based software solution that can accurately classify images of celestial bodies and provide relevant information about each image. The overarching objectives of the project are as follows:

Goals:

Develop an AI model for image classification: The project aims to train an AI model using a diverse dataset of labeled celestial body images. The model will be trained to recognize and classify different types of celestial bodies, including stars, galaxies, nebulae, and planets.
Objectives:

Enable automated data extraction: The project will implement algorithms and techniques to extract relevant data about each classified celestial body. This includes extracting information such as the celestial body's name, type, distance from Earth, and other pertinent information.
Enhance accuracy and efficiency: The project aims to improve the accuracy and efficiency of the image classification process by leveraging AI techniques. By

automating the classification process, the project aims to reduce reliance on manual analysis, which can be time-consuming and prone to human errors.
Foster scientific research and discovery: By developing an AI-based software solution for space exploration, the project aims to contribute to scientific research and discovery in the field of astronomy. The accurate classification of celestial body images and extraction of relevant data will provide researchers with valuable insights into the universe, enabling them to study and understand celestial objects more effectively. This can lead to new discoveries, theories, and advancements in our understanding of the cosmos.
In summary, the "Space Exploration using AI Model" project aims to develop an AI-based software solution that can accurately classify celestial body images and extract relevant information about them. This will enhance the efficiency and accuracy of space exploration and reduce the need for manual analysis. Additionally, the project aims to foster scientific research and contribute to new discoveries in the field of astronomy.

1.5 Project Scope

the project scope can be defined as follows:

1. Astronomical Research: The classification results contribute to understanding the structure, formation, and evolution of celestial bodies, aiding ongoing astronomical research.

2. Astronomical Image Analysis: The project automates the identification and characterization of celestial bodies in astronomical images, saving time and effort in manual analysis.

3. Education and Outreach: The project serves as an educational tool, introducing students and the public to celestial bodies classification, fostering interest in astronomy.

The utilization of the project depends on available datasets, collaboration opportunities, and the specific needs of astronomical research projects or organizations.

1.6 Work Breakdown Structure & Gantt Chart

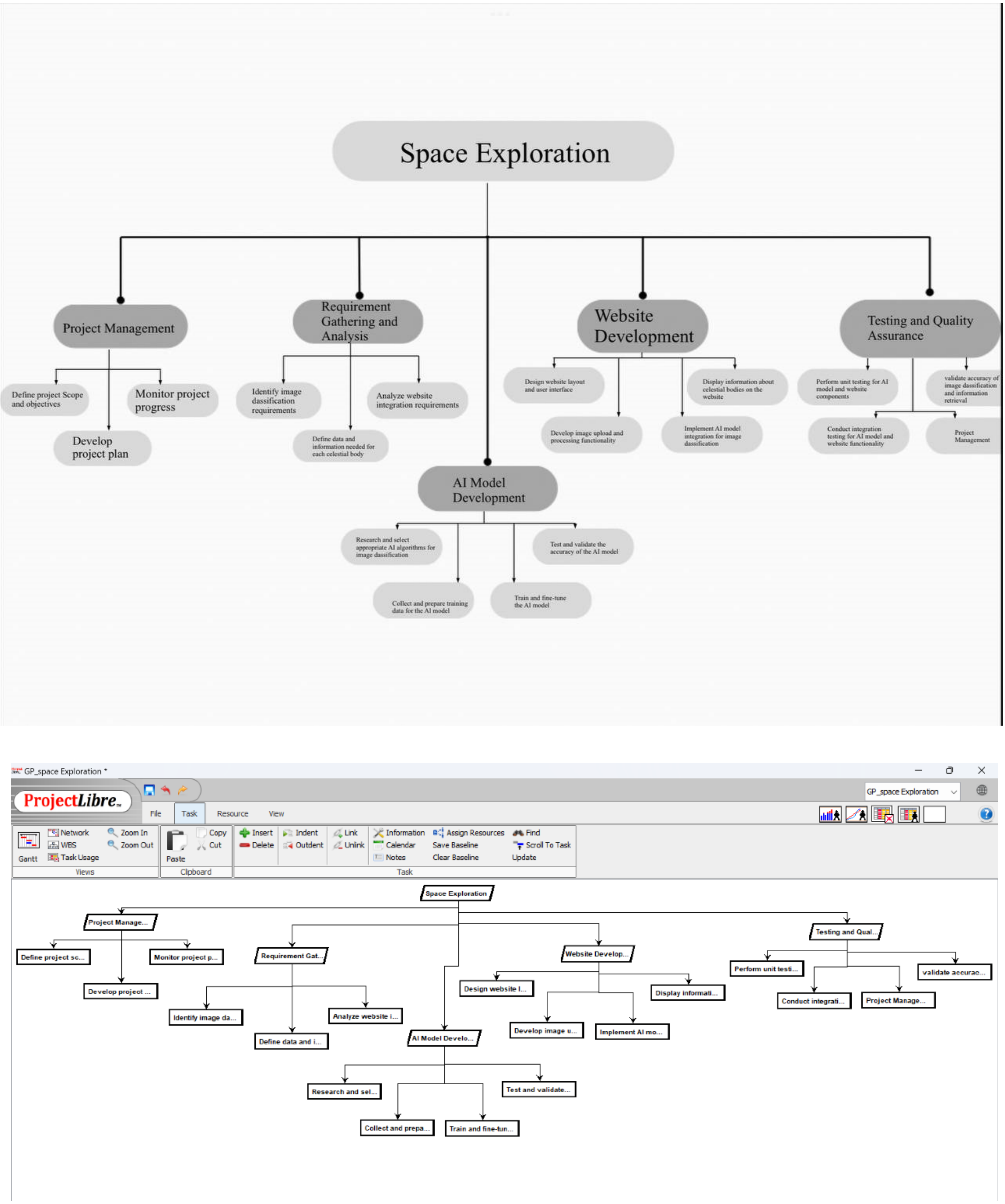
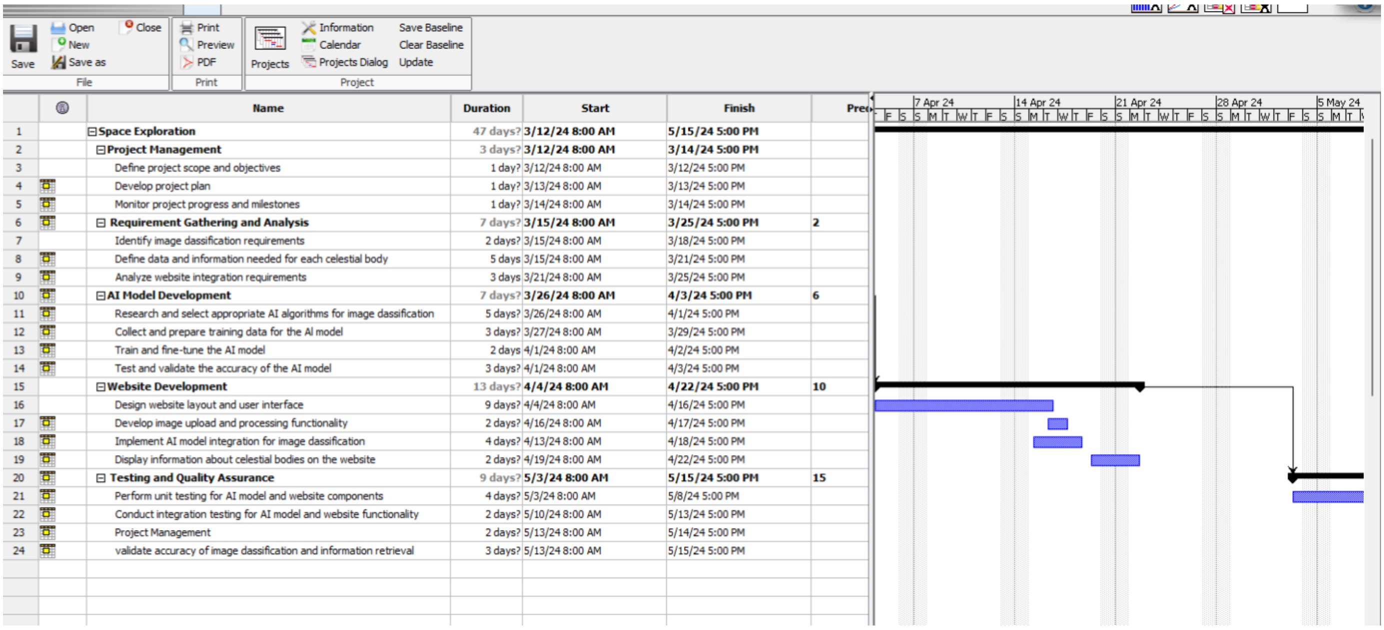


Figure 1 WBS



	Name	Duration	Start	Finish	Pred
1	**Space Exploration**	47 days?	**3/12/24 8:00 AM**	**5/15/24 5:00 PM**	
2	**Project Management**	3 days?	**3/12/24 8:00 AM**	**3/14/24 5:00 PM**	
3	Define project scope and objectives	1 day?	3/12/24 8:00 AM	3/12/24 5:00 PM	
4	Develop project plan	1 day?	3/13/24 8:00 AM	3/13/24 5:00 PM	
5	Monitor project progress and milestones	1 day?	3/14/24 8:00 AM	3/14/24 5:00 PM	
6	**Requirement Gathering and Analysis**	7 days?	**3/15/24 8:00 AM**	**3/25/24 5:00 PM**	2
7	Identify image classification requirements	2 days?	3/15/24 8:00 AM	3/18/24 5:00 PM	
8	Define data and information needed for each celestial body	5 days	3/15/24 8:00 AM	3/21/24 5:00 PM	
9	Analyze website integration requirements	3 days	3/21/24 8:00 AM	3/25/24 5:00 PM	
10	**AI Model Development**	7 days?	**3/26/24 8:00 AM**	**4/3/24 5:00 PM**	6
11	Research and select appropriate AI algorithms for image classification	5 days?	3/26/24 8:00 AM	4/1/24 5:00 PM	
12	Collect and prepare training data for the AI model	3 days?	3/27/24 8:00 AM	3/29/24 5:00 PM	
13	Train and fine-tune the AI model	2 days	4/1/24 8:00 AM	4/2/24 5:00 PM	
14	Test and validate the accuracy of the AI model	3 days?	4/1/24 8:00 AM	4/3/24 5:00 PM	
15	**Website Development**	13 days?	**4/4/24 8:00 AM**	**4/22/24 5:00 PM**	10
16	Design website layout and user interface	9 days?	4/4/24 8:00 AM	4/16/24 5:00 PM	
17	Develop image upload and processing functionality	2 days?	4/16/24 8:00 AM	4/17/24 5:00 PM	
18	Implement AI model integration for image classification	4 days?	4/13/24 8:00 AM	4/18/24 5:00 PM	
19	Display information about celestial bodies on the website	2 days?	4/19/24 8:00 AM	4/22/24 5:00 PM	
20	**Testing and Quality Assurance**	9 days?	**5/3/24 8:00 AM**	**5/15/24 5:00 PM**	15
21	Perform unit testing for AI model and website components	4 days?	5/3/24 8:00 AM	5/8/24 5:00 PM	
22	Conduct integration testing for AI model and website functionality	2 days?	5/10/24 8:00 AM	5/13/24 5:00 PM	
23	Project Management	2 days?	5/13/24 8:00 AM	5/14/24 5:00 PM	
24	validate accuracy of image classification and information retrieval	3 days?	5/13/24 8:00 AM	5/15/24 5:00 PM	

Figure2 Gantt Chart

2. SYSTEM ANALYSIS

2.1 Development Methodology

The software development life cycle (SDLC) is a commonly used methodology for the development of projects.

1. Requirements Gathering: In this phase, the project requirements are gathered and defined. This involves understanding the goals and objectives of the project, as well as the specific functionalities and features required for the AI model and website.

2. Analysis and Design: In this phase, the system architecture and design are developed. This includes designing the classification algorithms for the AI model and determining the structure and layout of the website. The data flow, database design, and user interface design may also be considered during this phase.

3. Development: Once the design is finalized, the development phase begins. This involves coding the AI model algorithms for image classification and developing the website functionality. The development process may involve programming languages like Python for the AI model and web technologies like HTML, CSS, and JavaScript for the website.

4. Testing: Testing is a critical phase to ensure the functionality and quality of the developed system. Different types of testing, such as unit testing, integration testing, and system testing, are performed to identify and fix any issues or bugs.

5. Deployment: After successful testing, the project is deployed to a production environment. This involves setting up the AI model on a server or cloud platform and deploying the website to a web server. Configuration and optimization are performed to ensure the system runs smoothly in the live environment.

2.2 User and System Requirements

2.2.1 Functional Requirements

Functional requirements define specific behaviors or functions of a system.

1. **Image Upload**:

Users can upload images directly from their device.

Users can submit images via a URL link.

2. **Image Processing**:

The system should validate that the file uploaded is in an acceptable image format (e.g., JPG, PNG).

The system should be able to retrieve images from specified URLs.

3. **Classification**:

The uploaded or linked images are processed using the trained AI model.

The model classifies images as either a planet, a galaxy, or other celestial bodies.

4. **Results Display**:

Display classification results clearly on the website.

Provide details about the classified objects, such as type, estimated size, distance, and other relevant astronomical data.

2.2.2 Non-Functional Requirements

Non-functional requirements describe the system's quality attributes or characteristics.

1. **Performance**:

The website should load within 5 seconds.

Image processing and classification should be completed within 10 seconds for optimal user experience.

2. **Usability**:

The website should be user-friendly with intuitive navigation and accessible features.

The website should be responsive, providing an optimal experience across various devices and screen sizes.

3. **Reliability**:

The system should be available 99.9% of the time.

The classification results should be accurate and consistent.

4. **Scalability**:

The system should handle a large number of users and simultaneous requests without degradation in performance.

2.3 System Analysis Models

2.3.1 Use Case Diagram

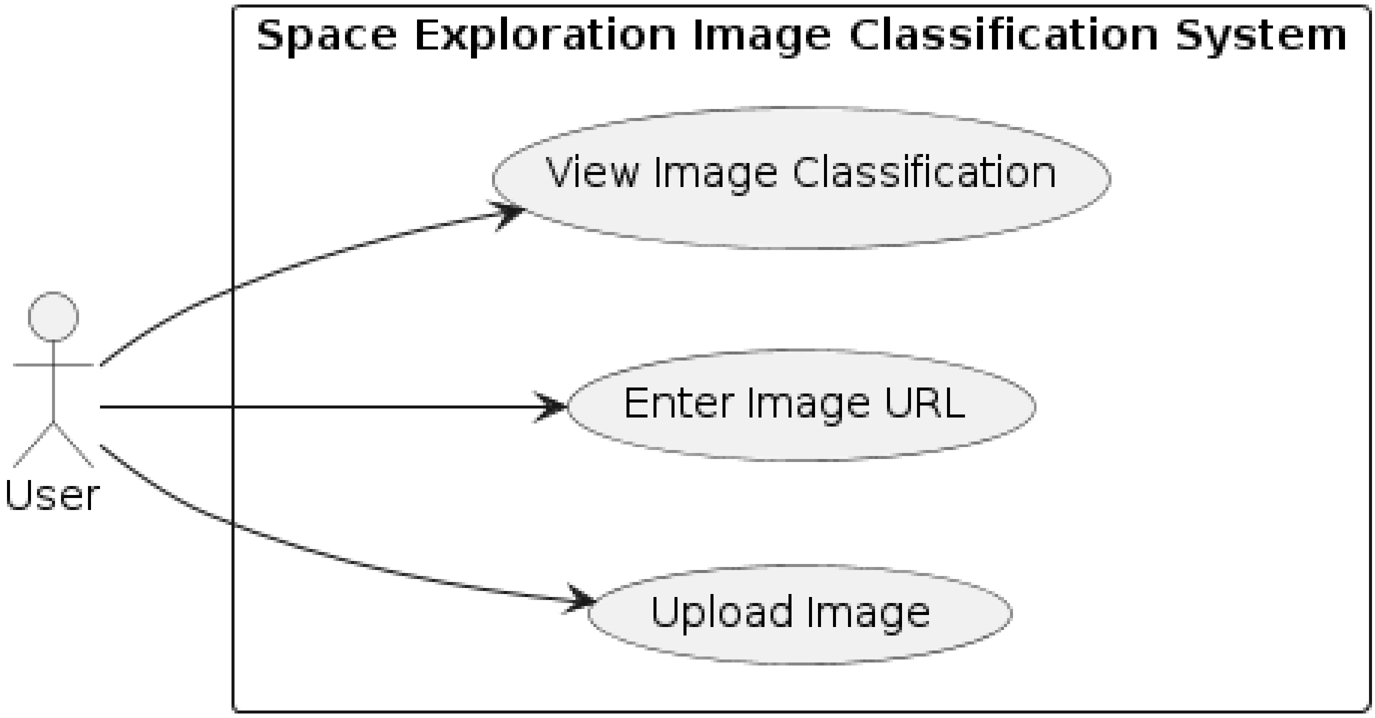


Figure3 Use Case Diagram

2.3.2 Sequential Diagram

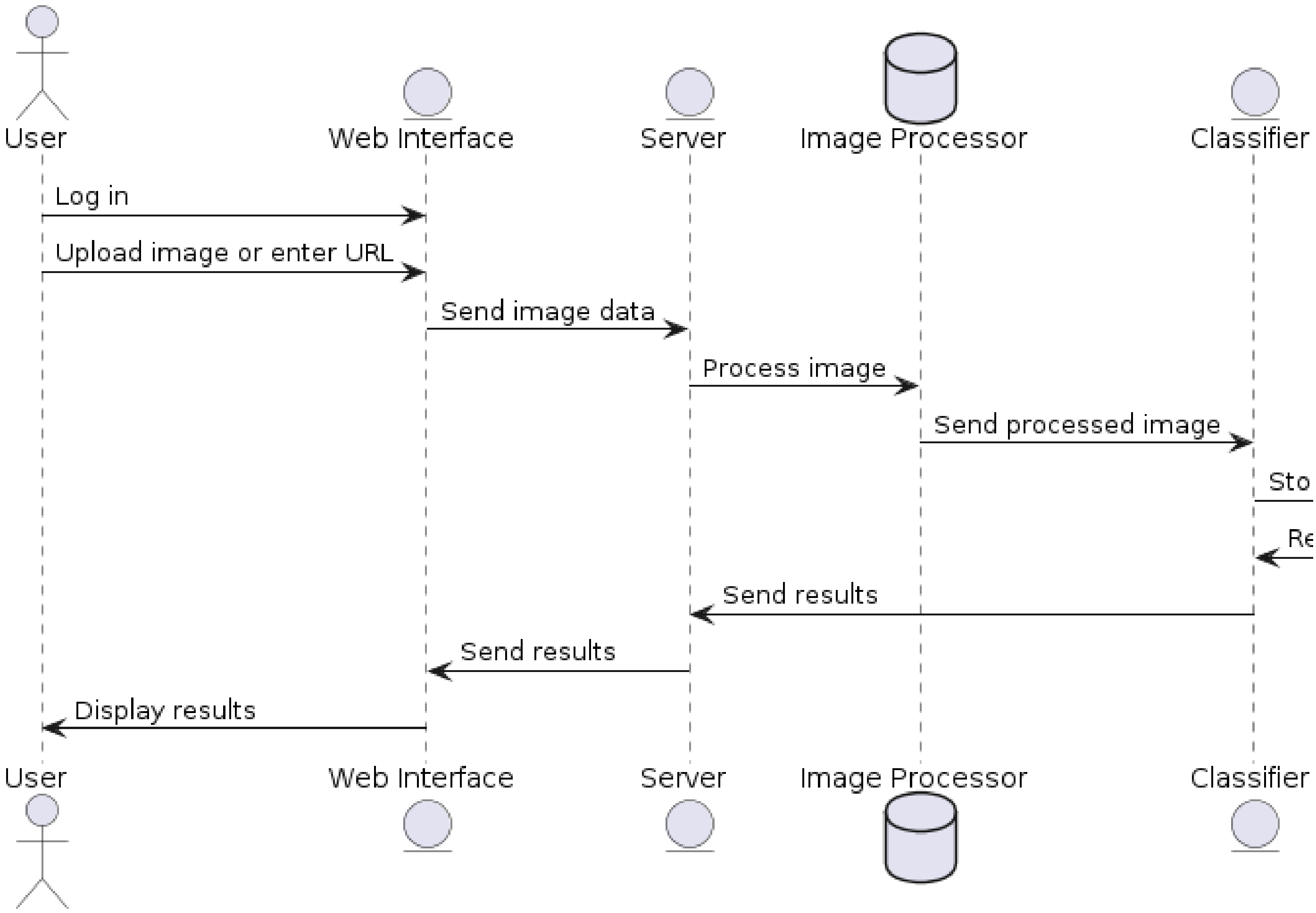


Figure4 Sequential Diagram

2.3.3 Activity Diagram

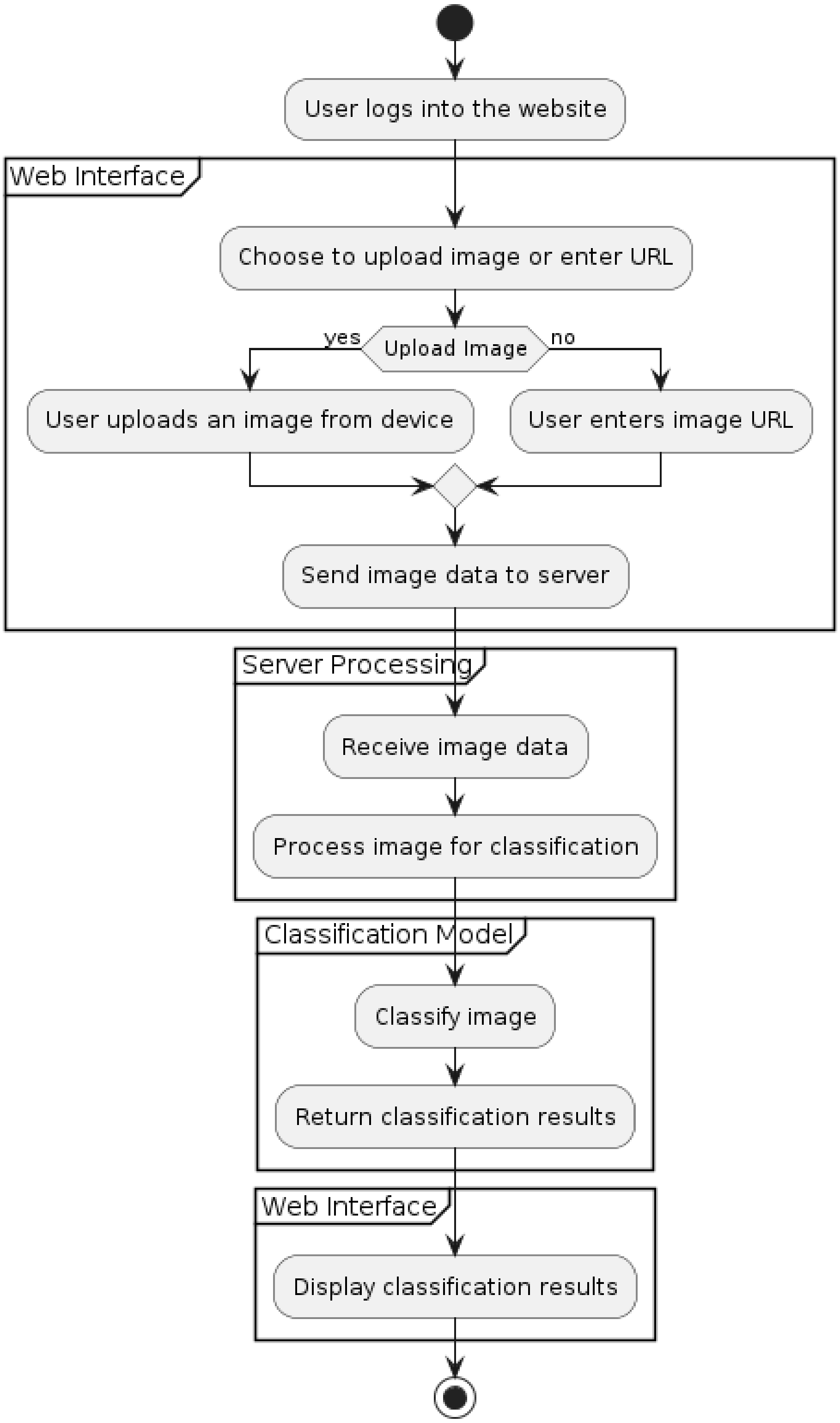


Figure5 Activity Diagram

3.SYSTEM DESIGN

3.1 System Architecture

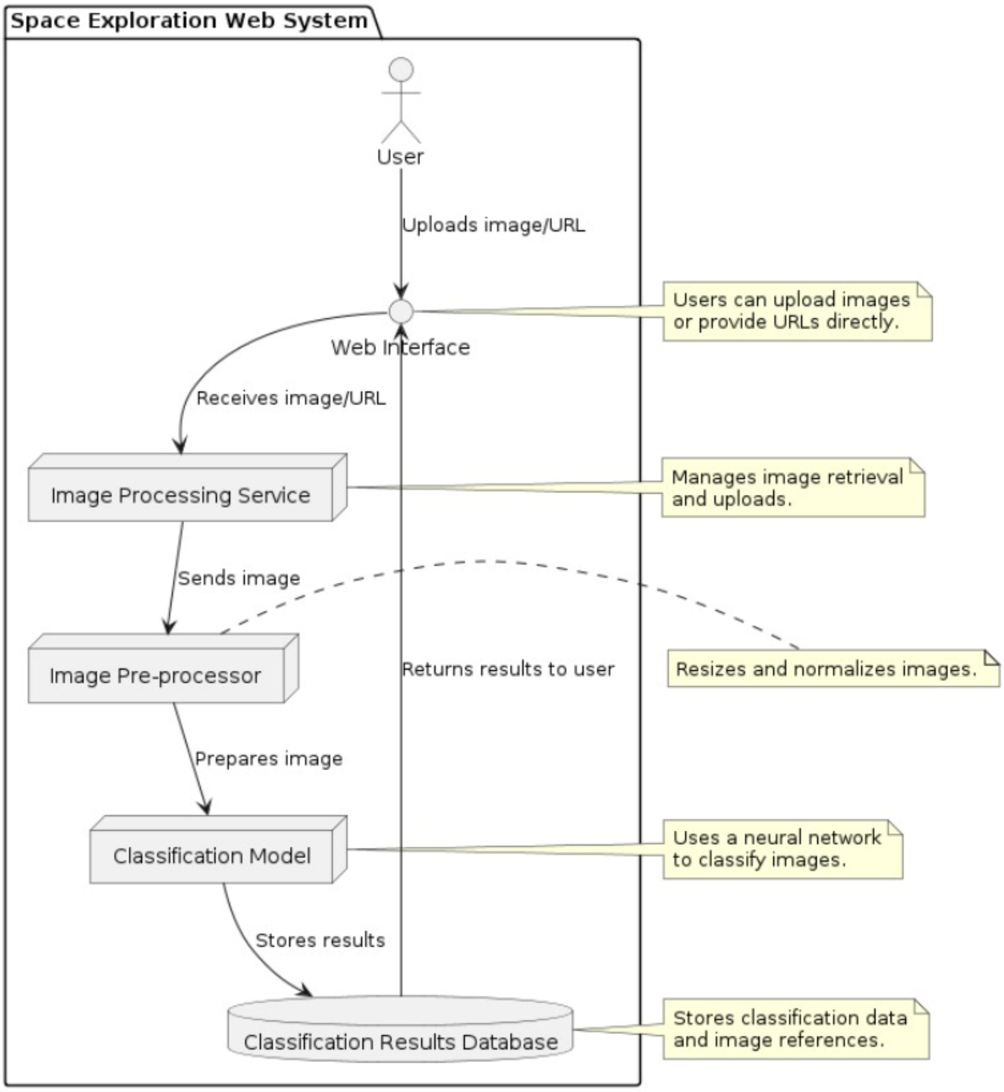


Figure6 System Architecture

- **DFD Level 0**

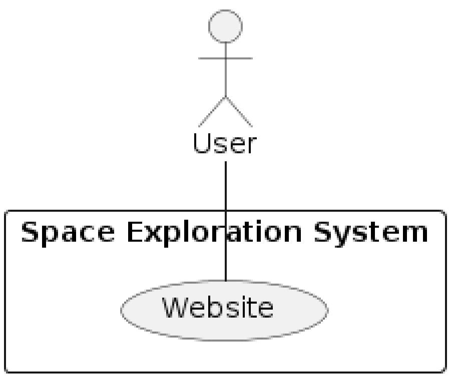


Figure7 Data Flow Diagram Level 0

- **DFD Level 1**

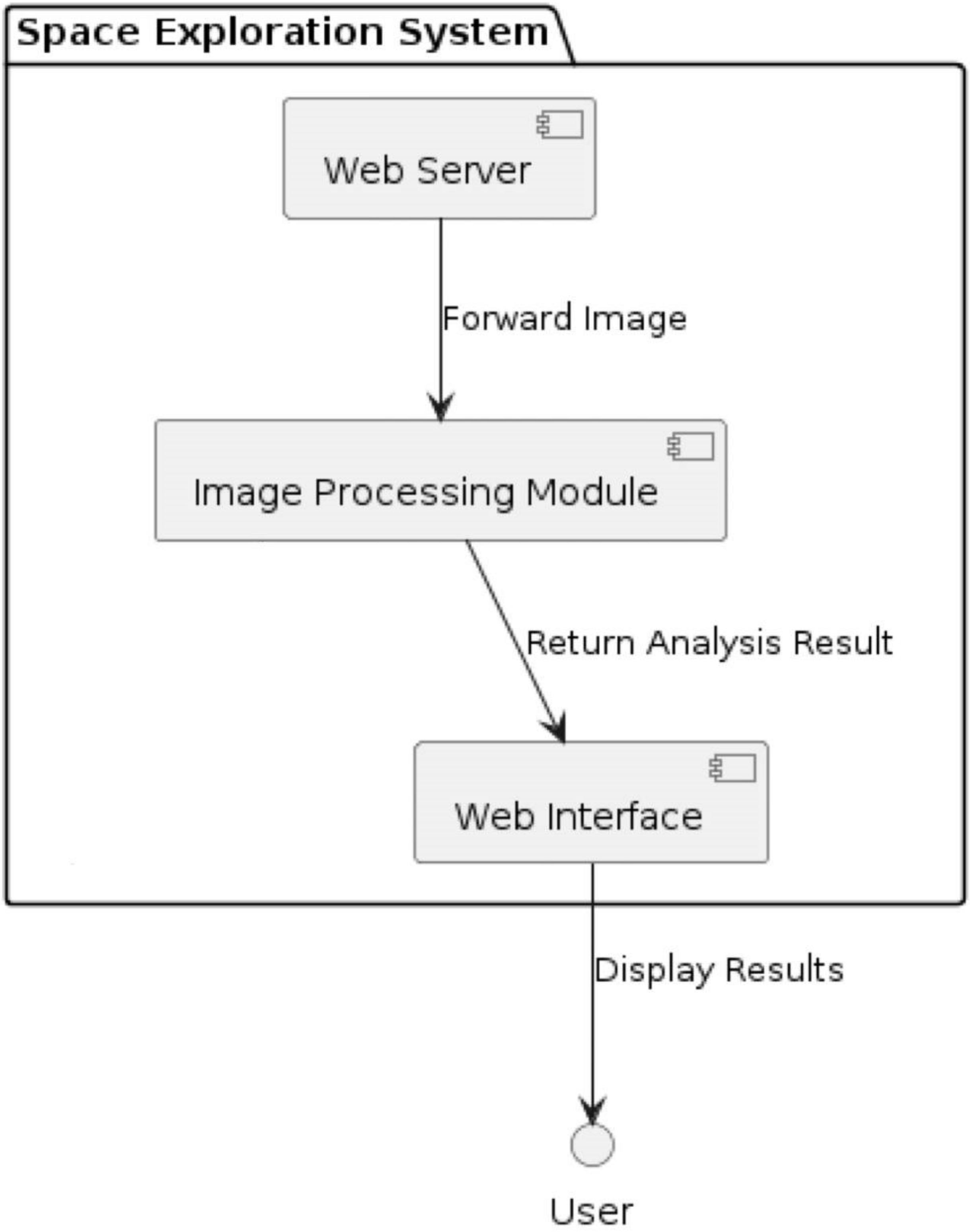


Figure8 Data Flow Diagram Level 1

- **DFD Level 2**

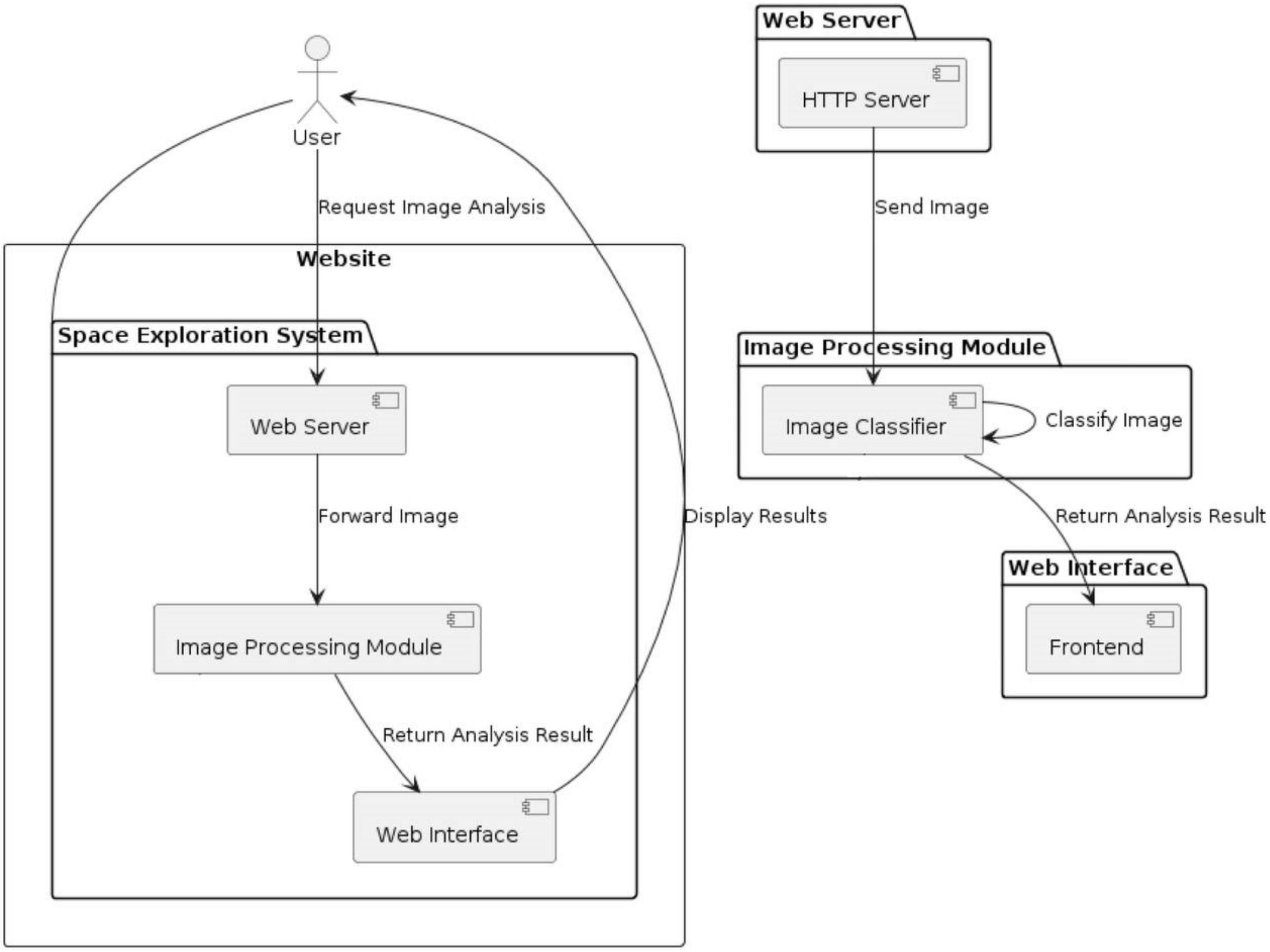


Figure9 Data Flow Diagram Level 2

3.3 Class Diagram

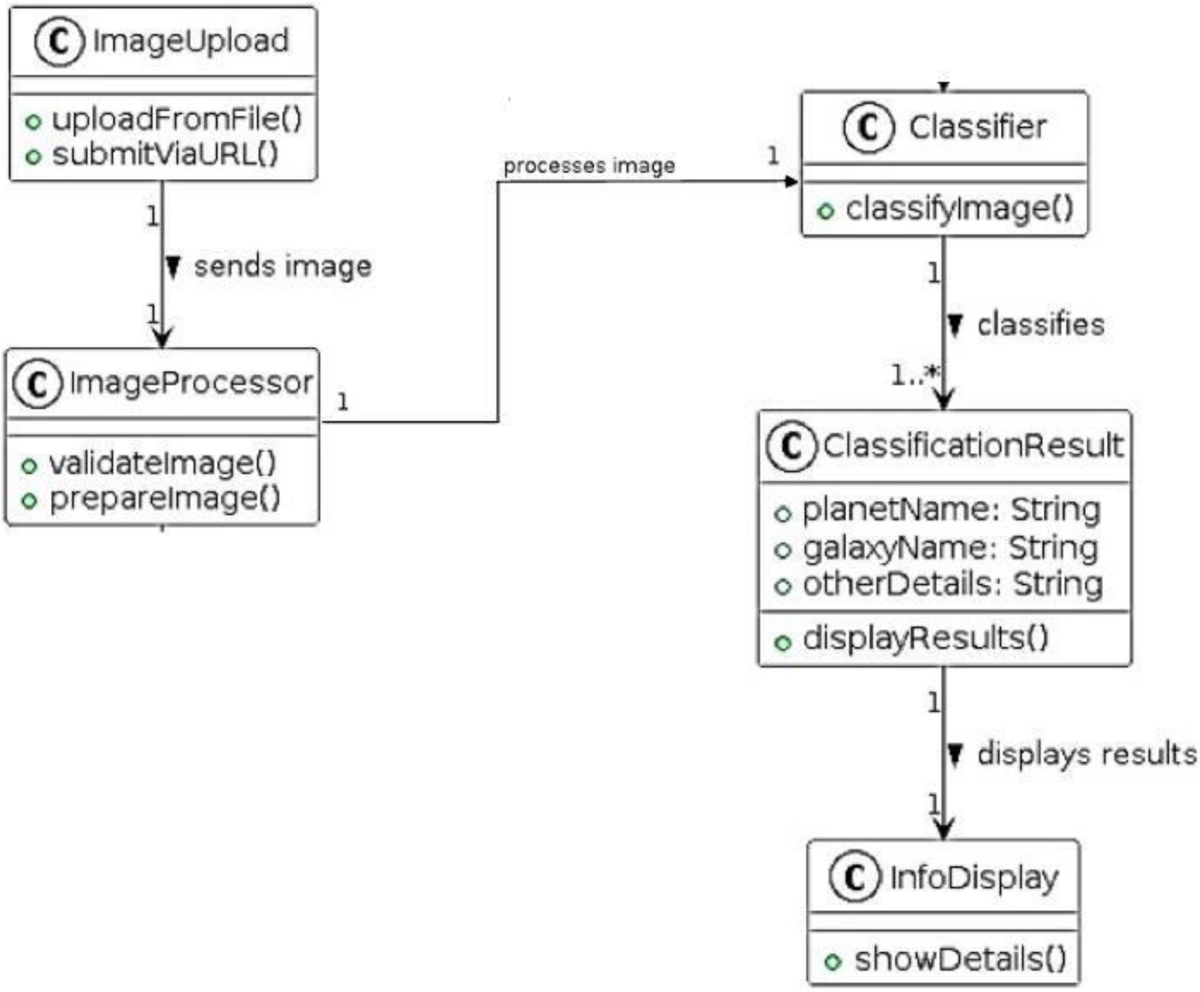


Figure10 Class Diagram

4. SYSTEM IMPLEMENTATION

4.1 Required Hardware & Software

- **Hardware**

S.No	Item	Description
1.	Processor	Core 15
2.	RAM	16.0 GB
3.	Processor Speed	3.0 GB
4.	Hard Disk	80 GB
5.	System type	64-bit operating system

- **Software**

S.No	Item	Description
1.	Google Colab	training model environment
2.	Visual studio	Web sit environment
3.	HTML	For coding the websit
4.	Python	For Connect model with html
5.	Windows	Windows11

4.2 System Implementation

1. User Interface Development:

   We developed an attractive and user-friendly interface for the website. It included options for uploading an image or entering a Google image link.

2. Image Upload

   When the user uploaded an image or entered a Google image link, we implemented a function to upload the image to our server.

3. Image Analysis using the Model:

   We executed the model we created for classifying astronomical images. We utilized deep learning or artificial intelligence techniques to analyze the image and extract astronomical information from it.

4. Displaying the Results:
   After analyzing the image, we displayed the results to the user on the web page. We used the user interface to present the extracted astronomical information, such as the name of the planet or galaxy in the image.

4.3 Sample Code

Model:

```
import numpy as np

import pandas as pd

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers


data_train_path = "Data/training_data"

data_test_path = "/Data/test_data"

data_val_path = "/Data/Val"


img_width =180

img_height = 180


data_train = tf.keras.utils.image_dataset_from_directory(

    data_train_path,

    shuffle=True,

    image_size=(img_width, img_height),

    batch_size=32,
```

```
    validation_split=False)


data_cat = data_train.class_names
data_cat


data_val=tf.keras.utils.image_dataset_from_directory(
    data_val_path,
    shuffle=False,
    image_size=(img_width,img_height ),
    batch_size=32,
    validation_split=False)


data_test=tf.keras.utils.image_dataset_from_directory(
    data_test_path,
    shuffle=False,
    image_size=(img_width,img_height ),
    batch_size=32,
    validation_split=False)


plt.figure(figsize=(10,10))
for image, labels in data_train.take(1):
  for i in range(9):
    plt.subplot(3,3,i+1)
    plt.imshow(image[i].numpy().astype("uint8"))
```

```
    plt.title(data_cat[labels[i]])
    plt.axis("off")


from tensorflow.keras.models import Sequential

model = Sequential([
    layers.Rescaling(1./255),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32,3, padding='same',activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dropout(0.2),
    layers.Dense(128),
    layers.Dense(len(data_cat))
])

model.compile(optimizer='adam',
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
metrics=['accuracy'])

epochs_size = 25
```

```
history = model.fit(data_train, validation_data=data_val, epochs=epochs_size)


keras.utils.plot_model(model, show_shapes=True, rankdir="LR")
epochs_range = range(epochs_size)
plt.figure(figsize=(8,8))
plt.subplot(1,2,1)


plt.plot(epochs_range,history.history['accuracy'],label = 'Training Accuracy')
plt.plot(epochs_range, history.history['val_accuracy'],label = 'Validation Accuracy')
plt.title('Accuracy')
plt.subplot(1,2,2)


plt.plot(epochs_range,history.history['loss'],label = 'Training Loss')
plt.plot(epochs_range, history.history['val_loss'],label = 'Validation Loss')
plt.title('Loss')


from keras.models import save_model
save_model(model, 'space.h5')
```

app.py:

```
from flask import Flask, request, render_template, redirect, url_for,
send_from_directory
from werkzeug.utils import secure_filename
import os  # For file operations
```

```
import tensorflow as tf
import numpy as np
from PIL import Image
from io import BytesIO
import requests
import pandas as pd

# Configure Flask app
app = Flask(__name__)

# Define allowed extensions for uploaded files
ALLOWED_EXTENSIONS = {'png', 'jpg', 'jpeg'}

# Function to check allowed file extension
def allowed_file(filename):
  return '.' in filename and \
         filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

def getData(label):
  df = pd.read_excel('static/SpaceExploration_data.xlsx')
  data = df[label]
  return data

@app.route('/', methods=['GET', 'POST'])
```

```
def upload_image():
  if request.method == 'GET':
    # Render the upload.html template for initial page load
    return render_template('upload.html')
  else:
    # Handle file upload on POST request
    file = request.files['file']
    image_url = request.form.get('imageLink')

    if image_url:
        # Handle image URL download
        try:
          response = requests.get(image_url, stream=True)
          response.raise_for_status()  # Raise an exception for bad status codes
          image = 'ss'
          # Extract filename from URL
          filename = image_url.split('/')[-1]
          if response.status_code == 200:
                # Open the image and save it to the static folder
                image = Image.open(BytesIO(response.content))
                image_path = os.path.join('static', 'downloaded_image.png')
                image.save(image_path)

          # Continue with the prediction process using 'static/filename'
```

```
labels = ['asteroids',
          'earth',
          'elliptical',
          'jupiter',
          'mars',
          'mercury',
          'moon',
          'neptune',
          'pluto',
          'saturn',
          'spiral',
          'uranus',
          'venus']

image = tf.keras.utils.load_img(os.path.join('static', 'downloaded_image.png'),
target_size=(180, 180))
img_arr = tf.keras.utils.array_to_img(image)
img_bat = tf.expand_dims(img_arr, 0)  # Add batch dimension
#model = tf.keras.models.load_model(os.path.join('static','space.h5'))
model = tf.keras.models.load_model('models/space.h5')


prediction = model.predict(img_bat)
score = tf.nn.softmax(prediction)
prediction_text = labels[np.argmax(score)]
planet_data = getData(prediction_text)
```

```
            if prediction_text:
                return render_template('result.html', filename=filename,
image_url='downloaded_image.png', prediction=prediction_text,
planet_data=planet_data)
            else:
                return render_template('upload.html', filename=filename,
image_url=f'/{filename}', prediction='')
        except requests.exceptions.RequestException as e:
            return render_template('upload.html', message=f'Error downloading image:
{e}')
    elif file and allowed_file(file.filename):

        # Handle file upload
        filename = secure_filename(file.filename)
        file.save(os.path.join('static', filename))
        labels = ['asteroids',
            'earth',
            'elliptical',
            'jupiter',
            'mars',
            'mercury',
            'moon',
            'neptune',
            'pluto',
            'saturn',
```

```
            'spiral',
            'uranus',
            'venus']
        image = tf.keras.utils.load_img(os.path.join('static', filename), target_size=(180,
180))
        img_arr = tf.keras.utils.array_to_img(image)
        img_bat = tf.expand_dims(img_arr, 0)  # Add batch dimension
        # model = tf.keras.models.load_model('models/modelSaved.keras')
        model = tf.keras.models.load_model('models/space.h5')


        prediction = model.predict(img_bat)
        score = tf.nn.softmax(prediction)
        prediction_text = labels[np.argmax(score)]
        planet_data = getData(prediction_text)
        if prediction_text:
          return render_template('result.html', filename=filename,
image_url=f'/{filename}', prediction=prediction_text, planet_data=planet_data)
        else:
          return render_template('upload.html', filename=filename,
image_url=f'/{filename}', prediction='')
    else:
        return render_template('upload.html', message='No file selected or image URL
provided')


@app.route('/result/<filename>')
```

```
def show_result(filename):
  # No logic needed here, redirect to upload page
  return redirect(url_for('upload_image'))  # Redirects to upload page


@app.route('/static/<path:filename>')
def serve_static(filename):
  return send_from_directory('static', filename)


if __name__ == '__main__':
  # Create uploads directory if it doesn't exist
  #  os.makedirs('uploads', exist_ok=True)  # Create uploads directory if missing
  app.run(debug=True)
```

html:

- **None.html**

```
<!DOCTYPE html>
<html lang="en">
<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Space Exploration</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
```

```
</head>
<body>

    <div class="background"></div>
    <div class="container">
        <h1>Space Exploration</h1>
        <div class="search-box">
            <h2>Search by Link</h2>
            <input type="text" id="imageLink" placeholder="Enter image link">
        </div>

        <div class="upload-box">
            <h2>Upload Image</h2>
            <form method="post" enctype="multipart/form-data"> <input type="file"
id="imageUpload" name="file"
                    accept="image/*">
                <button type="submit">Search</button>

 </form>
        </div>
    </div>
    <div id="result">
        <img id="imageDisplay" src="{{ image_url or '' }}" alt="Celestial Image">
        <div id="imageInfo">{{ prediction or '' }}</div>
    </div>
```

```
    <script src="{{ url_for('static', filename='script.js') }}"></script>
</body>
</html>
```

- **Result.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Prediction Result</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">

</head>
<body>
    <div class="background"></div>
    <div style="display: flex;">
        <div class="container">
            <div style="display: flex; flex-direction: column;">
                <div id="imageInfo">{{ prediction }}</div>
                {% if planet_data.empty %}
                <p>No data available for the predicted planet.</p>
```

```
{% else %}
<table>
    <tr>
        <th>
            Average Distance from the sun
        </th>
        <td>
            {{ planet_data[0] }}
        </td>
    </tr>
    <tr>
        <th>
            Diameter
        </th>
        <td>
            {{ planet_data[1] }}
        </td>
    </tr>
    <tr>
        <th>
            Surface Composition
        </th>
        <td>
```

```
        {{ planet_data[2] }}
    </td>
</tr>
<tr>
    <th>
        Rotation Period
    </th>
    <td>
        {{ planet_data[3] }}
    </td>
</tr>
<tr>
    <th>
        Revolution Period
    </th>
    <td>
        {{ planet_data[4] }}
    </td>
</tr>
<tr>
    <th>
        Known No. of Moons
    </th>
    <td>
```

```
                    {{ planet_data[5] }}
                  </td>
                </tr>
              </table>
              <h1 style="margin-top: 50px"></h1>
              <p style="text-align: justify;"> {{ planet_data[6] }}</p>
              {% endif %}
            </div>
        </div>
        <div class="container">
            <img id="imageDisplay" width="400px" src="{{ url_for('static', filename=" +
image_url) }}"
              alt="Celestial Image">
        </div>
    </div>

    <script src="{{ url_for('static', filename='script.js') }}"></script>
</body>
</html>
```

- **Upload.html**
-

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Space Exploration</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
</head>
<body>
    <div class="background"></div>
    <div class="container">
        <h1>Space Exploration</h1>
        <form method="post" enctype="multipart/form-data">
            <div class="search-box">
                <h2>Search by Link</h2>
                <input type="text" id="imageLink" name="imageLink" placeholder="Enter
image link">
            </div>

            <div class="upload-box">
                <h2>Upload Image</h2>
                <input type="file" id="imageUpload" name="file" accept="image/*">
            </div>
            <button type="submit">Search</button>
        </form>
    </div>


    </div>
```

```
    <script src="{{ url_for('static', filename='script.js') }}"></script>
</body>
</html>
```

- **Script.js**

```
console.log("Hello from scripts.js!");
function upload_image() {
    // show loading gif when processing
    console.log("uploading");
    var x = document.getElementsByClassName("loader");
    var i;
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "block";
    }
    var x = document.getElementsByClassName("my-image-form");
    var i;
    for (i = 0; i < x.length; i++) {
        x[i].style.display = "none";
    }
}

function open_browser(){
    // open file browser
    $('#image_file').trigger('click');
```

```
}


function hide_know_more(){
   // when no button is clicked
   $("#know_more")[0].style.display="none";
}


$(document).ready(function(){
   $("input[type=file]").on('change',function(){
       // when file selected
       document.getElementById("image_url").value="";
       document.getElementById("imageName").style.display = "block";
       // document.getElementById("imageName").style.color = "white";
       document.getElementById("imageName").innerText = this.files[0].name;
   });
});
```

- **Styles.css**

```
/* Global styles */
body {
   margin: 0;
   font-family: Arial, sans-serif;
   background-color: #000;
   color: #fff;
```

```css
}

.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
    text-align: center;
}
h2 {
    font-size: 24px;
    margin-bottom: 10px;
}
input[type="text"],
input[type="file"] {
    width: 100%;
    padding: 15px;
    border: none;
    border-radius: 25px;
    font-size: 16px;
    background-color: rgba(255, 255, 255, 0.1);
    color: #fff;
    margin-bottom: 20px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.3) inset;
}
```

```css
button {
    padding: 15px 30px;
    background-color: #0066ff;
    color: #fff;
    border: none;
    border-radius: 25px;
    cursor: pointer;
    font-size: 16px;
    transition: background-color 0.3s ease;
}
button:hover {
    background-color: #0052cc;
}
/* Home page specific styles */
.background {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-image: url('s.jpg');

    /* Add your background image */
    z-index: -1;
```

```
    animation: starsAnimation 20s linear infinite;
}

@keyframes starsAnimation {
    from {
        background-position: 0 0;
    }
    to {
        background-position: 1000px 1000px;
        /* Adjust as needed */
    }
}

.search-box,
.upload-box {
    background-color: rgba(0, 0, 0, 0.7);
    border-radius: 15px;
    padding: 20px;
    margin-bottom: 20px;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.3);
}

/* Add more styles as needed */
#imageInfo {
```

```
    font-size: 90px;
    font-weight: bold;
    text-transform: capitalize;
}

/* td {
    background-color: #fff;
    color: #000;
    border-radius: 50%;
} */

table,
td,
th {
    border: 1px #fff solid;
}

table {
    height: 150px;
    width: 100%;
    border-collapse: collapse;
```

5. SYSTEM TESTING

5.1 Unit Testing

Unit	Testing	Result
Check model	Objective: Verify if the model correctly analyzes images and provides the names of celestial bodies.	The check was successful; the model accurately identifies celestial bodies.
Check connection model	Ensure the connection between the model and the web interface using Flask is functioning properly.	The check was successful; the model successfully connects with the Flask web interface.
Check URL	Confirm if the website properly receives image links copied by the user.	The check was successful; the website accurately receives image links.
Check Upload	Validate if the website correctly receives and processes uploaded image files.	The check was successful; the website accurately receives and processes uploaded images.

5.2 Integration Testing

In space exploration, a model has been created to classify images of celestial objects, and it has been integrated with a Flask web page using HTML. The model has been tested and has successfully achieved excellent performance in classification. Additionally, performance testing has been conducted to ensure that the system responds quickly and efficiently to user requests on the website.

5.3 Testing reports

The testing process, comprising unit and integration testing, confirms the robustness and reliability of the system. The model accurately analyzes celestial objects, connects seamlessly with the web interface, and efficiently handles user requests. Performance testing further assures users of a responsive and efficient experience on the website.

6. SYSTEM DEMONSTRATION

6.1System Screens Snapshots

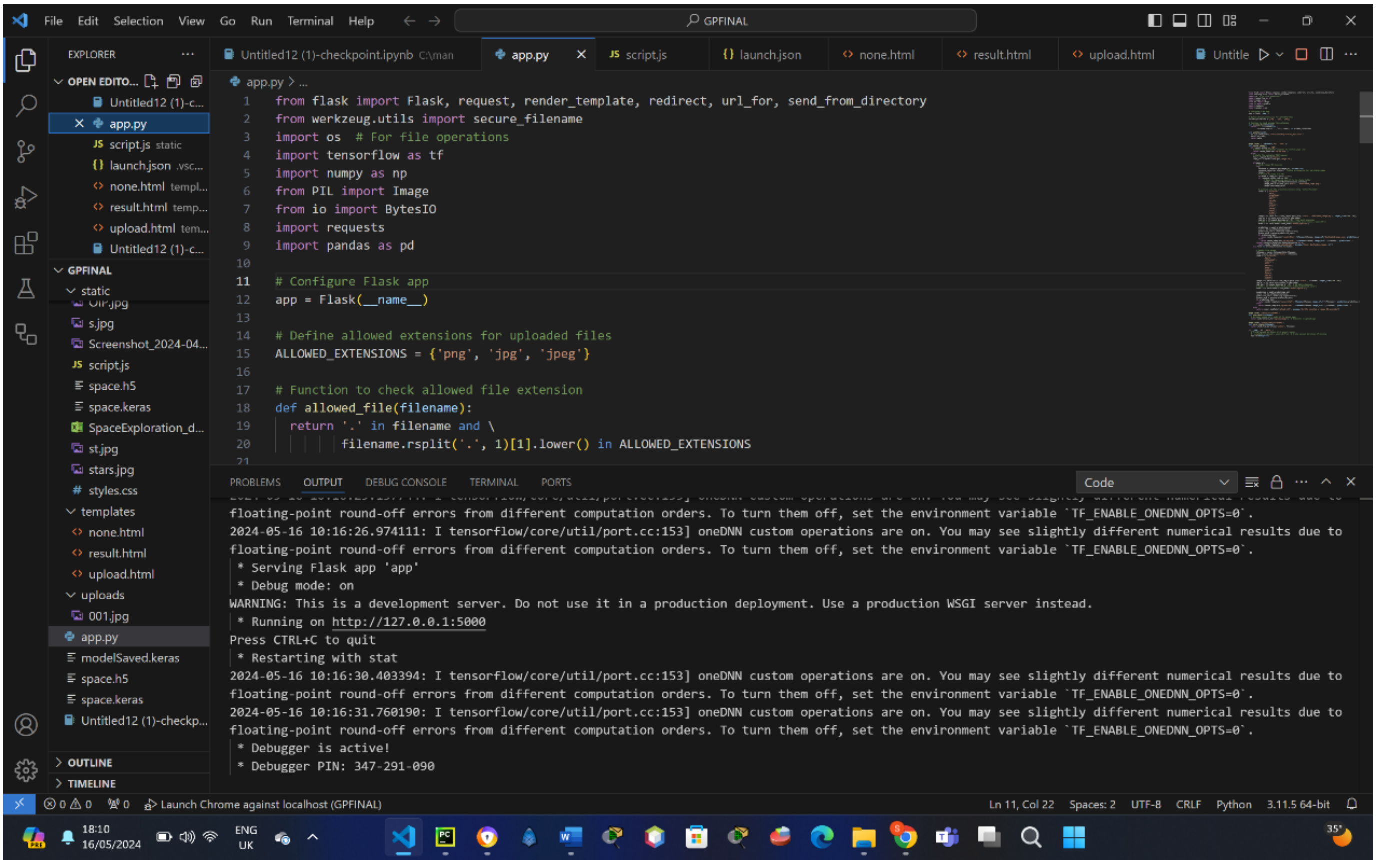


Figure11 Link

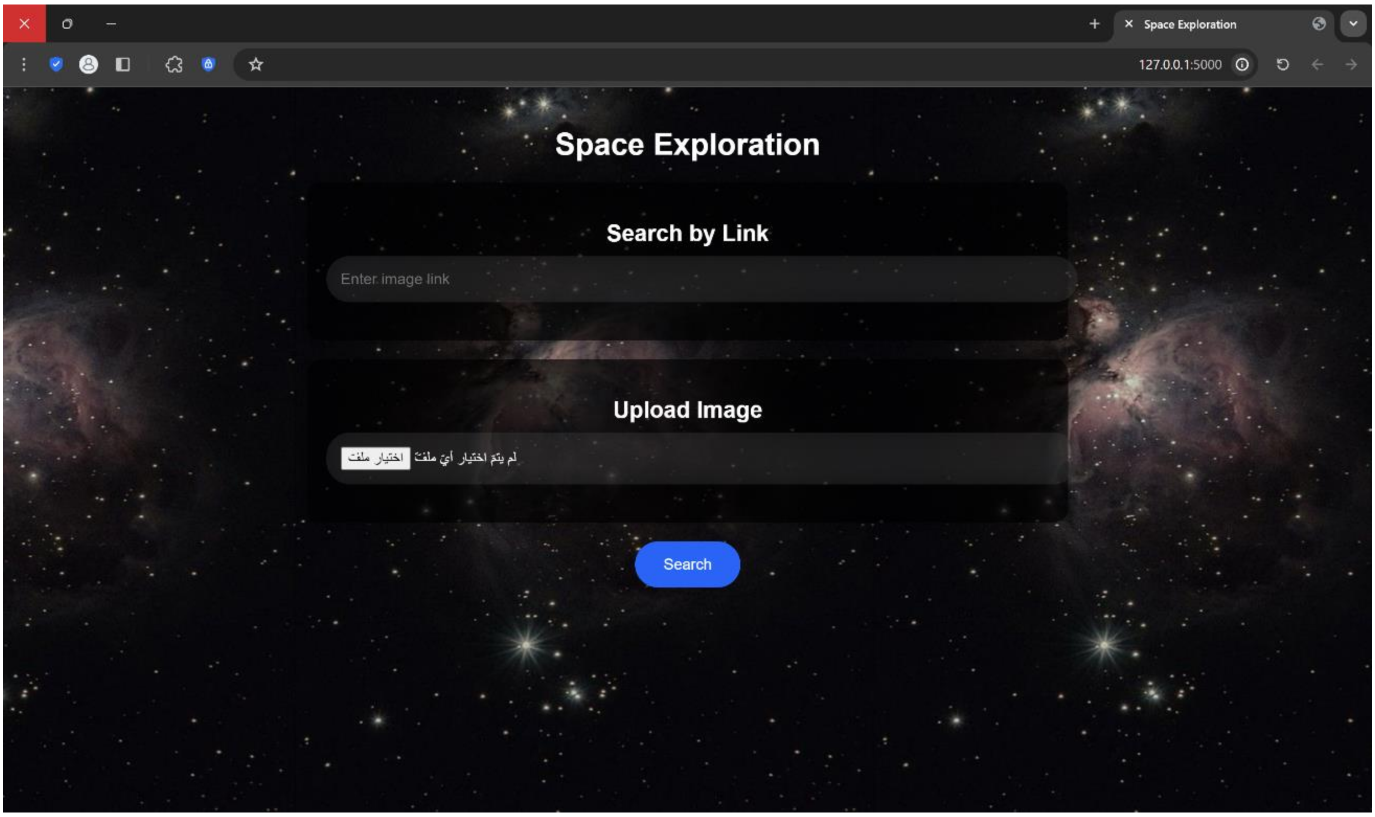


Figure 12 websit

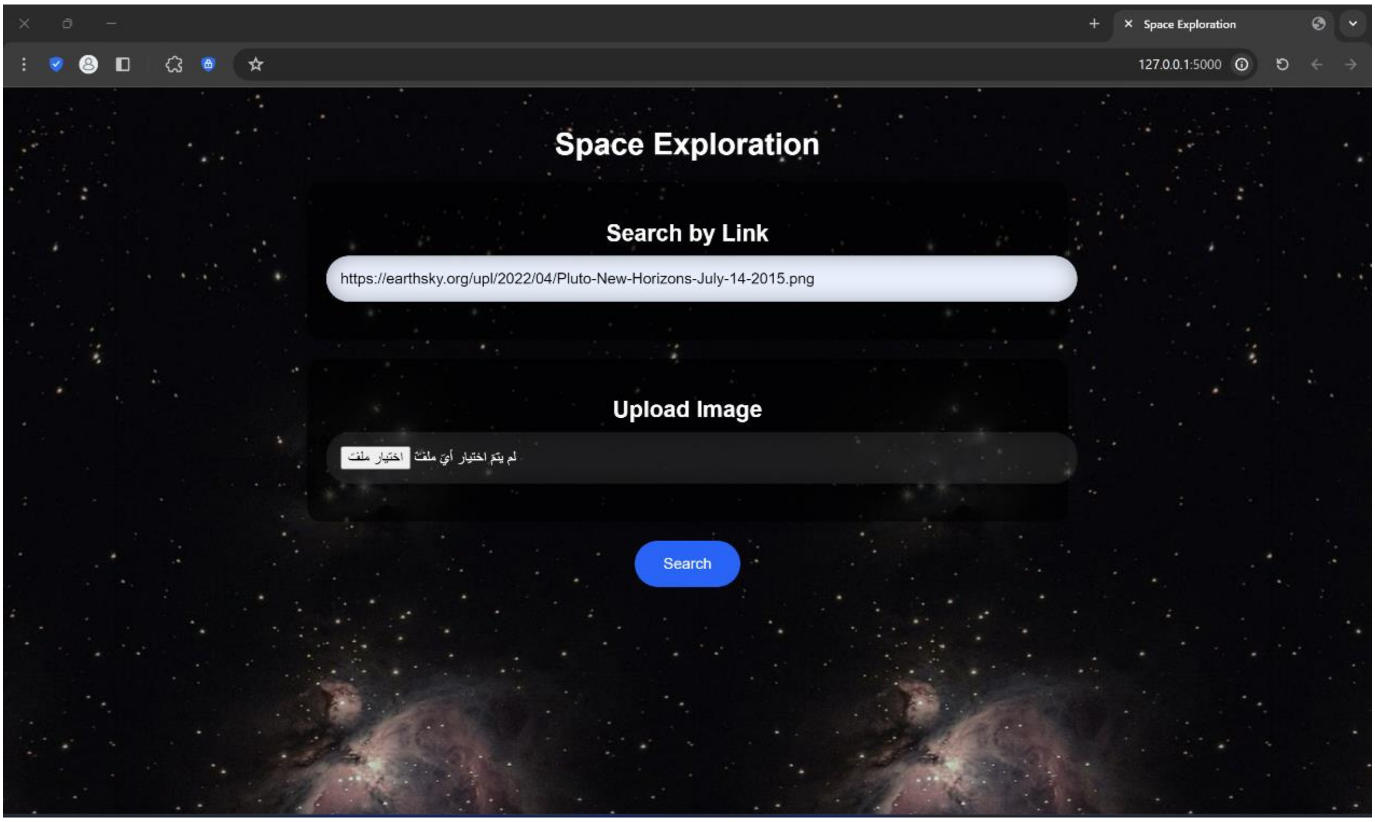


Figure 14 Search by Link

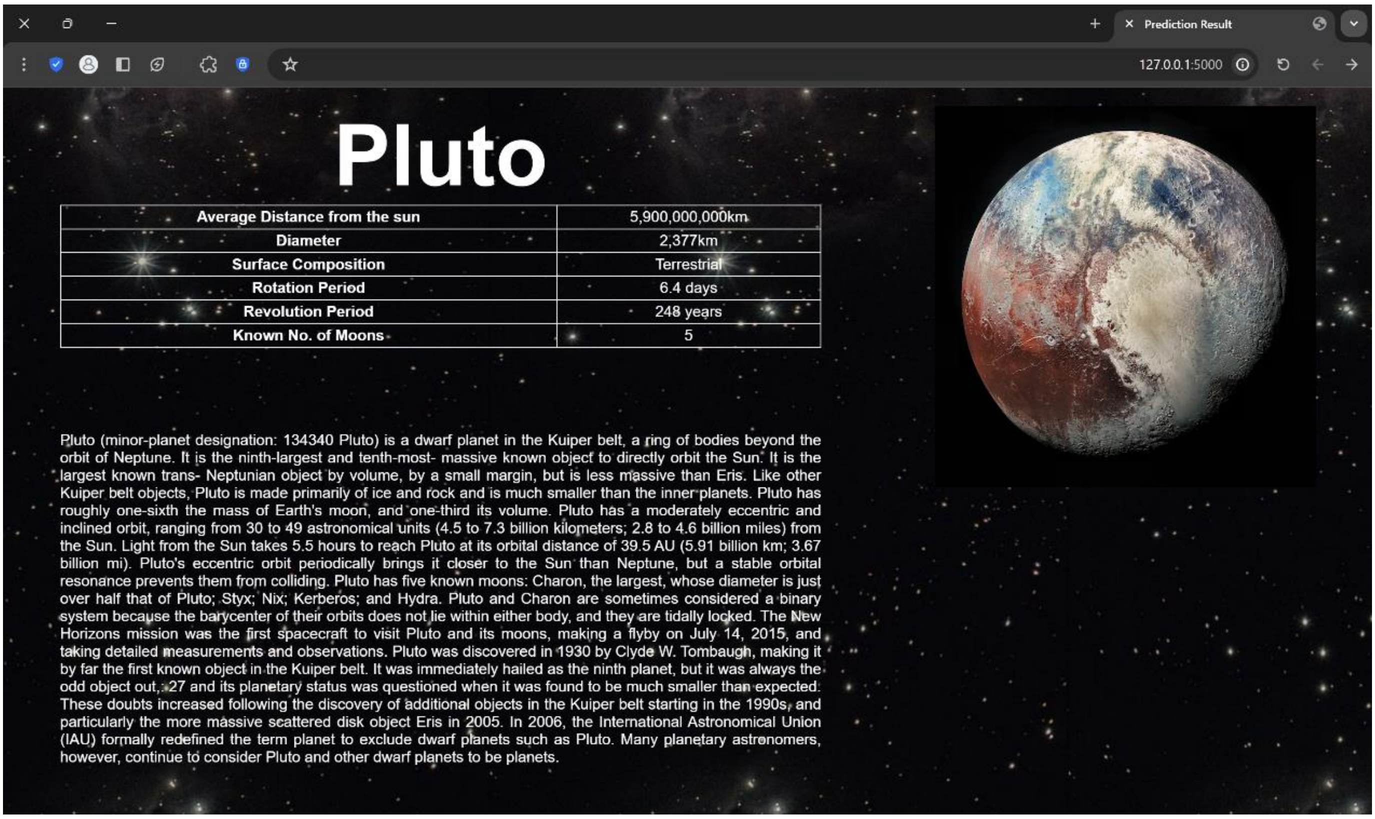


Figure13 result of Link

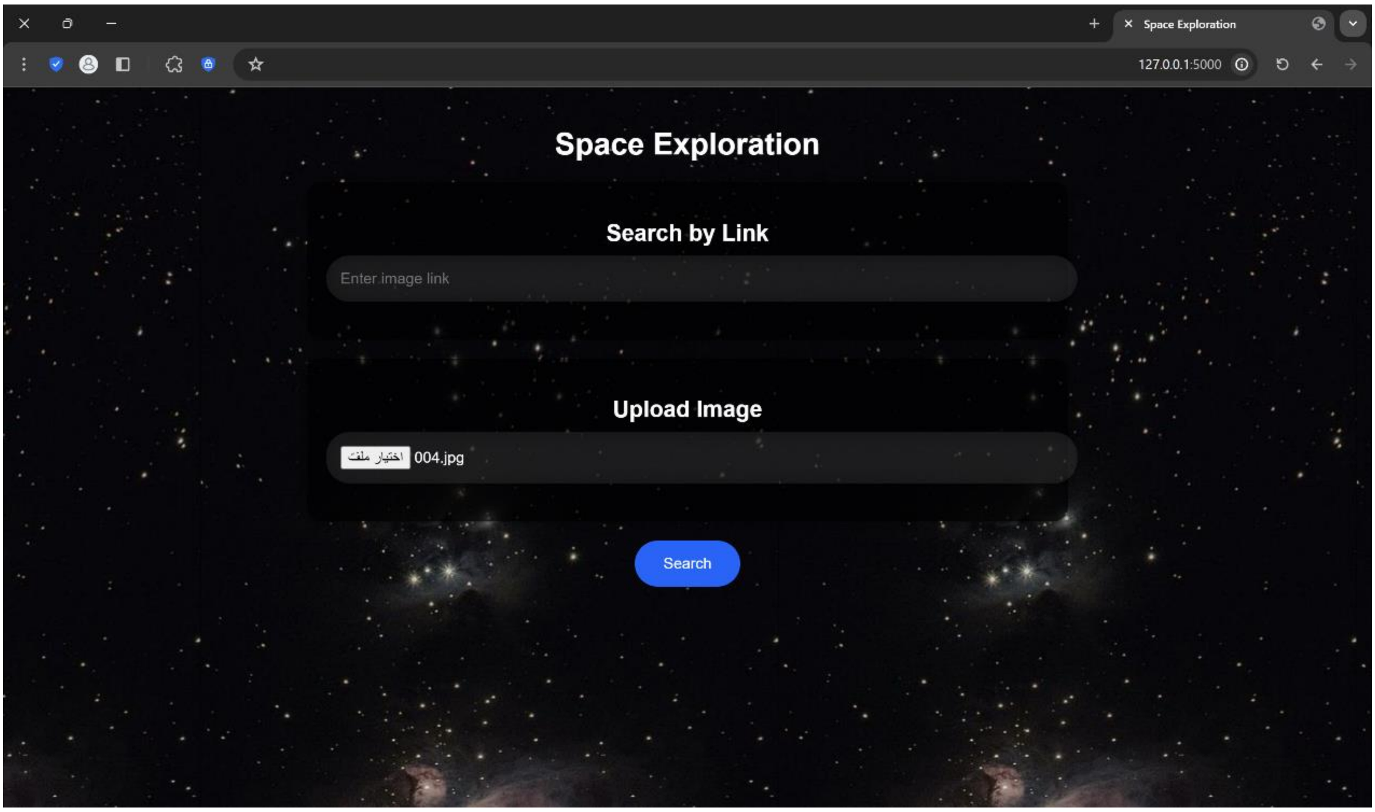


Figure14 search by file

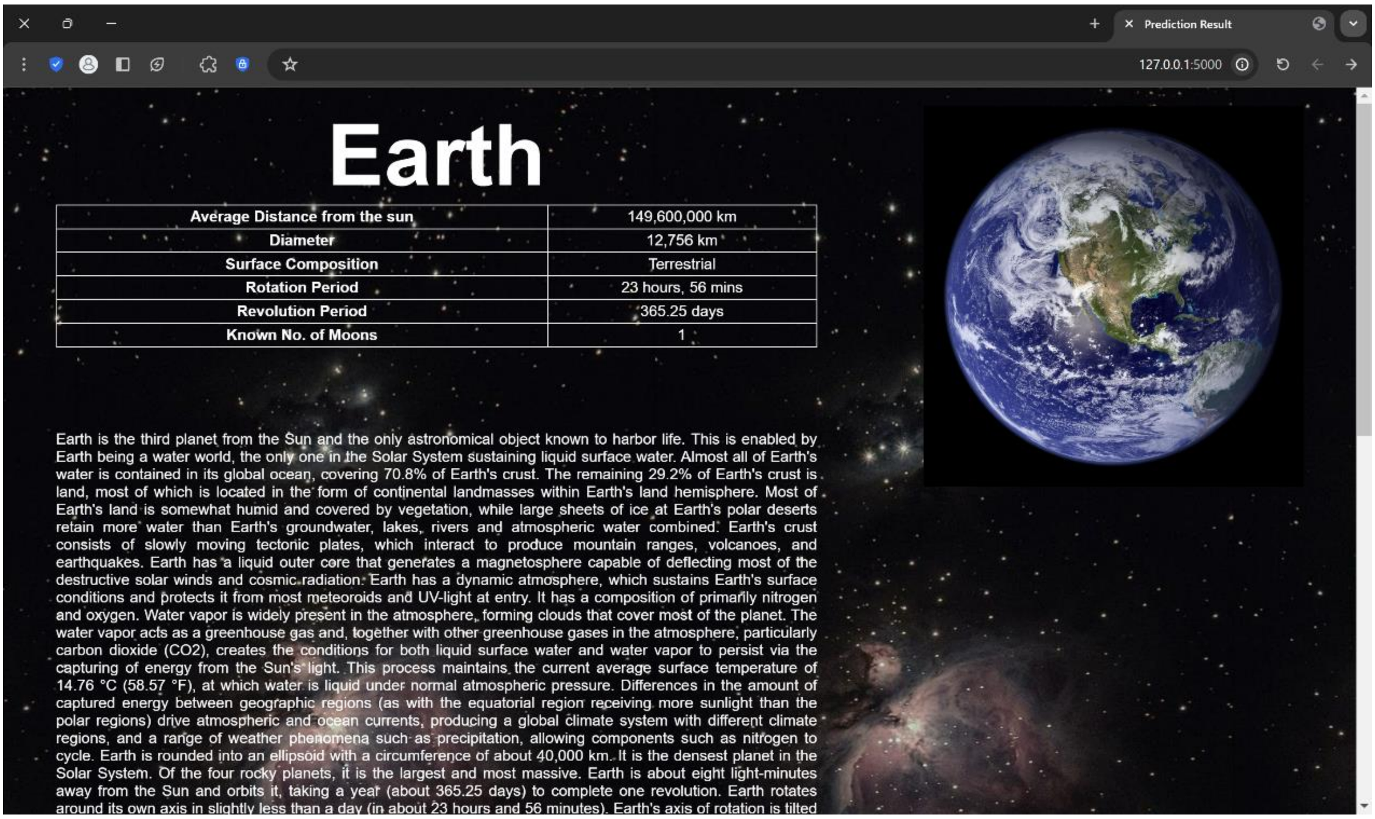


Figure 15 result from file

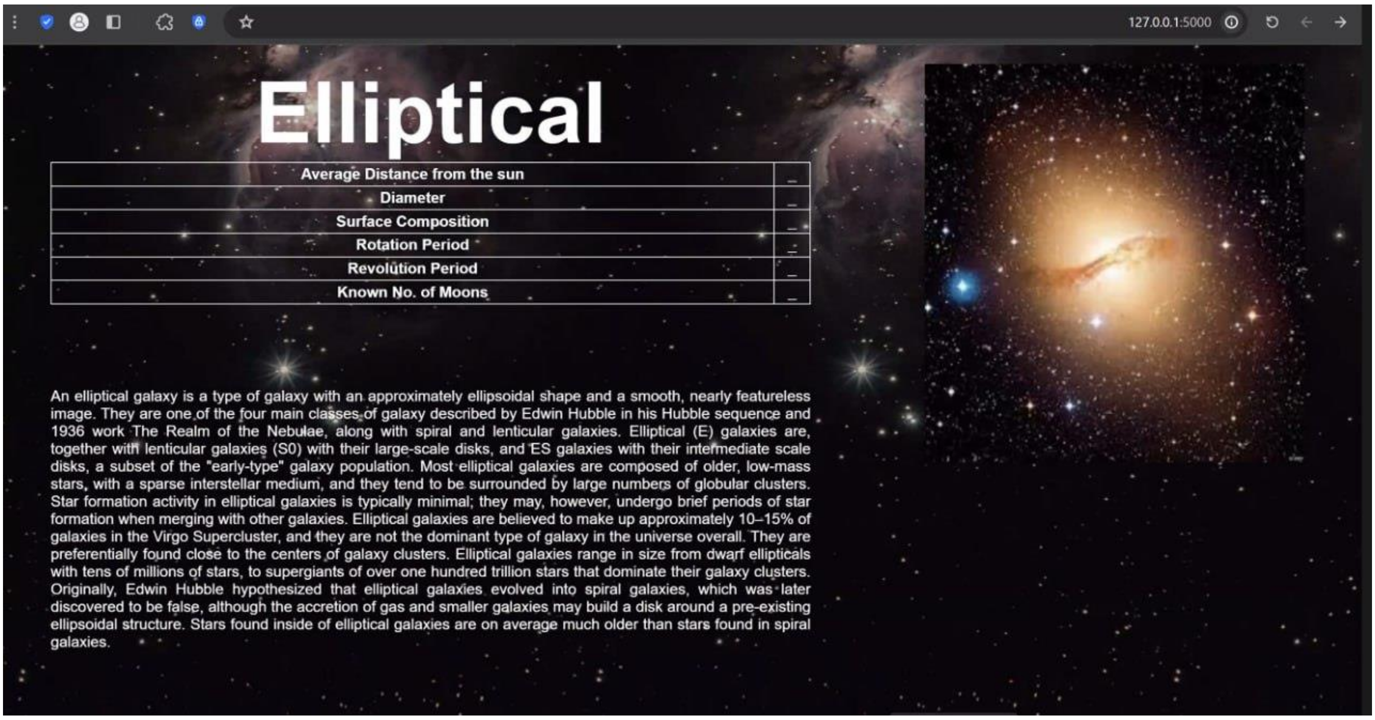


Figure 16 another Output

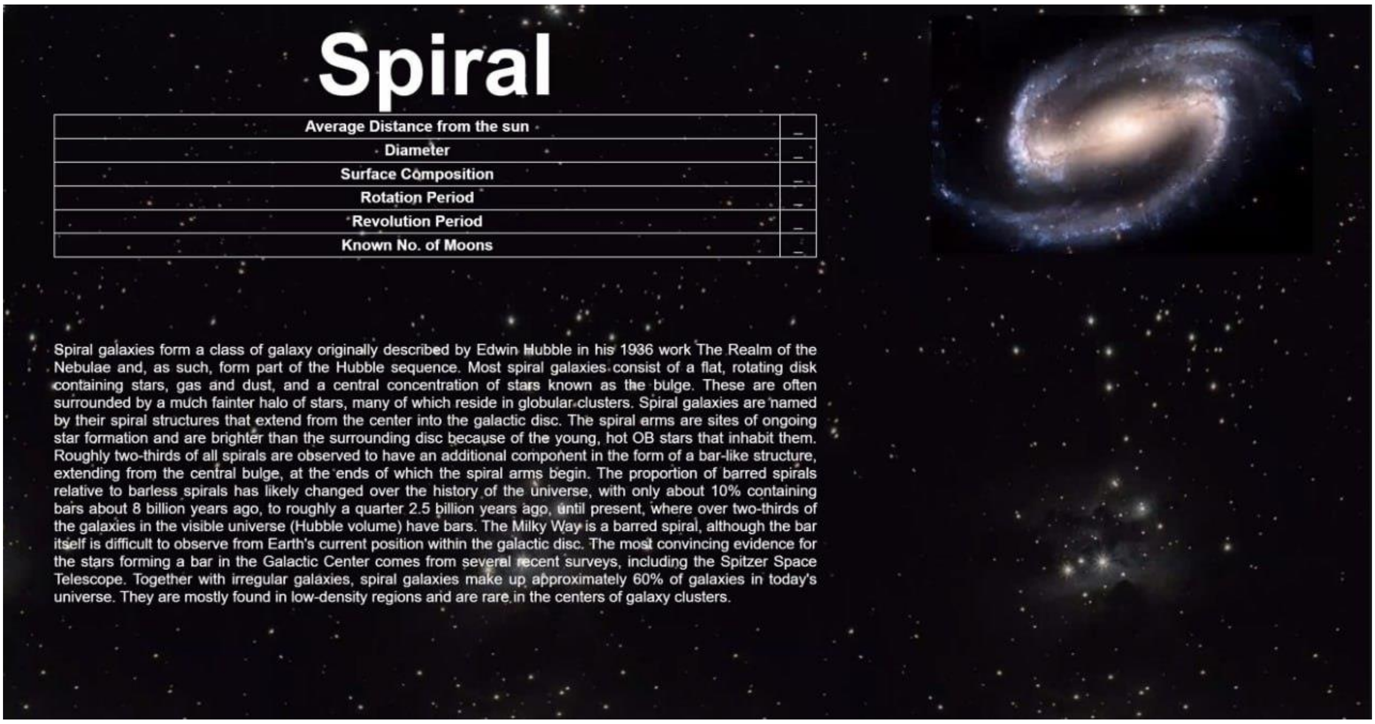


Figure 17 another Output

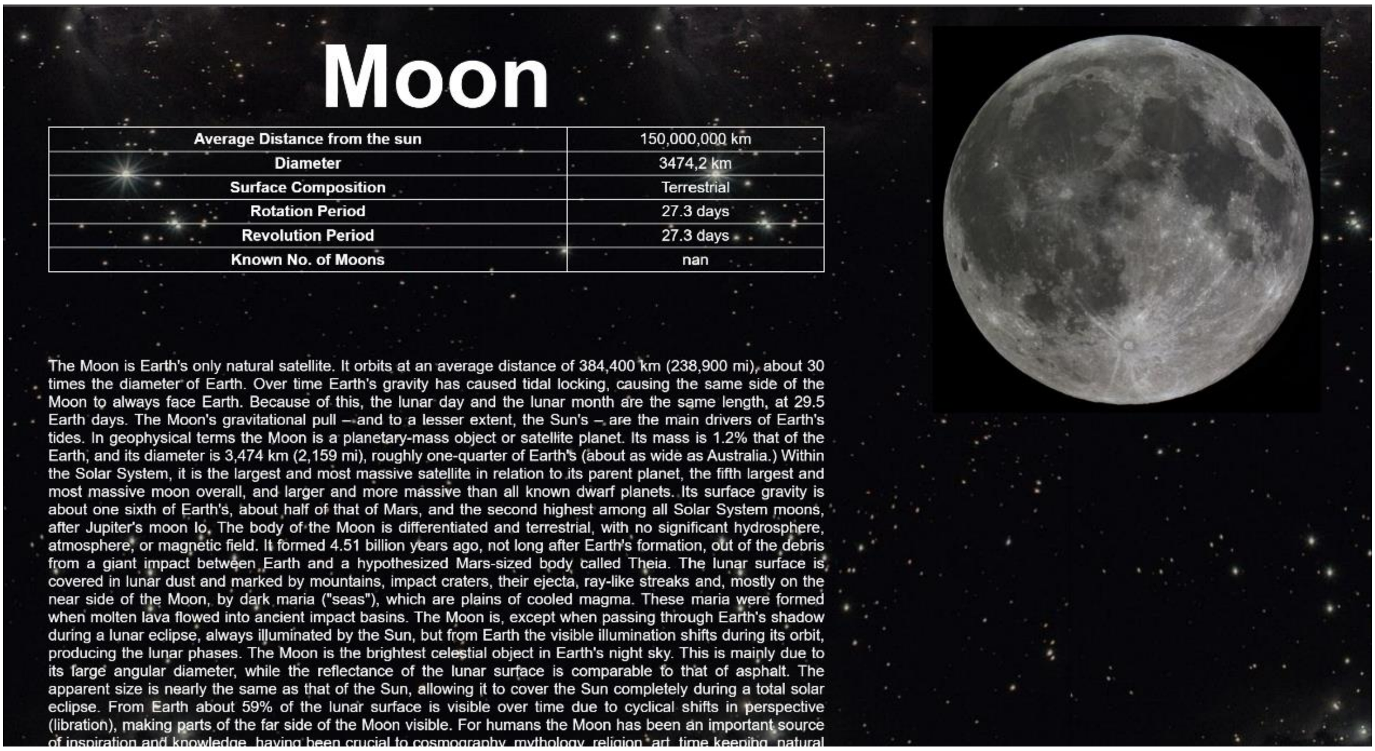


Figure 18 Output

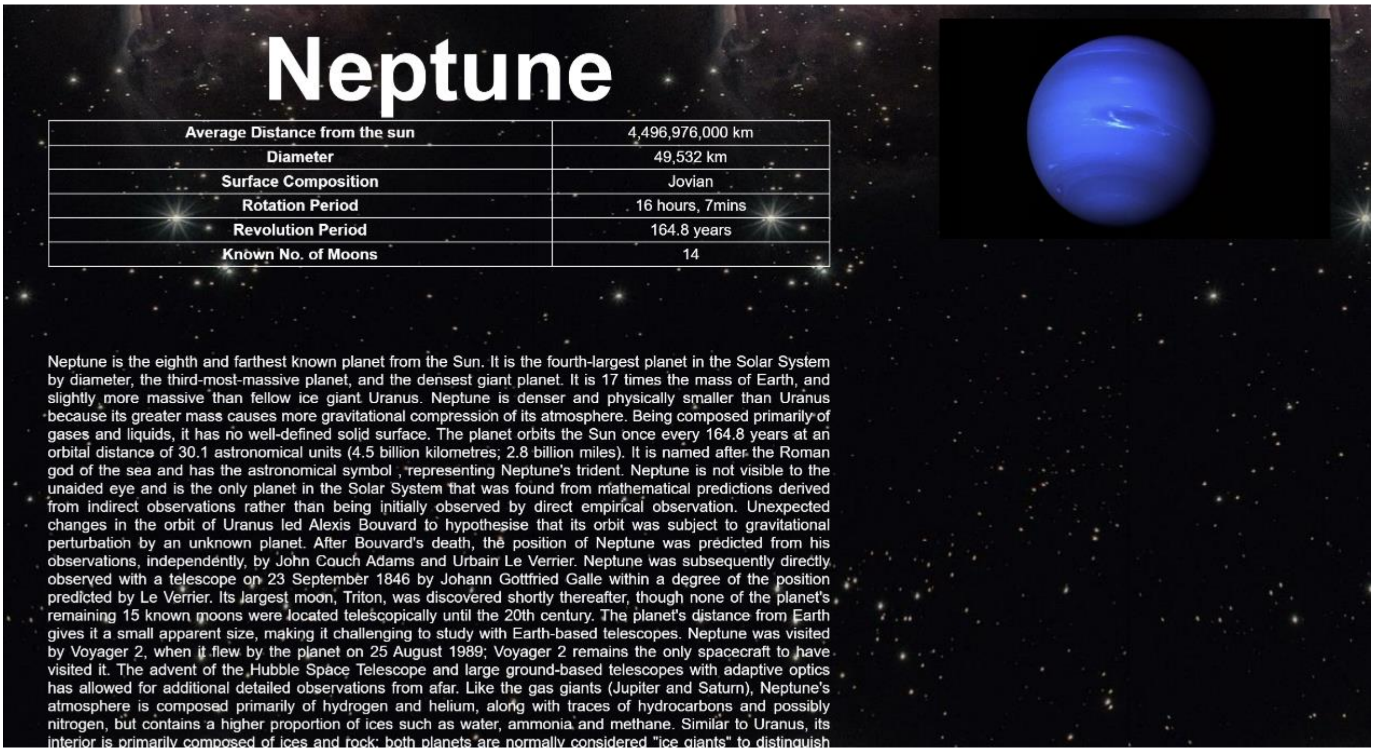


Figure19 Output

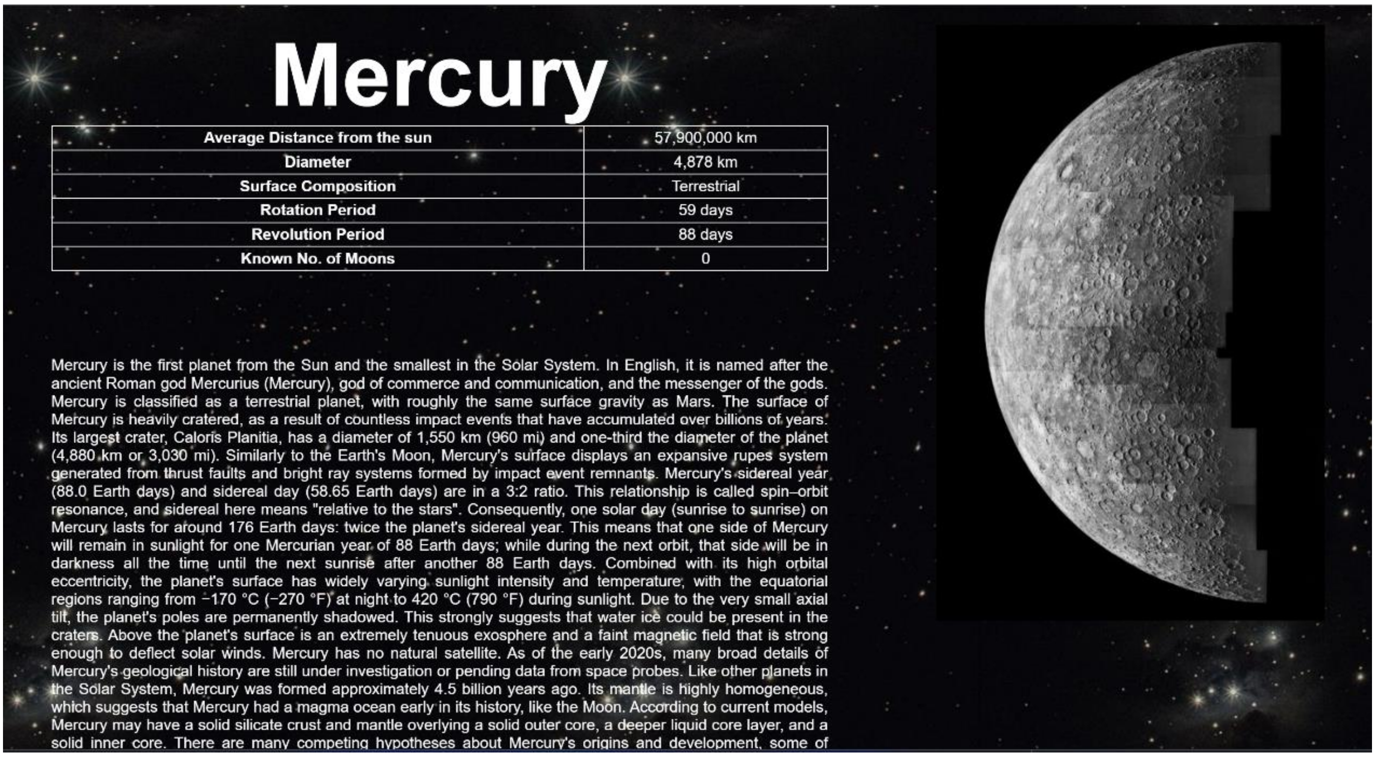
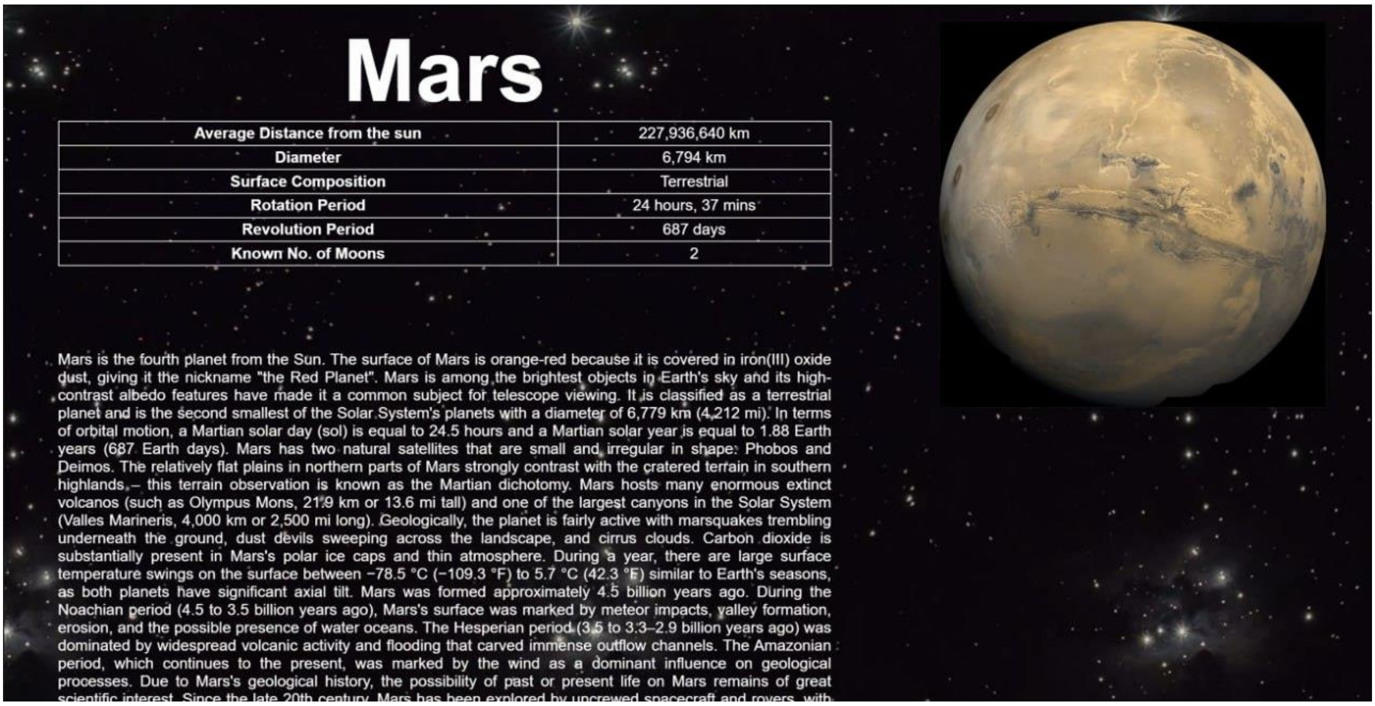


Figure20 Output



Output21 Figure

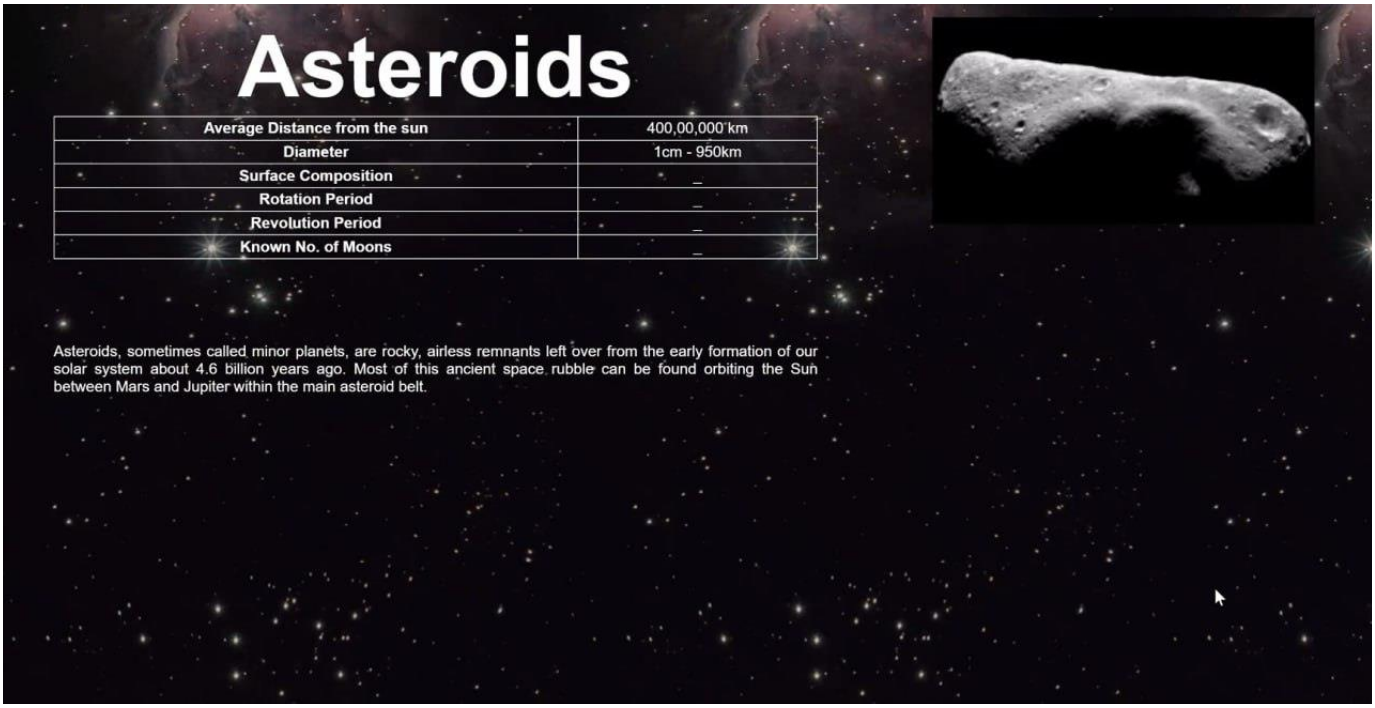


Figure22 Output

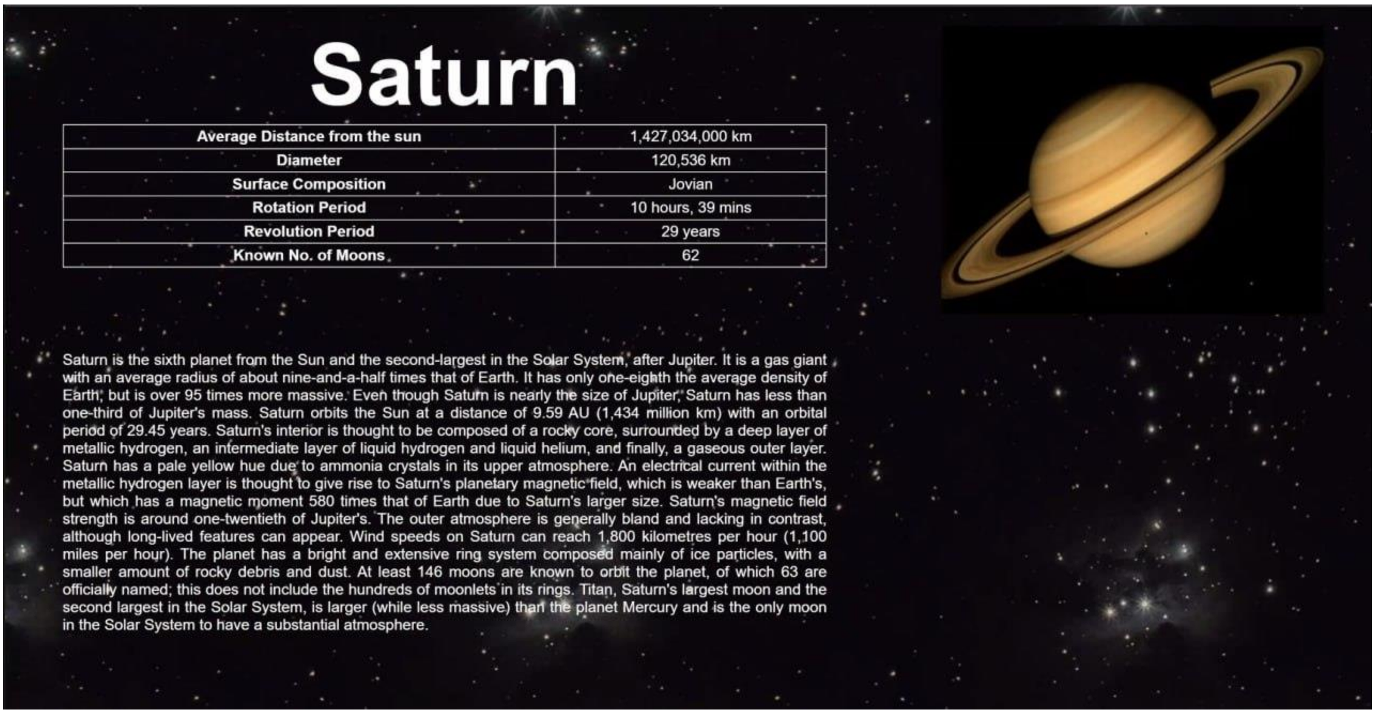


Figure 23 Output

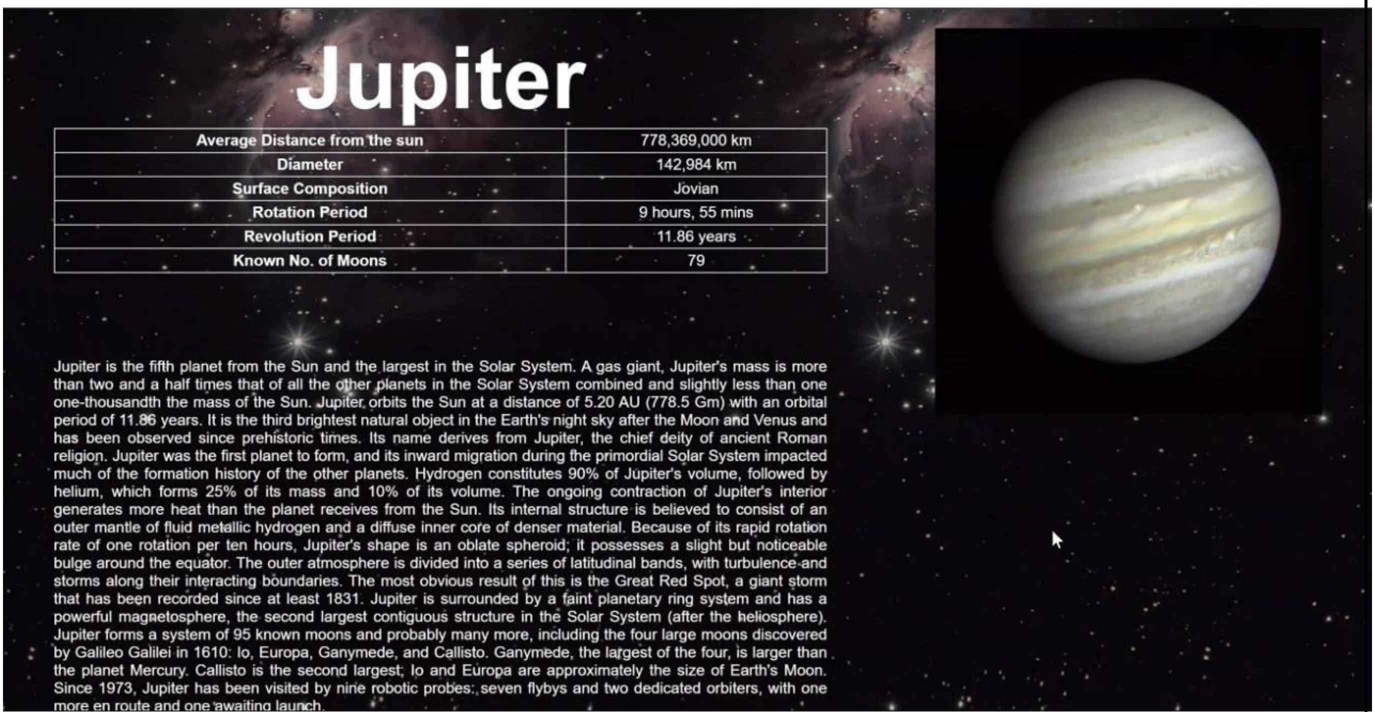


Figure 24 Output

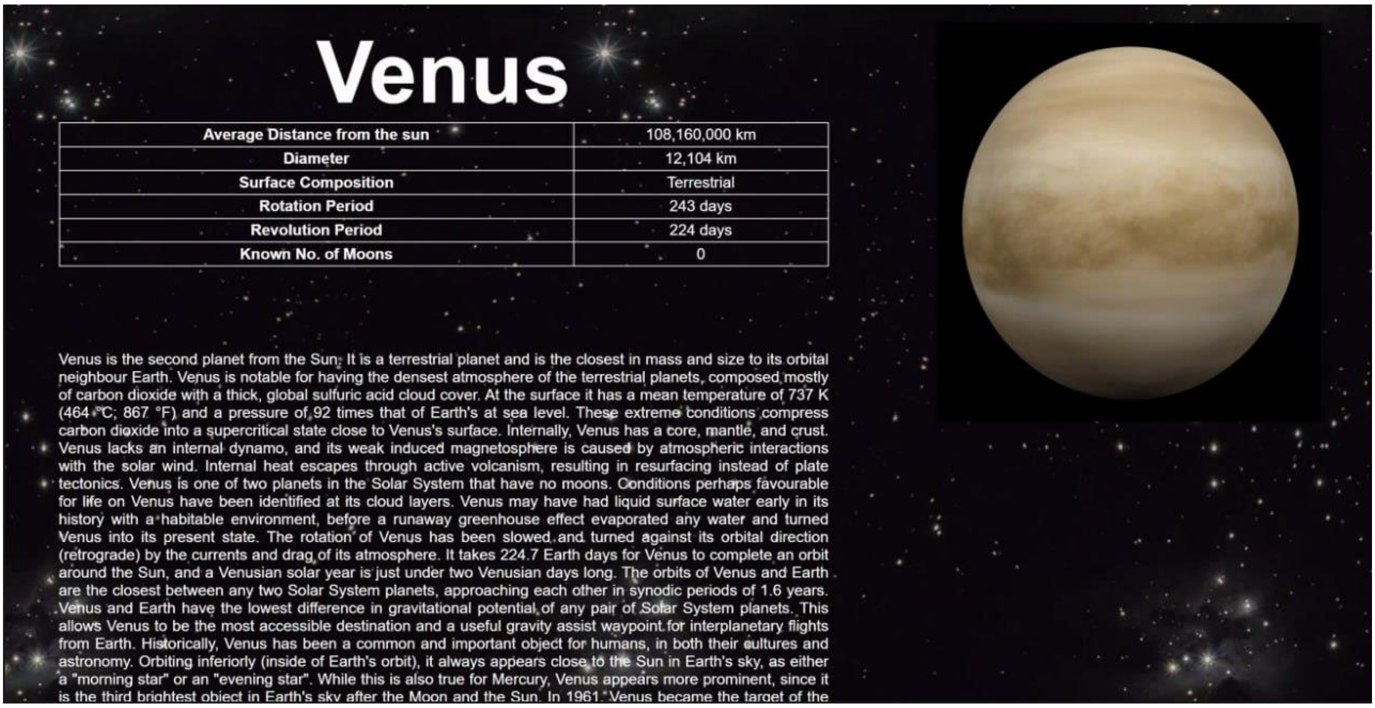


Figure25 Output

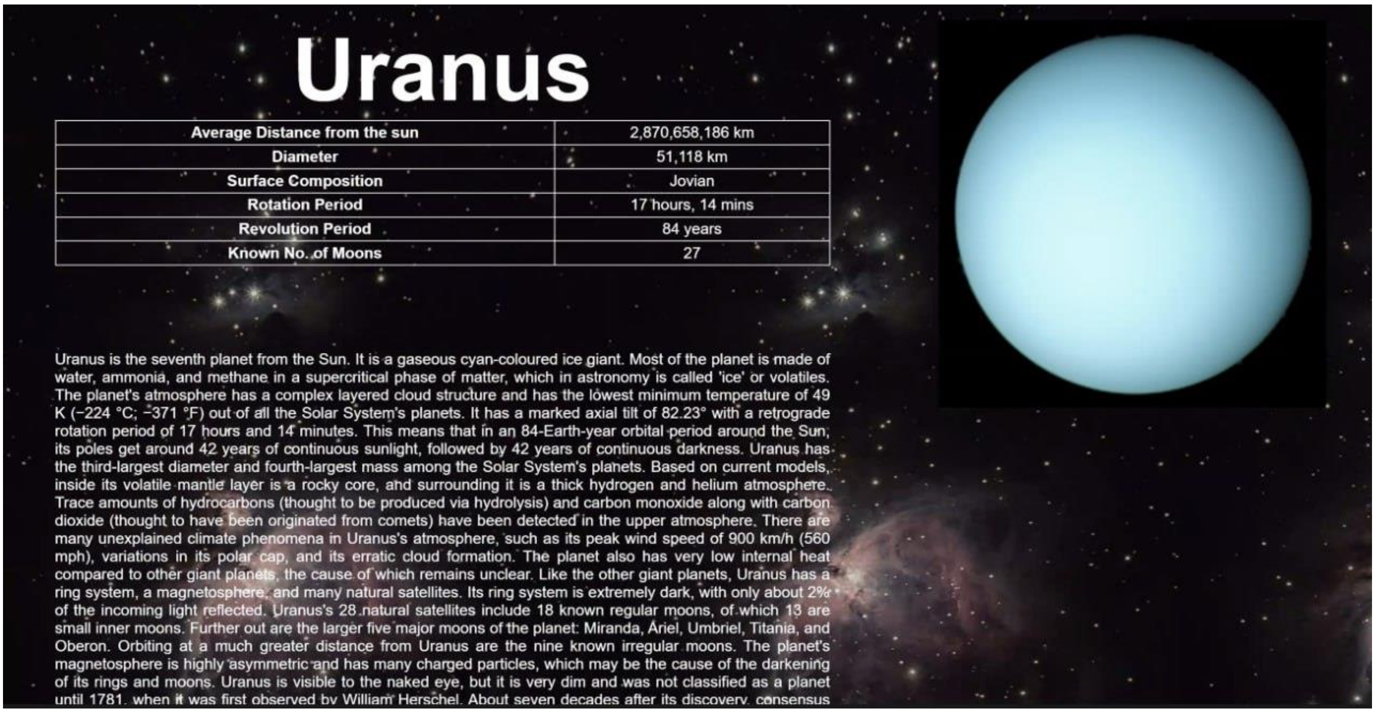
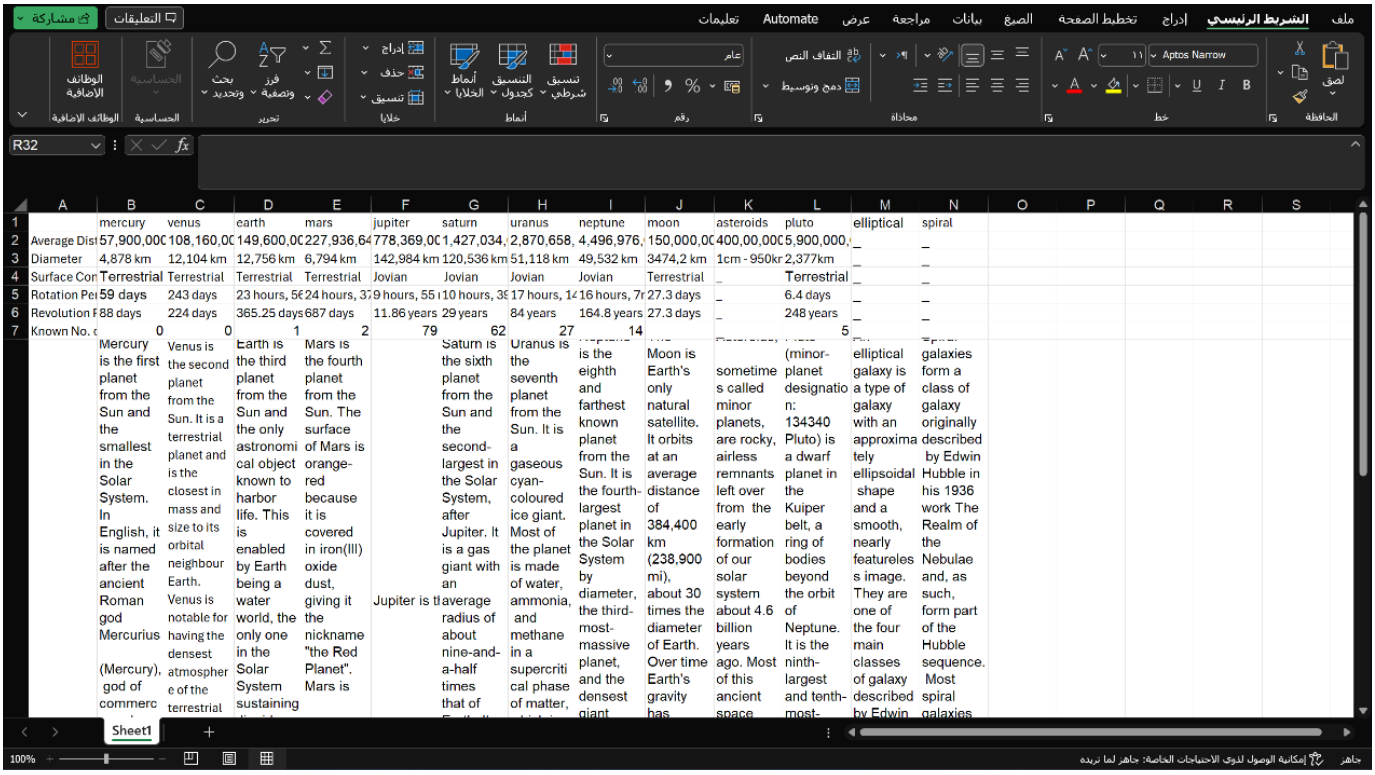


Figure26 Output



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		mercury	venus	earth	mars	jupiter	saturn	uranus	neptune	moon	asteroids	pluto	elliptical	spiral
2	Average Dist	57,900,00C	108,160,0C	149,600,0C	227,936,64	778,369,0C	1,427,034,	2,870,658,	4,496,976,	150,000,0C	400,00,00C	5,900,000,	_	_
3	Diameter	4,878 km	12,104 km	12,756 km	6,794 km	142,984 km	120,536 km	51,118 km	49,532 km	3474,2 km	1cm - 950kr	2,377km	_	_
4	Surface Con	Terrestrial	Terrestrial	Terrestrial	Terrestrial	Jovian	Jovian	Jovian	Jovian	Terrestrial	_	Terrestrial	_	_
5	Rotation Pe	59 days	243 days	23 hours, 5€	24 hours, 37	9 hours, 55	10 hours, 3£	17 hours, 14	16 hours, 7r	27.3 days	_	6.4 days	_	_
6	Revolution F	88 days	224 days	365.25 days	687 days	11.86 years	29 years	84 years	164.8 years	27.3 days	_	248 years	_	_
7	Known No. c	0	0	1	2	79	62	27	14		_	5	_	_
		Mercury is the first planet from the Sun and the smallest in the Solar System. In English, it is named after the ancient Roman god Mercurius (Mercury), god of commerc	Venus is the second planet from the Sun. It is a terrestrial planet and is the closest in mass and size to its orbital neighbour Earth. Venus is notable for having the densest atmospher e of the terrestrial	Earth is the third planet from the Sun and the only astronomi cal object known to harbor life. This is enabled by Earth being a water world, the only one in the Solar System sustaining	Mars is the fourth planet from the Sun. The surface of Mars is orange-red because it is covered in iron(III) oxide dust, giving it the nickname "the Red Planet". Mars is	Jupiter is tl	Saturn is the sixth planet from the Sun and the second-largest in the Solar System, after Jupiter. It is a gas giant with an average radius of about nine-and-a-half times that of	Uranus is the seventh planet from the Sun. It is a gaseous cyan-coloured ice giant. Most of the planet is made of water, ammonia, and methane in a supercriti cal phase of matter,	is the eighth and farthest known planet from the Sun. It is the fourth-largest planet in the Solar System by diameter, the third-most-massive planet, and the densest giant	Moon is Earth's only natural satellite. It orbits at an average distance of 384,400 km (238,900 mi), about 30 times the diameter of Earth. Over time Earth's gravity has	sometime s called minor planets, are rocky, airless remnants left over from the early formation of our solar system about 4.6 billion years ago. Most of this ancient space	(minor-planet designatio n: 134340 Pluto) is a dwarf planet in the Kuiper belt, a ring of bodies beyond the orbit of Neptune. It is the ninth-largest and tenth-most-	elliptical galaxy is a type of galaxy with an approxima tely ellipsoidal shape and a smooth, nearly featureles s image. They are one of the four main classes of galaxy described by Edwin	galaxies form a class of galaxy originally described by Edwin Hubble in his 1936 work The Realm of the Nebulae and, as such, form part of the Hubble sequence. Most spiral galaxies

Figure27 data set

7. CONCLUSION

7.1 Summary

The project "Space Exploration using AI Model" focuses on automating the classification and analysis of celestial body images through the use of artificial intelligence. The manual process of analyzing these images is labor-intensive and subjective, leading to inconsistencies and inefficiencies. By leveraging AI techniques such as deep learning and image recognition algorithms, an AI model is developed to accurately classify celestial body images and provide informative descriptions.

The model is trained using a large dataset of labeled images, enabling it to recognize patterns and features specific to different celestial bodies. Once trained, the model can classify new images and extract relevant information about the identified celestial bodies. This automated approach eliminates the need for time-consuming manual analysis, allowing researchers to focus on higher-level tasks and interpretations.

The potential applications of this project range from assisting astronomers in their research to providing accessible and interactive information about celestial bodies to the general public. By automating the classification process, astronomers can efficiently analyze vast amounts of data, uncovering new insights and accelerating their discoveries. Additionally, the AI model can be integrated into educational platforms and public outreach initiatives, allowing people of all backgrounds to explore and learn about the wonders of the universe.

The utilization of AI techniques, such as deep learning and image recognition algorithms, ensures the model's adaptability to new and complex celestial images. As the model continues to learn and improve its classification accuracy, it can keep pace with evolving scientific knowledge and advancements in space exploration. This dynamic nature of the AI model enables researchers to stay at the forefront of astronomical discoveries and encourages further innovation in the field.

The successful implementation of the "Space Exploration using AI Model" project highlights the transformative power of AI in pushing the boundaries of space exploration, unlocking new discoveries, and inspiring future advancements in our understanding of the universe. By combining the strengths of artificial intelligence and human expertise, we can delve deeper into the mysteries of space, expand our knowledge, and foster a greater appreciation for the vastness and complexity of the cosmos.

7.2 Limitations and Future Work

Limitations :

- The application interface is only in English

- The project does not cover predicting or providing information about upcoming celestial events or phenomena.

Future Work:

To improve our website in the future, we have the following plans:

- Celestial Events Focus: We will provide predictions and information about upcoming celestial events or phenomena.
- Multilingual Support: Our website will support multiple languages to cater to a wider audience.
- Mobile Application: We will develop a user-friendly application to make accessing our website easier for users.

References:

- https://www.tutorialsfreak.com/ai-tutorial/ai-space-exploration
- https://medium.com/@sandanshikunti/nasa-using-ai-and-machine-learning-for-space-exploration-e61af699307a
- https://youtu.be/uP-eXrc44tU?si=c5Irmy9bsiFVfItB
- https://www.geeksforgeeks.org/levels-in-data-flow-diagrams-dfd/
- https://youtu.be/i3RMlrx4ol4?si=rAgrlOrq5XUaAKYp
- https://www.workbreakdownstructure.com/work-breakdown-structure-examples
- https://www.lucidchart.com/pages/uml-class-diagram
- https://youtu.be/h6TJiGrYINk?si=g-kE6WFCG4PGTbW