# **NASA EPIC Data Viewer**

A Flask-based Web Application for Displaying NASA EPIC Satellite Data

## Introduction

The **NASA EPIC Data Viewer** is a simple web application built using the Flask framework in Python. It fetches and displays recent image data and metadata from NASA's EPIC (Earth Polychromatic Imaging data Camera) API. Users can view image information such as captions, date, coordinates, and image identifiers.

## Features

Integration with NASA's EPIC API

Displays:

* + Image Identifier
  + Caption
  + Capture Date
  + Image File Name
  + Centroid Coordinates (Latitude & Longitude)

Displays up to 4 of the latest EPIC records

Basic error handling for API issues

Simple and clean web interface using HTML, CSS, and Jinja2 templates

## System Requirements

* Python 3.13.2
* Flask
* requests library
* NASA API Key (Free from <https://api.nasa.gov/>)

## Technologies Used

| **Technology** | **Purpose** |
| --- | --- |
| Python | Backend Programming |
| Flask | Web Framework |
| Requests | HTTP API Requests |
| NASA EPIC API | Data Source |
| Jinja2 | Templating Engine |
| HTML & CSS | Frontend Design |

## Installation Steps

1. Install dependencies:
2. pip install flask requests
3. Get a NASA API key by registering at <https://api.nasa.gov/>
4. Replace the NASA\_API variable in app.py with your API key:

## Folder Structure

This is the normal structure of every flask-based website project. This is a simple flask project. In this project the folder structure and workflow are given below:

├── app.py Main Application Script

├── templates/

│ └── index.html Frontend Template

├── static/

│ └── css/

│ └── styles.css Optional CSS File

## How it Works

1. User accesses the home page (/).
2. The application sends a GET request to NASA's EPIC API.
3. Receives JSON response containing data of recent EPIC images data.
4. Displays the first 4 records (metadata).
5. Displays an error message if data is unavailable.

## 8. Code Explanation

**app.py**

* Imports required modules.
* Defines the NASA\_API key and API endpoint.
* Uses Flask to create a simple web server.
* Fetches data from NASA's EPIC API.
* Sends data to the index.html template.

**index.html**

index file is existed in template file. The is used to make a frontend of this EPIC app.

* Displays data dynamically using Jinja2 templating.
* Displays image metadata such as:
  + Identifier
  + Caption
  + Date
  + Image name
  + Centroid Coordinates
* Displays an error message if the data fetch fails.

## Sample Code Snippet (app.py)

In this section the code is shown that control our all work in such manner that data is fetching by request and then send to html file for representing.

@app.route("/",methods= ["GET"])

def index():

par = {"api\_key":NASA\_API}

response = requests.get(EPIC\_URL, params=par)

if response.status\_code == 200:

epic\_data = response.json()[:4]

return render\_template("index.html", EP = epic\_data)

else:

return render\_template("index.html", EP=None, error="Failed to fetch data from NASA API.")

## Running the Application

Run the application using the command: python app.py

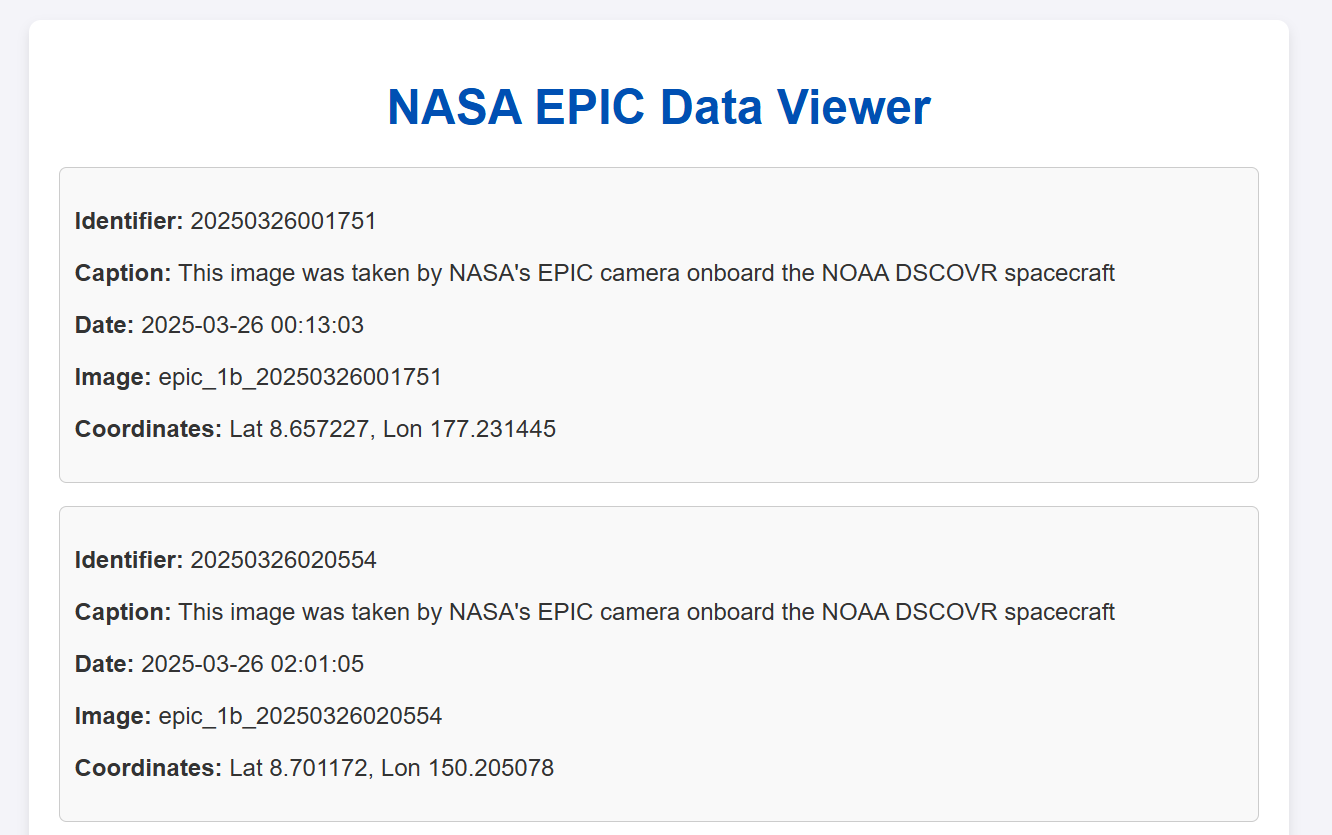
The application will start on: http://127.0.0.1:5000/

Open this URL in your browser.

## Notes

* The project is designed as a starting point for NASA EPIC API integration.
* You can extend it to:
  + Show actual EPIC images (not just metadata)
  + Create galleries or interactive maps
  + Add pagination to load more images

## Screenshot



## Author

* Developer: *M. Jawad Ahsan*
* Contact: [*m.jawadahs@gmail.com*](mailto:m.jawadahs@gmail.com)
* *Student: Superior University*