Lab 8: APOD

INFSCI 0201
Intermediate Programming (Spring 2025)

In this week's lab, we are continuing our journey into web development in Python. But this time, we are looking up to the sky. The task is simple: developing a web application showing NASA's current and past Astronomy Picture of the Day. You will get your hands dirty using web services and get more familiar with the web framework.

Submission Details

You are to write a Python program that meets the requirements outlined in the Lab 8 Tasks section.

- As with all programming assignments in this course, you must submit Lab8 through GitHub in the git repository the instructor created for you. You must place your Lab 8 eclipse project in the folder **/labs/lab08/**
- If you did this all correctly, this project will be in the directory **/labs/lab08/**

Testing

The main test for this lab is whether your web application can properly display the current Astronomy Picture of the Day and also display APOD based on the date given.

Note: Style guide violations will count as test failures! Be sure you're strictly adhering to the Python coding style guide on Canvas.

Lab 8 Tasks

1. First things first, you will need to go to the NASA API portal: https://api.nasa.gov/. It is recommended to get yourself a NASA developer key before proceeding to the next step. In NASA's case, you don't need the developer key to explore the API service (NASA has a default DEMO_KEY for this). However, most of the web services you will encounter in the future will ask for developer keys for full-scope access (This is also how these web services charge you for using their services).
2. Go to the APOD API section and read the instructions for this API endpoint. You can play around with the parameters for this request at this time.
3. Using Python to make API calls is relatively easy. You will need the `requests` package, which is the de facto standard for making HTTP requests in Python(https://requests.readthedocs.io/en/latest/).

```python
import requests

response = requests.get("https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY")

response.status_code
# check the HTTP status code 200 for success.
if response:
    print("Success!")
else:
    raise Exception(f"Non-success status code: {response.status_code}")

response.content # get the response’s content in bytes
response.text # get the response’s content in pure text format
response.json() # get the response’s content in a JSON object
```

4. For the application you are going to build for this lab, it is very simple. You will have two pages for your web app:
    a. Landing Page (Home Page): This is where you show the APOD for the current date. You also need to show the current date, the associated description for the image, and copyright information (if available). You should also have a link to your second page - History Page.
    b. History Page: On this page, you will ask the user to input a valid date (a. not in the future; b. not before June 16, 1995, when APOD first started), and you will display the APOD image of the specified date with the date, the associated description, and copyright information (if available). In addition, you should have a link for users to go back to the home page.
        i. Hint: You can use <input type="date"> to serve as the date picker. This tag will return a value that includes the year, month, and day. (https://www.w3schools.com/tags/att_input_type_date.asp)

5. **Some constraints**: Conceptually, this web application treats the APOD API endpoint as a data source. It is your model from MVC’s standpoint. So, you should treat the code that is responsible for retrieving APOD images as your model. And at least put them in a separate module. You will get a 10-point deduction if you don’t separate the model (image retrieval) and the controller for this lab.