Design document

Project summary:

The purpose of my application is providing some images about cosmos and astronomy. And users can learn some details about astronomy and rockets stuffs. Besides, users can do some quiz about astronomy and rockets. Make people be interested in astronomy and rockets! Features I plan to include: (Search: if you are using the NASA API, you could allow the user to search for the astronomy picture in a specific date. Details: if you are using the SpaceX Shows API, you could allow the user to view details about a specific rocket, such as if it is reused, if it is a successful landing, what is the landing type about it etc. Data export: if using a NASA API, students could allow users to export picture of astronomy in JPG. Gallery: if you are using the NASA API, you could allow the user to view a gallery of Mars images which are taken by specific camera. Quiz: students could create a quiz that tests users' knowledge of Astronomy.)

Description of REST API:

## REST API: APOD

URL: https://api.nasa.gov/planetary/apod

Documentation: https://github.com/nasa/apod-api

Description: I will fetch data about planet picture from the nasa apod REST API. I will use the data to display pictures about planets in the specific day.

### Endpoints:

* /planetary/apod/ start_date=2017-07-08&end_date=2017-07-10 – get the astronomy picture from 2017-07-08 to 2017-07-10

## REST API: **Mars Rover Photos**

URL: https://api.nasa.gov/mars-photos/

Documentation: https://github.com/corincerami/mars-photo-api

Description: I will fetch data about mars photo from the **Mars Rover Photos** REST API. I will display the photos in galleries categoried by different cameras.

### Endpoints:

* sol=1000&camera=fhaz – get the photos taken by camera fhaz
* earth_date=2015-6-3&camera=pancam – get the photos taken by camera pancam

## REST API: **Get all rockets**

URL: https://api.spacexdata.com/v4/rockets

Documentation: https://github.com/r-spacex/SpaceX-API/blob/master/docs/rockets/v4/all.md

Description: I will fetch data about rockets from the Get all rockets REST API. I will use the data to display information about rockets, including the name, the height, the mass, and the image.

### Endpoints:

* `/rockets/:id `– get a single rocket information
* /rockets/ ` - get all rockets information

List of features:

## Feature: Search for astronomy picture

Description: Users can search for astronomy by specific date. The search results will appear the picture in specific date.

Model (data class): `AstronomyPicture `

REST API endpoint: `/search/planetary/apod `

Pages: `search_page`

## Feature: details for rockets in SpaceX

Description: Users can find details about rockets in SpaceX. The detail results will include like the height, the mass, the name and the picture of the rockets.

Model (data class): `Rocket `

REST API endpoint: `/detail/rockets `

Pages: `detail_page`

## Feature: data export for astronomy pictures

Description: Users can export the astronomy picture in the specific day.

Model (data class): ` AstronomyPicture `

REST API endpoint: `/search/planetary/apod `

Pages: `search_page`

## Feature: gallery for Mars photos

Description: Users can see the gallery of the Mars photos

Model (data class): ` MarsPhotos`

REST API endpoint: `/gallery/mars_photos/api/v1/rovers/curiosity/photo `

Pages: `gallery_page`

## Feature: quiz for rockets in SpaceX

Description: Users can do some quiz about rockets in Spacex to get new knowledge

Model (data class): ` Rocket `

REST API endpoint: `/detail/rockets `

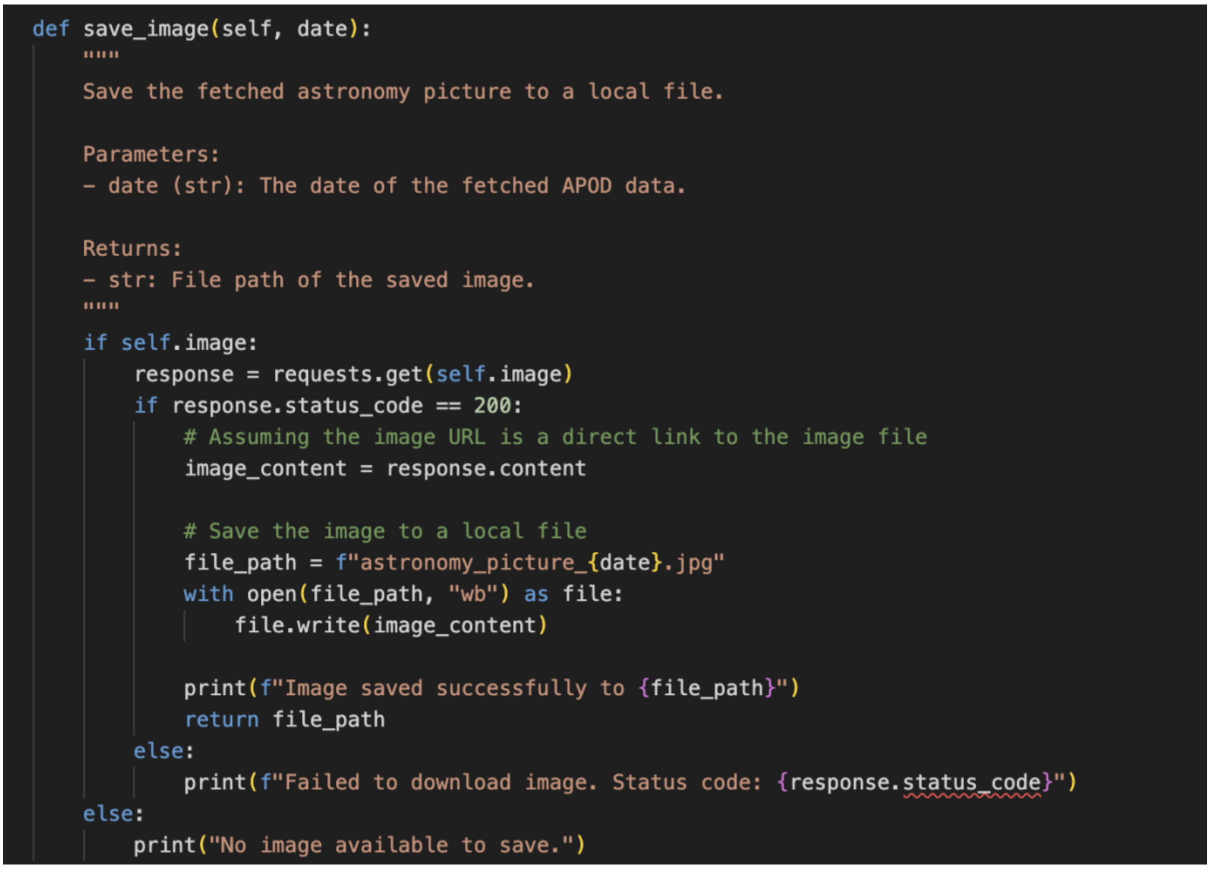
Pages: `quiz_page`

Reference:

1: https://northeastern.instructure.com/courses/156822/pages/tutorial-mocking-rest-api-calls?module_item_id=9667274

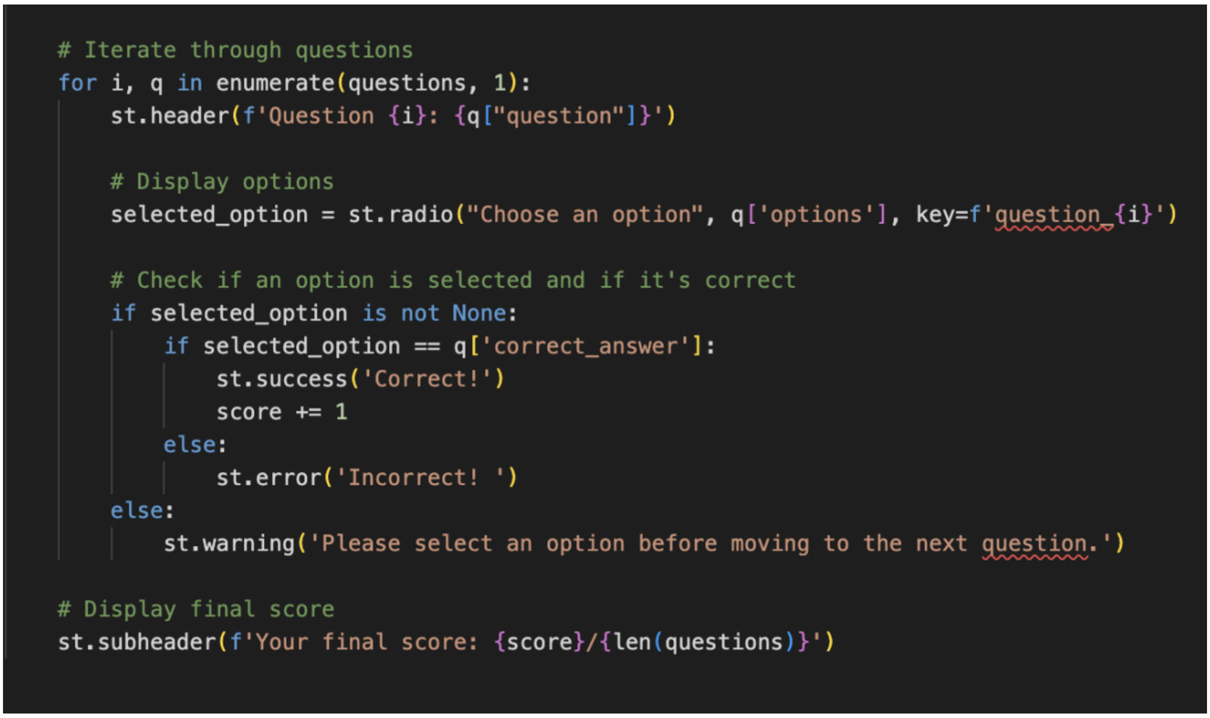
2: https://northeastern.instructure.com/courses/156822/pages/tutorial-passing-data-between-pages?module_item_id=9665380

Code highlights:



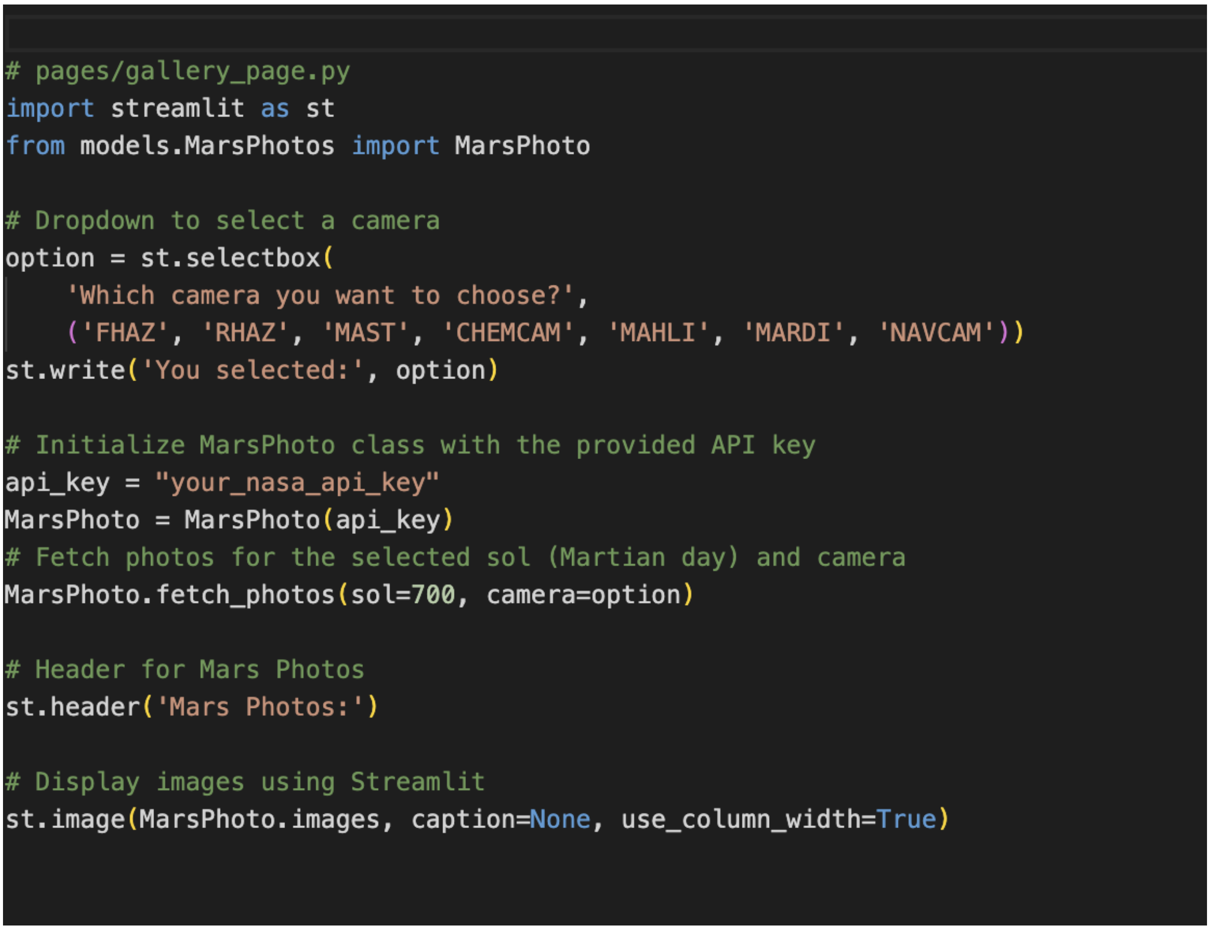
```
def save_image(self, date):
    """
    Save the fetched astronomy picture to a local file.

    Parameters:
    - date (str): The date of the fetched APOD data.

    Returns:
    - str: File path of the saved image.
    """
    if self.image:
        response = requests.get(self.image)
        if response.status_code == 200:
            # Assuming the image URL is a direct link to the image file
            image_content = response.content

            # Save the image to a local file
            file_path = f"astronomy_picture_{date}.jpg"
            with open(file_path, "wb") as file:
                file.write(image_content)

            print(f"Image saved successfully to {file_path}")
            return file_path
        else:
            print(f"Failed to download image. Status code: {response.status_code}")
    else:
        print("No image available to save.")
```

I think this save_image part makes me proud of it. Because when I want to make a download button, I need a image url to link it. So I made this function save_image to do it. And it is a very struggle process for me a rookie. But I made it finally, I use with open() as file In the function and return the file_path. It makes me link the image url to download button.



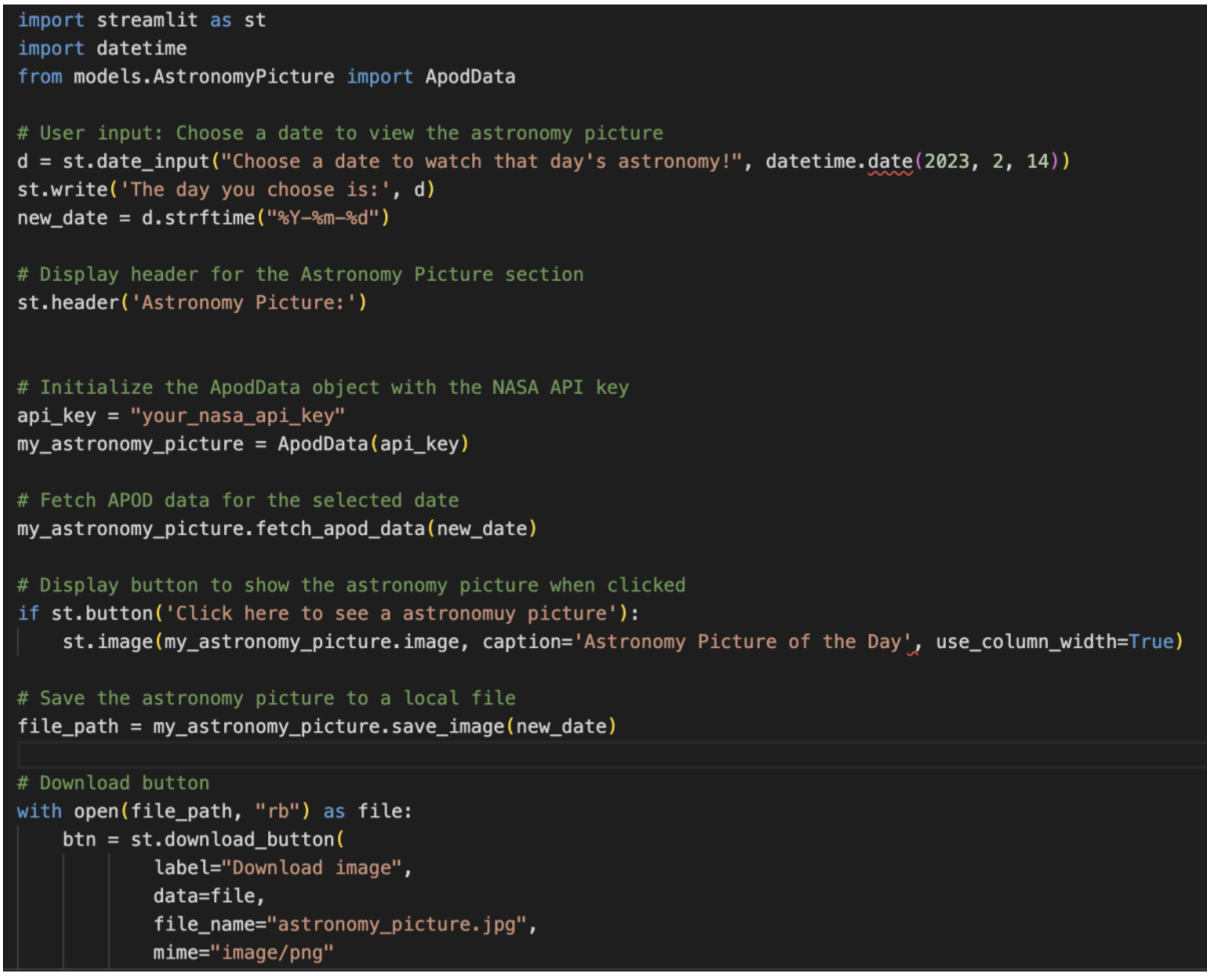
```
# Iterate through questions
for i, q in enumerate(questions, 1):
    st.header(f'Question {i}: {q["question"]}')

    # Display options
    selected_option = st.radio("Choose an option", q['options'], key=f'question_{i}')

    # Check if an option is selected and if it's correct
    if selected_option is not None:
        if selected_option == q['correct_answer']:
            st.success('Correct!')
            score += 1
        else:
            st.error('Incorrect! ')
    else:
        st.warning('Please select an option before moving to the next question.')

# Display final score
st.subheader(f'Your final score: {score}/{len(questions)}')
```

I just use enumerate to make the quiz app and use the st.radio to do the options in the quiz app.



```python
# pages/gallery_page.py
import streamlit as st
from models.MarsPhotos import MarsPhoto

# Dropdown to select a camera
option = st.selectbox(
    'Which camera you want to choose?',
    ('FHAZ', 'RHAZ', 'MAST', 'CHEMCAM', 'MAHLI', 'MARDI', 'NAVCAM'))
st.write('You selected:', option)

# Initialize MarsPhoto class with the provided API key
api_key = "your_nasa_api_key"
MarsPhoto = MarsPhoto(api_key)
# Fetch photos for the selected sol (Martian day) and camera
MarsPhoto.fetch_photos(sol=700, camera=option)

# Header for Mars Photos
st.header('Mars Photos:')

# Display images using Streamlit
st.image(MarsPhoto.images, caption=None, use_column_width=True)
```

I use st.selectbox to make the options to let users watch the Mars photos by different cameras.



```
import streamlit as st
import datetime
from models.AstronomyPicture import ApodData

# User input: Choose a date to view the astronomy picture
d = st.date_input("Choose a date to watch that day's astronomy!", datetime.date(2023, 2, 14))
st.write('The day you choose is:', d)
new_date = d.strftime("%Y-%m-%d")

# Display header for the Astronomy Picture section
st.header('Astronomy Picture:')


# Initialize the ApodData object with the NASA API key
api_key = "your_nasa_api_key"
my_astronomy_picture = ApodData(api_key)

# Fetch APOD data for the selected date
my_astronomy_picture.fetch_apod_data(new_date)

# Display button to show the astronomy picture when clicked
if st.button('Click here to see a astronomuy picture'):
    st.image(my_astronomy_picture.image, caption='Astronomy Picture of the Day', use_column_width=True)

# Save the astronomy picture to a local file
file_path = my_astronomy_picture.save_image(new_date)

# Download button
with open(file_path, "rb") as file:
    btn = st.download_button(
            label="Download image",
            data=file,
            file_name="astronomy_picture.jpg",
            mime="image/png"
```

I use strftime to convert the date format to the suitable data format to be used in fetch_apod_data. It is very tricky I think. So It is also my code highlight.

Next steps:

1. I think I can make the quiz page app more perfect. I just use st.radio to make the options, but I can not choose nothing when you enter the quiz page(It means I must

choose one option in the first). So I will use other methods to make it in the future I think.

2. I think in detail page, I will make the numeric details dataframe more neated, because the data in the frame may not matched much to the unit now. So I want to figure it out in the future.
3. I think in gallery page, sometimes there are no Mars photos by some cameras. So I want to cancel these choices when there are no Mars photos by some specific cameras to make the entire app much neat and useful

Reflection:

I learned a lot from this project, like how to use streamlit and how to use the REST API. And some tricky things like strftime to convert date format and how to save an image content to a file path and use the file path do something else like download this image. I think the most challenge part is the test part and how to use the data from REST API, because the data type are so complex and different. I think the most rewarding part is making the download image button, it makes me know how save an image content to a file path and link it to the download button to achieve the goal. Next time I would do something more challenge and complex, like make some dynamic picture of the Astronomy and make some dynamic chart about planets. I think these things would be very interesting for me!