



Nasa API Project

Documentation

Application Frameworks - SE3040

Assignment 2

IT Number - IT21334238

Name - Sembakutti S.C

Table of Contents

1. NASA APIs	3
NASA APIs used in the system.....	3
APOD – Astronomy Picture of the Day.....	3
EPIC - Earth Polychromatic Imaging Camera	
EONET - The Earth Observatory Natural Event Tracker	Error! Bookmark not defined.
2. Challenges Faced and How They Resolved	Error! Bookmark not defined.
3. Responsive User Interface	Error! Bookmark not defined.

1. NASA APIs

NASA APIs used in the System.

APOD – Astronomy Picture of the Day

APOD provides stunning images of the cosmos along with informative descriptions, making it engaging and captivating for users interested in astronomy and space exploration. This API enhance the visual appeal of your application, providing users with awe-inspiring views of space and celestial objects. This feature can attract users and encourage exploration of the app's content. The informative descriptions accompanying APOD images provide valuable context and insights into astronomical topics. Users can learn about different aspects of astronomy, from galaxies and nebulae to planets and stars, enriching their understanding of the universe. By incorporating the APOD API, the application can display a new astronomical image and description each day, offering users a daily dose of cosmic wonder. This regular updating of content keeps the app dynamic and encourages user engagement over time.

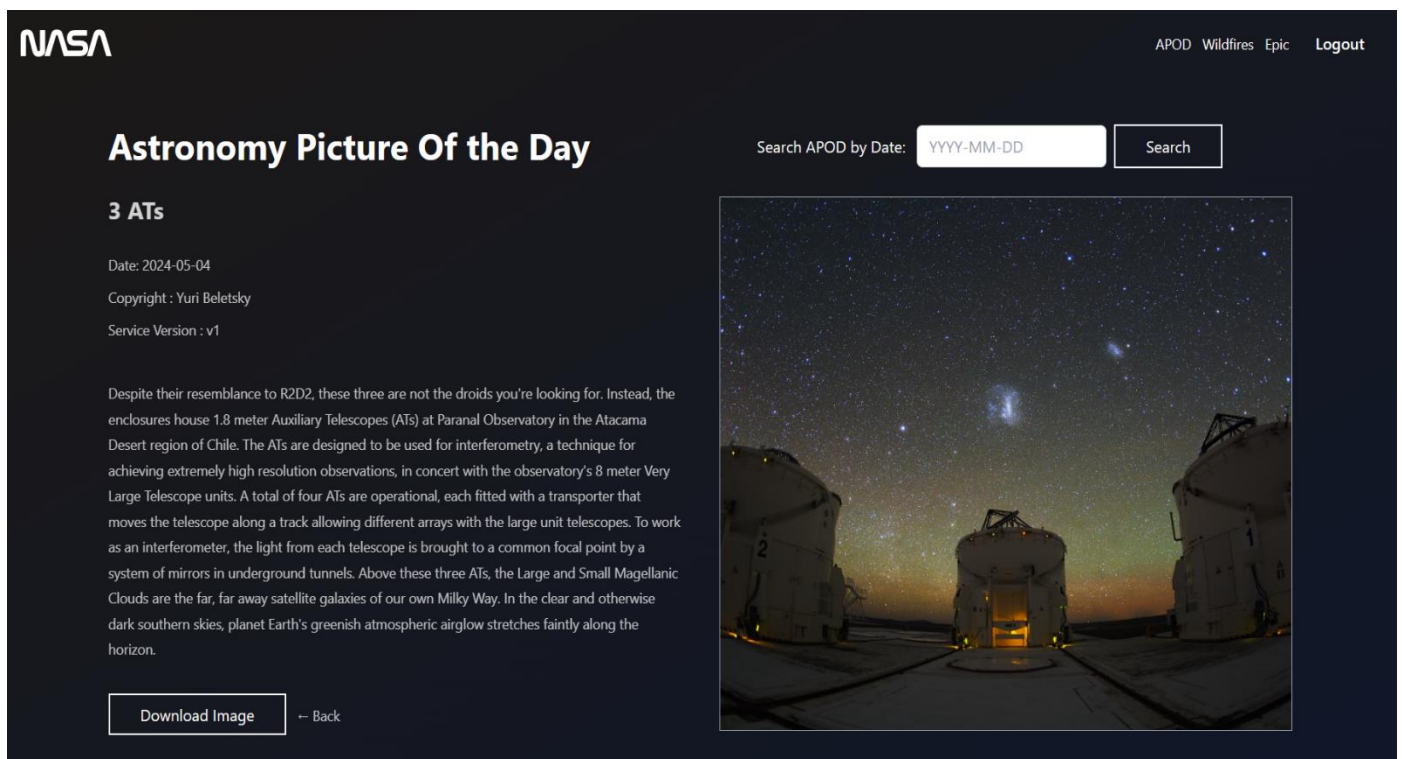


Figure 1: Astronomy Picture of the Day Page

EPIC - Earth Polychromatic Imaging Camera

The Earth Polychromatic Imaging Camera (EPIC) API, developed by NASA, provides a vast collection of daily Earth imagery captured by the EPIC instrument aboard the DSCOVR satellite. This API was chosen for its ability to offer users access to high-resolution images of Earth from a unique perspective in space. In this project, the EPIC API serves as the primary data source, allowing users to search for imagery based on specific dates and explore Earth's natural beauty through an intuitive web interface. By leveraging the EPIC API, users can gain insights into various natural phenomena, such as cloud formations, atmospheric events, and changes in Earth's surface over time, fostering scientific understanding and public engagement with Earth science.

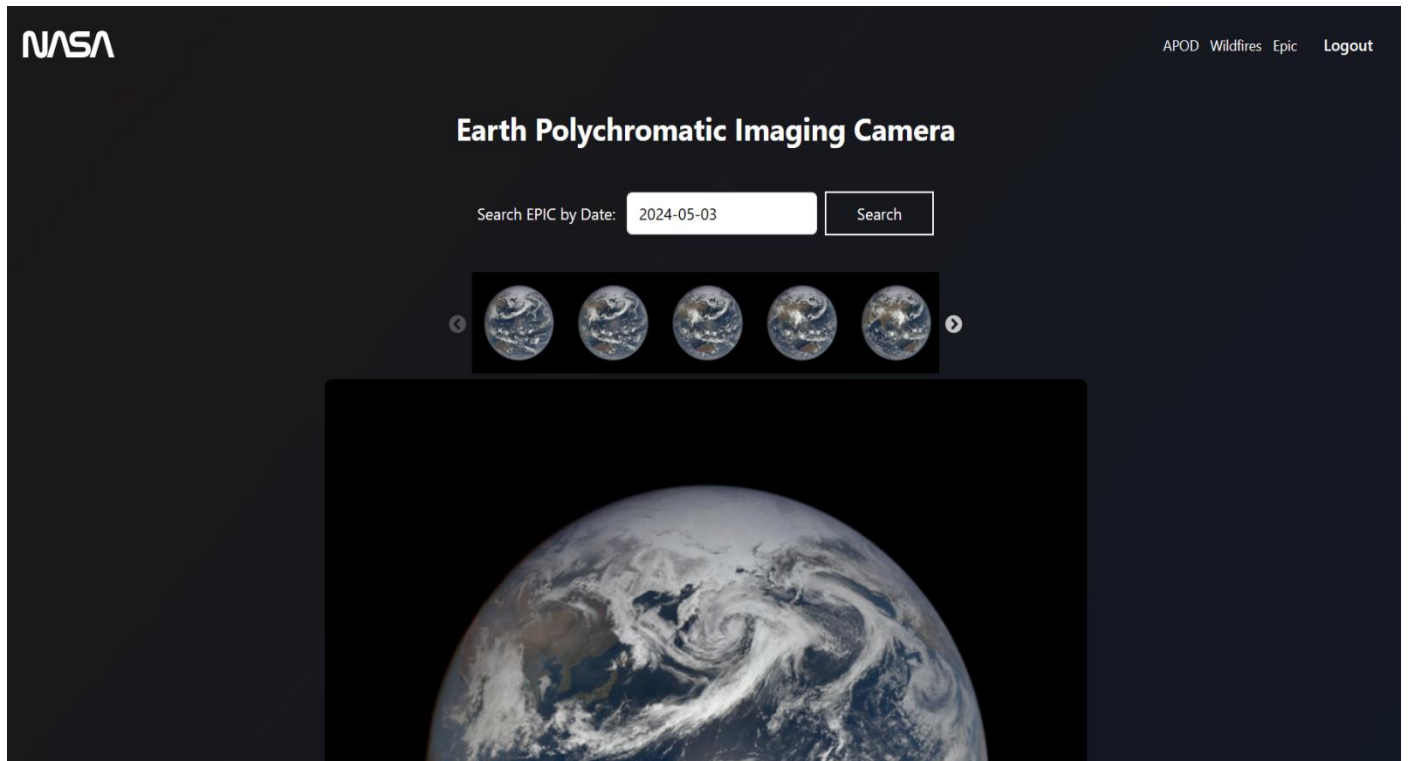


Figure 2: Epic Page

EONET - Earth Observatory Natural Event Tracker (Wildfire)

The NASA Earth Observatory Natural Event Tracker (EONET) API was meticulously chosen for its capacity to furnish real-time and historical data regarding natural phenomena, particularly wildfires, aligning seamlessly with the project's core objective of providing users with timely and insightful information. By harnessing the EONET API, the application offers a comprehensive suite of functionalities, empowering users to track, visualize, and analyze wildfire occurrences with precision and accuracy. EONET's extensive dataset, encompassing vital details such as wildfire locations, dates, and categories, serves as the cornerstone of the application's wildfire monitoring system

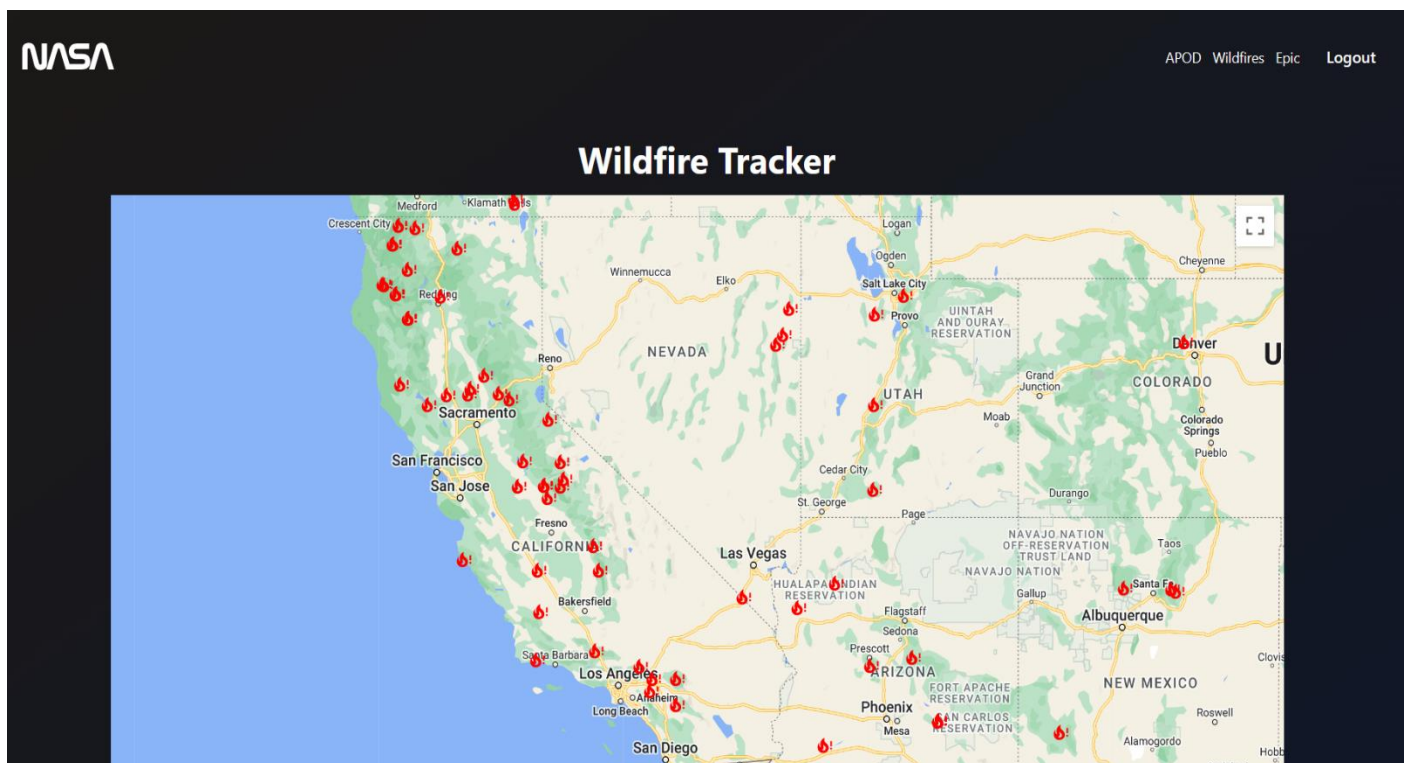


Figure 3: Wildfire Page

2. Challenges Faced and How They Resolved

1. One of the major hurdles I faced in my project was grappling with the complexities of the NASA Open API library and integrating its APIs effectively. Understanding how these APIs functioned and seamlessly incorporating them into my project required significant effort and perseverance. Additionally, I struggled with generating and correctly configuring the API key in the frontend initially. To overcome this challenge, I turned to online resources and stumbled upon a helpful YouTube tutorial by Jonathan Terry. Following his instructions, I successfully integrated the APOD API into my website, overcoming the initial obstacles and laying the groundwork for further development. This experience underscored the importance of perseverance and resourcefulness in navigating challenges during the development process.

URL: <https://www.youtube.com/watch?v=hNUOK9BcSLo&t=226s>

2. Another challenge I faced was figuring out how to use the Google Maps API to implement the wildfire tracking feature. It was tough at first to understand the API and how to integrate it into my project. However, after watching some YouTube tutorials and searching online, I found a solution. By following the instructions in these resources, I successfully added the Google Maps functionality to visualize and track wildfires in the application. This experience showed me the importance of persistence and self-guided learning when tackling technical challenges during development.

URL: <https://www.youtube.com/watch?v=hsNlz7-abd0>

3. Another challenge I encountered was implementing a responsive image slider for browsing through miniature images. It took some trial and error to ensure the slider worked seamlessly across different screen sizes. After watching a YouTube tutorial by a web development channel and utilizing the `react-slick` library, I adjusted the settings based on breakpoints to achieve the desired functionality. This experience demonstrated the value of leveraging online resources and experimenting with different tools to overcome development challenges effectively.

URL: <https://www.youtube.com/watch?v=hZlwvuSXpc>

3. Responsive User Interface

