SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY

SE3040 – Application Frameworks

2024 – Assignment 02

NASA API Frontend Application Documentation– AstroVerse

(IT21252754)

BSc (Hons) in Information Technology Specialized in Software Engineering

Contents

Introduction

This document provides a thorough overview of the features and development process of a front-end application constructed with React functional components. To deliver engaging astronomy-related material to users, the program makes advantage of NASA's public APIs, particularly the Astronomy Picture of the Day (APOD) and Mars Rover Photos APIs. The project, which was completed as part of an assignment, tries to show mastery of frontend programming, hosting platform deployment, API integration, usability improvements using a CSS framework, user session management, version control using Git, and extensive testing.

The application provides users with an enjoyable way to explore the universe by using visually spectacular photography and educational material that is obtained straight from NASA. Utilizing cutting-edge web technologies and industry best practices, the project demonstrates how frontend development can be used to create interactive and instructive user experiences.

Technology Stack

1. **Frontend**: React (with functional components)
2. **Language**: JavaScript
3. **CSS Framework**: Tailwind CSS
4. **Backend**: REST API for user management (Deployed)

    - Login: http://chamaththa.infinitoapparel.ca/api/users/login
    - Register: http://chamaththa.infinitoapparel.ca/api/users/register

5. **Hosting**: Vercel

    https://astro-verse-5apl.vercel.app/

6. **Version Control:** Git (commits made regularly to GitHub)

API Integration

Astronomy Picture of the Day (APOD) API

To retrieve daily astronomy-related data, the application interfaces with NASA's Astronomy Picture of the Day (APOD) API. A selection of excellent photos, films, and educational materials about space travel, celestial phenomena, and astronomical events are available through this API. The application dynamically pulls the most recent astronomy image or video that is available, together with its related description and pertinent metadata, by submitting queries to the APOD API.

Functionality:

- Daily Astronomy Content: Users can view the astronomy picture or video of the day, along with a brief description.
- Date Range Selection: users can specify date range to view historical astronomy content for a particular day.

Implementation:

- **API Requests:** The application makes HTTP requests to the APOD API endpoint, specifying parameters such as the date to retrieve specific content.
- **Data Parsing:** Upon receiving API responses, the application parses the JSON data to extract relevant information, including the image/video URL, description, and other metadata.
- **Dynamic Rendering:** To ensure an interactive and captivating user experience, the fetched data is dynamically rendered within the application's UI components using React state management and component lifecycle methods.

Mars Rover Photos API

The program interfaces with NASA's Mars Rover Photos API in addition to the APOD API to retrieve photos taken by Mars rovers as they explore the Martian surface. With the help of this API, users can browse through an extensive library of images captured by several Mars rovers that highlight the planet's topography, natural characteristics, and scientific findings. The program allows users to remotely explore Mars and see fascinating photographs taken by NASA's robotic explorers by utilizing the Mars Rover Photos API.

Functionality:

- **Mars Rover Images:** Users can browse through a curated selection of images captured by Mars rovers, organized by rover mission and sol (Martian day).

- **Camera Selection:** Users can choose from multiple rover cameras, including Front Hazard Avoidance Camera, Rear Hazard Avoidance Camera, and Mast Camera, to view images specific to each mission.
- **Image Gallery:** The application presents images in a visually appealing gallery format, allowing users to scroll through thumbnails and view selected images.

Implementation:

- **API Requests:** Similar to the APOD API integration, the application makes HTTP requests to the Mars Rover Photos API endpoint, specifying parameters such as the rover’s name, sol (Martian day), and camera type.
- **Data Processing:** Upon receiving API responses containing image metadata, the application processes the data to extract image URLs, camera information, rover information and other relevant details.
- **Image Gallery:** The application dynamically creates an image gallery with thumbnails and allows users to browse full-size images with interactive features like zooming and image details. It does this by utilizing React components and state management techniques.

Functional Requirements

01.User Management

- Signing up as a user:
  Enable users to register for accounts by entering required data, including email address, password, and first and last names.

- Verification of the User:
  Provide users with a login form so they can access their accounts by entering their login information (password and email).
  Mechanisms of Authentication: To confirm users' identities, employ secure authentication methods like OAuth, JSON Web Tokens (JWT), or session-based authentication.

- Management of Sessions:
  Session Tokens: To keep user sessions alive and follow authenticated users through several requests, generate secure session tokens or cookies after successful authentication.

02.APOD Section

- Users can view daily or historical astronomy-related data fetched from the APOD API.
- Users can view today APOD details.

- Data is displayed dynamically based on user interactions, such as date selection for APOD images
- Users can provide date range using date picker and view historical APOD details.

03. Mars Rover Photos Section

- Data is displayed dynamically based on user interactions, such as camera selection for Mars Rover Photos.
- Only logged users can view and interact with mas rover api.(Authentication for personalized features)

Testing

The testing phase of the project involved both unit and integration tests to ensure the reliability and correctness of the application's components and functionality.

Unit Testing

Unit tests were conducted using Jest, a popular JavaScript testing framework, along with React Testing Library. Unit tests focus on testing individual units of code, typically functions or React components, in isolation from the rest of the application. This approach helps identify and fix bugs early in the development process, promoting code quality and maintainability.

Challenges Faced:

1. **Comprehending the format of API Responses and Managing Data Parsing:** Comprehending the format of API responses from NASA's APOD and Mars Rover Photos APIs was one of the first issues that arose. These APIs frequently yield intricate JSON data structures with a variety of arrays and nested objects. It took careful reading of the API documentation and some trial and error with various endpoints and parameters to parse this data successfully and extract the necessary information, including image URLs, titles, descriptions, and dates.

Resolution: Much time was spent reviewing the NASA API documentation in order to get beyond this obstacle. The sample replies were closely examined, and other API endpoints were tested until the data's structure became clear.(Using postman)

2. **Implementing Dynamic Rendering of photographs and Data Based on User Interactions:** Choosing a date for APOD photographs and a rover for Mars Rover photos

were two examples of the challenges encountered when implementing dynamic rendering of images and data based on user interactions. In order to provide a smooth and user-friendly experience, careful planning of component architecture and state management was necessary when designing the application to dynamically fetch and display data based on user input.

Solution: React state management features were used to save and update the user-selected parameters, like the rover selection for Mars Rover Photos and the date for APOD photos, in order to overcome this difficulty.

Setup the project environment

React vite has been used for frontend development. Libraries should be installed using several steps to run the project.

1. Clone the repository.
2. Cd frontend
3. Npm install (Install the libraries)
4. Npm run dev (Run the frontend project)