Space information Website

1. **API:** NASA API
2. **Frontend:** HTML/CSS
3. **Backend:** JavaScript

User Story:

- As a user, I want to search for celestial events on a specific date so that I can view the details and images of significant astronomical phenomena.

Steps:

1. **Set Up the Frontend:**
    - Create an HTML page with a search form that includes a date input field and a search button.
    - Use Bootstrap to ensure the page is responsive and visually appealing.
2. **Design the User Interface:**
    - Implement a layout with Bootstrap classes to create a clean and professional look.
    - Include sections for displaying the celestial event image and details, and ensure they are hidden by default.
3. **Handle User Input:**
    - Add JavaScript to handle the date input and search button click event.
    - On clicking the search button, send a request to the NASA API to fetch data for the specified date.
4. **Fetch and Display Data:**
    - Use the NASA API to retrieve the celestial event data for the given date.
    - If data is available:
        - Display the image, title, and explanation in the designated sections (`#image-section` and `#space-info-section`).
    - If no data is found:
        - Inform the user that no information is available for the selected date.
5. **Implement Loading and Error Handling:**
    - Provide visual feedback to users while data is being fetched, such as loading indicators.
    - Handle API errors gracefully and display user-friendly error messages if the request fails or if the API returns no data.
6. **Ensure Responsiveness:**
    - Use Bootstrap to make sure the application is responsive and looks good on both desktop and mobile devices.

Summary of Steps:

1. **Create HTML Page:**
    - Include a search form with a date input field and a search button.
    - Add sections to display the celestial event image and details.
2. **Style with Bootstrap:**
    - Apply Bootstrap classes to make the application responsive and visually appealing.
3. **Add JavaScript:**
    - Implement functionality to handle user input and fetch data from the NASA API.
    - Show or hide sections based on the search results.
4. **Fetch Data from API:**
    - Retrieve celestial event data based on the selected date.
    - Display the results or provide appropriate feedback if no data is available.
5. **Handle Errors and Loading States:**
    - Implement error handling and loading indicators to enhance the user experience.