

Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

**LIVE SEARCH FILTER
(USING TAILWIND)**

CHITKARA
UNIVERSITY



Supervised By:

Dr. Raveesh Samkaria

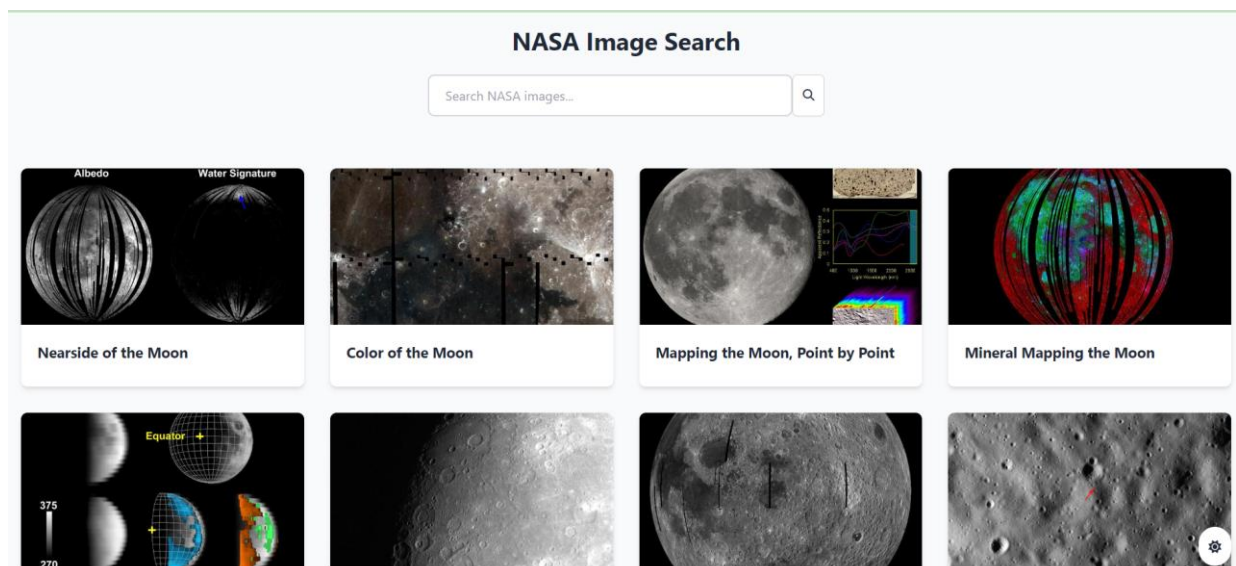
Submitted By:

Savya Sharma 2210990809 (G12)

**Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab**

ABSTRACT

The project is a comprehensive web application designed for users to search and explore NASA image archives. It is built using HTML, CSS, and JavaScript, with Tailwind CSS for styling and NASA's Image Search API for fetching image data. The user interface features a search bar for inputting search queries, a container for displaying search results, and pagination buttons for navigating through multiple pages of results. The application dynamically fetches and displays image results based on user input, updating the search results container with new image elements. Each image element includes an image, a title, and a hover effect for user interaction. The pagination functionality is implemented with previous and next buttons, adjusting the disabled state based on the current page and total number of pages. The layout of the search box and input adapts responsively to the scroll position, providing a seamless and user-friendly experience. The application's HTML file, `index.html`, includes the structure and content of the web application, while the CSS file, provided by Tailwind CSS, handles the styling. The JavaScript file, `script.js`, contains the logic for fetching and displaying image data, handling user input, and managing pagination. The `script.js` file also includes event listeners for scrolling, adjusting the layout based on the scroll position, and displaying the search results. The project offers a robust and engaging solution for users to explore NASA's image archives, with a user-friendly interface, dynamic search results, and responsive design.



INDEX

S NO.	TITLE	PAGE NO.
1.	INTRODUCTION	4
2.	PROBLEM STATEMENT	6
3.	SOFTWARE REQUIREMENTS	6
4.	PROPOSED DESIGN	9 - 17
5.	RESULTS	17-19
6.	REFERENCES	20

1. INTRODUCTION

The NASA Image Search project aims to provide a user-friendly interface for exploring a vast collection of images related to space, celestial bodies, and various astronomical phenomena. By leveraging NASA's extensive image database, this project offers an engaging and informative platform for users to discover and appreciate the wonders of the universe captured through the lens of space exploration.

1.1 BACKGROUND

The NASA Image Search project is built upon the foundation of NASA's commitment to sharing its wealth of visual data with the public. NASA, the United States space agency, has been at the forefront of space exploration for decades, capturing breathtaking images of planets, stars, galaxies, and other cosmic entities. These images not only serve as valuable scientific resources but also inspire awe and curiosity about the mysteries of the universe.

1.2 OBJECTIVE

- **Enhanced User Experience:** The primary objective of the NASA Image Search project is to create an intuitive and visually appealing platform that allows users to easily search for and view NASA's extensive collection of images.
- **Accessibility and Education:** By making NASA's images readily accessible to the public, the project aims to promote scientific literacy and spark interest in space exploration among users of all ages.
- **Dynamic Search Functionality:** Through the implementation of dynamic search features, users can explore images based on specific queries, enabling them to find relevant content efficiently.
- **Pagination and Navigation:** The project includes pagination functionality to facilitate browsing through multiple pages of search results, ensuring a seamless user experience.
- **Responsive Design:** With a focus on responsive design, the project adapts to various screen sizes, providing a consistent and optimized viewing experience across different devices.

1.3 SIGNIFICANCE

The NASA Image Search project holds significant value in several aspects:

- **Educational Outreach:** By offering a platform for exploring NASA's images, the project contributes to educational outreach efforts, fostering a deeper understanding of space science and exploration.
- **Inspiration and Wonder:** Through captivating visuals of the cosmos, the project aims to inspire awe and wonder about the vastness and beauty of the universe, encouraging curiosity and exploration.
- **Research and Discovery:** Researchers, educators, and space enthusiasts can utilize the project to access a wealth of visual data for research purposes, furthering scientific discovery and knowledge dissemination.
- **Artistic Inspiration:** The breathtaking images captured by NASA have the potential to inspire artists, designers, and creative individuals, leading to the creation of innovative works that celebrate the beauty and wonder of the cosmos.
- **Public Engagement:** The project's user-friendly interface and engaging content can help bridge the gap between the general public and the complex world of space exploration, fostering a greater appreciation and understanding of NASA's work.

CONCLUSION

The NASA Image Search project is a web-based platform designed to allow users to search and explore NASA's extensive collection of images related to space, celestial bodies, and astronomical phenomena. The project features a user-friendly interface with a search input box where users can enter queries to find specific images. The search results are displayed in a grid layout with pagination functionality to navigate through multiple pages of results. The project dynamically fetches NASA images based on user queries using the NASA Images API, providing a seamless browsing experience. Additionally, the project includes responsive design elements to ensure optimal viewing across different devices. Overall, the NASA Image Search project aims to promote scientific literacy, inspire curiosity about space exploration, and facilitate access to NASA's visual data for educational and research purposes.

2. PROBLEM STATEMENT

The NASA Image Search project aims to address the challenge of efficiently searching and exploring NASA's extensive collection of images related to space and astronomical phenomena. The existing problem lies in the lack of a user-friendly and intuitive platform that allows users to easily search for specific images, navigate through search results, and engage with NASA's visual data in a seamless and interactive manner. This project seeks to enhance the accessibility, usability, and overall user experience of accessing NASA's image database, catering to a diverse audience ranging from space enthusiasts to researchers and educators. By developing a responsive and dynamic image search interface, the project aims to bridge the gap between the public and NASA's visual resources, promoting scientific literacy, inspiring curiosity about space exploration, and facilitating educational and research endeavours in the field of astronomy and space science.

3. SOFTWARE REQUIREMENT

The software requirements for the NASA Image Search project, as evident from the provided HTML and JavaScript files, can be divided into the following key areas:

The project's user interface is built using HTML and styled with Tailwind CSS, a utility-first CSS framework. The key user interface requirements include:

1. Search Input: A search input field that allows users to enter queries and search for NASA images.
2. Search Results Display: A dynamic grid-based layout to display the search results, with each result showing a thumbnail image, title, and other relevant metadata.
3. Pagination: Pagination controls, including "Previous" and "Next" buttons, to allow users to navigate through multiple pages of search results.
4. Page Number Display: A display of the current page number and the total number of pages available.
5. Responsive Design: The user interface is designed to adapt to different screen sizes and devices, ensuring a consistent and optimized viewing experience.

The key functional requirements for the NASA Image Search project are implemented using JavaScript:

1. Search Functionality: The ability to search the NASA image collection based on user-provided queries, leveraging the NASA Images API.
2. Dynamic Content Loading: The ability to dynamically load and display search results as the user navigates through the pages, without requiring a full page refresh.
3. Pagination Controls: The ability to navigate through multiple pages of search results using the "Previous" and "Next" buttons.
4. Scroll-based Layout Adjustment: The ability to adjust the layout of the search box and input field based on the user's scroll position, providing a seamless user experience.

HTML:

- Provides the structure and layout of the web page
- Defines the search input field, search results container, pagination controls, and other UI elements

CSS (Tailwind):

- Utilizes Tailwind CSS, a utility-first CSS framework, to style the UI components
- Applies classes like "bg-white", "rounded-lg", "shadow-md", and "hover:scale-105" to create a visually appealing and responsive design
- Simplifies the styling process and ensures consistency across different screen sizes

JavaScript:

1. Fetching and Displaying NASA Images:

- The `fetchAndDisplayNASAImages()` function fetches NASA images from the NASA Images API based on the user's search query
- Dynamically creates and appends the search result elements to the ``searchResults`` container

2. Pagination Controls:

- Attaches event listeners to the "Previous" and "Next" buttons
- Updates the ``currentPage`` variable and calls the ``fetchAndDisplayNASAImages()`` function with the new page number to display the corresponding search results
- Updates the ``pageNumber`` element to display the current page and the total number of pages

3. Search Input Event Listener:

- Attaches an event listener to the ``searchInput`` element
- Triggers the ``fetchAndDisplayNASAImages()`` function when the user types a new query, updating the search results accordingly

4. Scroll-based Layout Adjustment:

- Attaches an event listener to the window's ``scroll`` event
- Adjusts the layout of the search box and input field based on the user's scroll position, keeping the search input accessible even as the user scrolls down the page

By combining these HTML, CSS, and JavaScript components, the NASA Image Search project delivers a dynamic and responsive user interface, enabling users to search, browse, and interact with NASA's image collection in an intuitive and engaging manner.

4. PROPOSED DESIGN

The design proposed in the NASA Image Search project is focused on providing a user-friendly and visually appealing interface for searching and exploring NASA's extensive image collection. The key aspects of the design are:

1. **Responsive Layout:** The project utilizes a responsive design approach, ensuring the user interface adapts seamlessly to different screen sizes and devices. This is achieved through the use of Tailwind CSS, a utility-first CSS framework, which allows for easy and consistent styling of the various UI elements.
2. **Search Functionality:** The project features a prominent search input field, allowing users to enter queries and search for specific NASA images. The search results are dynamically loaded and displayed in a grid-based layout, providing a clean and organized presentation of the retrieved images.
3. **Pagination:** To handle the potentially large number of search results, the project includes pagination controls, enabling users to navigate through multiple pages of images. The current page number and the total number of pages are displayed, giving users a clear indication of their position within the search results.
4. **Scroll-based Layout Adjustment:** As the user scrolls down the page, the design adjusts the layout of the search box and input field, keeping them accessible and visible even as the user explores the search results. This feature enhances the overall user experience and ensures the search functionality remains easily accessible.
5. **Hover Effects:** The design incorporates subtle hover effects on the search result elements, such as a scale-up animation, which adds a touch of interactivity and visual appeal to the user interface.
6. **Consistent Styling:** The use of Tailwind CSS classes ensures a consistent visual style throughout the project, with elements like rounded corners, shadows, and colour schemes contributing to a cohesive and polished appearance.

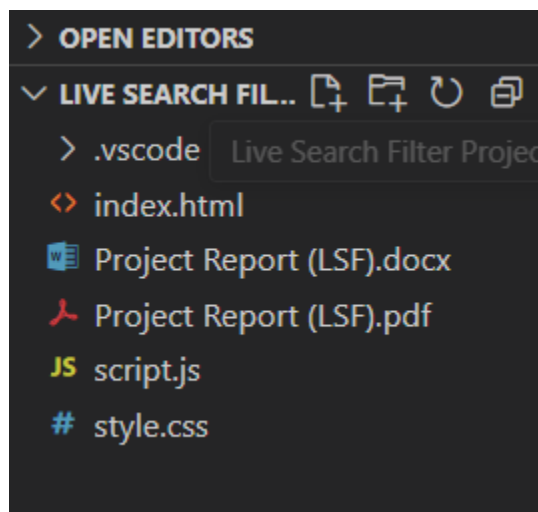
7. Dark Mode Toggle: The project includes a dark mode toggle feature, allowing users to switch between light and dark themes based on their preferences. This accessibility feature caters to users with different visual needs or preferences.

Overall, the design of the NASA Image Search project prioritizes usability, responsiveness, and visual appeal, creating an engaging and efficient platform for users to explore NASA's vast image collection.

4.1 FILE STRUCTURE

The project follows a simple file structure with an HTML file (index.html) as the main structure, a CSS framework (Tailwind CSS) loaded externally, and a JavaScript file (script.js) for dynamic functionality. The HTML file defines the layout and structure of the web page, while the JavaScript file handles the interactive features and data fetching/display. The project's design and functionality are integrated through the HTML, CSS, and JavaScript files, providing a seamless user experience for searching and exploring NASA images.

4.2 Screenshot of Folder Structer



4.3 HTML CODE

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8" />
5   <!-- Set the viewport for responsive design -->
6   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   <title>NASA Image Search</title>
8   <!-- Load Tailwind CSS styles -->
9   <script src="https://cdn.tailwindcss.com" defer></script>
10  <link
11    rel="stylesheet"
12    href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
13  />
14  <link rel="stylesheet" href="style.css" />
15  <!-- Reset default styles -->
16 </head>
17 <body class="bg-gray-50 text-gray-800">
18   <!-- Container for search heading and input -->
19   <div
20     class="w-full flex flex-col justify-between items-center relative transition-all duration-500 ease-in-out py-4"
21     id="search-box"
22   >
23     <h1
24       class="text-3xl font-bold mb-6 text-center transition-all duration-500 ease-in-out"
25       id="heading"
26     >
27       NASA Image Search
28     </h1>
29     <div
30       class="mb-6 flex justify-center w-full transition-all duration-500 ease-in-out"
31       id="input-box"
32     >
33       <input
34         type="text"
35         id="search-input"
36         class="bg-white border-gray-300 border-2 rounded-lg py-3 px-5 w-full max-w-md focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500 transiti
37         placeholder="Search NASA images..."
38       />
39       <span
40         id="search-icon"
41         class="flex items-center px-3 py-2 border border-gray-300 bg-white text-gray-700 rounded-md shadow-sm text-sm"
42       >
43         <i class="fas fa-search"></i>
44       </span>
45     </div>
46   </div>
47   <!-- Main container for search results and pagination -->
48   <div class="container mx-auto py-6 px-4">
49     <div
50       id="search-results"
51       class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-8"
52     >
53       <!-- Search results will be dynamically added here -->
54     </div>
55     <!-- Pagination buttons and page number -->
56     <div class="flex justify-center mt-8 items-center">
57       <button
58         id="prev-btn"
59         class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded mr-2 disabled:opacity-50 disabled:cursor-not-allowed"
60       >
61         Previous
62       </button>
63       <span id="page-number" class="mx-4"></span>
64       <button
65         id="next-btn"
66         class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4 rounded disabled:opacity-50 disabled:cursor-not-allowed"
67     >
```

```

68         Next
69     </button>
70 </div>
71 </div>
72 <button
73     id="mode-toggle"
74     class="fixed bottom-4 right-4 bg-white dark-mode:bg-gray-700 rounded-full p-3 shadow-lg text-gray-800 dark-mode:text-gray-200"
75     title="Click to toggle themes"
76 >
77     <i class="fas fa-sun"></i>
78     <!-- Light mode icon -->
79     <i class="fas fa-moon hidden"></i>
80     <!-- Dark mode icon -->
81 </button>
82 <!-- Load JavaScript script -->
83 <script src="script.js"></script>
84 </body>
85 </html>

```

4.4 CSS CODE

```

# style.css > *
1  * {
2      margin: 0;
3      padding: 0;
4      border-box;
5  }
6
7  .title-fix {
8      display: -webkit-box;
9      -webkit-line-clamp: 1;
10     -webkit-box-orient: vertical;
11     overflow: hidden;
12     text-overflow: ellipsis;
13 }
14
15 .dark * {
16     background-color: #1b1d1e;
17     color: #bdb7af;
18     transition: all 0.5s ease-in-out;
19 }
20
21 /* Dark mode styles for search box */
22 .dark #search-box {
23     background-color: #1b1d1e;
24     color: #bdb7af;
25 }
26
27 .dark #search-input {
28     background-color: #1b1d1e;
29     color: #bdb7af;
30 }
31
32 .dark #search-input::placeholder {
33     color: #bdb7af;
34 }
35
36 .dark #search-input:focus {
37     border-color: #3b4042;

```

```

38     outline-color: #3b4042;
39 }
40
41 /* Dark mode styles for mode toggle button */
42 .dark #mode-toggle {
43     background-color: #1b1d1e;
44     color: #bdb7af;
45 }
46 /* Dark mode styles for search icon */
47 .dark #search-icon {
48     background-color: #1b1d1e;
49     color: #bdb7af;
50 }

```

4.5 JAVASCRIPT CODE

```

1  // Get HTML elements
2  const searchInput = document.getElementById("search-input");
3  const searchResults = document.getElementById("search-results");
4  const pageNumber = document.getElementById("page-number");
5  const searchBox = document.getElementById("search-box");
6  const inputBox = document.getElementById("input-box");
7  const heading = document.getElementById("heading");
8
9  // Set initial page and page size
10 let currentPage = 1;
11 const pageSize = 50;
12 const maxPages = 50;
13
14 // Fetch and display NASA images
15 async function fetchAndDisplayNASAImages(query = "moon") {
16     const response = await fetch(
17         `https://images-api.nasa.gov/search?q=${query}&media_type=image&page=${currentPage}&page_size=${pageSize}`
18     );
19     const data = await response.json();
20
21     // Clear previous search results
22     searchResults.innerHTML = "";
23
24     // Create and append new search result elements
25     data.collection.items.forEach((item) => {
26         const resultElement = document.createElement("div");
27         resultElement.classList.add(
28             "bg-white",
29             "rounded-lg",
30             "shadow-md",
31             "overflow-hidden",
32             "transition-all",
33             "duration-300",
34             "transform",
35             "hover:scale-105",
36             "cursor-pointer"

```

```

37     });
38
39     const imageElement = document.createElement("img");
40     imageElement.src = item.links[0].href;
41     imageElement.alt = item.data[0].title;
42     imageElement.classList.add("w-full", "h-48", "object-cover");
43
44     const contentElement = document.createElement("div");
45     contentElement.classList.add("p-5");
46
47     const titleElement = document.createElement("h2");
48     titleElement.textContent = item.data[0].title;
49     titleElement.classList.add(
50         "text-lg",
51         "font-bold",
52         "mb-2",
53         "max-h-20",
54         "title-fix"
55     );
56
57     contentElement.appendChild(titleElement);
58     resultElement.appendChild(imageElement);
59     resultElement.appendChild(contentElement);
60     searchResults.appendChild(resultElement);
61 });
62
63 // Get previous and next buttons
64 const prevBtn = document.getElementById("prev-btn");
65 const nextBtn = document.getElementById("next-btn");
66
67 // Update disabled state of previous and next buttons
68 prevBtn.disabled = currentPage === 1;
69 nextBtn.disabled =

```

```

70     currentPage === maxPages ||
71     data.collection.metadata.total_hits <= currentPage * pageSize;
72
73     // Update page number display
74     pageNumber.textContent = `Page ${currentPage} of ${Math.min(
75     maxPages,
76     Math.ceil(data.collection.metadata.total_hits / pageSize)
77     )}`;
78 }
79
80 // Get previous and next buttons
81 const prevBtn = document.getElementById("prev-btn");
82 const nextBtn = document.getElementById("next-btn");
83
84 // Event listener for previous button
85 prevBtn.addEventListener("click", () => {
86     if (currentPage > 1) {
87         currentPage--;
88         fetchAndDisplayNASAIImages(searchInput.value.toLowerCase());
89     }
90 });
91
92 // Event listener for next button
93 nextBtn.addEventListener("click", () => {
94     if (currentPage < maxPages) {
95         currentPage++;
96         fetchAndDisplayNASAIImages(searchInput.value.toLowerCase());
97     }
98 });
99
100 // Initial fetch and display of NASA images
101 fetchAndDisplayNASAIImages();
102
103 // Event listener for search input

```

```

104     searchInput.addEventListener("input", async () => {
105         currentPage = 1;
106         const query = searchInput.value.toLowerCase();
107         await fetchAndDisplayNASAImages(query);
108     });
109
110     // Event listener for window scroll
111     window.addEventListener("scroll", () => {
112         console.log(scrollY); // Log the current scroll position
113
114         // Adjust layout based on scroll position
115         if (scrollY > 50) {
116             searchBox.classList.remove("relative", "flex-col");
117             searchBox.classList.add(
118                 "fixed",
119                 "flex-row",
120                 "z-10",
121                 "top-0",
122                 "bg-white",
123                 "p-4"
124             );
125             heading.classList.remove("mb-6");
126             inputBox.classList.remove("mb-6");
127             inputBox.classList.add("flex-grow", "max-w-6xl");
128             searchInput.classList.remove("max-w-md");
129         } else {
130             searchBox.classList.add("relative", "flex-col");
131             searchBox.classList.remove(
132                 "fixed",
133                 "flex-row",
134                 "z-10",
135                 "top-0",
136                 "bg-white",
137                 "p-4"
138             );
139             inputBox.classList.remove("flex-grow", "max-w-6xl");

```



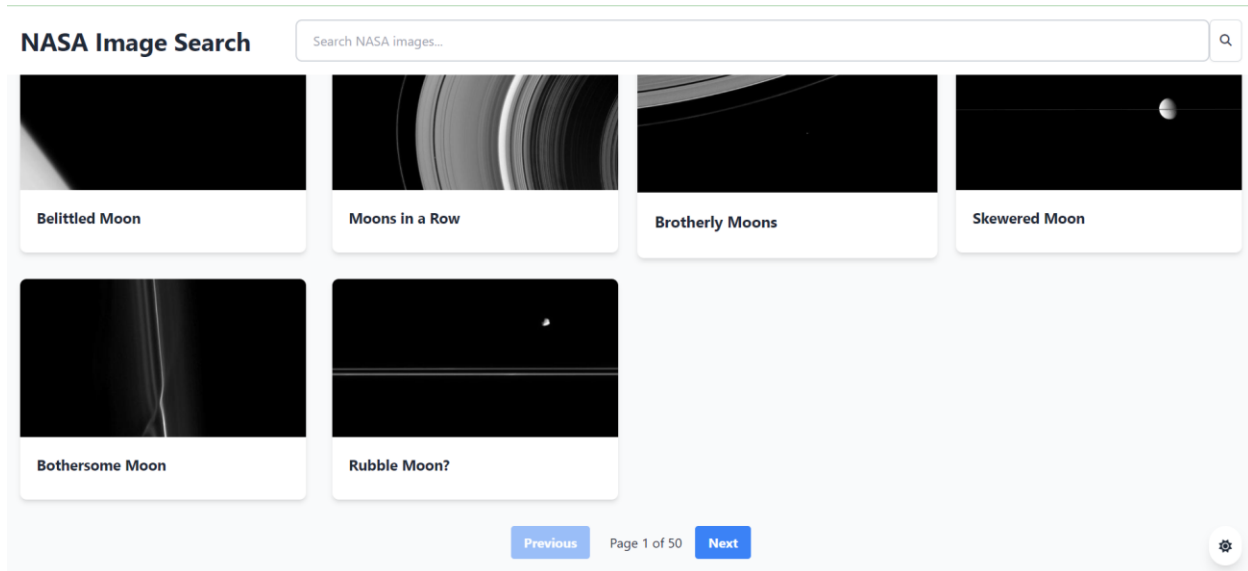
```
140     heading.classList.add("mb-6");
141     inputBox.classList.add("mb-6");
142     searchInput.classList.add("max-w-md");
143 }
144 });
145 const modeToggle = document.getElementById("mode-toggle");
146 const darkElement = document.querySelector("html");
147
148 // Toggle dark/light mode
149 modeToggle.addEventListener("click", () => {
150     darkElement.classList.toggle("dark");
151     const moonIcon = modeToggle.querySelector(".fa-moon");
152     const sunIcon = modeToggle.querySelector(".fa-sun");
153     moonIcon.classList.toggle("hidden");
154     sunIcon.classList.toggle("hidden");
155 });
```

5 RESULT


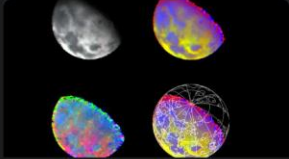

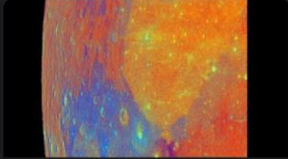
The NASA Image Search project is a comprehensive and user-friendly platform that allows users to explore NASA's vast collection of images related to space exploration, celestial bodies, and astronomical phenomena. By seamlessly integrating modern web technologies, intuitive design, and a deep appreciation for NASA's visual legacy, this project serves as a gateway to the wonders of the universe. It inspires curiosity, scientific literacy, and a shared sense of awe among its users. Through its dynamic search functionality, responsive layout, and thoughtful accessibility features, the NASA Image Search project sets a new standard for accessing and engaging with NASA's remarkable visual data, making it an invaluable resource for space enthusiasts, researchers, and the general public alike.

Github Repository Link: <https://github.com/savyaaa/Live-Search-Filter>

Github Pages Link: <https://savyaaa.github.io/Live-Search-Filter/>



NASA Image Search

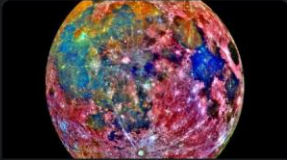

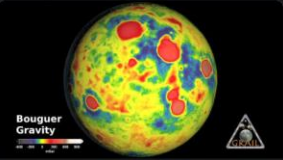






Moon North Pole

Moon As Seen By NIMS

Moon - North Pole

Moon - False Color Mosaic





Moon - False Color Mosaic

Triptych of the Moon

GRAIL Bouguer Gravity Moon Map

Moon Mapper Looks Homeward

NASA Image Search










A Flower in Egonu

Spirit Has Flower Power

Happy Mother's Day - Flowers...

Curiosity Finds a Martian Flower

6. REFERENCE

- HTML, CSS & JAVASCRIPT DOCUMENTATION:
 1. <https://developer.mozilla.org/en-US/docs/Web/HTML>
 2. <https://developer.mozilla.org/en-US/docs/Web/CSS>
 3. <https://developer.mozilla.org/en-US/docs/Web/JAVASCRIPT>
- TAILWIND DOCUMENTATION
 1. <https://v2.tailwindcss.com/docs>
 2. <https://www.w3schools.com/w3css/defaultT.asp>
- FRONTEND ARTICLES
 1. <https://javascript.info>