Project Based Learning (PBL)

OF

(Advanced Java)

CSP0409

Submitted To:-

Mr. Neeraj Goyal

Associate Professor Dept.

of CSA/SOET

Submitted By:-

Amaan Haque

(BETN1CS22100)

B.Tech 4th SEM

(Data Science)

Project on Topic:

Search Engine (Image)

Introduction:

Welcome to my Project-Based Learning (PBL) adventure focusing on the development of an "Image Search Engine" using HTML, CSS, and JavaScript. In today's digital age, where visual content dominates the online landscape, creating an efficient and user-friendly image search tool is not only relevant but also a rewarding endeavor.

Throughout this PBL journey, we will dive into the fundamentals of web development, leveraging HTML for structuring our web pages, CSS for styling them, and JavaScript for adding interactivity and functionality. Our goal is to combine these technologies to construct a dynamic and responsive image search engine that enables users to explore and discover visual content effortlessly.

By embarking on this project, we will not only enhance our skills in web development but also gain valuable insights into the complexities of information retrieval and user interface design. We will learn to manipulate and display images, implement search algorithms, and optimize the performance of our application to ensure a seamless user experience.

Furthermore, this project will foster collaboration, critical thinking, and problem-solving skills as we work together to overcome challenges and iterate on our ideas. Through hands-on experimentation and experimentation,

We will refine our understanding of web technologies and cultivate a mindset of continuous learning and improvement.

Join me on this exciting journey as we embark on the creation of an innovative image search engine that combines the power of HTML, CSS, and JavaScript to redefine how users can interact with visual content on the web. Let's explore the project and push the boundaries of what's possible in web development!

Working:

Let's walk through how the "Image Search Engine" project would work without diving into the actual code.

1. **HTML Structure (index.html):**

   - The **index.html** file defines the structure of the webpage.
   - It contains a header section with a title "Image Search Engine", an input field for users to enter their search queries, and a search button to trigger the search action.
   - Below the header, there's three div sections (one **<search-results> and then two <search-result>**) with image sources (src) and their links if wanted (**using **) and finally the show more button for generating more images if the user want them to be displayed.
   - The HTML file also includes references to the CSS file (**style.css**) **[<link rel="stylesheet" href="style.css">]** for styling and the JavaScript file (**script.js**) for interactivity.

2. **Styling (style.css):**

   - The **style.css** file defines the visual appearance of the webpage elements.
   - It sets the font family, background colors, padding, margins, and other visual properties to create a visually appealing layout.
   - Styles are applied to elements such as the header, search-input, search-button, search-result, show-more-button and individual images to ensure consistency and a pleasing aesthetic including some hover effects.

3. **JavaScript Functionality (script.js):**

   - The **script.js** file adds interactivity to the webpage.
   - It handles user interactions, such as clicking the search button.
   - When the search button is clicked, the JavaScript code retrieves the value entered in the search input field.

- It then triggers a function (**searchImages()**) to perform the search operation based on the user's query.

- Inside the **searchImages()** async function, we have implement the logic to search for images. This involves querying an external API **(Unspalsh.com)**, searching a local , or processing data stored within the application.

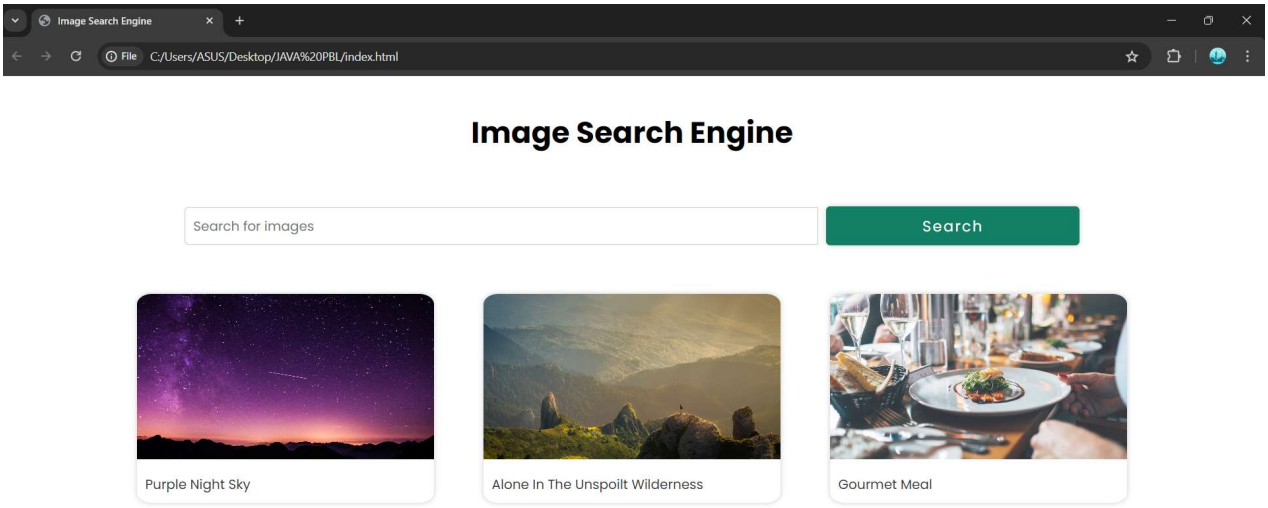
- For demonstration purposes, let's assume the **searchImages()** async function retrieves a list of image URLs based on the user's query.

- By using async function **searchImages() ,** we are getting the response from the user (i.e., the search-input) and then our script.js **fetches** the urls from the API and give the urls as the (search-results) output to the users.

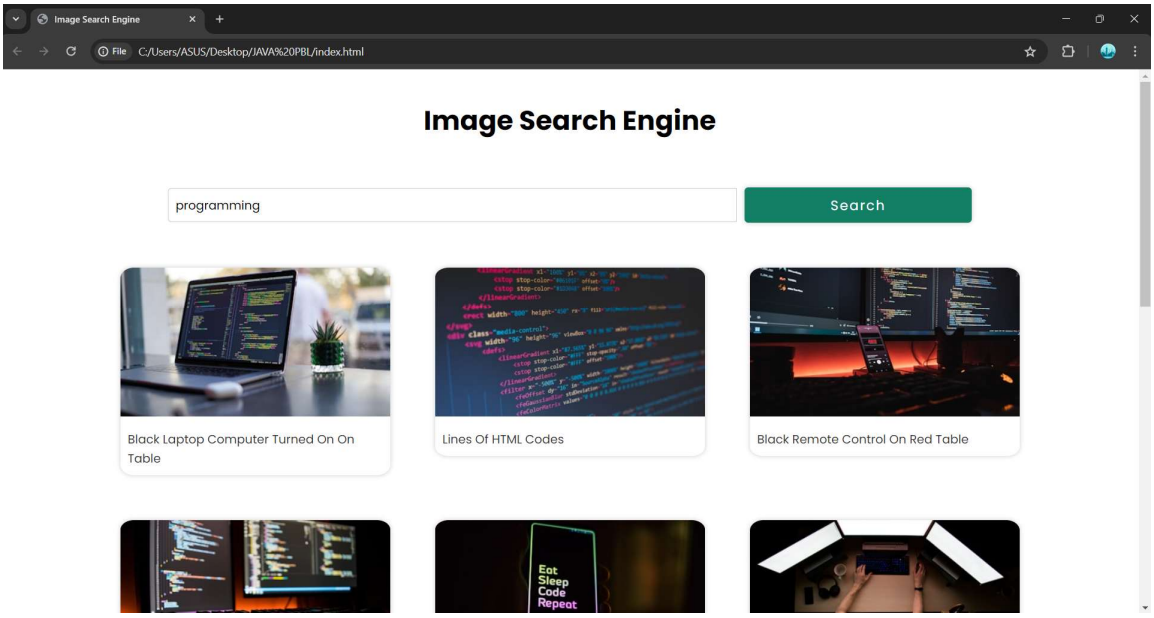
- As a result, when the search button is clicked, the webpage updates to display the search results (images) relevant to the user's query.

Outputs / Results:

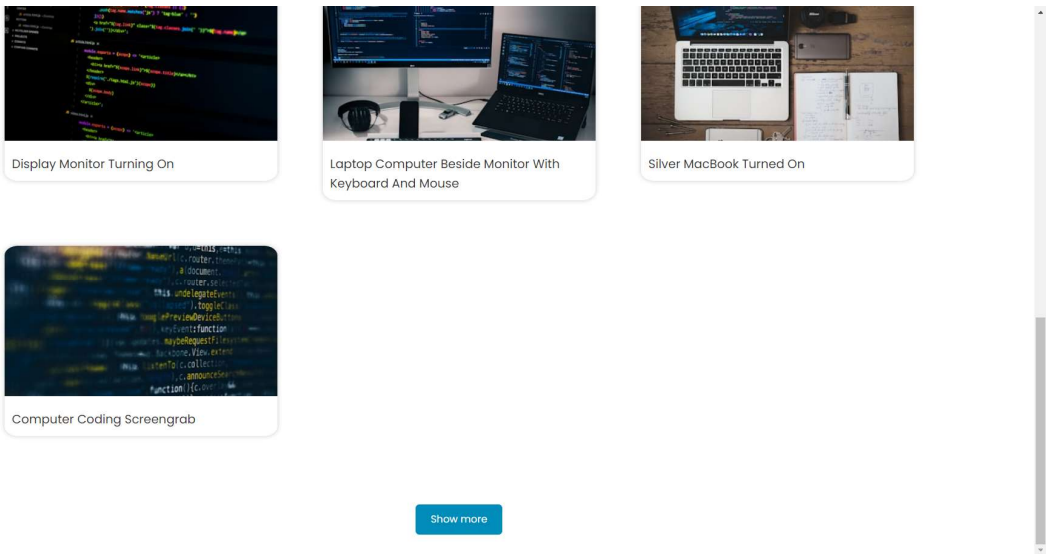
➢ **Main Interface -**



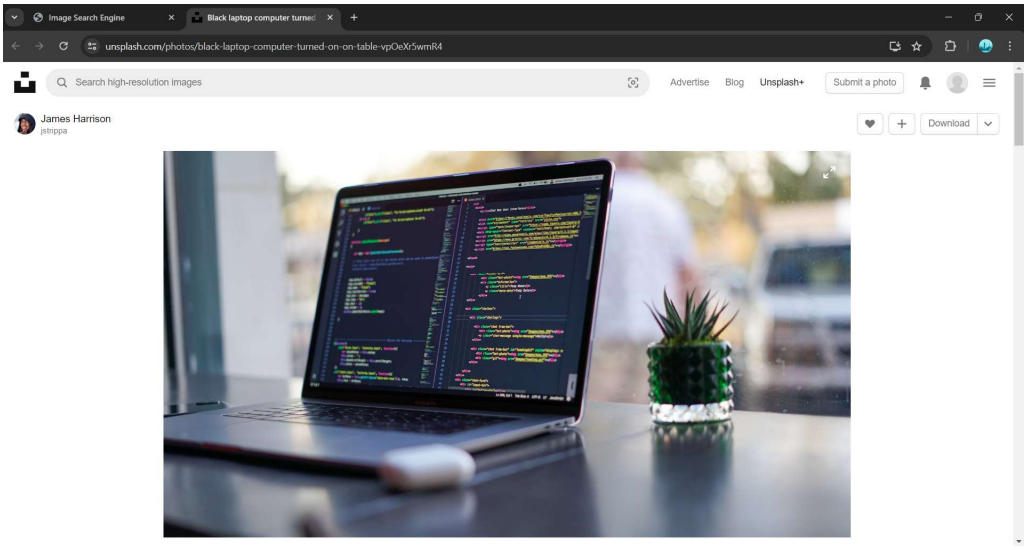
- **Result when the user searches for anything -**



- **Here we can see the "Show more" Button at the bottom of webpage –**



- **User gets Redirected to the main link (when clicking on the text written below the images) from the image is taken from -**



Conclusion:

In conclusion, my journey in developing the "Image Search Engine" project using HTML, CSS, and JavaScript has been both rewarding and enlightening. Through the guidance and support of my esteemed teacher and my friends , I've delved deep into the intricacies of web development, gaining invaluable skills and insights along the way.

I am immensely grateful to Neeraj Sir for his expertise, patience, and dedication in nurturing our learning experience. His mentorship has been instrumental in shaping our understanding of web technologies and fostering a spirit of curiosity and exploration.

With gratitude,

Amaan Haque