**Mentor Project**

**A Mentor Project Report**

**Submitted in Partial fulfillment for the award of**

**Bachelor of Technology**

**In**

**Computer Science & Engineering**

Submitted to

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA**

**BHOPAL (M.P)**



**MENTOR PROJECT- REPORT**

Submitted by

Harsh Muralia [0103CS191087]



**Department of Computer Science & Engineering**

**Lakshmi Narain College of Technology, Bhopal (M.P.)**

**Session**

**Nov. 2021**

LAKSHMI NARAIN COLLEGE OF TECHNOLOGY, BHOPAL

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



### CERTIFICATE

This is to certify that the work embodied in this project work entitled **“Music Player”** has been satisfactorily completed by the **Harsh Muralia** (0103CS191087). It is a bonafide piece of work, carried out under the guidance in **Department of Computer Science & Engineering**, **Lakshmi Narain College of Technology, Bhopal** for the partial fulfillment of the **Bachelor of Technology** during the academic year 2021-2022.

Approved By

**Professor & Head**

# Department of Computer Science & Engineering

LAKSHMI NARAIN COLLEGE OF TECHNOLOGY, BHOPAL

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Acknowledgement**

We express our deep sense of gratitude to department of Computer Science & Engineering L.N.C.T., Bhopal. Whose kindness valuable guidance and timely help encouraged me to complete this project.

A special thank goes to Dr. Sadhna K. Mishra (HOD) who helped me in completing this project work. He exchanged his interesting ideas & thoughts which made this project work successful.

I would also thank our institution and all the faculty members without whom this project work would have been a distant reality.

**Signature**

Harsh Muralia

0103CS191087

**INTRODUCTION**

Search Engine technology was born about the same time as the World Wide Web, and has improved significantly over the past decade and has become an integral part of everyone's web browsing experience, especially after the amazing success of Google.

At first glance, the search engines seem to have been well-read, and many articles and theories including the founders' page of Google [Brin 1998] have been published to explain and analyze their internal methods. However, in this report the author will point out that there are many unresolved issues in Search Engines and unspecified usage details, due to the immaturity of this technology or just the business environment of their owners.

The start is the Web Crawler (or spider) for retrieving all the Web pages: simply cutting off the entire Web or its own small set, to download the pages or files it encounters and save some of the components it will use. The actual cross-cutting algorithm varies depending on the application; depth first, width first, or random cut are all used to meet different design goals.

Simple or complex, a search engine should have a way of deciding which pages are the most important, and presenting them to users in a certain way. This is called the Ranking System. The most popular is the Page Rank Algorithm published by Google's founders [Brin 1998].

In addition to well-known commercial products, such as Google2, Yahoo3 and MSN4, there are open source search engines, for example, ASPSeek5, BBDBot6, Datapark Search7, and ht: // Dig8. Assessing their benefits and risks is not the purpose of this thesis, but based on reviews and responses from other people [Morgan 2004], they are unique only in one area, or they do not use high-quality algorithms, or have not been stored somewhere. for a long time.

**PROBLEM STATEMENT AND OBJECTIVES**

There are times faced by people when they want to search something specifically on wikipedia directly and are not interested to see the recent news or myths about their search query.

Here comes in picture a need of an easy to use software which can directly show the search results of their query directly from wikipedia in an instant.

Nobody likes to watch ads and recent news which are usually very off topic and myths. This app is for people who want to directly search wikipedia results which this app can do it easily.

**PROPOSED DESIGN AND METHODOLOGY**

Attempting to solve these two problems and inspired by previous research, I designed while providing the extra feature of direct search.

It is not my goal to produce an identical mathematical match to the original Page, which will be used as comparison baseline nevertheless, but to provide reasonable importance measurements of the URLs as well as a grouping feature that the end user can enjoy.

Keeping the goals in mind, I have taken various intuitive shortcuts and heuristic approaches without mathematical proofs.

**HARDWARE AND SOFTWARE USED**

According to my project requirements, I have used HTML5 for structuring our website, For styling I used CSS3, BootStrap 4 and for interactivity, I used Vanilla Javascript.

At last concluding this up, with the help of Github I have hosted my project app to make searching song lyrics easier for users in their day to day life.

**IMPLEMENTATION**

**INDEX.html**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css" integrity="sha512-+4zCK9k+qNFUR5X+cKL9EIR+ZOhtIloNl9GIKS57V1MyNsYpYcUrUeQc9vNfzsWfV28IaLL3i96P9sdNyeRssA==" crossorigin="anonymous" />**

**<link rel="preconnect" href="https://fonts.gstatic.com">**

**<link href="https://fonts.googleapis.com/css2?family=Roboto&display=swap" rel="stylesheet">**

**<link rel="stylesheet" href="dist/css/style.min.css" />**

**<script src="./js/main.js" type="module"></script>**

**<title>SearchMe</title>**

**</head>**

**<body>**

**<main>**

**<h1 class="offscreen">Search Me A Wikipedia Powered Search Engine</h1>**

**<section class="searchEntry">**

**<h2 class="offscreen">Search Term Entry</h2>**

**<div class="logo" aria-hidden="true">**

**<span class="blue">S</span>**

**<span class="red">e</span>**

**<span class="yellow">a</span>**

**<span class="blue">r</span>**

**<span class="green">c</span>**

**<span class="red">h</span>**

**<span class="blue">M</span>**

**<span class="yellow">e</span>**

**<span class="red exclaim">!</span>**

**</div>**

**<form id="searchBar" class="searchBar" role="search">**

**<label class="offscreen" for="search">Enter a search term or phrase</label>**

**<input id="search" type="text" autocomplete="off" role="searchbox" />**

**<div id="clear" class="none button clear" role="button" tabindex="0" aria-label="Clear search terms">**

**<i class="fas fa-times"></i>**

**</div>**

**<button id="searchButton" class="button searchButton" type="submit" aria-label="Submit button">**

**<i class="fas fa-search blue"></i>**

**</button>**

**</form>**

**</section>**

**<section class="results">**

**<h2 class="offscreen">Search Results</h2>**

**<div class="statsBar">**

**<div id="stats" class="stats" aria-live="assertive"></div>**

**</div>**

**<div id="searchResults" class="searchResults"></div>**

**</section>**

**</main>**

**<footer>**

**<p><span aria-hidden="true">&lt;&lt;</span> Powered by <a href="https://www.wikipedia.org" target="\_blank">Wikipedia</a> <span aria-hidden="true">&gt;&gt;</span></p>**

**</footer>**

**</body>**

**</html>**

**Java Script**

*import {*

*setSearchFocus,*

*showClearTextButton,*

*clearSearchText,*

*clearPushListener*

*} from "./searchBar.js";*

*import {*

*deleteSearchResults,*

*buildSearchResults,*

*clearStatsLine,*

*setStatsLine*

*} from "./searchResults.js";*

*import { getSearchTerm, retrieveSearchResults } from "./dataFunctions.js";*

*document.addEventListener("readystatechange", (event) => {*

*if (event.target.readyState === "complete") {*

*initApp();*

*}*

*});*

*const initApp = () => {*

*setSearchFocus();*

*const search = document.getElementById("search");*

*search.addEventListener("input", showClearTextButton);*

*const clear = document.getElementById("clear");*

*clear.addEventListener("click", clearSearchText);*

*clear.addEventListener("keydown", clearPushListener);*

*const form = document.getElementById("searchBar");*

*form.addEventListener("submit", submitTheSearch);*

*};*

*// Procedural "workflow" function*

*const submitTheSearch = (event) => {*

*event.preventDefault();*

*deleteSearchResults();*

*processTheSearch();*

*setSearchFocus();*

*};*

*// Procedural*

*const processTheSearch = async () => {*

*clearStatsLine();*

*const searchTerm = getSearchTerm();*

*if (searchTerm === "") return; //TODO:*

*const resultArray = await retrieveSearchResults(searchTerm);*

*if (resultArray.length) buildSearchResults(resultArray);*

*setStatsLine(resultArray.length);*

*};*

*export const getSearchTerm = () => {*

*const rawSearchTerm = document.getElementById("search").value.trim();*

*const regex = /[ ]{2,}/gi;*

*const searchTerm = rawSearchTerm.replaceAll(regex, " ");*

*return searchTerm;*

*};*

*export const retrieveSearchResults = async (searchTerm) => {*

*const wikiSearchString = getWikiSearchString(searchTerm);*

*const wikiSearchResults = await requestData(wikiSearchString);*

*let resultArray = [];*

*if (wikiSearchResults.hasOwnProperty("query")) {*

*resultArray = processWikiResults(wikiSearchResults.query.pages);*

*}*

*return resultArray;*

*};*

*const getWikiSearchString = (searchTerm) => {*

*const maxChars = getMaxChars();*

*const rawSearchString = `https://en.wikipedia.org/w/api.php?action=query&generator=search&gsrsearch=${searchTerm}&gsrlimit=20&prop=pageimages|extracts&exchars=${maxChars}&exintro&explaintext&exlimit=max&format=json&origin=\*`;*

*const searchString = encodeURI(rawSearchString);*

*return searchString;*

*};*

*const getMaxChars = () => {*

*const width = window.innerWidth || document.body.clientWidth;*

*let maxChars;*

*if (width < 414) maxChars = 65;*

*if (width >= 414 && width < 1400) maxChars = 100;*

*if (width >= 1400) maxChars = 130;*

*return maxChars;*

*};*

*const requestData = async (searchString) => {*

*try {*

*const response = await fetch(searchString);*

*const data = await response.json();*

*return data;*

*} catch (err) {*

*console.error(err);*

*}*

*};*

*const processWikiResults = (results) => {*

*const resultArray = [];*

*Object.keys(results).forEach((key) => {*

*const id = key;*

*const title = results[key].title;*

*const text = results[key].extract;*

*const img = results[key].hasOwnProperty("thumbnail")*

*? results[key].thumbnail.source*

*: null;*

*const item = {*

*id: id,*

*title: title,*

*img: img,*

*text: text*

*};*

*resultArray.push(item);*

*});*

*return resultArray;*

*};*

export const setSearchFocus = () => {

document.getElementById("search").focus();

};

export const showClearTextButton = () => {

const search = document.getElementById("search");

const clear = document.getElementById("clear");

if (search.value.length) {

clear.classList.remove("none");

clear.classList.add("flex");

} else {

clear.classList.add("none");

clear.classList.remove("flex");

}

};

export const clearSearchText = (event) => {

event.preventDefault();

document.getElementById("search").value = "";

const clear = document.getElementById("clear");

clear.classList.add("none");

clear.classList.remove("flex");

setSearchFocus();

};

export const clearPushListener = (event) => {

if (event.key === "Enter" || event.key === " ") {

event.preventDefault();

document.getElementById("clear").click();

}

};

export const deleteSearchResults = () => {

const parentElement = document.getElementById("searchResults");

let child = parentElement.lastElementChild;

while (child) {

parentElement.removeChild(child);

child = parentElement.lastElementChild;

}

};

export const buildSearchResults = (resultArray) => {

resultArray.forEach((result) => {

const resultItem = createResultItem(result);

const resultContents = document.createElement("div");

resultContents.classList.add("resultContents");

if (result.img) {

const resultImage = createResultImage(result);

resultContents.append(resultImage);

}

const resultText = createResultText(result);

resultContents.append(resultText);

resultItem.append(resultContents);

const searchResults = document.getElementById("searchResults");

searchResults.append(resultItem);

});

};

const createResultItem = (result) => {

const resultItem = document.createElement("div");

resultItem.classList.add("resultItem");

const resultTitle = document.createElement("div");

resultTitle.classList.add("resultTitle");

const link = document.createElement("a");

link.href = `https://en.wikipedia.org/?curid=${result.id}`;

link.textContent = result.title;

link.target = "\_blank";

resultTitle.append(link);

resultItem.append(resultTitle);

return resultItem;

};

const createResultImage = (result) => {

const resultImage = document.createElement("div");

resultImage.classList.add("resultImage");

const img = document.createElement("img");

img.src = result.img;

img.alt = result.title;

resultImage.append(img);

return resultImage;

};

const createResultText = (result) => {

const resultText = document.createElement("div");

resultText.classList.add("resultText");

const resultDescription = document.createElement("p");

resultDescription.classList.add("resultDescription");

resultDescription.textContent = result.text;

resultText.append(resultDescription);

return resultText;

};

export const clearStatsLine = () => {

document.getElementById("stats").textContent = "";

};

export const setStatsLine = (numberOfResults) => {

const statLine = document.getElementById("stats");

if (numberOfResults) {

statLine.textContent = `Displaying ${numberOfResults} results.`;

} else {

statLine.textContent = "Sorry, no results.";

}

};

**CSS**

main,footer,.

searchEntry,.results,.results

.searchResults .resultItem

{display:-webkit-box;

display:-ms-flexbox;

display:flex;

-webkit-box-align:center;

-ms-flex-align:center;

align-items:center

}

\*{padding:0;

margin:0;

-webkit-box-sizing:border-box;

box-sizing:border-box}

html,body{

background-color:#fff;

color:#000;

width:100vw;

min-height:100vh;

font-family:"Roboto",Arial,sans-serif;font-size:22px

}

main{

-webkit-box-orient:vertical;

-webkit-box-direction:normal;

-ms-flex-direction:column;

flex-direction:column;

-webkit-box-pack:start;

-ms-flex-pack:start;

justify-content:flex-start;

min-height:calc(100vh – 60px)

}

footer{

width:100%;

height:60px;

-webkit-box-pack:center;

-ms-flex-pack:center;

justify-content:center

}

footer p{

color:#70757a;

font-size:0.5rem

}

@media only screen and (min-width: 768px){

footer p{font-size:1rem}

}

footer p a{color:#70757a}img{display:block}.blue{color:#4885ed}.red{color:#db3236}.yellow{color:#ffc107}

.green{color:#3cba54}

.exclaim{

display:inline-block;

font-size:2.5rem;

-webkit-transform:rotate(12deg);

transform:rotate(12deg)

}

@media only screen and (min-width: 768px)

{.exclaim{font-size:5rem}}

.offscreen{position:absolute;

left:-10000px}

.none{display:none}

.flex{

display:-webkit-box;

display:-ms-flexbox;

display:flex}

.searchEntry{-webkit-box-orient:vertical;

-webkit-box-direction:normal;

-ms-flex-direction:column;

flex-direction:column;

-webkit-box-pack:start;

-ms-flex-pack:start;

justify-content:flex-start;

padding-top:40px}

.searchEntry .logo{

letter-spacing:-5px;

font-size:2rem;

font-weight:600;

margin-bottom:0.5rem

}

@media only screen and (min-width: 768px)

{

.searchEntry .logo{

font-size:4rem;

letter-spacing:-10px

}

}

.searchEntry .searchBar{

width:90vw;

display:-webkit-box;

display:-ms-flexbox;

display:flex;

border:2px solid #e6e6e6;

border-radius:500px;

padding:0.15rem 0.25rem 0.15rem 0.75rem

}

@media only screen and (min-width: 768px){

.searchEntry .searchBar

{

width:80vw;

padding:1.25rem 1.5rem}

}

@media only screen and (min-width: 1025px)

{.searchEntry .searchBar{width:60vw}}

.searchEntry .searchBar input[type="text"]

{-webkit-box-flex:1;

-ms-flex-positive:1;

flex-grow:1;

font-size:0.75rem;

text-align:left;

letter-spacing:0.1rem;

border:0;

outline:none;

min-width:150px}

@media only screen and (min-width: 768px)

{

.searchEntry .searchBar input[type="text"]

{

fontsize:1.25rem}

}

.searchEntry .searchBar .button{

cursor:pointer;

border:0;

background:transparent;

min-width:48px;

min-height:48px;

outline:none

}

.searchEntry .searchBar .button i{

font-family:"Font Awesome 5 Free";

font-size:1rem}

@media only screen and (min-width: 768px){

.searchEntry .searchBar .button i{font-size:1.5rem}

}

.searchEntry .searchBar .button:hover i,.searchEntry .searchBar .button:focus i{

padding-bottom:0.5rem;

border-bottom-width:1px;

border-bottom-style:solid

}

.searchEntry .searchBar .searchButton:hover i,.searchEntry .searchBar .searchButton:focus i{color:#3cba54}

.searchEntry .searchBar .clear:hover i,.searchEntry .searchBar .clear:focus I

{color:#db3236}

@media only screen and (min-width: 768px)

{

.searchEntry .searchBar .searchButton{padding-left:1rem}

}

.searchEntry .searchBar .clear{

-webkit-box-pack:center;

-ms-flex-pack:center;

justify-content:center;

-webkit-box-align:center;

-ms-flex-align:center;

align-items:center;

color:#70757a;

border-right:thin solid #d9d9d9

}

@media only screen and (min-width: 768px){

.searchEntry .searchBar .clear{

padding:0 1rem}

}

.searchEntry .searchBar:hover,.searchEntry .searchBar:focus-within{

-webkit-box-shadow:0 2px 5px 2px #e6e6e6;

box-shadow:0 2px 5px 2px #e6e6e6

}

.results{

-webkit-box-orient:vertical;

-webkit-box-direction:normal;

-ms-flex-direction:column;

flex-direction:column;

-webkit-box-pack:start;

-ms-flex-pack:start;

justify-content:flex-start;

padding:0.5rem 1rem;

width:90vw

}

@media only screen and (min-width: 768px){

.results{width:75vw}

}

@media only screen and (min-width: 1025px){

.results{width:55vw}

}

.results .statsBar{width:100%}

.results .statsBar .stats{

color:#70757a;

font-size:0.75rem

}

@media only screen and (min-width: 768px){

.results .statsBar .stats{

font-size:1rem}

}

.results .searchResults{width:100%}

.results .searchResults .resultItem{

-webkit-box-orient:vertical;

-webkit-box-direction:normal;

-ms-flex-direction:column;

flex-direction:column;

-webkit-box-pack:start;

-ms-flex-pack:start;

justify-content:flex-start;

width:100%;

padding:0.25rem 0

}

@media only screen and (min-width: 768px){

.results .searchResults .resultItem{

padding:0.5rem 0}

}

.results .searchResults .resultItem .resultTitle{

width:100%;

text-align:left;

font-size:1rem;

line-height:1.5rem;

margin-bottom:0.25rem;

overflow:hidden;

white-space:nowrap;

text-overflow:ellipsis;

padding:3px 2px

}

@media only screen and (min-width: 768px){

.results .searchResults .resultItem .resultTitle

{

font-size:1.5rem;

line-height:1.75rem;

margin-bottom:0.5rem}

}

.results .searchResults .resultItem .resultTitle a{

color:#1a0dab;

text-decoration:none;

cursor:pointer}

.results .searchResults .resultItem .resultTitle a:visited{color:#609}.results .searchResults .resultItem .resultTitle a:hover{text-decoration:underline}

.results .searchResults .resultItem .resultTitle a:focus{

outline:2px solid #000

}

.results .searchResults .resultItem .resultContents{

display:-webkit-box;

display:-msflexbox;

display:flex;

width:100%

}

.results .searchResults .resultItem .resultContents .resultImage{

margin-right:0.5rem

}

.results .searchResults .resultItem .resultContents .resultText{

-webkit-box-flex:1;

-ms-flex-positive:1;

flex-grow:1;

font-size:14px;

line-height:20px;

max-height:60px;

overflow:hidden;

text-overflow:ellipsis

}

@media only screen and (min-width: 768px){.results .searchResults .resultItem .resultContents .resultText{font-size:1rem;line-height:28px}}

/\*# sourceMappingURL=style.min.css.map \*/

{

"version": 3,

"mappings": "AKFJ,EUFCN,BFVR...",

"sources": [

"../../sass/style.scss",

"../../sass/\_base.scss",

"../../sass/\_sharedClasses.scss",

"../../sass/\_searchEntry.scss",

"../../sass/\_searchResults.scss"

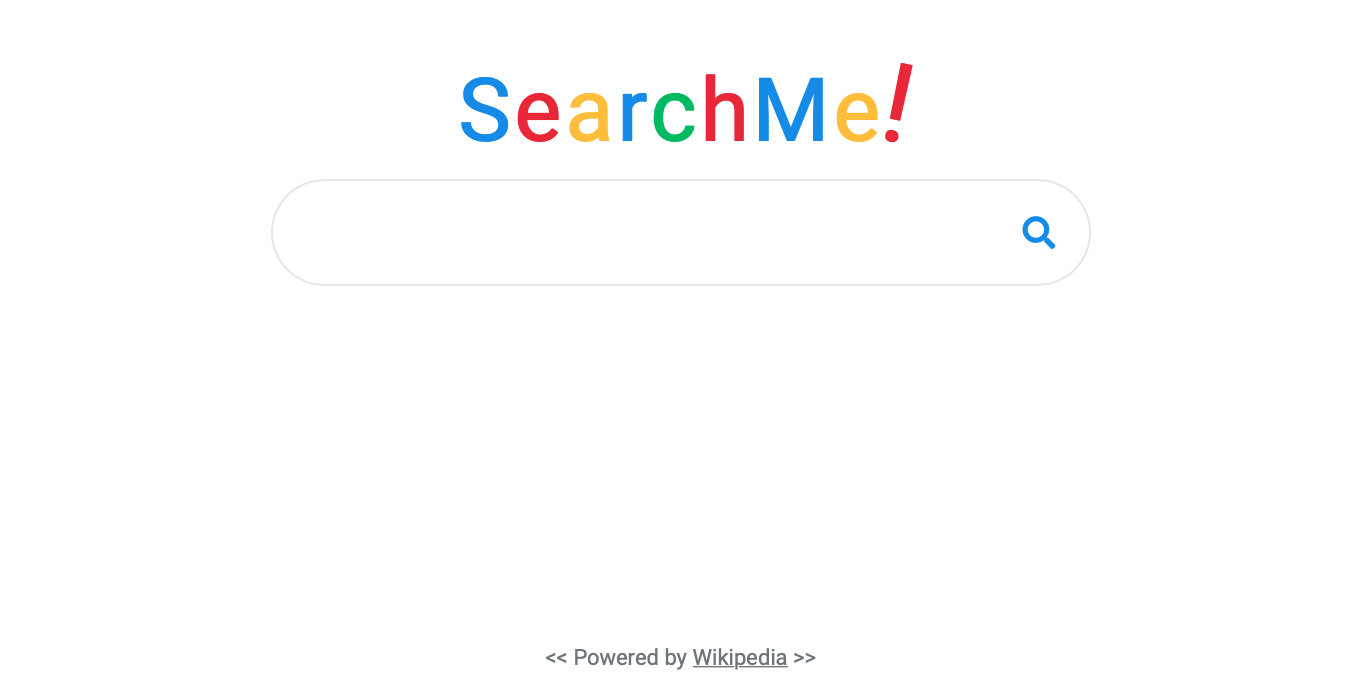
],

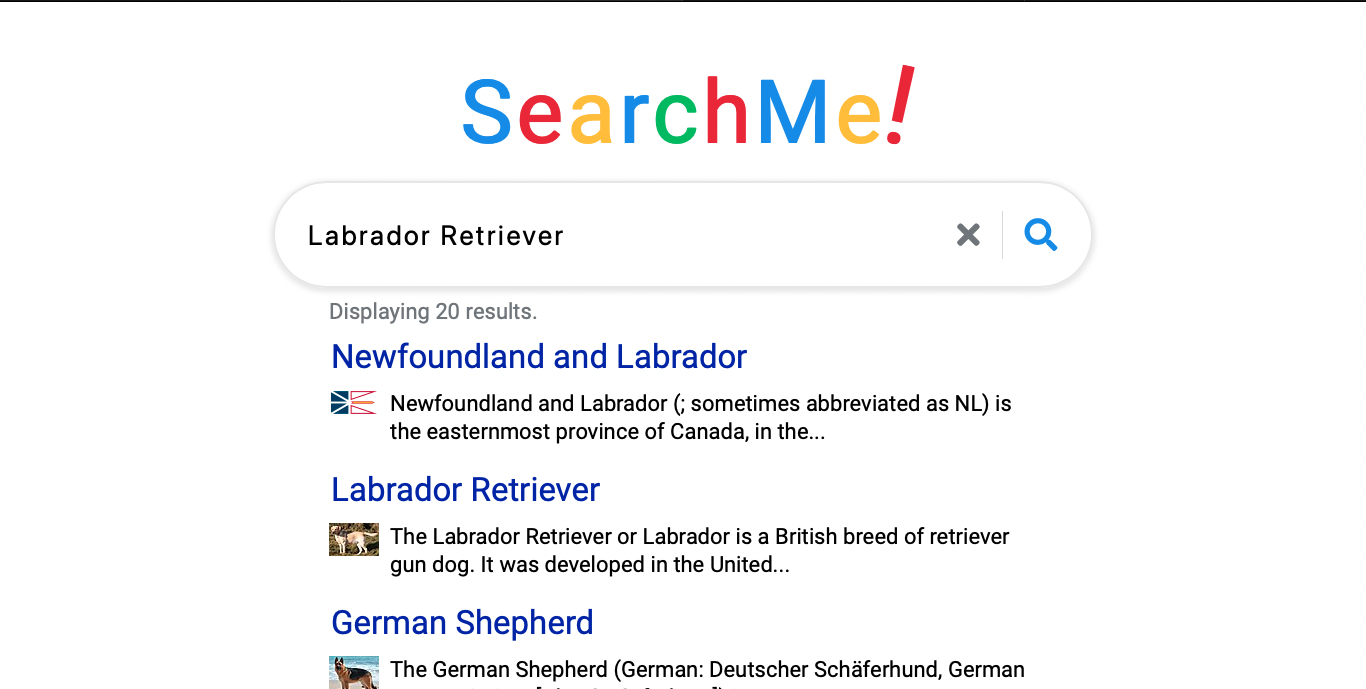
"names": [],

"file": "style.min.css"

}

**RESULT SNAPSHOT**

****



**References**

JAVASCRIPT: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

HTML: <https://devdocs.io/html/>

CSS: <https://developer.mozilla.org/en-US/docs/Web/CSS>

Wikipedia API: https://en.wikipedia.org/w/api.php