**Movie-App-By-HTML-CSS-JAVASCRIPT-**

**Introduction:**

I have Made this Movie App with the Help of HTML, CSS & JavaScript. In this Movie App you can Watch any Movie trailers as many as you want and You can Search Any Movie you like or your Favorite also you can Select any Movie Genre you want to watch movie also you can Go to next, current or previous page to Latest Release Hollywood & Bollywood Movie Trailers from YouTube.

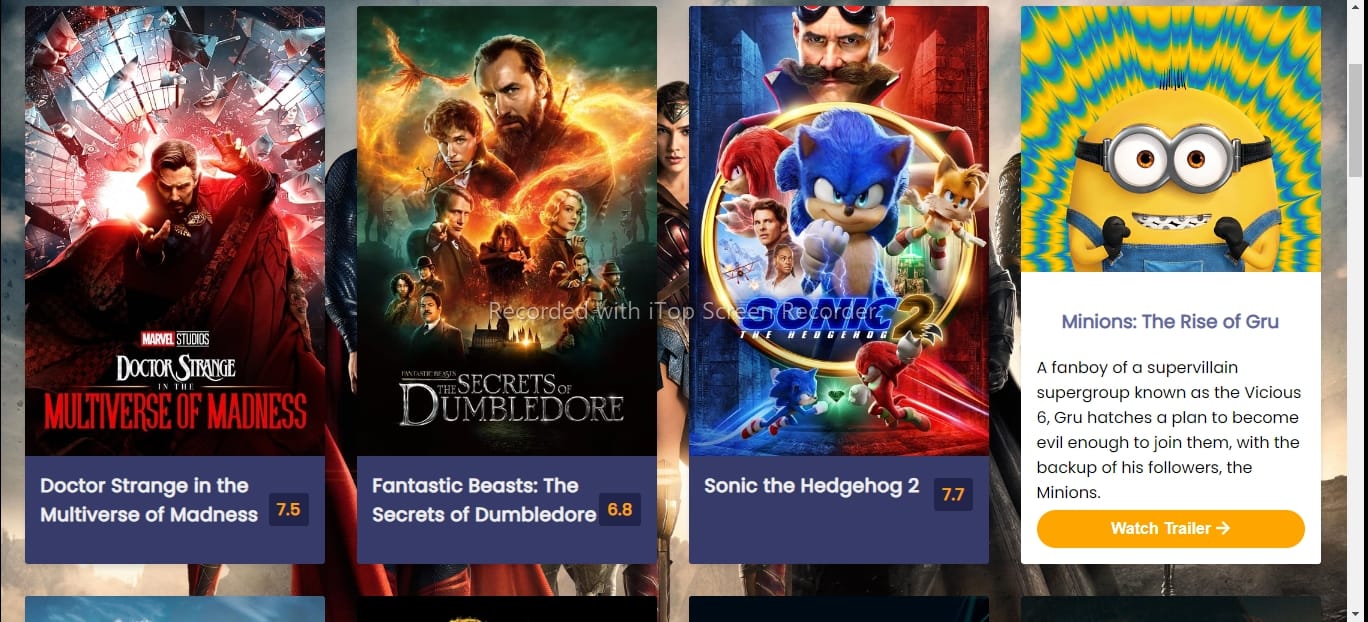
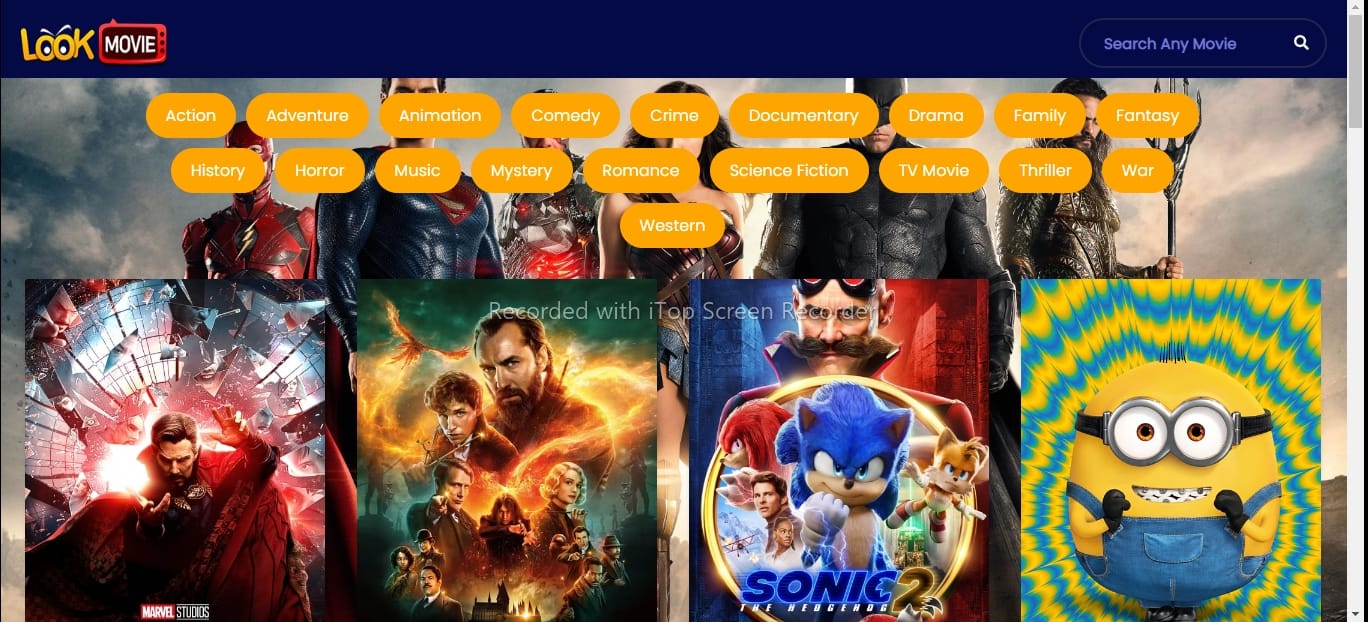
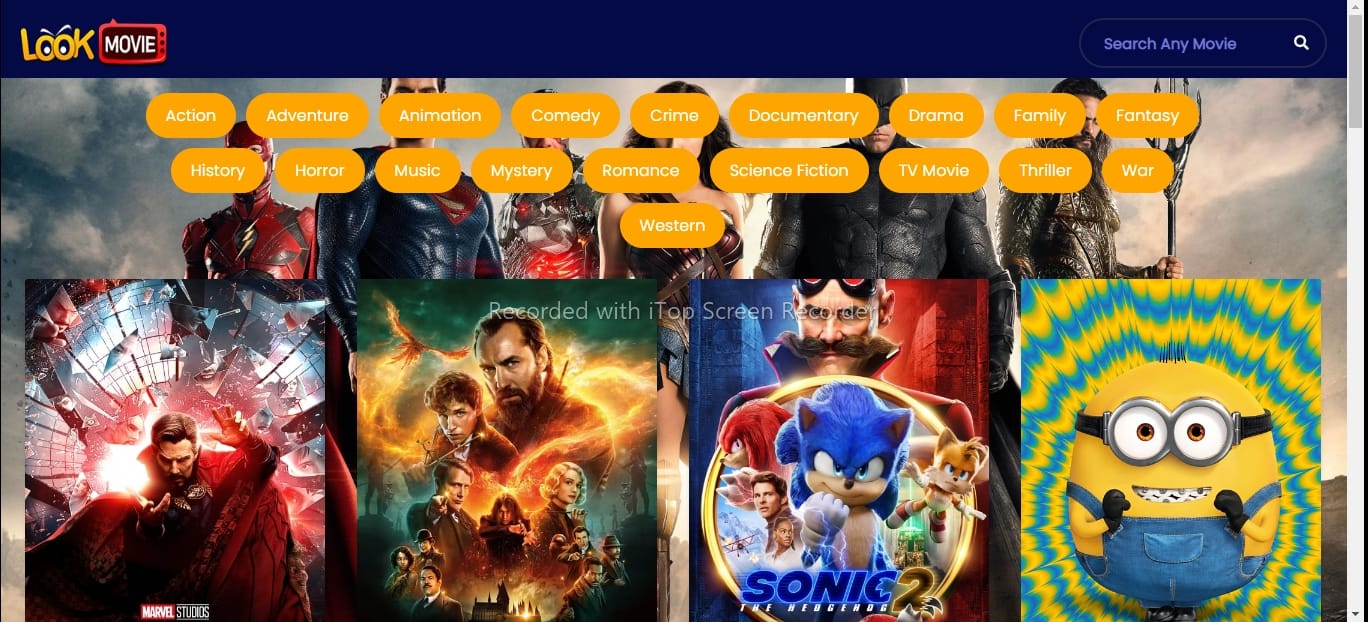
**Live Demo:**

**Run This Project on** ➡️ [**Movie App**](https://movie-app-by-html-css-javascript-.hamzajaffar.repl.co/)

**Materials Used:**

* [TMDB API](https://api.themoviedb.org/3/discover/movie?sort_by=popularity.desc&api_key=3fd2be6f0c70a2a598f084ddfb75487c&page=1&list) (The Movie Database)
* Read Movie Data from JSON File
* Convert to String
* Create a Responsive Mobile Friendly Movie App
* **ENJOY & WATCH MOVIES ON Movie APP 🚀**

**Screenshots:**



**Demo Full Clip if you want to see** [**Click Here ⤴️**](https://github.com/hamzajaffar/Movie-App-By-HTML-CSS-JAVASCRIPT-/blob/main/final%20demo%20clip%20(1).mp4) **Download that file and then watch**

**Licence**

MIT LICENCE Read Licence document for more details ->

**Bye! 👋**

HOPE U LIKE THIS PROJECT PLEASE! IF YOU ARE USING THIS CODE PLEASE GIVE **HAMZA ZAIDI** SOME CREDITS BY LIKE & GIVE A HEART!!! 😃👍💛

* Html code -

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

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

<meta

property="og:description"

content="Watch free 2022 newest Releases movies & series on LookMovie in HD Quality without Ads."

/>

<a href="/"><link

rel="icon"

href="https://lookmovie.la/favicon.ico"

type="image/icon"

/></a>

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

<script

src="https://kit.fontawesome.com/4e948b40f1.js"

crossorigin="anonymous"

></script>

<title>LookMovie | Watch Any Latest Movie for Free</title>

</head>

<body>

<header>

<img class="logo" src="/assets/logo.webp" alt="Logo" />

<form id="form">

<input

type="text"

placeholder="Search Any Movie"

autocomplete="off"

id="search"

class="search"

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

</form>

</header>

<div id="tags"></div>

<div id="myNav" class="overlay">

<!-- Button to close the overlay navigation -->

<a href="javascript:void(0)" class="closebtn" onclick="closeNav()"

>&times;</a

>

<!-- Overlay content -->

<div class="overlay-content" id="overlay-content"></div>

<a href="javascript:void(0)" class="arrow left-arrow" id="left-arrow"

><i class="fas fa-arrow-left"></i

></a>

<a href="javascript:void(0)" class="arrow right-arrow" id="right-arrow"

><i class="fas fa-arrow-right"></i

></a>

</div>

<main id="main"></main>

<div class="pagination">

<div class="page" id="prev">Previous Page</div>

<div class="current" id="current">1</div>

<div class="page" id="next">Next Page</div>

</div>

<script src="script.js"></script>

</body>

</html>

* Javascript –

const API\_KEY = "api\_key=1cf50e6248dc270629e802686245c2c8";

const BASE\_URL = "https://api.themoviedb.org/3";

const API\_URL = BASE\_URL + "/discover/movie?sort\_by=popularity.desc&" + API\_KEY;

const IMG\_URL = "https://image.tmdb.org/t/p/w500";

const searchURL = BASE\_URL + "/search/movie?" + API\_KEY;

const genres = [

{

id: 28,

name: "Action",

},

{

id: 12,

name: "Adventure",

},

{

id: 16,

name: "Animation",

},

{

id: 35,

name: "Comedy",

},

{

id: 80,

name: "Crime",

},

{

id: 99,

name: "Documentary",

},

{

id: 18,

name: "Drama",

},

{

id: 10751,

name: "Family",

},

{

id: 14,

name: "Fantasy",

},

{

id: 36,

name: "History",

},

{

id: 27,

name: "Horror",

},

{

id: 10402,

name: "Music",

},

{

id: 9648,

name: "Mystery",

},

{

id: 10749,

name: "Romance",

},

{

id: 878,

name: "Science Fiction",

},

{

id: 10770,

name: "TV Movie",

},

{

id: 53,

name: "Thriller",

},

{

id: 10752,

name: "War",

},

{

id: 37,

name: "Western",

},

];

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

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

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

const tagsEl = document.getElementById("tags");

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

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

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

var currentPage = 1;

var nextPage = 2;

var prevPage = 3;

var lastUrl = "";

var totalPages = 100;

// Get Movie Genres

var selectedGenre = [];

setGenre();

function setGenre() {

tagsEl.innerHTML = "";

genres.forEach((genre) => {

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

t.classList.add("tag");

t.id = genre.id;

t.innerText = genre.name;

t.addEventListener("click", () => {

if (selectedGenre.length == 0) {

selectedGenre.push(genre.id);

} else {

if (selectedGenre.includes(genre.id)) {

selectedGenre.forEach((id, idx) => {

if (id == genre.id) {

selectedGenre.splice(idx, 1);

}

});

} else {

selectedGenre.push(genre.id);

}

}

console.log(selectedGenre);

getMovies(API\_URL + "&with\_genres=" + encodeURI(selectedGenre.join(",")));

highlightSelection();

});

tagsEl.append(t);

});

}

// Highlight Tag

function highlightSelection() {

const tags = document.querySelectorAll(".tag");

tags.forEach((tag) => {

tag.classList.remove("highlight");

});

clearBtn();

if (selectedGenre.length != 0) {

selectedGenre.forEach((id) => {

const hightlightedTag = document.getElementById(id);

hightlightedTag.classList.add("highlight");

});

}

}

// Clear Highlighted Tag

function clearBtn() {

let clearBtn = document.getElementById("clear");

if (clearBtn) {

clearBtn.classList.add("highlight");

} else {

let clear = document.createElement("div");

clear.classList.add("tag", "highlight");

clear.id = "clear";

clear.innerText = "Clear x";

clear.addEventListener("click", () => {

selectedGenre = [];

setGenre();

getMovies(API\_URL);

});

tagsEl.append(clear);

}

}

// Get Movies

getMovies(API\_URL);

function getMovies(url) {

lastUrl = url;

fetch(url)

.then((res) => res.json())

.then((data) => {

console.log(data.results);

if (data.results.length !== 0) {

showMovies(data.results);

currentPage = data.page;

nextPage = currentPage + 1;

prevPage = currentPage - 1;

totalPages = data.total\_pages;

current.innerText = currentPage;

if (currentPage <= 1) {

prev.classList.add("disabled");

next.classList.remove("disabled");

} else if (currentPage >= totalPages) {

prev.classList.remove("disabled");

next.classList.add("disabled");

} else {

prev.classList.remove("disabled");

next.classList.remove("disabled");

}

tagsEl.scrollIntoView({ behavior: "smooth" });

} else {

main.innerHTML = `<div class="no-results">

<h1>The <span class="searchItem">${search.value}</span> You have Searched</h1>

<h2>No Results Found</h2>

<img class="sad-face" src="/assets/sad-face.svg" alt="Error" />

</div>`;

}

});

}

// Show Movies Images & Info

function showMovies(data) {

main.innerHTML = "";

data.forEach((movie) => {

const { title, poster\_path, vote\_average, overview, id } = movie;

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

movieEl.classList.add("movie");

movieEl.innerHTML = `

<img src="${

poster\_path ? IMG\_URL + poster\_path : "/assets/cinema.jpg"

}" alt="${title}">

<div class="movie-info">

<h3>${title}</h3>

<span class="${getColor(vote\_average)}">${vote\_average}</span>

</div>

<div class="overview">

<h3>${title}</h3>

${overview}

<br/>

<button class="know-more" id="${id}">Watch Trailer <i class="fas fa-arrow-right"></i></button

</div>

`;

main.appendChild(movieEl);

document.getElementById(id).addEventListener("click", () => {

console.log(id);

openNav(movie);

});

});

}

/\* Open when someone clicks on the span element \*/

const overlayContent = document.getElementById("overlay-content");

function openNav(movie) {

let id = movie.id;

fetch(BASE\_URL + "/movie/" + id + "/videos?" + API\_KEY)

.then((res) => res.json())

.then((videoData) => {

console.log(videoData);

if (videoData) {

document.getElementById("myNav").style.width = "100%";

if (videoData.results.length > 0) {

var embed = [];

var dots = [];

videoData.results.forEach((video, idx) => {

let { name, key, site } = video;

if (site == "YouTube") {

embed.push(`

<iframe width="560" height="315" src="https://www.youtube.com/embed/${key}" title="${name}" class="embed hide" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

`);

dots.push(`

<span class="dot">${idx + 1}</span>

`);

}

});

var content = `

<h1 class="no-results">${movie.original\_title}</h1>

<br/>

${embed.join("")}

<br/>

<div class="dots">${dots.join("")}</div>

`;

overlayContent.innerHTML = content;

activeSlide = 0;

showVideos();

} else {

overlayContent.innerHTML = `<h1 class="no-results">No Results Found 😥</h1>`;

}

}

});

}

/\* Close when someone clicks on the "x" symbol inside the overlay \*/

function closeNav() {

document.getElementById("myNav").style.width = "0%";

}

var activeSlide = 0;

var totalVideos = 0;

function showVideos() {

let embedClasses = document.querySelectorAll(".embed");

let dots = document.querySelectorAll(".dot");

totalVideos = embedClasses.length;

embedClasses.forEach((embedTag, idx) => {

if (activeSlide == idx) {

embedTag.classList.add("show");

embedTag.classList.remove("hide");

} else {

embedTag.classList.add("hide");

embedTag.classList.remove("show");

}

});

dots.forEach((dot, indx) => {

if (activeSlide == indx) {

dot.classList.add("active");

} else {

dot.classList.remove("active");

}

});

}

const leftArrow = document.getElementById("left-arrow");

const rightArrow = document.getElementById("right-arrow");

leftArrow.addEventListener("click", () => {

if (activeSlide > 0) {

activeSlide--;

} else {

activeSlide = totalVideos - 1;

}

showVideos();

});

rightArrow.addEventListener("click", () => {

if (activeSlide < totalVideos - 1) {

activeSlide++;

} else {

activeSlide = 0;

}

showVideos();

});

function getColor(vote) {

if (vote >= 8) {

return "green";

} else if (vote >= 5) {

return "orange";

} else {

return "red";

}

}

// Search Any Movie on Search Bar

form.addEventListener("submit", (e) => {

e.preventDefault();

const searchTerm = search.value;

selectedGenre = [];

setGenre();

if (searchTerm) {

getMovies(searchURL + "&query=" + searchTerm);

} else {

getMovies(API\_URL);

}

});

// Go to current, Next & Previous Page

prev.addEventListener("click", () => {

if (prevPage > 0) {

pageCall(prevPage);

}

});

next.addEventListener("click", () => {

if (nextPage <= totalPages) {

pageCall(nextPage);

}

});

function pageCall(page) {

let urlSplit = lastUrl.split("?");

let queryParams = urlSplit[1].split("&");

let key = queryParams[queryParams.length - 1].split("=");

if (key[0] != "page") {

let url = lastUrl + "&page=" + page;

getMovies(url);

} else {

key[1] = page.toString();

let a = key.join("=");

queryParams[queryParams.length - 1] = a;

let b = queryParams.join("&");

let url = urlSplit[0] + "?" + b;

getMovies(url);

}

}

* Css code-

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;400&display=swap');

:root {

--primary-color: #22254b;

--secondary-color: #373b69;

}

\* {

box-sizing: border-box;

}

body {

font-family: 'Poppins', sans-serif;

margin: 0;

background-image: url(/assets/bg.jpg);

background-position: cover;

background-repeat: no-repeat;

background-size: cover;

background-attachment: fixed;

}

.background-image {

position: fixed;

left: 0;

right: 0;

z-index: 1;

display: block;

background-image: url(/assets/bg.jpg);

background-position: cover;

background-repeat: no-repeat;

background-size: cover;

background-attachment: fixed;

/\* Add the blur effect \*/

filter: blur(5px);

-webkit-filter: blur(5px);

width: 1200px;

height: 800px;

-webkit-filter: blur(5px);

-moz-filter: blur(5px);

-o-filter: blur(5px);

-ms-filter: blur(5px);

filter: blur(5px);

}

header {

position: sticky !important;

padding: 10px;

height: 80px;

display: flex;

justify-content: space-between;

background-color: #030a46;

}

.logo {

width: 150px;

height: 50px;

margin: 10px;

}

form {

border: 2px solid var(--primary-color);

background-color: transparent;

border-radius: 50px;

font-family: inherit;

font-size: 15px;

padding: 0.5rem 1rem;

color: #fff;

user-select: none;

outline: none;

height: 50px;

margin: 10px;

}

fas fa{

color: white;

/\* font-size: 18px;

font-weight: bold; \*/

margin-right: 5px;

}

i fas fa-play {

color: white;

margin-right: 5px;

}

.search {

background-color: transparent;

outline: none;

border: none;

font-family: inherit;

font-size: 15px;

color: #fff;

height: 20px;

margin: 5px;

justify-content: center;

}

.search::placeholder {

color: #7378c5;

font-weight: bold;

}

.search:focus {

outline: none;

background-color: #030a46;

}

main {

display: flex;

flex-wrap: wrap;

justify-content: center;

}

.movie {

width: 300px;

margin: 1rem;

border-radius: 3px;

box-shadow: 0.2px 4px 5px rgba(0, 0, 0, 0.1);

background-color: var(--secondary-color);

position: relative;

overflow: hidden;

}

.movie img {

width: 100%;

}

.movie-info {

color: #eee;

display: flex;

align-items: center;

justify-content: space-between;

padding: 0.5rem 1rem 1rem;

letter-spacing: 0.5px;

}

.movie-info h3 {

margin-top: 0;

}

.movie-info span {

background-color: var(--primary-color);

padding: 0.25rem 0.5rem;

border-radius: 3px;

font-weight: bold;

}

.movie-info span.green {

color: lightgreen;

}

.movie-info span.orange {

color: orange;

}

.movie-info span.red {

color: red;

}

.overview {

position: absolute;

left: 0;

right: 0;

bottom: 0;

background-color: #fff;

padding: 1rem;

max-height: 100%;

transform: translateY(101%);

transition: transform 0.3s ease-in;

cursor: pointer;

user-select: none;

overflow: scroll;

}

.overview h3 {

color: #6368a3;

cursor: pointer;

text-align: center;

font-weight: bold;

}

.movie:hover .overview {

transform: translateY(0)

}

#tags {

width: 80%;

display: flex;

flex-wrap: wrap;

justify-content: center;

align-items: center;

margin: 10px auto;

user-select: none;

}

.tag {

color: white;

padding: 10px 20px;

background-color: orange;

border-radius: 50px;

margin: 5px;

display: inline-block;

cursor: pointer;

}

.tag.highlight {

background-color: red;

}

.no-results {

display: flex;

flex-direction: column;

font-weight: bold;

user-select: none;

justify-content: center;

text-align: center;

align-items: center;

}

.no-results h1 {

color: white;

font-weight: bold;

}

h1 .searchItem {

color: #6368a3;

text-decoration: underline;

font-weight: normal;

cursor: pointer;

}

.no-results h2 {

color: white;

font-weight: normal;

}

.no-results img {

width: 200px;

height: 200px;

user-select: none;

text-align: center;

justify-content: center;

}

.pagination {

display: flex;

margin: 10px 30px;

align-items: center;

justify-content: center;

color: white;

}

.page {

padding: 20px;

cursor: pointer;

user-select: none;

}

.page.disabled {

cursor: not-allowed;

color: grey;

}

.current {

padding: 10px 20px;

border-radius: 50%;

border: 5px solid orange;

font-size: 20px;

font-weight: 600;

user-select: none;

}

.know-more {

background-color: orange;

color: white;

font-size: 16px;

font-weight: bold;

border: 0;

border-radius: 50px;

padding: 10px 20px;

margin-top: 5px;

width: 100%;

justify-content: center;

text-align: center;

cursor: pointer;

}

/\* The Overlay (background) \*/

.overlay {

/\* Height & width depends on how you want to reveal the overlay (see JS below) \*/

height: 100%;

width: 0;

position: fixed;

/\* Stay in place \*/

z-index: 1;

/\* Sit on top \*/

left: 0;

top: 0;

background-color: rgb(0, 0, 0);

/\* Black fallback color \*/

background-color: rgba(0, 0, 0, 0.9);

/\* Black w/opacity \*/

overflow-x: hidden;

/\* Disable horizontal scroll \*/

transition: 0.5s;

user-select: none;

/\* 0.5 second transition effect to slide in or slide down the overlay (height or width, depending on reveal) \*/

}

/\* Position the content inside the overlay \*/

.overlay-content {

position: relative;

top: 10%;

/\* 25% from the top \*/

width: 100%;

/\* 100% width \*/

text-align: center;

/\* Centered text/links \*/

margin-top: 30px;

user-select: none;

/\* 30px top margin to avoid conflict with the close button on smaller screens \*/

}

/\* The navigation links inside the overlay \*/

.overlay a {

padding: 8px;

text-decoration: none;

font-size: 36px;

color: #818181;

display: block;

/\* Display block instead of inline \*/

transition: 0.3s;

user-select: none;

/\* Transition effects on hover (color) \*/

}

/\* When you mouse over the navigation links, change their color \*/

.overlay a:hover,

.overlay a:focus {

color: #f1f1f1;

}

/\* Position the close button (top right corner) \*/

.overlay .closebtn {

position: absolute;

top: 20px;

right: 45px;

font-size: 60px;

}

/\* When the height of the screen is less than 450 pixels, change the font-size of the links and position the close button again, so they don't overlap \*/

@media screen and (max-height: 450px) {

.overlay a {

font-size: 20px

}

.overlay .closebtn {

font-size: 40px;

top: 15px;

right: 35px;

}

}

.embed.hide {

display: none;

}

.embed.show {

display: inline-block;

}

.arrow {

position: absolute;

font-size: 40px;

}

.arrow.left-arrow {

top: 50%;

left: 5%;

transform: translateY(-50%);

}

.arrow.right-arrow {

top: 50%;

transform: translateY(-50%);

right: 5%;

}

.dots {

margin-top: 30px;

margin-bottom: 10px;

}

.dots .dot {

padding: 5px 15px;

border-radius: 50%;

border: 5px solid var(--secondary-color);

color: white;

font-size: 20px;

font-weight: 600;

margin: 5px;

}

.dots .dot.active {

border-color: orange;

}