

HTML, JAVA SCRIPT AND CSS CODES

(GROUP 18)

HTML CODE:

```
<!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="style.css">
  <title>Cinereview</title>
</head>
<body>

  <header>
    <form id="form">
      <input type="text" placeholder="Search" id="search"
class="search">
    </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">&#8656;</a>

    <a href="javascript:void(0)" class="arrow right-arrow" id="right-
arrow" >&#8658;</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>
```

JAVA SCRIPT CODE:

```
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 = 7;

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='+encodeURIComponent(selectedGenre.join(',')))
      highlightSelection()
    })
    tagsEl.append(t);
  })
}

```

```

function highlightSelection() {
  const tags = document.querySelectorAll('.tag');
  tags.forEach(tag => {
    tag.classList.remove('highlight')
  })
  clearBtn()
  if(selectedGenre.length !=0){
    selectedGenre.forEach(id => {
      const highlightedTag = document.getElementById(id);
      highlightedTag.classList.add('highlight');
    })
  }
}

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);
  }
}

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.scrollToView({behavior : 'smooth'})

    }else{
        main.innerHTML= `<h1 class="no-results">No Results Found</h1>`
    }

    })
}

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 = `
            

            <div class="movie-info">
                <h3>${title}</h3>
                <span class="${getColor(vote_average)}">${vote_average}</span>
            </div>

            <div class="overview">

                <h3>Overview</h3>
                ${overview}
                <br/>
                <button class="know-more" id="${id}">Know More</button>
            </div>
        `
    })
}

```

```

        main.appendChild(movieEl);

        document.getElementById(id).addEventListener('click', () => {
            console.log(id)
            openNav(movie)
        })
    })
}

const overlayContent = document.getElementById('overlay-content');
/* Open when someone clicks on the span element */
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, idx) => {
        if(activeSlide == idx){
            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'  
  }  
}
```

```
form.addEventListener('submit', (e) => {  
  e.preventDefault();  
  
  const searchTerm = search.value;  
  selectedGenre=[];  
  setGenre();  
  if(searchTerm) {  
    getMovies(searchURL+'&query='+searchTerm)  
  }else{  
    getMovies(API_URL);  
  }  
  
})
```

```
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);
  }
}

```

STYLE CSS CODE:

```

@import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;400;700&display=swap');

*{
  box-sizing: border-box;
}

:root{
  --primary-color:#22254b;
  --secondary-color: #373b69;
}

body{
  background-color: var(--primary-color);
  font-family: 'Poppins', sans-serif;
  margin: 0;
}

```

```
}

header{
  padding:1rem;
  display:flex;
  justify-content: flex-end;
  background-color: var(--secondary-color);
}

.search{
  background-color: transparent;
  border: 2px solid var(--primary-color);
  padding:0.5rem 1rem;
  border-radius: 50px;
  font-size: 1rem;
  color:#fff;
  font-family: inherit;
}

.search:focus{
  outline:0;
  background-color: var(--primary-color);
}

.search::placeholder{
  color: #7378c5;
}

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;
}

.movie:hover .overview{
  transform:translateY(0)
}
```

```
#tags{
  width:80%;
  display:flex;
  flex-wrap: wrap;
  justify-content: center;
  align-items: center;
  margin: 10px auto;
}

.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{
  color:white;
}

.pagination{
  display:flex;
  margin:10px 30px;
  align-items: center;
  justify-content: center;
  color:white;
}

.page{
  padding:20px;
  cursor:pointer;
}

.page.disabled{
  cursor:not-allowed;
  color:grey;
}

.current{
  padding: 10px 20px;
  border-radius: 50%;
}
```

```

border: 5px solid orange;
font-size: 20px;
font-weight: 600;
}

.know-more{
background-color: orange;
color:white;
font-size: 16px;
font-weight: bold;
border:0;
border-radius: 50px;
padding: 10px 20px;
margin-top: 5px;
}

/* 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; /* 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; /* 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; /* 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;
}

.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;
}
```