

Web Development Finals | BSCpE - 2C

Cabrera, John Wesley R.  
David, Kirby O.  
Mangalindan, Anthony Jr. S.  
Miguel, John Daniel T.

MAIN OBJECTIVE

The main purpose of this Musika web application is combining the use of Spotify and Youtube all in one place. It can be used to find your favorite songs and their accompanying music videos posted on Youtube.

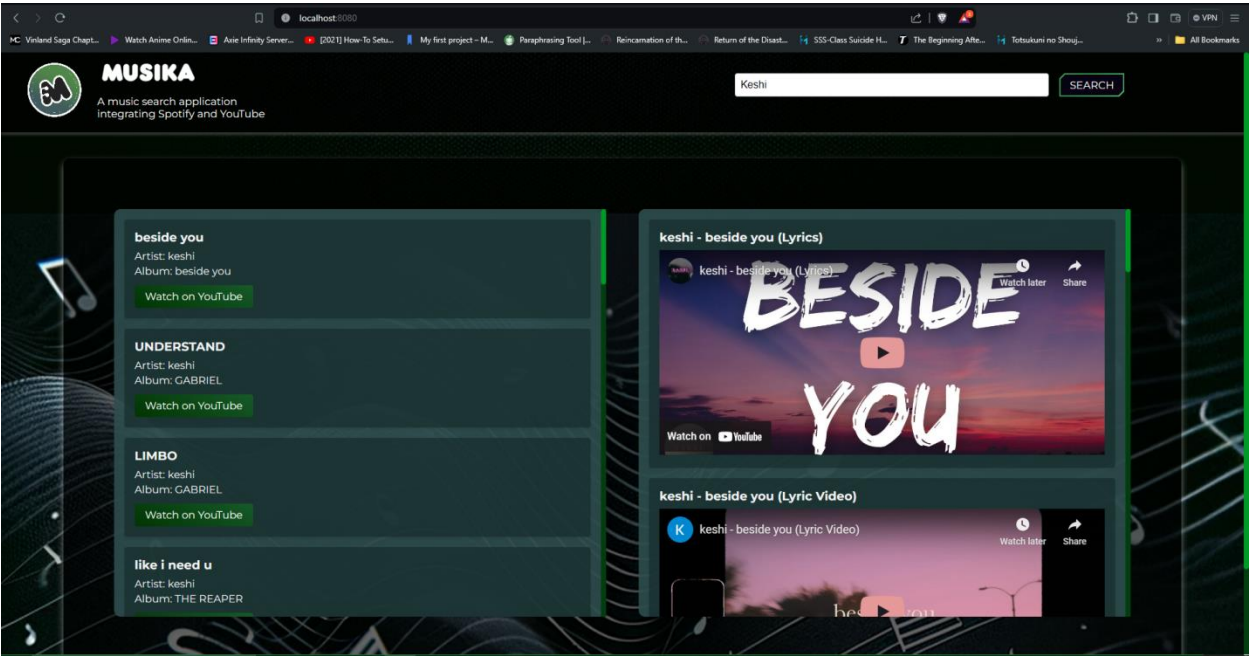
FEATURES AND FUNCTIONALITIES

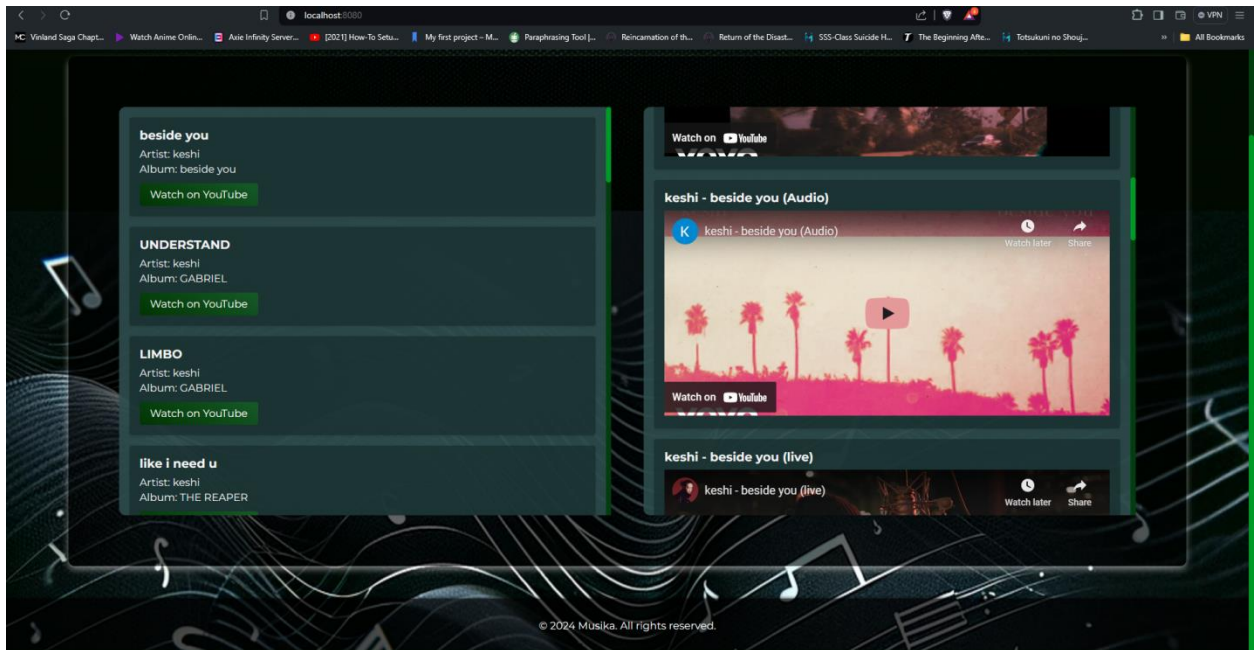
Musika leverages JavaScript to manipulate the DOM, allowing it to dynamically update content based on user interactions, creating a highly responsive and interactive user experience. The application seamlessly integrates with the Spotify API and the YouTube Data API, enabling it to fetch and display comprehensive music information, including details about songs, artists, albums, and related videos. By incorporating WebSocket functionality, Musika facilitates real-time updates, ensuring that users receive the most current information without needing to refresh the page.

One of the core features of the Musika web application is the music search button, which allows the user to easily search for specific songs, albums, and artists. After a search is done, a button is present that will show any related videos available on YouTube that might be related.

The overall design of the Musika web application keeps user experience as its top most priority. As such the design of the web application is made responsive, allowing for it to be used on a wide range of devices. To improve the overall feel and look of the Musika web application, custom assets such as the scroll bar, animated elements, and consistent design theme are made to give a distinct yet unique look.

SAMPLE OF THE WEBSITE.





## PROGRAM STRUCTURE

The project consists of the following key files and directories:

### 1. index.html

- This file contains the HTML structure of the web application. It includes the main layout, header, search input, and sections for displaying music information from Spotify and related videos from YouTube.

### 2. styles.css

- This file contains the CSS styles for the web application. It styles the layout, fonts, buttons, and other elements to ensure a cohesive and visually appealing design. It also includes custom scrollbar styles and media queries for mobile responsiveness.

### 3. button.css

- This file contains additional CSS styles specifically for button elements. It includes styles for button appearance, hover effects, and animations to enhance the user interaction experience.

### 4. app.js

- This file contains the client-side JavaScript code. It sets up the WebSocket connection, handles user interactions (such as searching for music), and dynamically updates the DOM with music information and related videos. It also includes error handling for failed API requests.

### 5. server.js

- This file contains the server-side code using Express.js and WebSocket. It serves static files, sets up WebSocket connections, and processes messages from the client to fetch and return music data from the Spotify API and related videos from the YouTube Data API.

INDEX.HTML

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Musika</title>
7      <link rel="stylesheet" href="style.css">
8      <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&display=swap" rel="stylesheet">
9      <link rel="stylesheet" href="button.css">
10 </head>
11 <body>
12     <header>
13         <div class="logo-container">
14             
15             <div class="header-text">
16                 
17                 <p>A music search application <br>
18                 integrating Spotify and YouTube</p>
19             </div>
20         </div>
21         <div class="search-container">
22             <input type="text" id="search" placeholder="Search for songs, artists, albums...">
23             <button id="searchButton" class="btn">Search</button>
24         </div>
25     </header>
26     <main>
27         <section class="container">
28             <div class="results-container">
29                 <div id="results" class="spotify-results"></div>
30                 <div id="youtubeResults" class="youtube-results"></div>
31             </div>
32         </section>
33     </main>
34     <footer>
35         <p>&copy; 2024 Musika. All rights reserved.</p>
36     </footer>
37     <script src="app.js"></script>
38 </body>
39 </html>
40
```

## SERVER.JS



```
1  const express = require('express');
2  const path = require('path');
3  const WebSocket = require('ws');
4
5  const app = express();
6
7
8  app.use(express.static(path.join(__dirname, 'public')));
9
10
11 const server = require('http').createServer(app);
12 const wss = new WebSocket.Server({ server });
13
14 wss.on('connection', ws => {
15   console.log('Client connected');
16   ws.on('message', message => {
17     console.log(`Received message: ${message}`);
18
19     wss.clients.forEach(client => {
20       if (client.readyState === WebSocket.OPEN) {
21         client.send(message);
22       }
23     });
24   });
25
26   ws.on('close', () => {
27     console.log('Client disconnected');
28   });
29 });
30
31 server.listen(8080, () => {
32   console.log('Server is running on http://localhost:8080');
33   console.log('WebSocket server is running on ws://localhost:8080');
34 });
35
36 app.use((req, res, next) => {
37   res.setHeader(
38     'Permissions-Policy',
39     'geolocation=(self), microphone=(), camera=(), fullscreen=(self)'
40   );
41   next();
42 });
43
44 app.listen(8000, () => {
45   console.log('Server running on port 8000');
46 });
```

## STYLE.CSS

```
1  body {
2    font-family: 'Montserrat';
3    font-weight: 400;
4    background-image: url('background_1920x1080.png');
5    background-size: cover;
6    background-repeat: no-repeat;
7    background-attachment: fixed;
8    margin: 0;
9    padding: 0;
10   display: flex;
11   flex-direction: column;
12   min-height: 100vh;
13   color: #fff;
14   overflow-x: scroll;
15 }
16 header {
17   background-color: rgba(0, 0, 0, 0.7);
18   padding: 0.5rem 2rem;
19   display: flex;
20   align-items: center;
21   justify-content: space-between;
22   box-shadow: 0 4px 4px rgba(255, 251, 251, 0.2);
23   width: 100%;
24   margin-bottom: 40px;
25   padding-bottom: 15px;
26 }
27 header .logo {
28   height: 100px;
29   width: 100px;
30   border-radius: 50%;
31
32   object-fit: cover;
33 }
34 .logo-container{
35   display: flex;
36   width: 50%;
37 }
38 .logo-name{
39   width: 10rem;
40 }
41 .header-text {
42   flex-direction: column;
43   text-align: left;
44   flex-grow: 1;
45   padding-left: 1rem;
46 }
47 header h1 {
48   margin: 0;
49   font-size: 2rem;
50 }
51 header p {
52   margin: 0.5rem 0 0;
53 }
54 main {
55   box-shadow: 0 5px 10px rgba(255, 251, 251, 0.5);
56   padding: 3rem;
57   width: 90%;
58   margin: 0 auto;
59   box-sizing: border-box;
60   flex: 1;
61   background-color: rgba(0, 0, 0, 0.6);
62   border-radius: 10px;
63   overflow: hidden;
64 }
65 .search-container {
66   display: flex;
67   justify-content: center;
68   margin-bottom: 1rem;
69   width: 50%;
70   margin-right: 50px;
71 }
72 .search-container input {
73   padding: 0.5rem;
74   font-size: 1rem;
75   border: 1px solid #ccc;
76   border-radius: 4px;
77   width: 50%;
78 }
```

```
1  .search-container button {
2    padding: 0.5rem 1rem;
3    font-size: 1rem;
4    border: none;
5    background-color: #1DB954;
6    color: white;
7    cursor: pointer;
8    border-radius: 4px;
9    margin-left: 1rem;
10 }
11 .search-container button:hover {
12   background-color: #17a047;
13 }
14 #results{
15   margin-right: 50px;
16 }
17 .results-container {
18   display: flex;
19   justify-content: space-between;
20   padding: 50px;
21   height: calc(100vh - 300px);
22   overflow: hidden;
23   padding: 30px;
24 }
25 .spotify-results, .youtube-results {
26   width: 48%;
27   max-height: 100%;
28   background-color: rgba(47, 79, 79, 0.9);
29   border-radius: 10px;
30   padding: 1rem;
31   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
32   color: #000;
33   overflow-y: auto;
34 }
35 .track, .video {
36   background-color: rgba(18, 41, 41, 0.644);
37   padding: 1rem;
38   margin-bottom: 1rem;
39   border-radius: 4px;
40   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
41 }
42 .track h3, .video h3 {
43   margin: 0 0 0.5rem;
44   color: whitesmoke;
45 }
46 .track p {
47   margin: 0.25rem 0;
48   color: whitesmoke;
49 }
50 .track button {
51   padding: 0.5rem 1rem;
52   font-size: 1rem;
53   border: none;
54   background-color: black;
55   color: white;
56   cursor: pointer;
57   border-radius: 4px;
58   margin-top: 0.5rem;
59 }
60 .track button:hover {
61   background-color: darkslategray;
62 }
63 iframe {
64   width: 100%;
65   max-width: 100%;
66 }
67 footer {
68   text-align: center;
69   justify-content: center;
70   margin-top: 50px;
71   background-color: rgba(0, 0, 0, 0.7);
72   color: white;
73   padding: 1rem;
74   box-shadow: 0 -2px 4px rgba(0, 0, 0, 0.5);
75   width: 100%;
76   box-sizing: border-box;
77   display: flex;
78 }
79 ::-webkit-scrollbar{
80   height: .1rem;
81   width: .5rem;
82 }
83 ::-webkit-scrollbar-track{
84   background-color: rgb(3, 62, 16);
85 }
86 ::-webkit-scrollbar-thumb{
87   background-color: rgb(0, 154, 38);
88   border-radius: 15rem;
89 }
```



## APP.JS

```
1  const clientId = 'ee48dca882d64b2494c2120b3c4b81db';
2  const clientSecret = '52a1424173834365a5257e028979f34a';
3  const redirectUri = 'http://localhost:8000';
4
5  let accessToken = '';
6
7
8  const ws = new WebSocket('ws://localhost:8080');
9
10 ws.onopen = () => {
11   console.log('Connected to WebSocket server');
12 };
13
14 ws.onmessage = (event) => {
15   const message = event.data;
16
17   console.log('Received message:', message);
18 };
19
20 ws.onclose = () => {
21   console.log('Disconnected from WebSocket server');
22 };
23
24 function sendMessage(message) {
25   if (ws.readyState === WebSocket.OPEN) {
26     ws.send(message);
27   }
28 }
29
30
31 document.getElementById('searchButton').addEventListener('click', () => {
32   const query = document.getElementById('search').value;
33   searchSpotify(query);
34   sendMessage(`New search: ${query}`);
35 });
36
37
38 const searchSpotify = async (query) => {
39   try {
40     if (!accessToken) {
41       await getAccessToken();
42     }
43     const response = await fetch(`https://api.spotify.com/v1/search?q=${query}&type=track,artist,album`, {
44       headers: {
45         'Authorization': `Bearer ${accessToken}`
46       }
47     });
48     const data = await response.json();
49     displayResults(data);
50   } catch (error) {
51     console.error('Error fetching data from Spotify:', error);
52   }
53 };
54
55
56 const getAccessToken = async () => {
57   const authString = btoa(`${clientId}:${clientSecret}`);
58   const response = await fetch('https://accounts.spotify.com/api/token', {
59     method: 'POST',
60     headers: {
61       'Content-Type': 'application/x-www-form-urlencoded',
62       'Authorization': `Basic ${authString}`
63     },
64     body: 'grant_type=client_credentials'
65   });
66   const data = await response.json();
67   accessToken = data.access_token;
68 };
69
```

```

1  const displayResults = (data) => {
2    const resultsDiv = document.getElementById('results');
3    resultsDiv.innerHTML = '';
4    if (data.tracks.items.length > 0) {
5      data.tracks.items.forEach(item => {
6        const trackElement = document.createElement('div');
7        trackElement.classList.add('track');
8        trackElement.innerHTML = `
9          <h3>${item.name}</h3>
10         <p>Artist: ${item.artists[0].name}</p>
11         <p>Album: ${item.album.name}</p>
12         <button onclick="searchYouTube('${item.name} ${item.artists[0].name}')">Watch on YouTube</button>
13       `;
14        resultsDiv.appendChild(trackElement);
15      });
16    } else {
17      resultsDiv.innerHTML = '<p>No results found</p>';
18    }
19  };

```

```

1  const searchYouTube = async (query) => {
2    try {
3      const response = await fetch('https://www.googleapis.com/youtube/v3/search?part=snippet&type=video&maxResults=10&q=${encodeURIComponent(query)}&key=AIzaSyCSz12XVZNo3Un6KGACz1rA2Wj-x8q3nCA');
4      const data = await response.json();
5      displayYouTubeResults(data);
6    } catch (error) {
7      console.error('Error fetching data from YouTube:', error);
8    }
9  };

```

```

1  const displayYouTubeResults = (data) => {
2    const youtubeResultsDiv = document.getElementById('youtubeResults');
3    youtubeResultsDiv.innerHTML = '';
4    if (data.items.length > 0) {
5      data.items.forEach(item => {
6        const videoId = item.id.videoId;
7        const videoElement = document.createElement('div');
8        videoElement.classList.add('video');
9        videoElement.innerHTML = `
10         <h3>${item.snippet.title}</h3>
11         <iframe width="560" height="315" src="https://www.youtube.com/embed/${videoId}" frameborder="0" allowfullscreen></iframe>
12       `;
13        youtubeResultsDiv.appendChild(videoElement);
14      });
15    } else {
16      youtubeResultsDiv.innerHTML = '<p>No results found</p>';
17    }
18  };

```



## BUTTON.CSS

```
1  .btn {
2    --border-color: linear-gradient(-45deg, #7af0ad, #389b4f, #79bab9);
3    --border-width: 0.125em;
4    --curve-size: 0.5em;
5    --blur: 30px;
6    --bg: #080312;
7    --color: #ffffff;
8    color: var(--color);
9    cursor: pointer;
10
11    position: relative;
12    isolation: isolate;
13    display: inline-grid;
14    place-content: center;
15    padding: 0.5em 1.5em;
16
17    border: 0;
18    text-transform: uppercase;
19    box-shadow: 10px 10px 20px rgba(0, 0, 0, 0.6);
20    clip-path: polygon(
21      0% var(--curve-size),
22      var(--curve-size) 0,
23      100% 0,
24      100% calc(100% - var(--curve-size)),
25      calc(100% - var(--curve-size)) 100%,
26      0 100%
27    );
28    transition: color 250ms;
29  }
30
31  .btn::after,
32  .btn::before {
33    content: "";
34    position: absolute;
35    inset: 0;
36  }
37
38  .btn::before {
39    background: var(--border-color);
40    background-size: 300% 300%;
41    animation: move-bg7234 5s ease infinite;
42    z-index: -2;
43  }
44
45  @keyframes move-bg7234 {
46    0% {
47      background-position: 31% 0%;
48    }
49
50    50% {
51      background-position: 70% 100%;
52    }
53
54    100% {
55      background-position: 31% 0%;
56    }
57  }
58
59  .btn::after {
60    background: var(--bg);
61    z-index: -1;
62    clip-path: polygon(
63      var(--border-width)
64      calc(var(--curve-size) + var(--border-width) * 0.5),
65      calc(var(--curve-size) + var(--border-width) * 0.5) var(--border-width),
66      calc(100% - var(--border-width)) var(--border-width),
67      calc(100% - var(--border-width))
68      calc(100% - calc(var(--curve-size) + var(--border-width) * 0.5)),
69
70      calc(100% - calc(var(--curve-size) + var(--border-width) * 0.5))
71      calc(100% - var(--border-width)),
72      var(--border-width) calc(100% - var(--border-width))
73    );
74    transition: clip-path 500ms;
75  }
```



```
1  .btn:where(:hover, :focus)::after {
2    clip-path: polygon(
3      calc(100% - var(--border-width))
4      calc(100% - calc(var(--curve-size) + var(--border-width) * 0.5)),
5      calc(100% - var(--border-width)) var(--border-width),
6      calc(100% - var(--border-width)) var(--border-width),
7      calc(100% - var(--border-width))
8      calc(100% - calc(var(--curve-size) + var(--border-width) * 0.5)),
9
10     calc(100% - calc(var(--curve-size) + var(--border-width) * 0.5))
11     calc(100% - var(--border-width)),
12
13     calc(100% - calc(var(--curve-size) + var(--border-width) * 0.5))
14     calc(100% - var(--border-width))
15   );
16   transition: 200ms;
17 }
18
19 .btn:where(:hover, :focus) {
20   color: #fff;
21 }
22 button {
23   border: none;
24   color: #fff;
25   background-image: linear-gradient(30deg, #023604, #175b29);
26   border-radius: 20px;
27   background-size: 100% auto;
28   font-family: inherit;
29   font-size: 17px;
30   padding: 0.6em 1.5em;
31 }
32
33 button:hover {
34   background-position: right center;
35   background-size: 200% auto;
36   -webkit-animation: pulse 2s infinite;
37   animation: pulse512 1.5s infinite;
38 }
39
40 @keyframes pulse512 {
41   0% {
42     box-shadow: 0 0 0 0 #60606066;
43   }
44
45   70% {
46     box-shadow: 0 0 0 10px rgb(218 103 68 / 0%);
47   }
48
49   100% {
50     box-shadow: 0 0 0 0 rgb(218 103 68 / 0%);
51   }
52 }
```