

Project presentation

Presented by Akai Mansur, Orazgeldi Nursultan

Project aims and goals

01.

Create a user-friendly and efficient digital music library platform that allows users to organize, explore, and interact with music content such as songs, albums, artists, and playlists.

02.

Design a system that facilitates personalized music collections, easy access to music data, and the generation of valuable insights about user preferences, popular genres, and artists, meeting the modern needs of digital music management.

Relevance

The "Music Library" project is highly relevant in today's digital age, as it addresses the growing demand for efficient music organization and management. By leveraging NoSQL databases like MongoDB, the project demonstrates scalable and flexible data handling for songs, albums, artists, and playlists. It provides users with a customizable solution for organizing their music collections while offering developers hands-on experience with modern web development practices. Additionally, the project has potential for future enhancements, such as integration with streaming APIs or AI-powered recommendations, making it a valuable foundation for real-world applications. Overall, it bridges the gap between academic knowledge and practical implementation, showcasing its importance in both learning and real-world problem-solving.



01

Spotify is one of the most popular music streaming platforms, offering millions of songs, podcasts, and personalized playlists. It uses advanced recommendation algorithms to create custom listening experiences for users. Spotify's focus on user engagement through features like playlist sharing and collaborative playlists aligns closely with the goals of our "Music Library" project.

02

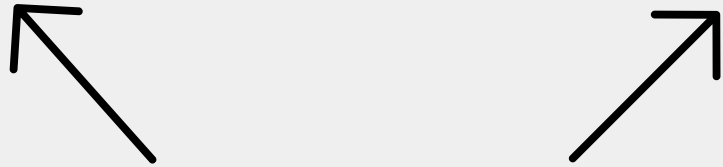
Apple Music provides users with a vast library of songs, curated playlists, and personalized music recommendations. It integrates seamlessly with Apple devices and offers offline listening options for subscribers. The curated approach to playlists and artist-focused content management inspired parts of our project.

03

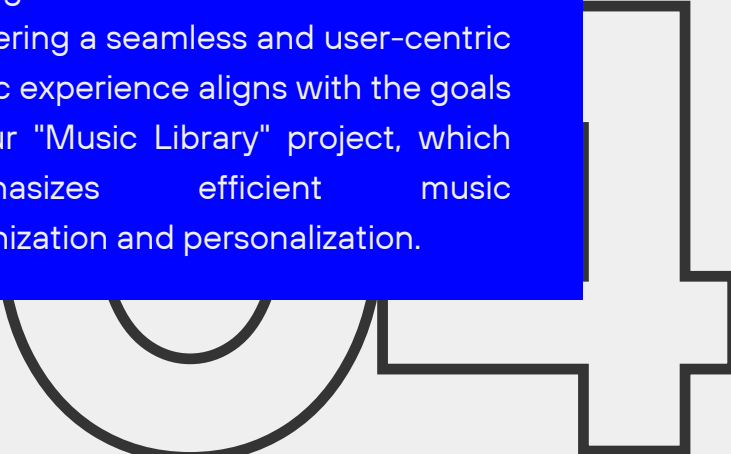
YouTube Music combines traditional music streaming with access to live performances, music videos, and covers. It leverages YouTube's extensive video library to enhance the user experience. The platform's focus on combining different media types influenced our approach to organizing and retrieving diverse music-related data.

04

Deezer is a popular music streaming platform known for its vast library of songs, curated playlists, and personalized music recommendations through its Flow feature. Its focus on delivering a seamless and user-centric music experience aligns with the goals of our "Music Library" project, which emphasizes efficient music organization and personalization.

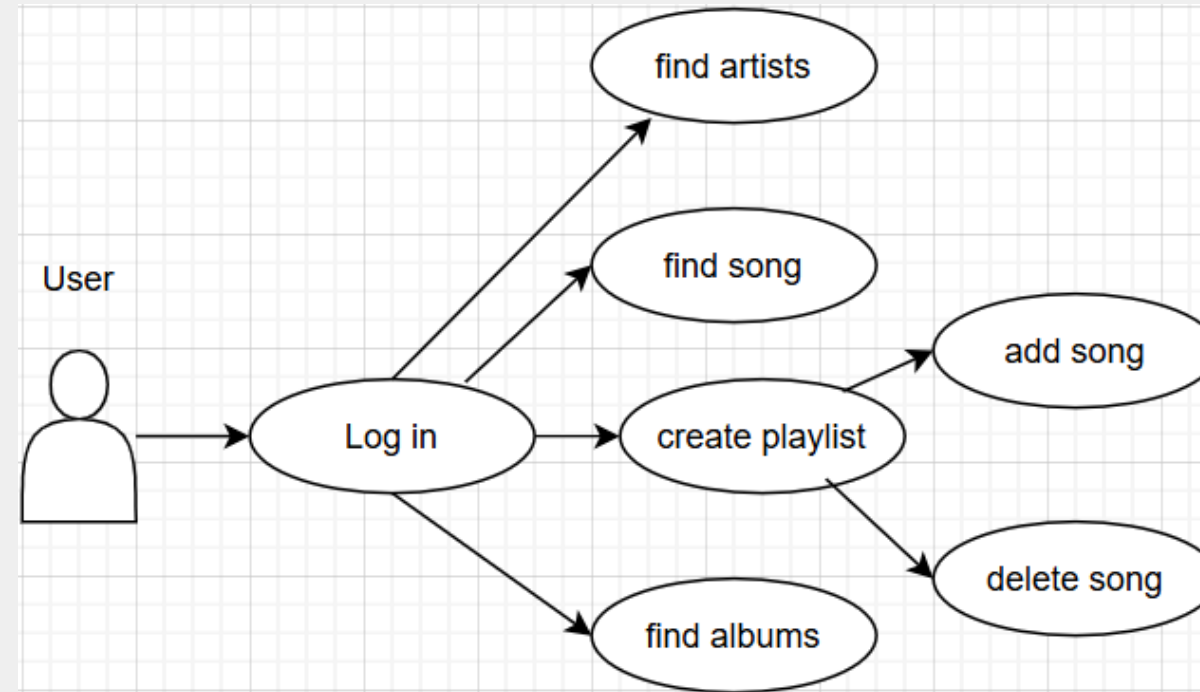


Analysis of existing systems

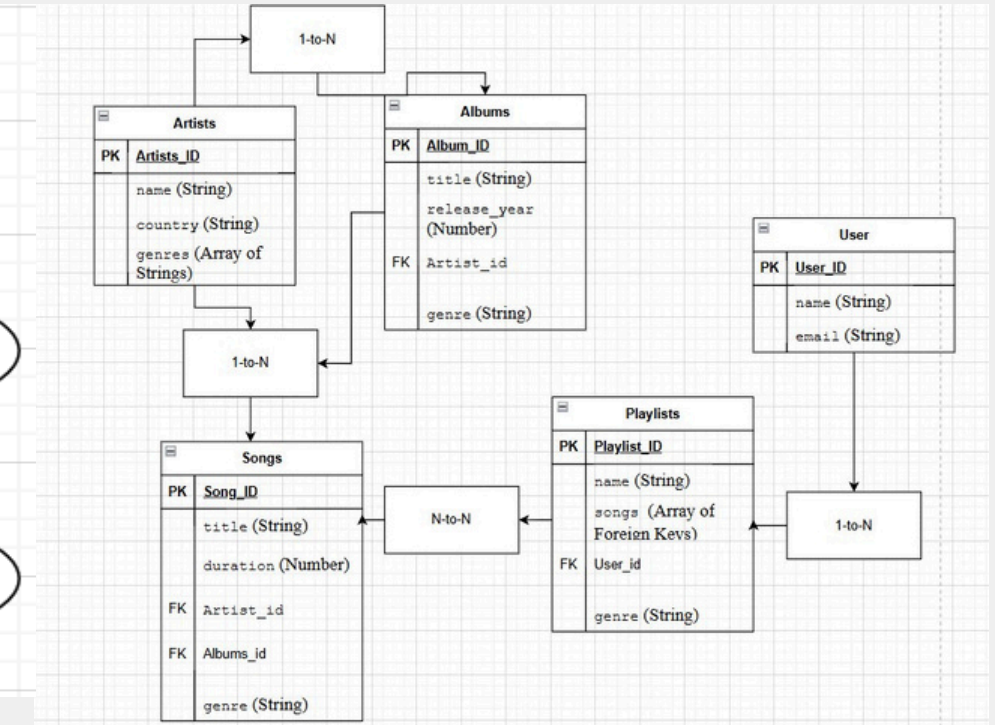


05 Diagrams

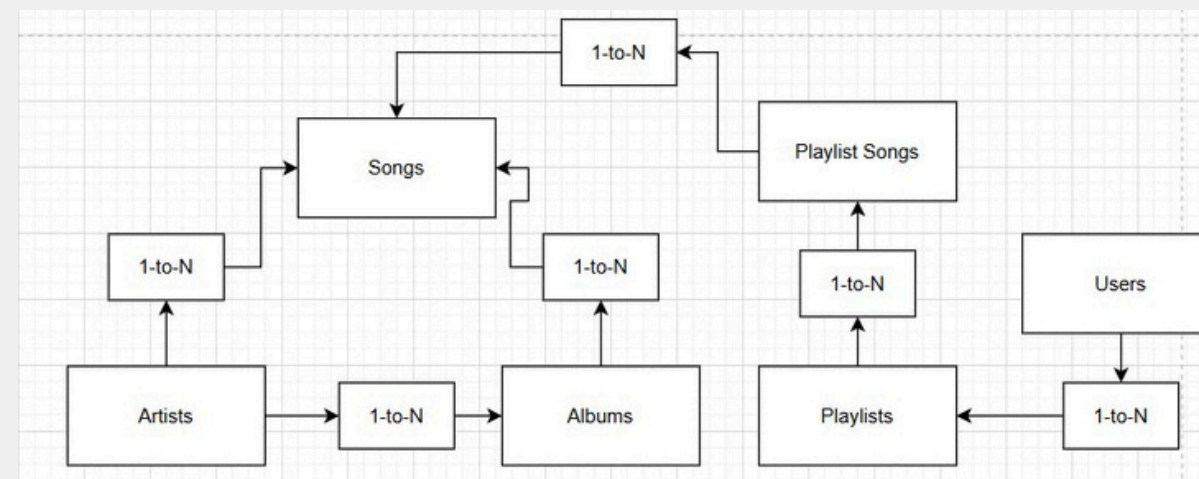
■ User case



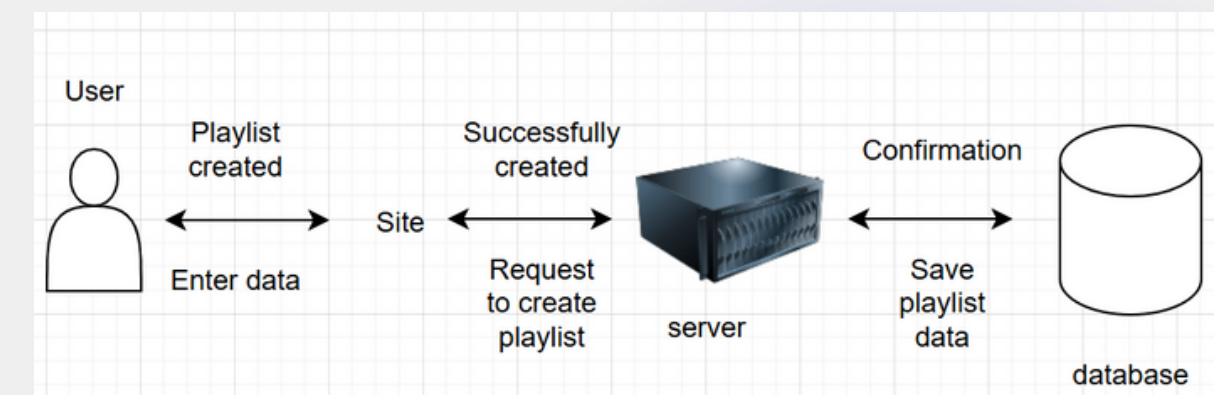
■ ERD



■ CRD

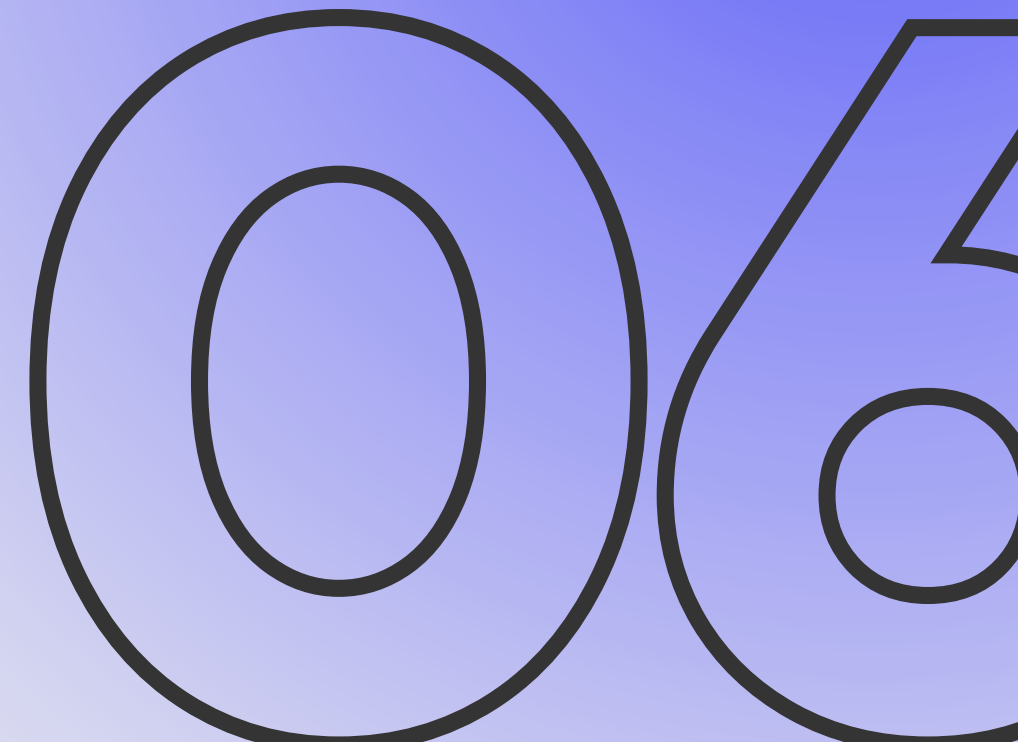


■ Sequence



Creating databases(music_library) and collections(songs, artists, albums, playlists, users):

```
> use music_library
< switched to db music_library
> db.createCollection("songs");
db.createCollection("artists");
db.createCollection("albums");
db.createCollection("users");
< { ok: 1 }
> db.createCollection("playlists")
< { ok: 1 }
```



07

Inserting data to collections:

```
> db.artists.insertMany([
  { _id: 1, name: "Michael Jackson", country: "USA", genres: ["Pop", "Soul"] },
  { _id: 2, name: "Queen", country: "UK", genres: ["Rock"] },
  { _id: 3, name: "Adele", country: "UK", genres: ["Pop", "Soul"] }
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': 1,
    '1': 2,
    '2': 3
  }
}
music_library>
```

```
> db.songs.insertMany([
  { _id: 101, title: "Billie Jean", duration: 292, genre: "Pop", album_id: 1, artist_id: 1 },
  { _id: 102, title: "Bohemian Rhapsody", duration: 355, genre: "Rock", album_id: 2, artist_id: 2 },
  { _id: 103, title: "Rolling in the Deep", duration: 228, genre: "Pop", album_id: 3, artist_id: 3 }
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': 101,
    '1': 102,
    '2': 103
  }
}
music_library>
```

```
> db.albums.insertMany([
  { _id: 1, title: "Thriller", release_year: 1982, artist_id: 1, genre: "Pop" },
  { _id: 2, title: "A Night at the Opera", release_year: 1975, artist_id: 2, genre: "Rock" },
  { _id: 3, title: "21", release_year: 2011, artist_id: 3, genre: "Pop" }
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': 1,
    '1': 2,
    '2': 3
  }
}
music_library>|
```

CRUD – operations:

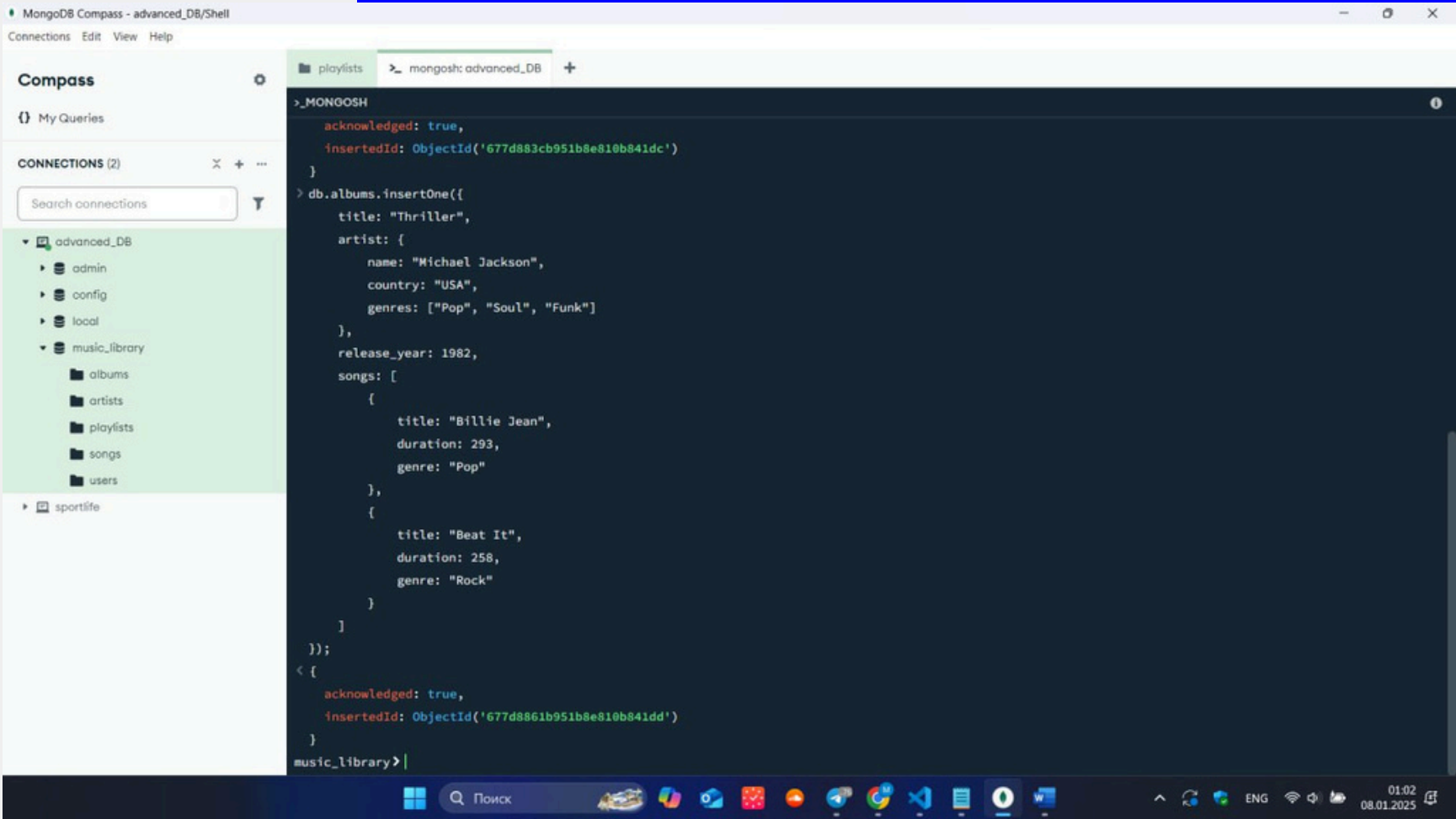
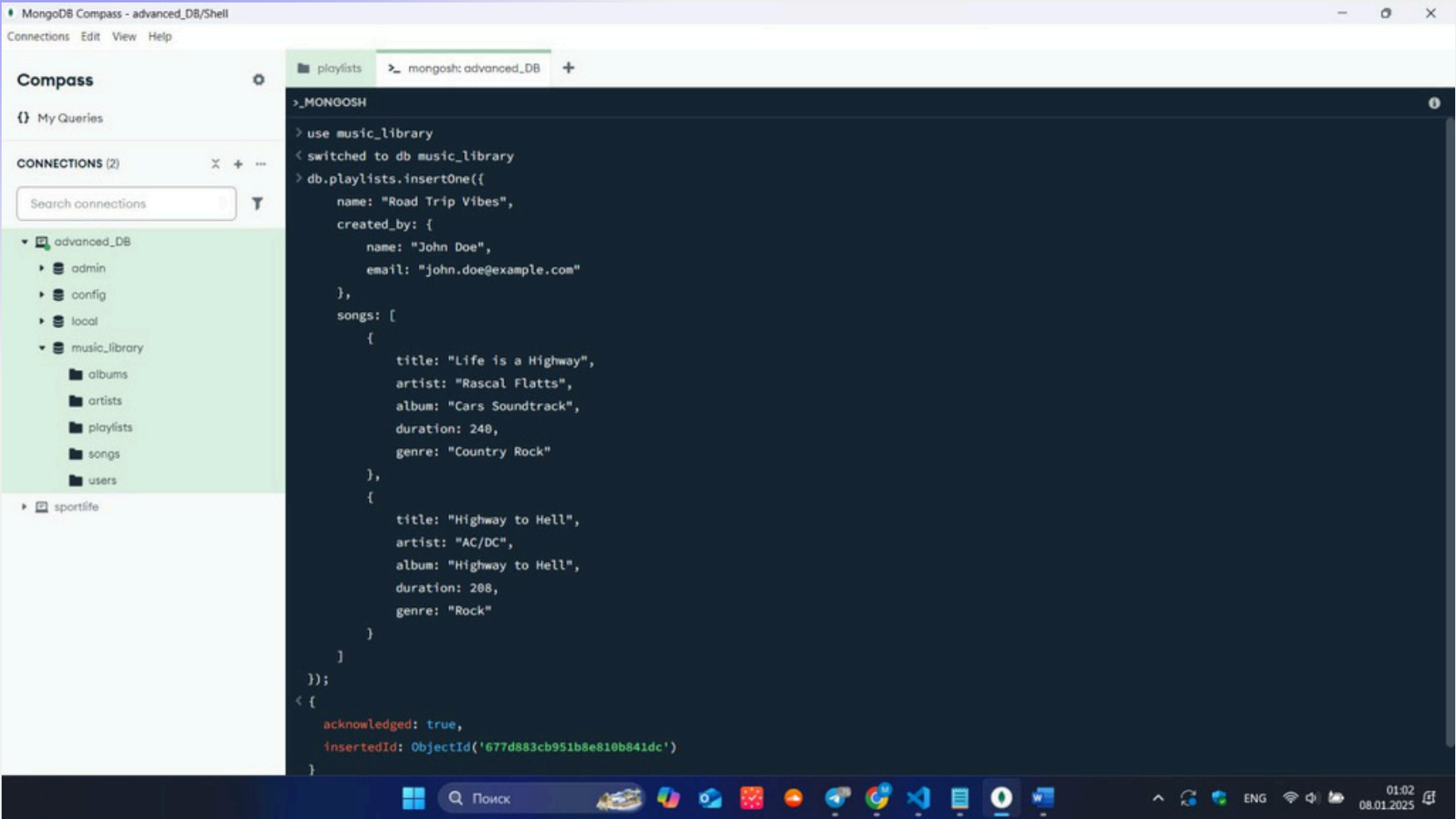
```
> db.songs.insertOne({
  title: "Smooth Criminal",
  duration: 256,
  genre: "Pop",
  album_name: "Thriller",
  artist_name: "Michael Jackson"
});
< {
  acknowledged: true,
  insertedId: ObjectId('677d51acea3ab93c34a3f1fb')
}
music_library>
```

```
> db.songs.find({ artist_name: "Michael Jackson" });
< {
  _id: ObjectId('677d5160ea3ab93c34a3f1f4'),
  title: 'Billie Jean',
  duration: 292,
  genre: 'Pop',
  album_name: 'Thriller',
  artist_name: 'Michael Jackson'
}
{
  _id: ObjectId('677d51acea3ab93c34a3f1fb'),
  title: 'Smooth Criminal',
  duration: 256,
  genre: 'Pop',
  album_name: 'Thriller',
  artist_name: 'Michael Jackson'
}
music_library>
```

```
> db.songs.deleteOne({ title: "Smooth Criminal" });
< {
  acknowledged: true,
  deletedCount: 1
}
music_library>
```

```
> db.albums.updateOne(
  { title: "Thriller" },
  { $set: { genre: "Pop/Rock" } }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
music_library>
```


CRUD – operations:



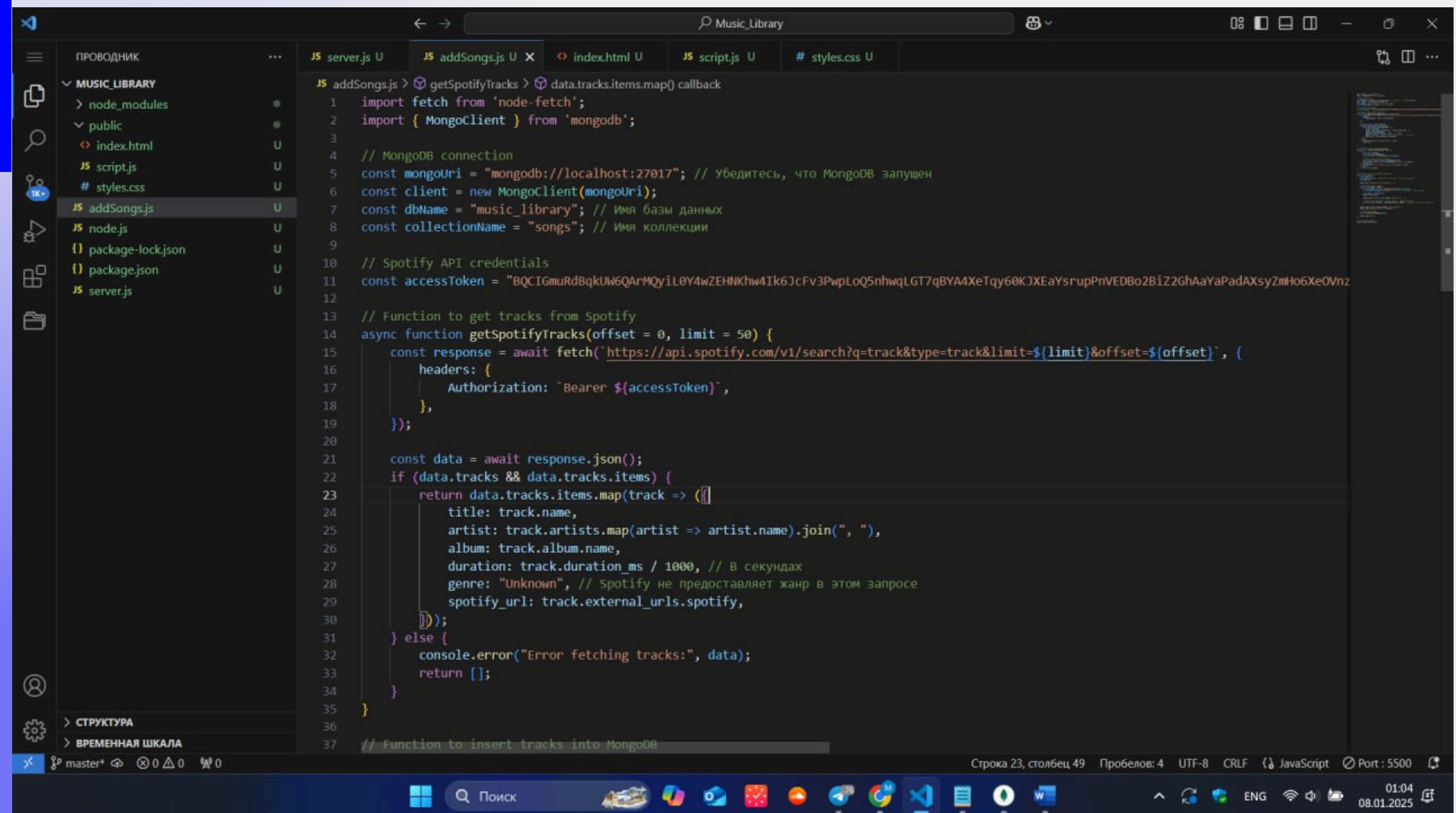
Music Library

-

Project

10

adding 1000 doc by using Api



```
JS server.js U JS addSongs.js U X index.html U JS script.js U # styles.css U

PROВOДНИК
MUSIC_LIBRARY
  > node_modules
  > public
  > index.html
  JS script.js
  # styles.css
  JS addSongs.js
  JS node.js
  {} package-lock.json
  {} package.json
  JS server.js

СТРУКТУРА
ВРЕМЕННАЯ ШКАЛА

JS addSongs.js > getSpotifyTracks > data.tracks.items.map() callback
1 import fetch from 'node-fetch';
2 import { MongoClient } from 'mongodb';
3
4 // MongoDB connection
5 const mongoUri = "mongodb://localhost:27017"; // Убедитесь, что MongoDB запущен
6 const client = new MongoClient(mongoUri);
7 const dbName = "music_library"; // Имя базы данных
8 const collectionName = "songs"; // Имя коллекции
9
10 // Spotify API credentials
11 const accessToken = "BQCIGmuRd8qkUW6QArMQyiL0Y4wZEHNKhW4Ik6JcFv3PwpLoQ5nhwqLGT7qBYA4XeTqy60KJXEaYsrupPnVED8o2BiZ2GhAaYaPadAXsyZmHo6Xe0Vnz";
12
13 // Function to get tracks from Spotify
14 async function getSpotifyTracks(offset = 0, limit = 50) {
15   const response = await fetch(`https://api.spotify.com/v1/search?q=track&type=track&limit=${limit}&offset=${offset}`, {
16     headers: {
17       Authorization: `Bearer ${accessToken}`,
18     },
19   });
20
21   const data = await response.json();
22   if (data.tracks && data.tracks.items) {
23     return data.tracks.items.map(track => ({
24       title: track.name,
25       artist: track.artists.map(artist => artist.name).join(", "),
26       album: track.album.name,
27       duration: track.duration_ms / 1000, // в секундах
28       genre: "Unknown", // Spotify не предоставляет жанр в этом запросе
29       spotify_url: track.external_urls.spotify,
30     }));
31   } else {
32     console.error("Error fetching tracks:", data);
33     return [];
34   }
35 }
36
37 // Function to insert tracks into MongoDB
```

Site:

Music Library

Songs

Artists

Albums

Playlists

Music Library

Songs

Artists

Albums

Playlists

Songs

Search for a song

Find

Billie Jean by Unknown Artist (Album: Unknown Album)

Bohemian Rhapsody by Unknown Artist (Album: Unknown Album)

Rolling in the Deep by Unknown Artist (Album: Unknown Album)

Track 03 by Husky, maslo chernogo tmina, Andy Panda (Album: Y)

track 07 by title it yourself (Album: Possibly in Michigan)

Track in Time - Piano Version by Dennis Kuo (Album: Track in Time (Piano Version))

Mobius (Bonus Track) by Alexey Ormelchuk (Album: Metro: Last Light (Official Soundtrack))

Dope Track by Kordhell (Album: Beat Tape 3)

Анечка (Bonus track) by ANNAASTI (Album: Феникс)

Songs

Billie Jean

Find

Billie Jean by Unknown Artist (Album: Unknown Album)

Search Results

Billie Jean by Unknown Artist (Album: Unknown Album)

Back

Presentation 2025

-

Music Library

Site:

Music Library

Songs

Artists

Albums

Playlists

Artists

Search for an artist

Find

Michael Jackson (USA)

Queen (UK)

Adele (UK)

Music Library

Songs

Artists

Albums

Playlists

Albums

Search for an album

Find

Thriller by Unknown Artist (Year: 1982)

A Night at the Opera by Unknown Artist (Year: 1975)

21 by Unknown Artist (Year: 2011)

Thriller by Unknown Artist (Year: 1982)

12

Site:

Playlists

Create New Playlist

My Favorites

Show Songs

Add Song

Delete Playlist

Bohemian Rhapsody

Remove

Workout Hits

Show Songs

Add Song

Delete Playlist

Rolling in the Deep

Remove

Vibe songs

Show Songs

Add Song

Delete Playlist

No songs in this playlist.

Music Library

Songs

Artists

Albums

Playlists

Playlists

Create New Playlist

My Favorites

Show Songs

Add Song

Delete Playlist

Workout Hits

Show Songs

Add Song

Delete Playlist

Vibe songs

Show Songs

Add Song

Delete Playlist

Gym

Show Songs

Add Song

Delete Playlist

Hiking

Show Songs

Add Song

Delete Playlist

Doping Track

Show Songs

Add Song

Delete Playlist

Road Trip Vibes

Show Songs

Add Song

Delete Playlist

13

Site:

Create New Playlist

Playlist Name

Create Playlist

Add Song to Playlist

Enter Song Name: e.g., Billie Jean

Add Song

My Favorites

Show Songs

Add Song

Delete Playlist

Bohemian Rhapsody

Remove

Workout Hits

Show Songs

Add Song

Delete Playlist

Rolling in the Deep

Remove

Vibe songs

Show Songs

Add Song

Delete Playlist

No songs in this playlist.

14

Thank You

for your attention