Department of Computer Science and Engineering

Chitkara University

Project report on ‘GrooveUp’ (a music platform)

Submitted by :-

Name - Yashasvi

Roll no - 2110991582

Group - G7 (5th Sem)

[Batch 2021-25]

Introduction :

This project presents a full stack functional music platform ‘GrooveUp Music’ developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). The application facilitates user registration, music exploration, playlist creation, and playback control, offering a robust and user-friendly music streaming experience, using backend APIs which are implemented from scratch.

Technologies Used:

Frontend: React.js with Tailwind.css for a user friendly and intuitive interface.

Backend: Express.js for API development and Node.js for server-side operations

Database: MongoDB for storing user data, playlists, and song information, Cloudinary for song uploading.

Features:

1) **User Authentication with Extended Session Duration**: The application prioritizes user convenience by implementing a token-based authentication system that maintains user sessions for 30 days. This feature significantly reduces the frequency of logins, enhancing user experience and engagement.

2) **Track Uploading**: Users can seamlessly upload their own music tracks to the platform, accompanied by visually appealing thumbnails that enhance

discoverability. This feature empowers creators to share their music with a wider audience and enriches the platform's content diversity.

3) **Playback Controls**: Play, pause, skip, and shuffle songs within the application. with a song control bar at the bottom of the screen.

4) **Extensive Search Capabilities**: The application offers comprehensive search functionality where users can efficiently pinpoint specific songs by their titles, ensuring effortless navigation and retrieval of desired music. from all over the songs database (including songs created by other artists)

5) **Playlist Creation and Management**: Users enjoy the freedom to curate personalized playlists that reflect their unique tastes and moods. The application supports the seamless addition of any song from the database to playlists, fostering music discovery and customization, which helps them combine songs of similar taste and type for different moods and occasions.

Challenges and Solutions:

1. **Authentication Challenges**:

    - Secure Token-Based Approach: Employed Passport.js and JSON Web Tokens (JWTs) to effectively authenticate users and safeguard their accounts.

- Token Generation and Validation: Upon successful login, a unique JWT containing user information and an expiration time is generated and stored securely on the client-side. This token is sent with each subsequent API request, ensuring user identity and access control. (token is saved in cookies)
- Password Hashing : Password of user is encrypted and converted to a hashcode using ‘bcrypt’ package of node. and the hashed password is stored in MongoDB.

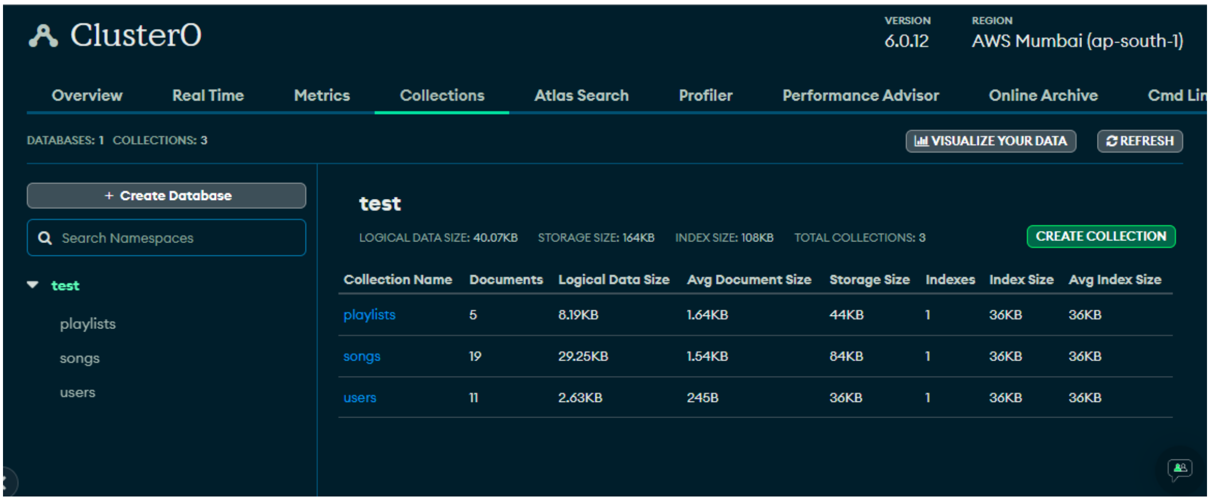
2. Playback Control Challenges:

- Seamless Playback Across Routes: Addressed the challenge of maintaining uninterrupted audio playback when users navigate between different application routes by implementing effective state management techniques: Centralized State with React Context: Created a React Context to store and manage playback-related data, including the current song, playback status, and shuffle mode, making it accessible throughout the application.
- State Updates and UI Synchronization: Ensured that UI elements accurately reflect the current playback state by subscribing to context changes and updating components accordingly.
- Route Changes and Audio Continuity: Prevented audio interruption when users switch routes , by making a shared component for the song control bar.

Demystifying GrooveUp's Database: A Peek into MongoDB

In GrooveUp's musical database, three key collections take center stage:

- Users
- Playlists
- Songs



DATABASES: 1 COLLECTIONS: 3

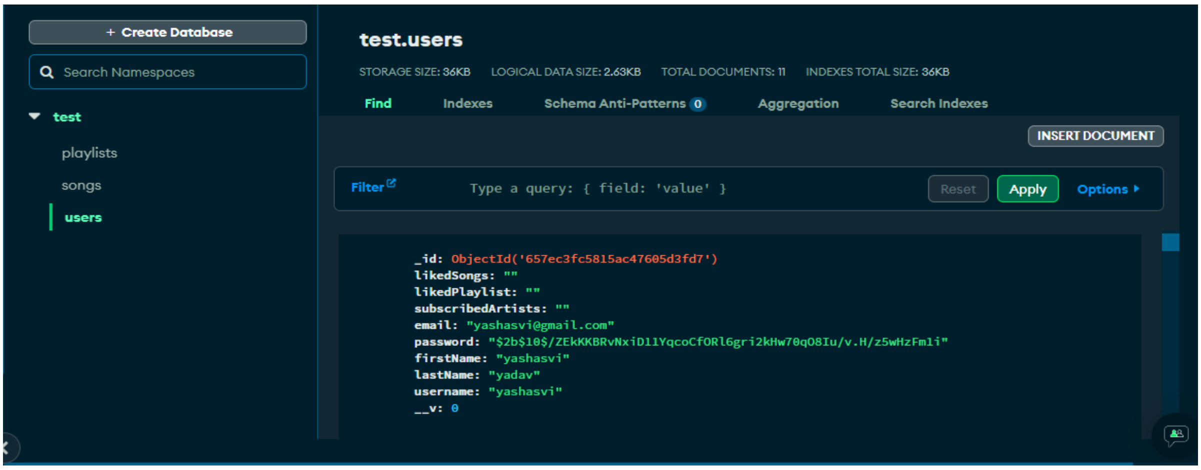
test

LOGICAL DATA SIZE: 40.07KB STORAGE SIZE: 164KB INDEX SIZE: 108KB TOTAL COLLECTIONS: 3

Collection Name	Documents	Logical Data Size	Avg Document Size	Storage Size	Indexes	Index Size	Avg Index Size
playlists	5	8.19KB	1.64KB	44KB	1	36KB	36KB
songs	19	29.25KB	1.54KB	84KB	1	36KB	36KB
users	11	2.63KB	245B	36KB	1	36KB	36KB

1. Users:

Each document in this collection captures a unique listener's identity (assigned by mongoDb i.e _id), storing essential details like username, email, playlist owner, its like a personal details card, granting access to the platform.

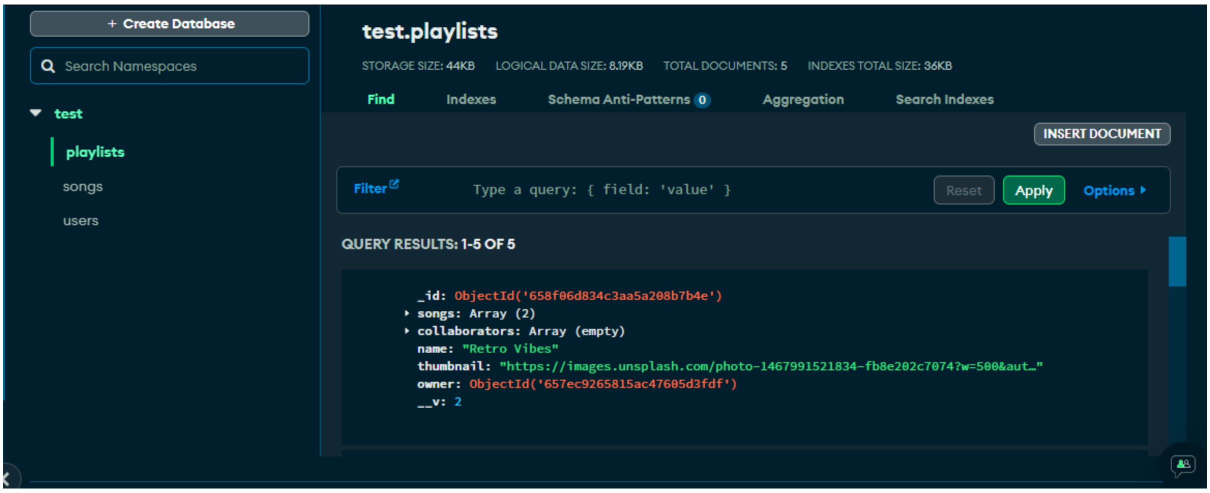


```
_id: ObjectId('657ec3fc5815ac47605d3fd7')
likedSongs: ""
likedPlaylist: ""
subscribedArtists: ""
email: "yashasvi@gmail.com"
password: "$2b$10$/ZEkKKBRvNxiD11YqcoCfORl6gri2kHw70qO8Iu/v.H/z5wHzFm1i"
firstName: "yashasvi"
lastName: "yadav"
username: "yashasvi"
__v: 0
```

2. Playlists:

Each playlist object will have its details like unique id (assigned by mongoDb),

These documents curate the soundtrack of each user's life, holding arrays of song references that reflect moods, moments, and memories. User creates a playlist and in that playlist's object key 'songs' will have value as array of songs linked to that playlist.

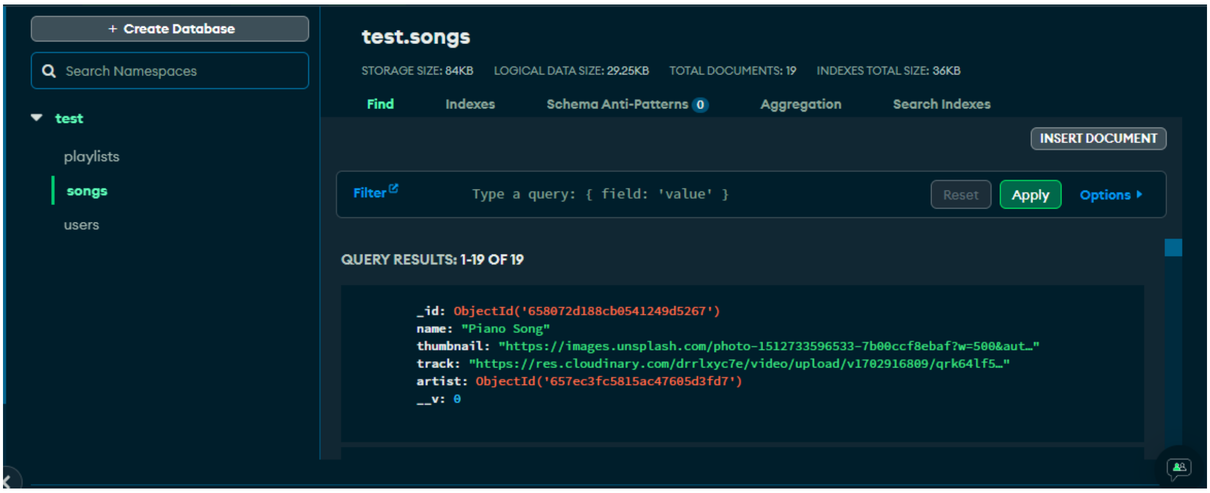


QUERY RESULTS: 1-5 OF 5

```
  _id: ObjectId('658f06d834c3aa5a208b7b4e')
▸ songs: Array (2)
▸ collaborators: Array (empty)
  name: "Retro Vibes"
  thumbnail: "https://images.unsplash.com/photo-1467991521834-fb8e202c7074?w=500&aut…"
  owner: ObjectId('657ec9265815ac47605d3fdf')
  __v: 2
```

3. Songs

The heart of the collection, each song document embodies a musical track, storing its unique id, title, artist(who uploaded that song) , thumbnail. It's a digital embodiment of the creative spark that ignites our auditory senses.



STORAGE SIZE: 84KB LOGICAL DATA SIZE: 29.25KB TOTAL DOCUMENTS: 19 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-19 OF 19

```
_id: ObjectId('658072d188cb0541249d5267')
name: "Piano Song"
thumbnail: "https://images.unsplash.com/photo-1512733596533-7b00ccf8ebaf?w=500&aut…"
track: "https://res.cloudinary.com/drrlxyc7e/video/upload/v1702916809/qrk64lf5…"
artist: ObjectId('657ec3fc5815ac47605d3fd7')
__v: 0
```

Future Considerations:

1. Liked Songs: Enables users to have all of his liked songs saved in a different section for listening on repeat.
2. Progress Bar : Enables users to navigate within a song by dragging a progress bar or clicking on specific points in the timeline.
3. Volume Control: Allows users to adjust the audio output volume to their preference.
4. Playback Speed Adjustment: Provides options to slow down or speed up playback, catering to different listening preferences and learning needs.
5. Listening History : on the Home page all the tracks that user recently listened to will be displayed for better user experience.

Important Links of project :-

GitHub Repository:

I have diligently maintained the progress building this project from scratch by regularly committing changes to the GitHub repository. The codebase is hosted on GitHub at https://github.com/yashasviyadav1/grooveup-music This version control allows for easy collaboration, tracking of changes, and provides a transparent record of the development process.

Deployment on Vercel:

The MERN project, named GrooveUp Music, has been successfully deployed for public access. The live version is hosted on Vercel, providing a seamless and efficient user experience. You can explore the deployed application at this link :

https://grooveup-music.vercel.app/

This deployment not only serves as a demonstration of the project's functionality but also ensures accessibility for users to interact with the application in a real-world environment.

Thank You for reading.