# Axis School Of Fintech

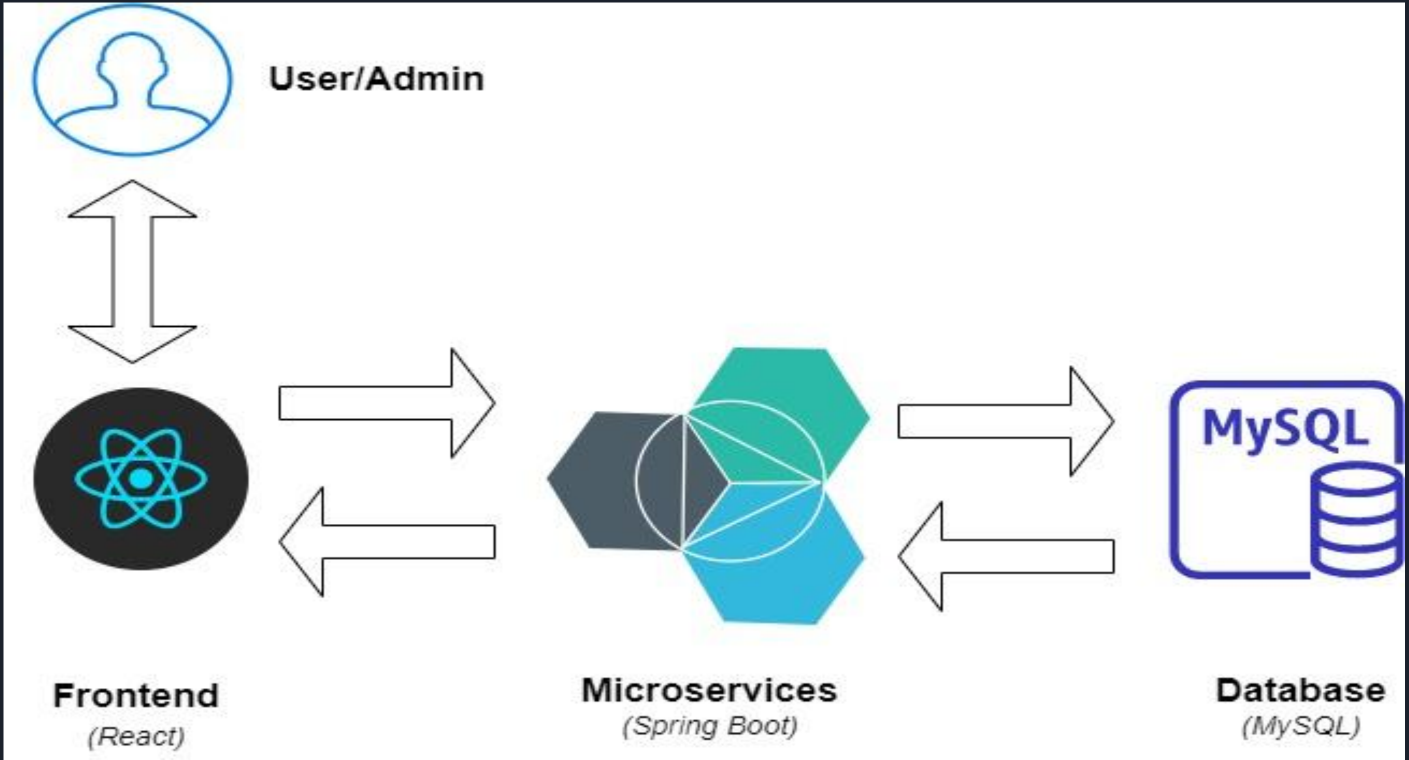
# Music Application

Chappidi Sai Chaitanya Reddy

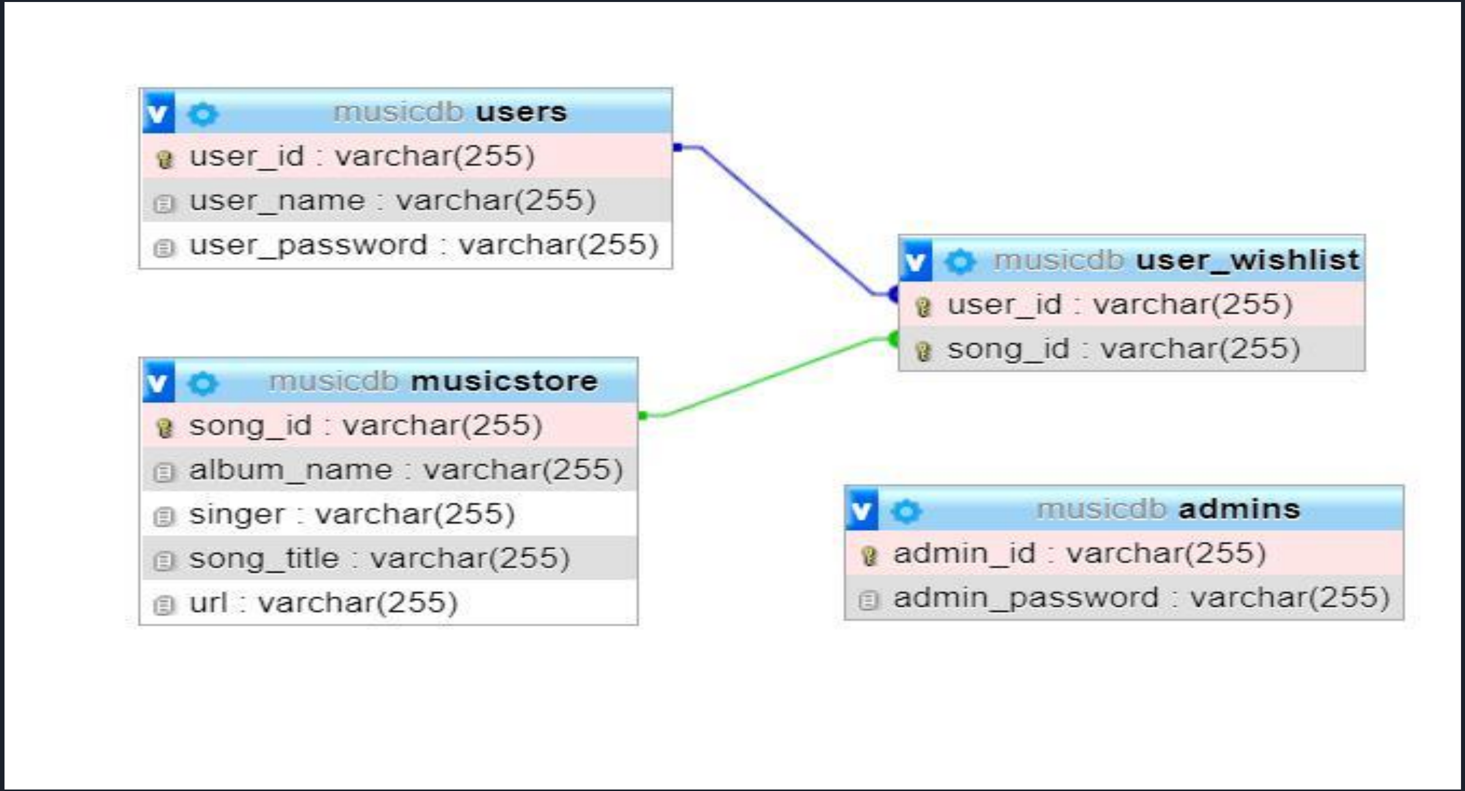
# About

- ❑ Developed a Music Application to perform CRUD operations performed on MySQL database.
- ❑ In this application there are two roles:  Admin and User.
- ❑ Admin is responsible for adding songs to the database.
- ❑ User can play the songs available in the database and add songs to his wishlist.
- ❑ Main elements of the app are:
  - ★ Discovery: Songs can be searched based on album name, singer and title.
  - ★ Personalization: Songs can be added to the wishlist.

# Overall Architecture



# Database Schema



musicdb **users**
- user_id : varchar(255)
- user_name : varchar(255)
- user_password : varchar(255)

musicdb **user_wishlist**
- user_id : varchar(255)
- song_id : varchar(255)

musicdb **musicstore**
- song_id : varchar(255)
- album_name : varchar(255)
- singer : varchar(255)
- song_title : varchar(255)
- url : varchar(255)

musicdb **admins**
- admin_id : varchar(255)
- admin_password : varchar(255)

# Admin Functionalities

❑ Admin can login with his credentials.

❑ Admin can update his/her password.

❑ Admin can see all the songs present in the database.

❑ Admin can search song based on song title.

❑ Admin can search song based on album name.

❑ Admin can search song based on singer.

❑ Admin and add songs to database.

❑ Admin can update the song details.

❑ Admin can delete the song from the database.

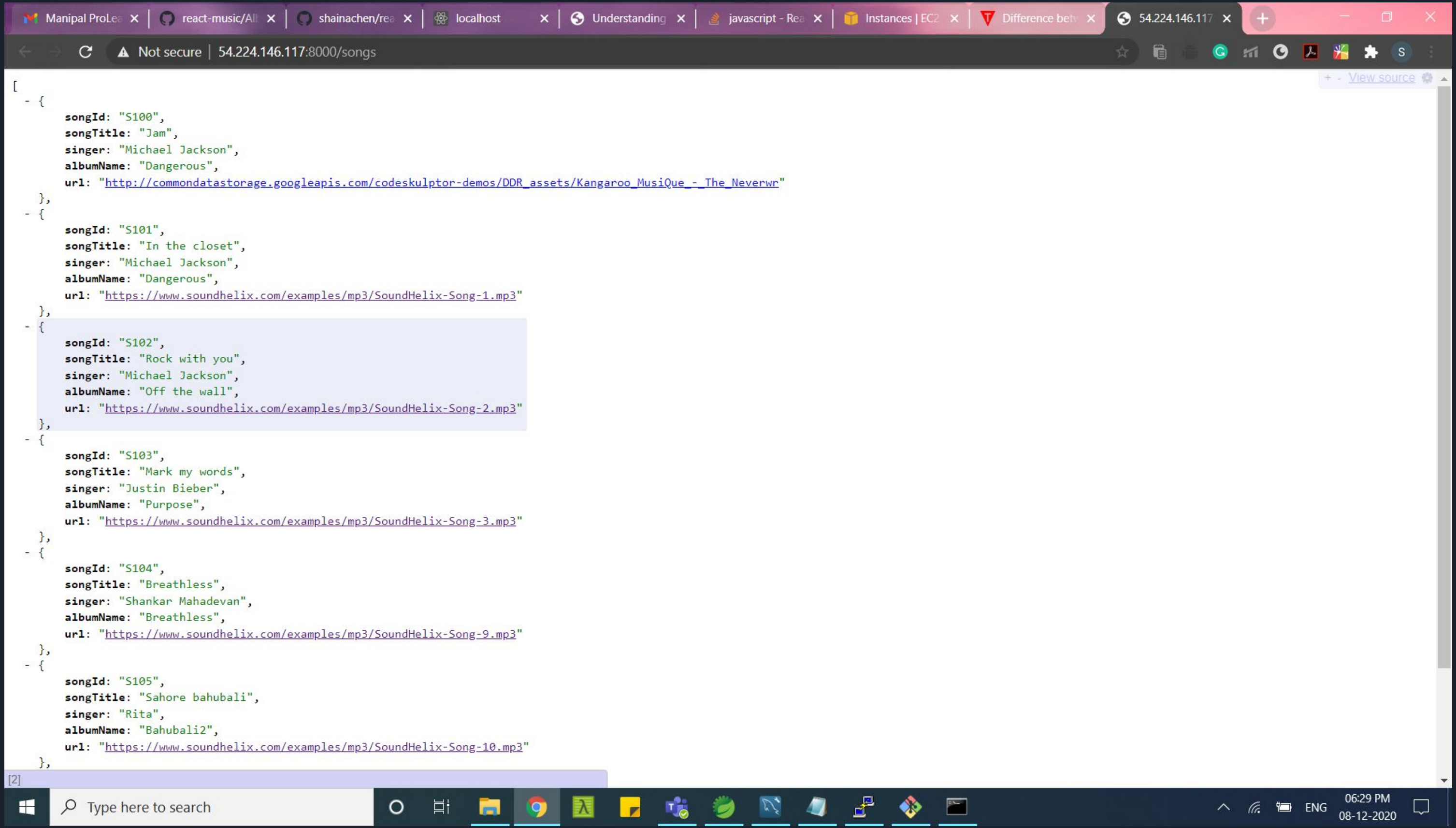
# Services provided to User

- ❑ User can register.
- ❑ User can login with his credentials.
- ❑ User can see all the songs present in the database.
- ❑ User can search song based on song title.
- ❑ User can search song based on album name.
- ❑ User can search song based on singer.
- ❑ User can add song to his wishlist.
- ❑ User can delete song from his wishlist.
- ❑ User can see all the songs present in his wishlist.
- ❑ User can search song based on song title in his wishlist.
- ❑ User can search song based on album name in his wishlist.
- ❑ User can update his details(username and password).
- ❑ User can delete his account.

# Technologies Used

| | |
|---|---|
| ❏ Language : | Java, Kotlin, React |
| ❏ Development Framework : | Spring Boot |
| ❏ API Gateway: | Zuul |
| ❏ Load Balancer: | Eureka |
| ❏ Testing Framework : | Mockito, Enzyme, Selenium |
| ❏ Documentation Framework : | Swagger |
| ❏ Build Tools : | Maven |
| ❏ Database : | MySQL |
| ❏ Static Code Analysis: | SonarQube |
| ❏ IDE : | IntelliJ, STS, Eclipse |
| ❏ Deployment: | AWS RDS |

# AWS RDS



```
[
  - {
        songId: "S100",
        songTitle: "Jam",
        singer: "Michael Jackson",
        albumName: "Dangerous",
        url: "http://commondatastorage.googleapis.com/codeskulptor-demos/DDR_assets/Kangaroo_MusiQue_-_The_Neverwr"
    },
  - {
        songId: "S101",
        songTitle: "In the closet",
        singer: "Michael Jackson",
        albumName: "Dangerous",
        url: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-1.mp3"
    },
  - {
        songId: "S102",
        songTitle: "Rock with you",
        singer: "Michael Jackson",
        albumName: "Off the wall",
        url: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-2.mp3"
    },
  - {
        songId: "S103",
        songTitle: "Mark my words",
        singer: "Justin Bieber",
        albumName: "Purpose",
        url: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-3.mp3"
    },
  - {
        songId: "S104",
        songTitle: "Breathless",
        singer: "Shankar Mahadevan",
        albumName: "Breathless",
        url: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-9.mp3"
    },
  - {
        songId: "S105",
        songTitle: "Sahore bahubali",
        singer: "Rita",
        albumName: "Bahubali2",
        url: "https://www.soundhelix.com/examples/mp3/SoundHelix-Song-10.mp3"
    },
```

# Future Implementations

- ❏ Songs can be searched based on language.
- ❏ Notify user if any song is added.
- ❏ Implement Spring security.
- ❏ Application Deployment using Docker, Kubernetes.

# Thank You