



YouRMusic

12.08.2019

Khoa Pham, Tom Zheng & Johnny Phenglavong
Team 20

CS 157A - Professor **Mike Wu**
San Jose State University
Project Final Report

Table of content:

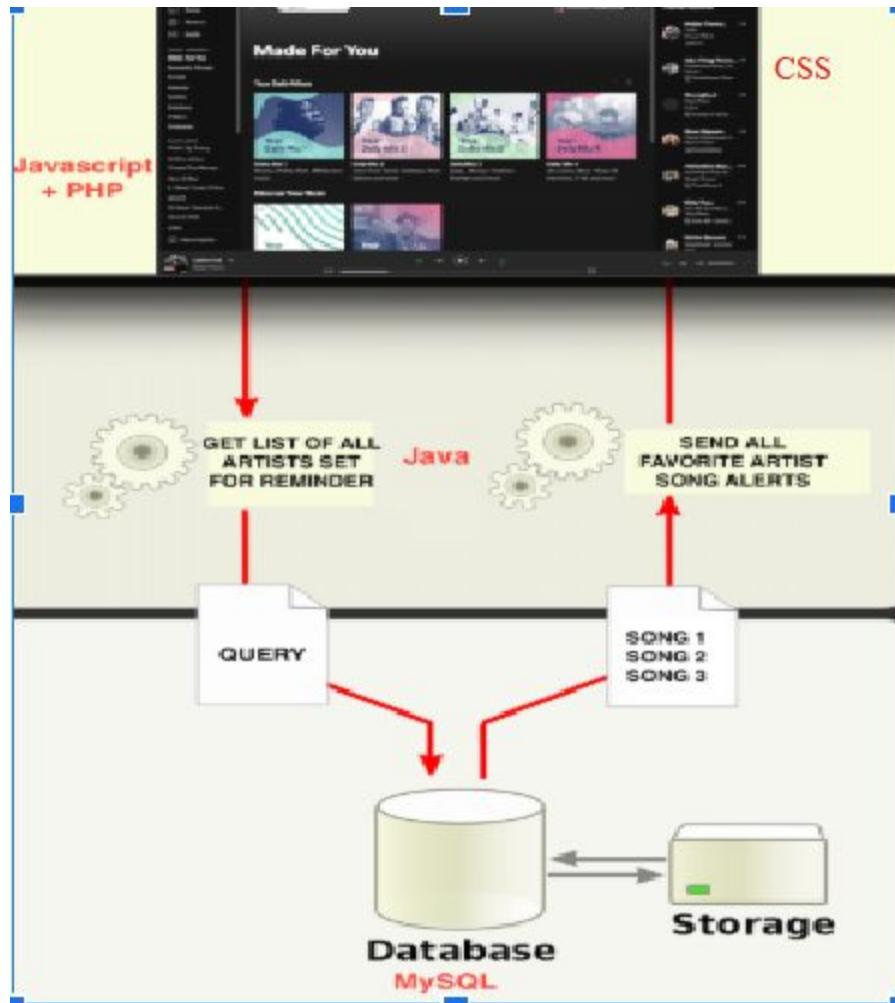
| | |
|---|---------|
| 1. Project Requirement | page 2 |
| a. Project Description | page 2 |
| b. System Environment | page 3 |
| c. Functional Requirements | Page 5 |
| d. Non-functional Issues | page 6 |
| 2. Project Design | |
| a. E/R Diagram | page 10 |
| b. Schemas of E/R diagram | page 11 |
| 3. Project Installation & Cloning from Github | page 20 |
| 4. Implementation | page 30 |
| 5. JDBC | page 71 |
| 6. Conclusion | |
| a. Lessons Learned from Team 20 | page 93 |
| b. Future Improvements | page 94 |

Project Requirement

Project Description

- Our group will be implementing a website for storing audio files and keeping track of the latest music releases. Our stakeholders will be music listeners and anyone involved in the entertainment industry. The problem with the current music industry is that there are few options to-go-to when it comes to creating your own playlist and streaming it while you go. Apple Music has a cluttered and complex user interface that isn't so intuitive, while Spotify is one of the top streaming services that we would like to improve upon.
- This project will be important in retaining customer attention and awareness of the latest songs. In addition, the entertainment industry can profit from this project as more product awareness is always a good thing. In this project, we will benefit the users by having the ability for the user to add, delete, update their account information and the songs we have. Besides, we have a service that lets the user upload and plays the music they want in case we don't have that music yet.

System Environment



In this project, we will use:

1. **HTML CSS**

- a. Our team use HTML and CSS to make our front end web view looks well-arranged and good-looking for the users to use

2. **JavaScript + jQuery**

- a. jQuery is a lightweight JavaScript library that can help users reduce the code but still maintain the efficiency of the program. JavaScript will be applied to help with View site for the user to see.

3. PHP + MySQL + jQuery

- a. We will use PHP to write SQL code to interact with MySQL to manage the database
- b. PHP also will interact with the front end html code when pulling information from the database.

4. VUE.JS

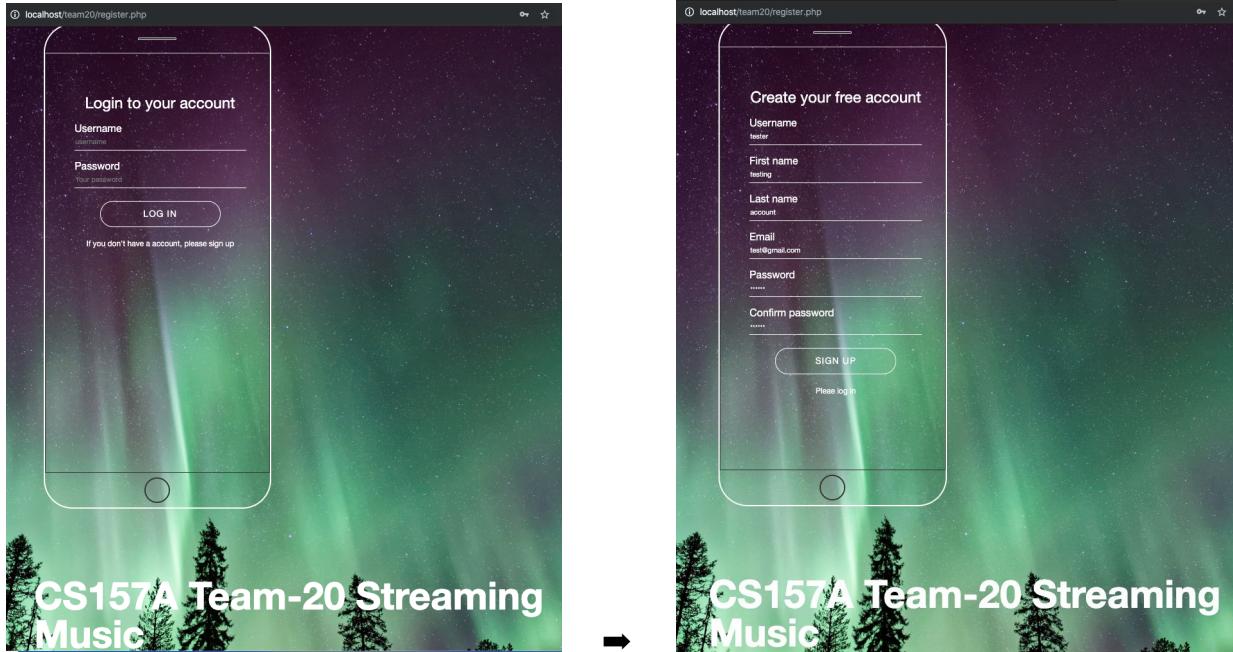
5. JDBC (Java + SQL + HTML CSS)

- a. JDBC is good for academic purpose so we use JDBC to work with our back-end program. Basically, it will connect with MySQL and manipulate Insert, Delete, Update, and Sort the data. With the help of HTML and CSS, again, we can show the database in our website project.

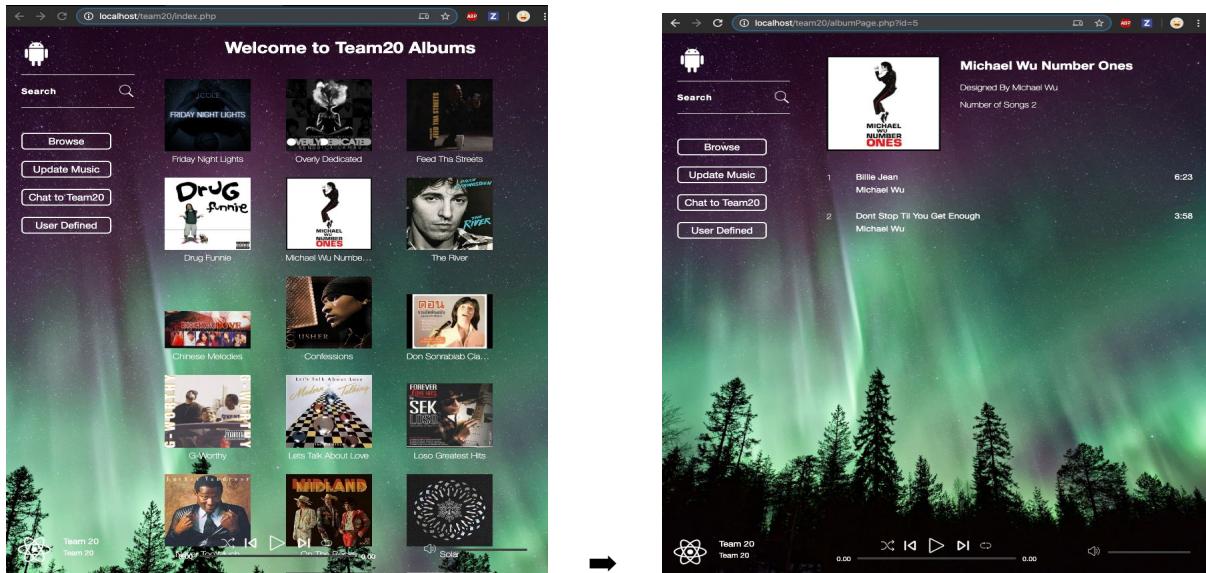
Functional Requirements

Team 20 Project Functionality Demo

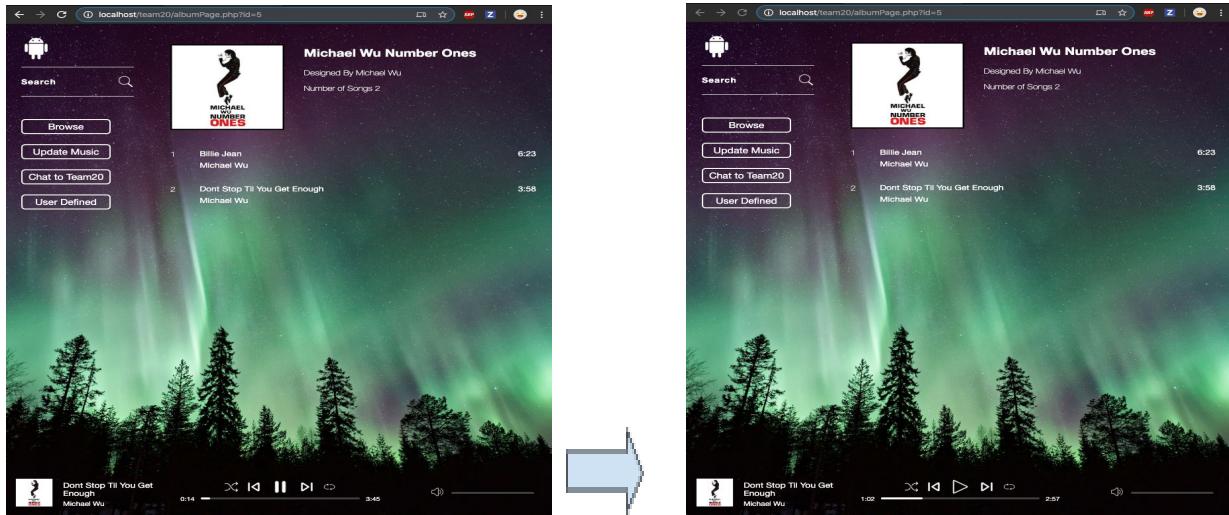
1. Users will be able to register for an account via register.php. They can do so by clicking sign up and then filling out the account information.



2. Users will be able to select various albums in the main homepage index.php. They will be able to click on the album artwork and view a list of playable songs.



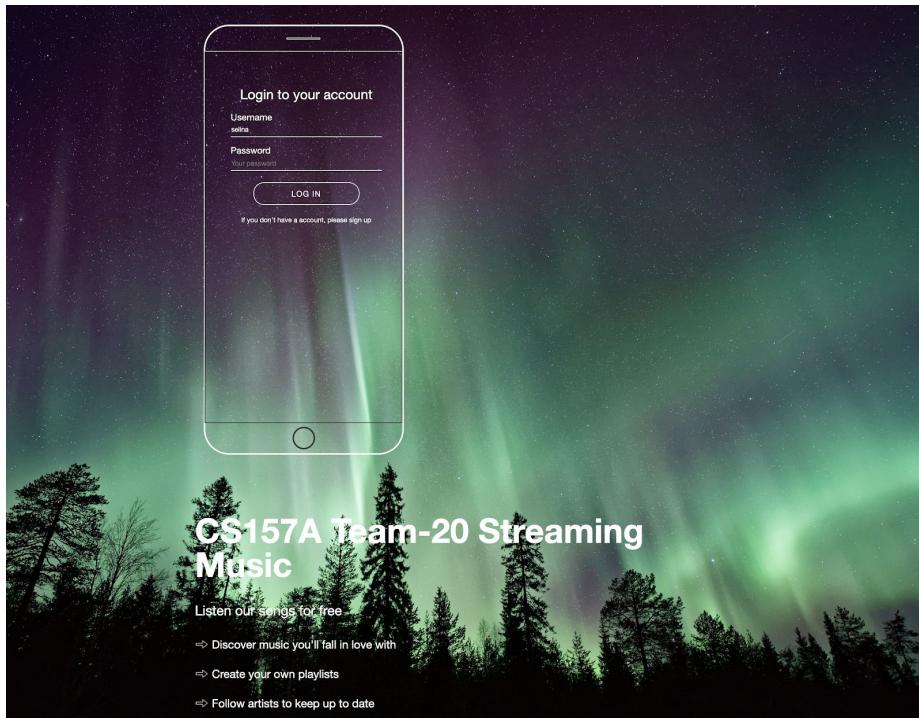
3. Users will be able to play a song and pause it as well. They can skip throughout the track duration to any time they desire.



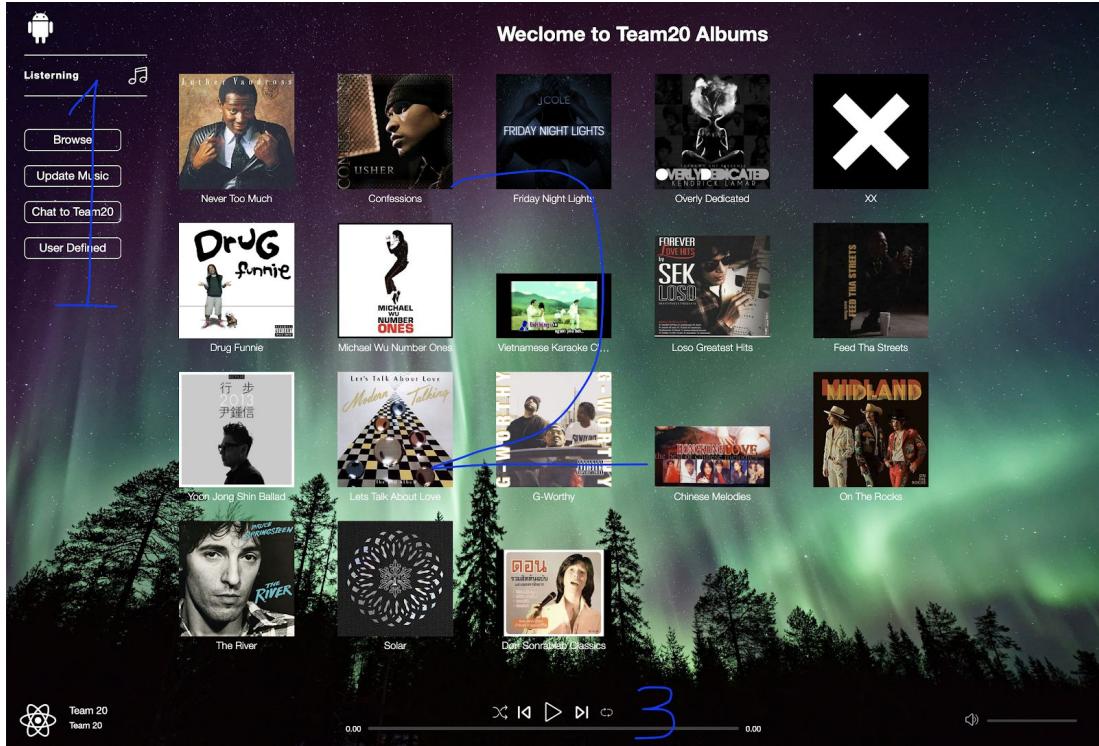
Non-Functional Requirements

Graphical User Interface

Our Graphical User Interface or GUI for the register page consisted of a picture background, mobile phone overlay for signup/login functionality and text as shown in the photo below.

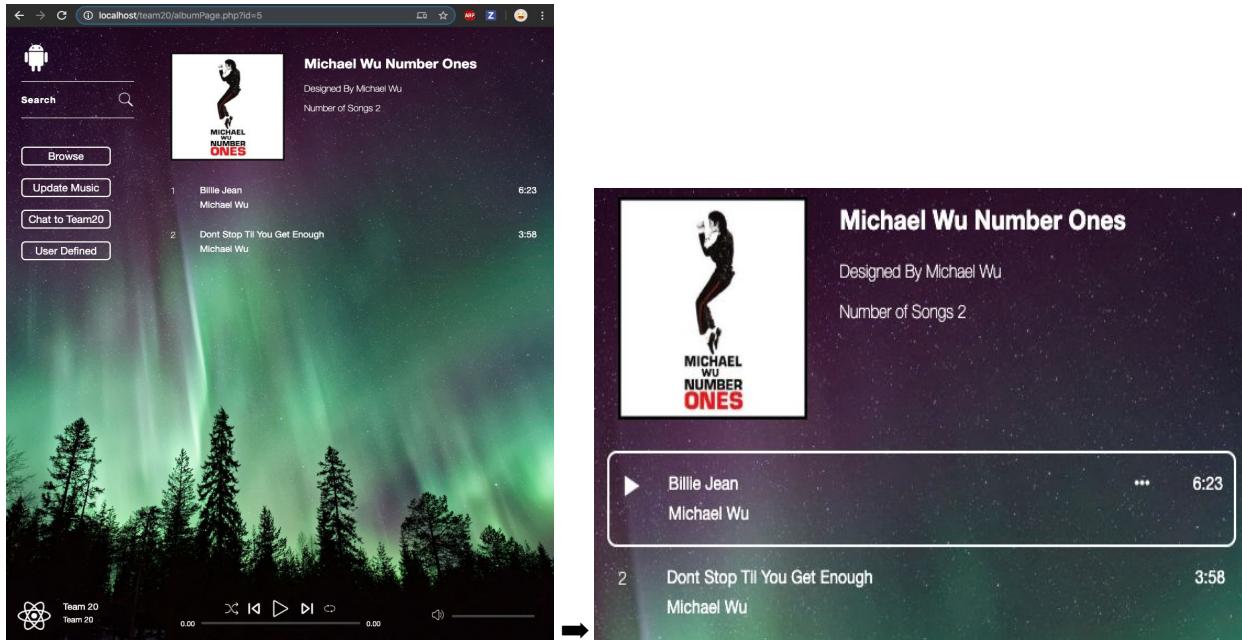


Once the user is signed in the same background picture is used, but we have different overlays on top of it depending on which page the user is on. In this example, the home page has 3 overlays: 2 types of navigation bars that will follow the user throughout the site (1 & 3) and the grid of albums (2).



When the user selects an album such as Michael Wu Number Ones, they will be redirected to that Albums page and will find a list of playable songs. As the user hovers over each song, a white rounded box will appear to let

the user know which song they are about to play. It is a very useful function for user interface/experience.



Security & Access Control

To maintain the security and access control of our website, our users table store the password after encryption via salt, hash and Secure Hash Algorithm (SHA-1). The reason for the repeated hashed password is only due to our test account having the same password for ease of testing. The benefits of storing the encrypted password is that no insider or admin may maliciously have access to customer account information.

Database changed

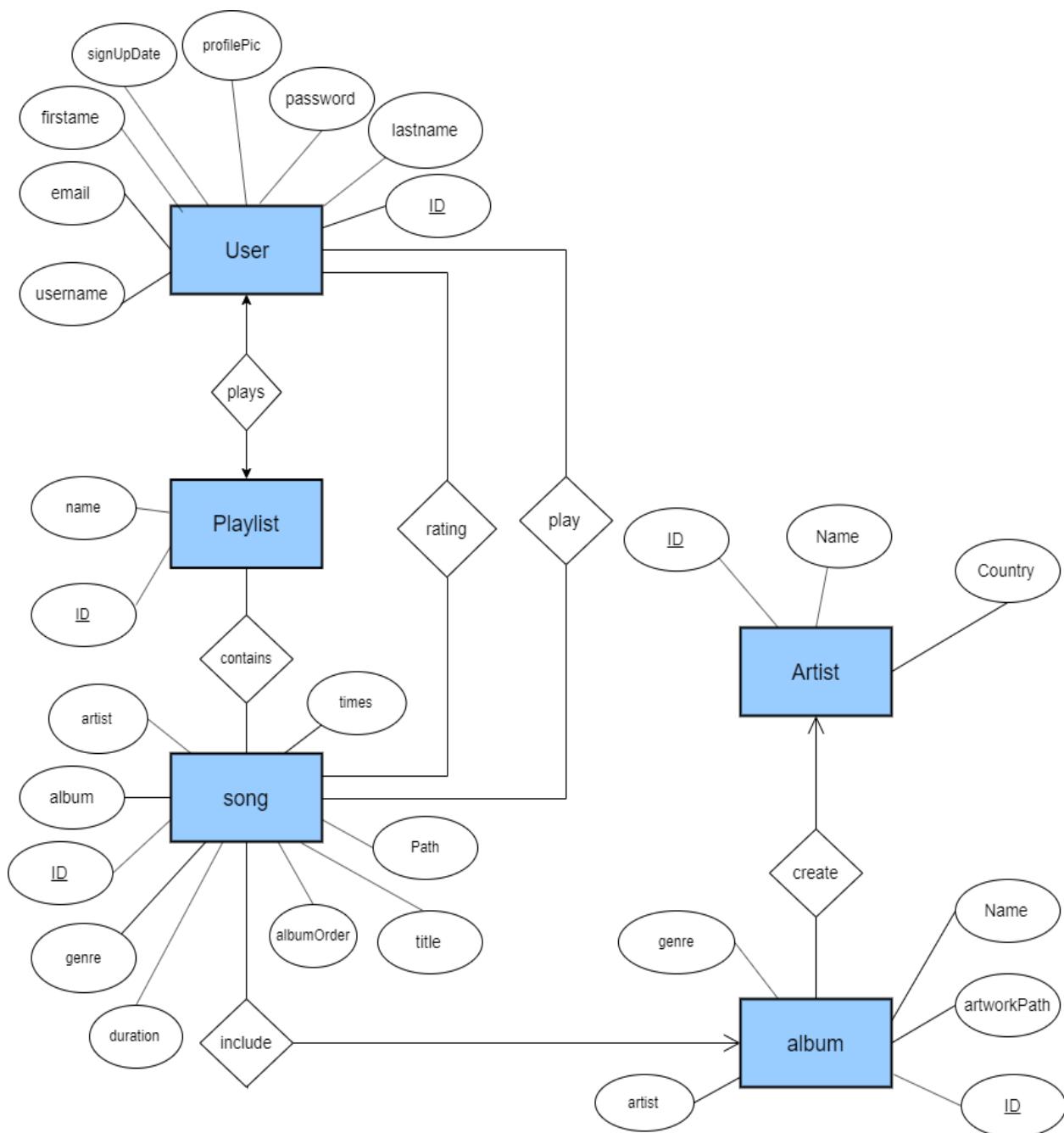
MariaDB [team20]> select * from users;

| id | username | firstName | lastName | email | password | signUpDate |
|----|--------------|-----------|----------|-------------------|--|---------------------|
| 1 | selina | Selina | Ding | Selina@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 2 | tim | Tim | Cook | Tim@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 3 | mike | Mike | Wu | Mike@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 5 | jack | Jack | Jack | Jack@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 6 | rose | Rose | Jack | Rose@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 7 | tom | Tom | Zheng | Tom@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 9 | newoton | New | Ton | Newton@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 10 | clinton | Clinton | Clinton | Clinton@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 11 | tesla | Tesla | Abc | Tesla@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 12 | mickey | Mickey | Mouse | Mickey@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 13 | lili | Lily | Absfs | Lili@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 14 | lucy | Lucy | Dsfadsaf | Lucy@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 15 | lilei | Li | Lei | Lilei@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 16 | apple | Ios | Tim | Apple@gmail.com | 8cb2237d0679ca88db6464eac60da96345513964 | 2019-11-16 00:00:00 |
| 17 | jobfair | job | fair | job@gmail.com | 5f4dcc3b5aa765d61d8327deb882cf99 | 2017-06-28 00:00:00 |
| 18 | mackookPro15 | machook | nro15 | mac@omail.com | 5f4dcc3b5aa765d61d83277uen882cf99 | 2019-11-28 02:47:27 |

We also have user validation for unique account names so if a new person tries to recreate a previous account name, there will be an error message displayed to the user to try a different account name. All these measures have been implemented in hopes that no one can simply access the site without proper registration. Implementation details down to the code can be found under the Implementation section of this report.

Project Design

ERD diagram Model for the application:



Entities:

1. User: holders that listen to music (everyone can have an account).
2. Song: Media that users can listen to.
3. Artist: Content creators for our music platform.
4. Album: A collection of songs from artists.
5. Playlist: User selected collection of music.

Attributes:

1. User: Username, password, ID, Lastname, firstname, email, profilePic, signUpDate.
2. Song: ID, Title and Duration, artist, album, genre, plays, path, albumOrder.
3. Artist: Name, Country, and ID.
4. Album: ID, Name and artist, artworkpath.
5. Playlist: ID, and Name.

Relationships:

1. Plays: The user with a user account can play music in the playlist
2. Plays(between user and song): The users can listen to many songs
3. Rating(between user and song): the users can leave the rating for many songs
4. Contains: The playlist will contain a number of songs, which can be none
5. Include: Song is included in a specific album
6. Create: An artist will create an album

Dependency:

1. A user with an account can play one playlist at a time
2. One Playlist can contain many songs and a song can be included in many playlists
3. A song can only be in one album but an album can have many songs
4. An album can only be created by one artist but an artist can create many albums

Perform the normalization process: They are in BCNF It doesn't have to perform normalization process

Create and show at least 10 tables according to schemes and model:

| | <u>id</u> | <u>name</u> |
|---|-----------|----------------------------|
| ▶ | 1 | J. Cole |
| | 2 | Kendrick Lamar |
| | 3 | Roddy Ricch |
| | 4 | Ezale |
| | 5 | Michael Wu |
| | 6 | Bruce Springsteen |
| | 7 | Various Chinese Artists |
| | 8 | Usher |
| | 9 | Don Sonrabiab |
| | 10 | G-Worthy |
| | 11 | Modern Talking |
| | 12 | Loso |
| | 13 | Luther Vandross |
| | 14 | Midland |
| | 15 | Taeyang |
| | 16 | Various Vietnamese Artists |
| | 17 | The XX |
| | 18 | Yoon Jong Shin |

Figure 1: Table artists

Figure 1: artists(id , name)

| | <u>id</u> | <u>title</u> | <u>artist</u> | <u>album</u> | <u>genre</u> | <u>duration</u> | <u>path</u> | <u>albumOrder</u> | <u>plays</u> |
|---|-----------|------------------------------|---------------|--------------|--------------|-----------------|--|-------------------|--------------|
| ▶ | 1 | 2Face | 1 | 1 | 1 | 4:46 | resource/music/Friday Night Lights/2Face.mp3 | 1 | 0 |
| | 2 | Back To The Topic Freestyle | 1 | 1 | 1 | 3:00 | resource/music/Friday Night Lights/Back To The ... | 2 | 0 |
| | 3 | Before Im Gone | 1 | 1 | 1 | 4:24 | resource/music/Friday Night Lights/Before Im G... | 3 | 0 |
| | 4 | Best Friend | 1 | 1 | 1 | 3:25 | resource/music/Friday Night Lights/Best Friend.... | 4 | 0 |
| | 5 | Alien Girl | 2 | 2 | 1 | 4:00 | resource/music/Overly Dedicated/Alien Girl.mp3 | 1 | 0 |
| | 6 | Growing Apart | 2 | 2 | 1 | 4:23 | resource/music/Overly Dedicated/Growing Apar... | 2 | 0 |
| | 7 | Ignorance Is Bliss | 2 | 2 | 1 | 3:31 | resource/music/Overly Dedicated/Ignorance Is ... | 3 | 0 |
| | 8 | PP 15 | 2 | 2 | 1 | 6:02 | resource/music/Overly Dedicated/PP 15.mp3 | 4 | 0 |
| | 9 | The Heart Pt 2 | 2 | 2 | 1 | 4:54 | resource/music/Overly Dedicated/The Heart Pt ... | 5 | 0 |
| | 10 | Chase Tha Bag | 3 | 3 | 1 | 3:33 | resource/music/Feed The Streets/Chase Tha Ba... | 1 | 0 |
| | 11 | Fucc It Up | 3 | 3 | 1 | 3:23 | resource/music/Feed The Streets/Fucc It Up.mp3 | 2 | 0 |
| | 12 | Plottin | 3 | 3 | 1 | 2:53 | resource/music/Feed The Streets/Plottin.mp3 | 3 | 0 |
| | 13 | Position | 3 | 3 | 1 | 3:19 | resource/music/Feed The Streets/Position.mp3 | 4 | 0 |
| | 14 | Uber | 3 | 3 | 1 | 2:51 | resource/music/Feed The Streets/Uber.mp3 | 5 | 0 |
| | 15 | 5 Minutes of Funktown | 4 | 4 | 1 | 5:01 | resource/music/Drug Funnies/5 Minutes of Funkt... | 1 | 0 |
| | 16 | Come And Talk To Me | 4 | 4 | 1 | 3:16 | resource/music/Drug Funnies/Come And Talk To ... | 2 | 0 |
| | 17 | Foreal Foreal | 4 | 4 | 1 | 3:20 | resource/music/Drug Funnies/Foreal Foreal.mp3 | 3 | 0 |
| | 18 | Billie Jean | 5 | 5 | 2 | 6:23 | resource/music/Number Ones/Billie Jean.mp3 | 1 | 0 |
| | 19 | Dont Stop Til You Get Enough | 5 | 5 | 2 | 3:58 | resource/music/Number Ones/Dont Stop Til You ... | 2 | 0 |
| | 20 | My Love | 7 | 7 | 7 | 4:34 | resource/music/Chinese Melodies/My Love.mp3 | 1 | 0 |

A

| | <u>id</u> | <u>title</u> | <u>artist</u> | <u>album</u> | <u>genre</u> | <u>duration</u> | <u>path</u> | <u>albumOrder</u> | <u>plays</u> |
|---|-----------|-----------------------------|---------------|--------------|--------------|-----------------|---|-------------------|--------------|
| | 21 | Theres Only You In My Heart | 7 | 7 | 7 | 5:00 | resource/music/Chinese Melodies/Theres Only Y... | 2 | 0 |
| | 22 | Confessions Pt 2 | 8 | 8 | 15 | 4:34 | resource/music/Confessions/Confessions pt2.mp3 | 1 | 0 |
| | 23 | Let It Burn | 8 | 8 | 15 | 5:00 | resource/music/Confessions/Let It Burn.mp3 | 2 | 0 |
| | 24 | Savan Pid | 9 | 9 | 12 | 3:42 | resource/music/Don Sonrabiaab Classics/Savan P... | 1 | 0 |
| | 25 | Tang Ruk See Dum | 9 | 9 | 12 | 4:29 | resource/music/Don Sonrabiaab Classics/Tang Ru... | 2 | 0 |
| | 26 | Foolish | 10 | 10 | 14 | 3:17 | resource/music/G-Worthy/Foolish.mp3 | 1 | 0 |
| | 27 | Never Miss | 10 | 10 | 14 | 3:44 | resource/music/G-Worthy/Never Miss.mp3 | 2 | 0 |
| | 28 | Cheri Cheri Lady | 11 | 11 | 5 | 3:17 | resource/music/Lets Talk About Love/Cheri Che... | 1 | 0 |
| | 29 | Youre My Heart Youre My ... | 11 | 11 | 5 | 3:15 | resource/music/Lets Talk About Love/Youre My ... | 2 | 0 |
| | 30 | Jai Sung Mai | 12 | 12 | 13 | 3:30 | resource/music/Loso Greatest Hits/Jai Sung Mai... | 1 | 0 |
| | 31 | Som Sarn | 12 | 12 | 13 | 5:14 | resource/music/Loso Greatest Hits/Som Sarn.mp3 | 2 | 0 |
| | 32 | Dont You Know That | 13 | 13 | 10 | 4:05 | resource/music/Never Too Much/Dont You Kno... | 1 | 0 |
| | 33 | Never Too Much | 13 | 13 | 10 | 4:41 | resource/music/Never Too Much/Never Too Muc... | 2 | 0 |
| | 34 | Burn Out | 14 | 14 | 3 | 3:07 | resource/music/On The Rocks/Burn Out.mp3 | 1 | 0 |
| | 35 | Drinkin Problem | 14 | 14 | 3 | 3:48 | resource/music/On The Rocks/Drinkin Problem.... | 2 | 0 |
| | 36 | I Need A Girl | 15 | 15 | 8 | 3:37 | resource/music/Solar/I Need A Girl.mp3 | 1 | 0 |
| | 37 | Wedding Dress | 15 | 15 | 8 | 4:00 | resource/music/Solar/Wedding Dress.mp3 | 2 | 0 |
| | 38 | Sad Eyes | 6 | 6 | 9 | 3:51 | resource/music/The River/Sad Eyes.mp3 | 1 | 0 |
| | 39 | Take Em As They Come | 6 | 6 | 9 | 4:36 | resource/music/The River/Take Em As They Co... | 2 | 0 |
| * | 40 | That Tinh | 16 | 16 | 11 | 6:34 | resource/music/Vietnamese Karaoke Classics/Th... | 1 | 0 |

Activate Window

| | <u>id</u> | <u>title</u> | <u>artist</u> | <u>album</u> | <u>genre</u> | <u>duration</u> | <u>path</u> | <u>albumOrder</u> | <u>plays</u> |
|---|-----------|----------------------|---------------|--------------|--------------|-----------------|--|-------------------|--------------|
| | 41 | Vang Trang Khoc | 16 | 16 | 11 | 5:10 | resource/music/Vietnamese Karaoke Classics/Va... | 2 | 0 |
| | 42 | Intro | 17 | 17 | 14 | 2:08 | resource/music/XX/Intro.mp3 | 1 | 0 |
| | 43 | Islands | 17 | 17 | 14 | 2:42 | resource/music/XX/Islands.mp3 | 2 | 0 |
| | 44 | From January To June | 18 | 18 | 16 | 4:15 | resource/music/Yoon Jong Shin Ballad/From Jan... | 1 | 0 |
| | 45 | I Like | 18 | 18 | 16 | 5:28 | resource/music/Yoon Jong Shin Ballad/I Like.mp3 | 2 | 0 |
| * | HULL | HULL | HULL | HULL | HULL | HULL | HULL | HULL | HULL |

Activ

Figure 2: table songs with 45 songs

Figure 2: songs(id, title, artist, album, genre, duration, path, albumOrder, plays)

| | id | title | artist | genre | artworkPath |
|---|----|----------------------------|--------|-------|--|
| ▶ | 1 | Friday Night Lights | 1 | 1 | resource/images/artwork/FridayNightLightsFront.jpg |
| | 2 | Overly Dedicated | 2 | 1 | resource/images/artwork/OverlyDedicatedFront.jpg |
| | 3 | Feed Tha Streets | 3 | 1 | resource/images/artwork/FeedThaStreetsFront.jpg |
| | 4 | Drug Funnies | 4 | 1 | resource/images/artwork/DrugFunniesFront.jpg |
| | 5 | Michael Wu Number Ones | 5 | 2 | resource/images/artwork/MichaelWuNumberOnesFront.jpg |
| | 6 | The River | 6 | 9 | resource/images/artwork/The River.jpeg |
| | 7 | Chinese Melodies | 7 | 7 | resource/images/artwork/Chinese Melodies.jpeg |
| | 8 | Confessions | 8 | 15 | resource/images/artwork/Confessions.jpeg |
| | 9 | Don Sonrabiab Classics | 9 | 12 | resource/images/artwork/Don Sonrabiab Classics.jpeg |
| | 10 | G-Worthy | 10 | 14 | resource/images/artwork/G-Worthy.jpeg |
| | 11 | Lets Talk About Love | 11 | 5 | resource/images/artwork/Lets Talk About Love.jpeg |
| | 12 | Loso Greatest Hits | 12 | 13 | resource/images/artwork/Loso Greatest Hits.jpeg |
| | 13 | Never Too Much | 13 | 10 | resource/images/artwork/Never Too Much.jpeg |
| | 14 | On The Rocks | 14 | 3 | resource/images/artwork/On The Rocks.jpeg |
| | 15 | Solar | 15 | 8 | resource/images/artwork/Solar.jpeg |
| | 16 | Vietnamese Karaoke Clas... | 16 | 11 | resource/images/artwork/Vietnamese Karaoke Classics.jpeg |
| | 17 | XX | 17 | 4 | resource/images/artwork/XX.jpeg |
| | 18 | Yoon Jong Shin Ballad | 18 | 6 | resource/images/artwork/Yoon Jong Shin Ballad.jpeg |

Figure 3: table albums with 18 albums

Figure 3: albums(id, title, artist, genre, artworkPath)

| | id | name |
|---|-----|-----------------------------------|
| ▶ | 1 | Ends of the year |
| | 2 | animes series |
| | 3 | Hip-hop |
| | 4 | Pop music |
| | 5 | gym songs |
| | 6 | anniversary |
| | 7 | Love lists of mine |
| | 8 | my friend bad list |
| | 9 | fairy tale |
| | 10 | love songs |
| | 11 | happy songs |
| | 12 | workout |
| | 13 | music of the morning |
| | 14 | Night is not bad |
| | 15 | new branch will have to listen to |
| * | NUL | NUL |

Figure 4: table of the playlist

Figure 4: playlist(id, name)

| | artistID | albumID |
|---|----------|---------|
| ► | 15 | 2 |
| | 2 | 9 |
| | 4 | 7 |
| | 2 | 8 |
| | 4 | 3 |
| | 1 | 1 |
| | 1 | 4 |
| | 6 | 5 |
| | 8 | 10 |
| | 9 | 6 |
| | 10 | 11 |
| | 11 | 12 |
| | 13 | 14 |
| | 14 | 13 |
| | 10 | 15 |

Figure 6: table artist_create_alb

Figure 6: artist_create_alb (artistID, albumID)

| | SongID | PlaylistID |
|---|--------|------------|
| | 1 | 1 |
| | 2 | 2 |
| | 3 | 3 |
| | 4 | 6 |
| | 5 | 5 |
| ► | 6 | 9 |
| | 7 | 4 |
| | 8 | 7 |
| | 9 | 8 |
| | 10 | 11 |
| | 11 | 10 |
| | 12 | 15 |
| | 13 | 13 |
| | 14 | 12 |
| | 15 | 14 |
| * | HULL | HULL |

Figure 7: table contains

Figure 7: contains(SongID, albumID)

| | <u>id</u> | name |
|--|-----------|------------------|
| | 11 | Rock |
| | 12 | Pop |
| | 13 | canon |
| | 14 | Bach |
| | 15 | piano |
| | 1 | Rap |
| | 2 | Pop |
| | 3 | Country |
| | 4 | Electropop |
| | 5 | Dance-pop |
| | 6 | Korean Ballad |
| | 7 | Chinese Melodies |
| | 8 | KPop |
| | 9 | Rock |
| | 10 | Soul |
| | 11 | Vietnamese |
| | 12 | Thai |
| | 13 | Thai Rock |
| | 14 | Gangsta Rap |
| | 15 | R&B |

Figure 8: table genre

Figure 8: genre(id, name)

| | albumID | songID |
|---|---------|--------|
| ▶ | 1 | 1 |
| | 1 | 2 |
| | 1 | 3 |
| | 1 | 4 |
| | 2 | 5 |
| | 2 | 6 |
| | 2 | 7 |
| | 2 | 8 |
| | 2 | 9 |
| | 3 | 10 |
| | 3 | 11 |
| | 3 | 12 |
| | 3 | 13 |
| | 3 | 14 |
| | 4 | 15 |
| | 4 | 16 |
| | 4 | 17 |
| | 6 | 38 |
| | .. | .. |
| | albumID | songID |
| | 6 | 39 |
| | 5 | 18 |
| | 5 | 19 |
| | 7 | 20 |
| | 7 | 21 |
| | 8 | 22 |
| | 8 | 23 |
| | 9 | 24 |
| | 9 | 25 |
| | 10 | 26 |
| | 10 | 27 |
| | 11 | 28 |
| | 11 | 29 |
| | 12 | 30 |
| | 12 | 31 |
| | 13 | 32 |
| | 13 | 33 |
| | 14 | 34 |
| | .. | .. |
| | 14 | 35 |
| | 15 | 36 |
| | 15 | 37 |
| | 16 | 40 |
| | 16 | 41 |
| | 17 | 42 |
| | 17 | 43 |
| | 18 | 44 |
| | 18 | 45 |

Figure 9: table include

Figure 9: include(albumID, songID)

| | userID | songID |
|---|--------|--------|
| ▶ | 3 | 10 |
| | 1 | 2 |
| | 1 | 8 |
| | 1 | 3 |
| | 2 | 4 |
| | 2 | 6 |
| | 2 | 9 |
| | 8 | 11 |
| | 5 | 13 |
| | 6 | 15 |
| | 2 | 16 |
| | 3 | 18 |
| | 4 | 14 |
| | 5 | 15 |
| | 6 | 17 |

Figure 10: table play

Figure 10: play(userID,songID)

| | UserID | PlaylistID |
|---|--------|------------|
| ▶ | 1 | 3 |
| | 2 | 2 |
| | 3 | 1 |
| | 4 | 6 |
| | 5 | 7 |
| | 6 | 9 |
| | 7 | 4 |
| | 8 | 5 |
| | 9 | 8 |
| | 10 | 11 |
| | 11 | 10 |
| | 12 | 15 |
| | 13 | 13 |
| | 14 | 12 |
| | 15 | 14 |
| * | NULL | NULL |

Figure 11: table plays

Figure 11: plays(userID, songID)

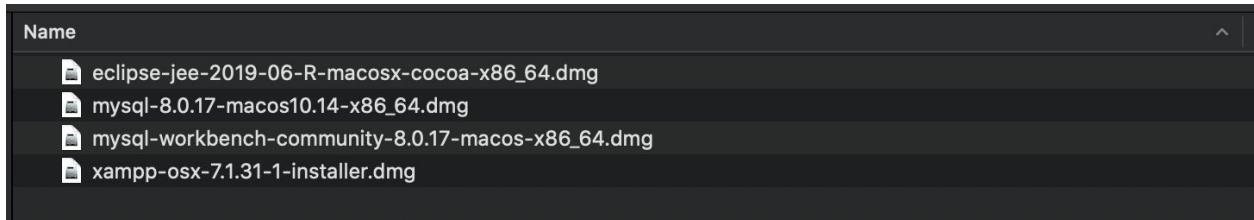
```
[MariaDB [(none)]]> use team20;
Database changed
[MariaDB [team20]]> DESC users;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) | NO | PRI | NULL | auto_increment |
| username | varchar(25) | NO | | NULL | |
| firstName | varchar(50) | NO | | NULL | |
| lastName | varchar(50) | NO | | NULL | |
| email | varchar(200) | NO | | NULL | |
| password | varchar(40) | NO | | NULL | |
| signUpDate | datetime | NO | | NULL | |
| profilePic | varchar(500) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.070 sec)

[MariaDB [team20]]> select * from users;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | username | firstName | lastName | email | password | signUpDate | profilePic |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | selina | Selina | Ding | Selina@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 2 | tim | Tim | Cook | Tim@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 3 | mike | Mike | Wu | Mike@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 5 | jack | Jack | Jack | Jack@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 6 | rose | Rose | Jack | Rose@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 7 | tom | Tom | Zheng | Tom@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 9 | newoton | New | Ton | Newton@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 10 | clinton | Clinton | Clinton | Clinton@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 11 | tesla | Tesla | Abc | Tesla@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 12 | mickey | Mickey | Mouse | Mickey@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 13 | lili | Lily | Absfs | Lili@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 14 | lucy | Lucy | Dsfadsaf | Lucy@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 15 | lilei | Li | Lei | Lilei@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 16 | apple | Ios | Tim | Apple@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png |
| 17 | jobfair | job | fair | job@gmail.com | 5f4dcc3b5a755d1d3327de882cf99 | 2017-06-28 08:00:00 | resource/images/profile-pics/head.png |
| 18 | mackookPro15 | macbook | pro15 | mac@gmail.com | 5f4dcc3b5a755d1d3327de882cf99 | 2019-11-28 02:47:27 | resource/images/profile/head.png |
| 21 | M.Craft | M. | Craft | m@gmail.com | 5f4dcc3b5a755d1d3327de882cf99 | 2019-11-28 03:07:18 | resource/images/profile/head.png |
| 22 | lambdaExpression | lambda | Expression | lambda@gmail.com | 5f4dcc3b5a755d1d3327de882cf99 | 2019-11-16 08:00:00 | resource/images/profile/head.png |
| 34 | cs157aFinalProject | Final | Project | Cs157afinal@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-12-04 08:00:00 | resource/images/profile/head.png |
| 33 | lambdaSortByEmail | sort | byEmail | by@gmail.com | 5f4dcc3b5a755d1d3327de882cf99 | 2019-11-30 23:03:02 | resource/images/profile/head.png |
+-----+-----+-----+-----+-----+-----+-----+-----+
22 rows in set (0.011 sec)
```

Figure 12: table users

Figure 12: users(id,username,firstname,lastname,email,password,signUpDate,profilePic)

Installation



1. We should install XAMPP for mac or windows by this link:

<https://www.apachefriends.org/download.html> and then choose the one fit with your machine

| Version | Checksum | Size |
|---------------------|---------------------------|--------------------------|
| 7.1.33 / PHP 7.1.33 | What's Included? md5 sha1 | Download (64 bit) 141 Mb |
| 7.2.25 / PHP 7.2.25 | What's Included? md5 sha1 | Download (64 bit) 145 Mb |
| 7.3.12 / PHP 7.3.12 | What's Included? md5 sha1 | Download (64 bit) 146 Mb |

2. After installed, you can see folder XAMPP

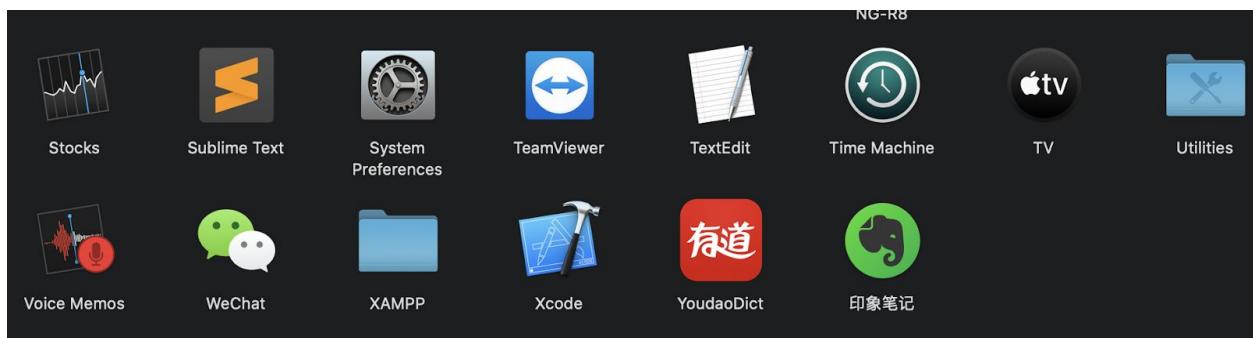


Image 2.1: XAMPP folder is installed

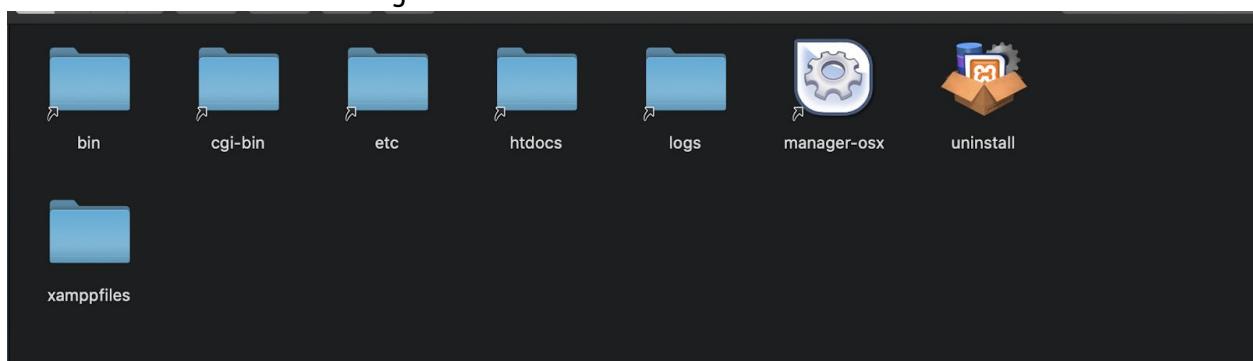


Image 2.2: Here is what is inside XAMPP looks like

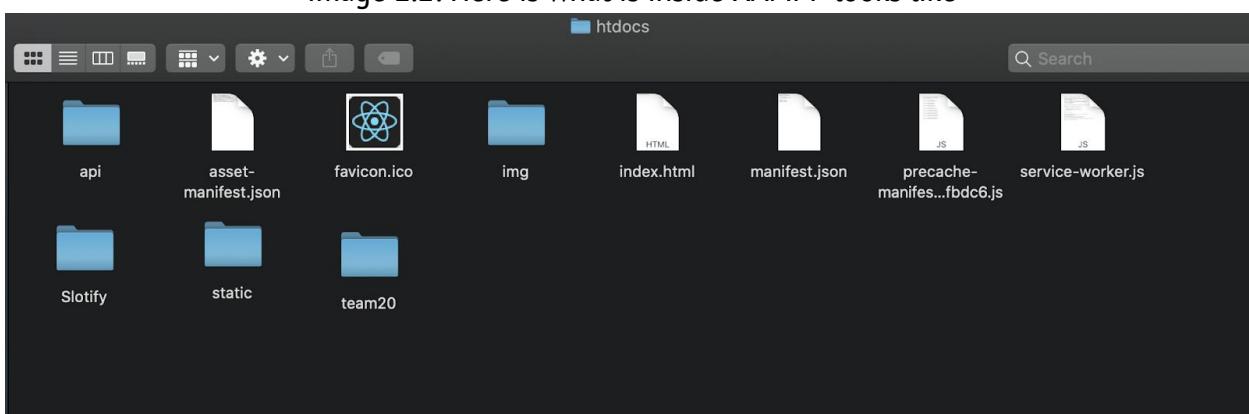


Image 2.3: Inside htdocs file

3. Cloning code from GitHub: click on the green button Clone and Downloaded on the top right corner of the list of folders to get HTTPS of the codes folder and copy that link

corner of the list of folders to get HTTPS of the codes folder and copy that link

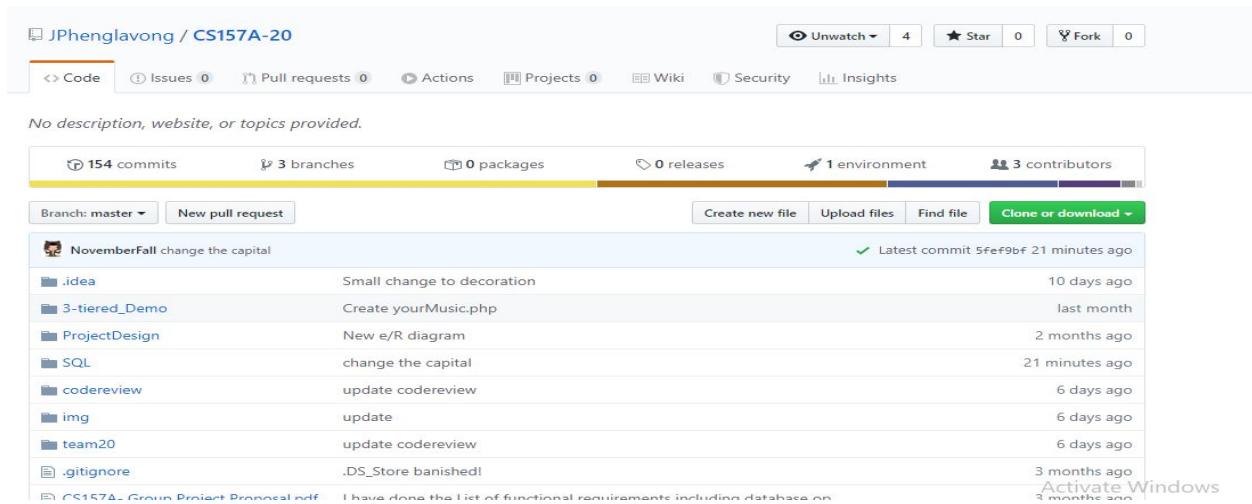


Image 3.1

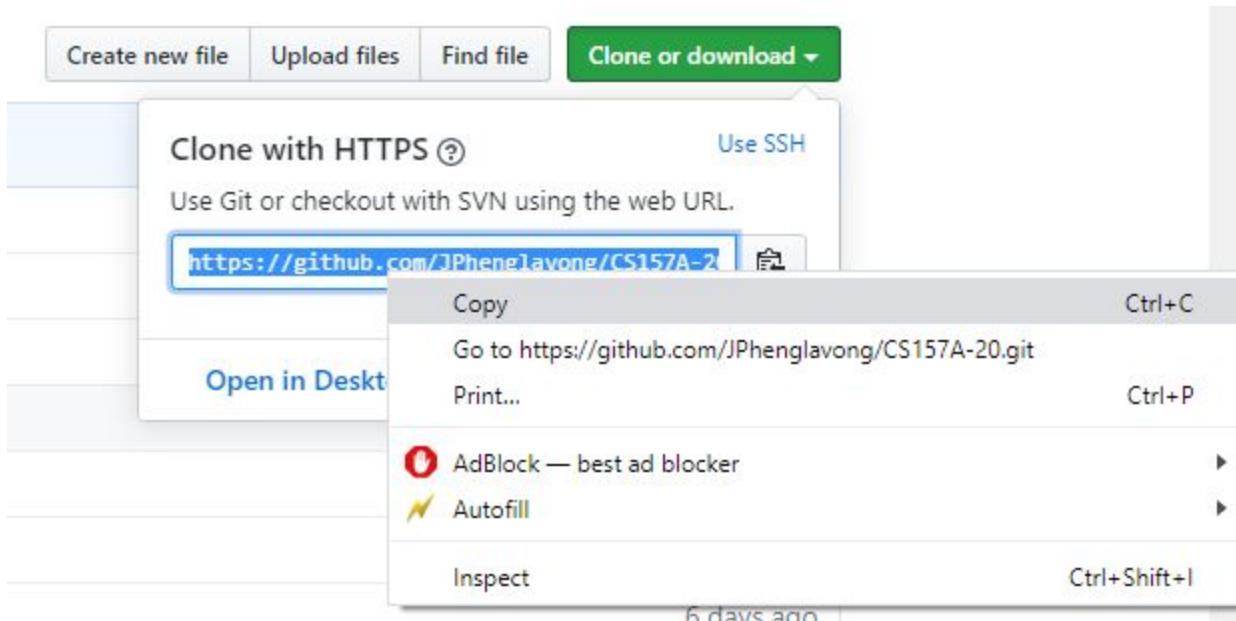


Image 3.2

4. At this point, you can clone a version of our code from our GitHub by using the command: git clone <The link you copy from the image 3.2>
5. Open our folder after cloning and copy folder team20 into htdocs folder like in image 2.3
6. Click XAMPP folder => htdocs => team20
7. Now, as you see team20 is our PHP+MySQL+JavaScript project

8. Use VS CODE to open our project. First, follow this link to download a version of Visual Studio

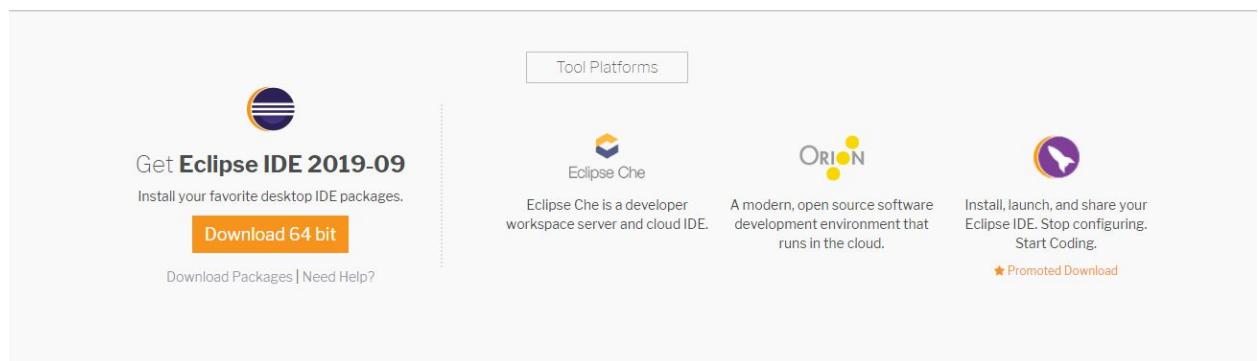
Code for mac/ Window/ or Linux: <https://code.visualstudio.com/docs/setup/setup-overview>

The screenshot shows the Visual Studio Code documentation website. The top navigation bar includes links for Visual Studio Code, Docs, Updates, Blog, API, Extensions, and FAQ. A search bar and a 'Download' button are also present. A banner at the top states 'Version 1.40 is now available! Read about the new features and fixes from October.' On the left, a sidebar menu lists various setup guides: Overview, SETUP (with Overview selected), GET STARTED, USER GUIDE, LANGUAGES, NODEJS / JAVASCRIPT, TYPESCRIPT, PYTHON, JAVA, C++, AZURE, and REMOTE. The main content area is titled 'Setting up Visual Studio Code'. It discusses getting up and running quickly, installing on Linux, macOS, and Windows, and provides cross-platform support. It also covers update cadence and system requirements. On the right, there's an 'Edit' button, an 'IN THIS ARTICLE' sidebar with links to Cross platform, Update cadence, Insiders nightly build, Additional components, Extensions, Next steps, and Common questions, and a social sharing sidebar with links for Tweet this link, Subscribe, Ask questions, Follow @code, Request features, Report issues, and Watch videos.

9. Install Eclipse (everyone should have installed this software), for the JDBC part. Here is the

link to the download page of Eclipse IDE 2019-19 version:

<https://www.eclipse.org/downloads/>



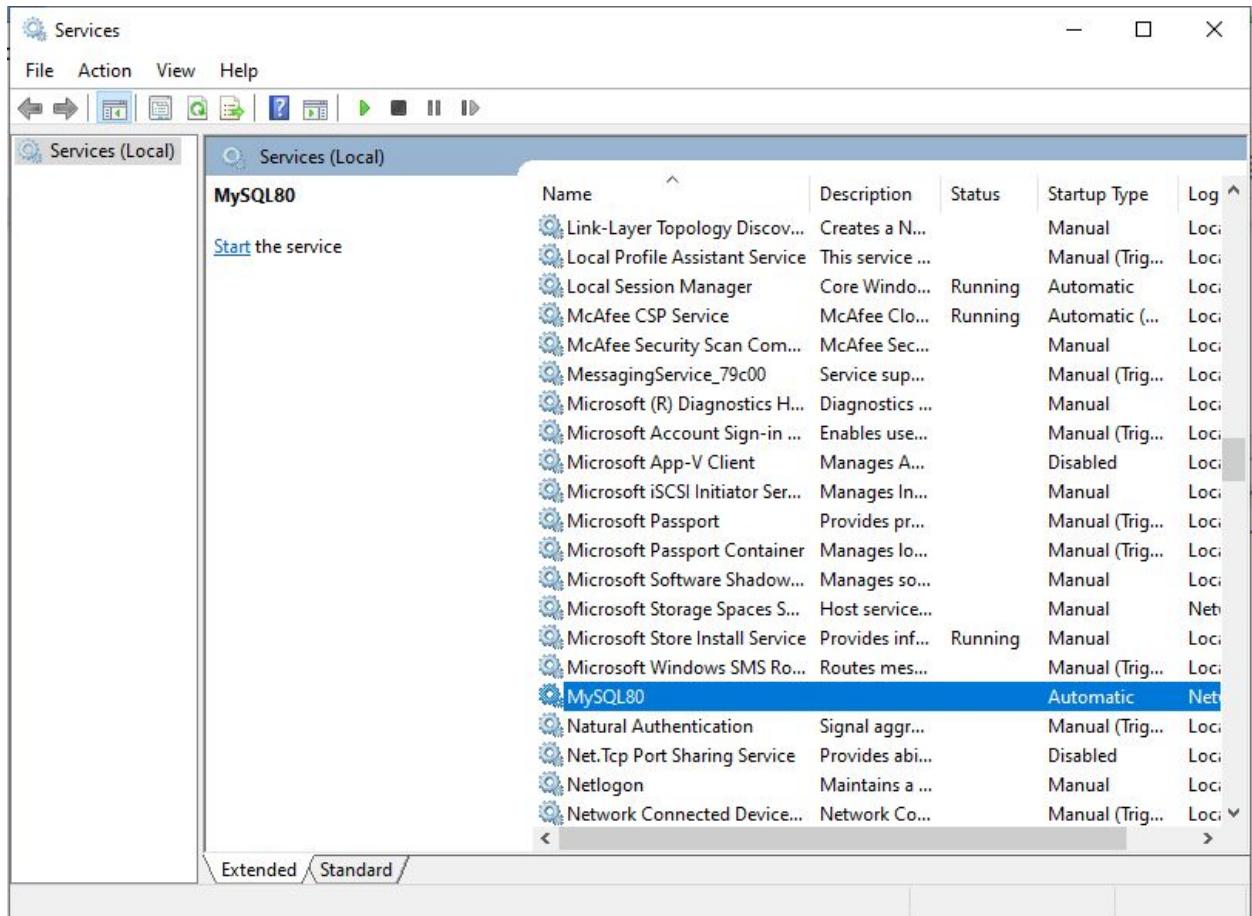
10. Since the XAMPP include Apache server, so we **don't need** to install Apache
11. You will need to install MySQL to get the database on track. Follow this link to download MySQL <https://www.mysql.com/downloads/>

A screenshot of the MySQL Downloads page. At the top, there's a navigation bar with links for 'MySQL.COM', 'DOWNLOADS' (which is highlighted in orange), 'DOCUMENTATION', and 'DEVELOPER ZONE'. The main content area has a blue header 'MySQL Enterprise Edition'. Below it, a sub-header states 'MySQL Enterprise Edition includes the most comprehensive set of advanced features and management tools for MySQL.' There are two columns of bullet points: one on the left and one on the right, both listing various MySQL features. At the bottom, there are three call-to-action buttons: 'Learn More', 'Customer Download', and 'Trial Download'. A small 'Activate' link is visible in the bottom right corner.

12. After download MySQL, open XAMPP to get the gate before connecting to MySQL



Click the Start button of Apache and MySQL, and if you get the problem that can't start MySQL to get Port, that's because MySQL80 is running inside the service. To solve the problem we search for “service” in cmd command, find MySQL80 which is running and stop it



Then you can continue getting the gate

13. Open folder SQL in our code folder you can find all the code to create table and data inserting that the program need

| khoaLib Update include.sql | | ✓ Latest commit 473c93e 44 minutes ago |
|--------------------------------------|--|--|
| 📁 .idea | Small change to decoration | 10 days ago |
| 📁 3-tiered_Demo | Create yourMusic.php | last month |
| 📁 ProjectDesign | New e/R diagram | 2 months ago |
| 📁 SQL | Update include.sql | 44 minutes ago |
| 📁 codereview | update codereview | 6 days ago |
| 📁 img | update | 6 days ago |
| 📁 team20 | John Changed some album pictures | 3 hours ago |
| 📄 .gitignore | .DS_Store banished! | 3 months ago |
| 📄 CS157A- Group Project Proposal.pdf | I have done the List of functional requirements including database op... | 3 months ago |
| 📄 CS157A_Team20_GroupProjectProp... | Since we change the topic.now we update the new Proposal | 3 months ago |
| 📄 Khoa_Functional requirement | Create Khoa_Functional requirement | 3 months ago |
| 📄 Project Reauirement Document_Kh... | Mv contribution is hiahliated bv areen | 3 months ago |

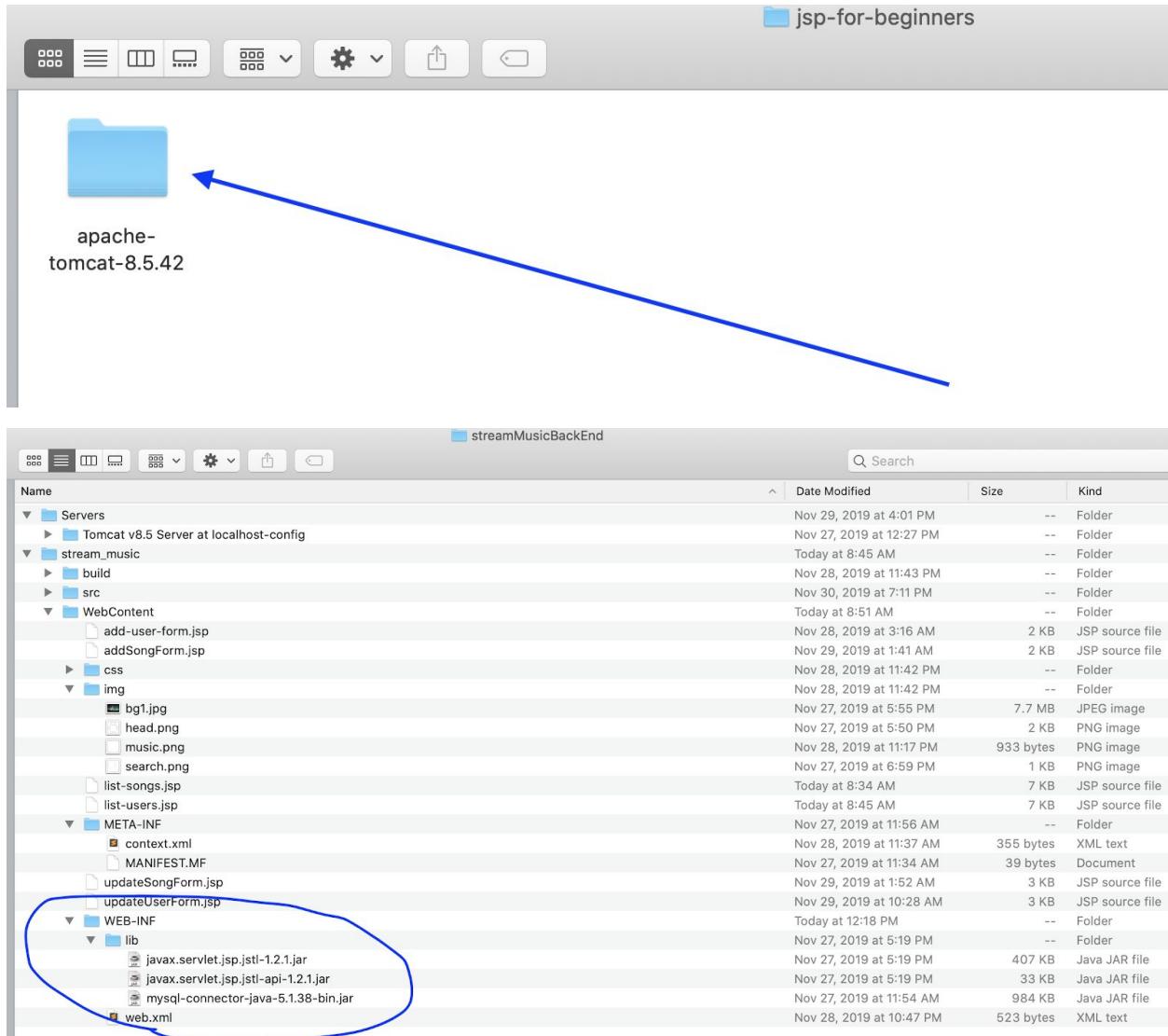
| khoaLib Update include.sql | | ✓ Latest commit 473c93e 44 minutes ago |
|----------------------------|--------------------------------------|--|
| .. | | |
| 📄 Playlist.sql | update playlist sql code | 6 days ago |
| 📄 albums.sql | Update albums.sql | 3 hours ago |
| 📄 artist.sql | Change some artists's names | 4 hours ago |
| 📄 artist_create_alb.sql | Update artist_create_alb.sql | 3 hours ago |
| 📄 contains.sql | adding sql codes | 6 days ago |
| 📄 genres.sql | Update genres.sql | 3 hours ago |
| 📄 include.sql | Update include.sql | 44 minutes ago |
| 📄 musics.sql | Upload some change to table contents | 13 days ago |
| 📄 play.sql | Update play.sql | 2 hours ago |
| 📄 plays.sql | adding sql codes | 6 days ago |
| 📄 rating.sql | Update rating.sql | 2 hours ago |
| 📄 songs.sql | Update songs.sql | 3 hours ago |
| 📄 user.sql | update sql | 10 days ago |

14. You can now run our code start with register.php

Installation for Project PartB: JDBC

Stream Music BackEnd

- First, we need to install eclipseEE
- Download apache-tomcat-8.5.42
- You can download these from google, it is easy to download, everything is free



- We also need to import these three libraries

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays a Java project named "gitJspMusic". It contains several packages and files, including "Servers", "Tomcat v8.5 Server at localhost-config", "JAX-WS Web Services", "Java Resources", "JavaScript Resources", "build", "WebContent" (containing "css", "img", "META-INF", and "WEB-INF" which includes "lib" with JAR files like "javax.servlet.jsp.jstl-1.2.1.jar", "javax.servlet.jsp.jstl-api-1.2.1.jar", and "mysql-connector-java-5.1.38-bin.jar"), and "WEB-INF" (containing "web.xml" and JSP files like "add-user-form.jsp", "addSongForm.jsp", "list-songs.jsp", "list-users.jsp", "updateSongForm.jsp", and "updateUserForm.jsp"). On the right, the code editor window displays the Java code for "UserDatabaseUtil.java". The code imports various JDBC classes and defines a class that interacts with a database to retrieve user data. Three blue arrows point from the text "This is the all files on eclipse project including libraries and tomcat" to the "Tomcat v8.5 Server at localhost-config", "lib", and "WEB-INF" sections in the Project Explorer.

```
1 package com.cs157a.web.jdbc;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.sql.Statement;
8 import java.util.ArrayList;
9 import java.util.Collections;
10
11 import javax.sql.DataSource;
12
13 public class UserDatabaseUtil {
14
15     private DataSource dataSource;
16
17     public UserDatabaseUtil(DataSource ds) {
18         dataSource = ds;
19     }
20
21     public ArrayList<user> getUsers() throws Exception{
22         ArrayList<user> users = new ArrayList<>();
23
24         Connection conn = null;
25         Statement stmt = null;
26         ResultSet rs = null;
27
28         try {
29             //get a connection
30             conn = dataSource.getConnection();
31             //create sql statement
32             String sql = "SELECT * FROM users";
33             stmt = conn.createStatement();
34             //execute query
35             rs = stmt.executeQuery(sql);
36
37             //process result set
38             while(rs.next()) {
```

- This is the all files on eclipse project including libraries and tomcat

Implementation

Input the address: <http://localhost/team20/register.php>

Event 1: Register, create a new account, and insert data to our DB

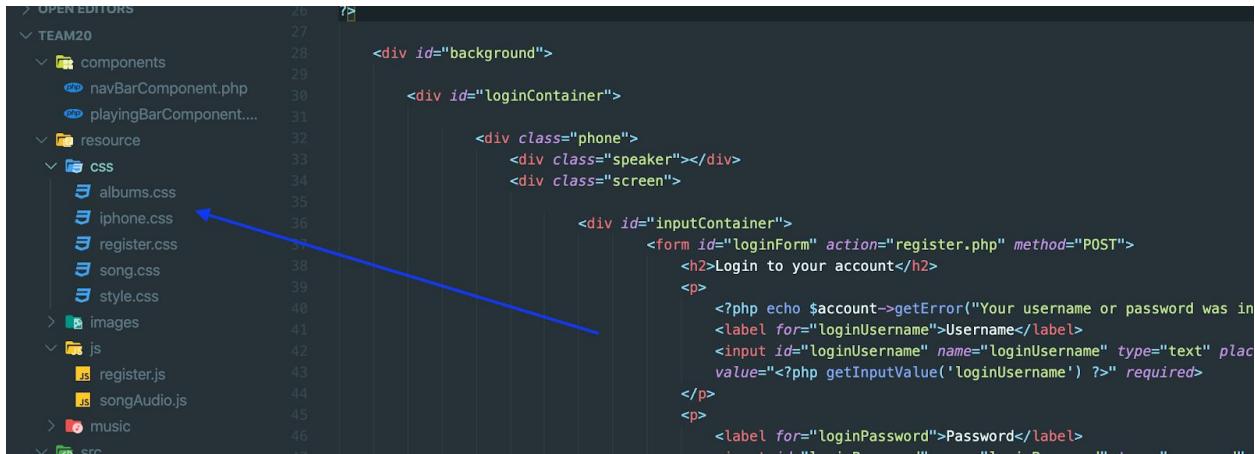
1. Create a register.php

```

<div id="background">
    <div id="loginContainer">
        <div class="phone">
            <div class="speaker"></div>
            <div class="screen"></div>
        </div>
        <div id="inputContainer">
            <form id="loginForm" action="register.php" method="POST">
                <h2>Login to your account</h2>
                <p>
                    <?php echo $account->getError("Your username or password was incorrect"); ?>
                    <label for="loginUsername">Username</label>
                    <input id="loginUsername" name="loginUsername" type="text" placeholder="username" value="<?php getInputValue('loginUsername') ?>" required>
                </p>
                <p>
                    <label for="loginPassword">Password</label>
                    <input id="loginPassword" name="loginPassword" type="password" placeholder="Your password" required>
                </p>
                <button type="submit" name="loginButton">LOG IN</button>
            </form>
            <div class="hasAccountText">
                <span id="hideLogin">If you don't have a account, please sign up</span>
            </div>
        </div>
    </div>
<form id="registerForm" action="register.php" method="POST">
    <h2>Create your free account</h2>
    <p>
        <?php echo $account->getError("Your username must be between 3 and 25 characters"); ?>
        <?php echo $account->getError("Your username already exists"); ?>
        <label for="username">Username</label>
        <input id="username" name="username" type="text" placeholder="username" value="<?php getInputValue('username') ?>" required>
    </p>
    <p>
        <?php echo $account->getError("Your first name must be between 2 and 25 characters"); ?>
        <label for="firstName">First name</label>
        <input id="firstName" name="firstName" type="text" placeholder="first name" value="<?php getInputValue('firstName') ?>" required>
    </p>

```

- We have background, loginContainer, phone, tag name
- Using CSS to create a new UI, create these .css files inside the CSS folder



The screenshot shows a code editor with a file tree on the left and a code editor window on the right.

File Tree:

- > OPEN EDITORS
- > TEAM20
 - components
 - navBarComponent.php
 - playingBarComponent...
- > resource
 - css
 - albums.css
 - iphone.css
 - register.css
 - song.css
 - style.css
- > images
- > js
 - register.js
 - songAudio.js
- > music

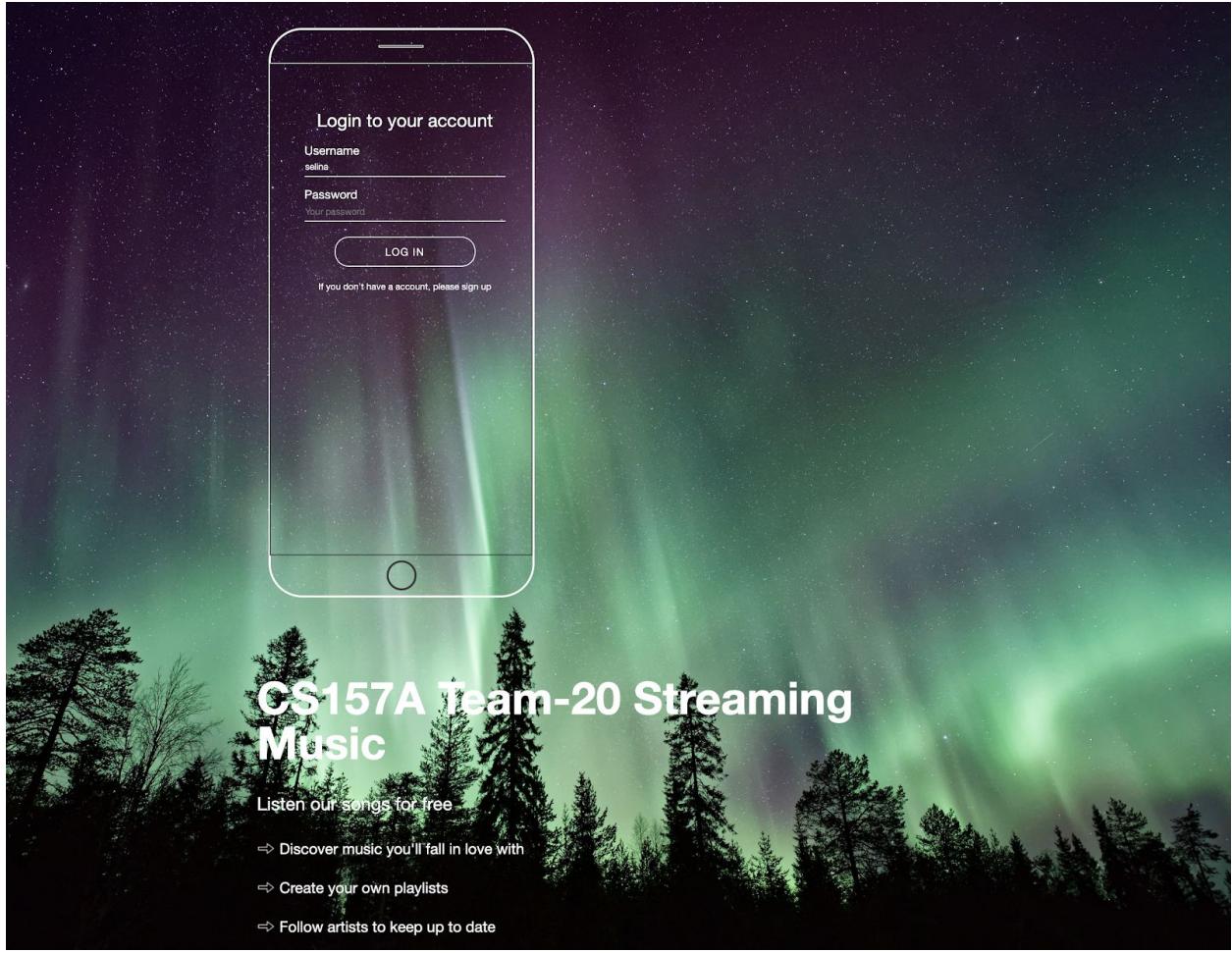
Code Editor (register.php):

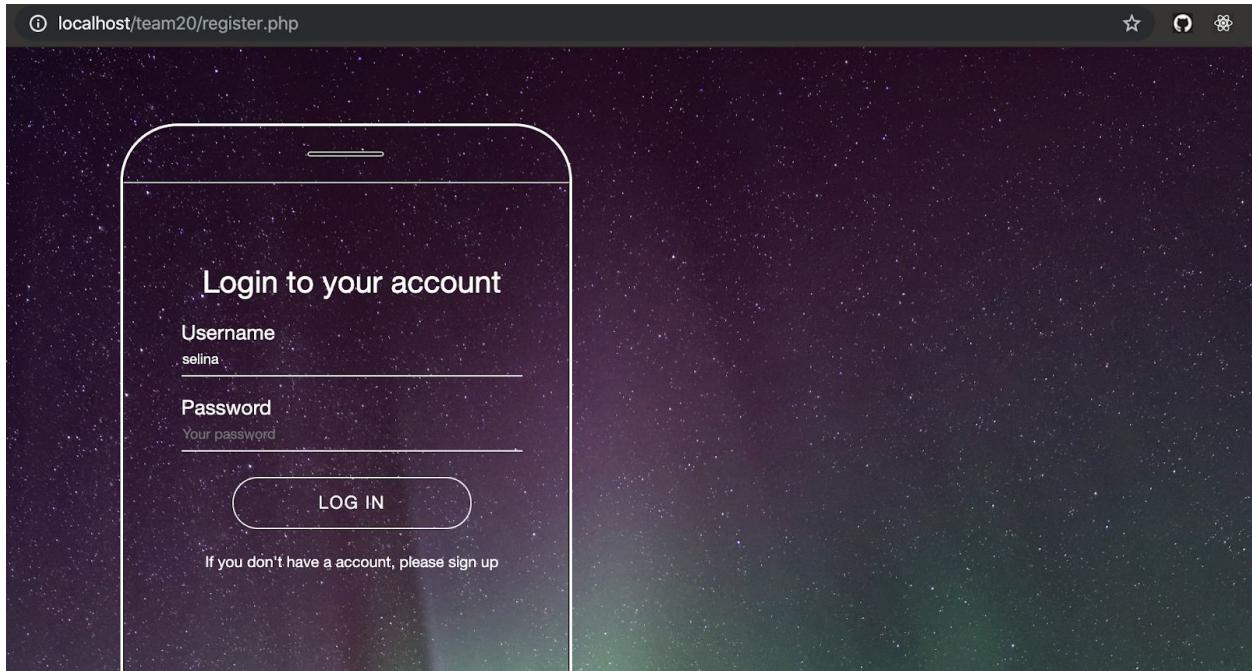
```

26 <div id="background">
27   <div id="loginContainer">
28     <div class="phone">
29       <div class="speaker"></div>
30       <div class="screen">
31
32         <div id="inputContainer">
33           <form id="loginForm" action="register.php" method="POST">
34             <h2>Login to your account</h2>
35             <p>
36               <?php echo $account->getError("Your username or password was in<br>correct") ?>
37               <label for="loginUsername">Username</label>
38               <input id="loginUsername" name="loginUsername" type="text" placeholder="Your username" value="<?php getInputValue('loginUsername') ?>" required>
39             </p>
40             <p>
41               <label for="loginPassword">Password</label>
42               <input id="loginPassword" name="loginPassword" type="password" placeholder="Your password" required>
43             </p>
44             <button type="submit" value="LOG IN">LOG IN</button>
45           </form>
46         </div>
47       </div>
48     </div>
49   </div>
50 </div>

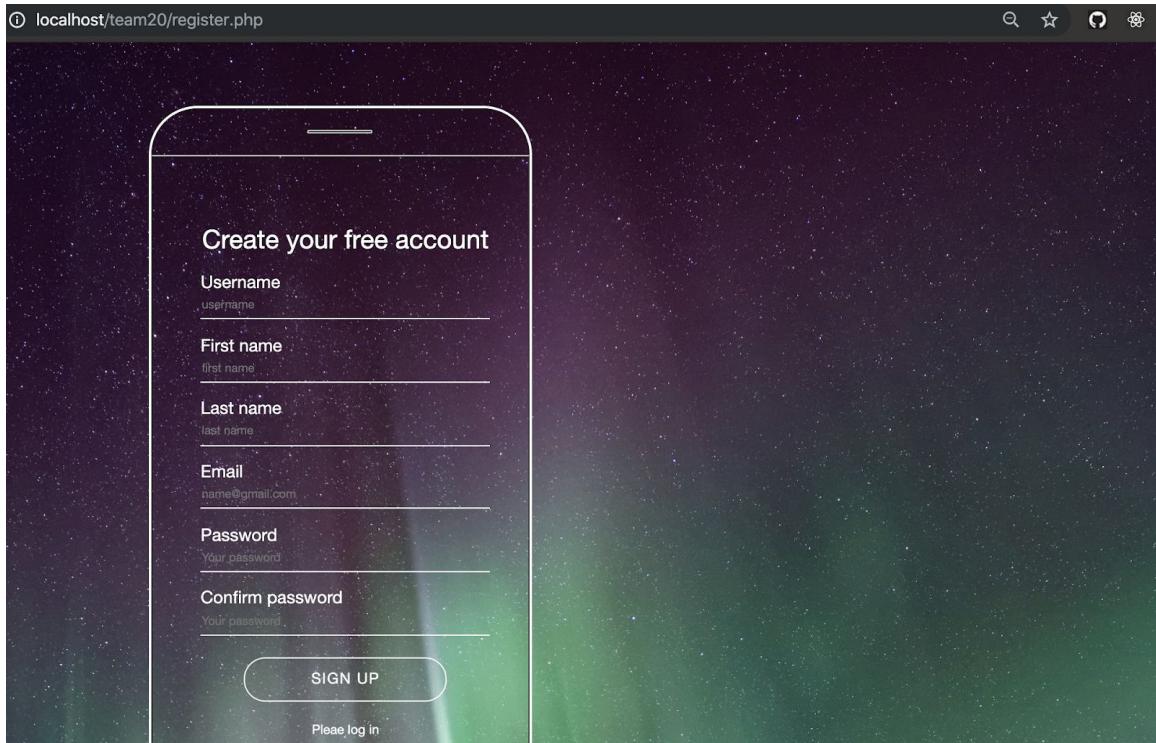
```

- After creating these CSS files, then our register.php becomes the following





- Look at this picture, the address is “localhost/team20/register.php”, that means you run our project successfully since it runs on apache.
- Since we don’t have an account for the first time, we need to create a new account
- We click “if you don’t have an account, please sign up”
- After you click this statement, the page turns to:



- When you typing your information, it will automatically insert to our database,(table: users), for example:

This screenshot shows the same registration form as the previous one, but with different input values:

- Username: cs157aFinalProject
- First name: final
- Last name: project
- Email: cs157afinal@gmail.com
- Password: (represented by four dots)
- Confirm password: (represented by four dots)

Below the form is a "SIGN UP" button and a "Please log in" link.

| MariaDB [team20]> select * from users; | | | | | | | |
|--|----------|------------|-----------------------|--|---------------------|---------------------------------------|------------|
| id | username | firstName | lastName | email | password | signUpDate | profilePic |
| 1 selina | Selina | Ding | Selina@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 2 tim | Tim | Cook | Tim@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 3 mike | Mike | Wu | Mike@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 4 JohnSmith | Jhon | smith | John@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 5 jack | Jack | Jack | Jack@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 6 rose | Rose | Jack | Rose@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 7 tom | Tom | Zheng | Tom@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 8 jerry | Tom | Jerry | Jerry@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 9 newton | New | Ton | Newton@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 10 clinton | Clinton | Clinton | Clinton@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 11 tesla | Tesla | Abc | Tesla@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 12 mickey | Mickey | Mouse | Mickey@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 13 lili | Lily | Lili | Lily@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 14 lucy | Lucy | Dsfadsaf | Lucy@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 15 lilei | Li | Lei | Lilei@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 16 apple | Ios | Tim | Apple@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-11-16 08:00:00 | assets/images/profile/head.png | |
| 17 jobfair | job | fair | job@gmail.com | 5f4dcc3b5aa765d61d8327de882cf99 | 2017-06-28 08:00:00 | resource/images/profile-pics/head.png | |
| 18 mackookPro15 | macbook | pro15 | mac@gmail.com | 5f4dcc3b5aa765d61d8327de882cf99 | 2019-11-28 02:47:27 | resource/images/profile/head.png | |
| 21 M.Craft | M. | Craft | m@gmail.com | 5f4dcc3b5aa765d61d8327de882cf99 | 2019-11-28 03:07:10 | resource/images/profile/head.png | |
| 22 lambdaExpression | lambda | Expression | lambda@gmail.com | 5f4dcc3b5aa765d61d8327de882cf99 | 2019-11-16 08:00:00 | resource/images/profile/head.png | |
| 34 cs157aFinalProject | Final | Project | Cs157aFinal@gmail.com | 8cb2237d0679ca88db6464eac68da96345513964 | 2019-12-08 08:00:00 | resource/images/profile/head.png | |
| 33 lambdaSortByEmail | sort | byEmail | by@gmail.com | 5f4dcc3b5aa765d61d8327de882cf99 | 2019-11-30 23:03:02 | resource/images/profile/head.png | |

22 rows in set (0.000 sec)

- Look at row 34, that is the new data we just inserted into our DB.
- There is a functionality for a registered event in handlers/register-handler.php and classes/Account.php
- And as you can see in the users table, we also have handled password security

```
<?php
class Account {
    private $con;
    private $errorArray;

    public function __construct($con) {
        $this->con = $con;
        $this->errorArray = array();
    }

    public function login($un, $pw)
    {
        $pw = md5($pw);
        $pw = sha1($pw); //The sha1() function uses the US Secure Hash Algorithm 1.
        $query=mysqli_query($this->con,"SELECT * FROM users WHERE username='$un' AND password='$pw'");
        if (mysqli_num_rows($query) == 1) {
            return true;
        } else {
            array_push($this->errorArray, "Your username or password was incorrect");
            return false;
        }
    }

    /**
     * For security function
     */
    public function encryptPassword($password){
        $salt1 = "*&g!";
        $salt2 = "hb%";
        $token = hash('ripemd128', "$salt1$password$salt2");
        return $token;
    }
}
```

- Look at this piece of codes, we call the sha1 function, and we also have our encryptPassword() function
- After called these two functions, the passwords are stored encrypted in the database

Database changed

```
MariaDB [team20]> select * from users;
```

| id username firstName lastName email password signUpDate |
|---|
| 1 selina Selina Ding Selina@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 2 tim Tim Cook Tim@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 3 mike Mike Wu Mike@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 4 JohnSmith Jhon smith John@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 5 jack Jack Jack Jack@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 6 rose Rose Jack Rose@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 7 tom Tom Zheng Tom@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 8 jerry Tom Jerry Jerry@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 9 newoton New Ton Newton@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 10 clinton Clinton Clinton Clinton@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 11 tesla Tesla Abc Tesla@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 12 mickey Mickey Mouse Mickey@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 13 lili Lily Abfs Lily@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 14 lucy Lucy Dsfadsaf Lucy@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 15 lilei Li Lei Lilei@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 16 apple Ios Tim Apple@gmail.com 8cb2237d0679ca88db6464eac60da96345513964 2019-11-16 00:00:00 |
| 17 jobfair job fair job@gmail.com 5f4dcc3b5aa765d61d8327deb882cf99 2017-06-28 00:00:00 |
| 18 mackookPro15 macbook nro15 mac@gmail.com 5f4dcc3b5aa765d61d8327deb882cf99 2019-11-28 02:47:27 |

- That means, whatever password the user input, no one can see it.

```
EXPLORER          Album.php      navBarComponent.php      register-handler.php      Account.php ×
> OPEN EDITORS
  < TEAM20
    < components
      < navBarComponent.php
      < playingBarComponent.....
    < resource
      < css
        albums.css
        iphone.css
        register.css
        song.css
        style.css
      < images
      < js
        register.js
        songAudio.js
      < music
      < src
        < AJAX
          getAlbumConvertToJ...
          getArtistConvertToJ...
          getSongConvertToJ...
          updatePlayCounts.php
        < classes
          Account.php
          Album.php
          Artists.php
          randomTenSongs.php
          Songs.php
        < handlers
          login-handler.php
          register-handler.php
          switchBtn.php
        config.php
        manipulateDatabase.p...
        albumPage.php
        index.php
        register.php
      
```

```
<?php
class Account {
    private $con;
    private $errorArray;

    public function __construct($con) {
        $this->con = $con;
        $this->errorArray = array();
    }

    public function login($un, $pw) {
        $pw = md5($pw);
        $pw = sha1($pw); //The sha1() function uses the US Secure Hash Algorithm 1.
        $query=mysqli_query($this->con,"SELECT * FROM users WHERE username='$un' AND password='$pw'");
        if(mysqli_num_rows($query) == 1) {
            return true;
        } else {
            array_push($this->errorArray, "Your username or password was incorrect");
            return false;
        }
    }

    /**
     * For security function
     */
    public function encryptPassword($password){
        $salt1 = "$g!";
        $salt2 = "hb%";
        $token = hash('ripemd128', "$salt1$password$salt2");
        return $token;
    }

    public function register($un, $fn, $ln, $em, $pw, $pw2) {
        $this->validateUsername($un);
        $this->validateFirstName($fn);
        $this->validateLastName($ln);
        $this->validateEmails($em);
        $this->validatePasswords($pw, $pw2);

        if(empty($this->errorArray) == true) { //empty - Determine whether a variable is empty
            //Insert into db
            return $this->insertUserDetails($un, $fn, $ln, $em, $pw);
        }
        else {
            return false;
        }
    }
}
```

```

EXPLORER
OPEN EDITORS
TEAM20
components
resource
src
AJAX
classes
Account.php
Album.php
Artists.php
randomTenSongs.php
Songs.php
handlers
login-handler.php
register-handler.php
switchBtn.php
config.php
manipulateDatabase.php
albumPage.php
index.php
register.php

register-handler.php
Account.php

1 <?php
2
3 if(isset($_POST['registerButton'])) {
4     //Register button was pressed
5     $username = sanitizeFormUsername($_POST['username']);
6     $firstName = sanitizeFormString($_POST['firstName']);
7     $lastName = sanitizeFormString($_POST['lastName']);
8     $email = sanitizeFormString($_POST['email']);
9     $password = sanitizeFormPassword($_POST['password']);
10    $password2 = sanitizeFormPassword($_POST['password2']);
11
12    $wasSuccessful = $account->register($username, $firstName, $lastName, $email, $password, $password2);
13
14    if($wasSuccessful == true) {
15        $_SESSION['userLoggedIn'] = $username;
16        header("Location: index.php");
17    } else{
18        echo "error";
19    }
20    echo $firstName; //test if successful
21
22
23 function sanitizeFormPassword($inputText) {
24     $inputText = strip_tags($inputText);
25     return $inputText;
26 }
27
28 function sanitizeFormUsername($inputText) {
29     //The strip_tags() function is used to strip HTML and PHP tags from a string.
30     $inputText = strip_tags($inputText);
31     $inputText = str_replace(" ", "", $inputText);
32     return $inputText;
33 }
34
35 function sanitizeFormString($inputText) {
36     $inputText = strip_tags($inputText);
37     $inputText = str_replace(" ", "", $inputText);
38     //The ucfirst() function is used to convert the first character of a string to uppercase.
39     $inputText = ucfirst(strtolower($inputText));
40     return $inputText;
41 }

```

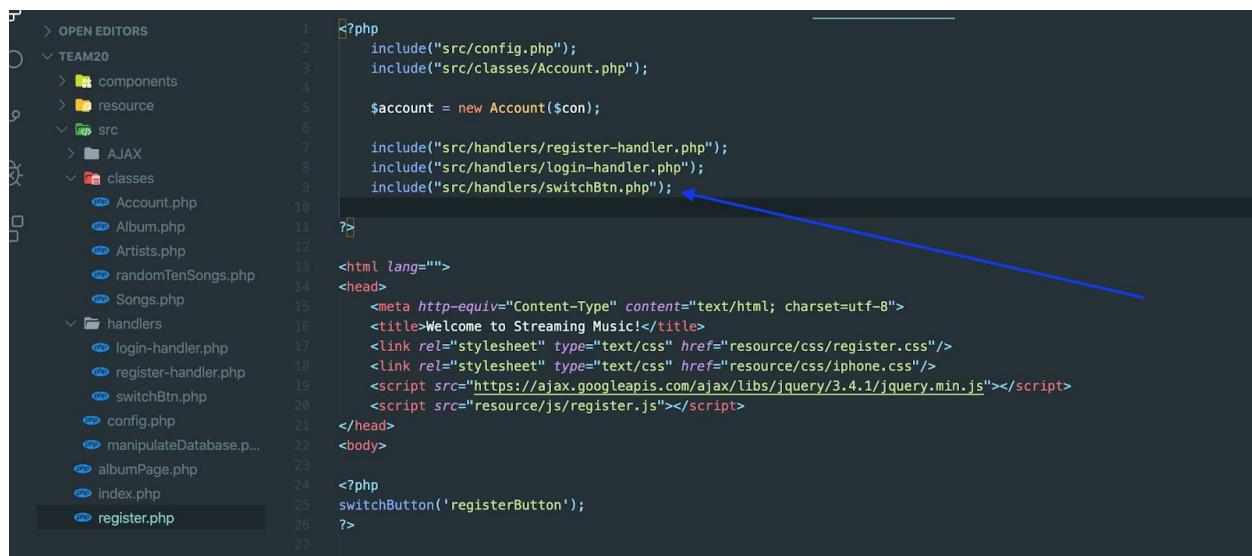
- If we don't have an account, we call the switchButton() function to help switch page

```

OPEN EDITORS
TEAM20
components
resource
src
AJAX
classes
Account.php
Album.php
Artists.php
randomTenSongs.php
Songs.php
handlers
login-handler.php
register-handler.php
switchBtn.php
config.php

switchBtn.php
1 <?php
2
3 function switchButton($buttonName){
4     if(isset($_POST[$buttonName])) {
5         echo '<script>';
6             $(document).ready(function() {
7                 $("#loginForm").hide();
8                 $("#registerForm").show();
9             });
10            </script>';
11     }
12     else {
13         echo '<script>';
14             $(document).ready(function() {
15                 $("#loginForm").show();
16                 $("#registerForm").hide();
17             });
18            </script>';
19     }
20 ?>

```



```

<?php
    include("src/config.php");
    include("src/classes/Account.php");

    $account = new Account($con);

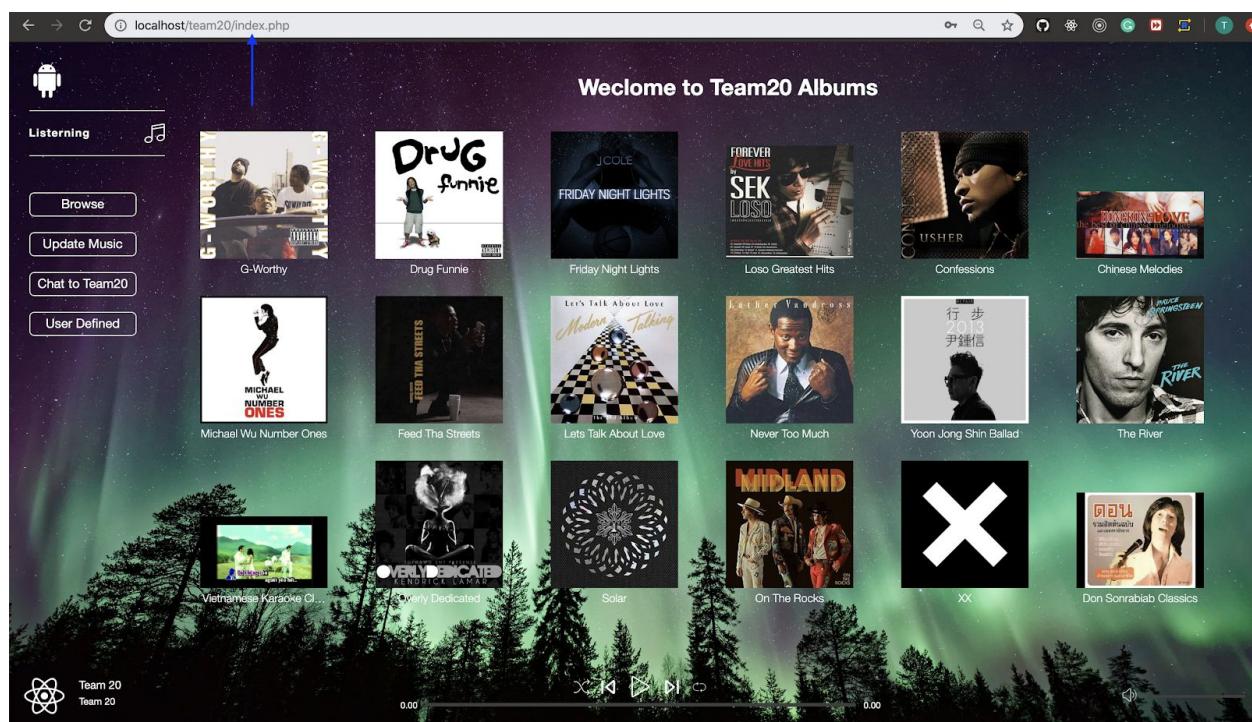
    include("src/handlers/register-handler.php");
    include("src/handlers/login-handler.php");
    include("src/handlers/switchBtn.php");
?>

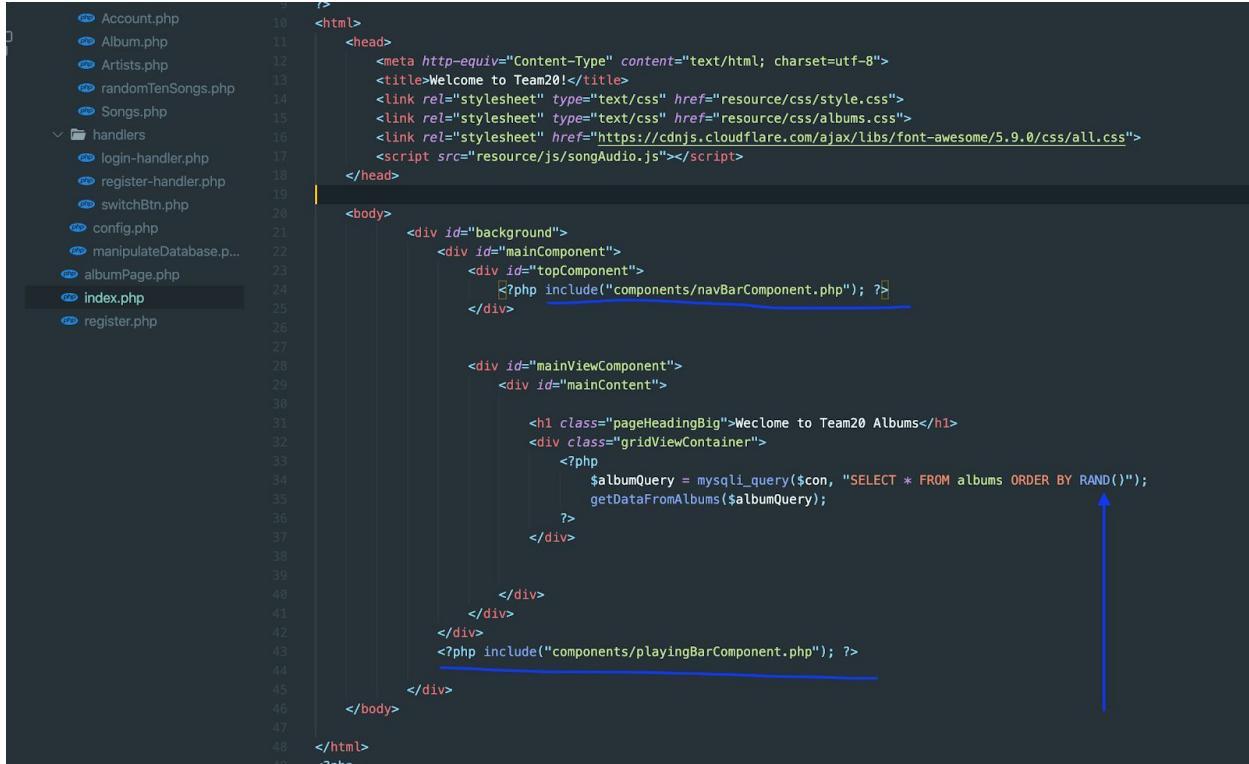
<html lang="">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Welcome to Streaming Music!</title>
    <link rel="stylesheet" type="text/css" href="resource/css/register.css"/>
    <link rel="stylesheet" type="text/css" href="resource/css/iphone.css"/>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="resource/js/register.js"></script>
</head>
<body>

<?php
switchButton('registerButton');
?>

```

Event 2: After login, we browse all albums and go to index.php



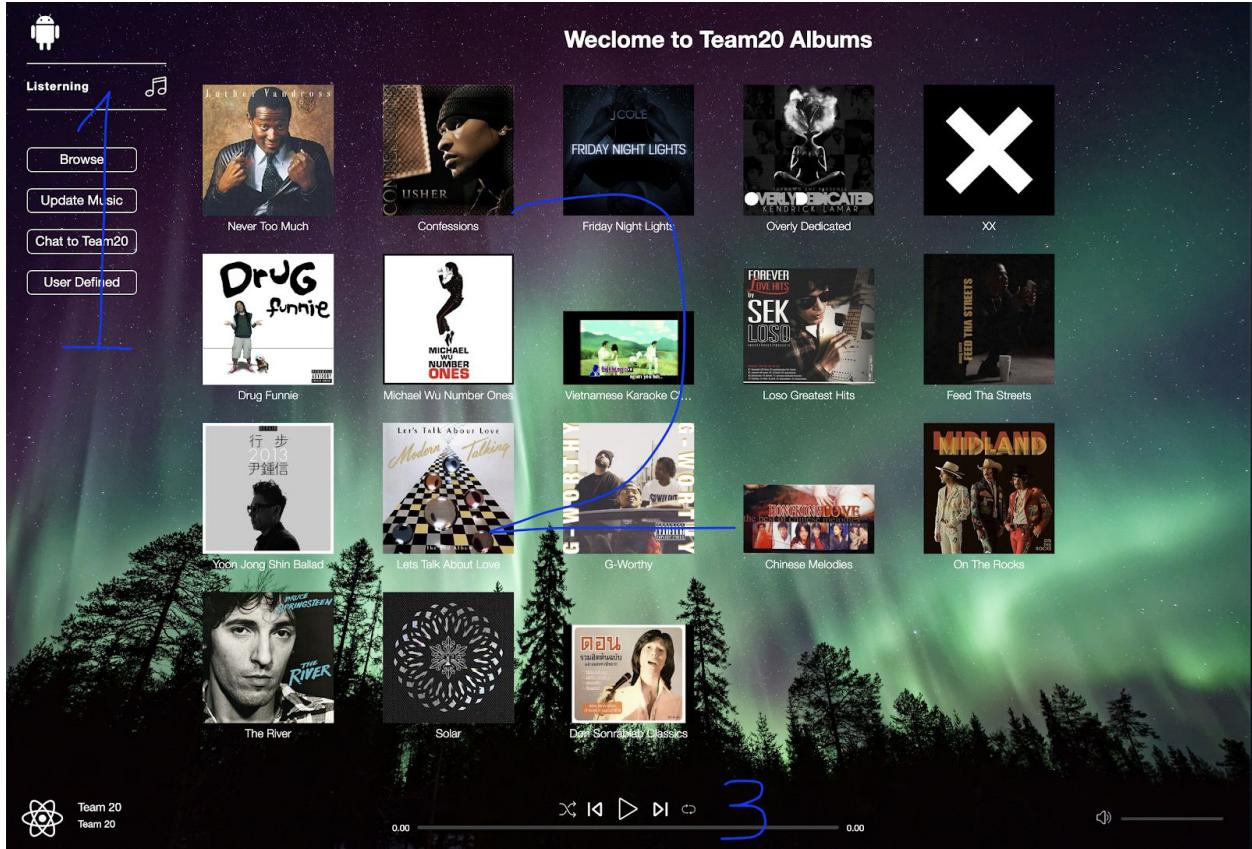


```

 9  </>
10 <html>
11   <head>
12     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
13     <title>Welcome to Team20!</title>
14     <link rel="stylesheet" type="text/css" href="resource/css/style.css">
15     <link rel="stylesheet" type="text/css" href="resource/css/albums.css">
16     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.9.0/css/all.css">
17     <script src="resource/js/songAudio.js"></script>
18   </head>
19
20 <body>
21   <div id="background">
22     <div id="mainComponent">
23       <div id="topComponent">
24         <?php include("components/navBarComponent.php"); ?>
25       </div>
26
27       <div id="mainViewComponent">
28         <div id="mainContent">
29           <h1 class="pageHeadingBig">Weclome to Team20 Albums</h1>
30           <div class="gridViewContainer">
31             <?php
32               $albumQuery = mysqli_query($con, "SELECT * FROM albums ORDER BY RAND()");
33               getDataFromAlbums($albumQuery);
34             ?>
35           </div>
36
37         </div>
38       </div>
39
40     </div>
41   </div>
42
43   <?php include("components/playingBarComponent.php"); ?>
44
45 </div>
46 </body>
47
48 </html>

```

- We separate the navigation bar, homepage, and playing bar
- There are three parts in the index.php



- We can see three parts on this page, 1: navigation bar, 2. User homepage, 3. playing bar
- For part1 and part3's source codes, you can review:

Screenshot of a code editor showing the file structure and content of index.php.

EXPLORER

- OPEN EDITORS
- TEAM20
 - components
 - navBarComponent.php
 - playingBarComponent.php
 - resource
 - AJAX
 - classes
 - Account.php
 - Album.php
 - Artists.php
 - randomTenSongs.php
 - Songs.php
 - handlers
 - login-handler.php
 - register-handler.php
 - switchBtn.php

index.php — team20

```

<?php
include("src/config.php"); //run session_start() firstly
include("src/manipulateDatabase.php"); //manipulate data from our database
//session_destroy(); //this function will make you every time log out
afterUserLogin();
?>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Welcome to Team20!</title>
    <link rel="stylesheet" type="text/css" href="resource/css/style.css">
    <link rel="stylesheet" type="text/css" href="resource/css/albums.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.9.0/css/all.min.css" type="text/css">
    <script src="resource/js/songAudio.js"></script>
  </head>
  <body>
    <div id="background">
      <div id="mainComponent">
        <div id="topComponent">

```

- Part 1 and Part 3 is HTML with CSS codes

```

1  this navBarComponent is the interface in index.php -->
2  id="navBarComponent">
3      <nav class="navBar">
4          <a href="index.php" class="logo">
5              
6          </a>
7
8          <div class="group">
9              <div class="navItem">
10                 <a href="index.php" class="navItemLink">Listerning
11                     
12                 </a>
13             </div>
14         </div>
15
16         <div class="group">
17             <div class="navItem">
18                 <!-- <a href="browse.php" class="navItemLink">Browse</a> -->
19                 <button class="navItemLinkBtn" id="browse" onclick="window.location.href = 'http://localhost/team20/index.php';">
20                     Browse
21                 </button>
22             </div>
23             <div class="navItem">
24                 <button class="navItemLinkBtn" onclick="window.location.href = 'http://localhost/team20/index.php';">
25                     Update Music
26                 </button>
27             </div>
28             <div class="navItem">
29                 <!-- <a href="http://localhost:8080" class="navItemLink">Team 20</a> -->
30                 <button class="navItemLinkBtn" onclick="window.location.href = 'https://novemberfall.github.io/musicChat/';">
31                     Chat to Team20
32                 </button>
33             </div>
34             <div class="navItem">
35                 <!-- <a href="http://localhost:3000" class="navItemLink">User Defined</a> -->
36                 <button class="navItemLinkBtn" onclick="window.location.href = 'http://localhost:3000';">
37                     User Defined
38                 </button>
39             </div>
40         </div>
41     </nav>
42

```

This is part1: navBarComponent.php

- We can see there are some links to link other functionalities(other pages).

```

<body>
    <div id="background">
        <div id="mainComponent">
            <div id="topComponent">
                <?php include("components/navBarComponent.php"); ?>
            </div>

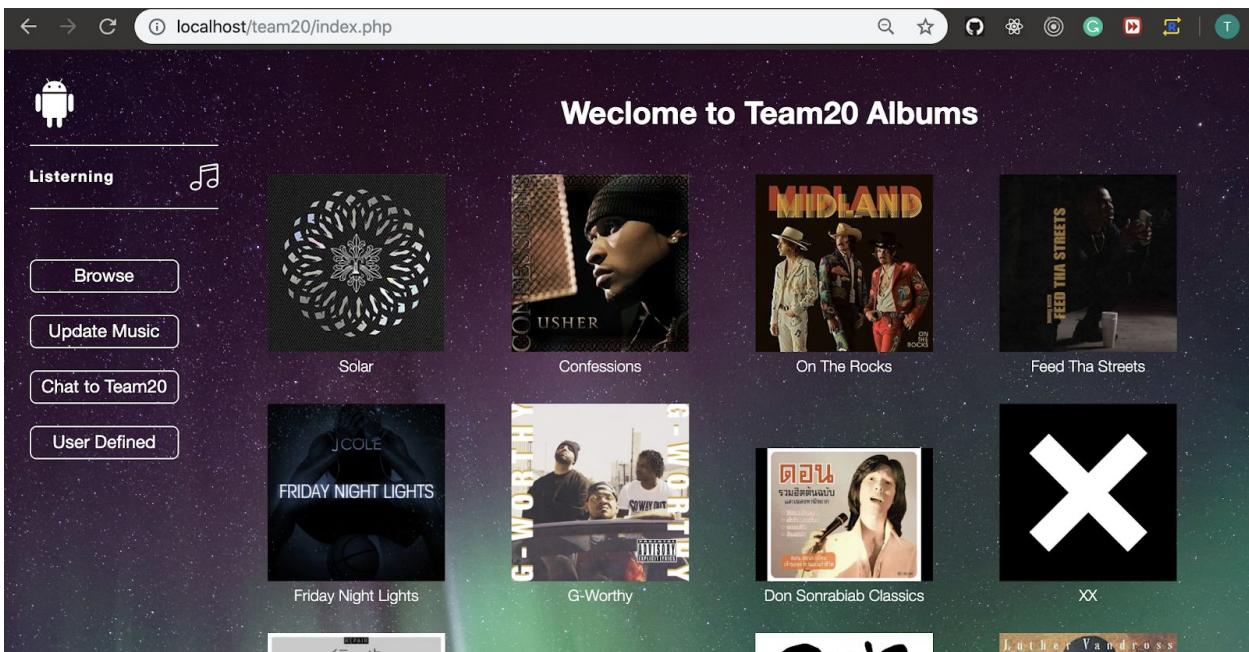
            <div id="mainViewComponent">
                <div id="mainContent">
                    <h1 class="pageHeadingBig">Weclome to Team20 Albums</h1>
                    <div class="gridViewContainer">
                        <?php
                            $albumQuery = mysqli_query($con, "SELECT * FROM albums ORDER BY RAND()");
                            getDataFromAlbums($albumQuery);
                        ?>
                    </div>
                </div>
            </div>
            <?php include("components/playingBarComponent.php"); ?>
        </div>
    </body>

```

- The index.php, we have this statement: SELECT * FROM albums ORDER BY RAND()

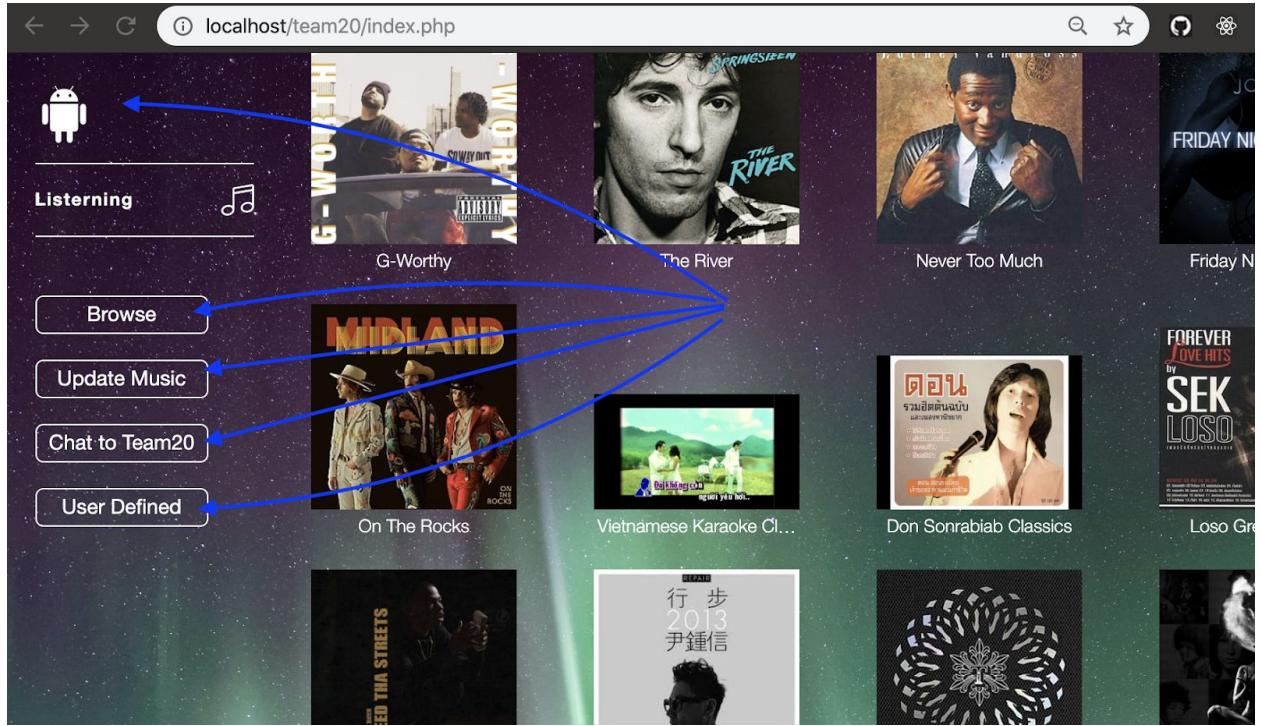
That means every time you refresh the homepage, it will display different albums

For example:

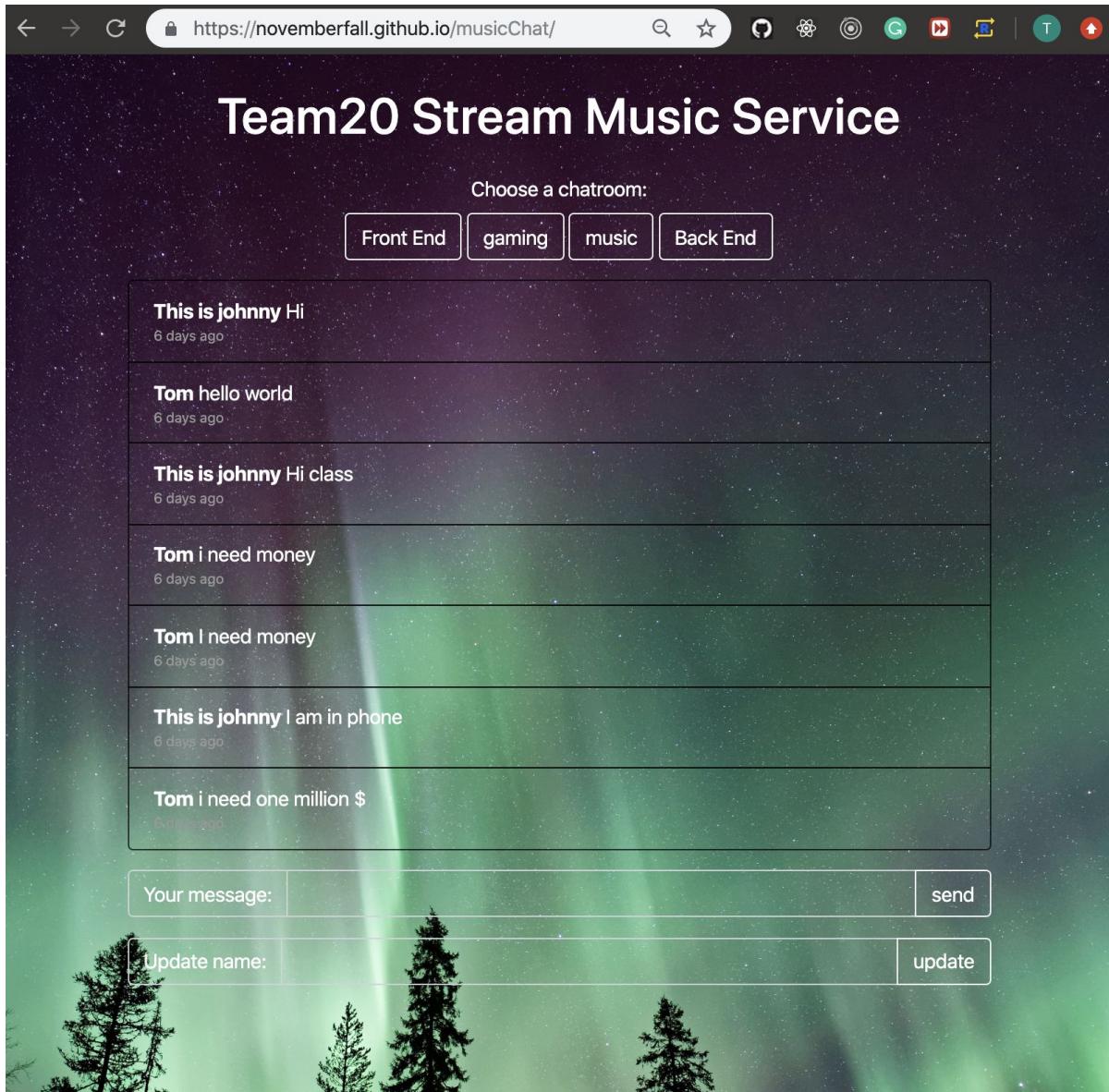


- Obviously, the order has changed.

Then, let us talk about the navigation bar firstly.



1. We can see the Android robot icon, this is a users' icons, but for here we use only one icon just represents the different user profile picture.
2. If you click 'Browse' button, that means you still go to index.php, you browse all albums by random order.
3. Click Team20 button, you will chat to team20 back-end engineers



- This functionality we have shown on our volunteer presentation

```

25      <button class="navItemLinkBtn" onclick="window.location.href = 'http://localhost/team20/index.php';">
26          Update Music
27      </button>
28  </div>
29  <div class="navItem">
30      <!-- <a href="http://localhost:8080" class="navItemLink">Team 20</a> -->
31      <button class="navItemLinkBtn" onclick="window.location.href = 'https://novemberfall.github.io/musicChat/';">
32          Chat to Team20
33      </button>
34  </div>
35  <div class="navItem">
36      <!-- <a href="http://localhost:3000" class="navItemLink">User Defined</a> -->

```

- Here is source codes, you will be linked to the page, then you can talk to us.
- Because for this chatBox, I have updated my github, and anyone can use it
- It's similar to a real-life online chat room.

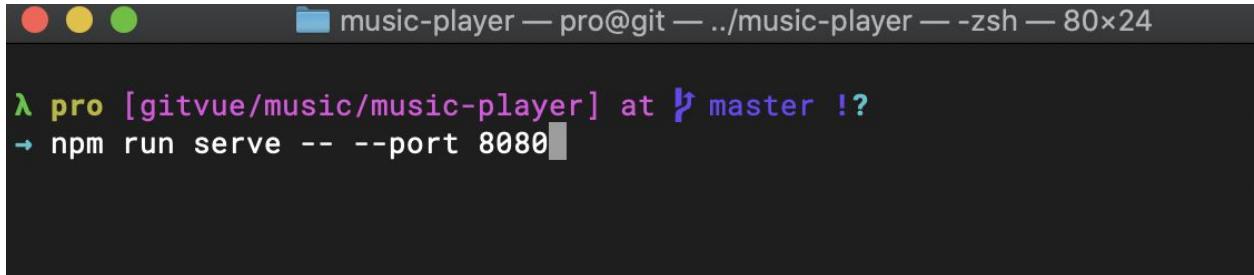
4. If you click update music button, you will jump to JDBC back-end homepage, for example, some user hope can update his own music, then he can tell us, the back-end engineer will help them update the songs.
5. If the user clicks 'User Defined' button, that means he can play his song locally

```
</div>
<div class="navItem">
    <!-- <a href="http://localhost:3000" class="navItemLink">User Defined</a> -->
    <button class="navItemLinkBtn" onclick="window.location.href = 'http://localhost:3000';">
        User Defined
    </button>
</div>
</div>
```

- Look at the last statement, it will jump to `http://localhost:3000`
- This Music Player is written by vue.js
- When user click button 'User Defined', we have a link : `http://localhost:8080/`
- The link is in `navBarComponent.php`

```
</div>
<div class="navItem">
    <!-- <a href="http://localhost:3000" class="navItemLink">User Defined</a> -->
    <button class="navItemLinkBtn" onclick="window.location.href = 'http://localhost:8080';">
        User Defined
    </button>
</div>
</div>
```

- Then we on mac terminal, cd to the folder, run

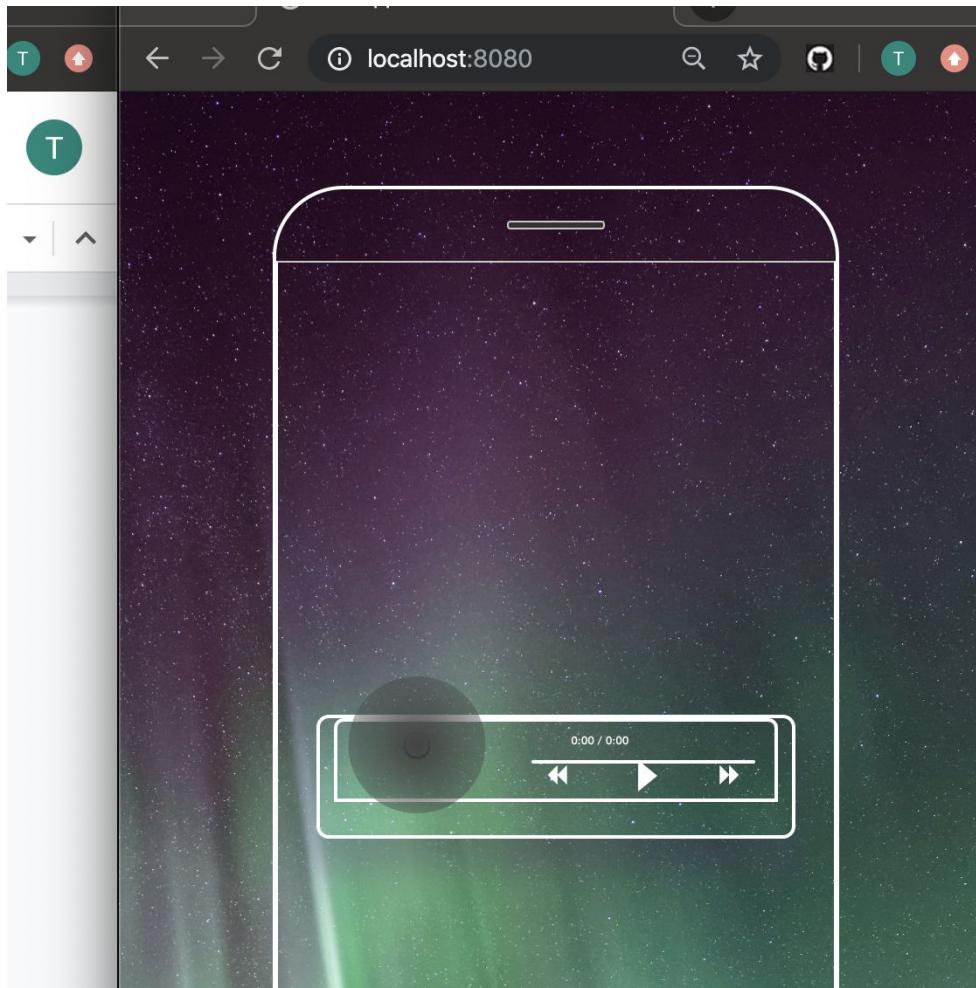


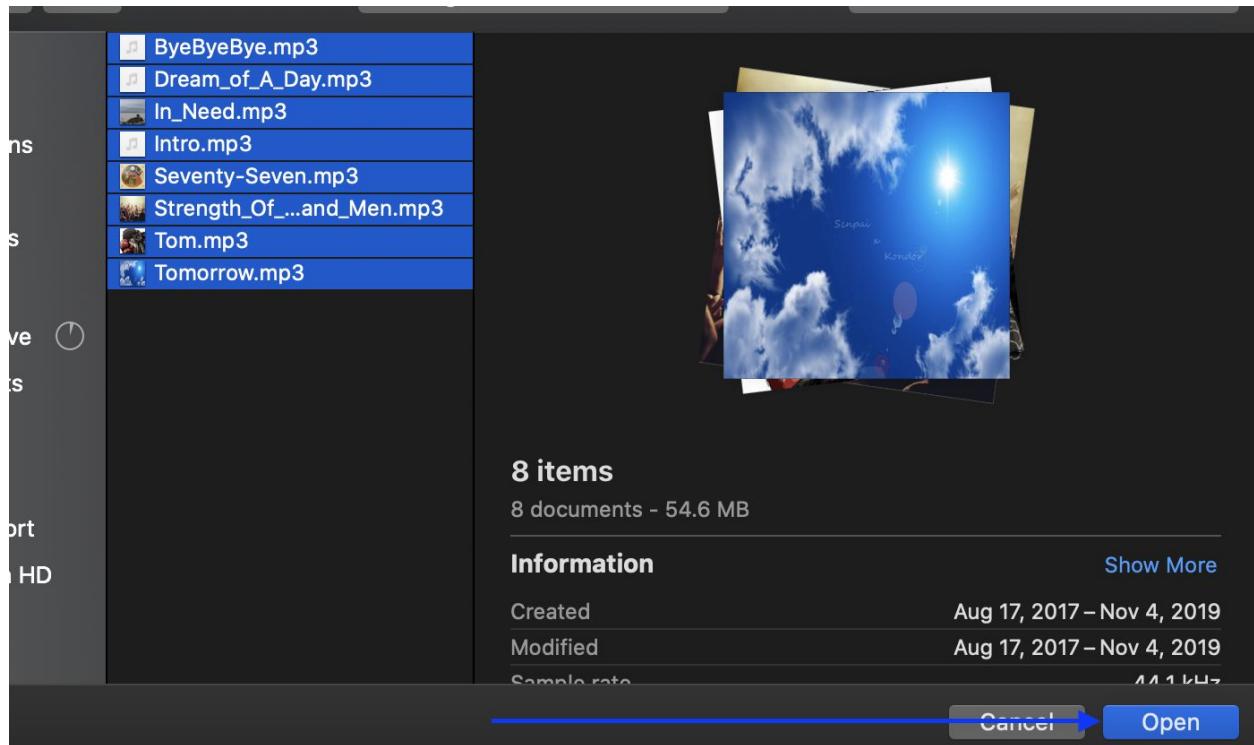
```
● ● ● music-player — pro@git — ..../music-player — -zsh — 80x24
λ pro [gitvue/music/music-player] at ↵ master !?
→ npm run serve -- --port 8080
```

```
music-player — npm run serve -- --port 8080 — npm — node ↵ npm TMPDIR=/var/ ↵ DONE Compiled successfully in 1469ms 6

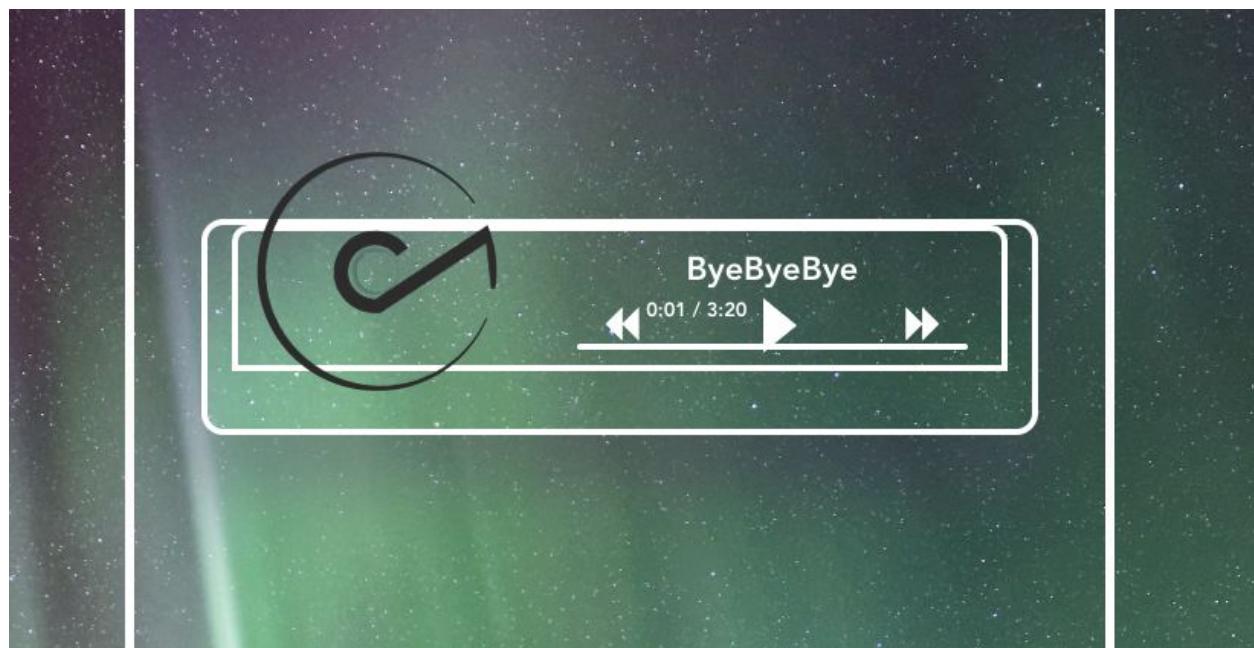
App running at:
- Local:  http://localhost:8080/
- Network: http://192.168.1.13:8080/

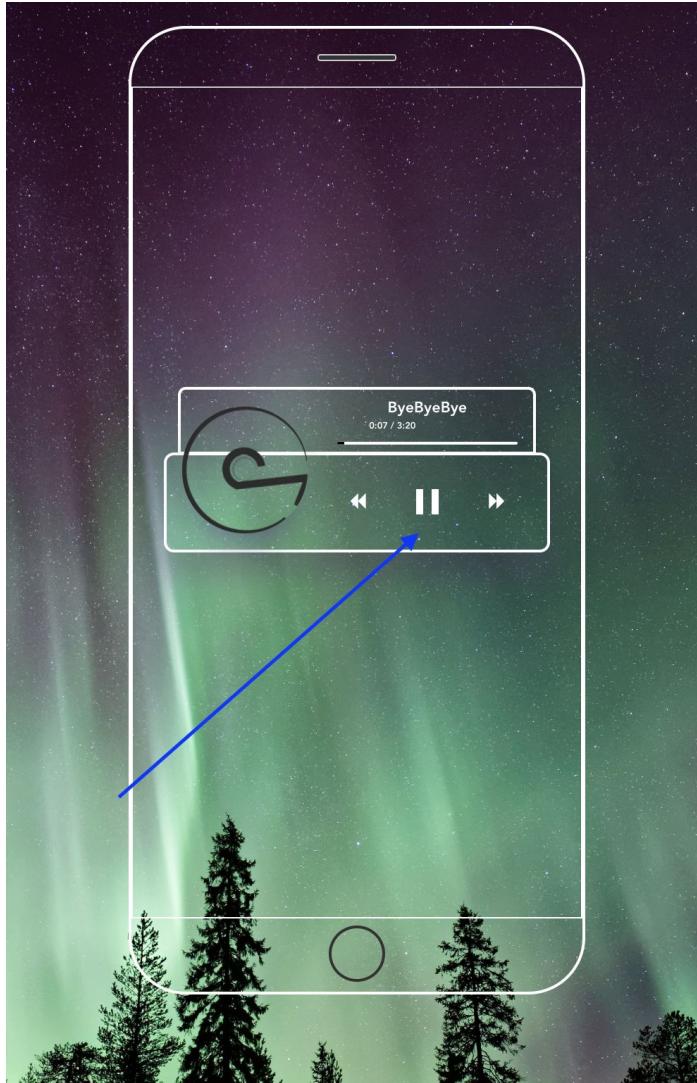
Note that the development build is not optimized.
To create a production build, run npm run build.
```





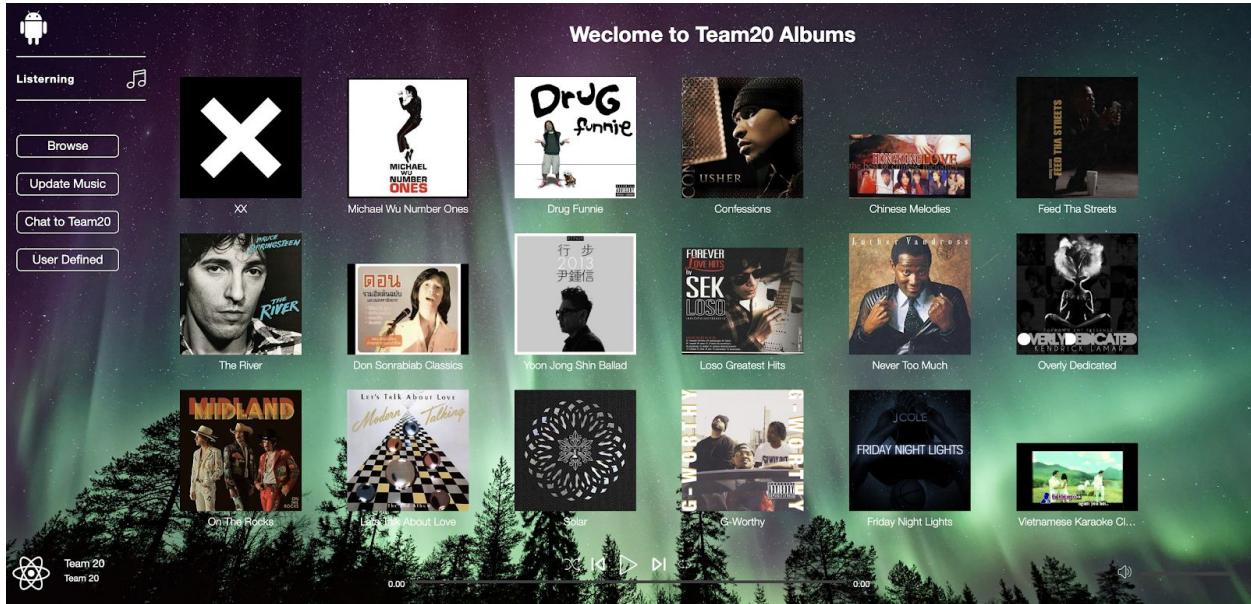
- We can choose these songs
- That means, sometimes the users hope that they can play music by themselves.



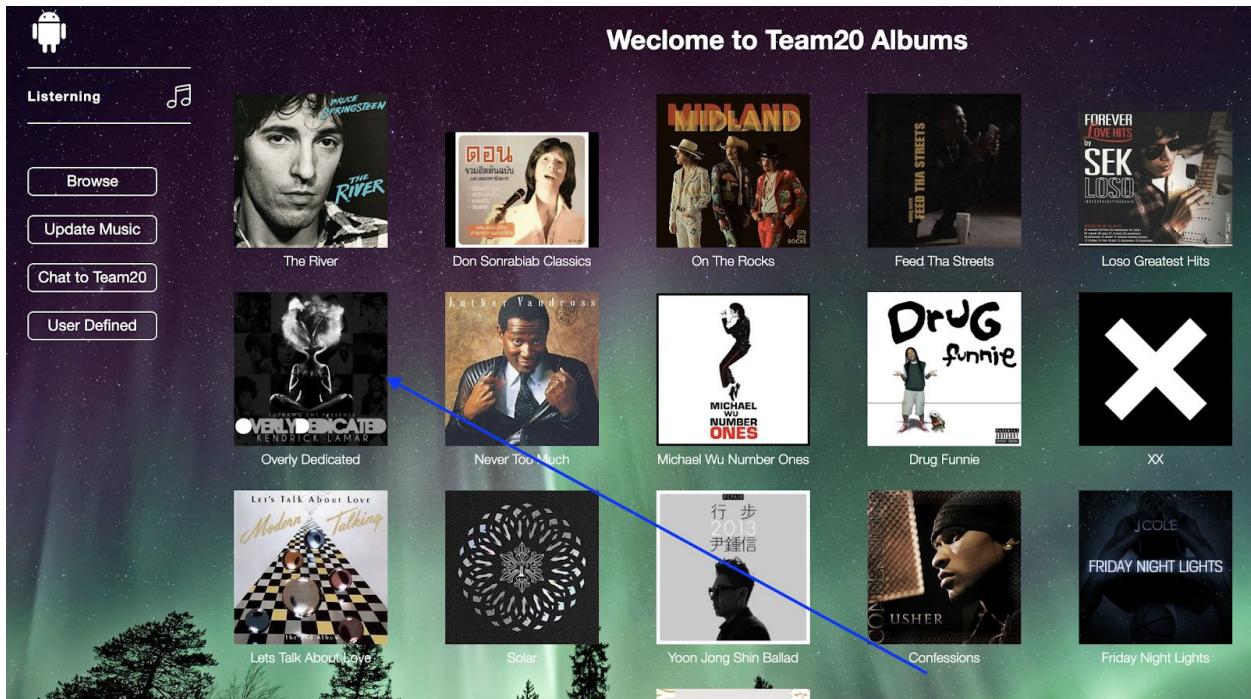


- When clicking the 'play' button, you can play the music
- You also can click go to next song, or previous song

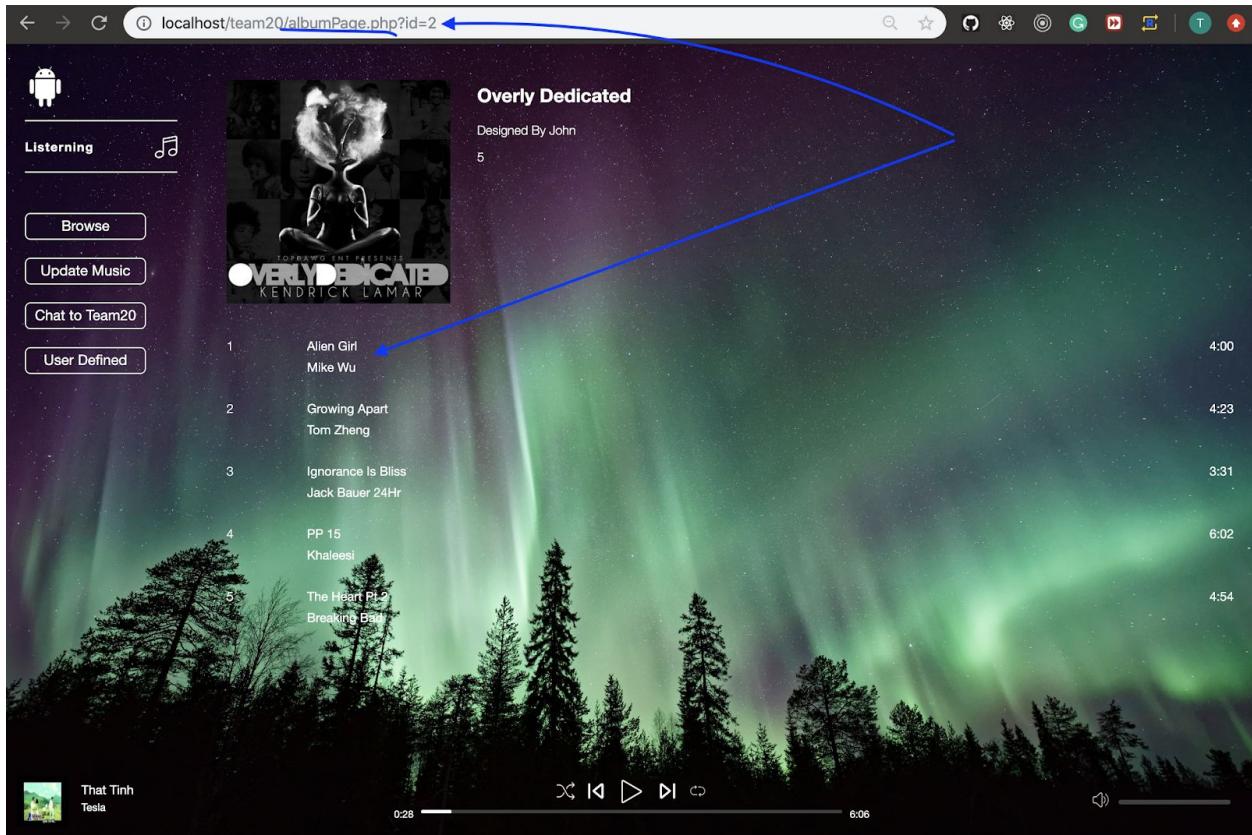
Event 2: the user's homepage(Part 2)



- We mentioned this page on Part1: navigation bar
- The user can pick one of the albums which is his favorite
- For example, users pick 'Overly Dedicated', click this album



- And we see the address following id = 2, that means you pick the album which id = 2



- Look at the source codes from albumPage.php,

```

albumPage.php X
33     songAudioElement.playBGM();
34 </script> -->
35
36     <div id="background">
37         <div id="mainComponent">
38             <div id="topComponent">
39                 <?php include("components/navBarComponent.php"); ?>
40             </div>
41
42
43         <div id="mainViewComponent">
44             <div id="mainContent">
45
46                 <?php
47                     if(isset($_GET['id'])){
48                         $albumId = $_GET['id'];
49                         // echo "id= " . $albumId. "<br>";
50                     }else{
51                         header("Location: index.php");
52                     }
53                     $albumQuery = mysqli_query($con, "SELECT * FROM albums WHERE id='$albumId'");
54                     $album = mysqli_fetch_array($albumQuery);
55                 ?>
56
57                 <div class="entityInfo">
58                     <div class="leftPart">
59                         
60                     </div>
61
62                     <div class="rightPart">
63                         <h2><?php echo getAlbumTitle($con, $albumId); ?></h2>
64                         <p>Designed By <?php echo getArtistName($con, $albumId); ?></p>
65                         <p><?php echo getNumbersOfSongs($con, $albumId); ?></p>
66                     </div>
67
68                 </div>

```

When you click this album, \$albumId = \$_GET['id'] that means we assign the id to \$albumId

```
SELECT * FROM albums WHERE id=$albumId
```

Then we show this album which id = \$albumId on the album page

```

<div id="mainViewComponent">
    <div id="mainContent">

        <?php
            if(isset($_GET['id'])){
                $albumId = $_GET['id'];
                // echo "id= " . $albumId. "<br>";
            }else{
                header("Location: index.php");
            }
            $albumQuery = mysqli_query($con, "SELECT * FROM albums WHERE id='$albumId'");
            $album = mysqli_fetch_array($albumQuery);
        ?>

        <div class="entityInfo">
            <div class="leftPart">
                
            </div>

            <div class="rightPart">
                <h2><?php echo getAlbumTitle($con, $albumId); ?></h2>
                <p>Designed By <?php echo getArtistName($con, $albumId); ?></p>
                <p><?php echo getNumbersOfSongs($con, $albumId); ?></p>
            </div>
        </div>
    </div>

```



For here, we push this album's data on the album page, you can see the album's picture, title, designer, and the number of songs.

```

<?php
    if(isset($_GET['id'])){
        $albumId = $_GET['id'];
        // echo "id=" . $albumId. "<br>";
    }else{
        header("Location: index.php");
    }
    $albumQuery = mysqli_query($con, "SELECT * FROM albums WHERE id='$albumId'");
    $album = mysqli_fetch_array($albumQuery);
?>

```

Again, because this php function (in albumPage.php), so we get the albumId

In albumPage.php, we still have a track list event:

```

<div class="tracklistContainer">
    <ul class="tracklist">
        1
        <?php
        $songIdArray = getSongIds($con, $albumId);
        $i = 1;
        foreach($songIdArray as $albumId) { 2
            3
            $albumSong = new Song($con, $albumId);
            $albumArtist = getArtistName($con, $albumId); 4
            echo "<li class='tracklistRow'>
                <div class='trackCount'>
                    <img class='play-btn' src='resource/images/icons/play-btn.png'
                        onclick='setTrackFunction(" . $albumSong->getId() . ", tempPlaylist, true)'>
                    <span class='trackNumber'>$i</span> 5
                </div>
                <div class='trackInfo'>
                    <div class='trackName'>" . $albumSong->getTitle() . "</div>
                    <div class='artistName'>" . $albumArtist . "</div> 6
                </div>
                <div class='trackOptions'>
                    <img class='optionsButton' src='resource/images/icons/more.png'> 7
                </div>
                <div class='trackDuration'>
                    <div class='duration'>" . $albumSong->getDuration() . "</div>
                </div>
            </li>"; 8
            $i = $i + 1;
        }
        ?>

        <script>
            var tempSongIds = '<?php echo json_encode($songIdArray); ?>';
            tempPlaylist = JSON.parse(tempSongIds);
            console.log()
        </script>

    </ul>
</div>

```

I have marked this piece of codes, and there are totally 8 steps for list every track list.

```
getSongIds($con, $albumId);
```

Step1: we call this function

```
function getSongIds($connect, $ID) {

    $query = mysqli_query($connect, "SELECT id FROM songs WHERE album='$ID' ORDER BY albumOrder ASC");

    $array = array();
    while($row = mysqli_fetch_array($query)) {
        array_push($array, $row['id']);
    }
    return $array;
}
```

This function help us get the song's id, then we can put it on the track list

Step2: we are using a for loop to traverse every song with its title and artist's name

Step3: we create a new object song with the parameter \$albumId

```
/*
class Song {

    private $connect;
    private $id;
    private $mysql;
    private $title;
    private $artistId;
    private $albumId;
    private $genre;
    private $duration;
    private $path;

    /**
     * A constructor function to create a new object
     */
    public function __construct($con, $id) {
        $this->connect = $con;
        $this->id = $id;

        $query = mysqli_query($this->connect, "SELECT * FROM songs WHERE id='$this->id'");
        $this->mysql = mysqli_fetch_array($query);
        $this->title = $this->mysql['title'];
        $this->artistId = $this->mysql['artist'];
        $this->albumId = $this->mysql['album'];
        $this->genre = $this->mysql['genre'];
        $this->duration = $this->mysql['duration'];
        $this->path = $this->mysql['path'];
    }
}
```

By \$albumId, we found out the song's all information, (codes from Album.php)

Step4: call getArtistName() to get the artist's name:



```

1 <?php
2     function getArtistName($connect, $ID){
3         $artistQuery = mysqli_query($connect, "SELECT name FROM artists WHERE id='$ID'");
4         $artist = mysqli_fetch_array($artistQuery);
5         return $artist['name'];
6     }
7 ?>

```

- Pick its id(albumId), then find out the name of artist

Step5: for this step, lately we will discuss on PlayingBar event (playingBarComponent.php), so we go to the next step

Step6: we call the public function getTitle() from Songs new object we have created



```

34     /**
35      * Get the song's title
36      */
37     public function getTitle() {
38         return $this->title;
39     }
40
41     /**
42      * Get the song's ID
43      */
44     public function getId() {
45         return $this->id;
46     }
47

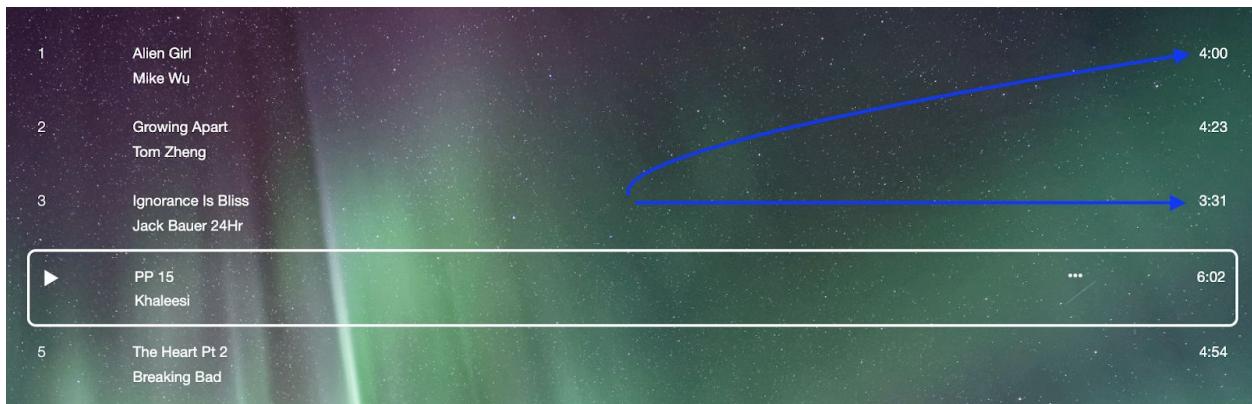
```

Step7: we output the more.png



Step8: we call song's getDuration()

```
/**  
 * get the duration of this song  
 */  
public function getDuration() {  
    return $this->duration;  
}
```



Note: for step5, we will talk on PlayingBar event because we will use AJAX, for AJAX, it uses jQuery library, get the data from database, and try to render our user homepage.

Event 3: PlayingBar on the button, this event we focus on how to play a music, how to pick a song which the user likes, and playing on the playingbar

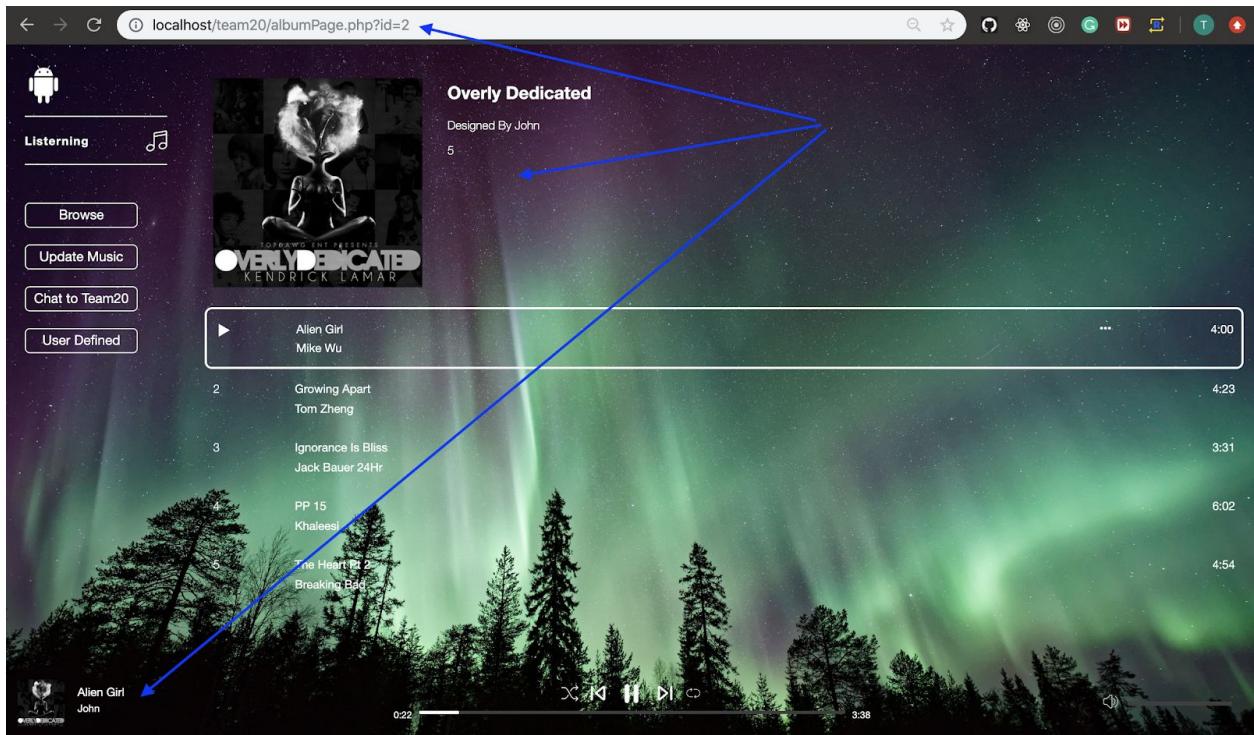
- Look at the playingBarComponent.php

```
5 <div id="playingBarCenter">
6
7     <div class="content playerControls">
8
9         <div class="buttons">
10            <button class="playingBarButton shuffle" title="Shuffle button">
11                
12            </button>
13            <button class="playingBarButton previous" title="Previous button">
14                
15            </button>
16            <button class="playingBarButton play" title="Play button"
17                onclick="audioElement.playOneSong('./src/AJAX/updatePlayCounts.php','.
18                    playingBarButton.play','.
19                    playingBarButton.pause')">
20                
21            </button>
22            <button class="playingBarButton pause" title="Pause button" style="display: none;">
23                onclick="audioElement.pauseOneSong('.playingBarButton.play','.
24                    playingBarButton.pause')">
25                    
26            </button>
27            <button class="playingBarButton next" title="Next button">
28                
29            </button>
30            <button class="playingBarButton repeat" title="Repeat button">
31                
32            </button>
33        </div>
34
35
36        <div class="playbackBar">
37            <span class="progressTime current">0.00</span>
38            <div class="progressBar">
39                <div class="progressBarBg">
40                    <div class="progress" id="progressID"></div>
41                </div>
42            </div>
43            <span class="progressTime remaining">0.00</span>
44        </div>
45
46    </div>
47
48    <div id="playingBarRight">
49        <div class="volumeBar">
50            <button class="playingBarButton volume" title="Volume button">
51                
52            </button>
53            <div class="progressBar">
54                <div class="progressBarBg">
55                    <div class="progress"></div>
56                </div>
57            </div>
58        </div>
59    </div>
60
61 </div>
```

We have 3 parts of this page: playingBarLeft, playingBarCenter, and playingBarRight



Let us go back to albumPage.php



1. We get the albumId
2. We output this album's all info
3. We try to display the album title, artist, and album picture on the left bottom

Look at our setTrackFunction() inside playingBarComponent.php

```
function setTrackFunction(trackId, newPlaylist, playFlag){
    $.post("./src/AJAX/getSongConvertToJson.php", {songId: trackId}, function(data){

        var track = JSON.parse(data);
        console.log(track);
        /**
         * A common use of JSON is to exchange data to/from a web server.
         * When receiving data from a web server, the data is always a string.
         * Parse the data with JSON.parse(), and the data becomes a JavaScript object.
         */
        $(".trackName span").text(track.title);

        $.post("./src/AJAX/getArtistConvertToJson.php", {artistId: track.artist}, function(data){

            currentIndex = currentPlaylist.indexOf(trackId);
            var artist = JSON.parse(data);
            console.log(artist.name);
            $(".artistName span").text(artist.name);
        });

        $.post("./src/AJAX/getAlbumConvertToJson.php", {albumId: track.album}, function(data){

            var album = JSON.parse(data);
            $(".albumLink img").attr("src", album.artworkPath);
        });

        audioElement.setTrack(track);
        audioElement.playOneSong('./src/AJAX/updatePlayCounts.php','playingBarButton.play','playingBarButton.pause');

    });

    if(playFlag == true){
        audioElement.play();
    }
}
```

Try to display all albums' title, pics, artist's name, and duration, we need to use AJAX + jQuery (jQuery is a high level library of JavaScript)

The screenshot shows a file structure on the left and a code editor on the right. The file structure includes a 'src' folder containing an 'AJAX' folder with four PHP files: 'getAlbumConvertToJson.php', 'getArtistConvertToJson.php', 'getSongConvertToJson.php', and 'updatePlayCounts.php'. It also contains a 'classes' folder with 'Account.php' and 'Album.php'. The code editor on the right displays a snippet of JavaScript:

```

49 var track = JSON.parse(data);
50 console.log(track);
51 /**
52 * A common use of JSON is to exchange data to/
53 * When receiving data from a web server, the d
54 * Parse the data with JSON.parse(), and the da
55 */
56 $(".trackName span").text(track.title);
57
58 $.post("./src/AJAX/getArtistConvertToJson.php",
59 currentIndex = currentPlaylist.indexOf(trac

```

So we create a folder named AJAX

The screenshot shows two code editors. The top editor is titled 'playingBarComponent.php' and contains the following PHP code:

```

1 <?php
2 include("../config.php");
3
4 if(isset($_POST['songId'])){
5     $songId = $_POST['songId'];
6
7     $query = mysqli_query($con, "SELECT * FROM Songs WHERE id ='$songId'");
8
9     $resultArray = mysqli_fetch_array($query);
10    //Fetch a result row as a numeric array and as an associative array:
11
12    echo json_encode($resultArray);
13    //The json_encode() function is an inbuilt function in PHP which is used to convert PHP array or object into JSO
14
15 }

```

The bottom editor is titled 'getSongConvertToJson.php' and contains the same code as the top editor.

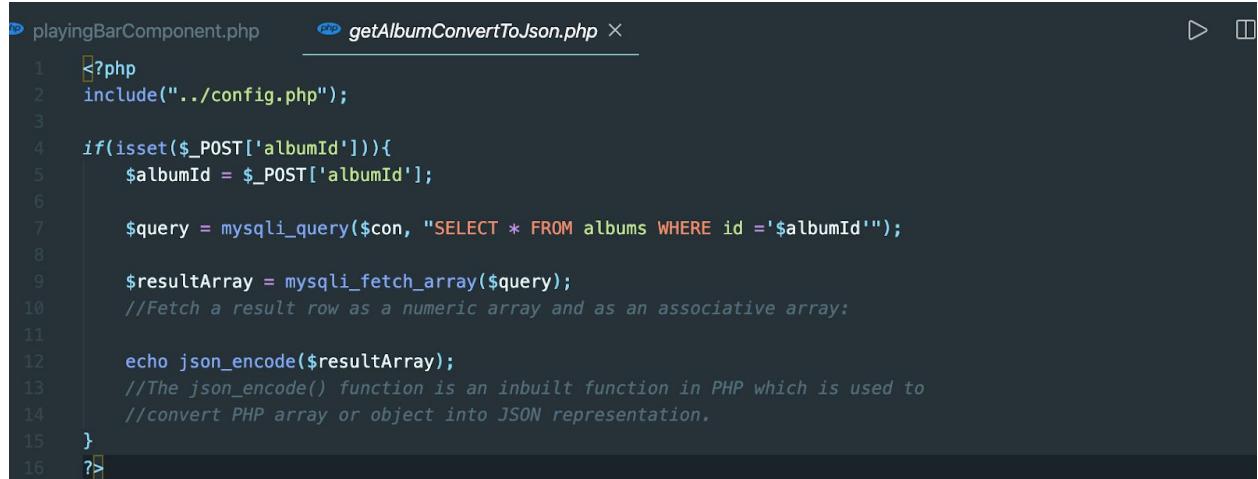
The second editor is titled 'playingBarComponent.php' and contains the following PHP code:

```

1 <?php
2 include("../config.php");
3
4 if(isset($_POST['artistId'])){
5     $artistId = $_POST['artistId'];
6
7     $query = mysqli_query($con, "SELECT * FROM artists WHERE id ='$artistId'");
8
9     $resultArray = mysqli_fetch_array($query);
10    //Fetch a result row as a numeric array and as an associative array:
11
12    echo json_encode($resultArray);
13    //The json_encode() function is an inbuilt function in PHP which is used to
14    //convert PHP array or object into JSON representation.
15 }

```

The bottom editor also has a tab for 'getArtistConvertToJson.php' which contains the same code as the top editor.



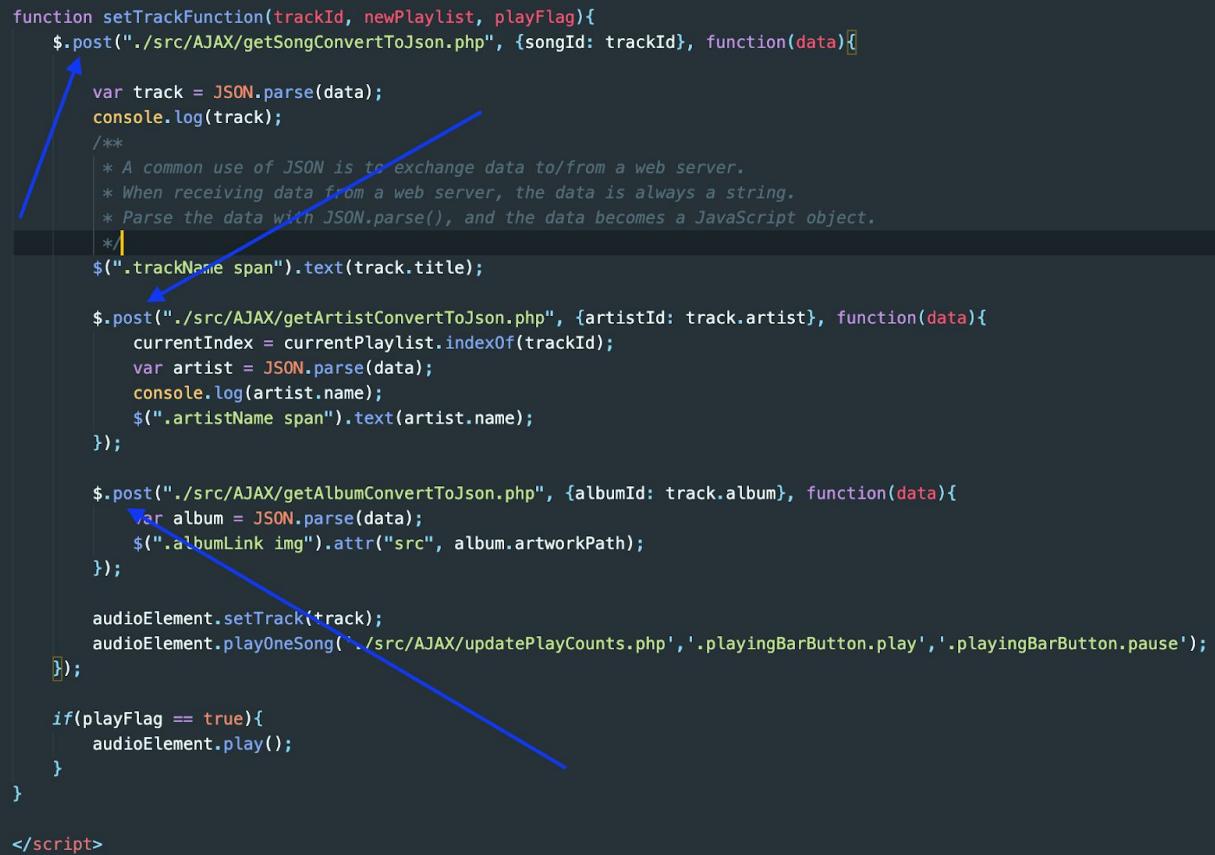
```

1 <?php
2 include("../config.php");
3
4 if(isset($_POST['albumId'])){
5     $albumId = $_POST['albumId'];
6
7     $query = mysqli_query($con, "SELECT * FROM albums WHERE id ='$albumId'");
8
9     $resultArray = mysqli_fetch_array($query);
//Fetch a result row as a numeric array and as an associative array:
10
11     echo json_encode($resultArray);
//The json_encode() function is an inbuilt function in PHP which is used to
12     //convert PHP array or object into JSON representation.
13 }
14 ?>

```

If you look at our codes, you will see every time we use albumId, artistId, and SongId to be the WHERE condition, that means we pick this album's song and artist.

Now we go back to playingBarComponent.php



```

function setTrackFunction(trackId, newPlaylist, playFlag){
    $.post("./src/AJAX/getSongConvertToJson.php", {songId: trackId}, function(data){
        var track = JSON.parse(data);
        console.log(track);
        /**
         * A common use of JSON is to exchange data to/from a web server.
         * When receiving data from a web server, the data is always a string.
         * Parse the data with JSON.parse(), and the data becomes a JavaScript object.
         */
        $(".trackName span").text(track.title);

        $.post("./src/AJAX/getArtistConvertToJson.php", {artistId: track.artist}, function(data){
            currentIndex = currentPlaylist.indexOf(trackId);
            var artist = JSON.parse(data);
            console.log(artist.name);
            $(".artistName span").text(artist.name);
        });

        $.post("./src/AJAX/getAlbumConvertToJson.php", {albumId: track.album}, function(data){
            var album = JSON.parse(data);
            $(".albumLink img").attr("src", album.artworkPath);
        });

        audioElement.setTrack(track);
        audioElement.playOneSong("./src/AJAX/updatePlayCounts.php','playingBarButton.play','playingBarButton.pause');
    });

    if(playFlag == true){
        audioElement.play();
    }
}

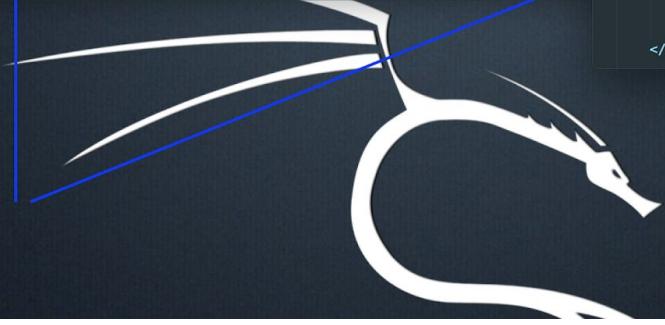
</script>

```

You can see this setTrackFunction() second time

We use http.post() function, this function will convert the data from my database to JSON data. This purpose is easy for JavaScript's function to get the data from AJAX. Anyway, whatever you get the data from database, the best way is to convert these data to JSON data. In the future, you will use this technique.

1. We output the album's pic and title on the left bottom



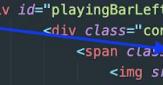
```

function setTrackFunction(trackId, newPlaylist, playFlag){
  $.post("./src/AJAX/getSongConvertToJson.php", {songId: trackId}, function(data){
    var track = JSON.parse(data);
    console.log(track);
    //...
    $(".trackName span").text(track.title);

    $.post("./src/AJAX/getArtistConvertToJson.php", {artistId: track.artist}, function(data){
      currentIndex = currentPlaylist.indexOf(trackId);
      var artist = JSON.parse(data);
      console.log(artist.name);
      $(".artistName span").text(artist.name);
    });
  });

  $.post("./src/AJAX/getAlbumConvertToJson.php", {albumId: track.album}, function(data){
    var album = JSON.parse(data);
    $(".albumLink img").attr("src", album.artworkPath);
  });
}

```



Team 20

Team 20

Now let us focus on the main functionality: playing music!

Exciting Moment!

Step1: look at the codes from albumPage.php

```

<div class="tracklistContainer">
    <ul class="tracklist">

        <?php
        $songIdArray = getSongIds($con, $albumId);
        $i = 1;
        foreach($songIdArray as $albumId) {

            $albumSong = new Song($con, $albumId);
            $albumArtist = getArtistName($con, $albumId);

            echo "<li class='tracklistRow'>
                    <div class='trackCount'>
                        <img class='play-btn' src='resource/images/icons/play-btn.png'
                            onclick='setTrackFunction(" . $albumSong->getId() . ", tempPlaylist, true)'>
                        <span class='trackNumber'>$i</span>
                    </div>
                    <div class='trackInfo'>
                        <div class='trackName'>" . $albumSong->getTitle() . "</div>
                        <div class='artistName'>" . $albumArtist . "</div>
                    </div>
                    <div class='trackOptions'>
                        <img class='optionsButton' src='resource/images/icons/more.png'>
                    </div>
                    <div class='trackDuration'>
                        <div class='duration'>" . $albumSong->getDuration() . "</div>
                    </div>
                </li>";
            $i = $i + 1;
        }
    ?>

    <script>
        var tempSongIds = '<?php echo json_encode($songIdArray); ?>';
        tempPlaylist = JSON.parse(tempSongIds);
        console.log()
    </script>

```

There is onclick function, when you click the playing button:



The image shows two code snippets side-by-side. On the left is a PHP file with code that includes a loop to get song IDs and artist names, and a script block at the bottom. On the right is a JavaScript file with a function named 'setTrackFunction'. Two blue arrows originate from the play/pause button in the mobile interface and point to the play/pause button in the PHP code and the start of the 'setTrackFunction' definition in the JavaScript code.

```

<div class="tracklistContainer">
    <ul class="tracklist">
        <?php
        $songIdArray = getSongIds($con, $albumId);
        $i = 1;
        foreach($songIdArray as $albumId) {
            $albumSong = new Song($con, $albumId);
            $albumArtist = getArtistName($con, $albumId);

            echo "<li class='trackListRow'>
                    <div class='trackCount'>
                        <img class='play-btn' src='resource/images/icons/play-btn.png'
                            onclick='setTrackFunction(\"$albumSong->getId()\", \"$tempPlaylist\", true)'>
                        <span class='trackNumber'>$i</span>
                    </div>
                    <div class='trackInfo'>
                        <div class='trackName'>" . $albumSong->getTitle() . "</div>
                        <div class='artistName'>" . $albumArtist . "</div>
                    </div>
                    <div class='trackOptions'>
                        <img class='optionsButton' src='resource/images/icons/more.png'>
                    </div>
                    <div class='trackDuration'>
                        <div class='duration'>" . $albumSong->getDuration() . "</div>
                    </div>
                </li>";
            $i = $i + 1;
        }
    ?>

    <script>
        var tempSongIds = '<?php echo json_encode($songIdArray); ?>';
        tempPlaylist = JSON.parse(tempSongIds);
        console.log();
    </script>

```

```

function setTrackFunction(trackId, newPlaylist, playFlag){
    $.post("./src/AJAX/getSongConvertToJson.php", {songId: trackId}, function(data){
        var track = JSON.parse(data);
        console.log(track);
        /*
        * A common use of JSON is to exchange data between a web server.
        * When receiving data from a web server, the data is always a string,
        * Parse the data with JSON.parse(), and the data becomes a JavaScript object.
        */
        $(".trackName span").text(track.title);

        $.post("./src/AJAX/getArtistConvertToJson.php", {artistId: track.artist}, function(data){
            currentIndex = currentPlaylist.indexOf(trackId);
            var artist = JSON.parse(data);
            console.log(artist.name);
            $(".artistName span").text(artist.name);
        });

        $.post("./src/AJAX/getAlbumConvertToJson.php", {albumId: track.album}, function(data){
            var album = JSON.parse(data);
            $(".albumLink img").attr("src", album.artworkPath);
        });

        audioElement.setTrack(track);
        audioElement.playOneSong("./src/AJAX/updatePlayCounts.php", ".playingBarButton.play", ".playingBarButton.pause");
    });

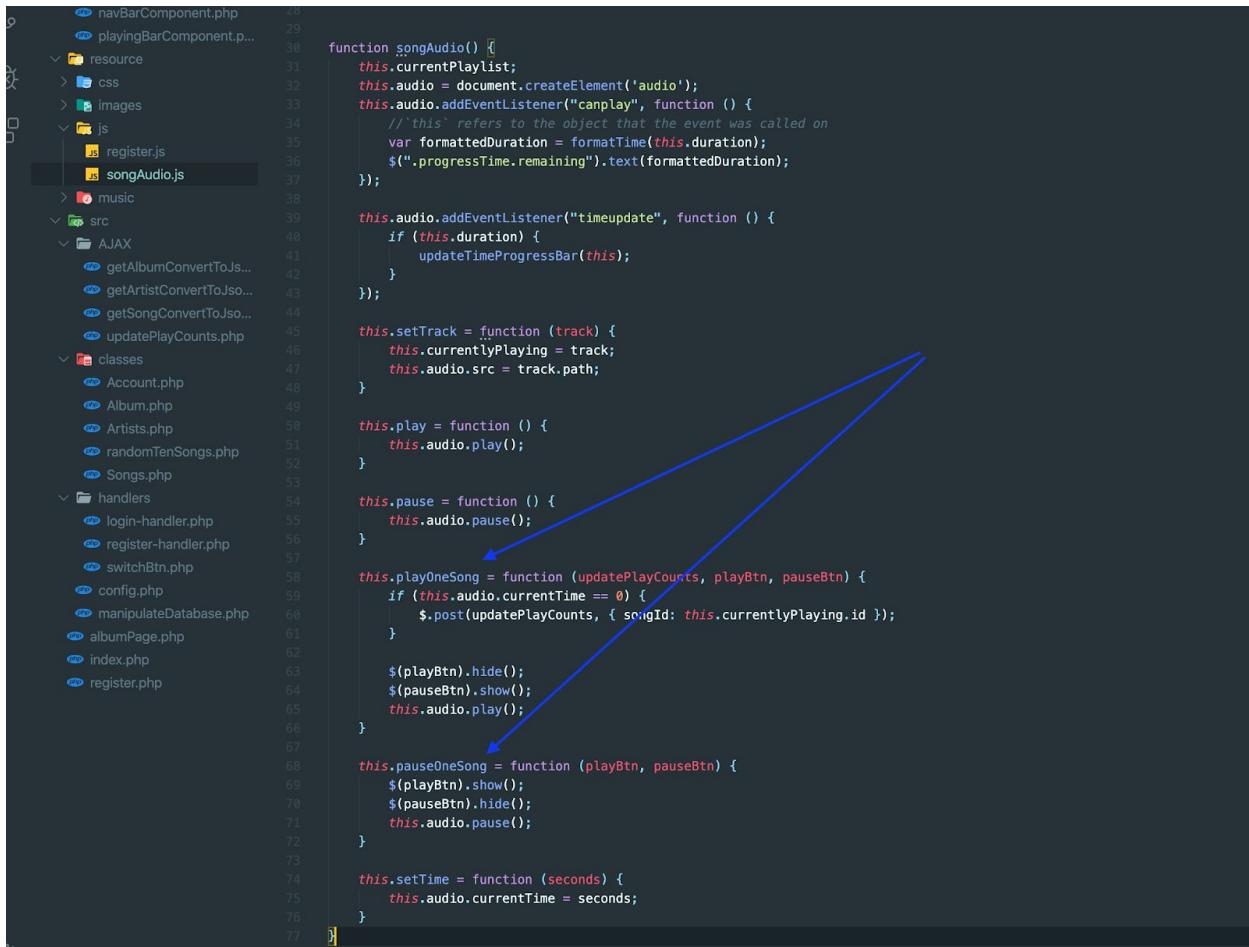
    if(playFlag == true){
        audioElement.play();
    }
}

```

Obviously, we have two parameters: 1 tempPlaylist, 2 \$albumSong

We assign the two parameters into our function setTrackFunction()

We create a new class songAudio.js to play the music in the folder named js



```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
    
```

```

function songAudio() {
  this.currentPlaylist;
  this.audio = document.createElement('audio');
  this.audio.addEventListener("canplay", function () {
    //`this` refers to the object that the event was called on
    var formattedDuration = formatTime(this.duration);
    $(".progressTime.remaining").text(formattedDuration);
  });

  this.audio.addEventListener("timeupdate", function () {
    if (this.duration) {
      updateTimeProgressBar(this);
    }
  });

  this.setTrack = function (track) {
    this.currentlyPlaying = track;
    this.audio.src = track.path;
  }

  this.play = function () {
    this.audio.play();
  }

  this.pause = function () {
    this.audio.pause();
  }

  this.playOneSong = function (updatePlayCounts, playBtn, pauseBtn) {
    if (this.audio.currentTime == 0) {
      $.post(updatePlayCounts, { songId: this.currentlyPlaying.id });
    }

    $(playBtn).hide();
    $(pauseBtn).show();
    this.audio.play();
  }

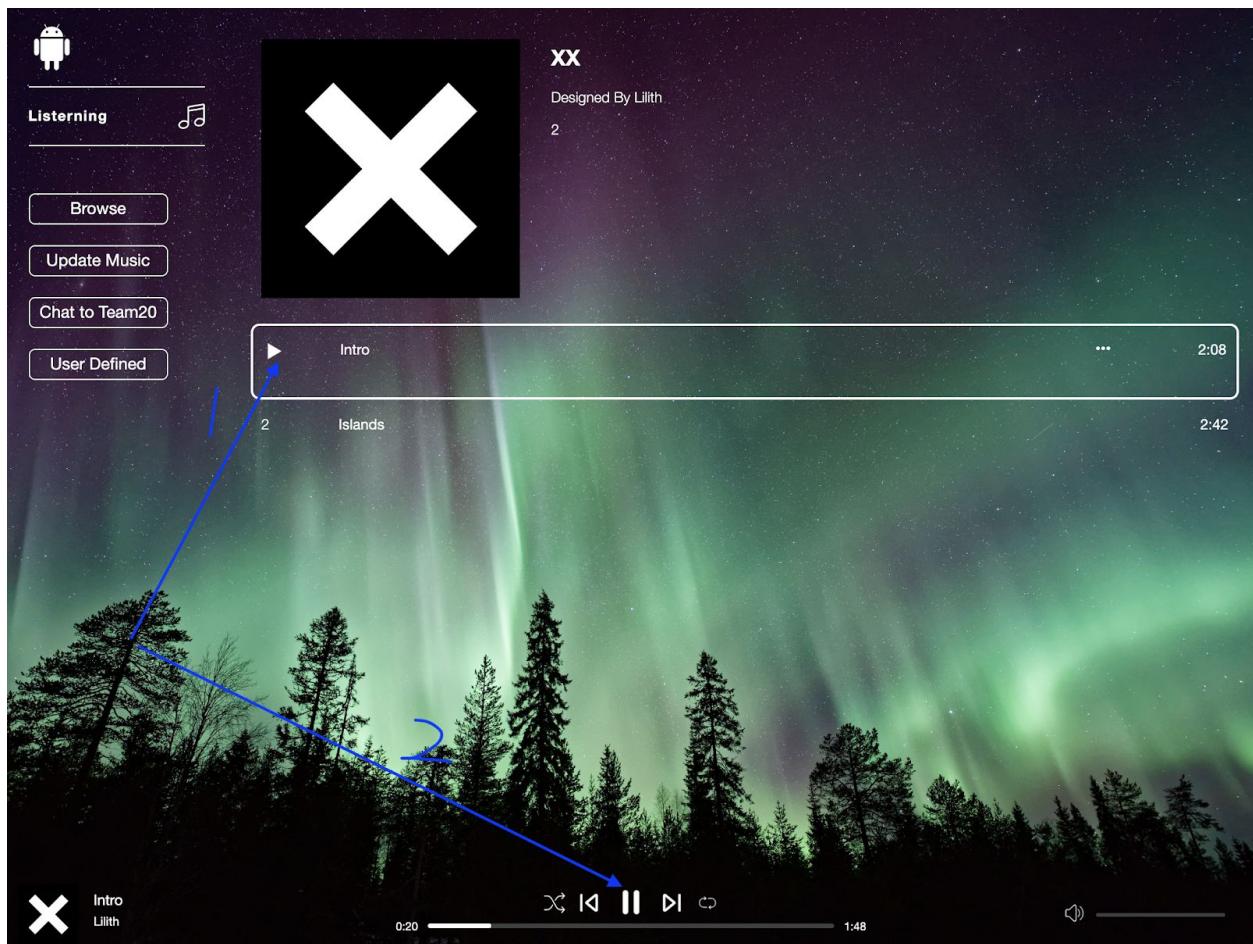
  this.pauseOneSong = function (playBtn, pauseBtn) {
    $(playBtn).show();
    $(pauseBtn).hide();
    this.audio.pause();
  }

  this.setTime = function (seconds) {
    this.audio.currentTime = seconds;
  }
}

```

In this class, we have two functions: playOneSong() and pauseOneSong()

So when we click the trackList's playing button, we will fire the event: playing a song



1. You click this playing button close to Intro
2. The playing button 2 will hide, and the pause button is displaying

```
this.playOneSong = function (updatePlayCounts, playBtn, pauseBtn) {
    if (this.audio.currentTime == 0) {
        $.post(updatePlayCounts, { songId: this.currentlyPlaying.id });
    }

    $(playBtn).hide();
    $(pauseBtn).show();
    this.audio.play();
}

this.pauseOneSong = function (playBtn, pauseBtn) {
    $(playBtn).show();
    $(pauseBtn).hide();
    this.audio.pause();
}
```

We can see if you click playing button, we bind a button to call the hide() function, if you click pause button we bind pauseBtn to call the show() function.

Let us focus the second main functionality: displaying duration

```
1 <div class="playbackBar">
    <span class="progressTime current">0.00</span>
    <div class="progressBar">
        <div class="progressBarBg">
            <div class="progress" id="progressID"></div>
        </div>
    </div>
    <span class="progressTime remaining">0.00</span>
</div>

</div>

2 <div id="playingBarRight">
    <div class="volumeBar">
        <button class="playingBarButton volume" title="Volume button">
            
        </button>
        <div class="progressBar">
            <div class="progressBarBg">
                <div class="progress"></div>
            </div>
        </div>
    </div>
</div>
```

This piece of codes is in playingBarComponent.php

Note: when the user is playing the music, it will update the duration time

```

1  JS songAudio.js ×
2
3  var currentPlaylist = [];
4
5  function formatTime(seconds) {
6      var time = Math.round(seconds);
7      var minutes = Math.floor(time / 60);
8      var seconds = time - minutes * 60;
9
10     var extraZero;
11     if (seconds < 10) {
12         extraZero = "0";
13     } else {
14         extraZero = "";
15     }
16     return minutes + ":" + extraZero + seconds;
17 }
18
19
20 function updateTimeProgressBar(audio) {
21     $(".progressTime.current").text(formatTime(audio.currentTime));
22     $(".progressTime.remaining").text(formatTime(audio.duration - audio.currentTime));
23
24     var progress = audio.currentTime / audio.duration * 100;
25     // $(".playbackBar .progress").css("width", progress + "%");
26     document.getElementById("progressID").style.width = progress + "%";
27 }
28
29
30 function songAudio() {
31     this.currentPlaylist;
32     this.audio = document.createElement('audio');
33     this.audio.addEventListener("canplay", function () {
34         // `this` refers to the object that the event was called on
35         var formattedDuration = formatTime(this.duration);
36         $(".progressTime.remaining").text(formattedDuration);
37     });
38
39     this.audio.addEventListener("timeupdate", function () {
40         if (this.duration) {
41             updateTimeProgressBar(this);
42         }
43     });
44 }

```

Step1: we have formatTime() function to convert the song's time format

Step2: we have a addEventListener() function to listen someone click this button, and then fire this event, call `updateTimeProgressBar(this)` function

Also we have drag the playingProgressBar event:

```

$(document).ready(function() {
    currentPlaylist = <?php echo $jsonArray; ?>;
    // console.log(currentPlaylist);
    audioElement = new songAudio();
    setTrackFunction(currentPlaylist[0], currentPlaylist, false);

    $(".playbackBar .progressBar").mousedown(function() {
        mouseDown = true;
    });

    $(".playbackBar .progressBar").mousemove(function(e) {
        if(mouseDown == true) {
            //Set time of song, depending on position of mouse
            timeFromOffset(e, this);
        }
    });

    $(".playbackBar .progressBar").mouseup(function(e) {
        timeFromOffset(e, this);
    });

    $(document).mouseup(function() {
        mouseDown = false;
    });

});

function timeFromOffset(mouse, progressBar) {
    var percentage = mouse.offsetX / $(progressBar).width() * 100;
    var seconds = audioElement.audio.duration * (percentage / 100);
    audioElement.setTime(seconds);
}

```

For here, 2 is the helper function to check the time off set.

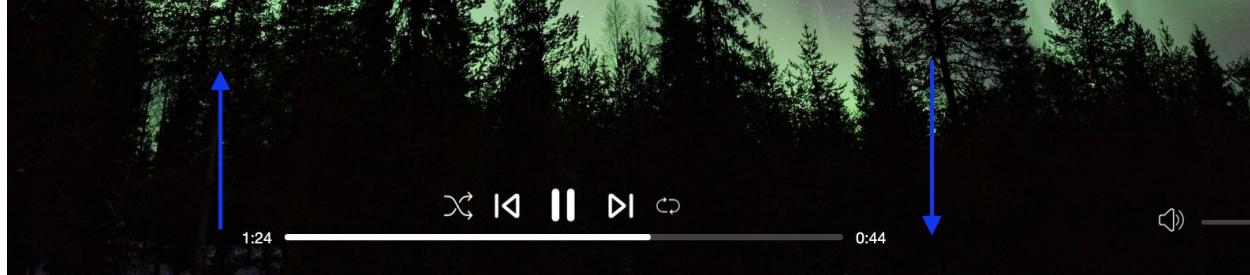
For duration:



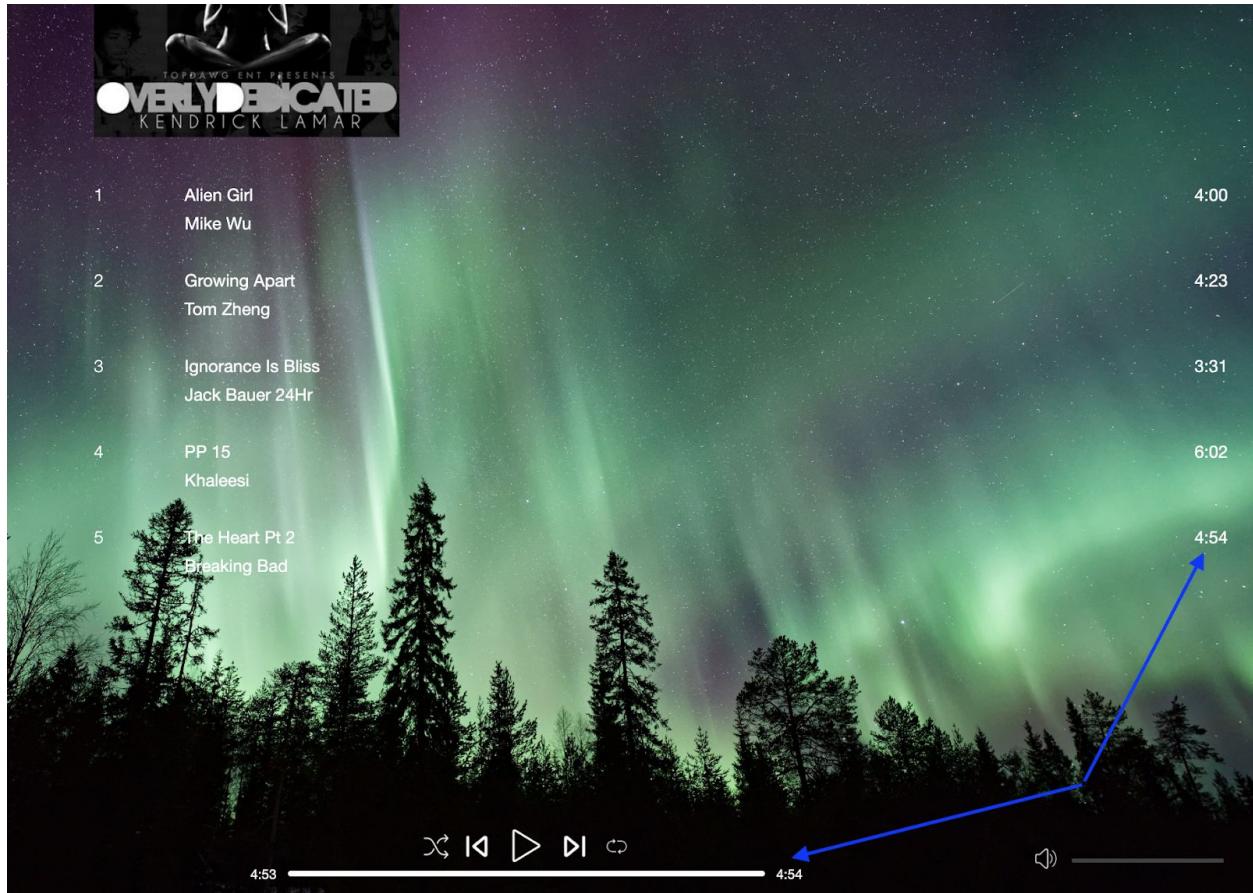
```

js songAudio.js × 📺 playingBarComponent.php
1 var currentPlaylist = [];
2 var audioElement;
3 var mouseDown = false;
4 var tempPlaylist = [];
5
6 function formatTime(seconds) {
7     var time = Math.round(seconds);
8     var minutes = Math.floor(time / 60);
9     var seconds = time - minutes * 60;
10
11     var extraZero;
12     if (seconds < 10) {
13         extraZero = "0";
14     } else {
15         extraZero = "";
16     }
17     return minutes + ":" + extraZero + seconds;
18 }
19
20 function updateTimeProgressBar(audio) {
21     $(".progressTime.current").text(formatTime(audio.currentTime));
22     $(".progressTime.remaining").text(formatTime(audio.duration - audio.currentTime));
23
24     var progress = audio.currentTime / audio.duration * 100;
25     // $(".playbackBar .progress").css("width", progress + "%");
26     document.getElementById("progressID").style.width = progress + "%";
27 }

```



- Look at this picture, when left hand side is increasing, the right hand side is decreasing.
- Because we have the function updateTimeProgressBar() function, when someone playing a song, the totally duration will minus the current time.



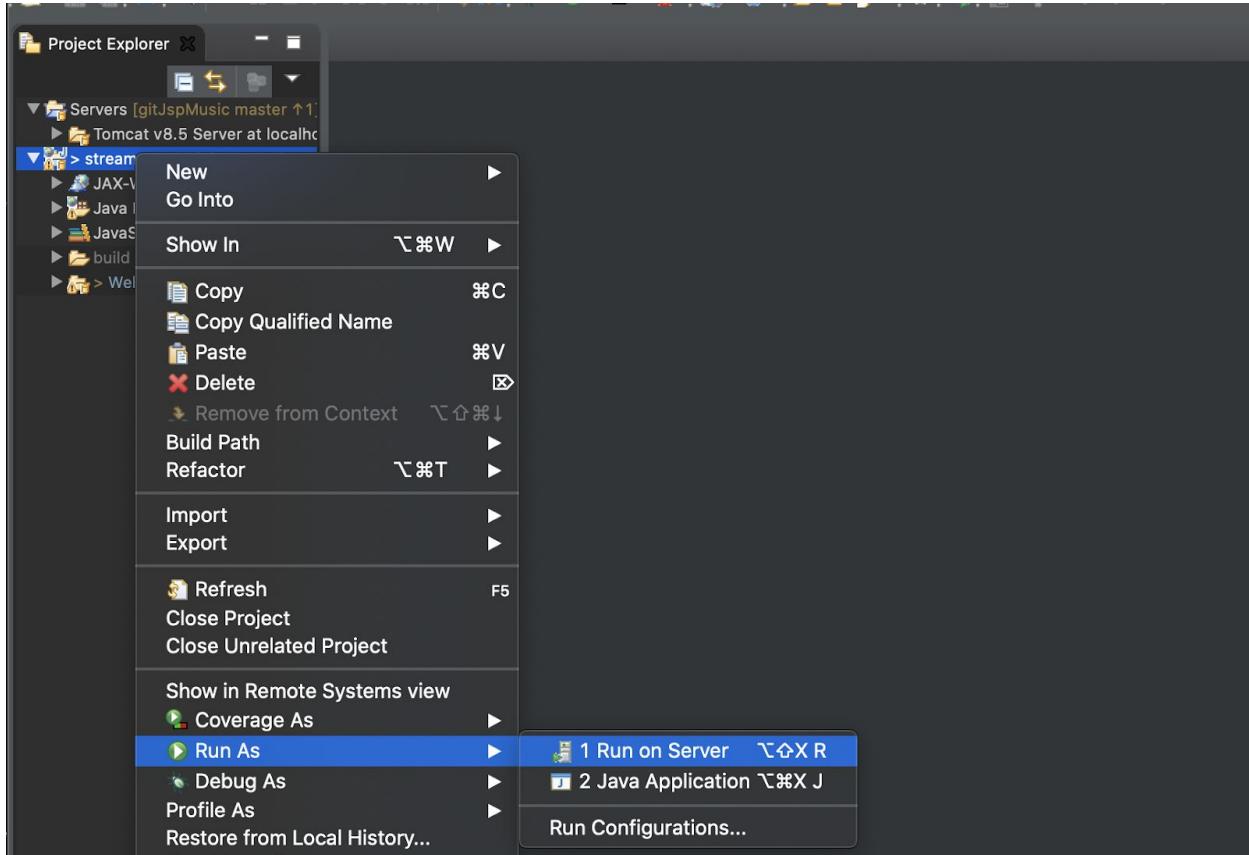
- You see the duration will keep the same, because we get the data from database and push onto our album page.
- But at the bottom, you see the album is 'Fucc It Up' and Artist is 'Jack', because after you click the album, it will pick any song by random

Project PartB: JDBC

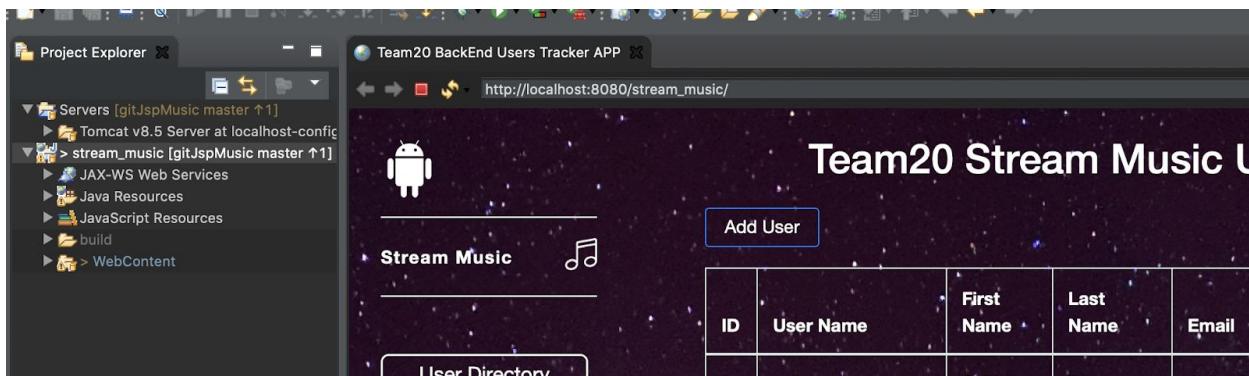
Stream Music BackEnd

For JDBC's installation, I have mentioned in the 'Installation' page.

How to run this project?



Run on Server



We copy this web address, then we open a new browser like google chrome.

Input: http://localhost:8080/stream_music/

| ID | User Name | First Name | Last Name | Email | photo | Action |
|----|-----------|------------|-----------|------------------|-------|---|
| 1 | selina | Selina | Ding | Selina@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 2 | tim | Tim | Cook | Tim@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 3 | mike | Mike | Wu | Mike@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 5 | jack | Jack | Jack | Jack@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 6 | rose | Rose | Jack | Rose@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 7 | tom | Tom | Zheng | Tom@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |
| 9 | newton | New | Ton | Newton@gmail.com | | Update Delete lambdaSort sqlSortMail heapSortID |

This JDBC project is only for our back-end engineers to manipulate the database.

There are five buttons.

1. User Directory: it will link to the homepage

http://localhost:8080/stream_music/ we can browse all user's information

2. Songs Directory: it will link to

http://localhost:8080/stream_music/songControllerServlet

We can browse all songs' information

3. Chat to Team20: you also can chat to team20, since we have talked on the previous pages, for here we don't talk.

4. User DataStructure

It also links to http://localhost:8080/stream_music/

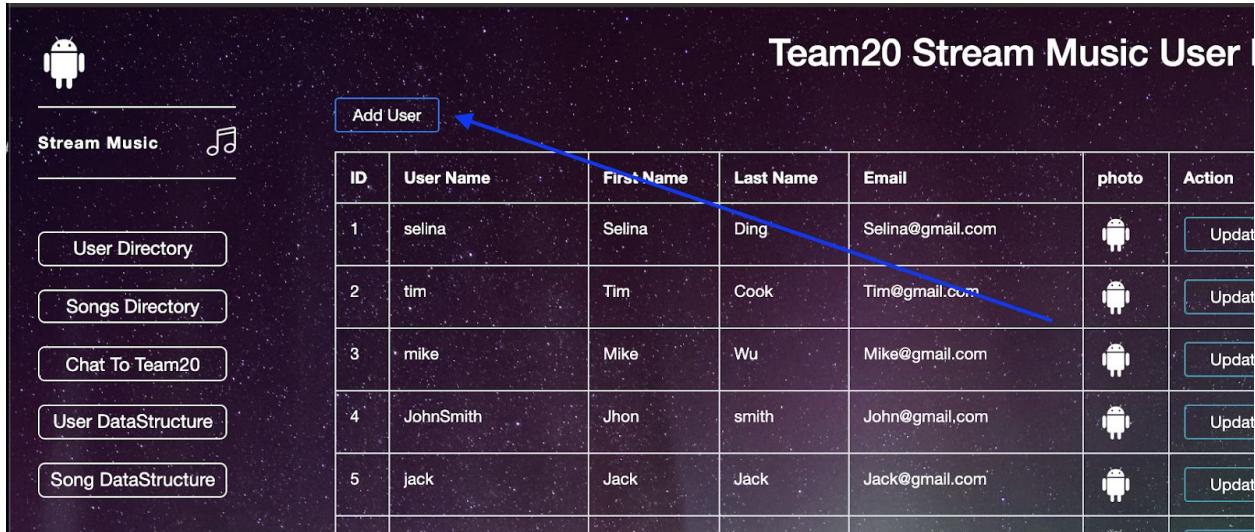
5. Song DataStructure

It also links to http://localhost:8080/stream_music/

- There are two events: 1. Users CRUD
2. Songs CRUD

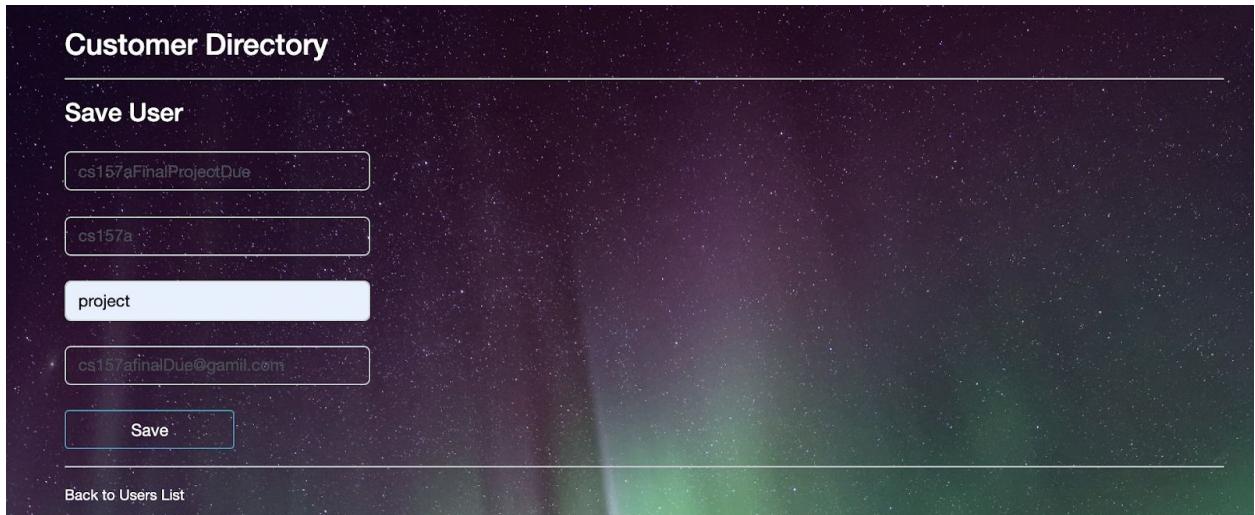
Event 1: Users CRUD

1. Click User Directory
2. Add a new User



The screenshot shows a mobile application interface titled "Team20 Stream Music User". On the left, there's a sidebar with icons for an Android phone, Stream Music, User Directory, Songs Directory, Chat To Team20, User DataStructure, and Song DataStructure. The main screen displays a table of users with columns: ID, User Name, First Name, Last Name, Email, photo, and Action. The table has 5 rows of data. An arrow points from the "Add User" button at the top right of the screen towards the "User Directory" button in the sidebar.

| ID | User Name | First Name | Last Name | Email | photo | Action |
|----|-----------|------------|-----------|------------------|-------|-------------------------|
| 1 | selina | Selina | Ding | Selina@gmail.com | | <button>Update</button> |
| 2 | tim | Tim | Cook | Tim@gmail.com | | <button>Update</button> |
| 3 | mike | Mike | Wu | Mike@gmail.com | | <button>Update</button> |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | | <button>Update</button> |
| 5 | jack | Jack | Jack | Jack@gmail.com | | <button>Update</button> |



The screenshot shows a "Customer Directory" page with a "Save User" form. The form fields include: "User Name" (cs157aFinalProjectDue), "Email" (cs157a), "Project" (project), and "Email" (cs157afinalDue@gmail.com). At the bottom is a "Save" button and a link "Back to Users List".

- Click Save
- Then we add a new user into our database

| | lambdaExpression | lambda | Expression | lambda@gmail.com | | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
|----|-----------------------|--------|------------|--------------------------|--|--------|--------|------------|-------------|------------|
| 22 | | | | | | | | | | |
| 34 | cs157aFinalProject | Final | Project | cs157afinal@gmail.com | | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 33 | lambdaSortByEmail | sort | byEmail | by@gmail.com | | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 35 | cs157aFinalProjectDue | cs157a | project | cs157afinalDue@gmail.com | | Update | Delete | lambdaSort | sqlSortMail | heapSortID |

| +-- | id username | firstName | lastName | email | password | signUpDate | profilePic |
|---|---------------|-----------|----------|---------|----------|------------|------------|
| +-- | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ | +-----+ |
| 1 selina Selina Ding Selina@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 2 tim Tim Cook Tim@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 3 mike Mike Wu Mike@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 4 JohnSmith Jhon smith John@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 5 jack Jack Jack Jack@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 6 rose Rose Jack Rose@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 7 tom Tom Zheng Tom@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 8 jerry Tom Jerry Jerry@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 9 clinton Tom New New@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 10 clinton Clinton Clinton Clinton@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 11 tesla Tesla Abc Tesla@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 12 mickey Mickey Mouse Mickey@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 13 lili Lily Absfs Lili@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 14 lucy Lucy Dsfadsf Lucy@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 15 lilei Lei Lilei@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 16 apple Ios Tim Apple@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-11-16 00:00:00 assets/images/profile/head.png | | | | | | | |
| 17 jobfair job fair job@gmail.com 5f4dc0c3b5aa765d1d8327debb882cf99 2017-06-28 00:00:00 resource/images/profile-pics/head.png | | | | | | | |
| 18 mackookPro15 macbook pro15 mac@gmail.com 5f4dc0c3b5aa765d1d8327debb882cf99 2019-11-28 02:47:27 resource/images/profile/head.png | | | | | | | |
| 21 M.Craft M. Craft m@gmail.com 5f4dc0c3b5aa765d1d8327debb882cf99 2019-11-28 03:07:18 resource/images/profile/head.png | | | | | | | |
| 22 lambdaExpression lambda Expression lambda@gmail.com 5f4dc0c3b5aa765d1d8327debb882cf99 2019-11-16 00:00:00 resource/images/profile/head.png | | | | | | | |
| 34 cs157aFinalProject Final Project Cs157afinal@gmail.com 8cb2237d0679ca88db6464eac69da96345513964 2019-12-08 00:00:00 resource/images/profile/head.png | | | | | | | |
| 33 lambdaSortByEmail sort byEmail by@gmail.com 5f4dc0c3b5aa765d1d8327debb882cf99 2019-11-30 23:03:02 resource/images/profile/head.png | | | | | | | |
| 35 cs157aFinalProjectDue cs157a project cs157afinaldue@gmail.com 5f4dc0c3b5aa765d1d8327debb882cf99 2019-12-11 05:15:49 resource/images/profile/head.png | | | | | | | |

23 rows in set (0.006 sec)

In our JDBC project folder, under the src folder, there is a `userControllerServlet.java`

Project

```

    userControllerSer...
    45 /**
    46 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
    47 */
    48 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
    49 {
    50     try {
    51         //read the "command" parameter
    52         String theCommand = request.getParameter("command");
    53
    54         // if the command is missing, then default to listing users
    55         if (theCommand == null) {
    56             theCommand = "LIST";
    57         }
    58
    59         // route to the appropriate method
    60         switch (theCommand) {
    61
    62             case "LIST":
    63                 listUsers(request, response);
    64                 break;
    65
    66             case "ADD": ←
    67                 addUsers(request, response);
    68                 break;
    69
    70             case "LOAD":
    71                 loadUsers(request, response);
    72                 break;
    73
    74             case "UPDATE":
    75                 updateUsers(request, response);
    76                 break;
    77
    78             case "DELETE":
    79                 delete(request, response);
    80                 break;
    81
    82             case "LAMBDA_EXPRESSION_SORT_USERNAME":
    83                 \LambdaExpressionSortByUsername(request, response);
    84                 break;
    85
    86             case "SQL_QUERY_SORT_EMAIL":
    87                 sqlQuerySortByEmail(request, response);
    88                 break;
    89
    90             case "HEAP_SORT_ID":
    91                 heapSortById(request, response);
    92                 break;
    93
    94             default:
    95                 listUsers(request, response);
    96
    
```

WebContent

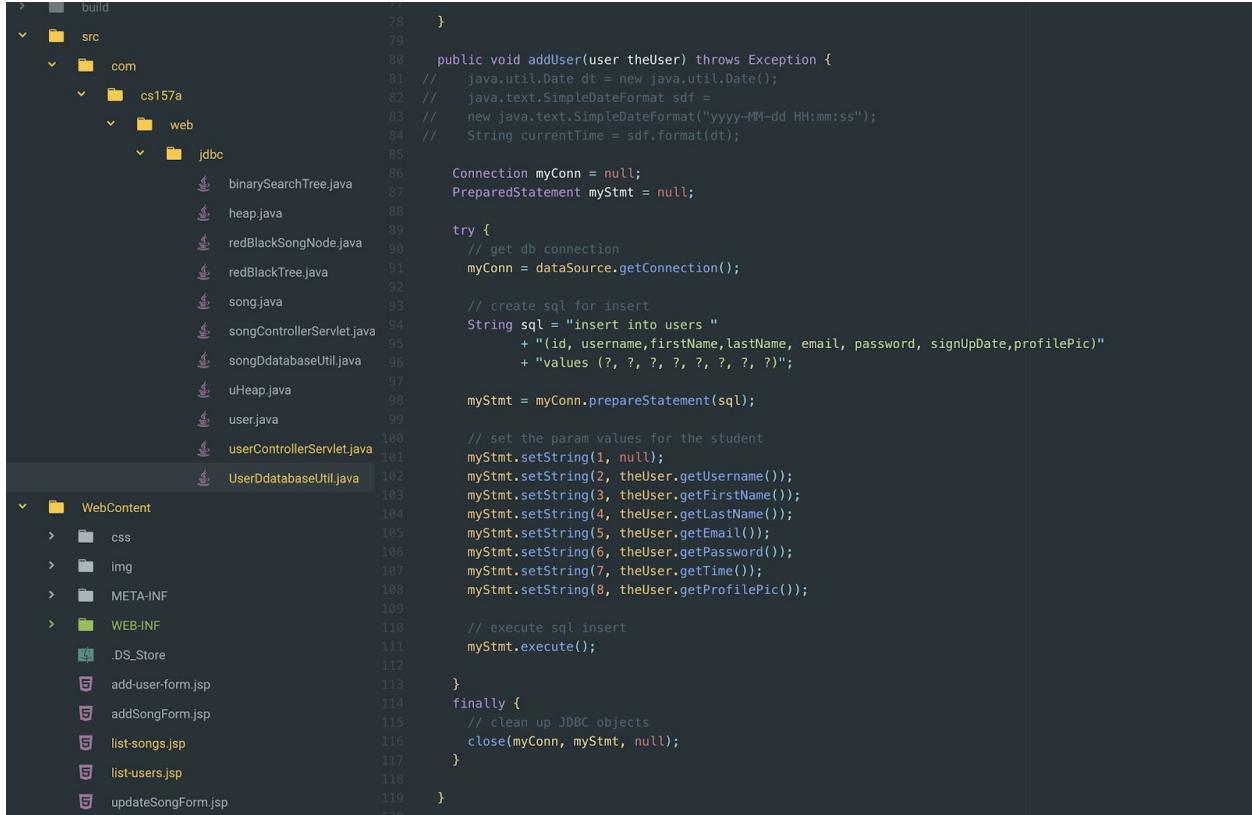
```

    binarySearchTree.java 204 dispatcher.forward(request, response);
    heap.java 205
    206 }
    207
    208 redBlackSongNode.java 208
    209 redBlackTree.java 209
    210 song.java 210
    211 songControllerServlet.java 212
    212 songDatabaseUtil.java 213
    213 uHeap.java 214
    214 user.java 215
    215 userControllerServlet.java 216
    216 UserDdatabaseUtil.java 217
    217
    218
    219
    220
    221
    222
    223
    224
    225
    226
    227
    
```

WebContent

```

    css
    img
    META-INF
    
```



The screenshot shows a Java code editor with a dark theme. The code is a Java class named `UserDatabaseUtil.java` located in the `src/com/cs157a/web/jdbc` package. The code implements a method `addUser` which performs an insert operation into a database table named `users`. The code uses JDBC to connect to a database, prepare a statement, set parameters, and execute the statement. It also handles closing the connection and statement objects.

```

    ...
    public void addUser(user theUser) throws Exception {
        // java.util.Date dt = new java.util.Date();
        // java.text.SimpleDateFormat sdf =
        //     new java.text.SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        // String currentTime = sdf.format(dt);

        Connection myConn = null;
        PreparedStatement myStmt = null;

        try {
            // get db connection
            myConn = dataSource.getConnection();

            // create sql for insert
            String sql = "insert into users "
                + "(id, username,firstName,lastName, email, password, signUpDate,profilePic)"
                + "values (?, ?, ?, ?, ?, ?, ?, ?)";

            myStmt = myConn.prepareStatement(sql);

            // set the param values for the student
            myStmt.setString(1, null);
            myStmt.setString(2, theUser.getUsername());
            myStmt.setString(3, theUser.getFirstName());
            myStmt.setString(4, theUser.getLastName());
            myStmt.setString(5, theUser.getEmail());
            myStmt.setString(6, theUser.getPassword());
            myStmt.setString(7, theUser.getTime());
            myStmt.setString(8, theUser.getProfilePic());

            // execute sql insert
            myStmt.execute();

        } finally {
            // clean up JDBC objects
            close(myConn, myStmt, null);
        }
    }
}

```

- This our java codes for how to add a user into our database
- I only introduce this major codes for main event ADD, so the following I will not introduce source codes

Delete a user:

A screenshot of a web browser displaying a user management interface for 'Stream Music'. The main page shows a table of users with columns: ID, User Name, First Name, Last Name, Email, and photo. A blue arrow points from the text 'Click yes, you will delete a user' to the 'Delete' button in a modal dialog box. The dialog box has a title 'localhost:8080 says' and the message 'Are you sure you want to delete this customer?'. It contains 'Cancel' and 'OK' buttons.

| ID | User Name | First Name | Last Name | Email | photo | Action |
|----|-----------|------------|-----------|------------------|-------|--|
| 1 | selina | Selina | Ding | Selina@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 2 | tim | Tim | Cook | Tim@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 3 | mike | Mike | Wu | Mike@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 5 | jack | Jack | Jack | Jack@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 6 | rose | Rose | Jack | Rose@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 7 | tom | Tom | Zheng | Tom@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 9 | newoton | New | Ton | Newton@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |

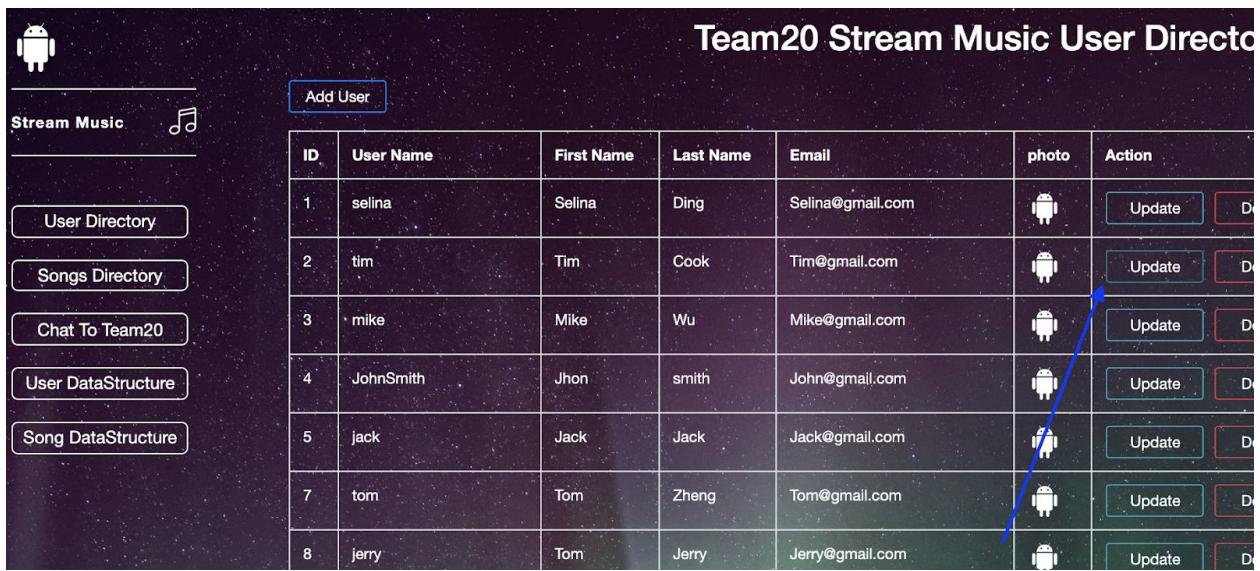
- If you click the delete button
 - It will give you hint: Are you sure you want to delete the customer?
- Click yes, you will delete a user

A screenshot of the 'Team20 Stream Music User Directory' page. The table now has 10 rows of data. A blue arrow points from the text 'Now id 6 is gone.' to the row where user ID 6 was located. The row is missing, indicating it has been deleted.

| ID | User Name | First Name | Last Name | Email | photo | Action |
|----|-----------|------------|-----------|-------------------|-------|--|
| 1 | selina | Selina | Ding | Selina@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 2 | tim | Tim | Cook | Tim@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 3 | mike | Mike | Wu | Mike@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 5 | jack | Jack | Jack | Jack@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 7 | tom | Tom | Zheng | Tom@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 9 | newoton | New | Ton | Newton@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |
| 10 | clinton | Clinton | Clinton | Clinton@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSort</button> |

- Now id 6 is gone.

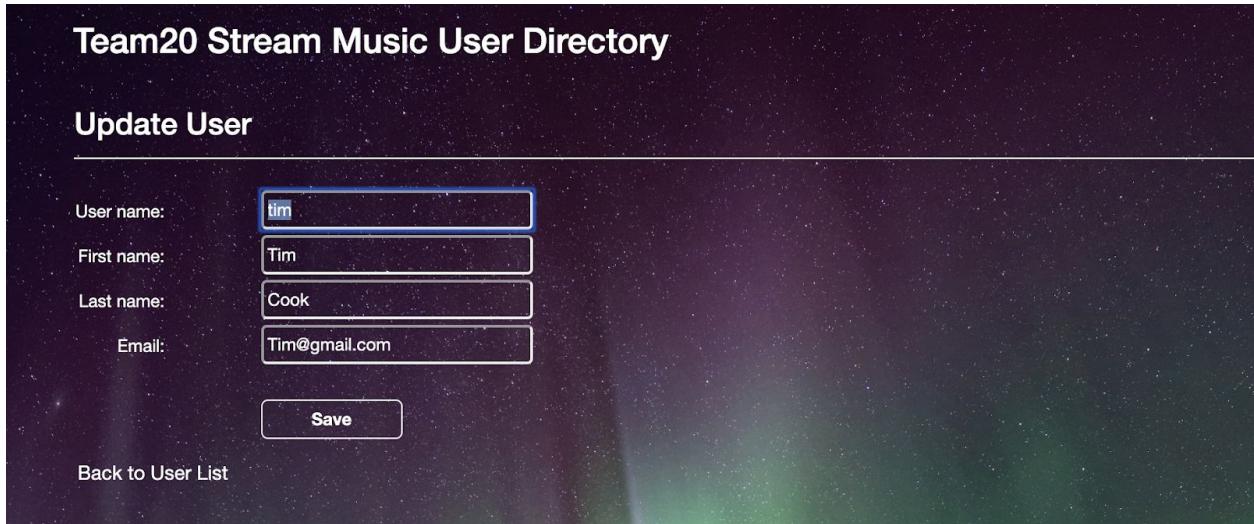
Update a user



The screenshot shows a mobile application interface titled "Team20 Stream Music User Directory". On the left, there's a sidebar with icons for "User Directory", "Songs Directory", "Chat To Team20", "User DataStructure", and "Song DataStructure". The main area displays a table of users:

| ID | User Name | First Name | Last Name | Email | photo | Action |
|----|-----------|------------|-----------|------------------|-------|---|
| 1 | selina | Selina | Ding | Selina@gmail.com | | <button>Update</button> <button>Delete</button> |
| 2 | tim | Tim | Cook | Tim@gmail.com | | <button>Update</button> <button>Delete</button> |
| 3 | mike | Mike | Wu | Mike@gmail.com | | <button>Update</button> <button>Delete</button> |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | | <button>Update</button> <button>Delete</button> |
| 5 | jack | Jack | Jack | Jack@gmail.com | | <button>Update</button> <button>Delete</button> |
| 7 | tom | Tom | Zheng | Tom@gmail.com | | <button>Update</button> <button>Delete</button> |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | | <button>Update</button> <button>Delete</button> |

- Click update button



The screenshot shows the "Update User" form. It has four input fields: "User name" (containing "tim"), "First name" (containing "Tim"), "Last name" (containing "Cook"), and "Email" (containing "Tim@gmail.com"). Below the fields is a "Save" button. At the bottom, there's a link "Back to User List".

Team20 Stream Music User Directory

Update User

User name:

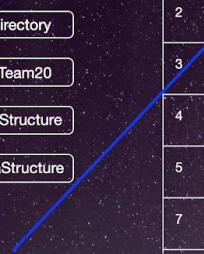
First name:

Last name:

Email:

[Back to User List](#)

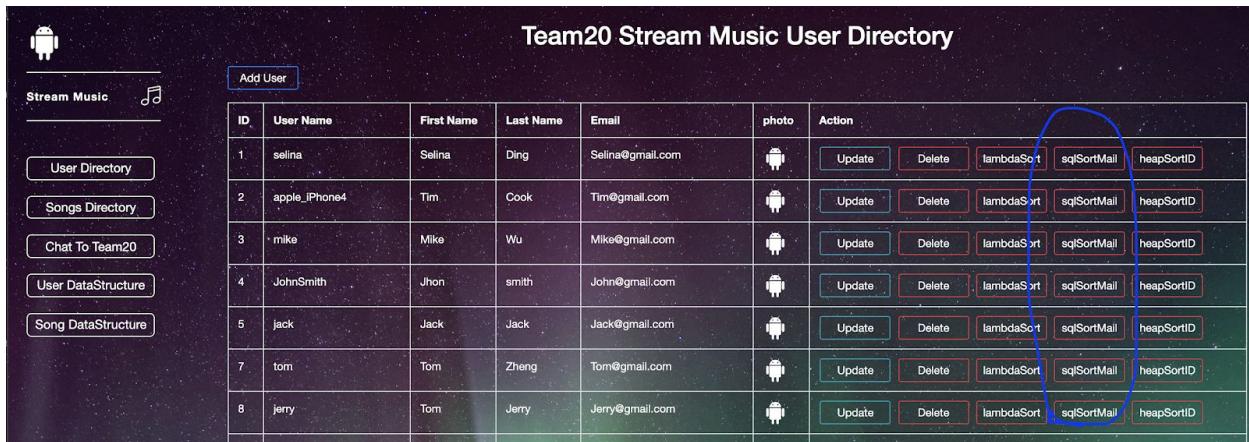
- Click save



| ID. | User Name | First Name | Last Name | Email | photo | Action |
|-----|---------------|------------|-----------|------------------|---|---|
| 1 | selina | Selina | Ding | Selina@gmail.com |  | <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="lambdaSort"/> |
| 2 | apple_iPhone4 | Tim | Cook | Tim@gmail.com |  | <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="lambdaSort"/> |
| 3 | mike | Mike | Wu | Mike@gmail.com |  | <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="lambdaSort"/> |
| 4 | JohnSmith | Jhon | smith | John@gmail.com |  | <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="lambdaSort"/> |
| 5 | jack | Jack | Jack | Jack@gmail.com |  | <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="lambdaSort"/> |
| 7 | tom | Tom | Zheng | Tom@gmail.com |  | <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="lambdaSort"/> |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com |  | <input type="button" value="Update"/> <input type="button" value="Delete"/> <input type="button" value="lambdaSort"/> |

We changed this user's username

Sql Query Sort the users by their email string



| ID | User Name | First Name | Last Name | Email | photo | Action |
|----|---------------|------------|-----------|------------------|-------|--|
| 1 | selina | Selina | Ding | Selina@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 2 | apple_iPhone4 | Tim | Cook | Tim@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 3 | mike | Mike | Wu | Mike@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 5 | jack | Jack | Jack | Jack@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 7 | tom | Tom | Zheng | Tom@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |

- Click this button



| ID | User Name | First Name | Last Name | Email | photo | Action |
|----|-----------------------|------------|------------|--------------------------|-------|--|
| 16 | apple | Ios | Tim | Apple@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 33 | lambdaSortByEmail | sort | byEmail | by@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 10 | clinton | Clinton | Clinton | Clinton@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 34 | cs157aFinalProject | Final | Project | Cs157afinal@gamil.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 35 | cs157aFinalProjectDue | cs157a | project | cs157afinalDue@gamil.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 5 | jack | Jack | Jack | Jack@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 17 | jobfair | job | fair | job@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 4 | JohnSmith | Jhon | smith | John@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 22 | lambdaExpression | lambda | Expression | lambda@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 15 | lilei | Li | Lei | Lilei@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 13 | lili | Lily | Absfs | Lili@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 14 | lucy | Lucy | Dsfadsaf | Lucy@gmail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |
| 21 | M.Craft | M. | Craft | m@mail.com | | <button>Update</button> <button>Delete</button> <button>lambdaSort</button> <button>sqlSortMail</button> <button>heapSortID</button> |

- The order has changed



```

    ...
    public ArrayList<User> getSqlQuerySortByEmailUsersList() throws SQLException{
        ArrayList<User> users = new ArrayList<>();
        Connection conn = null;
        Statement stmt = null;
        ResultSet rs = null;
        try {
            //get a connection
            conn = dataSource.getConnection();
            //create sql statement
            String sql = "SELECT * FROM users ORDER BY email";
            stmt = conn.createStatement();
            //execute query
            rs = stmt.executeQuery(sql);
            ...
            while(rs.next()) {
                //retrieve data from result set row
                int id = rs.getInt("id");
                String username = rs.getString("username");
                String firstName = rs.getString("firstName");
                String lastName = rs.getString("lastName");
                String email = rs.getString("email");
                ...
                //create new user object
                User tempUser = new User(id, username, firstName, lastName, email);
                //add it to the list of users
                users.add(tempUser);
            }
            ...
            Collections.sort(users, User.comparatorByEmail(true));
        } finally {
            ...
            //close JDBC objects
            close(conn, stmt, rs);
        }
    }
}

```

- We call sql query order by email

Java Object-Oriented using lambda expression to sort all users

| Team20 Stream Music User Directory | | | | | | | |
|------------------------------------|-----------------------|------------|-----------|--------------------------|---|--------------------------|------------------------|
| ID | User Name | First Name | Last Name | Email | photo | Action | |
| 16 | apple | Ios | Tim | Apple@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 33 | lambdaSortByEmail | sort | byEmail | by@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 10 | clinton | Clinton | Clinton | Clinton@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 34 | cs157aFinalProject | Final | Project | Cs157afinal@gamil.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 35 | cs157aFinalProjectDue | cs157a | project | cs157afinalDue@gamil.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 5 | jack | Jack | Jack | Jack@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 8 | jerry | Tom | Jerry | Jerry@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 17 | jobfair | job | fair | job@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 4 | JohnSmith | Jhon | smith | John@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |

| Team20 Stream Music User Directory | | | | | | | |
|------------------------------------|-----------------------|------------|-----------|--------------------------|---|--------------------------|------------------------|
| ID | User Name | First Name | Last Name | Email | photo | Action | |
| 4 | JohnSmith | Jhon | smith | John@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 21 | M.Craft | M. | Craft | m@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 16 | apple | Ios | Tim | Apple@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 2 | apple_iPhone4 | Tim | Cook | Tim@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 10 | clinton | Clinton | Clinton | Clinton@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 34 | cs157aFinalProject | Final | Project | Cs157afinal@gamil.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 35 | cs157aFinalProjectDue | cs157a | project | cs157afinalDue@gamil.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |
| 5 | jack | Jack | Jack | Jack@gmail.com |  | Update Delete lambdaSort | sqlSortMail heapSortID |

- Order has changed by username's string comparing

```

songDatabaseUtil.java 112
uHeap.java 113
user.java 114
userControllerServlet.java 115
UserDatabaseUtil.java 116
117
118 @Override
119 public String toString() {
120     return "User [id=" + id + ", username=" + username + ", firstName=" + firstName + ", lastName=" + lastName
121     + ", email=" + email + "]";
122 }
123
124 /**
125 * Interface
126 * @param ascending sorting order
127 * @return an Interface
128 */
129 public static Comparator<User> comparatorByUserName(boolean ascending) {
130     return (first, second) -> (ascending ? 1 : -1) * first.getUsername().compareTo(second.getUsername());
131 }
132
133 /**
134 * Interface
135 * @param ascending sorting order
136 * @return an Interface
137 */
138 public static Comparator<User> comparatorByEmail(boolean ascending) {
139     return (first, second) -> (ascending ? 1 : -1) * first.getEmail().compareTo(second.getEmail());
140 }
141
142 }
143

```

```

.settings 229
build 230 }
src 231
src/com 232
src/com/cs157a 233
src/com/cs157a/web 234
src/com/cs157a/web/jdbc 235
src/com/cs157a/web/jdbc/binarySearchTree.java 236
src/com/cs157a/web/jdbc/heap.java 237
src/com/cs157a/web/jdbc/redBlackSongNode.java 238
src/com/cs157a/web/jdbc/redBlackTree.java 239
src/com/cs157a/web/jdbc/song.java 240
src/com/cs157a/web/jdbc/songControllerServlet.java 241
src/com/cs157a/web/jdbc/songDatabaseUtil.java 242
src/com/cs157a/web/jdbc/uHeap.java 243
src/com/cs157a/web/jdbc/user.java 244
src/com/cs157a/web/jdbc/userControllerServlet.java 245
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 246
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 247
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 248
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 249
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 250
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 251
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 252
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 253
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 254
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 255
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 256
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 257
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 258
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 259
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 260
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 261
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 262
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 263
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 264
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 265
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 266
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 267
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 268
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 269
src/com/cs157a/web/jdbc/UserDatabaseUtil.java 270
src/META-INF 229
src/WEB-INF 230
src/WEB-INF/.DS_Store 231
src/WEB-INF/add-user-form.jsp 232
src/WEB-INF/addSongForm.jsp 233
src/WEB-INF/list-songs.jsp 234

```

DataStructure:

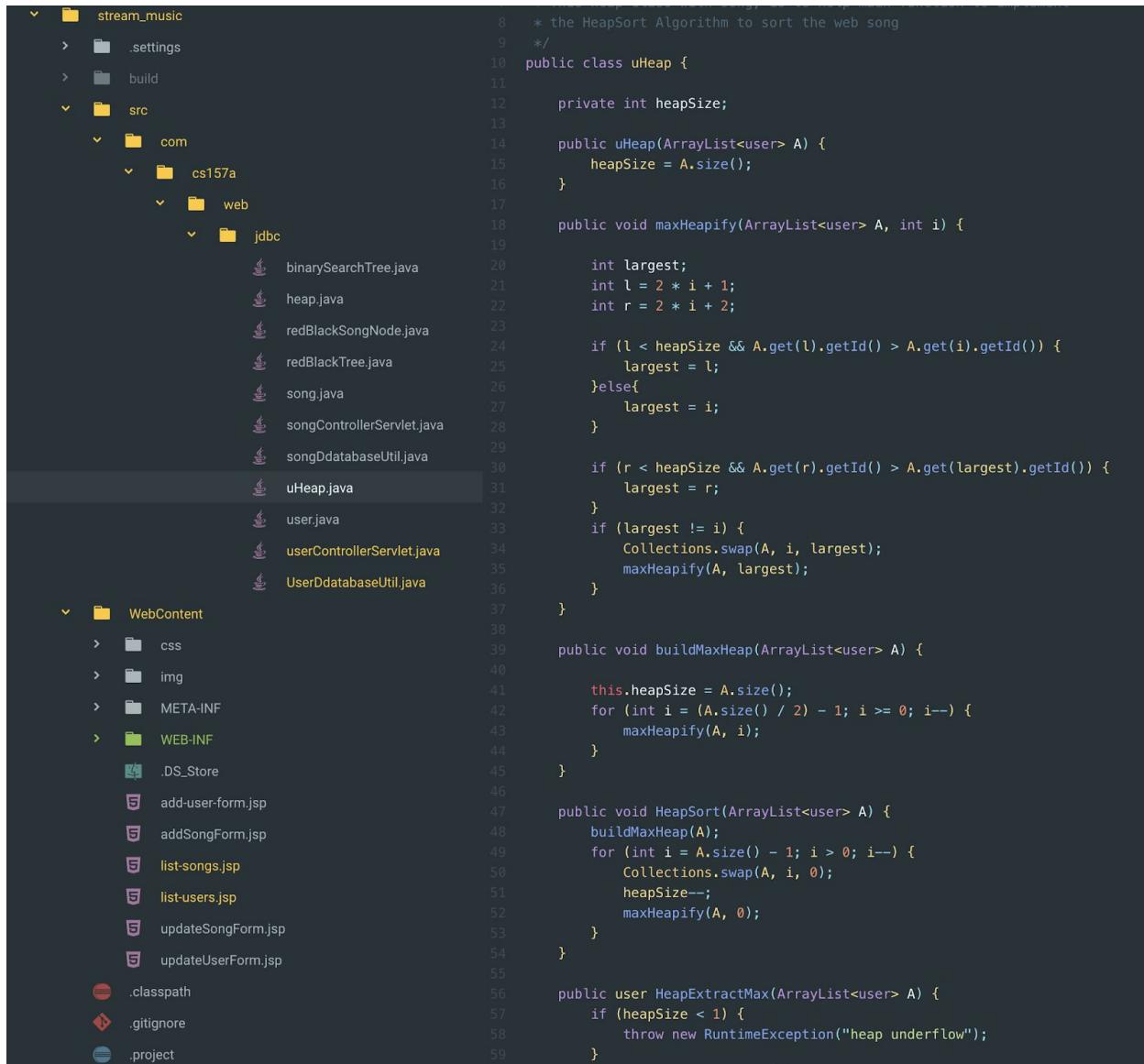
1. heapSort
2. Binary Search Tree
3. Red Black Tree

4. All DataStructure source codes I convert the pseudocode to source codes from CS146 textbook.

HeapSort:

team20 Stream MUSIC User Directory

| team20 Stream MUSIC User Directory | | | | | | | | | | |
|------------------------------------|-----------------------|------------|-----------|--------------------------|---|------------------------|------------------------|----------------------------|-----------------------------|----------------------------|
| Add User | | | | | | | | | | |
| ID | User Name | First Name | Last Name | Email | photo | Action | | | | |
| 4 | JohnSmith | Jhon | smith | John@gmail.com |  | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 21 | M.Craft | M. | Craft | m@gmail.com |  | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 16 | apple | Ios | Tim | Apple@gmail.com |  | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 2 | apple_iPhone4 | Tim | Cook | Tim@gmail.com |  | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 10 | clinton | Clinton | Clinton | Clinton@gmail.com |  | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 34 | cs157aFinalProject | Final | Project | Cs157afinal@gmail.com |  | Update | Delete | lambdaSort | sqlSortMail | heapSortID |
| 35 | cs157aFinalProjectDue | cs157a | project | cs157afinalDue@gmail.com |  | Update | Delete | lambdaSort | sqlSortMail | heapSortID |



The screenshot shows a Java project structure in an IDE. The project root is named 'stream_music'. It contains a '.settings' folder, a 'build' folder, and a 'src' folder. The 'src' folder is further divided into 'com' and 'cs157a', which contains 'web' and 'jdbc' packages. The 'jdbc' package contains several Java files: binarySearchTree.java, heap.java, redBlackSongNode.java, redBlackTree.java, song.java, songControllerServlet.java, songDatabaseUtil.java, uHeap.java, user.java, userControllerServlet.java, and UserDatabaseUtil.java. Below the 'src' folder is a 'WebContent' folder containing 'css', 'img', 'META-INF', 'WEB-INF', and a '.DS_Store' file. Under 'WEB-INF', there are JSP files: add-user-form.jsp, addSongForm.jsp, list-songs.jsp, list-users.jsp, updateSongForm.jsp, and updateUserForm.jsp. At the bottom of the project view are configuration files: '.classpath', '.gitignore', and '.project'.

```

8 * the HeapSort Algorithm to sort the web song
9 */
10 public class uHeap {
11     private int heapSize;
12
13     public uHeap(ArrayList<user> A) {
14         heapSize = A.size();
15     }
16
17     public void maxHeapify(ArrayList<user> A, int i) {
18
19         int largest;
20         int l = 2 * i + 1;
21         int r = 2 * i + 2;
22
23         if (l < heapSize && A.get(l).getId() > A.get(i).getId()) {
24             largest = l;
25         } else {
26             largest = i;
27         }
28
29         if (r < heapSize && A.get(r).getId() > A.get(largest).getId()) {
30             largest = r;
31         }
32         if (largest != i) {
33             Collections.swap(A, i, largest);
34             maxHeapify(A, largest);
35         }
36     }
37
38
39     public void buildMaxHeap(ArrayList<user> A) {
40
41         this.heapSize = A.size();
42         for (int i = (A.size() / 2) - 1; i >= 0; i--) {
43             maxHeapify(A, i);
44         }
45     }
46
47     public void HeapSort(ArrayList<user> A) {
48         buildMaxHeap(A);
49         for (int i = A.size() - 1; i > 0; i--) {
50             Collections.swap(A, i, 0);
51             heapSize--;
52             maxHeapify(A, 0);
53         }
54     }
55
56     public user HeapExtractMax(ArrayList<user> A) {
57         if (heapSize < 1) {
58             throw new RuntimeException("heap underflow");
59         }

```

Event 2 : Songs CRUD

- For here we don't introduce its Update, Delete, Add
- Because it is similar to User CRUD
- We only talk about Binary Search Tree and Red Black Tree

Click Songs Directory

Team20 Stream Music Songs Directory

| ID | Title | Artist | album | genre | duration | MP3 Files | abOrder | Plays | Action' |
|----|-----------------------------|--------|-------|-------|----------|-----------|---------|-------|---|
| 1 | 2Face | 1 | 1 | 1 | 4:46 | 🎵 | 1 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 2 | Back To The Topic Freestyle | 1 | 1 | 1 | 3:00 | 🎵 | 2 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 3 | Before Im Gone | 1 | 1 | 1 | 4:24 | 🎵 | 3 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 4 | Best Friend | 1 | 1 | 1 | 3:25 | 🎵 | 4 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 5 | Alien Girl | 2 | 2 | 1 | 4:00 | 🎵 | 1 | 10 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 6 | Growing Apart | 2 | 2 | 1 | 4:23 | 🎵 | 2 | 4 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 7 | Ignorance Is Bliss | 2 | 2 | 1 | 3:31 | 🎵 | 3 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 8 | PP 15 | 2 | 2 | 1 | 6:02 | 🎵 | 4 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 9 | The Heart Pt 2 | 2 | 2 | 1 | 4:54 | 🎵 | 5 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |

- Click bstInorder, that means it will sort by BST Inorder Walk by genre

Team20 Stream Music Songs Directory

| ID | Title | Artist | album | genre | duration | MP3 Files | abOrder | Plays | Action' |
|----|-----------------------------|--------|-------|-------|----------|-----------|---------|-------|---|
| 1 | 2Face | 1 | 1 | 1 | 4:46 | 🎵 | 1 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 2 | Back To The Topic Freestyle | 1 | 1 | 1 | 3:00 | 🎵 | 2 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 3 | Before Im Gone | 1 | 1 | 1 | 4:24 | 🎵 | 3 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 4 | Best Friend | 1 | 1 | 1 | 3:25 | 🎵 | 4 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 5 | Alien Girl | 2 | 2 | 1 | 4:00 | 🎵 | 1 | 10 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 6 | Growing Apart | 2 | 2 | 1 | 4:23 | 🎵 | 2 | 4 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |
| 7 | Ignorance Is Bliss | 2 | 2 | 1 | 3:31 | 🎵 | 3 | 2 | <button>Update</button> <button>Delete</button> <button>HeapSort</button> bstInorder <button>rbtInorder</button> |

Project

```

    stream_music
      .settings
      build
      src
        com
          cs157a
            web
              jdbc
                binarySearchTree.java
                heap.java
                redBlackSongNode.java
                redBlackTree.java
                song.java
                songControllerServlet.java
                songDatabaseUtil.java
                uHeap.java
                user.java
                userControllerServlet.java
                UserDatabaseUtil.java
      WebContent
        css
        img
        META-INF
        WEB-INF
          DS_Store
          add-user-form.jsp
          addSongForm.jsp
          test-project.jsp

```

binarySearchTree.java

```

1 package com.cs157a.web.jdbc;
2
3 import java.util.ArrayList;
4 import java.util.LinkedHashSet;
5 import java.util.Set;
6
7 /**
8 * A class that we can use this binary search tree both as a dictionary and as a priority queue.
9 */
10 public class binarySearchTree {
11
12     /* Create a Root song for this binary search Tree */
13     private song root;
14     protected ArrayList<song> songlist;
15
16     /* default constructor */
17     public binarySearchTree(){
18         songlist = new ArrayList<>();
19     }
20
21     public void setRoot(song root){
22         this.root = root;
23     }
24
25     public song getRoot(){
26         return root;
27     }
28
29     //Insert a name to the Retirement BST based on his/her SSN.
30     public void Tree_Insert(song z){
31         song y = null;
32         song x = this.root;
33         while(x != null){
34             y = x;
35             if(z.getGenre() < x.getGenre()){
36                 x = x.getLeft();
37             }else{
38                 x = x.getRight();
39             }
40         }
41         z.setParent(y);
42         if(y == null){
43             this.setRoot(z);

```

binarySearchTree.java — ~/gits/DSP/music/streamMusicBackend

```

    stream_music
      .settings
      build
      src
        com
          cs157a
            web
              jdbc
                binarySearchTree.java
                heap.java
                redBlackSongNode.java
                redBlackTree.java
                song.java
                songControllerServlet.java
                songDatabaseUtil.java
                uHeap.java
                user.java
                userControllerServlet.java
                UserDatabaseUtil.java
      WebContent
        css
        img
        META-INF
        WEB-INF
          DS_Store
          add-user-form.jsp
          addSongForm.jsp
          list-songs.jsp
          list-users.jsp
          updateSongForm.jsp
          updateUserForm.jsp
          .classpath
          .gitignore
          project

```

binarySearchTree.java

```

66     /*****BST Minimum*****
67     public song BST_Minimum(song x){
68         while(x.getLeft() != null){
69             x = x.getLeft();
70         }
71         return x;
72     }
73
74     public song BST_Maximum(song x){
75         while(x.getRight() != null){
76             x = x.getRight();
77         }
78         return x;
79     }
80
81     public void BST_Delete(song z){
82         if(z.getLeft() == null){
83             Transplant(z, z.getRight());
84         }else if(z.getRight() == null){
85             Transplant(z, z.getLeft());
86         }else{
87             song y = BST_Minimum(z.getRight());
88             if(y.getParent() != z){
89                 Transplant(y, y.getRight());
90                 y.setRight(z.getRight());
91                 y.getRight().setParent(y);
92             }
93             Transplant(z, y);
94             y.setLeft(z.getLeft());
95             y.getLeft().setParent(y);
96         }
97     }
98
99     public void Inorder_BST_Walk(song x){
100         if(x != null){
101             Inorder_BST_Walk(x.getLeft());
102             songlist.add(x);
103             Inorder_BST_Walk(x.getRight());
104         }
105     }
106
107     public song BST_Search(song x, int k){
108         if(x == null || k == x.getGenre()){
109             return x;
110         }
111         if(k < x.getGenre()){
112             return BST_Search(x.getLeft(), k);
113         }else{
114             return BST_Search(x.getRight(), k);
115         }
116     }
117 }

```

Red Black Tree Inorder Walk

- Click this button
- It will sort by RBTree - Inorder - Walk



| Song Database | | | | | | | | | | |
|---------------|-----------------------------|--------|-------|-------|----------|-----------|---------|-------|--|--|
| ID | Title | Artist | Album | Genre | Duration | MP3 Files | abOrder | Plays | Action | |
| 1 | 2Face | 1 | 1 | 1 | 4:46 | 🎵 | 1 | 2 | Update Delete HeapSort bstInorder rbtInorder | |
| 2 | Back To The Topic Freestyle | 1 | 1 | 1 | 3:00 | 🎵 | 2 | 2 | Update Delete HeapSort bstInorder rbtInorder | |
| 3 | Before Im Gone | 1 | 1 | 1 | 4:24 | 🎵 | 3 | 2 | Update Delete HeapSort bstInorder rbtInorder | |
| 4 | Best Friend | 1 | 1 | 1 | 3:25 | 🎵 | 4 | 2 | Update Delete HeapSort bstInorder rbtInorder | |
| 5 | Alien Girl | 2 | 2 | 1 | 4:00 | 🎵 | 1 | 10 | Update Delete HeapSort bstInorder rbtInorder | |
| 6 | Growing Apart | 2 | 2 | 1 | 4:23 | 🎵 | 2 | 4 | Update Delete HeapSort bstInorder rbtInorder | |
| 7 | Ignorance Is Bliss | 2 | 2 | 1 | 3:31 | 🎵 | 3 | 2 | Update Delete HeapSort bstInorder rbtInorder | |
| 8 | PP 15 | 2 | 2 | 1 | 6:02 | 🎵 | 4 | 2 | Update Delete HeapSort bstInorder rbtInorder | |
| 9 | The Heart Pt. 2 | 2 | 2 | 1 | 4:54 | 🎵 | 5 | 2 | Update Delete HeapSort bstInorder rbtInorder | |

```

1 package com.cs157a.web.jdbc;
2
3 import java.util.ArrayList;
4 import java.util.LinkedHashSet;
5 import java.util.Set;
6
7 /**
8  * A red-black tree is a binary search tree with one extra bit of storage per node: its color,
9  * which can be either RED or BLACK. By constraining the node colors on any simple path from the root to a leaf,
10 * red-black trees ensure that no such path is more than twice as long as any other,
11 * so that the tree is approximately balanced.
12 */
13 public class redBlackTree {
14
15     private redBlackSongNode root;
16     private redBlackSongNode nil;
17
18     //    protected Set<String> rblist;
19     protected ArrayList<redBlackSongNode> redBlackNodeslist;
20
21     public redBlackTree(){
22         //        nil = new redBlackSongNode("Tom", 99999999,"BLACK");
23         nil = new redBlackSongNode(45, "Red Black Tree", 17, 5, 7, "2:08", "resource/music/Intro.mp3", 5, 5, "BLACK");
24         setRoot(getNil());
25     //        rblist = new LinkedHashSet<>();
26     //        redBlackNodeslist = new ArrayList<>();
27     }
28
29     public redBlackSongNode getNil(){
30         return nil;
31     }
32
33     public void setRoot(redBlackSongNode root){
34         this.root = root;
35     }
36
37     public redBlackSongNode getRoot() {
38         return root;
39     }
40
41     public void Left_Rotate(redBlackSongNode x){
42         redBlackSongNode y = x.getRight();
43         x.setRight(y.getLeft());
44         if(y.getLeft() != getNil()){
45             y.getLeft().setParent(x);
46         }
47         y.setParent(x.getParent());
48         if(x.getParent() == getNil()){
49             setRoot(y);
50         }else if(x == x.getParent().getLeft()){
51             x.getParent().setLeft(y);
52         }else{
53             x.getParent().setRight(y);
54         }
55         y.setLeft(x);
56     }
57 }

```

```

99         RB_Insert_Fixup(z);
100    }
101
102    public void RB_Insert_Fixup(redBlackSongNode z){
103
104        while(z.getParent().getColor().equals("RED")){
105            if(z.getParent() == z.getParent().getParent().getLeft()){
106                //set the uncle to y, its grandparent right child
107                redBlackSongNode y = z.getParent().getParent().getRight();
108
109                if(y.getColor().equals("RED")){
110                    z.getParent().setColor("BLACK");           //case 1
111                    y.setColor("BLACK");                     //case 1
112                    z.getParent().getParent().setColor("RED"); //case 1
113                    z = z.getParent().getParent();           //case 1
114                }else{
115                    if(z == z.getParent().getRight()){
116                        z = z.getParent();                  //case 2
117                        Left_Rotate(z);                 //case 2
118                    }
119                    z.getParent().setColor("BLACK");       //case 3
120                    z.getParent().getParent().setColor("RED"); //case 3
121                    Right_Rotate(z.getParent().getParent()); //case 3
122                }
123            }else if(z.getParent() == z.getParent().getParent().getRight()){
124                redBlackSongNode y = z.getParent().getParent().getLeft();
125                if(y.getColor().equals("RED")){
126                    z.getParent().setColor("BLACK");           //case 1
127                    y.setColor("BLACK");                     //case 1
128                    z.getParent().getParent().setColor("RED"); //case 1
129                    z = z.getParent().getParent();           //case 1
130                }else{
131                    if(z == z.getParent().getLeft()){
132                        z = z.getParent();                  //case 2
133                        Right_Rotate(z);                //case 2
134                    }
135                    z.getParent().setColor("BLACK");       //case 3
136                    z.getParent().getParent().setColor("RED"); //case 3
137                    Left_Rotate(z.getParent().getParent()); //case 3
138                }
139            }
140        }
141        root.setColor("BLACK");
142    }
143
144    public void RB_Transplant(redBlackSongNode u, redBlackSongNode v){
145        if(u.getParent() == getNil()){
146            setRoot(v);
147        }else if(u == u.getParent().getLeft()){
148            u.getParent().setLeft(v);
149        }else{
150            u.getParent().setRight(v);
151        }

```

```
186     }
187
188     public void RB_Delete_Fixup(redBlackSongNode x){
189         while(x != root && x.getColor().equals("BLACK")){
190             if(x == x.getParent().getLeft()){
191                 redBlackSongNode w = x.getParent().getRight();
192                 if(w.getColor().equals("RED")){
193                     w.setColor("BLACK");                                //Case 1
194                     x.getParent().setColor("RED");                  //Case 1
195                     Left_Rotate(x.getParent());                   //Case 1
196                     w = x.getParent().getRight();                //Case 1
197                 }
198                 if(w.getLeft().getColor().equals("BLACK") && w.getRight().getColor().equals("BLACK")){
199                     w.setColor("RED");                            //Case 2
200                     x = x.getParent();                         //Case 2
201                 }
202                 else {
203                     if(w.getRight().getColor().equals("BLACK")){
204                         w.getLeft().setColor("BLACK");           //Case 3
205                         w.setColor("RED");                    //Case 3
206                         Right_Rotate(w);                  //Case 3
207                         w = x.getParent().getRight();        //Case 3
208                     }
209                     w.setColor(x.getParent().getColor());      //Case 4
210                     x.getParent().setColor("BLACK");        //Case 4
211                     w.getRight().setColor("BLACK");        //Case 4
212                     Left_Rotate(x.getParent());          //Case 4
213                     setRoot(x);                          //Case 4
214                 }
215             }
216             else {
217                 redBlackSongNode w = x.getParent().getLeft();
218                 if(w.getColor().equals("RED")){
219                     w.setColor("BLACK");                  //Case 1
220                     x.getParent().setColor("RED");        //Case 1
221                     Right_Rotate(x.getParent());       //Case 1
222                     w = x.getParent().getLeft();        //Case 1
223                 }
224                 if(w.getRight().getColor().equals("BLACK") && w.getLeft().getColor().equals("BLACK")){
225                     w.setColor("RED");                  //Case 2
226                     x = x.getParent();               //Case 2
227                 }
228             }
229         }
230         if(w.getLeft().getColor().equals("BLACK")){
231             w.getRight().setColor("BLACK");            //Case 3
232             w.setColor("RED");                      //Case 3
233             Left_Rotate(w);                      //Case 3
234             w = x.getParent().getLeft();           //Case 4
235         }
236         w.setColor(x.getParent().getColor());        //Case 4
237         x.getParent().setColor("BLACK");           //Case 4
238         w.getLeft().setColor("BLACK");            //Case 4
239         Right_Rotate(x.getParent());            //Case 4
240         setRoot(x);                          //Case 4
241     }
242 }
```

Summary: for User CRUD and Songs CRUD are done, this is JDBC project for all It's really helpful for a backend engineer to learn how to manipulate database and build a website by using JAVA, JSP + SERVLET.

Conclusion

Lessons Learned from Team 20

1. **Tom Zheng** - I learn a lot from this project, there are two parts in our project: PHP+MySQL and JDBC. Especially the php and java can perfectly manipulate the database. I try to make the part A close to the real life. A user can pick a favorite album, and playing a song. This is the first time I try to clone a project like Spotify streaming music. However, when I really started working on this project, I encountered a lot of challenges. I learn how to manipulate the database by using MySQL workbench. I understand how to get the data from database and output the data on our website; I learn how to use AJAX to convert the data from database to JSON data. This is a really important technique.
2. **Khoa Pham** - This project is the first time I have a chance to interact with the web design project type. The database system is a good way to learn and it is useful for managing databases if I want to work with it in the future. I realized that the program can't handle all of the data itself and creating punch of array and objects are not the only thing we can do with programming. More important thing is that we can work together to handle the music website. I like how we can play music in the website part. If I mess up with something in the databases, a lot of things will need to be changed or else the bug will just pop up. We, together, have learned new things from using github. SQL code make it great to demonstrate the way we need to know to take data in tons of database.
3. **Johnny Phenglavong** - This project has taught me a lot about communication, collaboration, version control and documentation. Our original goal was to create a database of movies that would alert users based on time of release for movies they chose to be reminded of. After some team discussion, we decided to switch into a database of songs for streaming as the functionalities of play/pause/seek/etc.. would be more appealing and interesting for the audience. The lessons learned came through various mistakes during our project, such as version control and documentation. I felt that had our documentation had been more comprehensive and straight to the point, we would have been able to clearly list our functional requirements for the demo as we ended up adding functions that were not originally in the requirements. Also, version control was a big obstacle for me as when

changes were thrown into our master branch some things did not work for my machine as it did on Tom's and Khoa's. This lead to me branching my latest version with organized albus into jp408 on github. The issue was that was that now when Tom added a bit more functionality on his end and into master, we were unable to merge properly with the fear that things would break before the demo so we scrambled for dummy data to present.

Future Improvements

Future improvements that we had in mind for our project included a search function, volume adjustment and also the ability to play music outside of the album page as when a user exits the album page, the music does not continue as it should. For the demo, we had a lot of visible features that made the project look presentable, but working on each functionality for a music streaming website was very difficult and time consuming. Especially since adding the data files and pathing it for our database took a lot of time to do as well. Communication, time and project management is a critical skill to have for group projects and with the workload of this course along with all of our other courses, it was definitely a huge learning lesson and obstacle we had to overcome.