

# Music Wire

BY

APOORVA SHUKLA (as11445)

SNEHA MUNDEN (sm7352)

# **Project Description**

Music Wire is a dynamic web based application driven by a database system to manage application data. It provides a seamless music experience where users can listen and browse music as per their choice. In this web application, the users can search for various tracks, artist, albums. Also, the users can create their own playlist, can follow another user's playlist. The users can follow the other users, like an artist. The users are given various recommendations about the tracks, albums, artists based on their likes and follows.

We have used MySQL and PHP to design this application.

GitHub Link for the project:

<https://github.com/Sneha2302/Music Wire>

## Table of Contents

<b>1. Back-end Design .....</b>	<b>4</b>
<b>1.1. Database Design .....</b>	<b>4</b>
<b>1.2. Entity Relationship Diagram: .....</b>	<b>5</b>
<b>1.3 IMPORTANT ENTITIES IN THE ER DIAGRAM: .....</b>	<b>6</b>
<b>1.4 Changes from Project 1.....</b>	<b>7</b>
<b>1.5 Validation and Sample Queries .....</b>	<b>8</b>
Part 1: Features .....	8
Part 2: Displaying the contents of the tables.....	14
<b>1.6 SQL Injections .....</b>	<b>17</b>
<b>1.7 Concurrency Control .....</b>	<b>17</b>
<b>2. Front End design .....</b>	<b>18</b>
<b>2.1 Website Structure.....</b>	<b>18</b>
<b>2.2 Application Features.....</b>	<b>19</b>
<b>3. Bibliography .....</b>	<b>29</b>

## 1. Back-end Design

The back-end for the application consists of a relational schema structured in-order to effectively store and manage music streaming. The data consists of information about songs, artists, users and tracks. This application helps us to stream tracks, like a track, give ratings, search and follow other users and displays several playlists like top tracks, top albums, latest releases, recently played and related artists. The numerous details about the database and the design is explained below.

### 1.1. Database Design

The design and schema structure is the main pillar and support for the application. This provides the foundation on which the web application is built and executes the required functions. The format and structure is defined in this section.

Our schema consists of the following tables:

1. User: The user table consists of all the information about the users like their username, password, email. The information is collected from the users via a signup form.
2. Artist: This contains the artists name and description which includes the genre of the artist.
3. Album: This table contains the information about Albums created by the Artists.
4. Playlist: This table contains the information about the Playlist created by the Users and also whether the user created playlist is Public or Private along with the Timestamp when the Playlist was created
5. Playlist Track: This table is used for storing information about which track exists in which playlist along with the Timestamp when the track was added by the user to the Playlist.
6. Track: The track table consists of the track details like title, duration, album to which it belongs and the track number of the album for the ordering of the tracks.
7. Likes: This table stores a record for a user when he/she likes an artist along with the timestamp of the like.
8. Follows: This table stores a record for a user when he/she follows another user along with the timestamp of the follow action.
9. Rating: This table stores a record for a user when he/she gives a rating (out of 5 stars) to a track along with the timestamp of the rating given.
10. Plays: This table stores a record for a user when he/she plays a track along with the timestamp of the play action.

## 1.2. Entity Relationship Diagram:

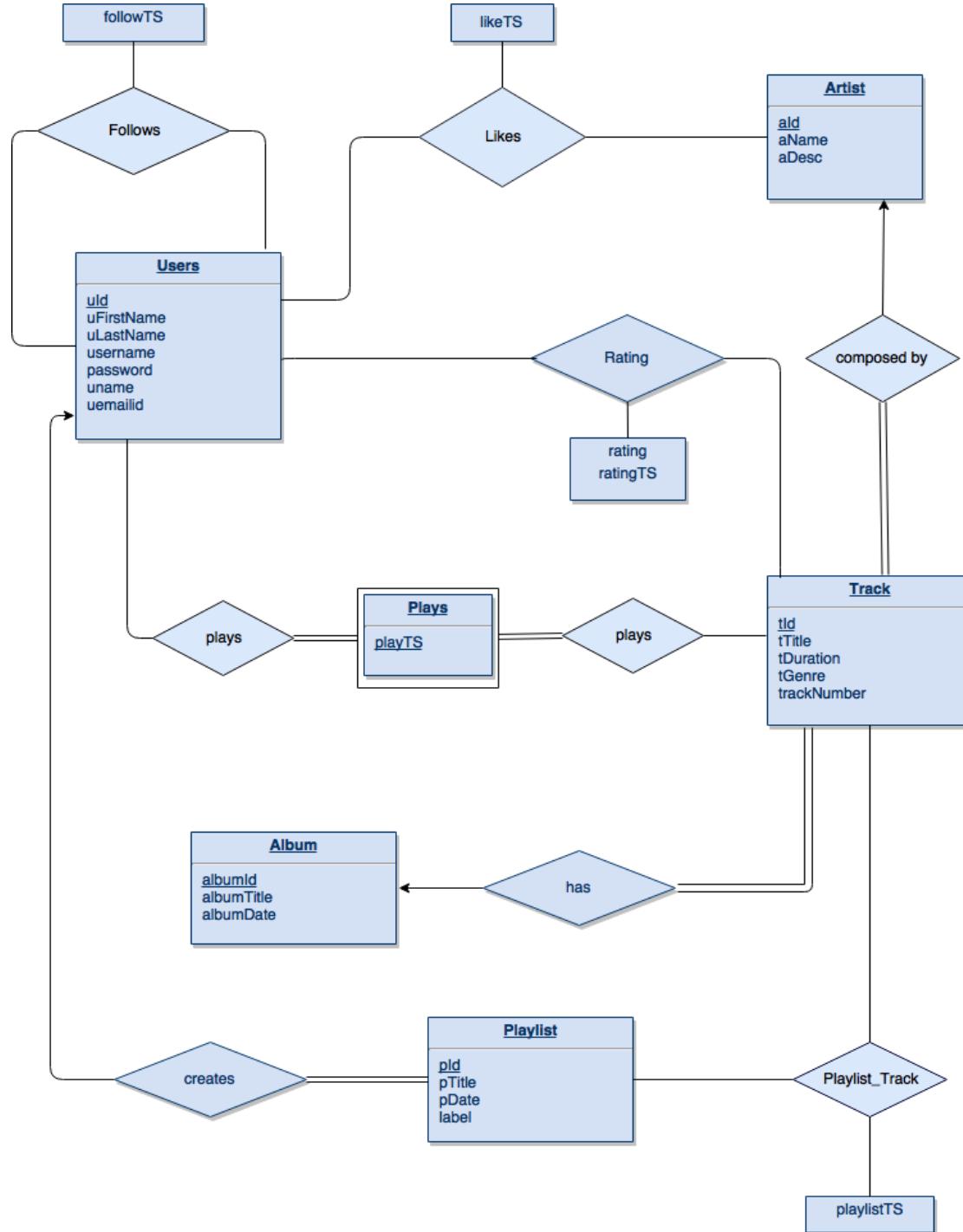


Fig 1. Entity Relationship Diagram of Music Wire

### 1.3 IMPORTANT ENTITIES IN THE ER DIAGRAM:

1. Users
2. Artist
3. Album
4. Track

Further describing each entity:

- a) Users: This entity contains detailed description about the user who is using the web application. This entity contains the attributes such as uid, username, uemailid, password, firstName, lastname which contains information such as user ID, username and password for logging into the website. Also it contains information such Email ID, First name and Last name of user given by the user during the sign up.
- b) Artist: This entity contains the detailed description of the artist such as ArtistID, artist name and description. An artist here is basically owner of the track. Here by an artist we mean a single person, a duo or a band. This attribute also has an attribute adesc which contains the description of the artist. By description we mean genre of the artist.
- c) Album: This entity contains details of albums which are an abstraction for both an album or a playlist in the database. An album is a collection of tracks. We can distinguish between an album and playlist by specifying an owner tag i.e N/Y. For album, we use owner tag N. For playlist, we use owner tag Y which means owner is a user.
- d) Track: This entity contains the information about the Tracks such as TrackId, Track duration and also the artist of the track.
- e) Playlist: This entity contains the detailed information about the user created Playlist. It contains the attribute such as pid which gives us Playlist ID and ptitle which gives us the name of the playlist given by the user. We also keep a record of the timestamp when the playlist was created by the user in the pDate and also we store the information whether the Playlist is Public or Private in Plabel attribute. If the user has marked is Playlist as Public then his playlist is visible to other users following him and if the user has marked it Private then it is not visible to other users.
- f) Plays: This is a weak entity. It is used to record each time a user plays a track along with the timestamp. It maps users to tracks.

The above entities are connected via the following relationships:

1. Follows: This relationship maps users to other users Cardinality: many to many
2. Likes: This relationship maps users to artists. Cardinality: many to many
3. Rating: This relationship maps users to tracks Cardinality: many to many
4. Playlist\_Track: This relationship maps playlists to the tracks that it contains. Cardinality: many : many

## 1.4 Changes from Project 1

Below are the changes that we have made in our database design in Project part 2

In our project 1 database schema we had only 9 tables whereas in project 2 database schema we have 10 tables. We have added a new table Playlist for keeping record of the playlist created by the user. In project one we were using Albums table to keep a record of both artist created album and user created playlist and distinguishing the two by using an owner tag. If the playlist has been created by the user we used tag Y else we used tag N to specify artist created Album.

Since we were using Albums table for keeping record of user created playlist in project 1 we used a table name Album\_track to record which track is added in which playlist by the user and the track number of the track in that album and also the timestamp to record when that track was added to the playlist. Here in Project 2 we have changed the name of the table from Albums\_Track to Playlist\_Track to function in the similar way i.e to keep the record of the tracks being added to the playlist of the user by the user and also the time at which the track is being added. And also, we don't need Albums\_Track table because as per our dataset each track is linked with only one album.

## 1.5 Validation and Sample Queries

From the database schema, it can be seen that there are distinguishable attributes, numerous references and relationships between data, dependencies between tables and many other components present in the model. The creation of an efficient data model must be verified and validated by using sample data. The sample test data must be designed carefully to examine and verify all functionality.

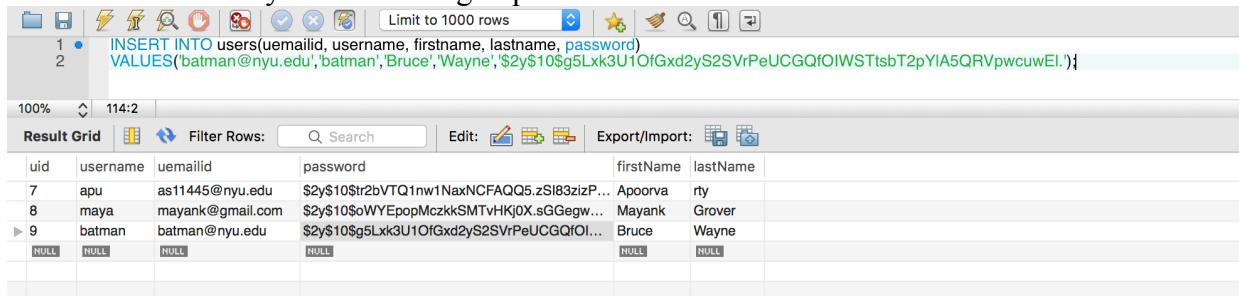
We will be validating our sample data if it can be validated against the required functionalities in the following three parts using our user Bruce Wayne.

### Part 1: Features

#### i. User Sign-up form –

For using the website, a user needs to sign up. A user can create his account by creating a username and a password and registering it with an Email ID and providing details such as First name and Last name. After signing up a user can log into the website using his username and password. All the passwords are encrypted using Hash and Salt before storing in the database.

New user Bruce Wayne wants to sign up!



The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL editor window contains the following code:

```
1 • INSERT INTO users(umailid, username, firstname, lastname, password)
2   VALUES('batman@nyu.edu','batman','Bruce','Wayne','$2y$10$g5Lxk3U1OfGxd2yS2SVrPeUCGQfOIWSTtsbT2pYIA5QRVpwcuwEl.')
```

The status bar at the bottom indicates "100%" and "114:2". Below the editor is the "Result Grid" tab, which displays the following data:

uid	username	umailid	password	firstname	lastName
7	apu	as11445@nyu.edu	\$2y\$10\$tr2bVTQ1nw1NaxNCFAQQ5.zSI83zizP...	Apoorva	rty
8	maya	mayank@gmail.com	\$2y\$10\$oWYEpOpMczikSMTvHKj0X.sGGe...w...	Mayank	Grover
9	batman	batman@nyu.edu	\$2y\$10\$g5Lxk3U1OfGxd2yS2SVrPeUCGQfOI...	Bruce	Wayne

Here, the password is being **encrypted** before inserting in the database using the hash function.

```
$email = stripslashes($email);
$username = stripslashes($username);
$fname = stripslashes($fname);
$lname = stripslashes($lname);
$password = stripslashes($password);
// To encrypt the password
$password_hash = password_hash($password, PASSWORD_DEFAULT);

// To protect MySQL injection for Security purpose
if (!$insert = $connection->prepare("INSERT INTO users(umailid, username, firstname, lastname, password) VALUES(?, ?, ?, ?, ?)")) {
    echo "Prepare failed: (" . $connection->errno . ") " . $connection->error;
}
if (!$insert->bind_param('sssss', $email, $username, $fname, $lname, $password_hash)) {
    echo "Binding parameters failed: (" . $insert->errno . ") " . $insert->error;
}

if (!$insert->execute()) {
    echo "Execute failed: (" . $insert->errno . ") " . $insert->error;
}
```

#### ii. User Update profile –

A user can also update his profile any time using the ‘Update details’ or ‘Change password’ option.

Existing user Bruce wants to edit his existing profile:

```

1 • UPDATE users SET firstname='Bruce', lastname='Wayne', uemailid='batman@gmail.com' WHERE uid=9;
2 • SELECT * FROM users;

```

uid	username	uemailid	password	firstName	lastName
7	apu	as11445@nyu.edu	\$2y\$10\$tr2bVTQ1nw1NaxNCFAQQ5.zSI83zizP...	Apoorva	rty
8	maya	mayank@gmail.com	\$2y\$10\$oWYEpopMczzkSMTvHKj0X.sGGegw...	Mayank	Grover
9	batman	batman@gmail.com	\$2y\$10\$g5Lxk3U1OfGxd2yS2SVrPeUCGQfOI...	Bruce	Wayne
NULL	NULL	NULL	NULL	NULL	NULL

iii. Like an Artist–

Bruce wants to like an artist

```

1 • INSERT INTO likes(uid, aid, likeTS) VALUES(9, '0du5cEVh5yTK9QJze8zA0C', '2017-12-12 19:39:52');
2 • SELECT * FROM likes;

```

uid	aid	likeTS
7	1uNFoZAHBGtllmzznpCl3s	2017-12-12 22:27:58
7	1Xyo4u8uXC1ZmMpatF05PJ	2017-12-12 19:39:52
9	0du5cEVh5yTK9QJze8zA0C	2017-12-12 19:39:52
9	1uNFoZAHBGtllmzznpCl3s	2017-12-12 20:20:01
9	1Xyo4u8uXC1ZmMpatF05PJ	2017-12-12 20:12:29
NULL	NULL	NULL

Unlike an Artist –

Bruce wants to unlike an Artist.

```

1 • DELETE FROM likes WHERE uid=9 and aid='0du5cEVh5yTK9QJze8zA0C';
2 • SELECT * FROM likes;

```

uid	aid	likeTS
7	1uNFoZAHBGtllmzznpCl3s	2017-12-12 22:27:58
7	1Xyo4u8uXC1ZmMpatF05PJ	2017-12-12 19:39:52
9	1uNFoZAHBGtllmzznpCl3s	2017-12-12 20:20:01
9	1Xyo4u8uXC1ZmMpatF05PJ	2017-12-12 20:12:29
NULL	NULL	NULL

Bruce wants to follow other users

1 • INSERT INTO follows(uid1, uid2, followTS) VALUES(9,7,'2017-12-12 19:39:52');  
 2 • SELECT \* FROM follows;

100% 23:2

**Result Grid** Filter Rows: Search Edit: Export/Import:

uid1	uid2	followTS
7	8	2017-12-12 19:28:24
7	9	2017-12-12 20:21:22
8	7	2017-12-12 19:26:39
9	7	2017-12-12 19:39:52
9	8	2017-12-12 20:17:28

Bruce wants to unfollow the users he followed.

1 • DELETE FROM follows WHERE uid1=9 and uid2=7;  
 2 • SELECT \* FROM follows;

100% 1:3

**Result Grid** Filter Rows: Search Edit: Export/Import:

uid1	uid2	followTS
7	8	2017-12-12 19:28:24
7	9	2017-12-12 20:21:22
8	7	2017-12-12 19:26:39
9	8	2017-12-12 20:17:28
NULL	NULL	NULL

#### iv. Rate a track –

Bruce wants to rate a track

1 • UPDATE rating SET rating=4 , ratingTS='2017-12-12 19:39:52' WHERE uid=9 and trackid='15tZCtYNEcpiePliMoF7W';  
 2 • SELECT \* FROM rating;

100% 22:2

**Result Grid** Filter Rows: Search Edit: Export/Import:

uid	trackId	rating	ratingTS
7	09CtPGIpYB4BrO8qb1RGsF	4	2017-12-12 18:54:00
7	0QLXkoV7vX86QEibgLe6zQ	4	2017-12-12 22:23:05
7	15tZCtYNEcpiePliMoF7W	4	2017-12-12 19:08:22
7	25khomWgBVamSdKw7hzm3l	5	2017-12-12 21:38:54
7	2BPX0vWwwwXjuJuXV1Ucd5b	5	2017-12-12 19:18:56
7	4ttUYy7HMYQ0nhflLJNPR	1	2017-12-12 21:44:19
7	6b8Be6ljOzmkOmFslEb23P	2	2017-12-12 21:30:45
7	6RsWqX8zABZLhZydXxFOM	5	2017-12-12 19:39:12
7	7t5m6zRwlEBxmrPt8lNoPPi	5	2017-12-12 19:39:33
9	0QLXkoV7vX86QEibgLe6zQ	4	2017-12-12 20:14:12
9	66UVpCZ5aH3VV3lc3PBUrP	3	2017-12-12 20:14:48

#### v. Create/Delete a playlist

Bruce wants to create a public playlist

The screenshot shows the MySQL Workbench interface with two queries in the SQL editor:

```

1 • INSERT INTO playlist(uid, ptitle, pdate, plabel) VALUES(9,'New Playlist','2017-12-12 19:39:52','public');
2 • SELECT * FROM playlist;

```

The Result Grid displays the following data:

plid	pTitle	pDate	uid	pLabel
10	My favs	2017-12-12 19:08:35	7	public
11	Mayank's playlist	2017-12-12 19:27:50	8	private
12	maya's open playlist	2017-12-12 19:32:23	8	public
13	Batman's Playlist	2017-12-12 20:12:19	9	public
14	My favs	2017-12-12 20:16:17	9	public
16	New Playlist	2017-12-12 19:39:52	9	public

#### vi. Add to playlist:

Bruce wants to add new track to his playlist

The screenshot shows the MySQL Workbench interface with two queries in the SQL editor:

```

1 • INSERT INTO playlist_track(pid, tid, playlistTS) VALUES(10,'0tgVpDi06FyKpA1z0VMD4v','2017-12-12 19:39:52');
2 • SELECT * FROM playlist_track;

```

The Result Grid displays the following data:

pid	tid	playlistTS
11	09CtPGIpYB4BrO8qb1RGsF	2017-12-12 19:27:52
13	09FhDfmrSDN162X1yFfapl	2017-12-12 20:12:21
13	0QLXkoV7vX86QEibgLe6zQ	2017-12-12 20:14:22
10	0tgVpDi06FyKpA1z0VMD4v	2017-12-12 19:39:52
10	2BPX0vWwwwXjJuXV1UcD5b	2017-12-12 19:16:54
13	3lU6xbfNZosvTdRrGs8b2u	2017-12-12 20:13:11
10	4B0JvthVoAAuygILe3n4Bs	2017-12-12 19:22:40
14	4B0JvthVoAAuygILe3n4Bs	2017-12-12 20:16:21
10	4ttUYy7HMYQ0nhfllJNPR	2017-12-12 21:44:26
12	5SqSckut3FcoQKmGkMWgp1	2017-12-12 19:32:25
10	6b8Be6ljOzmkOmFslEb23P	2017-12-12 21:30:54
10	6RsWqX8zABZLhZydXxEFOm	2017-12-12 19:39:03
10	7t5m6zRwlEBxmPt8lNoPPi	2017-12-12 19:39:28

#### vii. Remove from playlist

Bruce wants to delete a track from his playlist

The screenshot shows the MySQL Workbench interface with two queries in the SQL editor:

```

2 • DELETE FROM playlist_track WHERE pid=10 and tid='0tgVpDi06FyKpA1z0VMD4v';
3 • SELECT * FROM playlist_track;
4 •
5 •

```

The Result Grid displays the following data:

pid	tid	playlistTS
11	09CtPGIpYB4BrO8qb1RGsF	2017-12-12 19:27:52
13	09FhDfmrSDN162X1yFfapl	2017-12-12 20:12:21
13	0QLXkoV7vX86QEibgLe6zQ	2017-12-12 20:14:22
10	2BPX0vWwwwXjJuXV1UcD5b	2017-12-12 19:16:54
13	3lU6xbfNZosvTdRrGs8b2u	2017-12-12 20:13:11
10	4B0JvthVoAAuygILe3n4Bs	2017-12-12 19:22:40
14	4B0JvthVoAAuygILe3n4Bs	2017-12-12 20:16:21
10	4ttUYy7HMYQ0nhfllJNPR	2017-12-12 21:44:26
12	5SqSckut3FcoQKmGkMWgp1	2017-12-12 19:32:25
10	6b8Be6ljOzmkOmFslEb23P	2017-12-12 21:30:54
10	6RsWqX8zABZLhZydXxEFOm	2017-12-12 19:39:03
10	7t5m6zRwlEBxmPt8lNoPPi	2017-12-12 19:39:28

viii. Most Played Tracks by all the users

Bruce wants to view **most played tracks** by all the users

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
SELECT distinct tid,title,aname from ((select trackid, count(*) cnt from plays group by trackid order by cnt desc limit 10) a join track on a.trackid=track.tid) natural join artist;
```

The result grid displays 10 rows of data:

tid	title	aname
0QLXkoV7vX86QEibgLe6zQ	I Miss You	Adele
09FhdImrSDN162X1yFfapl	False Alarm	The Weeknd
4ttUYy7HMYQ0nhflIJNPR	Miss You So Much	Miley Cyrus
25khomWgBVamSdKw7hzm3l	The Hills	The Weeknd
2BPXoVWwwwXjuXv1Ucd5b	Key to the Highway - Live in San Diego	Eric Clapton
6RsWqX8zABZLhZydxEFOm	Can't Feel My Face	The Weeknd
09CIPGlpYB4BrOBqp1RGsF	Sorry	Justin Bieber
715m6zRwlEBxmP8lNoPPi	Starboy	The Weeknd
66UVpCZ5aH3VV3lc3PBUrP	Shameless	The Weeknd
6b8Be6ljOzmkOmFsIb23P	24K Magic	Bruno Mars

ix. Best Rated Tracks by all the users

Bruce wants to view **best rated tracks**.

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
SELECT tid,title,aname from ((SELECT trackid, sum(rating) cnt FROM rating group by trackid order by cnt desc limit 10) a join track on a.trackid=track.tid) natural join artist;
```

The result grid displays 10 rows of data:

tid	title	aname
0QLXkoV7vX86QEibgLe6zQ	I Miss You	Adele
25khomWgBVamSdKw7hzm3l	The Hills	The Weeknd
6RsWqX8zABZLhZydxEFOm	Can't Feel My Face	The Weeknd
2BPXoVWwwwXjuXv1Ucd5b	Key to the Highway - Live in San Diego	Eric Clapton
715m6zRwlEBxmP8lNoPPi	Starboy	The Weeknd
09CIPGlpYB4BrOBqp1RGsF	Sorry	Justin Bieber
15iZCYNEcpIEPiIMOfe7W	Kisi Roz Tumse	Kumar Sanu
66UVpCZ5aH3VV3lc3PBUrP	Shameless	The Weeknd
6b8Be6ljOzmkOmFsIb23P	24K Magic	Bruno Mars
4ttUYy7HMYQ0nhflIJNPR	Miss You So Much	Miley Cyrus

x. Most Liked (Top artists) by all the users

Bruce wants to view **most liked artist**.

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
SELECT aid,aname from (SELECT aid, count(*) cnt FROM likes group by aid order by cnt desc limit 10) a natural join artist;
```

The result grid displays 2 rows of data:

aid	aname
1uNFoZAHBGtllmzznpCl3s	Justin Bieber
1Xyo4u8uXC1ZmMpatF05PJ	The Weeknd

xi. Latest tracks released by Artists the user liked.

Bruce wants to view **latest tracks released** by the artist he liked

The screenshot shows a MySQL Workbench interface with a query editor and a result grid. The query is:

```

4 • SELECT distinct tid,ttitle,aname
  from artist natural join track natural join album
  where aid in (select aid from likes where uid=7) order by albumDate desc limit 10;

```

The result grid displays the following data:

tid	ttitle	aname
09FhDfmrSDN162X1yFapl	False Alarm	The Weeknd
Ayv6465LBwy7tOx78bc	Party Monster	The Weeknd
6kP57bEoT8R9yltGUfUTmB	Secrets	The Weeknd
3IU6xbfNZosvTdRrGs8b2u	Reminder	The Weeknd
4B0JvthVoAAuygILe3n4Bs	baby	Justin Bieber
6RsWqX8zABZLhZydXxEFOm	Can't Feel My Face	The Weeknd
09CtPGlpYB4BrO8qb1RGsF	Sorry	Justin Bieber
5SqSckut3FcoQKmGkMWgp1	Often	The Weeknd
2SnNaoNhMjC1WRMTWD8qTX	Mark My Words	Justin Bieber
50kpGaPAhYJ3sGmk6vplg0	Love Yourself	Justin Bieber

- xii. New playlists created by users they follow –  
Bruce wants to see playlist created by other users he follows

The screenshot shows a MySQL Workbench interface with a query editor and a result grid. The query is:

```

4 • SELECT distinct pid, ptitle
  FROM playlist WHERE uid=7 and plabel='public';

```

The result grid displays the following data:

pid	ptitle
10	My favs

- xiii. Related Artists –

Bruce can also see related artist who belongs to the same genre when he opens an artist page

The screenshot shows a MySQL Workbench interface with a query editor and a result grid. The query is:

```

4 • SELECT distinct aid,aname
  FROM artist
  WHERE adesc REGEXP 'dancelpoplpoplpost-teenlpop' and aid!='4dpARuHxo51G3z768sgnrY' limit 10;

```

The result grid displays the following data:

aid	aname
0du5cEvh5yTK9QJze8zA0C	Bruno Mars
1mYsTxnqsietFxj1OgoGbG	A.R. Rahman
1uNFoZAHBGtlImzznpCl3s	Justin Bieber
1Xyo4u8uXC1ZmMpaf05PJ	The Weeknd
2DIGxzQSjYe5N6G9nkYghR	Jennifer Lopez
3TVXtAsR1Inumwj472S9r4	Drake
4K6blSRoklNdpw4mzLxwfn	Kumar Sanu
5YGY8feqx7naU7z4HrwZM6	Miley Cyrus
6eUKZXaKkcvilH0Ku9w2n3V	Ed Sheeran
6jJ0s89eD6GaHleKKya26X	Katy Perry

xiv. Recently Played

The user also has the capability of going back to the songs he/she recently played. This is displayed using the timestamp value of the play table.

Bruce can also see most recently played songs by him.

```

SELECT distinct tid, ttitle, aname
from plays,track,artist
where plays.trackid = track.tid and track.aid=artist.aid and uid=9 order by playsTS desc limit 10;

```

tid	ttitle	aname
66UVpcZ5aH3VV3lc3PBUrP	Shameless	The Weeknd
0QLXkoV7vX86QEibgLe6zQ	I Miss You	Adele
6RsWqX8zABZLhZydXxF0m	Can't Feel My Face	The Weeknd
09FhDfmrSDN162X1yFapl	False Alarm	The Weeknd

Part 2: Displaying the contents of the tables

i. User

uid	username	uemailid	password	firstName	lastName
7	apu	as11445@nyu.edu	\$2y\$10\$tr2bVTQ1nw1NaxNCFAQQ5.zSI83zizP...	Apoorva	rty
8	maya	mayank@gmail.com	\$2y\$10\$oWYEpopMczkkSMTvHKj0X.sGGegw...	Mayank	Grover
9	batman	batman@gmail.com	\$2y\$10\$g5Lxk3U1OfGxd2yS2SVrPeUCGQfOI...	Bruce	Wayne

ii. Artist

aid	aname	adesc
0du5cEVh5yTK9QJze8zA0C	Bruno Mars	dance pop pop pop rap post-teen pop
1mYsTxnqsietFxj1OgoGbG	A.R. Rahman	desi desi hip hop filmi indian pop
1uNFoZAHBGtllmzznpCl3s	Justin Bieber	canadian pop dance pop pop pop christmas pos...
1Xyo4u8uXC1ZmMpatF05PJ	The Weeknd	canadian pop pop rap
2DGxzQSjYe5N6G9nkYghR	Jennifer Lopez	dance pop hip pop pop pop rap post-teen pop r...
3TVXtAsR1Inumwj472S9r4	Drake	canadian pop hip hop pop rap
4dpARuHxo51G3z768sgnrY	Adele	dance pop pop post-teen pop
4K6blSRoklNdpw4mzLxwfn	Kumar Sanu	desi desi hip hop filmi indian folk indian pop
4VMYDCV2lEDYJArk749S6m	Daddy Yankee	latin latin hip hop reggaeton tropical
5YGY8feqx7naU7z4HrwZM6	Miley Cyrus	dance pop pop pop christmas post-teen pop
6eUKZXaKkcvIh0Ku9w2n3V	Ed Sheeran	pop
6jJ0s89eD6GaHleKKyA26X	Katy Perry	dance pop pop pop christmas post-teen pop
6PAAt558ZEZl0DmdXInjMgD	Eric Clapton	album rock blues-rock classic rock electric blues...
6S2OmqaRrzbos0tKUEyXyp	Demi Lovato	dance pop pop pop christmas post-teen pop
6zFYqv1mOsgBRQbae3JJ9e	Billy Joel	adult standards album rock classic rock folk roc...
7bXgB6jMjp9ATFy66eO08Z	Chris Brown	dance pop pop pop christmas pop rap r&b rap s...
7oPftvlwr6VrsViSDV7fJY	Green Day	alternative rock modern rock permanent wave p...

### iii. Albums

albumId	albumTitle	albumDate
► 03ntx95u0wotf68NnE3aGw	Witness	0006-09-17 00:00:00
09fggMHib4YkOtWQNKEBII	Legend of the Fall	2017-12-05 00:00:00
0AcfF3TmyOFCA49pIeIN9x	The Flying Lotus	2010-06-17 00:00:00
0K4pI0OsJ9lK8OjrZfxzD	25	0000-00-00 00:00:00
1XY618HWkwYKJWBRYR4MK	More Life	0000-00-00 00:00:00
1YauqJNIl3nVYKglGnUdCU	Live In The Studio - Sigma Studios 1972	0009-07-17 00:00:00
28ZKQMoNBB0etKXZ97G2SN	Beauty Behind The Madness	0000-00-00 00:00:00
3eGhplCuytpKNL1CBluOU	Live in San Diego (with Special Guest JJ Cale	0000-00-00 00:00:00
3T4tUhGYeRNvUGevb0wThu	It (Deluxe)	0003-03-17 00:00:00
3zak0kNLcOY5vFcB3IpSKP	Heartbreak on a Full Moon	0000-00-00 00:00:00
4PgleR09JVnm3zY1fW3XBA	24K Magic	0000-00-00 00:00:00
511XZdiHwdw9etY3oCKXPa	Bhala Pauchi	0000-00-00 00:00:00
5a3LqvNt2nv1B4aRKXmgOV	Revolution Radio	2010-07-16 00:00:00
5m43SVd4ajhA7M88UwzU8a	King Daddy	0001-01-13 00:00:00
5Slv8bswuGB6inCuWbWCa2	Tell Me You Love Me	0000-00-00 00:00:00
5xG9gJcs9ut3qDWezHUIsX	Younger Now	0000-00-00 00:00:00
6Fr2rQkZ383FcMqFyT7yPr	Purpose (Deluxe)	0000-00-00 00:00:00
7DoFFUz6BAVBwUFaMCTTC	A.K.A. (Deluxe)	0000-00-00 00:00:00

### iv. Track

tid	aid	ttitle	tduration	albumId	tlink	tracknumber
09FhDfmrSDN162X1yFfapl	1xyo4u8uXC1zMrpaf05PJ	False Alarm	220306	09fggMHib4YkOtWQNKEBII	<a href="https://www.youtube.com/embed/CW5oGRx9CLM">https://www.youtube.com/embed/CW5oGRx9CLM</a>	2
0AQQWNhWxNY1GXfsylN6X	4K6blSRokNdpw4mzLxwfn	Sunara Sabari	333320	511XZdiHwdw9etY3oCKXPa	<a href="https://www.youtube.com/embed/nU1uHUtSQA">https://www.youtube.com/embed/nU1uHUtSQA</a>	5
0KKkJNfGyhQ5aFogxQAPU	0du5cEVh5yTK9Qjze8zA0C	That's What I Like	206693	4PgleR09JVnm3zY1fW3XBA	<a href="https://www.youtube.com/embed/PMivT7Mj41M">https://www.youtube.com/embed/PMivT7Mj41M</a>	4
0Kn8xEmgMW9mh7UmDYHJP	0du5cEVh5yTK9Qjze8zA0C	Versace On The Floor	261240	4PgleR09JVnm3zY1fW3XBA	<a href="https://www.youtube.com/embed/FyEnolgTM">https://www.youtube.com/embed/FyEnolgTM</a>	5
0mBKv9DkYfQHjdMcw2dyJ	0du5cEVh5yTK9Qjze8zA0C	Chunky	186973	4PgleR09JVnm3zY1fW3XBA	<a href="https://www.youtube.com/embed/oacq_1TkMU">https://www.youtube.com/embed/oacq_1TkMU</a>	2
0PAirNkCRRIhV7hShF5s7Rr	7oPftlw6VrsVisDV7JY	Boulevard of broken dreams	302333	5a3LqvNt2nv1B4aRKXmgOV	<a href="https://www.youtube.com/embed/Soa3gO7Lc">https://www.youtube.com/embed/Soa3gO7Lc</a>	5
0QLXkoV7vX86QEibgLe6zQ	4dpARuHxo51G3z768sgnRY	I Miss You	348626	0K4pI0OsJ9lK8OjrZfxzD	<a href="https://www.youtube.com/embed/JDBO9x5YwDU">https://www.youtube.com/embed/JDBO9x5YwDU</a>	3
0SNIArtCPVVLoGEpcuHSIC	1uNfoZAHBGtlmzznpCl3s	What do you mean	199946	6Fr2rQkZ383FcMqFyT7yPr	<a href="https://www.youtube.com/embed/DK_0jXPulr">https://www.youtube.com/embed/DK_0jXPulr</a>	2
0tgVpDi06FyKpA1z0VMD4v	6eUKZXaKkcvih0Ku9w2n3V	Perfect	263400	3T4tUhGYeRNvUGevb0wThu	<a href="https://www.youtube.com/embed/2Vv-BfVoq4g">https://www.youtube.com/embed/2Vv-BfVoq4g</a>	5
13e6f8t7RKxuZ0daauJRG	3TVXAsR1lnumwji472S9r4	Hotline Bling	107673	1XY618HWkwYKJWBRYR4MK	<a href="https://www.youtube.com/embed/uxpDa-c-4Mc">https://www.youtube.com/embed/uxpDa-c-4Mc</a>	4
15tZCtYNEcpiePlMOFE7W	Kisi Roz Tumse		44880	511XZdiHwdw9etY3oCKXPa	<a href="https://www.youtube.com/embed/7ViolQgm_80">https://www.youtube.com/embed/7ViolQgm_80</a>	4
19vd1Zl8WmYrl0upZlOK9	2DGxzQSjYe5n6G9nKyghR	Dance on the floor	228666	7DoFFUz6BAVBwUFaMCTTC	<a href="https://www.youtube.com/embed/t4H_Zoh7G5A">https://www.youtube.com/embed/t4H_Zoh7G5A</a>	1
1g8znKyO3F1ykMHCGrkKq	4VMYDCV2IEDYJArk749S6m	D?_nde Es El Party	214066	5m43SVd4ajhA7M88UwzU8a	<a href="https://www.youtube.com/embed/rAwXrmeFMK4">https://www.youtube.com/embed/rAwXrmeFMK4</a>	5

### v. Plays

trackId	uld	playsTS
09FhDfmrSDN162X1yFfapl	7	2017-12-12 18:55:04
09FhDfmrSDN162X1yFfapl	7	2017-12-12 20:48:19
09FhDfmrSDN162X1yFfapl	7	2017-12-12 20:59:23
09FhDfmrSDN162X1yFfapl	7	2017-12-12 21:35:48
09FhDfmrSDN162X1yFfapl	7	2017-12-12 21:36:43
0QLXkoV7vX86QEibgLe6zQ	7	2017-12-12 19:04:30
0QLXkoV7vX86QEibgLe6zQ	7	2017-12-12 19:12:56
0QLXkoV7vX86QEibgLe6zQ	7	2017-12-12 19:18:32
0QLXkoV7vX86QEibgLe6zQ	7	2017-12-12 21:35:32
0QLXkoV7vX86QEibgLe6zQ	7	2017-12-12 21:35:42
0QLXkoV7vX86QEibgLe6zQ	7	2017-12-12 22:22:43
0QLXkoV7vX86QEibgLe6zQ	7	2017-12-12 22:25:34
15tZCtYNEcpiePlMOFE7W	7	2017-12-12 19:08:05
15tZCtYNEcpiePlMOFE7W	7	2017-12-12 21:36:51
25khomWgBVamSdKw7hzm3l	7	2017-12-12 21:38:18
25khomWgBVamSdKw7hzm3l	7	2017-12-12 21:38:30
25khomWgBVamSdKw7hzm3l	7	2017-12-12 21:38:51
25khomWgBVamSdKw7hzm3l	7	2017-12-12 21:42:44

vi. Follows

	uld1	uld2	followTS
▶	7	8	2017-12-12 19:28:24
	7	9	2017-12-12 20:21:22
	8	7	2017-12-12 19:26:39
	9	7	2017-12-12 19:39:52
	9	8	2017-12-12 20:17:28

vii. Likes

	uld	aid	likeTS
▶	7	1uNFoZAHBGtllmzznpCl3s	2017-12-12 22:27:58
	7	1Xyo4u8uXC1ZmMpatF05PJ	2017-12-12 19:39:52
	9	1uNFoZAHBGtllmzznpCl3s	2017-12-12 20:20:01
	9	1Xyo4u8uXC1ZmMpatF05PJ	2017-12-12 20:12:29

viii. Rating

	uld	trackId	rating	ratingTS
▶	7	09CtPGIpYB4BrO8qb1RGsF	4	2017-12-12 18:54:00
	7	0QLXkoV7vX86QEibgLe6zQ	4	2017-12-12 22:23:05
	7	15tZCtYNNEcpiePliMoF7W	4	2017-12-12 19:08:22
	7	25khomWgBVamSdKw7hzm3I	5	2017-12-12 21:38:54
	7	2BPX0vWwwwuXjJuXV1UcD5b	5	2017-12-12 19:18:56
	7	4ttUYy7HMYQ0nhflLJNPR	1	2017-12-12 21:44:19
	7	6b8Be6ljOzmkOmFslEb23P	2	2017-12-12 21:30:45
	7	6RsWqX8zABZLhZydXxFOM	5	2017-12-12 19:39:12
	7	7t5m6zRwlEBxmPt8lNoPPi	5	2017-12-12 19:39:33
	9	0QLXkoV7vX86QEibgLe6zQ	4	2017-12-12 20:14:12
	9	66UVpCZ5aH3VV3lc3PBUrP	3	2017-12-12 20:14:48

ix. Playlist

	pld	pTitle	pDate	uld	pLabel
▶	10	My favs	2017-12-12 19:08:35	7	public
	11	Mayank's playlist	2017-12-12 19:27:50	8	private
	12	maya's open playlist	2017-12-12 19:32:23	8	public
	13	Batman's Playlist	2017-12-12 20:12:19	9	public
	14	My favs	2017-12-12 20:16:17	9	public
	16	New Playlist	2017-12-12 19:39:52	9	public

## x. Playlists\_Track

plId	tId	playlistTS
11	09CtPGIpYB4BrO8qb1RGsF	2017-12-12 19:27:52
13	09FhDfmrSDN162X1yFapl	2017-12-12 20:12:21
13	0QLXkoV7vX86QEibgLe6zQ	2017-12-12 20:14:22
10	2BPX0vWwwuXjJuXV1UcD5b	2017-12-12 19:16:54
13	3IU6xbfNZosvTdRrGs8b2u	2017-12-12 20:13:11
10	4B0JvthVoAAuyglLe3n4Bs	2017-12-12 19:22:40
14	4B0JvthVoAAuyglLe3n4Bs	2017-12-12 20:16:21
10	4tUYy7HMYQ0nhfillJNPR	2017-12-12 21:44:26
12	5SqSckut3FcoQKmGkMWgp1	2017-12-12 19:32:25
10	6b8Be6ljOzmkOmFslEb23P	2017-12-12 21:30:54
10	6RsWqX8zABZLhZydXxFOM	2017-12-12 19:39:03
10	7t5m6zRwIEBxmPt8lNoPPi	2017-12-12 19:39:28

## 1.6 SQL Injections

SQL injection (or a SQL injection attack) occurs when a user provides SQL code as user input for a Web page, and the SQL code is then executed in the database. We are using prepared statements and parameterized queries in PHP to protect our website against cross-site scripting. These are SQL statements that are sent to and parsed by the database server separately from any parameters. So, we execute mysqli queries using prepared statements in the following way:

Example:

```
// To protect MySQL injection for Security purpose
if (!$insert = $connection->prepare("INSERT INTO users(umailid, username, firstname, lastname, password) VALUES(?, ?, ?, ?, ?)")) {
    echo "Prepare failed: (" . $connection->errno . ") " . $connection->error;
}
if (!$insert->bind_param('sssss', $email, $username, $fname, $lname, $password_hash)) {
    echo "Binding parameters failed: (" . $insert->errno . ") " . $insert->error;
}

if (!$insert->execute()) {
    echo "Execute failed: (" . $insert->errno . ") " . $insert->error;
}
```

## 1.7 Concurrency Control

In a multiprogramming environment where multiple transactions can be executed simultaneously, it is highly important to control the concurrency of transactions. In this application, it is possible for multiple users to use the site at the same time.

## 2. Front End design

In an increasingly web-based world, a webpage is often its first impression as well as its first interaction with customers. We need to take this into account as we plan out a successful front-end development plan. The attractiveness of the website alone is not enough, it must interact with the database efficiently and at the same time be user friendly and simple.

### 2.1 Website Structure

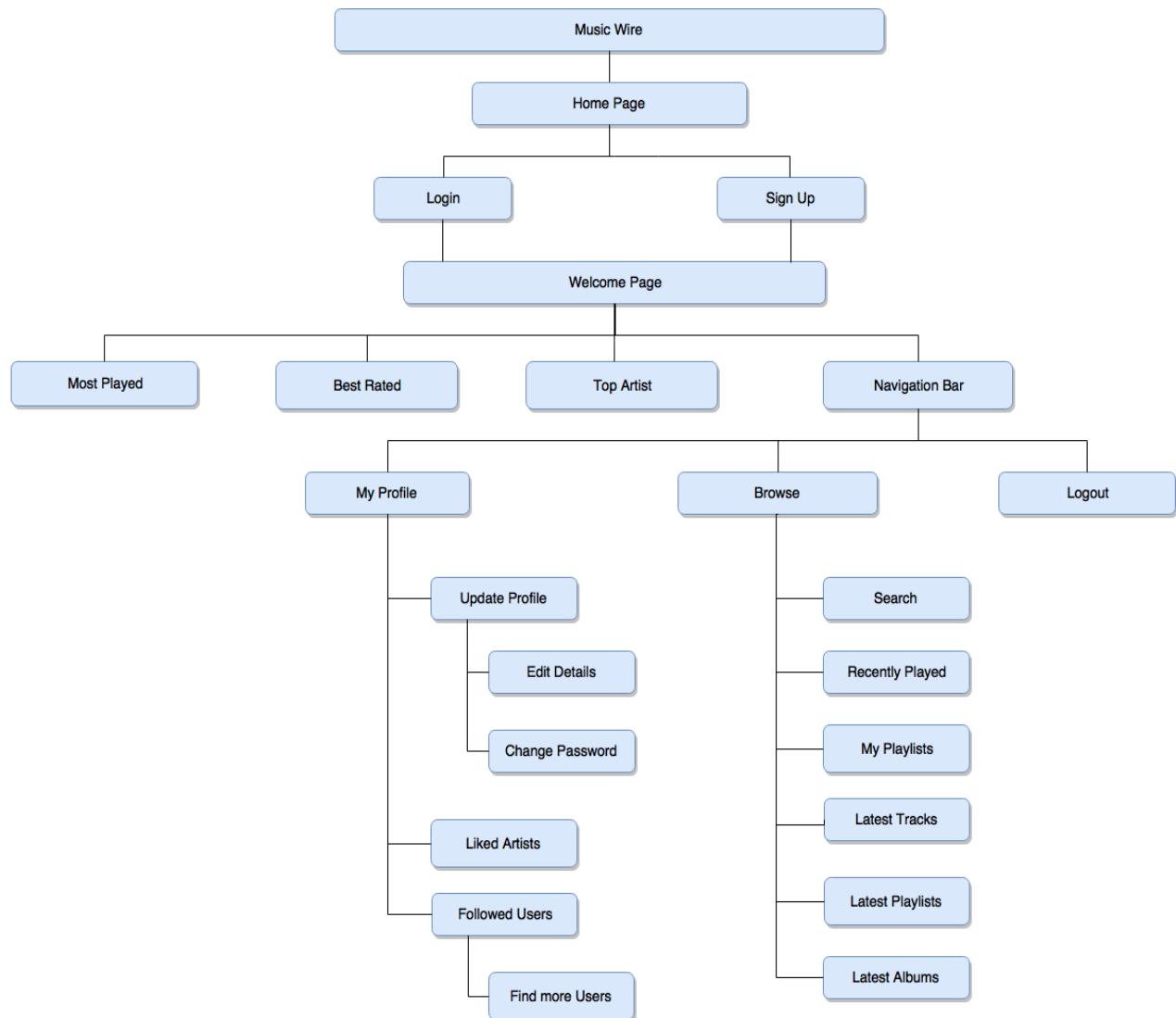


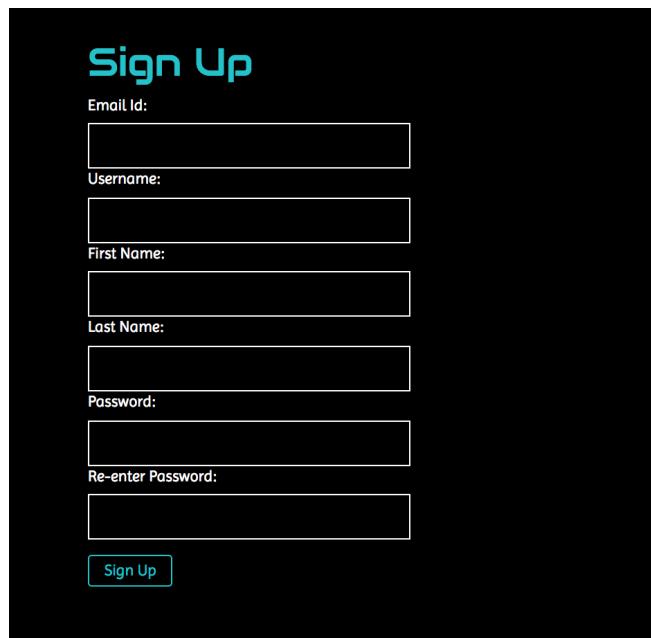
Fig 2. Music Wire website structure

## 2.2 Application Features

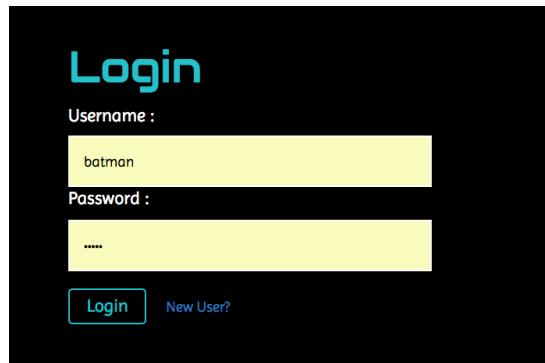
Music Wire contains numerous features to provide its users with a seamless music experience.

### Login/Sign Up:

The user is first greeted in the Home page from where he can select Login/Sign Up. To create an account, the user can click on the Sign-Up button. In this page, the user can provide his details and create credentials to login to the website. The sign up page also checks if an email ID is already registered on the website. There is also a feature to edit the profile information if the user wishes to do so. The timestamp at which users create their account, edits his profile and last logs are recorded and updated if required. The password is being encrypted before storing in the database.



The image shows a 'Sign Up' form with a black background and white text. The title 'Sign Up' is at the top in a large, bold font. Below it are six input fields with labels: 'Email Id:', 'Username:', 'First Name:', 'Last Name:', 'Password:', and 'Re-enter Password:'. Each label is followed by a rectangular input box. At the bottom is a blue 'Sign Up' button.



The image shows a 'Login' form with a black background and white text. The title 'Login' is at the top in a large, bold font. Below it are two input fields with labels: 'Username:' and 'Password:'. The 'Username:' field contains the text 'batman'. The 'Password:' field contains four dots ('....'). At the bottom are two buttons: a blue 'Login' button on the left and a link 'New User?' on the right.

## Welcome Page:



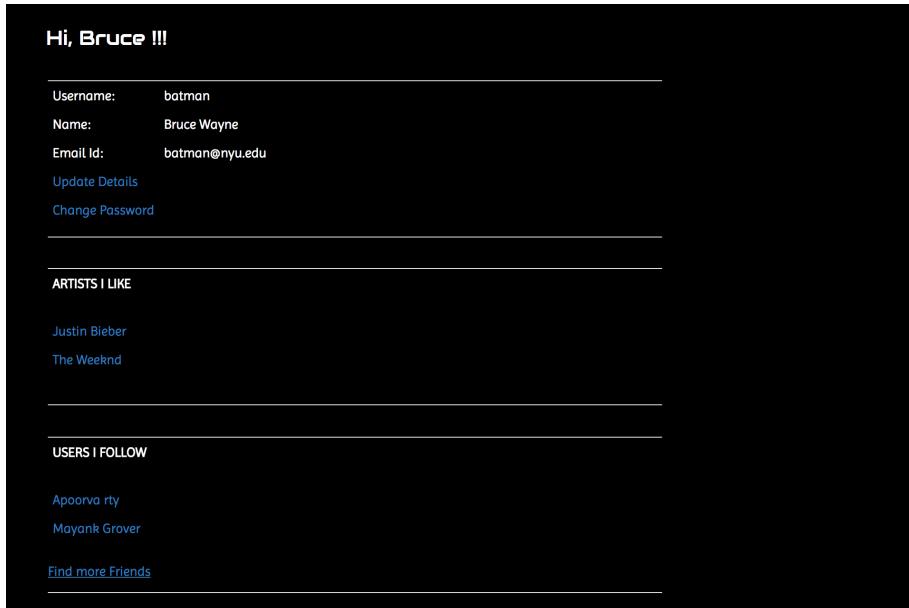
## Navigation Bar:

Once the user has logged in he can access the Navigation Bar on the top left corner of the screen. Buttons such as My Profile, Browse and Logout is available for the user to navigate the site.



## My Profile:

On clicking the My Profile button the user can view his account details and also can update his details and change his password. The user can also see the list of Artist he likes and the users he follows. Also, the user search for more users using find more friend option.



## Update Details

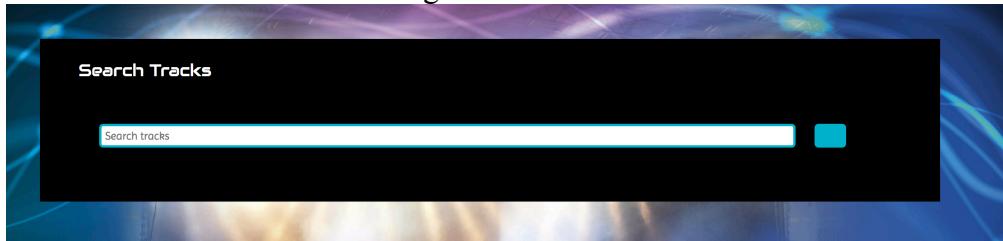
First Name :

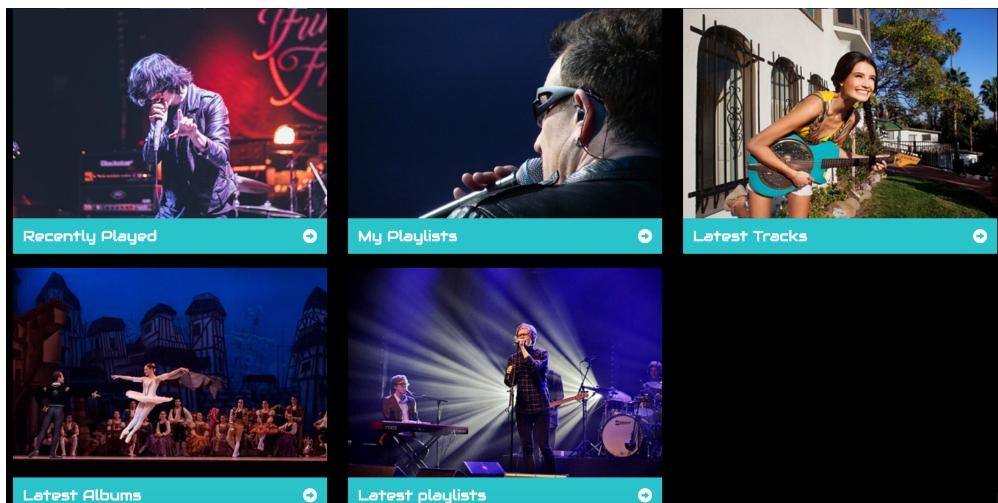
Last Name :

Email ID :

## Browse:

On clicking the Browse button the user is redirected to the search page where he can search for other users, artist, tracks etc. and also shows the recently played tracks, playlists, and some other track recommendations according to the user likes and follows.





### Most Played:

This link is displayed on the welcome page. On clicking this link a page appears where a user can see most played tracks by all the users.

**Most Played Tracks**

I Miss You - Adele	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
False Alarm - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
The Hills - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Key to the Highway - Live in San Diego - Eric Clapton	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Miss You So Much - Miley Cyrus	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Sorry - Justin Bieber	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Starboy - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Shameless - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
24K Magic - Bruno Mars	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Can't Feel My Face - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>

### Best Rated:

This link is displayed on the welcome page. On clicking this link a page appears where a user can see all the tracks which have the highest rating given by the other users.



**Best Rated Tracks**

I Miss You - Adele	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Can't Feel My Face - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Key to the Highway - Live in San Diego - Eric Clapton	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Starboy - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
The Hills - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Sorry - Justin Bieber	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Kisi Roz Tumse - Kumar Sanu	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Shameless - The Weeknd	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
24K Magic - Bruno Mars	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>
Miss You So Much - Miley Cyrus	<a href="#">Play</a>	<a href="#">Add to a Playlist</a>

### Top Artist:

This link is also displayed on the welcome page. On clicking this link a page appears where a user can view all the artist which are most liked by the other users.



**Top Artists**

Justin Bieber
The Weeknd

### My Playlists:

The user can create/view his Playlist. He can also add or remove a track from his playlist.

**My Playlists**

My favs

### Add to Playlist

Select Playlist:

Batman's Playlist [Add](#)

My favs [Add](#)

Create New Playlist

Playlist Name :

Private  
 Public

[Create](#)

### Users I follow:

On clicking this link, a user can view other users he is following.

Apoorva rty Following

---

Username: apu  
Email Id: ap11445@nyu.edu

---

PLAYLISTS

[My favs](#)

---

LIKED ARTISTS

Justin Bieber  
The Weeknd

---

FOLLOWED USERS

Mayank Grover  
Bruce Wayne

## Play, Rate & add a Track to playlist:

User can play a track, rate it as well as add it to a playlist. The track page has the YouTube video embedded so that the user can listen to it.

The screenshot shows a dark-themed track details page for 'Party Monster' by The Weeknd. At the top, there's a 'Track Details' header and a title 'Party Monster'. Below the title are buttons for 'Add to a Playlist' and 'Rating'. A five-star rating is shown with four stars highlighted in yellow. Below the rating is a thumbnail for the track's video, which features a red car and the text 'THE WEEKND party monster DIRECTED BY BRTHR vevo'. Underneath the video thumbnail, there's a section titled 'RELATED TRACKS' with five tracks listed: 'False Alarm - The Weeknd', 'The Hills - The Weeknd', 'Reminder - The Weeknd', 'Often - The Weeknd', and 'Shameless - The Weeknd'. Each track entry includes a 'Play' button and an 'Add to a Playlist' button.

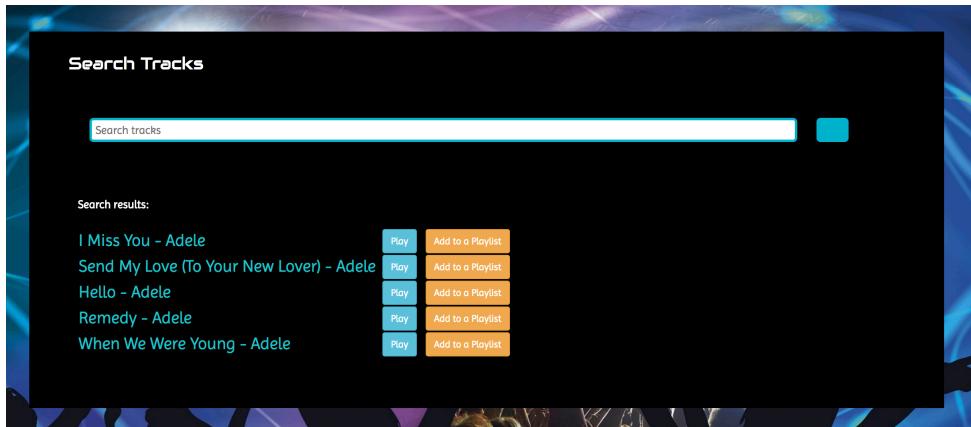
## Search other users:

A user can search for other users view their profile and follow them. The user profile shows the details of the user, the artists they like, their public playlists and also who they are following.

The screenshot shows a dark-themed 'Search and Follow Friends' page. At the top, there's a header 'Search and Follow Friends' and a search bar with the placeholder 'Search users'. Below the search bar, there's a section titled 'Search results:' with a single result: 'Apoorva rty'. There's a blue button at the bottom right of the search results section.

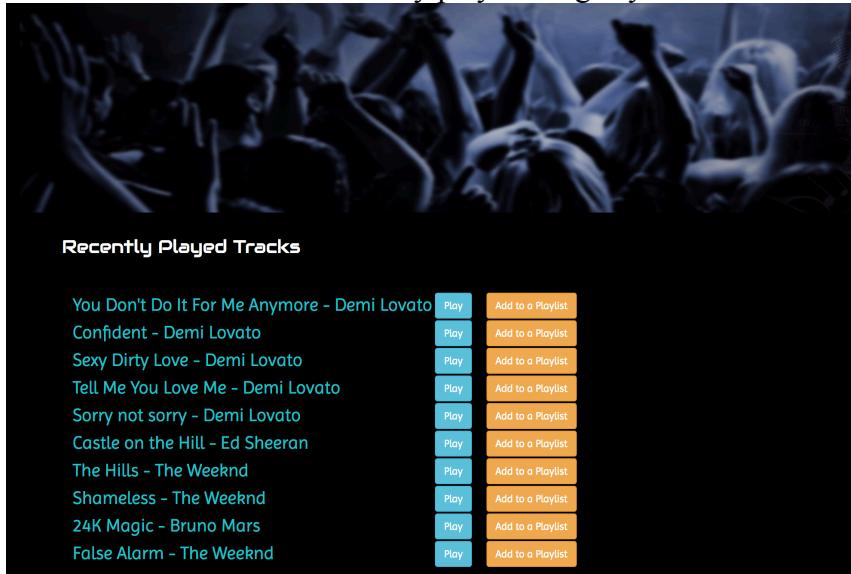
## Search tracks:

User can search for tracks based on the artist name, track name or the genre.



## Recently Played:

Users can view the most recently played songs by him.



## Latest Tracks by artists they like:

Here, user can see list of latest tracks by the artists he has liked.



**Latest Tracks by Liked Artists**

False Alarm - The Weeknd	Play	Add to a Playlist
Party Monster - The Weeknd	Play	Add to a Playlist
Secrets - The Weeknd	Play	Add to a Playlist
Reminder - The Weeknd	Play	Add to a Playlist
Sorry - Justin Bieber	Play	Add to a Playlist
Often - The Weeknd	Play	Add to a Playlist
Mark My Words - Justin Bieber	Play	Add to a Playlist
Love Yourself - Justin Bieber	Play	Add to a Playlist
Starboy - The Weeknd	Play	Add to a Playlist
What do you mean - Justin Bieber	Play	Add to a Playlist

### Latest Albums by Artist they like:

Here, user can see list of latest albums by the artists he has liked.



**Latest Albums by Liked Artists**

Legend of the Fall
Purpose (Deluxe)
Beauty Behind The Madness

### Latest Playlist by followed users:

Here, user can see list of latest playlist by the users he is following.



**Latest Playlists by Followed Users**

My favs
Batman's Playlist
maya's open playlist

## Album Details:

The user can navigate to the album details page which displays album details, tracks in the album as well as other related albums.

The screenshot shows the 'Album Details' page for 'Legend of the Fall' by The Weeknd. At the top, it displays the album title 'Legend of the Fall'. Below that is the release date '2017-12-05 00:00:00'. A horizontal line separates this from the 'LIST OF TRACKS' section. This section contains four tracks by The Weeknd, each with a 'Play' button and an 'Add to a Playlist' button. The tracks are: 'Party Monster', 'False Alarm', 'Reminder', and 'Secrets'. Another horizontal line separates this from the 'RELATED ALBUMS' section, which lists 'Beauty Behind The Madness'.

## Artist Details:

The user can navigate to the artist details page which displays artist details, list of albums by the artist as well as other related artists.

The screenshot shows the 'Artist Details' page for The Weeknd. At the top, it displays the artist name 'The Weeknd' with a 'Liked' button next to it. Below that is a 'Description' field containing 'canadian pop pop rap'. A horizontal line separates this from the 'LIST OF ALBUMS' section, which lists 'Legend of the Fall' and 'Beauty Behind The Madness'. Another horizontal line separates this from the 'RELATED ARTISTS' section, which lists various artists including Bruno Mars, A.R. Rahman, Justin Bieber, Jennifer Lopez, Drake, Adele, Kumar Sanu, Miley Cyrus, Ed Sheeran, and Katy Perry.

### 3. Bibliography

- Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan. *Database system concepts*. Vol. 4. New York: McGraw-Hill, 1997.
- MYSQL Documentation - mysql.org/docs/
- W3Schools - w3schools.com/sql/
- Tutorialspoint - tutorialspoint.com/sql/
- Stack Overflow - stackoverflow.com