



Bilkent University

Department of Computer Engineering

---

# CS353 Course Project

TunedIn

## Project Final Report

Furkan Kazım Akkurt, Cemal Arda Kızılkaya, Khasmamad Shabanovi, Mehmet Bora Kurucu

Instructor: Özgür Ulusoy  
Teaching Assistant: Arif Usta

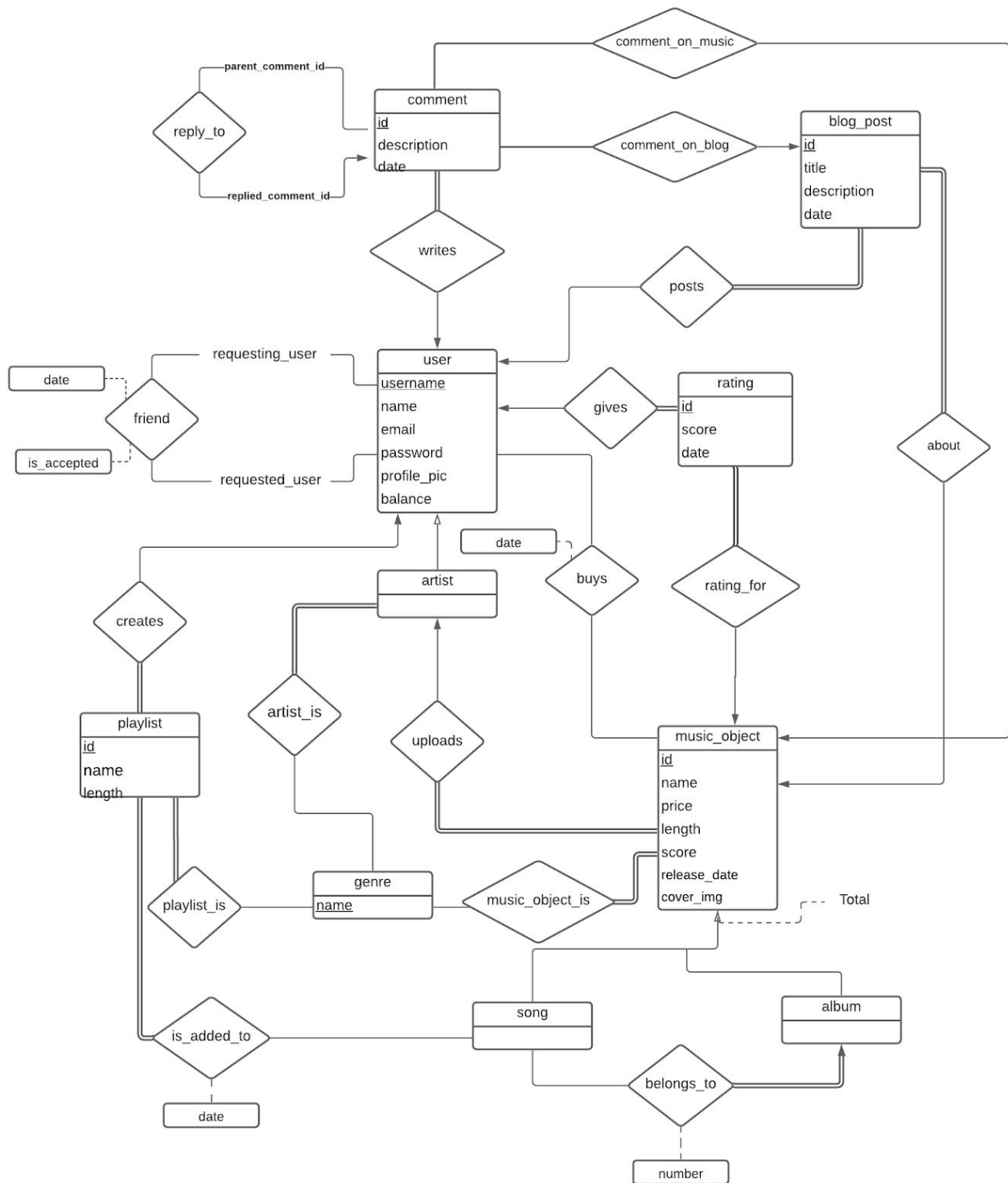
January 4, 2021

<b>1. Overview</b>	<b>3</b>
<b>2. Final E/R</b>	<b>4</b>
<b>3. Table Schemas</b>	<b>5</b>
<b>4. Implementation</b>	<b>6</b>
4.1. Tools	6
4.2. Challenges	6
4.3. Implementation of Features	7
<b>5. Advanced Database Components</b>	<b>7</b>
5.1. Reports	7
5.2. Views	8
5.3. Triggers	9
5.4. Constraints	9
5.5. Secondary Indices	9
<b>6. User's Manual</b>	<b>10</b>
6.1. Login	10
6.2. Sign up	10
6.3. Welcome Page	11
6.4. Search	11
6.5. Songs	12
6.5.1. Songs list	12
6.5.2 Song profile	12
6.5.2.1 Song profile (before purchasing)	12
6.6. Album profile	13
6.7. Friends	13
6.7.1 Friends list	13
6.7.2 Friend profile	14
6.8. Playlists	14
6.8.1. List of playlists	14
6.8.2. Create a playlist	15
6.9. Artist homepage	15
6.10. Upload content	16
6.10.1. Upload album	16
6.10.2. Upload song	16

# 1. Overview

TunedIn is a music data management application. There are two types of users in the application: artists and standard users. Artists can upload new songs and albums to the system and specify the genre and price for the music products they are uploading. Additionally, the artists can gain insight about the popularity of their music files by checking how many times a music product is purchased or how many times a music product appears in different users' playlists. On the other hand, the standard users can purchase albums and songs uploaded by artists and create playlists with the songs they have purchased. Also, the users can associate their playlists with different genres to categorize them. The users can rate the music products they have purchased and give feedback to them. Also, the standard users can search for other users and send friend requests to them. If accepted, they can see their friends' activities, such as recent purchases and feedback, in their feed. Also, the users can manage the friend requests they have received from the Social tab.

## 2. Final E/R



### 3. Table Schemas

User(username, name, email, password, profile\_pic, balance)

Artist(artist\_username)

- foreign key artist\_username references User(username)

MusicObject(music\_object\_id, name, price, length, score, release\_date, cover\_img, artist\_username)

- foreign key artist\_username references Artist(artist\_username)

Album(album\_id)

- foreign key album\_id references MusicObject(music\_object\_id)

Song(song\_id, album\_id, number)

- foreign key song\_id references MusicObject(music\_object\_id)
- foreign key album\_id references Album(album\_id)

Playlist(playlist\_id, name, length, username)

- foreign key username references User(username)

Rating(rating\_id, score, date, username, music\_object\_id)

- foreign key username references User(username)
- foreign key music\_object\_id references MusicObject(music\_object\_id)

BlogPost(blog\_post\_id, title, description, date, username, music\_object\_id)

- foreign key username references User(username)
- foreign key music\_object\_id references MusicObject(music\_object\_id)

Genre(name)

Comment(comment\_id, description, date, username, blog\_post\_id, music\_object\_id)

- foreign key blog\_post\_id references BlogPost(blog\_post\_id)
- foreign key music\_object\_id references MusicObject(music\_object\_id)
- foreign key username references User(username)

ReplyTo(reply\_comment\_id, parent\_comment\_id)

- foreign key parent\_comment\_id references Comment(comment\_id)
- foreign key reply\_comment\_id references Comment(comment\_id)

Friends(requesting\_username, requested\_username, request\_date, is\_accepted)

- foreign key requesting\_username references User(username)
- foreign key requested\_username references User(username)

MusicObjectGenre(music\_object\_id, genre\_name)

- foreign key music\_object\_id references MusicObject(music\_object\_id)
- foreign key genre\_name references Genre(genre\_name)

ArtistGenre(artist\_username, genre\_name)

- foreign key artist\_username references Artist(artist\_username)
- foreign key genre\_name references Genre(genre\_name)

Buys(username, music\_object\_id, date)

- foreign key username references User(username)
- foreign key music\_object\_id references MusicObject(music\_object\_id)

PlaylistSongs(playlist\_id, song\_id, date)

- foreign key playlist\_id references Playlist(playlist\_id)
- foreign key song\_id references Song(song\_id)

PlaylistGenre(playlist\_id, genre\_name)

- foreign key playlist\_id references Playlist(playlist\_id)
- foreign key genre\_name references Genre(genre\_name)

## 4. Implementation

### 4.1. Tools

We used PHP and MySQL for the back-end of our application. On the other hand, we used HTML, CSS, JavaScript, and Bootstrap for the front-end of the application. We tested our implementations on a local database created using XAMPP.

### 4.2. Challenges

None of the team members received a formal education of the above-mentioned tools prior to the implementation of the project. Not knowing the good practices, we might have had to reinvent the wheel.

One of the problems we faced concerned passing data between files (same or different) through HTML forms. While it's relatively straightforward to carry the value of one variable to another file, the process is trickier when you have multiple variables and their values to take care of. To achieve that we used two different methods. First is using the "hidden" input type. This input type can be given default values that can be read on the file the form is referring to. Second option is sending a GET form through links. For example, the title of a song/album corresponds to a link which redirects the user to the song or album profile page. The link contains the id of the song/album in the following form: ".../?song\_id=xx" (or album\_id). Multiple values can be added to the link in a similar fashion by combining them with "&". This approach also allows the user to navigate between pages easily, as sending a GET form creates a new link for each visited page, which can be revisited by the user again.

## 4.3. Implementation of Features

Furkan Kazım Akkurt:

- Creating playlists
- Adding songs to a playlist
- Removing songs from a playlist
- Categorizing playlists by genre
- Filtering/sorting purchased songs

Cemal Arda Kızılkaya:

- Login, signup, and logout for both user types
- Searching for users, artists, and music files
- Sending friend requests to other users
- Viewing current friends and managing friend requests

Khasmamad Shabanovi:

- Music file profile pages
- Purchasing a music file
- Rating a music file
- Commenting on a music file
- Replying to comments on music files

Mehmet Bora Kurucu:

- Uploading music files as an artist, setting prices and genres for them
- Artist home page, including sorting and filtering on songs

## 5. Advanced Database Components

### 5.1. Reports

- Most purchased songs of an artist that are purchased at least twice:  

```
SELECT name, COUNT(*) as noOfPurchases
FROM Buys NATURAL JOIN MusicObject
      INNER JOIN Song on song_id = music_object_id
WHERE MusicObject.artist_username = @artist_username and
GROUP BY music_object_id
HAVING noOfPurchases > 1
ORDER BY noOfPurchases DESC;
```

Example output for the artist named “nakedTurtle”:

name	noOfPurchases
Green Turtle	4
I am naked	2

- Songs that appear most frequently in a user's playlists:  
 SELECT mo.name as name, COUNT(\*) as count  
 FROM Playlist NATURAL JOIN PlaylistSong  
                   INNER JOIN MusicObject mo ON music\_object\_id = song\_id  
 WHERE username = @username  
 GROUP BY song\_id  
 ORDER BY count DESC;

Example output:

name	count
Green Turtle	4
Rainbow Turtle	4
Get faster	4
Black Fox	3
I am naked	2
Violet Fox	2
Green Fox	1
Orange Fox	1
Angry fox	1

## 5.2. Views

- Total user expenditures in the last 12 months:  
 CREATE VIEW expenditures AS  
 SELECT username, SUM(price) as totalExp  
 FROM Buys NATURAL JOIN MusicObject  
 WHERE date > NOW() - INTERVAL 12 month  
 GROUP BY username  
 ORDER BY totalExp DESC;

Example output:

username	totalExp
Khasmamad	48.88
Kurucu	27.96
purpleFox	17.97
angryPuppy	9.97



- Favorite genres of a user based on the number of purchased songs of each genre:

```
CREATE VIEW favGenres AS
SELECT genre_name, COUNT(*) as count
FROM Buys NATURAL JOIN MusicObject
      NATURAL JOIN MusicObjectGenre
      INNER JOIN Song ON music_object_id = song_id
WHERE username = "khasmamad"
GROUP BY genre_name
ORDER BY count DESC;
```

Example output:

genre_name	count
Pop	5
Hip-hop	4
Indie	1

### 5.3. Triggers

- When a user purchases a music file, the user's account balance will be updated accordingly.
- When a user purchases an album, the user will also purchase all the songs inside that album.
- When a comment is deleted, the replies to that comment will be deleted as well.

### 5.4. Constraints

- A user cannot purchase a music file if the balance on the account is not sufficient.
- A user cannot rate or comment on an album if it is not purchased by the user.
- A user cannot see the activities, such as ratings, comments, purchases, of another user if they are not friends.
- A user can create a playlist with only the songs that s/he has purchased.

### 5.5. Secondary Indices

- 'date' as a secondary index in the tables which has a 'date' column use (for sorting);
- 'album\_id' as a secondary index in the table 'Song';

## 6. User's Manual

### 6.1. Login

The screenshot below shows the login page where regular users and artists login to application by inputting the correct values for the fields required. An error message is shown to the user in case of any invalid or wrong value. After making sure that the correct values are entered, users can login to the system by clicking on the Log In button below the input fields. This will redirect users to the welcome page of the application, where they see their feed, and artists to a page where they can see their music products that they uploaded to the system. If the visitor of the page does not have an account, one can create an account by clicking on the Sign Up link below the Log In button and visiting the Sign Up page.

TunedIn

TunedIn

Email address or username

Password

Log In

Sign Up Here!

©2020

### 6.2. Sign up

This page is for the users that do not own an account or want to create another one. By entering the values into corresponding input fields, one can easily create a new account if the values entered are valid. If the user is an artist, s/he must select the I'm an artist option from the selection box on the right hand side of the form. After all, clicking on the Sign Up button will create a new account, which later can be used to Log In to the system.

TunedIn

TunedIn

Full name

Full name

Username

Username

Email

Email

I'm a/an

Choose...

Password

Password

Repeat password

Repeat password

Sign Up

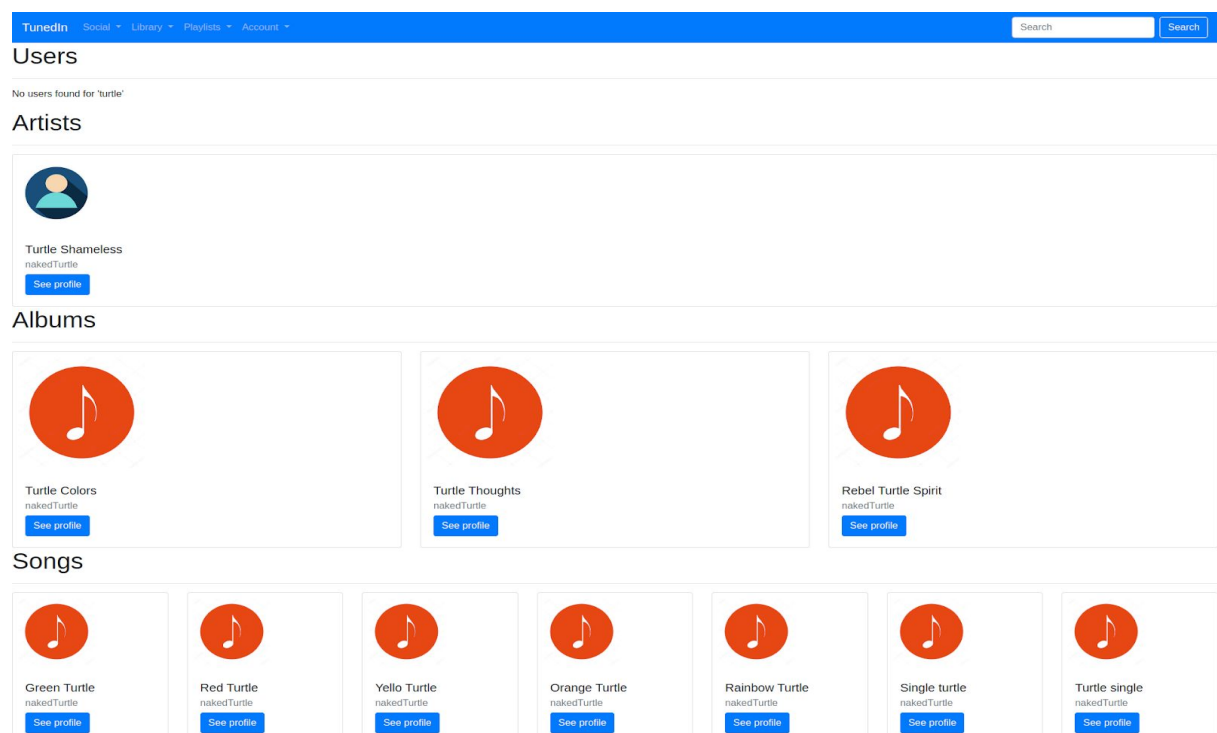
## 6.3. Welcome Page

The screenshot below shows the welcome page the regular users are redirected to after they login to the system. In this page they can see the recently bought albums and songs. From the navigation bar above, they can redirect into other pages such as song library, profile, and etc. Also, they are able to search for users, artists, songs, or albums using the search field on the right side of the navigation bar.



## 6.4. Search

After the search operation is done using the search field in the header, users are redirected to a page where they can see related results to their keyword that they have entered into the input field. If the corresponding query cannot find any results in the database, an appropriate message is shown, as it can be seen from the screenshot below, where there is not a user in the database containing the search words. If some results returned from the corresponding query, they are presented to the user in the page.



## 6.5. Songs

### 6.5.1. Songs list

The screenshot below shows the page where the regular users can see the songs that they have bought so far. Clicking on the name of a song will redirect the user to this particular song's profile, where s/he can see the details regarding the song. Additionally, users can search for songs or filter them by using the below search, filter and selection boxes. Users do not have to fill all the input fields, they can use any search combination as they wish.

TunedIn

SocialLibraryPlaylistsAccount

Search

Search

Songs

Name	Artist	Length	Purchase Date
Green Turtle	nakedTurtle	420	2020-12-28 12:04:41
Rainbow Turtle	nakedTurtle	420	2020-12-28 12:04:41
Get faster	nakedTurtle	500	2020-12-29 10:07:40
I am naked	nakedTurtle	600	2020-12-29 10:07:35
Orange Fox	purpleFox	480	2020-12-29 10:08:03
Green Fox	purpleFox	480	2020-12-29 10:08:25
Violet Fox	purpleFox	480	2020-12-29 12:28:12
Pink Fox	purpleFox	480	2020-12-29 12:29:42
Black Fox	purpleFox	480	2020-12-29 12:30:06
Angry fox	purpleFox	640	2020-12-29 10:08:35

SORT

Select one

Filter by purchase date (enter your input in the following format: YYYY-MM-DD)

Minimum date

Maximum date

Search & Filter

### 6.5.2 Song profile

#### 6.5.2.1 Song profile (before purchasing)

The screenshot below shows the page where the detailed information about a song can be found. Song information, such as its name, rating, release date can be found on the left side of the screen below the cover image. Button that displays the price of the song below those information can be used to purchase the song. If the user purchases the song or it is already purchased by the user, the purchase button becomes disabled. Comments on the songs are displayed below the page and the user is able to write a comment if s/he has already bought the song. On the right side of the screen, similar songs are displayed considering their genres, and users can click on the names to view their pages.


TunedIn

SocialLibraryPlaylistsAccount

Search

Search

Freedom song by nakedTurtle



Genre: Hip-hop

Rating: 8.2

Length: 10:0

Release date: 2020-03-03

2.99\$

Similar to this:

Name	Artist
Rainbow Turtle	nakedTurtle
I am naked	nakedTurtle
Green Turtle	nakedTurtle
Get faster	nakedTurtle

Comments:

No comments yet

## 6.6. Album profile

Similar to the song profile page, this page shows the information about the album. Right hand side of the screen also contains a part where the songs in the album are displayed. In the figure one can see that the album is already bought, thus, the button is disabled in order to not let users buy the album again, and the comment option is available in case if user wants to express their opinions on the album.

TunedIn

Social

Library


Playlists

Account

Search

Search

Turtle Colors Album by nakedTurtle



Genre: Rap

Rating: 8.2

Your rating: 6

Length: 35:0

Release date: 2020-01-01

Purchased

Comments:

comment

No comments yet

Songs in this album:

Name	Len	Price
<a href="#">Green Turtle</a>	7:0	owned
<a href="#">Red Turtle</a>	7:0	owned
<a href="#">Yellow Turtle</a>	7:0	owned
<a href="#">Orange Turtle</a>	7:0	owned
<a href="#">Rainbow Turtle</a>	7:0	owned

Similar to this:

Name	Artist
<a href="#">Rebel Turtle Spirit</a>	<a href="#">nakedTurtle</a>

## 6.7. Friends

### 6.7.1 Friends list

The screenshot below shows the friends that the user has so far and the friend requests that s/he has received. See Profile button below the profile of a user redirects the current user to the profile page of friend. If any friend requests are pending, two options, Accept and Reject, as well as See Profile option are presented to the user.

TunedIn

Social

Library


Playlists

Account

Search

Search

Friends



Bora

Kurucu

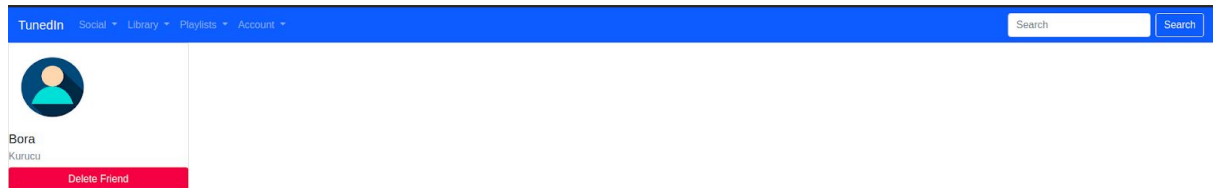
See profile

Friend Requests

You don't have any friend requests

## 6.7.2 Friend profile

The screenshot below shows the profile of a friend. This page is currently empty, however, further implementations are thought to contain this page as a page where detailed information can be found regarding the user. Users can delete the friend if they wish to, using the Delete Friend button.



## 6.8. Playlists

### 6.8.1. List of playlists

The screenshot below shows the playlists of the user. For each genre that the user has a playlist, the genre and the names of the playlists of that genre is displayed.



### 6.8.2. Create a playlist

The screenshot below shows the screen of creating a playlist. User selects the songs s/he wants to put into the playlist, by either selecting them directly or searching from the search bar. After selecting the songs, user enters the name of the playlist, then s/he selects at least one genre from the genre list. When all the previous steps are completed, user can create the playlist by hitting the play button.

TunedIn

SocialLibraryPlaylistsAccount

Search

Search

Name of the Playlist

Enter the name of the playlist

Select to add	Name	Artist	Length
<input type="checkbox"/>	Green Turtle	nakedTurtle	420 seconds
<input type="checkbox"/>	Rainbow Turtle	nakedTurtle	420 seconds
<input type="checkbox"/>	Get faster	nakedTurtle	500 seconds
<input type="checkbox"/>	I am naked	nakedTurtle	600 seconds
<input type="checkbox"/>	Orange Fox	purpleFox	480 seconds
<input type="checkbox"/>	Green Fox	purpleFox	480 seconds
<input type="checkbox"/>	Violet Fox	purpleFox	480 seconds
<input type="checkbox"/>	Pink Fox	purpleFox	480 seconds
<input type="checkbox"/>	Black Fox	purpleFox	480 seconds
<input type="checkbox"/>	Angry fox	purpleFox	640 seconds

You need to select at least one genre from below

☐ Classical

☐ Country

☐ Electronic Dance

☐ Hip-hop

☐ Indie

☐ Jazz

☐ K-pop

☐ Metal

☐ Oldies

☐ Pop

☐ Rap

☐ Rhythm & Blues

☐ Rock

☐ Techno

Create

### 6.9. Artist homepage

In the artist homepage, all the songs that the artist created are displayed with their ids, names, prices, lengths, genres, score, release dates and covers. Artists can see at most ten songs per page, and can see the remaining songs by changing the page from the bottom of the page.

TunedIn

My MusicContentAccount

My Songs

Search

ID	Name	Price	Length	Genre	Score	Release Date	Cover
36	First Song	2.00	300	-	10.0	2018-05-05	
37	bigdata	3.00	320	-	8.0	2020-08-09	
38	Khasmamad Shabanovi	4.00	200	-	5.0	1999-04-04	
39	bigdata	3.00	320	-	8.0	1999-04-05	
40	song 1	8.00	320	Indie	8.0	1999-04-04	
41	Song 1	1.00	320	Hip-hop	10.0	1999-02-05	
42	Song 2	2.10	310	Hip-hop	10.0	1999-12-09	
43	Song 3	3.00	320	Electronic Dance	10.0	1999-02-05	
44	Song 4	2.00	200	Hip-hop	7.0	1999-04-04	
45	new song	2.00	220	Jazz	10.0	1999-04-04	

Showing 1 to 10 of 11 rows

10 rows per page

1

2

## 6.10. Upload content

### 6.10.1. Upload album

To upload an album, the artists first fill the attributes of the songs that s/he wants to put into the album, uploads them one by one by pressing the upload song button. After finishing uploading all the songs, the name and the cover image of the album is entered, and officially created by pressing finalize album button.

TunedIn

My Music

Content

Account

Album details

Name

Cover Image

Album name

Image link

Finalize Album

Upload a song to this album

Name

Song name

Price

Length (seconds)

\$

Length (seconds)

Genre

Score

Choose...

Score

Release Date

Link to cover image

mm/dd/yyyy

Link to cover image

Upload Song

Songs added to album

Name	Price	Length	Genre	Score	Release Date	Cover
No matching records found						

### 6.10.2. Upload song

To upload a song, the artist writes the name, price, length, genre, score, release date and the link of the song. After filling all those required information, the artist finished the upload by pressing the upload song button.

TunedIn

My Music

Content

Account

Upload a song

Name

Song name

Price

Length (seconds)

\$

Length (seconds)

Genre

Score

Choose...

Score

Release Date

Link to cover image

mm/dd/yyyy

Link to cover image

Upload Song



## 7. Website

The report is available at <https://kizilkayaarda.github.io/CS353/>.