



SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING

Review -2

**Project Report
On
Spotify Clone**

**Submitted For Course: Open Source Programming
(ITE-1008)**

Submitted By:

AKASH KUMAR(17BIT0055)

Slot: B2+TB2

**Under Guidance Of :
Prof. Rath R.**

Abstract:

The project, it contains an admin side and a user where a user can view available albums and play the music easily. Additional, the admin plays an important role in the management of this system. In this project, the user has to perform all the main functions from the admin side. the user can simply view all the new releases, top 6 songs with rankings, news, featured songs, vote their favorite song and listen to the song. The users can simply view all the albums and click any one of them to listen their songs. From the admin panel, the admin has full access to the system. He/she can manage all the music albums. In order to add albums, the admin has to select category, name, singer, writer, description and cover photo. The admin can also add album categories. Besides, he/she can simply add songs to the existing album records. It also contains **user** management system for the easy management of the system.

Features:

1. Admin panel
2. Featured songs
3. New release
4. Play music
5. Poll system
6. Manage album
7. Manage categories
8. User management system

Design and Markup:

- HTML5, CSS3, JavaScript & jQuery for markup, design and interaction behavior
- Bootstrap / Angular JS will be used for making the site responsive and as a framework.

Server side language (anyone can be used depending upon your preference)

- PHP (codeigniter, CakePHP, Phalcon or Laravel) OR .Net with framework

Database Software (anyone can be used depending upon your preference and server setup)

- MySQL

- SQL Server

Hardware Requirements:

Intel Core processor

RAM 4GB and Above

HDD 100 GB Hard Disk Space and Above

64 or 32 bit OS Windows 8 and above

CODES and INTERFACE:

1. php files

(i) User.php

```
<?php class User {  
  
    private $con;  
  
    private $username;  
  
        public function __construct($con, $username) {  
  
            $this->con = $con;  
  
            $this->username = $username;  
  
        }  
  
        public function getUsername() {  
  
return $this->username;  
  
        }  
  
        public function getEmail() {  
  
            $query = mysqli_query($this->con, "SELECT email FROM users  
WHERE username='$this->username'");  
            $row =  
mysqli_fetch_array($query);  
            return $row['email'];  
  
        }  
  
        public function getFirstAndLastName() {  
  
            $query = mysqli_query($this->con, "SELECT concat(firstName, ' ', lastName) as 'name'  
FROM users WHERE username='$this->username'");  
  
            $row =  
mysqli_fetch_array($query);  
            return  
$row['name'];  
  
        }  
  
    }  
?>
```

(ii) register.php

```
<?php
```

```

        include("includes/config.php");
include("includes/classes/Account.php");
include("includes/classes/Constants.php"); $account = new Account($con);
        include("includes/handlers/register- handler.php");
include("includes/handlers/login-handler.php"); function
getInputValue($name) {    if(isset($_POST[$name])) {
                echo
$_POST[$name];
            }
        }
?>
<html>
<head>
    <title>Welcome to Spotify Clone!</title>
    <link rel="stylesheet" type="text/css" href="assets/css/register.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
    <script src="assets/js/register.js"></script>
</head>
<body>  <?php
if(isset($_POST['registerButton'])) {  echo
'<script>
                $(document).ready(function() {
                    $("#loginForm").hide();
                    $("#registerForm").show();
                });
            </script>';
        }
    else

```

```

{
    echo '<script>

        $(document).ready(function() {

            $("#loginForm").show();

            $("#registerForm").hide();

        });

    </script>';
}

?>

<div id="background">

    <div id="loginContainer">

        <div id="inputContainer">

            <form id="loginForm" action="register.php" method="POST">

                <h2>Login to your account</h2>

                <p>

                    <?php echo $account->getError(Constants::$loginFailed); ?>    <label
for="loginUsername">Username</label>

                        <input id="loginUsername"
name="loginUsername" type="text" placeholder="e.g. bartSimpson" value="<?php
getInputValue('loginUsername') ?>" required autocomplete="off">

                    </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>

                            <div class="hasAccountText">

                                <span id="hideLogin">Don't have an account yet? Signup here.</span>

                            </div>

```

```
</form>

<form id="registerForm" action="register.php" method="POST">

    <h2>Create your free account</h2>

    <p>

        <?php
echo $account->getError(Constants::$usernameCharacters); ?>

        <?php
echo $account->getError(Constants::$usernameTaken); ?>

        <label for="username">Username</label>

        <input id="username" name="username" type="text" placeholder="e.g.
bartSimpson" value="<?php getInputValue('username') ?>" required>

    </p>

    <p>

        <?php
echo $account->getError(Constants::$firstNameCharacters); ?>

        <label for="firstName">First name</label>

        <input id="firstName" name="firstName" type="text" placeholder="e.g.
Bart" value="<?php getInputValue('firstName') ?>" required>

    </p>

    <p>

        <?php
echo $account->getError(Constants::$lastNameCharacters); ?>

        <label for="lastName">Last name</label>

        <input id="lastName" name="lastName" type="text" placeholder="e.g.
Simpson" value="<?php getInputValue('lastName') ?>" required>

    </p>

    <p>

        <?php
echo $account->getError(Constants::$emailsDoNotMatch); ?>

        <?php
```

```

echo $account->getError(Constants::$emailInvalid); ?>

        <?php echo $account->getError(Constants::$emailTaken); ?>

        <label for="email">Email</label>

                <input      id="email"      name="email"      type="email"
placeholder="e.g. bart@gmail.com" value="<?php getInputValue('email') ?>" required>

        </p>

        <p>

                <label for="email2">Confirm email</label>

                <input      id="email2" name="email2"      type="email" placeholder="e.g.
bart@gmail.com" value="<?php getInputValue('email2') ?>" required>

        </p>

        <p>

                <?php

echo $account->getError(Constants::$passwordsDoNoMatch); ?>

                <?php

echo $account->getError(Constants::$passwordNotAlphanumeric); ?>

                <?php

echo $account->getError(Constants::$passwordCharacters); ?>

                <label for="password">Password</label>

                <input id="password" name="password" type="password" placeholder="Your
password" required>

        </p>

        <p>

                <label for="password2">Confirm password</label>

                <input      id="password2"      name="password2"
type="password" placeholder="Your password" required>

        </p>

        <button      type="submit"      name="registerButton">SIGN
UP</button>

        <div class="hasAccountText">

                <span id="hideRegister">Already have an account? Log in

```


here.

</div>

</form>

</div>

<div id="loginText">

<h1>Get great music, right now</h1>

<h2>Listen to loads of songs for free</h2>

Discover music you'll fall in love with

Create your own playlists

Follow artists to keep up to date

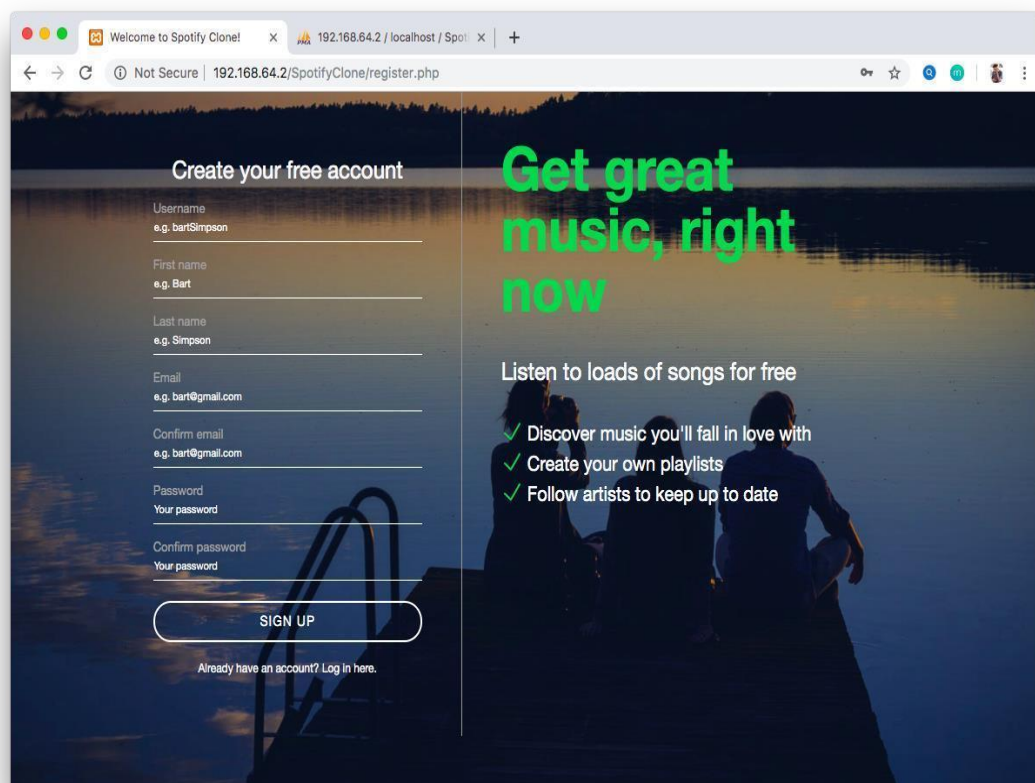
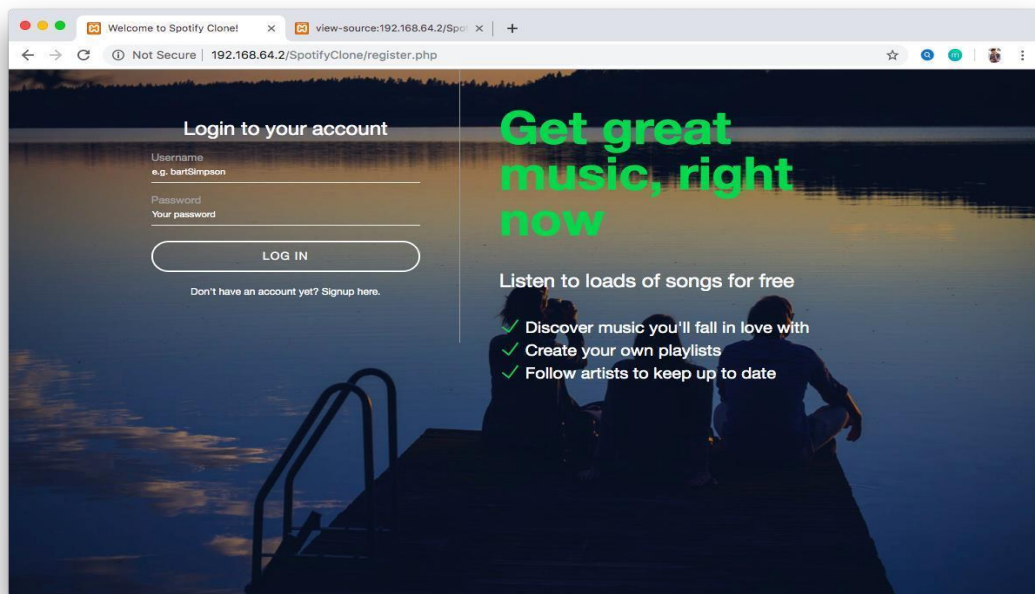
</div>

</div>

</div>

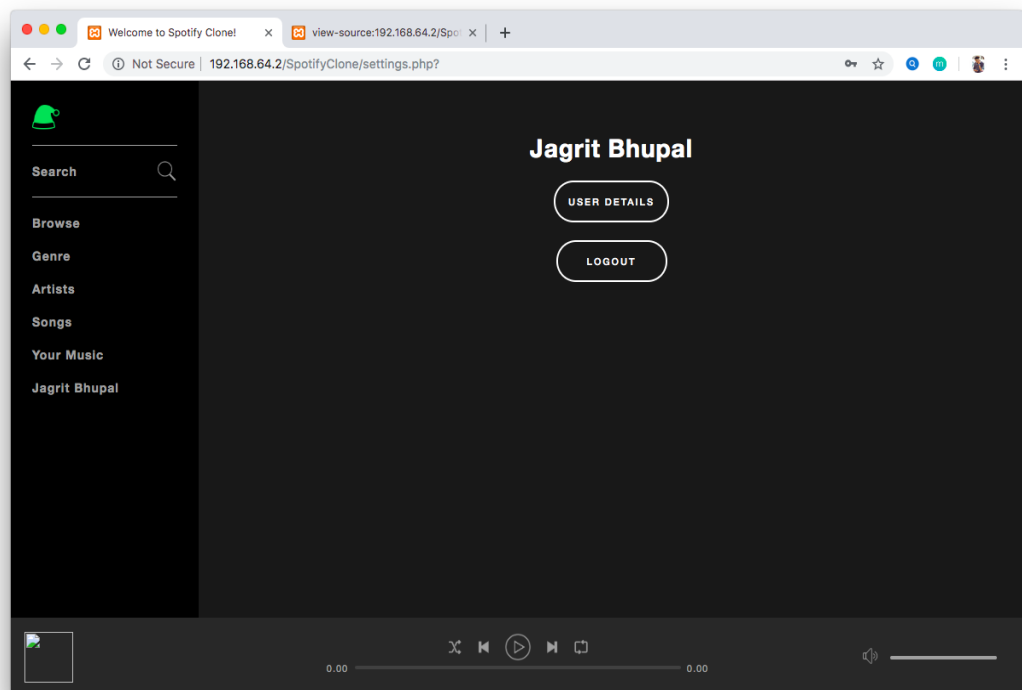
</body>

</html>



(iii) settings.php

```
<?php
include("includes/includedFiles.php");
?>
<div class="entityInfo">
    <div class="centerSection">
        <div class="userInfo">
            <h1><?php echo $userLoggedIn->getFirstAndLastName(); ?></h1>
        </div>
    </div>
    <div class="buttonItems">
        <button class="button" onclick="openPage('updateDetails.php')>USER
DETAILS</button>
        <button class="button" onclick="logout()">LOGOUT</button>
    </div>
</div>
```



(iv) updateDetails.php

```
<?php
include("includes/includedFiles.php");
?>

<div class="userDetails">

    <div class="container borderBottom">

        <h2>EMAIL</h2>

        <input type="text" class="email" name="email" placeholder="Email address..."
value="<?php echo $userLoggedIn->getEmail(); ?>">

        <span class="message"></span>

        <button class="button" onclick="updateEmail('email')">SAVE</button>

    </div>

    <div class="container">

        <h2>PASSWORD</h2>

        <input type="password" class="oldPassword" name="oldPassword"
placeholder="Current password">

        <input type="password" class="newPassword1" name="newPassword1"
placeholder="New password">

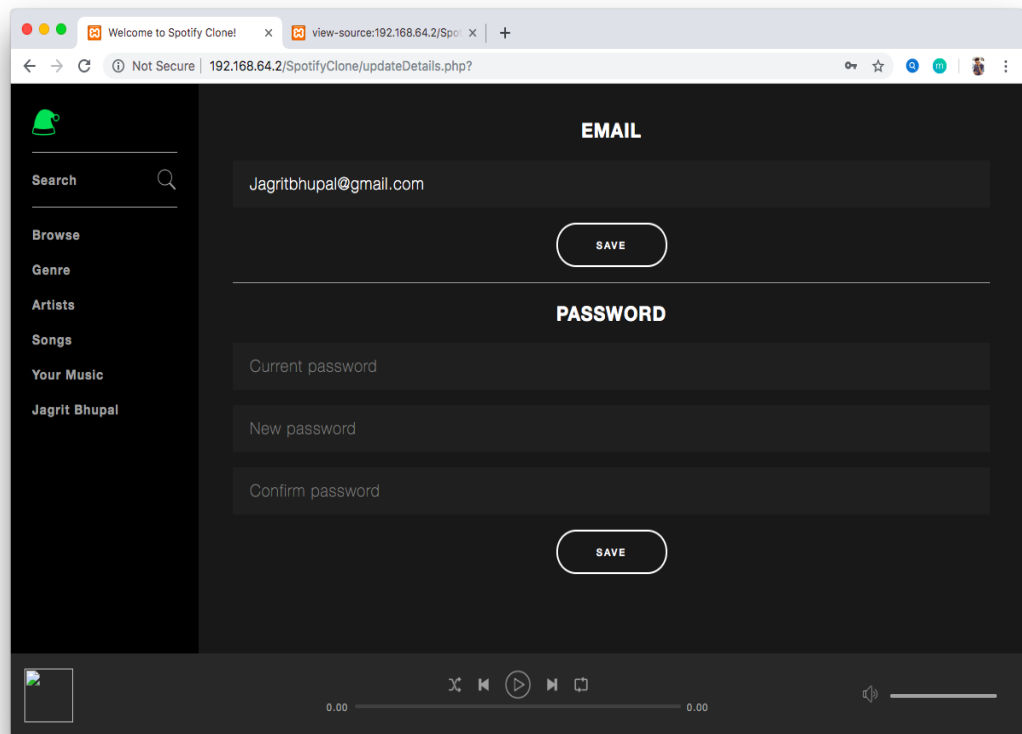
        <input type="password" class="newPassword2" name="newPassword2"
placeholder="Confirm password">

        <span class="message"></span>

        <button class="button" onclick="updatePassword('oldPassword',
'newPassword1', 'newPassword2')">SAVE</button>

    </div>

</div>
```



(v) yourMusic.php

```
<?php
```

```
include("includes/includedFiles.php");
```

```
?>
```

```
<div class="playlistsContainer">
```

```
    <div class="gridViewContainer">
```

```
        <h2>Playlists</h2>
```

```
        <div class="buttonItems">
```

```
            <button class="button green" onclick="createPlaylist()">NEW PLAYLIST</button>
```

```
        </div>
```

```
    <?php
```

```
        $username = $userLoggedIn->getUsername();
```

```
        $playlistsQuery = mysqli_query($con, "SELECT * FROM playlists  
WHERE owner='$username'");
```

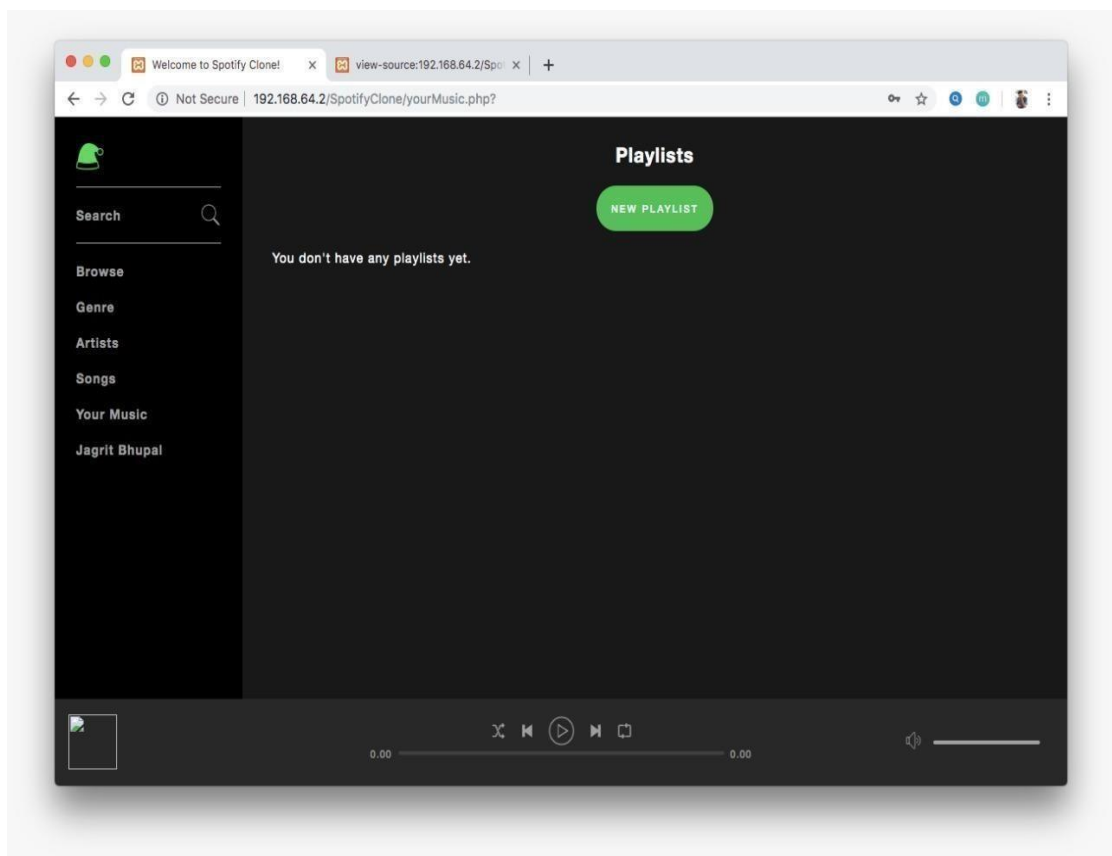
```
        if(mysqli_num_rows($playlistsQuery) == 0) {            echo "<span  
class='noResults'>You don't have any playlists yet.</span>";
```

```

    }

    while($row = mysqli_fetch_array($playlistsQuery)) {
        $playlist = new Playlist($con, $row);
        echo "<div class='gridViewItem' role='link' tabindex='0'
                onclick='openPage(\"playlist.php?id="
                .
                $playlist->getId() . "\")'>
                <div class='playlistImage'>
                    <img src='assets/images/icons/playlist.png'>
                </div>
                <div class='gridViewInfo'>\"
                    . $playlist-
                    >getName() .
                    \"</div>
                </div>";
    }
    ?>
</div>
</div>

```



(vi) song.php

```
<?php class Song {  
    private  
    $con;          private $id;  
    private  
    $mysqliData;   private  
    $title;        private  
    $artistId;     private  
    $albumId; private $genre;  private  
    $duration;  
    private $path;
```

```

    public function __construct($con, $id) {

        $this->con = $con;

        $this->id = $id;

        $query = mysqli_query($this->con, "SELECT * FROM Songs WHERE id='$this->id'");

        $this->mysqliData = mysqli_fetch_array($query);

        $this->title = $this->mysqliData['title'];

        $this->artistId = $this->mysqliData['artist'];

        $this->albumId = $this->mysqliData['album'];

        $this->genre = $this->mysqliData['genre'];

        $this->duration = $this->mysqliData['duration'];

        $this->path = $this->mysqliData['path'];

    }

    public function getTitle() {

        return $this->title;

    }

    public function getId() {
return $this->id;

    }

    public function getArtist() {

        return new Artist($this->con, $this->artistId);

    }

    public function getAlbum() {

        return new Album($this->con, $this->albumId);

    }

    public function getPath() {

        return $this->path;

    }

    public function getDuration() {

```



```

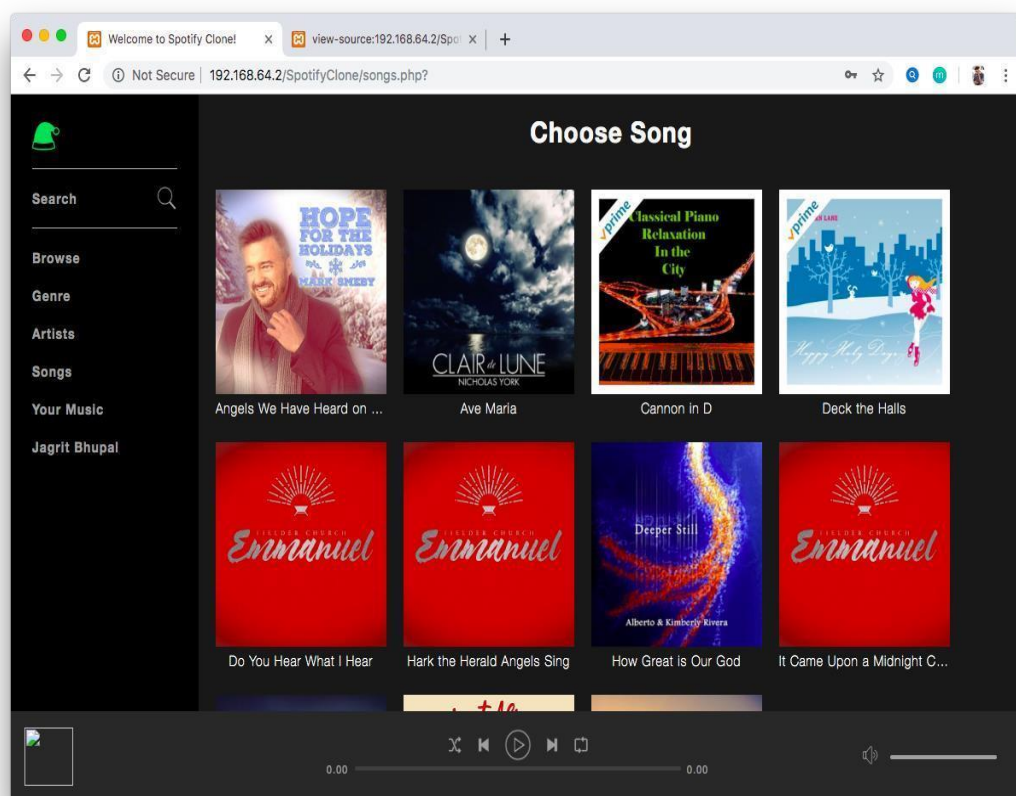
        return $this->duration;
    }

    public function getMysqliData() {
        return $this->mysqliData;
    }

    public function getGenre() {
        return $this->genre;
    }
}

?>

```



(vi) Artist.php

```

<?php class Artist {
    private $con;
    private $id;

```

```

        public function __construct($con, $id) {

            $this->con = $con;

            $this->id = $id;

        }

        public function getId() {

            return $this->id;

        }

        public function getName() {

            $artistQuery = mysqli_query($this->con, "SELECT name FROM artists WHERE
id='$this->id'");

            $artist =

mysqli_fetch_array($artistQuery);

            return

$artist['name'];

        }

        public function getSongIds() {

            $query = mysqli_query($this->con, "SELECT id FROM Songs WHERE
artist='$this->id' ORDER BY plays ASC");

            $array = array();

            while($row = mysqli_fetch_array($query)) {

                array_push($array, $row['id']);

            }

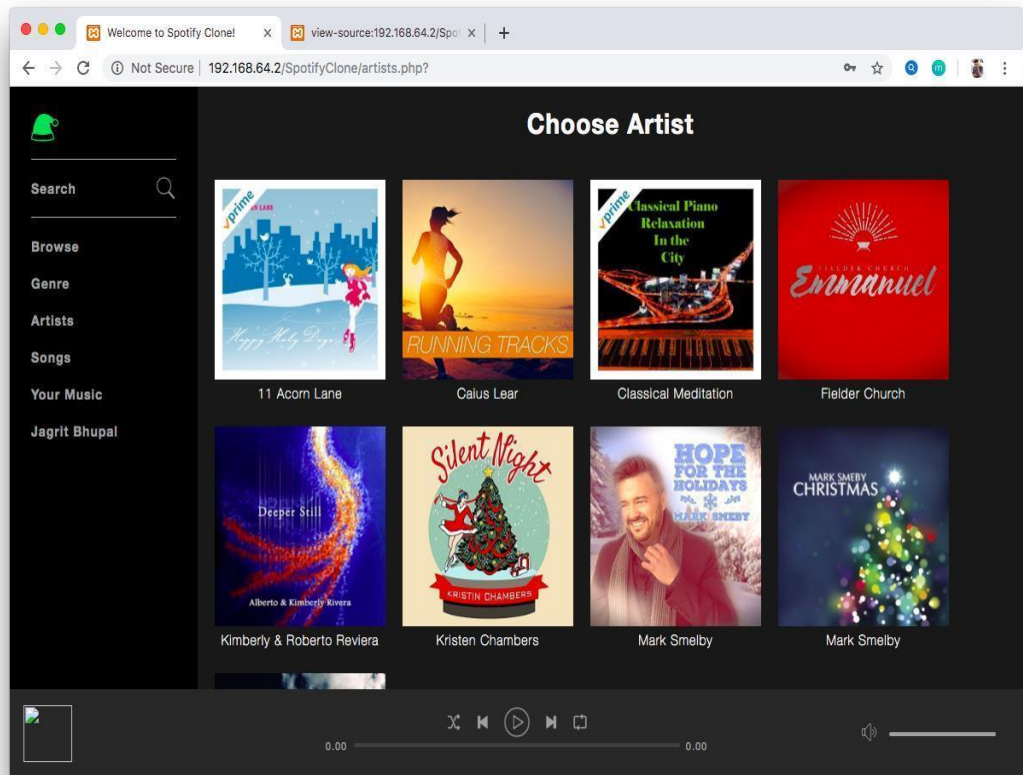
            return $array;

        }

    }

?>

```



2. css files

(i) register.css html,

```
body { padding: 0;
margin: 0; height:
100%;
} *:focus { outline:
none; }
#background { background-color:
#000; background-image: url(../images/bg.jpg); background-position:
center; background-size:
cover; display: table; height: 100%; width:
100%;
}
#loginContainer * { color:
#fff;
font-family: "Helvetica Neue", "Helvetica", "Arial", sans-serif; fontweight:
normal; line-height: 1em; box-sizing:
border-box;
}
#loginContainer { width:
80%; margin: 0 auto;
position: relative; max-
width: 1024px;
}
::-webkit-input-placeholder { /* WebKit, Blink, Edge */ color:
#fff;
}
:-moz-placeholder { /* Mozilla Firefox 4 to 18 */ color:
```

```
#fff; opacity: 1;
}
::-moz-placeholder { /* Mozilla Firefox 19+ */ color:
#fff; opacity: 1;
}
:-ms-input-placeholder { /* Internet Explorer 10-11 */ color:
#fff;
}
::-ms-input-placeholder { /* Microsoft Edge */
color: #fff; } #inputContainer { width:
400px; padding: 45px; float: left; border-
right: 1px solid
#999;
}
#inputContainer h2 { text-align:
center;
}
#inputContainer input[type="text"],
#inputContainer input[type="email"],
#inputContainer input[type="password"] {
display: block; background-color:
transparent; border: 0; borderbottom:
1px solid #fff; height: 27px; lineheight:
27px; width: 100%;
}
#inputContainer label { color:
#a0a0a0; font-size: 13px;
margin-top: 15px; display:
block;
```

```
}

#inputContainer button {
background-color: transparent;
border: 2px solid #fff;
borderradius: 250px;  color: #fff;
display: block;  font-size: 14px;
letter-spacing: 1px;  margin: 20px
auto;  height: 41px;  width:
100%;
}

#inputContainer button:hover {  cursor:
pointer;
}

.hasAccountText span {
font-weight: bold;  font-
size: 12px;  cursor: pointer;
}

.hasAccountText {
textalign: center; }

#registerForm,
#loginForm      {
display: none;  }
#loginText      {
padding:      45px;
display: table-cell; }
#loginText  h1  {
color:      #07d159;
font-size:      60px;
font-weight: bold; }
#loginText  h2  {
```

```

margin: 35px 0; }
#loginText ul
{ padding: 0;
}
#loginText li { font-size: 20px; list-style-type: none; padding:
5px 30px;
background: url(../images/icons/checkmark.png) no-repeat 0 0;
}
#inputContainer .errorMessage { color:
#07d159; font-size:
12px; display: block;
}

```

(ii) **Style.css** html,

```

body { padding: 0;
margin: 0; height:
100%;
} *:focus { outline:
none;
}
* {
font-family: "Helvetica Neue", Helvetica, Arial, sans-serif; color:
#fff; letter-spacing: 1px;
} body

```

```
{ backgr  
oundcol  
or:  
#1818 18;  
fontsize:  
14px;  
minwidt  
h:  
720px  
;  
}  
  
#nowPlayingBarContainer { width:  
100%; backgroundcolor:  
#282828; bottom: 0; position:  
fixed; min-width:  
620px;  
}  
  
#nowPlayingBar {  
display: flex; height:  
90px; padding: 16px;  
box-sizing: border-box; }  
  
#nowPlayingLeft,  
#nowPlayingRight {  
width: 30%; min-width:  
180px;  
}  
  
#nowPlayingRight { position:  
relative; margintop:  
16px;  
}
```



```
#nowPlayingCenter { width:
40%; max-width: 700px;
display: flex; flexdirection:
column; alignitems: center;
}
#nowPlayingBar .content { width:
100%; height:
57px;
}
.playerControls .buttons { margin:
0 auto; display:
table; }
.controlButton { background-
color: transparent; border: none; vertical-align:
middle; }
.controlButton img { width:
20px; height: 20px; }
.controlButton.play img,
.controlButton.pause img {
width: 32px; height: 32px; }
.controlButton:hover {
cursor: pointer;
}
.progressTime { color:
#a0a0a0; fontsize:
11px; minwidth:
40px; text-align:
center; }
.playbackBar {
display: flex; }
```

```
.progressBar {      width:
100%;  height: 12px;
display: inline-flex;
cursor: pointer;
}

.progressBarBg {  background-color:
#404040;  height: 4px;  width:
100%;  border-radius: 2px;
}

.progress {  background- color:
#a0a0a0;  height: 4px;  width:
0;  border-radius: 2px;
}

.playbackBar .progressBar {  margintop:
3px;
}

#nowPlayingLeft .albumArtwork {
height: 100%;  max-width: 57px;
margin-right:
    15px;
    float: left;
    background-size: cover;
}

#nowPlayingLeft .trackInfo {  display:
table;
}

#nowPlayingLeft .trackInfo .trackName {
margin: 6px 0;  display: inline-block;
width: 100%;
```

```

}

#nowPlayingLeft .trackInfo .artistName span {  fontsize:
12px;  color: #a0a0a0;
}

.volumeBar {  width: 180px;  position:
absolute;  right:
0;
}

.volumeBar .progressBar
{  width: 125px;  }

#topContainer      {
minheight:      100%;
width:
100%;
}

#navBarContainer      {
background-color: #000; width:
220px; height: 100vh;
    position: fixed;  top: 0;  left:
0; } .navBar {  padding: 25px;
display: flex;  flexdirection:
column;
    -ms-flex-direction: column;
}

.logo {  margin-bottom:
15px;  }

.logo img {
width: 32px;
}  .navBar

```

```
.group {
border-top:
1px solid
#a0a0a0;
padding:
10px 0; }
.navItem {
padding:
10px 0; font-size: 14px;
font-weight: 700;
display: block;
letterspacing: 1px;
position: relative;

} .navItemLink {
color: #a0a0a0;
text-decoration: none; }
.navItemLink:hover {
color: #ffffff; }
.navItemLink .icon {
position: absolute; right:
0; top: 6px; width:
25px;
}
#mainViewContainer { margin-left:
220px; padding-bottom:
90px; width: calc(100% -
220px);
}
```

```
#mainContent {
padding: 0 20px; }

.pageHeadingBig {
padding: 20px;
textalign: center; }

.gridViewItem { display:
inline-block; margin-
right: 20px; width: 29%;
max-width: 200px; min-width: 150px; margin-bottom:
20px;
}

.gridViewItem img {
width: 100%; }

.gridViewInfo {
fontweight: 300; textalign:
center; padding:
5px 0; text-overflow:
ellipsis; overflow:
hidden; whitespace:
nowrap;
}

.gridViewItem a { textdecoration:
none;
}

.entityInfo { padding: 40px
0 10px 0; display: inline-
block; width: 100%;
}

.entityInfo .leftSection {
width: 30%; float:
```

```
left; max-width:
250px;
}
.entityInfo .leftSection img { width:
100%;
}
.entityInfo .rightSection { width:
70%; float: left; padding: 5px 10px
5px 40px; box-sizing: borderbox; }
.entityInfo .rightSection h2 {
margintop: 0px;
}
.entityInfo .rightSection p {
color: #939393;
fontweight: 200; }
.tracklist { padding: 0; }
.tracklistRow { height: 40px;
padding: 15px 10px; list-style:
none; } .tracklistRow span { color:
#939393; font-weight: 200; }
.tracklistRow:hover { backgroundcolor:
#282828;
}
.tracklistRow .trackCount {
width: 8%; float: left;
}
.tracklistRow .trackCount img { width:
20px; visibility: hidden; position:
absolute; cursor:
```

```
pointer;
}

.tracklistRow:hover .trackCount img {  visibility:
visible;

}

.tracklistRow .trackCount span {  visibility:
visible;
}

.tracklistRow:hover .trackCount span
{  visibility: hidden; } .tracklistRow
.trackInfo {  width: 75%;  float:
left;
}

.tracklistRow .trackInfo span {  display:
block;
}

.tracklistRow .trackOptions {  width:
5%; float: left; text-align:
right;
}

.tracklistRow .trackOptions img {  width:
15px;  visibility: hidden;
}

.tracklistRow:hover .trackOptions img {  visibility:
visible;
}

.tracklistRow .trackDuration {  width:
12%;  float: left;  text-align:
right;
```

```

}

.tracklistRow .trackInfo .trackName { color:
#fff; margin-bottom: 7px;
} .artistInfo {
textalign: center; }
.button { color:
#fff; cursor:
pointer; margin-
bottom: 10px;
background-color:
transparent;
fontweight: 500;
letter-spacing:
2px; border: 2px solid #fff;
border-radius: 500px; padding:
15px; minwidth:
130px;
}

.button.green { backgroundcolor:
#2ebd59; border-color:
#2ebd59;
}

.borderBottom { border-bottom: 1px solid
#939393; margin-bottom: 20px; }

.tracklistContainer h2, .gridViewContainer
h2,
.artistsContainer h2 { textalign:
center; } [role="link"] {
cursor: pointer; }

```



```
.searchContainer {
background-color: #242424;
padding: 20px 28px; }
.searchContainer h4 { margin:
0; fontweight: 300;
}
.searchContainer .searchInput { width:
100%; background-color: transparent;
border: none; color: #fff; font-weight:

700; font-size:
25px; letter-spacing:
2px; margin-top: 5px; }
input:focus { outline:
none; } .searchResultRow
{ padding: 15px 10px;
}
.searchResultRow:hover { backgroundcolor:
#282828;
}
.searchResultRow .artistName span { color:
#fff; } .noResults { margin:
15px; display: block; } .buttonItems
.button { display: block; margin:
0 auto 20px auto; }
.playlistsContainer { padding:
10px 0;
}
.playlistImage { border: 3px solid #282828;
padding:
```

```
10px;

}

.optionsMenu { position: fixed; background-color:
#282828; width: 200px; border: 1px solid rgba(0,
0, 0,
0.15); border-radius: 3px; z-index:
1000; display: none; }

.optionsMenu .item { width: 100%;
padding: 12px 20px; box-sizing:
border-box; fontweight: 400;
color: rgba(147, 147, 147, 0.8);
cursor: pointer; height: 40px;
background-color:
#282828; font-size: 14px;
}

.optionsMenu .item:hover { color:
#fff; backgroundcolor:
#333; border-color: #333; }

.optionsMenu select {
border: none;

-webkit-appearance: none; -moz-appearance: none;
}

.userInfo h1 {
text-align: center; }

.userDetails {
padding: 20px;
}

.userDetails .container h2 { textalign:
center;
}
```

```
.userDetails .container input[type="text"],
.userDetails .container input[type="password"],
.userDetails .container .button { display:
block; margin: 16px auto;
}
.userDetails .container input[type="text"], .userDetails
.container input[type="password"] { height: 50px;
width: 100%; background: #1f1f1f; border:
none; font-size: 18px; font-weight: 300;
padding: 0 20px;
}
.userDetails .container .message { color:
#2ebd59; font-weight:
300;}
```