Spring Fundamentals Exam

Spotify Playlist Application

Exam for the ["Spring Fundamentals" course @ SoftUni](https://softuni.bg).

The **Spotify Playlist** Application is here to help us keep in mind our shopping needs. The application is here for your music taste and aims to make music a more social experience. In Spotify Playlist, you can find hundreds of new songs that people had discovered and then you can add them to your customized playlist or maybe add your favorite songs so other people can listen to them. You can find new musical experiences by searching for styles of music you prefer.

There are several requirements you must follow in the implementation:

1. Database Requirements

The **Database** of the **Spotify Playlist** application needs to support **3 entities**:

User

- Has an **Id – UUID-String or Long**
- Has a **Username (unique, not null)**
  - Username length must be between 3 and 20 characters (inclusive of 3 and 20).
- Has a **Password (not null)**
  - Password length must be between 3 and 20 characters (inclusive of 3 and 20).
- Has an **Email (unique, not null)**
  - Must contain '@'.
- Has a **Playlist**
  - The playlist contains songs. One **user** may have many **songs** and one **song** can be **saved** by many **users to their playlist**.

Song

- Has an **Id – UUID-String or Long**
- Has a **Performer (unique, not null)**
  - Performer name length must be between 3 and 20 characters (inclusive of 3 and 20).
- Has a **Title (not null)**
  - Title length must be between 2 and 20 characters (inclusive of 2 and 20).
- Has a **Duration (not null)**
  - The duration must be a positive number
- Has a **Release date (optional)**
  - The **Date** that cannot be in the future
- Has a **Style (not null)**
  - One song **has one** style and one style **can have many** songs.

Style

- Has an **Id – UUID-string or Long**
- Has a **Style name (unique, not null)**
    - an option between (**POP**, **ROCK**, **JAZZ**)
- Has a **Description (optional)**

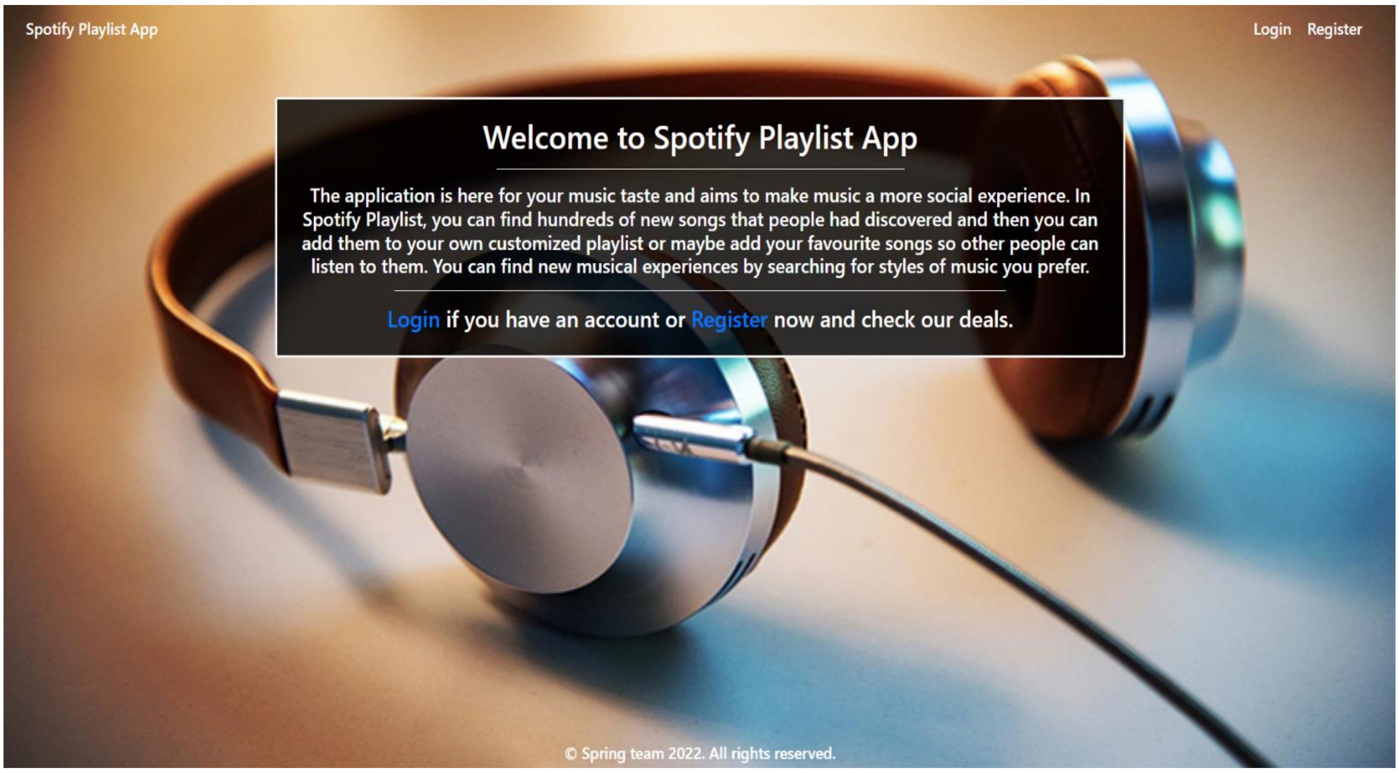
Implement the entities with the **correct data types** and implement **repositories** for them.

2. Initialize styles

- Implement a method that checks (when the app started) if the database does not have styles and initialize them
    - **You are free to do it in different ways.**

3. Page Requirements

Index Page (logged out user)



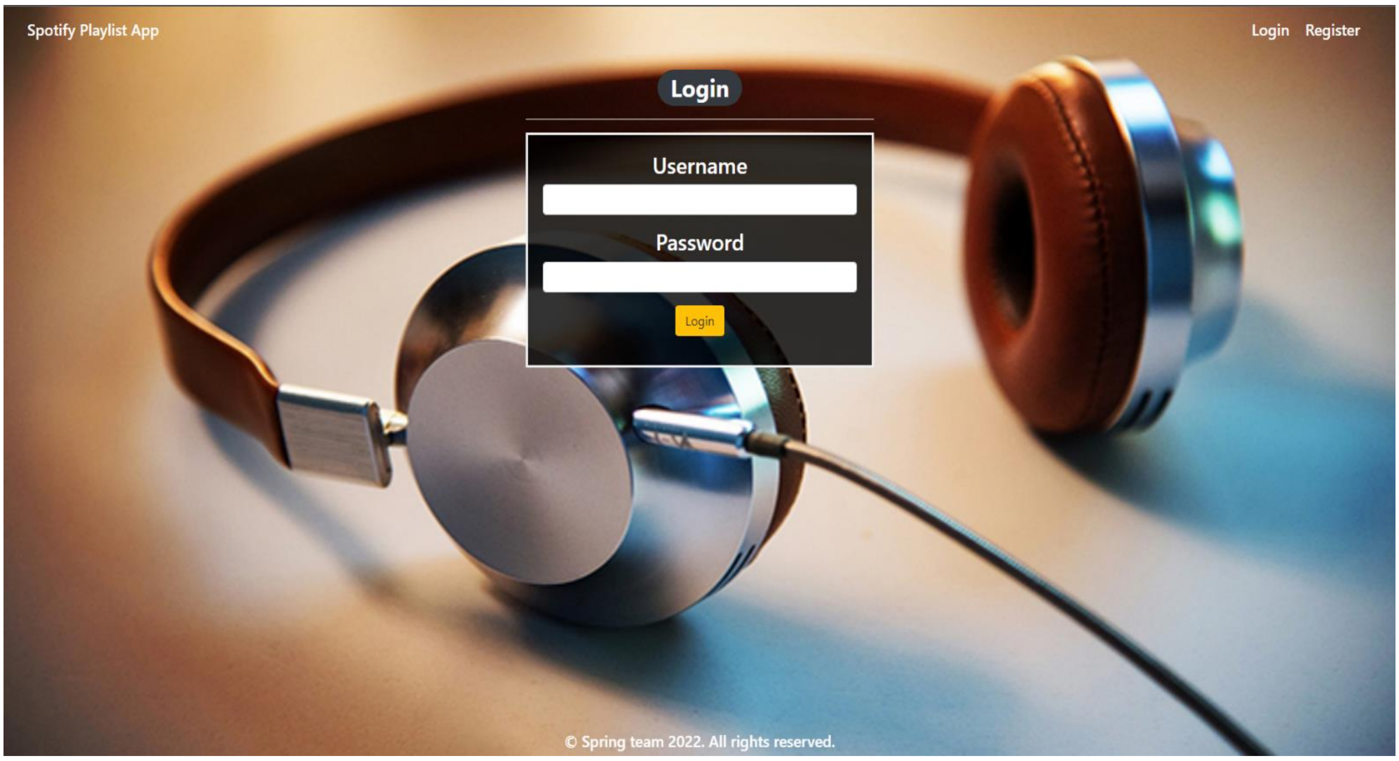
Welcome to Spotify Playlist App

The application is here for your music taste and aims to make music a more social experience. In Spotify Playlist, you can find hundreds of new songs that people had discovered and then you can add them to your own customized playlist or maybe add your favourite songs so other people can listen to them. You can find new musical experiences by searching for styles of music you prefer.

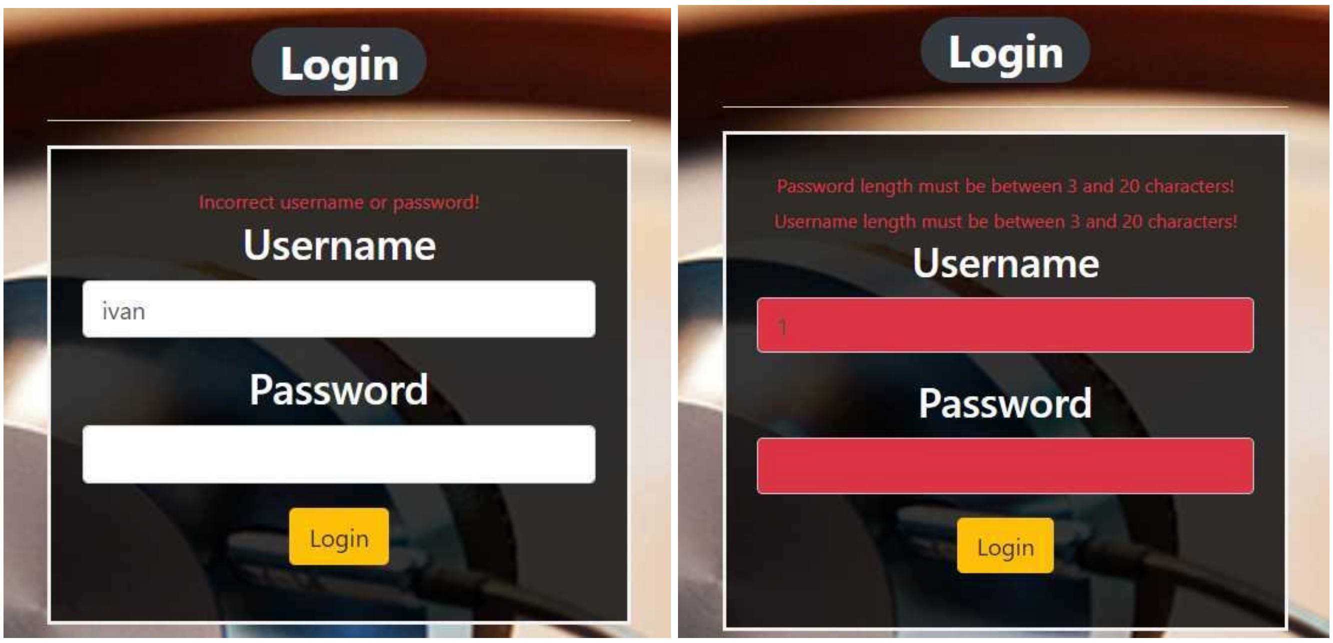
Login if you have an account or Register now and check our deals.

© Spring team 2022. All rights reserved.

Login Page (logged out user)



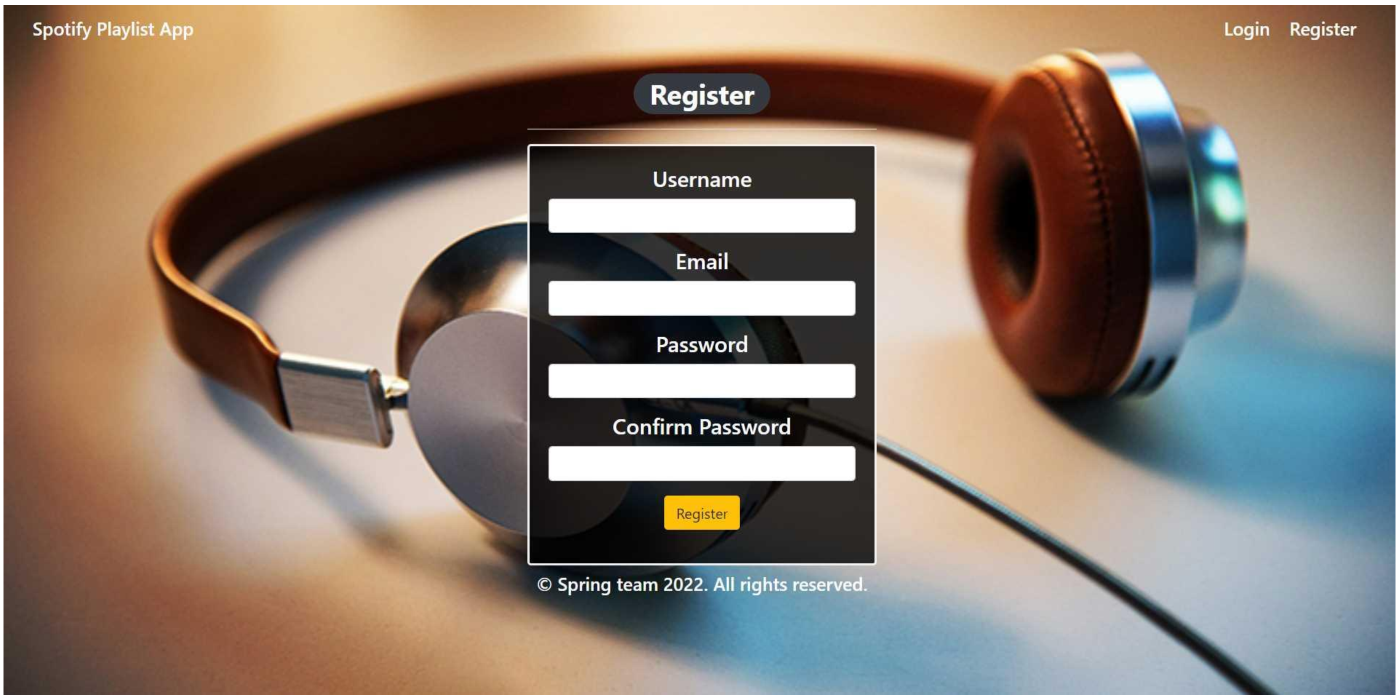
Login Page validations


Incorrect username or password!

Password length must be between 3 and 20 characters!

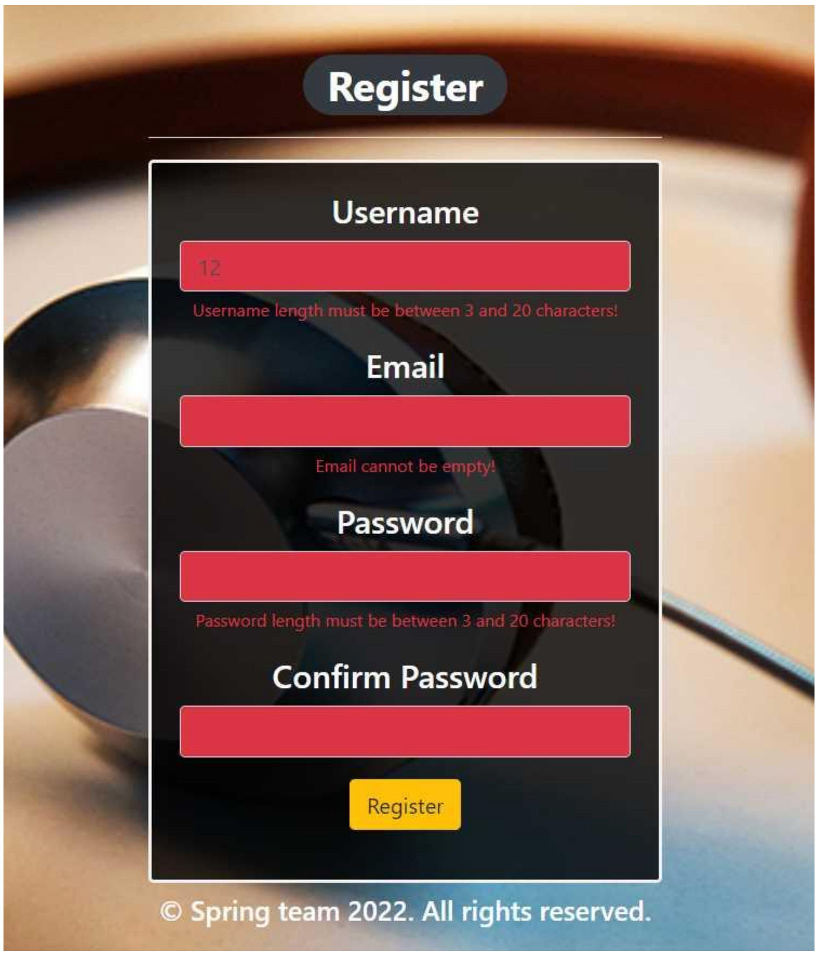
Username length must be between 3 and 20 characters!

Register Page (logged out user)



© Spring team 2022. All rights reserved.

Register Page validations



Username length must be between 3 and 20 characters!

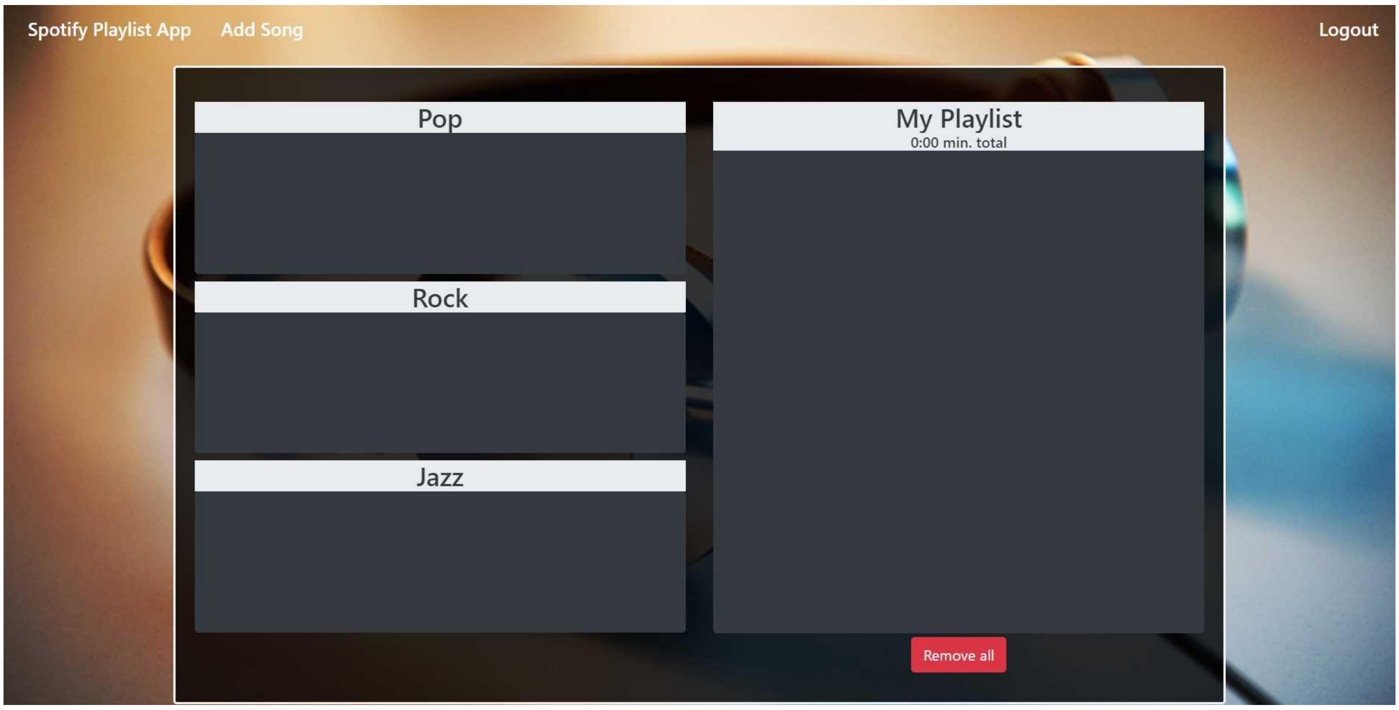
Email cannot be empty!

Password length must be between 3 and 20 characters!

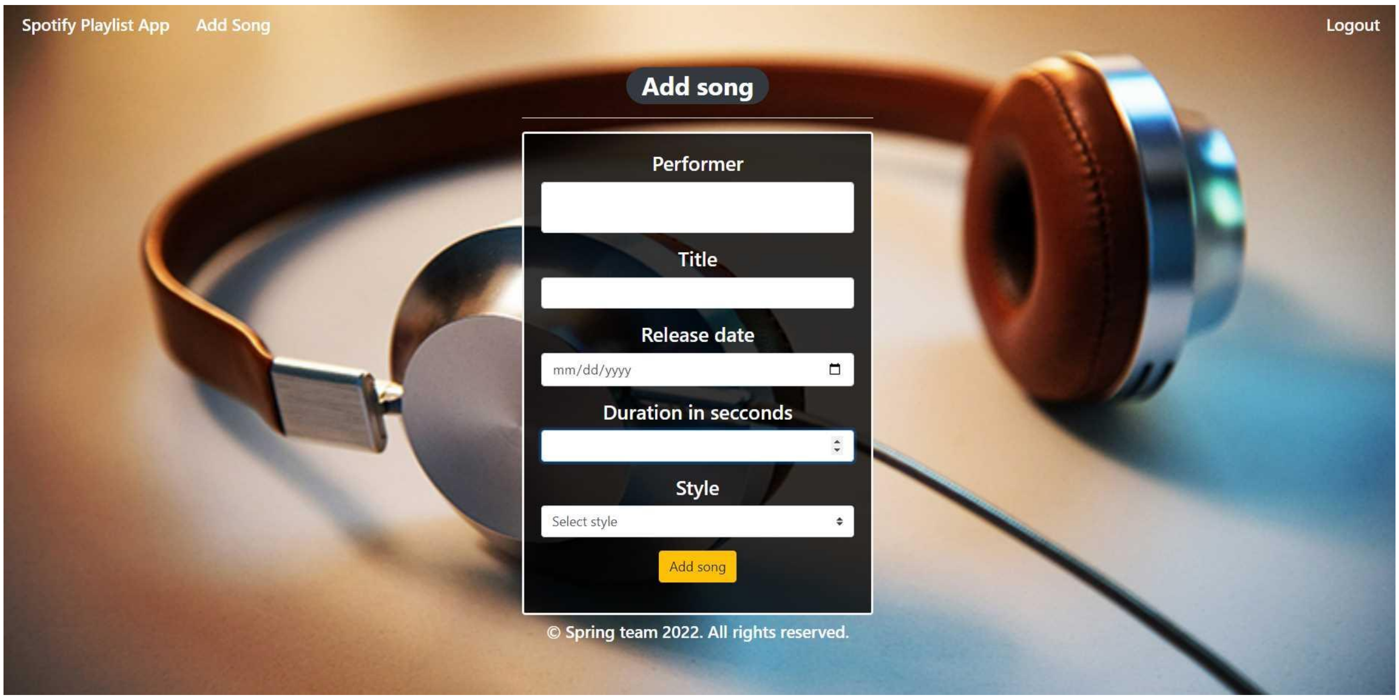
© Spring team 2022. All rights reserved.

Home Page (without having any songs)

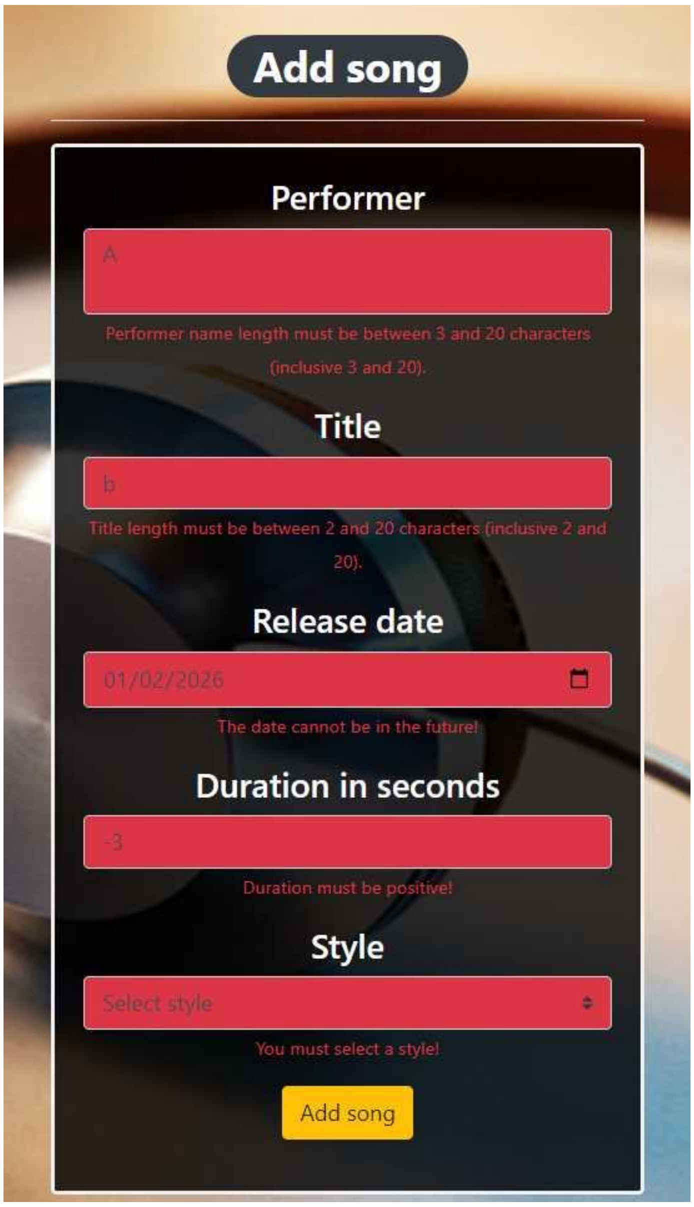
- Note: The left section of the page should visualize **all of the songs** from the database.
- Note: The right section (My Playlist) of the page should visualize **only songs** from the playlist of the **current logged user**.



Add songs



Add songs validation



Performer name length must be between 3 and 20 characters (inclusive 3 and 20).

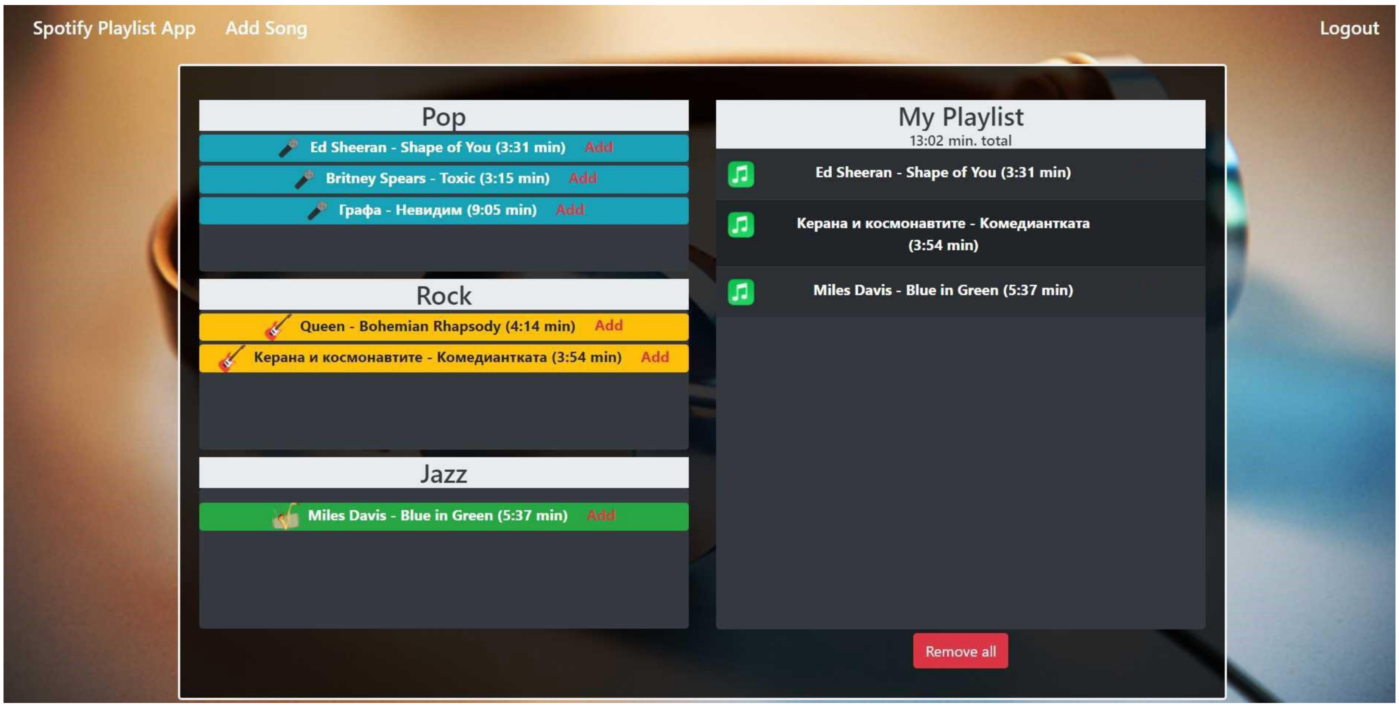
Title length must be between 2 and 20 characters (inclusive 2 and 20).

The date cannot be in the future!

Duration must be positive!

You must select a style!

Home Page (with songs)



Pop
Ed Sheeran - Shape of You (3:31 min) Add
Britney Spears - Toxic (3:15 min) Add
Графа - Невидим (9:05 min) Add

Rock
Queen - Bohemian Rhapsody (4:14 min) Add
Керана и космонавтите - Комедиантката (3:54 min) Add

Jazz
Miles Davis - Blue in Green (5:37 min) Add

My Playlist
13:02 min. total
Ed Sheeran - Shape of You (3:31 min)
Керана и космонавтите - Комедиантката (3:54 min)
Miles Davis - Blue in Green (5:37 min)

The templates have been given to you in the application skeleton, so make sure you implement the pages correctly.

NOTE: The templates should look **EXACTLY** as shown above.

NOTE: The templates do **NOT require additional CSS** for you to write. Only **bootstrap** and the **given CSS** are enough.

4. Functional Requirements

The Functionality Requirements describe the functionality that the application must support.

The application should provide **Guest** (not logged in) users with the functionality to log in, register and view the Index page.

The application should provide **Users** (logged in) with the functionality to **log out, add a new song (Add song page), view all songs (Home page) and add the songs to My Playlist or Remove All songs from My Playlist.**

Spotify Playlist App in navbar should **redirect** to the appropriate URL depending on that if the **user is logged in**.

The application should provide functionality for **adding songs** with styles of **Pop, Rock, or Jazz**.

The songs should be separated into different sections according to their styles.

The image also depends on the item's category.

When the user clicks on the **Add button of some song**, he saves the song to **his playlist**. You should **not delete** this song from DB. When he clicks on **Remove all songs**, just delete all songs in the **user's playlist** in DB and again redirect to the home page.

The Remove all songs button **should remove all songs only from the current user's** playlist.

Bellow the My Playlist banner is located an info bar that shows the **sum of the duration** of all added songs in My Playlist.

The application should store its data in a MySQL database.

5. Security Requirements

The **Security Requirements** are mainly access requirements. Configurations about which users can access specific functionalities and pages.

- **Guest** (not logged in) users can access the **Index** page.
- **Guest** (not logged in) users can access the **Login** page.
- **Guest** (not logged in) users can access the **Register** page.
- **Users** (logged in) can access the **Home** page.
- **Users** (logged in) can access **Add Song** page.
- **Users** (logged in) can access **Logout** functionality.

6. Scoring

Database – 10 points.

Pages – 25 points.

Functionality – 35 points.

Security – 5 points.

Validations – 15 points.

Code Quality – 10 points.