



## My Music Database Application

### Regular Exam [Spring Fundamentals]

The **My Music DB Application** is here to help us gather information about our favorite music artists and their albums. Of course, this is a lot of work, but for starters we will start with three of the most popular artists. Once we find all their albums, we will add more artists so that our application can have even more data, to help users find information about their favorite artists and tiny albums. At this stage, the functionality of the application is small, but later I believe that you could future develop it by adding many new futures.

There are several requirements you must follow in the implementation:

#### 1. Database Requirements

The **Database** of the **My Music DB** application needs to support **3 entities**:

##### User

- **Id** – Accepts **UUID-String** or **Long** values
- **Username**
  - The **length** of the **values** should be **between 3 and 20** characters long (both numbers are **INCLUSIVE**)
  - The values should be **unique** in the database
- **Full name**
  - The **length** of the **values** should be **between 3 and 20** characters long (both numbers are **INCLUSIVE**)
- **Password**
  - The **length** of the **values** should be **between 4 and 20** characters long (both numbers are **INCLUSIVE**)
- **Email**
  - The values should contain a '@' symbol)
  - The values should be **unique** in the database

##### Album

- **Id** – Accepts **UUID-String** or **Long** values
- **Name**
  - The **length** of the **values** should be **between 3 and 20** characters long (both numbers are **INCLUSIVE**)
- **Image Url**
  - The **length** of the **values** should be **at least 5** characters long
  - The values should be **unique** in the database
- **Description**
  - The **length** of the **values** should be **at least 5** characters long
  - The description is a long text field
- **Copies**
  - All sold copies of the album
  - The values should be **not less than 10**
- **Price**
  - The values must be a **positive** numbers
- **Released Date**
  - **Date** values, that **cannot** be in the **future**

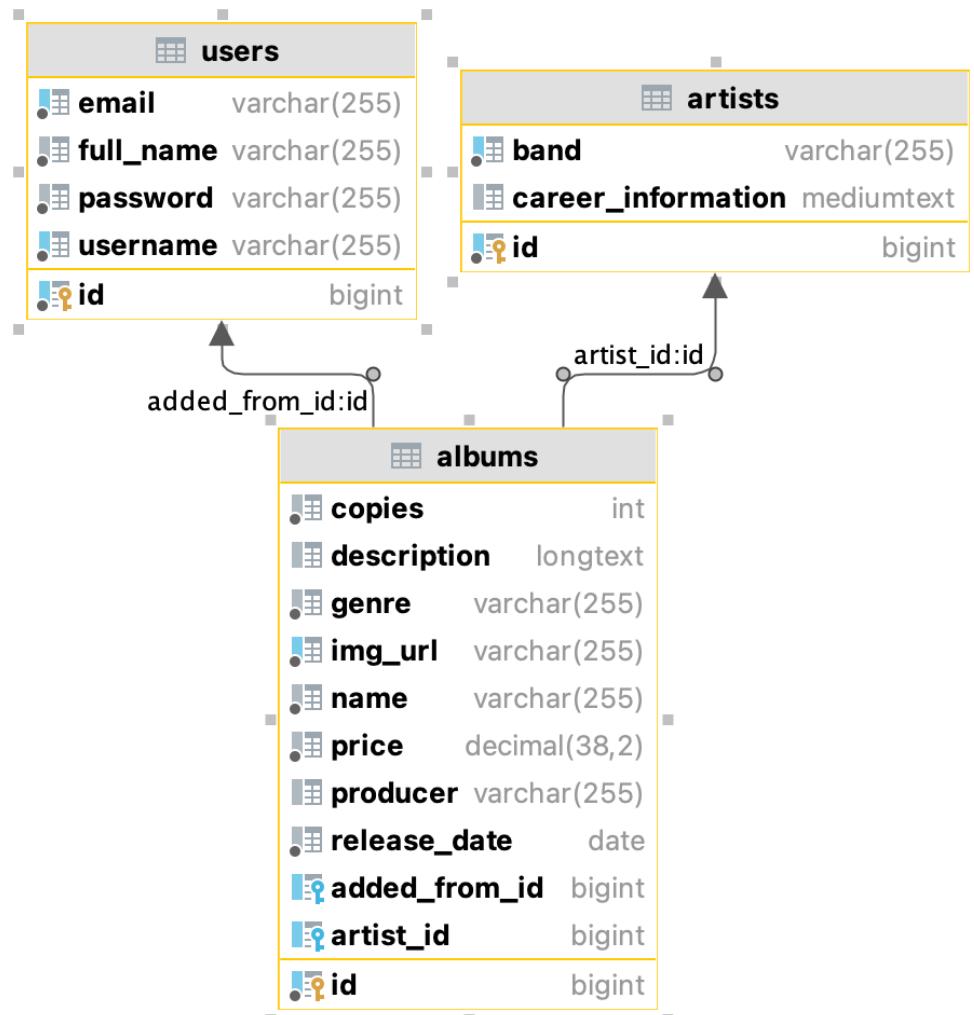
- **Producer**
  - Name of producer
  - Can be null
- **Genre**
  - Option between (**POP, ROCK, METAL or OTHER**)
  - This is just an enumeration, not an entity
- **Artist**
  - Has only one artist
  - This is the relation with artists
- **Added From**
  - A user that **adds this album** to the **My Music DB** application

### **Artist**

- **Id** – Accepts **UUID-String** or **Long** values
- **Band/Singer name**
  - Name of the singer or the band
  - an option between (**QUEEN, METALLICA, or MADONNA**)
- **Career Information**
  - The career information is a long text field
  - This is optional, can be set to null

**Nullable/Empty/Blank values are not allowed unless explicitly mentioned.** Implement the entities with the **correct data types** and implement the **repositories** for them.

ER Diagram:

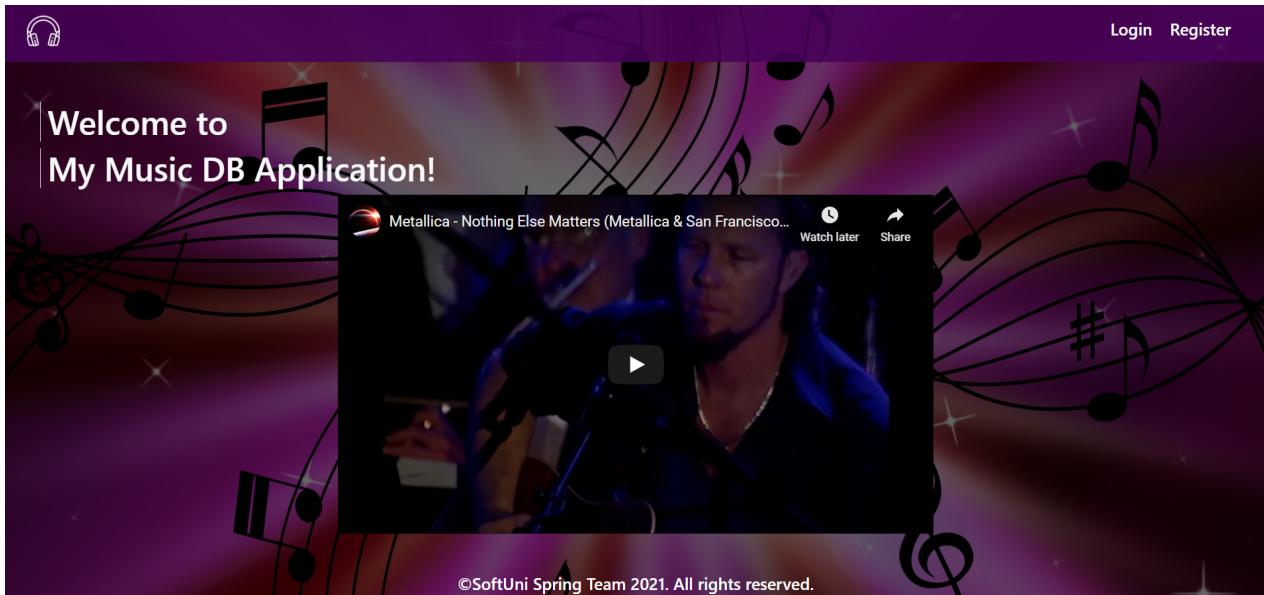


## 2. Initialize artists

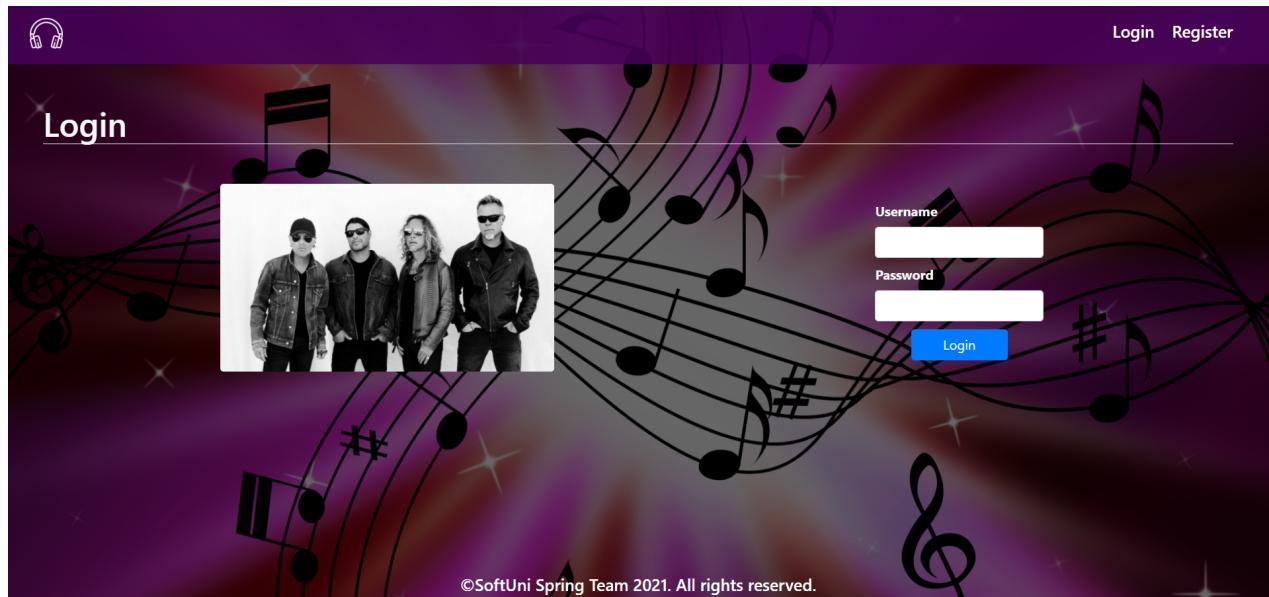
- Implement a method that checks (when app started) if the database does not have any category and initialize them
  - You are free to do this in some different ways
  - You can skip the description if you want
  - If you want, you can skip the career information to be null or add this:
    - **Queen Career information:**
      - Queen are a British rock band formed in London in 1970. Their classic line-up was Freddie Mercury (lead vocals, piano), Brian May (guitar, vocals), Roger Taylor (drums, vocals) and John Deacon (bass). Their earliest works were influenced by progressive rock, hard rock and heavy metal, but the band gradually ventured into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock.
    - **Metallica Career information:**
      - Metallica is an American heavy metal band. The band was formed in 1981 in Los Angeles by vocalist/guitarist James Hetfield and drummer Lars Ulrich, and has been based in San Francisco for most of its career. The band's fast tempos, instrumentals and aggressive musicianship made them one of the founding "big four" bands of thrash metal, alongside Megadeth, Anthrax and Slayer. Metallica's current lineup comprises founding members and primary songwriters Hetfield and Ulrich, longtime lead guitarist Kirk Hammett, and bassist Robert Trujillo. Guitarist Dave Mustaine and bassists Ron McGovney, Cliff Burton and Jason Newstead are former members of the band.
    - **Madonna Career information:**
      - Madonna Louise Ciccone - born and raised in Michigan, Madonna moved to New York City in 1978 to pursue a career in modern dance. After performing as a drummer, guitarist, and vocalist in the rock bands Breakfast Club and Emmy, she rose to solo stardom with her debut studio album, Madonna (1983). She followed it with a series of successful albums, including all-time bestsellers Like a Virgin (1984) and True Blue (1986) as well as Grammy Award winners Ray of Light (1998) and Confessions on a Dance Floor (2005).

## 3. Page Requirements

### Index Page (logged out user)



## Login Page (logged out user)



## Login Page validations

The image contains two side-by-side screenshots of the login page, demonstrating validation errors.

**Left Screenshot:** This shows the login form with validation errors. The "Username" field contains the character "1", which is too short, so a red error message below it states "Length must be between 3 and 20 characters". The "Password" field is empty, so a red error message below it states "Length must be between 5 and 20 characters". Both fields have a red background color. A blue "Login" button is at the bottom.

**Right Screenshot:** This shows the login form after entering "ivan" into the "Username" field. A large red banner at the top of the page displays the error message "User doesn't exist". The "Username" field now has a white background and contains "ivan". The "Password" field is empty and has a red background. A blue "Login" button is at the bottom.

## Registration Page (logged out user)



## Registration Page validations

- Note: it is not necessary to show message for different passwords, just not save the user and redirect again to the register page.

The validation errors are displayed in red boxes below each input field:

- Username:** 1. Username length must be between 3 and 20 characters.
- Full Name:** 1. Full name length must be between 3 and 20 characters.
- Email:** invalid\_mail. Must be valid email.
- Password:** Password length must be between 5 and 20 characters.
- Confirm Password:** Password length must be between 5 and 20 characters.

A blue 'Register' button is at the bottom.

## Navigation (Guest user)

- Note: can access only to **Index**, **Login**, **Register** pages.

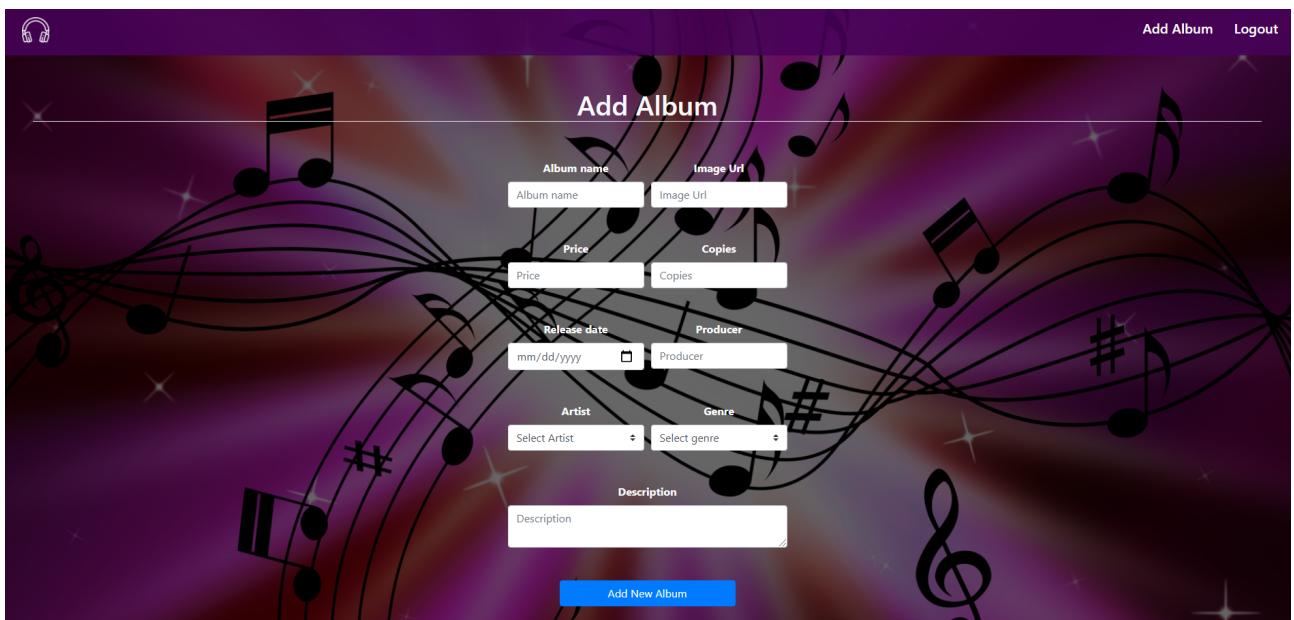


## Navigation (Registered user)

- Note: can access only to **Home** and **Add Order** pages and **Logout** functionality.



## Add Album



The image shows a form titled "Add Album" set against a background of musical notes and stars. The form fields include:

- Album name (input field)
- Image Url (input field)
- Price (input field)
- Copies (input field)
- Release date (input field with calendar icon)
- Producer (input field)
- Artist (dropdown menu)
- Genre (dropdown menu)
- Description (text area)
- A blue "Add New Album" button at the bottom.

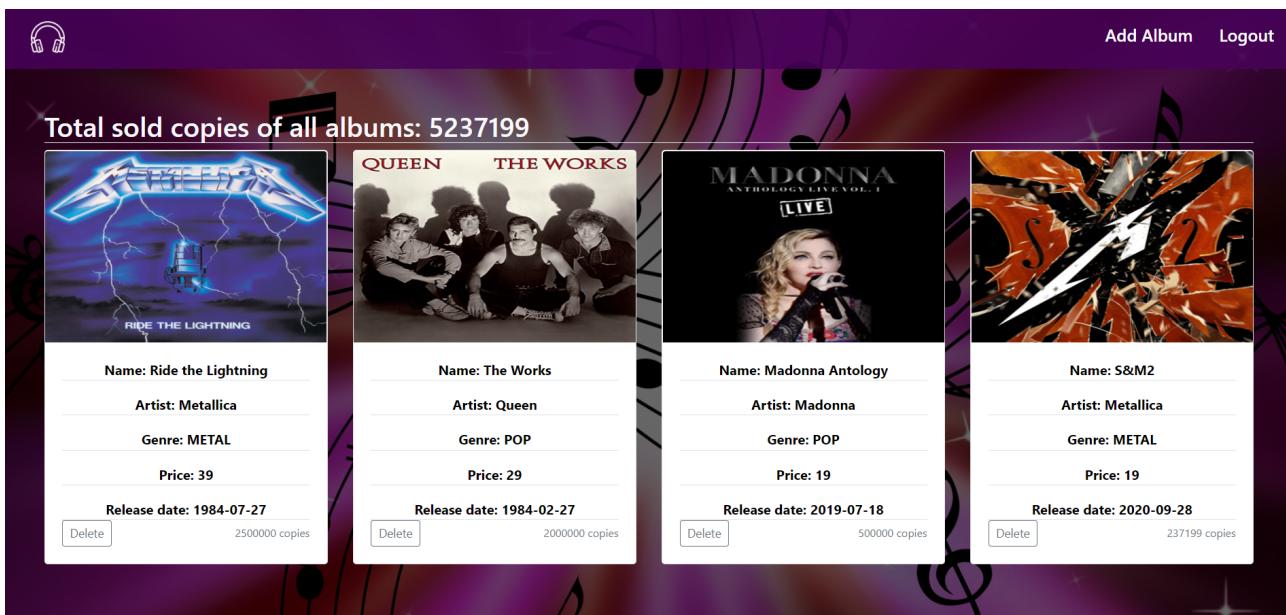
## Add Album Validation

The form consists of several input fields:

- Album name:** Value: 1. Error message: Name must be between 3 and 20 characters.
- Image Url:** Value: 1. Error message: Image url must be minimum 5 characters.
- Price:** Value: -1. Error message: Price must be positive number.
- Copies:** Value: 1. Error message: Copies must be more than 10.
- Release date:** Value: 10/12/2022. Error message: Release date cannot be in the future.
- Producer:** Value: Producer.
- Artist:** Value: Select Artist. Error message: You must select artist.
- Genre:** Value: Select genre. Error message: You must select genre.
- Description:** Value: 1. Error message: Description length must be minimum 5 characters.

**Add New Album** button at the bottom.

## Home Page (with orders)



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

### NOTES:

- The templates should look **EXACTLY** as shown above.
- The templates do **NOT require additional CSS** for you to write. Only **bootstrap** and the **given CSS** are enough.
- Albums should be sorted based on the **copies sold in descending order**.

## 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 **login**, **register** and **view the Index** page.

The application should provide Users (logged in) with the functionality to **logout**, **add an Album**, **view all Albums in DB (Home page)** and **Delete a single one** from albums.

In **My Music DB Application**, the navbar should redirect to appropriate **URL depending** on that if the user is logged in.

The **application** should provide **functionality** for **adding albums** with **artist** and **category**. Also can delete them from the database.

**Delete** button remove the selected album from the database and again **redirect** to the **home** page.

The first **header** on the **home** page shows the **total sum of all copies** of albums in the database at this moment

The application should store its data in a **MySQL** database or **PostgreSQL** 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:
  - **Index** page;
  - **Login** page;
  - **Register** page.
- **Users** (logged in) can access:
  - **Home** page;
  - **Add Album** page;
  - **Logout** functionality.

## **6. Scoring**

**Database – 10 points**

**Pages – 25 points**

**Functionality – 35 points**

**Security – 5 points**

**Validations – 15 points**

**Code Quality – 10 points**