



## Like Book Application

### Regular Exam [Spring Fundamentals]

With its easy-to-use interface and detail-rich profiles, **LikeBook** App makes it clear why it's one of the most enduring social network apps. This is the service to try if you're looking for a connection with friends and new communities.

There are several requirements you must follow in the implementation:

#### 1. Database Requirements

The **Database** of the **Like Book** 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
- **Password**
  - The **length** of the **values** should be **at least 3** and **20** characters long (both numbers are **INCLUSIVE**)
- **Email**
  - The values should contain a '@' symbol)
  - The values should be **unique** in the database

##### Post

- **Id** – Accepts **UUID-String** or **Long** values
- **Content**
  - The **length** of the **values** should be **between 2** and **150** characters long (both numbers are **INCLUSIVE**)
- **User**
  - The creator of the post. One **post** can have only one **user** as creator and one **user** can have many **posts**
- **User Likes**
  - The **user likes** contains users. One **user** may like many **posts** and one **post** can be liked by many **users**
- **Mood**
  - One post **has one** mood and one mood **can have many** posts

##### Mood

- **Id** – Accepts **UUID-String** or **Long** values
- Has a **Mood name**
  - an option between (**HAPPY**, **SAD** and **INSPIRED**)
- **Description**
  - optional

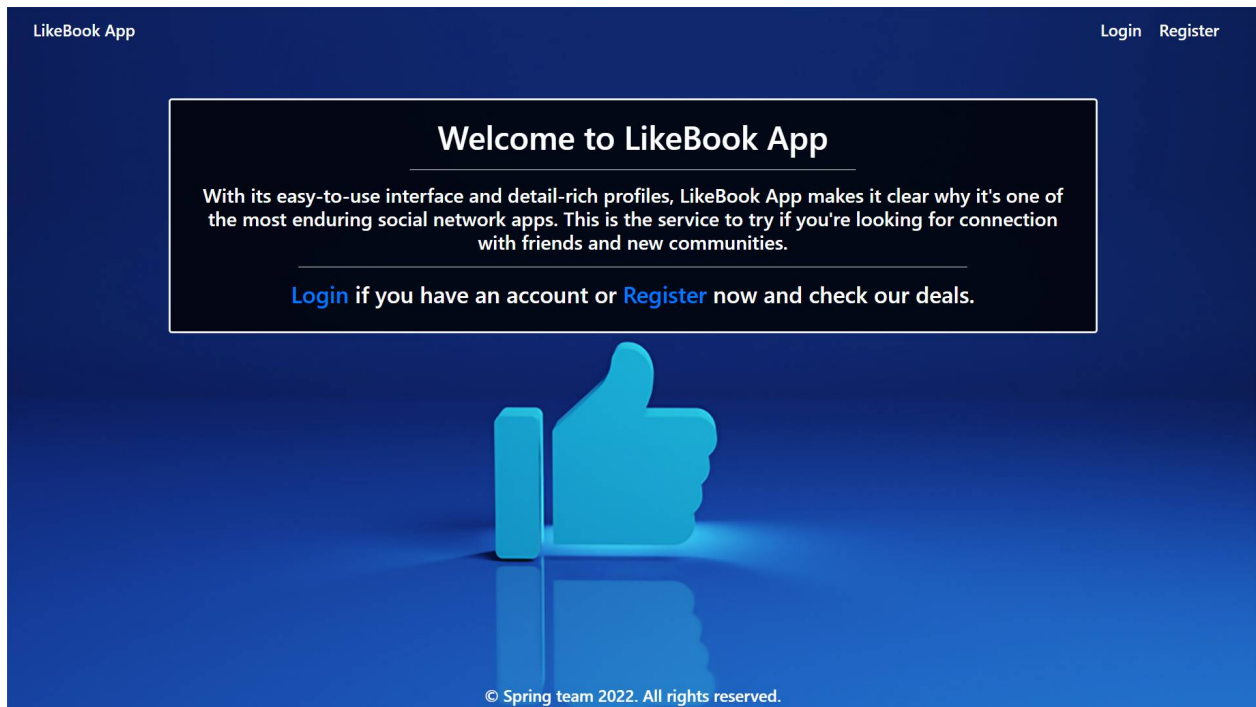
**Nullable/Empty/Blank values are not allowed unless explicitly mentioned.** Implement the entities with the **correct data types** and implement the **repositories** for them.

## 2. Initialize categories

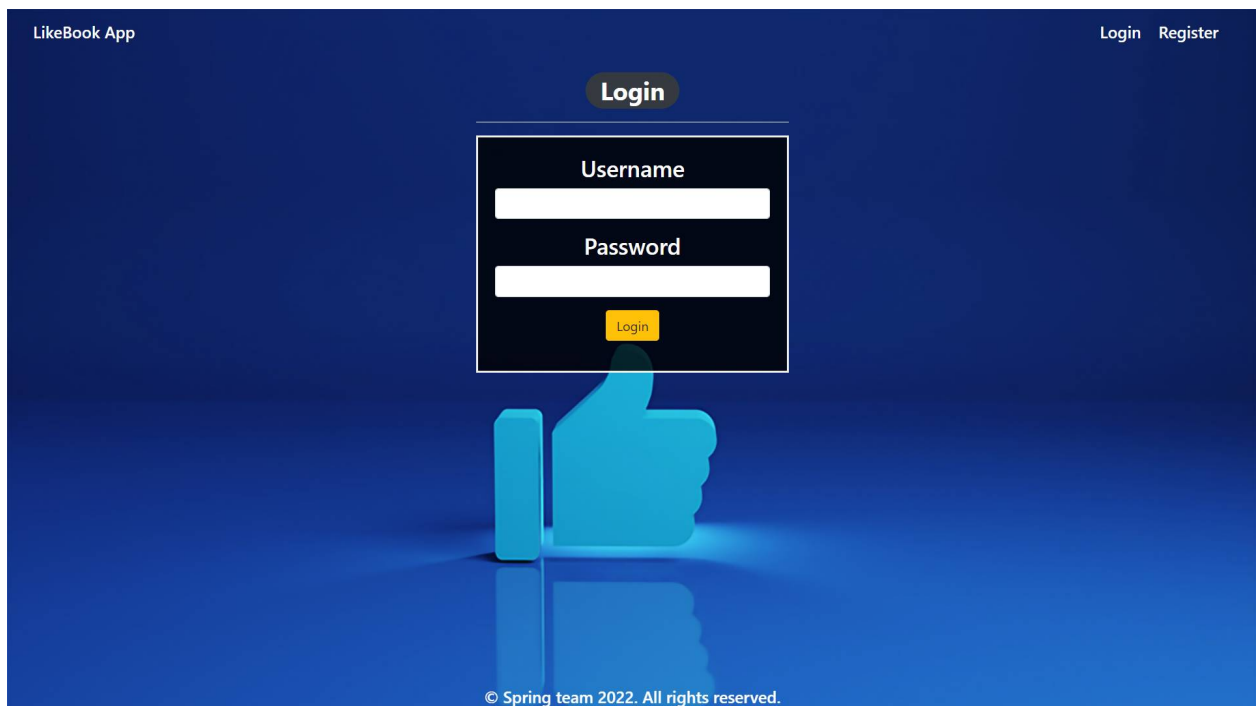
- Implement a method that checks (when app started) if the database does not have any category and initialize them
  - You are free to do this in some different ways
  - You can skip the description if you want

## 3. Page Requirements

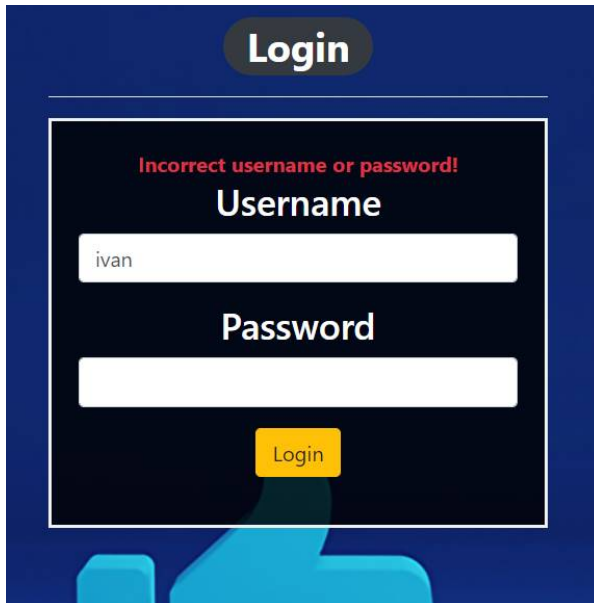
### Index Page (logged out user)



### Login Page (logged out user)



## Login Page validations



The image shows a login form titled "Login" with a dark blue background. A white-bordered box contains the form fields. At the top of the box, a red error message reads "Incorrect username or password!". Below this, the "Username" field contains the text "ivan". The "Password" field is empty. A yellow "Login" button is at the bottom of the box.

**Login**

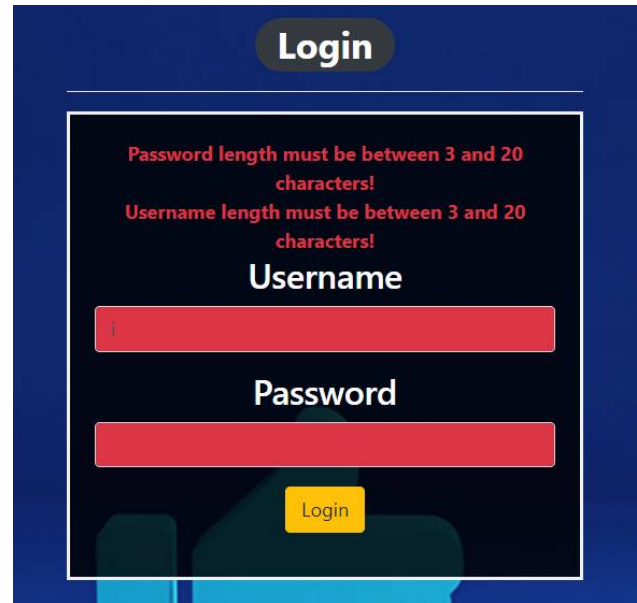
Incorrect username or password!

**Username**

ivan

**Password**

Login



The image shows a login form titled "Login" with a dark blue background. A white-bordered box contains the form fields. At the top of the box, two red error messages are displayed: "Password length must be between 3 and 20 characters!" and "Username length must be between 3 and 20 characters!". Below these, the "Username" field contains a single character "i". The "Password" field is empty. A yellow "Login" button is at the bottom of the box.

**Login**

Password length must be between 3 and 20 characters!

Username length must be between 3 and 20 characters!

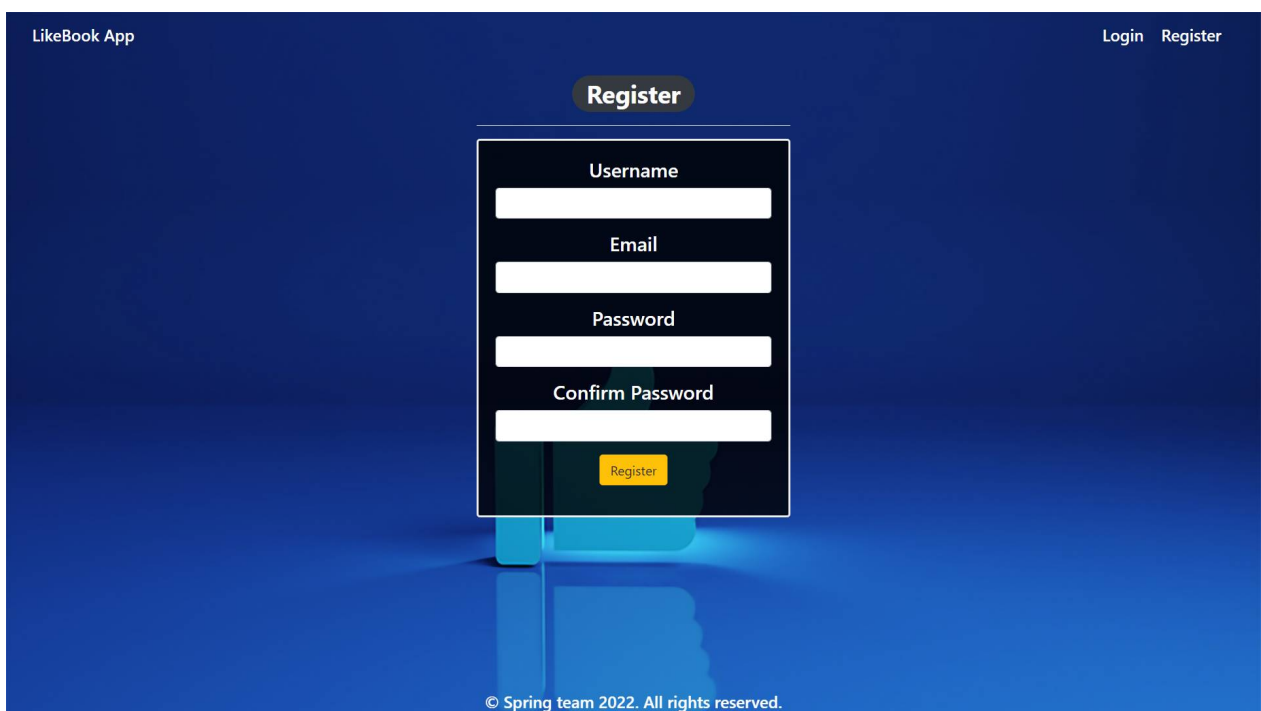
**Username**

i

**Password**

Login

## Registration Page (logged out user)



The image shows a registration page titled "LikeBook App" in the top left corner. In the top right corner, there are links for "Login" and "Register". The main content is a "Register" form with a dark blue background. The form has a white-bordered box containing four input fields: "Username", "Email", "Password", and "Confirm Password". A yellow "Register" button is at the bottom of the box. At the bottom of the page, there is a copyright notice: "© Spring team 2022. All rights reserved."

LikeBook App

Login Register

**Register**

Username

Email

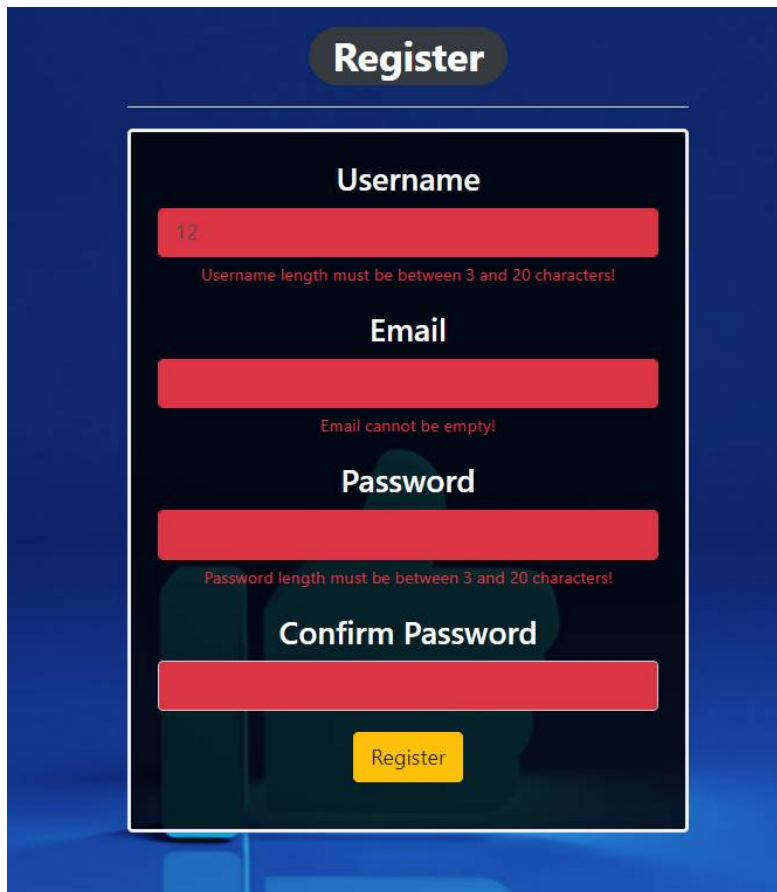
Password

Confirm Password

Register

© Spring team 2022. All rights reserved.

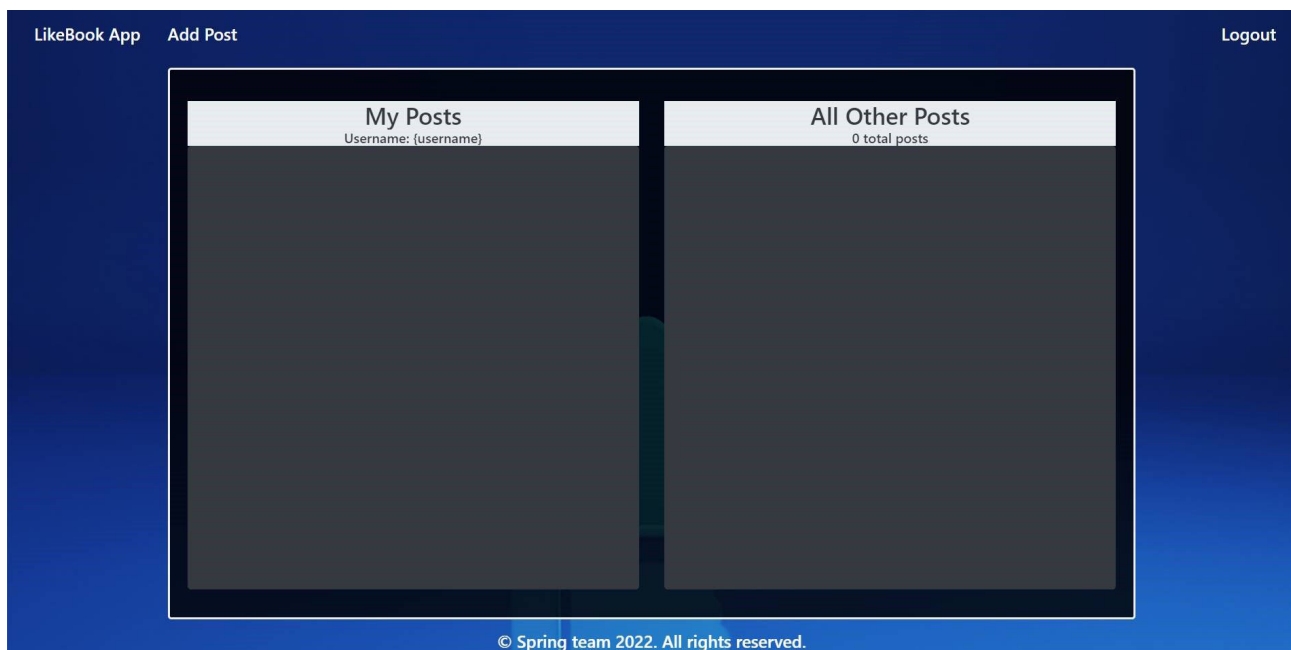
## Registration Page validations



The registration form is titled "Register" and is set against a dark blue background. It contains four input fields, each with a red border and a red error message below it. The fields are: Username (with the value "12" and the message "Username length must be between 3 and 20 characters!"), Email (with the message "Email cannot be empty!"), Password (with the message "Password length must be between 3 and 20 characters!"), and Confirm Password. A yellow "Register" button is located at the bottom of the form.

## Home Page (without having any posts)

- The left section of the page should visualize current logged user posts from the database.
- The right section (All Other Posts) of the page should visualize posts created by other users.



## Add post

LikeBook App Add Post Logout

**Add post**

**Content**

**Mood**

Select mood

Add post

© Spring team 2022. All rights reserved.

## Add post validation

**Add post**

**Content**

1

Content length must be between 2 and 50 characters!

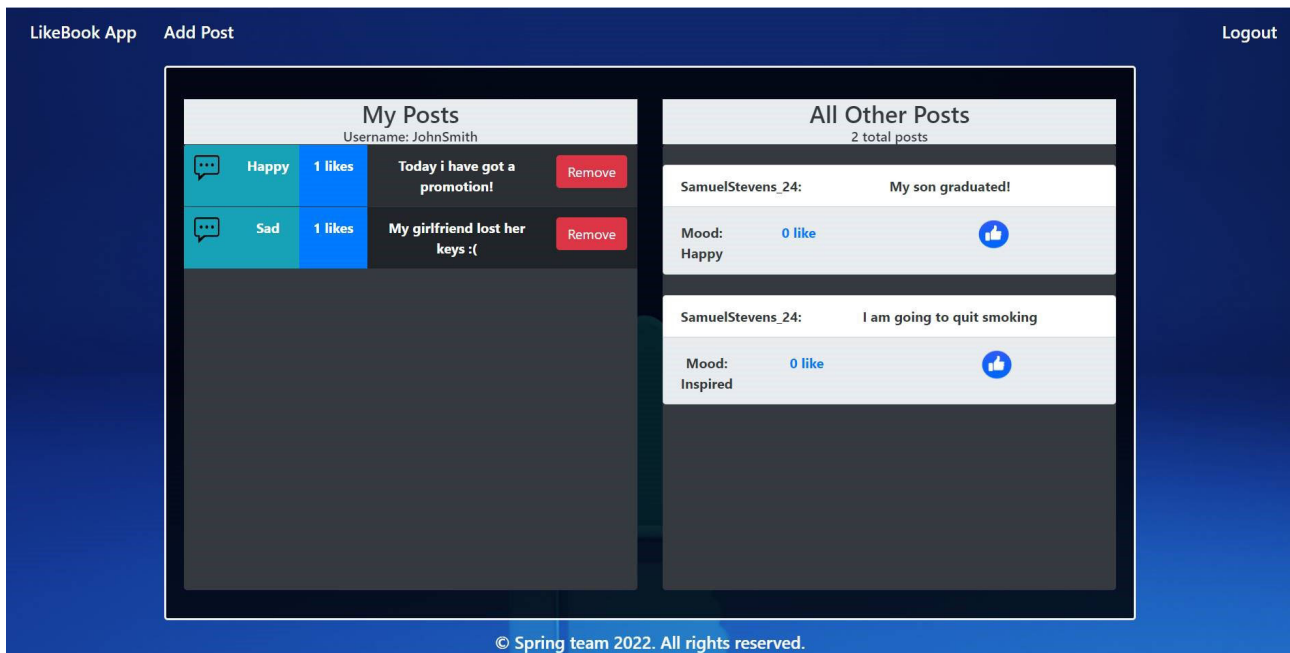
**Mood**

Select mood

You must select a mood!

Add post

## Home Page (with posts)



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.

## 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 a new post (Add Post Page)**, **view all posts (Home page)** and **like other people's posts from the All Other Posts section** and **remove their posts from the My Posts section**.

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

The application should provide functionality for **adding posts** with moods of **Happy, Sad and Inspired**.

The posts should be separated into different sections according to their creator.

When the user clicks on the **Like button of some post**, he likes the post. When he clicks on the **Remove** button, just delete the post from his **posts** in DB and redirect to the home page.

Bellow the All Other Posts banner is located an info bar that shows the **sum of all posts made by other users**.

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 Post** page;
- **Logout** functionality.

## 6. Scoring

**Database – 10 points**

**Pages – 25 points**

**Functionality – 35 points**

**Security – 5 points**

**Validations – 15 points**

**Code Quality – 10 points**