# **Lab: Spring MVC with Thymeleaf Basics**

*In this exercise, you will practice building a simple* ***Spring Boot*** *web application that connects a backend service with a user-friendly frontend using* ***Spring MVC*** *and* ***Thymeleaf****.*

### **Resources**

You will be provided with a fully functioning **Spring Boot** project. The project includes:

* **Models**: Entity classes for **User** and **Transaction**
* **Services**: Service layer for handling business logic
* **Repositories**: JPA repositories for database interactions

#### **HTML Files**

You are provided with two **raw** HTML files:

1. **users.html**: A page to display a list of all users
2. **user-details.html**: A page to display details for a single user

#### **Note**

The database configuration is located in the **application.properties** file. You must update the **username** and **password** according to your local database setup.

**There will be already data inserted into the database upon application startup for several users and transactions!**

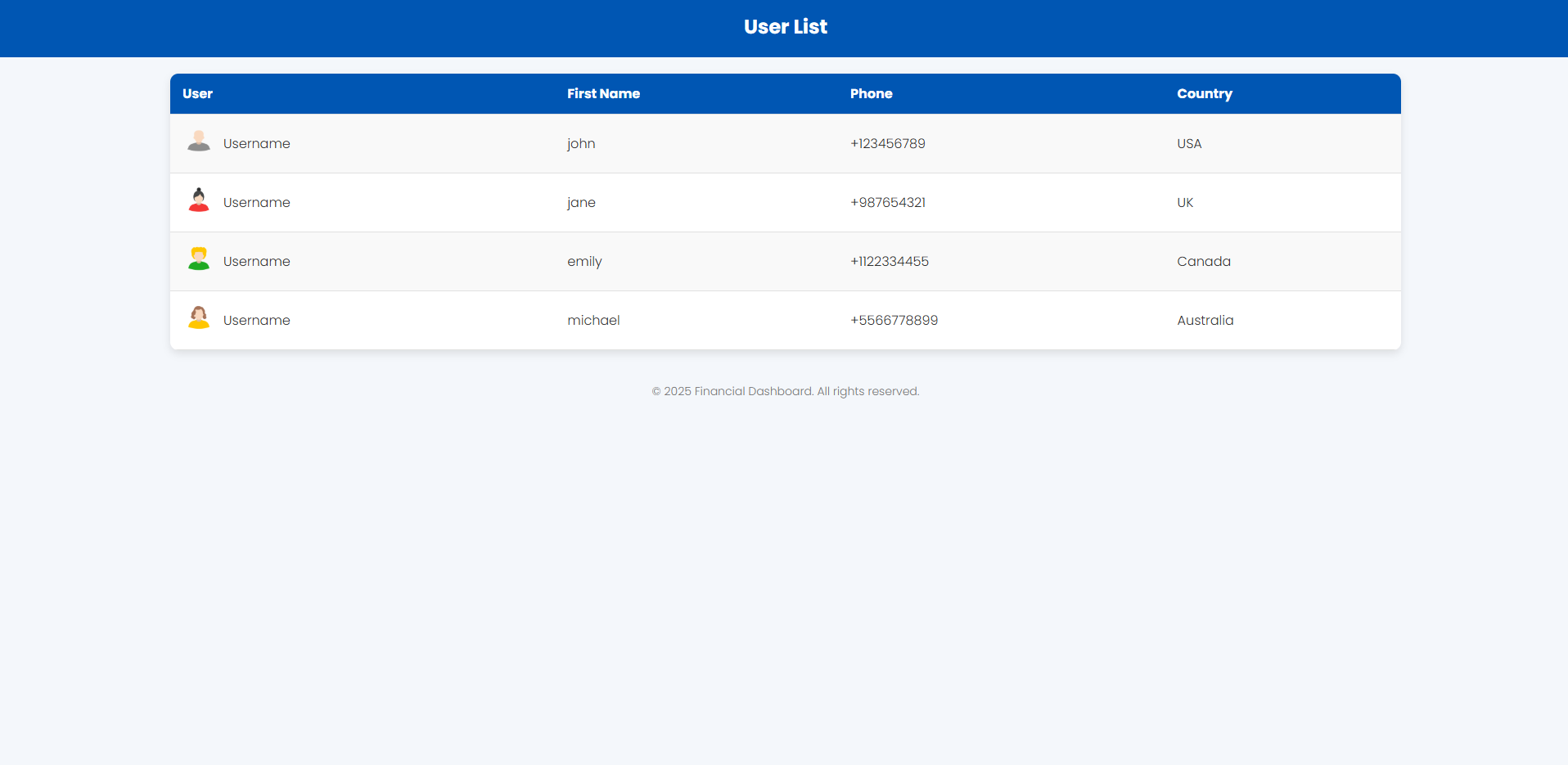
### **Exercises**

Your main goal is to integrate the HTML files with **Spring MVC** + **Thymeleaf**, and make the pages functional by populating them with **dynamic data** passed from the controller to the views.

#### **Exercise 1: Display List of Users**

Create a simple Controller and implement a method to handle a **GET** request to **/users** endpoint.

This method is supposed to fetch all the users from the database, add them to a model and render them on the **users.html** page. There is an anchor tag around each user's profile picture and you need to add functionality to send **GET** request to **/users/{id}** endpoint, that you will be implementing in Task 2.



#### **Exercise 2: Display Single User Details**

Add another handler method and implement functionality to process a **GET** request to **/users/{id}** endpoint. The **{id}** is a path variable and will be replaced by an actual id in the browser. *Example: /users/1 - that request is supposed to return details for a user with id 1.*

This method is supposed to fetch the details of a user by their id from the database, add the user's details to a model and render them on **user-details.html** view. Use a button labeled "**Users**" to navigate back to the **users.html** page.

