Spring MVC Workshop
Lab 2

1 Lab setup

Make sure you have the following items installed

- Latest version of JDK 11 (note: labs are tested with JDK 11 but should work with higher versions with no or minimal changes)
- Eclipse Enterprise Edition for Java (or a suitable alternative IDE for Enterprise Java)
- Tomcat 9.0 installed as a Windows service
- Latest version of Maven
- A suitable text editor (Notepad ++)
- A utility to extract zip files (7-zip)

In each of the main lab folders, there are two subfolders: `changes` and `final`. The `changes` subfolder just holds the source code files for the lab, while the `final` subfolder holds the complete Eclipse project starting from its project root folder. We will use the code from the `changes` subfolder to build up our applications from scratch and you can always fall back on the complete Eclipse project if you encounter any errors while building up the application.

2 Form tag library

The source code for this lab is found in `SpringMVC-FormTagLibrary/changes` folder.

Using either STS or Spring Initializr, create a Spring Boot Project with the following details:

Name: `FormTagSpringMVC`

Group: `com.workshop`
Artifact: `FormTagSpringMVC`
Version: `0.0.1-SNAPSHOT`
Description: `Demo project for Form Tag Library`
Package: `com.workshop.mvc`

Ensure you include the following dependencies:

```
Web -> Spring Web
Developer Tools -> Spring Boot DevTools
```

Add the following dependencies to the <dependencies> list in the POM:

```
<!-- This allows the compilation and rendering of JSP pages -->
<dependency>
  <groupId>org.apache.tomcat.embed</groupId>
  <artifactId>tomcat-embed-jasper</artifactId>
  <scope>provided</scope>
</dependency>

<!-- This provides the JSTL tags supported required in JSP pages -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
</dependency>
```

Save the POM and do a Maven -> Update Project

Create a `src/main/webapp/WEB-INF/views` folder in the project folder structure.
Make sure that `src/main/webapp` is on the project build path. Right click -> Build Path -> Use as Source Folder.

Place the following properties into `application.properties`

```
#Configuring Spring MVC
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

#Provides a context-root to deploy the app on
server.servlet.context-path=/FormTagSpringMVC
```

2.1 Text field tag <form:input>

Copy the following files from `changes` into `src/main/webapp/WEB-INF/views`

```
employee-details.jsp
employee-form.jsp
main-menu.jsp
```

Place `log4j2.properties` in `src/main/resources`

Copy the following files from `changes` into `com.workshop.mvc` in `src/main/java`:

```
Employee.java
EmployeeController.java
HomeController.java
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.2 Text area tag <form:textarea>

Make changes to the following files in `com.workshop.mvc` in `src/main/java` from `changes`:

```
EmployeeController-v2.java
Employee-v2.java
```

Make changes to following files in `WEB-INF/views` from `changes`:

```
employee-form-v2.jsp
employee-details-v2.jsp
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.3 Drop down list <form:select> with options in form

Make changes to the following files in `com.workshop.mvc` in `src/main/java` from `changes`:

```
EmployeeController-v3.java
Employee-v3.java
```

Make changes to following files in `WEB-INF/views` from `changes`:

```
employee-form-v3.jsp
employee-details-v3.jsp
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.4 Drop down list <form:select> with options from POJO

We can make a slight modification so that the list of country options is provided from the Employee class as a HashMap rather than hardcoded in the employee form itself:

Make changes to the following files in `com.workshop.mvc` in `src/main/java` from `changes`:

```
Employee-v3-2.java
```

Make changes to following files in `WEB-INF/views` from `changes`:

```
employee-form-v3-2.jsp
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.5 Drop down list <form:select> with options from @Controller class

We can make a slight modification so that the list of country options is provided from the EmployeeController class as a HashMap. Notice that we place this HashMap into our model through the use of the @ModelAttribute annotation applied to a method. All methods with this annotation will get their return values placed into the Model automatically without an explicit `addAttribute` method call.

Make changes to the following files in `com.workshop.mvc` in `src/main/java` from `changes`:

```
Employee-v3-3.java
EmployeeController-v3-2.java
```

Make changes to following files in `WEB-INF/views` from `changes`:

```
employee-form-v3-3.jsp
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.6 Drop down list <form:select> with options from properties file

We can make a slight modification so that the list of country options is read from a properties file, then stored in a field in EmployeeController and then added to the Model to be extracted and used in the form itself:

Make changes to the following files in `com.workshop.mvc` in `src/main/java` from `changes`:

`EmployeeController-v3-3.java`

Make changes to following files in `WEB-INF/views` from `changes`:

`employee-form-v3-4.jsp`

Add the additional map property to `application.properties`

```
#Configuring Spring MVC
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

#Provides a context-root to deploy the app on
server.servlet.context-path=/FormTagSpringMVC

#Properties for the country options
country.options={ 'USA': 'United States of America', 'IND': 'India', 'CHN':
'China','AUS': 'Australia'}
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.7 Radio buttons <form:radiobuttons> with options from @Controller class

Make changes to the following files in `com.workshop.mvc` in `src/main/java` from `changes`:

```
EmployeeController-v4.java
Employee-v4.java
```

Make changes to following files in `WEB-INF/views` from `changes`:

```
employee-form-v4.jsp
employee-details-v4.jsp
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.8 Check boxes <form:checkboxes> with options from properties file

Make changes to the following files in `com.workshop.mvc` in `src/main/java` from `changes`:

```
EmployeeController-v5.java
Employee-v5.java
```

Make changes to following files in `WEB-INF/views` from `changes`:

```
employee-form-v5.jsp
employee-details-v5.jsp
```

Add the additional map property to `application.properties`

```
#Configuring Spring MVC
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

#Provides a context-root to deploy the app on
server.servlet.context-path=/FormTagSpringMVC

#Properties for the country options
country.options={ 'USA': 'United States of America', 'IND': 'India', 'CHN': 
'China','AUS': 'Australia'}

#Properties for the framework options
framework.options=SpringMVC,Struts,Laravel,NodeJS,Django
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.

2.9 Using ModelAndView vs Model

Make the following changes:

```
EmployeeController-v6.java
```

Run the app and access it at:

http://localhost:8080/FormTagSpringMVC/

Click on the employee details form link and complete the form that is returned. Check that the Employee field values are correctly set and displayed.