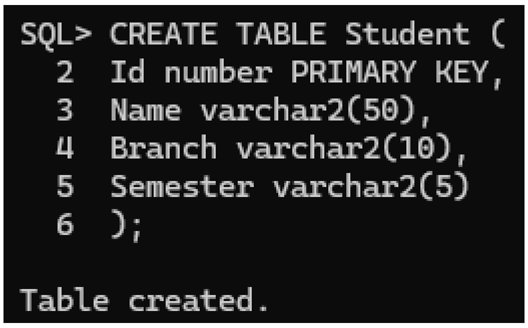
Name: - Shankar Singh Mahanty
Roll No. : CSE21238
Regd. No. : 2101020758
Branch : CSE Group : 3 Sem : 6th

EJAVA ASSIGNMENT : - 05

Q1. Develop a simple web application using Spring MVC framework. The application should include features such as user registration, login, and basic CRUD operations on a database.

Solution:

Create Table

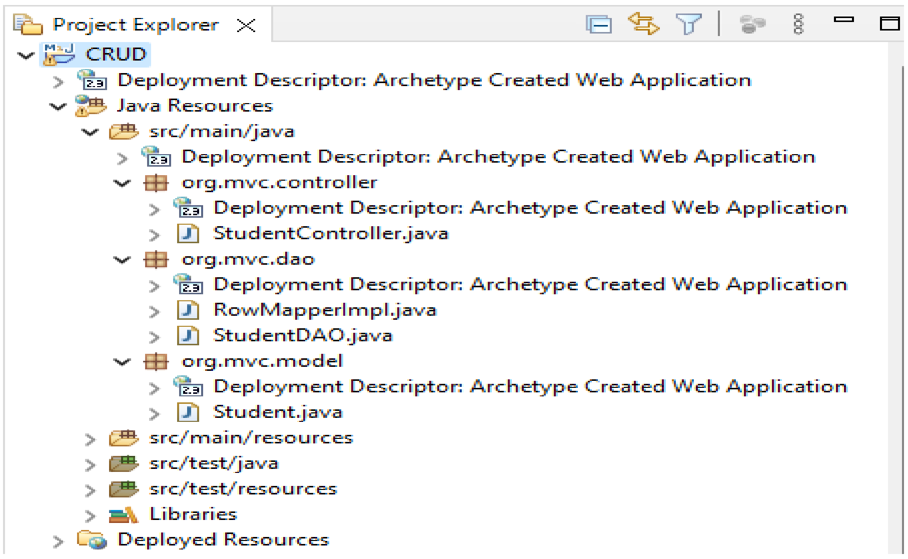


```
SQL> CREATE TABLE Student (
  2  Id number PRIMARY KEY,
  3  Name varchar2(50),
  4  Branch varchar2(10),
  5  Semester varchar2(5)
  6  );

Table created.
```

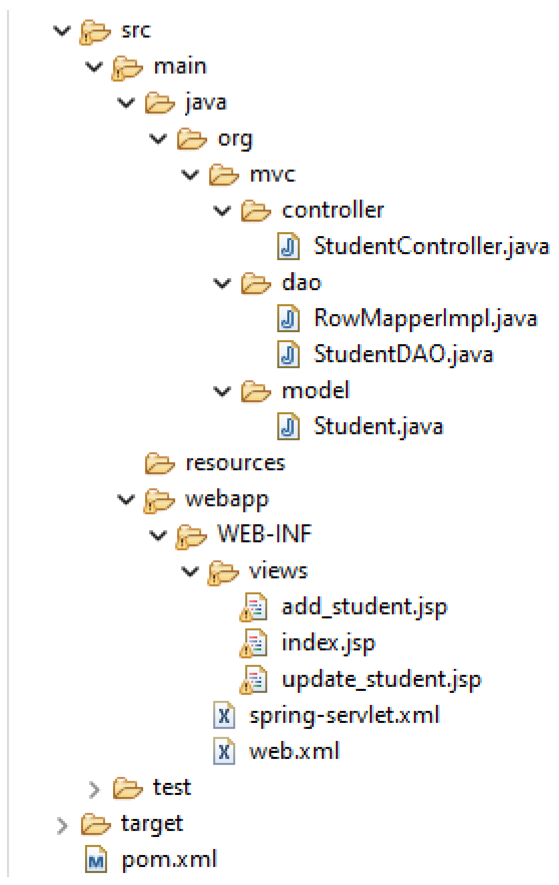
Create a Project in Spring STS

Create a Spring Project Go to File> New > Other > Search maven > Select Maven Project > Next > Search Filter org.apche.maven.archetypes/webapp > Next > Enter Group Id & Archetype id > Finish.

The project structure should like the following:



```
Project Explorer
v CRUD
  > Deployment Descriptor: Archetype Created Web Application
  v Java Resources
    v src/main/java
      > Deployment Descriptor: Archetype Created Web Application
      v org.mvc.controller
        > Deployment Descriptor: Archetype Created Web Application
        > StudentController.java
      v org.mvc.dao
        > Deployment Descriptor: Archetype Created Web Application
        > RowMapperImpl.java
        > StudentDAO.java
      v org.mvc.model
        > Deployment Descriptor: Archetype Created Web Application
        > Student.java
    > src/main/resources
    > src/test/java
    > src/test/resources
    > Libraries
  > Deployed Resources
```



```
src
  main
    java
      org
        mvc
          controller
            StudentController.java
          dao
            RowMapperImpl.java
            StudentDAO.java
          model
            Student.java
    resources
    webapp
      WEB-INF
        views
          add_student.jsp
          index.jsp
          update_student.jsp
        spring-servlet.xml
        web.xml
  test
target
pom.xml
```

Add the following dependencies into the pom.xml files

```xml
<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.34</version>
</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>4.0.1</version>
    <scope>provided</scope>
</dependency>

<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>

<!-- https://mvnrepository.com/artifact/com.oracle.database.jdbc/ojdbc10 -->
<dependency>
    <groupId>com.oracle.database.jdbc</groupId>
    <artifactId>ojdbc10</artifactId>
    <version>19.22.0.0</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.3.34</version>
</dependency>
```

Create a web.xml file

```
<!DOCTYPE web-app PUBLIC
"-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
    <display-name>Archetype Created Web Application</display-name>

    <!-- Configure dispatcher servlet  -->

    <servlet>
        <servlet-name>spring</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>/WEB-INF/spring-servlet.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>spring</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>
</web-app>
```

Create a spring-servlet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/spring-context.xsd">
    <!-- component scan -->
    <context:component-scan base-package="org.mvc"></context:component-scan>

    <bean
        class="org.springframework.web.servlet.view.InternalResourceViewResolver"
name="viewResolver">
        <property name="prefix" value="/WEB-INF/views/" />
        <property name="suffix" value=".jsp" />
    </bean>

    <!-- Initialize Data Source -->
    <bean id="dataSource"
        class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName"
            value="oracle.jdbc.driver.OracleDriver" />
        <property name="url" value="jdbc:oracle:thin:@localhost:1521:xe" />
        <property name="username" value="system" />
        <property name="password" value="Shan1506" />
    </bean>

    <bean id="jdbcTemplate"
        class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="dataSource"></property>
    </bean>

    <!-- Define StudentDAO bean -->
    <bean id="studentdao" class="org.mvc.dao.StudentDAO">
        <property name="jdbcTemplate" ref="jdbcTemplate"></property>
    </bean>
</beans>
```

Create a Model class – Student.java

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<%@ taglib prefix="f" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="ISO-8859-1">
        <title>Edit Student</title>
        <!-- CSS only -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
        <!-- JavaScript Bundle with Popper -->
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
    </head>

    <body style="background-color: #e2e2e2">
        <div class="container mt-5">
            <div class="card " style="width: 25rem; ">
                <div class="card-body">
                    <h5 class="card-title " align="center" >Edit Student</h5>
                    <f:form modelAttribute="student" action="/CRUD/update"
method="post">
                        <f:hidden path="id"/>
                        <div class="form-group">
                            <label for="name">Name</label>
                            <f:input path="name" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="branch">Branch</label>
                            <f:input path="branch" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="semester">Semester</label>
                            <f:input path="semester" class="form-control" />
                        </div><br>
                        <div class="form-group text-center">
                            <button type="submit" class="btn btn-warning">Update
Student</button>
                        </div>
                    </f:form>
                </div>
            </div>
        </div>
    </body>
</html>
```

Create a DAO class – StudentDAO.java

```
package org.mvc.dao;

import java.util.List;
import org.mvc.model.Student;
import org.springframework.jdbc.core.JdbcTemplate;

public class StudentDAO {

    private JdbcTemplate jdbcTemplate;

    public void setJdbcTemplate(JdbcTemplate jdbcTemplate) {
        this.jdbcTemplate = jdbcTemplate;
    }

    public int insert(Student student) {
        String sql = "INSERT INTO Student (id, name, branch, semester) VALUES (?, ?, ?,
?)";
        return jdbcTemplate.update(sql, student.getId(), student.getName(),
student.getBranch(), student.getSemester());
    }

    public int update(Student student) {
        String sql = "UPDATE Student SET name = ?, branch = ?, semester = ? WHERE id =
?";
        return jdbcTemplate.update(sql, student.getName(), student.getBranch(),
student.getSemester(), student.getId());
    }

    public int delete(int id) {
        String sql = "DELETE FROM Student WHERE id = ?";
        return jdbcTemplate.update(sql, id);
    }

    public Student getStudentById(int id) {
        String sql = "select * from student where id = '" + id + "'";
        return jdbcTemplate.queryForObject(sql, new RowMapperImpl());
    }


    public List<Student> getListOfStudents() {
        String sql = "SELECT * FROM Student";
        return jdbcTemplate.query(sql, new RowMapperImpl());
    }
}
```

<u>Create a class RowMapperImpl inside dao package to convert the resultset from the table to an object & use it in StudentDAO.java</u>

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<%@ taglib prefix="f" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="ISO-8859-1">
        <title>Edit Student</title>
        <!-- CSS only -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
```

```
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
        <!-- JavaScript Bundle with Popper -->
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
    </head>

    <body style="background-color: #e2e2e2">
        <div class="container mt-5">
            <div class="card " style="width: 25rem; ">
                <div class="card-body">
                    <h5 class="card-title " align="center" >Edit Student</h5>
                    <f:form modelAttribute="student" action="/CRUD/update"
method="post">
                        <f:hidden path="id"/>
                        <div class="form-group">
                            <label for="name">Name</label>
                            <f:input path="name" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="branch">Branch</label>
                            <f:input path="branch" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="semester">Semester</label>
                            <f:input path="semester" class="form-control" />
                        </div><br>
                        <div class="form-group text-center">
                            <button type="submit" class="btn btn-warning">Update
Student</button>
                        </div>

                    </f:form>

                </div>
            </div>
        </div>
    </body>
</html>
```

Create a Controller – StudentController

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<%@ taglib prefix="f" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
    <head>
```

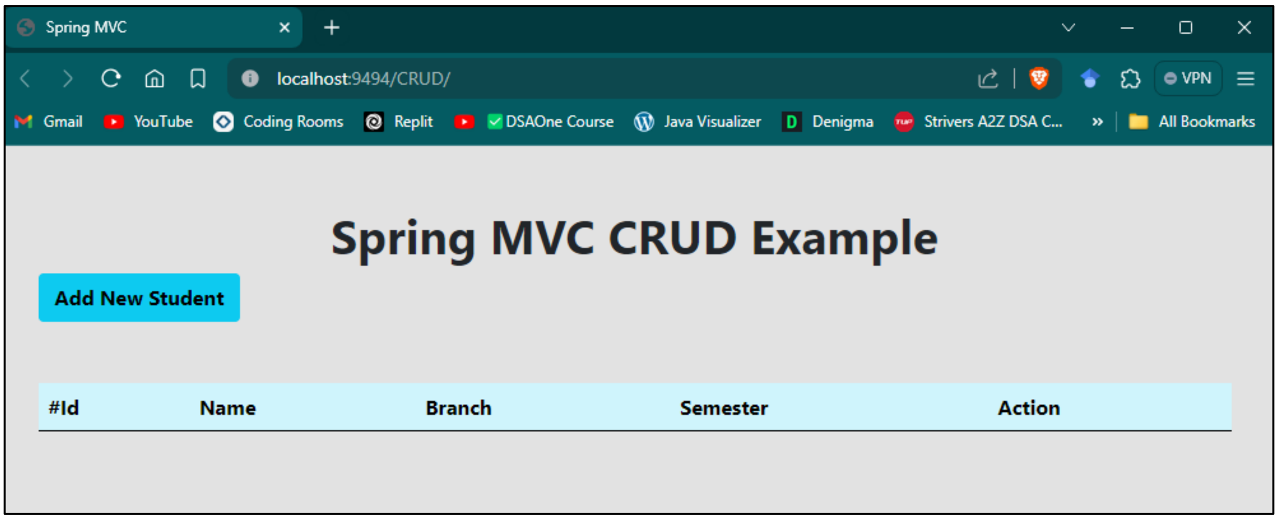
```
        <meta charset="ISO-8859-1">
        <title>Edit Student</title>
        <!-- CSS only -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
        <!-- JavaScript Bundle with Popper -->
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
    </head>

    <body style="background-color: #e2e2e2">
        <div class="container mt-5">
            <div class="card " style="width: 25rem; ">
                <div class="card-body">
                    <h5 class="card-title " align="center" >Edit Student</h5>
                    <f:form modelAttribute="student" action="/CRUD/update"
method="post">
                        <f:hidden path="id"/>
                        <div class="form-group">
                            <label for="name">Name</label>
                            <f:input path="name" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="branch">Branch</label>
                            <f:input path="branch" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="semester">Semester</label>
                            <f:input path="semester" class="form-control" />
                        </div><br>
                        <div class="form-group text-center">
                            <button type="submit" class="btn btn-warning">Update
Student</button>
                        </div>
                    </f:form>
                </div>
            </div>
        </div>
    </body>
</html>
```

Create views

All the views will be created in WEB-INF/views/ folder.

Index.jsp

```
<%@taglib uri="http://java.sun.com/jstl/core" prefix="c"%>
<%@ taglib prefix="f" uri="http://www.springframework.org/tags/form"%>
<html>
    <head>
        <!-- CSS only -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
        <!-- JavaScript Bundle with Popper -->
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
        <title>Spring MVC</title>
    </head>

    <body style="background-color: #e2e2e2">
        <div class="container mt-5">
            <h1 align="center" style="font-weight: bold; ">Spring MVC CRUD Example
</h1>
            <a href="studentform" class="btn btn-info" style="font-weight:
bold;">Add New Student</a>
            <table class="table mt-5 table-info ">
                <thead>
                    <tr>
                        <th scope="col">#Id</th>
                        <th scope="col">Name</th>
                        <th scope="col">Branch</th>
                        <th scope="col">Semester</th>
                        <th scope="col">Action
                        </th>
                    </tr>
                </thead>
                <tbody>
                    <c:forEach var="stud" items="${studentsList}">
                    <tr>
                        <th scope="row"><c:out value="${stud.id }"/></th>
                        <td><c:out value="${stud.name}"/></td>
                        <td><c:out value="${stud.branch}"/></td>
                        <td><c:out value="${stud.semester}"/></td>
                        <td><a href="editstudentform/<c:out value="${stud.id}"/>"
class="btn btn-warning">Edit</a>
                            <a href="deletestudent/<c:out value="${stud.id}"/>"
class="btn btn-danger">Delete</a>
                        </td>
                    </tr>
                    </c:forEach>
                </tbody>
            </table>
        </div>
    </body>
</html>
```



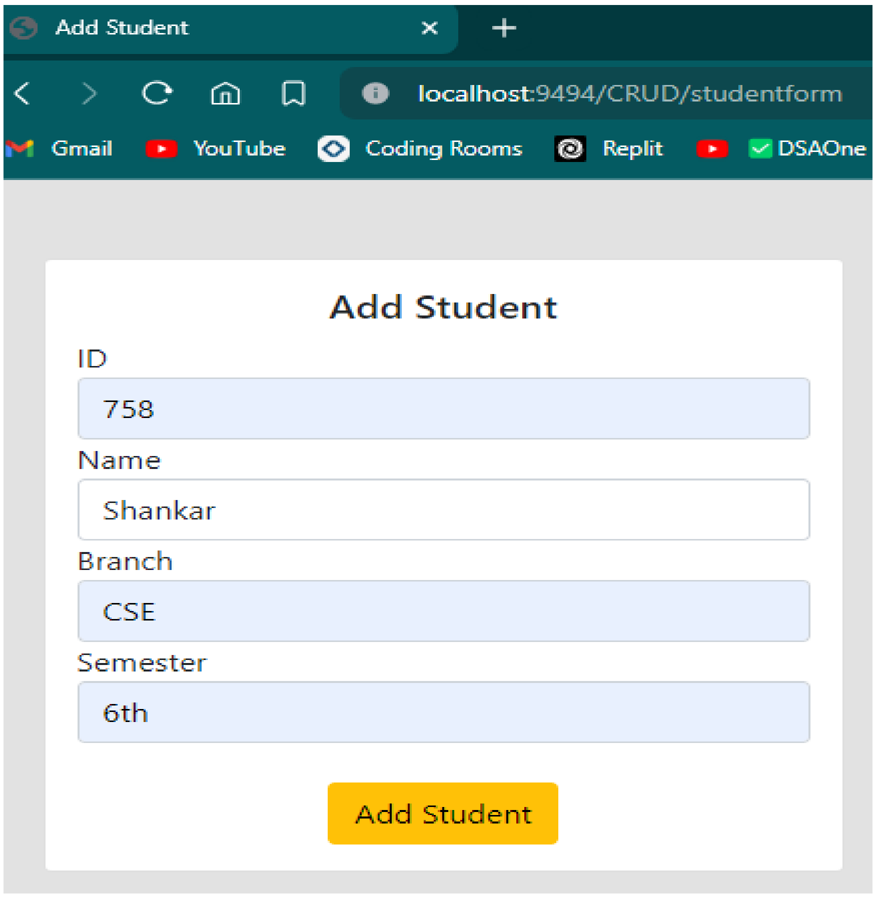
Spring MVC CRUD Example

Add New Student

#Id	Name	Branch	Semester	Action

Add_student.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="f" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
    <head>
    <meta charset="UTF-8">
    <title>Add Student</title>
    <!-- CSS only -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
    </head>

    <body style="background-color: #e2e2e2">
        <div class="container mt-5">
            <div class="card " style="width: 25rem;">
                <div class="card-body">
                    <h5 class="card-title" align="center">Add Student</h5>
                    <f:form modelAttribute="student" action="add" method="post">
                        <div class="form-group">
                            <label for="id">ID</label>
                            <f:input path="id" class="form-control" id="id" />
                        </div>
                        <div class="form-group">
                            <label for="name">Name</label>
                            <f:input path="name" class="form-control" id="name"
/>
                        </div>
                        <div class="form-group">
                            <label for="branch">Branch</label>
                            <f:input path="branch" class="form-control"
id="branch" />
                        </div>
                        <div class="form-group">
                            <label for="semester">Semester</label>
                            <f:input path="semester" class="form-control"
id="semester" />
                        </div><br>
                        <div class="form-group text-center">
                            <button type="submit" class="btn btn-warning">Add
Student</button>
                        </div>
                    </f:form>
                </div>
            </div>
        </div>
    </body>
</html>
```



Add Student

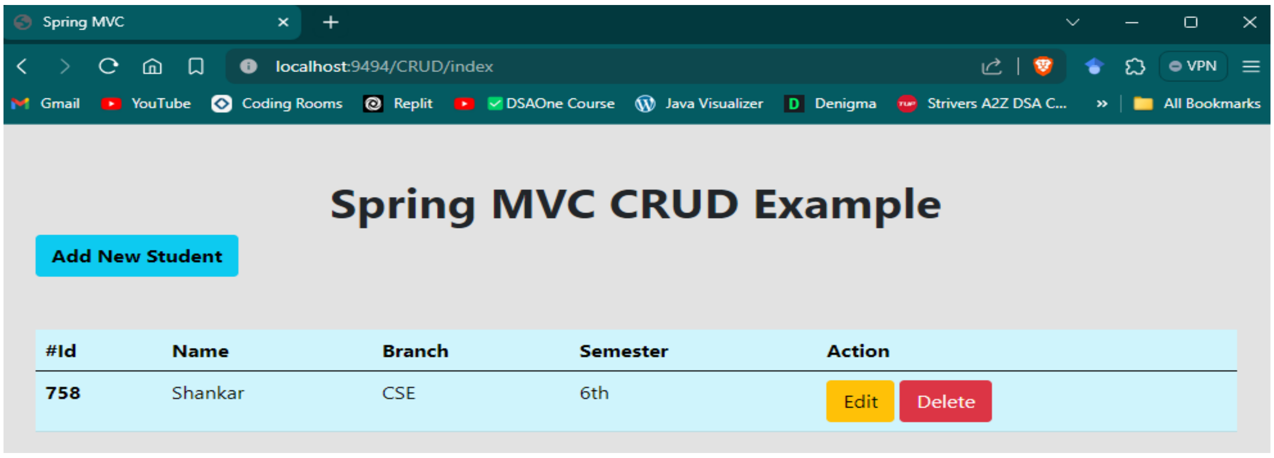
ID
758

Name
Shankar

Branch
CSE

Semester
6th

Add Student



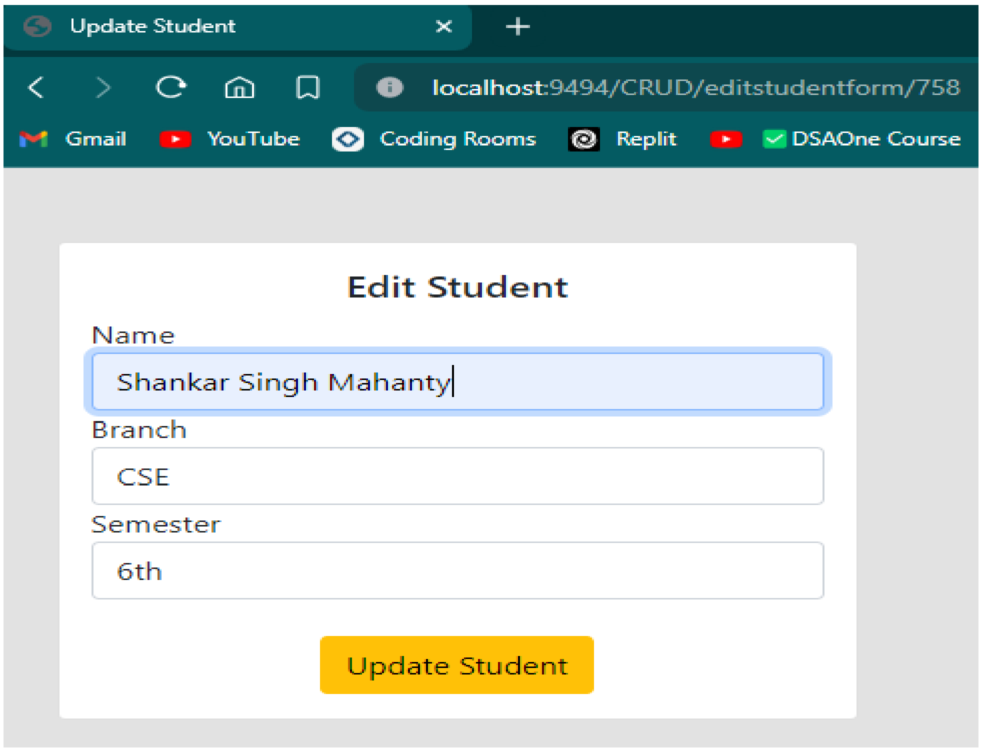
Spring MVC CRUD Example

Add New Student

#Id	Name	Branch	Semester	Action
758	Shankar	CSE	6th	Edit Delete

Update_student.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<%@ taglib prefix="f" uri="http://www.springframework.org/tags/form"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="ISO-8859-1">
        <title>Edit Student</title>
        <!-- CSS only -->
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
        <!-- JavaScript Bundle with Popper -->
        <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
    </head>

    <body style="background-color: #e2e2e2">
        <div class="container mt-5">
            <div class="card " style="width: 25rem; ">
                <div class="card-body">
                    <h5 class="card-title " align="center" >Edit Student</h5>
                    <f:form modelAttribute="student" action="/CRUD/update"
method="post">
                        <f:hidden path="id"/>
                        <div class="form-group">
                            <label for="name">Name</label>
                            <f:input path="name" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="branch">Branch</label>
                            <f:input path="branch" class="form-control" />
                        </div>
                        <div class="form-group">
                            <label for="semester">Semester</label>
                            <f:input path="semester" class="form-control" />
                        </div><br>
                        <div class="form-group text-center">
                            <button type="submit" class="btn btn-warning">Update
Student</button>
                        </div>
                    </f:form>
                </div>
            </div>
        </div>
    </body>
</html>
```



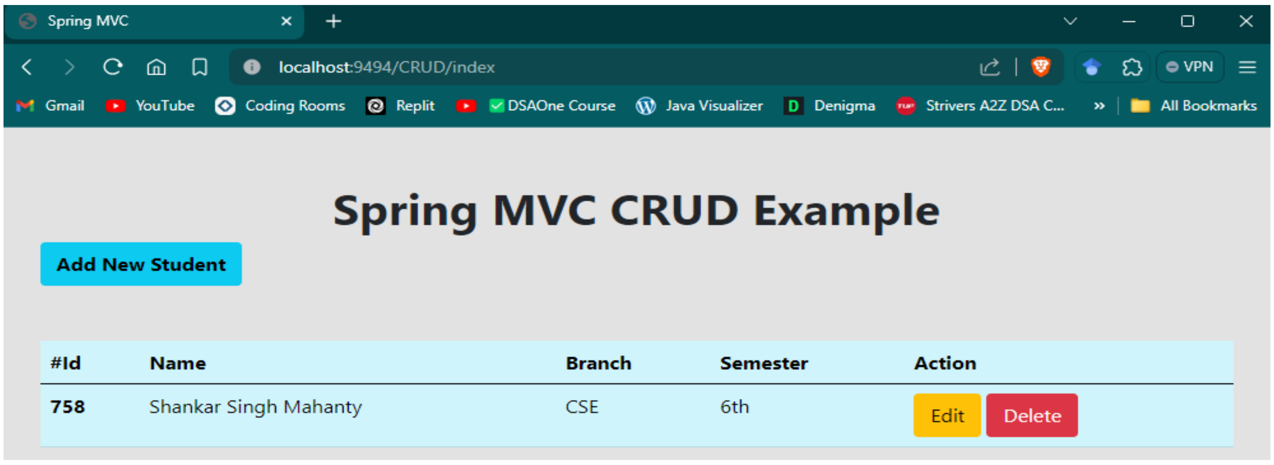
Edit Student

Name: Shankar Singh Mahanty

Branch: CSE

Semester: 6th

Update Student



Spring MVC CRUD Example

Add New Student

#Id	Name	Branch	Semester	Action
758	Shankar Singh Mahanty	CSE	6th	Edit Delete

Q2. Create a small project to demonstrate dependency injection in Spring. Use both constructor injection and setter injection to inject dependencies into various components of the application.

Solution: -

- Using Setter Method

  Create a Class Person having a name attribute/property and a setter method.

```
package com.Assignment5.SpringDependencyInjection;

public class Person {

	private String name;

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}
	public void display() {
		System.out.println("Name of the person is:: " +
name);
	}
}
```

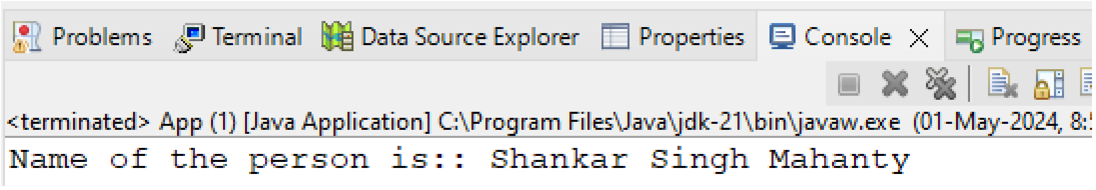
  Create config.xml file with <property> element to invoke setter method.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="
	http://www.springframework.org/schema/context"
	xmlns:p="http://www.springframework.org/schema/p"
	xsi:schemaLocation="http://www.springframework.org/schema
/beans
	http://www.springframework.org/schema/beans/spring-
beans.xsd
	http://www.springframework.org/schema/context
	http://www.springframework.org/schema/context/spring-
context.xsd">

	<bean id="person"
class="com.Assignment5.SpringDependencyInjection.Person">
    <property name="name" value="Shankar"></property>
	</bean>
</beans>
```

In App.java create the Object of Application Context and call the display() method.

```
package com.Assignment5.SpringDependencyInjection;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationCon
text;

public class App
{
    public static void main( String[] args )
    {
     ApplicationContext context = new
ClassPathXmlApplicationContext("config.xml");
        Person p1 = context.getBean("person",Person.class);
        p1.display();
    }
}
```

Now, the value will be set and you will get the following Output.



```
<terminated> App (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (01-May-2024, 8:
Name of the person is:: Shankar Singh Mahanty
```

- <u>Using Constructor Injection</u>

  Create a class Person with a name & city attribute.

```
package com.Assignment5.SpringDependencyInjection;

public class Person {

     private String name, city;

     public Person(String name, String city) {
          this.name = name;
          this.city = city;
     }

     public void display() {
           System.out.println("Name of person is :: " + name +
"\nCity of person is :: " + city);
     }
}
```

Create config.xml file with <constructor-arg> property

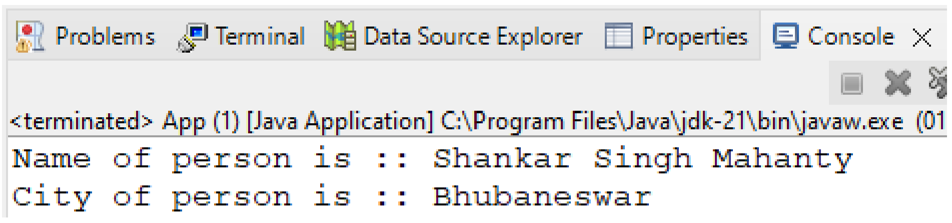
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
```

```
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="
	http://www.springframework.org/schema/context"
	xmlns:p="http://www.springframework.org/schema/p"
	xsi:schemaLocation="http://www.springframework.org/schema
/beans
	http://www.springframework.org/schema/beans/spring-
beans.xsd
	http://www.springframework.org/schema/context
	http://www.springframework.org/schema/context/spring-
context.xsd">

	<bean id="person"
class="com.Assignment5.SpringDependencyInjection.Person">
		<constructor-arg name="name" value="Shankar Singh
Mahanty" ></constructor-arg>
		<constructor-arg name="city" value="Bhubaneswar"
></constructor-arg>
	</bean>

</beans>
```

In App.java create the Object of Application Context and call the display() method.

```
package com.Assignment5.SpringDependencyInjection;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationCon
text;

public class App
{
    public static void main( String[] args )
    {
     ApplicationContext context = new
ClassPathXmlApplicationContext("config.xml");
        Person p1 = context.getBean("person",Person.class);
        p1.display();
    }
}
```

Output: -

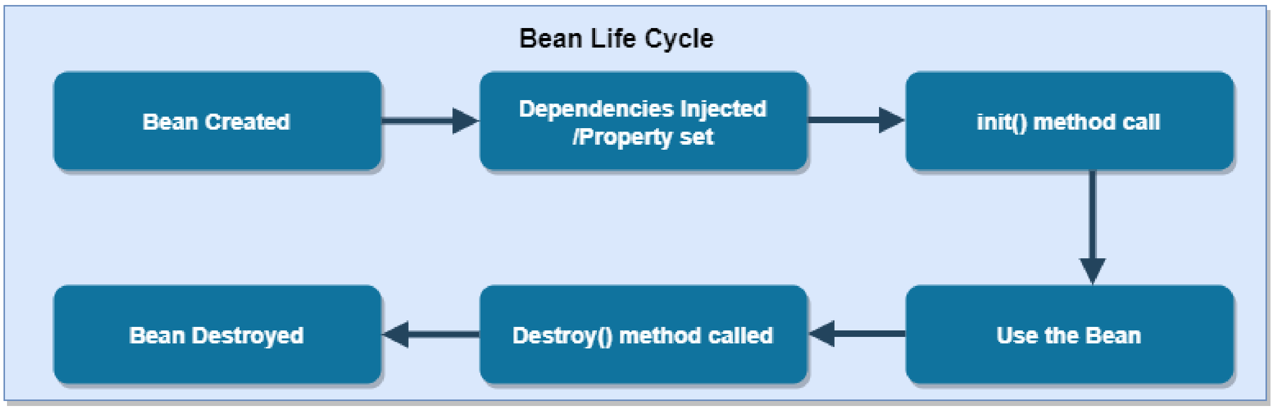

Problems Terminal Data Source Explorer Properties Console

```
<terminated> App (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (01
Name of person is :: Shankar Singh Mahanty
City of person is :: Bhubaneswar
```

Q3. Explain the phases of the bean lifecycle in a Spring bean factory. Describe each phase in detail and provide examples to illustrate the lifecycle transitions.

Solution: -

The Lifecycle of any bean means how the bean is created, how it goes from one stage to another, what are the stages involved from creation to destruction of any object or bean. Beans are nothing but objects in Spring. IOC container in Spring is responsible for creating objects from the configuration metadata that we supply in a config file. The container is responsible to manage the overall life cycle of the Spring beans i.e from the creation of a bean to its destruction. Following is the diagrammatical representation of Spring Bean Life cycle process flow.



- Firstly, the Spring Container creates the object of the bean. Here, Bean gets instantiated.
- Then, the Dependencies are Injected using XML or annotation-based configuration.
- Now, the init() method is called where the initialization is done. for eg., if you have any connection to initialize you can initialize them inside this method.
- Now, we read the bean and use them and call the respective methods.
- At last, the destroy() method is called for some clean-up process and then the beans are destroyed.

Thus, this is the process flow of Bean Life Cycle in Spring Bean. The important thing here is that the init() method is called after the properties are set i.e after injecting the dependencies.

Example to understand flow of Bean Life Cycle

There are 3 ways in which we can implement Bean Life Cycle:

- By using XML Based Configuration
- By using Annotation Based Configuration
- By using Java Based Configuration

In this example, we will implement using XML Based Configuration.

Create a Spring Project Go to File> New > Other > Search maven > Select Maven Project > Next > Search Filter org.apche.maven.archetypes/webapp > Next > Enter Group Id & Archetype id > Finish.

- Create a class Animal having a name attribute and a setter & getter method.
- Add two methods of the life cycle: init() method and destroy() methods.

Animal.java

```java
package com.Assignment5.BeanLifeCycle;

public class Animal {

    private String name;

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void show() {
        System.out.println("Name of the Animal is : " + name);
    }
    public void init() {
        System.out.println("<----------Init method is called---------->");
    }
    public void destroy() {
        System.out.println("<----------Destroy method is called----------->");
    }
}
```

create a beans.xml file with <bean> element having init-method and destroy-method.

beans.xml

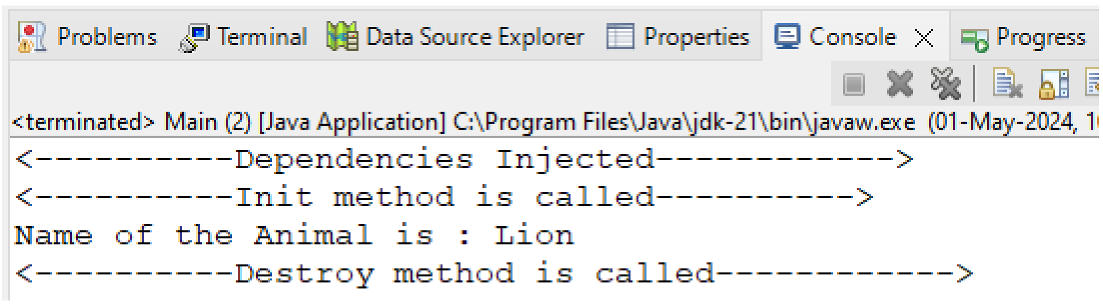
```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        https://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="animal" class="com.Assignment5.BeanLifeCycle.Animal" init-
method="init" destroy-method="destroy">
        <property name="name" value="Lion"></property>
    </bean>
</beans>
```

Create an App.java file. In which, we will use AbstractApplicationContext Interface and will call the registerShutdownHook() method to call destroy() method.

App.java

```
package com.Assignment5.BeanLifeCycle;

import org.springframework.context.support.AbstractApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {
    public static void main(String[] args) {
        AbstractApplicationContext context = new
ClassPathXmlApplicationContext("beans.xml");
        Animal a1= context.getBean("animal",Animal.class);
        a1.show();
        //When I want to call destroy()
        context.registerShutdownHook();
    }
}
```

<u>Output: -</u>

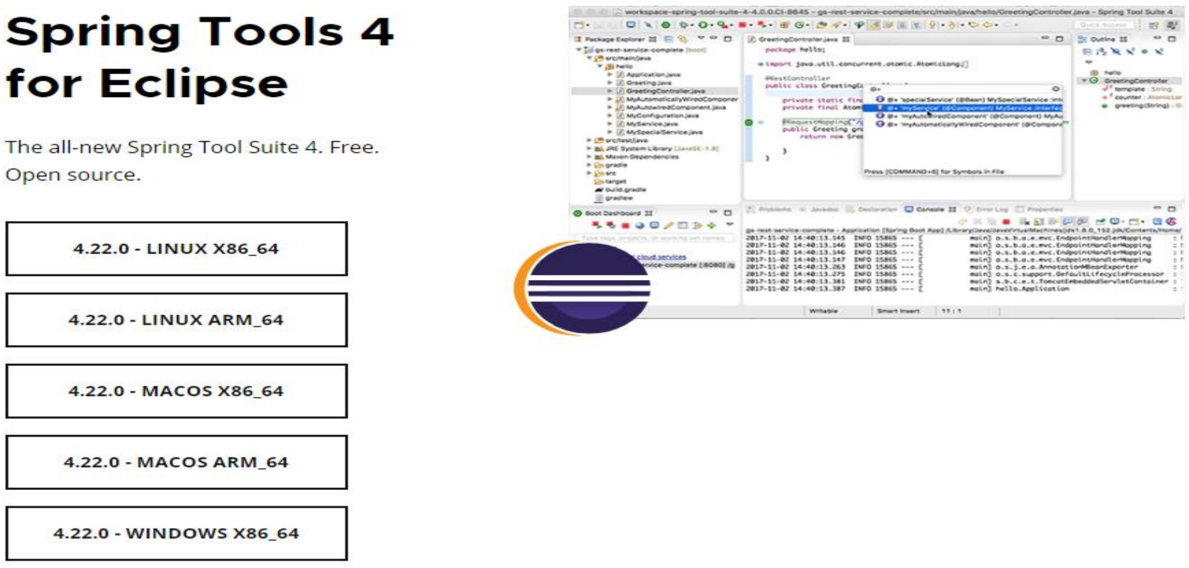


```
<terminated> Main (2) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (01-May-2024, 1
<----------Dependencies Injected------------>
<----------Init method is called---------->
Name of the Animal is : Lion
<----------Destroy method is called------------>
```

Q4. Implement authentication and authorization using Spring Security. Develop a web application where users can sign up, log in, and access specific resources based on their roles and permissions.

<u>Solution: -</u>

<u>Install STS 4 for eclipse</u>



Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4. Free. Open source.

4.22.0 - LINUX X86_64

4.22.0 - LINUX ARM_64

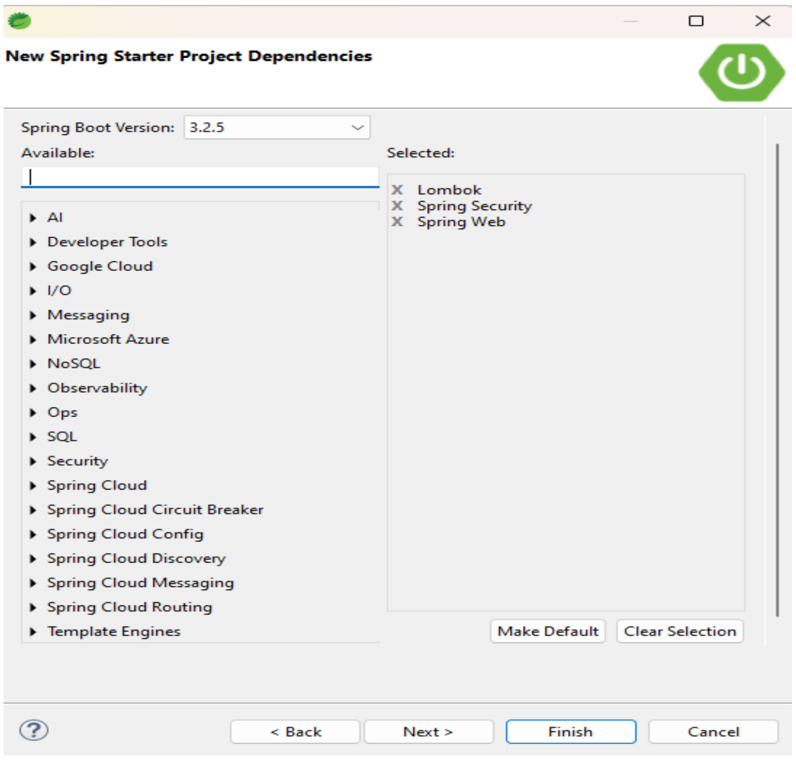
4.22.0 - MACOS X86_64

4.22.0 - MACOS ARM_64

4.22.0 - WINDOWS X86_64

Unlike old style of web.xml and spring xml configurations, we will use “In-Memory Store” option to store and manage User Credentials in our spring application according to the newer security module versions and features.

<u>Add the following dependencies</u>



New Spring Starter Project Dependencies

Spring Boot Version: 3.2.5

Available:

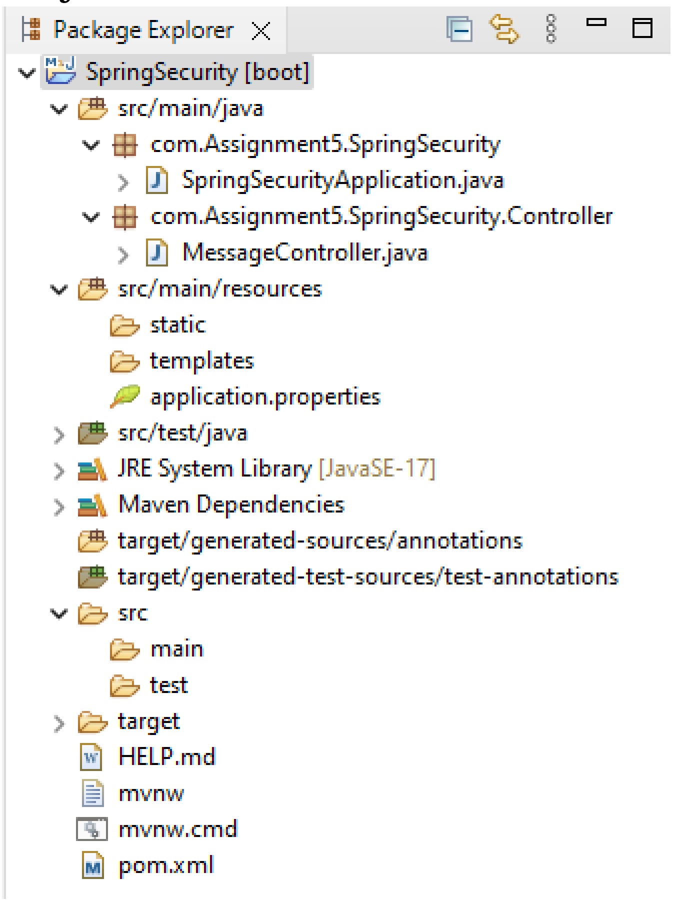
- AI
- Developer Tools
- Google Cloud
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Observability
- Ops
- SQL
- Security
- Spring Cloud
- Spring Cloud Circuit Breaker
- Spring Cloud Config
- Spring Cloud Discovery
- Spring Cloud Messaging
- Spring Cloud Routing
- Template Engines

Selected:

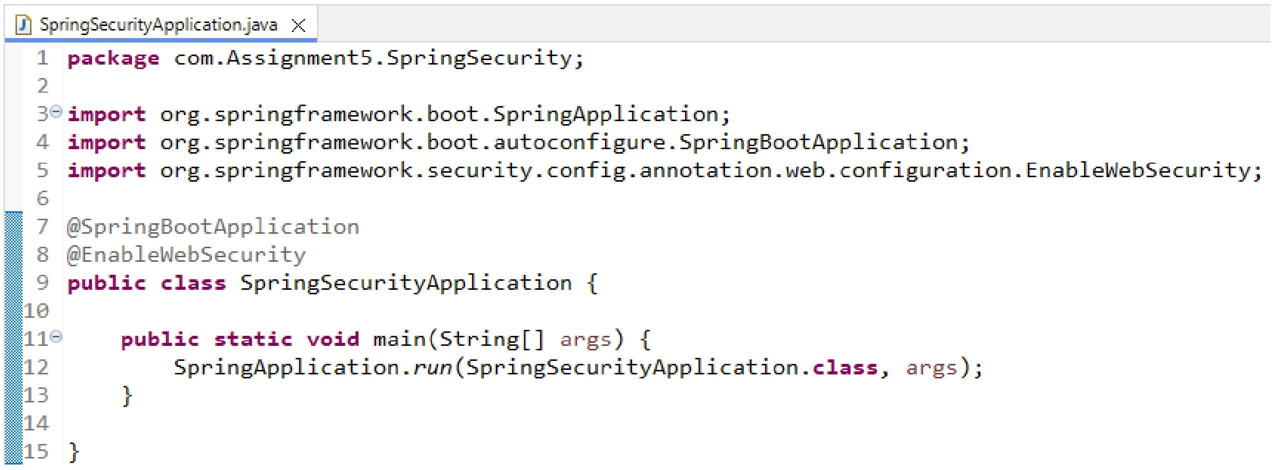
- X Lombok
- X Spring Security
- X Spring Web

- <u>Default Security</u>

  Project Structure

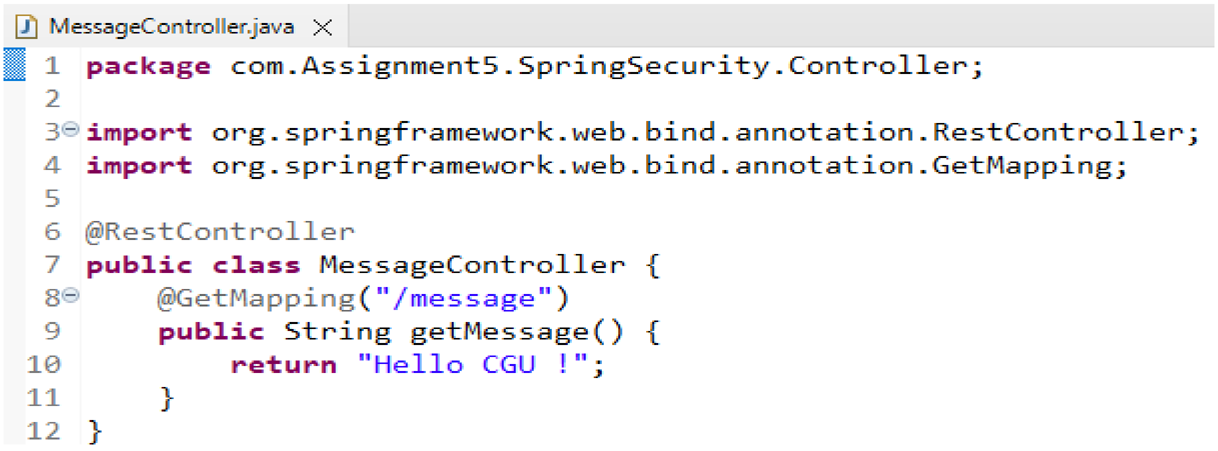


```
Package Explorer
SpringSecurity [boot]
  src/main/java
    com.Assignment5.SpringSecurity
      SpringSecurityApplication.java
    com.Assignment5.SpringSecurity.Controller
      MessageController.java
  src/main/resources
    static
    templates
    application.properties
  src/test/java
  JRE System Library [JavaSE-17]
  Maven Dependencies
  target/generated-sources/annotations
  target/generated-test-sources/test-annotations
  src
    main
    test
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
```



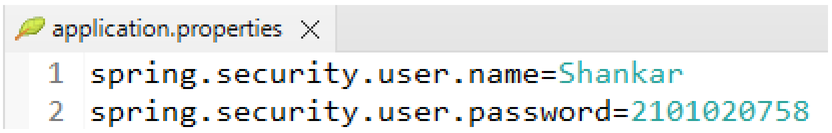
SpringSecurityApplication.java

```java
package com.Assignment5.SpringSecurity;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

@SpringBootApplication
@EnableWebSecurity
public class SpringSecurityApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringSecurityApplication.class, args);
    }

}
```



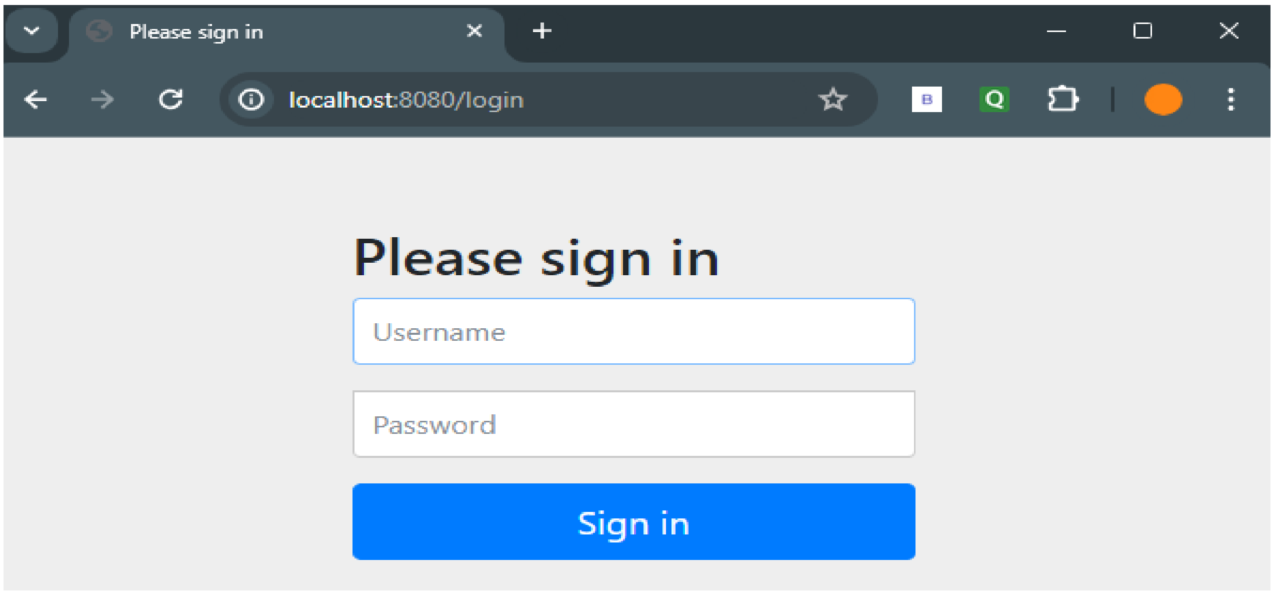
MessageController.java

```java
package com.Assignment5.SpringSecurity.Controller;

import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.bind.annotation.GetMapping;

@RestController
public class MessageController {
    @GetMapping("/message")
    public String getMessage() {
        return "Hello CGU !";
    }
}
```


application.properties

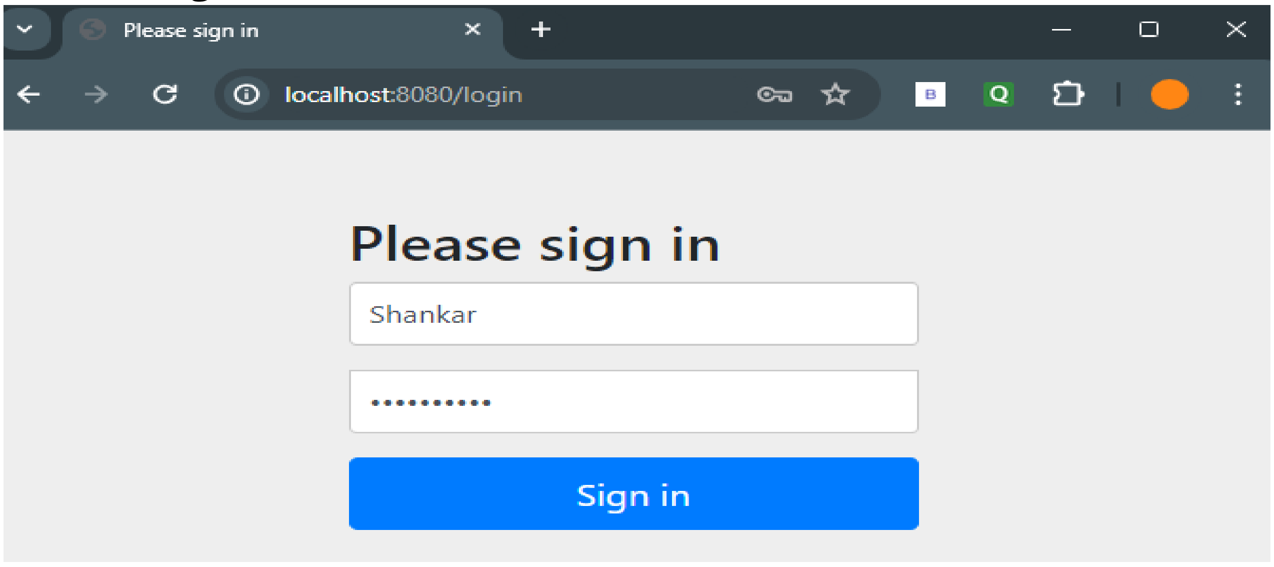
```
spring.security.user.name=Shankar
spring.security.user.password=2101020758
```

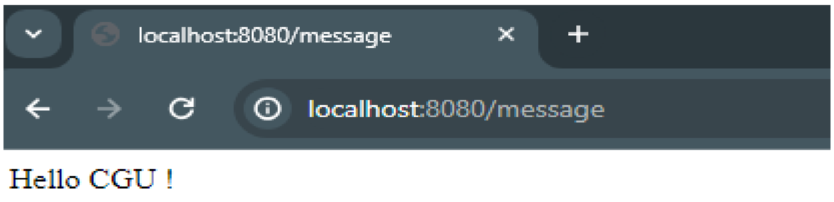
Output: -



If trying to hit the endpoint (/message) along with localhost:8080/ it automatically redirects to /login as the part of security without authentication cannot view the message.

After giving the correct credentials, it will sign up and redirect to show the message.

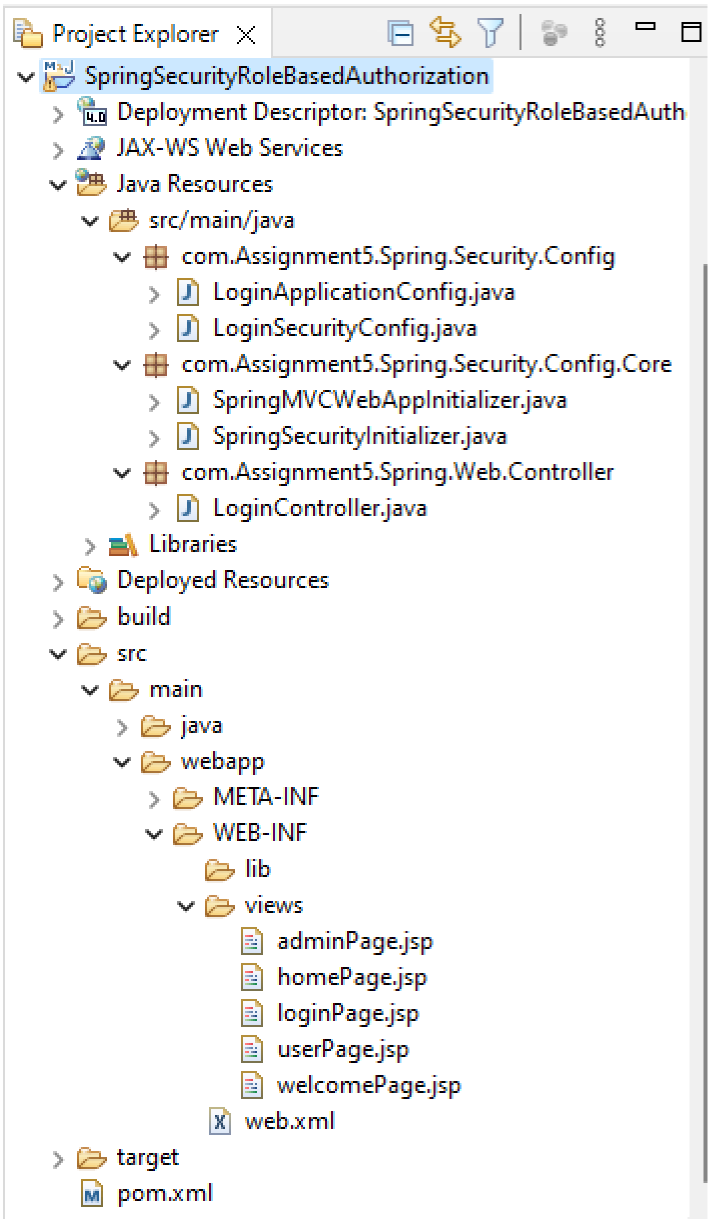




```
Hello CGU !
```

- <u>Role based Authentication to Authorize specific access</u>

  Create a simple spring web maven project with the following project structure



```
Project Explorer
SpringSecurityRoleBasedAuthorization
  Deployment Descriptor: SpringSecurityRoleBasedAuth
  JAX-WS Web Services
  Java Resources
    src/main/java
      com.Assignment5.Spring.Security.Config
        LoginApplicationConfig.java
        LoginSecurityConfig.java
      com.Assignment5.Spring.Security.Config.Core
        SpringMVCWebAppInitializer.java
        SpringSecurityInitializer.java
      com.Assignment5.Spring.Web.Controller
        LoginController.java
    Libraries
  Deployed Resources
  build
  src
    main
      java
      webapp
        META-INF
        WEB-INF
          lib
          views
            adminPage.jsp
            homePage.jsp
            loginPage.jsp
            userPage.jsp
            welcomePage.jsp
          web.xml
  target
  pom.xml
```

<u>pom.xml file</u>

```
<?xml version="1.0" encoding="UTF-8"?>

<project
   xsi:schemaLocation="https://maven.apache.org/POM/4.0.0
   https://maven.apache.org/xsd/maven-4.0.0.xsd"
   xmlns="https://maven.apache.org/POM/4.0.0"
   xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance">

      <modelVersion>4.0.0</modelVersion>
      <groupId>com.journaldev</groupId>
      <artifactId>SpringMVCSecruityMavenApp</artifactId>
      <packaging>war</packaging>
      <version>1.0</version>

      <properties>
```

```
          <java.version>1.8</java.version>
          <spring.version>4.0.2.RELEASE</spring.version>

<spring.security.version>4.0.2.RELEASE</spring.security.version>
          <servlet.api.version>3.1.0</servlet.api.version>
          <jsp.api.version>2.2</jsp.api.version>
          <jstl.version>1.2</jstl.version>
      </properties>

      <dependencies>
            <dependency>
                  <groupId>org.springframework</groupId>
                  <artifactId>spring-core</artifactId>
                  <version>${spring.version}</version>
            </dependency>
            <dependency>
                  <groupId>org.springframework</groupId>
                  <artifactId>spring-web</artifactId>
                  <version>${spring.version}</version>
            </dependency>
            <dependency>
                  <groupId>org.springframework</groupId>
                  <artifactId>spring-webmvc</artifactId>
                  <version>${spring.version}</version>
            </dependency>
            <dependency>
                  <groupId>org.springframework.security</groupId>
                  <artifactId>spring-security-web</artifactId>
                  <version>${spring.security.version}</version>
            </dependency>
            <dependency>
                  <groupId>org.springframework.security</groupId>
                  <artifactId>spring-security-config</artifactId>
                  <version>${spring.security.version}</version>
            </dependency>
            <dependency>
                  <groupId>javax.servlet</groupId>
                  <artifactId>javax.servlet-api</artifactId>
                  <version>${servlet.api.version}</version>
            </dependency>
            <dependency>
                  <groupId>javax.servlet.jsp</groupId>
                  <artifactId>jsp-api</artifactId>
                  <version>${jsp.api.version}</version>
            </dependency>
            <dependency>
                  <groupId>jstl</groupId>
                  <artifactId>jstl</artifactId>
                  <version>${jstl.version}</version>
            </dependency>
      </dependencies>

      <build>
          <finalName>SpringMVCSecruityMavenApp</finalName>
          <plugins>
            <plugin>
                 <groupId>org.apache.maven.plugins</groupId>
                 <artifactId>maven-compiler-plugin</artifactId>
                 <version>3.1</version>
                 <configuration>
```

```
                    <source>${java.version}</source>
                    <target>${java.version}</target>
                   </configuration>
             </plugin>
             <plugin>
                    <groupId>org.apache.maven.plugins</groupId>
                    <artifactId>maven-war-plugin</artifactId>
                    <configuration>
                       <failOnMissingWebXml>false</failOnMissingWebXml>
                    </configuration>
             </plugin>
           </plugins>
      </build>
</project>
```

LoginApplicationConfig.java

```
package com.Assignment5.Spring.Security.Config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Import;
import
org.springframework.web.servlet.config.annotation.EnableWebMvc;
import
org.springframework.web.servlet.view.InternalResourceViewResolver;
import org.springframework.web.servlet.view.JstlView;

@EnableWebMvc
@Configuration
@ComponentScan({ "com.Assignment5.Spring.*" })
@Import(value = { LoginSecurityConfig.class })
public class LoginApplicationConfig {
    @Bean
    public InternalResourceViewResolver viewResolver() {
        InternalResourceViewResolver viewResolver = new
InternalResourceViewResolver();
        viewResolver.setViewClass(JstlView.class);
        viewResolver.setPrefix("/WEB-INF/views/");
        viewResolver.setSuffix(".jsp");
        return viewResolver;
    }
}
```

LoginSecurityConfig.java

```
package com.Assignment5.Spring.Security.Config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Configuration;
```

```java
import
org.springframework.security.config.annotation.authentication.build
ers.AuthenticationManagerBuilder;
import
org.springframework.security.config.annotation.web.builders.HttpSec
urity;
import
org.springframework.security.config.annotation.web.configuration.En
ableWebSecurity;
import
org.springframework.security.config.annotation.web.configuration.We
bSecurityConfigurerAdapter;

@Configuration
@EnableWebSecurity
public class LoginSecurityConfig extends
WebSecurityConfigurerAdapter {

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder
authenticationMgr) throws Exception {
        authenticationMgr.inMemoryAuthentication()
            .withUser("shanuser").password("shan@758").authorities(
"ROLE_USER")
            .and()
            .withUser("Shanadmin").password("Shan@758").authorities
("ROLE_USER","ROLE_ADMIN");
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {


        http.authorizeRequests()
            .antMatchers("/homePage").access("hasRole('ROLE_USER')
or hasRole('ROLE_ADMIN')")
            .antMatchers("/userPage").access("hasRole('ROLE_USER')"
)
            .antMatchers("/adminPage").access("hasRole('ROLE_ADMIN'
)")
            .and()
                .formLogin().loginPage("/loginPage")
                .defaultSuccessUrl("/homePage")
                .failureUrl("/loginPage?error")
                .usernameParameter("username").passwordParameter("p
assword")
            .and()
                .logout().logoutSuccessUrl("/loginPage?logout");
    }
}
```

SpringMVCWebAppInitializer.java

```java
package com.Assignment5.Spring.Security.Config.Core;

import
org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherSer
vletInitializer;
import com.Assignment5.Spring.Security.Config.LoginApplicationConfig;

public class SpringMVCWebAppInitializer extends
AbstractAnnotationConfigDispatcherServletInitializer {

    @Override
    protected Class<?>[] getRootConfigClasses() {
        return new Class[] { LoginApplicationConfig.class };
    }

    @Override
    protected Class<?>[] getServletConfigClasses() {
        return null;
    }

    @Override
    protected String[] getServletMappings() {
        return new String[] { "/" };
    }
}
```

SpringSecurityInitializer.java

```java
package com.Assignment5.Spring.Security.Config.Core;

import
org.springframework.security.web.context.AbstractSecurityWebApplicationInitia
lizer;

public class SpringSecurityInitializer extends
AbstractSecurityWebApplicationInitializer {

}
```

LoginController.java

```java
package com.Assignment5.Spring.Web.Controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

@Controller
public class LoginController {
    @RequestMapping(value = { "/"}, method = RequestMethod.GET)
    public ModelAndView welcomePage() {
        ModelAndView model = new ModelAndView();
        model.setViewName("welcomePage");
        return model;
    }

    @RequestMapping(value = { "/homePage"}, method = RequestMethod.GET)
    public ModelAndView homePage() {
        ModelAndView model = new ModelAndView();
        model.setViewName("homePage");
        return model;
    }

    @RequestMapping(value = {"/userPage"}, method = RequestMethod.GET)
    public ModelAndView userPage() {
        ModelAndView model = new ModelAndView();
        model.setViewName("userPage");
        return model;
    }

    @RequestMapping(value = {"/adminPage"}, method = RequestMethod.GET)
    public ModelAndView adminPage() {
        ModelAndView model = new ModelAndView();
        model.setViewName("adminPage");
        return model;
    }

    @RequestMapping(value = "/loginPage", method = RequestMethod.GET)
    public ModelAndView loginPage(@RequestParam(value = "error",required =
false) String error,
    @RequestParam(value = "logout", required = false) String logout) {

        ModelAndView model = new ModelAndView();
        if (error != null) {
            model.addObject("error", "Invalid Credentials provided.");
        }

        if (logout != null) {
            model.addObject("message", "Logged out from Ejava Assignments
successfully.");
        }

        model.setViewName("loginPage");
        return model;
    }

}
```

welcomePage.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-
8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Welcome Page</title>
</head>
<body>
<h3>Welcome to Ejava Assignments</h3>
<a href="${pageContext.request.contextPath}/loginPage">Login
to Assignments</a>
</body>
</html>
```

loginPage.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-
8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Login Page</title>
</head>
<body onload='document.loginForm.username.focus();'>
     <h3>Ejava Assignments</h3>

     <c:if test="${not empty
error}"><div>${error}</div></c:if>
     <c:if test="${not empty
message}"><div>${message}</div></c:if>

     <form name='login' action="<c:url value='/loginPage' />"
method='POST'>
          <table>
               <tr>
                    <td>UserName:</td>
                    <td><input type='text' name='username'
value=''></td>
               </tr>
               <tr>
                    <td>Password:</td>
                    <td><input type='password' name='password'
/></td>
               </tr>
               <tr>
                    <td colspan='2'><input name="submit"
type="submit" value="submit" /></td>
               </tr>
          </table>
          <input type="hidden" name="${_csrf.parameterName}"
value="${_csrf.token}" />
     </form>
</body>
</html>
```

homePage.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-
8"
    pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Home Page</title>
</head>
<body>
<a href="${pageContext.request.contextPath}/userPage">Shan as
User</a> | <a
href="${pageContext.request.contextPath}/adminPage">Shan as
Admin</a> | <a
href="javascript:document.getElementById('logout').submit()">L
ogout</a>
<h3>Welcome to Ejava Assignments</h3>
<ul>
     <li>CRUD</li>
     <li>Spring Boot</li>
     <li>Custom Tags</li>
     <li>Spring Security</li>
</ul>

<c:url value="/logout" var="logoutUrl" />
<form id="logout" action="${logoutUrl}" method="post" >
  <input type="hidden" name="${_csrf.parameterName}"
value="${_csrf.token}" />
</form>
</body>
</html>
```

userPage.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-
8"
    pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>User Page</title>
</head>
<body>
```

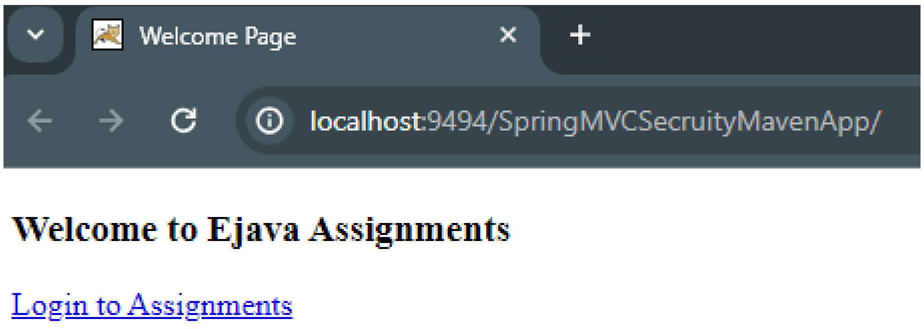
```
<h3>Welcome to Ejava Assignments</h3>
<h3>You are inside User Page</h3>

<c:url value="/logout" var="logoutUrl" />
<form id="logout" action="${logoutUrl}" method="post" >
  <input type="hidden" name="${_csrf.parameterName}"
value="${_csrf.token}" />
</form>
<c:if test="${pageContext.request.userPrincipal.name !=
null}">
     <a
href="javascript:document.getElementById('logout').submit()">L
ogout</a>
</c:if>
</body>
</html>
```

adminPage.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-
8"
    pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Admin Page</title>
</head>
<body>
<h3>Welcome to Ejava Assignments</h3>
<h3> You are inside Admin Page</h3>

<c:url value="/logout" var="logoutUrl" />
<form id="logout" action="${logoutUrl}" method="post" >
  <input type="hidden" name="${_csrf.parameterName}"
value="${_csrf.token}" />
</form>
<c:if test="${pageContext.request.userPrincipal.name !=
null}">
     <a
href="javascript:document.getElementById('logout').submit()">L
ogout</a>
</c:if>
</body>
</html>
```

Outputs: -

Run on server : It will access default Application welcome page as shown below:

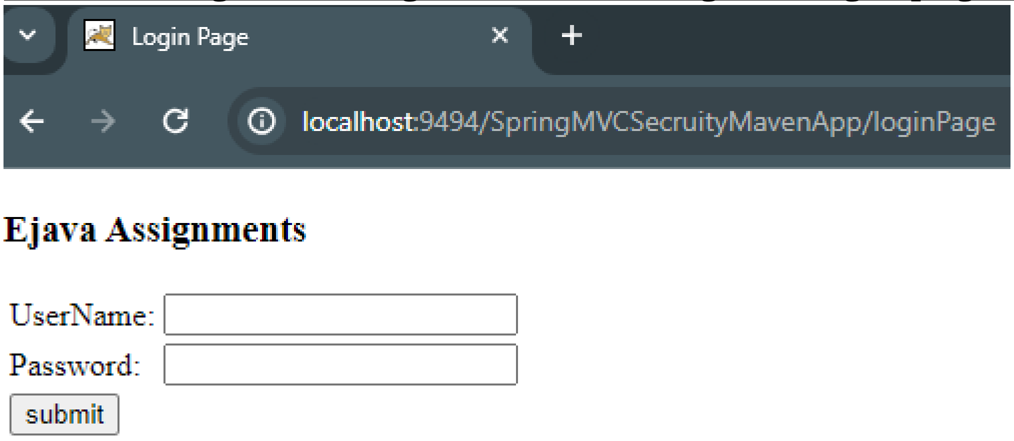

Welcome Page

localhost:9494/SpringMVCSecruityMavenApp/

Welcome to Ejava Assignments

Login to Assignments

Click on Login to Assignments link to go to Login page :


Login Page

localhost:9494/SpringMVCSecruityMavenApp/loginPage

Ejava Assignments

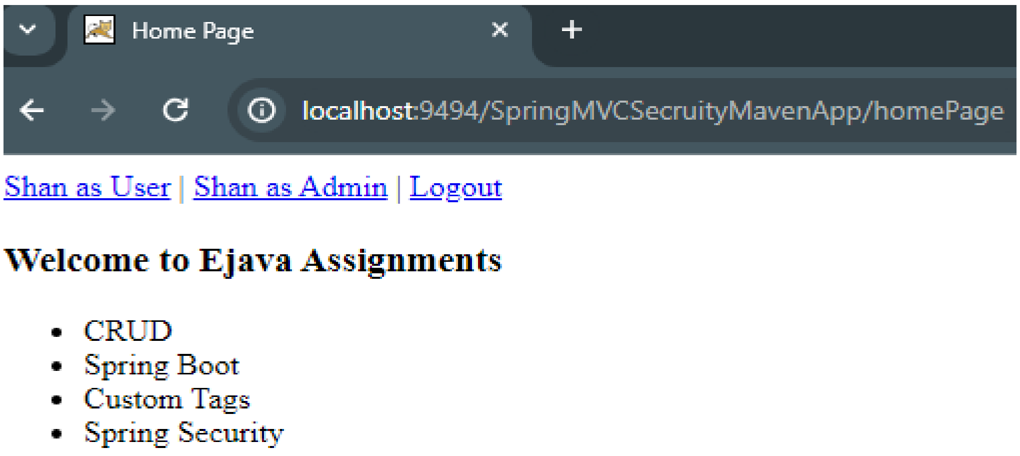
UserName:	
Password:	

submit

First Login with user credentials

UserName: shanuser

Password: shan@758


Home Page

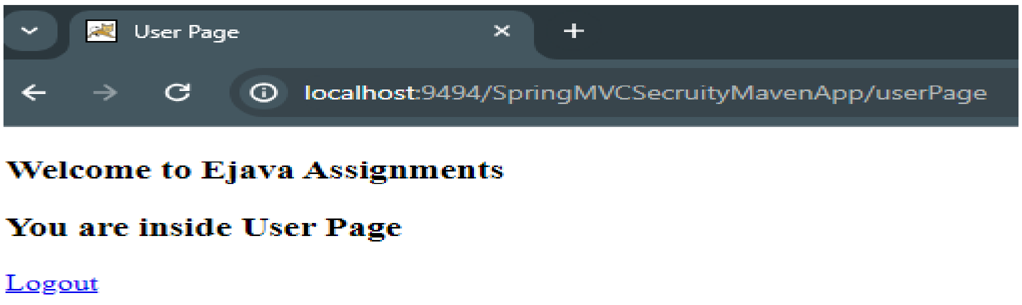
localhost:9494/SpringMVCSecruityMavenApp/homePage

Shan as User | Shan as Admin | Logout

Welcome to Ejava Assignments

- CRUD
- Spring Boot
- Custom Tags
- Spring Security

Now we will see Application HomePage with 3 Menu Options: “Shan as User”, “Shan as Admin” and “Logout”. Click on “Shan as User” link. As we have logged into application using “USER” Role Credentials, We can access this link as shown below.


User Page

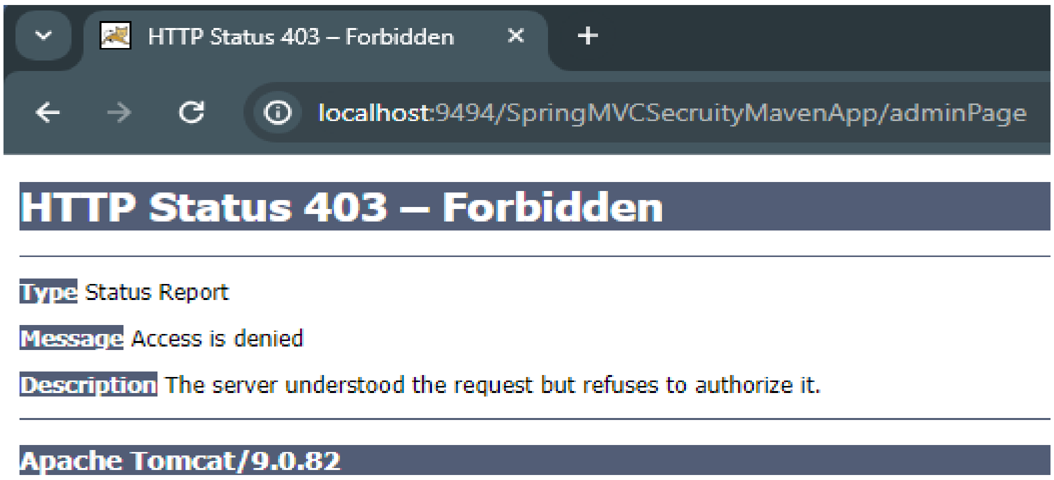
localhost:9494/SpringMVCSecruityMavenApp/userPage

Welcome to Ejava Assignments

You are inside User Page

Logout

Just use backword arrow and this time click on “Shan as Admin” Link.


HTTP Status 403 – Forbidden

localhost:9494/SpringMVCSecruityMavenApp/adminPage

HTTP Status 403 – Forbidden

Type Status Report

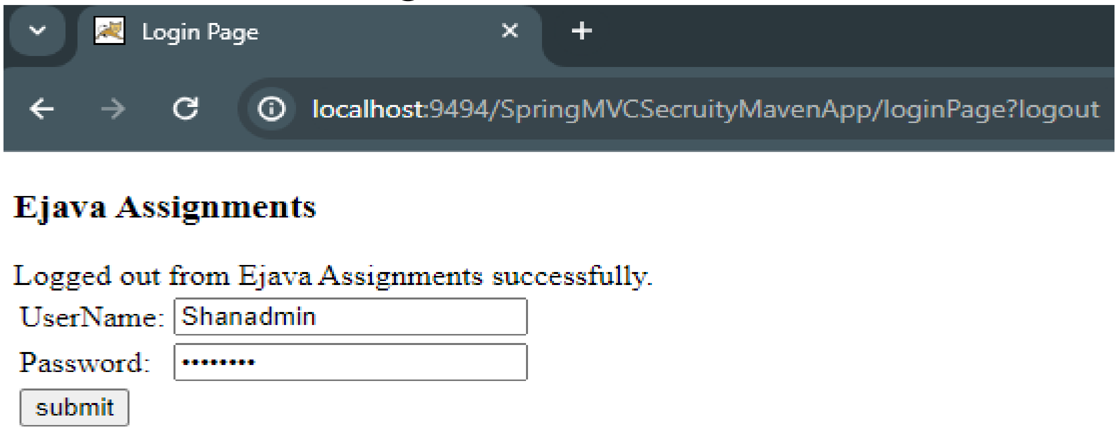
Message Access is denied

Description The server understood the request but refuses to authorize it.

Apache Tomcat/9.0.82

As we have logged in with “USER” Role Credentials, We cannot access this link. That is why we saw this error message: “403 Access is denied / Forbidden”.

Now Logged out and again login with ADMIN Role Credentials
Username: Shanadmin
Password: Shan@758


Login Page

localhost:9494/SpringMVCSecruityMavenApp/loginPage?logout

Ejava Assignments

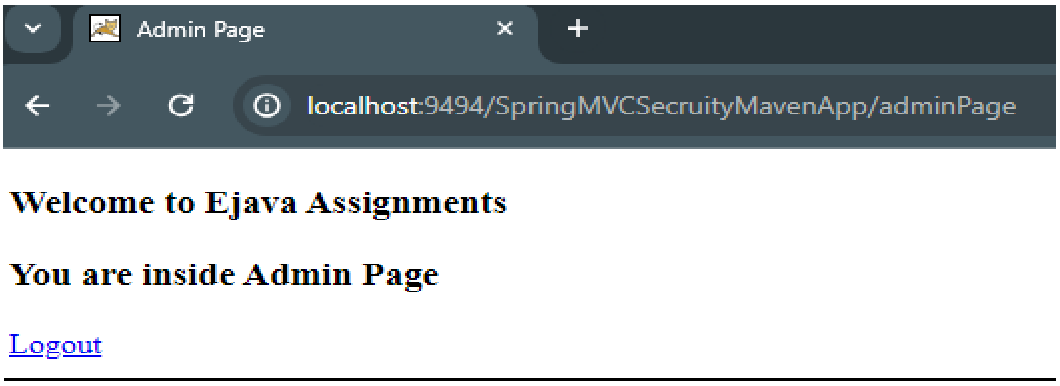
Logged out from Ejava Assignments successfully.

UserName: Shanadmin

Password: ••••••••

submit

You will now redirect to home page and try to access Shan as Admin link this time.


Admin Page

localhost:9494/SpringMVCSecruityMavenApp/adminPage

Welcome to Ejava Assignments

You are inside Admin Page

Logout

Admin Page is accessed successfully and admin can also access the user page as admin can also be an user.