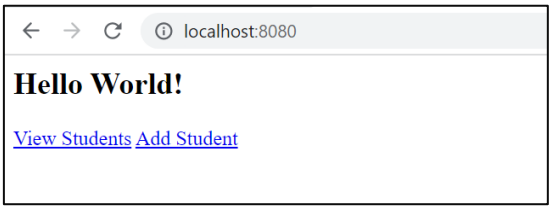
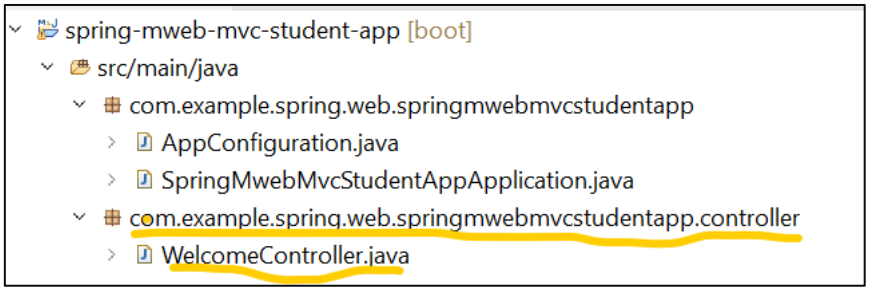
Part II will display index.jsp page and view students in the web browser.



1. Under **WEB-INF/views/** folder, create index.jsp page and add the following code.

```
<html>
<body>
<h2>Hello World!</h2>
<!--  studentform is the url that will be handled by
StudentController  -->
<a href="show/students"> View Students</a>
<a href="studentform"> Add Student </a>
</body>
</html>
```

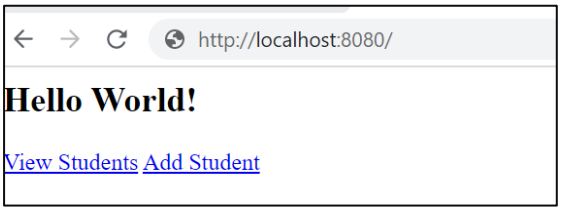
2. Create a new package to add controllers and add WelcomeController class.



3. Add following code to the WelcomeController class.

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class WelcomeController {

    @RequestMapping("/")
    public String firstPage() {
            return "index"; // index.jsp page
    }
}
```

4. Run the application – Right-click on project > Run as > Java Application. Access the url in the browser - http://localhost:8080/



Next, we need to display student details on clicking View Students link.

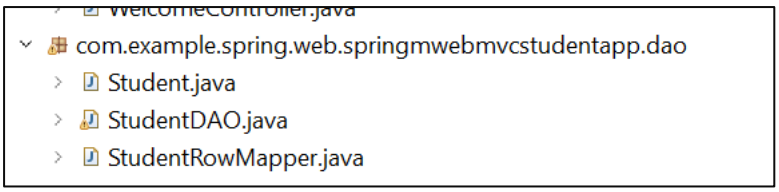
To do so:

5. Add StudentController class with the @RequestMapping and @Controller annotations.

```
package com.example.spring.web.springmwebmvcstudentapp.controller;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import com.example.spring.mvc.dao.Student;
import com.example.spring.mvc.dao.StudentDAO;
import com.example.spring.mvc.formdata.StudentFormDTO;

@Controller
public class StudentController {
    private StudentDAO studentDAO;

    @Autowired
    public StudentController(StudentDAO studentDAO) {
        System.out.println("StudentController bean is registered in the spromng
container");
        this.studentDAO = studentDAO;
    }

    / /show/students - http://localhsot:8080/show/students
    // the method will be given with Model object has a key and  value pair
    @RequestMapping("/show/students")
    public String loadStudents(Model model) {

     // model which holds data
      List<Student> studentList = this.studentDAO.getStudents();
      System.out.println(studentList);
       model.addAttribute("student", studentList);
       return "show-student"; // name of the view
    }
}
```

6. The StudentController is autowired with the **StudentDAO** bean. The StudentDAO bean will be fetching data from the MySQL database.

Create a new package for DAO components.



7. Student.java

```
package com.example.spring.web.springmwebmvcstudentapp.dao;

// Entity class - An Entity class represents the table structure
public class Student {
    private int id;
    private String name;
    private long mobile;
    private String country;

// getter and setter methods to be generated in IDE

    @Override
    public String toString() {
            return "Student [id=" + id + ", name=" + name + ", mobile=" + mobile
+ ", country=" + country + "]";
    }

}
```

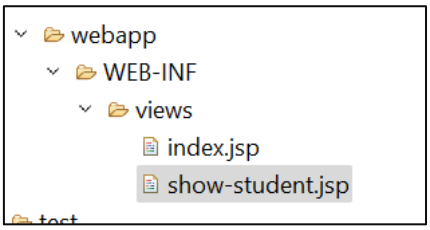
8. StudentDAO.java

```
package com.example.spring.web.springmwebmvcstudentapp.dao;

import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;
import org.springframework.stereotype.Component;
import org.springframework.stereotype.Repository;

// @Repository is a special annotation for DAO classes
@Repository
public class StudentDAO {

        private JdbcTemplate jdbcTemplate; // I need object of JDBCTemplate in this object at runtime

        @Autowired
        public StudentDAO(JdbcTemplate jdbcTemplate) {
                System.out.println("StudentDAO bean registered in the spring container");
                this.jdbcTemplate = jdbcTemplate;
        }

        // run select query -  returns multiple records
        public List<Student> getStudents() {
                String sql = "SELECT * FROM  students";
                List<Student> list = jdbcTemplate.query(sql, new StudentRowMapper());
                return list;
        }

}
```

9. StudentRowMapper.java – to map resultset data to the Student objects.

```
package com.example.spring.web.springmwebmvcstudentapp.dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import org.springframework.jdbc.core.RowMapper;

public class StudentRowMapper implements RowMapper<Student> {

// this is the code to be added by JDBCTemplate within the while(rs.next()){..}
        @Override
        public Student mapRow(ResultSet rs, int rowNum) throws SQLException {
        Student s = new Student();
                s.setId(rs.getInt(1));
                s.setName(rs.getString(2));
                s.setMobile(rs.getLong(3));
                s.setCountry(rs.getString(4));
         return s;

        }

}
```

10. Add show-student.jsp view page to display student details as table.

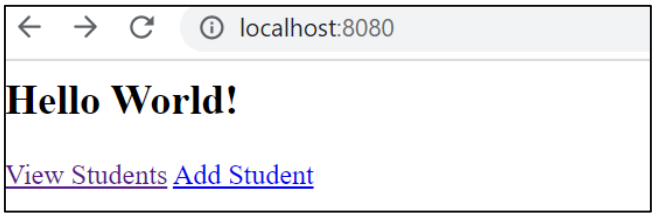


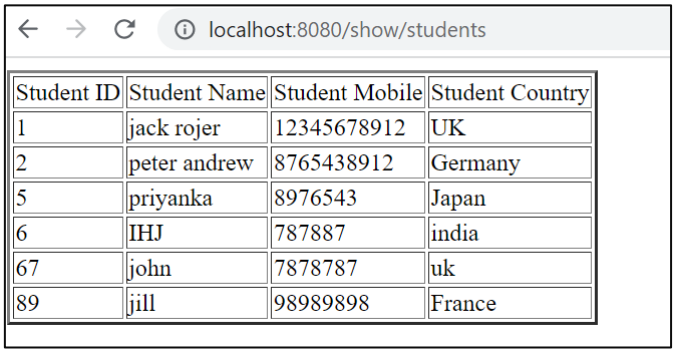
show-student.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
        <!-- <h1>Student Details - ${student}</h1>  -->
        <div align="centre">

          <table border="2">
             <thead>
                <tr>

                <td> Student ID </td>
                <td> Student Name </td>
                <td> Student Mobile </td>
                <td> Student Country </td>
               </tr>
             </thead>

           <tbody>
           <c:forEach var="s" items="${student}">
              <tr>
                  <td> ${s.id} </td>
                  <td> ${s.name} </td>
                  <td> ${s.mobile} </td>
                  <td> ${s.country} </td>
              </tr>
           </c:forEach>
           </tbody>
```

```
            </table>

        </div>

</body>
</html>
```

11. Run the application – Right-click Project > Run as > Java Application.
    Access the first page, click view students link to display student details web page.





Student ID	Student Name	Student Mobile	Student Country
1	jack rojer	12345678912	UK
2	peter andrew	8765438912	Germany
5	priyanka	8976543	Japan
6	IHJ	787887	india
67	john	7878787	uk
89	jill	98989898	France

==========End==========