OOADJ ASSIGNMENT
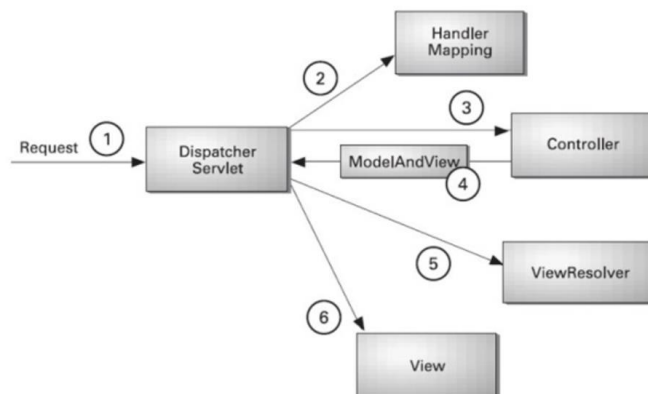
MODEL VIEW CONTROLLER IMPLEMENTATION USING SPRING FRAMEWORK

PROBLEM STATEMENT: To build a To-Do Application

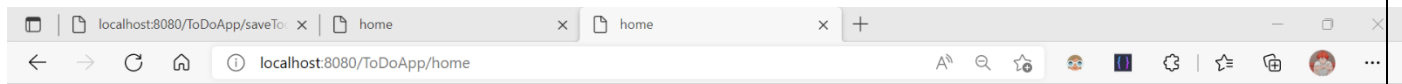| NAME	SIDDHI PATIL
SRN	PES2UG19CS389
SECTION	6-F

DEPENDENCIES USED:

1. **SPRING :The Spring Framework is an application framework and inversion of control container for the Java platform.**

2. **MAVEN: Maven is a build automation tool used primarily for Java projects.**

3. **JDBC: Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database**

4. **Eclipse IDE**

Output screenshots:

Initial Homepage before adding of To-do tasks:

Welcome to TODO Manager

Menu

Add TODO

View TODO

All TODOs

Add page:

Welcome to TODO Manager

Menu

Add TODO

View TODO

Add TODO!

Enter your todo title

Enter your todo description

Add Todo

Addition of data by user: (previously added swimming):

Welcome to TODO Manager

Add TODO!

Enter your todo title

Saved data   ✕

Swimming

Add Todo

Welcome to TODO Manager

Add TODO!

Running

Calorie burn - 220
Duration - 2 hours
distance covered - 5 km

Add Todo

Added running task

Display success message:

Welcome to TODO Manager

Successfully Added

Menu
Add TODO
View TODO

View TODO: (displays all to-do tasks added)

Welcome to TODO Manager
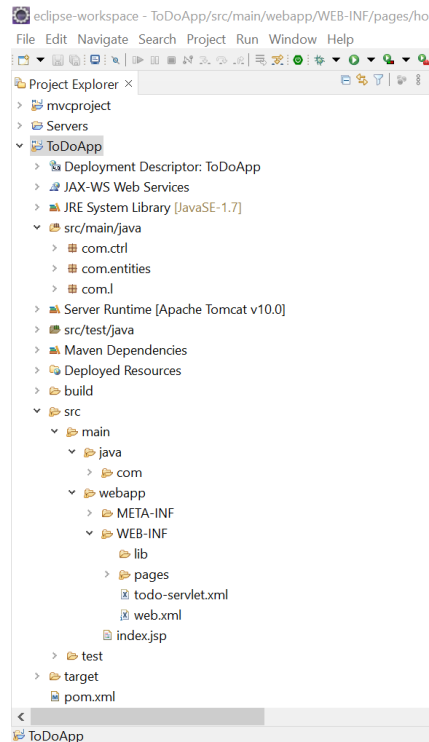
Menu
Add TODO
View TODO

All TODOs

Swimming
Swim for 3 hours on 20th december 2021

Running
Calorie burn - 220 Duration - 2 hours distance covered - 5 km

Directory Structure:

1. Project dependencies configured in pom.xml file

```xml
2.  <?xml version="1.0" encoding="UTF-8"?>
3.  #PES2UG19CS389
4.  <project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
    4.0.0.xsd">
5.      <modelVersion>4.0.0</modelVersion>
6.
7.      <groupId>com.siddhi</groupId>
8.      <artifactId>ToDoApp</artifactId>
9.      <version>0.0.1-SNAPSHOT</version>
10.     <packaging>war</packaging>
11.
12.     <name>ToDoApp Maven Webapp</name>
13.     <!-- FIXME change it to the project's website -->
14.     <url>http://www.example.com</url>
15.
16.     <properties>
17.         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
18.         <maven.compiler.source>1.7</maven.compiler.source>
19.         <maven.compiler.target>1.7</maven.compiler.target>
20.     </properties>
21.
22.     <dependencies>
23.
24.         <dependency>
25.             <groupId>junit</groupId>
26.             <artifactId>junit</artifactId>
27.             <version>4.11</version>
28.             <scope>test</scope>
29.         </dependency>
30.         <dependency>
31.             <groupId>javax.servlet</groupId>
32.             <artifactId>javax.servlet-api</artifactId>
33.             <version>3.0.1</version>
34.             <scope>provided</scope>
```

```xml
35.            </dependency>
36.            <!-- https://mvnrepository.com/artifact/jstl/jstl -->
37.            <dependency>
38.                <groupId>jstl</groupId>
39.                <artifactId>jstl</artifactId>
40.                <version>1.2</version>
41.            </dependency>
42.
43.
44.            <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -
    ->
45.            <dependency>
46.                <groupId>org.springframework</groupId>
47.                <artifactId>spring-webmvc</artifactId>
48.                <version>4.3.13.RELEASE</version>
49.            </dependency>
50.
51.        </dependencies>
52.
53.        <build>
54.            <finalName>ToDoApp</finalName>
55.            <pluginManagement>
56.                <!-- lock down plugins versions to avoid using Maven defaults (may
    be moved to parent pom) -->
57.                <plugins>
58.                    <plugin>
59.                        <artifactId>maven-clean-plugin</artifactId>
60.                        <version>3.1.0</version>
61.                    </plugin>
62.                    <!-- see http://maven.apache.org/ref/current/maven-
    core/default-bindings.html#Plugin_bindings_for_war_packaging -->
63.                    <plugin>
64.                        <artifactId>maven-resources-plugin</artifactId>
65.                        <version>3.0.2</version>
66.                    </plugin>
67.                    <plugin>
68.                        <artifactId>maven-compiler-plugin</artifactId>
69.                        <version>3.8.0</version>
70.                    </plugin>
71.                    <plugin>
72.                        <artifactId>maven-surefire-plugin</artifactId>
73.                        <version>2.22.1</version>
74.                    </plugin>
75.                    <plugin>
76.                        <artifactId>maven-war-plugin</artifactId>
77.                        <version>3.2.2</version>
78.                    </plugin>
79.                    <plugin>
80.                        <artifactId>maven-install-plugin</artifactId>
81.                        <version>2.5.2</version>
82.                    </plugin>
83.                    <plugin>
84.                        <artifactId>maven-deploy-plugin</artifactId>
85.                        <version>2.8.2</version>
86.                    </plugin>
87.                </plugins>
88.            </pluginManagement>
89.        </build>
90. </project>
```

91. **Create the controller class**

To create the controller class, we are using two annotations @Controller and
@RequestMapping.

The @Controller annotation marks this class as Controller. The @Requestmapping annotation is used to map the class with the specified URL name.

```java
package com.ctrl;
import java.util.Date;
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.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.entities.Todo;


import javax.servlet.ServletContext;
@Controller
public class HomeCtrl {
        @Autowired
        ServletContext context;
        @RequestMapping("/home")
        public String home(Model m) {
                String str ="home";
                m.addAttribute("page",str);
                List<Todo> list=(List<Todo>)context.getAttribute("list");
                m.addAttribute("todos",list);
                return "home";
        }
        @RequestMapping("/add")
        public String addTodo(Model m){
                Todo t=new Todo();
                m.addAttribute("page","add");
                m.addAttribute("todo",t);
                return "home";
        }
        @RequestMapping(value="/saveTodo",method=RequestMethod.POST)
        public String saveTodo(@ModelAttribute("todo") Todo t, Model m)
        {
                System.out.println(t);
                t.setTodoDate(new Date());
                //get todo list from context
                List<Todo> list=(List<Todo>)context.getAttribute("list");
                list.add(t);
                m.addAttribute("msg", "Successfully Added");

                return "home";
        }
}
```

92. **Provide the entry of controller in the web.xml file**
    In this xml file, we are specifying the servlet class DispatcherServlet that acts as the front controller in Spring Web MVC. All the incoming request for the html file will be forwarded to the DispatcherServlet.

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

<web-app>

  <display-name>Archetype Created Web Application</display-name>
  <servlet>
       <servlet-name>todo</servlet-name>
       <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
```

```
            <load-on-startup>1</load-on-startup>
        </servlet>

        <servlet-mapping>
            <servlet-name>todo</servlet-name>
            <url-pattern>/</url-pattern>
        </servlet-mapping>

        <listener>
        <listener-class>com.l.MyListener</listener-class>
        </listener>

</web-app>
```

93. Define the bean in the xml file

This is the important configuration file where we need to specify the View components.The context:component-scan element defines the base-package where DispatcherServlet will search the controller class.This xml file should be located inside the WEB-INF directory

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

    <context:component-scan base-package="com.ctrl"></context:component-scan>
    <mvc:annotation-driven />

       <mvc:annotation-driven
            enable-matrix-variables="true"/>

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

94. Display the messages in the JSP page

This is the simple JSP page, displaying the message returned by the Controller.:

```jsp
<%@ taglib prefix= "c" uri = "http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form"%>
<%@page isELIgnored="false" %>
<!doctype html>
<html lang="en">
<head>
<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->
<link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
```

```jsp
      crossorigin="anonymous">

<title><c:out value="${page }"></c:out></title>
</head>
<body>

      <div class="container mt-3">
            <h1 class="text-center">Welcome to TODO Manager</h1>
            <br>
            <c:if test="${not empty msg }">
            <div class="alert alert-success">
                  <b><c:out value="${msg }"></c:out></b></div>
            </c:if>
            <div class="row mt-6">
                  <div class="col-md-2">

                        <div class="list-group">
                              <button type="button"
                                    class="list-group-item list-group-item-action active"
                                    aria-current="true">Menu</button>
                              <a href='<c:url value='/add'></c:url>'
                                    class="list-group-item list-group-item-action">Add
TODO</a>

                              <a href='<c:url value='/home'></c:url>'
                                    class="list-group-item list-group-item-action">View
TODO</a>


                        </div>
                  </div>
                  <div class="col-md-10">
                        <c:if test="${page=='home' }">
                              <h3 class="text-center">All TODOs</h3>
                              <c:forEach items="${todos }" var="t">
                                    <div class="card">
                                          <div class="card-body">
                                                <h3>
                                                      <c:out value="${t.todoTitle
}"></c:out>

                                                </h3>
                                                <p>
                                                      <c:out value="${t.todoContent
}"></c:out>

                                                </p>
                                          </div>
                                    </div>
                              </c:forEach>
                        </c:if>
                        <c:if test="${page=='add' }">
                        <h3 class="text-center">Add TODO!</h3>
                        <br>
                        <form:form action="saveTodo" method="post" modelAttribute="todo">
                              <div class="form-group">
                                    <form:input path="todoTitle" cssClass="form-control"
placeholder="Enter your todo title"></form:input>
                              </div>
                              <br>
                              <div class="form-group">
                                    <form:textarea path="todoContent" cssClass="form-
control" placeholder="Enter your todo description" cssStyle="height:300px;"/>
                              </div>
                              <br>
                              <div class="container text-center">
                              <button class="btn btn-outline-success">Add
Todo</button></div>
```

```
                    </form:form>
                    </c:if>
                </div>
            </div>
        </div>

        <!-- Optional JavaScript; choose one of the two! -->

        <!-- Option 1: Bootstrap Bundle with Popper -->
        <script
            src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
            integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
            crossorigin="anonymous"></script>

        <!-- Option 2: Separate Popper and Bootstrap JS -->
        <!--
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"
integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"
integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"
crossorigin="anonymous"></script>
        -->
</body>
</html>
```