1. Create maven project for war packaging
   Add all required dependency in pom.xml
2. <mark>Add web.xml file</mark>
   <mark>Add following entry</mark>
   ```xml
   <servlet>
           <servlet-name>spring</servlet-name>
           <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
           <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
      <servlet-name>spring</servlet-name>
      <url-pattern>/</url-pattern>
    </servlet-mapping>
   ```
3. <mark>Create spring configuration file by name spring-servlet.xml inside WEB-INF folder</mark>
   ```xml
   <beans xmlns="http://www.springframework.org/schema/beans"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xmlns:aop="http://www.springframework.org/schema/aop"
           xmlns:mvc="http://www.springframework.org/schema/mvc"
           xmlns:p="http://www.springframework.org/schema/p"
           xmlns:context="http://www.springframework.org/schema/context"
           xsi:schemaLocation="http://www.springframework.org/schema/aop
   http://www.springframework.org/schema/aop/spring-aop.xsd
                   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:annotation-config></context:annotation-config>
   <context:component-scan base-package="com.demo"></context:component-scan>
   <mvc:annotation-driven></mvc:annotation-driven>

   </beans>
   ```

4. <mark>Write controller class inside com.demo.controller package and annotate the class with @Controller</mark>

```java
@RequestMapping("/")

        public String displayhomepage() {

                System.out.println("in display login");

                return "index";

        }

        @RequestMapping("/hello")

        public ModelAndView sayHello() {
```

String message="this is message from say hello";

//message data can be accessed in hello.jsp page by name msg

return new ModelAndView("hello","msg",message);

}

```
<bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver"
        p:prefix="/WEB-INF/views/" p:suffix=".jsp"
p:viewClass="org.springframework.web.servlet.view.JstlView">
        </bean>
```
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
 <h3> Hello World!!</h3>
 <h4>Message : ${msg}</h4>
</body>
</html>
```
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <h2>welcome to index.jsp</h2>
</body>
</html>
```

In controller class method
Writing @RequestParam("uname") String unm   is same as
String unm=request.getParameter("uname")