==========
Spring Web MVC
==========

=> One of the most famous & important module in spring framework

=> Using Spring Web MVC we can develop 2 types of applications.

1) Web Applications (C2B)

2) Distributed Applications (B2B)

==========
Web Application
==========

=> Web Applications are used for Customer to Business Communication.

Ex: amazon, flipkart, naukri, ashokit

Note: In Web application we will have 3 components

1) Presentation Components (UI)

2) Business Components (Controllers + Services)

3) Data Access Components (Repositories)

Note: To develop presentation (UI) components in Spring Web MVC application we can use JSP and Thymeleaf.

==========
What is Distributed application
==========

=> Distributed applications are called as Webservices / Rest APIs.

=> Webservices are used to communicate from one application to another application.

ex: passport ----------> aadhar
gpay ----------> sbi bank
makemytrip ----> irctc

Note: In distributed appliations UI will not be available (pure backend apis).

==========
Spring Web MVC Architecture
==========

1) Dispatcher Servlet

2) Handler Mapper

3) Controller / Request Handler

4) ModelAndView

5) ViewResolver

6) View

==========
DispatcherServlet
==========

=> It is predefined class in spring web mvc

=> It acts as front controller (main gate of the house)

=> It is responsible to recieve request and send the response to client.

Note: It is also called as framework servlet class.

==========
Handler Mapper
==========

=> It is predefined class in spring web mvc

=> It is responsible to identify controller class to handle the request based on url-pattern and give controller class details to dispatcher servlet.

==========
Controller
==========

=> Controllers are java classes which are used to handle the request (request processing).

=> DispatcherServlet will call controller class methods.

=> After processing request, controller method will return ModelAndView object to dispatcher servlet.

Model -> It is a map to represent data in key-value format

View -> It represents view page name

==========
View Resolver
==========

=> It is used to identify view files location.

=> Dispatcher Servlet will give view name to View Resolver then it will identify the view file location and give it to Dispatcher Servlet.

========
View
========

=> It is responsible to render model data on the view page and give it to dispatcher servlet.

Note: DispatcherServlet will send final response to client.

==========
Developing First Spring Web MVC Based App
==========

Step-1 : Create boot app with below dependencies

a) web-starter

b) Thymeleaf

c) devtools

Step-2 : Create Controller class with required methods

Step-3 : Create View Page and access Model data in view page

Views Location : src/main/resources/templates/

Step-4 : Run the application and test it using browser

```
@Controller
@RequestMapping("/msg")
public class MsgController {

	// URL : http://localhost:8080/msg/greet

	@GetMapping("/greet")
	public ModelAndView greetMsg() {

		ModelAndView mav = new ModelAndView();
		mav.addObject("msg", "Good Morning..!!");
		mav.setViewName("index");

		return mav;
	}

	// URL : http://localhost:8080/msg/welcome

	@GetMapping("/welcome")
	public String welcomeMsg(Model model) {

		model.addAttribute("msg", "Welcome to Ashok IT..!!");

		return "index";
	}

}
```

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Ashok IT</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
4Q6Gf2aSP4eDXB8Miphtr37CMZZQ5oXLH2yaXMJ2w8e2ZtHTl7GptT4jmndRuHDT"
crossorigin="anonymous">
  </head>
  <body>
    <h1 th:text="${msg}"></h1>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.6/dist/js/bootstrap.bundle.min.js
" integrity="sha384-
j1CDi7MgGQ12Z7Qab0qlWQ/Qqz24Gc6BM0thvEMVjHnfYGF0rmFCozFSxQBxwHKO"
crossorigin="anonymous"></script>
  </body>
```

```
</html>
-------------------------------------------------------------------------------


================================================================================
=================
Assignment : Develop spring boot application to retreive users_table data from
database and display users data in html page in the table format. Develop this
project using layered architecture.
================================================================================
===============



@Controller + @ResponseBody = RestController

============================
What is Request Parameter ?
============================

=> Request Parameters also called as Query Parameters.

=> These are used to send data from client to server in URL.

=> Request Parameters will represent data in key-value format.

                Ex : https://www.youtube.com/watch?v=McmckGLzZ4Q&t=11700s

=> Request Parameters will start with ? and will be seperate by &.

=> To read Request Parameters from the URL we will use @RequestParam annotation.

-------------------------------------------------------------------------------
---
@Controller
public class MsgController {

        // URL : http://localhost:8080/greet?name=raj
        @GetMapping("/greet")
        @ResponseBody
        public String greetMsg(@RequestParam("name") String name) {

                String msg = name + ", Good Morning..!!";

                return msg;
        }

        // URL : http://localhost:8080/course?c=sbms&t=ashok
        @GetMapping("/course")
        @ResponseBody
        public String getCourse(@RequestParam("c") String course,
@RequestParam("t") String trainer) {

                String msg = course + " By " + trainer + " will start soon...";

                return msg;
        }
}
-------------------------------------------------------------------------------
--

========================
What is Path Variable ?
```

==========

=> Path Variables also called as URI variables and Path Parameters.

=> These are used to send data from client to server in URL.

Ex : https://www.instagram.com/reel/{DJtyr-igaPD}/

Note: Path Variables will represent data directley without any key.

Note: Path Variables position we need to represent in URL template.

Ex: @GetMapping("/greet/{name}")

=> To read path variables from URL we will use @PathVariable annotation.

```
// URL : http://localhost:8080/welcome/raj
        @GetMapping("/welcome/{name}")
        @ResponseBody
        public String getWelcomeMsg(@PathVariable("name") String name) {

                String msg = name + ", Welcome to Ashok IT";

                return msg;
        }
```

Path Variable : Used to uniquely identify a resource

Request Parameter : Used to filter, sort, or provide optional input.

=> We have below limitations with URL data

1) Data is exposing in browser URL (others can read it who are sitting beside us)

2) URL length limitation

3) Sensitive data we can't send in URL

4) Will not support for binary data (ex: images, videos, audios, files etc..)

==========
What is Request Body ?
==========

=> Using Request Body we can send data from client to server without exposing in URL.

=> Using Request Body we can send binary data also (ex: images, videos, audios, files etc..) to the server.

Note: When we are submitting forms then we willuse Request Body to send form data to Controller.