**3. Spring MVC (Model-View-Controller) 1. Spring MVC Overview:**

• **Theory**

**1.** **Explanation of the MVC design pattern:** Model, View, and Controller:- MVC (Model-View-Controller) Design Pattern

MVC is a software design pattern used for developing web applications by separating concerns into three interconnected components:

1. Model (Data & Business Logic)
   * Represents the application's data and business logic.
   * Interacts with the database.
   * Example: A User class that fetches and processes user details from a database.
2. View (User Interface)
   * Handles presentation and UI.
   * Displays data from the Model.
   * Example: A JSP, HTML, or frontend framework like React showing user details.
3. Controller (Request Handling)
   * Manages user input and request processing.
   * Calls the Model for data and updates the View accordingly.
   * Example: A Servlet or Spring Controller handling user login requests.

**2. How Spring MVC handles incoming web requests and maps them to the correct controller:-** Spring MVC follows a request-processing workflow to handle web requests and map them to the correct controller:

1. Client Sends Request

* A user makes an HTTP request (e.g., GET /users or POST /login).

2. DispatcherServlet Receives the Request

* DispatcherServlet (Front Controller) is the main servlet in Spring MVC.
* It intercepts all incoming requests and delegates them for further processing.

3. Handler Mapping Identifies the Correct Controller

* DispatcherServlet consults Handler Mapping to determine which controller method should handle the request.
* It uses annotations like @RequestMapping, @GetMapping, @PostMapping, etc.

4. Controller Processes the Request

* The appropriate controller method executes business logic and interacts with the model (database).