**Họ và tên: Trần Thị Huyền Trang**

**Lớp hành chính: DCCNTT 13.10.21 Mã sinh viên: 20224034**

**Đề: Tìm hiểu đặc điểm mô hình MVC**

Bài làm

#### **1. Đặc điểm mô hình MVC**

**Mô hình MVC (Model – View – Controller)** là một mẫu kiến trúc phần mềm phổ biến, giúp tách biệt các thành phần xử lý nghiệp vụ, giao diện người dùng và xử lý tương tác, từ đó giúp dễ dàng bảo trì và mở rộng ứng dụng.

##### ***a. Các thành phần trong MVC***

**Model**

* Chịu trách nhiệm quản lý dữ liệu, trạng thái và logic nghiệp vụ của ứng dụng.
* Thường bao gồm các class mô hình, các đối tượng truy vấn cơ sở dữ liệu, xử lý thêm, sửa, xóa hoặc đọc dữ liệu.
* Ví dụ: Lớp User, Product, các DAO (Data Access Object), Repository...

**View**

* Là thành phần hiển thị dữ liệu cho người dùng (UI).
* Nhận dữ liệu từ Model thông qua Controller và trình bày chúng.
* Bao gồm: HTML, JSP, Thymeleaf, XML, hoặc các UI component khác (nút bấm, menu, form nhập liệu…).

**Controller**

* Là “bộ điều khiển” – đóng vai trò cầu nối giữa View và Model.
* Tiếp nhận yêu cầu từ người dùng (qua giao diện), xử lý yêu cầu đó, gọi đến Model nếu cần thiết, sau đó trả kết quả về cho View để hiển thị.
* Controller chứa logic điều hướng và xử lý các tương tác như đăng nhập, lưu form, chuyển hướng trang…

##### ***b. Luồng xử lý của mô hình MVC***

* Người dùng gửi yêu cầu (Request) từ trình duyệt (ví dụ: nhấn nút "Đăng nhập").
* **Controller** nhận yêu cầu và xử lý nghiệp vụ tương ứng.
* **Model** được Controller gọi đến để xử lý dữ liệu (ví dụ: kiểm tra thông tin đăng nhập từ database).
* Sau khi xử lý xong, dữ liệu được trả về **Controller**, sau đó gửi cho **View**.
* **View** nhận dữ liệu và hiển thị kết quả cho người dùng (ví dụ: "Đăng nhập thành công").

#### **2. Mô hình MVC trong Spring Framework**

Spring là một framework mạnh mẽ và phổ biến của Java, hỗ trợ rất tốt cho việc phát triển các ứng dụng web theo mô hình **MVC**. Spring MVC giúp quản lý cấu trúc rõ ràng giữa các phần: Model, View và Controller.

##### ***a. Đặc điểm của Spring MVC***

* Sử dụng **DispatcherServlet** làm trung tâm xử lý mọi yêu cầu HTTP.
* Hỗ trợ nhiều công nghệ View: JSP, Thymeleaf, FreeMarker...
* Tích hợp dễ dàng với cơ sở dữ liệu, REST API, các thư viện bảo mật, validation…

##### ***b. Cấu trúc Spring MVC***

* **Model:** Các POJO class (Plain Old Java Object), thường đi kèm với JPA Entity, Repository.
* **View:** JSP hoặc Thymeleaf là phổ biến. Dùng để hiển thị kết quả cho người dùng.
* **Controller:** Annotated với @Controller hoặc @RestController, chứa các phương thức xử lý yêu cầu (@GetMapping, @PostMapping,...).

##### ***c. Luồng xử lý trong Spring MVC***

* Trình duyệt gửi yêu cầu đến DispatcherServlet.
* DispatcherServlet tìm Controller phù hợp dựa trên URL.
* Controller xử lý logic, gọi đến Service và Repository để truy xuất dữ liệu (Model).
* Dữ liệu sau xử lý được đưa vào ModelAndView và gửi đến View (JSP/Thymeleaf).
* View hiển thị dữ liệu cho người dùng.