1. What is Spring MVC?

Interview answer:
“Spring MVC is a web framework based on the Model-View-Controller design pattern. It is used to build web applications by separating business logic, UI, and request handling.”

---

2. What problem does Spring MVC solve?

Interview answer:
“It provides a clean separation of concerns, centralized request handling, and makes web applications easier to develop, test, and maintain.”

---

3. Explain MVC architecture in Spring MVC.

Interview answer:
“Model holds business data, View is responsible for presentation, and Controller handles incoming requests and coordinates between Model and View.”

---

4. What is DispatcherServlet?

Interview answer:
“DispatcherServlet is the front controller in Spring MVC. It receives all incoming requests and dispatches them to the appropriate controller.”

---

5. Explain request flow in Spring MVC.

Interview answer:
“Client sends request → DispatcherServlet receives it → HandlerMapping finds the controller → Controller processes request → returns view name → ViewResolver resolves the view → response is sent to client.”

---

6. What is a Controller in Spring MVC?

Interview answer:
“A controller is a Spring-managed component that handles web requests and returns a response or view.”

---

7. Difference between @Controller and @RestController?

Interview answer:
"@Controller returns a view, while @RestController returns data directly as JSON or XML. @RestController is a combination of @Controller and @ResponseBody."

---

8. What is @RequestMapping?

Interview answer:
"@RequestMapping is used to map HTTP requests to controller methods based on URL and HTTP method."

---

9. Difference between @GetMapping and @PostMapping?

Interview answer:
"They are specialized versions of @RequestMapping used for GET and POST HTTP methods, improving readability."

---

10. What is @ResponseBody?

Interview answer:
"@ResponseBody tells Spring to write the return value of a method directly into the HTTP response body."

---

11. What is @RequestBody?

Interview answer:
"@RequestBody is used to bind HTTP request body data (usually JSON) to a Java object."

---

12. Difference between @RequestParam and @PathVariable?

Interview answer:
"@RequestParam reads data from query parameters, while @PathVariable reads data from the URL path."

---

13. What is @ModelAttribute?

Interview answer:
"@ModelAttribute binds request parameters to a model object and also makes it available to the view."

---

14. What is ViewResolver?

Interview answer:
“ViewResolver maps a logical view name to an actual view technology like JSP, Thymeleaf, or FreeMarker.”

---

15. What is Model and ModelAndView?

Interview answer:
“Model is used to pass data to the view, while ModelAndView holds both model data and view name together.”

---

16. What is form validation in Spring MVC?

Interview answer:
“Spring MVC supports form validation using JSR-303 annotations like @NotNull, @Size, and @Email.”

---

17. What is BindingResult?

Interview answer:
“BindingResult holds validation errors and must be placed immediately after the validated object.”

---

18. How does exception handling work in Spring MVC?

Interview answer:
“Using @ExceptionHandler at controller level or @ControllerAdvice for global exception handling.”

---

19. What is @ControllerAdvice?

Interview answer:
“@ControllerAdvice is used for global exception handling and common model attributes across controllers.”

---

20. What is Interceptor in Spring MVC?

Interview answer:
“Interceptors allow pre-processing and post-processing of requests, commonly used for logging and authentication.”

21. Difference between Filter and Interceptor?

Interview answer:
“Filters are part of Servlet API and work at container level, while interceptors are Spring-specific and work at framework level.”

---

22. How does Spring MVC support RESTful services?

Interview answer:
“By using @RestController, @RequestBody, @ResponseBody, and proper HTTP methods.”

---

23. What is Content Negotiation?

Interview answer:
“It allows Spring MVC to return responses in different formats like JSON or XML based on client request.”

---

24. What is HttpMessageConverter?

Interview answer:
“It converts Java objects to JSON/XML and vice versa during request and response processing.”

---

25. Difference between Spring MVC and Servlet?

Interview answer:
“Servlet requires manual request handling, while Spring MVC provides a structured framework with DispatcherServlet, annotations, and view resolvers.”

---

26. How do you secure Spring MVC applications?

Interview answer:
“By integrating Spring Security for authentication, authorization, and CSRF protection.”

---

27. Real-time use of Spring MVC in projects?

Interview answer:
“Spring MVC is used to build controllers, handle web requests, manage form submissions, validation, and integrate frontend with backend.”