NOTE:- This is a question bank for EPJ Subject. <mark>This is only for practice purpose</mark>

QUESTION BANK

◆ Unit 1 – XML (10 Questions)

1. What is XML and why is it used in web applications?

2. Differentiate between XML and HTML.

3. What is a well-formed and valid XML document?

4. Explain XML Schema (XSD) and its advantages over DTD.

5. What are XML elements, attributes, and namespaces?

6. What is the purpose of the `<!DOCTYPE>` declaration in XML?

7. Explain the difference between DOM and SAX parsers.

8. How is XML used in configuration files like `web.xml` and `pom.xml`?

9. What is XSLT and how does it transform XML into HTML?

10. Write a simple XML document to store student details (id, name, marks).

11. Explain the lifecycle of a servlet with proper methods.

12. Write a simple JSP page to display a user's name using request parameters.

13. List and describe the different types of JDBC drivers.

14. What are the advantages of using PreparedStatement in JDBC?

15. Write a simple JSP program to calculate and display the square of a number using request parameter.

16. Explain how session tracking works in servlets. Provide code snippets.

17. Write a JDBC program to connect to a MySQL database and retrieve data from the "students" table.

18. Write a Java servlet to display user login details from a form (username and password) using doPost().

19. Write a complete JDBC program to update user email in the "users" table based on user ID.

◆ Unit 2 – JDBC (20 Questions)

11. What is JDBC? Explain its architecture.

12. List and describe the different types of JDBC drivers.

13. What are the steps to connect Java with MySQL using JDBC?

14. Explain the purpose of `DriverManager`, `Connection`, and `Statement` interfaces.

15. What is the difference between `Statement` and `PreparedStatement`?

16. Explain the advantages of using `PreparedStatement` in JDBC.

17. What is a `ResultSet`? How do you retrieve data from it?

18. What is transaction management in JDBC? Explain `commit()`, `rollback()`, and `setAutoCommit()`.

19. What is connection pooling and why is it important?

20. What is batch processing in JDBC?

21. How can you call a stored procedure using JDBC?

22. Explain `ResultSetMetaData` and its usage.

23. Write JDBC pseudocode to insert a record into the student table.

24. Write a complete JDBC program to update user email in the `users` table.

25. How do you handle SQL exceptions in JDBC?

26. What are savepoints in JDBC?

27. Explain the Type-4 JDBC driver.

28. What is the difference between `execute()`, `executeQuery()`, and `executeUpdate()`?

29. How to connect JDBC with Servlets for dynamic web pages?

30. What is the role of JDBC in ERP applications?

31. Create a servlet that handles a feedback form submission and displays the user's name, email, and message using doPost().

32. Explain the complete life cycle of a servlet with the help of a diagram.

33. Compare and contrast GenericServlet and HttpServlet classes. Provide examples.

34. Describe the role of HttpServletRequest and HttpServletResponse in servlet communication.

35. Discuss the various ways to handle session management in servlets. Explain the use of HttpSession.

36. Write a servlet that takes user input from an HTML form and displays a customized welcome message.

37. What are RequestDispatchers in servlets? Explain forward() vs include() with examples.

◆ Unit 3 – Servlets (20 Questions)

31. What is a servlet and what are its advantages?

32. Explain the **lifecycle of a servlet** with proper methods.

33. Differentiate between `GenericServlet` and `HttpServlet`.

34. What is the role of `web.xml` deployment descriptor?

35. Explain `init()`, `service()`, and `destroy()` methods in servlet.

36. What is `HttpServletRequest` and `HttpServletResponse`?

37. Explain the difference between `doGet()` and `doPost()` methods.

38. What is RequestDispatcher? Differentiate between `forward()` and `include()`.

39. Explain session tracking methods in servlets with examples.

40. What is the purpose of `ServletContext` and `ServletConfig`?

41. What is a cookie and how is it used in servlets?

42. How can data be shared between two servlets?

43. Write a servlet to display login details using `doPost()`.

44. What are filters in servlets?

45. Explain exception handling in servlets.

46. What is the difference between `sendRedirect()` and `forward()`?

47. What is a servlet container (e.g., Tomcat)?

48. Explain how to upload files using servlets.

49. What is annotation-based servlet configuration?

50. Explain MVC architecture in servlet-based applications.

51. explain how to read and write cookies in servlets. What are the security considerations?

52. How can you share data between servlets in a web application?

53. Explain using ServletContext and RequestDispatcher.

54. Describe the structure of a web.xml deployment descriptor.

55. Discuss MVC architecture in the context of servlet-based web applications.

56. JDBC Pseudocode – Insert a student record: a. Load driver, connect DB, create PreparedStatement, set values, execute, close.

57. JSP Pseudocode – Show "Welcome" and current date: a. Use <%= new java.util.Date() %>

58. Servlet Pseudocode – Two static pages with navigation via button: a. Create two servlets; one serves form with button, second serves the content

---

◆ Unit 4 – JSP (Java Server Pages) (15 Questions)

51. What is JSP and how is it different from Servlets?

52. Explain the lifecycle of a JSP page.

53. What are scripting elements in JSP?

54. What are implicit objects? List all nine.

55. What is the purpose of the `page` directive in JSP?

56. Explain the difference between include directive and include action.

57. How does JSP handle exceptions?

58. Explain Expression Language (EL) and its benefits.

59. What is the use of `<jsp:useBean>`, `<jsp:setProperty>`, and `<jsp:getProperty>`?

60. What is JSTL and why is it used?

61. Write a JSP page to display a user's name using request parameters.

62. Write a JSP program to calculate the square of a number.

63. Explain MVC in JSP with servlet integration.

64. How can JSP be connected to a database?

65. Explain how JSP is converted into a servlet during execution.

◆ Unit 5 – Hibernate (15 Questions)

66. What is Hibernate and why is it used?

67. Explain the architecture of Hibernate.

68. What is ORM (Object Relational Mapping)?

69. What is the purpose of `hibernate.cfg.xml`?

70. What are persistent, transient, and detached states in Hibernate?

71. Explain annotations like `@Entity`, `@Table`, and `@Id`.

72. What is SessionFactory and Session object?

73. What is HQL? How does it differ from SQL?

74. Explain one-to-many and many-to-many mappings with examples.

75. What is lazy loading vs eager loading?

76. What are cascading operations in Hibernate?

77. How to handle transactions in Hibernate?

78. What is the second-level cache?

79. How does Hibernate improve performance compared to JDBC?

80. Explain how Hibernate integrates with Spring Boot.

◆ Unit 6 – Spring & Spring Boot (20 Questions)

81. What is Spring Framework?

82. Explain Dependency Injection (DI) and Inversion of Control (IoC).

83. What is BeanFactory and ApplicationContext?

84. What are @Component, @Service, and @Repository annotations?

85. Explain the lifecycle of a Spring Bean.

86. What is the role of `DispatcherServlet` in Spring MVC?

87. What is the purpose of `@Controller` and `@RestController` annotations?

88. Explain `@RequestMapping`, `@GetMapping`, and `@PostMapping`.

89. What is ModelAndView in Spring MVC?

90. What is Spring Boot and how does it simplify development?

91. What is auto-configuration in Spring Boot?

92. Explain the purpose of `application.properties` and `application.yml`.

93. What is a starter dependency in Spring Boot?

94. What is Spring Boot Actuator and its use?

95. What is Spring Data JPA and its repositories?

96. How do you connect Spring Boot to MySQL database?

97. What is `@Transactional` annotation used for?

98. What is microservice architecture?

99. How to create and test RESTful APIs in Spring Boot?

100. Explain how Maven integrates with Spring Boot for dependency management.