Full Stack Development Question Bank

UNIT-1

SL. NO	Question Number	Marks	BL
UNIT-1			
1	NMIT wants to create a website for the college. As per the survey Angular JS and React JS are most widely used in web application development. Assume you are a web developer and suggest which JS to use for creating the website and specify why you have chosen that particular JS with the various differences & advantages	6	3
2	What is DOM? Explain how every html tag will become a DOM object with suitable example.	8	2
3	Write a react simple application for html document setup and display the simple text message by rendering the ReactDOM. Explain each step of code written in the above program.	8	3
4	Suppose you join as a web designer in a company and you have been told to use React JS to develop web applications. Everything in React is referred to as components. Now	6	2
5	Suppose you join as a web designer in a company and you have been told to use React JS to develop web applications. Everything in React is referred to as components. Now Identify and list out merits and demerits of each component type with suitable examples that are required for the development. Also specify the differences between the two components.	7	4
6	Write a example to Display component that display only a single child and discuss the uses of React.children	6	3
7	Company 'A' designed a web application which consists of ten components out of which six components contain similar functionality. This is not a good practice to have redundancy in code. Assume you are a web developer, analyse the above problem and come out with a solution such that all the components which consist similar functionality can be made into a single function and can be used wherever required. Demonstrate with an example	7	4
8	Consider an online shopping application which consists of various ways the payment can be made. There is one dedicated button for each payment method. For example, a button for Google Pay, a different button for PhonePe etc. When the customer clicks on the particular payment method an event occurs i.e., different application page loads up. Apply this scenario	8	4
9	Write a JSX for unorder list. with an example explain React elements as JSX	8	2
10	How do you pass data to React components? Demonstrate with examples	6	2

11	Write a solution such that all the components which consist similar functionality can be made into a single function and can be used wherever required. Demonstrate with an example and can be used wherever required. Demonstrate with an example	8	3
12	Write a program to find the factorial of a given number using ReactJS		3
13	Difference between class component and functional component? Explain with an example.		2
14	What is the use of references (refs) in React. Write a code to AddColorForm component renders the HTML with a text input and a color input for collecting hex values from the color wheel.	6	2
15	What are the React built-in automatic property validation for the variable types	6	2
16	Mr. Alex has recently joined as a web designer in a company and he has been told to use React JS to develop web applications. Everything in React is referred to as components. Before using components, he needs to understand the complete life-cycle of a component. Assume you are Mr. Alex and specify why life-cycle methods are important and also specify the various phases along with methods present in every component life-cycle.	8	3
17	Design a React JS application using forms. The application should contain a text area and the input text values should change when the state is updated. When the user changes the state, the application should print the updated value on the browser screen.	7	4
`18	A login page for an application needs to be designed using Refs. The page should contain a label, two text areas named username & password and a submit button. Assume you are the designer, analyze the given problem and provide a solution. Make sure that when the application loads on the browser the default focus should be on username.	7	
19	Explain the following terms with respect to ReactJS. a. Components, b. Props c. States		tg
20	How do you create and handle an event in in ReactJS? Explain with an example		
21	Create a web page having a login form such as user name and password, and a submit button, when user click on submit, and the user name and password are same then your web page color should change to green and welcome “user name” should be displayed on the screen		
UNIT-II			
1	Draw the architecture of spring framework and explain the components		
2	What are the different forms of Dependency injection? Write a java program for whether services sample code of method injection.		
3	How can we initialize the BeanFactory class in spring. Give the example of Bean Life cycle for initialization callbacks and destruction callbacks.		
4	Which are the important beans lifecycle methods? Can you override them?		
5	Elaborate on the important components constituting the “Spring Frame work runtime.”		

6	Dependency Injection is one of the core functions of Spring. Clearly state the need for this activity and give a suitable example.	8	4
7	Draw the architecture of spring framework and explain the components.	6	2
8	Analyze the different forms of Dependency Injection (constructor, setter, method injection) and evaluate their trade-offs in terms of testability, lifecycle management, encapsulation, and readability. Write a java program for whether services sample code of method injection.		

9	**Find the below code for tight coupling:**
	class Course {
	Concept t = new Concept();
	public void startReading()
	{ t.understand(); } }
	class Concept { public void understand() {
	System.out.println("Tight coupling concept");
	}}
	Write a sample code to convert above code to loose coupling by adding interface.		

10	Which are the important beans lifecycle methods? Can you override them?		
11	Analyze the process of initializing the BeanFactory class in Spring and explain how initialization and destruction callbacks are managed within the bean lifecycle. How does this differ from using ApplicationContext?		
12	Demonstrate how to use the p and c namespaces in Spring XML configuration to inject properties and constructor arguments. Explain how it is initialized in the XML file.		
	Unit-III		
1	Demonstrate how to use @Autowired, @Required, and @Resource annotations to inject dependencies in a Spring-managed bean. What are the differences in their behavior during runtime?		
2	Implement a Spring application using @Component and @ComponentScan to automatically detect and register beans. How would you configure component scanning in a Java-based configuration class?		
3	Apply lifecycle annotations such as @PostConstruct and @PreDestroy in a Spring bean. Show how these annotations affect the bean lifecycle when using AnnotationConfigApplicationContext.		
4	Analyze the role of @Configuration in Java-based Spring configuration. How does it enable XML-free configuration, and what are the implications for maintainability and scalability?		
5	Compare and contrast the use of component filters in @ComponentScan with manual bean registration. How do include and exclude filters affect the application context setup? Provide examples to support your analysis.		

6	Implement a Spring application using @Component and @ComponentScan to automatically detect and register beans. How would you configure component scanning in a Java-based configuration class?		
7			
8	Summarize the role of lifecycle annotations such as @PostConstruct and @PreDestroy in managing bean initialization and destruction.		
9	After migrating, identify at least three maintainability benefits and three potential refactoring or scalability challenges introduced by moving to @Configuration classes. For each point, suggest a concrete mitigation or best practice.		
10	Write a minimal Spring app with @Component on a GreetingService and configure package scanning via a @Configuration class. Show the AppConfig and main that retrieves the bean and prints a greeting.		
11	Given packages com.nmit.service and com.nmit.web, configure @ComponentScan to scan only com.nmit.service. Add an excludeFilters rule to skip classes annotated with @Service. Show the exact annotation parameters and explain the effect at runtime.		
12	Given two beans EmailService and SmsService that both implement MessageService, show a controller with: ● Field A injected using @Autowired + @Qualifier("smsService") ● Field B injected using @Resource(name="emailService") ● Run-time: explain which bean each injection receives and why.		
13	Implement a cache bean that loads data in @PostConstruct and clears resources in @PreDestroy. Show how these hooks trigger when using AnnotationConfigApplicationContext in a try-with-resources main method.		
14	Differentiate between Multiview and MultipleController features of SpringMVC.		
15	Outline the features of spring data JPA along with **three** repositories of Spring Data JPA repository.		
15	How does Spring MVC handle the separation of concerns in web application development?		
16	You need to create a REST endpoint that returns a JSON representation of a `Product` entity. The endpoint should accept a `GET /products/{id}` request and support optional `fields` query parameter to return a subset of fields (e.g.,		

	`fields=name,price`). Sketch the controller method and outline how you would implement the field filtering.		
17	Compare `RestTemplate` with the newer `WebClient` (Spring WebFlux) for consuming REST services in a Spring MVC application. In what scenarios would you choose one over the other?		
18	You are designing a REST API for a microservice that serves JSON and XML representations of resources. Evaluate the trade-offs of using @RequestMapping(produces = { "application/json", "application/xml" }) with a single controller method versus creating separate methods annotated with @GetMapping(... produces = "...") for each media type.		
UNIT-IV			
1	You have a Spring Boot application that must run in multiple environments (dev, test, prod). How would you structure the configuration to use environment-specific values without changing code?		
2	What is Spring boot? List Features of Spring Boot that make it different?		
3	Illustrate the four key components of spring-boot. Explain in detail		
4	How the data exchange will be done in between JSON and XML,Write a suitable example		
5	How can I reload my Spring Boot changes without restarting the server ?		
6	How can we override default properties in Spring boot?		
7	Explain how Spring Boot creates and configures the application context. Given a `@Service` and a `@Repository` interacting with a `DataSource`, describe the bean lifecycle from startup to a simple service method invocation.		
8	How would you configure logging levels differently for development and production in Spring Boot? Provide concrete file snippets.		
9	You want to add data access with JDBC in a Spring Boot application. Which starter would you include, and how would you configure a DataSource with HikariCP?		
10	How does Spring Boot's component scanning work, and what would happen if a class annotated with `@Component` resides outside the scanned base package?		
12	Describe how a `@Repository` interface in Spring Data JPA translates into SQL operations. Provide an example with a custom query method.		
13	How do you expose JPA repository entities as REST resources using Spring Data REST? What dependencies and minimal configuration are required?		
14	You want to customize the base path of all Spring Data REST endpoints and disable certain HTTP methods for a given entity. How would you implement this?		

15	Explain how Spring Data REST maps repository methods to HTTP methods. If you add a derived query method in a repository, how is it exposed via REST?		
16	What are the default HTTP status codes for common operations in Spring Data REST (e.g., create, update, delete), and how can you customize them?		
17	How does Spring Data REST handle associations between entities? What happens when you fetch a parent resource—how are related child resources represented, and how can you customize association exposure?		
18	You need to expose a read-only view of users with only `id`, `name`, and `email`, and also filter users by status using a derived query method. How would you implement this with Spring Data REST?		
19	You're building a Spring Boot app with JPA entities and a REST layer exposed via Spring Data REST. A client needs to fetch a list of active products and simultaneously discover how to create a new product. Outline the series of steps the client would perform, including the endpoints and HTTP methods involved.		
20	What considerations would you take to ensure consistent transactional behavior when a REST client triggers a multi-step operation that involves multiple repositories/entities? Propose a design pattern using Spring Boot components.		
UNIT-V			
1	What are the core operation of DevOps with application development and infrastructure?		
2	Elaborate the principles of DevOps in detail?		
3	Differentiate between Agile Methodology and DevOps		
4	List the top DevOps tools. Which tools have you worked on?		
5	Explain the different phases in DevOps methodology ?		
6	What is the difference between continuous delivery and continuous deployment?		
7	What are the differences between continuous integration, continuous delivery, and continuous deployment (CI/CD)?		
8	Developer (Ram) has developed an application module and while developing he has made around 100 commits locally, now he need to push all his changes in remote develop branch as a single commit { we don't want developers messy history in develop branch} Hint:- Git squash		
9	Assume a business organization moves ahead in the competitive market and becomes more efficient in delivering the best features to the end-users in the set time. Mention some of the prime benefits a company can enjoy after adopting the DevOps way of working		

10	You’re tasked with designing a CI/CD pipeline for a microservices-based project that uses containers. Propose a pipeline that ensures each microservice builds independently, runs its own tests, and supports efficient deployments with minimal blast radius. Include: ● How you’d structure pipelines (per-service vs. monorepo) ● Build tooling and containerization strategy ● Testing strategy across services (unit, contract, integration) ● Deployment strategy (canary/blue-green) and rollback procedures		
11	As part of a security-conscious DevOps team, you need to enforce secure code practices on GitHub for a repository used by multiple teams. Propose concrete, implementable measures to achieve: ● Secure branching, PR review, and code ownership ● Secrets management and avoidance of leaking credentials ● Automated quality gates (linting, tests, security scans) and policy enforcement ● Handling of dependency updates and vulnerability remediation ● Incident response and post-mortem process		
12	Explain in your own words what a branch is in Git and why teams use branches during development.		
13	What is a pull request (PR) on GitHub, and what purpose does it serve in collaborative projects?		
14	Compare and contrast working locally with Git versus working remotely with Git. How do these workflows interact, and what potential issues might arise when synchronizing changes between local and remote repositories?		c