RBT Level : L1- Remembering, L2 – Understanding, L3 – Applying, L4 – Analyzing, L5 – Evaluating, L6 – Creati

PART – A (10x2=20 Marks) C

1. Define the concept of multi-tier web architecture. Provide an example illustrating the role of each tier in a typical enterprise application. C(

2. Differentiate between synchronous and asynchronous execution in Node.js with a real-time execution context. C(

3. Write the syntax and purpose of a basic route declaration in ExpressJS. C

4. Enumerate and briefly explain two categories of middleware in ExpressJS and their operational flow. C

5. Summarize the lifecycle of an HTTP request in Django and the framework's mechanisms for generating responses. C

6. In the context of high-volume transactional web apps, evaluate two advantages of SQLAlchemy over traditional ORMs in terms of performance and flexibility. C

7. Outline the commands required to create a database in MongoDB. Explain how these can be automated through scripting. C

8. You are tasked with deploying a globally accessible SaaS application. Justify the use of two specific services from MongoDB Atlas to ensure performance and scalability. C

9. Define Spring Boot and explain its benefits over the conventional Spring Framework in terms of developer productivity and configuration. C

10. What is the significance of integrating Hibernate with Log4j in a Spring application, and how does it benefit application monitoring? C

11. a) Apply the functionality of the V8 JavaScript engine in a Node.js-based streaming application. Illustrate how it influences request concurrency and execution performance.

(OR)

b) Evaluate the impact of using package-lock.json in CI/CD workflows. Construct a scenario demonstrating how it prevents version mismatch issues in multi-team environments.

12. a) Develop a route management module in ExpressJS capable of dynamically loading controllers. Explain how this design supports scalability and modularity in large applications.

(OR)

b) Analyze an ExpressJS shopping cart service where middleware fails during validation. Propose and implement a structured error-handling approach for such scenarios.

13. a) Apply SQLAlchemy within a Django application to manage complex relational models. Demonstrate CRUD operations and transaction handling using code snippets.

(OR)

b) Evaluate the trade-offs between using Django templates and frontend frameworks like React for dynamic page rendering. Provide a hybrid implementation strategy.

14. a) i) Analyze the connection flow between a Node.js server and MongoDB. Discuss common errors during deployment and how to resolve them using Mongoose best practices.

ii) Create a reusable Mongoose schema for logging user activities, ensuring timestamps and role-based access control are implemented efficiently.

(OR)

b) Construct a full migration pipeline to transition a normalized SQL schema into a MongoDB structure. Include schema transformation, indexing strategies, and performance tuning.

15. a) Evaluate and improve the setup process of Spring Tool Suite in corporate environments using containerization tools like Docker. Highlight dependencies and environment configurations.

(OR)

b) Design and implement a Hibernate-powered Spring Boot REST API to support product management with advanced filtering, validation, and error logging mechanisms.