Comprehension-Based Questions

1. What is the primary difference between authentication and authorization in an Express.js application?
2. Describe the analogy used to explain authentication and authorization in the context of a coffee shop.
3. List at least two common authentication methods supported in web applications.
4. What are the main steps in the authentication process in Express.js?
5. Name three common authorization models mentioned in the material.
6. How does the authorization process work in Express.js after a user is authenticated?
7. What is the purpose of using JWTs (JSON Web Tokens) in Express authentication?
8. In the provided code, what is the role of the `verifyUserToken` middleware?
9. How does the `IsAdmin` middleware function restrict access to certain routes?
10. Why should sensitive data never be stored in JWT payloads?
11. What are some best practices for securing authentication and authorization in Express.js applications?
12. How does middleware contribute to modular and reusable security logic in Express?
13. What is a Data Transfer Object (DTO) and why is it useful in Express.js applications?
14. Explain the suitcase analogy for DTOs and the role of `@Expose` and `@Exclude` decorators.
15. What does the `plainToClass` function from class-transformer do?
16. Why should you use `{ excludeExtraneousValues: true }` when converting objects to DTOs?
17. How does class-transformer help in shaping API responses for clients?
18. What is the recommended way to handle sensitive fields like passwords in DTO classes?
19. Why is it important to keep DTOs simple and free of business logic?
20. How can you use class-transformer to ensure only intended properties are included in API responses?