Transportation Fleet Management System (TFMS) – Interview Questions

1. Explain your TFMS project.

A fleet management system using Spring Boot, MVC, JPA, MySQL, JSP. Handles Vehicles, Trips, Fuel, Maintenance.

2. Why choose Spring Boot?

Auto-configuration, embedded Tomcat, fast development, easy integration.

3. Explain the architecture.

Controller → Service → Repository → JPA/Hibernate → MySQL → JSP.

4. Explain request flow for adding a Vehicle.

JSP form → POST /vehicles/add → Controller → Service → Repository → DB → Redirect → JSP.

5. What validations were applied?

@NotBlank, @NotNull, unique registration number, etc.

6. Explain @Controller vs @RestController.

@Controller returns JSP views, @RestController returns JSON.

7. Difference between @GetMapping and @PostMapping.

GET retrieves data; POST submits data.

8. Purpose of RedirectAttributes.

Used for flash messages after redirect.

9. Role of the Service layer.

Contains business logic, prevents fat controllers.

10. Explain JPA Entity mapping.

@Entity, @Table, @Id for primary keys, @OneToMany, @ManyToOne for relationships.

11. Explain OneToMany in Vehicle → Trip.

One vehicle can have many trips; Trip holds foreign key.

12. Why LocalDateTime instead of java.util.Date?

Better API, immutable, thread-safe.

13. Why JSTL?

For rendering data using loops, conditions, and EL expressions.

14. Why fmt:formatDate fails with LocalDate?

fmt expects java.util.Date. Solution: add formatted getters.

15. Explain database schema.

(vehicle, trip, fuel, maintenance) with foreign key constraints.

16. Handling delete with linked data.

Prevent deletion or handle cascade carefully.

17. Performance improvements.

Add indexes, pagination, caching, optimize SQL.

18. Convert to REST API.

Use @RestController and return ResponseEntity.

19. Add authentication.

Use Spring Security + JWT + roles.

20. Integrate GPS tracking.

Create GPS endpoint, push updates via WebSockets.