Java Web & Data Persistence Training Module

Project Case Study: Food Delivery Application (FoodieExpress)
Duration: 1-Day Intensive Training

==========
SERVLETS vs SPRING BOOT (IN-DEPTH COMPARISON)
==========

1. Architecture Level Difference

SERVLETS:
- Low-level Java web technology
- Developer manually handles:
  - Request mapping
  - Thread management
  - Response creation
  - Database connections
- Tight coupling between layers

SPRING BOOT:
- High-level framework built on top of Servlets
- Embedded server (Tomcat)
- Auto-configuration and dependency injection
- Clear separation of layers (Controller, Service, Repository)

---
2. Request Handling Comparison

SERVLET:
```
@WebServlet("/place-order")
public class OrderServlet extends HttpServlet {
   protected void doPost(...) { }
}
```

SPRING BOOT:
```
@PostMapping("/place-order")
public ResponseEntity<?> placeOrder(@RequestBody OrderRequest request) { }
```

Key Difference:

- Servlet directly handles HTTP objects
- Spring Boot abstracts HTTP into business-level objects

---
3. JDBC vs Spring Data JPA

SERVLET + JDBC:
- Driver loading required
- Manual SQL writing
- Manual resource closing
- Error-prone & verbose

SPRING BOOT + JPA:
- No driver loading
- No SQL (optional)
- Auto connection pooling
- Cleaner, readable code

---
4. Same Order Insert Example

JDBC CODE (~20 lines):
Class.forName(...)
Connection con = ...
PreparedStatement ps = ...
ps.executeUpdate()
con.close()

SPRING BOOT (1 line):
orderRepository.save(order);

---
5. Performance & Scalability

SERVLET:
- New DB connection per request
- No caching
- Hard to scale

SPRING BOOT:

- HikariCP connection pooling
- Lazy loading
- Cloud-ready

WHY LEARN SERVLETS FIRST?

- Understand what Spring Boot hides
- Debug production issues confidently
- Master fundamentals before frameworks