4. ☑ Integration Testing (JUnit + TestContainers)

✅ **This is the only use-case where IntelliJ may be used.**

- [x] Integration tests executed via IntelliJ
- [x] TestContainers used to spin up real MongoDB container
- [x] ProductServiceApplicationTests includes:
- [x] POST /api/product – Create Product
- [x] GET /api/product – Retrieve All Products
- [x] PUT /api/product/{id} – Update Product
- [x] DELETE /api/product/{id} – Delete Product
- [x] Tests pass and are visibly demonstrated
- [x] Use of assertions and repository checks shown

5. 🗀 Submission Deliverables

- [x] ✅ 1. Completed ICE 1/Lab 1 Status Checklist (this very checklist)
- [x] ✅ 2. URL to GitLab private repo (**Reporter** access provided)
- [x] ✅ 3. ICE 1/Lab 1 demonstration video (≤ 5 minutes)

6. 💭 Wrap-Up

- [x] Student reflects on what they learned
- [ ] (Optional) Share biggest challenge or most rewarding moment

COMP3095 – Lab 1 Demonstration Checklist

Project Name: microservices-parent
Module: product-service
Course: COMP3095
Lab: ICE 1 /Lab 1 – CRUD Microservice with MongoDB
Video Duration Limit: ≤ 5 minutes

1. Introduction Slide (Start of Video)

- [✓] Names, Student IDs, Course Code, and Lab Title
- [✓] Real photos (no avatars)
- [✓] Title: 'Lab 1 – Product Service Microservice'
- [✓] Technologies Used: Spring Boot, MongoDB, Docker, TestContainers

2. GitLab Project Setup

- [✓] GitLab private repository created
- [✓] Professors added with **Reporter** access

3. Environment Setup & API Testing (Postman)

- [✓] Run the product-service in IntelliJ for testing purposes.
- [✓] All action in endpoint must be tested using Postman.
- [✓] Docker containers started (mongo, mongo-express)
- [✓] Demonstrate container status using **Docker Desktop**
- [✓] API Testing done via Postman:
- [✓] POST /api/product – Create Product
  - [✓] GET /api/product – Retrieve All Products
  - [✓] PUT /api/product/{id} – Update Product
  - [✓] DELETE /api/product/{id} – Delete Product
  - [✓] Correct status codes shown (201, 200, 204)