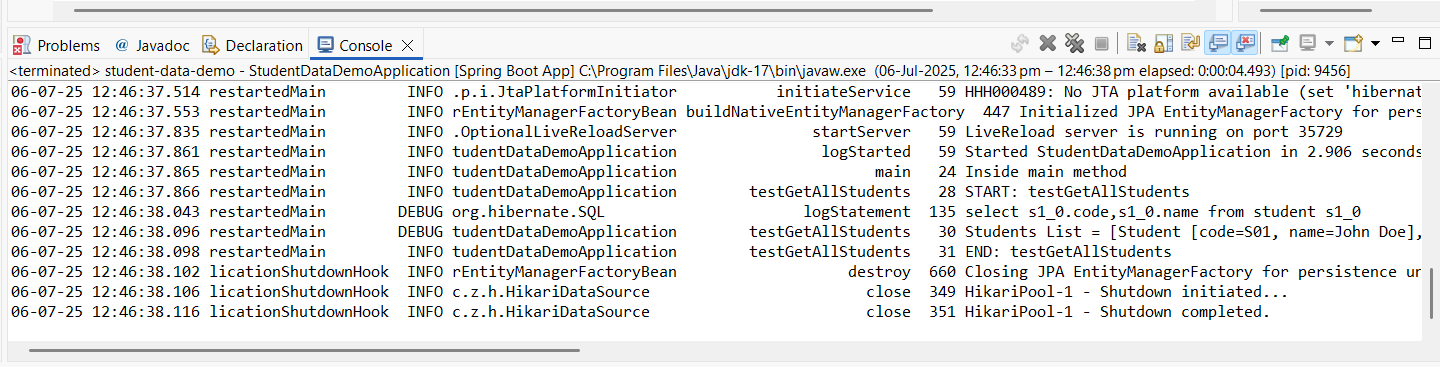
**Spring Data JPA with Hibernate**

**Output Screens**

**Hands on-1**

**Hands on-4**

**Difference between JPA, Hibernate and Spring Data JPA**

| **Feature** | **JPA (Java Persistence API)** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| **Type** | Specification (Interface/API) | Implementation (Framework) | Abstraction built on top of JPA |
| **Purpose** | Defines standard for ORM in Java | Provides actual ORM functionality | Simplifies JPA-based persistence using Spring |
| **Provider** | Maintained by Java EE / Jakarta EE | Developed by Red Hat | Maintained by the Spring Team |
| **Dependency** | javax.persistence / jakarta.persistence | org.hibernate | org.springframework.data.jpa |
| **Code Boilerplate** | Requires more manual coding | Slightly less than JPA | Minimizes boilerplate with repository interfaces and custom queries |
| **Query Language** | JPQL (Java Persistence Query Language) | Supports JPQL and HQL (Hibernate Query Language) | Uses JPQL, native SQL, and derived query methods from method names |
| **Ease of Use** | Moderate — needs EntityManager, etc. | Easier than plain JPA | Very easy — most CRUD operations are automatic |
| **Custom Queries** | Must be written manually | Can use HQL/Criteria API | Supports @Query, native queries, method naming for automatic query generation |
| **Transaction Management** | Manual or container-managed | Supports manual and JTA | Integrated with Spring’s transaction management |
| **Vendor Neutrality** | Yes — designed to be implemented by many (Hibernate, EclipseLink, etc.) | No — Hibernate-specific features | No — ties to Spring ecosystem |
| **Main Advantage** | Platform-independent ORM standard | Powerful and flexible ORM engine | Rapid development with minimal configuration |