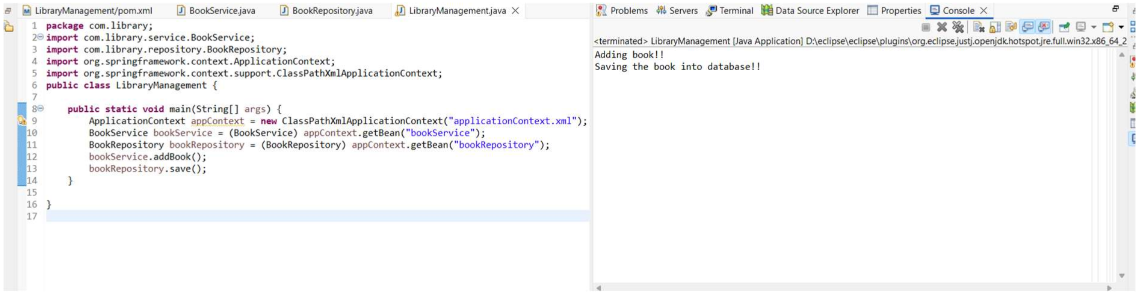
Spring Core and Maven

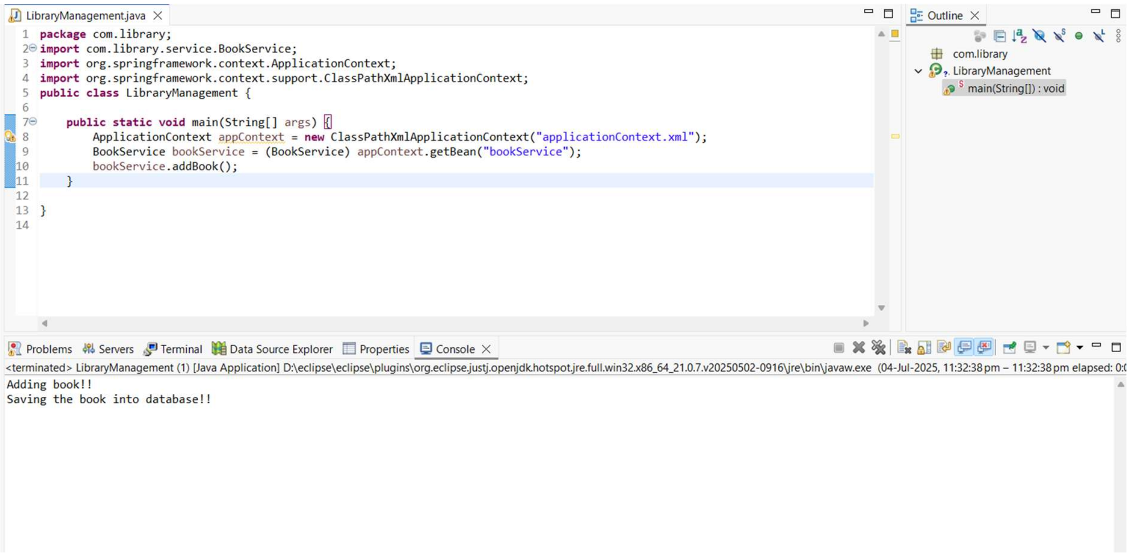
Exercise 1: Configuring a Basic Spring Application



```java
package com.library;
import com.library.service.BookService;
import com.library.repository.BookRepository;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class LibraryManagement {

    public static void main(String[] args) {
        ApplicationContext appContext = new ClassPathXmlApplicationContext("applicationContext.xml");
        BookService bookService = (BookService) appContext.getBean("bookService");
        BookRepository bookRepository = (BookRepository) appContext.getBean("bookRepository");
        bookService.addBook();
        bookRepository.save();
    }

}
```

```
<terminated> LibraryManagement [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
Adding book!!
Saving the book into database!!
```

Exercise 2: Implementing Dependency Injection

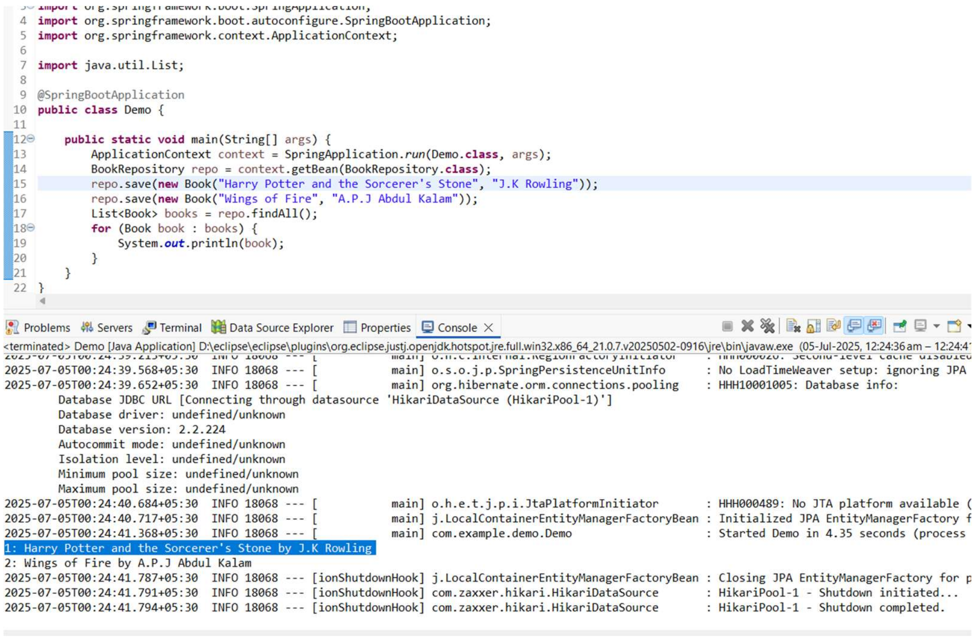


```java
package com.library;
import com.library.service.BookService;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class LibraryManagement {

    public static void main(String[] args) {
        ApplicationContext appContext = new ClassPathXmlApplicationContext("applicationContext.xml");
        BookService bookService = (BookService) appContext.getBean("bookService");
        bookService.addBook();
    }

}
```

```
<terminated> LibraryManagement (1) [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe  (04-Jul-2025, 11:32:38 pm – 11:32:38 pm elapsed: 0:0
Adding book!!
Saving the book into database!!
```

Spring data JPA with Spring boot, Hibernate

Exercise 1: Spring data JPA – Quick Example



```java
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;

import java.util.List;

@SpringBootApplication
public class Demo {

    public static void main(String[] args) {
        ApplicationContext context = SpringApplication.run(Demo.class, args);
        BookRepository repo = context.getBean(BookRepository.class);
        repo.save(new Book("Harry Potter and the Sorcerer's Stone", "J.K Rowling"));
        repo.save(new Book("Wings of Fire", "A.P.J Abdul Kalam"));
        List<Book> books = repo.findAll();
        for (Book book : books) {
            System.out.println(book);
        }
    }
}
```

```
<terminated> Demo [Java Application] D:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bin\javaw.exe  (05-Jul-2025, 12:24:36 am – 12:24:4
2025-07-05T00:24:39.215+05:30  INFO 18068 --- [           main] o.h.c.internal.RegionFactoryInitiator    : HHH000026: Second-level cache disabled
2025-07-05T00:24:39.568+05:30  INFO 18068 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver setup: ignoring JPA
2025-07-05T00:24:39.652+05:30  INFO 18068 --- [           main] org.hibernate.orm.connections.pooling    : HHH10001005: Database info:
	Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']
	Database driver: undefined/unknown
	Database version: 2.2.224
	Autocommit mode: undefined/unknown
	Isolation level: undefined/unknown
	Minimum pool size: undefined/unknown
	Maximum pool size: undefined/unknown
2025-07-05T00:24:40.684+05:30  INFO 18068 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489: No JTA platform available (
2025-07-05T00:24:40.717+05:30  INFO 18068 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory f
2025-07-05T00:24:41.368+05:30  INFO 18068 --- [           main] com.example.demo.Demo                    : Started Demo in 4.35 seconds (process
1: Harry Potter and the Sorcerer's Stone by J.K Rowling
2: Wings of Fire by A.P.J Abdul Kalam
2025-07-05T00:24:41.787+05:30  INFO 18068 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for p
2025-07-05T00:24:41.791+05:30  INFO 18068 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2025-07-05T00:24:41.794+05:30  INFO 18068 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
```

Exercise 2: Difference between JPA, Hibernate and Spring Data JPA

1. Java Persistence API(JPA)

- JPA is a specification that defines how Java objects are mapped to relational databases
- It does not have its own implementation
- It provides annotations such as @Entity, @Id.

Example –

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("");
EntityManager em = emf.createEntityManager();
em.getTransaction().begin();
em.persist(new Employee("John"));
em.getTransaction().commit();
em.close();
```

2. Hibernate

- Hibernate is a ORM (Object Relational Model) that implements JPA
- It has additional features like caching, HQL
- We use Session instead of EntityManager

Example –

```
Session session = sessionFactory.openSession();
Transaction tx = session.beginTransaction();
session.save(new Employee("Sai"));
tx.commit();
session.close();
```

3. Spring Data JPA

- It is not a JPA implementation
- It uses spring dependency injection and JpaRepository to provide ready-made CRUD operations
- SQL code is hardly used

Example-

```
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {}

@Autowired

private EmployeeRepository repo;

repo.save(new Employee("Sai"));
```