

HANDSON EXERCISES - WEEK 2

Skill : Spring Core and Maven

Exercise 1: Configuring a Basic Spring Application

Scenario:

Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.5.3</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>LibraryManagement</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>LibraryManagement</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
  </scm>
```

```
<properties>
  <java.version>21</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter</artifactId>
  </dependency>
```

```
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-test</artifactId>
      <scope>test</scope>
    </dependency>
  </dependencies>
```

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>
```

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">
```

```
<bean id="bookRepository" class="com.library.repository.BookRepository"/>
```

```
<bean id="bookService" class="com.library.service.BookService">
  <property name="bookRepository" ref="bookRepository"/>
</bean>
</beans>
```

BookService.java

```
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void addBook(String bookName) {
        bookRepository.saveBook(bookName);
        System.out.println("Book added successfully: " + bookName);
    }
}
```

BookRepository.java

```
package com.library.repository;

public class BookRepository {
    public void saveBook(String bookName) {
        System.out.println("Book saved: " + bookName);
    }
}
```

LibraryManagementApplication.java

```
package com.example.LibraryManagement;

import com.library.service.BookService;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

@SpringBootApplication
public class LibraryManagementApplication {
```

```
public static void main(String[] args) {
    SpringApplication.run(LibraryManagementApplication.class, args);
    ApplicationContext context = new
ClassPathXmlApplicationContext("applicationContext.xml");
    BookService bookService = context.getBean("bookService",
BookService.class);
    bookService.addBook("Effective Java");
}
```

OUTPUT :

```

"C:\Program Files\Java\jdk-21\bin\java.exe" ...

  _   _          _    _
 / \ / \ _ _ _ _ _ ( ) _ _ _ _ \ \ \ \
( ( ) \ _ _ | ' _ | ' _ | ' _ \ _ \ \ \ \
\ \ / _ _ | | _ | | | | | ( _ | ) ) ) )
 ' | _ _ | _ _ | | _ | | \ _ _ | / / / /
=====|_|=====|___/ _ / _ / _

:: Spring Boot ::                (v3.5.3)

2025-07-02T14:21:27.730+05:30 INFO 7700 --- [LibraryManagement] [    main] c.e.l.LibraryManagementApplication : Starting LibraryManagementApplication using Java
2025-07-02T14:21:27.735+05:30 INFO 7700 --- [LibraryManagement] [    main] c.e.l.LibraryManagementApplication : No active profile set, falling back to 1 default
2025-07-02T14:21:28.433+05:30 INFO 7700 --- [LibraryManagement] [    main] c.e.l.LibraryManagementApplication : Started LibraryManagementApplication in 1.311 sec

Book saved: Effective Java
Book added successfully: Effective Java

Process finished with exit code 0

```

Exercise 5: Configuring the Spring IoC Container**Scenario:**

The library management application requires a central configuration for beans and dependencies.

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd">

    <bean id="bookRepository" class="com.library.repository.BookRepository"/>

    <bean id="bookService" class="com.library.service.BookService">
        <property name="bookRepository" ref="bookRepository"/>
    </bean>

</beans>
```

BookService.java

```
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void addBook(String bookName) {
        bookRepository.saveBook(bookName);
        System.out.println("Book added successfully: " + bookName);
    }
}
```

BookRepository.java

```
public class BookRepository {
    public void saveBook(String bookName) {
        System.out.println("Book saved: " + bookName);
    }
}
```

```
public static void main(String[] args) {
    SpringApplication.run(LibraryManagementApplication.class, args);
    ApplicationContext context = new
    ClassPathXmlApplicationContext("applicationContext.xml");
    BookService bookService = context.getBean("bookService",
    BookService.class);
    bookService.addBook("Effective Java");
}
```

```
C:\Program Files\Java\jdk-21\bin\java.exe" ...  
  
 _   _      _ _      _ _ _ _  
/ \ / ___'_ _ _ _ _ _ _ _ _ _  
( ) _ _ _ | '_ | '_ | '_ V _' | _ _ _  
\V / _ _ | |_) | | | | | | ( | | ) ) )  
 ' | _ _ | |_) | | | | | _ _ | / / / /  
=====|_|=====|_/_=/././  
  
:: Spring Boot ::                (v3.5.3)  
  
2025-07-02T14:44:11.965+05:30 INFO 19792 --- [LibraryManagement] [main] c.e.L.LibraryManagementApplication : Starting LibraryManagementApplication using Java  
2025-07-02T14:44:11.970+05:30 INFO 19792 --- [LibraryManagement] [main] c.e.L.LibraryManagementApplication : No active profile set, falling back to 1 default  
2025-07-02T14:44:12.698+05:30 INFO 19792 --- [LibraryManagement] [main] c.e.L.LibraryManagementApplication : Started LibraryManagementApplication in 1.361 se  
Book saved: Effective Java  
Book added successfully: Effective Java  
  
Process finished with exit code 0
```

Exercise 2: Implementing Dependency Injection**Scenario:**

In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xsi:schemaLocation="  
        http://www.springframework.org/schema/beans  
        http://www.springframework.org/schema/beans/spring-beans.xsd">
```

```
    <bean id="bookRepository" class="com.library.repository.BookRepository"/>
```

```
    <bean id="bookService" class="com.library.service.BookService">  
        <property name="bookRepository" ref="bookRepository"/>  
    </bean>
```

```
</beans>
```

BookService.java

```
package com.library.service;
```

```
import com.library.repository.BookRepository;
```

```
public class BookService {  
    private BookRepository bookRepository;
```

```
    public void setBookRepository(BookRepository bookRepository) {  
        this.bookRepository = bookRepository;  
    }
```

```
    public void addBook(String bookName) {  
        bookRepository.saveBook(bookName);  
        System.out.println("Book added successfully: " + bookName);  
    }
```

BookRepository.java

```
public class BookRepository {  
    public void saveBook(String bookName) {  
        System.out.println("Book saved: " + bookName);  
    }  
}
```

```
public static void main(String[] args) {
    SpringApplication.run(LibraryManagementApplication.class, args);
    ApplicationContext context = new
    ClassPathXmlApplicationContext("applicationContext.xml");
    BookService bookService = context.getBean("bookService",
    BookService.class);
    bookService.addBook("Effective Java");
}
```

```
C:\Program Files\Java\jdk-21\bin\java.exe" ...  
  
 _   _      _         _   _     _   _  
/\ / ____'__--_(_)--_--_\ \ \ \  
( )_____| '_|'_||'_|\V_' | \ \ \  
\V ___||_| ||| |||(| | )) ) )  
' |_____||_|_|_|_|_|_/ // //  
=====|_|=====|___/=/_./._/  
  
:: Spring Boot ::                (v3.5.3)  
  
2025-07-02T14:44:11.965+05:30 INFO 19792 --- [LibraryManagement] [          main] c.e.L.LibraryManagementApplication : Starting LibraryManagementApplication using Java  
2025-07-02T14:44:11.970+05:30 INFO 19792 --- [LibraryManagement] [          main] c.e.L.LibraryManagementApplication : No active profile set, falling back to 1 default  
2025-07-02T14:44:12.698+05:30 INFO 19792 --- [LibraryManagement] [          main] c.e.L.LibraryManagementApplication : Started LibraryManagementApplication in 1.361 se  
Book saved: Effective Java  
Book added successfully: Effective Java  
  
Process finished with exit code 0
```


Exercise 7: Implementing Constructor and Setter Injection**Scenario:**

The library management application requires both constructor and setter injection for better control over bean initialization.

applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">

  <bean id="bookRepository" class="com.library.repository.BookRepository"/>

  <bean id="bookService" class="com.library.service.BookService">
    <property name="bookRepository" ref="bookRepository"/>
  </bean>

</beans>
```

BookService.java

```
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
  private BookRepository bookRepository;

  public void setBookRepository(BookRepository bookRepository) {
    this.bookRepository = bookRepository;
  }

  public void addBook(String bookName) {
    bookRepository.saveBook(bookName);
    System.out.println("Book added successfully: " + bookName);
  }
}
```

BookRepository.java

```
public class BookRepository {
    public void saveBook(String bookName) {
        System.out.println("Book saved: " + bookName);
    }
}
```

```
public static void main(String[] args) {
    SpringApplication.run(LibraryManagementApplication.class, args);
    ApplicationContext context = new
    ClassPathXmlApplicationContext("applicationContext.xml");
    BookService bookService = context.getBean("bookService",
    BookService.class);
    bookService.addBook("Effective Java");
}
```

[illegible]

Scenario:

You need to set up a new Maven project for the library management application and add Spring dependencies.

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.5.3</version>
    <relativePath/>
  </parent>

  <groupId>com.example</groupId>
  <artifactId>LibraryManagement</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>LibraryManagement</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
    <developer/>
  </developers>
  <scm>
    <connection/>
    <developerConnection/>
    <tag/>
    <url/>
  </scm>

  <properties>
    <java.version>21</java.version>
```

```
</properties>
```

```
<dependencies>
```

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-test</artifactId>
```

```
<scope>test</scope>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-context</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-aop</artifactId>
```

```
</dependency>
```

```
<dependency>
```

```
<groupId>org.springframework</groupId>
```

```
<artifactId>spring-webmvc</artifactId>
```

```
</dependency>
```

```
</dependencies>
```

```
<build>
```

```
<plugins>
```

```
<plugin>
```

```
<groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-maven-plugin</artifactId>
```

```
</plugin>
```

```
</plugins>
```

```
</build>
```

```
</project>
```

Scenario:

You need to create a Spring Boot application for the library management system to simplify configuration and deployment.



Project
☐ Gradle - Groovy ☐ Gradle - Kotlin ☒ **Java** ☐ Kotlin ☐ Groovy
☒ **Maven**

Spring Boot
☐ 4.0.0 (SNAPSHOT) ☐ 3.5.4 (SNAPSHOT) ☒ **3.5.3** ☐ 3.4.8 (SNAPSHOT)
☐ 3.4.7 ☐ 3.3.13

Project Metadata
Group
Artifact
Name
Description
Package name
Packaging ☒ **Jar** ☐ War
Java ☐ 24 ☒ **21** ☐ 17

Dependencies ADD DEPENDENCIES... CTRL + B

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database SQL
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.5.3</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.example</groupId>
  <artifactId>LibraryManagement</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>LibraryManagement</name>
  <description>Demo project for Spring Boot</description>
  <url/>
  <licenses>
    <license/>
  </licenses>
  <developers>
```

```
<developer/>
</developers>
<scm>
  <connection/>
  <developerConnection/>
  <tag/>
  <url/>
</scm>
<properties>
  <java.version>21</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
```

```
  <dependency>
    <groupId>com.h2database</groupId>
    <artifactId>h2</artifactId>
    <scope>runtime</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
</project>
```

application.properties:

```
spring.application.name=LibraryManagement
spring.datasource.url=jdbc:h2:mem:librarydb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.hibernate.ddl-auto=update
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
```

Book.java

```
package com.example.LibraryManagement.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private String title;
    private String author;

    public Book() {}

    public Book(String title, String author) {
        this.title = title;
        this.author = author;
    }

    // Getters and Setters
    public Long getId() { return id; }
    public void setId(Long id) { this.id = id; }

    public String getTitle() { return title; }
```

```
public void setTitle(String title) { this.title = title; }
```

```
public String getAuthor() { return author; }  
public void setAuthor(String author) { this.author = author; }  
}
```

BookRepository.java

```
package com.example.LibraryManagement.repository;
```

```
import com.example.LibraryManagement.model.Book;  
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface BookRepository extends JpaRepository<Book, Long> {  
}
```

BookController.java

```
package com.example.LibraryManagement.controller;
```

```
import com.example.LibraryManagement.model.Book;  
import com.example.LibraryManagement.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.*;
```

```
import java.util.List;
```

```
@RestController  
@RequestMapping("/api/books")  
public class BookController {
```

```
@Autowired  
private BookRepository bookRepository;
```

```
@GetMapping  
public List<Book> getAllBooks() {  
    return bookRepository.findAll();  
}
```

```
@PostMapping  
public Book createBook(@RequestBody Book book) {  
    return bookRepository.save(book);  
}
```



```
@GetMapping("/{id}")
public Book getBookById(@PathVariable Long id) {
    return bookRepository.findById(id).orElse(null);
}
```

```
@PutMapping("/{id}")
public Book updateBook(@PathVariable Long id, @RequestBody Book
updatedBook) {
    return bookRepository.findById(id).map(book -> {
        book.setTitle(updatedBook.getTitle());
        book.setAuthor(updatedBook.getAuthor());
        return bookRepository.save(book);
    }).orElse(null);
}
```

```
@DeleteMapping("/{id}")
public void deleteBook(@PathVariable Long id) {
    bookRepository.deleteByld(id);
}
}
```

LibraryManagementApplication.java

```
package com.example.LibraryManagement;
```

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
public class LibraryManagementApplication {
```

```
    public static void main(String[] args) {
        SpringApplication.run(LibraryManagementApplication.class, args);
    }
```

```
}
```

```
C:\Program Files\Java\jdk-21\bin\java.exe ...
```

```
.   _.._          ._. .._      \___/_
/\ / ___'__.'(C)_____\_____\\ \\ \\ 
(C)_...'_-'_'_'_'_'_'_'V'-'''\_\\\_\\ 
VV ____|_|_|_|_|_|_|_|_|_|)) ) ) ) )
    '_||_|_|_|_|_|_|_|_|_|_/ // // /
=====|_|=====|_|=/-/_-//_-//

:: Spring Boot ::                (v3.5.3)
```

```
2025-07-06T22:05:16.485+05:30 INFO 11248 --- [LibraryManagement] [ main] c.e.l.LibraryManagementApplication : Starting LibraryManagementApplication using Java
2025-07-06T22:05:16.489+05:30 INFO 11248 --- [LibraryManagement] [ main] o.a.c.LibraryManagementApplication : No active profile set, falling back to default
2025-07-06T22:05:17.368+05:30 INFO 11248 --- [LibraryManagement] [ main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in D
2025-07-06T22:05:17.428+05:30 INFO 11248 --- [LibraryManagement] [ main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 55
2025-07-06T22:05:18.100+05:30 INFO 11248 --- [LibraryManagement] [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 9090 (http)
2025-07-06T22:05:18.112+05:30 INFO 11248 --- [LibraryManagement] [ main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2025-07-06T22:05:18.112+05:30 INFO 11248 --- [LibraryManagement] [ main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.42
2025-07-06T22:05:18.174+05:30 INFO 11248 --- [LibraryManagement] [ main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2025-07-06T22:05:18.175+05:30 INFO 11248 --- [LibraryManagement] [ main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization complete
2025-07-06T22:05:18.510+05:30 INFO 11248 --- [LibraryManagement] [ main] o.hibnerna.jpse.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name:
2025-07-06T22:05:18.589+05:30 INFO 11248 --- [LibraryManagement] [ main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.6.18.Fin
2025-07-06T22:05:18.641+05:30 INFO 11248 --- [LibraryManagement] [ main] o.h.c.internal.RegionFactoryInitiator : HHH000626: Second-level cache disabled
2025-07-06T22:05:19.020+05:30 INFO 11248 --- [LibraryManagement] [ main] o.o.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup; ignoring JPA class tran
2025-07-06T22:05:19.058+05:30 INFO 11248 --- [LibraryManagement] [ main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2025-07-06T22:05:19.271+05:30 INFO 11248 --- [LibraryManagement] [ main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn0: url=jdbc:
2025-07-06T22:05:19.273+05:30 INFO 11248 --- [LibraryManagement] [ main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2025-07-06T22:05:19.356+05:30 INFO 11248 --- [LibraryManagement] [ main] org.hibernate.orm.connections.pooling : HHH1000100S: Database info:
```

```
Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']
Database driver: undefined/unknown
Database version: 2.3.232
Autocommit mode: undefined/unknown
Isolation level: undefined/unknown
Minimum pool size: undefined/unknown
Maximum pool size: undefined/unknown

2025-07-06T22:05:20.080+05:30 INFO 11248 --- [LibraryManagement] [ main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibe
2025-07-06T22:05:20.143+05:30 INFO 11248 --- [LibraryManagement] [ main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persis
2025-07-06T22:05:20.901+05:30 INFO 11248 --- [LibraryManagement] [ main] o.s.b.a.h2.H2ConsoleAutoConfiguratio : H2 console available at '/h2-console/'. Database
2025-07-06T22:05:21.051+05:30 INFO 11248 --- [LibraryManagement] [ main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 9090 (http) with context
2025-07-06T22:05:21.063+05:30 INFO 11248 --- [LibraryManagement] [ main] c.e.l.LibraryManagementApplication : Started LibraryManagementApplication in 5.135 s
```