

### 1. Problem Statement: Library Management System

Build a system to manage library operations including book lending, returns, and inventory management.

**Answer :**

```
package com.example.library;
```

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.provisioning.InMemoryUserDetailsManager;
import org.springframework.security.web.SecurityFilterChain;
```

```
import javax.persistence.*;
import java.time.LocalDate;
import java.util.List;
```

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

```
@Entity
class Book {
    @Id @GeneratedValue
    private Long bookId;
    private String title;
    private String author;
    private String category;
    private boolean available = true;
    // getters and setters
}
```

```
@Entity
```

```
class User {
    @Id @GeneratedValue
    private Long userId;
    private String name;
    private String membershipType;
    // getters and setters
}
```

@Entity

```
class Transaction {
    @Id @GeneratedValue
    private Long transactionId;

    @ManyToOne
    private Book book;

    @ManyToOne
    private User user;

    private LocalDate issueDate;
    private LocalDate returnDate;
    private String status;
    // getters and setters
}
```

```
interface BookRepository extends org.springframework.data.jpa.repository.JpaRepository<Book,
Long> {
    List<Book> findByTitleContaining(String title);
    List<Book> findByAuthor(String author);
    List<Book> findByCategory(String category);
}
```

```
interface UserRepository extends org.springframework.data.jpa.repository.JpaRepository<User,
Long> {}
```

```
interface TransactionRepository extends
org.springframework.data.jpa.repository.JpaRepository<Transaction, Long> {
    List<Transaction> findByUser(User user);
    List<Transaction> findByStatusAndReturnDateBefore(String status, LocalDate date);
}
```

@Configuration

@EnableWebSecurity

class SecurityConfig {

    @Bean

    public InMemoryUserDetailsManager userDetailsService() {

        UserDetails librarian = User.withDefaultPasswordEncoder()

            .username("librarian")

            .password("password")

            .roles("LIBRARIAN")

            .build();

        return new InMemoryUserDetailsManager(librarian);

    }

    @Bean

    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {

        http.csrf().disable()

            .authorizeRequests()

            .anyRequest().authenticated()

            .and()

            .formLogin();

        return http.build();

    }

}