Sure, Varun! Here's a detailed project structure for your online bookstore using Maven, Spring, and JdbcTemplate.

**Project Structure**

css

Copy code

onlinebookstore

├── pom.xml

└── src

└── main

├── java

│ └── com

│ └── flm

│ ├── dao

│ │ ├── AdminDao.java

│ │ ├── AdminDaoImpl.java

│ │ ├── UserDao.java

│ │ ├── UserDaoImpl.java

│ │ ├── BookDao.java

│ │ └── BookDaoImpl.java

│ ├── controller

│ │ └── AdminController.java

│ ├── enums

│ │ ├── PaymentMethod.java

│ │ ├── OrderStatus.java

│ │ ├── Gender.java

│ │ └── BookStatus.java

│ ├── model

│ │ ├── Admin.java

│ │ ├── User.java

│ │ ├── Book.java

│ │ ├── Order.java

│ │ ├── Review.java

│ │ ├── Category.java

│ │ ├── Cart.java

│ │ └── Payment.java

│ └── service

│ ├── AdminService.java

│ ├── AdminServiceImpl.java

│ ├── UserService.java

│ ├── UserServiceImpl.java

│ └── BookService.java

│ └── BookServiceImpl.java

├── resources

│ ├── application.properties

│ └── online-servlet.xml

└── webapp

└── WEB-INF

├── web.xml

└── jsp

├── login.jsp

├── home.jsp

├── bookDetails.jsp

└── adminDashboard.jsp

**File Details**

**1. pom.xml**

This is the configuration file for Maven, where all dependencies (Spring, JDBC, etc.) are added. You can refer to the earlier pom.xml I provided for all required dependencies.

**2. src/main/java/com/flm/dao/**

Contains DAO (Data Access Object) interfaces and their implementations, which are responsible for interacting with the database.

* **AdminDao.java**  
  Interface for admin-related database operations.
* **AdminDaoImpl.java**  
  Implements AdminDao and provides logic for creating, updating, deleting, and retrieving admin details from the database.
* **UserDao.java**  
  Interface for user-related database operations.
* **UserDaoImpl.java**  
  Implements UserDao for handling user data (CRUD operations).
* **BookDao.java**  
  Interface for book-related database operations.
* **BookDaoImpl.java**  
  Implements BookDao for managing books in the database (CRUD operations).

**3. src/main/java/com/flm/controller/**

Contains controllers that handle the HTTP requests and responses (if you add REST API or JSP-based views).

* **AdminController.java**  
  Controller to handle admin-related HTTP requests (admin login, dashboard, etc.).

**4. src/main/java/com/flm/enums/**

Contains enum classes for various statuses and categories.

* **PaymentMethod.java**  
  Enum for payment methods like Credit Card, Debit Card, Wallet, etc.
* **OrderStatus.java**  
  Enum for order statuses: completed, pending, cancelled.
* **Gender.java**  
  Enum for gender values: male, female.
* **BookStatus.java**  
  Enum for book statuses: active, inactive.

**5. src/main/java/com/flm/model/**

Contains model classes that represent the database entities.

* **Admin.java**  
  Represents an admin entity.
* **User.java**  
  Represents a user entity.
* **Book.java**  
  Represents a book entity.
* **Order.java**  
  Represents an order entity.
* **Review.java**  
  Represents a review entity for books.
* **Category.java**  
  Represents a book category.
* **Cart.java**  
  Represents the cart entity for storing user's cart items.
* **Payment.java**  
  Represents a payment entity related to orders.

**6. src/main/java/com/flm/service/**

Contains service classes, which contain the business logic and call the DAO layer to perform operations.

* **AdminService.java**  
  Interface for the service layer, defining methods for admin operations.
* **AdminServiceImpl.java**  
  Implements AdminService to manage admin business logic.
* **UserService.java**  
  Interface for the service layer, defining methods for user operations.
* **UserServiceImpl.java**  
  Implements UserService for handling user-related business logic.
* **BookService.java**  
  Interface for the service layer, defining methods for book operations.
* **BookServiceImpl.java**  
  Implements BookService to manage book-related business logic.

**7. src/main/resources/**

Contains application configuration files and the Spring configuration XML file.

* **application.properties**  
  Contains properties like database connection URL, username, password, etc.
* **online-servlet.xml**  
  Spring configuration file, where beans like JdbcTemplate, DAOs, services, and data sources are configured.

**8. src/main/webapp/WEB-INF/**

Contains web application-specific resources like web.xml and JSP views.

* **web.xml**  
  The deployment descriptor for your web application, defining servlet mappings, filters, and other configurations.
* **jsp/**  
  Contains all JSP pages for your web UI.
  + login.jsp  
    Page for user login.
  + home.jsp  
    Main page of the bookstore.
  + bookDetails.jsp  
    Displays details of a selected book.
  + adminDashboard.jsp  
    Admin interface to manage users, books, orders, etc.