电子书阅读平台项目说明文档

## 一、项目概述

本项目是基于 Spring Boot 开发的轻量化电子书阅读平台，核心目标是打通 “读者阅读” 与 “作者创作” 的双向需求：

* 对读者：提供电子书浏览、搜索、分类查看、在线阅读等功能，满足便捷阅读需求；
* 对作者：支持作品创建、章节发布 / 管理、数据统计（阅读量、收藏量、收入）等操作，辅助作者高效创作。

平台整体设计遵循 “简洁易用、性能稳定” 原则，适配 PC 端主流浏览器，确保不同角色用户的操作体验。

## 二、技术栈与环境要求

### 1. 技术栈明细

|  |  |  |
| --- | --- | --- |
| 层面 | 技术 / 工具 | 说明 |
| 后端 | Spring Boot 2.7.x | 快速搭建服务端，简化配置与依赖管理 |
|  | Spring Data JPA | 简化数据库 CRUD 操作，减少重复代码 |
| 前端 | Thymeleaf | 动态渲染 HTML 页面，实现前后端数据交互 |
|  | HTML/CSS/JavaScript | 构建页面结构与交互逻辑 |
|  | 第三方库 | Lucide（图标）、Chart.js（数据可视化） |
| 数据库 | MySQL 8.0+ | 存储用户、作品、章节等核心数据 |
| 构建工具 | Maven 3.6+ | 项目编译、依赖管理、打包 |

### 2. 运行环境要求

* 操作系统：Windows 10/11；
* JDK 版本：JDK 11；
* 数据库：MySQL 8.0 ；
* 浏览器：Chrome 90+、Edge 90+、Firefox 88+。

## 三、项目结构

### 1. 代码包结构（后端）

|  |
| --- |
| com.example.zjsu  ├─ ZjsuApplication.java // 项目启动类（含main方法）  ├─ controller // 控制层：处理前端请求，返回视图/数据  │ ├─ ChapterController.java // 章节管理：查询、发布、编辑、删除章节  │ ├─ EbookController.java // 作品管理：创建、编辑、查询电子书  │ ├─ PageController.java // 页面跳转：首页、登录页、详情页等通用跳转  │ └─ UserController.java // 用户管理：注册、登录、个人信息维护  ├─ entity // 实体层：与数据库表一一对应  │ ├─ ChapterInfo.java // 章节实体：章节ID、标题、内容、所属作品ID等  │ ├─ EbookInfo.java // 作品实体：作品ID、标题、简介、类型、作者ID等  │ └─ UserInfo.java // 用户实体：用户ID、用户名、密码、角色（读者/作者）等  └─ repository // 数据访问层：继承JpaRepository，操作数据库  ├─ ChapterInfoRepository.java  ├─ EbookInfoRepository.java  └─ UserInfoRepository.java |

### 2. 资源文件结构（前端）

|  |
| --- |
| resources  ├─ static // 静态资源：无需动态渲染的文件  │ └─ images // 图片资源：作品封面、用户头像、默认图标等  └─ templates // 模板文件：Thymeleaf页面，动态生成前端界面  ├─ author-center.html // 作者中心：作品管理、数据统计、章节发布  ├─ book-detail.html // 作品详情：作品信息、章节列表、阅读入口  ├─ category.html // 分类页面：按类型筛选电子书  ├─ error.html // 错误页面：系统异常时展示  ├─ forgetPW.html // 找回密码：用户密码重置  ├─ index.html // 平台首页：推荐作品、分类导航、搜索框  ├─ log.html // 登录页面：用户账号登录  └─ read.html // 阅读页面：章节内容展示、翻页控制 |

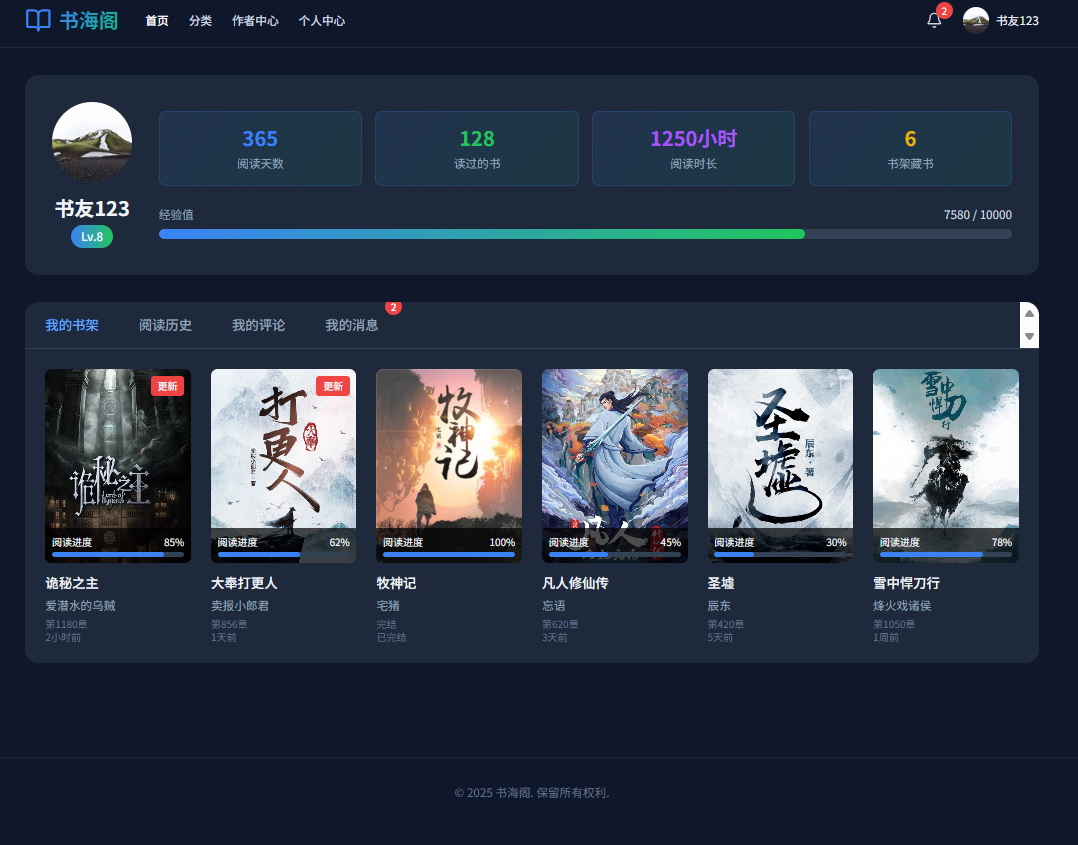
## 四、核心功能说明

### 1. 用户模块（读者 / 作者通用）

* 注册登录：支持邮箱注册，密码加密存储；登录后根据角色（读者 / 作者）展示对应功能菜单；

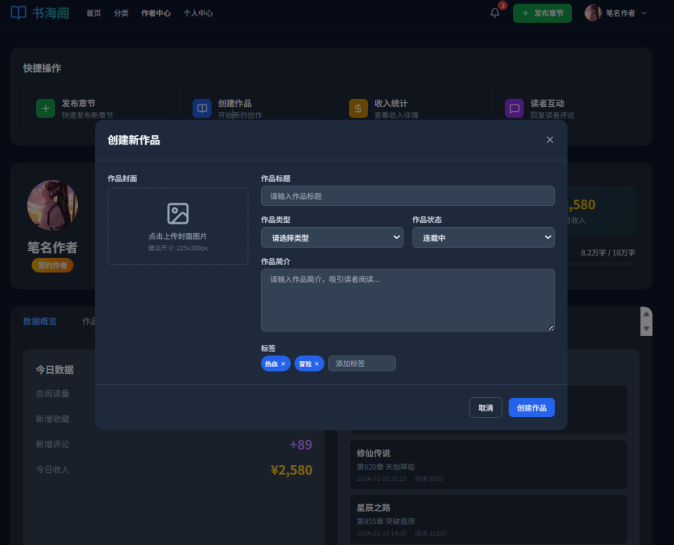
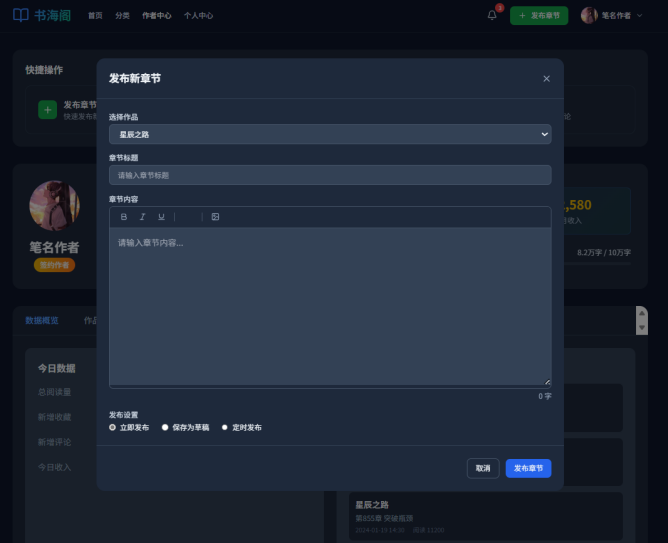
 

* 个人中心：查看 / 编辑个人信息（昵称、头像）；作者可从个人中心跳转至 “作者中心”。

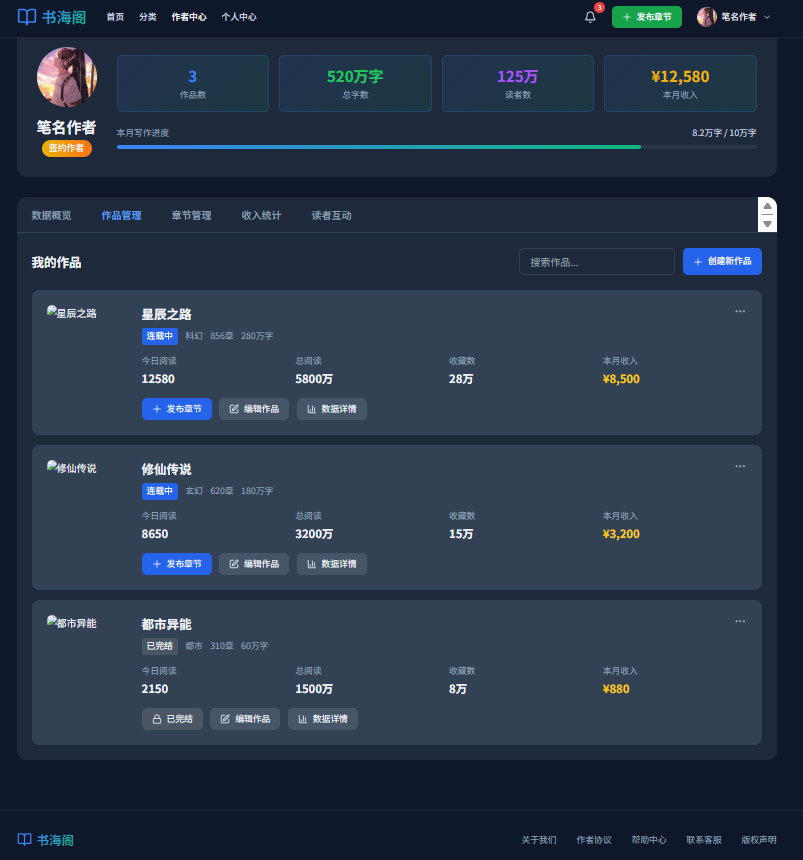


### 作品模块（作者核心操作）

* 作品创建：填写作品标题、简介、类型，上传封面图，生成初始作品；

* 作品管理：编辑作品信息、删除已创建作品（仅作者本人可见）；

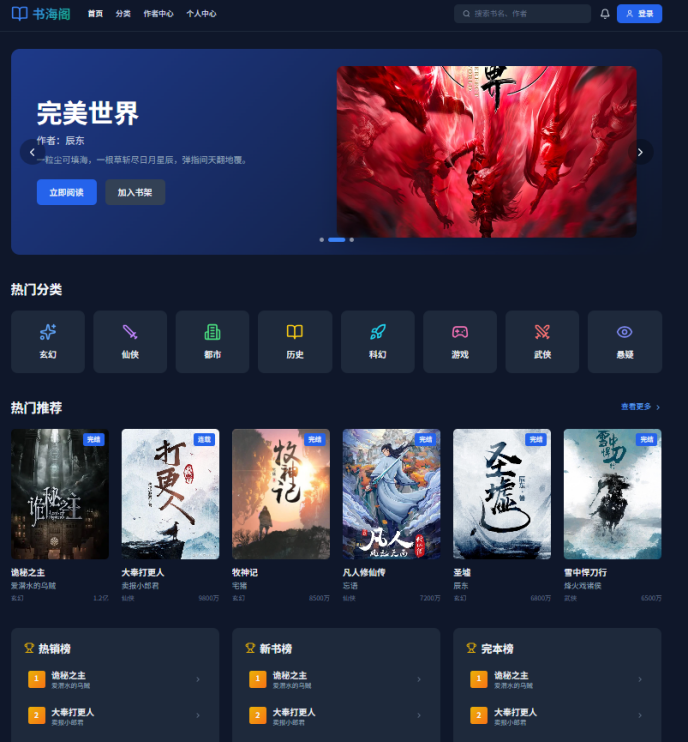
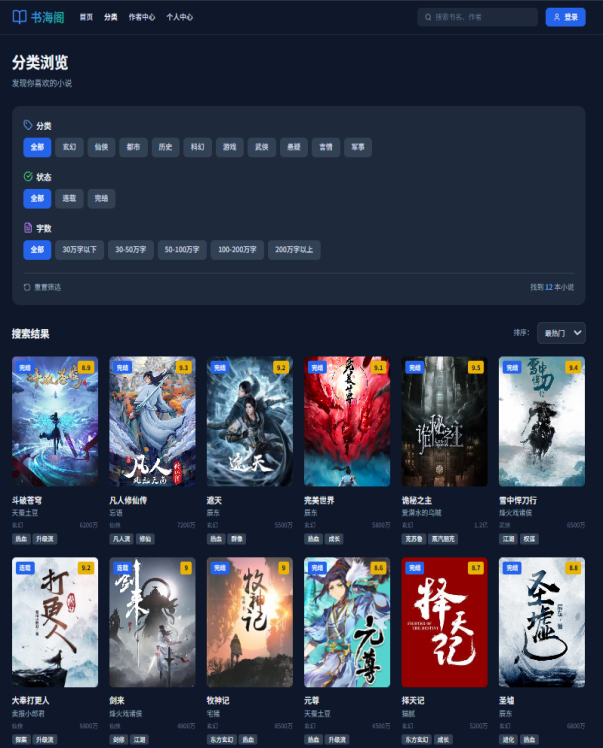


* 章节管理：为作品添加新章节（支持 “立即发布”“保存草稿”“定时发布”）、编辑章节内容、删除章节。



### 3. 阅读模块（读者核心操作）

* 作品发现：首页推荐、分类筛选、关键词搜索；

* 在线阅读：进入作品详情页后，选择章节进入阅读界面，支持上下翻页、字体大小调整；





### 4. 数据统计模块（作者核心操作）

* 实时数据：在作者中心查看 “今日阅读量”“新增收藏”“今日收入”；



* 趋势分析：通过图表展示 “阅读量趋势”、“作品数据分布”）、；



* 收入统计：展示 “本月收入”“累计收入”“待结算金额”，支持申请提现（需对接支付接口，可作为扩展功能）



## 五、部署与运行步骤

### 1. 前置准备

1. 安装 JDK 11：配置环境变量（JAVA\_HOME），确保命令行输入java -version能显示版本；
2. 安装 MySQL 8.0+：创建空数据库，设置数据库编码为 UTF-8；
3. 安装 Maven 3.6+：配置环境变量，确保命令行输入mvn -v能显示版本。

### 2. 项目配置

打开项目配置文件resources/application.properties，修改数据库连接信息：

|  |
| --- |
| spring.application.name=zjsu  spring.thymeleaf.prefix=classpath:/templates/  spring.thymeleaf.suffix=.html  spring.thymeleaf.mode=HTML  spring.thymeleaf.encoding=UTF-8  spring.thymeleaf.cache=false  spring.datasource.url=jdbc:mysql://localhost:3306/inovel\_per?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Shanghai&useSSL=false  spring.datasource.username=你的账号  spring.datasource.password=你的密码  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  spring.jpa.hibernate.ddl-auto=update  spring.jpa.show-sql=true  spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect  spring.datasource.hikari.maximum-pool-size=10  spring.datasource.hikari.minimum-idle=5  spring.datasource.hikari.idle-timeout=300000  spring.datasource.hikari.connection-timeout=20000 |

### 3. 项目构建与运行

1. 进入项目根目录（含pom.xml的文件夹），打开命令行，执行打包命令：

|  |
| --- |
| mvn clean install |

打包完成后，在target文件夹下会生成ebook-platform-0.0.1-SNAPSHOT.jar（文件名以实际项目为准）。

1. 运行项目：在命令行执行以下命令（替换 jar 包名为实际文件名）：

|  |
| --- |
| java -jar target/ebook-platform-0.0.1-SNAPSHOT.jar |

1. 访问平台：打开浏览器，输入http://localhost:8080，即可进入平台首页。若需修改端口，可在配置文件中添加server.port=8081（将 8081 改为目标端口）。

## 六、注意事项

1. 数据库初始化：项目首次启动时，JPA 会自动根据实体类创建数据表，无需手动建表；
2. 静态资源：若需替换默认图片（如作品封面、头像），直接替换static/images下的文件即可；
3. 功能扩展：若需添加 “支付接口”“消息通知” 等功能，可基于现有架构扩展 Controller 和 Repository；
4. 问题排查：若启动失败，优先检查数据库连接（用户名、密码、端口）、JDK 版本是否符合要求。