**项目名称​：小说阅读网站 (iNovel Website)**

**​技术栈​：前端React + 后端Node.js + MySQL数据库**

**第一版创造者：zzh**

**注意：在client文件里的node\_modules依赖文件已删除，自己安装环境配置解决**

**技术架构说明​**

**后端技术栈​**

| **技术项** | **说明** | **关键位置** |
| --- | --- | --- |
| **​运行时​** | **Node.js** | **server/ 目录** |
| **​框架​** | **Express.js** |  |
| **​数据库​** | **MySQL** | **server/config/** |
| **​认证​** | **JWT** | **AuthContext.js** |
|  |  |  |

**前端技术栈​**

| **技术项** | **版本/说明** | **关键文件** |
| --- | --- | --- |
| **​框架​** | **React 18+** | **App.jsx, App.tsx** |
| **​语言​** | **JavaScript + TypeScript** | **.js/.jsx/.ts/.tsx** |
| **​构建工具​** | **Create React App** | **package.json** |
| **​样式方案​** | **CSS Modules + 传统CSS** | **各组件对应的.css文件** |
| **​路由管理​** | **React Router** | **各页面组件** |
| **​状态管理​** | **Context API** | **AuthContext.js** |

**数据库：MySQL**

**所用数据库已本地导出**

**名为inovel\_backup\_2025-10-03**

**用户密码3147**

**放在文件夹了，安装MySQL使用**

**开发环境要求​**

**Node.js >= 14.0.0**

**npm >= 6.0.0**

**MySQL >= 5.7**

**VS Code**

**MySQL**

**Git**

**项目本地运行（在cmd输入以下指令）：**

**启动MySQL服务：net start mysql**

**登录进入mysql命令行 ："G:\mysql\bin\mysql.exe" -u root –p**

**切换数据库： USE inovel;**

**启动后端**

**cd F:\code\web\inovel-website\server**

**node server.js**

**启动前端**

**cd F:\code\web\inovel-website\client**

**npm start**

**然后便会弹出网页**

**项目文件功能详解**

**client/ 前端项目主目录**

**client/**

**node\_modules/ # npm依赖包（自动生成，无需手动修改）**

**public/ # 静态资源目录**

**src/ # 源代码目录**

**.gitignore # Git忽略文件配置**

**package.json # 项目配置和依赖管理**

**package-lock.json # 依赖版本锁定文件**

**README.md # 项目说明文档**

**tsconfig.json # TypeScript配置文件**

**public/ 静态资源目录**

| **文件** | **类型** | **功能说明** |
| --- | --- | --- |
| **images/default-cover.jpg** | **图片资源** | **默认小说封面图** |
| **favicon.ico** | **图标** | **浏览器标签页图标** |
| **index.html** | **HTML模板** | **应用入口HTML文件** |
| **logo192.png logo512.png** | **PWA图标** | **不同尺寸的应用图标** |
| **manifest.json** | **PWA配置** | **定义应用名称、主题色等** |
| **robots.txt** | **SEO配置** | **搜索引擎爬虫规则** |

**src/ 源代码目录**

| **组件文件** | **功能描述** | **业务作用** |
| --- | --- | --- |
| **CollapsibleCategory.js/css** | **折叠式分类导航** | **小说分类筛选和展示** |
| **Layout.js/css** | **页面布局组件** | **提供统一的页面框架** |
| **NovelCard.js/css** | **小说卡片组件** | **展示小说基本信息** |
| **Pagination.js/css** | **分页组件** | **列表数据分页显示** |
| **SearchBar.js/css** | **搜索栏组件** | **提供搜索输入和触发** |

**context/ - 全局状态管理​**

| **文件** | **功能描述** |
| --- | --- |
| **AuthContext.js** | **用户认证状态管理，提供登录/注册/登出功能** |

**核心功能​：**

**维护用户登录状态**

**提供认证相关方法**

**跨组件共享用户信息**

**pages/ - 页面组件​**

| **页面文件** | **对应路由** | **功能描述** |
| --- | --- | --- |
| **Home.js** | **/** | **首页，展示小说列表** |
| **CategoryPage.js/css** | **/category/:id** | **分类小说列表页** |
| **Login.js** | **/login** | **用户登录页面** |
| **Register.js** | **/register** | **用户注册页面** |
| **NovelDetail.js/css** | **/novel/:id** | **小说详情页面** |
| **Reader.js/css** | **/reader/:id** | **小说阅读器页面** |
| **Profile.js** | **/profile** | **用户个人资料页** |
| **SearchResults.js/css** | **/search** | **搜索结果展示页** |

**styles/ - 样式文件​**

| **样式文件** | **作用范围** |
| --- | --- |
| **AuthForm.css** | **登录/注册表单样式** |
| **Profile.css** | **用户资料页样式** |
| **Reader.css** | **阅读器页面样式** |
| **App.css** | **全局应用样式** |

**根目录核心文件​**

| **文件** | **类型** | **功能描述** |
| --- | --- | --- |
| **App.js App.tsx** | **主组件** | **应用根组件，路由配置** |
| **App.test.tsx** | **测试文件** | **单元测试用例** |
| **index.js index.tsx** | **入口文件** | **React应用启动入口** |
| **index.css** | **全局样式** | **基础样式和变量定义** |
| **logo.svg** | **图标资源** | **React logo图标** |
| **react-app-env.d.ts** | **TypeScript声明** | **环境类型定义** |
| **reportWebVitals.ts** | **性能监控** | **应用性能指标收集** |
| **setupTests.ts** | **测试配置** | **测试环境初始化** |

**server/ - 后端项目根目录**

**db.js - 数据库连接配置​**

***功能：MySQL数据库连接池配置和管理***

***核心职责：***

***数据库连接参数配置***

***连接池优化设置***

***连接错误处理和重试机制***

***数据库字符集和时区配置***

**controllers/ - 业务逻辑控制器​**

**​novelController.js - 小说业务控制器​**

**功能：处理小说相关的所有业务逻辑**

**包含方法：**

**getNovels() // 获取小说列表（支持分页、筛选）**

**getNovelById() // 获取特定小说详情**

**createNovel() // 创建新小说**

**updateNovel() // 更新小说信息**

**deleteNovel() // 删除小说**

**searchNovels() // 搜索小说**

**middlewares/ - 中间件层​**

**​auth.js - 认证中间件​**

***功能：用户身份验证和权限控制***

***包含中间件：***

***authenticateToken() // JWT令牌验证***

***authorize() // 权限验证（管理员/普通用户）***

***requireAuth() // 强制登录验证***

**errorHandler.js - 全局错误处理中间件​**

***功能：统一错误处理和响应格式化***

***处理类型：***

***数据库错误（连接失败、查询错误）***

***验证错误（参数校验失败）***

***业务逻辑错误（数据不存在、权限不足）***

***未知错误（服务器内部错误）***

**models/ - 数据模型层​**

**​novelModel.js - 小说数据模型​**

***功能：封装所有小说相关的数据库操作***

***包含方法：***

***findAll() // 查询所有小说（支持条件筛选）***

***findById() // 根据ID查询小说***

***create() // 创建新小说记录***

***update() // 更新小说信息***

***delete() // 删除小说***

***findByCategory() // 按分类查询***

***search() // 全文搜索***

**routes/ - 路由定义层​**

**​categoryRoutes.js - 分类路由​**

***路由定义：***

***GET /api/categories // 获取所有分类***

***GET /api/categories/:id // 获取特定分类详情***

***POST /api/categories // 创建新分类（需要管理员权限）***

***PUT /api/categories/:id // 更新分类信息***

***DELETE /api/categories/:id // 删除分类***

**novelRoutes.js - 小说路由​**

**路由定义：**

**GET /api/novels // 小说列表（分页、筛选）**

**GET /api/novels/:id // 小说详情**

**GET /api/novels/:id/chapters // 小说章节列表**

**POST /api/novels // 创建小说（需要权限）**

**PUT /api/novels/:id // 更新小说**

**DELETE /api/novels/:id // 删除小说**

**searchRoutes.js - 搜索路由​**

***路由定义：***

***GET /api/search?q=关键词 // 全文搜索***

***GET /api/search/suggest // 搜索建议（自动完成）***

***GET /api/search/advanced // 高级搜索（多条件）***

**userRoutes.js - 用户路由​**

***路由定义：***

***POST /api/users/register // 用户注册***

***POST /api/users/login // 用户登录***

***GET /api/users/profile // 获取用户资料***

***PUT /api/users/profile // 更新用户资料***

***POST /api/users/avatar // 上传用户头像***

***GET /api/users/favorites // 获取用户收藏***

**根目录核心文件​**

**​.env - 环境变量配置**

**package.json - 项目配置和依赖管理**

**server.js - 应用入口文件​**

***功能：Express应用初始化和配置***

***核心职责：***

***应用实例创建和中间件注册***

***路由挂载和错误处理***

***服务器启动和端口监听***

***全局异常处理***

**配置层​：环境配置和数据库连接**

**模型层​：纯数据操作，不包含业务逻辑**

**控制器层​：业务逻辑处理**

**路由层​：URL映射和请求分发**

**中间件层​：横切关注点处理**

**数据流架构​**

**HTTP请求 → Express应用 → 中间件层 → 路由层 → 控制器层 → 模型层 → 数据库**

**HTTP响应 ← Express应用 ← 中间件层 ← 路由层 ← 控制器层 ← 模型层 ← 数据库**

**前后端协作流程**

**用户注册流程**

**前端页面 → 表单提交 → POST /api/users/register → 后端验证 → 数据库存储**

**← 响应结果 ← 返回JWT令牌 ← 密码加密**

**小说浏览流程**

**前端页面 → 组件挂载 → GET /api/novels → 后端查询 → MySQL数据库**

**← 渲染列表 ← 返回JSON数据 ← 数据处理**

**搜索功能流程**

**搜索框输入 → GET /api/search?q=关键词 → 数据库全文搜索 → 返回结果**

**← 实时显示结果 ← JSON格式数据**

**核心功能模块**

**​1. 用户认证系统​**

**组件文件​：Login.js, Register.js, AuthContext.js**

**样式文件​：AuthForm.css**

**功能描述​：用户注册、登录、状态管理、Token验证**

**​2. 小说浏览模块​**

**组件文件​：Home.js, NovelCard.js, CategoryPage.js**

**功能描述​：小说列表展示、分类筛选、分页浏览**

**​3. 阅读体验模块​**

**组件文件​：NovelDetail.js, Reader.js**

**样式文件​：Reader.css**

**功能描述​：小说详情展示、章节阅读、阅读进度跟踪**

**​4. 搜索功能模块​**

**组件文件​：SearchBar.js, SearchResults.js**

**功能描述​：全文搜索、结果展示、关键词高亮**

**​5. 用户中心模块​**

**组件文件​：Profile.js**

**样式文件​：Profile.css**

**功能描述​：个人信息管理、阅读历史、收藏功能**

**​6. 布局导航模块​**

**组件文件​：Layout.js, CollapsibleCategory.js**

**功能描述​：统一页面布局、分类导航、响应式设计**

**项目开发进度**

**目前完成或者基本完成功能**

**1.小说管理功能**

**小说列表展示**

**分页加载**

**分类筛选**

**基础搜索功能**

**小说详情查看**

**完整信息展示（标题、封面、作者、字数等）**

**章节列表展示**

**章节阅读**

**章节内容加载**

**基础阅读体验**

**2. 用户系统**

**用户认证**

**登录/注册功能**

**密码加密存储**

**基础权限控制**

**路由级访问控制**

**用户上下文管理**

**3. 搜索功能**

**基础搜索**

**标题关键词搜索**

**简单结果排序**

**全文检索能力**

**分类筛选支持**

**4. 分类系统**

**分类导航**

**分类列表展示**

**分类下小说列表**

**分类管理**

**未完成或未能实现功能（未来拓展方向）**

1. **小说评分功能**
2. **小说评论区**
3. **小说排行榜**
4. **根据用户喜好进行小说推荐**
5. **免费与vip用户功能**
6. **用户个人的收藏和阅读历史功能**
7. **作者专区功能**
8. **付费阅读和作者入驻系统**
9. **广告植入**