**项目目录结构**

根据 **前后端分离** 的最佳实践，整个 **exam-system** 项目将采用以下目录结构：

📂 ~/Projects/exam-system

│── 📂 frontend # 前端（React/Next.js）

│ ├── 📂 public # 静态资源（图片、favicon 等）

│ ├── 📂 src # 源代码

│ │ ├── 📂 components # 可复用的 UI 组件

│ │ ├── 📂 pages # 页面文件

│ │ ├── 📂 styles # 样式文件（Tailwind CSS）

│ │ ├── 📂 utils # 工具函数

│ │ ├── 📂 hooks # 自定义 React Hooks

│ │ ├── 📂 context # 状态管理（Context API）

│ │ ├── 📂 services # API 请求封装

│ │ └── 📄 App.js # 入口文件

│ ├── 📄 package.json # 前端依赖配置

│ ├── 📄 tailwind.config.js # Tailwind CSS 配置

│ ├── 📄 next.config.js # Next.js 配置

│ ├── 📄 .eslintrc.js # ESLint 配置

│ ├── 📄 .prettierrc # Prettier 代码格式化配置

│ ├── 📄 README.md # 前端 README 说明

│ └── 📄 .gitignore # Git 忽略文件

│

│── 📂 backend # 后端（Node.js + Express）

│ ├── 📂 src # 源代码

│ │ ├── 📂 controllers # 处理 HTTP 请求

│ │ ├── 📂 routes # 路由定义

│ │ ├── 📂 models # 数据库模型

│ │ ├── 📂 middleware # 中间件（认证、日志）

│ │ ├── 📂 config # 配置文件（数据库、环境变量）

│ │ ├── 📂 utils # 工具类（JWT 生成、加密）

│ │ └── 📄 server.js # 服务器入口文件

│ ├── 📄 package.json # 后端依赖配置

│ ├── 📄 .env # 环境变量（数据库连接、密钥）

│ ├── 📄 .eslintrc.js # ESLint 配置

│ ├── 📄 README.md # 后端 README 说明

│ ├── 📂 tests # 后端测试（Jest）

│ ├── 📄 Dockerfile # Docker 配置（可选）

│ ├── 📄 .gitignore # Git 忽略文件

│ └── 📂 scripts # 数据库迁移、种子数据

│

│── 📂 database # 数据库相关文件（MySQL）

│ ├── 📄 schema.sql # 数据库结构定义

│ ├── 📄 seed.sql # 初始数据填充

│ ├── 📄 backup.sql # 备份数据

│

│── 📂 docs # 文档（需求分析、API 文档）

│ ├── 📄 system\_design.md # 系统设计

│ ├── 📄 api\_spec.md # API 规范

│ ├── 📄 db\_schema.md # 数据库设计

│ ├── 📄 ui\_wireframes.md # UI 线框图

│ └── 📄 troubleshooting.md # 常见问题 & 解决方案

│

│── 📂 scripts # 项目管理脚本

│ ├── 📄 setup.sh # 一键安装依赖、初始化环境

│ ├── 📄 start.sh # 启动前后端

│ ├── 📄 deploy.sh # 部署脚本（可选）

│

│── 📄 .gitignore # Git 忽略文件

│── 📄 README.md # 总体 README 说明

│── 📄 package.json # 项目依赖（Monorepo 可选）

│── 📄 docker-compose.yml # Docker 容器编排（可选）

│── 📄 LICENSE # 开源许可证

│── 📄 CONTRIBUTING.md # 贡献指南（如果是开源项目）

**📌 说明**

1. **frontend/** 📂 → **React/Next.js** 前端代码

• **使用 Tailwind CSS** 来实现 UI

• **服务端请求** 通过 services/ 目录封装 API

2. **backend/** 📂 → **Node.js + Express** 后端

• 采用 **MVC 架构**：controllers/、routes/、models/

• middleware/ 目录用于 **身份认证、日志记录**

• config/ 目录用于 **数据库连接、环境变量**

3. **database/** 📂 → **MySQL 数据库文件**

• schema.sql **创建表**

• seed.sql **填充初始数据**

• backup.sql **备份数据库**

4. **docs/** 📂 → **所有文档**

• system\_design.md 📄 **系统需求分析**

• api\_spec.md 📄 **API 说明**

• db\_schema.md 📄 **数据库设计**

• ui\_wireframes.md 📄 **UI 线框图**

• troubleshooting.md 📄 **常见错误及解决方案**

5. **scripts/** 📂 → **项目管理自动化脚本**

• setup.sh 📄 **一键初始化**

• start.sh 📄 **启动前后端**

• deploy.sh 📄 **部署脚本（可选）**

### 📌 学生学习模块新增内容

\*\*学习模块文件结构\*\*  
  
 📂 `frontend/src/pages/learn/`  
 - 📄 `index.js` ➝ 选择 \*\*学习模式（Section / Chapter / Key Points）\*\*  
 - 📄 `section.js` ➝ 目录级学习  
 - 📄 `chapter.js` ➝ 章节级学习  
 - 📄 `keypoint.js` ➝ 知识点学习  
 - 📄 `progress.js` ➝ 学习进度  
  
 📂 `backend/src/routes/`  
 - 📄 `learn.js` ➝ 处理学习数据 API  
 - 📄 `keypoints.js` ➝ 处理知识点匹配 API