

**课 程 实 验 报 告**

**专 业 年 级 信工2023(专升本)**

**课 程 名 称 医学电子商务**

**学 生 姓 名 王周龙**

**学 号 20231201652**

**实 验 日 期 2025年10月21日**

**实 验 地 点 明志楼210**

**实 验 成 绩**

**2025 年 10月**

|  |  |
| --- | --- |
| **实验项目**  **名称** | Django百度百科风格简易项目 |
| **实验目的及要求** | （1）确定项目需要实现的核心Django组件，例如模型定义，视图函数和URL路由；  （2）设计数据模型结构：规划百科词条的核心字段，如标题、内容、创建时间和分类等；  （3）制定视图功能清单：列出学生必须实现的关键页面，如词条列表页、详情页和创建/编辑页；  （4）设定前端展示要求：规定基本的模板继承结构和用户交互界面，确保界面简洁清晰。技术方案：vue+react  （5）作业提交内容：明确需要提交的代码文件、运行说明和学习总结报告。 |
| **实验**  **内容** | 1.Idea  1.1方案、规划、框架(对应文档)。。  1.2确立产品原型，制作产品脑。  2.需求分析  1.1 确定主要用户群体和app实现的功能。  1.2设计可以参考pixso来看确定结合更多的第三方功能(推送，地图，社交分享,第三方登录等)。  确认需求创建一个百科项目：使用Django创建一个类似百度百科一样的百科项目。需要规划百科词条的核心字段，如标题、内容、创建时间和分类等。需要有词条列表页、详细页和创建/编辑等。  1.3绘制简单的界面图  3.1设计html前缀的效果图，包括每一个分页，页面的效果图。 |
| **实验步骤** | 规划整个项目的结构和任务。绘制ER图    创建项目的基础结构和配置。安装相关依赖，创建Vue前端应用。首先创建Vue项目的配置文件。创建React前端应用，然后创建React应用的基础结构。开始测试和使用所有功能。  发现部分功能页面尚未开发，创建前端词条分类功能和个人中心页面，修复无法创建词条和删除词条功能。    发现搜索功能正在开发中，开发搜索功能，发现词条显示有些问题，词条列表也存在作者和分类信息显示为JSON格式的问题。  发现只运用了vue没有使用react，让我们了解实现一个简单的混合方案需要了解的问题重新配置构建工具（Vite/Webpack），处理两个框架的依赖隔离，设计组件间的通信机制。让我们使用react创建一个数据可视化图表。安装React相关依赖和构建工具  创建React图表组件（词条统计图表）  配置Vite支持React组件构建  创建Vue包装器组件，集成React图表  在个人中心页面添加数据可视化图表  测试React图表在Vue项目中的运行效果    不断修复部分功能，最终达到可用状态，完成项目。    了解搜索功能如何实现  1．搜索触发机制  - 搜索框位置 ：位于 `App.vue` 的导航栏中  - 触发方式 ：用户在搜索框输入关键词后点击"搜索"按钮或按回车键  - 路由跳转 ：搜索时通过 router.push(/search?q=${encodeURIComponent(value.trim())}) 跳转到搜索结果页面  2. 搜索组件结构  - 主搜索页面 ： `SearchResults.vue`  - 状态管理 ：使用Pinia store进行状态管理  - API调用 ：通过axios实例进行HTTP请求          ### 4. 后端API接口  - 请求URL ： GET /api/entries/  - 搜索参数 ： search 查询参数  - 后端实现 ：Django后端通过 Q(title\_\_icontains=search) | Q(content\_\_icontains=search) | Q(summary\_\_icontains=search) 进行模糊搜索 |
| **实验环境** | Trae  Django框架  Vue+React前端 |
| **实验结果** | React前端优点 ：  - 使用Ant Design UI组件库，界面美观  - 使用Context API进行状态管理  - 结构相对简洁  Vue前端优点 ：  - 使用Pinia进行状态管理（比Context API更强大）  - 使用Element Plus UI组件库  - 功能更完整（有分类管理、收藏功能等）  - 路由结构更完善  共同问题 ：  - 两个前端功能重复，维护成本高  - 技术栈分散，学习成本高  ## 实现的功能特性  - 混合框架集成 ：Vue + React的混合开发模式  - 响应式图表 ：图表支持响应式布局，适配不同屏幕尺寸  - 美观的UI设计 ：使用Ant Design Vue的卡片布局，图表配色协调  - 模拟数据展示 ：目前使用模拟数据，可轻松替换为真实API数据  ## 🔧 技术栈详情  - 前端框架 ：Vue 3 + React 19  - 图表库 ：Recharts  - 构建工具 ：Vite  - UI组件库 ：Ant Design Vue  现在您可以在个人中心页面看到两个数据可视化图表：  1. 词条统计趋势图 - 展示词条数量、浏览量和点赞数的月度变化趋势  2. 词条分类分布图 - 展示词条在不同分类中的分布比例 |
| **教师评语** |  |

**注：可根据实际情况加页**