湖南科技大学计算机科学与工程学院

综合实践实训（1） 课程设计报告

**专业班级：** 23大数据一班

**姓 名：** 袁权

**学 号：** 2305060114

**指导教师：** 姜磊

**时 间**： 2025.5.26-2025.6.6

**地 点**： 逸夫楼433

|  |
| --- |
| **指导教师评语：**  **成绩： 等级：**  **签名：**  **年 月 日** |

目录

[一、 实验题目 1](#_Toc17235)

[二、 实验目的 2](#_Toc237)

[三、总体设计 2](#_Toc27981)

[1. 知识背景 2](#_Toc23505)

[前端 2](#_Toc19403)

[后端 2](#_Toc1527)

[2. 模块介绍 3](#_Toc17174)

[前端 3](#_Toc27394)

[1. 用户角色 3](#_Toc11991)

[2.主要模块 3](#_Toc25659)

[后端 5](#_Toc26194)

[1.三层架构 5](#_Toc19061)

[2.主要模块 5](#_Toc11077)

[3.后端系统特点 6](#_Toc15327)

[3. 整体设计步骤 6](#_Toc8416)

[前端 7](#_Toc23971)

[后端 7](#_Toc4986)

[安全 8](#_Toc24388)

[四、详细设计 8](#_Toc24299)

[1. 项目结构 8](#_Toc30226)

[前端: 8](#_Toc28853)

[后端: 10](#_Toc31032)

[2. 前后端接口设计: 14](#_Toc3162)

[登录: 14](#_Toc19239)

[普通员工: 14](#_Toc27544)

[部门管理员 17](#_Toc25453)

[薪资管理员 20](#_Toc24935)

[人事管理员 22](#_Toc9276)

[系统管理员 24](#_Toc7652)

[3. 数据库设计 27](#_Toc16681)

[1. 用户表(sys\_user) 27](#_Toc25229)

[2. 角色表(sys\_role) 28](#_Toc26866)

[3. 用户角色关联表(sys\_user\_role) 28](#_Toc12676)

[4. 员工信息表(hr\_employee) 28](#_Toc12864)

[5. 部门表(hr\_department) 29](#_Toc6056)

[6. 职位表(hr\_position) 30](#_Toc3016)

[7. 薪酬表(hr\_salary) 30](#_Toc28877)

[8. 人事调整记录表(hr\_transfer) 31](#_Toc15960)

[五、实验结果与分析 32](#_Toc749)

[视图 32](#_Toc11268)

[登录页 32](#_Toc30699)

[系统管理员 32](#_Toc31654)

[人事管理员 33](#_Toc6176)

[薪资管理员 34](#_Toc26858)

[部门管理员 35](#_Toc20714)

[普通员工 36](#_Toc29189)

[六、 小结与心得体会 37](#_Toc2770)

1. 实验题目

**人力资源管理系统**

功能模块:

1. 用户认证

登录模块 ：用户登录验证

2. 员工管理

个人信息 ：查看和修改个人信息

薪资查询 ：员工查询自己的薪资信息

3. 部门管理

部门列表 ：查看所有部门信息

添加部门 ：创建新部门

编辑部门 ：修改部门信息

4. 财务管理

薪资管理 ：查询和管理员工薪资

编辑薪资 ：修改员工薪资信息

5. 人力资源管理

调动记录 ：查看已处理的员工调动记录

待审批调动 ：处理待审批的员工调动申请

6. 系统管理

用户管理 ：查看系统所有用户

添加用户 ：创建新用户账号

1. 实验目的

WEB应用技术课程设计是计算机软件工程专业的一个综合性实践教学环节，是学习完《高级WEB技术》课程后进行的一次全面的综合练习。其目的在于促进学生复习和巩固计算机软件设计知识，加深对软件设计方法、软件设计技术和设计思想的理解，并能运用所学软件设计知识和WEB工程技术进行综合软件设计，通过本课程设计让学生进行简单WEB应用软件系统的开发，掌握软件设计的方法和面向对象程序设计的基本技术，提高学生在软件开发方面的综合应用能力。

三、总体设计

1. **知识背景**

前端

Vite: 快速构建vue框架

Vue 3：使用Composition API (setup script)开发组件

Vue Router：处理页面路由

Pinia：状态管理,前端数据存储

Axios：处理HTTP请求

后端

基于Java Spring Boot的人事管理系统。

Spring: IOC容器负责管理对象生命周期

Spring MVC: 处理Web请求

Spring Boot: 快速搭建SSM项目框架

MyBatis: 持久层框架

MySQL: 数据存储

1. **模块介绍**

**前端**

1. 用户角色

系统支持多种用户角色，每种角色拥有不同的权限和功能：

系统管理员(ROLE\_ADMIN)：负责用户管理，添加和查看所有用户

人事经理(ROLE\_HR)：负责处理人事调动申请和查看历史记录

财务专员(ROLE\_FINANCE)：负责薪资管理

部门主管(ROLE\_DEPT\_HEAD)：负责部门管理

普通员工(ROLE\_EMPLOYEE)：可以查看个人信息和薪资

2.主要模块

①登录模块Login.vue

提供用户登录功能，包含用户名和密码验证

登录成功后根据用户角色跳转到相应页面

②员工模块

主页面：/employee/Employee.vue

个人信息：/employee/PersonalInfo.vue

显示员工详细信息，包括基本信息、部门、职位等

支持修改个人信息

薪资查询：/employee/SalaryQuery.vue

允许员工查询特定年月的薪资信息

③ 部门管理模块

主页面：/department/Department.vue

显示所有部门信息列表

提供添加、编辑、删除部门功能

添加部门：/department/AddDepartment.vue

包含部门名称、编号、上级部门ID、部门领导ID等信息

编辑部门：/department/EditDepartment.vue

修改现有部门信息

④财务管理模块

主页面：/finance/Finance.vue

提供薪资查询功能，可按员工ID和年月查询

编辑薪资：/finance/EditSalary.vue

修改员工薪资信息

⑤人事管理模块

主页面：/hr/Hr.vue

调动记录：/hr/TransferList.vue

显示已处理的人事调动记录

待审批调动：/hr/TransferNotApply.vue

显示待审批的人事调动申请

提供审批功能

⑥系统管理员模块

主页面：/admin/Admin.vue

用户列表：/admin/UserList.vue

显示所有用户信息

提供删除用户功能

添加用户：/admin/AddUser.vue

创建新用户，包括基本信息、部门、角色等

**后端**

1.三层架构

①表现层：Controller层，处理HTTP请求和响应

②业务层：Service层，实现业务逻辑

③持久层：Mapper层，负责数据库操作

2.主要模块

①用户登录模块

相关类：UserController,UserService,SysUser,SysUserMapper, UserMapper

功能：用户登录、认证和权限管理

②员工管理模块

相关类: HrEmployee ,EmployeeController,

EmployeeService, HrEmployeeMapper

功能：员工信息的增删改查，包括个人基本信息管理

③部门管理模块

相关类：HrDepartment, DepartmentController,

DepartmentService,HrDepartmentMapper

功能：部门信息的增删改查

④ 薪资管理模块

相关类：HrSalary,,FinanceController,FinanceService,HrSalaryMapper

功能：员工薪资信息的查询和管理

⑤人事调动模块

相关类：HrTransfer , HrController , HrService,HrTransferMapper

功能：员工调动申请、审批流程管理

⑥主要实体类

SysUser：系统用户，用于登录认证

HrEmployee：员工信息，包含详细的个人资料

HrDepartment：部门信息

HrSalary：薪资信息

HrTransfer：调动信息

SysUserRole: 用户权限信息

3.后端系统特点

分层清晰：采用MVC架构，代码结构清晰

接口规范：API接口设计规范，使用RESTful风格

数据封装：使用DTO模式封装数据传输对象，提高安全性

统一响应：使用`Result`类统一封装响应结果，提高前后端交互的一致性

1. **整体设计步骤**

**前端**

①用户通过登录页面验证身份

②登录成功后，用户信息存储在Pinia状态管理中（userStore）

③路由守卫确保未登录用户无法访问受保护页面,以及视图跳转

④ 各功能模块通过axios调用后端API获取或提交数据

⑤ 部分模块（如部门管理、财务管理）使用专门的Store管理状态

**后端**

1. 项目结构搭建

创建Spring Boot项目

配置MyBatis、数据库连接等

搭建基础包结构

2. 实体类编写

通过Mybatis-Generator插件自动将表映射成对应的实体类

添加额外的注解和属性

3. 控制层实现

编写Controller类，处理HTTP请求

调用Service接口，返回统一格式的响应

4. 业务层实现

编写Service接口和实现类

实现业务逻辑，调用Mapper接口

5. 持久层实现

编写Mapper接口，定义数据库操作方法

编写对应的XML映射文件

**安全**

全局路由守卫确保用户必须登录才能访问系统功能

用户密码采用正则表达式验证，确保安全性

敏感操作（如删除）提供二次确认

这个人事管理系统提供了企业内部人事管理所需的基本功能，包括员工信息管理、部门管理、薪资管理和人事调动等核心业务流程。

四、详细设计

1. **项目结构**

**前端:**

├── Personnel-management.zip

└── Personnel-management\

├── .gitignore

├── .vscode\

│ └── extensions.json

├── README.md

├── index.html

├── package-lock.json

├── package.json # 项目依赖配置

├── public\

│ └── vite.svg

├── src\ # 源代码目录

│ ├── App.vue # 根组件

│ ├── assets\ # 静态资源

│ │ └── vue.svg

│ ├── components\ # 组件目录

│ │ ├── Login.vue # 登录组件

│ │ ├── admin\ # 管理员相关组件

│ │ │ ├── AddUser.vue

│ │ │ ├── Admin.vue

│ │ │ └── UserList.vue

│ │ ├── department\ # 部门管理相关组件

│ │ │ ├── AddDepartment.vue

│ │ │ ├── Department.vue

│ │ │ └── EditDepartment.vue

│ │ ├── employee\ # 员工相关组件

│ │ │ ├── Employee.vue

│ │ │ ├── PersonalInfo.vue

│ │ │ └── SalaryQuery.vue

│ │ ├── finance\ # 财务相关组件

│ │ │ ├── EditSalary.vue

│ │ │ └── Finance.vue

│ │ └── hr\ # 人力资源相关组件

│ │ ├── Hr.vue

│ │ ├── TransferList.vue

│ │ └── TransferNotApply.vue

│ ├── main.js # 入口文件

│ ├── pinia.js # Pinia配置

│ ├── routers\ # 路由配置

│ │ └── router.js

│ ├── store\ # 状态管理

│ │ ├── adminStore.js

│ │ ├── departmentStore.js

│ │ ├── financyStore.js

│ │ ├── hrStore.js

│ │ └── userStore.js

│ └── utils\ # 工具函数

│ └── request.js # Axios请求封装

└── vite.config.js # Vite配置文件

**后端:**

**项目整体结构:**

src/

├── main/ # 主要源代码

│ ├── java/ # Java源代码

│ │ └── com/hnust/personnelmanagement/ # 主包

│ └── resources/ # 资源文件

│ ├── application.yml # 应用配置文件

│ └── mapper/ # MyBatis映射文件

└── test/ # 测试源代码

**工作目录:**

com.hnust.personnelmanagement/

├── PersonnelManagementApplication.java # 应用程序入口

├── ServletInitializer.java # Servlet初始化器

├── bean/ # 数据模型

│ ├── dto/ # 数据传输对象

│ ├── pojo/ # 普通Java对象

│ ├── repository/ # 数据库实体

│ └── vo/ # 视图对象

├── config/ # 配置类

│ └── CorsConfig.java # 跨域配置

├── controller/ # 控制器

│ ├── AdminController.java # 管理员控制器

│ ├── DepartmentController.java # 部门控制器

│ ├── EmployeeController.java # 员工控制器

│ ├── FinanceController.java # 财务控制器

│ ├── HrController.java # 人力资源控制器

│ └── UserController.java # 用户控制器

├── mapper/ # MyBatis映射接口

│ └── UserMapper/ # 用户相关映射

│ ├── HrDepartmentMapper.java # 部门映射

│ ├── HrEmployeeMapper.java # 员工映射

│ ├── HrSalaryMapper.java # 薪资映射

│ ├── HrTransferMapper.java # 调动映射

│ ├── SysUserMapper.java # 系统用户映射

│ ├── SysUserRoleMapper.java # 用户角色映射

│ └── UserMapper.java # 用户映射

├── service/ # 服务层

│ ├── AdminService.java # 管理员服务接口

│ ├── DepartmentService.java # 部门服务接口

│ ├── EmployeeService.java # 员工服务接口

│ ├── FinanceService.java # 财务服务接口

│ ├── HrService.java # 人力资源服务接口

│ ├── UserService.java # 用户服务接口

│ └── impl/ # 服务实现

│ ├── AdminServiceImpl.java # 管理员服务实现

│ ├── DepartmentServiceImpl.java # 部门服务实现

│ ├── EmployeeServiceImpl.java # 员工服务实现

│ ├── FinanceServiceImpl.java # 财务服务实现

│ ├── HrServiceImpl.java # 人力资源服务实现

│ └── UserServiceImpl.java # 用户服务实现

└── util/ # 工具类

**MyBatis映射目录:**

mapper/UserMapper/

├── HrDepartmentMapper.xml # 部门映射XML

├── HrEmployeeMapper.xml # 员工映射XML

├── HrSalaryMapper.xml # 薪资映射XML

├── HrTransferMapper.xml # 调动映射XML

├── SysUserMapper.xml # 系统用户映射XML

├── SysUserRoleMapper.xml # 用户角色映射XML

└── UserMapper.xml # 用户映射XML

SpringBoot的yml配置

# application.yml 主要配置

mybatis:

type-aliases-package: com.hnust.personnelmanagement.bean # MyBatis别名包

mapper-locations: classpath:mapper/\*\*/\*.xml # 映射文件位置

configuration:

map-underscore-to-camel-case: true # 开启驼峰命名转换

spring:

datasource: # 数据库配置

url: jdbc:mysql://localhost:3306/personnel\_management?allowMultiQueries=true

username: root

password: \*\*\*\*\*\*

driver-class-name: com.mysql.cj.jdbc.Driver

logging:

level:

com.hnust.personnelmanagement.mapper: debug # Mapper日志级别

1. **前后端接口设计:**

登录:

URL: /pmsys/login

请求方式:GET

请求参数:

{

username: "zhangsan",

password: "zhangsan123"

}

响应结果:

{

"code": 200, "message": "登录成功",

"data": {

"id": "5",

"username": "zhangsan",

"password": null,

"realName": "张三",

"roleCode": "ROLE\_EMPLOYEE"

}

}

普通员工:

个人信息

URL: /pmsys/employee/info/{realName}

方法:GET

请求参数:

{

"realName": "张三"

}

响应结果:

{

"code": 200,

"message": "success",

"data": {

"id": 1,

"employeeNo": "EMP001",

"name": "张三",

"gender": 1, // 1表示男性，0表示女性

"birthDate": "1990-01-01",

"idCard": "110101199001011234",

"phone": "13800138000",

"email": "zhangsan@example.com",

"address": "北京市朝阳区xxx街道",

"education": "本科",

"positionId": 1,

"positionName": "软件工程师",

"departmentId": 1,

"departmentName": "研发部",

"entryDate": "2020-01-01",

"status": 1,

"createTime": "2020-01-01 09:00:00",

"updateTime": "2022-01-01 10:00:00"

}

}

薪资查询

URL: /pmsys/employee/salaryInfo

方法:GET

请求参数:

{

id: userStore.id,

salaryYear: salaryYear.value,

salaryMonth: salaryMonth.value

}

响应结果:

{

"code": 200,

"message": "获取用户工资信息成功",

"data": {

"id": 1,

"employeeId": 5,

"basicSalary": 30000.00,

"positionSalary": 10000.00,

"performanceSalary": 5000.00,

"bonus": 20000.00,

"insurance": 5000.00,

"tax": 4500.00,

"actualSalary": 55500.00,

"salaryMonth": "2025-05",

"status": true,

"createTime": "2025-05-27T02:04:37.000+00:00",

"updateTime": "2025-05-27T02:04:37.000+00:00"

}

}

部门管理员

获取部门列表

URL: /pmsys/department/list

方法:GET

请求参数: 无

响应结果:

{

"id": 1,

"deptName": "技术研发部",

"deptCode": "TECH\_001",

"parentId": 0, -->parentName

"leaderId": 1001, -->"leaderName"

"description": "负责产品研发和技术创新",

"createTime": "2025-05-20T08:30:00",

"updateTime": "2025-05-25T14:15:00",

}

添加部门

URL: /pmsys/departments/add

方法: POST

请求参数:

{

"deptName": "测试部",

"deptCode": "TEST",

"parentId": 0,

"leaderId": 1,

"description": "负责公司测试工作"

}

响应结果:

{

"code": 200,

"message": "添加成功",

"data": {

"id": 3,

"deptName": "测试部",

"deptCode": "TEST"

}

}

获取部门信息

URL: /pmsys/departments/{id}

方法: GET

请求参数:

{

"id" : 1

}

响应结果:

{

"code": 200,

"message": "查询部门成功",

"data": {

"id": 1,

"deptName": "总公司",

"deptCode": "DEPT\_ROOT",

"parentId": 0,

"parentName": null,

"leaderId": 1,

"leaderName": "张三",

"description": "公司总部",

"createTime": "2025-05-27T01:24:55.000+00:00",

"updateTime": "2025-05-27T01:24:55.000+00:00"

}

修改部门

URL: /pmsys/departments/{id}

方法: PUT

请求参数:

{

"deptName": "测试部",

"deptCode": "TEST",

"parentId": 0,

"leaderId": 1,

"description": "负责公司测试工作"

}

响应结果:

{"code": 200,"message": "修改成功","data": null}

薪资管理员

获取薪酬列表

URL: /pmsys/salaries

方法: GET

请求参数:

{

page: 1

size: 10

employeeId: 可选，员工ID

salaryMonth: 可选，工资月份(格式:yyyy-MM)

status: 可选，状态(0:未发放,1:已发放)

}

响应结果:

{

"code": 200,

"message": "获取员工工资信息成功",

"data": [

{

"id": 1,

"employeeId": 5,

"basicSalary": 30000.00,

"positionSalary": 10000.00,

"performanceSalary": 5000.00,

"bonus": 20000.00,

"insurance": 5000.00,

"tax": 4500.00,

"actualSalary": 55500.00,

"salaryMonth": "2025-05",

"status": true,

"createTime": "2025-05-27T02:04:37.000+00:00",

"updateTime": "2025-05-27T02:04:37.000+00:00"

},

{

"id": 2,

"employeeId": 6,

"basicSalary": 20000.00,

"positionSalary": 8000.00,

"performanceSalary": 4000.00,

"bonus": 5000.00,

"insurance": 4000.00,

"tax": 2700.00,

"actualSalary": 30300.00,

"salaryMonth": "2025-05",

"status": true,

"createTime": "2025-05-27T02:04:37.000+00:00",

"updateTime": "2025-05-27T02:04:37.000+00:00"

},

{

"id": 3,

"employeeId": 7,

"basicSalary": 15000.00,

"positionSalary": 5000.00,

"performanceSalary": 3000.00,

"bonus": 3000.00,

"insurance": 3000.00,

"tax": 1800.00,

"actualSalary": 22200.00,

"salaryMonth": "2025-05",

"status": true,

"createTime": "2025-05-27T02:04:37.000+00:00",

"updateTime": "2025-05-27T02:04:37.000+00:00"

}

]

}

修改薪酬

URL: /pmsys/salary

方法: PUT

请求参数:

{

"id": 3,

"employeeId": 7,

"basicSalary": 15000.00,

"positionSalary": 5000.00,

"performanceSalary": 3000.00,

"bonus": 3000.00,

"insurance": 3000.00,

"tax": 1800.00,

"actualSalary": 22200.00,

}

响应结果:

{"code": 200, "data": null}

人事管理员

获取已处理调动记录列表

URL: /pmsys/transfers/completedTransferList

方法: GET

请求参数:无

响应结果:

{ "code": 200,"message": "操作成功",

"data": {

"total": 100,

"list": [

{

"id": 1,

"employeeId": 1,

"employeeName": "张三",

"beforeDepartmentName": "技术部",

"afterDepartmentName": "产品部",

"beforePositionName": "Java工程师",

"afterPositionName": "产品经理",

"transferType": 3,

"transferReason": "岗位调整",

"transferDate": "2023-01-01",

"status": 1,

"createTime": "2022-12-25 00:00:00"

}

]

}

}

查询未审批的调动

URL: /pmsys/transfer/notApply

方法: GET

请求参数:无

响应结果:

{ "code": 200,"message": "操作成功",

"data": {

"total": 100,

"list": [

{

"id": 1,

"employeeId": 1,

"employeeName": "张三",

"beforeDepartmentName": "技术部",

"afterDepartmentName": "产品部",

"beforePositionName": "Java工程师",

"afterPositionName": "产品经理",

"transferType": 3,

"transferReason": "岗位调整",

"transferDate": "2023-01-01",

"status": 0,

"createTime": "2022-12-25 00:00:00"

}

]

}

}

审批调动

URL: /pmsys/transfers/{id}/approve

方法: POST

请求参数:无

响应结果:

{   "code": 200,  "message": "审批成功",  "data": null}

系统管理员

添加用户

URL: /pmsys/admin/add

方法: POST

请求参数:

sys\_user , hr\_employee , sys\_user\_role

{

"userInfo": {

"username": "zhangsan",

"password": "zhangsan123",

"email": "zhangsan@company.com",

"phone": "13800138000",

"realName": "张三"

},

"employeeInfo": {

"employeeNo": "EMP003",

"gender": 1,

"birthDate": "1995-08-12",

"idCard": "110101199508123456",

"address": "北京市朝阳区",

"education": "硕士研究生",

"positionId": 3,

"departmentId": 3,

},

"roleId": 5 第二个默认为5 普通员工

}

响应结果:

{   "code": 200,   "message": "添加成功",  "data": null}

查询所有用户

URL: /pmsys/admin/userList

方法: GET

请求参数:无

响应结果:

{"code": 200,"message": "success",

"data": {

"userList": [

{

"id": 1,

"username": "admin",

"password": null,

"realName": "系统管理员",

"email": "admin@company.com",

"phone": "13800138000",

"status": 1,

"createTime": "2025-05-27T01:12:51.000+00:00",

"updateTime": "2025-05-27T01:12:51.000+00:00"

},

{

"id": 2,

"username": "hrmanager",

"password": null,

"realName": "人事经理",

"email": "hr@company.com",

"phone": "13800138001",

"status": 1,

"createTime": "2025-05-27T01:12:51.000+00:00",

"updateTime": "2025-05-27T01:12:51.000+00:00"

}

...

]

}

}

删除用户

URL: /pmsys/admin/deleteUser/{id}

方法: DELETE

请求参数:无

响应结果:

{"code": 200, "message": "删除成功", "data": null}

1. **数据库设计**

1. 用户表(sys\_user)

CREATE TABLE `sys\_user` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '用户ID',

`username` varchar(50) NOT NULL COMMENT '用户名',

`password` varchar(100) NOT NULL COMMENT '密码',

`real\_name` varchar(50) DEFAULT NULL COMMENT '真实姓名',

`email` varchar(100) DEFAULT NULL COMMENT '邮箱',

`phone` varchar(20) DEFAULT NULL COMMENT '手机号',

`status` tinyint(1) DEFAULT '1' COMMENT '状态(0:禁用,1:正常)',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `username` (`username`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='系统用户表';

2. 角色表(sys\_role)

CREATE TABLE `sys\_role` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '角色ID',

`role\_name` varchar(50) NOT NULL COMMENT '角色名称',

`role\_code` varchar(50) NOT NULL COMMENT '角色编码',

`description` varchar(255) DEFAULT NULL COMMENT '角色描述',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `role\_code` (`role\_code`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='角色表';

3. 用户角色关联表(sys\_user\_role)

CREATE TABLE `sys\_user\_role` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT 'ID',

`user\_id` bigint(20) NOT NULL COMMENT '用户ID',

`role\_id` bigint(20) NOT NULL COMMENT '角色ID',

PRIMARY KEY (`id`),

KEY `idx\_user\_id` (`user\_id`),

KEY `idx\_role\_eid` (`role\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='用户角色关联表';

4. 员工信息表(hr\_employee)

CREATE TABLE `hr\_employee` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '员工ID',

`employee\_no` varchar(20) NOT NULL COMMENT '员工编号',

`name` varchar(50) NOT NULL COMMENT '姓名',

`gender` tinyint(1) DEFAULT NULL COMMENT '性别(0:女,1:男)',

`birth\_date` date DEFAULT NULL COMMENT '出生日期',

`id\_card` varchar(20) DEFAULT NULL COMMENT '身份证号',

`phone` varchar(20) DEFAULT NULL COMMENT '手机号',

`email` varchar(100) DEFAULT NULL COMMENT '邮箱',

`address` varchar(255) DEFAULT NULL COMMENT '住址',

`education` varchar(50) DEFAULT NULL COMMENT '学历',

`position\_id` bigint(20) DEFAULT NULL COMMENT '职位ID',

`department\_id` bigint(20) DEFAULT NULL COMMENT '部门ID',

`entry\_date` date DEFAULT NULL COMMENT '入职日期',

`status` tinyint(1) DEFAULT '1' COMMENT '状态(0:离职,1:在职)',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `employee\_no` (`employee\_no`),

KEY `idx\_position\_id` (`position\_id`),

KEY `idx\_department\_id` (`department\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='员工信息表';

5. 部门表(hr\_department)

CREATE TABLE `hr\_department` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '部门ID',

`dept\_name` varchar(50) NOT NULL COMMENT '部门名称',

`dept\_code` varchar(50) NOT NULL COMMENT '部门编码',

`parent\_id` bigint(20) DEFAULT '0' COMMENT '父部门ID',

`leader\_id` bigint(20) DEFAULT NULL COMMENT '部门负责人ID',

`description` varchar(255) DEFAULT NULL COMMENT '部门描述',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `dept\_code` (`dept\_code`),

KEY `idx\_parent\_id` (`parent\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='部门表';

6. 职位表(hr\_position)

CREATE TABLE `hr\_position` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '职位ID',

`position\_name` varchar(50) NOT NULL COMMENT '职位名称',

`position\_code` varchar(50) NOT NULL COMMENT '职位编码',

`position\_level` int(11) DEFAULT NULL COMMENT '职位等级',

`department\_id` bigint(20) DEFAULT NULL COMMENT '所属部门ID',

`description` varchar(255) DEFAULT NULL COMMENT '职位描述',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `position\_code` (`position\_code`),

KEY `idx\_department\_id` (`department\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='职位表';

7. 薪酬表(hr\_salary)

CREATE TABLE `hr\_salary` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '薪酬ID',

`employee\_id` bigint(20) NOT NULL COMMENT '员工ID',

`basic\_salary` decimal(10,2) DEFAULT NULL COMMENT '基本工资',

`position\_salary` decimal(10,2) DEFAULT NULL COMMENT '职位工资',

`performance\_salary` decimal(10,2) DEFAULT NULL COMMENT '绩效工资',

`bonus` decimal(10,2) DEFAULT NULL COMMENT '奖金',

`insurance` decimal(10,2) DEFAULT NULL COMMENT '保险',

`tax` decimal(10,2) DEFAULT NULL COMMENT '个税',

`actual\_salary` decimal(10,2) DEFAULT NULL COMMENT '实发工资',

`salary\_month` varchar(7) NOT NULL COMMENT '工资月份(格式:yyyy-MM)',

`status` tinyint(1) DEFAULT '0' COMMENT '状态(0:未发放,1:已发放)',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`),

UNIQUE KEY `idx\_employee\_month` (`employee\_id`,`salary\_month`),

KEY `idx\_salary\_month` (`salary\_month`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='薪酬表';

8. 人事调整记录表(hr\_transfer)

CREATE TABLE `hr\_transfer` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT COMMENT '调动ID',

`employee\_id` bigint(20) NOT NULL COMMENT '员工ID',

`before\_department\_id` bigint(20) DEFAULT NULL COMMENT '调动前部门ID',

`after\_department\_id` bigint(20) DEFAULT NULL COMMENT '调动后部门ID',

`before\_position\_id` bigint(20) DEFAULT NULL COMMENT '调动前职位ID',

`after\_position\_id` bigint(20) DEFAULT NULL COMMENT '调动后职位ID',

`transfer\_type` tinyint(1) DEFAULT NULL COMMENT '调动类型(1:部门调动,2:职位调动,3:同时调动)',

`transfer\_reason` varchar(255) DEFAULT NULL COMMENT '调动原因',

`transfer\_date` date DEFAULT NULL COMMENT '调动日期',

`approver\_id` bigint(20) DEFAULT NULL COMMENT '审批人ID',

`status` tinyint(1) DEFAULT '0' COMMENT '状态(0:待审批,1:已审批,2:已拒绝)',

`create\_time` datetime DEFAULT NULL COMMENT '创建时间',

`update\_time` datetime DEFAULT NULL COMMENT '更新时间',

PRIMARY KEY (`id`),

KEY `idx\_employee\_id` (`employee\_id`),

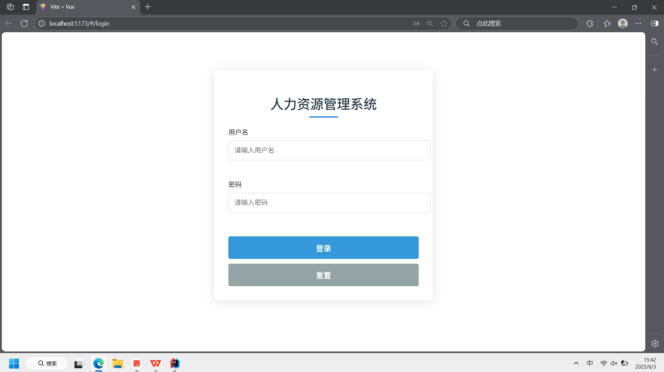
KEY `idx\_transfer\_date` (`transfer\_date`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='人事调整记录表';

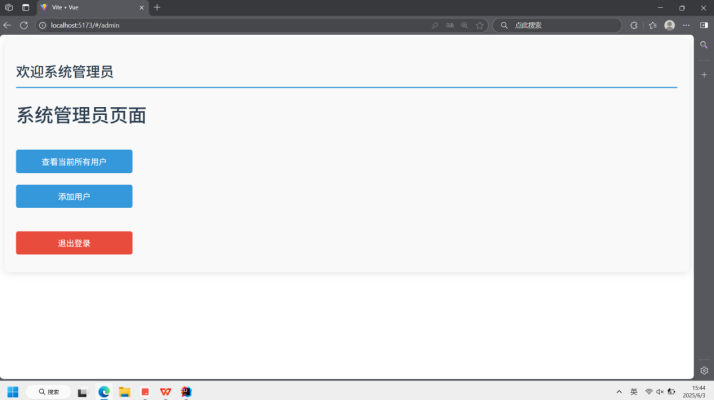
五、实验结果与分析

视图

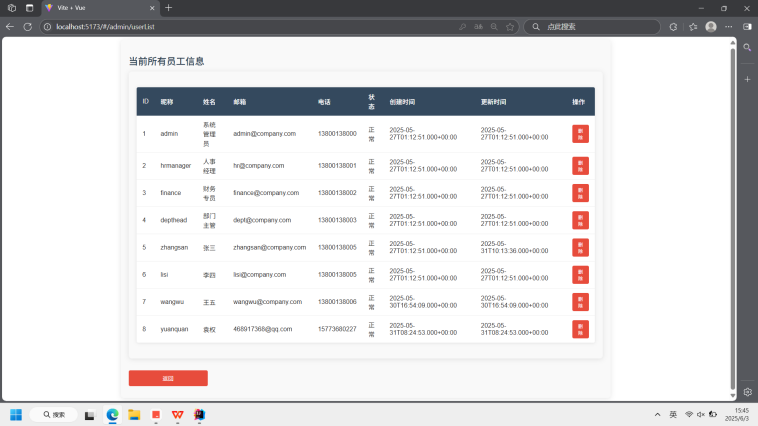
登录页



系统管理员



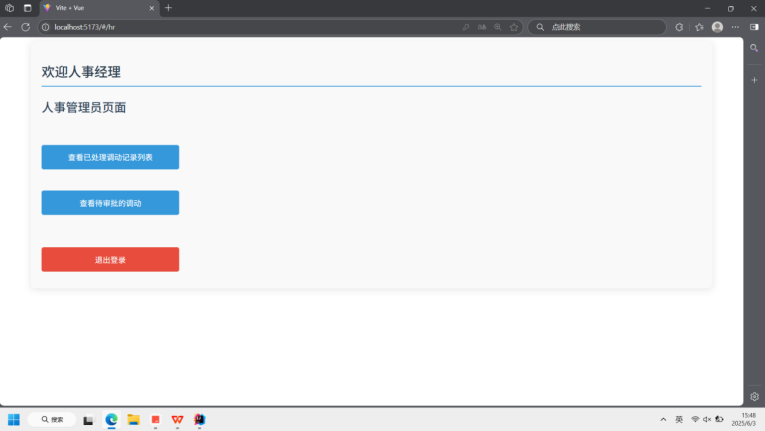
查看当前所有用户



添加用户



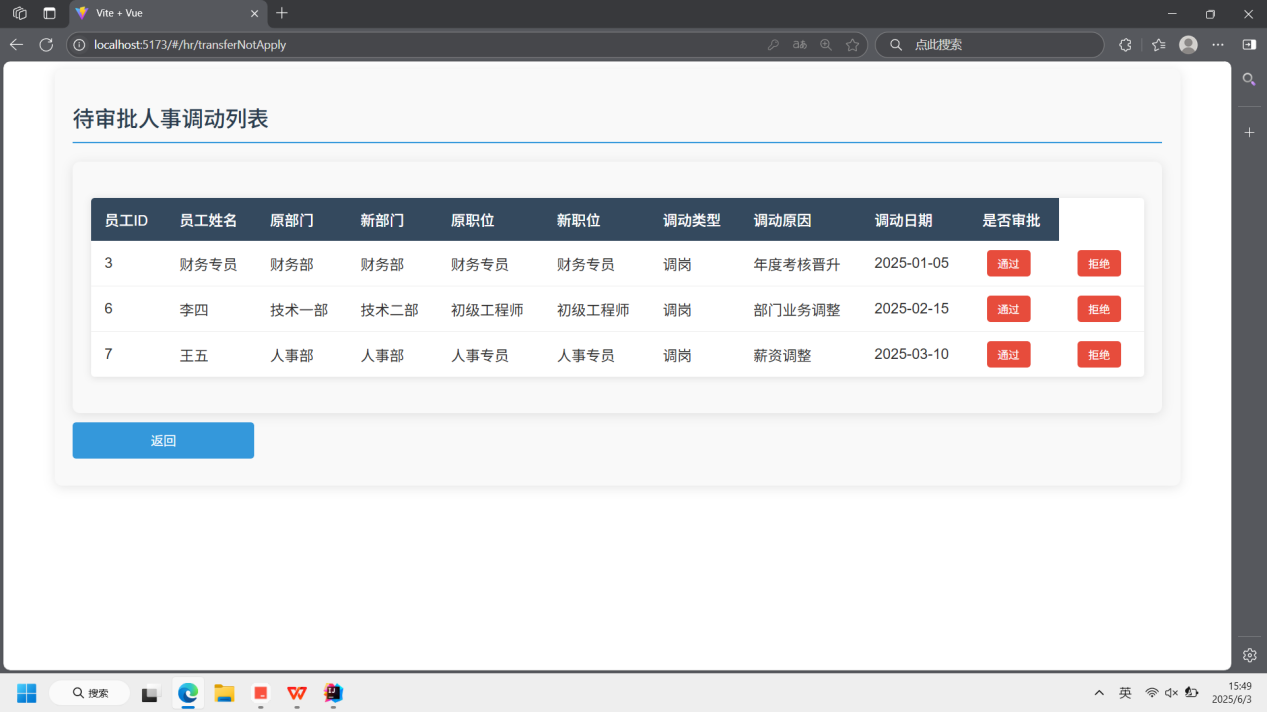
人事管理员



查看已处理调动记录列表

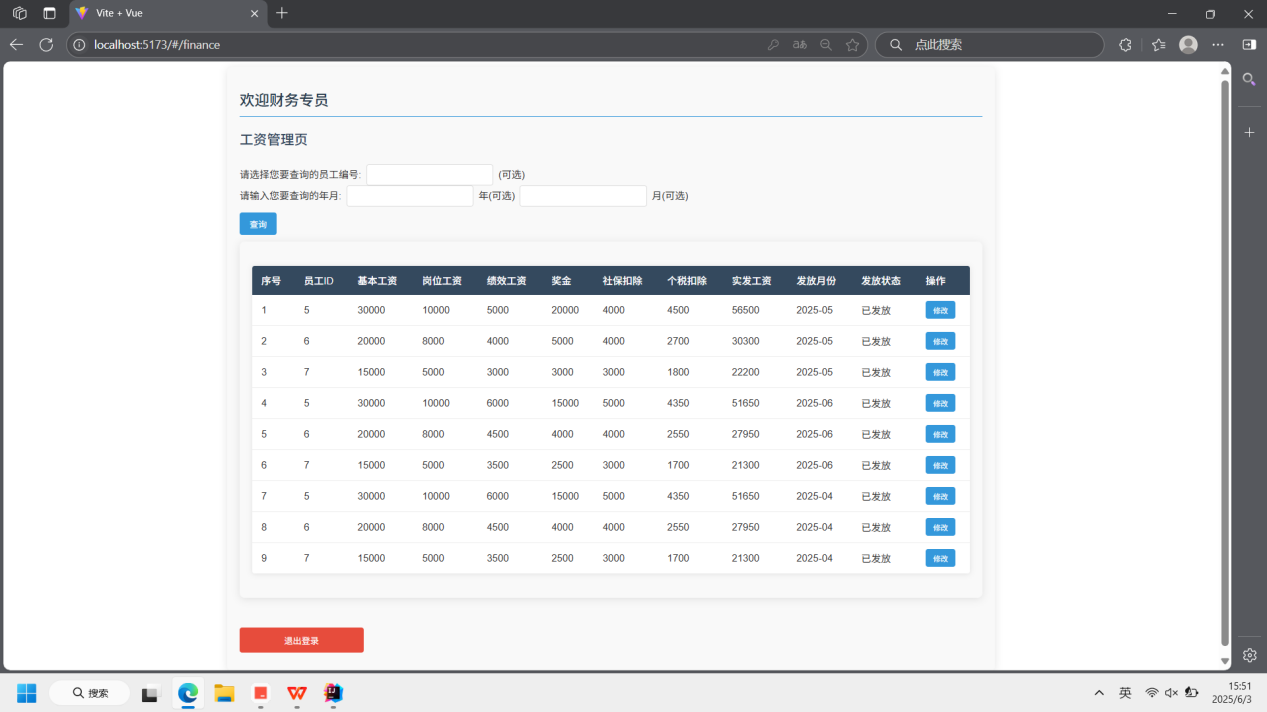


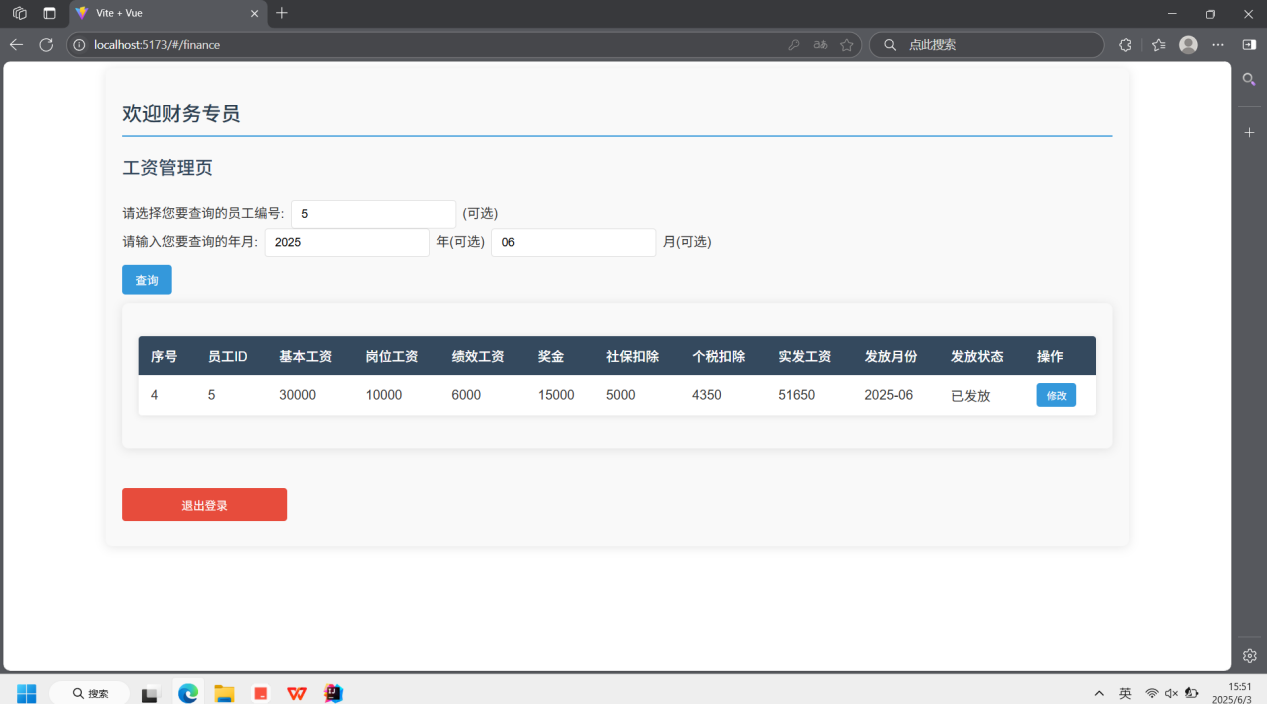
查看待审批的调动



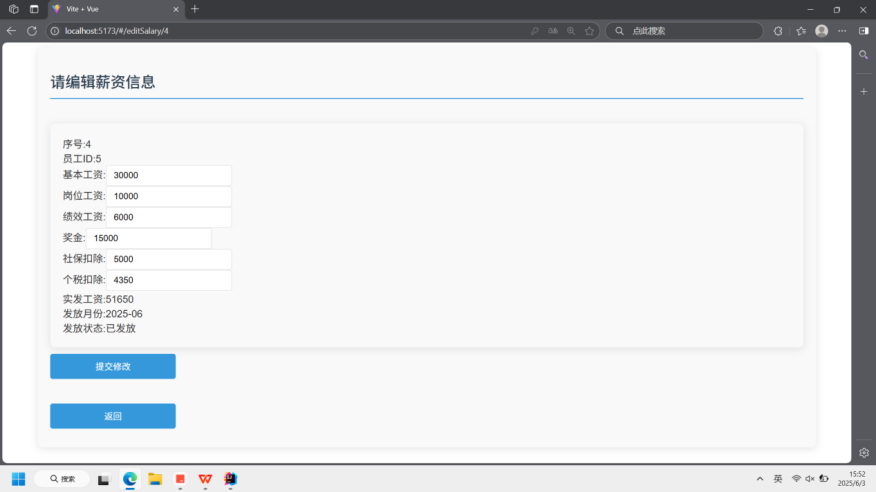
薪资管理员

薪资查询页



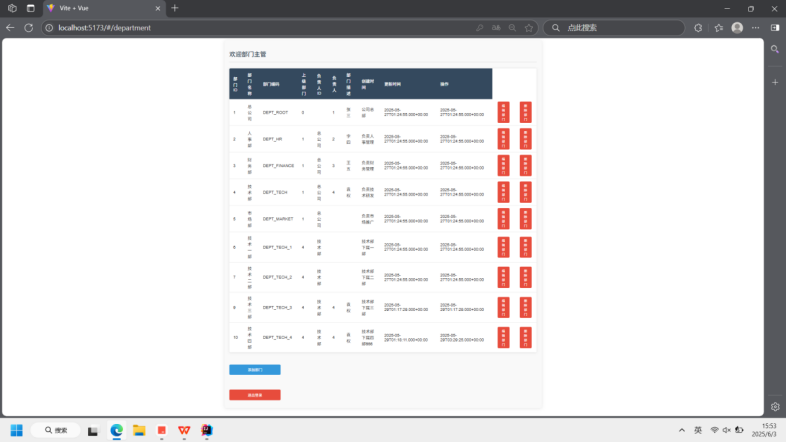


薪资修改

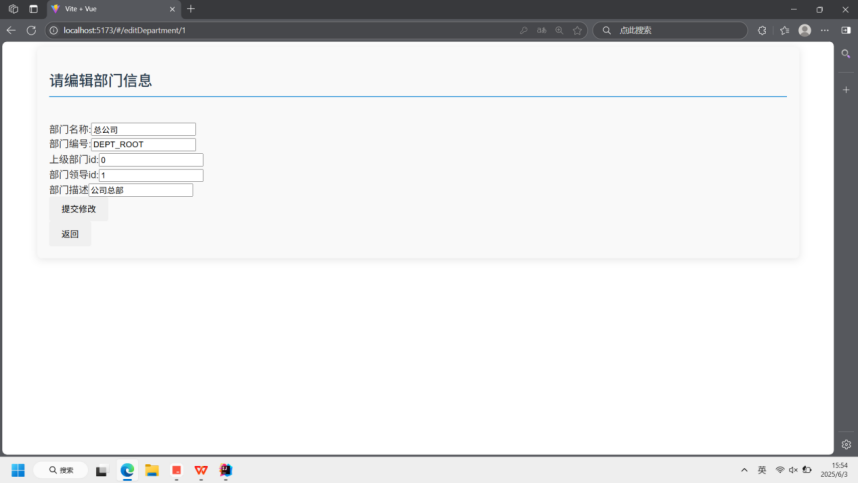


部门管理员

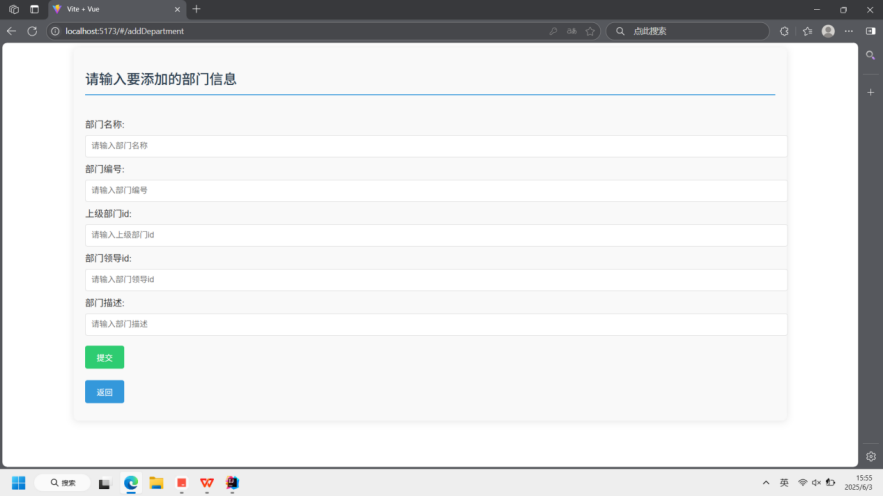
部门查询



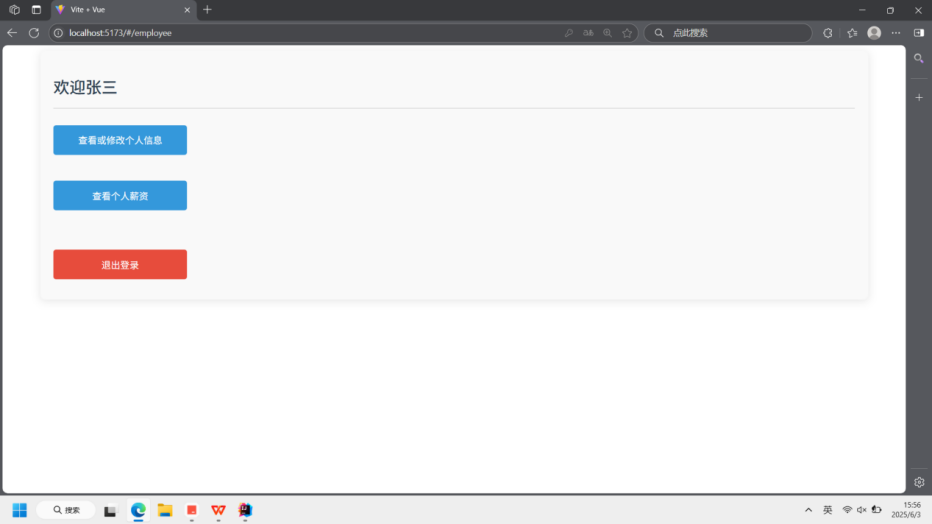
编辑部门信息



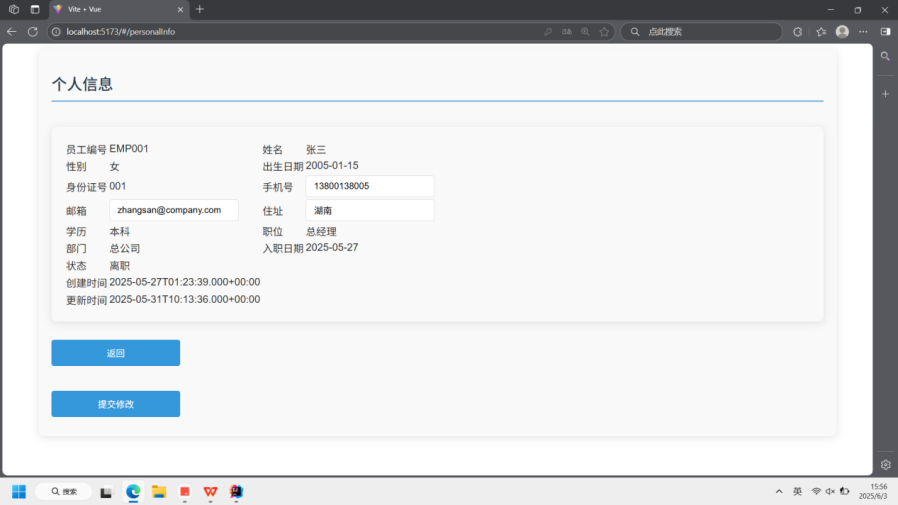
新增部门



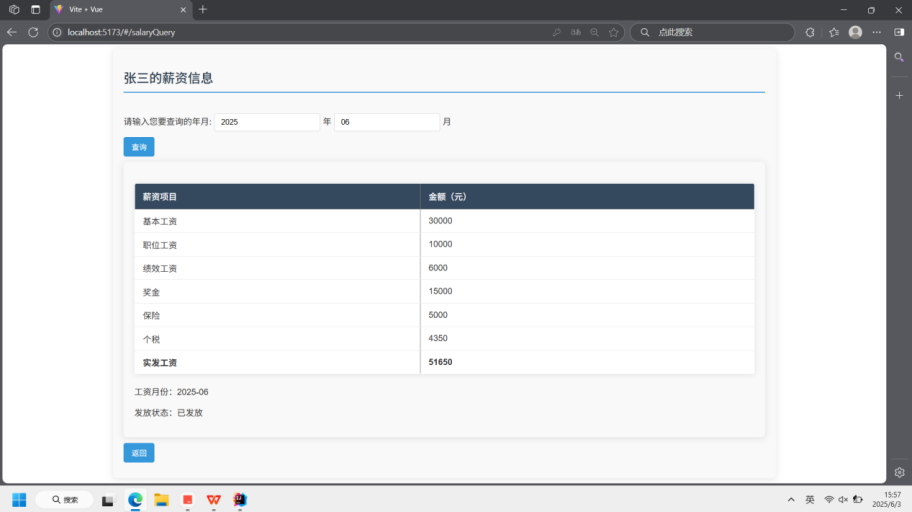
普通员工



查看或修改个人信息



查看个人薪资



1. 小结与心得体会

通过这次的课程设计,我再一次锻炼了自己的代码能力。同时遇到了许多麻烦,但是这些问题都被迎刃而解了,对于代码纠错的能力有了很大的提高。由于有了上学期数据库课程设计的web项目的经验,这一次的课设对我来说,更加得心应手。并且这一次将我之前所学习的vue前端知识进行了巩固,加深了理解,同时在这两次课设之间,我学习了Spring,SpringMVC,MyBatis,SpringBoot等后端框架,这次的项目我就使用这些框架来搭建后端。相比于之前的课设项目(之前用的三层架构),结构更加清晰,代码的耦合度更低,聚合度更高,更加适合于国内主流的业务开发。

同时这次为团队合作,提高团队协作的能力的同时,也发现了队员之间各自的优势和不足。

总结来说，本次课程设计是一次理论联系实际的深度演练，它切实提升了我的前后端开发能力和团队协作素养，为未来的学习和职业发展奠定了更坚实的基础。