# 实验二：部门新增、编辑与查询 实验报告

## 一、基本信息

* **项目**：甜点信息管理系统 — 部门管理模块
* **实验名称**：实验二：部门新增、编辑与查询
* **学号**：4123080199
* **姓名**：
* **日期**：2025-10-31

## 二、实验目的

* **掌握** 基于 Spring Boot + MyBatis 的后端增删改查接口开发与分页查询。
* **掌握** 基于 Vue3 + Element Plus 的前端表单校验、分页、对话框及与后端联调。
* **掌握** MySQL 表结构设计、初始化脚本编写与数据联调。

## 三、实验环境

* **操作系统**：Linux
* **后端技术栈**：Java 17、Spring Boot、MyBatis、PageHelper
* **数据库**：MySQL（示例连接：root/1234，数据库名 desserts）
* **前端技术栈**：Node.js、Vite、Vue3、Element Plus、Axios

## 四、数据库设计与初始化

表结构设计及样例数据见如下脚本，执行后生成 department 表并插入 3 条初始记录。

-- department\_setup.sql  
USE desserts;  
DROP TABLE IF EXISTS department;  
CREATE TABLE department (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(200) NOT NULL,  
 leader VARCHAR(100),  
 phone VARCHAR(50),  
 descp VARCHAR(500)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_unicode\_ci;  
  
INSERT INTO department(name, leader, phone, descp)  
 VALUES ('人力资源部', '王强', '010-88990001', '负责公司人力资源规划与员工发展'),  
 ('研发部', '李敏', '010-88990002', '负责产品研发与技术支持'),  
 ('财务部', '赵丽', '010-88990003', '负责公司财务预算和成本控制');

## 五、后端实现

### 5.1 通用控制器父类 BaseController

统一分页处理与响应包装。

// src/main/java/com/example/prjbackend/common/core/controller/BaseController.java  
public class BaseController {  
 protected final Logger logger = LoggerFactory.getLogger(this.getClass());  
  
 protected void startPage() {  
 PageDomain pageDomain = PageParamUtil.createPageRequest();  
 Integer pageNum = pageDomain.getPageNum();  
 Integer pageSize = pageDomain.getPageSize();  
 String orderBy = pageDomain.getOrderBy();  
 PageHelper.startPage(pageNum, pageSize, orderBy);  
 }  
  
 protected TableDataInfo getDataByPage(List<?> list) {  
 TableDataInfo tableData = new TableDataInfo();  
 tableData.setCode(200);  
 tableData.setMsg("查询成功");  
 tableData.setRows(list);  
 tableData.setTotal(new PageInfo(list).getTotal());  
 return tableData;  
 }  
  
 protected AjaxResult toAjax(int rows) {  
 return rows > 0 ? AjaxResult.success() : AjaxResult.error();  
 }  
}

### 5.2 领域对象 Department

// src/main/java/com/example/prjbackend/domain/Department.java  
@Data  
public class Department {  
 Long id;  
 String name;  
 String leader;  
 String phone;  
 String descp;  
}

### 5.3 数据访问层 DepartmentMapper

动态 SQL 支持按名称、负责人、电话模糊查询；支持批量删除。

// src/main/java/com/example/prjbackend/mapper/DepartmentMapper.java  
@Mapper  
public interface DepartmentMapper {  
 @Insert("insert into department(name, leader, phone, descp) values(#{name}, #{leader}, #{phone}, #{descp})")  
 int insert(Department department);  
  
 @Select("<script>select id, name, leader, phone, descp from department"  
 + "<where>"  
 + " <if test='name!=null and name!=""'>and name like concat('%',#{name},'%')</if>"  
 + " <if test='leader!=null and leader!=""'>and leader like concat('%',#{leader},'%')</if>"  
 + " <if test='phone!=null and phone!=""'>and phone like concat('%',#{phone},'%')</if>"  
 + "</where> order by id desc" + "</script>")  
 List<Department> list(Department department);  
  
 @Update("<script>update department <set>"  
 + " <if test='name!=null'>name=#{name},</if>"  
 + " <if test='leader!=null'>leader=#{leader},</if>"  
 + " <if test='phone!=null'>phone=#{phone},</if>"  
 + " <if test='descp!=null'>descp=#{descp},</if>"  
 + "</set> where id=#{id}</script>")  
 int edit(Department department);  
  
 @Delete("<script>delete from department where id in"  
 + " <foreach collection='array' item='id' open='(' close=')' separator=','>#{id}</foreach>"  
 + "</script>")  
 int delete(Long[] ids);  
}

### 5.4 业务层 DepartmentService

// src/main/java/com/example/prjbackend/service/DepartmentService.java  
@Service  
public class DepartmentService {  
 @Autowired  
 DepartmentMapper departmentMapper;  
  
 public int add(Department department) {  
 return departmentMapper.insert(department);  
 }  
  
 public List<Department> list(Department department) {  
 return departmentMapper.list(department);  
 }  
  
 public int edit(Department department) {  
 return departmentMapper.edit(department);  
 }  
  
 public int delete(Long[] ids) {  
 return departmentMapper.delete(ids);  
 }  
}

### 5.5 控制层 DepartmentController

提供 RESTful 接口：新增、分页查询、编辑、删除（批量）。

// src/main/java/com/example/prjbackend/controller/DepartmentController.java  
@CrossOrigin  
@RestController  
@RequestMapping("/department")  
public class DepartmentController extends BaseController {  
 @Autowired  
 DepartmentService departmentService;  
  
 @PostMapping  
 public AjaxResult add(@RequestBody Department department) {  
 return toAjax(departmentService.add(department));  
 }  
  
 @GetMapping  
 public TableDataInfo list(Department department) {  
 startPage();  
 List<Department> list = departmentService.list(department);  
 return getDataByPage(list);  
 }  
  
 @PutMapping  
 public AjaxResult edit(@RequestBody Department department) {  
 return toAjax(departmentService.edit(department));  
 }  
  
 @DeleteMapping("/{ids}")  
 public AjaxResult remove(@PathVariable Long[] ids) {  
 return toAjax(departmentService.delete(ids));  
 }  
}

## 六、前端实现

### 6.1 API 封装

// src/api/Department.js  
import axios from "../utils/request"  
  
export function add(department) {  
 return axios.post("/department", department.value)  
}  
  
export function list(queryParams) {  
 return axios.get("/department", { params: queryParams.value })  
}  
  
export function edit(department) {  
 return axios.put("/department", department.value)  
}  
  
export function del(ids2del) {  
 return axios.delete("/department/" + ids2del)  
}

### 6.2 路由与导航

// src/router/index.js  
import Department from "../views/Department.vue"  
  
const routes = [  
 { path: '/', name:'Home', component: () => import('../views/Home.vue') },   
 { path: '/Register', name:'Register', component:Register },  
  
 { path: '/Category', name:'Category', component:Category },   
 { path: '/Dessert', name:'Dessert', component:Dessert },  
  
 { path: '/Department', name:'Department', component:Department },  
  
]

<!-- src/App.vue 侧边导航新增入口 -->  
<el-aside width="200px">  
 <router-link to="/Register" class="nav-link module">用户注册</router-link>   
 <router-link to="/Category" class="nav-link module">分类管理</router-link>   
 <router-link to="/Dessert" class="nav-link module">甜点管理</router-link>  
  
 <router-link to="/Department" class="nav-link module">部门管理</router-link>  
  
</el-aside>

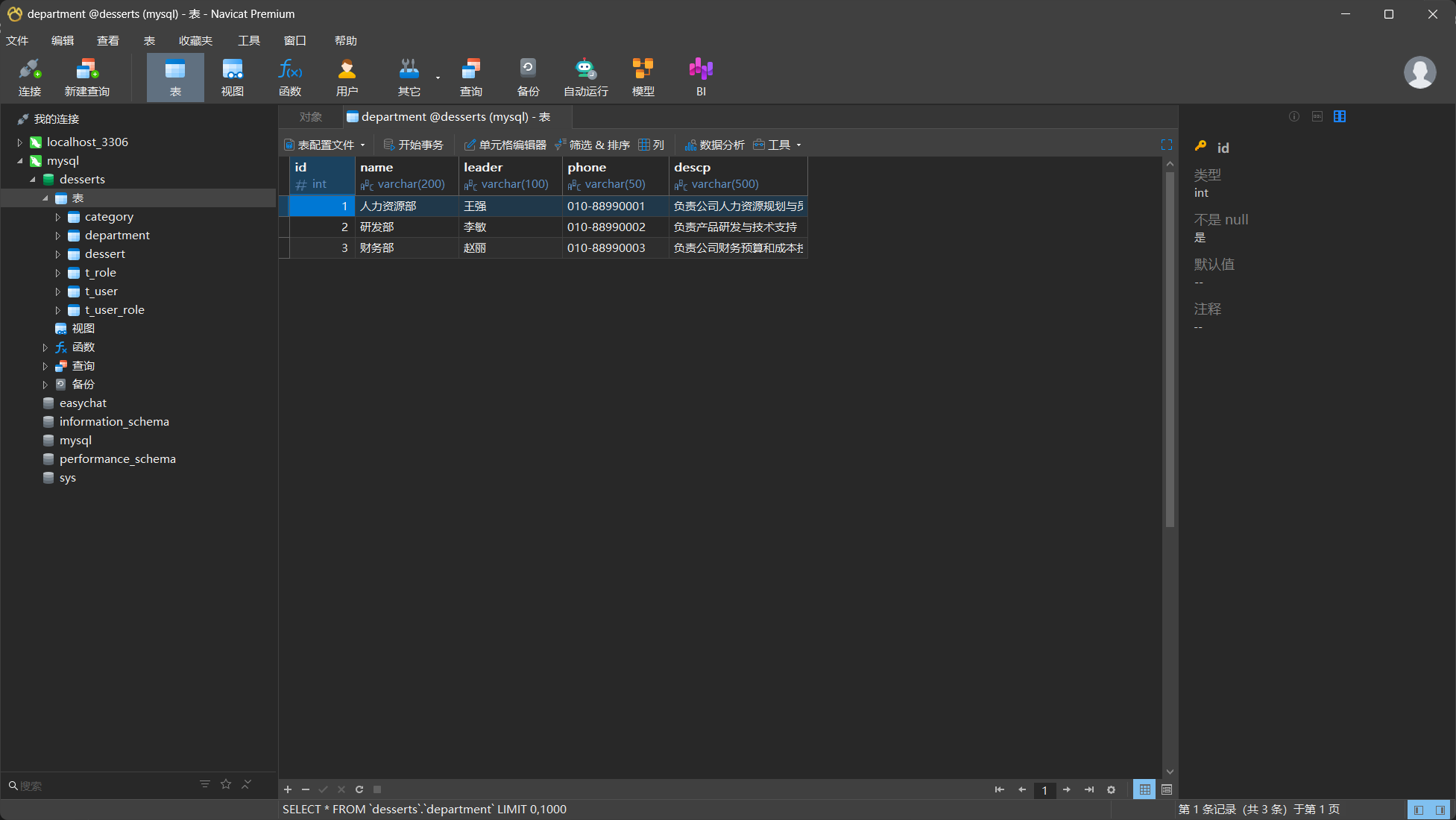
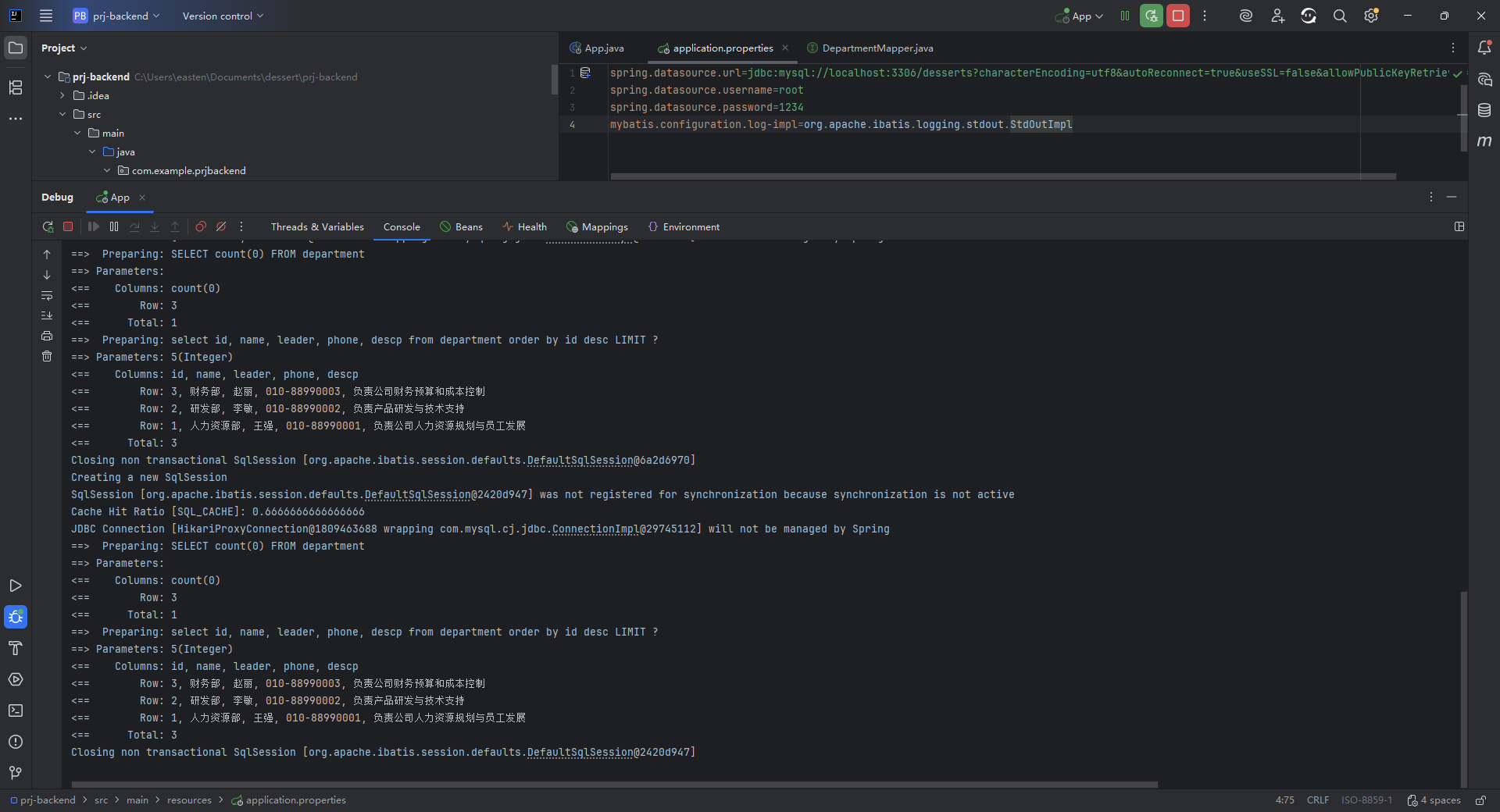
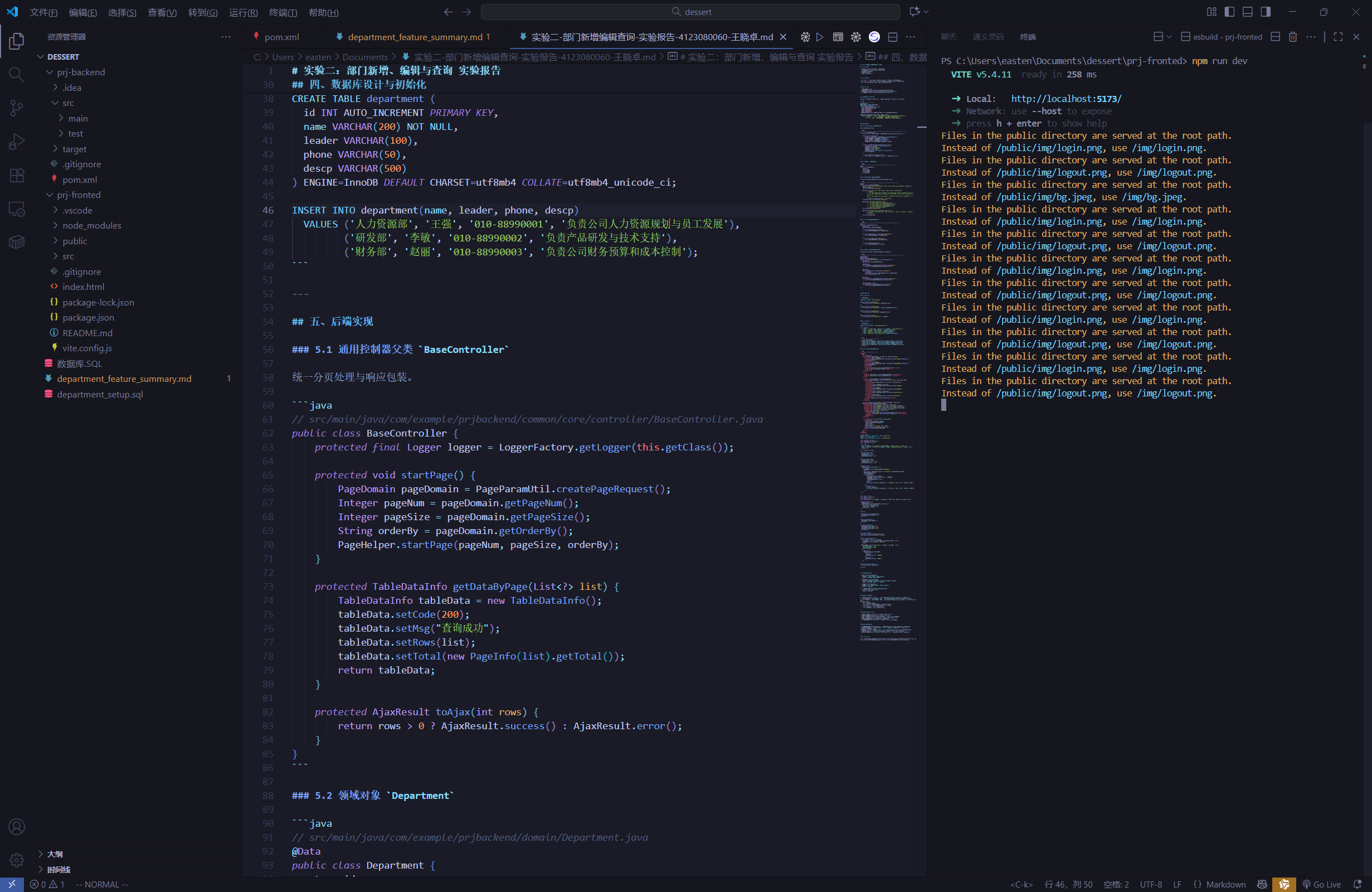
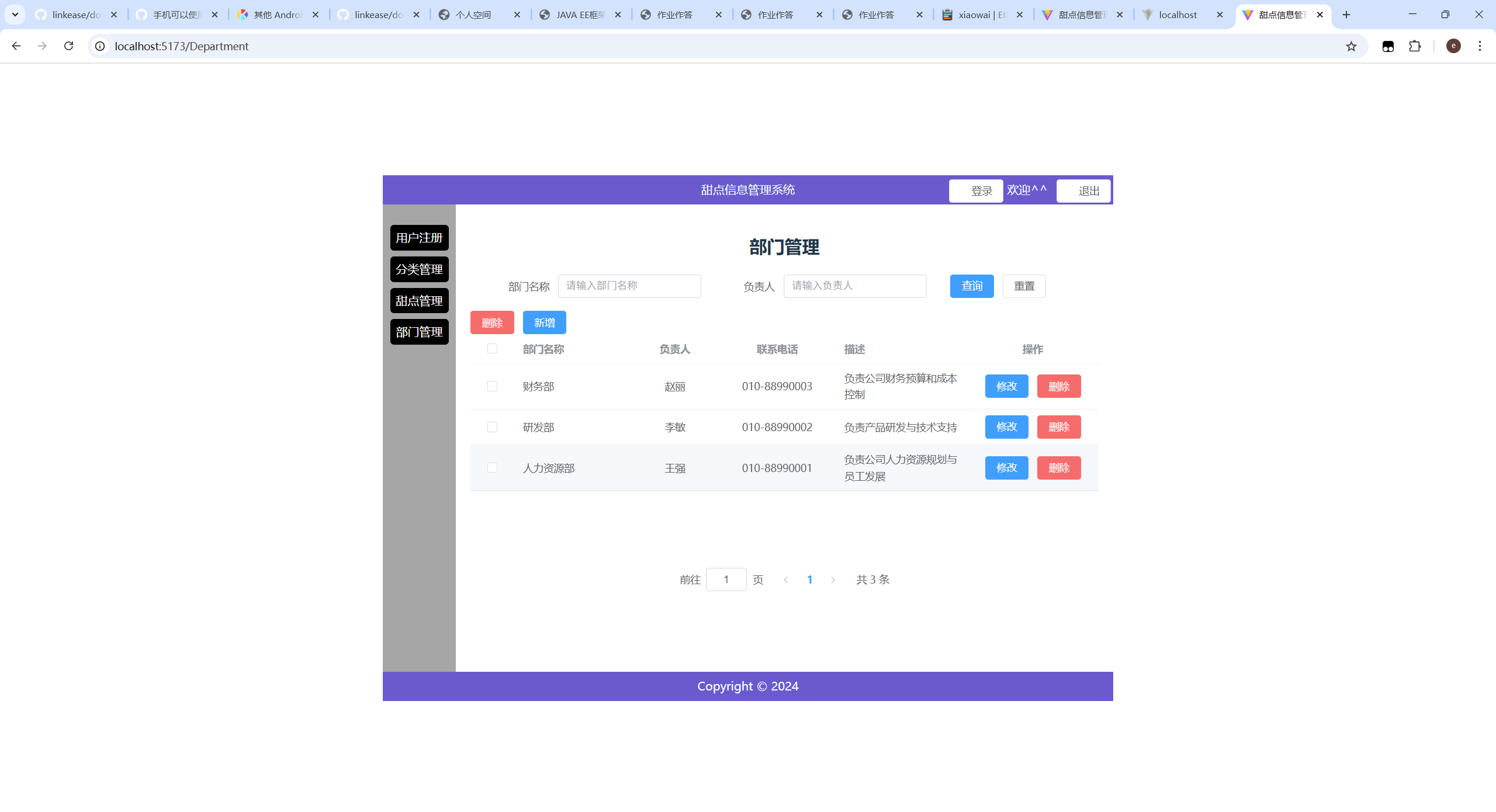
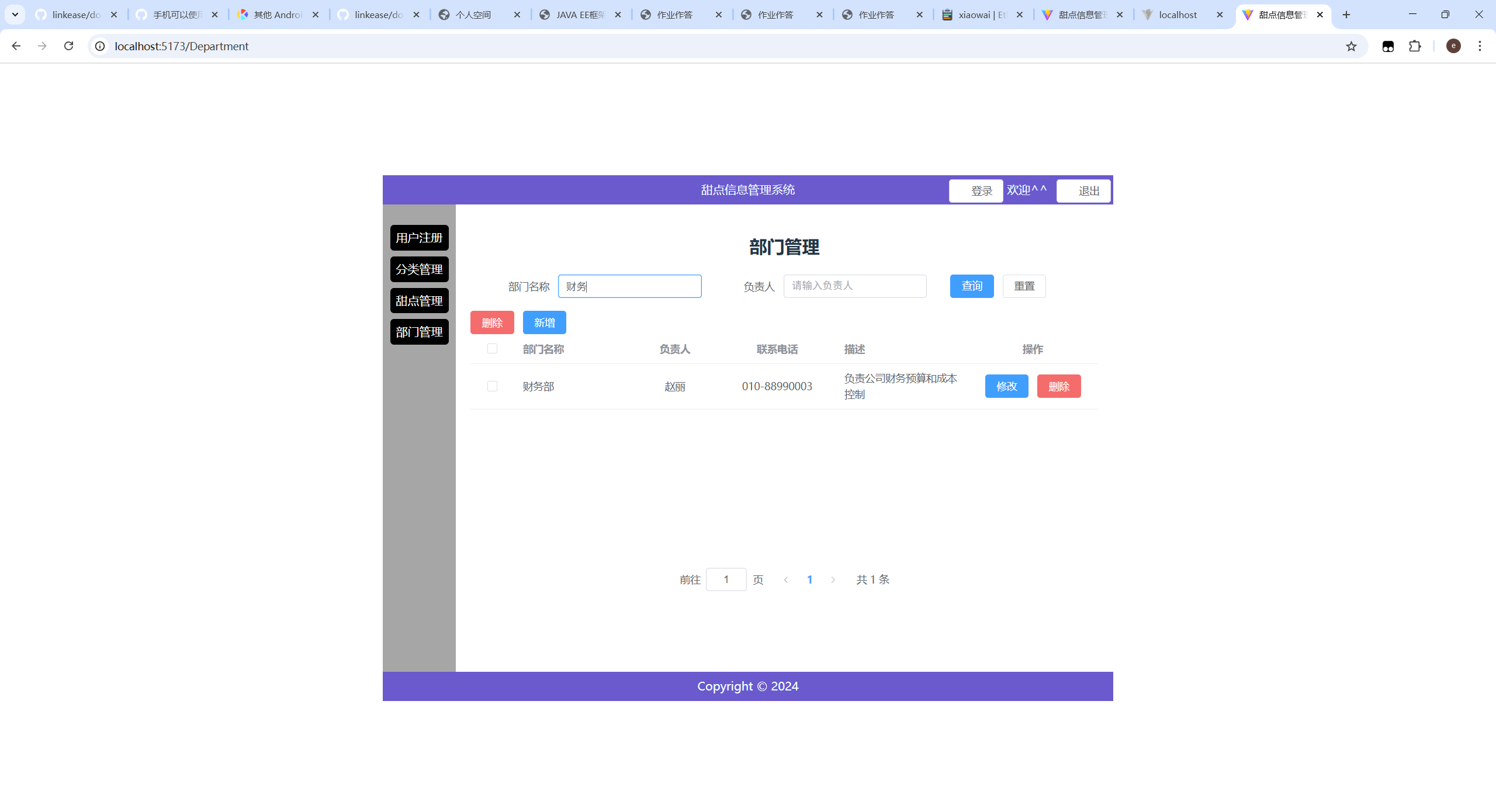
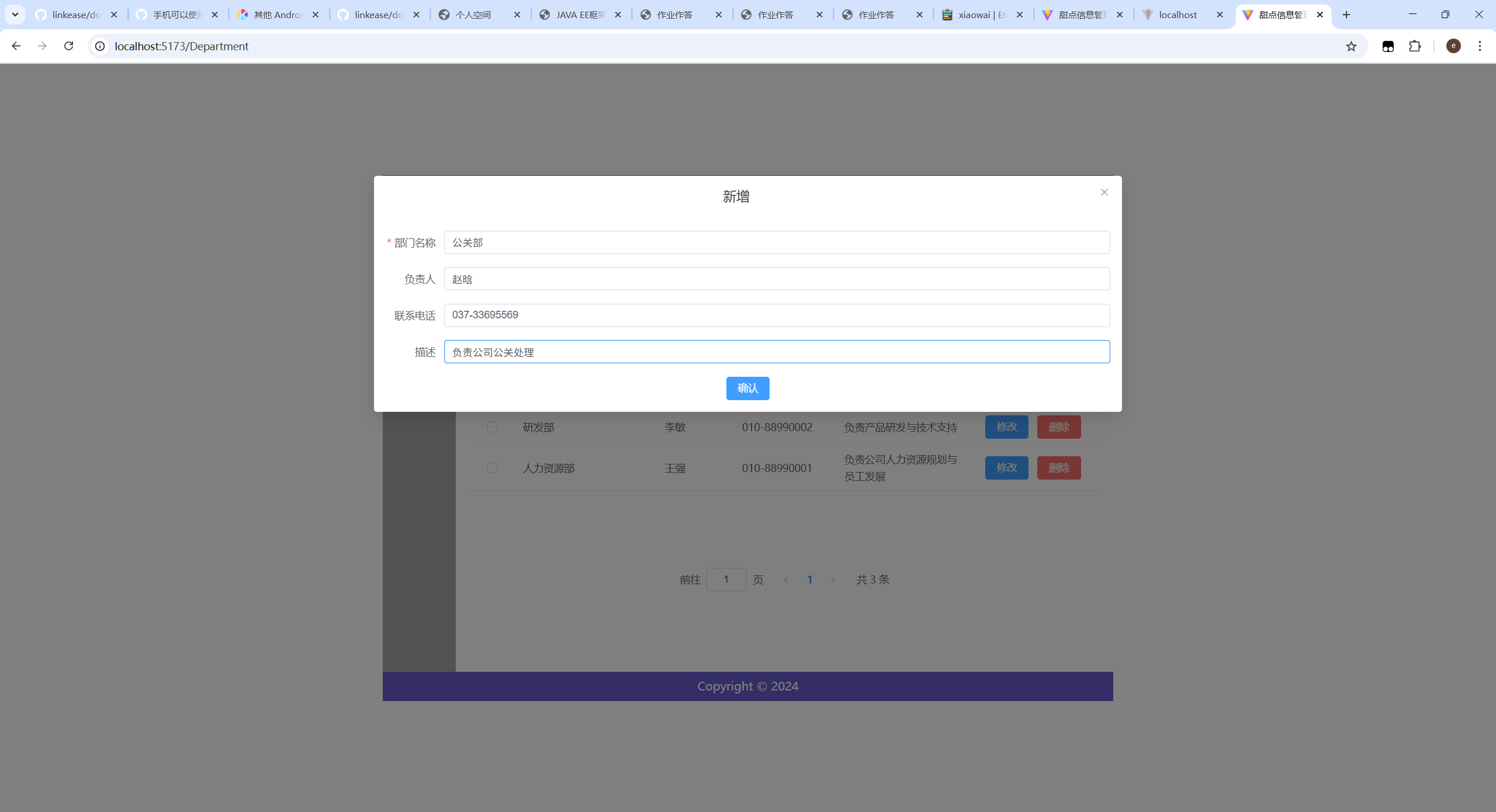
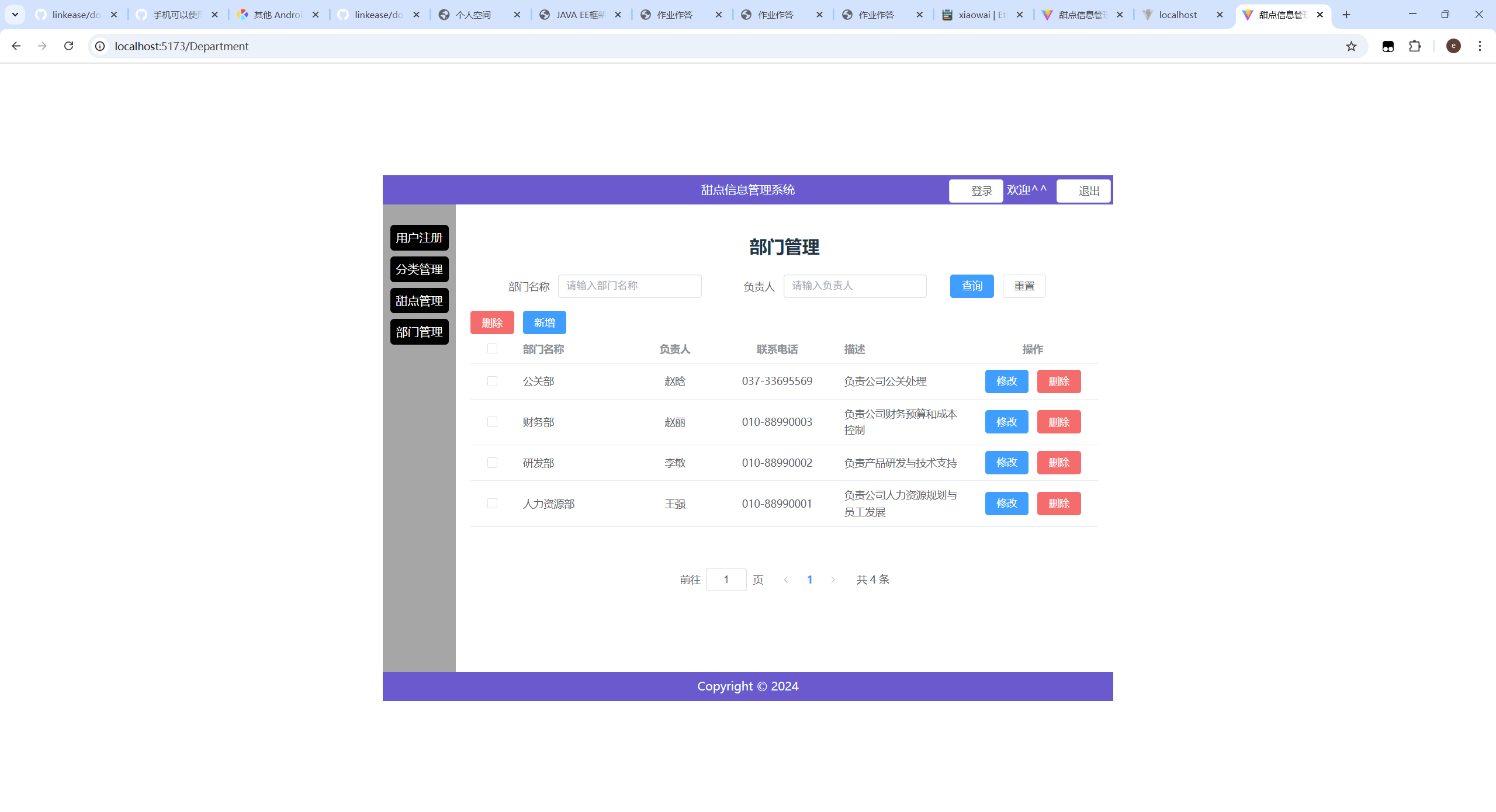
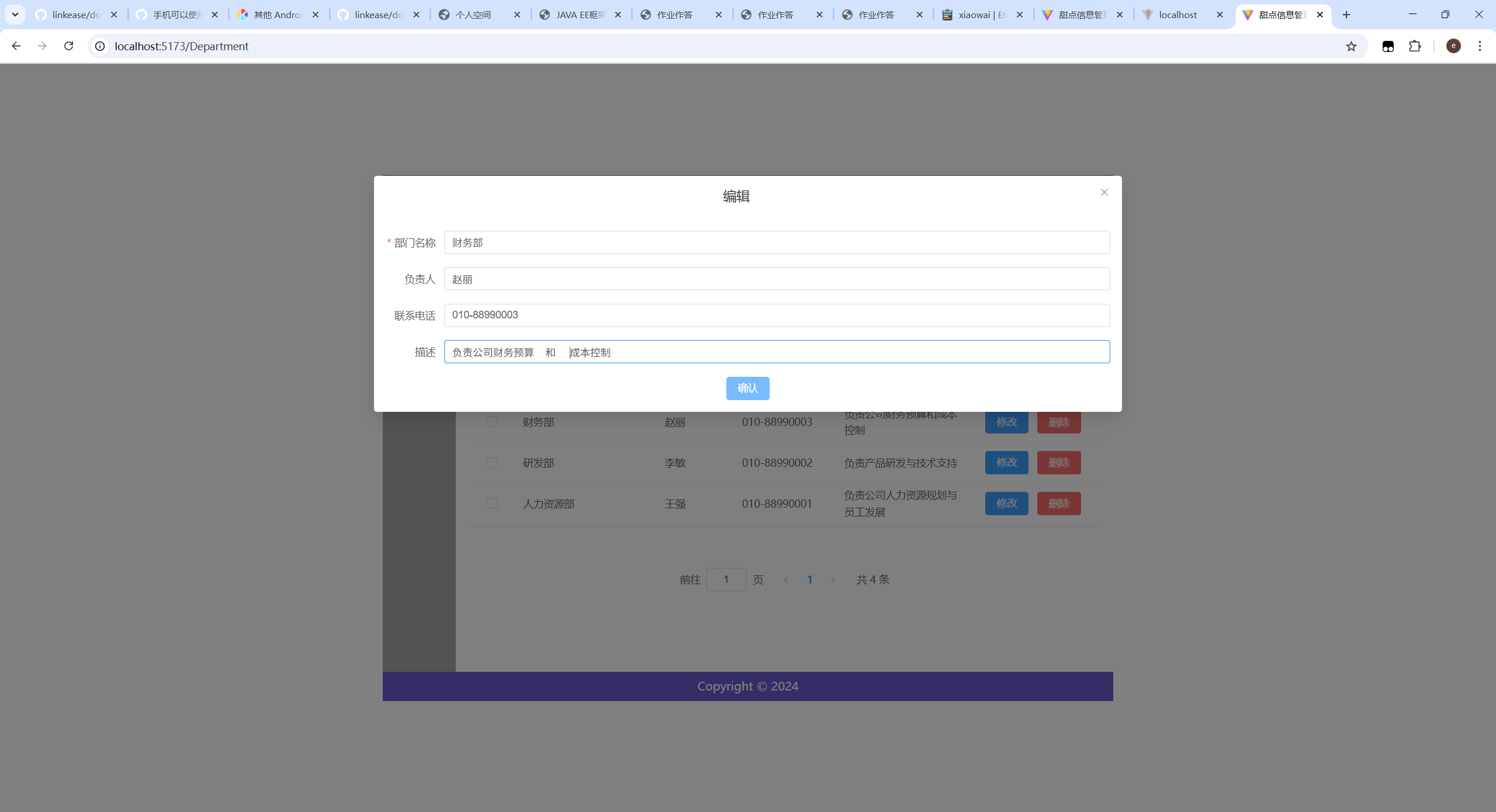
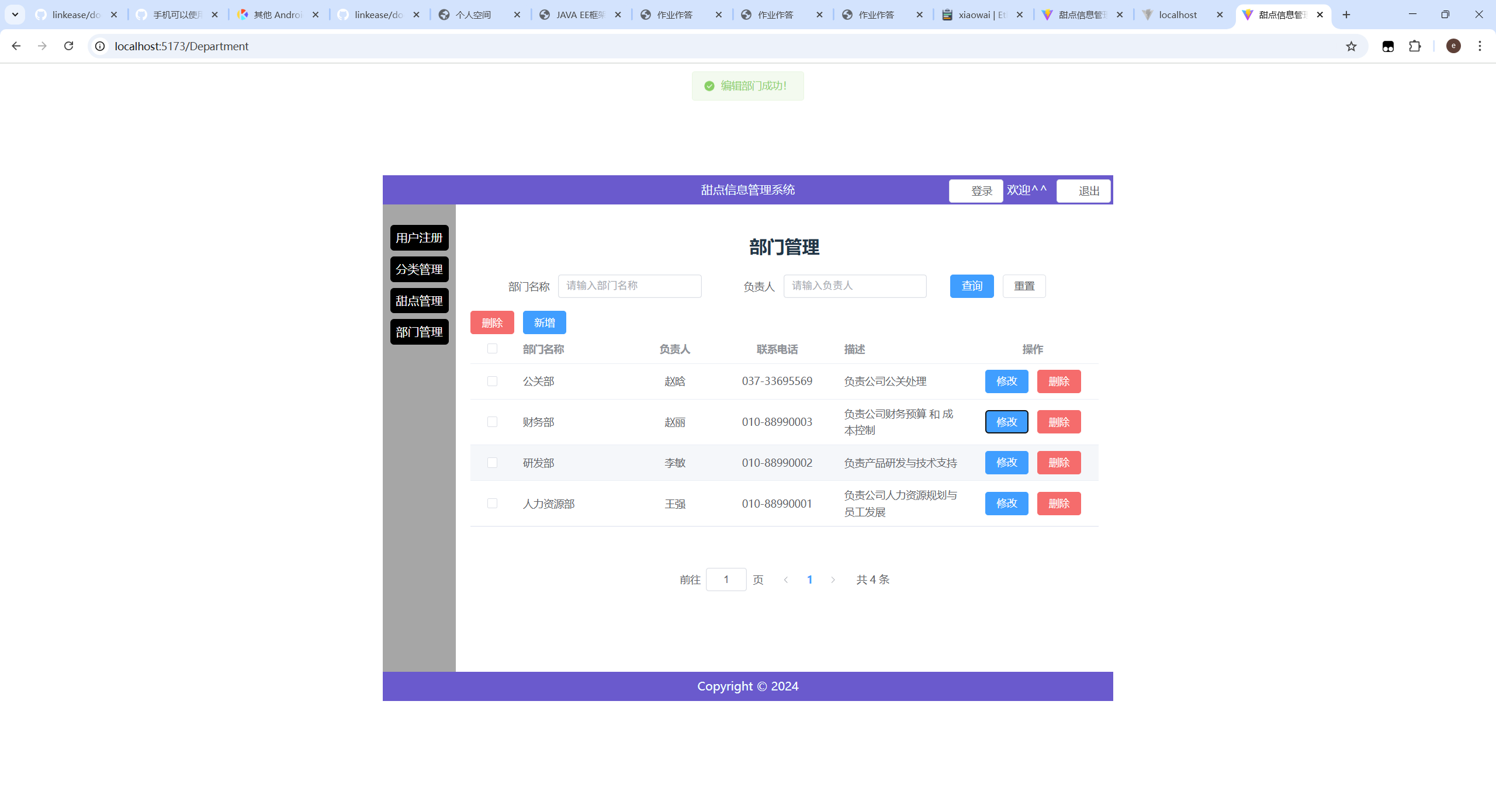
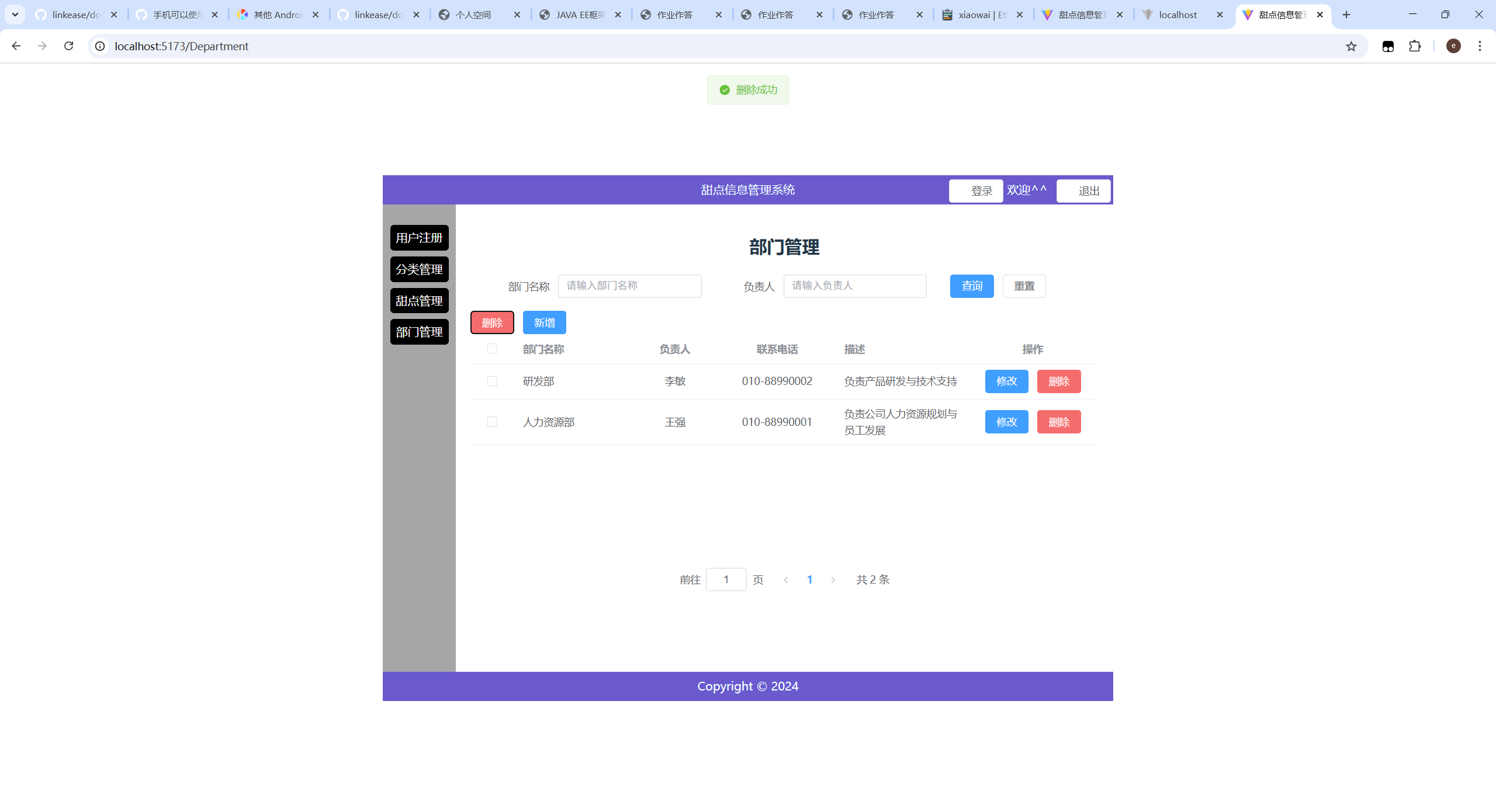
### 6.3 视图组件 Department.vue

<!-- src/views/Department.vue -->  
<template>  
 <div>  
 <h2>部门管理</h2>  
 <el-form :model="queryParams" :inline="true" label-width="80px">  
 <el-form-item label="部门名称">  
 <el-input v-model="queryParams.name" placeholder="请输入部门名称" @keyup.enter.native="handleQuery" />  
 </el-form-item>  
 <el-form-item label="负责人">  
 <el-input v-model="queryParams.leader" placeholder="请输入负责人" @keyup.enter.native="handleQuery" />  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" @click="handleQuery">查询</el-button>  
 <el-button @click="resetQuery">重置</el-button>  
 </el-form-item>  
 </el-form>  
  
 <el-row>  
 <el-button type="danger" @click="handleDeleteBatch">删除</el-button>  
 <el-button type="primary" @click="openCreate">新增</el-button>  
 </el-row>  
  
 <el-dialog :title="titleOp" v-model="visibleDialog">  
 <el-form :model="department" :rules="rules" ref="saveForm" label-width="80px">  
 <el-form-item>  
 <el-input v-model="department.id" v-if="false" :disabled="true" />  
 </el-form-item>  
 <el-form-item label="部门名称" prop="name">  
 <el-input v-model="department.name" placeholder="请输入部门名称" />  
 </el-form-item>  
 <el-form-item label="负责人">  
 <el-input v-model="department.leader" placeholder="请输入负责人" />  
 </el-form-item>  
 <el-form-item label="联系电话" prop="phone">  
 <el-input v-model="department.phone" placeholder="请输入联系电话" />  
 </el-form-item>  
 <el-form-item label="描述">  
 <el-input v-model="department.descp" placeholder="请输入描述" />  
 </el-form-item>  
 <el-button type="primary" @click="save">确认</el-button>  
 </el-form>  
 </el-dialog>  
  
 <el-table style="min-height:300px" v-loading="loading" :data="list" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="60" align="center" />  
 <el-table-column label="部门名称" align="left" prop="name" width="160" />  
 <el-table-column label="负责人" align="center" prop="leader" width="120" />  
 <el-table-column label="联系电话" align="center" prop="phone" width="160" />  
 <el-table-column label="描述" align="left" prop="descp" />  
 <el-table-column label="操作" align="center">  
 <template #default="scope">  
 <el-button type="primary" @click="openEdit(scope.row)">修改</el-button>  
 <el-button type="danger" @click="handleDelete(scope.row)">删除</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
  
 <el-row type="flex" justify="center" align="middle">  
 <el-pagination  
 :current-page="queryParams.pageNum"  
 :page-size="queryParams.pageSize"  
 :total="total"  
 :pager-count="0"  
 layout="jumper, prev, pager, next, total"  
 style="text-align: center;margin-top: 20px;"  
 @current-change="handleCurrentChange"  
 />  
 </el-row>  
 </div>  
</template>  
  
<script setup>  
import { ElMessage, ElMessageBox } from 'element-plus'  
import { ref } from 'vue'  
import \* as DepartmentService from '../api/Department'  
  
const titleOp = ref('新增')  
const visibleDialog = ref(false)  
const department = ref({})  
  
const rules = {  
 name: [{ required: true, message: '请输入部门名称', trigger: ['blur', 'submit'] }],  
 phone: [{ pattern: /^(|[0-9\-+()]{5,20})$/, message: '请输入合法联系电话', trigger: ['blur', 'submit'] }]  
}  
const saveForm = ref({})  
  
function openCreate() {  
 titleOp.value = '新增'  
 department.value = {}  
 visibleDialog.value = true  
}  
  
function openEdit(row) {  
 titleOp.value = '编辑'  
 department.value = { ...row }  
 visibleDialog.value = true  
}  
  
function save() {  
 saveForm.value.validate(valid => {  
 if (!valid) {  
 ElMessage.warning('表单数据有问题，请先修正！')  
 } else {  
 const action = department.value.id === undefined ? DepartmentService.add : DepartmentService.edit  
 action(department)  
 .then(resp => {  
 if (resp.data.code === 200) {  
 ElMessage.success(titleOp.value + '部门成功！')  
 department.value = {}  
 visibleDialog.value = false  
 getList()  
 } else {  
 ElMessage({ message: titleOp.value + '部门失败！', type: 'error', duration: 12000 })  
 }  
 })  
 .catch(error => {  
 if (error.response) {  
 ElMessage({ message: titleOp.value + '部门异常！', type: 'error', duration: 120000 })  
 }  
 })  
 }  
 })  
}  
  
const list = ref([])  
const loading = ref(true)  
const total = ref(0)  
const queryParams = ref({ pageNum: 1, pageSize: 5, name: null, leader: null, phone: null })  
  
function getList() {  
 loading.value = true  
 DepartmentService.list(queryParams).then(resp => {  
 list.value = resp.data.rows  
 total.value = resp.data.total  
 loading.value = false  
 })  
}  
  
getList()  
  
function handleCurrentChange(val) {  
 queryParams.value.pageNum = val  
 getList()  
}  
  
function handleQuery() {  
 queryParams.value.pageNum = 1  
 getList()  
}  
  
function resetQuery() {  
 queryParams.value.name = null  
 queryParams.value.leader = null  
 queryParams.value.phone = null  
 handleQuery()  
}  
  
const ids = ref([])  
function handleSelectionChange(selection) {  
 ids.value = selection.map(item => item.id)  
}  
  
function handleDelete(row) {  
 const ids2del = row.id || ids.value  
 if (!ids2del || (Array.isArray(ids2del) && ids2del.length === 0)) {  
 ElMessage.warning('请先选择要删除的部门')  
 return  
 }  
 ElMessageBox.confirm('确认删除ID为' + ids2del + '的部门？', '警告', {  
 confirmButtonText: '删除',  
 cancelButtonText: '取消',  
 type: 'warning'  
 })  
 .then(() => {  
 DepartmentService.del(ids2del)  
 .then(() => {  
 getList()  
 ElMessage.success('删除成功')  
 })  
 .catch(() => {  
 ElMessage.warning('删除失败')  
 })  
 })  
}  
  
function handleDeleteBatch() {  
 handleDelete({ id: ids.value })  
}  
</script>

## 七、接口与数据约定

* **新增**：POST /department
  + 请求体：{ name, leader, phone, descp }
  + 成功：{ code:200, msg:"success" }
* **分页查询**：GET /department
  + 查询参数：pageNum、pageSize、name?、leader?、phone?
  + 成功：{ code:200, rows:[...], total:n }
* **编辑**：PUT /department
  + 请求体：{ id, name?, leader?, phone?, descp? }
  + 成功：{ code:200 }
* **删除（单/批量）**：DELETE /department/{ids}
  + 路径参数：ids 支持 1,2,3
  + 成功：{ code:200 }

## 八、运行与联调步骤

1. **数据库初始化**：在 MySQL 中执行上述 department\_setup.sql，确保存在库 desserts。
2. ​​
3. **启动后端**：在 prj-backend/ 下运行 mvn spring-boot:run（账号 root/1234 可连库）。
4. ​​
5. **启动前端**：在 prj-fronted/ 下运行 npm install（首次）后 npm run dev，浏览器访问 Vite 地址。
6. ​​
7. **功能验证**：
   * 打开 “部门管理” 页面。
   * ​​
   * 在顶部表单输入部门名称/负责人进行查询并验证分页。
   * ​​
   * 点击 “新增” 打开对话框，填写名称、电话等进行保存。
   * ​​
   * ​​
   * 选择一条记录点击 “修改” 编辑并保存。
   * ​​
   * ​​
   * 勾选多条记录点击 “删除” 验证批量删除。
   * ​​

## 九、常见问题与解决

* **跨域问题（CORS）**：后端控制器已加 @CrossOrigin，确保本地前后端端口不同场景可联调。
* **分页参数缺失**：前端 queryParams 初始化 { pageNum:1, pageSize:5 }，切页事件使用 handleCurrentChange 同步。
* **MyBatis 动态 SQL 尾逗号**：<set> 标签自动处理字段逗号；仅更新传入字段，避免全量覆盖。
* **电话校验**：使用 /^(|[0-9\-+()]{5,20})$/，允许空或 5-20 位常见电话字符。
* **数据库连接失败**：确认 desserts 数据库存在且凭据为 root/1234，字符集 utf8mb4。

## 十、实验小结

本实验完成了部门管理模块端到端实现：数据库表设计、后端 REST 接口、前端页面与交互、以及分页和校验逻辑。通过联调掌握了前后端协作与问题排查的方法，为后续更复杂业务模块（如权限、审计等）打下基础。