

**数据库课程实践报告**

学院名称： 数学与统计学院

专业： 信息与计算科学（嵌入式培养）

班级： 18信嵌（1）班

学号：

姓名：

2020年 12月30日

**目录**

[一、项目简介 2](#_Toc60232696)

[1.1项目介绍 2](#_Toc60232697)

[1.2开发工具 2](#_Toc60232698)

[1.3开发环境 2](#_Toc60232699)

[二、需求分析 2](#_Toc60232700)

[三、详细设计 2](#_Toc60232701)

[3.1登录功能 2](#_Toc60232702)

[3.2用户管理 4](#_Toc60232703)

[3.3角色管理 10](#_Toc60232704)

[3.4菜单管理 15](#_Toc60232705)

[四、总结与展望 19](#_Toc60232706)

# 一、项目简介

**1.1项目介绍**

本项目主要运用Java语言，利用docker、vue-js以及VMware Workstation Pro在eclipse环境下，练习了简单基础的人事管理系统项目并制作了相关网页。

利用该项目设计成果能够简单地完成对员工数据的筛选、更改及更新，方便便于实际应用中的人事管理。

**1.2开发工具**

Java语言 、docker、VMware Workstation Pro、vue-js

**1.3开发环境**

Eclipse开发环境

# 二、需求分析

人事管理系统应有简单的登录功能及用户管理、角色管理、菜单管理。作为练习，我们主要完成了用户管理及角色管理部分。

# 三、详细设计

**3.1登录功能**

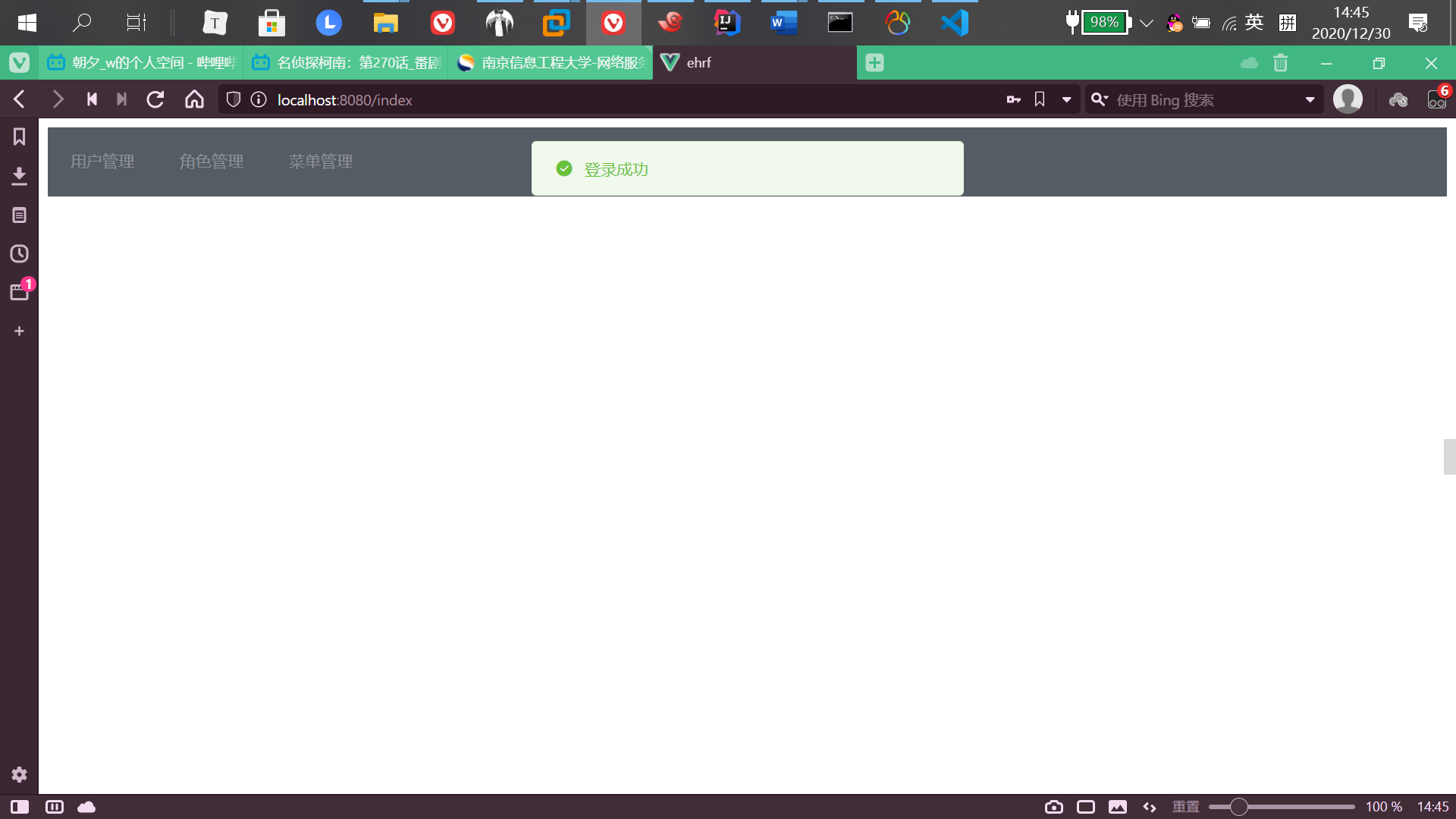
代码：

|  |
| --- |
| <template>    <div>      <h1>ehr人事管理系统</h1>      <div *class*="input-box">        <el-input  *v-model*="user.username" *placeholder*="请输入用户名" />      </div>      <div *class*="input-box">        <el-input  *v-model*="user.password" *placeholder*="请输入密码"  *show-password*/>      </div>      <div *class*="input-box">          <el-button *type*="primary" *v-on*:*click*="submitUser">登录</el-button>          <el-button *v-on*:*click*="clearForm">清空</el-button>      </div>    </div>  </template>  <script>  export default {  **data**:function(){          return {              user:{                  username:"",                  password:""              }          }      },      methods:{  *//clearForm是一个方法，将输入框内容清空*  **clearForm**:function(){              this.user.username ="";              this.user.password = "";          },  **submitUser**:function(){  *//将user的数据提交给后台服务器,*  *//post方法接收两个参数，第一个是url，第二个是数据，以json格式提交*              this.$axios.**post**('/login',this.user).**then**(resp=>{                  this.**$message**({                      type:"success",                      message:"登录成功"                  });  *//路由到首页,push方法指定跳转的页面地址，其中地址由path属性绝定*                  this.$router.**push**({path:"/index"})              }).**catch**(error=>this.**$message**({                  type:"error",                  message:"登录失败，请重试"              }))          }      }  };  </script>  <style *scoped*>  .input-box{        margin:0 auto;      margin-bottom:20px;      width: 400px;  }  </style> |

成果展示：



登录成功提示：



**3.2用户管理**

代码：

|  |
| --- |
| <template>    <div>  *<!-- 弹出对话框 -->*      <el-dialog :*visible*.*sync*="dialogVisible" :*title*="formTitle" *v-on*:*close*="clearForm">        <el-form :*model*="formData" :*rules*="rules" *ref*="userForm">          <el-form-item *label*="用户名" *label-width*="100px" *prop*="username">            <el-input *v-model*="formData.username" />          </el-form-item>          <el-form-item *label*="密码" *label-width*="100px" *prop*="password">            <el-input *v-model*="formData.password" />          </el-form-item>          <el-form-item *label*="姓名" *label-width*="100px" *prop*="name">            <el-input *v-model*="formData.name" />          </el-form-item>          <el-form-item *label*="联系方式" *label-width*="100px" *prop*="phone">            <el-input *v-model*="formData.phone" />          </el-form-item>          <el-form-item *label*="角色列表" *label-width*="100px" *prop*="roleList">            <el-select *v-model*="formData.roleList" *multiple*>              <el-option  *v-for*="role in roles"                :*key*="role.id"                :*value*="role.id"                :*label*="role.roleName"              ></el-option>            </el-select>          </el-form-item>        </el-form>        <div>          <el-button *type*="primary" *v-on*:*click*="saveUser">确定</el-button>          <el-button *v-on*:*click*="**clearForm**()">取消</el-button>        </div>      </el-dialog>      <el-button *type*="success" *v-on*:*click*="openNew">新增用户</el-button>      <el-table :*data*="userList">  *<!-- 设置具体的数据列 -->*        <el-table-column *prop*="username" *label*="用户名"></el-table-column>        <el-table-column *prop*="name" *label*="姓名"></el-table-column>        <el-table-column *prop*="phone" *label*="联系方式"></el-table-column>  *<!-- 自定义列 -->*        <el-table-column>          <template *slot*="header" *slot-scope*="scope">            <el-input  *placeholder*="输入用户名进行查询"  *v-model*="usernameQuery"  *v-on*:*keyup*.native.*enter*="findUsers"            />          </template>          <template *slot-scope*="scope">            <el-button *type*="danger" *v-on*:*click*="**deleteUser**(scope.row)"              >删除</el-button            >            <el-button *v-on*:*click*="**openEdit**(scope.row)">编辑</el-button>          </template>        </el-table-column>      </el-table>    </div>  </template>  <script>  export default {  **created**() {  *//去查询userList，向后台查询*      this.**findUsers**();  *//打开页面后去查询所有的可选角色信息*      this.**findAllRoles**();    },    methods: {  **clearForm**(){        this.$refs['userForm'].**resetFields**();        this.dialogVisible = false;      },  */\*\**  *\* 查询所有的角色信息*  *\*/*  **findAllRoles**() {        this.$axios.**get**("/role/all").**then**((resp) => {          this.roles = resp.data;        });      },  */\*\**  *\* 更新一条数据*  *\*/*  **saveUser**() {  *//先验证，验证成功了才能发请求*        this.$refs["userForm"].**validate**((valid) => {  *//如果结果为真，就触发请求发送*          if (valid) {  *//修改formData中roleList的状态*            let newRoles = [];            this.formData.roleList.**forEach**((value) => {              newRoles.**push**({ id: value });            });            this.formData.roleList = newRoles;            if (this.formTitle == "新增用户信息") {  *//通过请求提交formData*              this.$axios.**post**("/user", this.formData).**then**((resp) => {                if (resp.data == "insert success") {                  this.**$message**({                    type: "success",                    message: "新增成功",                  });  *//关闭窗口*                  this.dialogVisible = false;  *//刷新数据*                  this.**findUsers**();                }              });            } else {  *//通过请求提交formData*              this.$axios.**post**("/user/update", this.formData).**then**((resp) => {                if (resp.data == "update success") {                  this.**$message**({                    type: "success",                    message: "更新成功",                  });  *//关闭窗口*                  this.dialogVisible = false;  *//刷新数据*                  this.**findUsers**();                }              });            }          } else {            return false;          }        });      },  **openNew**() {  *//请空验证的状态*  *//设置表单title*        this.formTitle = "新增用户信息";        this.dialogVisible = true;        this.formData.id = "";        this.formData.username = "";        this.formData.password = "";        this.formData.name = "";        this.formData.phone = "";  *//清空角色清单*        this.formData.roleList = [];      },  **openEdit**(row) {  *//请空验证的状*  *//设置表单title*        this.formTitle = "更新用户信息";        this.dialogVisible = true;  *//将这条数据的值设置为formData*        this.formData.id = row.id;        this.formData.username = row.username;        this.formData.password = row.password;        this.formData.name = row.name;        this.formData.phone = row.phone;  *//加载用户已有的角色清单*  *//this.formData.roleList = row.roleList;*  *//将从后台获取的某一个用户的rolelist集合中的每一个role的id存入formData中的roleList数组中*        this.formData.roleList = [];        row.roleList.**forEach**((role) => {          this.formData.roleList.**push**(role.id);        });      },  **findUsers**() {  *//具体的查询逻辑*        this.$axios          .**post**("/user/find", { username: this.usernameQuery })          .**then**((resp) => (this.userList = resp.data));      },  *//row就是要删除的一行数据*  **deleteUser**(row) {  **alert**(JSON.**stringify**(row));  *//弹出确认框，确认是否删除，如果确认就向后台发出删除请求*        this.**$confirm**("是否确认删除用户信息？", "警告", {          confirmButtonText: "确定",          cancelButtonText: "取消",          type: "warning",        }).**then**(() => {  *//去删除数据*          this.$axios.**delete**("/user/" + row.id).**then**((resp) => {            if (resp.data == "delete success") {  *//删除成功了，给一个消息提示，并且刷新前台的user信息*              this.**$message**({                type: "success",                message: "删除成功！",              });              this.**findUsers**();            }          });        });      },    },  **data**() {  *//验证手机规则的方法*      var **checkPhone** = (rule,value,callback)=>{            if(/^1(?:70\d|(?:9[89]|8[0-24-9]|7[135-8]|66|5[0-35-9])\d|3(?:4[0-8]|[0-35-9]\d))\d{7}$/.**test**(value)){  **callback**();          }else{            return **callback**(new **Error**("手机号格式非法！"));          }      };      return {  */\* 验证规则对象 \*/*        rules: {          username: [  */\* 判定是否为空 \*/*            { required: true, message: "用户名不能为空", trigger: "blur" },  */\* 判定是否长度在3-8之间 \*/*            { min: 3, max: 8, message: "用户名长度3-8之间", trigger: "blur" },          ],          password:[  */\* 判定是否为空 \*/*            { required: true, message: "密码不能为空", trigger: "blur" }          ],          name:[            { required: true, message: "姓名不能为空", trigger: "blur" },            { min: 2, max: 5, message: "姓名长度2-5之间", trigger: "blur" },          ],          phone:[            { required: true, message: "联系方式不能为空", trigger: "blur" },            {validator:checkPhone,trigger:"blur"}          ],          roleList:[            { required: true, message: "角色不能为空", trigger: "change" }          ]        },        roles: [], *//所有可选角色*        formTitle: "",        formData: {          id: "",          username: "",          password: "",          name: "",          phone: "",          roleList: [],        },        usernameQuery: "",        userList: [],        dialogVisible: false,      };    },  };  </script>  <style>  </style> |

成果展示：



**3.3角色管理**

代码：

|  |
| --- |
| <template>    <div>  *<!-- 弹出对话框 -->*      <el-dialog :*visible*.*sync*="dialogVisible" :*title*="formTitle" *v-on*:*close*="clearForm">        <el-form :*model*="formData" *ref*="userForm">          <el-form-item *label*="角色名" *label-width*="100px" *prop*="roleName">            <el-input *v-model*="formData.roleName" />          </el-form-item>          <el-form-item *label*="菜单列表" *label-width*="100px" *prop*="menuList">            <el-select *v-model*="formData.menuList" *multiple*>              <el-option  *v-for*="menu in menus"                :*key*="menu.id"                :*value*="menu.id"                :*label*="menu.menuName"              ></el-option>            </el-select>          </el-form-item>        </el-form>        <div>          <el-button *type*="primary" *v-on*:*click*="saveRole">确定</el-button>          <el-button *v-on*:*click*="**clearForm**()">取消</el-button>        </div>      </el-dialog>      <el-button *type*="success" *v-on*:*click*="openNew">新增角色</el-button>      <el-table :*data*="roleList">  *<!-- 设置具体的数据列 -->*        <el-table-column *prop*="roleName" *label*="角色名"></el-table-column>  *<!-- 自定义列 -->*        <el-table-column>          <template *slot*="header" *slot-scope*="scope">            <el-input  *placeholder*="输入角色名进行查询"  *v-model*="rolenameQuery"  *v-on*:*keyup*.native.*enter*="findRoles"            />          </template>          <template *slot-scope*="scope">            <el-button *type*="danger" *v-on*:*click*="**deleteRole**(scope.row)"              >删除</el-button            >            <el-button *v-on*:*click*="**openEdit**(scope.row)">编辑</el-button>          </template>        </el-table-column>      </el-table>    </div>  </template>  <script>  export default {  **created**() {  *//去查询userList，向后台查询*      this.**findRoles**();  *//打开页面后去查询所有的可选角色信息*      this.**findAllMenus**();    },    methods: {  **clearForm**(){        this.$refs['userForm'].**resetFields**();        this.dialogVisible = false;      },  */\*\**  *\* 查询所有的角色信息*  *\*/*  **findAllMenus**() {        this.$axios.**get**("/menu/all").**then**((resp) => {          this.menus = resp.data;        });      },  */\*\**  *\* 更新一条数据*  *\*/*  **saveRole**() {            let newMenus = [];            this.formData.menuList.**forEach**((value) => {              newMenus.**push**({ id: value });            });            this.formData.menuList = newMenus;            if (this.formTitle == "新增角色信息") {  *//通过请求提交formData*              this.$axios.**post**("/role", this.formData).**then**((resp) => {                if (resp.data == "insert success") {                  this.**$message**({                    type: "success",                    message: "新增成功",                  });  *//关闭窗口*                  this.dialogVisible = false;  *//刷新数据*                  this.**findRoles**();                }              });            } else {  *//通过请求提交formData*              this.$axios.**post**("/role/update", this.formData).**then**((resp) => {                if (resp.data == "update success") {                  this.**$message**({                    type: "success",                    message: "更新成功",                  });  *//关闭窗口*                  this.dialogVisible = false;  *//刷新数据*                  this.**findRoles**();                }              });            }          },  **openNew**() {  *//请空验证的状态*  *//设置表单title*        this.formTitle = "新增角色信息";        this.dialogVisible = true;        this.formData.id = "";        this.formData.roleName = "";  *//清空角色清单*        this.formData.menuList = [];      },  **openEdit**(row) {  *//请空验证的状*  *//设置表单title*        this.formTitle = "更新角色信息";        this.dialogVisible = true;  *//将这条数据的值设置为formData*        this.formData.id = row.id;        this.formData.roleName = row.roleName;  *//加载用户已有的角色清单*  *//this.formData.roleList = row.roleList;*  *//将从后台获取的某一个用户的rolelist集合中的每一个role的id存入formData中的roleList数组中*        this.formData.menuList = [];        row.menuList.**forEach**((menu) => {          this.formData.menuList.**push**(menu.id);        });      },  **findRoles**() {  *//具体的查询逻辑*        this.$axios          .**post**("/role/find", { roleName: this.rolenameQuery })          .**then**((resp) => (this.roleList = resp.data));      },  *//row就是要删除的一行数据*  **deleteRole**(row) {  **alert**(JSON.**stringify**(row));  *//弹出确认框，确认是否删除，如果确认就向后台发出删除请求*        this.**$confirm**("是否确认删除用户信息？", "警告", {          confirmButtonText: "确定",          cancelButtonText: "取消",          type: "warning",        }).**then**(() => {  *//去删除数据*          this.$axios.**delete**("/role/" + row.id).**then**((resp) => {            if (resp.data == "delete success") {  *//删除成功了，给一个消息提示，并且刷新前台的user信息*              this.**$message**({                type: "success",                message: "删除成功！",              });              this.**findRoles**();            }          });        });      },  },  **data**() {      return {          menus: [], *//所有可选菜单*          formTitle: "",          formData: {              id: "",              roleName: "",              menuList: [],          },          rolenameQuery: "",          roleList: [],          dialogVisible: false,      };  },  };  </script>  <style>  </style> |

成果展示：



**3.4菜单管理**

代码：

|  |
| --- |
| <template>    <div>      <el-dialog :*visible*.*sync*="dialogVisible" :*title*="formTitle" *v-on*:*close*="clearForm">        <el-form :*model*="formData" *ref*="userForm">          <el-form-item *label*="菜单名" *label-width*="100px" *prop*="menuName">            <el-input *v-model*="formData.menuName" />          </el-form-item>            <el-form-item *label*="url" *label-width*="100px" *prop*="url">            <el-input *v-model*="formData.url" />          </el-form-item>        </el-form>        <div>          <el-button *type*="primary" *v-on*:*click*="saveMenu">确定</el-button>          <el-button *v-on*:*click*="**clearForm**()">取消</el-button>        </div>      </el-dialog>      <el-button *type*="success" *v-on*:*click*="openNew">新增菜单</el-button>      <el-table :*data*="MenuList">  *<!-- 设置具体的数据列 -->*        <el-table-column *prop*="menuName" *label*="页面名"></el-table-column>  *<!-- 自定义列 -->*        <el-table-column>          <template *slot*="header" *slot-scope*="scope">            <el-input  *placeholder*="输入菜单名进行查询"  *v-model*="menunameQuery"  *v-on*:*keyup*.native.*enter*="findMenus"            />          </template>          <template *slot-scope*="scope">            <el-button *type*="danger" *v-on*:*click*="**deleteMenu**(scope.row)"              >删除</el-button            >            <el-button *v-on*:*click*="**openEdit**(scope.row)">编辑</el-button>          </template>        </el-table-column>      </el-table>    </div>  </template>  <script>  export default {  **created**() {          this.**findMenus**();      },      methods: {  **clearForm**(){              this.$refs['userForm'].**resetFields**();              this.dialogVisible = false;          },  **saveMenu**() {              if (this.formTitle == "新增菜单信息") {  *//通过请求提交formData*                  this.$axios.**post**("/menu", this.formData).**then**((resp) => {                  if (resp.data == "insert success") {                      this.**$message**({                      type: "success",                      message: "新增成功",                      });  *//关闭窗口*                      this.dialogVisible = false;  *//刷新数据*                      this.**findMenus**();                  }                  });              } else {  *//通过请求提交formData*                  this.$axios.**post**("/menu/update", this.formData).**then**((resp) => {                  if (resp.data == "update success") {                      this.**$message**({                      type: "success",                      message: "更新成功",                      });  *//关闭窗口*                      this.dialogVisible = false;  *//刷新数据*                      this.**findMenus**();                  }                  });              }              },  **openNew**() {  *//请空验证的状态*  *//设置表单title*          this.formTitle = "新增菜单信息";          this.dialogVisible = true;          this.formData.id = "";          this.formData.menuName = "";          this.formData.url = "";          },  **openEdit**(row) {  *//请空验证的状*  *//设置表单title*          this.formTitle = "更新菜单信息";          this.dialogVisible = true;  *//将这条数据的值设置为formData*          this.formData.id = row.id;          this.formData.menuName = row.menuName;          this.formData.url = row.url;          },  **findMenus**() {  *//具体的查询逻辑*          this.$axios              .**post**("/menu/find", { menuName: this.menunameQuery })              .**then**((resp) => (this.MenuList = resp.data));          },  *//row就是要删除的一行数据*  **deleteMenu**(row) {  **alert**(JSON.**stringify**(row));  *//弹出确认框，确认是否删除，如果确认就向后台发出删除请求*          this.**$confirm**("是否确认删除用户信息？", "警告", {              confirmButtonText: "确定",              cancelButtonText: "取消",              type: "warning",          }).**then**(() => {  *//去删除数据*              this.$axios.**delete**("/menu/" + row.id).**then**((resp) => {              if (resp.data == "delete success") {  *//删除成功了，给一个消息提示，并且刷新前台的user信息*                  this.**$message**({                  type: "success",                  message: "删除成功！",                  });                  this.**findMenus**();              }              });          });          },      },  **data**() {          return {              formTitle: "",              formData: {                  id: "",                  menuName: "",                  url: "",              },              menunameQuery: "",              MenuList: [],              dialogVisible: false,          };      }  };  </script>  <style>  </style> |

成果展示：



# 四、总结与展望

此次课程设计，在老师的悉心指导与帮助下最终顺利完成。通过本次实践，我学会了如何运用java语言，利用vue-js以及VMware Workstation Pro在eclipse环境下制作基础简单的页面，学习到了很多有用的知识。