**广州城市理工学院**

**2021-2022学年度第2学期**

前端技术开发项目

实训报告

学 院： 计算机工程学院

专业班级： 2019计算机科学与技术5班

姓　　名： 周家振

学　　号： 201910089324

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 一级指标 | 二级指标 | | 三级指标及分值 | | 分值 | 评分及成绩 |
| 得分 |
| 代码  （占60%） | 1.达到基本要求，功能点完整（35%） | | 网页页面布局 | 15 | 45 |  |
| 版面内容 | 10 |
| 素材处理及应用 | 10 |
| 网页对象效果设置 | 10 |
| 3.原创程度（10%） | | | | 10 |  |
| 4.运行测试（5%） | | | | 5 |  |
| 项目报告（20%） | 1.内容与结构完整（10%） | | | | 10 |  |
| 2.规格满足要求，文字描述规范，思路描述清晰（10%） | | | | 10 |  |
| 答辩成绩 （20%） | 1.对实训设计技术和结构熟悉，能进行合理说明（10%） | | | | 10 |  |
| 2.能较好的回答答辩所提问题，解释合理清晰（10%） | | | | 10 |  |
| 合计（百分制） | | | | | |  |
| 总评 | | □优 □良 □中 □及格 □不及格 | | | | 签名:  日期: |

目录

[**一、** **开发环境介绍** 1](#_Toc102151147)

[1.1 开发环境 1](#_Toc102151148)

[1.2 选题原因 1](#_Toc102151149)

[1.3 Vue以及部分依赖项的介绍 1](#_Toc102151150)

[**二、** **页面需求分析** 2](#_Toc102151151)

[2.1 设计要求 2](#_Toc102151152)

[2.2 网站概述 2](#_Toc102151153)

[2.3 可行性分析 2](#_Toc102151154)

[**三、** **页面开发实现** 3](#_Toc102151155)

[3.1页面布局 3](#_Toc102151156)

[3.2功能模块 5](#_Toc102151157)

[3.3数据库相关信息 12](#_Toc102151158)

[**四、** **页面运行测试** 12](#_Toc102151159)

[**五、** **设计总结** 15](#_Toc102151160)

1. **开发环境介绍**

1.1 开发环境

本次的实训的开发环境是在 window11 系统上面实现的，用到的框架是Vue3.0、element-plus和node.js等等，其中还使用了MySQL数据库和express.js进行简单的数据交互，因为能力水平还没有达到完全开发的程度，所以只是做了登录注册两个小接口，只完成注册界面和登录界面的数据交互；编写代码和功能调试都是在Visual Studio Code 中进行，效果图是在 Edge 浏览器上面预览的。

1.2 选题原因

本次项目的实训我的题目是 “学生宿舍管理系统”。在目前的宿舍管理情况中，我了解到宿管阿姨在管理宿舍方面存在一定的困难，所以我选择了这个项目，当然目前这个版本只是面对宿舍管理员有登录注册功能。后续的学生身份登录的话，会在以后的版本中增加。

1.3 Vue以及部分依赖项的介绍

Vue 是一套构建用户界面的渐进式框架，Vue 和其他重量级的框架不同，Vue采用的是自低向上增量开发的设计，Vue的核心库只关注视图层，对于前端新手来说是比较友好；容易和其他库或既有的项目进行整合，另一方面，当与现代化工具链以及各种支持类库结合使用时，Vue也完全能够为复杂的单页面应用提供驱动。

Element-plus是element-ui的升级版本，对于vue3.0的项目更加的友好。element-plus是目前Vue生态圈最火的桌面UI组件库之一，在使用方面，element-plus的视觉设计很符合国人的观赏体验。

Node.js是一个开源和跨平台的JavaScript运行时环境，它几乎是任何类型项目的流行工具。node.js在浏览器以外运行V8 JavaScript引擎（Google Chrome的内核）。这使得node.js的性能非常好。而且npm以其简单的结构帮助node.js生态系统蓬勃发展，现在npm仓库托管了超过1000000个开源包，可以供用户自由使用其中的库。express.js是基于node.js平台，是一个快速、开放、极简的Web开发框架，它提供了一系列的强大特性，能够帮助用户创建各种web应用，我们可以使用npm install express 命令进行下载。Express是一个轻量级的开发框架，封装了node中http核心的模块，专注于业务逻辑的开发。

Vue-router是Vue的路由管理器，是Vue的核心插件，因为Vue项目中一般都是单页面的应用，所以vue-router是应用在单页面中的。在vue项目web应用开发中，路由指的就是URL分配到响应的处理页面，vue-router通过对URL的管理，实现组件和URL的一一对应，以及通过标签<router-link to=’’></router-link>进行组件之间的跳转。

ES6的全称是ECMAScript6，是2015年正式发布的JavaScript脚本语言标准。它的目标是让JavaScript语言能编写复杂的大型应用程序，成为企业级开发语言。

Webpack是代码的编译工具，是一个用于现代JavaScript应用程序的静态模块打包工具，当webpack处理应用程序，会在构建一个依赖图，该依赖图会对应映射到项目所需的模块，并生成相应的bundle。

1. **页面需求分析**

2.1 设计要求

1.根据网站的演示，确定使用页面布局技术（表格，vue.js框架，DIV+CSS或者混合使用）进行页面的设计，使网站功能齐全，界面美观大方，有一定的交互性。

2.要利用CSS+DIV+Vue.js框架技术解决实际的页面设计问题。

3.在编写代码时必须对HTML，CSS,Vue.js等部分代码进行必要的注释，以提高程序代码的可读性；同时代码采用锯齿结构（向右缩进方式），保持代码结构清晰。

2.2 网站概述

本次前端项目实训我选择自己写一个前端小系统。页面包括管理员的注册和登录，同时我用axios实现前端和本地数据库进行数据交互，因为是刚刚开始接触后端，所以只是写了登录和注册接口，实现注册和登录功能。这一次的前端项目我全局使用了element-plus框架。在组件这方面 我用了三个组件，一个头部head组件、一个导航栏nav组件和一个尾部footer组件。

管理员在该系统中，要是之前在该系统的数据库中存在自己的账号密码，就无需再进行管理员身份的注册，就可以在登录页进行登录；要是在该数据库中没有存在相应账号密码，就需要进行点击注册按钮跳转到注册界面进行注册，因为后端逻辑属于刚刚接触，所以没有在数据库中另作表来验证是否能注册。

管理员登录成功后，在页面的最顶部能看见关于学生宿舍管理系统的样式和管理员的姓名和注销功能，在导航栏上面有四个导航菜单，当管理员成功登录，会默认跳到考勤管理，其他的三个导航栏菜单名字分别是荣誉宿舍，用来记录本栋楼层哪一间宿舍获得的荣誉称号；维修工单，当宿舍管理员记录到哪一些宿舍需要进行维修，可以在该页面写上，同时会显示当前维修状态以及进行删除和编辑；个人信息，会显示当前管理员个人的信息，以及方便管理员查看当前日期。

2.3 可行性分析

本次前端项目实训的可行性可以分为三部分，分别是技术可行性、效益可行性、风险可行性。

技术可行性：本次的Vue项目是基于vue+css+div实现的，部分代码涉及到js可选链以及axios，基本上都能实现相应的功能，所以在技术层面上是可以实现的。

效益可行性：本次的Vue项目有关的依赖项都是开源的，但是涉及到后端，在后端维护这方面会产生一定的维护费用，但是本次的vue项目属于课程实践项目，所用到的资源都是本地资源，所以在这个方面也是可以实现的。

风险可行性：本次的前端项目实训属于课程实践项目，不会对外发布，所以不没有存在违法违规行为，也是可行的。

1. **页面开发实现**

3.1页面布局



图3-1-1 登录界面

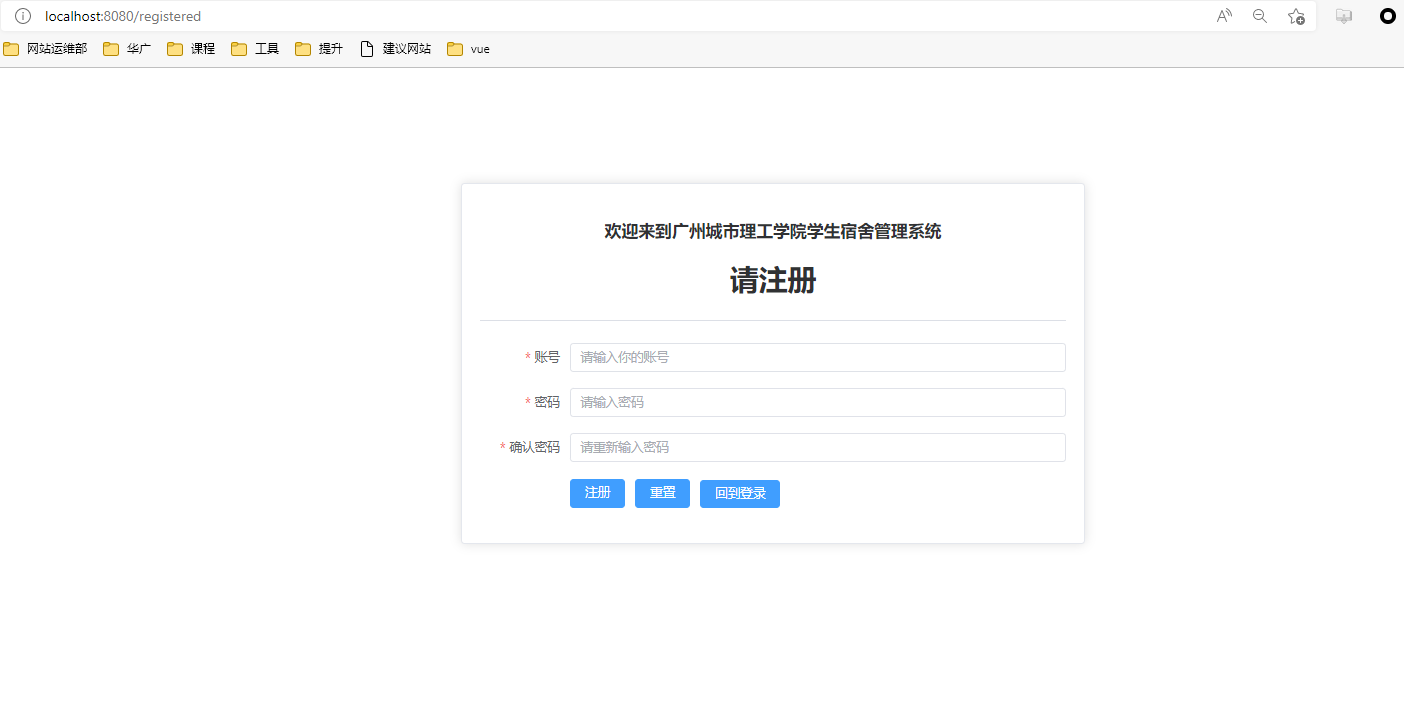


图3-1-2 注册页面



图3-1-3 登录成功界面



图 3-1-4 荣誉宿舍界面



图 3-1-5 维修工单界面

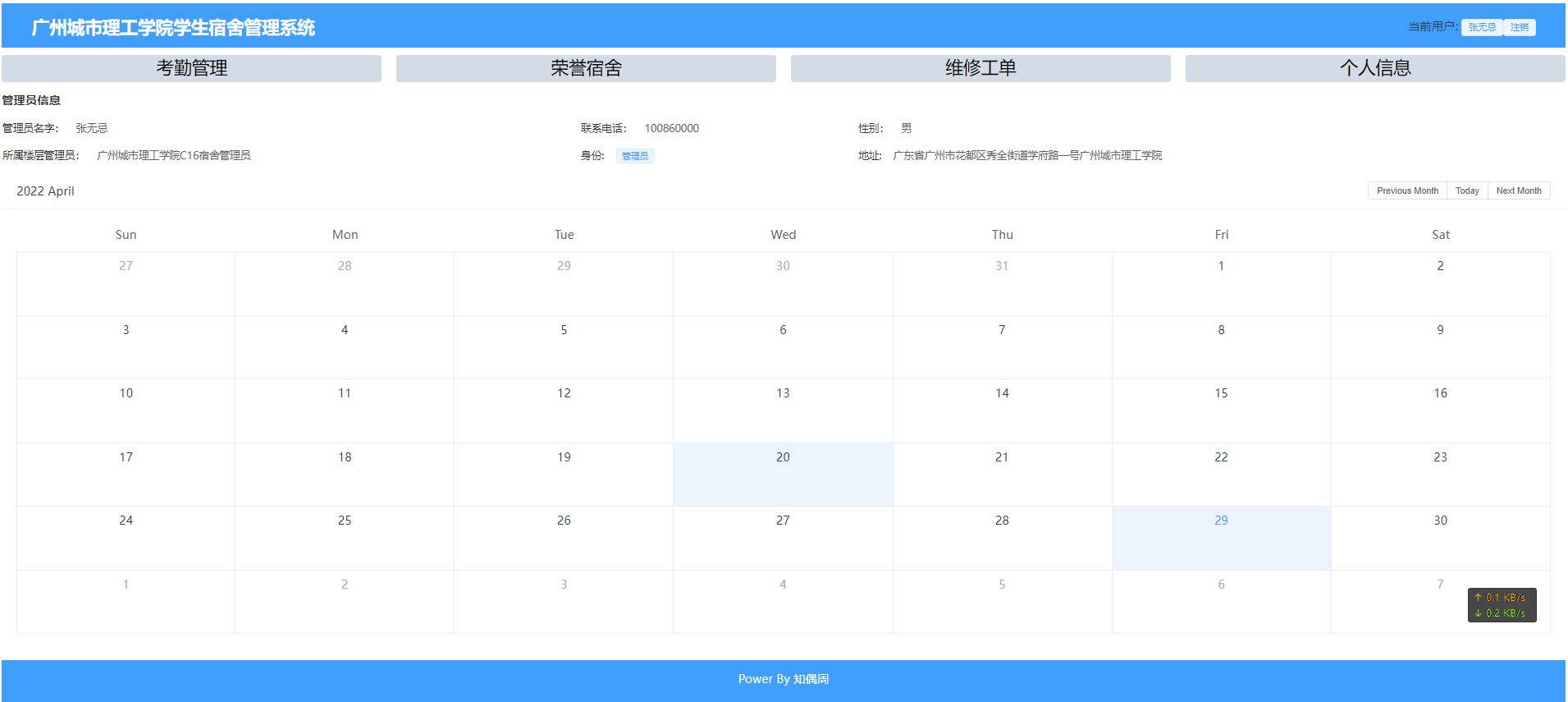


图 3-1-6 宿舍管理员个人信息界面

因为我定义了三个组件，分别是顶部的header组件、菜单的nav组件和尾部的footer组件，所以我每一个页面都用上了这三个组件。

3.2功能模块

图3-1-1和图3-1-2都是用了element-plus中的表单，其中在输入框中，不需要在后端进行是否为空的判断，能够在前端进行判断，能够有效提升用户体验，不需要等到后端返回判断结果，再在前端告诉用户需要填账号密码。以下为代码：

<el-form-item label="密码" prop="userpassword":rules="[ { required: true, message: '密码不能为空' }, { type: 'string' }, ]">

<el-input placeholder="请输入密码" type="password" v-model="info.userpassword" show-password></el-input>

</el-form-item>

代码解读：label是输入框前面的文字，因为el-form-item需要和数据进行绑定才能会判断是否为空，所以需要用到prop去绑定data下面的变量，要是没有进行绑定，就不会进行判断是否为空。:rules是判断的规则；el-input是输入框标签，placeholder是占位符，可以提示用户输入相应的选项，type是代表这个input的类型，可以写text，password等等。v-model是绑定data中的变量，数据的双向绑定也是vue的一大特性。Show-password是可以让用户选择是否显示密码。

输入框下面的几个按钮采用了@click=“函数名字”的点击事件，并且在下面的定义好相应的函数，点击即可触发。以下为代码：

样式代码：

<el-form-item>

<el-button type="primary" @click="onPostBtn">登录</el-button>

<el-button type="primary" @click="reset">重置</el-button>

<el-button type="primary" @click="userregistered">注册</el-button>

</el-form-item>

函数代码：

onPostBtn: function () {

axios({

url: "http://127.0.0.1:3000/login",//进行请求的接口

method: "POST",//是用POST的方式进行请求

data: {

account: this.info.username,//通过双向绑定，能够将前端的数据赋值给后端

password: this.info.userpassword,

},

}).then((res) => {

// //要是成功查询到，就进行路由的跳转

console.log("成功登录");

if (res.data.say === "1") {//在控制台打印出来登录成功，方便排查bug

this.$router.push("/admin");//路由的跳转

} else {

alert(res.data.say);

}

});

},

reset() {

this.info.username = null;

this.info.userpassword = null;

},

userregistered() {

if (true) {

this.$router.push("/registered");

}

},

图3-1-2、图3-1-3、图3-1-4我都采用了想用的结构，在页面的中间都用了element-plus的el-table标签，并且在相应的vue组件中定义好一批数据，再用vue的特性渲染到前端，例如考勤管理，代码如下：

<el-table

:data="tableData"//绑定data中的tableData

height="500"

style="width: 100%"

empty-text="暂无数据"//要是tableData中没有数据，就显示暂无数据

bord

>

<el-table-column

fixed

prop="date"

label="时间"

width="500"

></el-table-column>

<el-table-column prop="name" label="名字" width="580"></el-table-column>

<el-table-column prop="dsc" label="状态"></el-table-column>

<el-table-column fixed="right" label="操作" width="120">

//以下是定义右侧的编辑和删除按钮

<template #default>

<el-button type="text" size="small">删除</el-button>

<el-button type="text" size="small">编辑</el-button>

</template>

</el-table-column>

</el-table>

定义好的一批数据的代码：

//定义的是一个数组

tableData: [

{

date: "2019-05-03",

name: "张无忌",

dsc: "夜不归宿",

},

{

date: "2011-05-04",

name: "杨过",

dsc: "不在宿舍",

},

{

date: "2018-06-08",

name: "小龙女",

dsc: "使用大功率电器",

},

{

date: "2019-05-06",

name: "黄蓉",

dsc: "熄灯还在大声喧哗",

},

{

date: "2019-05-07",

name: "张三丰",

dsc: "聚众赌博",

},

{

date: "2019-05-08",

name: "燕破岳",

dsc: "熄灯还在锻炼身体",

},

{

date: "2019-05-10",

name: "张良",

dsc: "熄灯跑去吓唬别人",

},

{

date: "2019-05-11",

name: "后羿",

dsc: "违规使用大功率电器",

},

{

date: "2019-05-12",

name: "李太白",

dsc: "打麻将喝酒",

},

{

date: "2019-06-13",

name: "法外狂徒张三",

dsc: "夜不归宿",

},

{

date: "2019-05-12",

name: "李太白",

dsc: "打麻将喝酒",

},

{

date: "2019-05-12",

name: "黄药大师",

dsc: "夜不归宿",

},

],

后端的代码

const express = require("express");

const cors = require("cors");

let mysql = require("mysql2");

let bodyParser = require("body-parser");

const app = express();

app.use(

cors({

origin: "\*",

})

);

app.use(bodyParser.json());

//这是连接数据库的

let dbconfig = {

host: "localhost",

user: "root",

password: "root",

port: "3306",

database: "login",

};

let db = mysql.createConnection(dbconfig);

//以下是登录的接口

app.post("/login", (req, res) => {

const reqBody = req.body;

console.log(reqBody);

const account = reqBody.account;

const password = reqBody.password;

db.query(

"select \* from logininfo where account=? and pwd=?",

[account, password],

(err, result) => {

if (err) throw err;

const dbAccount = result[0]?.account; //用了js的可选链就避免了数据库中没有的数据导致我的后端挂掉了

const dbPassword = result[0]?.pwd; //用了js的可选链就避免了数据库中没有的数据导致我的后端挂掉了

if (dbAccount == account && dbPassword == password) {//判断前端传过来的值是否与数据库中的值相等

res.send({

say: "1",

});

} else {

res.send({

say: "请正确输入账号密码",

});

}

}

);

});

//以下是注册接口

app.post("/registered", (req, res) => {

const reqBody = req.body;

console.log(reqBody);

const account = reqBody.account;

const password = reqBody.password;

db.query(

"INSERT INTO logininfo(account,pwd) VALUES (?,?) ",

[account, password],

(err, results) => {

if (err) throw err; //若是有异常就抛出异常

if (results.affectedRows == 1) {

res.send({

say: "1",

});

} else {

res.send({

say: "0",

});

}

}

);

});

app.listen(3000, () => {

console.log("正在监听3000端口");

});

3.3数据库相关信息

数据库logininfo表信息

|  |  |
| --- | --- |
| account | 存储用户的账号 |
| pwd | 存储用户密码 |

1. **页面运行测试**



图 4-1-1 登录界面

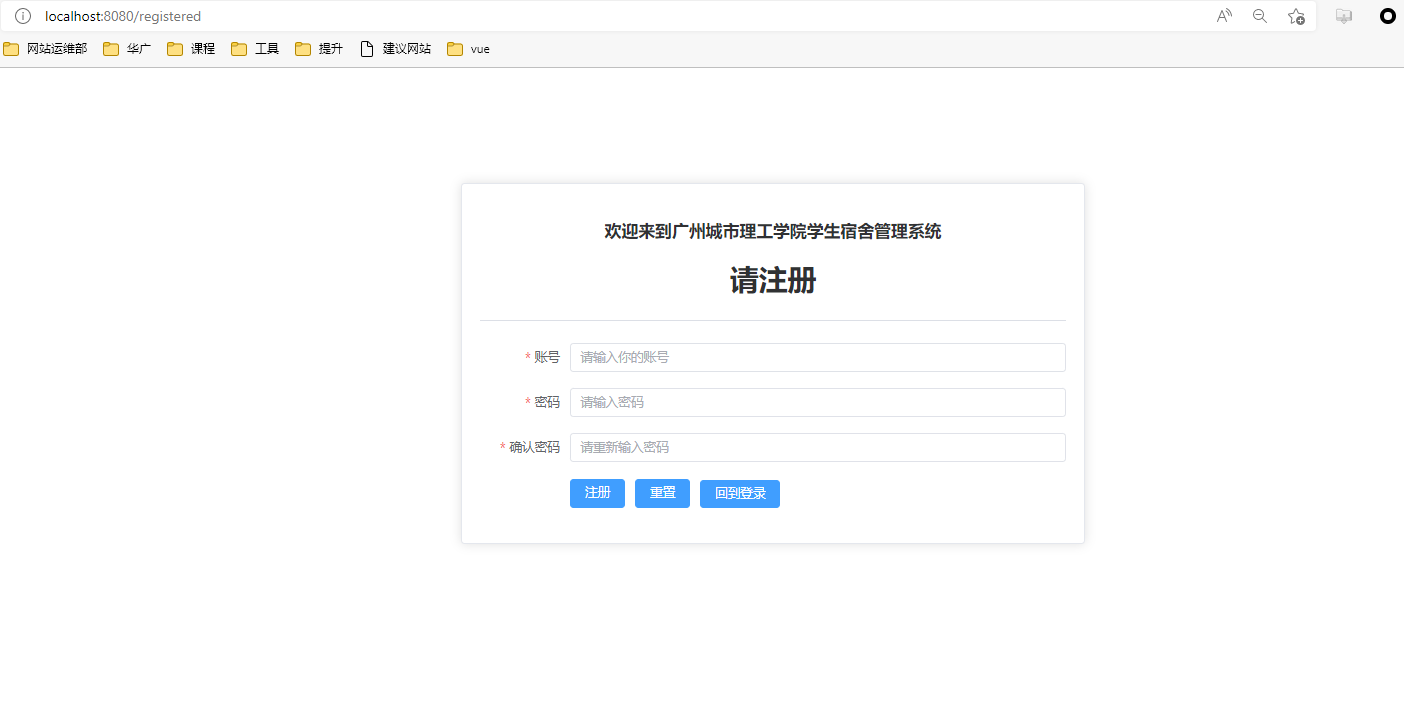


图 4-1-2 注册界面



图 4-1-3 登录成功并默认跳到考勤管理



图 4-1-4 荣誉宿舍



图 4-1-5 维修工单

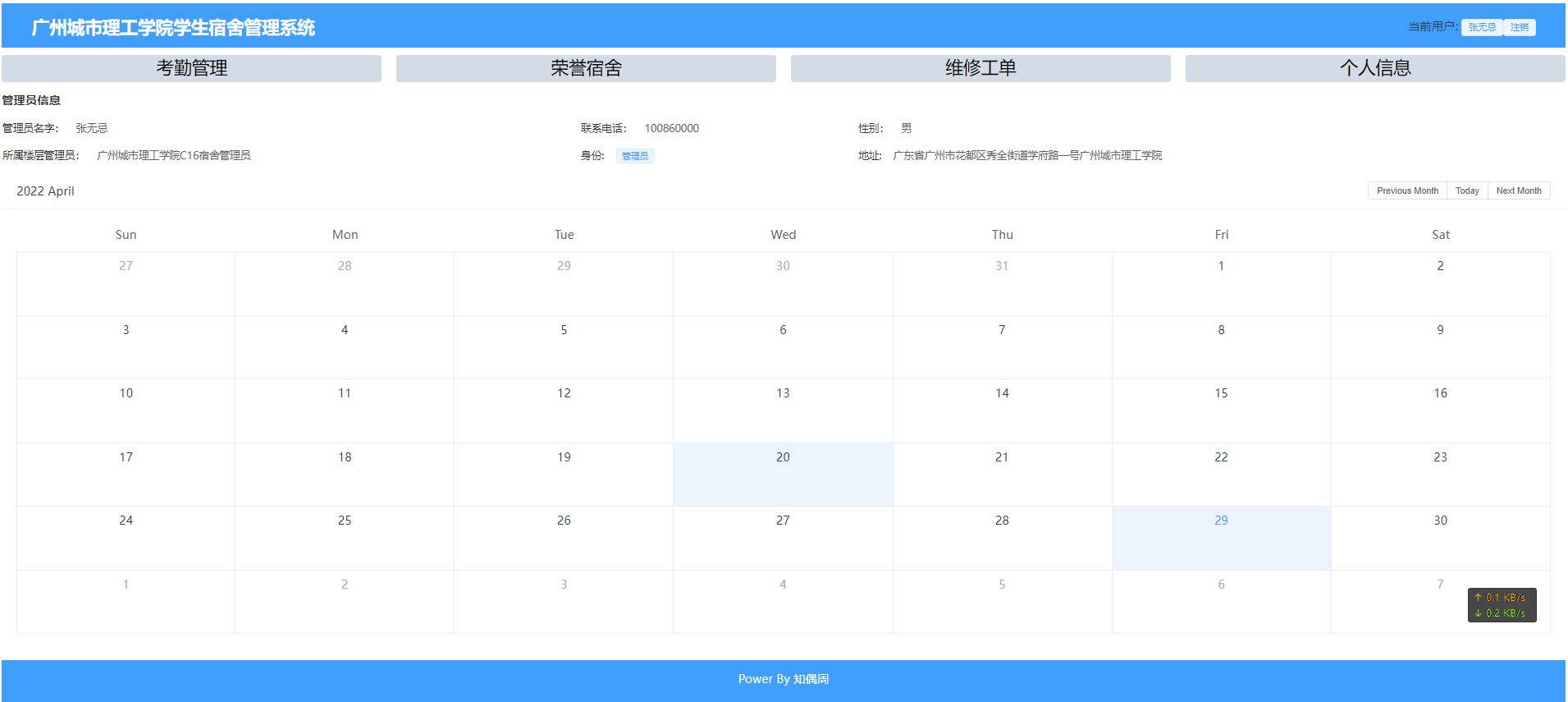


图 4-1-6 个人信息



图4-1-7 当没有输入账号会在前端提示用户输入

1. **设计总结**

在这一次的前端项目实训前，我稍微接触过vue，但是没有很深入去了解，刚刚好趁着这一次的前端项目实训，把vue深入了解了一下，把自己之前没有理解的东西都慢慢地理解了一下。还学习了一个新的框架，就是我在项目实训中用的element-plus框架，也叫饿了么框架。在这一次的实训中，我也屡屡碰壁，会经常出现报错，但在这一次的实训中，我不仅仅学了一个框架，对于我来说，我更是学会了怎么使用框架。因为学会使用框架，就不用自己去写css样式，在这一步就省了很多时间。对于Vue，一开始我的理解仅仅是一个框架，但随着慢慢的了解，我对vue全家桶有了更进一步的了解，像vue-router、vuex等等，一边写项目实训，一边学习vue。更让我拥有成就感的是我自己学习了后端，并写出一个简陋的接口，因为是刚刚接触的，所以这个接口也存在很大的问题，过于简陋。

通过这一次的前端项目实训，我感觉我学到了很多，而且对于我以后的就业也有了一定的帮助，还让我学会了怎么样去耐心解决报错问题。我相信我接下来会更努力学习关于这一方面的知识，提升自己，为以后的工作做好准备。