总成绩



**个性化教育系列**

**创新创业实训**

学 院 信息科学与工程学院

专业班级 软件工程2102

学生学号 2107090411

学生姓名 刘敬超

指导教师 张坤

2023 年 6 月

**成绩评定表**

|  |  |  |
| --- | --- | --- |
| **指标** | **评价内容** | 评价等级  （A、B、C、D） |
| **知识要求** | 知识运用熟练程度 |  |
| **内容**  **要求** | 内容分析是否深刻、透彻；是否详略得当；条理是否清晰；是否能够很好实现目标 |  |
| **结构**  **要求** | 结构是否完整、是否严谨；逻辑性是否体现；层次是否分明； |  |
| **思路创新能力** | 思路是否新颖，是否有独特的视角；创新性有否明显体现；是否能独立运用本门课程的有关理论知识分析和解决问题 |  |
| **规范**  **要求** | 书写、格式、体例是否规范；文字表达是否准确、流畅；是否符合学术道德规范；字数是否符合要求 |  |
| **其他说明** | **项目自己独立完成，严禁抄袭，出现雷同，一律以零分记录。** | |
| **评阅老师：**  **评阅时间：** | | |

**目 录**

一、学习计划 1

二、学习目的 1

三、学习内容 1

四、项目介绍 2

五、实习总结 3

1. **学习计划**

计划学习Vue框架，并使用Vue框架来构建应用程序，包括组件的创建、状态管理和路由控制等方面。学习Vue框架的设计原则，理解Vue框架内部的工作原理。

**二、学习目的**

Vue是一种流行的JavaScript框架，用于构建单页应用程序。在现代的Web开发中，Vue已经变成了必备技能之一。

本实训从Vue学习开始，通过一周的实训，学生利用Vue技术完成简单的网站的搭建和设计。为创新创业项目建设宣传网页做准备。

**三、学习内容**

**一、Vue基础知识：**

1、认识Vue程序的基本结构。

2、学习使用Vue绑定数据。

3、学习使用Vue绑定事件。

4、学习使用Vue为表单元素实现双向绑定：

5、条件渲染与循环渲染。

**二、vue-cli项目搭建**

**三、Vue组件学习**

**四、项目总结**

**第一天**

本次课程笔记主要介绍了前端开发的基本技术和工具，以及Vue.js框架的使用方法。在HTML部分，我们学习了如何创建一级标题、超级链接和图片。在CSS部分，我们学习了如何使用样式表来控制页面的样式，包括如何设置背景颜色和图片大小。在JavaScript部分，我们学习了如何使用Vue.js框架来构建动态交互式页面，包括如何使用data属性、methods属性和事件绑定等。最后，我们通过几个例子来演示了Vue.js的基本用法，包括如何统计按钮点击次数、显示当前日期时间和改变图片和背景颜色等。

**第二天**

学习了三个 Vue.js 的小项目，包括三联滑动门、复选框的操作和颜色拾取器。这些项目涵盖了 Vue.js 常用的指令和特性，适合初学者进行练手。

三联滑动门  
这个项目展示了如何使用 Vue.js 实现三联滑动门，即点击不同的标题，显示对应的内容。其中，使用了 v-show 指令控制元素的显示和隐藏，使用了 :class 绑定 class 样式，使得当前选中的标题有不同的样式。同时，在数据中设置一个 current 变量，用于记录当前选中的标题编号，从而实现对应内容的显示。

复选框的操作  
这个项目展示了如何使用 Vue.js 实现复选框的操作。其中，使用了 v-model 指令实现双向绑定，将备选项的选中情况保存在 selected 数组中。同时，使用 @change 监听全选复选框的变化，实现全选和取消全选功能。

颜色拾取器  
这个项目展示了如何使用 Vue.js 实现颜色拾取器。其中，使用了 input 标签的 type="range" 属性实现滑块控件，使用 v-model.number 指令实现双向绑定。同时，使用计算属性 computed 计算三个单色变量的值，将它们转换为十六进制颜色值，并将其绑定到背景颜色样式上，实现实时预览效果。

**第三天**

学习了Vue.js框架中的组件和CLI。组件是Vue.js中非常重要的概念，可以帮助我们将页面分解成独立的、可复用的部分，使代码更加清晰和易于维护。在组件的定义中，我们可以使用多种方式，包括注册、模板字面量、x-template、render函数、内联模板和单文件组件等。其中，单文件组件是Vue.js中最常用的一种组件定义方式，它将组件的模板、脚本和样式封装在一个.vue文件中，使组件更加独立和可复用。

我们还学习了Vue CLI，它是一个官方提供的脚手架工具，可以帮助我们快速创建Vue.js项目和组件，并提供了丰富的插件和工具链来简化开发过程。使用Vue CLI，我们可以通过命令行或图形化界面创建新项目、管理依赖、开发和调试代码，以及构建和部署应用程序。

最后，我们讲解了如何定义一个Vue.js组件，包括使用Vue.extend(options)创建、将data属性定义为函数、使用template配置组件结构等。这些知识点对于理解和使用Vue.js框架都非常重要，可以帮助我们更好地开发和维护Vue.js应用程序。

**五、实习总结**

1. 使用vue-cli创建项目

Vue版本选择使用vue3, 包管理工具选择yarn， 安装路由和babal， 路由的跳转模式选择history， 然后手动安装pinia和axios。项目创建完成。

1. 设置目录结构

我们的代码都放在src目录下，src目录的结构如下：

├─api 接口请求相关代码

├─assets 静态资源，包括css,字体,图片等

├─components 存放组件

├─router 路由配置文件

├─stores 状态管理

├─styles 全局样式

├─utils 工具方法

└─views 页面

├─App.vue 根组件

└─main.js 入口文件

1. 配置路由

import { createRouter, createWebHistory } from 'vue-router'

import MainView from '../views/Main.vue'

import MessageWallView from "@/views/MessageWallView.vue";

import ImageWallView from "@/views/ImageWallView.vue";

const routes = [

  {

    path: '/',

    name: 'home',

    component: MainView,

    children: [

      {

        path: '/',

        redirect: '/wall'

      },

      {

        path: '/wall',

        name: 'wall',

        component: MessageWallView,

      },

      {

        path: '/img',

        name: 'img',

        component: ImageWallView

      }

    ]

  },

]

const router = createRouter({

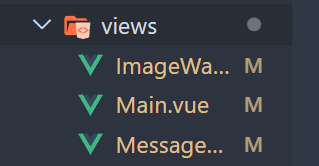
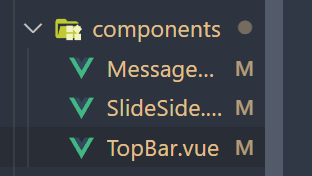
  history: createWebHistory(process.env.BASE\_URL),

  routes

})

export default router

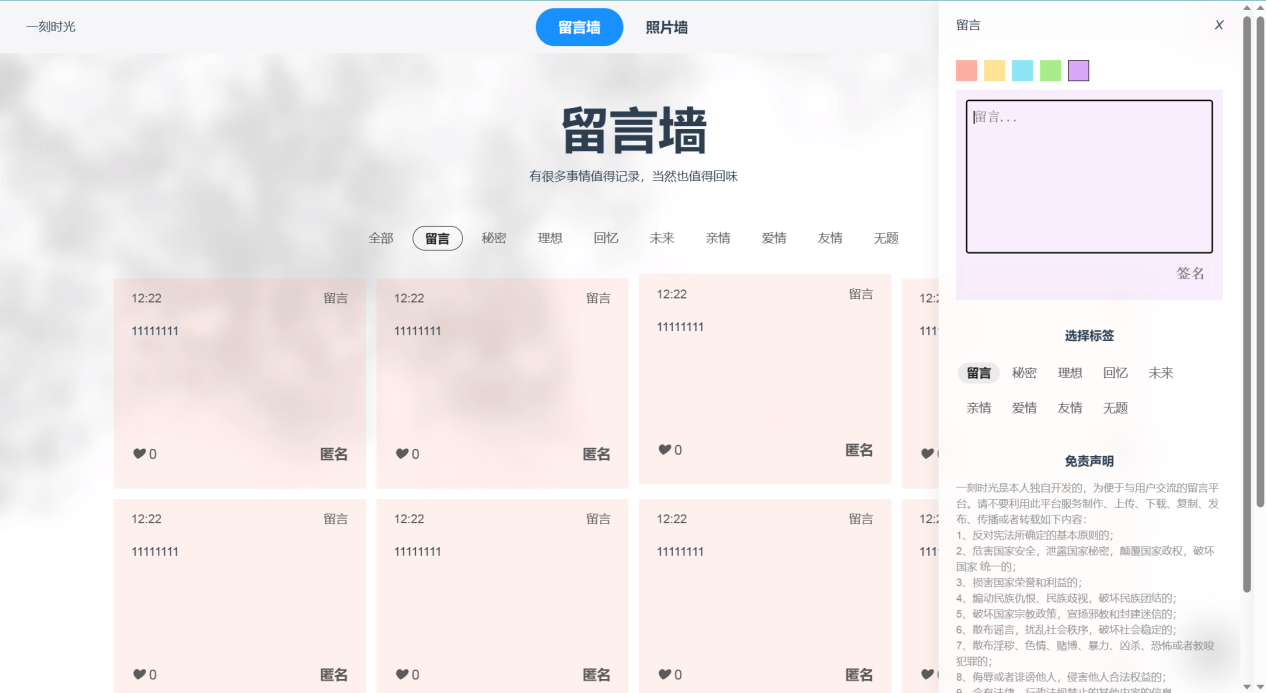
1. 编写页面和组件

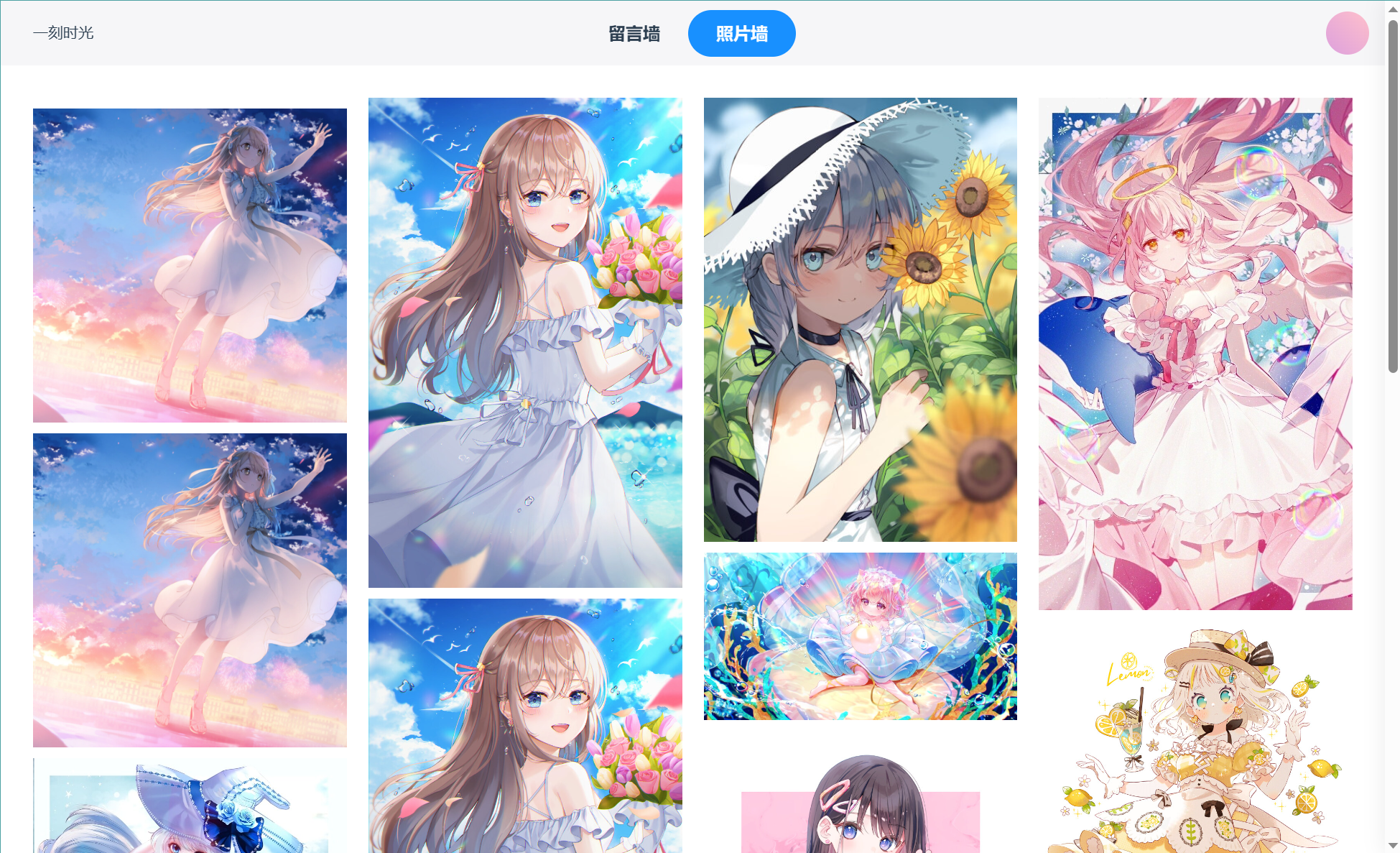


1. 最终页面的效果如下

在根目录输入yarn serve构建页面







项目还没有完成，之后会继续完善。