



南京邮电大学

服务外包项目实训报告

(2023/ 2024 学年 第 1 学期)

题 目： 仿 www.booking.com 网站

专 业 软件工程（嵌入式培养）

学 生 姓 名 李俊凯

班 级 学 号 B20031910

指 导 教 师 沈殷涛/ 成小惠

指 导 单 位 北京中软国际教育科技股份有限公司

南京邮电大学计算机学院

日 期 2023.10.30 - 2023.11.10

仿 www.booking.com 网站

一、概论

1.1. 课题目标:

本次项目实训中,我们选择的课题是“仿 www.booking.com 网站”。在项目中,我主要负责“热门目的地——区域、城市、景点”这个 Vue 界面的代码编写与逻辑交互。在这个过程中,用到以下技能和知识:

- 前端开发框架和技术: Node.js、Vue2.0
- 基于 Vue2.0 的组件库: element-ui 桌面端组件库。

1.2. 任务分析:

我所负责的 Vue.js 组件旨在通过选项卡界面展示不同类别的热门旅游目的地。它提供了一种视觉上吸引人的方式来呈现有关各种目的地的信息,并允许用户轻松切换类别。在代码实现的过程中,主要面临如下任务:

- 任务 1: 该组件提供了一个选项卡界面,用于在不同类别之间切换热门旅游目的地: 区域、城市和周边地标/景点。
- 任务 2: 每个选项卡显示一个目的地网格,其中每个目的地由一个带有链接和可用住宿数量的 `<el-col>` 表示。
- 任务 3: 活动选项卡由 `activeName` 属性控制,点击选项卡会触发 `handleClick` 方法,该方法记录有关点击选项卡的信息。
- 任务 4: 撰写文档,包括实验报告和其他相关文档,记录实验过程、设计决策、实现细节和测试结果。

1.3. 内容描述:

(1) 在任务 1“该组件提供了一个选项卡界面,用于在不同类别之间切换热门旅游目的地: 区域、城市和周边地标/景点。”中,主要包含如下内容:

- 在模板中使用 `<el-tabs>` 组件,通过 `v-model` 绑定 `activeName` 属性,以便跟踪当前选中的选项卡。
- 使用 `<el-tab-pane>` 组件表示每个选项卡,设置 `label` 属性为选项卡的名称,设置 `name` 属性为选项卡的标识符。
- 在每个选项卡下设置目的地信息的网格布局。

(2) 在任务 2“每个选项卡显示一个目的地网格,其中每个目的地由一个带有链接和可用住宿数量的 `<el-col>` 表示。”中,主要包含如下内容:

- 在每个 `<el-tab-pane>` 下使用 `<el-row>` 和 `<el-col>` 组件创建网格布局。
- 使用 `v-for` 指令遍历每个目的地,为每个目的地创建一个 `<el-col>`。
- 在每个 `<el-col>` 中使用 `<el-link>` 组件创建链接,并显示相应的目的地信息。

(3) 在任务 3“活动选项卡由 `activeName` 属性控制,点击选项卡会触发

handleClick 方法，该方法记录有关点击选项卡的信息。”中，主要包含如下内容：

- 在 `<script>` 部分的 `data` 属性中添加 `activeName` 属性，初始值为默认选项卡的标识符。
- 创建 `handleClick` 方法，该方法接收选项卡信息，并在点击时将其输出到控制台。
- 在 `<el-tabs>` 上通过 `@tab-click` 监听选项卡点击事件，触发 `handleClick` 方法。

（4）在任务 4“撰写文档，包括实验报告和其他相关文档，记录实验过程、设计决策、实现细节和测试结果。”中，主要包含如下内容：

- 测试和调试：进行系统级和单元级测试，确保实现在功能和性能方面正常工作。使用调试工具和技术解决可能出现的问题和错误。
- 撰写文档：编写实验报告，记录实验过程、设计决策、实现细节和测试结果。撰写其他相关文档，如用户手册或技术文档，以便其他人能够理解和使用我们的实现。

二、设计与实现

在我主要负责“热门目的地——区域、城市、景点”这个 Vue 界面中，主要基于 Vue2.0 框架和 Element-ui 组件库，结合 HTML，CSS，JavaScript 技术实现。程序的整体架构、主要功能流程、主要数据结构的设计、数据设计以及部分关键代码如下所示：

2.1.程序的整体架构

本界面采用了典型的 Vue 组件结构，分为模板（`template`）、脚本（`script`）、和样式（`style`）三个部分。

（1）模板（`template`）部分：

- 外层容器使用 `<div>` 标签，设置了宽度为 80%，并居中显示。
- 标题部分使用 `<div>` 标签，设置了居左对齐、宽度为 100%、字体大小为 `x-large`、颜色为黑色。
- 采用 `<el-tabs>` 组件创建选项卡界面，通过 `v-model` 绑定 `activeName` 属性，以实现选项卡的切换。
- 三个选项卡分别使用 `<el-tab-pane>` 组件表示，分别为“区域”、“城市”和“周边地标/景点”，每个选项卡下包含一个 `<el-row>`。

（2）脚本（`script`）部分：

- 使用 `data` 方法定义了 `activeName` 属性，初始值为默认选项卡的标识符。
- 定义了 `handleClick` 方法，用于处理选项卡点击事件，将选项卡信息输出到控制台。

（3）样式（`style`）部分：

- 使用 `scoped` 属性确保样式只作用于当前组件。
- 定义了一个名为 `.bg-purple-light` 的样式类，设置了背景颜色为白色。

综上所述，本 Vue 界面程序的整体架构具有如下特点：

- ✓ 整体架构简洁明了，采用了 Vue 的组件化思想。
- ✓ 通过 `<el-tabs>` 和 `<el-tab-pane>` 实现了选项卡的切换。
- ✓ 使用了网格布局，每个选项卡下都有多个 `<el-col>` 用于显示不同目的地的信息。
- ✓ 通过 `activeName` 属性和 `handleClick` 方法实现了对选项卡状态的控制和事件处理。

2.2.主要功能的流程

本 Vue 界面实现了一个简单的热门目的地展示界面，用户可以通过选项卡切换不同类别的目的地，并查看每个目的地的基本信息。这是一个静态展示页面，通过点击事件演示了基本的交互处理。主要功能的流程可以描述如下：

（1）初始化：

- 组件加载时，`activeName` 被初始化为 `'first'`，即默认显示“区域”选项卡。
- 页面上显示“热门目的地”标题和三个选项卡：“区域”、“城市”、“周边地标/景点”。

（2）选项卡切换：

- 用户可以点击选项卡切换不同的热门目的地类别。
- 切换选项卡时，`@tab-click` 事件触发 `handleClick` 方法，输出选项卡信息到控制台。

（3）目的地网格显示：

- 每个选项卡下都有一个网格布局，通过 `<el-row>` 和多个 `<el-col>` 组成。
- 每个 `<el-col>` 表示一个目的地，包含了一个带链接的 `<el-link>` 和一个显示住宿数量的 ``。

（4）数据动态展示：

- 每个目的地的具体信息（名称、住宿数量）通过动态绑定显示，数据来源于模板中的静态代码。
- 这些信息来自后端 API，通过异步请求获取，但在此代码中是静态的。

（5）点击事件处理：

- 点击每个目的地的链接时，可触发页面跳转或其他交互，但在此代码中是通过 `handleClick` 方法输出选项卡信息到控制台。

（6）样式控制：

- 使用 `<style scoped>` 部分对样式进行了控制，确保 `.bg-purple-light` 样式只在当前组件生效，设置了目的地网格的背景颜色。

2.3.主要数据结构的设计

在本界面中，主要的数据结构设计涉及到热门目的地的信息展示，包括目的地的类别、名称和住宿数量。在 Vue 组件中，这些数据通过 Vue 的 data 属性进行定义和管理。在这个设计中，destinations 对象包含了三个属性，分别对应不同选项卡（'first'、'second'、'third'）。每个属性的值是一个数组，包含了该类别下各个目的地的信息，每个目的地信息是一个包含名称（name）和住宿数量（accommodations）的对象。这种设计使得数据的维护和扩展更加方便，可以轻松增加或修改目的地的信息。具体代码如下所示：

```
data() {
  return {
    activeName: 'first',
    destinations: {
      'first': [ // 区域
        { name: '马尔代夫', accommodations: 807 },
        { name: '西双版纳', accommodations: 81 },
        { name: '澳大利亚', accommodations: 233 },
        // ... 其他区域的目的地
      ],
      'second': [ // 城市
        { name: '胡志明市', accommodations: 5607 },
        { name: '台南', accommodations: 583 },
        { name: '高雄', accommodations: 332 },
        // ... 其他城市的目的地
      ],
      'third': [ // 周边地标/景点
        { name: '吉隆坡中央车站', accommodations: 4226 },
        { name: '香港迪士尼乐园', accommodations: 343 },
        { name: '云顶高原', accommodations: 655 },
        // ... 其他周边地标/景点的目的地
      ],
    },
  };
},
methods: {
  handleClick(tab, event) {
```

```

    console.log(tab, event);
  },
}

```

2.4.数据的设计

我所负责的“热门目的地——区域、城市、景点”界面所涉及到数据如：目的地信息等，均以硬编码的方式嵌入在组件中，每个目的地都有名称和住宿数量。

2.5.部分关键的代码

以下是本界面代码的关键部分：

(1) Vue 模板部分：

```

<template>
  <div style="width: 80%;margin: auto">
    <!-- ... -->
    <el-tabs v-model="activeName" type="border-card" @tab-click="handleClick">
      <el-tab-pane label="区域" name="first">
        <el-row>
          <!-- 区域目的地列表 -->
          <el-col :span="4" v-for="(destination, index) in
destinations.first" :key="index">
            <div class="grid-content bg-purple-light">
              <div><el-link>{{ destination.name }}</el-link></div>
              <span>{{ destination.accommodations }}家住宿</span>
            </div>
          </el-col>
        </el-row>
        <!-- ... 其他区域的 el-row -->
      </el-tab-pane>
      <el-tab-pane label="城市" name="second">
        <!-- 城市目的地列表 -->
        <el-row>
          <el-col :span="4" v-for="(destination, index) in
destinations.second" :key="index">
            <div class="grid-content bg-purple-light">
              <div><el-link>{{ destination.name }}</el-link></div>
              <span>{{ destination.accommodations }}家酒店</span>
            </div>

```

```

        </el-col>
      </el-row>
      <!-- ... 其他城市的 el-row -->
    </el-tab-pane>
    <el-tab-pane label="周边地标/景点" name="third">
      <!-- 周边地标/景点目的地列表 -->
      <el-row>
        <el-col :span="4" v-for="(destination, index) in
destinations.third" :key="index">
          <div class="grid-content bg-purple-light">
            <div><el-link>{{ destination.name }}</el-link></div>
            <span>{{ destination.accommodations }}家住宿</span>
          </div>
        </el-col>
      </el-row>
      <!-- ... 其他周边地标/景点的 el-row -->
    </el-tab-pane>
  </el-tabs>
</div>
</template>

```

(2) **Vue 脚本部分**：这部分代码主要是通过 **Vue** 的模板语法和数据绑定，将不同选项卡下的目的地信息动态展示在界面上。每个选项卡下都有一个 **el-row**，包含多个 **el-col**，每个 **el-col** 对应一个目的地的展示。目的地的数据通过 **Vue** 的数据驱动进行绑定，实现了动态的界面展示。关键代码如下：

```

<script>
export default {
  data() {
    return {
      activeName: 'first',
      destinations: {
        'first': [
          { name: '马尔代夫', accommodations: 807 },
          // ... 其他区域的目的地数据
        ],
        'second': [

```

```

        { name: '胡志明市', accommodations: 5607 },
        // ... 其他城市的目的地数据
    ],
    'third': [
        { name: '吉隆坡中央车站', accommodations: 4226 },
        // ... 其他周边地标/景点的目的地数据
    ],
    },
    };
},
methods: {
    handleClick(tab, event) {
        console.log(tab, event);
    },
},
};
</script>

```

（3）Vue 样式部分：这是样式部分的关键代码。在这里，使用了 `scoped` 属性，确保样式仅在当前组件内部生效。样式定义了一个名为 `.bg-purple-light` 的类，将背景颜色设为白色。这个类被应用于目的地信息的容器，以确保它们有相同的背景样式。

```

<style scoped>
    .bg-purple-light {
        background-color: white;
    }
</style>

```

三、系统测试

（1）“热门区域按钮”测试：

热门目的地

区域	城市	周边地标/景点			
马尔代夫	西双版纳	澳大利亚	薄荷岛	富士山	
807家住宿	81家住宿	233家住宿	764住宿	548家住宿	
英格兰	上海	海南	苏梅岛	大理	
807家住宿	2181家住宿	1323家住宿	256家住宿	523家住宿	
台南地区	兰卡威	普吉府	金马伦高原	巴淡岛	
207家住宿	181家住宿	13家住宿	323家住宿	135家住宿	

图 1“热门区域按钮”功能测试

(2) “热门城市按钮”测试:

热门目的地

区域	城市	周边地标/景点		
马尔代夫	西双版纳	澳大利亚	薄荷岛	富士山
807家住宿	81家住宿	233家住宿	764住宿	548家住宿
英格兰	上海	海南	苏梅岛	大理
807家住宿	2181家住宿	1323家住宿	256家住宿	523家住宿
台南地区	兰卡威	普吉府	金马伦高原	巴淡岛
207家住宿	181家住宿	13家住宿	323家住宿	135家住宿

图 2 “热门城市按钮”功能测试

(3) “热门周边\景点按钮”测试:

热门目的地

区域	城市	周边地标/景点		
吉隆坡中央车站	香港迪士尼乐园	云顶高原	四面佛	公主港
4226家住宿	343家住宿	655家住宿	423家住宿	51家住宿
上海迪士尼乐园	浅水湾	第一世界广场	奥兰多迪士尼乐园	双威水上乐园
1321家住宿	223家住宿	398家住宿	424家住宿	521家住宿
马来西亚乐高乐园	台中火车站	台北车站	新关仔角	逢甲夜市
12家住宿	237家住宿	328家住宿	487家住宿	580家住宿

图 3 “热门周边\景点按钮”功能测试

(4) 结果分析:

测试结果良好, 3 个 el-tab-pane 界面的功能均正常实现。

四、评价与结论

在完成这个 Vue 项目的过程中, 我学到了许多有关前端开发的知识, 并积累了宝贵的经验。以下是我对这个项目的一些心得体会:

❖ 项目亮点:

1. 组件化开发:

通过使用 Vue 的组件化开发, 我成功地将页面分解成多个可复用的组件, 如 <el-tabs>、<el-tab-pane>、<el-row>和<el-col>。这不仅提高了代码的可维护性, 而且使得整个项目结构清晰明了。

2. 数据驱动页面:

Vue 的数据绑定机制让页面能够根据数据的变化实时更新, 这使得页面呈现的内容更加灵活。特别是通过使用 v-model 绑定 activeName, 实现了标签页的动态切换。

➤ 需要改进的地方:

1. 数据的硬编码:

项目中的数据(如地区、城市、景点等)被硬编码在模板中, 这样的设计使得代码的灵活性降低。下一步, 我打算将数据设计为可动态传入的形式, 以便更好地应对不同的需求。

2. 命名规范和文档:

在编写代码时, 我注意到样式和类名的命名规范上存在一些不一致。为了提

高团队协作效率，我计划在以后的项目中更加严格地遵循一致的命名规范，并提供清晰的文档。

综上所述，这个项目对我来说是一个重要的学习经历，我不仅加深了对 **Vue** 框架的理解，还学到了许多关于前端开发的最佳实践。通过克服项目中的挑战，我相信我已经变得更加熟练，为将来的项目做好了更充分的准备。在不断迭代和改进中，我期待能够打造出更加优秀的前端项目。

评分内容	评分细则		优秀	良好	中等	差
平时成绩 (20%)	学习态度：遵守实验室规定；（10分）					
	考勤情况（10分）					
设计开发成果和验收答辩 (60%)	程序演示与项目介绍：课题功能丰富，程序运行结果正确；界面友好，设计合理，创新性强。（10分）					
	程序开发与实现能力：工作量饱满；代码质量和运行性能良好，算法正确；（30分）					
	答辩时内容阐述清晰准确；问题回答正确（20分）					
设计报告质量 (20%)	报告文档书写格式规范、排版美观、文字流畅（10分）					
	内容正确详实、结构合理、反映系统设计流程（10分）					
评分等级						
指导教师 签名		日期	2023-11-12			
备注	评分等级有五种：优秀、良好、中等、及格、不及格					