

# 实训报告

## 实验目的

本实训的目的是完成商品管理界面及功能，通过使用HTML、CSS和Vue.js技术实现页面布局和交互，以及进行相关的数据管理操作。

## 实验原理

### 1. HTML (HyperText Markup Language):

HTML是一种标记语言，用于创建网页的结构和内容。在本实验中，HTML用于构建商品管理页面的基本结构，包括表单、表格、按钮等页面元素。

### 2. CSS (Cascading Style Sheets):

CSS是用于定义网页的样式和布局的样式表语言。它使页面外观更加吸引人，包括颜色、字体、边距、宽度等方面的设计。CSS用于美化商品管理页面，使其更具吸引力和可读性。

### 3. Vue.js:

Vue.js是一款用于构建用户界面的JavaScript框架。Vue 3.x版本基于MVVM (Model-View-ViewModel) 模式，其中的主要原理和概念包括：

- Composition API: Vue 3引入了Composition API，这是一种新的API风格，允许更好地组织和重用组件逻辑，提高了代码的可读性。
- 数据驱动: Vue 3将数据驱动视图，当数据改变时，视图会自动更新，反之亦然。Vue 3通过数据绑定机制实现这一特性。
- 组件化: Vue 3将应用程序划分为多个组件，每个组件拥有自己的数据和视图，以提高代码的可维护性和可重用性。组件可以嵌套，形成组件树。
- 双向数据绑定: Vue 3允许双向数据绑定，这意味着数据的变化会自动反映在视图上，视图上的操作也会更新数据。这通过 `v-model` 等指令实现。
- 事件处理: Vue 3提供了事件处理机制，可以在视图上监听用户操作，触发相关的方法，实现交互逻辑。
- 指令: Vue 3提供了各种指令（如 `v-model`、`v-for`、`v-if`）用于页面操作和数据展示，以简化开发过程。

### 4. 数据绑定:

数据绑定是Vue 3的核心特性，它确保数据和视图保持同步。通过在模板中使用 `{{ data }}` 或指令（如 `v-bind` 和 `v-model`）将数据与视图关联，实现了数据的自动更新。

## 5. 事件处理:

Vue 3允许在视图中监听各种事件（如点击、输入、鼠标悬停），并将事件处理函数与这些事件关联。当事件发生时，相关函数被触发，从而实现了页面交互逻辑。

## 6. 条件渲染和列表渲染:

Vue 3提供了条件渲染（`v-if`、`v-else`）和列表渲染（`v-for`）的指令，使开发者能够根据条件来渲染页面元素，或者循环渲染列表数据。

## 7. 表单验证:

在本实验中，表单验证是通过在提交时检查表单字段是否为空来实现的。Vue 3中的方法和数据属性用于验证表单输入，确保不允许空值的字段得到正确的处理。

# 实验内容

本实验主要包括以下内容：

1. 使用HTML和CSS实现商品管理页面的布局和样式设计。
2. 使用Vue3.js构建商品管理页面，实现数据的展示、添加、编辑和删除等功能。
3. 通过事件处理和数据绑定，实现页面的交互逻辑。

# 实验步骤

## 页面布局和样式设计

- 使用HTML构建页面结构，包括表单、表格等元素。
- 使用CSS设计页面样式，包括布局、颜色、边框等。

## 集成Vue3.js并创建Vue实例

- 在HTML文件中引入Vue3.js库。
- 创建Vue实例，定义数据和方法。

## 实现数据展示和交互功能

- 使用Vue数据绑定将数据展示在页面上。
- 实现添加商品、编辑商品、删除商品、清空表单等功能的方法。
- 使用事件处理绑定按钮的点击事件。

## 实验测试和调试

- 打开页面进行功能测试，包括添加、编辑、删除商品等操作，检查页面交互是否正常。
- 根据测试结果进行调试，修复可能存在的问题。

# 实验小结

下图是实验的结果。

# 商品管理

编号：

名称：

价格：

提交

清空

编号	名称	价格	操作
1	康师傅方便面	6	<a href="#">修改</a>   <a href="#">删除</a>
2	百岁山矿泉水	3	<a href="#">修改</a>   <a href="#">删除</a>
3	德芙巧克力	10	<a href="#">修改</a>   <a href="#">删除</a>
4	酱福猪肘子	40	<a href="#">修改</a>   <a href="#">删除</a>

通过本次实验，学生成功独立完成了商品管理界面的开发，掌握了HTML、CSS、Vue.js的基本使用。学生能够理解Vue.js的数据绑定、事件处理机制，并能运用Vue.js构建动态交互的Web页面。此实验也为学生后续学习和实践Vue.js以及Web应用开发奠定了基础。

## 思考题

### 1. 什么是MVVM模式，它的优势是什么？

MVVM模式是一种架构模式，它将应用程序分为三个主要部分：Model、View、和 ViewModel。以下是MVVM的主要组成部分和优势：

- Model（模型）：表示应用程序的数据和业务逻辑。这是应用程序的数据层，负责管理数据和提供操作数据的方法。
- View（视图）：表示用户界面的可视部分。它展示数据，接收用户输入，并显示用户界面。

- **ViewModel（视图模型）**：充当Model和View之间的中介，它维护应用程序的状态，并负责处理View的展示逻辑。ViewModel通过数据绑定将数据从Model传递给View，实现了双向数据绑定，确保View和Model保持同步。

MVVM的优势包括：

- **分离关注点**：MVVM模式将应用程序的不同关注点分离，使代码更易于维护和测试。Model负责数据处理，View负责UI展示，ViewModel负责逻辑和数据绑定。
- **双向数据绑定**：ViewModel和View之间的双向数据绑定使数据的自动同步更容易实现，减少了手动DOM操作。
- **可测试性**：由于关注点分离，ViewModel可以更容易地进行单元测试，而不需要UI交互。
- **适用于大型应用**：MVVM适用于大型和复杂的应用程序，因为它使代码结构更清晰，组织更有条理。

## 2. 在选择商品时如何加入全选和反选功能？

全选和反选功能通常可以通过以下步骤来实现：

- 首先，在Vue的数据模型中添加一个属性，例如 `selectAll`，用于跟踪是否全选。
- 在页面中添加一个全选的复选框，并将其与 `selectAll` 进行数据绑定。
- 在点击全选复选框时，触发一个方法，例如 `selectAllItems`，在该方法内，遍历商品列表，将每个商品的选中状态与 `selectAll` 保持一致。
- 在每个商品条目的前面添加一个复选框，并将其与商品的选中状态进行数据绑定。
- 在点击每个商品前面的复选框时，触发一个方法，例如 `selectItem`，在该方法内，将商品的选中状态更新，并检查是否所有商品都被选中，如果是，则更新 `selectAll` 为 `true`，否则更新为 `false`。

这样，用户可以通过点击全选复选框来全选或反选所有商品，同时也可以点击单个商品的复选框来选择或取消选择单个商品。

## 3. 提交商品时如何检验商品信息不可以为空？

检验商品信息不可以为空可以通过以下方式来实现：

- 在提交商品时，添加一个验证步骤，确保所有必填字段（例如，名称和价格）都有值。
- 在Vue的方法中，创建一个检验函数，例如 `validateForm`。
- 在 `validateForm` 中，检查每个字段是否为空，如果有任何字段为空，可以显示错误消息或者阻止提交，并提示用户填写必要信息。
- 如果所有字段都有值，继续执行提交操作。
- 在页面中，可以使用条件渲染来显示错误消息，以提醒用户必填字段为空。

例如：

```
1 methods: {  
2   validateForm: function () {  
3     if (!this.name || !this.price) {  
4       // Show an error message or prevent submission  
5       // You can also set a flag to display error messages in the view  
6       this.errorMessage = 'Please fill in all required fields.';  
7     } else {  
8       // Continue with submission  
9       this.errorMessage = ''; // Clear any previous error message  
10      this.handle(); // Call your submit method  
11    }  
12  }  
13 }  
14
```

在上述示例中，如果名称或价格为空，会设置一个错误消息，否则将继续执行提交操作。