

4. 品項新增

功能說明:

1. 點擊新增會顯示出下方產品名稱與產品圖片的表單
2. 點擊確認會新增該筆資料並渲染至上方表格
3. 點擊編輯會將該筆資料帶入下方產品名稱與產品圖片並電擊確認後會更新資料
4. 勾選銷售狀況會改變該筆的背景色
5. 下方表單在需要時才會出現

新增

| 標題 | 圖片 | 銷售狀況 | 編輯 |
|----------|--|-------------------------------------|---------------|
| 豪華版蛋糕001 |  | <input checked="" type="checkbox"/> | <div>編輯</div> |
| 豪華版蛋糕002 |  | <input type="checkbox"/> | <div>編輯</div> |

產品名稱

產品圖片

確認

1. 基本架構

綁定產品名稱與產品圖片並將圖片顯示出來

```

<script>
  const App = {
    data() {
      return {
        temp: {
          name: '',
          imageUrl: ''
        }
      }
    },
  }
  Vue.createApp(App).mount('#app');
</script>

```

```

<div class="container" id="app">
  <div class="row">
    <div class="col-md-8 offset-2 card mt-3">
      <div class="form-group">
        <div class="mb-3">
          <label for="">產品名稱</label>
          <input type="text" class="form-control" name="" id="" v-model="temp.name">
        </div>
        <div class="mb-3">
          
          <hr>
          <label for="">產品圖片</label>
          <input type="text" class="form-control" name="" id="" v-model="temp.imageUrl">
        </div>
        <button class="btn btn-primary">確認</button>
      </div>
    </div>
  </div>
</div>

```

產品名稱



產品圖片

Elements Console Vue >>

Find components...

<Root> 13 ms

<Root> Filter state...

▼ data

▼ temp: Object

- imageUrl: "img/car01.png"
- name: "豪華汽車001"

2. 設定預設資料

新增預設資料products, 在利用created生命週期將資料載入

```
<script>
  const products = [
    {
      id: '1',
      imageUrl: 'img/cake01.jpg',
      name: '豪華版蛋糕001',
      onStock: false,
    },
    {
      id: '2',
      imageUrl: 'img/cake02.jpg',
      name: '豪華版蛋糕002',
      onStock: false,
    }
  ]

  const App = {
    data() {
      return {
        products: [],
        temp: {
          name: '',
          imageUrl: ''
        }
      }
    },
    created() {
      this.products = products;
    },
  }

  Vue.createApp(App).mount('#app');
</script>
```

將this.products 渲染至table

```

<table class="table">
  <tr>
    <th>標題</th>
    <th>圖片</th>
    <th>銷售狀況</th>
    <th>編輯</th>
  </tr>
  <tbody>
    <tr v-for="item in products" :key="item.id" >
      <td width="20%">{{ item.name }}</td>
      <td width="40%">
        
      </td>
      <td width="15%">
        <input type="checkbox" v-model="item.onStock">
      </td>
      <td width="15%">
        <button class="btn btn-outline-success">編輯</button>
      </td>
    </tr>
  </tbody>
</table>

```

| 標題 | 圖片 | 銷售狀況 | 編輯 |
|----------|---|--------------------------|---------------------|
| 豪華版蛋糕001 |  | <input type="checkbox"/> | <button>編輯</button> |
| 豪華版蛋糕002 |  | <input type="checkbox"/> | <button>編輯</button> |

產品名稱

產品圖片

確認

3. 點擊確認按鈕新增資料

以Date().getTime()時間標籤取得唯一值

```

methods: {
  confirmEdit() {
    this.temp.id = new Date().getTime();
    this.temp.onStock = false;
    this.products.push(this.temp);
    this.temp = {};
  },
},




```

監聽確認按鈕

```


<div class="form-group">
  <div class="mb-3">
    <label for="">產品名稱</label>
    <input type="text" class="form-control" name="" id="" v-model="temp.name">
  </div>
  <div class="mb-3">
    
    <hr>
    <label for="">產品圖片</label>
    <input type="text" class="form-control" name="" id="" v-model="temp.imageUrl">
  </div>
  <button class="btn btn-primary" @click='confirmEdit'>確認</button>
</div>

```

| 標題 | 圖片 | 銷售狀況 | 編輯 |
|----------|---|--------------------------|--------------------|
| 豪華版蛋糕001 |  | <input type="checkbox"/> | 編輯 |
| 豪華版蛋糕002 |  | <input type="checkbox"/> | 編輯 |
| 豪華版蛋糕003 |  | <input type="checkbox"/> | 編輯 |

產品名稱

豪華版蛋糕004



產品圖片

img/cake04.jpg

[確認](#)

4. 點擊編輯按鈕將該筆資料帶入表單編輯

```

methods: {
  confirmEdit() {
    this.temp.id = new Date().getTime();
    this.temp.onStock = false;
    this.products.push(this.temp);
    this.temp = {};
  },
  editItem(item) {
    this.temp = item;
  }
},
created() {
  this.products = products;
},

```



編輯功能需要傳遞該筆資料

```

<tbody>
  <tr v-for="item in products" :key="item.id" >
    <td width="20%">{{ item.name }}</td>
    <td width="40%">
      
    </td>
    <td width="15%">
      <input type="checkbox" v-model="item.onStock">
    </td>
    <td width="15%">
      <button class="btn btn-outline-success" @click="editItem(item)">編輯</button>
    </td>
  </tr>
</tbody>


```

但編輯時會發現上方資料也會更著一起變動

| 標題 | 圖片 | 銷售狀況 | 編輯 |
|----------|---|--------------------------|--------------------|
| 豪華版蛋糕001 |  | <input type="checkbox"/> | 編輯 |
| 豪華版蛋糕eee |  | <input type="checkbox"/> | 編輯 |

產品名稱

豪華版蛋糕eee



產品圖片



img/cake02.jpg

[確認](#)

修改綁定資料寫法


```
methods: {
  confirmEdit() {
    this.temp.id = new Date().getTime();
    this.temp.onStock = false;
    this.products.push(this.temp);
    this.temp = {};
  },
  editItem(item) {
    this.temp = {...item};
  }
},
created() {
  this.products = products;
},
```


編輯時會發現上方資料不會更著一起變動

| 標題 | 圖片 | 銷售狀況 | 編輯 |
|----------|---|--------------------------|--------------------|
| 豪華版蛋糕001 |  | <input type="checkbox"/> | 編輯 |
| 豪華版蛋糕002 |  | <input type="checkbox"/> | 編輯 |

產品名稱

豪華版蛋糕XXX



產品圖片

img/cake01.jpg

[確認](#)

5. 更改編輯判斷方式

temp.id不存在表為新增資料，直接新增至products。

temp.id存在表示為更新資料，利用forEach找到該筆資料並更新。

```

confirmEdit() {
  if (!this.temp.id) {
    this.temp.id = new Date().getTime();
    this.temp.onStock = false;
    console.log(this.temp);
    this.products.push(this.temp);
    this.temp = {};
  } else {
    this.products.forEach((item, i) => {
      if (item.id === this.temp.id) {
        this.products[i] = this.temp;
      }
    });
    this.temp = {};
  }
},

```

6. 利用銷售狀態動態顯示class

| 標題 | 圖片 | 銷售狀況 | 編輯 |
|----------|---|-------------------------------------|---------------------|
| 豪華版蛋糕001 |  | <input type="checkbox"/> | <button>編輯</button> |
| 豪華版蛋糕002 |  | <input checked="" type="checkbox"/> | <button>編輯</button> |

產品名稱

產品圖片

確認

```

<tbody>
  <tr v-for="item in products" :key="item.id" :class="{ 'table-info' : item.onStock}">
    <td width="20%">{{ item.name }}</td>
    <td width="40%">
      
    </td>
    <td width="15%">
      <input type="checkbox" v-model="item.onStock">
    </td>
    <td width="15%">
      <button class="btn btn-outline-success" @click="editItem(item)">編輯</button>
    </td>
  </tr>
</tbody>

```

7. 新增按鈕來定表單是否出現

利用isNew來決定表單是否出現。

編輯狀態時表單也必須出現，因此使用v-if="isNew || temp.id"。

```

<div class="container" id="app">
  <div class="row">
    <div class="col-md-8 offset-2 card mt-3">
      <div class="text-right">
        <button class="btn btn-success my-3" v-on:click="addItem">新增</button>
      </div>
      <table class="table">...
      </table>
      <div class="form-group" v-if="isNew || temp.id">
        <div class="mb-3">
          <label for="">產品名稱</label>
          <input type="text" class="form-control" name="" id="" v-model="temp.name">
        </div>
        <div class="mb-3">
          
          <hr>
          <label for="">產品圖片</label>
          <input type="text" class="form-control" name="" id="" v-model="temp.imageUrl">
        </div>
        <button class="btn btn-primary" @click='confirmEdit'>確認</button>
      </div>
    </div>
  </div>
</div>

```

```
methods: {  
  addItem() {  
    this.isNew = true;  
    this.temp = {}  
  },  
  confirmEdit() {  
    if (!this.temp.id) {  
      this.temp.id = new Date().getTime();  
      this.temp.onStock = false;  
      console.log(this.temp);  
      this.products.push(this.temp);  
      this.temp = {};  
    } else {  
      this.products.forEach((item, i) => {  
        if (item.id === this.temp.id) {  
          this.products[i] = this.temp;  
        }  
      });  
      this.temp = {};  
    }  
    this.isNew = false;  
  },  
  editItem(item) {  
    this.temp = {...item};  
  }  
},
```