

DynamicComponents

"meta component"

<https://vuejs.org/api/built-in-special-elements.html#component>

Що робити, якщо потрібно мати різні відображення списку продуктів?

Mode **rows**

Менеджер Продуктів

Ноутбук Dell XPS 13
Ціна: 3499.99 грн

Смартфон iPhone 13 Pro
Ціна: 1299.99 грн

Телевізор Samsung QLED Q80A
Ціна: 1699.99 грн


Кавоварка Nespresso VertuoPlus
Ціна: 179.99 грн

Фотокамера Sony Alpha a7 III
Ціна: 1999.99 грн


Додати новий товар

Mode **cards**


Менеджер Продуктів




Ноутбук Dell XPS 13
Ціна: 3499.99 грн




Смартфон iPhone 13 Pro
Ціна: 1299.99 грн



Телевізор Samsung QLED Q80A
Ціна: 1699.99 грн



Кавоварка Nespresso VertuoPlus
Ціна: 179.99 грн



Фотокамера Sony Alpha a7 III
Ціна: 1999.99 грн

Додати новий товар

Mode **table**

Менеджер Продуктів

Title	Price	Image
Ноутбук Dell XPS 13	3499.99 грн	
Смартфон iPhone 13 Pro	1299.99 грн	
Телевізор Samsung QLED Q80A	1699.99 грн	
Кавоварка Nespresso VertuoPlus	179.99 грн	
Фотокамера Sony Alpha a7 III	1999.99 грн	


Додати новий товар

Що робити, якщо потрібно мати різні відображення списку продуктів? => Можна використати дочріні компоненти


Mode **rows**

Менеджер Продуктів


Ноутбук Dell XPS 13
Ціна: 3499.99 грн




Смартфон iPhone 13 Pro
Ціна: 1299.99 грн




Телевізор Samsung QLED Q80A
Ціна: 1699.99 грн



Кавоварка Nespresso VertuoPlus
Ціна: 179.99 грн




Фотокамера Sony Alpha a7 III
Ціна: 1999.99 грн




Додати новий товар

Mode **cards**


Менеджер Продуктів




Ноутбук Dell XPS 13
Ціна: 3499.99 грн




Смартфон iPhone 13 Pro
Ціна: 1299.99 грн



Телевізор Samsung QLED Q80A
Ціна: 1699.99 грн



Кавоварка Nespresso VertuoPlus
Ціна: 179.99 грн



Фотокамера Sony Alpha a7 III
Ціна: 1999.99 грн

Додати новий товар

Mode **table**

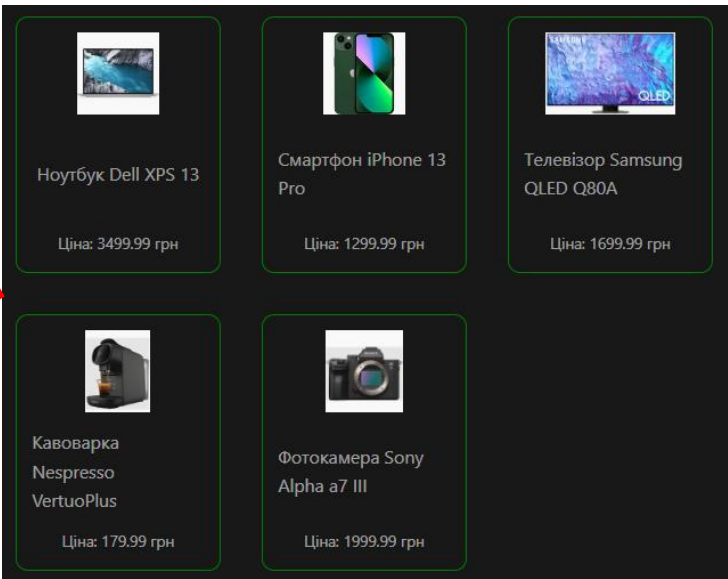
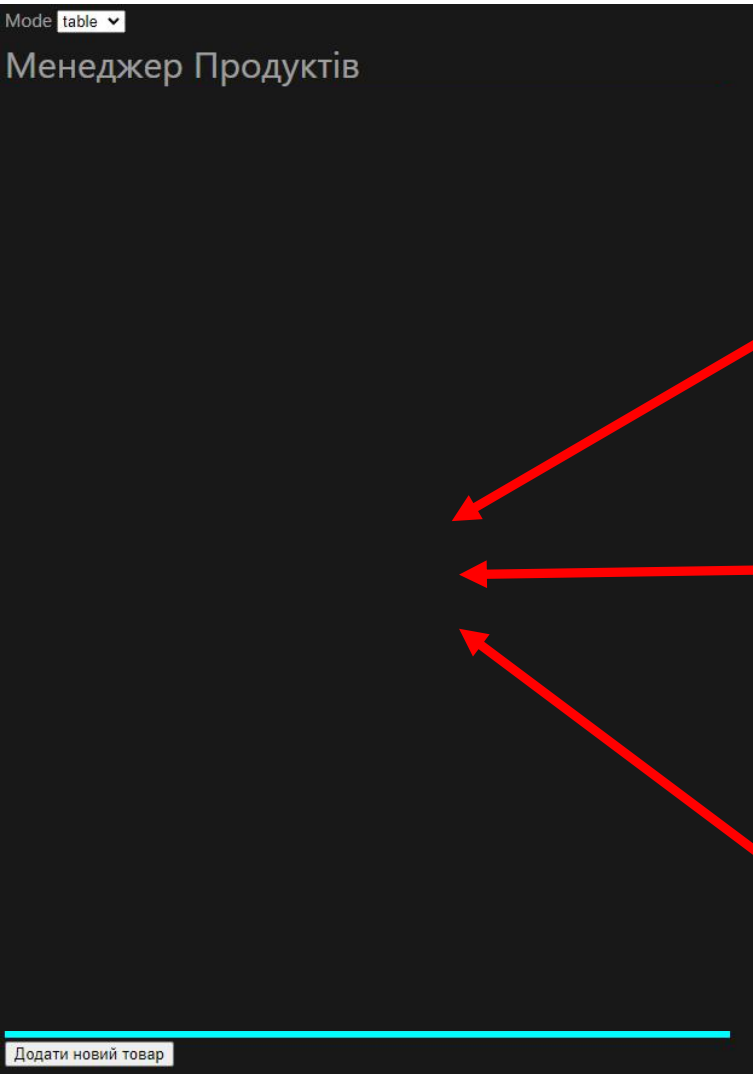
Менеджер Продуктів

Title	Price	Image
Ноутбук Dell XPS 13	3499.99 грн	
Смартфон iPhone 13 Pro	1299.99 грн	
Телевізор Samsung QLED Q80A	1699.99 грн	
Кавоварка Nespresso VertuoPlus	179.99 грн	
Фотокамера Sony Alpha a7 III	1999.99 грн	

Додати новий товар

Що робити, якщо деякий фрагмент батьківського компоненту у залежності від умови замінюється на один з дочірніх?

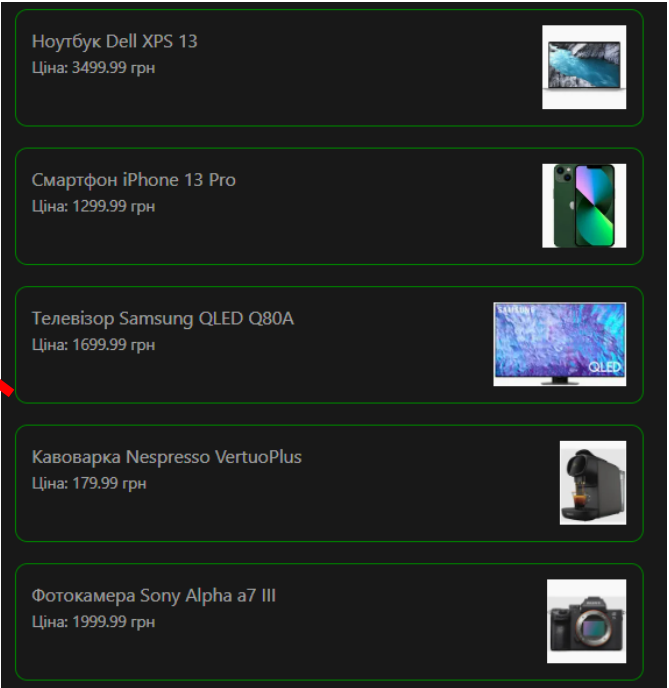
ProductsManager.vue



CardsList.vue

TableRows.vue

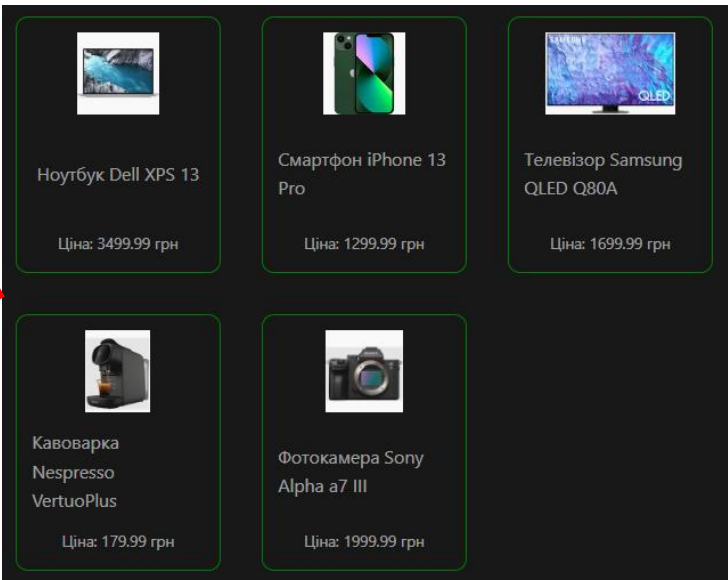
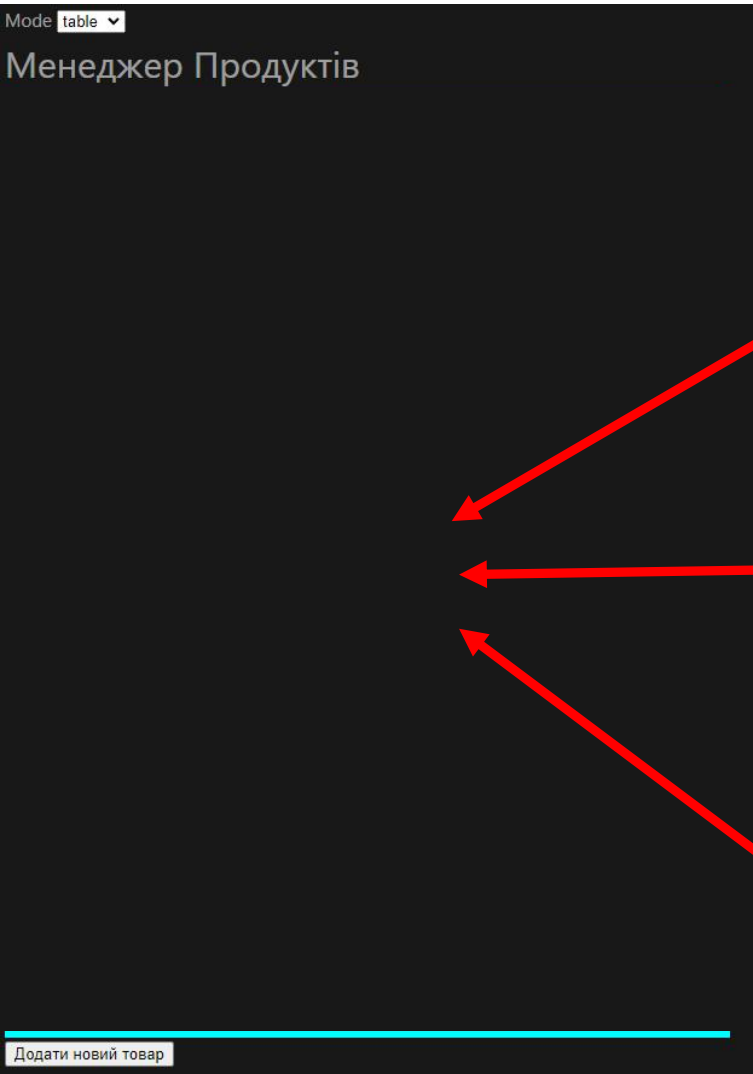
Title	Price	Image
Ноутбук Dell XPS 13	3499.99 грн	
Смартфон iPhone 13 Pro	1299.99 грн	
Телевізор Samsung QLED Q80A	1699.99 грн	
Кавоварка Nespresso VertuoPlus	179.99 грн	
Фотокамера Sony Alpha a7 III	1999.99 грн	



RowsList.vue

Що робити, якщо деякий фрагмент батьківського компоненту у залежності від умови замінюється на один з дочірніх?

ProductsManager.vue

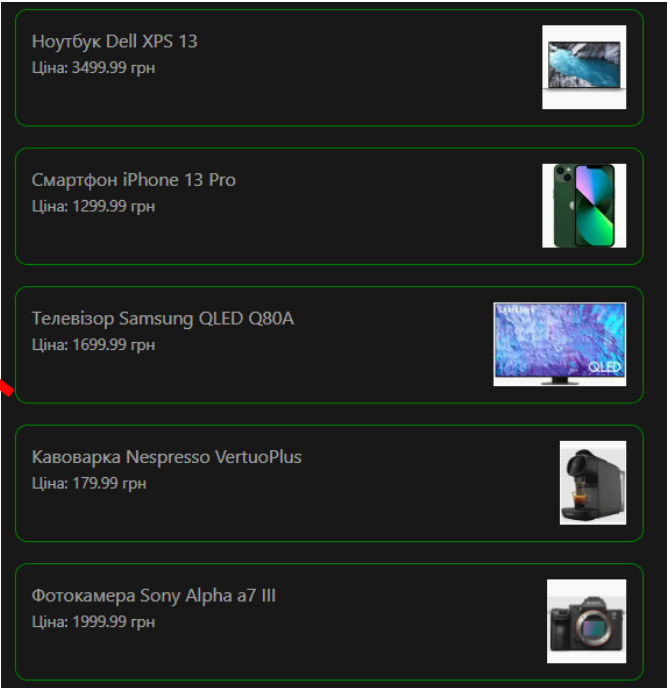


CardsList.vue

Можна використати умовне відображення v-if

TableRows.vue

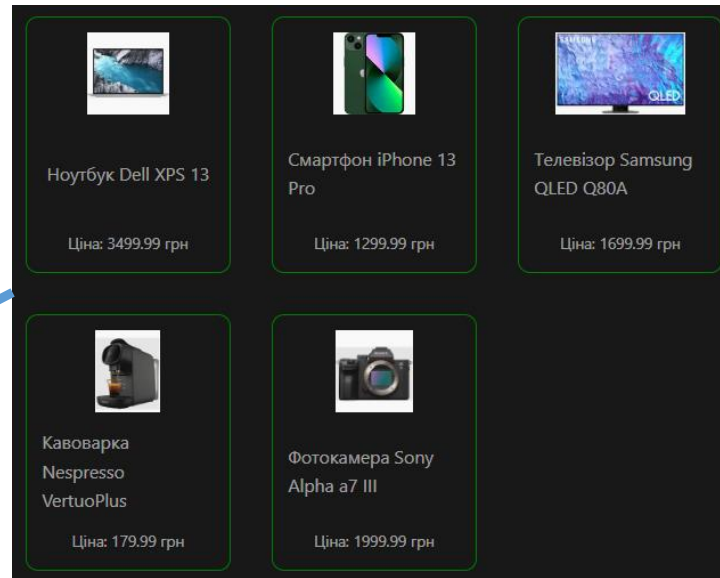
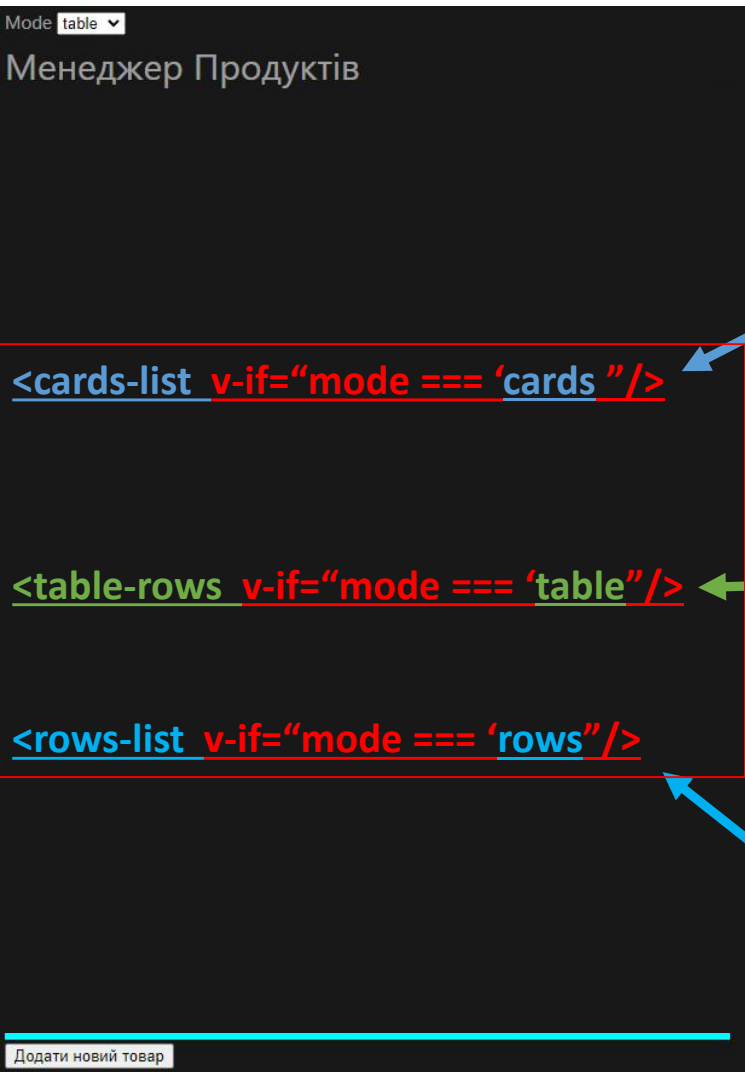
Title	Price	Image
Ноутбук Dell XPS 13	3499.99 грн	
Смартфон iPhone 13 Pro	1299.99 грн	
Телевізор Samsung QLED Q80A	1699.99 грн	
Кавоварка Nespresso VertuoPlus	179.99 грн	
Фотокамера Sony Alpha a7 III	1999.99 грн	



RowsList.vue

Що робити, якщо деякий фрагмент батьківського компоненту у залежності від умови замінюється на один з дочірніх?

ProductsManager.vue

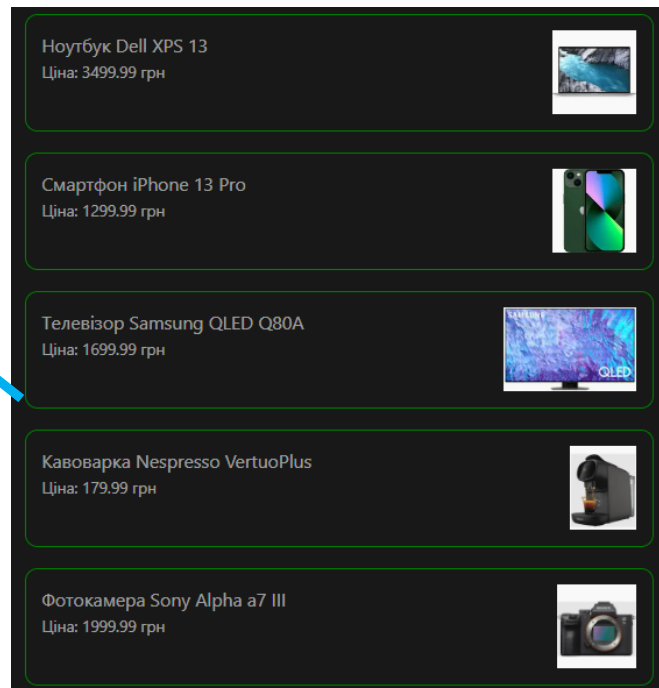


CardsList.vue

Можна використати
умовне відображення
v-if

TableRows.vue

Title	Price	Image
Ноутбук Dell XPS 13	3499.99 грн	
Смартфон iPhone 13 Pro	1299.99 грн	
Телевізор Samsung QLED Q80A	1699.99 грн	
Кавоварка Nespresso VertuoPlus	179.99 грн	
Фотокамера Sony Alpha a7 III	1999.99 грн	



RowsList.vue

Можна використати "meta component" **<component>** для динамічного генерування потрібного компоненту

<component :is=" string | Component" />

Якщо тип значення **is** :

- string – то значенням є HTML tag або ім'я компонента при реєстрації
- Component – то значенням є імпортований дочірній компонент

```
<script>
import Foo from './Foo.vue'
import Bar from './Bar.vue'

export default {
  components: { Foo, Bar },
  data() {
    return {
      view: 'Foo'
    }
  }
}
</script>

<template>
  <component :is="view" />
</template>
```



<https://vuejs.org/api/built-in-special-elements.html#component>

Що робити, якщо деякий фрагмент батьківського компоненту у залежності від умови замінюється на один з дочірніх?

ProductsManager.vue

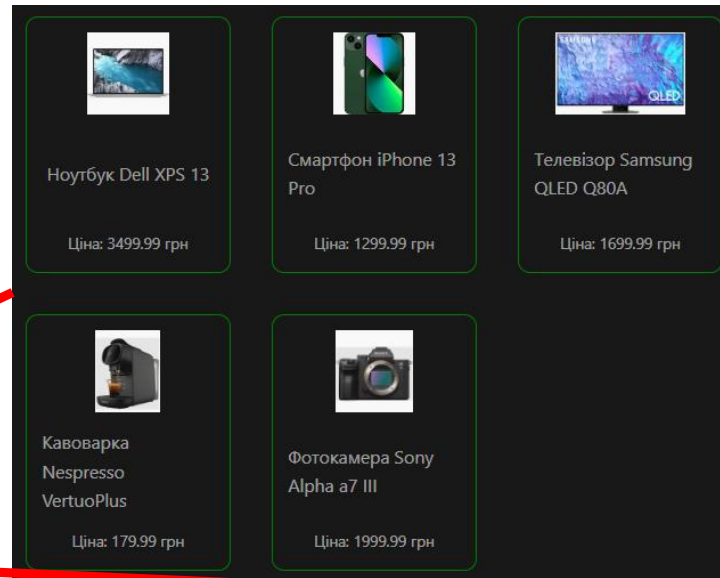
Mode **table** ▼

Менеджер Продуктів

```
<component
  :is="currentComponent"
  :products="products"
/>

const currentComponent = computed(()
=> {
  switch (props.mode) {
    case 'cards': return CardsList
    case 'rows': return RowsList
    case 'table': return TableRows
    default: return CardsList
  }
})
```

Додати новий товар

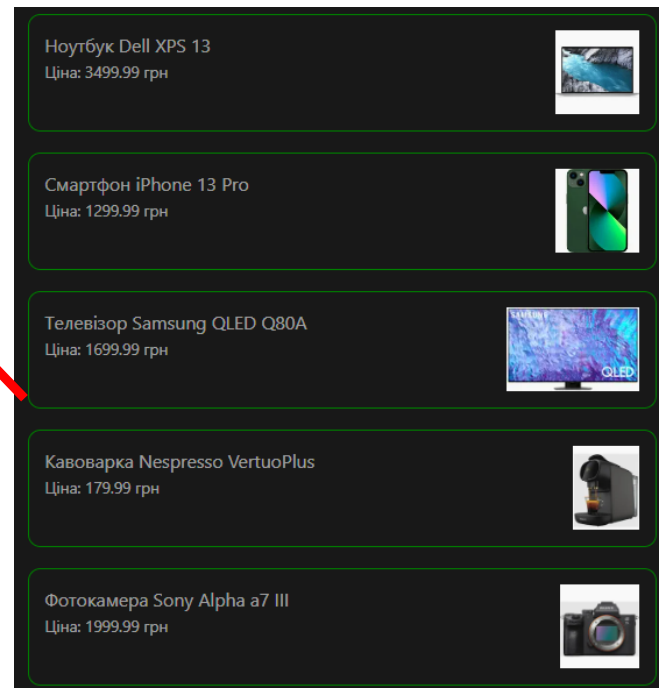


CardsList.vue

Можна використати умовне відображення v-if

TableRows.vue

Title	Price	Image
Ноутбук Dell XPS 13	3499.99 грн	
Смартфон iPhone 13 Pro	1299.99 грн	
Телевізор Samsung QLED Q80A	1699.99 грн	
Кавоварка Nespresso VertuoPlus	179.99 грн	
Фотокамера Sony Alpha a7 III	1999.99 грн	



RowsList.vue


```

<!-- ProductsManager.vue -->
<template>
  <div>
    <h1>{{ title }}</h1>
    <component :is="currentComponent"
      :products="products" />
    <button @click="addNewProduct">Додати новий
      товар</button>
  </div>
</template>

<script setup>
import { computed } from 'vue'
import CardsList from './CardsList/CardsList.vue'
import RowsList from './RowsList/RowsList.vue'
import TableRows from './TableRows/TableRows.vue'

const title = 'Менеджер Продуктів'
const props = defineProps({ ...

const currentComponent = computed(() => {
  switch (props.mode) {
    case 'cards': return CardsList
    case 'rows': return RowsList
    case 'table': return TableRows
    default: return CardsList
  }
})

```

```

<!-- CardsList.vue -->
<template>
  <div class="cards-list">
    <ProductCard v-for="product in
      products" :key="product.id"
      :product="product" />
  </div>
</template>

<script setup>
import ProductCard from './ProductCard.
vue'
defineProps(['products'])
</script>

```

```

<!-- RowsList.vue -->
<template>
  <div class="rows-list">
    <ProductRow v-for="product in
      products" :key="product.id"
      :product="product" />
  </div>
</template>

<script setup>
import ProductRow from './ProductRow.
vue'
defineProps(['products'])
</script>

```

```

<!-- TableRows.vue -->
<template>
  <table class="table-rows">
    <thead>
      <tr>
        <th>Title</th>
        <th>Price</th>
        <th>Image</th>
      </tr>
    </thead>
    <tbody>
      <TableProductRow v-for="product
        in products" :key="product.id"
        :product="product" />
    </tbody>
  </table>
</template>

<script setup>
import TableProductRow from './
TableProductRow.vue'
defineProps(['products'])
</script>

```