

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Hiểu và áp dụng các khái niệm về component trong Vue, bao gồm khai báo, sử dụng props, emit, và các kỹ thuật nâng cao như slots và project/inject.
- ✓ Phát triển khả năng xây dựng các component tương tác và sử dụng hiệu quả Composition API.

NỘI DUNG

PHẦN I:

Bài 1 (2 điểm): Tạo và sử dụng một component đơn giản với Props

Yêu cầu:

- Tạo một component BookCard.vue nhận các props: id, title, author, price, và category.
- Hiển thị thông tin sách trong giao diện và sử dụng Bootstrap để định dạng.
- Sử dụng component này trong App.vue.

Hướng dẫn:

Bước 1: Tạo 1 file mới trong thư mục **Labs/Lab-5/Bai1.vue**.

```
<template>
<div class="card mb-3">
  <div class="card-body">
    <h5 class="card-title">{{ title }}</h5>
    <h6 class="card-subtitle mb-2 text-muted">Tác giả: {{ author }}</h6>
    <p class="card-text">Giá: {{ price }} VND</p>
    <p class="card-text">Thể loại: {{ category }}</p>
  </div>
</div>
</template>
<script setup>
import { defineProps } from "vue";
const props = defineProps({
  id: String,
  title: String,
  author: String,
  price: Number,
  category: String,
});
</script>
```

Bước 2: Gọi component Labs/Lab-5/Bai1.vue ở App.vue

```
<!-- App.vue -->

<template>

<div class="container mt-4">

<Bai1 id="1" title="Cuốn sách 1" author="Tác giả A" :price="150000" category="Fiction" />

</div>

</template>

<script setup>

import Bai1 from "../Labs/Lab-5/Bai1.vue";

</script>
```

Bài 2: Tạo component và sử dụng emit để gửi sự kiện
Yêu cầu:

1. Tái sử dụng code bài 1, tạo một component **BookForm.vue** cho phép người dùng nhập thông tin sách với các trường: title, author, price, và category.
2. Khi người dùng nhấn nút “Thêm sách”, component sẽ emit một sự kiện add-book với dữ liệu sách mới.
3. Sử dụng component này trong App.vue và lắng nghe sự kiện để hiển thị thông tin sách.

Hướng dẫn:

1. Tạo 1 file mới trong thư mục **Labs/Lab-5/Bai2.vue**.

```
<template>
<form @submit.prevent="submitForm" class="mb-4">
  <div class="mb-3">
    <input v-model="title" type="text" class="form-control" placeholder="Tiêu đề" />
  </div>
  <div class="mb-3">
    <input v-model="author" type="text" class="form-control" placeholder="Tác giả" />
  </div>
  <div class="mb-3">
    <input v-model="price" type="number" class="form-control" placeholder="Giá" />
  </div>
  <div class="mb-3">
    <input v-model="category" type="text" class="form-control" placeholder="Thể loại" />
  </div>
</form>
```

```

    </div>
    <button type="submit" class="btn btn-primary">Thêm sách</button>
  </form>
</template>

<script setup>
  import { ref, defineEmits } from "vue";
  const emit = defineEmits();
  const title = ref("");
  const author = ref("");
  const price = ref(0);
  const category = ref("");
  const submitForm = () => {
    emit("add-book", {
      title: title.value,
      author: author.value,
      price: price.value,
      category: category.value,
    });
    title.value = "";
    author.value = "";
    price.value = 0;
    category.value = "";
  };
</script>

```

2. Kết hợp cùng **Bai1.vue**, để hiển thị danh sách, nhúng component **Bai2.vue** vào **App.vue**

```

<<!-- App.vue -->

<template>

<div class="container mt-4">

<Bai2 @add-book="addBook" />

<Bai1

  v-for="(book, index) in books"

  :key="index"

  :title="book.title"

  :author="book.author"

  :price="book.price"

  :category="book.category"

/>

</div>

</template>

```

```
<script setup>

import { ref } from "vue";

import Bai1 from "../Labs/Lab-5/Bai1.vue";

import Bai2 from "../Labs/Lab-5/Bai2.vue";

const books = ref([]);

const addBook = (book) => {

    books.value.push(book);

};

</script>
```



PHẦN II:

Bài 3: Sử dụng Slots để tạo component linh hoạt

Yêu cầu:

- Tạo component CardContainer.vue sử dụng slot để hiển thị các nội dung khác nhau.
- Cho phép người dùng truyền vào nội dung tùy chỉnh thông qua slot.

Hướng dẫn:

Bước 1: Tạo component Labs/Lab-5/Bai3.vue

```
<template>
<div class="card mb-3">
  <div class="card-body">
    <slot></slot>
  </div>
</div>
</template>
```

Bước 2: Gọi component Bai3.vue ở App.vue

```
<!-- App.vue -->

<template>

<div class="container mt-4">

  <Bai3>
```

```

<h5 class="card-title">Tiêu đề sách</h5>

<h6 class="card-subtitle mb-2 text-muted">Tác giả: Tác giả A</h6>

<p class="card-text">Giá: 150000 VND</p>

<p class="card-text">Thể loại: Fiction</p>

</Bai3>

</div>

</template>

<script setup>

import Bai3 from "../Labs/Lab-5/Bai3.vue";

</script>

```

Bài 4: Sử dụng Scoped Slots và Project/Inject

Yêu cầu:

- Tạo component ThemeProvider.vue để cung cấp một chủ đề cho các component con thông qua provide/inject.
- Sử dụng scoped slots để cho phép các component con truy cập chủ đề.

Hướng dẫn:

- Tạo component **Labs/Lab-5/Bai4.vue**

```

<template>
  <div :class="theme">
    <slot :theme="theme"></slot>
  </div>
</template>
<script setup>
  import { provide, ref } from "vue";
  const theme = ref("dark-theme");
  provide("theme", theme);
</script>

```

Bước 2: Gọi Bai4.vue tại App.vue

```

<!-- App.vue -->
<template>
<div class="container mt-4">
<!-- App.vue -->

<template>

<div class="container mt-4">

<Bai4>

    <template #default="{ theme }">

    <h2 :class="theme">Chủ đề hiện tại: {{ theme }}</h2>

    <Bai3>

    <h5 class="card-title">Tiêu đề sách</h5>

    <h6 class="card-subtitle mb-2 text-muted">Tác giả: Tác giả A</h6>

    <p class="card-text">Giá: 150000 VND</p>

    <p class="card-text">Thể loại: Fiction</p>

    </Bai3>

    </template>

</Bai4>

</div>

</template>

<script setup>

import Bai3 from "./Labs/Lab-5/Bai3.vue";

import Bai4 from "./Labs/Lab-5/Bai4.vue";

</script>

```

**** Giảng viên có thể cho các bài tập phù hợp phục vụ assignment.**

***** Yêu cầu nộp bài:**

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện

trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM

--- Hết ---