**8 bài Labs thực hành**

[**Lab1: Làm quen react**](https://docs.google.com/document/d/1W6wHyeRZWuQ16eRewD-744cvtBv9WpqyFtMc2450UNE/edit#bookmark=id.p38c5kw42mil)

* Cài đặt môi trường: có thể cài thuần với create hoặc cài với vite (nhanh)
* JSX
* Components
* State & Props
* [**Demo**](https://docs.google.com/document/d/1W6wHyeRZWuQ16eRewD-744cvtBv9WpqyFtMc2450UNE/edit#bookmark=id.p38c5kw42mil)
  + Cài đặt
  + Chuyển HTML thành JSX (phụ lục 1)
  + Helloworld
  + Profile (1 hoặc nhiều người)
  + Ví dụ số đếm
  + Ví dụ bật tắt đèn (kèm hình)

**Lab2: Component và Router**

* Layout web bootstrap
* Function Component
* Class Component
* Các kiểu component: lồng nhau (nesting), sử dụng lại (reuse)
* Route
* **Demo**
  + Chuyển một website html thành web react
  + Điều hướng các trang
  + Load dữ liệu từ mảng

**Lab3: Load dữ liệu từ json, API**

* Các loại data: mảng, object,
* Tạo file json
* Dữ liệu giả: faker.js
* Ôn API Nodejs (mySQL, Mongoo)
* Demo
  + Load dữ liệu trang chủ
  + Trang danh mục
  + Trang chi tiết

**Lab4: Form & Events**

**Lab5: Redux**

* **Cart, Checkout**
* **Login / Logout**

**Lab6: Authen & Google API**

**-**

**Lab7:**

**Lab8:**

**Lab1: Làm quen với react**

***1.1. Khởi tạo dự án***

* Tạo một app react đơn giản với

npx create-react-app my-first-app

cd my-first-app

npm start

* Cài react với vite

**Cách 1: gõ lệnh**

npm create vite@latest

Và thực hiện theo hướng dẫn

**Cách 2: gõ lệnh**

npm create vite@latest my-react-app -- --template react

Tạo dự án react với template react

Cài đặt các phụ thuộc cần thiết: npm install

Chạy server: npm run dev

* Tìm hiểu cấu trúc thư mục cổ điển

* Cấu trúc thư mục sử dụng vite

my-react-app/

├── node\_modules/

├── public/

│   └── vite.svg

├── src/

│   ├── assets/

│   │   └── react.svg

│   ├── App.css

│   ├── App.jsx

│   ├── index.css

│   ├── main.jsx

├── .gitignore

├── index.html

├── package.json

├── README.md

├── vite.config.js

* **Demo**: Hello word
  + Tạo component
  + Import và nhúng component

***Thực hành component + app:***

A screenshot of a phone

Description automatically generated

* “public/img/tenhinh.png”

A screen shot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

⇒ nâng cấp 1: sử dụng **props** để truyền chuỗi tên hình ảnh

A screen shot of a computer code

Description automatically generated

**⇒ nâng cấp 2: Khai báo mảng chứa hình ảnh. Đọc mảng gán cho biến props**

A screenshot of a computer code

Description automatically generated

**⇒ cả 2 nâng cấp đều phải fix lại component**

A screenshot of a computer code

Description automatically generated

***1.2. JSX là gì? Component là gì?***

* JSX là cú pháp mở rộng của JavaScript, cho phép viết HTML trong JavaScript.
* [Cách chuyển HTML thành JSX](https://divmagic.com/vi/tools/html-to-jsx)
* Component là khối xây dựng cơ bản của ứng dụng React.
* **Demo**:
  + Tạo component hello, profile
  + Mở app.js import component
  + Khai báo sử dụng

import React from 'react';

import Profile from './components/Profile';

function App() {

    return (

        <div className="App">

            <Profile />

        </div>

    );

}

export default App;

**1.3. State và Props**

* Khái niệm:
  + State: Quản lý dữ liệu động trong component.
  + Props: Truyền dữ liệu từ component cha xuống component con.
* Demo state

import React, { useState } from 'react';

function Counter() {

  const [count, setCount] = useState(0);

  return (

    <div>

      <p>Bạn đã nhấn {count} lần</p>

      <button onClick={() => setCount(count + 1)}>

        Nhấn tôi

      </button>

    </div>

  );

}

* Demo props

* ***Props có thể thay đổi***

import React from 'react';

*// Định nghĩa component con*

function Greeting(props) {

  return <h1>Xin chào, {props.name}!</h1>;

}

// Định nghĩa component cha

function App() {

  return (

    <div>

      <Greeting name="Nguyễn Văn A" />

      <Greeting name="Trần Thị B" />

      <Greeting name="Lê Văn C" />

    </div>

  );

}

export default App;

* ***Props là hằng, không thay đổi trong component con***

import React from 'react';

***// Định nghĩa component con với nhiều props***

function UserCard(props) {

  return (

    <div className="user-card">

      <h2>{props.name}</h2>

      <p>Email: {props.email}</p>

      <p>Age: {props.age}</p>

    </div>

  );

}

***// Định nghĩa component cha***

function App() {

  return (

    <div>

      <UserCard name="Nguyễn Văn A" email="a.nguyen@example.com" age={25} />

      <UserCard name="Trần Thị B" email="b.tran@example.com" age={30} />

      <UserCard name="Lê Văn C" email="c.le@example.com" age={22} />

    </div>

  );

}

export default App;

***Demo kết hợp useState và Props (đếm số)***

import React, { useState } from 'react';

function Counter(props) {

  const [count, setCount] = useState(0);

  return (

    <div>

      <h1>{props.title}</h1>

      <p>Bạn đã nhấn {count} lần</p>

      <button onClick={() => setCount(count + 1)}>

        Nhấn tôi

      </button>

    </div>

  );

}

function App() {

  return (

    <div>

      <Counter title="Counter 1" />

      <Counter title="Counter 2" />

    </div>

  );

}

export default App;

***Một ví dụ khác: sử dụng props và state (mở đèn):***

import React, { useState } from 'react';

function ToggleLight({ label }) {

  const [isOn, setIsOn] = useState(false);

  return (

    <div>

      <h1>{label}</h1>

      <p>Đèn đang {isOn ? 'bật' : 'tắt'}</p>

      <button onClick={() => setIsOn(!isOn)}>

        {isOn ? 'Tắt đèn' : 'Bật đèn'}

      </button>

    </div>

  );

}

export default ToggleLight;

—-----

import React from 'react';

import ToggleLight from './ToggleLight';

function App() {

  return (

    <div>

      <ToggleLight label="Đèn phòng khách" />

      <ToggleLight label="Đèn phòng ngủ" />

    </div>

  );

}

export default App;

—---

***Demo Component “bật đèn” với hình ảnh***

import React, { useState } from 'react';

function ToggleLight({ label }) {

  const [isOn, setIsOn] = useState(false);

  return (

    <div>

      <h1>{label}</h1>

      <p>Đèn đang {isOn ? 'bật' : 'tắt'}</p>

      <img

        src={isOn ? 'path/to/light-on.png' : 'path/to/light-off.png'}

        alt={isOn ? 'Light On' : 'Light Off'}

        style={{ width: '100px', height: '100px' }}

      />

      <button onClick={() => setIsOn(!isOn)}>

        {isOn ? 'Tắt đèn' : 'Bật đèn'}

      </button>

    </div>

  );

}

export default ToggleLight;

—---

import React from 'react';

import ToggleLight from './ToggleLight';

function App() {

  return (

    <div>

      <ToggleLight label="Đèn phòng khách" />

      <ToggleLight label="Đèn phòng ngủ" />

    </div>

  );

}

export default App;

—-------

**Phụ lục 1: *Chuyển đổi HTML thành JSX***

**Lý thuyết:**

* JSX (JavaScript XML) là một cú pháp mở rộng của JavaScript, được sử dụng với React để mô tả giao diện người dùng.
* JSX gần giống với HTML, nhưng có một số khác biệt cần chú ý khi chuyển đổi từ HTML sang JSX.

**Các quy tắc chuyển đổi:**

1.  **Thuộc tính trong HTML và JSX**:

* class -> className
* for -> htmlFor
* style -> đối tượng JavaScript { }
* onclick, onchange -> onClick, onChange, ...

2.  **Tự đóng thẻ**:

* Tất cả các thẻ tự đóng như <img>, <input>, <br> cần được đóng bằng dấu gạch chéo <img />, <input />, <br />.

3.  **Javascript trong JSX**:

* Biểu thức JavaScript trong JSX được đặt trong dấu ngoặc nhọn {}.

4.  **Các thuộc tính Boolean**:

* Trong JSX, các thuộc tính Boolean như checked, disabled, readonly cần được đặt giá trị true hoặc false.

**Ví dụ cụ thể và chuyển đổi từ HTML sang JSX:**

1. **Ví dụ 1: Chuyển đổi đoạn HTML cơ bản**

HTML:

<div class="container">

<h1>Hello, World!</h1>

<p>This is a paragraph.</p>

<a href="https://www.example.com" target="\_blank">Visit Example</a>

</div>

**JSX**:

<div **className**="container">

<h1>Hello, World!</h1>

<p>This is a paragraph.</p>

<a href="https://www.example.com" target="\_blank">Visit Example</a>

</div>

1. **Ví dụ 2: Chuyển đổi đoạn HTML có thuộc tính Boolean**

HTML:

<input type="checkbox" checked>

<button disabled>Submit</button>

**JSX**:

<input type="checkbox" **checked={true}** />

<button disabled={true}>Submit</button>

1. **Ví dụ 3: Chuyển đổi đoạn HTML có sự kiện**

HTML:

<button onclick="handleClick()">Click me</button>

JSX:

<button **onClick={handleClick}**>Click me</button>

1. **Ví dụ 4: Chuyển đổi đoạn HTML có thuộc tính style**

HTML:

<div style="color: red; font-size: 16px;">

This is styled text.

</div>

JSX:

<div **style={{ color: 'red', fontSize: '16px' }}**>

This is styled text.

</div>

1. **Ví dụ 5: Chuyển đổi đoạn HTML phức tạp**

HTML:

<div class="profile">

<img src="profile.jpg" alt="Profile Picture">

<h2>John Doe</h2>

<p>Web Developer at Example Inc.</p>

<a href="https://www.example.com" target="\_blank">Visit My Website</a>

</div>

JSX:

<div **className**="profile">

<img src="profile.jpg" alt="Profile Picture" />

<h2>John Doe</h2>

<p>Web Developer at Example Inc.</p>

<a href="https://www.example.com" target="\_blank">Visit My Website</a>

</div>

**Thực hành:**

1. **Chuyển đổi đoạn HTML sau sang JSX và hiển thị trong một thành phần React**:

HTML:

<form>

<label for="name">Name:</label>

<input type="text" id="name" name="name">

    <label for="email">Email:</label>

<input type="email" id="email" name="email">

    <button type="submit">Submit</button>

</form>

JSX:

import React from 'react';

function MyForm() {

return (

     <form>

         <label htmlFor="name">Name:</label>

         <input type="text" id="name" name="name" />

         <label htmlFor="email">Email:</label>

         <input type="email" id="email" name="email" />

         <button type="submit">Submit</button>

     </form>

);

}

export default MyForm;

1. **Chuyển đổi đoạn HTML sau sang JSX và hiển thị trong một thành phần React**:

HTML:

<nav>

<ul>

     <li><a href="/home">Home</a></li>

     <li><a href="/about">About</a></li>

     <li><a href="/contact">Contact</a></li>

</ul>

</nav>

JSX:

import React from 'react';

function Navbar() {

return (

     <nav>

         <ul>

             <li><a href="/home">Home</a></li>

             <li><a href="/about">About</a></li>

             <li><a href="/contact">Contact</a></li>

         </ul>

     </nav>

);

}

export default Navbar;