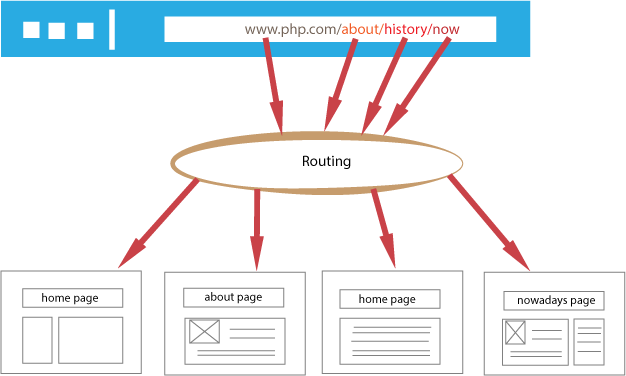
**REACT 12: REACT ROUTER**

A – LÝ THUYẾT

I – GIỚI THIỆU VỀ REACT ROUTER

**1. Router là gì**

*Router là một phân đoạn trên thanh địa chỉ của trình duyệt (gần giống với khái niện URL), mà mỗi những Router khác nhau đó được chạy trên trình duyệt thì tương đương với việc Client đang gửi những Request tương ứng lên Server*



**2. React Router**

React thông thường chỉ truyền và nhận các giá trị qua lại giữa các Component, nếu muốn xây dựng các ứng dụng SPA (Single Page Application), có nghĩa là thay đổi dữ liệu thông qua việc thay đổi các Page khác nhau thì phải sử dụng thêm Module React Router

**3. Cài đặt và sử dụng**

|  |
| --- |
| npm install react-router-dom --save |

**Code JS:**

|  |
| --- |
| import { BrowserRouter, Routes, Route, Link } from "react-router-dom"; |

**Trong đó:**

* **BrowserRouter**: đối tượng được sử dụng để bao lấy toàn bộ View của App. Nó sẽ cung cấp môi trường để từ đó có thể sử dụng các đối tượng và phương thức khác trong Module React Router.
* **Routes**: là đối tượng được sử dụng để chứa tất cả các Router trong ứng dụng.
* **Route**: đây là đối tượng được sử dụng để triệu gọi các component tương ứng cho mỗi Router khác nhau.
* **Link**: đối tượng liên kết thay thế cho thẻ A, với mục đích làm cho việc thay đổi giữ liệu nhưng không tải lại trang (reload)

II – LÀM VIỆC VỚI MODULE REACT ROUTER DOM

**1. Đối tượng BrowserRouter**

Đối tượng BrowserRouter được sử dụng để bao bọc lấy toàn bộ App, để có thể bắt được và hiểu được các Router rồi từ đó sẽ gọi các Component tương ứng

**Code JS:**

|  |
| --- |
| <BrowserRouter>  <div>  <ul>  <li><a href="#">Home</a></li>  <li><a href="#">About</a></li>  <li><a href="#">Blog</a></li>  <li><a href="#">Contact</a></li>  </ul>  <div>  <h2>Home</h2>  <h2>About</h2>  <h2>Blog</h2>  <h2>Contact</h2>  <h2>Page 404</h2>  </div>  </div>  </ BrowserRouter > |

**2. Đối tượng Link**

Đối tượng này sẽ thay thế cho các thẻ liên kết A trong HTML, tuy nhiên bản chất nó vẫn là thẻ A, nhưng đã được xử lý return FALSE khi click vào để không có hiệu ứng tải trang.

**Code JS:**

|  |
| --- |
| < BrowserRouter >  <div>  <ul>  <li><Link to="/">Home</Link></li>  <li><Link to="/about">About</Link></li>  <li><Link to="/blog">Blog</Link></li>  <li><Link to="/contact">Contact</Link></li>  </ul>  <div>  <h2>Home</h2>  <h2>About</h2>  <h2>Blog</h2>  <h2>Contact</h2>  <h2>Page 404</h2>  </div>  </div>  </ BrowserRouter > |

**3. Đối tượng Routes và Route**

Mỗi một **Route** sẽ có nhiệm vụ bắt các Router trên URL, từ đó phân tích và triệu gọi các Component tương ứng.

Trong một ứng dụng thì sẽ có rất nhiều **Route** để bắt các Router tương ứng, và tất cả các Route này sẽ được đặt trong một môi trường Router đó là đối tượng **Routes**

**Bước 1:**

Bóc tất cả các thẻ Heading thành các Component riêng biệt

**Code Component Home**

|  |
| --- |
| import React from "react"  const Home = ()=>{  return (  <h2>Home</h2>  )  }  export default Home |

**Code Component About**

|  |
| --- |
| import React from "react"  const About = ()=>{  return (  <h2>About </h2>  )  }  export default About |

**Code Component Blog**

|  |
| --- |
| import React from "react"  const Blog = ()=>{  return (  <h2>Blog </h2>  )  }  export default Blog |

**Code Component Contact**

|  |
| --- |
| import React from "react"  const Contact = ()=>{  return (  <h2>Contact </h2>  )  }  export default Contact |

**Code Component Page404**

|  |
| --- |
| import React from "react"  const Page404 = ()=>{  return (  <h2>Page404 </h2>  )  }  export default Page404 |

**Bước 2:**

Sử dụng đối tượng Route cho phần Main, cơ chế hoạt động của đối tượng này khá đơn giản đó là môi khi nhận được một Router nào đó thì nó sẽ triệu gọi một Component tương ứng

**Code JS:**

|  |
| --- |
| < BrowserRouter >  <div>  <ul>  <li><Link to="/">Home</Link></li>  <li><Link to="/About">About</Link></li>  <li><Link to="/Blog">Blog</Link></li>  <li><Link to="/Contact">Contact</Link></li>  </ul>  <div>  <Routes>  <Route path="/" element={<Home />} />  <Route path="/about" element={<About />} />  <Route path="/blog" element={<Blog />} />  <Route path="/contact" element={<Contact />} />  </Routes>  </div>  </div>  </ BrowserRouter > |

**Trong đó:**

* **path**: bắt các phân đoạn của Router được truyền lên URL
* **element**: gọi một Component bất kỳ khi bắt được Router từ path

**5. Router Not Found**

Chúng ta cũng có thể chỉ định cho đối tượng Route triệu gọi một Component bất kỳ nếu như không nhận được một Router hợp lệ. Cơ chế này hoạt động rất giống **default** trong **switch**, đó là cho phép triệu gọi một Component được chỉ định trước nếu như các Route không đọc được một Router hợp lệ

**Code JS:**

|  |
| --- |
| < BrowserRouter >  <div>  <ul>  <li><Link to="/">Home</Link></li>  <li><Link to="/about">About</Link></li>  <li><Link to="/blog">Blog</Link></li>  <li><Link to="/contact">Contact</Link></li>  </ul>  <div>  <Routes>  <Route path="/" element={<Home />} />  <Route path="/about" element={<About />} />  <Route path="/blog" element={<Blog />} />  <Route path="/contact" element={<Contact />} />  <Route path="\*" element={<Page404 />} />  </Routes>  </div>  </div>  </ BrowserRouter > |

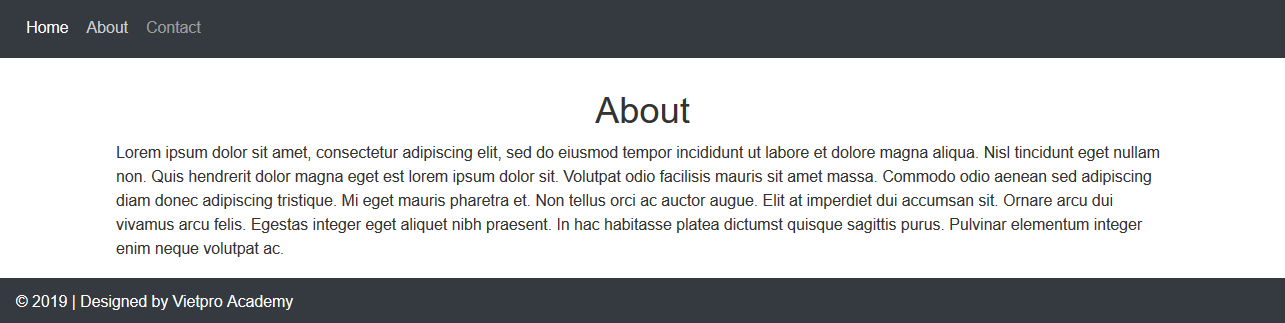
B – THỰC HÀNH

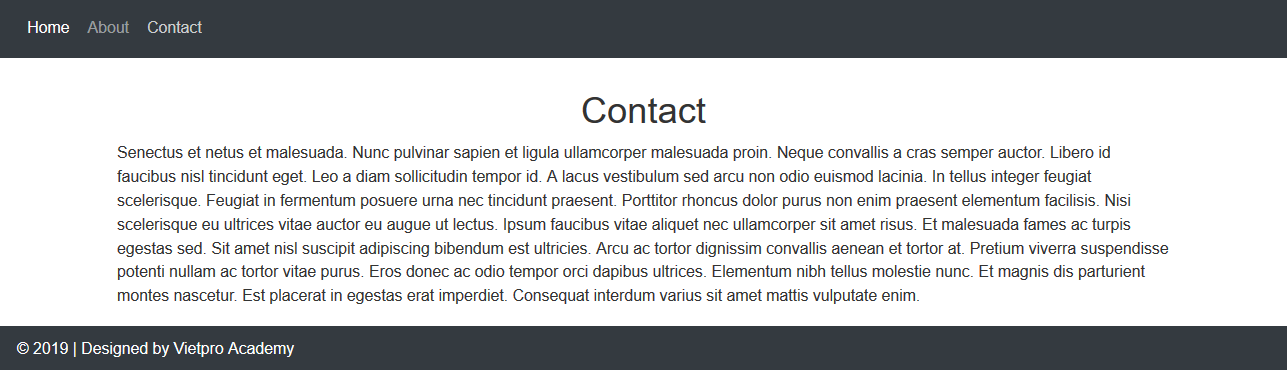
I – BÀI TẬP THỰC HÀNH

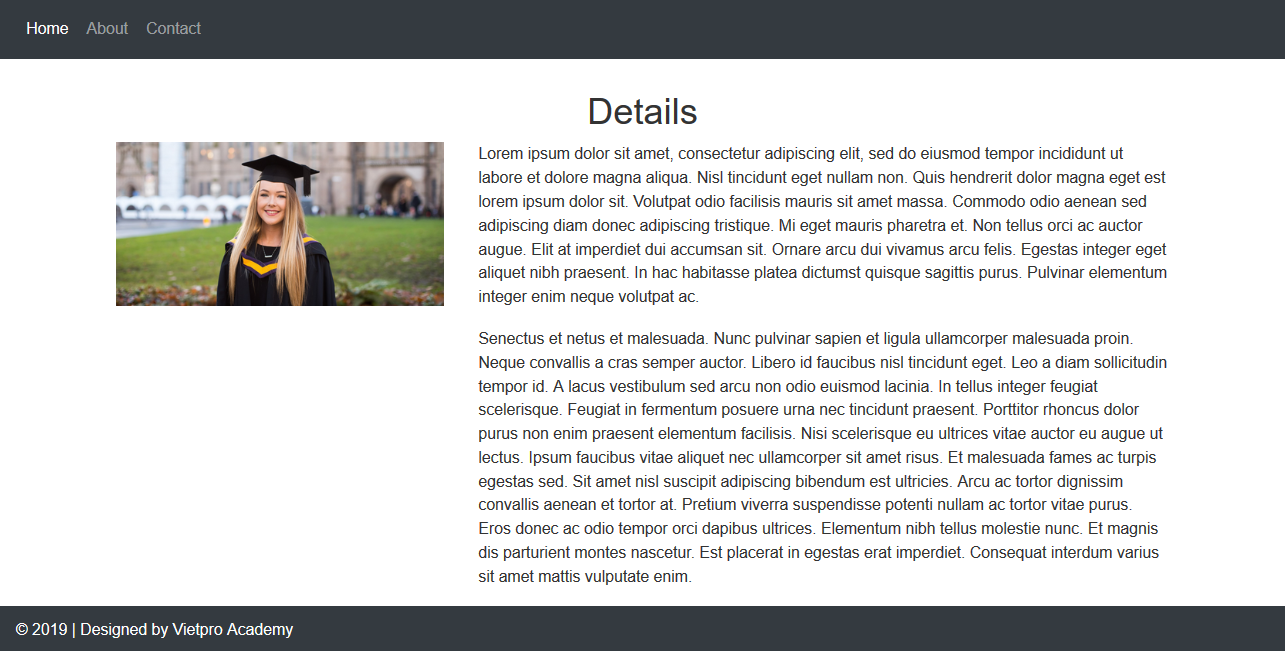
**Bài 1:**

Xây dựng ứng dụng web có tính năng chuyển trang như dưới đây









II – BÀI TẬP VỀ NHÀ

**Bài 1:**

Xây dựng ứng dụng web có tính năng chuyển trang như dưới đây

