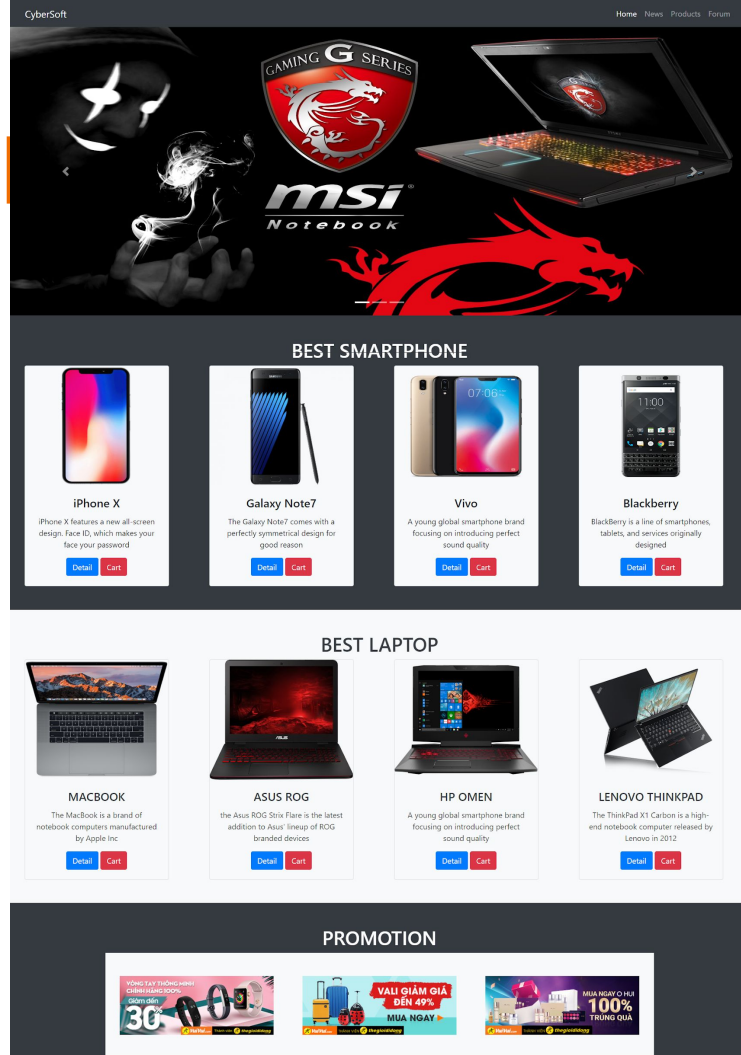

REACTJS

PHÓ NGHĨA VĂN
CYBERSOFT.EDU.VN

PROJECT 02 - BÁN HÀNG ONLINE

- Dựa trên layout có sẵn
- Xây dựng component để tái sử dụng:
 - Sản phẩm smartphone
 - Sản phẩm Laptop
 - Khuyến mãi
- Chức năng:
 - Xem chi tiết sản phẩm → popup modal
 - Đặt hàng
- Kiến thức:
 - JSX (= html + JS)
 - Stateless và stateful component
 - Props and State
 - Loops/Conditions
 - Handle event



Cài đặt

- Visual studio code
- NodeJS – phiên bản LTS

Cài đặt - Command line/Terminal

- npm install create-react-app (chỉ cài cho lần đầu tiên)
- create-react-app {my-app} (nếu không được thì thêm npx vào phía trước)
- cd {my-app}
- npm start
- npm install -g react-create (cài đặt react-create để tạo component bằng command line)
- react-create component Home --entry (tại component tên Home)
- Đổi port tại package.json

Cài đặt bootstrap 4

Có 3 cách import bootstrap 4 vào project:

Link CDN

- Download file zip và cài đặt
- `npm install bootstrap`

React.createElement()

React.CreateElement("div", {}, "Nội dung") → gồm 3 thành phần: tag html, properties dạng object và nội dung trong tag html

```
class App extends Component {  
  render() { //JSX  
  
    return React.createElement("div", {className:"App"}, React.createElement(  
      "p", {className:"App-intro"}, "Hôm nay là bài học đầu tiên về ReactJS"  
    ));  
  }  
}
```

Hôm nay là bài học đầu tiên về ReactJS

→ Gồm nhiều React.creatElement lồng vào nhau rất phức tạp

JSX

- JSX = HTML + JavaScript
- Cấu trúc tường minh hơn
- <https://magic.reactjs.net/htmltojsx.htm>
(Tool chuyển HTML sang JSX)
- Có thể tự build tool chuyển HTML sang JSX bằng ReactJS

```
class App extends Component {  
  render() { //JSX  
    return (  
      <div className="App">  
        <p className="App-intro">  
          Hôm nay là bài học đầu tiên về ReactJS  
        </p>  
      </div>  
    );  
  }  
}
```

Convert HTML to JSX

- Vào trang web sau: <https://magic.reactjs.net/htmltojsx.htm>
- Copy HTML code vào ô bên trái, JSX code tương ứng sẽ được tạo bên phải

Live HTML Editor

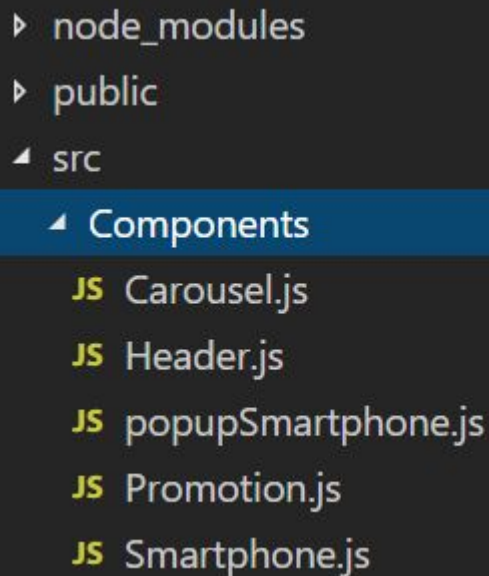
```
<!-- Hello world -->
<div class="awesome" style="border: 1px solid red">
  <label for="name">Enter your name: </label>
  <input type="text" id="name" />
</div>
<p>Enter your HTML here</p>
```

```
var NewComponent = React.createClass({
  render: function() {
    return (
      <div>
        { /* Hello world */ }
        <div className="awesome" style={{border: '1px solid red'}}>
          <label htmlFor="name">Enter your name: </label>
          <input type="text" id="name" />
        </div>
        <p>Enter your HTML here</p>
      </div>
    );
  }
});
```


Components

Tạo một folder chứa tất cả các components, tương ứng với các thành phần của layout

- Header (Stateless)
- Carousel (Stateless)
- Promotion (Stateless)
- Smartphone (Stateful)
- popupSmartphone (Stateful)



```
node_modules
public
src
  Components
    JS Carousel.js
    JS Header.js
    JS popupSmartphone.js
    JS Promotion.js
    JS Smartphone.js
```

Components

Muốn sử dụng Components ở đâu thì import vào đó

```
JS App.js  x
1  import React, { Component } from 'react';
2  import './App.css';
3  import swal from 'sweetalert';
4
5  // IMPORT COMPONENT
6  import Header from './Components/Header';
7  import Carousel from './Components/Carousel';
8  import Promotion from './Components/Promotion';
9  import Smartphone from './Components/Smartphone';
10
```

Stateless component

- Là một arrow function, return cấu trúc JSX
- Không thể sử dụng lifecycle
- Không thể sử dụng refs
- Dễ viết, dễ kiểm tra
- Tránh hoàn toàn từ khóa this
- Không quản lý các state nào
- Nên sử dụng nhiều stateless nhất có thể → dễ quản lý logic của app, dễ test và kiểm tra

```
1  import React from 'react';
2
3  const Header = () => {
4    |    return (
5    |        // Cấu trúc JSX
6    |    )
7  }
8
9  export default Header;
```

Stateless component - Header.js

```
1 import React from 'react';
2
3 const Header = () => {
4   |   return (
5   |       // Cấu trúc JSX
6   |   )
7 }
8
9 export default Header;
```

```
const Header = () => {
  return (
    <nav className="navbar navbar-expand-md navbar-dark bg-dark">
      <div className="container-fluid">
        <a href="#" className="navbar-brand">CyberSoft</a>
        <button className="navbar-toggler" data-toggle="collapse" data-target="#myMenu">
          <span className="navbar-toggler-icon" />
        </button>
        <div className="collapse navbar-collapse" id="myMenu">
          <ul className="navbar-nav ml-auto">
            <li className="nav-item">
              <a href="#" className="nav-link active">Home</a>
            </li>
            <li className="nav-item">
              <a href="#" className="nav-link">News</a>
            </li>
            <li className="nav-item dropdown">
              <a href="#" className="nav-link" data-toggle="dropdown">Products</a>
              <div className="dropdown-menu">
                <a href="#" className="dropdown-item">Smarphone</a>
                <a href="#" className="dropdown-item">Laptop</a>
              </div>
            </li>
            <li className="nav-item">
              <a href="#" className="nav-link">Forum</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
  )
}
```

Stateless component - Carousel.js

```
const Carousel = () => [{
  return (
    <div id="demo" className="carousel slide" data-ride="carousel">
      {/* Indicators */}
      <ul className="carousel-indicators">
        <li data-target="#demo" data-slide-to={0} className="active" />
        <li data-target="#demo" data-slide-to={1} />
        <li data-target="#demo" data-slide-to={2} />
      </ul>
      {/* The slideshow */}
      <div className="carousel-inner">
        <div className="carousel-item active">
          <img src={require('../img/slide_1.jpg')} alt="Los Angeles" width={'100%'} height={500} />
        </div>
        <div className="carousel-item">
          <img src={require('../img/slide_2.jpg')} alt="Chicago" width={'100%'} height={500} />
        </div>
        <div className="carousel-item">
          <img src={require('../img/slide_3.jpg')} alt="New York" width={'100%'} height={500} />
        </div>
      </div>
      {/* Left and right controls */}
      <a className="carousel-control-prev" href="#demo" data-slide="prev">
        <span className="carousel-control-prev-icon" />
      </a>
      <a className="carousel-control-next" href="#demo" data-slide="next">
        <span className="carousel-control-next-icon" />
      </a>
    </div>
  )
}]
```

Stateless component - Carousel.js

```
const Promotion = () => {
  return (
    <section id="promotion" className="container-fluid pt-5 pb-5">
      <h1 className="text-center text-white">PROMOTION</h1>
      <div className="container bg-light pt-5 pb-5">
        <div className="row">
          <div className="col-xs-12 col-sm-12 col-md-4">
            <div className="container">
              <img src={require('../img/promotion_1.png')} alt="" style={{ maxWidth: '100%' }} />
            </div>
          </div>
          <div className="col-xs-12 col-sm-12 col-md-4">
            <div className="container">
              <img src={require('../img/promotion_2.png')} alt="" style={{ maxWidth: '100%' }} />
            </div>
          </div>
          <div className="col-xs-12 col-sm-12 col-md-4">
            <div className="container">
              <img src={require('../img/promotion_3.jpg')} alt="" style={{ maxWidth: '100%' }} />
            </div>
          </div>
        </div>
      </div>
    </section>
  )
}
```

Định nghĩa style cho components

```
const Promotion = () => {  
  const style = {  
    width: '100%',  
    height: '400px',  
    backgroundColor: 'rgba(0,0,0,0.5)'  
  }  
  return (  
    <section id="promotion" className="container-fluid pt-5 pb-5" style={style}>  
      <h1 className="text-center text-white">PROMOTION</h1>  
      <div className="container bg-light pt-5 pb-5">  
        <div className="row">  
          <div className="col-xs-12 col-sm-12 col-md-4">
```

Stateful component

- Cho phép sử dụng **state**
- Định nghĩa **constructor** với **props** là tham số đầu vào
- **this.props** được sử dụng trong **render(){ ... }**
- Nếu sử dụng hàm constructors → phải gọi **super()**
- Nếu có dùng **this.props** thì khai báo **super(props)**, nếu không thì chỉ cần dùng **super()**

Stateful component

```
class Smartphone extends Component {  
  render() { //JSX  
    console.log(this.props.id);  
    return (  
      //JSX code  
    );  
  }  
}  
  
export default Smartphone;
```

Stateful component - Smartphone.js

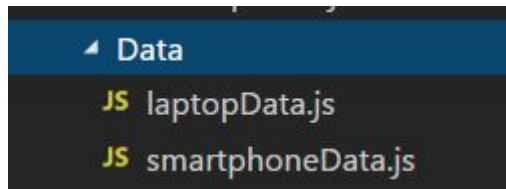
```
class Smartphone extends Component {  
  render() { //JSX  
    console.log(this.props.id);  
    return (  
      <div className="Smartphone">  
        <div className="col-xs-12 col-sm-6 col-md-6 col-lg-3 col-lg-3">  
          <div className="container">  
            <div className="card bg-light" style={{ width: 300 }}>  
              <img className="card-img-top" src={} alt="Card image" style={{ maxWidth: '100%', height: 250 }} />  
              <div className="card-body text-center">  
                <h4 className="card-title text-center">{}</h4>  
                <h4 className="card-title text-center">{}</h4>  
                <p className="card-text">{}</p>  
                <a className="btn btn-primary" data-toggle="modal" data-target={}>Detail</a>  
                <a className="btn btn-danger" >Cart</a>  
              </div>  
            </div>  
          </div>  
        </div>  
      </div>  
    );  
  }  
}
```

Stateful component - Popupsmartphone.js

```
class Popupsmartphone extends Component {
  render() { //JSX
    return (
      <div className="modal fade" id={} >
        <div className="modal-dialog">
          <div className="modal-content">
            { /* Modal Header */ }
            <div className="modal-header">
              <h4 className="modal-title">{}</h4>
              <button type="button" className="close" data-dismiss="modal">x</button>
            </div>
            { /* Modal body */ }
            <div className="modal-body">
              <table className="table">
                <tbody>
                  <tr>
                    <td>Price</td>
                    <td>{}</td>
                  </tr>
                  <tr>
                    <td>Screen</td>
                    <td>{}</td>
                  </tr>
                  <tr>
                    <td>frontCamera</td>
                    <td>{}</td>
                  </tr>
                  <tr>
                    <td>backCamera</td>
                    <td>{}</td>
                  </tr>
                </tbody>
              </table>
            </div>
            { /* Modal footer */ }
            <div className="modal-footer">
              <button type="button" className="btn btn-danger" data-dismiss="modal">Close</button>
            </div>
          </div>
        </div>
      </div>
    );
  }
}
```

Data

Tạo folder chứa các dữ liệu



```
let smartphones = [  
  {  
    id: 'sp_1',  
    name: 'iphoneX',  
    price: '30.000.000 VNĐ',  
    screen: 'screen_1',  
    backCamera: 'backCamera_1',  
    frontCamera: 'frontCamera_1',  
    img: './img/sp_iphoneX.png',  
    desc: 'iPhone X features a new all-screen design. Face ID, which makes your face your password'  
  },  
  {  
    id: 'sp_2',  
    name: 'Note 7',  
    price: '20.000.000 VNĐ',  
    screen: 'screen_2',  
    backCamera: 'backCamera_2',  
    frontCamera: 'frontCamera_2',  
    img: './img/sp_note7.png',  
    desc: 'The Galaxy Note7 comes with a perfectly symmetrical design for good reason'  
  }  
]
```

State

- Biểu diễn trạng thái của component
- Tính private: Chỉ sử dụng trong component đó
- Thiết lập state: **this.state = { key_1: value_1, key_2: value_2 }**
- Thay đổi state: **this.setState = { name: 'Hackagon' }**

```
class App extends Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      spInfo: smartphones  
    }  
  }  
}
```

props

- Cho phép truyền dữ liệu từ component cha vào component con
- Sử dụng: trong component con **this.props. ...**

```
constructor(props){  
  super(props);  
  this.state = {  
    spInfo: smartphones  
  }  
}
```

Data

App.js

Smartphone.js

popupSmartphone.js

```
<Popupsmartphone id={this.props.id}  
  phoneName= {this.props.phoneName}  
  desc={this.props.desc}  
  price = {this.props.price}  
  screen = {this.props.screen}  
  frontCamera = {this.props.frontCamera}  
  backCamera = {this.props.backCamera}  
>
```

```
<Smartphone  
  id = {sp.id}  
  phoneName= {sp.name}  
  desc={sp.desc}  
  imgUrl = {require(`_${sp.img}`)}  
  price = {sp.price}  
  screen = {sp.screen}  
  frontCamera = {sp.frontCamera}  
  backCamera = {sp.backCamera}  
>
```

Thư viện sweetalert.js

- Sử dụng npm
 - Cài đặt: **npm install sweetalert --save**
 - Sử dụng: **import swal from 'sweetalert';**
- Sử dụng CDN:
 - **<script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>**

Thư viện sweetalert.js

Kết hợp với event

```
handleCart = () => {  
  swal({  
    title: "Are you sure?",  
    text: "Your money will be discount",  
    icon: "warning",  
    buttons: true,  
    dangerMode: true,  
  })  
  .then((willDelete) => {  
    if (willDelete) {  
      swal("You lost your money", {  
        icon: "success",  
      });  
    } else {  
      swal("Your money was safe");  
    }  
  });  
}
```

```
<a className="btn btn-primary" data-toggle="modal" data-target={`#${this.props.id}`}>Detail</a>  
<a className="btn btn-danger" onClick={this.handleCart}>Cart</a>
```


Tổng kết kiến thức

- Cài đặt môi trường và khởi động project
- Cài đặt bootstrap 4, thư viện sweetalert
- Cấu trúc JSX
- Stateful và stateless component - khởi tạo và sử dụng
- Cách phân chia các component
- Tạo dữ liệu giả
- State → private: sử dụng trong một component
- Props → truyền dữ liệu từ component cha sang con