

MỤC TIÊU:

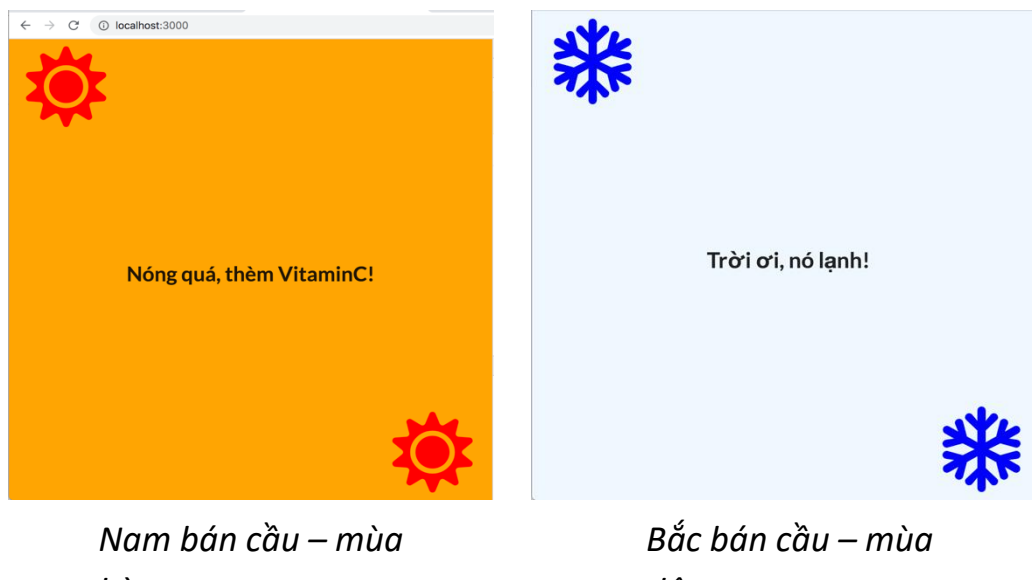
Kết thúc bài thực hành này bạn có khả năng

- ✓ State.
- ✓ Lifecycle.

NỘI DUNG

Tạo project có tên là seasons với cấu trúc và yêu cầu như sau

Nội dung hiển thị



Yêu cầu

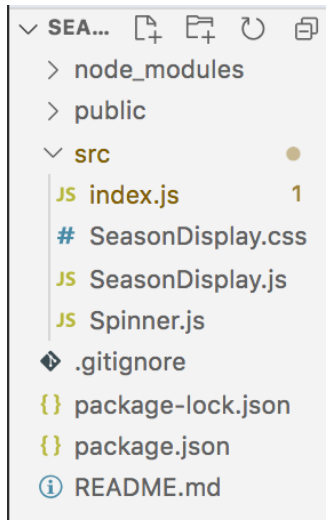
Yêu cầu của app

Cần lấy vị trí (location) của người dùng
(thay đổi)

Xác định được tháng (month) hiện hành
(thay đổi)

Thay đổi nội dung hiển thị và style theo
location + month

Cấu trúc project



Bài 1: Lấy vị trí location dùng state

SeasonDisplay.js	Index.js
<pre> JS SeasonDisplay.js × src > JS SeasonDisplay.js > ... 1 import React from 'react'; 2 3 const SeasonDisplay = () => { 4 return <div>Season Display</div>; 5 }; 6 7 export default SeasonDisplay; </pre>	<pre> JS index.js × src > JS index.js > ... 1 import React from 'react'; 2 import ReactDOM from 'react-dom'; 3 4 class App extends React.Component { 5 constructor(props) { 6 super(props); 7 8 // THIS IS THE ONLY TIME we do direct assignment 9 // to this.state 10 this.state = { lat: null, errorMessage: '' }; 11 12 window.navigator.geolocation.getCurrentPosition(13 position => { 14 // we called setState!!!! 15 this.setState({ lat: position.coords.latitude }); 16 }, 17 err => { 18 this.setState({ errorMessage: err.message }); 19 } 20); 21 } 22 23 // React says we have to define render!! 24 render() { 25 if (this.state.errorMessage && !this.state.lat) { 26 return <div>Error: {this.state.errorMessage}</div>; 27 } 28 29 if (!this.state.errorMessage && this.state.lat) { 30 return <div>Latitude: {this.state.lat}</div>; 31 } 32 33 return <div>Loading!</div>; 34 } 35 } 36 37 ReactDOM.render(<App />, document.querySelector('#root')); 38 </pre>

Bài 2: Life cycle

SeasonDisplay.js	Index.js
<pre> JS SeasonDisplay.js × src > JS SeasonDisplay.js > ... 1 import React from 'react'; 2 3 const SeasonDisplay = () => { 4 return <div>Season Display</div>; 5 }; 6 7 export default SeasonDisplay; </pre>	<pre> JS index.js × src > JS index.js > ... 1 import React from 'react'; 2 import ReactDOM from 'react-dom'; 3 4 class App extends React.Component { 5 state = { lat: null, errorMessage: '' }; 6 7 componentDidMount() { 8 window.navigator.geolocation.getCurrentPosition(9 position => this.setState({ lat: position.coords.latitude }), 10 err => this.setState({ errorMessage: err.message }) 11); 12 } 13 14 // React says we have to define render!! 15 render() { 16 if (this.state.errorMessage && !this.state.lat) { 17 return <div>Error: {this.state.errorMessage}</div>; 18 } 19 20 if (!this.state.errorMessage && this.state.lat) { 21 return <div>Latitude: {this.state.lat}</div>; 22 } 23 24 return <div>Loading!</div>; 25 } 26 } 27 28 ReactDOM.render(<App />, document.querySelector('#root')); </pre>

Bài 3: Chuyển state như props

SeasonDisplay.js	Index.js
<pre> JS SeasonDisplay.js X src > JS SeasonDisplay.js > ... 1 import React from 'react'; 2 3 const SeasonDisplay = props => { 4 console.log(props.lat); 5 6 return <div>Season Display</div>; 7 }; 8 9 export default SeasonDisplay; 10 </pre>	<pre> JS index.js X src > JS index.js > ... 1 import React from 'react'; 2 import ReactDOM from 'react-dom'; 3 import SeasonDisplay from './SeasonDisplay'; 4 5 class App extends React.Component { 6 state = { lat: null, errorMessage: '' }; 7 8 componentDidMount() { 9 window.navigator.geolocation.getCurrentPosition(10 position => this.setState({ lat: position.coords.latitude }), 11 err => this.setState({ errorMessage: err.message }) 12); 13 } 14 15 // React says we have to define render!! 16 render() { 17 if (this.state.errorMessage && !this.state.lat) { 18 return <div>Error: {this.state.errorMessage}</div>; 19 } 20 21 if (!this.state.errorMessage && this.state.lat) { 22 return <SeasonDisplay lat={this.state.lat} />; 23 } 24 25 return <div>Loading!</div>; 26 } 27 } 28 29 ReactDOM.render(<App />, document.querySelector('#root')); 30 </pre>

Bài 4: Xác định tháng (month) hiện hành

- Cách xác định thời tiết dựa vào tháng.
- Bắc bán cầu

0 +	1	2	3	4	5	6	7	8	9	10	11
Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec

- Nam bán cầu

0	1	2	3	4	5	6	7	8	9	10	11
Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec

SeasonDisplay.js	Index.js
<pre> JS SeasonDisplay.js × src > JS SeasonDisplay.js > ... 1 import React from 'react'; 2 3 const getSeason = (lat, month) => { 4 if (month > 2 && month < 9) { 5 return lat > 0 ? 'summer' : 'winter'; 6 } else { 7 return lat > 0 ? 'winter' : 'summer'; 8 } 9 }; 10 11 const SeasonDisplay = props => { 12 const season = getSeason(props.lat, new Date().getMonth()); 13 console.log(season); 14 return <div>Season Display</div>; 15 }; 16 export default SeasonDisplay; 20 </pre>	<pre> JS index.js × src > JS index.js > ... 1 import React from 'react'; 2 import ReactDOM from 'react-dom'; 3 import SeasonDisplay from './SeasonDisplay'; 4 5 class App extends React.Component { 6 state = { lat: null, errorMessage: '' }; 7 8 componentDidMount() { 9 window.navigator.geolocation.getCurrentPosition(10 position => this.setState({ lat: position.coords.latitude }), 11 err => this.setState({ errorMessage: err.message }) 12); 13 } 14 15 // React says we have to define render!! 16 render() { 17 if (this.state.errorMessage && !this.state.lat) { 18 return <div>Error: {this.state.errorMessage}</div>; 19 } 20 21 if (!this.state.errorMessage && this.state.lat) { 22 return <SeasonDisplay lat={this.state.lat} />; 23 } 24 25 return <div>Loading!</div>; 26 } 27 } 28 29 ReactDOM.render(<App />, document.querySelector('#root')); 30 </pre>

Bài 5: Trích xuất options thành config

SeasonDisplay.js	Index.js
<pre> const seasonConfig = { summer: { text: "Nóng quá, thêm VitaminC!", iconName: 'sun' }, winter: { text: 'Trời ơi, nó lạnh!', iconName: 'snowflake' } }; const getSeason = (lat, month) => { if (month > 2 && month < 9) { return lat > 0 ? 'summer' : 'winter'; } else { return lat > 0 ? 'winter' : 'summer'; } }; const SeasonDisplay = props => { const season = getSeason(props.lat, new Date().getMonth()); const { text, iconName } = seasonConfig[season]; return (<div> <i className={` \${iconName} icon`} /> <h1>{text}</h1> <i className={` \${iconName} icon`} /> </div>); }; export default SeasonDisplay; </pre>	<pre> JS index.js × src > JS index.js > ... 1 import React from 'react'; 2 import ReactDOM from 'react-dom'; 3 import SeasonDisplay from './SeasonDisplay'; 4 5 class App extends React.Component { 6 state = { lat: null, errorMessage: '' }; 7 8 componentDidMount() { 9 window.navigator.geolocation.getCurrentPosition(10 position => this.setState({ lat: position.coords.latitude }), 11 err => this.setState({ errorMessage: err.message }) 12); 13 } 14 15 // React says we have to define render!! 16 render() { 17 if (this.state.errorMessage && !this.state.lat) { 18 return <div>Error: {this.state.errorMessage}</div>; 19 } 20 21 if (!this.state.errorMessage && this.state.lat) { 22 return <SeasonDisplay lat={this.state.lat} />; 23 } 24 25 return <div>Loading!</div>; 26 } 27 } 28 29 ReactDOM.render(<App />, document.querySelector('#root')); 30 </pre>

Bài 6: Style

SeasonDisplay.css X

```
src > # SeasonDisplay.css > ...
1  .icon-left {
2    position: absolute;
3    top: 10px;
4    left: 10px;
5  }
6
7  .icon-right {
8    position: absolute;
9    bottom: 10px;
10   right: 10px;
11 }
12
13 .season-display.winter i {
14   color: blue;
15 }
16
17 .season-display.summer i {
18   color: red;
19 }
20
21 .season-display {
22   display: flex;
23   justify-content: center;
24   align-items: center;
25   height: 100vh;
26 }
27
28 .winter {
29   background-color: #aliceblue;
30 }
31
32 .summer {
33   background-color: #orange;
34 }
```

```
> const seasonConfig = { ...
};

> const getSeason = (lat, month) => { ...
};

const SeasonDisplay = props => {
  const season = getSeason(props.lat, new Date().getMonth());
  const { text, iconName } = seasonConfig[season];

  return (
    <div className={`season-display ${season}`}>
      <i className={`icon-left massive ${iconName} icon`} />
      <h1>{text}</h1>
      <i className={`icon-right massive ${iconName} icon`} />
    </div>
  );
};

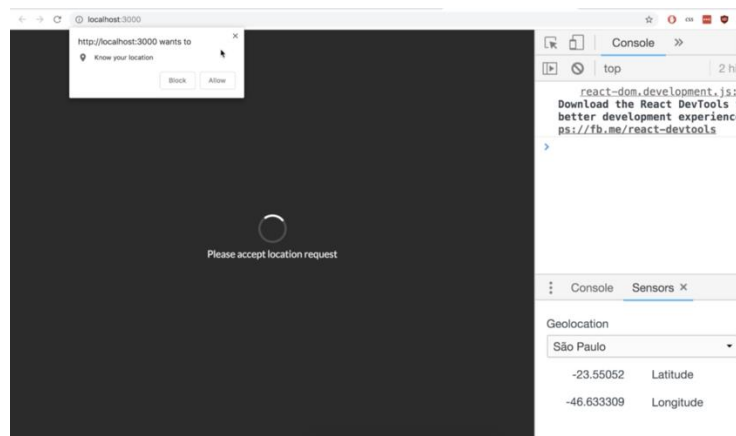
export default SeasonDisplay;
```

Bài 5: Tạo spinner

- Tạo spinner như mẫu

Gợi ý:

- o Tạo spinner.js
- o Thay điều kiện loading bằng spinner
- o Dùng default props với mess là “Please accept location request”



- Khắc phục render: tránh sử dụng điều kiện khi render

Gợi ý:

- Tạo function RenderContent().
- Khi render gọi RenderContent()

```
import React from 'react';
import ReactDOM from 'react-dom';
import SeasonDisplay from './SeasonDisplay';
import Spinner from './Spinner';

class App extends React.Component {
  state = { lat: null, errorMessage: '' };

  componentDidMount() {
    window.navigator.geolocation.getCurrentPosition(
      position => this.setState({ lat: position.coords.latitude }),
      err => this.setState({ errorMessage: err.message })
    );
  }

  // React says we have to define render!!
  render() {
    if (this.state.errorMessage && !this.state.lat) {
      return <div>Error: {this.state.errorMessage}</div>;
    }

    if (!this.state.errorMessage && this.state.lat) {
      return <SeasonDisplay lat={this.state.lat} />;
    }

    return <Spinner message="Please accept location request" />;
  }
}

ReactDOM.render(<App />, document.querySelector('#root'));
```

Tránh dùng điều kiện khi render

***** 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 ---