Đinh Minh Bảo – 19CNTN Khoa CNTT HCMUS

Tạo responsive navbar và side drawer với ReactJS

https://www.youtube.com/watch?v=l6nmysZKHFU

Tạo folder Toolbar để chứa các component như navbar và side drawer

Tạo file Toolbar.js

Thẻ header: là 1 container cho các nội dung giới thiệu hoặc 1 tập hợp navigation link

Thẻ nav: 1 tập hợp navigation link

Giải thích:

div thứ nhất chứa button để mở ra side drawer

div thứ 2 chứa logo

div thứ 3 chứa danh sách các nav link

Ta muốn Logo và các nav-link tách nhau ra, logo nằm bên trái và nav-link nằm bên phải, nên ta sẽ dùng 1 div gọi là “spacer”

Giờ ta muốn không thấy “toolbar-navigation-items” khi ta đang ở mobile

Code:

toolBar.js

```
const toolBar=props => (
      <header className=”toolbar”>
            <nav className=”toolbar__navigation”>
                  <div className=”toolbar__toggle-button”>
                        <DrawerToggleButton
                  click={props.drawerClickHandler}/>
                  <div>
                  <div className=”toolbar__logo”><a href=”#”>LOGO</a></div>
                  <div className=”spacer”/>
                  <div classNane=”toolbar__navigation-items”>
                        <ul>
                              <li><a></a></li>
                              <li><a></a></li>
```

```
                        <li><a></a></li>
                    </ul>
                </div>
            </nav>
        </header>
)
export default toolBar;
```

Vì toolbar có postion là fixed nên các element trong phần main có thể bị che khuất bởi toolbar, nên ta phải thiết lập chiều cao cho nó.

Trong phần main của web, ta sẽ thiết lập: `<main style={{marginTop: 64px}}></main>`

Dấu ngoặc nhọn đầu tiên là để biểu thị 1 giá trị dynamic, luôn thay đổi

Dấu ngoặc nhọn thứ hai là để biểu thị đây là 1 Javascript object

Ta muốn các thành phần trong navbar phải nằm chính giữa theo chiều dọc của navbar, nên ta sẽ muốn dùng tính chất `align-items: center;` của flexbox: `display: flex;`

Ta muốn các <li> nằm kế bên nhau nên ta sẽ thêm `display: flex;` vào

class spacer có `flex: 1;` nghĩa là nó lấy hết tất cả space có thể lấy, và sẽ đẩy toolbar__navigation-items về bên phải

toolBar.css

```
.toolbar{
    position: fixed;
    width: 100%;
    top: 0;
    left: 0;
    background-color: purple;
    height: 56px;
}

.toolbar__navigation{
```

```
      display: flex;
      height: 100%;
      align-items: center;
      padding: 0 1rem;
}

.spacer{
      flex: 1;
}

.toolbar__logo a{
      color: white;
      text-decoration: none;
      font-size: 1.5rem;
      margin-left: 1rem;
}

.toolbar__navigations-items ul{
      list-style: none;
      margin: 0;
      padding: 0;
      display: flex;
}

.toolbar__navigation-items li{
      padding 0 0.5rem;
}

.toolbar__navigation-items a{
      color: white;
```

```
      text-decoration: none;
}

.toolbar__navigation-items a:hover,
.toolbar__navigation-items a:active{
      color: orange;
}

@media (max-width: 768px){
      .toolbar-navigation-items{
            display: none;
      }
}

@media (min-width: 769px){
      .toolbar__toggle-button{
            display: none;
      }
      .toolbar__logo{
            margin-left: 0;
      }
}
```

Ta sẽ tạo ra 1 folder mới là SideDrawer nằm cùng directory với Toolbar folder, và tạo ra 2 file mới trong SideDrawer là DrawerToggleButton.js và SideDrawer.js

DrawerToggleButton.js

```
const drawerToggleButton = props => (
      <button className=”toggle-button” onClick={props.click}>
            <div className=”toggle-button__line”/>
```

```
            <div className=”toggle-button__line”/>
            <div className=”toggle-button__line”/>
      </button>
);
export default drawerToggleButton;
```

Để bỏ đi background ở toggle-button, ta sử dụng `background: transparent;`

Nếu muốn dùng nhiều div trong toggle-button, ta phải setup cho toggle-button: `display: flex;`

Trong toggle-button__line, width 100% sẽ không hoạt động

Thêm `justify-content: space-between;` để ta đảm bảo sẽ có khoảng trắng giữa các toggle-button__line

Ta có thể chuyển từ space-between thành space-around để đảm bảo là không chỉ có space giữa các line, mà còn có space giữa line đầu với trần button, line cuối và sàn button

DrawerToggleButton.css

```
.toggle-button{
      display: flex;
      flex-direction: column;
      justify-content: space-around;
      height: 24px;
      width: 30px;
      background: transparent;
      border: none;
      cursor: pointer;
      padding: 0;
      border-box: box-sizing;
}

.toggle-button:focus{
      outline: none;
}
```

```
.toggle-button__line{
      width: 24px;
      height: 2px;
}
```

Vấn đề là ta đặt SideDrawer ở đâu, ta có thể đặt nó trong Toolbar, nhưng nó thực sự không phải là 1 phần của Toolbar, nên đẹp nhất là ta đặt nó trong App.js

Giờ ta muốn 1 component nào đó để thu lại SideDrawer, và nó cũng không phải là 1 thành phần của SideDrawer, nên ta sẽ tạo 1 thành phần mới là Backdrop

SideDrawer.js

```
const sideDrawer = props => {
      let drawerClasses = 'side-drawer';
      if(props.show){
            drawerClasses='side-drawer open';
      }
      return (
      <nav className={drawerClasses}>
            <ul>
                  <li><a href="/"></a></li>
            </ul>
      </nav>
      );
};
export default sideDrawer;
```

SideDrawer.css

```
.side-drawer{
      height: 100%;
      background: white;
      box-shadow: 2px 0px 5px rgba(0, 0, 0, 0.5);
```

```
      position: fixed;
      top: 0;
      left: 0;
      width: 70%;
      max-width: 400px;
      z-index: 200;
      transform: translateX(-100%);
      transition: transform 0.3s ease-out;
}

.side-drawer ul{
      height: 100%;
      list-style: none;
      display: flex;
      flex-direction: none;
      justify-content:center;
}

.side-drawer.open{
      transform: translateX(0);
}

.side-drawer li{
      margin: 0.5rem 0;
}

.side-drawer a{
      color: purple;
      text-decoration: none;
      font-size: 1.2rem;
```

```
}


.side-drawer a:hover,
.side-drawer a:active{
      color: orange;
}


@media (min-width: 769px){
      .side-drawer{
            display: none;
      }
}
```

Ta thêm `height: 100%;` cho div lớn vì ta muốn SideDrawer có thể đạt 100% chiều cao. Ta cũng có thể set cho SideDrawer là `height: 100vh;` nhưng không phải browser nào cũng hỗ trợ vh

Lý do vì sao dùng callback cho setState: vì setState hoạt động theo hướng bất đồng bộ, nghĩa là ngay sau khi gọi setState, this.state không được update ngay lập tức, nên nếu bạn muốn thực hiện 1 hành động ngay sau khi setState, 1 callback sẽ rất hữu dụng

https://stackoverflow.com/questions/42038590/when-to-use-react-setstate-callback

Ta truyền hàm drawerToggleClickHandler vào <Toolbar/> vì trong Toolbar có drawerButton

App.js

```
class App extends React.Component{
      state = {
            sideDrawerOpen: false,
      }
      drawerToggleClickHandler = () => {
            this.setState((prevState)=>{
                  return {sideDrawerOpen: !prevState.sideDrawerOpen}
```

```
            });
      }
      backdropClickHandler = () => {
            this.setState({sideDrawerOpen: false});
      }
      render(){
            let sideDrawer;
            let backdrop;
            if(this.state.sideDrawerOpen){
                  backdrop=<Backdrop click={this.backdropClickHandler}/>
            }
            return (
                  <div style={{height: 100%}}>
                        <Toolbar
drawerClickHandler={this.drawerToggleClickHandler}/>
                        <SideDrawer show={this.state.sideDrawerOpen}/>
                        {backdrop}
                        <main style={{marginTop: 64px}}>
                              <p>This is page content</p>
                        </main>
                  </div>
            )
      }
}
```

Ở file index.css (file phụ trách css cho cả page), ta cũng muốn body có height là 100%

index.css

```
body{
      margin: 0;
      padding: 0;
```

```
	font-family: sans-serif;
	height: 100%;
}

html{
	height: 100%;
}
```

Tạo folder Backdrop chung directory với SideDrawer

Tạo file Backdrop.js trong folder Backdrop

Backdrop.js

```
const backdrop = props => (
	<div className=”backdrop” onClick={props.click}/>
);
export default backdrop;
```

Backdrop.css

```
.backdrop{
	position: fixed;
	top: 0;
	left: 0;
	width: 100%;
	height: 100%;
	background: rgba(0, 0, 0, 0.3);
	z-index: 100;
}
```

Logic là DrawerToggleButton là mở Sidebar và Backdrop sẽ close sidebar, và ta sẽ kiểm soát điều đó ở App.js, nơi mà ta có thể control tất cả

Giờ ta cần thêm animation cho sideDrawer, và sideDrawer nên luôn luôn present, và ta dùng CSS transition để move nó ra và move nó vào