**웹 시스템 설계 – 14주차 학습 보고서 2 (2020년 12월 16일까지 제출)**

|  |  |  |
| --- | --- | --- |
| 성명 : 유 상 정 | 학과 : 사이버보안학과 | 학번 : 201620641 |

※ 본 학습보고서를 작성함에 있어 다른 학생의 문서로부터 일부 또는 전체를 복사하였습니까?

예( ) 아니오(✓) (복사 하였다면 예에 체크하고 아니라면 아니오에 체크하시오)

**I. Objectives**

* React Router를 활용하여 SPA(Single Page Application)에서 라우팅 경로를 정의할 수 있다.

**II. Exercises**

* 14주차 학습과제 수행 시, React Router의 컴포넌트인 Link, Route, Switch가 활용된 부분의 코드를 복사 및 붙여넣기 하고 설명하시오.

|  |  |
| --- | --- |
| **React Router를 활용한 부분 설명하기** | class App extends React.Component {      render() {          return (              <Router>                  <Header />                  <Switch>                      <Route exact path="/" component={Reviews} />                      <Route exact path="/review/new" component={CreateReview} />                  </Switch>              </Router>          );      }  }  export default App;  class Header extends Component {    render(){      const categoryStyle = {        color:"white",        textDecoration: "none",        padding: "8px"      }      return (        <>        <Navbar collapseOnSelect expand="lg" bg="dark" variant="dark">          <Navbar.Brand href="/">Homework2-201620641</Navbar.Brand>          <Navbar.Toggle aria-controls="responsive-navbar-nav" />          <Navbar.Collapse id="responsive-navbar-nav">            <Nav className="mr-auto">              <Link to="/" style={categoryStyle}>Home</Link>              <Link to="/review/new" style={categoryStyle}>                <button style={{color:"white",backgroundColor:"#343a40",border: 0,outline: 0}}>                  New                </button>              </Link>            </Nav>          </Navbar.Collapse>        </Navbar>        </>      )    }  }  React Router는 React 기반의 웹 어플리케이션에서 선언적으로 구성되는 navigational components의 컬렉션이며 해당 과제에서 활용된 부분은 App.js와 Header.js이다.  App.js에서 React Router의 Router component를 선언하고 Switch component를 하위에 선언해준다. 그리고 switch 내부에 Route Component를 나열하는데, 만약 Router 태그 내 정의된 path로 Link로부터 요청이 오면, Switch 태그는 렌더링 될 때, Route 태그에 내 정의된 path가 match하는 경우를 탐색한다. 그리고, 현재 URL 경로와 path가 매치될 경우 해당 Route 컴포넌트에 지정된 component로 렌더링 한다. 이 때, Route Matcher는 Switch 컴포넌트와 Switch 컴포넌트의 하위 컴포넌트인 Route를 통해 수행된다.  즉, Header.js 내의 Link 태그의 to=”url” 형식으로 routing 요청을 보내서 Reviews Component또는 CreateReview Component로 렌더링되도록 React Router가 활용되고 있다.  class Form extends Component {    constructor(props){      super(props);      this.form = {        movie\_name: React.createRef(),        review\_content: React.createRef(),        rate: React.createRef()      }    }      async submit(){      let movie\_name= this.form.movie\_name.current.value;      let review\_content= this.form.review\_content.current.value;      let rate= this.form.rate.current.value;        await this.props.onSubmit(movie\_name,review\_content,rate);      this.form.movie\_name.current.value = '';      this.form.review\_content.current.value = '';      this.form.rate.current.value = '';      }    cancel = () =>{      this.props.history.goBack();    }    render(){      return (        <div>        <h2>Movie Review</h2>        <br/>        <InputGroup className="mb-3">          <InputGroup.Prepend>            <InputGroup.Text id="basic-addon3">              I'm gonna review            </InputGroup.Text>          </InputGroup.Prepend>          <FormControl            id="movie\_name"            name="movie\_name"            type="text"            ref={this.form.movie\_name}            placeholder="Movie Name" />        </InputGroup>        <InputGroup className="mb-3">          <InputGroup.Prepend>            <InputGroup.Text>My review is</InputGroup.Text>          </InputGroup.Prepend>          <FormControl            id="review\_content"            name="review\_content"            type="text"            placeholder="Review"            ref={this.form.review\_content} />        </InputGroup>        <InputGroup>          <InputGroup.Prepend>            <InputGroup.Text>So, I'll give this move a rating of</InputGroup.Text>          </InputGroup.Prepend>          <FormControl            id="rate"            name="rate"            type="text"            placeholder="Rating"            ref={this.form.rate}          />        </InputGroup>        <br/>        <Button onClick={this.submit.bind(this)}>Ok</Button>        <Button variant="warning" onClick={this.cancel.bind(this)}>Cancel</Button>        </div>      )    }  }  export default withRouter(Form);  React Router의 컴포넌트는 기존 React로 구성되는 SPA와 달리 location이나 history 같은 브라우저의 내장 API와 연동되어 여러 이점을 얻을 수 있다. 그러나 Route 태그 내 렌더링할 컴포넌트로 지정되지 않은 컴포넌트는 withRouter를 쓰지않고는 location, match, history 정보에 접근할 수 없다. 이 때, 과제 코드에서는 Form Component에서 withRouter를 선언해줬다. 그리고 Cancel 버튼을 누르면 this.cancel함수를 띄우고, 여기서 this.props.history.goBack() 함수를 활용하여 뒤로가기 이점을 활용하였다. |