|  |  |  |  |
| --- | --- | --- | --- |
| **Lab No** | 09 | **Reg. No** | 224921 |
| **Student Name** | Muhammad Saad Tariq | **Section** | A |

|  |
| --- |
| App2.js |
| import React from 'react';  import './App2.css';  import Header from './Header';  import Books from './Books';  class App extends React.Component{    render(){      return (      <div className="App">      <Header></Header>      <Books/>      </div>    );    }  }  export default App; |

|  |
| --- |
| Index.js |
| import React from 'react';  import ReactDOM from 'react-dom';  import './index.css';  import App from './cart/App2';  import \* as serviceWorker from './serviceWorker';  ReactDOM.render(<App />, document.getElementById('root'));  // If you want your app to work offline and load faster, you can change  // unregister() to register() below. Note this comes with some pitfalls.  // Learn more about service workers: https://bit.ly/CRA-PWA  serviceWorker.unregister(); |

|  |
| --- |
| Header.js |
| import React from 'react';  const Header = () =>{    return(      <header>      <h1>Book Cart</h1>      </header>    )  }  export default Header; |

|  |
| --- |
| SearchArea.js |
| import React from 'react';  const SearchArea = (props) =>{    return(    <div className="search-area">    <form onSubmit={props.searchBook}action="">    <input onChange={props.handleSearch} type="text"/>    <button type="submit">Search</button>    <select defaultValue="Sort" onChange={props.handleSort}>    <option disabled value="Sort">Sort</option>    <option value="Newest">Newest</option>    <option value="Oldest">Oldest</option>    </select>    </form>    </div>    )  }    export default SearchArea; |

|  |
| --- |
| BookCard.js |
| import React from 'react';  const BookCard = (props) =>{    return(    <div className="card-container">    <img src={props.image} alt=""/>    <div className="desc">    <h2><strong>{props.title} </strong></h2>    <br/>    <a href={props.previewLink}> Add to Cart</a>    <h3><strong>Author: </strong>{props.author}</h3>    <p><strong>Published date: </strong>    {props.published === '0000'? 'Not available' : props.published.substring(0,4)}</p>  <p><strong>Category:  </strong>{props.categories}</p>  <p><strong>Page count:  </strong>{props.pageCount}</p>  <p><strong>average rating:  </strong>{props.averageRating}</p>  <br/>  <p><em>{props.description}</em></p>    </div>    </div>  );  }  export default BookCard; |

|  |
| --- |
| BookList.js |
| import React from 'react';  import BookCard from './BookCard'  const BookList = (props) =>{    return(    <div className="list">    {      props.books.map((book,i)=>{        return <BookCard        key={i}        image={book.volumeInfo.imageLinks.thumbnail}        title={book.volumeInfo.title}        author={book.volumeInfo.authors}        published={book.volumeInfo.publishedDate}        //description={book.volumeInfo.description}        categories = {book.volumeInfo.categories}        pageCount = {book.volumeInfo.pageCount}        averageRating = {book.volumeInfo.averageRating}        previewLink = {book.volumeInfo.previewLink}        />      })    }    </div>    )  }  export default BookList; |

|  |
| --- |
| Books.js |
| import React from 'react';  import SearchArea from './SearchArea'  import BookList from './BookList'  import request from 'superagent';  class Books extends React.Component{    constructor(props){      super(props);      this.state={        books: [],        searchField:'',        sort:''      }    }    searchBook=(e)=>{      e.preventDefault();      request        .get("https://www.googleapis.com/books/v1/volumes")        .query({q:this.state.searchField})        .then((data)=>{          console.log(data);          const cleanData= this.cleanData(data);          this.setState({books: cleanData})        })    }    handleSearch=(e)=>{      this.setState({searchField: e.target.value});    }    handleSort=(e)=>{      this.setState({sort: e.target.value});    }    cleanData=(data)=>{      const cleanedData =data.body.items.map((book)=>{        if(book.volumeInfo.hasOwnProperty('publishedDate')=== false){          book.volumeInfo['publishedDate']= '0000';        }        else if (book.volumeInfo.hasOwnProperty('imageLinks')=== false){          book.volumeInfo['imageLinks']= { thumbnail: 'https://vignette.wikia.nocookie.net/pandorahearts/'};        }        return book;      })      return cleanedData;    }    render(){      const sortedBooks= this.state.books.sort((a,b)=>{        if (this.state.sort==='Newest'){          return parseInt(b.volumeInfo.publishedDate.substring(0,4))- parseInt(a.volumeInfo.publishedDate.substring(0,4))        }        else if (this.state.sort==='Oldest'){          return parseInt(a.volumeInfo.publishedDate.substring(0,4))- parseInt(b.volumeInfo.publishedDate.substring(0,4))        }      })      return (      <div >      <SearchArea      searchBook={this.searchBook}      handleSearch={this.handleSearch}      handleSort={this.handleSort}>      </SearchArea>      <BookList      books={sortedBooks}>      </BookList>      </div>    );    }  }  export default Books; |



