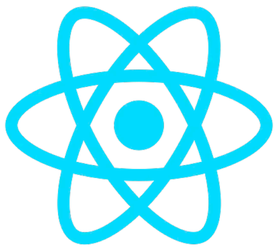
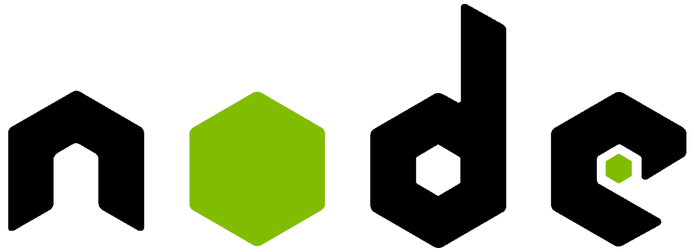
JAVASCRIPT

REACT FRONT-END PART 2

Anirach Mingkhwan

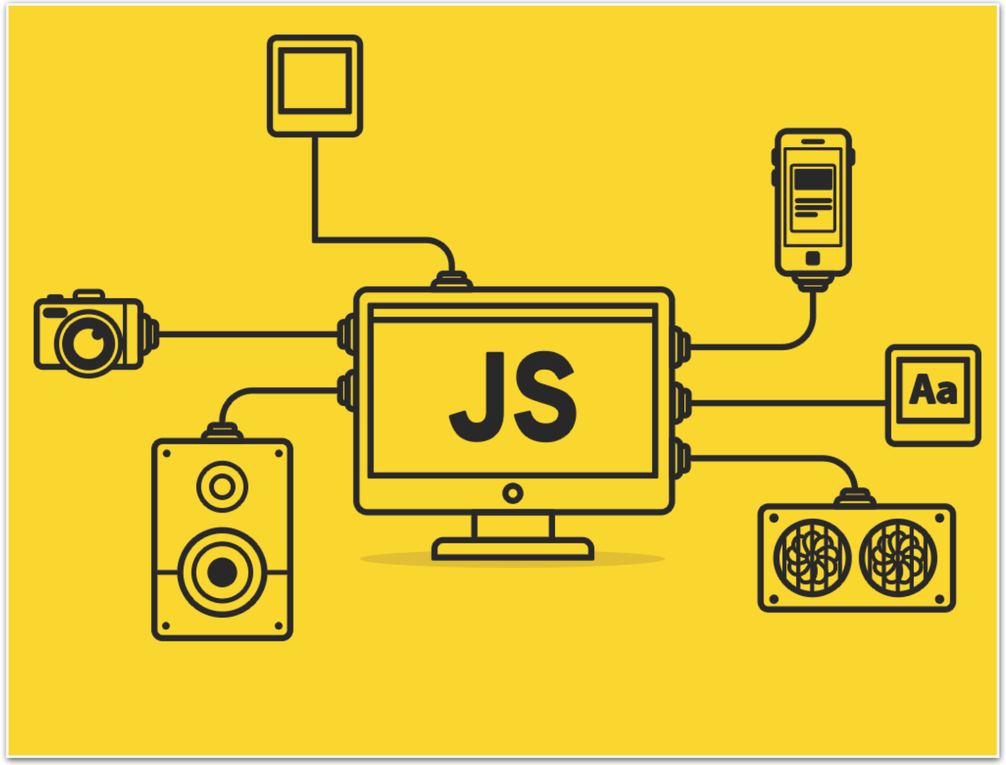
Anirach.m@fitm.kmutnb.ac.th





OUTLINE

- Thinking React II
- Form and OnChange
- React CRUD



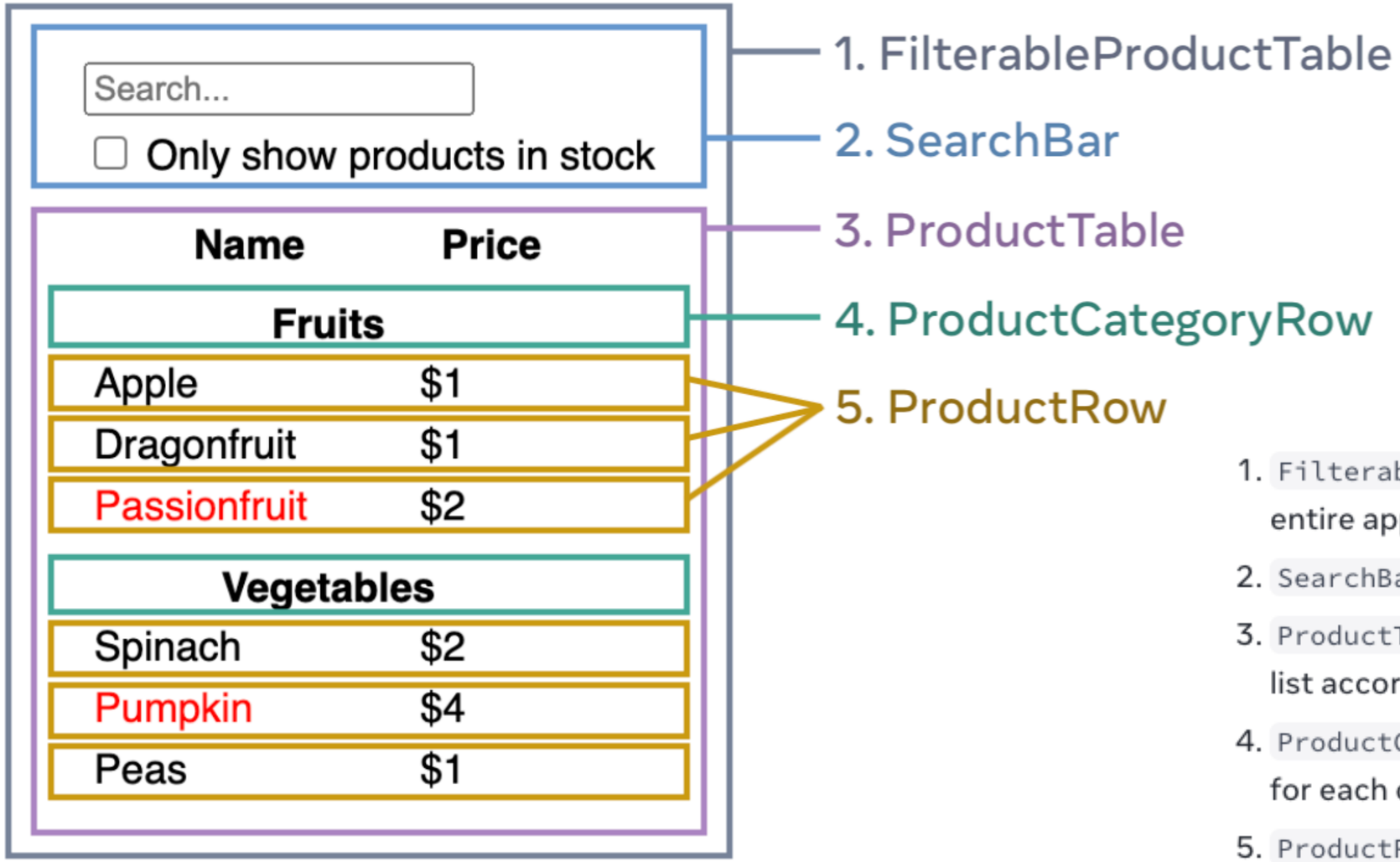
THINKING REACT II

The mockup look like this:

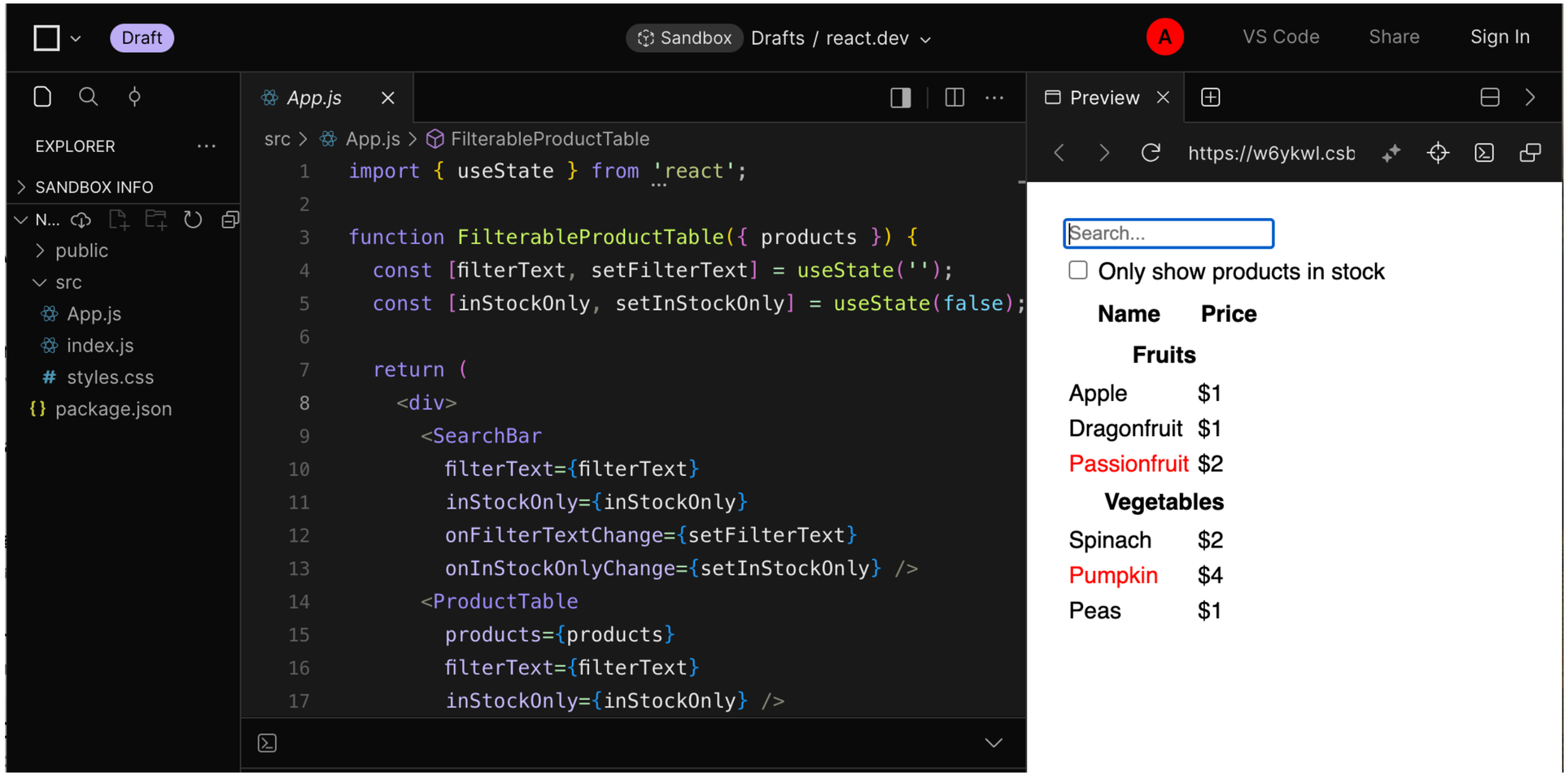

```
[
  { category: "Fruits", price: "$1", stocked: true, name: "Apple" },
  { category: "Fruits", price: "$1", stocked: true, name: "Dragonfruit" },
  { category: "Fruits", price: "$2", stocked: false, name: "Passionfruit" },
  { category: "Vegetables", price: "$2", stocked: true, name: "Spinach" },
  { category: "Vegetables", price: "$4", stocked: false, name: "Pumpkin" },
  { category: "Vegetables", price: "$1", stocked: true, name: "Peas" }
]
```

Step 1: Break the UI into a component hierarchy

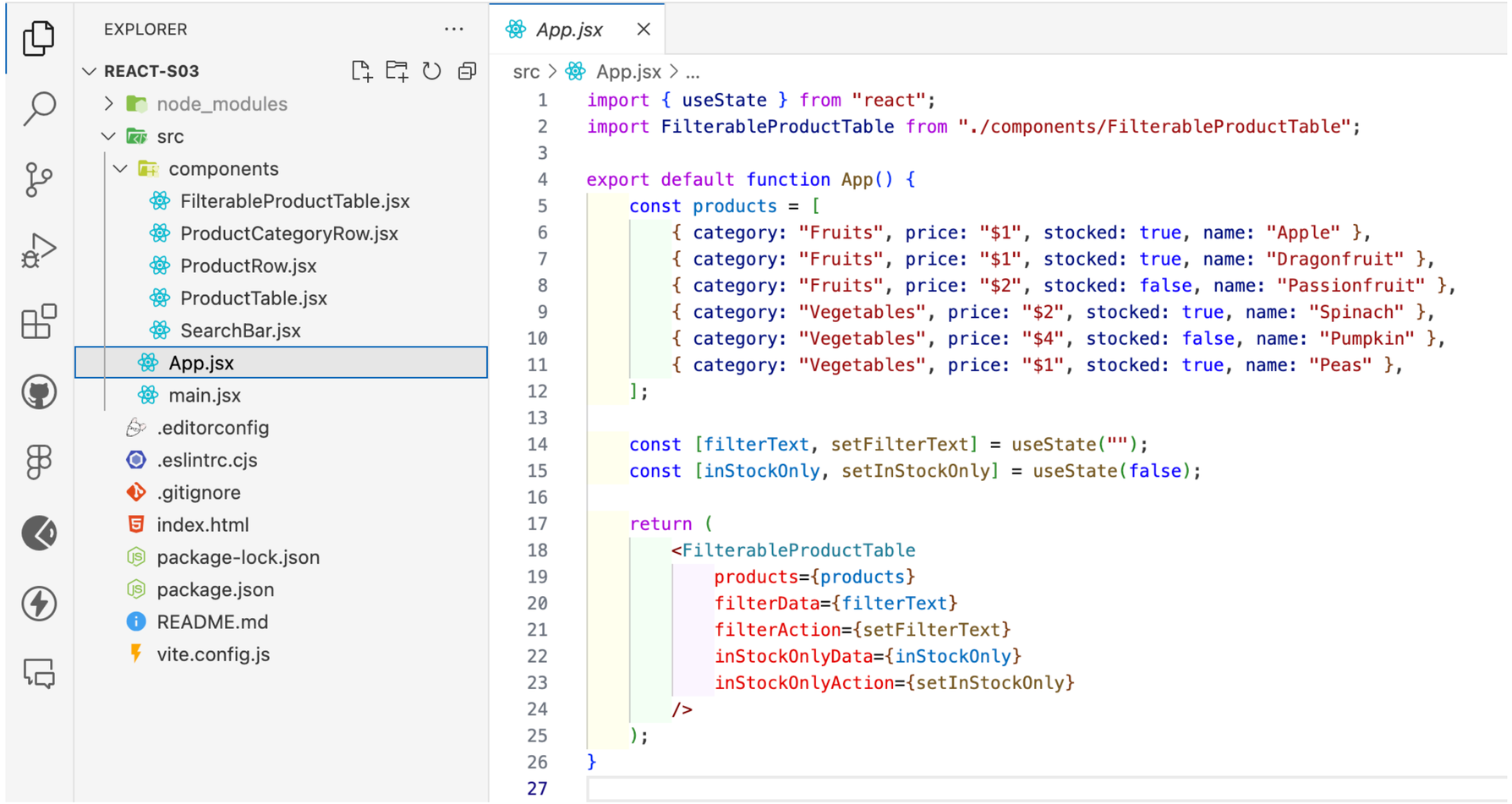
There are five components on this screen:



1. `FilterableProductTable` (grey) contains the entire app.
2. `SearchBar` (blue) receives the user input.
3. `ProductTable` (lavender) displays and filters the list according to the user input.
4. `ProductCategoryRow` (green) displays a heading for each category.
5. `ProductRow` (yellow) displays a row for each product.



https://codesandbox.io/p/sandbox/react-dev-w6ykwl?file=%2Fsrc%2FApp.js&utm_medium=sandpack



https://github.com/Anirach/React-S03.git

FilterableProductTable.jsx M

src > components > FilterableProductTable.jsx > ...

```jsx
import PropTypes from "prop-types";

import SearchBar from "./SearchBar";
import ProductTable from "./ProductTable";

export default function FilterableProductTable(
    { products, filterData, filterAction, inStockOnlyData, inStockOnlyAction }) {
    return (
        <div>
            <SearchBar
                filterText={filterData}
                inStockOnly={inStockOnlyData}
                onFilterTextChange={filterAction}
                onInStockOnlyChange={inStockOnlyAction}
            />
            <ProductTable
                products={products}
                filterText={filterData}
                inStockOnly={inStockOnlyData}
            />
        </div>
    );
}

FilterableProductTable.propTypes = {
    products: PropTypes.object.isRequired,
    filterData: PropTypes.string.isRequired,
    filterAction: PropTypes.func.isRequired,
    inStockOnlyData: PropTypes.bool.isRequired,
    inStockOnlyAction: PropTypes.func.isRequired
};
```

SearchBar.jsx

src > components > SearchBar.jsx > ...

```jsx
import PropTypes from "prop-types";

export default function SearchBar({
    filterText,
    inStockOnly,
    onFilterTextChange,
    onInStockOnlyChange
}) {
    return (
        <form>
            <input
                type="text"
                value={filterText} placeholder="Search..."
                onChange={(e) => onFilterTextChange(e.target.value)} />
            <label>
                <input
                    type="checkbox"
                    checked={inStockOnly}
                    onChange={(e) => onInStockOnlyChange(e.target.checked)} />
                {' '}
                Only show products in stock
            </label>
        </form>
    );
}

SearchBar.propTypes = {
    filterText: PropTypes.string.isRequired,
    inStockOnly: PropTypes.bool.isRequired,
    onFilterTextChange: PropTypes.func.isRequired,
    onInStockOnlyChange: PropTypes.func.isRequired,
};

```

ProductTable.jsx

src > components > ProductTable.jsx > ProductTable > constructor

```jsx
import PropTypes from "prop-types";

import ProductRow from "./ProductRow";
import ProductCategoryRow from "./ProductCategoryRow";

export default function ProductTable({ products, filterText, inStockOnly }) {
    const rows = [];
    let lastCategory = null;

    products.forEach((product) => {
        if (
            product.name.toLowerCase().indexOf(
                filterText.toLowerCase()
            ) === -1
        ) {
            return;
        }
        if (inStockOnly && !product.stocked) {
            return;
        }
        if (product.category !== lastCategory) {
            rows.push(
                <ProductCategoryRow
                    category={product.category}
                    key={product.category} />
            );
        }
        rows.push(
            <ProductRow
                product={product}
                key={product.name} />
        );
        lastCategory = product.category;
    });

```

```
35
36      return (
37          <table>
38              <thead>
39                  <tr>
40                      <th>Name</th>
41                      <th>Price</th>
42                  </tr>
43              </thead>
44              <tbody>{rows}</tbody>
45          </table>
46      );
47  }
48
49  ProductTable.propTypes = {
50      products: PropTypes.object.isRequired,
51      filterText: PropTypes.string.isRequired,
52      inStockOnly: PropTypes.bool.isRequired
53  };
54
```

ProductCategoryRow.jsx

src > components > ProductCategoryRow.jsx > ...

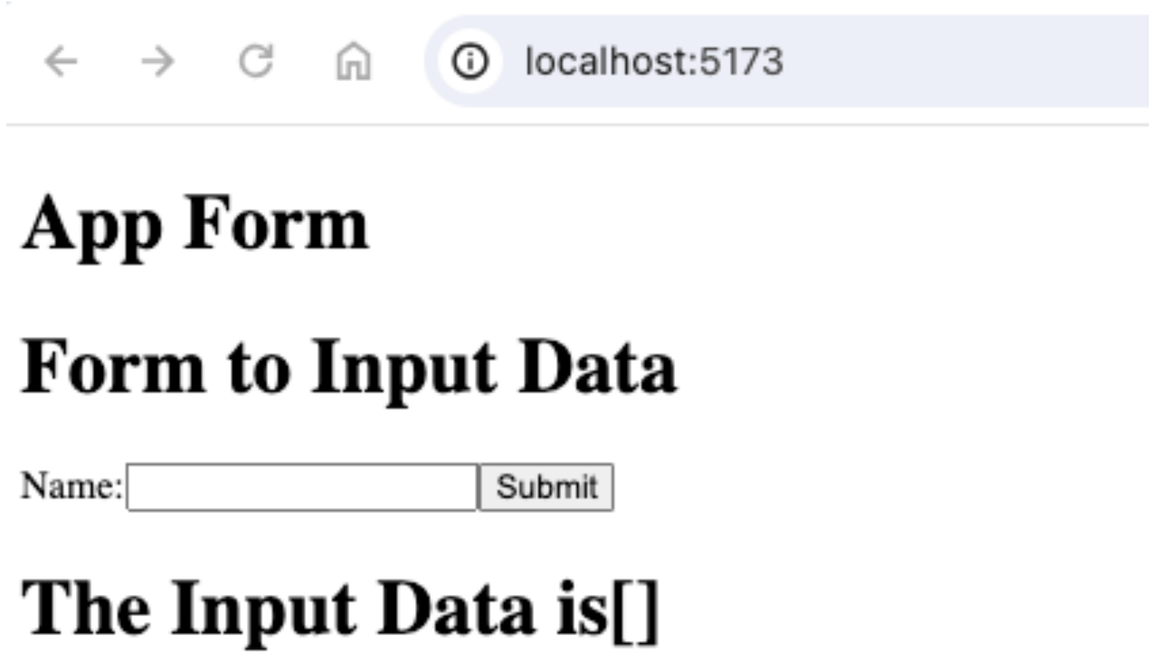
```jsx
import PropTypes from "prop-types";

export default function ProductCategoryRow({ category }) {
    return (
        <tr>
            <th colSpan="2">
                {category}
            </th>
        </tr>
    );
}

ProductCategoryRow.propTypes = {
    category: PropTypes.string.isRequired,
};
```

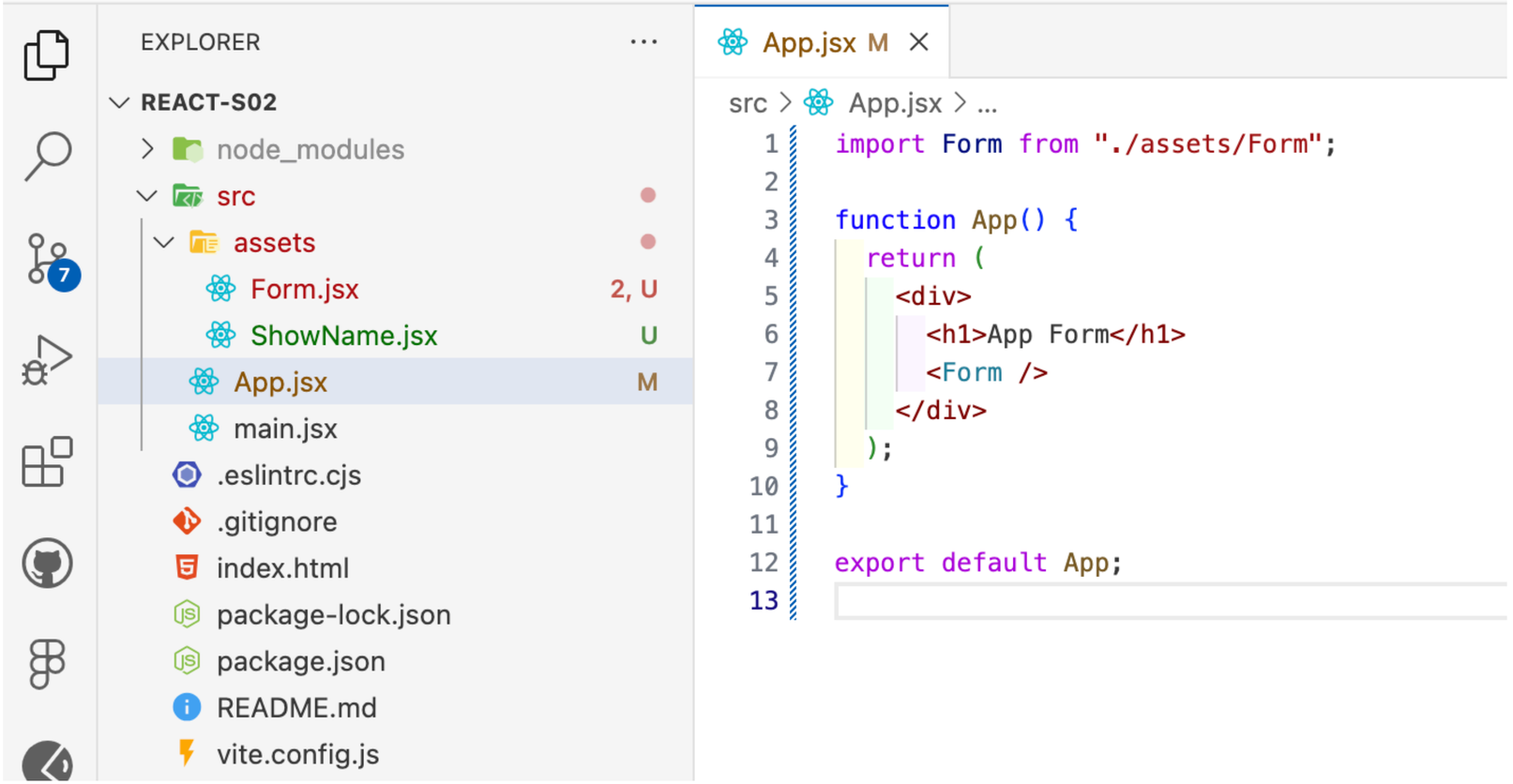
ProductRow.jsx

src > components > ProductRow.jsx > ...

```jsx
import PropTypes from "prop-types";

export default function ProductRow({ product }) {
    const name = product.stocked ? product.name :
        <span style={{ color: 'red' }}>
            {product.name}
        </span>;

    return (
        <tr>
            <td>{name}</td>
            <td>{product.price}</td>
        </tr>
    );
}

ProductRow.propTypes = {
    product: PropTypes.object.isRequired,
};
```

FORM AND ON CHANGE


localhost:5173
App Form
Form to Input Data
Name:
Submit
The Input Data is[]


EXPLORER
REACT-S02
node_modules
src
assets
Form.jsx 2, U
ShowName.jsx U
App.jsx M
main.jsx
.eslintrc.cjs
.gitignore
index.html
package-lock.json
package.json
README.md
vite.config.js
App.jsx M
src > App.jsx > ...
1 import Form from "./assets/Form";
2
3 function App() {
4 return (
5
6 <h1>App Form</h1>
7 <Form />
8
9);
10 }
11
12 export default App;
13

Form.jsx

src > assets > Form.jsx > ...

```jsx
import { useState } from "react";
import ShowName from "./ShowName";

export default function Form() {
  const [name, setName] = useState("");
  const [displayName, setDisplayName] = useState(false);

  const handleChange = (event) => {
    console.log(event);
    setName(event.target.value);
  };

  const handleSubmit = (event) => {
    event.preventDefault();
    alert("A name was submitted: " + name);
    console.log("A name was submitted: " + name);

    name.length >= 5 ? setDisplayName(true) : setDisplayName(false);
  };

  return (
    <div>
      <h1>Form to Input Data</h1>
      <form>
        <label>
          Name:
          <input type="text" value={name} onChange={handleChange} />
        </label>
        <input type="submit" value="Submit" onClick={handleSubmit} />
      </form>
      <ShowName name={name} show={displayName} />
    </div>
  );
}

```

ShowName.jsx M

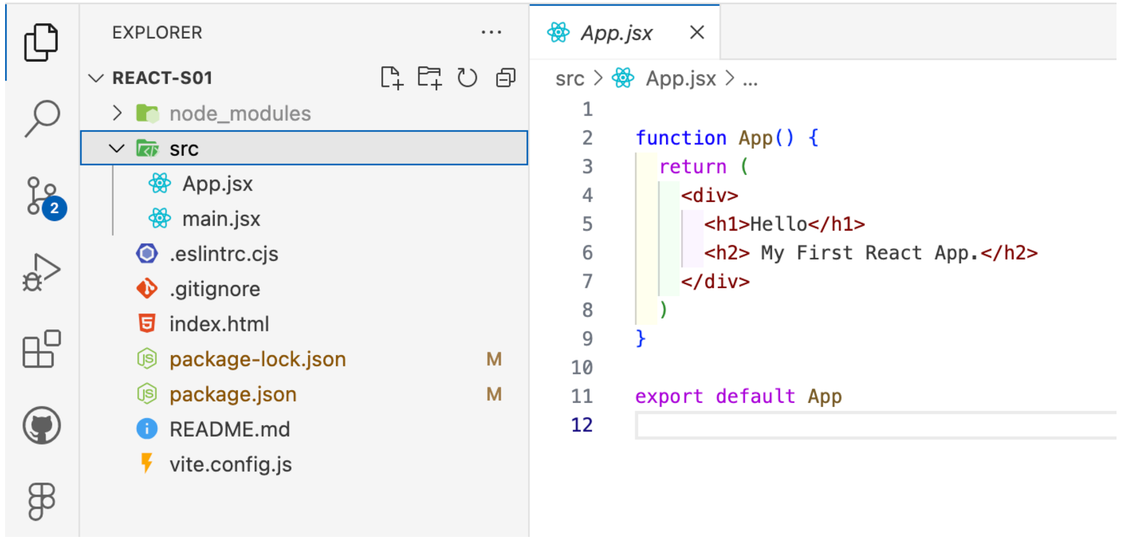
src > assets > ShowName.jsx > ...

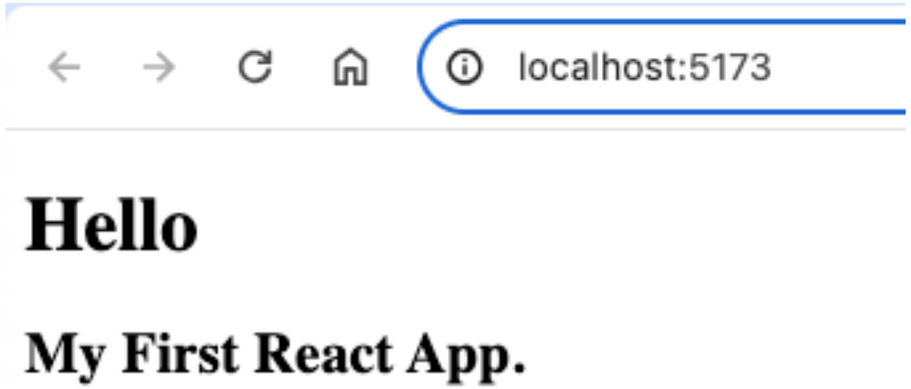
```jsx
import PropTypes from "prop-types";

export default function ShowName({ name, show }) {
  console.log("name:", name);
  console.log("show:", show);

  return <div>{show && <h2>Your name is: {name}</h2>}</div>;
}

ShowName.propTypes = {
  name: PropTypes.string.isRequired,
  show: PropTypes.bool.isRequired,
};
```

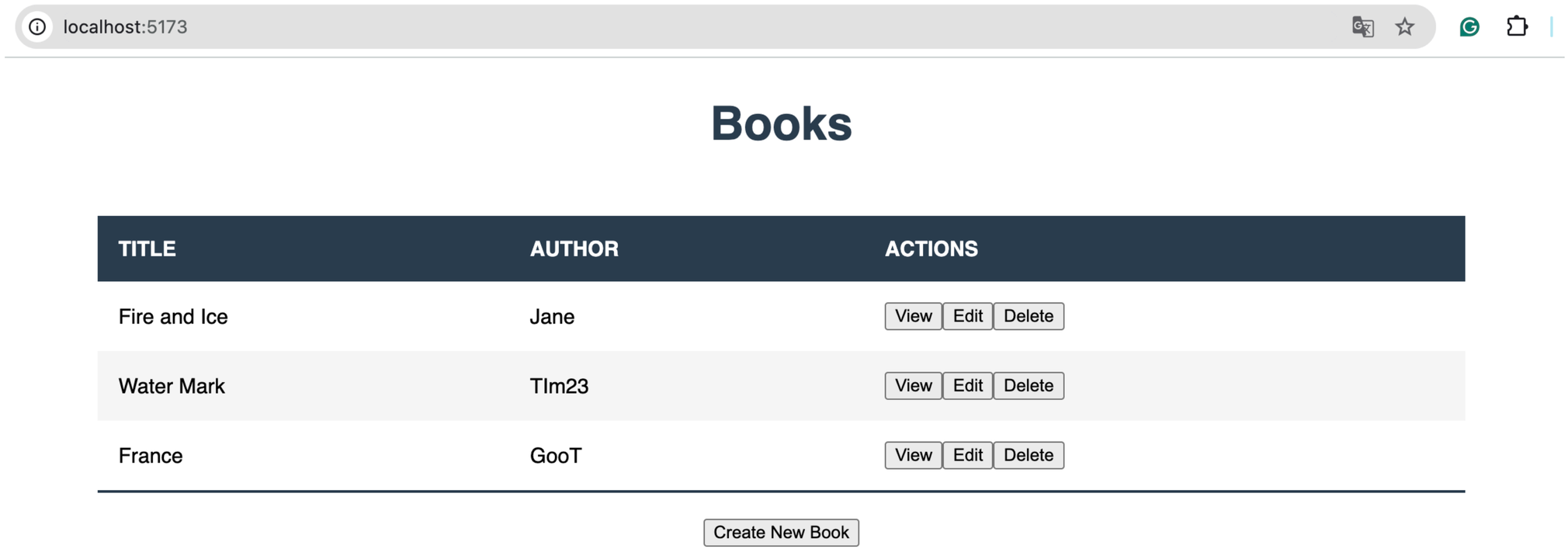

▼ SyntheticBaseEvent {_reactName: 'onChange', _targetInst: null, type: 'change',
bubbles: true
cancelable: false
currentTarget: null
defaultPrevented: false
eventPhase: 3
▶ isDefaultPrevented: ƒ functionThatReturnsFalse()
▶ isPropagationStopped: ƒ functionThatReturnsFalse()
isTrusted: true
▶ nativeEvent: InputEvent {isTrusted: true, data: 'h', isComposing: false, inp
▼ target: input
value: "Anirach"
▶ __reactEvents$m869wjrdwn: Set(1) {'invalid__bubble'}
▶ __reactFiber$m869wjrdwn: FiberNode {tag: 5, key: null, elementType: 'input
▶ __reactProps$m869wjrdwn: {type: 'text', value: 'Anirach', onChange: ƒ}
▶ _valueTracker: {getValue: ƒ, setValue: ƒ, stopTracking: ƒ}
InputEvent

REACT WITH API


node65644-rach-app.proen
node65644-rach-app.proen.app.ruk-com.cloud/books
Pretty-print
[{"id":2,"title":"Fire and Ice","author":"Jane","createdAt":"2024-09-11T03:06:44.740Z","updatedAt":"2024-09-11T03:06:44.740Z"},
{"id":1,"title":"Water Mark","author":"TIm23","createdAt":"2024-09-11T03:06:33.119Z","updatedAt":"2024-09-17T07:42:42.508Z"},
{"id":3,"title":"France","author":"GooT","createdAt":"2024-09-17T07:47:45.889Z","updatedAt":"2024-09-17T07:47:45.889Z"}]


EXPLORER
REACT-S01
node_modules
src
App.jsx
main.jsx
.eslintrc.cjs
.gitignore
index.html
package-lock.json
M
package.json
M
README.md
vite.config.js
App.jsx
src > App.jsx > ...
1
2 function App() {
3 return (
4
5 <h1>Hello</h1>
6 <h2> My First React App.</h2>
7
8)
9 }
10
11 export default App
12


localhost:5173
Hello
My First React App.


localhost:5173
Books
TITLE
AUTHOR
ACTIONS
Fire and Ice
Jane
View
Edit
Delete
Water Mark
TIm23
View
Edit
Delete
France
GooT
View
Edit
Delete
Create New Book

View Book

TITLE	AUTHOR
Fire and Ice	Jane

Edit Delete Back to List

Create Book

Title:

Author:

Save

Back to List

Edit Book

Title:

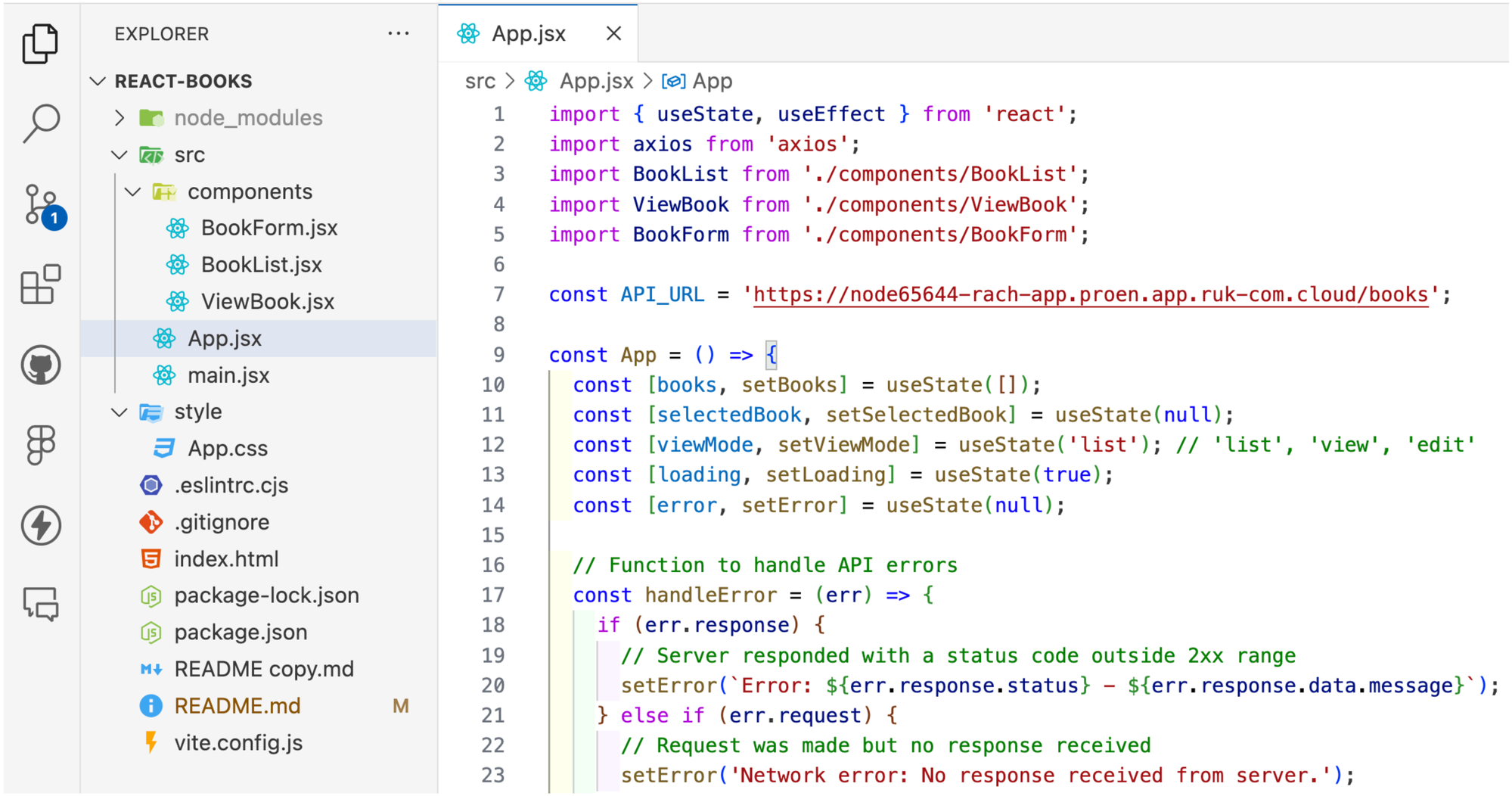
Water Mark

Author:

TIm23

Save

Back to List


EXPLORER
REACT-BOOKS
node_modules
src
components
BookForm.jsx
BookList.jsx
ViewBook.jsx
App.jsx
main.jsx
style
App.css
.eslintrc.cjs
.gitignore
index.html
package-lock.json
package.json
README copy.md
README.md
M
vite.config.js
App.jsx
src > App.jsx > App
import { useState, useEffect } from 'react';
import axios from 'axios';
import BookList from './components/BookList';
import ViewBook from './components/ViewBook';
import BookForm from './components/BookForm';
const API_URL = 'https://node65644-rach-app.proen.app.ruk-com.cloud/books';
const App = () => {
const [books, setBooks] = useState([]);
const [selectedBook, setSelectedBook] = useState(null);
const [viewMode, setViewMode] = useState('list'); // 'list', 'view', 'edit'
const [loading, setLoading] = useState(true);
const [error, setError] = useState(null);
// Function to handle API errors
const handleError = (err) => {
if (err.response) {
// Server responded with a status code outside 2xx range
setError(`Error: ${err.response.status} - ${err.response.data.message}`);
} else if (err.request) {
// Request was made but no response received
setError('Network error: No response received from server.');

src > App.jsx > App > useEffect() callback

```jsx
import { useState, useEffect } from 'react';
import axios from 'axios';
import BookList from './components/BookList';
import ViewBook from './components/ViewBook';
import BookForm from './components/BookForm';

const API_URL = 'https://node65644-rach-app.proen.app.ruk-com.cloud/books';

const App = () => {
  const [books, setBooks] = useState([]);
  const [selectedBook, setSelectedBook] = useState(null);
  const [viewMode, setViewMode] = useState('list'); // 'list', 'view', 'edit'
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  // Function to handle API errors
  const handleError = (err) => {
    if (err.response) {
      // Server responded with a status code outside 2xx range
      setError(`Error: ${err.response.status} - ${err.response.data.message}`);
    } else if (err.request) {
      // Request was made but no response received
      setError('Network error: No response received from server.');
    } else {
      // Something else caused the error
      setError(`Error: ${err.message}`);
    }
  };

```

```
29
30  // Fetch books when component mounts
31  useEffect(() => {
32    // Fetch books from API
33    const fetchBooks = async () => {
34      try {
35        setLoading(true);
36        const response = await axios.get(API_URL);
37        setBooks(response.data);
38        setError(null);  // Clear any previous errors
39        setLoading(false);
40      } catch (err) {
41        handleError(err);
42        setLoading(false);
43      }
44    };
45
46    fetchBooks();
47  }, []);
48
49  const handleView = (id) => {
50    setSelectedBook(books.find((book) => book.id === id));
51    setViewMode('view');
52  };
53
54  const handleEdit = (id) => {
55    setSelectedBook(books.find((book) => book.id === id) || null);
56    setViewMode('edit');
57  };
58
59  const handleDelete = async (id) => {
```

```
59  const handleDelete = async (id) => {
60    try {
61      await axios.delete(`${API_URL}/${id}`);
62      setBooks(books.filter((book) => book.id !== id));
63      setViewMode('list');
64      setError(null);  // Clear any previous errors
65    } catch (err) {
66      handleError(err);
67    }
68  };
69
70  const handleSave = async (book) => {
71    try {
72      if (book.id) {
73        // Update existing book
74        await axios.put(`${API_URL}/${book.id}`, book);
75        setBooks(books.map((b) => (b.id === book.id ? book : b)));
76      } else {
77        // Create new book
78        const response = await axios.post(API_URL, book);
79        setBooks([...books, response.data]);
80      }
81      setViewMode('list');
82      setError(null);  // Clear any previous errors
83    } catch (err) {
84      handleError(err);
85    }
86  };
87
88  const handleBack = () => {
89    setViewMode('list');
```

```
89      setViewMode('list');
90    };
91
92    // Function to handle creating a new book
93    const handleCreateNewBook = () => {
94      setSelectedBook(null); // Reset selectedBook to null for new book creation
95      setViewMode('edit');    // Switch to 'edit' mode to show the form
96    };
97
98    // Rendering loading and error states
99    if (loading) {
100     return <div>Loading...</div>;
101   }
102
103   if (error) {
104     return <div style={{ color: 'red' }}>{error}</div>;
105   }
106
107   return (
108     <div>
109       {viewMode === 'list' && (
110         <div>
111           <BookList
112             books={books}
113             onView={handleView}
114             onEdit={handleEdit}
115             onDelete={handleDelete}
116             onCreate={handleCreateNewBook}  // Pass onCreate prop to BookList
117           />
118         </div>
```

```
118                 </div>
119               )}
120               {viewMode === 'view' && (
121                 <ViewBook
122                   book={selectedBook}
123                   onEdit={handleEdit}
124                   onDelete={handleDelete}
125                   onBack={handleBack}
126                 />
127               )}
128               {viewMode === 'edit' && (
129                 <BookForm book={selectedBook} onSave={handleSave} onBack={handleBack} />
130               )}
131             </div>
132           );
133         };
134
135         export default App;
136
```

src > components > BookList.jsx > BookList

```jsx
import PropTypes from 'prop-types'; // Import PropTypes

const BookList = ({ books, onView, onEdit, onDelete, onCreate }) => {
  return (
    <div className="book-list-container">
      <h1 className="center">Books</h1> {/* Center the title */}
      <table className="book-table"> {/* Use global table styles */}
        <thead>
          <tr>
            <th>TITLE</th>
            <th>AUTHOR</th>
            <th>ACTIONS</th>
          </tr>
        </thead>
        <tbody>
          {books.map((book) => (
            <tr key={book.id}>
              <td>{book.title}</td>
              <td>{book.author}</td>
              <td className="action-buttons">
                <button onClick={() => onView(book.id)}>View</button>
                <button onClick={() => onEdit(book.id)}>Edit</button>
                <button onClick={() => onDelete(book.id)}>Delete</button>
              </td>
            </tr>
          ))}
        </tbody>
      </table>

      {/* Center the Create New Book button */}
      <div className="button-container">
        <button className="create-button" onClick={onCreate}>Create New Book</button>
```

```
32            <button className="create-button" onClick={onCreate}>Create New Book</button>
33          </div>
34        </div>
35      );
36    };
37
38    BookList.propTypes = {
39      books: PropTypes.array.isRequired,
40      onView: PropTypes.func.isRequired,
41      onEdit: PropTypes.func.isRequired,
42      onDelete: PropTypes.func.isRequired,
43      onCreate: PropTypes.func.isRequired, // Add onCreate as a required prop
44    };
45
46    export default BookList;
```

src > components > ViewBook.jsx > ViewBook

```jsx
import PropTypes from 'prop-types';

const ViewBook = ({ book, onEdit, onDelete, onBack }) => {
  const handleDelete = () => {
    onDelete(book.id);
  };

  return (
    <div className="view-book-container">
      <h1 className="center">View Book</h1>
      <table className="book-table">
        <thead>
          <tr>
            <th>TITLE</th>
            <th>AUTHOR</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>{book.title}</td>
            <td>{book.author}</td>
          </tr>
        </tbody>
      </table>
```

```
23            </tbody>
24          </table>
25          <div className="button-container">
26            <button className="edit-button" onClick={() => onEdit(book.id)}>Edit</button>
27            <button className="delete-button" onClick={handleDelete}>Delete</button>
28            <button className="back-button" onClick={onBack}>Back to List</button>
29          </div>
30        </div>
31      );
32    };
33
34    ViewBook.propTypes = {
35      book: PropTypes.object.isRequired,
36      onEdit: PropTypes.func.isRequired,
37      onDelete: PropTypes.func.isRequired,
38      onBack: PropTypes.func.isRequired,
39    };
40
41    export default ViewBook;
```

src > components > BookForm.jsx > BookForm

```jsx
import { useState, useEffect } from 'react';
import PropTypes from 'prop-types';

const BookForm = ({ book, onSave, onBack }) => {
  const [title, setTitle] = useState('');
  const [author, setAuthor] = useState('');

  useEffect(() => {
    if (book) {
      setTitle(book.title);
      setAuthor(book.author);
    }
  }, [book]);

  const handleSubmit = (e) => {
    e.preventDefault();
    onSave({ id: book ? book.id : null, title, author });
  };

  return (
    <div>
      <h1>{book ? 'Edit Book' : 'Create Book'}</h1>
      <form onSubmit={handleSubmit}>
        <div>
          <label>Title:</label>
          <input
            type="text"
            value={title}
            onChange={(e) => setTitle(e.target.value)}
          />
        </div>
        <div>
```

```
30            />
31          </div>
32          <div>
33            <label>Author:</label>
34            <input
35              type="text"
36              value={author}
37              onChange={(e) => setAuthor(e.target.value)}
38            />
39          </div>
40          <button type="submit">Save</button>
41        </form>
42        <div className="button-container">
43          <button onClick={onBack}>Back to List</button>
44        </div>
45
46      </div>
47    );
48  };
49
50  BookForm.propTypes = {
51    book: PropTypes.object,
52    onSave: PropTypes.func.isRequired,
53    onBack: PropTypes.func.isRequired,
54  };
55
56  export default BookForm;
57
```

style > App.css > ...

```css
/* Set font family and size for the whole page */
body {
  font-family: 'Open Sans', sans-serif;
  font-size: 16px;
}

/* Center the main heading */
h1 {
  text-align: center;
  margin-top: 30px;
  margin-bottom: 50px;
  font-size: 36px;
  color: #2C3E50;
}

/* Style the table */
table {
  border-collapse: collapse;
  width: 100%;
}

th, td {
  text-align: left;
  padding: 16px;
}

th {
  background-color: #2C3E50;
  color: white;
  font-weight: 600;
  text-transform: uppercase;
}
```

```
32  }
33
34  tr:nth-child(even) {
35    background-color: #f5f5f5;
36  }
37
38  /* Style the form */
39  form {
40    width: 50%;
41    margin: auto;
42    border: 2px solid #2C3E50;
43    padding: 20px;
44    border-radius: 5px;
45  }
46
47  label {
48    display: block;
49    font-weight: bold;
50    margin-bottom: 10px;
51  }
52
53  input[type="text"] {
54    width: 100%;
55    padding: 8px;
56    box-sizing: border-box;
57    border-radius: 5px;
58    border: none;
59    background-color: #f2f2f2;
60  }
61
62  input[type="submit"] {
```

```
61
62  input[type="submit"] {
63    background-color: #2C3E50;
64    color: white;
65    border: none;
66    padding: 10px 16px;
67    font-size: 16px;
68    cursor: pointer;
69    border-radius: 5px;
70    margin-top: 10px;
71  }
72
73  /* Style the links */
74  a {
75    display: inline-block;
76    margin: 10px;
77    color: #2C3E50;
78    text-decoration: none;
79    font-size: 18px;
80    font-weight: 600;
81    border-bottom: 2px solid transparent;
82    transition: border-bottom-color 0.2s;
83  }
84
85  a:hover {
86    border-bottom-color: #2C3E50;
87  }
88
89  .link-center {
90    text-align: center;
91    margin-top: 20px;
92  }
```

```
92  }
93
94  .link-center a {
95    display: inline-block;
96    margin: 10px;
97    color: #2C3E50;
98    text-decoration: none;
99    font-size: 18px;
100   font-weight: 600;
101   border-bottom: 2px solid transparent;
102   transition: border-bottom-color 0.2s;
103 }
104
105 /* Center the button */
106 .button-container {
107   text-align: center; /* Center the Create New Book button */
108   margin-top: 20px;
109 }
110
111 .link-center a:hover {
112   border-bottom-color: #2C3E50;
113 }
114
115 /* Style the table */
116 .book-table {
117   border-collapse: collapse;
118   width: 75%;
119   margin: 0 auto; /* center the table */
120   margin-bottom: 20px;
121   border-bottom: 2px solid #2C3E50;
122 }
```

```
122   }
123
124   th, td {
125     text-align: left;
126     padding: 16px;
127   }
128
```

Books

TITLE	AUTHOR	ACTIONS
Fire and Ice	Jane	View Edit Delete
Water Mark	TIm23	View Edit Delete
France	GooT	View Edit Delete

Create New Book

THANK YOU

- React Introduction
- Fundamentals of React
- Create React Project
- React props/state
- Thinking React