Exercise 1:

import React from 'react';

function App() {

// Bước 1: Khai báo object employee

const employee = { name: "John Doe", age: 30, department: "IT" };

// Bước 2: Dùng object destructuring

const { name, age, department } = employee;

// Bước 3: Trả về JSX với thông tin đã destructure

return (

<div>

<h1>{name}</h1>

<p>Age: {age}</p>

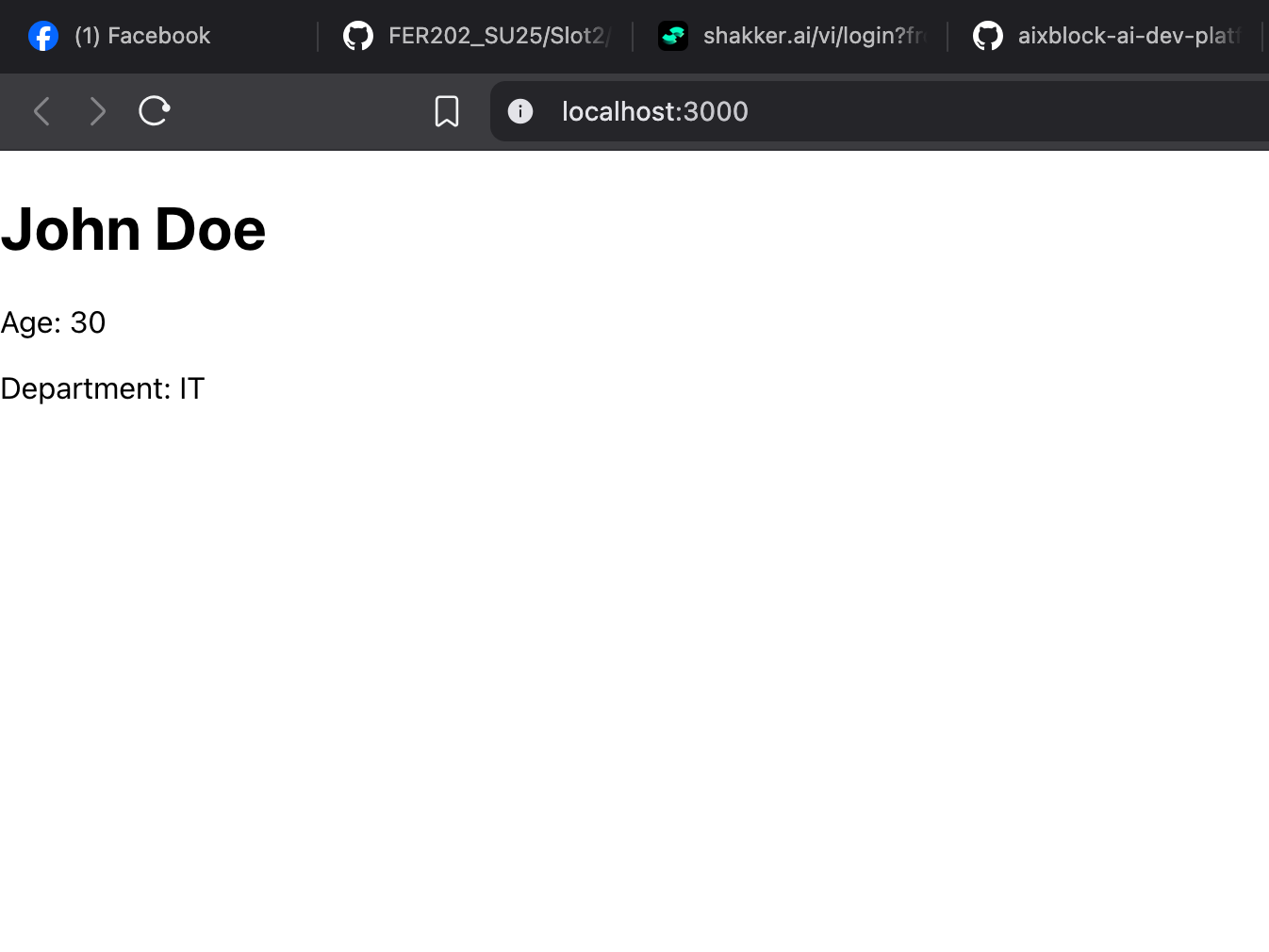
<p>Department: {department}</p>

</div>

);

}

export default App;

  
Exercise 2:

import React from 'react';

function App() {

// Danh sách nhân viên

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

return (

<div>

<h1>Employee List</h1>

<ul>

{employees.map((employee, index) => (

<li key={employee.id || index}>

{employee.name} - {employee.department}

</li>

))}

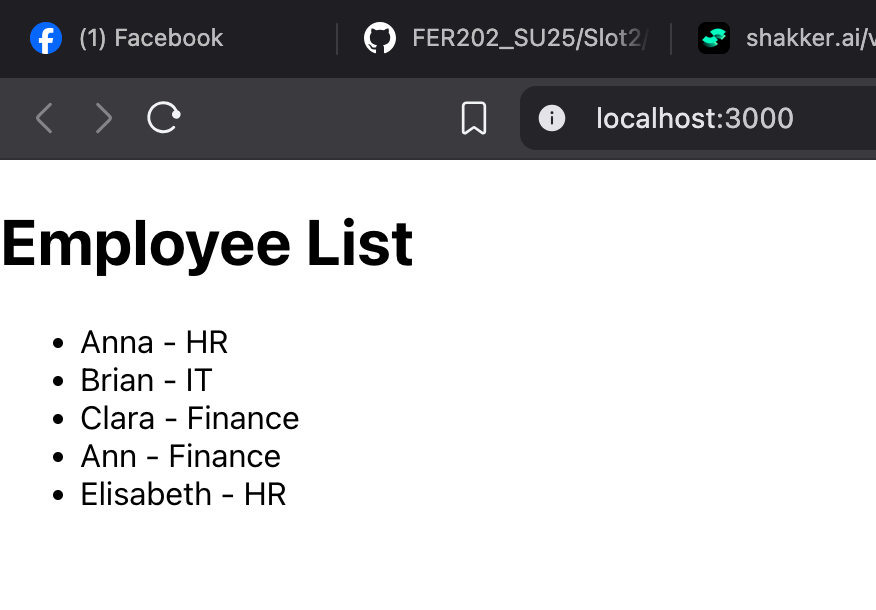
</ul>

</div>

);

}

export default App;



Exercise 3:

import React from 'react';

function App() {

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

return (

<div style={{ padding: "20px" }}>

<h1>Employee Table</h1>

<table border="1" cellPadding="10" cellSpacing="0">

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Department</th>

</tr>

</thead>

<tbody>

{employees.map((employee, index) => (

<tr key={employee.id || index}>

<td>{employee.id || index + 1}</td>

<td>{employee.name}</td>

<td>{employee.department}</td>

</tr>

))}

</tbody>

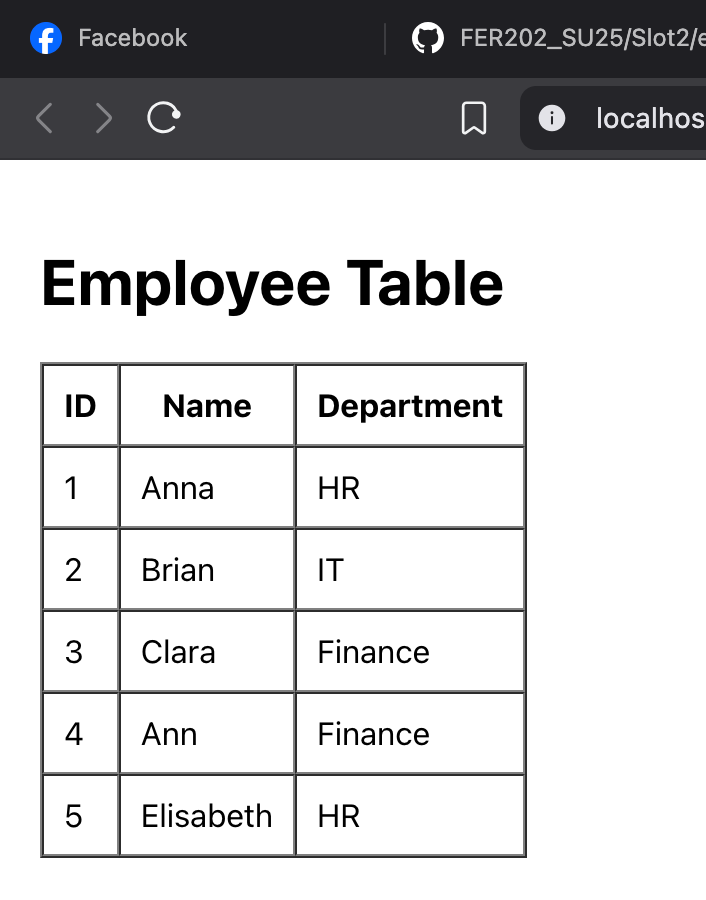
</table>

</div>

);

}

export default App;

  
Exercise 4:

import React from 'react';

function App() {

// Arrow function sử dụng rest parameters

const averageAge = (...ages) => {

if (ages.length === 0) return 0;

const total = ages.reduce((sum, age) => sum + age, 0);

return (total / ages.length).toFixed(2); // Làm tròn 2 chữ số

};

// Gọi hàm với danh sách tuổi

const avg = averageAge(50, 40, 19, 22, 16);

return (

<div style={{ padding: "20px" }}>

<h1>Average Age Calculator</h1>

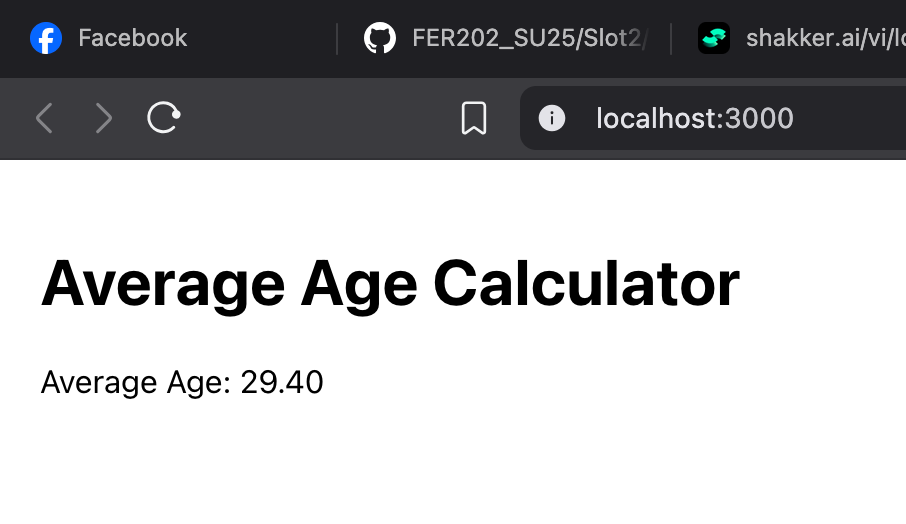
<p>Average Age: {avg}</p>

</div>

);

}

export default App;

  
Exercise 5:

import React from 'react';

function App() {

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

return (

<div style={{ padding: "20px" }}>

<h1>Employee Dropdown</h1>

<label htmlFor="employeeSelect">Choose an employee: </label>

<select id="employeeSelect">

{employees.map((employee, index) => (

<option key={employee.id || index} value={employee.name}>

{employee.name}

</option>

))}

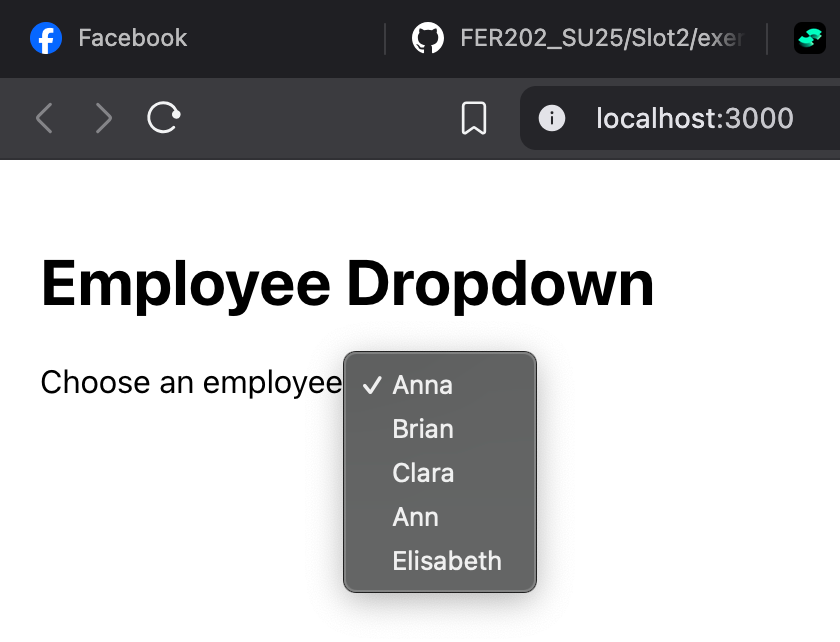
</select>

</div>

);

}

export default App;



Exercise 6:

import React from 'react';

function App() {

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

// Lọc nhân viên phòng IT

const itEmployees = employees.filter(emp => emp.department === "IT");

return (

<div style={{ padding: "20px" }}>

<h1>IT Department Employees</h1>

<ul>

{itEmployees.map((employee, index) => (

<li key={employee.id || index}>{employee.name}</li>

))}

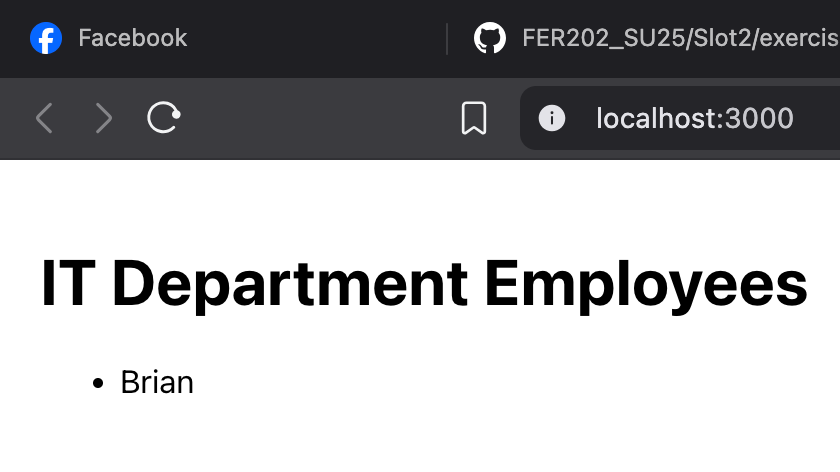
</ul>

</div>

);

}

export default App;



Exercise 7:

import React from 'react';

function App() {

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

// Tạo bản sao mảng và sắp xếp

const sortedEmployees = [...employees].sort((a, b) => {

const deptCompare = a.department.localeCompare(b.department);

if (deptCompare !== 0) {

return deptCompare; // khác phòng -> dùng kết quả so sánh phòng ban

}

return a.name.localeCompare(b.name); // cùng phòng -> so sánh tên

});

return (

<div style={{ padding: "20px" }}>

<h1>Sorted Employees</h1>

<ul>

{sortedEmployees.map((employee, index) => (

<li key={employee.id || index}>

{employee.name} - {employee.department}

</li>

))}

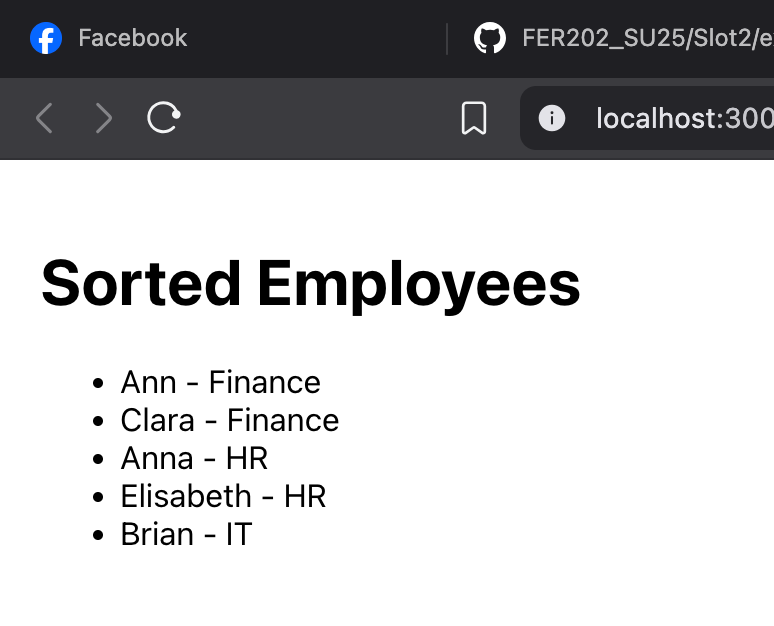
</ul>

</div>

);

}

export default App;



Exercise 8:

import React from 'react';

function App() {

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

// Nhóm nhân viên theo phòng ban

const groupedByDepartment = employees.reduce((grouped, emp) => {

const dept = emp.department;

if (!grouped[dept]) {

grouped[dept] = [];

}

grouped[dept].push(emp);

return grouped;

}, {});

return (

<div style={{ padding: "20px" }}>

<h1>Employees Grouped by Department</h1>

{Object.entries(groupedByDepartment).map(([department, empList]) => (

<div key={department}>

<h3>{department}</h3>

<ul>

{empList.map((emp, index) => (

<li key={emp.id || index}>{emp.name}</li>

))}

</ul>

</div>

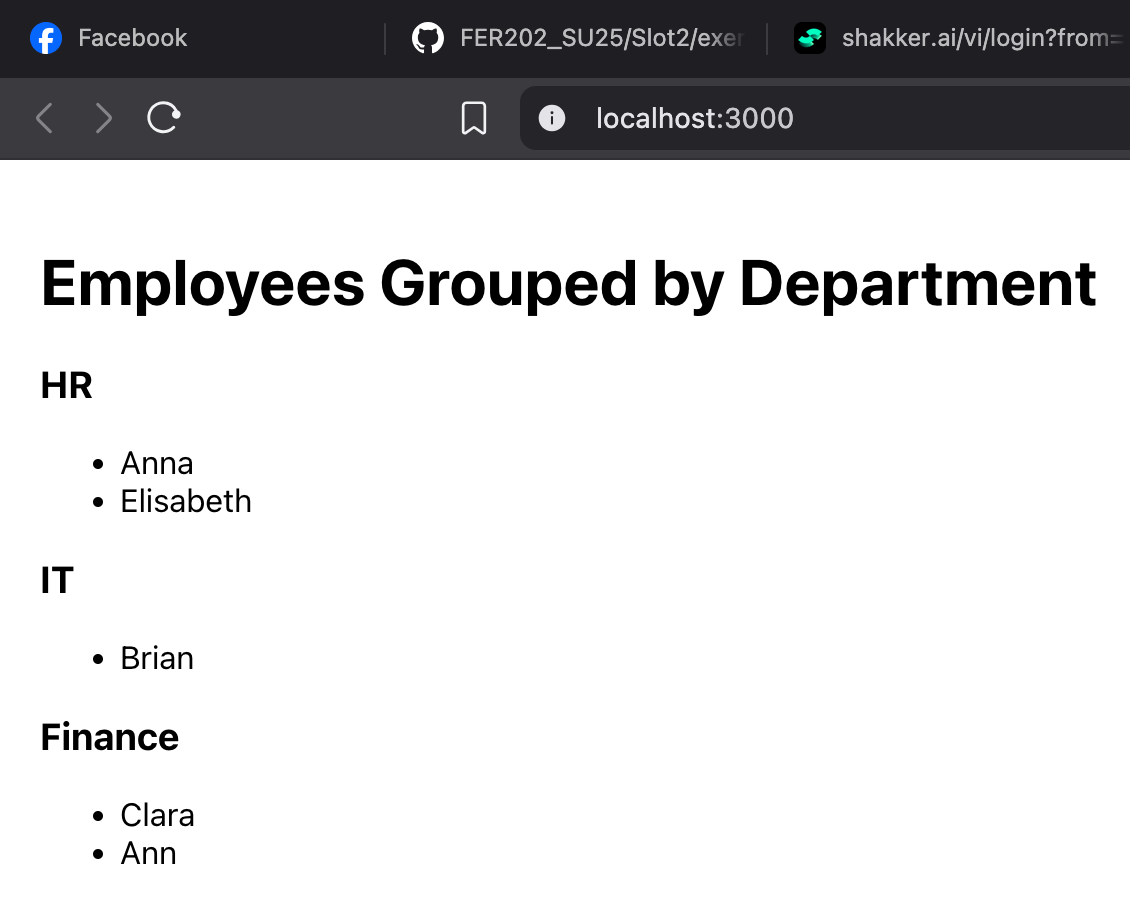
))}

</div>

);

}

export default App;

  
Exercise 9:

import React from 'react';

function App() {

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

// Kiểm tra nếu có nhân viên là thiếu niên (10 <= age <= 20)

const isTeenager = employees.some(e => e.age >= 10 && e.age <= 20);

return (

<div style={{ padding: "20px" }}>

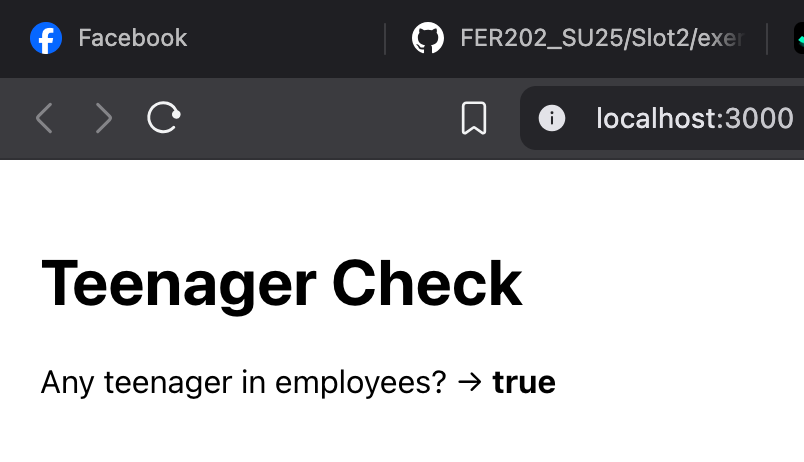
<h1>Teenager Check</h1>

<p>Any teenager in employees? → <strong>{isTeenager.toString()}</strong></p>

</div>

);

}

export default App;  


Exercise 10:

import React, { useState } from 'react';

function App() {

const employees = [

{ id: 1, name: "Anna", department: "HR", age: 50 },

{ id: 2, name: "Brian", department: "IT", age: 40 },

{ id: 3, name: "Clara", department: "Finance", age: 19 },

{ name: "Ann", department: "Finance", age: 22 },

{ name: "Elisabeth", department: "HR", age: 16 }

];

const [searchTerm, setSearchTerm] = useState("");

// Lọc nhân viên dựa trên tên (không phân biệt hoa thường)

const filteredEmployees = employees.filter((emp) =>

emp.name.toLowerCase().includes(searchTerm.toLowerCase())

);

return (

<div style={{ padding: "20px" }}>

<h1>Search Employees</h1>

<input

type="text"

placeholder="Enter name..."

value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)}

style={{ padding: "8px", fontSize: "16px", width: "300px" }}

/>

<ul style={{ marginTop: "20px" }}>

{filteredEmployees.length > 0 ? (

filteredEmployees.map((emp, index) => (

<li key={emp.id || index}>

{emp.name} - {emp.department}

</li>

))

) : (

<li>No employees found</li>

)}

</ul>

</div>

);

}

export default App;

