**Lab 1**

**Using JSX and ES6 in React**

**By traltb@fe.edu.vn**

Create a React application, using jsx and ES6 in React to complete all exercises 1.

**Exercise 1 – Render employee details using object destructuring**

**Objective**:  
Use ES6 object destructuring to extract properties and render them using JSX.

**Given**:

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

**Task**:  
Render the employee's name inside an <h1>, and their age and department inside <p> tags.

import React from 'react';

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

function *Ex1*() {

const { name, age , department} = employee;

return (

<div>

<h1>-----------------------------------Question1-------------------------------------</h1>

<h1>{name}</h1>

<h2>Age: {age}</h2>

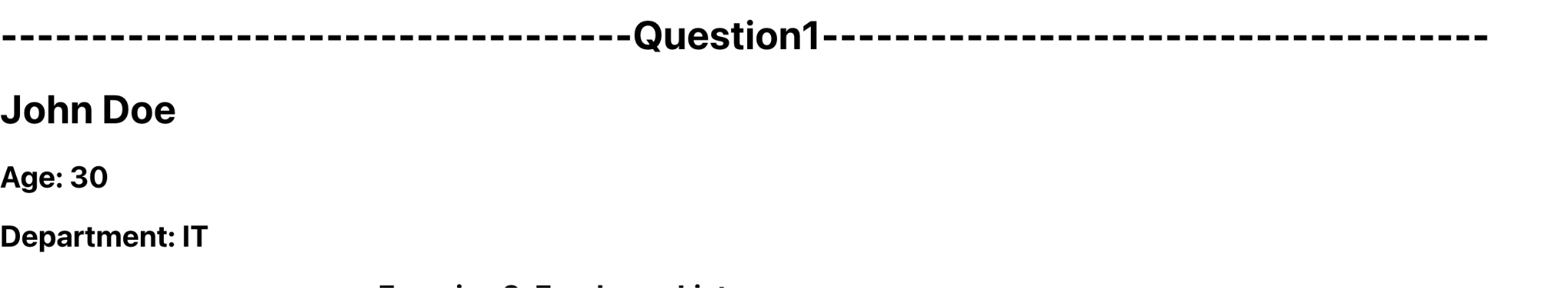
<h2>Department: {department}</h2>

</div>

)

}

export default *Ex1*;



**Exercise 2 – Display a list of employees using map()**

**Objective**:  
Use Array.map() to render a list of employees using <ul> and <li>.

**Given**:

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 }

];

**Task**:  
Render a list showing each employee’s name and department using <ul> and <li>. Use index or id as the key.

import React from 'react';

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 *Ex2* = () => {

return (

<div>

<h2>Exercise 2: Employee List</h2>

<ul >

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

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

{employee.name} - {employee.department}

</li>

))}

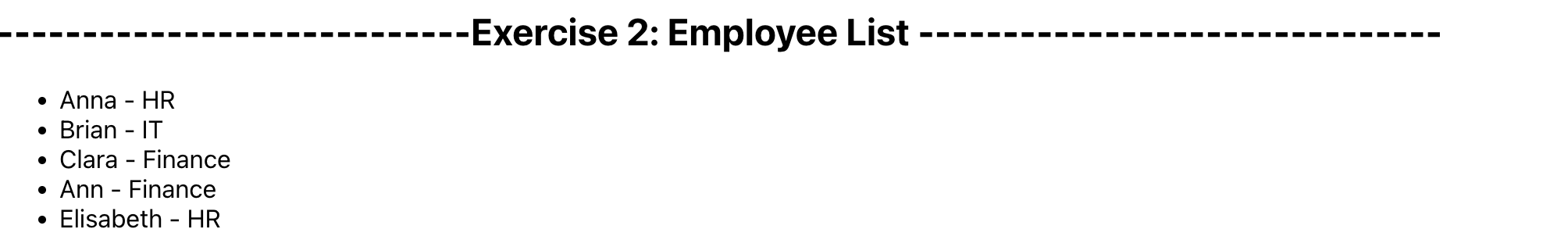
</ul>

</div>

);

};

export default *Ex2*;



**Exercise 3 – Render a table of employees**

**Objective**:  
Use JSX to display employees in a structured table format with headers.

**Task**:  
Render a <table> showing the following columns: ID (or index if missing), Name, Department. Use <thead> and <tbody>.

import React from 'react';

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 Ex3 = () => {

return (

<div>

<h2>----------------------------Exercise 3: Employee Table----------------------------</h2>

<table>

<thead > {/\*hàng tiêu đề\*/}

<tr>

<th>ID</th>

<th>Name</th>

<th>Department</th>

</tr>

</thead>

<tbody> {/\* Hàng chứa data\*/}

{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>

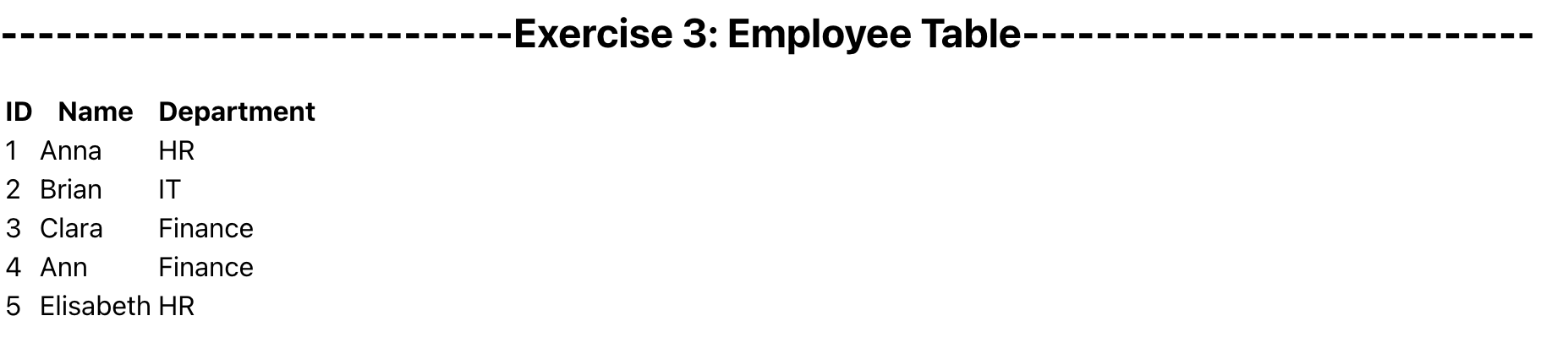
</table>

</div>

);

};

export default Ex3;



**Exercise 4 – Calculate average age using rest parameters**

**Objective**:  
Use ES6 rest parameters to collect a dynamic list of ages and calculate their average.

**Task**:  
Create an arrow function averageAge(...ages) that returns the average, and render the result inside a <p>.

import React from "react";

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 averageAge = (ages) => {

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

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

return sum / ages.length;

};

const Ex4 = () => {

const ages = employees.map((employee) => employee.age); // dùng hàm call back lấy các cột age từ employees

const avgAge = averageAge(ages);

return (

<div>

<h2>

----------------------------Exercise 4: Employee Table with Average

Age----------------------------

</h2>

<table>

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Department</th>

<th>Age</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>

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

</tr>

))}

</tbody>

</table>

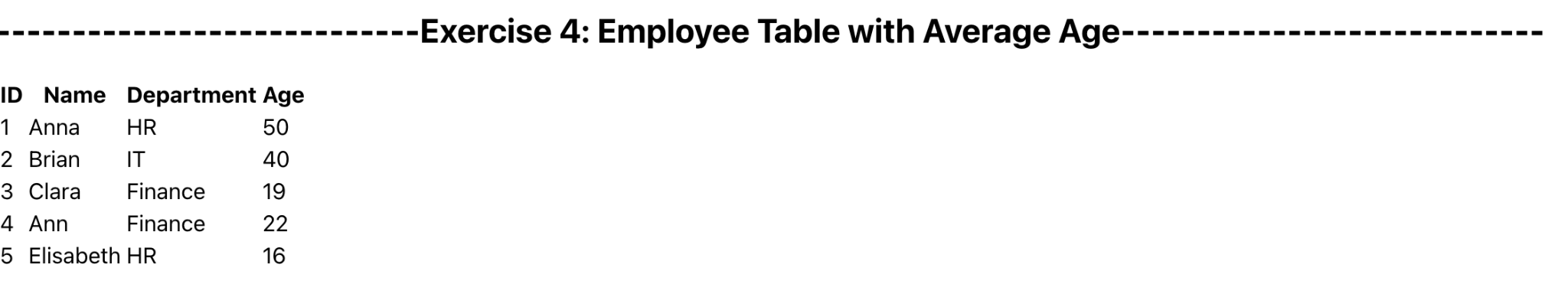
<p>Average Age: {avgAge.toFixed(2)}</p>

</div>

);

};

export default Ex4;



**Exercise 5 – Render a dropdown menu of employee names**

**Objective**:  
Use JSX and <select> to create a dropdown from the employee list.

**Task**:  
Map over the employees array and render each name as an <option> in a <select> dropdown.

import React from "react";

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 Ex5 = () => {

return (

<div>

<h2>Exercise 5: Employee Dropdown Menu</h2>

<select>

<option value="" disabled selected>

Select an employee

</option>

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

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

{employee.name}

</option>

))}

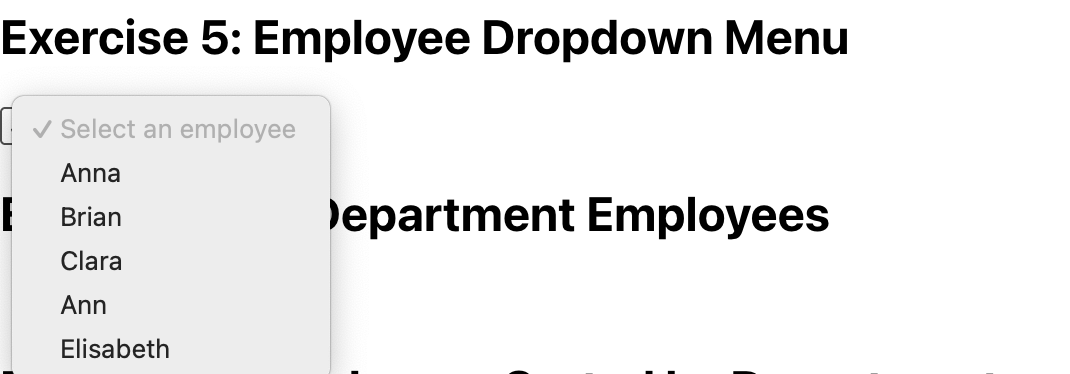
</select>

</div>

);

};

export default Ex5;



**Exercise 6 – Filter and display IT department employees**

**Objective**:  
Use Array.filter() and Array.map() to filter employees by department.

**Task**:  
Render only employees from the "IT" department inside a <ul> list.

import React from "react";

const Ex6 = () => {

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 itEmployees = employees.filter(

(employee) => employee.department === "IT"

);

return (

<div>

<h2>Exercise 6: IT Department Employees</h2>

{itEmployees.length === 0 ? (

<p>No employees in IT department</p>

) : (

<ul>

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

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

{employee.name} - {employee.department}

</li>

))}

</ul>

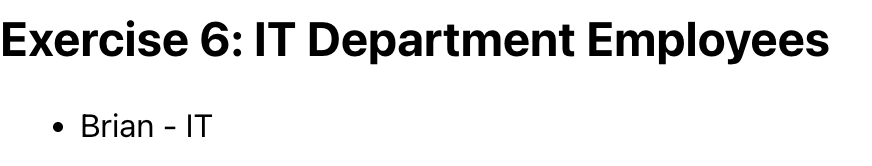
)}

</div>

);

};

export default Ex6;



**Exercise 7 – Sort employees by department, then by name3**

**Objective**:  
Sort the employee list first by department alphabetically, then by name if departments match.

**Task**:  
Display the sorted list inside a <ul>. Use Array.sort() and localeCompare() for string sorting.

import React from "react";

const Ex7 = () => {

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 },

];

if (!employees || !Array.isArray(employees)) {

return <div>Invalid employee data</div>;

}

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

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

if (deptCompare !== 0) return deptCompare;

return a.name.localeCompare(b.name);

});

return (

<div>

<h2>Exercise 7: Employees Sorted by Department and Name</h2>

{sortedEmployees.length === 0 ? (

<p>No employees to display</p>

) : (

<ul>

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

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

{employee.name} - {employee.department}

</li>

))}

</ul>

)}

</div>

);

};

export default Ex7;



**Exercise 8 – Group employees by department**

**Objective**:  
Use ES6 and array methods to group employees by department.

**Task**:  
Render department headings (e.g., <h3>HR</h3>) followed by a list of employees in each department using <ul>.

import React from "react";

const Ex8 = () => {

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 },

];

if (!employees || !Array.isArray(employees)) {

return <div>Invalid employee data</div>;

}

// Nhóm các đối tượng theo department

const groupedEmployees = employees.reduce((acc, employee) => {

const dept = employee.department;

acc[dept] = acc[dept] || [];

acc[dept].push(employee);

return acc;

}, {});

return (

<div>

<h2>Exercise 8: Employees Grouped by Department</h2>

{Object.keys(groupedEmployees).length === 0 ? (

<p>No employees to display</p>

) : (

Object.entries(groupedEmployees).map(([dept, empList]) => (

<div key={dept}>

<h3>{dept}</h3>

<ul>

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

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

{employee.name} - {employee.department}

</li>

))}

</ul>

</div>

))

)}

</div>

);

};

export default Ex8;



**Exercise 9 – Check if any employee is a teenager**

**Objective**:  
Use Array.some() to check if any employee's age is between 10 and 20.

**Task**:  
Display the result (true or false) inside a <p> tag. You can use:

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

import React from "react";

const Ex9 = () => {

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 },

];

if (!employees || !Array.isArray(employees)) {

return <div>Invalid employee data</div>;

}

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

return (

<div>

<h2>Exercise 9: Check for Teenager Employees</h2>

<p>Is there any teenager (age 10-20)? {isTeenager.toString()}</p>

</div>

);

};

export default Ex9;



**Exercise 10 – Search for an employee by name**

**Objective**:  
Use Array.find() or filter() to search the array by employee name (case-insensitive).

**Task**:  
Create a simple search bar using <input type="text" />, and display matching employees dynamically as the user types.

import React, { useState } from "react";

const Ex10 = () => {

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("");

if (!employees || !Array.isArray(employees)) {

return <div>Invalid employee data</div>;

}

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

employee.name.toLowerCase().includes(searchTerm.toLowerCase())

);

return (

<div>

<h2>Exercise 10: Search Employees by Name</h2>

<input

type="text"

placeholder="Search by name..."

value={searchTerm}

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

/>

{filteredEmployees.length === 0 ? (

<p>No employees found</p>

) : (

<ul>

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

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

{employee.name} - {employee.department} (Age: {employee.age})

</li>

))}

</ul>

)}

</div>

);

};

export default Ex10;

