**Job Simulation: Employee Dashboard Simulation**

**Project Title:**

Implementation of an Employee Management Dashboard

**Role:**

Front-End Development Inter

**Technology Stack:**

HTML, CSS, JavaScript

**Objective:**

To design and implement a fully responsive dashboard that displays, filters, and analyzes employee data. The project introduces practical experience with JavaScript's higher-order functions such as map(), filter(), reduce(), find(), and forEach(). You will create a basic UI using HTML and CSS, and integrate interactivity using modular JavaScript code.

**Task Overview:**

You are required to build a dynamic, user-friendly dashboard that allows users to:

* View a table of employee data
* Filter employees by department
* Search employees by name
* Calculate and display average salary
* Convert all employee names to uppercase
* View summarized results in a separate panel

This task simulates real-world employee data management with visual interactivity.

**Task Requirements:**

**1. Functionality:**

* Create a JavaScript array of employee objects with these properties:
  + name, age, department, role, salary
* Use forEach() to dynamically generate table rows for each employee
* Use map() to convert all employee names to uppercase when a button is clicked, then update the table
* Use filter() to filter employees based on selected department from a dropdown menu
* Use reduce() to calculate and display the **average salary** in the result panel
* Use find() to search and display full details of an employee when a name is entered in the search field

**2. User Interface (UI):**

Build a responsive layout with the following sections:

* **Header:**  
  Title: *Employee Management Dashboard*
* **Controls Panel:**
  + Search bar to input employee names
  + Dropdown to filter by department
  + Buttons for:
    - Convert Names to Uppercase
    - Calculate Average Salary
* **Employee Table:**  
  Columns: Name, Age, Department, Role, Salary
* **Result Panel:**  
  Displays:
  + Average salary
  + Searched employee’s full information

**CSS Styling Requirements:**

* Use consistent padding, spacing, and modern color schemes
* Apply background color or card-style boxes to different sections
* Make the employee table scrollable and fully responsive
* Highlight table rows on hover with smooth transitions

**3. Code Structure:**

* Use **separate files**:
  + index.html – HTML layout and structure
  + style.css – CSS styles for layout, color, and responsiveness
  + script.js – JavaScript logic for all data operations and UI updates
* Create **modular JavaScript functions**, such as:
  + filterByDepartment()
  + calculateAverageSalary()
  + convertNamesToUpperCase()
  + searchEmployeeByName()
  + displayEmployeeTable()
* Bind UI elements and events using document.querySelector() and related DOM methods
* Add **2–3 line comments** in script.js explaining how each array method (map, filter, reduce, find, forEach) is being used

**Bonus (Optional Features):**

* Add a **light/dark mode toggle** using CSS class switching
* **Validate search input** to display “No match found” when the entered name doesn’t exist
* Animate row highlighting using transition and :hover
* Show the **total number of employees** currently displayed in the table

**Deliverables:**

Submit a project folder containing:

* index.html – Markup for the dashboard layout
* style.css – Styles for layout, responsive design, and theming
* script.js – JavaScript for managing data and updating the UI

Make sure it includes inline comments for every higher-order function used.

**Learning Outcomes:**

By completing this project, you will gain practical experience in:

* Using JavaScript’s higher-order functions to process and manipulate arrays
* Creating and updating DOM elements dynamically
* Designing modular and maintainable JavaScript code
* Building responsive and styled layouts using HTML and CSS
* Handling and validating user inputs to enhance interactivity



