Node.js

Student Name: Akshit Sharma **UID:23BCS10929**
Branch: BE CSE **Section/Group: 622-B**
Semester: 5
Subject Name: Full Stack - I **Subject Code: 23CSP-339**

Practice 1 - Node.js

Title : CLI Employee Management System Using Node.js and Arrays

Objective : Learn how to build an interactive command-line interface (CLI) application using Node.js that manages data in memory with arrays. This task strengthens your understanding of basic Node.js, user input handling, and array operations.

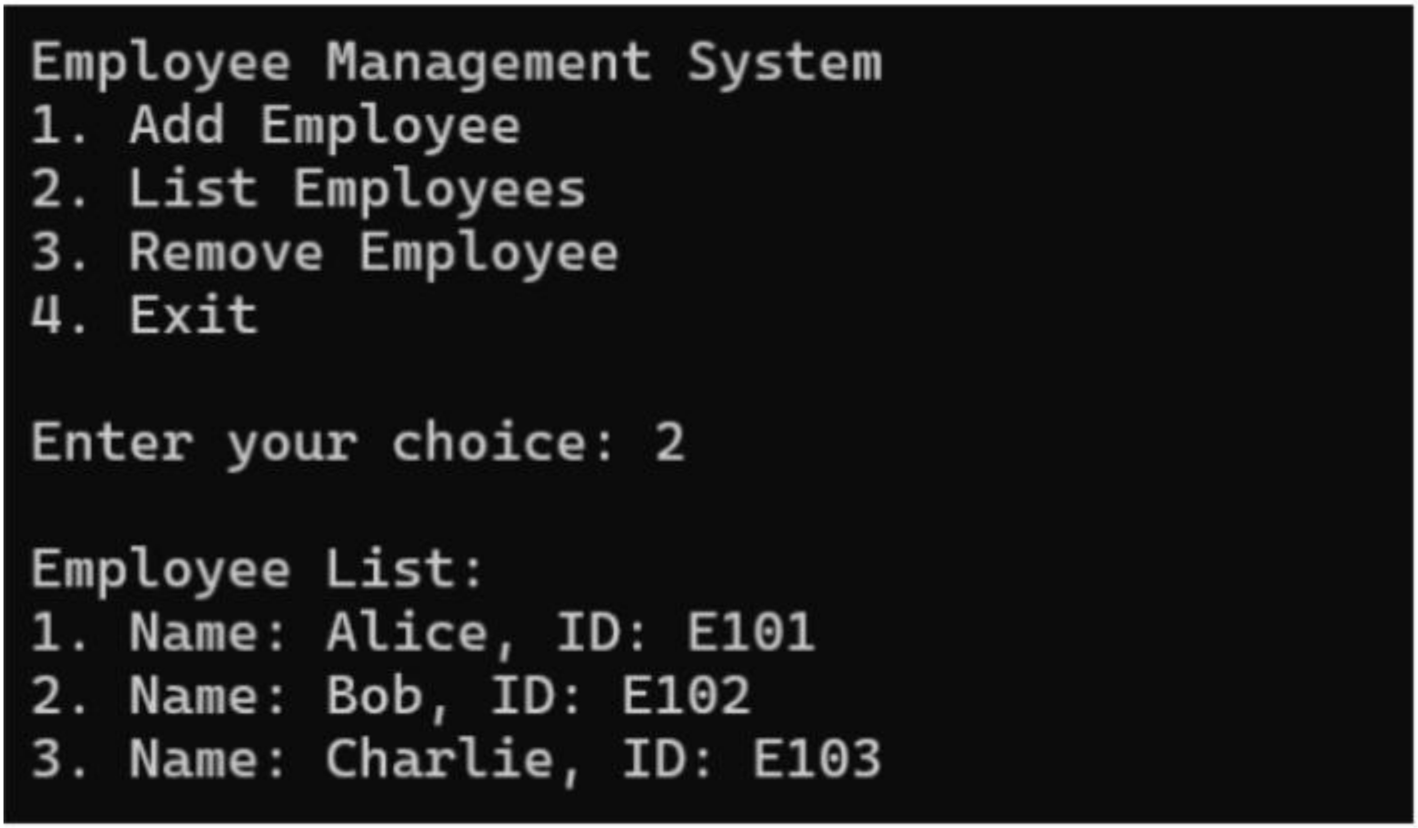
Task Description
Create a simple Node.js CLI application that manages a list of employees stored in an array. The app should allow the user to perform actions like adding a new employee (with name and ID), listing all employees, and removing an employee by ID. Use built-in Node.js modules like readline to capture user input interactively in the terminal. All data should be stored and updated directly in an array during the session, without using any external databases or files.

Code :

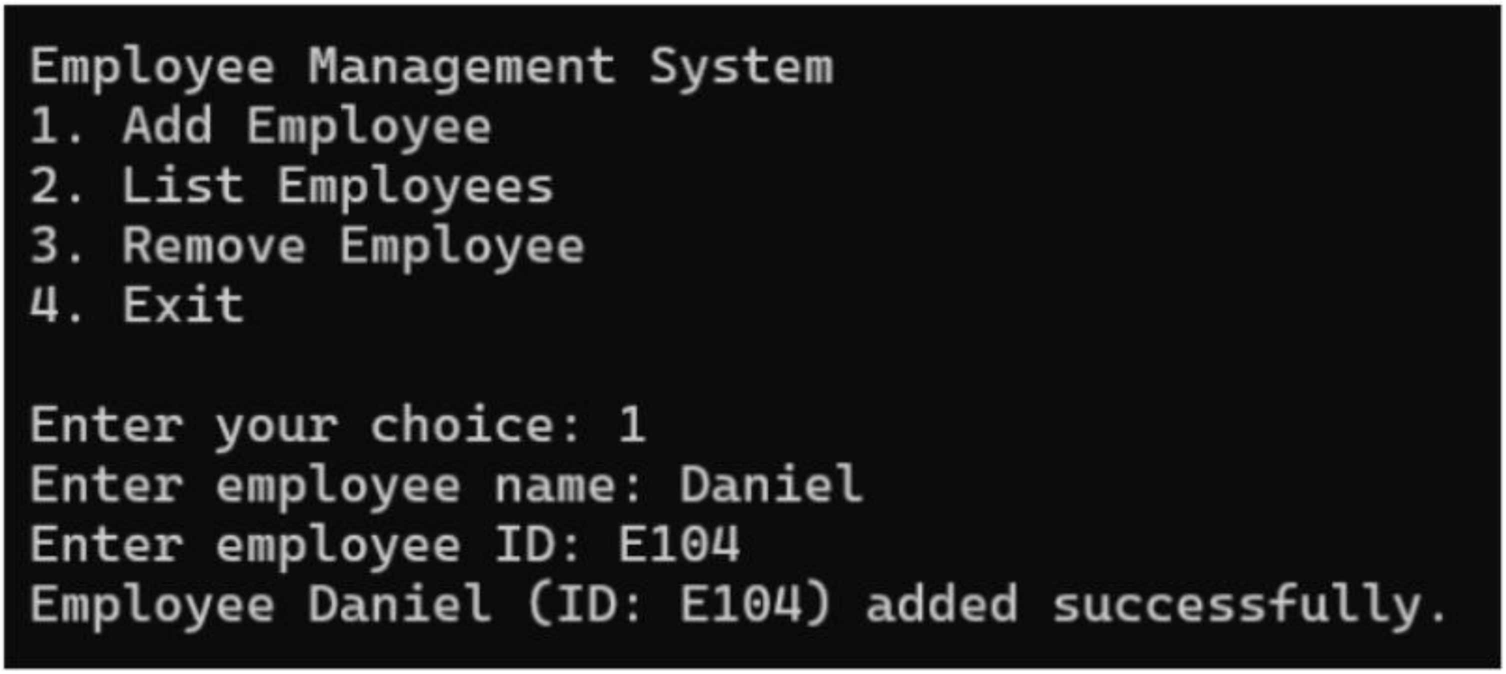
```
const readline=require("readline")
    const rl=readline.createInterface({
        input:process.stdin,
        output:process.stdout
    });

    const employees=[
        {
            id:"1",
            name:"Alice",
            Age:34
        },
        {
            id:"2",
            name:"inBorder",
            Age:30
        },
        {
            id:"3",
            name:"land",
            Age:34
        }
    ];

    function showMenu(){
        console.log("\n=== Employee Manager ===");
        console.log("1. Add Employee");
        console.log("2. List Employees");
        console.log("3. Remove Employee");
        console.log("4. Exit\n");

        rl.question("Enter your choise:",(choice)=>{
            switch (choice.trim()){
                case "1":
                    addEmployee();
                    break;
                case "2":
                    listEmployees();
                    break;
                case "3":
                    removeEmployee();
                    break;
                case "4":
                    console.log("Exiting...");
                    rl.close();
                    break;
                default:
                    console.log("Invalid choice! Try again.");
                    showMenu();
            }
        });
    }
function addEmployee() {
  rl.question("Enter Employee ID: ", (id) => {
    rl.question("Enter Employee Name: ", (name) => {
      if (employees.find(emp => emp.id === id.trim())) {
        console.log("Employee with this ID already exists!");
      } else {
        employees.push({ id: id.trim(), name: name.trim() });
        console.log("Employee added successfully!");
      }
      showMenu();
    });
  });
}

function listEmployees() {
  console.log("\n--- Employee List ---");
  if (employees.length === 0) {
    console.log("No employees found.");
  } else {
    employees.forEach(emp => {
      console.log(`ID: ${emp.id}, Name: ${emp.name}`);
    });
  }
  showMenu();
}
```

```
function removeEmployee() {
  rl.question("Enter Employee ID to remove: ", (id) => {
    const index = employees.findIndex(emp => emp.id === id.trim());
    if (index === -1) {
      console.log("Employee not found!");
    } else {
      employees.splice(index, 1);
      console.log("Employee removed successfully!");
    }
    showMenu();
  });
}

// Start the CLI
showMenu();
```

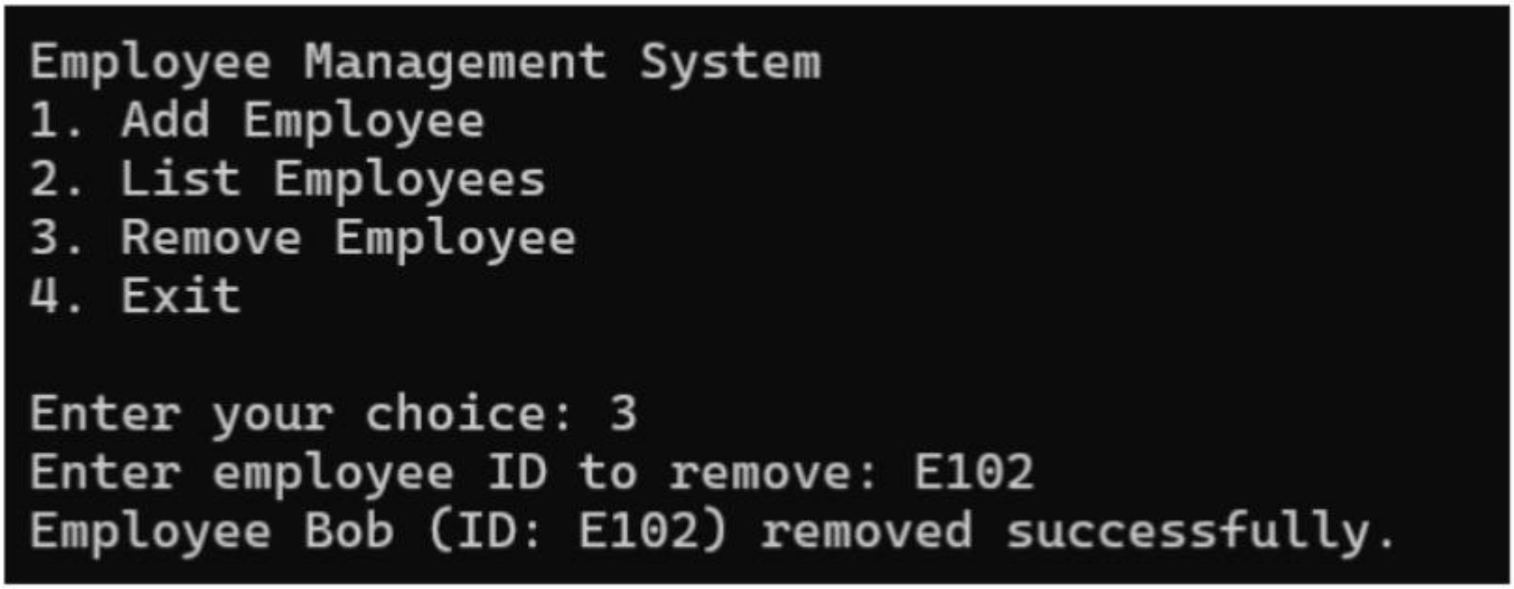
Expected Output



```
Employee Management System
1. Add Employee
2. List Employees
3. Remove Employee
4. Exit

Enter your choice: 2

Employee List:
1. Name: Alice, ID: E101
2. Name: Bob, ID: E102
3. Name: Charlie, ID: E103
```



```
Employee Management System
1. Add Employee
2. List Employees
3. Remove Employee
4. Exit

Enter your choice: 1
Enter employee name: Daniel
Enter employee ID: E104
Employee Daniel (ID: E104) added successfully.
```



```
Employee Management System
1. Add Employee
2. List Employees
3. Remove Employee
4. Exit

Enter your choice: 3
Enter employee ID to remove: E102
Employee Bob (ID: E102) removed successfully.
```