Mini Project Report - 10

Master of Computer Application – General
Semester – III

Sub: Web Technologies

Topic: Modules(Employee details)

By
Name: SANDRA B
Reg no.: 24110222500001

Faculty Name: VEERA RAGHAV K

Faculty Signature: _____

Department of Computer Application
Alliance University
Chandapura - Anekal Main Road, Anekal
Bengaluru - 562 106

August 2025

Mini Project Report: Employee Module using JavaScript

1. Introduction

JavaScript is one of the most widely used programming languages for developing web applications. Over time, the language has evolved significantly, with the introduction of **ECMAScript 6 (ES6)** features making development more structured, modular, and efficient. One of the most powerful features of ES6 is the concept of **modules**, which allows developers to split their code into reusable components that can be imported and exported across files.

This mini project demonstrates the practical use of **ES6 modules** by creating a simple Employee class in one file (emp.js) and then importing it into another file (modulesResult.js) to display the details of an employee.

The project highlights the importance of modularity, code reusability, and separation of concerns. In real-world applications, such modular designs make software systems easier to maintain and extend.

---

2. Objective

The main objectives of this mini project are:

1. To learn and implement the concept of **ES6 modules** in JavaScript.
2. To create a simple **Employee class** that encapsulates employee-related information such as ID, name, email, phone number, and salary.
3. To demonstrate the **import and export** mechanism in ES6 by separating class definitions from the main program logic.
4. To practice object-oriented programming (OOP) in JavaScript using **classes and constructors**.
5. To build a working console-based project that displays employee details in a structured way.

---

3. Tools and Technologies Used

The tools and technologies used in this project are:

- **Programming Language**: JavaScript (ES6)
- **Environment**: Node.js
- **Editor**: Visual Studio Code (VS Code)
- **Operating System**: Windows 10
- **Libraries/Modules**: ES6 built-in module system (import/export)

Why these tools?

- **Node.js** allows JavaScript to run outside the browser, making it perfect for testing console-based applications.
- **VS Code** provides a modern, lightweight, and powerful IDE for JavaScript projects.
- **ES6 modules** offer a cleaner and more maintainable way to organize and structure code.

---

4. Description of Code

The project consists of **two JavaScript files**:

(a) emp.js – Employee Class

This file contains the definition of the Employees class.

```
export class Employees {
   constructor(empId, name, email, phone, salary) {
      this.empId = empId;
      this.name = name;
      this.email = email;
      this.phone = phone;
      this.salary = salary;
   }
}
```

Explanation:

- The Employees class is exported using the export keyword.
- A **constructor** method initializes the employee properties when a new object is created.
- Attributes include:
    - empId – Unique employee identification number
    - name – Employee name
    - email – Employee email address
    - phone – Contact number
    - salary – Monthly salary

(b) modulesResult.js – Main Program

```
import { Employees } from "./emp.js";

let emp1 = new Employees(101, "Sandra", "sandra@email.com", "9876543210", 50000);

console.log("Employee ID:", emp1.empId);
console.log("Name:", emp1.name);
console.log("Email:", emp1.email);
console.log("Phone:", emp1.phone);
console.log("Salary: ₹", emp1.salary);
```

Explanation:

- The Employees class is imported from emp.js using the import statement.
- An object emp1 is created with employee details.
- The employee details are displayed using console.log().

---

5. Working of the Project

1. **Class Creation**: The Employees class is created in emp.js and exported.
2. **Importing the Module**: In modulesResult.js, the Employees class is imported.
3. **Object Instantiation**: A new employee object is created with values such as empId, name, email, phone, and salary.
4. **Displaying Results**: The employee details are printed on the console using console.log().
5. **Execution**: The project is executed in the Node.js environment by running:
6. node modulesResult.js

---

6. Features

- **Modular Design**: Code is separated into modules, making it reusable and easy to maintain.
- **Object-Oriented Approach**: Employee details are encapsulated within a class.
- **Scalable**: More employees can easily be added by creating new objects.
- **Readable and Organized**: Clear structure with imports and exports.
- **Realistic**: Mimics how data is stored and managed in real-world employee management systems.

---

7. Output

Console Output:

Employee ID: 101

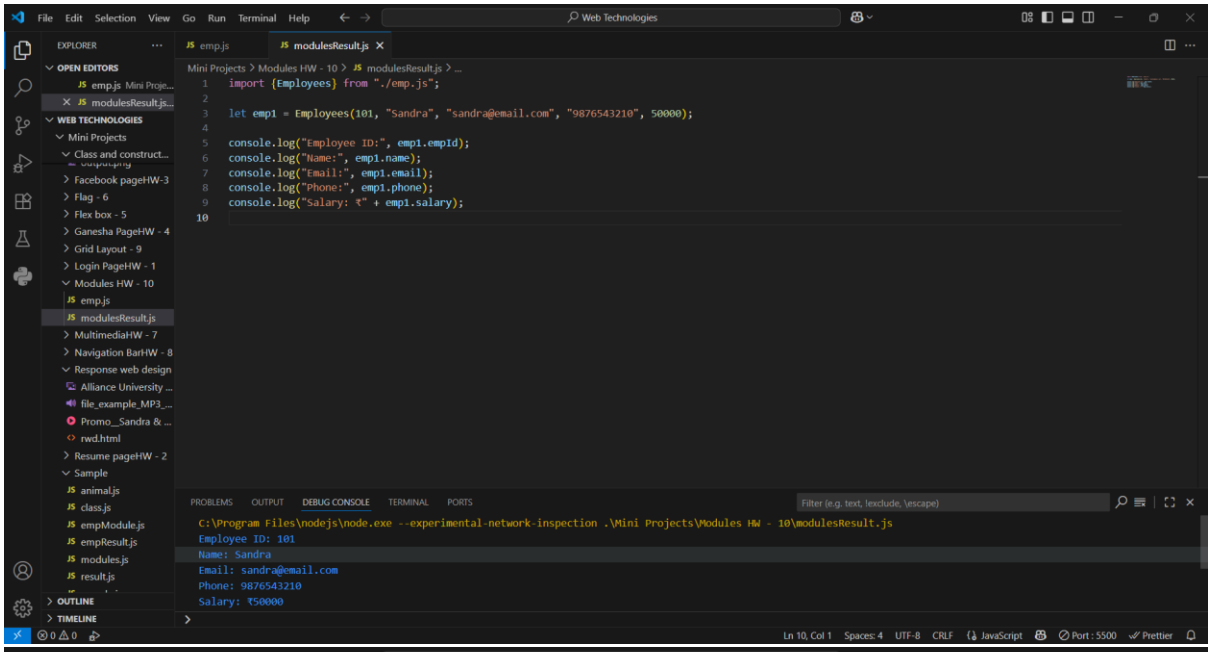
Name: Sandra

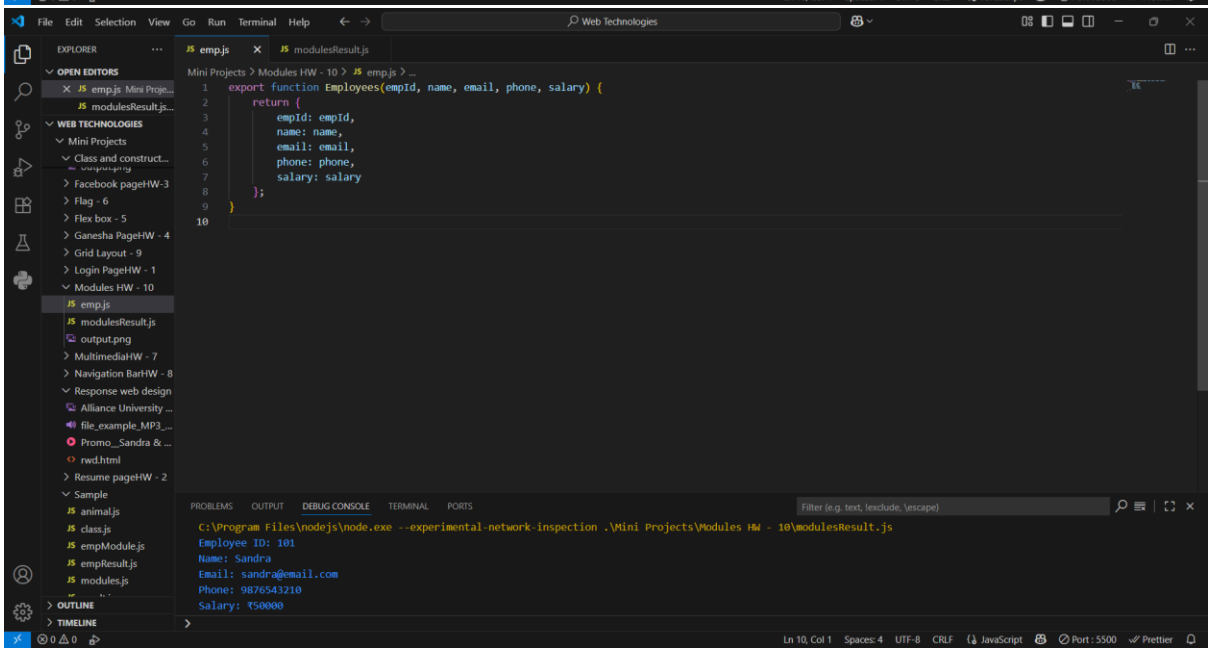
Email: sandra@email.com

Phone: 9876543210

Salary: ₹ 50000

This output confirms that the program successfully imported the class, created an object, and displayed the employee details.





8. Advantages

- **Code Reusability**: The Employees class can be reused in other projects.
- **Maintainability**: Modular programming ensures easy maintenance.
- **Scalability**: More features like department, designation, or address can be added without affecting existing code.
- **Separation of Concerns**: Business logic (employee definition) is separated from application logic (displaying results).
- **Modern Practice**: Uses latest ES6 features, making the code future-proof.

9. Future Scope

This simple project can be extended in multiple ways:

1. **Multiple Employees**: Store and display details of multiple employees using arrays.
2. **CRUD Operations**: Add Create, Read, Update, Delete operations for employee management.
3. **Database Integration**: Connect with a database (like MongoDB or MySQL) to store employee data permanently.
4. **Web Interface**: Create a frontend web page to input and display employee details dynamically.
5. **REST API**: Develop an API for employee management using Node.js and Express.js.
6. **Validation**: Add validation for fields like email format and phone number length.

---

10. Conclusion

This mini project successfully demonstrates the **concept of ES6 modules in JavaScript**. By exporting a class from one file and importing it into another, the project highlights the benefits of modular programming.

It provides a clear example of how object-oriented principles and modularity can be applied in JavaScript to build scalable and maintainable projects. Though small in scope, this project lays the foundation for larger applications such as **employee management systems, HR applications, and web-based business tools**.

Thus, the project achieves its objective of implementing **modules in JavaScript** and enhances understanding of how modular programming contributes to cleaner and more efficient software development.