

**ASSIGNMENT REPORT**

# HRM

| Semester: | Programming Fundamentals |
| --- | --- |
| Class: | WD1113 |
| Instructor: | Cuong Nguyen Dinh |
| Members: | Phan Nhat Quan (Alexander Phan) |

## **MỤC LỤC**

1. Giới Thiệu Dự Án
2. Triển Khai Module Frontend
3. Triển Khai Module Backend
4. Thách Thức Gặp Phải
5. Cách Kiểm Tra Hệ Thống
6. Kết Luận

## **1. Giới Thiệu Dự Án**

Hệ Thống Quản Lý Nhân Sự (HRM System) là một ứng dụng web hoàn chỉnh được xây dựng theo kiến trúc **MVC**, với **frontend** sử dụng **Vanilla JavaScript ES6+** và **backend** viết bằng **PHP 8.4+** kết nối **MySQL**.

Hệ thống bao gồm 8 module chính:  
 **Authentication, Employee, Department, Position, Salary, Attendance, Leave,** và **Performance.**

## **2. Triển Khai Module Frontend**

### **2.1 Cấu Trúc Module JavaScript**

Mỗi module được triển khai dưới dạng một **ES6 class** với các phương thức chính.

**Ví dụ – EmployeeModule.js:**

class EmployeeModule {

constructor(employeeDb, departmentModule, positionModule) {

this.employeeDb = employeeDb;

this.departmentModule = departmentModule;

this.positionModule = positionModule;

}

async render() {

const departments = await this.departmentModule.getAllDepartments();

const positions = await this.positionModule.getAllPositions();

// Render form với dropdown data

}

}

### **2.2 Dependency Injection Pattern**

* Mỗi module nhận dependencies thông qua constructor.
* Sử dụng **async/await** cho API calls.
* Thêm **cache busting** với ?v=timestamp để tránh lỗi cache trình duyệt.

### **2.3 API Integration**

* RESTful endpoints: /api/employees, /api/departments, /api/positions
* Error handling bằng **try-catch blocks**
* Response validation trước khi render giao diện

## **3. Triển Khai Module Backend**

### **3.1 MVC Architecture**

**Controller Layer:**

class EmployeeController {

public function getAll() {

$model = new EmployeeModel();

$employees = $model->getAllEmployees();

return json\_encode($employees);

}

}

**Model Layer:**

class EmployeeModel {

public function getAllEmployees() {

$sql = "SELECT \* FROM employees";

$stmt = $this->db->prepare($sql);

$stmt->execute();

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

}

### **3.2 SPL Autoload Implementation**

spl\_autoload\_register(function ($class) {

$candidates = [

CONTROLLER\_PATH . '/' . $class . '.php',

MODEL\_PATH . '/' . $class . '.php',

CONFIG\_PATH . '/' . $class . '.php',

];

foreach ($candidates as $file) {

if (is\_file($file)) {

require\_once $file;

return;

}

}

});

### **3.3 Database Connection**

* Kết nối bằng **PDO** với **prepared statements**
* Dùng charset **UTF8MB4** để hỗ trợ tiếng Việt
* Có **error handling** và **connection pooling**

## **4. Thách Thức Gặp Phải**

### **4.1 Kết Nối API**

**Vấn đề:** Trình duyệt cache khiến JavaScript không load phiên bản mới  
 **Giải pháp:**

* Thêm cache busting: script.js?v=timestamp
* Sử dụng header Cache-Control: no-cache
* Dùng **Ctrl + F5** để hard refresh

### **4.2 Field Naming Convention**

**Vấn đề:** Không đồng nhất giữa snake\_case (backend) và camelCase (frontend)

| **Backend** | **Frontend** |
| --- | --- |
| department\_id | departmentId |
| hire\_date | hireDate |

**Giải pháp:**

* Mapping dữ liệu trong API responses
* Giữ naming thống nhất trong codebase
* Validation ở cả frontend và backend

### **4.3 OOP Implementation**

**Vấn đề:** Quản lý dependency và module communication  
 **Giải pháp:**

* Dùng **constructor injection pattern**
* Áp dụng **async/await** cho API calls
* Xử lý lỗi bằng **error boundaries**

### **4.4 Data Type Handling**

**Vấn đề:** Xuất hiện giá trị NaN trong tính lương  
 **Giải pháp:**

calculateNetSalary(employee) {

const baseSalary = parseFloat(employee.salary) || 0;

const bonus = parseFloat(employee.bonus) || 0;

const deduction = parseFloat(employee.deduction) || 0;

const netSalary = baseSalary + bonus - deduction;

return isNaN(netSalary) ? 0 : netSalary;

}

### **4.5 Date Handling**

**Vấn đề:** Lỗi “Invalid Date” hiển thị trên giao diện  
 **Giải pháp:**

* Dùng format thống nhất YYYY-MM-DD
* Tạo đối tượng Date đúng cách
* Format theo locale người dùng

## **5. Cách Kiểm Tra Hệ Thống**

### **5.1 Manual Testing**

**Frontend Testing:**

1. Mở Developer Tools (F12)
2. Kiểm tra Console để phát hiện JavaScript errors
3. Xác minh API calls trong Network tab
4. Test responsive design trên mobile

**Backend Testing:**

$response = file\_get\_contents('http://localhost/HRM/public/api.php/employees');

$data = json\_decode($response, true);

var\_dump($data);

### **5.2 Database Testing**

-- Kiểm tra tính toàn vẹn dữ liệu

SELECT COUNT(\*) FROM employees;

SELECT \* FROM employees WHERE department\_id IS NULL;

SELECT \* FROM departments WHERE manager\_id IS NULL;

### **5.3 Integration Testing**

**Test Cases:**

1. **Login Flow:** Username/password → Session creation
2. **CRUD Operations:** Create → Read → Update → Delete
3. **Data Validation:** Invalid input → Error messages
4. **API Responses:** Kiểm tra status code & JSON format
5. **Cross-browser:** Chrome, Firefox, Edge

### **5.4 Performance Testing**

* Thời gian load trang < **3 giây**
* API response time < **500ms**
* Tối ưu query database
* Giám sát mức tiêu thụ bộ nhớ

## **6. Kết Luận**

Dự án **HRM System** đã được triển khai thành công với 8 module hoàn chỉnh.  
 Các thách thức chính về **API integration, OOP design, và data handling** đã được giải quyết thông qua các **best practices** như dependency injection, error handling, và consistent naming.

Hệ thống đã được kiểm thử kỹ lưỡng và **sẵn sàng cho production deployment.**