Portfolio of Evidence (PoE) – Part 2

# Contract Monthly Claim System (CMCS) – Prototype Implementation

## 1. Introduction

This part of the Portfolio of Evidence (PoE) focuses on the implementation of the Contract Monthly Claim System (CMCS) prototype. The system is designed to streamline the submission, tracking, and approval of monthly claims for Independent Contractor lecturers. It is based on the design, structure, and specifications set out in Part 1, which involved planning and developing a non-functional prototype.

## 2. Application Overview

The CMCS prototype allows lecturers to submit claims for their working hours, track the status of submitted claims, and have them reviewed by Programme Coordinators and Academic Managers. The system is developed using ASP.NET MVC and a SQL Server database, which stores lecturer details, claim submissions, module information, and claim statuses.

## 3. Project Implementation

3.1. Technologies Used:

- Front-end: HTML, CSS, JavaScript  
- Back-end: ASP.NET MVC Framework  
- Database: SQL Server  
- Development Environment: Visual Studio  
- Testing Framework: MSTest (for unit tests)

3.2. Database Structure

The database includes the following tables:  
- Lecturer: Stores lecturer details, including their name, contact information, and employee number.  
- Module: Contains details about the modules lecturers are teaching.  
- Claim: Stores claims submitted by lecturers, including the submission date, status, module claimed for, hours worked, and hourly rate.

3.3. Key Pages and Functionality

1. Login Page:  
- This page allows users to log into the system. The system accepts any valid input as a login and shows a 'Login Successful' message.  
  
2. Submit Claim Page:  
- Lecturers can input the details of their claims, including hours worked, hourly rate, and the module. Once submitted, the claim status is set to 'Pending.'  
  
3. Track Status Page:  
- Lecturers can view the status of their submitted claims. Statuses can be 'Pending,' 'Approved,' or 'Rejected.'  
  
4. Admin Page:  
- This page is accessible by Programme Coordinators and Academic Managers, who can approve or reject claims submitted by lecturers.  
  
5. File Upload Feature:  
- Lecturers can attach supporting documents to their claim submissions.

## 4. Code Implementation

The CMCS system has been developed with a focus on modularity, scalability, and clean code. Below is an outline of the main controllers and models used in the project:  
  
- ClaimController: Handles claim submission, retrieval, and status updates.  
- StatusController: Manages the status tracking feature.  
- ClaimViewModel: This model contains properties such as ClaimId, LecturerId, ModuleId, HoursWorked, HourlyRate, and SubmissionDate.

## 5. Testing and Validation

Unit tests have been implemented to ensure the correct functionality of critical components. Below are some of the tests included:  
- LoginTest: Ensures that the login function correctly processes user inputs.  
- SubmitClaimTest: Validates that claims are properly created and saved in the database.  
- TrackStatusTest: Verifies that the correct claim status is returned for a given lecturer.

## 6. User Experience and Design

The CMCS system was designed with a focus on usability, responsiveness, and accessibility. It adheres to WCAG 2.0 guidelines to ensure that it is accessible to all users. The layout is responsive across devices, including mobile phones, tablets, and desktops, providing a seamless user experience.

## 7. Challenges Encountered and Solutions

Several challenges were encountered during the development process:  
- Database Connectivity: Issues with database connections were resolved by refining the SQL scripts and adjusting the connection string in the application.  
- Error Handling: Robust error handling mechanisms were introduced to ensure that invalid data inputs are gracefully handled without crashing the system.

## 8. Conclusion

Part 2 of the CMCS project focused on bringing the design to life through the development of a functional prototype. With features for submitting claims, tracking statuses, and administrative review, the system fulfills the requirements set out in Part 1. The system is now fully functional and ready for further testing and user feedback.