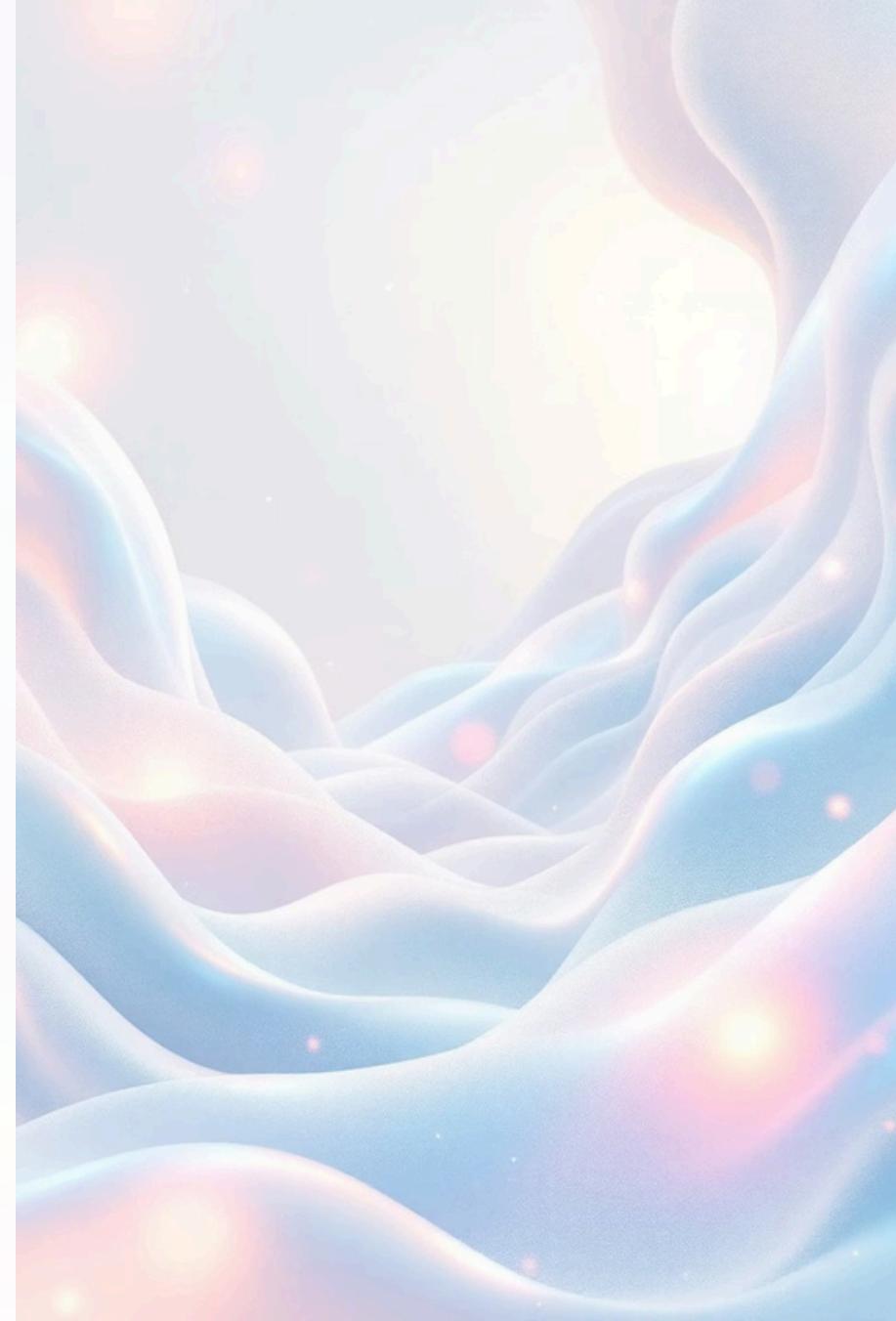


Contract Monthly Claim System

PROG6212 Portfolio of Evidence - Part 3 Presentation

"This presentation showcases the implementation and automation of the Contract Monthly Claim System (CMCS) as per the PROG6212 POE requirements, focusing on the sophisticated enhancements developed in Part 3."

Presented by: Naoyuki Christopher Higaki - ST10462415



Presentation Roadmap: Enhancing the CMCS



Project Recap & System Demonstration

A high-level overview of the functional CMCS prototype.



Deep Dive into Part 3 Automation Features

Examining the auto-calculation, validation, and workflow enhancements.



Technical Implementation & Code Quality

Reviewing architecture, unit testing, and professional practices.



Conclusion & Value Proposition

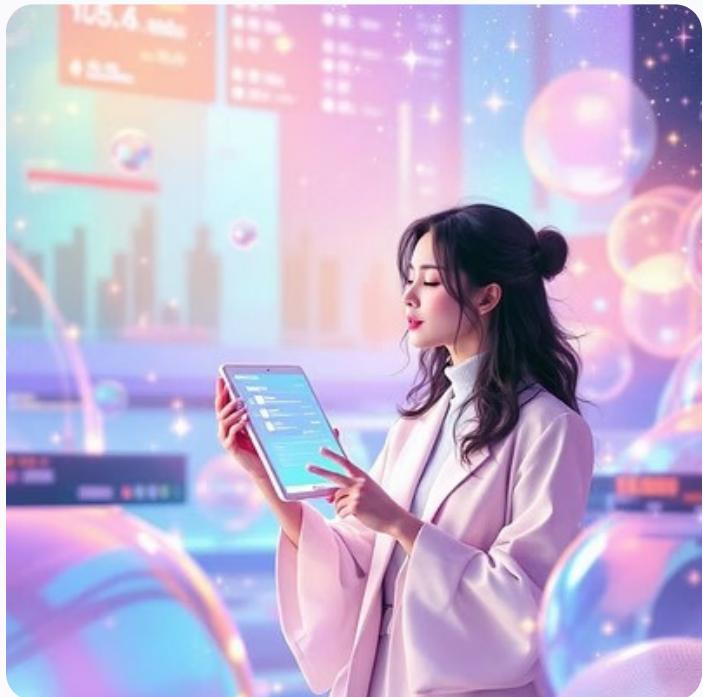
Summarizing the system's impact and POE fulfillment.



POE Structure Fulfilled

- Part 1: Project Planning & Prototype (UML, Project Plan, GUI Design)
- Part 2: Functional Web Application (Claim Submission, Approval, Tracking)
- **Part 3: Application Enhancement & Automation** (Focus of this Presentation)

System Demonstration: Key User Journeys



Lecturer Login & Claim Submission

Intuitive form for inputting hours and rates, designed for speed and clarity.



Coordinator Approval Workflow

Clear, organized views of pending claims with simple one-click approval mechanisms.



Academic Manager Oversight

Highlights automated reporting and analytical capabilities for budget oversight and planning.



Claim Status Transparency

Real-time status updates accessible to all stakeholders from submission to settlement.

Part 3 Automation: Enhancing the Lecturer View

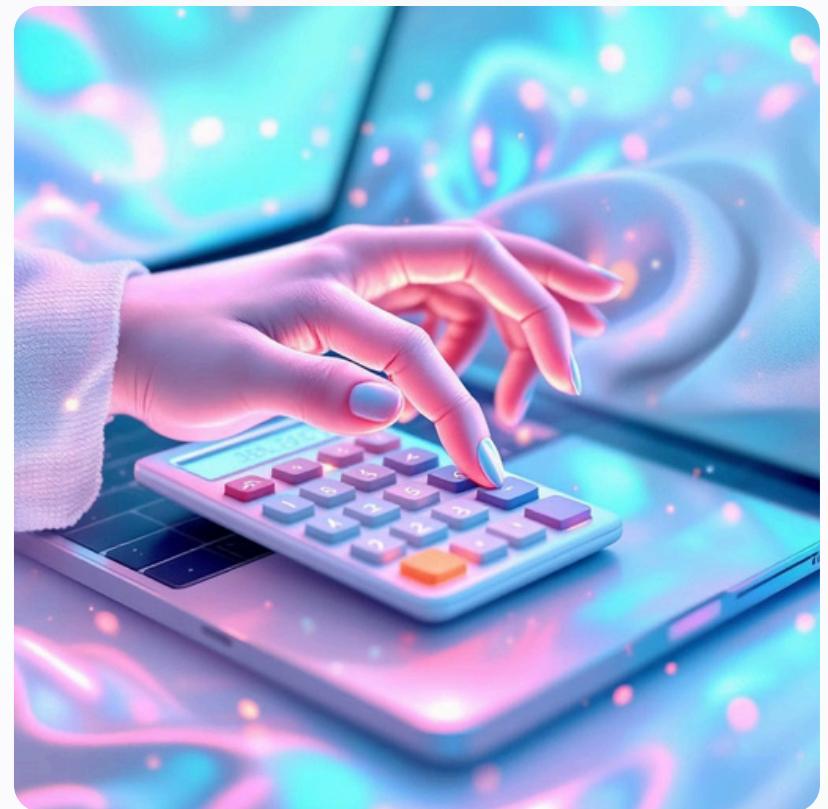
Automation Feature 1: Auto-Calculation

→ Instant Payment Calculation

The final payment amount is automatically calculated as the lecturer inputs Hours Worked and the agreed-upon Hourly Rate.

→ Technical Foundation

Utilizes JavaScript/jQuery for immediate client-side feedback, with server-side validation and secure storage via C# models.



Automation Feature 2: Validation Checks

→ Automated Data Integrity

Automated data validation prevents submission errors (e.g., hours cannot exceed the maximum for a month, ensuring positive numeric values only).

→ Robust Rule Implementation

Implemented using ASP.NET Core Model Validation integrated with custom business logic to ensure regulatory compliance.

These features directly address the Part 3 rubric requirement for an "auto-calculation feature and validation checks implemented effectively."

Automating the Review Process: Coordinator & Manager View

The CMCS implements sophisticated workflows to streamline oversight, minimize manual scrutiny, and ensure compliance.

1

Automated Claim Verification

The system **automatically flags claims** for required review based on configurable criteria, such as exceeding a specific monetary threshold (e.g., claims over R5,000) or unusual hour totals.

- Focuses reviewer attention on outliers.
- Reduces time spent on standard claims.

2

State-Based Approval Workflows

A clear workflow automates status transitions: **Submitted** ³ **ApprovedByCoordinator** ³ **ApprovedByManager** ³ **Paid**.

- Workflow buttons trigger automated status transitions.
- Ensures a consistent, auditable, and traceable approval history.

This fulfills the Part 3 rubric criterion: "automated verification and approval processes are implemented effectively, enhancing efficiency and accuracy."



Strategic Automation for HR & Financial Oversight

Automated Reporting & Analytics

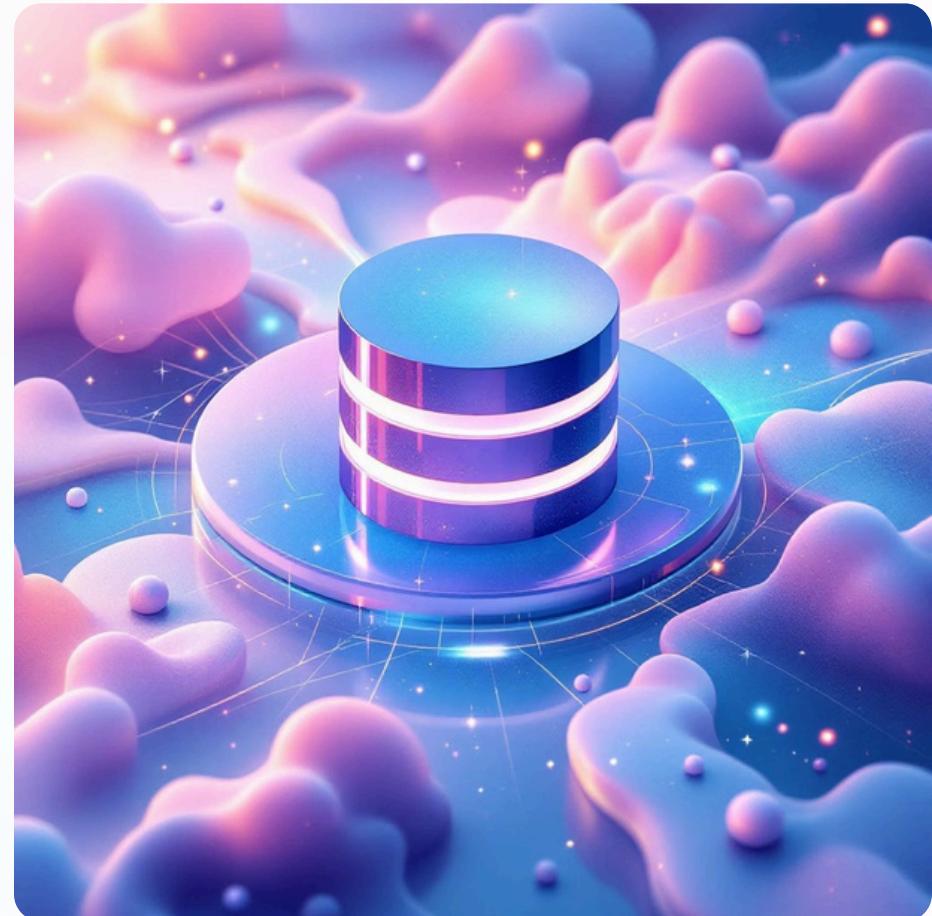
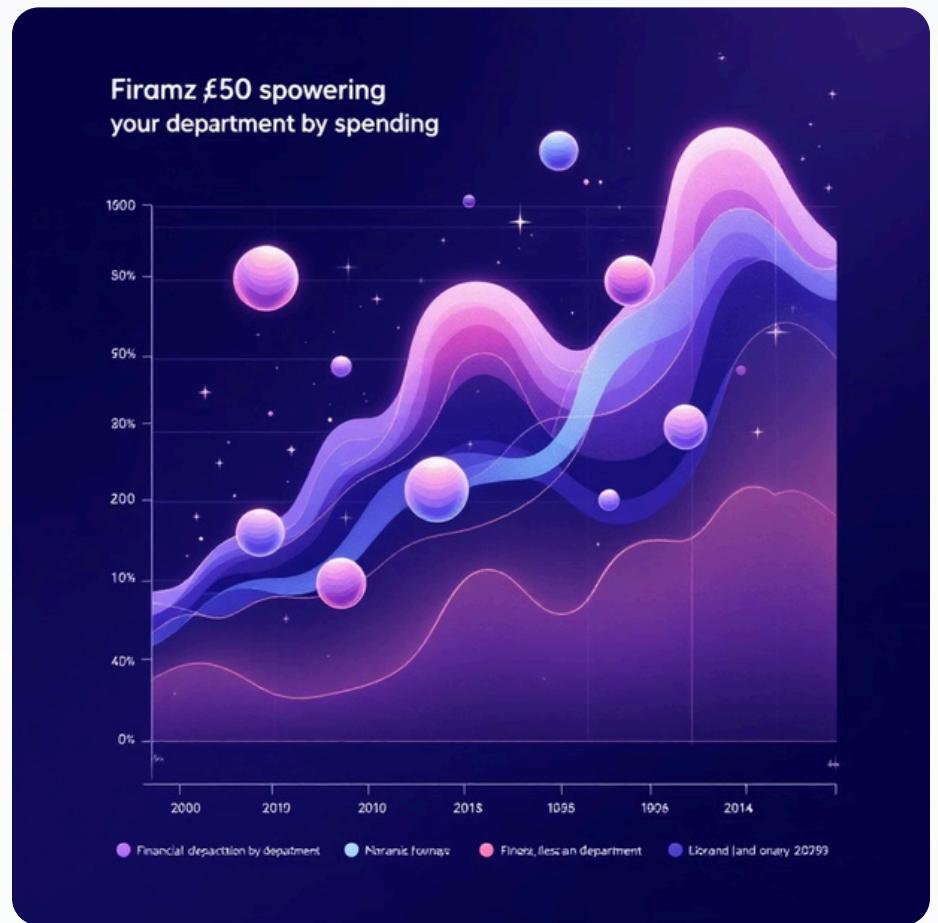
The HR Dashboard provides critical oversight, replacing tedious manual data aggregation with instant, dynamic reports:

Total claims processed per academic period.

Departmental spending breakdowns to aid budget forecasting.

Analysis of approval rates and average processing times.

Data aggregation is handled efficiently using **LINQ queries** and displayed dynamically in secure Razor views.



Centralized Lecturer Data Management

The system offers a centralized, automated interface for viewing and managing all lecturer information. This data is **directly linked** to their comprehensive claim history.

- Eliminates data silos and redundant entry.
- Provides a single source of truth for HR audits.

This addresses the requirement for "automation of claim processing and lecturer data management tasks," significantly reducing manual effort.

Technical Excellence: Architecture and Code Quality



Robust Architecture

Built on the **ASP.NET Core 8.0** **MVC** framework, ensuring scalability, maintainability, and security for a professional-grade web application.

Data and Security

The Data Layer uses a robust Text File Service (JSON) with atomic writes for data integrity, complemented by Session-Based Authentication. Comprehensive **Unit tests** validate controllers, models, and critical business logic, ensuring the reliability of all auto-calculation and validation features.

Unit Testing Coverage

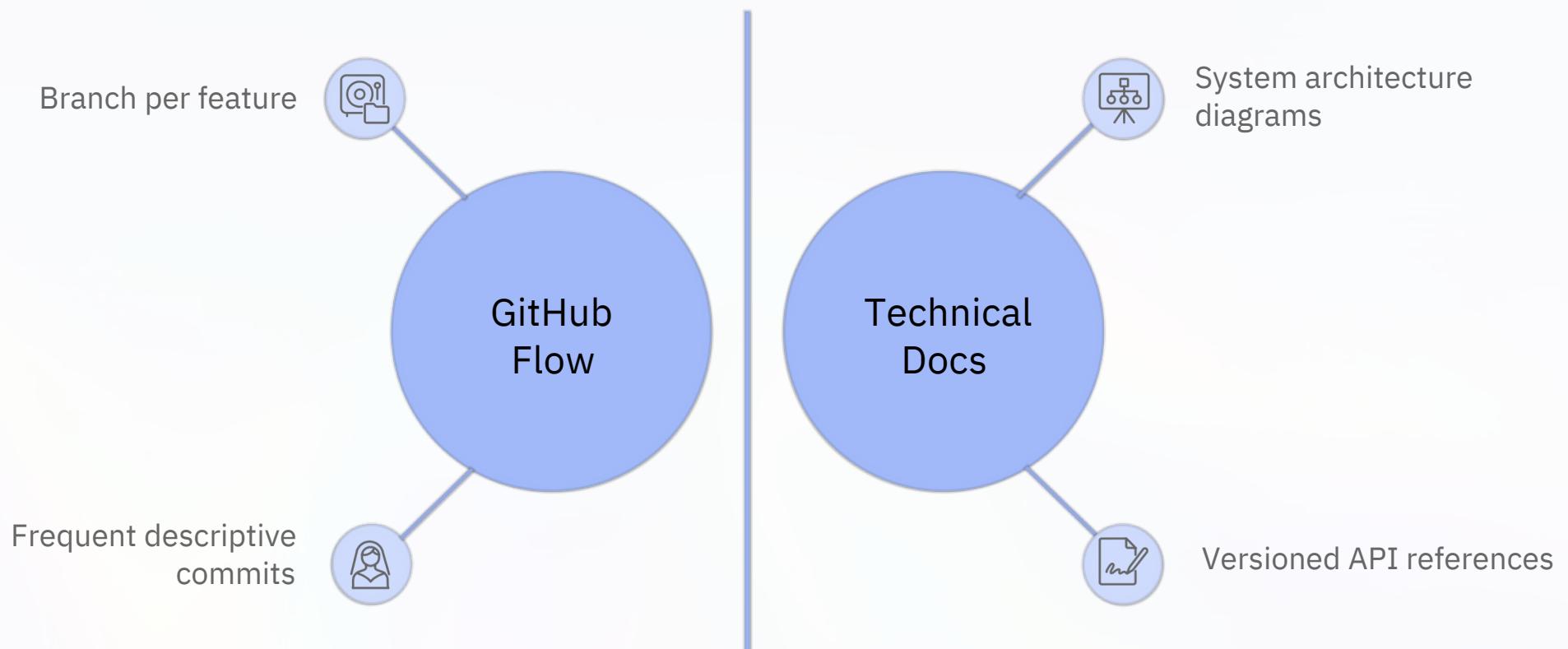
Comprehensive **Unit tests** validate controllers, models, and critical business logic, ensuring the reliability of all auto-calculation and validation features.

Error Handling

Graceful exception handling throughout the application provides user-friendly error messages and prevents data loss, ensuring a consistent user experience.

The focus on "comprehensive unit testing" and "robust error handling" meets the high standards required for the POE.

Professional Development: Version Control & Documentation



Structured Version Control

All development, particularly Part 3's complex automation, was tracked in **GitHub**.

Over **10+ descriptive commits** for Part 3.

Commits followed a logical project timeline.

Example: "FEAT: Implement auto-calculation with overtime in ClaimController"

Comprehensive Documentation

Technical documentation provides a complete reference for future development or maintenance.

- Details the System Architecture (MVC Pattern).
- Explains the Data Model and relationships.
- Documents the API endpoints and security implementation.

Conclusion: Delivering Advanced Automation



Functional & Robust

Successfully delivered a complete, web-based CMCS prototype.



Advanced Automation

Implemented crucial automation features across all four user roles, minimizing manual steps.



Intuitive Design

Features a user-friendly GUI, enhancing adoption and minimizing training time.

The Core Value of the CMCS

Efficiency

Automates manual calculations, reporting, and complex approval workflows.

Accuracy

Reduces human error through automated validation and auto-calculation.

Preparedness

Demonstrates proficiency in .NET Core MVC, software design principles, and professional development practices for PROG6212.



Thank You

Questions?