Contract Monthly Claim System – Prototype Project Documentation

1. Documentation

* Introduction

The Contract Monthly Claim System is a non-functional prototype developed using ASP.NET Core MVC. The purpose of this prototype is to demonstrate the layout, structure, and interface of a claim management system for Coordinators and Lecturers before the system is made fully functional. The prototype provides navigation, sample dashboards, claim tables, and approval/rejection pages without database integration or backend logic.

* Architecture Choice: MVC Pattern

I used the Model-View-Controller (MVC) pattern with ASP.NET Core for several reasons:

* Separation of Concerns: MVC separates the application logic, user interface, and data processing into distinct units.
* Maintainability: Any module can be modified separately without affecting others.
* Scalability: Easy to scale with new features and controllers as it grows.
* Industry Standard: MVC is widely used as a standard web application standard and is easily understood by future developers.
* Technology Stack
* **ASP.NET Core MVC**: For web application framework
* **HTML/CSS**: For frontend presentation
* **Font Awesome**: For consistent iconography
* **Bootstrap**: For responsive design components
* Database Structure Design
* User Management: Lecturers and Coordinators with different roles and permissions.
* Claim Processing: Submission of claims by lecturers for approval.
* Document Management: Handling of supporting documents linked to claims.
* Status Tracking: Monitoring claim statuses such as Pending, Approved, and Rejected.
* GUI Layout Design
* Fixed Sidebar Navigation: Provides consistent access throughout the application
* Role-based Dashboards: Separate interfaces for Lecturers and Coordinators.
* Summary Cards: Visual representations of claim statistics for quick insights.
* Data Tables: Clear presentation of detailed claim information.
* Personalized Touch: Included my profile picture on the navigation bar for better user identification.

Assumptions and Constraints

* Assumptions Made:

One Institution: Only one educational institution is served by the system.  
Monthly Claims: Every month, claims are filed.  
Lecturers have the ability to upload supplementary documentation.  
Two roles for users: The system is only used by lecturers and coordinators.  
Web-based Access: Web browsers are used by users to access the system.  
Authentication is necessary. To access the system, each user needs to log in.

* Technical Constraints:

Absence of Database Implementation The current prototype only uses static data. Authentication: Simple login using modals without backend validation  
Absence of File Storage The ability to upload documents is UI-only.  
Static Data: For demonstration purposes, all shown data is hardcoded.

* Business Constraints:

Permission Process: Coordinator permission is needed for claims.  
Status Monitoring: Throughout their existence, claims must have a clear status.  
Roles of Users: Different permissions for coordinators and lecturers

1. UML Class Diagram for Database