Programming 2B PoE Part 1

# Name: Titus

# Surname: Ndaba

# Student Number: ST10448883

# Module Code: PROG6212

Table of Contents

[Name: Titus 1](#_Toc208951775)

[Surname: Ndaba 1](#_Toc208951776)

[Student Number: ST10448883 1](#_Toc208951777)

[Module Code: PROG6212 1](#_Toc208951778)

[Introduction 3](#_Toc208951779)

[Project Planning 3](#_Toc208951780)

[System Design Choices 5](#_Toc208951781)

[Database Design 6](#_Toc208951782)

[Graphical User Interface (Prototype) 7](#_Toc208951783)

[Assumptions and Constraints 10](#_Toc208951784)

[Conclusion 11](#_Toc208951785)

Contract Monthly Claim System (CMCS) Documentation

## Introduction

The Contract Monthly Claim System (CMCS) is a web application based on .NET, created to simplify the monthly claim procedure for Independent Contractor (IC) instructors. Naturally the system claims are presented and examined by hand. The Contract Monthly Claim System resolves this issue by offering an organized platform for lecturers to submit claims, include supporting documents, and monitor the status of their claims openly.

## Project Planning

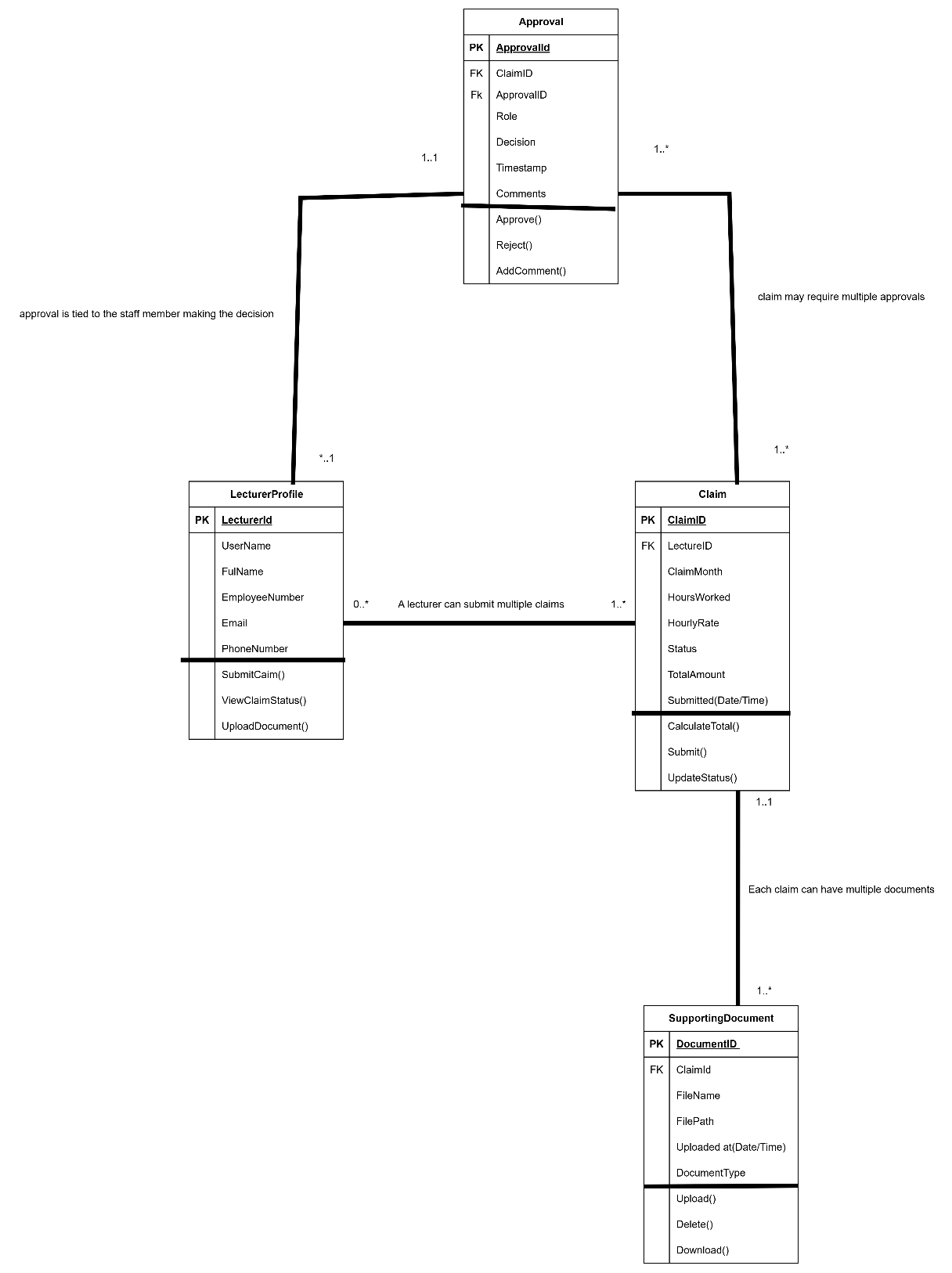
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Project Participants | | | | |
| **Name** | Contract Monthly Claim System | | | |
| **Stakeholder** | **Project Sponsor** | Finance Department | | |
| **Customers** | Independent Contractor Lecturers | | |
| **Contractors** | Software development team(Titus Ndaba) | | |
| **Project Team and Manager** | | | |
| **Project Management** | **Project Manager** | | | Mr. T Ndaba |
| **Assistance Project Manager** | | | Miss J Mabuza |
| **Website** | <https://github.com/NdabaTitus/TheContractMonthlyClaimSystem.git> | | | |
| **Project Team** | ICT Department Head, Company CIO | | | |
| Project Description | | | | |
| **Goal Statement** | The goal of the CMCS project is to create and execute an automated, secure, and user-friendly web system that simplifies the submission, verification, and approval of monthly claims for Independent Contractor (IC) lecturers. The objective of the system is to lessen administrative burdens, decrease mistakes, and offer real-time updates on claim status for all participants | | | |
| **Description And Background** | At present, claims from IC lecturers are handled manually through spreadsheets, emails, or paper forms. This approach is labor-intensive, susceptible to errors, and lacks clarity. CMCS offers a unified platform for lecturers to file claims, include necessary documents, and automatically compute totals based on hours worked and hourly pay rates | | | |
| **Objective** | Allow lecturers to submit claims with hours, rates, and supporting documents.  Automate calculations and data validation to reduce errors. | | | |
| **Scope** | CMCS will include a web application developed with ASP.NET Core MVC and C#, featuring role-based authentication, claim submission forms, automated calculations, document uploads, and status tracking | | | |
| **Deliveries** | GUI Prototype for lecturers, coordinators, managers, and HR.  Database Design including entities (LecturerProfile, Claim, SupportingDocument, Approval).  Documentation: Project plan, UML diagram, GUI wireframes, and system assumptions.  Future Implementation Plan for automation and HR reporting in later phases. | | | |
| **Schedules** | **Start Date** | | 01 September 2025 | |
| **End Date** | | 25 December 2025 | |
| **Time Report** | Weeks 1–2: Project Initiation & Approval   * Finalize project charter * Secure resources   Weeks 3–4: Requirements Gathering & Analysis   * Collect detailed requirements from lecturers, coordinators, managers, and HR staff * Define system workflows and user roles   Weeks 5–6: System & UI Design   * Develop UML diagrams and database schema * Create GUI wireframes   Weeks 7–8: Prototype Development   * Build GUI prototype * Implement submission forms, dashboards, and document upload features   Weeks 9–10: Feature Integration & Enhancement   * Implement automated calculations and validation checks * Configure role-based access for different users   Week 11: Testing & Quality Assurance   * Conduct functional and usability testing   Week 12: Documentation & Reporting   * Compile all project documentation, UML diagrams, and wireframes * Prepare system assumptions and notes   End of Week 12: Project Review & Sign-Off  Obtain approval for the GUI prototype and documentation | | | |
| Assumption, Constraints, Dependencies, Impacts And Risks | | | | |
| **Assumptions** | Lecturers are expected to provide precise claims along with accompanying documents in a digital format. Users must have internet connectivity and compatible devices. The development team will maintain ongoing access to necessary tools, frameworks, and resources, while stakeholders will deliver prompt feedback and approvals throughout the project | | | |
| **Constraints** | The CMCS Part 1 output is restricted to just a non-functional GUI prototype. At this point, full automation, real-time approval workflows, payroll integration, and enhanced reporting remain beyond the current scope. The security of the system and role-specific access rely on the correct setup of ASP.NET Identity. Budget and time constraints limit extra features in the initial development stage | | | |
| **Risks** | Technical risks involve possible framework or database problems that could hinder development progress. Operational risks can occur if users struggle to use the system effectively. Resource risks encompass restricted access to developers or postponed feedback from stakeholders. Security threats arise when sensitive claim data is improperly managed, and financial threats may emerge if unforeseen expenses occur during development or implementation | | | |
| Cost | | | | |
| **Costing** | R1,200,000 | | | |
| Resources | | | | |
| **Technologies** | ASP.NET Core MVC  C#  SQL Server  Visual Studio 2022  GitHub | | | |
| **Name/ Signature** | **Roles And Responsibilities** | | | |
| Titus Ndaba | Project Manger | | | |
| Thando Ndlovu | Technical Lead | | | |
| Connie MThintelwa | Developer | | | |
| Vuyiswa Ndaba | Quality Analyst | | | |
| Agreement | | | | |
| **Signature** |  | | | |
| **Date** | 01 September 2025 | | | |
| **Project Sponsor** | Finance Department | | | |

## System Design Choices

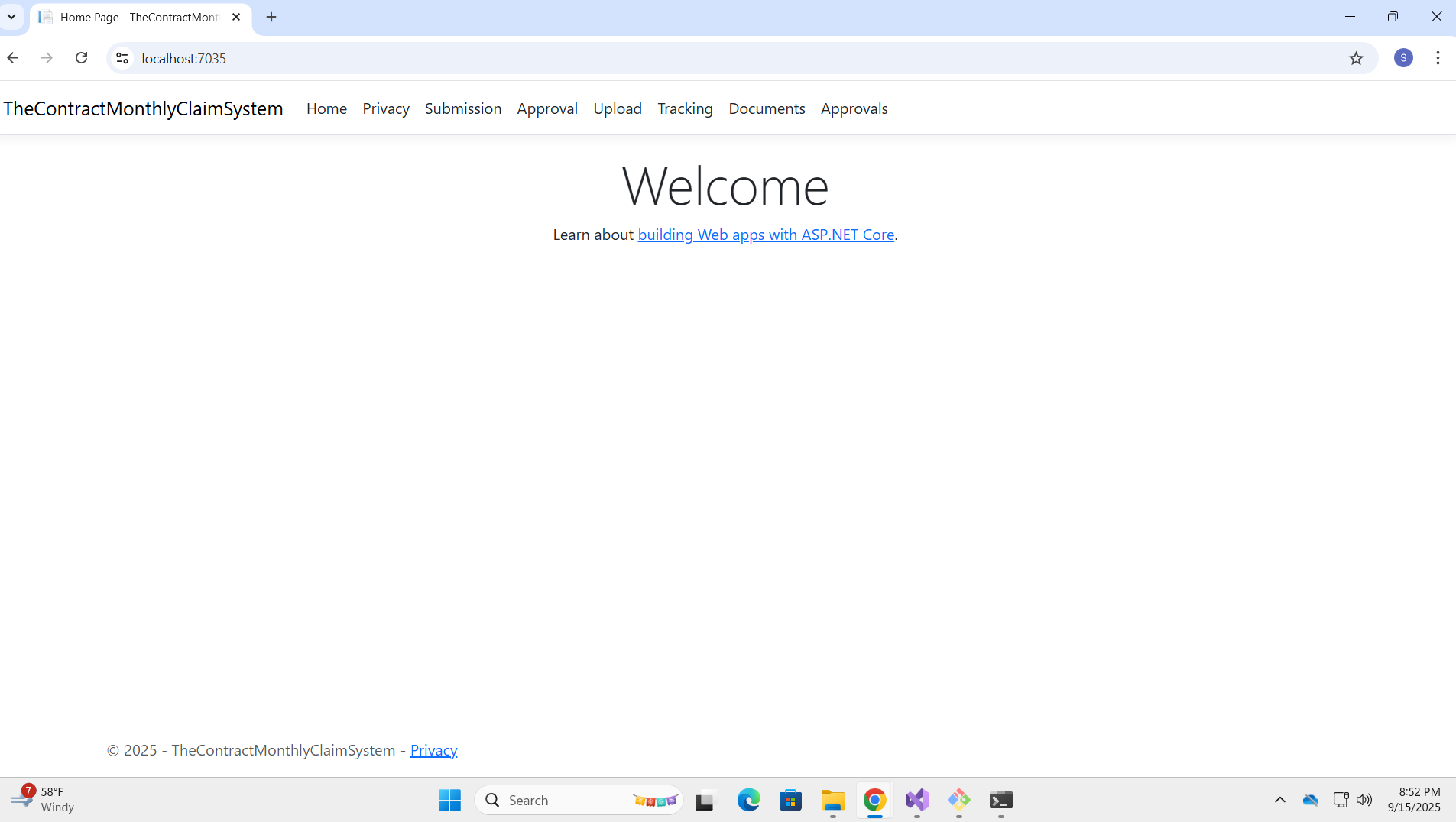
The Contract Monthly Claim System (CMCS) was created to offer a user-focused, role-oriented platform for Independent Contractor (IC) educators, Program Coordinators, Academic Managers, and HR personnel. My design decisions were influenced by three key principles: clarity, responsibility, and adaptability.

* Technology Stack: I chose ASP.NET Core MVC with C# as it offers a robust framework for developing modular web applications, supports role-based authentication (through ASP.NET Identity), and seamlessly integrates with Entity Framework Core for database management.
* Role-Based Access: The system employs Identity roles (Lecturer, Coordinator, Manager) to guarantee that every user engages solely with the features pertinent to their duties. This enhances safety and simplifies the user interface.
* Workflow Overview: Claims progress through stages (Draft → Submitted → Verified → Approved/Rejected). This design decision reflects actual approval procedures and guarantees responsibility at every level.
* Automation: To minimize human errors, the system computes claim totals (HoursWorked × HourlyRate) automatically and checks inputs for accuracy. Upcoming improvements could involve automated alerts and checks of policy rules.

## Database Design

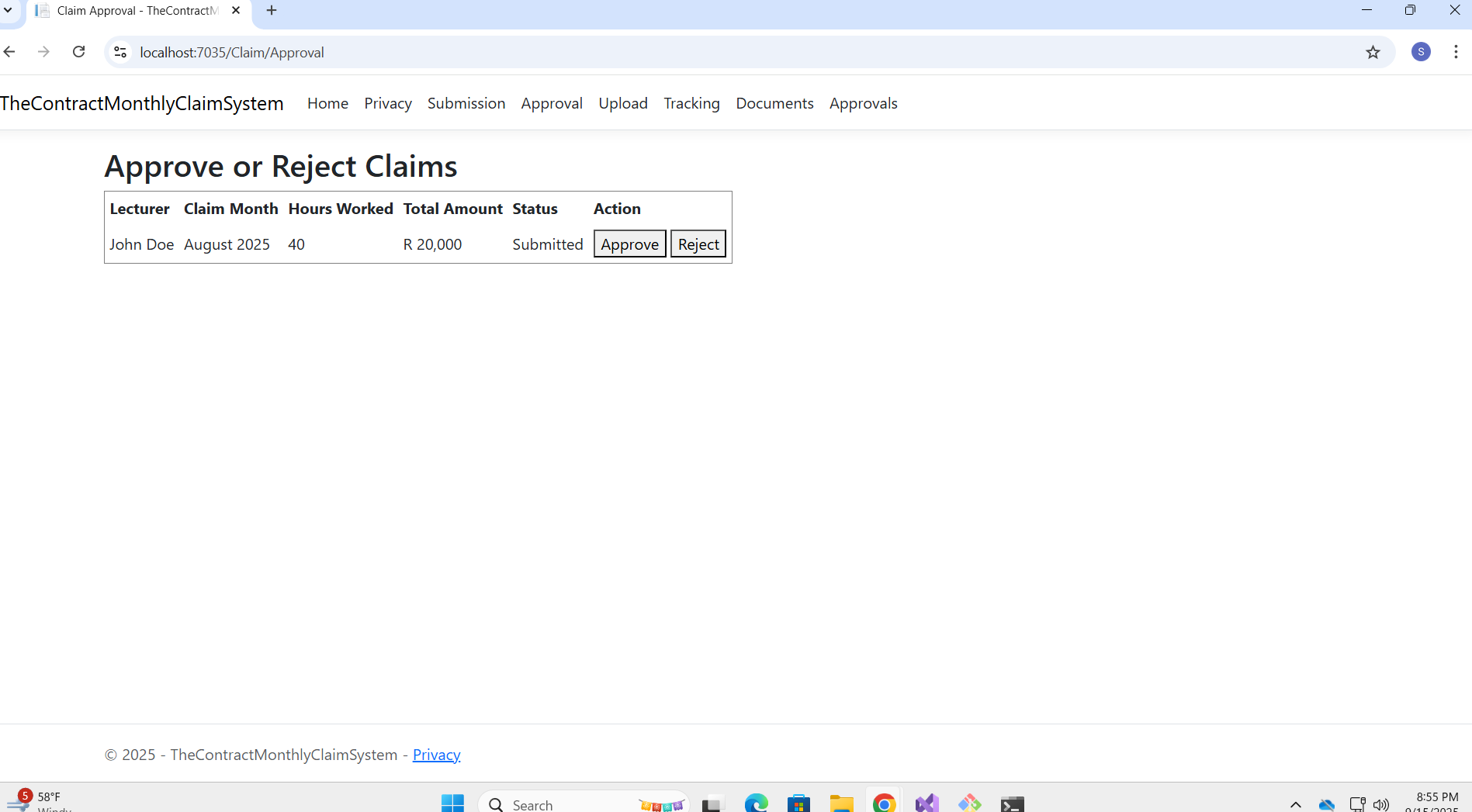


## Graphical User Interface (Prototype)



A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

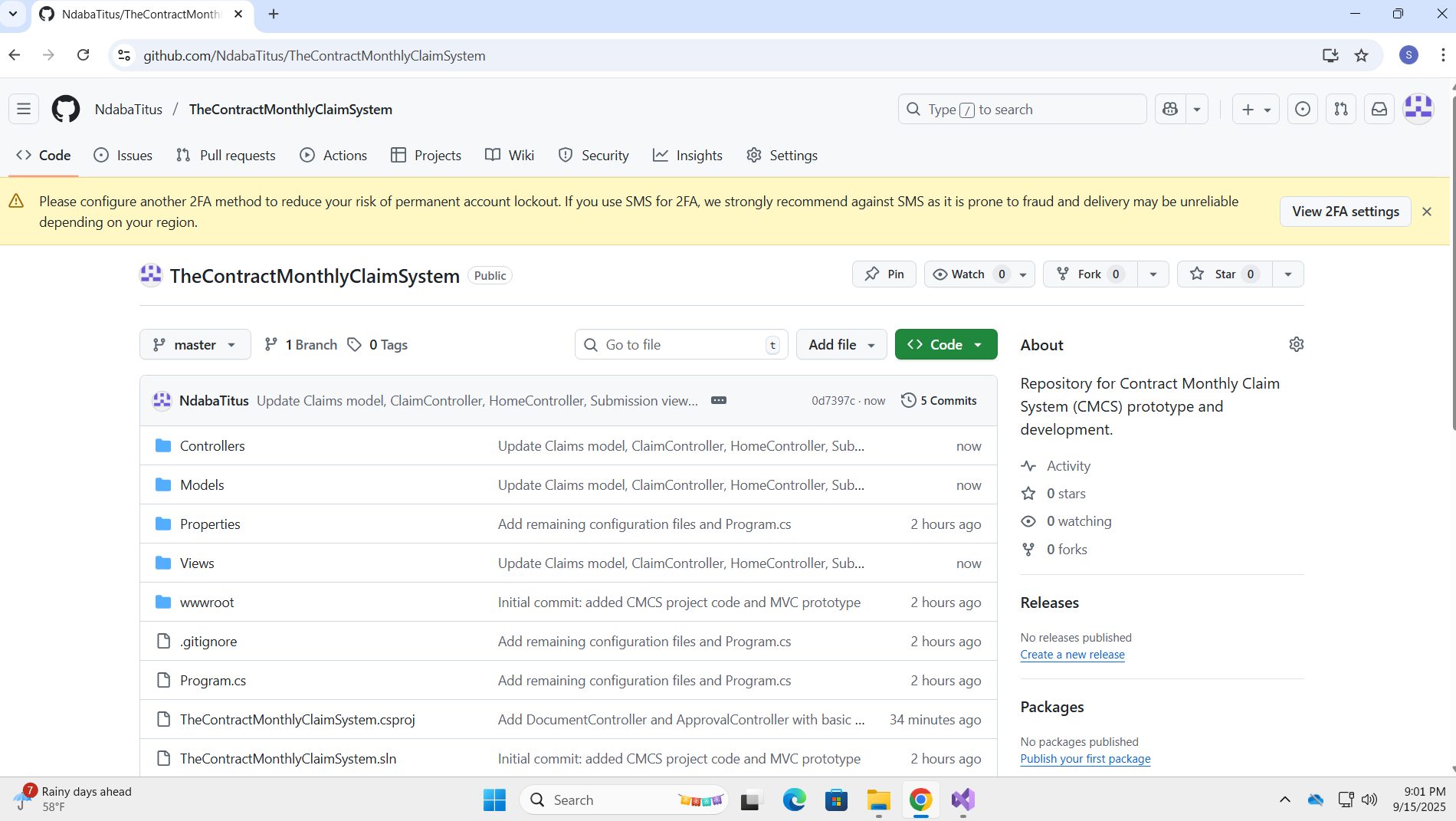
Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



## Assumptions and Constraints

In this prototype, it is presumed that lecturers will enter claims into the system, while administrative personnel will assess and authorize them. The claims procedure depends on established forms and paperwork, including timesheets and corroborative documentation. Constraints involve the non-functional aspect of this Part 1 prototype, indicating that live database integration, real-time alerts, and complete user authentication are not currently in place. The existing design is restricted to local testing and demonstration functions, lacking integration with external systems or automated validation beyond fundamental form checks.

## Conclusion

In the first part of the Contract Monthly Claim System (CMCS) project, we created a prototype graphical user interface with ASP.NET Core MVC, featuring screens for Claim Submission, Approval, Document Upload, and Tracking. This visual prototype showcases the system's planned workflow and user-friendliness, offering a distinct framework for lecturers, coordinators, and managers. It establishes a solid base for future growth by outlining layout, navigation, and user interactions, ensuring effective and consistent integration of database features, automated calculations, and claim validation.