PROG 6212

POE PART 2

Sajana Bidesi
ST10249843

TABLE OF CONTENTS

TABLE OF CONTENTS 1
LINKS 2
CONTRACT MONTHLY CLAIM SYSTEM (CMCS) 3
WHAT IS CMCS? 3
PURPOSE AND OBJECTIVES 3
- Improved Efficiency: 3
- Faster Claim Processing: 3
- Precise Data Entry and Calculations: 3
- Enhanced Transparency: 3

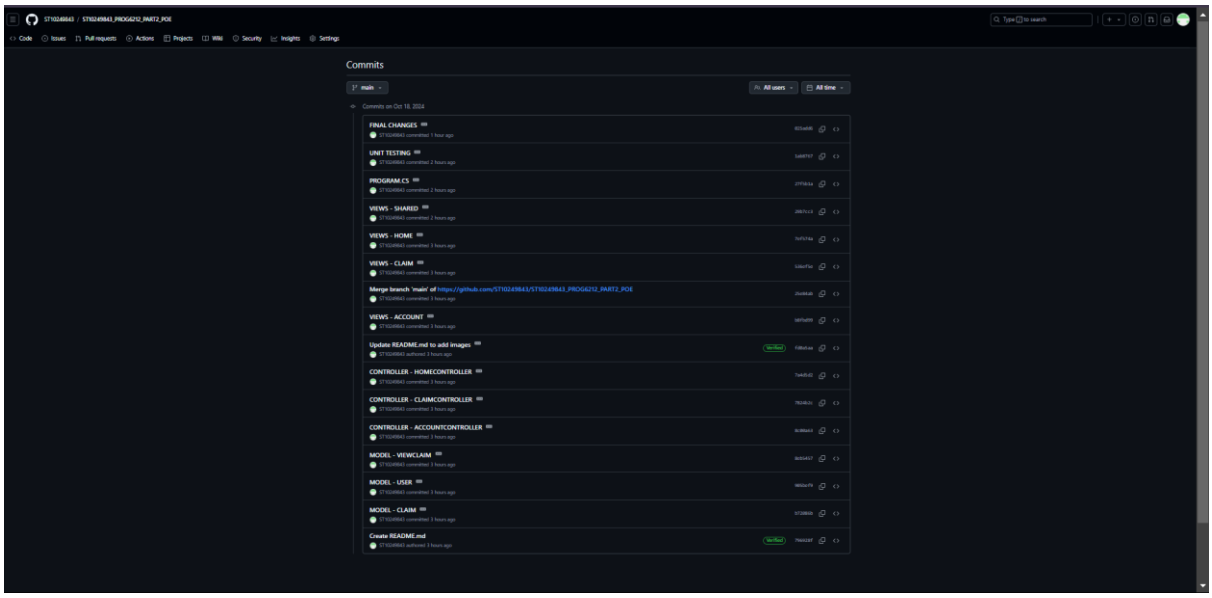
KEY FEATURES 4
CLAIM SUBMISSION 4
CLAIM VERIFICATION AND APPROVAL 4
STATUS MONITORING 4
ADMINISTRATIVE MANAGEMENT 4
PROJECT OVERVIEW 5
TARGET AUDIENCE 5
SYSTEM REQUIREMENTS 6
HARDWARE REQUIREMENTS 6
SOFTWARE REQUIREMENTS 6
TECHNOLOGY USED 7
INSTALLATION INSTRUCTIONS 8
TESTING INSTRUCTIONS 8
Unit Testing: 8
TECHNICAL ASPECTS 9
WEB-BASED APPLICATION 9
BUILT WITH .NET CORE MVC 9
COLOUR PALETTE 10
CONCLUSION 10
REFERENCES 11

LINKS

GITHUB: https://github.com/ST10249843/ST10249843_PROG6212_PART2_POE.git

YOUTUBE: https://youtu.be/S1aupcfLCDI

GITHUB COMMIT HISTORY



CONTRACT MONTHLY CLAIM SYSTEM (CMCS)

WHAT IS CMCS?

The Contract Monthly Claim System (CMCS) is a sophisticated web-based application specifically designed to simplify and enhance the management of monthly claims submitted by Independent Contractor (IC) lecturers. As educational institutions increasingly rely on IC lecturers to provide quality instruction, the CMCS serves as a vital tool to streamline the often complex and labour-intensive task of processing claims for services rendered.

This application not only addresses the administrative burdens associated with claim submissions but also ensures that all stakeholders have a transparent and efficient system to track and manage claims.

PURPOSE AND OBJECTIVES

The primary purpose of the CMCS is to enhance the submission and approval process of claims, optimizing administrative efficiency and ensuring accuracy throughout the workflow. The key objectives include:

- Improved Efficiency: The CMCS automates routine tasks related to claim processing, thereby reducing manual effort and the potential for human error.
- Faster Claim Processing: By streamlining the workflow, the application reduces the time taken for claims to be reviewed and approved, allowing lecturers to receive timely compensation for their services.
- Precise Data Entry and Calculations: The system ensures accurate data entry and calculations, which are critical for maintaining the integrity of the claims process and ensuring that payments are correct.
- Enhanced Transparency: The CMCS offers stakeholders visibility into the status of claim processing, from submission to final approval or rejection, fostering trust and accountability.

KEY FEATURES

CLAIM SUBMISSION

- Intuitive Interface: The CMCS provides a user-friendly interface that allows lecturers to submit claims with ease. This is crucial for ensuring that users of varying technical skills can navigate the system effectively.
- Essential Information Capture: The system captures vital information such as hours worked, hourly rates, as well as any other important information through an online form, ensuring all necessary data is collected upfront.
- Document Upload: Lecturers can upload supporting documents like timesheets or invoices, which are essential for substantiating their claims and ensuring comprehensive review.

CLAIM VERIFICATION AND APPROVAL

- Automated Workflows: The CMCS introduces automated workflows that enhance the verification and approval processes, minimizing manual intervention and streamlining communication between users.
- Systematic Routing: Claims are automatically routed to Programme Coordinators and Academic Managers for processing, ensuring that each claim is reviewed by the appropriate personnel.
- Decision Recording: All decisions made regarding claims (approvals, rejections) are recorded and tracked within the system to ensure comprehensive transparency and accountability for all stakeholders involved.

STATUS MONITORING

- Real-Time Tracking: Users can track the status of their claims in real-time, providing both lecturers and administrators with clear visibility of where each claim stands in the approval process.

ADMINISTRATIVE MANAGEMENT

- Data Management Capabilities: CMCS empowers Human Resources staff and administrators with a wide range of data management functionalities, including maintaining lecturer profiles and claim histories.
- Seamless Database Integration: The system can connect seamlessly with existing databases, allowing for better insights and facilitating informed decision-making based on historical data and trends.

PROJECT OVERVIEW

TARGET AUDIENCE

The CMCS caters to a diverse group of users within educational institutions, each with distinct roles and responsibilities:

- Lecturers: As the primary users of the system, lecturers are responsible for submitting claims, attaching supporting documents, and providing detailed information regarding their services. The system simplifies this process, ensuring timely and accurate submissions.
- Programme Coordinators: These users are tasked with scrutinizing claims to ensure accuracy and adherence to institutional standards. They verify that all claims submitted meet the necessary criteria before forwarding them for final approval.
- Academic Managers: Serving as the final approvers, Academic Managers review verified claims and make decisions based on established guidelines. Their oversight ensures that claims are managed with full accountability and a high level of scrutiny.
- Human Resources Staff: HR personnel manage the administrative tasks associated with maintaining lecturer data and processing claims. They utilize the CMCS to monitor claim histories, updates, and compliance reporting, promoting accuracy and efficiency in recordkeeping.

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

To ensure optimal performance of the CMCS, the following hardware specifications are recommended:

Processor: 1 GHz or faster

RAM: 2 GB or more

Hard Disk Space: Minimum of 500 MB of free space

Display: 1024 x 768 resolution or higher

SOFTWARE REQUIREMENTS

The following software components are necessary for the successful deployment and operation of the CMCS:

Operating System: Windows 7 or later

.NET Framework: Version 4.7.2 or later (will be installed with the application)

IDE: Visual Studio (for development and maintenance)

Database Management: SQL Server Management Studio (SSMS)

Version Control: GitHub (for source code management)

Framework: .NET Core for MVC application development

TECHNOLOGY USED

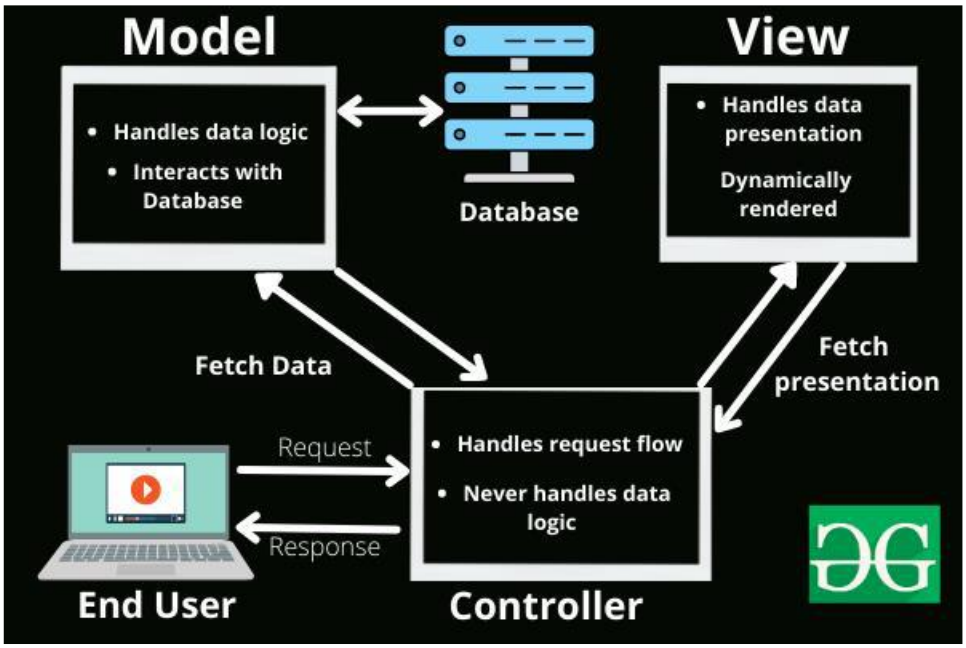
Programming Language: C#

Framework: ASP.NET Core MVC

Database: SQL Server (for data management)

Version Control: Git for source code management

Web Technologies: HTML, CSS, JavaScript for front-end development



INSTALLATION INSTRUCTIONS

To install the CMCS, follow these steps:

1. Download the Application: Obtain the CMCS source code from the GitHub repository
2. Install Dependencies: Ensure that you have .NET Core SDK installed on your machine. You can download it from the .NET website.
3. Install SQL Server Management Studio (SSMS) to manage the database.
4. Build the Application:
5. Open the solution in Visual Studio and build the application by selecting Build > Build Solution.
6. Run the Application:
7. Start the application using Ctrl + F5 or by clicking the Start button in Visual Studio. This will launch the application in your default web browser.

TESTING INSTRUCTIONS

To ensure the CMCS functions as intended, follow these testing instructions:

Unit Testing:

Utilize a testing framework like NUnit or XUnit to create and run unit tests for individual components of the application.

Ensure that all core functionalities, such as claim submission, verification, and approval workflows, are covered by unit tests.

TECHNICAL ASPECTS

WEB-BASED APPLICATION

The CMCS is designed as a web-based application, ensuring accessibility from any device with an internet connection. This flexibility allows users to engage with the system from various locations, enhancing usability and ensuring that critical functionalities are always available.

BUILT WITH .NET CORE MVC

Developed using the .NET Core framework with ASP.NET MVC (Model-View-Controller) architecture, CMCS provides robust performance and scalability. This structure allows for effective management of user interactions and data processing, improving both the development process and overall user experience.

COLOUR PALETTE

Brown: Provides grounding warmth and stability, akin to the comforting scent of leather-bound books.

Black: Introduces bold sophistication, ensuring clarity and precision in navigation.

Green: Symbolizes growth and renewal, highlighting key actions within the interface.

White: Serves as a clean backdrop, allowing elements to stand out with elegance.

CONCLUSION

The Contract Monthly Claim System (CMCS) is an essential tool for managing claims efficiently within educational institutions. By automating processes and enhancing transparency, CMCS not only benefits lecturers but also streamlines administrative tasks, paving the way for better resource management and improved institutional efficiency.

REFERENCES

FreeFrontend (2023). *29 CSS Login / Registration Forms*. [online] Free Frontend. Available at: https://freefrontend.com/css-login-forms/.

GeeksforGeeks (2018). *C# | List Class*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/c-sharp-list-class/.

Hadiya, Y. (2022). *Create Login (Sign In) And Registration (Sign Up) Form In C# Windows Form With Database*. [online] www.c-sharpcorner.com. Available at: https://www.c-sharpcorner.com/article/create-loginsign-in-and-registration-sign-up-form-in-c-sharp-windows-form-with-da/.

HaiyingYu (2022). *Upload a file to a Web site by using Visual C# - ASP.NET*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/troubleshoot/developer/webapps/aspnet/development/upload-file-to-web-site.

Rick-Anderson (2023). *Role-based authorization in ASP.NET Core*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/aspnet/core/security/authorization/roles?view=aspnetcore-8.0.

tdykstra (2023). *Handle errors in ASP.NET Core*. [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/aspnet/core/fundamentals/error-handling?view=aspnetcore-8.0.

W3Schools (n.d.). *How To Filter Elements*. [online] www.w3schools.com. Available at: https://www.w3schools.com/howto/howto_js_filter_elements.asp.

w3schools (2019). *CSS Website Layout*. [online] W3schools.com. Available at: https://www.w3schools.com/css/css_website_layout.asp.