PROG6212-Part1

~ST10446545~

Contents

Introduction

CMCS is a .NET web-based tool that simplifies submitting and approving monthly claims for Independent Contractor lecturers. Right now, lecturers struggle with repetitive paperwork, slow approval times, and unclear claim status updates. CMCS offers a centralized, easy-to-use system that boosts efficiency, ensures accurate claims, and increases transparency in the submission process.

Purpose of the system:

- Enable lecturers to submit claims quickly with accurate details.
- Allow Programme Coordinators and Academic Managers to review, verify, and approve claims.
- Provide functionality for lecturers to upload supporting documents.
- Automate administrative processes to reduce human error and improve efficiency.

Key users:

- Lecturers: Submit claims and supporting documents.
- Programme Coordinators: Review claims for accuracy.
- Academic Managers: Approve or reject claims based on verification.

Scope of Part 1:
This part concentrates on planning and design. It covers documenting design decisions, creating a UML class diagram for the database, creating a project plan, and designing a prototype GUI/UI. At this stage, no backend functionality will be implemented.

Design Choices

Structure & Layout

CMCS will use a Model-View-Controller (MVC) architecture to keep logic, data, and presentation separate. The layout will include:

- Navigation menu with options: Dashboard, Submit Claim, Uploaded Documents, Approvals, Profile.
- Dashboard: Overview of claim statuses and notifications.
- Submit Claim Page: Form-based interface for claim entry.
- Approvals Page: For coordinators/managers to approve or reject claims.

Colour Scheme

- Blue (#0047AB): Represents trust, professionalism, and reliability.
- White background: Improves readability and reduces eye strain.
- Green highlights (#28A745): Indicate success and approved claims.
- Red highlights (#DC3545): Indicate errors or rejected claims.

These colors provide clear visual feedback for users and follow professional system design standards.

Assumptions

- Lecturers submit accurate and honest claim details.
- Claims will be submitted monthly without exception.
- Users have access to reliable internet connections.

Constraints

- File uploads are limited to .pdf, .docx, and .xlsx formats.
- The system is designed primarily for desktop use, with limited mobile responsiveness in the prototype stage.
- Only Programme Coordinators and Academic Managers have claim approval permissions.

UML Class Diagram for Database

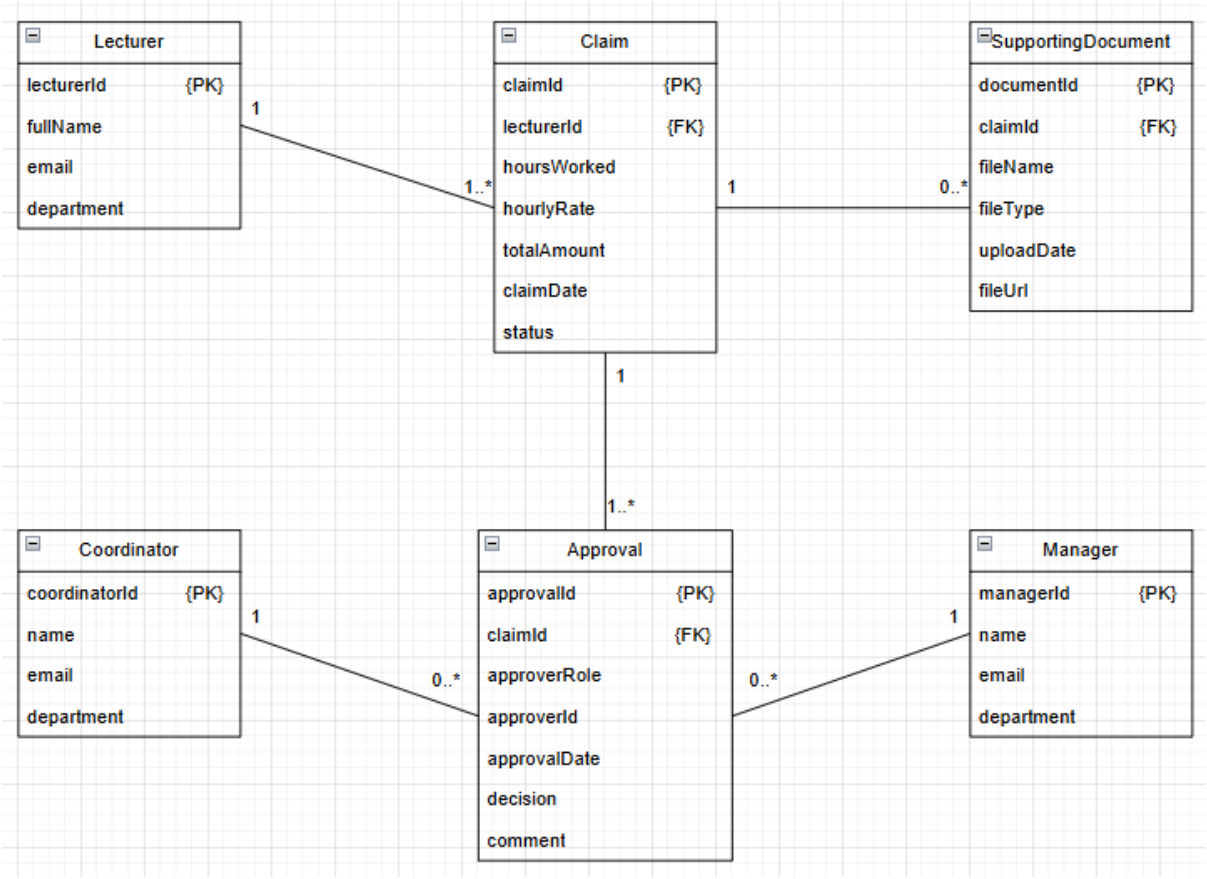

Figure 1: ST10446545_UML_Diagram

The UML class diagram outlines the key entities of the Contract Monthly Claim System (CMCS) and their relationships. A Lecturer (lecturerId) can submit multiple Claims, which include attributes like hoursWorked, hourlyRate, totalAmount, claimDate, and status. Each claim may have several SupportingDocuments (fileName, fileType, uploadDate, fileUrl) to ensure transparency. Claims go through an approval workflow depicted by the Approval entity, which records approverRole, approverId, approvalDate, decision, and comment.

Both Coordinators and Managers can create multiple approvals for a claim, reflecting a multi-stage verification process. The diagram distinguishes between primary keys (PK) and foreign keys (FK) for database integrity. Multiplicity notation shows that one lecturer can have many claims, a claim can include multiple documents, and approvals must be linked to a coordinator or manager. This design effectively models the monthly claim process and sets a solid groundwork for future implementation.

Project Plan

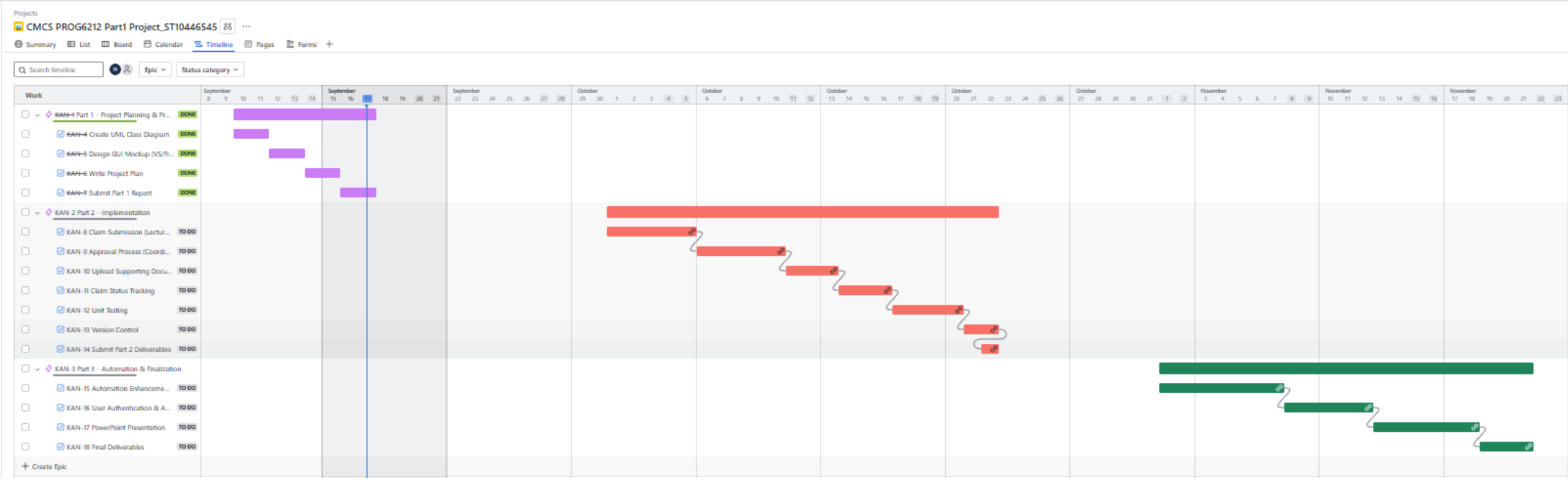

Figure 2: ST10446545_Jira_ProjectPlan

GUI Design

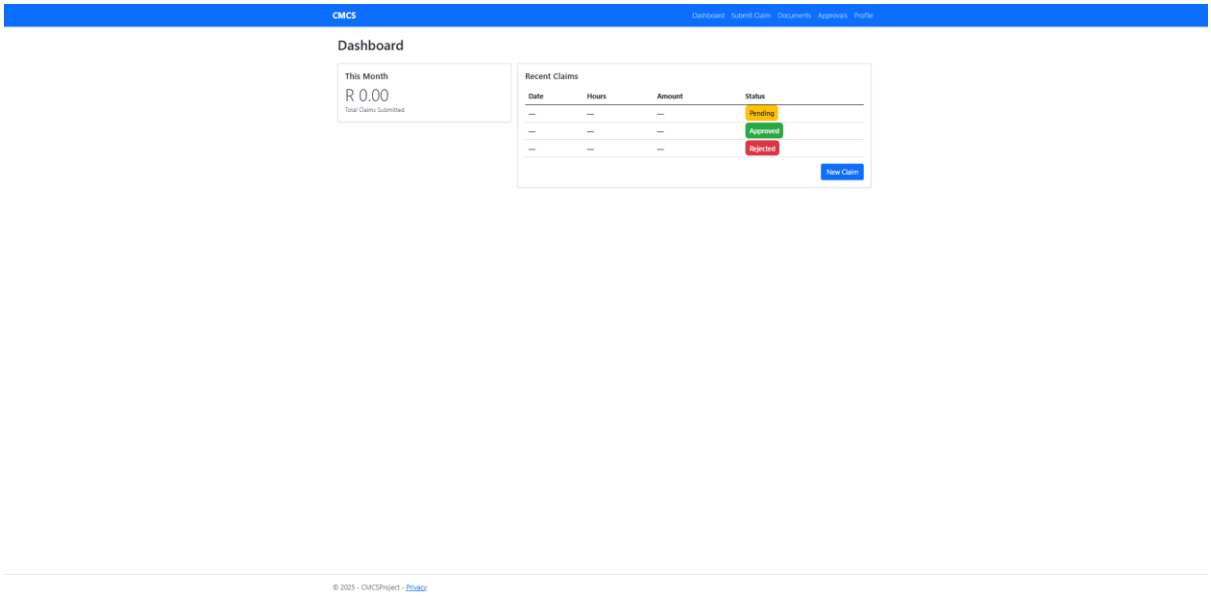

Figure 3: ST10446545_GUI_DashBoard

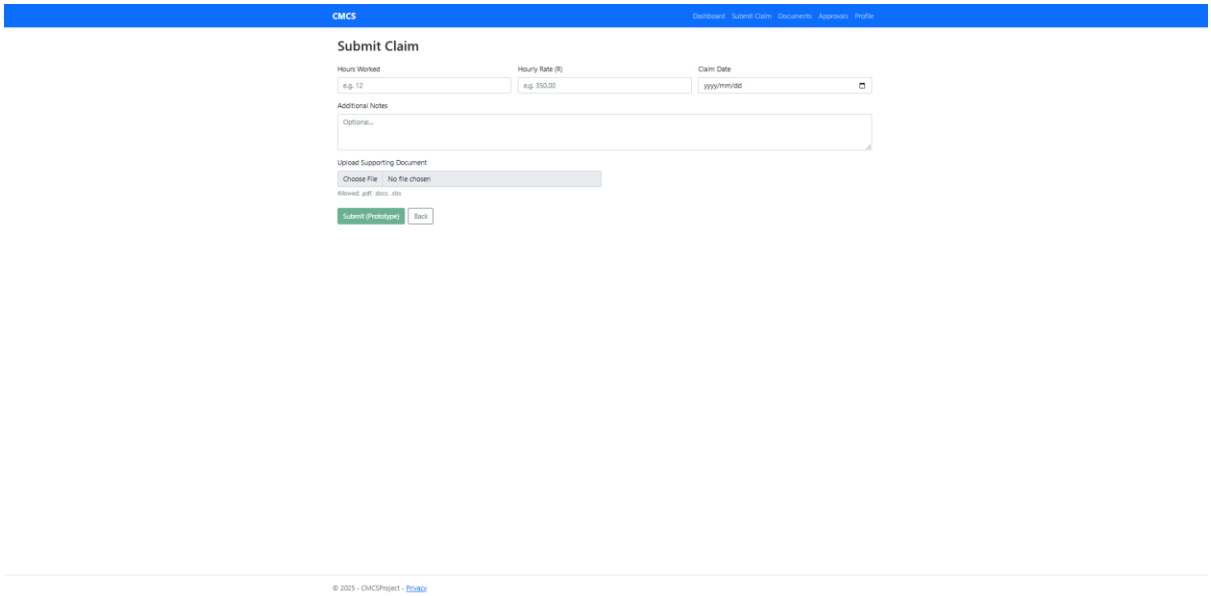

Figure 4: ST10446545_SubmitClaim

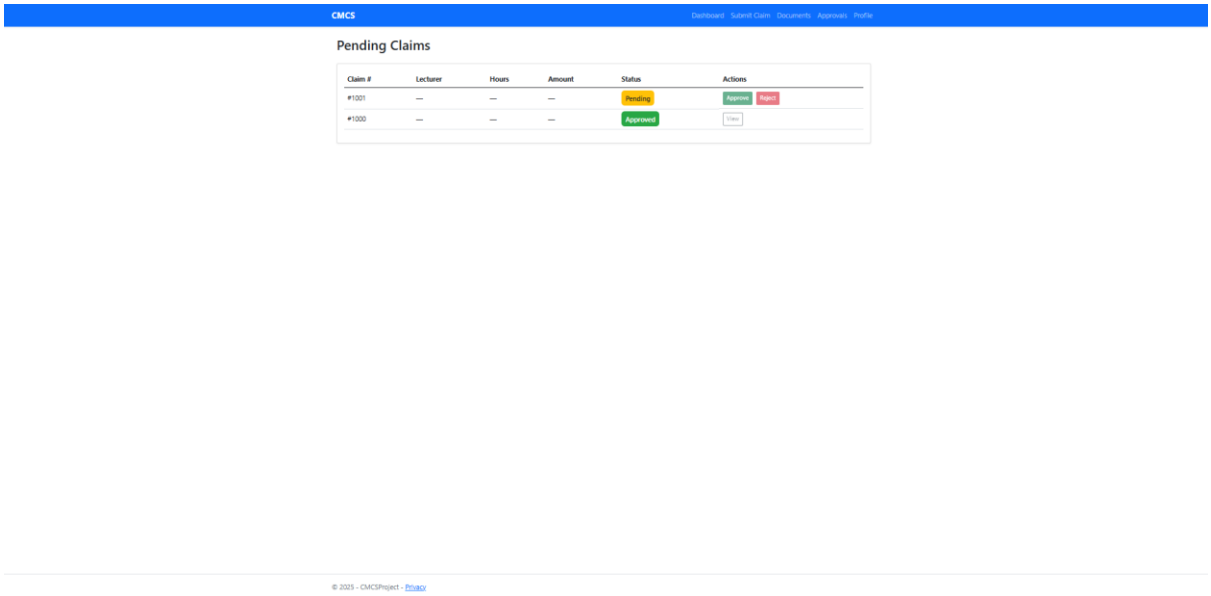

Figure 5: ST1044654_Claims

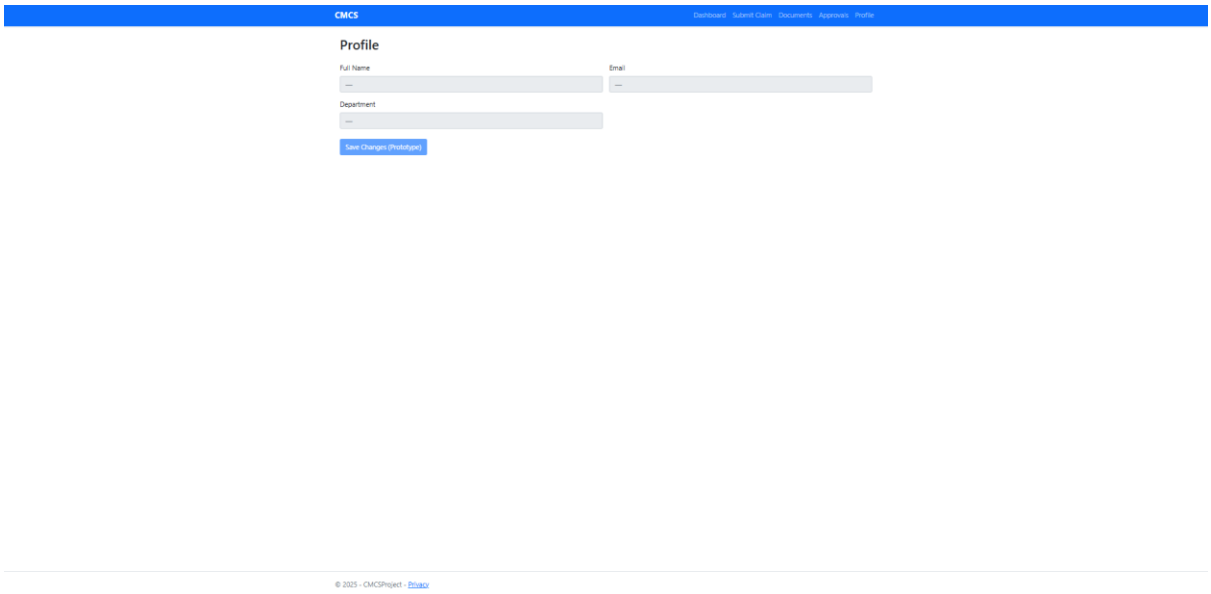

Figure 6: ST10446545_Profile

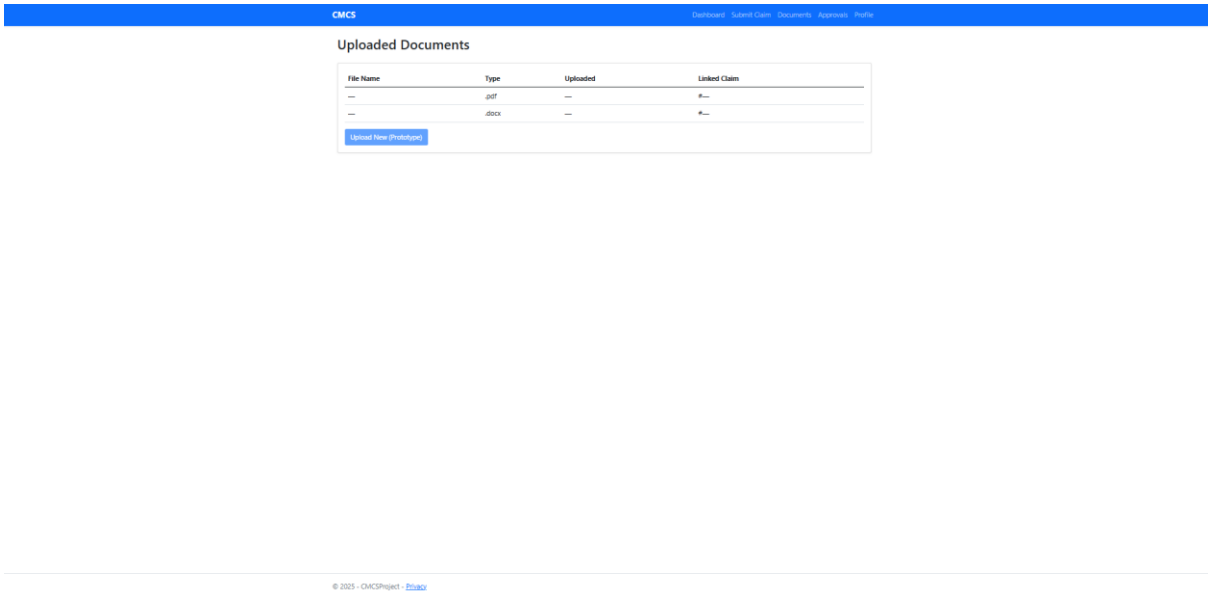

Figure 7: ST10446545_Documents

GUI Prototype

My current GUI design is a basic prototype created in Visual Studio. Its goal is to show the layout, navigation, and key components of the CMCS system, not to create a fully working app at this point.

- Dashboard Page: Shows a summary of key information (like claims overview and recent activity) using cards, tables, and sample data. This sets the stage for how end users will view system metrics.
- Submit Claim Page: Offers a straightforward input form layout for submitting claims. Although the form elements aren't linked to backend logic yet, they show where users will enter their information.
- Uploaded Documents Page: Shows a table-style layout with placeholder rows for files, representing how document management will be presented.
- Approvals Page: Uses a similar table layout with mock “Approve”/“Reject” buttons to visualize the approval workflow.
- Profile Page: Displays user information in a structured format, acting as a placeholder for personal and professional details.

Every page uses Bootstrap and shared CSS styling (like colored badges for claim status) to maintain consistency. For now, the prototype doesn't include any real data — its main goal is to show how the application will look and feel when it's fully implemented in Part 2 of the project.

References

OpenAI. (2025). ChatGPT conversation share: Colours and custom badges. Available at: https://chatgpt.com/share/68cade7e-7858-8008-9341-9ee725b43ce5 (Accessed: 17 September 2025).

OpenAI. (2025). ChatGPT conversation share: Gantt chart and project planning guidance. Available at: https://chatgpt.com/share/68cae5ab-0dd0-8008-b9fc-30b9e122903c (Accessed: 17 September 2025).

diagrams.net. (2025). diagrams.net – Free online diagram software. Available at: https://app.diagrams.net/ (Accessed: 17 September 2025).

Atlassian. (2025). Jira Software – Project management tool. Available at: https://www.atlassian.com/software/jira (Accessed: 17 September 2025).

OpenAI. (2025). ChatGPT conversation share: GUI prototype and design guidance. Available at: https://chatgpt.com/share/68caeee0-0910-8008-9007-630faf547119 (Accessed: 17 September 2025).

AI Usage and Disclaimer

Sections

Part1- Colours and custom badges(GUI)

Part1 - Gantt chart and project planning guidance

Part1 - GUI prototype example and design guidance.

AI Tools

ChatGPT - 5

Its Purpose

Part 1 – Colours and custom badges (GUI)
Purpose: ChatGPT was used to provide CSS styling examples for custom badges (e.g., pending, approved, rejected). This helped establish a visual identity and consistent styling theme for the GUI prototype.

Part 1 – Gantt chart and project planning guidance
Purpose: ChatGPT assisted in explaining how to create a Gantt chart in Jira, including setting up tasks, dependencies, and timelines. The purpose was to support structured project planning and ensure tasks followed a clear sequence.

Part 1 – GUI Design Guidance
Purpose: ChatGPT offered examples and suggestions for creating a static prototype in Visual Studio using .cshtml pages. This provided a basic design guide to help me visualize my system layout before adding full functionality.

Evidence

Colours and custom badges - https://chatgpt.com/share/68cade7e-7858-8008-9341-9ee725b43ce5

Gantt chart and project planning guidance - https://chatgpt.com/share/68cae5ab-0dd0-8008-b9fc-30b9e122903c

GUI design guidance - https://chatgpt.com/share/68caeee0-0910-8008-9007-630faf547119