# PROG6212

ST10257779

## Table of Contents

# AI Declaration and Design:

## Declaration on Generative AI and Sources:

Writing and Scope:
The domain logic and core solution were written by me. This covers the application's fundamental implementation, routing, view models, and the general ASP.NET Core MVC framework. I am solely accountable for the final deliverables, architecture, and functionality.

## How I Employed ChatGPT's AI:

I simply utilised ChatGPT (GPT-5 Thinking, 2025 version) as a helper to polish my already written code. My objectives were to:

- Update and simplify the Razor (.cshtml) user interface and CSS for a neater, more standardised design.
- To make the interface read and act more professionally, improve the text, space, and component organisation.
- To improve the GUI and layout while maintaining control over the application's logic and design decisions, I regenerate specific UI sections (using my own original code).
- Receive recommendations for non-essential fixes that didn't alter my planned behaviour as well as minor code cleanliness (names, formatting, and minor refactors).
- I modified and altered the results as ChatGPT suggested regenerated snippets. The final implementation is different from the raw AI suggestions if you compare them to the code; I only included the elements that enhanced readability, structure, or UI refinement and matched my own design.

### What I Wrote vs. Where AI Helped:

Areas with AI assistance:

- Claims pages (UI refinement and layout enhancements)
  Approval pages (refinements in structure, readability, and style)
- Home layout (tweaks to the typography, section alignment, and landing page design)
  Independently written by me:
- Generate pages (such as Create.cshtml and the controller/model logic that supports it).
- Generate pages (such as Create.cshtml and the controller/model logic that supports it).
- Details pages (such as Details.cshtml and the logic that supports controllers and models)
- I wrote the domain models, view models, controllers, and the original Razor structure, which form the foundation of the entire code. In order to enhance the layout and the GUI as a whole, I subsequently used ChatGPT to regenerate and refine the UI sections.

### Additional Learning Resources:

To gain a better understanding of the patterns and strategies for a C# MVC implementation, I watched older PROG6212 lesson videos on YouTube for a related project. I used these movies to support my arguments and validate patterns rather than copying code word for word.

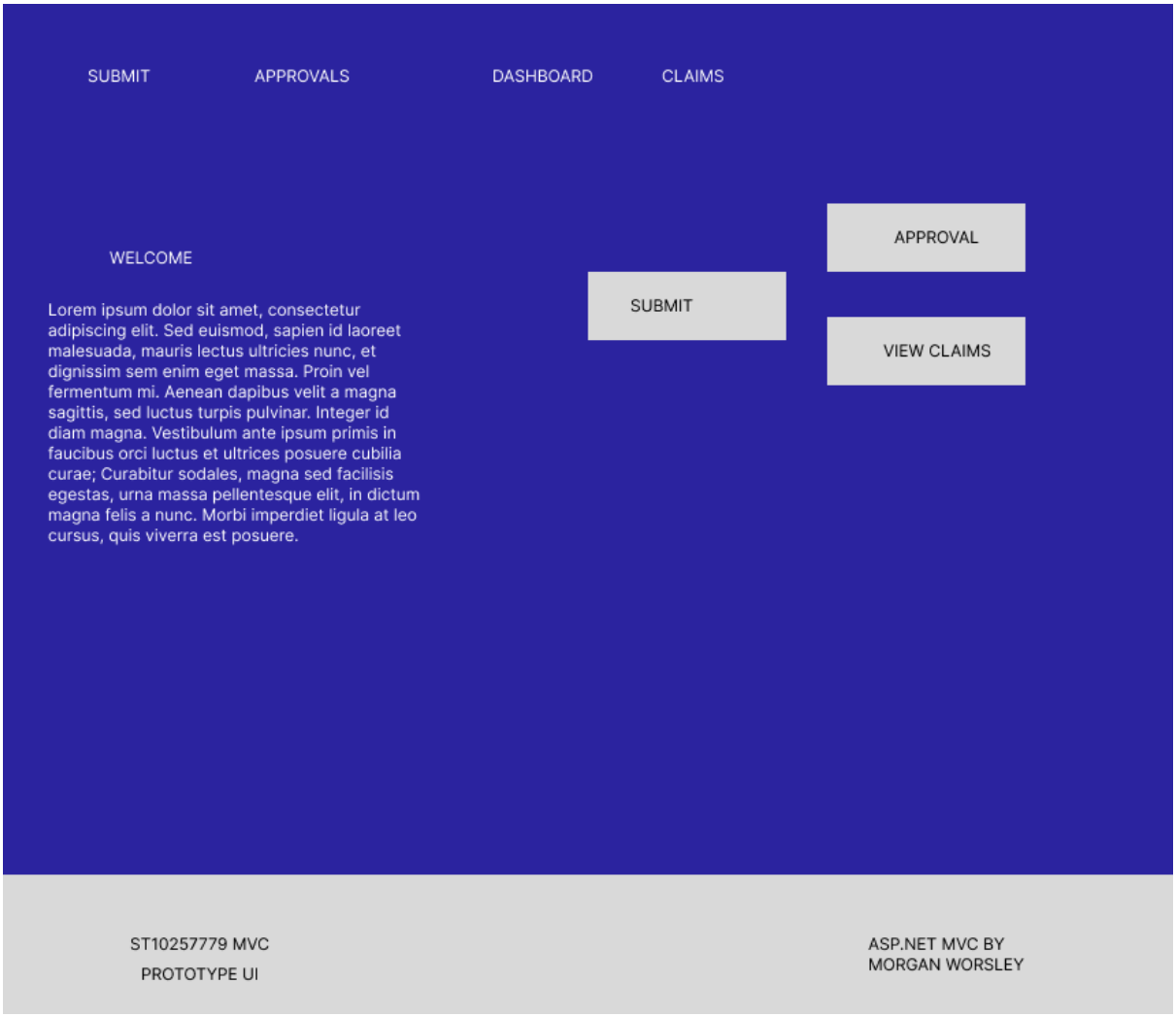
To make sure the project is completely working, I looked to Stack Overflow for focused fixes and particular C# error resolutions. These were carefully added and modified for my codebase.
The ChatGPT (GPT-5 Thinking) 2025 tool and version were used.

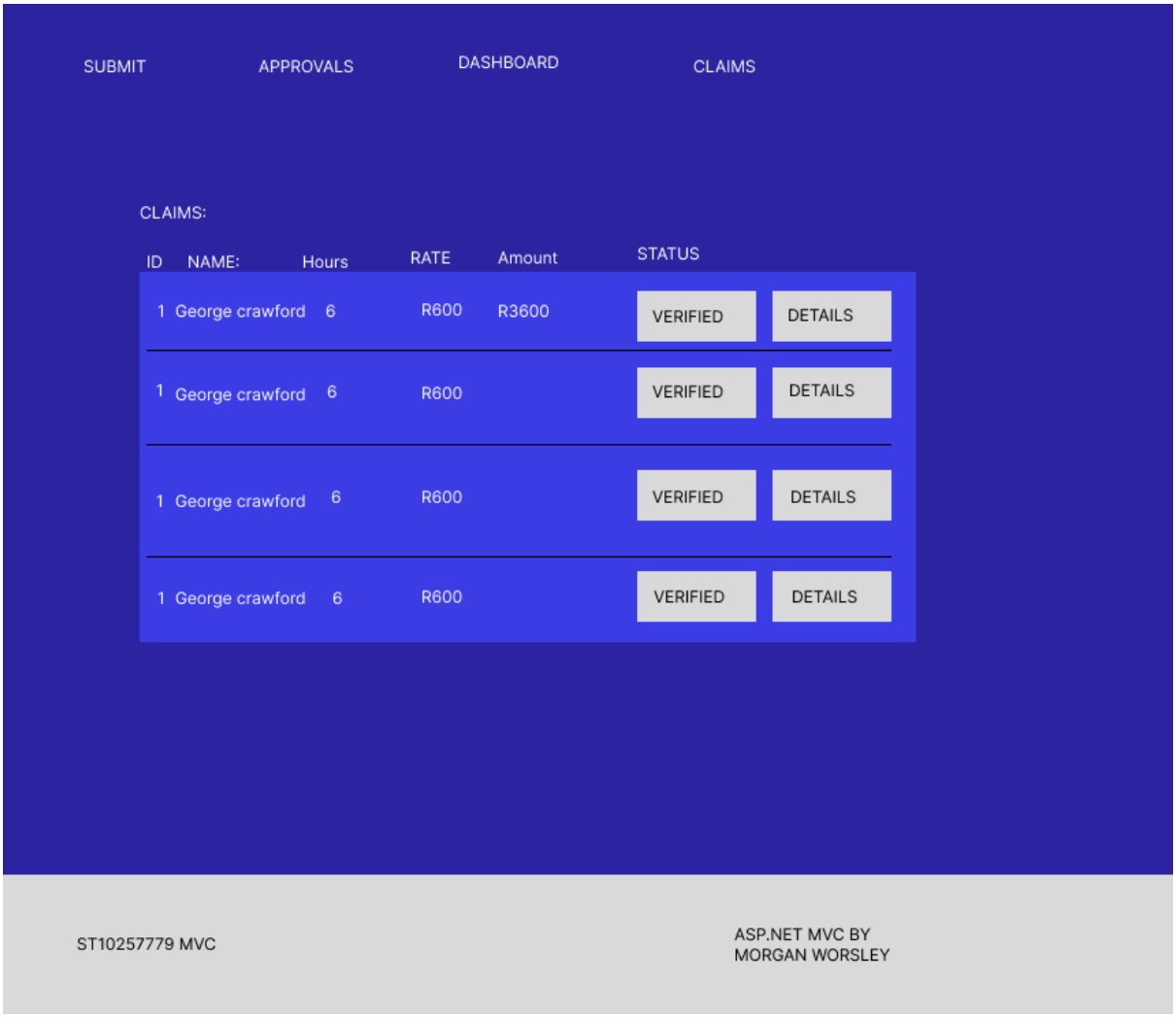
### The Citation and Attribution:

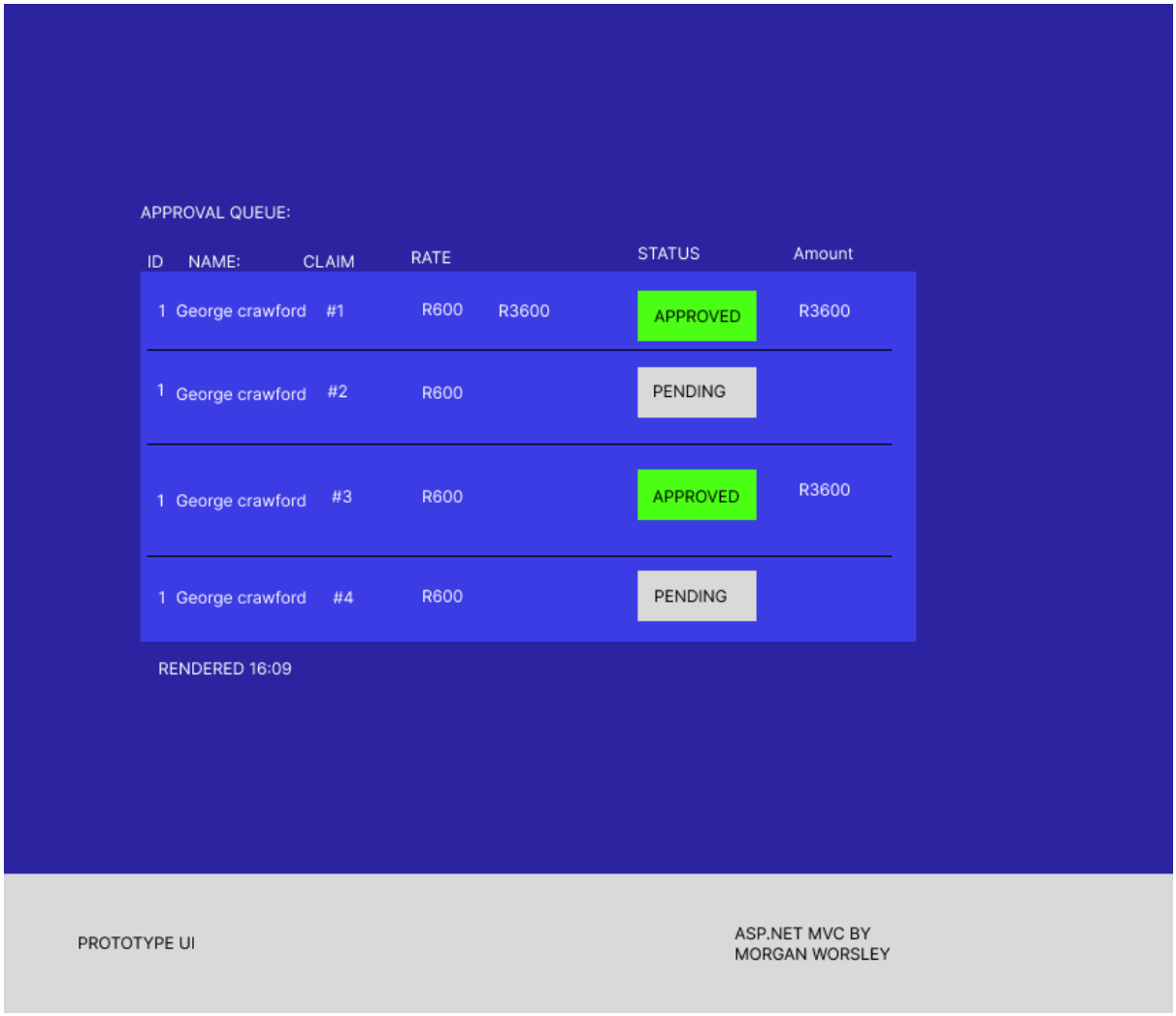
I acknowledged the usage of Stack Overflow and ChatGPT in project-level comments in my C# code.
I cited these references at the project level rather than within each Razor file because inline citations are impractical index.cshtml markup for every minor layout change.

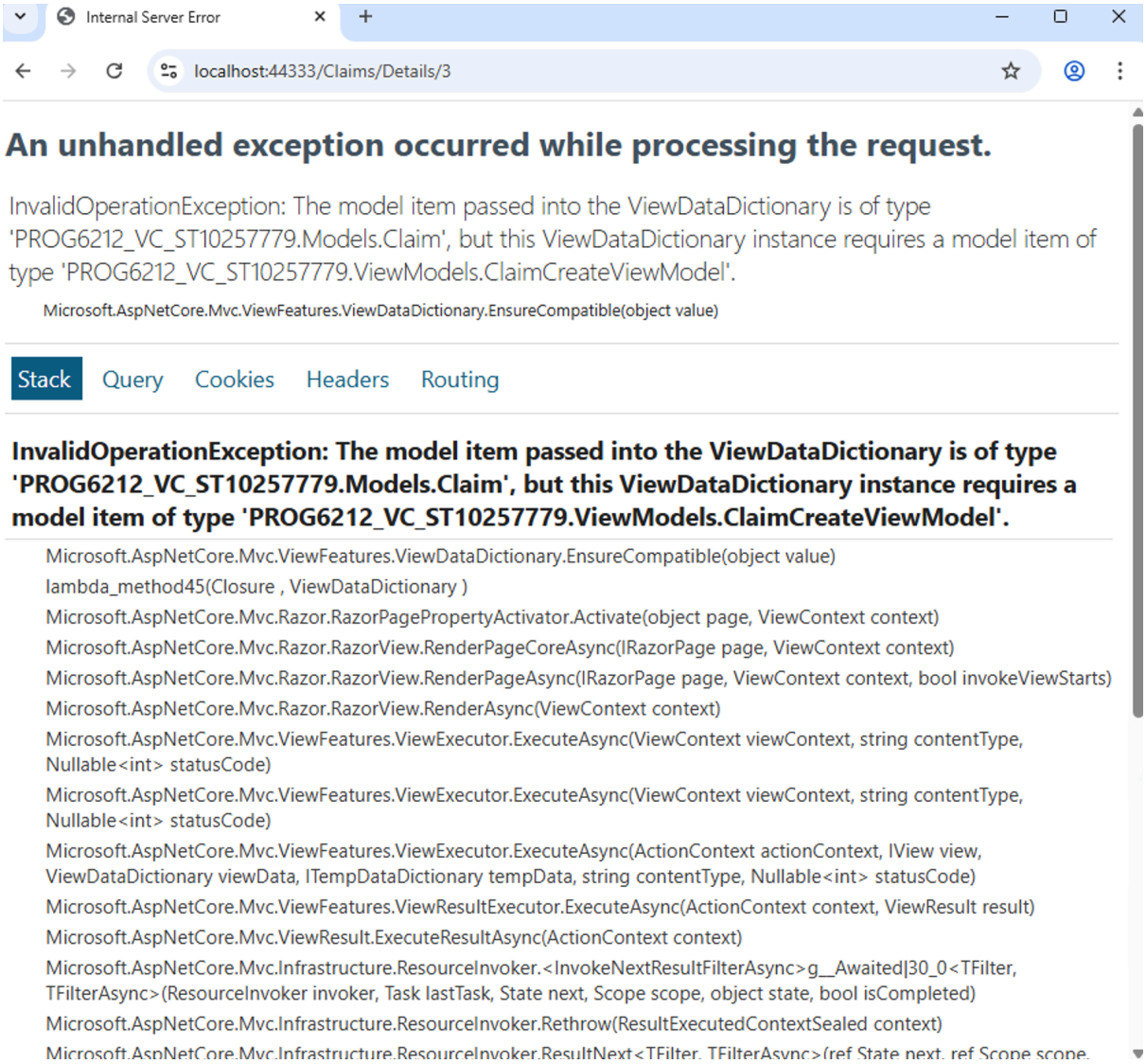
## Integrity Statement

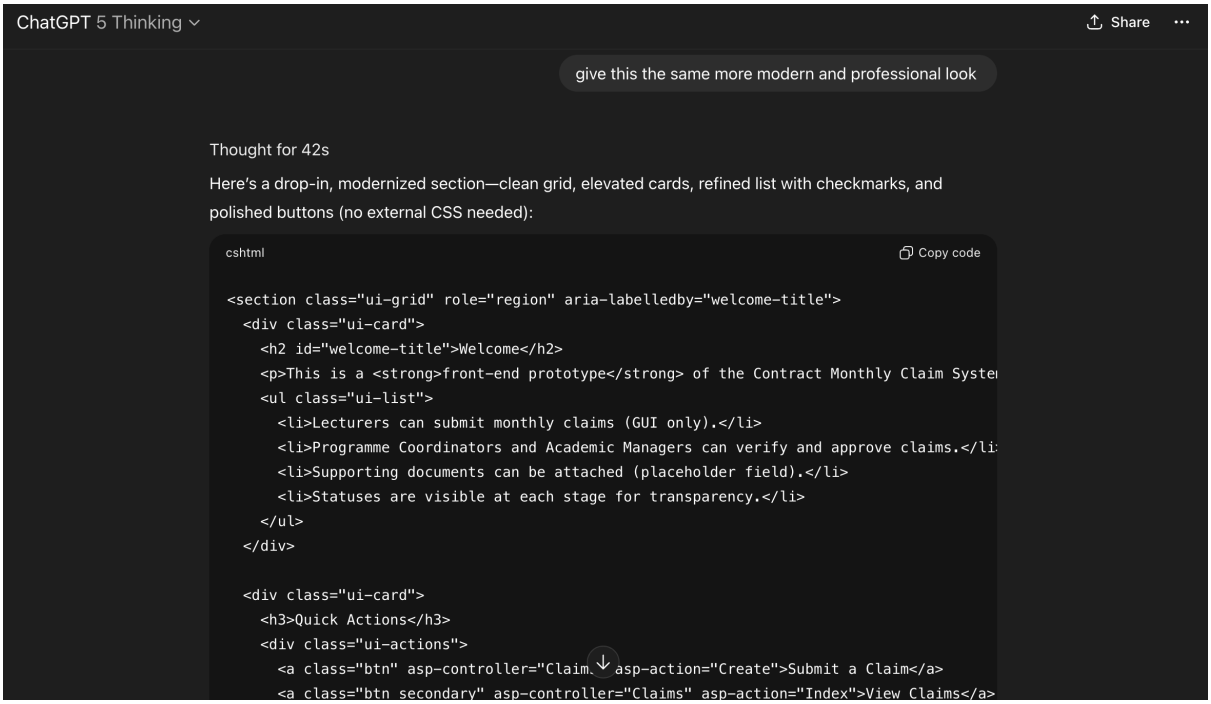
- I acknowledged the usage of Stack Overflow and ChatGPT in project-level comments in my C# files.
- I cited these sources at the project level rather than within each Razor file because inline citations are impractical inside.cshtml markup for every minor layout change.


SUBMIT
APPROVALS
DASHBOARD
CLAIMS
WELCOME
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod, sapien id laoreet malesuada, mauris lectus ultricies nunc, et dignissim sem enim eget massa. Proin vel fermentum mi. Aenean dapibus velit a magna sagittis, sed luctus turpis pulvinar. Integer id diam magna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Curabitur sodales, magna sed facilisis egestas, urna massa pellentesque elit, in dictum magna felis a nunc. Morbi imperdiet ligula at leo cursus, quis viverra est posuere.
SUBMIT
APPROVAL
VIEW CLAIMS
ST10257779 MVC
PROTOTYPE UI
ASP.NET MVC BY
MORGAN WORSLEY


SUBMIT
APPROVALS
DASHBOARD
CLAIMS
CLAIMS:
ID NAME: Hours RATE Amount STATUS
1 George crawford 6 R600 R3600 VERIFIED DETAILS
1 George crawford 6 R600 VERIFIED DETAILS
1 George crawford 6 R600 VERIFIED DETAILS
1 George crawford 6 R600 VERIFIED DETAILS
ST10257779 MVC
ASP.NET MVC BY
MORGAN WORSLEY


APPROVAL QUEUE:
ID NAME: CLAIM RATE STATUS Amount
1 George crawford #1 R600 R3600 APPROVED R3600
1 George crawford #2 R600 PENDING
1 George crawford #3 R600 APPROVED R3600
1 George crawford #4 R600 PENDING
RENDERED 16:09
PROTOTYPE UI
ASP.NET MVC BY
MORGAN WORSLEY

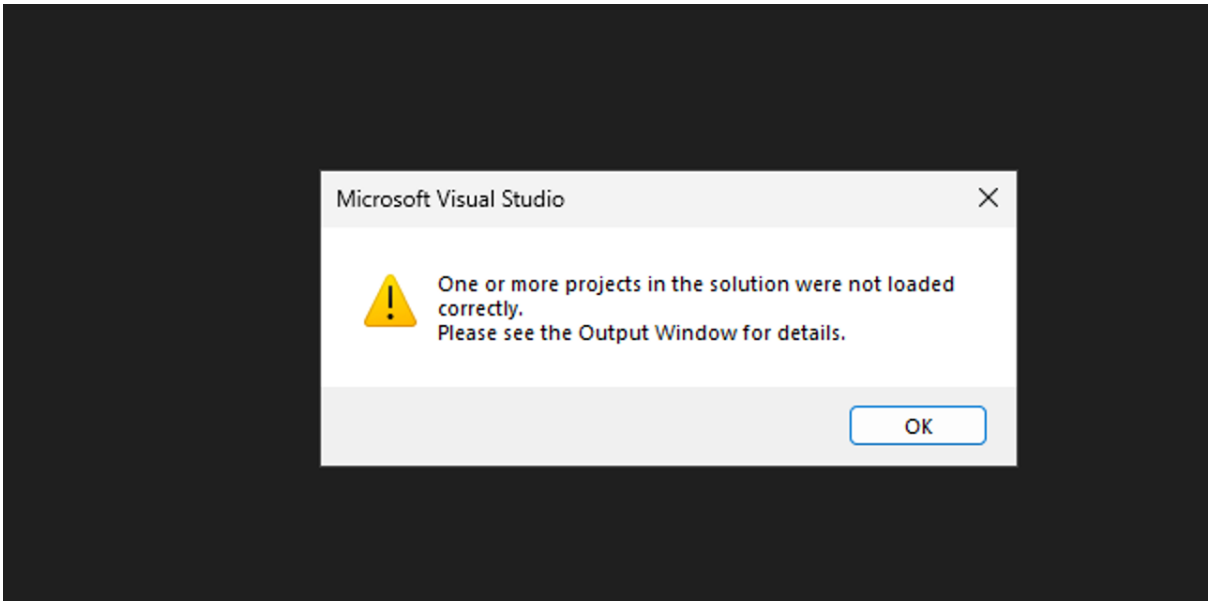

Internal Server Error
localhost:44333/Claims/Details/3
An unhandled exception occurred while processing the request.
InvalidOperationException: The model item passed into the ViewDataDictionary is of type 'PROG6212_VC_ST10257779.Models.Claim', but this ViewDataDictionary instance requires a model item of type 'PROG6212_VC_ST10257779.ViewModels.ClaimCreateViewModel'.
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary.EnsureCompatible(object value)
Stack
Query
Cookies
Headers
Routing
InvalidOperationException: The model item passed into the ViewDataDictionary is of type 'PROG6212_VC_ST10257779.Models.Claim', but this ViewDataDictionary instance requires a model item of type 'PROG6212_VC_ST10257779.ViewModels.ClaimCreateViewModel'.
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewDataDictionary.EnsureCompatible(object value)
lambda_method45(Closure , ViewDataDictionary )
Microsoft.AspNetCore.Mvc.Razor.RazorPagePropertyActivator.Activate(object page, ViewContext context)
Microsoft.AspNetCore.Mvc.Razor.RazorView.RenderPageCoreAsync(IRazorPage page, ViewContext context)
Microsoft.AspNetCore.Mvc.Razor.RazorView.RenderPageAsync(IRazorPage page, ViewContext context, bool invokeViewStarts)
Microsoft.AspNetCore.Mvc.Razor.RazorView.RenderAsync(ViewContext context)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewExecutor.ExecuteAsync(ViewContext viewContext, string contentType, Nullable<int> statusCode)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewExecutor.ExecuteAsync(ViewContext viewContext, string contentType, Nullable<int> statusCode)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewExecutor.ExecuteAsync(ActionContext actionContext, IView view, ViewDataDictionary viewData, ITempDataDictionary tempData, string contentType, Nullable<int> statusCode)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewResultExecutor.ExecuteAsync(ActionContext context, ViewResult result)
Microsoft.AspNetCore.Mvc.ViewResult.ExecuteResultAsync(ActionContext context)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeNextResultFilterAsync>g__Awaited|30_0<TFilter, TFilterAsync>(ResourceInvoker invoker, Task lastTask, State next, Scope scope, object state, bool isCompleted)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.Rethrow(ResultExecutedContextSealed context)


ChatGPT 5 Thinking
Share
give this the same more modern and professional look
Thought for 42s
Here's a drop-in, modernized section—clean grid, elevated cards, refined list with checkmarks, and polished buttons (no external CSS needed):
cshtml
Copy code
<section class="ui-grid" role="region" aria-labelledby="welcome-title">

<h2 id="welcome-title">Welcome</h2>
<p>This is a front-end prototype of the Contract Monthly Claim Syste
<ul class="ui-list">
<li>Lecturers can submit monthly claims (GUI only).</li>
<li>Programme Coordinators and Academic Managers can verify and approve claims.</li
<li>Supporting documents can be attached (placeholder field).</li>
<li>Statuses are visible at each stage for transparency.</li>
</ul>

<h3>Quick Actions</h3>

<a class="btn" asp-controller="Claim" asp-action="Create">Submit a Claim</a>
<a class="btn secondary" asp-controller="Claims" asp-action="Index">View Claims</a>




## Purpose

Design choices: The prototype is implemented with ASP.NET Core MVC (.NET 8). MVC was chosen to mirror the final solution architecture and to cleanly separate concerns: controllers handle the web flow, strongly typed models represent domain data, and Razor views render the interface. No database is used yet — the app seeds in-memory data so that the GUI can run 100% without extra packages. This matches the rubric's requirement that the application should not be functional at
this stage while still compiling and presenting all screens.

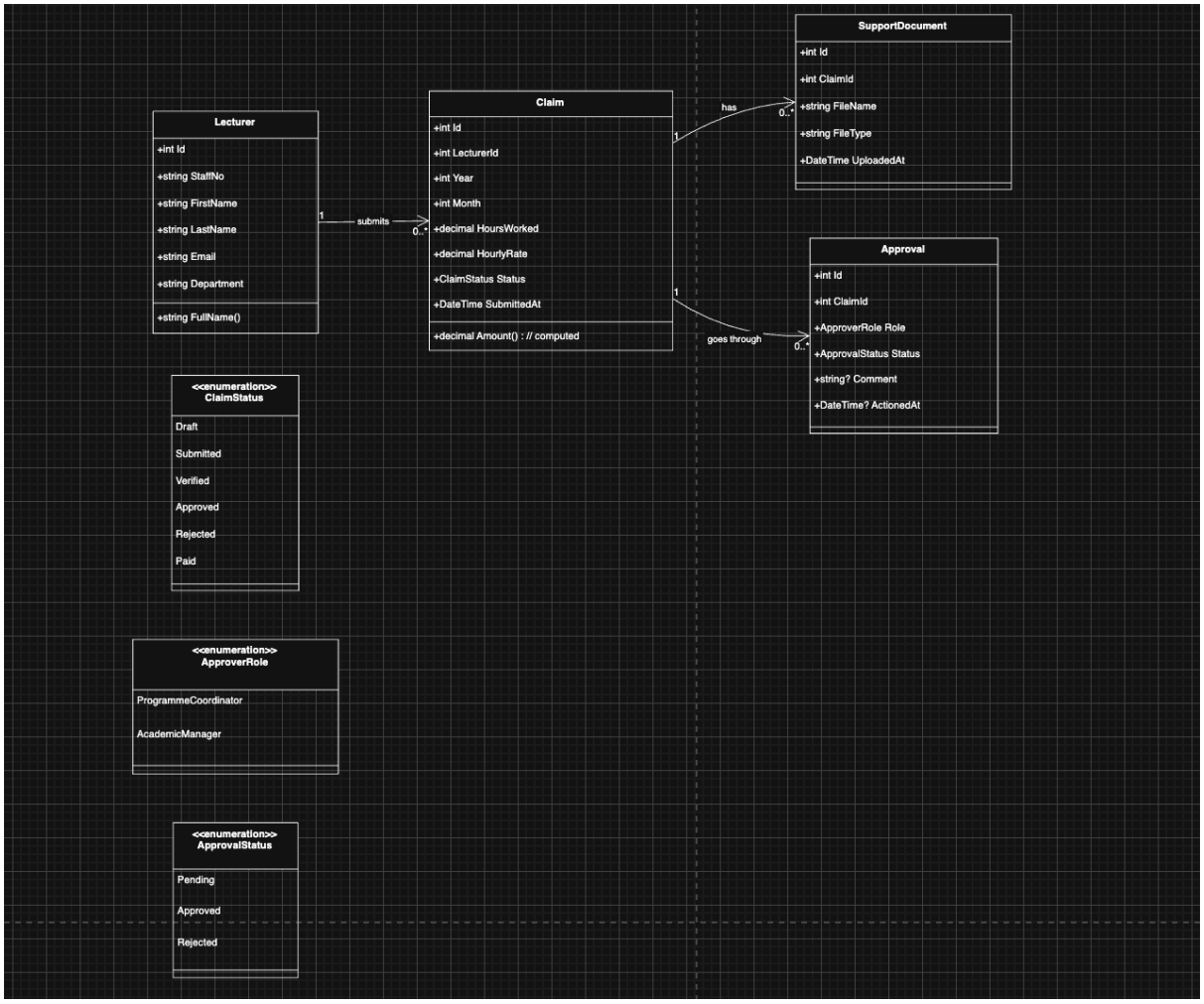
The planned database structure is: A lecturer may make a variety of claims. Each claim will go through two approval processes—the Academic Manager and the Program Coordinator—and may include numerous supporting documents. The claim includes the month, year, number of hours worked, hourly rate, and a calculated sum. Progress of status values Draft → Confirmed → Accepted → Denied → Accrued. Acceptances timestamp, comment, store role, and status.

GUI design: The Dashboard, Claims List, Give Claim, and Approval Queue are all accessible via the top navigation. Lecturer, time, hours, rate, and a placeholder file picker for supporting documentation are all recorded in the Submit Claim form. With a Details page that summaries papers and the approval timeline, the Claims list displays the time limit, totals, and status. The page for the Approval Queue

lists both positions completed and pending approvals. Data is not saved; all screens are front-end only. Premises and limitations: (a) Each lecturer may submit one claim per month, (b) papers are submitted as PDFs or photos, (c) the institution provides

academic roles and names, (d) ZAR is the currency, and (e) accessibility aims for WCAG AA for keyboard navigation and colour contrast. At this point, no EF or diagnostic packages are installed.

## UML Class Diagram for Databases



## Project Plan

| Task | Dependency | Owner | Timeline |
|---|---|---|---|
| The Plan scope & rubric mapping | Planning | Student | Day 1 |
| Design data model plus UML | Scope | Student | Day 1–2 |
| Scaffold MVC project + navigation | Design | Student | Day 2 |
| Implementation of GUI pages (Claims, Create, Details, Approvals) | Scaffold | Student | Day 3–4 |
| Style and accessibility (CSS, keyboard, labels) | GUI pages | Student | Day 4 |
| Preparation of documentation and Word report | All of screens | Student | Day 5 |
| Version control: 5 commits & pushes | Ongoing | Student | Throughout progression |

## Version control:

- feat(Ui): upload placeholders; submit claim form plus client validation
- Ui: approvals queue and status badges; - docs: add project plan, UML, and Part 1
- Word report
- layout polish
- Task:.NET 8 MVC project and solution skeletal scaffolding .
- In-memory seed and model additions; claims list and information

## GUI / UI coverage (per rubric)

- Lecturers can use a specific form (front-end only) to submit claims.
- Academic managers and program coordinators can examine a queue for approval or verification.
- A placeholder, non-persistent upload field for supplementary documents.
- For clear tracking, claim status is displayed on the list and details pages.
- Consistent validation, labels, and layout for an easy-to-use interface.
- Reference’s

# GitHub Link:

https://github.com/ihategithub691738/PROG6212-VC-ST10257779

# Reference's and AI reference


ChatGpt. (2025, 09 16). *Modernised look for homepage*. Retrieved from Chatgpt: https://chatgpt.com/share/68caba44-a844-800e-9ed9-206583c756eb

OpenAI. (2025, 09 16). *making my code look above standard and fixing details*. Retrieved from Chatgpt: https://chatgpt.com/share/68caba44-a844-800e-9ed9-206583c756eb

STACKFLOW. (2018, 04 20). *While I switched my MVC Web Application from ISS Express to Local ISS error*. Retrieved from STACKFLOW: https://stackoverflow.com/questions/50076033/while-i-switched-my-mvc-web-application-from-iss-express-to-local-iss-error

STACKFLOW. (2023, 05 04). *Directing user to another page in ASP.Net MVC application leads to syntax error in auto-generated code*. Retrieved from STACKFLOW: https://stackoverflow.com/questions/67550980/directing-user-to-another-page-in-asp-net-mvc-application-leads-to-syntax-error

Youtube. (2021, 09 18). *PROG6212 Help Video*. Retrieved from Youtube: https://www.youtube.com/watch?v=ynCfI0zy0OE