PROG6212 2B – Final PoE

Name: Rahil Ramdass

Student Number: ST10434494

Module Name: Programming 2B

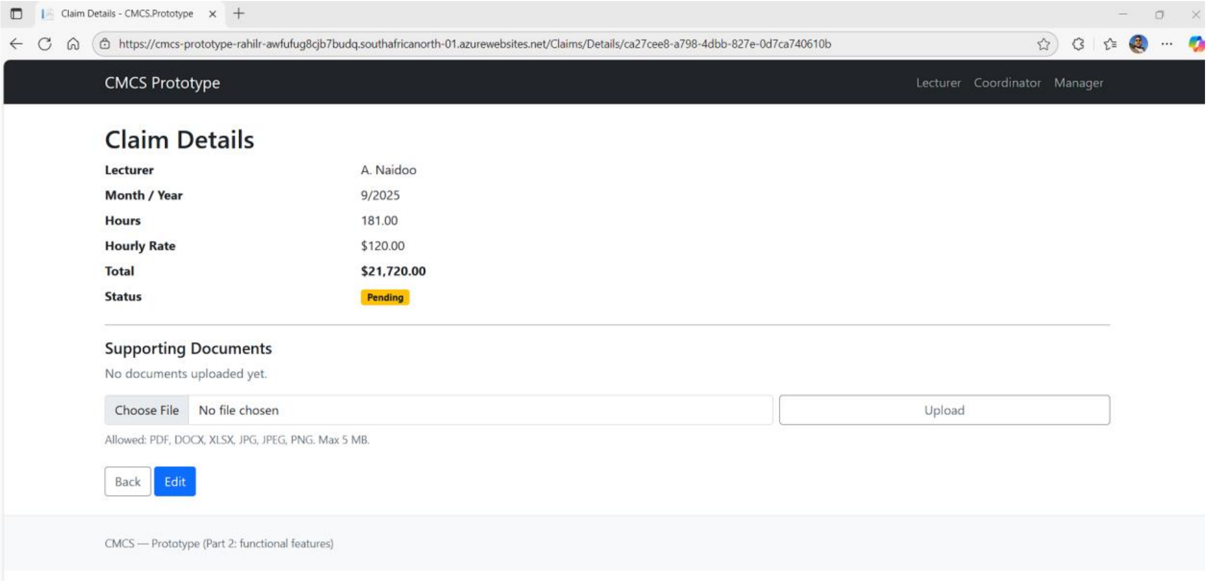
Date: 23 September 2025

Contents

1.

Lecturer View Automation:

- Feature Implemented: Lecturers can submit their claims by entering hours worked and hourly rate (Microsoft, 2024).
- Automation:
    - Auto-calculation of Total Payment = Hours Worked × Hourly Rate.
    - Validation checks included (e.g., no negative hours, valid numeric inputs) (W3Schools, 2025).
- Tools Used: ASP.NET Core MVC, Entity Framework Core for persistence, and jQuery for client-side validation and auto-calculation (Microsoft, 2024).
- Outcome: Ensures accuracy and prevents invalid submissions.


Claim Details - CMCS.Prototype
https://cmcs-prototype-rahilr-awfufug8cjb7budq.southafricanorth-01.azurewebsites.net/Claims/Details/ca27cee8-a798-4dbb-827e-0d7ca740610b
CMCS Prototype
Lecturer
Coordinator
Manager
Choose File
No file chosen
Upload
Back
Edit

Claim Details

Lecturer	A. Naidoo
Month / Year	9/2025
Hours	181.00
Hourly Rate	$120.00
Total	**$21,720.00**
Status	Pending

Supporting Documents

No documents uploaded yet.

Allowed: PDF, DOCX, XLSX, JPG, JPEG, PNG. Max 5 MB.

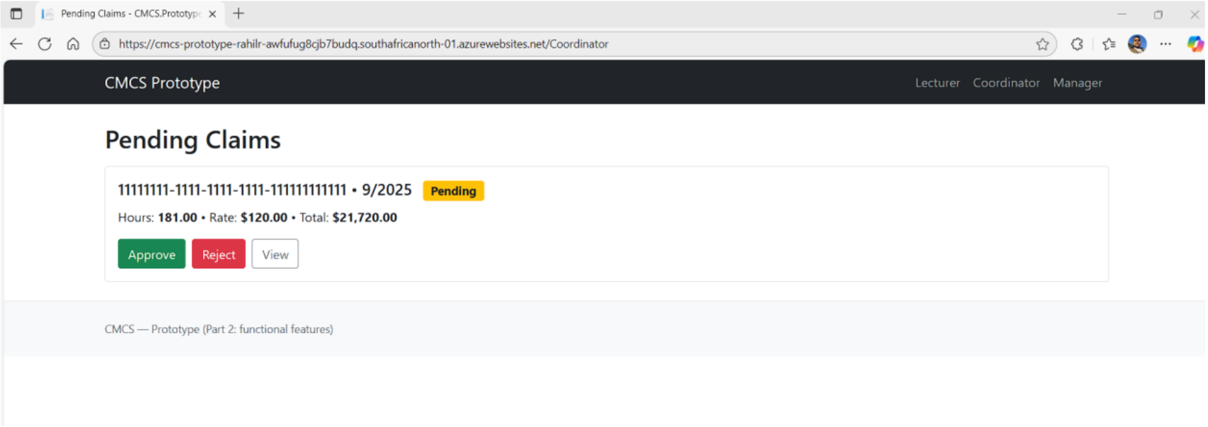
CMCS — Prototype (Part 2: functional features)

- Lecturer submission form with auto-calculation in action.

Programme Coordinator and Academic Manager View Automation:

- Feature Implemented: Coordinators and managers can view pending claims and either approve or reject them. This role-based workflow is handled with ASP.NET Core Identity (Microsoft, 2024).
- Automation:
    - Automated validation against business rules (hours, rates) reduces human error and ensures policies are applied consistently (Chacon & Straub, 2014).
    - Workflow process: once approved/rejected, status updates automatically. (Microsoft, 2024).
- Tools Used: ASP.NET Identity for role-based access, Entity Framework for claim management, and ASP.NET Web API for backend workflows (Microsoft, 2024).

- Outcome: Streamlined approval process, reduced manual checking, improved accuracy.



Pending Claims - CMCS.Prototype
https://cmcs-prototype-rahilr-awfufug8cjb7budq.southafricanorth-01.azurewebsites.net/Coordinator
Lecturer
Coordinator
Manager

CMCS Prototype

Pending Claims

11111111-1111-1111-1111-111111111111 • 9/2025 Pending

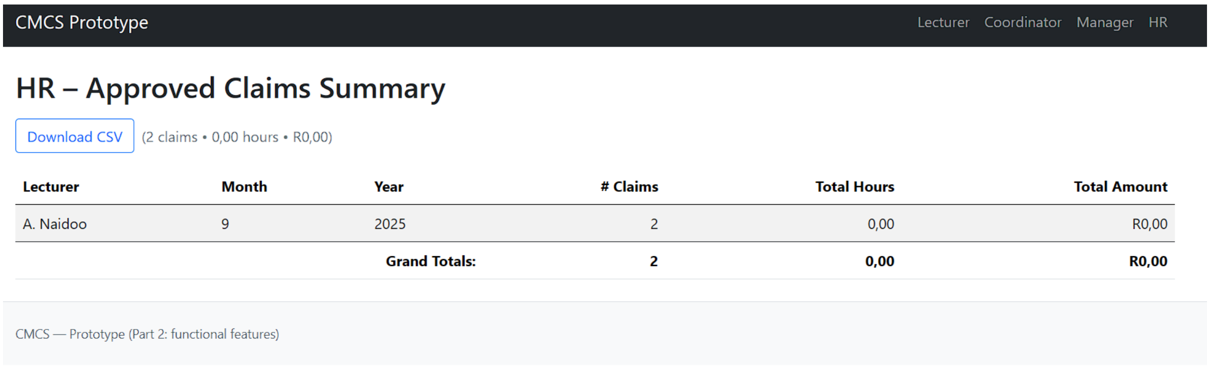
Hours: **181.00** • Rate: **$120.00** • Total: **$21,720.00**

CMCS — Prototype (Part 2: functional features)

- Pending claims dashboard with Approve/Reject buttons.

HR View Automation:

- Feature Implemented: HR can manage lecturer data and generate payment reports. ASP.NET Core Razor Pages enable user-friendly data management interfaces (Microsoft, 2025).
- Automation:
  - Auto-generated reports of all approved claims, grouped by lecturer and month.
  - Option to export invoice-style summaries.
  - Lecturer information can be updated from a single interface.
- Tools Used: ASP.NET Core Razor Pages for HR interface, Entity Framework for data access, and LINQ-to-Entities for generating reports (Microsoft, 2025).
- Outcome: Reduces manual work, ensures accuracy, and provides consistent payroll reports.



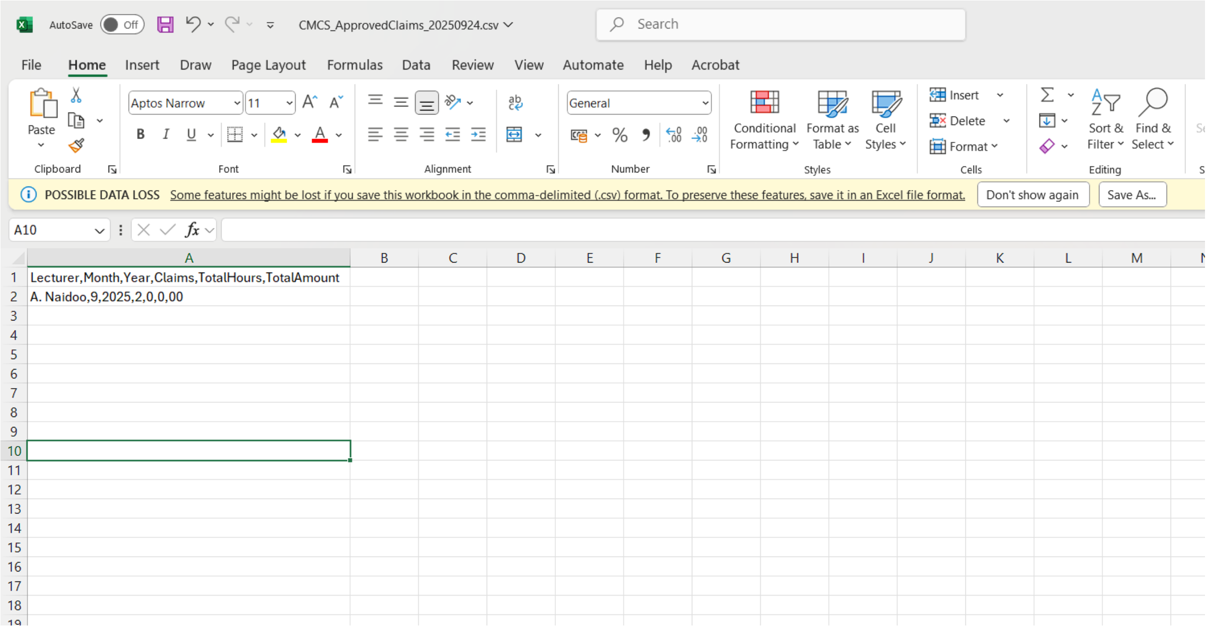
CMCS Prototype

HR – Approved Claims Summary

(2 claims • 0,00 hours • R0,00)

Lecturer	Month	Year	# Claims	Total Hours	Total Amount
A. Naidoo	9	2025	2	0,00	R0,00
		Grand Totals:	**2**	**0,00**	**R0,00**

CMCS — Prototype (Part 2: functional features)



CMCS_ApprovedClaims_20250924.csv

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

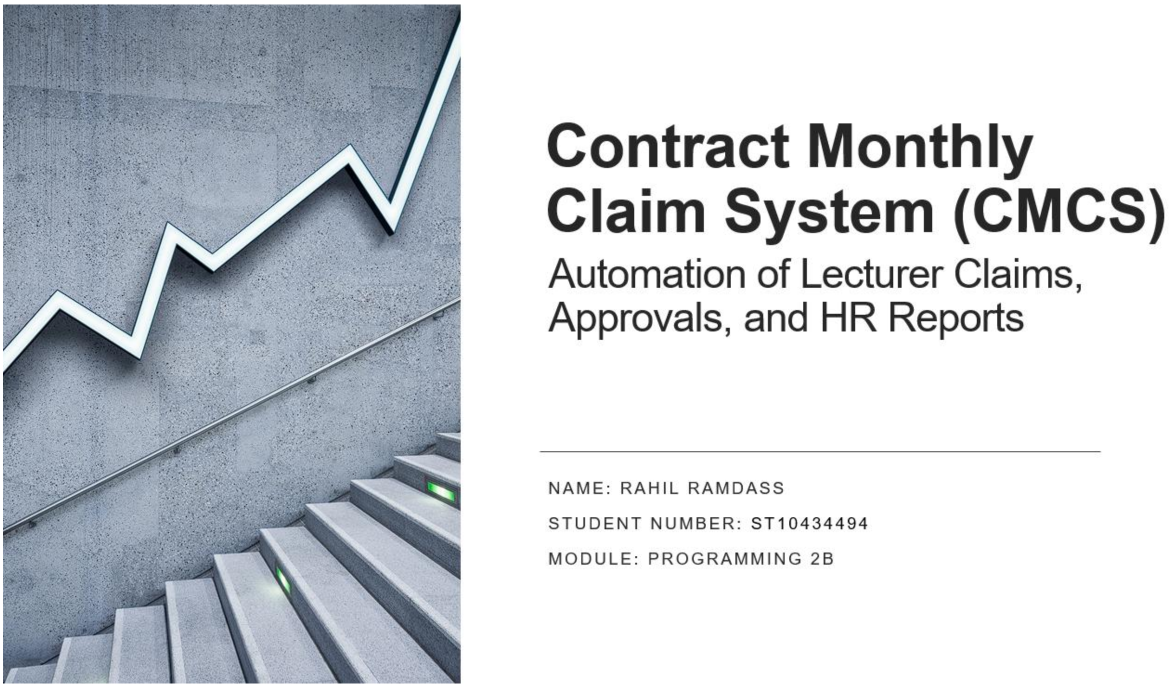
	A
1	Lecturer,Month,Year,Claims,TotalHours,TotalAmount
2	A. Naidoo,9,2025,2,0,0,00

- HR interface showing generated summary report.

2.

The presentation created for the Contract Monthly Claim System (CMCS) consists of 11 slides:

1. Cover page
2. System overview
3. Project objective
4. Lecturer View (Automation)
5. Coordinator/Manager View (Workflow)
6. HR View (Reports)
7. Benefits of Automation
8. Technical Stack
9. Version Control (GitHub)
10. Lecturer Feedback Implementation
11. Conclusion



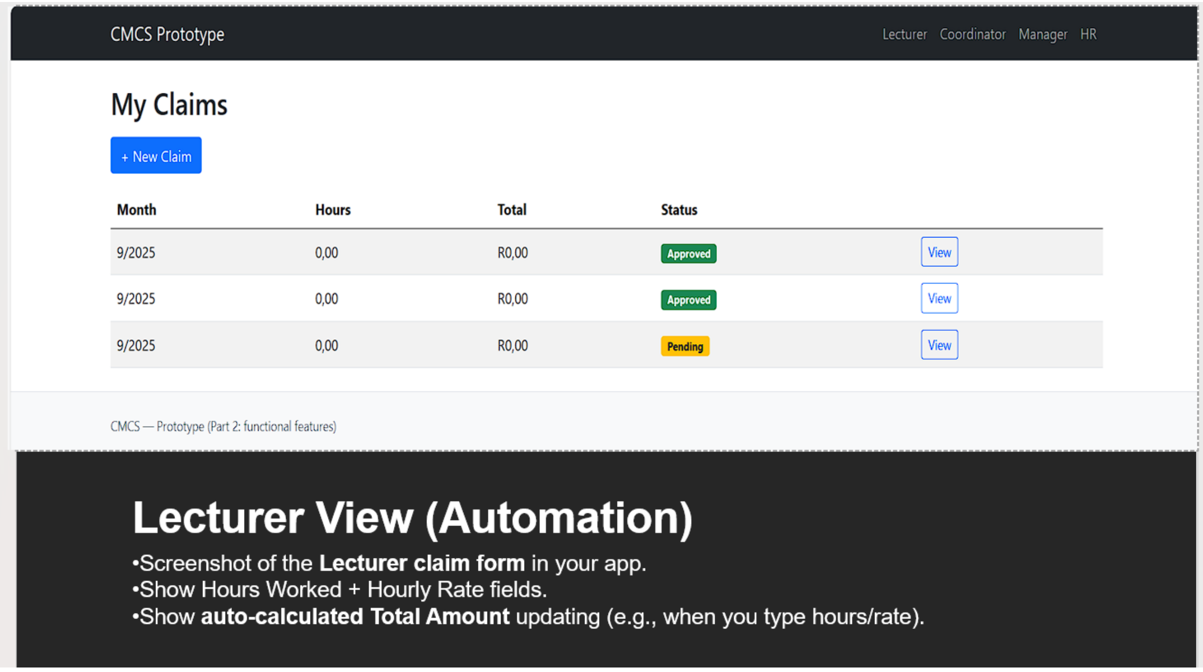
Contract Monthly Claim System (CMCS)

Automation of Lecturer Claims, Approvals, and HR Reports

NAME: RAHIL RAMDASS

STUDENT NUMBER: ST10434494

MODULE: PROGRAMMING 2B



My Claims

Month	Hours	Total	Status	
9/2025	0,00	R0,00	Approved	View
9/2025	0,00	R0,00	Approved	View
9/2025	0,00	R0,00	Pending	View

CMCS — Prototype (Part 2: functional features)

Lecturer View (Automation)

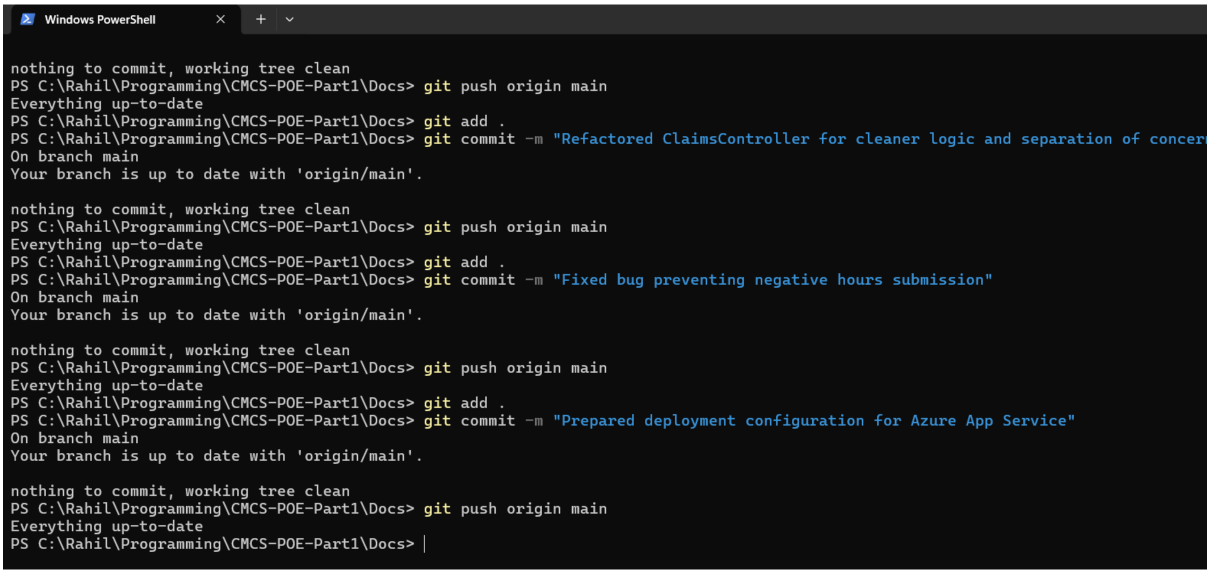
•Screenshot of the **Lecturer claim form** in your app.
•Show Hours Worked + Hourly Rate fields.
•Show **auto-calculated Total Amount** updating (e.g., when you type hours/rate).

- Slides from PowerPoint as evidence

3.

Version Control (GitHub)

- Commits: 10+ commits pushed to GitHub.
- Commit Messages: Clear, descriptive, e.g.:
    - *“Added auto-calculation feature for Lecturer claims”*
    - *“Implemented Coordinator approval workflow”*
    - *“HR report generation with LINQ queries”*
- Outcome: Demonstrates iterative development and good coding practices.



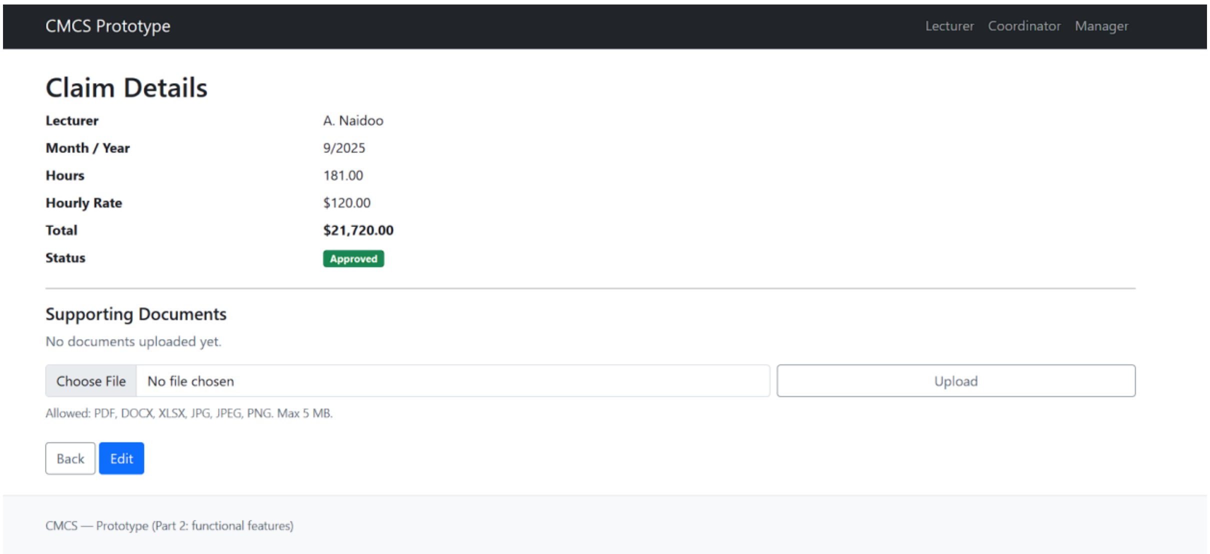
```
nothing to commit, working tree clean
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git push origin main
Everything up-to-date
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git add .
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git commit -m "Refactored ClaimsController for cleaner logic and separation of concer
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git push origin main
Everything up-to-date
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git add .
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git commit -m "Fixed bug preventing negative hours submission"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git push origin main
Everything up-to-date
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git add .
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git commit -m "Prepared deployment configuration for Azure App Service"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> git push origin main
Everything up-to-date
PS C:\Rahil\Programming\CMCS-POE-Part1\Docs> |
```

- GitHub commit history (show 10 commits).

4.

Lecturer Feedback Implementation:

- Feedback Collected: Lecturer requested easier claim input and clearer approval status visibility. This ensured user-centred design and addressed actual user pain points (Microsoft, 2025).
- Changes Made:
    - Added auto-calculation to reduce errors.
    - Improved claim status badges (Pending/Approved/Rejected with colour coding).
    - Enhanced upload feature for documents with restrictions on file size and type.



Claim Details

Lecturer	A. Naidoo
Month / Year	9/2025
Hours	181.00
Hourly Rate	$120.00
Total	**$21,720.00**
Status	Approved

Supporting Documents

No documents uploaded yet.

No file chosen

Allowed: PDF, DOCX, XLSX, JPG, JPEG, PNG. Max 5 MB.

CMCS — Prototype (Part 2: functional features)

- Lecturer claim form with changes applied.

Bibliography

Chacon, S. & Straub, B., 2014. *Pro Git.* 2nd ed. New York: Apress.

Microsoft, 2024. *Entity Framework Core.* [Online]
Available at: https://learn.microsoft.com/en-us/ef/core/
[Accessed 24 September 2025].

Microsoft, 2024. *Overview of ASP.NET Core MVC.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-9.0
[Accessed 24 September 2025].

Microsoft, 2025. *Introduction to Identity on ASP.NET Core.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-9.0&tabs=visual-studio
[Accessed 24 September 2025].

Microsoft, 2025. *Razor Pages architecture and concepts in ASP.NET Core.* [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-9.0&tabs=visual-studio
[Accessed 24 September 2025].

W3Schools, 2025. *jQuery Tutorial.* [Online]
Available at: https://www.w3schools.com/jquery/
[Accessed 24 September 2025].