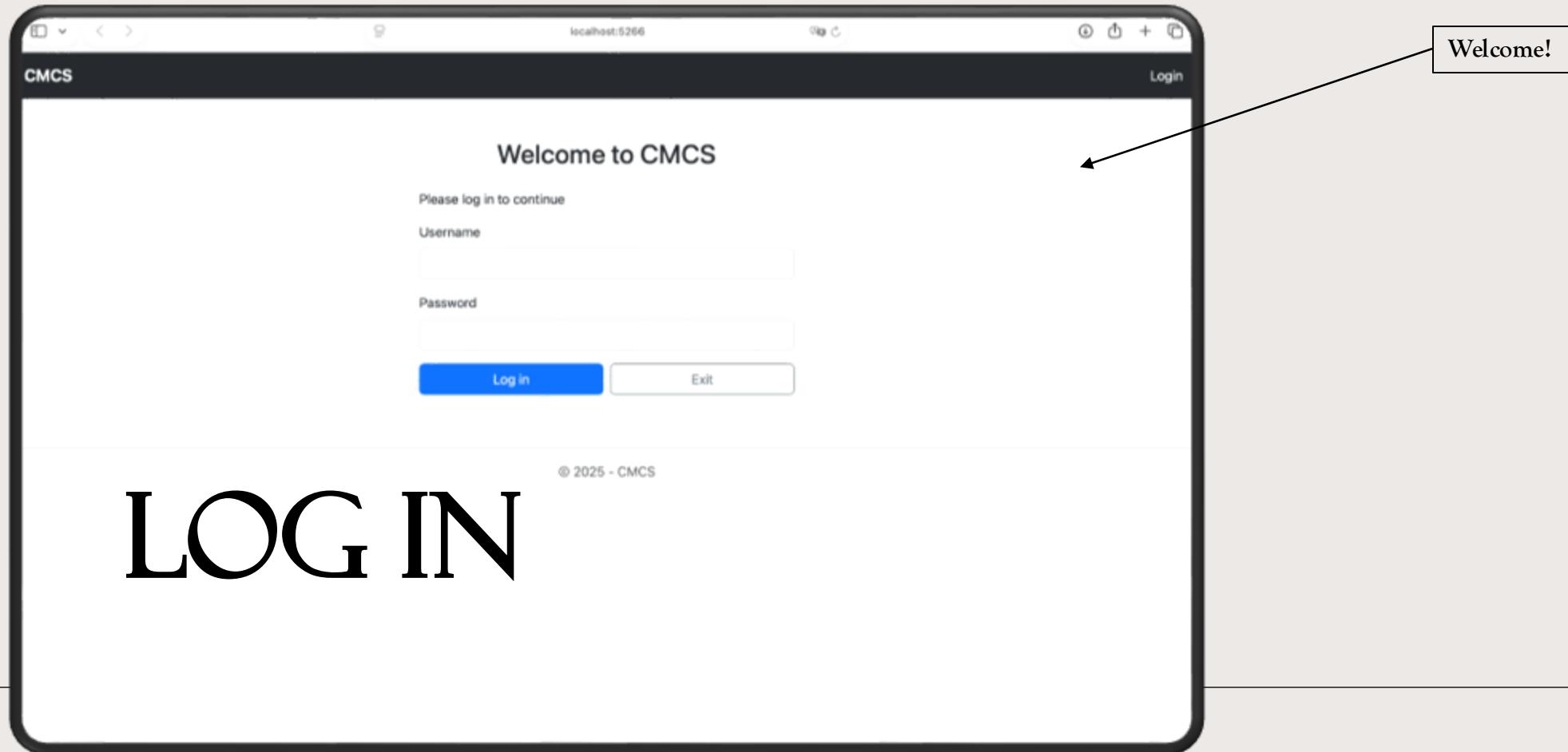




CONTRACTOR MONTHLY CLAIM SYSTEM

Programming 2B - PROG6212 – Portfolio of Evidence
ST10082027 – John Hoffman Prinsloo



LOG IN

LOG IN: CODE

```
public class AccountController : Controller
{
    public async Task<ActionResult> Login(LoginViewModel model, string? returnUrl = null)
    {
        if (!ModelState.IsValid)
        {
            return View("~/Views/GenericViews/Login.cshtml", model);
        }

        var user = await _db.Users
            .FirstOrDefaultAsync(u =>
                u.UserName == model.Username || u.Email == model.Username);

        if (user == null || !PasswordHelper.VerifyPassword(model.Password, user.PasswordHash))
        {
            ModelState.AddModelError(string.Empty, "Invalid username or password.");
            return View("~/Views/GenericViews/Login.cshtml", model);
        }

        HttpContext.Session.SetString("UserId", user.Id);
        HttpContext.Session.SetString("UserEmail", user.Email);
        HttpContext.Session.SetString("UserRole", user.Role);
        HttpContext.Session.SetString("UserFullName", $"{user.FirstName} {user.LastName}");

        _logger.LogInformation("User {UserName} logged in with role {Role}", user.UserName, user.Role);

        if (!string.IsNullOrEmpty(returnUrl) && Url.IsLocalUrl(returnUrl))
        {
            return Redirect(returnUrl);
        }

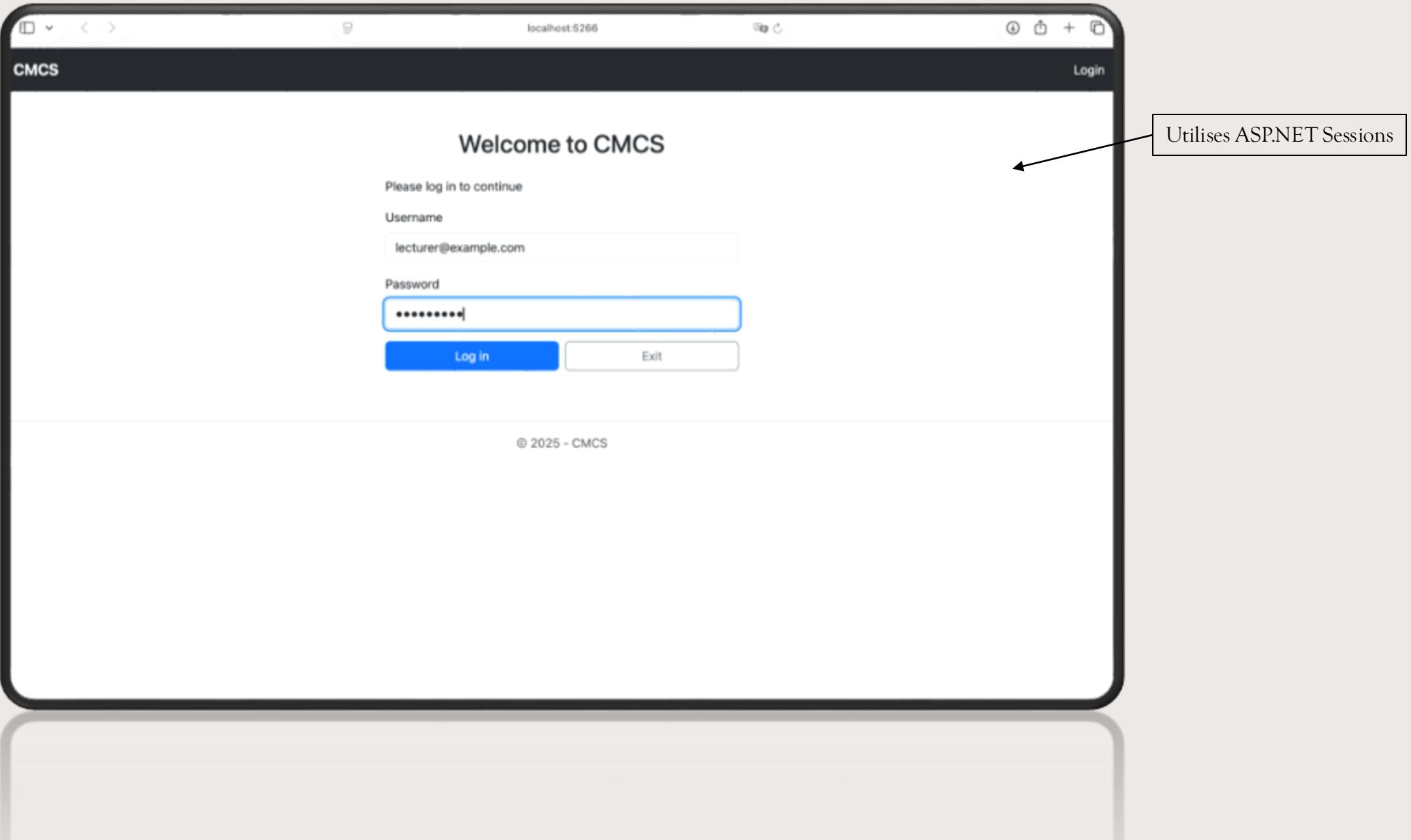
        return user.Role switch
        {
            "Lecturer" => RedirectToAction("Index", "LecturerDashboard"),
            "Coordinator" => RedirectToAction("Index", "ProgrammeCoordinatorDashboard"),
            "AcademicManager" => RedirectToAction("Index", "AcademicManagerDashboard"),
            "HR" => RedirectToAction("Index", "HRDashboard"),
            _ => RedirectToAction("Index", "Home")
        };
    }
}
```

Secure, session-driven authentication with intelligent role-specific dashboard redirection

LECTURER

- ✓ Submit monthly claims with documentation.
- ✓ Enter hours and review calculations.
- ✓ Track claim status through workflow.
- ✓ View history of submitted claims.

LECTURER: LOG IN



LECTURER: DASHBOARD

The screenshot shows a web browser window for the CMCS Lecturer Dashboard at localhost:5266/LecturerDashboard/index. The dashboard is personalized for 'Lerato Lecturer'.

Welcome, Lerato Lecturer

Role: Lecturer

Create a new monthly claim

Capture your hours for the current month and submit them for verification and approval.

New Monthly Claim

View my claims

Track the status of your claims: draft, pending verification, approved or rejected.

My Claims

How the claim process works

Submit your claim with accurate hours and documentation. Your Programme Coordinator will verify your claim, then your Academic Manager will approve it for payment. You can track your claim status at any time from the "My Claims" page.

- Draft your claim and save it before submission.
- Submit when you're satisfied with your hours and attachments.
- Track statuses: Draft → Pending → Verified → Approved / Rejected.

Open "<https://localhost:5266/Claims/New>" in a new tab

LECTURER: CREATE CLAIM

localhost:5266/Claims/New

Create Claim

Work calendar

Month	Year
Aug	2024
Sep	2025
Oct	2026
Nov	

Scroll the wheels to choose your month and year.
The calendar below updates automatically.

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1 No hours	2 No hours	3 8.00 h	4 8.00 h	5 8.00 h	6 8.00 h	7 4.00 h
Click to add hours	09:00 – 17:00	09:00 – 17:00	09:00 – 17:00	09:00 – 17:00	09:00 – 13:00	Click to add hours
8 No hours	9 No hours	10 8.00 h	11 7.63 h	12 7.97 h	13 8.00 h	14 4.00 h
Click to add hours	09:00 – 17:00	09:00 – 17:00	09:00 – 17:00	09:00 – 17:00	09:00 – 13:00	Click to add hours
15 No hours	16 No hours	17 11.00 h	18 8.09 h	19 8.00 h	20 8.00 h	21 4.00 h
Click to add hours	09:00 – 20:00	09:00 – 17:00	09:00 – 17:00	09:00 – 17:00	09:00 – 13:00	Click to add hours
22 No hours	23 No hours	24 8.00 h	25 8.00 h	26 8.00 h	27 8.00 h	28 4.00 h
Click to add hours	09:00 – 17:00	09:00 – 17:00	09:00 – 17:00	09:00 – 17:00	09:00 – 13:00	Click to add hours
31 No hours						

Work Summary

Date	Start	End	Hours	
Mon 3 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Tue 4 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Wed 5 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Thu 6 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Fri 7 Nov	09:00	13:00	4.00	<button>Edit</button> <button>Delete</button>
Mon 10 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Tue 11 Nov	09:22	17:00	7.63	<button>Edit</button> <button>Delete</button>
Wed 12 Nov	09:02	17:00	7.97	<button>Edit</button> <button>Delete</button>
Thu 13 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Fri 14 Nov	09:00	13:00	4.00	<button>Edit</button> <button>Delete</button>
Mon 17 Nov	09:00	20:00	11.00	<button>Edit</button> <button>Delete</button>
Tue 18 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Wed 19 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Thu 20 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Fri 21 Nov	09:00	13:00	4.00	<button>Edit</button> <button>Delete</button>
Mon 24 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Tue 25 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>
Wed 26 Nov	09:00	17:00	8.00	<button>Edit</button> <button>Delete</button>

Log hours worked

180 work hour monthly limits

LECTURER: EDIT CLAIM

localhost:5266/Claims/New

Edit Day

Date: Mon 17 Nov

Start time: 09:00 End time: 20:00

Hours are calculated automatically from your start and end times.

Cancel Save

No hours Click to add hours

3. 8.00 h 09:00 – 17:00

4. 8.00 h 09:00 – 17:00

5. 8.00 h 09:00 – 17:00

1. No hours Click to add hours

10. 8.00 h 09:00 – 17:00

11. 7.83 h 09:22 – 17:00

12. 7.87 h 09:02 – 17:00

13. No hours Click to add hours

17. 11.00 h 09:00 – 20:00

18. 8.00 h 09:00 – 17:00

19. 8.00 h 09:00 – 17:00

2. No hours Click to add hours

24. 8.00 h 09:00 – 17:00

25. 8.00 h 09:00 – 17:00

26. 8.00 h 09:00 – 17:00

27. 8.00 h 09:00 – 17:00

28. 4.00 h 09:00 – 13:00

2. No hours Click to add hours

Tue 11 Nov

09:22 17:00 7.63 Edit Delete

09:02 17:00 7.97 Edit Delete

09:00 17:00 8.00 Edit Delete

09:00 13:00 4.00 Edit Delete

09:00 20:00 11.00 Edit Delete

09:00 17:00 8.00 Edit Delete

09:00 13:00 4.00 Edit Delete

Mon 24 Nov

Tue 25 Nov

Wed 26 Nov

Thu 27 Nov

Fri 28 Nov

Total Hours 146.60

Get Quote from Server

Next

© 2025 - CMCS

Effortlessly edit hours

LECTURER: QUOTE

localhost:5266/Claims/New

Click a day to add or update your start/end time for that day. Click again to edit; use the delete button in the table to remove.

	Thu 20 Nov	09:00	17:00	8.00	Edit	Delete
24	09:00 - 17:00	8.00 h				
25	09:00 - 17:00	8.00 h				
26	09:00 - 17:00	8.00 h				
27	09:00 - 17:00	8.00 h				
28	09:00 - 13:00	4.00 h				

Total Hours 146.60

Server Quote

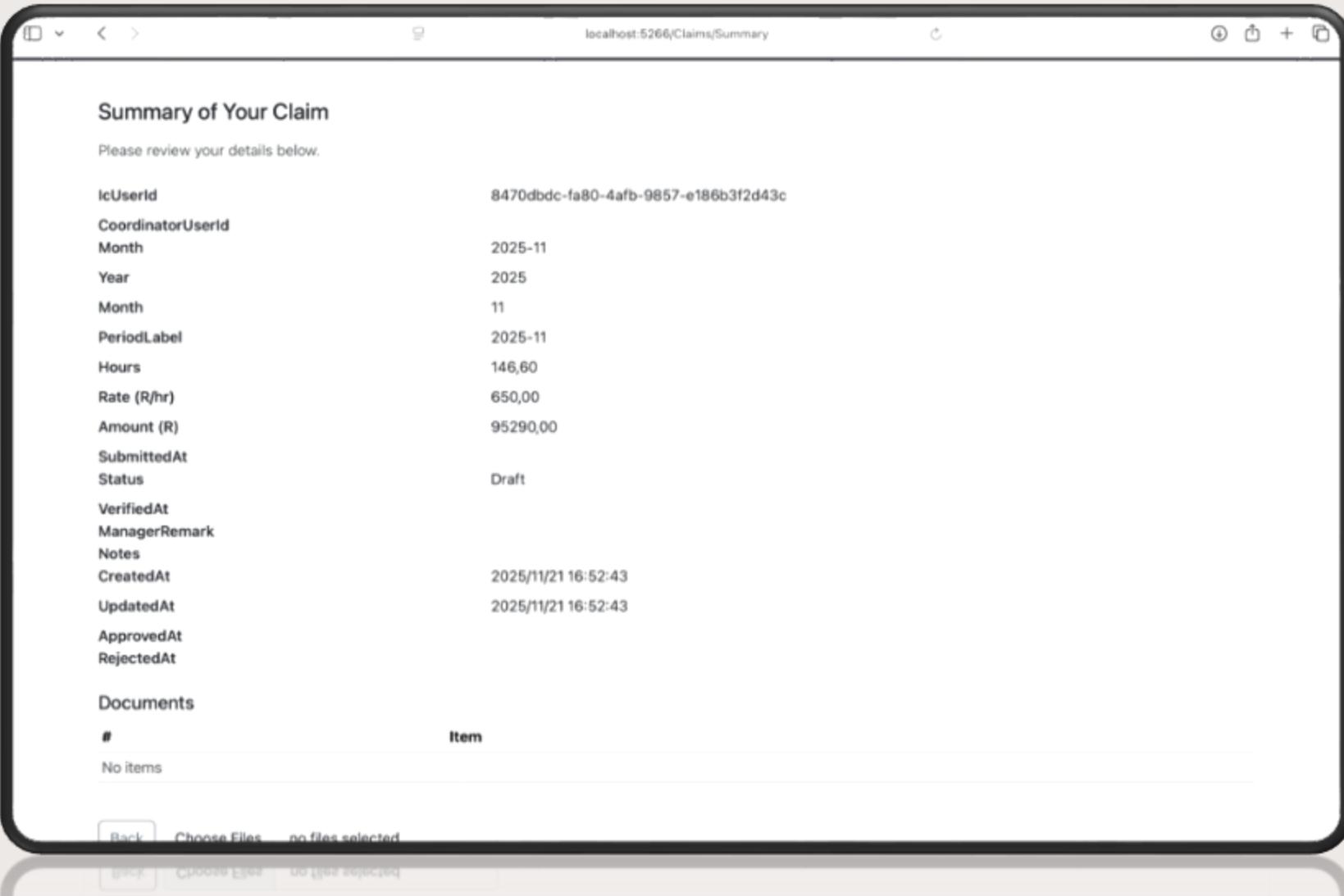
Server quote:
Month: 2025-11
Hours: 146.6
Rate: R 650
Amount: R 95290

Get Quote from Server

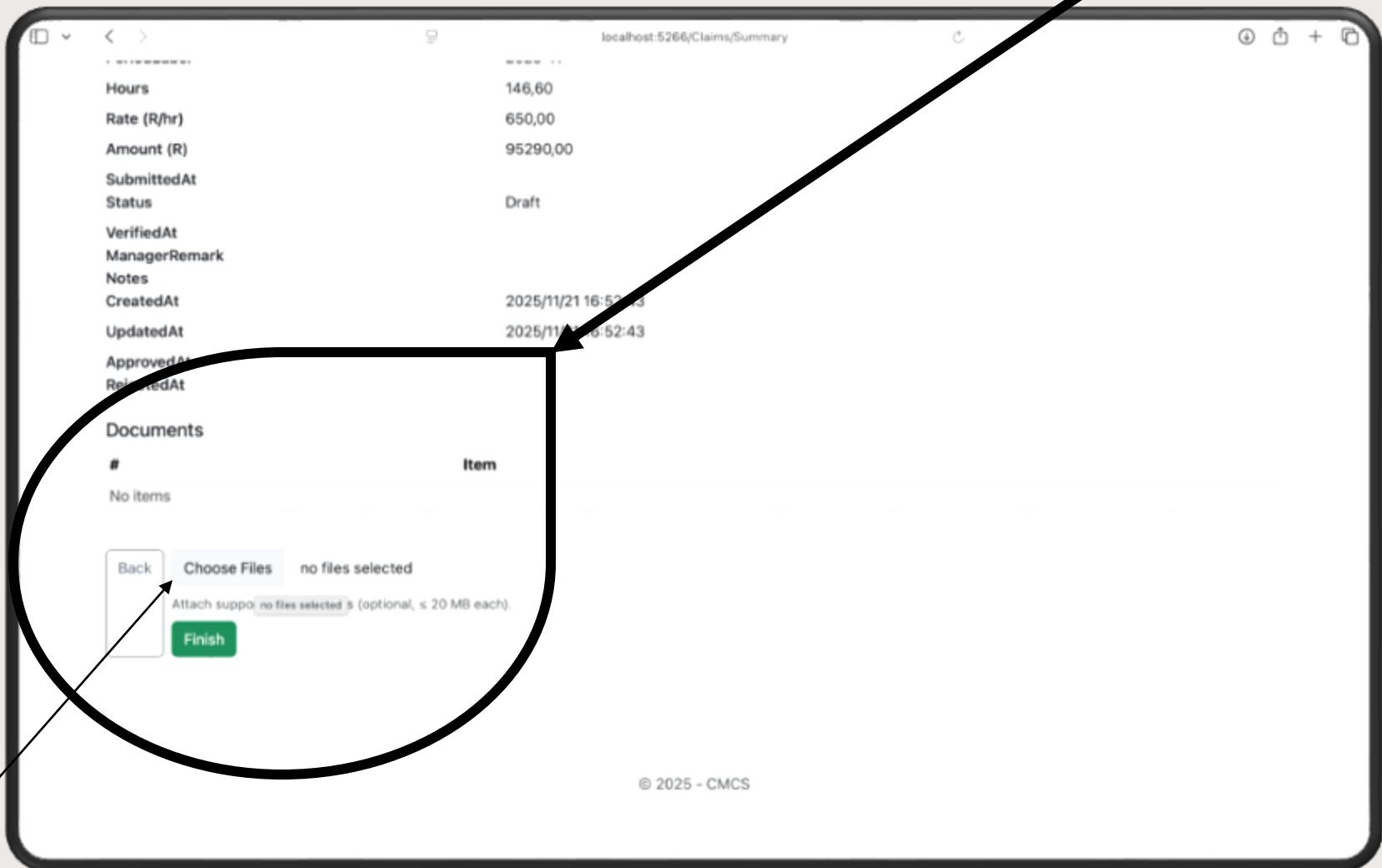
Auto-calculation using hours × hourly rate

Hourly rate is auto-pulled from Human Resources data

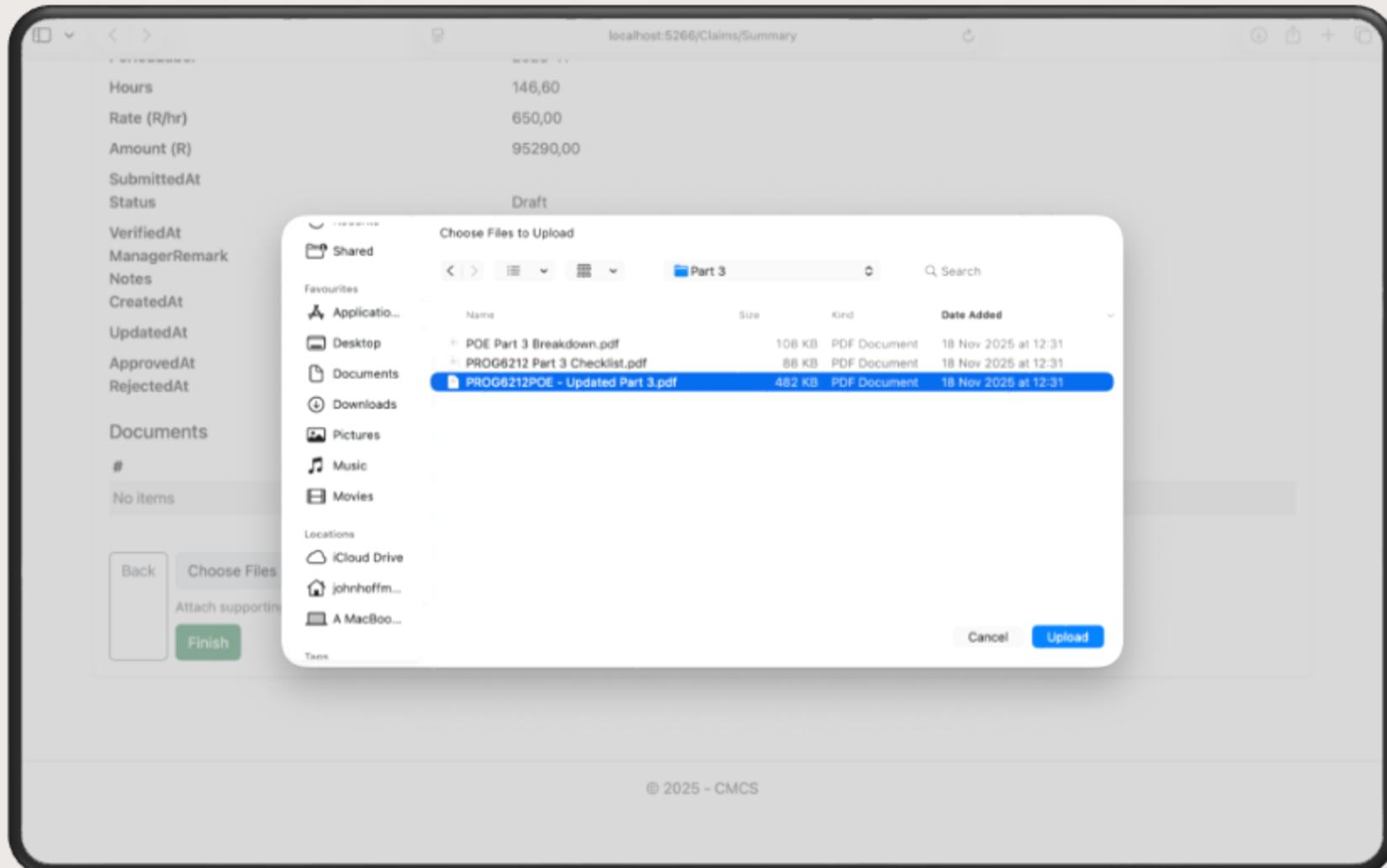
LECTURER: SUMMARY OF CLAIM



LECTURER: ADD SUPPORTING DOCUMENTS



LECTURER: UPLOAD SUPPORTING DOCUMENTS



LECTURER: CLAIMS

The screenshot shows a web browser window with the URL `localhost:5266/Claims/My`. The page title is "CMCS" and the sub-page title is "Lecturer Dashboard". The main content area is titled "My Claims" and displays a table of two rows of claim data. A callout box with the text "Instant claim status updates through all stages" points to the first row of the table.

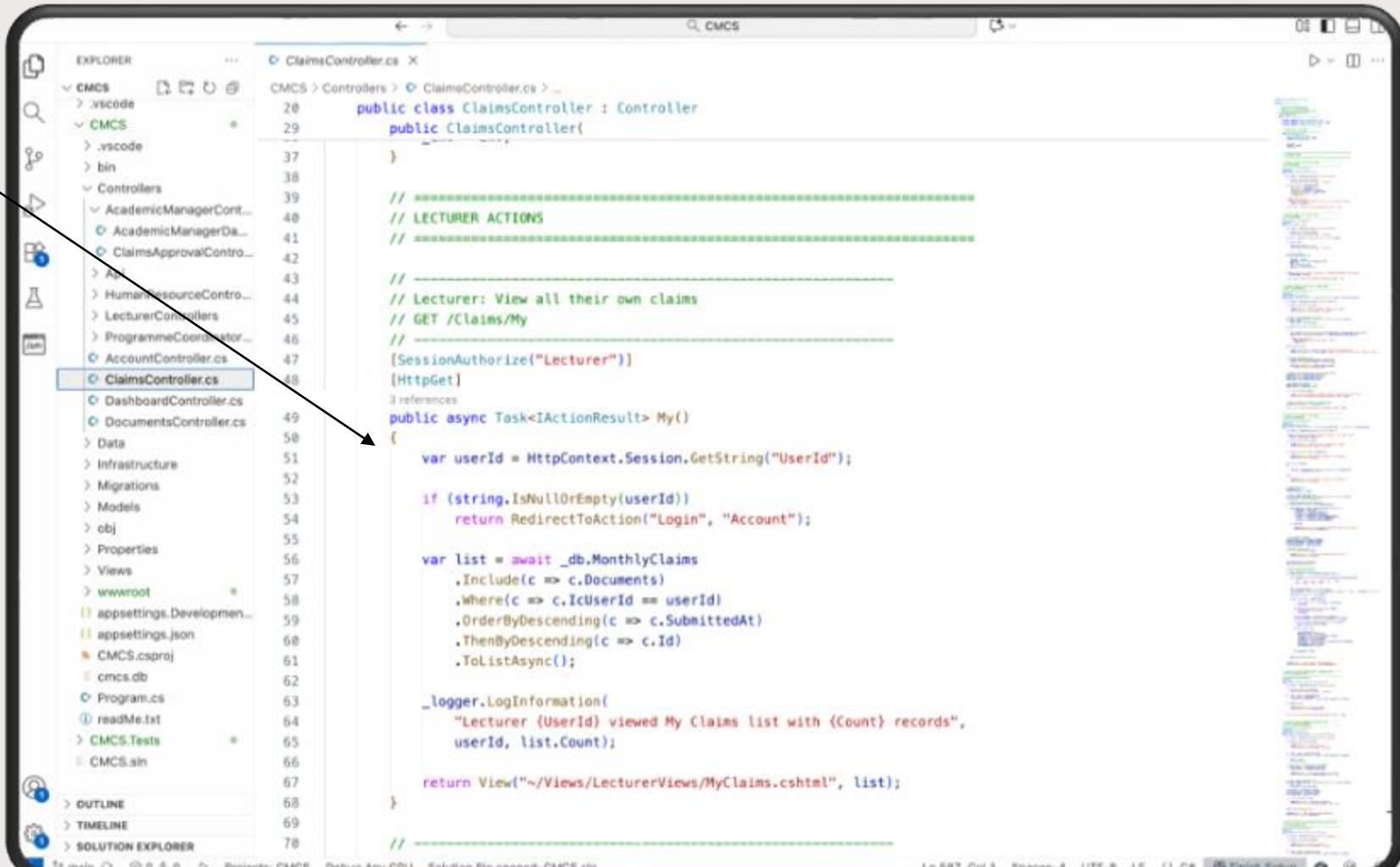
Month	Status	Total Hours	Amount	Submitted	Updated	Actions
2025-02	Pending	4	R 2 600,00	2025-11-21	2025-11-21	No actions available
2025-11	ApprovedByManager	65,1	R 42 315,00	2025-11-21	2025-11-21	No actions available

© 2025 - CMCS

Instant claim status updates
through all stages

LECTURER: MY CLAIMS CODE

Loads and displays lecturer specific claims



```
public class ClaimsController : Controller
{
    public ClaimsController()
    {
    }

    // -----
    // LECTURER ACTIONS
    //

    // -----
    // Lecturer: View all their own claims
    // GET /Claims/My
    //
    [SessionAuthorize("Lecturer")]
    [HttpGet]
    public async Task<IActionResult> My()
    {
        var userId = HttpContext.Session.GetString("UserId");

        if (string.IsNullOrEmpty(userId))
            return RedirectToAction("Login", "Account");

        var list = await _db.MonthlyClaims
            .Include(c => c.Documents)
            .Where(c => c.IUserId == userId)
            .OrderByDescending(c => c.SubmittedAt)
            .ThenByDescending(c => c.Id)
            .ToListAsync();

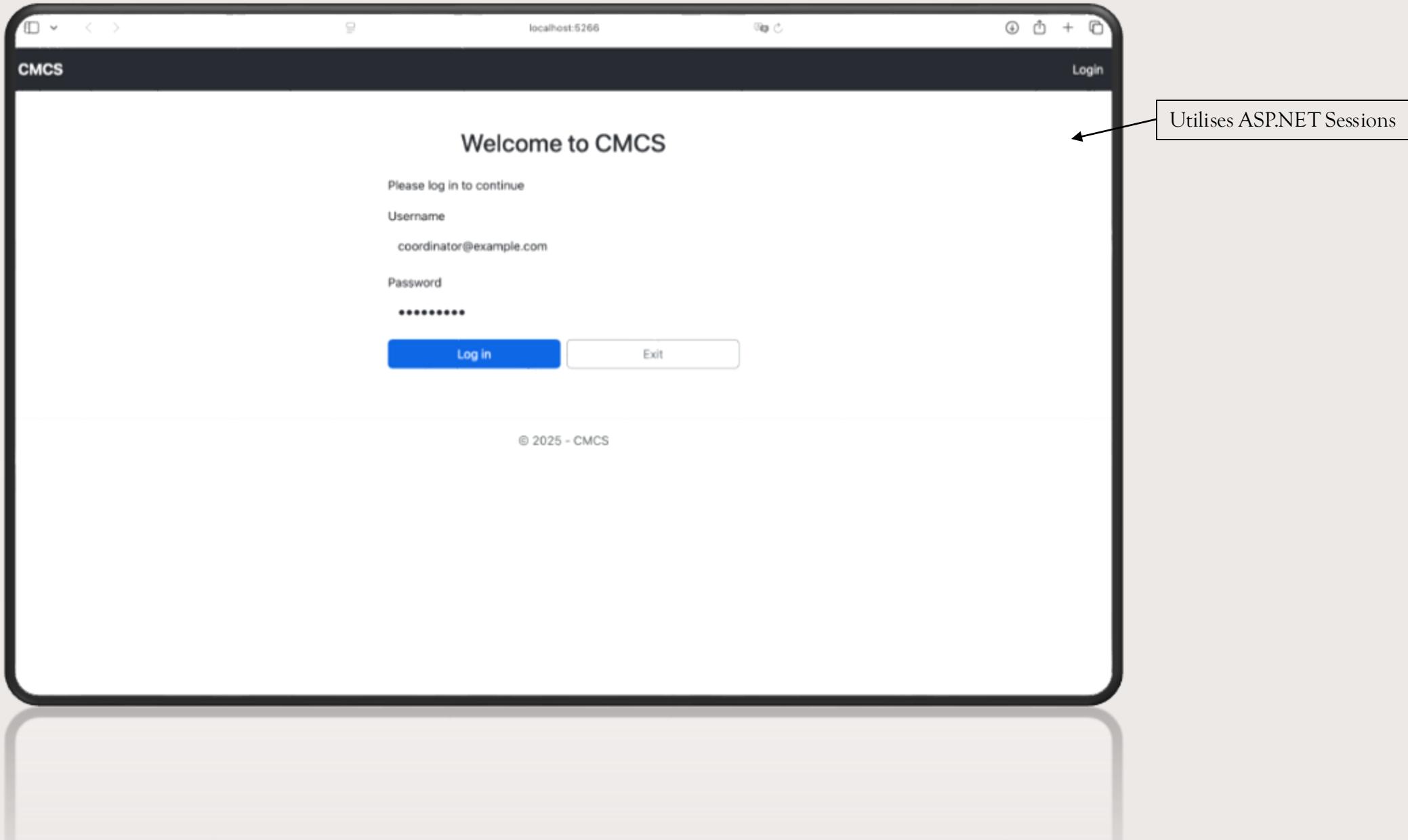
        _logger.LogInformation(
            "Lecturer ({UserId}) viewed My Claims list with {Count} records",
            userId, list.Count);

        return View("~/Views/LecturerViews/MyClaims.cshtml", list);
    }
}
```

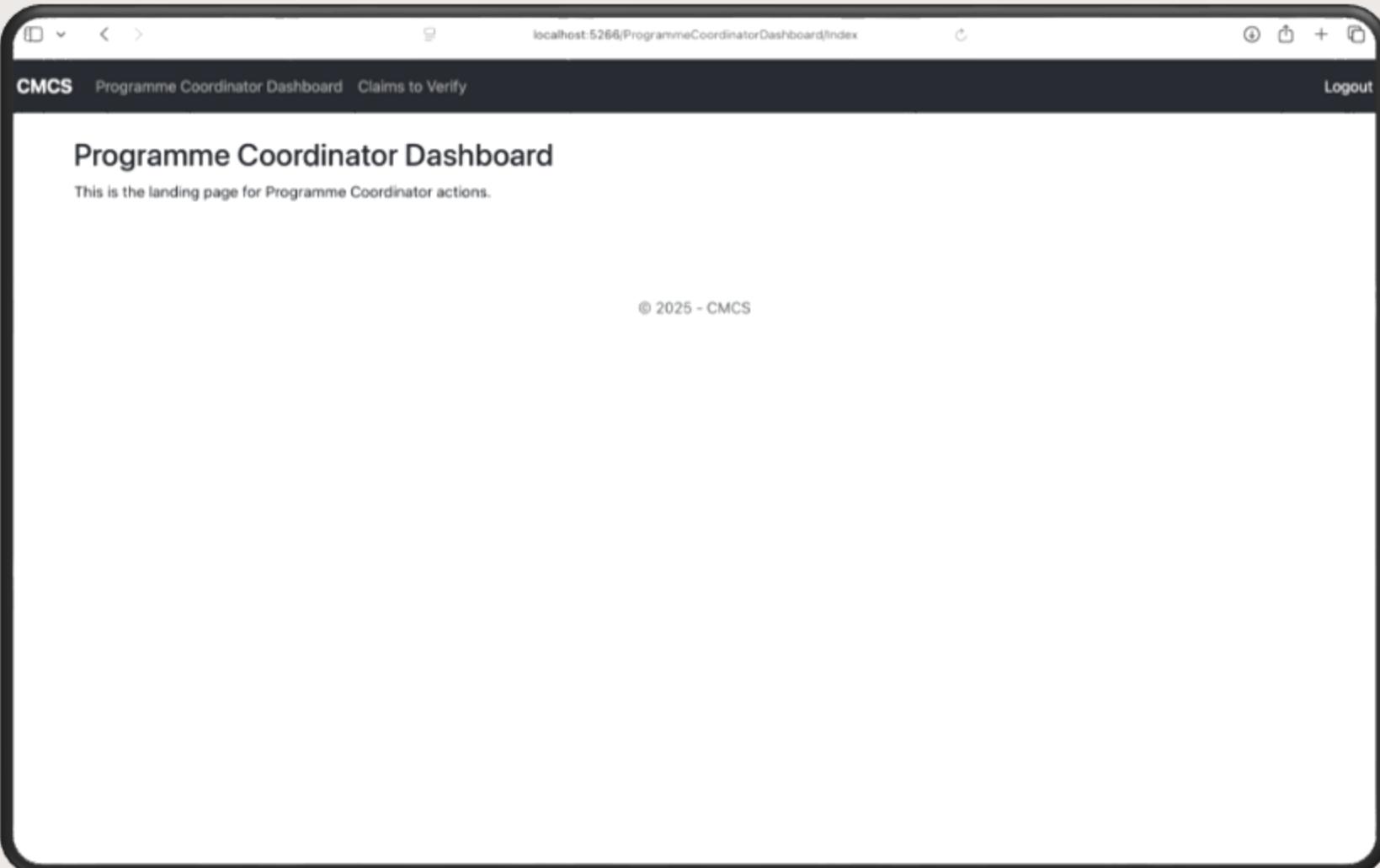
PROGRAMME COORDINATOR

- ✓ Review lecturer claims for accuracy.
- ✓ Verify submitted hours and documents.
- ✓ Forward valid claims for approval.

PROGRAMME COORDINATOR: LOG IN



PROGRAMME COORDINATOR: DASHBOARD



PROGRAMME COORDINATOR: CLAIMS TO VERIFY

The screenshot shows a web browser window titled "CMCS Programme Coordinator Dashboard Claims to Verify" with the URL "localhost:5266/ClaimsReview/Queue". The page displays a table of claims with three entries. Each entry includes the Lecturer (Lerato Lecturer), Month (2025-01, 2025-10, 2025-02), Hours (46, 46, 4), Amount (R29 900,00, R29 900,00, R2 600,00), and Submitted date (2025-11-21 19:03, 2025-11-21 19:01, 2025-11-21 16:39). To the right of each row is a blue "Review" button. A callout box with the text "View all lecturer submitted claims awaiting verification" points to the first row of the table.

Lecturer	Month	Hours	Amount	Submitted	Action
Lerato Lecturer	2025-01	46	R29 900,00	2025-11-21 19:03	<button>Review</button>
Lerato Lecturer	2025-10	46	R29 900,00	2025-11-21 19:01	<button>Review</button>
Lerato Lecturer	2025-02	4	R2 600,00	2025-11-21 16:39	<button>Review</button>

View all lecturer submitted claims awaiting verification

PROGRAMME COORDINATOR: VERIFY / REJECT CLAIM

localhost:5288/ClaimsReview/Review/4

PROGRAMME COORDINATOR
Review claim for Lerato Lecturer
Month: 2025-01 - Status: Pending

LECTURER
Lerato Lecturer
lecturer@example.com

PERIOD
2025-01

HOURS
46,0 hrs

TOTAL CAPTURED HOURS
46,0 hrs

Submitted: 21 Nov 2025 19:03

WORK CALENDAR
Days & hours (read-only)

This view is for verification only. You cannot edit the lecturer's hours here.

SUN	MON	TUE	WED	THU	FRI	SAT
5 No hours	6 No hours	7 No hours	8 No hours	9 No hours	10 No hours	11 No hours
No hours Sun 5 Jan	No hours Mon 6 Jan	No hours Tue 7 Jan	No hours Wed 8 Jan	No hours Thu 9 Jan	No hours Fri 10 Jan	No hours Sat 11 Jan
12 No hours	13 No hours	14 No hours	15 No hours	16 No hours	17 No hours	18 No hours
No hours Sun 12 Jan	No hours Mon 13 Jan	No hours Tue 14 Jan	No hours Wed 15 Jan	No hours Thu 16 Jan	No hours Fri 17 Jan	No hours Sat 18 Jan
19 No hours	20 No hours	21 No hours	22 No hours	23 No hours	24 No hours	25 No hours
No hours Sun 19 Jan	No hours Mon 20 Jan	No hours Tue 21 Jan	No hours Wed 22 Jan	No hours Thu 23 Jan	No hours Fri 24 Jan	No hours Sat 25 Jan
26 No hours	27 No hours	28 No hours	29 No hours	30 No hours	31 No hours	
No hours Sun 26 Jan	No hours Mon 27 Jan	No hours Tue 28 Jan	No hours Wed 29 Jan	No hours Thu 30 Jan	No hours Fri 31 Jan	

COORDINATOR DECISION
Add an optional remark for the Academic Manager or a reason if you reject.

Remark to Academic Manager (optional)
E.g. 'Hours match timetable. Ready for approval.'

Verify & send to Academic Manager

Reason for rejection
E.g. 'Hours exceed agreed maximum for this month.'

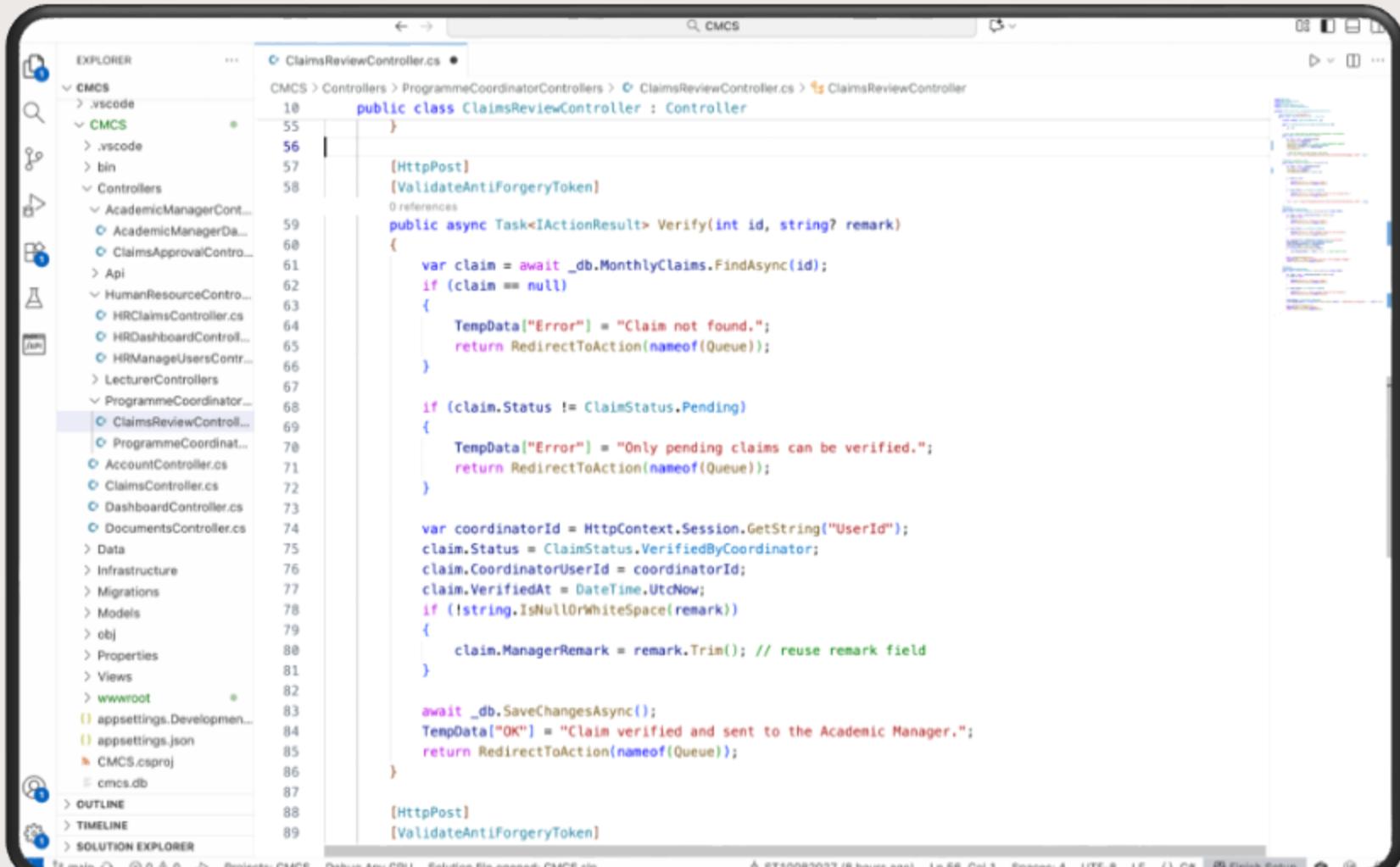
Reject claim

Note: This view intentionally hides the lecturer's hourly rate and total payable amount. Coordinators verify hours and documentation only.

Updates claim status

Approve or reject lecturer claims based on accuracy and completeness

PROGRAMME COORDINATOR: VERIFY CLAIMS CODE



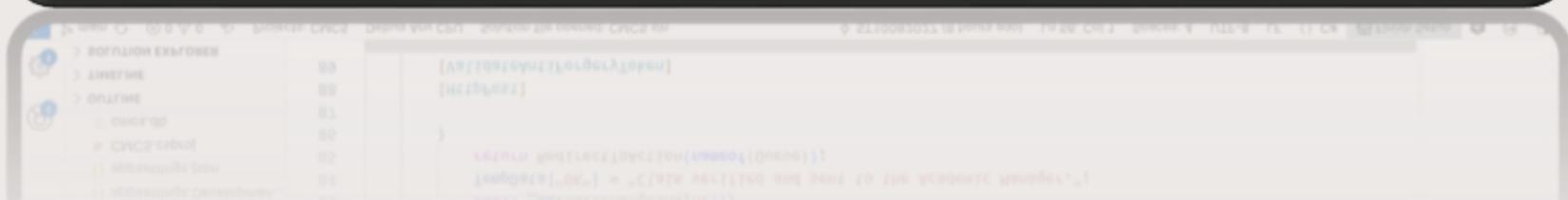
```
public class ClaimsReviewController : Controller
{
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Verify(int id, string? remark)
    {
        var claim = await _db.MonthlyClaims.FindAsync(id);
        if (claim == null)
        {
            TempData["Error"] = "Claim not found.";
            return RedirectToAction(nameof(Queue));
        }

        if (claim.Status != ClaimStatus.Pending)
        {
            TempData["Error"] = "Only pending claims can be verified.";
            return RedirectToAction(nameof(Queue));
        }

        var coordinatorId = HttpContext.Session.GetString("UserId");
        claim.Status = ClaimStatus.VerifiedByCoordinator;
        claim.CoordinatorUserId = coordinatorId;
        claim.VerifiedAt = DateTime.UtcNow;
        if (!string.IsNullOrWhiteSpace(remark))
        {
            claim.ManagerRemark = remark.Trim(); // reuse remark field
        }

        await _db.SaveChangesAsync();
        TempData["OK"] = "Claim verified and sent to the Academic Manager.";
        return RedirectToAction(nameof(Queue));
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
```



```
public class ClaimsReviewController : Controller
{
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Verify(int id, string? remark)
    {
        var claim = await _db.MonthlyClaims.FindAsync(id);
        if (claim == null)
        {
            TempData["Error"] = "Claim not found.";
            return RedirectToAction(nameof(Queue));
        }

        if (claim.Status != ClaimStatus.Pending)
        {
            TempData["Error"] = "Only pending claims can be verified.";
            return RedirectToAction(nameof(Queue));
        }

        var coordinatorId = HttpContext.Session.GetString("UserId");
        claim.Status = ClaimStatus.VerifiedByCoordinator;
        claim.CoordinatorUserId = coordinatorId;
        claim.VerifiedAt = DateTime.UtcNow;
        if (!string.IsNullOrWhiteSpace(remark))
        {
            claim.ManagerRemark = remark.Trim(); // reuse remark field
        }

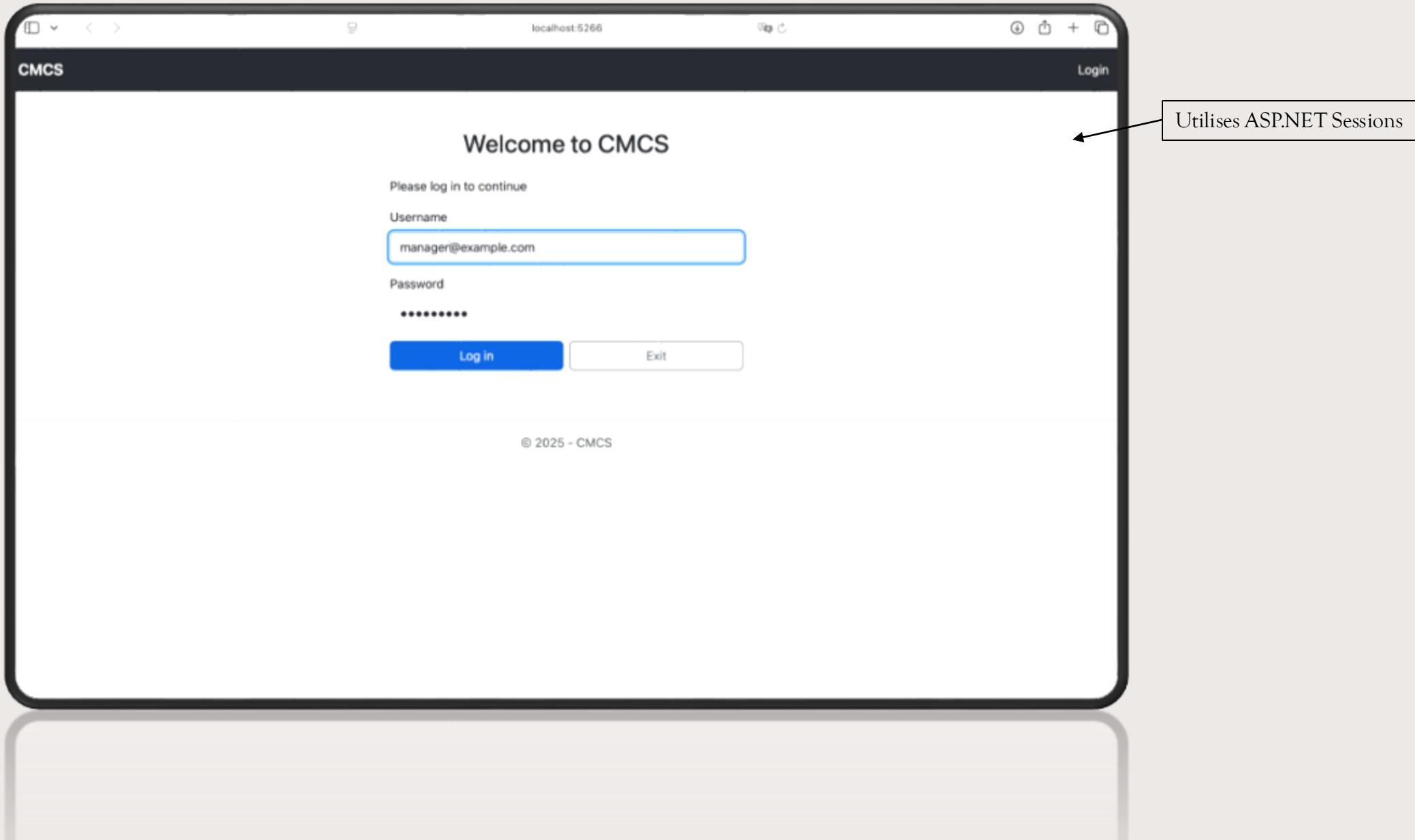
        await _db.SaveChangesAsync();
        TempData["OK"] = "Claim verified and sent to the Academic Manager.";
        return RedirectToAction(nameof(Queue));
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
```

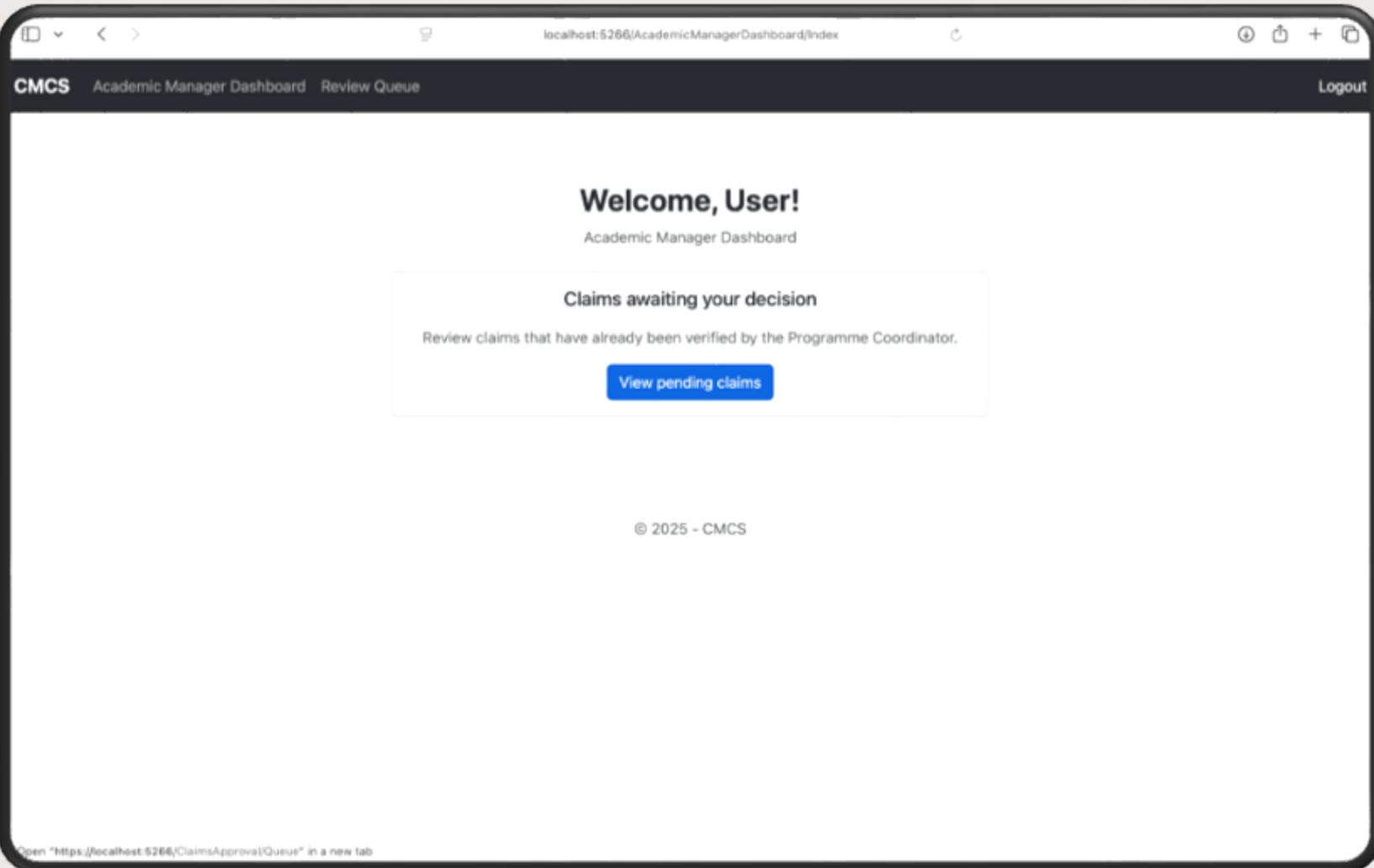
ACADEMIC MANAGER

-
- ✓ Approve coordinator-verified monthly claims.
 - ✓ Reject invalid claims with justification.
 - ✓ Ensure workflow integrity and compliance.
 - ✓ Authorize claims for HR processing.

ACADEMIC MANAGER: LOG IN



ACADEMIC MANAGER: DASHBOARD



ACADEMIC MANAGER: PENDING CLAIMS

The screenshot shows a web browser window titled "CMCS Academic Manager Dashboard Review Queue" at the URL "localhost:5266/ClaimsApproval/Queue". The main content area is titled "Pending Claims" and displays a single claim from "lecturer@example.com" for the month of "2025-01". The claim details are: Hours: 46, Rate: 650,00, Amount: 29900,00. There is one document attached. The claim was submitted on "Fri, 21 Nov 2025 19:03". Below the table, it says "Showing 1 pending claim." To the right of the table, there are "Actions" buttons: "Approve" (green) and "Reject" (red). A status message at the top right says "1 claim awaiting approval" with a link to "Click Approve to finalise, or Reject with a short reason." A callout box with an arrow points to the "Actions" buttons with the text "Approve or reject claims". Another callout box with an arrow points to the "Approve" and "Reject" buttons with the text "Actions trigger status updates visible".

localhost:5266/ClaimsApproval/Queue

CMCS Academic Manager Dashboard Review Queue

Logout

Pending Claims

Review and approve independent contractor monthly claims.

1 claim awaiting approval

Pending claims

Click Approve to finalise, or Reject with a short reason.

Lecturer	Month	Hours	Rate	Amount	Docs	Submitted	Actions
lecturer@example.com	2025-01	46	650,00	29900,00	1 file	Fri, 21 Nov 2025 19:03	<button>Approve</button> <button>Reason...</button> <button>Reject</button>

Showing 1 pending claim.

© 2025 - CMCS

Approve or reject claims

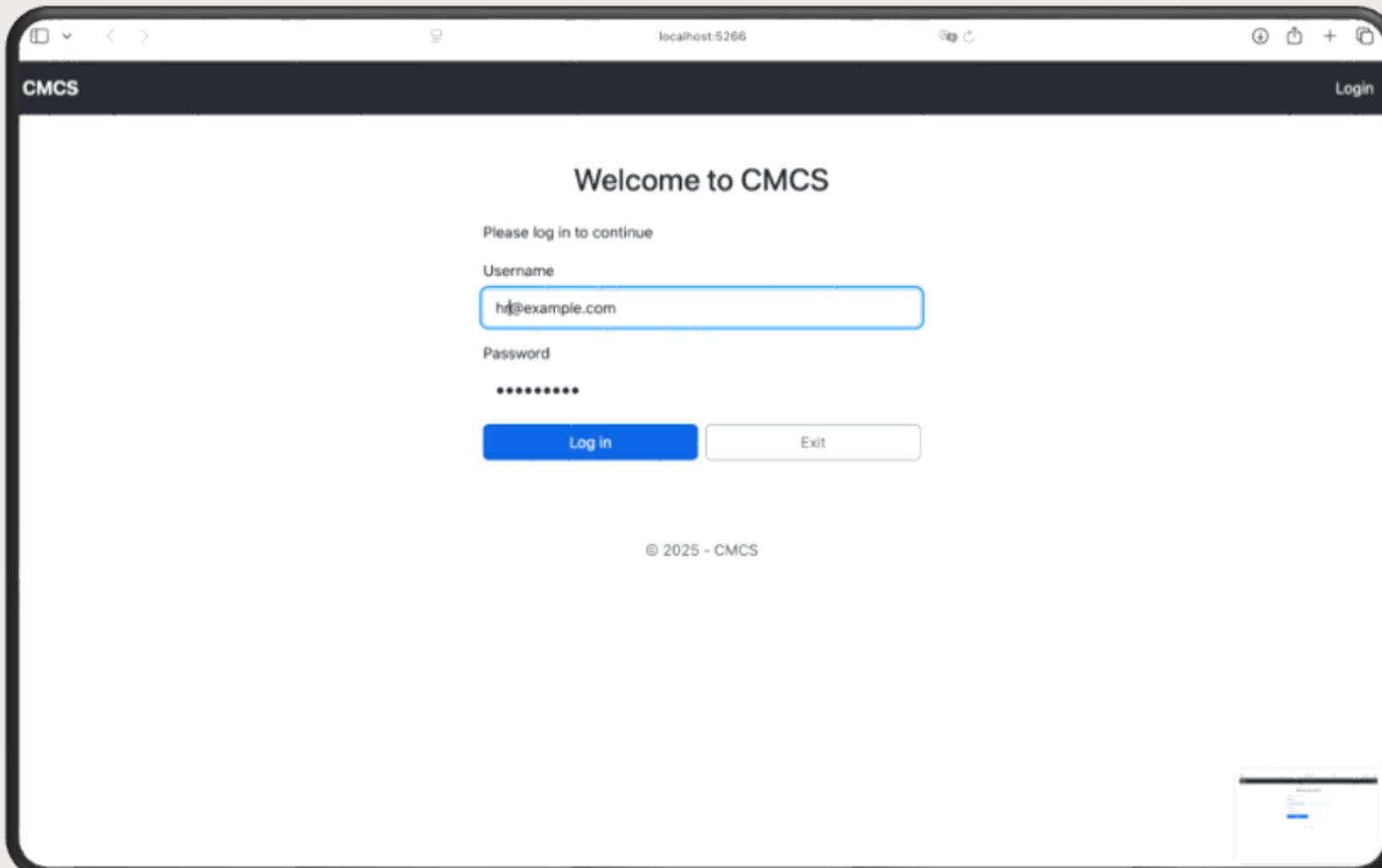
Actions trigger status updates visible

ACADEMIC MANAGER: REVIEW QUEUE CODE

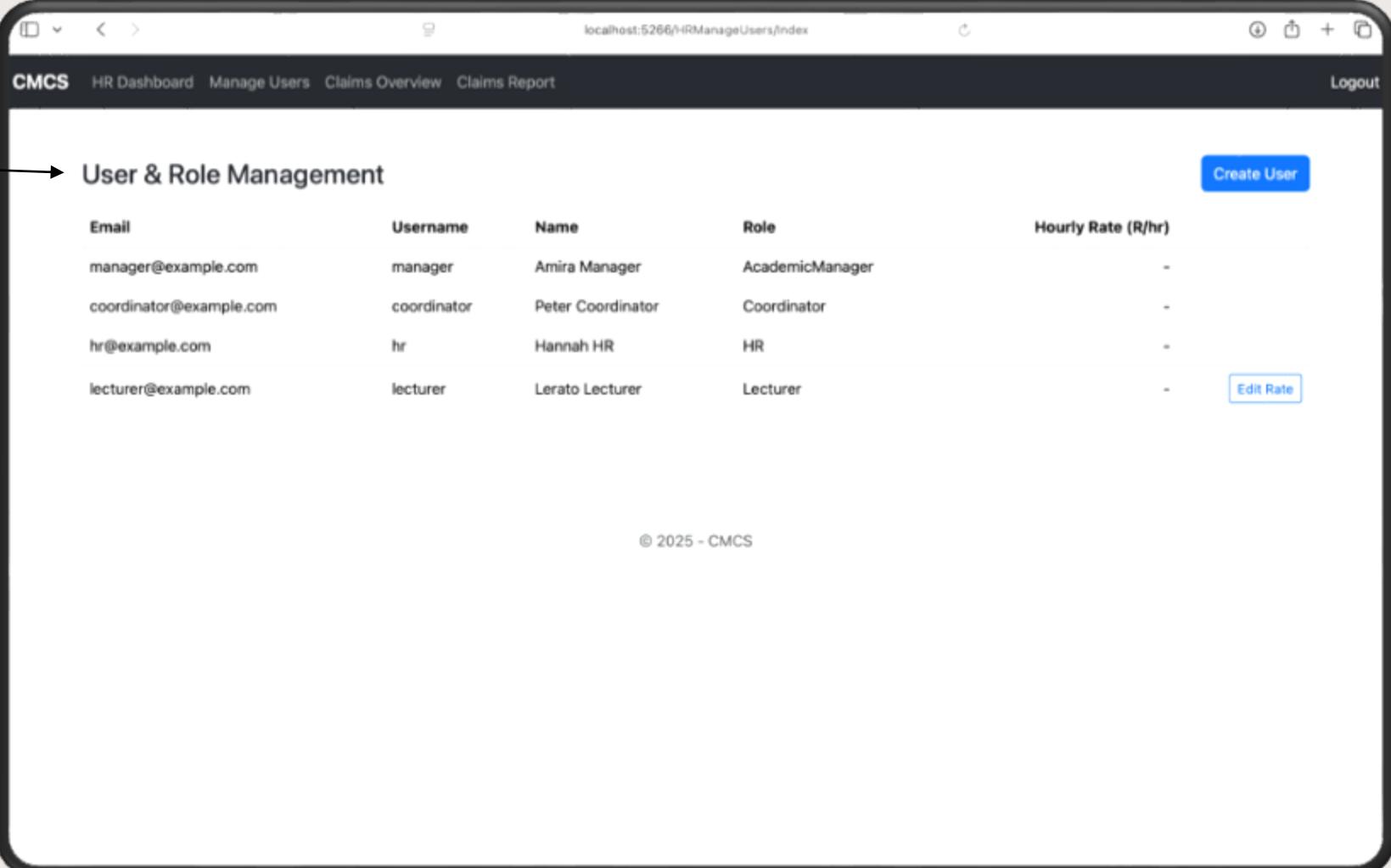
HUMAN RESOURCES

- ✓ Create and manage all users.
- ✓ Assign lecturer hourly payment rates.
- ✓ Generate detailed claims reporting.
- ✓ Oversee system-wide administrative operations.
- ✓ Finalize approved claims for payment.

HUMAN RESOURCES: LOG IN



HUMAN RESOURCES: DASHBOARD



Manages all user accounts → **User & Role Management**

Email	Username	Name	Role	Hourly Rate (R/hr)
manager@example.com	manager	Amira Manager	AcademicManager	-
coordinator@example.com	coordinator	Peter Coordinator	Coordinator	-
hr@example.com	hr	Hannah HR	HR	-
lecturer@example.com	lecturer	Lerato Lecturer	Lecturer	-

[Create User](#) [Edit Rate](#)

© 2025 - CMCS

HUMAN RESOURCES: CREATE USER

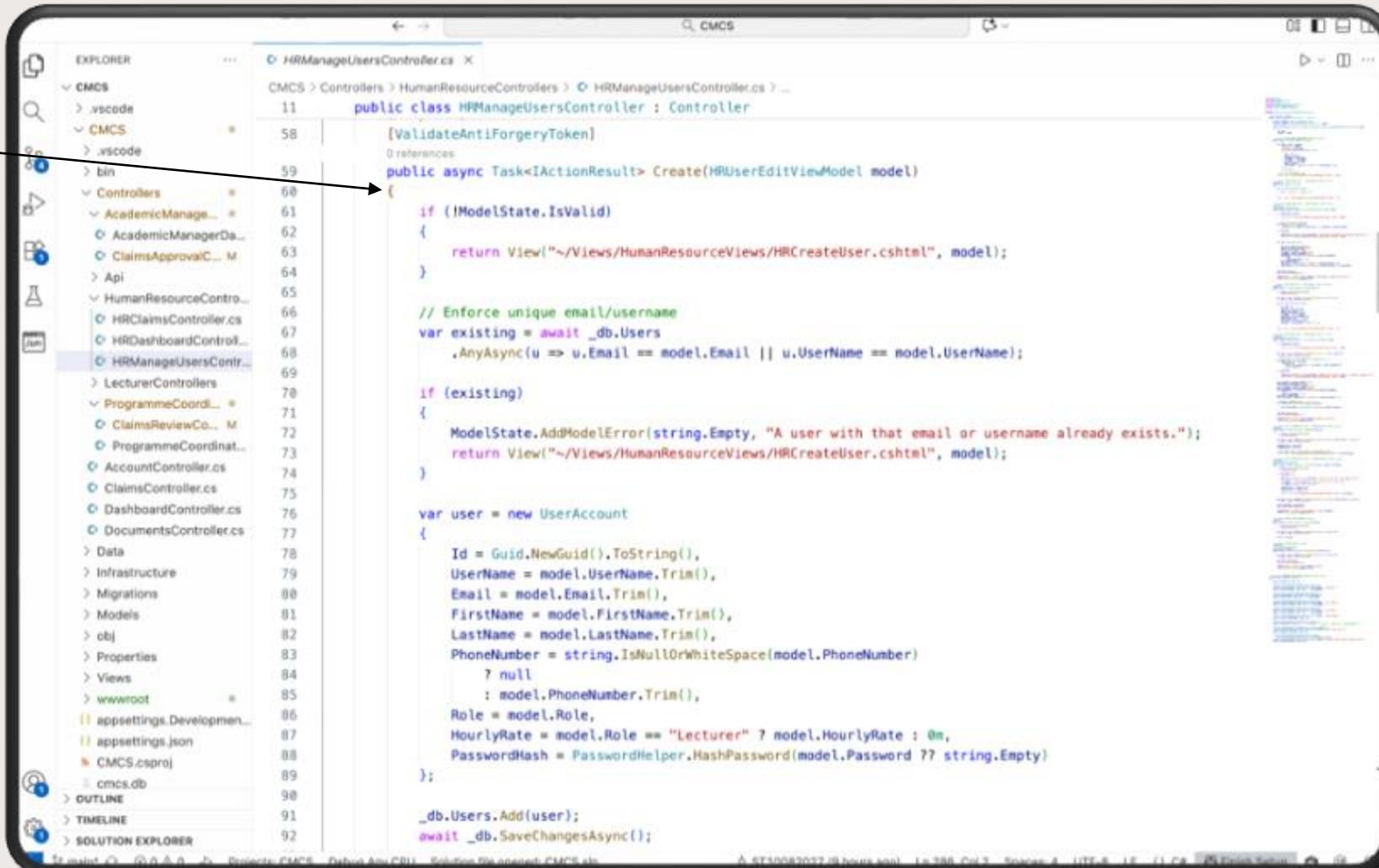
The screenshot shows a web browser window with the URL `localhost:5266/HRManageUsers/Create`. The page title is "Create User". The form fields are as follows:

- Email: st10082027@vcconnect.edu.za
- Username: J.H.Prinsloo
- First Name: John-Hoffman
- Last Name: Prinsloo
- Phone Number: 012 354 7689
- Role: Lecturer
- Hourly Rate (R/hr): 1075
- Only required for Lecturers (IC users).
- Password (leave blank to keep current):
- Initial password for the new user.
- Buttons: Cancel (grayed out) and Create (blue)

Acts as the super user / system admin

HUMAN RESOURCES: CREATE USER CODE

HR creates user accounts, assigns roles and lecturer hourly rates



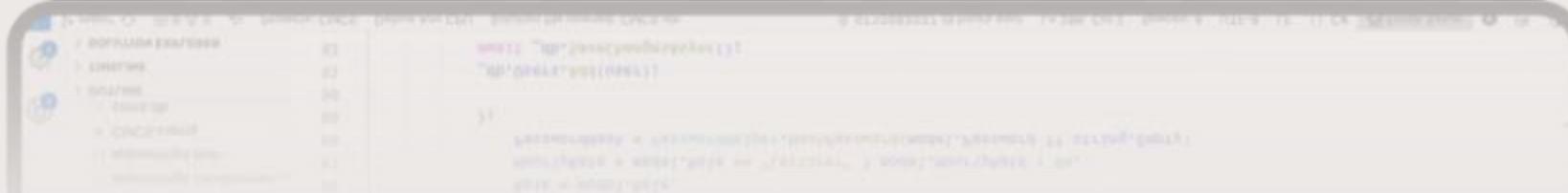
```
public class HRManageUsersController : Controller
{
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Create(HRUserEditViewModel model)
    {
        if (!ModelState.IsValid)
        {
            return View("~/Views/HumanResourceViews/HRCreatelUser.cshtml", model);
        }

        // Enforce unique email/username
        var existing = await _db.Users
            .AnyAsync(u => u.Email == model.Email || u.UserName == model.UserName);

        if (existing)
        {
            ModelState.AddModelError(string.Empty, "A user with that email or username already exists.");
            return View("~/Views/HumanResourceViews/HRCreatelUser.cshtml", model);
        }

        var user = new UserAccount
        {
            Id = Guid.NewGuid().ToString(),
            UserName = model.UserName.Trim(),
            Email = model.Email.Trim(),
            FirstName = model.FirstName.Trim(),
            LastName = model.LastName.Trim(),
            PhoneNumber = string.IsNullOrWhiteSpace(model.PhoneNumber)
                ? null
                : model.PhoneNumber.Trim(),
            Role = model.Role,
            HourlyRate = model.Role == "Lecturer" ? model.HourlyRate : 0m,
            PasswordHash = PasswordHelper.HashPassword(model.Password ?? string.Empty)
        };

        _db.Users.Add(user);
        await _db.SaveChangesAsync();
    }
}
```



HUMAN RESOURCES: ALL CLAIMS

The screenshot shows a web-based Human Resources management system (CMCS) interface. The title bar includes the CMCS logo, HR Dashboard, Manage Users, Claims Overview, Claims Report, and Logout links. The main content area is titled "All Claims" and displays a table of four rows representing different claims. The columns are labeled: IC, Month, Hours, Rate, Amount, Docs, Status, Submitted, and Remark. The data in the table is as follows:

IC	Month	Hours	Rate	Amount	Docs	Status	Submitted	Remark
lecturer@example.com	2025-01	46	650	29900	1	ApprovedByManager	2025-11-21 19:03	Approved by Academic Manager.
lecturer@example.com	2025-10	46	650	29900	1	Pending	2025-11-21 19:01	
lecturer@example.com	2025-02	4	650	2600	1	Pending	2025-11-21 16:39	
lecturer@example.com	2025-11	65,1	650	42315,0	1	ApprovedByManager	2025-11-21 16:10	Approved by Academic Manager.

A callout box with the text "Finalises approved claims before payment processing" has an arrow pointing to the "ApprovedByManager" status column in the third row of the table.

© 2025 - CMCS

HUMAN RESOURCES: CLAIMS REPORT

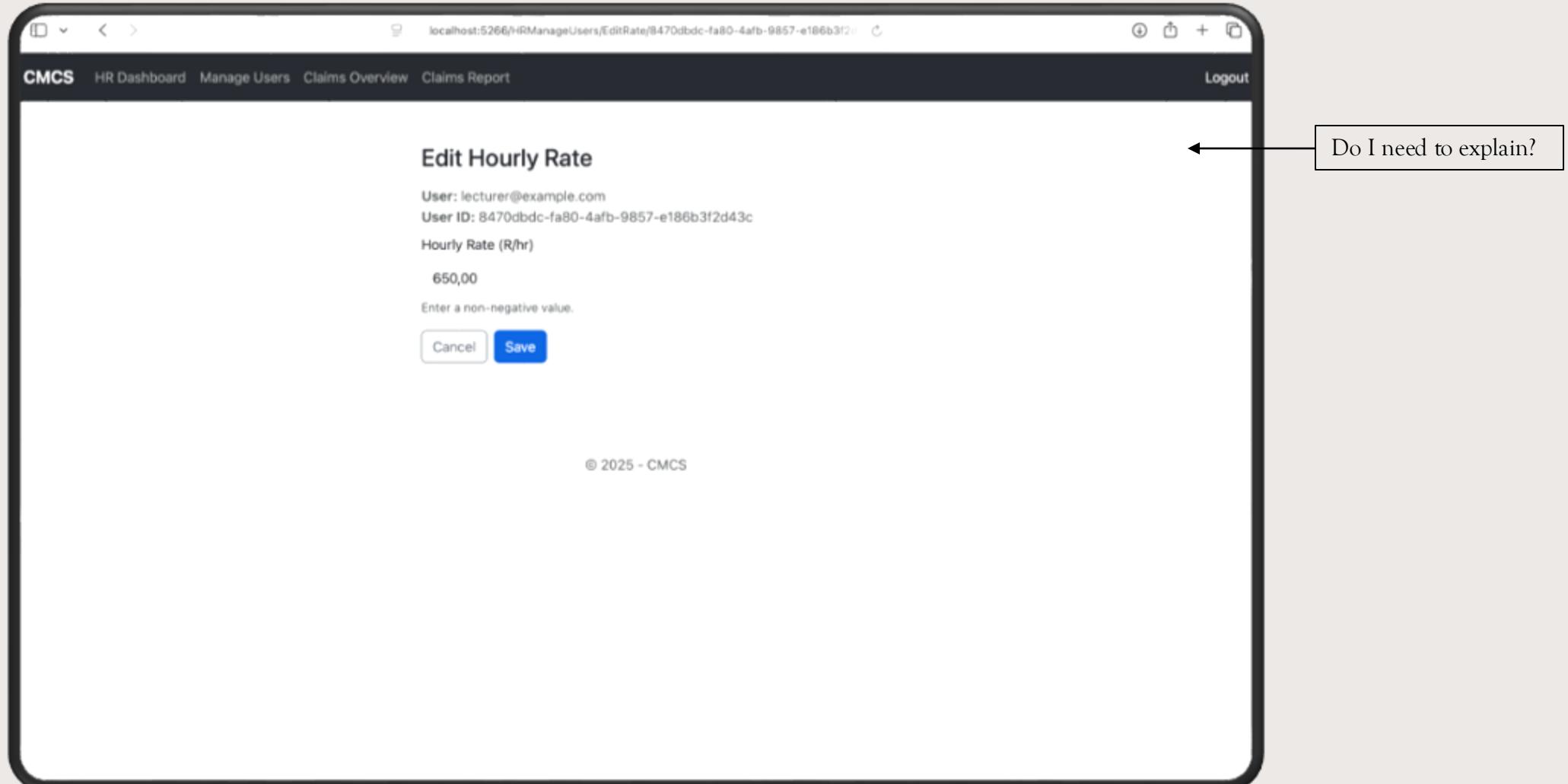
The screenshot shows a web browser window for the CMCS HR Dashboard. The title bar reads "localhost:5266/HRClaims/Report". The main content area is titled "Claims Report" and displays an automatic summary of claims for payment processing. It includes filters for "Month" (All months) and "Status" (ApprovedByManager), with a "Generate report" button. Summary statistics are shown: TOTAL CLAIMS (2), TOTAL HOURS (111,1), and TOTAL AMOUNT (R) 72215,00. Below this is a detailed table of claims:

Claim #	Lecturer	Lecturer Email	Role	Month	Hours	Rate (R/hr)	Amount (R)	Status	Created	Submitted	Verified	Approved	Reject
4	Lerato Lecturer	lecturer@example.com	Lecturer	2025-01	46	650,00	29900,00	ApprovedByManager	2025-11-21 19:03	2025-11-21 21:19:03	2025-11-21 19:08	2025-11-21 19:10	-
1	Lerato Lecturer	lecturer@example.com	Lecturer	2025-11	65,1	650,00	42315,00	ApprovedByManager	2025-11-21 16:10	2025-11-21 21:16:10	2025-11-21 16:14	2025-11-21 21:16:15	-

© 2025 - CMCS

A callout box on the right side of the interface points to the "Generate report" button with the text: "Generates invoices or reports for approved claims".

HUMAN RESOURCES: EDIT HOURLY RATE





THANK YOU FOR YOUR TIME