

Installation Guide and User Manual

Project: UniManage - University Course Management System (UCMS)

Module: CS6004ES Application Development

Assessment: Coursework 2

Student: Yasas Pasindu Fernando

Student ID: 25026764

Submission Type: Approved individual submission

Companion Documents: Main Application Development Report and Individual Contribution Report

1. Introduction

This document is the installation guide and user manual for UniManage, the University Course Management System developed for CS6004ES Application Development Coursework 2. It is written for the marker, external moderator, and any reader who needs to install, run, demonstrate, or operate the application. The steps are based on the actual project files and the local development setup.

The manual is organised into fourteen sections. Sections 1 to 5 cover requirements, package contents, installation, and first-time setup. Sections 6 to 9 explain login and the Student, Lecturer, and Administrator workflows. The later sections cover validation messages, reports, security notes, troubleshooting, responsive use, and the viva demonstration order.

Following approval to complete the coursework individually, the author completed the implementation, testing, documentation, and final preparation as an individual submission. The manual uses UK English and a practical step-by-step style.

2. System Requirements

The following items are required to install and run UniManage on a development workstation.

Item	Requirement	Notes
Operating system	Windows 10 or later	The development environment is Windows; macOS and Linux can run the application using the .NET 8 SDK and Pomelo MySQL provider, but Visual Studio integration is best on Windows.
Development environment	Visual Studio 2022 (Community edition is sufficient)	Visual Studio 2022 or a compatible version is recommended because the project targets .NET 8.
.NET runtime	.NET 8 SDK	Required for dotnet build, dotnet run, and the Visual Studio project.
Database engine	MySQL Server 8.x	The project uses the Pomelo Entity Framework Core MySQL provider.
Database tool	Optional MySQL Workbench, HeidiSQL, or DBeaver	Used to inspect tables and capture database evidence.
Browser	Microsoft Edge, Google Chrome, or Mozilla Firefox (latest stable)	Used to access the running application.
Internet connection	Required for the first run	Used to restore NuGet packages, optional Google login, optional SMTP email, and the small number of CDN-hosted assets.
Disk space	About 1 GB free	Project source, NuGet cache, and database storage.

If the marker uses SQL Server / LocalDB instead of MySQL, the connection string and the EF Core provider must be replaced. The current implementation uses MySQL through Pomelo and is therefore the supported configuration. SQL Server / LocalDB compatibility is treated as a future compatibility improvement.

3. Project Package Contents

The project package contains the following items.

Item	Path	Purpose
Solution file	AD COURSEWORK 2.sln	Visual Studio solution file.
Project file	AD COURSEWORK 2.csproj	Web project file with package references.
Application entry point	Program.cs	Service registration, middleware pipeline, and seeding.
Controllers	Controllers/	Thirteen controllers handling the request flow.
Domain models	Models/	Twelve model classes for academic entities.
View models	ViewModels/	Eighteen view models for forms and dashboards.
Razor views	Views/	Fifty .cshtml views including shared layouts and partials.
Data access	Data/	ApplicationDbContext, factory, seeder, and EF Core migrations.
Infrastructure	Infrastructure/	Thirteen infrastructure files: audit logger, email service, file store, security headers middleware, CSV writer, PDF report builder, calendar link helper, meeting link helper, grading helper, and supporting interfaces.
Static assets	wwwroot/	Bootstrap, custom CSS, custom JavaScript, library folder, and images.
Configuration	appsettings.json, appsettings.Development.json	Connection string, email settings, optional Google credentials.
Launch settings	Properties/launchSettings.json	HTTP and HTTPS launch profiles for development.
Documentation	documentation_application_development/	Main report, individual contribution report, user manual, final evidence summary, source/build

Item	Path	Purpose
		links, diagrams, screenshots, and supporting appendices.
Project README	README.md	Top-level project description.

4. Installation Guide

The installation guide assumes that the marker has Windows, Visual Studio 2022, the .NET 8 SDK, and a running MySQL 8 server. The steps are kept short so that the application can be opened, configured, built, and run without unnecessary detours.

4.1 Extracting the Project

If the project has been delivered as a ZIP archive, extract the archive into a short local path such as C:\src\unimanage or C:\workspace\AD COURSEWORK 2. Avoid placing the project under a deep path with spaces and special characters because some EF Core tooling commands behave better with shorter paths.

If the project has been delivered as a Git repository, clone the repository into a similar local path:

git clone <https://github.com/YasasPasinduFernando/AD-COURSEWORK-2.git>

After extraction or cloning, the local folder should contain AD COURSEWORK 2.sln, AD COURSEWORK 2.csproj, Program.cs, the source folders listed in Section 3, and the documentation_application_development folder.

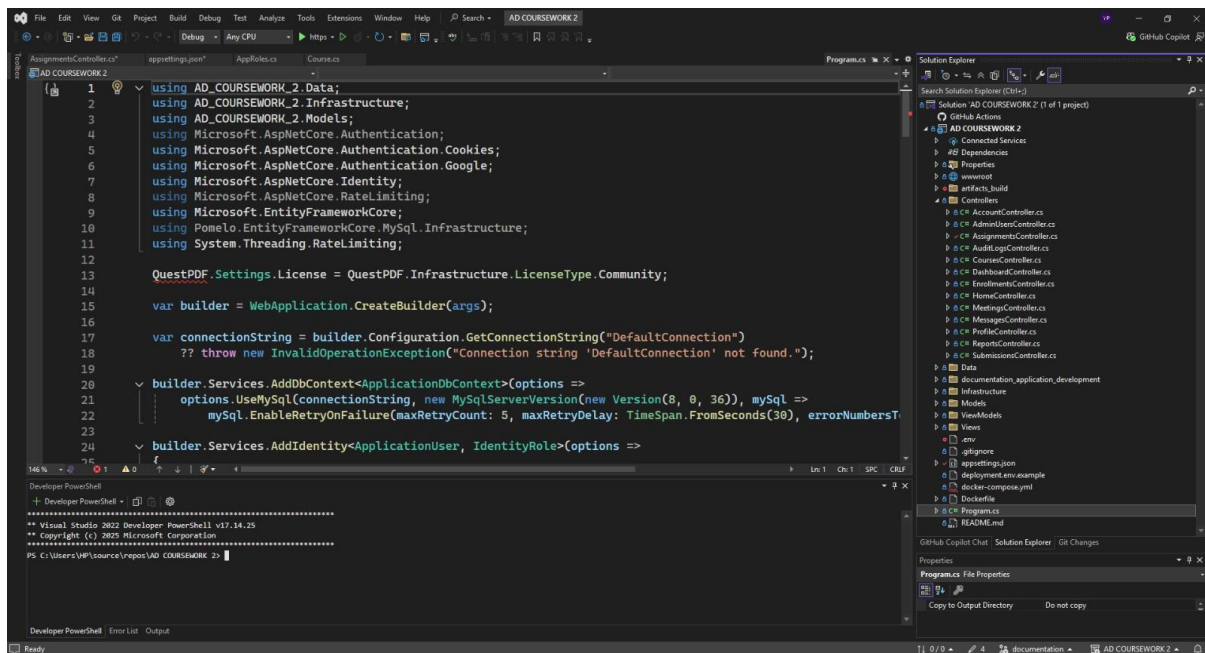


Figure UM1: The solution loaded

This screenshot shows the solution loaded in Visual Studio with the source folders visible in Solution Explorer.

4.2 Opening the Solution in Visual Studio

Double-click AD COURSEWORK 2.sln to open the solution in Visual Studio 2022. Confirm that the project loads without missing project errors. The status bar should report that the solution is ready and the Solution Explorer should display the AD COURSEWORK 2 web project together with the documentation_application_development folder if it is included in the solution.

If Visual Studio prompts to install missing components, accept the prompts so that the .NET 8 SDK and the ASP.NET Core workload are present. If the prompt suggests SQL Server LocalDB, that component is not required because the project uses MySQL.

4.3 Restoring NuGet Packages

NuGet packages are restored automatically the first time Visual Studio opens the solution. To restore manually from the command line, open a PowerShell window in the project folder and run:

```
dotnet restore
```

The main packages used by the project include:

Microsoft.AspNetCore.Identity.EntityFrameworkCore

Microsoft.EntityFrameworkCore and Microsoft.EntityFrameworkCore.Design

Pomelo.EntityFrameworkCore.MySql

Microsoft.AspNetCore.Authentication.Google

QuestPDF

If a restore failure occurs, confirm that NuGet.org is reachable and that the workstation has internet access. The exact pinned versions are recorded in AD COURSEWORK 2.csproj.

4.4 Checking appsettings.json

Open appsettings.json in the project root. Confirm the following sections.

ConnectionStrings:DefaultConnection contains the MySQL server, port, database name, user, and password to use during development.

Email contains the SMTP host, port, sender name, sender email, username, and password used by the password reset flow. If email is not required for the marker's environment, the application still runs because email is only sent during password reset.

Authentication:Google contains the optional Google client identifier and client secret. If both values are present, Google login is enabled in Program.cs. If either value is empty, Google login is disabled and the standard Identity login is used.

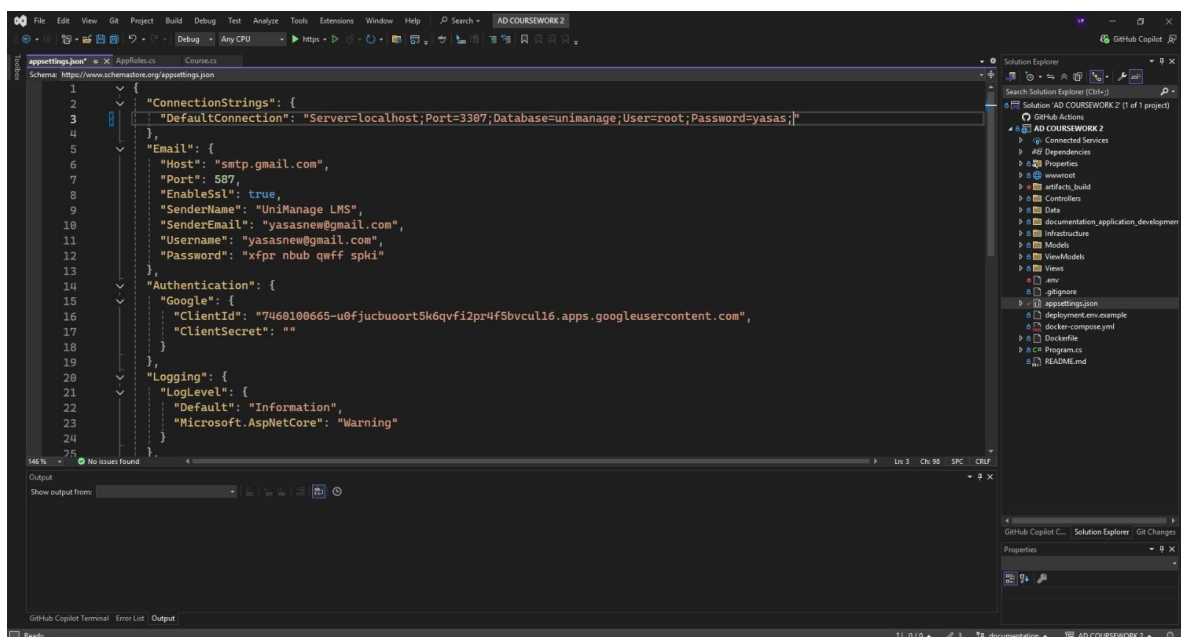


Figure UM2: The configuration file

This screenshot shows the configuration file structure used by the application.

Sensitive credentials such as database, SMTP and Google OAuth values are excluded from the report and shared screenshots. Configuration evidence uses redacted or non-sensitive values.

4.5 Configuring the Database Connection

The default connection string targets a local MySQL server on a non-standard port. The format is:

```
Server=localhost;Port=3307;Database=unimanager;User=root;Password=YOUR_PASSWORD;  
;TreatTinyAsBoolean=true;CharSet=utf8mb4;
```

To adapt the connection string to the marker's machine, edit the Server, Port, Database, User, and Password parts as appropriate. If the local MySQL server uses the default port, change 3307 to 3306. If a different database name is preferred, change unimanager to the chosen name; the application will create the schema during the first run.

The marker can also place the connection string into appsettings.Development.json so that the value is only used during development. Sensitive credentials must remain outside of source control and outside of the report.

4.6 Applying Migrations or Creating the Database

The project applies pending migrations automatically at startup through Data/DbInitializer.cs. As soon as the application runs, the schema is created and the demonstration roles, users, and sample data are seeded.

If the marker prefers to apply migrations manually before the first run, open a PowerShell window in the project folder and run:

```
dotnet ef database update
```

The current migrations are:

```
20260408041036_InitialUniManage
```

```
20260501122117_AddAuditLogAndSecurity
```

```
20260501125135_AddMeetings
```

20260506142007_AddApplicationUserDateOfBirth

To list the migrations from the command line, run:

```
dotnet ef migrations list
```

The output is the evidence captured for Figure 7b in the main report. Raw SQL backup scripts and SQL Server / LocalDB scripts are not included in the project; the EF Core migrations are the supported approach.

4.7 Building the Project

To build the project from the command line, run the following from the project folder:

```
dotnet build --no-restore
```

The expected output is:

Build succeeded.

0 Warning(s)

0 Error(s)

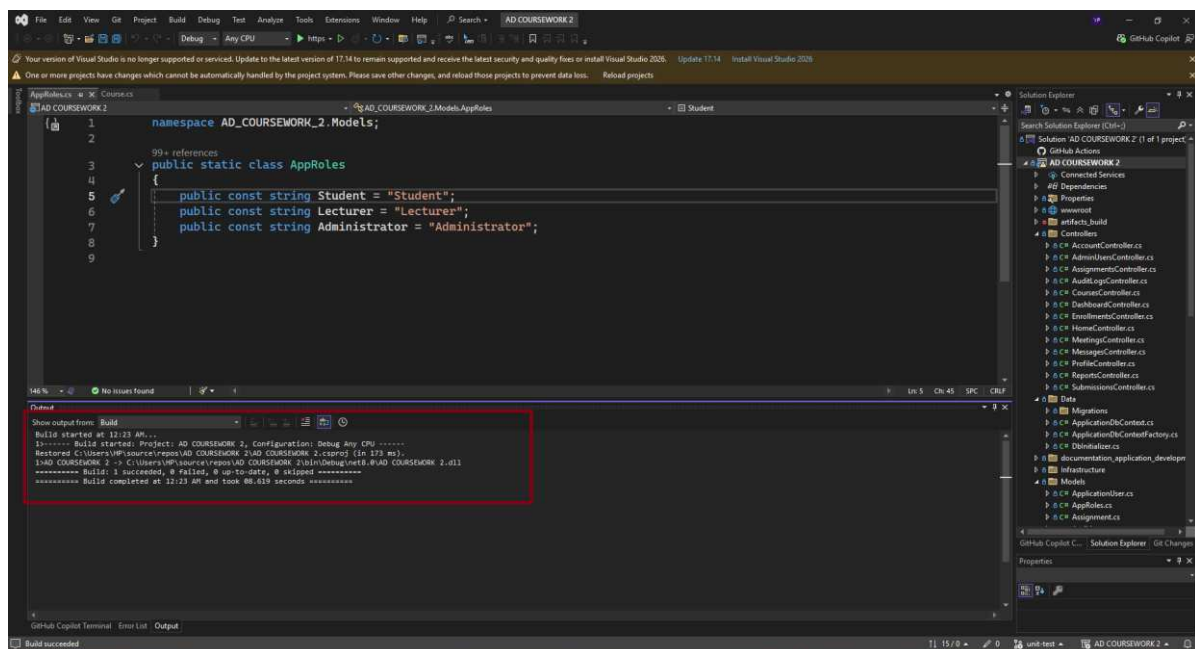


Figure UM3: Successful Build Output

This screenshot shows the Visual Studio build output with the build completed successfully and no failed projects.

To build inside Visual Studio, select Build, then Build Solution from the top menu. The Output window displays the same zero warning and zero error result.

4.8 Running the Application

To run the application from the command line, use:

```
dotnet run
```

To run inside Visual Studio, press F5 or click the Start button on the main toolbar. The Visual Studio launch profile uses the HTTPS profile by default.

The detected launch URLs are:

<https://localhost:7212>

<http://localhost:5103>

The first run applies the migrations, seeds the demonstration roles and users, and starts listening on the launch URLs. Open a browser and navigate to the HTTPS URL. The home page is displayed. From there, the user can log in or register.

4.9 Troubleshooting Build Errors

Error	Likely cause	Suggested fix
The .NET 8 SDK is missing	Visual Studio components are out of date	Open Visual Studio Installer, modify the workload, and add the .NET 8 SDK.
NuGet restore fails	No internet access or NuGet feed not reachable	Confirm internet access, run <code>dotnet nuget list source</code> , and add https://api.nuget.org/v3/index.json if missing.
EF Core tools not found	Global tool not installed	Run <code>dotnet tool install --global dotnet-ef</code> .
Database connection refused	MySQL not running or wrong port	Start the MySQL service and confirm the port matches the connection string.
Access denied for the database user	Wrong user or password	Recreate the user, grant rights to the unimanage database, and update the connection string.
Migration fails	Database user does not have CREATE privilege	Grant the user the necessary privileges or use a higher-privilege user for the first run.
dotnet build fails	Stale obj/bin folders or wrong .NET SDK	Run <code>dotnet clean</code> and <code>dotnet build --no-restore</code> again.
Port already in use	Another process is using 5103 or 7212	Stop the other process or change the port in <code>Properties/launchSettings.json</code> .

Error	Likely cause	Suggested fix
Static files not served	wwwroot missing or build skipped	Confirm that the wwwroot folder is present and that the build completed.
HTTPS certificate trust prompt	Development certificate not trusted	Run dotnet dev-certs https --trust once on the workstation.

5. First-Time Setup

The first time the application runs, DbInitializer creates three roles (Administrator, Lecturer, Student) and three demonstration users. Sample courses, enrolments, and assignments may also be created where the seeder has been configured to do so. The marker can sign in straight away using the demonstration credentials listed in Section 6.

If the seeded users are not visible after the first run, restart the application or check the logs for an error during seeding. The seeder is idempotent, so restarting the application should not create duplicate users, modules, enrolments, assignments, submissions, or meetings.

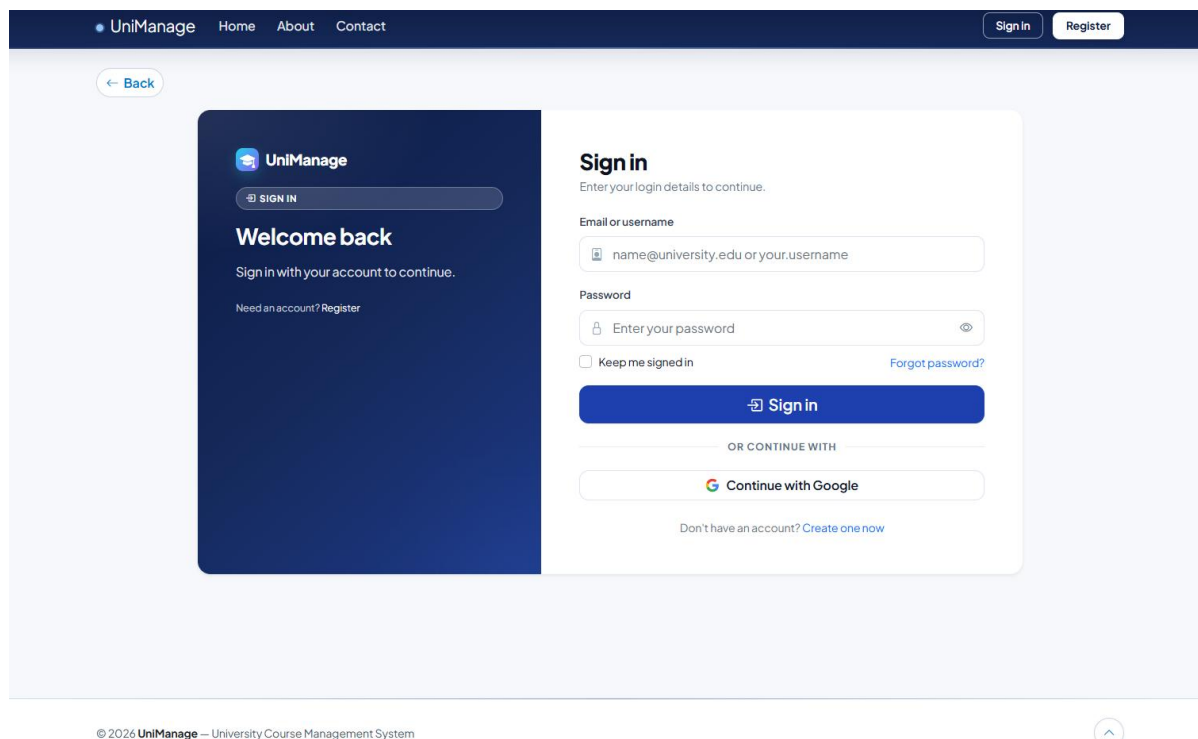


Figure UM4: Login Page

This screenshot shows the login page after the first run.

Figure UM5: User Registration Page

This screenshot shows the registration page with the role selector visible.

5A. Hosted Demonstration Access

In addition to the local installation route, a supporting hosted demonstration was prepared on Google Cloud. The custom domain is <https://lms.yasaboy.com/>, with hosted application evidence in Figure UM19 and DNS/server evidence in Figures UM21 and UM22. The hosted route can support viva and marker review, but it is supporting evidence only. The local Visual Studio route remains the primary assessment path.

The hosted URL is provided to make review and testing easier. If the hosted service is unavailable, the local Visual Studio installation and run method remains the primary assessment route.

5A.1 Hosted URL

The primary hosted demonstration URL is:

<https://lms.yasaboy.com/>

The earlier IP-based endpoint, which was used during the initial deployment phase and is retained as a backup in case the custom-domain endpoint is temporarily unreachable, is:

<http://34.171.6.234:8080>

The custom domain was configured for the hosted demonstration version. HTTPS/domain access was used where available during final testing. The earlier IP-based URL is kept only as fallback evidence, and the local Visual Studio route remains the main assessment route.

5A.2 Browser Access Steps

Open a modern browser such as Microsoft Edge, Google Chrome, or Mozilla Firefox.

Go to <https://lms.yasaboy.com/>.

Confirm that the UniManage homepage loads. If the page is correct, the address bar reflects the HTTPS or HTTP state for that deployment route.

Use Sign in or Register depending on the demonstration task. The seeded demonstration credentials listed in Section 6 of this manual are sufficient.

If the hosted URL is unavailable, use the local Visual Studio run method described in Section 4 of this manual. The earlier IP-based endpoint <http://34.171.6.234:8080> may also be tried as a temporary backup.

5A.3 Expected Landing Page

The expected landing page is the UniManage public home page rendered by Views/Home/Index.cshtml. The page offers two primary actions: "Create your account" linking to /Account/Register and "I already have an account" linking to /Account/Login. The footer shows the project name and the copyright line.

5A.4 Login on the Hosted Version

After the home page loads, the marker can sign in using the seeded demonstration users defined in `Data/DbInitializer.cs`. The credentials are listed in Section 6 of this manual. The hosted version uses the same role-based redirection as the local version, so signing in as the seeded administrator, lecturer, or student opens the appropriate dashboard.

No additional credentials are exposed beyond the seeded demonstration users. Real student data, real lecturer data, and production-style credentials are not used on the hosted demonstration.

5A.5 Availability Warning

The hosted demonstration depends on the Google Cloud virtual machine and Docker container being in a running state. Cloud instances can be stopped, restarted, or reconfigured between the time of writing and the marking session. Before relying on the hosted URL, the marker should open it in a browser. If the custom-domain URL does not open, the earlier IP endpoint may be tried as a fallback. If neither endpoint is reachable, use the local installation route in Section 4.

5A.6 Security Notes for the Hosted Demonstration

The intended custom-domain endpoint is HTTPS, but the wider deployment is still a coursework demonstration rather than a production service. The marker is asked not to enter real personal data on the demonstration site.

Database passwords, SMTP passwords, Google client secrets, SSH keys, DNS or provider credentials, and cloud service account keys are not included in this manual or in the body of the report. These values should be supplied only through local configuration or environment variables when running the application.

If the marker chooses to demonstrate the hosted login on screen, the seeded demonstration credentials are sufficient. There is no need to expose any other account information.

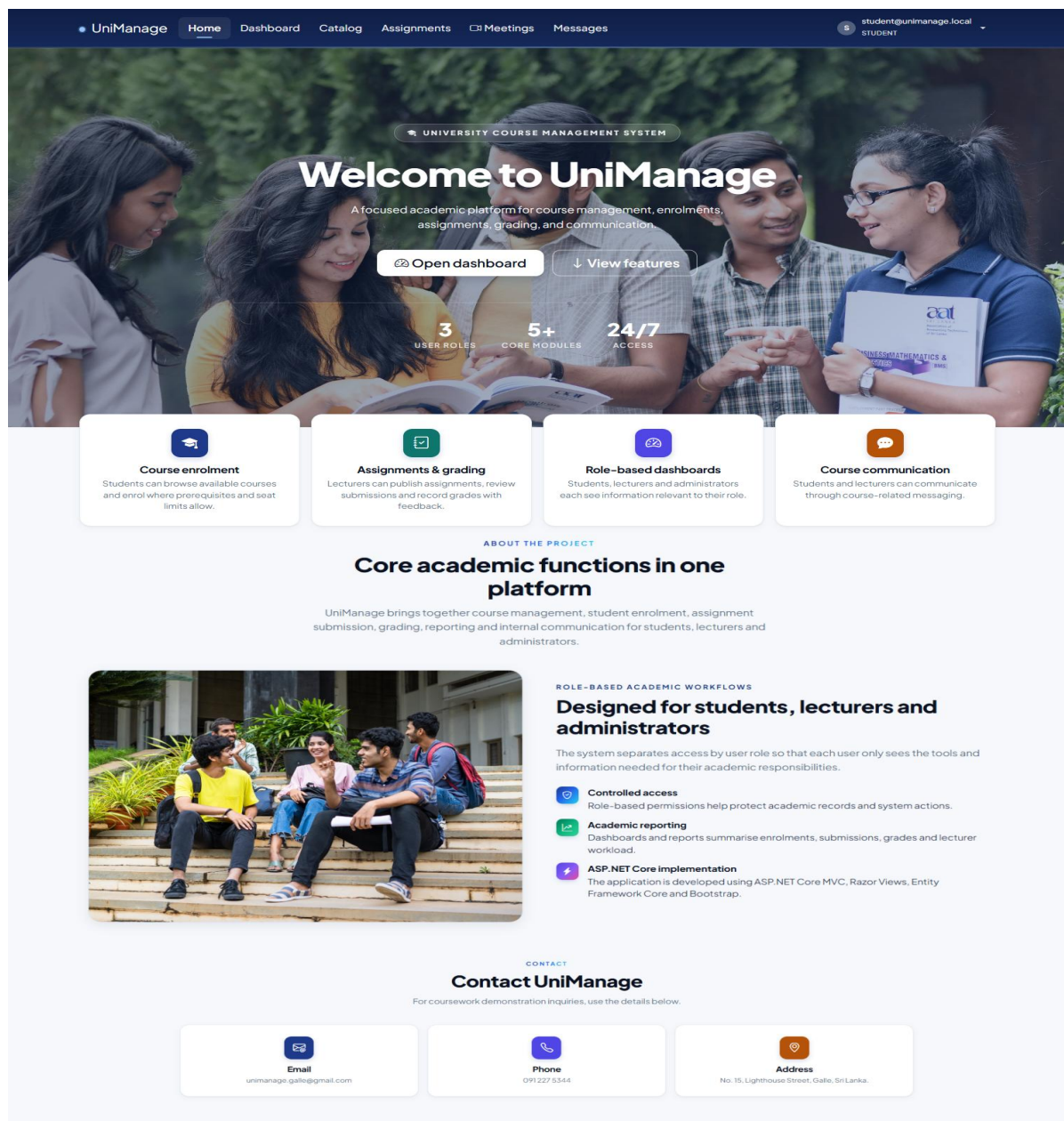


Figure UM19: Hosted UniManage Application

This screenshot shows the UniManage application running through the hosted custom domain.

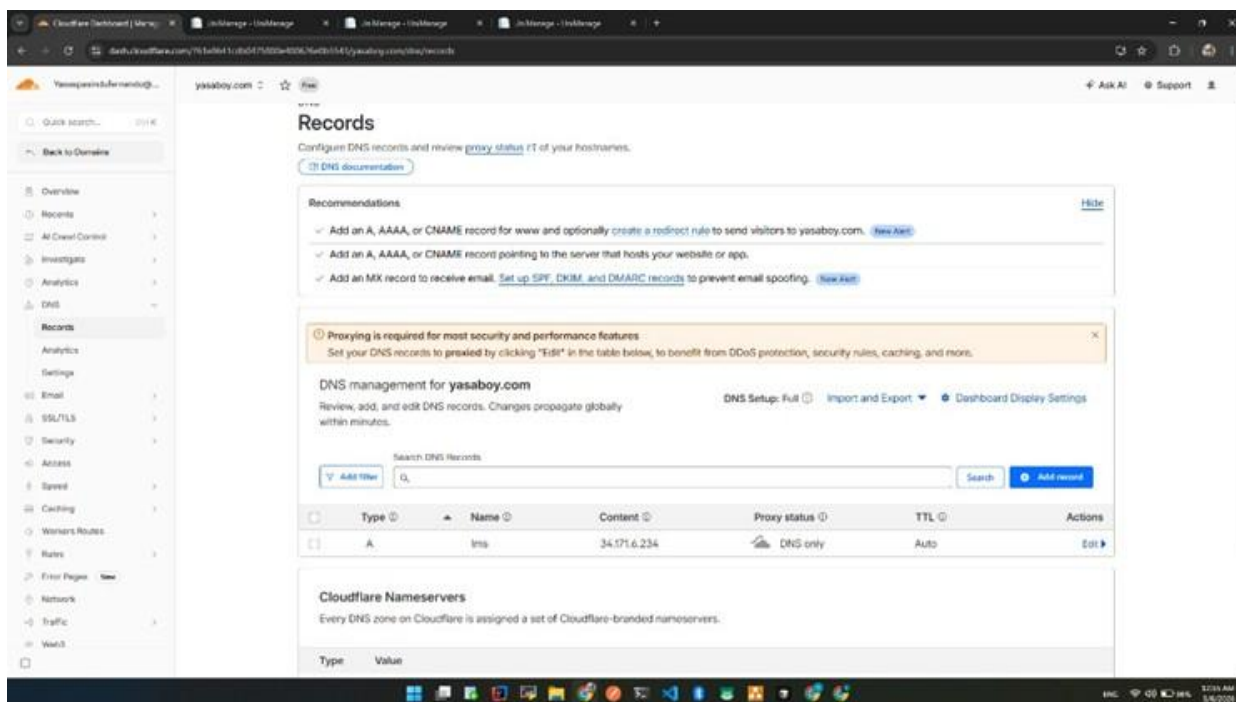


Figure UM21: Cloudflare DNS Record Evidence

This screenshot shows the Cloudflare DNS A record used to point the lms.yasaboy.com subdomain to the Google Cloud public IP address.

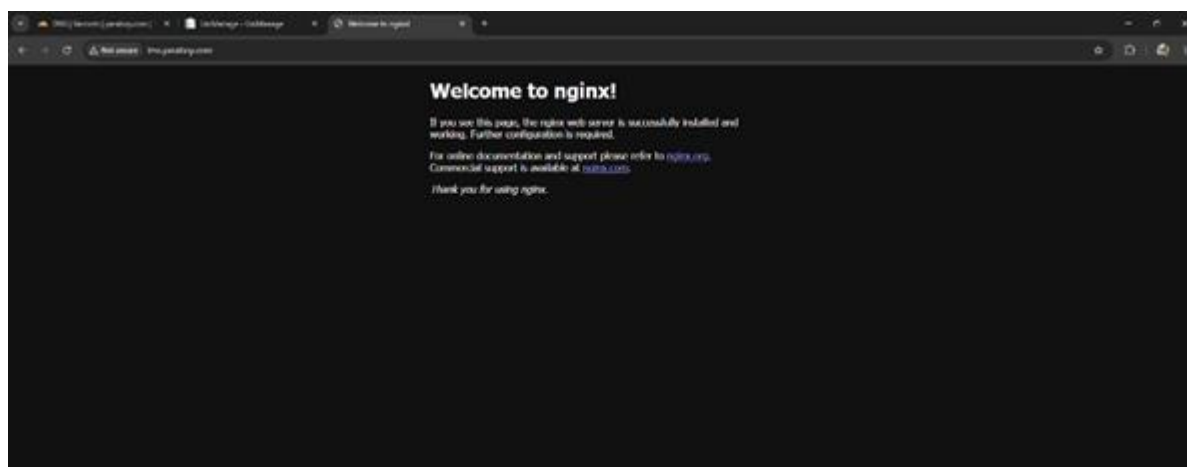


Figure UM22: Nginx Server Reachability Evidence

This screenshot shows the default Nginx page loading from the lms.yasaboy.com host, confirming that Nginx was installed and reachable on the server at the time of evidence capture.

6. Login and User Roles

The seeded demonstration accounts are created by Data/DbInitializer.cs. The credentials are intended for the viva and for marking, not for production use.

Role	Email	Password
Administrator	admin@gmail.com	Admin123!
Lecturer	lecturer@gmail.com	Lecturer123!
Student	lms@gmail.com	Student123!

To log in, open the application, navigate to the login page, enter one of the credentials above, and click Sign in. The system applies the password rules, the lockout policy (five failed attempts, fifteen minute lockout), and the rate limiter on the authentication endpoint (ten requests per minute per IP).

After a successful login, the user is redirected to the dashboard appropriate to their role. The Administrator goes to the Administrator dashboard, the Lecturer goes to the Lecturer dashboard, and the Student goes to the Student dashboard.

To register a new account, click the Register link on the login page. The user is asked for a full name, an email address, a phone number, a password, and a role (Student or Lecturer). The Administrator role is not available through self-registration. After a successful registration, the new user is signed in automatically and redirected to the role dashboard.

If Google OAuth is configured, the login page also allows Google sign-in. This option depends on the Google OAuth Client ID, Client Secret, and authorised redirect URI being configured for the current local or hosted URL. If these values are missing, Google sign-in is disabled by design and the normal seeded Identity credentials remain available.

7. Student User Guide

After logging in, the Student is redirected to the Student dashboard. The dashboard shows enrolled courses, upcoming deadlines, recent grades, recent messages, the calendar, and meetings.

The Student can perform the following actions.

View enrolled courses. The dashboard lists the Student's current enrolments. Each course title links to a detailed view.

Browse the course catalogue. Use the navigation menu to open the course browse page. The page lists eligible courses with their code, name, credits, lecturer, prerequisite, and current enrolment count.

Enrol on a course. Click the Enrol button next to an open course. The system checks that the course is not full, that the Student has not already enrolled, and that any prerequisite has been satisfied. If the checks pass, the enrolment is saved and a confirmation message is displayed. If a check fails, an error message explains the reason.

View assignments. Open the Assignments link in the navigation menu. The page lists assignments for the Student's enrolled courses with the due date and status.

Submit an assignment. Open an assignment, enter the submission text, optionally attach a file, and click Submit. The system validates the file extension, the file size, and the Student's enrolment, then stores the submission.

View grades and feedback. Open the assignment after grading. The Student can read the grade and the lecturer's feedback. After grading, the submission is locked and cannot be edited.

Send a message to a lecturer. Open the Messages page, click Compose, choose a Lecturer who teaches a course on which the Student is enrolled, enter a subject and content, and click Send. The system enforces the allowed-recipient rule.

View meetings. Open the Meetings page. Meetings linked to the Student's enrolled courses are listed with the scheduled time and the join link.

Update profile. Open the Profile page to view and update the full name, email, and phone number.

Change password. Open the Change Password page from the Profile menu, enter the current and new passwords, and click Save.

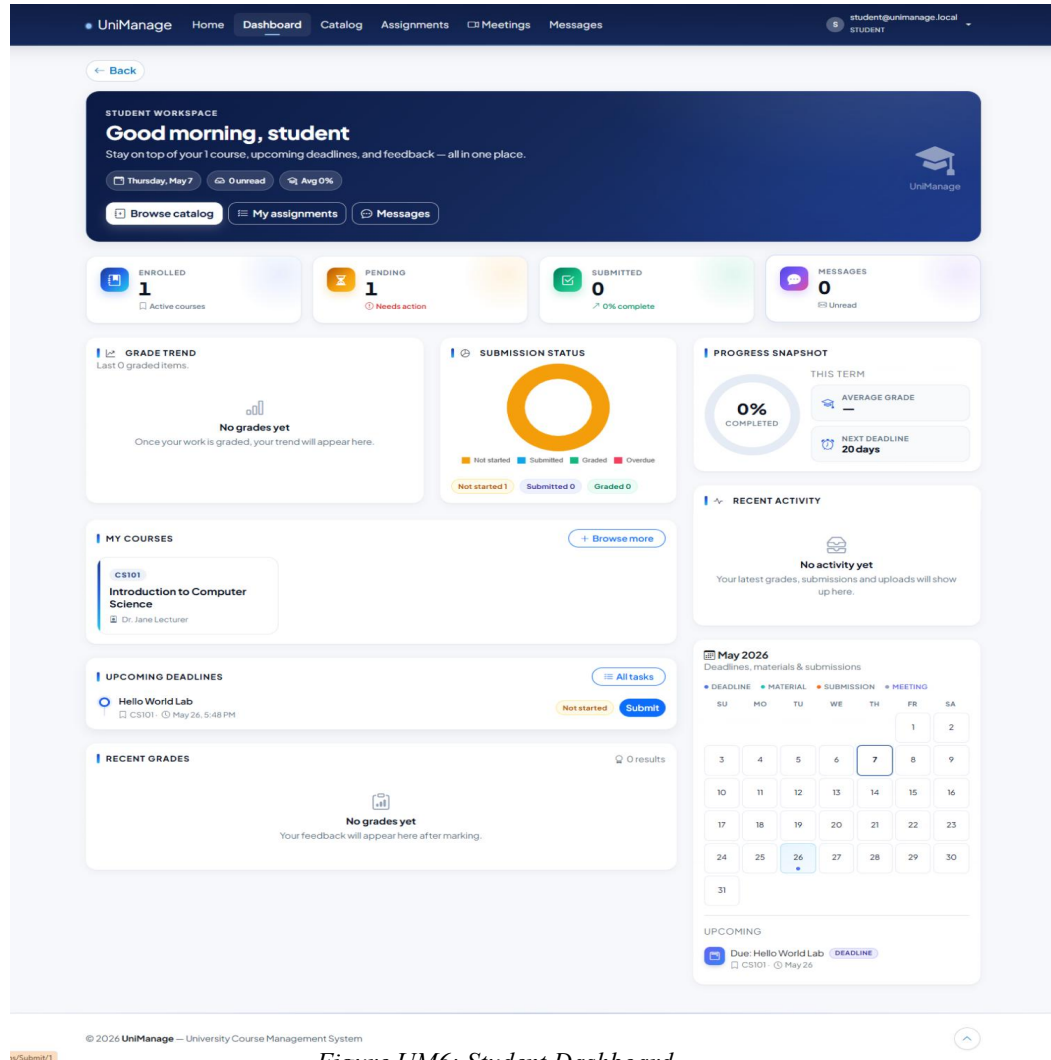


Figure UM6: Student Dashboard

This screenshot shows the Student dashboard after a successful login.

UniManage Home Dashboard **Catalog** Assignments Meetings Messages student@unimanager.local STUDENT

← Back

STUDENT CATALOG

Discover your next course
Browse 7 courses, check prerequisites, and enroll instantly.

1 joined

AVAILABLE 7 OPEN 6 ENROLLED 1 BLOCKED 0

All 7 Open 6 Enrolled 1 Blocked 0 Search by code, name

AD602 Open
Application Development
Dr. Nimal Perera 20 credits
Seats 1/120
Ready to join Enroll

AI603 Open
Artificial Intelligence
Dr. Nimal Perera 20 credits
Seats 1/120
Ready to join Enroll

ASE601 Open
Advanced Software Engineering
Dr. Nimal Perera 20 credits
Seats 1/120
Ready to join Enroll

CS101 Enrolled
Introduction to Computer Science
Dr. Jane Lecturer 4 credits
Seats 1/50
Open course Joined

CS201 Open
Data Structures
Dr. Jane Lecturer 4 credits
CS101 - Introduction to Computer Science
Seats 0/35
Ready to join Enroll

IP605 Open
Individual Project
Dr. Nimal Perera 40 credits
Seats 1/120
Ready to join Enroll

Figure UM10: Student Course Enrolment Page

This screenshot shows the course browse page with the Enroll button visible.

Figure UM12: Student Assignment Submission Page

This screenshot shows the assignment submission form with text and file fields.

8. Lecturer User Guide

After logging in, the Lecturer is redirected to the Lecturer dashboard. The dashboard shows the Lecturer's teaching courses, enrolment counts, assignment counts, pending submissions, recently graded submissions, recent messages, recent meetings, and recent activity.

The Lecturer can perform the following actions.

View taught courses. Open the My Courses page. The page lists only the courses owned by the signed-in Lecturer.

Open course details. Click a course to open its details, including the description, credits, enrolment limit, lecturer, prerequisite, enrolled students, assignments, materials, and meetings.

Upload course materials. From the course details page, click Upload Material, enter a title, select a file, and click Save. The system validates the file extension and size and stores the metadata in the database. The actual file is stored on disk through the upload service.

Create an assignment. Open the Assignments page for an owned course, click Create, and complete the form. The form requires a title, description, due date, and maximum points. After submission, the assignment is saved and listed.

Edit or delete an assignment. From the assignments list, click Edit or Delete next to an assignment. The system checks that the assignment belongs to a course owned by the Lecturer.

View submissions. Click an assignment to view its submissions. Each submission shows the Student, the submitted time, the file name where applicable, and the status.

Grade a submission. Click Grade next to a submission, enter a grade and feedback, and click Save. The system validates the grade range and the feedback length. After grading, the submission is locked and the Student can see the grade and feedback.

Send a message to a Student. Open the Messages page, click Compose, choose a Student enrolled on a course owned by the Lecturer, and send the message.

Schedule a meeting. Open the Meetings page, click Create, select a course owned by the Lecturer, enter a title, scheduled time, duration, and meeting URL, then click Save. The system validates the URL format and the schedule.

Edit or delete a meeting. From the meeting list, click Edit or Delete next to a meeting. The Lecturer can only manage meetings for owned courses.

Update profile and change password. Use the Profile menu in the same way as the Student.

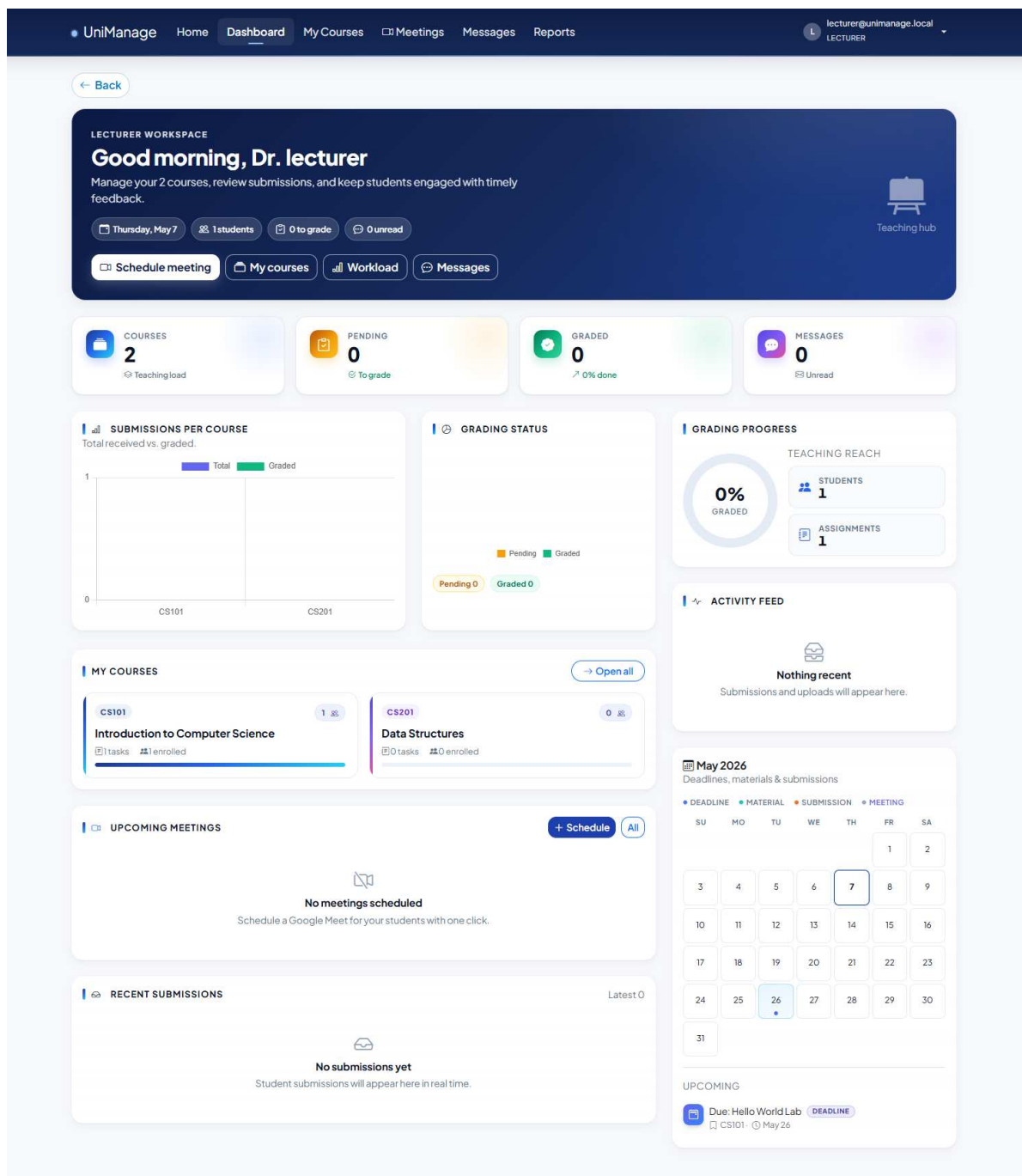


Figure UM7: Lecturer Dashboard

This screenshot shows the Lecturer dashboard with teaching summary and pending submissions.

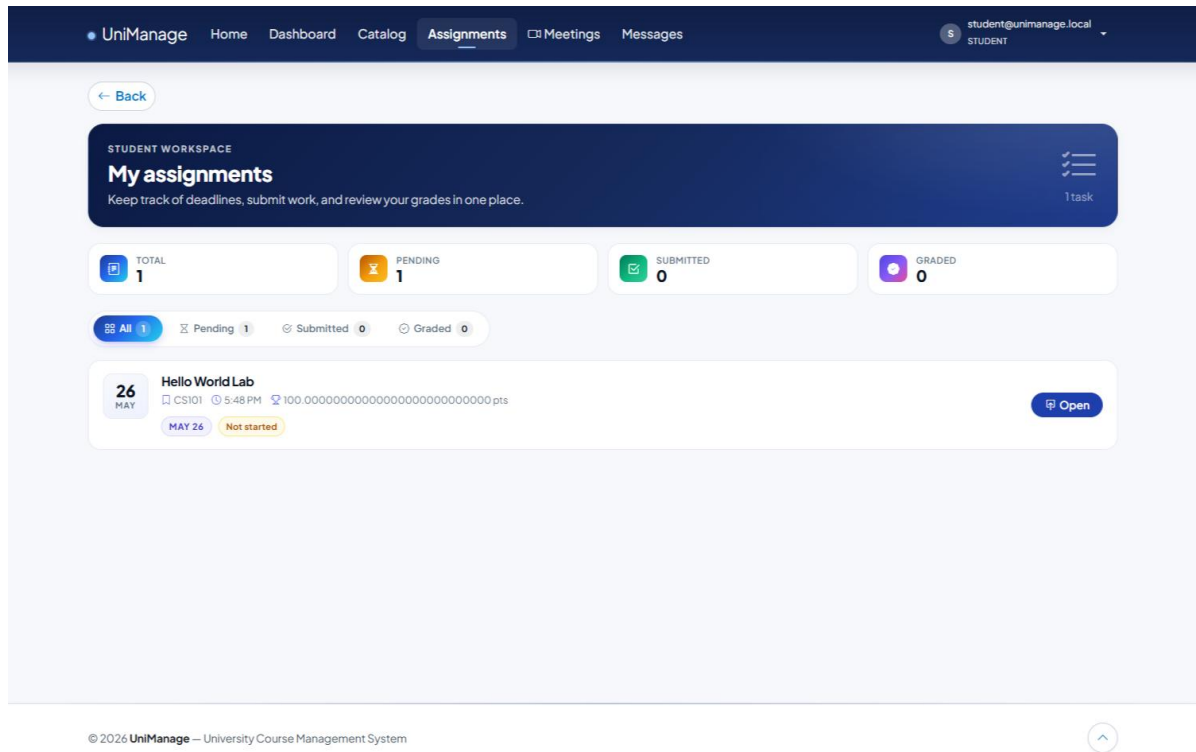


Figure UM11: Assignment Management Screen

This screenshot shows the assignment creation form.

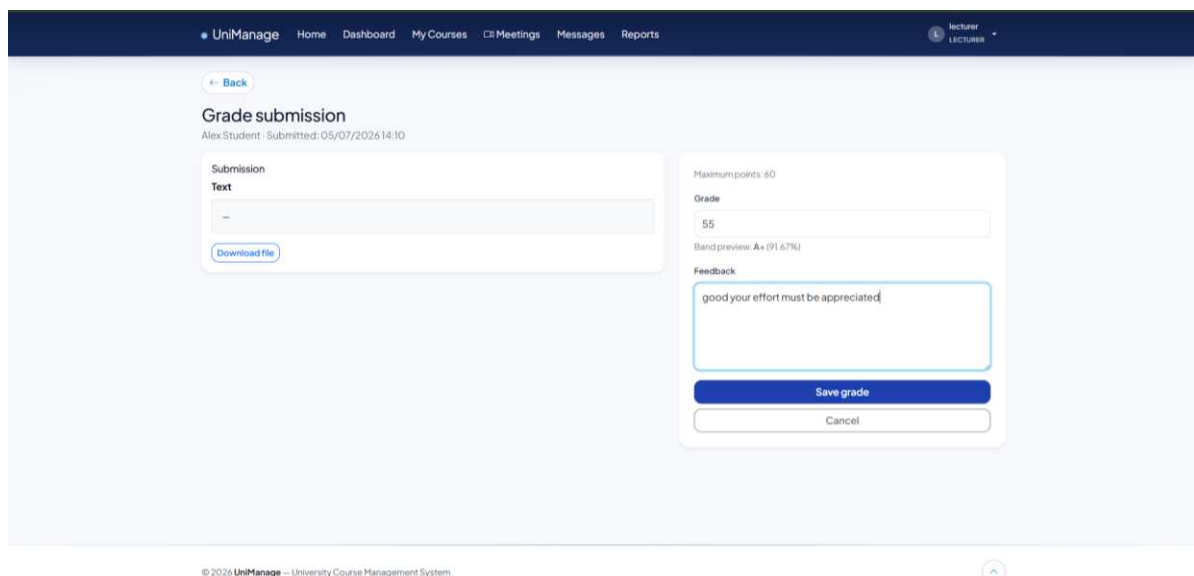


Figure UM13: Submission Grading Page

This screenshot shows the grading form with grade and feedback fields.

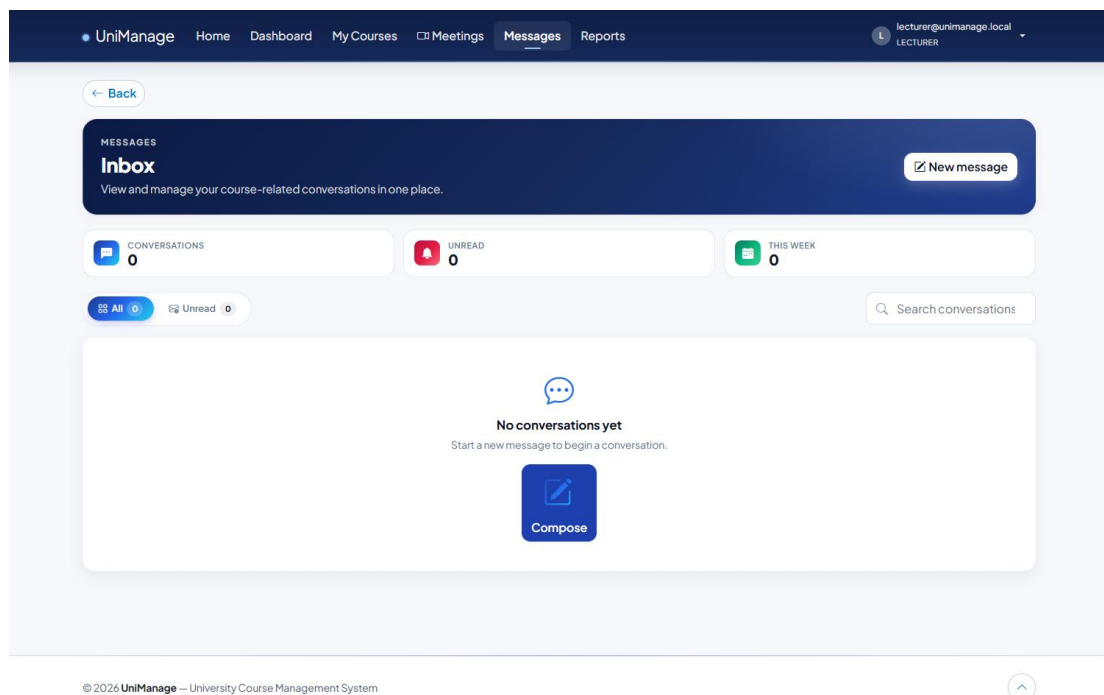


Figure UM15: Messaging Page

This screenshot shows the inbox or compose page.

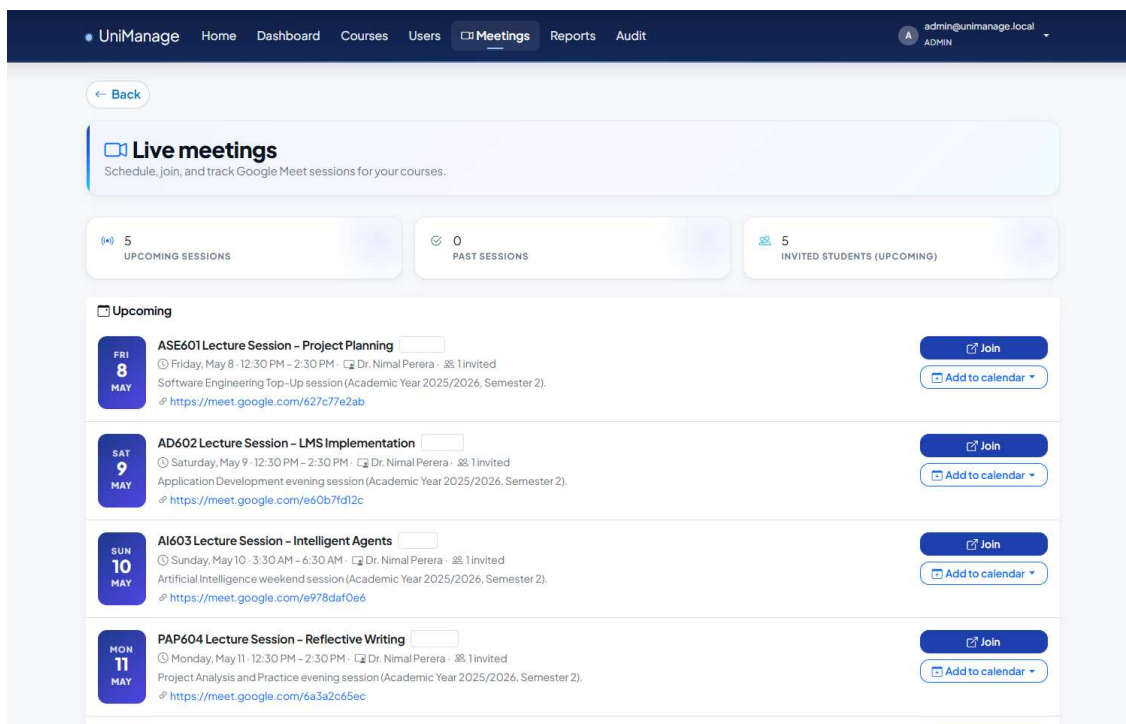


Figure UM16: Meeting Scheduling Screen

This screenshot shows the meeting create form.

9. Administrator User Guide

After logging in, the Administrator is redirected to the Administrator dashboard. The dashboard shows system-wide counts (users, courses, enrolments, assignments, submissions, messages), popular courses by enrolment count, and recent audit log activity.

The Administrator can perform the following actions.

View the dashboard. The dashboard provides a quick overview of the system.

Manage courses. Open the Courses page to list, search, page, view, create, edit, or delete courses. The course form requires the code (unique), name, description, credits, enrolment limit, lecturer, and optional prerequisite.

Manage users. Open the Admin User Management page to review registered users. The list shows username, full name, email, phone, role, lock status, and email confirmation status.

Use Create to add an account, Edit to update details or role, and Lock/Unlock to control account access.

Use account safety controls. The system blocks unsafe actions such as locking the signed-in administrator account, deleting the signed-in administrator account, deleting administrator accounts, or changing the only remaining administrator in a way that could remove administrator access.

View reports. Open the Reports page to access the six reports: course popularity, student performance, lecturer workload, enrolments, pass and fail, and assignment attendance. Each report can be exported to CSV. Selected reports can be exported to PDF through QuestPDF.

Review the audit log. Open the Audit Logs page to inspect recent events. The log records the category, action, detail, user identifier (where available), success flag, and timestamp.

Manage meetings (where authorised). Administrators can review meetings as part of the academic oversight responsibility.

Update profile and change password. Use the Profile menu in the same way as the other roles.

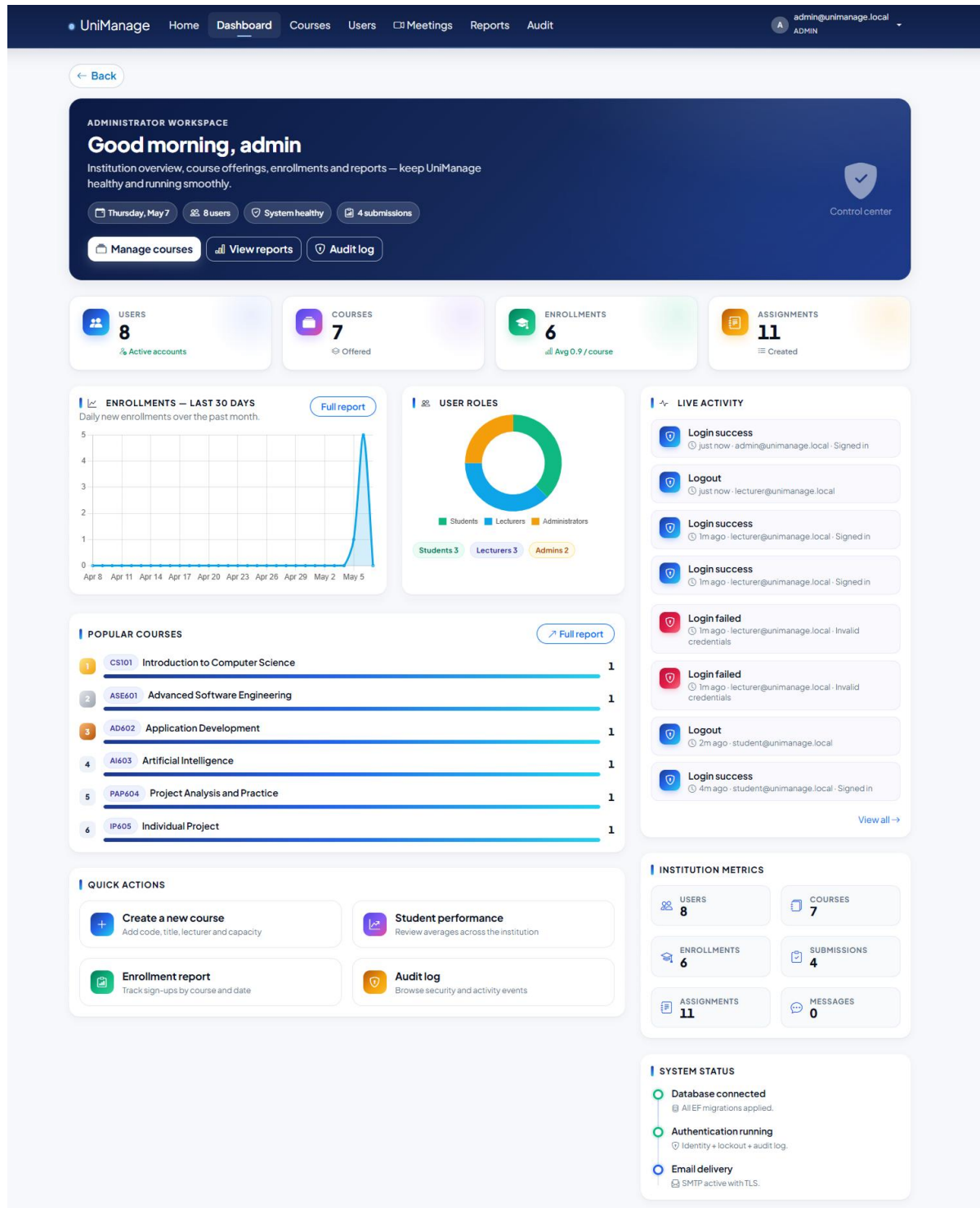


Figure UM8: Administrator Dashboard

This screenshot shows the Administrator dashboard with counts and popular courses.

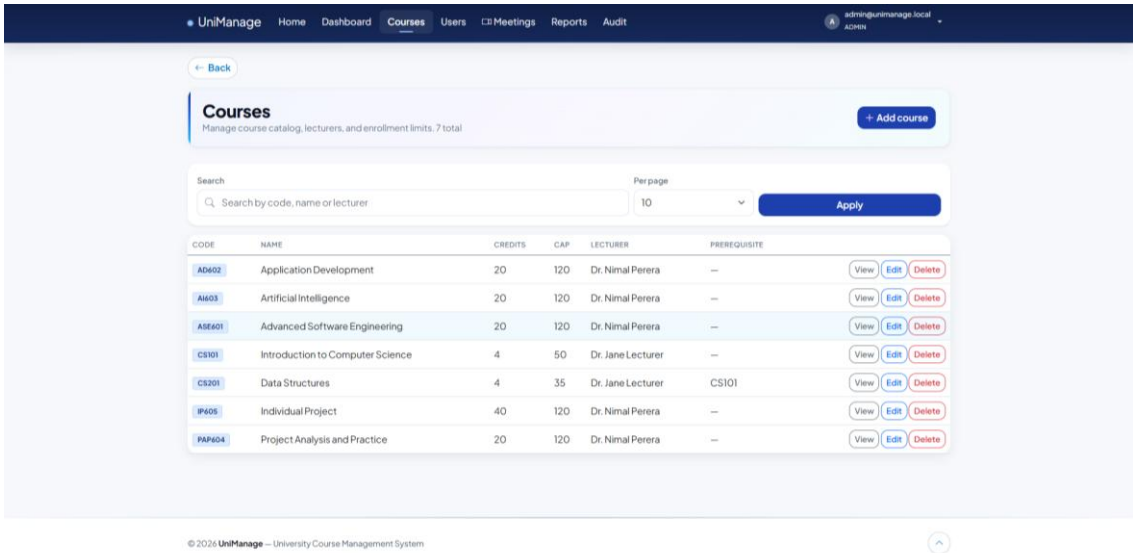


Figure UM9: Course Management Screen

This screenshot shows the course list with search and paging visible.

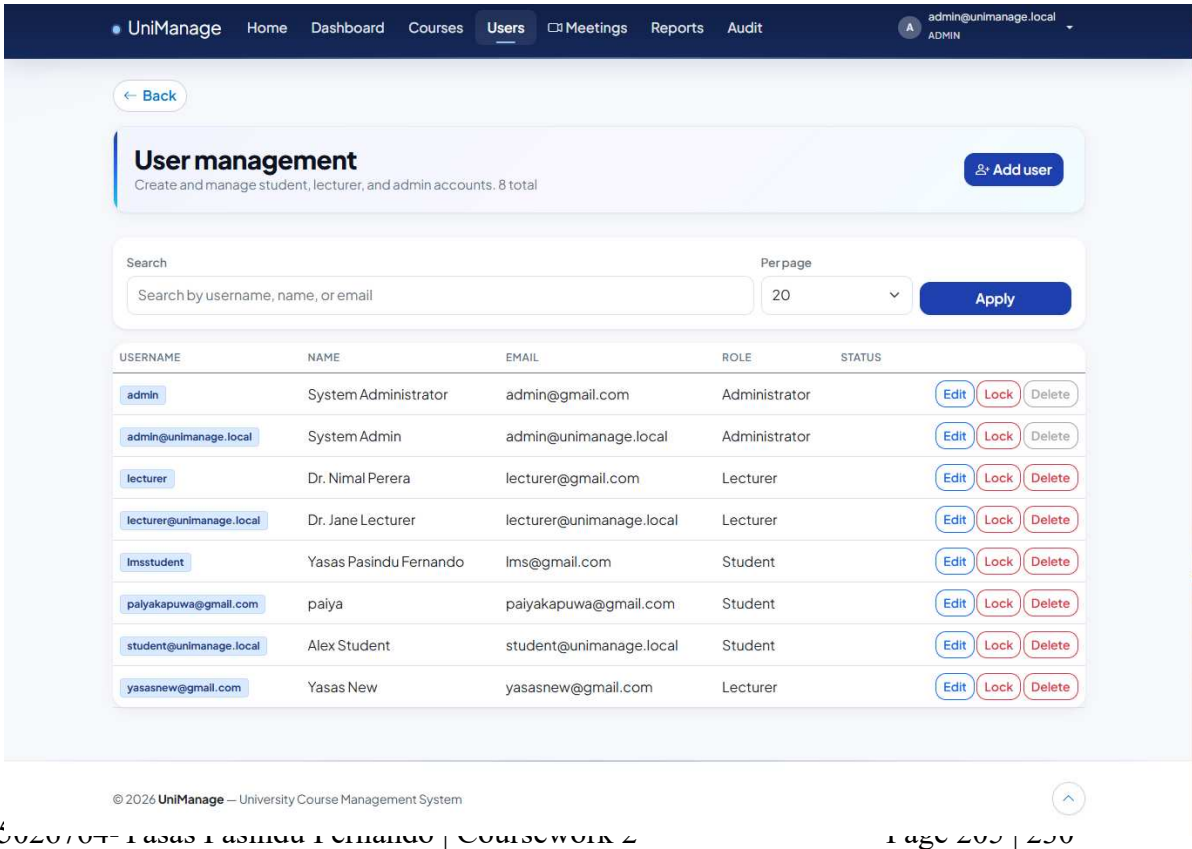


Figure UM9b: Admin User Management Screen

This screenshot shows the Admin User Management list with lock status and role information.

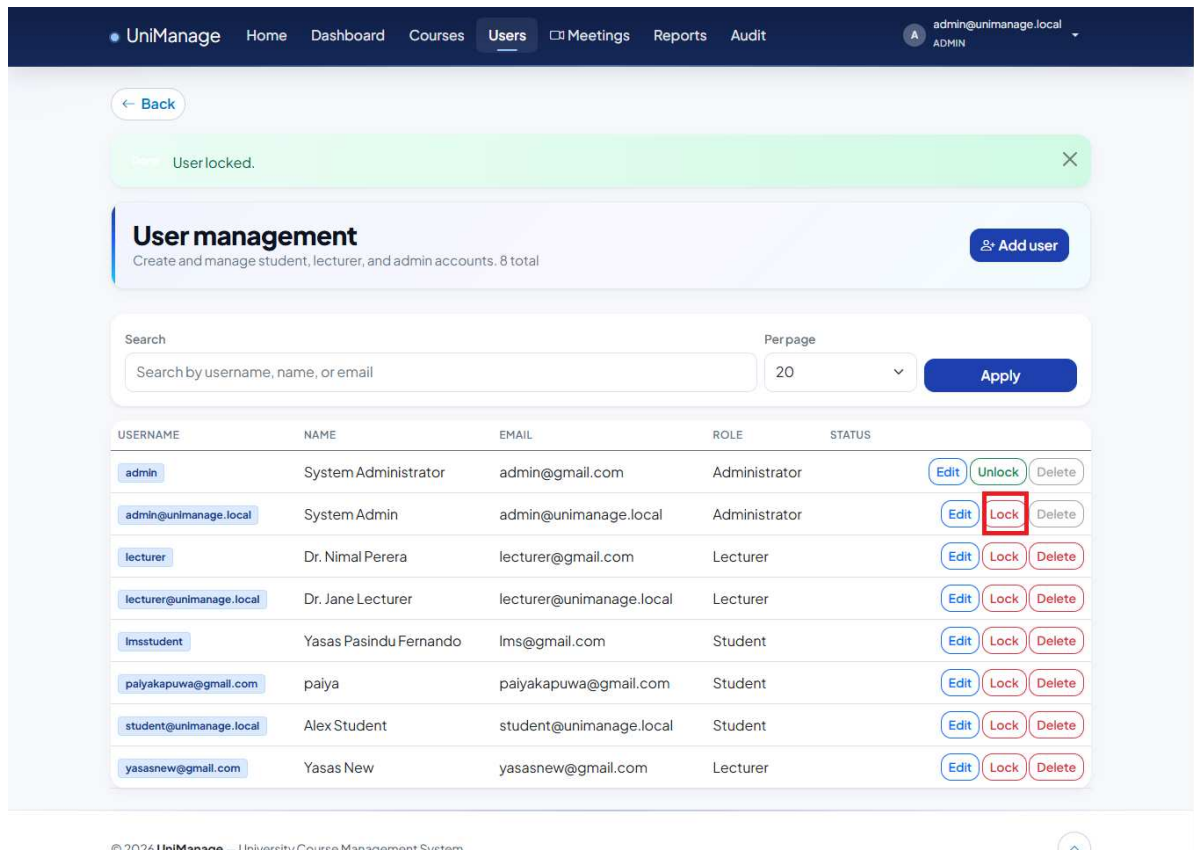


Figure UM9c: User Lock and Unlock Evidence

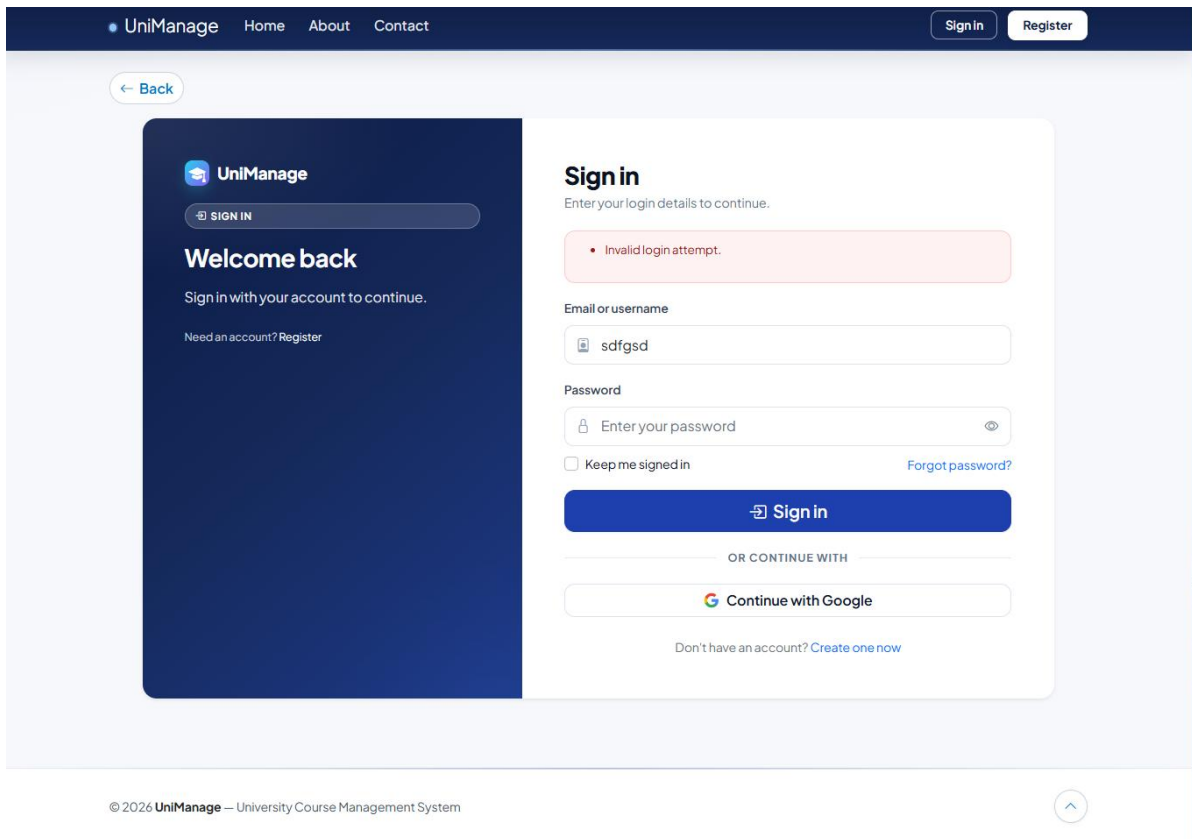
This screenshot shows the lock or unlock confirmation message for a managed account.

10. Validation and Error Messages

The application uses data annotations on view models and additional checks in controller actions. The following validation behaviour is expected during normal operation.

Scenario	Expected Validation Message
Required field missing	The field is required.
Invalid email format	The Email field is not a valid email address.

Scenario	Expected Validation Message
Weak password	The password must be at least 8 characters and include an uppercase letter, a lowercase letter, and a digit.
Confirm password does not match	The new password and confirmation password do not match.
Course code already exists	A course with this code already exists.
Enrolment full	This course has reached its enrolment limit.
Already enrolled	You are already enrolled on this course.
Prerequisite not satisfied	You must complete the prerequisite course before you can enrol.
Wrong file extension	The file type is not allowed.
File too large	The file is larger than the allowed size.
Disallowed message recipient	You are not allowed to message this user.
Invalid grade	The grade must be within the allowed range.
Anti-forgery token missing	The form has expired. Reload the page and try again.
Rate limit hit	Too many requests. Please wait and try again.
Access denied	You do not have permission to view this page.



The screenshot shows the UniManage web application interface. At the top, there is a dark blue navigation bar with links for Home, About, and Contact, along with Sign In and Register buttons. Below the navigation bar, a light blue sidebar on the left contains the UniManage logo, a SIGN IN button, and a 'Welcome back' message with a link to register. The main content area is white and features a 'Sign in' heading with the instruction 'Enter your login details to continue.' Below this, a red error message states 'Invalid login attempt.' The login form includes fields for 'Email or username' (containing 'sdfgsd') and 'Password' (with a toggle for visibility). There are checkboxes for 'Keep me signed in' and a link for 'Forgot password?'. A large blue 'Sign in' button is positioned below the form. At the bottom of the form, there is a section for 'OR CONTINUE WITH' featuring a 'Continue with Google' button and a link to 'Create one now' for users who don't have an account. The footer of the page includes the copyright notice '© 2026 UniManage — University Course Management System' and a small upward arrow icon.

Figure UM17: Validation and Error Handling Evidence

This screenshot shows a form with one or more validation messages displayed.

The friendly error page Views/Shared/Error.cshtml is shown in production for any unhandled exception. The page does not reveal stack traces or internal server information. In development, the developer exception page provides detail to help locate the cause.

11. Reporting and Analytics

The Reports area is available to the Administrator, and selected report views are available to Lecturers with lecturer-scoped data. The following reports are implemented.

Report	Description	Source-Code Evidence
Course Popularity	Lists courses ordered by enrolment count.	ReportsController.CoursePopularity, Views/Reports/CoursePopularity.cshtml
Student Performance	Aggregates grades per student.	ReportsController.StudentPerformance, Views/Reports/StudentPerformance.cshtml
Lecturer Workload	Aggregates assignments and pending submissions per lecturer.	ReportsController.LecturerWorkload, Views/Reports/LecturerWorkload.cshtml
Enrolments	Lists enrolments with optional filters.	ReportsController.Enrollments, Views/Reports/Enrollments.cshtml
Pass and Fail	Groups submissions into pass and fail buckets using a configurable threshold.	ReportsController.PassFail, Views/Reports/PassFail.cshtml
Assignment Attendance	Shows the proportion of enrolled students who submitted each assignment.	ReportsController.AssignmentAttendance, Views/Reports/AssignmentAttendance.cshtml

To export a report, open the report and click the Export CSV or Export PDF button where present. Full export actions are administrator-focused where the controller restricts them to the Administrator role. The CSV file is generated through Infrastructure/CsvWriter.cs and the PDF file is generated through Infrastructure/PdfReport.cs using QuestPDF.

UniManage Home Dashboard Courses Users Meetings **Reports** Audit

admin@unimanager.local
ADMIN

← Back

Reports & analytics

Insights across courses, students, lecturers, and academic outcomes. Every report exports to CSV & PDF.

Course popularity
Enrollment counts and fill rates by course, with chart.

Open CSV PDF

Student performance
Average outcomes for graded submissions, ranked by score.

Open CSV PDF

Pass / fail analysis
Pass and fail counts per course based on a configurable pass mark.

Open CSV PDF

Assignment attendance
Turn-in rate per assignment — submitted vs. enrolled students.

Open CSV PDF

Lecturer workload
Courses, assignments and submissions per lecturer.

Open CSV PDF

Enrollment log
Recent enrollment activity over time.

Open CSV PDF

Audit log
Security and activity trail across the platform.

Open

© 2026 UniManage — University Course Management System

Figure UM14: Reporting and Analytics Screen

This screenshot shows the reports landing page

12. Security Notes for Users

The following security notes apply to all users of UniManage.

The application uses HTTPS by default during local development. If the browser shows a certificate warning, run `dotnet dev-certs https --trust` once to trust the development certificate.

The application uses anti-forgery tokens on every form. Submitted forms must include the token; otherwise the request is rejected.

The application uses rate limiting on authentication and upload endpoints. Submitting too many requests in a short period of time produces a 429 Too Many Requests response.

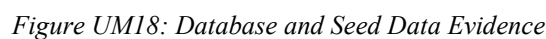
The application uses account lockout. After five failed sign-in attempts, the account is locked for fifteen minutes.

Passwords are not stored in plain text. The Identity framework hashes passwords using PBKDF2 with a per-user salt.

Audit log entries are recorded for security-sensitive actions. The Administrator can review these entries through the Audit Logs page.

Sensitive configuration values such as database, SMTP and Google OAuth values are kept outside the report and shared screenshots. In deployment, these values are supplied through environment variables or server-level configuration rather than being written into the public report.

Personal data shown in screenshots must also be redacted before inclusion in the final submission.



12A. Responsive / Mobile Usage Note

The responsive check supports the interface design and usability evidence, but it is not a full accessibility audit across every device. The screenshots below provide selected mobile viewport evidence.

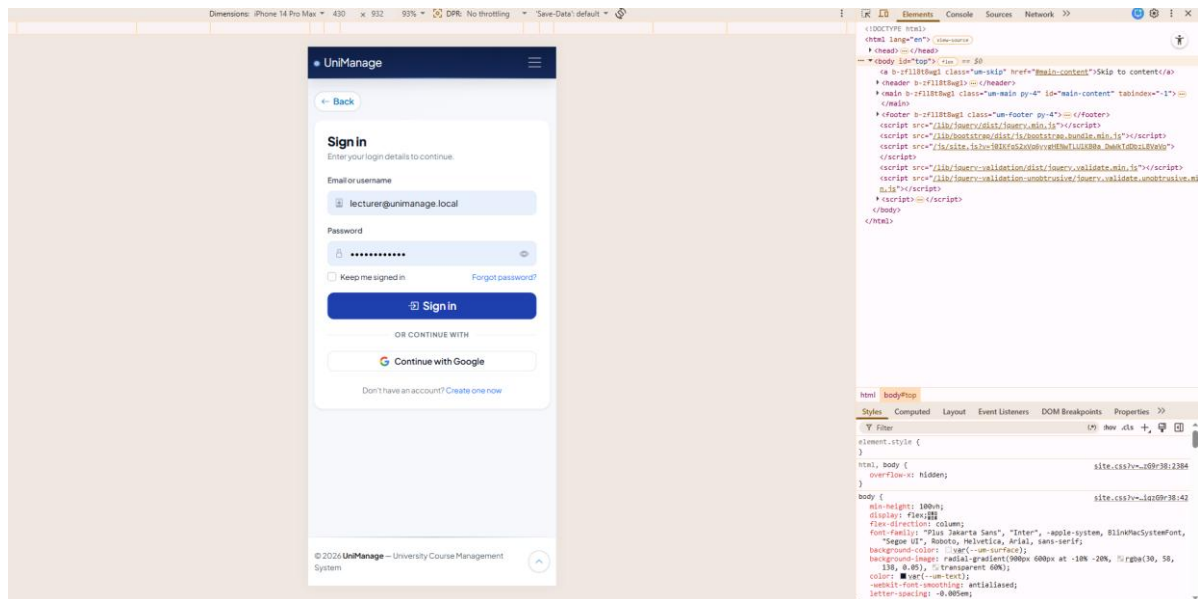


Figure M1: Mobile View of Login Page

This screenshot shows the login page in a mobile browser viewport.

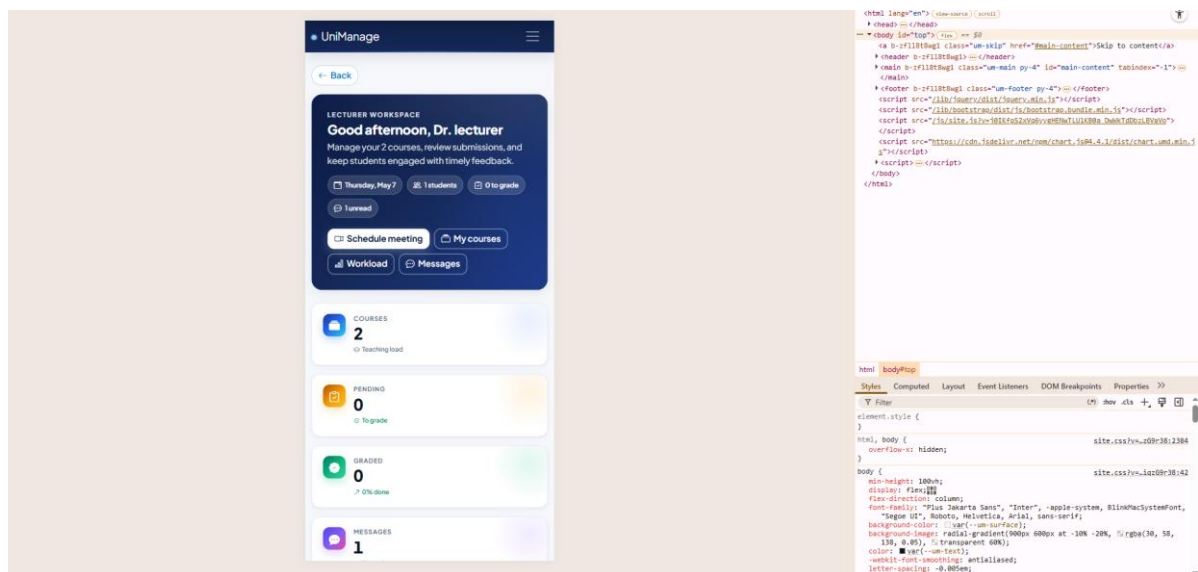


Figure M2: Mobile View of Dashboard

This screenshot shows a role dashboard in a mobile browser viewport.

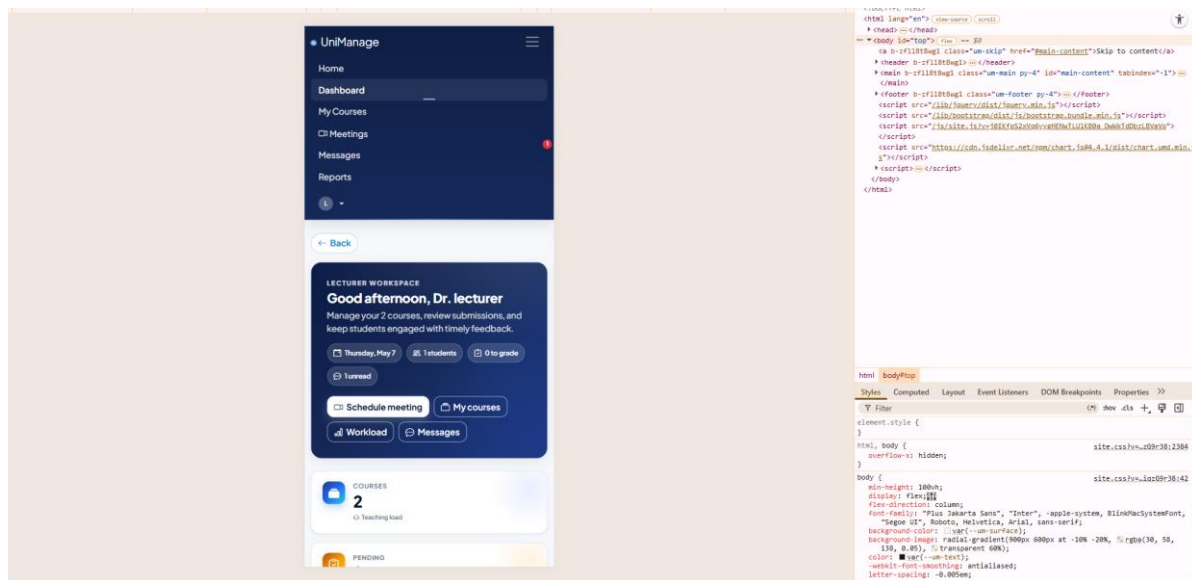


Figure M3: Mobile Responsive Navigation

This screenshot shows the collapsed navigation menu in a mobile browser viewport.

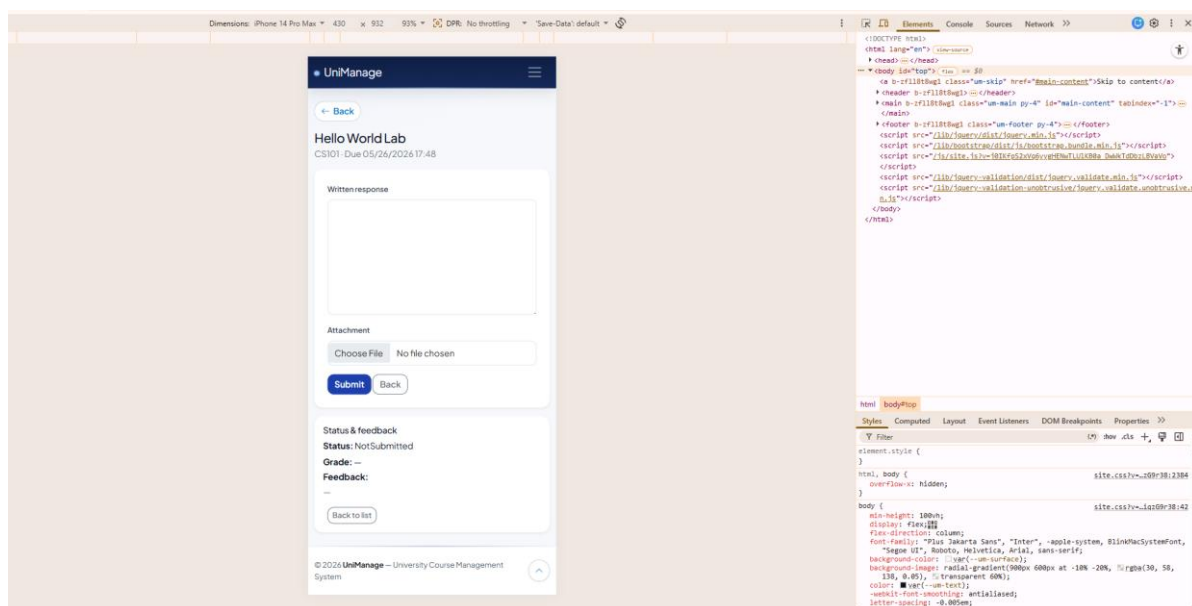


Figure M4: Mobile View of Course or Assignment Form

This screenshot shows an assignment submission form in a mobile browser viewport.

13. Troubleshooting

Problem	Suggested fix
The application does not start	Confirm that the .NET 8 SDK is installed and that the connection string is valid. Inspect the console output for an exception.
The login page rejects valid credentials	Confirm that the seeded users have been created. Check the application logs and re-run the application.
The login page shows a rate-limit error	Wait one minute and try again. The auth policy permits ten requests per minute per IP.
The account is locked	Wait fifteen minutes for the lockout window to expire, or reset the password.
A POST form is rejected	Reload the page to obtain a new anti-forgery token, then submit again.
File upload fails	Confirm that the file extension is allowed and that the file is under the maximum size.
The dashboard is empty	Confirm that the seeded data has been created. The Administrator dashboard counts use the seeded users and any data added during testing.
The friendly error page is shown	Inspect the application logs for the exception detail; the page does not reveal stack traces.
Email is not delivered	Check the Email section in appsettings.json and confirm that the SMTP server is reachable.
Google login is not available	Confirm that the Authentication:Google:ClientId and Authentication:Google:ClientSecret values are populated. If either is empty, Google login is disabled by design.
Migrations fail	Confirm database privileges. Run dotnet ef migrations list to see the current migrations and dotnet ef database update to apply them.
The reports page is empty	Confirm that data has been seeded or added through the application. Empty reports indicate no underlying data, not an error.

14. Notes for Viva Demonstration

The viva demonstration should follow the full script provided in Appendix H of the main technical report. During the demonstration, the author should be ready to show the application structure, successful build output, local or hosted application launch, and the main workflows for Administrator, Lecturer, and Student users.

The demonstration should also include one validation example, one audit log example, and a brief walkthrough of Program.cs to explain Identity, EF Core, rate limiting, anti-forgery protection, security headers, and routing.

The author should refer to the final evidence summary for screenshots, diagrams, database evidence, build output, migration evidence, selected unit-test output, source/build links, and the user manual. Genuine limitations, such as formal accessibility testing and production-level deployment hardening, should be mentioned honestly if asked during the viva..