Student Assignment Brief

This document is intended for Coventry University Group students for their own use in completing their assessed work for this module. It must not be passed to third parties or posted on any website. If you require this document in an alternative format, please contact your Module Leader.

Contents:

The work you submit for this assignment must be your own independent work, or in the case of a group assignment your own groups' work. More information is available in the 'Assignment Task' section of this assignment brief.

Assignment Information

Module Name: Secure Design and Development

Module Code: 7032CEM

Assignment Title: UITS

Assignment Due: 06/08/2025 at 6:00 PM (BST)

Assignment Credit: 15 Credits

Word Count (or equivalent): 2000 words

Assignment Type: CW

Percentage Grade (Applied Core Assessment). You will be provided with an overall grade between 0% and 100%. You have one opportunity to pass the assignment at or above 40%.

Assignment Task

Overview

This assignment requires you to design and develop **strictly** with the aid of a Large language Model (LLM), a functional secure system. The assignment consists of the following parts:

- A functional prototype of a secure online system. This part will be evidenced via the access links to CU GitHub repository for the complete source code and video report showcasing prototype.
- An individual report of 2000 words that describes the design consideration, development and testing of the developed prototype.
- The report **must** contain correct links both to the **github.coventry.ac.uk** repository and **Microsoft OneDrive** with video.
- Any test/login credentials to the prototype should be included in the repository files.

The weighting is 100% including all elements. Practical work, written and video reports will be assessed against the same grading rubric table.

Practical Part Brief

Creative SkillZ LLC hired you as external consultants to help creating their newly proposed "PixelForge Nexus" system.

Your task is to implement a secure online system, tentatively named "PixelForge Nexus," using any language/system of your choice with the aid of an LLM (ChatGPT or Google Gemini). The prototype should contain the following functionality:

Core Functionality

1. **Project Management:**
    a. **Add/Remove Projects:** Admins can add new game projects (with name, description, and an initial deadline). Admins can also mark projects as "Completed"
    b. **View Projects:** All users can view a list of active projects.
2. **Team Assignment:**
    a. **Assign Team Members:** Project Leads can assign developers to their specific projects.
    b. **View Assigned Projects:** Developers can see a list of projects they are currently assigned to.
3. **Basic Asset & Resource Management:**
    a. **Upload Project Documents:** Admins and Project Leads can upload general project documents (e.g., design docs, meeting notes) associated with a project. *We'll skip version control for assets to simplify, but you can do it for 80%+.*
    b. **View Documents:** All users assigned to a project can view its uploaded documents.

Privilege Separation

- **Admin:** Can add/remove projects, manage all user accounts (create, edit roles), and upload documents for any project.
- **Project Lead:** Can assign developers to their projects and upload documents for their projects.
- **Developer:** Can view projects they are assigned to and access associated project documents.

Login Security

- **Robust Login System:** Essential for secure password hashing and storage (e.g., using bcrypt).
- **MFA Implementation: (**Optional but highly recommended): Adding Multi-Factor Authentication would significantly boost security.

Proposed Pages

- **Sign In/Register:**
    - Allows existing team members to log in.
    - Admin-only functionality to register new team members (no self-registration for simplification).
- **User Dashboard (Single Dashboard for All Roles):**
    - Developers: See a list of their assigned projects with links to documents.
    - Project Leads: See projects they lead, with options to assign team members and upload documents.
    - Admins: Access to add/mark projects as complete, and manage user accounts (add/edit roles).
- **Account Settings:**
    - Users can update their password.
    - MFA setup (if implemented).
- **Project Details Page:**
    - Displays project name, description, deadline, assigned team members, and uploaded documents.

Practical Element Information

During the practical/lab sessions, you will be required to complete a set of workshops that will help you to familiarise yourself with the tools and practices needed for the completion of the practical element, which covers all the learning outcomes. Your work will be marked using the provided grading rubric.

Evidence for Practical Part / Overview of the marking rubric

1. System Design (35%):
    a. Produce a design for the system to be implemented and explain what design and security principles have been considered and why.
    b. Describe how the chosen principles or features enhance the functioning and security of that system in this stage of the development life-cycle.
2. Security testing and analysis (35%):
    a. Critically evaluate the application of security techniques and propose the possible solutions for the issues discovered.
    b. Propose solutions for the issues discovered during the testing and analysis process showing how they can mitigate the detected problems.
3. System Development (20%):
    a. Develop and demonstrate functional prototype that complies with the design provided above, including legal and ethical context of the development.
    b. Demonstrate the proper functioning and security mechanisms of the developed system in accordance with the existing secure development standards and methodologies.
4. Formal Methods (10%):
    a. Application of formal methods, to produce the behavioural model of the system that will be based on the design or development stage of the system life-cycle.
    b. Verify the correctness of the system with respect to its specification using the appropriate verification techniques and tools.

Submission Instructions

Submission

Deadline is **06/08/2025 at 6:00 PM (BST).** Note that you should submit your own work.

For your submission, you will create and submit a written report (.docx file) that contains 2 links at the top:

- 1 link will be to your **Coventry University GitHub coursework repository** that is part of the **7032CEM-2526MAYSEP organisation**.
- 1 link will be to your **Microsoft OneDrive online hosted video report.**
- The rest of the document will be the **Individual Report Brief**.

You will upload this .**docx file to the Aula submission point** for this assignment

About GitHub

You must use the **Coventry University GitHub** (https://github.coventry.ac.uk/) with your Git Repository placed into the **7032CEM-2526MAYSEP organisation** (https://github.coventry.ac.uk/7032CEM-2526MAYSEP). You will include a link to the Git Repository in your Aula submission in the .docx file.

- Your Coventry University GitHub Repository should be named in the format of **"7032CEM-2526MAYSEP_Your Student ID"**.
    - for example, if my student ID was 12345, the repository name would be 7032CEM-2526MAYSEP_12345.
- Your repository must be **private**.
- The module staff must be added to your repository as **collaborators** to **allow them access.** (Mia Mohac – ad9235, Antal Goldschmidt – ab2216)
- **If you do not use the 7032CEM-2526MAYSEP organisation and/or do not add staff as collaborators, you may receive a mark of 0.**

You are expected to make **appropriate use of version control (CU GitHub)** to manage and record the development **of your solution**. You are expected to use this repository through the assignment and make regular commits.

Both, the CU GitHub (github.coventry.ac.uk) commits, and work on your prototype MUST be completed before the submission deadline. **If a commit is made after the deadline or if module staff cannot access the repository, this may result in 0 marks being awarded. It is your responsibility to make sure your repository is viewable by the markers.**

The complete source code and related documentation MUST be supplied. The source code should be appropriately and correctly commented. You should also identify any assumptions that you have made.

Your work will be marked using the Aula grading rubric.

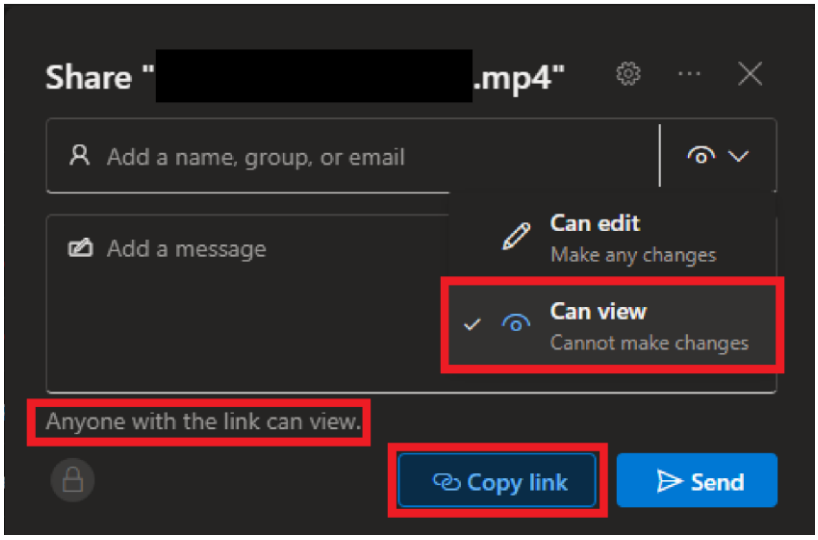
About Video Report

You will also record a video report (a singular video, totalling **5 minutes or less**) which shows:

- A fly-through of the output of the coursework, highlighting the significant aspects of the work (see the marking rubric below for the aspects that should be covered).
- You are required to use voice-overs and/or text overlays to explain what is happening in the video.

The video should be uploaded to **Microsoft OneDrive** (provided by the university), and you should copy a link to the video which allows people to View the video (see figure below). You will include a link to the Microsoft OneDrive online hosted video in your Aula submission in the .docx file.

Figure 1
The sharing options that should be chosen to allow the markers to view the Microsoft OneDrive hosted video with the URL link



If you are uncertain about how to do this, contact a member of the module team **well before the submission date**. It is up to students to test this uploaded video so that it is viewable.

If the video cannot be accessed by the module team in any way (incorrect URL, privacy settings of the video etc.), this may result in a reduction in your mark.

About Individual Report Brief

At the top of the brief after including two links (for video recording and GitHub repository), you must include your full name and Student ID.

The following issues, in relation to the prototype developed in the practical element of the assignment, must be included in system documentation:

a) Explanation of the methods and techniques followed in the practical task for the development of the system
b) Discussion of all the stages of the development life-cycle (e.g., specification, design, development, etc.)

Individual Report Information

At the end of the module, you will be expected to submit an individual report. This must be entirely your own work. The report is based on the prototype that you have completed and should report on the development process and the evaluation of the prototype system. You should also identify any assumptions that you have made.

This part is **not assessed separately** from the practical part of the assessment. Therefore, your report will be marked using the **same grading rubric as that of the practical part**.

Evidence (Indicative) for the report

1. The **access links** to:
    a. CU GitHub repository
    b. Microsoft OneDrive video
2. **Full name and Student ID**
3. **Explanation** of the methods and techniques followed
4. **The stages** of the life-cycle that have been followed for the development of the system
    a. Describe the design and development of the system.
    b. Explain the deployment and testing approach, limitations of the prototype and propose further improvements.
    c. Explanation of the methods and security techniques followed in the practical task
    d. Formal methods application – the model you followed (e.g. a user login, verification, etc.)
5. **Required appendix**:
    a. LLM prompt history; listing
    b. Any other resources used (APA style referencing!)

Submission

Submission arrangement online via Aula**: Yes**

File types and method of recording:

- **MS Word (.docx) for report**
    - CU GitHub repository **link** of practical development (**github.coventry.ac.uk**)
    - Video uploaded to **Microsoft OneDrive link**

Assignment/Coursework Resit

The three parts of the assignment described above are **NOT** assessed separately. To pass the module, the grade for the assignment needs to be 40% or above. If your grade falls below this, then you need to resit the coursework. If you defer the assignment, you will be required to accomplish the resit assignment and you will **not** be allowed to resubmit your original work.

Marking and Feedback

How will my assignment be marked?

Your assignment will be marked the module team.

How will I receive my grades and feedback?

Provisional marks will be released once internally moderated.

Feedback will be provided by the module team alongside grades release.

You will access the feedback via Aula on the Turnitin platform.

Your provisional marks and feedback should be available within 2 weeks.

What will I be marked against?

Details of the marking criteria for this task can be found at the bottom of this assignment brief.

Assessed Module Learning Outcomes

The Learning Outcomes for this module align to the marking criteria which can be found at the end of this brief. Ensure you understand the marking criteria to ensure successful achievement of the assessment task. The following module learning outcomes are assessed in this task:

- **LO1**. Evaluate a range of platforms for systems and applications, against standards and methodologies for secure development, incorporating issues raised by the legal and ethical context of the development, such as IP law, privacy and data-protection.
- **LO2.** Design applications that adhere to secure development methodologies.
- **LO3.** Develop applications that implement secure principles and are fully tested against software security and quality guidelines.
- **LO4**. critically evaluate existing software, using methods such as code review, static and dynamic analyses.
- **LO5.** Apply formal methods to different stages of the system development life cycle like system specification, design, development and testing.

Assignment Support and Academic Integrity

If you have any questions about this assignment please see the Student Guidance on Coursework for more information.

Spelling, Punctuation, and Grammar:

You are expected to use effective, accurate, and appropriate language within this assessment task.

Academic Integrity:

The work you submit must be your own, or in the case of groupwork, that of your group. All sources of information need to be acknowledged and attributed; therefore, you must provide references for all

sources of information and acknowledge any tools used in the production of your work, including Artificial Intelligence (AI). We use detection software and make routine checks for evidence of academic misconduct.

Definitions of academic misconduct, including plagiarism, self-plagiarism, and collusion can be found on the Student Portal. All cases of suspected academic misconduct are referred for investigation, the outcomes of which can have profound consequences to your studies. For more information on academic integrity please visit the Academic and Research Integrity section of the Student Portal.

Support for Students with Disabilities or Additional Needs:

If you have a disability, long-term health condition, specific learning difference, mental health diagnosis or symptoms and have discussed your support needs with health and wellbeing you may be able to access support that will help with your studies.

If you feel you may benefit from additional support, but have not disclosed a disability to the University, or have disclosed but are yet to discuss your support needs it is important to let us know so we can provide the right support for your circumstances. Visit the Student Portal to find out more.

Unable to Submit on Time?

The University wants you to do your best. However, we know that sometimes events happen which mean that you cannot submit your assessment by the deadline or sit a scheduled exam. If you think this might be the case, guidance on understanding what counts as an extenuating circumstance, and how to apply is available on the Student Portal.

Administration of Assessment

Module Leader Name: Mia Mohac

Module Leader Email: ad9235@coventry.ac.uk

Attempt Type: MAINSIT

Component Code: CW

Assessment Marking Criteria

	System Design Weighting**: 35%** **(also Demonstrated in Video)**	**Security Testing and Analysis** Weighting: **35%** **(also Demonstrated in Video)**	**System Development** Weighting: **20%**	**Formal Methods** Weighting: **10%**
80 to 100%	Marks above 80 will be awarded for going above and beyond the requirements for a distinction and demonstrate and innovative approach and insight. A system design of professional standards, which is optimized for the given scenario. The system design demonstrates a deep understanding of secure design principles. The system design includes a comprehensive threat model that identifies and prioritizes potential security risks. The system design includes, clear access control mechanisms, comprehensive data encryption strategies. Detailed documentation illustrates how security principles are applied throughout the system design	Marks above 80 will be awarded for going above and beyond the requirements for a distinction and demonstrate and innovative approach and insight. Professional standard security analysis and testing followed by security measures that significantly improve the overall system security and functioning. The entire security process is comprehensively documented. The report includes comprehensive test cases and results, including code scanning reports	Marks above 80 will be awarded for going above and beyond the requirements for a distinction and demonstrate and innovative approach and insight. System development is professionally executed, fully complying with all the respective methods and standards. The codebase displays exceptional code quality and adherence to secure coding practices. The documentation of the process is thorough, including detailed explanations of secure coding practices.	Marks above 80 will be awarded for going above and beyond the requirements for a distinction and demonstrate and innovative approach and insight. Formal modelling and verification processes are professionally applied to the system covering and examining all the potential issues. The report demonstrates a deep understanding of formal methods, with minimal errors in their application. The process is fully documented.
70 to 79%	Detailed system design fully based on the secure design	An extensive security analysis and testing has been conducted	System development fully complies with the secure	Formal modelling and verification processes fully examine the

	methodologies and standards. Thorough and well-justified documentation. The system design is well structured and display a clear understanding of security principles and a fully detailed threat model. Most security controls are appropriately integrated into the design, with effective strategies for reducing threat.	providing mitigation techniques for the identified security issues. The testing process covers key aspects of security. Test cases and results provide a detailed overview of the security testing process. Thorough and well-justified Documentation	development methods and standards. The codebase demonstrates a high degree of adherence to secure coding practices, with minimal security vulnerabilities. Thorough and well-justified documentation.	functioning and security issues of the system Formal methods are effectively used to analyse and address security concerns, with minimal errors. The formal analysis is very well justified in the report.
60 to 69%	System design incorporates an extensive range of principles which are very well discussed in the report. The system design shows awareness of security concerns but may lack detail in threat modelling. Most security controls are present but need further clarification.	A medium range of security analysis and testing techniques have been used providing effective solution for the detected problems. Process is very well described in the report.	System development fully complies with the proposed design, but partially with the secure development methods and standards. Process is very well described in the report.	Formal modelling and verification processes extensively examine a wide range of functioning and security issues. The formal analysis is very well described in the report
50 to 59%	System design incorporates the principles required for the proper functioning and security of the system but required more details. Adequate discussion in the report.	A small range of security analysis and testing techniques have been used providing the respective solutions. Test cases and results are provided but may lack completeness. Adequate discussion in the report	System development considers basic functioning and limited security of the system with good explanation of the development stage. The system development phase has basic security measures in place, but there are notable areas where secure coding practices could be improved.	Behavioural model presents the basic functioning of the system incorporating security aspects as well. Verification examines the basic security issues. Adequate discussion in the report.

40 to 49%	System design meets the basic requirements with lack of details. Short discussion in the report. Security controls are present, but it is generic or incomplete.	Very limited security analysis and testing of the system with very few solutions provided. Test cases and results are limited in scope. Short discussion in the report.	System development complies with the requirement of a basic design. Short discussion in the report.	A very basic attempt of modelling the behaviour of the system and then verify it. Short discussion in the report.
Fail 30, 35%	No system design built or incomplete design provided. which is severely deficient in security aspects	There is no evidence of security testing or analysis provided or very poor system security and testing carried out with no recommended solutions.	No system developed or incomplete implementation of the system. No attention to system security	No or very poor application of formal methods to the system analysis.
Fail 0 to 29%	The system design is entirely devoid of security considerations and outcome not met or no system design built	No attempt or no system security testing carried .	Outcome not met or no system developed	No attempt or no application of formal methods to the system analysis.