

Coursework Overview and Assessment Criteria

Module Title:	Cloud Native Development
Module Code:	COM682 (UU, SUST, QAHE)
Module Coordinator:	Zeeshan Tariq-UU/QAHE, Muhammad Shafi-SUST
Teaching Staff Responsible:	Zeeshan(UU), Shafi Muhammad(SUST), Rashid Kamal (UU), Ibrahim (QAHE)
Semester (s) Taught:	One
Course / Year Group:	UU, SUST, QAHE
Coursework / Exam Weighting:	100/0

Coursework Assessment Overview

This module has two points of assessment:

- 1) Written Assignment (50%) submitted mid-semester.
- 2) Project (50%) submitted at the end of the semester.

Feedback for both components of the assessment will be provided as specified by the university's assessment code of practice (currently working 20 days).

The university has a number of rules and regulations surrounding assessment, late submissions, and illness. These are in the student guide [1] - ensure you read this and understand the impact of these rules and regulations.

These coursework assignments are detailed below.

Coursework 1 – Written Assignment [50%]

Released: Week-1

Submission Deadline: 10th November, 2025

Feedback Date: 8th December, 2025

- Students will individually **design** a cloud-native solution in the form of a web app.
- The designed solution should leverage a range of cloud-native technologies and concepts as taught within the module and module materials.

Related Learning Outcomes:

1. Assess the concepts behind a range of cloud native development techniques and critically evaluate when to apply these paradigms to realisation of solutions.
2. Demonstrate a comprehensive understanding of modern cloud development, techniques and practice and how it may be leveraged to address related challenges.

3. Comprehend administrative aspects related to cloud native development such as pricing concerns and access control.
4. Autonomously and independently identify deficiencies when interacting with a range of architectures and deployment paradigms, leveraging knowledge of these deficiencies to improve future practice.

Solution Designing:

Design a scalable, cloud native, web-application which acts as a **multimedia sharing platform** facilitating sharing of multimedia content. The exact media type(s) which can share (from what's covered in the labs) are at the discretion of the student. The ability to share multiple content types is encouraged.

The **designed** solution should leverage a range of cloud native technologies and concepts as taught within the module and module materials. You may choose to design a **serverless architecture using Azure Functions** as the core compute instead of VMs/App Service.

Your design should include the following, but not limited to:

- Design a multimedia hosting webpage as a Cloud service where the user can upload multimedia files.
- You should choose to create page wireframes. Present this in the form of website design layout diagram. (Covered during week-5 lecture)
- Design an architecture of your solution consisting of VMs, DBs, Web servers, etc. Architecture is to be presented in the form of network diagrams, as mentioned during Week-4 lecture.
- Design a Database schema to store the user entries. You need to present it in the form of list of DB resources. You should also present in the form of ERD diagram. Non-relational databases may be modelled as commented JSON or a table as shown in week 3.
- Design the REST API to support the creation, retrieval, updating, and deletion of various asset records using the Logic Apps. This design needs to be presented in the form of UML diagram, as mentioned during Week-5 lecture.

Ideally, your designed solution would integrate the following, but not limited to:

- Static HTML hosting of the content of a web page that interacts with a web backed through REST calls hosted at App Services, covered in lab 5.
- Hosting of REST endpoints (URLs) which provides service logic and connections to all necessary elements such as storage.
- Use of Databases for hosting SQL and NoSQL structures.

Note: Marks for design or UX are only covered in the Rubric section 'Overview of the designed technical solution'.

Submission:

Students will submit a slide deck which details the designed solution (details below). Students must follow the below **content outline** for slides:

- Title Slide: Project name, one line description. Student name, student number.
- Discussion of the technical problem and identification of the issues related to scalability for the resources used in your project. Note, this is not a description of the objective of the project, instead it is an appraisal of the technical issues which may affect traditional solutions and how cloud based systems address these.
- Solution architecture of the project (Only use resources taught in the module)
- An overview of advanced features that you intend to develop in the final solution. Note these are use of advanced platform features, not additional buttons etc.
- An assessment of the limitations of the solution
- Assessment of Scalability of your designed solution
- Concluding comments, such as how this solution can be improved, what are design limitations which can be improved? How solution can be useful for commercial purpose? Is there any ethical comments?
- References
- Submission Checklist: Appendix III.

Please submit this checklist at the end of the PowerPoint slide deck. Checklist slide will not be counted in the 15 slides limit.

Slides should be produced in the ***PowerPoint format*** and will need to be uploaded to the relevant assessment area on Blackboard. Your presentation should not exceed more than 15 slides. Slide notes and slides after slide 15, will not be assessed.

<>>

Coursework 2 – Project [50%]

Released: Week-1

Submission Deadline: 9th January, 2026

Feedback Date: 30th January, 2026

Related Learning Outcomes:

1. Demonstrate a comprehensive understanding of modern cloud development, techniques and practice and how it may be leveraged to address related challenges.
2. Comprehend administrative aspects related to cloud native development such as pricing concerns and access control.
3. Autonomously and independently identify deficiencies when interacting with a range of architectures and deployment paradigms, leveraging knowledge of these deficiencies to improve future practice.

Implementation of design:

Implement, deploy, and test the solution designed in CW1. This should be implemented and deployed using the Microsoft Azure cloud platform taught and used within the practical exercises associated with this module.

The **developed** solution should leverage a range of cloud native technologies and concepts as taught within the module and module materials. These include the following, but not limited to:

- Implement the solution based on the design produced in Part-1
- Use scalable cloud native storage for storing binary blob data, such as files
- Leverage a cloud native NoSQL database to store metadata
- Design the REST API to support the creation, retrieval, updating, and deletion of various asset records using the Logic Apps
- Create endpoint URLs that will store images and metadata.
- Implement CI/CD using Git.
- Add advanced services, such as Azure App Monitor, App Insights, etc that are covered in labs.

Submission:

Each student will individually submit 5 mins video of the built solution:

- A video where the student provides 5-minute maximum walk-through of the developed, tested and deployed solution.
 - This video should be used to showcase the functionality of the solution and show its deployment to Azure.
 - You should present all of the used Azure resources.
 - Do not run through slides of Part-1 during the video.
 - If there is no certain excuse, the video should include you (turn camera on, if possible).
- <>>
- You should present following, but not limited to:
 - Your running application, where you would perform CRUD operation using RESTful APIs. Uploading, editing, and deleting functions should be working to get maximum marks. You will still get marks if the solution is partially completed.
 - Your backend Azure resources, logic apps, functions, DB, CI/CD functionality.
 - URLs referring to the Azure resources which are used for CRUD.
 - Clearly mention which advanced features you have used, and show their working. Also mention if solution has no advanced features added.
- Please use **Panopto Capture** to record your video screencast - accessed by clicking the '**Panopto**' link from the menu on the left of the Blackboard site. While **Panopto Capture** is preferred, several free tools are available to support the dual recording feature – computer screen and presenter – (e.g., <https://screencast-o-matic.com/>).
- Please refer to the University's Panopto support pages for information on Using Panopto Capture and Submitting to a Panopto Student Video Assignment.

Related Learning Outcomes:

1. Assess the concepts behind a range of cloud native development techniques and critically evaluate when to apply these paradigms to realisation of solutions.
2. Demonstrate a comprehensive understanding of modern cloud development, techniques and practice and how it may be leveraged to address related challenges.

3. Comprehend administrative aspects related to cloud native development such as pricing concerns and access control.
4. Autonomously and independently identify deficiencies when interacting with a range of architectures and deployment paradigms, leveraging knowledge of these deficiencies to improve future practice.

Time Penalties: These deductions are from rubric section 'Video Quality and Presentation'

Within 5 mins – No penalty

> 30sec and < 1min – 10% deduction

>1min and above – 20% deduction

Plagiarism

N.B. Students should be aware of the plagiarism policy of the University and submit their coursework in accordance with this. Plagiarism is the unattributed copying of the work of another person, either from a published work or the work of another student. It is a form of literary theft and is not permitted under any circumstances. Plagiarism is regarded by the University as a very serious offence and subject to formal disciplinary proceedings. See [5] for more details.

I declare that this is all my own work. Any material I have referred to has been accurately referenced and any contribution of Artificial Intelligence technology has been fully acknowledged. I understand the importance of academic integrity and have read and understood the University's General Regulation: Student Academic Integrity and the Academic Misconduct Procedure. I understand that I must not upload my work before, during or after submission to any unapproved plagiarism detectors or answer sharing platforms, or equivalent, and that only University-approved platforms should be used.

The assessment criteria for coursework 2 and rubric is presented as an appendix to this document.

References

- [1] "Ulster University Student Guide." [Online]. Available: <https://www.ulster.ac.uk/connect/guide>.
- [2] IEEE, "Manuscript Templates for Conference Proceedings." [Online]. Available: https://www.ieee.org/conferences_events/conferences/publishing/templates.html.
- [3] IEEE, "IEEE Citation Reference." [Online]. Available: <https://www.ieee.org/documents/ieeecitationref.pdf>.
- [4] Mendeley Ltd, "Mendeley Citation Manager." [Online]. Available: <https://www.mendeley.com/>.
- [5] https://www.ulster.ac.uk/_data/assets/pdf_file/0019/152218/Plagiarism-Policy.pdf.pdf

Appendix I – assessment criteria coursework 1

	0%-39% (Fail)	40%-49% (3rd)	50%-59% (2.2)	60%-69% (2.1)	70%-79% (1st)	80%-100% (High 1st)	Total CW1
	Poor, Insufficient, Incorrect	Basic, Weak, Flawed	Limited, Inconsistent, Unmanaged	Good, Consistent, Controlled	Excellent, Comprehensive, Reflective	Outstanding, Insightful, Professional	
Problem Definition and Discussion	Little description of the overall problem was provided; poor justification of why a cloud model needs to be adopted.	There was insufficient explanation of the issue, as well as inadequate justification for why a cloud-based model should be implemented.	Moderate description of the overall problem was provided, adequate justification presented of why a cloud solution needs to be developed.	A sufficient amount of justification was provided as to why a cloud-based solution needs to be developed, and a good description of the problem as a whole was provided.	Good description of the overall problem with a good justification of why a cloud solution needs to be developed.	Excellent description of the issue as a whole, along with a full and comprehensive justification of why a solution based on the cloud ought to be developed.	20%
	Limited critical appraisal of the use of cloud technologies.	A constrained and selective analysis of the use of cloud computing technologies.	Adequate critical appraisal of the use of cloud technologies and related patterns.	Appropriate and critical evaluation of how cloud technologies and related patterns are being used in the solution.	Strong critical appraisal of the use of cloud technologies, related patterns, and architectural components.	An in-depth analysis and evaluation of how cloud computing, associated patterns, and architectural components are being used.	
Overview of the designed technical solution	No justification for the choice of technology applied to the problem was minimal.	Justification for the choice of technology applied to the problem was minimal.	The technology used to produce the solution was appropriate given the development problem.	Given the development issue, the technology used to produce the solution was appropriate, and sufficient details were provided.	The technology used to produce the solution was carefully examined and logically chosen – given the development problem.	The technology that was used to produce the answer was thoroughly investigated, analysed, and selected in a logical manner, along with the reasoning behind the selection.	40%
	No design presented	Design was poorly informed and did not incorporate many clouds native elements. No wireframes were produced	Moderate effort was made to incorporate cloud native components. The design was satisfactorily informed by cloud native design patterns.	Cloud native components were incorporated with an appropriate amount of effort. Cloud native design patterns successfully informed the design.	Alternative technologies were examined and excluded accordingly. A wide range of cloud native components were incorporated into the solution.	Alternative technologies were investigated, detailed with pro and cons. The solution included a wide range of cloud native components with architecture diagram.	
	No meaningful solution architecture was presented.	A moderate solution architecture was presented.	An architectural diagram of the developed solution was presented. Wireframes were produced	An architectural diagram of the developed solution was presented with detailed cloud components. Wireframes were produced	The solution architecture was documented well incorporating control flows and software architecture diagrams. The design of the solution was considered and	Insights for selection of the solution and its advantages over other cloud-based solutions available in the literature. Wireframes were produced	

					justified through cloud native design patterns. Wireframes were produced		
Assessment of Limitations	The limitations of the solution were discussed without any reflection	The limitations of the solution were discussed with some awareness of how to remedy these presented.	"-3 limitations of the solution were discussed with some awareness of how to remedy these presented.	A broad appraisal of the limitations of the solution were presented. Strategies to address these were presented.	A comprehensive analysis of the constraints imposed by the solution was provided here. Presented here are some potential solutions to these problems.	15%	
Assessment of Scalability	The solution integrates no elements to offer scalable operation.	Scalability was partially catered for in the solution.	Scalability was catered for most of the resources in the solution.	Scalability was well catered for with multiple cloud native elements applied to achieve this.	Scalability was effectively addressed by employing a number of cloud-native components throughout the development process.	15%	
Concluding comments	No reflection was applied to the solution, its functionality, limitations and potential applicability.	The proposed solution, its functionality, its limitations, and its potential applicability were all given some reflections	Meaningful reflection was applied to the solution, its functionality, limitations, and potential applicability.	Variety of meaningful reflection was applied to the solution, its functionality, limitations, and potential applicability.	Insightful reflection was applied to the solution, its functionality, limitations and potential applicability. Weaknesses were identified and improvements were suggested.	The solution, its functionality, its limitations, and its potential applicability were all given careful consideration in this in-depth analysis. It was determined what the problems were, and some potential solutions were proposed.	5%
Referencing	No referencing.	Inadequate or incorrect referencing.	Only few references provide which doesn't relate to the solution	Relatable references provided	Correct and appropriate referencing.	Correct and appropriate referencing and within document citation.	5%

Appendix II – assessment criteria coursework 2

	<i>0%-39% (Fail)</i>	<i>40%-49% (3rd)</i>	<i>50%-59% (2.2)</i>	<i>60%-69% (2.1)</i>	<i>70%-79% (1st)</i>	<i>80%-100% (High 1st)</i>	<i>Total CW2</i>
	<i>Poor, Insufficient, Incorrect</i>	<i>Basic, Weak, Flawed</i>	<i>Limited, Inconsistent, Unmanaged</i>	<i>Good, Consistent, Controlled</i>	<i>Excellent, Comprehensive, Reflective</i>	<i>Outstanding, Insightful, Professional</i>	
Implementation	Incomplete or incorrect implementation. Key components are missing, non-functional, or improperly deployed.	Basic implementation with significant issues or missing components.	Adequate implementation with most components functional, though minor errors may exist.	Comprehensive implementation with all major components functional and well-integrated.	Detailed and highly effective implementation with all components seamlessly integrated and functional.	Exceptional and innovative implementation with all components expertly integrated and functional.	35%
Use of Azure Resources	Poor or incorrect use of Azure resources. Key services are missing or non-functional.	Some Azure resources used, but implementation is flawed or incomplete.	Adequate use of Azure resources with most components correctly deployed and functional.	Effective use of Azure resources with all components correctly deployed and well-integrated.	Excellent use of Azure resources with all components flawlessly deployed and integrated.	Masterful use of Azure resources with flawless deployment and integration.	35%
Use of Advanced Features	Little to no attempt to integrate advanced features. Features are non-functional or incorrectly implemented.	Minimal integration of advanced features, with limited functionality.	Some integration of advanced features, with basic functionality.	Effective integration of advanced features with good functionality.	Advanced features are expertly integrated and fully functional, significantly enhancing the solution.	Advanced features integrated at an expert level, adding significant value.	20%
Video Quality and Presentation	Poorly structured video, unclear, lacks key elements. May not appear in video.	Somewhat clear video but may be poorly structured or lack essential content.	Clear video that presents the solution adequately but may lack depth.	Well-structured and clear video, presenting the solution effectively.	Highly professional video, clearly structured, within the time limit.	Exemplary video with clear, well-structured presentation, highly professional.	10%

Appendix III. Coursework 1 Submission Checklist

Before submitting, make sure you have addressed the following:

General

- Did you include a **title slide** with project name, one-line description, your name, and student number?
- Is your submission in **PowerPoint format** and within **15 slides** (excluding references)?

Problem & Context

- Did you **discuss the problem** you are solving?
- Did you **identify issues related to scalability** of resources in your project?

Solution Design

- Did you include a **solution architecture diagram** using only **Azure resources**?
- Did you design and present a **multimedia hosting webpage layout** (wireframe or design diagram)?
- Did you include a **network/architecture diagram** showing compute, storage, networking, etc.?
- Did you design a **database schema** (list of resources and/or ERD diagram) for storing user entries?
- Did you design the **REST API** for CRUD operations and present it in a **UML diagram**?
- Did you integrate (or at least propose integration of):
 - Static HTML hosting of a webpage frontend?
 - REST endpoints for service logic and storage connections?
 - Use of both **SQL** and **NoSQL** database structures?

Critical Evaluation

- Did you provide an **overview of advanced features** you intend to develop in the final solution?
- Did you include an **assessment of limitations** of your designed solution?
- Did you include an **assessment of scalability** (horizontal/vertical scaling, autoscaling, etc.)?
- Did you justify **why you selected specific Azure services** over alternatives?

Conclusion & References

- Did you include **concluding comments** summarising your design?
- Did you provide **references** to sources, frameworks, or Azure documentation you used?

If you can tick all these boxes, your submission is likely to meet the expectations.

Please submit this checklist at the end of the PowerPoint slide deck. Checklist slide will not be counted in the 15 slides limit.