Diploma in Information Technology
CIT2C24 – Cloud Application Development
Project Specification
AY2025/2026 April Semester

Introduction

The project consists of three parts and involves developing a cloud-based web application using services from Amazon Web Services (AWS).

1. Part 1 is to produce a proposal for your cloud-based web application and develop a low-fidelity prototype.
2. Part 2 is to implement at least 1 microservice that includes the HTTP GET, POST, PUT, DELETE functions/API that links to the manipulation of database with inclusion of business logic when applicable.
3. Part 3 is to develop a cloud-based web application that consumes the microservices created in Part 2.

Part 1 – Project Proposal (20%)

You are required to submit a proposal for the cloud-based web application that you will develop in Part 2 and 3. There is no restriction to the nature of the application, but it must fulfil the project requirements stated in Part 2 and 3.

The proposal (refer to LMS for template) must include the following minimum requirements:

Application Description	State the name of your application and give a brief description of the proposed functionalities, target audience and their intended controls. Minimum of 4 functionalities. Ensure that all HTTP methods are covered, that is, at least one GET, POST, PUT and DELETE function.
Microservices Architecture Design	Provide a diagram to describe the microservices architecture design of your cloud application with reference to AWS services. Describe and support the reason for each microservice creation (minimum of 1 microservice).

Design Screenshots	Low-fidelity design screenshots of the cloud application and navigation flows to illustrate the working of the application.
Database Design	Relational DB: Data dictionary and Entity-Relationship diagram. NoSQL DB: Data dictionary
API	API documentation that defines the usage of each microservice.

Some suggested web applications are as follows:

- Assets Management
- Career Guide
- E-Commerce Stores
- Facilities Booking
- Food Ordering
- Learning Management
- Restaurant Reservation
- Social Platform for Hobbyists
- Task Management
- Traveling

You are free to be as creative as you can and do NOT have to select from the above. However, please ensure that the cloud-based web application can be completed within the specified timeframe. Do discuss your ideas with your tutor and submit a project proposal report by **Week 5 (22 May, Thursday, 11:59pm)**.

Submission Instructions

Submit your Microsoft Word document into TP LMS under the Assessments folder. Save your proposal as ***YourName_StudentID_YourClass.docx*** (e.g.: *JohnTan_1234567D_P01.docx*).

Part 2 – Microservices Implementation (20%) and Demonstration (10%)

You are required to implement the API gateway and the proposed microservices using services in AWS. Your implementation should include:

- Databases with good set of records/items
- Full set of working API routes

You will use POSTMAN to demonstrate your microservices implementation to your tutor. You are encouraged to explore AWS-related additional features that are relevant to this section such as other AWS Database engines, Lambda Authorizer to protect the API endpoints, etc.

Besides submitting your project source codes on **Week 11 (30 June, Monday, 8:30am)**, you are to present and demonstrate your work during your assignment evaluation, which will be held in **Week 11 (Week of 30 June) - Timings to be advised by your tutor**.

Submission Instructions

Compress and submit the following as a zipped file in TP LMS under the Assessments folder.

- AWS Lambda functions (refer to LMS for instructions to export Lambda functions)
- API documentation (refer to LMS for template)
- JSON exports of your API routes (refer to LMS for instructions)
- Database exports (refer to LMS for instructions)

Your zipped document file should be named according to the following format:

YourName_StudentID_YourClass.zip
e.g.: JohnTan_1234567D_P01.zip

Please ensure that you have a backup copy of your zipped file in case there is a problem with the online submission.

Part 3 – Cloud Application (30%) and Demonstration (10%)

You are required to develop a cloud-based web application using AWS services. The cloud application should have a simple UI/UX and must consume **at least 4** of the API endpoints that were created in Part 2. You are not restricted to only using Amazon S3 as your web server but are free to explore other possible AWS services. You are encouraged to explore AWS-related additional features that are relevant to this section such as securing your web application, monitoring, etc. (You may refer to the list of possible additional features in the LMS)

Besides submitting your project source codes on **Week 15 (28 Jul, Monday, 8:30am)**, you are to present and demonstrate your work during your assignment evaluation, which will be held in **Week 15 (Week of 28 Jul) - Timings to be advised by your tutor**.

Submission Instructions

Compress and submit the following as a zipped file in TP LMS under the Assessments folder.

- Web application source files
- AWS exports: Database, Lambda functions + API routes
- URL of your web application (.txt file)
- Recording of your web application (maximum 10 mins)

Your zipped document file should be named according to the following format:

YourName_StudentID_YourClass.zip
e.g.: JohnTan_1234567D_P01.zip

Please ensure that you have a backup copy of your zipped file in case there is a problem with the online submission.

Penalty for Late Submission

late and <1 day : 10% deduction from absolute mark given for the assignment
late >=1 and <2 days : 20% deduction from absolute mark
late >=2 days : No marks awarded

Project - Grading Criteria

The grading criteria for Part 1 (20%) will be based on the following:

Criteria	**In Context**	**Performance Level**				
		A	**B**	**C**	**D**	**F**
Application Description (8)	Depth of description (6) Visual Aids (2)	Description of proposed functionalities, target audience and their intended access control is comprehensive and meaningful. Visual aid such as use case diagram is used to clearly illustrate the intended access control.	Description of proposed functionalities, target audience and their intended access control is comprehensive. Visual aid such as use case diagram is used to clearly illustrate the intended access control.	Description of proposed functionalities and target audience is completed but limited elaboration. No visual aids or irrelevant visual aids.	Description of proposed functionalities and target audience is completed with no elaboration. No visual aids or irrelevant visual aids.	Description of proposed functionalities and target audience is incomplete. No visual aids or irrelevant visual aids. Non submission of all required items Clear evidence of plagiarism detected (disciplinable offence)
Microservices Architecture Design (8)	Diagram (5) Reasons (3)	Comprehensive architecture design that contains multiple components, comprises of comprehensive relationships between the different services. Reasons given for microservices creation is valid and sound.	Complete architecture design that contains few components, comprises of some relationships between the different services. Reasons given for microservices creation is valid.	Limited architecture design that contains few components, lack of some relationships between the different services. Reasons given for microservices creation is limited and vague.	Sparse architecture design that contains few components, missing relationships between the different services. No or unsound reasons given for microservice creation.	Sparse architecture design that contains many missing components and/or relationships between the different services. No or unsound reasons given for microservice creation Non submission of all required items Clear evidence of

						plagiarism detected (disciplinable offence)
API Documentation (8)	Comprehensiveness (2) Correctness (4) 4 API endpoints covering 4 HTTP methods (2)	Listing and description of API is comprehensive and detailed, supporting all cases from proposed functionalities. At least 4 API endpoints covering GET, POST, PUT and DELETE.	Listing and description of API is comprehensive, supporting all proposed functionalities. At least 4 API endpoints covering GET, POST, PUT and DELETE.	Listing and description of API is limited, with no elaboration, supporting some proposed functionalities. At least 4 API endpoints covering GET, POST, PUT and DELETE.	Listing and description of API is incomplete, supporting a few proposed functionalities. Less than 4 API endpoints or does not cover full set of GET, POST, PUT and DELETE.	Listing and description of API is mostly incorrect, supporting a few proposed functionalities. Less than 4 API endpoints or does not cover full set of GET, POST, PUT and DELETE. Non submission of all required items Clear evidence of plagiarism detected (disciplinable offence)
Database design diagram and Data Dictionary (8)	ER Diagram (4) Data Dictionary (4)	Extensive database design diagram that contains appropriate tables and constraints used. All data items in the tables are elaborated with more details with the aid of their respective data dictionary.	Database design diagram that contains mostly correct tables and constraints used. All data items in the tables are elaborated with more details with the aid of their respective data dictionary.	Partial database design diagram that contains some correct tables and constraints used. Data Dictionary that consists of incorrect information.	Sparse database design diagram that contains few correct tables and constraints used. Data Dictionary that consists of incorrect information.	Sparse database design diagram that contains few correct tables and constraints used. Data Dictionary that consists of incorrect information. Non submission of all required items Clear evidence of plagiarism detected (disciplinable offence)

Design Screenshots (8)	Comprehensiveness (4) UI/UX Principles (4)	Low-fidelity design screenshots clearly and fully illustrate all functionalities highlighted in the project scope. Design is Intuitive, easy to learn, and easy to use.	Low-fidelity design screenshots clearly and fully illustrate most functionalities highlighted in the project scope. Design is easy to use once you learn it.	Low-fidelity design screenshots partially illustrate the functionalities highlighted in the project scope. Design is easy to use once you learn it.	Low-fidelity design screenshots barely illustrate the functionalities as highlighted in the project scope. Design is easy to use once you learn it.	Non submission of all required items Clear evidence of plagiarism detected (disciplinable offence)

The grading criteria for Part 2 - Microservices Implementation (20%) will be based on the following:

Criteria	**In Context**	**Performance Level**				
		A	**B**	**C**	**D**	**F**
Microservices Implementation (40)	API Implementation (30)	API implementation is working, supports all proposed functionalities, and cover a good set of exceptional cases.	API implementation is working and supports all proposed functionalities.	API implementation is working but supports only some proposed functionalities.	API implementation is incomplete, supporting only a few proposed functionalities.	API implementation is mostly not working, supporting only a few proposed functionalities. Non submission of all required items Clear evidence of plagiarism detected (disciplinable offence)
	Database (10)	Database implementation is completed and tables contain comprehensive set of meaningful records/items.	Database implementation is completed and tables contain good set of records/items.	Database implementation is complete and tables contain very few records/items.	Database implementation is incomplete and/or tables contain very few records/items.	Database implementation is largely incomplete and/or tables contain very few records/items. Non submission of all required items Clear evidence of plagiarism detected (disciplinable offence)

The grading criteria for Part 2 – Solution Demonstration (10%) will be based on the following:

Criteria	In Context	Performance Level				
		A	B	C	D	F
Solution Demonstration (20)	Demo (10)	Flawless execution with no errors; smooth, polished, and professional demo.	A few minor errors occurred, but they were handled well and did not affect the overall flow or clarity of the demo.	Some errors were present, causing slight disruption, but the demo remained mostly clear and understandable.	Several errors affected the flow and made the solution harder to execute, but effort to demo was evident.	The solution could not be demonstrated successfully due to unresolved technical issues, despite attempts to present. Absent without valid reason.
	Delivery (10)	Delivery is very fluent and well-paced Able to explain well and beyond expectation.	Delivery is mostly fluent and well-paced Able to explain well.	Delivery is somewhat fluent and well-paced Able to explain partially.	Delivery is somewhat fluent and well-paced Able to explain with some difficulty.	Absent without valid reason.

The grading criteria for Part 3 - Cloud Application (30%) will be based on the following:

Criteria	In Context	Performance Level				
		A	B	C	D	F
Cloud Application (60)	API Consumption (40)	All required features are implemented and fully functional. The solution handles all valid and invalid inputs gracefully, with clear and helpful feedback messages. No crashes or unexpected behavior.	Most features are implemented, and most of them work as expected. Handles most edge cases and invalid inputs with appropriate feedback. Minor lapses in error handling.	Some major features are missing or incomplete, and not working as expected. Limited error handling. Some invalid inputs cause issues or are not managed properly.	The solution is largely incomplete, with limited working features. Minimal or no error handling.	Solution is barely completed and not sufficiently tested with all normal data. Non submission of all required items. Clear evidence of plagiarism detected (disciplinable offence)
	Additional features (20)	Demonstrates thoughtful integration of multiple additional AWS services with seamless interaction. Services are highly relevant, enhancing functionality, scalability, performance, or security in meaningful ways.	The additional AWS services are well-integrated with some interaction between components. Services are mostly relevant and contribute to the project's success, though with some room for improvement or optimization	Basic implementation of additional AWS services with limited integration or cohesion. Relevance is partial, services provide utility but do not significantly enhance the project.	Implementation of additional AWS services is minimal, poorly implemented, or disconnected from the project's goals. Relevance is unclear or superficial, with little to no impact on the overall quality of the project	No additional feature implemented. Clear evidence of plagiarism detected (disciplinable offence)

The grading criteria for Part 3 – Solution Demo (10%) will be based on the following:

Criteria	In Context	Performance Level				
		A	B	C	D	F
Solution Demonstration (20)	Demo (10)	Flawless execution with no errors; smooth, polished, and professional demo.	A few minor errors occurred, but they were handled well and did not affect the overall flow or clarity of the demo.	Some errors were present, causing slight disruption, but the demo remained mostly clear and understandable.	Several errors affected the flow and made the solution harder to execute, but effort to demo was evident.	The solution could not be demonstrated successfully due to unresolved technical issues, despite attempts to present. Absent without valid reason.
	Delivery (10)	Delivery is very fluent and well-paced Able to explain well and beyond expectation.	Delivery is mostly fluent and well-paced Able to explain well.	Delivery is somewhat fluent and well-paced Able to explain partially.	Delivery is somewhat fluent and well-paced Able to explain with some difficulty.	Absent without valid reason.

End of Project Specification