

Individual Project (20 Marks)

Due at 3 PM 24/10/2025 (Friday in Week 12)

Overview and objectives:

The **goal of this project** is to propose a cloud-based application in a proposal (5 marks) and implement the application (15 marks) using learned cloud computing technologies. You need to choose one of two types of projects below and find a specific topic suitable for the chosen type. Some project ideas will be demonstrated in video recordings of previous student projects that won the student project competition last year. A novel project with excellent implementation will be nominated for the student project competition awards this year. In this open project, your **understanding of cloud computing concepts and your ability to use respective cloud technologies** will be assessed. As an individual project, you are required to complete both the proposal and implementation all by yourself. The proposal submission will be automatically checked against the existing documents, including academic publications, books, and articles on the web in the Turnitin databases. Cases with a HIGH similarity score will be sent to the Academic Ethics and Integrity Committee for further investigation. The implementation will be marked by tutors **in person** in marking sessions in Week 13. You must submit the proposal and the source code on the blackboard before **3:00 PM 24/10/2025**. It is your responsibility to ensure the submission is on time.

Proposal (5 Marks):

The proposal aims to test your understanding of the concepts, characteristics, and relevant technologies in Cloud Computing. Moreover, your ability to design a cloud-based application with a reasonable budget will be assessed. The project proposal must NOT exceed 1000 words and must be within two A4 pages. Your proposal should include the following sections:

- Introduction:
 - Give background information about this project: *What is this project about?*
 - Explain the motivation of this project: *Why is this project important?*
 - Describe the overall objective and features of the project: *What features does this project have?*
 - Explain the limitations of traditional computing solutions: *Why doesn't traditional computing solve the problem well?*
 - Explain the benefits brought by cloud computing: *How do cloud computing technologies fit into this project?*
- Technical Solutions:
 - Describe what cloud technologies you've used in this project (e.g., Kubernetes for deployment, NoSQL for the back-end data storage).
 - Provide a monthly cost estimation of all the cloud resources used in this project (e.g., costs of VMs, K8s cluster, networking, Load Balancers, etc.).
- Architecture Design:
 - Depict the workflow or framework of the project in a figure (e.g., micro-service framework).

To broadly accommodate students with different backgrounds, the projects are categorized into two types:

- **Type I - Highly scalable and available web application:**

This project type requires a scalable and reliable web-based application using Micro-service architecture and related technologies. This project type focuses on developing and deploying scalable, reliable, and resilient applications in the cloud.

- e.g., PHP/JavaScript + MySQL/NoSQL + Docker/K8s + Load Balancing.
- The deployed application should be scalable and support rolling updates and rollback features.

- **Type II - Big Data query and analytics:**

This project type requires a data-analytic application using computing models for Big Data. This project type focuses on solving complicated real-world problems with a large amount of data by either querying from databases or analyzing with Machine Learning algorithms.

- e.g., Jupyter Notebook + Scala/Python/Java + Spark SQL / Streaming / MLlib
- Real-world problems should be solved by analyzing the large-scale data.

You can only choose **one type** and find out the specific topic that suits the type according to your background and preference. No matter which type you choose, the cloud infrastructure costs should be reasonable. You can use the Google Cloud Platform Pricing Calculator when making the budget plan in the proposal.

Implementation (15 Marks):

The implementation aims to test your ability to build and deliver a cloud application as proposed in your project proposal. The implementation should be technically consistent with the features and functionalities outlined in the proposal. You are allowed to use your own code in the previous programming/analytical courses or an open-source web project (with consent). For example, you may reuse your previous web development project from INFS3202, but you need to containerize the application and deploy it on the cloud. If the previous project was a group one, it is your responsibility to obtain the team members' consent and you must clearly inform the team members that the workload of the individual project in INFS3208 is to deploy the application in the cloud with the taught technologies.

To demonstrate the implementation, you need to present your implementation to your tutor in the marking sessions in Week 13. The presentation should include, but not be limited to, the following parts:

- Background & Motivation.
- Project architecture and applied technologies.
- For Type I projects, database design (for databases) or data storage (for distributed file systems).
- For Type II projects, descriptions of the used data and analytic models (Spark programming).
- Results & Discussion

Technical requirements:

There are some technical requirements that must be met in your proposal and implementation. These technical requirements should be demonstrated in your presentation as well.

If your project belongs to **Type I**:

- For the **front-end** design, you should have a functional user interface allowing users to interact with the application. And there should be at least 4 different functionalities in your application.

- To support the overall objectives of this project, you should have a **back-end** design (e.g., database) working together with the front-end UI.
- You should use docker technologies to **containerize** your applications with multiple containers running in a **microservice architecture**.
- To make your application scalable, reliable, and resilient, you should use either **Docker Swarm** or **Kubernetes** to orchestrate multiple containers across hosts for your application.

If your project belongs to **Type II**:

- You need to perform a **challenging** analytic problem that requires large-scale data processing (e.g., big data queries, classification, regression, clustering, association rule mining over big data). You need to pick up a **large dataset in the real world (must be at least 10, 000 records and contain multiple data files or tables)**. Some publicly big datasets are recommended below.
- To persist the data, you need to **store** the data in either a database (relational/non-relational databases, e.g., MySQL/Redis/MongoDB), a vector database (e.g., Faiss) or a distributed file system (e.g., HDFS).
- To avoid “Garbage in Garbage out”, you need to perform **data pre-processing** (e.g., removing or replacing NaN values, converting data types, imputation, etc.) to ensure the data quality for the downstream analytical tasks.
- You should run the analytical task on a Spark cluster using **Spark programming** techniques and the related **Spark built-in libraries** (i.e., SQL, MLlib, Streaming, and GraphX). You can choose either Python or Scala as the programming language.
- To demonstrate the analysis results, your program should **visualize** results with some tools (e.g., matplotlib in Python) in Jupyter Notebook.

You MUST back up the implementation code and data regularly (at least every week). It is your responsibility to ensure that you fully understand the technical requirements of this individual project. If you have any concerns or problems, please contact your session’s tutor, or ask questions on Ed Discussion.

For students using GCP, you should use the credit wisely and avoid all unnecessary expenses on cloud services during the project. Also, you should monitor the balance of the credits during the implementation. **It is strongly recommended to develop, test, and debug locally before deploying the project on the cloud.**

Submission:

You should make an online submission **before 3 PM 24/10/2025 (Friday in Week 12)**:

- Your proposal should be no more than 1,000 words in two pages.
- The source code and the relevant data should be compressed into a single file and should be submitted to Blackboard.
- If the data is too large for uploading, you should provide an external link (e.g., OneDrive or Dropbox link) in the submission and ensure the data source is valid.

You are welcome to discuss your proposal and implementation with the teaching team during tutorial/practical sessions or on Ed Discussion.

*Publicly available datasets:

1. Queensland Government Open Data Portal, <https://www.data.qld.gov.au/>
2. Datasets on Kaggle, <https://www.kaggle.com/datasets>
3. Awesome Public Dataset list on Github, <https://github.com/awesomedata/awesome-public-datasets>
4. MovieLens dataset, <https://grouplens.org/datasets/movielens/>

