




## School of Computing Technologies

# COSC2639 Cloud Computing

## Assessment 1

	<b>Assessment Type:</b> This is an <b>individual</b> assignment. Submit your solutions online via Canvas→Assignments→Assessment 1. Marks are awarded for meeting the requirements as closely as possible. Clarifications/updates/corrections may be made via announcements/relevant discussion forums.
	<b>Due date: 8:00AM; Monday 2<sup>nd</sup> October 2023.</b> Please check Canvas→Assignments→Assessment 1 for the most up-to-date information. A university-standard late penalty of 10% per each working day applies for up to 5 working days late, unless an extension or special consideration has been granted.
	<b>Weighting:</b> 30 marks

### 1. Overview

This assignment consists of 3 tasks:

- Task 1: Develop a cloud-based application and deploy it in Google App Engine.
- Task 2: Develop a Google BigQuery application.
- Task 3: Draft a proposal for Assessment 3 (Specification of the Assignment 3 is to be released in early Week 3).

**[Note:]** Students are free to use *any programming language of their choice* for this assignment. However, using a programming language that is supported out-of-the-box by the Google App Engine (GAE) is highly **recommended**.

### 2. What the assignment depends on

This assignment is predominantly based on what you will learn in **Week 2: Build cloud applications** section in Canvas->Modules. More specifically, it focuses on your understanding on,

- Google App Engine
- Google Datastore/Firestore (**NOT** Firebase!)
- Google Cloud Storage
- Google BigQuery

### 3. Assessment Criteria

This assessment will strengthen your ability to

- develop cloud-based applications using Google App Engine.
- develop Google BigQuery based applications to perform data analytics using Google BigQuery and visualize the results.

### 4. Learning Outcomes

This assessment is relevant to the following Learning Outcomes:

- Develop and deploy cloud application using popular cloud platforms.

## 5. Assessment details

### Task-1

[23 marks]

Create a simple online forum application with Google Datastore/Firestore and Google Cloud Storage as well as deploy it to Google App Engine using your learned knowledge in **Week 2: Building cloud applications** section. This application will have following components and functions.

1. Create 10 entities in Google Datastore/Firestore (Namespace: default, Kind: user, Key: default), which contain the following properties and values. **1 mark**

id (Type: String)	user_name (Type: String)	password (Type: String)
s3#####0 i.e. your RMIT student id+'0'	Firstname Lastname0 i.e. your name+'0'	012345
s3#####1 i.e. your RMIT student id+'1'	Firstname Lastname1 i.e. your name+'1'	123456
...	...	...
s3#####9 i.e. your RMIT student id+'9'	Firstname Lastname9 i.e. your name+'9'	901234

2. Store 10 images into Google Cloud Storage either manually or programmatically. Each image shows a unique single digit (from 0 to 9). These images will be used as default user images.  
**[Note:]** These images could potentially be sourced from a free icon provider platform E.g. <https://pixabay.com/vectors/search/numbers/>. Feel free to use your own digit images, if you prefer. **No marks allocated**

3. Login page **2 marks**

The login page contains an “ID” text field, a “Password” field, and a “Login” button as well as a “Register” link. When user clicks the “Login” button, it would validate if the user entered credentials match with the information stored in the Datastore/Firestore.

- 3.1. If the user credential is invalid, the login page will display “ID or password is invalid”. **1 mark**
- 3.2. If the user credential is valid, it will be redirected to the forum page. **1 mark**

4. Register page **5 marks**

The register page contains an “ID” text field, a “Username” and a “Password” field, a “User Image” uploading field and a “Register” button. When user clicks the “Register” button, it would validate if the user entered id and username match with the information stored in the Google Datastore/Firestore.

- 4.1. If the entered id matches with the id stored in the Google Datastore/Firestore, the register page will show “The ID already exists”. **1 mark**
- 4.2. If the entered username matches with the user\_name stored in the Google Datastore/Firestore, the register page will show “The username already exists”. **1 mark**

- 4.3. If both the entered id and username are unique,
  - 4.3.1. the new user information will be stored in the Google Datastore/Firestore, **1 mark**
  - 4.3.2. the uploaded user image will be stored in Google Cloud Storage, and **1 mark**
  - 4.3.3. the user will be redirected to the login page, where user can login with the new user\_id and password. **1 mark**

5. Forum page **8 marks**

The forum page contains three main areas: a user area, a message posting area and a message display area and a “Logout” link.

- 5.1. After a user logs in, the user area will show
  - 5.1.1. the login user\_name (which is a hyperlink - if user clicks the user\_name link, it will be redirected to the user page), and **1 mark**
  - 5.1.2. the login user image **1 mark**.

For example, the area should show “your name+’1” and image “1” after the user with ID “s3#####1” logins.

**[Note:]** Make sure the displayed user image size is no greater than 120 pixel x 120 pixel.

- 5.2. The message posting area contains a “Subject” field (nonempty), a “Message Text” field, an “Image” uploading field, and a “Submit” button.
  - 5.2.1. Once the user clicks the “Submit” button, the subject, the message text and the uploaded image will be displayed in the message display area, **1 mark**
  - 5.2.2. the subject and the message text will be stored in Google Datastore/Firestore, and **1 mark**
  - 5.2.3. the image will be stored in Google Cloud Storage. **1 mark**
- 5.3. The message display area should show 10 latest posted messages (including subjects, message texts and images), their posting date and time, corresponding user\_names and user images. **2 marks**
- 5.4. If a user clicks the “Logout” link, the user will be redirected to the login page. **1 mark**

6. User page **7 marks**

The user page should contain a password edit area and user posts edit area.

- 6.1. The password edit area contains an old password field, a new password field and a “Change” button, where user needs to enter both the old password and a new password.
  - 6.1.1. If user clicks the “Change” button, it will validate whether the entered old password matches with the information stored datastore. If the old password is incorrect, it will display “The old password is incorrect”. **1 mark**
  - 6.1.2. If the old password is correct, it will update the user password stored in the Google Datastore/Firestore and be redirected to the login page, where user can login with the new password. **1 mark**

## 6.2. The user post edit area

- 6.2.1. displays all the message information (message subject and posting date and time) posted by the login user. Each message is followed by an “Edit” button. **1 mark**
- 6.2.2. Once the user clicks the “Edit” button of a posted message, it will display text fields prefilled with the detailed post information (message subject and message text) and previously uploaded image that allow the user to edit the post, followed by an “Update” button. **1 mark**
- 6.2.3. Once the user enters the new post information into the edit area and clicks the “Update” button,
  - 6.2.3.1. the user will be redirected to the forum page, where the updated post will be displayed in on top of the message display area with the new posting date and time, **1 mark**
  - 6.2.3.2. the subject and the message text stored in the Google Datastore/Firestore will be updated, and **1 mark**
  - 6.2.3.3. the image stored in the Google Cloud Storage will be updated. **1 mark**

### Important Notes:

1. The solution to Task 1 **MUST** be demonstrated via a web-application deployed in Google App Engine. Any work hosted locally is **NOT** accepted as a valid solution.
2. The whole application must be **COMPLETELY** programmed by yourself. You **CANNOT** use any external source code (e.g. no external libraries to implement sign-in, etc).
3. The whole application must be **FULLY** deployed in Google App Engine, otherwise **NO MARKS** will be given for Task 1.

## Task-2

**[6 marks]**

This task is based on the work done in relation to **Google BigQuery**.

- 1) Use the **a1.zip** dataset (three csv files: gsquarterlySeptember20, services\_classification, country\_classification) to write some queries to retrieve the following information.
- 2) Use BigQuery API Client Libraries to create an application based on your queries. (**Note:** This is a compulsory requirement otherwise Task-2 **WON'T** be marked).
- 3) Deploy your application in your Google App Engine so that the query results can be displayed on the webpage once you type in your Google App Engine URL (**Note:** This is a compulsory requirement otherwise Task-2 **WON'T** be marked).

(Each correctly displayed query result is worth 2 marks)

1. Show top 10 time slots (year and month) with the highest trade value (i.e. import value + export value). **2 marks**  
**Note:** Show “time\_ref” and “trade value” in your results.

2. Show top 40 countries with the highest total trade deficit value (i.e. import value - export value) of goods from 2013 to 2015 where status is "F". **2 marks**  
**Note:** Show "country\_label" (full country name), "product\_type", "trade deficit value" (in a descending order), and "status" in your result.
3. Show top 25 services with the highest total trade surplus value (i.e. export value - import value) in the top 10 time slots of Query Result 1 and the top 40 countries of Query Result 2. **2 marks**  
**Note:** Show "service\_label" (full service name) and "trade surplus value" (in a descending order) in your result.

**Important Notes:**

1. Running queries on the BigQuery console and demonstrating the outputs during the evaluation is **NOT** accepted as a valid solution to Task 2. The query result **MUST** be displayed on a web page, which is part of an application deployed into Google App Engine.
2. Solutions to all three queries can be demonstrated as part of a single web application deployed into Google App Engine.
3. The whole application must be **COMPLETELY** programmed by yourself. You **CANNOT** use any external source code (e.g. no external libraries to implement sign-in, etc).
4. The whole application must be **FULLY** deployed in Google App Engine, otherwise **NO MARKS** will be given for Task 3.

**Task-3****[1 mark]**

Write at least 100 words to describe the preliminary plan of your Assessment 3 (to be released in **early Week 4**), including the motivation, the objective, the possible system components, the proposed functions and the possibly adopted AWS services and/or external APIs for each component.

**Note:**

- 1) References should be included when needed (excluded in the word count).
- 2) This task only aims to encourage you to start preparing for your Assessment 3 early. The quality of Task-3 content will not be strictly assessed. You may discuss the details of your preliminary plan with the tutor assigned to your class during assignment consultation sessions.

**5. Referencing guidelines**

**What:** This is an individual assignment, and all submitted contents must be your **OWN**. If you have used sources of information other than the contents directly under Canvas→Modules, you must acknowledge the relevant sources and provide references using IEEE referencing style.

**Where:** Add a code comment near the work to be referenced and include the reference in the IEEE style.

**How:** To generate a valid IEEE style reference, please use the [citethisforme tool](#) if unfamiliar with this style. Add the detailed reference before any relevant code (within code comments).

## 6. Submission format

Create a .zip file and name it [your\_student\_number].zip (e.g. s1234567.zip). This .zip file will contain items in the following order.

1. A .txt file with your name and student number, as well as your response to “Task 3”.
2. Code of Task 1 (app.yaml, relevant source files based on your preferred programming language) within a directory named “Task 1”.
3. Code of Task 2 (app.yaml, relevant source files based on your preferred programming language) within a directory named “Task 2”.

Submit it into [Canvas→Assignments→Assessment 1](#) before the deadline. You will be marked during your demo time. This submission is only for keeping the records. However, your assessment will **NOT** be allowed to be demonstrated in Canvas until you submit your file.

## 7. Demonstration

You must demonstrate your project online to the tutor assigned to your class (i.e. Class A – Prabath; Class B – Tim) by making an appointment with the tutor in **Week 6**. The demo booking form will be made available to students in **Week 5**, and all the demonstrations must be completed within **Week 6**. There will be a **penalty** if you fail to complete and demonstrate your work by Week 6. The demo is around **20 minutes** for each student. Keep everything ready and make your application live (on 2 Google Cloud projects) before your demo begins.

**Note:** Each student can only attend one demonstration for Assessment 1. **DO NOT** overbook otherwise you will receive an academic penalty. If you are not satisfied with your marks, you may attend the online remarking session organized by the Offering Coordinator after **Week 6** (remarking time and booking will be made available to students through announcements).

## 8. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own. RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the [University website](#).

## 9. Assessment declaration

When you submit work electronically, you agree to the [assessment declaration](#).