UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

FACULTY OF ENGINEERING AND APPLIED SCIENCE

Department of Electrical, Computer and Software Engineering

SOFE 4630U: Cloud Computing – Winter 2019 (Dr. Q. Mahmoud)

Individual Programming Assignment #1

Demo and (code on Github) by 5:00pm on Jan 29 & Report on Blackboard by 11:59pm on Jan 29

The objective of this individual programming assignment is to get a flavour of the effort involved in designing and developing fully-functional web services in the cloud. You will practice designing and developing an innovative cloud-based application of your choice using a cloud provider offering infrastructure as a service (i.e. you must instantiate a VM and configure it yourself).

Task and Guidelines

Design and develop an innovative, easy to use, and useful cloud-based application of your choice. It must accomplish something useful and has some innovative features. Your cloud application must provide at least 4 functions (services or features; note that authentication will not count as a feature in this assignment), and must be built according to the following requirements:

1) Must instantiate a VM of your choice (Linux or Windows Server)
2) Must install and run a Web server of your choice on the VM
3) Must install and run a SQL database of your choice such as MySQL, PostgreSQL, MariaDB, etc. Your app must use a SQL database (number of tables is up to you) for storage/retrieval of data/info.
4) Must install a Web framework of your choice, and it can be a basic framework such as LAMP for Linux or WAMP for Windows; or advanced frameworks such as Angular, React, Laravel, NodeJS, Ruby on Rails.

You may use any programming language including: Java, JavaScript, Python, Ruby, PHP, etc.

You may use any cloud service provider, but must use infrastructure as a service (instantiate a virtual machine in the cloud): AWS EC2, Google Cloud, IBM Bluemix, Microsoft Azure, etc.

Important Notes

- You need to demo your working cloud application (7 minutes) to the Teaching Assistant (Ahmed Badr). Email him at Ahmed.Badr@uoit.net to book an appointment.
- Deadline: Assignment#1 must be demoed **by 5:00pm on Tuesday, January 29**. No extensions no matter what is the reason, so plan accordingly.
- Your solution must be designed and developed by yourself (your own work).
- While students are encouraged to discuss the assignment and general ideas for solutions, each student must design and develop his/her own assignment and code. No code sharing allowed, and no students have the same application.
- The assignment will be assessed based on the grading rubrics provided on page 2 of this document.
- If during the demo, your application hangs in the middle of a connection, you will have a second chance to demo again (must be within 5 minutes). If this behaviour persists, it means your application is not completely functional, and you will lose 75% of the whole assignment mark. Your application

must be fully functional in order to get reasonable marks on the other assessment items (see grading rubrics on next page).

Submission Guidelines (pay special attention to the deadlines)

1) Source code and readme file of your application – must be submitted on Github by **5:00pm on Tue, Jan 29** – go to the following link for Assignment1: https://classroom.github.com/a/u2zk-Hed

2) The report must be submitted through Blackboard by **11:59pm on Tue, Jan 29** (look under Content -> Assignments –> Assignment #1 Report Submission). It must be in PDF or Word and must be one full-page (approx. 500 words) detailing your application, innovative feature(s), challenges and solutions. You may include one clear diagram but no screenshots of the application.

Grading Rubrics

Item (%)	**Excellent (90-100%)**	**Very Good (75-89%)**	**Satisfactory (50-74%)**	**Unsatisfactory (0-49)**
Functionality (20)	Fully functional with no errors or warning.	Functional but nothing special about it and sometimes no response.	Basic functionality beyond code covered in class.	Doesn't run or nothing functional.
Usefulness and usability (20)	Useful and intuitive to use	Works but nothing special	Requires a manual to use.	Non-existent.
Innovative features (20)	Creative and offers innovative functionalities.	Doesn't implement sufficient unique functionalities.	Nothing special.	Non-existent.
Report (20)	Clearly documented and well organized with innovative features, challenges and solutions.	Readable but not well organized or missing parts.	Documentation is brief.	No documentation.
Source code (20)	Follows coding standards (name, date, title, meaningful variable names, whitespaces, etc.) and code is fully documented.	Readable source code. Doesn't follow coding standards.	Spaghetti code.	No source code provided or the link to the source code is not accessible.