**SINGULAR SYSTEMS**

**TECHNICAL CHALLENGE - DEVOPS**

# INSTRUCTIONS

This practical assessment aims to assess your ability to solve technical problems given a predefined specification.

## What to submit

A link to your GitHub repository containing the following:

* The full commit history as you build the solution (incremental commits are preferred over a single large commit).
* The final solution, and its assets.

## Technologies to use

* PowerShell 7
* HTML/YAML
* Docker
* GitHub
* Suggested Editor: Visual Studio Code (PowerShell extension recommended)

## Guidelines

* Ensure that you are the creator of the project and that you did not use existing projects found online.
* This assessment does not aim to be over-prescriptive - create a solution that you feel comfortable with to accurately portray your skill set.
* You may research and use technologies with which you are not familiar.
* The use of advanced PowerShell features such as functions, classes, pipelines, etc will be rewarded.
* Being creative and going the extra mile will count in your favour.
* Have fun!

# SCENARIO

You have been tasked with downloading and analysing an application’s log files in order to provide a report on the number of info, warning and error messages being logged per month. The log files sit online in an Azure storage account. An index file containing a list of the log files is located here: <https://files.singular-devops.com/challenges/01-applogs/index.txt>

The log files sit under the same folder as the index file. They are stored in a fixed width CSV format. Some basic schema detail is available here:

<https://files.singular-devops.com/challenges/01-applogs/schema.md>

# TASKS

Using PowerShell, write a script that performs the following actions:

* Download and read the contents of the index file.
* Use the index file to generate links for and download each of the application log files, and save them to a local folder in the current working directory named logs.
* Run through the contents of each log file and extract the following information:
  + The month and year.
  + The number of info, warning, and error messages.
* Generate a report file in JSON format that contains an array of the monthly statistics:
  + The year and month
  + Number of info, warning, and error messages
  + The percentage increase or decrease in warnings and errors from the previous month
* Save the report asset as a file named report.json in a report folder under the current working directory.
* In addition to the report.json asset, generate a human-readable HTML asset named report.html, based on the report.json, in the same report folder. The styling can be kept as basic or as advanced as you choose.
* In addition to