**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>

# The business would like a web view of the analysed application’s log assets. They also require a functional POC that can host a containerised application that displays the produced static content in a web-app.

# 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 index.html, based on the report.json, in the same report folder. The styling can be kept as basic or as advanced as you choose.

**Alongside the report assets, we would like you to perform the following:**

* Use a public AI service of your choice (like ChatGPT, Claude, etc) to generate a favicon.ico for the web-app.
* You are required to supply publicly accessible link(s) to all interactions with AI services while building your solution, for [example](https://chatgpt.com/share/68c7edfb-b4f4-8005-b371-c713c8fe6b66).

**You are also tasked with building a POC to meet the following objectives:**

* a GitHub Action file named deploy.yaml that will deploy the static assets in the report folder to your GitHub repository’s public Pages site.
* A Dockerfile that can be used to host the web-app, the Dockerfile should be accompanied by a script that can be invoked to start/run the containerised application.

**Finally, you are also required to create a README.MD write-up file containing at least the following:**

* All setup and deployment instructions.
* All of the shared development chats with the public AI service. Please ensure the links are accessible before sharing your final solution, for [example](https://chatgpt.com/share/68c7edfb-b4f4-8005-b371-c713c8fe6b66).
* Your thoughts on the Technical Challenges you faced, how you went about analysing and solving them. Given more time, what would you have liked to implement or do differently?