**ITLB354 MIB Automation**

**Home Coding Assignment**

**Task Description**

Please write a**1,000-word** code and documentation, as a **ZIP file**, that fulfils the following specification and has documentation on the execution and the features of the application. Upload the ZIP file to Moodle.

**Deadline**: Please see on the Moodle page of the module.

The assignment will be assessed based on the following criteria (see the grid on Moodle):

* Code Validity (50%)
* Clean Code (30%)
* Documentation (20%)

The **resit arrangement** for the assignment is the same as above; you may resubmit the same paper, with corrections, that you submitted by the original deadline. The resubmission deadline will be specified on Moodle after the grades for the original submission are published.

**Academic conduct notice**

Where the Academic Conduct Officer has reason to suspect that a piece of work submitted by a student was wholly or in part written by someone other than the student who submitted it, and this has not been disclosed by the student, they may call for the student to defend the work in a **viva or a written comprehension test**. The burden of proof in such a viva or test will be upon the student to demonstrate to the examination panel’s satisfaction his/her full comprehension of the work s/he has submitted. Failure to appear without satisfactory explanation will result in immediate failure of that assessment, with consequences of academic misconduct and application of sanctions.

# Project: Scraping Automation

Create an automation process for surveying the popularity of last month’s top Billboard artists. The process involves gathering data from the Billboard website, combining the collected data, then (based on the gathered information) generating a form and processing the survey results.

**Documentation**

Please provide documentation for the automation process in Markdown format. The documentation should provide details on how to run and install the application in a new environment including any dependencies; describe the functionality; and mention any limitation and edge case that may arise.

# Stories

## Data Extraction

Create a web scraping application that gathers the names and rankings of the artists from the Billboard Artist 100 lists of the past 4 weeks. The current week’s list can be found at <https://www.billboard.com/charts/artist-100/>. The preceding weeks’ lists can be found by clicking the calendar icon at the top of the page.

Make sure that dates are not hard-coded, i.e. whenever the application is executed it will get the actual week’s and the previous 3 weeks’ lists.

## Data Transformation

Combine the names of the top 5 artists from each of the Artist 100 lists of the past 4 weeks into one list. Make sure that every artist is present in the combined list once (there is no duplication). The artists should be sorted alphabetically.

The transformation step may either be implemented as a separate script or as part of the scraping application.

## Data Loading

Load the combined list of artists into a new Google Sheet.

The data loading step may either be implemented as a separate script or as part of the scraping application.

## Scheduling

The above extract, transform, load pipeline should be scheduled to run once a month.

Add instructions on how to configure the scheduler to the documentation. You can assume that the pipeline is executed on a Linux server.

## Form Generation

Create a Google Apps Script that generates a Google Form that lists all artists from the uploaded Google Sheet, and allows users to rate each of them on a one-to-five scale.

## Data Summary

Create a Google Apps Script that calculates the average rating of each artist from the form responses, and inserts the rating averages next to the artists’ names in the uploaded Google Sheet.

For the Form Generation and Data Summary steps, copy-paste your solutions from the online Google Apps Script text editor into a **GS** file and include it in the uploaded ZIP file.