2023

Technical Manual

TOPIC: TRACKING HDR TEACHING HOURS

- **SRIRAJ Y RAO – 101856783**
- **HEWA PERUMA KAPUGAMAGE ROSHANI NADEERA - 102878571**
- **SONU CHANDRAKANT GAUTAM - 103812813**
- **YASHODHA MALLUWA WADU - 103843033**
- **UDARI CHETHANA SENANAYAKE - 102626741**

1.Introduction

Before the implementation of this system, the academic staff at Swinburne University made incorrect decisions regarding the allocation of sessional staff to classes, resulting in an invalid payment processing system. According to the client, there should be a maximum 8-hour teaching cap per PhD student, but this information was not accurately captured due to manual Excel manipulation tasks.

The newly developed system automates the manual process by reading various Excel files and generating a final, interactive live consolidation with the required fields and total teaching hours.

This document's main objective is to deeply discuss the technical background of this developed system.

2. System Landscape Diagram

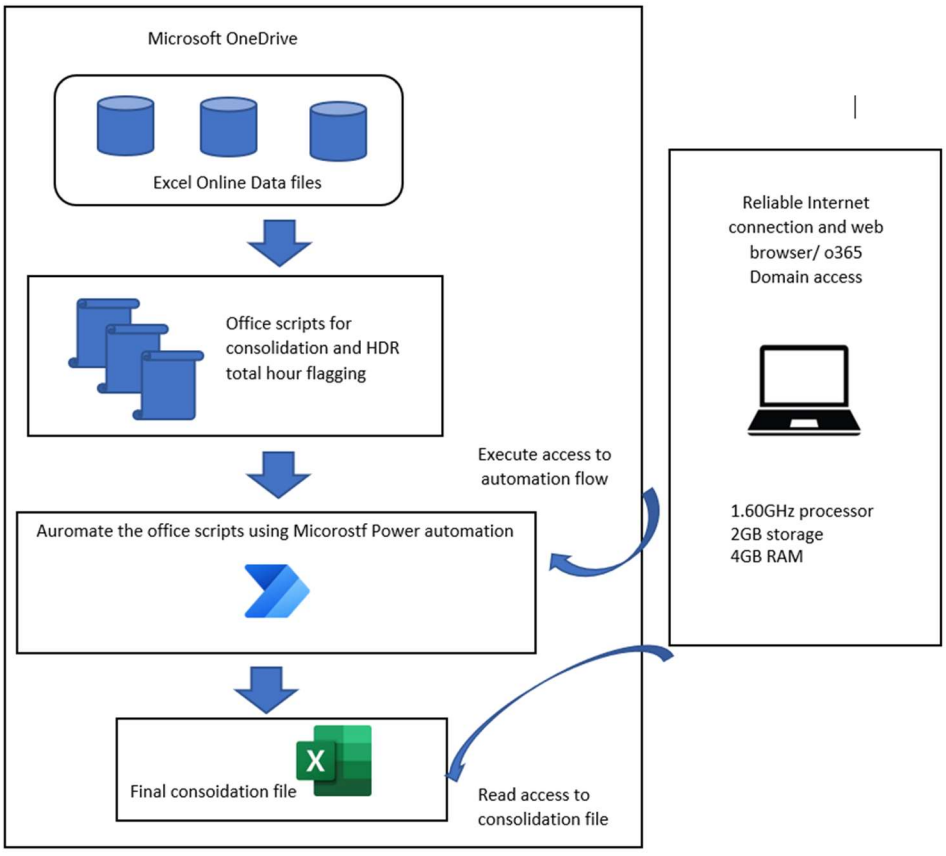
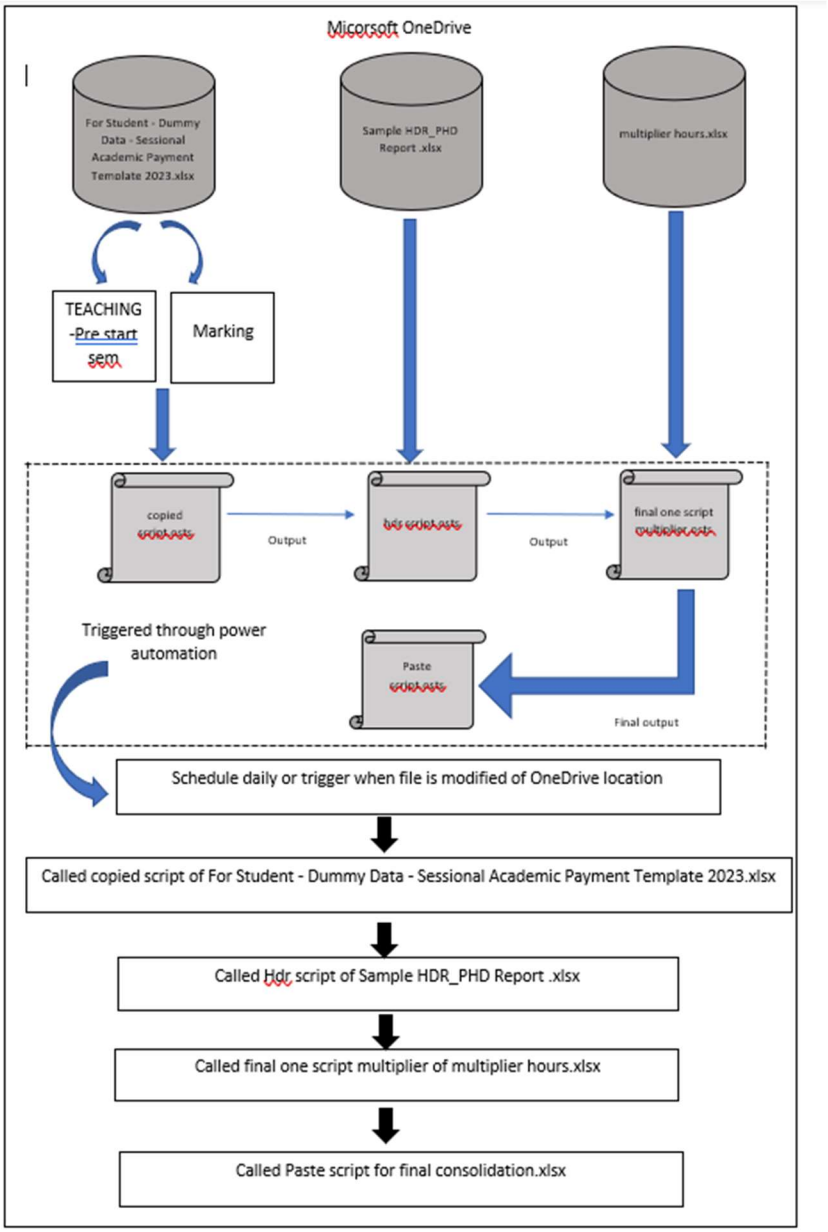


Figure 1: System landscape diagram

3. System architecture diagram



No.	Object	Development Stack
1	Consolidation and HDR hour calculation logic	Office Scripts
2	Trigger the Office scripts	Microsoft power automation

4. Basic Office Scripts and Automation tasks

Office scripts are a type of script that represents a specific version of JavaScript. They are easy to build for Excel automation. Microsoft Power Automate is an excellent option for triggering these office scripts and creating the final consolidation.

While there is an option to trigger changes in any files and initiate the flow (live update), we recommend scheduling it hourly or daily due to the large amount of data available in the files. Running the flow each time carries a risk of Excel clashes.

5. Debugging

Although the office scripts trigger through an automation flow, these scripts can be debugged manually as below.

- **Copied Script**

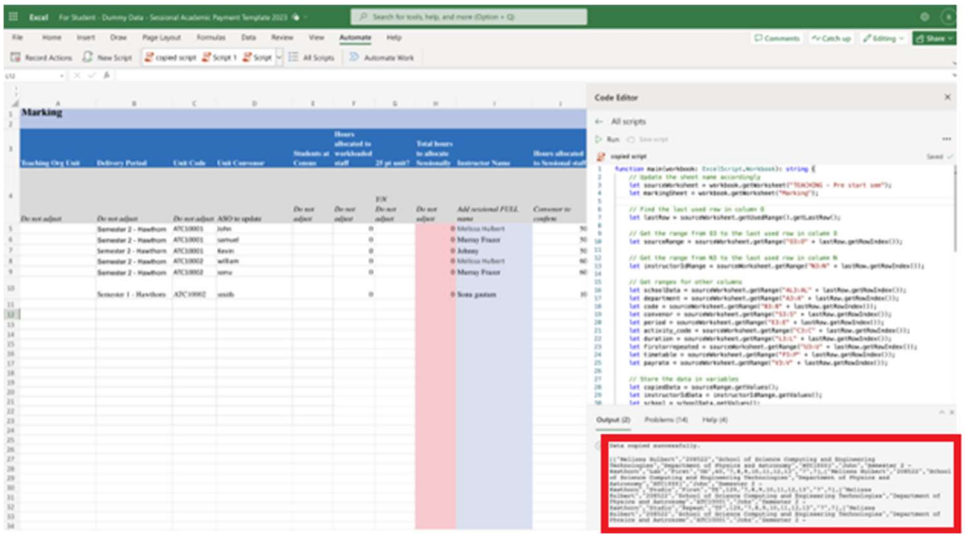


Figure 2: The image shown is copied script

When you open the copied script through the code editor, as shown in the above screenshot, and click the 'Run' button, the output is displayed in a red-highlighted window with a friendly message and an output JSON array.

- **HDR script**

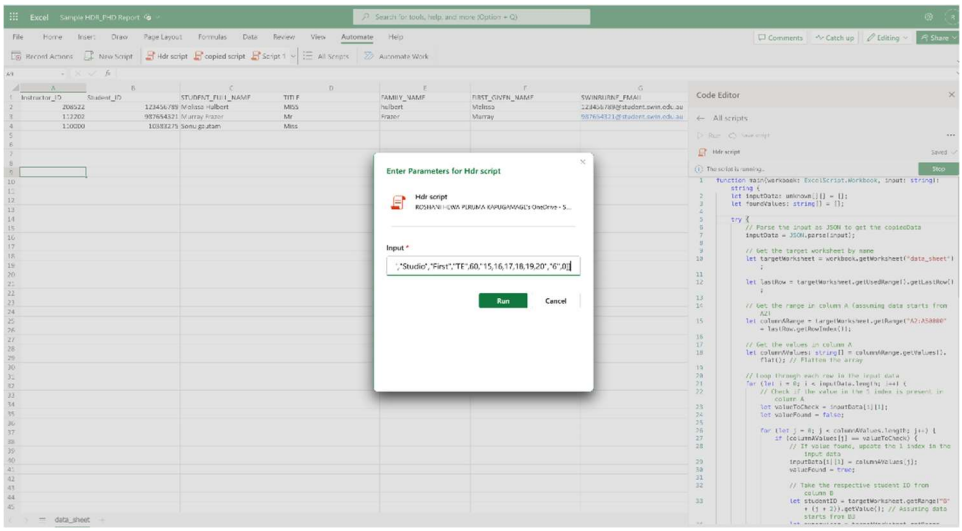


Figure 3: Input data in the script

When you run the HDR script by opening the HDR report, the above windows has been appeared, requesting input parameter values. The output JSON array should be entered as the input parameter for this script.

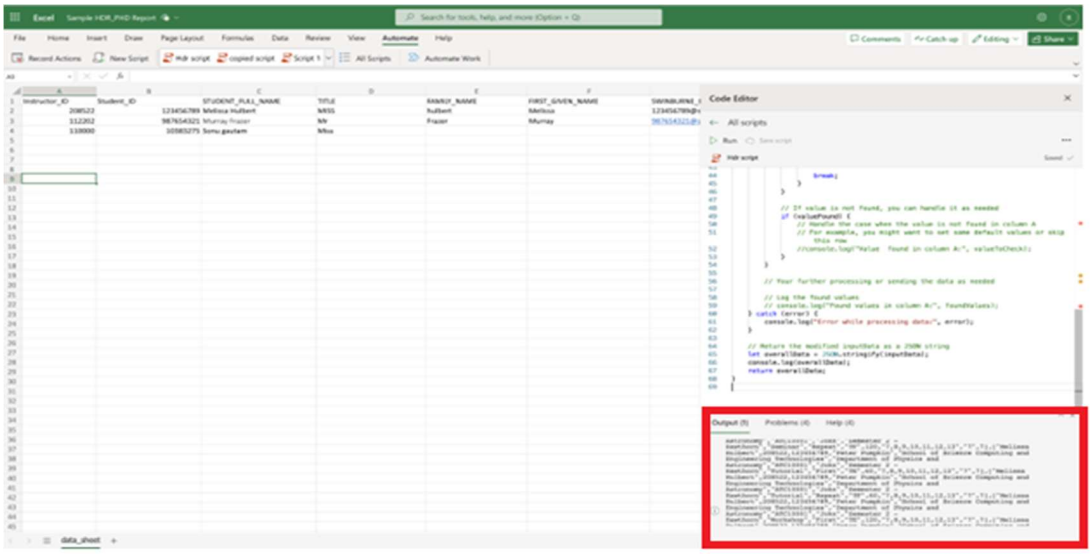


Figure 4: Output data after performing the operation.

If this script runs properly, the JSON array output highlighted above will be displayed without any errors.

- **Final one script multiplier**

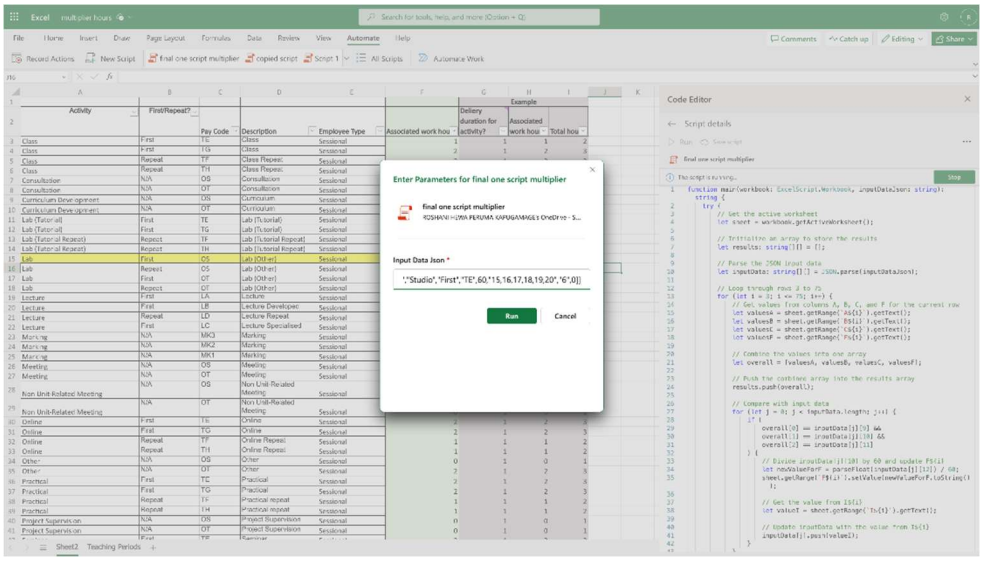


Figure 5: Input data into the script

Similarly, to the HDR report template, when running the multiplier hours template, you need to enter the input parameter, which is the output of the HDR report.

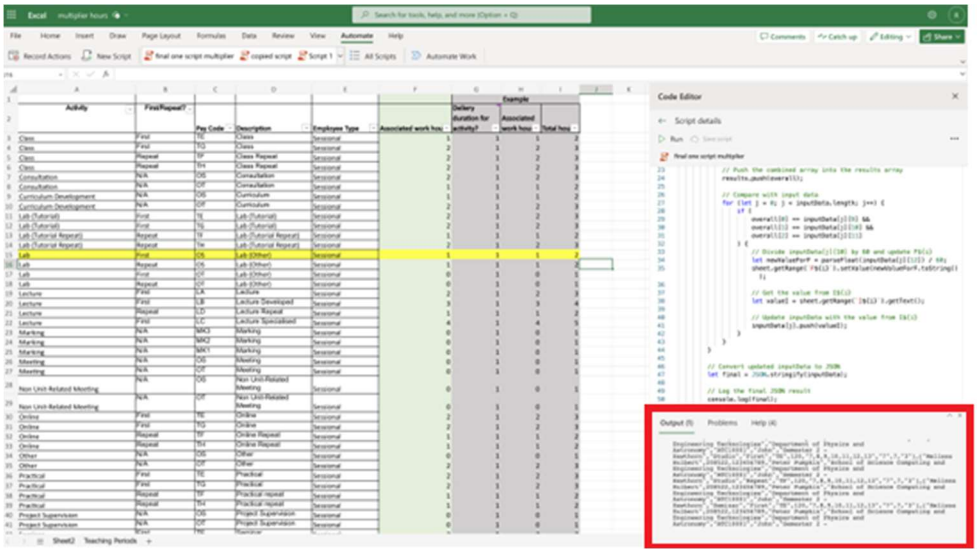


Figure 6: output after performing operation.

If this script runs successfully, the output shown above will be in the form of a JSON string.

- **Paste script.**

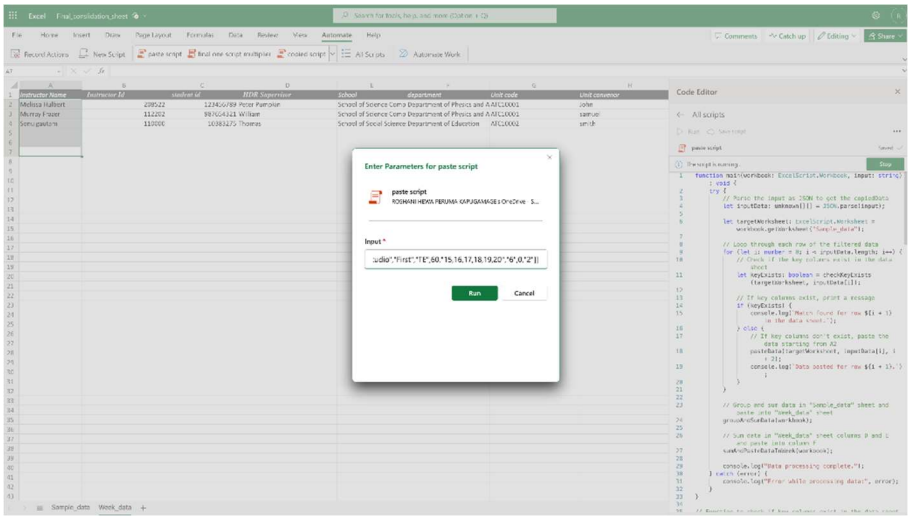


Figure 7: input to the script

This is the final script which generates the consolidation file by the final output Json of above file.

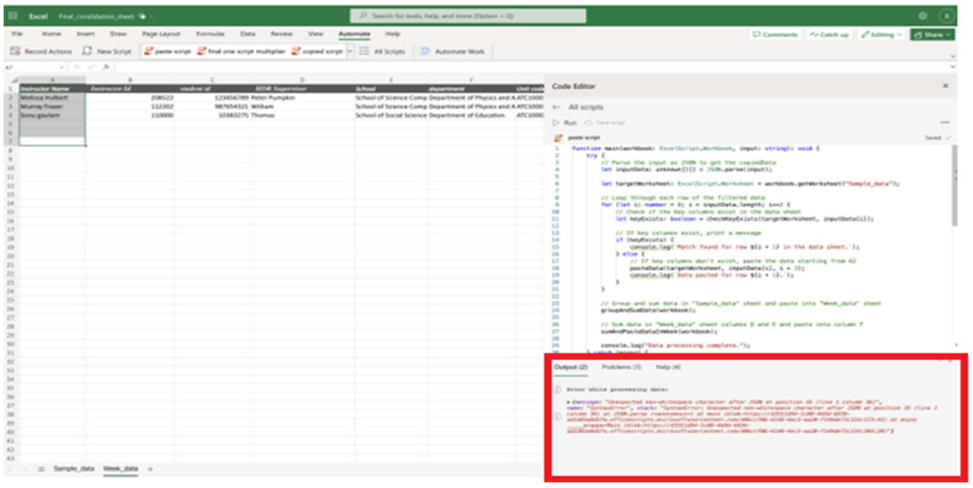


Figure 8: Final output of the script and display of the final data.

Once the script runs with errors, the screen above displays a detailed error message. The error is legible and easily fixable. In this case, it indicates a whitespace character issue. We need to resolve this error to achieve the final desired outcome.

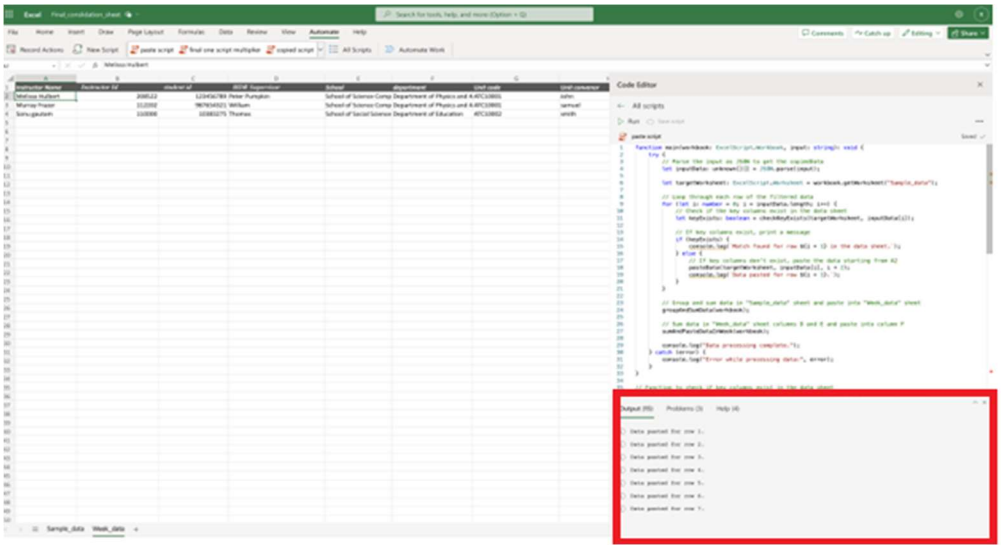


Figure 9: checking for duplicate and pasting the data.

Once the final step is completed successfully, the rows are copied to the final consolidation file, and the result will be displayed as shown in the messages above.