UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Malaysia-Japan International Institute of Technology (MJIIT)

SEMESTER 1 SESSION 2022/2023

SMJE4383 - ADVANCED PROGRAMMING

ASSIGNMENT 1

AUTOMATE THE CSV GENERATION PROCESS USING ROBOTIC PROCESS AUTOMATION (RPA) & PYTHON

NAME	Matric No.
MUHAMMAD HANIF BIN ABU KASSIM	A20MJ0060
SITI NUR IZZATI BINTI SANOSI	A19MJ3063

Section: 01

Lecturer Name: Assoc. Prof. Ir. Dr. Zool Hilmi Bin Ismail

Submission Date: 7 February 2023

Github link: https://github.com/IzzatiSanosi/Advanced-Programming/tree/Assignment1

TABLE OF CONTENTS

1.0 INTRODUCTION

The process of generating CSV (Comma-Separated Values) files can often be time-consuming and repetitive, requiring manual data entry and manipulation. With the growth of Robotic Process Automation (RPA) technology, however, there is now an opportunity to automate this process and save valuable time.

RPA is a type of software that automates repetitive, routine tasks, freeing employees to focus on more valuable work. It can be used to automate a wide range of processes, including data entry, data manipulation, and report generation. By combining RPA with the power of Python programming language, organizations can streamline their CSV manipulation process, resulting in improved efficiency, accuracy, and cost savings.

In this report, we will explore the steps required to automate the CSV generation process using RPA and Python. We will cover the installation and setup of RPA software, the creation of Python scripts to automate copying data, and the integration of these scripts into an RPA workflow.

2.0 OBJECTIVES

Below are the objectives for this project:

i. To create a sample of CSV file using CSVpad software.
ii. To write a python code that can import and export CSV file and send notification to email once the process is completed.
iii. To analyze, debug, solve and automate the CSV Generation Process using RPA & Python.

3.0 METHODOLOGY

Below are the process flow for this project:

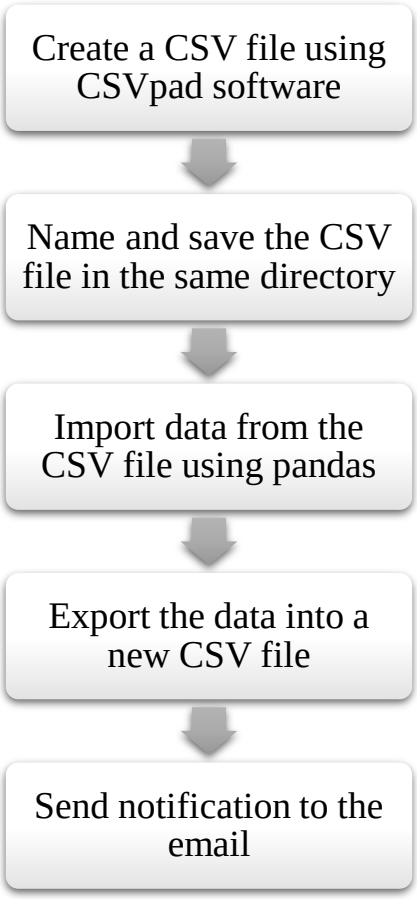


Figure 3-1: Process flow of the program

Firstly, a CSV file was created using CSVpad software. CSVpad software is a handy free CSV (Comma-separated values) editor. It supports Unicode and it is a portable application. CSVpad can be used to generate and manipulate CSV files. In this project, a CSV file that contained the students grade for test 2 Arab language was created using the CSVpad software.

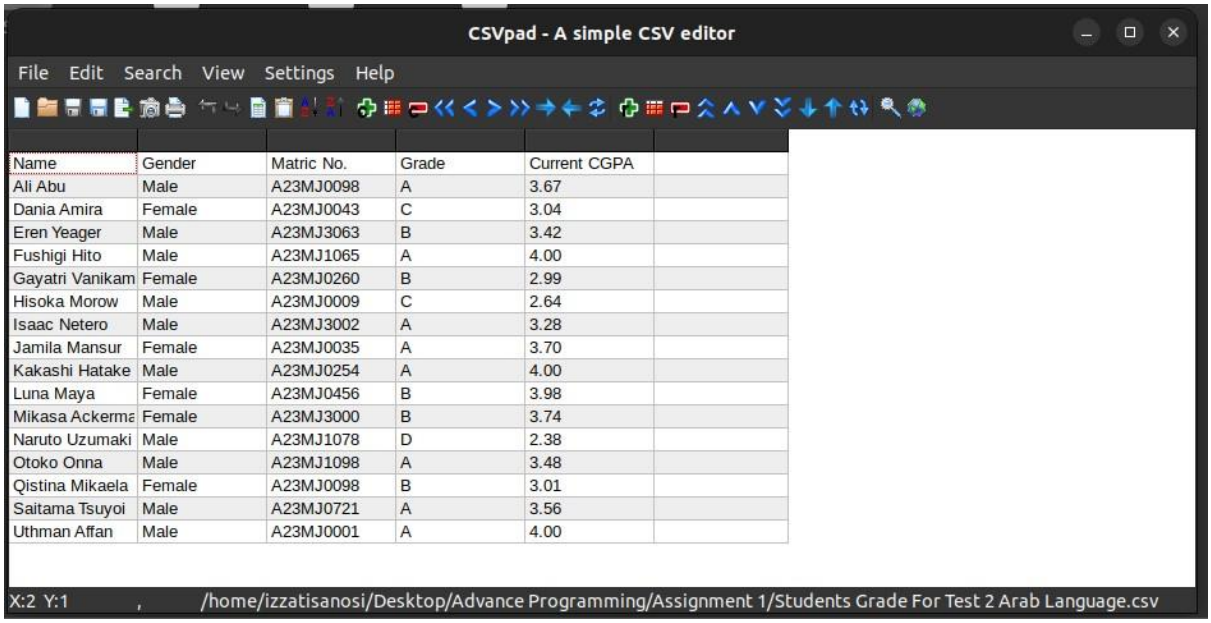


Figure 3-2: CSV file created using CSVpad software.

Later, the file is named as Students Grade For Test 2 Arab Language.csv and saved in the same directory.

Next, Python script was executed to import the file using pandas and export the imported data into a new file. Lastly, a notification will be sent to email once the process is completed. Sending an email using Python requires additional steps as we have to create app password for the sender's email. Below are the steps to create the app password:

i. Go to Google Account > Security > enable the 2-Step Verification.

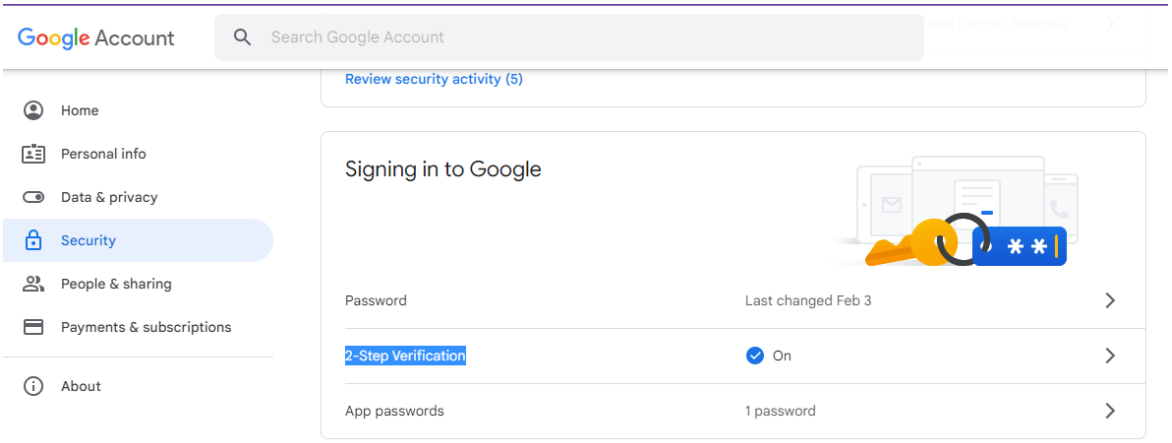


Figure 3-3: Enabling the 2-step verification.

ii. Go to app passwords > Select app > Other (Custom name) > write Python > click Generate.

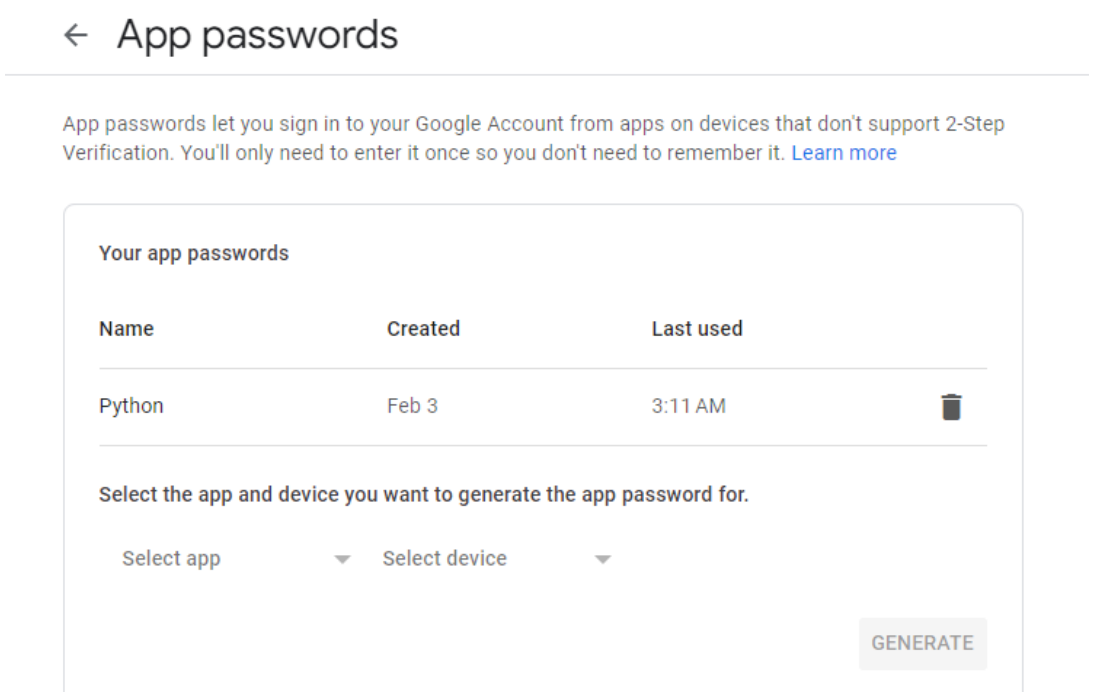


Figure 3-4: App password named Python.

iii. 16 Character password is automatically generated. This password is needed for the sender email to send email using Python.

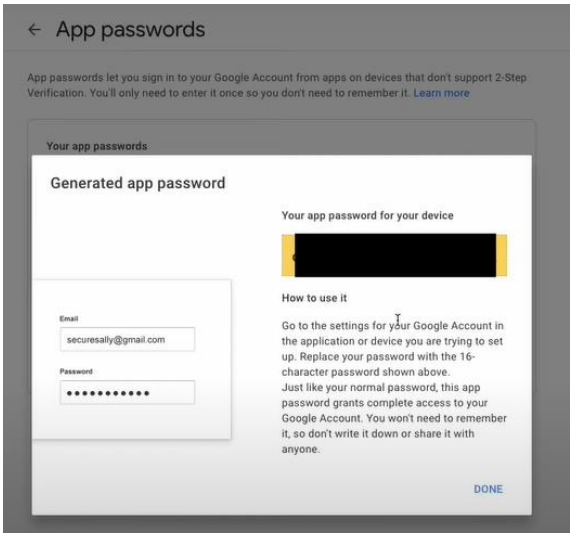


Figure 3-5: 16-character password is generated.

4.0 RESULT AND DISCUSSION

Below is the Python script used for the program:

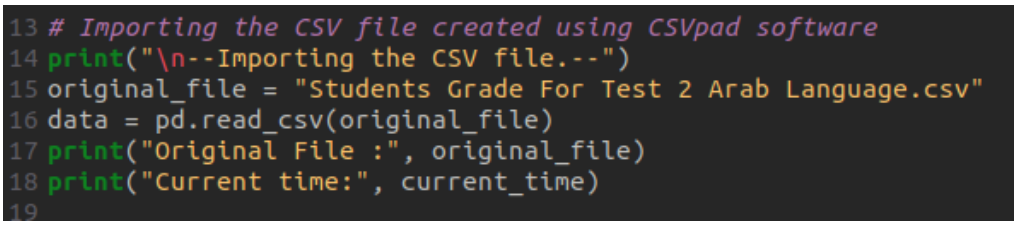
```
import datetime
import os
import pandas as pd
import smtplib

#To print time and current directory path
current_time = datetime.datetime.now()
curDir = os.getcwd()

print("\nCurrent directory:", curDir)
print("Task: Automatically copy data from a CSV file into a new CSV and send notification to email once completed.")
```

Figure 4-1: Modules used in the script.

As a programmer, it is recommended to state the date, time, directory and the function of the script in our program. The date and time is crucial to check weather the output file is updated after the script was run or not. Stating the function of the script helps user to understand the usage of the script. The functions of the imported libraries in Figure 4-1 are:

a) "datetime" is a library that provides classes for working with dates and times, such as obtaining the current date and time.
b) "os" is a library that provides a way to interact with the operating system, such as getting the current working directory.
c) "pandas" is a library that provides data structures and data analysis tools for working with structured data, such as reading and writing CSV files.
d) "smtplib" is a library that provides a way to send email messages using the Simple Mail Transfer Protocol (SMTP).

In this project, the function is the script is to automatically copy data from a CSV file into a new CSV and send notification to email once completed.



```
# Importing the CSV file created using CSVpad software
print("\n--Importing the CSV file.--")
original_file = "Students Grade For Test 2 Arab Language.csv"
data = pd.read_csv(original_file)
print("Original File :", original_file)
print("Current time:", current_time)
```

Figure 4-2: Importing the CSV file.

The code in Figure 4-2 imports a CSV file named "Students Grade For Test 2 Arab Language.csv" using the pandas library. The file is assigned to the variable "original_file" and the contents of the file are loaded into a pandas dataframe called "data". Finally, it prints the name of the original file and the current time for the user.

```
# Saving the dataframe to a new CSV file
print("\n--Copying the CSV file.--")
copied_file = "[FINAL] Students Grade For Test 2 Arab Language.csv"
data.to_csv(copied_file, index=False)
print("Copied File :", copied_file)
print("Current time:", current_time)
```

Figure 4-3: Copying the CSV file.

The code in Figure 4-3 creates a copy of the CSV file loaded into the pandas dataframe "data". The copy is named "[FINAL] Students Grade For Test 2 Arab Language.csv" and assigned to the variable "copied_file". The pandas function "to_csv" is used to save the dataframe "data" to the specified file, with the option "index=False" meaning that the index values of the dataframe will not be written to the file. Finally, it prints the name of the copied file and the current time.

```
# Specify output path
path = curDir + "/" + copied_file

# Check whether the specified output path exists or not, send email if exist
isExist = os.path.exists(path)
if isExist == True:
    print("The output file exists. Sending notification to email...")

    sender_email = 'izzatisanosi98@gmail.com'
    password = [redacted]
    receiver_email = 'sitinurizzati@graduate.utm.my'

    message = """Subject: !!SUCCESSFULLY EXPORT CSV!!

    Konnichiwa!
    The data has been successfully copied and saved to a new CSV file.
    Arigatou gozaimashita."""

    server = smtplib.SMTP("smtp.gmail.com", 587)
    server.ehlo()
    server.starttls()
    server.login(sender_email, password)
    server.sendmail(sender_email, receiver_email, message)
    server.quit()

    print("\nCopy Completed! Email notification was sent to", receiver_email, "!")
    print("Current time:", current_time, "\n")
```

Figure 4-4: Send notification to email if the process is completed.

The code in Figure 4-4 checks if the specified output path (constructed by concatenating the current directory path and the name of the copied file) exists using the "os.path.exists" function. If the output file exists (indicated by the variable "isExist" being equal to "True"), the code proceeds to send an email notification.

The email is sent from the Gmail account "izzatisanosi98@gmail.com" with the 16 character of app password . The recipient of the email is "sitinurizzati@graduate.utm.my". The message of the email is a pre-defined string that indicates that the data has been successfully copied and saved to a new CSV file.

The email is sent using the SMTP (Simple Mail Transfer Protocol) library in Python (smtplib). The Gmail SMTP server is connected to using "smtp.gmail.com" and port 587. The email sender logs into the Gmail account using the "login" method, sends the email using the "sendmail" method, and finally quits the SMTP server using the "quit" method.

Finally, the code prints a message indicating that the email notification has been sent to the specified recipient and the current time.

When the script is executed and there is no error, the information in Figure 4-5 below will be printed on the terminal.

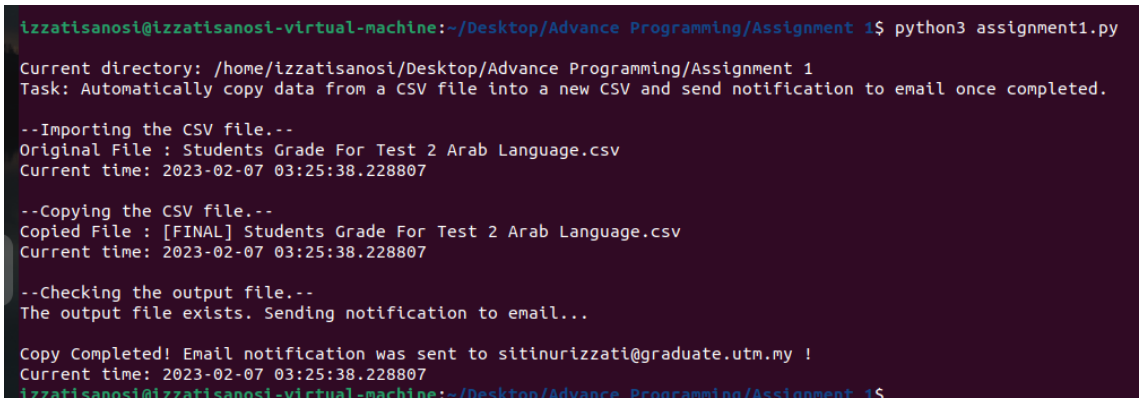


Figure 4-5: Output in terminal

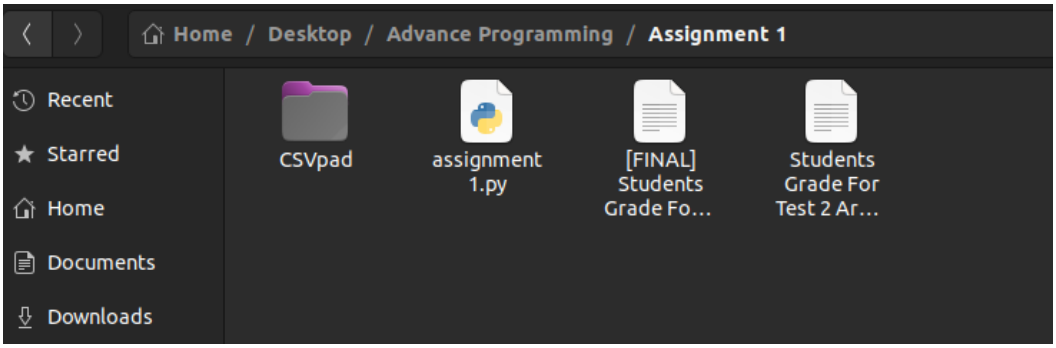


Figure 4-6: Input, output, and Python script in the same directory.

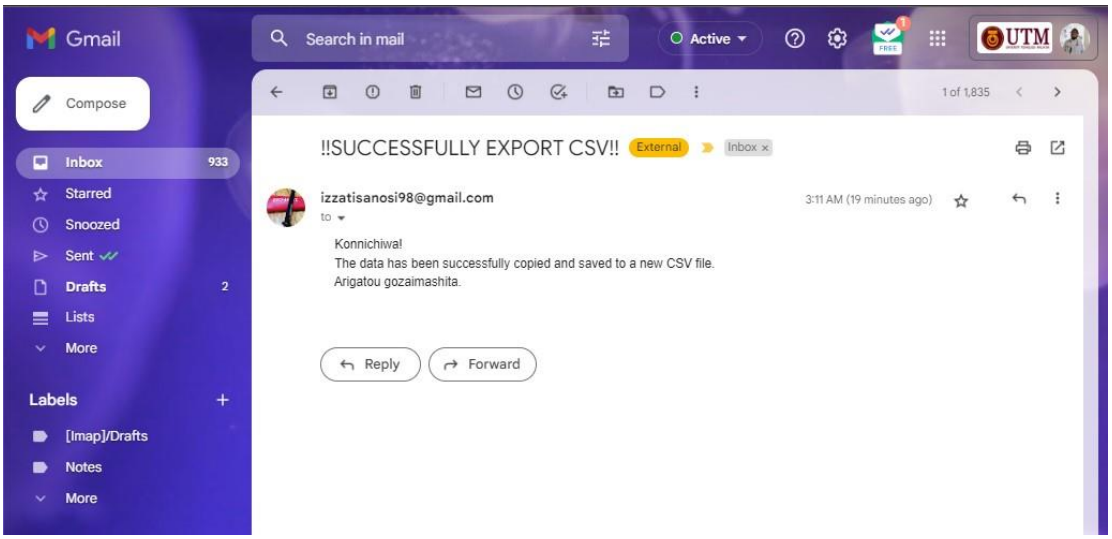


Figure 4-7: Email successfully received.

Figure 4-6 and Figure 4-7 shows the output of the program. Output file is created in the same directory and an email is received by the receiver stating that the process of copying the CSV file have been completed.

5.0 CONCLUSION

In conclusion, the integration of Robotic Process Automation (RPA) and Python to automate the CSV generation process is a game changer for any organization that relies on this type of data processing. The use of RPA to automate routine and repetitive tasks, combined with the versatility and power of Python programming, delivers a highly effective and efficient solution. With RPA and Python, the process of generating CSV files can be done faster, more accurately, and with a higher degree of consistency than traditional manual methods.

The implementation of this automation solution can bring numerous benefits to an organization, including increased productivity, reduced errors, and cost savings. The RPA technology automates routine tasks, freeing up valuable time and resources that can be used for more important tasks. Meanwhile, the Python programming language provides the customization and fine-tuning capabilities necessary to meet specific business requirements. The result is a streamlined and efficient process that delivers high-quality results.

In short, automating the CSV generation process using RPA and Python is a smart choice for any organization looking to improve efficiency, accuracy, and speed while reducing costs. It has the potential to transform data processing and drive significant business value.

6.0 REFERENCES

[1]UiPath Inc, “What Is Robotic Process Automation?,” Uipath.com, 2017. https://www.uipath.com/rpa/robotic-process-automation

[2]Python Software Foundation, “Datetime — Basic Date and Time Types — Python 3.7.2 Documentation,” Python.org, 2002. https://docs.python.org/3/library/datetime.html

[3]Real Python, “Sending Emails with Python,” Realpython.com, Dec. 05, 2018. https://realpython.com/python-send-email/

[4]“CSVpad Main Page (Official Page),” www.trustfm.net. https://www.trustfm.net/software/utilities/CSVpad.php#:~:text=CSVpad%20is%20a%20handy%20free

[5]Google, “Sign in with App Passwords - Gmail Help,” support.google.com. https://support.google.com/mail/answer/185833?hl=en

[6] McKinney, W. (2012). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc.

[7]Tripathi, A. M. (2020). A Beginner's Guide to RPA: Automating Business Processes with UiPath. Packt Publishing.