**Project Based Learning Report**

on

**Email Automation using Python and Excel**

Submitted in the partial fulfillment of the requirements For the Project based learning in (**Essentials of Data Science**)

in

Electronics & Communication Engineering

By

**Ayush Rai 2014111092**

**Praveen Tiwari 2014111128**

**Ishit Verma 2014111135**

Under the guidance of Course In-charge

Prof. Dnyanesh S. Lavhkare

Department of Electronics & Communication Engineering

Bharati Vidyapeeth

(Deemed to be University)

College of Engineering,

Pune – 4110043

**Academic Year: 2021-22**

**Bharati Vidyapeeth**

**(Deemed to be University)**

**College of Engineering,**

**Pune – 411043**

**DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING**

**CERTIFICATE**

Certified that the Project Based Learning report entitled, **“Email Automation using Python and Excel”** is work done by

**2014111092 Ayush Rai**

**2014111128 Praveen Tiwari**

**2014111135 Ishit Verma**

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in **Essentials of Data Science Course** of Bachelor of Technology Semester IV, Electronics & Communication Engineering.

**Date: 21 May 2022**

**Prof. Dnyanesh S. Lavhkare Dr. Tanuja S. Dhope**

**Course In-charge PBL Co-Ordinator**

**Dr. Arundhati A. Shinde**

**Professor & Head**

**ELECTRONICS & COMMUNICATION ENGINEERING**

|  |  |
| --- | --- |
| **Index: -** | |
| Page No. | Contents |
| 4-4 | Problem Statement with Solution |
| 5-6 | Description about project |
| 7-7 | Software Used |
| 7-13 | Results with Analysis |
| 14-14 | Conclusion & Outcome |

**Problem Statement :-**

We need to put all email ids with the receiver's name in *email.xlsx* [ any spreadsheet ]. We will read them all, and send mails one by one in your automated python script.

**Solution :-**

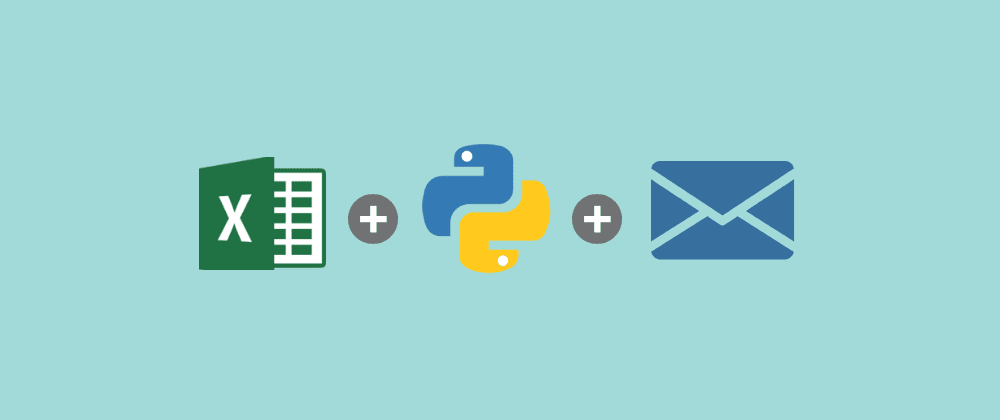
Email automation is a way to create emails that reach the right people with the right message at the right moment—without doing the work every time, sending automated messages leveraging a marketing automation tool. In email automation, a trigger is a specific date, event, or contact’s activity that tells your system to send out a related message. You can [choose triggers and triggered messages](https://mailchimp.com/resources/email-marketing-automation-triggers/) based on your contact’s journey. As the world becomes more and more personalized, the benefits of email automation will undoubtedly make your job as an email marketer easier. Using automated email in your email campaign helps increase engagement and provide a more tailored email experience for your recipients.

**Email Automation Using Python and Excel**

**Email Automation**

Email automation is an effective way to communicate with potential customers because it allows you to send emails at specific times. Most email apps have pre-made automated campaign workflows—sequences of emails strung together.

The biggest value of email automation is the ability to send relevant emails to the right people at the right time. If you link your CRM to an email marketing app, you can nurture subscribers effectively thanks to the data about their purchases and behaviors.



Automated emails include:

* Order confirmations
* Shipping confirmations
* Birthday greetings
* Abandoned cart notifications
* Welcome messages for new subscribers
* Product abandonment messages
* Cross-sell and upsell messages

**Benefits of email automation**

Automated emailing is a godsend to time-strapped marketers and business owners. Here are several great reasons to automate your emails.

* **It’s Save time** that’s a no-brainer—you can save time by sending automated emails to many subscribers at once. And the best thing is that you can still personalize your campaign content.
* **Make the most of email marketing** you can create automated email workflows to communicate with as many subscribers as you want, at any time you prefer.
* **Increase customer retention** email automation is a great way to rejuvenate the bond with subscribers who haven’t opened your emails or bought from you in a while

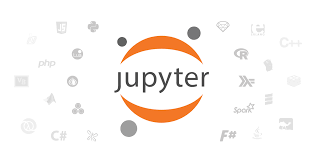
**Libraries used: -**

1. Pandas – for loading our excel spreadsheet from data directory.
2. Message – for formatting email contents.
3. Smtplib - used for email authentication and email transaction (for connecting Gmail account)

**Software Used**

**Jupyter Notebook**

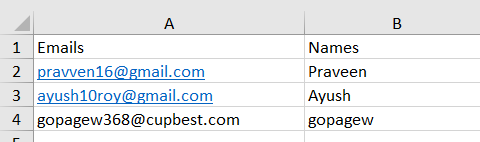
Jupyter Notebook (formerly IPython Notebooks) is a [web-based interactive](https://en.wikipedia.org/wiki/Web_application) computational environment for creating [notebook](https://en.wikipedia.org/wiki/Notebook_interface) documents. A Jupyter Notebook document is a browser-based [REPL](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) containing an ordered list of input/output cells which can contain code, text (using [Markdown](https://en.wikipedia.org/wiki/Markdown)), mathematics, [plots](https://en.wikipedia.org/wiki/Plot_(graphics)) and [rich media](https://en.wikipedia.org/wiki/Interactive_media). Underneath the interface, a notebook is a [JSON](https://en.wikipedia.org/wiki/JSON) document, following a versioned schema, usually ending with the “.ipynb” extension. Jupyter notebooks are built upon a number of popular [open-source](https://en.wikipedia.org/wiki/Open-source_software) libraries.

Jupyter Notebook can connect to many *kernels* to allow programming in different languages. A Jupyter kernel is a program responsible for handling various types of requests ([code execution](https://en.wikipedia.org/wiki/Execution_(computing)), [code completions](https://en.wikipedia.org/wiki/Autocomplete), inspection), and providing a reply. Kernels talk to the other components of Jupyter using [ZeroMQ](https://en.wikipedia.org/wiki/ZeroMQ), and thus can be on the same or [remote machines](https://en.wikipedia.org/wiki/Remote_computer). Unlike many other Notebook-like interfaces, in Jupyter, kernels are not aware that they are attached to a specific document, and can be connected to many clients at once. Usually, kernels allow execution of only a single language, but there are a couple of exceptions. By default, Jupyter Notebook ships with the IPython kernel. As of the 2.3 release (October 2014), there are 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell.

**Result with Analysis**

**Analysis of the code: -**

* **Read the Spreadsheet using the pandas library. The structure of the spreadsheet used in here:**

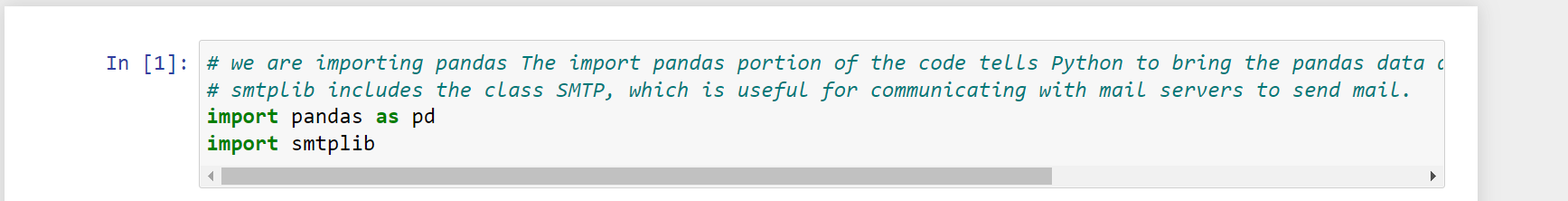
****

* **Establish connection with your Gmail account using smtplib library.**
* **Extract the names and email addresses from the spreadsheet.**
* **Run a loop and for every record send an email.**
* **Close the smtp server.**

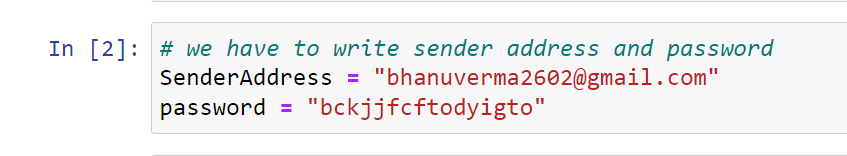
**Screenshots of code: -**

* **Import Libraries**

Import both pandasandsmtplib to read spreadsheets and send emails respectively. smptlib comes pre-installed so you don’t need to do anything but if you haven’t installed pandas then go install pandas via “pip install pandas” in command Prompt.



* **Specify** **Your Credentials**



* **Set Up the Email Server to Send the Email**

We are going to set up the smtp server for Gmail. There are two crucial things to watch out for here:

1. If you are sending the emails from any other email besides Gmail, then you should change the **‘smtp.gmail.com’**in line 13 to the email provider you are using. Here is a [list of some smtp values](https://serversmtp.com/what-is-my-smtp/) based on your email provider

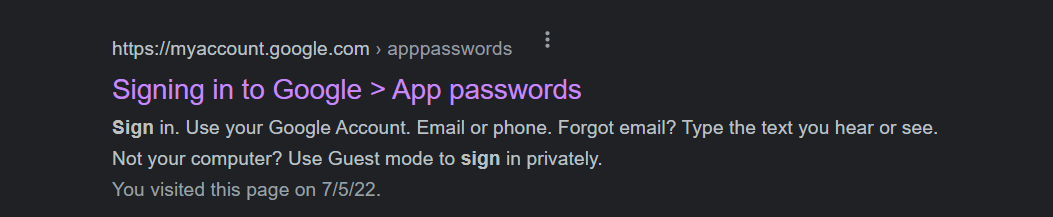
|  |  |  |
| --- | --- | --- |
| **Outlook.com (former Hotmail)** | Outlook.com | Smtp.live.com |
| **Yahoo** | Yahoo.com | Smtp.mail.yahoo.com |

1. Depending on your email provider security setting, the login in line 15 might error out this script. For example, if you are using Gmail you must allow your Gmail account access to a less secure app. [Here is how to disable that option](https://help.dreamhost.com/hc/en-us/articles/115001719551-Troubleshooting-GMAIL-SMTP-authentication-errors).

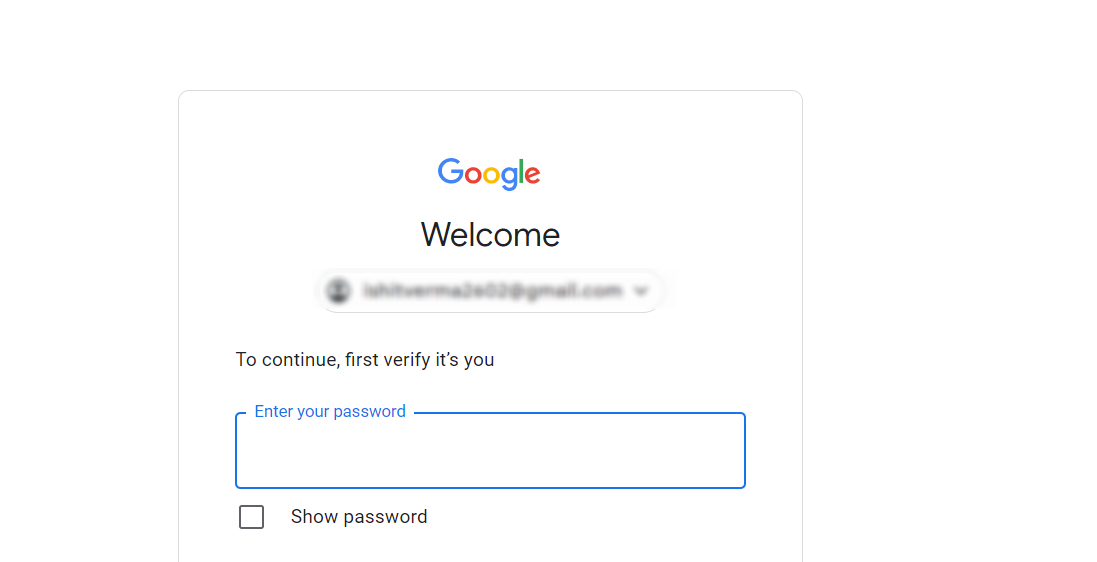
**Graphical user interface, text, application, email

Description automatically generated**

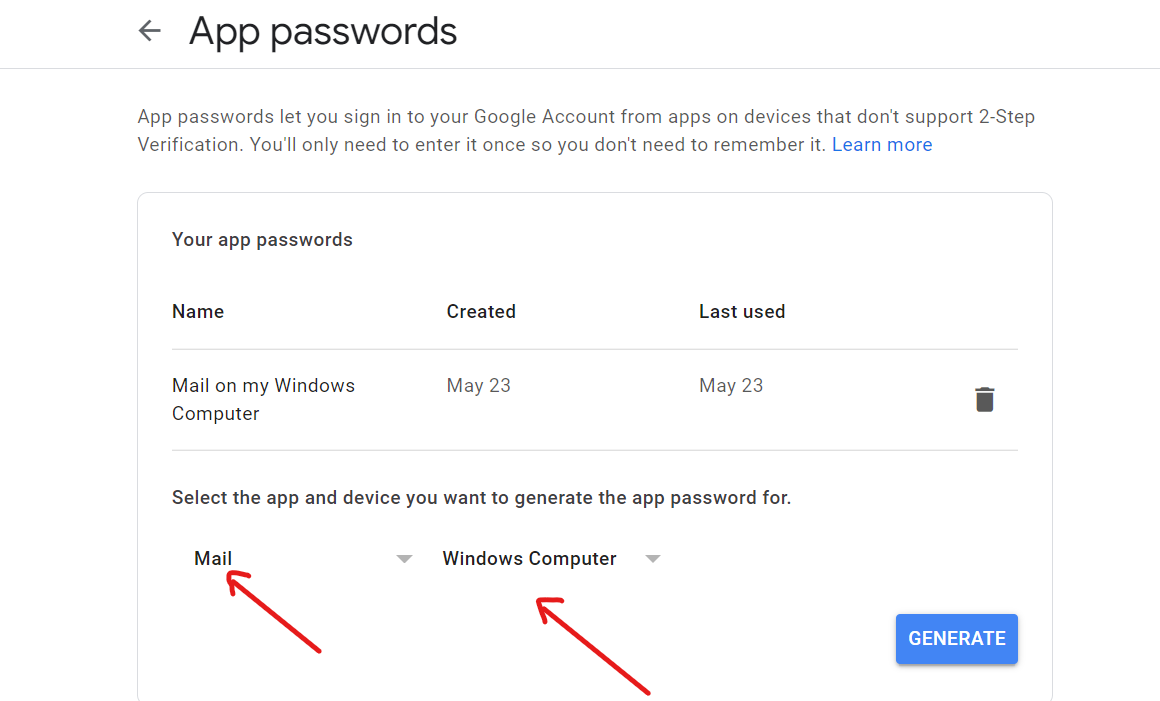
1. And then go to the browser and search for app password and click in this link: -



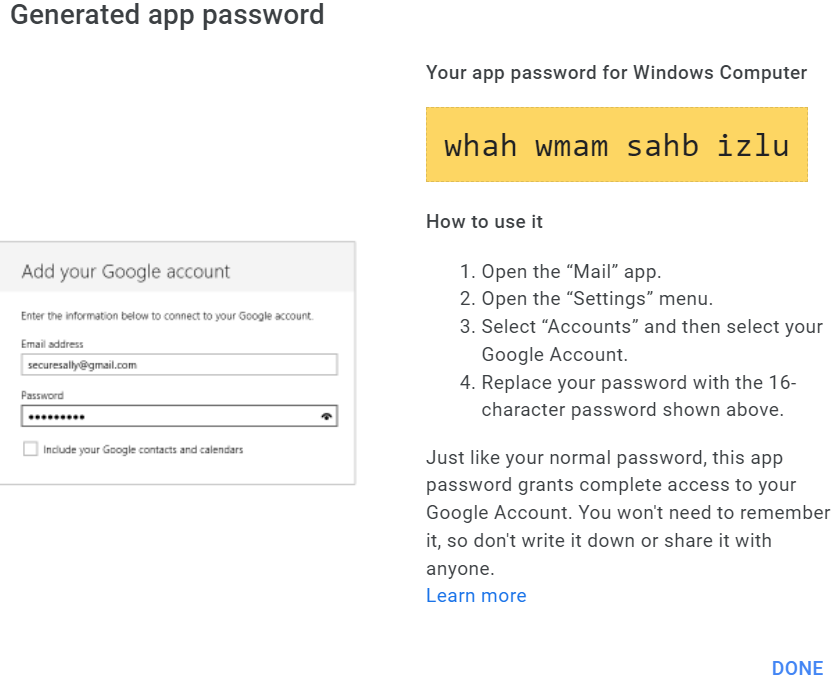
1. Then this site is showing to login with your password



1. Then you have to **Select > Mail** and then **Select** **> Your device (I am using windows computer that’s why I select that option)** and then click on **Generate**

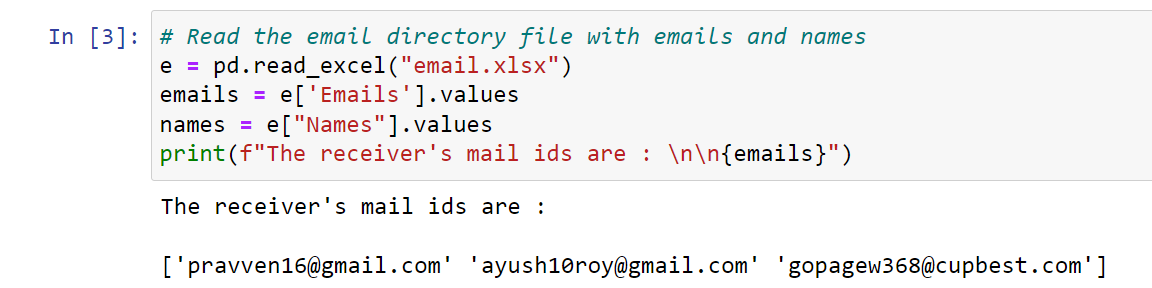


1. And then it will generate your password and use this password to send an email.



# **Read the File**

We are going to be reading the spreadsheet using pandas and storing the result in a data frame email list



* **Get all the Values from the File**

In pandas, the way to read columns is to access them as follows. In this case, we are going to access all four of our columns, Name, Email, Subject, Message and store them in variables. If you have any other columns that you need to get, you can use the same syntax to get those values.

A picture containing application

Description automatically generateddf

* **Send the Email**

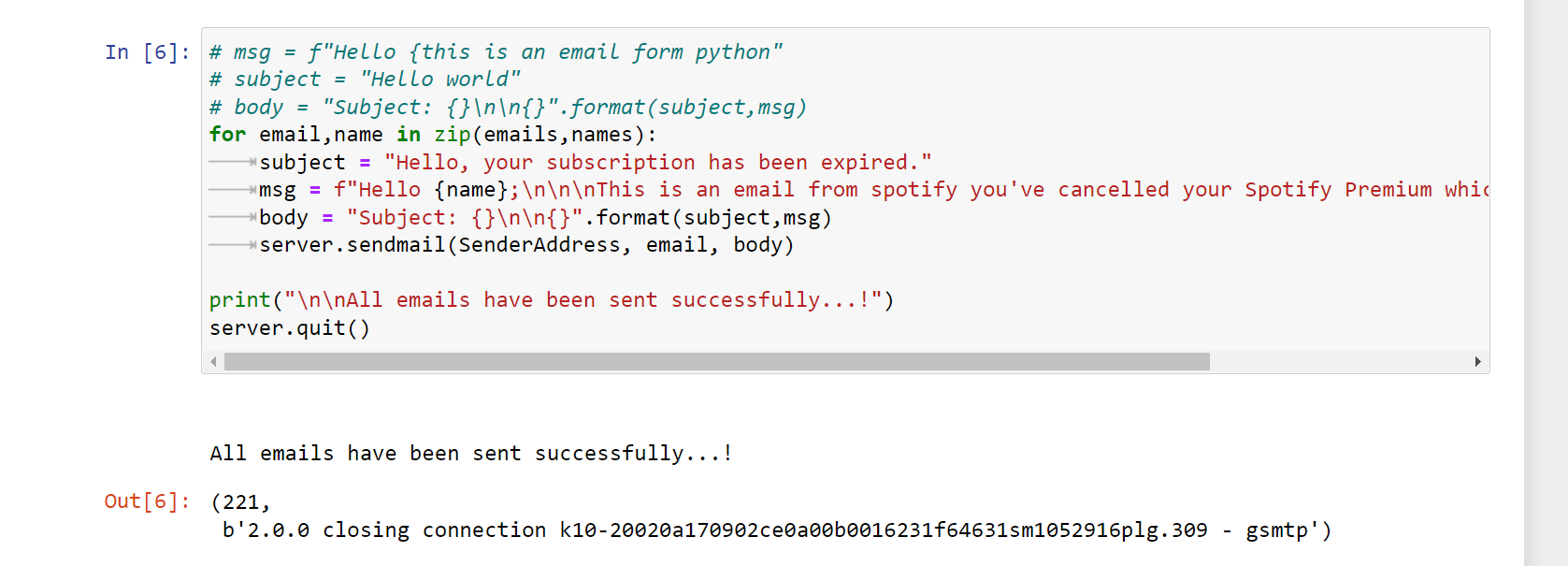
Send the email using smtplib.SMTP server and port “587” via login using your gmail account and password.

Text

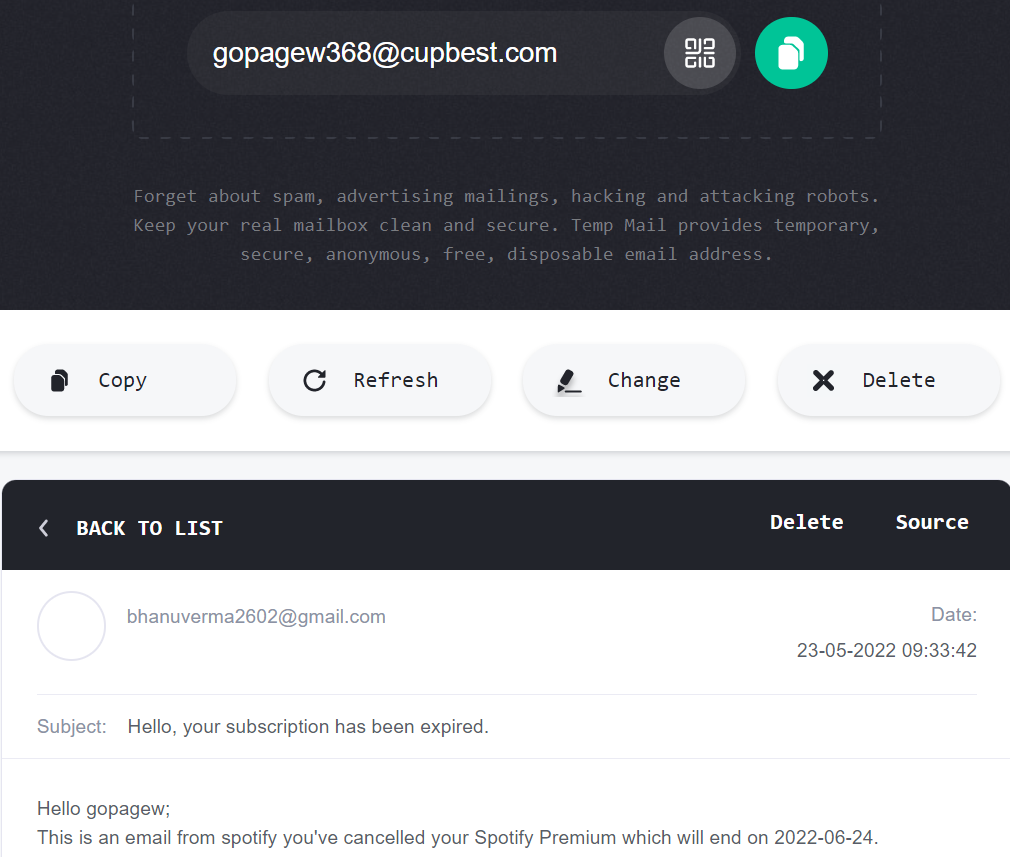
Description automatically generated

# **Create the Email to Send & Close the Server**

This is the meat of our code where everything comes together. We will create the email that we want to send. The *{}*notation specifies that we want there to be a placeholder. The value inside the bracket specifies which variable should be filled in depending on its location inside the *format*method.



**Output: -**



**Project Outcome: -**

From this project, we learned to describe how email automation works and we learned about email automation enables you to send hyper-personalized emails based on the user’s actions and preferences. We also learned about how we can send emails and attachments to multiple audiences using python and excel sheets.

**Project Conclusion: -**

From this project, we gained the knowledge of software – Jupyter Notebook.We learned about various libraries in python.We learned about various ways for sending an email using pyhton and excel sheets.