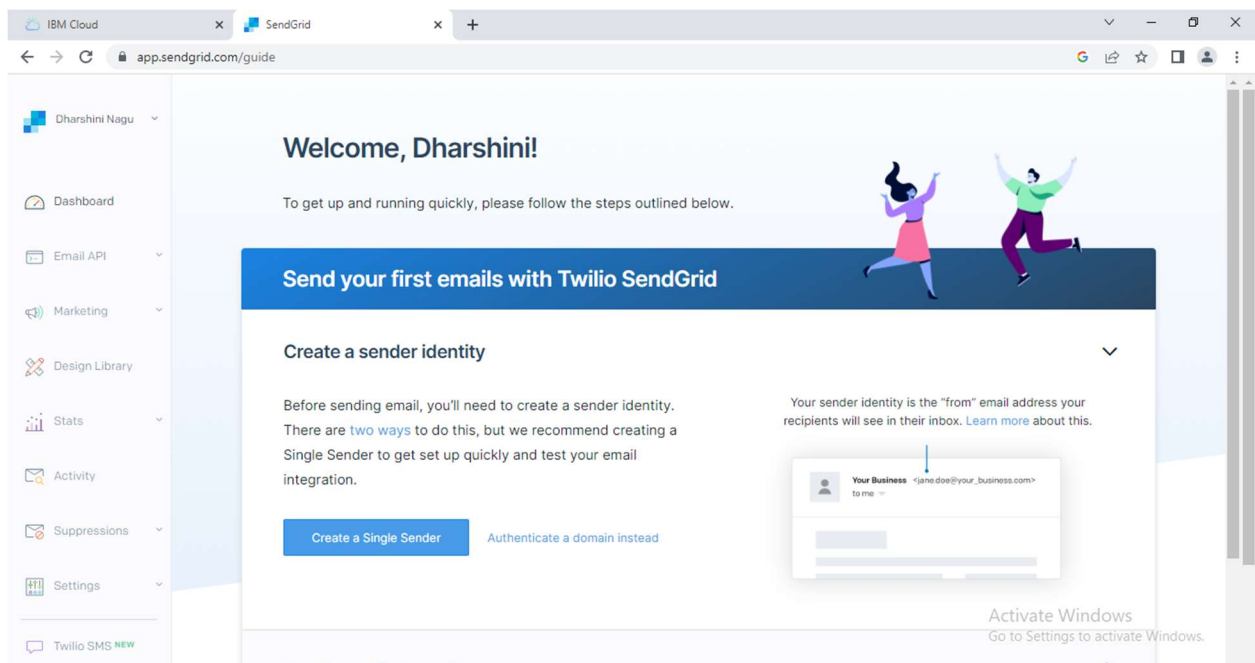


SendGrid Integration with Python Code

Team ID	PNT2022TMID48233
Project Name	PERSONAL EXPENSE TRACKER
Date	15-11-2022

Step 1: Creating account in SendGrid



Step 2:

The screenshot shows the SendGrid web interface. The browser tab is labeled 'SendGrid' and the address bar shows 'mc.sendgrid.com/dynamic-templates'. The left sidebar contains a user profile 'Dharshini Nagu' and a menu with items: Dashboard, Email API, Dynamic Templates (selected), Integration Guide, Marketing, Design Library, Stats, Activity, Suppressions, and Settings. The main content area is titled 'Dynamic Templates' and features a large box with the heading 'Create your first dynamic template'. Below this heading is a sub-header 'Dynamic email templates allow you to easily build, personalize and manage emails that can be sent using an API call.' and a blue button labeled 'Create a Dynamic Template'. At the bottom of the main area, there is a link: 'Need Help? Learn more about creating dynamic templates.' In the bottom right corner, there is a Windows watermark: 'Activate Windows Go to Settings to activate Windows.'

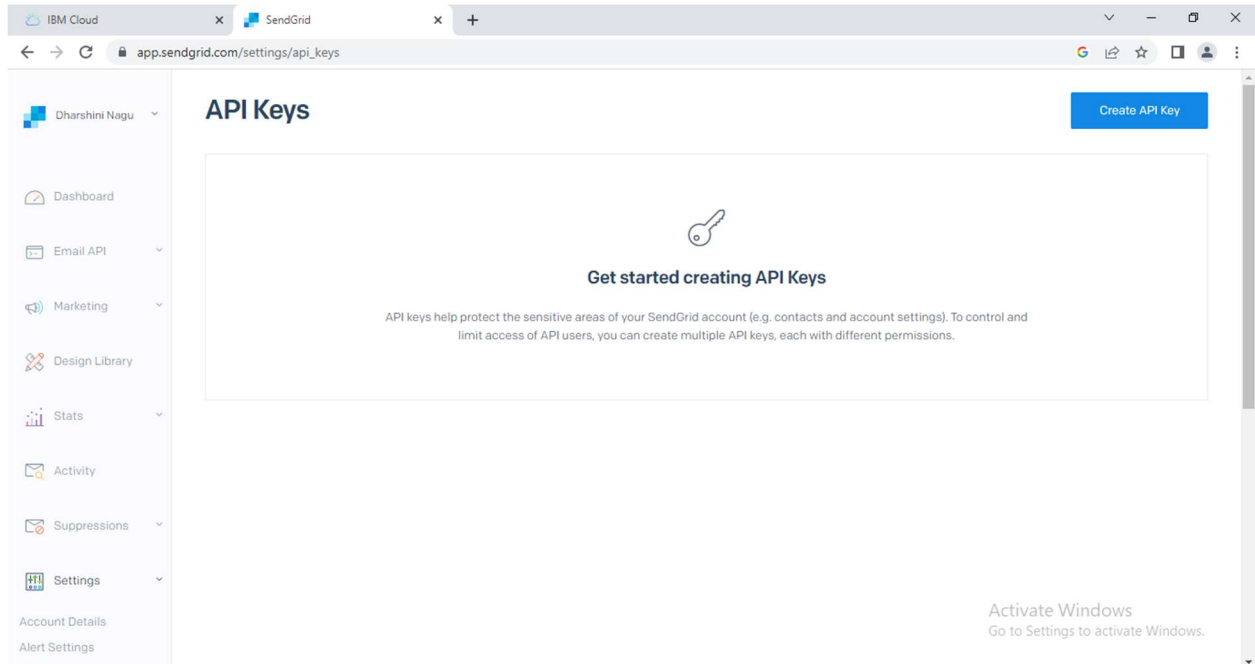
Step 3:

The screenshot shows the SendGrid dashboard. The left sidebar is identical to the previous screenshot. The main content area has a blue banner at the top with the text 'Ready to unlock the full potential of Twilio SendGrid?' and 'Finish your account setup in order to select a paid Email API or Marketing Campaigns plan.' with a 'Let's do it!' button. Below the banner, it says 'Hello Dharshini! Here's your recent email activity.' with a toggle switch set to 'Wk'. Below this is a table of email activity metrics:

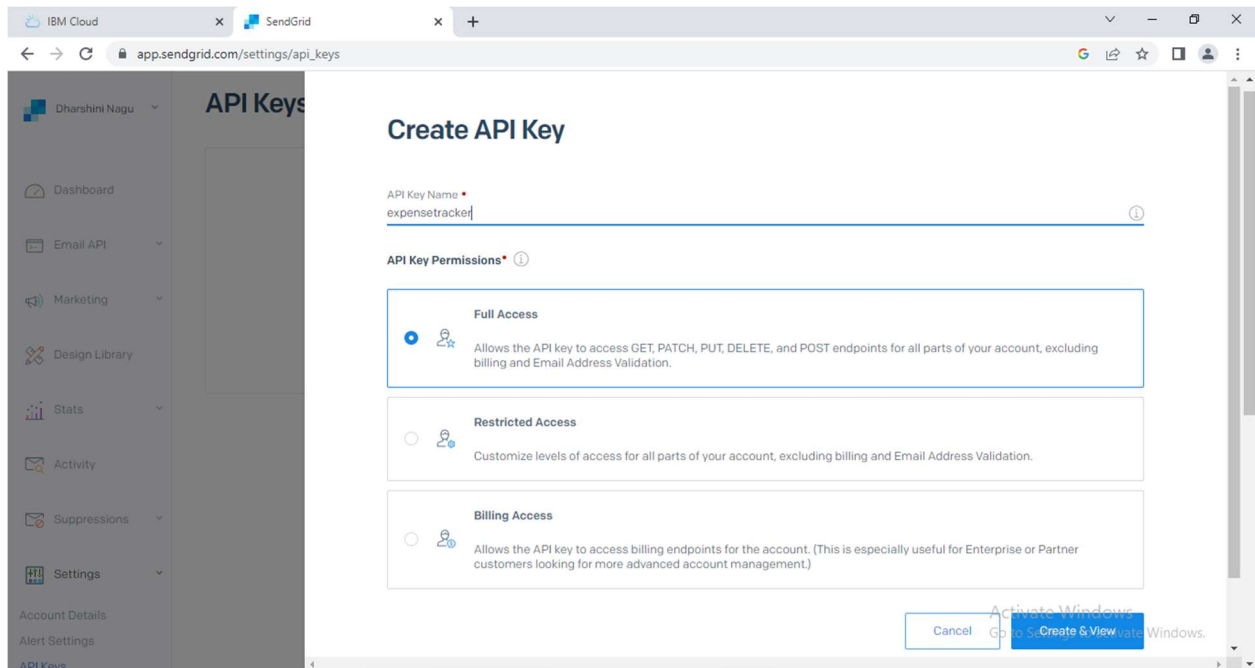
REQUESTS	DELIVERED	OPENED	CLICKED	BOUNCES	SPAM REPORTS
0	N/A 0	N/A 0	N/A 0	N/A 0	N/A 0

At the bottom right, there is a Windows watermark: 'Activate Windows Go to Settings to activate Windows.'

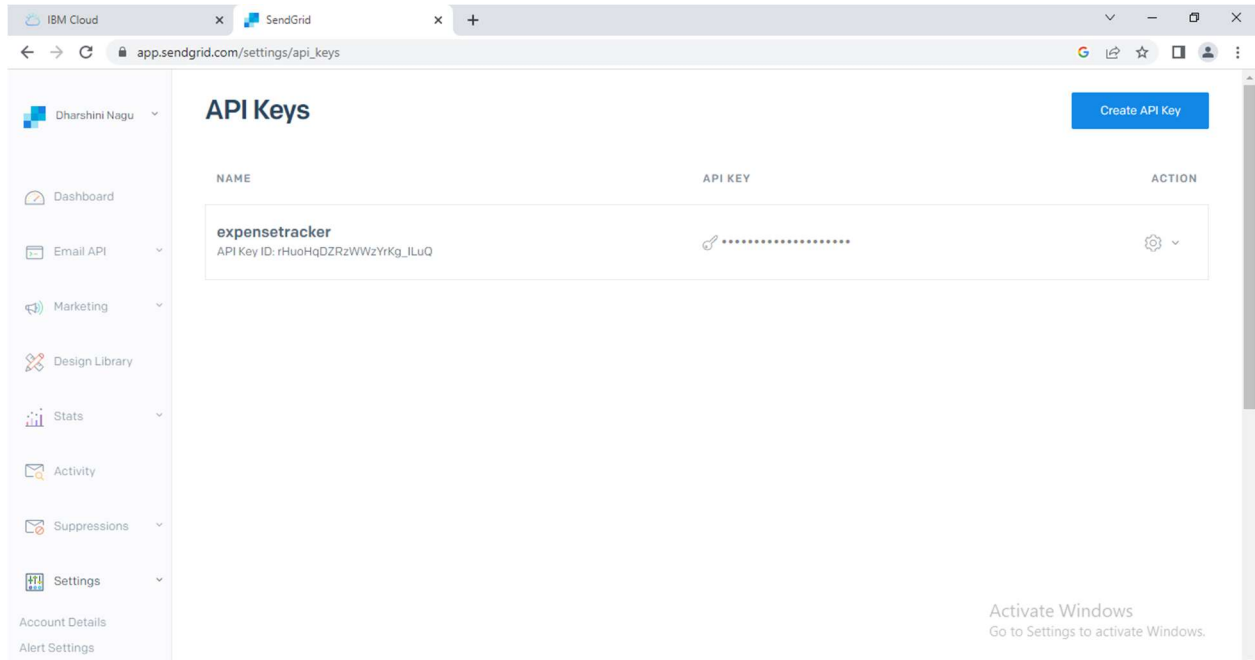
Step 4: Create API Key



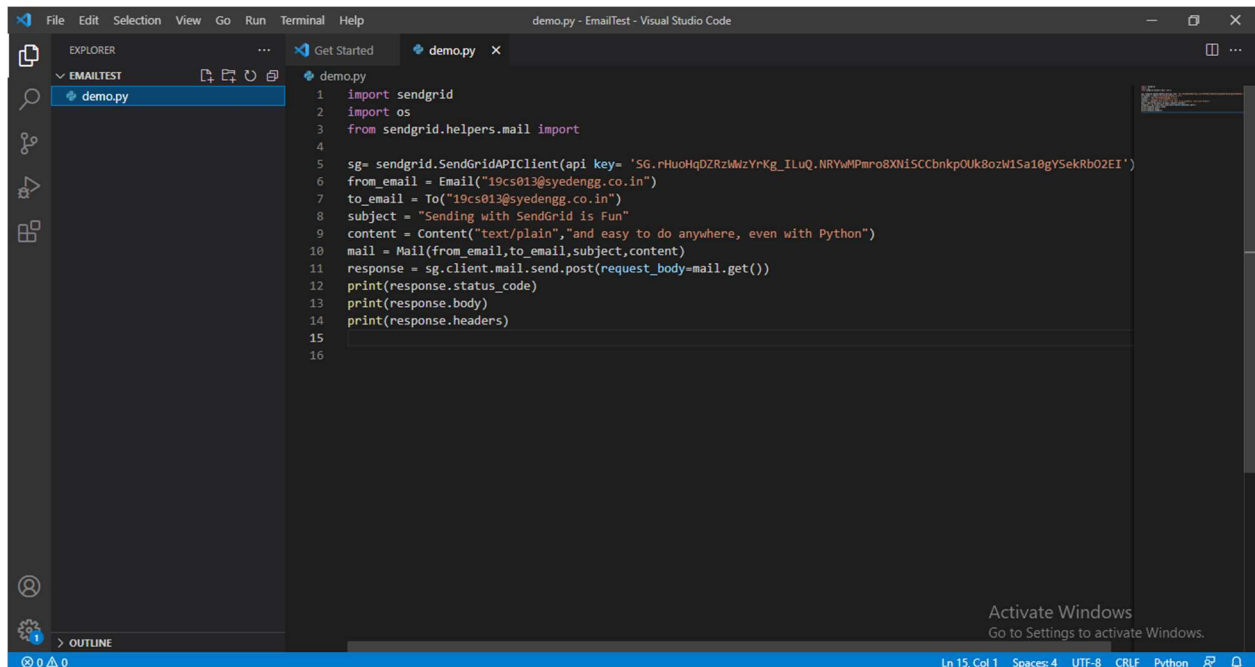
Step 5: API Key Name - expensetracker



Step 6: API Key “expensetracker” is created



Step 7: Creating a Virtual Environment “EmailTest” and creating “demo.py” file with the following code inside the virtual environment.



Step 8: Running the Python file through the terminal.

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'JUPYTER'. The terminal displays the output of a Python script, which is a string of raw HTTP response data. The data starts with 'b'' and includes headers such as 'Server: nginx', 'Date: Tue, 15 Nov 2022 07:57:45 GMT', 'Content-Length: 0', 'Connection: close', 'X-Message-Id: UfsTvTgqSoCKPyCSqdILZw', 'Access-Control-Allow-Origin: https://sendgrid.api-docs.io', 'Access-Control-Allow-Methods: POST', 'Access-Control-Allow-Headers: Authorization, Content-Type, On-behalf-of, x-sg-elas-acl', 'Access-Control-Max-Age: 600', 'X-No-CORS-Reason: https://sendgrid.com/docs/Classroom/Basics/API/cors.html', and 'Strict-Transport-Security: max-age=600; includeSubDomains'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
b''
Server: nginx
Date: Tue, 15 Nov 2022 07:57:45 GMT
Content-Length: 0
Connection: close
X-Message-Id: UfsTvTgqSoCKPyCSqdILZw
Access-Control-Allow-Origin: https://sendgrid.api-docs.io
Access-Control-Allow-Methods: POST
Access-Control-Allow-Headers: Authorization, Content-Type, On-behalf-of, x-sg-elas-acl
Access-Control-Max-Age: 600
X-No-CORS-Reason: https://sendgrid.com/docs/Classroom/Basics/API/cors.html
Strict-Transport-Security: max-age=600; includeSubDomains
```

Step 9: Finally, Test mail is received via SendGrid.