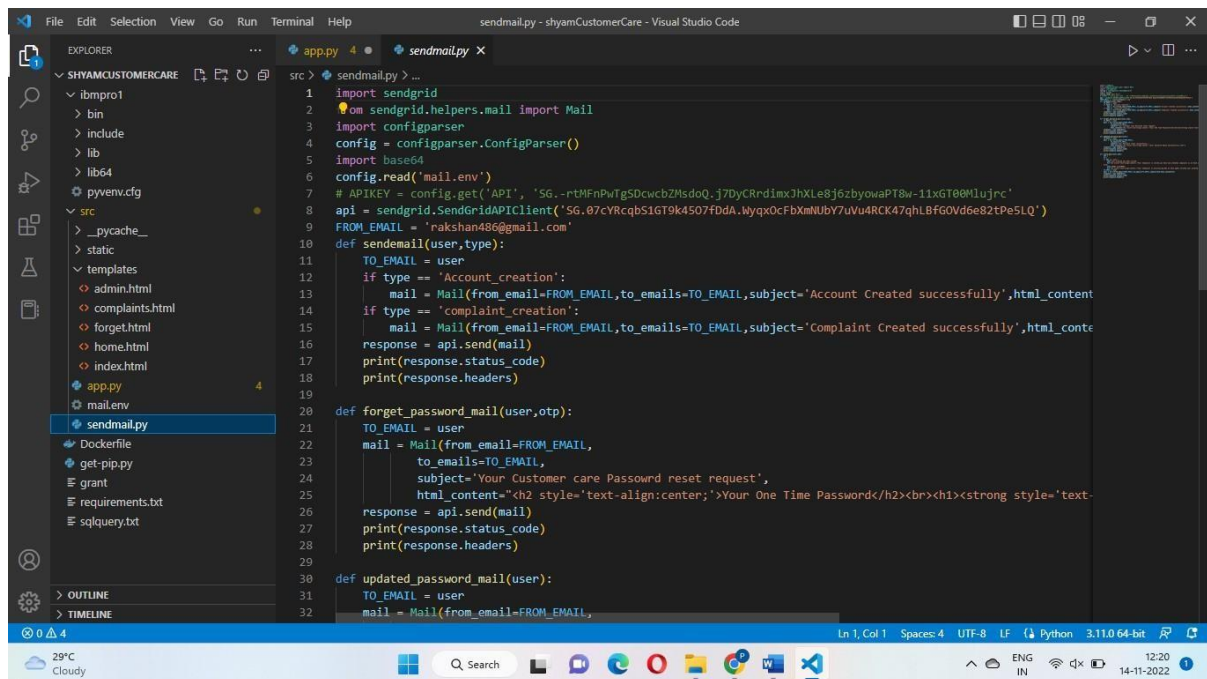


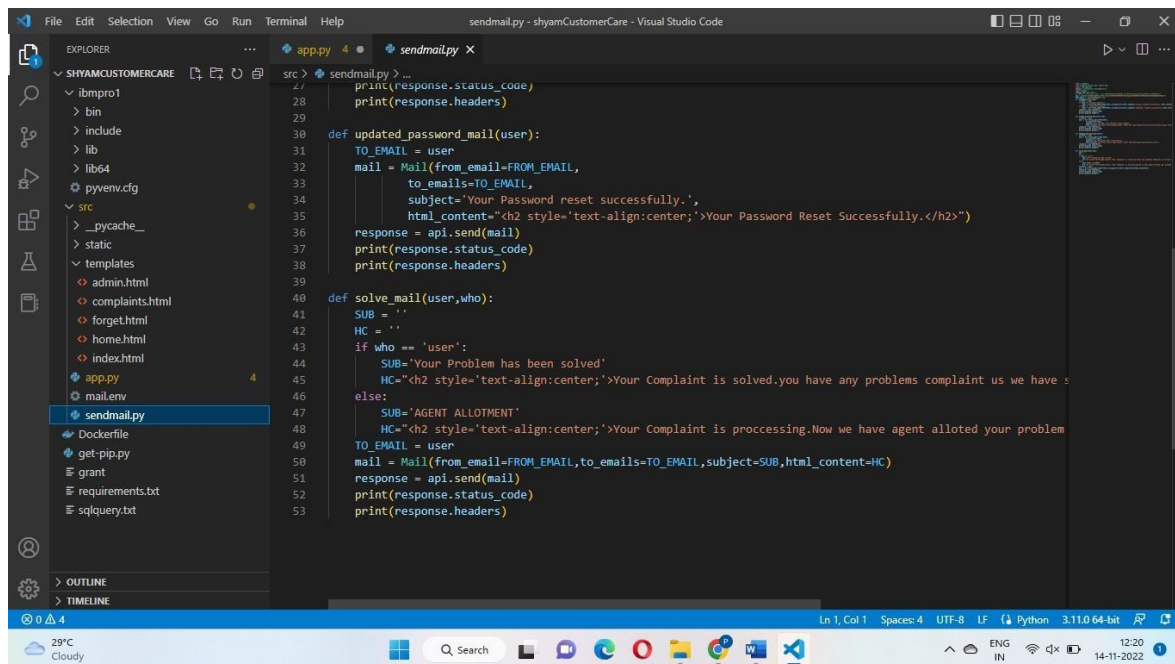
SendGrid integration with python code

| | |
|--------------|---------------------------------|
| Date | 17 November 2022 |
| Team ID | PNT2022TMID43019 |
| Project Name | Nutrition Assistant Application |



The screenshot shows the Visual Studio Code editor with a Python file named `sendmail.py`. The code defines a `sendmail` function that takes a user and a type as arguments. It uses the `sendgrid` library to send emails. The code includes comments for API keys and email addresses. The `sendmail` function has two main branches: one for account creation and one for complaint creation. The `forget_password_mail` function is also defined, which sends a password reset request. The `updated_password_mail` function is partially visible at the bottom.

```
src > sendmail.py > ...
1 import sendgrid
2 from sendgrid.helpers.mail import Mail
3 import configparser
4 config = configparser.ConfigParser()
5 import base64
6 config.read('mail.env')
7 # APIKEY = config.get('API', 'SG.-rtMFnPwTgSDcwbZMsdQ.j7DyCRdimxJhXLe8j6zyowaPT8w-11xGT08Mlujrc')
8 api = sendgrid.SendGridAPIClient('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')
9 FROM_EMAIL = 'rakshan486@gmail.com'
10 def sendmail(user,type):
11     TO_EMAIL = user
12     if type == 'Account_creation':
13         mail = Mail(from_email=FROM_EMAIL,to_emails=TO_EMAIL,subject='Account Created successfully',html_content
14     if type == 'complaint_creation':
15         mail = Mail(from_email=FROM_EMAIL,to_emails=TO_EMAIL,subject='Complaint Created successfully',html_conte
16     response = api.send(mail)
17     print(response.status_code)
18     print(response.headers)
19
20 def forget_password_mail(user,otp):
21     TO_EMAIL = user
22     mail = Mail(from_email=FROM_EMAIL,
23               to_emails=TO_EMAIL,
24               subject='Your Customer care Passowrd reset request',
25               html_content="<h2 style='text-align:center;'>Your One Time Password</h2><br><h1><strong style='text-
26     response = api.send(mail)
27     print(response.status_code)
28     print(response.headers)
29
30 def updated_password_mail(user):
31     TO_EMAIL = user
32     mail = Mail(from_email=FROM_EMAIL,
```



The screenshot shows the continuation of the `sendmail.py` file. It includes the `updated_password_mail` function, which sends a password reset confirmation email. The `solve_mail` function is also defined, which sends a response email based on the user's complaint. The code uses the `sendgrid` library to send emails. The `solve_mail` function has two main branches: one for user problems and one for agent allotment.

```
27     print(response.status_code)
28     print(response.headers)
29
30 def updated_password_mail(user):
31     TO_EMAIL = user
32     mail = Mail(from_email=FROM_EMAIL,
33               to_emails=TO_EMAIL,
34               subject='Your Password reset successfully.',
35               html_content="<h2 style='text-align:center;'>Your Password Reset Successfully.</h2>")
36     response = api.send(mail)
37     print(response.status_code)
38     print(response.headers)
39
40 def solve_mail(user,who):
41     SUB = ''
42     HC = ''
43     if who == 'user':
44         SUB='Your Problem has been solved'
45         HC="<h2 style='text-align:center;'>Your Complaint is solved.you have any problems complaint us we have s
46     else:
47         SUB='AGENT ALLOTMENT'
48         HC="<h2 style='text-align:center;'>Your Complaint is processing.Now we have agent allotted your problem
49     TO_EMAIL = user
50     mail = Mail(from_email=FROM_EMAIL,to_emails=TO_EMAIL,subject=SUB,html_content=HC)
51     response = api.send(mail)
52     print(response.status_code)
53     print(response.headers)
```

