

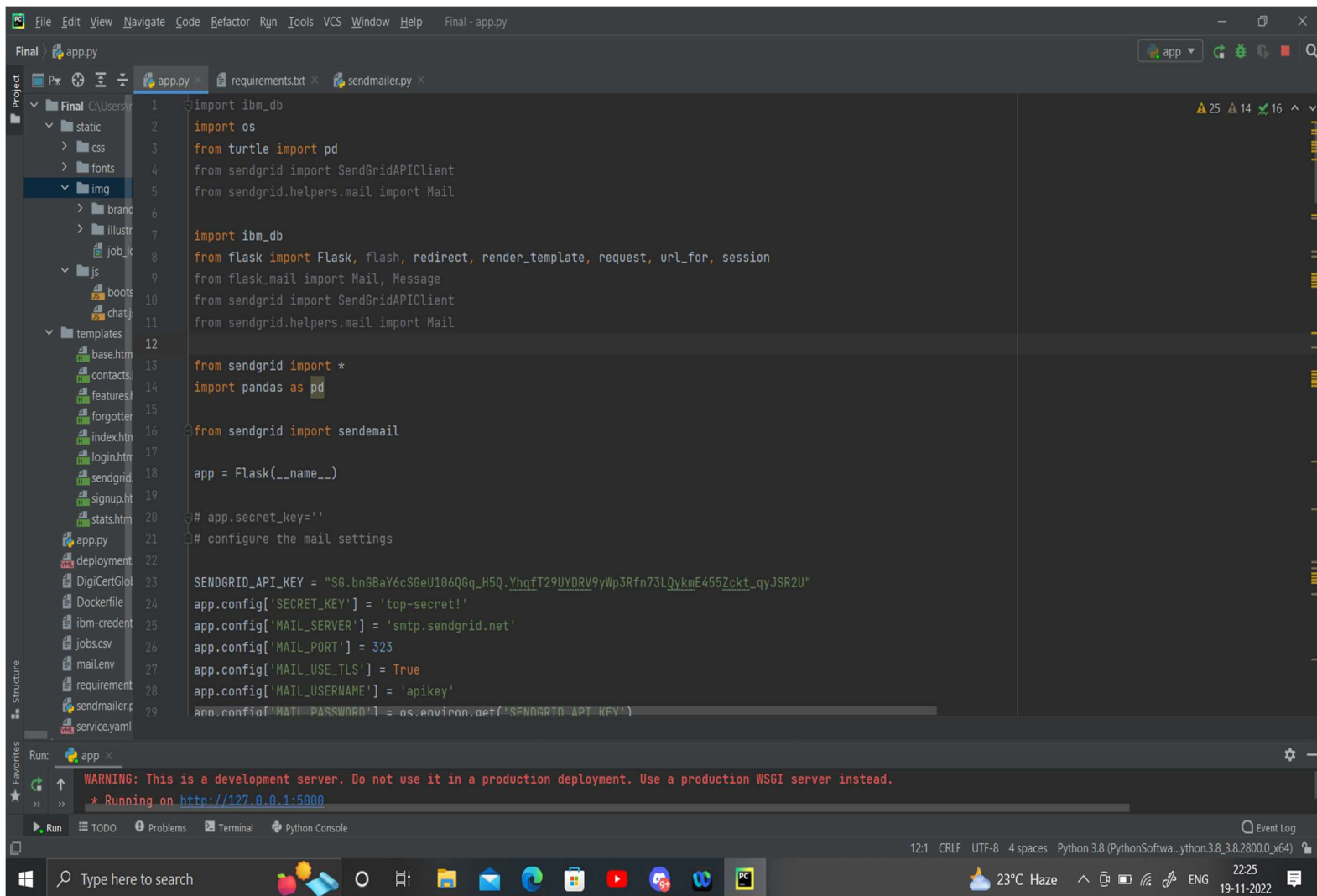
PROJECT DEVELOPMENT PHASE

Integrating SendGrid Service

SPRINT-2

Date	05 November 2022
Team ID	PNT2022TMID09785
Project Name	Skill / Job Recommender Application

Importing the required libraries for SendGrid

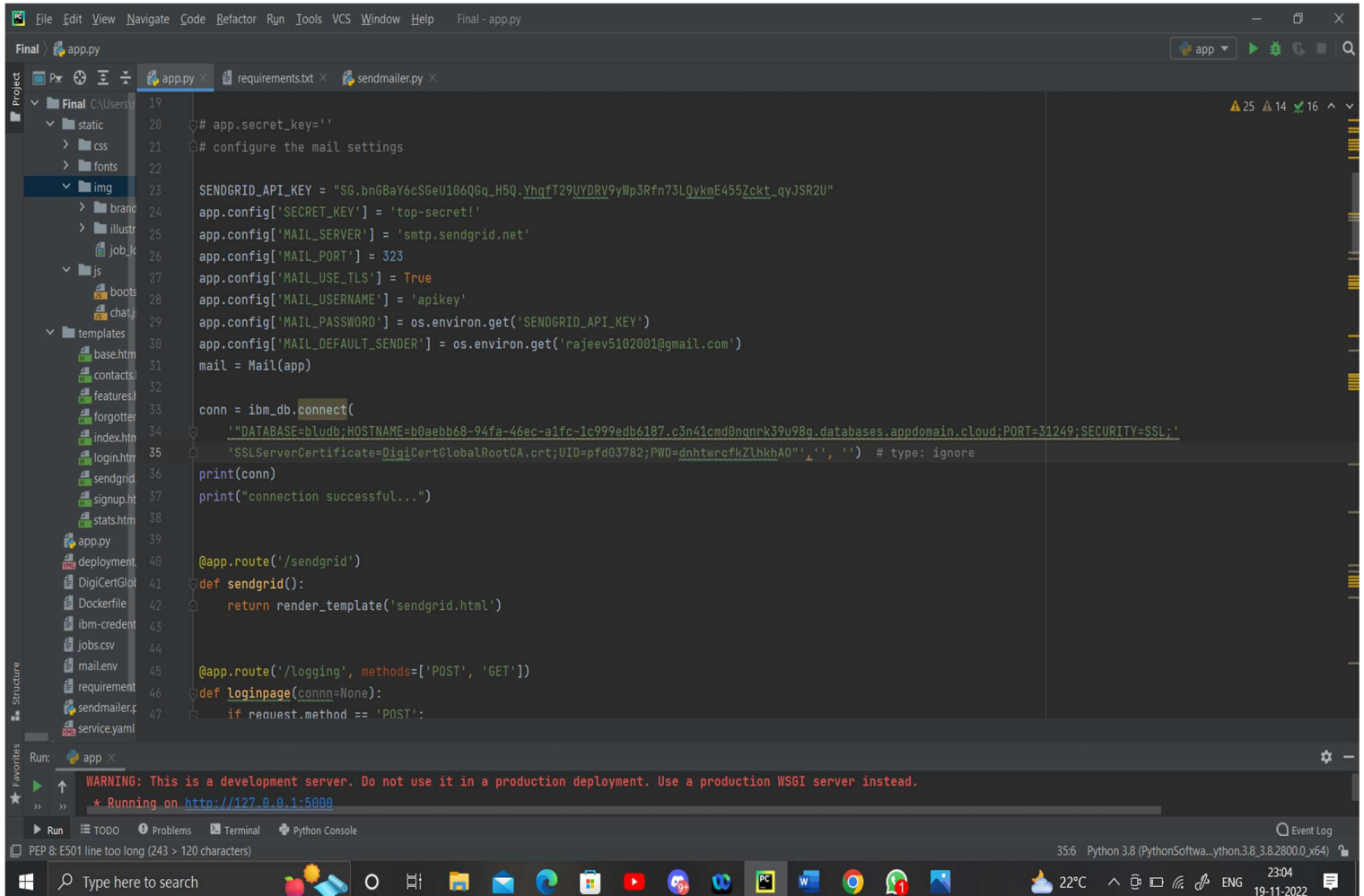


```
1 import ibm_db
2 import os
3 from turtle import pd
4 from sendgrid import SendGridAPIClient
5 from sendgrid.helpers.mail import Mail
6
7 import ibm_db
8 from flask import Flask, flash, redirect, render_template, request, url_for, session
9 from flask_mail import Mail, Message
10 from sendgrid import SendGridAPIClient
11 from sendgrid.helpers.mail import Mail
12
13 from sendgrid import *
14 import pandas as pd
15
16 from sendgrid import sendemail
17
18 app = Flask(__name__)
19
20 # app.secret_key=''
21 # configure the mail settings
22
23 SENDGRID_API_KEY = "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"
24 app.config['SECRET_KEY'] = 'top-secret!'
25 app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
26 app.config['MAIL_PORT'] = 323
27 app.config['MAIL_USE_TLS'] = True
28 app.config['MAIL_USERNAME'] = 'apikey'
29 app.config['MAIL_PASSWORD'] = os.environ.get('SENDGRID_API_KEY')
```

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on <http://127.0.0.1:5000>

Configuring the SendGrid and mail settings



The screenshot shows a code editor with a Python file named `app.py`. The code is configured to use SendGrid for email sending. It includes the following configuration:

```
19
20 # app.secret_key='
21 # configure the mail settings
22
23 SENDGRID_API_KEY = "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"
24 app.config['SECRET_KEY'] = 'top-secret!'
25 app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
26 app.config['MAIL_PORT'] = 323
27 app.config['MAIL_USE_TLS'] = True
28 app.config['MAIL_USERNAME'] = 'apikey'
29 app.config['MAIL_PASSWORD'] = os.environ.get('SENDGRID_API_KEY')
30 app.config['MAIL_DEFAULT_SENDER'] = os.environ.get('rajeev5102001@gmail.com')
31 mail = Mail(app)
32
33 conn = ibm_db.connect(
34     "DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnk39u98g.databases.apdomain.cloud;PORT=31249;SECURITY=SSL;"
35     'SSLCertificate=DigiCertGlobalRootCA.crt;UID=pf03782;PWD=dnhtwrcfkZLhkhA0"', '' ) # type: ignore
36
37 print(conn)
38 print("connection successful...")
39
40 @app.route('/sendgrid')
41 def sendgrid():
42     return render_template('sendgrid.html')
43
44
45 @app.route('/logging', methods=['POST', 'GET'])
46 def loginpage(conn=None):
47     if request.method == 'POST':
```

The code also includes a database connection configuration for IBM Db2 Cloud. The `conn` variable is used to connect to the database. The `sendgrid` function is a route handler that renders the `sendgrid.html` template. The `loginpage` function is a route handler that handles POST and GET requests.

The IDE interface shows the project structure on the left, with the `app.py` file selected. The bottom status bar indicates the code is running on `http://127.0.0.1:5000`. A warning message is displayed: "WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead."

Contact Us page

127.0.0.1

Contacts - Dream Team Recruitment

127.0.0.1:5000/contacts

Home Discover Contact Us Logout

Got any questions?

Our team is always here to help. Send us a message and we'll get back to you shortly.

Name

Email

Message

Send

Type here to search

22°C

ENG

23:08 19-11-2022

Password reset page

