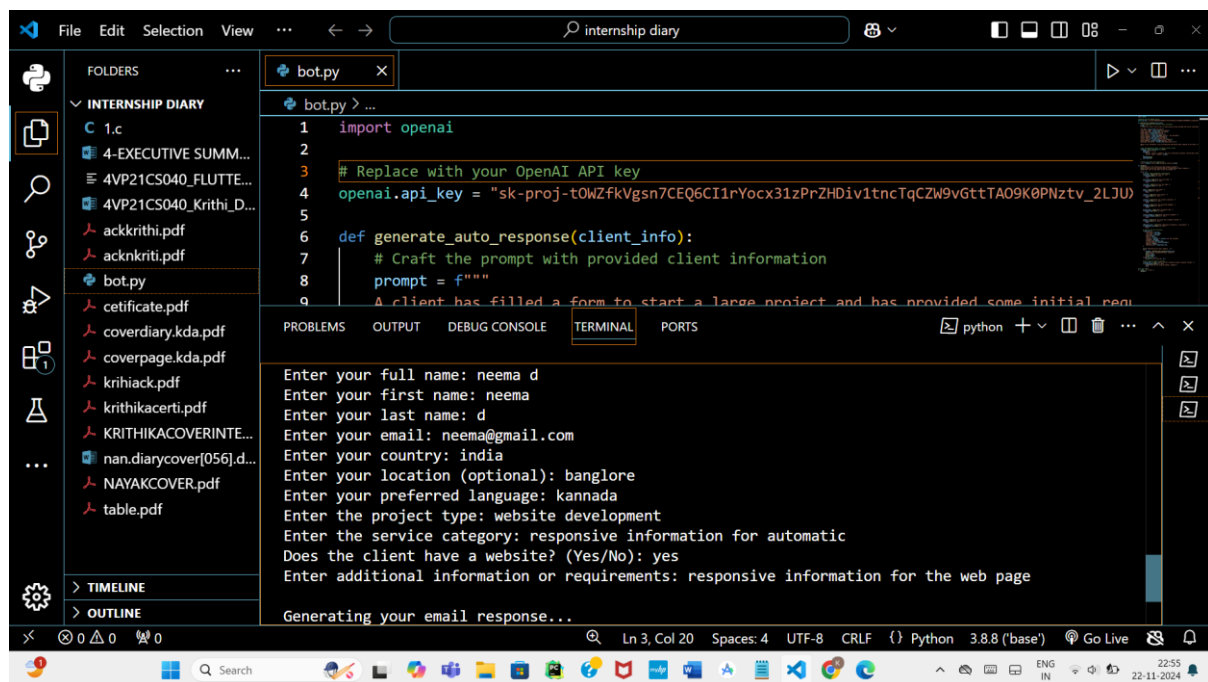


TASK 1: SMART RESPONSE

The code provided for generating email responses using OpenAI's API has some issues that need to be fixed to work properly. First, the API key is written directly in the code, which is unsafe because others could misuse it. To fix this, the API key should be stored securely in environment variables or a separate file. Next, the code doesn't check if the inputs are valid or missing, which can cause problems. This can be solved by validating the inputs and providing default values for optional ones. The repeated checks for "exit" after every input make the code harder to read and use. A better way is to create a function to handle user input and check for "exit" in one place.

If the user gives a lot of extra information, the text sent to OpenAI might become too long and cause errors. This can be fixed by limiting the length of extra information. The code also doesn't handle errors from the API properly, like if the network fails or the API key is incorrect. Adding error handling can prevent the program from crashing and show helpful messages instead. Additionally, the code could run into errors if it tries to access missing data in the input, which can be avoided by using a safe method to fetch data with default values. Finally, after generating an email, the program should ask the user if they want to create another or exit. These changes will make the code easier to use, safer, and able to handle errors better.

output:



The screenshot shows a VS Code editor window with a file explorer on the left and a terminal at the bottom. The file explorer shows a folder named 'INTERNSHIP DIARY' containing several PDF files and a Python file named 'bot.py'. The 'bot.py' file is open in the editor, showing the following code:

```
1 import openai
2
3 # Replace with your OpenAI API key
4 openai.api_key = "sk-proj-tOWZfkVgsn7CEQ6CI1rYocx31zPrZHDv1tncTqCZW9vGttTA09K0PNztv_2LJU"
5
6 def generate_auto_response(client_info):
7     # Craft the prompt with provided client information
8     prompt = f"""
9     A client has filled a form to start a large project and has provided some initial req
```

The terminal window shows the output of the program, which prompts the user for various pieces of information and then generates an email response:

```
Enter your full name: neema d
Enter your first name: neema
Enter your last name: d
Enter your email: neema@gmail.com
Enter your country: india
Enter your location (optional): banglore
Enter your preferred language: kannada
Enter the project type: website development
Enter the service category: responsive information for automatic
Does the client have a website? (Yes/No): yes
Enter additional information or requirements: responsive information for the web page

Generating your email response...
```

