Building a Terminal-Based Chat Application with a Given Dataset

Introduction:

In this document, we will guide you through the creation of a basic terminal-based chat application using a provided dataset. Instead of real-time user interactions, we will simulate a chat conversation using pre-defined messages. This project can serve as a foundation for more complex chat applications or for testing and experimenting with chat-related features.

Prerequisites:

Before you begin, ensure you have the following:

- Python (3.x) installed on your system.
- A provided dataset of chat messages.
- Basic knowledge of Python programming.
- Terminal or Command Prompt

Implementation:

Follow these steps to create a basic terminal-based chat application using a given dataset:

1.Create Project Directory:

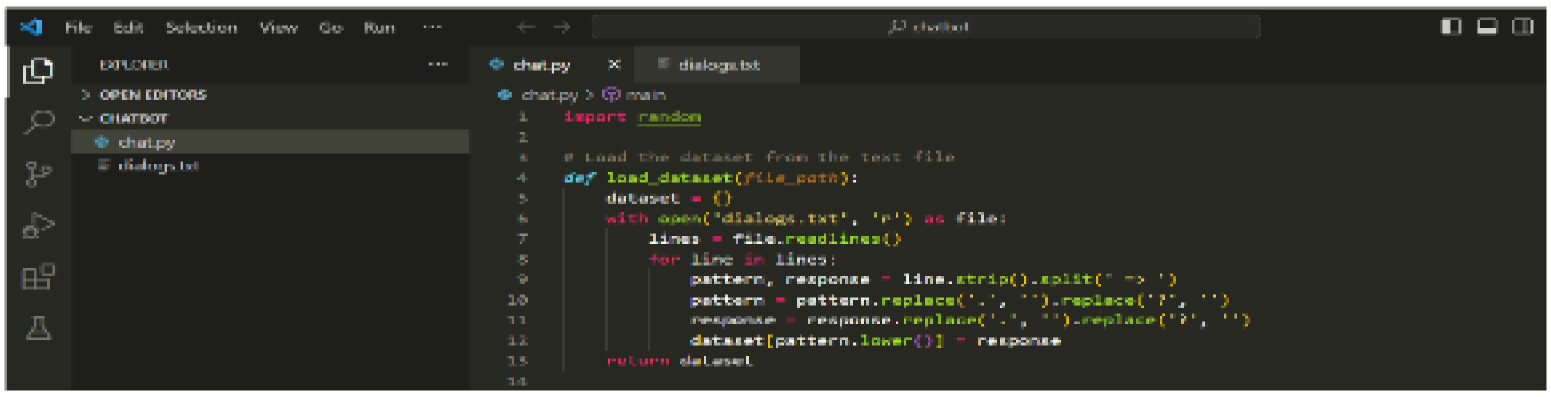
Start by creating a project directory for your application.

2. Create Python Script:

Create a Python script for the chat application. You can use your favorite code editor to create a file, e.g., chat_app.py

3. Import Libraries:

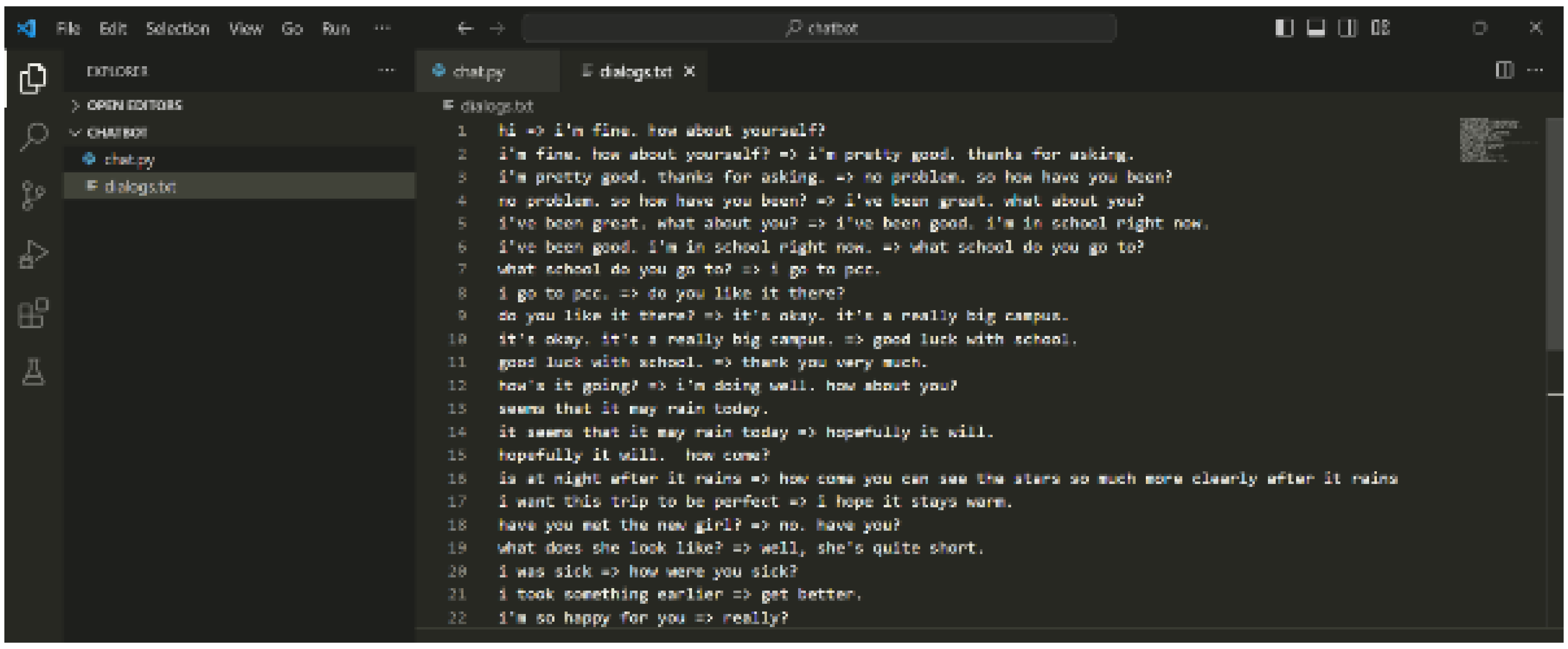
Import the necessary libraries at the beginning of your Python script.



```python
import random

# Load the dataset from the text file
def load_dataset(file_path):
    dataset = {}
    with open('dialogs.txt', 'r') as file:
        lines = file.readlines()
        for line in lines:
            pattern, response = line.strip().split(' => ')
            pattern = pattern.replace('.', '').replace('?', '')
            response = response.replace('.', '').replace('?', '')
            dataset[pattern.lower()] = response
    return dataset
```

4.Load the Dataset:

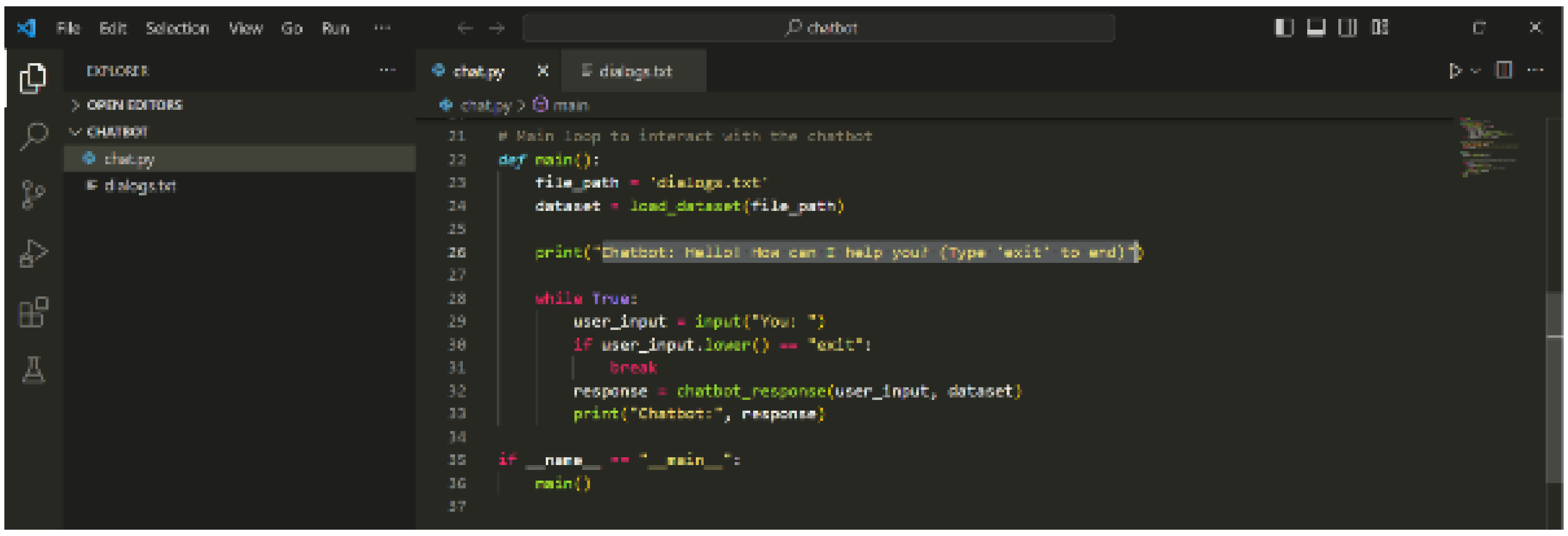
Load the provided dataset into your Python script. You can store the messages in a list or a data structure of your choice



```
hi => i'm fine. how about yourself?
i'm fine. how about yourself? => i'm pretty good. thanks for asking.
i'm pretty good. thanks for asking. => no problem. so how have you been?
no problem. so how have you been? => i've been great. what about you?
i've been great. what about you? => i've been good. i'm in school right now.
i've been good. i'm in school right now. => what school do you go to?
what school do you go to? => i go to pcc.
i go to pcc. => do you like it there?
do you like it there? => it's okay. it's a really big campus.
it's okay. it's a really big campus. => good luck with school.
good luck with school. => thank you very much.
how's it going? => i'm doing well. how about you?
seems that it may rain today.
it seems that it may rain today => hopefully it will.
hopefully it will.  how come?
is at night after it rains => how come you can see the stars so much more clearly after it rains
i want this trip to be perfect => i hope it stays warm.
have you met the new girl? => no. have you?
what does she look like? => well, she's quite short.
i was sick => how were you sick?
i took something earlier => get better.
i'm so happy for you => really?
```

5.Simulate the Chat:

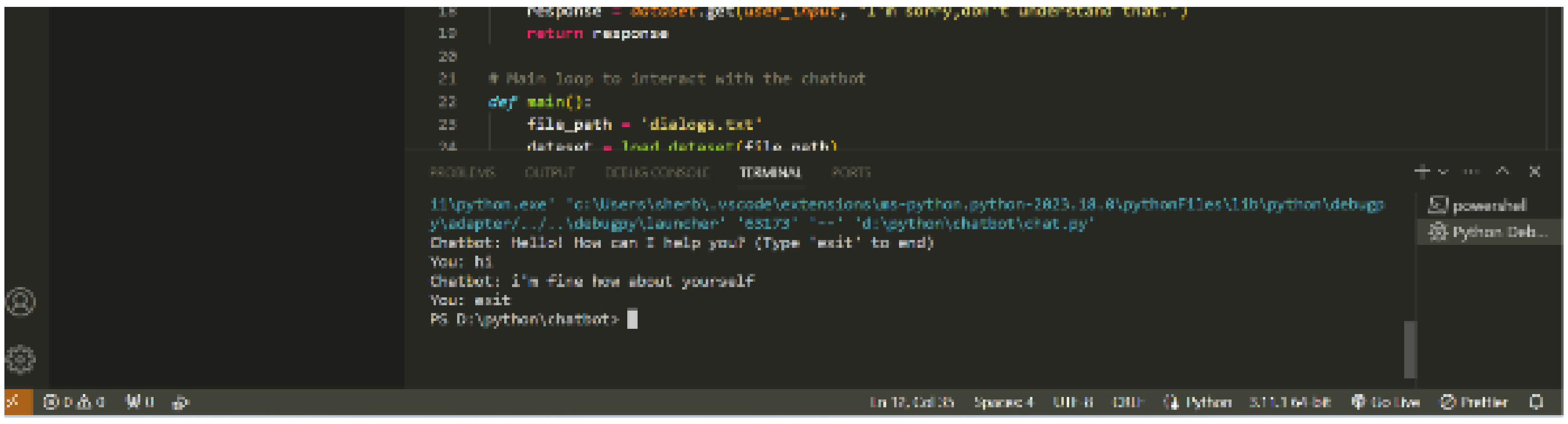
Write code to simulate the chat using the loaded dataset. You can create a loop that iterates through the messages and prints them in the terminal



```python
# Main loop to interact with the chatbot
def main():
    file_path = 'dialogs.txt'
    dataset = load_dataset(file_path)

    print("Chatbot: Hello! How can I help you? (Type 'exit' to end)")

    while True:
        user_input = input("You: ")
        if user_input.lower() == "exit":
            break
        response = chatbot_response(user_input, dataset)
        print("Chatbot:", response)

if __name__ == "__main__":
    main()
```

6. User Interface:

You can create a basic user interface that allows you to start and stop the chat simulation.



```python
        response = dataset.get(user_input, "I'm sorry,don't understand that.")
        return response

# Main loop to interact with the chatbot
def main():
    file_path = 'dialogs.txt'
    dataset = load_dataset(file_path)
```

```
11\python.exe' 'c:\Users\sherb\.vscode\extensions\ms-python.python-2023.18.0\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '65173' '--' 'd:\python\chatbot\chat.py'
Chatbot: Hello! How can I help you? (Type 'exit' to end)
You: hi
Chatbot: i'm fine how about yourself
You: exit
PS D:\python\chatbot>
```

Sample Code:

```python
import random
```

```python
# Load the dataset from the text file
def load_dataset(file_path):
    dataset = {}
    with open('dialogs.txt', 'r') as file:
        lines = file.readlines()
        for line in lines:
            pattern, response = line.strip().split(' => ')
            pattern = pattern.replace('.', '').replace('?', '')
            response = response.replace('.', '').replace('?', '')
            dataset[pattern.lower()] = response
    return dataset

# Generate a response based on user input
def chatbot_response(user_input, dataset):
    user_input = user_input.lower()
    response = dataset.get(user_input, "I'm sorry,don't understand that.")
    return response

# Main loop to interact with the chatbot
def main():
    file_path = 'dialogs.txt'
    dataset = load_dataset(file_path)

    print("Chatbot: Hello! How can I help you? (Type 'exit' to end)")

    while True:
        user_input = input("You: ")
        if user_input.lower() == "exit":
            break
        response = chatbot_response(user_input, dataset)
        print("Chatbot:", response)

if __name__ == "__main__":
    main()
```

Conclusion:

This document provides a basic foundation for creating a terminal-based chat application using a given dataset to simulate a chat conversation. You can extend this application by adding more features and interactivity, or by using larger and more complex datasets for testing and experimentation