Code:

import os

import re

class KeywordSearchSystem:

    def \_\_init\_\_(self, file\_path):

        self.file\_path = file\_path

        self.content = ""

        self.load\_file()

    def load\_file(self):

        """Load the text file."""

        if not os.path.exists(self.file\_path):

            raise FileNotFoundError(f"File not found: {self.file\_path}")

        with open(self.file\_path, 'r', encoding='utf-8') as file:

            self.content = file.readlines()

    def search(self, keyword, case\_sensitive=False):

        """Search for the keyword in the text file."""

        if not keyword.strip():

            raise ValueError("Keyword cannot be empty or whitespace.")

        results = []

        for line\_number, line in enumerate(self.content, start=1):

            if case\_sensitive:

                matches = [m.start() for m in re.finditer(re.escape(keyword), line)]

            else:

                matches = [m.start() for m in re.finditer(re.escape(keyword), line, re.IGNORECASE)]

            if matches:

                results.append((line\_number, matches))

        return results

# Main program

def main():

    print("Welcome to the Keyword Search System")

    file\_path = input("Enter the path to your text file: ")

    try:

        search\_system = KeywordSearchSystem(file\_path)

    except FileNotFoundError as e:

        print(e)

        return

    while True:

        query = input("\nEnter the keyword to search (or type 'exit' to quit): ")

        if query.lower() == 'exit':

            print("Goodbye!")

            break

        case\_option = input("Perform case-sensitive search? (yes/no): ").strip().lower()

        case\_sensitive = case\_option == 'yes'

        try:

            results = search\_system.search(query, case\_sensitive)

            if results:

                print("\nSearch Results:")

                for line\_number, positions in results:

                    print(f"Line {line\_number}: Positions {positions}")

            else:

                print("No matches found.")

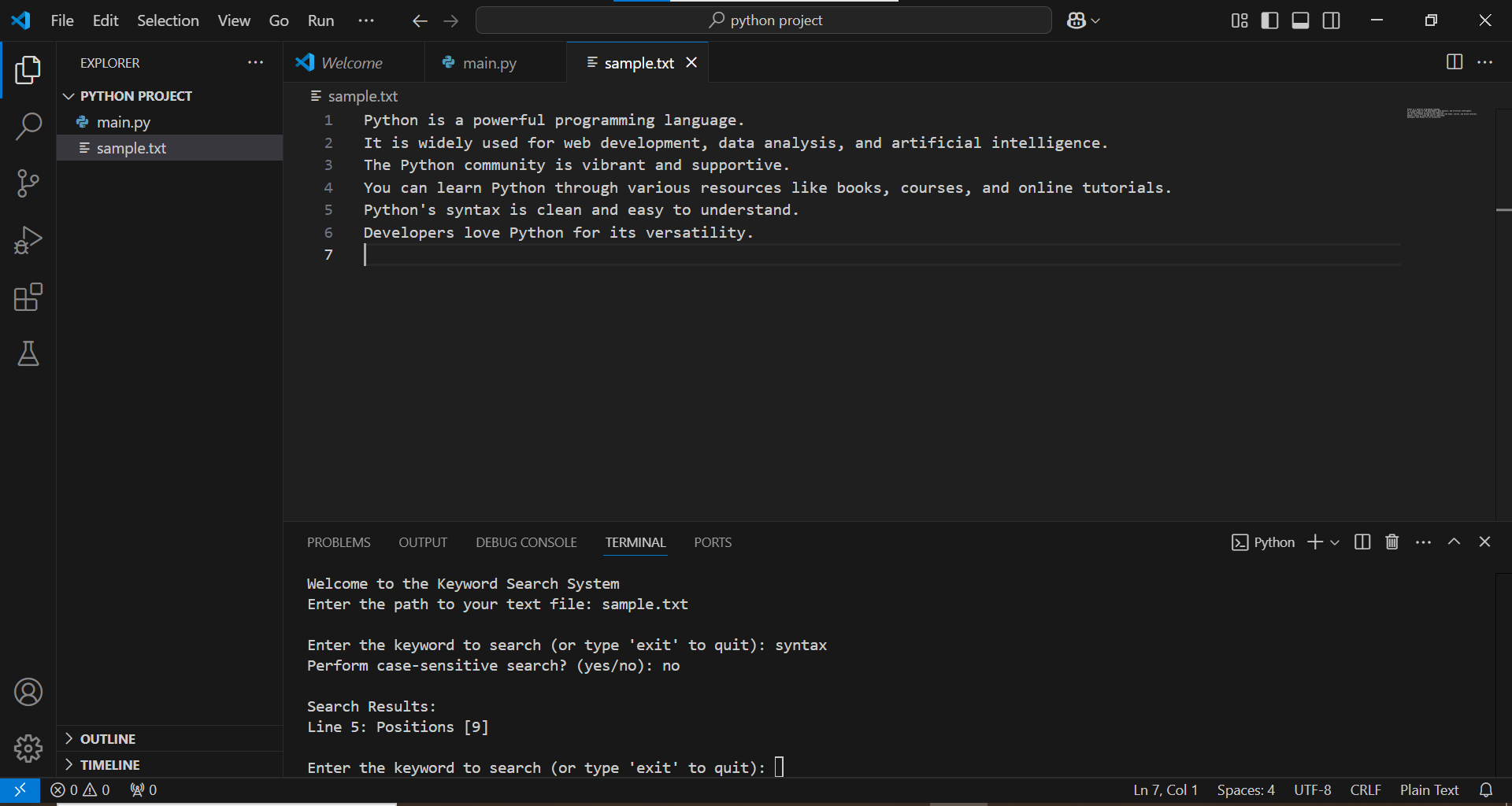
        except ValueError as e:

            print(e)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Output:



A screenshot of a computer

Description automatically generated