AI ASSSITED CODING

LABTEST-04

NAME: TAFSIL AHMED

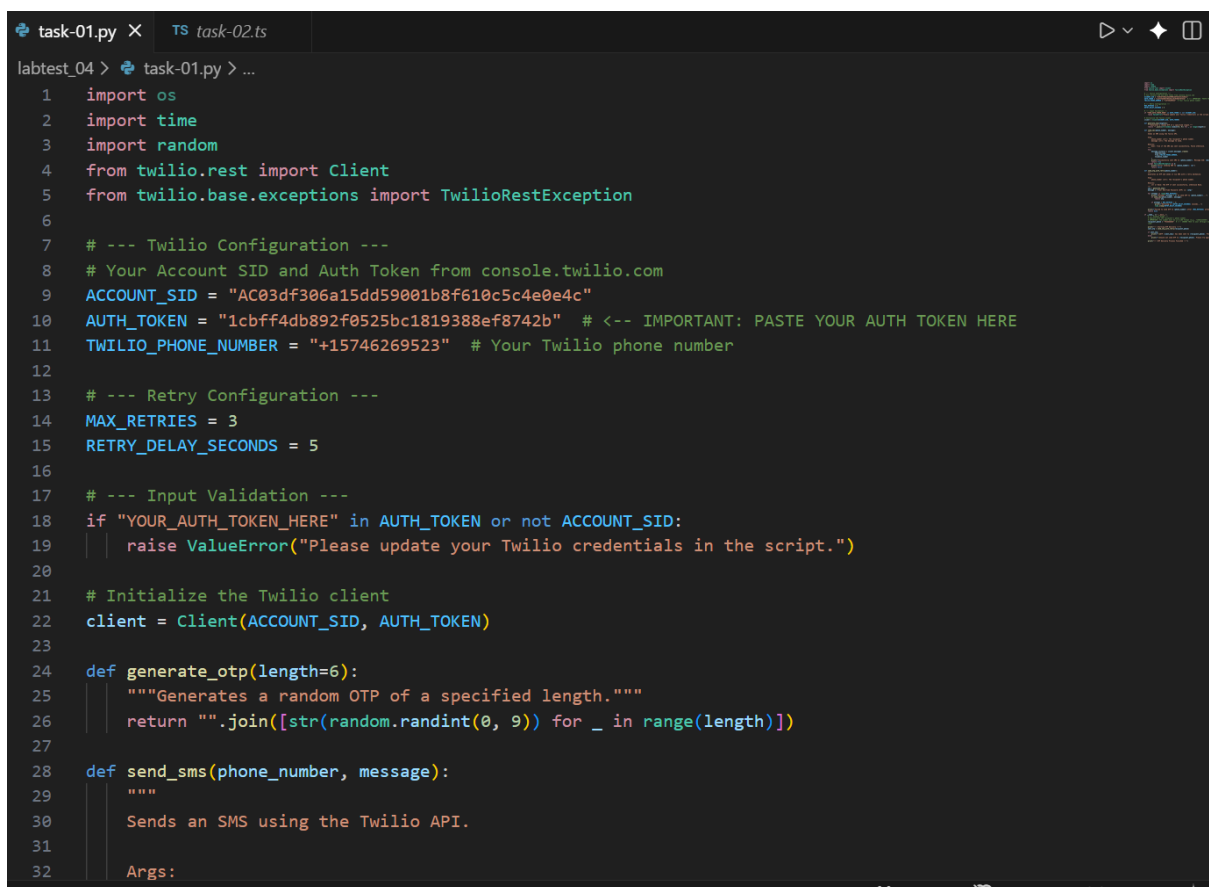ENROLLNO:2403A52071

BATCH:04

SET-02:

Q1. (API Integration)
a) Integrate with SMS gateway API for OTP delivery.
b) Implement retry when SMS fails

PROMPT:

Generate the python script which Integrates the system with an SMS gateway API to send OTP codes. If an SMS delivery fails, implement an automatic retry mechanism with a limited number of attempts.

CODE:

```python
import os
import time
import random
from twilio.rest import Client
from twilio.base.exceptions import TwilioRestException

# --- Twilio Configuration ---
# Your Account SID and Auth Token from console.twilio.com
ACCOUNT_SID = "AC03df306a15dd59001b8f610c5c4e0e4c"
AUTH_TOKEN = "1cbff4db892f0525bc1819388ef8742b"  # <-- IMPORTANT: PASTE YOUR AUTH TOKEN HERE
TWILIO_PHONE_NUMBER = "+15746269523"  # Your Twilio phone number

# --- Retry Configuration ---
MAX_RETRIES = 3
RETRY_DELAY_SECONDS = 5

# --- Input Validation ---
if "YOUR_AUTH_TOKEN_HERE" in AUTH_TOKEN or not ACCOUNT_SID:
    raise ValueError("Please update your Twilio credentials in the script.")

# Initialize the Twilio client
client = Client(ACCOUNT_SID, AUTH_TOKEN)

def generate_otp(length=6):
    """Generates a random OTP of a specified length."""
    return "".join([str(random.randint(0, 9)) for _ in range(length)])

def send_sms(phone_number, message):
    """
    Sends an SMS using the Twilio API.

    Args:
```

```python
28   def send_sms(phone_number, message):
33           phone_number (str): The recipient's phone number.
34           message (str): The message to send.
35
36       Returns:
37           bool: True if the SMS was sent successfully, False otherwise.
38       """
39       try:
40           message_instance = client.messages.create(
41               body=message,
42               from_=TWILIO_PHONE_NUMBER,
43               to=phone_number
44           )
45           print(f"Successfully sent SMS to {phone_number}. Message SID: {message_instance.sid}")
46           return True
47       except TwilioRestException as e:
48           print(f"Error sending SMS to {phone_number}: {e}")
49           return False
50
51   def send_otp_with_retry(phone_number):
52       """
53       Generates an OTP and sends it via SMS with a retry mechanism.
54
55       Args:
56           phone_number (str): The recipient's phone number.
57
58       Returns:
59           str or None: The OTP if sent successfully, otherwise None.
60       """
61       otp = generate_otp()
62       message = f"Your One-Time Password (OTP) is: {otp}"
```
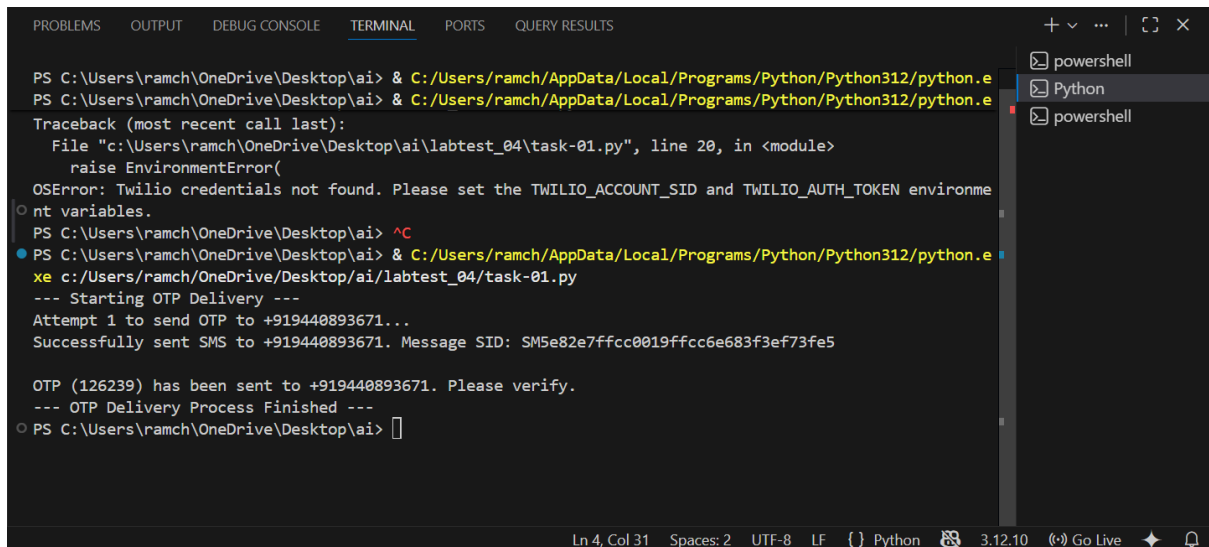
```python
51   def send_otp_with_retry(phone_number):
62       message = f"Your One-Time Password (OTP) is: {otp}"
63
64       for attempt in range(MAX_RETRIES):
65           print(f"Attempt {attempt + 1} to send OTP to {phone_number}...")
66           if send_sms(phone_number, message):
67               return otp
68
69           if attempt < MAX_RETRIES - 1:
70               print(f"Retrying in {RETRY_DELAY_SECONDS} seconds...")
71               time.sleep(RETRY_DELAY_SECONDS)
72
73       print(f"Failed to send OTP to {phone_number} after {MAX_RETRIES} attempts.")
74       return None
75
76   if __name__ == "__main__":
77       # --- Example Usage ---
78       # Replace with the recipient's phone number.
79       # IMPORTANT: The number must be in E.164 format (e.g., +15551234567)
80       recipient_phone = "+919440893671"  # <--- CHANGE THIS to your 10-digit mobile number with country code!
81
82       print("--- Starting OTP Delivery ---")
83       sent_otp = send_otp_with_retry(recipient_phone)
84
85       if sent_otp:
86           print(f"\nOTP ({sent_otp}) has been sent to {recipient_phone}. Please verify.")
87       else:
88           print(f"\nCould not send OTP to {recipient_phone}. Please try again later.")
89
90       print("--- OTP Delivery Process Finished ---")
```

OUTPUT:

```
PS C:\Users\ramch\OneDrive\Desktop\ai> & C:/Users/ramch/AppData/Local/Programs/Python/Python312/python.e   ▷ powershell
PS C:\Users\ramch\OneDrive\Desktop\ai> & C:/Users/ramch/AppData/Local/Programs/Python/Python312/python.e   ▷ Python
Traceback (most recent call last):                                                                         ▷ powershell
  File "c:\Users\ramch\OneDrive\Desktop\ai\labtest_04\task-01.py", line 20, in <module>
    raise EnvironmentError(
OSError: Twilio credentials not found. Please set the TWILIO_ACCOUNT_SID and TWILIO_AUTH_TOKEN environme
nt variables.
PS C:\Users\ramch\OneDrive\Desktop\ai> ^C
PS C:\Users\ramch\OneDrive\Desktop\ai> & C:/Users/ramch/AppData/Local/Programs/Python/Python312/python.e
xe c:/Users/ramch/OneDrive/Desktop/ai/labtest_04/task-01.py
--- Starting OTP Delivery ---
Attempt 1 to send OTP to +919440893671...
Successfully sent SMS to +919440893671. Message SID: SM5e82e7ffcc0019ffcc6e683f3ef73fe5

OTP (126239) has been sent to +919440893671. Please verify.
--- OTP Delivery Process Finished ---
PS C:\Users\ramch\OneDrive\Desktop\ai> []
```

Ln 4, Col 31   Spaces: 2   UTF-8   LF   {} Python   3.12.10   (•) Go Live

OBSERVATION:

The system needs a reliable way to send OTPs using an SMS gateway.
If the SMS doesn't go through, it should automatically try sending it again.
There should be a limit to how many times the system retries.
The process must ensure users receive their OTP without delays.
Clear feedback should be provided when the SMS is sent or if it ultimately fails.

Q2. (Code Translation)
a) Convert a Java class into TypeScript.
b) Add type definitions to improve safety.

PROMPT:

Convert this Java class into TypeScript and add clear type definitions to make the code safer.

CODE:

labtest_04 > TS task-02.ts > Person > constructor

```typescript
 5    class Person {
33        */
34        public getDetails(): string {
35            return `Name: ${this.getFullName()}, Age: ${this.age}`;
36        }
37
38        /**
39         * Increments the person's age by one.
40         */
41        public haveBirthday(): void {
42            this.age++;
43        }
44    }
45
46    // --- Example Usage ---
47
48    // Create a new instance of the Person class
49    const john = new Person("John", "Doe", 30);
50
51    // Log details to the console
52    console.log(john.getDetails()); // Output: Name: John Doe, Age: 30
53
54    // Increment the person's age
55    john.haveBirthday();
56
57    // Log details again to see the change
58    console.log(john.getDetails()); // Output: Name: John Doe, Age: 31
59
60    // The following would cause a TypeScript error if you tried to compile it,
61    // demonstrating the benefit of type safety:
62    // const jane = new Person("Jane", "Doe", "twenty-five"); // Error: Argument of type 'string' is not assignable to parameter of type
        'number'.
```

```typescript
 1
 2    /**
 3     * Represents a person with a first name, last name, and age.
 4     */
 5    class Person {
 6        private firstName: string;
 7        private lastName: string;
 8        private age: number;
 9
10        /**
11         * Constructs a new Person instance.
12         * @param firstName The person's first name.
13         * @param lastName The person's last name.
14         * @param age The person's age.
15         */
16        constructor(firstName: string, lastName: string, age: number) {
17            this.firstName = firstName;
18            this.lastName = lastName;
19            this.age = age;
20        }
21
22        /**
23         * Gets the person's full name.
24         * @returns A string representing the full name.
25         */
26        public getFullName(): string {
27            return `${this.firstName} ${this.lastName}`;
28        }
29
30        /**
31         * Gets a detailed string representation of the person.
32         * @returns A string with the person's name and age.
```

Ln 20, Col 6   Spaces: 4   UTF-8   LF   { } TypeScript   (•) Go Live

OUTPUT:

```
PS C:\Users\ramch\OneDrive\Desktop\ai>


PS C:\Users\ramch\OneDrive\Desktop\ai>

PS C:\Users\ramch\OneDrive\Desktop\ai>
PS C:\Users\ramch\OneDrive\Desktop\ai> npx ts-node labtest_04\task-02.ts
Name: John Doe, Age: 30
Name: John Doe, Age: 31
PS C:\Users\ramch\OneDrive\Desktop\ai>
```

OBSERVATION:

The task is to change an existing Java class into TypeScript code.
While converting, the code should include proper type definitions.
Adding types will make the code safer and reduce errors.
The structure and logic from the Java version should be kept clear in TypeScript.
Overall, the goal is to improve readability, safety, and maintainability.