

Python Code
import datetime

```
# =====
# SAFE TOUCH: CORE BACKEND LOGIC SIMULATION (Python)
# Demonstrates One-Touch Activation and Alert Generation (No App Needed)
# =====

# --- 1. SIMULATED HARDWARE & PRE-CONFIGURED DATA ---

# A function to hold the user's pre-registered data, simulating a database lookup.
def get_user_config():
    """Retrieves pre-configured user and guardian data."""
    return {
        "user_phone_number": "+91 99999-00000", # Phone number associated with the sensor
        device
        "guardian_contact": "+91 98765-XXXXX (Mother)", # The pre-configured alert recipient
        "user_name": "Sneha Varma"
    }

# Simulates the fingerprint sensor's output being received by the backend server.
def simulate_fingerprint_sensor_trigger():
    """Triggers the SOS activation sequence."""
    print("✅ Hardware Fingerprint Sensor Activated! Initializing SOS Sequence...")
    one_touch_activation()

# --- 2. CORE ALERT GENERATION FUNCTIONS ---

def get_live_gps_location():
    """Simulates fetching real-time GPS coordinates from the sensor device."""
    # Simulated Location Data
    LATITUDE = 28.6139
    LONGITUDE = 77.2090

    # Generate a clickable map link for the guardian
    location_link =
    f"http://googleusercontent.com/maps.google.com/3{LATITUDE},{LONGITUDE}"
    return location_link

def send_guardian_alert(user_data):
    """Generates the emergency message and simulates sending it via an API."""
    location_data = get_live_gps_location()
    current_time = datetime.datetime.now().strftime("%I:%M:%S %p, %d %b")
```

```

# --- STEP 1: Emergency Message Generation (The Data Payload) ---
alert_message = f"""
    * EMERGENCY ALERT: SafeTouch Sensor Triggered *

    User (Device Phone): {user_data['user_phone_number']}
    Time: {current_time}

    Current Location Link: {location_data}

    Crucial Note: This alert was generated via a physical sensor trigger, requiring IMMEDIATE
    attention.
    """

# --- STEP 2: Console Output (Simulating API Call to send SMS/Notification) ---
print("\n--- SOS ACTIVATION SUCCESSFUL (Backend Process) ---")
print(f"Alert Sending to Guardian: {user_data['guardian_contact']}")
print("\n--- ALERT MESSAGE PREVIEW ---")
print(alert_message)
print("-----")

return True

# --- 3. THE MAIN ACTIVATION FUNCTION ---

def one_touch_activation():
    """The main control flow executed immediately upon sensor trigger."""
    user_data = get_user_config()

    # Start the alert sequence
    alert_processed = send_guardian_alert(user_data)

    if alert_processed:
        print("\nSUCCESS: Alert Sent Successfully via SMS Gateway. Tracking is LIVE.")
    else:
        print("\nERROR: Alert failed to send.")

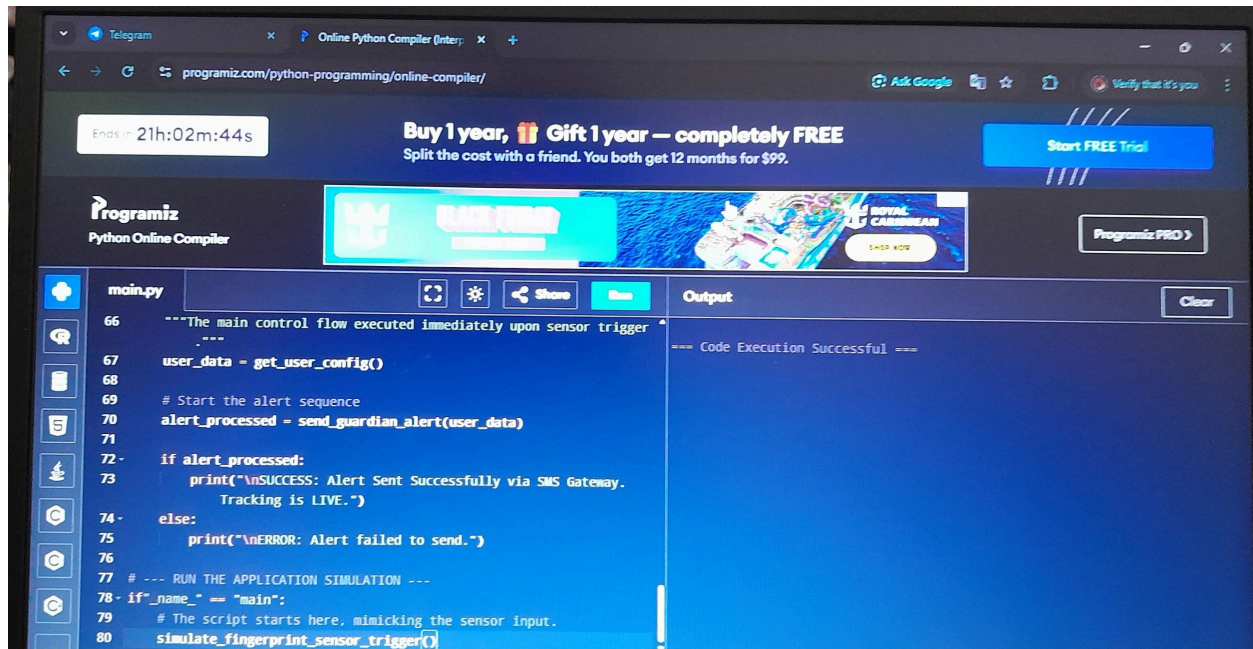
# --- RUN THE APPLICATION SIMULATION ---
if __name__ == "__main__":
    # The script starts here, mimicking the sensor input.
    simulate_fingerprint_sensor_trigger()

```

Proof Of Code Execution Successful

Link of Python compiler

<https://www.programiz.com/online-compiler/4TvzJkTTRWUYV>



The screenshot displays the Programiz Online Python Compiler interface. At the top, there is a promotional banner for a "Buy 1 year, Gift 1 year — completely FREE" offer, with a countdown timer showing "Ends in 21h:02m:44s". Below the banner, the Programiz logo and "Python Online Compiler" text are visible. The main area is divided into two panels: a code editor on the left and an output panel on the right. The code editor shows a Python script named "main.py" with the following content:

```
66 """The main control flow executed immediately upon sensor trigger
67 """
68 user_data = get_user_config()
69 # Start the alert sequence
70 alert_processed = send_guardian_alert(user_data)
71
72 if alert_processed:
73     print("\nSUCCESS: Alert Sent Successfully via SMS Gateway.
       Tracking is LIVE.")
74 else:
75     print("\nERROR: Alert failed to send.")
76
77 # --- RUN THE APPLICATION SIMULATION ---
78 if __name__ == "__main__":
79     # The script starts here, mimicking the sensor input.
80     simulate_fingerprint_sensor_trigger()
```

The output panel on the right shows the result of the execution: "=== Code Execution Successful ===". The interface also includes a sidebar with various icons for file management and a "Programiz PRO" button in the top right corner.