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