Emergency SMS feature

SMS-Based Emergency Alerts: Rationale and Benefits

Overview

Short Message Service (SMS) offers a reliable and efficient mechanism for disseminating emergency alerts in time-critical scenarios. This approach is especially suitable for embedded health monitoring systems or IoT applications where immediate notification to caregivers or emergency contacts is essential.

Why Use SMS?

- **Ubiquitous Reach**
    - Works across virtually all mobile networks and devices—even feature phones.
    - No need for internet connectivity, making it effective in remote or offline areas.
- **Low Latency**
    - Delivers messages instantly with minimal delay, ideal for urgent alerts like health anomalies or accident detection.
- **Resource-Efficient**
    - Minimal power and bandwidth requirements, compatible with low-energy embedded systems like Arduino or ESP modules.
- **High Reliability**
    - Robust under network congestion and during natural disasters when data services may fail.
- **Simplicity & Standardization**
    - Easy to implement using existing GSM modules and AT commands.

    (here I use textbee.dev ‘s opensource API key to convert the android device to a self-hosted gateway with global coverage)

    - Supported by numerous microcontroller libraries and open APIs.

Use Case Example

> In NeuroFit, SMS can be triggered when biometric thresholds (e.g., sudden heart rate spikes) cross predefined limits. A preformatted message alerts caregivers with GPS location and patient status, ensuring rapid response.

Implementation code

```
@app.route('/api/emergency-alert', methods=['POST'])
@login_required
def emergency_alert():
    data = request.json
    email = data.get('email')
    phone = data.get('phone')  # Expect phone number in request
    message = data.get('message')
    location = data.get('location')
    subject = "NeuroFit Emergency Alert"
    body = f"{message}\nLocation: {location}\nUser: {current_user.username} ({current_user.email})"
    status_msgs = []  # ← Always define at the top

BASE_URL = 'https://api.textbee.dev/api/v1'
        API_KEY = 'API_KEY'  # Replace with your actual API key
        DEVICE_ID = 'DEVICE_ID'  # Replace with your actual device ID

        url = f"{BASE_URL}/gateway/devices/{DEVICE_ID}/send-sms"
        headers = {
            'x-api-key': API_KEY,
            'Content-Type': 'application/json'
        }
        payload = {
            'recipients': [phone],
            'message': f"NeuroFit Emergency Alert: {message}\nLocation: {location}, current user: {current_user.username} ({current_user.email})"
        }
        try:
            response = requests.post(url, headers=headers, data=json.dumps(payload))
            response.raise_for_status()  # Raise an exception for HTTP errors (4xx or 5xx)
            status_msgs.append("SMS Sent Successfully!")
        except requests.exceptions.HTTPError as err:
            print(f"HTTP error occurred: {err}")
            print(f"Response content: {response.text}")
            status_msgs.append("Failed to send SMS (HTTP error).")
        except requests.exceptions.ConnectionError as err:
            print(f"Connection error occurred: {err}")
            status_msgs.append("Failed to send SMS (connection error).")
        except Exception as err:
            print(f"An unexpected error occurred: {err}")
            status_msgs.append("Failed to send SMS (unexpected error).")

    return jsonify({"status": " ".join(status_msgs)})
```

---

advanced part : create my own API to send messages.

we can use the user’s native SMS app to send message