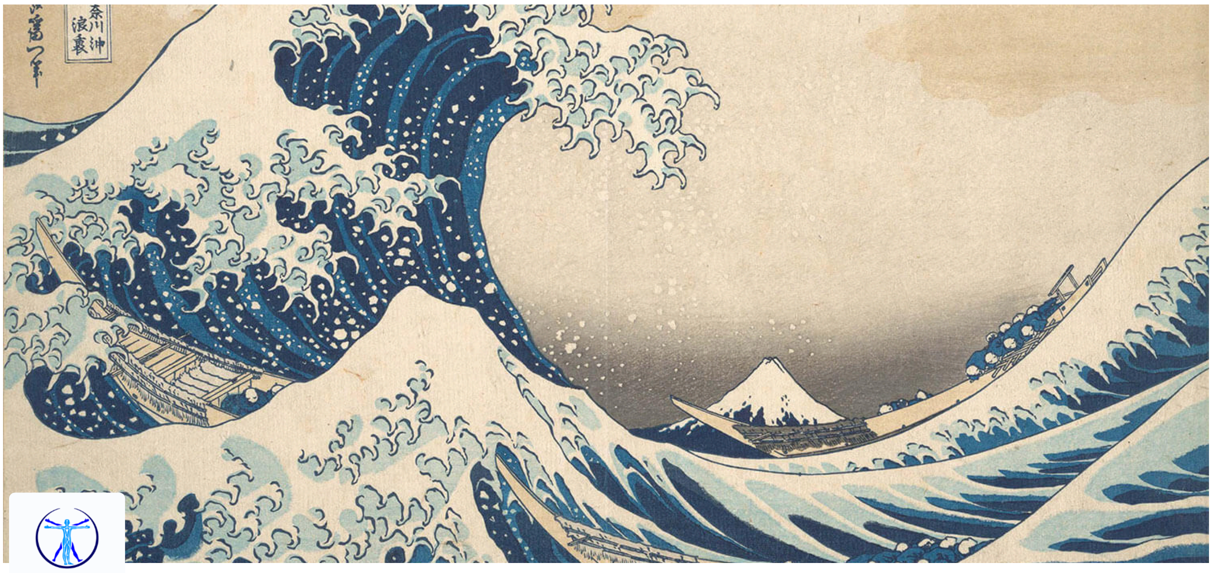

# NeuroFit Headband

## Mind and Health Tracker Device

### Project Documentation

#### 1. Introduction

The **Mind and Health Tracker** is an advanced wearable device that monitors the user's mental and physical health in real time. The device integrates key health sensors in a headband form factor and transmits data via Bluetooth Low Energy (BLE) to a web-based application. The application not only visualizes this data but also leverages machine learning to predict the user's mood and offers guided mental wellness support. The system includes automatic emergency alerts based on abnormal readings.

---

#### 2. System Overview

#### Key Features:

- Real-time monitoring of brain activity, heart rate, and SpO2.
- BLE-based data transfer to a secure web application.
- Mood prediction using machine learning.

- Guided meditation sessions based on health analytics.
- Emergency alert system with GPS location sharing.

---

## 3. Hardware Components

### 1. Seed Studio Xiao nRF52840

- Microcontroller handling sensor inputs, BLE communication, and data transmission.

### 2. BioAmp EXG Pill

- EEG sensor for tracking brain wave activity.
- Analyzes Delta, Theta, Alpha, Beta, and Gamma frequencies.

### 3. MAX30102 Pulse Oximeter

- Measures heart rate and oxygen saturation (SpO2).

### 4. Vibration Motor

- Provides haptic feedback for notifications (e.g., emergency or meditation prompts).

---

## 4. Brain Waves: Classification and Significance

The brain communicates via electrical impulses. These impulses occur at various frequencies and are classified into five main types of brain waves. Understanding these waves helps determine the user's mental state.

### A. Delta Waves (0.5 – 4 Hz)

- **Associated State:** Deep sleep, unconsciousness
- **Function:** Body healing, regeneration
- **Indicators:** High delta activity can indicate restorative sleep or potential neurological issues if seen during wakefulness

### B. Theta Waves (4 – 8 Hz)

- **Associated State:** Light sleep, deep meditation, creativity
- **Function:** Subconscious insight, memory access

- **Indicators:** Increased theta is linked to creativity but may indicate inattention in active settings

## C. Alpha Waves (8 – 12 Hz)

- **Associated State:** Relaxed alertness, calm focus
- **Function:** Stress reduction, coordination
- **Indicators:** Balanced alpha activity is ideal for relaxed mental states

## D. Beta Waves (12 – 30 Hz)

- **Associated State:** Active thinking, problem-solving, alertness
- **Function:** Cognitive tasks, focus, logical reasoning
- **Indicators:** High beta may indicate stress or anxiety

## E. Gamma Waves (30 – 100 Hz)

- **Associated State:** High-level information processing, learning
- **Function:** Memory retention, perception
- **Indicators:** High gamma levels correlate with cognitive sharpness

The EEG sensor (BioAmp EXG Pill) will measure these frequencies, and the system will classify mental states accordingly. Mood prediction and guided meditation are aligned with this analysis.

---

# 5. Software Architecture

## A. Frontend (Web Application)

- **Technologies:** HTML, CSS, JavaScript, Bootstrap
- **Libraries:** Chart.js (for visualization), Socket.io (for real-time updates if needed)

## Features:

- **Dashboard** displaying live heart rate, SpO2, and brainwave data.
- **Mood Score Panel** updated in real time using ML output.
- **Historical Analysis** to view trends over time.
- **Meditation Guide** recommendations based on current stress levels.

## B. Backend (Django Framework)

- **Functions:** BLE data reception, user authentication, database operations, ML processing
- **Database:** PostgreSQL or MySQL

## Modules:

- **Data Ingestion**: Receives BLE data and logs it.
- **Authentication**: Manages secure sign-ups/logins.
- **Mood Prediction Engine**: Uses a trained ML model (TensorFlow or scikit-learn).
- **Emergency System**: Detects anomalies and triggers alerts with GPS data.

---

## 6. Machine Learning Model

## Objective:

Predict user's mood (e.g., Calm, Stressed, Focused, Happy) based on EEG, heart rate, and SpO2 data.

## Model Details:

- **Type**: Supervised Learning
- **Input**: Brain wave frequency bands, HRV, SpO2
- **Output**: Mood class/score
- **Tools**: TensorFlow Lite (for embedded), Python (for backend training)
- **Training Dataset**: Pre-collected health data labeled with mood states

---

## 7. System Workflow

1. **Data Collection:**
    - Brainwave (EEG), heart rate, and SpO2 are recorded continuously by sensors.
2. **BLE Transmission:**
    - The Xiao nRF52840 sends data via BLE to the frontend or backend.
3. **Backend Processing:**

- Data is stored in the database.
- Analyzed by ML models.
- Emergency triggers checked.

4. **Frontend Visualization:**
   - Data visualized with real-time charts.
   - Mood predictions and meditation guides updated.
5. **Emergency Handling:**
   - Abnormal patterns (e.g., low SpO2 or EEG spike) trigger alerts.
   - Location and status sent to emergency contact.

---

## 8. Features Summary

| Feature | Description |
|---|---|
| **Sign-Up/Login** | Secure user authentication. |
| **Dashboard** | Real-time health data visualization. |
| **Historical Analytics** | Filter and view past data trends. |
| **Mood Prediction** | AI-based mood scoring. |
| **Meditation Mode** | Guided relaxation sessions. |
| **Emergency Alerts** | GPS-linked SMS/email alerts. |

---

## 9. Technology Stack

| Component | Technology |
|---|---|
| Microcontroller | Seed Studio Xiao nRF52840 |
| EEG Sensor | BioAmp EXG Pill |
| Pulse Sensor | MAX30102 |
| Frontend | HTML, CSS, JS, Chart.js, Bootstrap |
| Backend | Django, PostgreSQL/MySQL |
| ML Model | TensorFlow, Scikit-learn |

---

## 10. Project Timeline

| Milestone | Timeframe |
|---|---|
| Requirements & Design | Week 1-2 |
| Hardware Integration | Week 3-5 |
| Frontend Development | Week 6-8 |
| Backend Development | Week 7-9 |
| ML Model Integration | Week 8-10 |
| Testing & Debugging | Week 10-12 |
| Final Deployment | Week 13 |

## 11. Conclusion

The Mind and Health Tracker is designed to be a powerful, compact, and user-friendly solution for mental and physical health monitoring. It combines sensor technology, BLE communication, web development, and machine learning to provide real-time feedback and actionable health insights. This innovation will be valuable not only to individual users but also in healthcare and wellness research.

By providing real-time analysis of brain waves and other physiological signals, this system gives users actionable insights into their mental state. With emergency alert systems and personalized meditation guides, it moves beyond monitoring to actively supporting well-being and safety.

#### Github Repository Structure (Sample)

```
NeuroFit/
├── Arduino/
│   ├── MindHealthTracker.ino
│   ├── config.h
│   ├── BLEService.h
│   ├── heartRate.h
│   └── spo2_algorithm.c
│
├── FlutterApp/ (optional if using Flutter frontend)
│   ├── lib/
```

```
│   │   ├── main.dart
│   │   ├── screens/
│   │   ├── services/
│   │   └── widgets/
│   ├── pubspec.yaml
│   └── android/app/src/main/AndroidManifest.xml
│
├── WebApp/
│   ├── index.html
│   ├── style.css
│   ├── app.js
│   └── assets/
│       └── charts.js
│
├── Backend/
│   ├── manage.py
│   ├── tracker/
│   │   ├── models.py
│   │   ├── views.py
│   │   ├── urls.py
│   │   └── ble_receiver.py
│   ├── templates/
│   └── static/
│
├── ML_Model/
│   ├── train_model.ipynb
│   ├── mood_classifier.tflite
│   └── utils.py
│
├── Hardware_Schematics/
│   ├── schematic.png
│   ├── wiring_diagram.pdf
│   └── BOM.csv
│
├── README.md
├── LICENSE
└── .gitignore
```

### Arduino Code (Sample)

```
// ========== Arduino/MindHealthTracker.ino ==========

#include <Arduino_LSM9DS1.h>
#include <Wire.h>
#include <MAX30105.h>
#include "heartRate.h"
#include <bluefruit.h>
#include "config.h"
#include "BLEService.h"

MAX30105 particleSensor;
unsigned long lastSend = 0;
const int interval = 1000;

void setup() {
  Serial.begin(115200);
  pinMode(EEG_PIN, INPUT);
  pinMode(VIBRATION_PIN, OUTPUT);
  setupBLE();

  Wire.begin();
  if (!particleSensor.begin(Wire, I2C_SPEED_STANDARD)) {
    Serial.println("MAX30102 not found!");
    while (1);
  }
  particleSensor.setup();

  if (!IMU.begin()) {
    Serial.println("IMU not found!");
    while (1);
  }
  Serial.println("MindHealthTracker started");
}
```

```
void loop() {
  int eegVal = analogRead(EEG_PIN);
  String brainState = classifyBrainWave(eegVal);

  float hr = 0, spo2 = 0;
  byte hrValid, spo2Valid;
  readHRSpO2(&hr, &spo2, &hrValid, &spo2Valid);

  bool fallDetected = detectFreeFall();
  bool emergency = (spo2Valid && spo2 < 90) || eegVal > 950 || fallDetecte
d;
  digitalWrite(VIBRATION_PIN, emergency ? HIGH : LOW);

  if (millis() - lastSend >= interval) {
    lastSend = millis();
    char buffer[60];
    snprintf(buffer, sizeof(buffer), "HR:%.1f,SpO2:%.1f,EEG:%d,%s%s",
             hr, spo2, eegVal, brainState.c_str(), fallDetected ? ",FALL" : "");
    Serial.println(buffer);
    customChar.notify((uint8_t*)buffer, strlen(buffer));
  }
}

String classifyBrainWave(int eeg) {
  if (eeg < 250) return "Delta";
  else if (eeg < 450) return "Theta";
  else if (eeg < 650) return "Alpha";
  else if (eeg < 850) return "Beta";
  else return "Gamma";
}

bool detectFreeFall() {
  float x, y, z;
  if (IMU.accelerationAvailable()) {
    IMU.readAcceleration(x, y, z);
    float magnitude = sqrt(x * x + y * y + z * z);
    return magnitude < FALL_THRESHOLD;
```

```
  }
  return false;
}

void readHRSpO2(float* hr, float* spo2, byte* hrValid, byte* spo2Valid) {
  const int samples = 100;
  uint32_t irBuffer[samples];
  uint32_t redBuffer[samples];
  for (int i = 0; i < samples; ++i) {
    while (!particleSensor.available()) delay(1);
    redBuffer[i] = particleSensor.getRed();
    irBuffer[i] = particleSensor.getIR();
    particleSensor.nextSample();
  }
  maxim_heart_rate_and_oxygen_saturation(irBuffer, samples, redBuffer, sp
o2, spo2Valid, hr, hrValid);
}

// ========== Arduino/config.h ==========

#ifndef CONFIG_H
#define CONFIG_H

#define EEG_PIN A0
#define VIBRATION_PIN 10
#define FALL_THRESHOLD 2.5 // g-forces
#define SERVICE_UUID "6E400001-B5A3-F393-E0A9-E50E24DCCA9E"
#define CHARACTERISTIC_UUID "6E400003-B5A3-F393-E0A9-E50E24DC
CA9E"

#endif

// ========== Arduino/BLEService.h ==========

#ifndef BLE_SERVICE_H
#define BLE_SERVICE_H

#include <bluefruit.h>
```

```
extern BLEService customService;
extern BLECharacteristic customChar;

BLEService customService = BLEService(SERVICE_UUID);
BLECharacteristic customChar = BLECharacteristic(CHARACTERISTIC_UUI
D);

void setupBLE() {
  Bluefruit.begin();
  Bluefruit.setName("MindHealthTracker");
  customService.begin();
  customChar.setProperties(CHR_PROPS_NOTIFY);
  customChar.setPermission(SECMODE_OPEN, SECMODE_NO_ACCESS);
  customChar.setFixedLen(20);
  customChar.begin();
  Bluefruit.Advertising.addService(customService);
  Bluefruit.Advertising.start();
}

#endif

// ========== Arduino/heartRate.h ==========

#ifndef HEART_RATE_H
#define HEART_RATE_H

#include "spo2_algorithm.h"

void readHRSpO2(float* hr, float* spo2, byte* hrValid, byte* spo2Valid);

#endif

// ========== Arduino/spo2_algorithm.c ==========

// NOTE: Insert full source from Maxim's spo2_algorithm.c here.
// Download from SparkFun or Maxim official repo:
// https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/tree/m
aster/examples/Example9_SPO2
```

## Created by Anurag Panda

**B. Tech in CSE**

**Email : anuragpanda.dev@gmail.com**

**Phone number: 9883863117**

**NSHM Knowledge Campus, Durgapur**

**Durgapur - 713212**