**React Native Developer Assignment – Bluetooth & WebSocket**

**Objective**

Develop a **React Native mobile app** that connects to **at least three real nearby Bluetooth (BLE) devices**, retrieves live data in real-time, and streams updates to a **React-based web app** using **WebSocket (Socket.io)** in a shared room for real-time monitoring.

**Task Overview**

You will build **two parts**:

1. A **React Native mobile app**.

2. A **React web app** for displaying live updates.

**Mobile App Requirements**

● Scan and display a list of nearby BLE devices.

● Allow **manual connection** to **at least three physical devices**.

● Receive **live data streams** from connected devices.

● Manage **disconnects and automatic reconnections**.

● Display **real-time connection status and incoming data**.

● Emit **Socket.io events** to a shared room:

○ device\_connected when a device connects.

○ device\_data when new data is received.

○ device\_disconnected when a device disconnects (optional).

**Web App Requirements**

● A **React (or HTML+JS) web app** that connects to the same Socket.io room.

● Display incoming events **grouped by device**:

○ Show device name, ID, connection status, and live data.

○ Highlight new connections and updates in real-time.

**Bluetooth Requirements**

● Use **real BLE scanning** to detect nearby devices.

● Display a **list of available devices** with name and MAC address.

● Allow **manual connection** to devices.

● Subscribe to and read **live data** from the devices.

● Handle **disconnects and reconnections** reliably.

**WebSocket (Socket.io) Requirements**

● Use a **free or local Socket.io server**.

● Emit the following events to a shared room:

○ device\_connected with device ID, name, and connection timestamp.

○ device\_data with device ID, value, unit, and timestamp.

○ device\_disconnected with device ID and disconnect timestamp (optional).

**Deliverables**

● A **GitHub repository** containing:

○

○

○

○

Full **React Native mobile app code**.

Full **React web app code**.

A **README file** with setup and run instructions.

An **optional short demo video** showing scanning, connecting, data streaming,

and live updates.

**Evaluation Criteria**

● Real **BLE integration** with physical devices (not mock data).

● Proper **Socket.io implementation** using events and rooms.

● **Clear, responsive UI** for both mobile and web.

● **Code quality, structure, and readability**.

● Bonus points for:

○

○

○

Automatic reconnection handling.

Clean, structured logging for debugging.

UX improvements for a smoother user experience.

**Notes**

You may use **React Native CLI or Expo (bare workflow)**.

You may run your **Socket.io server locally or on a free cloud instance** as needed.

Use **clean architecture and clear code separation** for easy maintenance and future extension.