BLE ECG/EEG Monitoring System

Final Software Architecture & Technology Justification

Project Overview

This project is a **desktop-based biomedical monitoring system** that visualizes, processes, and exports real-time **ECG and EEG signals** via **BLE (Bluetooth Low Energy)** integration. While the final BLE connectivity will be integrated in hardware testing, the current application supports **fully simulated signal streams**, advanced **signal processing**, **secure encrypted data storage**, and **multi-format export options** (CSV, JSON, EDF).

Tech Stack Overview

Technology	Purpose
Electron.js	Cross-platform desktop application framework
React.js (via Vite)	Responsive frontend UI with state management for dynamic visualization
Chart.js	Real-time signal graph plotting (ECG & EEG)
SQLite (better-sqlite3)	Embedded local database for session-based encrypted data storage
Node.js (CommonJS)	Backend logic, database operations, export handling
json2csv	Converts JSON-formatted session data into CSV format
node-edf	EDF medical format export library for EEG/ECG signals
fft-js	FFT analysis for EEG frequency bands (alpha, beta, theta, delta)
Custom Encryption	Ensures biometric data is safely stored and unreadable without decryption
date-fns / Intl	For precise timestamp formatting and conversion to local timezone

Why This Stack?

Electron.js vs Flutter/JavaFX:

- Electron provides **native OS integration**, rich Node.js ecosystem access, and seamless packaging.
- **React + Electron** allows fast iteration and flexible component reuse.

React.js vs Vanilla JS / Angular:

- Minimal overhead via **Vite**, easy component breakdown.
- Community support and readable state logic using hooks (useState, useEffect).

better-sqlite3 vs knex / sequelize / MySQL:

- Zero config, synchronous operations (ideal for Electron), and ultra-fast.
- Perfect for **session-based small data volumes**.

node-edf vs edf-writer / custom implementation:

- node-edf is one of the few that complies with **EDF+ specs** in modern ESM-compatible format.
- Supports chunked streaming, ideal for EEG signal blocks.

FFT & Signal Processing with fft-js:

- Lightweight, no dependency on Python/scipy, and effective for basic EEG band detection.

Final Folder Structure (Key Files)

```
/ble-ecg-eeg-app/
├── /client/
│   ├── /components/
│   │   ├── GraphDisplay.jsx
│   │   ├── DataTable.jsx
│   │   ├── Settings.jsx
│   │   └── UIContainer.jsx
│   ├── /utils/
│   │   └── constants.js
│   ├── App.jsx
│   ├── main.jsx
│   └── styles/
│       └── App.css
├── /electron/
│   ├── main.js
│   ├── preload.js
│   ├── db.js
│   ├── dataSimulator.js
│   ├── dataProcessor.js
│   └── encryption.js
├── /assets/icons/
│   └── app-icon.ico
├── package.json
└── .vscode/, .gitignore, README.md
```

Functional Breakdown

1. Data Acquisition

- Simulates **ECG (Lead I)** and **EEG (Fp1, Fp2)** signals.
- Signals processed for:
  - ECG: Heart Rate, RR Intervals
  - EEG: Alpha, Beta, Theta, Delta band powers
- Real-time updates with millisecond-level timestamping.

2. Signal Processing

- ECG filtering: Baseline removal, Notch, Bandpass
- QRS detection, HRV calculation
- EEG FFT analysis for brainwave band identification

3. Storage & Encryption

- Each session logs data in a local SQLite database.
- All data stored using AES-256-like symmetric encryption.
- Auto-session rotation maintains, last 10 sessions.

4. User Interface

- Responsive dashboard layout with:
    - Mode selector (ECG/EEG/BOTH)
    - Sampling rate configuration
    - Real-time metric table
    - Chart.js dynamic line plot
    - Export buttons with dialog confirmations

5. Export

- All data exports to /ECG_EEG_Exports in user's Downloads folder.
- Formats:
    - CSV
    - JSON
    - EDF (medical format with signal metadata)

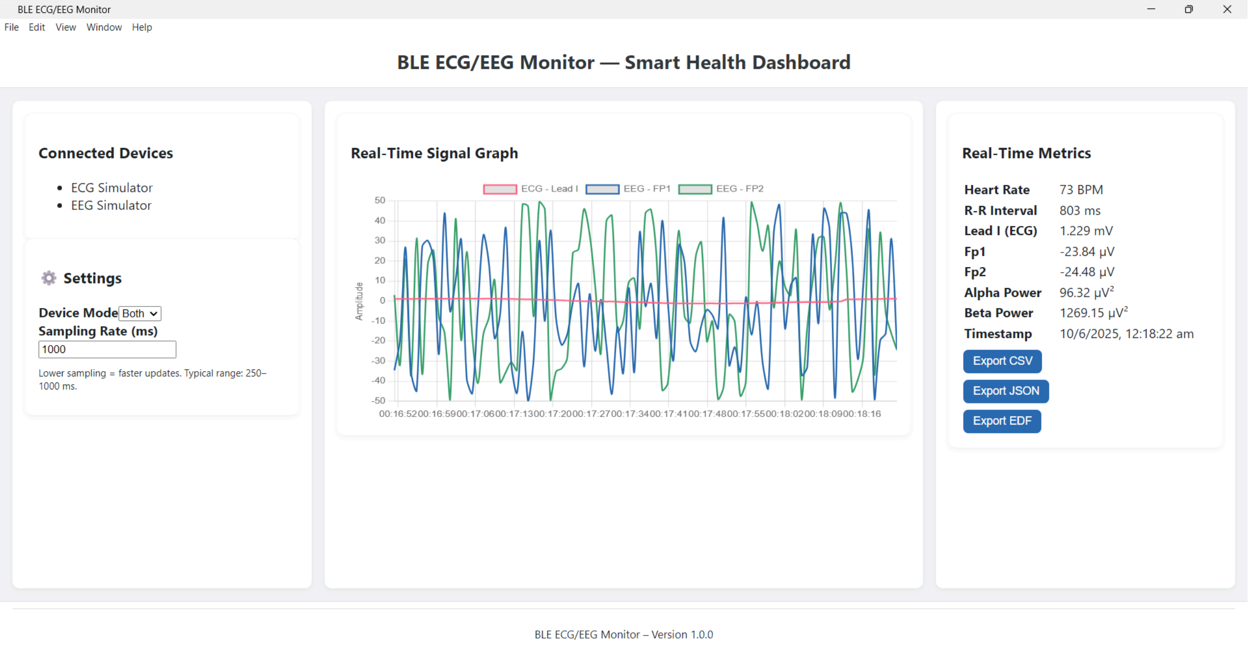
Final Notes

- The app is built for **offline-first**, **portable**, and **secure** usage.
- Works across **Windows/Linux/macOS**.
- Future extensions include:
    - Live BLE device parsing
    - AI signal anomaly detection
    - Cloud sync and remote analysis

Project Status

- Fully functional simulated data.
- Graphs, metrics, storage, export working.
- Production-ready with customizable frontend.
- Awaiting actual BLE hardware integration.

Screenshots of the working Application



BLE ECG/EEG Monitor — Smart Health Dashboard

Connected Devices

- ECG Simulator
- EEG Simulator

Settings

Device Mode Both

Sampling Rate (ms) 1000

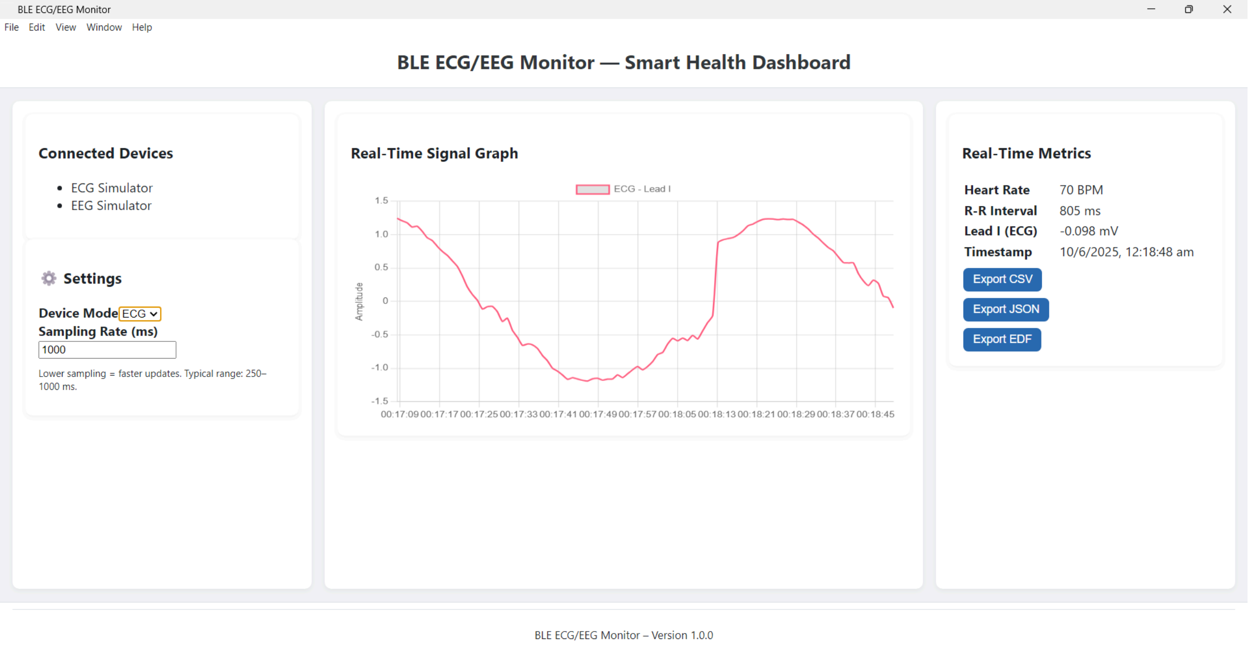
Lower sampling = faster updates. Typical range: 250–1000 ms.

Real-Time Metrics

Heart Rate	73 BPM
R-R Interval	803 ms
Lead I (ECG)	1.229 mV
Fp1	-23.84 μV
Fp2	-24.48 μV
Alpha Power	96.32 μV^2
Beta Power	1269.15 μV^2
Timestamp	10/6/2025, 12:18:22 am

BLE ECG/EEG Monitor – Version 1.0.0

Fig 1.1: Simulated data visualization of both ECG and EEG



BLE ECG/EEG Monitor — Smart Health Dashboard

Connected Devices

- ECG Simulator
- EEG Simulator

Settings

Device Mode ECG

Sampling Rate (ms) 1000

Lower sampling = faster updates. Typical range: 250–1000 ms.

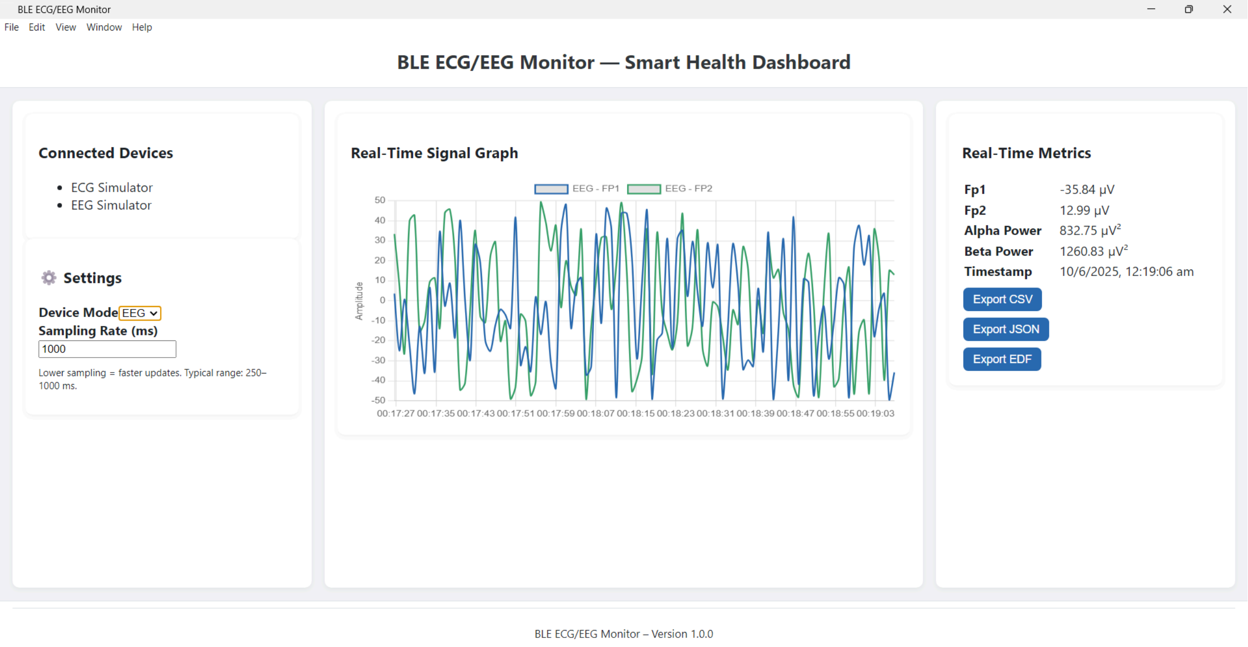
Real-Time Signal Graph

Real-Time Metrics

Heart Rate	70 BPM
R-R Interval	805 ms
Lead I (ECG)	-0.098 mV
Timestamp	10/6/2025, 12:18:48 am

BLE ECG/EEG Monitor – Version 1.0.0

Fig 1.2: Simulated data visualization of ECG



BLE ECG/EEG Monitor — Smart Health Dashboard

Connected Devices

- ECG Simulator
- EEG Simulator

Settings

Device Mode EEG

Sampling Rate (ms) 1000

Lower sampling = faster updates. Typical range: 250–1000 ms.

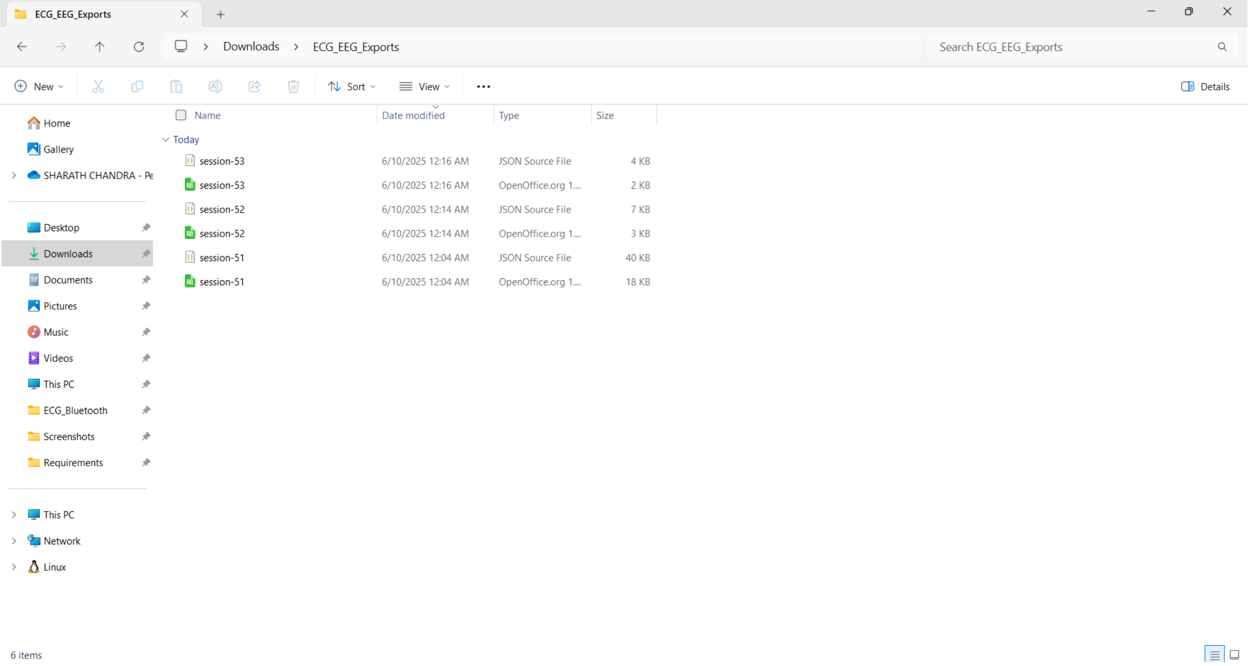
Real-Time Signal Graph

Real-Time Metrics

Fp1	-35.84 µV
Fp2	12.99 µV
Alpha Power	832.75 μV^2
Beta Power	1260.83 μV^2
Timestamp	10/6/2025, 12:19:06 am

BLE ECG/EEG Monitor – Version 1.0.0

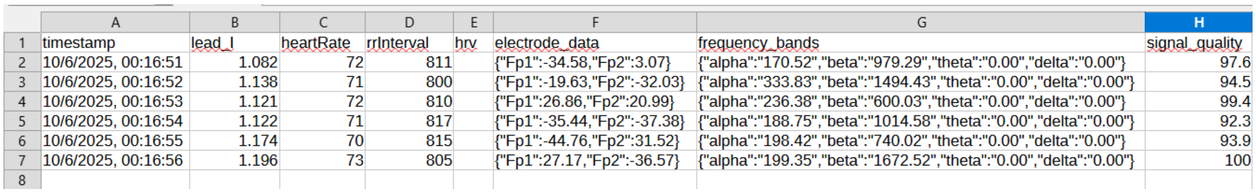
Fig 1.3: Simulated data visualization of EEG



Today

Name	Date modified	Type	Size
session-53	6/10/2025 12:16 AM	JSON Source File	4 KB
session-53	6/10/2025 12:16 AM	OpenOffice.org 1...	2 KB
session-52	6/10/2025 12:14 AM	JSON Source File	7 KB
session-52	6/10/2025 12:14 AM	OpenOffice.org 1...	3 KB
session-51	6/10/2025 12:04 AM	JSON Source File	40 KB
session-51	6/10/2025 12:04 AM	OpenOffice.org 1...	18 KB

Fig 1.4: ECG_EEG_Export folder containing exported data files



	A	B	C	D	E	F	G	H
1	timestamp	lead_I	heartRate	rrInterval	hrv	electrode_data	frequency_bands	signal_quality
2	10/6/2025, 00:16:51	1.082	72	811		{"Fp1":-34.58,"Fp2":3.07}	{"alpha":"170.52","beta":"979.29","theta":"0.00","delta":"0.00"}	97.6
3	10/6/2025, 00:16:52	1.138	71	800		{"Fp1":-19.63,"Fp2":-32.03}	{"alpha":"333.83","beta":"1494.43","theta":"0.00","delta":"0.00"}	94.5
4	10/6/2025, 00:16:53	1.121	72	810		{"Fp1":26.86,"Fp2":20.99}	{"alpha":"236.38","beta":"600.03","theta":"0.00","delta":"0.00"}	99.4
5	10/6/2025, 00:16:54	1.122	71	817		{"Fp1":-35.44,"Fp2":-37.38}	{"alpha":"188.75","beta":"1014.58","theta":"0.00","delta":"0.00"}	92.3
6	10/6/2025, 00:16:55	1.174	70	815		{"Fp1":-44.76,"Fp2":31.52}	{"alpha":"198.42","beta":"740.02","theta":"0.00","delta":"0.00"}	93.9
7	10/6/2025, 00:16:56	1.196	73	805		{"Fp1":27.17,"Fp2":-36.57}	{"alpha":"199.35","beta":"1672.52","theta":"0.00","delta":"0.00"}	100
8								

Fig 1.5: Example of CSV file data