

Sleep Monitor

AA 2024/2025

Giorgio Bonetti VR488066

Francesco Frison VR500307

Obiettivi

L'obiettivo principale del progetto è la realizzazione di un'applicazione web per **l'analisi dei dati relativi al sonno** di una persona. Attraverso la visualizzazione grafica dei dati raccolti (suddivisi in fasi come sonno leggero, profondo, REM e veglia), l'applicazione consente all'utente di comprendere l'andamento del proprio riposo nel tempo e identificare eventuali pattern o anomalie.

Funzionalità

L'applicazione implementa le seguenti funzionalità principali:

- **Caricamento e gestione dati:** l'utente può caricare i propri file CSV tramite un'interfaccia grafica; i dati vengono memorizzati in un database e associati all'utente autenticato.
- **Parsing:** i dati grezzi vengono processati per suddividerli per giorno e fase del sonno, calcolando anche la durata totale.
- **Visualizzazione tramite grafici:**
 - Un **grafico a torta** mostra le fasi del sonno per avere un'idea di quanto tempo per ogni fase si ha dormito
 - Un **grafico a punti** che mostra per ogni minuto la fase di sonno in cui si era, in modo che visivamente sia più impattante
 - Un **grafico a barre sovrapposte** mostra le diverse fasi del sonno per ciascun giorno
- **Adattamento del periodo visualizzato:** l'utente può visualizzare i dati in formato settimanale o mensile, con il grafico che si adatta automaticamente aggiungendo giorni mancanti se necessario.
- **Gestione multi-utente:** ogni utente può registrarsi ed effettuare l'accesso per gestire in modo indipendente i propri dati del sonno.
- **Consigli personalizzati:** grazie all'integrazione con una IA locale (*Ollama*), vengono generati suggerimenti su misura per migliorare la qualità del sonno in base ai pattern rilevati nei dati.

Tecnologie e Tecniche Utilizzate

Il progetto è stato sviluppato utilizzando:

- **React** con **TypeScript** per la struttura dell'interfaccia utente e la tipizzazione.
- **Vite** come bundler e ambiente di sviluppo.

- **G2Plot** (@antv/g2plot) per la realizzazione dei grafici interattivi.
- **Bootstrap** per il layout **responsive** e accessibile.
- **Papaparse** per il parsing dei file CSV.
- **Express.js** come **server back-end**.
- **Supabase / supabase-js** per il collegamento al database.
- **Crypto-js** per la cifratura delle password.
- **Dotenv** per avere un file di configurazione dove salvare le credenziali per l'accesso al database.
- **Html-to-image** per trasformare un elemento html in un'immagine.
- **Jspdf** per la generazione del pdf.
- **React-router-dom** per la navigazione tra più pagine.

Limitazioni Attuali

- **Assenza di analisi avanzate:** la logica attuale si basa su semplici correlazioni temporali; sarebbe possibile integrare modelli statistici o predittivi per approfondimenti clinici.
- **Ollama esegue analisi in locale:** questo limita la scalabilità dell'app in ambienti con risorse hardware ridotte.

Sviluppi Futuri Possibili

- Miglioramento dell'interfaccia utente e aggiunta di **personalizzazione** visiva.
- Ottimizzazione delle **prestazioni** del motore **IA** per l'analisi su grandi dataset.
- Esportazione dei dati e report PDF migliore di quello attuale.
- Integrazione con dispositivi **wearable** per acquisizione dati automatica.