

PROGETTO APPLICAZIONE IN REACT

Università di Verona – Maggio 2025

PRESENTAZIONE DEL PROGETTO

Il progetto in questione si pone come obiettivo quello di creare un'applicazione utilizzando il framework React per monitorare l'attività fisica durante la giornata di una persona.

In particolare, tramite opportune misure registrare in un file, è possibile visualizzare vari aspetti dell'attività fisica svolta nei vari giorni di rilevazione.

Oltre a questo, l'applicazione è in grado di generare degli esercizi fisici personalizzati in base alle preferenze dell'utente, comprendenti livello di difficoltà (beginner, intermedio e avanzato) e gruppo muscolare da allenare (braccia, spalle, schiena, gambe, petto).

INTERFACCIAMENTO INIZIALE

All'avvio dell'applicazione, viene mostrata una pagina in cui l'utente può inserire il file da cui prendere i dati e inserire il numero di passi da impostare come obiettivo.

Una volta eseguito il Submit del form iniziale, tutti i dati vengono salvati nel localStorage del browser in modo da poterli utilizzare in tutta l'applicazione e l'utente viene reindirizzato automaticamente alla pagina di Dashboard.

A light gray rounded rectangular form with a soft shadow. At the top, it says "Benvenuto!" in bold black text, followed by "Inserisci i tuoi dati per proseguire!" in smaller gray text. Below this, there are two sections: "Obiettivo giornalieri di passi:" with a text input field containing "0", and "File sorgente:" with a file selection button labeled "Scegli file" and a blue button labeled "Nessun file selezionato". At the bottom center is a blue "Invia" button.

Benvenuto!

Inserisci i tuoi dati per proseguire!

Obiettivo giornalieri di passi:

File sorgente:

DASHBOARD

Una volta inseriti i dati iniziali, l'utente viene reindirizzato nella pagina iniziale. Qui si possono vedere varie informazioni quali:

- Obiettivo dei passi
- Selezione della data per le informazioni dello storico
- Riepilogo dell'andamento (sia tramite dati che grafici)




OBIETTIVO PASSI

Durante l'interfacciamento iniziale, l'utente ha inserito il proprio numero di passi che vuole impostare come obiettivo. Tale informazione viene mostrata sulla dashboard e presa in considerazione per la visualizzazione di dati e grafici in confronto ai passi svolti.

Se l'utente non raggiunge il numero di passi giornaliero, viene informato tramite una notifica.

 Obiettivo passi: **5000**


Dati del Giorno


 Passi: 17487


 Distanza: 11633 m

 Calorie: 594.36 kCal


Progressi settimanali:


 Passi: 51729

 Distanza: 33.87 km

 Calorie: 1878.48 kCal

Progressi mensili:

 Passi: 100222

 Distanza: 64.61 km


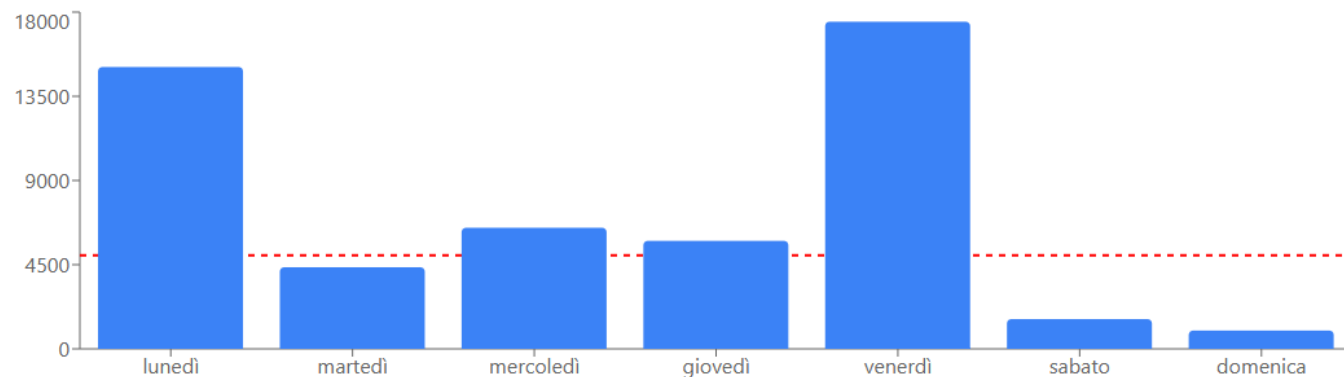
 Calorie: 3728.44 kCal

Grafico Passi

👉 Completamento Obiettivo



Andamento settimanale passi:



RIEPILOGO ANDAMENTO

In qualsiasi momento, è possibile visualizzare l'andamento in base alla data scelta. Oltre i dati del giorno, si possono consultare i grafici presenti per capire il procedimento dei passi lungo il periodo trascorso prima della data in questione. Gli schemi rappresentano il giorno stesso, la settimana e il mese percorsi.

LIBRERIE E TECNOLOGIE UTILIZZATE

- Tailwind CSS: libreria per una stilizzazione più semplice e veloce
- Lucid-react: per l'utilizzo di icone ottimizzate
- Date-fns: per l'utilizzo di funzioni ottimizzate per gli oggetti di tipo Date
- React con TypeScript: per la struttura dell'interfaccia utente
- Vite: come bundler e ambiente di sviluppo
- Papaparse: per eseguire in modo semplice il parsing del file caricato manualmente dall'utente
- Recharts: per la costruzione di grafici

CHALLENGE RISOLTE

- Come eseguire il parsing in modo efficiente del file: in definitiva abbiamo scoperto l'esistenza della libreria 'Papaparse' che ci ha aiutati in questo.
- Come caricare automaticamente un file: per fare ciò, abbiamo usato la libreria vista a lezione 'axios' per eseguire una 'GET' per reperire il file, successivamente 'parsato' con Papaparse.
- Passaggio dei dati tra le varie pagine: per risolvere questo problema abbiamo pensato di salvare i dati principali (quelli che fornisce l'utente nella pagina di Landing) nel 'localStorage' del browser.