



Samb, Arici, Mattia, Brembati

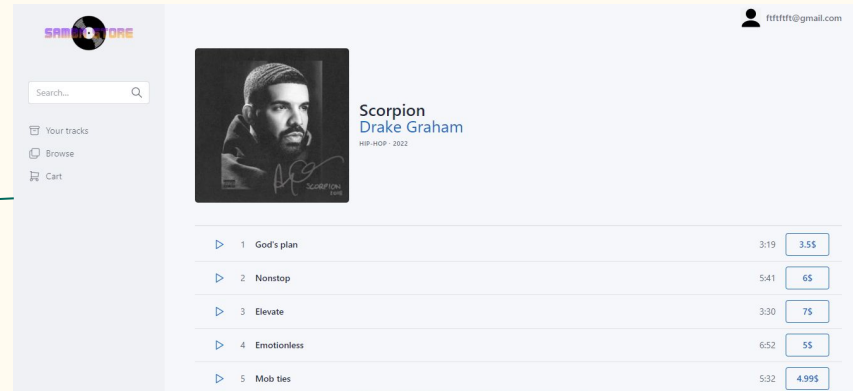
# La consegna

- Realizzazione di un sito web in grado di gestire l'acquisto online di tracce musicali
- Utilizzo di Firestore e Realtime, entrambi database di Firebase
- Utilizzo di metodi CRUD al fine di eseguire diverse funzioni presenti nel progetto

# Analisi

- L'analisi della consegna ha portato alla produzione di un modello E/R che prevede le seguenti entità:
  - Cliente
  - Brano
  - Album
  - Artista
- L'effettivo modello non è stato utilizzato, essendo Firestore un database NoSQL, ma ha permesso di comprendere la realtà da analizzare e ha dato una chiara idea di come strutturare i nuovi database
- Un esempio è la scelta di poter comprare solamente i brani e non interi album

# Reperibilità del progetto



due repository Github diversi

# Funzionalità implementate nel sito

- Possibilità di acquistare brani
- Ricerca completa di brani, artisti, album e genere (per nome, genere, stato)
- Funzione per l'anteprima brani
- Pagina per visualizzare tutti i brani comprati
- Sezione di annunci/novità
- Pagina per visualizzare il carrello

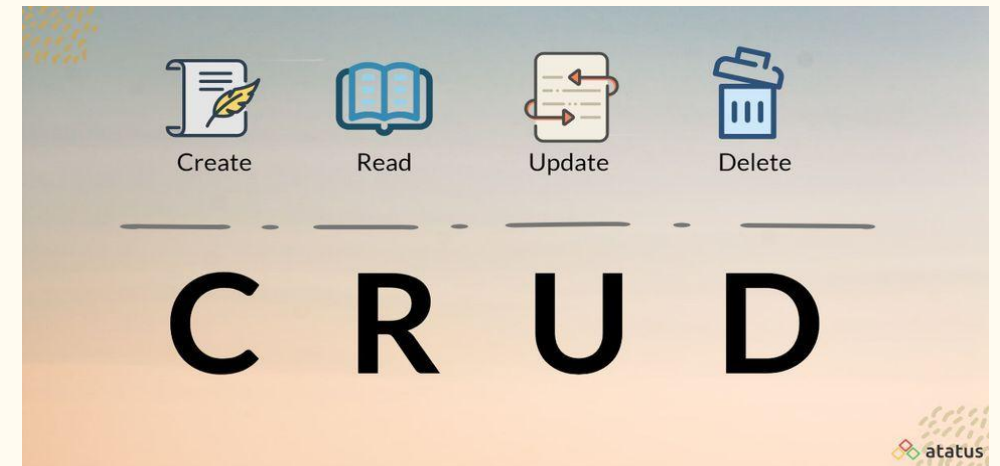
# Differenze Realtime - Firestore

Realtime DB	Firestore
Archivia i dati come un grande albero JSON.	Memorizza i dati come raccolte di documenti.
I dati complessi e gerarchici sono più difficili da organizzare su larga scala.	I dati complessi e gerarchici sono più facili da organizzare su larga scala, utilizzando sottoraccolte all'interno dei documenti.
Può registrare lo stato della connessione del client e fornire aggiornamenti ogni volta che lo stato della connessione del client cambia.	Non supportato in modo nativo.
Query profonde	Query superficiali
Fino a 200.000 connessioni simultanee	Fino a 1.000.000 connessioni simultanee
Fino a 1.000 scritture al secondo	Fino a 10.000 scritture al secondo
Addebiti solo per larghezza di banda e spazio di archiviazione	Addebiti principalmente sulle operazioni eseguite nel database

# Il CRUD

Le operazioni del CRUD ci consentono di:

- C – create - aggiungere elementi al database;
- R – read - leggere specifici elementi del database;
- U – update - modificare gli elementi;
- D – delete – rimuovere elementi dal database.



# Conclusioni

- Il progetto ha permesso l'apprendimento di nuovi database con un funzionamento diverso rispetto a quelli SQL già trattati, oltre a migliorare la conoscenza di javascript
- Il database Realtime si è presentato molto veloce e più semplice da implementare, al contrario di Firestore che è risultato più lento e complesso
- La suddivisione del lavoro è risultata equa per tutti



<https://samba-store-realtime.netlify.app/browse.html>

—