

Tweb Report

Valerio Gozzellino - Alessandro Mao

February 2, 2024

1 Introduzione

Questo documento dettaglia le decisioni implementative e gli aspetti tecnici distintivi del nostro progetto di applicazione web dedicata al calcio. Esso offre aggiornamenti in tempo reale, dati approfonditi su partite, giocatori e competizioni, e si propone come un punto di riferimento per gli appassionati del pallone.

2 Soluzione del Frontend:

Per quanto riguarda l'architettura del frontend che è stata sviluppata con l'utilizzo di React in quanto permette una facile implementazioni e gestione del codice ed una elevatissima personalizzazione del sito, il nostro approccio è stato quello di suddividere le pagine in più livelli, implementando una struttura modulare. Abbiamo organizzato il codice seguendo una gerarchia di componenti, composte da atomi e card. Questa metodologia ci ha permesso di mantenere il codice ordinato e facilmente gestibile. Inoltre, con l'intento di garantire pulizia e coerenza del codice, abbiamo istituito una directory dedicata esclusivamente alla gestione delle richieste verso i vari server, assicurando così una separazione chiara e una maggiore manutenibilità del nostro progetto.

2.1 Autenticazione e Gestione della Sessione:

L'autenticazione è fondamentale nelle sezioni dell'applicazione che richiedono l'accesso a dati personali o sensibili. Il nostro flusso di login è stato progettato per soddisfare questa necessità critica di sicurezza.

2.2 Implementazione del Login:

Abbiamo sviluppato un sistema di registrazione e autenticazione che non solo facilita l'interazione con la nostra chat, ma offre anche un'esperienza personalizzata agli utenti registrati. Prevediamo di arricchire ulteriormente questa personalizzazione con aggiornamenti automatici e vantaggi esclusivi per gli utenti registrati in futuro. Nonostante ciò, il nostro sito rimane accessibile anche agli utenti non registrati, permettendo loro di consultare le ultime notizie e informazioni sulle partite senza la necessità di un account.

2.3 Design e Personalizzazione:

l'implementazione del design visivo è stata fatta mediante l'uso di CSS e componenti personalizzate, e le strategie per arricchire l'esperienza utente.

2.4 Gestione del Loading:

Approfondimento sull'uso di useMemo per ottimizzare il caricamento delle pagine e garantire un'interfaccia reattiva, inoltre è stata implementato un meccanismo di loading che evita che la pagina sia visualizzata prima che sia effettivamente montata.

2.5 Chat e Interazione:

La pagina è stata progettata con tecnologia WebSocket per fornire un'esperienza di messaggistica in tempo reale. Sono disponibili tre stanze di chat distinte, ognuna dedicata a temi di discussione specifici, per facilitare conversazioni organizzate e mirate. L'accesso alla chat è riservato agli utenti autenticati e può essere effettuato tramite un'icona di chat che è facilmente accessibile da qualsiasi parte del sito.

3 Soluzione Persistenza dei Dati e Architettura Server

3.1 Persistenza dei dati:

Persistenza dei Dati e Architettura Server Nella progettazione della nostra architettura server, abbiamo adottato un approccio bifocale per la persistenza dei dati, utilizzando MongoDB e PostgreSQL per soddisfare le diverse esigenze del nostro sistema. PostgreSQL, con le sue capacità transazionali e il modello di dati relazionale, è stato scelto per gestire dati statici e fortemente strutturati come competizioni, club e informazioni sui giocatori. La sua affidabilità e integrità referenziale lo rendono ideale per assicurare la consistenza di dati che richiedono relazioni definite e operazioni CRUD rigorose.

D'altra parte, abbiamo sfruttato la flessibilità e la scalabilità di MongoDB per memorizzare dati dinamici e in rapida evoluzione, quali i dettagli delle partite, gli eventi di gioco e le prestazioni dei giocatori nelle partite. La natura schema-less di MongoDB ci permette di adattarci facilmente a strutture di dati complesse e in continua evoluzione, ottimizzando le operazioni di lettura/scrittura per dati che presentano variazioni frequenti e che beneficiano di una modellazione meno rigida.

3.2 ArchitetturaServer:

Per orchestrare l'interazione tra il client e i sistemi di gestione dei database, abbiamo implementato una costellazione di server. Il Main Server funge da intermediario, ricevendo le richieste del client e instradandole ai server dedicati. Questi ultimi, configurati specificamente per interagire con MongoDB e PostgreSQL, gestiscono efficientemente le operazioni sui rispettivi database. Questa separazione dei compiti non solo chiarifica la struttura del back-end ma garantisce anche che le operazioni sui dati siano eseguite in modo ottimale, con la possibilità di bilanciare il carico e di scalare orizzontalmente i servizi in base alla domanda.

Analisi delle strutture server e del loro ruolo nella gestione dei dati persistenti, con particolare enfasi sulle richieste eseguite al database Postgres e al MongoServer. Modularizzazione del Codice: Illustrazione dell'organizzazione del codice server e della segmentazione in moduli per massimizzare l'efficienza e la manutenibilità, inoltre grazie a questa sua caratteristica potrebbe garantire una scalabilità efficiente in caso di ampio traffico di utenti. Per quanto riguarda Springboot abbiamo i pochi endpoint in un unico controller poi suddividendo la gestione delle query nei vari service e successivi repository. Invece per MongoDB abbiamo deciso di suddividere in routes per i gruppi di dati simili e utilizzare i modelli che si riferiscono ai dati nel database.

4 Suddivisione dei compiti e Documentazione

4.1 Suddivisione dei compiti:

Per realizzare il progetto in modo efficiente e rapido, abbiamo adottato un approccio basato sulla suddivisione dei compiti relativi ai server. Essendo il nostro team composto da due persone, abbiamo diviso le responsabilità iniziali: uno si è dedicato alla configurazione di Postgres e l'altro a quella di MongoDB. Questo passaggio iniziale ha favorito una specializzazione preliminare, ma nel corso del progetto entrambi i membri hanno collaborato attivamente su entrambi i server.

4.2 Documentazione:

Riguardo alla documentazione, abbiamo assicurato che il codice fosse accuratamente documentato, descrivendo ogni endpoint, i parametri richiesti e i valori restituiti.

5 Extra information:

Per migliorare l'esperienza dell'utente abbiamo inserito una richiesta ad una API esterna che si occupa di news calcistiche, all'interno del progetto è fornita la chiave che permette di ottenere i dati. Inoltre è stato modificato il database postgres per migliorare la visualizzazione dei dati. I dataset sono forniti all'interno del progetto.

6 Bibliografica:

La documentazione del nostro progetto si è avvalsa di un'ampia gamma di fonti autorevoli, inclusi siti educativi di riferimento come W3Schools, forum per sviluppatori come Stack Overflow, e la robusta libreria di componenti MUI di Material-UI. In aggiunta, abbiamo utilizzato l'intelligenza artificiale per il debugging del codice, migliorando significativamente la precisione.

7 Conclusione:

Il nostro sito web è l'ideale per gli appassionati di calcio. Offre funzionalità come aggiornamenti in tempo reale delle partite, le ultime notizie e spazi di discussione per gli utenti. Ci impegniamo a rendere la nostra applicazione un punto di riferimento quotidiano per gli utenti, garantendo un'esperienza utente coinvolgente e soddisfacente.