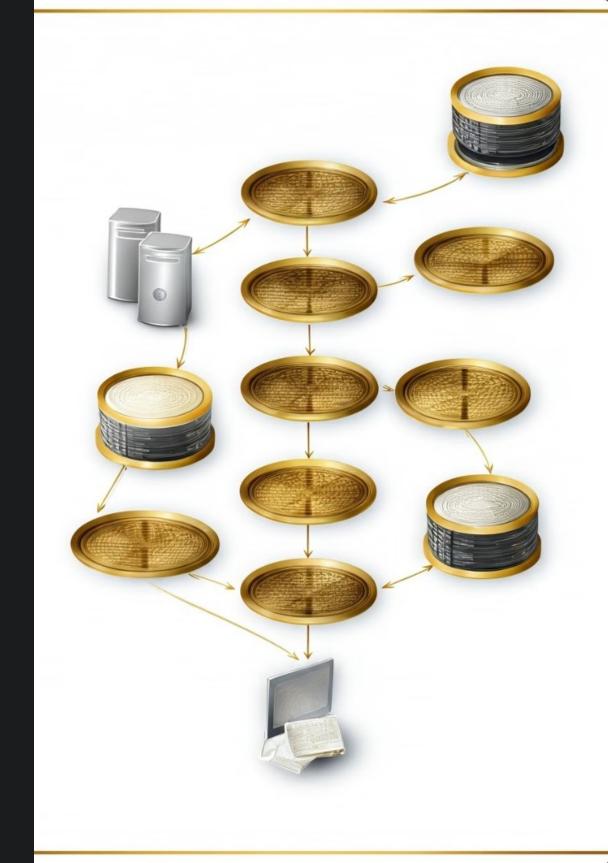
Sistema di Ticketing e Prenotazione da Remoto

Questa presentazione delinea un sistema client/server multi thread per la gestione remota di ticketing e prenotazioni per eventi. Utilizzando MongoDB come database, questo sistema mira a fornire un'esperienza senza interruzioni ed efficiente per gli utenti.





Vantaggi di un Sistema di Ticketing da Remoto

1 Accessibilità

Gli utenti possono acquistare biglietti da qualsiasi luogo con una connessione internet, indipendentemente dalla posizione.

2 Comodità

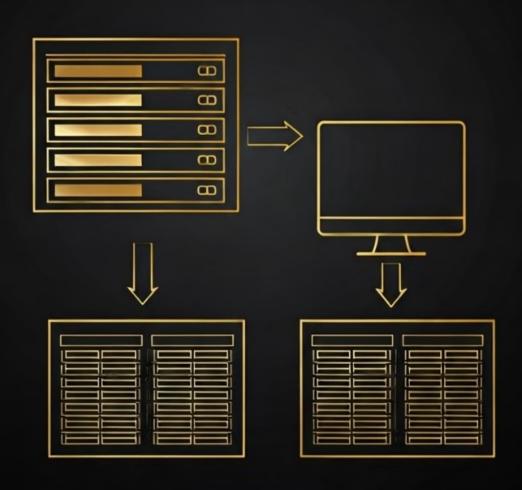
Gli utenti possono accedere al sistema 24/7 e gestire i propri biglietti e prenotazioni a loro piacimento.

3 Aggiornamenti in tempo reale

Gli utenti possono vedere i posti disponibili, i prezzi dei biglietti e le informazioni sugli eventi in tempo reale.

4 Efficienza

Il sistema snellisce il processo di ticketing, riducendo i tempi di attesa e la burocrazia.



MongoDB come Database scelto

Scalabilità

MongoDB può gestire grandi quantità di dati e traffico utente, garantendo un funzionamento regolare anche durante i periodi di punta.

Flessibilità

La struttura basata su documenti di MongoDB consente una modellazione flessibile dei dati, adattandosi a informazioni diverse sugli eventi e sui dati degli utenti.

Performance

Le capacità di indicizzazione e ottimizzazione delle query di MongoDB garantiscono un recupero rapido dei dati, offrendo un'esperienza utente reattiva.

Architettura del Sistema e Componenti

Client

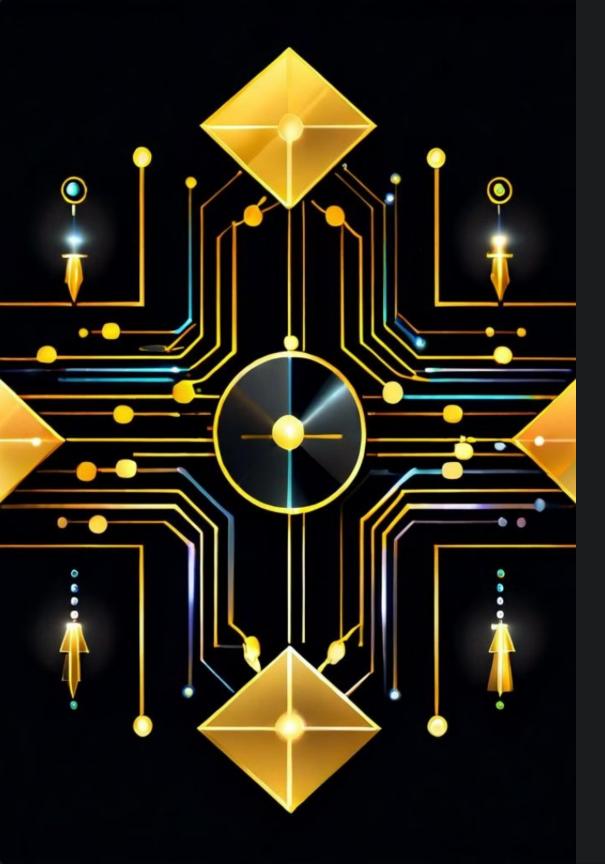
L'interfaccia utente permette agli utenti di navigare tra gli eventi, acquistare biglietti e gestire le proprie prenotazioni.

Server

Il server gestisce le richieste dei client, amministra il database e processa le transazioni.

Database

MongoDB memorizza i dati degli eventi, le informazioni sugli utenti, gli acquisti dei biglietti e i dettagli delle prenotazioni.



Gestione delle Richieste dei Client e delle Risposte del Server

ı

Richiesta

Il client invia una richiesta al server, specificando l'azione desiderata, come visualizzare i dettagli di un evento o acquistare biglietti.

Elaborazione

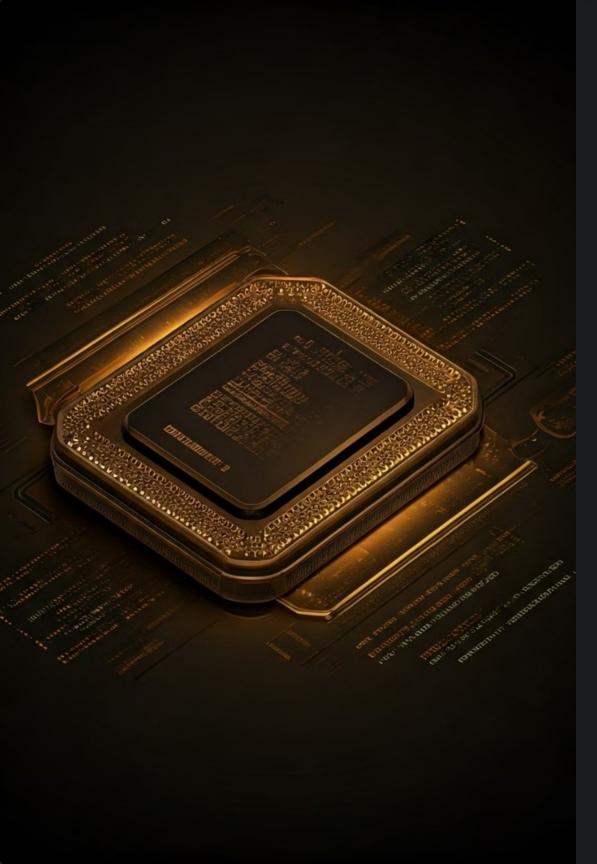
Il server riceve la richiesta, la convalida e la elabora utilizzando il database.

Risposta

Il server invia una risposta al client, fornendo le informazioni richieste o confermando il completamento dell'azione.

_

3



Implementazione della Concorrenza e Gestione dei Thread

	estire efficientemente le richieste encorrenti dei client.
lock	engono utilizzati meccanismi come ck e semafori per prevenire la orruzione dei dati quando più threac ccedono a risorse condivise.

Gestione dei Thread

Il sistema gestisce la creazione, la

pianificazione e la terminazione dei

thread per ottimizzare le prestazioni

e l'uso delle risorse.

Memorizzazione e Recupero Dati in MongoDB



Dati sugli eventi

Ogni evento è memorizzato come documento contenente dettagli sull'evento, date, luogo e tipi di biglietti.



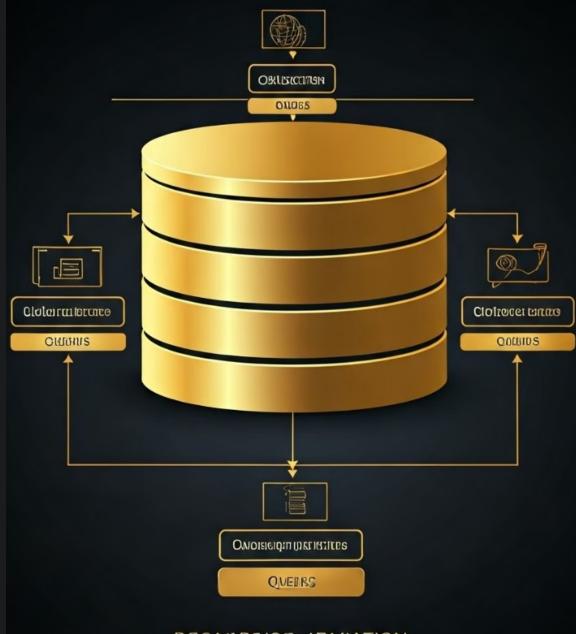
Informazioni sugli utenti

I profili degli utenti sono memorizzati con i loro dettagli di contatto, storico degli acquisti e informazioni sulle prenotazioni.



Biglietti

Gli acquisti dei biglietti sono registrati con dettagli sull'utente, evento, tipo di biglietto.



DECA/ABS IOB CIEMMATION

Lowern pastern atherisans until you about their constant and their and anomalian religious religions to the constant the constant their constant to the constant their constant to the constant constant the constant to the c

Conclusione e Miglioramenti Futuri

Questo sistema client/server multi thread offre una soluzione robusta per la gestione remota di biglietti e prenotazioni per eventi. I miglioramenti futuri potrebbero includere l'integrazione con applicazioni mobili, aggiornamenti in tempo reale sulla disponibilità dei posti e analisi avanzate per la gestione degli eventi.

