# Documentazione Progetto BookMind

## Panoramica

BookMind è un'applicazione web che offre raccomandazioni di libri personalizzate basate sulle preferenze dell'utente. Utilizza l'intelligenza artificiale per generare suggerimenti di lettura e offre un'interfaccia utente intuitiva per la gestione delle preferenze e la visualizzazione delle raccomandazioni.

## Tecnologie Utilizzate

* Frontend: Vue.js 3 con Composition API
* Linguaggio: TypeScript
* Styling: Tailwind CSS
* State Management: Pinia
* Routing: Vue Router
* Build Tool: Vite
* Testing: Vitest
* Autenticazione: Firebase Authentication
* Database: Firebase Firestore (per backup remoto) e IndexedDB (per storage locale)
* AI: OpenAI API per generazione raccomandazioni

## Struttura del Progetto

Copy

bookmind/

├── public/

├── src/

│ ├── assets/

│ ├── components/

│ │ └── layout/

│ │ └── Menu.vue

│ ├── router/

│ │ └── index.ts

│ ├── services/

│ │ ├── firebase/

│ │ │ └── config.ts

│ │ ├── indexedDB/

│ │ │ └── userPreferences.ts

│ │ └── openai/

│ │ ├── bookRecommendation.ts

│ │ ├── config.ts

│ │ └── usageMonitor.ts

│ ├── stores/

│ │ └── auth.ts

│ ├── types/

│ │ └── userPreferences.ts

│ ├── views/

│ │ ├── HomePage.vue

│ │ ├── LoginPage.vue

│ │ ├── RegisterPage.vue

│ │ └── UserPreferences.vue

│ ├── App.vue

│ └── main.ts

├── .eslintrc.js

├── .gitignore

├── index.html

├── package.json

├── README.md

├── tailwind.config.js

├── tsconfig.json

├── tsconfig.app.json

├── tsconfig.node.json

├── tsconfig.vitest.json

├── vite.config.ts

└── vitest.config.ts

## Componenti Principali

### App.vue

Componente radice dell'applicazione. Gestisce il layout principale e il routing.

### HomePage.vue

Pagina principale che mostra il benvenuto all'utente e gestisce la richiesta e la visualizzazione delle raccomandazioni di libri.

### LoginPage.vue

Gestisce l'autenticazione dell'utente.

### RegisterPage.vue

Gestisce la registrazione di nuovi utenti.

### UserPreferences.vue

Permette agli utenti di impostare e modificare le loro preferenze di lettura.

### Menu.vue

Componente di navigazione principale dell'applicazione.

## Servizi

### Firebase (config.ts)

Configura e inizializza Firebase per l'autenticazione e Firestore.

### IndexedDB (userPreferences.ts)

Gestisce il salvataggio e il recupero delle preferenze utente e dell'ultima raccomandazione nel database locale IndexedDB.

### OpenAI (bookRecommendation.ts, config.ts, usageMonitor.ts)

Gestisce le chiamate all'API di OpenAI per generare raccomandazioni di libri e monitora l'utilizzo dell'API.

## Store

### auth.ts

Store Pinia per la gestione dello stato di autenticazione dell'utente.

## Tipi

### userPreferences.ts

Definisce l'interfaccia e le funzioni di validazione per le preferenze dell'utente.

## Configurazione

### vite.config.ts

Configura Vite per il building e lo sviluppo dell'applicazione.

### tailwind.config.js

Configura Tailwind CSS per lo styling dell'applicazione.

### tsconfig.json e varianti

Configurano TypeScript per il progetto.

## Funzionalità Principali

1. Autenticazione Utente
   * Registrazione nuovo utente
   * Login utente esistente
   * Logout
2. Gestione Preferenze Utente
   * Impostazione preferenze di lettura (genere, lunghezza libro, periodo, complessità, scopo)
   * Salvataggio preferenze in IndexedDB e sincronizzazione con Firestore
3. Generazione Raccomandazioni Libri
   * Utilizzo dell'API OpenAI per generare raccomandazioni basate sulle preferenze dell'utente
   * Salvataggio dell'ultima raccomandazione in IndexedDB
4. Persistenza Dati
   * Utilizzo di IndexedDB per storage locale delle preferenze e dell'ultima raccomandazione
   * Backup su Firestore per sincronizzazione tra dispositivi
5. Interfaccia Utente Reattiva
   * Design responsive con Tailwind CSS
   * Navigazione fluida tra le pagine con Vue Router
6. Monitoraggio Utilizzo API
   * Tracking del numero di richieste e token utilizzati per l'API OpenAI

## Flusso dell'Applicazione

1. L'utente accede alla pagina di login o registrazione
2. Dopo l'autenticazione, viene reindirizzato alla HomePage
3. Se è la prima visita, l'utente viene incoraggiato a impostare le preferenze di lettura
4. Nella HomePage, l'utente può richiedere una raccomandazione di libro
5. L'app genera una raccomandazione utilizzando l'API OpenAI e la mostra all'utente
6. L'ultima raccomandazione viene salvata e mostrata al successivo accesso

## Sicurezza

* Autenticazione gestita da Firebase
* Dati sensibili (come le chiavi API) sono memorizzati in variabili d'ambiente
* Validazione delle input utente sia lato client che lato server

## Performance

* Lazy loading dei componenti Vue per ottimizzare il caricamento iniziale
* Utilizzo di IndexedDB per ridurre le chiamate al backend
* Caching delle raccomandazioni per ridurre le chiamate API

## Prossimi Passi

1. Implementazione della configurazione Amazon per i link ai libri
2. Sviluppo di funzionalità per link dinamici ai libri raccomandati
3. Miglioramento del sistema di caching per le raccomandazioni
4. Implementazione di test unitari e di integrazione
5. Ottimizzazione delle performance e dell'accessibilità