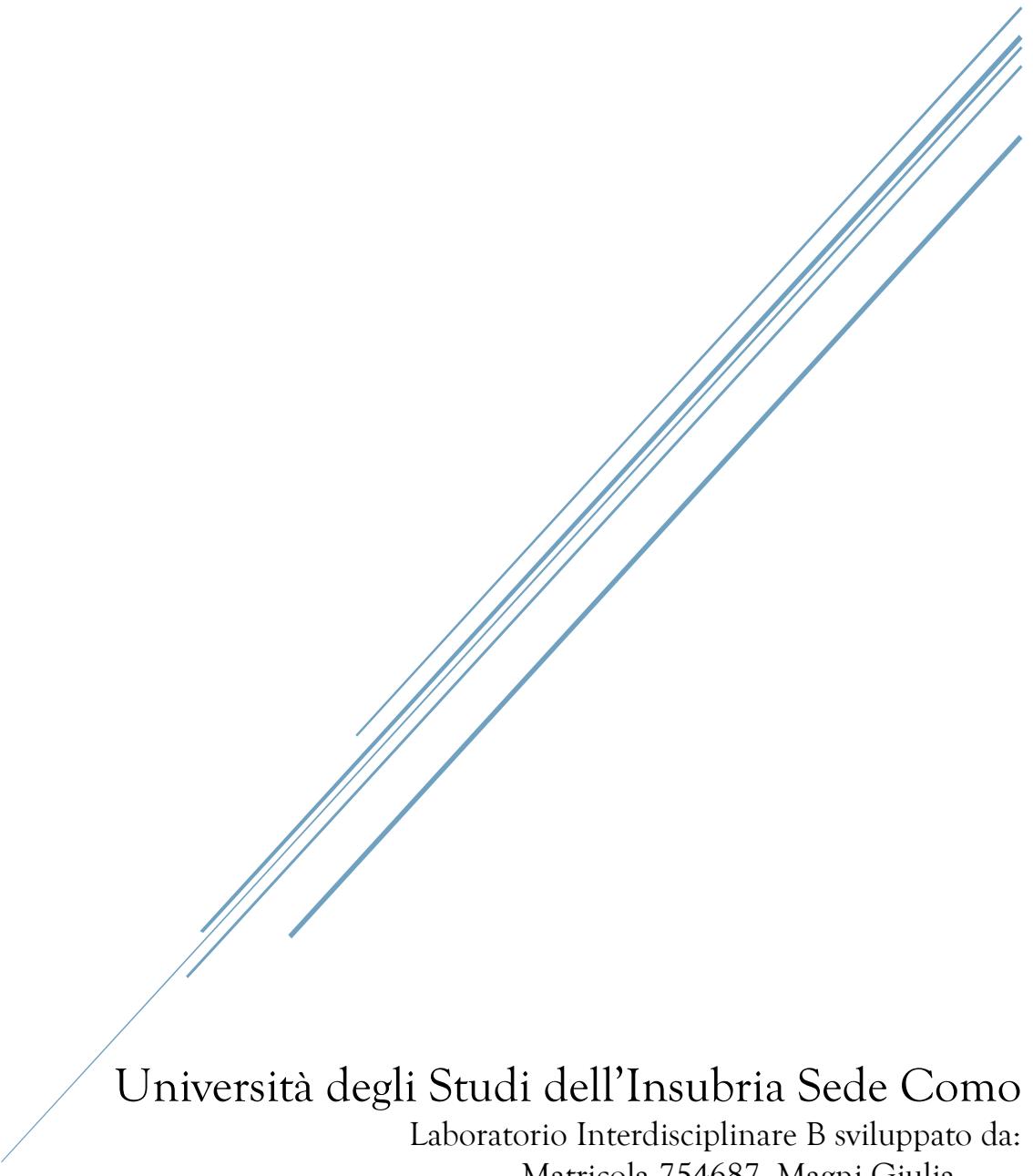


BOOK RECOMMENDER

MANUALE UTENTE

versione 2.0.B | Data: luglio 2025



Università degli Studi dell'Insubria Sede Como

Laboratorio Interdisciplinare B sviluppato da:

- Matricola 754687, Magni Giulia -
- Matricola 758987, Basso Leonardo -
- Matricola 754625, Beretta Lorenzo -



Indice

1. Introduzione	2		
2.	Requisiti di sistema	2	
3.	Installazione	2	
3.1	Setup ambiente	2	
3.2	Installazione programma	3	
❖	Frontend	3	
❖	Backend & DB	4	
4.		Esecuzione ed uso	5
4.1	Uso delle funzionalità	5	
👤	Utente generico (non registrato)	5	
👤	Utente registrato (login richiesto)	5	
5.	Programma	5	
Schermata menu:	5		
Schermata registrazione:	6		
Schermata Utente:	6		
Pop-Up Libreria:	7		
Pop-Up per aggiungere un libro alla libreria:	7		
Pop-Up per scegliere un libro da consigliare:	7		
Schermata della propria libreria:	7		
Pop-Up Valutazione:	8		
Schermata valutazione, commenti e libri consigliati:	8		
6.		Limiti della soluzione sviluppata	9
7.		Sitografia / Bibliografia	9
❖	POSTGRESQL - https://www.postgresql.org/	9	
Caratteristiche principali	9		
Perché usarlo?	9		
Vuoi approfondire?	9		
❖	APACHE MAVEN - https://maven.apache.org/	10	
Cosa fa Maven?	10		
Il cuore di Maven: il file pom.xml	10		
Perché è utile?	10		
❖	TAURI DOCUMENTATION – https://tauri.app/	11	
Documentazione ufficiale	11		
Caratteristiche principali	11		
❖	VUE.JS OFFICIAL GUIDE – https://vuejs.org/	11	
Dove trovarla	11		
Cosa contiene	11		
❖	JAVALIN – https://javalin.io/	12	
Caratteristiche principali	12		
Vuoi approfondire?	12		

1. Introduzione

Benvienuto in Book Recommender, l'app che consente di consultare, valutare e consigliare libri.

Pensata in grado di permettere agli utenti registrati di inserire recensioni ed una gestione personalizzata delle librerie digitali, mentre a tutti gli utenti di consultare le valutazioni e ricevere consigli di lettura.

In questo manuale troverai istruzioni passo-passo per installare, configurare e utilizzare tutte le funzionalità principali.

2. Requisiti di sistema

- ❖ **Sistema operativo:** Windows, Linux
 - ❖ **Java:** versione 17 o compatibile
 - ❖ **Database:** PostgreSQL installato e configurato
 - ❖ **Strumenti aggiuntivi:** Apache Maven o Ant per compilazione
 - ❖ **RAM consigliata:** 4 GB minimo
-

3. Installazione

3.1 Setup ambiente

1. **Installare Java JDK 17**
Scaricabile da <https://jdk.java.net/17/>
2. **Installare Apache Maven**
Da <https://maven.apache.org/download.cgi>
Verificare l'installazione con: mvn -v
3. **Installare Node.js e Rust**
Node.js ≥ 18: [Node.js – Run JavaScript Everywhere](#)
Rust ≥ 1.70: [Rust Programming Language](#)
4. **Installare Docker** per eseguire tutto in contenitori isolati
Docker Desktop (Windows)
Docker Engine (Linux)
5. **(Consigliato) Installare un IDE Java** come IntelliJ IDEA, WebStorm, Eclipse, VS Code o altri
6. Maggiori informazioni sul processo di installazione sono riportate sul file README.md nella repository GitHub del progetto oppure proseguire nel paragrafo successivo 3.2 Installazione programma.

3.2 Installazione programma

❖ Frontend

Vi si potrà accedere tramite eseguibile situato nella cartella bin per:

- ◆ **Linux:** (usare install.sh)
.rpm, .deb(non testato)
- ◆ **Windows:**
.msi, .exe

Questa parte è utile se si vuole compilare il progetto da sorgente.

Installare le dipendenze del programma:

- ◆ **Linux:**

◊ Debian/Ubuntu:

```
sudo apt update  
sudo apt install libwebkit2gtk-4.0-dev \  
    build-essential \  
    curl \  
    wget \  
    file \  
    libssl-dev \  
    libgtk-3-dev \  
    libayatana-appindicator3-dev \  
    librsvg2-dev
```

◊ Arch:

```
sudo pacman -Syu  
sudo pacman -S --needed \  
    webkit2gtk \  
    base-devel \  
    curl \  
    wget \  
    file \  
    openssl \  
    appmenu-gtk-module \  
    gtk3 \  
    libappindicator-gtk3 \  
    librsvg \  
    libvips
```

◊ Fedora:

```
sudo dnf check-update  
sudo dnf install webkit2gtk4.0-devel \  
    openssl-devel \  
    curl \  
    wget \  
    file \  
    libappindicator-gtk3-devel \  
    librsvg2-devel  
sudo dnf group install "C Development Tools and Libraries"
```

◊ OpenSuse:

```
sudo zypper up  
sudo zypper in webkit2gtk3-soup2-devel \  
    libopenssl-devel \  
    curl \  
    wget \  
    file \  
    libappindicator3-1 \  
    librsvg-devel  
sudo zypper in -t pattern devel_basis
```

È possibile installare Rust con:

```
curl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh
```

- ◆ **Windows:**

È necessario installare i Build Tools di C++ e le webview2 di rust con:

```
winget install --id RustLang.Rustup  
rustup default stable-msvc
```

Per far funzionare la prima volta il programma usare:

```
cd frontend  
npm install
```

||PRIMO AVVIO||

☞ Importante:

È necessario avere Node installato sul proprio computer.

Per far partire il programma usare: npx tauri dev

Il programma è in Developer Mode.

❖ Backend & DB

||PRIMO AVVIO||

☞ Importante:

È necessario avere Docker installato sul proprio computer.

Prima di utilizzare Docker è necessario generare il jar andando nella cartella di backend ed eseguendo: mvn clean package oppure si può fare anche graficamente da un IDE.

Su Linux (SystemD) è necessario avviare Docker con: sudo systemctl start docker

Su Windows tutti i comandi citati vanno eseguiti da amministratore.

Per avviare il progetto usare nel terminale:

- ◆ **Linux:**

```
sudo docker compose up --build
```

- ◆ **Windows (Powershell):**

```
docker-compose up --build
```

Se vengono apportati cambiamenti al progetto non rilevati da docker, usare:

- ◆ **Linux:**

```
sudo docker system prune -a  
sudo docker compose up --build
```

- ◆ **Windows (Powershell):**

```
docker system prune -a  
docker-compose up -build
```

Per interrompere Docker è necessario scrivere:

- ◆ **Linux:**

```
sudo docker compose down
```

- ◆ **Windows (Powershell):**

```
docker-compose down
```

4. Esecuzione ed uso

4.1 Uso delle funzionalità

L'applicazione si presenta con un **menu principale** chiamato Home, accessibile a tutti.

Le funzioni sono suddivise per tipo di utente:

Utente generico (non registrato):

- Cerca libri per titolo, autore, anno
- Visualizza informazioni e valutazioni aggregate
- Visualizza suggerimenti di lettura dati da altri utenti
- Possibilità di registrarsi tramite l'apposita funzione

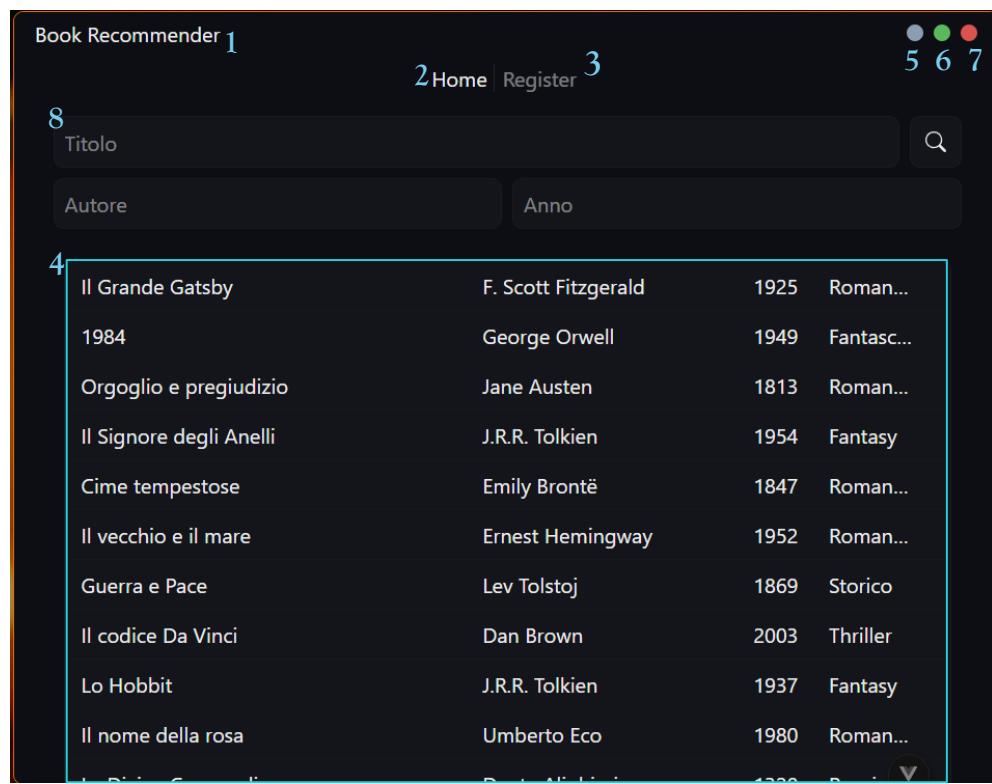
Utente registrato (login richiesto):

- Login
- Crea librerie personali con la possibilità di inserire i libri scelti
- Inserisci valutazioni su libri
- Suggerisci altri libri correlati (fino a 3 per libro)

5. Programma

Schermata menu:

1. Nome applicazione
2. Schermata principale
3. Pagina di registrazione utenti
4. Lista di libri presenti nell'applicazione
5. Diminuzione schermata
6. Ingrandimento schermata
7. Chiusura applicazione
8. Barre di ricerca

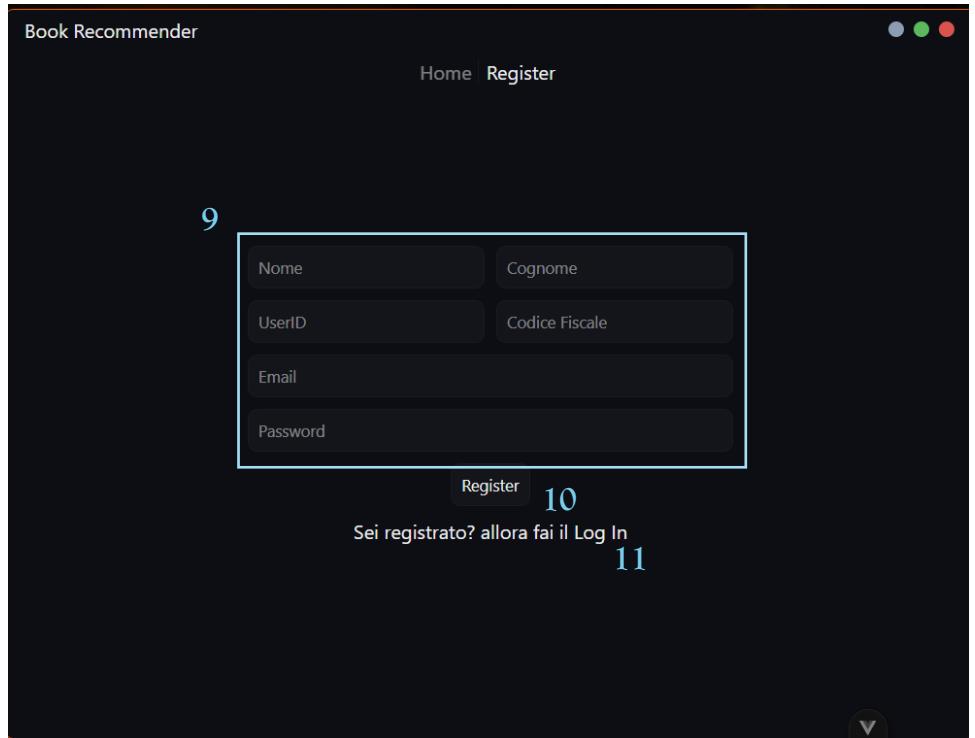


The screenshot shows the main screen of the Book Recommender application. At the top, there is a navigation bar with the title "Book Recommender" and icons for "Home" (highlighted in blue), "Register", and three dots for more options. Below the navigation bar is a search bar with the placeholder "Titolo" and a magnifying glass icon. Underneath the search bar are two input fields: "Autore" and "Anno". The main content area displays a list of books in a table format. The columns are: Titolo, Autore, Anno, and Genere. The books listed are:

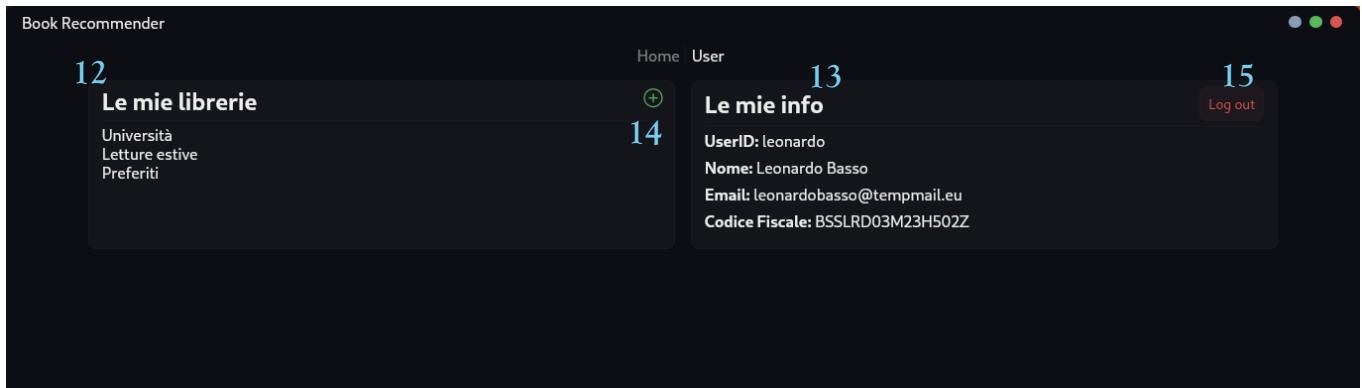
Titolo	Autore	Anno	Genere
Il Grande Gatsby	F. Scott Fitzgerald	1925	Roman...
1984	George Orwell	1949	Fantasc...
Orgoglio e pregiudizio	Jane Austen	1813	Roman...
Il Signore degli Anelli	J.R.R. Tolkien	1954	Fantasy
Cime tempestose	Emily Brontë	1847	Roman...
Il vecchio e il mare	Ernest Hemingway	1952	Roman...
Guerra e Pace	Lev Tolstoj	1869	Storico
Il codice Da Vinci	Dan Brown	2003	Thriller
Lo Hobbit	J.R.R. Tolkien	1937	Fantasy
Il nome della rosa	Umberto Eco	1980	Roman...

Schermata registrazione:

9. Caselle in cui inserire i dati necessari per la registrazione
10. Tasto di conferma della registrazione
11. Link per il login del proprio account



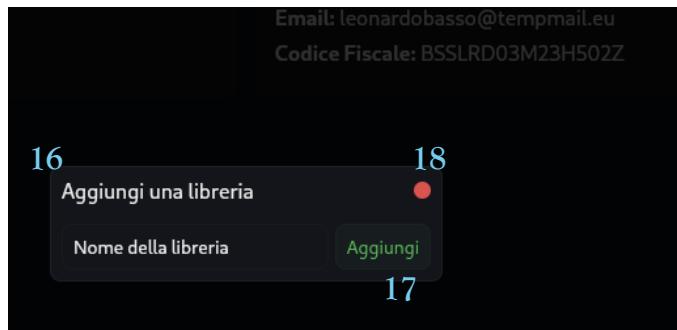
Schermata Utente:



12. Elenco delle librerie create
13. Informazioni inserite dall'utente
14. Il simbolo verde è per aggiungere nuove librerie
15. Tasto per uscire del profilo utente

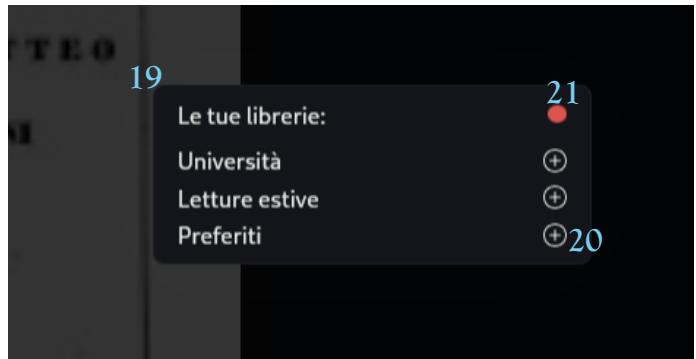
Pop-Up Libreria:

16. Finestra ove è possibile creare e scrivere il nome della propria libreria
17. Tasto per confermare la creazione della libreria
18. Tasto di uscita



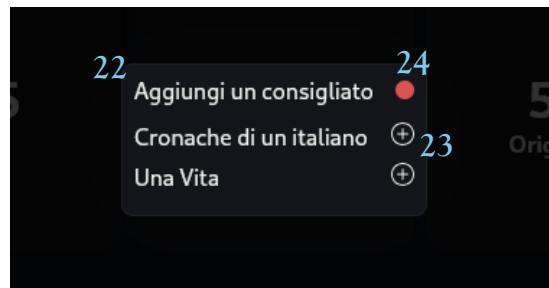
Pop-Up per aggiungere un libro alla libreria:

19. Finestra ove si può scegliere in che libreria si vuole salvare il libro
20. Tasto per aggiungere il libro nella libreria desiderata
21. Tasto di uscita

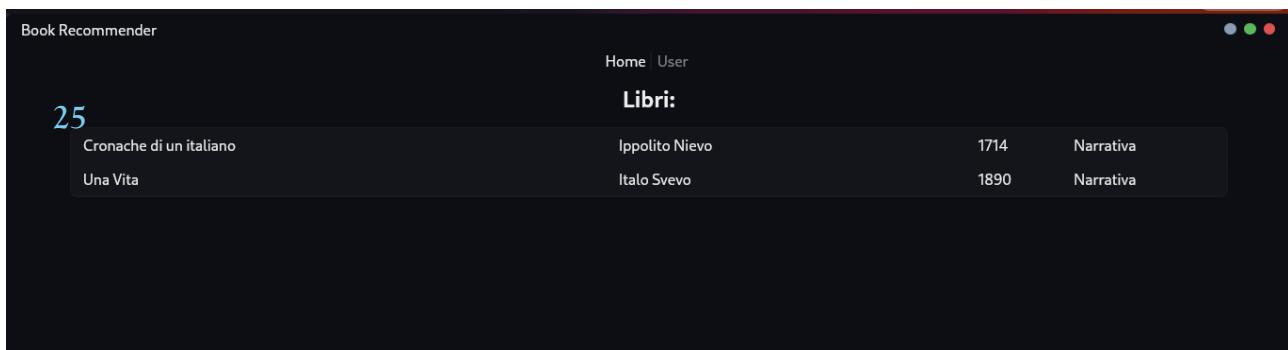


Pop-Up per scegliere un libro da consigliare:

22. Finestra ove si può scegliere quale libro consigliare al pubblico
23. Tasto per aggiungere il libro alla lista dei libri consigliati
24. Tasto di uscita

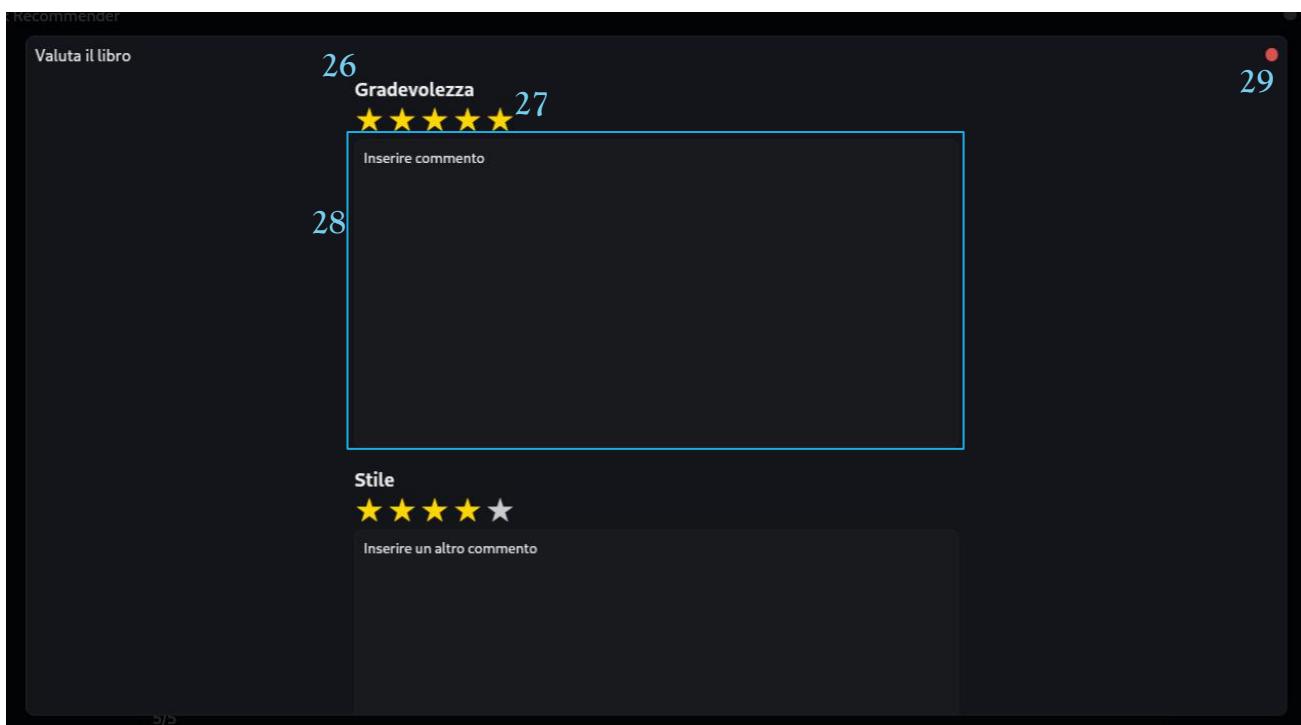


Schermata della propria libreria:



25. Lista di libri salvati nella propria libreria scelta

Pop-Up Valutazione:



26. Categoria di valutazione
27. Stelline selezionabili per dare la valutazione desiderata
28. Spazio destinato alla stesura di un commento
29. Tasto per uscire dal pop-up dopo aver dato una valutazione

Schermata valutazione, commenti e libri consigliati:



30. Vengono mostrati i libri consigliati da altri utenti
31. Vengono mostrati i libri che l'utente registrato ha consigliato
32. Zona destinata ai commenti e alle valutazioni degli utenti registrati
33. Tasto per salvare il libro selezionato in una libreria a scelta di quelle create

6. Limiti della soluzione sviluppata

- ❖ Non è previsto un sistema di recupero password
 - ❖ Nessuna gestione avanzata dei ruoli (es. admin, moderatore).
 - ❖ Backup manuale
 - ❖ Le stelline hanno qualche problema, ma funzionano per la valutazione.
-

7. Sitografia / Bibliografia

❖ *POSTGRESQL* – <https://www.postgresql.org/>

PostgreSQL è un sistema di gestione di database relazionali ad oggetti open-source, noto per la sua **affidabilità, scalabilità e conformità agli standard SQL**.

È spesso chiamato “Postgres” ed è utilizzato da aziende, università e sviluppatori in tutto il mondo per gestire dati strutturati in modo sicuro ed efficiente.

Caratteristiche principali

- **Supporto ACID:** garantisce integrità delle transazioni (Atomicità, Consistenza, Isolamento, Durabilità)
- **Estensibilità:** puoi creare tipi di dato, funzioni e operatori personalizzati
- **Gestione avanzata delle query:** ottimizzatore intelligente, supporto a viste, trigger, stored procedure
- **Supporto JSON e dati geospaziali:** ideale anche per applicazioni moderne e flessibili
- **Multi-Version Concurrency Control (MVCC):** consente accessi concorrenti senza blocchi pesanti

Perché usarlo?

- È **gratuito e open-source**, con una comunità attiva e documentazione completa
- Funziona su **Windows, Linux, macOS** e altri sistemi
- È adatto sia a piccoli progetti che a sistemi enterprise con milioni di record

Vuoi approfondire?

Puoi esplorare tutto sul sito ufficiale di <https://www.postgresql.org/download/>, oppure leggere una panoramica dettagliata su <https://it.wikipedia.org/wiki/PostgreSQL>.

❖ **APACHE MAVEN** – <https://maven.apache.org/>

Apache Maven è uno strumento di **automazione del build** per progetti software, principalmente Java, sviluppato dalla Apache Software Foundation.

Serve a gestire in modo centralizzato e automatizzato tutte le fasi del ciclo di vita di un progetto: dalla compilazione al test, dal packaging alla distribuzione.

Cosa fa Maven?

- **Compila** il codice sorgente Java
- **Esegue test** automatici (unitari e d'integrazione)
- **Crea pacchetti** (JAR, WAR, ecc.)
- **Gestisce le dipendenze**: scarica automaticamente le librerie necessarie
- **Genera documentazione** (es. JavaDoc)
- **Distribuisce** il progetto su repository remoti o locali

Il cuore di Maven: il file pom.xml

Questo file XML descrive:

- Le coordinate del progetto (groupId, artifactId, version)
- Le dipendenze esterne (librerie da scaricare)
- I plugin da usare (per test, compilazione, ecc.)
- Le fasi del ciclo di vita (compile, test, package, install, deploy)

Perché è utile?

- Evita conflitti tra versioni di librerie
- Standardizza la struttura del progetto
- Automatizza operazioni ripetitive
- Facilita il lavoro in team

Puoi approfondire tutto questo sulla pagina ufficiale di Apache Maven o leggere una guida dettagliata in italiano su <https://www.nextre.it/cose-maven-si-usa/>.

❖ **TAURI DOCUMENTATION** – <https://tauri.app/>

Tauri è un framework open-source per creare applicazioni desktop **piccole, veloci e sicure** usando tecnologie web (HTML, CSS, JavaScript) per il frontend e Rust per il backend.

È pensato per chi vuole costruire app multiplataforma (Windows, macOS, Linux, Android, iOS) partendo da una singola base di codice.

Documentazione ufficiale

Puoi esplorare la documentazione completa su:

- Sito ufficiale Tauri 2.0 – guida per iniziare, configurare e distribuire la tua app
- Configurazione Tauri – dettagli su tauri.conf.json, Tauri.toml e configurazioni specifiche per ogni sistema operativo
- Guide Tauri v1 – se stai usando una versione precedente
- GitHub Tauri Docs – repository con tutta la documentazione tecnica

Caratteristiche principali

- **Frontend indipendente**: puoi usare qualsiasi framework (Vue, React, Svelte, ecc.)
- **Backend in Rust**: compilato in binari ultra-leggeri (anche < 1MB)
- **Sicurezza avanzata**: sandboxing, CSP, isolamento del codice
- **Configurazione flessibile**: JSON, TOML, JSON5
- **Plugin modulari**: per notifiche, tray icon, filesystem, ecc.

❖ **VUE.JS OFFICIAL GUIDE** – <https://vuejs.org/>

Vue.js è un framework JavaScript progressivo per la creazione di interfacce utente moderne e reattive.

La sua **guida ufficiale** è una risorsa completa che ti accompagna passo dopo passo, sia che tu sia un principiante sia uno sviluppatore esperto.

Dove trovarla

- **Vue.js Official Guide** – Introduzione: panoramica del framework, concetti chiave e primi esempi: <https://vuejs.org/guide/introduction.html>
- **Vue.js in italiano**: traduzione ufficiale della documentazione per chi preferisce leggere in italiano: <https://it.vuejs.org/guide/introduction.html>
- **Vue.js Homepage**: accesso diretto a tutorial, API, esempi e community: <https://vuejs.org/>

Cosa contiene

- **Introduzione al framework**: cos'è Vue, come funziona la reattività, come si scrivono componenti
- **Guida passo-passo**: installazione, configurazione, struttura del progetto
- **API Reference**: tutte le funzioni e metodi disponibili
- **Tutorial interattivi**: esempi pratici da provare direttamente nel browser
- **Stili di programmazione**: Options API vs Composition API, con esempi per entrambi

❖ **JAVALIN** – <https://javalin.io/>

Javalin è un framework web **leggero e moderno** per lo sviluppo di applicazioni in **Java e Kotlin**, pensato per essere semplice, veloce e altamente interoperabile.

È costruito sopra il server Jetty e permette di creare API REST, WebSocket e interfacce asincrone con pochissime righe di codice.

Caratteristiche principali

- **Semplicità estrema:** niente annotazioni, niente classi da estendere, solo codice diretto.
- **Supporto a Java e Kotlin:** puoi scrivere la tua app in entrambi i linguaggi senza differenze strutturali.
- **WebSocket e HTTP/2:** supporta comunicazioni in tempo reale e protocolli moderni.
- **Asincrono e performante:** può gestire richieste non bloccanti e scalare facilmente.
- **Compatibile con OpenAPI:** puoi documentare le tue API con Swagger UI o ReDoc

Vuoi approfondire?

Puoi esplorare la documentazione ufficiale su <https://javalin.io> oppure leggere una guida introduttiva su <https://blogs.oracle.com/javamagazine/post/javalin-a-simple-modern-web-server-framework> o <https://www.baeldung.com/javalin-rest-microservices>.