

Basi di Dati Mod. 2 - Progetto A.A. 2021/2022

Stefano Calzavara

April 15, 2022

1 Introduzione

L'obiettivo del progetto è lo sviluppo di una web application che si interfaccia con un database relazionale. Il progetto deve essere sviluppato in Python, utilizzando le librerie Flask e SQLAlchemy. La scelta del DBMS da utilizzare è invece libera e lasciata ai singoli gruppi. Siete invitati a leggere interamente questo documento con attenzione ed a chiarire col docente eventuali punti oscuri prima dello sviluppo del progetto.

2 Temi per il Progetto

Il progetto può essere svolto liberamente su uno dei seguenti due temi. La scelta del tema non avrà nessuno impatto sulla valutazione finale del progetto: entrambi i temi possono garantire il punteggio massimo all'esame. I temi sono deliberatamente presentati ad alto livello e si prestano allo sviluppo di progetti più o meno complicati, a seconda dei gusti, della fantasia e dell'abilità dei componenti del gruppo. Considerate questi temi sostanzialmente come semplici spunti da cui partire: potete utilizzare la vostra esperienza con applicazioni simili per identificare un insieme di funzionalità interessanti da implementare.

2.1 Gestione delle Attività di Orientamento

Vi viene chiesto di curare il design e l'implementazione di una web application per la gestione delle attività di orientamento (PCTO) del DAIS. L'applicazione deve permettere la creazione di nuovi corsi, ciascuno composto da una o più lezioni tematiche, da svolgersi in presenza oppure online. Gli studenti devono potersi iscriversi ai corsi, mentre i docenti devono avere accesso ad un'interfaccia di analitica relativa ai corsi, che riporti almeno il numero di iscrizioni per ciascun corso e la demografica degli studenti partecipanti. I docenti devono avere inoltre la possibilità di inserire nuovi corsi.

Vengono forniti alcuni spunti possibili per arricchire il progetto, senza pretesa di esaustività:

- Le attività in presenza hanno vincoli fisici, per esempio relativi alla capienza delle aule. Inserire un limite al numero di iscrizioni per ciascun corso, permettendo ai docenti di configurare appropriate politiche di controllo (es. richiedere che ciascuno studente possa iscriversi al massimo ad un corso).
- Alla fine di un corso gli studenti potrebbero desiderare un attestato per il riconoscimento dell'attività. Associare un token segreto a ciascuna lezione del corso per permettere agli studenti di confermare la loro presenza tramite di esso ed inserire una funzionalità di richiesta attestato alla fine del corso, che riconosca l'attività se è stato sostenuto un numero minimo di lezioni.

2.2 Piattaforma di Streaming Audio

Vi viene chiesto di curare il design e l'implementazione di una piattaforma di streaming audio simile a Spotify. Chiaramente il progetto richiede di gestire solo i "metadati" associati alle canzoni, come l'artista, il titolo, l'album, l'anno di produzione, ecc. Gli utenti devono poter creare le proprie playlist, mentre gli

artisti devono avere accesso ad un'interfaccia di analitica relativa alle loro produzioni, che riporti le principali statistiche relative ai loro album ed alle relative canzoni. Gli artisti devono avere inoltre la possibilità di inserire nuovi album e canzoni.

Vengono forniti alcuni spunti possibili per arricchire il progetto, senza pretesa di esaustività:

- La piattaforma può implementare un semplice meccanismo di raccomandazione, che sia in grado di suggerire nuovi canzoni agli utenti sulla base dei loro gusti, per esempio sulla base delle loro playlist o delle canzoni già ascoltate.
- La piattaforma può implementare specifiche politiche di protezione del copyright, per esempio distinguendo fra utenti free e premium, oppure richiedendo una sottoscrizione speciale per accedere a contenuti riservati. E' possibile che certe canzoni siano disponibili solo fino ad una certa data di scadenza, che la piattaforma può decidere di rinnovare o meno in futuro.

3 Requisiti del Progetto

Il progetto richiede come minimo lo svolgimento dei seguenti punti:

1. Progettazione concettuale e logica dello schema della base di dati su cui si appoggerà all'applicazione, opportunamente commentata e documentata.
2. Creazione di un database, anche artificiale, tramite l'utilizzo di uno specifico DBMS. La creazione delle tabella e l'inserimento dei dati può essere effettuato anche con uno script esterno al progetto.
3. Implementazione di un front-end minimale basato su HTML e CSS. E' possibile utilizzare framework CSS esistenti come W3.CSS, Bootstrap o altri. E' inoltre possibile fare uso di JavaScript per migliorare l'esperienza utente, ma non è strettamente necessario e non influirà sulla valutazione finale.
4. Implementazione di un back-end basato su Flask e SQLAlchemy (o Flask-SQLAlchemy).

Per migliorare il progetto e la relativa valutazione è raccomandato gestire anche i seguenti aspetti:

1. Integrità dei dati: definizione di vincoli, trigger, transazioni per garantire l'integrità dei dati gestiti dall'applicazione.
2. Sicurezza: definizione di opportuni ruoli e politiche di autorizzazione, oltre che di ulteriori meccanismi atti a migliorare il livello di sicurezza dell'applicazione (es. difese contro XSS e SQL injection).
3. Performance: definizione di indici o viste materializzate sulla base delle query più frequenti previste.
4. Astrazione dal DBMS sottostante: uso di Expression Language o ORM per astrarre dal dialetto SQL.

E' possibile focalizzarsi solo su un sottoinsieme di questi aspetti, ma i progetti eccellenti cercheranno di coprirli tutti ad un qualche livello di dettaglio. E' meglio approfondire adeguatamente solo alcuni di questi aspetti piuttosto che coprirli tutti in modo insoddisfacente.

4 Documentazione

Il progetto deve essere corredato da una relazione in formato PDF opportunamente strutturata, che discuta nel dettaglio le principali scelte progettuali ed implementative. Una struttura plausibile per la relazione può essere la seguente:

1. Introduzione: descrizione ad alto livello dell'applicazione e struttura del documento.
2. Funzionalità principali: una descrizione delle principali funzionalità fornite dall'applicazione, che aiuti a comprendere come avete declinato lo spunto di partenza relativo al tema scelto per il progetto.

3. Progettazione concettuale e logica della basi di dati, opportunamente spiegate e motivate.
4. Query principali: una descrizione di una selezione delle query più interessanti che sono state implementate all'interno dell'applicazione, utilizzando una sintassi SQL opportuna.
5. Principali scelte progettuali: politiche di integrità e come sono state garantite in pratica (es. trigger), definizione di ruoli e politiche di autorizzazione, uso di indici, ecc. Tutte le principali scelte progettuali devono essere opportunamente commentate e motivate.
6. Ulteriori informazioni: scelte tecnologiche specifiche (es. librerie usate) e qualsiasi altra informazione sia necessaria per apprezzare il progetto.

Il codice del progetto deve essere inoltre opportunamente strutturato e commentato per favorirne la manutenzione e la leggibilità.

5 Consegna e Valutazione

Ciascun gruppo deve consegnare il progetto tramite un unico file ZIP caricato tramite Moodle nelle finestre dedicate, tipicamente in prossimità delle sessioni di esame. Il file ZIP deve contenere sia il progetto che la documentazione (in un unico file in formato PDF). Ogni componente del gruppo è responsabile di chiarire il proprio contributo al progetto durante la presentazione orale dello stesso, ma la valutazione del progetto sarà unica per l'intero gruppo a meno di casi eccezionali (es. copie). Durante la presentazione orale del progetto uno o più componenti del gruppo mostreranno all'opera l'applicazione sviluppata, facendo vedere le funzionalità principali e rispondendo alle domande del docente. E' sufficiente che l'applicazione funzioni localmente su una delle macchine del gruppo: non è richiesto che sia accessibile tramite Internet.

Il progetto verrà valutato rispetto ai seguenti parametri:

1. Documentazione: qualità e completezza della documentazione allegata.
2. Database: qualità della progettazione ed uso appropriato degli strumenti presentati nel corso.
3. Funzionalità: quantità e qualità delle funzionalità implementate dall'applicazione.
4. Codice: qualità complessiva del codice prodotto (robustezza, leggibilità, generalità, riuso...).

Si noti che eventuali progetti artificialmente complicati potrebbero essere penalizzati: implementare funzionalità complesse, ma non appropriatamente pensate o motivate, non è una buona strategia per migliorare la valutazione del proprio progetto.

Nota a questa edizione. I migliori gruppi che svilupperanno il primo fra i due temi suggeriti potranno essere contattati per portare in produzione una versione opportunamente rivista ed estesa del loro progetto. In particolare c'è un forte interesse del prof. Filippo Bergamasco nello sviluppo di un'applicazione di questo tipo, di conseguenza potrebbe essere possibile sviluppare ulteriormente il progetto come parte dell'attività di tesi triennale.