

Basi di Dati Mod. 2 - Progetto A.A. 2024/2025

Luca Cosmo

April 2, 2025

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 (di due o tre persone), anche se è consigliato l'utilizzo di Postgres. Siete invitati a leggere interamente questo documento con attenzione ed a chiarire col docente eventuali punti oscuri prima dello sviluppo del progetto.

2 Traccia: piattaforma per la gestione di viaggi aerei

Si vuole realizzare un'applicazione web per permettere la ricerca di voli delle principali compagnie e l'acquisto dei relativi biglietti. Il sistema deve gestire due tipologie di utenti: le compagnie aeree e i passeggeri. Le compagnie possono inserire delle tratte servite dai propri aerei. Ciascuna tratta collega uno specifico aeroporto (città) di partenza ed uno di destinazione da uno o più aerei in determinati periodi. Ciascun volo ha associato dei costi a seconda della tipologia di biglietto (economy, business, first class). Inoltre, le compagnie possono visualizzare statistiche sul numero di passeggeri dei loro voli, il guadagno totale e le tratte maggiormente richieste.

I passeggeri possono cercare i voli per viaggiare da una sorgente ad una destinazione. Il viaggio può comprendere una sosta intermedia purché offra un lasso di tempo sufficiente per effettuare il cambio (almeno 2 ore). L'applicazione deve permettere l'ordinamento delle alternative possibili in base a vari parametri (costo, tempo, numero di soste, etc.). Una volta selezionata un'alternativa, i passeggeri possono procedere all'acquisto dei biglietti selezionando il posto a sedere per ciascun volo ed eventuali extras (bagaglio aggiuntivo, posti più larghi, etc.)

Infine, il sistema deve fornire agli utenti l'informazione in tempo reale sulla disponibilità di posti di ciascun volo. Il numero di posti disponibili deve cioè aggiornarsi automaticamente appena un utente completa l'acquisto di un biglietto.

La ricerca di voli può essere effettuata da qualsiasi utente, anche anonimo. Tutte le altre operazioni richiedono invece autenticazione.

La descrizione del progetto è deliberatamente presentata ad alto livello e si presta allo sviluppo di progetti più o meno complicati, a seconda dei gusti, della fantasia e dell'abilità dei componenti del gruppo. Considerate la traccia come un semplice spunto da cui partire: potete utilizzare la vostra creatività e la vostra esperienza con applicazioni simili per identificare nuove funzionalità interessanti da implementare.

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 secondo la notazione introdotta nel Modulo 1 del corso.

2. Creazione di un database, anche artificiale, tramite l'utilizzo di uno specifico DBMS. La creazione delle tabelle 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. Potete riutilizzare il front-end sviluppato per il progetto del corso "Tecnologie e applicazioni web". 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 richiesto 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. La documentazione deve anche chiarire (in appendice) il contributo al progetto di ciascun componente del gruppo. Viene raccomandata la seguente struttura per la relazione:

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. La presentazione deve seguire la notazione grafica introdotta nel Modulo 1 del corso.
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.
7. Contributo al progetto (appendice): una spiegazione di come i diversi membri del gruppo hanno contribuito al design ed allo sviluppo.

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 all'interno di un unico file ZIP caricato tramite Moodle nelle finestre dedicate, tipicamente in prossimità delle sessioni di esame. Il file ZIP deve contenere:

- Il codice sorgente del progetto e le relative risorse (immagini, fogli di stile...). Non è richiesto un dump del database usato in fase di sviluppo e testing.
- La documentazione, in un unico file in formato PDF. Assicuratevi che la documentazione rispetti le indicazioni della sezione precedente.
- Un video della durata massima di 10 minuti in cui viene fatta una demo dell'applicazione. Il video deve mostrare uno screen capture che faccia vedere l'applicazione funzionante, fornendo una panoramica delle principali funzionalità implementate. Il video deve essere opportunamente commentato tramite una voce fuori campo.

Il progetto verrà valutato rispetto ai seguenti quattro parametri:

1. Documentazione: qualità, correttezza 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.