

Basi di Dati Mod. 2 - Progetto A.A. 2022/2023

Stefano Calzavara

April 20, 2023

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 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 creatività e la vostra esperienza con applicazioni simili per identificare nuove funzionalità interessanti da implementare.

2.1 Valutazione di Progetti di Ricerca

Viene richiesto di curare il design e l'implementazione di una web application per la valutazione interna dei progetti di ricerca da sottoporre per finanziamento all'Unione Europea. I ricercatori devono poter creare nuovi progetti da sottoporre a valutazione, ciascuno dei quali comprende una descrizione testuale ed uno o più documenti in formato PDF da valutare, possibilmente di tipo diverso (data management plan, ethics deliverable, ecc.). Ciascun progetto ha uno stato, fra cui:

- approvato: i valutatori hanno espresso parere favorevole e non sono più possibili modifiche;
- sottomesso per valutazione: sottoposto per la prossima finestra di valutazione, ma non ancora valutato;
- richiede modifiche: i valutatori hanno richiesto modifiche in vista di un'ulteriore valutazione;
- non approvato: i valutatori hanno espresso parere contrario e non sono più possibili modifiche.

I valutatori possono accedere ai diversi progetti da valutare nella prossima finestra di valutazione, scaricare i documenti da valutare e creare un report di valutazione per ognuno di essi, oltre ad aggiornare lo stato del progetto, per esempio ad "approvato". I ricercatori hanno accesso ai report di valutazione e possono rivedere i loro progetti alla luce degli stessi, finché non vengono approvati o rifiutati.

Vengono forniti alcuni spunti possibili per arricchire lo scenario, senza pretesa di esaustività:

- Ogni progetto ha uno storico di versione per tenere traccia delle diverse iterazioni del processo di valutazione: in particolare, per ogni documento del progetto, i ricercatori ed i valutatori devono poter accedere a tutte le versioni sottomesse precedentemente ed al relativo report di valutazione. Le versioni possono includere informazioni ausiliarie, che dettaglino i cambiamenti rispetto alle versioni precedenti.

- Ogni progetto include una componente di messaggistica, tramite la quale i ricercatori ed i valutatori possono interagire. Per esempio i ricercatori possono chiedere ulteriori chiarimenti riguardo ad una valutazione ricevuta, a cui i valutatori possono rispondere in maniera anonima.
- I valutatori possono inserire commenti puntuali direttamente all'interno dei PDF sottomessi per valutazione, per esempio in forma di note. In questo modo i report di valutazione possono fare riferimento anche a tali commenti puntuali (es. "si veda la nota a pagina 4").

2.2 Gestione di Esami Universitari

Viene richiesto di sviluppare un applicativo per la gestione di esami universitari. I **docenti** possono creare nuovi **esami**, ciascuno dei quali richiede il superamento di $n \geq 1$ prove. **Quando tutte le n prove sono state superate e sono ancora valide, è possibile registrare il voto.** Ogni **studente** può sostenere una delle prove ad uno degli appelli. Quando si supera una **prova**, il suo superamento ha una data che registra quando è stata sostenuta e una data che registra la sua scadenza. **Dopo la scadenza una prova non è più valida: rimane nel sistema, ma dal punto di vista della registrazione è come se non fosse stata superata.** Un'altra ragione di invalidazione di una prova è il sostenimento della stessa prova ad un appello successivo. Se l'appello più recente è superato, allora il voto precedente si invalida e quello nuovo ne prende il posto. Se invece l'appello più recente non è superato, la prova precedente viene invalidata e si registra la nuova come un fallimento.

Il sistema deve essere in grado di visualizzare lo stato di studenti (prove valide sostenute, storico delle prove sostenute...) ed **appelli** (studenti che hanno superato le diverse prove). Deve essere possibile ottenere **l'elenco degli studenti che sono in condizione di avere l'esame registrato** (ma non lo hanno ancora fatto) perchè tutte le n prove sono superate e valide, mostrando per ciascuno le caratteristiche delle prove che abilitano la registrazione dell'esame.

Vengono forniti alcuni spunti possibili per arricchire lo scenario, senza pretesa di esaustività:

- **Ogni prova ha una diversa tipologia** (scritto, orale, progetto...) **ed un diverso tipo di ricaduta sul superamento dell'esame** (idoneità, voto che contribuisce alla media pesata, bonus al voto...). Il docente deve poter configurare questo aspetto dell'esame, andando a definire i criteri per la valutazione finale.
- **E' possibile definire dei vincoli sulle prove d'esame**, richiedendo per esempio che la discussione del progetto debba essere effettuata solo dopo che l'esame scritto è stato superato. **Certe prove d'esame potrebbero essere opzionali**, cioè lo studente può scegliere se svolgerle o meno, o mutuamente esclusive con altre, **cioè lo studente deve scegliere una fra più opzioni disponibili.**
- **Le diverse prove d'esame possono essere gestite da docenti diversi**, come nel caso dei compiti. In questo caso un docente non può interferire con la valutazione data da un altro docente, però deve essere possibile avere visibilità dell'esame nel suo complesso. I compiti potrebbero essere soggetti a regole speciali (es. essere disponibili solo in un certo appello d'esame).

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. 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 indicativa 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.

Nota a questa edizione. I due gruppi che svilupperanno il miglior progetto per ciascun tema potranno essere contattati per portare in produzione una versione opportunamente rivista ed estesa del loro applicativo. Sarà possibile quindi sviluppare ulteriormente il progetto come parte dell'attività di tesi triennale, cercando di arrivare ad un prodotto finito da utilizzare all'interno dell'ateneo.