

Basi di Dati Mod. 2 - Progetto A.A. 2023/2024

Luca Cosmo

April 3, 2024

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 Piattaforma di E-commerce

Si richiede di curare il design e l'implementazione di un portale di e-commerce, dove gli utenti possono comprare e vendere i prodotti online. Il sistema include le seguenti funzionalità:

- Gestione degli utenti: Implementare funzionalità di autenticazione e autorizzazione degli utenti. Gli utenti dovrebbero poter registrarsi, accedere e gestire i propri profili. Inoltre, dovrebbero esserci ruoli utente differenti come acquirenti e venditori, ognuno con il proprio insieme di permessi.
- Gestione dei prodotti: Creare un database per memorizzare informazioni sui prodotti, inclusi nome, descrizione, categoria, prezzo, disponibilità, ecc. I venditori dovrebbero poter aggiungere, modificare ed eliminare i propri prodotti.
- Ricerca e Filtri: Implementare una funzionalità di ricerca che permetta agli utenti di cercare prodotti basati su parole chiave, categorie o altri attributi. Inoltre, fornire opzioni di filtro per affinare i risultati della ricerca basati su intervallo di prezzo, marca, ecc.
- Carrello della spesa: Implementare una funzionalità di carrello della spesa che permetta agli utenti di aggiungere prodotti al proprio carrello, aggiornare le quantità e procedere al pagamento. Il sistema dovrebbe gestire i livelli di inventario e aggiornare la disponibilità dei prodotti di conseguenza.
- Gestione degli ordini: Sviluppare un sistema per gestire gli ordini effettuati dagli utenti. Gli utenti dovrebbero poter visualizzare la loro cronologia degli ordini, monitorare lo stato dei propri ordini e ricevere notifiche sugli aggiornamenti degli ordini. I venditori dovrebbero anche avere accesso ai dettagli degli ordini per i prodotti che hanno venduto e poterne aggiornare lo stato.

- Recensioni e Valutazioni (Opzionale se meno di tre persone): Consentire agli utenti di lasciare recensioni e valutazioni per i prodotti che hanno acquistato. Visualizzare le valutazioni medie e fornire opzioni di ordinamento basate sulle valutazioni per aiutare gli utenti a prendere decisioni informate.

2.2 Piattaforma di Social Networking con Funzionalità Pubblicitarie

Progettare e implementare una piattaforma di social networking con funzionalità pubblicitarie integrate, consentendo alle aziende di promuovere i propri prodotti e servizi a un pubblico mirato. La piattaforma dovrebbe soddisfare sia gli utenti regolari che gli inserzionisti, con il sistema che tiene traccia degli annunci sui quali ogni utente clicca. Ecco le principali caratteristiche:

- Gestione utenti: Implementare funzionalità di autenticazione e autorizzazione degli utenti regolari e degli inserzionisti. Gli utenti dovrebbero poter registrarsi, accedere e gestire i propri profili. Gli inserzionisti dovrebbero avere funzionalità aggiuntive per gestire le proprie campagne pubblicitarie.
- Gestione delle amicizie: Consentire agli utenti di connettersi tra loro, inviare richieste di amicizia e gestire la propria lista di amici. Gli utenti regolari dovrebbero poter interagire con i post (compresi i commenti) e i contenuti dei propri amici all'interno della piattaforma.
- Condivisione dei Contenuti: Consentire agli utenti di creare e condividere (ad amici o al pubblico) vari tipi di contenuti come testo, foto e video. Gli utenti dovrebbero poter mettere "mi piace" e commentare i contenuti postati dai propri amici e da altri utenti nella rete.
- Gestione degli annunci: Fornire agli inserzionisti la possibilità di creare e gestire campagne pubblicitarie all'interno della piattaforma. Gli inserzionisti dovrebbero poter definire criteri di targeting (in base alle caratteristiche dell'utente), durata e monitorare le performance dei loro annunci.
- Gestione avanzata annunci (Opzionale per gruppi inferiori a tre): implementare la possibilità di impostare un budget per una particolare campagna (ogni click ricevuto costa). Impostare anche una priorità agli annunci, annunci con priorità più elevata avranno più probabilità di venire visualizzati. Opzionalmente, implementare un semplice sistema di raccomandazione delle inserzioni (i.e. per definire quale inserzione mostrare in base al comportamento dell'utente e alle sue amicizie).

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 artificiosamente 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.