INTRODUZIONE

L'applicazione che stiamo descrivendo è una piattaforma di social media che consente agli utenti di condividere post, seguire amici, e interagire con annunci pubblicitari mirati. Questa piattaforma è progettata per supportare due ruoli principali: gli utenti normali e gli inserzionisti pubblicitari. Gli utenti possono creare profili, postare contenuti, seguire amici e interagire con i post attraverso commenti e like. Gli inserzionisti possono creare annunci mirati per un pubblico specifico basato su sesso, età e interessi.

Il documento principale, ovvero app.py, è strutturato come segue:

* Descrizione Generale delle Tabelle: Una panoramica ad alto livello delle tabelle principali utilizzate nell'applicazione.
* Definizione delle Classi: Una descrizione dettagliata delle classi utilizzate per mappare le tabelle nel database.
* Funzioni Utili: Funzioni di utilità per la gestione delle password, email, e altre operazioni comuni.
* Rotte dell'Applicazione: Le rotte principali per il login, logout, registrazione, gestione degli interessi e home page.

Vista ad albero di tutta la cartella del progetto:

│ app.py // file py con database, route e funzioni necessarie. Con questo file si fa partire il tutto

├───contenuti // contiene tutte le immagini ed è qui che le foto e i video dei post vengono salvati

├───database // database e le insert in sql

├───templates // tutti i file html necessari

├───venv

└───\_\_pycache\_\_

Abbiamo scelto di inserire tutto nello stesso file app.py così da non avere problemi con i vari import e far funzionare nel modo più semplice possibile il database e il sito.

PROGETTAZIONE CONCETTUALE E LOGICA DELLA BASE DI DATI

Il database progettato rappresenta un sistema complesso di gestione utenti, contenuti, annunci pubblicitari e interazioni sociali. L'obiettivo principale del design è fornire una piattaforma robusta e scalabile che supporti un ampio spettro di funzionalità sociali, dalla pubblicazione di post alla gestione degli annunci pubblicitari, includendo anche le interazioni tra utenti come commenti, like e messaggi.

Users

La tabella Users contiene le informazioni principali degli utenti registrati sulla piattaforma. Include dati come username, nome, cognome, email, genere, età, ruolo e una biografia opzionale. L'uso di enumerazioni per i campi sesso e ruolo garantisce coerenza e facilita la gestione dei permessi e delle preferenze.

Interessi

La tabella Interessi elenca gli interessi disponibili sulla piattaforma, come sport, musica, arte, ecc. Questo permette agli utenti di selezionare i loro interessi personali e facilitare la personalizzazione dei contenuti.

UserInteressi

La tabella UserInteressi è una tabella di associazione molti-a-molti che collega gli utenti con i loro interessi. Questo consente di registrare e gestire i vari interessi degli utenti in modo efficiente.

Amici

La tabella Amici traccia le relazioni di amicizia tra gli utenti, indicando chi segue chi e il momento in cui l'amicizia è stata instaurata. Questo permette di gestire le reti sociali degli utenti.

Post

La tabella Post gestisce i contenuti pubblicati dagli utenti, come immagini, video e testi. Include informazioni sull'utente che ha pubblicato il post, il tipo di contenuto, il testo e il timestamp di creazione.

PostComments

La tabella PostComments contiene i commenti fatti sui post dagli utenti. Ogni commento è associato a un post specifico e a un utente, e include il contenuto del commento e il timestamp di creazione.

PostLikes

La tabella PostLikes tiene traccia dei like sui post. Registra quale utente ha messo like a quale post e quando è avvenuta l'azione. Questo permette di misurare l'engagement degli utenti con i contenuti.

Annunci

La tabella Annunci gestisce gli annunci pubblicitari. Include informazioni sull'inserzionista, il tipo di annuncio, il target di genere ed età, l'interesse target, e le date di inizio e fine dell'annuncio. Questo consente una gestione dettagliata delle campagne pubblicitarie.

AnnunciClicks

La tabella AnnunciClicks traccia i click sugli annunci, registrando quale utente ha cliccato su quale annuncio e quando è avvenuto il click. Questo è utile per analizzare l'efficacia degli annunci.

AnnunciComments

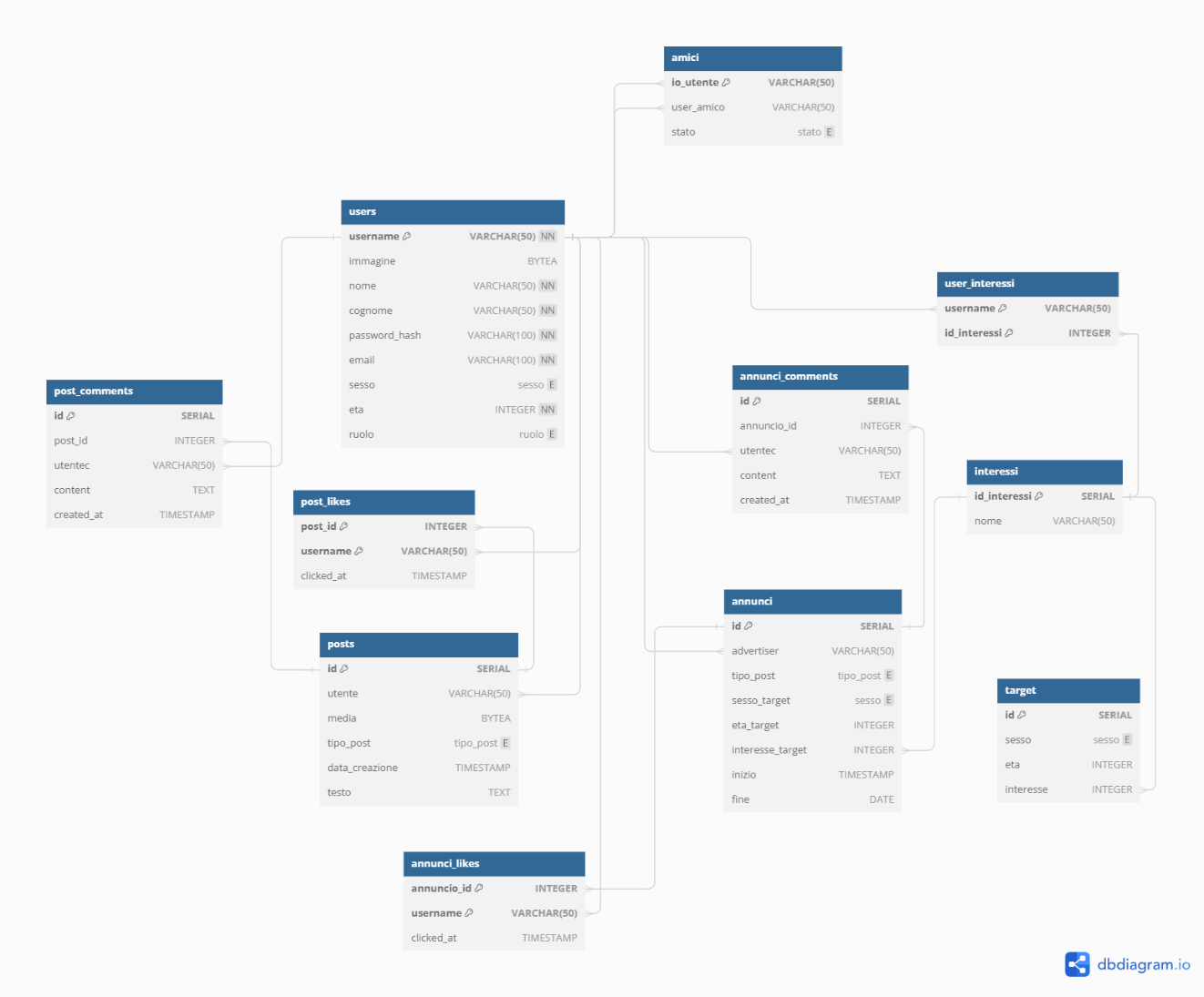
La tabella AnnunciComments contiene i commenti fatti sugli annunci. Ogni commento è associato a un annuncio specifico e a un utente, e include il contenuto del commento e il timestamp di creazione.

AnnunciLikes

La tabella AnnunciLikes tiene traccia dei like sugli annunci. Registra quale utente ha messo like a quale annuncio e quando è avvenuta l'azione, permettendo di misurare l'engagement con i contenuti pubblicitari.

Messaggi

La tabella Messaggi gestisce i messaggi privati scambiati tra gli utenti. Include il testo del messaggio, l'ID del mittente e del destinatario, il timestamp di creazione e un riferimento opzionale a un post inviato con il messaggio. Questo facilita la comunicazione diretta tra gli utenti della piattaforma.



PRINCIPALI SCELTE PROGETTUALI:

Politiche di integrità

Le politiche di integrità sono cruciali per garantire che i dati siano corretti, coerenti e affidabili. Nel progetto, sono state adottate diverse misure per mantenere l'integrità dei dati:

Integrità referenziale: Viene garantita utilizzando chiavi esterne tra le tabelle. Ad esempio, ogni post, commento, e like fa riferimento a un utente e un post specifico, assicurando che non ci siano riferimenti orfani.

Trigger e vincoli: L'uso di trigger e vincoli a livello di database aiuta a mantenere la coerenza. Ad esempio, un commento non può essere aggiunto se non esiste il post associato, e un like non può essere registrato se l'utente ha già messo mi piace allo stesso post.

Gestione delle eccezioni: In caso di errori di integrità (ad esempio, tentativo di duplicazione di chiavi uniche), vengono catturate le eccezioni e il database viene riportato in uno stato coerente.

Definizione di ruoli e politiche di autorizzazione

La gestione dei ruoli e delle autorizzazioni è fondamentale per garantire che solo gli utenti autorizzati possano accedere a determinate funzionalità:

Autenticazione e autorizzazione: Utilizzo di @login\_required per garantire che solo gli utenti autenticati possano accedere a determinate route. Gli utenti non autenticati vengono reindirizzati alla pagina di login.

Ruoli utente: Distinzione tra utenti normali e pubblicitari (is\_pubblicitario). Questo permette di differenziare le funzionalità disponibili per ciascun tipo di utente, come la pubblicazione di annunci.

Controlli di accesso: Verifica che l'utente corrente abbia i permessi necessari per eseguire certe azioni, come l'eliminazione di post e commenti. Ad esempio, solo l'autore di un post o commento può eliminarlo.

Uso di indici

L'uso di indici è stato adottato per migliorare le performance delle query:

Indici sulle chiavi esterne: Gli indici sono stati creati sulle colonne che contengono chiavi esterne, come utente\_id nei post, commenti e likes. Questo accelera le query che filtrano i dati basandosi su queste colonne.

Indici per le ricerche: Per migliorare le performance delle ricerche sugli utenti (username), è stato aggiunto un indice sulla colonna username.

Scelte specifiche per le funzionalità

Pubblicazione post

Gestione file multimediali: I file multimediali (immagini e video) vengono salvati in una directory dedicata (contenuti). Prima di salvare il file, il sistema verifica l'esistenza della directory e la crea se necessario.

Salvataggio sicuro dei file: Viene utilizzato un percorso di salvataggio ben definito per evitare conflitti e garantire un accesso sicuro ai file multimediali.

Notifiche

Raccolta notifiche: Le notifiche vengono raccolte da diverse tabelle (likes, commenti, richieste di amicizia) e combinate in una singola lista ordinata per data. Questo permette agli utenti di vedere tutte le notifiche in ordine cronologico.

Funzione time\_since: Per calcolare e visualizzare il tempo trascorso dalla notifica, viene utilizzata una funzione che converte il timestamp in un formato leggibile, considerando il fuso orario italiano.

Chat tra amici

Organizzazione delle conversazioni: I messaggi vengono organizzati per conversazione, permettendo agli utenti di visualizzare le chat con i loro amici in maniera ordinata.

Invio messaggi e post: Gli utenti possono inviare sia messaggi di testo che condividere post direttamente nella chat, fornendo una maggiore flessibilità nelle comunicazioni.

Annunci

Targeting degli annunci: Gli annunci possono essere targettizzati in base al sesso, età e interessi degli utenti. Questo permette agli inserzionisti di raggiungere il pubblico desiderato in modo più efficace.

Gestione della durata: Gli annunci hanno una durata specificata dall'inserzionista. Dopo la scadenza, l'annuncio non viene più mostrato agli utenti, garantendo la rilevanza dei contenuti visualizzati.

Contributo al progetto:

Sia Marta Cazzin che Asia Carraro hanno messo mano in tutte le parti del progetto, ecco una suddivisione del lavoro piú dettagliata:  
Marta Cazzin 40 commit

Asia Carraro

Back End:

* Asia Carraro
* Marta Cazzin

Front End:

* Asia Carraro
* Marta Cazzin

Realizzazione dell’idea:

* Asia Carraro
* Marta Cazzin

Creazione del document:

* Marta Cazzin