

# Progetto: “fotogram”

## Premessa

Questo documento descrive il progetto d'esame di laboratorio dell'insegnamento di [Basi di Dati e Web](#). Si consiglia di verificare periodicamente sul [sito dell'insegnamento](#), in particolare prima della consegna, la presenza di eventuali aggiornamenti al documento.

Nel seguito viene descritto in maniera informale il dominio applicativo di interesse. Come nei casi reali, la specifica può essere incompleta e presentare contraddizioni o ambiguità. È compito dello studente individuare i punti critici, analizzarli e proporre soluzioni progettuali congruenti con l'intero sviluppo del progetto. Il documento presenta sia aspetti riguardanti la base di dati, sia le operazioni che dovranno essere implementate ed esposte mediante API REST per permettere a un ipotetico front-end di interagire con la base di dati. Lo studente ha il compito di comprendere quali parti sono da considerare nella modellazione della base di dati, e quali nella definizione delle operazioni e delle API REST.

## Documentazione Descrittiva

Si vuole realizzare una base di dati e le API REST per interagire con essa, per il social network “fotogram” basato sulla condivisione di immagini.

### Registrazione e Login

Prima di usare il servizio, un utente si deve registrare, fornendo una mail, uno username e una password. Non è possibile usare una mail o uno username già utilizzati. In una fase successiva, l'utente può anche inserire/modificare un'immagine di profilo (di massimo 100kb). Una volta registrato, l'utente può fare login, verificando che lo username e la password siano corretti e iniziando così una sessione. L'utente deve essere collegato per usare le altre funzionalità del sistema. Un utente collegato può fare logout, terminando in tal modo la sessione. Un amministratore ha la facoltà di rendere gli altri utenti moderatori e ha i poteri di un moderatore. Di un moderatore si conosce la data in cui è stato nominato moderatore.

### Follow

Un utente può seguire altri utenti, in modo analogo ad altri social network. Questa relazione è asimmetrica (es: come in twitter), quindi un utente può seguire un'altro utente senza essere a sua volta seguito. Per trovare gli utenti da seguire è possibile fare una ricerca per nome, che restituisce una lista di utenti il cui nome contiene la stringa cercata. È possibile per un utente visionare una lista di utenti seguiti e la lista di coloro che lo seguono e accedere ai loro profili utente.

### Post

Un utente collegato può pubblicare post contenenti un'immagine (di max 100kb) oppure un messaggio testuale. L'utente può vedere nella propria bacheca, che è la schermata principale del sistema, i post pubblicati dall'utente stesso e da tutti gli utenti che l'utente segue, ordinati per momento di creazione.

Soltanto un numero limitato di post è visibile alla volta; l'utente dovrà andare alle pagine successive per vedere i post più vecchi. Un utente può mettere un like a un post (o toglierlo se già stato dato). Per ogni post è mostrato: il nome e l'immagine di profilo del creatore del post, l'immagine o il testo del post, e il numero dei like che il post ha ricevuto.

## Moderazione

Un utente può flaggare un post come inappropriato. I post flaggati da un utente sono visibili come tali all'utente stesso ma non agli altri. L'utente può togliere il flag a un post che ha flaggato (es: se lo ha fatto per sbaglio). Un moderatore può accedere a una lista con tutti i post flaggati, ordinati per numero di flag ricevuti, e può moderare i post che reputa effettivamente inappropriati. I post moderati sono visibili ai soli moderatori. Un utente che ha 3 post moderati nell'arco di 30 giorni non può creare nuovi post.

## Profili utente

Selezionando un utente dalla lista degli utenti cercati, dalla lista di quelli che si segue, dalla lista di quelli da cui si è seguiti, oppure selezionando il nome dell'autore di un post, vengono mostrati i dati dell'utente (nome, mail e immagine profilo), quanti utenti segue e da quanti è seguito, e i suoi post. Dal profilo deve essere possibile eliminare l'utente da quelli seguiti o aggiungerlo se non è ancora seguito.

## Svolgimento del progetto

Il progetto dovrà essere svolto individualmente. Dovrà includere necessariamente i seguenti passi che dovranno essere adeguatamente documentati. La documentazione dovrà essere presentata in un file di testo chiamato **documentazione.txt** al quale saranno da aggiungersi altri file specificati sotto.

### 1. Progettazione e implementazione della base di dati

1. Definire lo schema E-R per la base di dati con eventuali vincoli di cardinalità e di identificazione. Riportare l'immagine dello schema (**ER.png**) e il file **ER.er** prodotto da <https://designer.polito.it/>. Motivare le scelte effettuate ed eventuali vincoli (in linguaggio naturale) nella documentazione.
2. Effettuare la traduzione dello schema E-R in uno schema E-R ristrutturato equivalente. Riportare l'immagine dello schema ristrutturato (**ER-ristrutturato.png**) e, nella documentazione, eventuali altri vincoli o modifiche rispetto allo schema E-R originale, motivando le scelte effettuate.
3. Effettuare la traduzione dello schema E-R ristrutturato in un equivalente schema relazionale. Riportare nella documentazione lo schema relazionale ed eventuali ottimizzazioni dello schema. Indicare gli indici, i valori unici, chiavi esterne, e i valori null per ogni relazione specificata.

### 2. Progettazione delle API REST e delle query SQL

1. In base alla richiesta progettuale, definire e riportare nella documentazione le API REST. Indicare per ogni chiamata il tipo, i parametri, i vincoli d'accesso e lo schema JSON della risposta (se c'è).
2. Definire e includere nella documentazione le query SQL necessarie per implementare le API REST specificate al punto precedente. Descrivere in linguaggio naturale gli aspetti implementativi che non possono essere gestiti attraverso query ma che saranno implementati in Javascript.

3. Fornire un file **creazione.txt** con gli script per la creazione delle strutture della base di dati in accordo allo schema relazionale ottenuto alla fine della fase di progettazione. Il file dovrà anche contenere script per popolare la base di dati creata con tutte le informazioni che si ritengono necessarie per una simulazione realistica, per verificare che i vincoli di dominio espressi siano corretti e che le operazioni di cui si richiede l'implementazione funzionino correttamente.

### 3. Realizzazione delle API REST con Node.JS

1. Scrivere un server Node.JS che implementi gli endpoint delle API (usando la libreria *express*), effettuando query su DB Postgres (con la libreria *pg*), gestendo le risposte, eventuali vincoli non gestibili mediante query, casi d'errore e mancanza di autorizzazione. Lo script di ingresso dovrà essere chiamato **index.js** e dovrà essere presente un **package.json** con le dipendenze necessarie. Il codice potrà essere ragionevolmente organizzato in diversi file .js nella root del package.
2. Le autorizzazioni d'accesso dovranno essere gestite utilizzando JSON Web tokens (JWT), generati durante la procedura di login (con la libreria *jsonwebtoken*), passati come header per ogni API che necessita di autorizzazione, e invalidati con la procedura di logout.
3. Implementare l'interfaccia Swagger (con librerie *swagger-ui-express* e *swagger-autogen*) per testare le API prodotte.

## Consegna del progetto

Il progetto dovrà essere caricato su [Upload](#) in "Basi di dati e web (ComDig) - Lab" nella sessione d'esame che si intende sostenere. La consegna è da effettuare da un mese fino a una settimana prima dell'appello ed è considerata come iscrizione all'appello. La consegna dovrà essere un file zip contenente:

1. Tutti i file di documentazione specificati sopra nella root della cartella
2. Tutti i file del server Node.Js nella root della cartella. **NON** includere la cartella **node\_modules**
3. Eventuali file d'esempio usati dall'applicativo (es: immagini dei post) nella cartella **resources**

## Esame di Laboratorio

All'esame sarà richiesto di progettare, implementare e documentare una nuova funzione a partire dal progetto consegnato. Durante la prova, lo studente dovrà scaricare il progetto consegnato sulla macchina d'esame e installarlo, effettuare le modifiche richieste, e consegnare il progetto modificato. La consegna dovrà avere la stessa struttura descritta sopra e includere tutta la documentazione aggiornata.

Se il materiale consegnato è incompleto o non consente l'installazione dell'applicativo, la prova sarà considerata non superata. Si consiglia vivamente di effettuare delle prove sulle macchine di laboratorio.