

Progetto di Basi di Dati e Laboratorio Web

Studente: Lorenzo Simoni

Matricola: 549507

IL PROGETTO

Il progetto consiste nella realizzazione di una **piattaforma** che permetta agli utenti di interagire con dei blog. Agli utenti viene concessa la possibilità di registrare un account per commentare e mettere mi piace a dei post. Se l'utente decide di verificare il proprio account avrà anche la possibilità di creare a sua volta un numero limitato di blog (al massimo 6). Pertanto, la piattaforma si configura come un gestore di blog (come Wordpress) che permette di pubblicare nei blog **testi e foto**. Ogni blog è associato ad una specifica **categoria** e ad una **sottocategoria**, che identificano molteplici aree di interesse (attualità, tecnologia, etc.), di conseguenza il sito si rivolge ad un **pubblico generalista**.

REQUISITI FUNZIONALI DEL PROGETTO

- **Registrazione, login e disconnessione**

La registrazione prevede l'inserimento da parte dell'utente di una serie di dati: nome utente, nome, cognome, email, password, numero di telefono e documento. I campi nome utente, email, numero di telefono e documento sono univoci. Se la registrazione va a buon fine, all'utente verrà anzitutto assegnato un id univoco, che verrà poi registrato nel database. Nello stesso momento, verranno inizializzate una serie di variabili di sessione come l'id dell'utente, il nome utente, lo status di verifica dell'account (inizializzato a 0) e il numero di blog posseduti (inizializzato a 0). Dopodiché l'utente sarà reindirizzato alla pagina *index.php?p=1*, ovvero la prima pagina della homepage. Viene così creato un **utente registrato**. In qualsiasi momento, l'utente potrà disconnettersi dal proprio account, chiudendo la sessione. Per accedere al sito l'utente dovrà inserire nella pagina *login.php* il proprio nome utente e la propria password. Esistono 3 tipi di utente:

- **Utente Ospite:** ovvero il visitatore, può visualizzare i post i blog e i commenti;
- **Utente Registrato:** stessi permessi dell'utente Ospite ma in aggiunta può commentare e mettere like ai post;
- **Utente Verificato:** stessi permessi dell'utente Registrato ma in aggiunta può creare/cancellare blog, essere coautore di un blog, aggiungere coautori ai suoi blog e scrivere/cancellare post.

- **Verifica, personalizzazione e cancellazione account**

L'utente registrato non è tuttavia verificato. Per verificare il proprio un utente deve visitare la pagina *profilo.php* e deve modificare il proprio **avatar** e aggiungere una **descrizione** al suo profilo. In questa pagina, infatti, l'utente avrà la possibilità di personalizzare alcune elementi del suo account come il **nome e il cognome**, l'avatar e per l'appunto la descrizione. Viene anche visualizzata la data di iscrizione. Inoltre, in questa pagina l'utente potrà eventualmente scegliere di eliminare il proprio account. Per farlo sarà necessario cliccare sul tasto rosso "*cancella il profilo*" che lo reindirizzerà alla pagina *cancellaUtente.php*. In questa pagina, per evitare che l'account venga eliminato accidentalmente, l'utente dovrà confermare la propria password, ultimando così il processo di cancellazione. La cancellazione dell'account utente, sia esso registrato o verificato, prevede l'eliminazione totale di tutti i suoi contenuti, tra cui tutti i suoi blog, tutti i suoi post, tutti i like e tutti i commenti. Inoltre, per evitare di appesantire il server con file inutili, il processo di cancellazione prevede anche di eliminare l'avatar, contenuto nella cartella e le immagini dei post.

- **Creazione di un blog e visualizzazione della lista dei blog**

Solo gli utenti verificati possono creare un blog, per un massimo di 6 blog per utente. Per creare un blog l'utente potrà cliccare sul pulsante "*Crea un blog*" presente sulla navbar, oppure recarsi direttamente sulla pagina *listaBlog.php*, dove sarà possibile visualizzare i blog creati ed amministrati dall'utente. Nel caso in cui esso non abbia creato alcun blog, allora l'utente verrà direttamente reindirizzato sulla pagina *creazioneBlog.php* dove sarà necessario specificare un titolo, una descrizione, una categoria, una sottocategoria ed un tema grafico per poter procedere alla creazione. Una volta che l'utente creerà il blog verrà reindirizzato nuovamente sulla pagina *listaBlog.php*. Se l'utente raggiunge il limite di 6 blog creati, allora non potrà più accedere alla pagina di creazione dei blog e il pulsante sulla navbar e all'interno della pagina della lista dei Blog spariranno.

- **Aggiunta di una categoria e di una sottocategoria al blog**

Come detto in precedenza, durante la creazione del blog è necessario specificare una categoria ed una sottocategoria, che verranno assegnate al blog. L'utente dovrà scegliere una categoria attraverso un menù a tendina. La sottocategoria dipenderà

ovviamente dalla categoria principale scelta. Nella pagina di visualizzazione della lista dei blog *listaBlog.php* sarà possibile accedere alla pagina *gestisciBlog.php?id=<id del blog>* dove sarà possibile aggiornare la categoria e la rispettiva sottocategoria.

- **Personalizzazione e cancellazione dei blog**

L'utente potrà scegliere di modificare alcuni elementi dei propri blog. Gli basterà cliccare sul pulsante di personalizzazione in *listaBlog.php*, relativo allo specifico blog, per essere reindirizzato sulla pagina di personalizzazione del blog, cioè *personalizzaBlog.php?id=<id del blog>*. Qui è possibile modificare il titolo del blog, la descrizione e il tema grafico. L'utente può anche cancellare il blog cliccando sul pulsante cancella, eliminando così in maniera totale tutti i post, e le rispettive immagini, tutti i commenti e tutti i commenti ai post.

- **Aggiunta e rimozione di un coautore**

Nella pagina *gestisciBlog.php?id=<id del blog>* sarà possibile aggiungere e rimuovere uno o più coautori al blog. Solo il creatore del blog potrà aggiungere un coautore. Prima che un utente possa essere aggiunto come coautore di un blog si verifica che l'utente esista, che sia verificato, che non sia già un coautore e che non sia il creatore stesso. Gli stessi controlli vengono applicati alla rimozione del coautore. Un coautore ha solo la possibilità di scrivere un nuovo post, non può personalizzare, gestire o cancellare il blog di cui è coautore.

- **Creazione di un post su un blog**

L'utente potrà creare un post accedendo alla lista dei blog e cliccando sul pulsante "Scrivi". In alternativa è possibile creare un post direttamente dal blog, attraverso il pulsante "Scrivi". In entrambi i casi il pulsante reindirizzerà l'utente sulla pagina *creazionePost.php?id=<id del blog>* dove sarà necessario specificare il titolo del post, una immagine di copertina, il testo del post e una seconda immagine facoltativa, che apparirà alla fine del contenuto.

- **Visualizzazione di un blog ed eliminazione dei post**

I post, oltre ad apparire in ordine di pubblicazione sull'homepage *index.php?p=1*, appaiono sul blog dove sono stati creati. Nella pagina del blog sono presenti i post

fatti dal creatore e da eventuali coautori. La grafica di questa pagina varia in base al tema grafico scelto dall'utente in fase di creazione o in fase di personalizzazione. Graficamente, in questa pagina, viene mostrato il titolo, la descrizione del blog, il pulsante “*Scrivi*” e la lista dei post. Il creatore, o un coautore, potrà anche eliminare i post direttamente dalla pagina del blog, nella quale i post sono contenuti, cliccando sulla X a destra del titolo.

- **Visualizzazione del post**

La pagina dove sono contenute le informazioni di un post del blog, *post.php*, manterrà il tema grafico del blog su cui il post è pubblicato. Inoltre, a livello grafico, vengono mostrate la copertina, il titolo, l'autore, la data di creazione e il blog di provenienza, il testo del post ed un'eventuale immagine secondaria.

- **Aggiunta e rimozione di commenti e like**

Sempre nella pagina di visualizzazione del post sarà presente un box per aggiungere i commenti degli utenti registrati e verificati. Un commento può essere cancellato dall'utente che lo ha scritto e pubblicato, dal creatore del blog o dal coautore del blog che ha scritto il post. Inoltre sarà presente anche un pulsante like a forma di cuore che può essere utilizzato da utenti registrati ed utenti verificati per specificare il post è piaciuto, utile come indicatore per verificare il tasso di gradimento del post.

- **Visualizzazione blog per categoria**

Sulla navbar del sito è presente un menù a tendina contenente le categorie principali dei blog. Cliccando su un elemento del menù a tendina verranno visualizzati tutti i blog che appartengono a quella specifica categoria.

- **Ricerca di un blog per titolo del blog, per autore, per categoria e sottocategoria**

Sempre sulla navbar del sito è presente una barra di ricerca da cui l'utente potrà cercare i blog specificando il titolo, il nome dell'autore, oppure la categoria, o la sottocategoria, assegnata a cui appartiene il blog. Quando l'utente invierà la ricerca verrà reindirizzato alla pagina *cerca.php?q= <stringa di ricerca>*.

- **Ricerca di un post per titolo**

Sempre nella pagina *cerca.php* verranno visualizzati i post che avranno nel titolo la

stringa specificata nella ricerca dell'utente.

- **Visualizzazione della homepage**

Nella homepage *index.php* vengono visualizzati gli ultimi 5 post creati, con la possibilità di visualizzare anche post meno recenti grazie ai link delle pagine presenti in fondo alla homepage (sistema simile al motore di ricerca Google).

LA STRUTTURA DEL DATABASE

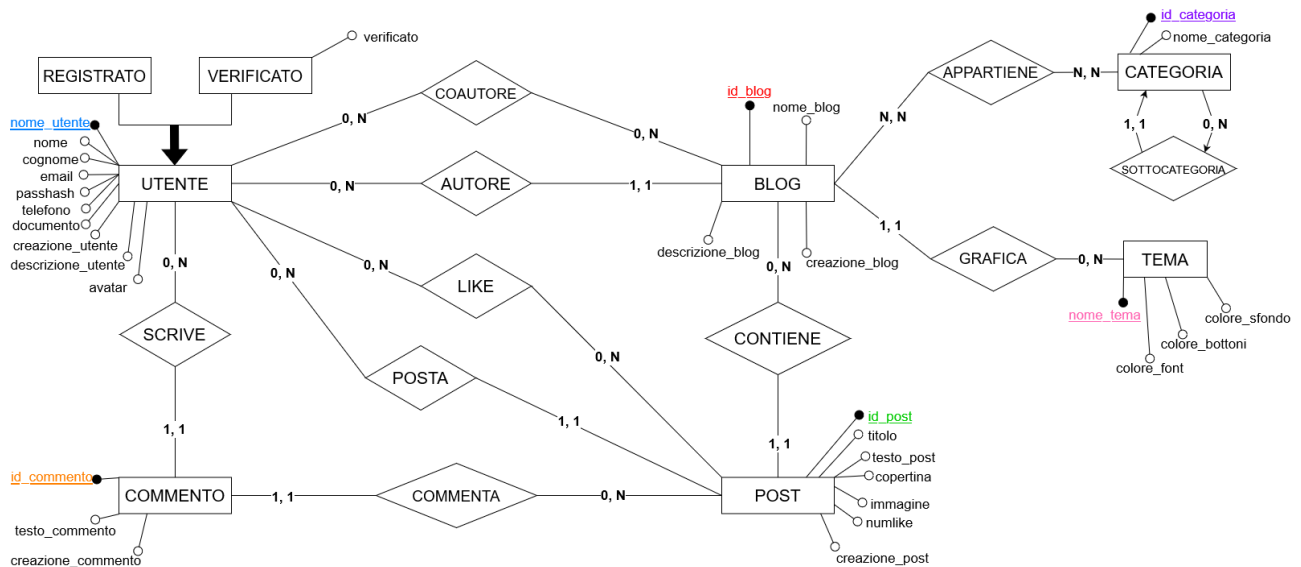


Figura 1: **Diagramma Entità-Relazione**

OPERAZIONI SUI DATI

- **Operazioni sull'utente**
 - Login
 - Logout
 - Registrazione
 - Modifica profilo utente
 - Verifica account utente
 - Eliminazione dell'account utente
- **Operazioni blog**
 - Creazione blog
 - Eliminazione blog
 - Visualizzazione blog
 - Modifica blog
 - Assegnazione coautore blog

- Rimozione coautore blog

- **Operazione post**

- Creazione post
- Cancellazione post
- Visualizzazione post
- Votazione post
- Eliminazione votazione post
- Commento post
- Eliminazione commento
- Assegnazione categoria
- Rimozione categoria blog

- **Operazioni di ricerca**

- Per titolo del post
- Per titolo del blog
- Per nome dell'autore
- Per categoria o sottocategoria del blog

Entità		
Entità	Descrizione	Attributi
Utente	Utente registrato alla piattaforma	<u>id_utente</u> , nome_utente, nome, cognome, email, passhash, telefono, documento, creazione_utente, avatar, descrizione_utente, verificato
Blog	Blog creato dall'utente verificato	<u>id_blog</u> , nome_blog, descrizione_blog, creazione_blog,
Tema	Tema grafico per i blog	<u>nome_tema</u> , colore_sfondo, colore_font, colore_bottoni
Post	Post creato dall'utente verificato	<u>id_post</u> , titolo, testo_post, creazione_post, copertina, immagine, numlike
Categoria	Categoria del blog	<u>id_categoria</u> , nome_categoria
Commento	Commento lasciato da un utente sul blog	<u>id_commento</u> , testo_commento, creazione_commento

Relazioni		
Relazioni	Descrizione	Entità coinvolte
Autore	Ogni utente può essere autore di più blog. Un blog può avere un solo autore.	Utente (0, N) – Blog (1, 1)
Coautore	Ogni utente può essere coautore di più blog. Un blog può avere più coautori.	Utente (0, N) – Blog (0, N)
Like	Ogni utente può mettere like a più post. Un post può ricevere like da più utenti.	Utente (0, N) – Post (0, N)
Posta	Ogni utente può postare più post su un blog. Un post può essere associato ad un solo utente.	Utente (0, N) – Post (1, 1)

Scrive	Un utente può scrivere più commenti. Un commento può essere associato ad un solo utente.	Utente (0, N) – Commento (1, 1)
Commenta	Un commento può essere associato ad un solo post. Un post può avere più commenti.	Commento (1, 1) – Post (0, N)
Contiene	Un blog può avere più post. Un post può essere pubblicato in un solo blog.	Blog (0, N) – Post (1, 1)
Appartiene	Più blog possono essere associati ad una categoria. Più categorie possono essere associate a più blog.	Blog (N, N) – Categoria (N, N)
Sottocategoria	Si tratta di una relazione riflessiva sulla stessa tabella Categoria. Una categoria può avere più sottocategorie. Una sottocategoria appartiene ad una sola categoria principale.	Categoria (0, N) – Sottocategoria (1, 1)
Grafica	Un blog deve per forza associato ad un tema. Un tema può essere associato a 0 o più blog.	Blog (1, 1) – Tema (0, N)

SCHEMA LOGICO

Lo schema logico del database SQL è formato da 7 tabelle: Utente, Blog, Tema, Coautore, Post, Like, Categoria, Appartiene e Commento. Gli attributi sottolineati sono le chiavi, i colori indicano a quale attributo fanno riferimento le foreign key.

Utente (id_utente, nome_utente, nome, cognome, email, passhash, telefono, documento, creazione_utente, avatar, descrizione_utente, verificato)

La tabella Utente contiene le informazioni relative agli utenti registrati nel sistema, come l'ID dell'utente, il nome utente, nome, cognome, email, password criptata, telefono, documento, data di creazione del profilo, avatar, descrizione dell'utente e lo stato di verifica del profilo. L'utente diventa verificato inserendo un avatar e una descrizione nel suo profilo.

Blog (id_blog, nome_blog, descrizione_blog, creazione_blog, autore_blog, tema)

La tabella Blog contiene le informazioni sui blog creati dagli utenti, come l'ID del blog, il nome del blog, la descrizione del blog, la data di creazione, l'autore del blog e il tema associato al blog.

Tema (nome_tema, colore_sfondo, colore_font, colore_bottoni)

La tabella Tema contiene le informazioni sui temi disponibili per i blog, come il nome del tema, il colore dello sfondo, il colore del font e il colore dei bottoni.

Coautore (id_utente, id_blog)

La tabella Coautore contiene le informazioni sui coautori dei blog, come l'ID utente e l'ID del blog. I coautori possono essere aggiunti solo dall'autore del blog e possono solo modificare il blog ma non cancellarlo.

Post (id_post, titolo, testo_post, creazione_post, copertina, immagine, numlike, blog, autore_post)

La tabella Post contiene le informazioni sui post pubblicati dai vari autori sui blog, come l'ID del post, il titolo del post, il testo del post, la data di creazione del post, le immagini associate al post, il numero di like, l'ID del blog di appartenenza e l'autore del post.

Like ([id_utente](#), [id_post](#))

La tabella **Like** contiene le informazioni sui voti assegnati dagli utenti ai vari post, come l'ID utente e l'ID del post.

Categoria ([id_categoria](#), nome_categoria, parent_id)

La tabella **Categoria** contiene le informazioni sulle categorie utilizzate per organizzare i post dei vari blog, come l'ID della categoria, il nome della categoria e l'ID della categoria genitore.

Appartiene ([id_blog](#), [id_categoria](#))

La tabella **Appartiene** è una tabella di associazione tra i blog e le categorie, contenente l'ID del blog e l'ID della categoria.

Commento ([id_commento](#), testo_commento, creazione_commento, [post](#), [autore_commento](#))

La tabella **Commento** contiene le informazioni sui commenti lasciati dagli utenti ai vari post, come l'ID del commento, il testo del commento, la data di creazione del commento, l'ID del post a cui si riferisce e l'autore del commento.

BUSINESS RULES

1. Se la registrazione va a buon fine, i dati dei visitatori sono registrati nel database, ed il visitatore diventa automaticamente un Utente Registrato, che ottiene diversi privilegi.
2. Un visitatore per registrarsi deve inserire dei dati nel form di input che siano validi e che non costituiscano un pericolo per il sito.
3. Un Utente Registrato può visualizzare dei Blog, dei Post, lasciare un commento oppure un like ad un Post.
4. Un utente Registrato non può creare Blog e di conseguenza non può creare Post.
5. Un utente Registrato diventa un Utente Verificato solo dopo aver aggiornato il suo avatar e la sua descrizione del suo profilo.
6. Un utente Verificato può creare al massimo 6 blog, essere coautore di più Blog e

creare dei Post.

7. Un Utente Verificato nell'esatto momento in cui crea un Blog, ne diventa **Autore**. Nessun altro utente può revocargli questo privilegio.
8. Un Autore, o un Coautore, non può modificare un post che è stato creato, neanche se è creato da sé stesso.
9. Ogni Utente Registrato può dare una valutazione a qualsiasi post nei Blog, solo una volta. Il punteggio incrementerà perciò di 1 o decrementerà di 1.
10. Un Utente Verificato può concedere ad un altro Utente Verificato di essere coautore di uno dei suoi Blog. Il coautore può essere rimosso dall'Autore in qualsiasi momento tramite il bottone di gestione del Blog.
11. Un Coautore può solamente scrivere e cancellare dei post sul Blog di cui è stato nominato Coautore. Non può personalizzare, gestire e cancellare il Blog.
12. Un Autore può eliminare i commenti nei Post scritti da lui stesso o da un Coautore.
13. Ogni Utente Registrato, così come ogni Utente Verificato, può cancellare i suoi commenti sotto un Post e revocare un Like.
14. Ogni Utente Registrato, così come ogni Utente Verificato, può scegliere di eliminare il suo profilo in qualsiasi momento. La piattaforma si occupa di eliminare totalmente tutti i dati legati all'utente.

ANALISI DELLE RIDONDANZE

L'analisi della ridondanza è stata effettuata solo nel caso dell'attributo *numlike*, appartenente alla tabella *post*. Questo attributo indica il numero di like totali di un determinato post. Sono stati presi in considerazione due schemi per capire se mantenere la ridondanza *numlike* fosse più efficiente rispetto al non mantenerla. Si suppone che l'operazione sui dati del visualizzare un post sia un'operazione batch (B) e che avvenga con una frequenza di **1110 volte al giorno**. Si suppone anche che l'operazione sui dati di aggiungere un like ad un post avvenga con una frequenza di **325 volte al giorno**. La formula che utilizziamo per calcolare il peso computazionale è la seguente:

Peso = $f * w_p * (a * N_{Accessi_{Scrittura}} + N_{Accessi_{Lettura}})$, dove $w_b = 0,2$, $w_i = 0,5$ e $a = 1,5$

In questa due tabelle si analizza il peso computazionale dell'operazione di visualizzazione del numero di like totali di un post e dell'aggiunta del like ad un post, senza la ridondanza. Il risultato si calcola come segue:

Schema senza ridondanza

Visualizzazione like totali post			
Concetto	Costrutto	Accessi	Tipo accesso
Like	Relazione	707	L
Peso		156828	
Memoria		X	

$$\text{Peso} = 1110 * 0,2 * ((0 * 0) + 707) = 156828$$

Aggiunta like al post			
Concetto	Costrutto	Accessi	Tipo accesso
Like	Relazione	1	S
Peso		243	
Memoria		X	

$$\text{Peso} = 325 * 0,5 * ((1,5 * 1) + 0) = 243$$

Schema con ridondanza

Poiché per implementare la ridondanza è necessario un attributo int(8) da 4 byte, allora poiché si suppone di avere 972 post a livello di volumi, la memoria necessaria sarà 3768.

Visualizzazione like totali post (rid)			
Concetto	Costrutto	Accessi	Tipo accesso
Post	Entità	1	L
Peso		222	
Memoria	X	+	3768

$$\text{Peso} = 1110 * 0,2 * ((0 * 0) + 1) = 222$$

Creazione like post (rid)			
Concetto	Costrutto	Accessi	Tipo accesso
Vota	Relazione	1	S
Post	Entità	1	S
Peso		487	
Memoria	X	+	3768

Peso = $325 * 0,5 * ((1,5 * 2) + 0) = 487$

Peso computazionale schema = $156828 + 243 = 157071$

Memoria schema = X

Peso computazionale schema rid = $222 + 487 = 709$

Memoria schema rid = X + 3768

Come si può vedere, mantenere la ridondanza *numlike* all'interno della tabella *Like* riduce notevolmente il calcolo computazionale a cui il database sarebbe sottoposta, con un aumento di memoria irrisorio. Pertanto, la ridondanza è stata implementata nello schema ER.

SCELTE E TECNOLOGIE ADOTTATE NEL PROGETTO

La fase di sviluppo della piattaforma, la fase di debugging, e la fase di testing sono avvenute in locale, utilizzando l'applicazione XAMPP, gratuita e open-source, che fornisce un ambiente di sviluppo locale per la creazione e il testing di siti web dinamici. Come richiesto, il progetto si è strutturato tramite l'utilizzo di alcune tecnologie, le quali **MySQL** come DBMS, **PHP** come linguaggio server-side.

Ho deciso di utilizzare **PDO** nel mio progetto PHP per una serie di ragioni che hanno dimostrato di essere estremamente vantaggiose durante lo sviluppo. PDO ha semplificato notevolmente le operazioni di accesso e gestione dei dati, rendendo il codice più pulito. Un altro vantaggio chiave di PDO è la sua sicurezza integrata per prevenire attacchi di SQL injection. PDO utilizza i prepared statements e i parametri per separare i dati dalle istruzioni SQL, riducendo il rischio di vulnerabilità di sicurezza. Per prevenire attacchi malevoli con SQL è stata utilizzata anche la funzione trim() e molteplici controlli client-side e specialmente server-side, per blindare i form e sanificare gli input.

In alcune porzioni di codice la piattaforma sfrutta **JQUERY**, per la sua flessibilità, ed **AJAX**, in quanto è un linguaggio che sfrutta chiamate asincrone per inviare e ricevere dati dal server in background, senza dover ricaricare l'intera pagina web. Questo permette di migliorare l'esperienza dell'utente. In particolare AJAX viene utilizzato:

- Per la creazione del Blog (*creazioneBlog.php*) per l'aggiunta della categoria e della sottocategoria.

- Per la modifica del Blog (*gestisciBlog.php*), nella modifica della categoria e della sottocategoria del blog.
- Per l'inserimento del like e dei commenti ai Post (*post.php*).

Ho scelto di utilizzare **Bootstrap** nel mio progetto PHP perché mi ha fornito un solido framework front-end che ha semplificato notevolmente il processo di progettazione e implementazione dell'interfaccia utente. Quindi, ho potuto sfruttare i componenti predefiniti di Bootstrap per accelerare lo sviluppo e concentrarmi sulla parte interessante per l'esame, ovvero sulle funzionalità della piattaforma.