

Blog Art

Relazione Progetto Basi di Dati e laboratorio web

Docenti

Fabrizio Baiardi

Barbara Guidi

Studenti

Bresciani Chiara

562017

Papa Federica

597084

SPECIFICHE DEL SISTEMA

PHP VERSION: 7.1.26

ENGINE: MariaDB

1. STRUTTURA DEL SITO WEB

DESCRIZIONE SOMMARIA

Questo sito si propone di offrire agli utenti che lo visitano la possibilità di esplorare tutti i contenuti creati dagli utenti registrati in base ai loro interessi. Una volta che l'utente si registra, può infatti personalizzare il proprio profilo e creare dei blog con lo stile che preferisce ma anche accedere ad ulteriori layout con la versione premium. Inoltre, i possessori di un profilo hanno la possibilità di lasciare una valutazione e un commento ad un post, in modo da favorire l'interazione e lo scambio di idee tra i vari utenti e arricchire il sito con quello che pubblicano.

1.1 UTENTI

1.1.1 Tipologie utenti

Il sito Blog Art ha 4 tipologie di utenti: Visitatore, Utente registrato, Utente registrato premium e Admin.

- **Utente Visitatore (non registrato)**

Un utente non registrato può solo visitare il sito attraverso l'input *Cerca* nella barra di navigazione, con cui può cercare lo username di un utente oppure un blog o un post attraverso il suo titolo o sottotitolo. Sempre dalla barra di navigazione, tramite il bottone esplora, può visualizzare i vari blog in base alla categoria e sottocategoria in cui sono divisi.

- **Utente registrato**

Un utente registrato è un visitatore che può loggarsi tramite la pagina *login.php* o *signup.php* ed entrare nella sua area riservata, nella quale può creare dei blog in base ai suoi interessi, selezionando le categorie che

ritiene più opportune tra quelle proposte. Può inoltre creare dei post, commentarli e valutarli.

Un utente registrato può scegliere due tipologie di pacchetti: il pacchetto gratuito o pacchetto premium, che forniscono diversi permessi e possibilità di azione.

Con il **pacchetto gratuito** l'utente registrato può creare fino a 5 blog ed ha una scelta di layout limitata a 4.

- **Utente premium**

Questo utente, grazie al **pacchetto premium**, può scegliere fra 10 tipologie di layout diversi e creare blog senza limiti.

• **Come diventare utente premium**

Avvenuta la registrazione, ogni utente può passare alla versione premium attraverso il bottone presente nella pagina del suo profilo o attraverso il bottone presente nella pagina principale del sito (*index.php*), con cui verrà indirizzato alla pagina *premium.php*. In questa pagina viene presentato nel dettaglio il pacchetto premium e i servizi che esso offre.

In particolare, è possibile effettuare un pagamento selezionando il bottone Paypal e inserendo i dati dell'account nel form. Una volta premuto sul bottone *acquista* si avvia un pagamento PayPal che, in caso di avvenuto pagamento, manda l'utente alla pagina *success.php* e permette al suo account di passare al piano premium. In caso di errore invece, porta l'utente alla pagina *OnCancel.php*.

Per realizzare questo pagamento è stato utilizzato un tool disponibile su PayPal denominato *Sandbox test account*, che tramite il proprio account PayPal permette di creare nell'area developer account fittizi per testare la parte e-commerce di un sito web. Tutte le specifiche del servizio sono disponibili all'indirizzo: <https://developer.paypal.com/developer/applications>.¹

- **Utente Admin**

L'utente Admin possiede tutti i privilegi dell'utente premium, ma in aggiunta ha la possibilità di visualizzare una lista di tutti gli utenti tramite il bottone *gestisci utenti*, che solo lui può vedere nella sua area profilo. Questo lo porta alla pagina *admin.php*, dove viene mostrato un elenco di

¹ I dati dell'account PayPal sono disponibili nella sezione delle credenziali

tutti gli iscritti al sito. L'Admin può decidere in autonomia di eliminare un qualsiasi utente.

1.1.2 Gestione utenti

La gestione dei vari utenti è stata fatta tramite il campo *Livello* nella tabella *iscritto*. All'inizio della registrazione l'utente presenta un livello pari a 0, mentre nel caso in cui decida di passare al pacchetto premium il livello viene modificato e impostato ad 1.

L'utente Admin invece è tale quando ha un livello pari a 2.

Un utente non può passare al livello Admin, perché questo livello è stato pensato solo per i proprietari del sito.

1.2 FUNZIONALITÀ

1.2.1 Registrazione:

Per potersi registrare un utente deve inserire in un form posizionato nella pagina *signup.php* i seguenti dati:

- Nome Utente (nome univoco)
- Nome
- Cognome
- E-mail (univoca)
- Numero di telefono fisso o cellulare
- Estremi di un documento di identità
- Password

Una volta avvenuta la registrazione l'utente viene indirizzato in *profile/index.php*.

1.2.2 Login e Logout

Nel caso in cui l'utente sia già registrato, per entrare nella sua area personale può accedere attraverso il bottone *accedi*, collocato nella pagina principale *index.php*, oppure tramite il menu a tendina che compare nel momento in cui questo preme sull'immagine utente circolare nella navbar. Questi faranno avvenire l'accesso tramite la pagina *login.php*. In alternativa, l'accesso può anche verificarsi tramite la pagina *signup.php* se l'utente preme il link *Hai già un account?* Situato sotto il form di registrazione.

Una volta arrivato al form di login, per accedere al profilo l'utente deve inserire i seguenti dati:

- Username o indirizzo email
- Password

1.2.3 Contatta Admin

Nella pagina principale del profilo è presente una sezione con la quale è possibile inviare dei messaggi alla mail del sito Blog Art inserendo i propri dati (come nome, cognome e email) in modo da poter essere ricontattati.

Per realizzare questa sezione abbiamo inserito nel form un link esterno (<https://formspreet.io/f/mbjwjykb>) appartenente all'azienda che offre questo servizio di messaggistica. In questo modo, tramite POST, i dati scritti dall'utente vengono inviati direttamente all'azienda, che si occupa di reindirizzare quello che ha ricevuto alla mail di Blog Art (che abbiamo precedentemente registrato in questo sito aziendale). Una volta completata l'operazione di invio dei dati, per evitare di uscire da Blog Art, è stato utilizzato il taget "_blank" per far aprire una pagina affianco, dove viene comunicato all'utente che il messaggio è stato inviato.

Questa scelta di usare un sito esterno è stata fatta per evitare di appesantire il database.

1.3 GESTIONE PROFILO UTENTE

Nella pagina profilo, solo l'utente creatore del profilo può vedere determinati bottoni:

- Cambia immagine del profilo
- Modifica profilo
- Elimina profilo
- Crea un nuovo blog
- Passa a Premium (se il suo livello è 0)
- Gestisci utenti (se il suo livello è 2, quindi Admin)

Sempre nella pagina di profilo, l'utente può visualizzare la lista dei blog creati o la lista dei blog in cui collabora come coautore e premendo sul titolo del blog può accedere al blog stesso.

1.3.1 Cambia immagine profilo

Appena un utente si registra, gli viene assegnata un'immagine del profilo provvisoria, ma può inserirne una diversa, che sarà visibile sia nella pagina del profilo che nella barra di navigazione.

1.3.2 Modifica profilo

Tramite il bottone modifica profilo l'utente viene indirizzato alla pagina *profile/modifica-profilo.php*, dove ha la possibilità di modificare:

Username, nome, cognome, email, password, numero di telefono.

In questa sezione può anche aggiungere una biografia, il contatto FB e Instagram, che può poi modificare in qualsiasi momento.

1.4 CREAZIONE BOLD

Il bottone Crea Blog nella pagina del profilo utente permette di creare un blog tramite un form nel quale deve essere inserito:

- Titolo (univoco)
- Sottotitolo
- Descrizione del blog
- Layout

Una volta compilata questa parte, l'utente viene indirizzato ad un ulteriore form nel quale deve inserire gli argomenti di cui parlerà il blog, ma queste categorie non sono modificabili da parte degli utenti.

Dopo aver creato il blog l'utente può tornare al profilo e vedere nell'area "I miei Blog" il blog appena creato.

1.4.1 Layout

In base al layout scelto, il path del blog cambia.

Per poter permettere all'utente di usufruire di layout con struttura della pagina e stili diversi, abbiamo deciso di realizzare 10 "personalblog.php", i quali vengono gestiti tramite una variabile *idLayout* che indirizza al percorso corretto. In base al numero di layout scelto, la variabile assume il medesimo valore e va a completare il path *personalblog(idLayout).php*. Ad esempio, se si sceglie il layout 1, l'utente verrà indirizzato alla pagina *personalblog1.php*.

1.4.2 Gestione blog

Il proprietario di un blog può modificare il suo blog tramite i vari bottoni che si presentano nella pagina *personalblog(idLayout).php* . Questi bottoni permettono di:

Modificare: titolo, sottotitolo, descrizione, layout del blog

Inserire: post, coautore

Eliminare: blog, coautore

Le Sottocategorie scelte all'inizio della creazione del blog sono visibili nella pagina del blog e premendo su una di queste si viene riportati alla pagina *SearchSottocategoria.php*, che permette di visualizzare tutti i blog che parlano di quel determinato argomento.

Tramite il bottone vai al profilo dell'autore, situato in basso a sinistra, è possibile visitare il profilo del proprietario del blog.

Nella parte inferiore di *personalblog* possiamo trovare l'area dedicata alle anteprime dei post scritti dall'autore o dal coautore.

1.4.3 Permessi autore e coautore

Il coautore può modificare i dati del blog, creare post e commenti, ma a differenza dell'autore non può eliminare il blog, aggiungere un coautore, eliminare commenti di altri utenti o modificare i dati dei post creati dall'autore.

1.4.4 Visite blog

Tramite la funzione *getContatoreVisite* ogni qualvolta un utente visita il blog, viene incrementata di uno la colonna *Visite* nella tabella *blog*. Questo dato viene poi utilizzato per creare una lista dei blog più popolari, disponibile nella sezione blog più popolari della home del sito.

1.5 CREAZIONE POST

La creazione di un post avviene tramite un form, avviato da *Crea Post* (all'interno del bottone *Edit*) in *personalblog.php*. Questo bottone rinvia alla pagina *Post_create.php*, che permette di inserire nel post:

- Titolo
- Sottotitolo
- Anteprima
- Testo

1.5.1 Gestione Post

Una volta creato il post, lo si può visualizzare nella pagina *post.php*, dove è possibile:

- Caricare una o due immagini, che si possono poi modificare caricandone altre o eliminare del tutto
- Modificare titolo, sottotitolo e testo del post
- Valutare un post
- Inserire un commento

1.5.3 Valutazione Post

La valutazione si può effettuare scegliendo un punteggio da 1 a 5 tramite l'input di valutazione. Per confermare il voto bisogna premere su *invia valutazione*. Una volta inviato il voto, questo può essere modificato anche in seguito.

1.5.3.1 Gestione valutazioni

Tramite la funzione *AggiungiMedia* viene creata una media delle valutazioni che sono state eseguite su un post. Questo permette di creare una lista dei post ordinati in base al valore della loro media dei voti ricevuti, disponibile nella sezione post più votati della home.

1.6 CREAZIONE COMMENTI

Tramite l'input di commento un utente registrato può commentare un post.

1.6.1 Gestione commenti

I commenti possono essere modificati/cancellati dal loro creatore o dal creatore del blog.

I commenti vengono visualizzati in base al proprio ID, quindi a partire dal più recente, che si troverà in cima all'elenco. Ogni post riporta la data di inserimento, l'orario e l'icona con l'immagine del profilo e il nome utente del commentatore. Tramite icona o Username è possibile visitare il profilo dell'utente che ha rilasciato il commento. L'utente che scritto il commento può modificare e/o eliminare il commento.

2. SICUREZZA SITO

Per rendere il sito più sicuro sono stati utilizzati:

- Prepared statements per le query che interagiscono con il database e lo modificano, in modo da prevenire SQL Injection.
- Funzione *htmlspecialchars()* per gestire come una stringa di testo il codice html o xss che l'utente potrebbe inserire nei campi di testo.
- Controlli sui GET passati come parametro, in modo tale che si possa accedere ad un'area privata come quella del *modifica-profilo.php* solo se il parametro del GET passato come parametro è uguale al valore della sessione dell'utente connesso in quel momento.

3. CONNESSIONE

La connessione al database è stata fatta utilizzando PDO.

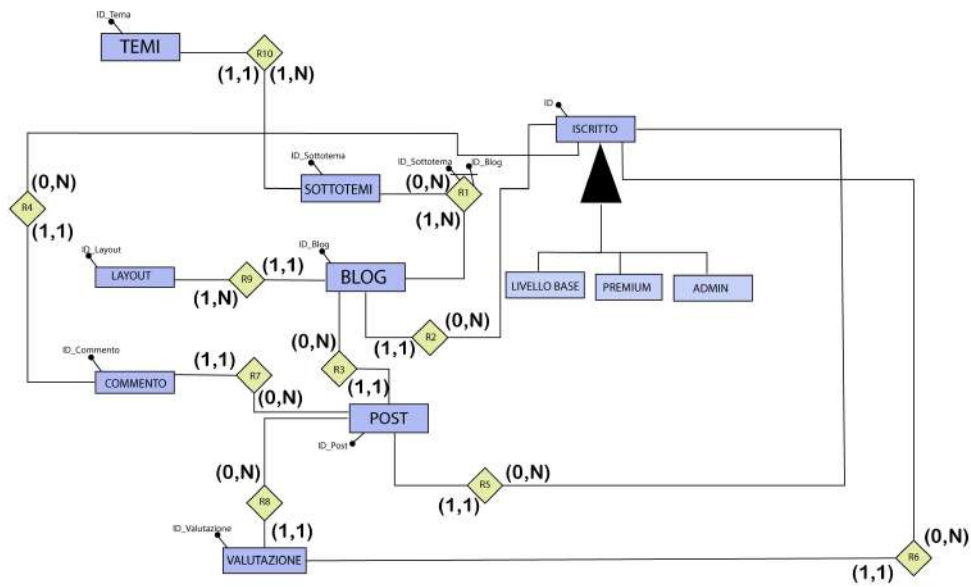
4. PERCORSI

Per rendere possibile la navigazione del sito sono stati utilizzati sia dei percorsi relativi che dei percorsi assoluti. In particolare, è stato necessario utilizzare il percorso assoluto per permettere che tutte le funzionalità della barra di navigazione siano disponibili in qualsiasi punto del sito in cui ci si trovi. Questo percorso è comunque modificabile grazie ad una variabile (**\$path**) che è stata inserita nel file *connessione/config.php*.

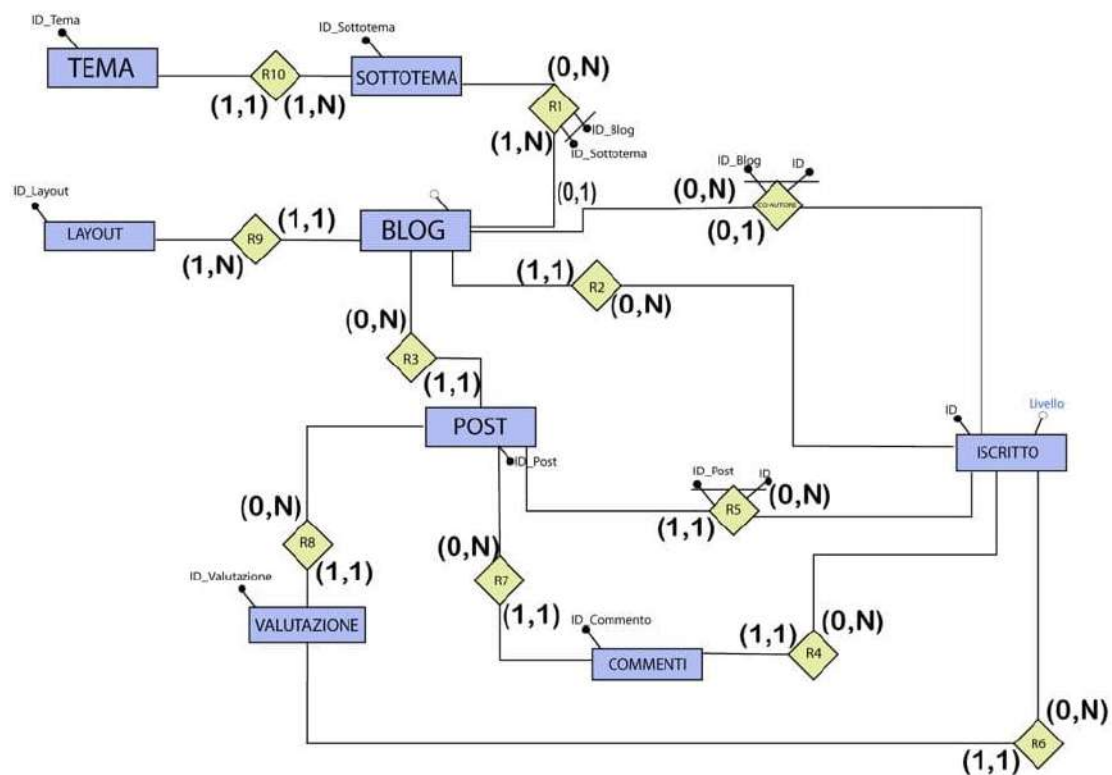
Quindi, nel momento in cui si decida di cambiare il nome della cartella che contiene il sito o di inserirla all'interno di altre cartelle di localhost, sarà necessario aggiornare questa variabile \$path in *connessione/config.php* e, in aggiunta, modificare l'indirizzo url nel file *search.js* e *premium.js* mettendo il percorso giusto al posto del nome della cartella attuale.

5. STRUTTURA DATABASE

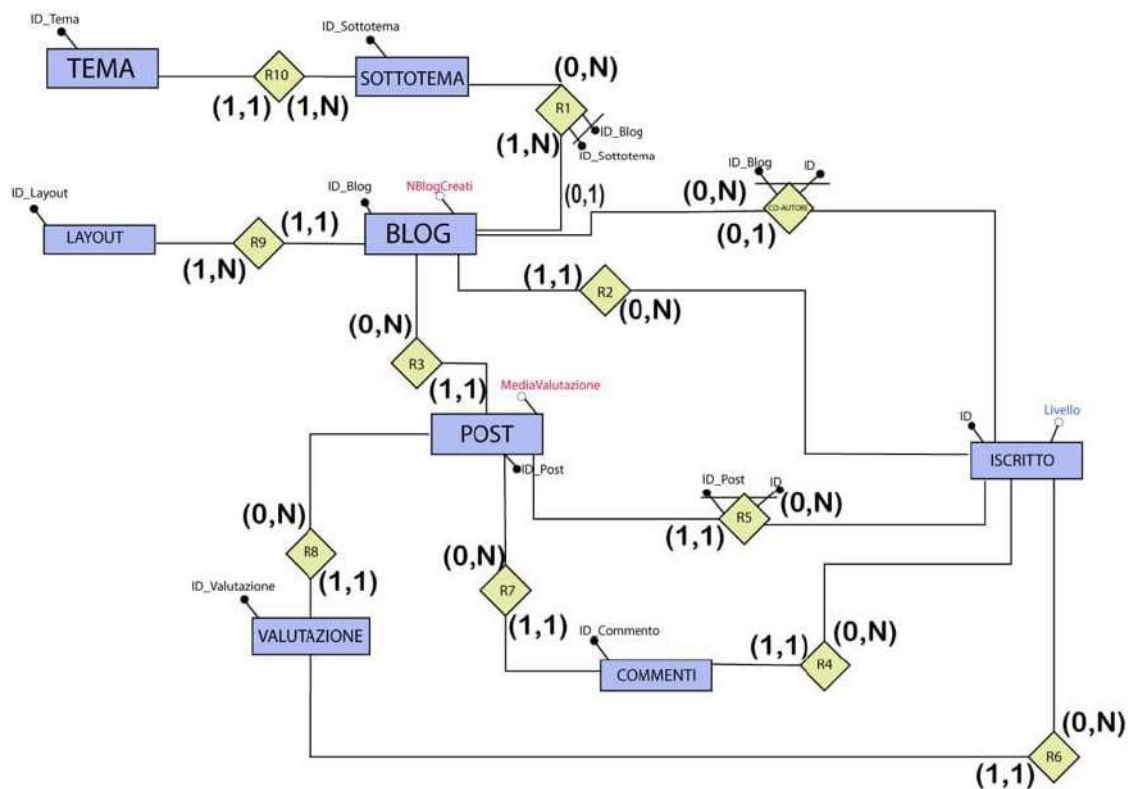
5.1 SCHEMA E-R di partenza



5.2 SCHEMA E-R senza ridondanze



5.3 SCHEMA E-R con ridondanze



5.4 DIZIONARIO ENTITA'

Entità	Descrizione	Attributi	Identificatore univoco
iscritto	Dati di un utente registrato	ID, ID_User, Nome, Cognome, Email, UPassword, Telefono, Avatar, N_doc, Livello, NBlogCreati, Bio, Facebook, instagram	ID
blog	Dati di un blog	ID_Blog, Autore, Titolo, Sottotitolo, Descrizione, Data_Creazione, Visite	ID_Blog
post	dati di un post	ID_Post, Titolo, Sottotitolo, Testo, Data_Pubblicazione, Anteprima, Img1, Img2	ID_Post
commento	dati di un commento	ID_Commento, Testo, Data_comm,	ID_Commento
valutazione	dati di una valutazione	ID_Valutazione, Valutazione	ID_Valutazione
tema	dati delle categorie trattate nel blog	ID_Tema, Nome_Tema	ID_Tema
sottotema	dati delle sottocategorie trattate nel blog	ID_Sottotema, Nome_Sottotema, ID_Tema	ID_Sottotema
layout	dati della parte grafica di un blog	ID_Layout, Nome_Layout	ID_Layout

5.5 DIZIONARIO RELAZIONI

*Nota: quando si parla di utente si intende utente iscritto

Relazione	Descrizione	Tipo	Componenti
R1 (blogsottotema)	Associa un blog ai suoi sottotemi. Un blog può avere zero o più sottotemi. Ogni sottotema può appartenere a più blog.	Molti a molti	blog (1,N) sottotema(0,N)
R2	Associa un blog a un utente. Ogni utente* può creare da zero a più blog. Un blog può avere un solo creatore.	Uno a molti	iscritto (0,N) blog(1,1)
R3	Associa un post a un blog. Ogni blog può avere zero o più post.	Uno a molti	blog(0,N) post(1,1)
R4	Associa un commento all'utente che lo scrive. Ogni utente può scrivere zero o più commenti.	Uno a molti	iscritto(0,N) commento(1,1)
R5 (hascritto)	Associa un post all'utente che lo scrive. Ogni utente può scrivere zero o più post.	Uno a molti	iscritto(0,N) post(1,1)
R6	Associa una valutazione ad un utente. Ogni utente può fare zero o più	Uno a molti	iscritto(0,N) valutazione(1,1)
R7	Associa un commento a un post. Ogni post può avere zero o più commenti	Uno a molti	post(0,N) commenti(1,1)

Relazione	Descrizione	Tipo	Componenti
R8	Associa una valutazione ad un post. Ogni post può avere zero o più valutazioni.	Uno a molti	post(0,N) valutazioni(1,1)
R9	Associa un layout ad un blog. Ogni layout può appartenere a più blog.	Uno a molti	layout(1,N) blog(1,1)
R10	Associa un sottotema ad un tema. Ogni tema può avere uno o più sottotemi.	Uno a molti	sottotema(1,N) tema(1,1)
coautore	Associa un blog ad un utente che ne è coautore. Ogni utente può essere coautore di più blog. Ogni blog può avere solo un coautore.	Uno a molti	iscritto(0,N) blog(0,1)

5.6 DIZIONARIO CHIAVI ESTERNE

Tabella di partenza	Chiave primaria	Tabella di arrivo	Chiave esterna	On DELETE/ UPDATE
iscritto	ID	valutazione	ID_User	Cascade
iscritto	ID	hascritto	ID_User	Cascade
iscritto	ID	commento	ID_User	Cascade
iscritto	ID	blog	Autore	Cascade
iscritto	ID	coautore	ID_Utente	Cascade
blog	ID_Blog	coautore	ID_Blog	Cascade
blog	ID_Blog	post	ID_Blog	Cascade
blog	ID_Blog	blogsottotema	ID_Blog	Cascade
post	ID_Post	valutazione	ID_Post	Cascade
post	ID_Post	hascritto	ID_Post	Cascade
post	ID_Post	commento	ID_Post	Cascade
tema	ID_Tema	sottotema	ID_Tema	Cascade
sottotema	ID_Sottotema	blogsottotema	ID_Sottotema	Cascade
layout	ID_Layout	blog	ID_Layout	Cascade

5.6.1 Vincoli di integrità

Come descritto in tabella, il vincolo ON DELETE CASCADE e ON UPDATE CASCADE è stato scelto per fare in modo che:

- Quando viene cancellato un utente questo porti alla cancellazione di tutto quello a cui è legato, ovvero alla cancellazione di:
 - Valutazioni fatte
 - Post creati
 - Commenti scritti
 - Blog creati
 - Collaborazioni con altri utenti

- Quando si cancella un blog, automaticamente vengano eliminati:
 - Collaborazioni con altri utenti
 - Post contenuti al suo interno
 - Collegamento tra blog e le sottocategorie/categorie in cui era stato catalogato, in modo che non possa comparire se lo si cerca in queste
- Quando viene cancellato un post, spariscano insieme a lui anche:
 - Valutazioni che aveva ricevuto
 - Collegamento tra post e colui che lo ha scritto
 - I commenti fatti su quel post
- Quando si cancella un tema, verranno cancellati automaticamente anche i sottotemi a lui legati
- Quando si cancella un sottotema sparisce anche il collegamento tra lui e i blog con cui era associato
- Quando viene eliminato un layout è come se venisse eliminata la struttura di un blog, quindi in seguito alla sua cancellazione sparirà anche tutto il blog.

Lo stesso discorso fatto per il DELETE vale anche per le modifiche del ON UPDATE CASCADE, portando automaticamente ad un aggiornamento di tutti i campi coinvolti nella relazione.

6. ANALISI MEMORIA E RIDONDANZE

6.1 STIMA APPROSSIMATA DEI COSTI

6.1.1 TAVOLA DEI VOLUMI

Concetto	Tipo	Volume
iscritto	Entità	10 000
blog	Entità	30 000
post	Entità	60 000
commento	Entità	100 000

Concetto	Tipo	Volume
valutazione	Entità	150 000
tema	Entità	50
sottotema	Entità	300
layout	Entità	20
blogsottotema	relazione	90 000
coautore	Relazione	20 000
hascritto	Relazione	50 000

I dati inseriti in tabella sono puramente fittizi al fine di poter dare una stima approssimata dei costi.

Tabella delle operazioni:

- operazioni frequenti: 1000 volte al giorno
- operazioni con frequenza media: 500 volte al giorno
- operazioni a bassa frequenza: 50 volte al giorno

$w_i = 1$ → indica il peso delle operazioni interattive

$w_b = 1$ → indica il peso delle operazioni batch

$a = 2$ coefficiente moltiplicativo per le operazioni di scrittura

Primo Caso (iscritto) NBlogCreati

1 dato ridondante: NBlogCreati

- Estrazione numero di blog creati conoscendo l'ID della tabella iscritto dell'utente interessato

Occupazione memoria:

Definisco l'occupazione di memoria della base di dati, senza dato ridondante, con X.

A questo dato andrò ad aggiungere l'occupazione di memoria data dal dato ridondante NBlogCreati.

Memoria = $X + (4\text{byte} \cdot 30000) = 120\,000$ byte circa

Calcolo dei costi:

Suppongo che l'utente sia autore di 4 blog

Tabella con ridondanza:

Concetto	Costrutto	Accessi	Tipo
blog	Entità	1	Lettura

—> Costi: $100 \cdot 1 \cdot (2 \cdot 0 + 4) = 400$

Tabella senza ridondanza:

Gli accessi in tabella senza dato ridondante saranno 4.

Concetto	Costrutto	Accessi	Tipo
blog	entità	4	Lettura

—> Costi: $100 \cdot 1 \cdot (2 \cdot 0 + 1) = 100$

Responso:

La tabella con ridondanza è più efficiente in termini di prestazioni e il consumo extra di memoria è minimo.

MediaValutazioni

- Estrarre la media delle valutazioni di un dato post avendo ID_Post della tabella post.

Suppongo che un post sia stato valutato 10 volte.

Occupazione memoria:

Definisco l'occupazione di memoria della base di dati, senza dato ridondante, con X.

A questo dato andrò ad aggiungere l'occupazione di memoria data dal dato ridondante MediaValutazioni.

Memoria = $X + (4\text{byte} \cdot 150\,000) = X + 600\,000$

Calcolo dei costi:

Tabella con ridondanza

Concetto	Costrutto	Accessi	Tipo
post	entità	1	Lettura

$$100 * 1 * (2 * 0 + 1) = 100$$

Tabella senza ridondanza:

Concetto	Costrutto	Accessi	Tipo
valutazioni	entità	10	Lettura

$$100 * 1 * (2 * 0 + 10) = 1000$$

Responso:

Si osserva come il caso con ridondanza permetta prestazioni più efficienti, anche se a discapito dell'occupazione di memoria.

7. CREDENZIALI

7.1 Dati account Paypal

Email: sb-6vbf9071829@personal.example.com

Password: X:Q>"G4l

7.2 Dati account BlogArt

Queste sono le credenziali dell'account Google di blogart dove arrivano i dati del form del *contatta Admin*.

Nome: Chiara

Cognome: Bresciani

Nome utente: blogart0097@gmail.com

Password: federica0028

7.3 Dati di accesso utenti

Nome utente: Giulietta

email: giuliabal@gmail.com

Livello: base

password: Ciao1234

Nome utente: Ian
email: lucaian@gmail.com

Livello: premium
password: Ciao1234

Nome utente: Prince
email: priscillagiacobini@gmail.com

Livello: base
password: Ciao1234

Nome utente: Chiaerae
email: chiarabresciani97@gmail.com

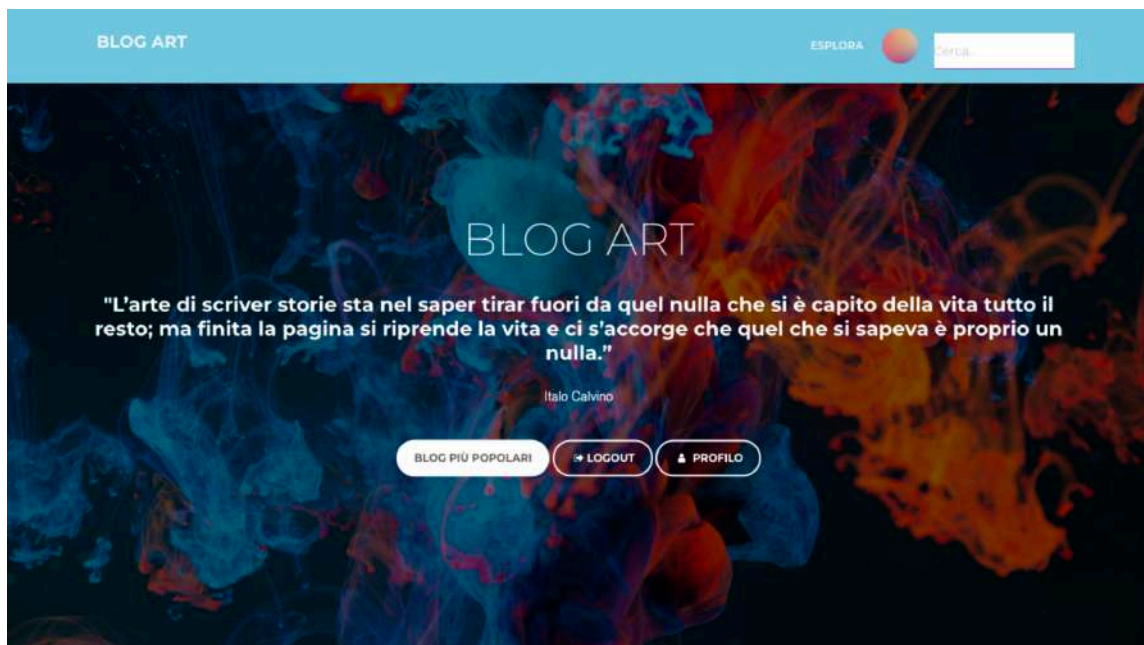
Livello: Admin
password: Ciao1234

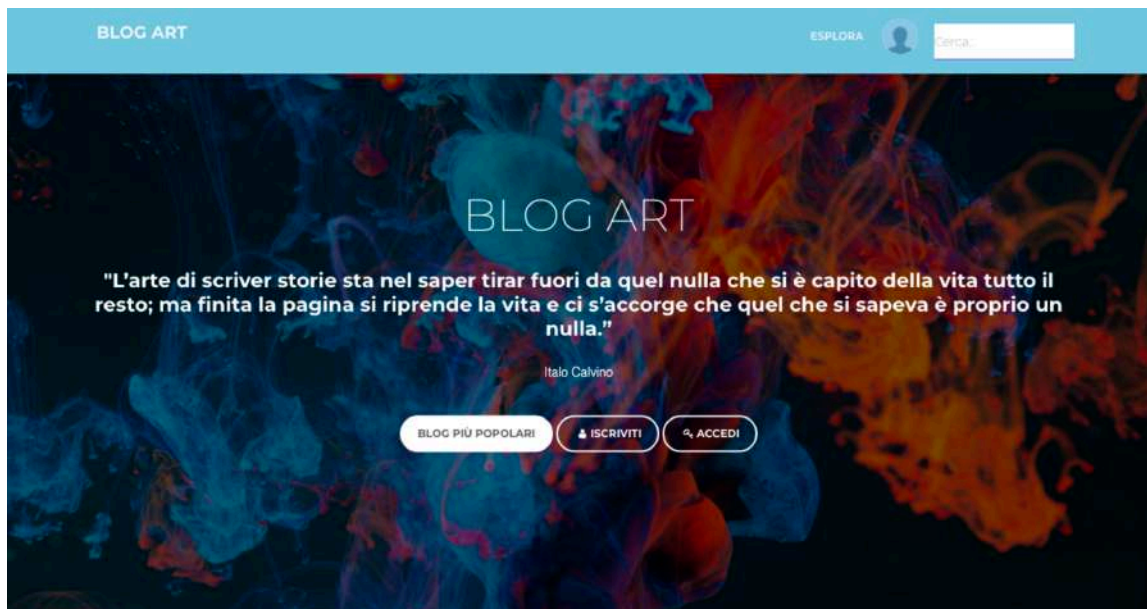
Nome utente: Fede
email: f.papa4@studenti.unipi.it

Livello: Admin
password: Ciao12345

8. ACCESSIBILITÀ

8.1 Home Page: la home-page presenta dei bottoni che permettono all'utente di accedere all'area personale o di iscriversi. Nel caso in cui l'utente abbia già effettuato l'accesso i bottoni cambiano e non permettono più all'utente di iscriversi o di accedere ma gli permettono di visitare la propria aria privata e di effettuare il logout. Queste azioni sono state gestite tramite l'utilizzo dei GET e delle SESSIONI.



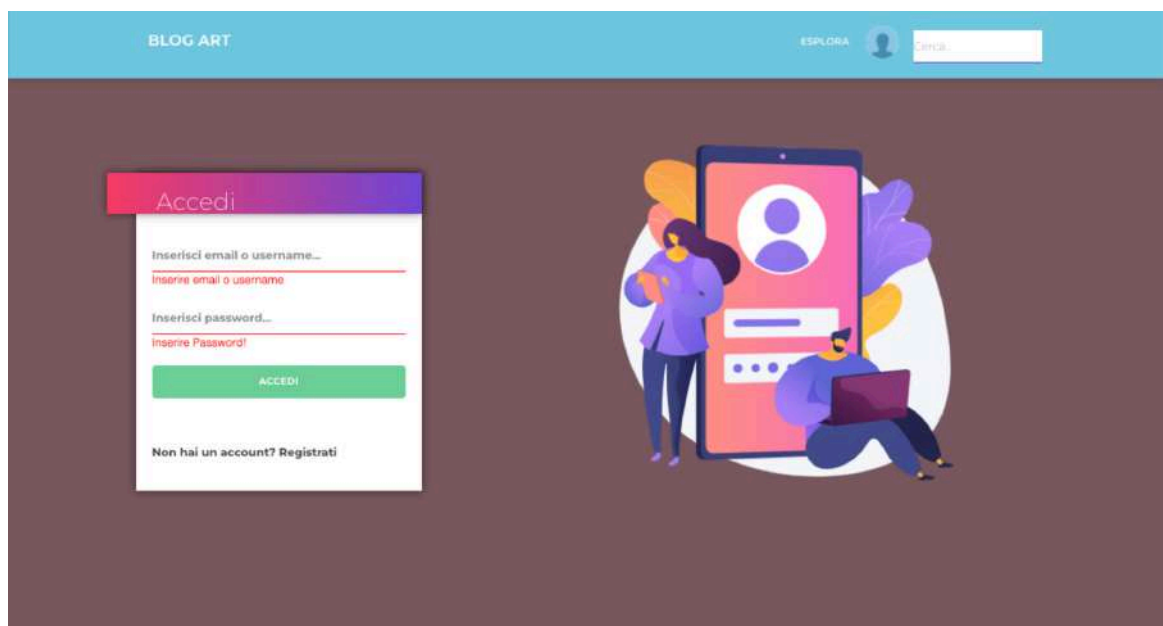


8.2 Sign Up & Login:

Nella pagina Login e Sign Up abbiamo deciso di creare dei controlli per i vari input tramite js con l'utilizzo di chiamate Ajax con lo scopo di:

- Verificare l'univocità dell'username, della mail, del numero di telefono e del documento;
- Verificare che i dati siano in un formato consono tramite l'utilizzo di espressioni regolari.

Abbiamo deciso di utilizzare il js con chiamate Ajax perché permette di manipolare gli errori a video utilizzando le classi HTML. Questo rende per l'utente più semplice capire se i dati inseriti rispettino i parametri adottati dal sito.



BLOG ART

ESPLORA

Crea il tuo blog
I contenuti che pubblichi sul tuo blog sono il cuore pulsante della tua intera presenza online. Sono ciò che ti fanno conoscere al pubblico e sono ciò che dice loro chi sei, cosa fai, e come lo fai.

Rimani in contatto con tutti i tuoi lettori
Condividi contenuti di valore e fai crescere il numero di visualizzazioni sui tuoi post.

Rispetta gli altri utenti
Rispetta le idee altrui e non offendere gli altri utenti della community.

Registrati

Username
Inserire Username

Nome
Inserire Nome

Cognome
Inserire Cognome

Email
Inserire Email

Telefono
Inserire un numero di telefono fisso o cellulare

Documento
Inserire un numero di Documento valido

Password
Inserire Password

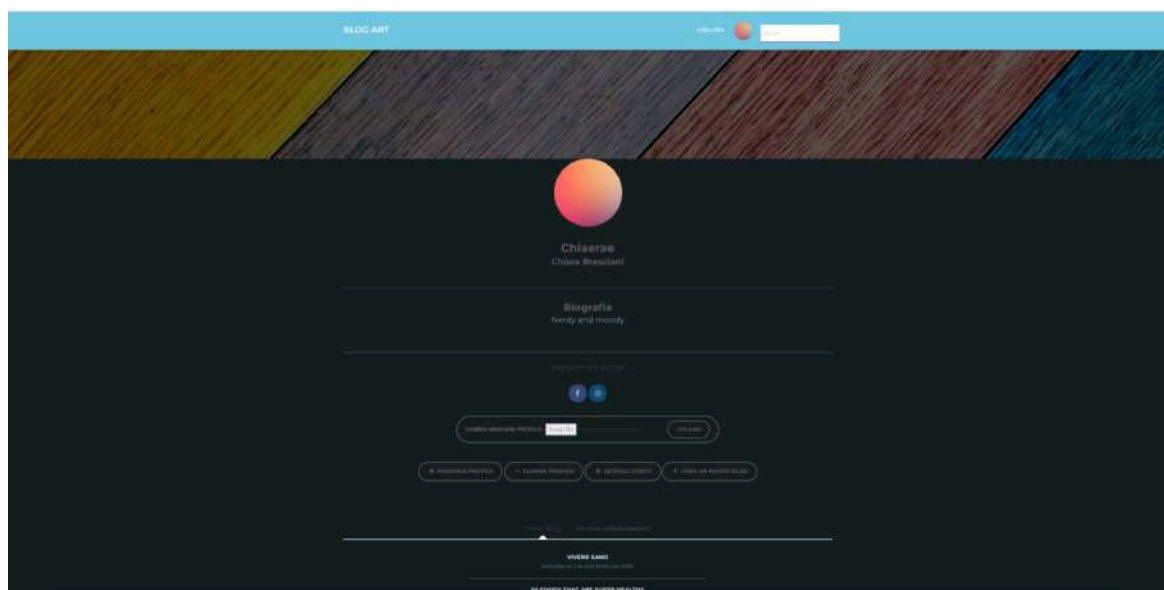
Conferma password
Confermare la Password

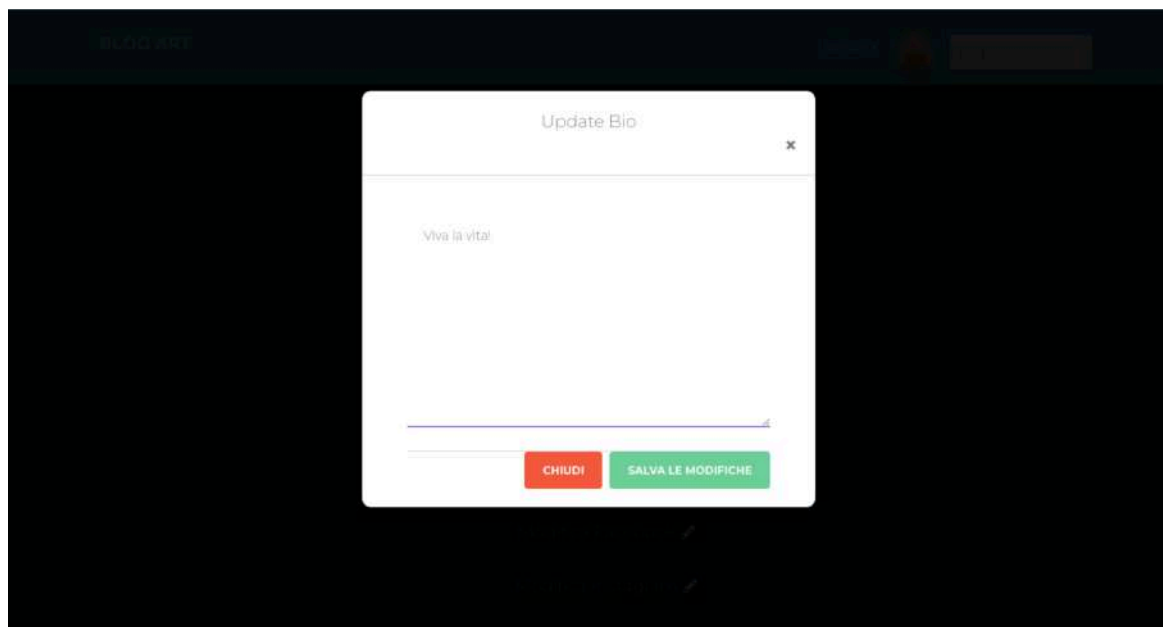
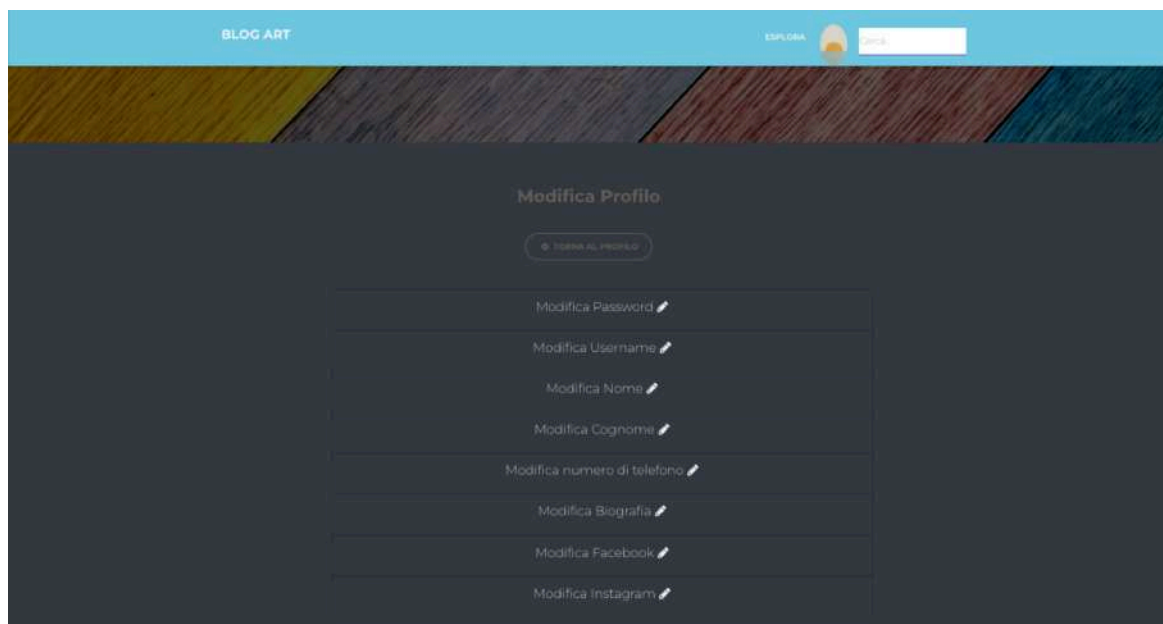
REGISTRATI

Hai già un account?

8.3 Pagina di profilo e modifica profilo:

Nella pagina Index del profilo i vari bottoni sono gestiti da HTML, CSS e PHP. In questa pagina il .js non è stato utilizzato perché le modifiche dei dati utente si svolgono altrove. Nella sezione di modifica profilo ogni area adibita alla aggiunta o modifica dei dati viene gestita tramite “modal”. Questo tag HTML permette di creare dei pop up gradevoli all’utente. Anche qui le modifiche dei dati avvengono tramite js che invia (in caso di non errori da parte dell’utente nell’inserimento dei dati) a funzioni PHP i dati da modificare o aggiungere al record dell’ID utente interessato.

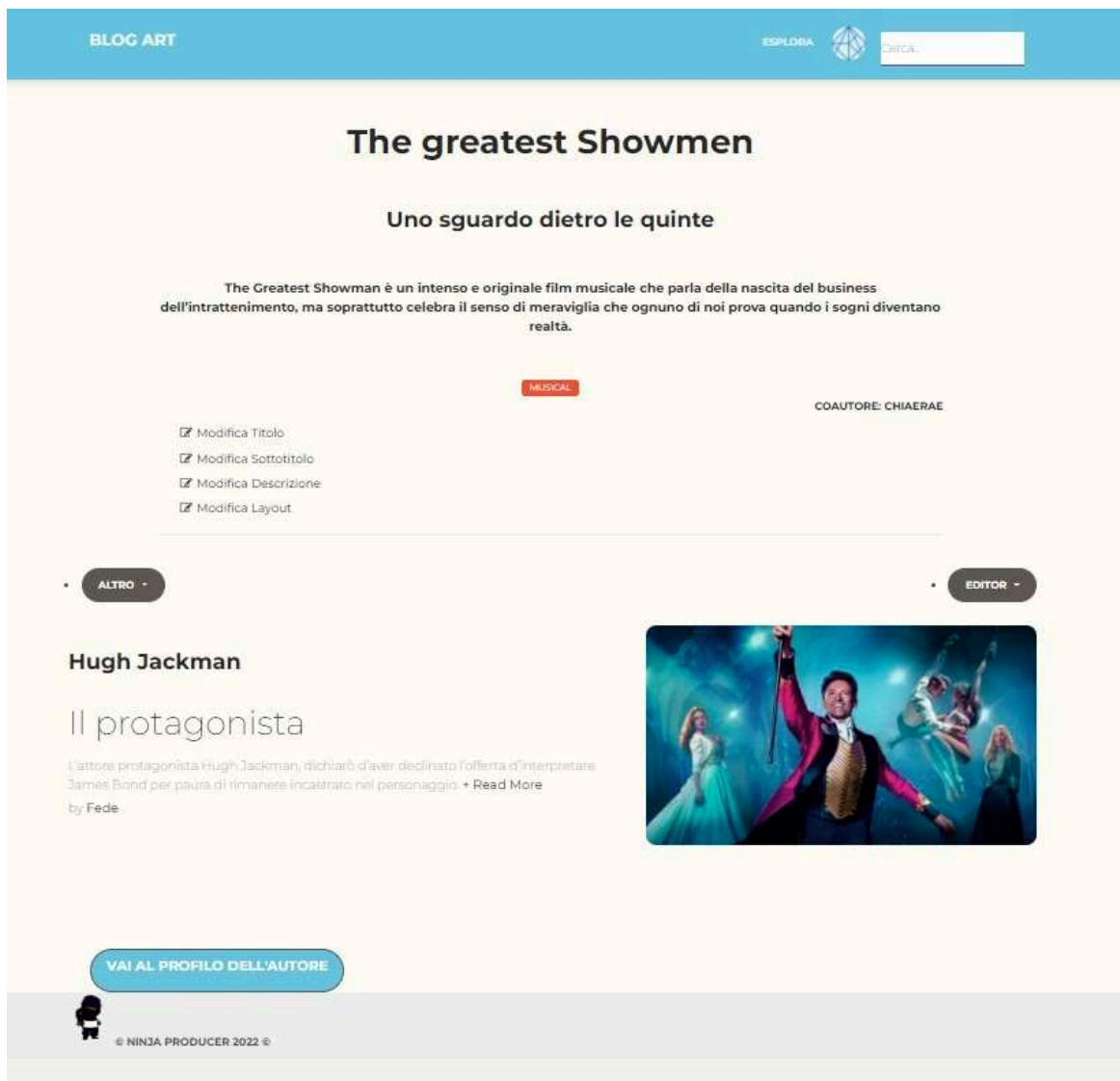
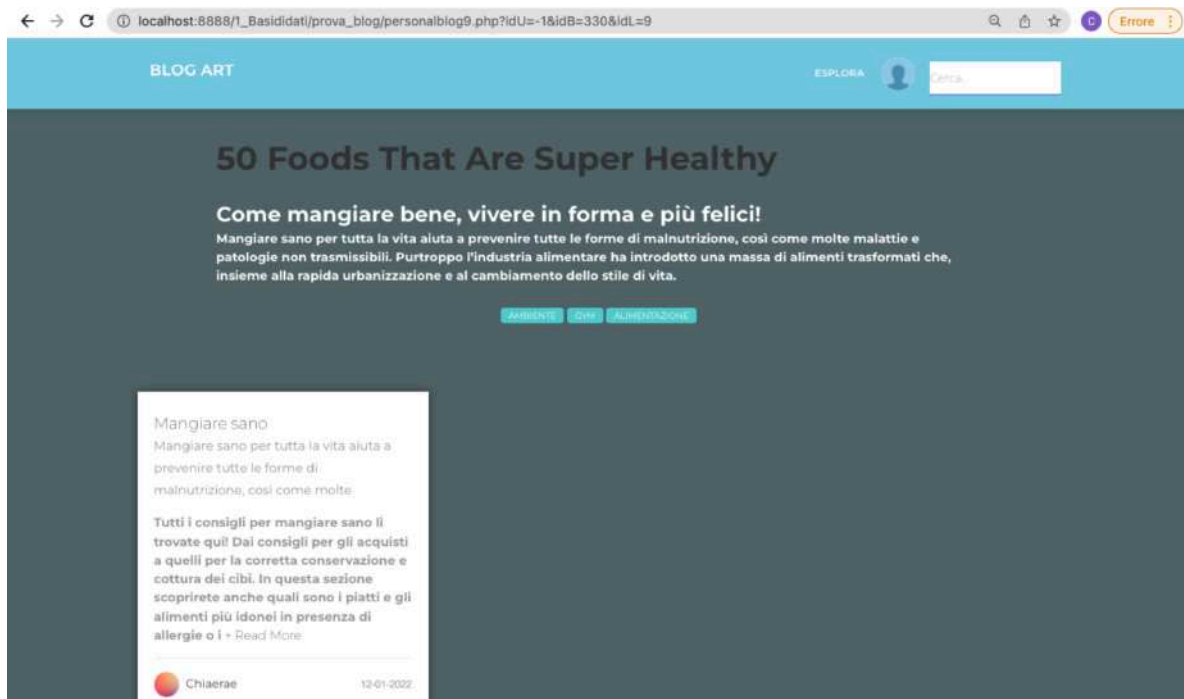




8.4 Pagina PersonalBlog

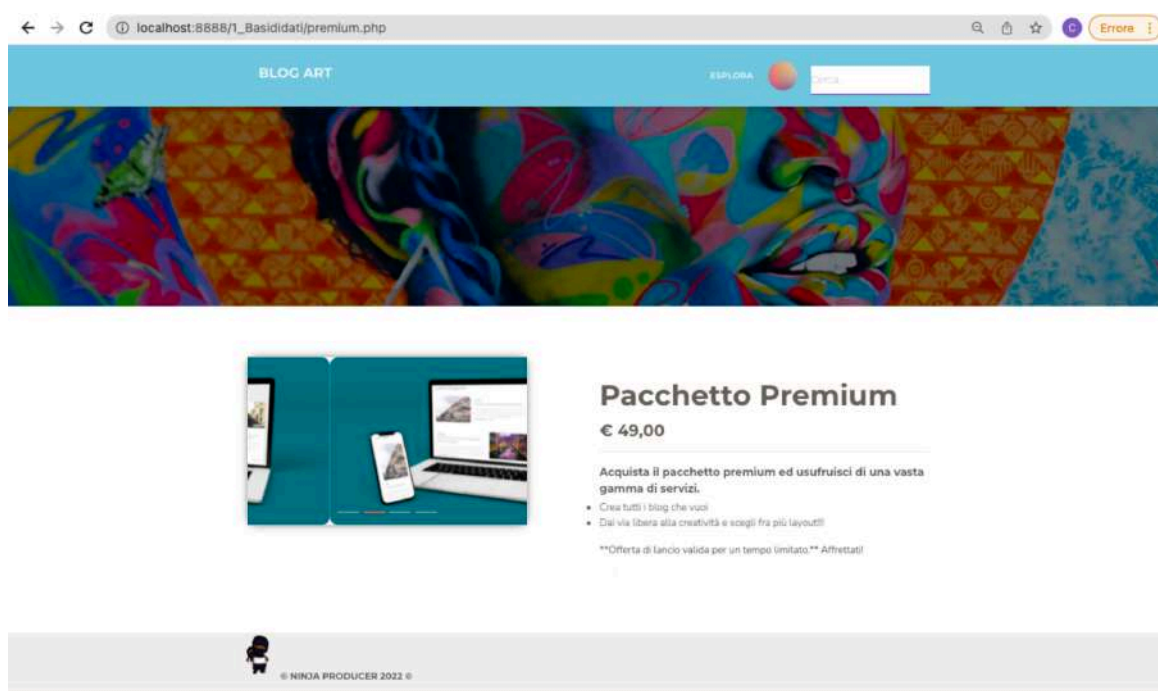
Per la gestione delle modifiche si utilizzano dei bottoni che portano a dei form. Questi form sono gestiti da Ajax. I bottoni sono visibili solo agli amministratori del blog (Autore e Coautore).

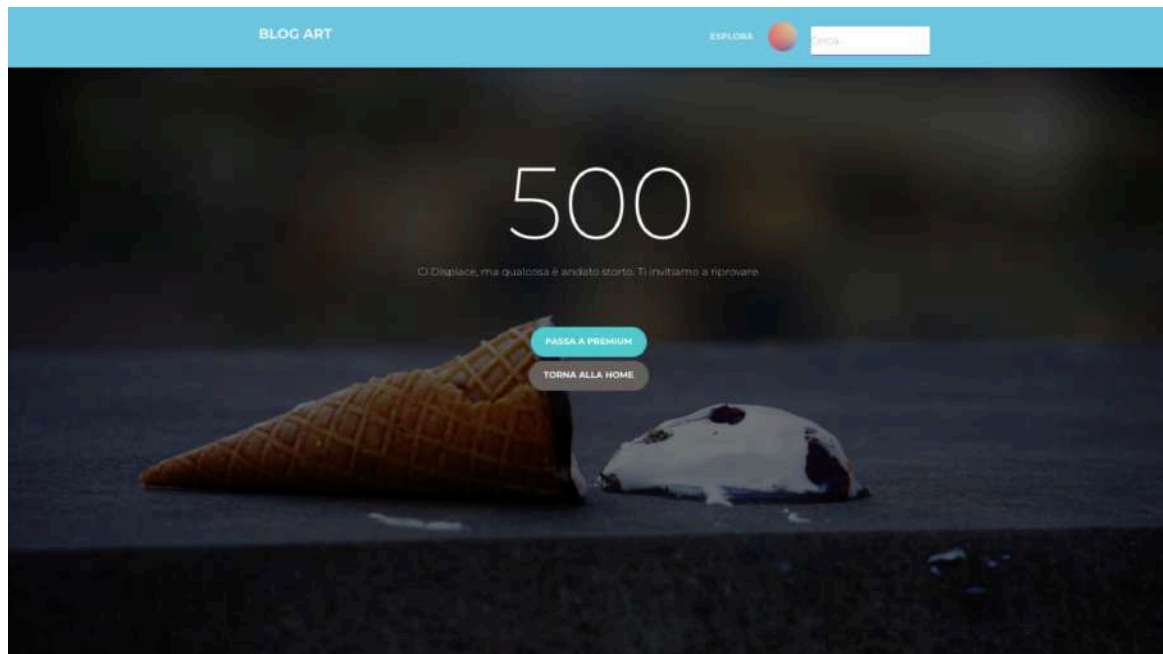
Per quanto riguarda invece la gestione dei layout non abbiamo utilizzato Ajax perché volevamo creare dei layout molto diversi tra loro strutturalmente. Per quanto riguarda lo stile grafico abbiamo deciso di lavorare partendo da dei template che abbiamo adattato in base alle nostre esigenze grafiche. I template utilizzati fanno parte del pacchetto Paper kit di Creative Tim.



8.5 Diventa Premium

La pagina premium.php permette a un utente di Livello base di poter diventare, dopo l'avvenuto successo del pagamento tramite PayPal o carta di credito, un utente Premium. La gestione del pagamento avviene tramite js e in caso di successo (avvenuto pagamento) viene effettuata una chiamata ajax per la modifica del livello utente nel suo record nella tabella iscritto. In caso di insuccesso l'utente viene indirizzato ad una pagina di errore.



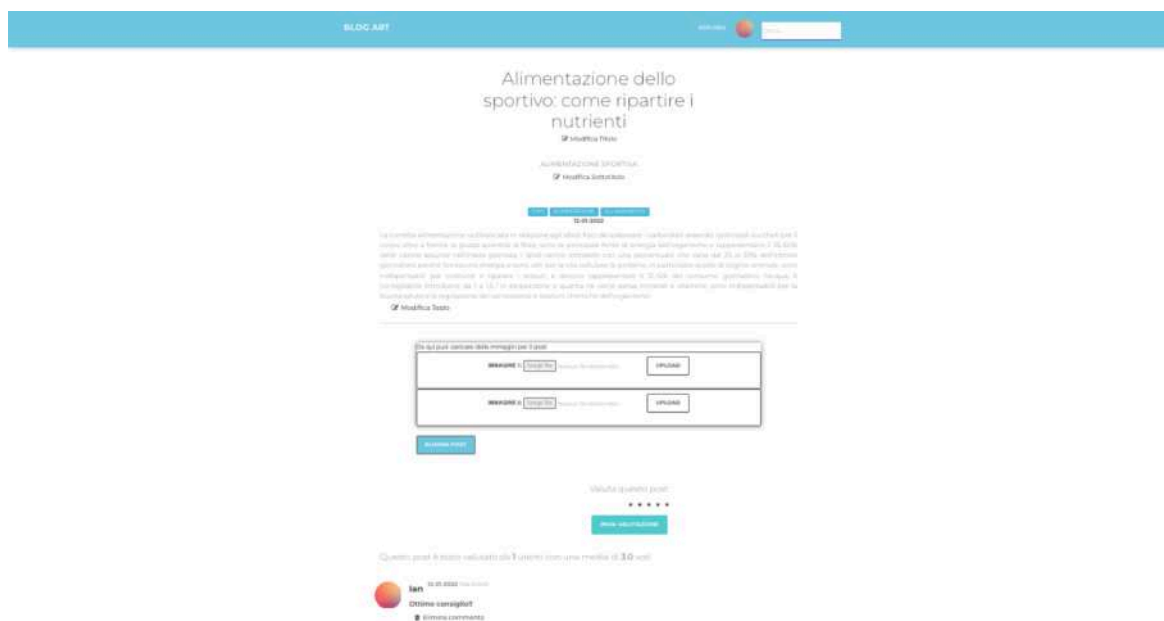


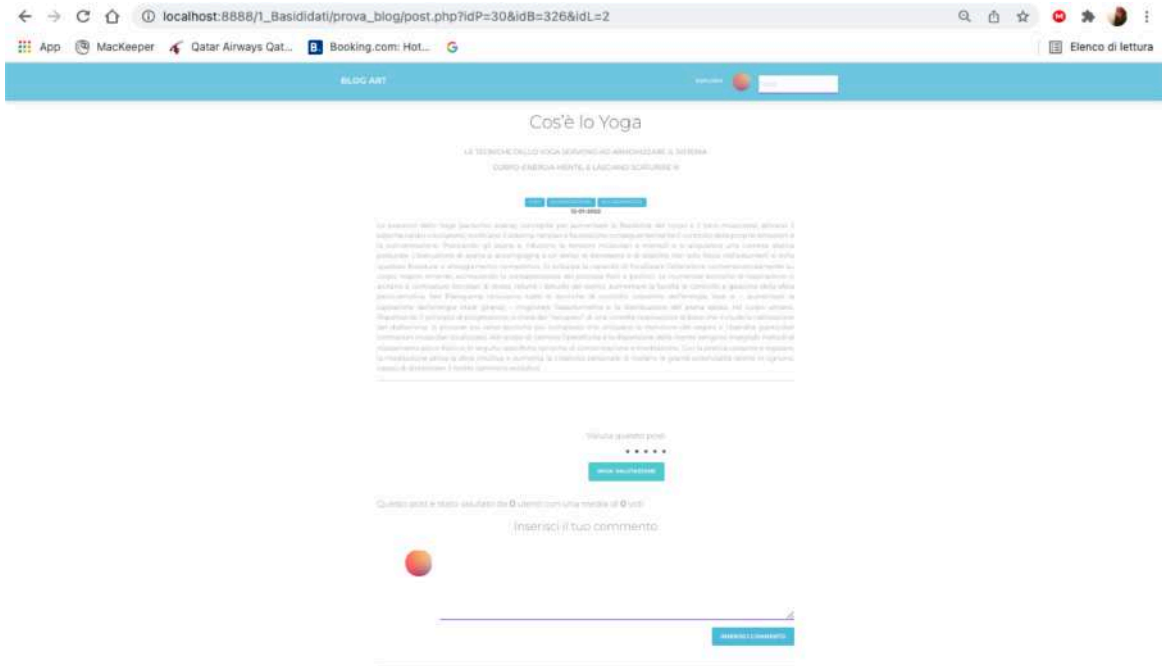
8.6 Pagina Post

La pagina “*post.php*” in base ai privilegi che l’utente possiede (se è un solo visitatore, iscritto o autore del post) la visualizzazione del post cambia.

Un visitatore può solo leggere testo e commenti e vedere la media delle valutazioni del post. Un iscritto ha la facoltà di poter commentare e valutare il post.

L’autore del post e l’autore del blog hanno facoltà di modifica del post, cancellazione dei commenti di altri utenti e eliminazione del post stesso.





8.7 Pagina Categorie-Sottocategorie

Questa sezione del blog è visibile a tutti i tipi di utente che visitano il sito nel medesimo modo. La sezione è organizzata in una navbar che ordina i blog in base alla categorie e successivamente in base alla sottocategoria.

