



Blog di Pica Simone

Analisi dell'applicazione web

I.I.S "G.Ferraris-F.Brunelleschi"

Maturità 2015/2016

INTRODUZIONE

Lo scopo di questo progetto è realizzare un blog personale al fine di dimostrare le capacità acquisite nel corso dell'anno scolastico. La scelta del blog è stata un buon compromesso per integrare un progetto di notevoli dimensioni in un piano di studi che prevedesse anche una preparazione per l'esame finale.

Di conseguenza, la realizzazione del progetto, è servita per approfondire conoscenze tecniche, ma anche, inserendo articoli riguardanti argomenti scolastici, come strumento di ripasso per le materie letterarie che, seppur secondarie nel piano di studi scelto, rivestono un ruolo importante nella formazione di uno studente.

Il documento analizza le varie fasi che sono state compiute per la realizzazione della web application.

Come prima cosa ho ricercato le principali caratteristiche di siti web simili percorrendo i tratti della loro evoluzione. Inizialmente i blog erano formati da una lista di collegamenti catalogati per data. Nel corso degli anni hanno subito radicali modifiche, tant'è che oggi consistono in una serie di articoli ordinati e suddivisi in categorie. In aggiunta, con la loro evoluzione, è cambiato anche il ruolo del lettore che è diventato un utente che interagisce commentando i vari post.

Successivamente a questa prima fase, puramente documentativa, ho stilato una lista degli elementi rilevanti con l'aggiunta di qualche considerazione personale per arricchire funzionalmente l'applicazione. Da questa lista ne ho derivato il corrispettivo diagramma ER attraverso la fase di progettazione concettuale. Il risultato di tale elaborazione è stato uno degli elementi richiesti per procedere con la progettazione logica che, come risultato, ha prodotto lo schema logico. Tale schema è stata la base dell'effettiva implementazione del database.

In parallelo a questa attività di realizzazione della base di dati, ho elaborato il layout del blog e attraverso il linguaggio CSS ne ho definito l'aspetto grafico. Utilizzando la versione CSS3 ho reso il blog responsive, vale a dire in grado di adattarsi graficamente in modo automatico al dispositivo con il quale viene visualizzato.

Inoltre piccole animazioni sono state realizzate sfruttando la potenzialità del JavaScript ed eventuali proprietà dei fogli di stile. L'utilizzo del linguaggio JavaScript, che non è stato affrontato in classe, vuole avvalorare la versatilità, appresa nell'arco del triennio della scuola media superiore, nell'approccio ad un nuovo linguaggio.

Terminate le fasi di progettazione del sito e del database, ho realizzato l'interazione tra le due parti avvalendomi dei linguaggi PHP e SQL in modalità embedded. La libreria mysql è stata utilizzata per garantire il corretto funzionamento delle query all'interno del linguaggio ospite.

Infine, per pubblicare il blog in rete, ho acquistato presso l'azienda Shellrent il pacchetto di Web Hosting Linux e registrato il dominio:

simonepica.it

INDICE

1. **Storia dei blog** - *pagina 4*
2. **Analisi della Web Application** - *pagina 5:11*
3. **Progettazione concettuale – Diagramma ER** - *pagina 12,13*
4. **Progettazione logica – Schema Logico** - *pagina 14:16*
5. **Programming Languages used to implement my web application** - *page 17*
6. **Organizzazione della web application** - *pagina 18:30*
7. **Pagine del blog** - *pagina 31:59*
8. **Hosting** - *pagina 60*
9. **Bibliografia / Sitografia** - *pagina 61*

STORIA DEI BLOG

Un blog è un particolare tipo di sito web caratterizzato dalla successione cronologica di **articoli pubblicati da uno o più autori** in forma più o meno professionale e periodica.

La loro nascita risale al 18 luglio 1997 quando **Dave Winer** rilasciò Frontier's NewsPage, il primo software di comunicazione multimediale: si trattava di un PROTO-BLOG. Il primo blog fu definitivamente aperto il 23 dicembre del 1997 da **Jorn Berger** che, affidandosi alla piattaforma di Winer, pubblicò una lista di link riguardanti la sua passione di caccia.

In realtà la forma utilizzata da Berger non fu innovativa in quanto già nel 1993 **Sir Timothy Berners-Lee**, il co-inventore del World Wide Web, aveva pubblicato una lista di collegamenti. Analogamente, nel giugno dello stesso anno, il National Center for Supercomputing Applications (NCSA) realizzò un sito web, che prese il nome di "What's News?", in cui si elencavano i nuovi siti internet **catalogati per data**.

Berger, al contrario dei siti web pubblicati nel 1993, utilizzò la parola "weblog" per descrivere la propria lista di link e per questo motivo, il suo blog, viene considerato il primo della storia. Come si può facilmente dedurre il termine weblog deriva dalla fusione delle parole Web e Log che possiamo tradurre in "Diario in rete".

Nel 1999 la parola venne troncata in "blog" da **Peter Merholz** che lo utilizzò come verbo, scrivendo "we blog" nel suo sito Peterme.com e dando origine ad una serie di vocaboli da esso derivati: *blogger* (colui che scrive un blog), *bloggare* (scrivere un blog), *blogosfera* (insieme di tutti i blog).

In Italia i primi blog si diffusero dal 2001 a seguito della nascita delle prime piattaforme americane di blogging. Nel 1999 gli statunitensi Evan Williams e Meg Hourihan svilupparono "**Blogger**", il primo software che consentì di creare e gestire il proprio blog. Negli anni successivi nacquero altre piattaforme di gestione come *WordPress*, *MySpace* e la piattaforma italiana *Altevista* il cui uso non è limitato solo al diario virtuale.

Con i software di gestione cambiò radicalmente anche la struttura dei blog, infatti, dalle prime liste di link di Berger, si iniziarono ad avere i primi "Post": **articoli identificati da un titolo, data di pubblicazioni e alcune parole chiave: i tag**.

Tra il 2002 e il 2007, la possibilità di aprire un blog senza possedere un dominio e la facilità nel farlo, fece sì che i blog godessero di un periodo di grande fortuna presentandosi come il metodo più rivoluzionario di comunicazione on line e raggiungendo i 70 milioni di blog aperti in tutto il mondo a marzo 2007.

La diffusione portò nuovi strumenti come la possibilità di **lasciare commenti da parte dei lettori** ed il raggruppamento dei messaggi già pubblicati **negli archivi ordinati per mesi o per anni**. Inoltre, per agevolare la ricerca, molti blogger decisero di **suddividere i loro post in categorie**.

Tra il 2009 e il 2010, la nascita dei social network fece registrare un periodo di crisi del blog; tuttavia, la loro diffusione fu tale da portare una classificazione che attualmente risulta molto varia e che differisce, non solo per il contenuto, ma anche per come viene scritto.

Generalmente vengono catalogati in blog diario, blog tematico e blog letterario.

Nel blog diario l'autore scrive le proprie esperienze giornaliere al contrario dei blog tematici che sono dedicati ad un argomento specifico. Gli ultimi sono caratterizzati dalla presenza, al loro interno, del lessico letterario.

ANALISI DELLA WEB APPLICATION

Sulla base della storia è possibile individuare le caratteristiche principali dei blog che verranno riportate di seguito:

- Articoli visualizzati in ordine cronologico (Catalogati per data).
- Articoli pubblicati da uno o più autori.
- Articolo identificato da un titolo, data di pubblicazione e alcuni tag.
- Commenti da parte dei lettori.
- Articoli raccolti in archivi ordinati per mesi o per anni.
- Articoli suddivisi in categorie.

Oltre a ciò nella web application ho deciso di aggiungere altre caratteristiche per rendere il mio diario virtuale personalizzato. Per questo motivo gli utenti, oltre alla possibilità di inserire commenti, potranno salvare o apprezzare gli articoli.

Inoltre ogni utente sarà caratterizzato da un ruolo così da poter suddividere i lettori, i blogger e gli amministratori.

I blogger saranno utenti abilitati a inserire articoli, al contrario degli utenti amministratori che avranno pieni poteri nel blog.

Per arricchirlo ulteriormente ho dotato il diario di rete di un spazio in cui è possibile caricare foto. Di conseguenza possiamo affermare che la web application rientri pure nell'ambito dei photoblog, anche se viene privilegiato il testo alle immagini.

Anche in questo caso una persona iscritta può decidere di apprezzare una immagine inserita nella galleria.

Entità Articolo:

L'entità *articolo* contiene tutte le informazioni sugli articoli. Ogni articolo è caratterizzato da un titolo e un contenuto che sono obbligatori, contrariamente all'immagine di copertina (sfondo/background) che può essere omessa. Inoltre ogni post deve essere inserito in una delle categorie.

Oltre a ciò, ad un articolo, si possono associare una serie di parole chiave che permetteranno una ricerca migliore all'interno del blog ed un indicizzazione più efficace da parte dei motori di ricerca.



Entità Foto:

L'entità *foto* contiene tutte le immagini che compongono la galleria. L'attributo titolo non è obbligatorio diversamente dall'attributo immagine e dal legame con l'entità categoria.

L'attributo immagine è di tipo stringa la cui funzione è di puntatore ad una foto salvata nel file system storage. La scelta di non utilizzare il campo blob, che mi avrebbe permesso di salvare le immagini direttamente nel database, è dovuta ad una maggiore semplicità gestionale. Inoltre ricerche su questo tipo di campo non si sono rivelate positive, in quanto l'occupazione di più spazio comporta degli svantaggi in termini di backup e prestazioni.



Entità Categoria:

L'entità *categoria* contiene le informazioni sulle categorie del blog. Per ciascuna categoria viene richiesto obbligatoriamente il nome, al contrario della descrizione e dell'icona che sono facoltative. L'attributo icona è di tipo stringa e si appoggia sul font-awesome sviluppato da Dave Gandy. Nella web application quando vogliamo associare una figura ad una categoria basterà inserire la stringa che identifica l'icona scelta.



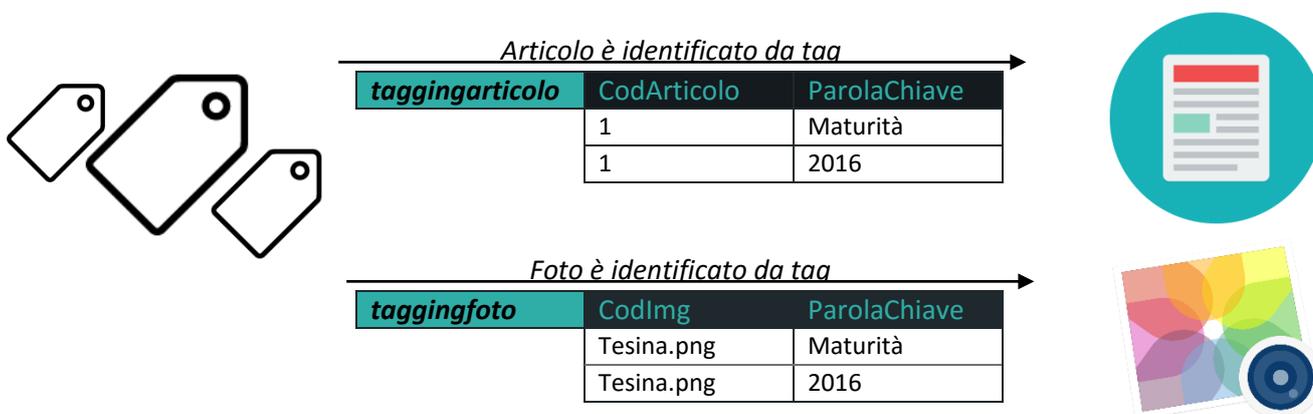
Entità Tag:

L'entità *tag* consiste in una raccolta di parole. In altri termini può essere definita come un dizionario che cresce in rapporto alla pubblicazione di nuovi articoli o foto.



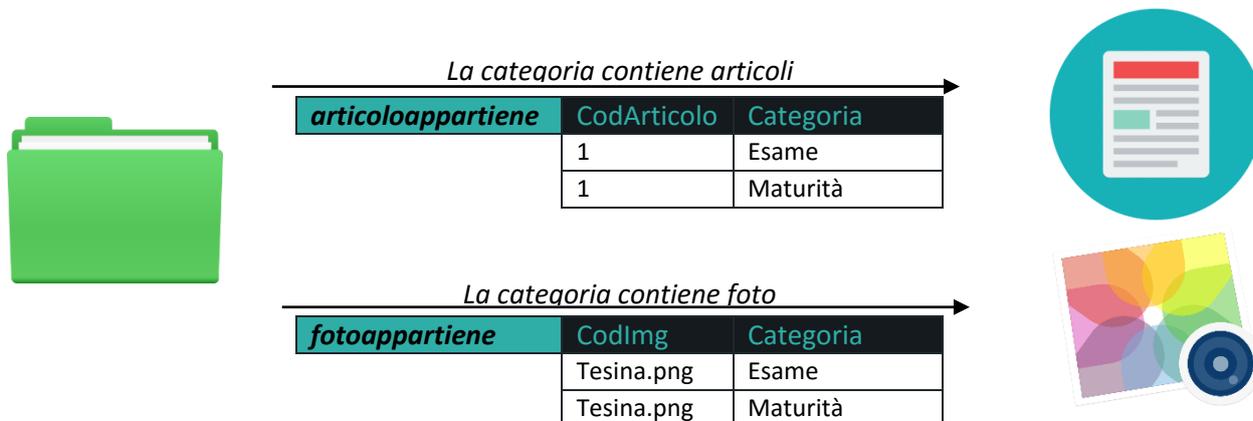
Associazione èLegato (Articolo-Tag/Foto-Tag):

L'associazione *èLegato* realizza i legami che gli articoli e le foto hanno con le parole chiave. Si tratta di una semplice associazione con cardinalità N:N che nella fase di traduzione viene mappata creando una nuova relazione che contiene le chiavi esterne delle entità che lega e che insieme compongono la chiave primaria.



Associazione appartiene(Articolo-Categoria/Foto-Categoria):

L'associazione *appartiene* realizza i legami che gli articoli e foto hanno con le categorie. Anche in questo, nella traduzione in schema logico, viene mappata seguendo la procedura delle associazioni con cardinalità N:N.

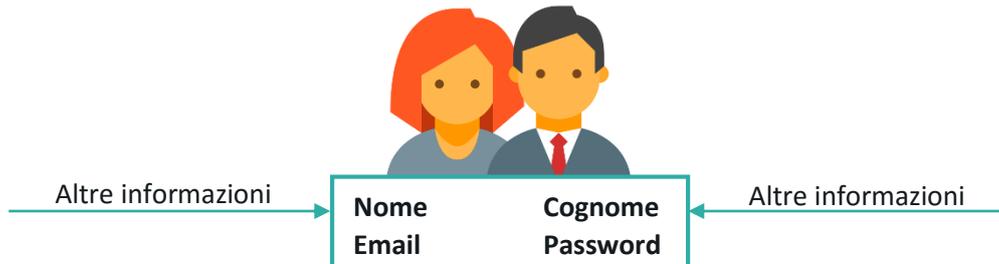


Entità Utente:

L'entità *utente* contiene le informazioni relative alle persone iscritte al blog. Per ogni utente si vuole memorizzare obbligatoriamente nome, cognome, email, password e ruolo, al contrario tutti gli altri dati

sono facoltativi. Il campo avatar, così come per le foto della galleria, rappresenta un puntatore ad immagini caricate nel file system storage. L'indirizzo email viene utilizzato per effettuare il login al blog, di conseguenza deve essere univoco. Nonostante ciò, come chiave primaria, è stata creata una chiave artificiale per consentire all'utente di mantenere nascosta la propria email nella web application.

Osservando il diagramma ER notiamo subito che l'entità si lega: con quattro associazioni all'entità articolo, con due associazioni all'entità foto, con un'associazione all'entità ruolo e un'ulteriore associazione lega la stessa entità.



Entità Ruolo:

L'entità contiene tutti i possibili ruoli che possono assumere gli utenti. Di default i ruoli che verranno inseriti sono:

Ruolo	Descrizione
Amministratore	<i>Piene funzioni nella web application.</i>
Blogger	<i>Funzioni aggiuntive per inserire/modificare/cancellare articoli/foto.</i>
Utente	<i>Identifica i normali lettori del blog.</i>



Associazione commenta(Utente-Articolo):

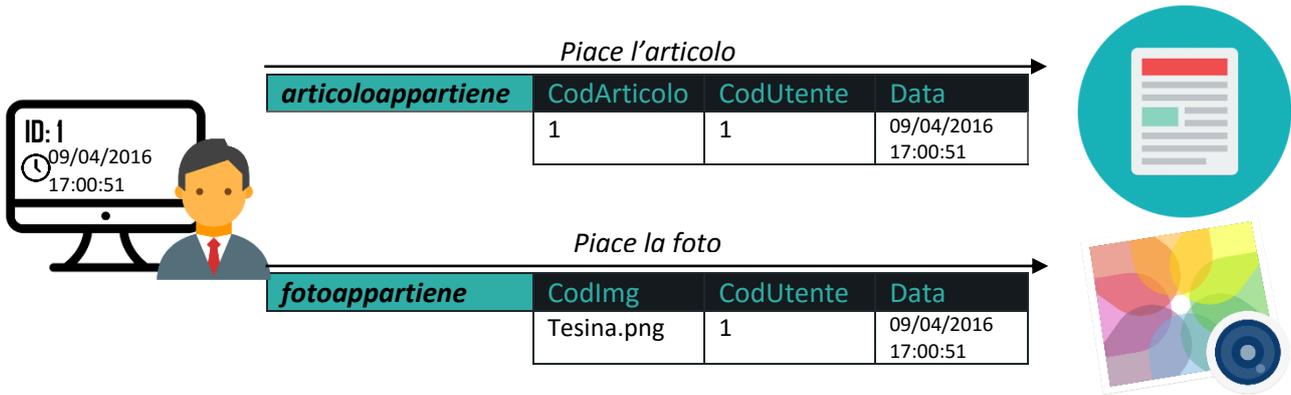
L'associazione *commenta* realizza i legami tra l'entità utente e l'entità articolo/foto. Questa associazione è caratterizzata da due attributi: data, memorizza il momento dell'azione, e contenuto registra il testo inserito dall'utente.

Durante la fase di mapping, nella relazione derivante dall'associazione, la chiave primaria era composta dalle chiavi esterne con l'aggiunta del campo data per permettere agli utenti di commentare più volte lo stesso articolo. Di conseguenza, data la lunghezza della chiave, ho deciso di creare una chiave artificiale.



Associazione piace(Utente-Articolo/Utente-Foto):

L'associazione *piace* determina quali utenti hanno apprezzato determinati articoli o foto. L'attributo dell'associazione (Data) viene inglobato nella relazione che nasce, durante la fase di traduzione in schema logico, delle associazioni con cardinalità N:N.



Associazione subscribe (Utente-Articolo):

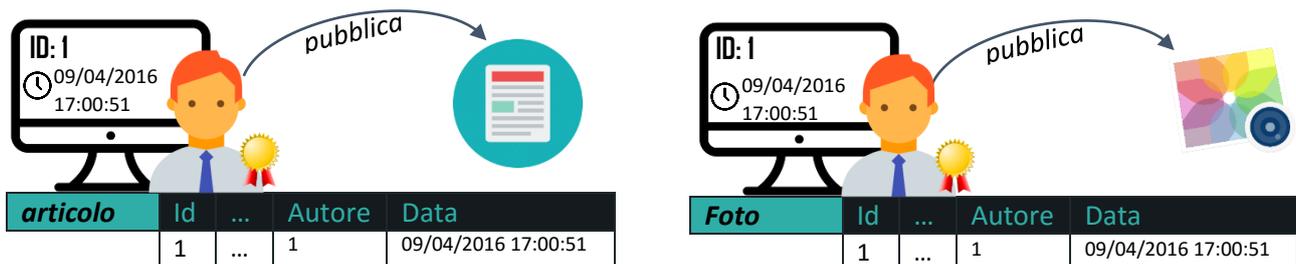
L'associazione *subscribe* permette agli utenti di salvare gli articoli per poi leggerli in futuro. Nella fase di mapping, l'attributo data, che indica il momento in cui l'utente si sottoscrive all'articolo, viene assorbito dalla relazione che deriva dalla traduzione dell'associazione con cardinalità N:N.



Associazione pubblica (Utente-Articolo/Utente-Foto):

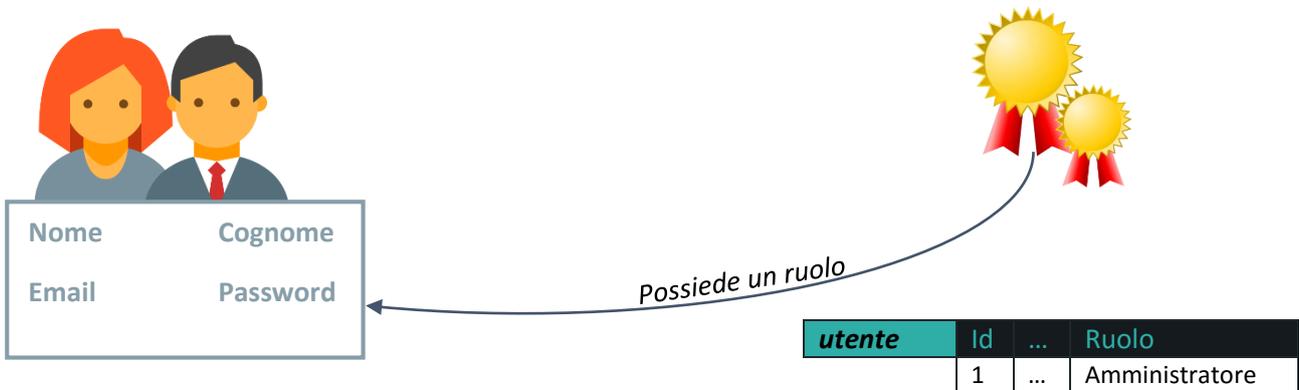
L'associazione *pubblica* determina l'autore dell'articolo/foto. La cardinalità è 1:N, per questo motivo viene mappata aggiungendo la chiave esterna e l'attributo data, che determina il momento della pubblicazione, alla relazione derivante dalla traduzione dell'entità articolo/foto.

Nell'implementazione, prima di confermare l'operazione di inserimento, verrà effettuato un controllo per determinare se l'utente è abilitato ad effettuare tale azione in quanto solamente gli utenti con *ruolo Amministratore o Blogger* sono autorizzati.



Associazione possiede(Utente-Ruolo):

L'associazione *possiede* lega l'utente al ruolo assegnato. Durante la fase di mapping l'associazione viene tradotta aggiungendo la chiave esterna alla relazione che deriva dall'entità utente.



Associazione èAmico(Utente-Utente):

Questa parte del database non è stata implementata per motivi di tempo, perciò la descrizione sarà più completa ed approfondita.
 I riferimenti all'associazione saranno presenti solo nella fase di progettazione del database

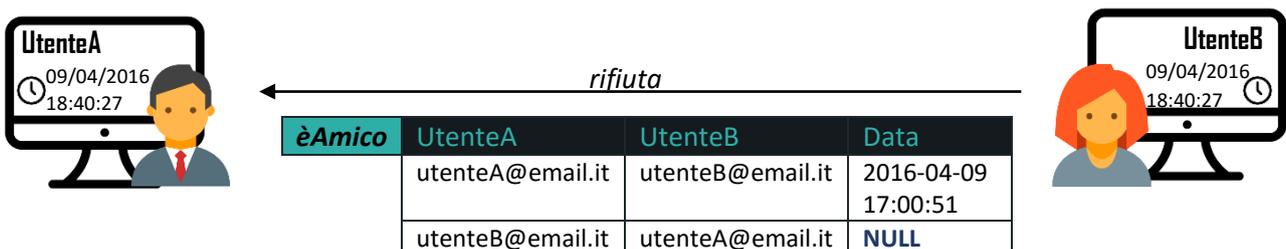
L'associazione *èAmico* rappresenta i legami di amicizia tra gli utenti. L'applicazione prevede che un UtenteA possa inviare una richiesta di amicizia ad un UtenteB che a sua volta può decidere di accettare o rifiutare. L'associazione ha cardinalità N:N per questo motivo nella traduzione a schema logico darà origine a una nuova relazione che avrà come chiave primaria la combinazione delle chiavi esterne che riferiscono entrambe alla relazione derivante dall'entità utente.

• COME FUNZIONA?



La richiesta di amicizia da parte dell'UtenteA verso l'UtenteB comporta l'inserimento di un record nella tabella *èAmico*. A questo punto l'UtenteB visualizzerà nel suo profilo una notifica che gli consentirà di accettare o rifiutare la richiesta.

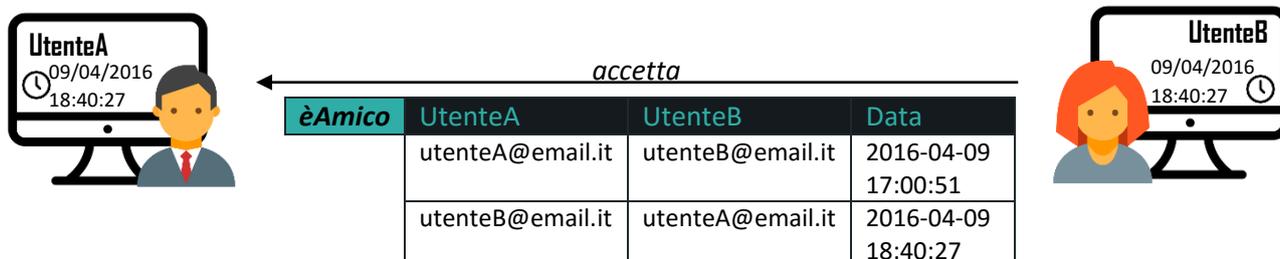
• UtenteB RIFIUTA LA RICHIESTA



La risposta dell'UtenteB alla richiesta di amicizia dell'UtenteA comporta l'inserimento di un nuovo record nella tabella con il valore del campo "Data" a **NULL** che ne indica il rifiuto.

In un secondo momento l'UtenteB potrà accettare l'amicizia modificando la data attraverso un link presente nel profilo dell'UtenteA.

• UtenteB ACCETTA LA RICHIESTA



Quando l'UtenteB accetta la richiesta, nella tabella viene inserito un nuovo record con il *campo Data* valorizzato che ne indica la conferma.

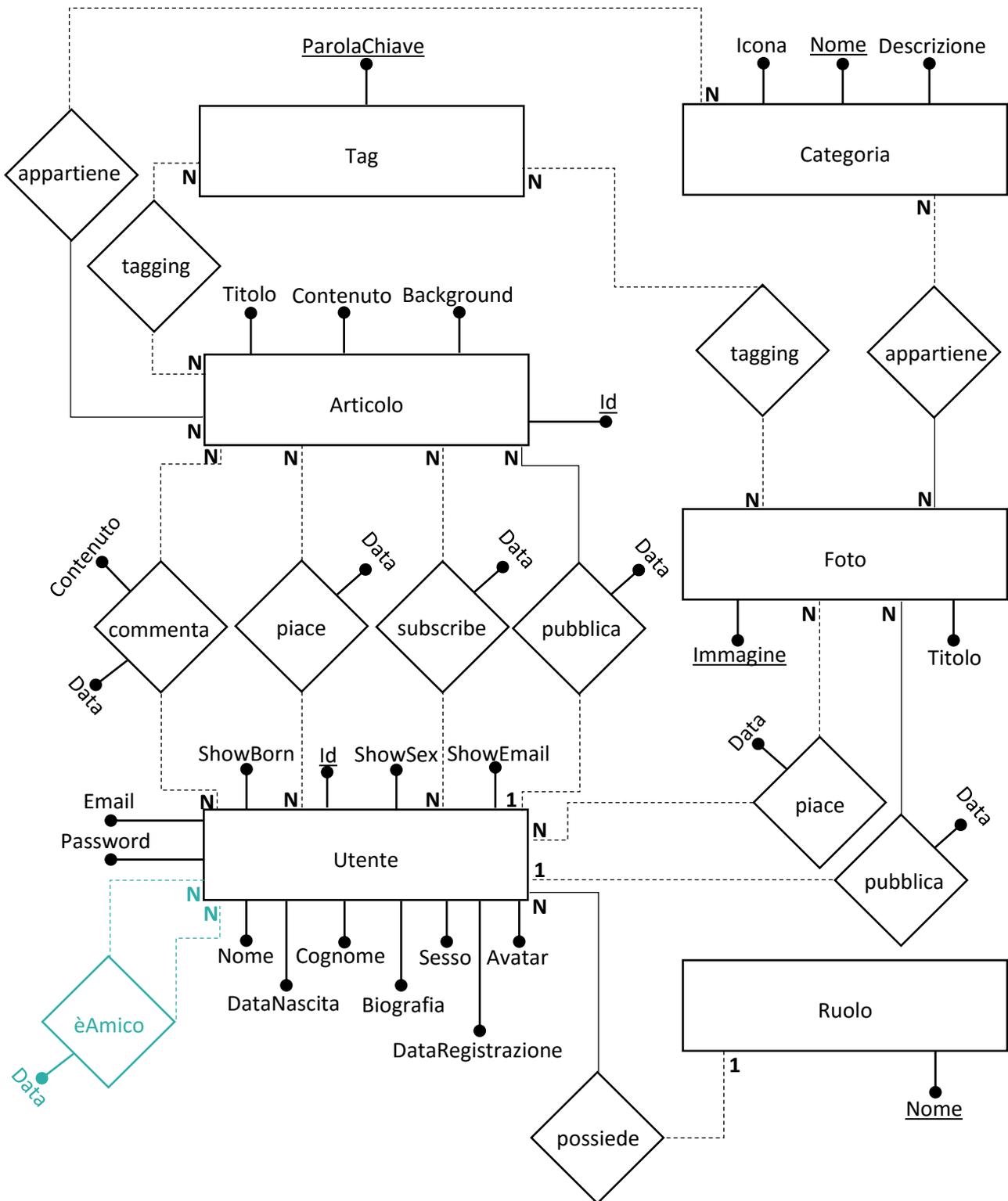
Questa tecnica permette di mantenere informazioni sui legami di amicizia, infatti analizzando la seguente tabella:

èAmico	UtenteA	UtenteB	Data
	utenteA@email.it	utenteB@email.it	2016-04-09 17:00:51
	utenteB@email.it	utenteA@email.it	2016-04-09 18:40:27

Possiamo affermare che:

- La combinazione di due record che coinvolgono gli stessi utenti rappresenta la richiesta di amicizia e la conferma/rifiuto.
- Confrontando le due date (2016-04-09 17:00:51 e 2016-04-09 18:40:27) è possibile determinare che il record con la data più recente indica la conferma da parte dell'utente indicato nel **campo UtenteA** alla richiesta di amicizia effettuata dall'utente presente nel **campo UtenteB**.
- Il **campo Data** uguale a **NULL** indica che l'utente specificato dal **campo UtenteA** ha attualmente rifiutato la richiesta dell'utente presente nel **campo UtenteB**.

PROGETTAZIONE CONCETTUALE - DIAGRAMMA ER



L'associazione colorata di celeste (èAmico) non è stata implementata nell'applicazione.

VINCOLI DI INTEGRITÀ:

V1(Articolo): Articolo.Titolo NOT NULL

V2(Articolo): Articolo.Contenuto NOT NULL

V1(Utente): Utente.DataNascita<Utente.DataRegistrazione

V2(Utente): Utente.DataNascita>DataAttuale-120Anni

V3(Utente): Utente.Sesso IN (M,F)

V4(Utente): Utente.Password NOT NULL

V5(Utente): Utente.Nome NOT NULL

V6(Utente): Utente.Cognome NOT NULL

V7(Utente): Utente.Email NOT NULL

V1(commenta): commenta.Contenuto NOT NULL

PROGETTAZIONE LOGICA - SCHEMA LOGICO

Articolo (Id, Titolo, Contenuto, Background, Autore, Data)

Foto (Immagine, Titolo, Autore, Data)

Categoria (Nome, Descrizione, Icona)

Tag (ParolaChiave)

Utente (Id, Email, Password, Nome, Cognome, DataNascita, Biografia, Sesso, Avatar, DataRegistrazione, ShowBorn, ShowSex, ShowEmail, Ruolo)

Ruolo (Nome)

articoloAppartiene (CodArticolo, Categoria)

fotoAppartiene (CodImg, Categoria)

taggingArticolo (CodArticolo, ParolaChiave)

taggingFoto (CodImg, ParolaChiave)

commenta (Id, CodArticolo, CodUtente, Data, Contenuto)

piaceArticolo (CodArticolo, CodUtente, Data)

subscribe (CodArticolo, CodUtente, Data)

piaceFoto (CodImg, CodUtente, Data)

èAmico (UtenteA, UtenteB, Data)

VINCOLI DI INTEGRITÀ:

V1(Articolo): Articolo.Titolo NOT NULL

V2(Articolo): Articolo.Contenuto NOT NULL

V3(Articolo): Articolo.Autore NOT NULL

V4(Articolo): Articolo.Autore=Utente.Id

V5(Articolo): Articolo.Data DEFAULT CURRENT_TIMESTAMP

V1(Foto): Foto.Autore NOT NULL

V2(Foto): Foto.Autore=Utente.Id

V3(Foto): Foto.Data DEFAULT CURRENT_TIMESTAMP

V1(Categoria): Categoria.Icona DEFAULT "fa-folder-open"

V1(Utente): Utente.DataNascita<Utente.DataIscrizione

V2(Utente): Utente.DataNascita>DataAttuale-120Anni

V3(Utente): Utente.Sesso IN ('M', 'F')

V4(Utente): Utente.Password NOT NULL
V5(Utente): Utente.Nome NOT NULL
V6(Utente): Utente.Cognome NOT NULL
V7(Utente): Utente.Email NOT NULL
V8(Utente): Utente.Ruolo=Ruolo.Nome
V9(Utente): Utente.Ruolo NOT NULL
V10(Utente): Utente.Dataiscrizione DEFAULT CURRENT_TIMESTAMP
V11(Utente): Utente.ShowBorn DEFAULT TRUE
V12(Utente): Utente.ShowSex DEFAULT TRUE
V13(Utente): Utente.ShowEmail DEFAULT TRUE
V14(Utente): Utente.Avatar DEFAULT default.jpg
V15(Utente): Utente.Email UNIQUE
V1(articoloAppartiene): articoloAppartiene.CodArticolo=Articolo.Id
V2(articoloAppartiene): articoloAppartiene.Categoria=Categoria.Nome
V1(fotoAppartiene): fotoAppartiene.CodImg=Foto.Immagine
V2(fotoAppartiene): fotoAppartiene.Categoria=Categoria.Nome
V1(taggingArticolo): taggingArticolo.CodArticolo=Articolo.Id
V2(taggingArticolo): taggingArticolo.ParolaChiave=Tag.ParolaChiave
V1(taggingFoto): taggingFoto.CodImg=Foto.Immagine
V2(taggingFoto): taggingFoto.ParolaChiave=Tag.ParolaChiave
V1(commenta): commenta.Contenuto NOT NULL
V2(commenta): commenta.CodArticolo=Articolo.Id
V3(commenta): commenta.CodUtente=Utente.Id
V4(commenta): commenta.Data DEFAULT CURRENT_TIMESTAMP
V5(commenta): commenta.CodArticolo NOT NULL
V6(commenta): commenta.CodUtente NOT NULL
V1(piaceArticolo): piaceArticolo.CodArticolo=Articolo.Id
V2(piaceArticolo): piaceArticolo.CodUtente=Utente.Id
V3(piaceArticolo): piaceArticolo.Data DEFAULT CURRENT_TIMESTAMP
V1(subscribe): subscribe.CodArticolo=Articolo.Id
V2(subscribe): subscribe.CodUtente=Utente.Id
V3(subscribe): subscribe.Data DEFAULT CURRENT_TIMESTAMP

V1(piaceFoto): piaceFoto.CodImg=Foto.Immagine

V2(piaceFoto): piaceFoto.CodUtente=Utente.Id

V1(èAmico): èAmico.UtenteA=Utente.Id

V2(èAmico): èAmico.UtenteB=Utente.Id

V3(èAmico): èAmico.Data DEFAULT CURRENT_TIMESTAMP

PROGRAMMING LANGUAGES USED TO IMPLEMENT MY WEB APPLICATION

In order to develop my blog I used five programming languages.



First of all I defined the structure of the web application by using HyperText Markup Language generally known as **HTML**. This language is not bound to a particular platform, but it requires only the use of a browser.

It was developed in 1989 by Tim Berners-Lee along with HTTP protocol and it represents the layout of the content. Moreover, to make the

website better-looking I used Cascading Style Sheets, another programming language which is typically called with its acronym **CSS**. The proprieties were enacted by W3C from 1996 and I used this language for describing the presentation of document written in HTML. It allows programmers to define the color of writing, the background and so on.



Despite this, CSS cannot be used to manage the events triggered by users. An event can be generated by mouse, keyboard, loading page and so on and it is handled by a programming language developed by Brendan Eich in 1995 the name of which was Mocha. Today it is known as **JavaScript** and with CSS and HTML they represent the three core technologies of World Wide Web content production.

Nevertheless this, great part of the code is written in **PHP**. It was originally the acronym of Hypertext Preprocessor and it was developed by Rasmus Lerdorf in 1994. I used this language to make dynamic web pages.



MySQL

Last, but not least I used **SQL** to control the database. It is the acronym of Structured Query Language and it performs the functions of DDL, DML and DCL.

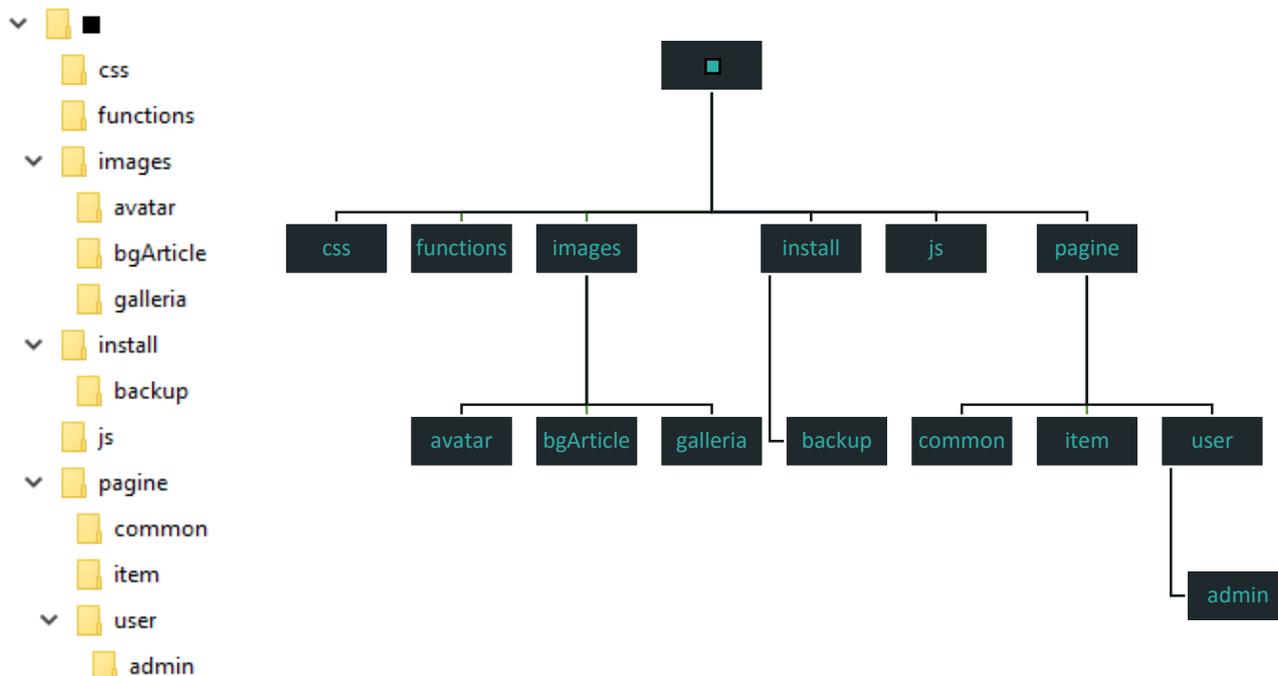
Data Definition Language defines the structure of the database like creating tables, views and so on. Whereas Data Manipulation Language provides instructions to manipulate data such as inserting, deleting and updating records from tables.

As far as security is concerned, the Data Control Language provides instructions to control accesses and user permissions.

In order to develop my blog I used SQL in embedded mode, thus the SQL instructions are within another programming language which in my case is PHP (Hypertext Preprocessor).

ORGANIZZAZIONE DELLA WEB APPLICATION

L'applicazione web è stata organizzata in cartelle. Ogni cartella raccoglie file che hanno caratteristiche comuni.



La directory principale è individuata dal ■ e contiene 6 sottocartelle:

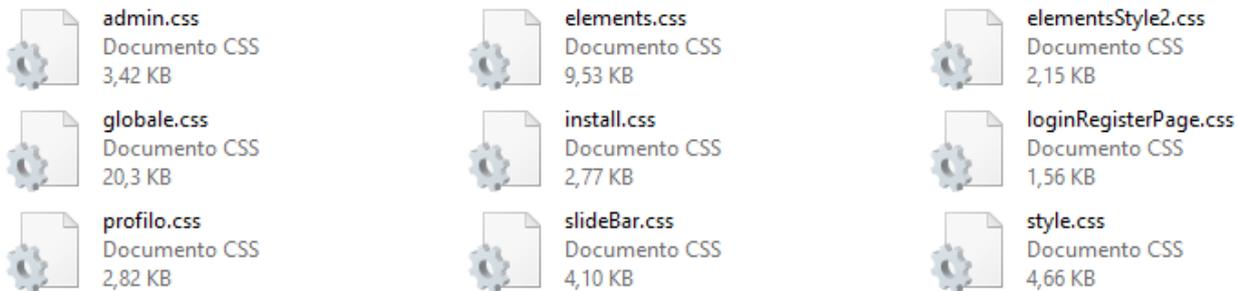
1. La cartella **css** contiene tutti i fogli di stile.
2. La cartella **functions** raccoglie tutte le pagine php che contengono le funzioni utilizzate nella realizzazione del blog. Inoltre include un file config.ini che memorizza alcune impostazioni del blog e del database.
3. La cartella **images** raccoglie immagini generali e non effettua una categorizzazione per estensioni, di conseguenza qualunque sia il tipo d'immagine (PNG, JPG, GIF, ...) sarà contenuto in questa cartella.

Come si può notare dal diagramma ad albero, images contiene a sua volta altre tre sottocartelle:

- a. **avatar**: include le immagini degli utenti iscritti al blog.
 - b. **bgArticle**: contiene tutte le immagini di copertina dei vari articoli del blog.
 - c. **galleria**: raccoglie le varie immagini che compongono la galleria.
4. La cartella **install** contiene il file index.php, utilizzato per avviare l'installazione del blog, e la sottocartella backup che raccoglie tutti i salvataggi del database.
 5. La cartella **js** viene utilizzata come raccoglitore dei file JavaScript.
 6. La cartella **pagine** contiene tutti i file che contengono la struttura delle pagine che compongono il blog. Inoltre, come si evince dal diagramma, include tre sottocartelle:
 - a. **common**: contiene parti della struttura che sono generalmente richieste da tutte, o quasi, le altre pagine.
 - b. **item**: raccoglie la struttura delle pagine che riguardano gli elementi principali del blog.
 - c. **user**: include la struttura delle pagine relative agli utenti del blog. Oltre a ciò contiene una sottocartella **admin** che comprende il pannello di amministrazione.

Cartella CSS:

I fogli di stile contenuti in questa cartella sono mostrati nella figura sottostante:



• FOGLIO ADMIN.CSS:

Contiene tutte le classi di stile utilizzate esclusivamente nel pannello di amministrazione:

Identificatore	Descrizione
section.adminSlide	Ridefinisce lo stile delle sezioni della slide bar già dichiarati in slideBar.css.
section.graph	Stile della sezione che contiene un grafico.
#icon	Stile del box che contiene le icone principali che possono essere assegnate ad una categoria.
.editor	Stile della lista dei bottoni che permettono di formattare il testo dell'articolo.

La lista riportata definisce solo gli stili principali

• FOGLIO ELEMENTS.CSS:

Raccoglie gli stili che definiscono l'aspetto grafico di alcuni oggetti presenti nel blog:

Identificatore	Descrizione
section.articles	Stile della sezione che contiene gli articoli.
section.articles>article	Stile del box principale che contiene un articolo.
section.info	Stile della sezione che contiene informazioni sull'articolo. (Nomi degli utenti che hanno apprezzato o salvato l'articolo, tag).
section.gallery	Stile del contenitore che raccoglie le foto che compongono la galleria.
section.gallery figure	Stile del box che contiene una singola immagine della galleria.
section.styleSection	Stile di una sezione generale: utilizzato per la rappresentazione grafica dei commenti, degli articoli salvati/apprezzati nel profilo utente, per la biografia...
#enterComment	Stile della sezione che contiene il form per inserire un commento.

La lista riportata definisce solo gli stili principali

• FOGLIO ELEMENTSSTYLE2.CSS:

Raccoglie gli stili che definiscono l'aspetto grafico di alcuni oggetti presenti nel blog:

Identificatore	Descrizione
section.description	Stile della descrizione di una categoria.
section.articlesStyle02>article	Stile dell'articolo con un aspetto grafico diverso da quello definito in elements.css.

La lista riportata definisce solo gli stili principali

• **FOGLIO GLOBALE.CSS:**

Contiene gli stili che definiscono l'aspetto globale del blog. Questo file viene importato in tutte le pagine del sito web:

Identificatore	Descrizione
body	Stile del corpo della pagina.
a	Stile dei collegamenti della pagina.
b	Stile dei caratteri in grassetto.
h1,h2,h3,h4,h5,h6	Stile delle intestazioni.
img	Stile delle immagini.
ul	Stile delle liste di elementi.
.clear	Stile che impedisce il posizionamento di altri elementi su entrambi i lati.
.left	Stile che sposta il box sul lato sinistro facendo scorrere il contenuto a destra.
.right	Stile che sposta il box sul lato destro facendo scorrere il contenuto a sinistra.
.mobile	Stile che nasconde elementi che devono essere mostrati solo su dispositivi mobili.
#loading-mask	Stile della maschera di caricamento della pagina.
#search	Stile della sezione che contiene il form per la ricerca di articoli/foto.
.maxSize	Stile che definisce le dimensioni massime dell'intestazione, corpo e footer della pagina.
#logo	Stile che definisce la sezione che contiene il logo del blog.
#panel	Stile del pannello login presente in tutte le pagine.
#userPanel	Stile del pannello utente presente in tutte le pagine.
body>nav	Stile del menu principale.
.menuFixed	Stile che fissa il menu alla pagina.
Main	Stile del corpo centrale (contenuto + slide bar).
#contenuto	Stile del box che racchiude il contenuto della pagina.
section	Stile generale delle sezioni.
body>footer	Stile della parte finale della pagina.
ul.tag	Stile della lista dei tag.
#stat	Stile della sezione statistiche.
#developer	Stile del box che contiene informazioni sullo sviluppatore (barra finale presente in tutte le pagine).
#fixedScreen	Stile utilizzato come sfondo quando viene aperta un'immagine.
#webMsg	Stile del contenitore che visualizza messaggi .
.print-message	Stile dei messaggi inseriti nel contenitore webMsg.
section.error	Stile del corpo centrale della pagina di errore.
.message	Stile di un messaggio generato dall'errore di un form (login errato, registrazione fallita, errore nel caricamento dell'avatar...).
#editForm	Stile del box utilizzato per modificare i commenti/la biografia degli utenti.

La lista riportata definisce solo gli stili principali

• FOGLIO INSTALL.CSS:

Questo file raccoglie gli stili che vengono utilizzati nella pagina d'installazione del blog:

Identificatore	Descrizione
body	Stile del corpo della pagina.
#error	Stile del messaggio di errore dovuto ad un'installazione incompleta.
#install	Stile del box che contiene il form per l'installazione.

La lista riportata definisce solo gli stili principali

• FOGLIO LOGINREGISTERPAGE.CSS:

Contiene gli stili utilizzati nelle pagine di login, registrazione e password perse. Lo stile principale è #loginBox, che definisce la sezione che contiene il form.

• FOGLIO PROFILO.CSS:

Contiene gli stili delle pagine che compongono il profilo e pannello utente:

Identificatore	Descrizione
section.styleSection	Stile che ridefinisce, con alcune proprietà diverse, la classe styleSection che rappresenta una sezione generale.
section.profile-user	Stile della sezione, contenuta nella slide bar, che contiene informazioni sull'utente.
section.profile-user>figure	Stile del box che contiene l'avatar dell'utente.
section.profile-user>#group	Stile del contenitore che visualizza il ruolo dell'utente.

La lista riportata definisce solo gli stili principali

• FOGLIO SLIDEBAR.CSS:

Contiene gli stili che definiscono graficamente le sezioni presenti nella slide bar:

Identificatore	Descrizione
aside	Stile della slide bar.
aside>section	Stile delle sezioni presenti nella barra di scorrimento.
section.sectionStyle-01	Stile alternativo delle sezioni.
section.navLink	Stile della sezione che contiene dei collegamenti di navigazione.
#statCategory	Stile della sezione che contiene le statistiche della categoria nella slide bar.

La lista riportata definisce solo gli stili principali

• FOGLIO STYLE.CSS:

Contiene gli stili che descrivono graficamente i tag HTML utilizzati nella stesura di un articolo:

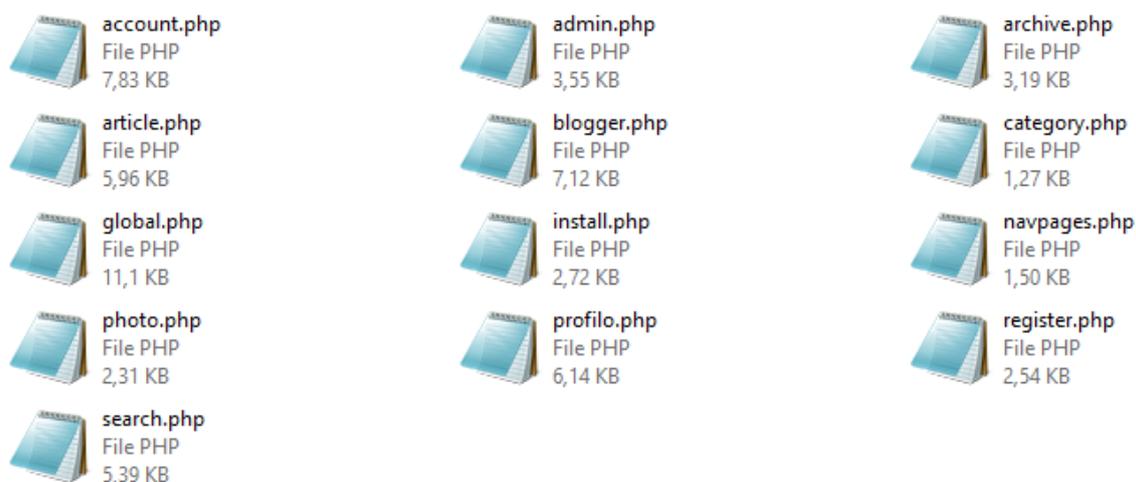
Identificatore	Descrizione
article>section.content>p	Stile del paragrafo all'interno di un articolo.
article>section.content>hX	Stile delle intestazioni presenti in un articolo.
article>section.content>img	Stile delle immagini nel contenuto di un articolo.
article>section.content>code	Stile del tag che riporta un pezzo di codice.
article>section.content>cite	Stile del tag utilizzato per inserire una citazione.

article>section.content>ul	Stile di una lista di elementi.
main form	Stile del form.
main form select	Stile campo di selezione.
main form input	Stile campo input.
main form label	Stile delle etichette.
main textarea	Stile del campo textarea.

La lista riportata definisce solo gli stili principali

Cartella Functions:

I file php contenuti in questa cartella sono mostrati nella figura sottostante:



• FILE ACCOUNT.PHP:

Raccoglie tutte le funzioni utilizzate per la gestione degli utenti e le azioni che possono effettuare:

Funzione	Descrizione
login()	Funzione che tenta di effettuare il login quando vengono impostati i parametri email e password. Se il tentativo termina con successo ritornano le informazioni sull'utente, altrimenti un errore indica il campo errato.
logout()	Funzione che disconnette l'utente connesso distruggendo la sessione.
profileOption(...)	Funzione in cui ritorna il codice html per modificare/cancellare un oggetto se l'utente loggato è colui che ha inserito l'elemento. (Link per modificare/cancellare un commento, modificare la biografia...)
userBoundTo(...)	Funzione che in base al valore ritornato dalla funzione existingBond(...) determina i parametri dell'url per creare un'associazione tra l'utente loggato e l'elemento passato come parametro.
existingBond(...)	Funzione che determina se nel database vi è un legame tra l'oggetto passato come parametro e l'utente loggato.
createBond(...)	Funzione che passato un elemento crea un legame con l'utente, se il legame è già esistente viene cancellato.
comment()	Funzione che inserisce un commento.
deleteComment()	Funzione che elimina un commento.
updateComment()	Funzione che modifica un commento già inserito.
pswLost(...)	Funzione che invia un email con la procedura per impostare una nuova password.

• FILE ADMIN.PHP:

Ogni qualvolta che il file admin.php viene incluso controlla che l'utente loggato abbia il ruolo di Amministratore. Inoltre il file contiene le seguenti funzioni:

Funzione	Descrizione
graph(...)	Funzione che passato il nome di una tabella ritorna i dati registrati negli ultimi 9 giorni. Da questi dati verranno poi stilati dei grafici.
role()	Funzione che inserisce un ruolo nel database.
allUsers()	Funzione che ritorna tutti gli utenti iscritti.
allRules()	Funzione che ritorna tutti i ruoli presenti nel database.
updateRole()	Funzione che aggiorna il ruolo di un utente iscritto.
checkCategory(...)	Funzione che determina se la categoria è già stata creata.
newCategory(...)	Funzione che aggiorna una categoria se è già stata creata, altrimenti la crea.

• FILE ARCHIVE.PHP:

Quando il file archive.php viene incluso automaticamente viene definito il tipo di archivio da mostrare: fotografico o testuale, di default viene scelto l'archivio degli articoli. Inoltre stabilisce la tipologia: mensile, annuale, su un periodo di tempo definito o settimanale che rappresenta l'impostazione di default.

Funzione	Descrizione
getArchive(...)	Funzione che ritorna un numero di elementi limitati e pubblicati durante un periodo di tempo definito.
allRecords(...)	Funzione che determina il totale degli elementi pubblicati durante un periodo di tempo definito.

• FILE ARTICLE.PHP:

Il file php contiene tutte le funzioni per la corretta visualizzazione degli articoli.

Funzione	Descrizione
print_article(...)	Funzione che stampa l'articolo passato come parametro.
print2_article(...)	Funzione che stampa l'articolo passato come parametro con un'interfaccia diversa dalla funzione print_article(...).
countComments(...)	Funzione che passato un identificatore ritorna il totale dei commenti legati all'articolo.
countSubscribers(...)	Funzione che passato un identificatore ritorna il totale degli utenti che hanno salvato l'articolo.
usersLink(...)	Funzione che ritorna la lista degli utenti che hanno salvato/apprezzano l'articolo.
postAuthor(...)	Funzione che passato un identificatore ritorna l'autore dell'articolo.
tagPost(...)	Funzione che passato un identificatore ritorna la lista dei tag relativi all'articolo.

• FILE **BLOGGER.PHP**:

Ogni qualvolta che il file blogger.php viene incluso controlla che l'utente loggato abbia il ruolo di Amministratore o Blogger. Inoltre il file contiene le seguenti funzioni:

Funzione	Descrizione
nameImg(...)	Funzione che sposta l'immagine nella cartella passata come parametro e ritorna il nome assegnato alla foto.
sendPhoto()	Funzione che carica una foto e inserisce l'identificatore nel database. Questa funzione viene utilizzata per caricare le immagini della galleria, ma anche per il caricamento delle copertine degli articoli.
deleteImage(...)	Funzione che elimina un'immagine della galleria sia dal database che dal file system storage.
imageFromTable(...)	Funzione che ricerca l'immagine passata nel database. Questa funzione serve per controllare la presenza del puntatore nella base di dati.
tag(...)	Funzione che crea un'associazione tra i tag passati e l'elemento (articolo/foto) inserito. I tag non presenti nel database verranno aggiunti alla tabella.
sendArticle()	Funzione che aggiorna un articolo se il suo identificatore è già presente nel database, altrimenti lo crea.
getTagArticle(...)	Funzione che passato un identificatore ritorna tutti i tag legati all'articolo sotto forma di stringa.
deleteArticle(...)	Funzione che elimina un articolo da database.

• FILE **CATEGORY.PHP**:

Il file raccoglie le funzioni utilizzate nella visualizzazione delle categorie.

Funzione	Descrizione
check (...)	Funzione che passato un identificatore determina l'esistenza della categoria
countRecords(...)	Funzione che calcola il numero di elementi legati alla categoria. I parametri della funzione determinano da quale tabella andare a prelevare i dati.

• FILE **GLOBAL.PHP**:

Il global.php è il primo file incluso in tutti i documenti. Lo script come prima cosa apre una sessione e definisce una costante che indica il percorso assoluto della directory principale. Inoltre controlla che config.ini sia presente per recuperare le informazioni relative al blog, la non presenza del file di configurazione comporta l'automatico caricamento della pagina d'installazione. Dopo aver effettuato i controlli relativi all'accesso e alle azioni dell'utente, vengono salvati tutti i parametri passati alla pagina in una variabile di sessione. Questa variabile verrà resettata ogni qualvolta che si carica una pagina del blog senza passare il parametro page, che indica il cambio della pagina, o action, che indica l'azione dell'utente.

Il global.php contiene le seguenti funzioni che possono essere richiamate in ogni pagina del blog:

Funzione	Descrizione
apriConnessione()	Funzione che effettua la connessione con il database.
terminaConnessione()	Funzione che termina la connessione con il database.
autore()	Funzione che ritorna informazioni sull'autore del blog.
statBlog()	Funzione che ritorna le statistiche del blog.
categorie(...)	Funzione che ritorna un numero di categorie limitato dal parametro passato alla funzione.

getPosts(...)	Funzione che ritorna un numero di articoli che variano a seconda dei parametri passati alla funzione.
getComments(...)	Funzione che ritorna i commenti inseriti nel blog secondo i parametri passati alla funzione.
countLikes(...)	Funzione che dato un identificatore conta il numero di "Mi Piace" inseriti dagli utenti.
getItemCategories(...)	Funzione che passato un identificatore ritorna le categorie dell'oggetto.
itemType(...)	Funzione che ritorna dati necessari per adattare le query alle tabelle degli articoli/delle foto.
message()	Funzione che richiama uno script javascript per stampare un messaggio.
loadImg(...)	Funzione che carica un'immagine e gli assegna il nome passato come parametro alla funzione.
moveImg(...)	Funzione che salva l'immagine passata come parametro nella cartella indicata e ritorna l'estensione.
allCategories()	Funzione che ritorna tutte le categorie del blog.
getTag(...)	Funzione che ritorna i tag più usati. Il numero dei record viene limitato dal parametro passato alla funzione.

• FILE INSTALL.PHP:

Le funzioni contenute in questo file vengono utilizzate per l'installazione del blog. Questo script controlla automaticamente l'esistenza del file config.ini, se il file esiste l'utente verrà reindirizzato alla homepage del blog in quanto l'installazione non può avvenire.

Funzione	Descrizione
getTables(...)	Funzione che ottiene le query per importare le tabelle. A seconda della scelta dell'utente viene importato un backup pulito o l'ultimo effettuato.
createDB(...)	Funzione che crea il database, importa le tabelle e crea il file config.ini.

• FILE NAVPAGES.PHP:

Questo file viene importato da tutti quegli script in cui il numero di elementi da visualizzare è elevato e per questo motivo vengono suddivisi in pagine.

Funzione	Descrizione
navPages(...)	Funzione che determina e ritorna la pagina corrente e il numero delle pagine totali.
print_navigation(...)	Funzione che stampa il menu di navigazione delle pagine.

• FILE PHOTO.PHP:

Questo file contiene le funzioni per recuperare e stampare le foto.

Funzione	Descrizione
photo(...)	Funzione che stampa l'immagine passata come parametro alla funzione.
getPhotos(...)	Funzione che passati determinati parametri ritorna delle informazioni dal database relative alle immagini.

• FILE PROFILO.PHP:

Le funzioni contenute in questo file vengono utilizzate nelle pagine relative al profilo degli utenti.

Funzione	Descrizione
userProfile(...)	Funzione che passato un identificatore ritorna le informazioni su un utente.
getUserPosts(...)	Funzione che ritorna gli articoli che sono piaciuti o salvati dall'utente passato come parametro.
countUserPosts(...)	Funzione che ritorna il numero degli articoli che sono piaciuti o salvati o commentati dall'utente passato come parametro alla funzione.
<i>Le funzioni di seguito descritte sono relative alla modifica del profilo dell'utente loggato</i>	
updateBio()	Funzione che aggiorna la biografia dell'utente.
updatePsw()	Funzione che aggiorna la password dell'utente.
updateEmail()	Funzione che aggiorna l'email dell'utente.
updateSetting()	Funzione che aggiorna le impostazioni dell'utente.

• FILE REGISTER.PHP:

All'interno del file è presente un unico script che permette di registrare un utente nel database. Prima di effettuare la registrazione verranno controllati che tutti i dati immessi siano corretti e al termine dell'operazione l'utente sarà automaticamente reindirizzato al suo profilo utente. In caso di errore verrà ricaricata la pagina di registrazione indicando quale campo non rispetta i requisiti necessari.

• FILE SEARCH.PHP:

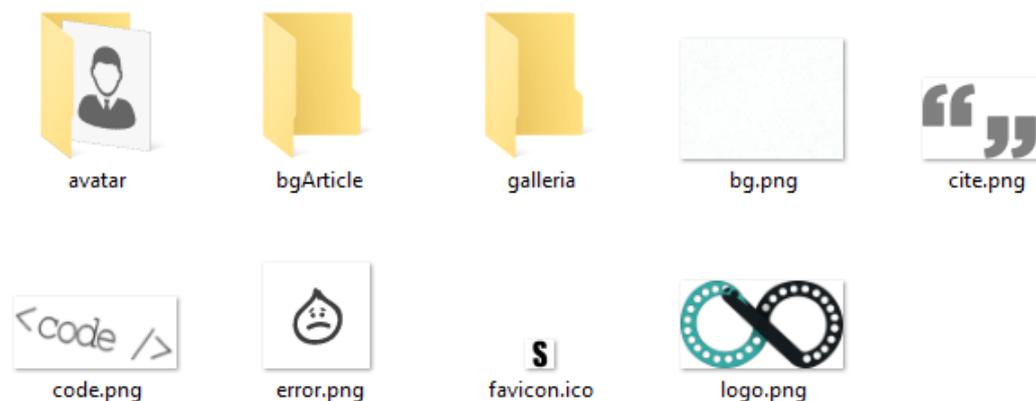
Il file comprende le funzioni necessarie a creare la query per la ricerca degli articoli/foto. La query di ricerca viene creata da una serie di funzioni che, passati determinati parametri, costruiscono query diverse.

Funzione	Descrizione
getItem(...)	Funzione che ritorna dei valori utilizzati per adattare la query nella ricerca di articoli o foto.
searchItems(...)	Funzione che passati determinati parametri ritorna il risultato della query di ricerca.
createQuery(...)	Funzione che passati determinati parametri costruisce la query di ricerca.
countItems(...)	Funzione che conta il totale degli elementi che ritornano dalla query di ricerca.
listKeys(...)	Funzione che ritorna la stringa di parole chiave, passate come parametro alla funzione, sotto forma di query.

Cartella Images:

La cartella **images** è composta da cinque immagini(png), una icona(ico) e tre sottocartelle.

La figura sottostante mostra il contenuto della cartella images:



Le immagini *bg.png* e *logo.png* vengono utilizzate nel header di ogni pagina del blog, al contrario *cite.png* e *code.png* vengono adoperate quando si utilizzano i rispettivi tag HTML `<cite>` e `<code>`.

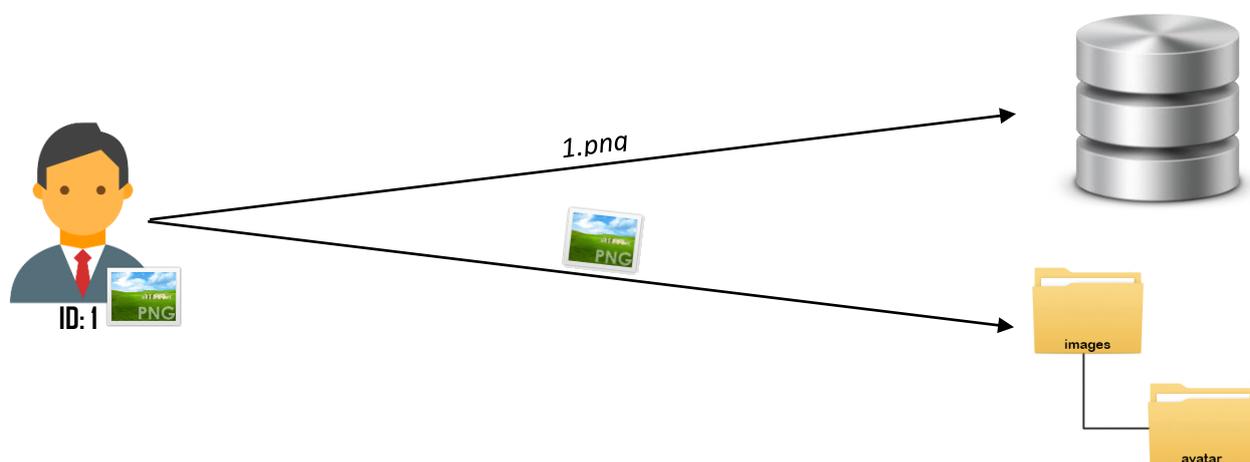
La figura *error.png* viene impiegata nella pagina di errore quando la risorsa non viene trovata, invece la *favicon.ico* indica l'icona associata al blog visualizzata alla sinistra dell'URL nella barra degli indirizzi del browser.

La cartella **avatar** contiene le immagine caricate dagli utenti del blog. La persona iscritta può caricare il proprio avatar al momento della registrazione o attraverso il form presente nel pannello di controllo utente. Una volta scelta l'immagine questa verrà inserita automaticamente nella cartella, con il nome uguale all'identificatore dell'utente, e nel database verrà registrato il puntatore alla figura caricata.

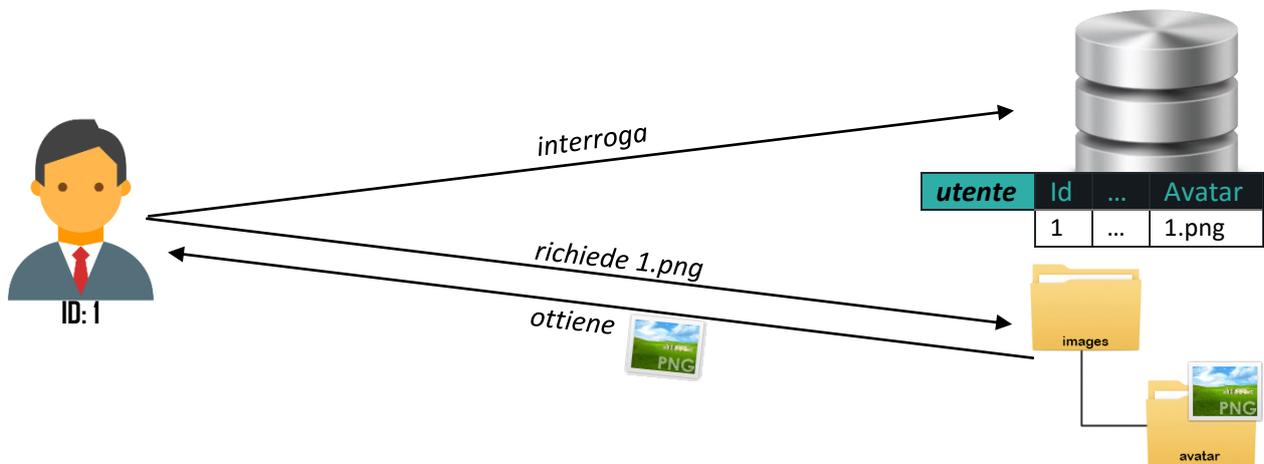
La cartella **bgArticle** funziona allo stesso modo della cartella avatar, ma all'interno verranno inserite le copertine degli articoli registrando nel database il puntatore alla figura che manterrà il nome originale. Analogamente l'ultima sottocartella contiene le foto che compongono la galleria.

Di default **bgArticle** e **galleria** sono cartelle vuote, contrariamente alla directory avatar che include l'immagine *default.jpg* a cui puntano tutti gli utenti iscritti che non hanno caricato una propria foto personale.

• UTENTE CARICA L'AVATAR

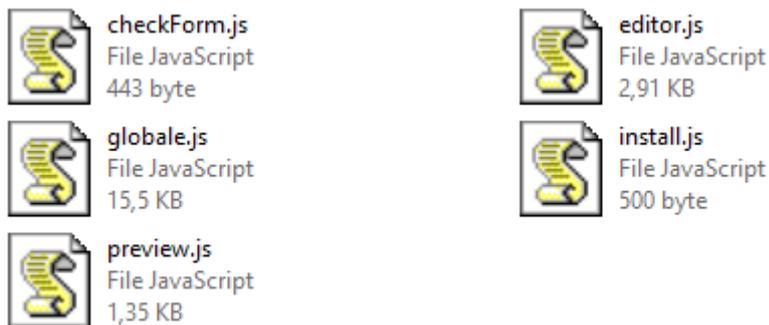


• **UTENTE RECUPERA L'AVATAR**



Cartella Js:

Il contenuto della cartella *js*, come si può vedere dall'immagine sottostante, è di cinque file JavaScript.



• **FILE CHECKFORM.JS:**

Il contenuto di questo file è una sola funzione che prende il nome di *check*. La funzione richiede l'identificatore di due campi di input e se il contenuto è uguale, allora il testo viene colorato di verde, altrimenti di rosso.

• **FILE EDITOR.JS:**

Il file raccoglie le funzioni utilizzate dall'editor di testo presente nella pagina di creazione di un articolo.

Funzione	Descrizione
editorTesto(...)	Funzione che applica il tag HTML passato come parametro al testo selezionato e lo inserisce nell'area di testo.
list(...)	Funzione che apre una serie di finestre di dialogo per permettere all'utente di creare una lista. La lista verrà automaticamente inserita nel contenuto dell'area di testo e il parametro passato identifica una lista puntata o numerata.
tagHeader()	Funzione che apre il menu che permette di decidere quale dei tag header utilizzare. Ogni singolo tag header richiamerà la funzione editorTesto(...) per essere applicato.
insertImg(...)	Funzione che apre una finestra di dialogo in cui chiede di inserire il percorso dell'immagine.

• FILE GLOBALE.JS:

Il file contiene una serie di funzioni che vengono utilizzate nella maggior parte delle pagine del blog.

Funzione	Descrizione
editForm(...)	Funzione che costruisce il form che permette di modificare i commenti e la biografia dell'utente. La funzione richiede dei parametri per identificare l'oggetto da modificare.
loginBox(...)	Funzione che apre il box per il login e permette di trasformare il form in recupera password. La funzione richiede un parametro che identifica l'azione effettuata dall'utente.
userPanel(...)	Funzione che apre e chiude il box che contiene il menu di navigazione utente.
loadMsg(...)	Funzione che apre e visualizza i messaggi. I parametri della funzione definiscono il colore e il testo del messaggio.
removeMsg()	Funzione che viene richiamata per chiudere un messaggio dopo 5000ms che è stato aperto.
sizeImg(...)	Funzione che apre e chiude ogni immagine presente nel blog. Al caricamento della pagina per ogni immagine viene aggiunto un ascoltatore "onclick" che richiama questa funzione.
controlSlide(...)	Funzione che permette di aprire o chiudere la slide bar sulla destra del blog.
footerBox(...)	Funzione che chiude le sezioni presenti nel footer della pagina. Il parametro passato serve per identificare la sezione su cui applicare la funzione.
scrollBar()	Funzione che controlla la distanza del menu dalla parte superiore per fissarlo alla pagina.
apriMenu(...)	Funzione che serve ad aprire il menu quando il blog viene visualizzato in modalità mobile.
setMinHeightContainer()	Funzione che determina la dimensione minima del corpo della pagina quando la slide bar ha un'altezza inferiore.
btnSearch()	Funzione che apre e chiude il form di ricerca.
headerBg(...)	Funzione che permette il movimento dello sfondo dell'intestazione delle pagine quando l'utente muove il mouse.

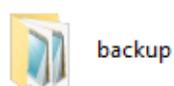
• FILE INSTALL.JS:

Questo file contiene una funzione JavaScript utilizzata dalla pagina d'installazione che permette di adattare i campi del form secondo il tipo d'installazione selezionata. Il nome della funzione è *typeOfInstall(...)* e richiede un parametro di tipo booleano che indica se è richiesto di importare un backup con dati o vuoto.

• FILE PREVIEW.JS:

Il file contiene un'unica funzione JavaScript che permette di avere un'anteprima del commento che si vuole inserire. Il nome della funzione è *anteprimaCommento(...)* e richiede come parametro il form con cui il commento viene inserito.

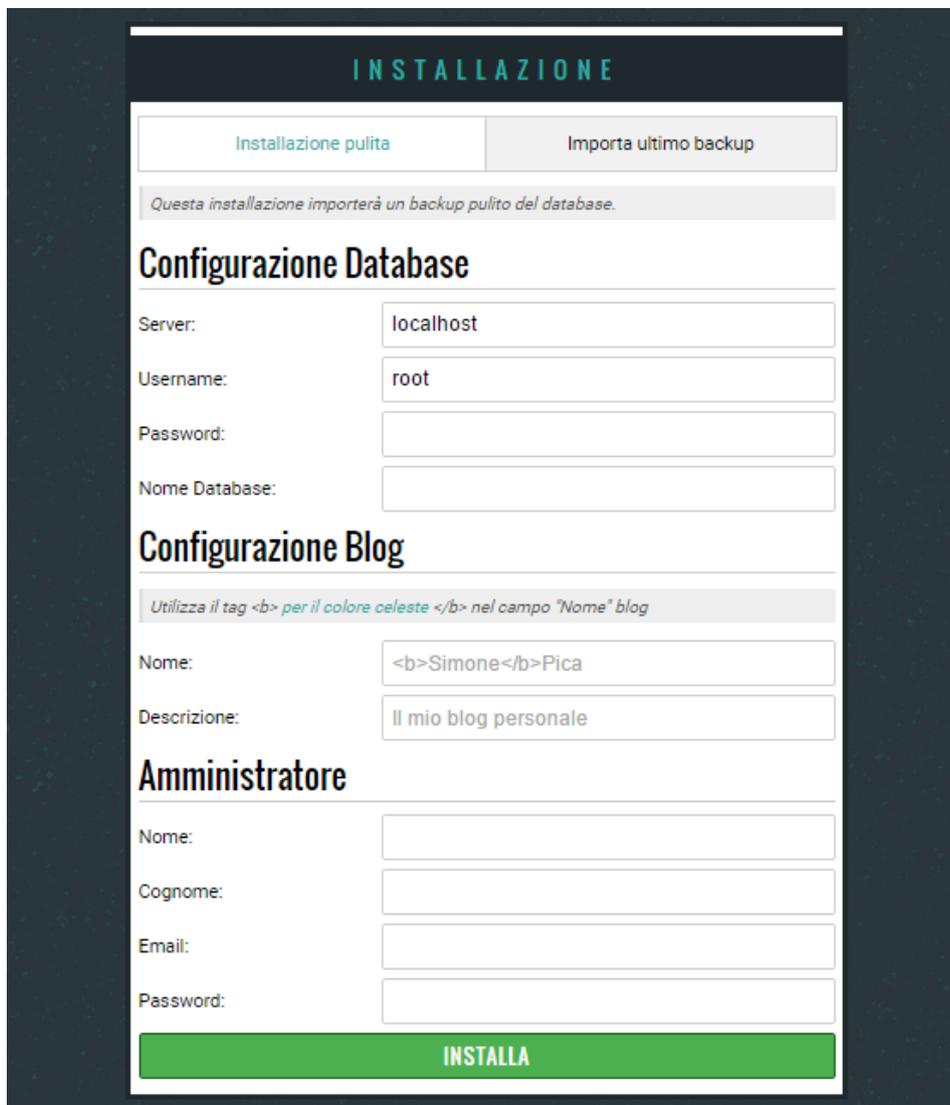
Cartella Install:



La cartella **install** è composta da una sottocartella **backup**, che contiene tutti i salvataggi del database, e da **index.php** che permette di avviare la procedura d'installazione.

• FILE INDEX.PHP:

La pagina d'installazione è composta da un unico box che contiene un form in cui si devono immettere i dati richiesti per importare le tabelle del database e avviare il blog.



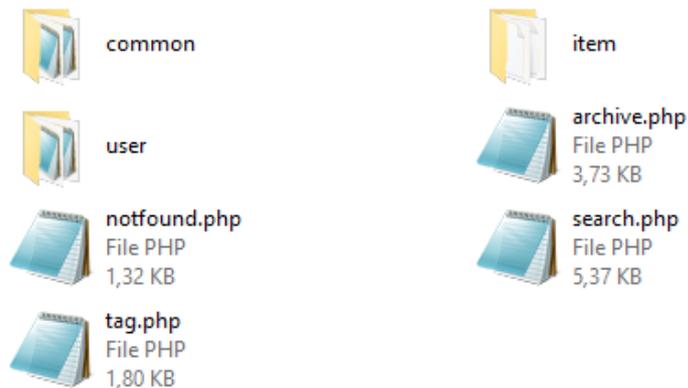
Il form è composto da un campo radio, che permette di decidere quale tipo di backup importare, e da una serie di campi input per configurare il database, il blog e creare il primo utente registrato.

CSS	PHP	JAVASCRIPT
install.css	install.php(DIR: functions)	install.js
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	getTables()	typeOfInstall(...)
	createDB()	

Di seguito verrà presentata la cartella pagine che contiene tutte le strutture che compongono il blog.

PAGINE DEL BLOG

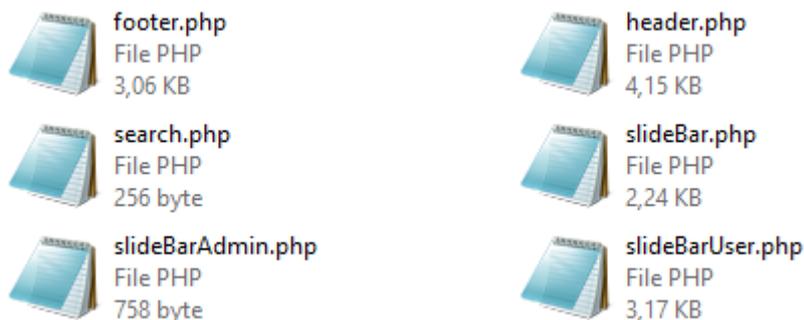
La foto mostra il contenuto della cartella **pagine**. Come si può vedere sono presenti quattro file php e tre sottocartelle.



Per una migliore organizzazione verranno descritti prima i file presenti in **common**.

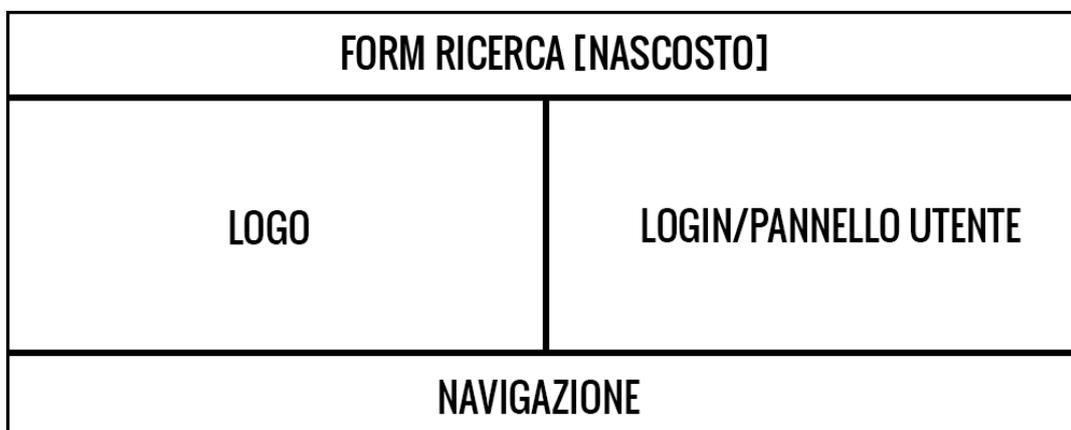
Cartella Common:

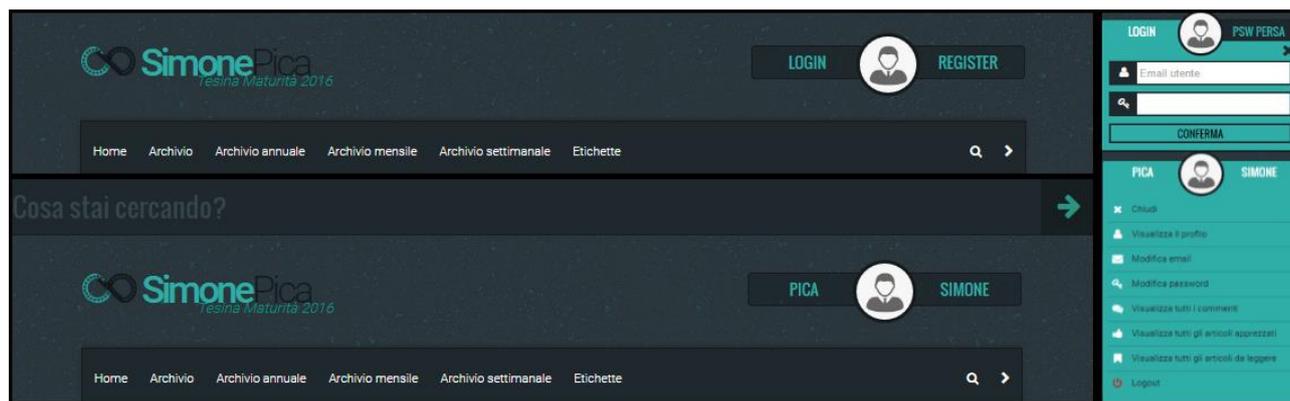
La cartella **common** contiene pezzi di codice che sono richiesti in tutte le pagine. I file presenti in questa cartella non possono essere avviati direttamente, ma devono essere inclusi nelle pagine del blog.



• FILE HEADER.PHP:

Il file *header* rappresenta l'intestazione di tutte le pagine del blog ed è composto dal box di ricerca, uno spazio per il logo, la sezione per gli utenti (Login/Pannello Utente) e il menu di navigazione.





In riferimento alla foto, all'estremità sinistra, possiamo notare il nome del blog e la descrizione che vengono inseriti al momento dell'installazione e recuperati dal file config.ini presente nella cartella **functions**.

Sul lato opposto sono presenti due bottoni: *Login e Register*. Quando la sezione d'accesso viene aperta, il bottone utilizzato per accedere alla pagina di registrazione, cambia la sua funzione in password persa. Inoltre tale sezione, in seguito all'autenticazione, viene sostituita dal pannello di controllo utente attraverso il quale si può accedere a pagine riservate od effettuare il *logout*.

Il menu di navigazione si compone di due tipologie di collegamenti:

1. Collegamenti per accedere ad altre pagine come agli archivi che catalogano gli articoli secondo periodi di tempo definiti.
2. Collegamenti che attivano funzioni JavaScript rappresentati sotto forma di icone:
 - a. La prima icona permette di aprire/chiedere il form di ricerca.
 - b. La seconda icona permette di aprire/chiedere la slide bar. Non tutte le pagine contengono la barra laterale, per questo motivo, appena la pagina viene caricata, lo script JavaScript ne controlla l'esistenza ed in caso di assenza elimina il bottone che ne permette la gestione.

CSS	PHP	JAVASCRIPT
Fogli importati dalla pagina che include header.php	Funzioni importate dalle pagine che includono l'header	Funzioni importate dalle pagine che includono l'header
	search.php (DIR: Common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	/	controlSlide(...)
		btnSearch()
	apriMenu(...)	
	loginBox(...)	
	userPanel(...)	

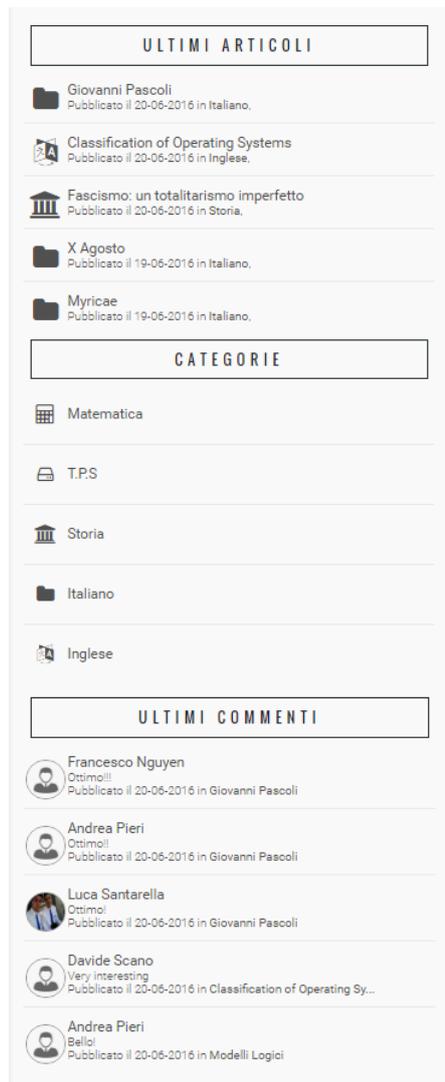
• FILE SEARCH.PHP:



Questo file contiene la struttura del form per la ricerca degli articoli. Il file viene incluso esclusivamente dal *header.php* (DIR: common) per completarne la struttura. Questo modulo permette di ricercare gli articoli sulla base delle parole digitate. Cliccando su conferma, la freccia all'estremità destra, verrà caricata la pagina search.php in cui sarà possibile, grazie agli strumenti messi a disposizione, ottimizzare i criteri di restrizione.

• FILE SLIDEBAR.PHP:

Il file definisce la struttura della barra laterale sulla parte destra delle pagine che compongono il blog.



La *slide bar* si compone di tre sezioni:

1. Ultimi cinque articoli inseriti nel blog.
2. Visualizzazione di cinque categorie presenti nel blog scelte in maniera casuale.
3. Ultimi cinque commenti inseriti nel blog.

Questa slide bar viene mostrata nella maggior parte delle pagine del blog ed è possibile chiuderla attraverso il bottone presente nel menu di navigazione.



CSS	PHP	JAVASCRIPT
Fogli importati dalla pagina che include slideBar.php	Funzioni importate dalle pagine che includono la slideBar	Funzioni importate dalle pagine che includono la slideBar
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	getPosts(...)	/
	getItemCategories(...)	
	categorie(...)	
	getComments(...)	

• FILE SLIDEBARUSER.PHP:



Questo file definisce un'ulteriore struttura della barra laterale che viene importata da tutte quelle pagine che compongono il profilo e il pannello degli utenti.

Anche in questo caso la slide bar si compone di 3 sezioni:

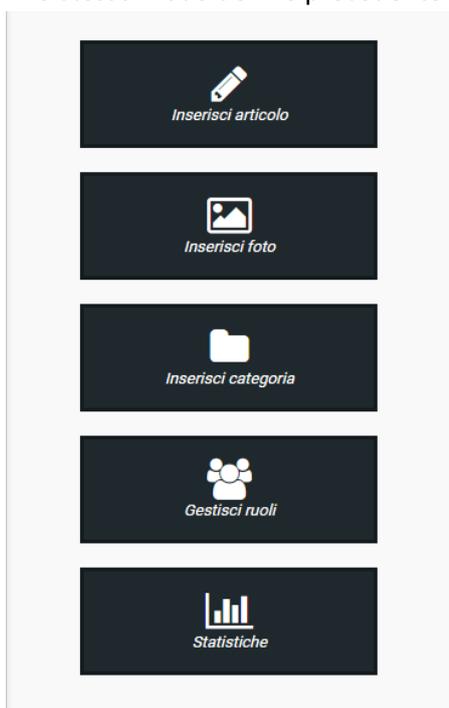
1. Specifiche e dettagli sull'utente.
2. La sezione contiene dei collegamenti per ricercare gli articoli legati all'utente. E' possibile cercare tutti gli articoli apprezzati e salvati. Inoltre un collegamento permette di trovare tutti i commenti pubblicati dall'utente.
3. L'ultima sezione viene visualizzata quando l'utente sta visualizzando il suo profilo in quanto rappresenta dei collegamenti a pagine che permettono di modificare le impostazioni dell'account. I link "Inserisci articolo" e "Inserisci foto" vengono mostrati se il ruolo dell'utente è "Blogger" od "Amministratore", al contrario "Pannello Amministratore" viene visualizzato solamente dall'utente "Amministratore".

Analogamente alla slide bar precedente, anche questa può essere chiusa mediante il bottone presente nel menu di navigazione.

CSS	PHP	JAVASCRIPT
Fogli importati dalla pagina che include slideBarUser.php	Importati dalle pagine che includono la slideBarUser	Importati dalle pagine che includono la slideBarUser
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	/	/

• FILE SLIDEBARADMIN.PHP:

Allo stesso modo del file precedente anche questo costituisce la struttura della slide bar. Il file viene incluso dalle pagine che rappresentano il pannello di amministrazione.



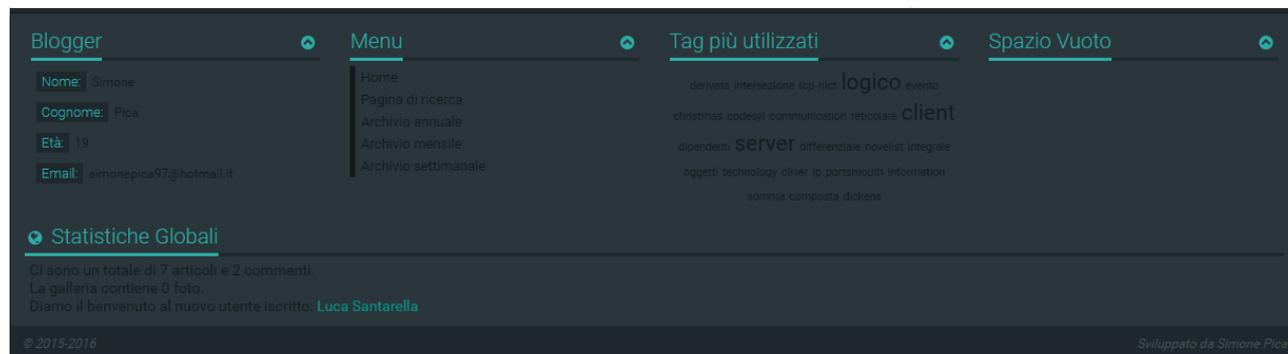
La struttura si compone di una semplice lista di elementi che permettono di accedere ad una serie di pagine riservate.

I primi due collegamenti sono visibili anche agli utenti che svolgono il ruolo di "Blogger", contrariamente agli ultimi tre che sono visibili ed accessibili solamente all'amministratore del blog.

CSS	PHP	JAVASCRIPT
Fogli importati dalla pagina che include slideBarAdmin.php	Funzioni importate dalle pagine che includono la slideBarAdmin	Funzioni importate dalle pagine che includono la slideBarAdmin
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	/	/

• FILE FOOTER.PHP:

Il footer è la parte finale delle pagine del blog e come l'intestazione viene incluso in tutti gli script che definiscono la struttura.



Come mostra la figura, il contenitore si compone di cinque sezioni con l'aggiunta di una barra finale. Le prime quattro sezioni possono essere chiuse/aperte mediante uno script JavaScript.

La lista sottostante definisce il contenuto dei vari blocchi:

1. Mostra informazioni sull'autore del blog (Nome, Cognome, Età, Email).
2. E' un menu di navigazione a fondo pagina.
3. Rappresenta i tag più utilizzati nel blog.
4. Spazio da riempire per scopi futuri.

Nell'ultima sezione vengono mostrate le statistiche del blog da cui si evince: il numero degli articoli, dei commenti e delle foto totali e l'ultimo utente iscritto. Inoltre la barra finale, contiene il nome dello sviluppatore del blog.

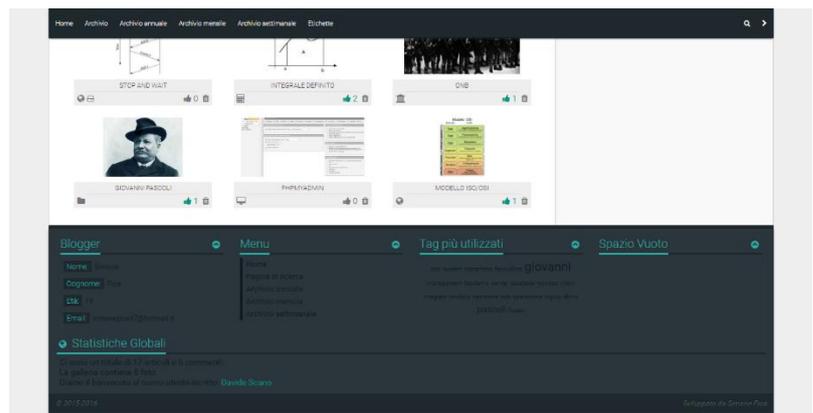
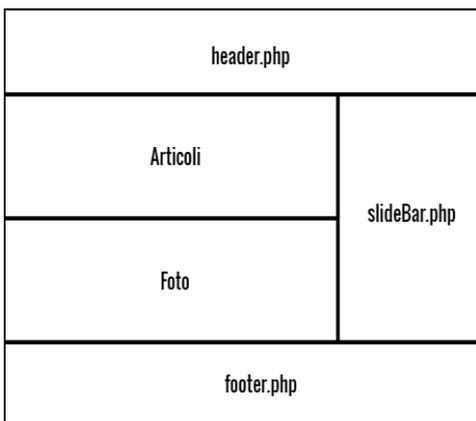
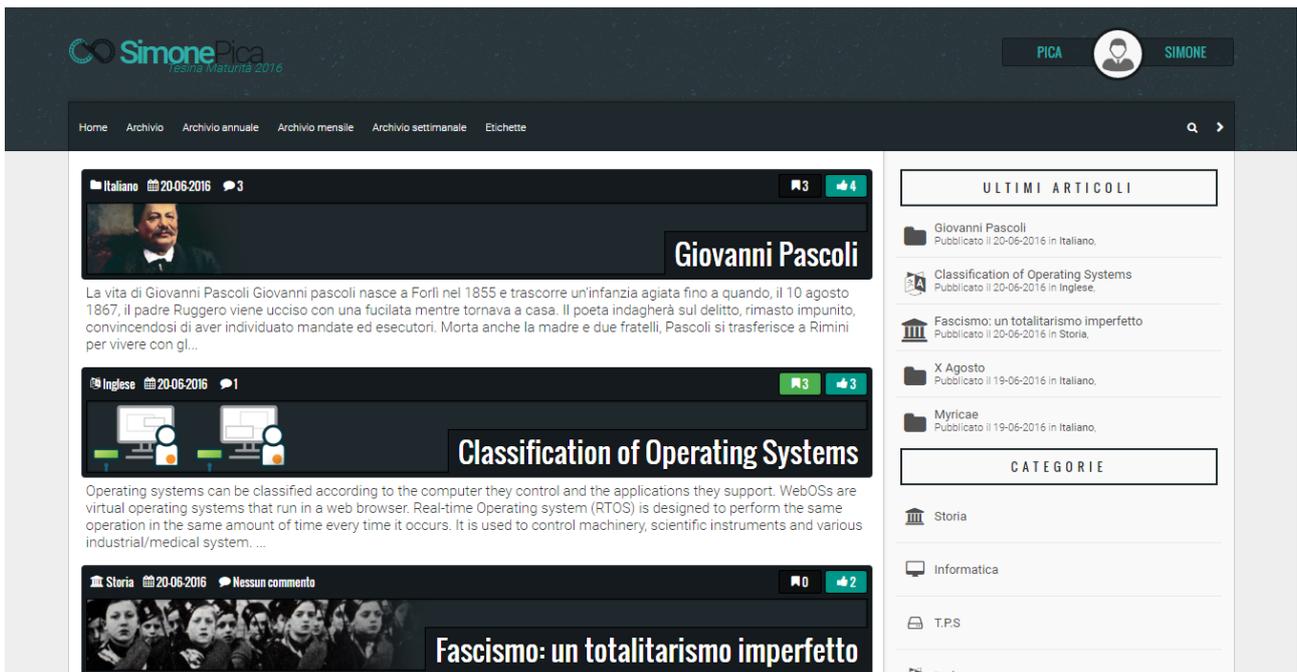
CSS	PHP	JAVASCRIPT
Fogli importati dalla pagina che include footer.php	Funzioni importate dalle pagine che includono il footer	Funzioni importate dalle pagine che includono il footer
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	autore()	footerBox(...)
	getTag(...)	
	message()	
	terminaConnessione()	

Prima di procedere con la descrizione dei file presenti nella directory pagine, verrà descritta la homepage del blog.

• HOMEPAGE - FILE INDEX.PHP:

Il file *index.php*, che rappresenta la homepage, si trova nella directory principale ed è il primo script che viene avviato quando si visita il blog.

Includendo i file precedentemente descritti la struttura finale di tale pagina è la seguente:



Il contenuto della homepage è suddiviso in due sezioni. La prima sezione mostra gli ultimi cinque articoli pubblicati, al contrario della sezione sottostante che visualizza le ultime sei foto inserite nella galleria. Il contenuto degli articoli è parziale e per ottenere il testo completo sarà necessario effettuare un click sul titolo del post desiderato.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	article.php(DIR: functions)	
elements.css	photo.php(DIR: functions)	
	header.php(DIR: common)	
	slideBar.php(DIR: common)	
	Footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	getPosts(...)	/
	getPhotos(...)	

	print_article(...)	
	photo(...)	

Cartella Item:

La cartella contiene la struttura dei principali elementi che compongono il blog come le categorie, gli articoli e i commenti.



• FILE ARTICLE.PHP:

Lo script definisce la struttura per visualizzare completamente un articolo. Inoltre permette di mostrare cinque commenti alla volta, inserire un commento ed avere delle informazioni sull'articolo che viene visualizzato. Richiamando lo script con un identificatore errato, o inesistente, verrà automaticamente caricata la pagina *notfound.php* che visualizza un messaggio d'errore. Questo perché nel database non risulta nessuna corrispondenza tra il parametro e i record memorizzati. Contrariamente, se il codice viene eseguito correttamente, verrà stampata la seguente pagina:

Nella parte superiore del contenuto abbiamo l'intestazione dell'articolo in cui è possibile notare:

1. Le categorie dell'articolo: cliccando sopra il nome della sezione verrà richiamato lo script *category.php*.
2. La data di pubblicazione: premendo il collegamento verrà effettuata una ricerca sugli articoli pubblicati in quel giorno.
3. Il numero di commenti: cliccandoci si accederà alla pagina *comment.php*.
4. Il numero di utenti che hanno deciso di salvare questo articolo nel proprio profilo.
5. Il numero totale di "Mi piace" inseriti degli utenti.
6. Il titolo.

Un utente loggato può decidere di salvare/apprezzare un articolo premendo sulle relative icone o negare l'azione effettuando nuovamente la medesima operazione.

Sotto l'intestazione abbiamo l'intero contenuto dell'articolo a cui segue l'autore e le informazioni. Nelle informazioni vengono stampati i nomi degli utenti che hanno salvato/apprezzato l'articolo e tutti i tag con cui l'articolo è associato. Cliccando sui singoli tag verrà effettuata una ricerca a tutti i post associati alla parola, contrariamente, cliccando sul nome degli utenti, otterremo il loro profilo utente.

Scendendo ancora verso il fondo pagina troviamo la sezione che permette di inserire un commento. Nella foto non essendoci nessun utente loggato, tale sezione viene sostituita da un messaggio che invita il visitatore ad effettuare il login oppure a registrarsi.



The screenshot shows a comment input form for a user named 'PICA SIMONE'. At the top right, there is a link that says 'Inserisci un commento'. Below the user's name is a large text input area with a placeholder 'Inserisci un commento'. At the bottom left of the form, there are two buttons: 'Invia' (with a checkmark icon) and 'Anteprima' (with an eye icon). At the bottom right, there is another link that says 'Inserisci un commento'.

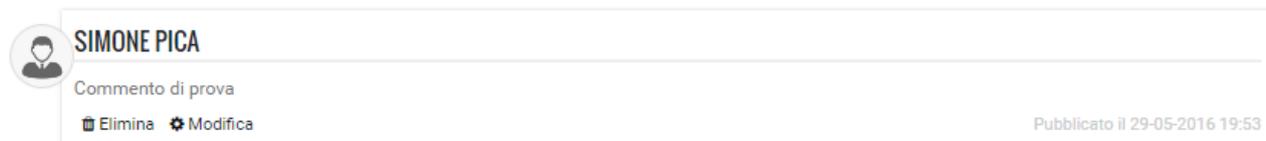
Commenti

La figura soprastante mostra come un utente loggato può inserire un commento.

Una volta scritto il contenuto si può decidere di ottenere un'anteprima, cliccando sul relativo pulsante, prima di inviare il commento.

Nella parte inferiore abbiamo i commenti già inseriti che verranno visualizzati per un massimo di cinque alla volta. Se l'articolo contiene più considerazioni queste verranno suddivise in pagine.

Quando un utente inserisce un proprio parere può decidere di modificarlo o cancellarlo in qualunque momento. Ciascun utente può ritrovare i propri commenti cercandoli negli articoli o attraverso l'apposito collegamento presente nel profilo utente.



The screenshot shows a comment card for a user named 'SIMONE PICA'. The comment text is 'Commento di prova'. Below the text, there are two buttons: 'Elimina' (with a trash can icon) and 'Modifica' (with a gear icon). At the bottom right, there is a timestamp: 'Pubblicato il 29-05-2016 19:53'.

MODIFICA COMMENTO

Commento di prova

AGGIORNA



The screenshot shows a comment card for a user named 'SIMONE PICA' in a list view. The comment text is 'Commento di prova'. At the bottom right, there is a timestamp: 'Pubblicato il 29-05-2016 19:53'.

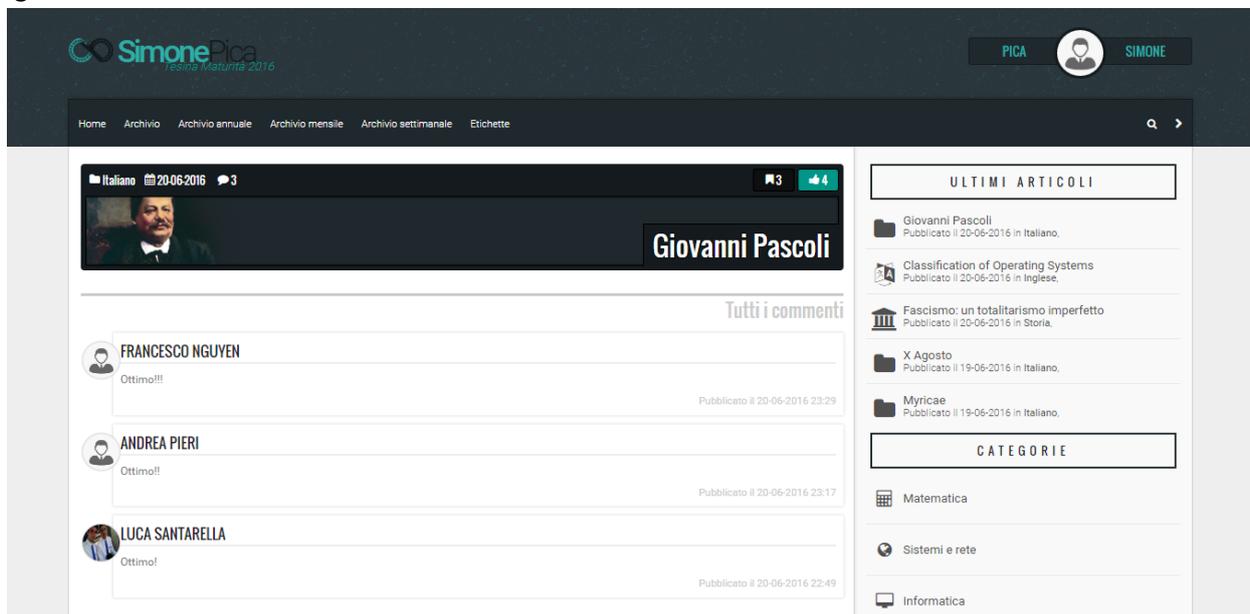


La pagina article.php aggiunge nella slide bar una sezione visibile solo agli utente che hanno il ruolo di "Blogger" o "Amministratore" e che contiene i collegamenti per cancellare e modificare il post visualizzato.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	article.php(DIR: functions)	preview.js
elements.css	navpages.php(DIR: functions)	
style.css	header.php(DIR: common)	
	slideBar.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	getPosts(...)	anteprimaCommento(...)
	deleteComment()	
	updateComment()	
	usersLink(...)	
	tagPost(...)	
	postAuthor(...)	
	navPages(...)	
	print_article(...)	
	getComments(...)	
	print_navigation(...)	

• FILE COMMENTS.PHP:

Il file contiene la struttura che permette di visualizzare i commenti di un articolo. La visualizzazione avviene in ordine cronologico, dal più recente al più remoto e saranno mostrati un massimo di dieci commenti per pagina.



La pagina mantiene l'intestazione dell'articolo presente in *article.php*, ma ne nasconde il contenuto che viene sostituito dalla sezione utilizzata per stampare i pareri degli utenti.

Quando il numero totale dei commenti è superiore a dieci verrà mostrato un menu che consente di visualizzare le altre considerazioni.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	article.php(DIR: functions)	
elements.css	navpages.php(DIR: functions)	
style.css	header.php(DIR: common)	
	slideBar.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	getPosts(...)	/
	deleteComment()	
	updateComment()	
	navPages(...)	
	print_article(...)	
	getComments(...)	
	print_navigation(...)	

• FILE CATEGORY.PHP:

Category.php definisce la struttura della pagina per visualizzare i dettagli della categoria. Come la homepage questo file contiene sia articoli che foto.

The screenshot shows the website's interface for the 'Matematica' category. The header includes the site logo 'Simone Pica' and navigation links like 'Home', 'Archivio', and 'Etichette'. The main content area is divided into two sections: 'Ultimi articoli' (Recent articles) and 'Ultime foto' (Recent photos). The 'Ultimi articoli' section lists three articles with titles like 'INTEGRALI IMMEDIATI DELLE FUNZIONI FONDAMENTALI' and 'INTEGRALE DEFINITO E DIFFERENZIALE'. The 'Ultime foto' section shows a grid of image thumbnails. On the right sidebar, there are five green buttons for 'Articoli', 'Galleria', 'Commenti', 'Mi Piace', and 'Articoli salvati', each with a small icon and a percentage indicating the number of items in that category.

Nella parte centrale della pagina possiamo notare due sezioni:

1. Contiene gli ultimi cinque articoli inseriti nella categoria.
2. Contiene le ultime sei foto inserite nella categoria.

Differentemente dalla homepage gli articoli assumono un aspetto più lineare in cui non viene visualizzata l'immagine di copertina.

INTEGRALE DEFINITO E DIFFERENZIALE

19-05-2016

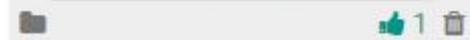
Significato geometrico dell'integrale definito Presa $y=f(x)$ continua in un intervallo $[a,b]$ rappresenta l'area della regione di piano compresa tra la funzione, l'asse delle ascisse e le due rette passanti per gli estremi a e b ed individuata da un numero. Differenziale di una funzione Si chiama differenziale di una funzione $y=f(x)$ relativo al punto x e all'incremento dx , il prodotto della de...

0 0 0 NESSUN COMMENTO

La foto soprastante rappresenta l'articolo. Nella parte superiore è possibile notare la presenza del titolo e della data, seguita da una porzione del contenuto. Nella parte inferiore vengono visualizzati dei collegamenti che rappresentano il numero di utenti che hanno apprezzato, salvato e commentato l'articolo. Inoltre nell'estremità opposta ci sono le icone delle categorie in cui l'articolo è inserito.



GIOVANNI PASCOLI



L'immagine a fianco rappresenta una foto della galleria.

Sotto il titolo è presente il contatore dei "Mi Piace" e sul lato opposto abbiamo le icone delle categorie della foto.

L'icona del cestino è visibile solo agli amministratori e ai blogger e permette di cancellare l'immagine dalla galleria.

Inoltre, contrariamente alle altre pagine, questo script presenta una slide bar diversa in cui mostra delle statistiche.

Articoli
 Questa categoria contiene il 33.3%(2) degli articoli totali.

Galleria
 Questa categoria contiene il 0%(0) delle foto totali.

Commenti
 Questa categoria contiene il 0%(0) dei commenti totali.

Mi Piace
 Questa categoria contiene il 0%(0) dei mi piace totali.

Articoli salvati
 Questa categoria contiene il 0%(0) degli articoli salvati totali.

CERCA

Tutti gli articoli della categoria

Tutte le foto della categoria

AMMINISTRAZIONE

Aggiorna categoria

In ordine abbiamo:

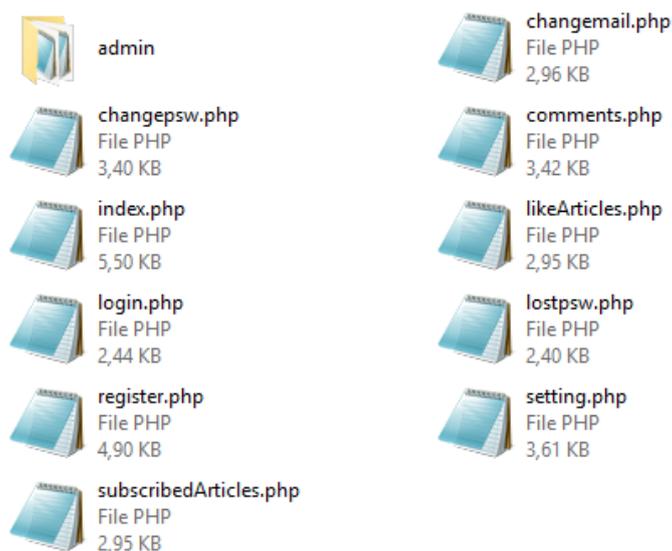
1. La percentuale degli articoli contenuti nella categoria.
2. La percentuale delle foto contenute in questa categoria.
3. La percentuale dei commenti ad articoli inseriti nella categoria.
4. La percentuale dei "Mi Piace" ad articoli che appartengono alla categoria.
5. La percentuale degli articoli salvati e che sono presenti nella categoria.

Inoltre, sotto le statistiche, è presente una sezione che contiene collegamenti alla pagina *search.php* (DIR: pagine) in cui si ricercano tutti gli articoli/foto presenti nella sezione.

Un altro aspetto è il contenitore Amministrazione, visibile solo agli amministratori e permette di modificare le informazioni della classe.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	article.php(DIR: functions)	
elements.css	photo.php(DIR: functions)	
style.css	category.php(DIR: functions)	
elementsStyle2.css	header.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	getPosts(...)	/
	getPhotos(...)	
	check(...)	
	countRecords(...)	
	print2_article(...)	
	photo(...)	

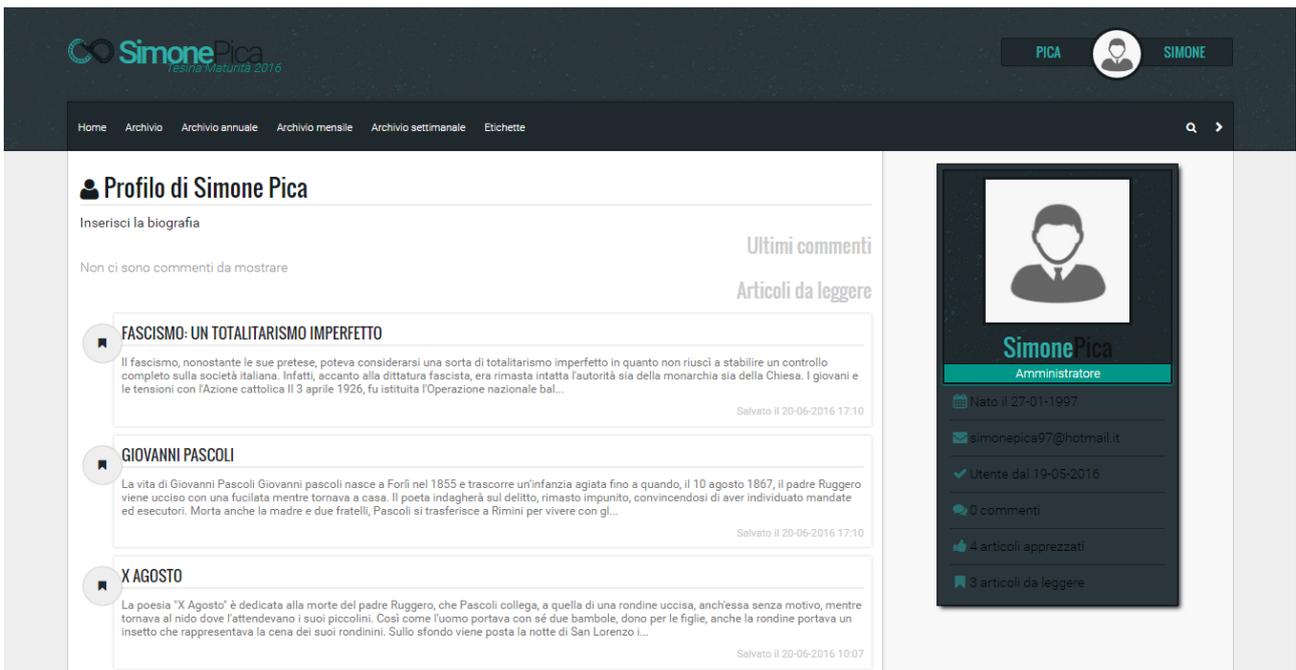
Cartella User:



La cartella **user** contiene tutte le pagine che compongono il profilo utente e il pannello di controllo. Inoltre è presente la sottocartella **admin** che raccoglie i file che costituiscono il pannello di amministrazione/blogger. Tutti i file presenti in questa directory includono la slide bar dell'utente (*slideBarUser.php*) o quella d'amministrazione (*slideBarAdmin.php*).

• FILE INDEX.PHP:

Il file *index* rappresenta il profilo utente. Sotto l'intestazione della pagina, in cui è presente il nome e il cognome dell'utente, sono presenti quattro sezioni.



La prima sezione rappresenta la biografia dell'utente. Questa porzione di pagina viene nascosta quando l'utente omette lo specifico campo.

Le seguenti sezioni mostrano:

1. Gli ultimi tre commenti pubblicati dall'utente.
2. Gli ultimi tre articoli apprezzati.
3. Gli ultimi tre articoli salvati.

Passando con il mouse sopra i vari contenitori verrà mostrato un menu che permette di effettuare delle azioni come modificare e cancellare un commento, togliere il "Mi Piace" all'articolo o eliminarlo dalla lista degli articoli salvati.



E' importante ricordare che tutti i collegamenti che permettono le modifiche vengono visualizzati solamente quando l'utente loggato visualizza il proprio profilo.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	profilo.php(DIR: functions)	
elements.css	slideBarUser.php(DIR: common)	
style.css	header.php(DIR: common)	
profilo.css	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT

	updateBio()	/
	deleteComment()	
	updateComment()	
	userProfile(...)	
	profileOption(...)	
	getComments(...)	
	getUserPosts(...)	

• FILE SUBSCRIBEDARTICLES.PHP – LIKEDARTICLES.PHP – COMMENTS.PHP:

Le seguenti pagine mostrano separatamente le tre sezioni presenti nella *index* del profilo:

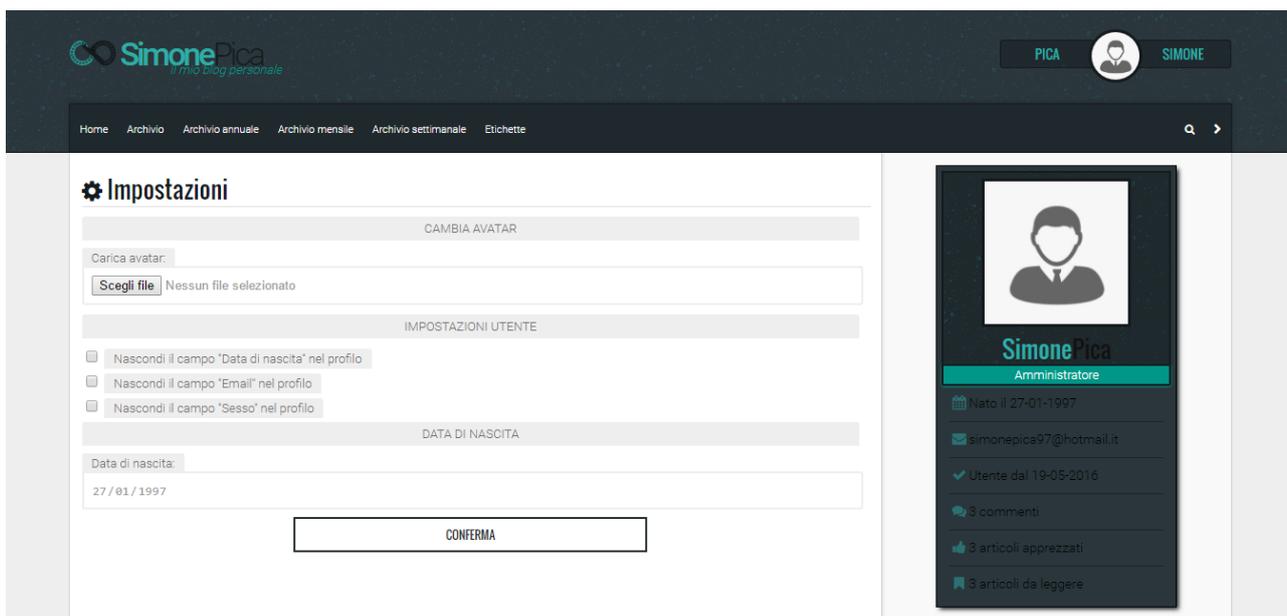
1. subscribedArticles.php: visualizza gli articoli salvati dall'utente.
2. likedArticles.php: mostra gli articoli apprezzati dall'utente.
3. comments.php: contiene tutti i commenti pubblicati dall'utente.

Ogni script illustra un massimo di dieci elementi. Quando questo numero viene superato, mediante un menu di navigazione, sarà possibile visualizzare gli oggetti più remoti.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	profilo.php(DIR: functions)	
elements.css	navpages.php(DIR: functions)	
style.css	slideBarUser.php(DIR: common)	
profilo.css	header.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	userProfile(...)	/

	navPages(...)	
	print_navigation(...)	
	SUBSCRIBEDARTICLES.PHP, LIKEDARTICLES.PHP	
	getUserPosts(...)	
	COMMENTS.PHP	
	deleteComment()	
	updateComment()	
	getComments(...)	

• FILE SETTING.PHP:



La pagina contiene il form per modificare le impostazioni del profilo utente. Come si può notare dalla foto il modulo si compone di tre parti:

1. Permette di caricare un nuovo avatar.
2. Consiste in una serie di checkbox che permettono di decidere se visualizzare o nascondere determinate informazioni del profilo.
3. E' presente il campo che permette di cambiare la data di nascita (Può risultare inutile, ma in realtà è necessario in quanto un utente alla registrazione può decidere di omettere la data di nascita ed inserirla in un secondo momento).

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	profilo.php(DIR: functions)	
style.css	slideBarUser.php(DIR: common)	
profilo.css	header.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	userProfile(...)	/
	updateSetting()	

• FILE CHANGEMAIL.PHP – CHANGEPSW.PHP:

Lo script *changemail.php* permette ad un utente di cambiare il proprio indirizzo email, contrariamente allo script *changePSW.php* che consente di modificare la password d'accesso. In entrambi i casi, quando il modulo viene confermato, sarà richiesto di effettuare nuovamente il login con i nuovi dati.

✉ Cambia Email

Questo modulo ti permetterà di cambiare l'email del tuo account!
Dopo aver compilato i vari campi, premi sul pulsante "Cambia Email" per essere reindirizzato alla pagina di login.
Al nuovo login dovrai utilizzare la nuova Email!!

Password:

Nuova email: Ripeti email:

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	profilo.php(DIR: functions)	
style.css	slideBarUser.php(DIR: common)	
profilo.css	header.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	userProfile(...)	/
	updateEmail()	

🔑 Cambia Password

Questo modulo ti permetterà di cambiare la password del tuo account!
Dopo aver compilato i vari campi, premi sul pulsante "Cambia Password" per essere reindirizzato alla pagina di login.
Al nuovo login dovrai utilizzare la nuova password!!

Vecchia Password:

Nuova Password: Ripeti Password:

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	profilo.php(DIR: functions)	
style.css	slideBarUser.php(DIR: common)	
profilo.css	header.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	userProfile(...)	/

	updatePsw()	
--	-------------	--

• FILE REGISTER.PHP:

PAGINA DI REGISTRAZIONE

Nome: <input type="text" value="Mario"/>	Data di nascita: <input type="text" value="gg/mm/aaaa"/>
Cognome: <input type="text" value="Rossi"/>	Sesso: <input type="text" value="Maschio"/>
Email: <input type="text" value="mario@rossi.it"/>	Password: <input type="text"/>
Ripeti email: <input type="text" value="mario@rossi.it"/>	Ripeti password: <input type="text"/>
Informazioni su di te: <div style="border: 1px solid #ccc; height: 60px; width: 100%;"></div>	
Carica avatar: <input type="button" value="Scegli file"/> Nessun file selezionato	
IMPOSTAZIONI UTENTE	
<input type="checkbox"/> Nascondi il campo "Data di nascita" nel profilo <input type="checkbox"/> Nascondi il campo "Email" nel profilo <input type="checkbox"/> Nascondi il campo "Sesso" nel profilo	
<input type="button" value="REGISTRATI"/>	

La pagina di registrazione è composta da un unico form che richiede di compilare una serie di campi. Per iscriversi i campi "Nome", "Cognome", "Email", "Password", "Ripeti email" e "Ripeti password" devono essere obbligatoriamente compilati. L'omissione di tali campi non permetterà di effettuare la registrazione.

Quando il form viene inviato lo script eseguirà una serie di controlli, tra i quali:

- Controlla l'uguaglianza tra "Email" e "Ripeti email".
- Controlla l'uguaglianza tra "Password" e "Ripeti password".
- Controlla che l'email inserita non sia già presente nel database.

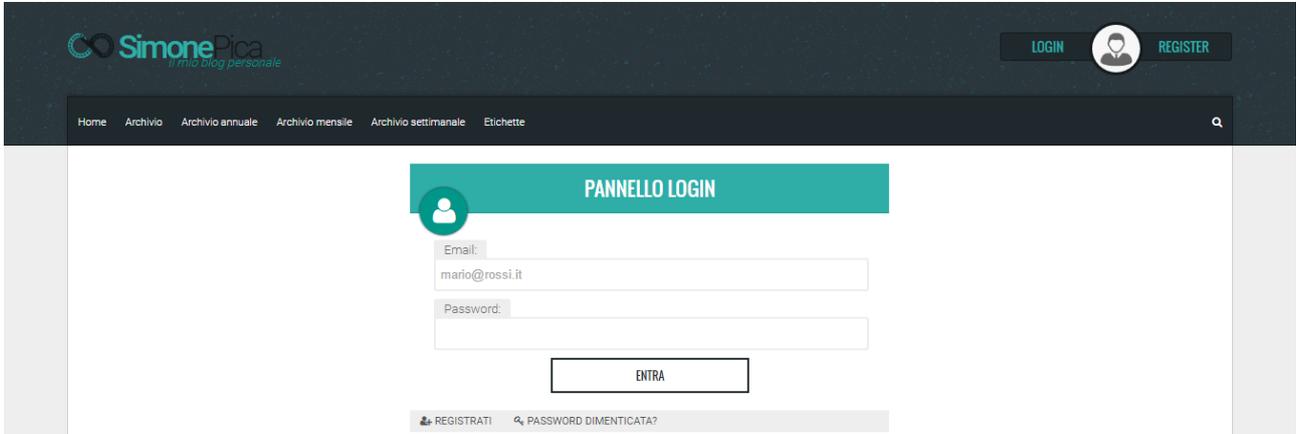
Inoltre sulla pagina agisce una funzione JavaScript che controlla il contenuto dei campi ripetuti colorandoli di verde, se corretti, altrimenti di rosso.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	register.php(DIR: functions)	
style.css	slideBar.php(DIR: common)	
loginRegisterPage.css	header.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT

	register()	check(...)
--	------------	------------

• FILE LOGIN.PHP:

La pagina di login è composta da un form che richiede l’email e la password.



In caso di accesso errato lo script ritornerà uno dei seguenti errori:

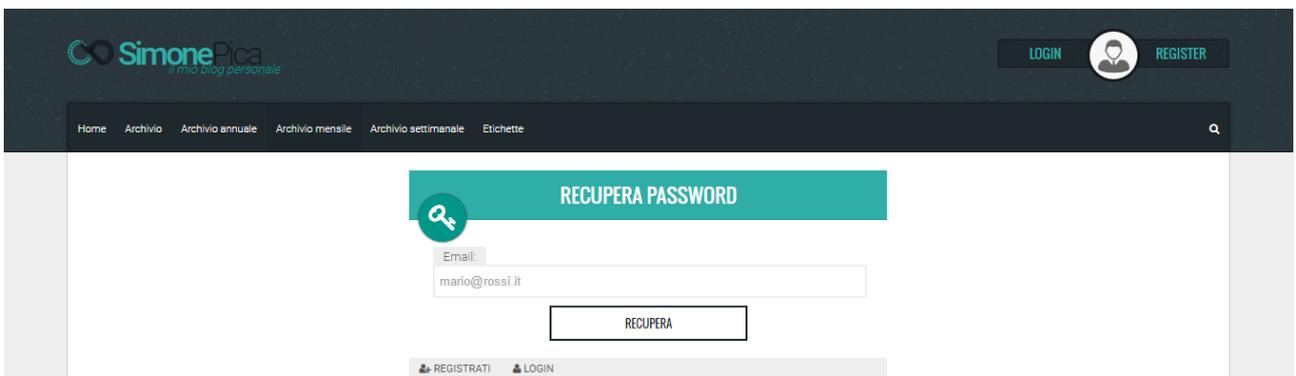
1. **Utente non presente nel database**, se si sta tentando l’accesso con un indirizzo email non registrato.
2. **La password inserita non è corretta**, se l’indirizzo email è presente nel database, ma la password inserita non coincide con quella memorizzata.

Nella parte finale della pagina sono presenti due collegamenti: uno che indirizza verso la pagina di registrazione e l’altro che carica lo script per reimpostare la password.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
style.css	header.php(DIR: common)	
loginRegisterPage.css	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP (Include dal global.php)	FUNZIONI PRINCIPALI JAVASCRIPT /

• FILE LOSTPSW.PHP:

Il file contiene la struttura della pagina che permette di reimpostare la password.



Come si può vedere dall’immagine il form della pagina è simile a quello presente in login.php, ma senza il campo “Password”. Per impostare una nuova password sarà necessario inserire l’indirizzo email nell’unico

campo presente e inviare il modulo. Successivamente, nella casella di posta elettronica, si riceverà un email simile a questa:



Cliccando sul collegamento verrà caricata la pagina *changepsw.php* in cui otterremo il campo "Vecchia Password" già compilato. Sarà necessario inserire la nuova password per reimpostarla.

Cambia Password

Questo modulo ti permetterà di cambiare la password del tuo account!
Dopo aver compilato i vari campi, premi sul pulsante "Cambia Password" per essere reindirizzato alla pagina di login.
Al nuovo login dovrai utilizzare la nuova password!!

Vecchia Password:

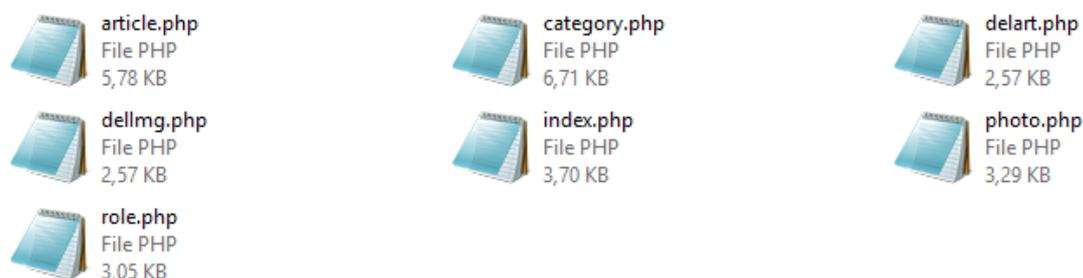
Nuova Password: Ripeti Password:

CAMBIA PASSWORD

Se si cerca di effettuare la procedura con un indirizzo email non presente nel database, lo script terminerà con un errore.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
style.css	header.php(DIR: common)	
loginRegisterPage.css	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	lostPsw(...)	/

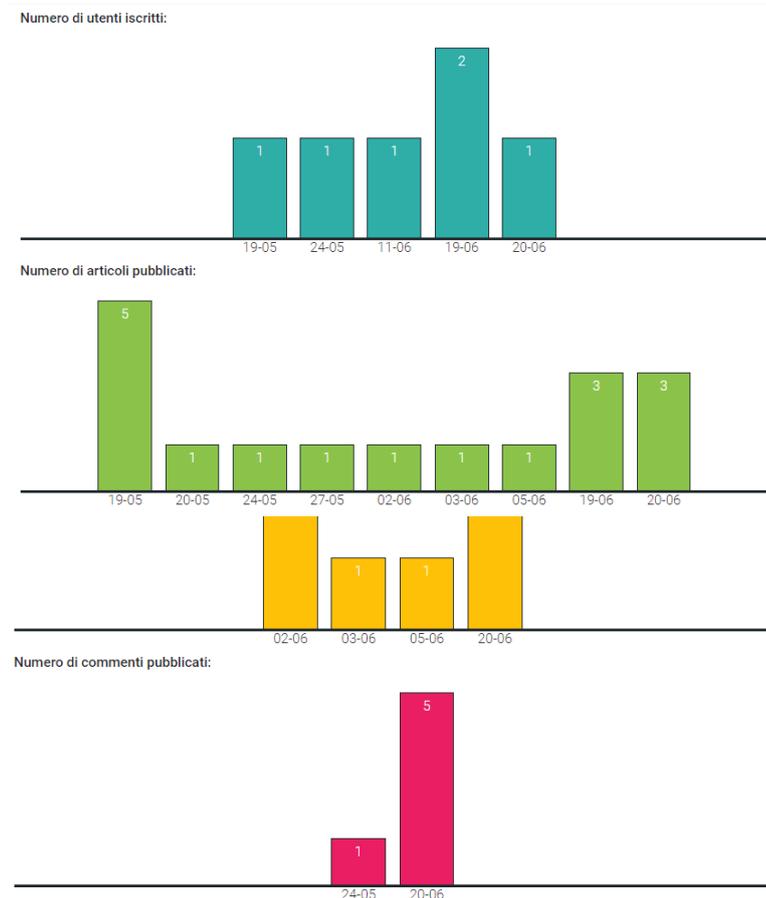
Cartella Admin:



La cartella **admin** contiene tutti i file che compongono il pannello di amministrazione. L'accesso alle pagine di questa cartella è riservato agli utenti che svolgono il ruolo di *Amministratore o Blogger*. Mentre l'amministratore può accedere a tutte le pagine, il blogger ne può visualizzare solo una parte.

• **FILE INDEX.PHP:**

La pagina principale della cartella rappresenta le statistiche del blog. Questa pagina è accessibile solo agli amministratori e contiene dei grafici che riportano la frequenza di alcuni dati in determinati giorni.



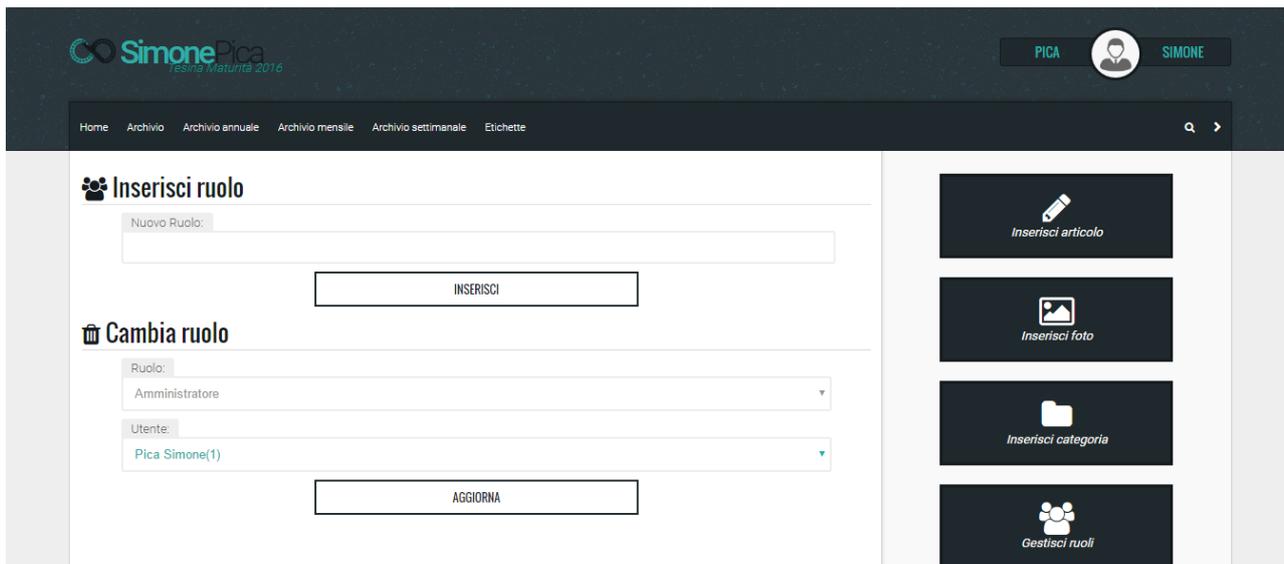
In ordine abbiamo che:

1. Il primo grafico mostra la frequenza degli utenti iscritti.
2. Il secondo grafico riporta la frequenza degli articoli pubblicati.
3. Il terzo grafico espone la frequenza delle foto pubblicate.
4. L'ultimo grafico mostra il totale dei commenti pubblicati.

Come si può evincere dalle immagini, nei grafici non vengono riportati i giorni in cui la frequenza è pari a zero.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	admin.php(DIR: functions)	
style.css	header.php(DIR: common)	
admin.css	slideBarAdmin.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	graph(...)	/

• FILE ROLE.PHP:



Analogamente alla pagina precedente anche questa è accessibile solo agli amministratori. La pagina si compone di due form distinti:

1. Il primo permette di creare un ruolo.
2. Il secondo è composto da due listbox che permettono di assegnare il ruolo scelto all'utente selezionato.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	admin.php(DIR: functions)	
style.css	header.php(DIR: common)	
admin.css	slideBarAdmin.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	role()	/
	updateRole()	

• FILE CATEGORY.PHP:

Questa pagina permette di creare una categoria o di aggiornarla se è già stata inserita nel database. All'interno del file viene incluso lo script *admin.php* e ciò lo rende accessibile esclusivamente agli amministratori.

La pagina è costituita da un modulo con tre campi:

1. Nome categoria: richiede d'inserire il nome della categoria.
2. Descrizione: consiglia di scrivere una breve descrizione sulla categoria (questo campo può essere omesso).
3. Icona: è un campo di sola lettura che viene modificato cliccando le icone presenti nel contenitore sottostante. Cliccando sul collegamento "Altre icone..." viene aperta una nuova finestra che carica la pagina del produttore del font in cui è possibile visualizzare tutte le immagini che possono essere assegnate alla categoria. Questa azione sbloccherà il campo "Icona" che diventerà editabile.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	admin.php(DIR: functions)	
style.css	header.php(DIR: common)	
admin.css	slideBarAdmin.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	newCategory(...)	Definisce all'interno la funzione insertCode(...) che permette di associare all'icona cliccata il valore.
	checkCategory(...)	

• FILE PHOTO.PHP:

Il file definisce la struttura della pagina che permette di inserire e cancellare una foto.

Carica immagine

Titolo immagine:

Carica immagine:

Nessun file selezionato

Tag (Separati da uno spazio vuoto):

CATEGORIE

- Informatica
- Inglese
- Matematica
- T.P.S

Elimina foto

Identificatore immagine:

Il primo form è composto da quattro campi:

1. Titolo immagine: è un campo facoltativo.
2. Carica Immagine: è obbligatorio e richiede di selezionare la foto da caricare. La dimensione della foto non deve superare 400Kb.
3. Tag: il campo può essere omissso, ma è fortemente consigliato riempirlo in quanto permette una ricerca più efficace della foto inserita.
4. Categorie: permette di selezionare più categorie in cui catalogare la foto. E' obbligatorio selezionare almeno una categoria.

Il secondo modulo richiede di inserire l'identificatore dell'immagine per eseguire lo script *dellmg.php* che elimina la foto. Ricordiamo che gli amministratori ed i blogger visualizzeranno a fianco di ogni foto l'icona del cestino che permetterà di richiamare automaticamente il codice che cancella la foto dal database.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	blogger.php(DIR: functions)	
style.css	header.php(DIR: common)	
admin.css	slideBarAdmin.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	sendPhoto()	/
	allCategories()	

• FILE DELIMG.PHP:

 Elimina immagine



Vuoi davvero eliminare questa immagine?

CONFERMA

La pagina viene utilizzata per confermare definitivamente la eliminazione di una foto. Una volta inviato il modulo verrà cancellato sia il puntatore presente nel database che l'immagine salvata nel file system storage. Analogamente alla pagina precedente, anche questa può essere visualizzata sia da amministratori che blogger.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	blogger.php(DIR: functions)	
style.css	header.php(DIR: common)	
admin.css	slideBarAdmin.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	imageFromTable(...)	/
	deletelImage(...)	

• FILE ARTICLE.PHP:

Il file definisce la pagina che permette di inserire un articolo. Richiedendo le funzioni di *blogger.php* rende automaticamente accessibile la pagina sia agli amministratori che ai blogger.

 Inserisci articolo

Nessun file selezionato
 Contenuto:
 B I U S    H      

 CATEGORIE
 Inglese
 Matematica
 T.P.S.

Il form è composto dai seguenti campi:

1. Titolo articolo: è obbligatorio e richiede di inserire un titolo all'articolo.
2. Sfondo articolo: è facoltativo e rappresenta l'immagine di copertina.
3. Contenuto: richiede di inserire obbligatoriamente nell'area di testo il contenuto dell'articolo. Sotto l'etichetta è presente un editor molto semplice che permette di applicare tag HTML, come ``, `<h1>`, `<h2>`, ... ,`<h6>`, `<i>`, `<s>`..., al testo selezionato. Inoltre l'editor consente di allineare il testo ed inserire delle liste od immagini.

4. Tag: è facoltativo, anche se è fortemente consigliato riempirlo in quanto migliora la ricerca all'interno del blog e l'indicizzazione da parte dei motori di ricerca.
5. Categorie: permette di selezionare più categorie in cui catalogare l'articolo. E' obbligatorio selezionare almeno una categoria.

Se l'invio del form non restituisce particolari problemi l'articolo verrà inserito correttamente.

Oltre a ciò se la pagina viene caricata passandogli un identificatore, lo script recupererà automaticamente l'articolo dal database e ne permetterà la modifica. Nel caso di identificatore errato verrà caricata la pagina `notfound.php(DIR: pagine)`

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	blogger.php(DIR: functions)	editor.js
style.css	header.php(DIR: common)	
admin.css	slideBarAdmin.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	sendArticle()	editorTesto(...)
	getPosts(...)	list(...)
	getTagArticle(...)	insertImg()
	getItemCategories(...)	tagHeader()
	allCategories()	

• FILE DELART.PHP:

 Elimina Articolo

Integrale definito e differenziale

Vuoi davvero eliminare questo articolo?

CONFERMA

Il file ha una funzione analoga allo script `delimg.php`, ma con lo scopo di rimuovere un articolo. Anche questa pagina può essere visualizzata da Amministratori e Blogger. La cancellazione di un articolo comporta l'eliminazione del record dal database con l'eventuale cancellazione, se presente, dell'immagine di copertina dal file system storage.

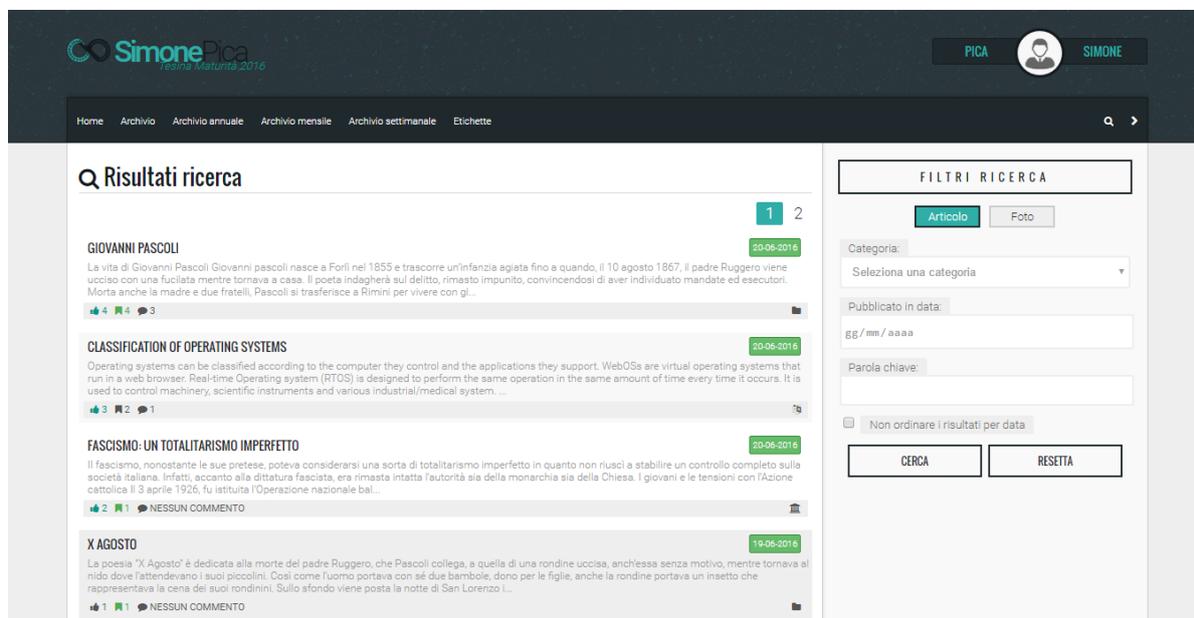
CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
slideBar.css	blogger.php(DIR: functions)	
style.css	header.php(DIR: common)	
admin.css	slideBarAdmin.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	getPosts(...)	/
	deleteArticle(...)	

File Pagine:

La cartella **pagine** contiene una serie di file che determinano la struttura di pagine generali.

• FILE SEARCH.PHP:

Lo script permette di ricercare articoli sulla base di alcuni criteri definiti dall'utente.



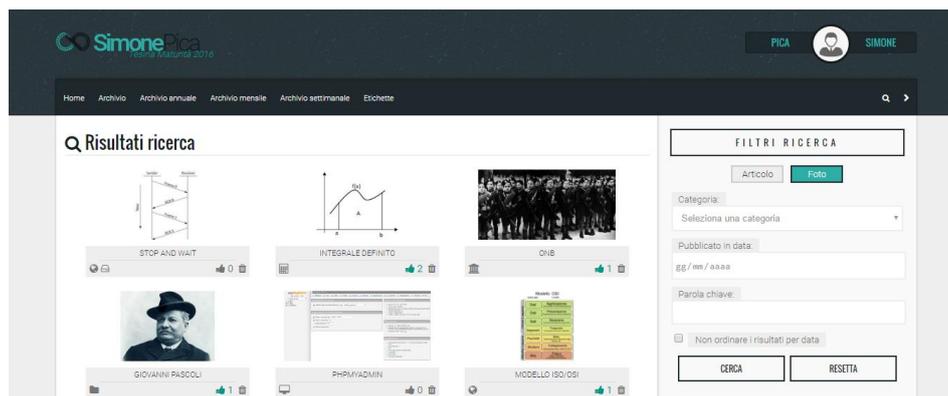
La pagina raccoglie una serie di elementi che ritornano dalla query di ricerca. Come si può notare dalla foto, nella barra laterale è presente il modulo che permette di impostare i parametri di ricerca.

Il form permette di:

1. Decidere se cercare articoli o foto, di default la ricerca avviene sugli articoli.
2. La combobox "Categoria" permette di ricercare gli elementi nella sezione indicata.
3. Il campo "Pubblicato in data" cerca gli elementi del blog nel giorno specificato.
4. Il campo "Parole chiave" limita la ricerca a tutti gli elementi che contengono uno dei vocaboli nel titolo o nei tag.
5. La checkbox, se selezionata, consente di ordinare i record ritornati dalla query in base al numero di corrispondenze con i criteri di ricerca.

In base ai principi definiti l'algoritmo sarà in grado di crearsi la query e ritornare i risultati. Analogamente a molti altri script, anche questo mostra un menu di navigazione qualora la ricerca produca un numero elevato di risultati.

Precisamente vengono visualizzati un massimo di dieci articoli e ventiquattro foto per pagina.



CSS		PHP		JAVASCRIPT
globale.css		global.php(DIR: functions)		globale.js
slideBar.css		navpages.php(DIR: functions)		
style.css		search.php(DIR: functions)		
elements.css	elementsStyle2.css	photo.php	article.php	
		(DIR: functions)		
		header.php(DIR: common)		
		footer.php(DIR: common)		
		FUNZIONI PRINCIPALI PHP		FUNZIONI PRINCIPALI JAVASCRIPT
		navPages(...)	/	
		searchItems(...)		
		print_navigation(...)		
		allCategories()		
		photo(...)	print2_article(...)	

• FILE ARCHIVE.PHP:

La pagina permette di raccogliere articoli e foto secondo un periodo di tempo definito. Questo sistema rappresenta una delle caratteristiche evidenti fin dai primi blog.

Allo stesso modo dello script search.php anche questo è costituito da una serie di elementi, visualizzati nel corpo della pagina, e dal modulo che definisce i criteri, presente nella barra laterale.

Nella slide bar possiamo notare un menu iniziale che permette di ottenere i seguenti archivi:

1. Archivio annuale.
2. Archivio mensile.
3. Archivio settimanale.

Questi raccoglitori sono accessibili anche dal menu principale del blog.

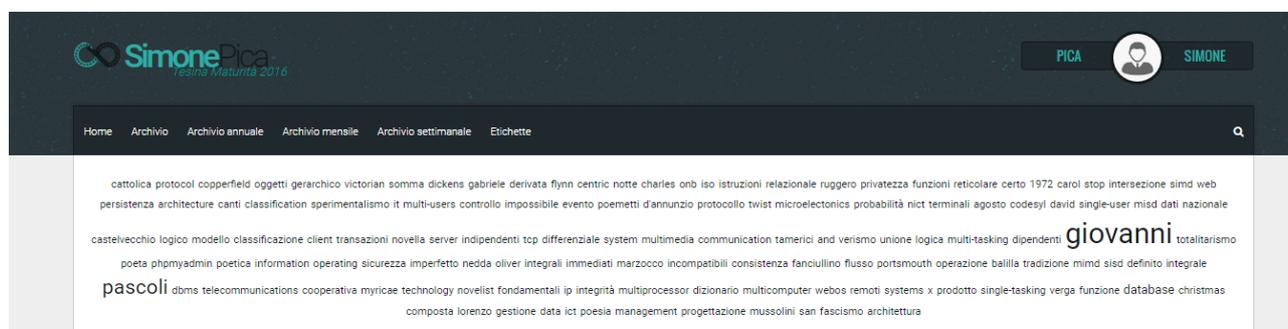
Sotto i collegamenti è presente un form che permette di accedere ad un tipo di archivio dinamico in cui la durata temporale non è definita a priori, ma viene definita dall'utente stesso. Inoltre il modulo consente di scegliere se visualizzare il raccoglitore di foto o di articoli.

Gli elementi visualizzati sono limitati e gestiti con lo stesso sistema utilizzato nello script *search.php* mostrando un massimo di dieci articoli e ventiquattro foto per pagina.

CSS		PHP	JAVASCRIPT
globale.css		global.php(DIR: functions)	globale.js
slideBar.css		navpages.php(DIR: functions)	
style.css		archive.php(DIR: functions)	
elements.css	elementsStyle2.css	header.php(DIR: common)	
		footer.php(DIR: common)	
		FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
		print_navigation(...)	/

• FILE TAG.PHP:

Questa pagina visualizza tutte i *tag* salvati nel database.



Le parole assumono una dimensione diversa che cresce in rapporto al numero di elementi che sono associati a quel termine. Inoltre, come si può notare dall'immagine, la pagina è composta solo dal contenitore centrale occupando anche lo spazio che nelle altre pagine è riservato alla barra laterale.

CSS		PHP	JAVASCRIPT
globale.css		global.php(DIR: functions)	globale.js
slideBar.css		header.php(DIR: common)	
style.css		footer.php(DIR: common)	
elements.css		FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
		getTag(...)	/

- FILE NOTFOUND.PHP:



Questa pagina viene caricata tutte le volte che uno script non viene eseguito correttamente o si sta tentando di visualizzare delle pagine riservate.

Analogamente alla struttura di tag.php, anche in questa è assente la barra laterale e l'intero contenuto viene utilizzato per mostrare un messaggio di errore.

CSS	PHP	JAVASCRIPT
globale.css	global.php(DIR: functions)	globale.js
	header.php(DIR: common)	
	footer.php(DIR: common)	
	FUNZIONI PRINCIPALI PHP	FUNZIONI PRINCIPALI JAVASCRIPT
	/	/

HOSTING

In informatica l'hosting consiste nell'allocare su un server web un sito rendendolo accessibile agli utenti di Internet. Generalmente utilizzando questo servizio più siti Web risiedono sulla stessa macchina, di conseguenza condividono memoria di massa e banda disponibile.

Per aprire il blog ho deciso di sfruttare un servizio di hosting acquistando il pacchetto Web hosting Linux PROFESSIONAL presso Shellrent.

Il pacchetto include:

- Spazio web illimitato.
- Traffico web illimitato.
- Un database con dimensione illimitata.
- Un dominio registrato come simonepica.it.
- Backup settimanale.
- Assistenza 24/24 7/7.
- Cinque caselle email con 1GB di spazio l'una.
- Molti altri servizi...



BIBLIOGRAFIA - SITOGRAFIA

Storia e caratteristiche dei blog: <http://informaticaitaliana.blogspot.it/2012/06/storia-e-caratteristiche-dei-blog-1.html> di Sandro Porràzzini *Professionista IT*

Nascita e storia dei blog:

http://www.partecipiamo.it/Comunicazione/valentina_pappagallo/nascita_e_storia_dei_blog.htm di Valentina Pappagallo

Blog, la nascita di un nuovo mass-media: <http://www.slideshare.net/shijokingo/blog-la-nascita-di-un-nuovo-mass-media-1876977> di Damiano Crognali *Creative Digital Strategist*

Blog: <https://it.wikipedia.org/wiki/Blog> da Wikipedia

<http://www.peterme.com/> di Peter Merholz

<http://www.desy.de/web/mosaic/archive-whats-new.html> di NCSA

Tim Barners-Lee: https://it.wikipedia.org/wiki/Tim_Barners-Lee da Wikipedia

Jorn Barger: https://en.wikipedia.org/wiki/Jorn_Barger da Wikipedia

Dave Winer: <http://davewiner.com/>