

Il tema del sito:

Il sito proposto è un blog personale dedicato alla cura dei capelli ricci. È stato sviluppato un blog che consente di pubblicare articoli, leggerli, lasciare un feedback sull'esperienza e ottenere informazioni relative ad esso.

Le sezioni principali:

La pagina iniziale del sito è gestita tramite il file index.php, che contiene tutti gli articoli resi visibili dall'amministratore. Una volta selezionato un articolo, si viene reindirizzati alla pagina post.php, dove vengono visualizzati il titolo e il contenuto dell'articolo.

Funzionalità:

Nella pagina register.php, è presente un modulo da compilare con l'Username, l'E-mail, la Password e la conferma della Password. Questi dati verranno controllati attraverso il file users.php. Se non ci sono errori durante la compilazione, l'utente verrà creato e le sue informazioni verranno inserite nel database.

La stessa procedura avviene nella pagina login.php, con la differenza che vengono richiesti solo l'Username e la Password. In questo modo, i campi verranno controllati con i dati presenti nel database. Quando si desidera effettuare il logout dal sito con le proprie credenziali, sarà sufficiente eseguire l'operazione di logout, che rimuoverà i dati dell'account dalla sessione.

Gli utenti avranno la possibilità di generare contenuto, come impostare un gradimento sull'esperienza avuta sul sito.

Sono stati introdotti due ruoli distinti all'interno del sistema: amministratore e utente base.

L'amministratore ha le stesse capacità di navigazione del blog come qualsiasi altro utente, ma gode anche di funzionalità aggiuntive grazie a un controllo specifico.

Attraverso il controllo `<?php if($_SESSION['admin']): ?>` l'amministratore può accedere alla sezione dedicata nella navbar denominata "dashboard". Questa sezione offre all'amministratore la possibilità di gestire in modo completo i contenuti del blog, inclusa la modifica, la cancellazione e l'aggiunta di nuovi elementi. In questo modo, l'amministratore ha un controllo totale sulla pubblicazione e l'amministrazione dei contenuti del blog.

Caratteristiche:

Interazione/animazione: Nella pagina principale index.php, è stata introdotta un'animazione che cattura l'attenzione degli utenti. Gli articoli pubblicati scorrono in modo autonomo, mostrandone contemporaneamente tre alla volta. Questa animazione dinamica crea un effetto accattivante e consente agli utenti di ottenere un'anteprima rapida di diversi articoli in modo fluido e intuitivo.

Validazione dei dati di input: Ogni volta che un utente inserisce dei dati, vengono effettuati controlli per verificare la correttezza dell'input. Questi controlli informeranno immediatamente l'utente sul successo o meno dell'azione compiuta. Ad esempio, si può utilizzare la seguente logica:

Utilizzando l'istruzione `if (empty($...))`, si verifica se è stato inserito un dato nel campo richiesto. Se non si verificano errori, il codice procede eseguendo le operazioni necessarie per portare a termine la richiesta.

Questo garantisce che i dati forniti siano validi e corretti, assicurando una corretta elaborazione delle informazioni inserite.

Usabilità: Come menzionato precedentemente, a seconda dell'esito dell'operazione richiesta, vengono generati messaggi che indicano il successo o il fallimento dell'azione compiuta. Sfruttiamo la possibilità di inviare messaggi attraverso la sessione utilizzando il comando `$_SESSION['...']`, seguito dal contenuto del messaggio che vogliamo visualizzare. Di seguito è riportato un estratto di codice dal file `users.php` che, in caso di assenza di errori, mostra il messaggio "Account eliminato con successo":

```
if (count($errors) === 0) {  
...  
$_SESSION['message'] = 'Account eliminato con successo';  
$_SESSION['type'] = 'success';
```

Nel caso di errori, seguiamo una procedura simile, utilizzando l'istruzione `array_push($errors, ...)` con il relativo messaggio di errore. Di seguito è riportato un esempio di codice che verifica se il campo 'username' dell'utente è vuoto e aggiunge un messaggio di errore appropriato:

```
if (empty($user['username'])){  
    array_push($errors, con relativo messaggio di errore.
```

In questo modo gestiamo in modo efficace la visualizzazione di messaggi di successo o di errore in base all'esito delle operazioni. È importante sottolineare che, in caso di errori durante il controllo della presenza di errori, l'apparizione di questi messaggi non comporta alcuna modifica ai dati.

Sicurezza:

Per garantire la sicurezza del sito, è stata implementata una robusta funzionalità di controllo dell'accesso tramite il file `middleware.php`, in collaborazione con il file `validateUser.php`. Questi componenti sono responsabili dell'autenticazione degli utenti e richiedono una corretta compilazione dei moduli di login o registrazione per consentire l'accesso al sito. Una volta completato con successo il processo di login o registrazione, l'utente viene automaticamente reindirizzato alla homepage del sito.

Il file `middleware.php`, insieme al file `users.php`, svolge un ruolo cruciale nella gestione dell'accesso al sito. Attraverso queste componenti, vengono verificate le credenziali degli utenti e viene garantita un'adeguata autenticazione per consentire l'accesso alle risorse protette.

Questa solida infrastruttura di gestione dell'accesso contribuisce a proteggere il sito da accessi non autorizzati, garantendo che solo gli utenti registrati e autenticati possano accedere alle funzionalità offerte dal sito. Grazie a questo approccio, si preserva la sicurezza dei dati e la privacy degli utenti, offrendo loro un'esperienza online affidabile e protetta.

Sessioni:

Come menzionato in precedenza, il file `config.php` assume la responsabilità di avviare la sessione utilizzando la funzione `session_start()`. Questo passaggio è essenziale per consentire l'accesso alle sessioni e stabilire una connessione al database.

Per terminare correttamente l'attività sul sito e disconnettersi, è necessario fare uso del file `logout.php`. Questo file si occupa di liberare le variabili di sessione e rimuovere i dati registrati dalla sessione, garantendo un effettivo logout dell'account dell'utente attraverso la funzione `session_destroy()`.

Attraverso questa procedura, gli utenti sono in grado di chiudere la propria sessione in modo sicuro, evitando che altre persone possano accedere ai loro dati personali. Il logout assicura la privacy e la sicurezza dell'utente, eliminando tutte le informazioni sensibili associate alla sessione attiva.

In sintesi, il file config.php si occupa di avviare la sessione e stabilire la connessione al database, mentre il file logout.php si occupa di liberare le variabili di sessione e rimuovere i dati registrati, offrendo agli utenti la possibilità di eseguire il logout in modo sicuro.

Interrogazione del db:

Le query al database consentono di manipolare i dati presenti nelle tabelle, consentendo operazioni come l'inserimento, il recupero, la modifica e la cancellazione dei dati. Queste interrogazioni vengono eseguite all'interno delle funzioni, in modo da essere attivate solo quando necessario. Un esempio di query potrebbe essere il seguente:

```
$sql = "SELECT p.*, u.username FROM posts AS p JOIN users AS u ON p.user_id=u.id WHERE p.published = ?";
```

 grazie al quale andiamo ad selezionare la tabella "post".

Questa query specifica ci permette di selezionare i dati dalla tabella "posts" e ottenere i campi desiderati, come il titolo dell'articolo e l'username dell'autore. Attraverso la clausola JOIN, si stabilisce una relazione tra la tabella "posts" e la tabella "users" utilizzando il campo "user_id". Inoltre, con la condizione "p.published = ?", possiamo filtrare i risultati in base alla pubblicazione degli articoli.

Le query al database consentono di effettuare operazioni mirate sui dati, fornendo un modo flessibile per interagire con le informazioni memorizzate nelle tabelle.

Presentazione:

Tutte le pagine web presentano un header fisso posizionato nella parte superiore dello schermo, seguito dal contenuto principale e infine da un footer. È stata adottata un approccio di design minimale, mantenendo un aspetto costante in tutte le pagine.

Front-end:

Per il progetto, conformemente alle richieste, sono state create diverse cartelle che contengono file con la stessa estensione. Di seguito sono elencate le cartelle e il loro contenuto:

- La cartella "images" contiene le immagini dal percorso relativo e sarà utilizzata per salvare le immagini caricate dall'amministratore tramite "create-post.php".
- La cartella "css" contiene i file di stile per le pagine pubbliche e private del sito.
- La cartella "php" contiene i file per le pagine pubbliche e private, nonché i file contenenti le funzioni necessarie al funzionamento del sito.
- La cartella "js" contiene i file JavaScript che rendono le pagine interattive. Al suo interno sono presenti i file "editor.js", che gestisce la barra di scrittura del corpo dei post, "feedback.js", che gestisce le recensioni fatte al sito, e infine "script.js", che si occupa delle animazioni del Post-Slider.

Back-end e comunicazione front/back-end:

Nella sezione riguardante le interrogazioni al database, abbiamo discusso delle query che vengono utilizzate per inserire, prelevare, modificare e cancellare i dati nel database. Questi dati sono organizzati in tabelle separate, ognuna con attributi e nomi distinti. Ecco un elenco delle tabelle e dei loro attributi:

- `posts` (`id`, `user_id`, `topic_id`, `title`, `image`, `body`, `published`, `created_at`) - in questa tabella vengono salvati i post.
- `topics` (`id`, `name`) - in questa tabella vengono salvati i topics.
- `users` (`id`, `admin`, `username`, `email`, `password`, `create_at`) - in questa tabella vengono salvati gli utenti.
- `valutazioni` (`id`, `name`, `valutazione`, `commento`) - in questa tabella vengono salvati i feedback degli utenti.

Le funzioni che gestiscono le interrogazioni al database sono contenute in un file dedicato, che consente di richiamarle e utilizzarle semplicemente associandole al punto appropriato del codice. Ad esempio, abbiamo `function getPublishedPosts()` nel file `users.php` viene utilizzata nel file `index.php` per ottenere i post pubblicati. Altre funzioni vengono richiamate tramite pulsanti presenti nel codice. Ad esempio, nel caso in cui sia attivato il pulsante 'update-topic', viene richiamata la funzione `$topic_id = update($table, $id, $_POST);`, la quale ha il compito di modificare il contenuto di un argomento specifico in base all'ID selezionato.

Nel codice per eliminare un oggetto specifico, viene utilizzata la variabile `$_GET`, che preleva il nome del pulsante, ad esempio `<a ... del_id=<?php echo $topic['id']; ?>" class="delete">delete`. Questo reindirizza all'apposita funzione correlata all'eliminazione. È importante sottolineare che, oltre a ciò, viene anche prelevato l'id dell'argomento, permettendo così di lavorare in modo specifico su un argomento selezionato senza influenzare i valori degli altri topics.

Utenti:

Nel database sono già presenti utenti con accesso attivo.

Username	Password	Admin
lara	admin	1
mirko	mirko	0
alice	alice	0
erica	erica	1