

## Relazione Tecnologie web

Università degli Studi di Torino, A.A. 2022-2023 Tecnologie Web (6 cfu)

Studente: Prandi Andrea (Matricola: 913224)

### Tema del Sito:

Il sito proposto è un blog personale dedicato all'universo dei videogiochi. Si tratta di una piattaforma online che permette agli appassionati di videogiochi di pubblicare articoli, leggerli, lasciare feedback sull'esperienza e accedere a informazioni correlate all'ambito videoludico.

### Sezioni Principali:

La pagina principale del sito, gestita attraverso il file `index.php`, presenta una selezione degli articoli pubblicati dagli utenti. Una volta selezionato un articolo, l'utente viene reindirizzato alla pagina `post.php`, dove può leggere il titolo e il contenuto completo dell'articolo.

### Funzionalità:

Nella pagina di registrazione (`register.php`), gli utenti possono compilare un modulo che richiede l'inserimento dell'Username, dell'E-mail, della Password e della conferma della Password. I dati inseriti in questo modulo verranno successivamente verificati attraverso il file `users.php`. Se la compilazione avviene senza errori, verrà creato un nuovo account utente e le relative informazioni saranno inserite nel database.

La stessa procedura di verifica dei dati dell'utente avviene nella pagina di login (`login.php`), con la differenza che in questo caso vengono richiesti solo l'Username e la Password per l'autenticazione. I dati inseriti verranno confrontati con quelli presenti nel database. Quando un utente desidera effettuare il logout dal sito utilizzando le proprie credenziali, basta eseguire l'operazione di logout. Questa operazione rimuoverà i dati dell'account utente dalla sessione, assicurando che l'utente sia effettivamente disconnesso dal sito.

Gli utenti hanno la possibilità di generare contenuto, ad esempio, possono esprimere un giudizio o un gradimento sull'esperienza che hanno avuto sul sito.

All'interno del sistema, sono stati introdotti due ruoli distinti: l'amministratore e l'utente base. L'amministratore gode delle stesse capacità di navigazione all'interno del blog come qualsiasi altro utente, ma ha anche accesso a funzionalità aggiuntive grazie a un controllo specifico. Attraverso una condizione come `<?php if($_SESSION['admin']): ?>`, l'amministratore può accedere a una sezione appositamente dedicata nella navbar denominata "dashboard". Questa sezione consente all'amministratore di gestire in modo completo i contenuti del blog, inclusa la possibilità di modificarli, cancellarli e aggiungere nuovi elementi. In questo modo, l'amministratore ha un controllo totale sulla pubblicazione e sull'amministrazione dei contenuti del blog.

### Caratteristiche:

### Interazione/Animazione:

La pagina principale `index.php` presenta un'animazione che cattura l'attenzione degli utenti. Gli articoli pubblicati scorrono in modo automatico, mostrandone contemporaneamente tre alla volta.

Questa animazione dinamica crea un effetto accattivante e permette agli utenti di visualizzare rapidamente diversi articoli in modo fluido e intuitivo.

## Validazione dei Dati di Input:

Ogni volta che un utente inserisce dati, vengono effettuati controlli per verificare la correttezza dell'input. Questi controlli forniscono all'utente un feedback immediato sull'esito dell'azione. Ad esempio, se un campo obbligatorio viene lasciato vuoto, l'utente riceverà un messaggio di errore.

## Usabilità:

In base all'esito delle azioni, vengono generati messaggi di successo o di errore. Questi messaggi vengono visualizzati attraverso sessioni e forniscono informazioni chiare sull'andamento delle operazioni.

## 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 precedentemente menzionato, a seconda dell'esito di un'operazione richiesta, vengono generati messaggi che indicano se l'azione è stata completata con successo o è fallita. Sfruttiamo questa funzionalità attraverso l'utilizzo delle sessioni utilizzando l'array `$_SESSION[ ' . . . ' ]`. In questo modo, è possibile fornire feedback agli utenti visualizzando i messaggi appropriati. Ecco un estratto di codice dal file `users.php` che mostra come gestire i messaggi di successo e di errore:

php

```
if (count($errors) === 0) {  
    // Nessun errore, l'operazione è stata completata con successo  
    $_SESSION['message'] = 'Account eliminato con successo';  
    $_SESSION['type'] = 'success';  
} else {  
    // Ci sono errori, aggiungi messaggi di errore all'array $errors  
    array_push($errors, "Messaggio di errore appropriato");  
}
```

In questo modo, gestiamo in modo efficace la visualizzazione dei messaggi di successo o di errore in base all'esito delle operazioni. Se non ci sono errori, viene impostato un messaggio di successo e un tipo di messaggio "success". In caso di errori, vengono aggiunti messaggi di errore appropriati all'array `$errors`.

È importante sottolineare che, se si verificano errori durante il processo di verifica degli errori stessi, la visualizzazione di tali messaggi non altera i dati precedentemente memorizzati. In altre parole, il sistema fornisce feedback senza apportare modifiche ai dati stessi, garantendo una gestione adeguata degli eventi sia in caso di successo che di errore.

## Interrogazione del Database:

**Query Prepare:** Le query SQL sono utilizzate in modo preparato, consentendo l'inserimento sicuro dei dati dell'utente. Ad esempio, quando si cerca un utente specifico nel database, viene utilizzato un segnaposto per inserire in modo sicuro il nome utente.

Un esempio di query potrebbe essere:

```
$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".
```

**Utilizzo di Funzioni per le Interrogazioni:** Per mantenere la coerenza e la sicurezza del codice, sono state implementate funzioni personalizzate per eseguire le interrogazioni al database. Ad esempio, la funzione `getPublishedPosts()` consente di recuperare in modo sicuro i post pubblicati.

**Accesso Controllato ai Dati:** L'accesso ai dati del database è rigorosamente limitato agli utenti autorizzati. L'autenticazione è un prerequisito per le operazioni che coinvolgono la scrittura o la modifica dei dati.

## Presentazione:

Tutte le pagine web seguono una struttura comune con un header fisso in alto, seguito dal contenuto principale e infine dal footer. Il design del sito è minimalista e coerente su tutte le pagine.

## Front-end:

Il progetto prevede diverse cartelle con file dello stesso tipo:

- La cartella "images" contiene immagini utilizzate nelle pagine web, inclusa la possibilità di caricare immagini per gli articoli.
- La cartella "css" contiene file di stile per tutte le pagine pubbliche e private del sito.
- La cartella "php" contiene file per le pagine pubbliche e private, nonché funzioni necessarie per il funzionamento del sito.
- La cartella "js" include file JavaScript che aggiungono interattività alle pagine web, come l'editor di testo per gli articoli e la gestione dei feedback.

## Back-end e Comunicazione Front/Back-end:

Le funzioni che gestiscono le interrogazioni al database sono collocate in un file dedicato, il che facilita il loro utilizzo all'interno del codice associandole ai punti appropriati. Ad esempio, nel file `users.php`, abbiamo una funzione denominata `getPublishedPosts()`, la quale viene utilizzata nel file `index.php` per recuperare i post che sono stati pubblicati e devono essere visualizzati nella pagina principale del blog.

Altre funzioni vengono richiamate attraverso pulsanti presenti nel codice. Ad esempio, se l'utente attiva il pulsante 'update-topic', viene invocata la funzione `$topic_id = update($table, $id, $_POST);`. Questa funzione ha la responsabilità di modificare il contenuto di un argomento specifico nel database in base all'ID selezionato.

Nel codice per eliminare un oggetto specifico, viene utilizzata la variabile `$_GET`, la quale preleva il nome del pulsante. Per esempio, nel codice HTML potresti avere un elemento come `<a ... del_id=<?php echo $topic['id']; ?>" class="delete">delete</a>`, che indica l'eliminazione di un argomento. Questo reindirizza l'utente all'apposita funzione correlata all'eliminazione. È importante sottolineare che oltre a ciò, viene anche prelevato l'ID dell'argomento specifico, consentendo di lavorare in modo specifico su un argomento selezionato senza influenzare i valori di altri argomenti. In altre parole, il sistema gestisce l'eliminazione in modo mirato senza coinvolgere dati non correlati