

config.php

Contiene le credenziali per la connessione al database:

Questo file viene utilizzato per configurare la connessione al database.

dashboard.php

Pagina principale della dashboard dopo il login.

- Controlla se l'utente è autenticato, altrimenti lo reindirizza a `login.php`.
- Mostra un messaggio di benvenuto e il numero di pagina selezionato dall'utente.
- Include un pulsante per effettuare il logout.

functions.php

Contiene funzioni di supporto per la gestione del database e degli utenti:

- **connectDB()**: Connette al database utilizzando le credenziali di `config.php`.
- **registerUser()**: Registra un nuovo utente con username, password hashata e numero di pagina.
- **checkUser()**: Verifica username e password durante il login, restituendo l'ID utente e il numero di pagina associato.

index.php

Pagina iniziale del sito.

- Controlla se l'utente è già autenticato e, in tal caso, lo reindirizza alla `dashboard.php`.
- Offre due pulsanti: uno per la registrazione (`register.php`) e uno per il login (`login.php`).

login.php

Gestisce il login degli utenti.

- Controlla username e password con la funzione `checkUser()`.
- Se l'accesso è corretto, avvia una sessione e reindirizza alla `dashboard.php`.
- In caso di errore, mostra un messaggio di avviso.

logout.php

Distrugge la sessione attuale e reindirizza l'utente a `login.php`.

- Serve per effettuare il logout in modo sicuro.

protected.php

Pagina protetta accessibile solo agli utenti autenticati.

- Se l'utente non è loggato, viene reindirizzato a `login.php`.
- Mostra un messaggio di benvenuto con un link per il logout.

register.php

Pagina per la registrazione di nuovi utenti.

- Utilizza `registerUser()` per salvare le credenziali nel database.
- Mostra un messaggio di conferma in caso di successo, altrimenti segnala eventuali errori.

MySql

```
-- Creazione del database
CREATE DATABASE IF NOT EXISTS prova_sql;
USE prova_sql;

-- Creazione della tabella degli utenti
CREATE TABLE IF NOT EXISTS users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  password VARCHAR(255) NOT NULL,
  page_number INT NOT NULL,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Il Funzionamento delle Sessioni in PHP

Le **sessioni** in PHP servono a mantenere lo stato tra diverse pagine web, permettendo di memorizzare informazioni sugli utenti (ad esempio ID, nome, preferenze) senza doverle passare tramite URL o form.

Avvio di una sessione

Prima di poter usare le sessioni, bisogna avviarle con:

```
session_start();
```

Questo comando:

Controlla se esiste già una sessione attiva.

Se esiste, la riprende con i dati memorizzati.

Se non esiste, ne crea una nuova.

Va sempre posizionato **all'inizio della pagina PHP**, prima di qualsiasi output HTML.

Memorizzazione di dati nella sessione

Dopo aver avviato la sessione, si possono salvare dati in un array speciale `$_SESSION`:

```
session_start();
$_SESSION['user_id'] = 1; // Salva l'ID utente
$_SESSION['username'] = "Mario"; // Salva il nome dell'utente
$_SESSION['role'] = "admin"; // Salva il ruolo dell'utente
```

Questi dati saranno disponibili su **tutte le pagine del sito** finché la sessione non viene distrutta.

Recupero di dati dalla sessione

Per leggere i dati della sessione in un'altra pagina:

```
session_start();
echo "Benvenuto " . $_SESSION['username']; // Output:
Benvenuto Mario
```

Se il dato non è stato impostato, restituirà un errore. Per sicurezza, è meglio controllare se esiste prima di usarlo:

```
if (isset($_SESSION['username'])) {  
    echo "Benvenuto " . $_SESSION['username'];  
} else {  
    echo "Nessun utente connesso."  
}
```

Protezione delle pagine con sessioni

Puoi usare le sessioni per limitare l'accesso alle pagine solo agli utenti autenticati. Ad esempio, nel file `dashboard.php`:

```
session_start();  
if (!isset($_SESSION['user_id'])) {  
    header("Location: login.php"); // Reindirizza alla pagina  
di login se l'utente non è loggato  
    exit();  
}
```

Se l'utente non è autenticato, verrà reindirizzato a `login.php`.

Distruzione della sessione (Logout)

Per terminare la sessione e rimuovere i dati salvati, si usa:

```
session_start();  
session_destroy(); // Distrugge la sessione  
header("Location: login.php"); // Reindirizza alla pagina di  
login  
exit();
```

Il file `logout.php` nel tuo progetto esegue proprio questa operazione.

Dove sono salvate le sessioni?

PHP salva l'ID della sessione in un **cookie** sul browser (PHPSESSID).

I dati della sessione sono invece memorizzati sul server, in una cartella temporanea.

Se l'utente chiude il browser, la sessione può scadere (dipende dalle impostazioni del server).