

Documentazione del Progetto - Quaderno di Informatica

Questo progetto rappresenta un quaderno digitale per documentare esercizi di sviluppo web e PHP. Ogni esercizio è stato implementato utilizzando concetti di base e intermedi per garantire chiarezza e funzionalità.

Esercizi Implementati:

- **student_table:** Creazione di una pagina in PHP che permette di creare una tabella al fine di memorizzare i voti degli studenti.
 - **Normalizzazione:** Creazione di una pagina in PHP che spiega la normalizzazione nelle tabelle.
 - **Esercizio A:** Creazione di una tabella Pitagorica dinamica in PHP.
 - **Esercizio B:** Messaggio personalizzato basato sull'ora del giorno e il browser.
 - **Esercizio C:** Generazione di triangoli con cicli `for` e simboli `*`.
 - **Esercizio D:** Sistema di login semplice con verifica di username e password.
 - **Esercizio E:** Creazione di una tabella con quadrati e cubi di numeri da 1 a N.
 - **Esercizio F:** Sistema di Sign In e Login con validazione dell'email.
 - **Esercizio G:** Spiegazione della normalizzazione nei database, anche implementata nel file `implementazione.php`.
 - **Database:** implementazione di un database che permette all'utente di usare un database scegliendo le tabelle che preferisce visualizzare.
-

Dettagli sugli Esercizi:

1. Normalizzazione:

- Spiega il fenomeno della normalizzazione nelle tabelle per la formazione dei database.
- Mostra esempi e spiegazioni, senza utilizzare particolari algoritmi.

2. Student_Table:

- Consente di creare una tabella per inserire le informazioni di uno studente con i relativi voti e materie.
- Calcola la media, salva i dati dello studente, aggiunge righe se necessario e consente di passare a un nuovo studente.

3. Esercizio A:

- Genera una tabella Pitagorica dinamica che moltiplica i numeri da 1 a 10 utilizzando cicli `for`.
- Ogni cella della tabella contiene il prodotto tra riga e colonna.

4. Esercizio B:

- Mostra un messaggio personalizzato all'utente in base all'ora corrente e al browser rilevato.
- Utilizza la variabile `$_SERVER` per ottenere le informazioni del browser.

5. Esercizio C:

- Genera diversi tipi di triangoli utilizzando cicli `for` per gestire gli spazi e gli asterischi.

6. **Esercizio D:**

- Implementa un sistema di login semplice con credenziali predefinite.

7. **Esercizio E:**

- Permette all'utente di scegliere un numero `N` (tra 1 e 10) tramite un menu a tendina.
- Genera una tabella con quadrati e cubi dei numeri da 1 a `N`.

8. **Esercizio F:**

- Sistema di Sign In e Login con validazione delle credenziali.
- Utilizza `filter_var()` per la validazione e salva temporaneamente i dati nella sessione per la verifica.

9. **Esercizio G:**

- Spiegazione della normalizzazione nei database, implementata in `implementazione.php`.

DATABASE

Relazioni

1. `films` ↔ `shows`

- **Relazione:** Un film può essere proiettato in più spettacoli (relazione 1:N).
- **Dipendenza:** La tabella `shows` dipende dalla tabella `films` tramite la chiave esterna `film_id`, che fa riferimento a `films.id`.
- **Regola di integrità referenziale:** Se un film viene eliminato (`films.id`), tutti gli spettacoli associati (`shows.film_id`) vengono eliminati automaticamente grazie a `ON DELETE CASCADE`.

2. `halls` ↔ `shows`

- **Relazione:** Una sala può ospitare più spettacoli (relazione 1:N).
- **Dipendenza:** La tabella `shows` dipende dalla tabella `halls` tramite la chiave esterna `hall_id`, che fa riferimento a `halls.id`.
- **Regola di integrità referenziale:** Se una sala viene eliminata (`halls.id`), tutti gli spettacoli associati (`shows.hall_id`) vengono eliminati automaticamente grazie a `ON DELETE CASCADE`.

3. `shows` ↔ `bookings`

- **Relazione:** Uno spettacolo può avere più prenotazioni (relazione 1:N).
- **Dipendenza:** La tabella `bookings` dipende dalla tabella `shows` tramite la chiave esterna `show_id`, che fa riferimento a `shows.id`.
- **Regola di integrità referenziale:** Se uno spettacolo viene eliminato (`shows.id`), tutte le prenotazioni associate (`bookings.show_id`) vengono eliminate automaticamente grazie a `ON DELETE CASCADE`.

