

Tema del sito e sezioni principali

MyLovedMovies è un sito che gestisce le informazioni principali di film e attori. Permette all'utente di creare liste personali di film e di rimanere sempre aggiornato sugli ultimi in uscita o quelli momentaneamente più votati, grazie a punteggi e recensioni.

Il sito è diviso principalmente nelle seguenti sezioni:

- **Login:** dove l'utente può accedere tramite le sue credenziali o registrarsi se non lo ha ancora fatto
- **Homepage:** vengono presentati i primi cinque film di tre categorie: recentemente usciti, più votati e preferiti dell'utente. L'utente può scegliere di quale categoria visualizzarne la lista completa.
- **All movies Page:** contiene la lista di tutti i film di una categoria in particolare e permette inoltre di effettuare una ricerca avanzata con vari filtri.
- **Movie Page:** contiene le informazioni specifiche di un film e permette all'utente di lasciare un voto, una recensione o aggiungere il film a una delle due liste personali (film preferiti, film da guardare)
- **Actor Page:** contiene le informazioni di un attore
- **Banner:** contiene, oltre al pulsante di log-out, una barra in cui si può effettuare la ricerca di un particolare film o attore.
- **Area Drag&Drop:** area a comparsa in cui si può rilasciare un film per aggiungerlo alla lista dei film da vedere

Funzionalità

Login/Registrazione

Al caricamento di ogni pagina del sito, viene effettuata inizialmente una richiesta AJAX per verificare se è presente una sessione con l'utente e in caso contrario si viene reindirizzati alla schermata di login. Qui l'utente può scegliere di effettuare il login con username e password oppure creare un nuovo account compilando l'apposito form. Lato client, in entrambi i casi, al click del tasto apposito e se nessun campo è vuoto, viene effettuata una richiesta POST tramite AJAX fornendo al web service i parametri necessari; nella funzione di callback, se tutto è andato a buon fine, si viene reindirizzati alla homepage, altrimenti viene visualizzato l'eventuale errore. Lato server, invece, nel caso del login viene verificata la corrispondenza tra email e password (che restituisce errore anche nel caso in cui l'utente non è presente nel database). Nel caso della registrazione, dopo aver verificato che l'email inserita o l'username inserito non siano già presenti, l'utente viene aggiunto nel database. Se le operazioni vanno a buon fine, viene rigenerato l'id della sessione (se ne era presente già una) e salvato il nome dell'utente in una variabile di sessione.

Logout

Se l'utente ha effettuato l'accesso viene visualizzato a destra nel banner un pulsante di logout. Al suo click, viene effettuato lato client una richiesta GET tramite AJAX, al termina della quale si viene reindirizzati alla schermata di login. Il lato server si occupa di effettuare il flushing dei dati per cancellare completamente la sessione tramite `session_unset` e terminare la sessione tramite `session_destroy`.

Gestione contenuto generato dall'utente

L'utente ha la possibilità sia di creare del nuovo contenuto (es. Aggiungendo una recensione) oppure organizzarlo a suo piacimento (es. Aggiungendo un film alla lista dei preferiti).

Nella pagina di un singolo film l'utente può visualizzare, oltre alle informazioni principali, una descrizione di esso, gli attori che hanno recitato e le recensioni lasciate dagli altri utenti.

Ogni utente, ha la possibilità di commentare un film inserendo la propria opinione in un form apposito. Nel momento in cui viene confermata la recensione cliccando l'apposito tasto, viene effettuata una richiesta POST tramite AJAX al web service incaricato di salvare la recensione nel database.

Oltre a una recensione, un utente può assegnare ad ogni film un voto da 1 a 5. Quando un utente clicca una stellina, viene eseguita una richiesta AJAX di tipo POST in cui si richiede di aggiornare nel database il punteggio assegnato dall'utente per quel film. Analogamente, un pulsante permette di rimuovere completamente un voto assegnato.

L'utente ha disposizione, inoltre, due pulsanti che permettono di aggiungere o rimuovere un film dalla rispettiva lista (preferiti o da guardare) tramite una richiesta AJAX di tipo POST. Il web service che riceverà il tipo di richiesta si conatterà al database ed effettuerà le modifiche necessarie. Inoltre, l'aggiunta dei film alla lista dei film "da guardare" è possibile effettuarla dalla homepage o dalla lista di tutti i film, semplicemente trascinando la locandina dei film nell'area apposita che compare quando si trascina un film.

Caratteristiche

Usabilità

Ogni azione che l'utente compie è seguita da un feedback visivo o, nel caso di eventuale errore, dalla comparsa di un messaggio. Alcuni esempi possono essere la visualizzazione del messaggio d'errore nel login se dei campi non sono stati compilati, il lampeggiamento delle stelle che confermano l'assegnazione del punteggio ad un film o lo "scuotimento" (shake) dell'area Drag&Drop se il film non è stato aggiunto correttamente alla lista.

Interazione/animazione

Nella sezione Homepage e All movies page, l'utente ha la possibilità di aggiungere un film alla lista dei film da guardare semplicemente effettuando un drag and drop, senza il bisogno di aprire la pagina del film specifico. Infatti, quando si prova a trascinare la locandina di un film, verrà scatenato l'evento dragstart e comparirà un'area dove vengono visualizzati i film già presenti nella lista e dove si può rilasciare un nuovo film per aggiungerlo. Questa lista viene aggiornata ogni volta al caricamento della pagina tramite una richiesta AJAX. Quando ci si trova nell'area corretta per rilasciare il film, il suo sfondo passerà da rosso ad arancione per indicare la giusta posizione; rilasciando il film in quest'area, viene effettuata una richiesta AJAX per aggiungerlo alla lista; se l'esecuzione va a buon fine lo sfondo dell'area diventerà verde, altrimenti rimarrà arancione ed effettuerà un'animazione di "shake", ad indicare il fallimento dell'operazione, magari perché il film in questione è già presente nella lista.

Sessioni

La sessione viene creata al momento del login e viene memorizzata una variabile contenente l'username dell'utente, utilizzata per identificarlo in tutte le operazioni successive. La sessione viene ripresa ogni qualvolta si utilizza una funzione php; infatti nel file common è presente un session_start che verrà richiamato prima di qualsiasi operazione. La sessione viene cancellata e terminata al momento del logout, come descritto nelle funzionalità.

Interrogazione del database

Qualsiasi informazione presente nel sito è memorizzata in un database, che viene interrogato con delle query più o meno complesse in base al tipo di operazione. Inoltre, nella sezione All movies page, in cui si può navigare tra le varie categorie, è presente un apposito form che permette una ricerca avanzata con vari filtri. In ogni ricerca è possibile, tramite il form, costruire delle query a proprio piacimento che permettono un'interrogazione più specifica.

Validazione dati input

Qualsiasi input inserito dall'utente viene validato in base al tipo di informazione e ne viene sempre verificata la sua esistenza. Per esempio, nel momento in cui un utente prova a registrarsi deve inserire una email valida (verifica lato client con funzione test ed espressione regolare, lato server funzione filter_val) altrimenti la richiesta AJAX per la registrazione non partirà e verrà visualizzato un messaggio d'errore. Così come per la password o per altri input come l'anno d'uscita di un film nella ricerca avanzata.

Sicurezza

Per prevenire attacchi XSS e HTML/SQL Injection qualsiasi input fornito dall'utente viene gestito lato server prima di essere utilizzato. Tutti i parametri forniti alle funzioni php vengono filtrati tramite la funzione `strip_tags` che elimina tutti i tag HTML, così come avviene per i parametri passati alle query che vengono filtrati tramite il metodo `quote`. Inoltre, nel database, nessuna password viene salvata in chiaro ma viene salvato l'hash generato dal metodo `md5`, utilizzato poi anche per verificare la sua correttezza al login.

Presentazione

Il design è semplice e il sito si presenta facile da utilizzare. Tutte le pagine hanno in comune un banner; in base al contenuto da visualizzare è stato scelto un layout a una o a due colonne; per esempio, nella pagina di un attore o di un film troviamo due colonne in cui a sinistra è presente la foto del film o dell'attore e a destra tutte le informazioni, così come avviene nella pagina in cui sono visualizzati tutti i film di una categoria.

Front-end

Il sito adotta uno stile unobtrusive: tutte le componenti quali presentazione/contenuto/comportamento sono separate. Lo stile e la presentazione sono gestiti esclusivamente dal file css contenuto nella cartella css, mentre il contenuto dai vari file presenti nella cartella html, ognuno dei quali rappresenta una pagina del sito. Il comportamento delle pagine è responsabilità dei file javascript contenuti nella cartella js, i quali si occupano di rispondere a tutti gli eventi causati dall'utente, recuperare gli elementi del DOM per aggiornare i contenuti del sito con le informazioni ricevute dalle funzioni situate nella cartella php, generare animazioni ed effetti. Ogni file javascript gestisce una singola pagina, mentre per le funzioni comuni è presente il file `common`; un altro file si occupa del drag e drop come specificato nella sezione Interazione/Animazione.

Back-end e comunicazione front/back-end

Lo schema relazionale del database è il seguente:

ATTORE (**Codice**, Nome, Cognome, DataNascita, LuogoNascita, Eta, Foto, Descrizione)

ATTOREFILM (**Attore**, **Film**, **Personaggio**)

FILMDAVEDERE (**Titolo**, **Utente**)

FILMFAVORITO (**Film**, **Utente**)

RECENSIONI (**Film**, **Utente**, Recensione)

REGISTA (**Codice**, Nome, Cognome, DataNascita, LuogoNascita, Foto)

UTENTE (**Username**, Password, Nome, Cognome, Email)

VOTOFILM (**Id**, **Utente**, **Film**, Punteggio)

FILM (**Titolo**, Musiche, DataUscita, Descrizione, Foto, MediaVoti, Genere, Paese, Distribuzione, Produzione, Durata, Sceneggiatura, Regista)

Le funzioni php rappresentano un web service il quale compito è quello di restituire le informazioni al client in maniera corretta, come fosse una API. I metodi utilizzati dalle funzioni php risiedono nel file `common` e svolgono principalmente le attività di connessione al database ed esecuzione delle query.

Tutte le comunicazioni tra back e front end avvengono tramite un canale di comunicazione AJAX per consentire l'aggiornamento dinamico delle pagine ed evitare l'abuso del pattern click-wait-refresh. Per esempio, se nella pagina che visualizza le informazioni di un film effettuiamo la ricerca di un altro film nella barra di ricerca, possiamo notare come la pagina aggiorni i contenuti, senza effettuare il refresh.

Tutte le funzioni, che vadano a buon fine o no, restituiscono il risultato o un messaggio d'errore in formato json. Ogni funzione, prima di effettuare l'operazione necessaria, verifica che sia presente la sessione utente; se così non fosse restituisce direttamente un errore: `{ "Error": "Request not valid. Please, login" }`. Anche il lato client verificherà sempre la corretta avvenuta dell'operazione nelle funzioni di callback e in caso contrario visualizzerà un messaggio d'errore.

getMoviesHomepage.php

La funzione recupera dal database titolo e locandina dei primi cinque film, se ci sono, per ogni categoria. E' possibile utilizzarla tramite una richiesta GET su *'https://localhost/progetto/php/getMoviesHomepage.php'*. Output d'esempio:

```
{
  "Featured": [
    { "title": "Riverdale", "photo": "link_to_img" },
    { "title": "The100", "photo": "link_to_img" },
    { "title": "Fata Madrina Cercasi", "photo": "link_to_img" }
  ],
  "Top": [ { ... }, { ... }, ... ],
  "Favourites": [ { ... }, { ... }, ... ]
}
```

La funzione di callback lato client crea per ogni film un nuovo elemento assegnando come attributo 'alt' il titolo e come attributo 'src' il link alla locandina. Cliccando su una locandina si viene reindirizzati a *'localhost/progetto/php/moviePage.php'* corrispondente alla pagina del singolo film, passando nella query string il titolo del rispettivo film.

getAllMovies.php

La funzione restituisce la lista di tutti i film di una particolare categoria o dei film che rispettano i filtri inseriti nel caso della ricerca avanzata. Si tratta di una richiesta GET in cui viene passato nella query string il parametro che rappresenta il tipo di lista, per esempio: *'http://localhost/progetto/php/getAllMovies.php?typeL=FAVOURITES'*.

Nel caso in cui non viene passato come parametro il tipo di lista o ne viene passato uno non valido, oppure nel caso in cui non ci siano film nel risultato, viene restituito un messaggio d'errore in formato json, come:

```
{ "Error" : "Invalid request - Request type not exists." } o { "Error": "No movies in this list" }
```

Se non si tratta di una richiesta di ricerca avanzata con filtri, vengono semplicemente recuperate le informazioni dei film dal database e vengono restituite, altrimenti oltre al tipo di richiesta devono essere passati come parametri i vari filtri quali genere del film, tipo di ordinamento, anno d'uscita minimo o massimo e scelta se ricercare tra tutti i film, solo tra i preferiti o solo tra i non preferiti. Il lato client, tramite la funzione di callback, si occupa di visualizzare i risultati, mostrando all'utente nella parte sinistra la locandina e il titolo dei film e nella parte destra una lista contenente tutte le informazioni

userLogin.php, userSignup.php, userLogout.php

Queste funzioni gestiscono la registrazione, l'accesso e il logout dell'utente. Nel caso del logout bisogna effettuare una richiesta GET e il web service si occuperà, come già visto in precedenza, di terminare la sessione. La funzione di callback lato client reindirizzerà l'utente nella schermata di login. Per quanto riguarda login e registrazione è necessaria una richiesta POST, fornendo nel primo caso username e password, mentre nel secondo anche nome, cognome e email dell'utente. Successivamente alle operazioni già descritte nella sezione funzionalità, viene restituito un risultato del genere: { "userIsLogged" : true, "name" : _nomeUtenteInSession_ } o { "userIsLogged" : false }

getMovieInfo.php, getActorInfo.php

Le funzioni restituiscono rispettivamente le informazioni di un singolo film o di un singolo attore. E' possibile utilizzarle attraverso una richiesta GET passando nella query string il titolo del film o l'id dell'attore, come per esempio:

'http://localhost/progetto/php/getMovieInfo.php?movieTitle=Riverdale'

Nel caso dell'attore anziché passare come parametro nella query string l'id si possono passare nome e cognome; verrà cercato nel database e in caso di omonimi verrà selezionato il primo risultato rilevante. Entrambe le funzioni in assenza di parametri o in caso di film/attore non trovato restituiscono un errore in formato json del tipo: { "Error" : " Actor not found " }.

Output corretto d'esempio:

```
{  "Info":
    [ {
      "title":"Riverdale","photo":"link_to_img",
      "description":".....",
      "genre":"Drama",
      "averageGrades":"4.80",
      "releaseDate":"2020-12-17",
      "movieDirectorName":"Roberto",
      "movieDirectorSurname":"Sacasa",
      "music":"Nicol King",
      "country":"American",
      "distribution":"Totally Distribution",
      "production":"HollyProduction",
      "length":"129",
      "screenplay":"Antony, San"
    } ],
    "Actors":
    [
      { "actor":"CAMM","character":"Veronica","photo":"link_to_img"},
      { "actor":"CLSP","character":"Jughead","photo":"link_to_img"},
      { "actor":"LILI","character":"Betty","photo":"link_to_igm"}
    ]
}
```

movieReviews.php

La funzione è responsabile della gestione delle recensioni degli utenti. Si invoca tramite una richiesta POST fornendo il tipo di operazione e il titolo del film in questione. In caso di assenza di parametri o parametri non corretti, viene restituito, come in altri casi, un errore in formato json. I tipi di operazione sono quattro: aggiungere una recensione, cancellarne una, restituire la recensione dell'utente specifico e del film in questione, restituire tutte le recensioni del film esclusa quella dell'utente specifico. In tutti e quattro i casi, sia che l'operazione vada a buon fine che non, viene restituito un risultato del tipo:

{ "Review" : "Very good movie" } oppure { "addReview" : "True" } oppure { "deleteReview" : "False" }

Il lato client si occuperà di ricevere il risultato, verificare che l'operazione sia andata a buon fine e aggiornare eventualmente l'interfaccia, aggiungendo o rimuovendo le recensioni e creando container appositi per ogni recensione da visualizzare.

movieInteraction.php

Questa funzione, invece, serve per gestire tutte le interazioni degli utenti con i film, quali aggiunta o rimozione alle proprie liste e aggiunta o rimozione di punteggi. Anch'essa viene utilizzata tramite una richiesta POST e richiede il titolo del film e il tipo di operazione. Oltre alle operazioni menzionate precedentemente le quali si occupano di aggiornare il database e restituire un messaggio che confermi o no l'avvenuta esecuzione, possiamo trovarne una di inizializzazione: essa fornisce infatti le informazioni iniziali che il client visualizza al caricamento della pagina, ovvero il voto presente inizialmente nel database (se esiste) e se un film è già presente o meno in una delle due liste, con risultati come:

{ "rate" : "4" } oppure { "isFavourite" : "False" }