

# Elaborato “AlphaBlock”



Documentazione del sito web “AlphaBlock”



Lucca Giacomo  
ITS Leonardo Da Vinci, 5A Info

## Sommario

<b>Sommario</b>	<b>2</b>
<b>Introduzione</b>	<b>3</b>
Descrizione dell'elaborato	3
Materiale utilizzato	3
Note	3
<b>Teoria</b>	<b>4</b>
HTML	4
CSS	4
JavaScript	5
PHP	5
Ajax	6
Database e SQL	7
<b>Realizzazione</b>	<b>8</b>
Modello E / R	8
Accorgimenti presi	9
Modello logico esteso	9
Giustificazioni	12
Disposizione dei file	13
Difficoltà incontrate	13
<b>Pagine web</b>	<b>14</b>
Funzioni comuni	14
Index.php	15
Login.php e registrazione.php	16
Utente.php (e pagine correlate)	17
Home.php (e pagine correlate)	18
Film.php e guarda.php	21
Impostazioni.php (e correlati)	21
<b>Possibili aggiunte</b>	<b>23</b>
Implementazione di HTTPS	23
Implementazioni degli abbonamenti	23
Password smarrita	24

## INTRODUZIONE – Descrizione dell'elaborato

L'elaborato consiste nella realizzazione di un sito web che gestisca l'erogazione di film in streaming. Per ogni utente che desidera iscriversi su tale piattaforma, si potrà creare fino a 5 profili diversi in cui sarà possibile visualizzare diversi film. Dopo aver scelto il profilo con cui si vuole guardare i film, l'utente potrà effettuare delle ricerche per trovare il film più adatto a lui (per esempio ricercare i film attraverso il titolo, attori coinvolti, ecc..) e dovrà essere mantenuta una cronologia in base ai film che l'utente ha visionato. Per ogni film, inoltre, dovranno essere memorizzati una breve descrizione e una locandina.

## INTRODUZIONE – Materiale utilizzato

I linguaggi utilizzati per la realizzazione del sito web sono stati: PHP, JavaScript, SQL, HTML e CSS. I software utilizzati per creare il sito web sono stati principalmente 3:

- **Notepad++** per quanto riguarda la scrittura del codice delle varie pagine web dell'elaborato.
- **EasyPHP Devserver-17** utilizzato per creare un server locale per testare il funzionamento delle varie pagine. Questo programma comprende un server HTTP, in grado di processare ed elaborare il linguaggio PHP utilizzando APACHE, e un Database server, ovvero un DBMS (*DataBase Management System*), un software in grado di gestire i database. EasyPHP utilizza MySQL come DBMS.
- **Paint.net** utilizzato per la realizzazione di alcune immagini presenti nel sito web (come ad esempio le immagini degli account) e per ritoccare le varie immagini dei vari film (principalmente per il ridimensionamento di esse).

## INTRODUZIONE – Note

In questo documento verrà spiegato il funzionamento del sito web tralasciando la descrizione della parte grafica.

Il sito è stato anche stato ottimizzato per funzionare sui dispositivi con una larghezza da 600px a 920px (tablet) utilizzando le tecniche di web responsive.

Su consiglio dei prof, il sito web è stato inserito all'interno di un server di hosting su cui attualmente risiede. Come servizio di hosting è stato utilizzato **Altervista** e il sito web è raggiungibile attraverso questo link:

[alphablock.altervista.org](http://alphablock.altervista.org)

## Teoria – HTML

### HTML



HTML, ovvero *HyperText Markup Language*, è un linguaggio utilizzato per la rappresentazione, attraverso l'impaginazione e la formattazione, di documenti ipertestuali mentre la parte grafica verrà gestita dal CSS.

HTML non può considerarsi un linguaggio di programmazione essendo che non è possibile dichiarare variabili, strutture dati, funzioni o strutture di controllo che sono normalmente presenti nei vari linguaggi di programmazione. Le istruzioni che utilizza HTML si chiamano **tag** e descrivono le varie caratteristiche che dovrà possedere quella determinata informazione. I tag vengono dichiarati nel modo seguente:

`<tag> ... </tag>`

I file creati in HTML potranno essere interpretati dai vari browser disponibili sul mercato come Google Chromium (da cui si basa Google Chrome e Microsoft Edge), Mozilla Firefox, Microsoft Internet Explorer, ecc.... I file creati in HTML verranno salvati con l'estensione **.HTML** oppure **.HTM**.

## TEORIA – CSS

Il CSS, ovvero *Cascading Style Sheets*, è un linguaggio che si occupa di realizzare la parte grafica di una pagina web scritta solitamente scritta in HTML. I vari file del CSS verranno a salvarsi con l'estensione **.CSS** e, questo file può essere utilizzato in un numero infinito di pagine



HTML ed evita, quindi, di riscrivere lo stesso codice in tutte le pagine. Per utilizzare questa proprietà, in ogni pagina HTML bisognerà inserire all'interno dell'HEAD:

`<link rel="stylesheet" type="text/css" href="style.css"/>`

oppure è possibile dichiarare il CSS all'interno dei singoli tag di HTML come segue:

`<tag style=" stile in CSS "> ... </tag>`

Il foglio di stile è strutturato in *regole*. Una regola permette di assegnare la stessa impostazioni ad un insieme di elementi che possono essere classi, id oppure i singoli tag e comprende:

#### Selettore

```
{  
    proprietà : valore;  
    ...  
}
```

- **Selettore** : definisce la porzione del documento a cui verrà applicata la regola.
- **Proprietà** : definisce l'aspetta che il selettore dovrà avere.
- **Valore** : definisce il valore che la proprietà dovrà rispettare.

Per realizzare la parte di web responsive, ovvero che la pagina web varia in base alla dimensione dello schermo, si è dovuto dichiarare all'interno dell'head di HTML:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

e all'interno del file CSS:

```
@media only screen and (max-width: 950px)
```

## TEORIA – JavaScript



JavaScript è un linguaggio di programmazione orientato agli oggetti utilizzato per realizzare applicativi web lato client. L'esecuzione del codice da parte del client porta al server una serie di vantaggi tra cui la possibilità di non sovraccaricarlo di lavoro. Come punto debole ha che, se il codice da inviare al client è abbastanza grande, risulta molto lento l'invio della pagina web al client e il JS non è in grado di interagire direttamente con i Database, infatti deve delegare il compito ad altri linguaggi di programmazione (nel nostro caso il PHP) attraverso AJAX. Il codice di JavaScript non viene compilato, ma è il browser del client che lo interpreta direttamente. Il JavaScript, contrariamente a quello che fa pensare il nome, è molto più simile al linguaggio C che al Java essendo che gli oggetti (object) tra Java e JavaScript sono incompatibili tra di loro. Ci sono due modi per creare un script in JavaScript: con un file esterno con estensione **.JS** e per richiamarlo bisogna dichiarare all'interno dell'HEAD di HTML:

```
<script src="cerca.js"></script>
```

oppure all'interno della pagina HTML attraverso:

```
<script> ... </script>
```

## TEORIA – PHP

PHP, ovvero *Hypertext Preprocessor*, è un linguaggio di programmazione lato server, ovvero che soltanto il server può leggere e interpretare il codice della pagina web e, dopo averla elaborata, inviare SOLTANTO la pagina web, in pratica il client non riuscirà a visualizzare il codice PHP. Come JavaScript, la sua sintassi è molto vicina a quella del C e possiede delle specifiche API per essere implementato con APACHE ed è perfettamente integrato con MySQLI. I file con estensione **.PHP** possono contenere al suo interno tutti i tipi di linguaggi che HTML supporta nativamente. Per richiamare la sezione PHP bisogna scrivere:

```
<?php ... ?>
```





AJAX, ovvero *Asynchronous JavaScript and XML*, è una tecnica di programmazione per la realizzazione pagine web interattive dove non necessita di ricaricare ogni volta la pagina HTML rendendo la pagina dinamica. Le richieste tra il client e il server avvengono in modo asincrono, il che significa che non dobbiamo necessariamente attendere che finisca l'operazione per effettuarne altre e lo scambio di dati avviene in background. Per dichiarare AJAX all'interno di una pagina web bisognerà scrivere:

```
File  
cerca.js  
  
if (window.XMLHttpRequest)  
    ajax = new XMLHttpRequest(); //per i browser moderni  
else  
    ajax = new ActiveXObject("Microsoft.XMLHTTP"); //vecchi browser  
ajax.onreadystatechange = function()  
{  
    if (ajax.readyState == 4 && ajax.status == 200)  
        document.getElementById("risposta").innerHTML = ajax.responseText;  
}  
ajax.open("GET", "ricerca.php?stringa=" + str, true);  
ajax.send();
```

Per comunicare con il server si utilizza:

- **XMLHttpRequest**: metodo usato per inviare le richieste (si possono anche inviare se non sono scritte in XML).
- **onreadystatechange**: metodo che funziona soltanto al cambiamento della proprietà *readyState*.
- **readyState**: restituisce lo stato corrente dell'istanza di *XMLHttpRequest* (il 4 sta per ricevuto).
- **status**: restituisce il codice HTTP dal server (200 sta per "OK" e 404 sta per "Not Found").
- **open**: specifica il metodo (se inviare le richieste con il GET, generalmente usato per l'invio dei dati, o con il POST, generalmente usato per ricevere i dati), URL e altri parametri opzionali per la richiesta.
- **send** serve per inviare le richieste.

Per richiamare la funzione, si è utilizzato, all'interno di un campo input, *onkeyup*:

```
<input type="search" id='cerca' placeholder='Cerca' onkeyup="mostra(this.value)"/>
```

dove viene richiamata la funzione *mostra* ogni volta che viene inserita una lettera nel campo input.

I Database è una raccolta organizzata di dati. Sono utilizzati per memorizzare grosse quantità di dati e rende possibile la loro ricerca attraverso le interrogazioni (le query). I Database vengono gestiti da appositi software, ovvero i DBMS (*DataBase Management System*).



Per interagire con i Database si utilizza SQL, *Structure Query Language*, ed è composta da:

- **DDL (*Data Definition Language*)**: che permette di creare e modificare la struttura di una tabella.
- **DML (*Data Manipulation Language*)**: che permette di inserire, modificare e cancellare dati all'interno del Database.
- **QL (*Query Language*)**: che permette di fare interrogazione all'interno del Database.
- **DCL (*Data Control Language*)**: che permette di revocare e fornire permessi ai vari utenti su che cosa possono fare all'interno del Database.

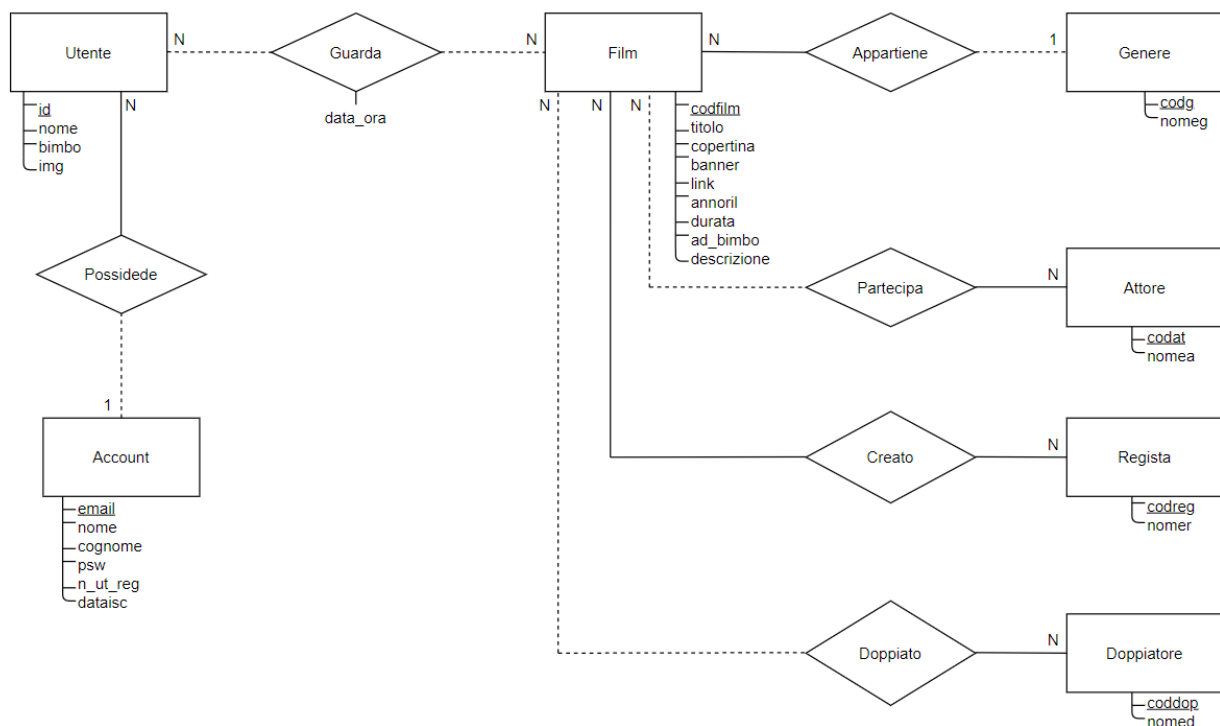
Non esiste una sintassi uguale di SQL, infatti ogni Database possiede la sua sintassi.

Per realizzare un Database occorre definire:

1. **Livello concettuale**: che rappresenta la realtà dei dati e le relazioni tra essi. Per fare questo livello si utilizza il modello E / R (Entità / Associazioni).
2. **Livello logico**: che descrive come sono organizzati i vari dati all'interno dei diversi archivi. Si realizza derivando il modello E / R attraverso delle semplici regole: ogni entità diventa una tabella, l'attributo univoco diventa chiave primaria, ogni attributo eredita le caratteristiche dell'attributo da cui deriva e i vari modi per la derivazione delle varie associazioni (1 a 1, 1 a molti, molti a molti).
3. **Livello fisico**: che rappresenta la reale installazione del database sul disco.

## Realizzazione – Modello E / R

Per la realizzazione dell'elaborato, bisogna creare un Database dove è possibile registrare un utente (account), a cui verrà affiancato fino ad un massimo di 5 profili diversi (utenti) e che verranno salvati i vari film che l'utente ha guardato, ovvero una cronologia. Inoltre il film dovrà possedere: gli attori con cui è stato registrato, eventuali doppiatore, i registi e il genere a cui appartiene. Per semplificare il lavoro è stato scelto di inserire soltanto film all'interno del sito, escludendo le serie TV. Di conseguenza, il modello E / R del sito web sarà:



Dove:

- **n\_ut\_reg** (presente in account): è un attributo che terrà conto di quanti profili, ovvero utente, l'account possiede.
- **psw** (in account): conterrà la password dell'account criptata con una funzione hash, nell'esattezza in md5.
- **bimbo** (in utente): è il parental control. Dirà se l'utente ha la possibilità di guardare i contenuti adulti o meno ed è una variabile boolean (0 adulto e 1 se è un bambino). Di default è impostata a 0, ovvero adulto.
- **img** (in utente): contiene il numero dell'immagine che l'utente possiede.
- **banner** (in film): è un immagine più grande della copertina utilizzata per la realizzazione del banner pubblicitario nella home del sito.
- **ad\_bimbo** (in film): variabile che dice se il film è disponibile alla visione da tutti (valore 0) oppure soltanto gli account "per adulti" possono vedere quel



determinato film (valore 1). Come la variabile bimbo contenuto in utente, è una variabile di tipo boolean con valore, di default, 0.

- **link** (in film): contiene l'ultima parte del link di YouTube a cui si riferisce il film.

## Realizzazione – Accorgimenti presi

Per semplicità nella realizzazione della pagina web:

- I vari campi dove è contenuto nome, legati alla tabella film (nomed, nomea e nomer) sarà contenuto sia il nome che il cognome dei vari doppiatori, registi e attori. Questo per rendere più completa la ricerca che verrà effettuata su tali campi e rendere più facile la realizzazione del codice della ricerca.
- I link dei film sono soltanto i link dei vari trailer presi da YouTube.
- Gli attributo del Database che contengono immagini (banner e copertina per i film) sono contenuti soltanto la posizione in cui risiedono le varie immagini partendo dalla cartella Root ("quella più in alto") del sito web. Di fatto, le immagini non vengono salvate all'interno del Database.
- Si è scelto di rendere il servizio gratuito. Questa scelta è stata presa principalmente per evitare di salvare i dati delle varie carte di credito all'interno di una tabella che porterebbe soltanto problemi legati alla sicurezza.
- Si è scelto, anche, di omettere i vari ruoli che i vari attori hanno avuto all'interno del film. Questo accorgimento è stato preso per rendere più veloce l'inserimento dei vari attori che hanno girato quel determinato film.
- I film sono riconducibili a soltanto un genere.
- Le immagine dei profili degli utenti sono scelte automaticamente dal sistema in base.

## Realizzazione – Modello logico esteso

*Account*

Campo	Tipo	Chiave
email	char(100)	Primary key
psw	char(32)	
nome	char(20)	
cognome	char(20)	
n_ut_reg	smallint(6)	
dataisc	timestamp	

### *Utente*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
id	int(11)	Primary key
email	char(50)	Foreign key
nome	char(20)	
bimbo	boolean	
img	smallint(6)	

### *Guarda*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
id	int(11)	Primary key + Foreign key
codfilm	int(11)	Primary key + Foreign key
data_ora	timestamp	Primary key + Foreign key

### *Film*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
codfilm	int(11)	Primary key
titolo	char(100)	
copertina	char(100)	
banner	char(100)	
link	char(15)	
annoril	year	
durata	time	
ad_bimbo	boolean	
descrizione	varchar(520)	
codg	int(11)	Foreign key

### *Genere*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
codg	int(11)	Primary key
nomeg	char(25)	

*Attore*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
codat	int(11)	Primary key
nomea	char(100)	

*Partecipa*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
codat	int(11)	Primary key + Foreign key
codfilm	int(11)	Primary key + Foreign key

*Regista*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
codreg	int(11)	Primary key
nomer	char(100)	

*Creato*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
codreg	int(11)	Primary key + Foreign key
codfilm	int(11)	Primary key + Foreign key

*Doppiatore*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
coddop	int(11)	Primary key
nomed	char(100)	

*Doppiato*

<b>Campo</b>	<b>Tipo</b>	<b>Chiave</b>
coddop	int(11)	Primary key + Foreign key
codfilm	int(11)	Primary key + Foreign key

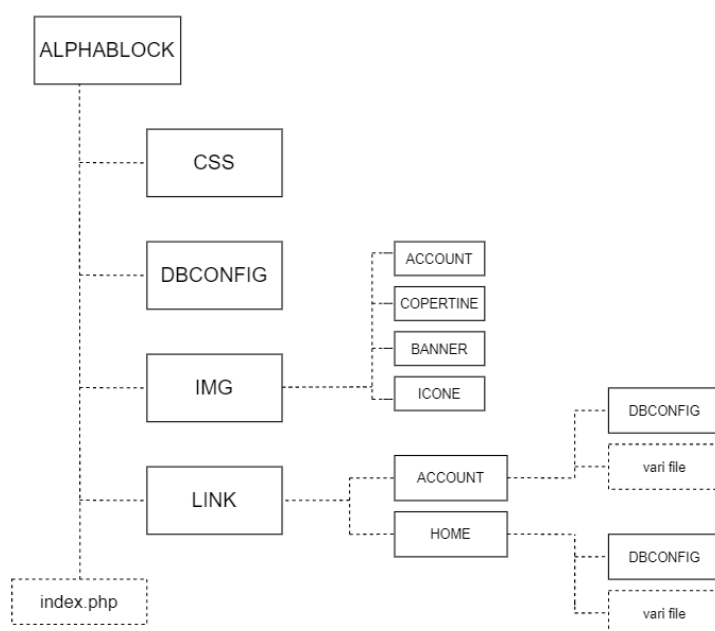
- **Account e Utente - 1 a molti:** Un account può avere più utente (infatti può averne fino a 5 massimo), e un utente può essere posseduto soltanto da un account.  
*Da Account a Utente facoltativo:* un account può non avere nessun profilo collegato in un momento specifico (questa condizione viene realizzata soltanto se si cancellano tutti gli utenti di uno specifico account).  
*Da Utente ad Account obbligatorio:* un utente, se registrato nel Database, è associato per forza ad un solo account.
- **Utente e Film - molti a molti:** Un utente può guardare più film presenti sulla piattaforma e un film può essere visto da più persone.  
*Da Utente a Film facoltativo:* un utente può essere registrato nel Database ma può non avere ancora visto nessun film (per esempio quando si crea il profilo).  
*Da Film a Utente facoltativo:* un film potrebbe non essere mai stato visto da nessun utente (per esempio se il film è appena stato inserito).
- **Film e Genere - 1 a Molti:** Un film appartiene soltanto ad un genere e ad un genere possono essere associati più film.  
*Da Film a Genere obbligatorio:* un film deve appartenere ad un genere.  
*Da Genere a Film facoltativo:* ad un genere possono non essere associati a nessun film.
- **Film e Attore - molti a molti:** In un film hanno recitato più attori e un attore ha recitato in più film nel corso della sua carriera.  
*Da Film ad Attore facoltativo:* possono esserci dei film in cui non sono presenti attori (come ad esempio i film da animazione).  
*Da Attore a Film obbligatorio:* sono presenti all'interno del Database soltanto attori che hanno partecipato ad almeno un film.
- **Film e Regista – molti a molti:** Un film può essere stato composto da diversi registi e un regista può aver diretto diversi film nel corso della sua carriera.  
*Da Film a Regista obbligatorio:* Un film è stato almeno diretto da un regista.  
*Da Regista a Film obbligatorio:* nel Database sono solo presenti soltanto registi che hanno diretto almeno un film.
- **Film e Doppiatore - molti a molti:** In un film possono essere presenti più doppiatori che hanno doppiato un film e un doppiatore può aver doppiato più film durante tutta la sua carriera.  
*Da Film a Doppiatore facoltativo:* possono esserci dei film in cui non necessitano doppiatori (per esempio se si trattano di film che sono stati prodotti su suolo nazionale).

*Da Doppiatore a Film obbligatorio*: sono presenti nel Database soltanto doppiatori che hanno doppiato film.

## Realizzazione – Disposizioni dei file

Essendo che il sito web è composto da 30 file (3 file CSS, 1 file HTML e 26 file PHP) e 56 immagini diverse, si è reso necessario organizzare il sito in diverse cartelle:

- **CSS**: contiene tutti i file css che si sono utilizzati nelle varie pagine.
- **DBCONFIG**: contiene i vari file contenenti le varie funzioni per la realizzazione della connessione con il Database.
- **IMG**: suddivisa in 4 sotto cartelle che contengono le varie icone del sito web e sfondi (icone), le immagini per gli utenti (account), le varie copertine dei film (copertine) e banner pubblicitari (banner).
- **LINK**: contiene le varie pagine della pagina web suddivise per la loro funzione, ovvero se interagiscono con la tabella degli utenti (film) oppure se interagisce con la tabella film (film). In ogni sezione è contenuta la cartella DBCONFIG.



## Realizzazione – Difficoltà incontrate

- **Realizzazione del Banner (*home.php*)**: all'inizio si voleva realizzare un banner pubblicitario con delle frecce che permettevano di cambiare il film. Non è stato possibile implementare tale funzione perché, a volte, saltava qualche film e non permetteva di cambiare film. Quindi si è optato per togliere queste frecce e che i film cambiassero da soli attraverso un timer.
- **Disposizione dei film (*home.php*)**: all'inizio si voleva realizzare una barra orizzontale (scrollbar) in cui fossero contenuti i vari film. Non è stato possibile implementare tale funzione e, i vari film, verranno visualizzati su diverse righe.

## Pagine web – Funzioni comuni

Ogni pagina PHP contiene questa funzione:

```
session_start();
if(isset($_SESSION['data'])) {
    $data=date('Y-m-d H:i:s');
    if($_SESSION['data']<$data) {
        session_destroy();
        header("Location:login.php");
    }else
        $_SESSION['data']=date('Y-m-dH:i:s',strtotime('+1
hour',strtotime(date ('Y-m-d H:i:s'))));
}
```

che, dopo aver avviato la sessione, controlla se l'utente, se non ha inserito l'opzione nella pagina di login di rimanere connesso al sito, ha eseguito operazioni (cambiare pagina o aggiornarla) entro 1 ora. Se non ha effettuato operazioni entro un ora, verrà distrutta la sessione, facendo un logout forzato all'utente, altrimenti si aggiorna la variabile di sessione contenente l'ora.

Altre funzioni che sono state utilizzate molto nelle varie pagine web sono le interrogazioni al database che vengono eseguite con:

```
include "dbconfig/dbopen.php";
$query = " query ";
$tabella=mysqli_query($dbconn,$query) or die(mysqli_error($dbconn));
while($riga=mysqli_fetch_array($tabella))
{ ... }
include "dbconfig/dbclose.php";
```

dove:

1. Viene incluso il file necessario per aprire la connessione con il database.
2. Viene creata la query.
3. Viene eseguita la query.
4. Nel *while* viene salvato in un vettore i vari risultati che restituirà la query.
5. All'interno del *while*, si stamperanno, per esempio, i vari risultati.
6. L'ultima istruzione servirà a invocare il file addetto alla chiusura della connessione con il database.

Le istruzioni all'interno di questi file sono:

*dbopen.php:*

```
include "dbconfig/dbconfig.php";  
$dbconn=mysqli_connect($dbhost,$dbuser,$dbpass) or die ("ERRORE  
connessione");  
mysqli_select_db($dbconn,$dbnome) or die ("NON ESISTE IL  
DATABASE");
```

All'interno del file *dbconfig.php* sono contenute le informazioni riguardanti il database e come accedervi (indirizzo IP, account, password, nome database).

*dbclose.php:*

```
mysqli_close($dbconn);
```

Questo file serve per chiudere la connessione con il database.


## Pagine web – Index.php

Pagina iniziale da cui si accede al sito web. Viene mostrato un'introduzione alla piattaforma, cosa offre, su quali dispositivi può funzionare e due pulsanti che consentono la registrazione o l'autenticazione da parte dell'utente.

In questa pagina verrà anche mostrato un film a caso presente all'interno del sito. Si legge quanti film sono presenti all'interno della tabella film (supponendo che la chiave primaria, *codfilm*, non contenga indici vuoti) e verranno salvate le informazioni che servono per stampare a video il film. Viene

fatto anche un controllo che permette di stampare la durata del film. Le informazioni aggiuntive, ovvero attori, registi e doppiatori, vengono ricercate attraverso delle query (ognuna per le varie informazioni ricercate). Se non sono presenti alcune informazioni relative soltanto ai vari attori o doppiatori, controllando che le righe che restituisce la query siano maggiori di zero, non verranno visualizzate tali informazioni. Il controllo viene fatto con la seguente funzione:

**Cosa potresti vedere su questo sito**



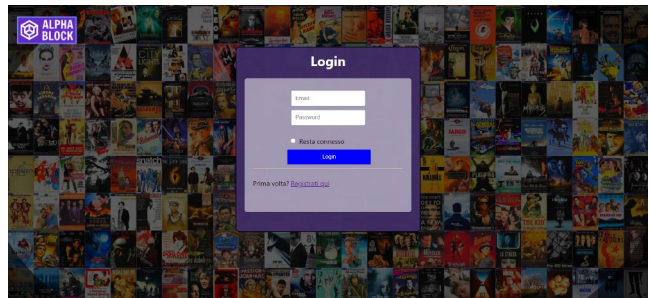
**Il gladiatore**  
Designato futuro imperatore da un morente Marco Aurelio, il generale Massimo viene ridotto in schiavitù e la sua famiglia uccisa ad opera di Commodus, figlio dell'anziano sovrano. Inizierà a combattere come gladiatore nelle arene delle province romane fin quando la sua fama non lo ricondurrà nella capitale, dove sfiderà l'imperatore in persona per ottenere vendetta.  
**Durata del film:** 2 ore e 35 minuti  
**Genere:** Drammatico  
**Regista:** Ridley Scott  
**Attori:** Russell Crowe, Joaquin Phoenix, Connie Nielsen, Oliver Reed  
**Doppiatori:** Luca Ward, Francesco Bulckaen, Chiara Salerno, Glauco Onorato

```
$num=mysqli_num_rows($tabella);  
if($num!=0)  
{ ... }
```

Questa pagina possiede anche un ulteriore controllo che, se l'utente si è già autenticato, verrà rimandato, in automatico, alla pagina *utente.php*, che permette di scegliere l'utente con il quale si vuole vedere i vari film. Questa funzione è presente anche nelle pagine web *login.php* e *registrazione.php*.

## Pagine web – Login.php e registrazione.php

Queste pagine consentono l'autenticazione da parte dell'utente e la registrazione se non si possiede l'account. Se l'utente commette degli errori durante il login o la registrazione, verrà mostrato un errore dicendo che non è stato possibile registrarsi o autenticarsi all'interno del sito.



Per evitare interrogazioni o manipolazione del database, attraverso la tecnica del *SQL Injection*, si utilizza questa funzione:

```
mysqli_real_escape_string($dbconn, $_POST['conferma']);
```

Funzione che serve a togliere gli apici alla stringa in input. Questo perché la stringa potrebbero contenere query al suo interno. Se non si utilizzasse questa funzione, durante l'inserimento dei vari campi all'interno della tabella, il database interpreta quella stringa come se fosse una query e la esegue portando a esiti anche catastrofici (per esempio un utente malevolo potrebbe decidere di cancellare una determinata tabella del database, perdendo tutte le informazioni in esso contenuto). Dopo essersi autenticato, si compongono le varie variabili di sessione che verranno utilizzate nelle varie pagine web successivi e si viene mandati alla pagina dedicata alla selezione dell'utente. Vengono composte, come variabili di sessione:

- **\$\_SESSION['utente']**: che contiene l'email dell'utente, utilizzata per ricercare i vari utenti registrati sotto quell'account.
- **\$\_SESSION['data']**: attiva solamente nel caso in cui non si seleziona la spunta per rimanere autenticati permanentemente al sito. Verrà salvata l'ora del login aggiungendo un ora in più che servirà a determinare il limite di tempo entro il quale l'utente può effettuare operazioni prima di essere "espulso" da sistema, facendo un logout forzato.



## Pagine web – Utente.php (e pagine correlate)

Nella pagina *utente.php* è possibile selezionare, creare e modificare l'utente con cui si vuole visionare il sito. In *utente.php* si possono eseguire 3 diverse operazioni:

- **Aggiungere un nuovo utente:** nel sito web è

possibile creare fino a 5 utenti diversi. Se la pagina non trova 5 utenti presenti su quel determinato account, verrà mostrato un'immagine che, se cliccata, rimanda alla pagina *aggiungi.php* che permette la registrazione di un nuovo utente solamente se il campo *n\_ut\_reg* (contenuto in *account*) sia minore di 5, altrimenti se si tenterà di aggiungere un profilo in più, farà visualizzare un errore e non gli permetterà di inserirne uno nuovo. Le immagini dei profili, sono scelte in automatico dal sistema, cercando il primo numero mancante dell'immagine (essendo numerate da 1 a 5) che non è presente nella tabella *utente* associato a quel determinato account con la seguente funzione:



```
$riga[0]=0;  
... //interrogazione del database  
$riga[$i]=$riga[$i-1]+1;  
for($j=0;$j<$i;$j++)  
    if($riga[$j]+1!=$riga[$j+1])  
        $img=$riga[$j]+1;  
if(!isset($img))  
    $img=$riga[$j];
```

Successivamente permette l'inserimento del nome dell'utente e, se l'utente verrà utilizzato da un minore, è possibile attivare il parental controll che servirà per controllare se quel determinato film è adatto da vedere a quell'utente. Dopo l'inserimento, i dati verranno salvati nella tabella *utente* e verrà aggiornata la variabile *n\_ut\_reg*, aggiungendo 1.

- **Modificare l'utente:** per eseguire questa operazione è necessario cliccare sul pulsante *modifica* (posizionato sotto gli utenti) che, cambierà URL e immagine tramite una funzione in JavaScript. Per cambiare l'immagine e gli URL si utilizza:

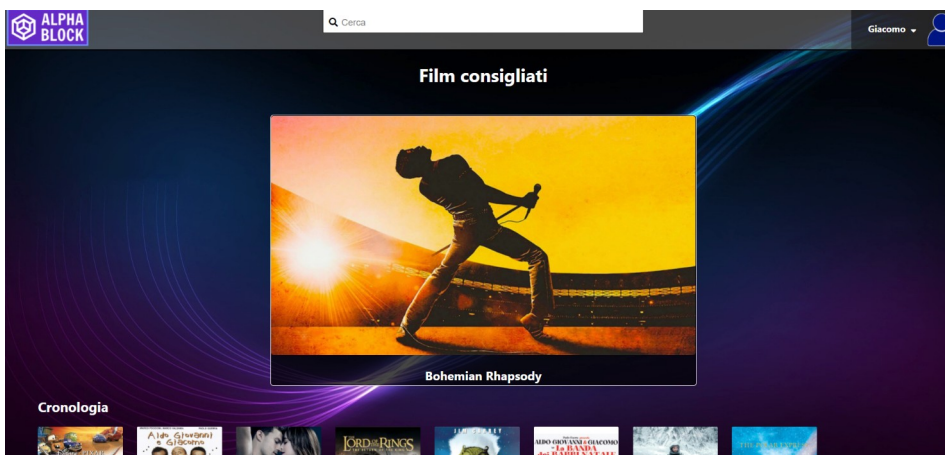
```
document.getElementById( ".$matrice[$j][0]." ).src='../img/account/'.  
$matrice[$j][3]."-modifica.png";  
document.getElementById( "rect".$matrice[$j][0]." ").setAttribute('href',  
'modifica.php?id=".$matrice[$j][0]." ');
```

La prima funzione, permette di modificare l'immagine dell'utente e la seconda serve a cambiare il link che porterà alla pagina dedicata alla modifica dell'utente.

Cliccando sempre sull'immagine profilo della modifica (capibile dal simbolo della matita in sovrapposizione sull'immagine) si verrà rimandati alla pagina *modifica.php* che permetterà di cancellare l'utente (se si clicca l'apposita funzione si verrà rimandati a *rimuovi.php* che cancellerà l'utente, aggiornerà la variabile *n\_ut\_reg*, presente in *account*, e verrà cancellata l'eventuale cronologia dei film che ha visionato) o di aggiornare il nome dell'utente.

- **Scegliere l'utente:** nella pagina *utente.php* è possibile scegliere l'utente con cui visionare i vari film cliccando sull'immagine dell'utente. Si verrà rimandati alla pagina *aut.php* che salverà, all'interno di una variabile di sessione, *\$\_SESSION['id']*, l'id dell'utente e servirà per: visualizzare la sua cronologia di film, e salvare i vari film che visualizza, e per determinare se l'utente può vedere i film non adatti ai minori. Successivamente si verrà automaticamente spostati sulla pagina *home.php*.

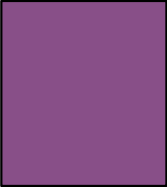
## Pagine web – Home.php (e pagine correlate)



Questa pagina web è la pagina principale di tutto il sito web dove è possibile eseguire la maggior parte di funzioni che sono disponibili all'interno del sito web. Sono possibili eseguire:


- **Ricerca i vari film:** usata anche in altre pagine del sito web. La ricerca di vari film viene eseguita da AJAX, utilizzando i *Google suggest*. Ogni volta che si vuole cercare un film attraverso diversi parametri (titolo del film, un determinato attore, regista e doppiatore che hanno realizzato il film e un genere) si dovrà utilizzare la barra di ricerca posizionata in cima alla pagina. Il campo di input contiene uno speciale metodo, *onkeyup*, che, ogni volta che

digitiamo sulla tastiera una lettera, verrà richiamata la funzione (passandogli anche la lettera che abbiamo digitato) *mostra(this.value)*, metodo contenuto nel file *cerca.js*. Questo file serve per invocare l'utilizzo dell'AJAX e, la ricerca vera e propria, verrà effettuata all'interno del file *ricerca.php*. Questa pagina, leggendo in input la stringa che la pagina *cerca.js* gli ha inviato attraverso un GET, ricercherà, all'interno delle varie tabelle, l'informazione voluta e verrà restituita alla pagina in cui si è stata chiamata la funzione (in questo caso *home.php*) un link che conterrà le informazioni necessarie affinché si possano visualizzare i vari film che rispecchino quella determinata condizione, ovvero i vari codici che funzionano da chiave primaria (coddop, codat, ...). Durante la ricerca dei vari attori e film si potrebbe incorrere ad attori che potrebbero aver fatto da registi in vari film (come, ad esempio, Aldo Baglio). In questo caso, durante la ricerca degli attori e dei registi, verranno esclusi i registi che sono inclusi nella tabella degli attori come segue:




```
SELECT codreg,nomer FROM regista
WHERE nomer like '%$q%'
and nomer not in(select nomea
                  from attore WHERE nomea like '%$q%')
```

Query che esclude i registi che sono contenuti nella tabella attori:



```
SELECT codat,nomea FROM attore
WHERE nomea like '%$q%' and nomea not in (select nomer
                                          from regista WHERE nomer like '%$q%')
```

Gli attori, che hanno fatto anche da registi, verranno ricercati con la seguente query:



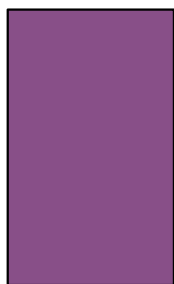
```
SELECT codat,nomea FROM attore
WHERE nomea like '%$q%' and nomea in (select nomer
                                       from regista WHERE nomer like '%$q%')
```

Successivamente, nella pagina *cerca.php*, verranno a visualizzati i film che rispettino le condizioni date dalla ricerca oppure reindirizzerà direttamente alla pagina del film desiderato se si è inserito solo il titolo. Per quanto riguarda gli attori e registi in comune, verranno a visualizzarsi tutti i film in cui gli attori sono presenti e in quelli in cui hanno diretto.

- **“Banner pubblicitario”**: come avviene in molti siti web di streaming, è presente una funzione che permette di visualizzare i film che il sito ti sta

consigliando in quel momento. Verranno visualizzati 5 film differente presi a caso dalla tabella film con la possibilità di cliccarci sopra (che porterà direttamente alla pagina del film). La funzione è scritta in JavaScript che verrà a sua volta composta dal PHP. Verrà visualizzato un film alla volta e, ogni 5 secondi, i film cambiano. Se l'utente possiede il parental controll, verranno in automatici esclusi tutti i film vietati ai minori (ovvero con `ad_bimbo = 1`).

- **Cronologia:** funzione che permette di vedere gli ultimi 8 film in ordine temporale, da quello visto più di recente a quello visto meno recente. In questa funzione verranno prelevati i singoli film che si sono visti e viene ricercata la data più recente in cui sono stati visti. Il risultato di queste 2 query, viene salvato in una matrice e, successivamente, si esegue un algoritmo per riordinare i film visti più recentemente perché, se non si facesse, i film sarebbero ordinato a caso all'interno della matrice. Per eseguire questo ordinamento, si è reso necessario realizzare un algoritmo di Bubble Sort ed è il seguente:



```
for($j=0;$j<$i;$j++)  
    for($k=0;$k<$i-1;$k++){  
        if($matrice[$k][3]<$matrice[$k+1][3]){  
            $temp=$matrice[$k];  
            $matrice[$k]=$matrice[$k+1];  
            $matrice[$k+1]=$temp; } }
```

Successivamente verranno visualizzati soltanto i primi 8 film che si sono guardati.

- **Film:** verranno visualizzati soltanto 5 generi presi a caso dalla tabella generi e, ognuno di questi genere, possederà al massimo 20 diversi film. I film che non sono adatti ai minori (visibili con il valore 1 nella variabile `ad_bimbo`) non verranno visualizzati e non verranno conteggiati nei 20 film. Se, per colpa del parental control, i generi che si vengono a mostrare sono meno di 5 (in questo caso il genere Horror che contiene solo 1 film), si verranno a mostrare forzatamente 5 generi ripartendo dal primo genere che è contenuto in un vettore.
- **Menù a tendina:** in questo menù è contenuto: Il nome dell'utente che si sta utilizzando in quel momento con l'immagine dell'utente, la possibilità di cambiare utente, di accedere alle impostazioni dell'account e la possibilità di uscire dall'account rimandando alla pagina *logout.php* dove verrà distrutto la session, con annessa le sue variabili, e si verrà rimandati alla pagina *login.php*.

## Pagine web – Film.php e guarda.php

In questa pagina è possibile vedere le informazioni del film scelto e la possibilità di guardare il film. Le informazioni del film che vengono a visualizzarsi sono: Titolo, una breve descrizione, i registi, gli eventuali attori e gli eventuali doppiatori del film. È presente un tasto che permette di tornare alla *home* del sito e un tasto che permette di visualizzare il film, aprendo un'altra pagina del browser, attraverso la pagina *guarda.php*. In entrambe le pagine verrà fatto un controllo se il film può essere visto dall'utente.



Altrimenti nella pagina *film.php* verrà visualizzato un errore e non sarà possibile visualizzare le informazioni sul film e nella pagina *guarda.php* si verrà immediatamente rimandati alla pagina *home.php* e non verrà salvato il film nella cronologia dell'utente. Se è possibile vedere il film, in *guarda.php*, verrà salvato nella cronologia, il film che si è visto insieme all'ora in cui è stato visto.

In *film.php* verrà anche visualizzati i correlati dei film di quel genere che sono possibili da vedere escludendo i film non adatti ai minori (se il parental control è attivo) e verranno visualizzati i primi 20 film di quel genere escludendo il film che si è scelto. Se non è presente nessun altro film di quel genere, verrà visualizzato niente. La sezione di codice che permette la visualizzazione del film selezionati (in *film.php*) è la medesima utilizzata in *index.php* e per i correlati la funzione, leggermente modificata, presente in *home.php*.

## Pagine web – Impostazioni.php (e correlati)



In questa pagina è possibile:

- **Visualizzare la cronologia:** di tutti i film visti dagli utenti insieme alla data in cui è stato guardato.
- **Informazioni dell'account:** verranno visualizzate i vari utenti che sono stati registrati sotto quel determinato

account, l'email utilizzata per registrarsi, la data di iscrizione e il nome e cognome dell'account.

- **Possibilità di modifica:** si può modificare l'email associata all'account nell'apposita pagina (*email.php*), la password (*password.php*) e si può anche cancellare l'account (*cancella.php*) dove è necessario confermare la cancellazione dell'account attraverso l'inserimento della password. Successivamente si procederà alla cancellazione degli utenti di quell'account e la cronologia di tutti gli utenti sotto quel determinato utente. Per la cronologia è stato necessario utilizzare, per la cancellazione:



```
DELETE guarda.* FROM utente INNER JOIN guarda
ON utente.id=guarda.id ON email=' ".$_SESSION['utente']."' '
```

Questa query permette di eliminare tutto il contenuto di *guarda* che ha come id l'id dell'utente che verrà cancellato successivamente. In pratica, *INNER JOIN*, permette di eseguire una congiunzione con la funzione *DELETE* che, nativamente, non supporta la congiunzione tra due diverse tabelle diverse.

- **Menù a tendina:** queste pagine, ha differenza delle altre pagine web, possiedono: un pulsante che consente di tornare a *home.php*, un tasto che consente di cambiare utente (*utente.php*) e la possibilità di fare il logout (*logout.php*) dal proprio account.



## Possibili aggiunte – Implementazione di HTTPS



HTTPS (*HyperText Transfer Protocol over Secure Socket Layer*) è un protocollo utilizzato per il trasferimento delle pagine web in modo sicuro, attraverso l'implementazione di SSL e dei certificati

digitali. Questo protocollo garantisce, al contrario di HTTP:

- **Autenticazione** da parte del client e del server. Serve per evitare attacchi hacker del tipo *man-in-the-middle*.
- **Riservatezza** del messaggio perché viene criptato con un algoritmo a chiave asimmetrica (chiave pubblica / chiave privata).
- **Integrità dei dati** garantendo che, i messaggi inviati, siano ricevuti in modo corretto, senza errori.

SSL (*Secure Socket Layer*), o la più recente versione TLS (*Transport Layer Secure*), è un protocollo che garantisce una connessione sicura tra il server e il client che utilizzano reti TCP / IP validando il certificato digitale che possiede il server. Fornisce: autenticazione (tra client e server), riservatezza del messaggio e integrità dei dati evitando, di conseguenza, la manomissione dei dati.

Un Certificato Digitale è un certificato rilasciato da un autorità competente (*Certification Authority*) che serve a garantire la validità e la proprietà della chiave pubblica del server. Questo certificato contiene diverse informazioni riguardanti le informazioni sul titolare della chiave pubblica, la chiave pubblica, la validità del certificato, il nome della CA che ha certificato la chiave e la firma della CA.

## Possibili aggiunte – Implementazione degli abbonamenti

Un'altra possibile aggiunta che si potrebbe inserire, all'interno della pagina web, è quella di prevedere un piano di abbonamento che permetterebbe la fruizione della piattaforma. Per aggiungere gli abbonamenti bisognerà creare un database apposito che memorizzi le informazioni delle carte di credito utilizzate per pagare

l'abbonamento. Questa tabella dovrà contenere i dati criptati delle carte di credito, in questo modo se un hacker tenterà di entrare nel sistema, non potrà immediatamente utilizzare questi numeri. Ma questa implementazione porterà a diversi problemi nella sicurezza della piattaforma tra cui il problema di dove salvare la chiave di criptazione dei dati. Per ovviare a questo problema si potrebbe omettere la memorizzazione delle carte di credito (non inserendo una tabella che memorizzi questi dati) e delegare, ad servizio esterno, il pagamento degli abbonamenti (per esempio delegare il compito a PayPal). Indipendentemente da



quale metodo si utilizzi, bisognerà implementare, all'interno della tabella *account*, un campo timestamp che salvi l'ultima data di pagamento dell'abbonamento. Successivamente, in ogni pagina, bisognerà fare un controllo sulla data pagata e sulla data massima in cui l'account può usufruire del servizio senza la necessità di pagare. Se si supererà tale data, bisognerà reindirizzare alla pagina per pagare l'abbonamento.

### Possibili aggiunte – Password smarrita



Un ulteriore aggiunta potrebbe essere recuperare la password di un account se non se la ricorda. Per fare questo procedimento bisognerà generare una password e bisognerà aggiornare la password dell'account con quella creata (in md5) nel database e spedirla all'email dell'account. Si potrebbe anche implementare che, se la password dell'account viene recuperata e memorizzare che tale operazione è stata eseguita tramite un flag, al primo accesso dell'account forzerà l'utente a cambiare password.