

Polytechnic University of Cartagena



Higher Technical School of Telecommunications  
Engineering

## **PRACTICAL APPLICATIONS ON THE INTERNET**

**Final project 2020/2021**

**Web application for movie rating**

Student:  
Matea Mijocevic

Teachers:  
Esteban Egea Lopez  
Juan José Alcaraz Espín

**Table of Contents**

1. Introduction.....3

2. Features.....3

3. Implementation.....6

    3.1. Configuration file.....7

    3.2. Navigation bar and style.....7

    3.3. Login, logout, register and change password.....7

    3.4. Adding and changing reviews.....8

# 1. Introduction

This document describes the application made for movie reviews and its implementation. Languages used for developing it are HTML, CSS, PHP and JavaScript.

## 2. Features

For unregistered users, there is a front page where they can see list of movies and their review calculated as average review out of all reviews that DB contains.

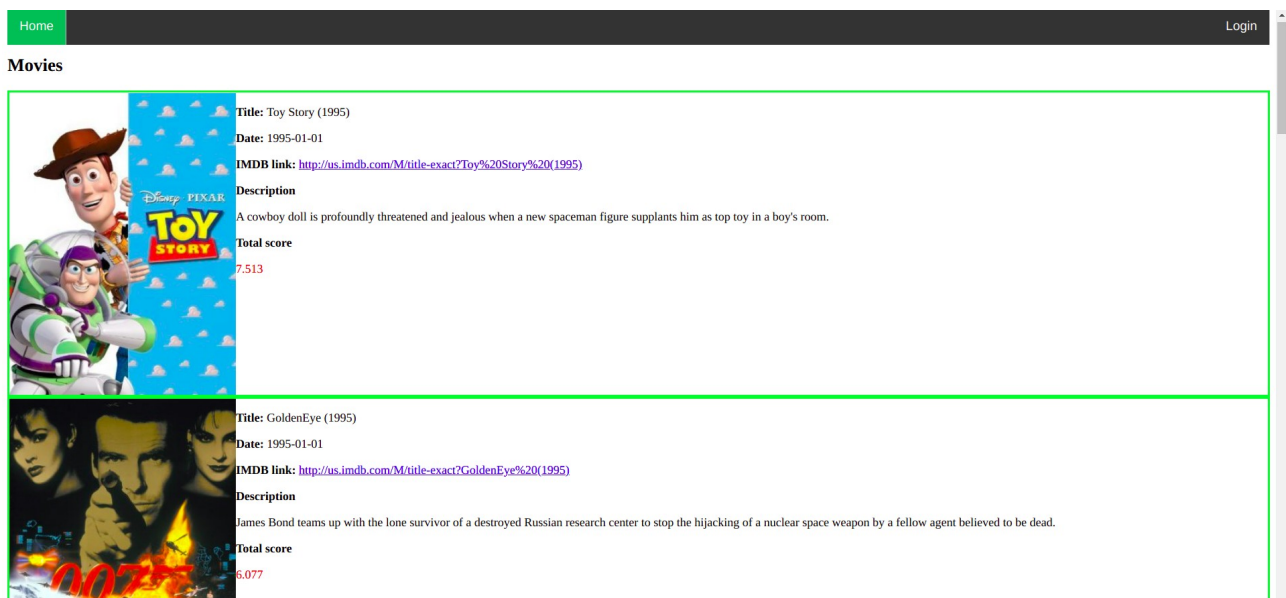


Figure 1: Front page for unregistered users

By clicking on *Login*, user can or login if it has an account, or register.

The screenshot shows the login page of the application. At the top, there is a dark navigation bar with a green 'Home' button on the left and a 'Login' button on the right. Below the navigation bar, the word 'Login' is displayed in a large font. Underneath, there is a prompt: 'Please fill in your credentials to login.' followed by two input fields: 'Username' with the value 'alumno' and 'Password' with a masked value '.....'. Below the password field is a blue 'Login' button. At the bottom of the form, there is a blue link that says 'Sign up'.

Figure 2: Login page

If user wants to register, he needs to write his username, password and repeat password. If either of these fields are missing, or the username is taken, there will be written error below these fields. Sex and Occupation are enum, and Age can be only number > 0. After registering, new user is added to DB, and user view is redirected to login page.

The screenshot shows a web application's 'Sign Up' page. At the top, there is a navigation bar with a green 'Home' button and a dark grey 'Login' button. The main heading is 'Sign Up' in a large, bold font. Below it, a subtitle reads 'Please fill this form to create an account.' The form contains several fields: 'Username' with the value 'alumno' and a red error message 'This username is already taken.' below it; 'Age' with an empty text input; 'Sex' with a dropdown menu showing 'M'; 'Occupation' with a dropdown menu showing 'administrator'; 'Password' with a masked input (dots); and 'Confirm Password' with an empty text input and a red error message 'Please confirm password.' below it. At the bottom of the form are two buttons: a blue 'Submit' button and a white 'Reset' button. Below the buttons is a link that says 'Login.'.

*Figure 3: Register page after trying to register with non-valid data*

After login, front page for user is changed(file is the same but the view is different). He can see his review(if exists) and a button for changing or adding new review for every movie. His navigation bar is also different. Now he can go on a page with only his reviews, he can go on a page where are details about his profile and where he can change his password, and also, there is logout button that logs him out and redirects on front page, now with a view for non registered users. Also, if user shrinks the window, navigation bar will change its look (see *Figure 5.*)

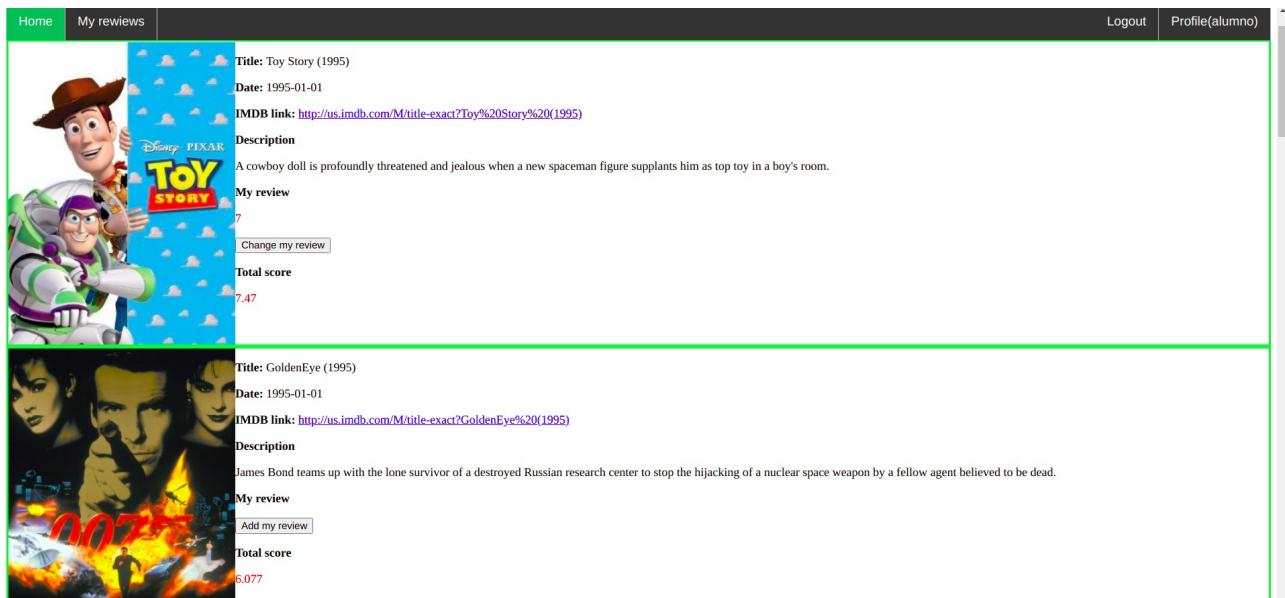


Figure 4: Front page for logged in users

The screenshot shows a profile page with a dark green header containing navigation links: Home, My reviews, Profile(alumno), and Logout. Below the header, the page is titled 'About me' and displays user information: Name (alumno), Age (24), Sex (M), and Occupation (administrator). Below this, there is a 'Reset Password' section with a message: 'Please fill out this form to reset your password.' It contains two input fields labeled 'New Password' and 'Confirm Password', followed by 'Submit' and 'Cancel' buttons.

Figure 5: Profile page with shrunk window

While clicking on button *Change my review* or *Add my review*, small textbox is dynamically added beside button that accepts only numbers from 1 to 10. After entering a review, by clicking again on the same number, review is being saved into DB. Using id of a button, application knows whether it needs to do INSERT or UPDATE in DB.




	<p><b>Title:</b> Toy Story (1995)</p> <p><b>Date:</b> 1995-01-01</p> <p><b>IMDB link:</b> <a href="http://us.imdb.com/M/ti">http://us.imdb.com/M/ti</a></p> <p><b>Description</b> A cowboy doll is profoundly threaten</p> <p><b>My review</b> 7</p> <p>Change my review <input type="text" value="8"/></p> <p><b>Total score</b> 7.47</p>
	<p><b>Title:</b> GoldenEye (1995)</p> <p><b>Date:</b> 1995-01-01</p> <p><b>IMDB link:</b> <a href="http://us.imdb.com/M/ti">http://us.imdb.com/M/ti</a></p> <p><b>Description</b> James Bond teams up with the lone s</p> <p><b>My review</b> <input type="button" value="Add my review"/></p> <p><b>Total score</b> 6.077</p>
	<p><b>Title:</b> Four Rooms (1995)</p> <p><b>Date:</b> 1995-01-01</p> <p><b>IMDB link:</b> <a href="http://us.imdb.com/M/ti">http://us.imdb.com/M/ti</a></p> <p><b>Description</b> Four interlocking tales that take place</p> <p><b>My review</b> <input type="button" value="Add my review"/></p> <p><b>Total score</b> 5.727</p>

Figure 6: User in process of changing review for Toy Story and adding review for Four Rooms

### 3. Implementation

Whole project is inside *iaproject* folder. For running the application, *iaproject* folder must be in `var/www/html/` folder, and link for homepage will be <http://localhost/iaproject/index.php>.

### 3.1. Configuration file

In config.php file, credentials for connecting to database are set. For connecting to database *moviesDB*, username must be *alumno*, and password *alumno123*. If connection is not possible, error is thrown “Could not connect”.

Every other file that uses DB has to connect to DB through this file. That is done by line of code on the beginning of php script:

```
require_once "config.php";
```

### 3.2. Navigation bar and style

Navigation bar is implemented in file header.php and it is using style that is described in style.css file. By checking session variable *loggedin*, navigation bar is showing different buttons. If user is logged in, navigation bar will contain Home, My Reviews, Logout, and Profile(<Username>) buttons. If not, there will be only Home and Login buttons.

Header is included in every other page by line:

```
<?php include('header.php'); ?>
```

The rest of the application is using bootstrap style-sheets.

### 3.3. Login, logout, register and change password

Login.php is first checking is user already logged in. If yes, it redirects him on index page. If it's not, first checks are all fields filled, and then can he find a unique user with that credentials (SQL command has to return 1 row). If user pass this check, session variables(loggedin, id and username of user) are being set and he is redirected to index page.

Logout.php just unsets all the session variables and redirects to index page.

Rreset-password.php is actually page that shows user data about him and gives him a opportunity to change password if he wants. For that he needs to enter new password 2 times, they must be the same and have 6 or more characters(if not, warning is shown). After updating password, user is logged out, and redirect to login page.

Register.php is already described a bit in section 2. It is checking if server request method is *post* (like every file that is inserting or changing something in DB), and then creates new user if all the credentials are valid. At the end, it redirects user to login page.

### 3.4. Adding and changing reviews

If user is logged in, he can access Home and My Reviews page implemented in files `index.php` and `myrec.php`. They are very similar. The difference is that Home page shows all the movies, and My Reviews page shows only movies that are already reviewed by the current user.

This paragraph will describe more in detail only `index.php` file.

Using `config.php` file and `$link` variable that is actually connection to DB, SQL query is executed. Query retrieves all the data about the movies and average calculated score for each of them. For each row, data about the movie is shown. Also, for each of the movie, new SQL query is executed that retrieves review of the movie that is given by the current user (userid is retrieved from session variable `id`). If user has given the review of the movie, button *Change my review* will be shown, if not, button *Add my review* will be shown (if user is not logged in, nothing about *my review* will be shown).

By clicking on either of these buttons, JavaScript method `addReview()` will be called with 3 parameters. Movie id, user id and button id. Using button id, it will be chosen whether to use INSERT or UPDATE method on DB.

When clicking first time on the button, new input text field will be created that accepts only numbers from 1 to 10.

By clicking second time on the same button, the value of the text field will be checked. If value is by the rules, with ajax, data will be sent to the file `save_js.php`. Data sent by the post method are `userid`, `movieid`, `review` and `request` (by checking button id, this will be either “update” or “insert”).

`Save_js.php` file is like bridge between JavaScript method and DB. It receives sent data through php global variable `$_POST`, and executes INSERT or UPDATE. If query is executed correctly, it will return status code 200, else 201.