# **Ottawa Movies - Functionality Guide**

"Ottawa Movies" is a dynamic web application designed to help users explore, search, and manage a collection of movies. Built using HTML, CSS, JavaScript, PHP, and a MySQL database, this project provides an interactive platform where users can browse a movie catalog, register an account, search for movies, and create a personalized playlist. This functionality guide explains how to use the key features of the application, ensuring a smooth and enjoyable experience.

## Key Features and How to Use Them

**1. Navigation Bar**

* **Purpose**: Provides easy access to all major features of the application, adapting dynamically based on the user’s login status.
* **Location**: Present across all pages (e.g., index.php, server/login.php, etc.).
* **How to Use**:
  1. Locate the navigation bar at the top of every page, featuring the "Ottawa Movies" logo (links to index.php) and a list of navigation links.
  2. Use the links to navigate:
     + "Home": Always visible, takes you to the homepage (index.php).
     + "Search": Always visible, opens the search page (server/search.php).
     + "Login/Register": Visible when not logged in, directs to the login page (server/login.php).
     + "My Playlist": Visible when logged in, opens your playlist (server/playlist.php).
     + "Logout": Visible when logged in, logs you out and returns to the homepage (server/logout.php).
* **Notes**:
  1. The navigation bar updates automatically upon login or logout, reflecting your session status via PHP’s $\_SESSION['user\_id'].

**2. Homepage (Movie Carousel)**

* **Purpose**: Displays a carousel of the "Best 100 Movies" for users to browse.
* **Location**: index.php
* **How to Use**:
  1. Open the application in your browser by navigating to the homepage.
  2. View the movie carousel under the "Best 100 Movies" section.
  3. Use the left (←) and right (→) arrow buttons to scroll through the movie list horizontally.
  4. Click on a movie image to visit its details page (server/movie\_details.php?id=[movie\_id]).
  5. If logged in, click the "Add to Playlist" button below a movie to save it to your personal playlist.
* **Notes**:
  1. The carousel is dynamically populated from the movies table in the database.
  2. Movie titles longer than 20 characters are truncated with an ellipsis (...) for better readability.

**Movie Carousel JavaScript Code**:

* 1. **Location**: js/index.js
  2. **Purpose**: Enables horizontal scrolling through the movie grid using left and right arrow buttons.

**3. User Registration**

* **Purpose**: Allows new users to create an account to access personalized features like playlists.
* **Location**: server/register.php
* **How to Use**:
  1. From the homepage, click the "Login/Register" link in the navigation bar (visible when not logged in).
  2. On the login page, click the "Register here" link to navigate to the registration form.
  3. Fill in the required fields:
     + **Username**: Enter a unique username (minimum 3 characters, enforced by JavaScript).
     + **Email**: Provide a valid email address (e.g., [user@example.com](mailto:user@example.com)).
     + **Password**: Enter a secure password (minimum 6 characters, enforced by JavaScript).
     + **Confirm Password**: Re-enter the password to confirm it matches.
     + **Terms Checkbox**: Check the box to agree to the terms and conditions.
  4. Click the "Register" button to submit the form.
  5. If successful, you’ll be redirected to the login page (server/login.php) to sign in with your new credentials.
* **Validation**:
  1. **Client-Side (JavaScript)**: js/register.js checks that all fields are filled, the email format is valid, passwords match, the username is at least 3 characters, the password is at least 6 characters, and the terms checkbox is checked. Error messages (e.g., "Passwords do not match") appear below invalid fields.
  2. **Server-Side (PHP)**: Ensures no duplicate usernames or emails exist and hashes the password using password\_hash() before storing it in the users table.
* **Notes**:
  1. User data is securely stored in the flixgo\_db database with hashed passwords for security.

**4. User Login**

* **Purpose**: Allows registered users to log in and access account-specific features.
* **Location**: server/login.php
* **How to Use**:
  1. From the homepage, click the "Login/Register" link in the navigation bar.
  2. Enter your registered username and password in the login form.
  3. Click the "Sign In" button to log in.
  4. Upon successful login, you’ll be redirected to the homepage (index.php), and the navigation bar will update to show "My Playlist" and "Logout" options.
* **Validation**:
  1. **Client-Side**: ensures both fields are filled before submission (basic check via required HTML attribute).
  2. **Server-Side**: Verifies the username exists in the users table and checks the password against the stored hash using password\_verify(). Displays errors like "Invalid username or password!" if login fails.
* **Notes**:
  1. The session is managed using PHP’s $\_SESSION['user\_id'] to track the logged-in user securely across pages.

**5. Search Movies**

* **Purpose**: Enables users to search for movies by title, cast, director, or writer.
* **Location**: server/search.php
* **How to Use**:
  1. Click the "Search" link in the navigation bar.
  2. On the search page, enter a keyword (e.g., "Star Wars", "Nolan") in the search input field.
  3. Click the "Search" button to submit the query.
  4. View the search results displayed in a grid below the search bar.
  5. Click on a movie card to visit its details page (server/movie\_details.php?id=[movie\_id]).
* **Features**:
  1. Results are dynamically fetched from the movies table using a combination of LIKE and SOUNDEX for fuzzy matching.
  2. If no movies match, a "No movies found" message appears with the search term.
* **Notes**:
  1. The search is case-insensitive and supports partial matches (e.g., "star" finds "Star Wars").
  2. Results include movie images, titles, genres, and ratings.

**6. Movie Details**

* **Purpose**: Provides detailed information about a selected movie.
* **Location**: server/movie\_details.php?id=[movie\_id]
* **How to Use**:
  1. From the homepage or search results, click a movie’s image or title.
  2. View the movie’s details, including:
     + Poster image
     + Title and year
     + Rank, rating (e.g., ★ 8.5), genre, certificate, runtime, tagline
     + Budget and box office (formatted with commas)
     + Cast, directors, and writers
  3. Return to the homepage or search page using the navigation bar.
* **Notes**:
  1. Data is dynamically pulled from the movies table based on the id parameter.
  2. The layout is responsive, styled with style/movie\_details.css, and adjusts for different devices.

**7. My Playlist**

* **Purpose**: Allows logged-in users to view and manage their saved movies.
* **Location**: server/playlist.php (data stored in the playlists table)
* **How to Use**:
  1. Log in to your account.
  2. Click the "My Playlist" link in the navigation bar.
  3. View your saved movies displayed in a grid, showing images, titles, genres, and ratings.
  4. Click a movie to see its details (server/movie\_details.php?id=[movie\_id]) or click "Remove" to delete it from your playlist.
* **Features**:
  1. Movies are fetched from the playlists table, linked to the user’s user\_id.
  2. Removing a movie updates the playlist instantly via a POST request to delete the entry from the database.
* **Notes**:
  1. Access requires login; unauthenticated users are redirected to server/login.php.
  2. The playlist is private to each user.

**8. Logout**

* **Purpose**: Ends the user’s session and returns them to the homepage.
* **Location**: server/logout.php
* **How to Use**:
  1. When logged in, click the "Logout" link in the navigation bar.
  2. You’ll be logged out and redirected to the homepage (index.php), with the navigation bar reverting to "Login/Register".
* **Notes**:
  1. The session is securely terminated using session\_destroy(), clearing all session data.
  2. Ensures user privacy by fully ending the session.

**Additional Technical Notes**

* **Database**: The application uses a MySQL database (flixgo\_db) with tables users, movies, and playlists. Connection details are in server/config.php.
* **Security**: Passwords are hashed, inputs are sanitized with htmlspecialchars(), and PDO prepared statements prevent SQL injection.
* **File Structure**:
  + PHP logic is in the server/ directory.
  + Styles are in style/.
  + JavaScript is in js/.
  + Images are in image/.