# **Movie Randomizer**

## 🔧 OVERVIEW

A full-stack PHP-MySQL web application that allows users to register, log in, specify movie preferences, spin a wheel to get a random movie suggestion, and manage a personalized movie collection. It is styled and structured to resemble a polished entertainment platform.

## 🧩 CORE FEATURES

### 🎯 1. User Registration

* Collects user data: username, email, password, genre, mood, and runtime range.
* Performs both **client-side** and **server-side** validation.
* Checks for duplicate username or email.
* Stores passwords securely using hashing.

### 🔐 2. User Login

* Secure login with password hashing (password\_verify).
* Redirects to main menu upon success.
* Protects access using session-based authentication.

### 🎡 3. Spin the Wheel

* Users can spin a visual wheel to get a random movie.
* Movie selection is based on:
  + Saved preferences
  + Newly entered preferences
* Uses **AJAX (fetch)** to send filtering data to spin\_wheel.php and receive a result **without reloading the page**.

### 📚 4. Manage Collection

* Users can:
  + View their movie collection
  + Search by title
  + Add movies
  + Remove movies
* Real-time updates using **AJAX** with fetch calls to:
  + collection\_get.php
  + collection\_add.php
  + collection\_remove.php

### 🔍 5. Search and Filtering

* Filters movies based on:
  + Genre
  + Mood
  + Runtime range
* Search function is integrated into the collection view.

## 🛠️ TECHNOLOGIES USED

### 🔙 Backend

* **PHP (Procedural)**
  + Handles registration, login, collection management, spinning logic
  + Validates data and communicates with the database
  + Returns either HTML or JSON (for dynamic JS responses)
* **MySQL**
  + Stores all user and movie data
  + Foreign keys, constraints, cascading deletes prevent orphaned records
  + Query filtering is used for personalized suggestions

### 🌐 Frontend

* **HTML5**
  + Semantically structured forms, pages, and sections
* **CSS**
  + One main stylesheet (style.css) for layout, colors, typography
  + Additional (spin.css) for visualizing the spinning wheel
  + Responsive design ensures usability across devices
* **JavaScript**
  + Handles all client-side form validation and dynamic interaction
  + Uses fetch() to communicate with PHP scripts
  + Provides user feedback (inline error messages, alerts)

### 📡 AJAX

* Used extensively for:
  + Registering users without page reload
  + Getting movie results from spin
  + Updating collections dynamically
* Implemented using fetch() with JSON responses

## 🧠 PROGRAMMING CONSIDERATIONS

### ✅ Security

* Passwords hashed using password\_hash()
* password\_verify() used for authentication
* User inputs sanitized using prepared statements (recommended, although not yet implemented)

### ✅ Usability

* Errors shown inline next to input fields using JS DOM manipulation
* Inline validation dynamically clears when typing
* Reset button clears all validation messages

### ✅ Responsiveness

* CSS grid/flexbox used for layout
* Forms and buttons adapt to screen size

### ✅ Modular Code

* JS code is separated into files like register.js, login.js, collection.js
* Each PHP script handles one task:
  + register.php: creates user
  + login.php: logs user in
  + spin\_wheel.php: returns JSON result
  + collection\_get.php: returns collection for the logged-in user

### 🧱 DATABASE STRUCTURE (Simplified)

**users**

* id (PK)
* username
* email
* password\_hash
* genre
* mood
* min\_runtime
* max\_runtime

**movies**

* id (PK)
* title
* genre
* mood
* runtime
* poster\_url

**collections**

* user\_id (FK → users.id)
* movie\_id (FK → movies.id)
* Composite PK: (user\_id, movie\_id)
* Prevents duplicate entries
* ON DELETE CASCADE ensures cleanup if a user/movie is deleted

## 📄 PAGE STRUCTURE

**index.html**

* Welcome page with CTA ("Get Started")

**register.html**

* Form for new users
* JS validates input
* Submits via register.php using fetch

**login.html**

* Login form
* Verifies credentials via login.php

**main\_menu.php**

* Logged-in user's dashboard with links to:
  + Watch
  + View Collection
  + Logout

**watch.php**

* Allows preference selection and spinning the movie wheel
* Uses spin\_wheel.php to get results

**manage\_collection.php**

* Displays user's movie list
* Add/remove/search using JS + PHP

## 🎨 UI/UX DESIGN HIGHLIGHTS

* Movie cards with posters and metadata
* Hero banner in homepage
* Wheel animation for spinning
* All forms have inline labels and error messages
* Clean, dark color palette with high contrast
* Feedback is immediate and visually integrated

## 🧪 VALIDATION & ERROR HANDLING

**Client-Side (JavaScript)**

* All required fields
* Email format
* Runtime positive check
* Password match
* Min < Max runtime

**Server-Side (PHP)**

* Validates required fields again
* Checks for existing username/email
* Returns structured JSON on failure
* Displays final success page or redirects

## 🧭 ARCHITECTURE FLOW

* **Frontend (HTML + JS)**  
  → uses fetch() →
* **Backend (PHP scripts)**  
  → queries/modifies →
* **MySQL database**  
  → returns data →
* **PHP encodes JSON or HTML → DOM update**