

```

<!DOCTYPE html>
<html lang="en" class="dark">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Streams Sphere IPTV Player</title>
  <!-- HLS.js for streaming -->
  <script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
  <!-- Tailwind CSS for styling -->
  <script src="https://cdn.tailwindcss.com"></script>
  <!-- Google Fonts for typography -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;60
0;700&family=Orbitron:wght@700&display=swap" rel="stylesheet">
  <!-- YouTube IFrame API -->
  <script src="https://www.youtube.com/iframe_api"></script>
<style>
  /* Aurora Glass Theme */
  :root {
    --bg-color: #0a0a14;
    --text-color: #f0f0f5;
    --text-muted: #a0a0b0;
    --accent-color: #8A2BE2; /* BlueViolet */
    --accent-glow: rgba(138, 43, 226, 0.5);
    --glass-bg: rgba(20, 20, 35, 0.55);
    --border-color: rgba(138, 43, 226, 0.2);
  }

  html, body {
    overflow: hidden;
    height: 100%;
  }

  body {
    font-family: 'Poppins', sans-serif;
    background: linear-gradient(135deg, #0d042f, #2c135c,
#0a192f, #3d2f7b);
    color: var(--text-color);
    background-size: 400% 400%;
    animation: aurora-flow 20s ease infinite;
  }

  @keyframes aurora-flow {
    0% { background-position: 0% 50%; }

```

```

        50% { background-position: 100% 50%; }
        100% { background-position: 0% 50%; }
    }

    .hidden { display: none; }

    .custom-scrollbar::-webkit-scrollbar { width: 8px; }
    .custom-scrollbar::-webkit-scrollbar-track { background:
transparent; }
    .custom-scrollbar::-webkit-scrollbar-thumb { background-color:
var(--accent-color); border-radius: 4px; }

    .glass-ui {
        background: var(--glass-bg);
        backdrop-filter: blur(16px);
        -webkit-backdrop-filter: blur(16px);
        border: 1px solid var(--border-color);
    }

    header, #sidebar {
        background: var(--glass-bg);
        backdrop-filter: blur(16px);
        -webkit-backdrop-filter: blur(16px);
        border-color: var(--border-color);
    }

    #player-container {
        box-shadow: 0 0 25px var(--accent-glow), 0 0 10px
rgba(0,0,0,0.5);
        border: 1px solid var(--border-color);
    }

    .header-title {
        font-family: 'Orbitron', sans-serif;
        font-weight: 700;
        text-shadow: 0 0 10px var(--accent-glow), 0 0 2px
rgba(0,0,0,0.5);
        animation: pulse-text-glow 3s infinite ease-in-out;
    }

    @keyframes pulse-text-glow {
        0%, 100% { text-shadow: 0 0 10px var(--accent-glow), 0 0
2px rgba(0,0,0,0.5); }
        50% { text-shadow: 0 0 20px rgba(138, 43, 226, 0.8), 0 0
2px rgba(0,0,0,0.5); }
    }

    .channel-item.active {

```

```

        background-color: rgba(138, 43, 226, 0.3);
        color: white;
        animation: pulse-aurora 2s infinite;
    }

    @keyframes pulse-aurora {
        0% { box-shadow: 0 0 0 0 var(--accent-glow); }
        70% { box-shadow: 0 0 0 10px rgba(138, 43, 226, 0); }
        100% { background-color: var(--accent-color); box-shadow:
0 0 0 0 rgba(138, 43, 226, 0); }
    }

    .channel-item.active .channel-title, .channel-item.active
.text-slate-400 {
        color: white;
    }

    .channel-item:hover { background-color: rgba(138, 43, 226,
0.2); }

    .favorite-btn.favorite { color: var(--accent-color); }

    .custom-input {
        background-color: rgba(0, 0, 0, 0.2);
        border: 1px solid var(--border-color);
        color: var(--text-color);
        border-radius: 0.5rem;
        outline: none;
        transition: all 0.2s ease-in-out;
    }

    .custom-input:focus {
        border-color: var(--accent-color);
        box-shadow: 0 0 8px var(--accent-glow);
    }

    #sidebar { transition: transform 0.3s ease-in-out; }

    .loader {
        border: 4px solid var(--border-color);
        border-top: 4px solid var(--accent-color);
        border-radius: 50%;
        width: 40px;
        height: 40px;
        animation: spin 1s linear infinite;
    }

    @keyframes spin {

```

```

        0% { transform: rotate(0deg); }
        100% { transform: rotate(360deg); }
    }

#source-selector-dropdown {
    border: 1px solid rgba(138, 43, 226, 0.5);
    box-shadow: 0 0 20px var(--accent-glow);
}

#youtube-player-container iframe {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
}

/* Football Ad Banner Styling */
.football-ad-banner {
    display: block;
    position: relative;
    width: 100%;
    max-width: 700px;
    aspect-ratio: 4 / 1;
    border-radius: 0.75rem;
    overflow: hidden;
    color: white;
    text-decoration: none;
    background-color: #1a0230;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
    animation: pulse-border-glow 4s infinite ease-in-out;
}

@keyframes pulse-border-glow {
    0%, 100% { box-shadow: 0 10px 30px rgba(0, 0, 0, 0.4), 0 0 20px var(--accent-glow); }
    50% { box-shadow: 0 10px 30px rgba(0, 0, 0, 0.5), 0 0 35px rgba(138, 43, 226, 0.8); }
}

.football-ad-banner:hover {
    transform: translateY(-5px) scale(1.02);
    box-shadow: 0 15px 40px rgba(0, 0, 0, 0.5), 0 0 30px var(--accent-glow);
}

.football-ad-banner .banner-bg {
    position: absolute;

```

```

        inset: 0;
        width: 100%;
        height: 100%;
        object-fit: cover;
        transition: transform 0.4s ease-out;
    }
    .football-ad-banner:hover .banner-bg { transform: scale(1.1);
}

.football-ad-banner .banner-overlay {
    position: absolute;
    inset: 0;
    background: linear-gradient(to right, rgba(10, 2, 30, 0.8)
0%, rgba(0,0,0,0.1) 50%, rgba(10, 2, 30, 0.8) 100%);
    z-index: 1;
}

.football-ad-banner .banner-content {
    position: relative;
    z-index: 2;
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
    padding: 1rem;
    text-align: center;
}

.football-ad-banner .cta-text { text-shadow: 0 2px 10px
rgba(0,0,0,0.8); }
.football-ad-banner .cta-text h2 {
    font-size: clamp(1.2rem, 4vw, 2rem);
    font-weight: 700;
    line-height: 1.1;
}

.football-ad-banner .cta-button {
    display: inline-block;
    margin-top: 0.75rem;
    padding: 0.5rem 1.5rem;
    background: linear-gradient(45deg, var(--accent-color),
#c367fd);
    border-radius: 999px;
    font-weight: 600;
    font-size: clamp(0.8rem, 2.5vw, 1rem);
    box-shadow: 0 0 15px var(--accent-glow);
    transition: all 0.2s ease;
    animation: pulse-glow 2.5s infinite;

```

```

}

.football-ad-banner:hover .cta-button {
    transform: scale(1.05);
    box-shadow: 0 0 25px var(--accent-glow);
}

@keyframes pulse-glow {
    0%, 100% { box-shadow: 0 0 15px var(--accent-glow); }
    50% { box-shadow: 0 0 25px rgba(138, 43, 226, 0.8); }
}

.football-ad-banner::before {
    content: '';
    position: absolute;
    top: 0;
    left: -150%;
    width: 100%;
    height: 100%;
    background: linear-gradient(90deg, transparent, rgba(255,
255, 255, 0.25), transparent);
    transition: left 0.6s ease-in-out;
    z-index: 3;
    transform: skewX(-25deg);
}

.football-ad-banner:hover::before { left: 150%; }

/* Enhanced DVD Case Styling */
.dvd-case {
    position: relative;
    background: linear-gradient(45deg, #1e103f, #0a0a14);
    border: 1px solid var(--border-color);
    border-radius: 0.5rem;
    overflow: hidden;
    transition: transform 0.3s ease, box-shadow 0.3s ease;
    cursor: pointer;
    display: flex;
    flex-direction: column;
    box-shadow: 0 5px 15px rgba(0,0,0,0.3), 0 0 5px
var(--accent-glow);
}

.dvd-case:hover {
    transform: scale(1.05) perspective(1000px) rotateY(2deg);
    box-shadow: 0 10px 25px rgba(0,0,0,0.5), 0 0 20px
var(--accent-glow);
}

```

```

.dvd-case-img-container {
    aspect-ratio: 2 / 3;
    background-color: #1a1a2e;
    display: flex;
    align-items: center;
    justify-content: center;
    overflow: hidden;
    position: relative;
}

.dvd-case-img-container::after { /* Inner shadow for depth */
    content: '';
    position: absolute;
    top: 0; left: 0; right: 0; bottom: 0;
    box-shadow: inset 0 0 15px rgba(0,0,0,0.5);
}

.dvd-case-img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    transition: transform 0.3s ease;
}

.dvd-case:hover .dvd-case-img { transform: scale(1.1); }

.dvd-case-title {
    padding: 0.75rem 0.5rem;
    font-size: 0.8rem;
    font-weight: 500;
    text-align: center;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    background: linear-gradient(to top, rgba(0,0,0,0.7),
    rgba(0,0,0,0.4));
    border-top: 1px solid var(--border-color);
}

/* Plastic wrap shine effect */
.dvd-case::before {
    content: '';
    position: absolute;
    top: 0;
    left: -85%;
    z-index: 2;
    display: block;

```

```

        width: 50%;
        height: 100%;
        background: linear-gradient(to right, rgba(255,255,255,0)
0%, rgba(255,255,255,0.3) 100%);
        transform: skewX(-25deg);
        transition: left 0.7s ease;
    }
    .dvd-case:hover::before { left: 110%; }
</style>
</head>
<body class="bg-slate-900">

<div class="relative h-full w-full flex flex-col">
    <!-- Header -->
    <header class="w-full border-b flex items-center justify-between
p-2 z-20 flex-shrink-0">
        <div class="flex items-center space-x-4">
            <button id="menu-toggle" class="md:hidden p-2
text-slate-400 hover:text-purple-400 transition-colors">
                <svg xmlns="http://www.w3.org/2000/svg" width="24"
height="24" viewBox="0 0 24 24" fill="none" stroke="currentColor"
stroke-width="2" stroke-linecap="round" stroke-linejoin="round"><line
x1="3" y1="12" x2="21" y2="12"></line><line x1="3" y1="6" x2="21"
y2="6"></line><line x1="3" y1="18" x2="21" y2="18"></line></svg>
            </button>
            <div class="flex items-center space-x-3">
                
                <span class="text-xl font-bold text-white
header-title">Streams Sphere</span>
            </div>
        </div>
    </header>

    <!-- Main Content -->
    <div class="flex flex-1 overflow-hidden min-h-0">
        <!-- Sidebar -->
        <aside id="sidebar" class="absolute md:relative top-0 left-0
h-full w-64 md:w-72 border-r flex flex-col z-30 transform
-translate-x-full md:translate-x-0 flex-shrink-0 min-h-0">
            <div class="p-4 border-b border-[var(--border-color)]">
                <div id="custom-select-container" class="relative">
                    <button id="source-selector-button" class="w-full
p-2 custom-input flex items-center justify-between text-left">
                        <span class="flex items-center space-x-2
truncate">
                            <span id="source-selector-icon"></span>

```



```

        <span id="source-selector-text">Select a
Playlist</span>
    </span>
    <svg class="w-4 h-4 text-slate-400"
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20"
fill="currentColor"><path fill-rule="evenodd" d="M5.293 7.293a1 1 0
011.414 0L10 10.586l3.293-3.293a1 1 0 111.414 1.414l-4 4a1 1 0
01-1.414 0l-4-4a1 1 0 010-1.414z" clip-rule="evenodd" /></svg>
    </button>
    <div id="source-selector-dropdown" class="absolute
hidden mt-1 w-full glass-ui rounded-md shadow-lg z-40 flex flex-col">
        <div class="p-2 border-b
border-[var(--border-color)]">
            <input type="text"
id="playlist-search-input" placeholder="Search playlists..."
class="w-full p-2 custom-input text-sm">
        </div>
        <div id="playlist-options-container"
class="max-h-60 overflow-y-auto custom-scrollbar"></div>
    </div>
</div>
    <div class="p-4 border-b border-[var(--border-color)]">
        <input type="text" id="search-input"
placeholder="Search channels..." class="w-full p-2 custom-input
text-sm">
    </div>
    <div id="playlist-container" class="flex-1 overflow-y-auto
custom-scrollbar">
        <div id="playlist-element"></div>
    </div>
</aside>

<!-- Player Area -->
<main class="flex-1 flex flex-col p-2 md:p-4 overflow-y-auto
custom-scrollbar">
    <div id="player-container" class="relative w-full
aspect-video bg-black rounded-lg flex-shrink-0">
        <video id="main-player" class="absolute top-0 left-0
w-full h-full rounded-lg" controls playsinline></video>
        <div id="youtube-player-container" class="absolute
top-0 left-0 w-full h-full hidden"></div>
        <div id="error-overlay" class="hidden absolute inset-0
bg-black bg-opacity-80 flex items-center justify-center text-center
p-4 rounded-lg">
            <p class="text-red-500 font-semibold">Could not
load stream. The source may be offline or invalid.</p>
        </div>

```

```

    </div>

    <!-- Premium Ad Banner -->
    <div id="premium-ad-banner-container" class="my-4 flex
justify-center px-2 flex-shrink-0">
        <a
href="https://www.profitableratecpm.com/qb069j7b6?key=e6b8dcd0a21df190
17732330d1236fd3" target="_blank" rel="noopener noreferrer"
class="football-ad-banner">
            
            <div class="banner-overlay"></div>
            <div class="banner-content">
                <div class="cta-text">
                    <h2>Unlock Every Goal</h2>
                    <span class="cta-button">LIVE Matches
Streaming Now</span>
                </div>
            </div>
        </a>
    </div>

    <!-- Featured Channels Section -->
    <div id="featured-container" class="my-4 px-2">
        <h3 class="text-lg font-bold text-white mb-4
header-title text-center">Featured Channels</h3>
        <div id="featured-channels-grid" class="grid
grid-cols-2 sm:grid-cols-3 md:grid-cols-4 lg:grid-cols-5 gap-4">
            <!-- Channels will be injected here by the script
-->
        </div>
    </div>
</main>
</div>

<!-- Ad Modal -->
<div id="ad-modal" class="hidden fixed inset-0 bg-black bg-opacity-75
flex items-center justify-center z-50">
    <div class="glass-ui p-4 text-center rounded-lg shadow-lg
shadow-purple-500/20 max-w-sm w-full mx-4">
        
        <button id="close-ad-btn" class="mt-2 p-2 w-full bg-purple-600
text-white font-bold rounded hover:bg-purple-500

```

```

transition-colors">Close Ad</button>
    </div>
</div>

<div id="mobile-overlay" class="fixed inset-0 bg-black bg-opacity-50
z-20 hidden md:hidden"></div>

<script>
document.addEventListener('DOMContentLoaded', function() {
    // DOM Element references
    const elements = {
        mainPlayer: document.getElementById('main-player'),
        youtubePlayerContainer:
document.getElementById('youtube-player-container'),
        playerContainer: document.getElementById('player-container'),
        errorOverlay: document.getElementById('error-overlay'),
        sourceSelectorButton:
document.getElementById('source-selector-button'),
        sourceSelectorIcon:
document.getElementById('source-selector-icon'),
        sourceSelectorText:
document.getElementById('source-selector-text'),
        sourceSelectorDropdown:
document.getElementById('source-selector-dropdown'),
        playlistSearchInput:
document.getElementById('playlist-search-input'),
        playlistOptionsContainer:
document.getElementById('playlist-options-container'),
        searchInput: document.getElementById('search-input'),
        playlistElement: document.getElementById('playlist-element'),
        adModal: document.getElementById('ad-modal'),
        adImage: document.getElementById('ad-image'),
        closeAdBtn: document.getElementById('close-ad-btn'),
        sidebar: document.getElementById('sidebar'),
        menuToggle: document.getElementById('menu-toggle'),
        mobileOverlay: document.getElementById('mobile-overlay'),
        featuredChannelsGrid:
document.getElementById('featured-channels-grid'),
    };

    // State variables
    let hls, ytPlayer, allChannelsInPlaylist = [], groupedChannels =
{};
    let favorites = JSON.parse(localStorage.getItem('iptv_favorites'))
|| [];
    let recentlyWatched =
JSON.parse(localStorage.getItem('iptv_recently_watched')) || [];
    let channelClickCounter = 0, activePlayer = null, urlToPlayAfterAd

```

```

= null;
    let lastClickedChannel = null;

    // TMDb API Key
    const TMDB_API_KEY = '3ea9e43d0b879725ac8fda707021e858';

    // Data
    const countryMap =
    {
      'af': {name: 'Afghanistan', flag: },
      'al': {name: 'Albania', flag: },
      'dz': {name: 'Algeria', flag: },
      'ad': {name: 'Andorra', flag: },
      'ao': {name: 'Angola', flag: },
      'ar': {name: 'Argentina', flag: },
      'am': {name: 'Armenia', flag: },
      'aw': {name: 'Aruba', flag: },
      'au': {name: 'Australia', flag: },
      'at': {name: 'Austria', flag: },
      'az': {name: 'Azerbaijan', flag: },
      'bs': {name: 'Bahamas', flag: },
      'bh': {name: 'Bahrain', flag: },
      'bd': {name: 'Bangladesh', flag: },
      'bb': {name: 'Barbados', flag: },
      'by': {name: 'Belarus', flag: },
      'be': {name: 'Belgium', flag: },
      'bz': {name: 'Belize', flag: },
      'bj': {name: 'Benin', flag: },
      'bm': {name: 'Bermuda', flag: },
      'bt': {name: 'Bhutan', flag: },
      'bo': {name: 'Bolivia', flag: },
      'ba': {name: 'Bosnia and Herzegovina', flag: },
      'bw': {name: 'Botswana', flag: },
      'br': {name: 'Brazil', flag: },
      'bn': {name: 'Brunei', flag: },
      'bg': {name: 'Bulgaria', flag: },
      'bf': {name: 'Burkina Faso', flag: },
      'bi': {name: 'Burundi', flag: },
      'kh': {name: 'Cambodia', flag: },
      'cm': {name: 'Cameroon', flag: },
      'ca': {name: 'Canada', flag: },
      'cv': {name: 'Cape Verde', flag: },
      'ky': {name: 'Cayman Islands', flag: },
      'cf': {name: 'Central African Republic', flag: },
      'td': {name: 'Chad', flag: },
      'cl': {name: 'Chile', flag: },
      'cn': {name: 'China', flag: },
      'co': {name: 'Colombia', flag: },
      'km': {name: 'Comoros', flag: },
      'cg': {name: 'Congo', flag: },
      'cd': {name: 'DR Congo', flag: },
      'cr': {name: 'Costa Rica', flag: },
      'ci': {name: 'Cote d'Ivoire', flag: },
      'hr': {name: 'Croatia', flag: },
      'cu': {name: 'Cuba', flag: },
      'cy': {name: 'Cyprus', flag: },
      'cz': {name: 'Czech Republic', flag: },
      'dk': {name: 'Denmark', flag: },
      'dj': {name: 'Djibouti', flag: },
      'dm': {name: 'Dominica', flag: },
      'do': {name: 'Dominican Republic', flag: },
      'ec': {name: 'Ecuador', flag: },
      'eg': {name: 'Egypt', flag: },
      'sv': {name: 'El Salvador', flag: },
      'gq': {name: 'Equatorial Guinea', flag: },
      'ee': {name: 'Estonia', flag: },
      'et': {name: 'Ethiopia', flag: },
      'fj': {name: 'Fiji', flag: },
      'fi': {name: 'Finland', flag: },
      'fr': {name: 'France', flag: },
      'ga': {name: 'Gabon', flag: },
      'gm': {name: 'Gambia', flag: },
      'ge': {name: 'Georgia', flag: },
      'de': {name: 'Germany', flag: },
      'gh': {name: 'Ghana', flag: },
      'gi': {name: 'Gibraltar', flag: },
      'gr': {name: 'Greece', flag: },
      'gl': {name: 'Greenland', flag: },
      'gd': {name: 'Grenada', flag: },
      'gp': {name: 'Guadeloupe', flag: },
      'gu': {name: 'Guam', flag: },
      'gt': {name: 'Guatemala', flag: },
      'gn': {name: 'Guinea', flag: },
      'gw': {name: 'Guinea-Bissau', flag: },
      'gy': {name: 'Guyana', flag: },
      'ht': {name: 'Haiti', flag: },
      'hn': {name: 'Honduras', flag: },
    }

```

: 'Honduras', flag: '', 'hk': {name: 'Hong Kong', flag: '', 'hu': {name: 'Hungary', flag: '', 'is': {name: 'Iceland', flag: '', 'in': {name: 'India', flag: '', 'id': {name: 'Indonesia', flag: '', 'ir': {name: 'Iran', flag: '', 'iq': {name: 'Iraq', flag: '', 'ie': {name: 'Ireland', flag: '', 'im': {name: 'Isle of Man', flag: '', 'il': {name: 'Israel', flag: '', 'it': {name: 'Italy', flag: '', 'jm': {name: 'Jamaica', flag: '', 'jp': {name: 'Japan', flag: '', 'je': {name: 'Jersey', flag: '', 'jo': {name: 'Jordan', flag: '', 'kz': {name: 'Kazakhstan', flag: '', 'ke': {name: 'Kenya', flag: '', 'xk': {name: 'Kosovo', flag: '', 'kw': {name: 'Kuwait', flag: '', 'kg': {name: 'Kyrgyzstan', flag: '', 'la': {name: 'Laos', flag: '', 'lv': {name: 'Latvia', flag: '', 'lb': {name: 'Lebanon', flag: '', 'ls': {name: 'Lesotho', flag: '', 'lr': {name: 'Liberia', flag: '', 'ly': {name: 'Libya', flag: '', 'li': {name: 'Liechtenstein', flag: '', 'lt': {name: 'Lithuania', flag: '', 'lu': {name: 'Luxembourg', flag: '', 'mo': {name: 'Macau', flag: '', 'mk': {name: 'North Macedonia', flag: '', 'mg': {name: 'Madagascar', flag: '', 'mw': {name: 'Malawi', flag: '', 'my': {name: 'Malaysia', flag: '', 'mv': {name: 'Maldives', flag: '', 'ml': {name: 'Mali', flag: '', 'mt': {name: 'Malta', flag: '', 'mq': {name: 'Martinique', flag: '', 'mr': {name: 'Mauritania', flag: '', 'mu': {name: 'Mauritius', flag: '', 'yt': {name: 'Mayotte', flag: '', 'mx': {name: 'Mexico', flag: '', 'md': {name: 'Moldova', flag: '', 'mc': {name: 'Monaco', flag: '', 'mn': {name: 'Mongolia', flag: '', 'me': {name: 'Montenegro', flag: '', 'ms': {name: 'Montserrat', flag: '', 'ma': {name: 'Morocco', flag: '', 'mz': {name: 'Mozambique', flag: '', 'mm': {name: 'Myanmar', flag: '', 'na': {name: 'Namibia', flag: '', 'np': {name: 'Nepal', flag: '', 'nl': {name: 'Netherlands', flag: '', 'nc': {name: 'New Caledonia', flag: '', 'nz': {name: 'New Zealand', flag: '', 'ni': {name: 'Nicaragua', flag: '', 'ne': {name: 'Niger', flag: '', 'ng': {name: 'Nigeria', flag: '', 'kp': {name: 'North Korea', flag: '', 'no': {name: 'Norway', flag: '', 'om': {name: 'Oman', flag: '', 'pk': {name: 'Pakistan', flag: '', 'ps': {name: 'Palestine', flag: '', 'pa': {name: 'Panama', flag: '', 'pg': {name: 'Papua New Guinea', flag: '', 'py': {name: 'Paraguay', flag: '', 'pe': {name: 'Peru', flag: '', 'ph': {name: 'Philippines', flag: '', 'pl': {name: 'Poland', flag: '', 'pt': {name: 'Portugal', flag: '', 'pr': {name: 'Puerto Rico', flag: '', 'qa': {name: 'Qatar', flag: '', 're': {name: 'Reunion', flag: '', 'ro': {name: 'Romania', flag: '', 'ru': {name: 'Russia', flag: '', 'rw': {name: 'Rwanda', flag: '', 'kn': {name: 'Saint Kitts and Nevis', flag: '', 'lc': {name: 'Saint Lucia', flag: '', 'vc': {name: 'Saint Vincent and the Grenadines', flag: '', 'sm': {name: 'San Marino', flag: '', 'st': {name: 'Sao Tome and Principe', flag: '', 'sa': {name: 'Saudi Arabia', flag: '', 'sn': {name: 'Senegal', flag: '', 'rs': {name: 'Serbia', flag: '', 'sc': {name: 'Seychelles', flag: '', 'sl': {name: 'Sierra Leone', flag: '', 'sg': {name: 'Singapore', flag: '', 'sk': {name: 'Slovakia', flag: '', 'si': {name: 'Slovenia', flag: '', 'so': {name: 'Somalia', f

```

lag: '🇺🇸'}, 'za': {name: 'South Africa', flag: '🇿🇦'}, 'kr': {name: 'South
Korea', flag: '🇰🇷'}, 'ss': {name: 'South
Sudan', flag: '🇸🇸'}, 'es': {name: 'Spain', flag: '🇪🇸'}, 'lk': {name: 'Sri
Lanka', flag: '🇱🇰'}, 'sd': {name: 'Sudan', flag: '🇸🇩'}, 'sr': {name: 'Suriname',
flag: '🇸🇷'}, 'se': {name: 'Sweden', flag: '🇸🇪'}, 'ch': {name: 'Switzerland', fla
g: '🇨🇭'}, 'sy': {name: 'Syria', flag: '🇸🇾'}, 'tw': {name: 'Taiwan', flag: '🇹🇼'}, '
tz': {name: 'Tanzania', flag: '🇹🇿'}, 'th': {name: 'Thailand', flag: '🇹🇭'}, 'tl':
{name: 'Timor-Leste', flag: '🇹🇱'}, 'tg': {name: 'Togo', flag: '🇹🇬'}, 'to': {name
: 'Tonga', flag: '🇹🇴'}, 'tt': {name: 'Trinidad and
Tobago', flag: '🇹🇹'}, 'tn': {name: 'Tunisia', flag: '🇹🇳'}, 'tr': {name: 'Turkey'
, flag: '🇹🇷'}, 'tm': {name: 'Turkmenistan', flag: '🇹🇲'}, 'ug': {name: 'Uganda', f
lag: '🇺🇬'}, 'ua': {name: 'Ukraine', flag: '🇺🇦'}, 'ae': {name: 'United Arab
Emirates', flag: '🇦🇪'}, 'gb': {name: 'United
Kingdom', flag: '🇬🇧'}, 'us': {name: 'United
States', flag: '🇺🇸'}, 'uy': {name: 'Uruguay', flag: '🇺🇾'}, 'uz': {name: 'Uzbekis
tan', flag: '🇺🇿'}, 've': {name: 'Venezuela', flag: '🇻🇪'}, 'vn': {name: 'Vietnam'
, flag: '🇻🇳'}, 'ye': {name: 'Yemen', flag: '🇾🇪'}, 'zm': {name: 'Zambia', flag: '🇿🇲'
}, 'zw': {name: 'Zimbabwe', flag: '🇿🇼'}}};
const countryCodes = Object.keys(countryMap);
const countryPlaylists = countryCodes.map(code => {
  const country = countryMap[code];
  const url = (code === 'gb')
    ?
'https://raw.githubusercontent.com/devsground/IPTV/master/country/T-Z/
united_kingdom.m3u'
    : `https://iptv-org.github.io/iptv/countries/${code}.m3u`;
  return { name: country.name, url: url, icon: `

```

```

        { "title": "The Big Picture", "url":
"http://www.youtube.com/watch?v=6DVjYiFCPBU", "group": "Comedy" },
        { "title": "Bad People (Comedy Movie, AWARD-WINNING, HD,
Full Film, English) free comedy movie on youtube", "url":
"http://www.youtube.com/watch?v=XFz1TWx-MY0", "group": "Comedy" },
        { "title": "HOUSE ARREST: THE FUNNIEST VIDEO LEGALLY ON
YOUTUBE", "url": "http://www.youtube.com/watch?v=9Ys_F4yHaCk",
"group": "Comedy" },
        { "title": "Scary Movie 2 Full Movie | LOL! Throwbacks",
"url": "http://www.youtube.com/watch?v=K3179BbyWyc", "group": "Comedy"
},
        { "title": "Bessie Ludd Movies The Good Ole Days Funniest
Movie Ever", "url": "http://www.youtube.com/watch?v=XJq-X5wJEao",
"group": "Comedy" },
        { "title": "Ernest Goes To Africa | English Full Movie |
Comedy", "url": "http://www.youtube.com/watch?v=d88bIPB_KJO", "group":
"Comedy" },
        { "title": "Lightning Jack 1994 (full movie)", "url":
"http://www.youtube.com/watch?v=kYsk-w7t0LA", "group": "Comedy" },
        { "title": "Slam Dunk Ernest | English Full Movie | Comedy
Family Sport", "url": "http://www.youtube.com/watch?v=raDsKjx_AQQ",
"group": "Comedy" },
        { "title": "Night of the Comet Widescreen Full Movie
1984", "url": "http://www.youtube.com/watch?v=me4pOQ0t20s", "group":
"Sci-Fi" },
        { "title": "The Distinguished Gentleman Full Film Eddie
Murphy (1992)", "url": "http://www.youtube.com/watch?v=q6kfDW7RkI0",
"group": "Comedy" },
        { "title": "Going Overboard | Adam Sandler (Click) |
COMEDY | Full Movie in English", "url":
"http://www.youtube.com/watch?v=MJD3P4n0m_Q", "group": "Comedy" },
        { "title": "Ladybugs 1992 Full Movie", "url":
"http://www.youtube.com/watch?v=3P9xHCED5C0", "group": "Comedy" },
        { "title": "Summertime Switch (1994) Full Movie", "url":
"http://www.youtube.com/watch?v=dmyWL6cqeVI", "group": "Comedy" },
        { "title": "Dealing With Dad (2024) | Full Movie | Comedy
Drama", "url": "http://www.youtube.com/watch?v=A0nCoX-Ck2c", "group":
"Comedy" },
        { "title": "You're Never Too Young 1955 Jerry Lewis Dean
Martin Full Length Comedy Movie", "url":
"http://www.youtube.com/watch?v=DYUF2hghP3w", "group": "Comedy" },
        { "title": "American Sophomore (Comedy Movie, Full Length
Film, English Flick, HD) watch free youtube films", "url":
"http://www.youtube.com/watch?v=I_adw7au0c4", "group": "Comedy" },
        { "title": "This Movie is Too Funny A Night In Compton
Free Movie", "url": "http://www.youtube.com/watch?v=u06GRLmhgms",
"group": "Comedy" },
        { "title": "Teenage Bank Heist Full Movie", "url":

```

```

"http://www.youtube.com/watch?v=41KimZRIMCI", "group": "Drama" },
    { "title": "The Heavenly Kid-1985", "url":
"http://www.youtube.com/watch?v=Fr_NWOUK3AY", "group": "Comedy" },
    { "title": "Holy Hustle | Full Movie | Crime Comedy",
"url": "http://www.youtube.com/watch?v=apCTIu7Mbxo", "group": "Comedy"
},
    { "title": "One Cop, One Arm, One Big Heist | FULL COMEDY
MOVIE | Mall Cop", "url":
"http://www.youtube.com/watch?v=Oacz2_nJ2ZA", "group": "Comedy" },
    { "title": "Daddy Daughter Trip | Rob Schneider | Full
2025 Comedy Family Movie", "url":
"http://www.youtube.com/watch?v=fIhM9MiEN50", "group": "Comedy" },
    { "title": "Hamburger: The Motion Picture (1986) FULL
MOVIE", "url": "http://www.youtube.com/watch?v=HBSla8UDqwY", "group":
"Comedy" },
    { "title": "Frank Vs. God Free Comedy Movie | Full Movie
Crack Up", "url": "http://www.youtube.com/watch?v=7pc759kA_mw",
"group": "Comedy" },
    { "title": "Held Up | Jamie Foxx (Back in Action) | COMEDY
Full Movie in English", "url":
"http://www.youtube.com/watch?v=HPONVxnqhyI", "group": "Comedy" },
    { "title": "Comedy Movies 2015", "url":
"http://www.youtube.com/watch?v=TGTlU57xZHk", "group": "Comedy" },
    { "title": "Africa Screams 1949 Bud Abbott, Lou Costello,
Shemp Howard And Joe Besser", "url":
"http://www.youtube.com/watch?v=DGRgyXohAlE", "group": "Comedy" },
    { "title": "Rat Race | Full Movie", "url":
"http://www.youtube.com/watch?v=127rJwowaAc", "group": "Comedy" },
    { "title": "Big Baby | FULL MOVIE | 2015 | Family, Comedy
Toddler turns 30! | Maureen McCormick", "url":
"http://www.youtube.com/watch?v=xEeeHUI4CXA", "group": "Comedy" },
    { "title": "Rock a Bye Baby 1958 Jerry Lewis Dean Martin
Full Length Comedy Movie", "url":
"http://www.youtube.com/watch?v=uHCRY7AS4SY", "group": "Comedy" },
    { "title": "Don't forget to leave a like and subscribe The
Original Kings of Comedy FULL MOVIE", "url":
"http://www.youtube.com/watch?v=bT_UP2iFtr0", "group": "Comedy" },
    { "title": "Volunteers 1985 Full Movie", "url":
"http://www.youtube.com/watch?v=arLMHM-Ou4Q", "group": "Comedy" },
    { "title": "Ace Ventura: Pet Detective Jr | Full English
Movie | Comedy Adventure", "url":
"http://www.youtube.com/watch?v=ibHswCbutJ4", "group": "Comedy" },
    { "title": "The Horse's Mouth | English Full Movie |
Comedy", "url": "http://www.youtube.com/watch?v=3sQM2FXuIPU", "group":
"Comedy" },
    { "title": "The Funniest Comedy Movie You Need to See
Right Now! The Standoff Hollywood Movies in English HD", "url":
"http://www.youtube.com/watch?v=Im08P56ANAE", "group": "Comedy" },

```



```

        { "title": "American Fork (2007) Comedy / Drama | Full
Movie", "url": "http://www.youtube.com/watch?v=luP6HPliZwY", "group":
"Comedy" },
        { "title": "28 Years Later (2025)", "url":
"https://uflix.cc/movie/28-years-later-2025", "group": "Horror" }
    ].map(video => ({ ...video, logo:
`https://i.ytimg.com/vi/${extractVideoID(video.url)}/hqdefault.jpg`
}))) },
    { name: 'YT Series', icon: '📺', categorized: true, channels:
[
    // Mr Bean Series
    { "title": "Detective Bean! | Mr Bean Live Action | Full
Episodes | Mr Bean", "url":
"http://www.youtube.com/watch?v=_wMDxrPVfak", "group": "Mr Bean" },
    { "title": "Mr Bean the Barber! | Mr Bean Full Episodes Mr
Bean", "url": "http://www.youtube.com/watch?v=Yh-UDswOC2c", "group":
"Mr Bean" },
    { "title": "Mind The Baby, Mr. Bean | Mr Bean S01 E09 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=J4Vcm8wF02g", "group": "Mr Bean" },
    { "title": "Merry Christmas, Mr. Bean | Mr Bean S01 E07
Full Episode HD Official Mr Bean", "url":
"http://www.youtube.com/watch?v=78LulOkGbVw", "group": "Mr Bean" },
    { "title": "How NOT To Drive a Car! | Mr Bean Live Action
Full Episodes Mr Bean", "url":
"http://www.youtube.com/watch?v=7kKUaJtcUcQ", "group": "Mr Bean" },
    { "title": "Hair by Mr. Bean of London | Mr Bean S01 E14
Full Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=K7wbeFle2x0", "group": "Mr Bean" },
    { "title": "Goodnight Mr. Bean | Mr Bean S01 E13 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=D84tnjUzEIA", "group": "Mr Bean" },
    { "title": "Mr. Bean in Room 426 | Mr Bean S01 E08 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=1WwuLyJo-co", "group": "Mr Bean" },
    { "title": "Mr. Bean Goes to Town | Mr Bean S01 E04 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=yT6_qOSN4TM", "group": "Mr Bean" },
    { "title": "Mr Bean's Brand new TV! | Mr Bean Funny Clips
Mr Bean", "url": "http://www.youtube.com/watch?v=aWuFbLY_u3g",
"group": "Mr Bean" },
    { "title": "Back To School Mr. Bean | Mr Bean S01 E11 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=jlg6samLRB8", "group": "Mr Bean" },
    { "title": "The Return of Mr. Bean | Mr Bean S01 E02 -
Full Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=U5yx197Dtfl", "group": "Mr Bean" },
    { "title": "Mr. Bean Rides Again | Mr Bean S01 E06 Full

```

```

Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=NIKCVZro1SE", "group": "Mr Bean" },
    { "title": "Mr Bean The Pickpocket! | Mr Bean Live Action
Full Episodes | Mr Bean", "url":
"http://www.youtube.com/watch?v=SCrkYBc0K6E", "group": "Mr Bean" },
    { "title": "Mr Bean | Mr Bean S01 E01 - Full Episode HD
Official Mr Bean", "url":
"http://www.youtube.com/watch?v=MxPvcU11cg", "group": "Mr Bean" },
    { "title": "The Curse of Mr. Bean | Mr Bean S01 E03 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=mcOsNvDbYqM", "group": "Mr Bean" },
    { "title": "Mr Bean Cheats in His Exam! | Mr Bean Live
Action | Full Episodes | Mr Bean", "url":
"http://www.youtube.com/watch?v=j9eGpP-cHeQ", "group": "Mr Bean" },
    { "title": "Tee Off, Mr. Bean | Mr Bean S01 E12 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=P4fKKxEddyY", "group": "Mr Bean" },
    { "title": "Do It Yourself Mr. Bean | Mr Bean S01 E10 Full
Episode HD | Official Mr Bean", "url":
"http://www.youtube.com/watch?v=CG2si2c80YM", "group": "Mr Bean" },
    { "title": "The Trouble with Mr. Bean | Mr Bean S01 E05
Full Episode HD Official Mr Bean", "url":
"http://www.youtube.com/watch?v=yNs_DhYrOpY", "group": "Mr Bean" },
    { "title": "Technology and Mr Bean are NOT a Great Match!
Mr Bean Live Action | Full Episodes Mr Bean", "url":
"http://www.youtube.com/watch?v=1-Nb5pxucjE", "group": "Mr Bean" },
    // Hell's Kitchen Series
    { "title": "Ep 01 | Young Guns Come Out Shooting | Full
Episode", "url": "http://www.youtube.com/watch?v=TdqiTPeW178",
"group": "Hells Kitchen" },
    { "title": "Ep 02 | Tempting the Meat | Full Episode",
"url": "http://www.youtube.com/watch?v=Kegwonz04dE", "group": "Hells
Kitchen" },
    { "title": "Ep 03 | Come Hell or High Water! | Full
Episode", "url": "http://www.youtube.com/watch?v=x78TjKwyb8o",
"group": "Hells Kitchen" },
    { "title": "Ep 04 | Young Guns Going Big | Full Episode",
"url": "http://www.youtube.com/watch?v=E1GtaWUPFgY", "group": "Hells
Kitchen" },
    { "title": "Ep 05 | Stirring the Pot | Full Episode",
"url": "http://www.youtube.com/watch?v=1TvPwxqI_hg", "group": "Hells
Kitchen" },
    { "title": "Ep 06 | A Ramsay Birthday in Hell! | Full
Episode", "url": "http://www.youtube.com/watch?v=20unnbgaXww",
"group": "Hells Kitchen" },
    { "title": "Ep 07 | If You Can't Stand the Heat... | Full
Episode", "url": "http://www.youtube.com/watch?v=VVQeSxVyFj4",
"group": "Hells Kitchen" },

```

```
        { "title": "Ep 08 | A Devilish Challenge | Full Episode",
"url": "http://www.youtube.com/watch?v=TjV_LEVCtGc", "group": "Hells
Kitchen" },
        { "title": "Ep 09 | A Game Show from Hell | Full Episode",
"url": "http://www.youtube.com/watch?v=aJVA9iN7nvY", "group": "Hells
Kitchen" },
        { "title": "Ep 10 | More Than a Sticky Situation | Full
Episode", "url": "http://www.youtube.com/watch?v=xVgH6g-e0Wc",
"group": "Hells Kitchen" },
        { "title": "Ep 11 | Swiping Right | Full Episode", "url":
"http://www.youtube.com/watch?v=1Bet3Z5cDXg", "group": "Hells Kitchen"
},
        { "title": "Ep 12 | All Hell Breaks Loose | Full Episode",
"url": "http://www.youtube.com/watch?v=3EP2_1lW3nA", "group": "Hells
Kitchen" },
        { "title": "Ep 13 | Social Media in Hell | Full Episode",
"url": "http://www.youtube.com/watch?v=pTkjCaKuAHE", "group": "Hells
Kitchen" },
        { "title": "Ep 14 | Hell Hath No Fury... | Full Episode",
"url": "http://www.youtube.com/watch?v=A2sWn0vmL74", "group": "Hells
Kitchen" },
        { "title": "Ep 15 | What the Hell | Full Episode", "url":
"http://www.youtube.com/watch?v=Y4P16a1CyYY", "group": "Hells Kitchen"
},
        { "title": "Ep 16 | Two Young Guns Shoot It Out | Full
Episode", "url": "http://www.youtube.com/watch?v=VLfVUdMXOTO",
"group": "Hells Kitchen" },
        // The Island With Bear Grylls Series
        { "title": "Season 1 | The First Island Experiment",
"url": "http://www.youtube.com/watch?v=q15h2QeL878", "group": "The
Island With Bear Grylls" },
        { "title": "Season 1 | ALL OF SEASON ONE IN ONE VIDEO",
"url": "http://www.youtube.com/watch?v=M1QpEcFmkdQ", "group": "The
Island With Bear Grylls" },
        { "title": "Season 2 | Women Touch Down on The Island
(Part 1)", "url": "http://www.youtube.com/watch?v=IEDwzzHId3c",
"group": "The Island With Bear Grylls" },
        { "title": "Season 2 | Men vs Women on The Island (Part
2)", "url": "http://www.youtube.com/watch?v=UjzmTHncN1M", "group":
"The Island With Bear Grylls" },
        { "title": "Season 2 | UK ALL EPS IN ONE VIDEO", "url":
"http://www.youtube.com/watch?v=a5PA4Pn5yMA", "group": "The Island
With Bear Grylls" },
        { "title": "Season 3 | Man vs Woman on the Island", "url":
"http://www.youtube.com/watch?v=efoMnfXXFxs", "group": "The Island
With Bear Grylls" },
        { "title": "Season 4 | The Young Vs. The Old", "url":
"http://www.youtube.com/watch?v=xVeZowNJGbo", "group": "The Island
```

```

With Bear Grylls" },
    { "title": "Season 5 | Stereotypes Exist For A Reason",
"url": "http://www.youtube.com/watch?v=LEv6AG8Hi-A", "group": "The
Island With Bear Grylls" },
    { "title": "Season 5 | All Of Season 5 In One Video",
"url": "http://www.youtube.com/watch?v=lHe9lz960eA", "group": "The
Island With Bear Grylls" },
    { "title": "Season 6 | All Of Season 6 In One Video",
"url": "http://www.youtube.com/watch?v=Ibu9H4nxcGM", "group": "The
Island With Bear Grylls" },
    { "title": "Celebrity Season 2 | Celebrities in the Wild",
"url": "http://www.youtube.com/watch?v=mINm0QWXDTY", "group": "The
Island With Bear Grylls" },
    { "title": "Celebrity Season 2 | All Of Season 2 Celeb
Island", "url": "http://www.youtube.com/watch?v=0RiBAAdkyI-U", "group":
"The Island With Bear Grylls" },
    { "title": "Celebrity Season 3 | Pete Wicks'
Dictatorship", "url": "http://www.youtube.com/watch?v=HqeRcU1zM3g",
"group": "The Island With Bear Grylls" },
    { "title": "Celebrity Seasons 2 & 3 | Full Celebrity
Island Seasons", "url": "http://www.youtube.com/watch?v=WoXi9xg68ME",
"group": "The Island With Bear Grylls" },
    ].map(video => ({ ...video, logo:
`https://i.ytimg.com/vi/${extractVideoID(video.url)}/hqdefault.jpg`
})),
    { name: 'Fire TV', url: 'https://www.apsattv.com/firetv.m3u',
icon: '🔥'},
    { name: 'Freetv', url: 'https://www.apsattv.com/freetv.m3u',
icon: '📺'},
    { name: 'Freemoviesplus', url:
'https://www.apsattv.com/freemoviesplus.m3u', icon: '🍿'},
    { name: 'Klowd TV', url: 'https://www.apsattv.com/klowd.m3u',
icon: '☁️'},
    { name: 'LG', url: 'https://www.apsattv.com/lg.m3u', icon:
'📺'},
    { name: 'Plex US', url:
'https://raw.githubusercontent.com/iptv-org/iptv/master/streams/us_ple
x.m3u', icon: '▶️'},
    { name: 'Public Domain', icon: '🏛️', categorized: true,
channels: [
        { title: 'The Death Kiss (1932)', url:
'https://archive.org/download/TheDeathKiss/TheDeathKiss.mp4', group:
'Mystery' },
        { title: 'Dracula (1931)', url:
'https://ia800501.us.archive.org/2/items/dracula-colored/Dracula%201
931.mp4', group: 'Horror' },
        { title: 'House on Haunted Hill (1959) Vincent Price',
url: 'https://youtu.be/VtwGHZk6UES', group: 'Horror' },

```

```

        { title: 'Invasion of the Bee Girls (1973)', url:
'https://ia601504.us.archive.org/34/items/InvasionOfTheBeeGirls/Invasi
onOfTheBeeGirls_512kb.mp4', group: 'Sci-Fi' },
        { title: 'Dalton\'s Women (1950)', url:
'https://dn720405.ca.archive.org/0/items/daltons-women-1950/Daltons%20
Women%201950%20colorized.mp4', group: 'Western' },
        { title: 'Earth vs the Flying Saucers (1956)', url:
'https://archive.org/download/earth-vs-the-flying-saucers-color/Earth%
20vs%20the%20Flying%20Saucers%20color.mp4', group: 'Sci-Fi'},
        { title: 'Sahara (1943)', url:
'https://archive.org/download/sahara-colorized/Sahara%201943.mp4',
group: 'War'},
        { title: 'Frankenstein (1931)', url:
'https://archive.org/download/frankenstein-1931-colorized/Frankenstein
%201931%20colorized.mp4', group: 'Horror'},
        { title: 'Creature from the Black Lagoon (1954)', url:
'https://archive.org/download/creature-from-the-black-lagoon-1954-colo
rized/Creature%20from%20the%20Black%20Lagoon%201954%20colorized.mp4',
group: 'Horror'},
        { title: 'Tarantula (1955)', url:
'https://archive.org/download/tarantula-1955-colorized/Tarantula%20195
5%20colorized.mp4', group: 'Sci-Fi'},
        { title: 'Run Silent Run Deep (1958)', url:
'https://archive.org/download/run-silent-run-deep-1958/Run%20Silent%20
Run%20Deep%201958.mp4', group: 'War'},
        { title: 'It Came from Outer Space (1953)', url:
'https://archive.org/download/it-came-from-outer-space-1953-colorized/
It%20Came%20from%20Outer%20Space%201953%20colorized.mp4', group:
'Sci-Fi'},
        { title: 'The Train (1964)', url:
'https://archive.org/download/train-1964/Train%201964.mp4', group:
'War'},
        { title: 'Gilda (1946)', url:
'https://archive.org/download/gilda-1946-colorized/Gilda%201946.mp4',
group: 'Drama'},
        { title: 'The Deadly Mantis (1957)', url:
'https://archive.org/download/deadly-mantis-1957-colorized/Deadly%20Ma
ntis%201957%20colorized.mp4', group: 'Sci-Fi'},
        { title: 'Invasion of the Body Snatchers (1956)', url:
'https://dn720306.ca.archive.org/0/items/invasion-of-the-body-snatcher
s-1956-colorized/Invasion%20of%20the%20Body%20Snatchers.mp4', group:
'Sci-Fi' },
        { title: 'The Mark of Zorro (1940)', url:
'https://ia601606.us.archive.org/35/items/mark-of-zorro-1940/Mark%20of
%20Zorro%201940.mp4', group: 'Action' },
        { title: 'It! The Terror from Beyond Space (1958)', url:
'https://archive.org/download/it-the-terror-from-beyond-space-colorize
d/It%20The%20Terror%20from%20Beyond%20Space.mp4', group: 'Sci-Fi' },

```

```

        { title: 'Curse of the Demon (1957)', url:
'https://archive.org/download/curse-of-the-demon/Curse%20of%20the%20De
mon.mp4', group: 'Horror' },
        { title: 'The Monster That Challenged the World (1957)',
url:
'https://archive.org/download/monster-that-challenged-the-world-1957-c
olorized/Monster%20That%20Challenged%20the%20World%201957%20colorized.
mp4', group: 'Sci-Fi' },
        { title: 'X the Unknown (1956)', url:
'https://archive.org/download/x-the-unknown-1956-colorized/X%20the%20U
nknown%201956%20colorized.mp4', group: 'Sci-Fi' },
        { title: 'The Incredible Shrinking Man (1957)', url:
'https://archive.org/download/incredible-shrinking-man-1957/Incredible
%20Shrinking%20Man.mp4', group: 'Sci-Fi' },
        { title: 'The Invisible Man (1933)', url:
'https://archive.org/download/invisible-man-1933/Invisible%20Man%20193
3.mp4', group: 'Sci-Fi' },
        { title: 'The Abominable Snowman (1957)', url:
'https://archive.org/download/abominable-snowman-1957-colorized/Abomin
able%20Snowman%201957%20colorized.mp4', group: 'Horror' },
        { "title": "Gargoyles S01E01 - The Awakening, Part 1",
"url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p
art-1/Gargoyles.S01E01.The.Awakening.Part.1.480p.DSNY.WEB-DL.AAC2.0.H.
264-TBD.mkv", "group": "Gargoyles" },
        { "title": "Gargoyles S01E02 - The Awakening, Part 2",
"url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p
art-1/Gargoyles.S01E02.The.Awakening.Part.2.480p.DSNY.WEB-DL.AAC2.0.H.
264-TBD.mkv", "group": "Gargoyles" },
        { "title": "Gargoyles S01E03 - The Awakening, Part 3",
"url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p
art-1/Gargoyles.S01E03.The.Awakening.Part.3.480p.DSNY.WEB-DL.AAC2.0.H.
264-TBD.mkv", "group": "Gargoyles" },
        { "title": "Gargoyles S01E04 - The Awakening, Part 4",
"url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p
art-1/Gargoyles.S01E04.The.Awakening.Part.4.480p.DSNY.WEB-DL.AAC2.0.H.
264-TBD.mkv", "group": "Gargoyles" },
        { "title": "Gargoyles S01E05 - The Awakening, Part 5",
"url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p
art-1/Gargoyles.S01E05.The.Awakening.Part.5.480p.DSNY.WEB-DL.AAC2.0.H.
264-TBD.mkv", "group": "Gargoyles" },
        { "title": "Gargoyles S01E06 - The Thrill of the Hunt",
"url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p

```

```

art-1/Gargoyles.S01E06.The.Thrill.of.the.Hunt.480p.DSNY.WEB-DL.AAC2.0.
H.264-TBD.mkv", "group": "Gargoyles" },
    { "title": "Gargoyles S01E07 - Temptation", "url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p
art-1/Gargoyles.S01E07.Temptation.480p.DSNY.WEB-DL.AAC2.0.H.264-TBD.mk
v", "group": "Gargoyles" },
    { "title": "Gargoyles S01E08 - Deadly Force", "url":
"https://archive.org/download/gargoyles-1994-s-01-e-01-the-awakening-p
art-1/Gargoyles.S01E08.Deadly.Force.480p.DSNY.WEB-DL.AAC2.0.H.264-TBD.
mkv", "group": "Gargoyles" }
  ]},
  { name: 'Weelly', url: 'https://www.apsattv.com/weelly.m3u',
icon: '📺'},
  { name: 'Vidaa', url: 'https://www.apsattv.com/vidaa.m3u',
icon: '📺'},
  { name: 'Xiaomi', url: 'https://www.apsattv.com/xiaomi.m3u',
icon: '📺'},
  { name: 'Xumo', url: 'https://www.apsattv.com/xumo.m3u', icon:
'📺'},
  { name: 'Zean', url: 'https://www.apsattv.com/zeasn.m3u',
icon: '📺'},
  ...countryPlaylists
].sort((a, b) => a.name.localeCompare(b.name));

const CORS_PROXIES = ['https://corsproxy.io/?',
'https://api.allorigins.win/raw?url='];
const CACHE_EXPIRATION_MS = 24 * 60 * 60 * 1000; // 24 hours

function initializePlayers() {
  if (Hls.isSupported()) {
    hls = new Hls({ fragLoadingMaxRetry: 4,
manifestLoadingMaxRetry: 1 });
    hls.attachMedia(elements.mainPlayer);
    hls.on(Hls.Events.ERROR, (event, data) => {
      if (data.fatal) {
        console.error('HLS Error:', data);
        showErrorOverlay();
      }
    });
  }
  elements.mainPlayer.addEventListener('error', () => {
    if (!hls?.url && elements.mainPlayer.getAttribute('src'))
    {
      console.error('Video Error');
      showErrorOverlay();
    }
  });
}

```

```

function showErrorOverlay() {
elements.errorOverlay.classList.remove('hidden'); }
function hideErrorOverlay() {
elements.errorOverlay.classList.add('hidden'); }

function showAd(isInitialAd = false) {
  if (isInitialAd) {
    const lastPlayedChannel =
JSON.parse(localStorage.getItem('iptv_last_played_channel'));
    if (lastPlayedChannel) {
      urlToPlayAfterAd = lastPlayedChannel.url;
    }
  }
  elements.mainPlayer.pause();
  if (activePlayer === 'youtube' && typeof ytPlayer !==
'undefined' && ytPlayer.pauseVideo) {
    ytPlayer.pauseVideo();
  }
  elements.adModal.classList.remove('hidden');
}

function hideAdAndOpenLink() {
  window.open('https://otieu.com/4/9634223', '_blank');
  elements.adModal.classList.add('hidden');
  if (urlToPlayAfterAd) {
    playStream(urlToPlayAfterAd);
    urlToPlayAfterAd = null;
  } else if (activePlayer === 'youtube' && typeof ytPlayer !==
'undefined' && ytPlayer.playVideo) {
    ytPlayer.playVideo();
  } else if (activePlayer === 'hls' && elements.mainPlayer) {
    elements.mainPlayer.play().catch(e =>
console.warn("Playback resume failed.", e));
  }
}

async function fetchWithFallback(url) {
  for (const proxy of CORS_PROXIES) {
    try {
      const response = await fetch(proxy +
encodeURIComponent(url));
      if (response.ok) return await response.text();
    } catch (e) {
      console.warn(`Proxy failed for ${url}:`, e);
    }
  }
  throw new Error(`All CORS proxies failed for ${url}.`);
}

```



```

    }

    async function parseM3U(data, playlistName) {
        const lines = data.split('\n');
        const channels = [];
        let currentGroups = {};
        for (let i = 0; i < lines.length; i++) {
            if (lines[i].startsWith('#EXTINF:')) {
                const info = lines[i];
                const url = lines[i + 1]?.trim();
                if (url && (url.startsWith('http') ||
url.startsWith('//'))) {
                    let title = info.match(/,(.+)/)?.[1]?.trim() ||
'Unknown';
                    const logo =
info.match(/tvlg-logo="([^\"]+)/)?.[1];
                    let group =
info.match(/group-title="([^\"]+)/)?.[1]?.trim() || 'Uncategorized';
                    const channelData = { title, logo, url, group };
                    channels.push(channelData);
                    if (!currentGroups[group]) {
                        currentGroups[group] = [];
                    }
                    currentGroups[group].push(channelData);
                }
            }
        }
        groupedChannels = currentGroups;
        allChannelsInPlaylist = channels;
    }

    async function loadDataForSource(sourceKey) {
        elements.playlistElement.innerHTML = `<div class="flex
justify-center items-center h-full"><div class="loader"></div></div>`;
        allChannelsInPlaylist = [];
        groupedChannels = {};

        if (sourceKey === 'Favorites') {
            if (favorites.length === 0) {
                elements.playlistElement.innerHTML = `<p class="p-4
text-center text-[var(--text-muted)]">No favorites yet.</p>`;
                return;
            }
            allChannelsInPlaylist = favorites;
            populatePlaylistUI(favorites, false);
            return;
        }
    }

```

```

        if (sourceKey === 'Recently Watched') {
            if (recentlyWatched.length === 0) {
                elements.playlistElement.innerHTML = `
```

```

    }
  }
  const data = await fetchWithFallback(playlist.url);
  localStorage.setItem(cacheKey, JSON.stringify({ timestamp:
Date.now(), data: data }));
  await parseM3U(data, playlist.name);
  await displayCategories();
} catch (error) {
  console.error(`Failed to load playlist ${playlist.name}`,
error);
  elements.playlistElement.innerHTML = `
```

```

        allChannelsDiv.innerHTML = `

<span>All Channels
(${allChannelsInPlaylist.length})</span>`;
        allChannelsDiv.addEventListener('click', () =>
populatePlaylistUI(allChannelsInPlaylist, true));
        elements.playlistElement.appendChild(allChannelsDiv);
    }

    const sortedCategories =
Object.entries(groupedChannels).sort(([a], [b]) =>
a.localeCompare(b));

    if (sortedCategories.length === 1) {
        populatePlaylistUI(sortedCategories[0][1], false);
        return;
    }

    for (const [category, channels] of sortedCategories) {
        const categoryDiv = document.createElement('div');
        categoryDiv.className = 'flex items-center space-x-3 p-3
cursor-pointer hover:bg-purple-500/20 transition-colors duration-200
border-b border-[var(--border-color)]';
        const sortedIconKeys = Object.keys(categoryIcons).sort((a,
b) => b.length - a.length);
        const iconKey = sortedIconKeys.find(key =>
category.toLowerCase().includes(key.toLowerCase())) || 'Default';
        const iconHTML = categoryIcons[iconKey];
        categoryDiv.innerHTML = `


```

```

`https://api.themoviedb.org/3/search/movie?api_key=${TMDB_API_KEY}&que
ry=${encodeURIComponent(searchTerm)}`;
    if (year) searchUrl += `&year=${year}`;
    try {
        const response = await fetch(searchUrl);
        const data = await response.json();
        if (data.results && data.results.length > 0 &&
data.results[0].poster_path) {
            return
`https://image.tmdb.org/t/p/w200${data.results[0].poster_path}`;
        }
        return null;
    } catch (error) {
        console.error('Error fetching movie poster:', error);
        return null;
    }
}

async function fetchTvShowPoster(seriesName) {
    if (!TMDB_API_KEY) return null;
    const searchUrl =
`https://api.themoviedb.org/3/search/tv?api_key=${TMDB_API_KEY}&query=
${encodeURIComponent(seriesName)}`;
    try {
        const response = await fetch(searchUrl);
        const data = await response.json();
        if (data.results && data.results.length > 0 &&
data.results[0].poster_path) {
            return
`https://image.tmdb.org/t/p/w200${data.results[0].poster_path}`;
        }
        return null;
    } catch (error) {
        console.error('Error fetching TV show poster:', error);
        return null;
    }
}

function populatePlaylistUI(channels, showBackButton = false) {
    elements.playlistElement.innerHTML = '';
    if (showBackButton) {
        const backButton = document.createElement('button');
        backButton.innerHTML = `<svg
xmlns="http://www.w3.org/2000/svg" width="20" height="20" viewBox="0 0
24 24" fill="none" stroke="currentColor" stroke-width="2"
stroke-linecap="round" stroke-linejoin="round"><polyline points="15 18
9 12 15 6"></polyline></svg> Back`;
        backButton.className = 'w-full flex items-center space-x-2

```

```

p-3 text-left text-purple-400 hover:bg-purple-500/20 border-b
border-[var(--border-color)];
    backButton.onclick = displayCategories;
    elements.playlistElement.appendChild(backButton);
}
const listContainer = document.createElement('div');
listContainer.className = 'flex flex-col p-1';
elements.playlistElement.appendChild(listContainer);

const currentPlaylistName =
elements.sourceSelectorText.textContent.trim();
if (currentPlaylistName !== 'YT Movies' && currentPlaylistName
!== 'YT Series' && currentPlaylistName !== 'Recently Watched') {
    channels.sort((a, b) => a.title.localeCompare(b.title));
}

channels.forEach(channel => {
    const channelDiv = document.createElement('div');
    channelDiv.className = 'channel-item flex items-center
space-x-3 p-2 cursor-pointer rounded-lg transition-all duration-200';
    channelDiv.dataset.url = channel.url;
    channelDiv.dataset.channel = JSON.stringify(channel);
    const isFavorite = favorites.some(f => f.url ===
channel.url);
    channelDiv.innerHTML = `
        <div class="flex-shrink-0 w-12 h-12 bg-black/20
rounded-md flex items-center justify-center overflow-hidden">
            <img src="" class="channel-logo w-full h-full
object-cover" style="display: none;">
            <div class="channel-logo-fallback w-full h-full
flex items-center justify-center text-xl font-bold
text-slate-500">${(channel.title || 'L').charAt(0)}</div>
        </div>
        <span class="channel-title flex-1 text-sm
text-slate-300 truncate">${channel.title || 'Loading...'}</span>
        <button class="favorite-btn p-1 text-lg text-slate-600
hover:text-purple-400 transition-colors ${isFavorite ? 'favorite' :
''}" data-channel='${JSON.stringify(channel)}'>★</button>
    `;
    listContainer.appendChild(channelDiv);

    const imgElement =
channelDiv.querySelector('.channel-logo');
    const fallbackDiv =
channelDiv.querySelector('.channel-logo-fallback');

    const fetchAndUpdatePoster = async () => {
        let posterUrl = null;

```

```

        const sourceName = channel.sourcePlaylistName ||
currentPlaylistName;
        const groupName = channel.group;

        if (sourceName === 'Public Domain') {
            if (groupName === 'Gargoyles') {
                posterUrl = await
fetchTvShowPoster(groupName);
            } else {
                posterUrl = await
fetchMoviePoster(channel.title);
            }
        } else if (sourceName === 'YT Series') {
            posterUrl = await fetchTvShowPoster(groupName ||
channel.title);
        }

        if (posterUrl) {
            imgElement.src = posterUrl;
            imgElement.style.display = 'block';
            fallbackDiv.style.display = 'none';
        }
    };

    if (['Public Domain', 'YT
Series'].includes(channel.sourcePlaylistName) || ['Public Domain', 'YT
Series'].includes(currentPlaylistName)) {
        fetchAndUpdatePoster();
    } else if (channel.logo) {
        imgElement.src = channel.logo;
        imgElement.style.display = 'block';
        fallbackDiv.style.display = 'none';
    }
});
}

async function populateFeaturedChannels() {
    const featuredGrid = elements.featuredChannelsGrid;
    if (!featuredGrid) return;

    let channelPool = [];
    const promises = PLAYLISTS.map(async (playlist) => {
        let channels = [];
        if (playlist.channels) {
            channels = playlist.channels.map(c => ({...c,
sourcePlaylistName: playlist.name}));
        } else if (playlist.url) {
            try {

```

```

        const cacheKey = `iptv_cache_${playlist.name}`;
        let data;
        const cachedItem = localStorage.getItem(cacheKey);
        if (cachedItem) {
            const { timestamp, data: cachedData } =
JSON.parse(cachedItem);
            if ((Date.now() - timestamp) <
CACHE_EXPIRATION_MS) {
                data = cachedData;
            }
        }
        if (!data) {
            data = await fetchWithFallback(playlist.url);
            localStorage.setItem(cacheKey,
JSON.stringify({ timestamp: Date.now(), data: data }));
        }
        await parseM3U(data, playlist.name);
        channels = allChannelsInPlaylist.map(c => ({...c,
sourcePlaylistName: playlist.name}));
    } catch (error) {
        console.error(`Could not fetch featured playlist
${playlist.name}:`, error);
    }
}
return channels;
});

const results = await Promise.all(promises);
channelPool = results.flat();

const priorityPlaylistNames = ['YT Movies', 'YT Series', 'YT
Live', 'Public Domain'];
let priorityPool = channelPool.filter(c =>
priorityPlaylistNames.includes(c.sourcePlaylistName));
let m3uPool = channelPool.filter(c =>
!priorityPlaylistNames.includes(c.sourcePlaylistName));

priorityPool.sort(() => 0.5 - Math.random());
m3uPool.sort(() => 0.5 - Math.random());

const prioritySelection = priorityPool.slice(0, 8);
const m3uSelection = m3uPool.slice(0, 2);

const selectedChannels = [...prioritySelection,
...m3uSelection].sort(() => 0.5 - Math.random());

featuredGrid.innerHTML = '';
for (const channel of selectedChannels) {

```



```

        const caseDiv = document.createElement('div');
        caseDiv.className = 'dvd-case';
        caseDiv.dataset.channel = JSON.stringify(channel); //
Store full channel object

        let logoUrl = '';
        if (channel.sourcePlaylistName === 'Public Domain') {
            if (channel.group === 'Gargoyles') {
                logoUrl = await fetchTvShowPoster(channel.group);
            } else {
                logoUrl = await fetchMoviePoster(channel.title);
            }
        } else if (channel.sourcePlaylistName === 'YT Series') {
            logoUrl = await fetchTvShowPoster(channel.group ||
channel.title);
        } else if (channel.logo) {
            const sourcePlaylist = PLAYLISTS.find(p => p.name ===
channel.sourcePlaylistName);
            if (sourcePlaylist && sourcePlaylist.url) { // It's an
M3U channel
                logoUrl = CORS_PROXIES[0] +
encodeURIComponent(channel.logo);
            } else { // It's a local playlist like YT Movies
                logoUrl = channel.logo;
            }
        }

        caseDiv.innerHTML = `
            <div class="dvd-case-img-container">
                
                <div class="channel-logo-fallback w-full h-full
flex items-center justify-center text-2xl font-bold text-slate-500"
style="${logoUrl ? 'display:none;' : ''}">${(channel.title ||
'L').charAt(0)}</div>
            </div>
            <div class="dvd-case-title">${channel.title}</div>
        `;

        caseDiv.addEventListener('click', () => {
            const clickedChannel =
JSON.parse(caseDiv.dataset.channel);
            handleChannelSelection(clickedChannel);
            goToChannel(clickedChannel);
        });
        featuredGrid.appendChild(caseDiv);
    }

```

```

    }

    async function goToChannel(channel) {
        document.querySelector('main').scrollTo({ top: 0, behavior:
'smooth' });

        const sourcePlaylist = allPlaylists.find(p => p.name ===
channel.sourcePlaylistName);
        if (!sourcePlaylist) return;

        // Switch to the correct playlist
        updateSourceSelectorButton(sourcePlaylist);
        await loadDataForSource(sourcePlaylist.name);

        // If the playlist is categorized, go into the sub-category
        if (sourcePlaylist.categorized && channel.group) {
            const categoryChannels = groupedChannels[channel.group];
            if (categoryChannels) {
                populatePlaylistUI(categoryChannels, true);
            }
        }

        // Highlight and scroll to the channel
        setTimeout(() => {
            document.querySelectorAll('.channel-item').forEach(item =>
item.classList.remove('active'));
            const targetChannelElement =
document.querySelector(`.channel-item[data-url="${CSS.escape(channel.u
rl)}"]`);
            if (targetChannelElement) {
                targetChannelElement.classList.add('active');
                targetChannelElement.scrollIntoView({ behavior:
'smooth', block: 'center' });
            }
        }, 200); // A small delay to ensure the DOM is updated
    }

    function updateRecentlyWatched(channel) {
        // Find and remove the channel if it already exists to move it
to the front
        const existingIndex = recentlyWatched.findIndex(c => c.url ===
channel.url);
        if (existingIndex > -1) {
            recentlyWatched.splice(existingIndex, 1);
        }
        // Add the new channel to the beginning of the array
        recentlyWatched.unshift(channel);
        // Limit the list to a reasonable number, e.g., 20
    }

```

```

        if (recentlyWatched.length > 20) {
            recentlyWatched.pop();
        }
        // Save to localStorage
        localStorage.setItem('iptv_recently_watched',
JSON.stringify(recentlyWatched));
    }

    function handleChannelSelection(channel) {
        if (!channel || !channel.url) {
            console.error("Invalid channel object provided:",
channel);
            return;
        }

        updateRecentlyWatched(channel);

        channelClickCounter++;
        if (channelClickCounter > 0 && channelClickCounter % 10 === 0)
{
    window.open('https://www.profitableratecpm.com/qb069j7b6?key=e6b8dcd0a
21df19017732330d1236fd3', '_blank');
            urlToPlayAfterAd = channel.url;
            showAd();
        } else {
            playStream(channel.url);
        }
    }

    function extractVideoID(url) {
        if (!url) return null;
        const regex =
/(?:(?:https?:\/\/)?(?:www\.)?(?:youtube\.com\/(?:^\\/\n\s]+\:\/\/S+\/|(?:(v|
e(?:mbed)?|live)\|S*?[?&]v=)|youtu\.be\/)([a-zA-Z0-9_-]{11}))/;
        const match = url.match(regex);
        return match ? match[1] : null;
    }

    function playStream(url) {
        hideErrorOverlay();
        const videoId = extractVideoID(url);
        if (videoId) {
            activePlayer = 'youtube';
            elements.mainPlayer.classList.add('hidden');

elements.youtubePlayerContainer.classList.remove('hidden');
            if (hls) hls.stopLoad();

```

```

        elements.mainPlayer.pause();
        elements.mainPlayer.removeAttribute('src');
        elements.youtubePlayerContainer.innerHTML = ``; // Clear
previous
        const iframe = document.createElement('iframe');
        iframe.setAttribute('src',
`https://www.youtube.com/embed/${videoId}?autoplay=1&rel=0&modestbrand
ing=1`);
        iframe.setAttribute('frameborder', '0');
        iframe.setAttribute('allow', 'accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture');
        iframe.setAttribute('allowfullscreen', '');
        elements.youtubePlayerContainer.appendChild(iframe);
    } else {
        activePlayer = 'hls';
        elements.youtubePlayerContainer.classList.add('hidden');
        elements.youtubePlayerContainer.innerHTML = '';
        elements.mainPlayer.classList.remove('hidden');
        let finalUrl = url;
        if (url.endsWith('.m3u8') && Hls.isSupported()) {
            hls.loadSource(finalUrl);
            hls.once(Hls.Events.MANIFEST_PARSED, () =>
elements.mainPlayer.play().catch(e => console.error("Playback error:",
e)));
        } else {
            if (hls) hls.stopLoad(); // Stop HLS if playing a
non-HLS stream
            elements.mainPlayer.src = finalUrl;
            elements.mainPlayer.play().catch(e =>
console.error("Playback error:", e));
        }
    }
    if (lastClickedChannel) {
        localStorage.setItem('iptv_last_played_channel',
JSON.stringify(lastClickedChannel));
    }
}

function toggleFavorite(button) {
    const channelData = JSON.parse(button.dataset.channel);
    const index = favorites.findIndex(f => f.url ===
channelData.url);
    if (index !== -1) {
        favorites.splice(index, 1);
        button.classList.remove('favorite');
    } else {
        favorites.push(channelData);
        button.classList.add('favorite');
    }
}

```

```

        }
        localStorage.setItem('iptv_favorites',
JSON.stringify(favorites));
        if
(elements.sourceSelectorText.textContent.trim().includes('Favorites'))
{
            loadDataForSource('Favorites');
        }
    }

function enterFullscreen() {
    let playerElement;
    if (activePlayer === 'youtube') {
        playerElement =
elements.youtubePlayerContainer.querySelector('iframe');
    } else {
        playerElement = elements.mainPlayer;
    }
    if (playerElement) {
        if (playerElement.requestFullscreen)
playerElement.requestFullscreen();
        else if (playerElement.mozRequestFullScreen)
playerElement.mozRequestFullScreen();
        else if (playerElement.webkitRequestFullscreen)
playerElement.webkitRequestFullscreen();
        else if (playerElement.msRequestFullscreen)
playerElement.msRequestFullscreen();
    }
}

function updateSourceSelectorButton(playlist) {
    elements.sourceSelectorIcon.innerHTML = `<span
class="text-xl">${playlist.icon || '★'}</span>`;
    elements.sourceSelectorText.textContent = playlist.name;
};

const favoritesPlaylist = { name: 'Favorites', icon: '★' };
const recentlyWatchedPlaylist = { name: 'Recently Watched', icon:
'🕒' };
const allPlaylists = [favoritesPlaylist, recentlyWatchedPlaylist,
...PLAYLISTS];

function renderPlaylistOptions(playlistsToRender) {
    elements.playlistOptionsContainer.innerHTML = '';
    playlistsToRender.forEach(playlist => {
        const optionDiv = document.createElement('div');
        optionDiv.className = 'flex items-center space-x-2 p-2
cursor-pointer hover:bg-purple-500/20';
    });
}

```

```

        optionDiv.innerHTML = `<span class="flex-shrink-0 w-6
text-center text-xl">${playlist.icon || ''}</span> <span
class="truncate">${playlist.name}</span>`;
        optionDiv.addEventListener('click', () => {
            localStorage.setItem('iptv_last_source',
playlist.name);
            updateSourceSelectorButton(playlist);
            loadDataForSource(playlist.name);

elements.sourceSelectorDropdown.classList.add('hidden');
        });
        elements.playlistOptionsContainer.appendChild(optionDiv);
    });
}

function initialize() {
    initializePlayers();
    renderPlaylistOptions(allPlaylists);
    populateFeaturedChannels();

    elements.sourceSelectorButton.addEventListener('click', () =>
{
elements.sourceSelectorDropdown.classList.toggle('hidden');
    });

    document.addEventListener('click', (e) => {
        if (!elements.sourceSelectorButton.contains(e.target) &&
!elements.sourceSelectorDropdown.contains(e.target)) {
elements.sourceSelectorDropdown.classList.add('hidden');
        }
    });

    elements.playlistSearchInput.addEventListener('click', (e) =>
e.stopPropagation());

    elements.playlistSearchInput.addEventListener('input', e => {
        const searchTerm = e.target.value.toLowerCase().trim();
        const filtered = allPlaylists.filter(p =>
p.name.toLowerCase().includes(searchTerm));
        renderPlaylistOptions(filtered);
    });

    elements.searchInput.addEventListener('input', e => {
        const searchTerm = e.target.value.toLowerCase().trim();
        if (!searchTerm) {
            displayCategories();

```

```

        return;
    }
    const filteredChannels = allChannelsInPlaylist.filter(c =>
c.title.toLowerCase().includes(searchTerm));
    populatePlaylistUI(filteredChannels, true);
});

elements.closeAdBtn.addEventListener('click',
hideAdAndOpenLink);
elements.adImage.addEventListener('click', hideAdAndOpenLink);
elements.playlistElement.addEventListener('click',
handlePlaylistClick);

elements.menuToggle.addEventListener('click', () => {
    elements.mobileOverlay.classList.toggle('hidden');
    elements.sidebar.classList.toggle('-translate-x-full');
});

elements.mobileOverlay.addEventListener('click', () => {
    elements.sidebar.classList.add('-translate-x-full');
    elements.mobileOverlay.classList.add('hidden');
});

const lastSourceName =
localStorage.getItem('iptv_last_source') || PLAYLISTS[0]?.name ||
'Favorites';
const lastSourcePlaylist = allPlaylists.find(p => p.name ===
lastSourceName) || favoritesPlaylist;
updateSourceSelectorButton(lastSourcePlaylist);
loadDataForSource(lastSourceName);

if (!sessionStorage.getItem('adShownThisSession')) {
    showAd(true);
    sessionStorage.setItem('adShownThisSession', 'true');
}
}

function handlePlaylistClick(event) {
    const favoriteBtn = event.target.closest('.favorite-btn');
    if (favoriteBtn) {
        event.stopPropagation();
        toggleFavorite(favoriteBtn);
        return;
    }

    const channelItem = event.target.closest('.channel-item');
    if (channelItem) {
        const channel = JSON.parse(channelItem.dataset.channel);

```

```

        if (lastClickedChannel && lastClickedChannel.url ===
channel.url) {
            if (activePlayer !== 'youtube') {
                enterFullscreen();
            }
            if (window.innerWidth < 768) {

elements.sidebar.classList.add('-translate-x-full');
                elements.mobileOverlay.classList.add('hidden');
            }
            lastClickedChannel = null; // Reset for next double
click
        } else {

document.querySelectorAll('.channel-item').forEach(item =>
item.classList.remove('active'));
            channelItem.classList.add('active');
            lastClickedChannel = channel;
            handleChannelSelection(channel);
        }
    }
}

        initialize();
    });
</script>
</body>
</html>

```