

**Movie Recommendation App using React**



Team Members

* Abdul Moamen Youssef Mohamed
* Abdul Rahman Gamal Abdul Rahman Ahmed
* Abdul Rahman Mohamed Helmy Shaban
* Hassan Khaled Hassan Mohamed
* Mohamed Sameh Mohamed Abd El Zaher
* Mohamed Mahmoud Taher Mohamed

Index:

**Project Description………………………………………………………… 4**

**Project Plan………………………………………………………………… 5**

**Project Milestones and Deliverables……………………………………… 6**

**Requirements, System Analysis and Design……………………………… 7**

**User Story…………………………………………………………………… 7**

**Use Cases…………………………………………………………………… 8**

**Functional and Non-Functional Requirements…………………………… 11**

**Data Flow & System Behaviour…………………………………………… 12**

**UX Design Principles……………………………………………………… 14**

**UI Design Mockups and Prototyping……………………………………… 15**

**Tools and Technologies…………………………………………………… 16**

**Project Description:**

* **Overview:**

We propose the development of a state-of-the-art movie streaming platform, inspired by industry leaders like Amazon Prime, to deliver a seamless and engaging cinematic experience. This platform will feature a diverse library of films, personalized recommendations, and high-quality streaming across multiple devices.

* **Description**:

Create an interactive app that provides users with movie recommendations based on genres, and user preferences. Use React to create an intuitive user interface, with features like search, filters (genre, year, rating), and pagination for browsing movies.

**Objectives:**

* Build a movie recommendation application with React, fetching movie data from a third-party API
* Deliver an intuitive, user-friendly interface for effortless navigation.
* Offer a diverse catalog of movies, including blockbusters, indie films, and international cinema.
* Ensure accessibility across smartphones, tablets, desktops, and smart TVs.
* Foster community engagement through user reviews, ratings, and watchlists.

**Project Plan:**

**Week 1: Planning & Design:**

* **Task 1: Project Kickoff & Requirements Review**
  + Duration: 2 days
  + Dependency: None (starts the project)
* **Task 2: UI/UX Design Mockups**
  + Duration: 4 days
  + Dependency: Task 1 completion
* **Task 3: Frontend Technical Specs**
  + Duration: 4 days
  + Dependency: Task 1 completion (assumes API docs provided)
* **Milestone 1: Design and Specs Approved**
  + Finalized UI and technical plan.

**Week 2: Development – Core Frontend:**

* **Task 4: Build Widget UI Components**
  + Duration: 6 days
  + Dependency: Task 2 completion
* **Task 5: Basic API Integration**
  + Duration: 4 days
  + Dependency: Task 3 completion (assumes backend API ready)
* **Milestone 2: Initial Widget Functional**
  + Basic version shows recommendations.

**Week 3: Refinement & Integration**

* **Task 6: Enhance UI/UX (Polishing)**
  + Duration: 4 days
  + Dependency: Task 4 completion
* **Task 7: Full API Integration & Error Handling**
  + Duration: 5 days
  + Dependency: Task 5 completion
* **Milestone 3: Frontend Feature Complete**
  + Fully functional widget ready for testing.

**Week 4: Testing & Deployment**

* **Task 8: QA Testing & Bug Fixes**
  + Duration: 4 days
  + Dependency: Task 7 completion
* **Task 9: User Acceptance Testing (UAT)**
  + Duration: 3 days
  + Dependency: Task 8 partial completion
* **Task 10: Deployment to Production**
  + Duration: 2 days
  + Dependency: Task 9 completion
* **Milestone 4: Widget Launched**
  + Live on the website.

**Project Milestones and Deliverables:**

* **Milestones:**

1. **Design and Specs Approved:** UI mockups and technical approach finalized.
2. **Initial Widget Functional:** Core frontend displays recommendations.
3. **Frontend Feature Complete:** Polished and integrated widget.
4. **Widget Launched:** Deployed and accessible to users.

* **Deliverables:**

**Week 1:**

* + Project plan document (frontend scope).
  + UI/UX mockups (PDF/image files).
  + Frontend technical specification (e.g., component structure).

**Week 2:**

* + Initial widget UI (codebase: HTML/CSS/JavaScript or framework like React).
  + Basic API-integrated demo (internal preview).

**Week 3:**

* + Polished widget UI (refined codebase).
  + Fully integrated feature demo (handles edge cases).

**Week 4:**

* + Tested and finalized widget (production-ready code).
  + Deployment confirmation (launch summary).

**Roles Assignment and Resource Allocation:**

* **Project Manager (PM):**
  + 15 hours/week (oversight, coordination).
  + **Key Tasks:** Kickoff, milestone reviews, deployment.
* **UI/UX Designer:**
  + 25 hours (Week 1 focus, minor Week 3 support).
  + **Key Tasks:** Mockups, refinement feedback.
* **Frontend Developer (2x):**
  + **Dev 1:** 40 hours (Weeks 2-3 focus).
    - **Key Tasks:** Widget UI build, polishing.
  + **Dev 2:** 40 hours (Weeks 2-3 focus).
    - **Key Tasks:** API integration, error handling.
* **QA Engineer:**
  + 25 hours (Week 4 focus, some Week 3 overlap).
  + **Key Tasks:** Testing, bug fixes.

**Requirements, System Analysis and Design**

**User Story:**

As a movie enthusiast (Hamdi), I'd like to be able to:

* Browse movies and shows organized into categories (like Action, Drama, or Sci-Fi).
* Make personalized watchlists to keep track of movies, TV series, live events, and upcoming releases.
* See the rating for each show so I can choose what to watch.

This will help me:

* Plan my movie nights based on what I'm in the mood for.
* Find good options to watch with family and friends, taking everyone's tastes into account.
* Decide what to watch based on ratings and other details.

**Acceptance Criteria:**

* The app should have pre-made categories for movies and shows (e.g., Action, Drama, Science Fiction).
* I should be able to create my own watchlists and give them names (e.g., "Family Movie Night," "Must-Watch Series").
* Adding movies, shows, live events, and upcoming events to my watchlists should be easy.
* I should be able to remove things from my watchlists.
* I should be able to see all my watchlists and easily find what's in them.
* My watchlists should be saved so I can see them every time I use the app.
* The app should show the rating (like the IMDb rating) for each movie and show.

**Use Cases**

**Use Case 1: Browsing Movies and TV Shows**

A diagram of a movie streaming application

AI-generated content may be incorrect.

* **Use Case Name:** Browse Content
* **Actor:** User
* **Pre-Conditions:** User is on the app's home screen or a browse/catalog screen.
* **Main Success Scenario:**
  1. User selects a browsing category (e.g., Genre, Trending, New Releases).
  2. System displays a list of movies and TV shows matching the selected category.
  3. User can further refine the browsing by applying filters (e.g., by sub-genre, release year, rating).
  4. System updates the display with the filtered results.
  5. User can scroll through the results and select a movie/show to view details.
* **Alternative Flows (Extensions):**
  1. If no results match the selected criteria, the system displays a "No results found" message.
* **Post-Conditions:** User has viewed a list of movies and TV shows based on the selected criteria.

**Use Case 2: Searching for Content**

A diagram of a movie streaming application

AI-generated content may be incorrect.

* **Use Case Name:** Search Content
* **Actor:** User
* **Pre-Conditions:** User is on the app's home screen or a search screen.
* **Main Success Scenario:**
  1. User enters a search term (e.g., movie title, actor name, keyword) in the search bar.
  2. User initiates the search.
  3. System displays a list of movies and TV shows matching the search term.
  4. User can select a movie/show from the search results to view details.
* **Alternative Flows (Extensions):**
  1. If no results match the search term, the system displays a "No results found" message.
* **Post-Conditions:** User has viewed a list of search results.

**Use Case 3: Viewing Detailed Information**

A diagram of a movie streaming application

AI-generated content may be incorrect.

* **Use Case Name:** View Details
* **Actor:** User
* **Pre-Conditions:** User has selected a movie or TV show from a browse list or search results.
* **Main Success Scenario:**
  1. User clicks/taps on a movie/show listing.
  2. System displays detailed information about the selected movie/show (synopsis, cast, ratings, etc.).
* **Post-Conditions:** User has viewed detailed information about a specific movie/show.

**Use Case 4: Selecting Preferred Genres**

A diagram of a movie streaming application

AI-generated content may be incorrect.

* **Use Case Name:** Select Genres
* **Actor:** User
* **Pre-Conditions:** User is logged in.
* **Main Success Scenario:**
  1. User navigates to the "Preferences" or "Profile" settings.
  2. User selects their preferred genres from a list of available genres.
  3. User saves their genre preferences.
* **Post-Conditions:** User's genre preferences are saved.

**Use Case 5: Creating and Managing Watchlists**

A diagram of a watchlist

AI-generated content may be incorrect.

* **Use Case Name:** Manage Watchlists
* **Actor:** User
* **Pre-Conditions:** User is logged in.
* **Main Success Scenario:**
  1. User navigates to the "My Watchlists" section.
  2. User can create a new watchlist by providing a name.
  3. User can add movies/shows to a watchlist from browse lists, search results, or details pages.
  4. User can remove movies/shows from a watchlist.
  5. User can view and organize their watchlists.
* **Post-Conditions:** User has created, modified, or viewed their watchlists.

**Functional and Non-Functional Requirements**

**1. Functional Requirements:**

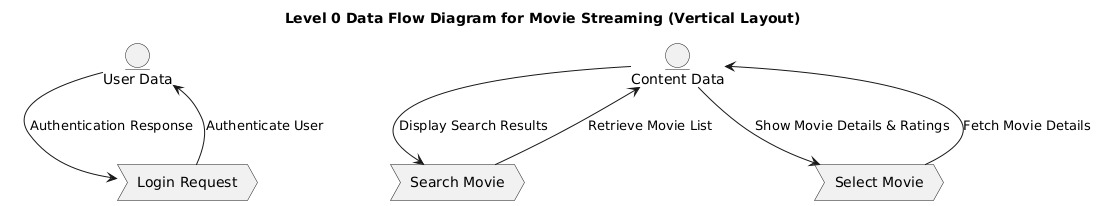
* Users should be able to look through movies and shows by category (like Action or Comedy), title, actor, and other things.
* Users should be able to find specific movies and shows by searching.
* Users should be able to see all the details about a movie or show (story summary, actors, ratings, etc.).
* Users should be able to make and organize different watchlists.
* Users should be able to pick the types of movies and shows they like.

**2. Non-Functional Requirements:**

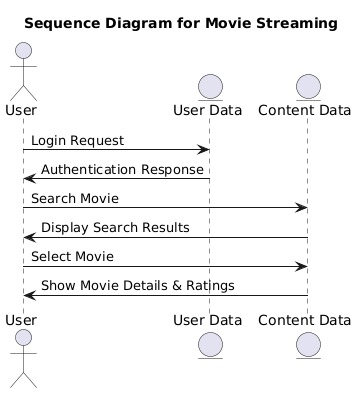
* Movie and show details should load fast (in less than 2 seconds).
* Search results should pop up almost right away (in less than 1 second).
* Streaming should work smoothly without any buffering if you have a decent internet connection.
* The app should be simple and easy for anyone to use.
* User accounts and payment info should be totally secure.
* The system should be safe from online attacks.

**Data Flow & System Behaviour**

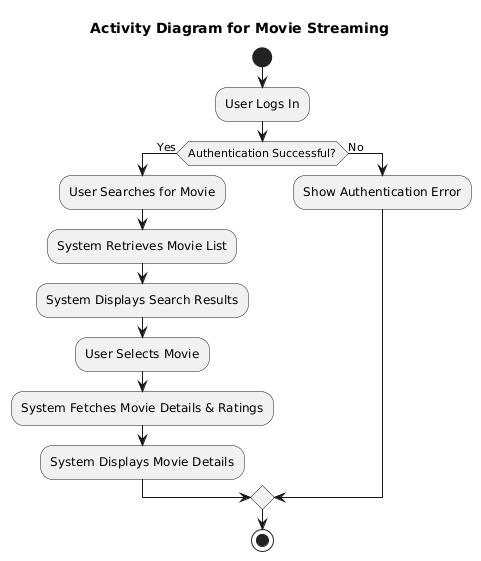
* **DFD (Data Flow Diagram) – Context-level and detailed levels showing how data moves through the system.**



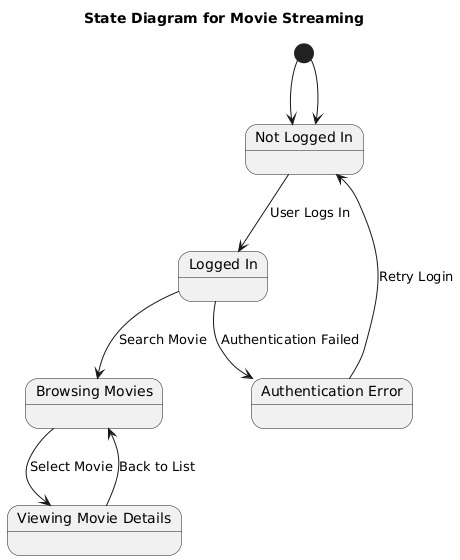
* **Sequence Diagrams – Process flow representation of key interactions between components.**



* **Activity Diagram – Visualizing the workflow of processes or user actions within the system.**



* **State Diagram – Represents different states of an object and how it transitions between them.**



**UX Design Principles**

The design of the movie recommendation app follows these core principles to ensure a seamless, engaging, and user-friendly experience:

**1. User-Centric Design**

* The interface is intuitive, allowing users to explore content easily
* Prioritizes accessibility with clear typography, high contrast, and easy navigation.

**2. Consistency and Visual Hierarchy**

* A uniform design language is maintained across all screens.
* Clear visual hierarchy guides users to key actions (add to watchlist, search).
* Responsive design ensures optimal viewing on different screen sizes and devices.

**3. Performance and Scalability**

* Fast-loading UI elements and optimized media delivery for smooth playback.
* Supports scalability to accommodate a growing user base without performance loss.

**4. Security and Privacy**

* Ensures user data protection with privacy policies.
* Provides parental controls and content restrictions for safe family viewing.

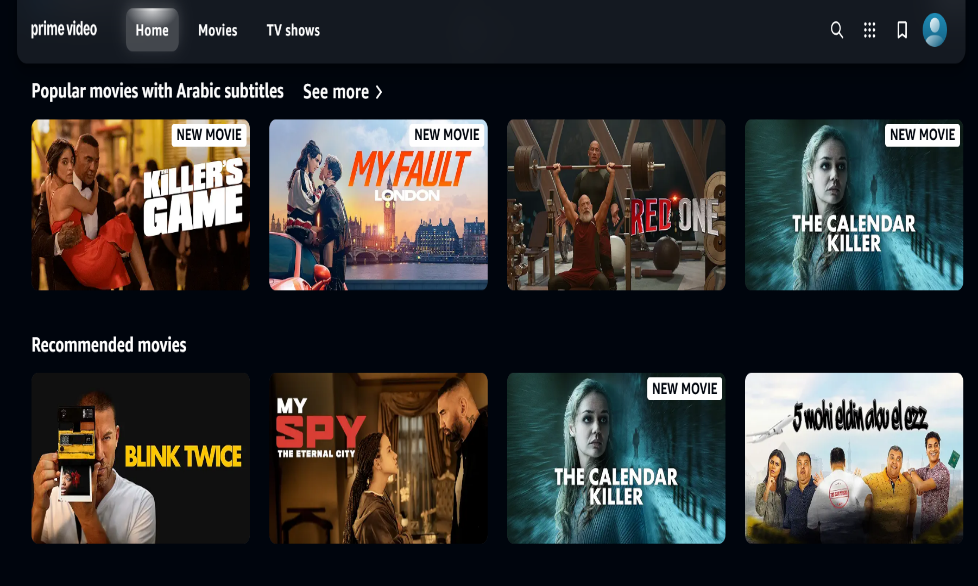
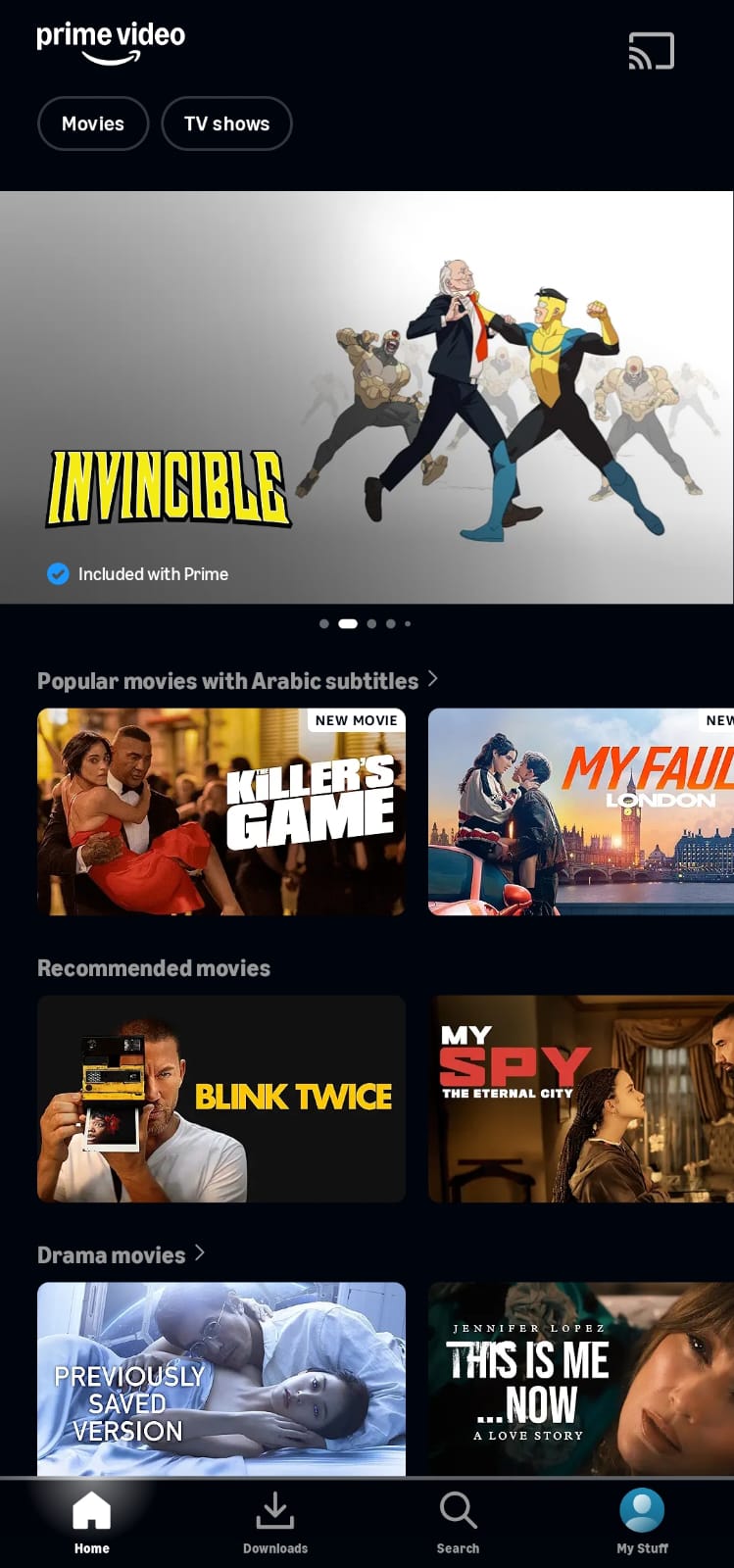
**5. Accessibility**

* Supports multiple languages and subtitle options.
* Provides alternative text for images and intuitive keyboard navigation.

**6. Engagement and Retention**

* Interactive features like ratings, reviews, and community discussions encourage engagement.

**UI Design Mockups and Prototyping**

****

**Tools and Technologies**

**Tools:**

* Figma (design)
* Git (version control)
* HTML/CSS/Javascript (development Languages)
* React/Vue (development Framework)
* Bootstrap (Styling Framework)
* Postman (API testing)
* Jira (tracking).

**Assumptions:**

* Backend team provides API support; hosting via existing infrastructure.