High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **Streamly - Entertainment Streaming UI**. Streamly is a modern, responsive frontend for an entertainment streaming service, designed with React and Tailwind CSS to deliver a seamless, futuristic user experience.

---

1. System Architecture Overview

Architecture Summary:
Streamly is a single-page application (SPA) built with React. It interacts with backend APIs to fetch and display media content. The UI is modular, responsive, and styled using Tailwind CSS.

Module/Component	Role/Responsibility
App Shell	Root component, routing, layout management
Media Grid	Displays media items in grid/list formats
Media Item Card	Shows individual media details (image, title, actions)
Search & Filter Bar	Enables searching and filtering of media content
Media Detail View	Detailed view for selected media item
API Service Layer	Handles API requests and data fetching
State Management	Manages UI and data state (React Context/State)
Responsive Layout	Ensures UI adapts to all device sizes

---

2. Component Interactions

Source Component	Target Component	Interaction Description
App Shell	All child components	Provides layout, navigation, and context
Search & Filter Bar	Media Grid	Updates displayed media based on user input
Media Grid	Media Item Card	Renders a list of media cards with fetched data
Media Item Card	Media Detail View	Triggers detail view on user selection
API Service Layer	State Management	Supplies fetched data to update UI state

Flow Summary:
User interacts with Search/Filter → Media Grid updates → User selects Media Item → Detail View opens. All data is fetched via the API Service Layer and managed in state.

---

3. Data Flow Overview

Data Source	Flow Direction	Data Consumer	Purpose
Backend API	→ API Service Layer	Fetches media data	Supplies media content to frontend
API Service Layer	→ State Management	Stores data	Centralizes data for UI components
State Management	→ UI Components	Renders UI	Drives display and interactivity
User Input	→ Search/Filter Bar	Updates state	Triggers new API requests or filtering

---

4. Technology Stack

Layer/Area	Technology/Framework
UI Framework	React (JavaScript)
Styling	Tailwind CSS, Advanced CSS
Markup	HTML5
State Management	React Context/State
API Communication	Fetch API / Axios
Tooling/Build	Vite or Create React App

---

5. Scalability & Reliability

- **Scalability:**
  Modular React components and stateless UI logic enable easy feature expansion and maintenance. Responsive design ensures compatibility across devices.
- **Reliability:**
  Error boundaries and loading states handle API failures gracefully. Stateless frontend ensures high availability.
- **Security:**
  No sensitive data stored on frontend; all API interactions use secure HTTPS endpoints.

End of Document