Product Requirements & Specification Document

Project Name

Streamly - Entertainment Streaming UI

Description

Streamly is a modern, responsive frontend UI for an entertainment streaming service. It features dynamic media grids, advanced search, and a seamless user experience. The design is futuristic and startup-inspired, leveraging React, JavaScript, HTML, CSS, and Tailwind CSS.

---

1. Goals & Objectives

Goal	Description
Modern UI	Deliver a visually appealing, futuristic interface
Responsive Design	Ensure optimal experience across devices and screen sizes
Efficient Media Browsing	Enable users to browse, search, and filter media content easily
Seamless User Experience	Fast, smooth interactions with minimal load times

---

2. Core Features

Feature	Description
Media Grid	Display movies/shows in a responsive, interactive grid
Search	Real-time search with instant filtering and suggestions
Media Details	Modal or page with detailed info, images, and playback option (mocked)
Categories/Filters	Filter media by genre, popularity, or other tags
Responsive Navigation	Adaptive navigation bar/menu for desktop and mobile
Theming	Futuristic, startup-inspired visuals using Tailwind and custom CSS
Loading States	Skeleton loaders and smooth transitions
Accessibility	Keyboard navigation, ARIA labels, and color contrast compliance

---

3. User Stories

As a…	I want to…	So that…
Visitor	Browse media in a grid	I can discover new content
User	Search for specific titles	I can quickly find what I want

User	View details about a media item	I can learn more before watching
User	Filter media by category	I can find content that matches my taste
Mobile User	Use the app comfortably on my phone	I have a seamless mobile experience

---

4. Technical Specifications

4.1 Stack

Layer	Technology
Framework	React (v18+)
Styling	Tailwind CSS, CSS
Language	JavaScript (ES6+)
Markup	HTML5

4.2 Architecture

- **Component-based** React structure
- **State management**: React hooks (useState, useEffect)
- **Routing**: React Router (if multi-page/modal)
- **Mocked data**: Local JSON or static files for media content

4.3 Key Components

Component	Purpose
`MediaGrid`	Displays media items in a responsive layout
`SearchBar`	Handles user input and filtering
`MediaCard`	Shows summary info for each media item
`MediaDetail`	Modal/page for detailed media info
`Navbar`	Main navigation, adapts to screen size
`CategoryFilter`	Filter controls for genres/tags
`Loader`	Skeleton/loading indicators

---

5. UI/UX Requirements

Requirement	Specification
Theme	Futuristic, clean, high-contrast, startup-inspired
Responsiveness	Mobile-first, supports all major breakpoints
Interactivity	Hover, focus, and active states for all interactive elements
Accessibility	WCAG 2.1 AA compliance, keyboard navigation, ARIA roles

Performance	Fast load, optimized images, minimal re-renders

---

6. Non-Functional Requirements

Requirement	Specification
Code Quality	Follows best practices, linted, well-commented
Documentation	README with setup, usage, and component docs
Testing	Basic unit tests for key components
Browser Support	Latest Chrome, Firefox, Safari, Edge

---

7. Out of Scope

- Backend integration (API, authentication, real playback)
- User accounts or personalization
- Payment or subscription flows

---

8. Acceptance Criteria

- All core features implemented and functional
- Responsive and accessible across devices
- Passes basic usability and accessibility checks
- Codebase is clean, documented, and ready for handoff

---

9. Example Component Structure

```
<App>
  <Navbar />
  <SearchBar />
  <CategoryFilter />
  <MediaGrid>
    <MediaCard />
  </MediaGrid>
  <MediaDetail /> {/* Modal or route */}
  <Loader />
</App>
```

---

10. Milestones & Timeline

Milestone	Description	Target Date
UI Design	Finalize layout & theme	Week 1

Core Components	MediaGrid, MediaCard, Navbar	Week 2
Search & Filtering	SearchBar, CategoryFilter	Week 3
Responsiveness & UX	Mobile, accessibility, loaders	Week 4
Testing & Handoff	QA, documentation, code review	Week 5

End of Document