

Design Document – Netflix’s Show Listing App

Author: Neha Lad

Role: Frontend Developer

Email: neha.s.lad@gmail.com

Date Created: June 27, 2025

1. Overview

This document outlines the frontend design and behavior of the Application. The app allows users to browse a featured show (Billboard), view a grid of available shows, and manage a personalized "My List". It consists of two main views—Home and My List—accessible via tab navigation

2. Goals

- Render an interactive, visually engaging homepage with a featured show (billboard) and a grid of shows.
- Allow users to add or remove shows from their personal list.
- Ensure responsive, accessible, and performant UI interactions.

3. Functional Requirements

Navigation Tabs

- Two tabs: Home and My List
- Black background, white text
- Hover: Simulate a "press" effect
- Click: Apply a white border to the active tab.

Home Page

- Billboard with show image, title, and logline
- Grid of shows below the billboard
- Each grid item:
 1. Displays the show image
 2. On hover: shows an "Add/Remove from My List" button.
 3. On click (outside button): highlights with a white border for 2 seconds.

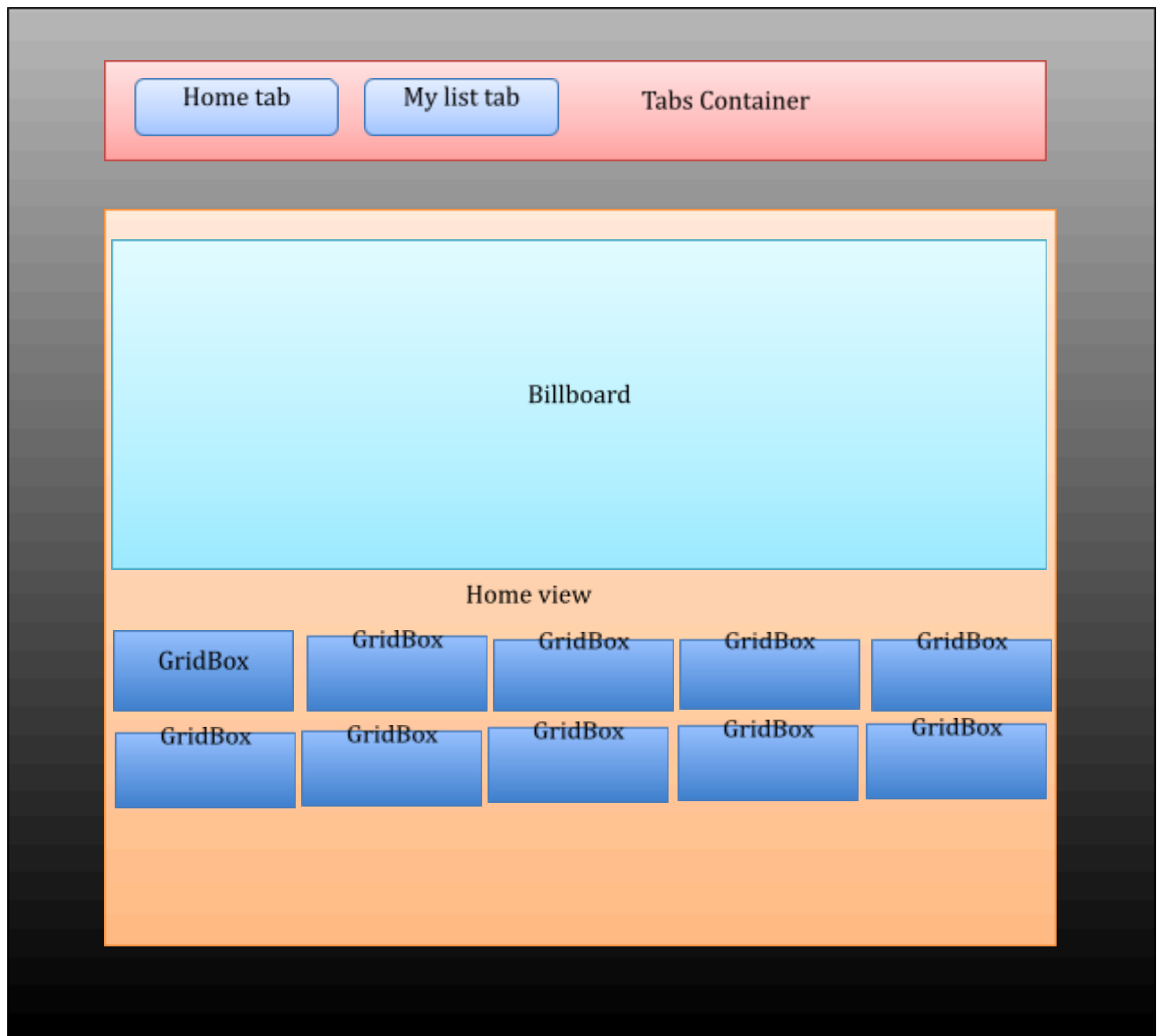
My List Page

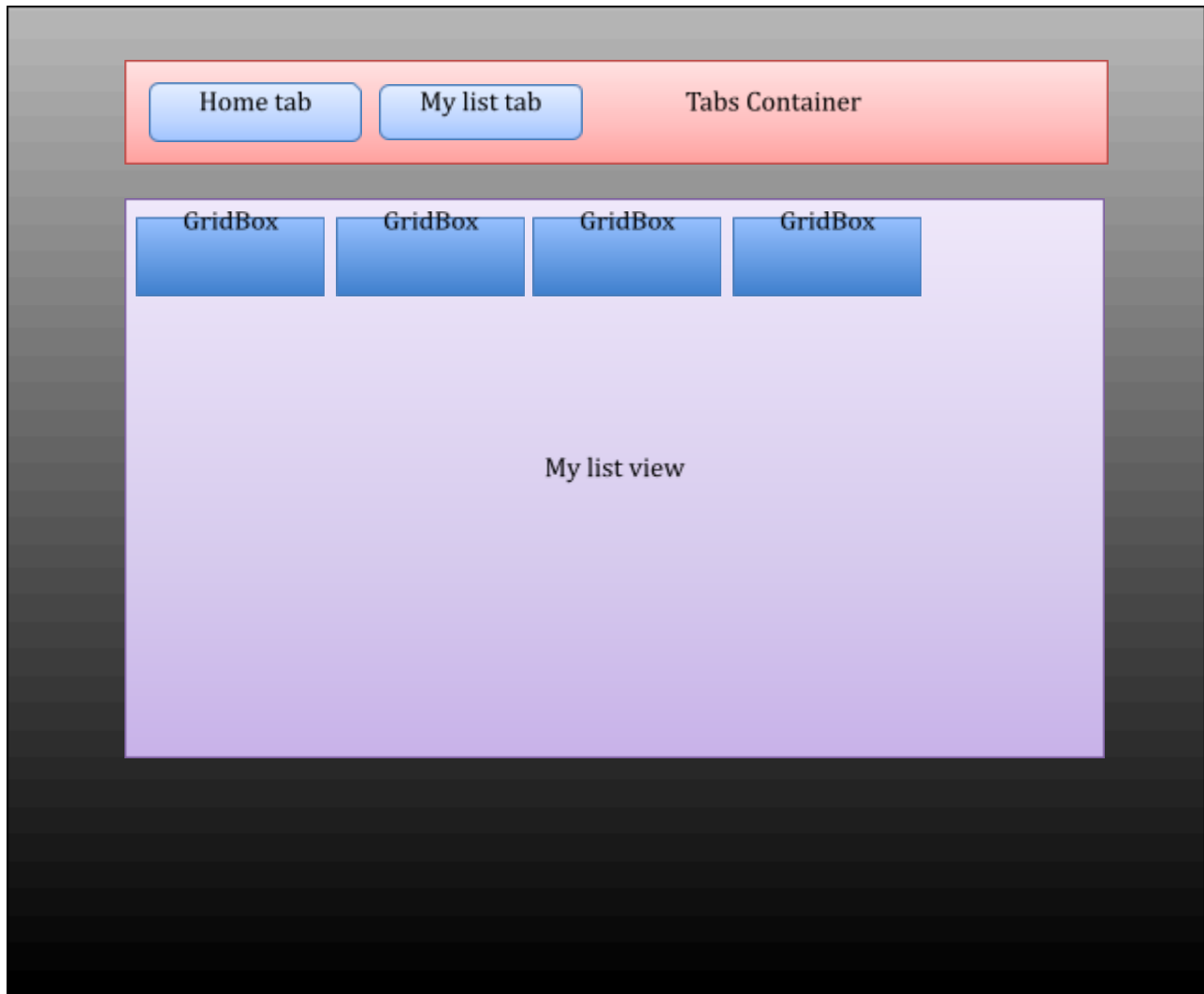
- Displays only the shows added by the user.
- Same hover and click behavior as Home

4. Data Sources

File Name	Purpose
Billboard.json	Metadata of the featured show
Grid.json	Array of show IDs for the grid
Videos.json	Metadata for all available shows

5. Component Architecture





6. State Management

- React useState for active tab, my shows and all shows.

7. UI/UX Design

Color Theme:

- Background color - Black,
- Text color - White
- Active tab border: 1px solid white
- Hover effect: `transform: scale(0.95)`, inset shadow

Billboard:

- Full-width background image

- Overlay title + logline in bottom-left

Show Grid:

- Center-aligned, responsive columns
- On hover: display Add/Remove button
- On click (excluding button): highlight with 2-sec white border

8. Accessibility

- Keyboard-navigable buttons
- Visible focus states
- ARIA labels for add/remove controls
- High contrast text/background

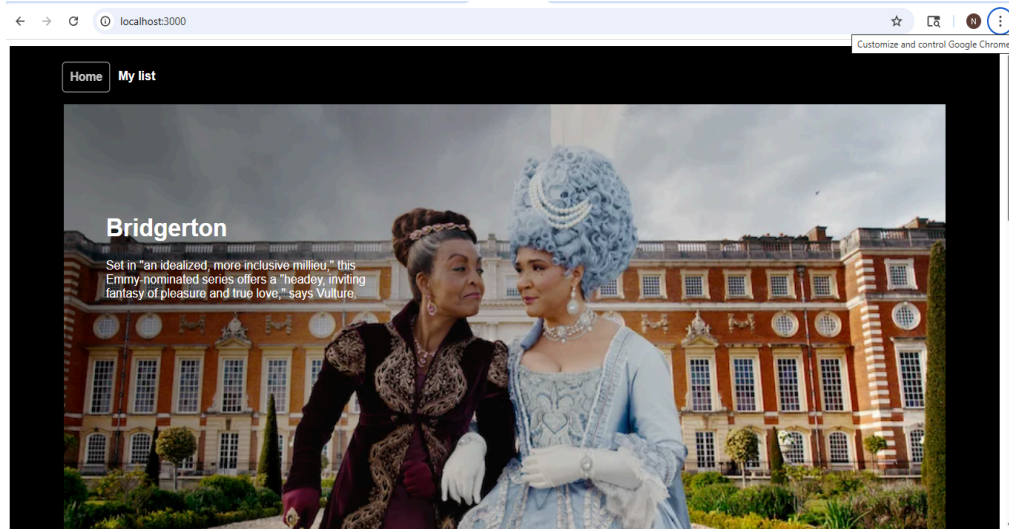
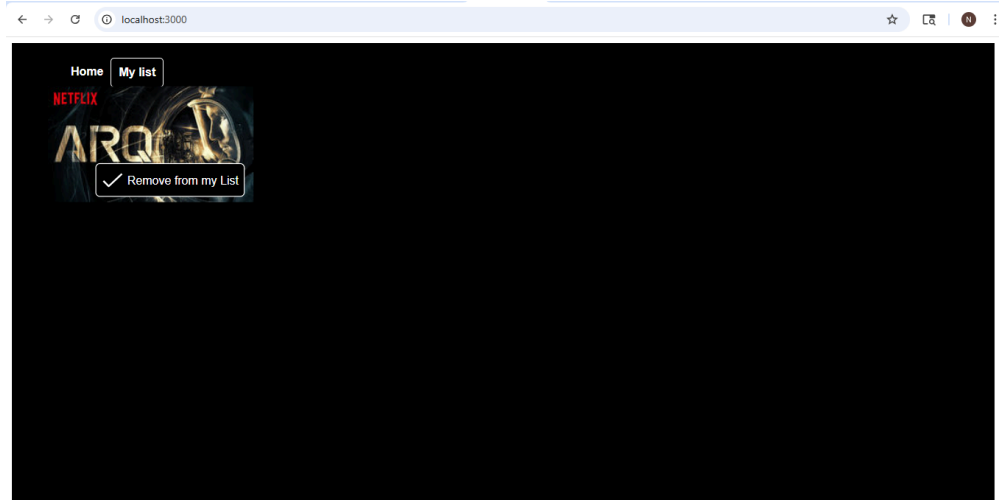
9. Performance

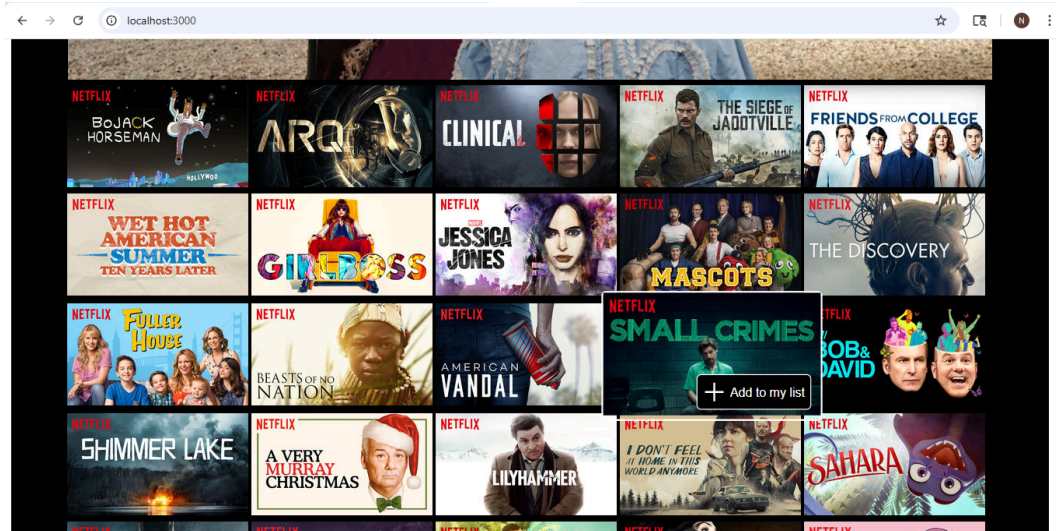
- Lazy-load images
- Smooth CSS transitions

10. Manual Test Log

ID	Feature	Test Scenario	Expected Scenario	Actual Scenario	Result
1	Navigation Tabs	Clicking on active tabs changes the active view	Corresponding view is rendered	As expected	Pass
2	Tab Styling	Active tab has a white border	Tab shows white border	As expected	Pass
3	Billboard	Billboard displays featured show with overlay text	Billboard with title + logline displayed	As expected	Pass
4	Show Grid	Grid shows all 35 items from <code>grid.json</code>	Grid shows expected number of items	As expected	Pass
5	Hover Effect	Hovering over show reveals Add button	Add/Remove button appears	As expected	Pass
6	Add to my list	Click Add → Item moves to My List	Item appears in My List	As expected	Pass
7	Remove Show	Click Remove → Item disappears from My List	Item is removed	As expected	Pass

11. Screenshots





12. Future Enhancements

- Billboard data could be made config driven.
- Infinite scroll or pagination functionality can be added to support a larger number of shows.
- Search/filter support can be added.
- Responsiveness can be refined to ensure consistent user experience across all devices.

Appendix: Sample JSON Files

billboard.json

```
{
  "id": "001",
  "title": "Featured Show",
  "logline": "A thrilling journey.",
  "metadata": {
    "storyart": "/assets/storyart.webp",
    "boxart": "/assets/boxart.jpg"
  }
}
```

```
}
```

grid.json

```
["001", "002", "003", "004"]
```

videos.json

```
[  
  { "id": "001", "title": "Show 1", "image": "/assets/show1.jpg" },  
  { "id": "002", "title": "Show 2", "image": "/assets/show2.jpg" }  
]
```