Netflix House Implementation Summary

Overview

This implementation provides a complete solution for the Netflix House interactive video experience. The system allows users to explore virtual Netflix House locations through interactive aerial maps with hotspots that trigger seamless video sequences.

Key Components

Backend (Node.js/Express)

1. **MongoDB Models**:
    - Location - Stores location information (King of Prussia, Dallas)
    - Asset - Manages video and image assets with appropriate categorization
    - Hotspot - Stores interactive points with polygon coordinates and types (PRIMARY/SECONDARY)
    - Playlist - Associates video sequences with PRIMARY hotspots
2. **API Endpoints**:
    - Asset management (upload, list, delete)
    - Hotspot creation and management
    - Playlist configuration
    - Location management
3. **AWS S3 Integration**:
    - Handles asset storage and retrieval
    - Manages file uploads
    - Serves video and image content

Frontend (React)

1. **User Experience**:
    - Menu page for location selection
    - Interactive aerial map view with hotspots
    - Seamless video sequence playback
    - Information panels for SECONDARY hotspots
    - Location navigation buttons
2. **Admin Panel**:
    - Assets tab for media management
    - Hotspots tab with visual polygon drawing tool
    - Playlists tab for video sequence assignment
    - Consistent save and publish mechanism
3. **Core Components**:
    - Custom video player for seamless transitions
    - Interactive canvas for hotspot creation
    - Preloading system for smooth video playback
    - Context providers for state management

Implementation Features

1. **Video Management**:
    - Preloads videos for instant playback
    - Handles seamless transitions between videos
    - Supports looping for aerial maps
    - Manages video asset categories
2. **Hotspot System**:
    - Support for polygon-shaped hotspots

- Different behaviors for PRIMARY and SECONDARY types
- Visual map pins over hotspot areas
- Interactive hover states

3. **Admin Interface**:
   - Visual tools for content configuration
   - Drag-and-drop asset management
   - Canvas-based hotspot creation
   - Video preview and selection
4. **Technical Highlights**:
   - Responsive design for iPad Pro
   - Clean context-based state management
   - Efficient video preloading system
   - Seamless asset management

Deployment

The application is designed to be deployed using GitHub and Railway.app with environment variables for configuration. All necessary setup is documented in the README.md file.

Future Enhancements

1. **Performance Optimization**:
   - Further optimize video preloading
   - Implement compression for faster loading
   - Add caching mechanisms
2. **Feature Extensions**:
   - Support for more interactive elements
   - Additional hotspot types with different behaviors
   - Multi-language support
   - Analytics for tracking user interactions
3. **Admin Improvements**:
   - Bulk upload capabilities
   - Template system for hotspots
   - Enhanced preview capabilities
   - User access controls

This implementation provides a solid foundation that meets all the specified requirements while maintaining a clean, maintainable code structure that can be extended as needed.