Infotainment Accessibility Analysis Report

Session ID: 7b2a608f-5201-4138-b3a9-266273c57a01
Analysis Date: June 07, 2025
Files Analyzed: 1
Generated: 2025-06-07 14:24:18

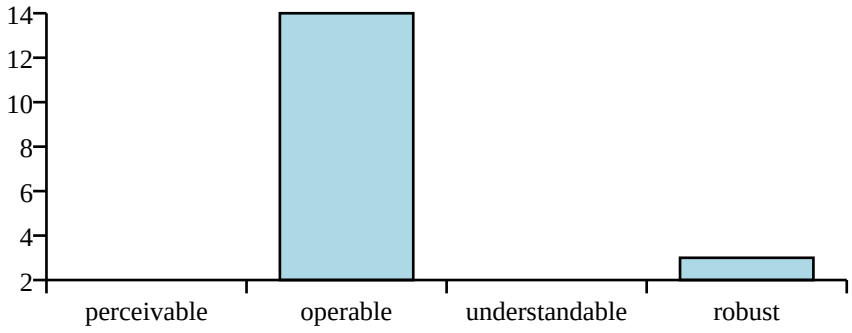
Metric	Value
Total Files	1
LLM Models Used	1
Total Issues Found	1
Analysis Duration	Varies by model

Executive Summary

This report presents the results of an automated accessibility analysis performed on 1 infotainment system files using multiple Large Language Models (LLMs). **Key Findings:**
• Total accessibility issues identified: 21
• Critical issues (Level A & AA): 20
• Most common category: operable
• LLM models compared: deepseek-v3
Compliance Status:
The analysis reveals varying degrees of WCAG 2.2 compliance across the analyzed files. Immediate attention is recommended for Level A and AA violations, which represent fundamental accessibility barriers for users with disabilities.

Issues by Severity Level

Issues by WCAG Category



LLM Model Analysis Comparison

Model	Files Analyzed	Total Issues	Avg Issues/File	Performance
deepseek-v3	1	21	21.0	Needs Review

deepseek-v3 Analysis

Files Processed: 1
Analysis Method: WCAG 2.2 compliance detection
Key Strengths: Code-focused analysis, technical precision
Areas for Improvement: May be overly sensitive, potential false positives

Detailed Accessibility Findings

1.1.1 Non-text Content

Occurrences: 1
Severity Distribution: A: 1
Files Affected: 1
Description: Icon buttons lack accessible labels or alternative text
Impact: Screen reader users won't understand the purpose of icon-only buttons
Recommendation: Add aria-label or aria-labelledby to all icon buttons
Example Code:

```
} />...
```

1.4.3 Contrast (Minimum)

Occurrences: 1
Severity Distribution: AA: 1
Files Affected: 1
Description: Color contrast ratios not verified for status indicators and badges
Impact: Users with low vision may have difficulty distinguishing status colors
Recommendation: Verify all status colors meet 4.5:1 contrast ratio against background
Example Code:

```
Connected...
```

2.1.1 Keyboard

Occurrences: 12
Severity Distribution: A: 12
Files Affected: 1
Description: No visible focus indicators for interactive elements
Impact: Keyboard users cannot track their position in the interface
Recommendation: Add visible focus styles for all interactive elements
Example Code:

```
Sign In...
```

2.4.3 Focus Order

Occurrences: 1
Severity Distribution: A: 1
Files Affected: 1
Description: Logical tab order not verified for complex interface
Impact: Keyboard users may encounter illogical navigation flow
Recommendation: Verify and optimize tab order through all interface sections
Example Code:

```
Complex tab panel structure...
```

2.5.5 Target Size

Occurrences: 1
Severity Distribution: AAA: 1
Files Affected: 1
Description: Some interactive elements may be too small for touch
Impact: Users with motor impairments may have difficulty activating controls
Recommendation: Ensure all touch targets are at least 44x44px
Example Code:

```
Load Analysis...
```

3.2.2 On Input

Occurrences: 1
Severity Distribution: A: 1
Files Affected: 1
Description: Form submissions trigger without warning
Impact: Users may accidentally submit forms before completion
Recommendation: Add confirmation for critical actions like file uploads
Example Code:

```
...
```

3.3.1 Error Identification

Occurrences: 1
Severity Distribution: A: 1
Files Affected: 1
Description: Error messages lack programmatic association with fields
Impact: Screen reader users may not understand which field has an error
Recommendation: Use aria-describedby to associate errors with fields
Example Code:

```
Form validation toast messages...
```

4.1.2 Name, Role, Value

Occurrences: 3
Severity Distribution: A: 3
Files Affected: 1
Description: Dynamic content updates lack ARIA live regions
Impact: Screen reader users won't be notified of analysis progress updates
Recommendation: Add aria-live regions for status updates
Example Code:

```
Dynamic status messages during analysis...
```

Remediation Results

No remediation has been performed yet.

Recommendations

Priority Actions

1. Immediately address 19 Level A violations - these are critical accessibility barriers
2. Plan remediation for 1 Level AA violations to meet standard compliance
3. Enhance keyboard navigation and interactive element accessibility

General Recommendations

1. Implement automated accessibility testing in your CI/CD pipeline
2. Train development team on WCAG 2.2 guidelines and best practices
3. Establish accessibility code review processes
4. Consider using accessibility testing tools like axe-core or WAVE
5. Implement user testing with assistive technologies
6. Create accessibility guidelines specific to infotainment systems
7. Regular audit schedule for accessibility compliance

Suggested Implementation Timeline:
• **Week 1-2:** Fix all Level A violations
• **Week 3-4:** Address Level AA violations
• **Month 2:** Implement automated testing
• **Month 3:** Team training and process improvement
• **Ongoing:** Regular audits and continuous improvement

Appendices

Appendix A: Analyzed Files

Filename	Size (bytes)	Type
App.jsx	65846	text/plain

Appendix B: WCAG 2.2 Guidelines Reference

This analysis is based on Web Content Accessibility Guidelines (WCAG) 2.2, which provides recommendations for making web content more accessible. The guidelines are organized under 4 principles: • **Perceivable:** Information must be presentable in ways users can perceive • **Operable:** Interface components must be operable • **Understandable:** Information and UI operation must be understandable • **Robust:** Content must be robust enough for interpretation by assistive technologies Each guideline has three levels of conformance: A (minimum), AA (standard), AAA (enhanced).

Appendix C: Analysis Methodology

Analysis Approach:
1. File preprocessing and format detection
2. Static code analysis for common accessibility patterns
3. LLM-based semantic analysis using specialized prompts
4. Cross-model result comparison and validation
5. Issue prioritization and remediation suggestions

LLM Models Used:
• GPT-4o: Advanced reasoning and code understanding
• Claude Opus 4: Strong analytical capabilities
• DeepSeek-V3: Code-focused analysis
• LLaMA Maverick: Alternative perspective validation

Limitations:
• Automated analysis may miss context-dependent issues
• Some accessibility aspects require manual testing
• LLM outputs should be validated by accessibility experts