Frontend Test Coverage Summary

Coverage report generated based on Vite/Vitest results.

# Overall Coverage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Statements | Branches | Functions | Lines | Not Covered Lines |
| 96.8% | 88.75% | 86.48% | 96.8% | — |

# Module Breakdown

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module/File | % Statements | % Branches | % Functions | % Lines | Not Covered Lines |
| All files | 96.8 | 88.75 | 86.48 | 96.8 | — |
| src/ | 92.19 | 84.61 | 83.33 | 92.19 | 36,96-99,170-172,175-177 |
| App.jsx | 92.19 | 84.61 | 83.33 | 92.19 | 36,96-99,170-172,175-177 |
| src/components/ | 98.14 | 92.1 | 85.71 | 98.14 | — |
| AudioUploadForm.jsx | 100 | 100 | 50 | 100 | — |
| AudioUploader.jsx | 93.98 | 79.31 | 100 | 93.98 | 76-79,94-97 |
| CalendarEventForm.jsx | 98.52 | 91.66 | 100 | 98.52 | 78-79 |
| DecisionsPanel.jsx | 100 | 100 | 100 | 100 | — |
| ErrorBoundary.jsx | 100 | 85.71 | 100 | 100 | 59 |
| NextcloudConnect.jsx | 100 | 100 | 100 | 100 | — |
| ReviewPanel.jsx | 100 | 97.5 | 62.5 | 100 | 92 |
| SummaryPanel.jsx | 100 | 100 | 100 | 100 | — |
| TranscriptPanel.jsx | 100 | 100 | 100 | 100 | — |
| src/hooks/ | 96.47 | 69.23 | 100 | 96.47 | — |
| UseMeetingState.jsx | 96.47 | 69.23 | 100 | 96.47 | 77-79 |
| src/utils/ | 96.07 | 85.71 | 100 | 96.07 | — |
| dateUtils.js | 96.07 | 85.71 | 100 | 96.07 | 41-42 |

# Highlights

• Most files/components are at or near 100% coverage.  
• App.jsx and AudioUploader.jsx have some uncovered lines.  
• Branches (conditional code) and functions have slightly lower coverage, mainly in App.jsx, AudioUploader.jsx, and UseMeetingState.jsx.

# Next Steps to 100%

Add tests to cover:  
- App.jsx: Lines 36, 96-99, 170-172, 175-177  
- AudioUploader.jsx: Lines 76-79, 94-97  
- CalendarEventForm.jsx: Lines 78-79  
- ErrorBoundary.jsx: Line 59  
- ReviewPanel.jsx: Line 92  
- dateUtils.js: Lines 41-42  
- UseMeetingState.jsx: Lines 77-79 (conditional/branch logic)  
  
Tip: Review uncovered lines in each file and write targeted tests for those branches, error paths, or components.

If you need more detail, reach out to your engineering lead for test-writing guidance.