STEP 28: Full In-Browser File Viewer & Attachment Preview System - Testing Guide

Implementation Date: November 30, 2025
Status: ✅ Complete

Overview

This document provides comprehensive testing scenarios for the Full In-Browser File Viewer & Attachment Preview System. The system supports previewing PDF, DOCX, XLSX, CSV, PPTX, images, videos, TXT, and MD files directly in the browser.

Test Scenarios

Test 1: PDF File Preview

Prerequisites:
- Upload a PDF file as an attachment to a supplier response
- Navigate to either supplier response form or buyer response detail view

Steps:
1. Locate the PDF attachment in the attachments list
2. Click the “Preview” button
3. Observe the preview modal opens
4. Verify PDF renders with page navigation controls
5. Test “Previous” and “Next” buttons to navigate pages
6. Click “Download Original” button
7. Close the modal using the X button

Expected Results:
- ✅ Preview modal opens with correct filename and file size
- ✅ PDF displays all pages correctly
- ✅ Page navigation works (Previous/Next buttons)
- ✅ Download button triggers file download
- ✅ Modal closes properly
- ✅ Activity log records ATTACHMENT_PREVIEWED event

Test 2: Word (DOCX) File Preview

Prerequisites:
- Upload a DOCX file as an attachment

Steps:
1. Click “Preview” on the DOCX attachment

2. Wait for text extraction to complete
3. Verify HTML content renders with formatting
4. Scroll through the document
5. Test download functionality
6. Close the modal

Expected Results:
- ✅ Loading spinner displays during text extraction
- ✅ Word document content displays with basic formatting
- ✅ Headers, paragraphs, and lists render correctly
- ✅ Download original file works
- ✅ Activity logged

Test 3: Excel (XLSX) File Preview

Prerequisites:
- Upload an XLSX file with multiple sheets

Steps:
1. Click “Preview” on the XLSX attachment
2. Observe the spreadsheet viewer loads
3. If multiple sheets, test the sheet selector dropdown
4. Verify table displays headers and rows correctly
5. Test horizontal scrolling for wide tables
6. Verify display limits to first 100 rows
7. Download the file

Expected Results:
- ✅ Spreadsheet loads with proper table structure
- ✅ Sheet selector shows all available sheets
- ✅ Switching sheets updates the display
- ✅ Headers are bold and distinct
- ✅ Data cells display correctly
- ✅ Large spreadsheets don’t cause performance issues
- ✅ Activity logged

Test 4: CSV File Preview

Prerequisites:
- Upload a CSV file

Steps:
1. Click “Preview” on CSV attachment
2. Verify table renders with comma-separated values
3. Check header row is formatted differently
4. Test scrolling for large CSV files
5. Download the original

Expected Results:
- ✅ CSV parses correctly as table

- ✅ No encoding issues with special characters
- ✅ Header row is visually distinct
- ✅ Performance is acceptable for large files
- ✅ Activity logged

Test 5: PowerPoint (PPTX) File Preview

Prerequisites:
- Upload a PPTX file

Steps:
1. Click “Preview” on PPTX attachment
2. View the slide-by-slide breakdown
3. Verify each slide shows title and content
4. Test scrolling through all slides
5. Download the file

Expected Results:
- ✅ Slides display in order
- ✅ Slide numbers shown
- ✅ Titles and bullet points extracted
- ✅ Note about full extraction limitations displayed
- ✅ Activity logged

Test 6: Image File Preview

Prerequisites:
- Upload PNG, JPG, or WEBP images

Steps:
1. Click “Preview” on image attachment
2. Verify image displays full-size
3. Test with different image sizes and aspect ratios
4. Verify image doesn’t overflow modal
5. Download the image

Expected Results:
- ✅ Image displays correctly
- ✅ Image scaling respects max-width/height
- ✅ No distortion or cropping
- ✅ Download works
- ✅ Activity logged

Test 7: Video File Preview

Prerequisites:
- Upload MP4 or WEBM video

Steps:
1. Click “Preview” on video attachment
2. Verify video player loads
3. Test play/pause controls
4. Test video timeline scrubbing
5. Test volume controls
6. Verify Range header support for streaming
7. Download the video

Expected Results:
- ✅ Video player renders
- ✅ Play/pause works
- ✅ Timeline scrubbing functional
- ✅ Volume controls work
- ✅ Video streams smoothly (206 Partial Content)
- ✅ Download works
- ✅ Activity logged

Test 8: Text (TXT) File Preview

Prerequisites:
- Upload a .txt file

Steps:
1. Click “Preview” on TXT attachment
2. Verify plain text displays with proper line breaks
3. Test with various text encodings (UTF-8)
4. Verify monospace font rendering
5. Download the file

Expected Results:
- ✅ Text content displays correctly
- ✅ Line breaks preserved
- ✅ No encoding issues
- ✅ Monospace font applied
- ✅ Activity logged

Test 9: Markdown (MD) File Preview

Prerequisites:
- Upload a .md file with various markdown syntax

Steps:
1. Click “Preview” on MD attachment
2. Verify markdown renders as formatted HTML
3. Test headings, lists, bold, italic
4. Test code blocks and links
5. Download the original

Expected Results:
- ✅ Markdown parses correctly
- ✅ All formatting elements render
- ✅ Code blocks have syntax highlighting (if supported)
- ✅ Links are clickable (in safe mode)
- ✅ Activity logged

Test 10: Unsupported File Type

Prerequisites:
- Upload a file type not in the supported list (e.g., .exe, .zip)

Steps:
1. Click “Preview” on unsupported file
2. Observe the “Unsupported Viewer” message
3. Verify download button still works
4. Close the modal

Expected Results:
- ✅ Clear message about unsupported type
- ✅ Download button available
- ✅ No errors in console
- ✅ Activity logged with “unsupported” type

Test 11: Authorization - Buyer Access

Prerequisites:
- Login as a buyer
- Access an RFP you own with supplier responses

Steps:
1. Navigate to a supplier response detail page
2. Click “Preview” on an attachment
3. Verify preview loads successfully

Expected Results:
- ✅ Buyer can preview attachments for their RFPs
- ✅ No authorization errors
- ✅ Activity logged with BUYER actor role

Test 12: Authorization - Supplier Access

Prerequisites:
- Login as a supplier
- Access your own response

Steps:
1. Navigate to your supplier response form
2. Click “Preview” on your uploaded attachment

3. Verify preview loads successfully
4. Try to access another supplier’s attachment (via direct URL if possible)

Expected Results:
- ✅ Supplier can preview their own attachments
- ✅ Supplier cannot access other suppliers’ attachments (403 error)
- ✅ Activity logged with SUPPLIER actor role

Test 13: Authorization - Cross-RFP Access

Prerequisites:
- Multiple RFPs with different owners

Steps:
1. Login as Buyer A
2. Try to access attachment from Buyer B’s RFP (via direct URL)
3. Verify 403 Forbidden error

Expected Results:
- ✅ Cross-RFP access blocked
- ✅ 403 error returned
- ✅ No data leakage

Test 14: File Not Found in Storage

Prerequisites:
- Attachment record exists but file deleted from storage

Steps:
1. Delete the actual file from uploads/attachments/ directory
2. Try to preview the attachment
3. Observe error handling

Expected Results:
- ✅ User-friendly error message
- ✅ No server crash
- ✅ 404 error in /text endpoint

Test 15: Large File Performance

Prerequisites:
- Upload a large file (>10MB PDF, >5000 row Excel)

Steps:
1. Click “Preview” on large file
2. Measure load time
3. Test scrolling and navigation performance
4. Verify memory usage doesn’t spike excessively

Expected Results:
- ✅ Loading spinner displays
- ✅ Preview loads within reasonable time (<5 seconds)
- ✅ Navigation remains responsive
- ✅ No browser crashes or memory leaks
- ✅ Excel limits to 100 rows for performance

Test 16: Activity Log Verification

Prerequisites:
- Access to activity log view

Steps:
1. Preview 3 different attachments
2. Navigate to the RFP's activity log page
3. Search/filter for ATTACHMENT_PREVIEWED events
4. Verify details are logged correctly

Expected Results:
- ✅ 3 preview events logged
- ✅ Each event has correct filename, actor, timestamp
- ✅ Details JSON includes attachmentId, fileType, fileSize
- ✅ Events visible in both buyer and supplier activity logs

Test 17: Mobile Responsiveness

Prerequisites:
- Access from mobile device or responsive mode

Steps:
1. Open preview modal on mobile
2. Test PDF page navigation on small screen
3. Test table scrolling in Excel/CSV viewers
4. Verify modal fits screen properly
5. Test close button is accessible

Expected Results:
- ✅ Modal adapts to mobile viewport
- ✅ Controls are touch-friendly
- ✅ Content scrolls properly
- ✅ No horizontal overflow issues
- ✅ Close button easy to tap

Test 18: Browser Compatibility

Prerequisites:
- Access to Chrome, Firefox, Safari

Steps:
1. Test preview functionality in each browser
2. Verify PDF.js worker loads correctly
3. Test video playback compatibility
4. Check for console errors

Expected Results:
- ✅ Works in all major browsers
- ✅ PDF.js worker loads from CDN
- ✅ Video codecs supported
- ✅ No browser-specific errors

Test 19: Network Error Handling

Prerequisites:
- Ability to simulate network issues

Steps:
1. Open preview modal
2. Disconnect network during loading
3. Observe error handling
4. Reconnect and retry

Expected Results:
- ✅ Error message displays
- ✅ No infinite loading state
- ✅ Retry mechanism available
- ✅ Graceful degradation

Test 20: Concurrent Previews

Prerequisites:
- Multiple attachments

Steps:
1. Open preview for Attachment A
2. Close and immediately open Attachment B
3. Verify state resets properly
4. Test opening preview while another is loading

Expected Results:
- ✅ Previous preview state clears
- ✅ New preview loads correctly
- ✅ No state leakage between previews
- ✅ Loading cancels when modal closes

API Verification

Test `/api/attachments/[id]/meta`

```
curl -X GET http://localhost:3000/api/attachments/[id]/meta \
  -H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Expected Response:

```json
{
  "id": "attachment-id",
  "fileName": "document.pdf",
  "fileType": "application/pdf",
  "attachmentType": "GENERAL",
  "fileSize": 524288,
  "createdAt": "2025-11-30T...",
  "canPreview": true,
  "safePreviewType": "pdf"
}
```

Test `/api/attachments/[id]/text`

```
curl -X GET http://localhost:3000/api/attachments/[id]/text \
  -H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Expected Response (DOCX):

```json
{
  "html": "<p>Document content...</p>"
}
```

Expected Response (XLSX):

```json
{
  "sheets": [
    {
      "name": "Sheet1",
      "rows": [[...]]
    }
  ]
}
```

Test `/api/attachments/[id]/download` with Range Header

```
curl -X GET http://localhost:3000/api/attachments/[id]/download \
  -H "Range: bytes=0-1023" \
  -H "Cookie: next-auth.session-token=YOUR_SESSION_TOKEN"
```

Expected Response:
- Status: 206 Partial Content
- Headers: Content-Range, Accept-Ranges, Content-Type
- Body: First 1024 bytes of file

Database Verification

Check Activity Logs

```
SELECT * FROM "ActivityLog"
WHERE "eventType" = 'ATTACHMENT_PREVIEWED'
ORDER BY "createdAt" DESC
LIMIT 10;
```

Expected Results:
- Rows with ATTACHMENT_PREVIEWED eventType
- Correct rfpId, userId, supplierContactId
- Details JSON with fileName, fileType, fileSize

Validation Checklist

Backend APIs

- [x] `/api/attachments/[id]/meta` created
- [x] `/api/attachments/[id]/text` created
- [x] `/api/attachments/[id]/download` enhanced with Range support
- [x] Authorization checks implemented
- [x] Activity logging integrated

Frontend Components

- [x] FilePreviewModal created with all viewers
- [x] PDFViewer implemented
- [x] WordViewer implemented
- [x] SpreadsheetViewer implemented
- [x] PowerPointViewer implemented
- [x] ImageViewer implemented
- [x] VideoViewer implemented
- [x] TextViewer implemented
- [x] MarkdownViewer implemented
- [x] UnsupportedViewer implemented

UI Integration

- [x] Supplier response form - Preview button added
- [x] Buyer response detail - Preview button added
- [x] FilePreviewModal integrated in both locations

Activity Logging

- [x] ATTACHMENT_PREVIEWED event type added
- [x] EVENT_TYPE_LABELS updated
- [x] Activity logged in /meta endpoint

Dependencies

- [x] pdfjs-dist installed
- [x] react-pdf installed
- [x] mammoth installed
- [x] react-markdown installed

Common Issues & Fixes

Issue: PDF.js Worker Not Loading

Solution:
- Verify CDN URL is accessible
- Check browser console for CORS errors
- Ensure pdfjs.GlobalWorkerOptions.workerSrc is set correctly

---

Issue: DOCX Shows Blank

Solution:
- Check file was uploaded correctly
- Verify mammoth library is installed
- Check /text endpoint returns valid HTML

Issue: Video Won’t Play

Solution:
- Verify video codec is browser-supported (H.264 for MP4)
- Check Range header support in download endpoint
- Test with smaller video file

Issue: 403 Forbidden on Preview

Solution:
- Verify user session is valid
- Check RFP ownership
- Confirm attachment belongs to accessible supplier response

Performance Metrics

Target Benchmarks:
- PDF preview: <2 seconds for first page
- DOCX extraction: <3 seconds for typical document
- XLSX parsing: <2 seconds for 100 rows
- Image load: <1 second
- Video start: <1 second (streaming)

Deployment Checklist

Pre-Deployment

- [ ] All dependencies installed in production
- [ ] PDF.js CDN accessible from production
- [ ] Upload storage path configured correctly
- [ ] Database migration applied (no schema changes)

Post-Deployment

- [ ] Test preview with production data
- [ ] Verify activity logs recording
- [ ] Check performance with real file sizes
- [ ] Monitor for errors in logs

Conclusion

The Full In-Browser File Viewer & Attachment Preview System provides comprehensive file preview capabilities across 9+ file types with robust security, activity logging, and user-friendly UI. All test scenarios should pass for production readiness.

Testing Complete: November 30, 2025
Next Steps: Build verification and Git commit (STEP 7)