Bid Proposals: PDF Parsing and Credits Error Handling Fix

Summary

Fixed critical issues in the bid proposal extraction system:
1. **PDF Parsing Error**: Installed missing `canvas` dependencies required by pdf-parse
2. **Credits Error Handling**: Implemented graceful error handling when LLM API credits are exhausted

Issues Fixed

1. PDF Parsing Error

Problem:

```
Warning: Cannot load "@napi-rs/canvas" package
Error extracting text from UU207666056.pdf: ReferenceError: DOMMatrix is not defined
```

Root Cause:
- The `pdf-parse` library requires `canvas` and `@napi-rs/canvas` for certain PDF operations
- These dependencies were not installed, causing DOMMatrix errors

Solution:
- Installed `canvas` and `@napi-rs/canvas` packages
- These provide the necessary polyfills for PDF parsing in the Node.js environment

Files Modified:
- `package.json` (via yarn add command)

2. Credits Error Handling

Problem:

```
Error: AI API error: {"success": false, "error":
"You have no remaining credits to use the LLM apis."}
```

Root Cause:
- When LLM credits were exhausted, the entire bid extraction endpoint would fail
- Users couldn't create bid proposals at all, even manually
- No clear error message was shown to users

Solution:
Implemented comprehensive error handling in the extraction API:

1. **Graceful Fallback**: Try AI extraction, but continue if it fails
2. **Default Values**: Set reasonable defaults when AI extraction fails
3. **User-Friendly Messages**: Clear error messages about credits
4. **Manual Entry**: Allow users to manually fill in details if AI fails

5. **Skip AI Generation**: Don't attempt proposal generation if credits are exhausted

Technical Implementation

API Route: `/api/bid-proposals/extract`

Before:

- AI extraction was required for bid proposal creation
- Any AI failure would crash the entire endpoint
- No fallback mechanism

After:

```
// Try AI extraction with error handling
let extractedInfo: any = null;
let aiExtractionError: string | null = null;

try {
  extractedInfo = await extractBidInformationFromDocuments(documentContents);
} catch (error) {
  // Detect credits error
  const errorMessage = error instanceof Error ? error.message : String(error);
  if (errorMessage.includes('no remaining credits') ||
errorMessage.includes('credits')) {
    aiExtractionError = 'AI extraction failed: You have run out of LLM API cred-
its...';
  } else {
    aiExtractionError = `AI extraction failed: ${errorMessage}...`;
  }

  // Set default values for manual entry
  extractedInfo = {
    solicitationNumber: 'N/A',
    title: 'Untitled Bid - Please Update',
    // ... other defaults
  };
}

// Create bid with status based on AI success
envelope1Status: aiExtractionError ? 'draft' : 'in_progress',
envelope2Status: aiExtractionError ? 'draft' : 'in_progress',

// Skip AI generation if extraction failed
if (!aiExtractionError) {
  // Generate proposals in background
} else {
  console.log('Skipping AI proposal generation due to earlier extraction failure');
}

// Return with warning if applicable
return NextResponse.json({
  success: true,
  id: bidProposal.id,
  message: aiExtractionError
    ? `Bid proposal created, but AI extraction failed. ${aiExtractionError}`
    : 'Bid proposal created. Proposals are being generated...',
  warning: aiExtractionError || undefined,
});
```

User Experience Improvements

When Credits Are Available:

1. Upload RFP documents
2. AI extracts bid information automatically
3. Proposals are generated in the background
4. User can review and edit as needed

When Credits Are Exhausted:

1. Upload RFP documents
2. Bid proposal is created with default values
3. Clear warning message about credits
4. User can manually fill in all details
5. User can manually write proposals
6. Documents are still securely stored in S3

Error Messages:

- **Credits Error**: “AI extraction failed: You have run out of LLM API credits. Please add more credits to enable AI-powered bid extraction. You can still manually fill in the bid details.”
- **Other Errors**: “AI extraction failed: [error message]. You can still manually fill in the bid details.”

Dependencies Added

```
{
  "canvas": "^3.2.0",
  "@napi-rs/canvas": "^0.1.x"
}
```

These packages provide:
- Canvas API for PDF rendering
- DOMMatrix polyfill for PDF operations
- Image processing capabilities for document extraction

Testing Results

TypeScript Compilation

✅ No type errors

Next.js Build

✅ Build successful
✅ All routes compiled
✅ No build errors

PDF Parsing

✅ PDF documents can now be parsed without DOMMatrix errors
✅ Document extraction service works correctly

Credits Error Handling

✅ Bid proposals can be created even when credits are exhausted
✅ Clear error messages displayed to users
✅ Users can proceed with manual entry
✅ Background generation is skipped when appropriate

Files Modified

1. `/app/api/bid-proposals/extract/route.ts`
   - Added try-catch for AI extraction
   - Implemented credits error detection
   - Added default values fallback
   - Modified envelope status logic
   - Added conditional AI generation
   - Updated response messages

2. `package.json` (via yarn add)
   - Added `canvas` package
   - Added `@napi-rs/canvas` package

Database Schema

No changes required - uses existing BidProposal schema

Migration Required

None - backward compatible with existing data

Future Enhancements

1. **Credits Dashboard**: Show remaining credits to users
2. **Credit Top-Up**: In-app credit purchase flow
3. **Usage Analytics**: Track credit consumption per feature
4. **Alternative Extraction**: Use text-only extraction when credits low
5. **Batch Processing**: Queue proposals for when credits replenish

Notes

- Pre-existing issues remain (broken external links, duplicate blog images)
- These don't affect bid proposal functionality
- PDF parsing now works in production builds
- Error handling is comprehensive and user-friendly

Deployment Status

✅ Ready for deployment
- All tests passing
- Build successful

- Error handling verified
- User experience improved

Document created: November 9, 2025
System: CDM Suite Website - Bid Proposals Module