

Progress Bar and Transaction Display Fixes

Issues Fixed

1. Progress Bar Stuck at 25%

Problem:

- The progress bar in the upload queue was stuck at 25% and didn't accurately reflect the processing status
- The frontend wasn't properly mapping all processing stages to progress percentages

Solution:

- **Updated progress mapping** in `components/bank-statements/bank-statement-uploader.tsx` :
- Added mapping for all processing stages including "UPLOADED" (10%), "PROCESSING" (20%)
- Properly handles "FAILED" status by setting progress to 0
- Maps stages: UPLOADED → 10%, PROCESSING → 20%, EXTRACTING_DATA → 40%, CATEGORIZING_TRANSACTIONS → 60%, ANALYZING_PATTERNS → 80%, DISTRIBUTING_DATA → 90%, COMPLETED → 100%
- **Added real-time database updates** in `app/api/bank-statements/process/route.ts` :
 - Updates `processingStage` to 'EXTRACTING_DATA' when extraction begins
 - Updates `recordCount` after each extraction layer completes
 - Updates `recordCount` with final deduplicated count
 - Ensures the frontend can poll and display accurate progress

Benefits:

- Users can now see exactly what stage the processing is at
 - Progress bar accurately reflects the actual processing status
 - More responsive UI that updates as processing progresses
-

2. Transaction Count Display

Problem:

- Users couldn't see how many transactions were being extracted during processing
- The upload queue didn't show transaction counts

Solution:

- **Added transaction count display** in `components/bank-statements/bank-statement-uploader.tsx` :
- Added `transactionCount` field to `UploadFile` interface
- Extracts `recordCount` and `processedCount` from status updates
- Displays transaction count in real-time next to the progress bar (e.g., "22 transactions")
- Shows count as it updates during extraction

Benefits:

- Users can see in real-time how many transactions are being extracted
 - Provides transparency into the extraction process
 - Helps identify if extraction is incomplete (e.g., only 22 instead of expected 118)
-

3. Enhanced Processing Status Updates

Problem:

- Frontend wasn't receiving timely updates during processing
- Polling interval was too fast (2 seconds) causing unnecessary server load

Solution:

- Improved polling mechanism:

- Increased polling interval from 2 to 3 seconds for better performance
- Added automatic refresh callback when processing completes
- Ensures upload history refreshes when all files are done processing

Benefits:

- Better performance with reduced server load
 - Automatic UI refresh when processing completes
 - Smoother user experience
-

4. Transaction Count Validation

Problem:

- No validation to detect when extraction fails to capture all transactions
- User reported getting 22 transactions instead of expected 118

Solution:

- **Added validation logic** in `app/api/bank-statements/process/route.ts` :
- Calculates expected minimum transactions based on file size (rough estimate: 1 transaction per 3KB)
- Logs warning if transaction count is suspiciously low
- Provides detailed extraction summary showing:
 - Total transactions before deduplication
 - Transactions after deduplication
 - Which extraction methods succeeded
 - File size and expected transaction count

Benefits:

- Helps identify incomplete extractions
 - Provides debugging information in server logs
 - Alerts developers to potential extraction issues
-

Files Modified

1. `app/components/bank-statements/bank-statement-uploader.tsx` :
 - Updated progress mapping for all stages
 - Added transaction count display
 - Improved polling mechanism
 - Added `transactionCount` to UploadFile interface
2. `app/app/api/bank-statements/process/route.ts` :
 - Added database updates for `processingStage` and `recordCount`
 - Implemented transaction count validation
 - Enhanced logging for debugging

Testing Instructions

Test Progress Bar

1. Log in to the app: `khouston@thebasketballfactorynj.com / hunterrrr777`
2. Navigate to Bank Statements page
3. Upload a PDF statement (e.g., Jan 2024.pdf)
4. Observe the upload queue:
 - Progress bar should show 10% → 20% → 40% → 60% → 80% → 90% → 100%
 - Each stage should be labeled (UPLOADING → PROCESSING → EXTRACTING_DATA → etc.)
 - Progress should smoothly update every 3 seconds

Test Transaction Count Display

1. During processing, watch the upload queue
2. You should see text like “22 transactions” appear next to the progress percentage
3. The count should update as extraction progresses
4. Final count should match the number of transactions in the statement

Verify Transaction Count

1. Check server logs for validation warnings if count is low
2. Look for messages like:

```
[Process Route] ! WARNING: Low transaction count (22) for file size (217.4KB)
[Process Route] ! Expected at least 72 transactions
```

Technical Details

Progress Calculation

The progress is calculated based on the `processingStage` field in the database:

- Default: 10%
- UPLOADED: 10%
- PROCESSING: 20%
- EXTRACTING_DATA: 40%
- CATEGORIZING_TRANSACTIONS: 60%
- ANALYZING_PATTERNS: 80%
- DISTRIBUTING_DATA: 90%
- COMPLETED: 100%
- FAILED: 0%

Polling Mechanism

The frontend polls the `/api/bank-statements/status?id={statementId}` endpoint every 3 seconds to get the latest status. The response includes:

- `status` : PENDING, PROCESSING, COMPLETED, FAILED
- `processingStage` : Current stage of processing
- `recordCount` : Number of transactions extracted
- `processedCount` : Number of transactions saved to database

Transaction Count Estimation

For validation, we use a rough estimate:

- Expected minimum = File size (KB) / 3
 - This assumes approximately 1 transaction per 3KB of PDF content
 - Warnings are only shown for files larger than 50KB to avoid false positives
-

Known Issues and Next Steps

Transaction Count Discrepancy

The user reported getting 22 transactions instead of 118 from the “Jan 2024.pdf” file. Database investigation shows:

1. **Latest upload** (User: cmgkjc8ggj0000ti1d76rcov7b): **22 transactions**
2. **Earlier upload** (User: cmhslln7n000331roqiivr13v): **116 transactions**

This suggests the extraction CAN work correctly (as evidenced by the 116-transaction upload), but something went wrong in the most recent upload. Possible causes:

1. **Different file uploaded:** The two different “Jan” files in /home/ubuntu/Uploads have different sizes:
 - Business Statement_Jan_8_2024.pdf: 52KB (likely 22 transactions)
 - Jan 2024.pdf: 217KB (likely 118 transactions)
2. **PDF Parser failure:** The PDF parser may have failed, causing fallback to OCR which only captured the first page
3. **Business Profile issue:** Both statements show `businessProfileId: null`, which might affect filtering

Recommendations

1. **Test with the correct file:** Ensure you’re uploading the 217KB “Jan 2024.pdf” file, not the 52KB “Business Statement_Jan_8_2024.pdf”
 2. **Check extraction logs:** When you upload, check the browser console and server logs for messages showing which extraction layers succeeded
 3. **Verify transaction sources:** After upload completes, the logs should show:

```
[Process Route] 📁 Transaction sources after deduplication:  
[Process Route]   pdf_parser: 118
```
 4. **Delete old statements:** Clear previous test uploads to avoid confusion about which statement is being displayed
-

Summary

All requested fixes have been implemented:

- ✓ **Progress bar** now correctly reflects all processing stages (no longer stuck at 25%)
- ✓ **Transaction count** is displayed in real-time during processing

- Validation** detects and warns about suspiciously low transaction counts
- Frontend refresh** automatically updates when processing completes
- Polling optimized** to 3-second interval for better performance

The app is ready for testing! Upload a PDF statement and watch the progress bar update smoothly from 10% to 100% with transaction counts displayed in real-time.