Transaction Loading and Saved Statements Implementation

Summary

Fixed the “load to database” error when loading manual transaction entries and implemented a permanent saved statements feature.

Issues Fixed

1. Transaction Loading Error

Problem: When clicking “Load to Database” for 118 transactions, the operation was failing.

Root Causes:
- Missing `statementPeriod` parameter in the API request
- Date parsing issues with MM/DD format
- Missing error handling and detailed error messages
- Database schema didn’t support MANUAL source type

Solution:
- Updated `/api/bank-statements/load-transactions` to:
- Accept `statementPeriod` parameter
- Create a `BankStatement` record for each manual entry
- Link all transactions to the statement via `bankStatementId`
- Improved date parsing to handle MM/DD and MM/DD/YY formats
- Added comprehensive error logging with transaction details
- Return the `statementId` in the response

- Updated Prisma schema:
- Added `MANUAL` to `StatementSourceType` enum
- Regenerated Prisma client

2. Missing Saved Statements Feature

Problem: After loading transactions, the card disappeared and there was no way to view or download them later.

Solution:
- Added saved statements state management:
- `savedStatements` state to store loaded statements
- `fetchSavedStatements()` function to retrieve statements from API
- Auto-load statements on component mount
- Refresh statements list after loading new transactions

- Created “Saved Statements” section at the bottom of the page:
- Shows all loaded statements with transaction count and date
- Displays statement period if provided

- Two action buttons per statement:
  - **View**: Opens transactions in a new tab filtered by statement ID
  - **Download**: Exports transactions as CSV file
- Added view and download handlers:

- `handleViewStatement()`: Fetches and displays transactions
- `handleDownloadStatement()`: Exports transactions to CSV format

Technical Changes

Files Modified:

1. `/app/app/api/bank-statements/load-transactions/route.ts`

- Added `statementPeriod` parameter to request body
- Created `BankStatement` record with:
- `sourceType: 'MANUAL'`
- `fileName`: “Manual Entry - [statement period]”
- `transactionCount`: Number of transactions
- `status: 'COMPLETED'`
- Linked all transactions to statement via `bankStatementId`
- Improved date parsing logic
- Enhanced error logging

2. `/app/components/bank-statements/bank-statements-client.tsx`

- Added saved statements state and fetch function
- Updated `handleLoadManualTransactions()` to:
- Extract statement period from card
- Pass `statementPeriod` to API
- Better error handling with detailed messages
- Refresh saved statements after loading
- Added `handleViewStatement()` and `handleDownloadStatement()` functions
- Created “Saved Statements” UI section at bottom
- Auto-load saved statements on mount

3. `/app/prisma/schema.prisma`

- Added `MANUAL` to `StatementSourceType` enum

Testing Instructions

1. **Login**:
   - Email: `khouston@thebasketballfactorynj.com`
   - Password: `hunterrr777`

2. **Load Manual Transactions**:
   - Go to Dashboard → Bank Statements
   - Enter statement date (e.g., “January 2024”)
   - Paste transaction text in the textarea
   - Click “Create Transaction Card”

- Verify the card shows correct transaction count and totals
- Click “Load to Database (118 transactions)”
- AI will classify transactions as Business/Personal
- Success message should show split (e.g., “85 Business + 33 Personal”)

3. **View Saved Statements**:
   - Scroll to bottom of page
   - “Saved Statements” section should appear
   - Each statement card shows:
     - Statement name/period
     - Transaction count
     - Upload date
     - Click “View” to see transactions in new tab
     - Click “Download” to export as CSV
4. **Verify Transaction Routing**:
   - Go to Dashboard (both profiles should show updated metrics)
   - Check Business profile: Should show business transactions
   - Check Personal/Household profile: Should show personal transactions

Features Implemented

✅ Transaction Loading

- Creates permanent BankStatement record
- Links all transactions to statement
- Proper date parsing (MM/DD, MM/DD/YY formats)
- AI classification (Business/Personal)
- Detailed error messages

✅ Saved Statements Section

- Shows all loaded statements
- Transaction count per statement
- Statement period display
- Upload date display
- View functionality (opens transactions page)
- Download functionality (CSV export)

✅ Data Persistence

- All manual entries saved to database
- Statements linked to transactions
- Can be viewed/downloaded anytime
- Survives page refreshes

Expected Output

Console Logs (Loading):

```
[Load Transactions] Processing 118 transactions for user khouston@thebasketballfactorynj.com
[Load Transactions] Statement Period: January 2024
[Load Transactions] Business Profile: The House of Sports (clxyz...)
[Load Transactions] Personal Profile: Personal/Household (clxyz...)
[Load Transactions] ✅ Created BankStatement: clxyz...
[Load Transactions] ✅ Saved: STRIPE PAYMENT → The House of Sports (BUSINESS)
[Load Transactions] 🎯 Complete: 118/118 transactions saved
[Load Transactions] 🏢 Business: 85 | 🏠 Personal: 33
```

UI Toasts:

1. “🤖 AI is classifying transactions as Business or Personal...”
2. “📊 Classified: 85 Business, 33 Personal”
3. “✅ Successfully loaded 85 Business + 33 Personal transactions!”

Saved Statements Display:

```
Saved Statements (1)
┌────────────────────────────────────────────────────────┐
│ Manual Entry - January 2024                            │
│ 118 transactions • January 2024 • 11/11/2025          │
│                                           [View] [Download] │
└────────────────────────────────────────────────────────┘
```

Status

✅ READY FOR TESTING

All fixes implemented and tested. Build successful with no TypeScript errors.