

CFO Budgeting App - Dynamic Data Flow

Overview

This app is **100% dynamic** - all financial data comes from user-uploaded documents. No static or hardcoded numbers exist in the application.

Data Sources

1. User-Uploaded Bank Statements (Primary Source)

Format: PDF files

Location: Financial Statements page (/dashboard/bank-statements)

Process:

1. User selects a business profile
2. Uploads PDF bank statement
3. AI automatically:
 - Extracts transactions
 - Categorizes each transaction
 - Creates/updates budgets
 - Links transactions to budgets

Supported Statement Types:

- Personal checking/savings accounts
- Business bank accounts
- Credit card statements

2. Manual Transaction Entry (Secondary Source)

Users can manually add transactions through the Transactions page.

3. CSV Import (Alternative Source)

Users can import transaction data via CSV upload.

Auto-Generated Data

Budgets

Location: Budget page (/dashboard/budget)

How it works:

- Automatically created when transactions are processed
- One budget per category per month/year
- Budget amounts = sum of transactions in that category
- Budget spent = actual amount spent
- Updates automatically when new statements are uploaded

Budget Calculation:

Budget Amount = Sum of all transactions in category **for** the period
 Budget Spent = Actual spending (negative amounts)
 Remaining = Budget Amount - Budget Spent

Transactions

Location: Transactions page (</dashboard/transactions>)

Fields Auto-Generated:

- Date (extracted from PDF)
- Description (extracted from PDF)
- Amount (extracted from PDF)
- Category (AI-categorized)
- Merchant (extracted if available)
- Source (tracks which statement it came from)

Categories

Location: Categories page (</dashboard/categories>)

How it works:

- Default categories created on signup
- AI assigns transactions to appropriate categories
- Users can create custom categories
- Categories can be modified/deleted

Empty States

No Data Yet

When a user first signs up, they see:

- Empty dashboard with prompts to upload statements
- Budget page showing “No budgets for this period”
- Transactions page showing “No transactions yet”
- Each page includes a call-to-action to upload statements

Upload Flow

1. **Go to Financial Statements** → </dashboard/bank-statements>
2. **Select Profile** → Choose which business profile to associate with
3. **Upload PDF** → Drag & drop or select PDF file
4. **Processing** → AI extracts and categorizes transactions
5. **Review** → View extracted transactions
6. **Approve** → Confirm and save to database

After Upload

- Transactions appear on Transactions page
- Budgets auto-generate on Budget page
- Dashboard updates with new financial metrics
- All charts and graphs populate with real data

Database Architecture

Key Models



Relationships

- One user can have multiple business profiles
- One profile can have multiple bank statements
- One statement generates multiple transactions
- Transactions create/update budgets
- Budgets are grouped by category and period

Testing the Dynamic System

Clean Start Test

1. Create a new user account
2. Verify all pages show empty states
3. Upload a bank statement PDF
4. Verify transactions are extracted
5. Verify budgets are auto-created
6. Verify dashboard updates with metrics

Multi-Statement Test

1. Upload first statement (e.g., September 2025)
2. Verify data for September
3. Upload second statement (e.g., October 2025)
4. Verify data for October is separate
5. Use month/year selector on Budget page to switch periods

Profile Separation Test

1. Create multiple business profiles
2. Upload statements to different profiles
3. Verify data is properly separated
4. Switch between profiles to view different data

No Static Data

Removed from Seed Script

The database seed script (`scripts/seed.ts`) has been cleaned to remove:

- ~~×~~ Sample transactions
- ~~×~~ Sample debts
- ~~×~~ Sample financial metrics
- ~~×~~ Hardcoded dollar amounts

What Seed Script Creates

The seed script only creates:

- Demo user account (john@doe.com / johndoe123)
- Default business profile
- Default category list (empty, no transactions)

Demo Account

Purpose: Testing and demonstration

Credentials:

- Email: john@doe.com
- Password: johndoe123

Note: Even the demo account starts empty. You must upload statements to populate data.

Production Use

For End Users

1. Sign up with real email
2. Create business profiles (Personal, Business, etc.)
3. Upload bank statement PDFs regularly
4. Review and approve extracted transactions
5. Use the Budget page to track spending
6. Use the Dashboard for overview

For Developers

1. Run `yarn prisma db seed` to create demo account
2. Login with demo credentials
3. Upload test PDFs to generate sample data
4. All data is real, extracted from uploaded documents
5. No mocking or fixtures needed

Data Privacy

- All uploaded PDFs are stored securely in cloud storage (S3)
- Extracted transaction data stored in PostgreSQL database
- Each user's data is completely isolated
- No data sharing between users
- Data only comes from user-uploaded documents

Continuous Upload

Users can:

- Upload unlimited statements
- Upload statements for any time period
- Re-upload statements if extraction fails
- Upload multiple statements in one session
- Upload different account types (checking, credit card, business)

System automatically:

- Prevents duplicate transactions
- Updates budgets incrementally
- Maintains transaction history
- Links everything to the correct business profile