Projections Widget - MVP

What You're Building

A financial projections widget that displays intelligent 12-month forward projections for Cash Flow, Revenue, and Net Profit. This widget uses actual seasonality patterns from historical data - NOT simple averages or flat growth rates.

Widget Visual Requirements

Chart Display:

- **Single line chart showing all 3 metrics simultaneously**
- **6 total lines:**
  - **Cash Flow: blue solid (historical, last 12mo) + blue dashed (projected, next 12mo)**
  - **Revenue: green solid (historical, last 12mo) + green dashed (projected, next 12mo)**
  - **Net Profit: purple solid (historical, last 12mo) + purple dashed (projected, next 12mo)**
- **Dual y-axis to handle different scales (revenue might be $500K while profit is $50K)**
- **Vertical line or visual separator between historical and projected data**
- **Markers/flags on timeline showing upcoming recurring expenses (e.g., "Apr 2025: $12K insurance")**
- **Hover tooltips showing exact values for each data point**

Insights Section (below chart): Display 3-5 automatically detected insights:

- **"Seasonality detected: December typically +25%, July typically -10%"**
- **"Recurring expense: $12K insurance payment due April 2025"**
- **"Trend: 5% year-over-year growth"**

How The Projection Algorithm Works

Step 1: Analyze Historical Patterns

- **Look back 24 months (or 12 months if 24 not available, or whatever exists if less than 12)**
- **Group data by calendar month (all Januaries together, all Februaries together, etc.)**

- **Calculate each month's typical variance from the annual average**
- **Example output: December is typically +25% above baseline, July is typically -10% below baseline**
- **Store these as seasonality multipliers**

Step 2: Detect Recurring Expenses

- **Find transactions >$2,000 that repeat annually within ±7 days**
- **Examples: insurance payments, quarterly taxes, annual subscriptions, equipment leases**
- **Flag these with their expected dates and amounts**

Step 3: Calculate Base Trajectory

- **Determine overall business trend (year-over-year growth rate)**
- **This trend gets applied ON TOP of seasonality, not instead of it**

Step 4: Generate Projections For each of the next 12 months, calculate:

```
projected_value = last_known_value
                × (1 + trajectory_growth_rate)
                × seasonality_multiplier_for_that_month
                - recurring_expenses_due_that_month
```

Example calculation:

- **Last December revenue: $100K**
- **Trajectory: +5% YoY growth**
- **December seasonality: +25% above baseline**
- **No recurring expenses in December**
- **Result: $100K × 1.05 × 1.25 = $131,250**

The key: projections should show the seasonal wave pattern, not a smooth line.

Database Work

Tables You'll Query:

- `monthly_pl` **- monthly revenue, expenses, net profit**
- `monthly_cash_flow` **- monthly cash inflows/outflows**
- `raw_transactions` **- to detect recurring large expenses**

Table You'll Create and Populate:

sql

```sql
CREATE TABLE IF NOT EXISTS projections_12m (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  company_id UUID REFERENCES companies(id) NOT NULL,
  month DATE NOT NULL,
  projected_revenue NUMERIC,
  projected_expenses NUMERIC,
  projected_cash_flow NUMERIC,
  projected_net_profit NUMERIC,
  seasonality_factor NUMERIC,
  recurring_expenses_flagged JSONB,
  generated_at TIMESTAMP DEFAULT NOW(),
  UNIQUE(company_id, month)
);
```

SQL Function to Build: Create `fn_generate_projection(company_id UUID)` that:

1. **Analyzes historical patterns**
2. **Detects recurring expenses**
3. **Calculates trajectory**
4. **Generates 12 months of projections**
5. **Inserts results into `projections_12m` table**

Frontend Requirements

Component Location: This widget lives on the main Overview dashboard page alongside other widgets.

What to Build:

- **React component that fetches projection data from Supabase**
- **Line chart with 6 lines (3 solid historical, 3 dashed projected)**
- **Use a charting library (Recharts, Chart.js, or your preference)**
- **Insights section below chart**
- **Responsive design (works on desktop and mobile)**

Color Coordination: Ben is handling the overall color palette - use blue for cash flow, green for revenue, purple for profit as placeholders. He'll adjust colors to match the design system.

Scheduled Updates

Set up a nightly job (Pipedream or Supabase cron) that:

- **Runs every night at 2 AM**
- **Calls `fn_generate_projection(company_id)` for each active company**
- **Keeps projections fresh as new transaction data comes in**

Edge Cases to Handle

Less than 24 months of data:

- **Prefer 24 months for seasonality detection**
- **Fall back to 12 months if 24 not available**
- **If less than 12 months, use whatever data exists (projections will be less accurate but still useful)**

New companies with minimal data:

- **Show a message: "Projections will improve as we collect more historical data"**
- **Still generate basic projections based on available data**

Missing data for specific months:

- **Interpolate or skip those months in seasonality calculation**
- **Don't let gaps crash the algorithm**

Testing Requirements

Before marking complete:

- **Run projection function for Budd's Company**
- **Verify projections show seasonal patterns (not flat growth)**
- **Confirm recurring expenses are flagged correctly**
- **Chart renders with all 6 lines visible**
- **Hover tooltips work**
- **Insights display detected patterns**
- **Widget loads in <2 seconds**
- **Mobile responsive**
- **Take screenshot of working widget**

Deliverables

1. **SQL function: `fn_generate_projection(company_id)`**
2. **`projections_12m` table populated with data**
3. **Frontend widget component**
4. **Nightly scheduled job configured**