What You're Building

A widget that allows users to compare key financial metrics across different time periods (week, month, quarter, year) with both sequential and year-over-year comparisons. The widget displays a side-by-side bar chart with AI-generated insights explaining the changes.

Widget Requirements

User Controls:

Dropdown 1 - Time Period Selection (8 options):

- Last Week vs Previous Week
- Last Month vs Previous Month
- Last Quarter vs Previous Quarter
- Last Year vs Previous Year
- This Week vs Same Week Last Year
- This Month vs Same Month Last Year
- This Quarter vs Same Quarter Last Year
- This Year vs Last Year

Dropdown 2 - Metric Selection (5 options):

- Gross Revenue
- COGS (Cost of Goods Sold)
- Gross Margin
- Fixed Overhead
- Net Margin

Visual Display

Main Chart Section:

Side-by-side horizontal bar chart showing:

- **Top bar:** Selected period (e.g., "Nov 2024: $80K")
- **Bottom bar:** Comparison period (e.g., "Nov 2023: $57K")
- **Below bars:** Change indicator showing both dollar and percentage: "+$23K (+40%) ↑"

Color Coding Rules:

- **Green bars:** Metric improved (revenue/margin up, costs down)
- **Red bars:** Metric declined (revenue/margin down, costs up)
- **Yellow bars:** No significant change (within ±2%)

Important: For COGS and Fixed Overhead, the logic reverses:

- Decrease = Green (good, costs went down)
- Increase = Red (bad, costs went up)

Supporting Context Section (below chart):

Show related metrics as text to provide context:

If user selects **Gross Margin**, show:

- Revenue: $120K (+26%)
- COGS: $40K (+5%)

If user selects **Net Margin**, show:

- Gross Margin: $80K (+40%)
- Fixed Overhead: $45K (+8%)

If user selects **Gross Revenue**, show:

- COGS: $40K (+5%)
- Gross Margin: $80K (+40%)

If user selects **COGS** or **Fixed Overhead**, show:

- Revenue: $120K (+26%)
- Net Margin: $35K (+67%)

AI Insights Section

Below the chart, display a short AI-generated paragraph (3-4 sentences) that explains:

1. **Why the metric changed** (seasonality, expense shifts, growth trends)
2. **Whether the change is good or bad** for the business
3. **Recommended action or observation**

Example Insight: "Your gross margin increased 40% compared to last November due to higher revenue (+26%) while keeping COGS growth minimal (+5%). This is excellent - you're scaling efficiently. Consider reinvesting some of this margin into growth initiatives while maintaining cost discipline."

AI Generation:

- Use Gemini or Claude to generate insights
- Provide context: selected metric, time periods, values, percentage changes, related metrics
- Keep response concise (3-4 sentences max)

Metric Calculation Definitions

Gross Revenue:

- Sum of all income/revenue transactions
- Query: `SELECT SUM(amount) FROM raw_transactions WHERE account_type = 'Income'`

COGS (Cost of Goods Sold):

- Sum of direct costs to deliver product/service
- Query: `SELECT SUM(amount) FROM raw_transactions WHERE category_id IN (SELECT id FROM categories WHERE type = 'COGS')`

Gross Margin:

- Calculation: `Gross Revenue - COGS`
- Display both dollar amount and percentage: `Gross Margin % = (Gross Margin / Gross Revenue) × 100`

Fixed Overhead:

- Sum of operating expenses (rent, salaries, utilities, admin costs)
- Query: `SELECT SUM(amount) FROM raw_transactions WHERE category_id IN (SELECT id FROM categories WHERE type = 'Operating Expense')`

Net Margin:

- Calculation: `Gross Margin - Fixed Overhead`
- Display both dollar amount and percentage: `Net Margin % = (Net Margin / Gross Revenue) × 100`

Database Queries

For Monthly/Quarterly/Yearly comparisons: Query `monthly_pl` table and aggregate as needed:

sql
```sql
-- Example: Last Month vs Previous Month
SELECT
  SUM(revenue) as gross_revenue,
  SUM(cogs) as cogs,
  SUM(revenue) - SUM(cogs) as gross_margin,
  SUM(expenses) as fixed_overhead,
  SUM(revenue) - SUM(cogs) - SUM(expenses) as net_margin
FROM monthly_pl
WHERE company_id = :company_id
  AND month = :selected_month
```

For Weekly comparisons: Query `raw_transactions` and aggregate by week:

sql
```sql
-- Example: This Week vs Same Week Last Year
SELECT
  DATE_TRUNC('week', date) as week,
  SUM(CASE WHEN account_type = 'Income' THEN amount ELSE 0 END) as revenue,
  SUM(CASE WHEN category_type = 'COGS' THEN amount ELSE 0 END) as cogs,
  -- etc
FROM raw_transactions
WHERE company_id = :company_id
  AND date BETWEEN :week_start AND :week_end
GROUP BY DATE_TRUNC('week', date)
```

Frontend Requirements (Next.js)

Component Structure:

- `PerformanceComparisonWidget.tsx` - Main container
- `ComparisonChart.tsx` - Horizontal bar chart component
- `ComparisonInsights.tsx` - AI-generated text section

Chart Library: Use Recharts or Chart.js for the horizontal bar chart.

Example Recharts structure:

tsx

```
<BarChart layout="vertical" data={comparisonData}>
  <XAxis type="number" />
  <YAxis type="category" dataKey="period" />
  <Bar dataKey="value" fill={getBarColor(change)} />
  <Tooltip />
</BarChart>
```

Responsive Design:

- Desktop: Full-width chart with dropdowns side-by-side
- Mobile: Stack dropdowns vertically, chart scales to fit screen

---

Edge Cases to Handle

No data for comparison period:

- Display message: "No data available for [time period]. Comparison cannot be generated."

Division by zero (calculating percentage change):

- If previous period = $0, show: "N/A (no prior data)"

Negative values:

- Handle negative net margin appropriately
- Color coding: moving toward $0 from negative = green (improvement)

First year of operation:

- Year-over-year comparisons won't work
- Show message: "Year-over-year data not yet available"

Testing Checklist

- Both dropdowns work and trigger chart refresh
- All 8 time period options calculate correctly
- All 5 metrics calculate correctly
- Bar colors match improvement/decline logic
- COGS and Fixed Overhead use reversed color logic (decrease = green)
- Supporting context shows relevant related metrics

- AI insights generate successfully
- Insights are concise (3-4 sentences)
- Dollar and percentage changes display correctly
- Chart renders properly on desktop
- Chart renders properly on mobile
- Widget loads in <2 seconds
- Edge cases handled gracefully

Deliverables

1. **Backend:**
    - SQL queries for all time period calculations
    - Metric calculation functions
    - AI insight generation integration
2. **Frontend:**
    - Complete widget component
    - Horizontal bar chart with color coding
    - Dropdown controls working
    - AI insights section displaying
    - Responsive design
3. **Testing:**
    - Test with real company data (Budd's)
    - Verify all time periods work
    - Verify all metrics calculate correctly
    - Screenshot of working widget
4. **Documentation:**
    - Loom video walkthrough showing:
        - Dropdown selections
        - Chart updating
        - Different metrics and time periods
        - AI insights generation

Start with the metric calculation logic and database queries first. Once those are solid and tested, build the frontend visualization and integrate AI insights last.