Invoice Total Amount Calculation Fix - Verification

Problem Summary

The invoice total amount was not properly including expenses. The implementation had two major issues:

Issue 1: Incorrect Margin Calculation

- **Before**: Margin was calculated on (baseAmount + expenses)
- **After**: Margin is calculated ONLY on baseAmount (work)

Issue 2: Fragile String Matching

- **Before**: Frontend relied on checking if line item descriptions contained “expense”
- **After**: Frontend uses actual Expense model data from `timesheet.expenses`

Changes Made

Backend Changes (`server/api/routers/invoice.ts`)

1. Updated `getById` query (lines 219-310)

- Added `timesheet` with `expenses` to the include statement
- Now properly fetches expenses from the Expense model

2. Fixed `createFromTimesheet` calculation (lines 1307-1320)

```
// OLD (INCORRECT):
const invoiceAmount = baseAmount.add(totalExpenses)
const marginCalculation = await MarginService.calculateMarginFromContract(
  timesheet.contractId!,
  parseFloat(invoiceAmount.toString())  // ❌ Margin on work + expenses
)
const totalAmount = marginCalculation?.totalWithMargin || invoiceAmount

// NEW (CORRECT):
const marginCalculation = await MarginService.calculateMarginFromContract(
  timesheet.contractId!,
  parseFloat(baseAmount.toString())  // ✅ Margin on work only
)
const marginAmount = new Prisma.Decimal(marginCalculation?.marginAmount || 0)
const totalAmount = baseAmount.add(marginAmount).add(totalExpenses)  // ✅ Total = work + margin + expenses
```

3. Updated invoice data (line 1358)

```
// OLD:
amount: invoiceAmount,  // ❌ Was baseAmount + expenses

// NEW:
amount: baseAmount,  // ✅ Work amount only
```

Frontend Changes (`app/(dashboard)/(modules)/invoices/[id]/page.tsx`)

1. Updated `lineItemsTotals` calculation (lines 163-183)

```
// OLD (INCORRECT):
data.lineItems.forEach((item: any) => {
  const amount = Number(item.amount || 0);
  if (item.description?.toLowerCase().includes('expense')) {  // ❌ String matching
    expenses += amount;
  } else {
    workTotal += amount;
  }
});

// NEW (CORRECT):
const workTotal = Number(data.baseAmount || data.amount || 0);  // ✅ From backend

let expenses = 0;
if (data.timesheet?.expenses) {  // ✅ From Expense model
  expenses = data.timesheet.expenses.reduce((sum: number, expense: any) => {
    return sum + Number(expense.amount || 0);
  }, 0);
}
```

2. Updated expenses display section (lines 834-876)

- Now displays expenses from `data.timesheet.expenses` (Expense model)
- Shows detailed expense information: title, description, category, date, amount
- No longer relies on string matching

3. Updated line items table (lines 800-816)

- Removed the filter for “expense” in descriptions
- All line items are now work-related (expenses shown separately)

Correct Calculation Logic

```
baseAmount = Sum of all timesheet entries (work hours × rate)
marginAmount = Margin calculated ONLY on baseAmount
totalExpenses = Sum of all expenses from Expense model
totalAmount = baseAmount + marginAmount + totalExpenses
```

Example Calculation

Given:
- baseAmount = $5,000 (work)
- margin = 10%
- expenses = $500

Correct Calculation:

1. baseAmount = $5,000
2. marginAmount = $5,000 × 10% = $500
3. totalExpenses = $500
4. **totalAmount = $5,000 + $500 + $500 = $6,000** ✅

Old (Incorrect) Calculation:
1. baseAmount = $5,000
2. invoiceAmount = $5,000 + $500 = $5,500
3. marginAmount = $5,500 × 10% = $550 ❌ (margin should not apply to expenses)
4. **totalAmount = $5,500 + $550 = $6,050** ❌ (WRONG!)

Difference: The old calculation incorrectly added $50 extra (margin on expenses)

Verification Checklist

- [x] Backend fetches expenses from Expense model
- [x] Margin calculated only on baseAmount (work)
- [x] totalAmount = baseAmount + marginAmount + totalExpenses
- [x] Frontend uses actual Expense model data
- [x] No string matching for identifying expenses
- [x] Line items display only work entries
- [x] Expenses display separately with full details

Files Modified

1. `server/api/routers/invoice.ts`
   - Line 272-276: Added timesheet with expenses to getById query
   - Lines 1307-1320: Fixed margin calculation logic
   - Line 1358: Fixed amount field

2. `app/(dashboard)/(modules)/invoices/[id]/page.tsx`
   - Lines 163-183: Fixed lineItemsTotals calculation
   - Lines 800-816: Removed string matching from line items
   - Lines 834-876: Updated expenses display to use Expense model

Benefits

1. **Accurate Calculations**: Margin is only applied to work, not expenses
2. **Data Integrity**: Uses proper Expense model instead of string matching
3. **Maintainability**: No fragile string matching logic
4. **Scalability**: Expenses properly linked through database relations
5. **Better UX**: Detailed expense information (category, date, description)