Invoice Expense Calculation Fix - Summary

✅ Task Completed Successfully

Commit Information

- **Commit Hash**: `d39e637`
- **Commit Message**: `fix: include expenses in invoice total amount`
- **Branch**: `expenses-structure`

🔧 Changes Made

1. Backend Fixes (`server/api/routers/invoice.ts`)

A. Updated `getById` Query (lines 271-276)

Added timesheet with expenses to properly fetch expense data:

```
// 🔥 NEW: Include timesheet with expenses
timesheet: {
  include: {
    expenses: true,
  },
},
```

B. Fixed Margin Calculation (lines 1312-1320)

Before (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
```

After (CORRECT):

```
// 🔥 FIX: Calculate margin ONLY on baseAmount (work), not on expenses
const marginCalculation = await MarginService.calculateMarginFromContract(
  timesheet.contractId!,
  parseFloat(baseAmount.toString())  // ✅ Margin on work only
)

// 🔥 FIX: Total = baseAmount + margin + expenses
const marginAmount = new Prisma.Decimal(marginCalculation?.marginAmount || 0)
const totalAmount = baseAmount.add(marginAmount).add(totalExpenses)
```

C. Fixed Invoice Amount Field (line 1358)

```
amount: baseAmount,  // 🔥 FIX: amount should be baseAmount only (work)
```

---

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

A. Fixed Totals Calculation (lines 163-183)

Before (INCORRECT - String Matching):

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

After (CORRECT - Uses Expense Model):

```
// 🔥 FIX: Use baseAmount for work total (already calculated on backend)
const workTotal = Number(data.baseAmount || data.amount || 0);

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

B. Updated Expenses Display (lines 834-876)

Now displays expenses from the Expense model with full details:
- Title and description
- Category
- Date
- Amount

Before: Filtered line items by checking if description contains “expense”
After: Uses `data.timesheet.expenses` from the Expense model

C. Removed Line Item Filtering (lines 800-816)

Removed the fragile `.filter((item) => !item.description?.toLowerCase().includes('expense'))` since expenses are now displayed separately from the Expense model.

📊 Correct Calculation Formula

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

Example:

- Base Amount (work): **$5,000**
- Margin (10%): **$500** (10% of $5,000)
- Expenses: **$500**
- **Total Amount: $6,000** ✅

✨ Benefits

1. ✅ **Accurate Calculations**: Margin only applies to work, not expenses
2. ✅ **Data Integrity**: Uses proper Expense model with database relations
3. ✅ **No String Matching**: Removed fragile string-based logic
4. ✅ **Better UX**: Displays detailed expense information (category, date, description)
5. ✅ **Maintainable**: Clean separation of work items and expenses
6. ✅ **Scalable**: Proper database relationships for future features

---

🧪 Verification

See `INVOICE_EXPENSE_FIX_VERIFICATION.md` for detailed verification and test cases.

📝 Files Modified

1. **server/api/routers/invoice.ts**
   - Added timesheet.expenses to getById query
   - Fixed margin calculation to use baseAmount only
   - Fixed totalAmount calculation

2. **app/(dashboard)/(modules)/invoices/[id]/page.tsx**
   - Removed string matching logic
   - Use Expense model data
   - Display expenses with full details

🚀 Next Steps

The fix is complete and committed. To deploy:

1. Review the changes in the code editor
2. Test with an existing invoice that has expenses
3. Verify the calculation is correct
4. Push the commit to remote when ready

Status: ✅ COMPLETED
Date: December 21, 2025