🔍 Schema Alignment Matrix

Systematic identification and resolution of schema mismatches

📋 Status Legend

- ✅ **Aligned** - No issues
- ❌ **Mismatch** - Needs fixing
- ⚠️ **Partial** - Some fields misaligned
- 🔄 **In Progress** - Being fixed

1. 📄 Document Model

Field	Prisma Schema	API Usage	Frontend	Status	Fix Required
id	`String @id`	❌ Missing in create	N/A	❌	Add to API create
filename	`String`	✅ `file.name`	N/A	✅	None
original-Name	`String`	✅ `file.name`	N/A	✅	None
mimetype	`String`	✅ `file.type`	N/A	✅	None
size	`Int`	✅ `file.size`	N/A	✅	None
cloudStoragePath	`String`	✅ `cloudStoragePath`	N/A	✅	None
uploaded-ById	`String?`	✅ `session.user?.id`	N/A	✅	None
category	`DocumentCategory`	✅ `'GENERAL'`	N/A	✅	None
description	`String?`	✅ `'AI Assistant...'`	N/A	✅	None
updatedAt	`DateTime`	❌ Missing in create	N/A	❌	Add to API create
createdAt	`DateTime @default(now())`	✅ Auto-generated	N/A	✅	None

🔧 **Fix**: Add missing fields to document creation API

2. 👥 Client Model

Field/Relation	Prisma Schema	API Usage	Frontend	Status	Fix Required
id	`String @id`	❌ Missing in create	TBD	❌	Add to client create
name	`String`	✅ Correct	✅	✅	None
email	`String?`	✅ Correct	✅	✅	None
clientNumber	`String? @unique`	✅ Generated	✅	✅	None
createdById	`String`	✅ Correct	✅	✅	None
updatedAt	`DateTime`	❌ Missing in create	TBD	❌	Add to client create
Project (relation)	`Project[]`	❌ Using `projects`	TBD	❌	Change to `Project`
ClientInvoice (relation)	`ClientInvoice[]`	❌ Using `clientInvoices`	TBD	❌	Change to `ClientInvoice`
_count.Project	Auto-generated	❌ Using `projects`	TBD	❌	Change to `Project`
_count.ClientInvoice	Auto-generated	❌ Using `clientInvoices`	TBD	❌	Change to `ClientInvoice`

🔧 **Fix**: Update relation names from lowercase to PascalCase

3. 🛒 Purchase Order Model

Field/Relation	Prisma Schema	API Usage	Frontend	Status	Fix Required
id	`String @id`	✅ Correct	✅	✅	None
poNumber	`String @unique`	✅ Auto-generated	✅	✅	None
vendorId	`String`	✅ Correct	✅	✅	None
requesterId	`String`	✅ Correct	✅	✅	None
createdById	`String`	❌ Using `createdBy`	TBD	❌	Change to `createdById`
approvedById	`String?`	❌ Using `approvedBy`	TBD	❌	Change to `approvedById`
Vendor (relation)	`Vendor`	✅ `vendor`	✅	✅	None
Project (relation)	`Project?`	❌ Using `project`	✅	❌	Change to `Project`
User (requester)	Complex relation	❌ Using `requester`	TBD	❌	Fix relation name
PurchaseOrderItem[]	`PurchaseOrderItem[]`	❌ Using `items`	TBD	❌	Change to `PurchaseOrderItem`
VendorInvoice[]	`VendorInvoice[]`	❌ Missing `vendorInvoices`	TBD	❌	Add proper include

🔧 **Fix**: Update all relation names to match Prisma schema exactly

4. 👤 User Model

Field	Prisma Schema	API Usage	Frontend	Status	Fix Required
id	`String @id`	❌ Missing in create	TBD	❌	Add to user create
email	`String @unique`	✅ Correct	✅	✅	None
firstName	`String?`	✅ Correct	✅	✅	None
lastName	`String?`	✅ Correct	✅	✅	None
role	`UserRole`	✅ Correct	✅	✅	None
password	`String?`	✅ Hashed	✅	✅	None
updatedAt	`DateTime`	❌ Missing in create	TBD	❌	Add to user create
createdAt	`DateTime @default(now())`	✅ Auto-generated	✅	✅	None

🔧 **Fix**: Add missing required fields to user creation

🚨 Critical Fixes Required

Priority 1: Relation Name Fixes

```
// ❌ WRONG - Current API usage
include: {
  projects: { ... },          // Should be: Project
  clientInvoices: { ... },    // Should be: ClientInvoice
  items: { ... },             // Should be: PurchaseOrderItem
  createdBy: { ... },         // Should be: relation by createdById
  approvedBy: { ... },        // Should be: relation by approvedById
}

// ✅ CORRECT - What it should be
include: {
  Project: { ... },
  ClientInvoice: { ... },
  PurchaseOrderItem: { ... },
  User_PurchaseOrder_createdByIdToUser: { ... },
  User_PurchaseOrder_approvedByIdToUser: { ... },
}
```

Priority 2: Missing Required Fields

```
// ❌ WRONG - Missing required fields
await prisma.document.create({
  data: {
    filename: file.name,
    // Missing: id, updatedAt
  }
})

// ✅ CORRECT - With required fields
await prisma.document.create({
  data: {
    id: generateId(),
    filename: file.name,
    updatedAt: new Date(),
    // ... other fields
  }
})
```

Priority 3: Count Field Alignment

```
// ❌ WRONG
_count: {
  projects: true,
  clientInvoices: true
}

// ✅ CORRECT
_count: {
  Project: true,
  ClientInvoice: true
}
```

⚒ Action Items Checklist

Phase 1: Critical API Fixes

- [ ] Fix Document API: Add `id` and `updatedAt` fields
- [ ] Fix Client API: Change `projects` → `Project`
- [ ] Fix Client API: Change `clientInvoices` → `ClientInvoice`
- [ ] Fix Client API: Add missing create fields
- [ ] Fix PurchaseOrder API: Change `createdBy` → `createdById` relation
- [ ] Fix PurchaseOrder API: Change `approvedBy` → `approvedById` relation
- [ ] Fix PurchaseOrder API: Change `items` → `PurchaseOrderItem`
- [ ] Fix PurchaseOrder API: Add `VendorInvoice` include

Phase 2: Validation & Type Safety

- [ ] Add proper TypeScript interfaces for all models
- [ ] Update Zod/validation schemas
- [ ] Add proper error handling for missing fields
- [ ] Test all CRUD operations

Phase 3: Frontend Alignment

- [ ] Review frontend form submissions
- [ ] Update API response handling
- [ ] Test end-to-end workflows
- [ ] Update documentation

🎯 Quick Win Priority

Start Here (15 min fixes):
1. Client API relation names (`projects` → `Project`)
2. Purchase Order relation names (`createdBy` → proper relation)
3. Add missing `updatedAt` to document creation

Medium Effort (30 min fixes):
4. Purchase Order complex relations
5. User creation schema fixes
6. Type safety improvements

High Impact Testing:
7. Test quotation creation (relies on Client API)
8. Test PO creation (relies on Purchase Order API)
9. Test document upload (relies on Document API)

💡 Pro Tip: Fix one model at a time, run `yarn tsc --noEmit` after each fix to verify progress.