ComplianceOS - Developer Onboarding Guide

Welcome to ComplianceOS! This guide will help you get started with developing new features using the Antigravity IDE.

🚀 Quick Start

Prerequisites

1. **Node.js** 18+ installed
2. **Git** configured with GitHub access
3. **Supabase** account (for database)
4. **Antigravity IDE** or VS Code with Gemini extension

Local Development Setup

```
# 1. Clone the repository
git clone https://github.com/bslate90/ComplianceOS.git
cd ComplianceOS

# 2. Install dependencies
npm install

# 3. Set up environment variables
# Copy .env.example to .env.local and fill in:
# - NEXT_PUBLIC_SUPABASE_URL
# - NEXT_PUBLIC_SUPABASE_ANON_KEY
# - SUPABASE_SERVICE_ROLE_KEY

# 4. Run the development server
npm run dev

# 5. Open http://localhost:3000
```

📁 Project Structure

```
ComplianceOS/
├── src/
│   ├── app/                    # Next.js App Router pages
│   │   ├── (auth)/             # Auth pages (login, register)
│   │   ├── (dashboard)/        # Dashboard pages
│   │   │   ├── ingredients/    # Ingredient management
│   │   │   ├── recipes/        # Recipe management
│   │   │   ├── labels/         # Label generation
│   │   │   ├── suppliers/      # Supplier management
│   │   │   └── organization/   # Org settings
│   │   └── api/                # API routes
│   ├── components/             # React components
│   │   └── ui/                 # Reusable UI components (shadcn)
```

```
│   ├── lib/                    # Utilities and services
│   │   ├── supabase/           # Supabase clients
│   │   ├── compliance/         # FDA compliance logic
│   │   ├── export/             # PDF/export generators
│   │   └── integrations/       # External integrations (PLEX)
│   └── hooks/                  # Custom React hooks
├── supabase/
│   ├── migrations/             # Database migrations (SQL)
│   └── scripts/                # Admin SQL scripts
└── .agent/
    └── workflows/              # AI assistant workflows
```

🗄️ Database Schema

The app uses Supabase (PostgreSQL). Key tables:

Table	Purpose
`organizations`	Multi-tenant orgs
`profiles`	User profiles with roles
`ingredients`	Ingredient library with nutrition
`recipes`	Recipe formulations
`recipe_ingredients`	Junction table for recipe components
`labels`	Generated nutrition labels
`suppliers`	Supplier directory
`supplier_documents`	Supplier certifications/docs
`compliance_rules`	FDA compliance rules
`compliance_reports`	Generated compliance checks
`organization_audit_log`	Activity audit trail

🔧 Tech Stack

Layer	Technology
Framework	Next.js 16 (App Router)
Language	TypeScript
Styling	Tailwind CSS v4
UI Components	shadcn/ui (Radix primitives)
Database	Supabase (PostgreSQL)

Auth	Supabase Auth
PDF Generation	@react-pdf/renderer
Notifications	Sonner

🤖 Using Antigravity IDE

Slash Commands (Workflows)

Type these commands to trigger AI-assisted workflows:

- `/SETUP_SUPABASE_VERCEL` - Configure Supabase and Vercel connection
- `/CAPA_MANAGEMENT` - Develop CAPA tracking features
- `/SUPPLIER_SCORECARDS` - Build supplier scoring system
- `/SPC_DATA_IMPORT` - Create SPC data import functionality

Best Practices with AI

1. **Be Specific**: Describe the feature with requirements
2. **Reference Existing Code**: "Similar to how recipes work..."
3. **Ask for DB First**: "Create the database migration first"
4. **Review Changes**: Always review generated code
5. **Test Incrementally**: Build features step by step

Example Prompts

```
"Create a CAPA management system similar to the existing recipe
management. I need to track corrective actions with due dates,
responsible parties, and status tracking. Start with the database
migration."

"Add a supplier scorecard feature that calculates scores based on:
- Delivery performance (on-time %)
- Quality metrics (defect rate)
- Document compliance (cert expiration)
- Response time"
```

📋 Pending Features to Develop

1. CAPA Management

- Track corrective/preventive actions
- Due date tracking and reminders
- Root cause analysis fields
- Evidence attachment support
- Status workflow (Open → In Progress → Verification → Closed)

2. Supplier Scorecards

- Automated scoring algorithm
- Performance trend charts
- Risk categorization (High/Medium/Low)

- Supplier comparison dashboard
- Integration with existing supplier module

3. SPC Data Import

- Import from PLEX or CSV
- Statistical calculations (Cp, Cpk, control limits)
- Control chart visualization
- Out-of-spec alerts
- Historical data storage

🧪 Testing

```
# Type checking
npm run lint

# TypeScript validation
npx tsc --noEmit

# Build test (production)
npm run build
```

📮 Deployment

The app deploys to **Vercel** automatically on push to `main` branch.

```
# Commit and push changes
git add -A
git commit -m "feat: Description of changes"
git push origin main
```

🔐 Environment Variables

Required for local development:

Variable	Description
`NEXT_PUBLIC_SUPABASE_URL`	Supabase project URL
`NEXT_PUBLIC_SUPABASE_ANON_KEY`	Supabase anon/public key
`SUPABASE_SERVICE_ROLE_KEY`	Supabase service role key (backend only)
`NEXT_PUBLIC_APP_URL`	App base URL (for webhooks)

💡 Getting Help

1. Check existing code patterns in similar features
2. Use Antigravity AI with specific questions

3. Review the `/api` routes for backend patterns
4. Check `supabase/migrations` for DB schema examples

Happy coding! 🎉