

# CogniFloe POC Presentation Reference Guide

AI Workflow Automation Recommendation Engine *Prepared for POC Demo Presentation - January 6, 2026*

## Project Overview

CogniFloe is an AI-powered workflow automation platform that:

- Analyzes business process descriptions using NLP
- Recommends AI agents tailored to the workflow
- Provides ML-powered predictions and analytics

## Tech Stack (ACCURATE)

Layer	Technology	Details
Frontend	React + TypeScript	Vite build tool
Styling	Tailwind CSS	Custom design system
Animations	Framer Motion	Premium micro-interactions
Backend	Python FastAPI	RESTful API
Database	Supabase	PostgreSQL cloud database
AI/NLP	OpenAI GPT-4	Workflow analysis, complexity scoring, recommendations
ML Models	Custom Python	Random Forest simulation, Z-Score anomaly detection
Auth	JWT + bcrypt	Secure token-based auth

## AI API Used: OpenAI GPT-4 (NOT Gemini)

- File: `backend/services/ml_service.py` - Line 12, 20
- API Key: `OPENAI_API_KEY` in `.env`

```
from openai import OpenAI
client = OpenAI(api_key=os.getenv("OPENAI_API_KEY"))
```

## File Structure Overview

```
CogniFloe/
  ├── frontend/
  |   ├── src/
  |   |   ├── pages/          # 11 main page components
  |   |   ├── components/    # Reusable UI components
  |   |   ├── context/       # React contexts (Theme, Workflow)
  |   |   └── hooks/         # Custom hooks
  |   └── index.html        # Entry HTML
  └── backend/
      ├── api/              # API route handlers
      ├── services/         # Business logic + ML
      ├── models/            # Pydantic models
      └── main.py           # FastAPI app entry
```

## ¶ 1. Home / Landing Page

**Description:** Public landing page showcasing CogniFloe capabilities

**Demo Points:**

- Animated neural network background
- Theme toggle (dark/light)
- Floating orbs animation
- Responsive navbar with scroll shrink

Component	File	Lines
Home Page	frontend/src/pages/Home.tsx	1-270
Hero Section	frontend/src/components/home/Hero.tsx	All
Features Grid	frontend/src/components/home/Features.tsx	All
Navbar (scroll-aware)	frontend/src/pages/Home.tsx	17-60
Theme Toggle	frontend/src/components/ThemeToggle.tsx	All
Logo	frontend/src/components/Logo.tsx	All

## ¶ 2. Authentication (Login/Signup)

**Description:** JWT-based authentication with Supabase

**Backend Flow:**

1. User submits credentials → POST /api/auth/login
2. Backend validates with bcrypt → Returns JWT token
3. Token stored in localStorage → Used for all API calls

Component	File	Lines
Login Page	frontend/src/pages/Login.tsx	All (280 lines)
Signup Page	frontend/src/pages/Signup.tsx	All (320 lines)
Auth API	backend/api/auth.py	1-200
Auth Service	backend/services/auth_service.py	All

## ¶ 3. Dashboard Overview

**Description:** Main dashboard with key metrics and workflow status

**Demo Points:**

- 4 stat cards (Workflows, Agents, Tasks, Success Rate)
- Recent activity timeline
- Quick action buttons

Component	File	Lines
Dashboard Page	frontend/src/pages/Dashboard.tsx	All (700 lines)
Top Navigation	frontend/src/components/dashboard/TopNav.tsx	1-293
Global Search	frontend/src/components/GlobalSearch.tsx	All (Cmd+K)
Layout	frontend/src/components/Layout.tsx	1-50

Component	File	Lines
-----------	------	-------

## 4. Workflow Input & AI Analysis (KEY DEMO)

Description: Core feature - Describe workflow → AI generates custom agent team

### Demo Example

Input (text or voice):

```
"Process customer emails, extract invoice data from PDF attachments,
validate amounts against our database, update the CRM, and send
Slack notifications to the finance team."
```

AI Output:

Agent	Purpose
Email Processor Agent	Triage, classify emails
Document Intelligence Agent	OCR, PDF extraction
Data Validator Agent	Validate against database
CRM Integration Agent	Update customer records
Notification Dispatcher	Slack/email alerts
Workflow Orchestrator	Coordinate all agents

### Code References

Component	File	Lines	Description
Workflow Input Page	frontend/src/pages/WorkflowInput.tsx	All (900+ lines)	Main page
Agent Templates (10 types)	frontend/src/pages/WorkflowInput.tsx	40-260	Email, Document, Data, API, CRM, etc.
Agent Matching Logic	frontend/src/pages/WorkflowInput.tsx	296-360	Keyword-based matching
Voice Input (enhanced)	frontend/src/pages/WorkflowInput.tsx	648-774	Animated mic, wave visualizer
Workflow API	backend/api/workflow_api.py	All	CRUD operations
Analysis API	backend/api/analysis_api.py	25-67	AI analysis endpoint
Workflow Analyzer	backend/services/workflow_analyzer.py	ALL	Complexity, bottlenecks, ROI

### Voice Input Features (Lines 648-774)

- 4 animated pulsing rings when recording
- 12-bar audio wave visualizer
- Stop button transform animation
- Pulsing "Listening..." text
- Live transcript with gradient accent

## 5. Agent Catalog

Description: Marketplace-style browsing of all 20+ AI agents

Demo Points:

- Grid view with filter categories
- Search functionality
- Agent detail modal with capabilities

## Agent Categories

1. **Data Processing** - Extract, analyze, transform
2. **Communication** - Email, notifications
3. **Integration** - CRM, API, database
4. **Intelligence** - ML, NLP, sentiment
5. **Monitoring** - Security, compliance

Component	File	Lines
Agent Catalog Page	frontend/src/pages/AgentCatalog.tsx	All (780 lines)
Agent Cards	frontend/src/pages/AgentCatalog.tsx	280-400
Category Filters	frontend/src/pages/AgentCatalog.tsx	50-100
Agent Detail Modal	frontend/src/components/dashboard/AgentDetailModal.tsx	All

---

## ☒ 6. Agent Architectures

**Description:** Visual workflow diagram showing agent connections

**Demo Points:**

- Draggable agent nodes
- Auto-generated connection lines
- Visual data flow representation

Component	File	Lines
Agent Architectures	frontend/src/pages/AgentArchitectures.tsx	All (750 lines)
Connection Arrows	frontend/src/pages/AgentArchitectures.tsx	200-350
Drag & Drop Logic	frontend/src/pages/AgentArchitectures.tsx	100-200

---

## ☒ 7. AI Predictions (☒ KEY AI DEMO)

**Description:** ML-powered predictions using OpenAI GPT-4

**Features Powered by GPT-4**

1. **Workflow Complexity Analysis** - GPT-4 scores complexity
2. **Completion Time Prediction** - ML model + GPT insights
3. **Success Probability** - Risk assessment
4. **Smart Recommendations** - GPT-4 generated advice
5. **Anomaly Detection** - Z-Score statistical analysis

**Example: Sentiment Analysis**

```
Input: "The product quality exceeded my expectations!"
Output: {
  sentiment: "Positive",
  confidence: 0.94,
  emotions: ["Joy", "Satisfaction"],
  score: +0.87
}
```

Component	File	Lines	AI Used
ML Predictions Page	frontend/src/pages/MLPredictions.tsx	All (900 lines)	-
Prediction Charts	frontend/src/pages/MLPredictions.tsx	300-500	-
ML API Endpoints	backend/api/ml_api.py	All (164 lines)	GPT-4
<b>ML Service (GPT-4)</b>	backend/services/ml_service.py	1-408	<b>OpenAI GPT-4</b>
GPT-4 Complexity	backend/services/ml_service.py	26-81	GPT-4 analysis
GPT-4 Recommendations	backend/services/ml_service.py	83-117	GPT-4 recommendations
Anomaly Detector	backend/services/ml_service.py	303-402	Z-Score stats

## Backend ML Endpoints

```
# backend/api/ml_api.py
POST /api/ml/predict      # Time & success prediction (GPT-4 enhanced)
POST /api/ml/detect-anomalies # Anomaly detection (Z-Score)
GET  /api/ml/health        # Model health check
GET  /api/ml/model-info    # Model details
```

## GPT-4 Integration Code

```
# backend/services/ml_service.py - Lines 59-67
response = client.chat.completions.create(
    model="gpt-4",
    messages=[
        {"role": "system", "content": "You are an expert workflow analyst..."},
        {"role": "user", "content": prompt}
    ],
    temperature=0.3,
    max_tokens=300
)
```

---

## 8. Shadow Mode

**Description:** Test agent decisions without real execution - humans review before production

**Demo Points:**

- Pending decisions queue
- Approve/Reject with feedback
- Agents learn from human feedback

Component	File	Lines
Shadow Mode Page	frontend/src/pages/ShadowMode.tsx	All (680 lines)
Decision Cards	frontend/src/pages/ShadowMode.tsx	200-400
Approval Logic	frontend/src/pages/ShadowMode.tsx	100-180

---

## 9. Metrics Dashboard

**Description:** Real-time analytics with interactive charts (Recharts library)

**Demo Points:**

- Task success rate trends
- Processing time distribution
- Agent utilization breakdown
- Error rate monitoring

Component	File	Lines
Metrics Page	frontend/src/pages/Metrics.tsx	All (780 lines)
Charts (Recharts)	frontend/src/pages/Metrics.tsx	300-600
Metrics API	backend/api/metrics_api.py	All (240 lines)

## ⚙ 10. Settings

**Description:** User preferences and configuration

Component	File	Lines
Settings Page	frontend/src/pages/Settings.tsx	All (350 lines)

## ✉ New Features Added Today

### 1. Toast Notifications

```
// Usage: const toast = useToast(); toast.success("Saved!");  
File: frontend/src/components/ui/Toast.tsx
```

### 2. Global Search (Cmd+K)

```
Press Ctrl+K → Search pages, agents, actions  
File: frontend/src/components/GlobalSearch.tsx
```

### 3. Skeleton Loaders

```
<SkeletonCard /> <SkeletonAgent /> <SkeletonChart />  
File: frontend/src/components/ui/Skeleton.tsx
```

### 4. Enhanced Voice Input

```
Animated pulsing rings, 12-bar audio visualizer  
File: frontend/src/pages/WorkflowInput.tsx (Lines 648-774)
```

### 5. Predictive Maintenance Widget

```
ML-powered workflow health monitoring  
File: frontend/src/components/dashboard/PredictiveMaintenance.tsx
```

## ✉ Backend API Summary

## Authentication

Endpoint	Method	Description
/api/auth/signup	POST	Register new user
/api/auth/login	POST	Login, get JWT
/api/auth/me	GET	Get current user

## Workflows

Endpoint	Method	Description
/api/workflows	GET	List user workflows
/api/workflows	POST	Create workflow
/api/workflows/{id}	PUT	Update workflow
/api/analyze	POST	AI analysis

## ML/AI (GPT-4 Powered)

Endpoint	Method	AI Used
/api/ml/predict	POST	GPT-4 + Custom ML
/api/ml/detect-anomalies	POST	Z-Score Stats
/api/ml/health	GET	-

## Metrics

Endpoint	Method	Description
/api/metrics/overview	GET	Summary stats
/api/metrics/performance	GET	Performance trends

## ▀ Demo Script (8-10 minutes)

### Opening (30 sec)

"CogniFloe is an AI-powered workflow automation platform that uses GPT-4 to analyze business processes and recommend intelligent AI agents."

### Demo Flow

1. **Home Page** (30s) - Landing, theme toggle, animations
2. **Login** (30s) - Quick authentication demo
3. **Dashboard** (1m) - Overview metrics, Global Search (Cmd+K)
4. ▶ **Workflow Input** (2-3m) - Voice input, AI analysis demo
5. **Agent Catalog** (1m) - Browse agents, categories
6. **Agent Architectures** (1m) - Visual workflow diagram
7. ▶ **AI Predictions** (1-2m) - GPT-4 predictions, sentiment demo
8. **Shadow Mode** (30s) - Human-in-the-loop concept

### Closing (30 sec)

"CogniFloe reduces workflow setup time by 85% through AI-powered automation and continuous optimization using GPT-4."

---

## ⌘ Quick Commands

---

```
# Frontend (runs on localhost:5173)
cd frontend && npm run dev

# Backend (runs on localhost:8000)
cd backend
.\venv\Scripts\activate
python -m uvicorn main:app --reload

# API Docs
http://localhost:8000/docs
```

---

## ⌘ Common Questions

---

**Q: What AI API do you use?** A: OpenAI GPT-4 for NLP analysis and recommendations. Custom ML models for predictions.

**Q: Is this production-ready?** A: This is a POC demonstrating core AI capabilities. Production would need scaling.

**Q: How do agents communicate?** A: Through a workflow orchestrator that manages data flow between agents.

**Q: What about data security?** A: JWT auth, bcrypt password hashing, Supabase secure database.

---

*Document Last Updated: January 6, 2026 CogniFloe POC v1.0*