**Sovereign AI Agent Development System**

**🎯 Agent Architecture Overview**

**Agent 1: Backend Generator**

**Purpose**: Creates complete Flask backend code from specifications **Input**: API requirements, prompt templates, database schema **Output**: Complete Python Flask application with all endpoints

**Agent 2: Frontend Generator**

**Purpose**: Creates complete React/JavaScript frontend from API contracts **Input**: API specifications, UI requirements, user flows **Output**: Complete frontend code with API integration

**Agent 3: Prompt Engineer**

**Purpose**: Optimizes and tests GDPR analysis prompts **Input**: Legal requirements, test scenarios, output formats **Output**: Production-ready Claude prompts with validation

**Agent 4: Integration Orchestrator**

**Purpose**: Coordinates all components and handles testing **Input**: All generated components **Output**: Working end-to-end system with error handling

**🚀 Agent 1: Backend Generator**

**Prompt Template for Backend Agent:**

You are an expert Flask backend developer creating a production-ready API for GDPR compliance analysis.

REQUIREMENTS:

- Flask application with file upload capabilities

- PDF text extraction using PyPDF2 + Google Vision OCR fallback

- 5-step Claude API integration for GDPR analysis

- Export functionality (PDF + CSV)

- Comprehensive error handling

- CORS support for frontend integration

API ENDPOINTS NEEDED:

1. POST /api/upload-document (file upload + text extraction)

2. POST /api/analyze-compliance (5-step analysis)

3. GET /api/export/pdf (PDF report generation)

4. GET /api/export/csv (CSV action items)

INTEGRATION REQUIREMENTS:

- Claude API integration with rate limiting

- File storage and management

- JSON response formatting

- Error handling and fallbacks

GENERATE:

1. Complete app.py with all endpoints

2. requirements.txt with dependencies

3. Configuration management

4. Error handling middleware

5. Export functionality implementation

OUTPUT: Complete, production-ready Flask application code

**Backend Generation Process:**

**Step 1**: Agent analyzes your API specifications **Step 2**: Agent generates complete Flask application structure **Step 3**: Agent implements Claude API integration **Step 4**: Agent adds file handling and export capabilities **Step 5**: Agent creates deployment configuration

**🎨 Agent 2: Frontend Generator**

**Prompt Template for Frontend Agent:**

You are an expert React/JavaScript developer creating a polished frontend for GDPR compliance analysis.

REQUIREMENTS:

- Clean, professional UI for compliance analysis

- File upload with drag-and-drop support

- Real-time analysis progress indicators

- Results display with categorized findings

- Export functionality (PDF/CSV downloads)

- Error handling and loading states

USER FLOW:

1. Upload privacy policy PDF

2. Enter AI system description

3. Run analysis with progress tracking

4. Display results in organized sections

5. Export reports

COMPONENTS NEEDED:

1. FileUploadComponent (drag-drop PDF upload)

2. AnalysisFormComponent (AI system description)

3. ProgressIndicator (analysis steps visualization)

4. ResultsDisplayComponent (categorized findings)

5. ExportComponent (PDF/CSV download buttons)

API INTEGRATION:

- Fetch calls to Flask backend

- Error handling for failed requests

- File download management

- State management across components

STYLING:

- Professional, compliance-focused design

- Responsive layout for desktop/mobile

- Clear visual hierarchy

- Accessibility compliance

OUTPUT: Complete React application with all components and styling

**Frontend Generation Process:**

**Step 1**: Agent creates component architecture **Step 2**: Agent implements file upload with validation **Step 3**: Agent builds analysis form and progress tracking **Step 4**: Agent creates results display components **Step 5**: Agent integrates export functionality

**🔧 Agent 3: Prompt Engineer**

**Prompt Template for Prompt Engineer Agent:**

You are an expert prompt engineer specializing in GDPR compliance and legal analysis.

TASK: Create and optimize 5 specialized prompts for GDPR compliance analysis of AI systems.

LEGAL REQUIREMENTS:

- Accurate GDPR article identification (Articles 5, 6, 9, 13, 22, 25)

- Bias and fairness assessment

- Privacy policy gap analysis

- Implementation planning with actionable steps

PROMPT OBJECTIVES:

1. GDPR Risk Assessment - Identify violations and severity

2. Document Cross-Reference - Compare policy vs system reality

3. Bias & Fairness Analysis - Detect discrimination risks

4. Ethics & Governance Review - Assess oversight and transparency

5. Implementation Planning - Create actionable remediation steps

OUTPUT REQUIREMENTS:

- Structured JSON responses

- Consistent severity scoring (1-10)

- Actionable recommendations

- Legal accuracy and precision

TESTING CRITERIA:

- Prompts work with various AI system types

- Consistent output format

- Realistic risk assessments

- Legally sound recommendations

FOR EACH PROMPT PROVIDE:

1. Optimized prompt text

2. Expected JSON schema

3. Test case validation

4. Error handling instructions

OUTPUT: 5 production-ready prompts with validation and testing

**Prompt Engineering Process:**

**Step 1**: Agent analyzes GDPR requirements and case law **Step 2**: Agent creates specialized prompts for each analysis step **Step 3**: Agent tests prompts with multiple scenarios **Step 4**: Agent optimizes for consistency and accuracy **Step 5**: Agent validates JSON output formats

**🔄 Agent 4: Integration Orchestrator**

**Prompt Template for Integration Agent:**

You are a senior software architect responsible for integrating all components into a working system.

COMPONENTS TO INTEGRATE:

1. Flask backend (from Backend Generator)

2. React frontend (from Frontend Generator)

3. Claude prompts (from Prompt Engineer)

4. Export functionality

5. Error handling and fallbacks

INTEGRATION REQUIREMENTS:

- Seamless frontend-backend communication

- Proper error handling across all components

- Performance optimization

- Security considerations

- Deployment readiness

TESTING SCOPE:

- End-to-end user flows

- API endpoint validation

- Error scenario handling

- Export functionality verification

- Performance under load

DELIVERABLES:

1. Complete integration code

2. Deployment configuration

3. Testing documentation

4. Demo setup instructions

5. Troubleshooting guide

QUALITY ASSURANCE:

- All components work together

- Graceful error handling

- Professional user experience

- Demo-ready stability

OUTPUT: Complete, tested, deployment-ready application

**Integration Process:**

**Step 1**: Agent combines all generated components **Step 2**: Agent implements cross-component communication **Step 3**: Agent adds comprehensive error handling **Step 4**: Agent optimizes performance and reliability **Step 5**: Agent prepares deployment package

**📋 Agent Execution Workflow**

**Phase 1: Requirements Gathering (30 minutes)**

1. **Input Collection**: Gather all your specifications, requirements, and constraints
2. **Requirement Analysis**: Each agent analyzes their specific domain
3. **Architecture Planning**: Agents coordinate overall system design

**Phase 2: Parallel Development (2 hours)**

1. **Backend Generation**: Agent 1 creates complete Flask application
2. **Frontend Development**: Agent 2 builds React components
3. **Prompt Optimization**: Agent 3 refines GDPR analysis prompts
4. **Integration Planning**: Agent 4 prepares integration strategy

**Phase 3: Integration & Testing (1 hour)**

1. **Component Integration**: Agent 4 combines all components
2. **End-to-End Testing**: Validate complete user flows
3. **Error Handling**: Implement comprehensive fallbacks
4. **Demo Preparation**: Create demo-ready package

**Phase 4: Deployment & Documentation (30 minutes)**

1. **Deployment Package**: Ready-to-deploy application
2. **Documentation**: Complete setup and usage guides
3. **Team Handover**: Instructions for your team members
4. **Backup Systems**: Fallback options for demo day

**🎯 Expected Deliverables**

**For You (Divya):**

* Complete Flask backend with Claude integration
* Production-ready GDPR analysis prompts
* Integration documentation and deployment guide
* Demo script and backup systems

**For Sneha:**

* Complete Flask application ready to run
* Step-by-step setup instructions
* API testing documentation
* Troubleshooting guide

**For Sathish:**

* Complete React application with all components
* Integration instructions for backend connection
* Styling and responsive design
* Export functionality implementation

**For Your Team:**

* Working end-to-end application
* Demo scenarios and test data
* Setup instructions for each component
* Presentation-ready system

**🚀 Agent Activation Commands**

**To Start Backend Generation:**

"Activate Backend Generator Agent with Flask, Claude API integration, file upload, and export capabilities for GDPR compliance analysis system."

**To Start Frontend Generation:**

"Activate Frontend Generator Agent with React, file upload UI, analysis display, and export functionality for compliance dashboard."

**To Start Prompt Engineering:**

"Activate Prompt Engineer Agent to create 5 GDPR compliance analysis prompts with JSON output and legal accuracy."

**To Start Integration:**

"Activate Integration Orchestrator Agent to combine all components into working end-to-end GDPR compliance system."

**⚡ Immediate Benefits**

**Time Savings:**

* **Manual Development**: 7-10 days for your team
* **AI Agent Approach**: 4 hours total
* **Time Saved**: 90%+ reduction in development time

**Quality Improvements:**

* **Error-Free Integration**: Agents handle complex integration logic
* **Professional Polish**: Production-ready code quality
* **Comprehensive Testing**: Built-in validation and testing
* **Demo Reliability**: Robust error handling and fallbacks

**Team Support:**

* **Working Code**: Each team member gets functional components
* **Clear Instructions**: Step-by-step setup guides
* **No Technical Barriers**: Copy-paste implementation
* **Confidence**: Tested, validated system ready for demo

**🎪 Demo Day Success**

With this AI agent system, you'll have:

* ✅ **Working End-to-End System**: Upload → Analysis → Export
* ✅ **Professional Quality**: Production-ready code and UI
* ✅ **Reliable Performance**: Comprehensive error handling
* ✅ **Team Confidence**: Everyone knows their working components
* ✅ **Backup Plans**: Multiple fallback options
* ✅ **Impressive Demo**: Sophisticated AI compliance analysis

**Ready to activate the agents?** Let's start with the Backend Generator to get your Flask application running!