Decentralized Farmers Collective - n8n Automation

A modular n8n workflow system for decentralized agricultural communities featuring AI crop planning, blockchain microinvestments, mutual aid networks, and producer management.

Overview

This system provides a complete automation infrastructure for farmers' collectives, enabling:

- **AI-powered crop planning** and optimization recommendations
- **Blockchain microinvestments** for community-funded agricultural tools
- **Mutual aid networks** for resource sharing and skill exchange
- **Producer management** with yield tracking and analytics
- **Merchant trading** with tiered pricing and bulk order processing

Architecture

The system uses a modular orchestrator pattern where a main workflow routes requests to specialized sub-workflows:

```
Main Orchestrator
├── AI Crop Planning Sub-Workflow
├── Blockchain Microinvestment Sub-Workflow
├── Mutual Aid Network Sub-Workflow
├── Merchant Trading Sub-Workflow
└── Producer Management Sub-Workflow
```

Repository Structure

```
farmers-collective-n8n/
├── README.md
├── LICENSE
├── .gitignore
├── docs/
│   ├── SETUP.md
│   ├── DATABASE_SCHEMA.md
│   ├── API_REFERENCE.md
│   └── DEPLOYMENT.md
├── workflows/
│   ├── main-orchestrator.json
│   ├── ai-crop-planning.json
│   ├── blockchain-microinvestment.json
│   ├── mutual-aid-network.json
│   ├── merchant-trading.json
│   └── producer-management.json
├── database/
│   ├── schemas/
│   │   ├── orchestrator.sql
│   │   ├── ai_planning.sql
│   │   ├── microinvestment.sql
│   │   ├── mutual_aid.sql
│   │   ├── merchant_trading.sql
│   │   └── producer_management.sql
│   └── sample_data/
│       ├── users.sql
│       ├── producers.sql
│       └── test_data.sql
├── config/
│   ├── credentials.template.json
│   ├── environment.template.env
│   └── variables.json
├── examples/
│   ├── sample_requests/
│   │   ├── ai_scheduling.json
│   │   ├── wishlist_creation.json
│   │   ├── microinvestment.json
│   │   ├── mutual_aid.json
│   │   ├── merchant_purchase.json
│   │   └── production_report.json
│   └── test_workflows/
├── scripts/
│   ├── setup.sh
│   ├── deploy.sh
│   └── backup.sh
└── tests/
    ├── unit/
    ├── integration/
    └── data/
```

Quick Start

Prerequisites

- n8n instance (v1.0+)
- PostgreSQL database
- Node.js 18+
- Docker (optional)

Installation

1. Clone the repository:

```bash
git clone https://github.com/your-org/farmers-collective-n8n.git
cd farmers-collective-n8n
```

2. Set up environment variables:

```bash
cp config/environment.template.env .env
# Edit .env with your configuration
```

3. Initialize the database:

```bash
./scripts/setup.sh
```

4. Import workflows into n8n:

```bash
# Import each workflow JSON file through n8n interface
# Or use n8n CLI if available
```

5. Configure credentials in n8n:

- Database connections
- API keys (HuggingFace, blockchain providers)
- SMTP settings
- Discord/Telegram bot tokens

Core Features

Blockchain Microinvestment System

The microinvestment system enables community members to fund productivity-enhancing tools for producers:

Wishlist Creation

- Producers add needed tools with cost estimates
- System calculates ROI and payback periods
- Community can view and fund items

Investment Processing

- Blockchain token transactions
- Proportional ownership tracking
- Automated return distribution

Return Calculation

- Based on reported production increases
- Proportional distribution to investors
- Transparent blockchain ledger

AI Crop Planning

AI-powered recommendations for:

- Planting schedules optimized for location and season
- Crop rotation suggestions
- Harvest timing predictions
- Tool investment recommendations

Mutual Aid Network

Community support system featuring:

- Resource sharing (tools, equipment, space)
- Skill exchange matching
- Volunteer coordination
- Aid request prioritization

Database Schema

The system uses separate PostgreSQL databases for each workflow:

- **Orchestrator DB**: Request routing and logging
- **AI Planning DB**: Crop plans and recommendations
- **Microinvestment DB**: Wishlist, investments, returns
- **Mutual Aid DB**: Resources, skills, volunteers
- **Merchant Trading DB**: Orders, payments, merchants
- **Producer Management DB**: Profiles, reports, analytics

See `docs/DATABASE_SCHEMA.md` for detailed schema information.

API Reference

Main Orchestrator Endpoint

```
POST https://your-n8n-instance.com/webhook/collective-main
```

Request Format:

```json
{
  "action": "ai_schedule|merchant_purchase|mutual_aid_request|wishlist_add|producer_register",
  "role": "grower|merchant|comrade|producer|investor",
  "userId": "string",
  "data": {
    // Action-specific data
  }
}
```

See `docs/API_REFERENCE.md` for complete endpoint documentation.

Configuration

Environment Variables

```bash
# Database
DB_HOST=localhost
DB_PORT=5432
DB_USER=collective_user
DB_PASS=your_password

# APIs
HUGGINGFACE_API_KEY=hf_your_token
BLOCKCHAIN_API_KEY=your_blockchain_key
COLLECTIVE_WALLET=0x...

# Messaging
DISCORD_BOT_TOKEN=your_discord_token
TELEGRAM_BOT_TOKEN=your_telegram_token
SMTP_HOST=smtp.gmail.com
SMTP_USER=collective@farmersnetwork.org
```

Credential Templates

The `config/credentials.template.json` file contains templates for all required n8n credentials.

Testing

Sample Requests

Use the examples in `examples/sample_requests/` to test each workflow:

```bash
# Test AI crop planning
curl -X POST https://your-n8n-instance.com/webhook/collective-main \
  -H "Content-Type: application/json" \
  -d @examples/sample_requests/ai_scheduling.json
```

Integration Tests

Run the full test suite:

```bash
npm test
```

Deployment

Docker Deployment

```bash
docker-compose up -d
```

Manual Deployment

1. Set up production database
2. Configure environment variables
3. Import workflows to production n8n
4. Run deployment script: `./scripts/deploy.sh`

Contributing

1. Fork the repository
2. Create a feature branch
3. Test your changes thoroughly
4. Submit a pull request

Development Guidelines

- Test all workflow changes with sample data
- Update documentation for new features
- Follow the modular architecture pattern
- Ensure database migrations are reversible

Blockchain Integration

The system supports multiple blockchain networks:

- **Ethereum mainnet/testnets**
- **Polygon** (recommended for lower fees)
- **Other EVM-compatible networks**

Token payments and microinvestments are recorded on-chain for transparency and trust.

Community Features

Role-Based Access

- **Producers**: Create wishlists, report yields, receive investments
- **Merchants**: Purchase products, pay fees, place bulk orders
- **Investors**: Fund producer tools, receive proportional returns
- **Community Members**: Share resources, exchange skills, offer aid

Governance Integration

The system can be extended to include:

- Voting mechanisms for collective decisions
- Proposal systems for new features
- Reputation tracking for community members

Security Considerations

- All database credentials stored securely in n8n
- API keys use environment variables
- Blockchain transactions require wallet signatures
- Input validation on all endpoints
- Rate limiting on external API calls

Monitoring and Analytics

Built-in tracking for:

- Investment performance metrics
- Producer yield improvements
- Community engagement levels
- System usage statistics

Roadmap

- [ ] Mobile app integration
- [ ] Advanced analytics dashboard
- [ ] Multi-language support
- [ ] Integration with IoT sensors
- [ ] Carbon credit tracking
- [ ] Supply chain transparency features

Support

For issues and questions:

- Create GitHub issues for bugs
- Use discussions for feature requests
- Check documentation in `/docs`
- Review sample requests in `/examples`

License

MIT License - see LICENSE file for details.

Built for agricultural communities worldwide. Fork, adapt, and grow together.