

AI Integration Progress Tracker

Overview

This document tracks the progress of integrating Ollama AI with the Platform Dashboard. It serves as a central reference point for all AI integration tasks, file changes, and implementation status.

Current Status

Phase: AI Integration (Phase 3) - Starting Implementation

Target Completion: TBD

Progress: 5%

Implementation Plan

1. Database Updates

- ☒ Define `ollama_settings` table in `DatabaseStructure.md`
- ☒ Create SQL migration script: `src/scripts/sql/create_ollama_settings.sql`
- ☒ Create `schema_migrations` table script: `src/scripts/sql/create_schema_migrations.sql`
- ☒ Fix migration scripts to handle proper execution order
- ☐ Apply migration scripts to the database
- ☐ Verify table creation and default settings

2. Backend Files to Create/Modify

New Files

- ☐ `src/routes/ollama.js` - Endpoints for Ollama settings and models
- ☐ `src/services/ollamaService.js` - Service for Ollama API communication
- ☐ `src/services/aiModelService.js` - Service for AI model management
- ☐ `src/middleware/adminRequired.js` - Middleware to restrict admin-only routes

Modifications

- ☐ `src/server.js` - Add new routes
- ☐ `src/routes/chatbot.js` - Update to use Ollama for AI responses
- ☐ `src/database.js` - Add query functions for Ollama settings

3. Frontend Files to Create/Modify

New Files

- ☐ `client/src/components/settings/OllamaSettings.tsx` - Admin settings component for Ollama
- ☐ `client/src/components/ai-chatbot.tsx` - AI integration logic for Chatbot
- ☐ `client/src/services/ollamaService.ts` - API client for Ollama endpoints

- ☐ `client/src/components/ModelSelector.tsx` - Component for selecting AI models

Modifications

- ☐ `client/src/pages/Settings.tsx` - Add Ollama settings tab for admins
- ☐ `client/src/pages/Chatbot.tsx` - Integrate with Ollama models
- ☐ `client/src/services/chatbotService.ts` - Update to support model selection
- ☐ `client/src/types/index.ts` - Add new types for Ollama integration

Features Checklist

Admin Settings

- ☐ Ollama server connection settings
 - ☐ Host and port configuration
 - ☐ Connection testing
 - ☐ Default model setting
- ☐ Model management
 - ☐ Fetch and display available models
 - ☐ Sync models with Ollama server
 - ☐ Enable/disable models for users

User Chat Interface

- ☐ Model selection in chat interface
 - ☐ Display available models to users
 - ☐ Remember user's model preference
 - ☐ Show current model in chat
- ☐ Enhanced chat capabilities
 - ☐ Support for code blocks and formatting
 - ☐ Model-specific formatting
 - ☐ Error handling for model issues

Backend Integration

- ☐ Ollama API proxy
 - ☐ Chat message forwarding
 - ☐ Model list retrieval
 - ☐ Error handling
- ☐ Security measures
 - ☐ Admin-only settings access
 - ☐ Secure storage of Ollama settings
 - ☐ Rate limiting for API calls

Development Status

Component	Status	Next Steps	Assigned To	Priority
Database Schema	In Progress	Apply migration scripts	-	High

Component	Status	Next Steps	Assigned To	Priority
Ollama API Service	In Progress	Create routes	-	High
Admin Settings UI	Not Started	Create component	-	Medium
Chat Integration	Not Started	Update chat service	-	High
Model Selection	Not Started	Design UI component	-	Medium
Documentation	In Progress	Add API docs	-	Low

Implementation Notes

Ollama API Requirements

The integration will use the following Ollama API endpoints:

- `GET /api/tags` - List available models
- `POST /api/chat` - Chat completion

Security Considerations

- Only admins can configure Ollama settings
- Connection details stored securely in PostgreSQL
- Input validation on all user-provided data

Testing Approach

- Unit tests for ollamaService
- Integration tests for API endpoints
- Manual testing with real Ollama instance

Next Immediate Steps

1. Apply database migration scripts in order:
 - First: `create_schema_migrations.sql`
 - Then: `create_ollama_settings.sql`
2. Create routes/ollama.js for backend API endpoints
3. Implement admin settings component
4. Update chat service to support Ollama models

References

- [Ollama API Documentation](#)
- [AI Integration Plan](#)
- [DatabaseStructure.md](#)