

# Digital Patient Feedback & Experience Agent

The **NHS Patient Feedback Agent** is a conversational AI system built to collect, analyze, and process patient feedback in real-time via a chatbot interface. It is designed to improve patient engagement and identify service gaps by intelligently interacting with patients, categorizing feedback, and providing structured insight to healthcare administrators.

## Overview:

- Automates collection of post-visit feedback and satisfaction surveys
- Implements sentiment analysis on patient reviews and comments
- Generates trend analysis reports for service quality metrics
- Identifies recurring issues and improvement opportunities
- Tracks feedback resolution and follow-up actions
- Provides real-time alerts for critical patient concerns

## Tech Stack



### Backend

- FastAPI: REST & WebSocket APIs
- pydantic-ai: Tool-based agent logic and context injection
- MongoDB: Database for user records and feedback collection
- OpenAI GPT-4o-mini: Language model for intelligent conversation
- TextBlob: Sentiment analysis engine
- Uvicorn: ASGI server
- Logfire: Centralized logging and tracing



### Frontend

- React (Vite setup)
- Tailwind CSS for design
- WebSocket API for real-time chat



# Database Collections Overview

## Users\_collection

Field	Type	Description
name	str	Patient's name
password	str	Hashed password
nhs number	dict	{ number, age, gender, treatment, issue }

## Feedback\_collection

Field	Type
patient_name	str
nhs_number	str
satisfaction_rating	int
comments	str
category	str

## Core Features (Completed)

### 1. Authentication System

- Users register/login using **NHS number + password**
- On login, user data is fetched from MongoDB (**users\_collection**)
- Authenticated user info is stored in **localStorage**

### 2. WebSocket-Based Real-Time Chat

- Maintains persistent chat sessions using FastAPI WebSockets
- Sends NHS number on connection for **contextual personalization**

### 3. AI Agent with Contextual Awareness

- Agent is powered by `pydantic-ai` with tools for:
  - Sentiment analysis
  - Follow-up questions
  - Feedback form initiation (interactive)
  - Feedback saving (to MongoDB)
  - Categorization
  - Finding common issues
  - Generating trend analysis (admin-side)
  - Detecting critical issues (admin-side)

### 4. User Context Injection

- Agent receives `ctx.deps` including:
  - Patient name
  - Age, gender, treatment date
  - Health issue
  - Feedback state (satisfaction rating, comments)
- These are dynamically injected when the chat session starts
- Agent uses this to personalize its conversation

### 5. Stateful Conversation Logic

- Agent knows when it is "waiting" for a rating or comment
- When rating is received, it triggers follow-up question
- When comments are received, it:
  - Categorizes using keyword-based matching
  - Saves feedback to `feedback_collection`

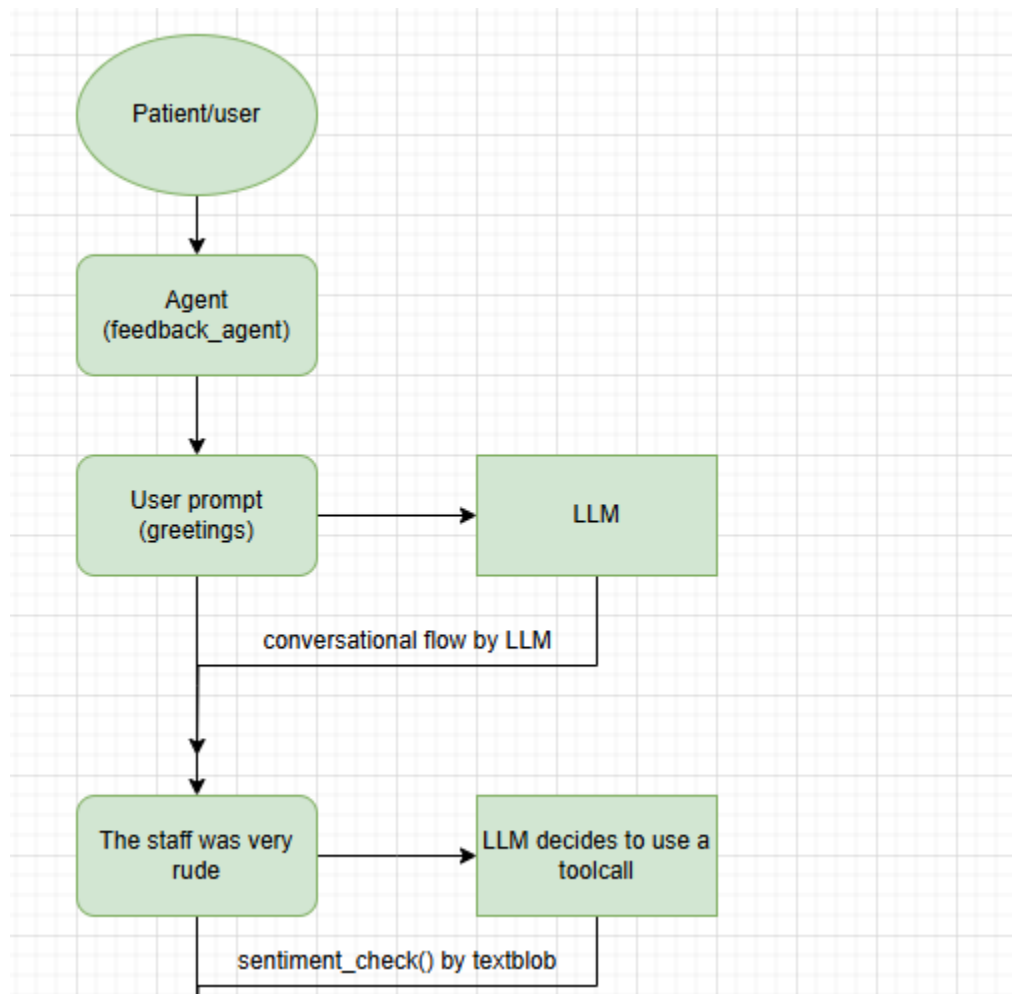
### 6. Feedback Analysis Tools

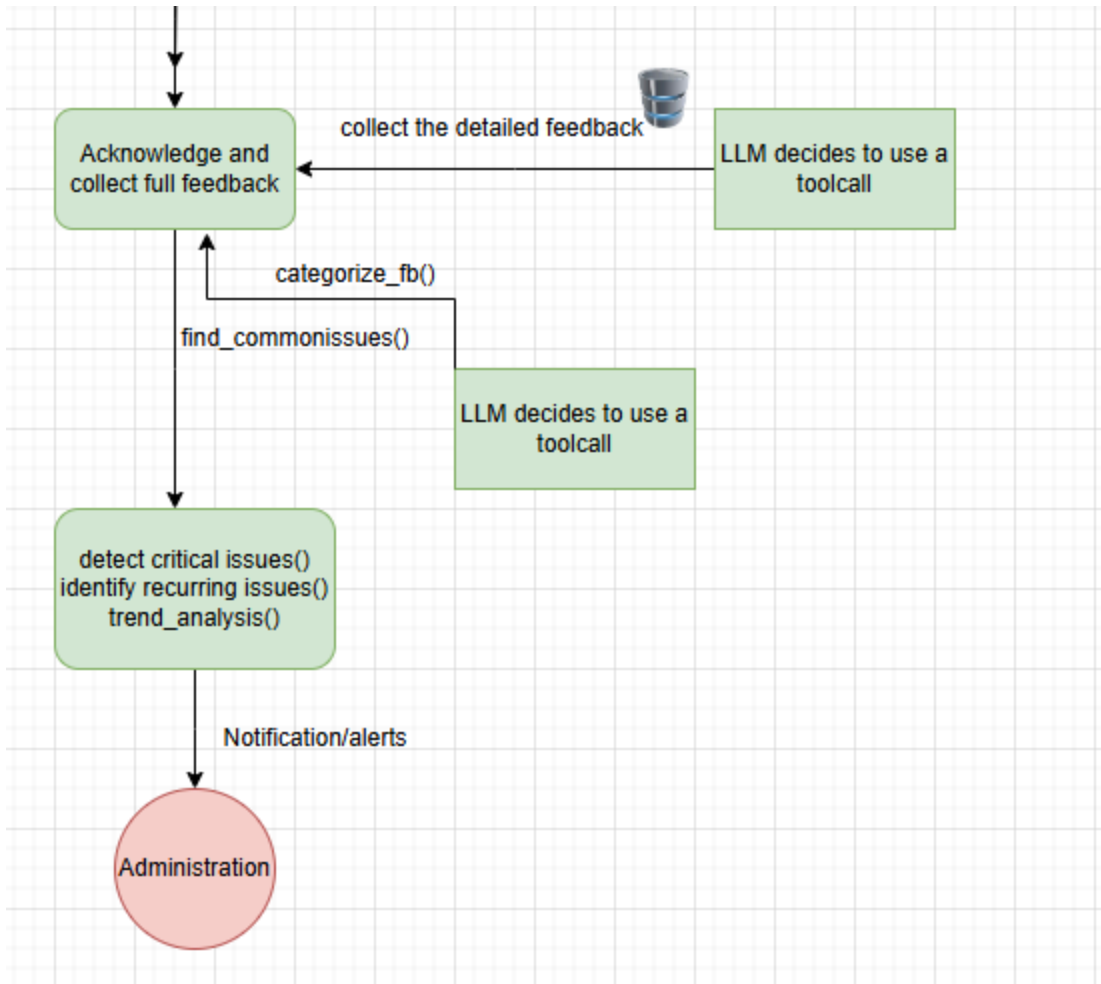
- `find_common_issues`: Analyzes DB for high-frequency complaints
- `generate_trend_analysis`: Calculates average ratings and other info.
- `detect_critical_issues`: Flags dangerous medical keywords

### 7. "New Chat" Flow

- Clears context and agent memory using a `new_chat` message
- Ensures a fresh conversation every time without confusion

### Flow Diagram:





## Features In Progress / To Be Implemented

### 1. Chat History System

- Save each message (user + bot) in a `chat_sessions` collection
- Each chat will have:
  - `session_id, timestamp, messages[ ], nhs_number`
- Chat list will populate the left sidebar in UI
- Allow users to revisit past chats

### 2. Scheduled Reporting

- Use `schedule` or `APScheduler` to:
  - Send trend analysis daily
  - Alert admins if critical issue keywords spike

