**Swiggy GenAI Intern Project Submission - Parthiban K**

**Github Repository: https://github.com/PARTHIBAN-007/Swiggy-Intern**

**Project Objective**

The objective of this project is to develop a **Feedback Management System** that allows users to submit feedback, automatically summarize it using AI, categorize it into different domains, and store it in a database. The system also provides functionalities to retrieve, analyze, and manage feedback efficiently.

**Tech Stack**

* **Backend:** FastAPI, SQLite, LangChain, HuggingFace API, Pydantic
* **Frontend:** React, Axios, Tailwind CSS
* **Database:** SQLite
* **AI Model:** DeepSeek AI (HuggingFace)

**Functionalities**

1. **Submit Feedback:** Users can enter feedback manually or via voice input.
2. **Summarization & Classification:** AI generates a summary, assigns a rating, and categorizes feedback.
3. **Storage in Database:** Feedback is stored with its summary, rating, and domain.
4. **Fetch Feedback Statistics:** Get a count of feedback per category and the number of resolved feedback.
5. **Retrieve Unresolved Feedback:** View pending feedback grouped by domain.
6. **Resolve Feedback:** Mark feedback as resolved once addressed.

**Working of the Project**

1. **User Submits Feedback**
   * Users can either type feedback or use voice input to provide their feedback.
   * The feedback is sent to the backend API for processing.
2. **AI-based Summarization & Categorization**
   * The backend uses the HuggingFace AI model to generate a summary, assign a rating, and categorize the feedback into predefined domains.
3. **Storage in Database**
   * The summarized feedback, along with its rating and domain, is stored in an SQLite database.
4. **Retrieval and Analysis**
   * Users can view statistics about feedback, including the count per domain and the number of resolved feedback.
   * Unresolved feedback can be retrieved per category to help in managing customer concerns effectively.
5. **Feedback Resolution**
   * Feedback entries can be marked as resolved once they have been addressed.

**Setup and Run Instructions**

**Prerequisites**

* Python 3.8+
* Node.js 14+
* SQLite
* HuggingFace API Key (for AI processing)

**Backend Setup**

1. Clone the repository:
2. git clone https://github.com/PARTHIBAN-007/Swiggy-Intern

cd Swiggy-Intern

1. Create a virtual environment and activate it:
2. python -m venv venv

source venv/bin/activate # On Windows use 'venv\Scripts\activate'

1. Install dependencies:

pip install -r requirements.txt

1. Set up environment variables in a .env file:

HUGGINGFACEHUB\_API\_TOKEN=your\_huggingface\_api\_key

1. Run the backend server:

uvicorn main:app --reload

The backend will run at http://localhost:8000

**Database Setup**

1. Initialize the database and populate it with sample data:

python db.py

**Frontend Setup**

1. Navigate to the frontend directory:

cd frontend

1. Install dependencies:

npm install

1. Start the React app:

npm start

The frontend will run at http://localhost:3000

**API Endpoints**

* POST /summarize - Generate a summary, rating, and category for feedback.
* POST /final-feedback - Store feedback in the database.
* GET /feedback-stats - Retrieve statistics on stored feedback.
* GET /unresolved-feedback/{domain} - Fetch unresolved feedback by domain.
* PUT /resolve-feedback/{feedback\_id} - Mark a feedback entry as resolved.

**Notes**

* Ensure that the backend is running before using the frontend.
* The AI-based summarization requires a valid HuggingFace API token.
* CORS is enabled for http://localhost:3000 for development.