Post Conversation Analysis - README

Overview

This project is a Django-based REST API for uploading and analyzing chat conversations. It uses Celery and Redis for background processing and automated daily analysis tasks. The system stores chat data in a database and provides structured insights on conversations.

Setup Instructions

1. Clone the repository: git clone https://github.com//post-conversation-analysis.git
2. Navigate to the project folder: cd post-conversation-analysis
3. Create and activate a virtual environment: python -m venv myenv && myenv\Scripts\activate
4. Install dependencies: pip install -r requirements.txt
5. Apply migrations: python manage.py migrate
6. Run the development server: python manage.py runserver

Celery and Redis Setup

1. Install Docker and run Redis container: docker run -d -p 6379:6379 redis
2. Start Celery worker: celery -A post_conversation_analysis worker -l info
3. Start Celery Beat scheduler: celery -A post_conversation_analysis beat -l info

API Endpoints

1. POST /api/conversations/ → Upload a new conversation
2. POST /api/analyse// → Analyze a conversation by ID
3. GET /api/reports/ → View list of analyzed reports

Cron Job Automation

A daily Celery Beat task automatically triggers conversation analysis every 24 hours. This ensures new conversations are analyzed regularly without manual execution.

Project Structure

1. analysis/ → Contains models, serializers, views, and tasks
2. post_conversation_analysis/ → Main Django project settings
3. requirements.txt → All dependencies
4. README.md → Project documentation

Conclusion

This project demonstrates the use of Django REST Framework with Celery for background automation, suitable for scalable backend systems requiring periodic analysis tasks.