Python Developer Assignment

Thank you for applying! We are testing your ability to handle requirements ambiguity, make independent decisions, and communicate proactively. Please complete at least 2-3 tasks from the list below, and create a Pull Request (PR) with your work. If you can do additional tasks or attempt them, that's even better. If you get stuck, push your incomplete code to the PR. There is no time limit, but try to complete the tasks as soon as possible. (4 days).

Have meaningful comments in commits and Pull requests,. Please reach out once you complete fully with PR link in the mail.

Github link :: https://github.com/soulpage/fullstack-assignment

Submission Guideline:: Please create a pull request against the above codebase, don't create a new repo or another fork in your account. Add you email id in you PR comments. Your pull request should reflect here https://github.com/soulpage/fullstack-assignment/pulls

Task 1: Basic Setup and Simple Feature

1. Setup the development environment and run the application.
2. Add a new field to the Conversation model.
3. Implement functionality to automatically generate and store conversation summaries.
4. Update the Django admin page to display the new field.

Task 2: Database Migration and Data Management

6. Migrate the database from SQLite to PostgreSQL.
7. Implement a management command to clean up old conversations.
8. Schedule the cleanup command to run periodically using Django's crontab or Celery.

Task 3: API Development and Integration

8. Develop a new API endpoint to retrieve conversation summaries, ensuring it supports pagination and filtering.

9. Develop a file upload endpoint.
    - Implement a file duplication check to prevent multiple uploads of the same file.
10. Create an endpoint to list uploaded files with metadata.
11. Implement file deletion functionality.

Task 4: Advanced Features

12. Add a new subpage for Retrieval-Augmented Generation (RAG) (Backend implementation).

13. Add a subpage for file uploads and processing (Backend implementation).
14. Implement role-based access control for file upload and management endpoints.
15. Develop a robust logging mechanism to track file uploads, deletions, and accesses.
16. Implement a caching mechanism to improve the performance of frequently accessed conversation summaries.
17. Create a comprehensive set of unit and integration tests for the new endpoints and features.

Please ensure your code is well-documented and includes tests where applicable. If you encounter any ambiguities or need clarifications, communicate proactively. We look forward to seeing your solutions and how you approach these tasks. Good luck!