AI Dispute Resolver – Complete Project Documentation

Abstract

The AI Dispute Resolver is a web-based platform designed to help two parties resolve disputes in a fair, neutral, and unbiased manner using artificial intelligence. The system enables users to file cases, communicate through a limited live chat, and receive AI-generated resolution suggestions. If both parties agree on a resolution, a formal agreement document is generated and made available for download.

Problem Statement

Traditional dispute resolution methods are time-consuming, expensive, and often biased. There is a need for a digital platform that enables quick, fair, and neutral conflict resolution without legal complexity. The AI Dispute Resolver addresses this gap by providing an AI-assisted, transparent, and consent-based solution.

Objectives

- To provide a neutral AI-assisted dispute resolution system.
- To ensure fairness and unbiased decision-making.
- To limit emotional escalation using controlled communication.
- To generate mutually agreed resolutions and formal agreements.
- To automate notifications and documentation.

Technology Stack

Frontend: React – Used for building a responsive and interactive user interface.
Backend: Python – Handles business logic, AI integration, and rule enforcement.
Database: Firestore – Stores real-time chat data, cases, and resolutions.
AI Integration: Kutrim API – Used for generating neutral and unbiased dispute resolutions.
Email Service: SMTP / Email API – Used for sending case-related notifications.
Document Generation: PDF generation library – Used to generate final agreements.

System Architecture

The system follows a three-tier architecture. The React frontend communicates with the Python backend through secure APIs. The backend processes requests, enforces business rules, interacts with Firestore for data storage, and integrates with the Kutrim AI API to generate neutral resolutions.

Functional Workflow

1. A user files a dispute case by providing case details.
2. The respondent receives an email notification and accepts the case.
3. A live chat session opens with a strict limit of 20 messages.
4. After chat closure, the AI analyzes the case and generates five neutral resolutions.

5. Users can regenerate resolutions if required.
6. If both users accept the same resolution, the case is finalized.
7. A formal agreement PDF is generated and made available for download.

Database Design (Firestore)

Firestore uses a document-based structure. Each case document contains subcollections for messages, resolutions, and acceptances. This structure supports real-time updates, scalability, and efficient data organization.

AI Neutrality and Ethics

The AI system is designed to remain neutral and unbiased. It does not assign blame, take sides, or provide legal judgments. The AI only suggests balanced and mutually acceptable resolutions. Human consent is mandatory before finalizing any agreement.

Security Considerations

User authentication is handled securely. API keys such as the Kutrim API key are stored only on the backend and never exposed to the frontend. Firestore security rules restrict unauthorized data access.

Advantages

- Fast and cost-effective dispute resolution.
- Fair and unbiased AI suggestions.
- Real-time communication.
- Automated documentation and notifications.
- Scalable and cloud-based architecture.

Future Enhancements

- Voice and video dispute sessions.
- Multilingual AI support.
- Integration with legal advisory systems.
- Advanced analytics and dispute insights.

Conclusion

The AI Dispute Resolver provides a modern, ethical, and efficient approach to conflict resolution. By combining React, Python, Firestore, and AI technologies, the system delivers a scalable and neutral platform suitable for real-world applications and academic projects.