Content Moderation API with Embedded Social Media App

Project Overview

This project consists of two main components:
1. **Content Moderation API**: Processes and moderates text, image, and video content for inappropriate or sensitive material.
2. **Social Media/Forum-like App**: A user-friendly platform for posting content, integrated with the moderation API for real-time safety compliance.

Features

Content Moderation API

- Text moderation: Filters profanity, hate speech, and spam.
- Image moderation: Detects nudity, violence, or explicit content.
- Video moderation: Analyzes videos frame-by-frame for inappropriate content.
- Real-time moderation results with logs for flagged content.

Social Media App

- User authentication and profile management.
- Content posting with real-time moderation feedback.
- Admin dashboard for reviewing flagged content.
- Notifications for users on guideline violations.

Tech Stack

Backend (API):

- Python: Flask/FastAPI for building the API.
- NLP Tools: Hugging Face transformers, spaCy.
- Image/Video Processing: OpenCV, TensorFlow.
- Database: PostgreSQL or MongoDB for moderation logs.

Frontend (App):

- Framework: React.js or Angular.
- State Management: Redux or Context API.
- Backend: Node.js (if required for additional logic).

Deployment:

- Cloud: AWS, GCP, or Azure.
- Containerization: Docker.
- Orchestration: Kubernetes.

Installation Instructions

Prerequisites

1. Python 3.8+ installed.
2. Node.js and npm installed.
3. Docker installed for containerization.

Step 1: Clone the Repository

1. $ git clone https://github.com/your-repo/content-moderation-app.git
2. $ cd content-moderation-app

Step 2: Setup Backend (API)

1. Navigate to the API directory:$ cd backend
2. Install dependencies:
3. $ pip install -r requirements.txt
4. Run the API locally:$ python app.py
5. Verify the API at http://localhost:5000.

Step 3: Setup Frontend (App)

1. Navigate to the frontend directory:$ cd ../frontend
2. Install dependencies:$ npm install
3. Run the app locally:
4. $ npm start
5. Access the app at http://localhost:3000.

Step 4: Containerization and Deployment

1. Build Docker images for the API and app:$ docker-compose build
2. Run the containers:$ docker-compose up
3. Access the app and API via the provided Docker URL (e.g., http://localhost).

Usage

For Users:

- Sign up and create a profile.
- Post content (text, image, video) and receive moderation feedback instantly.

For Admins:

- Use the dashboard to review flagged content.
- Take actions (approve/reject posts).

Future Enhancements

- Add multilingual support for text moderation.
- Improve AI models for faster processing.
- Enable user-generated reports for further moderation.

Contact

For any queries or support, contact Kamaleswari P at kamaleswarip.ece2022@citchennai.net.