Project Goal:

To create a plugin that automatically evaluates user-uploaded video media for violations of platform policies (nudity, copyright, fraud, inappropriate content requiring blur, etc.) and allows only moderated, compliant content to go live.

I. Core Functionality & Output:

1. **Input:**
    - **Video File:** The plugin must accept video files in various common formats (e.g., MP4, MOV, AVI, WMV).
    - **Plugin Configuration:** The plugin should accept configurations defining sensitivity levels for each type of check (e.g., strict/moderate/lenient for nudity detection), and acceptable threshold for copyright infringement.
    - **Watermark Overlay:** Ability to receive platform watermark to identify a video belong to which platform
2. **Processing & Analysis:** The plugin must perform the following checks on the uploaded video:
    - **Nudity Detection:**
        - **Output:** A score or confidence level indicating the presence and degree of nudity.
        - **Categorization:** Differentiate between types of nudity (e.g., partial, full, suggestive).
        - **Timestamping:** Identify the specific timestamps where nudity is detected.
    - **Copyright Infringement Detection:**
        - **Output:** A score or percentage indicating the likelihood of copyright infringement.
        - **Source Identification (Ideal):** If possible, identify potential sources of the copyrighted material (e.g., song titles, movie clips).
        - **Timestamping:** Mark the timestamps where potentially infringing content appears.
    - **Fraud Detection (Content-Based):**
        - **Output:** A score indicating the likelihood of fraudulent activity based on video content. (This is broad, so define specific fraud types).

        - **Examples:** Spammy links, misleading promotions, fake giveaways.
    - **Blur Detection (For Inappropriate Content):**
        - **Output:** Identification of timestamps and regions within the video that require blurring.
        - **Categorization:** Reasons for blur (e.g., violence, offensive gestures, personally identifiable information).
3. **Auto-Moderation Decision:**
    - **Output:** A clear "Approve" or "Reject" decision for the video based on the analysis results and configured sensitivity levels.
    - **Reasoning:** Provide a clear explanation for the decision. For example: "Rejected: Nudity detected at 0:15-0:20 exceeding the 'moderate' sensitivity threshold." or "Approved: All checks passed."
    - **Actionable Data:** If rejected, provide specific data points (timestamps, scores, categories) that triggered the rejection, so admin can review.
4. **Reporting & Logging:**
    - **Detailed Logs:** Record all analysis results, decisions, and reasoning for each video processed.
    - **Reporting Dashboard (Optional):** A dashboard displaying key metrics like:
        - Total videos processed
        - Approval/Rejection rates
        - Breakdown of rejection reasons
        - Performance metrics of the ML models (e.g., false positive rate, false negative rate)
    - **Alerts:** Ability to configure alerts for specific events (e.g., a sudden spike in rejections for a particular reason).

II. Technical Requirements:

1. **API Interface:** The plugin must expose a well-defined API for integration with the platform. This API should allow for:
    - Uploading videos for analysis
    - Retrieving analysis results and moderation decisions
    - Configuring plugin settings
2. **Scalability:** The plugin should be designed to handle a high volume of video uploads concurrently.
3. **Performance:** The analysis process should be reasonably fast to avoid delays in content publishing. Define acceptable processing times.
4. **Security:** The plugin must be secure and protect user data.

5. **ML Model Updates:** A mechanism for updating the underlying ML models to improve accuracy and adapt to evolving content trends.

III. Success Metrics:

1. **Accuracy:** High accuracy in detecting policy violations (low false positive and false negative rates). Define acceptable error rates.
2. **Efficiency:** Fast processing times.
3. **Scalability:** Ability to handle a large volume of video uploads.
4. **Reduced Manual Moderation:** Significant reduction in the amount of manual moderation required.

Example Workflow:

1. User uploads a video.
2. Platform sends the video to the plugin via the API.
3. Plugin analyzes the video for nudity, copyright infringement, fraud, etc.
4. Plugin generates a moderation decision (Approve/Reject) and a detailed report.
5. Platform receives the decision.
6. If approved, the video is published.
7. If rejected, the video is not published, and the platform displays the rejection reason (if possible) or notifies administrators for review.