IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, November 2026
Impact Factor: 8.2

A Social Communication Platform with Automated Toxicity Detection, Image Moderation and Secure Account Verification

Ayush A S[1], Benjamin Joby[2], Eric Rolly[3], Harigovind T B[4], Nivedita T S[5], Anjitha P[6]
Student, Department of Computer Science and Engineering[1,2,3,4]
Assistant Professor, Department of Computer Science and Engineering[5,6]
Thejus Engineering College, Thrissur, Kerala, India
ayushasofficial25@gmail.com, benjaminjoby07@gmail.com,
ericrolly41@gmail.com, harigovindtb117@gmail.com

Abstract: *Online social platforms still depend largely on user reports to find abusive messages and explicit images, so harm is often seen before it is removed. This paper presents a secure social communication and content-sharing platform that moderates content automatically at the time it is sent. Text and voice messages are screened by an NLP-based toxicity classifier (voice is first converted to text), and the same screening is applied to post comments. An escalating policy issues warnings after the first and second violations and blocks the account after the third. Uploaded images pass through an ML-based nudity detector, and an MTCNN-based face-recognition module finds registered users in uploaded photos and asks each of them to approve or reject the post. Registration is protected with email OTP verification and a duplicate-account check based on the profile picture. The platform is built with Django, MySQL and a Flutter/web front end. We describe the architecture, the moderation policy and the evaluation protocol, and we discuss the privacy and abuse risks of the face-based features.*

Keywords: Toxicity Detection, Content Moderation, NLP, MTCNN, Face Recognition, Nudity Detection, OTP Verification

I. INTRODUCTION

Social platforms let people chat, post and comment in real time, but they also make it easy to send abuse. Harassment and explicit images are frequently found only after another user reports them, and by then the content has already been read. Keyword filters are a common first defence, yet they miss context, sarcasm and spelling variations, and they are easy to evade with small changes in spelling [9]–[11]. Repeat offenders may also face no consistent consequence.

Two further problems are less discussed. First, a person can appear in a photo that someone else uploads and may never learn about it. Second, accounts are easy to create in bulk, which weakens any moderation that is tied to an identity.

This work addresses these problems in one platform. The main contributions of this work are:

- An NLP-based toxicity detection pipeline for text messages, voice messages (through speech-to-text) and post comments.
- An escalating enforcement policy: warnings after the first and second toxic messages and automatic blocking after the third.
- ML-based nudity detection that screens uploaded images before they are shown to other users.
- MTCNN-based face recognition that tells a registered user when their face appears in a post and lets them approve or reject it.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40718

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, November 2026
Impact Factor: 8.2

- Email OTP verification at registration with a profile-picture duplicate check, plus a cyber-cell function for reviewing flagged accounts.

The rest of the paper is organised as follows. Section II surveys related work, Section III describes the methodology, Section IV covers implementation, Section V presents the evaluation protocol, Section VI discusses limitations and ethics, and Section VII concludes.

II. LITERATURE SURVEY

A. Davidson et al. – Automated Hate Speech Detection and the Problem of Offensive Language [9]
The authors collected tweets and labelled them as hate speech, offensive but not hate speech, or neither. Classifiers trained on word n-gram features found it hard to separate hate speech from merely offensive language, and many abusive posts contained no obvious keyword. This shows the limits of word lists for moderation.

B. Wulczyn et al. – Ex Machina: Personal Attacks Seen at Scale [12]
The study trained a classifier on crowd-labelled comments from Wikipedia talk pages and used it to measure personal attacks across millions of comments. It showed that abuse can be detected automatically at a scale that manual review cannot match.

C. Hosseini et al. – Deceiving Google's Perspective API Built for Detecting Toxic Comments [10]
The authors showed that a widely used toxicity API could be misled by small changes such as misspelled words, inserted spaces or punctuation, which lowered the toxicity score of an abusive sentence. Moderation therefore needs models that cope with noisy and deliberately altered text.

D. Gröndahl et al. – All You Need Is "Love": Evading Hate Speech Detection [11]
The authors tested several hate-speech detectors and found that all of them could be evaded with simple changes, such as adding a positive word like "love". They concluded that the training data matters at least as much as the model architecture.

E. Devlin et al. – BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [1]
BERT is pre-trained on large text collections so that each word is represented using the words on both sides of it, and it can then be fine-tuned on a small labelled collection. This makes it suitable for toxicity classification, where the meaning of a word depends on the sentence.

F. Radford et al. – Robust Speech Recognition via Large-Scale Weak Supervision [16]
The Whisper models were trained on a very large amount of labelled audio and transcribe speech in many conditions without extra tuning. Such systems let a voice message be turned into text and checked by the same text classifier.

G. Bedapudi – NudeNet: Neural Nets for Nudity Classification, Detection and Selective Censoring [15]
NudeNet is an open-source project that uses convolutional networks to detect exposed body parts in images and return a confidence score. The score can be compared with a threshold to decide whether an image is explicit.

H. Zhang et al. – Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks [2]
MTCNN detects faces and facial landmarks with a three-stage cascade of convolutional networks, giving aligned face crops that can be passed to a recognition network. It is used in this work for finding every face in an uploaded image.

I. Schroff et al. – FaceNet: A Unified Embedding for Face Recognition and Clustering [3]
FaceNet maps each face image to a compact vector so that images of the same person lie close together. Faces can then be matched by comparing distances or cosine similarity between vectors, which is the approach used for finding registered users in a post.

J. Taigman et al. – DeepFace: Closing the Gap to Human-Level Performance in Face Verification [4]
DeepFace aligns each face and uses a deep network trained on a large set of identities to verify whether two photos show the same person, reaching accuracy close to that of humans on a standard benchmark.

K. Sun et al. – Deep Learning Face Representation by Joint Identification-Verification [7]
DeepID2 trains a network with two goals at once, identifying the person and judging whether two faces match, which produces face features that separate different people more clearly.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, November 2026
Impact Factor: 8.2

L. Deng et al. – ArcFace: Additive Angular Margin Loss for Deep Face Recognition [5]
ArcFace adds an angular margin to the training loss so that embeddings of the same person are packed more tightly and those of different people are pushed further apart.
M. Simonyan and Zisserman – Very Deep Convolutional Networks for Large-Scale Image Recognition [6]
The work showed that increasing the depth of convolutional networks with small filters improves image recognition, and such networks are widely used as general feature extractors.
N. Buolamwini and Gebru – Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification [17]
The study found that commercial face-analysis systems made far more errors on darker-skinned women than on lighter-skinned men. It is a reminder that face-matching thresholds must be chosen with care and tested on varied users.
Existing platforms usually rely on manual reports and basic keyword filters [9]. Few systems combine message toxicity, image moderation, consent over one's own face in other people's photos, and account verification in one design; this is the gap this work targets.

III. METHODOLOGY

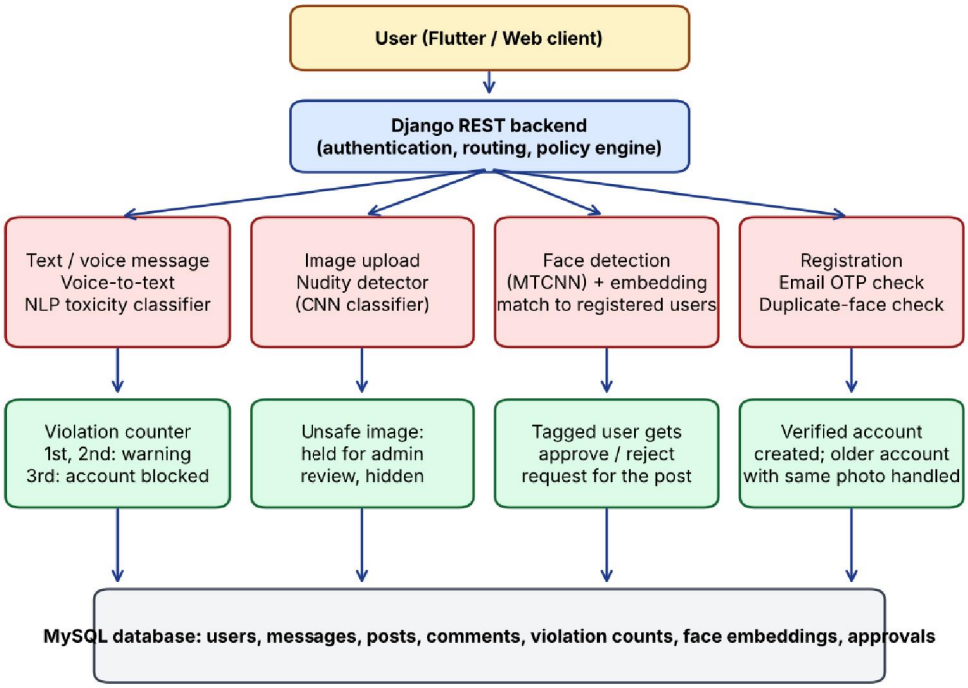


Fig. 1 Architecture of the proposed platform: client, Django backend, four moderation modules, enforcement actions and the MySQL database.

A. System overview

Users interact through a Flutter or web client. Every request goes to a Django backend, which authenticates the user and routes content to one of four modules (Fig. 1): message toxicity, image moderation, face recognition and account verification. Each module returns a decision to a policy layer, which updates the user's record in MySQL and returns the result to the client.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, November 2026 Impact Factor: 8.2

B. Toxicity detection for text, voice and comments

A text message is cleaned (lower-casing, removal of noise, normalisation of repeated characters) and passed to a classifier that outputs a toxicity score s in [0, 1]. The message is treated as toxic when

$$toxic(m) = 1 \text{ if } s(m) \geq \theta, \text{ else } 0 \qquad (1)$$

where θ is a threshold tuned on validation data. A contextual model of the BERT family [1] is used so that the decision depends on the whole sentence and not on single words; a TF-IDF classifier built with scikit-learn [8] can serve as a baseline for comparison. The score is stored with each message and the engine is accessed through an AI API integration. The classifier is a BERT-family model [1], [13] fine-tuned on the Jigsaw toxic-comment collection [14]; it is called through the backend and returns the score directly. The threshold is set to $\theta = 0.5$ by default and is tuned on a validation split so that recall is favoured over precision. A flagged message is recorded as a violation by the sender. For voice messages, the audio is first transcribed to text with a speech-to-text API (for example the Google Web Speech API, or an offline Whisper model [16]), the transcript is stored with the message, and it goes through the same classifier. Comments on posts are checked the same way before they are published.

C. Escalating enforcement policy

Each user has a violation counter v. When a message is flagged, v is incremented and the action is chosen by

$$action(v) = warn \text{ if } v \in \{1, 2\};\ block \text{ if } v \geq 3 \qquad (2)$$

The first and second violations show a warning to the sender, and the third blocks the account automatically. Each violation is also logged as a case (toxicity, nudity or hate comment) with its evidence and score. The cyber-cell officer can view blocked users and cases and mark a case as pending or resolved, and the administrator can block or unblock users. This gives the same consequence to every repeat offender without waiting for reports.

D. Nudity detection

Each image in a new post is passed through an ML-based nudity detection model (a CNN-based detector of the NudeNet type [15]), with a nudity score above a threshold of 0.6 treated as explicit before publication, and the resulting nudity score is stored with the post. If nudity is detected, the post status becomes Pending Admin Approval, so it stays hidden until an administrator approves or rejects it, and the case is logged for the cyber cell. Screening at upload time means other users do not see the image while it is under review, unlike report-based moderation.

E. Face recognition with user-controlled approval

When an image passes the nudity check, MTCNN [2] detects and aligns every face in it. Facial embeddings are extracted from each user's registered profile picture at sign-up and stored with the profile. Each embedding is a 512-dimensional vector produced by a FaceNet-type network [3] from the face cropped by MTCNN. Every face detected in an uploaded image is compared with all stored embeddings using cosine similarity:

$$sim(a, b) = (a \cdot b) / (\|a\| \|b\|) \geq \tau \qquad (3)$$

where τ is the match threshold, set to 0.6 and adjustable by the administrator. If a face matches a registered user, the match score is stored with the post, the post status becomes Pending User Approval, and an alert is sent to the recognised user. The post stays hidden from other users until that user approves it, and the user can also reject it. This gives people control over how their face is used in other people's content.

F. Secure registration

During registration the system emails a one-time password (OTP) to the address entered; the account is created only after the correct OTP is submitted before it expires. The registered profile picture must contain a clear face, because it is used to compute the user's stored embedding; registration fails otherwise. The profile picture of the new account is then compared with existing profile pictures using the face pipeline above, so the platform can detect an attempt to open a second account with the same picture and handle the older account (Section VI discusses this rule).

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, November 2026
Impact Factor: 8.2

G. Roles and data model

The platform has three roles. Registered users chat one-to-one by text or voice, create posts with images, like and comment on posts, edit or delete their own posts and comments, search for other users, and approve or reject posts in which their face is recognised. The administrator views user profiles, toxicity results and face-recognition alerts, blocks or unblocks users, approves or rejects reported posts, and reviews activity logs. The cyber-cell officer views users, blocked users and cases, updates case status and sends feedback to the administrator.

IV. IMPLEMENTATION

The back end is written in Python with Django and stores users, messages, posts, comments, violation counts, face embeddings and approvals in MySQL. The mobile and web clients use Flutter, HTML, CSS and JavaScript. Machine-learning components use TensorFlow and scikit-learn [8]. Table I lists the development environment and Table II the main database tables.

TABLE I: Development Environment

Component	Specification
Processor / RAM	Intel Core i5 or above; 4 GB or above
Storage	500 GB or above
Operating system	Windows 10 or above
Front end	HTML, CSS, Bootstrap, JavaScript, Flutter
Back end / database	Python, Django / MySQL
AI components	MTCNN, NLP classifier, API integration
Tools	PyCharm, VS Code, Android Studio

TABLE II: Main Database Tables

Table	Main fields
USER_PROFILE	Contact details, profile image, face embedding, bio
CHAT	Sender, receiver, message type, text, voice file, transcription, toxicity score, warning count, status
POST	Uploader, recognised user, caption, image, nudity score, face-match score, status
COMMENT	Post, user, text, toxicity score, status (visible/hidden)
LIKE	Post, user, time
CYBERCELL_CASE	Reported user, linked chat/comment/post, case type, evidence, score, status, remarks
NOTIFICATION	User, title, message, type, read flag

V. TESTING AND EVALUATION PLAN

Each module is tested on its own, because the modules solve different tasks. The toxicity and nudity detectors are binary classifiers and will be assessed with accuracy, precision, recall, F1-score and ROC-AUC on a held-out test split (for example 60:20:20 for training, validation and testing). Recall is given most weight, since a missed harmful message reaches the reader. Face matching will be assessed with the true-match and false-match rates at the chosen threshold τ, using public face-verification data such as LFW [18] in addition to registered test users. Response time per message will be measured to confirm that moderation does not delay chat.

The policy and verification logic is tested with scripted cases. Table III lists the cases and the behaviour the system is designed to show for each.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 2, November 2026
Impact Factor: 8.2

TABLE III: Functional Test Cases and Expected Behaviour

Test case	Expected behaviour
First toxic message	Message withheld; warning 1 shown; violation count = 1
Second toxic message	Message withheld; warning 2 shown; count = 2
Third toxic message	Account blocked; case logged for cyber cell
Toxic voice message	Transcript flagged; same warning sequence as text
Toxic comment on a post	Comment hidden; violation counted
Explicit image in a post	Status Pending Admin Approval; hidden from users
Post showing a registered user	Status Pending User Approval; alert sent; visible only after approval
Post rejected by tagged user	Post stays hidden
Correct OTP before expiry	Account created
Wrong or expired OTP	Registration refused
Profile picture with no face	Registration refused

Measured values for the classifiers will be reported in the final version of this work after these tests are run on the implemented system.

VI. DISCUSSION, LIMITATIONS AND ETHICS

Automated moderation makes mistakes. False positives can wrongly warn a user and, after three of them, block an account, so blocked accounts should be reviewable through the cyber-cell function and the threshold θ should favour fairness on slang and dialect. Toxicity models also perform worse on dialects, code-mixed text and low-resource languages, so the training data should be extended for the languages used on the platform.

Face recognition handles biometric data, so embeddings should be stored securely, used only for the approval and duplicate-check features, and covered by a clear user consent notice. Recognition errors could notify the wrong person or miss someone, so τ should be set conservatively; recognition accuracy is known to differ across demographic groups [17].

The rule that deletes an older account when a new verified account uses the same profile picture needs care. Anyone could upload another person's photo and cause that person's account to be removed. A safer design is to suspend rather than delete, to notify the older account's email and require confirmation, and to keep the account recoverable for a period of time. We recommend this refinement for deployment.

VII. CONCLUSION AND FUTURE WORK

The proposed platform combines message toxicity detection, image moderation, user-controlled face approval and OTP-based registration in a single social platform, and enforces a clear warning-then-block policy without waiting for manual reports. Future work includes measuring all modules on larger public datasets, supporting multilingual and code-mixed toxicity, adding an appeal process for blocked users, and replacing automatic account deletion with a verified recovery flow.

VIII. ACKNOWLEDGMENT

The authors thank their guide, Nivedita T S, and their project coordinator, Anjitha P, for their guidance and support, and the Department of Computer Science and Engineering, Thejus Engineering College, for the facilities provided for this work.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in Proc. NAACL-HLT, 2019, pp. 4171–4186.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 2, November 2026 Impact Factor: 8.2

[2] K. Zhang, Z. Zhang, Z. Li and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, 2016.
[3] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proc. IEEE CVPR, 2015, pp. 815–823.
[4] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in Proc. IEEE CVPR, 2014, pp. 1701–1708.
[5] J. Deng, J. Guo, N. Xue and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in Proc. IEEE CVPR, 2019, pp. 4690–4699.
[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. ICLR, 2015.
[7] Y. Sun, X. Wang and X. Tang, "Deep learning face representation by joint identification-verification," in Proc. NIPS, 2014, pp. 1988–1996.
[8] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[9] T. Davidson, D. Warmsley, M. Macy and I. Weber, "Automated hate speech detection and the problem of offensive language," in Proc. ICWSM, 2017, pp. 512–515.
[10] H. Hosseini, S. Kannan, B. Zhang and R. Poovendran, "Deceiving Google's Perspective API built for detecting toxic comments," arXiv:1702.08138, 2017.
[11] T. Gröndahl, L. Pajola, M. Juuti, M. Conti and N. Asokan, "All you need is 'love': Evading hate speech detection," in Proc. ACM Workshop on Artificial Intelligence and Security (AISec), 2018, pp. 2–12.
[12] E. Wulczyn, N. Thain and L. Dixon, "Ex machina: Personal attacks seen at scale," in Proc. Int. World Wide Web Conf. (WWW), 2017, pp. 1391–1399.
[13] A. Vaswani et al., "Attention is all you need," in Proc. NIPS, 2017, pp. 5998–6008.
[14] Jigsaw/Conversation AI, "Toxic Comment Classification Challenge," Kaggle, 2018. [Online]. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
[15] P. Bedapudi, "NudeNet: Neural nets for nudity classification, detection and selective censoring," open-source software, 2019.
[16] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in Proc. ICML, 2023, pp. 28492–28518.
[17] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in Proc. Conf. Fairness, Accountability and Transparency (PMLR 81), 2018, pp. 77–91.
[18] G. B. Huang, M. Ramesh, T. Berg and E. Learned-Miller, "Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments," Univ. Massachusetts Amherst, Tech. Rep. 07-49, 2007