Multilingual Romanized Toxic Comment Inhibition System

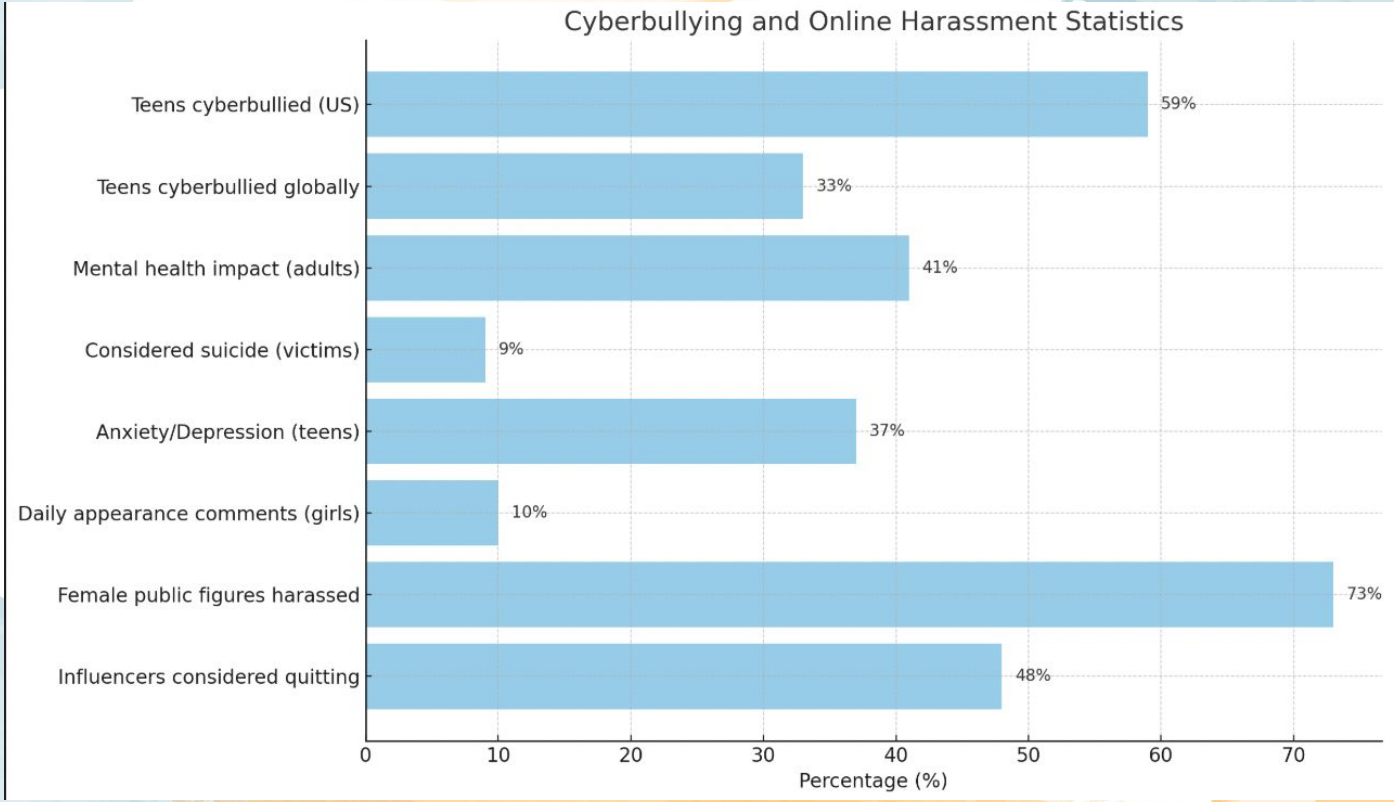
Presented by:

Gopika K (Team Lead)
Dharaneeswari G
Divyabharathi L
Dept. of Computer Science and Engineering
RMK Engineering College

INTRODUCTION

- In today’s hyper-connected world, toxic and mean comments—whether through social media, messaging platforms, or public forums—have become a **growing epidemic.**
- These seemingly "harmless" words can deeply **affect mental health, damage reputations,** and even **end lives.**
- From school children to global celebrities, **no one is immune** to the consequences of online harassment and cyberbullying.
- Despite awareness, the lack of strong moderation, digital empathy, and education continues to fuel this invisible war—**often leaving victims silent and alone.**

STATISTICS


Cyberbullying and Online Harassment Statistics
Teens cyberbullied (US)
Teens cyberbullied globally
Mental health impact (adults)
Considered suicide (victims)
Anxiety/Depression (teens)
Daily appearance comments (girls)
Female public figures harassed
Influencers considered quitting
59%
33%
41%
9%
37%
10%
73%
48%
0
10
20
30
40
50
60
70
Percentage (%)

WHERE THE ACTUAL PROBLEM LIES

- **Romanized Language Complexity -** Difficult to detect toxicity in mixed-language formats like Tanglish, Hinglish due to spelling variations, slang, and lack of standard grammar.
- **Post-Delete Abuse Loophole -** Users can post toxic comments and quickly delete them before moderation kicks in, allowing harmful exchanges to occur.
- **Lack of Real-Time Moderation -** Existing systems work after the comment is posted, not during typing or before submission.
- **Informal and Evolving Language Use -** Toxic content is often masked using creative spellings, emojis, or intentional obfuscation (e.g., "f@ke", "l0ser").
- **Low Accuracy in Low-Resource Languages -** Many local languages lack robust datasets, making training effective models challenging.

DESIGNING THE FIX

We developed a **Machine Learning-based system** that:

- **Analyzes comments in real-time** before they are posted, even in romanized and mixed languages
- **Detects toxic content**, including hate speech, bullying, and offensive language across multiple languages
- **Disables the post button** if the comment is flagged as toxic, preventing harmful content from going live
- **Displays a friendly warning** with community guidelines, encouraging users to rephrase their comment
- **Continuously learns and improves** using NLP techniques and real-world multilingual data

SYSTEM MODULES AND TECHNOLOGY STACK

- **Frontend** - To get the user input as comment and analysing it - **React.js**
- **Backend** - To manage the API requests and sending response back to the frontend - **Flask(Python)**
- **Preprocessing** - Data cleaning, noise removal and normalizing the comment - **Python, Regex**
- **Language Detection** - Detecting the language of the comment - **FastText(ML model), langdetect**
- **Translation & Transliteration** - Converting non-english or romanized text into English - **Google Translation API, indic-transliteration**
- **Toxicity Detection** - Classifying the text into various categories - **BERT**
- **Inhibition Module** - To block the toxic with warnings and allowing to post if non-toxic

WORKFLOW OF THE SYSTEM - FRONTEND

- Users type comments into a text area.
- On comment submission attempts, data is sent to the Flask backend.
- The response includes translated text (if applicable), detected language, and toxicity scores.
- Based on the result:

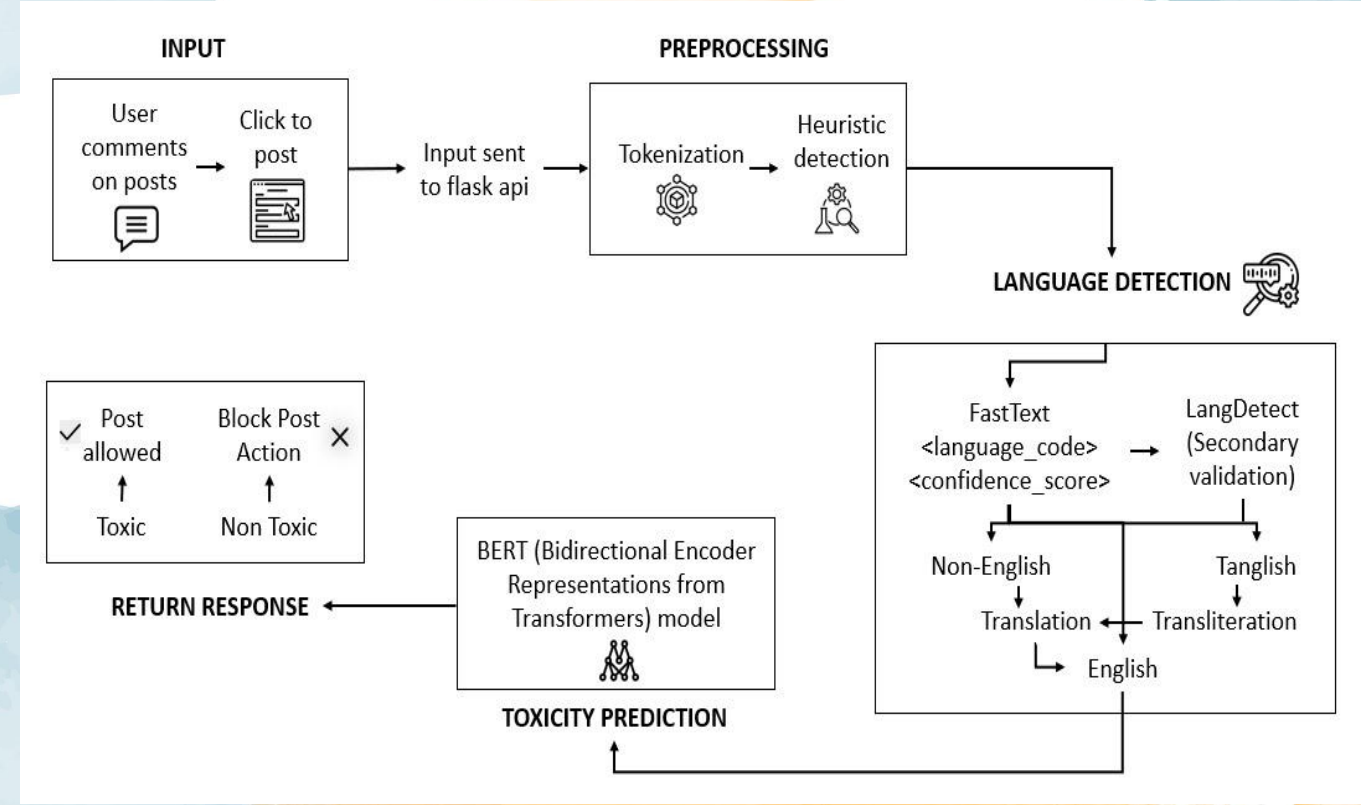
  - If non-toxic → “Allowed” is shown, the post button remains active.
  - If toxic → “Blocked” is displayed, and the post button is disabled, preventing the comment from being posted.

WORKFLOW OF THE SYSTEM - BACKEND

- The Flask backend exposes a REST endpoint that:
- Accepts user comment input.
- Performs preprocessing, language detection, and (if required) transliteration/translation.
- Applies BERT inference for toxicity detection.

- Sends back a structured JSON response containing:
  - Original language
  - Translated English text
  - Toxicity scores ○ Final decision (Allowed / Blocked)

ARCHITECTURE DIAGRAM



CONCLUSION

- This project introduces a smart and responsive comment moderation system that goes beyond detection—it actively **prevents the posting of toxic content in real time**.
- By supporting **romanized and code-mixed languages** across over 100 linguistic variations, the system ensures **inclusive and culturally aware moderation**.
- Leveraging **advanced NLP models** and a **confidence threshold mechanism**, it strikes a balance between **accuracy and user expression**.
- Unlike traditional moderation tools, this solution addresses the **real-world complexity** of informal and multilingual online communication, offering a **practical, scalable, and impactful** tool for creating safer digital spaces.

FUTURE WORK

- **Context-Aware Toxicity Detection**
  Understand conversation history to catch sarcasm, irony, and implicit toxicity.
- **Toxicity Highlighting & Suggestions**
  Highlight toxic words and offer real-time, polite rephrasing using NLP.
- **User Adaptivity**
  Personalize detection thresholds based on user behavior and platform policies.
- **Deployment at Scale**
  Integrate into live platforms with scalable, secure, and load-balanced backend infrastructure.

Thank You for Raising Awareness
Together, we can create a kinder digital world.