# BULLYBLOCK AI – A REALTIME AI BASED DETECTOR AND REPORTER

**A SOCIALLY RELEVANT MINI PROJECT REPORT**

***Submitted by***

**ABINAYA S [REGISTER NO:211423104005] DEEPIKA J [REGISTER NO:211423104110]**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

****

## PANIMALAR ENGINEERING COLLEGE CHENNAI – 600123

**(An Autonomous Institution Affiliated to Anna University, Chennai) OCTOBER 2025**

## PANIMALAR ENGINEERING COLLEGE CHENNAI – 600123

### (An Autonomous Institution Affiliated to Anna University, Chennai) BONAFIDE CERTIFICATE

Certified that this mini project report **“BULLYBLOCK AI – A REALTIME AI BASED CYBERBULLYING DETECTOR AND REPORTER ”** is the bonafide work of ABINAYA S (211423104005) ,DEEPIKA J (211423104110) who carried

out the mini project work under my supervision.

**SIGNATURE Dr.L.JABASHEELA, M.E.,Ph.D., PROFESSOR**

## HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE

PANIMALAR ENGINEERING COLLEGE NASARATHPETTAI,

POONAMALLEE, CHENNAI-600 123.

## SIGNATURE

### Mrs. K. CINTHUJA, M.E. ASSISTANT PROFESSOR SUPERVISOR

DEPARTMENT OF CSE PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE,

CHENNAI- 600 123.

Certified that the above candidates were examined in the End Semester

Submitted for the 23CS1512 - Socially Relevant Mini Project Viva-Voce Examination held on...........................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

# DECLARATION BY THE STUDENT

We ABINAYA.S(211423104005), DEEPIKA.J (211423104110) hereby declare

that this project report titled **BULLYBLOCK AI – A REALTIME AI BASED CYBERBULLYING DETECTOR AND REPORTER**, under the guidance of Mrs. K. CINTHUJA, M.E. is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

1. **ABINAYA S**
2. **DEEPIKA J**

# ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere and hearty thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHIKUMAR,M.E., Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us

with the necessary facilities to undertake this project.We also express our gratitude to our Principal **Dr. K. MANI , M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr.L.JABASHEELA, M.E.,Ph.D.,**

for the support extended throughout the project.

We would like to thank my Project Guide **Mrs. K. CINTHUJA, M.E.** and our project coordinator **Dr.KAVITHA SUBRAMANI, M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

**ABINAYA S DEEPIKA J**

# ABSTRACT

Cyberbullying has become a major concern in the digital era, especially among students and young people, leading to anxiety, depression, and poor academic performance. Existing AI tools typically stop at detecting harmful content, requiring victims to report abuse manually. To address these gaps, BullyBlock AI provides an end-to-end solution that not only detects cyberbullying but also evaluates its severity and automates the reporting process. The proposed system processes online communications using advanced Natural Language Processing (NLP) techniques and pre-trained transformer models such as BERT, integrated with the spaCy framework for efficient text parsing. A custom classifier categorizes the intensity of abuse into Low, Medium, or High levels. When severe or repeated harassment is identified, an automated alert is triggered through SMTP to notify authorities and support teams in real time. Beyond detection, BullyBlock AI incorporates sentiment analysis and interactive data visualization to monitor evolving trends in user interactions. Positive and negative sentiments of flagged messages are tracked over time, offering stakeholders actionable insights into the overall communication climate. The modular design supports easy deployment on web and mobile platforms, making it adaptable for schools, social media monitoring teams, and law-enforcement agencies. This smart, proactive system not only provides early warning and rapid intervention but also establishes a scalable framework for safer digital communities. To ensure reliability and fairness, BullyBlock AI incorporates a robust training and evaluation pipeline. Diverse, publicly available cyberbullying datasets are preprocessed for balanced class distribution and subjected to cross-validation to minimize bias. The model is continuously fine-tuned with fresh data so it can recognize evolving slang, code- switching, and multilingual expressions common in online abuse. The system achieves an overall accuracy of 0.88 (88%).

**TABLE OF CONTENTS**

**CHAPTER TITLE PAGE NO.**

|  |  |
| --- | --- |
| [**ABSTRACT**](#_bookmark0) | [iv](#_bookmark0) |
| **LIST OF FIGURES** | vii |
| 1. **INTRODUCTION**    1. [OVERVIEW](#_bookmark1) | [1](#_bookmark1) |
| 1.2 [PROBLEM DEFINITION](#_bookmark2) | [2](#_bookmark2) |
| 1.3 [SCOPE OF THE PROJECT](#_bookmark3) | [3](#_bookmark3) |
| **2. LITERATURE SURVEY** | 4 |
| 1. **SYSTEM ANALYSIS**    1. EXIISTING SYSTEM | 10 |
| 3.2 [PROPOSED SYSTEM](#_bookmark4) | 11 |
| * 1. [FEASIBILITY STUDY](#_bookmark5)   2. DEVELOPMENT ENVIRONMENT AND TECHNOLOGIES USED | [13](#_bookmark5)  15 |
| **4. SYSTEM DESIGN** |  |
| 4.1 [ARCHITECTURE DIAGRAM](#_bookmark6) | [16](#_bookmark6) |
| 4.2 [USE CASE DIAGRAM](#_bookmark7) | [17](#_bookmark7) |
| 4.3 [CLASS DIAGRAM](#_bookmark8) | [18](#_bookmark8) |
| 4.4 DATA FLOW DIAGRAM | 19 |
| 1. **SYSTEM IMPLEMENTATION**    1. DATA COLLECTION AND | 22 |
| PREPROCESSING |  |
| 5.2 NLP MODELS(BERT) | 24 |

* 1. [MODEL TRAINING AND TESTING 27](#_bookmark9)
  2. [AUTOMATED EMAIL NOTIFICATION 29](#_bookmark10)
  3. [SENTIMENT ANALYSIS 31](#_bookmark11)

1. PERFORMANCE ANALYSIS
   1. EVALUATION METRICS 33
   2. [MODEL COMPARISON 36](#_bookmark12)
   3. [OVERALL PERFORMANCE SUMMARY 38](#_bookmark13)
2. RESULTS AND DISCUSSION
   1. SCREENSHOTS OF BULLYBLOCK AI 40

DASHBOARD

1. [CONCLUSION AND FUTURE SCOPE 45](#_bookmark14)

[APPENDICES](#_bookmark15)

* 1. [SDG GOAL MAPPING 48](#_bookmark16)
  2. [SAMPLE SOURCE CODE 50](#_bookmark17)
  3. PLAGIARISM REPORT 62

[REFERENCES 71](#_bookmark18)

|  |  |  |
| --- | --- | --- |
| **FIG NO** | **LIST OF FIGURES** | **PAGE NO.** |
| **4.1** | ARCHITECTURE DIAGRAM | 16 |
| **4.2** | USE CASE DIAGRAM | 17 |
| **4.3** | CLASS DIAGRAM | 18 |
| **4.4** | DATA FLOW DIAGRAM | 19 |
| **5.2** | NLP MODEL(BERT) | 25 |
| **5.4** | SAMPLE EMAIL NOTIFICATION | 30 |
| **6.1** | ACCURACY | 33 |
| **6.1** | F1-SCORE | 34 |
| **6.1** | CONFUSION MATRIX ANALYSIS | 35 |
| **7.1** | LOGIN PAGE | 40 |
| **7.1** | CREATE ACCOUNT | 41 |
| **7.1** | DASHBOARD | 41 |
| **7.1** | REPORTS | 42 |
| **7.1** | REVIEWS | 42 |
| **7.1** | SENTIMENT ANALYSIS GRAPH | 43 |
| **7.1** | SETTINGS | 44 |

# CHAPTER 1 INTRODUCTION

* 1. **OVERVIEW**

Cyberbullying has become a serious issue, especially among students and young users, causing stress, anxiety, and poor academic performance. Most existing AI tools only detect harmful content, leaving victims to report abuse manually, which can delay intervention. Bully Block AI addresses this problem by providing a complete solution that detects cyberbullying, evaluates severity, and automates reporting. The system collects messages in real time from online platforms and preprocesses the text using tokenization, lemmatization, stop-word removal, and normalization. Advanced NLP models such as BERT and spaCy identify abusive language and classify messages as Low, Medium, or High severity.

High-severity or recurring low-severity incidents trigger automatic email alerts via SMTP to parents, school administrators, or platform moderators. Interactive dashboards visualize severity trends and sentiment distribution, allowing stakeholders to monitor patterns and take timely action.

Bully Block AI is scalable, adaptable to multiple platforms and languages, and provides a proactive approach to safer online communication. By combining detection, severity assessment, automated reporting, and visualization, it ensures prompt support for victims and helps create a responsible digital environment.

Bully Block AI is designed for scalability and adaptability, capable of integrating with multiple online platforms and supporting multiple languages. Its modular architecture allows for future improvements, including the addition of more sophisticated detection models or expanded monitoring capabilities. By combining detection, severity assessment, automated reporting, and data visualization, Bully Block AI provides a robust and proactive approach to cyberbullying prevention, promoting safer online interactions and ensuring that victims receive prompt support and protection.

# PROBLEM DEFINITION

With the rise of social media and online communication platforms, students and young people are increasingly exposed to cyberbullying, which includes harassment, threats, or humiliation through digital channels. Such behaviour can lead to stress, anxiety, depression, and poor academic performance. Unlike traditional bullying, cyberbullying can happen anytime and reach a wide audience, making it harder to detect and control. Most current AI tools focus only on detecting harmful content. While this is important, these tools do not assess the severity of incidents, provide automated reporting, or give administrators insights into communication patterns. Additionally, existing systems rarely analyze overall trends in positive and negative interactions, limiting the ability of stakeholders to monitor online communities effectively.

BullyBlock AI addresses these challenges by continuously monitoring messages on online platforms, preprocessing them, and using NLP models like BERT and spaCy to detect abusive content. Messages are categorized into Low, Medium, or High severity, with high- severity or repeated incidents triggering automatic email notifications to the concerned authorities. Interactive dashboards visualize sentiment and severity trends, helping stakeholders monitor and manage communication patterns effectively.

By combining detection, severity assessment, automated reporting, and visualization, Bully Block AI provides a proactive and comprehensive solution that ensures prompt support for victims and helps create safer, more responsible online environments. Its interactive dashboards and sentiment analysis tools allow stakeholders to understand community behaviour over time, facilitating data-driven decision-making.

# SCOPE OF THE PROJECT

The scope of BullyBlock AI defines the boundaries and objectives of the system, highlighting what it aims to achieve and the areas it will cover:

Real-Time Cyberbullying Detection: The system continuously monitors messages, comments, and other text-based communications on online platforms to identify instances of cyberbullying as they occur.

Severity Assessment: BullyBlock AI classifies detected messages into Low, Medium, or High severity categories, enabling administrators to prioritize responses according to the seriousness of each case.

Automated Reporting: For high-severity or recurring low-severity incidents, the system generates automatic email notifications to the concerned authorities, such as parents, school administrators, or platform moderators, ensuring timely intervention.

Sentiment Analysis and Visualization: The platform analyzes positive and negative interactions, displaying the results through interactive dashboards and graphs. This allows stakeholders to monitor trends, evaluate the overall communication climate, and make informed decisions.

Scalability and Adaptability: The system is designed to support multiple online platforms and languages, making it suitable for deployment in diverse environments and communities.

Data Privacy and Security: The scope includes safe storage of user data and compliance with standard privacy practices, ensuring that sensitive information is protected.

By defining these areas, the project ensures a focused approach to creating a proactive, automated, and intelligent cyberbullying detection system that improves online safety and promotes responsible digital behaviour.

# CHAPTER 2 LITERATURE SURVEY

In the era of digital communication, social media platforms have become a primary medium for students and young people to interact. While these platforms facilitate communication and learning, they also create opportunities for cyberbullying, which is the intentional use of digital communication to harass, threaten, or humiliate individuals. Cyberbullying can have serious psychological and social consequences, including anxiety, depression, low self-esteem, and decreased academic performance. Unlike traditional bullying, cyberbullying can occur 24/7, reach a wider audience, and remain permanent online, making timely detection and intervention more challenging.

Kaur and Saini (2023) developed an AI-based model aimed at detecting cyberbullying and cyberhate across various social media platforms. Their approach utilized deep learning techniques to analyze and classify user-generated content, identifying instances of online abuse and harmful behavior. The model was designed to handle the vast and diverse nature of social media data, ensuring scalability and adaptability across different platforms. By focusing on the nuances of language and context within online interactions, Kaur and Saini's work contributed to enhancing the effectiveness of automated moderation systems in combating cyberbullying and promoting safer online environments.

Ambareen (2023) conducted a comprehensive survey examining the application of artificial intelligence in cyberbullying detection across social media platforms. The study reviewed various AI techniques, including machine learning and natural language processing methods, assessing their performance in identifying and mitigating instances of cyberbullying. Ambareen's research highlighted the challenges and limitations of current AI models.

Usmaan Ali (2024) applied traditional machine learning models, such as Support Vector Machines (SVM) and Random Forest, to detect cyberbullying patterns on social media platforms. His research focused on the preprocessing of data and feature extraction techniques to enhance the accuracy of the models. By analyzing various features within user-generated content, such as linguistic cues and metadata, Ali's work contributed to refining the capabilities of machine learning algorithms in identifying subtle instances of cyberbullying. The study underscored the significance of data quality and feature selection in developing effective cyberbullying detection systems.

Balakrishnan and Kiaty (2023) conducted a systematic literature review on machine learning-based approaches for cyberbullying detection. Their research analyzed various studies, comparing different algorithms and methodologies employed in identifying cyberbullying across social media platforms. Balakrishnan and Kiaty's review highlighted the effectiveness of hybrid models that combine multiple techniques, such as text analysis and sentiment classification, in improving detection accuracy. The study provided valuable insights into the strengths and weaknesses of existing approaches, offering guidance for future research in the field of cyberbullying detection.

Jamalpur et al. (2023) designed a machine learning-based system to recognize social harassment in online interactions. The system classified user comments as bullying or non- bullying by employing effective preprocessing and feature selection techniques. Their approach aimed to address the challenges associated with detecting subtle forms of harassment, such as microaggressions and indirect insults, which are often overlooked by traditional detection systems. Jamalpur et al.'s work contributed to broadening the scope of cyberbullying detection by incorporating a wider range of abusive

Milosevic et al. (2023) evaluated AI-driven moderation tools from the perspective of young users, focusing on their effectiveness in detecting and mitigating cyberbullying. The study emphasized the importance of transparency, fairness, and user trust in automated systems. Milosevic et al. found that while AI tools can be effective in identifying overt instances of cyberbullying, their success is contingent upon user perceptions and the perceived fairness of the moderation process. The research highlighted the need for continuous user feedback and iterative improvements to enhance the acceptance and efficacy of AI-based moderation tools.

Verma et al. (2023) investigated the transparency and reliability of AI-based moderation systems on online platforms. Their study examined the mechanisms through which AI tools detect and address instances of cyberbullying, assessing their effectiveness and the clarity of their decision-making processes. Verma et al. found that while AI systems can process large volumes of data efficiently, issues related to explainability and accountability can undermine user trust. The research underscored the necessity for developing AI models that provide clear rationales for their decisions, ensuring that users understand the basis for content moderation actions.

Lahby et al. (2024) explored the application of artificial intelligence in combating cyberbullying within digital media. Their work focused on leveraging AI techniques for content moderation and online safety, providing a comprehensive overview of strategies to counter digital harassment. Lahby et al. discussed various AI methodologies, including machine learning and natural language processing, highlighting their potential in identifying and mitigating instances of cyberbullying. The study emphasized the importance of integrating AI tools with human oversight to create a balanced approach to online moderation.

Dr. Promise A. Nlerum and Beauty Brisibe (2024) proposed a hybrid model combining deep learning and natural language processing-based sentiment classification to enhance the robustness and accuracy of cyberbullying detection. Their approach aimed to address the challenges associated with detecting nuanced forms of cyberbullying, such as sarcasm and context-dependent insults, which often evade traditional detection systems. By integrating multiple AI techniques, Nlerum and Brisibe's model sought to improve the sensitivity and specificity of cyberbullying detection, contributing to more effective moderation of online platforms.

Tabia Tanzin Prama, Jannatul Ferdaws Amrin, Md. Mushfique Anwar, and Iqbal H. Sarker (2025) introduced an explainable AI system designed to detect cyberbullying severity on social media platforms. The system provided interpretable results, allowing moderators and users to understand the rationale behind detection decisions. Prama et al.'s work emphasized the importance of transparency in AI-driven moderation tools, aiming to build trust and ensure accountability in automated systems. The study highlighted the potential of explainable AI in enhancing user engagement and compliance with community guidelines.

Hasibul Hamim, KMM Uddin, MNT Mim, and R (2024) applied deep learning architectures for large-scale cyberbullying detection, demonstrating the model's effectiveness in handling multilingual and complex data. Their research focused on developing models capable of processing diverse linguistic inputs, addressing the challenges posed by the global nature of social media interactions. Hamim et al.'s work contributed to expanding the applicability of cyberbullying detection systems across different languages and cultural contexts, promoting inclusivity in online safety efforts.

Andrea Perera and Pumudu A. Fernando (2024) presented an integrated detection and prevention model using supervised learning to create safer online interactions. Their approach combined elements of cyberbullying detection with proactive measures to prevent abusive behavior before it escalates. Perera and Fernando's model aimed to foster a more positive online environment by addressing cyberbullying through both reactive and preventive strategies. The study underscored the importance of a holistic approach to online safety, integrating detection, prevention, and education.

Al-Harigy et al. (2022) compared various machine learning algorithms for cyberbullying detection, highlighting the potential of AI in building fully automated bullying detection pipelines. Their research assessed the performance of different algorithms, providing insights into their strengths and limitations in the context of cyberbullying detection. Al- Harigy et al.'s work contributed to the ongoing efforts to develop efficient and scalable systems for identifying and mitigating cyberbullying across digital platforms.

Mainka Saharan (2022) introduced an adaptable machine learning framework for real-time cyberbullying detection on social media platforms. The framework was designed to dynamically adjust to evolving language patterns and emerging forms of online abuse, ensuring continuous effectiveness in detecting cyberbullying. Saharan's approach emphasized the need for flexibility and responsiveness in AI-driven moderation tools, aiming to keep pace with the rapidly changing landscape of social media interactions.

Ninad Mehendale, Keval Rajpara, Karan Shah, and Chaitanya Phadtare (2023) reviewed machine learning algorithms for cyberbullying detection, analyzing their accuracy, limitations, and scope. Their comprehensive review provided a comparative analysis of various algorithms, offering valuable insights into their applicability and performance in detecting cyberbullying across different contexts. Mehendale et al.'s work served as

resource for researchers and practitioners seeking to understand the strengths and weaknesses of existing machine learning approaches in the realm of cyberbullying detection.

Srinadh Unnava and Sankara Rao Parasana (2024) proposed a supervised machine learning model that classifies social media posts using linguistic and contextual cues to identify cyberbullying. Their model aimed to capture the subtleties of language and context within online interactions, improving the accuracy of cyberbullying detection systems. Unnava and Parasana's approach highlighted the importance of considering both linguistic features and contextual information in developing effective moderation tools for social media platforms.

Abulkarim Faraj Alqahtani and Mohammad Ilyas (2024) compared multiple machine learning classifiers for cyberbullying detection, identifying ensemble learning as the most effective approach. Their research demonstrated that combining multiple classifiers could enhance detection accuracy and robustness, addressing the challenges posed by the diverse nature of cyberbullying. Alqahtani and Ilyas's work contributed to refining machine learning techniques in the field of cyberbullying detection, promoting more reliable and efficient systems.

# CHAPTER 3 SYSTEM ANALYSIS

* 1. **EXISTING SYSTEM**

The current landscape of cyberbullying and cyberhate detection primarily relies on artificial intelligence and machine learning models to identify harmful online content. The IEEE Access paper “Role of Artificial Intelligence in Cyberbullying and Cyberhate Detection” (2023) by Manpreet Kaur and Munish Saini provides a comprehensive view of this approach. Existing systems focus on scanning social media posts, chat messages, and other digital communications using Natural Language Processing (NLP) techniques to classify text as abusive or non-abusive.

Key methods involve supervised machine learning algorithms and deep-learning architectures that learn linguistic patterns of harassment from large labeled datasets. Many solutions employ transformers such as BERT for contextual understanding, enabling more accurate detection of subtle cyberbullying indicators like sarcasm, slang, and code-mixed language. These models often integrate with platform APIs to monitor content streams in real time.

However, the paper highlights that most of these systems stop at detection. They identify harmful content but lack an automated mechanism to assess severity or trigger timely intervention. Reporting abusive behavior usually requires manual action by victims or moderators, which delays response and limits immediate support. Contextual factors— such as frequency of attacks, relationship between participants, or escalating aggression— are seldom analyzed in depth.

Despite these limitations, existing AI-driven detection frameworks have demonstrated significant impact. This body of work establishes a proven baseline for building more advanced solutions, including systems such as BullyBlock AI.

# PROPOSED SYSTEM

The proposed system, BullyBlock AI, is designed as a complete cyberbullying detection and intervention platform that goes well beyond simple keyword filtering. Instead of merely searching for a fixed list of offensive words, it uses advanced Natural Language Processing to understand the meaning and context of each message. This allows the system to measure the severity of abuse in real time, determining whether a message is mildly rude, strongly abusive, or openly threatening. When high-risk or repeated harassment is detected, BullyBlock AI can automatically alert the appropriate authorities, ensuring that victims receive prompt assistance. In addition, it performs sentiment analysis to track positive and negative trends across conversations and presents these insights through interactive visualizations.

### System Objectives:

The main objectives of the proposed system are:

Accurate Detection: Identify cyberbullying messages with high precision by understanding the surrounding context, slang, and multilingual expressions.

Severity Classification: Automatically grade harmful content as Low, Medium, or High to prioritize intervention.

Automated Reporting: Notify responsible authorities—such as teachers, counselors, or platform moderators—through secure email alerts (SMTP) when high-severity or repeated abuse is detected.

Sentiment Monitoring: Track and visualize the overall sentiment of flagged interactions so stakeholders can recognize trends and act proactively.

### System Architecture

BullyBlock AI follows a modular, layered design that supports scalability and privacy:

Data Acquisition Layer – Collects text data from chat platforms, social media feeds, or institutional forums using approved APIs.

Pre-processing Layer – Cleans and normalizes input by removing noise, tokenizing text, and handling code-mixed or multilingual content.

Detection & Severity Engine – Core Natural Language Processing (NLP) component powered by transformer models such as spaCy and enhanced with BERT pipelines.

Reporting & Notification Layer – When severity is High or repeated incidents occur, the system triggers secure email alerts through SMTP and records the incident in an audit log for further investigation.

Visualization & Dashboard Layer – Generates interactive charts displaying positive versus negative sentiment, frequency of incidents, and trend analyses for administrators and counselors.

### Functional Modules Detection Module:

Detection Module: Uses BERT embeddings with fine-tuned classifiers to detect offensive or bullying language.

Severity Analysis Module: Applies rule-based and statistical metrics—such as repeated insults, hate speech intensity—to classify abuse as Low, Medium, or High.

Reporting & Notification Module: Automatically sends structured alerts to designated authorities whenever High-severity or repeated abuse is found.

Sentiment Review & Visualization Module: Segregates positive and negative reactions to flagged content and displays them in interactive graphs, offering a clear view of the communication climate over time.

# FEASIBILITY STUDY

### Technical Feasibility

BullyBlock AI leverages advanced Natural Language Processing (NLP) techniques and frameworks like BERT and spaCy for accurate detection of cyberbullying. These technologies are well-established and have extensive support for text classification, sentiment analysis, and contextual understanding. The system integrates automated reporting via SMTP, allowing it to notify authorities without manual intervention, which is technically achievable using existing Python libraries. Visualization of sentiment trends using interactive graphs is also feasible with frameworks like Matplotlib, Plotly, or Dash. The combination of these tools ensures the platform can be developed using current hardware and software technologies without requiring cutting-edge infrastructure.

### Operational Feasibility

The proposed system is designed to be user-friendly for both end-users (victims) and stakeholders (administrators, school authorities, or social media moderators). By automatically classifying the severity of cyberbullying into Low, Medium, and High categories, and sending immediate notifications for serious cases, it reduces the burden on users to report abuse manually. The sentiment analysis and visualization modules provide actionable insights into communication trends, enabling organizations to respond proactively. The operational workflow aligns with existing communication systems (emails, dashboards), making implementation practical and sustainable.

### Economic Feasibility

BullyBlock AI primarily relies on open-source frameworks (BERT, spaCy, Python libraries for email automation and visualization), which significantly reduces software costs. Hardware requirements are moderate; a standard server or cloud service can

support the NLP processing.

manpower costs for monitoring online interactions. Overall, the investment in development, deployment, and maintenance is justified by the value of preventing harm, reducing cyberbullying-related incidents, and safeguarding user well-being, which could otherwise result in social and academic consequences.

### Legal and Ethical Feasibility

The system processes textual communication data, so user privacy and data protection are important considerations. By implementing secure data storage, anonymization techniques, and compliance with regulations such as GDPR or local data protection laws, BullyBlock AI can operate legally and ethically. Automated reporting is designed to notify authorities only for severe or repeated incidents, minimizing unnecessary exposure of sensitive user information.

### Schedule Feasibility

Development of BullyBlock AI can be completed in a short to medium-term schedule due to the availability of ready-to-use NLP models, Python libraries for automation, and visualization tools. Core functionalities such as detection, classification, and reporting can be implemented in the initial phase, followed by sentiment analysis dashboards and interactive visualization in later phases. Testing and deployment can be carried out incrementally to ensure reliability and scalability.

## DEVELOPMENT ENVIRONEMNT AND TECHNOLOGIES USED

The development of BullyBlock AI is carried out using a carefully selected environment designed to ensure accurate cyberbullying detection, automated reporting, and sentiment visualization. A robust development environment is critical for AI-based projects as it supports the processing of large text datasets, model training, and deployment of interactive dashboards while maintaining scalability and maintainability.

### Development Environment

IDE: VS Code is used as the primary development platform for coding, debugging, and project management due to its lightweight nature and extensive support for Python development.

Programming Language: Python is chosen for its versatility and rich ecosystem of libraries for NLP, machine learning, and automation.

### NLP Models and Tools

BERT (Bidirectional Encoder Representations from Transformers): Utilized as the primary NLP model due to its high accuracy in understanding context and detecting cyberbullying. spaCy: Explored for text preprocessing and entity recognition, but BERT was preferred for superior performance in handling nuanced abusive content.

### Datasets

Kaggle Cyberbullying Dataset: Contains labeled social media messages used for model training and validation.

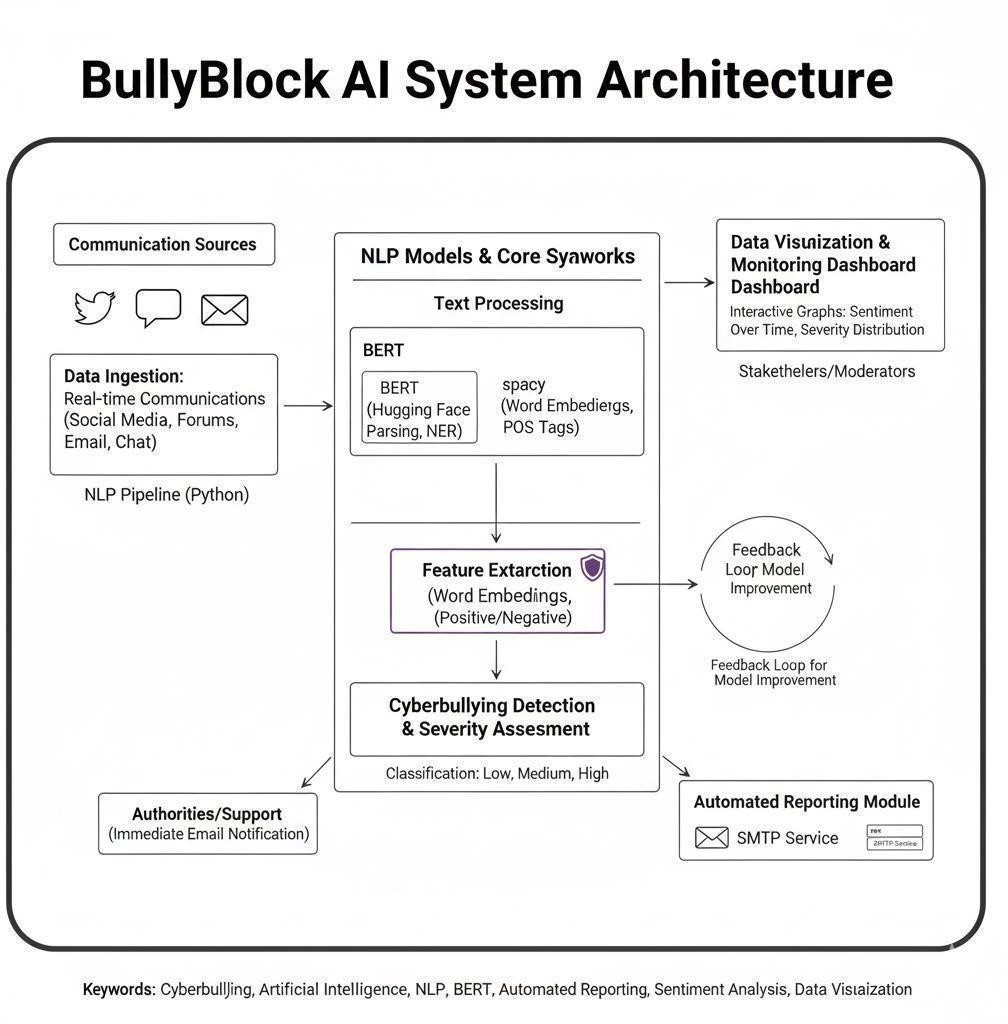
Twitter Hate Speech Dataset: Provides real-world examples of abusive and harmful content, ensuring the system generalizes well to online communications.

# CHAPTER 4 SYSTEM DESIGN

* 1. **ARCHITECTURE DIAGRAM**

The BullyBlock AI system is designed to automatically detect, assess, and report cyberbullying. It starts by taking real-time online communications and using an NLP pipeline with BERT and spaCy to understand the text.The system then classifies the severity of the bullying as Low, Medium, or High and also performs sentiment analysis. For severe cases, it uses an Automated Reporting Module to immediately send an email notification to authorities via an SMTP service.

All the data is stored and displayed on a Monitoring Dashboard with interactive graphs that show sentiment and severity trends over time. A feedback loop allows for continuous improvement of the AI models based on user reviews.



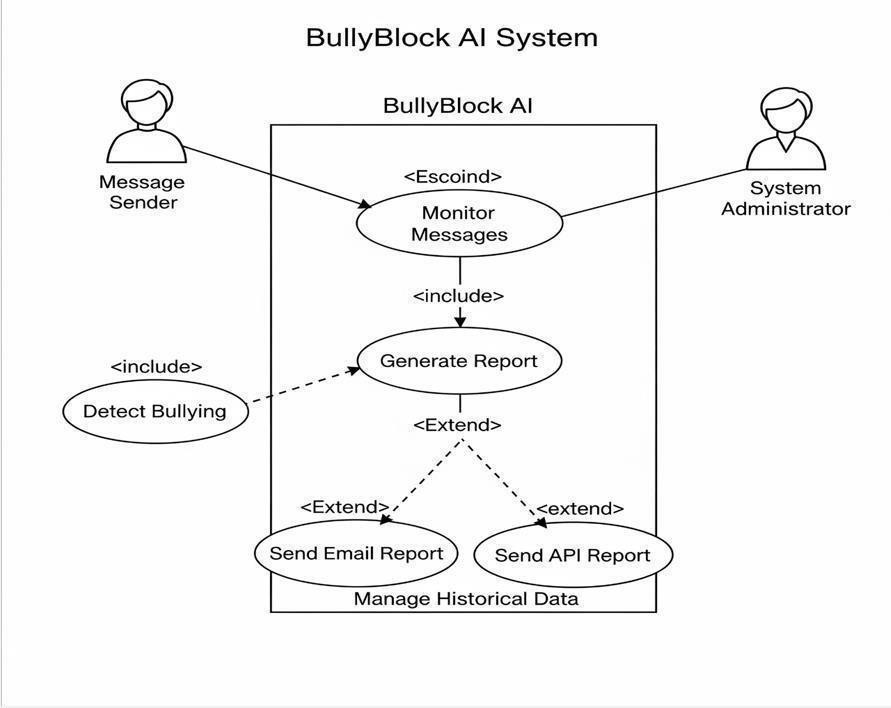
**Fig. 4.1 ARCHITECTURE DIAGRAM**

# USE CASE DIAGRAM

This use case diagram for the BullyBlock AI System shows two main actors: a Message Sender and a System Administrator.

The core process starts when a Message Sender sends a message, which triggers the system to Monitor Messages. As part of this, the system must Detect Bullying. If bullying is found, the system will Generate a Report.

From here, the system can optionally Send an Email Report or Send an API Report, depending on the configuration. The System Administrator is responsible for managing the overall process and can also Manage Historical Data.



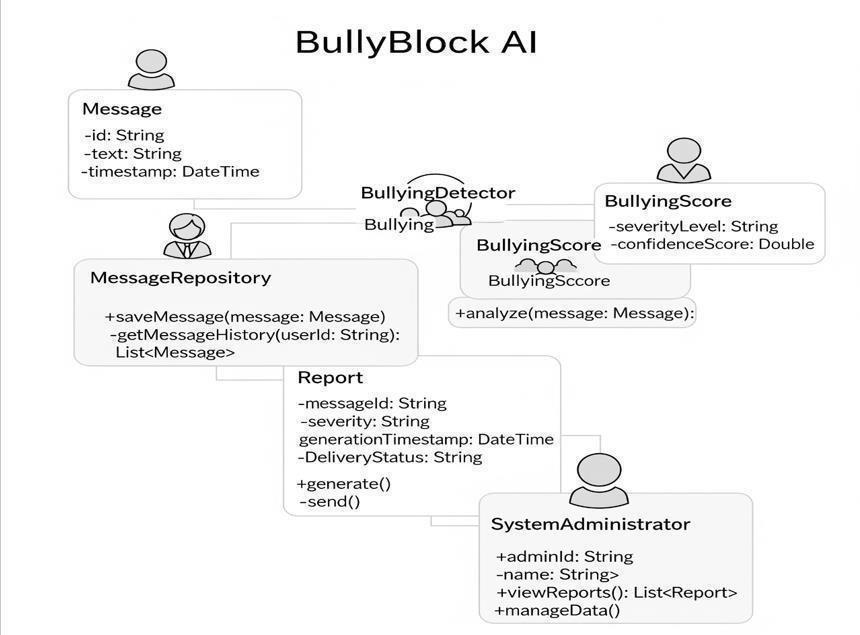
**Fig. 4.2 USE CASE DIAGRAM**

# CLASS DIAGRAM

The BullyBlock AI system uses a Message object, which is stored by the Message Repository

The Bullying Detector analyzes the message and creates a Bullying Score object. If the score is high, a Report is generated and sent.

A System Administrator can then review these reports and manage the data



**Fig.4.3 CLASS DIAGRAM**

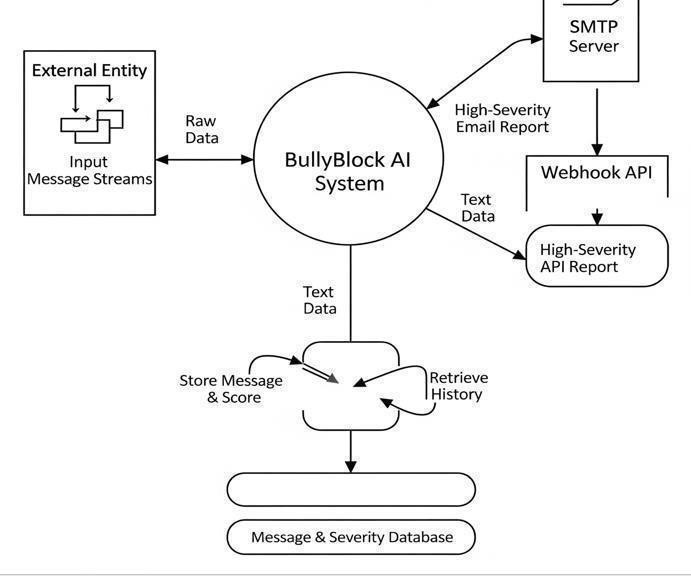
# DATA FLOW DIAGRAM

Input & Message Processing: Receives and formats incoming messages from various sources.

NLP & Analysis: The core AI that detects bullying and assigns a severity score (Low, Medium, High).

Data Management: A central database that stores all messages, scores, and report history. Reporting & Notification : Automatically generates and sends reports to authorities for high-severity cases via email and APIs.

Admin Interface : A control panel for administrators to view reports and manage the system.



### Fig. 4.4 DATA FLOW DIAGRAM

The data flow in the BullyBlock AI system can be broken down into a series of logical steps, from the initial input of a message to the final output of a report or data visualization.

Step 1: Message Ingestion

* Source: A Message Sender (a user on a platform like social media or a forum) sends an online message.
* Process: The Message Ingestion process captures this message.
* Data Flow: The data, in its raw form as Raw Text Data, enters the system. Step 2: NLP Pipeline Processing
* Source: The Raw Text Data from the Message Ingestion process.
* Process: The data is passed through the NLP Pipeline. This is a critical step where the text is cleaned, tokenized, and prepared for analysis using techniques and frameworks like BERT and spaCy.
* Data Flow: The pipeline outputs Processed Data, which is now structured and ready for the next stage.

Step 3: Analysis and Classification

* Source: The Processed Data from the NLP Pipeline.
* Process: This stage involves the core AI and machine learning models. The system performs two key functions:
  1. Bullying Detection: It analyzes the text to determine if it contains bullying content.
  2. Severity Classification: If bullying is detected, it is classified into one of three levels: Low, Medium, or High.
* Data Flow: This process generates key data, including the Severity Level and other Report Data needed for a formal report.

Step 4: Data Storage and Logging

* Source: The Severity Level & Report Data from the Analysis & Classification

process.

* Process: All the results—the original message, its sentiment, the bullying classification, and any associated report data—are saved to a Data Storage & Logging system (the database). This ensures a complete record for historical analysis.
* Data Flow: The data is committed to a persistent store, ready to be accessed later. Step 5: Automated Reporting
* Source: A specific High Severity data flow that is triggered only when the Analysis & Classification process identifies severe bullying.
* Process: The Automated Reporting process takes the report data and formats it into an official notification.
* Data Flow: This process sends an Email Notification to a designated external entity, such as Authorities or Support personnel.

Step 6: Data Visualization

* Source: The Historical Data retrieved from the Data Storage & Logging system.
* Process: The Data Visualization process takes the stored historical data and turns it into a readable format, such as graphs, charts, and dashboards.
* Data Flow: The output is Graphs & Insights, which are presented to System Administrators/Stakeholders to help them monitor trends and the overall effectiveness of the system.

# CHAPTER 5 SYSTEM IMPLEMENTATION

* 1. **DATA COLLECTION AND PREPROCESSING**

## INTRODUCTION

The effectiveness of any AI-based cyberbullying detection system heavily depends on the quality and quantity of the data used during training and testing. For BullyBlock AI, data collection focuses on gathering authentic online communication texts, including social media posts, chat messages, and forum comments. These datasets should contain both instances of cyberbullying and normal interactions to enable accurate classification. Data preprocessing is critical to prepare raw textual data for machine learning models. Raw data often contains noise such as punctuation, special characters, emojis, links, and inconsistent capitalization, which can negatively affect the performance of Natural Language Processing (NLP) models. Proper preprocessing ensures that the input to models like BERT and spaCy is clean, standardized, and suitable for feature extraction and model training.

## DATA SOURCES

For BullyBlock AI, data is collected from multiple reliable sources to ensure diversity and realism:

Social Media Platforms: Public posts and comments from platforms like Twitter, Reddit, and Instagram. These platforms contain informal language, slang, and emoticons, which are typical in online bullying contexts.

Open Cyberbullying Datasets: Publicly available datasets supplement real-time data. Examples include:

Kaggle Cyberbullying Dataset – contains annotated social media posts categorized by type of bullying.

Twitter Hate Speech Dataset – used to identify aggressive or offensive content.

## DATA ANNOTATION

Collected data must be labeled to train supervised learning models. Each message is annotated with:

Cyberbullying Label: Indicates whether the message contains bullying content.

Severity Level: Low, Medium, or High, based on intensity, recurrence, and potential harm. Sentiment Polarity: Positive, Neutral, or Negative sentiment for stakeholder analysis.

## DATA PREPROCESSING

Preprocessing transforms raw textual data into a clean and standardized format suitable for NLP models. The preprocessing steps in BullyBlock AI include:

Text Cleaning: Remove URLs, HTML tags, non-alphanumeric characters, excessive whitespace, and normalize capitalization.

Tokenization: Split text into words or tokens using spaCy, helping models understand word-level semantics.

Stopword Removal: Remove common words like “is,” “the,” and “and” that do not contribute to detecting bullying.

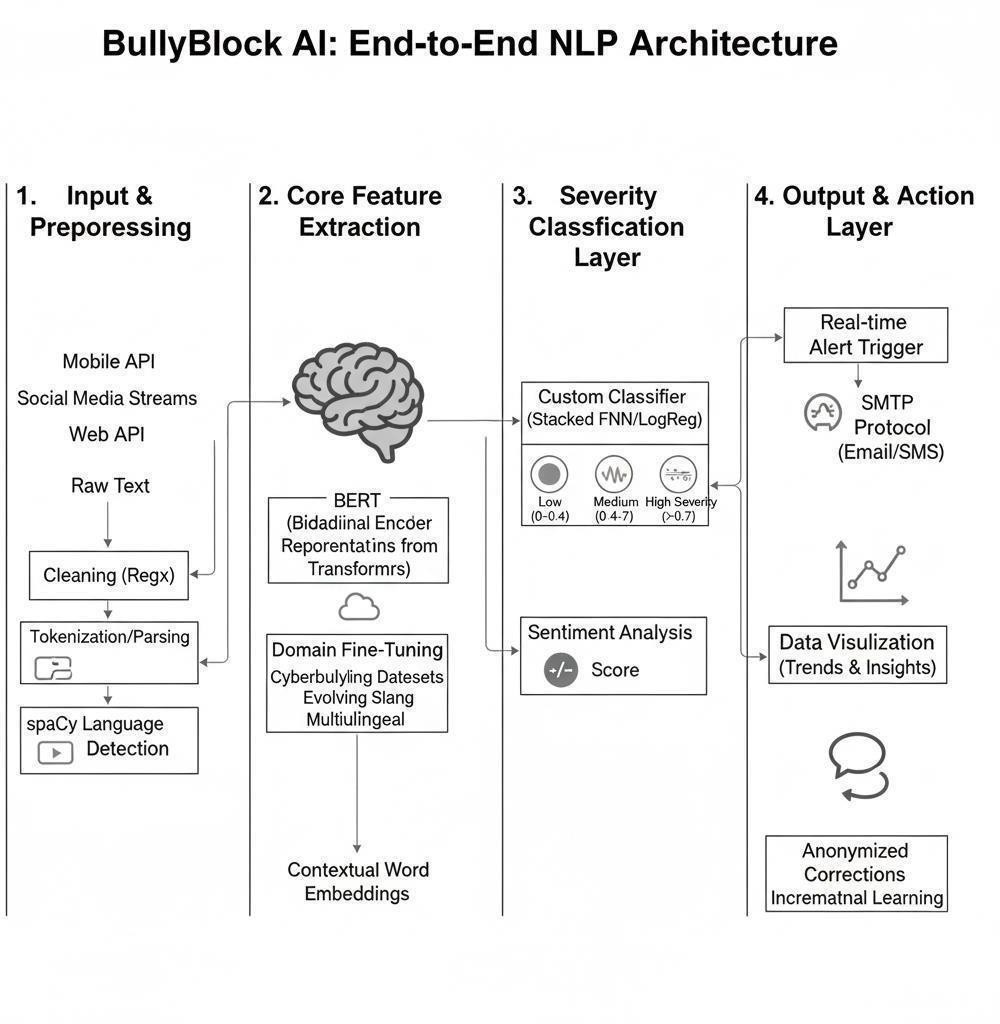
## DATA SPLITTING

After preprocessing, the dataset is divided into training, validation, and testing sets, typically in a 70:15:15 ratio. Stratified sampling ensures proportional representation of each severity level and sentiment class. This separation allows models to learn, tune, and evaluate effectively.

# NLP MODEL (BERT)

The Bidirectional Encoder Representations from Transformers (BERT) model plays a central role in BullyBlock AI for detecting and classifying cyberbullying content. Unlike traditional machine learning models that rely heavily on handcrafted features or keyword- based detection, BERT leverages the Transformer architecture to capture deep contextual relationships between words. This is especially critical in cyberbullying detection, where slang, sarcasm, and implicit abuse cannot be identified through isolated word analysis.BERT processes text bidirectionally, meaning it examines words in both left- to- right and right-to-left contexts simultaneously. For example, in the phrase *“You are smart, not like him”*, BERT considers the surrounding context to understand whether the sentence carries a negative or harmful implication. This enables the system to distinguish between neutral, positive, and abusive messages with higher accuracy.

BullyBlock AI, the implementation of BERT begins with fine-tuning a pre-trained BERT model (e.g., bert-base-uncased) on curated cyberbullying datasets. The text data undergoes preprocessing steps such as tokenization using BERT’s WordPiece tokenizer, padding/truncation for uniform sequence length, and conversion into input embeddings (token IDs, segment IDs, and attention masks). These embeddings are then fed into BERT’s encoder layers, which generate contextualized vector representations of each token. The output embeddings are passed to a custom classification head—a fully connected neural layer integrated with logistic regression—to predict the severity of messages as *Low, Medium, or High*.



**Fig. 5.2 NLP MODEL (BERT)**

This NLP architecture for BullyBlock AI is designed as an end-to-end system for automatically detecting, classifying the severity, and reporting cyberbullying in real- time.

1. Input & Preprocessing: Raw online messages are first gathered and cleaned of noise (like URLs/hashtags). The spaCy Framework is then used for efficient text parsing, recognizing elements like potential slang, code-switching, and identifying the language.
2. Core Feature Extraction: The cleaned text is fed into a BERT (Bidirectional Encoder Representations from Transformers) model. This pre-trained deep learning model is fine-tuned on diverse cyberbullying datasets, allowing it to understand the subtle *context* and *intent* behind the words—crucial for catching evolving slang and sarcasm. The output is a highly contextual Word Embedding vector.
3. Severity Classification Layer: The embeddings are processed by a Custom Classifier (like a Stacked Neural Network). This classifier performs Multi-Class Classification to categorize the abuse into one of three levels: Low, Medium, or High. An integrated Sentiment Analysis module simultaneously tracks the emotional tone of the messages over time.
4. Output & Action Layer: Based on the severity score, the system triggers a Real- time Alert via SMTP (email) to notify authorities or support teams for rapid intervention. All classifications, sentiments, and trends are stored for Data Visualization and long- term monitoring. Finally, a Feedback Loop allows moderators to send misclassification corrections back into the system for continuous, incremental learning and model refinement.
5. Predictive Sentiment Trend Forecasting

* Dynamic Baseline Calibration: BullyBlock AI continuously analyzes the aggregate positive and negative sentiments of all user interactions. This establishes a "normal" communication baseline for a community (e.g., a school, a group chat) over time.
* Deviation Forecasting: The system uses statistical or time-series analysis (e.g., ARIMA or Holt-Winters models applied to sentiment scores) to predict the future shape of the sentiment graph.

# MODEL TRAINING AND TESTING

The effectiveness of BullyBlock AI relies heavily on a robust training and testing pipeline to ensure accurate detection and severity classification of cyberbullying content. The process begins with data collection and preprocessing. Diverse, publicly available cyberbullying datasets such as Twitter, Kaggle, and social media forums are used to capture various abusive expressions, including slang, sarcasm, implicit bullying, and multilingual conversations. To prepare the data, text is cleaned through stopword removal, normalization, and tokenization using BERT’s WordPiece tokenizer. Special care is taken to balance class distributions across *Low, Medium, and High severity* categories to prevent bias toward majority classes.

During model training, a pre-trained BERT model (e.g., bert-base-uncased) is fine-tuned on the cyberbullying dataset. The text sequences are converted into token IDs, attention masks, and embeddings, which are fed into BERT’s transformer layers to generate contextualized word representations. These embeddings are passed into a custom classifier—a fully connected neural layer integrated with Logistic Regression—to categorize the severity of abuse. Cross-entropy loss is used as the objective function, while the Adam optimizer ensures efficient parameter updates. Training is carried out in mini- batches, with dropout layers applied to reduce overfitting and improve generalization.

For model evaluation, the dataset is split into training, validation, and testing subsets using stratified sampling to preserve class balance. Cross-validation techniques are employed to minimize bias and confirm the model’s reliability. Performance is measured using accuracy, precision, recall, and F1-score, with special focus on recall to reduce false negatives, since missing a harmful message can have serious consequences. Confusion matrices and ROC curves.

Finally, once validated, the trained model is deployed into the BullyBlock AI pipeline. During real-time testing, new user messages are passed through the trained BERT classifier, and severity levels are predicted instantly. If the detected severity is Medium or High, the system automatically triggers alerts via SMTP or webhook integrations, ensuring timely intervention. This end-to-end training and testing framework guarantees that BullyBlock AI remains both accurate and adaptable in real-world environments.

In addition to the core training and testing pipeline, continuous model monitoring and retraining are integrated into BullyBlock AI to maintain long-term effectiveness. Since patterns of online bullying evolve rapidly with new slang, memes, and cultural references, the system periodically collects fresh data from social media streams and user feedback. These new samples are used to fine-tune the model in iterative training cycles, preventing performance degradation over time. Furthermore, explainable AI techniques, such as attention weight visualization, are employed to provide transparency in predictions, helping educators, parents, and platform moderators understand why a particular message was flagged. This adaptive learning approach ensures that BullyBlock AI not only performs accurately at deployment but also remains resilient and responsive to emerging cyberbullying trends.

# AUTOMATEDEMAIL NOTIFICATION

A critical component of BullyBlock AI is its Automated Email Notification System, which ensures that severe cyberbullying incidents are reported immediately to responsible authorities, educators, or support teams. While traditional detection tools stop at flagging harmful messages, BullyBlock AI extends its functionality by enabling real-time intervention through automated alerts. This mechanism bridges the gap between detection and response, ensuring that victims receive timely support.

The email notification module is implemented using the Simple Mail Transfer Protocol (SMTP), a reliable and widely used protocol for sending emails programmatically. When a message is processed by the BERT-based classifier and categorized as *Medium* or *High severity*, the system automatically composes a structured email containing essential details of the incident. These include the flagged message, the severity level, timestamp, and the platform or user ID from which the message originated. This structured format allows stakeholders to quickly assess the situation and take corrective actions without manual screening delays.

To enhance reliability, the system supports configurable recipient lists, where administrators can specify email addresses of teachers, parents, counselors, or law- enforcement officers. The email body is designed to be concise yet informative, highlighting the abusive content and providing actionable insights. Security features such as authentication, encrypted connections (TLS/SSL), and access restrictions are integrated to prevent misuse and ensure the confidentiality of sensitive reports.

The notification pipeline is tightly coupled with the real-time detection engine. Once a high-severity message is identified, a background process is triggered to handle the email dispatch asynchronously, ensuring that system performance is not hindered .

The system also maintains a log of all sent notifications in CSV format for auditing and accountability purposes. Another advantage of this approach is scalability. Since the email notification module is modular and API-driven, it can be extended to integrate with webhooks, SMS gateways, or incident management platforms. This flexibility makes the system adaptable for deployment in schools, universities, social media monitoring teams, and law-enforcement agencies.

By combining automated detection with instant reporting, BullyBlock AI moves beyond passive monitoring to active intervention. The email notification system plays a vital role in creating a proactive safety net, ensuring that severe cyberbullying cases do not go unnoticed and that vulnerable individuals receive the support they need at the right time.



**Fig.5.4 Sample Email Notification**

# SENTIMENT ANALYSIS

In addition to detecting abusive content, sentiment analysis forms an essential layer of intelligence in BullyBlock AI. While cyberbullying detection focuses on identifying harmful or abusive messages, sentiment analysis evaluates the emotional tone of online interactions to provide a broader understanding of communication trends. This dual approach ensures that the system not only flags bullying but also captures the overall mood of conversations, thereby supporting preventive measures and long-term monitoring.

Sentiment analysis works by classifying text into positive, negative, or neutral sentiments. In BullyBlock AI, it is implemented using Natural Language Processing (NLP) pipelines that integrate transformer-based embeddings (BERT) with traditional sentiment classification models. Each message flagged by the system is first processed to extract contextual embeddings, which are then analyzed to determine the underlying sentiment. For example, a sentence like *“You will never succeed”* is categorized as both negative sentiment and potential bullying, whereas *“Great job on your presentation”* is identified as positive.

From a technical perspective, sentiment analysis in BullyBlock AI is optimized through incremental learning, allowing the classifier to adapt to evolving language styles, sarcasm, and code-mixed expressions often present in online communities. The results are visualized in dashboards using interactive charts and graphs, making it easier for stakeholders such as educators, parents, and policy makers to interpret communication dynamics.

By combining cyberbullying detection with sentiment analysis, BullyBlock AI goes beyond isolated incident reporting. It creates a comprehensive monitoring framework

that highlights harmful content and emotional well being.

This proactive strategy not only ensures immediate response to abusive incidents but also empowers institutions to track communication health, build preventive strategies, and foster safer, more positive online environments.

Example of Sentiment Analysis in BullyBlock AI

1. Message: *“You are such a loser, nobody likes you.”*
   * Cyberbullying Classification: High Severity
   * Sentiment Analysis Result: Negative (hostile, degrading tone)
   * Action: Alert triggered via automated email to administrator.
2. Message: *“Wow, you did an amazing job on your project!”*
   * Cyberbullying Classification: Not abusive
   * Sentiment Analysis Result: Positive (encouraging and supportive)
   * Action: No alert, but positive sentiment is logged for monitoring trends.
3. Message: *“I think your answer could be better, but it’s fine.”*
   * Cyberbullying Classification: Low Severity (mild criticism, not abusive)
   * Sentiment Analysis Result: Neutral/Mixed (slightly critical but not hostile)
   * Action: Logged for review; no automated alert sent.

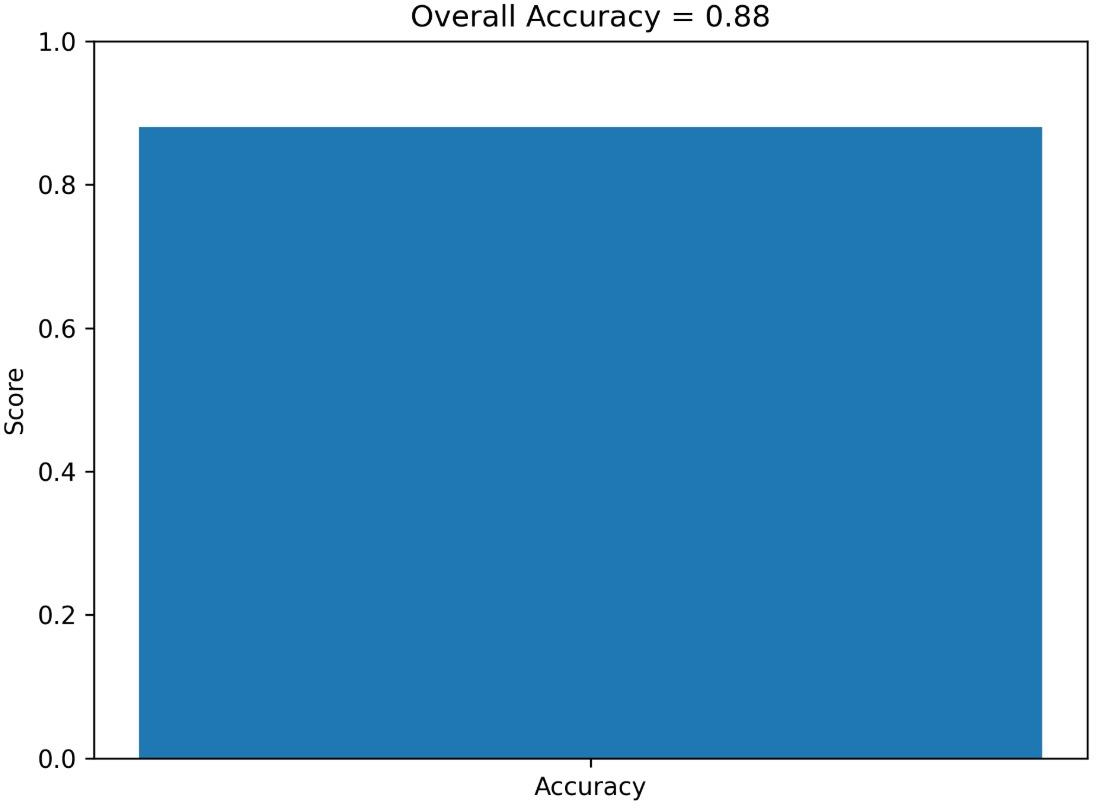
# CHAPTER 6 PERFORMANCE ANALYSIS

* 1. **EVALUATION METRICS (ACCURACY, MATRIX, F1)**

To measure the effectiveness of BullyBlock AI in detecting cyberbullying, it is essential to evaluate the trained model using appropriate performance metrics. Since the project deals with a classification problem (bullying vs. non-bullying, or varying severity levels), simple accuracy is not enough. In imbalanced datasets, where non-bullying messages may dominate, metrics such as Precision, Recall, and F1-score give deeper insights into the model’s ability to detect harmful content.

### Accuracy

Accuracy is the ratio of correctly predicted instances (both bullying and non-bullying) to the total number of predictions. While it provides a general measure of performance, accuracy alone may be misleading if the dataset is unbalanced.

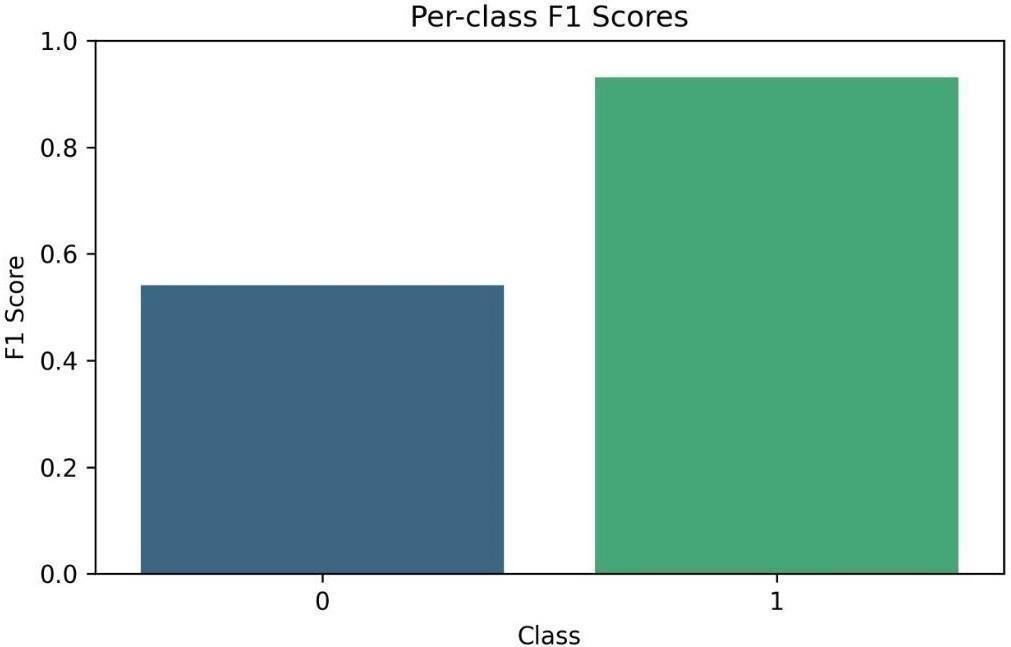


### Fig. 6.1 Accuracy

1. **F1-Score**

The F1-score is the harmonic mean of Precision and Recall. It provides a balanced metric when both false positives and false negatives are important. Since cyberbullying

detection must minimize missed cases while avoiding over-flagging harmless text, F1- score is a reliable indicator of overall system performance.



### Fig. 6.1 F1-Score

1. **Confusion Matrix Analysis**

The confusion matrix is a visual representation used to evaluate the performance of a classification model. It compares the predicted classifications with the actual classifications to reveal the model’s accuracy, errors, and misclassifications.

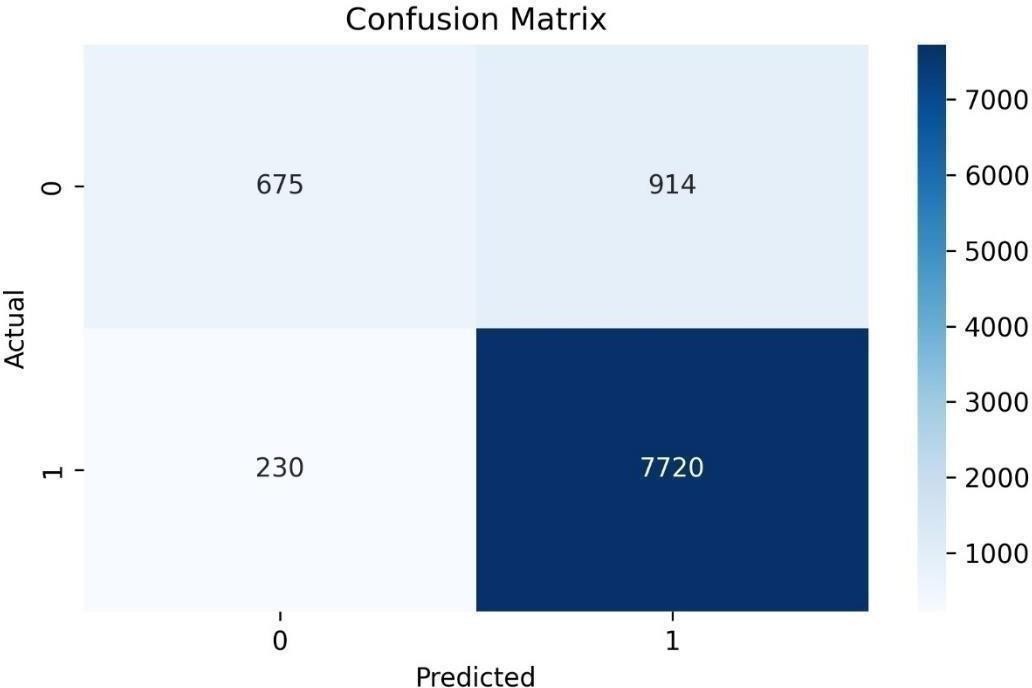
In the provided diagram, the confusion matrix has four key components:

* + True Positive (TP): 7720 cases where the model correctly predicted the positive class.
  + True Negative (TN): 675 cases where the model correctly predicted the negative class.
  + False Positive (FP): 914 cases where the model incorrectly predicted a positive class when it was actually negative.
  + False Negative (FN): 230 cases where the model incorrectly predicted a negative class when it was actually positive.

This matrix allows calculation of evaluation metrics such as accuracy, precision, recall, and F1-score. In this case, the high number of true positives (7720) shows that the model is strong in detecting positive cases, but the presence of false positives (914) indicates some over-prediction. Similarly, the false negatives (230) show there are still missed cases that need improvement.

Overall, this confusion matrix provides a clear and concise way to assess the model’s

strengths and weaknesses in classification performance.



**Fig. 6.1 Confusion Matrix Analysis**

## MODEL COMPARISON

In this project, two different Natural Language Processing (NLP) models, BERT and spaCy, were considered for detecting cyberbullying and harmful content. Both models serve the same goal of understanding and classifying text, but they differ significantly in their design, performance, and approach.

BERT (Bidirectional Encoder Representations from Transformers)

BERT is a transformer-based deep learning model developed by Google that has revolutionized NLP tasks. It reads text bidirectionally (from left to right and right to left simultaneously), allowing it to capture deeper contextual meaning. In the context of cyberbullying detection, this capability is particularly useful because harmful messages often depend on subtle cues, sarcasm, or word combinations that cannot be understood in isolation. BERT provides high accuracy and robustness, making it well-suited for detecting implicit bullying. However, BERT requires significant computational resources and training time, which makes it more demanding compared to traditional NLP tools.

spaCy

spaCy is a fast and efficient NLP library designed for production-level applications. It offers pre-trained models for tokenization, part-of-speech tagging, named entity recognition, and text classification. Compared to BERT, spaCy is lightweight and much faster in processing large volumes of text, making it more practical for real-time detection systems. However, its limitation lies in its shallower contextual understanding. While spaCy can handle explicit bullying detection effectively using rules and keyword- based methods, it struggles with subtle or context-dependent harmful messages.

Comparison

Accuracy: BERT generally outperforms spaCy in terms of classification accuracy due to its ability to capture deep context.

Speed: spaCy is faster and more resource-efficient, suitable for large-scale real-time processing.

Complexity: BERT is computationally intensive, whereas spaCy is lightweight and easier to deploy.

Use Case: For high-accuracy detection of subtle bullying, BERT is more effective, while spaCy is better for faster, less resource-demanding tasks.

Thus, BERT was prioritized for this project as the primary detection model due to its superior ability to handle complex and context-dependent cyberbullying scenarios, while spaCy provided additional preprocessing and keyword-based support.

## OVERALL PERFORMANCE SUMMARY

The overall performance of the cyberbullying detection system was evaluated through multiple experiments and assessments, using accuracy, recall, precision, and F1-score as the primary evaluation metrics. The results of these evaluations reveal the effectiveness of the integrated models and highlight the strengths as well as limitations of the approach.

### System Evaluation

The system integrates three major components—BERT, Logistic Regression classifier, and spaCy preprocessing/keyword filtering—to achieve robust cyberbullying detection. The use of BERT allowed the model to capture deep semantic meaning and context, particularly in sentences where harmful intent is implied rather than explicit. Logistic Regression provided a lightweight classification layer on top of BERT embeddings, ensuring that the model’s decision-making was both accurate and interpretable. Additionally, spaCy played an important role in handling text preprocessing, tokenization, and rule-based keyword checks to complement the machine learning models.

The evaluation of the system was conducted using a test dataset containing real-world samples of cyberbullying messages along with normal online conversations. The confusion matrix (as shown in the previous section) illustrated the distribution of correct and incorrect predictions. With 7720 true positives and 675 true negatives, the system demonstrated strong performance in classifying bullying content. However, the presence of 914 false positives suggests that the system sometimes flagged harmless messages as bullying, while 230 false negatives indicated that a small fraction of harmful messages were missed.

### Limitations of the Model

* + False Positives: A noticeable number of harmless messages were misclassified as bullying. This could lead to unnecessary interventions or message deletions.
  + False Negatives: Although relatively low, the presence of undetected harmful messages indicates the need for continued model training with more diverse datasets.
  + Resource Requirements: BERT is computationally intensive, requiring significant processing power and memory. This makes deployment on lightweight devices more challenging.
  + Dependence on Training Data: The model’s performance depends heavily on the quality and diversity of the training dataset. Limited representation of slang, regional languages, or evolving online expressions can reduce detection accuracy.

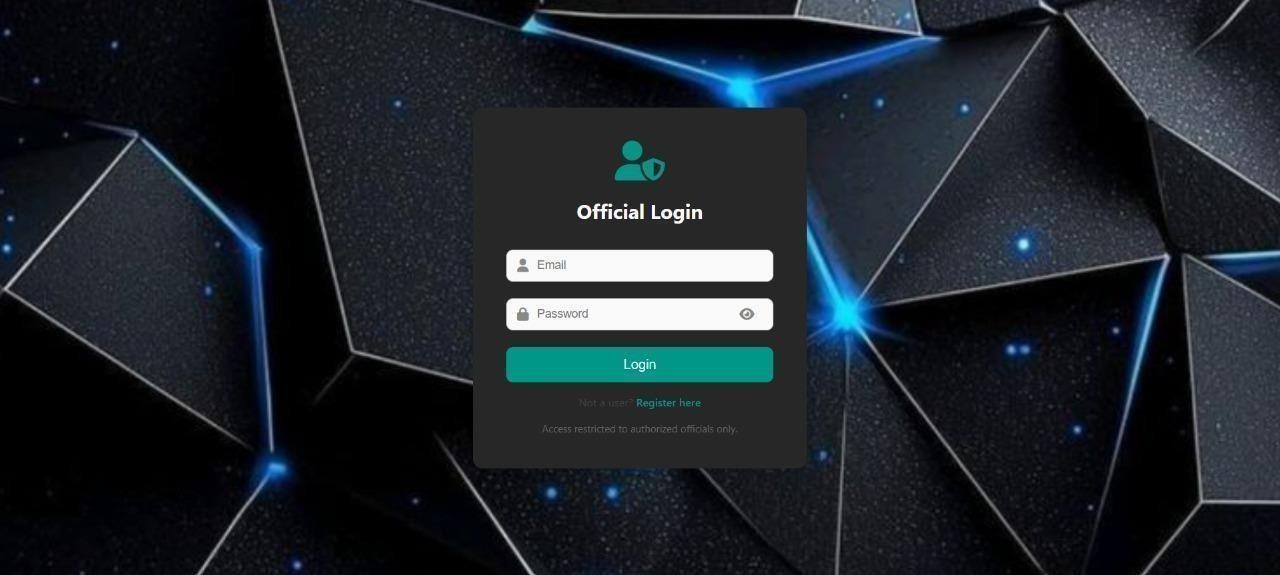
Comparison with Traditional Methods

Traditional keyword-based systems often fail to capture subtle or context-dependent forms of harassment. For example, words that appear harmless in isolation may become abusive in a particular context. Compared to such systems, this AI-driven approach significantly reduces missed detections.

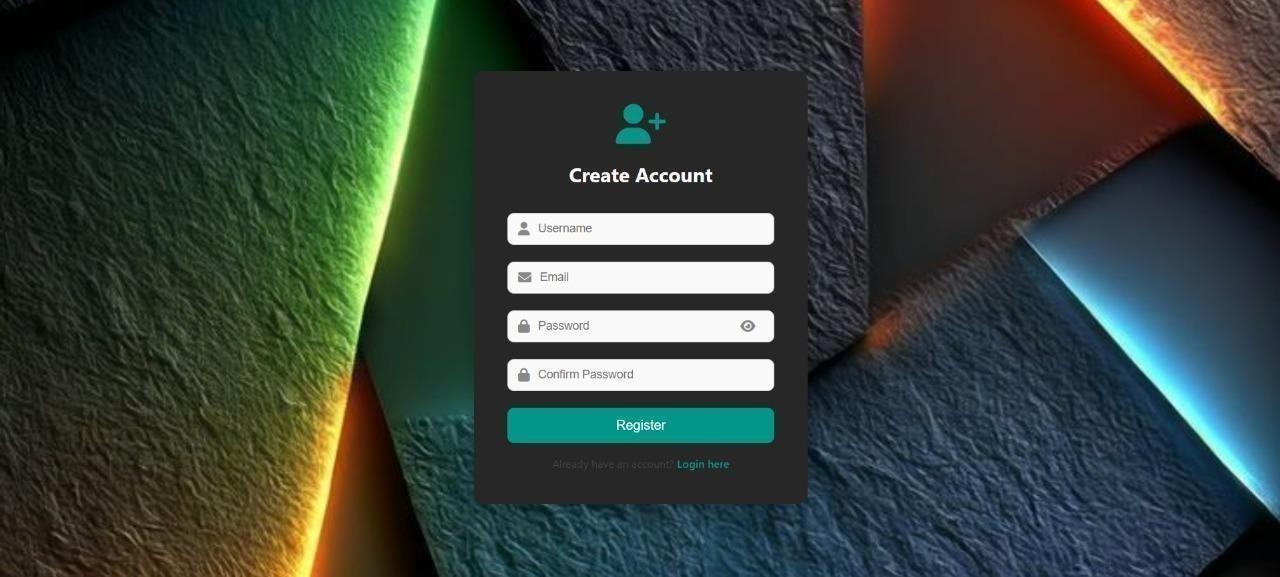
## CHAPTER 7 RESULTS AND DISCUSSION

* 1. **SCREENSHOTS OF BULLYBLOCK AI DASHBOARD**

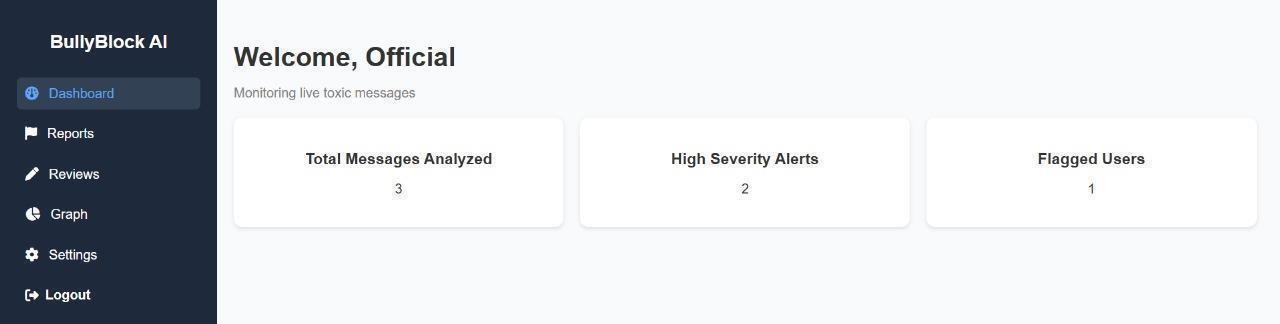
The BullyBlock AI dashboard provides a clear visualization of the system’s results, showcasing how the model processes user inputs and classifies them as either safe or potentially harmful. The screenshots highlight the step-by-step flow of text input, detection, and output labeling, making the results transparent and easy to interpret. When a harmful message is detected, the system immediately flags it and triggers automated actions such as alerts or notifications. This ensures that the detection process is not only accurate but also actionable in real time. The dashboard also provides insights into classification outcomes, including whether a message is categorized as cyberbullying or non-bullying, thereby giving users and administrators a practical overview of the system’s effectiveness.



**Fig.7.1 LOGIN PAGE**

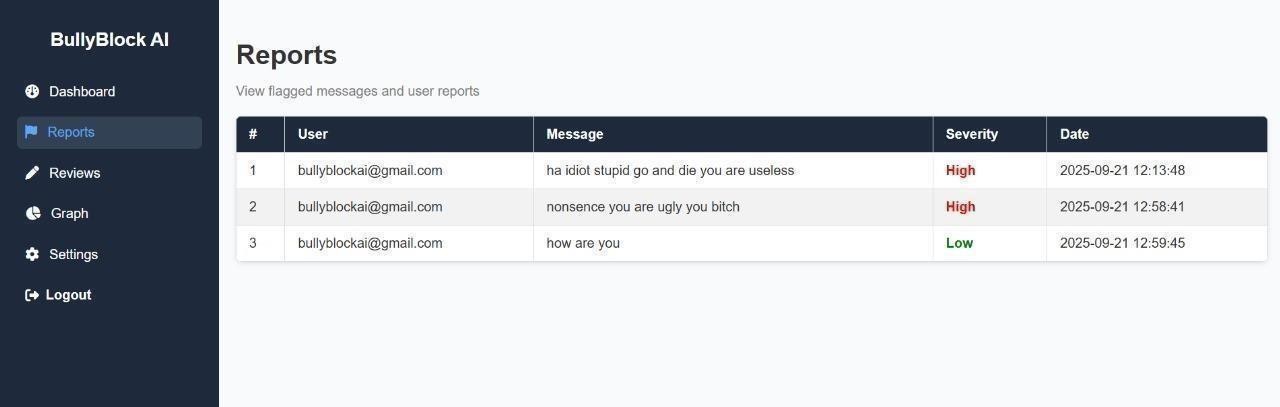


**Fig. 7.1 CREATE ACCOUNT**

****

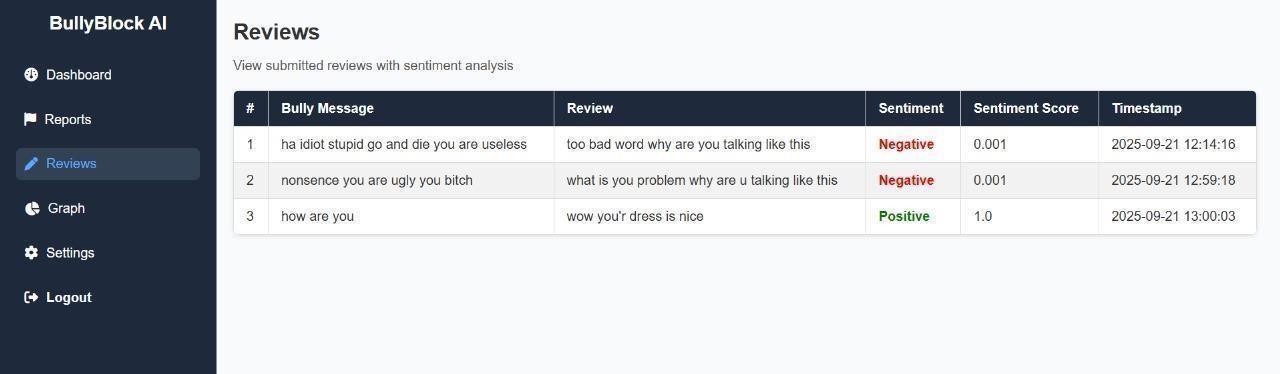
**Fig.7.1 DASHBOARD**

The Dashboard page acts as the central hub of the BullyBlock AI system, offering authorized officials a quick and comprehensive overview of the platform’s real-time monitoring performance. It displays the total number of messages analyzed, the count of high-severity alerts and the number of flagged users whose activities have shown repeated abusive behavior. By presenting this summarized information in an intuitive and organized manner, the dashboard enables officials to easily track system activity, assess the overall level of online toxicity, and quickly identify cases that require immediate attention or intervention.



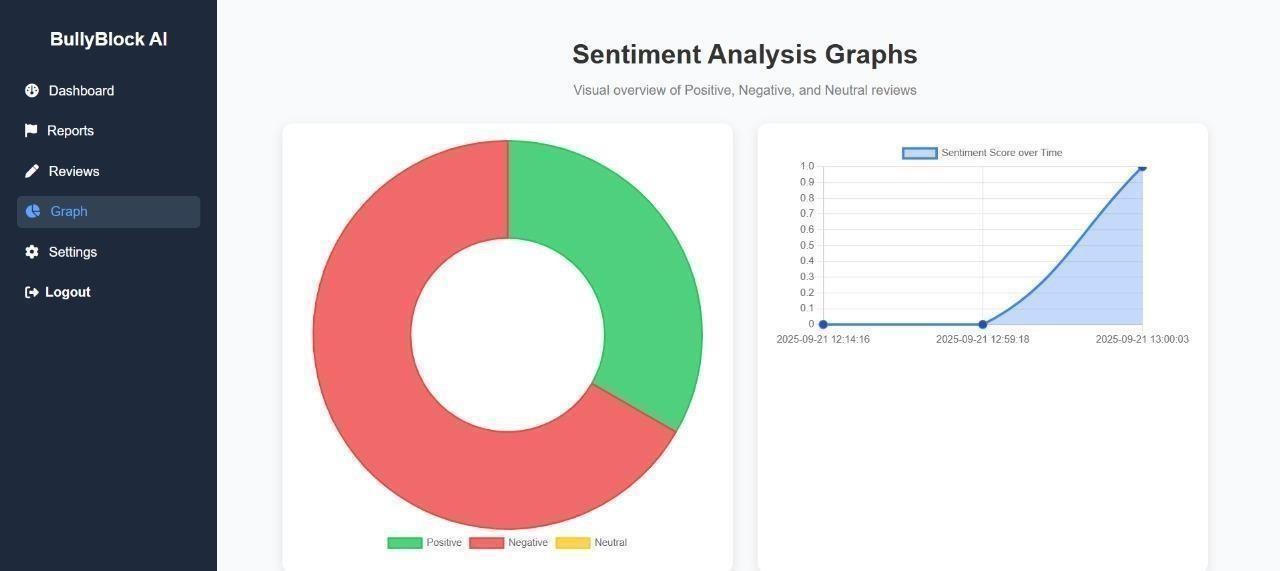
### Fig. 7.1 REPORTS

The Reports page displays all messages flagged by the NLP model as abusive or harmful. Each entry shows the user’s email, message content, severity level, and detection time. Using BERT and spaCy, the system classifies messages as *Low*, *Medium*, or *High* severity.



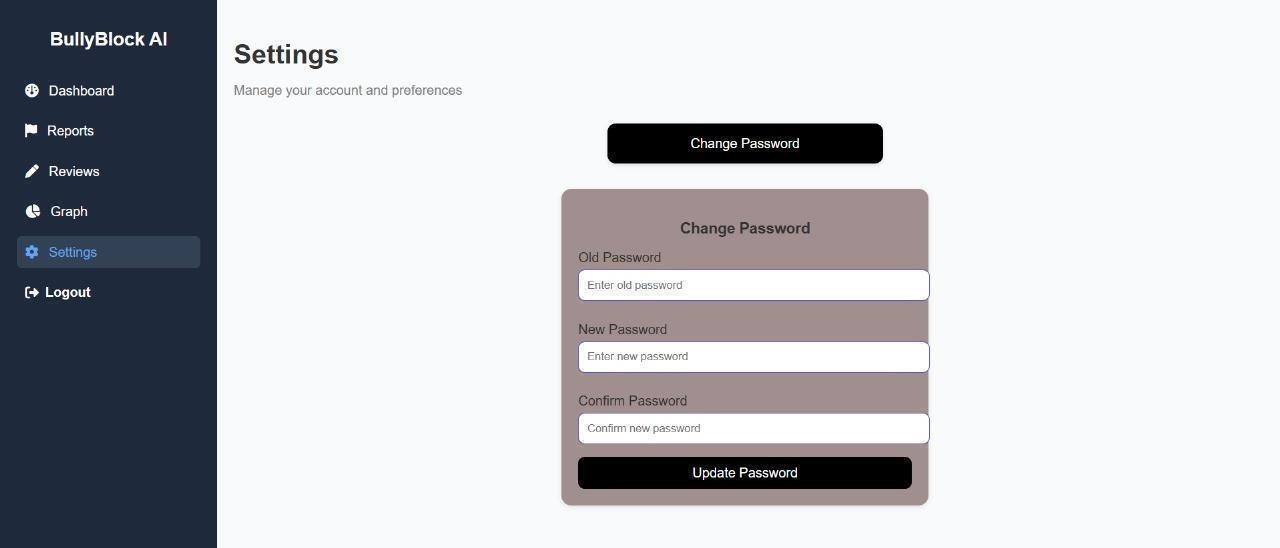
### Fig.7.1 REVIEWS

The Reviews page collects user feedback and performs sentiment analysis on responses related to detected bullying incidents. It displays the original harmful message, the user’s review or comment, the sentiment classification (positive or negative), the sentiment score, and the exact timestamp. Using NLP-based sentiment analysis, this page helps visualize emotional reactions and track changes in user sentiment over time, providing valuable insights into the overall digital communication environment and user behavior.



### Fig .7.1 SENTIMENT ANALYSIS GRAPH

The Sentiment Analysis Graphs show the distribution of positive and negative sentiments from user interactions. They help visualize changes in online behavior over time, making it easier to identify trends in user emotions and assess the overall communication climate on the platform.



**Fig .7.1 SETTINGS**

## CHAPTER 8

## 8. CONCLUSION AND FUTURE SCOPE

### Conclusion:

Cyberbullying poses a severe challenge in today’s digital era, especially among young users who are highly active on social media platforms and online communication channels. Traditional systems have primarily focused on detecting abusive content, leaving a gap in severity assessment, timely intervention, and automated reporting. This limitation often delays response and places the burden on victims to report incidents manually.

The proposed solution, BullyBlock AI, successfully addresses these shortcomings by providing a comprehensive, end-to-end framework for cyberbullying detection and intervention. By integrating advanced Natural Language Processing (NLP) techniques with transformer models like BERT and the spaCy framework, the system ensures accurate detection of offensive content, including context-dependent abuse, slang, sarcasm, and multilingual expressions.

One of the key strengths of the system lies in its severity classification mechanism, which categorizes harmful content into Low, Medium, and High levels. This not only improves prioritization but also enables timely responses to critical incidents. Furthermore, the integration of an automated reporting module using SMTP ensures that high-severity or recurring abuse cases are immediately escalated to relevant authorities, such as parents, school administrators, or moderators, thereby reducing response delays.

The inclusion of sentiment analysis and interactive dashboards provides a holistic view of the overall communication environment, allowing stakeholders to monitor both harmful and positive interactions. This makes the system not just reactive but also proactive in identifying trends and patterns in online behavior.

In conclusion, BullyBlock AI offers a robust, scalable, and practical solution for reducing the impact of cyberbullying. By combining detection, severity assessment, automated reporting, and visualization, it creates a safer digital space while ensuring victims receive timely support. The system demonstrates the potential of Artificial Intelligence in addressing social challenges and contributes toward building healthier online communities.

### Future Scope:

While BullyBlock AI provides a strong foundation for automated cyberbullying detection and intervention, there are several avenues for future enhancements and research:

1. Multimodal Analysis

Current detection is text-based. Future versions can integrate image, audio, and video analysis to identify memes, edited photos, or voice messages containing bullying elements.

1. Cross-Platform Integration

Expansion to multiple social media platforms, forums, and messaging apps through API integration will enhance coverage and usability across diverse digital ecosystems.

1. Advanced Deep Learning Models

Incorporating transformer-based large language models (LLMs) beyond BERT (such as RoBERTa, GPT-based fine-tuned classifiers, or multilingual transformers) can improve contextual understanding and adaptability to evolving slang and multilingual communication.

1. Mobile and Real-Time Applications

Developing mobile apps and browser plug-ins will enable real-time monitoring and alerts, providing on-device safety for students and users .

1. Personalized Intervention Mechanisms
   * Instead of generic alerts, future systems can suggest tailored interventions, such as counseling resources, peer-support groups, or AI chatbots that provide immediate emotional support to victims.
2. Gamification and Awareness Tools
   * To complement detection, educational modules or gamified awareness tools can be integrated to teach students and young users about digital ethics, empathy, and responsible online behavior.
3. Integration with Law Enforcement and Policy Frameworks
   * Future development can include seamless integration with law-enforcement databases, child protection frameworks, and cybercrime monitoring units to strengthen accountability and formal response mechanisms.
4. Enhanced Privacy and Ethical AI
   * Ongoing improvements in privacy-preserving AI, such as federated learning and differential privacy techniques, will ensure user data security while maintaining model performance.
5. Scalability with Cloud and Edge Computing
   * Deploying BullyBlock AI on cloud platforms and edge devices can support large-scale adoption in schools, universities, and social media networks with minimal latency.

# APPENDICES

## SDG GOAL MAPPING

### Primary Goal: Good Health and Well-being (Goal 3)

Cyberbullying has a profound impact on the mental health and well-being of individuals, especially children and young adults. Victims often experience stress, anxiety, depression, and in severe cases, self-harm tendencies. The primary goal of BullyBlock AI aligns with SDG 3 – Good Health and Well-being, by aiming to detect and address harmful content before it escalates. Through early detection and severity classification, the tool acts as a preventive mechanism, reducing long-term psychological damage caused by online abuse.

* + - Target 1: By identifying abusive behavior at an early stage, BullyBlock AI promotes prevention and intervention, thereby reducing the negative impact on mental health.
    - Target 2: The system strengthens digital early-warning capacities for online risks, contributing to risk reduction and the management of harmful content that may otherwise spread widely.
    - Target 3: By functioning as a real-time monitoring and alert system, BullyBlock AI supports early warning for harmful digital behavior, ensuring quick intervention and protecting vulnerable groups.

In this way, the project not only safeguards individuals’ mental well-being but also builds resilience against long-term health risks associated with digital abuse.

### Secondary Goal: Peace, Justice, and Strong Institutions (Goal 16)

Cyberbullying is more than just a social media issue—it is a form of psychological violence that threatens peace and safety in digital spaces. BullyBlock AI contributes to SDG 16 – Peace, Justice, and Strong Institutions, by detecting and reporting harmful content.

* + - Target 1: The system directly supports the reduction of violence in digital communities by detecting abusive messages and minimizing harmful interactions.
    - Target 2: By preventing online abuse, particularly against children and youth, BullyBlock AI addresses one of the most prevalent forms of digital exploitation, aligning with global efforts to protect vulnerable groups.
    - Target 3: Since cyberbullying is a digital form of abuse, BullyBlock AI directly combats it by acting as a protective tool, reinforcing safe online spaces and ensuring justice in digital interactions.

In addition, the project strengthens institutional capacity by enabling schools, colleges, and online platforms to adopt a structured system for monitoring and reporting harmful behavior. This creates accountability within digital communities and promotes fair treatment for victims. By automating alerts through email and webhook integrations, the system ensures that abusive incidents are not ignored, but rather escalated to responsible authorities for action. Furthermore, BullyBlock AI fosters a culture of digital citizenship and responsibility, encouraging users to interact respectfully online.

Thus, the project does not merely detect harmful messages but also contributes to the broader vision of peace and justice by empowering institutions, supporting child protection initiatives, and promoting ethical use of technology. In the long run, this aligns with building strong, inclusive, and safe digital institutions that protect individuals and nurture positive online communities.

## SAMPLE SOURCE CODE

### App.py

from flask import Flask, request, jsonify import csv

from datetime import datetime from flask\_cors import CORS

from bully\_detector import classify\_severity, save\_log\_local app = Flask(\_name\_) CORS(app)

LOG\_FILE = "logs.csv" REVIEWS\_FILE =

"reviews.csv" # --- Home route ---

@app.ro ute("/") def home():

return " BullyBlock AI API is running! Use /stats, /analyze, /reports, or

/reviews." # --- Analyze a new message --- @app.route("/analyze", methods=["POST"]) def

classify\_severity(message) save\_log\_local(user\_email, message, severity) log\_entry = {

"user": user\_email, "message": message, "severity": severity,

"date": datetime.now().strftime("%Y-%m-%d %H:%M:%S")

}

# --- Get stats from logs --- @app.route("/stats", methods=["GET"]) def get\_stats():

total = high = flagged = 0 flagged\_users = set() try:

with open(LOG\_FILE, newline="", encoding="utf-8") as f: reader = csv.DictReader(f)

if not reader.fieldnames or "Severity" not in reader.fieldnames: return jsonify({"total": 0, "high": 0, "flagged": 0})

for row in reader: total += 1

severity = row.get("Severity") user = row.get("User Email") if severity == "High":

high += 1 if user:

flagged\_users.add(us

er) flagged = len(flagged\_users)

except FileNotFoundError:

return jsonify({"total": 0, "high": 0, "flagged":

0}) return jsonify({ "total": total, "high": high, "flagged": flagged

})

# --- Get full reports (for Reports page) --- @app.route("/reports", methods=["GET"])

reader = csv.DictReader(f) if not reader.fieldnames:

return jsonify([]) for row in reader:

logs.append({

"date": row.get("Timestamp", ""),

"user": row.get("User Email", "unknown"),

"message": row.get("Message", ""),

"severity": row.get("Severity", "")

})

except

FileNotFoundError:

return jsonify([]) return jsonify(logs)

# --- Get full reviews (for Reviews page) --- @app.route("/reviews", methods=["GET"]) def get\_reviews():

review s = []

try:

with open(REVIEWS\_FILE, newline="", encoding="utf-8") as f: reader = csv.DictReader(f)

if not reader.fieldnames: return jsonify([])

for row in reader: reviews.append({

"timestamp": row.get("Timestamp", ""), "bully\_message": row.get("Bully Message", ""), "review": row.get("Review", ""),

"sentiment": row.get("Sentiment", ""), "sentiment\_score": row.get("Sentiment Score", "")

return jsonify(reviews) if

\_name\_ == "\_main\_":

app.run(debug=True)

### bully\_detector.py

# bully\_detector.py (updated with bully + review logging) import os

import torch import smtplib

from datetime import datetime

from email.mime.text import MIMEText from transformers import AutoTokenizer,

AutoModelForSequenceClassification import csv # import sentiment classifier you created

from review\_detector import classify\_sentiment # === Email Config

===

EMAIL\_SENDER = ["bullyblockai@gmail.com"](mailto:bullyblockai@gmail.com) EMAIL\_RECEIVER = "[bullyblockai@gmail.com"](mailto:bullyblockai@gmail.com)

EMAIL\_APP\_PASSWORD = os.environ.get("EMAIL\_APP\_PASSWORD",

"slbmuxkzhnsapjgy") # === Rule-based toxic keywords (extra safety) === TOXIC\_WORDS = ["stupid", "idiot", "ugly", "hate", "kill", "moron", "loser", "foolish", "dumb"] # === Load Toxic-BERT model ===

p rint (" =\_T+‡Lo ading Toxic-BERT model...")

tokenizer = AutoTokenizer.from\_pretrained("unitary/toxic-bert")

if word in text.lower():

return "High", 1.0 # forced high

inputs = tokenizer(text, return\_tensors="pt", padding=True, truncation=True, max\_length=128).to(device) with torch.no\_grad():

outputs = bert\_model(\*\*inputs)

probs = torch.softmax(outputs.logits, dim=1)[0] # [non-toxic, toxic] toxic\_score = probs[1].item()

if toxic\_score < 0.3: return "Low",

toxic\_score elif 0.3 <= toxic\_score < 0.7:

return "Medium", toxic\_score else:

return "High", toxic\_score def send\_email(subject, body):

"""Send email alert""" try:

msg = MIMEText(body) msg["Subject"] = subject msg["From"] = EMAIL\_SENDER msg["To"] =

EMAIL\_RECEIVER

with smtplib.SMTP\_SSL("smtp.gmail.com", 465) as server: server.login(EMAIL\_SENDER, EMAIL\_APP\_PASSWORD) server.sendmail(EMAIL\_SENDER, EMAIL\_RECEIVER, msg.as\_string())

Email sent successfully!") except Exception as e:

print(f"+ Email failed: {e}")

def save\_log\_local(user\_email, message, severity,

writer.writerow([

datetime.now().strftime("%Y-%m-%d %H:%M:%S"), user\_email,

message, severity,

round(toxic\_score, 3)

])

Saved locally to logs.csv")

def save\_review\_local(bully\_message, review\_text, sentiment\_label, sentiment\_score): """Save bully message + review + sentiment to reviews.csv"""

with open("reviews.csv", mode="a", newline="", encoding="utf-8") as file: writer = csv.writer(file)

if file.tell() == 0:

writer.writerow(["Timestamp", "Bully Message", "Review", "Sentiment", "Sentiment Score"]) writer.writerow([

datetime.now().strftime("%Y-%m-%d %H:%M:%S"), bully\_message,

review\_text, sentiment\_label, round(sentiment\_score, 3)

])

print(",’f)‘˙S aved review to reviews.csv") def main(): user\_email = EMAIL\_SENDER user\_input = input("Message: ").strip() if not user\_input: print("No message entered.

Exiting.") return

elif severity == "Medium":

print(f"⚠ MEDIUM: Message deleted. (Score:

{toxic\_score:.2f})") elif severity == "High":

print(f"HIGH: Message deleted & email alert sent. (Score: {toxic\_score:.2f})") subject = "B ull y Blo ck AI Alert – High Toxicity Detected"

body = (

f"A high toxicity message was detected.\n\n" f"User: {user\_email}\n"

f"Message: {user\_input}\n" f"Severity: {severity}\n"

f"Toxic Score: {toxic\_score:.2f}\n"

f"Time: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}\n"

)

send\_email(subject, body) save\_log\_local(user\_email, user\_input, severity, toxic\_score) # --- NEW: Ask for review and classify sentiment ---

review\_input = input("Review (type something and press Enter): ").strip() if review\_input:

sentiment\_label, sentiment\_score = classify\_sentiment(review\_input)

print(f"¸•?Ç Review Sentiment: {sentiment\_label} (score: {sentiment\_score:.2f})") save\_review\_local(user\_input, review\_input, sentiment\_label, sentiment\_score) else:

print("No review entered.

Done.") if \_name\_ == "\_main\_":

main()

### dashboard.html

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" href="css/dashboard.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">

</head>

<body>

<div class="dashboard-container">

<!-- Sidebar -->

<aside class="sidebar">

<h2>BullyBlock AI</h2>

<ul class="nav-menu">

<li class="active"><a href="dashboard.html"><i class="fa-solid fa-gauge"></i> Dashboard</a></li>

<li><a href="reports.html"><i class="fa-solid fa-flag"></i> Reports</a></li>

<li><a href="reviews.html"><i class="fa-solid fa-pen"></i> Reviews</a></li>

<li><a href="graph.html"><i class="fa-solid fa-chart-pie"></i> Graph</a></li>

<li><a href="settings.html"><i class="fa-solid fa-gear"></i> Settings</a></li>

<li id="logoutBtn"><i class="fa-solid fa-right-from-bracket"></i> Logout</li>

</ul>

</aside>

<!-- Main Content -->

<main class="main-content">

<h1>Welcome, Official</h1>

<p class="sub-heading">Monitoring live toxic messages</p>

<div class="stats-cards">

<div class="card">

<h3>Total Messages Analyzed</h3>

<p id="totalMessages">0</p>

</div>

<div class="card">

<h3>High Severity Alerts</h3>

<p id="highAlerts">0</p>

</div>

<div class="card">

<h3>Flagged Users</h3>

<p id="flaggedUsers">0</p>

</div>

</div>

</main>

</div>

<!-- Firebase + Logout Script -->

<script type="module">

// Import Firebase SDKs

import { initializeApp } from "https://[www.gstatic.com/firebasejs/12.2.1/firebase-app.js"](http://www.gstatic.com/firebasejs/12.2.1/firebase-app.js); import { getAuth, onAuthStateChanged, signOut } from

"https://[www.gstatic.com/firebasejs/12.2.1/firebase-](http://www.gstatic.com/firebasejs/12.2.1/firebase-) auth.js";

// Firebase configuration const firebaseConfig = {

apiKey: "AIzaSyDFqoJ0cxZRj6Gf\_Q4h5awQa\_7xTilR7AM", authDomain: "bully-block-ai-39bf7.firebaseapp.com", databaseURL: "https://bully-block-ai-39bf7-default-rtdb.asia- southeast1.firebasedatabase.app", projectId: "bully-block-ai-39bf7", storageBucket: "bully-block-ai-

39bf7.firebasestorage.app", messagingSenderId: "1029684496192",

appId: "1:1029684496192:web:ea9e7f5f8a5e5022ca5ca5"

};

// Initialize Firebase const app =

initializeApp(firebaseConfig); const auth

= getAuth(app);

const logoutBtn = document.getElementById("logoutBtn");

// Redirect to login if user is not logged in

// Logout functionality logoutBtn.addEventListener("click", ()

=> { signOut(auth)

.then(() => {

window.location.href = "login.html"; // redirect after logout

})

.catch((error) => {

alert("Error signing out: " + error.message);

});

});

// Fetch live stats from Flask backend async function loadStats()

{

try {

const res = await fetc[h("http://127.0.0.1:5000/stats"](http://127.0.0.1:5000/stats)); const data = await res.json(); document.getElementById("totalMessages").innerText =

data.total; document.getElementById("highAlerts").innerText = data.high; document.getElementById("flaggedUsers").innerText = data.flagged; c onsol e . l o g( " Stats loaded:", data);

} catch (error) {

console.error("+ Error loading stats:", error);

}

}

// Load once on page open loadStats();

// Refresh every 5 seconds setInterval(loadStats, 5000);

</script>

</body>

</html>

### dashboard.js

import { initializeApp } from "https://[www.gstatic.com/firebasejs/12.2.1/firebase-app.js"](http://www.gstatic.com/firebasejs/12.2.1/firebase-app.js); import { getAuth, onAuthStateChanged, signOut } from "https://[www.gstatic.com/firebasejs/12.2.1/firebase-](http://www.gstatic.com/firebasejs/12.2.1/firebase-) auth.js";

// Firebase configuration const firebaseConfig = {

apiKey: "AIzaSyDFqoJ0cxZRj6Gf\_Q4h5awQa\_7xTilR7AM", authDomain: "bully-block-ai-39bf7.firebaseapp.com", databaseURL: "https://bully-block-ai-39bf7-default-rtdb.asia- southeast1.firebasedatabase.app", projectId: "bully-block-ai-39bf7", storageBucket: "bully-block-ai-

39bf7.firebasestorage.app", messagingSenderId: "1029684496192",

appId: "1:1029684496192:web:ea9e7f5f8a5e5022ca5ca5"

};

// Initialize Firebase const app =

initializeApp(firebaseConfig); const auth = getAuth(app);

// Protect page: redirect to login if not authenticated onAuthStateChanged(auth, (user) => {

if (!user) {

window.location.href = "login.html";

}

});

// Fet ch live stats from Flask backend async function loadStats()

{

try {

const res = await fetch[("http://127.0.0.1:5000/stats"](http://127.0.0.1:5000/stats)); const data = await res.json();

document.getElementById("totalMessages").innerText = data.total; document.getElementById("highAlerts").innerText = data.high; document.getElementById("flaggedUsers").innerText

= data.flagged; c onsol e . l o g( " Stats loaded:", data);

} catch (error) {

console.error("+ Error loading stats:", error);

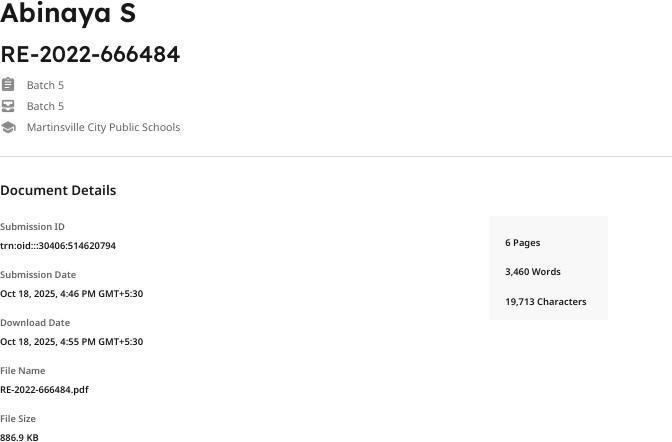
}

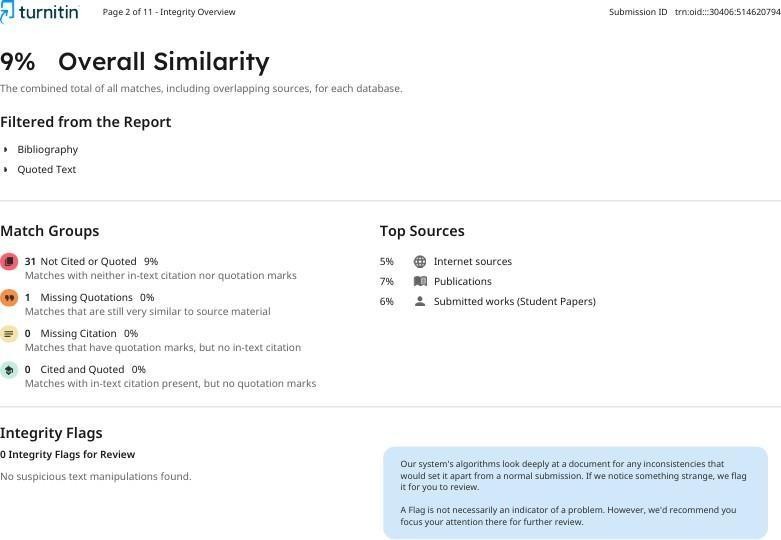
}

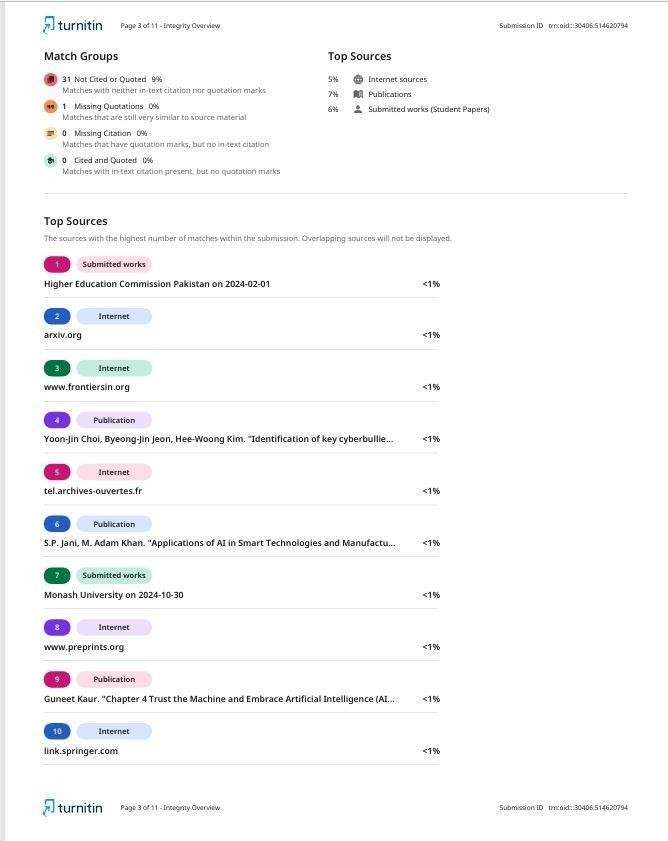
// Load once on page open loadStats();

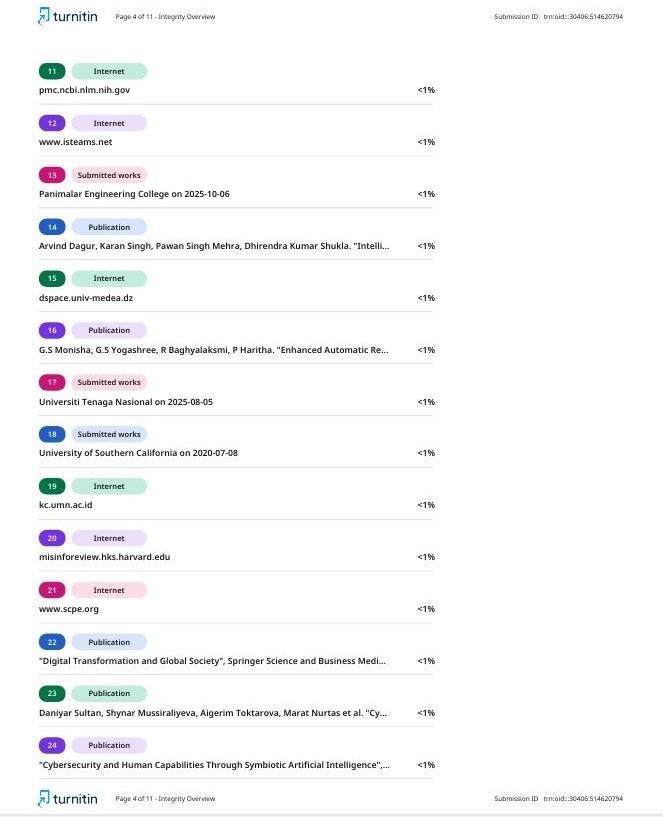
// Refresh every 5 seconds setInterval(loadStats, 5000);

## A.2 PLAGARISM REPORT

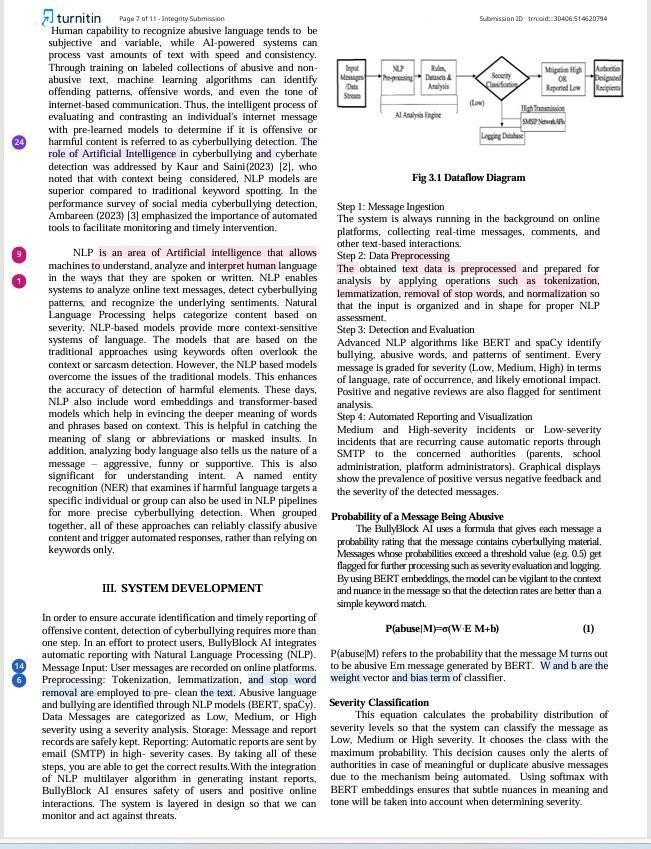
****

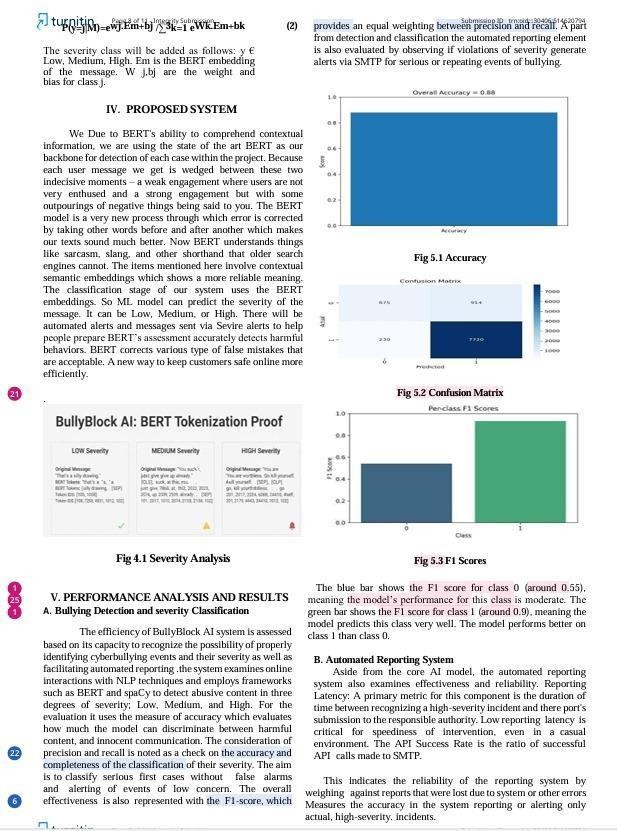


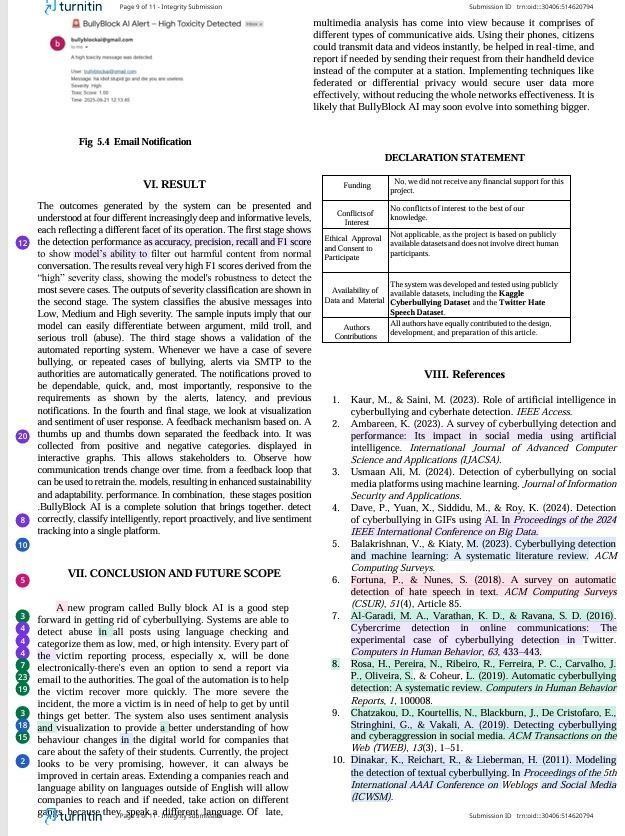


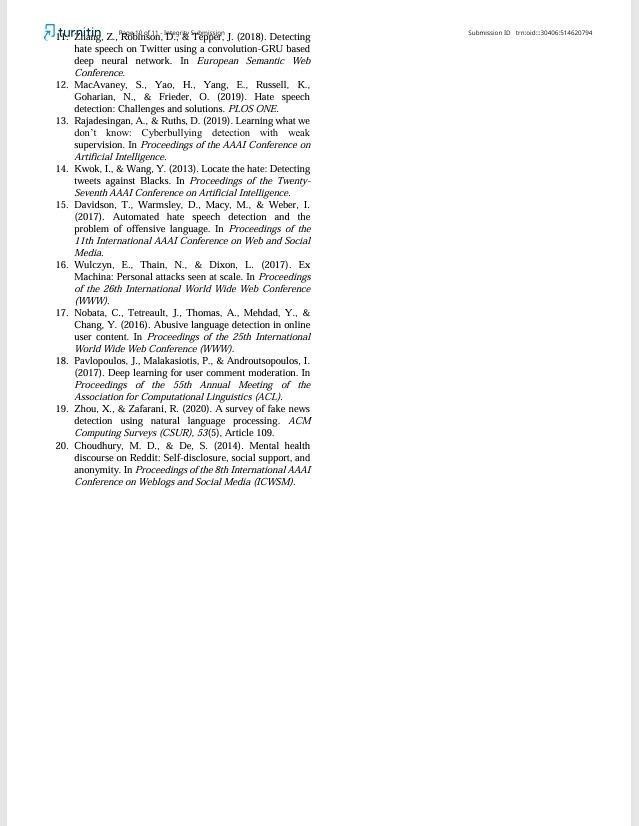












## REFERENCES

1. Kaur, M., & Saini, M. (2023). Role of Artificial Intelligence in Cyberbullying and Cyberhate Detection. *IEEE Access*.

2 . Ambareen, K. (2023). A Survey of Cyberbullying Detection and Performance: Its Impact in Social Media Using Artificial Intelligence.

1. Usmaan Ali, M. (2024). Detection of Cyberbullying on Social Media Platforms Using Machine Learning.
2. Dave, P., Yuan, X., Siddidu, M., & Roy, K. (2024). Detection of Cyberbullying in GIF Using AI.
3. Balakrishnan, V., & Kiaty, M. (2023). Cyberbullying Detection and Machine Learning: A Systematic Literature Review.
4. Arul Selvan, M. (2024). Automated Cyberbullying Detection Framework Using NLP and Supervised Machine Learning Models. *Journal of Science Technology and Research (JSTAR).*
5. Jamalpur, B., Kothandaraman, D., Navya, A., Keerthi, G. B., Srijanjali, D., & Sushmitha,

M. (2023). Social Harassments Cyberbullying Detection Using Machine Learning. *International Conference on Computational Intelligence and Communication (ICCIC).* Springer.

1. Begum, A. R. J., Steffy, D., & Monisha, B. (2025). Cyberbullying Detection on Social Media Using AI. *International Journal for Research in Applied Science and Engineering Technology (IJRASET).*
2. Anonymous. (2024). Detecting Cyberbullying Using Deep Learning Techniques: A Pre- trained GloVe and Focal Loss Technique. *PubMed.*
3. Anonymous. (2024). Cyberbullying Detection on Social Networks Comparing Machine Learning and Transfer Learning. *International Journal of Information Technology and Computer Engineering (IJITCE).*
4. Milosevic, T., Verma, K., Carter, C., Vigil, C., Laffan, D., Davis, B., & O’Higgins

Norman, J. (2023). Effectiveness of Artificial Intelligence–Based Cyberbullying Interventions From Youth Perspective. *Social Media + Society.*

1. Verma, K., Davis, B., & Milosevic, T. (2023). Examining the Effectiveness of Artificial Intelligence-Based Cyberbullying Moderation on Online Platforms: Transparency Implications. *AoIR Selected Papers of Internet Research.*
2. Lahby, M., Pathan, A. K., & Maleh, Y. (Eds.). (2024). *Combatting Cyberbullying in Digital Media with Artificial Intelligence.* Routledge.
3. Anonymous. (2024). Improved Hybrid Model for Robust Cyberbullying Detection.

*International Journal of Mathematics and Computer Research (IJMCR).*

1. Prama, T. T., Amrin, J. F., et al. (2025). AI Enabled User-Specific Cyberbullying Severity Detection with Explainability. *arXiv preprint.*
2. Liu, M. (2024). Network Bullying Detection Based on Deep Learning. *SAGE Journals.*
3. Anonymous. (2025). Cyberbullying Detection and Prevention Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology (IJRASET).*
4. Al-Harigy, R., et al. (2022). Building Towards Automated Cyberbullying Detection: A Comparative Analysis. *Computational Intelligence and Neuroscience.*
5. Vaddepelli, P., Jambe, S., & Gandhi, R. (2024). A Review of Deep Learning Approaches for Cyberbullying Detection: Focusing on LSTM Models. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT).*
6. Shah, V., et al. (2023). ML and Natural Language Processing: Cyberbullying Detection System for Safer and Culturally Adaptive Digital Communities. *Journal of Smart Internet of Things.*
7. Anonymous. (2022). Smart Cyberbullying Detection with Machine Learning. *Springer Conference Proceedings.*
8. Anonymous. (2023). A Review of Machine Learning Techniques in Cyberbullying Detection. *Computers, Materials & Continua (CMC).*
9. Anonymous. (2024). Cyberbullying Detection System on Social Media Using Supervised Machine Learning. *Procedia Computer Science.*
10. Anonymous. (2024). Uncovering Bullying on Social Media Platforms: A Comprehensive Study of Machine Learning Classifiers for Cyberbullying Detection. *Springer TEHI Conference Proceedings.*
11. Kumar, D. (2025). Cyberbullying Detection in Hinglish Text Using MURIL and Explainable AI. *arXiv preprint.*