International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.038 ISSN: 2581-4621

FIGURE IT OUT: ChatApp Using NLP Hate-speech Detection

Shaik.Meer Shubani Ali[1], Mohammed Shareef[2], G. Sujith Chowdary[3], M. Sri Varsha[4], T. Sri Veena Madhu[5]

[1]Associate Professor, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India
[2]Department of Computer Science and Engineering(AI&ML), Hyderabad Institute of Technology and Management, Hyderabad, India
[3]Department of Computer Science and Engineering(AI&ML), Hyderabad Institute of Technology and Management, Hyderabad, India
[4]Department of Computer Science and Engineering(AI&ML), Hyderabad Institute of Technology and Management, Hyderabad, India
[5]Department of Computer Science and Engineering(AI&ML), Hyderabad Institute of Technology and Management, Hyderabad, India

Abstract**--** The proposed project aims to develop a dynamic online knowledge-sharing platform specifically designed for students within our organization. This platform will serve as an interactive forum where students can post questions and provide answers, thereby fostering a collaborative learning environment. The main objective is to facilitate seamless communication and exchange of knowledge among students, enhancing their educational experience. A key feature of this platform is the integration of an automatic content moderation system. This system will scan responses for negative speech, and if detected, a dialogue box will prompt the user to rephrase their response. This ensures that all interactions on the platform remain constructive and respectful, maintaining a positive atmosphere conducive to learning. The platform will include user profiles, allowing students to manage their personal information and engage with content that aligns with their interests. The question-and-answer module will support various formatting options to improve readability and engagement.

Keywords**—ChatApp, Hate speech Detection, SVM NLP Algorithm, Tfidf Vectorizer,**

I. INTRODUCTION

Introducing our groundbreaking project: a fusion of cutting-edge technologies aimed at revolutionizing online communication while tackling the pervasive issue of hate speech. At its core, we are building a user-friendly Question and Answer platform tailored for college communities, facilitating knowledge exchange and information sharing. This endeavour seamlessly integrates web development, online chatApp functionality, and advanced Natural Language Processing (NLP) for hate speech detection. Through sophisticated algorithms and meticulous preprocessing, our system analyses textual inputs in real-time, swiftly identifying, and mitigating instances of hate speech. With an intelligently designed chatApp interface, users experience intuitive, interactive conversations, fostering a safer online environment and enabling seamless communication.

II. LITERATURE SURVEY

The literature survey on knowledge sharing and innovation, particularly in the context of building a documentation journal on chatApp development with NLP-based hate speech detection, encompasses various perspectives. Castaneda and Cuellar [1] (2020) explore the intricate relationship between knowledge sharing and innovation, emphasizing the importance of systematic approaches. Nguyen [2] (2020) introduces a fourdimensional model, shedding light on online organizational knowledge sharing dynamics, which could inform the structure of the documentation journal. Yao et al. [3] (2020)

delve into the knowledge sharing practices within Chinese software SMEs, offering insights into the technological innovation capabilities relevant to chatApp development. Sun et al. [4] (2019) provide a lens through which to examine the impact of enterprise social media platforms on knowledge sharing, offering potential strategies for leveraging such platforms in documenting chatbot building processes. Setialana et al. [5] (2021) present WeShare, a knowledge sharing platform that promotes freedom in learning, which could serve as a model for structuring the documentation journal to encourage collaborative knowledge dissemination. Rao and Kumar [6] (2021) discuss societal communication dynamics within Q&A communities, offering potential insights into fostering engagement within the documentation journal community. Furthermore, Prabowo and Herwanto [7] (year) propose a method for duplicate question detection in Q&A websites, which could enhance the efficiency of information retrieval within the documentation journal. Additionally, external resources such as the study by Lin et al. [8] (2008) on sentiment analysis using machine learning techniques and the work by Mikolov et al. [9] (2015) on word embeddings provide foundational knowledge in natural language processing, crucial for hate speech detection in the chatbot context. Lastly, the study by Fortuna et al. [10] (2019) highlights the importance of ethics in AI development, offering guidance on responsible deployment of chatApp equipped with hate speech detection capabilities.

III. METHODOLOGY

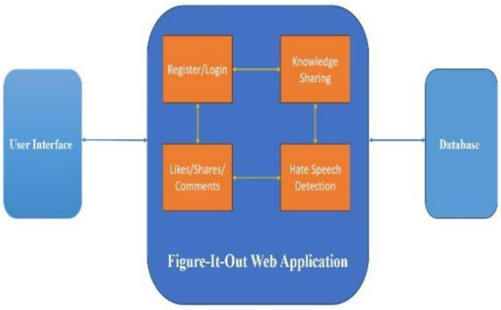


Fig. 1

The methodology for building a chatApp equipped with NLP-based hate speech detection using the SVM algorithm involves several key steps. Firstly, data collection is paramount. We gather a diverse dataset comprising both normal and hate speech-laden conversational texts, ensuring it represents various linguistic patterns and hate speech nuances.

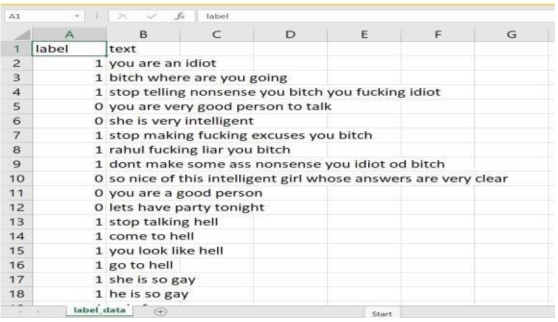
Preprocessing follows, involving tokenization, stop word removal, and stemming or lemmatization to standardize the text data. Feature extraction is then conducted, leveraging techniques like TF-IDF (Term FrequencyInverse Document Frequency) to transform the textual data into numerical feature vectors, capturing the essence of each text snippet. Next, model selection becomes crucial. We opt for the Support Vector Machine (SVM) algorithm due to its effectiveness in binary classification tasks like hate speech detection. The SVM model is trained on the pre-processed and feature-extracted dataset, utilizing both the textual features and corresponding labels (i.e., hate speech or non-hate speech). During training, hyperparameters tuning is performed to optimize the model's performance, employing techniques such as cross-validation to prevent overfitting and ensure generalizability. Once the SVM model is trained and validated, integration with the chatApp framework takes place. The chatApp interface is designed to interact seamlessly with users, processing their input texts and passing them through the SVM hate speech detection model. Upon detection of hate speech, appropriate actions are taken, such as flagging the message for review or providing educational responses to discourage hate speech. Furthermore, continuous monitoring and evaluation are essential components of the methodology. The performance of the chatAppand hate speech detection model is assessed regularly using metrics like accuracy, precision, recall, and F1-score. Feedback from users is also collected and analysed to iteratively improve the chatApp functionality and enhance hate speech detection accuracy. Lastly, deployment and maintenance aspects are considered. The chatApp with NLP-based hate speech detection is deployed on suitable platforms to make it accessible to users. Regular updates and retraining of the hate speech detection model are scheduled to adapt to evolving linguistic patterns and emerging hate speech trends. Additionally, ethical considerations are prioritized throughout the project to ensure responsible deployment and usage of the chatApp, safeguarding against unintended biases or discriminatory outcomes.

System Architecture:

The hate speech detection system comprises several key modules. Firstly, the Web Interface provides a user-friendly platform where users can input text for analysis. This interface facilitates easy interaction and seamless input provision for hate speech detection. Following input, the Preprocessing Module comes into play, conducting essential tasks such as tokenization, lowercasing, and removing stop words to standardize the text data. Subsequently, the Feature Extraction module transforms the preprocessed text into numerical features suitable for machine learning algorithms. Techniques like TF-IDF (Term Frequency-Inverse Document Frequency) are commonly employed for this purpose, capturing the essence of each text snippet effectively. The core of the system lies in the Support Vector Machine (SVM) Model, which is trained on labeled data to classify text into hate speech or non-hate speech categories. This model utilizes the numerical features extracted earlier to make accurate classifications. Integration becomes pivotal as the Web Interface and NLP model are seamlessly connected using an NLP (H5 File), ensuring smooth communication between the user and the hate speech detection system. Once the classification is performed, the Output Module comes into action, displaying the classification result (hate speech or non-hate speech) to the user via the web interface. If the text is classified as non-hate speech, the system initiates knowledge interaction, providing informative responses or engaging in further dialogue to enhance user experience and foster constructive communication.

Model Dataset:

The model utilized a custom dataset with two categories: 0 for non-hate speech and 1 for hate speech. This dataset was carefully curated and annotated to ensure accuracy. By leveraging this balanced dataset, model aimed to train a robust model to discern hate speech from non-hate speech effectively. The inclusion of diverse textual expressions facilitated the development of an accurate classification system, crucial for combating online hate speech.



	A	B
1	label	text
2	1	you are an idiot
3	1	bitch where are you going
4	1	stop telling nonsense you bitch you fucking idiot
5	0	you are very good person to talk
6	0	she is very intelligent
7	1	stop making fucking excuses you bitch
8	1	rahul fucking liar you bitch
9	1	dont make some ass nonsense you idiot od bitch
10	0	so nice of this intelligent girl whose answers are very clear
11	0	you are a good person
12	0	lets have party tonight
13	1	stop talking hell
14	1	come to hell
15	1	you look like hell
16	1	go to hell
17	1	she is so gay
18	1	he is so gay

Fig. 2

INTEGRATION OF SVM ALGORITHM MODEL AND CHATBOT:

The process of building and deploying a Support Vector Machine (SVM) model for hate speech detection involves several sequential steps. Initially, data preparation is crucial, involving the acquisition of a labelled dataset containing examples of text categorized as either 'hate speech' or 'non-hate speech'. Each example is represented as a feature vector, often derived from techniques like TFIDF values or word embeddings. Subsequently, the dataset is split into training and test sets for model training and evaluation, respectively. Feature extraction techniques, such as TF-IDF, are applied to convert the raw text data into numerical features understandable by the SVM model. The

SVM model is then trained using the training set, aiming to learn the optimal hyperplane that separates examples of different classes in the feature space while maximizing the margin between support vectors. Hyperparameter tuning is conducted to optimize model performance by selecting appropriate kernel functions and tuning parameters like the regularization parameter and kernel coefficient. Model evaluation is carried out using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC score to assess its generalization ability and classification accuracy on unseen data. Once trained and evaluated, the model can predict the labels of new text examples as 'hate speech' or 'non-hate speech'. Finally, if the model meets performance criteria, it can be deployed for real-world applications, possibly integrated into web applications, chatApp, or other systems requiring hate speech detection capabilities. Through these systematic steps, an SVM model effectively learns from data and makes predictions regarding hate speech in textual data. Integration with a chatApp entails incorporating the SVM hate speech detection model seamlessly into the chatApp framework to enhance its functionality. Once the SVM model is trained and evaluated, it becomes an integral component of the chatApp backend architecture. User input received by the chatApp is first passed through the hate speech detection model for analysis. If the text is classified as hate speech, appropriate actions are taken within the chatApp interface, such as flagging the message for review or providing warnings to users. Conversely, if the text is classified as non-hate speech, the chatApp proceeds with its intended interaction, providing responses or carrying out tasks as programmed. This integration enhances the chatApp capabilities by enabling it to detect and respond to instances of hate speech, contributing to a safer and more inclusive online environment. Additionally, continuous monitoring and refinement of the integration ensure optimal performance and adherence to ethical guidelines throughout the chatApp operation.

RESULT AND DISCUSSION:

The evaluation results depict exemplary performance of an NLP-based hate speech detection model on a given dataset. With an accuracy score of 1.0, the model demonstrates flawless classification, correctly identifying all instances as either 'hate speech' or 'non-hate speech'. Precision scores of 1.0 for both classes indicate that all classifications made by the model were accurate, with no false positives. Similarly, a recall score of 1.0 reflects the model's ability to capture all instances of both classes, achieving perfect sensitivity. The confusion matrix further illustrates this, showing no misclassifications with a diagonal arrangement of true positives and true negatives. In summary, these evaluation metrics affirm the robustness and reliability of the NLP-based hate speech detection model in accurately identifying instances of hate speech within the dataset.

PERFORMANCE METRICES:

```
Accuracy: 1.0,
 Recall: 1.0
F1-Score: 1.0
Precision: 1.0

Confusion Matrix:
[[ 98   0]
     [ 0 113]]

Classification Report:
          precision    recall f1-score   support

        0      1.00      1.00      1.00       98
```

```
           1       1.00      1.00      1.00       113

    accuracy                           1.00       211
   macro avg       1.00      1.00      1.00       211
weighted avg       1.00      1.00      1.00       211
```

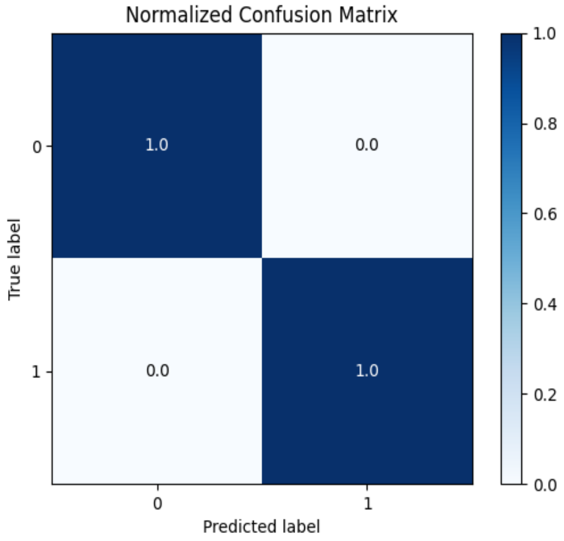


Fig. 3

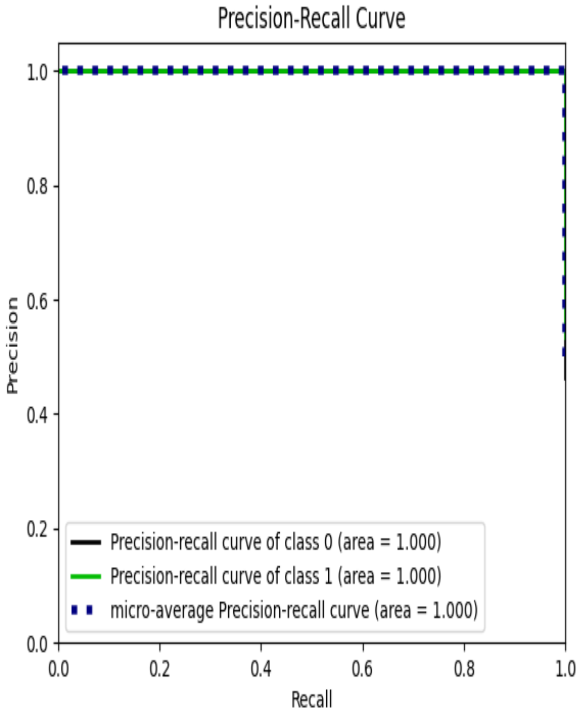


Fig. 4

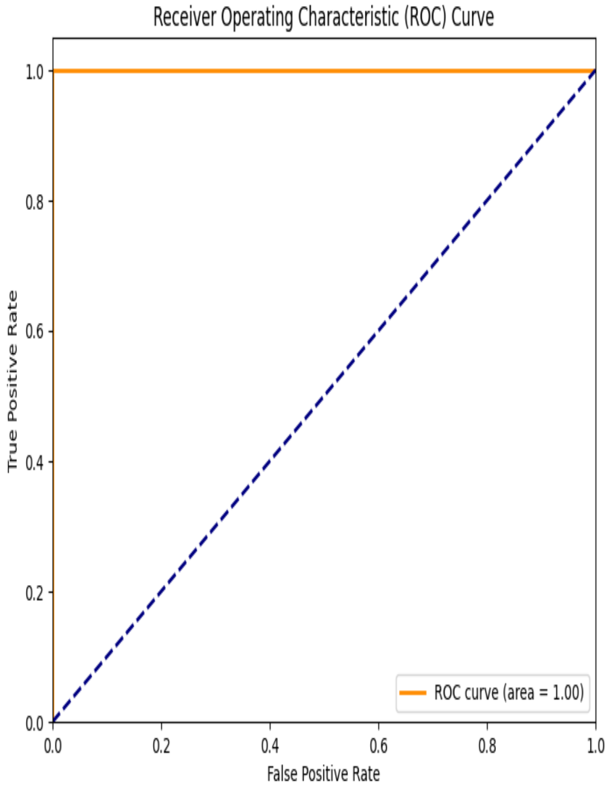
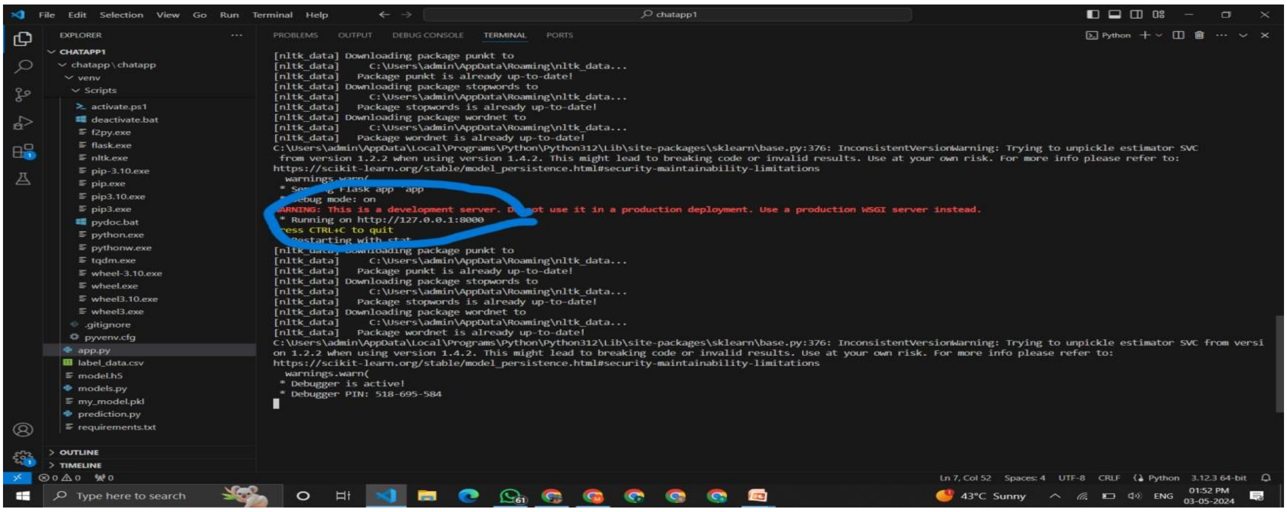


Fig. 5

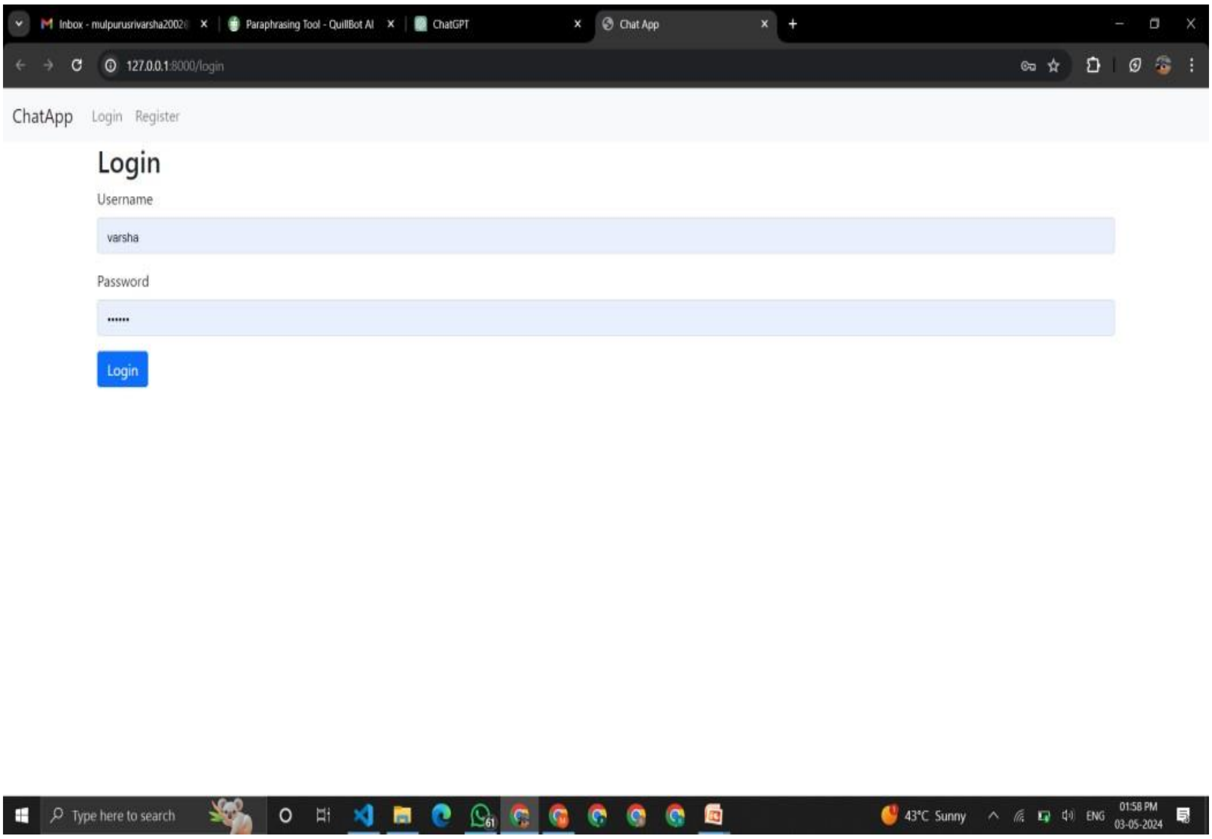
Running The ChatApp And NLP Trained Model On HTTP Protocol:



```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
C:\Users\admin\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning: Trying to unpickle estimator SVC
 from version 1.2.2 when using version 1.4.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:8000
Press CTRL+C to quit
 * Restarting with stat
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
C:\Users\admin\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\base.py:376: InconsistentVersionWarning: Trying to unpickle estimator SVC from versi
on 1.2.2 when using version 1.4.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/model_persistence.html#security-maintainability-limitations
  warnings.warn(
 * Debugger is active!
 * Debugger PIN: 518-695-584
```

Fig. 6

ChatApp Login:



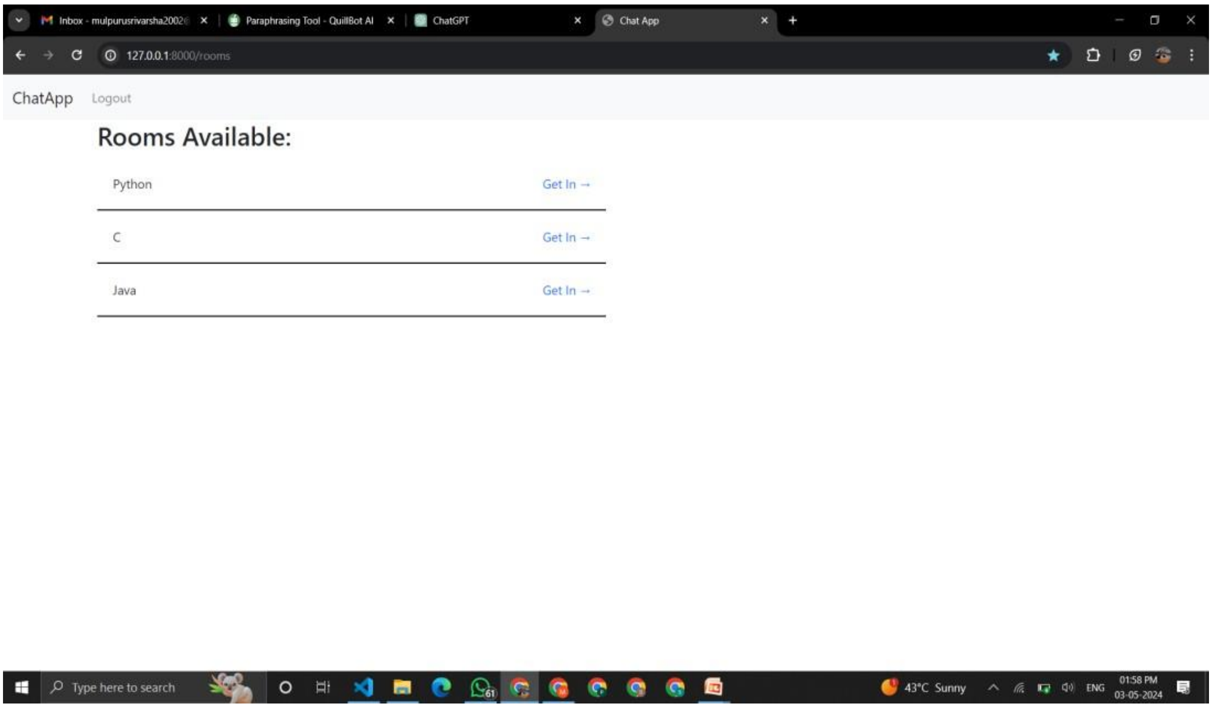
Login

Username: varsha

Password: ••••••

Fig. 7

Get In The Rooms Available In ChatApp:

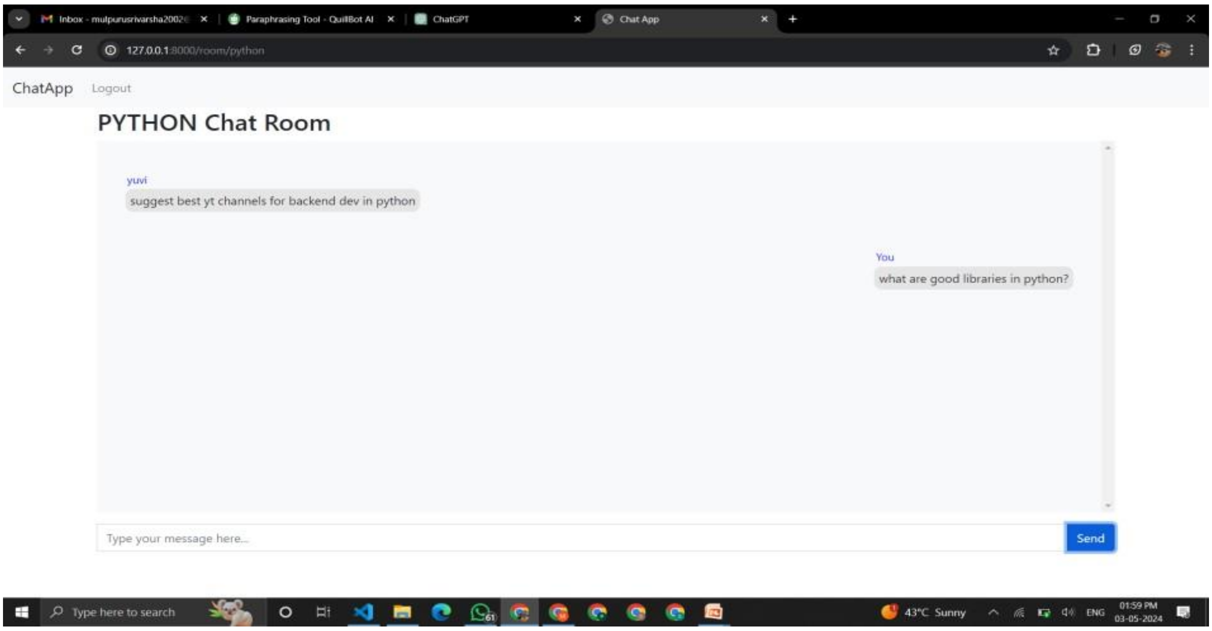


Rooms Available:

Python	Get In →
C	Get In →
Java	Get In →

Fig. 8

Chat Room Interaction By Knowledge Sharing:


Inbox - mulpurusrivarsha2002
Paraphrasing Tool - QuillBot AI
ChatGPT
Chat App
127.0.0.1:8000/room/python
ChatApp
Logout
Send
Type here to search
43°C Sunny
ENG
01:59 PM
03-05-2024

PYTHON Chat Room

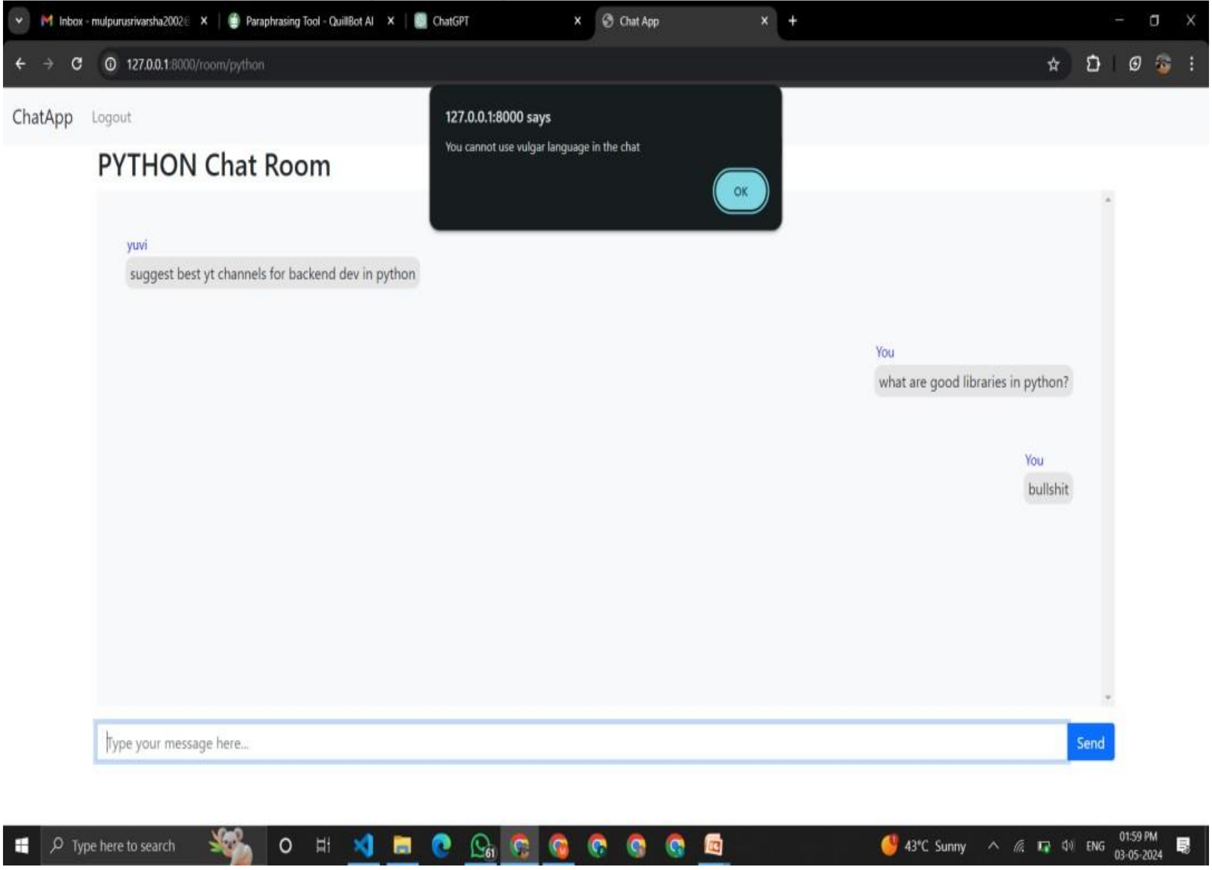
yuvi
suggest best yt channels for backend dev in python

You
what are good libraries in python?

Type your message here...

Fig. 9

Detection Of Hate Speech Using SVM Algorithm And ChatApp Interface:


Inbox - mulpurusrivarsha2002
Paraphrasing Tool - QuillBot AI
ChatGPT
Chat App
127.0.0.1:8000/room/python
ChatApp
Logout
OK
Send
Type here to search
43°C Sunny
ENG
01:59 PM
03-05-2024

127.0.0.1:8000 says
You cannot use vulgar language in the chat

PYTHON Chat Room

yuvi
suggest best yt channels for backend dev in python

You
what are good libraries in python?

You
bullshit

Type your message here...

Fig. 10

CONCLUSION:

In conclusion, our project represents a significant advancement in leveraging technology to address complex societal challenges. By combining web development with an online chatApp interface and integrating a state-of-the-art Natural Language Processing (NLP) model for hate speech detection, we have created a multifaceted solution with far-reaching implications. Through the seamless integration of these components, we have not only facilitated real-time communication but also empowered users to navigate online spaces with greater safety and confidence. Our system's ability to swiftly identify and mitigate instances of hate speech underscores its potential to foster inclusivity and promote respectful discourse in digital environments. Moving forward, continued refinement and optimization of our model, coupled with ongoing advancements in web development and conversational AI, hold the promise of further enhancing our project's impact and effectiveness in combating online hate speech.

REFERENCES:

[1] D. I. Castaneda and S. Cuellar, "Knowledge sharing and innovation: A systematic review," Knowledge and Process Management, vol. 27, p. 159–173, 2020.
[2] T.-M. Nguyen, "Four-dimensional model: a literature review in online organisational knowledge sharing," VINE Journal of Information and Knowledge Management Systems, 2020.
[3] J. Yao, A. Crupi, A. Di Minin and X. Zhang, "Knowledge sharing and technological innovation capabilities of Chinese software SMEs," Journal of Knowledge Management, 2020.
[4] Y. Sun, X. Zhou, A. Jeyaraj, R.-A. Shang and F. Hu, "The impact of enterprise social media platforms on knowledge sharing: An affordance lens perspective," Journal of Enterprise Information Management, 2019.
[5] P. Setialana, A. Fitria, R. Atika, R. N. Fadilla and others, "Development of WeShare As a Knowledge Sharing Platform to Realize the Freedom in Learning," in Journal of Physics: Conference Series, 2021.
[6] P. V. Rao and A. P. S. Kumar, "The societal communication of the Q&A community on topic modeling," The Journal of Supercomputing, vol. 78, p. 1117–1143, June 2021.
[7] D. A. Prabowo and G. B. Herwanto, "Duplicate Question Detection in Question Answer Website using Convolutional Neural
[8] https://link.springer.com/article/10.1007/s11280-008-0051-3[9]https://arxiv.org/abs/1506.04111
[9] https://arxiv.org/abs/1506.04111
[10]https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0221152#sec020