**Toxic Comments Classification In Social Networking**

**Submitted By**

**KADALI JAGADEESH**

**39110424**

**INDEX**

**1.Introduction**

1. Overview
2. Purpose
3. **Literature Survey**
   1. Existing Problem
   2. Proposed solution
4. **Theoretical Analysis**
   1. Block Diagram
   2. Hardware/Software designing
5. **Experimental investigations**
6. **Flowchart**
7. **Result**

**7.Advantages & Disadvantages**

1. **Applications**
2. **Conclusion**
3. **Future Scope**
4. **Bibliography**

**12.Appendix**

* 1. Source Code

b. UI output screenshot

**1. INTRODUCTION**

1. **Overview:**

Nowadays, the flow of data over the internet has grown dramatically, especially with the appearance of social networking sites. Social networks sometimes become a place for threats, insults, and other components of cyberbullying. A huge number of people are involved in online social networks.

Toxic comments are textual comments with threats, insults, obscene, racism, etc. In recent years there have been many cases in which authorities have arrested some users of social sites because of the negative (abusive) content of their personal pages. Hence, the protection of network users from anti-social behaviour is an important activity. One of the major tasks of such activity is automated detecting the toxic comments.

The project aims to build a multi-headed model that’s capable of detecting different types of of toxicity like threats, obscenity, insults, and identity-based hate, we will use a dataset of comments from Wikipedia’s talk page edits, collected by Kaggle. Improvements to the current model

1. **Purpose:**

We’ll be able to understand the problem to classify if it is a toxic comment or a normal comment. We will be able to know how to pre-process/clean thedata using different data pre-processing techniques. You will able to analyse or getinsights into data through visualization. Applying different algorithms according tothe dataset and based on visualization. We will be able to know how to build a webapplication using the Flask framework.

**2. LITERATURE SURVEY**

### a.Existing problem :

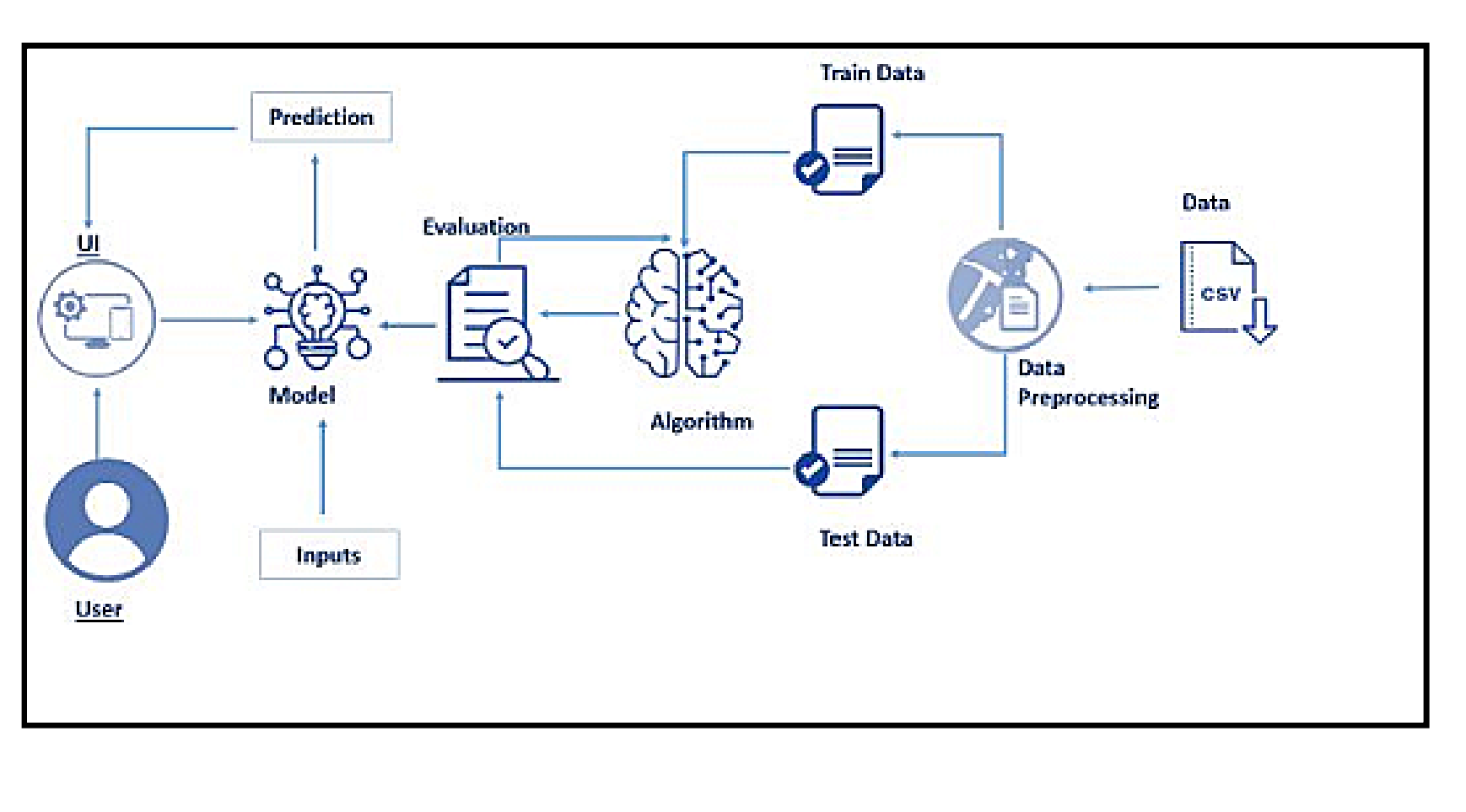
Bag of words statistics and bag of symbols statistics are the typical source information for the toxic comment’s detection. Usually, the following statistics-based features are used: length of the comment, number of capital letters, number of exclamation marks, number of question marks, number of spelling errors, number of tokens with non-alphabet symbols, number of abusive, aggressive, and threatening words in the comment, etc. A neural network model is used to classify the comments.

b. Proposed Solution :

The goal is to create a classifier model that can predict if input text is inappropriate (toxic). Explore the dataset to get a better picture of how the labels are distributed, how they correlate with each other, and what defines toxic or clean comments. Create a baseline score with a simple logistic regression classifier.Explore the effectiveness of multiple machine learning approaches and select the best for this problem.Select the best model and tune the parameters to maximize performance. Build a the final model with the best performing algorithm and parameters and test it on a holdout subset of the data.

**2. THEORETICAL ANALYSIS**

**3.1 Block Diagram**



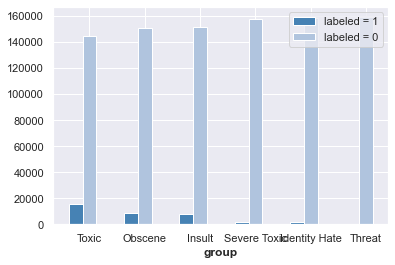
**b. Hardware/Software Requirements**

1. IDE: Jupyter Notebook, Spyder, Anaconda navigator
2. Programing Language (Back-end): Python 3.7
3. Front-end: HTML, CSS
4. Framework: Flask

**4. EXPERIMENTAL INVESTIGATIONS**

Data Exploration This dataset contains 159,571 comments from Wikipedia. The data consists of one input feature, the string data for the comments, and six labels for different categories of toxic comments: toxic, severe toxic, obscene, threat, insult, and identity hate. The figure on the following page contains a breakdown of how the labels are distributed throughout the dataset, including overlapping data. As you can see in the breakdown, while most comments with other labels are also toxic, not all of them are. Only “severe toxic” is clearly a subcategory of “toxic.” And it’s not close enough to be a labelling error. This suggests that “toxic” is not a catch-all label, but rather a subcategory in itself with a large amount of overlap. Because of this, I’m going to create a seventh label called “any\_label” to represent overall toxicity of a comment. From here on in, I’m going to refer to any labelled comments as toxic, and the specific “toxic” label (along with other labels) in quotation marks.



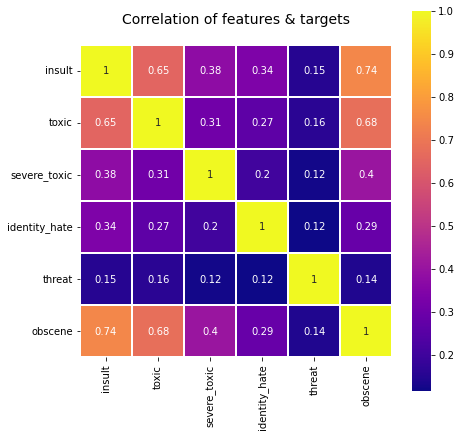




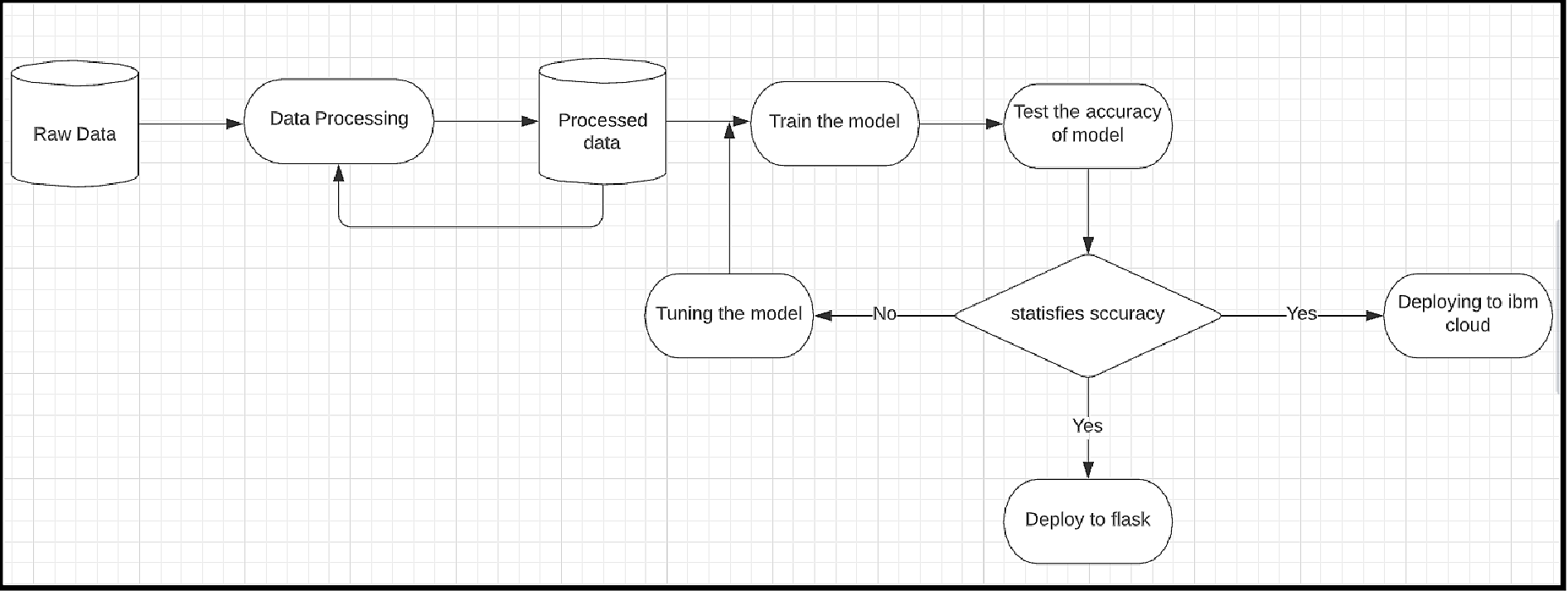
Only 39% of the toxic comments have only one label, and the majority have some sort of overlap. I believe that because of this, it will be much more difficult to train a classifier on specific labels than whether or not they are toxic. This ambiguity and the lack of explanation around it is what led me to select an aggregate label of general toxicity, what I’ve called “any\_label,” as the target. 16225 out of 159571 comments, or 10.17%, are classified as some category of toxic.

The correlation matrix below provides more insight into these overlapping categories. Threats are not likely to be severely toxic, nor are they likely to be racist or homophobic. But insults are often obscene, and identity hate really doesn’t have much overlap at all.

I believe the categories with significant overlap will be more difficult to predict, as they’ll have similar contributing features, but “identity\_hate” will have more unique attributes and be easier to predict.

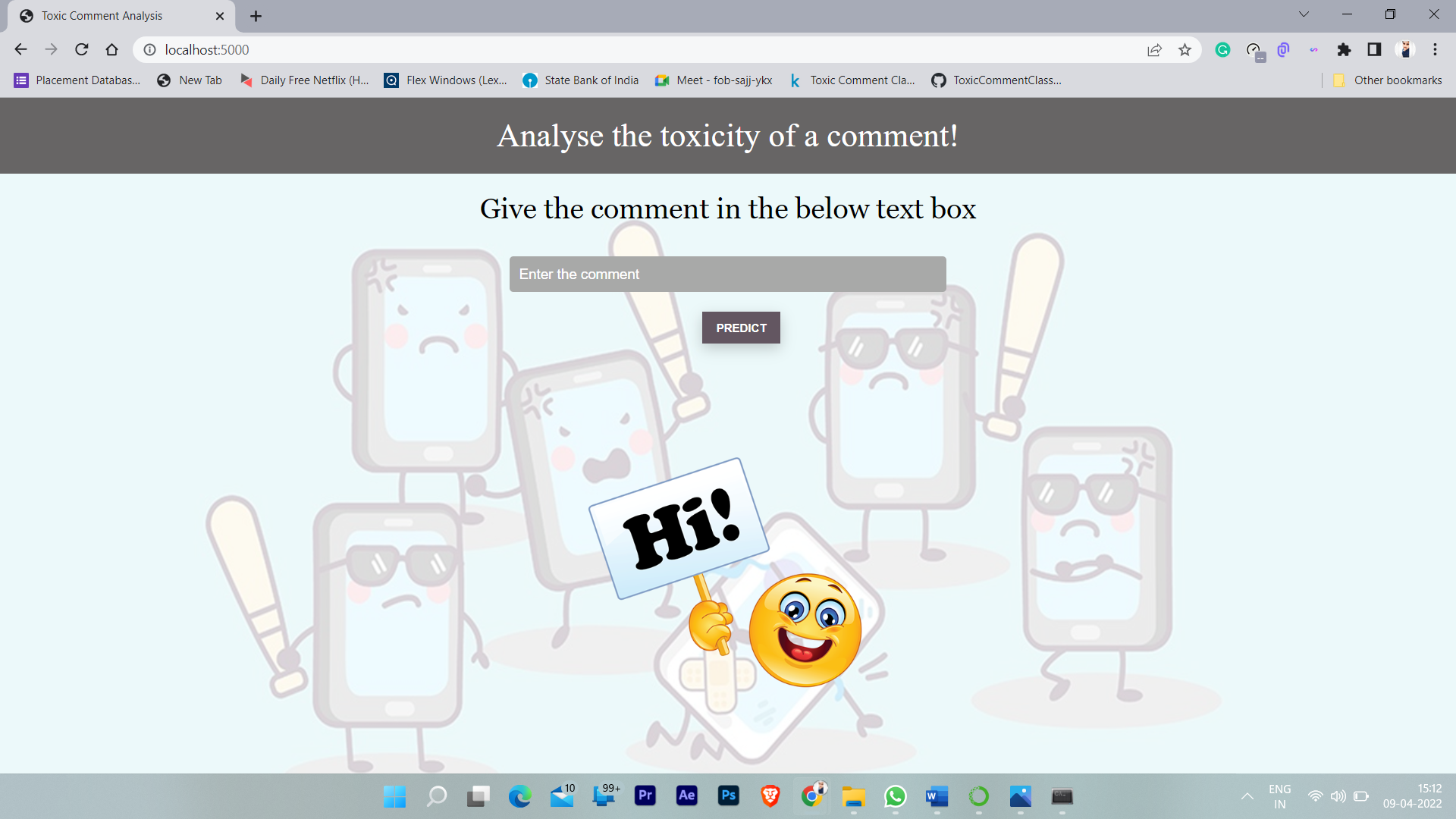


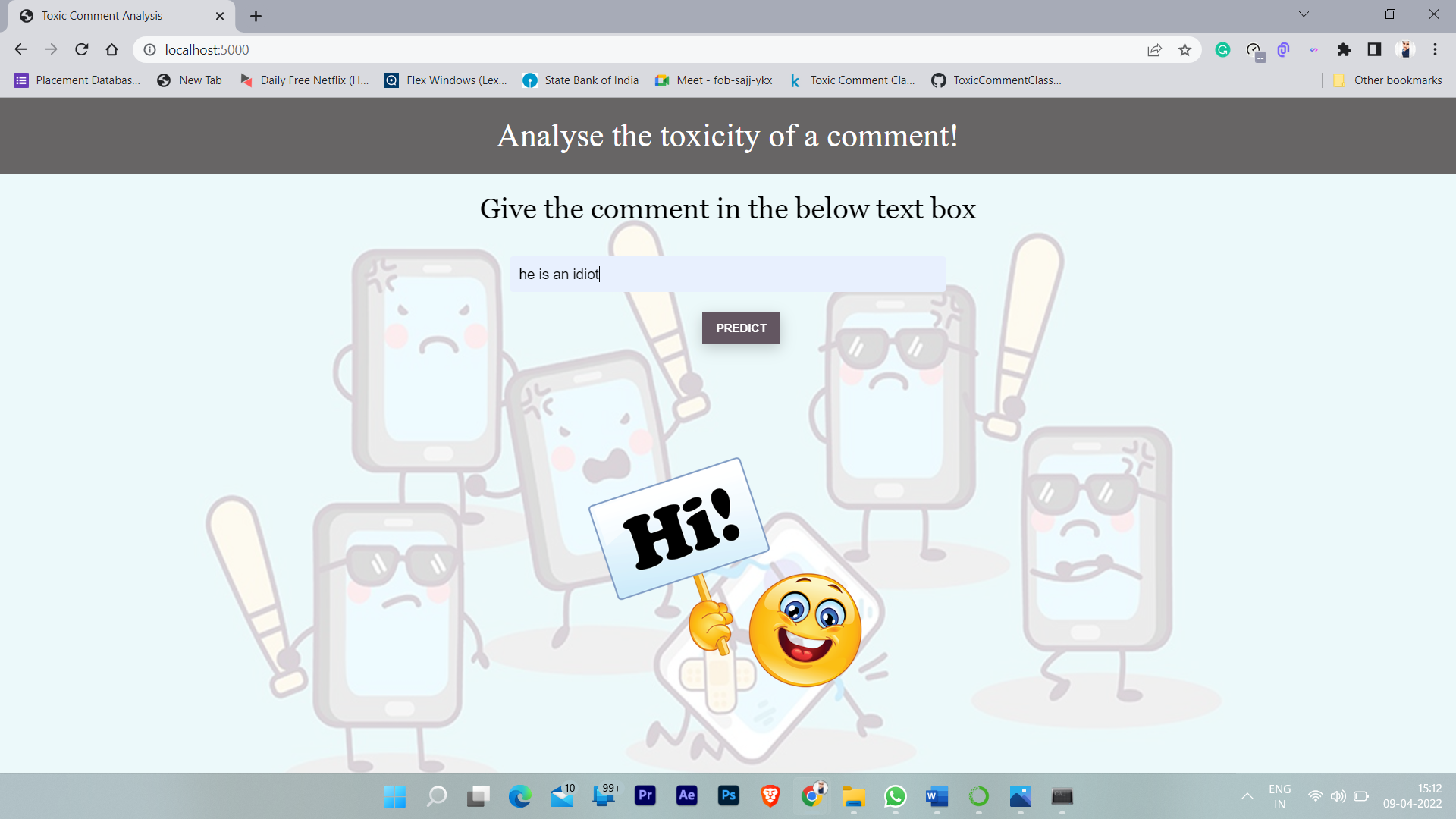
**FLOWCHART**



**6. RESULT**

We have successfully built the UI interface for classification of toxic comments, in which user can input any comment and then while pressing predict the user can find the comment is toxic or clean , this UI will help them to find the comment is toxic or clean .





**7. ADVANTAGES**

1. The interface is user friendly and easy to use and understand even to a person who has very little/no knowledge .

2. This model uses a huge amount of data the results generated are accurate and reliable.

3. Easy to use & has a user-friendly interface.

4. Results can be improved by training data to our choice of parameter.

5. We can easily do the classification of different comments whether the comment is toxic or clean using natural language processing.

**DISADVANTAGES**

1. we can't predict the comment if it is different language
2. the modal can't determine properly if the comment is big

**8. APPLICATIONS**

1. using this modal we can identify the toxic comments
2. Time saving & cost-efficient method as we don t need to read every comment

by using this modal we can skip the toxic comments

1. As the same way we can find the person who handles the toxic comments and we can block them
2. using this modal we can clear the unwanted and toxic comments
3. By clearing the waste/toxic data we can reduce the data storage.
4. As this modal is reliable we can use in different sectors to classified the comments.

**9. CONCLUSION**

A Machine Learning model, has been developed to **classification of toxic comments in social networking** . A simple, efficient and a versatile model is built keeping in mind the diversity of data, computational complexity and overhead involved in making API calls to the model for prediction. The product can increase the accuracy of finding the toxic comments in social network. It helps to classified the comments and identify the toxic comments.

**10.FUTURE SCOPE**

1. Recurrent neural networks, despite their increased overhead, could be a very effective solution if GPU resources are available for quick predictions

2. We could use a simple decision tree to feel comments into the model that would be most effective. A few that I can think of are:

* Short comments
* Long comments
* “Hot” threads where the rate of commenting is high and emotions may be high

**11. BIBLIOGRAPHY**

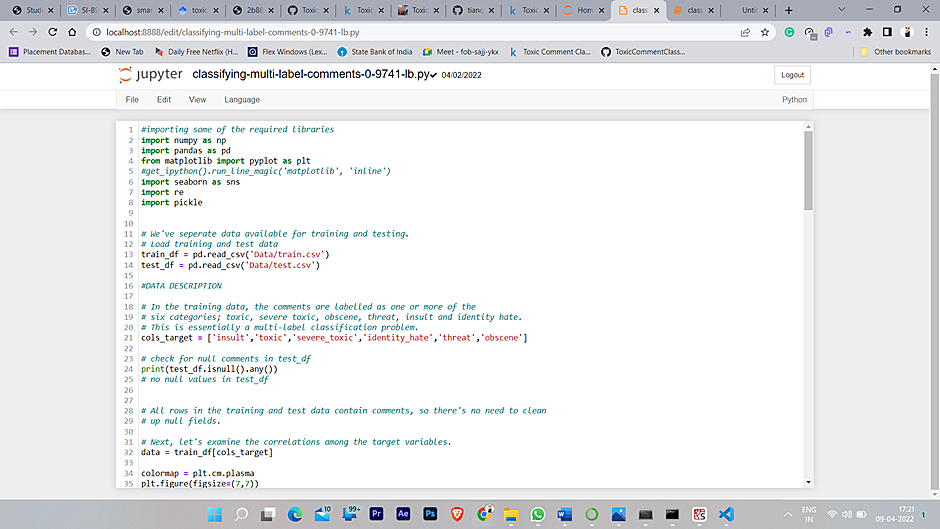
* DATASET--**<https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data>**

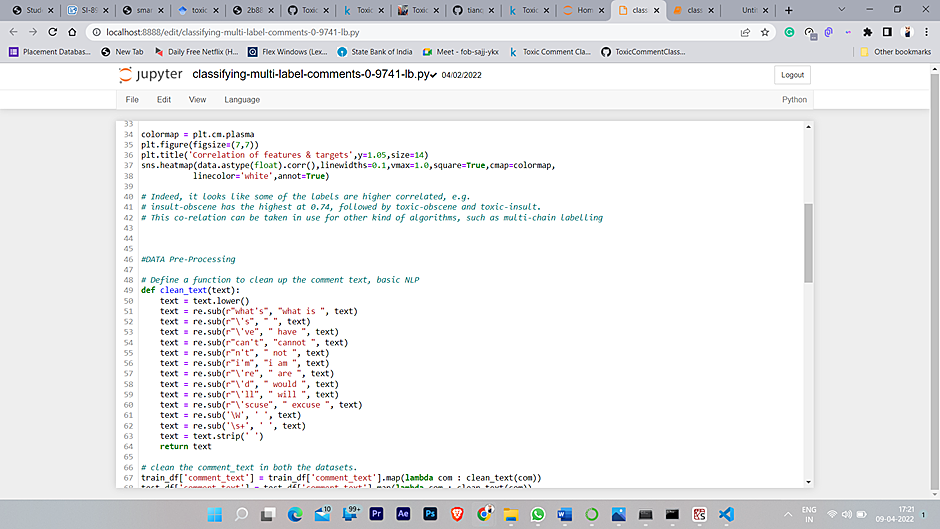
1. Toxic Comment Classification Group Project for MSDS621 Machine Learning atUniversity of San Francisco

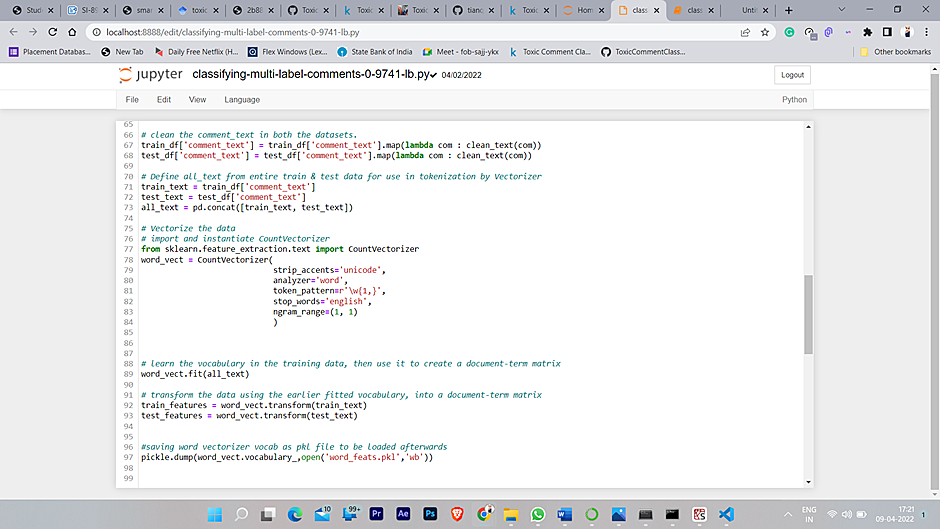
**12. APPENDIX**

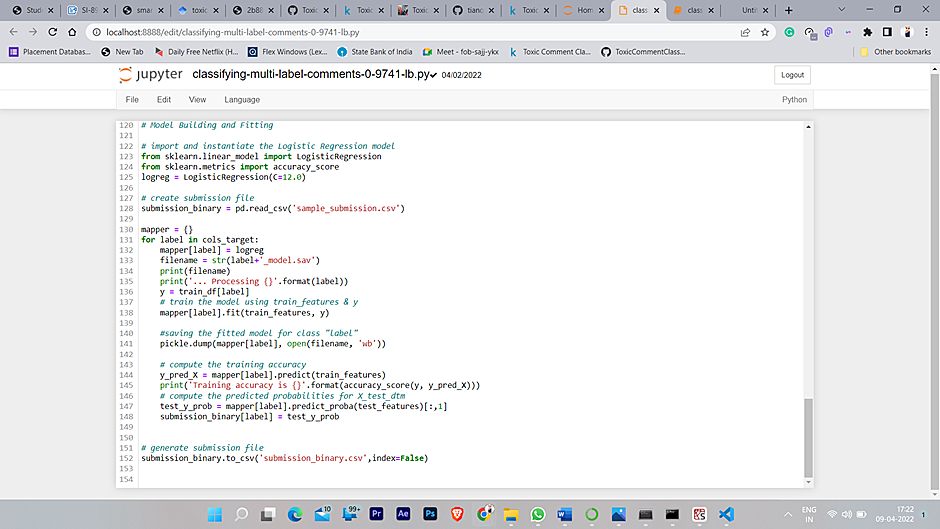
**a. Source code:**

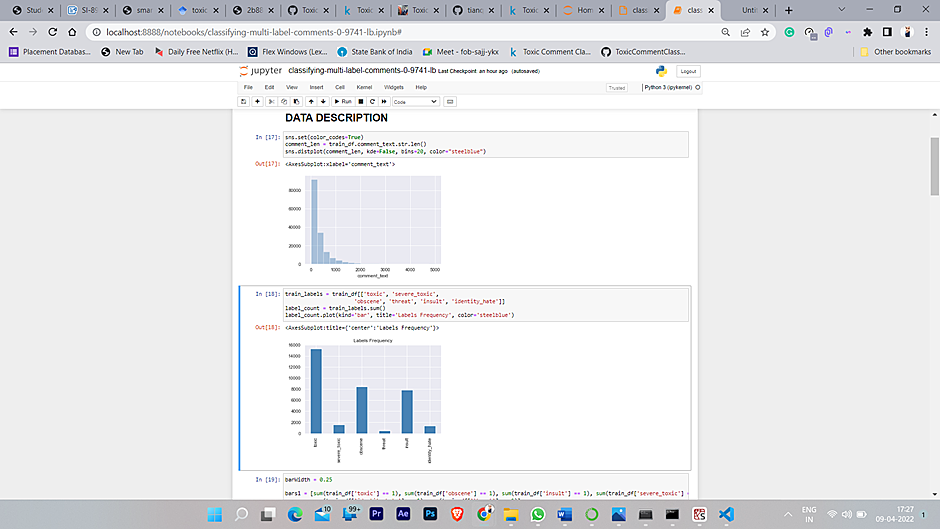
**classifying-multi-label-comments-0-9741-lb.py**

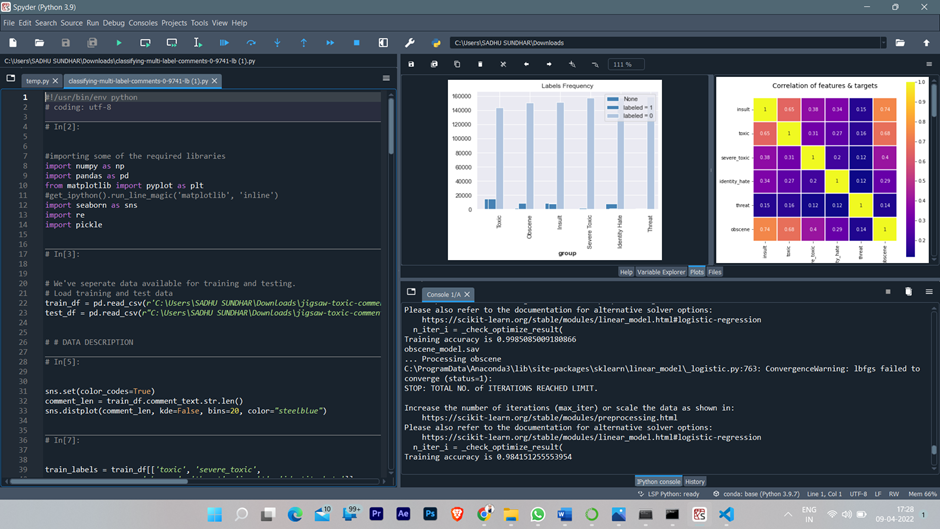




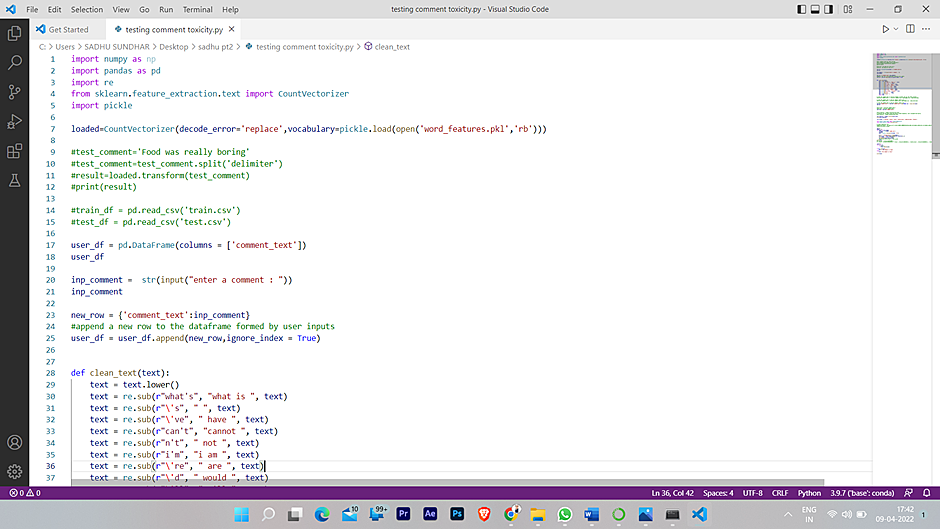


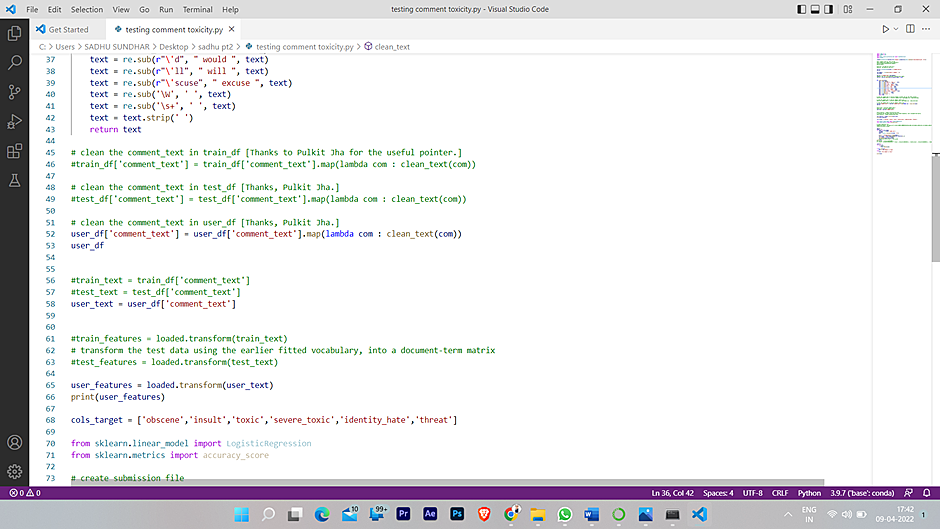


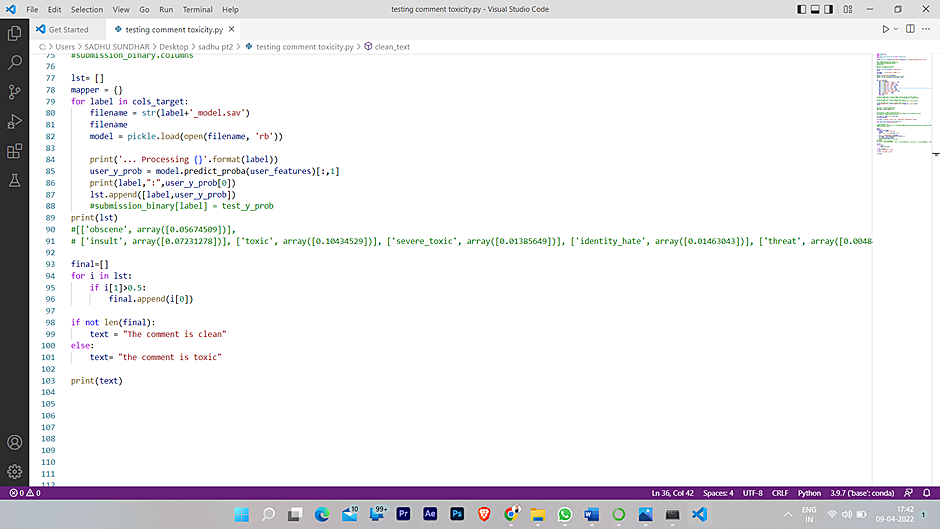


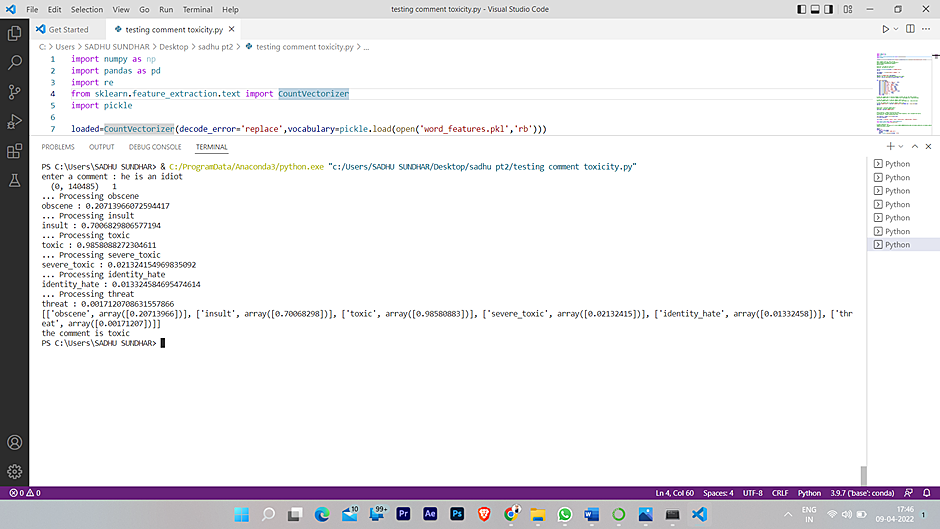


**testing comment toxicity.py**

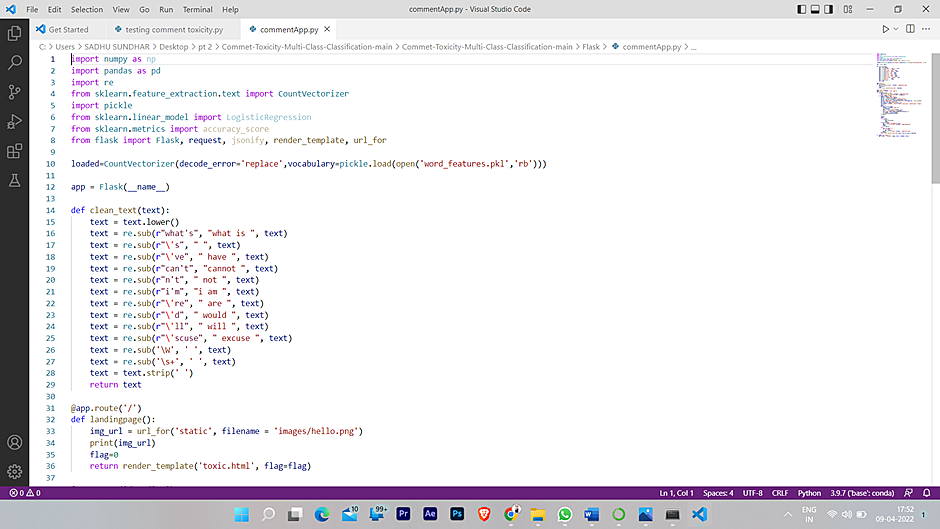


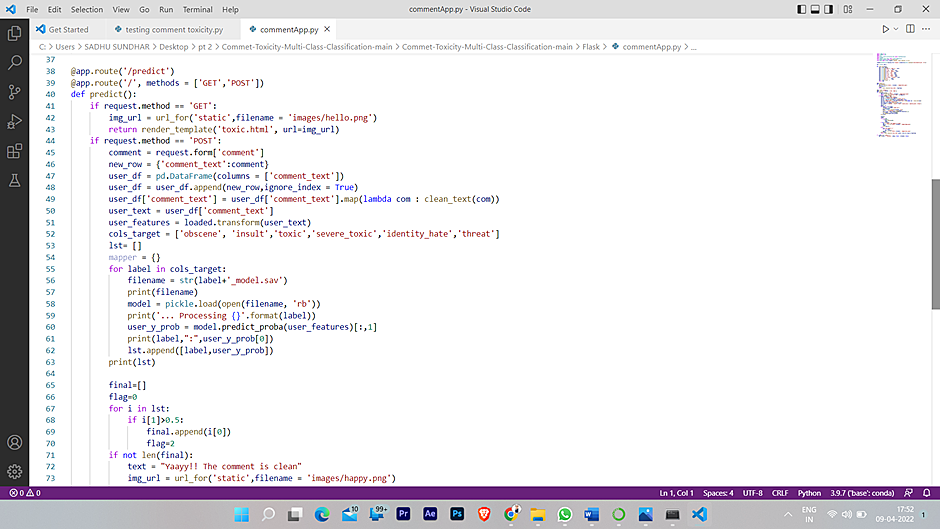


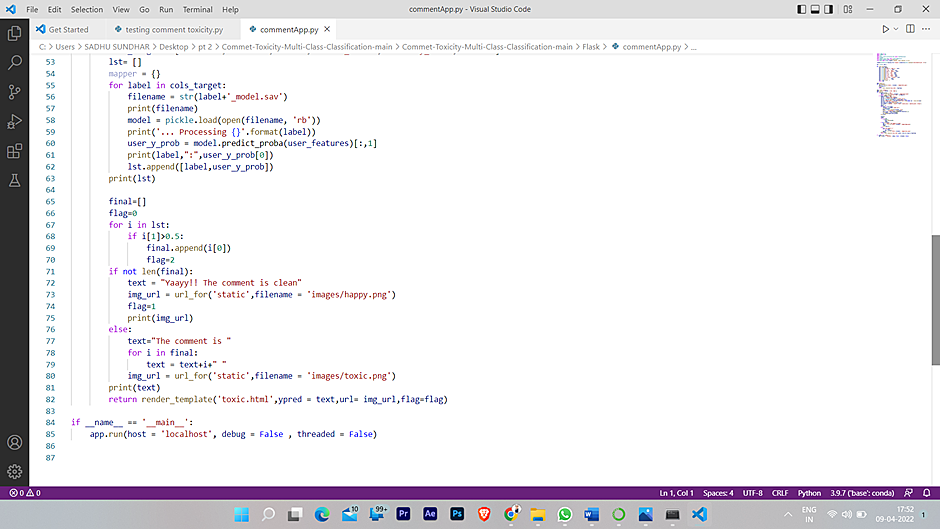


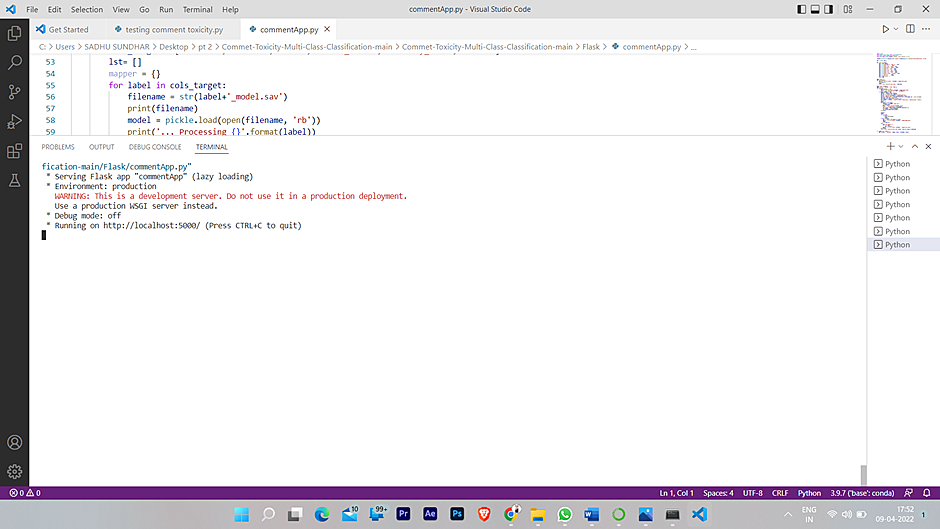


**commentApp.py**



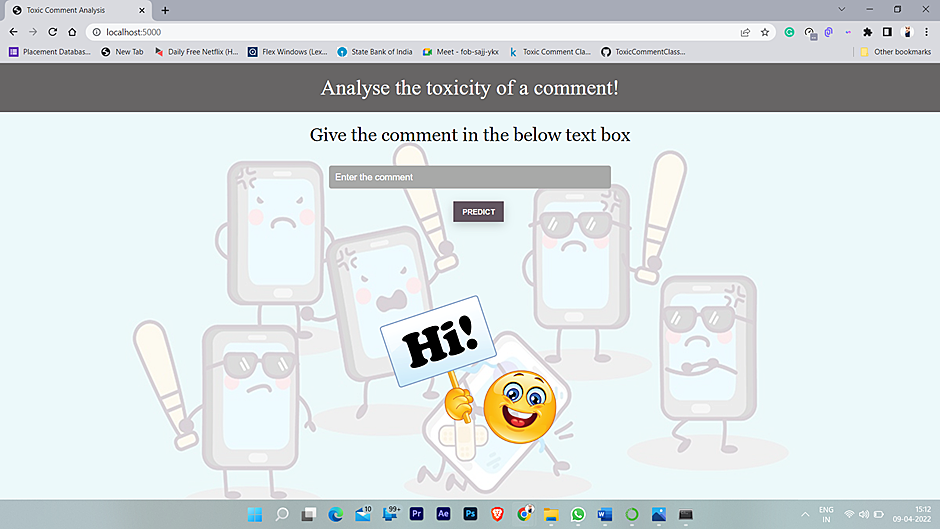




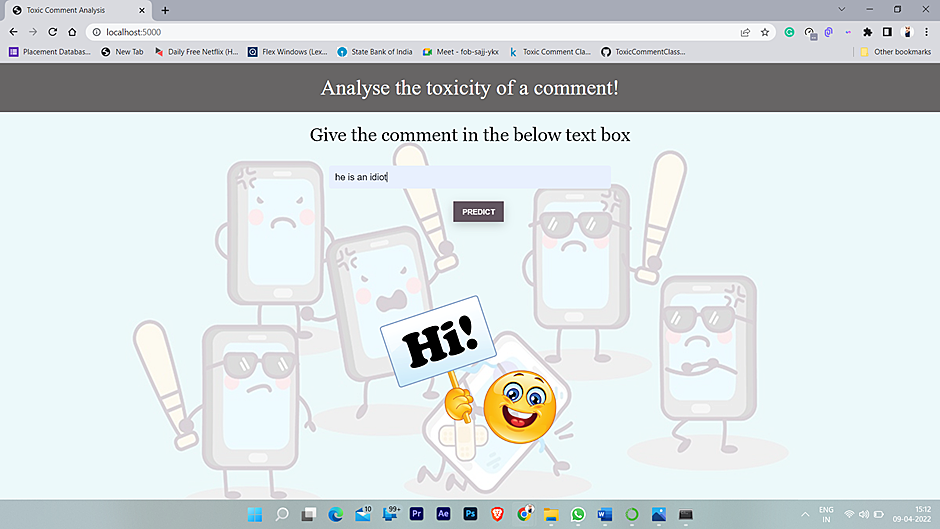


**b. UI Output Screenshot:**

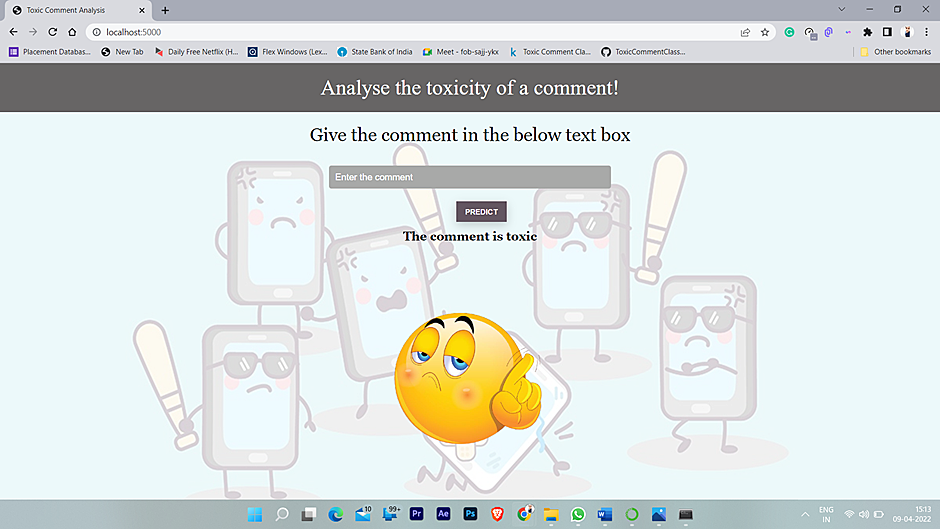
**FIG:1 HOME PAGE**



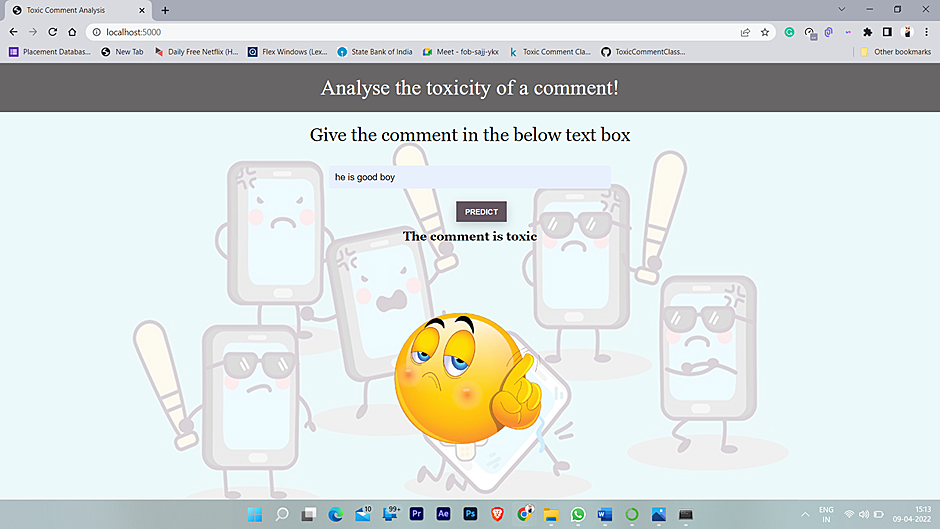
**FIG:2 ENTER A TOXIC COMMENT**



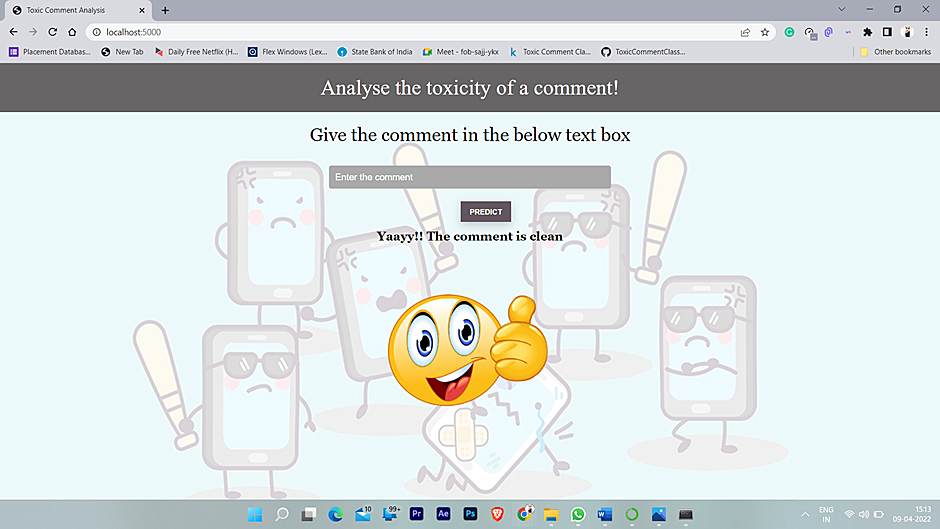
**FIG:3 SHOWING OUTPUT THE COMMENT IS TOXIC**



**FIG:4 ENTERING NORMAL COMMENT**



**FIG:5 SHOWING OUTPUT THE COMMENT IS CLEAN**



**THANK**

**YOU**

Submitted By

**KADALI JAGADEESH**

**39110424.**