

Prevention of Cyber Troll & Sarcasm System on Social Networking using Machine Learning with Bilingual Analytics

Project Synopsis
Submitted in Partial fulfillment of the requirements
For the degree of

BACHELOR OF TECHNOLOGY
BY

Tejas Karia

Priya Mane

Jeet Mehta

Pratik Merchant

Under the guidance of
Prof. Chirag Desai

DEPARTMENT OF INFORMATION TECHNOLOGY
K. J. Somaiya College of Engineering, Mumbai-77
(Autonomous College Affiliated to University of Mumbai)

2020-2021

Project synopsis

entitled 'Prevention of Cyber Troll & Sarcasm System on Social Networking using Machine Learning with Bilingual Analytics'

Submitted by:

Tejas Karia

Priya Mane

Jeet Mehta

Pratik Merchant

in Partial fulfillment of the degree of B. Tech. in Information Technology is approved.

Guide

Examiners

Head of Department

Principal

Date:

Abstract

With the recent growth in use of social media, the trend of having an opinion for every content published on the internet has also increased. The internet allows users to remain anonymous in the sense that there is no compulsion for authenticating oneself, because of which many users exhibit indecent behaviour by posting trolls, hate comments and spread negativity. The user who is the target of such actions may get seriously affected, he or she may slip into depression or lack of self-worth. To avoid such misconduct and essentially filter out such negativity, this paper proposes a web-application based deep learning solution to perform troll and sarcasm detection on comments received on a user's post and then using these comments, trace the user spreading hate and block or report him. The model consists of a Gated Recurrent Unit and a 4-layered neural network for troll detection and sarcasm detection respectively. As most of the comments posted in India are usually in Hinglish, which is a way of writing Hindi words using English letters, such comments will also be processed to determine the classification.

Contents

1	Introduction	6
1.1	Problem Definition	6
1.2	Motivation	6
1.3	Scope	6
1.4	Salient Contribution	7
1.5	Organization of the Synopsis	7
2	Literature Survey	8
3	Software Project Management Plan	10
3.1	Introduction	10
3.1.1	Project Overview	10
3.1.2	Project Deliverables	10
3.2	Project Organization	10
3.2.1	Software Process Model	10
3.2.2	Roles and Responsibilities	11
3.2.3	Tools and Techniques	12
3.3	Project Management Plan	12
3.3.1	Tasks	12
3.3.1.1	Requirement Analysis	12
3.3.1.1.1	Description	12
3.3.1.1.2	Deliverables and Milestones	12
3.3.1.1.3	Resources Needed	12
3.3.1.1.4	Dependencies and Constraints	13
3.3.1.1.5	Risks and Contingencies	13
3.3.1.2	Software Requirement Specification	13
3.3.1.2.1	Description	13
3.3.1.2.2	Deliverables and Milestones	13
3.3.1.2.3	Resources Needed	13
3.3.1.2.4	Dependencies and Constraints	13
3.3.1.2.5	Risks and Contingencies	13
3.3.1.3	APIs	13
3.3.1.3.1	Description	14
3.3.1.3.2	Deliverables and Milestones	14
3.3.1.3.3	Resources Needed	14
3.3.1.3.4	Dependencies and Constraints	14
3.3.1.3.5	Risks and Contingencies	14
3.3.1.4	Classification Models	14
3.3.1.4.1	Description	14
3.3.1.4.2	Deliverables and Milestones	14
3.3.1.4.3	Resources Needed	14
3.3.1.4.4	Dependencies and Constraints	14

3.3.1.4.5	Risks and Contingencies	15
3.3.1.5	User interface	15
3.3.1.5.1	Description	15
3.3.1.5.2	Deliverables and Milestones	15
3.3.1.5.3	Resources Needed	15
3.3.1.5.4	Dependencies and Constraints	15
3.3.1.5.5	Risks and Contingencies	15
3.3.1.6	Integration	15
3.3.1.6.1	Description	15
3.3.1.6.2	Deliverables and Milestones	16
3.3.1.6.3	Resources Needed	16
3.3.1.6.4	Dependencies and Constraints	16
3.3.1.6.5	Risks and Contingencies	16
3.3.1.7	Testing	16
3.3.1.7.1	Description	16
3.3.1.7.2	Deliverables and Milestones	16
3.3.1.7.3	Resources Needed	16
3.3.1.7.4	Dependencies and Constraints	16
3.3.1.7.5	Risks and Contingencies	16
3.3.2	Assignments	16
3.3.3	Timetable	17
4	Software Requirements Specification	19
4.1	Introduction	19
4.1.1	Product Overview	19
4.2	Specific Requirements	19
4.2.1	External Interface Requirements	19
4.2.1.1	User Interfaces	19
4.2.1.2	Hardware Interfaces	22
4.2.1.3	Software Interfaces	23
4.2.1.4	Communications Protocols	23
4.2.2	Software Product Features	23
4.2.3	Software System Attributes	23
4.2.4	Database Requirements	24
5	Software Design Description	25
5.1	Introduction	25
5.1.1	Design Overview	25
5.1.2	Requirements Traceability Matrix	25
5.2	System Architectural Design	26
5.2.1	Chosen System Architecture	26
5.2.2	Discussion of Alternative Designs	26
5.2.3	System Interface Description	26
5.2.3.1	User Interfaces	26
5.2.3.2	Hardware Interfaces	27
5.2.3.3	Software Interfaces	27

5.2.3.4	Communications Protocols	27
5.3	Detailed Description Of Components	27
5.3.1	Component 1: User	27
5.3.2	Component 2: API	27
5.3.3	Component 3: Sentiment Analysis	28
5.3.4	Component 4: Automating Replying/Blocking Process . .	28
5.3.5	Component 5: Database	28
5.4	User Interface Design	28
5.4.1	Description of the User Interface	28
5.4.1.1	Screen Images	28
5.4.1.2	Objects and Actions	
	31	
5.5	System Architecture	32
5.6	Data Flow Specifications	33
5.6.1	Level 0 DFD with description:	33
5.6.2	Level 1 DFD with description:	34
6	Software Test Document	35
6.1	Introduction	35
6.1.1	System Overview	35
6.1.2	Test Approach	35
6.1.2.1	Testing Method	35
6.1.2.2	Testing Strategies	35
6.2	Test Plan	35
6.2.1	Features To Be Tested	35
6.2.2	Features Not To Be Tested	36
6.2.3	Testing Tools and Environment	36
6.3	Test Cases	36
6.3.1	User Login: TC-1	36
6.3.1.1	Purpose	36
6.3.1.2	Inputs	36
6.3.1.3	Expected output and Pass/Fail Criteria	36
6.3.1.4	Test Procedure	36
6.3.2	Authorization of the web application: TC-2	36
6.3.2.1	Purpose	36
6.3.2.2	Inputs	36
6.3.2.3	Expected output and Pass/Fail Criteria	36
6.3.2.4	Test Procedure	37
6.3.3	User Dashboard: TC-3	37
6.3.3.1	Purpose	37
6.3.3.2	Inputs	37
6.3.3.3	Expected output and Pass/Fail Criteria	37
6.3.3.4	Test Procedure	37
6.3.4	Blocking/Reporting of malicious accounts: TC-4	37
6.3.4.1	Purpose	37

6.3.4.2	Inputs	37
6.3.4.3	Expected output and Pass/Fail Criteria	37
6.3.4.4	Test Procedure	37
7	Conclusion	38
	References	39
	Acknowledgment	40

1 Introduction

1.1 Problem Definition

This project will help to deal with online social media hate speech and automate the process of blocking such malicious accounts. The current process for the social media platforms are manual and there are no automated processes. Since the process is manual it becomes very difficult to keep track of such users who are habitual offenders. There are several categories of cyberhate and each of these are interpreted differently. The project has broken this down to mainly 2 categories: offensive and sarcastic depending on the sentiment & bilingual sentiment analysis on Hinglish comments. The main target for developing this tool is to empower influencers who do not have the time to tackle hate speech and thus they have to keep a social media manager who has to manually delete such malicious comments. Primarily, all the comments will be retrieved from the user created posts and then those will be classified using sentiment analysis. An automated response will be generated to the comments.

1.2 Motivation

With the rise of social media platform, the amount of hate spreading on such platforms has also been on the rise. Social media platforms have been trying to overcome such things by changing their policies time and again but have still failed to do so at large. People comment derogatory things on a users' post just to incite and evoke a reaction from the person. This in turn also affects the mental health of the person being subjected to such hate. Hence, we have taken an initiative through this project to eliminate such hateful comments from a user's post and thereby maintain the decorum of social media platforms.

1.3 Scope

The retrieval of comments on user created posts from the social media account will be done using APIs and displayed in a more comprehensive manner. Classification of the retrieved comments using sentiment analysis as offensive and sarcastic will be performed. For processing of comments posted in Hinglish i.e. typing Hindi using English alphabets, bilingual sentimental analysis will be performed. The process of deleting offensive comments and/or reporting the associated user would be automated. Automated responses to the comments received on the user's post would be provided to increase the interaction with the community. The application will be easy to use as the target audience for the application is the social media users. Hence, a similar seamless experience will be provided to the users of this website. The application will provide easy insights into the user's social media accounts and help them manage their social media accounts with ease. User credentials will be stored safely in the database by using a hash function. Hence, the original password the user enters is never revealed to anyone. The application will be available at all times i.e. all round

the year, only restricted by the down time of the server on which the system runs. Any changes in algorithms of sarcasm detection and offense detection as improvements should be easily pushed to the server.

1.4 Salient Contribution

The salient contribution of this project to world of social media would be the automation of some tedious processes such as replying to positive comments, deleting hate comments, blocking repeated offenders, etc. This in turn would help the users who could be small scale as well as large scale influencers to eliminate the task of explicitly appointing a social media manager to manage one's all social media handles. The easy to use user interface with a dashboard for every social media account linked with our web application would help in analysing the response received to the posts from the community in a much better way.

1.5 Organization of the Synopsis

The synopsis starts with the abstract followed by a brief introduction followed by the literature survey. Then start the various documents such as Software Project Management Plan, Software Requirements Specification, Software Design Description and Software Test Document. The synopsis finishes off with the conclusion followed by the references.

2 Literature Survey

A) Troll Classification: The work [3] in the research paper titled 'Incivility Detection in Online Comments' co-authored by Farig Sadeque, Stephen Rains, Yotam Shmargad, Kate Kenski, Kevin Coe and Steven Bethard was referred to. Highlights from the paper are as follows: The dataset used was russian troll dataset and the model was trained on newspaper comment data to detect any vulgar or offensive comments. A Recurrent Neural Network (RNN) architecture was used which had input, embedding, Gated Recurrent Unit (GRU), Average pooling, Max pooling, Concatenation layers, and finally sigmoid activation function was used to make the binary classification of whether the comment was civil or not civil. The paper provides evidence that the model can also detect incivilities in Twitter or any other social media platforms.

B) Sarcasm Detection: A lot of study has been conducted for detecting sarcasm in texts. The work [1] proposes a statistical approach for detecting sarcasm. The SemEval Task11 datasets were used and certain statistical methods were used to derive features from tweets. Based text cleaning was proposed and a set of 12 features were to be extracted from the tweets for classifying tweets as sarcastic or not. These 12 features were divided into 2 categories - sentiment based features (positive word count, negative word count, frequency of words having high emotional content that lies in positive category, frequency of words having high emotional content that lies in the negative category, count of verbs, count of nouns) and punctuation based features (count of repeated words, frequency count of dots, count of exclamations, frequency count of question marks, frequency count of quotes, frequency count of capital letters). Further, features selection is done using chi-square method since it can be used efficiently to determine useful features from the above extracted features. It gives an array which can be further sort to get the p-values. From which the top seven most informative features based on their p-values are selected for further usage in the first approach. In the first approach, selected features were used and use various machine learning algorithms like KNN, Decision Tree, Random Forest, SVM algorithms were implemented on these selected features and compare the results that are obtained with the results obtained when all the features are used. The features which were extracted after performing the feature selection are:

pw: count of positive words.

PW: frequency of words having high emotional content that lies in the positive category.

NW: frequency of words having high emotional content that lies in the negative category.

cV: frequency count of verbs.

cN: frequency count of nouns.

rep: number of repetitions of the letter in the tweet.

cap: number of capital letters in a tweet.

In the second approach, the total features with top 200 TF-IDF features were used to get a more complex decision boundary and to improve the accuracy, ensemble model was used. The learner method called voting classifier was used to get the best of each and the result is predicted based on majority voting. In the first approach, the SVM algorithm gives an accuracy of 74.59% which is highest as compared to other algorithms. Similarly, in the second approach Voting Classifier achieves an accuracy of 83.53%. The work [4] proposed sAtt-BLSTM convNet - a hybrid of soft attention-based bidirectional long short-term memory (sAtt-BLSTM) and convolution neural network (convNet) for sarcasm detection. It proposed a 8 layer model - Input Layer, Embedding Layer, BLSTM layer, Attention Layer, Convolution Layer, Activation Layer, Down-sampling Layer, Representation Layer. The accuracy metrics of the model are 97.87% for the Twitter dataset and 93.71% for the random-tweet dataset.

C) Hinglish: The work [2] in the research paper titled ‘Mixed Bilingual Social Media Analytics’ co-authored by Saurabh Malgaonkar, Aejazul Khan and Abhishek Vichare was referred to. Following were some of the highlights of the paper: The dataset used was that of Twitter Corpus. Text Mining Techniques spanning across NLP was the approach followed. Extraction of data followed by cleaning of data followed by keyword matching against the dictionaries (English, Hindi & Hinglish) was done. The algorithms used were Breen’s algorithm & Cholesky decomposition for determining unknown sentiment. For better results, the Hindi dictionary should be well equipped with spelling ambiguities.

3 Software Project Management Plan

3.1 Introduction

3.1.1 Project Overview

This project will help to deal with online social media hate speech and automate the process of blocking such malicious accounts. The current process for the social media platforms are manual and there are no automated processes. Since the process is manual it becomes very difficult to keep track of such users who are habitual offenders. There are several categories of cyberhate and each of these are interpreted differently. The project has broken this down to mainly 2 categories: offensive and sarcastic depending on the sentiment and bilingual sentiment analysis on Hinglish comments. Our main target for developing this tool is to empower influencers who do not have the time to tackle the hate speech and thus they have to keep a social media manager who has to manually delete such malicious comments. Primarily, all the comments will be retrieved from the user created posts and then those will be classified using sentiment analysis. An automated response will be generated to the comments.

3.1.2 Project Deliverables

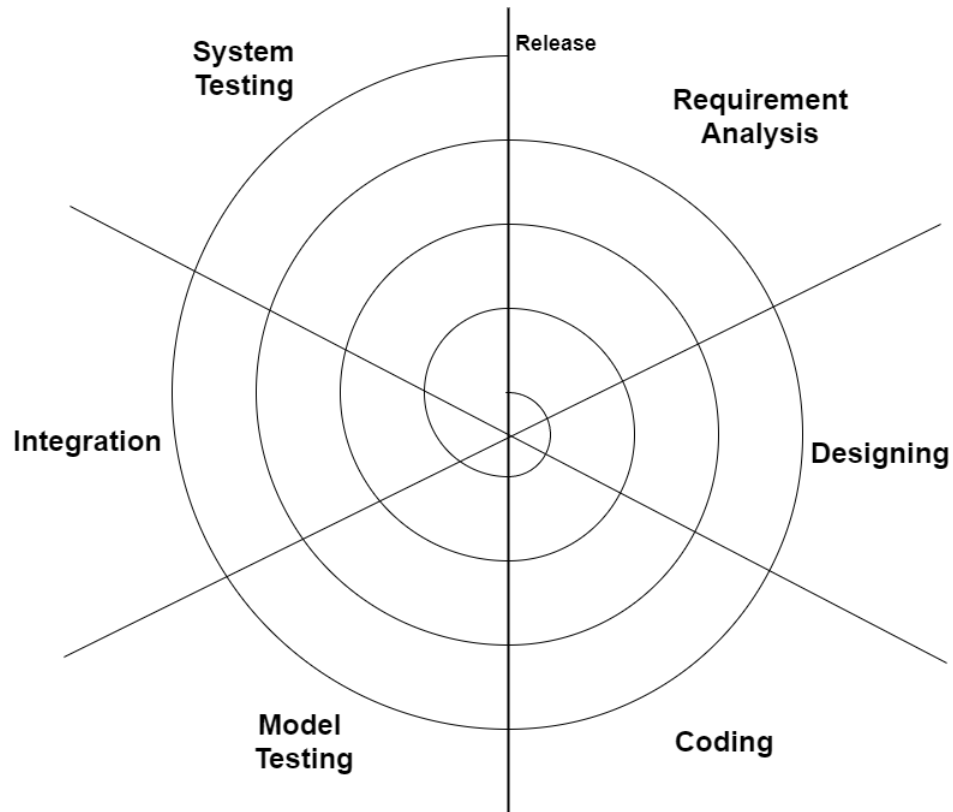
- Requirement Analysis - Mid September
- Software Requirement Specification Document - Mid October
- Algorithm for Troll Detection - Mid October
- Software Design Document - Mid October
- Software Test Document - October End
- Demonstrative User Interface - October End
- Algorithm for Sarcasm Detection - October End
- Project Synopsis - Mid November
- Algorithm for Bilingual analytics (Hinglish) - January End
- Complete User Interface and Database Connectivity - Mid February
- Integrated Web Application - February End
- Addition of Security Features - February End
- Tested Web Application - Mid March
- Final Project Report - Mid April

3.2 Project Organization

3.2.1 Software Process Model

The iterative spiral model approach would be used for this project. This model allows us to iterate through each process framework communication, planning, modelling, construction and delivery.

Figure 1: Process Model



3.2.2 Roles and Responsibilities

1. Tejas Karia (1714088) - Project Manager

The project manager will look after the overall functioning of the project and assign tasks to each of the members. He/She will act as the face of the project and will lead all the activities.

2. Pratik Merchant (1714093) - Designer

The designer will design the overall system and will lead all the activities of designing the system in the best and most efficient way possible.

3. Priya Mane (1714091) - Developer

The developer will define how the design is converted to reality by defining the implementation details of the various modules and successfully developing the same.

4. Jeet Mehta (1714092) - Tester

The tester will lead all the activities of testing each and every module that has

been implemented to determine the errors and will hence inform the developer to overcome the same. If a design flaw has been spotted, the designer would be informed first and then the developer.

3.2.3 Tools and Techniques

- The application will be developed for desktop usage. It would be compatible across any standard browsers such as Google Chrome, Mozilla Firefox, Internet Explorer, etc.
- SQLite database will be used to store user login credentials and details for the web application.
- The algorithms will be implemented using Python.
- For developing demo UI / Wireframes , a wireframing tool would be used.
- OAuth authentication service will be used to get special rights for performing operations on behalf of the user.
- The respective APIs will be used for accessing data from social media platforms.
- The web based application will be developed using the Django framework for backend and HTML,CSS for frontend development.
- For development synchronisation purposes, version control system would be used.
- Git will be as version control system as it can be easily used in android studio and simple to understand.

3.3 Project Management Plan

3.3.1 Tasks

3.3.1.1 Requirement Analysis

3.3.1.1.1 Description Requirement Analysis would be done to know the exact expectations of the client from the product. The functionalities and working of the product would also be clear by doing sufficient and effective requirement analysis.

3.3.1.1.2 Deliverables and Milestones Effectively communicate with all the actors involved in the working of the web application. By the end of this process, the design and development team will be sure of the functioning of the web application.

3.3.1.1.3 Resources Needed For effective requirements analysis, multiple meetings would have to be conducted with the stakeholders of the web application. Brainstorming sessions or joint discussions must be organised for effective communication and information gathering.

3.3.1.1.4 Dependencies and Constraints Task cannot be completed without conducting meetings with the stakeholders and knowing the expectations from the web application.

3.3.1.1.5 Risks and Contingencies Risks: The risk involved would be only failure to communicate with the organisations involved and users. Another issue could be miscommunication between the stakeholders and developers/designers.

Contingency: Can be tackled by having multiple sessions and creating well defined SRS and getting it approved by the client.

3.3.1.2 Software Requirement Specification

3.3.1.2.1 Description The users and the client get a brief idea about the software while in the initial stages. The purposes and intentions as well as the expected results are properly defined. It hence lays the outline for software design. The desired goals are defined thereby easing off the efforts of the developers in terms of time and cost. It forms a basis for the agreement between the client and the developer. It becomes easier while transferring and using the solution elsewhere or with new customers as the basis of functioning of the software is mentioned. It acts as a material for reference at a later stage. It acts as a basis for reviews.

3.3.1.2.2 Deliverables and Milestones The document focuses on briefing all the members of the team as well as the client about the specifications and functionalities of the software project.

3.3.1.2.3 Resources Needed Meetings with stake-holders and brainstorming sessions or joint discussions must be organised for requirement gathering.

3.3.1.2.4 Dependencies and Constraints Task cannot be completed without conducting meetings with the stakeholders and knowing the expectations from the web application.

3.3.1.2.5 Risks and Contingencies Risks: Matter of risk mainly revolves around communication and necessary documentation. If the SRS isn't well defined and well addressing each and every aspect of the project, then major miscommunication and false information transfer could take place. Clients may face issues on being on the same page as the developing team. The expectations and deliverables would have explosive differences between them.

Contingency: Can be tackled by having multiple sessions and creating well defined SRS and getting it approved by the client.

3.3.1.3 APIs

3.3.1.3.1 Description Using APIs of social media platforms, comments will be retrieved and automated responses would be generated. The flagged comments will be deleted or reported using the API calls.

3.3.1.3.2 Deliverables and Milestones The API will facilitate the retrieval of comments on an user's post. Based on the sentiment of the comment, then, an automated response will be provided. Deletion or reporting of flagged comments.

3.3.1.3.3 Resources Needed User login credentials and user authentication services.

3.3.1.3.4 Dependencies and Constraints The task cannot be completed without getting authentication from the respective social media platform regarding the API usage.

3.3.1.3.5 Risks and Contingencies Risk - The risk involved is with the user authentication for each social media platform, the authentication must allow access through some authentication service for the application to carry out tasks like posting replies and blocking users or deleting user comments based on the results of sarcasm detection and troll detection. Contingency - Proper authentication rights can be acquired by certain services offered by the particular social media platform. Exploring these services can help overcome the risk mentioned above.

3.3.1.4 Classification Models

3.3.1.4.1 Description A classification model for classifying offensive or sarcastic comments algorithms will be coded. It will include pre-processing, training and testing the data.

3.3.1.4.2 Deliverables and Milestones The model should be able to classify offensive or sarcastic comments with utmost accuracy and not overfit the data.

3.3.1.4.3 Resources Needed Huge amount of labelled dataset, Google Collaboratory for training and testing the models.

3.3.1.4.4 Dependencies and Constraints The accuracy of the model depends highly on the dataset and suitable processing power for executing the algorithms. The labelled dataset should be accurate and also from the same distribution for better results.

3.3.1.4.5 Risks and Contingencies Risks: Dataset not being labelled correctly and not from same distribution. Model overfitting or underfitting the data or have high bias and/or variance.
Contingencies: Select dataset with accurate labelling and from the same distribution. High bias can be solved with more data and high variance can be solved by dropout method.

3.3.1.5 User interface

3.3.1.5.1 Description This unit will be the most crucial part of the project since the target audience includes social media users and hence the ease of using this application must be comparable if not better, to the existing social media applications. The home screen will describe the application features and login/signup buttons. Once the user signs up for the application and logs into his/her account, the user is taken to a dashboard. The dashboard will have different tabs or sections for various social media platforms. For each application, there will be an authentication for requesting access to the user's account with rights to perform required operations. Once authentication is done, users will be displayed with the details of the comments and further analysis after applying algorithms for sarcasm detection, troll detection and bilingual analysis.

3.3.1.5.2 Deliverables and Milestones Easy to use and intuitive user interface.

3.3.1.5.3 Resources Needed Wireframing tool for creating demonstrative user interface web pages. HTML and CSS will be used primarily for developing the User interface.

3.3.1.5.4 Dependencies and Constraints Without adequate feedback from the client regarding ease of navigation and usage on the website, it would be difficult to design it effectively.

3.3.1.5.5 Risks and Contingencies Risk - The primary risk could be difficulty for a new user to navigate through the website or difficulty in comprehending the results of the analysis displayed on the dashboard.
Contingency - The risk can be mitigated by seeking continuous feedback from users.

3.3.1.6 Integration

3.3.1.6.1 Description Integration of various social media platforms, their services and algorithms for sarcasm and troll detection on the website to make the user dashboard.

3.3.1.6.2 Deliverables and Milestones Fully developed web-based application with insightful user interface.

3.3.1.6.3 Resources Needed The framework used for web development will be Django for backend and HTML, CSS for the frontend. SQLite database for storing user credentials.

3.3.1.6.4 Dependencies and Constraints All social media platform APIs must function smoothly under a single application.

3.3.1.6.5 Risks and Contingencies Risk: If any API fails, the user won't be able to have a wholesome experience. Integration may result in unforeseen bugs.
Contingency: The system integration can be tested by conducting rigorous integration tests to mitigate bugs.

3.3.1.7 Testing

3.3.1.7.1 Description Once the design and final software is developed, the application goes for testing. Testing is based on different criterias related to efficiency, bugs, performance, response time, correct functionality, etc.

3.3.1.7.2 Deliverables and Milestones The web application must be able to perform all the tasks successfully and give all expected results.

3.3.1.7.3 Resources Needed Dummy data and some users would be required to test the web application.

3.3.1.7.4 Dependencies and Constraints The step will be incomplete without the web application being handled by real users and getting their feedback.

3.3.1.7.5 Risks and Contingencies Risk: The web application might not be satisfactory or the functions/code might cause some anomalies and give wrong results.
Contingency: If the web application gives faulty results, it can be corrected by debugging the app and again presenting it to the customer.

3.3.2 Assignments

Task 1: Requirement Analysis - Tejas, Pratik

Task 2: Software Requirement Specification - Pratik, Jeet

Task 3: APIs - Tejas, Priya, Jeet, Pratik
















Task 4: Algorithms - Tejas, Priya, Jeet, Pratik

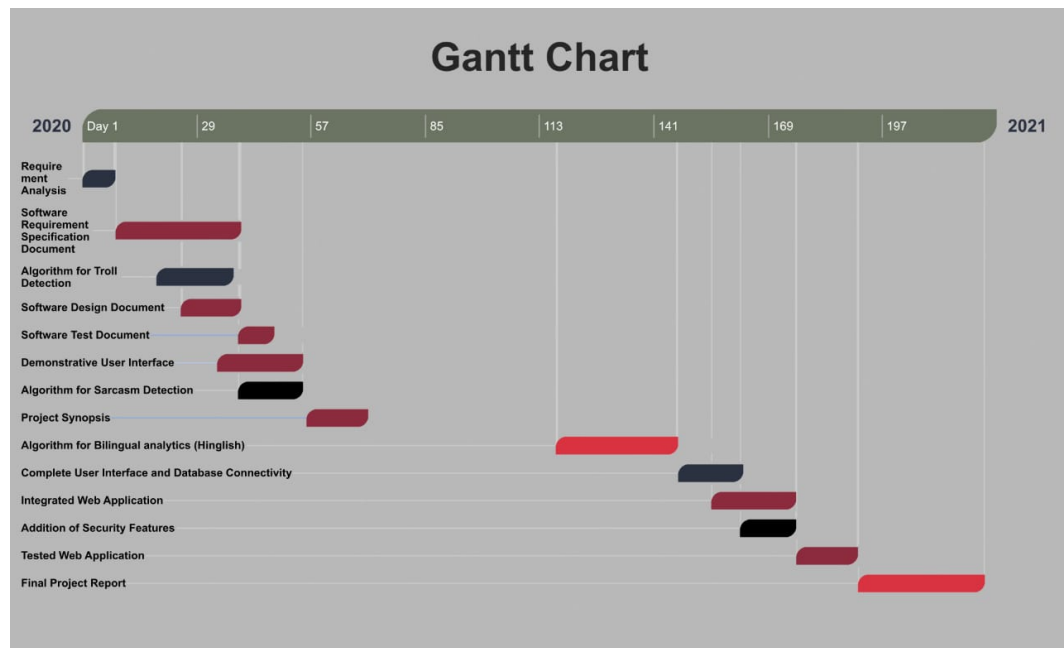
Task 5: User interface - Tejas, Priya

Task 6: Integration - Tejas, Priya, Jeet, Pratik

Task 7: Testing - Jeet, Priya

3.3.3 Timetable

 Title	T/M	Start	End
<input type="radio"/> Requirement Analysis	 T ▾	07/09/2020	14/09/2020
<input type="radio"/> Software Requirement Specification Document	 T ▾	15/09/2020	15/10/2020
<input type="radio"/> Algorithm for Troll Detection	 T ▾	25/09/2020	15/10/2020
<input type="radio"/> Software Design Document	 T ▾	01/10/2020	15/10/2020
<input type="radio"/> Software Test Document	 T ▾	15/10/2020	23/10/2020
<input type="radio"/> Demonstrative User Interface	 T ▾	10/10/2020	30/10/2020
<input type="radio"/> Algorithm for Sarcasm Detection	 T ▾	15/10/2020	30/10/2020
<input type="radio"/> Project Synopsis	 T ▾	01/11/2020	15/11/2020
<input type="radio"/> Algorithm for Bilingual analytics (Hinglish)	 T ▾	01/01/2021	30/01/2021
<input type="radio"/> Complete User Interface and Database Connectivity	 T ▾	31/01/2021	15/02/2021
<input type="radio"/> Integrated Web Application	 T ▾	08/02/2021	28/02/2021
<input type="radio"/> Addition of Security Features	 T ▾	15/02/2021	28/02/2021
<input type="radio"/> Tested Web Application	 T ▾	01/03/2021	15/03/2021
<input type="radio"/> Final Project Report	 T ▾	16/03/2021	15/04/2021



Gantt Chart for Timetable

4 Software Requirements Specification

4.1 Introduction

4.1.1 Product Overview

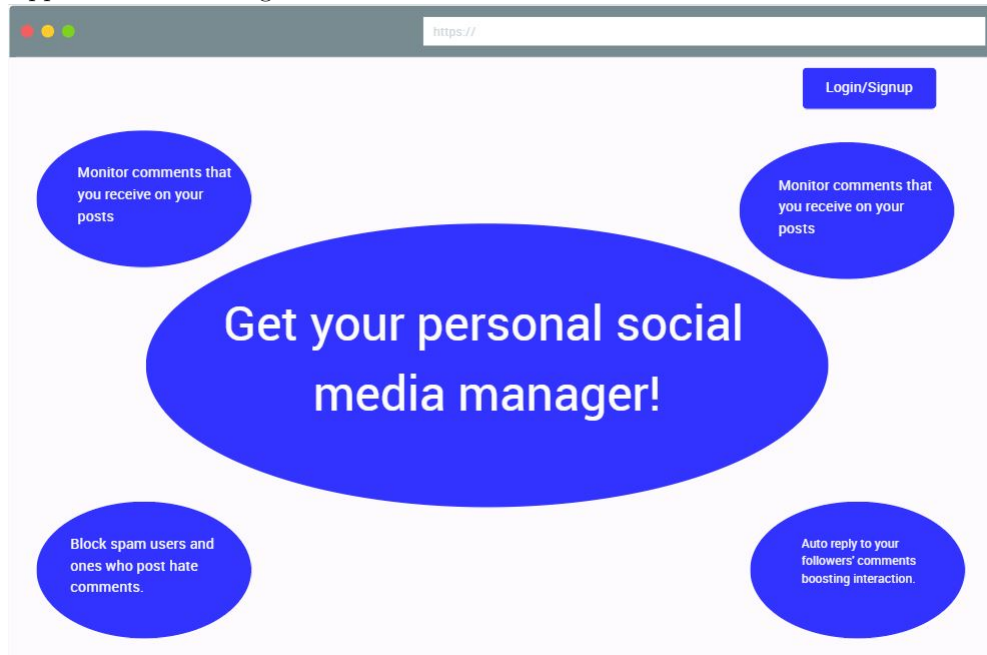
This project will help to deal with online social media hate speech and automate the process of blocking such malicious accounts. The current process for the social media platforms are manual and there are no automated processes. Since the process is manual it becomes very difficult to keep track of such users who are habitual offenders. There are several categories of cyberhate and each of these are interpreted differently. The project has broken this down to mainly 2 categories: offensive and sarcastic depending on the sentiment & bilingual sentiment analysis on Hinglish comments. Our main target for developing this tool is to empower influencers who do not have the time to tackle the hate speech and thus they have to keep a social media manager who has to manually delete such malicious comments. Primarily, all the comments will be retrieved from the user created posts and then those will be classified using sentiment analysis. An automated response will be generated to the comments.

4.2 Specific Requirements

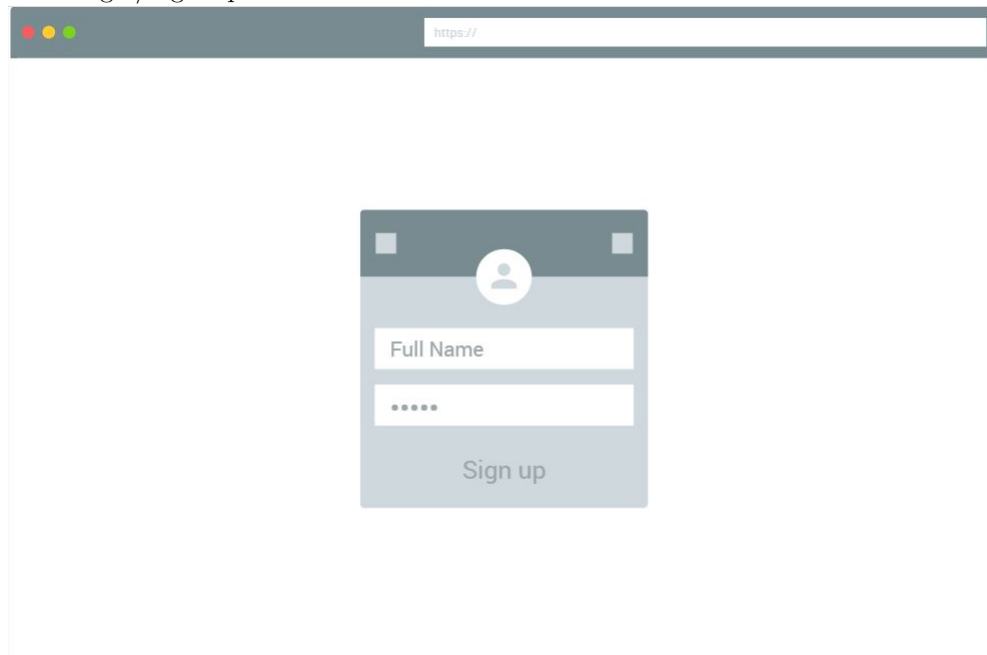
4.2.1 External Interface Requirements

4.2.1.1 User Interfaces

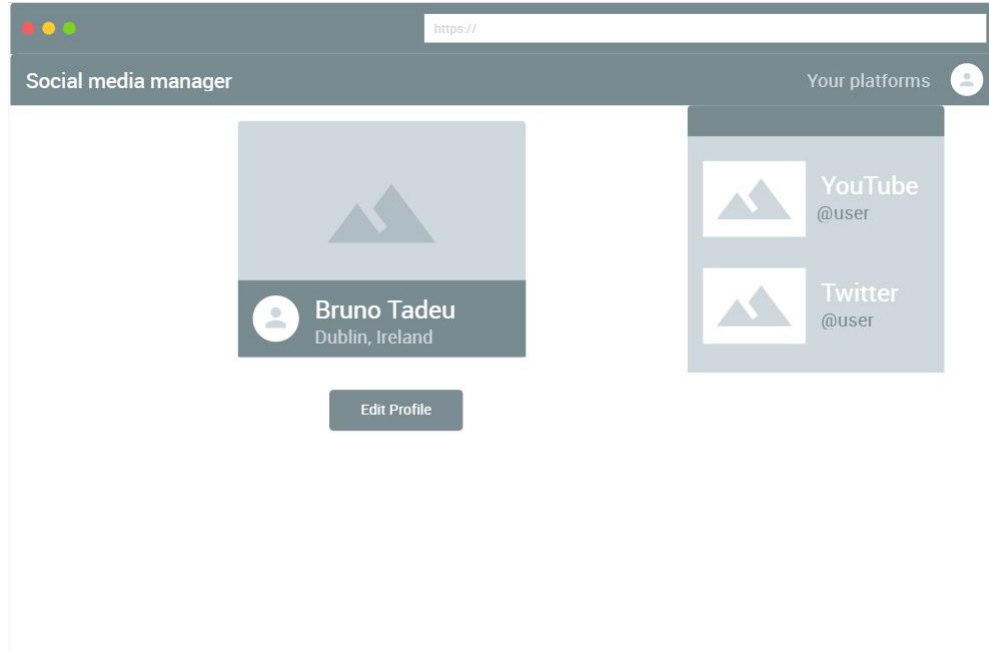
Application Home Page:



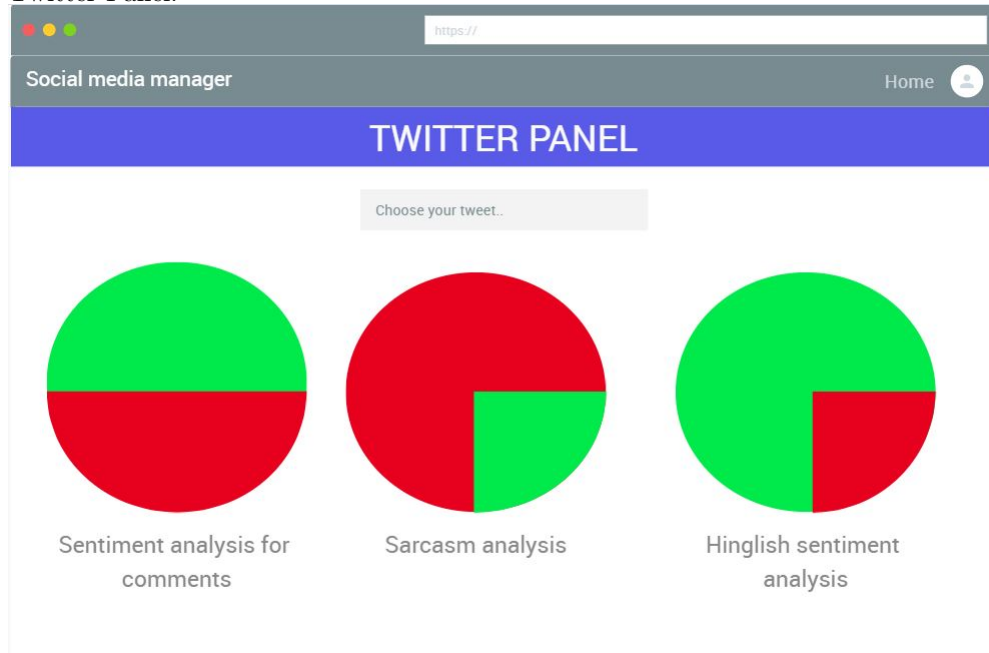
User Login/Sign Up:



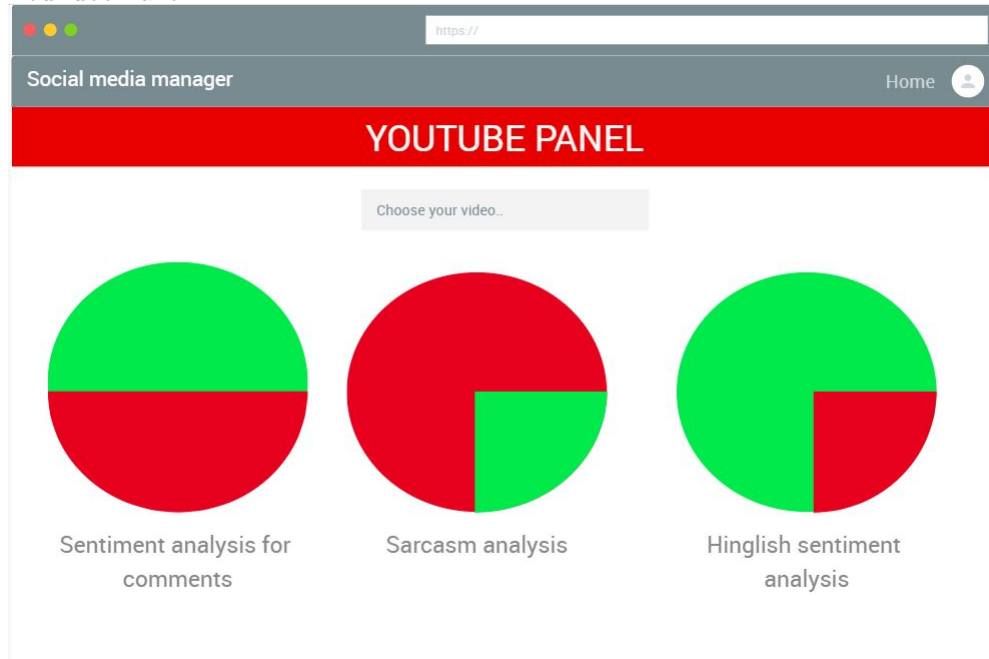
User Dashboard:



Twitter Panel:



YouTube Panel:



Detailed Review for YouTube:

The YouTube Panel Review interface features a red header with the title 'YOUTUBE PANEL REVIEW'. Below this is a grey navigation bar containing 'Social Media Manager' and a 'Home' button with a user icon. A table displays video reviews with the following columns: VIDEO, COMMENT, USER NAME, SENTIMENT, and SARCASM.

VIDEO	COMMENT	USER NAME	SENTIMENT	SARCASM
Holiday Vlog	Trash Vloggg	Nalini Roy	NEGATIVE	0
Holiday Vlog	great vlog, enjoyed it a lot!!	VK Kumar	POSITIVE	0

4.2.1.2 Hardware Interfaces There are no specific hardware requirements for using the website as it would be hosted over the internet.

4.2.1.3 Software Interfaces Operating System: The application would be compatible with all the operating systems which have the compatible browsers installed. The website would be accessible on various browsers such as Google Chrome, Mozilla Firefox and Microsoft Edge to name a few.
Database: For storing user data we will be making use of MySQL.

4.2.1.4 Communications Protocols For uploading data to the database and retrieving the data from the database over the internet, the relevant TCP protocols will be used. OAuth will be used for establishing communication with the APIs.

4.2.2 Software Product Features

Functional Requirements:

- Retrieval of comments on user created posts from the social media account using APIs and displaying it in a more comprehensive manner.
- Classification of the retrieved comments using sentiment analysis as offensive and sarcastic.
- For processing of comments posted in hinglish i.e. typing hindi using english alphabets, bilingual sentimental analysis will be performed.
- The process of deleting offensive comments and/or reporting the associated user would be automated.
- Automated responses to the comments received on the user's post would be provided to increase the interaction with the community.

4.2.3 Software System Attributes

Usability:

The application will be easy to use as the target audience for the application is the social media users. Hence, a similar seamless experience will be provided to the users of this website. The application will provide easy insights into the user's social media accounts and help them manage their social media accounts with ease.

Availability:

User credentials will be stored safely in the database by using a hash function. Hence, the original password the user enters is never revealed to anyone.

Security:

The application will be available at all times i.e. all round the year, only restricted by the down time of the server on which the system runs.

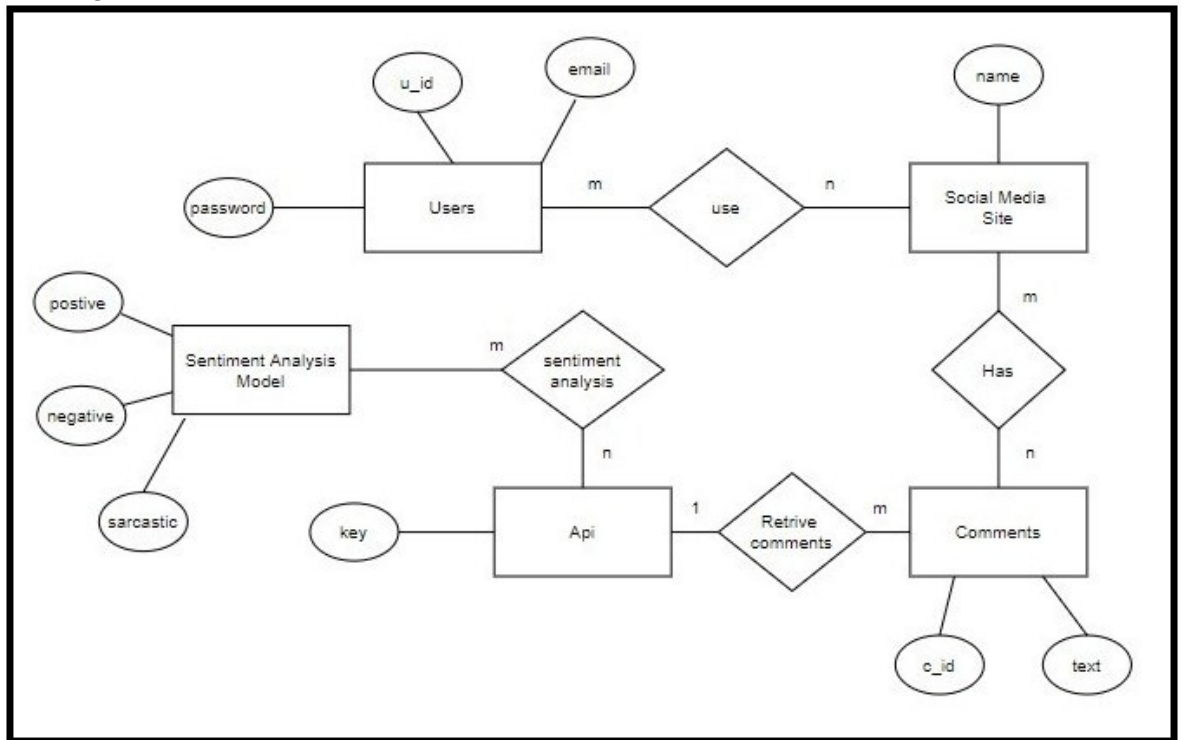
Maintainability:

Any changes in algorithms of sarcasm detection and offense detection as improvements should be easily pushed to the server.

4.2.4 Database Requirements

The user enters their login credentials, which will be used to generate the API key. Using this key, API requests can be made for retrieving comments on a user's post. These comments will be processed using the Sentiment Analysis Model to classify comments as positive, negative or sarcastic. Appropriate API requests can then be made to delete offensive comments, reply to comments or flag them. These are the necessary tables and their functions required for the proper functioning of the website.

ER Diagram:



5 Software Design Description

5.1 Introduction

5.1.1 Design Overview

This project will help to deal with online social media hate speech and automate the process of blocking such malicious accounts. The current process for the social media platforms are manual and there are no automated processes. Since the process is manual it becomes very difficult to keep track of such users who are habitual offenders. There are several categories of cyber hate and each of these are interpreted differently. The project has broken this down to mainly 2 categories: offensive and sarcastic depending on the sentiment & bilingual sentiment analysis on Hinglish comments. Our main target for developing this tool is to empower influencers who do not have the time to tackle the hate speech and thus they have to keep a social media manager who has to manually delete such malicious comments. Primarily, all the comments will be retrieved from the user created posts and then those will be classified using sentiment analysis. An automated response will be generated to the comments. The product will be designed using Bottom-up Object Oriented Design approach. All these separate modules will be generated first and then these subsystems will be clubbed to form our main system.

5.1.2 Requirements Traceability Matrix

	User	API/OAuth	Sentiment Analysis	Automation	Database
User Login	X				X
User Authentication for Social Media Platform.	X	X			
Retrieval & Classification of Comments	X	X	X		
Generate responses for comments received.			X	X	
Block spam users and report/delete negative comments		X	X	X	

5.2 System Architectural Design

5.2.1 Chosen System Architecture

Data-Centered Architecture:

The main component of the web application is the data pulled in from the various APIs. The secondary component consists of the modules processing the data. The user is a passive spectator and makes few decisions like assisting the auto-reply and blocking users. Hence, the architecture chosen is Data-Centered architecture. The major risk associated with the application would have been regarding user credentials security, but this risk can be eliminated by using OAuth provided by the social media platform itself. Another risk would be of misidentifying spam users or negative comments because of algorithmic flaws, to eliminate this risk, the application will present a preview for all the actions it will be performing to seek rectification from the user.

5.2.2 Discussion of Alternative Designs

The alternate application architecture can be Data-Centered Blackboard type. In Blackboard Architecture Style, the data stored is active and its clients are passive. Therefore, the logical flow is determined by the current data status in the data store. The Knowledge Sources (KS) is the data from APIs. Blackboard Data Structure is the user console/dashboard component. Control is the user itself. This architecture was not chosen because the application is not entirely passive as the user has to interact with the application for approving changes like automated replies and user blocking, etc.

5.2.3 System Interface Description

5.2.3.1 User Interfaces Virtual Social Media Manager is a web application that allows the user to analyse their social media comments and eliminate negative content by features like blocking users posting negative remarks and reporting/blocking hate comments.

- The web application consists of a first page i.e the application home, from where the user can navigate to login/sign up pages.
- On the sign up page, the user has to enter a few basic details and choose a safe password.
- On the Login page, the user has to enter his/her email ID and password to login.
- After Login, the user will be redirected to his home page/Dashboard containing his/her details. From here the user can navigate to various panels for each social media platform.
- On each panel, the user will be redirected to an OAuth page for that particular platform. The user has to authenticate himself and then he/she will be redirected to the panel page for that social media platform.
- On the panel page, all the user's posts are retrieved and the comments are

analysed and displayed in a tabular form. The user can also select to view content specific review.

5.2.3.2 Hardware Interfaces There are no specific hardware requirements for using the website as it would be hosted over the internet.

5.2.3.3 Software Interfaces Operating System: The application would be compatible with all the operating systems which have the compatible browsers installed. The website would be accessible on various browsers such as Google Chrome, Mozilla Firefox and Microsoft Edge to name a few.

Database: For storing user data we will be making use of MySQL.

5.2.3.4 Communications Protocols For uploading data to the database and retrieving the data from the database over the internet, the relevant TCP protocols will be used. OAuth will be used for establishing communication with the APIs.

5.3 Detailed Description Of Components

5.3.1 Component 1: User

Responsibilities	1. Create an account if not already created. 2. Login
Constraints	Password string should match with that in the database.
Composition	Frontend: HTML5, CSS3. Django Framework. Backend: MySQL
Interaction	Users log into the application. Provides the credential detail for OAuth. The user can review the changes like automated replies and block the users with malicious intent.
Resources	Developer

5.3.2 Component 2: API

Responsibilities	The responsibilities of API to provide broad access to public data that users have chosen to share. It also allows users to manage their own non-public information and provide information to developers.
Constraints	Requires an API key without which the access is not granted to the developer.
Composition	JSON
Interaction	This helps us in retrieval of comments and thus we can further use this data and implement various algorithms to calculate its sentiment value.
Resources	Developers should have appropriate API keys which are given upon after applying formally with all the verifications of the developer account done by the backend team.

5.3.3 Component 3: Sentiment Analysis

Responsibilities	Comments retrieved with the help of API will be then further classified into sarcasm or offensive according to the sentiment value it possesses.
Constraints	Requires a huge dataset. The model must perform with an acceptable accuracy.
Composition	Python and libraries such as NLP, <u>scikitlearn</u> , and NLTK.
Interaction	It interacts with the API and data obtained by the developer. Once the data is classified into sarcasm or offensive with the help of a value predefined for it then only the sentiment analysis is considered to be completed.
Resources	Algorithms and Google Colab for training the model.

5.3.4 Component 4: Automating Replying/Blocking Process

Responsibilities	Detect the hate comments which were classified as mentioned earlier and then block that account.
Constraints	User IDs identified to be blocked must be correct and sentiment value assessed must be accurate.
Composition	Will be coded such that the user will be blocked automatically and if the comment is appropriate then only it will be displayed on the post.
Interaction	The sentiment value calculated should help with identifying whether a particular account is to be blocked or not. An automated reply will be generated for appropriate comments.
Resources	Appropriate API endpoints.

5.3.5 Component 5: Database

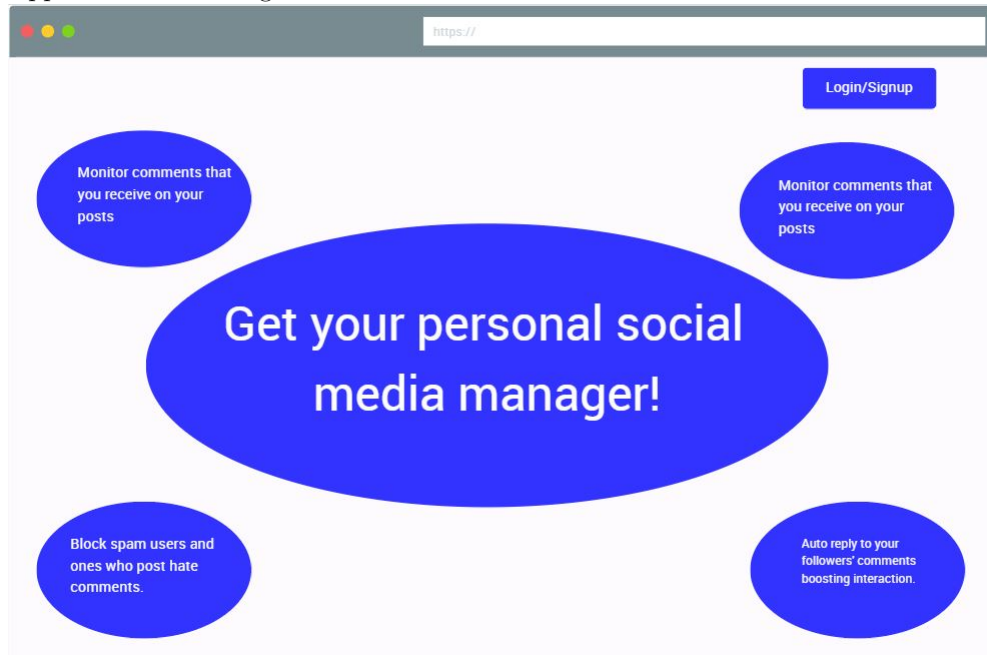
Responsibilities	Database stores information such as username & password.
Constraints	Erroneous data must be avoided .
Composition	MySQL.
Interaction	The application interacts with the database to ensure that credentials entered by the user is correct.
Resources	MySQL.

5.4 User Interface Design

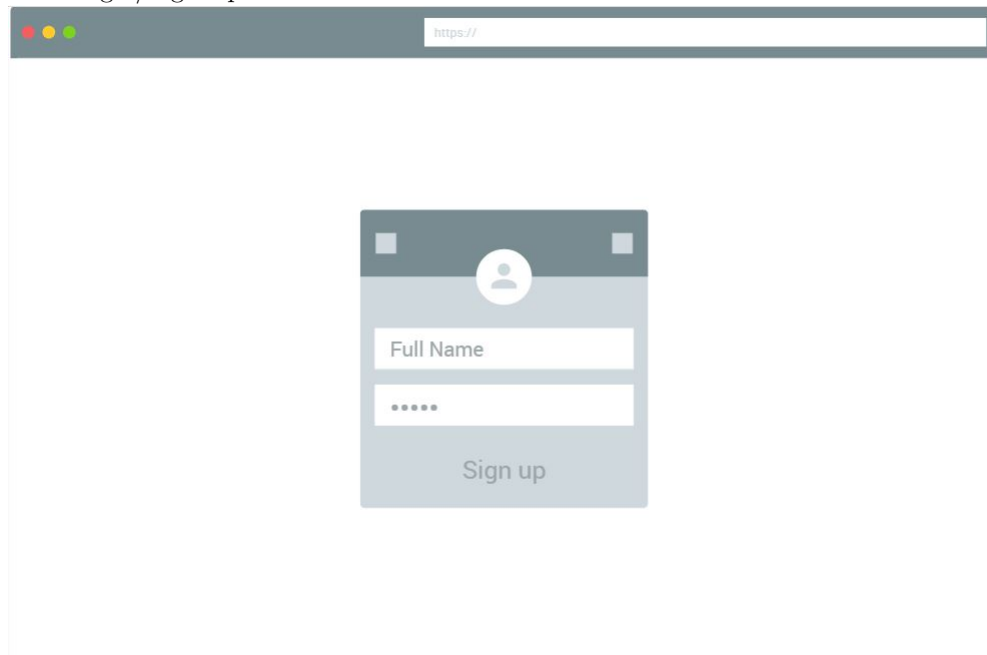
5.4.1 Description of the User Interface

5.4.1.1 Screen Images

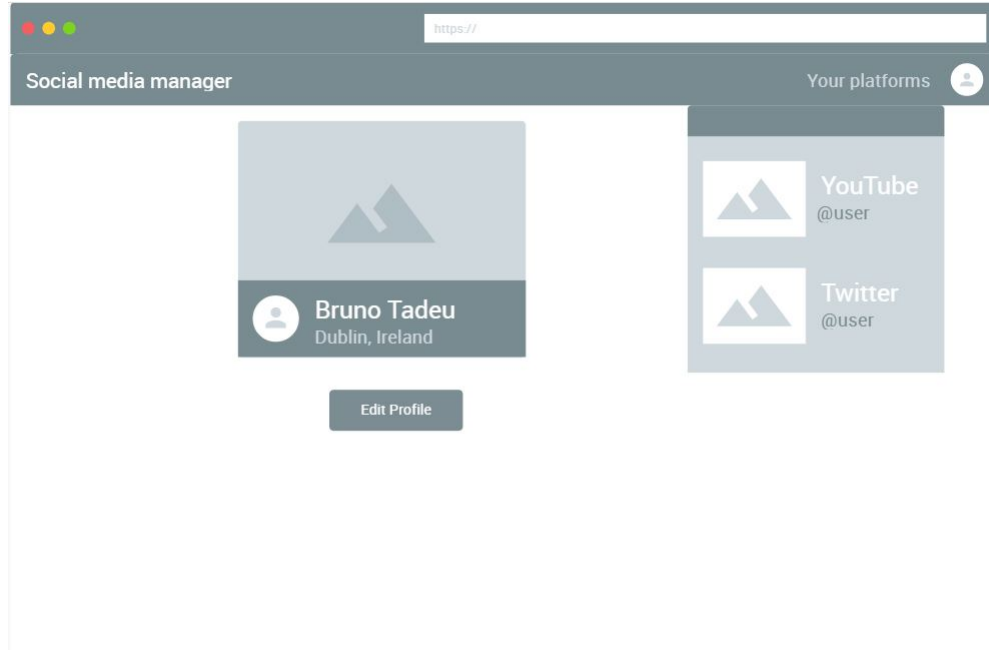
Application Home Page:



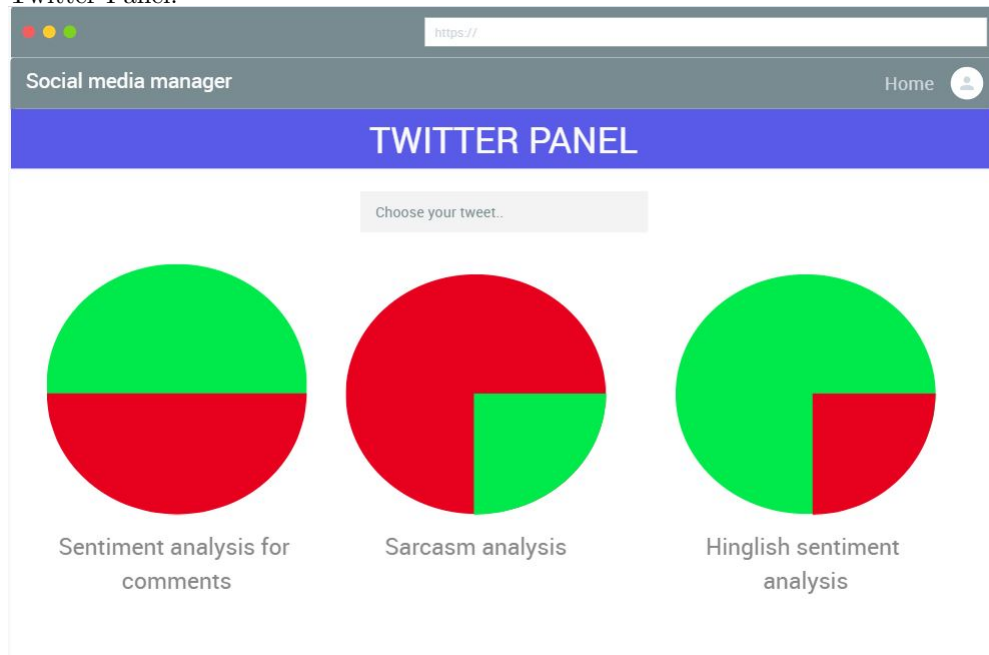
User Login/Sign Up:



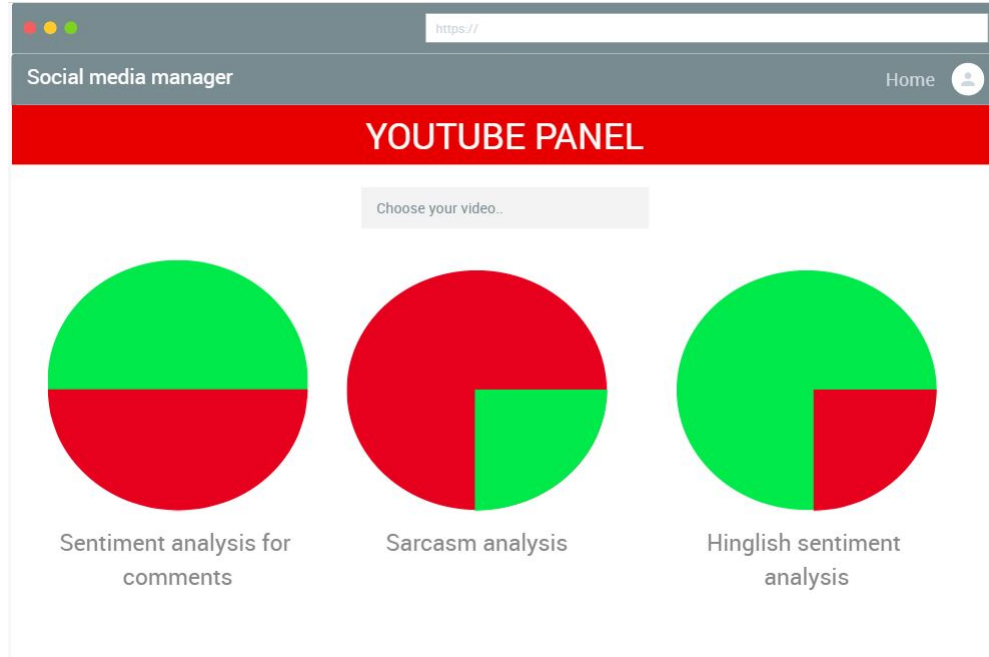
User Dashboard:



Twitter Panel:



YouTube Panel:



Detailed Review for YouTube:

The screenshot shows a web application titled "Social Media Manager" with a "Home" button. Below the header is a red banner labeled "YOUTUBE PANEL REVIEW". Below the banner is a table with the following data:

VIDEO	COMMENT	USER NAME	SENTIMENT	SARCASM
Holiday Vlog	Trash Vloggg	Nalini Roy	NEGATIVE	0
Holiday Vlog	great vlog, enjoyed it a lot!!	VK Kumar	POSITIVE	0

5.4.1.2 Objects and Actions

●Sign Up and Login

Sign Up and Login will be displayed in a common screen and the user has to enter only his/her email and password to create a new account as well as for login. The user can then access his account and edit all other details.

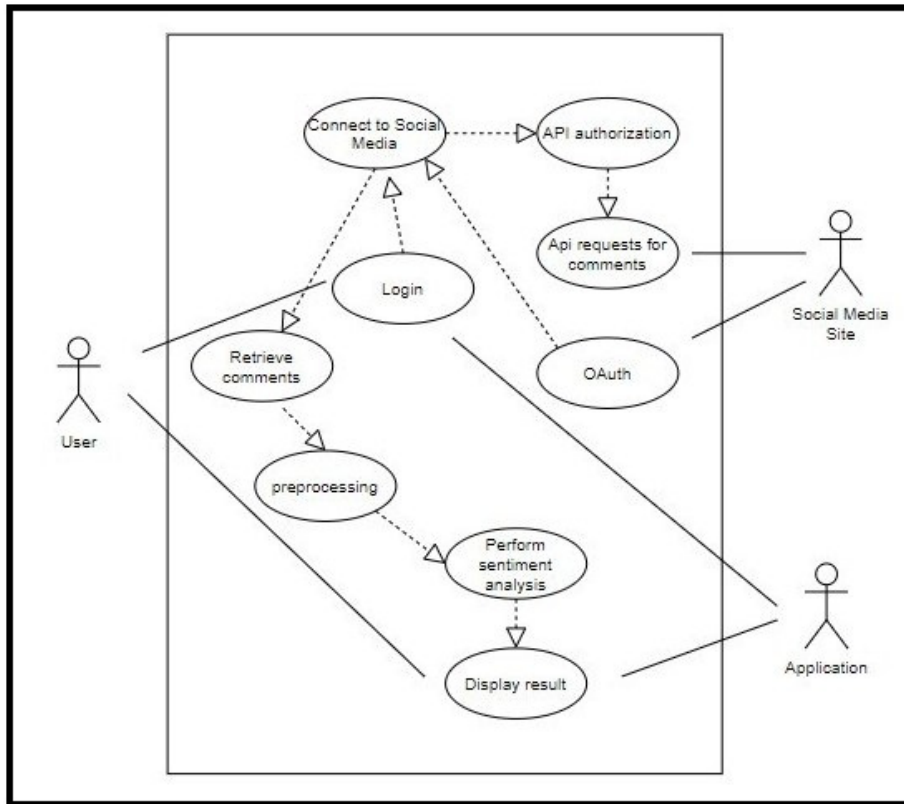
●User Dashboard

The user can view his details and edit his profile. From here he can navigate to various platform specific panels for viewing the analysis of their posts.

●Platform Specific Panel

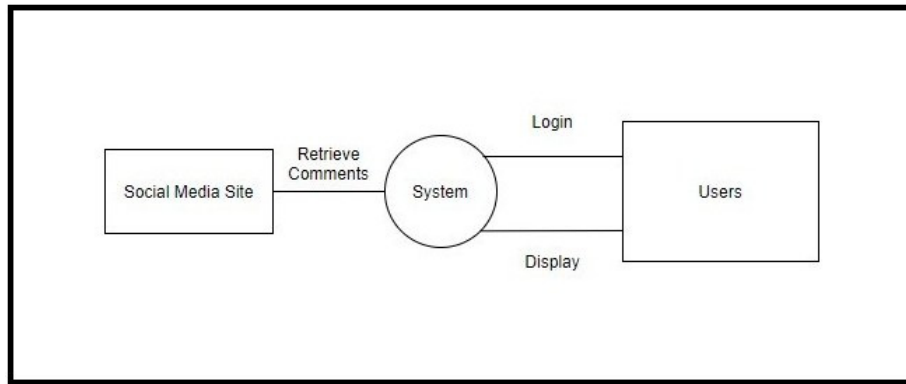
If the user is not logged in and not authenticated the web application, he will be redirected to the OAuth page for the specific platform. After a successful authentication, the user will be redirected to the panel page populated with the data and sentiment and sarcasm analysis done on the comments. The panel acts like a platform specific dashboard. The flagged comments will be highlighted to get user approval for reporting/blocking actions to be performed.

5.5 System Architecture



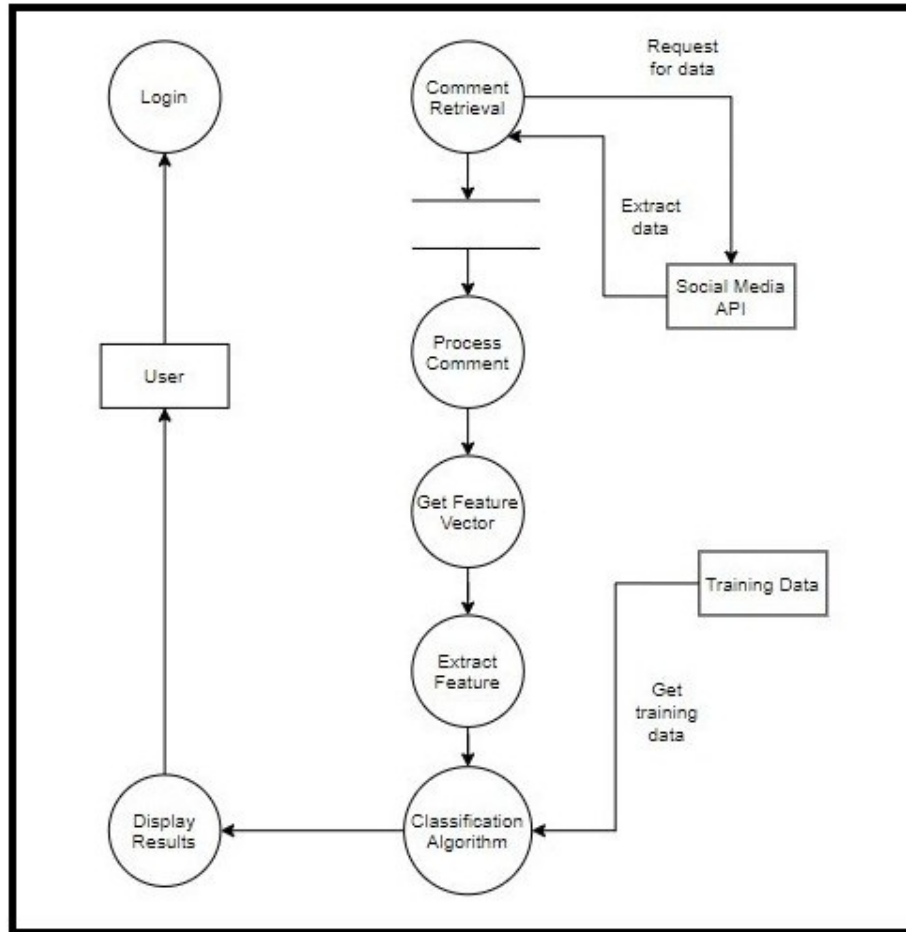
5.6 Data Flow Specifications

5.6.1 Level 0 DFD with description:



Context level data flow diagram depicts the relationship between the system and external data which behaves as source and sink. It is also known as "Level 0 DFD". Under a context diagram, the system interacts with outside agents. The entire system under level 0 DFD acts as a single process and there is no description about internal processing.

5.6.2 Level 1 DFD with description:



The high level data flow diagram shows the division of the system into sub-processes (systems), each describing the data flows with the outside world (agent). It also shows dataflow between each sub-system and performs internal data storage. The purpose of high level is to describe the major processes and their correlations. It can be balanced with its top parent level 0 DFD.

6 Software Test Document

6.1 Introduction

6.1.1 System Overview

The virtual social media managing platform will help the user to analyse the comments received on the posts created by the user. The current system used for monitoring social media platforms is manually analysing comments and deleting negative comments or manually blocking a regular spammer or hate promotor. This web application will eliminate the effort required to manually screen thousands of comments and block hundreds of spam users. The user has to register to the application and then provide permission to the web application for accessing his account by authenticating via the OAuth platform. The application then retrieves the comments on the posts. The suitable algorithms for sentiment analysis and sarcasm are applied on the comments and displays the aggregated results on the panel home. The user can search for specific posts and view analytics for those posts. The platform can also provide auto-replies for positive comments and reports/blocks negative comments. The application also blocks spam users. Hence, this application will highly contribute towards reducing online hate and help influencers to manage their social media profiles.

6.1.2 Test Approach

6.1.2.1 Testing Method Blackbox testing will be used to perform testing. Black Box is a testing method in which internal structure, design, implementation of the item being tested is not known to the tester. This method basically tests the functional requirements of the project. White box testing will be performed to test internal structure, design and coding of software to verify flow of input-output and to improve design, usability and security.

6.1.2.2 Testing Strategies The testing strategies used for our project are alpha testing. Alpha testing is performed by testers who are usually internal people of the team. As usually alpha testing involves both white-box and black-box testing, we will consider black-box testing as it is well suited for our project. In our project this testing will be done by our testers.

6.2 Test Plan

6.2.1 Features To Be Tested

Software modules that need to be tested are:

1. User Login: TC-1
2. Authorization of the web application: TC-2
3. User Dashboard: TC-3

4. Blocking/Reporting of malicious accounts: TC-4

6.2.2 Features Not To Be Tested

Automatic reply to comments: The testing of this feature is out of the scope of this project because it is a part of the services provided by the individual API of each social media platform.

6.2.3 Testing Tools and Environment

Manual approach as well as computerized approach would be used for testing of the application. The web application would be extensively used during the testing of this application. An active internet connection might be required during the testing of certain features. Microsoft Visual Code, CLI (Command line interface) and certain online open source testing tools will be used to perform the testing.

6.3 Test Cases

6.3.1 User Login: TC-1

6.3.1.1 Purpose To authenticate any user.

6.3.1.2 Inputs

1. Username
2. Password

6.3.1.3 Expected output and Pass/Fail Criteria It depends on the details entered by the user. After validating the inputs given by the user, then the user will be granted access to the application.

6.3.1.4 Test Procedure If an account is not created then the user will have to create an account and then login.

Constraint: Username and password used at the time of Sign Up should match.

6.3.2 Authorization of the web application: TC-2

6.3.2.1 Purpose Retrieval of comments and perform other actions using the respective user ID of the user.

6.3.2.2 Inputs API key provided to the developer.

6.3.2.3 Expected output and Pass/Fail Criteria A key will be generated to the developer. A failure will occur if and only if the application decides to dismiss the developer's request to generate an API key.

6.3.2.4 Test Procedure This module needs to be tested manually.
Constraint: Developer should have created an account for this.

6.3.3 User Dashboard: TC-3

6.3.3.1 Purpose To test the application for its usability and accessibility and implement the changes suggested by the tool

6.3.3.2 Inputs URL of the web application

6.3.3.3 Expected output and Pass/Fail Criteria The output would be in the form of a report that would grade the website on various parameters. The passing criteria would be to get a decent score on all the parameter grading.

6.3.3.4 Test Procedure Use an open source tool for testing the overall usability of the web application.
Constraint: The web application should be able to access the user's data.

6.3.4 Blocking/Reporting of malicious accounts: TC-4

6.3.4.1 Purpose The sentiment value generated by the algorithm will help us in identifying whether an account is abiding by the rules and guidelines of the application. If the values indicate that an account is posting vindictive comments then the account will be blocked.

6.3.4.2 Inputs User ID of the person posting hate comments.

6.3.4.3 Expected output and Pass/Fail Criteria The output obtained will be blocking of the account who is spreading hate on the website. If the blocking is done correctly then it's a success.

6.3.4.4 Test Procedure This will be tested manually with the help of the backend team.
Constraint: Only accounts violating the policies and guidelines of the social media platforms must be blocked.

7 Conclusion

The main aim here was to mitigate the spread of online hatred on social media platforms. The goal was to build a system that could filter the malicious comments and work hand-in-hand with the human moderators. This would help them reduce their workload and curb these kinds of activities. The technique used in Troll detection was with the help of Gated Recurrent Units and in sarcasm detection we used a 4 layered neural network mechanism. It was identified that we have a limitation in our work and that is in sarcasm detection it is difficult to make strong claims of our findings since we had a small dataset. In spite of having this limitation we assert that the model we have to offer is quite efficient in detecting sarcasm in comparison to the other models that currently exist. Better results can be achieved if more work is done in collecting data and test the models again for consistent and efficient success.

References

- [1] R. Gupta, J. Kumar, H. Agrawal and Kunal, "A Statistical Approach for Sarcasm Detection Using Twitter Data," 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2020, pp. 633-638, doi: 10.1109/ICICCS48265.2020.9120917.
- [2] Kaur, G.; Kaushik, A.; Sharma, S., "Cooking Is Creating Emotion: A Study on Hinglish Sentiments of Youtube Cookery Channels Using Semi-Supervised Approach" Big Data Cogn. Comput. 2019, 3, 37.
- [3] Sadeque, Farig, et al. "Incivility Detection in Online Comments." Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019). 2019.
- [4] L. H. Son, A. Kumar, S. R. Sangwan, A. Arora, A. Nayyar and M. Abdel-Basset, "Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network," in IEEE Access, vol. 7, pp. 23319-23328, 2019, doi: 10.1109/ACCESS.2019.2899260.

Acknowledgment

We are very pleased to present to one and all our project synopsis on Prevention of Cyber Troll & Sarcasm System on Social Networking using Machine Learning with Bilingual Analytics. We are very thankful to K.J Somaiya College Of Engineering, Vidyavihar for supporting our ideas and giving us an opportunity to express them. We would also like to express our special gratitude towards Dr. Shubha Pandit, Principal, K.J Somaiya College of Engineering, Vidyavihar. We are also very thankful to our guide Mr. Chirag Desai, Professor, Information Technology Department, KJSCE for his continuous guidance and support throughout the project.