Human-Computer Interaction

Assignment 2

-by Group 8

Group Members

Anish Borkar(210C2030036) (CSE 1)

Anoushka Srivastava(210C2030048)

(CSE 1)

Ankur Kaushal (210C2030218) (CSE 4)

Dyuti Dasmahapatra (210C2030142)

(CSE 1)

Godavarthi Sai Nikhil (210C2030030) (CSE 1)

Nichenametla Karthik Raja (210C2030113) (CSE 1)

Vandamasu Sai Sumanth (210C2030060) (CSE 2)

D Veera Harsha Vardhan Reddy (210C2030061)(CSE 2)

Himanshu Sharma(210C2030207) (CSE 4)

BML MUNJAL UNIVERSITY™

Department of Computer Science and Engineering
School of Engineering and Technology

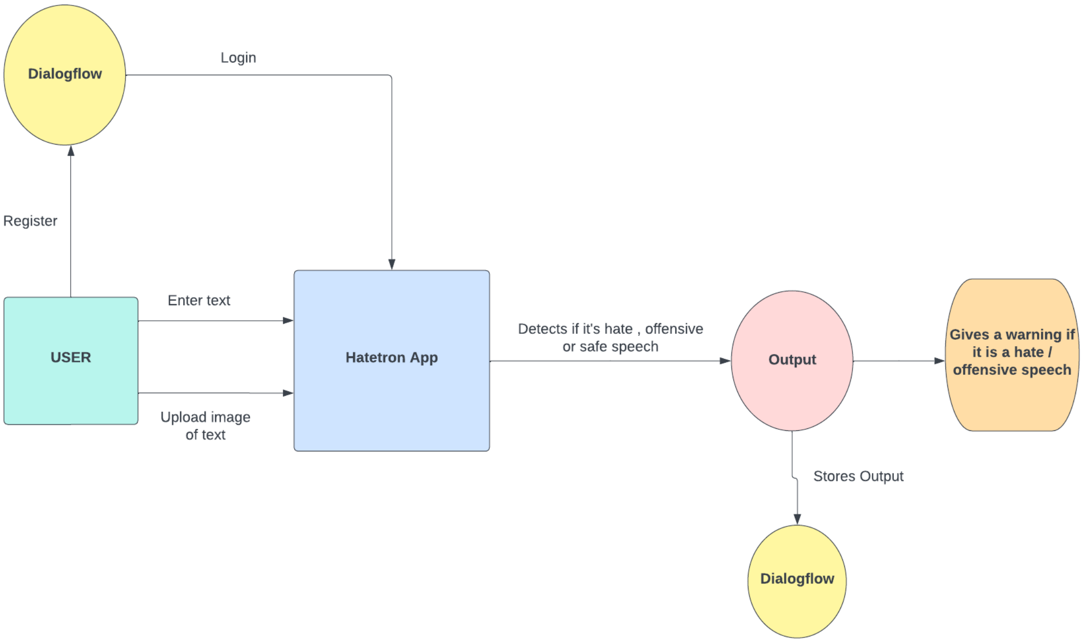
April 2023

1. Title and description of the product including high level solution design [use figure]

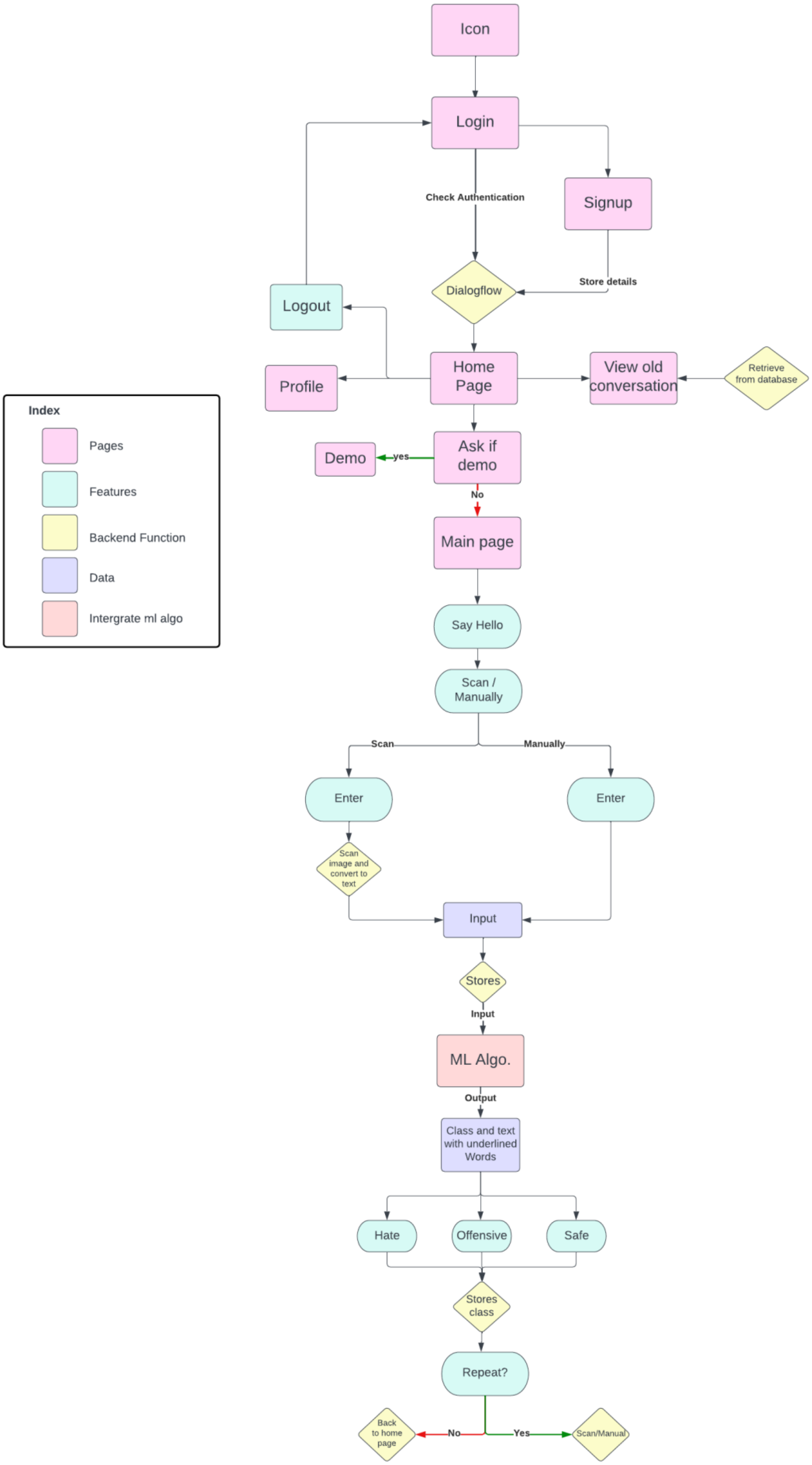
Name: Hate Speech Detection Chatbot or Hatetron

Description: Hate Speech Detection Chatbot App or Hateron is a system that can accurately identify and respond to instances of hate speech in real-time communications, while also respecting cultural and contextual nuances and protecting freedom of expression. The goal of hate speech detection is to provide a safe and inclusive online environment by reducing the spread of hate speech and promoting respectful communication.

High Level Solution Design:



If the user is new to the app, he or she registers. If not, he or she logs into the app. This is saved in the database by Dialogflow. The user has the option of manually entering the text or uploading an image of the text. The HATETRON app determines whether it is Hate Speech, Offensive Language, or Neutral Text. If the text entered or scanned contains Hate Speech or Offensive Language, an output warning is displayed. Dialogflow is used to save the output.


Icon
Login
Check Authentication
Signup
Store details
Dialogflow
Logout
Home Page
Profile
View old conversation
Retrieve from database
Ask if demo
Demo
yes
No
Main page
Index
Pages
Features
Backend Function
Data
Intergrate ml algo
Say Hello
Scan / Manually
Scan
Manually
Enter
Enter
Scan image and convert to text
Input
Stores
Input
ML Algo.
Output
Class and text with underlined Words
Hate
Offensive
Safe
Stores class
Repeat?
No
Yes
Back to home page
Scan/Manual

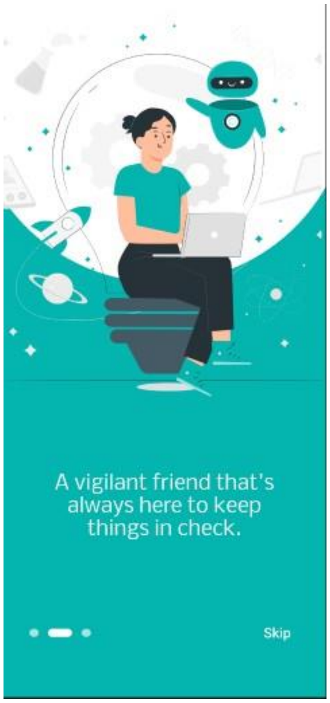
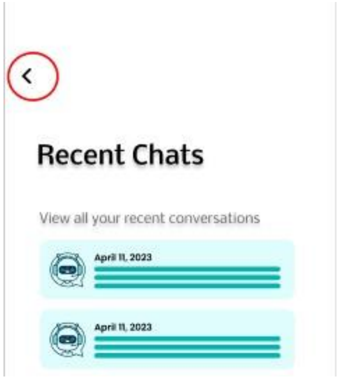
The user will open the app. The user will be taken to the icon screen before being redirected to the Login Page. If the user does not already have an account, he or she will proceed to the signup page and register. The information will be saved in the database. The user will be taken directly to the home page.

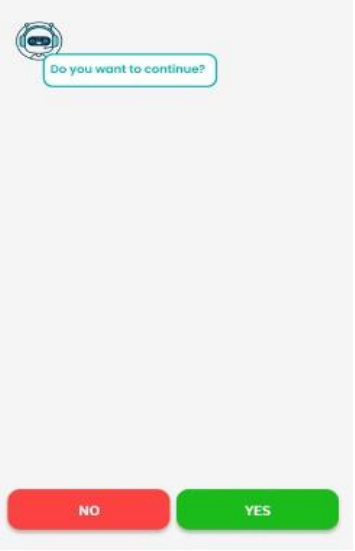
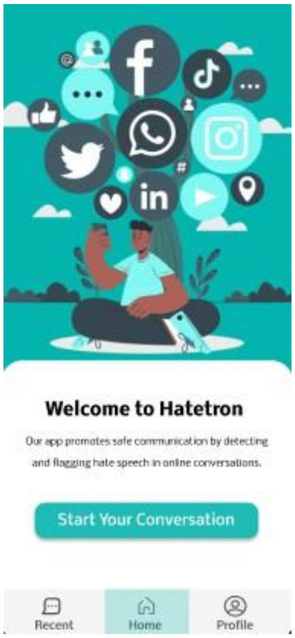
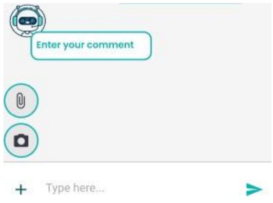
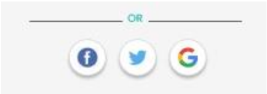
The user will be given several alternatives, such as starting a new chat, seeing recent conversations, starting a new conversation, viewing profile, or logging out. The user will be led to recent chats if they click the Recent button. The user will view the discussions after they are fetched from the database. If the user clicks on profile, the user will be shown details regarding the user. If the user clicks on logout, then the user will be redirected back to the login page.

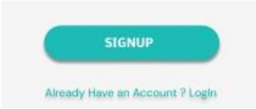
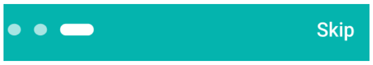
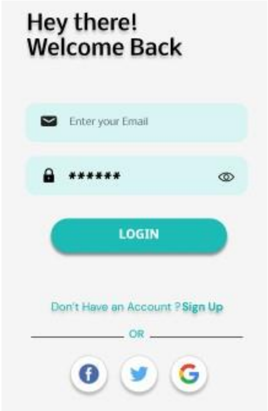
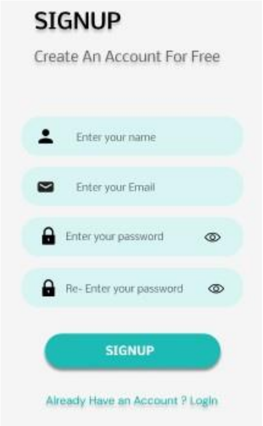
If the user clicks on 'Start a new conversation,' they will be asked if they require a demo. If yes, a demonstration of how to use the app will be shown. If no, the user is taken to the chat screen. The chatbot will begin the conversation by greeting the user. A button will be given that, when pressed, will send a greeting message. The chatbot will then ask the user whether they want to scan the text or type it in manually. Two buttons will be offered, from which the user can choose. The image will be processed to extract text from it in the scan option, while the text will be taken from the entered message when in the Manually entering mode. This text will then be saved in the database and fed into a machine learning algorithm, which will determine if it is hateful, offensive, or safe. The model will output the class it was classified into as well as the hate or offensive terms if it was classed as hate/offensive. The app will then display the warnings based on the class to the user. The class will then be saved in the database.

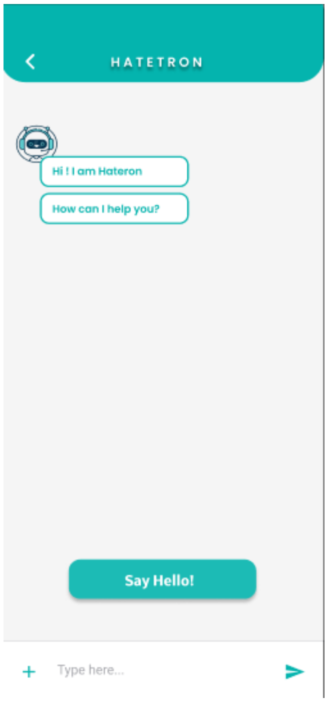
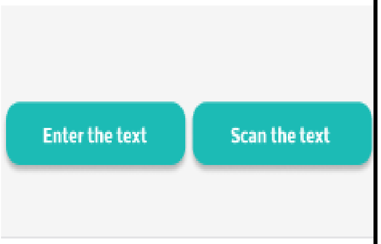
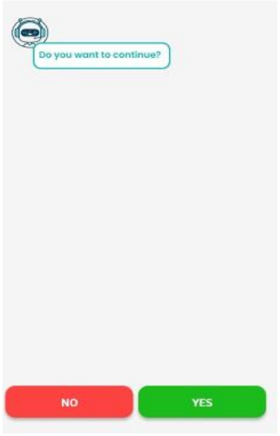
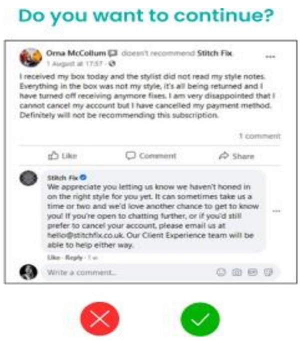
The app will then ask the user if they want to use it again. If yes, the scanning / manually entering process will be repeated; otherwise, the user will be led back to the main page.

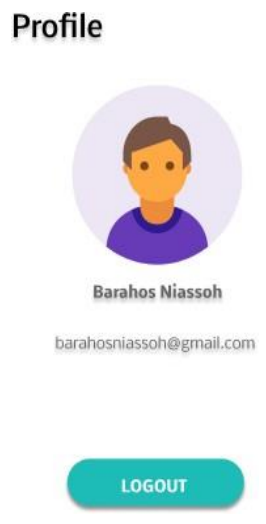
Q.2 Design principles used in your application – explain each feature with appropriate design principle, also explain the significance of using this design principle.In addition to this table, describe in detail each feature of the application.

Feature within the application	Associated Design Principle	Significance of the feature	Associated Image in the app
1)User friendly interface	User centered design	Aims to increase user satisfaction and management by making sure the app is simple to use, navigate, and understand for users of all ability levels.	 A vigilant friend that's always here to keep things in check.
2) Back option	Affordance, Consistency	When a user navigates backwards using the back button it redirects to the pages that user visited recently	 Recent Chats View all your recent conversations April 11, 2023 April 11, 2023

Feature within the application	Associated Design Principle	Significance of the feature	Associated Image in the app
3)Yes/No Buttons	Visibility	It allows the persons to actually continue into the app after acquiring some results	 Do you want to continue?
4) Start Your Conversation	Visibility, Affordance	It navigates to the front page of the app interface	 Welcome to Hatetron Our app promotes safe communication by detecting and flagging hate speech in online conversations.
5) Plus Button for camera and image attached for text detection	Visibility, Affordance, Feedback, Consistency	Gives options to either to capture a image or an attached image	 Enter your comment + Type here...
6) Login using Facebook, Google, or Twitter.	Affordance, Consistency, Visibility	The user can login in the application through facebook or google,twitter and facebook	

7) Sign Up	Affordance, Consistency,Visibility, Feedback	It redirects the user to sign in page where user enters the details like email, name and password	
8) Send button	Visibility , Affordance , Feedback , Consistency	This button transmits the input provided in the chat field.	
9) Type Here Chat field	Visibility , Affordance, Consistency, Feedback	The field enables the user to write text to chat with other users.	
10) Skip button on the three pages	Visibility,Affordance, Consistency	The user can skip the pages and directly head to the welcome page.	
11) Login via username and password; Login via Google, Facebook, or Twitter.	Constraints	Ensures that the user can login only after providing the correct username and password.	 Hey there! Welcome Back Enter your Email ****** LOGIN Don't Have an Account ? Sign Up OR
12) Creating Usernames and Passwords at the Signup Page	Constraints	Ensures that the username is unique and the password complies with the set constraints ; ensures its safe	 SIGNUP Create An Account For Free Enter your name Enter your Email Enter your password Re- Enter your password SIGNUP Already Have an Account ? Login

12) Starting a conversation	Constraints	A set command that is utilized to activate the chatbot.	 HATETRON Hi ! I am Hateron How can I help you? Say Hello! Type here...
14) Providing input via **Scan** or **Type** text feature	Constraints	Ensures the input is in the correct format and helps to provide accurate results.	 Enter the text Scan the text
15) Choose either the Yes or No option to continue or discontinue using the chatbot.	Constraints	It helps the chatbot know if the user wants to provide more input or not.	 Do you want to continue? NO YES  Do you want to continue? Orna McCollum doesn't recommend Stitch Fix 1 August at 17:57 I received my box today and the stylist did not read my style notes. Everything in the box was not my style, it's all being returned and I have turned off receiving anymore fixes. I am very disappointed that I cannot cancel my account but I have cancelled my payment method. Definitely will not be recommending this subscription. 1 comment Like Comment Share Stitch Fix We appreciate you letting us know we haven't honed in on the right style for you yet. It can sometimes take us a time or two and we'd love another chance to get to know you! If you're open to chatting further, or if you'd still prefer to cancel your account, please email us at hello@stitchfix.co.uk. Our Client Experience team will be able to help either way. Like · Reply · 1 w Write a comment...

16) Only one Logout option in the Profile Tab	Constraints and User-Centered Design	Only one logout option makes it easier for users to remember how to logout and tells the app that the user has logged out and needs to login again in the next session.	 Profile Barahos Niassoh barahosniassoh@gmail.com

Features in the app:

1) **User Registration:** An app's user registration feature is the procedure by which users can register an account within the app. This normally entails giving the app any additional information it may need, along with personal data like a name, email address, and password.

2) **Enter the text and Scan feature**: The app includes a few functions. things like "enter the text" and "scan text," which allow the user to determine whether the text detected is safe speech or offensive in terms of hate speech.

3) **Camera Icon and Attachment feature**: This feature contains a camera and attachment so that users may capture pictures and even attach pictures to have the text thoroughly scanned to determine whether it is hateful or objectionable.

4) **See recent text**: In the app, there is a feature that allows users to view their most recent chats and use them as notes.

5) **Profile Feature:** The Profile Tab contains the logout option and allows users to see their profile names, edit them, and log out.

6) **Login via Google, Facebook, or Twitter:** The ability to sign in using a social media account, such as Google, Facebook, or Twitter, simplifies the login process and frees users from having to remember their passwords every time they login.

3. Iterative design process – explain the usability issues identified at each stage of the product development.

The iterative design process involves collaborating with users to develop a prototype of the hate speech detection chatbot app. The prototype is tested with a small group of users, and feedback is collected to identify areas for improvement. This cycle continues until the app meets users' needs and objectives. The iterative process allows designers to create an effective and user-friendly app that promotes a safe and inclusive online environment.

Iteration	Date/ Month	Duration	Number of Participants	Type of Participants	Method of Feedback	Type of usability issue encountered	Improvementfrom previous iteration	Link to artifacts collected
1	2nd March	1 week	12	Students	Usability Testing and SUS	Difficulty in using the interface and unclear icons	N/A - First iteration	Link for SUS Result
2	10th March	3 days	4	Students	UX Heatmap	Navigation difficulties	Simplified interface design, redesigned icons	Screenshots of user testing UX Heatmap
3	13th March	1 day	5	General Public	Focus Groups	Poor terminology and poor typography	Simplified navigation	N/A
4	15th March	2 days	10	Students	Survey	Confusing feedback	Simplified terminology with consistent typography	Link to Google Forms
5	18th March	1 day	2	Design professionals	Expert Review	Poor color contrast and design	More intuitive feedback	Annotated Screenshots of design changes
6	20th March	1 week	6	All users	Focus Groups	Lack of features to complete key tasks	Improved color contrast and design	Feedback mentioned below
7	28th March	1 day	11	Students	SUS Report	Bad user experience due to long steps for completing key tasks	Improved by adding scan the text option	Link for SUS Report
8	1st April	3 days	16	All	Survey	difficulty in accessing/viewing previous records.	Improved user experience	Link to Google Forms

Link to artifacts collected

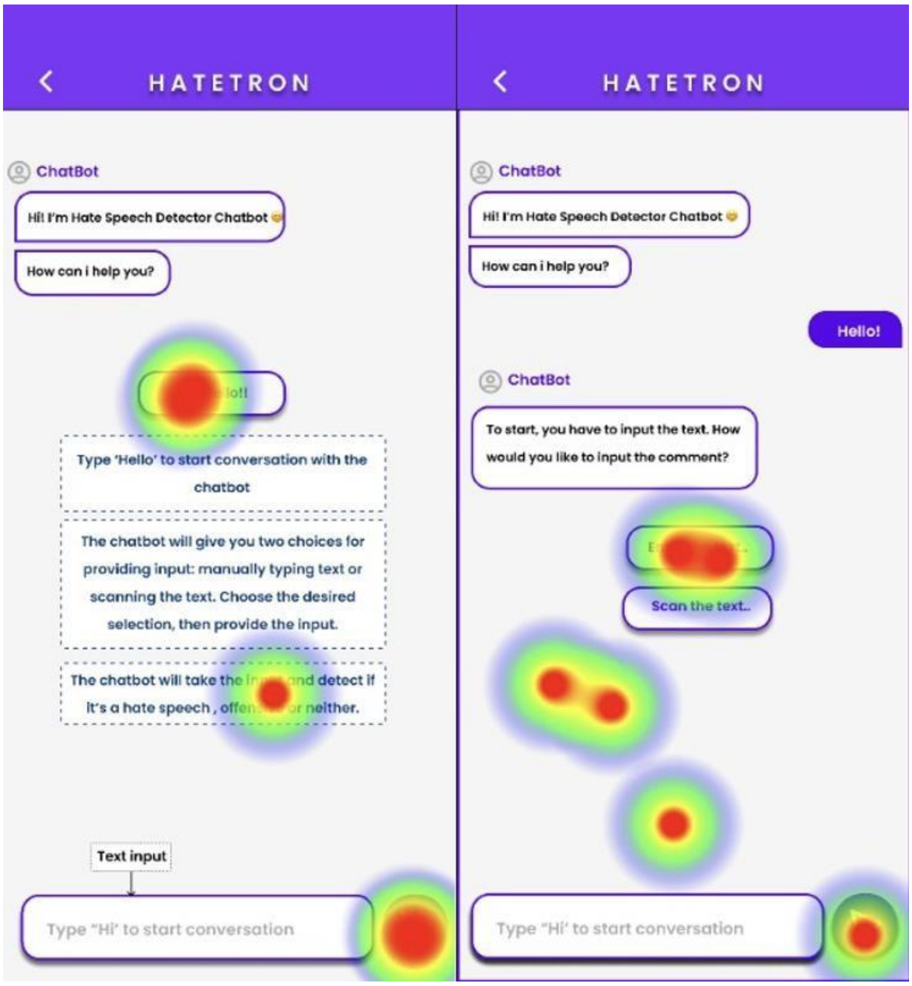
Iteration 1:

In the first iteration, we utilized the Maze application to conduct a survey where **12 users** were assigned a task to complete, and the report was generated based on their success rate.
The survey resulted in a **58%** success rate. During usability testing, participants reported difficulty using the interface, and some icons were unclear to them. Navigating through the product was difficult for participants as they were unsure of what certain buttons did. To collect feedback, we utilized both verbal feedback from participants during testing and the System Usability Scale (SUS) survey. We analyzed the data collected from the SUS survey to identify areas that require improvement. Additionally, we recorded verbal feedback for a later review. The link to the SUS result is Link for SUS Report

Iteration 2: Screenshots of user testing UX Heatmap

We conducted usability testing with students and used a UX Heatmap as the method of feedback. The usability issue encountered was navigation difficulties. Based on the feedback received from the UX Heatmap, we simplified the navigation to improve the overall usability of the prototype. Additionally, we collected screenshots of the user testing and the UX Heatmap to document our findings and track progress over time.
These are the screenshots of pages where users encountered difficulties, such as navigation.

User1:
We requested User1 to complete a task, and based on the data provided, we analyzed the heatmap and identified navigation issues on two specific pages.



Hi! I'm Hate Speech Detector Chatbot 😊

How can i help you?

Type 'Hello' to start conversation with the chatbot

The chatbot will give you two choices for providing input: manually typing text or scanning the text. Choose the desired selection, then provide the input.

The chatbot will take the i nd detect if it's a hate speech , offen or neither.

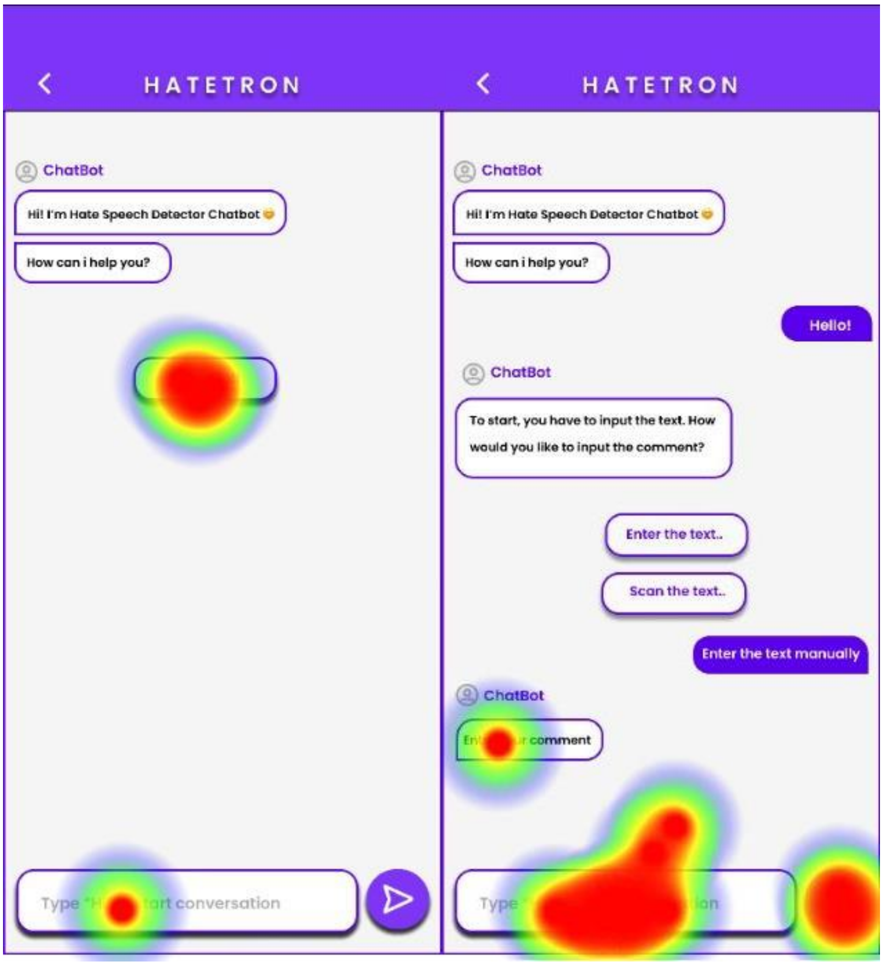
Hi! I'm Hate Speech Detector Chatbot 😊

How can i help you?

Hello!

To start, you have to input the text. How would you like to input the comment?

User2:



Hi! I'm Hate Speech Detector Chatbot 😊

How can i help you?

Type "H art conversation

Hi! I'm Hate Speech Detector Chatbot 😊

How can i help you?

Hello!

To start, you have to input the text. How would you like to input the comment?

Enter the text..

Scan the text..

Enter the text manually

En r comment

Type ion

Based on the user feedback, we confirmed that some users were having difficulty navigating through our design. Some users expressed confusion over the instructions provided, and others were unsure of how certain buttons and features were meant to function. We developed new designs and interaction models to improve user navigation.

Clearer instructions, intuitive button layouts, and effective visual cues were incorporated in the new design. Usability testing verified their effectiveness, showing that users can now navigate the product more easily and quickly understand its functions.

Iteration 3:

During the focus group, all users expressed problems and frustration with the terminology used throughout the interface. They felt that certain terms were confusing or unclear in the HateTron bot, which made it difficult to navigate and use it. Additionally, some users commented on the inconsistent typography, which made it difficult to quickly scan and read information.

Based on the feedback from the focus group, we simplified the terminology used throughout the interface and ensured that typography was consistent across all pages. We also made adjustments to font size and spacing to make the text easier to read.

Iteration 4:

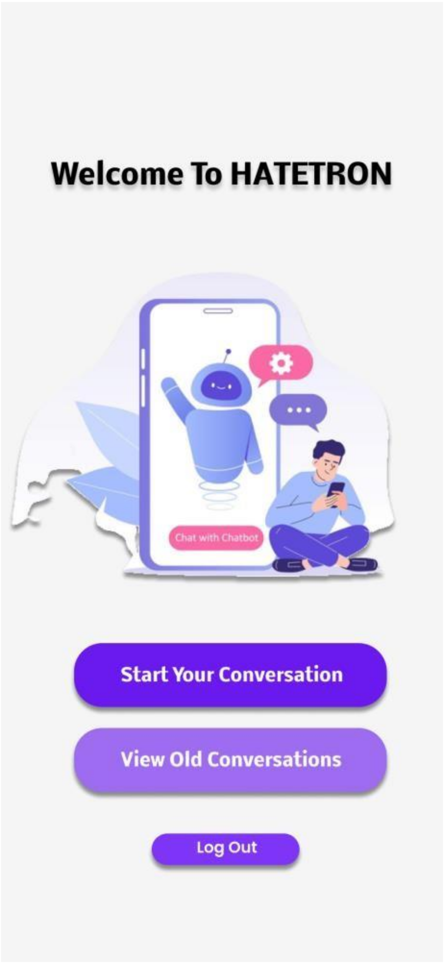
According to Survey, confusing feedback in the app was the most common issue identified after reviewing the Figma prototype of the hate speech detector chatbot. This suggests that the app's messaging and user interface design may need to be revised to provide more clear and intuitive feedback to users. Improving the app's feedback could enhance user experience and increase the effectiveness of the hate speech detection system. To address this issue, we worked to make the feedback more intuitive by providing more contextual information about why a certain input was flagged. We also incorporated feedback from human-computer interaction research, which suggests that clear and concise feedback is essential for users to understand and interact with a product effectively.

Iteration 5: Annotated Screenshots of design changes

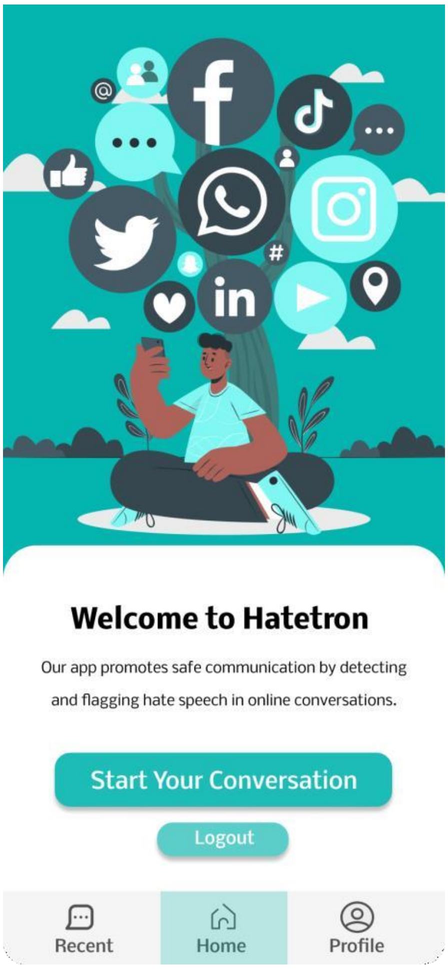
We requested two design professionals to give an expert review of the prototype. It was identified that the prototype had poor color contrast and design, which could impact the product's usability. To address this issue, our team members improved the color contrast and overall design of the prototype.

The Annotated screenshots of the design changes were displayed below:

In the images displayed above, we have presented a comparison between the initial prototype (on the left) and the updated prototype (on the right).

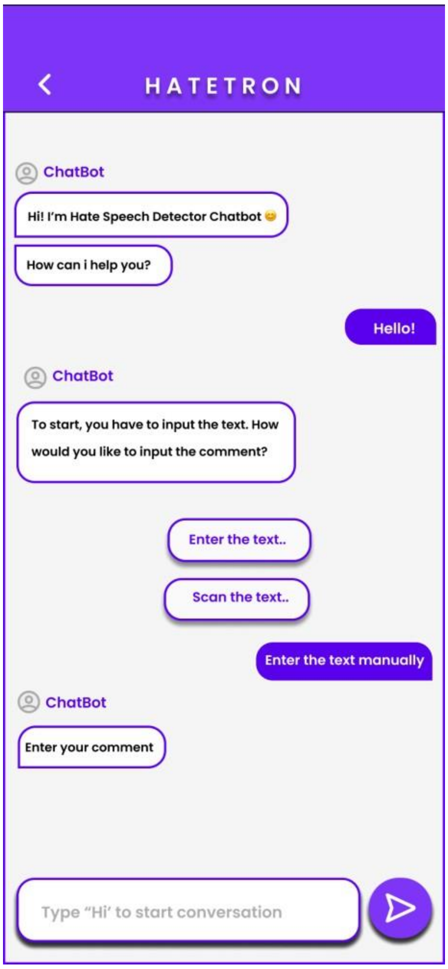


Welcome To HATETRON



Welcome to Hatetron

Our app promotes safe communication by detecting and flagging hate speech in online conversations.



ChatBot

Hi! I'm Hate Speech Detector Chatbot 😊

How can i help you?

Hello!

ChatBot

To start, you have to input the text. How would you like to input the comment?

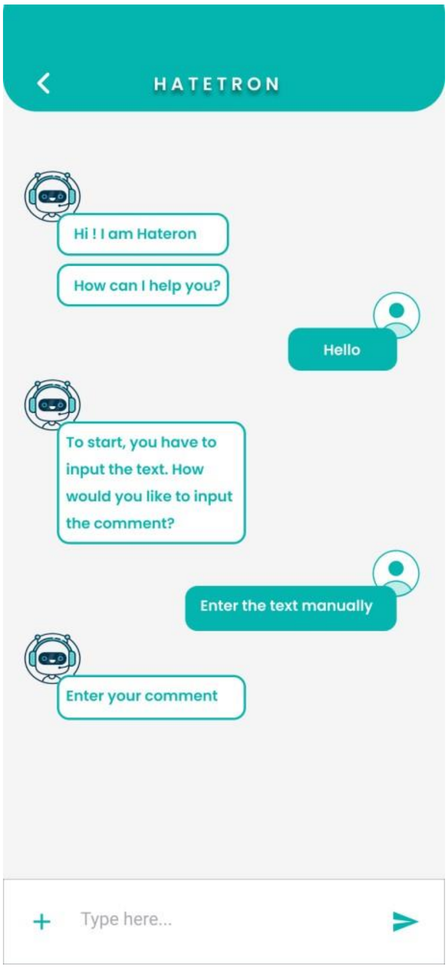
Enter the text..

Scan the text..

Enter the text manually

ChatBot

Enter your comment



Hi ! I am Hateron

How can I help you?

Hello

To start, you have to input the text. How would you like to input the comment?

Enter the text manually

Enter your comment

After making the necessary updates to the prototype based on the previous feedback, we conducted further user feedback testing, which yielded positive results.

Iteration 6:

Users found the chatbot prototype lacking in features to detect and flag hate/offensive speech. Users were frustrated with the lack of key features, especially in scanning text for hate speech/offensive language. This issue was identified through user behavior observations and verbal feedback in a focus group. We added a "scan text" option to address the usability issue, based on user feedback and human-computer interaction principles.

Iteration 7:

We used the Maze application to conduct a survey with **11 students**, in which they were assigned a task to complete a Figma prototype of a chatbot hate speech detector. The survey report showed that only **73%** of users were able to successfully complete the task, indicating that the prototype had usability issues.
We found that the length and complexity of certain tasks were causing frustration among users, as identified through an SUS report. To improve the user experience, we simplified the navigatio n and reduced the number of steps required to complete key tasks. These changes were based on the principles of human-computer interaction, which prioritize ease of use and minimizing cognitive load. The link to the SUS result is Link for SUS Report

Iteration 8:

We conducted a google survey with **16 users** of different backgrounds to gather feedback on the hate speech detector chatbot. We assigned tasks to users to complete, and the responses generated a report based on how the user performed. We found that users encountered difficulty in accessing and viewing previous records within the app. This issue was identified through user responses to the survey questions.

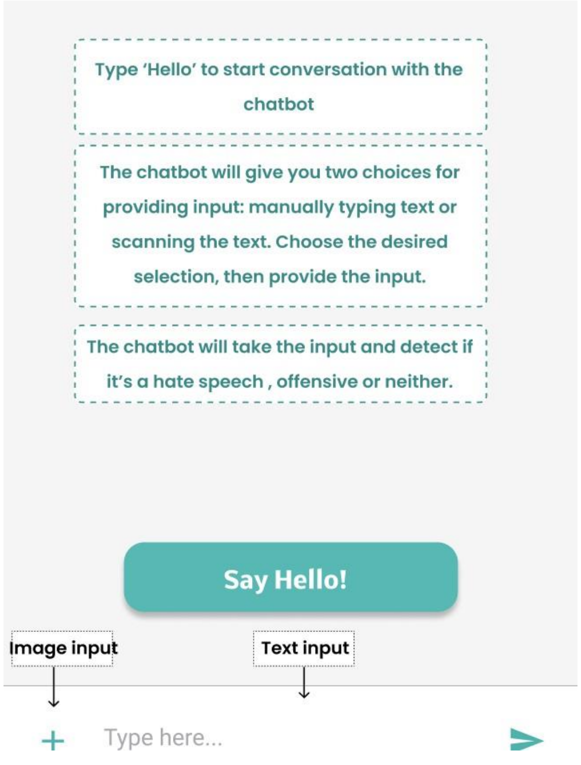
We improved the interface by adding a search function and a clear history button and simplified the language and provided more contextual information based on survey feedback and human-computer interaction principles.

The link to the google responses is Link to Google Forms

4. Identify the following parameters related to the usability of your application. Explain in detail how you ensure these usability guidelines are met in your application. You can talk about qualitative and quantitative evaluation of the above. Quantitative will get more weightage.

1. Learnability:

- A brief training for newcomers: This feature gives a brief tutorial for new users, helping them in understanding the app's features and functioning, hence improving learnability.

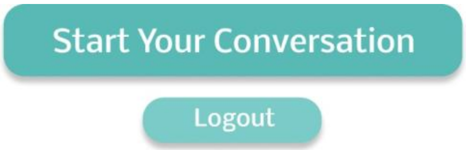


Type 'Hello' to start conversation with the chatbot

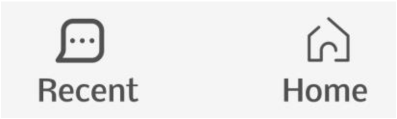
The chatbot will give you two choices for providing input: manually typing text or scanning the text. Choose the desired selection, then provide the input.

The chatbot will take the input and detect if it's a hate speech , offensive or neither.

- Buttons which state their purpose: This feature assists users in understanding the functionality of each button, making it easier for them to learn how to use the app properly.

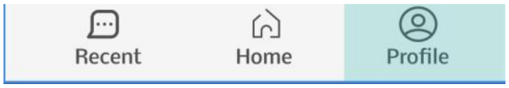


- Icons that correspond to their function: This feature uses icons that closely mimic their intended function, allowing users to immediately grasp the functioning of each icon.

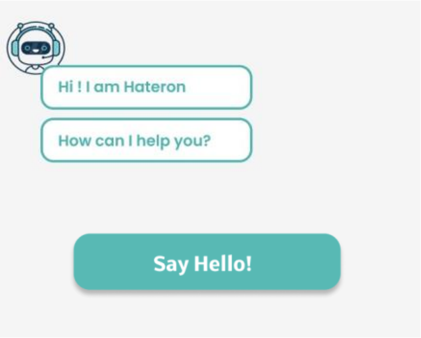


2. Efficiency:

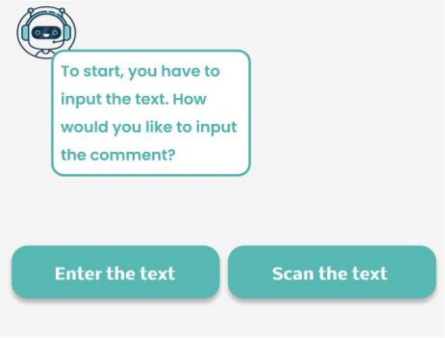
- Bottom Navigation Bar: This feature allows users to easily browse between various screens of the app, eliminating the steps required to switch between screens.



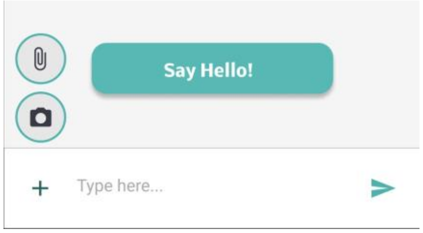
- The users have to just click the button and follow what the app says. Most of the work will be done by the app: This feature minimizes the effort required from users as they simply need to click buttons and follow the app's instructions, making the interaction efficient and straightforward.


Hi ! I am Hateron

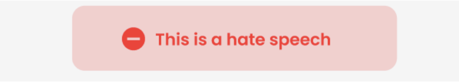
How can I help you?

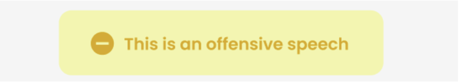

To start, you have to input the text. How would you like to input the comment?

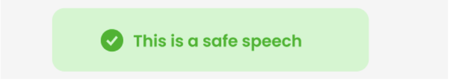
- Image to text converter: With this function, the user may just upload a screenshot of a comment and the app will convert the image to text, making the app more efficient to use



- Color-coded warnings: The use of color-coded warnings can facilitate efficient information processing, as users can quickly interpret and differentiate between different types of hate speech based on the visual cues provided by the colors. This can save time and effort in identifying and responding to potential hate speech.

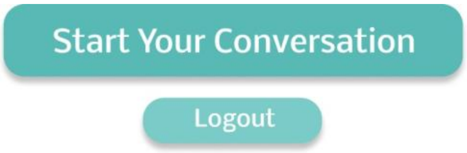

This is a hate speech


This is an offensive speech


This is a safe speech

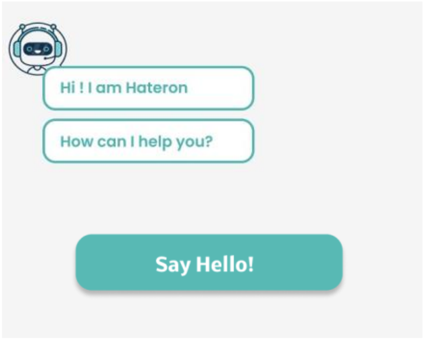
3. Memorability:

- Clearly labeled buttons: This feature assists users in understanding the functionality of each button, making it easier for them to learn how to use the app successfully.



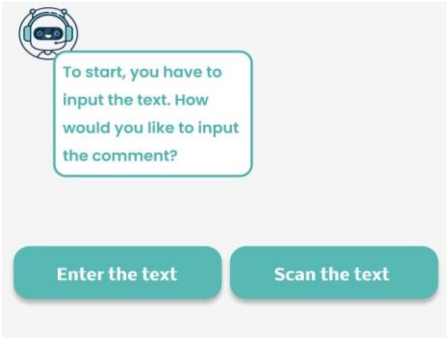
- Icons that correspond to their function: This feature uses icons that closely mimic their intended function, allowing users to immediately grasp the functioning of each icon.

- Contextual guidance: The app offers contextual guidance and hints at each step to assist users in understanding the information required and how to provide it. This helps users remember the necessary details needed for accurate hate speech detection, making it easier for them to provide the necessary information for accurate hate speech detection.



Hi ! I am Hateron

How can I help you?

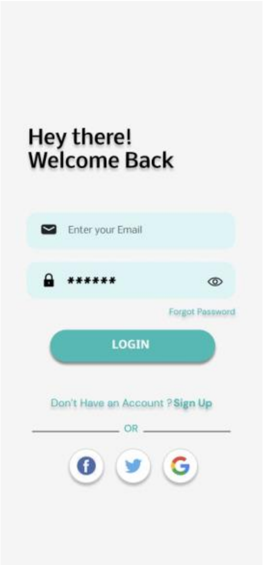


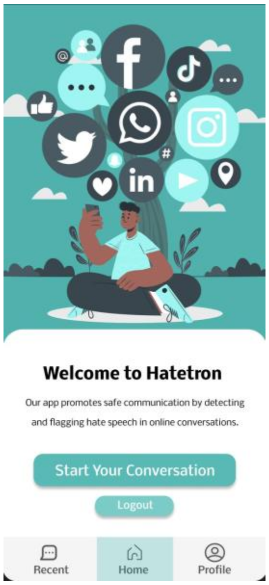
To start, you have to input the text. How would you like to input the comment?

4. Satisfaction:

- Responsiveness:
  This feature makes the app cross-platform compatible, with a responsive design that adapts to different screen sizes and orientations. It provides a consistent user experience across platforms, while leveraging platform-specific features, resulting in a seamless user experience and higher satisfaction.
- Accessibility:
  This app is made accessible to a wide range of users, including those with disabilities or limited technical skills. This improves customer satisfaction by making the application inclusive and easy to use for everyone.

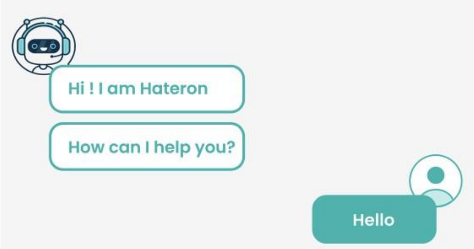
- User friendly design:
  The app has a clean and intuitive interface with a simple and easy-to-understand layout and will be able to navigate through the app without confusion.It provides clear instructions and prompts to guide users on how to interact with it effectively. Instructions are concise and easy to follow, helping users to use the app efficiently and achieve their desired outcomes, building user satisfaction.
- A neat and clean interface with only the necessary components:
  This feature ensures that the app's UI is clutter-free and only contains necessary components, lowering cognitive load and making it easier for users to remember how to use the app.




Welcome to Hatetron

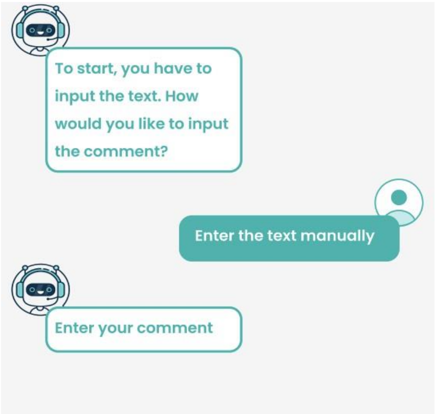
Our app promotes safe communication by detecting and flagging hate speech in online conversations.

- Quick and Accurate Detection:
  The app's ability to rapidly and precisely identify hate speech content is designed to enhance user satisfaction. The use of advanced algorithms enables the app to quickly and accurately flag hate and offensive speech incidents, providing users with a sense of satisfaction and trust in the app's capabilities.
- Conversational experience:
  Apps can mimic human-like conversations, giving users an engaging and entertaining conversational experience. Instead of using traditional forms or interfaces.This can increase user satisfaction by creating a more engaging and dynamic interaction.


Hi ! I am Hateron

How can I help you?

Hello

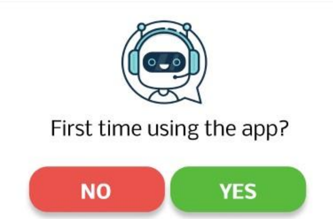

To start, you have to input the text. How would you like to input the comment?

Enter the text manually

Enter your comment

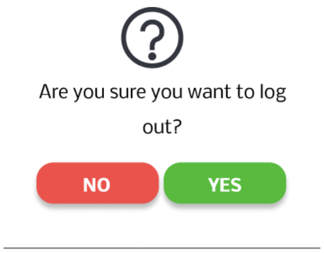
5. Error Protection:

- Red for no and green for yes buttons:
  This feature uses color-coded buttons, with red indicating "no" and green indicating "yes," which helps users avoid errors by providing clear visual cues for their selections.



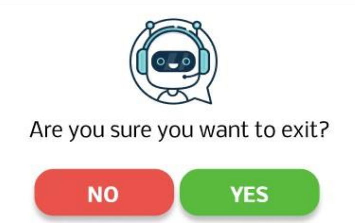
First time using the app?

- Warning Prompt before logging out:
  This feature helps prevent users from accidentally logging out of the application and potentially losing unsaved data or progress. By providing a warning prompt before logging out, the app protects users from making unintentional errors and promotes a positive user experience by mitigating the risk of data loss.



Are you sure you want to log out?

- Warning Prompt before exiting from chat:
  If a user accidentally clicks the back button during an ongoing chat session, the app displays a confirmation prompt asking the user to confirm their intention to leave the chat. This can prevent accidental exits and provide an opportunity for users to reconsider their action.



Are you sure you want to exit?

6. Utility:

- Accurately detect and flag hate & offensive speech content: The utility of the app lies in its ability to accurately detect and flag hate,safe & offensive speech content. This can be achieved through advanced algorithms that can effectively identify hate & offensive speech.
- Timeliness: The utility of the app also depends on its ability to provide timely and accurate warnings or alerts to users when hate speech is detected, allowing them to take appropriate actions.

Qualitative and Quantitative evaluation

1. Learnability:

Qualitative Evaluation:

- Usability Testing: Conducting usability testing sessions with representative users to observe how they interact with the app for the first time, observing their behavior and identifying any difficulties or challenges they face while using it. This can provide insights into the ease of learning and the effectiveness of the app's design.
- User Interviews: Conducting interviews with users to gather their feedback on their experience with learning to use the app. Asking questions related to their understanding of the app's functionalities, ease of use, and any barriers they encountered during the learning process.
- User Feedback Sessions: Conducting feedback sessions with users to gather their opinions on the app's learnability. Asking them to provide feedback on the app's interface, navigation, and overall ease of use. Their insights can help identify areas for improvement in terms of learnability.

Quantitative Evaluation:

- Completion Rates: Measuring the percentage of users who are able to complete tasks or achieve goals using the app without encountering errors or seeking external help. Higher completion rates indicate better learnability.
- Task Completion Time: Measuring the time taken by users to complete tasks or achieve goals using the app. Lower task completion times indicate faster learning and better learnability.
- Error Rates: Measuring the frequency and severity of errors encountered by users while using the app. Lower error rates indicate better learnability.
- User Ratings: Collecting ratings from users regarding their perception of the app's learnability, such as asking them to rate the ease of learning on a scale. Higher user ratings indicate better learnability.

2. Efficiency:

Qualitative evaluation:

- Usability Testing: Conducting usability testing sessions with representative users to observe how quickly and easily they are able to achieve their goals or complete tasks using the app. Identifying any challenges they face or areas where the interaction flow can be improved to enhance efficiency.
- User Interviews: Conducting interviews with users to gather their feedback on the efficiency of the app’s responses, navigation, and overall interaction flow. Ask questions related to their perception of the app's speed and effectiveness in helping them achieve their goals.

- User Feedback Sessions: Conducting feedback sessions with users to gather their opinions on the efficiency of the app’s performance. Asking them to provide feedback on the speed of responses, ease of navigation, and overall efficiency of their interactions with the app.

Quantitative evaluation:

- Number of Steps: Measuring the number of steps or interactions required for users to achieve their goals or complete tasks using the app. Fewer steps or interactions indicate a more efficient interaction flow.
- Task Completion Time: Measuring the time taken by users to complete tasks or achieve goals using the app. Lower task completion times indicate more efficient interactions and goal achievement.
- User Ratings: Collecting ratings from users regarding their perception of the app's efficiency, such as asking them to rate the speed and effectiveness of the app on a scale. Higher user ratings indicate higher efficiency.

3. Memorability:

Qualitative evaluation:

- Usability Testing: Conducting usability testing sessions with representative users who have used the app before and observe if they are able to remember how to use the app effectively. Identifying any difficulties they face in remembering the app's functionalities or navigation.
- User Interviews: Conducting interviews with users who have interacted with the app in the past to gather their feedback on the app's memorability. Asking questions related to their ability to recall how to use the app, and any issues they faced in remembering its functionalities or interaction flow.
- User Feedback Sessions: Conducting feedback sessions with users to gather their opinions on the memorability of the app. Asking them to provide feedback on how easily they were able to remember how to use the app after their initial interactions.

Quantitative evaluation:

- Recall Rate: Measuring the rate at which users are able to recall the app's functionalities or navigation without assistance or prompts. Higher recall rates indicate better memorability as users are able to remember how to use the app effectively.
- Task Completion Rate: Measuring the rate at which users are able to successfully complete tasks or achieve goals using the app after their initial interactions. Higher task completion rates indicate better memorability as users are able to recall how to use the app effectively.
- Error Rate: Measuring the rate at which users make errors or mistakes while using the app after their initial interactions. Lower error rates indicate better memorability as users are able to remember how to use the app without making frequent mistakes.

4. Satisfaction:

Qualitative evaluation:

- User Interviews: Conducting interviews with users to gather their opinions and feedback on their overall satisfaction with the app. Asking open-ended questions about their experience, satisfaction with the app's performance, and any areas for improvement.
- Focus Groups: Organizing focus groups with a diverse set of users to discuss their perceptions, opinions, and satisfaction with the app. Encouraging group discussions to capture different perspectives and gather insights on user satisfaction.
- Heuristic evaluation: It can provide valuable feedback on the app's design from a usability perspective, helping to identify areas for improvement and enhance user satisfaction.

Quantitative evaluation:

- User Satisfaction Surveys: Administer standardized surveys to users, asking them to rate their satisfaction with the app's performance, usability, and overall experience using a numerical or Likert scale. Analyzing the survey responses to obtain quantitative data on user satisfaction.
- Net Promoter Score (NPS): Use the NPS metric to measure user satisfaction by asking users to rate the likelihood of recommending the app to others on a scale of 0 to 10. Calculate the NPS score to gauge overall user satisfaction.
- Usage Metrics: Analyze usage metrics such as the frequency of app interactions, average session duration, and completion rates of tasks or goals to assess user satisfaction. Higher engagement and completion rates may indicate higher user satisfaction.

5. Error Protection:

Qualitative evaluation:

- Usability testing: Conducting usability testing with a diverse group of users can help identify which parts of the application are most prone to errors and where users may need additional protection. Observing users as they use the application and collecting feedback can provide valuable insights into how to make the application more error-resistant.
- User interviews: Conducting interviews with users who have encountered errors or mistakes while using the application can provide insights into the causes of those errors and how they can be prevented in the future. This can help guide the development team in improving the error protection of the application.
- Heuristic evaluation: Having experts in the field of user experience design evaluate the application can provide valuable insights into how to make it more error-resistant. These experts can provide feedback on the design, layout, and functionality of the application, and suggest changes that can improve its error protection.

Quantitative evaluation:

- Error analytics: Tracking the frequency and types of errors that users encounter while using the application can provide insights into which parts of the application may need additional error protection. Analyzing this data can help the development team identify patterns and trends that can guide improvements.
- A/B testing: Testing different versions of the application with different levels of error protection can provide insights into which approaches are most effective in reducing errors. By measuring the frequency of errors and user satisfaction with each version, the development team can determine which approach to error protection is most effective.
- System monitoring: Continuously monitoring the performance of the application and its underlying systems can provide early warning signs of potential errors or issues that can be addressed before they affect users.

6. Utility:

Qualitative evaluation:

- User testing: Conducting usability testing with a diverse group of users can help identify which parts of the application are most useful and which may need improvement. Observing users as they use the application and collecting feedback can provide valuable insights into how to make the application more useful.
- User interviews: Conducting interviews with users who have used the application can provide insights into which features they found most useful and which features they may not have found useful. This can help guide the development team in improving the utility of the application.
- Focus groups: Bringing together a group of users to discuss their experiences with the application can provide a deeper understanding of the utility of the application. This can help identify common pain points and areas for improvement.

Quantitative evaluation:

- Usage analytics: Tracking how frequently customers use different features of the application can provide insights into which features are most useful and which may need improvement. This data can be used to identify which features are popular and which may need improvement.
- A/B testing: Testing different versions of the application with different features or levels of functionality can provide insights into which features are most useful to users. By measuring user engagement with each version, the development team can determine which features are most effective.
- Conversion rate: Measuring the conversion rate of users who use the application to complete a specific task or goal can provide insights into the utility of the application. By measuring how many users complete a task or goal, the development team can determine whether the application is meeting the needs of its users.