**WORD TO EMOJY TRANSLATOR &**

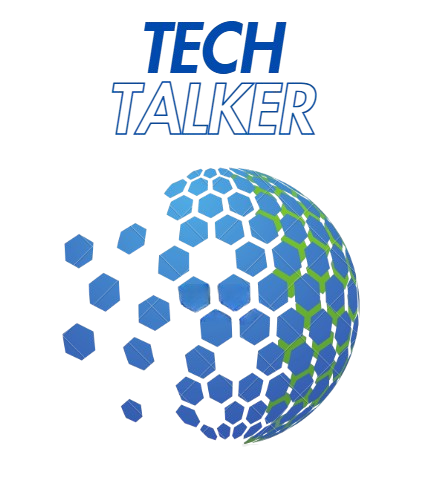
**MESSENGER**

About:

In this project we had to create a web app in which when we enter the text and sent it to the user then all the text is converted into the emojis.

It is used as the short steno types conversation between two different users in real time

Project Over View:

* In this Project We create a web app in the **React** with the **flask** in the server side, the react app is accessed by creating account from **sign in page** and then **login** the data through it.
* No token/ refresh cookies, authentication except password is set because of the time complexity.
* A user can add the person with whom he/she wanted to talked with using the add sign in the left corner sidebar.
* All users are Accessible in the user section. A **search** is also given which can be used to search the user from the list
* Chat pages open when we select the user and send chats are located at right and received at the left
* User can type and then able to send text through the bottom input box given for him
* A sign out page is created for logging out which is accessed through the user profile present at the header of the page.
* A Custom button is placed at the bottom-left of the page for customising the colour as the user preferred to use.

Technology User:

1. **a. React:** We used the react and its several libraries for delivering better program in less time.

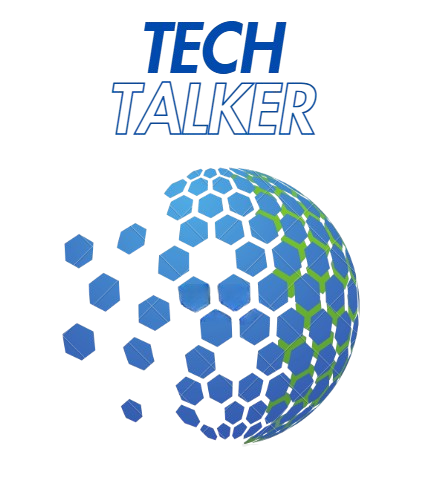
b. React is a powerful JS framework that help to create single page website and this is the reason why we used it for this, singe single page website is one of nicest for the applications.

c. It also provides several libraries that help in efficient and faster development.

1. **Python:** We used the python at the server end which increase the development speed but also provide vast opportunity for any task performance in the server side in the future.

Python with MySQL and SQL alchemy provides a single, fast and efficient data management. It can also be integrated with the MongoDB and firebase Api for further development in the future.

1. **Flask**: flask is a part of Python but here it also playing crucial role, flask proves the server and help in fetching and providing the API to the client for the sign and login request.
2. Socket.io: socket.io is used to create a web socket in server and communicate with the client through other socket of the react, it proved a fast and real time data access.
3. **My SQL:** my SQL data base is used here for storing the information of the users such as the friends, chats and the basic details of the user.
4. **Redux:** React Redux is used to manage the several states in the react app, it is not necessary to user but we used it for customisation of colour and single time data fetching of the main user.
5. **Other Libraries and modules**: several other modules are also used in it such as the react icons , datetime, date. React-Router-dom, etc. which are not mentioned here.



About Us

Team Name: Tech Talker

1. Abhay Pratap Singh (CSE Vth Sem)
2. Sakshi Rawat (CSE Vth Sem)
3. Piyush Amola (CSE Vth Sem)
4. Dakshita Pandey (CSE IIIrd Sem)

We all are student from the GP Kaladhungi and work on different expertise.

We all shows same interest toward the hackathon for testing ourself in this. We tried to give our best without cheating with ourself.

We manage to complete this task in around 2 days with the mid term examination in the morning time.

We feel very for this that we at least tested ourself out for some tasks like that with the pressure of minor, midterm and practical.

Resources Used:

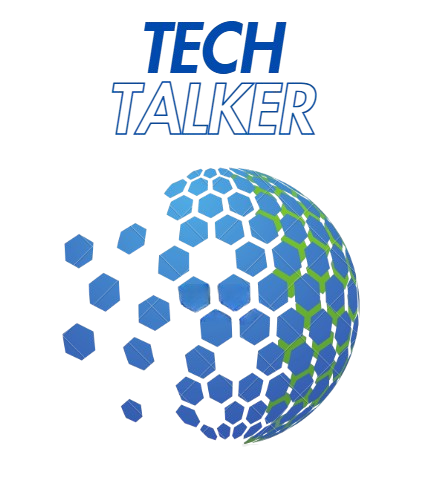
Server:

1. Flask + Python
2. My SQL

Client:

1. React
2. Redux

External:

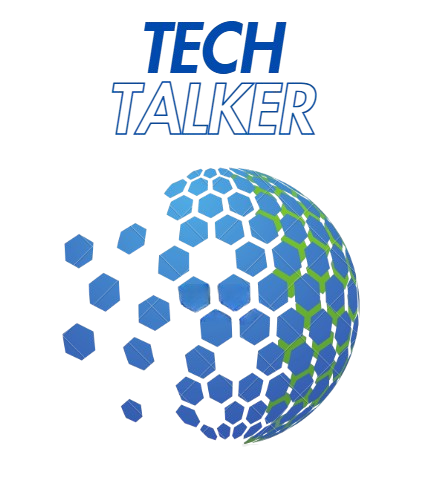
* Syntax of the Async Thun of react Redux
* Syntax of the SQL Alchemy in flask
* Web Socket boiler Plate code for using it in React and Flask with example
* Some SQL query like setting up database and table to handle unison code of 4 bytes.
* Emoji Array for searching and replacing words manually rather than using other medias and AI APIS.
* Open AI Integration code but discarded due to 429 error API problem.

Old Programs:

* Syntax of the Async Thunk
* Store and React-Router-Dom configuration
* Syntax of API creation in flask

Challenges Faced:

* **Mid Term Examination:** our midterm examination is going on these days hence we are not giving our 100% in both location
* **No WebSocket Experience:** No one had worked with the WebSocket yet so it is found to be difficult to preform complex task in it and hence resulted in the Bugs and a lot time wastage
* **Late Problem and limitation Clearness:** the Real Problem with limitation are declared late and confusing limitation makes to think several approaches that is unusable once final limitation releases.
* **Time:** Due to Mid term and some person works of us are lacking in the time.
* **Wasted time on finding OPENAI API’s (free) and Studying about WebSocket implementation:** we wasted around 3-6 hours on setting up and finding the OpenAI Api for emoji conversion through it but the problem arrives with 429 error and then we leave that topic and tried manual approach since there is not that much big time



Future Scopes:

1.In future the program has a lot of scopes since several parts such like ai configuration and ai with manual conversion code is already placed there commented so, it can be further developed.

2. Using Flask are the backend provide a lot of flexibility to the user due to its flexible nature of it.

3. Use of React Provide reusability of the code making it easy to manage in future.

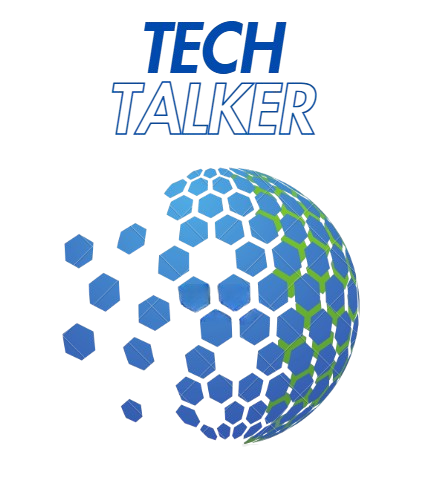
Time Line:

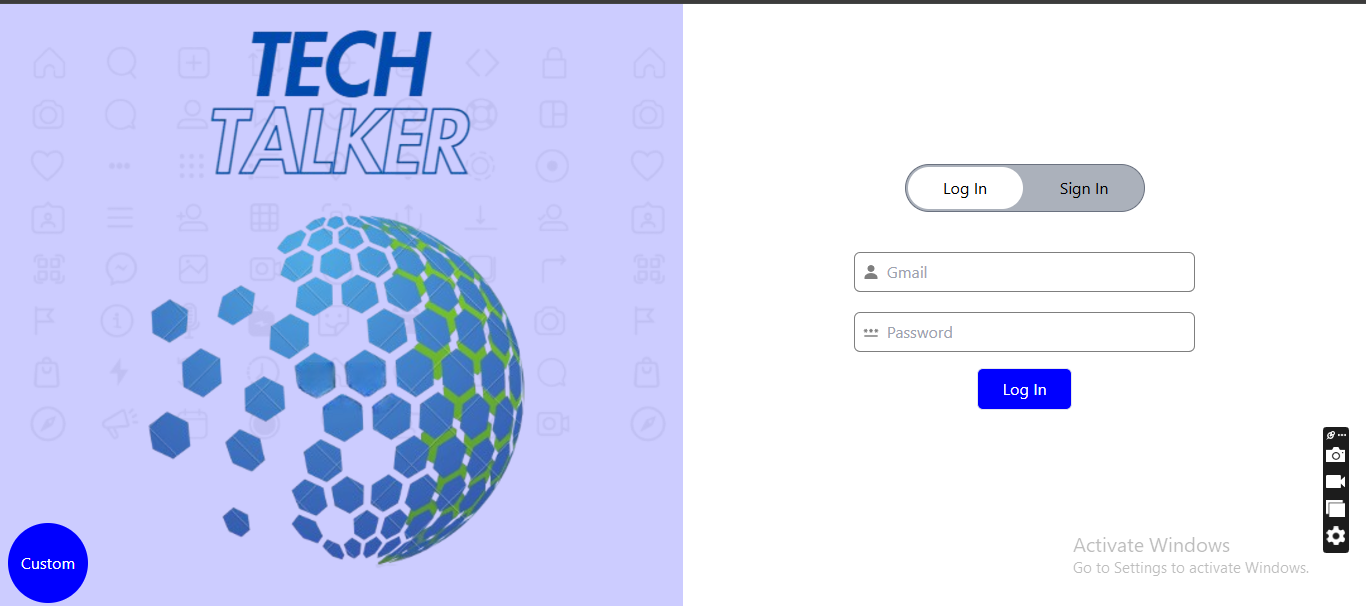
1. Day 1 (25-11-2024) Discussion on the problem.
2. Day 2 (26-11-2024) Basic UI Design with the Server Task related to APIs
3. Day 2 (night) Completion of basic UI design and web socket with open Ai discovering and implantation.
4. Day 3 (27-11-2024) WebSocket debugging, UI design fix, Documentation and PPT Designing.
5. Day 4(night) Final Finishing with UI And testing with the final documentation preparation.

Feature:

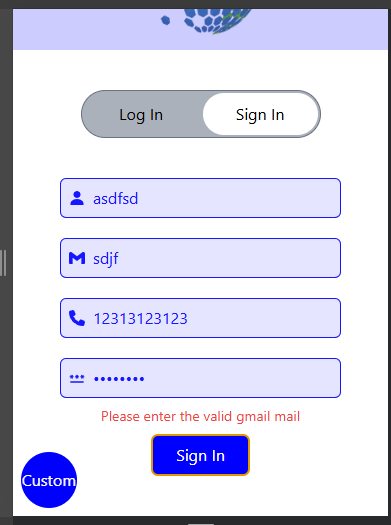
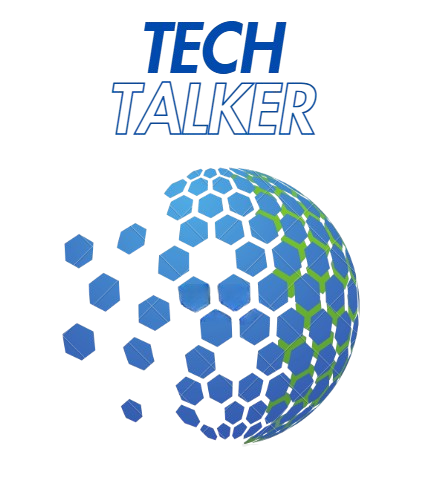
1. Not a single code and design is copied from any source (if you exclude the syntax of web-socket and some small other syntax)
2. Not a single code and design is copied from any source (if you exclude the syntax of web-socket and some small other syntaxes)
3. Custom Coloring options for the user, use change and use as he likes to look it like
4. Team work, every member is given an equal right to raise any objection.
5. Learning with creating, we learn from each other on how to perform these tasks in better and efficient way. While working in the project

Features of Our Program:

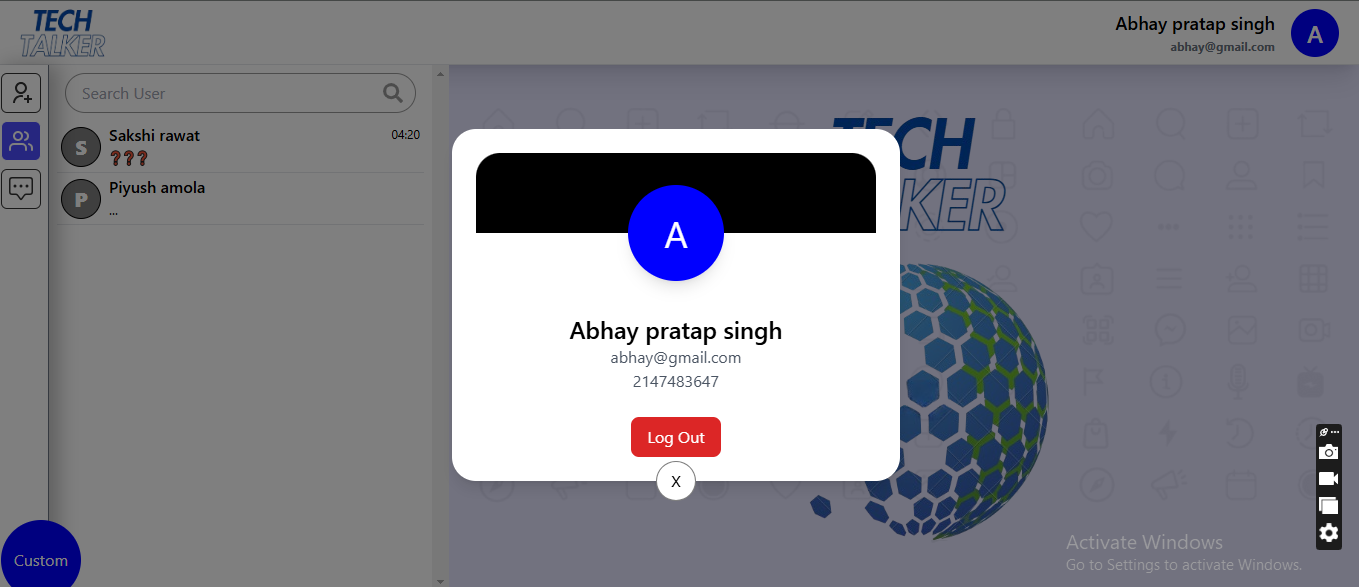
* Responsive in Nature
* Log In and Sign In Page.



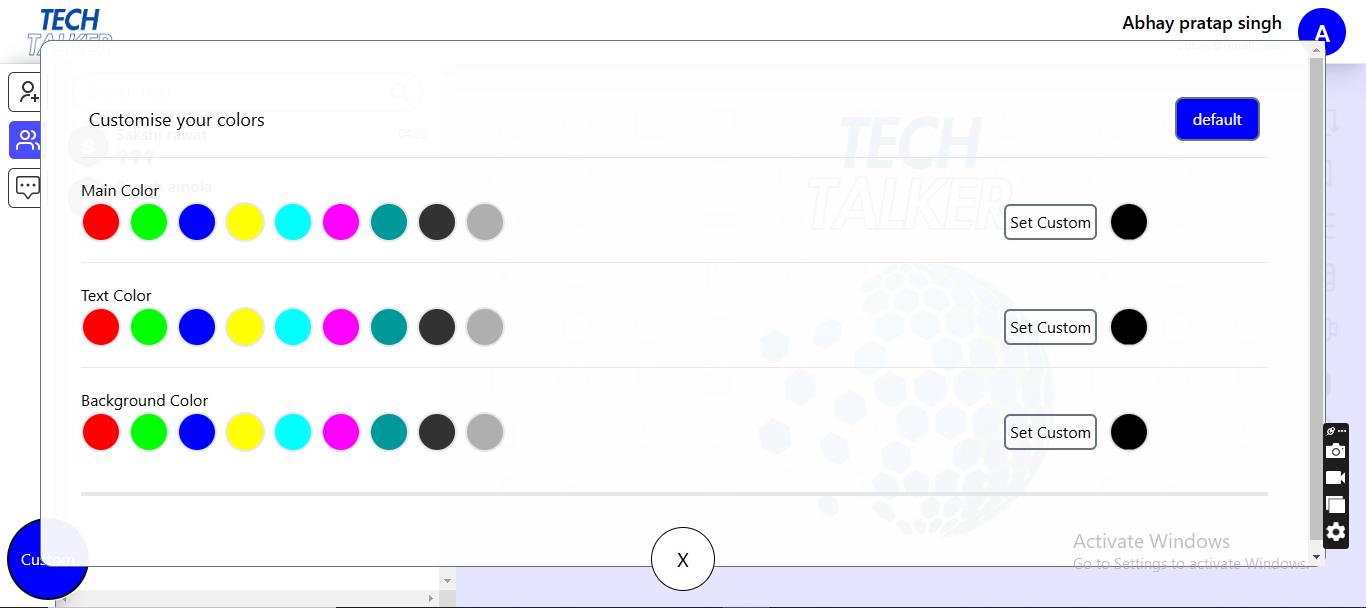
* Invalid email, phone no, and small and empty password



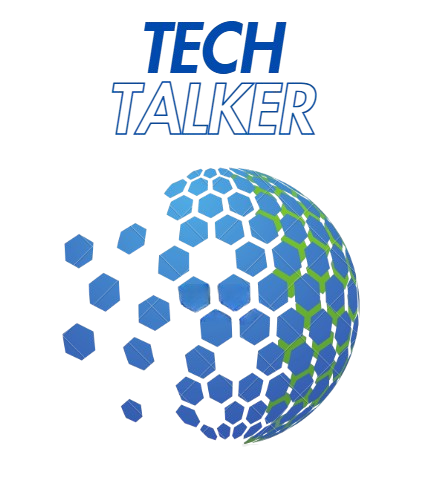
* User Profile Page For signing out



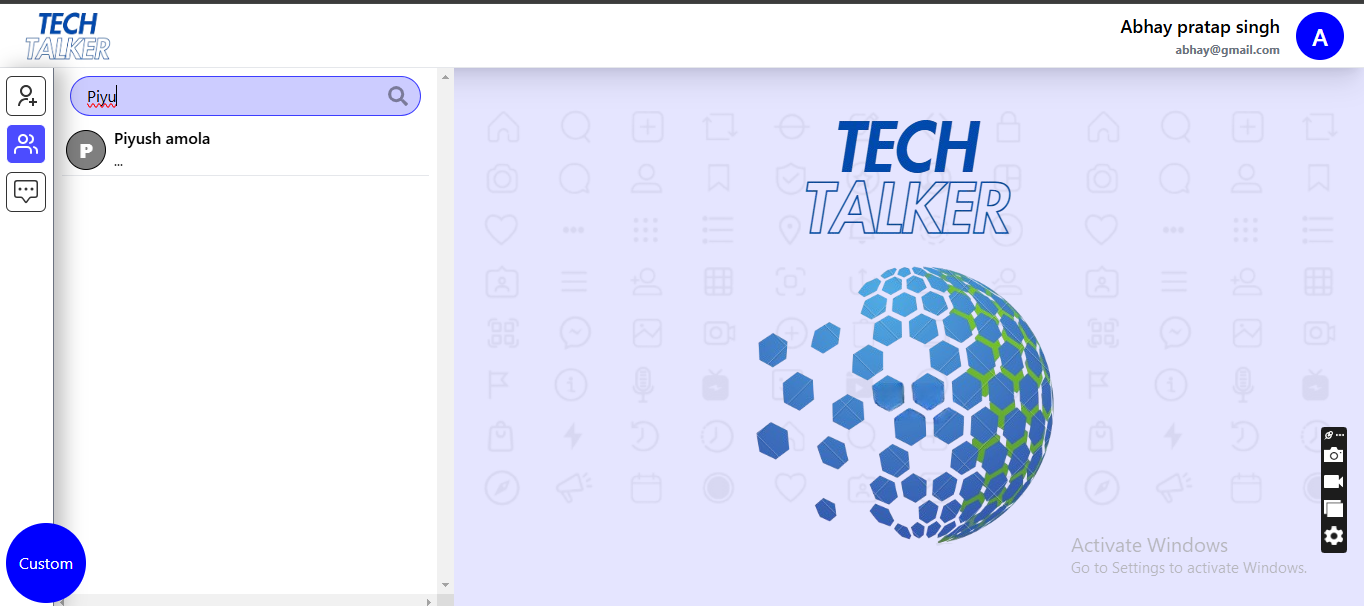
* Customisation Page for customising colour



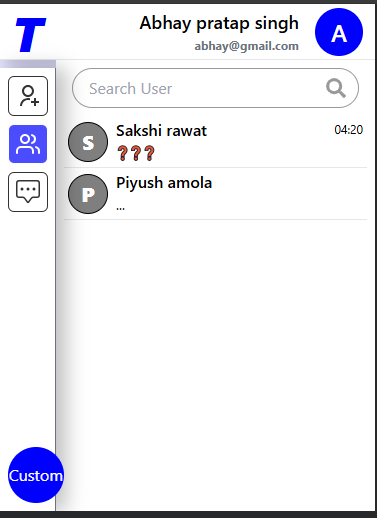
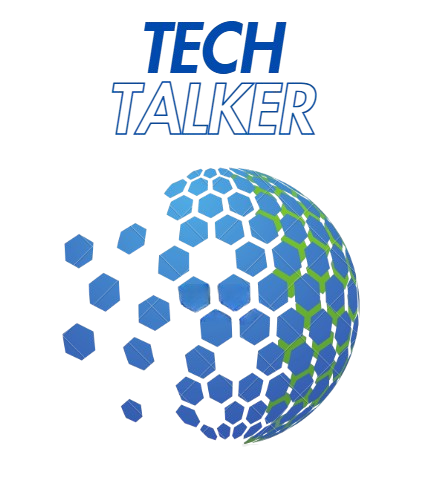
* Add new user as friend Page to add new user using their id (Gmail)



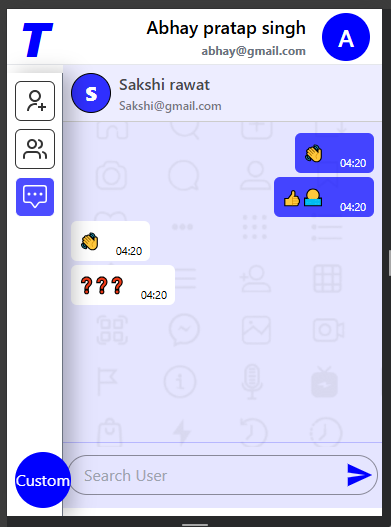
* Search Bar in the user’s detail column which is able to select by clicking input function



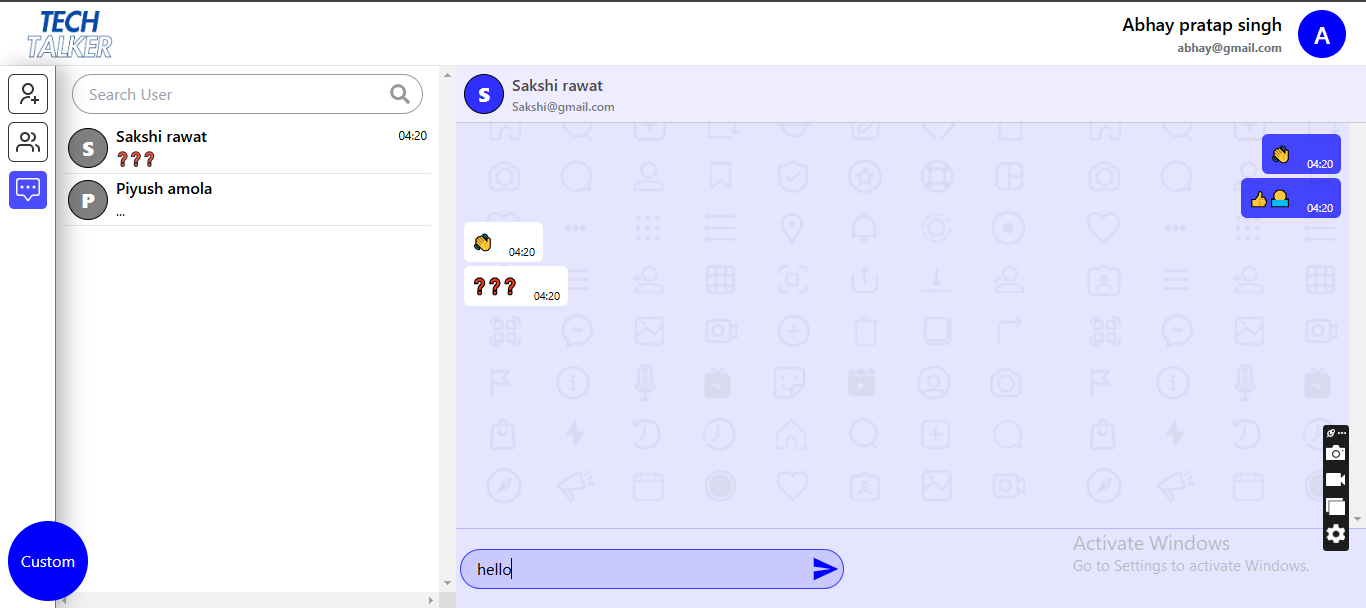
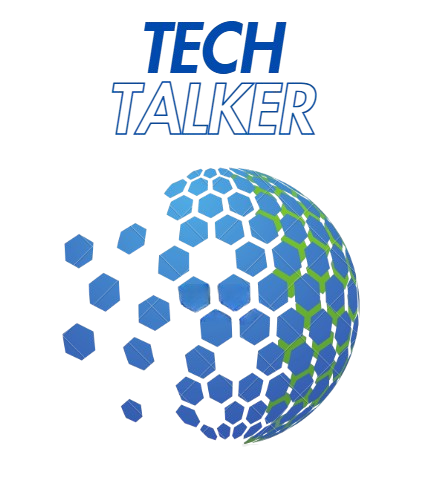
* All use Card in users with their last message, time and name of user



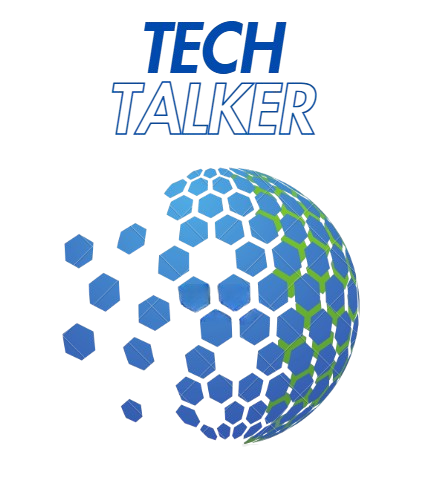
* When clicking user chats are open of user



* Chats that are send place in right and received are place left
* Input box is given in the centre bottom of the page responsible for the sending new messages



* Message are auto converted to emojis and text whose entire emoji is no found statically then ‘???’ symbol will come in place of it



***WE ARE NOT PERFECT***

***BUT***

***ONE DAY WE WILL!***