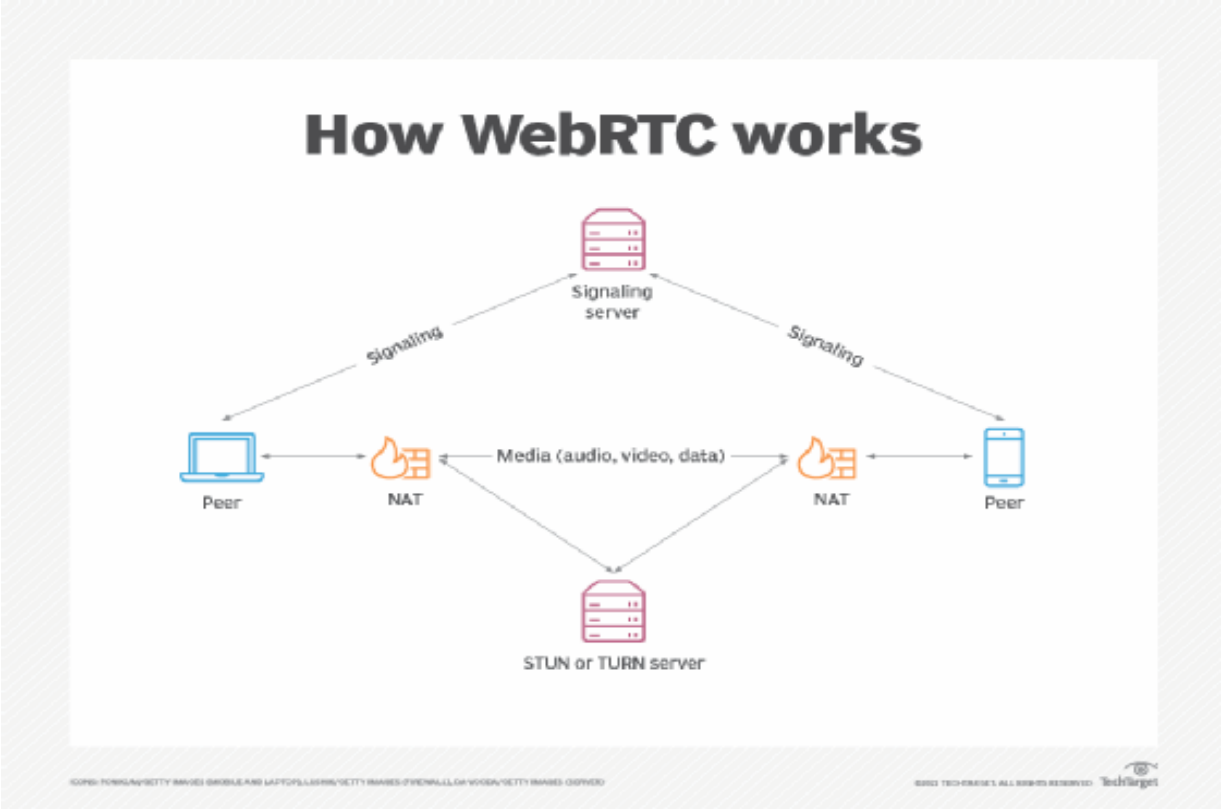
Collaborative Text Editor using webRTC and CRDT

Team Members:

- K Prudhvi - 2020201010
- CH N V B Dattatreya - 2020201011
- BVS Revanth - 2020201090
- T Yocahn Sandesh - 2020202001

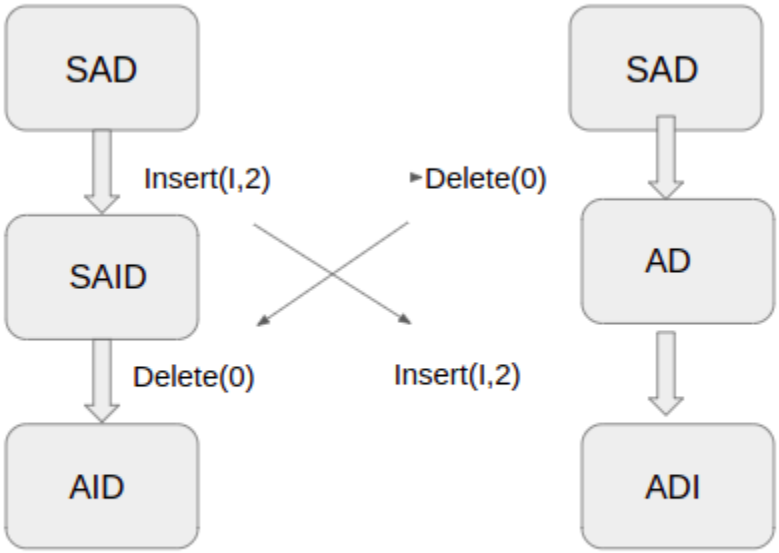
Introduction:

A collaborative real-time editor is a type of collaborative software or web application which enables real-time collaborative editing and simultaneous editing.Many users come together and apply changes to a document which should be reflected to other editors of the document as soon as possible.By using a centralized server we may not be able to meet the latency requirements.We use WebRTC to achieve real time communication between the users that come together to edit a document.WebRTC uses a lightweight Signaling server which allows users to connect to each other.



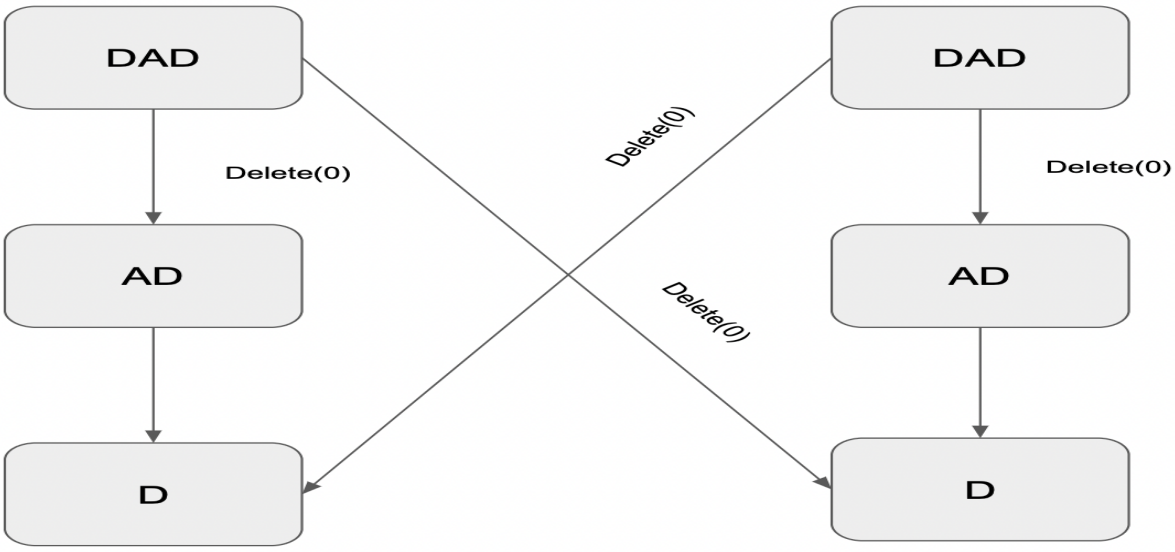
A basic Text editor can be seen as a sequence of characters having unique positions locally. But however we may get different conflicts when two or more users start collaborating.

Conflicts: There are some Conflicts that arise while building a collaborative text editor.



Initially,User1 and User2 have the same valueSAD in their text editor. User 1 tries to insert letter I at position2. After inserting, SAD became SAID.At the same time User2 tries to delete the letter at 0.After deleting SAD became AD. Now User1 receives delete command and user2 receives insert command. After respective operations, user1 has AID as final value and user2 has ADI as final value. Even though both users received the same commands, their documents did not converge to the same state. The reason for the different values is, both operations were applied in different orders i.e both operations did not commute.

Conflicts due to Deleting at same position:



Initially both users have DAD as text value in their local document. Both users wanted to delete the character at position 0.After applying delete command locally, both users received another delete command from each other which led to deletion of one more character.In this case, documents did converge but both users did not end up with desired. The reason for this result is that the same operation is applied twice.The operations are not idempotent.

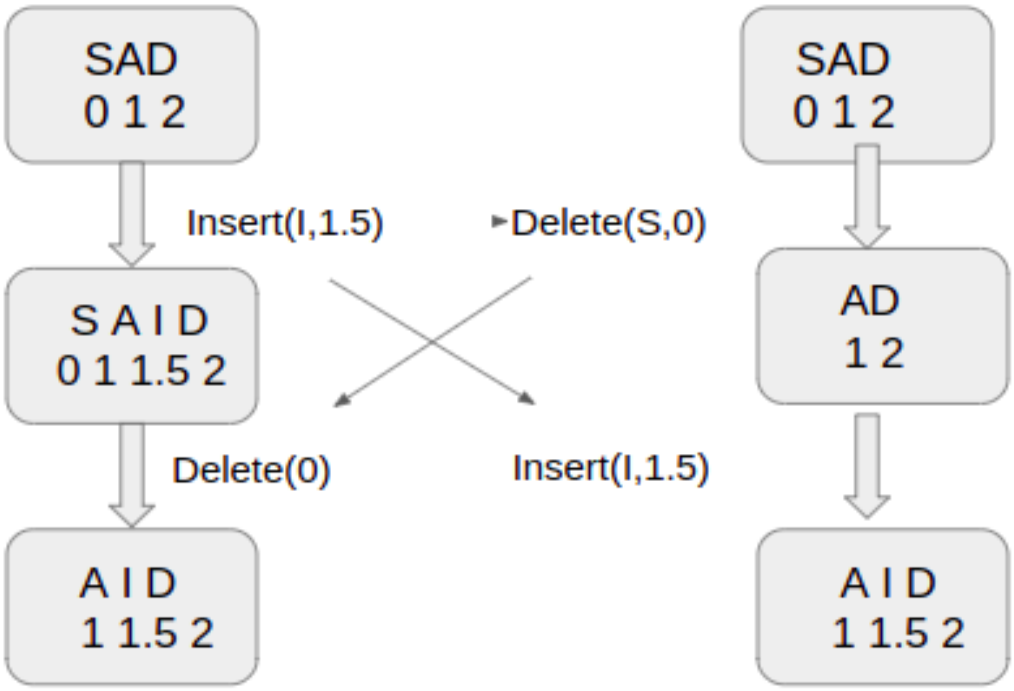
Conflicts due to Inserting at same position:

Initially both users have UT as text value in their local document. Both users wanted to insert a character at position 0(user1 -p,user2-c). Due to execution of both commands in different orders, Both users ended up with different values. In this also, documents did not converge.

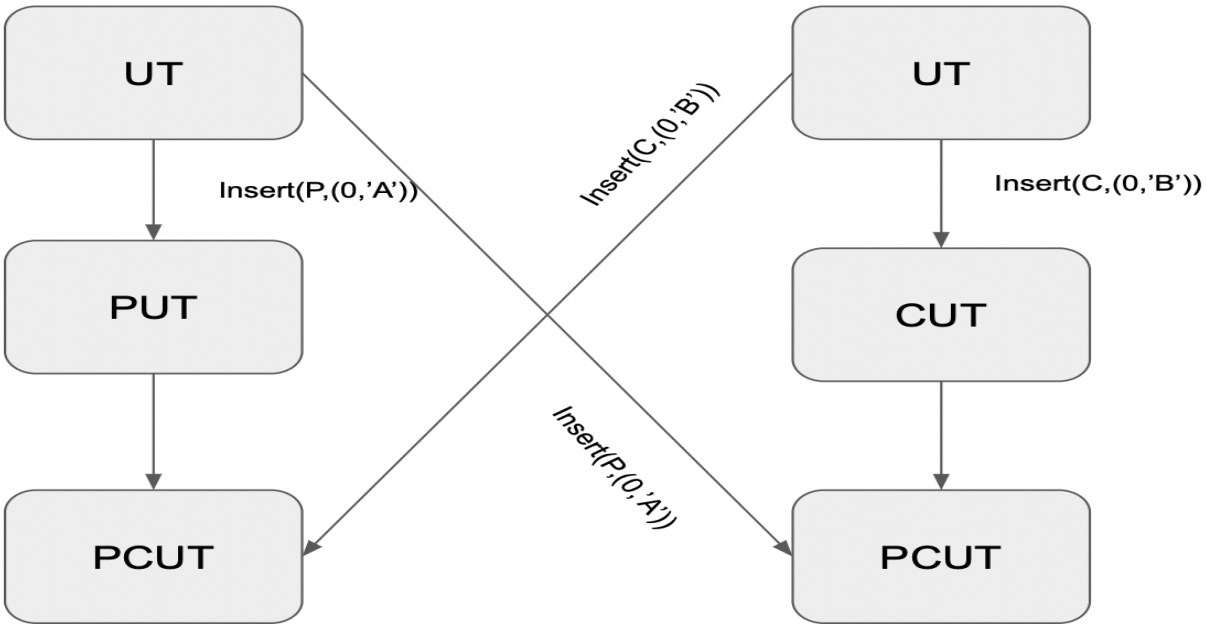
CRDT: Conflict Free Replicated Data Type.

In distributed computing, a conflict-free replicated data type (CRDT) is a data structure which can be replicated across multiple computers in a network, where the replicas can be updated independently and concurrently without coordination between the replicas, and where it is always mathematically possible to resolve inconsistencies that might come up.
In CRDT, Every character is globally unique. Each character has the same position identifier across all documents and all position identifiers are in sequential order corresponding to text. Position Identifiers h digit and site id as its members.

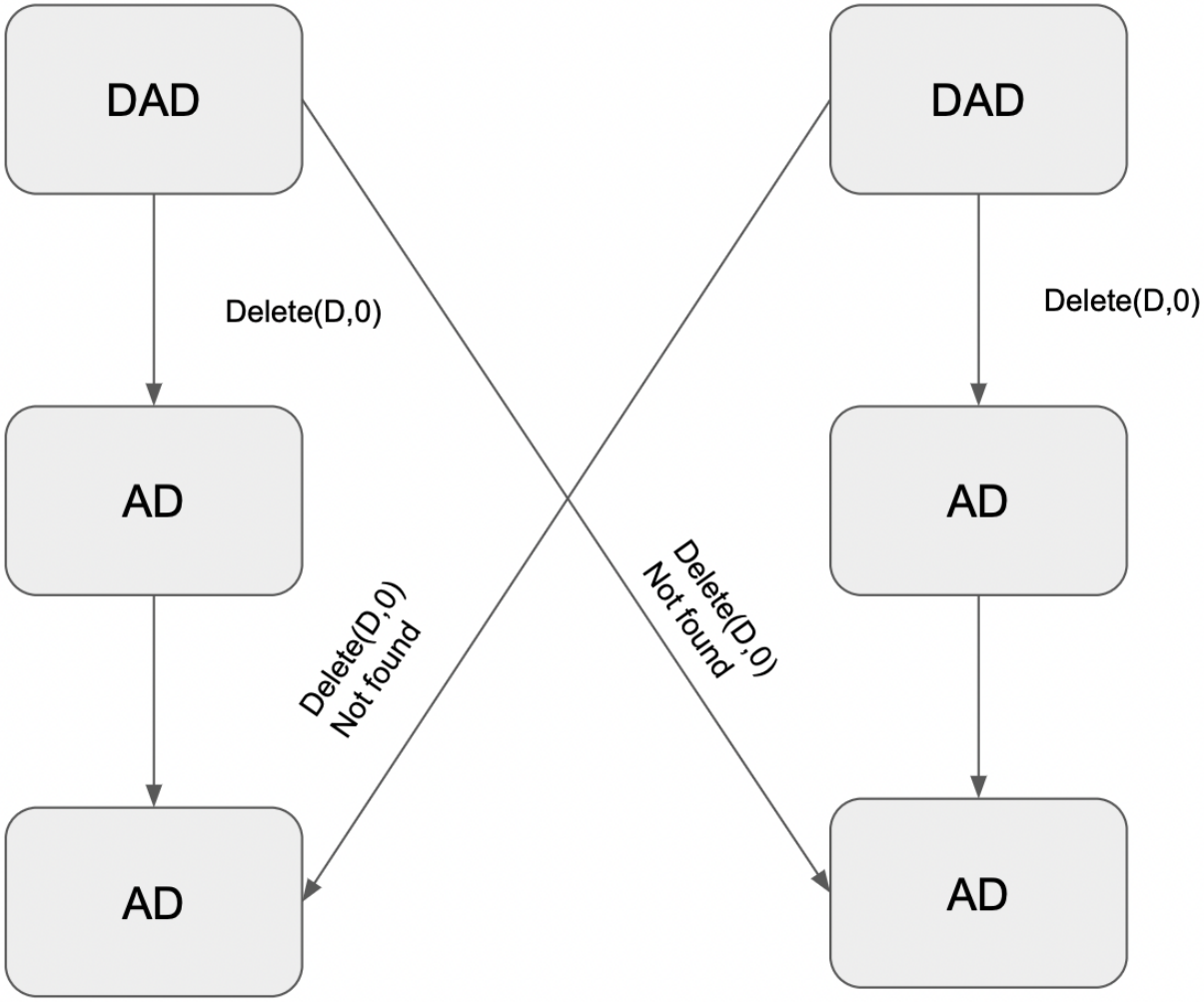
CRDT Solution for insert and delete at same time:



CRDT solution for inserting at same position:



CRDT solution for deletion at same position:



Algorithms:

The following section contains the major steps involved in four important operations that take place in a collaborative text editor

Local_insert(value,pos):

1. p1<-position identifier of previous character
2. p2<-position identifier of next character
3. p3<-Generate a new position identifier between p1 and p2
4. insert(value,p3) at crdt[pos]
5. broadcast('insert',value,p3) to other peers

Remote_insert(value,position_identifier):

1. v<-Find the correct position to insert the value by applying binary search on local crdt
2. Insert (value,position_identifier) at crdt[v]
3. Update local editor with the content of v
4. Broadcast to other peers

Local_delete(pos):

1. Get char at crdt[pos]
2. Delete the char and update the local crdt accordingly
3. Send (char,position_identifier) to other peers

Remote_delete(char,position_identifier)

1. Search for position_identifier in the local crdt using binary search
2. If the position_identifier is found delete that char from local crdt
3. Update local crdt
4. Broadcast to other peers

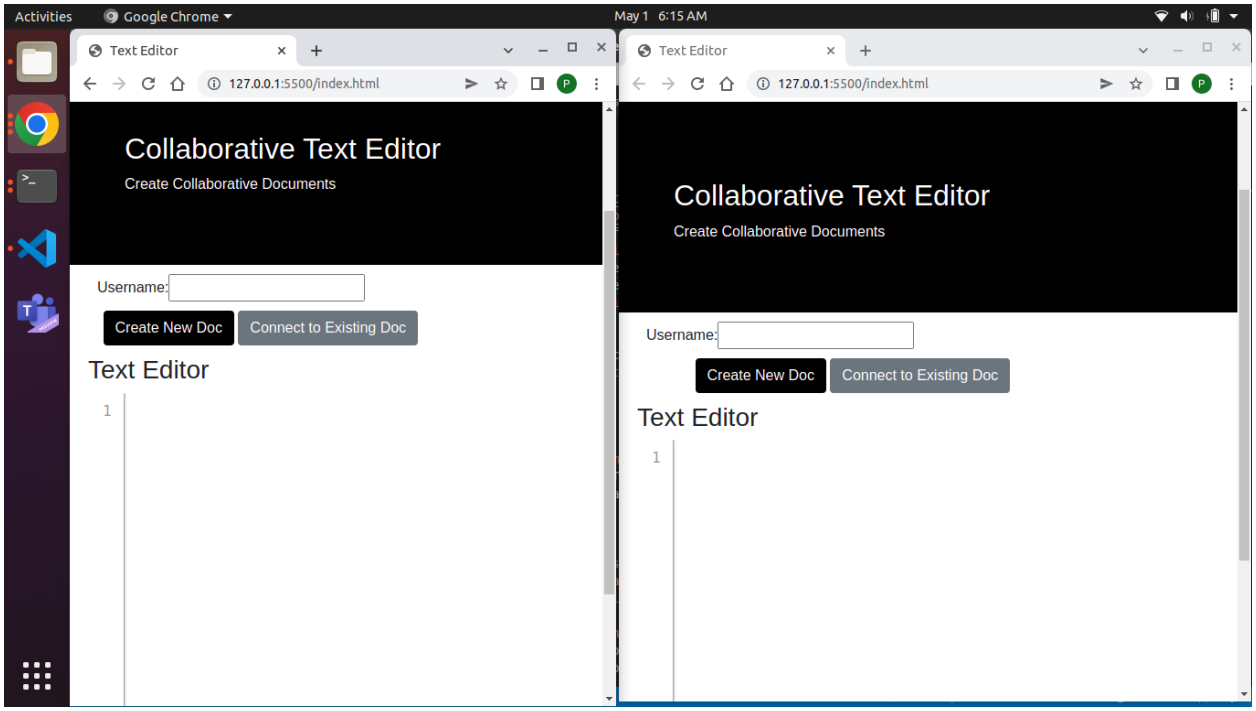
Libraries Used:

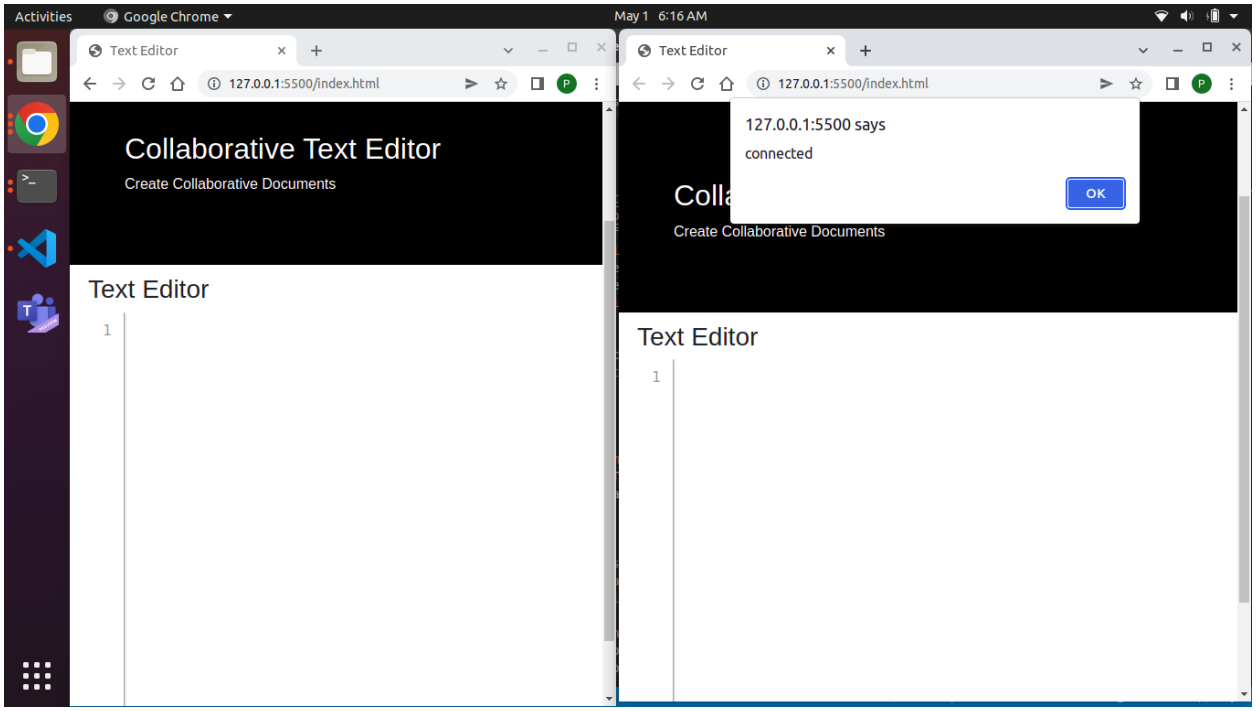
1. PeerJs which provides a wrapper for webRTC.
2. Codemirror which provides a custom text editor.

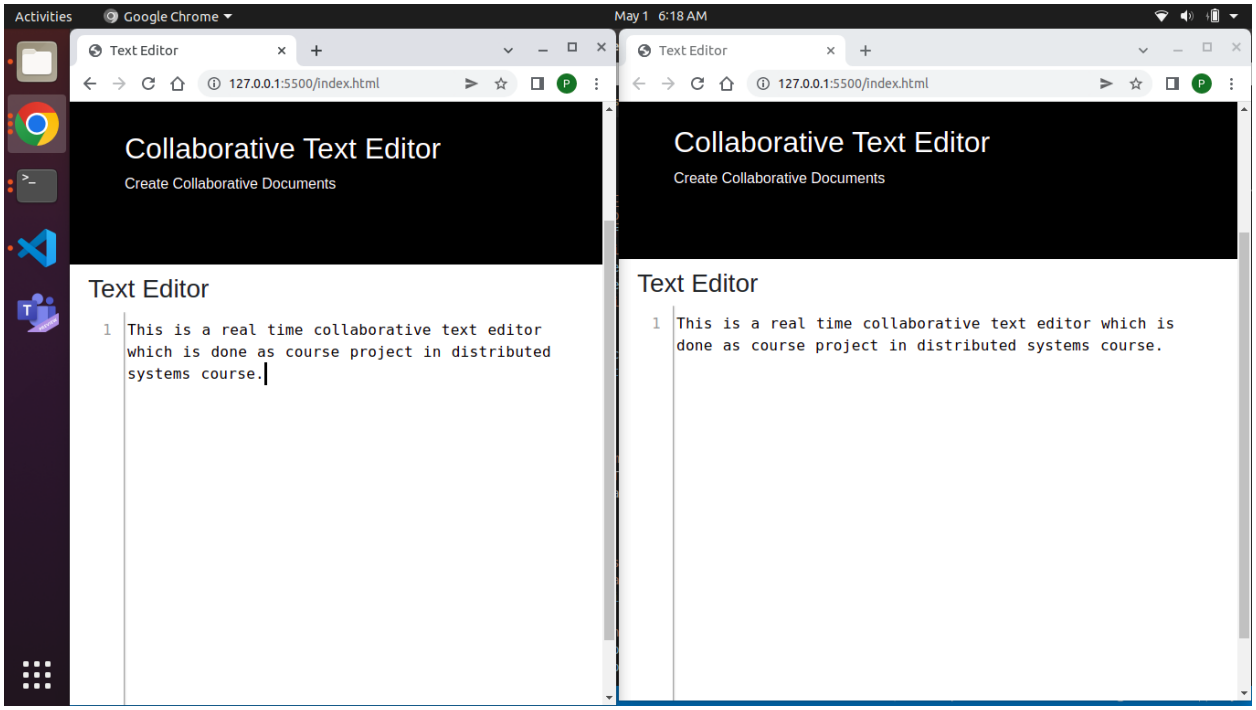
Outputs:

A User can either create a new document or edit existing document which is currently being edited by other users.

A user has to provide the other user id in order to start collaborative editing with other users







References:

1. https://conclave-team.github.io/conclave-site/#conflict-free-replicated-data-type-crdt
2. Logoot http://pagesperso.lina.univ-nantes.fr/~molli-p/pmwiki/uploads/Main/weiss09.pdf
3. LSEQ