I guess I can go with an approach like this

Make a callback function called setObserver(node):

The function takes an argument of the dom element and the function just does this

Observer.current.observe(node);

And since the function of the intersection will unobserve itself once its been activated once so it should work

Why is the usecallback function working in my situation cuz its designed to store a function and that function doesn’tt get remade every time the component re-renders and I am using a callback ref for the dom element

Do I need to use usecallback every time on other functions?

I don’t really know so lets find out

First let's see the code for the initial posts

I think why I am getting multiple posts is cuz the function is being recreated every time the component is being rerun that’s what been the conclusion so far by reading stuff but I am not really understanding what it means for the funcitoon to be recreated. But lets say if I made the additional posts be a usecallback function then I can

What I am thinking of doing now is that I want to make it so that once a friend is added you can start seeing his/her posts too for that I am thinking of adding a search bar on left top of the screen from where you can search username of a person and add them as a friend. I haven’t thought of adding a accept and reject system but will add it too later not now tho. So once the friend is added you can see their posts too in your feed.

How can I go about implementing this?

1.Create a search component

2. inside there will be a form with input and submit

3. add event listener to the form and on submit send friend request to server

4. inside the server the basic add friend syntax is already ready now we just need to check whether the given data is valid data or not

1. to check if the data is valid or not I will have to access the db and check whether the id sent by client is an actual user id or not

5 if validated will add friend that s simple

1. to add friend in friend list since an socket event is emitted I can just follow normal procedure

6. now once the friends are added if the post is empty then I will have request for intial posts from the client and everything will run smooth

7. if the posts aren’t empty then just continue once about 10 posts are passed the posts from the user will start to come

I don’t know what to do next.

Now I will make it so that once you comment you don’t need to reload it will just be added in the posts

I think I will implement this just like how facebook is doing it right now

I will select 2 comments in initial getting of posts and also while getting more posts then

When you click comments tab there will be a popop of commentsection where you can see all the comments

For that I need to create a component called commentsection

I already have a component named comment section I just need to so I need to make another component to show the first 2 comments below comments and once the button is pressed there will be a full viewport element that will be just comments

Made comments realtime so now I just need to make the posts be realtime as well for that I think I can do similar thing with comments I will pass the original posts from app and all the posts will be stored there but once a new post arrives I don’t need to do anything because that is already taken care of now I need to just handle the new post made by the user it will come on top and if they made new posts but if others post new stuff it will come automatically when the user scrolls and in ascending order

I have handled all post stuff in postContainer so I think I will emit an event from server once a post is added and will make it so that a new post is added the the viewport once the event from server is emitted

Fuck the posts aren’t coming to the client as expected and I don’t realy know why

I think the problem is that equation I made to get posts out of a certain length of time and I think that’s not working

Test dates

2024-06-27T06:27:19.027Z 2024-06-26T20:15:27.965Z

2024-06-27T06:27:19.027Z 2024-06-26T20:15:27.965Z

2024-06-27T09:13:54.305Z 2024-06-27T09:13:54.305Z

2024-06-27T09:13:54.305Z 2024-06-26T20:15:27.965Z

What needs to happen is that at first in initial request the dates are set at to be date of first and last posts. Now once another posts arrive I need to set dates to be according to them

New first date is the date of first post of additional posts

Old first date is the first date from before

New laast date is the date of laast post of additional posts

Old last date is the last date from before

What I want:

I want the posts to come as I keep scrolling and new posts to keep coming as they are posted but the ones that are already shown they should not be shown to the client side so for that I think I have implemented an infinite scrolling system where as the client keeps scrolling the next posts keep coming now the case in which new posts are posted be friends is handled but only if the nymber of new posts are less than 10 posts but once the number increases to more than 10 stuff starts to break because in my logic right now additional posts are quired based on the first and last post time where first post time is the posted time of the most latest post the client side is sent and the last post time is the date of the earliest post that was sent to the client now here is my current algorithm :

The server sends posts that have date>first post date or date<last date

First the client just receive the initial posts and first post date and last post date is set

For the sake of this example lets say first post date is 2024

And last post date is 2020

And I take 10 posts from the database at a time.

10 is the maximum amount I take for now I can change this to any number I want and there is no problem sending less than 10 posts

At this point lets say the client keeps scrolling and no new posts are made the client has no friends and is very lonely so the client will query for posts that are older than last post date i.e 2020 and since no new posts were posted the latest post that client has received is from 2024 so the server sends 10 posts to client from 2015 to 2020. Now the first post date and last post date will change accordingly where last post being 2015 and first post not changing and still being 2024 because of the reason explained below.

So at this point of time first post date is 2024

Last post date is 2015

Now while scrolling through these 10 posts other 5 posts were made so now the data base will send other 10 posts including these five posts and in doing so the first post date will change and one of those 5 post will be the latest post sent to client and for the sake of this example lets say the first post has changed to 2026 and as for the last post the last post date will be the oldest post from those additional posts lets say 2013

First date 2026

Last date 2013

Now still while scrolling through these posts lets say other 15 new posts were made now the database will send the 10 posts and lets say the first date will be 2050 and the last date from this additional post is 2030 this isn’t the original last post date just the last post of the current additional post date now iff we just set the last date to be the last additional post date the posts that have already been requested will be requested again so to prevent that I need a solution

Now

If first additional post date is greater than first date then we will change first date to additional post date or else no change

If last additional post date is greater than first date then change last date to last additional post date or else if last additional post date is smaller than last date then change last date to last additional post date

If last additional post date is greater than first date then the last date will be changed but push the last date to a stack first then change it

How to put dates in stack:

For starters if last additional post date is greater than first post then we will push this last additional post date and first post date in the stack

Now in another query if again last additional post date is greater than first post date then push the last additional post date and first post date in the stack

In case in another query if last additional post is less than last post date of top of stack then pop that date and check if the last post date is less than other last dates in stack if It is then just pop them and add this last post date along side with first post date of just latest popped date

First date:2026

Last date:2013

First date and last date will be stored in variables where first date is the most latest post date and last date is the most earliest post date