Pagination Steps Followed

1. First we need to fetch the data using useEffect and we have to store that in one variable
2. After that map the each item by using that array of an object variable and display specific number of items per page
3. We have to plan how many number of items are there in array and how many should display per page
4. Suppose 100 items are there in array 10 item or 12 items per page or so on
5. If 10 items per page totalItems/items per page=>100/10= 10 pages will be there total
6. And display items using slice(0,10) or slice(0,12) like that we should plan and according to that we should use logic

Slice(0,10)

Slice(10,20)

Slice(20,30)

Slice(30,40)

Try logic for this kind of displaying

(10\*1 -10 , 10\*1)

( 10\*2 -10 , 10\*2)

( 10\*3 -10 , 10\*3)

( 10\*4 -10 , 10\*4)

Her 1,2,3… will be the page number we are selecting ,so create state variable for that

Const[page,setPage]=useState(1)

<div className="grid-container" >

  {

    product.slice((page\*10-10),page\*10).map((prod)=>{

      return<div className="single-container">

           <img src={prod.thumbnail}/>

           <span>{prod.title}</span>

      </div>

    })

  }

  </div>

This for displaying the selected data

Now create Pagination Layout

Now we need to create one new array for index number with length of 10 items storage empty array

Now according to index number display 10 buttons with content of (i+1) because index start with ‘0’ and page number starts with ‘1’

And create forward and backward buttons also

<div>

      {

        product?.length>0 && <div className="pagination-container">

          <div className={(page<=1)?"disableBtn":"" } onClick={handlePagebackword}>⏪</div>

           {

            ([...Array(Math.ceil(product.length)/10)]

            .map((\_,i)=> <span

            key={i}

            className={page===i+1 ? "pageActive" : ""}

             onClick={()=>handlePage(i+1)}>{i+1}</span>))

           }

          <div className={(page>=10)?"disableBtn":"" }

          onClick={handlePageForword}>⏩</div>

        </div>

      }

    </div>

Below is the event handler of each page button

 const handlePage=(selectedPage)=>{

    if(

      selectedPage >0 && // always check possible conditions

before performing actions

  selectedPage <=10 &&

      selectedPage !==page

    ){

      setPage(selectedPage)

    }

  }

One more requirement is if the page is more than page.length and less than page.length yous should disable or hide the forward and backward button

For that use conditional class and hide the button using opacity:0

<div className={(page<=1)?"disableBtn":"" } onClick={handlePagebackword}>⏪</div>

.disableBtn{

  opacity: 0;

}