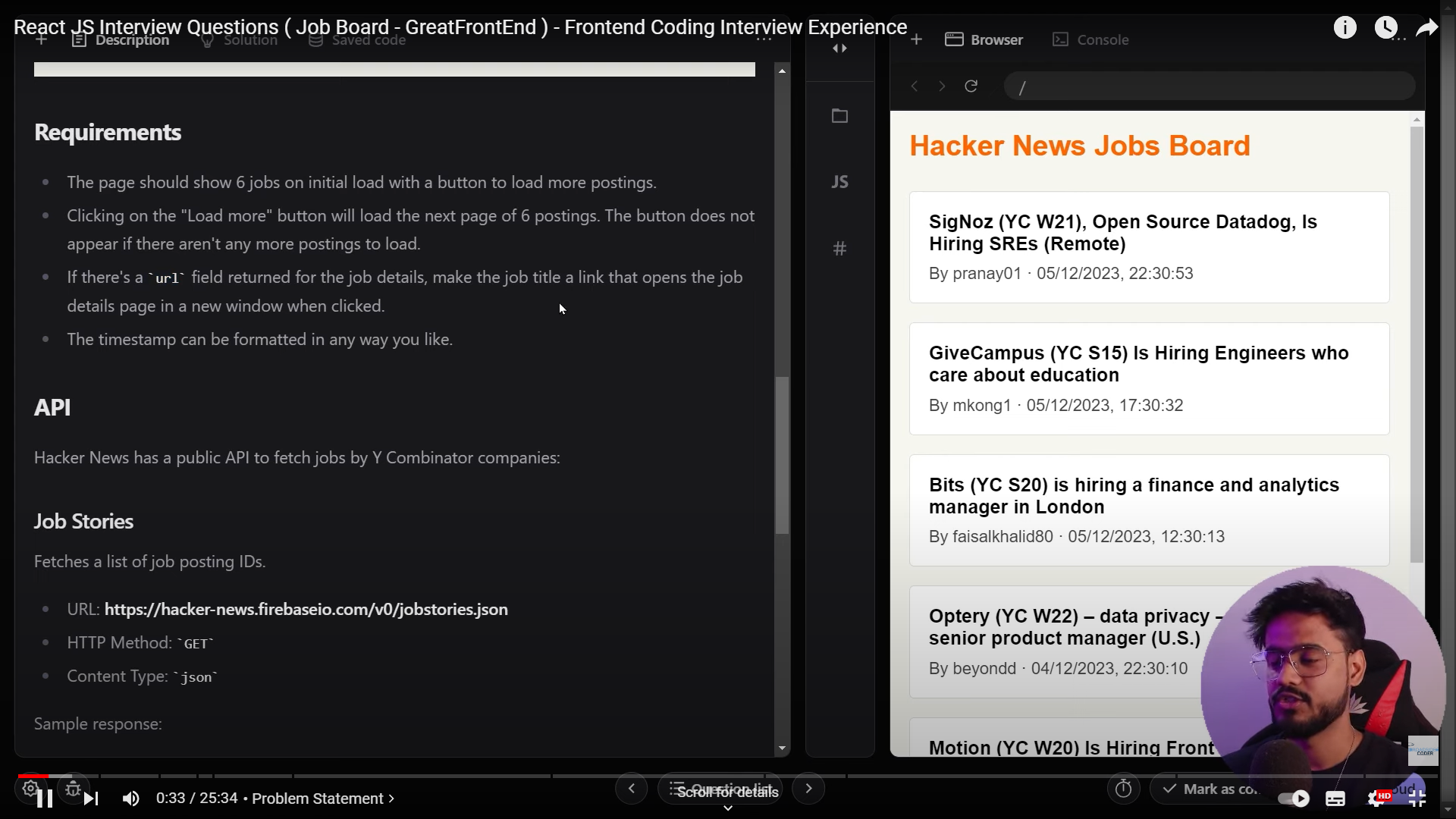
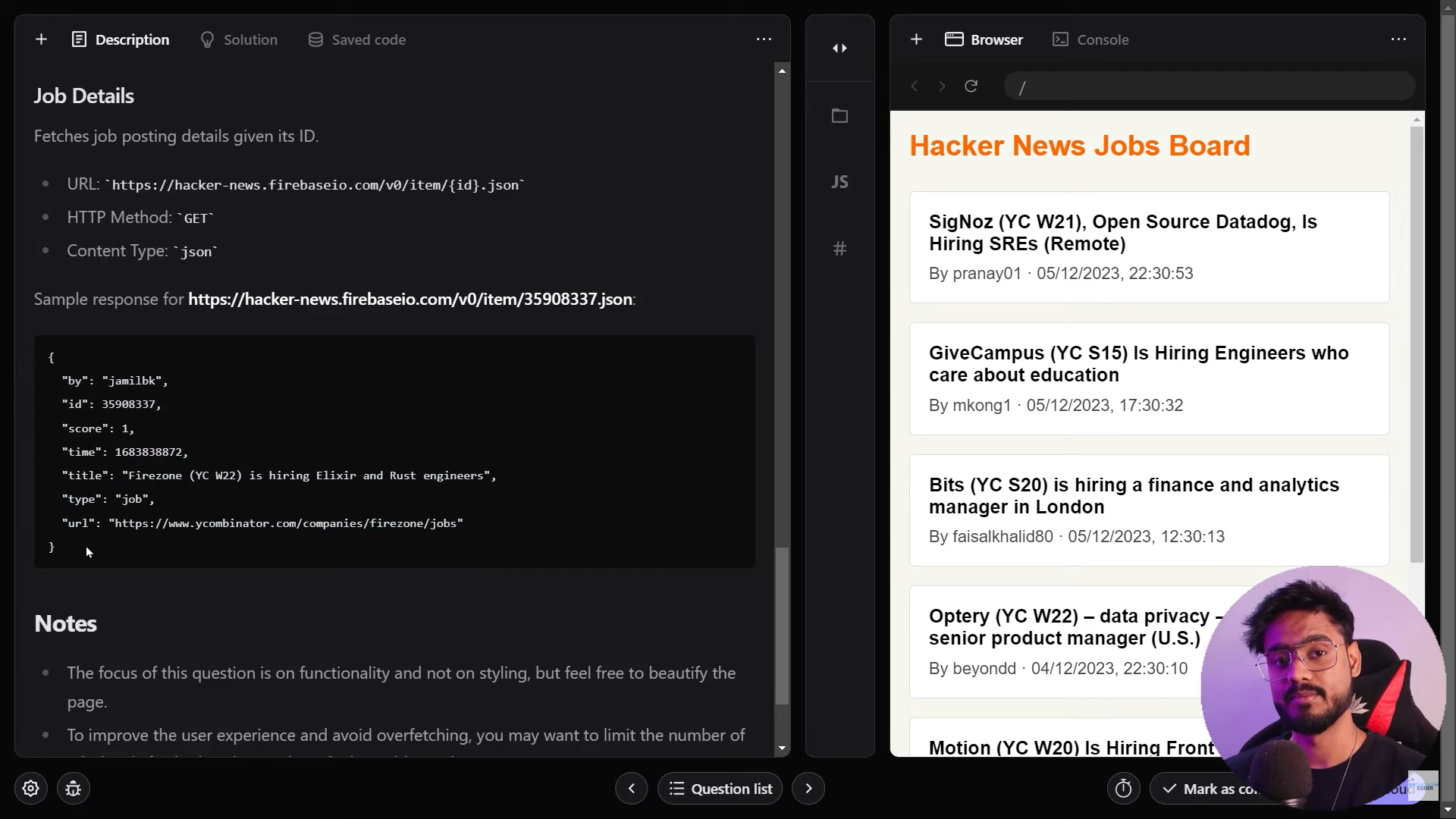
Job dashboard steps implemented:





1. Requirement is that in 1st page when you load the application by default it should show 6 jobs, when you click on load more button it should show another 6 pages, It is pagination concept
2. In requirement they have mentioned is 1st you need to fetch the job ids and using that job ids you have to load the 6 jobs each times when you have clicked on load button
3. And design the anchor tag if url not present don’t show pointer as cursor else show cursor
4. When the page 1st loads that time page number should be 0 by default and fetch api and when you are loading next 6 jobs update pages by adding 1

1st you need to create the state variable for fetching ids, only once need to fetch the data whenever that state variable is null, and slice that ,

After slicing the data you have to map each id, and again you have to fetch that single job details and store that in one normal variable and create second variable to store that fetched data

2nd state variable for storing the array of each job object after fetching those

3rd state variable for page number, default page number will be 0 and later you have to set the page number, when we click on load button again and again by adding one

4th variable for fetching status that is Boolean type

const fetchAPI = async (page) => {

    setCurPage(page); // this is important to set the page

after receiving

    setIsFetching(true);

    let idObj = jobIds;

    //let idObj;     don't use like this because    /\*\*\*\*\*\*\*\*\*very imp point  \*\*\*\*\*\*\*\*\*\*\*\*/

    //  only once we will fetching 1st  api when 1st time jobIds are null, if you keep idObj without assigning anything 1st time when we fetch the data it works, but next step when we are performing slice in next step when we again press the button it tells idObj is undefined bcz that will be null , and that was updated inside block scope if you make state variable of jobIds is equal to idObj in next rendering it takes already fetched data so no need to fetch again and again too

  if (jobIds === null) {

      const idJson = await fetch(`${api\_endpoints}/jobstories.json`);

      const Obj = await idJson.json();

      console.log(Obj);

      idObj = Obj;

      setJobIds(Obj);

    }

    const jobObjPerPage = idObj.slice(

      page \* items\_per\_page,

      page \* items\_per\_page + items\_per\_page

    );

    const singleJobObj = await Promise.all(

      jobObjPerPage.map((oneJobId) =>

        fetch(`${api\_endpoints}/item/${oneJobId}.json`).then((res) =>

          res.json()

        )

      )

    );

    console.log(singleJobObj);

    setJobArr([...JobArr, ...singleJobObj]);

    setIsFetching(false);

  };

  useEffect(() => {

    if (curPage === 0) fetchAPI(curPage);

  }, []);

Only once you need to fetch the ids when id state variable is null, after fetching that each time you have to fetch the default page and required page using Promise.all()