←

React-UseEffect-Basic-2

Ended

React

Easy

Max Marks - 15

DOWNLOAD ZIP ⬛

New Submission

Github URL *

SUBMIT 💾   Best Score -15

Latest Submission   ⌄

Description	Submissions

React useeffect 2

Submission Instructions [Please note]

Maximum Marks - 15 marks

- The Submission should not contain spaces, for example /rct-101 folder/eval will not work
- Do not push node_modules and package_lock.json to GitHub

```
✅ Able to submit the app - 1 mark ( minimum score )
✅ Check if the initial network request is made and get only 10 units of data - 2 marks
✅ check if the default page is 1 and button is disabled and page 1 results are correct - 2 marks
✅ check if page buttons are working fine and fetching the correct page data and updating the DOM - 3 marks
✅ The border color of the active page should be red - 2 marks
✅ check pagination component is dynamic - 5 marks;
```

Installation

- Use node version(LTS) should be `v16.16.0`
- Don't change/override package.json
- Download and unzip the boilerplate file and then copy the "**contents**" of the unzipped file in the Masai Folder.

```
// install npm packages
npm install --engine-strict

// start application locally
npm run start
```

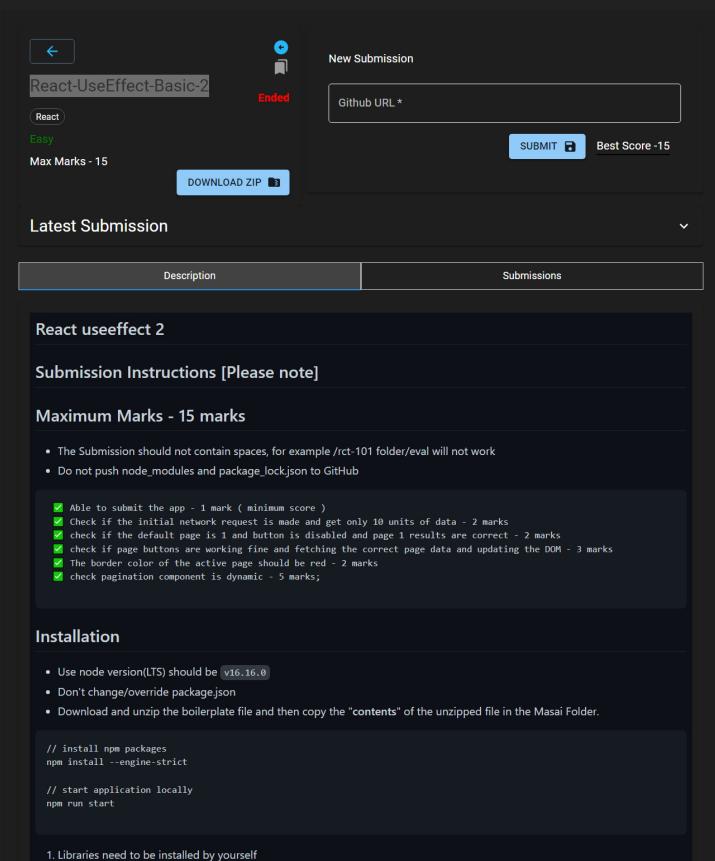1. Libraries need to be installed by yourself

Not following the above instructions will lead your test cases to fail

Description

- You need to make an application that lists employees list from an API.
- The user should be able to apply pagination.

Employees Dashboard

Name	Department	Image	Gender	Salary
Donald Surgen	engineering	https://i.imgur.com/9xEcfJj.jpg	female	57000
Brynn Wildblood	finance	https://i.imgur.com/9xEcfJj.jpg	female	27000
Etta McGairl	operations	https://i.imgur.com/9xEcfJj.jpg	male	42000
Alexandros Renihan	engineering	https://i.imgur.com/9xEcfJj.jpg	female	40000
Geri Maudlen	operations	https://i.imgur.com/9xEcfJj.jpg	others	40000
Grazia Pilkington	operations	https://i.imgur.com/9xEcfJj.jpg	others	50000
Norry Bodicum	operations	https://i.imgur.com/9xEcfJj.jpg	others	30000
Athena Broadey	marketing	https://i.imgur.com/9xEcfJj.jpg	male	66000
Luisa Artois	engineering	https://i.imgur.com/9xEcfJj.jpg	male	45000
Cookie Gimber	finance	https://i.imgur.com/9xEcfJj.jpg	male	25000

1 2 3 4 5 6 7 8 9 10

;

Boilerplate

- Components
  - Employeedashboard.jsx
    - user should be able to see a list of all employees fetched from API
    - there should be 10 results per page,
    - it should be in tabular format
    - On loading, the page make an API request with default page 1 and limit 10 end-point:- https://dbioz2ek0e.execute-api.ap-south-1.amazonaws.com/mockapi/get-employees.
    - query params:
      - page: a number representing the page number.
      - limit: a number representing the total number of results per page.
    - Check the response from api to know the data format you are getting.
    - Loop over the response data and create rows and append them inside the tbody given in the template(send them as props to required components and complete the functionality).
      - Each row will have the following columns:-
      - Name
      - Department
        - Image
      - Gender
      - Salary
      - Here all the data will come from the api request
    - only 10 results per page should be visible and there should be buttons to change the pages. The number of pages should be as per the no of results you are getting.
  - Employeecard.jsx
    - Component to display information about a single employee
    - The classnames mentioned in the boiler plate and just append the appropriate data.
  - pagination.jsx
    - component which will have 1,2,3,...n number of buttons
    - The current button will be disabled
    - This pagination will get the required props from other components.
      - it will accept the following properties

- It will accept the following properties
  - current - a number representing the current page
  - onChange - a callback which will be given the new page number (page)=>{})
  - total - a number representing the total pages present in the list(verify api response to get the total number of pages required)
  - Populate page numbers equal to the number of total pages received from the api response. (refer to the image)
  - The border color of the active button should be red(the border should be given as "1px solid red")
  - by default page number 1 should be the active page
  - on click of any button the new page number will be sent to the onChange callback.

Note - Make sure you use only the given components and don't create new files and folders. Changing the component name, and structures might result in giving you zero marks.

General guidelines

- The system on cp.masaischool.com may take between 1-20 minutes for responding,
- so we request you to read the problem carefully and debug it before itself
- we also request you not just submit it last minute
- try to keep one submission at a time