مکتب شریف

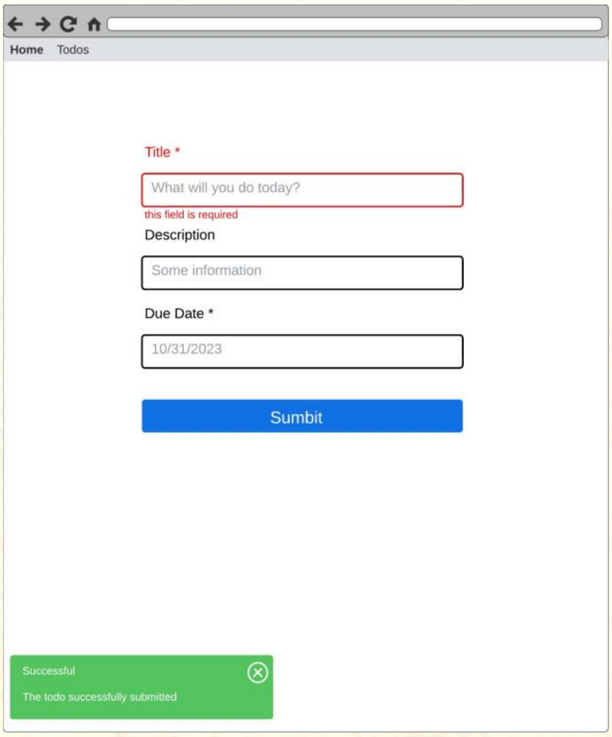
اولین بوتکمپ آموزشی - استخدامی ایران

React JS

Week 12 HW

بسم الله الرحمن الرحیم

1. Implement a to-do app using the wireframe below.



Title *

What will you do today?

this field is required

Description

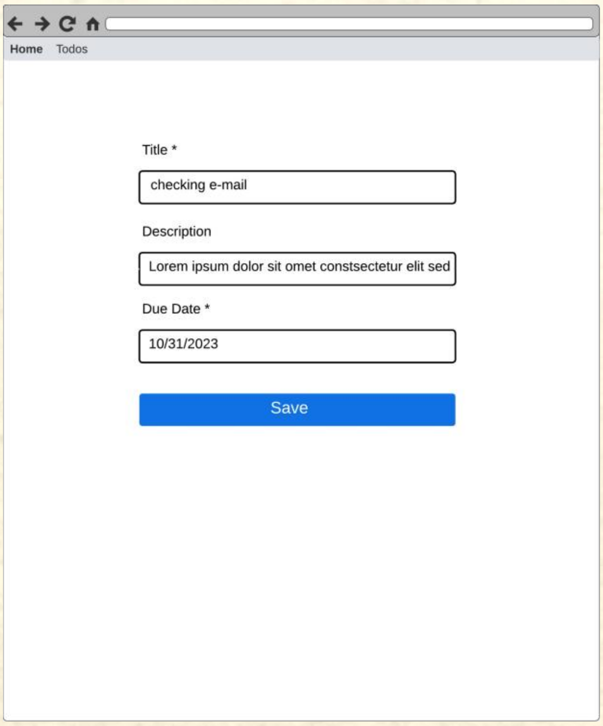
Some information

Due Date *

10/31/2023

Successful

The todo successfully submitted



Title *

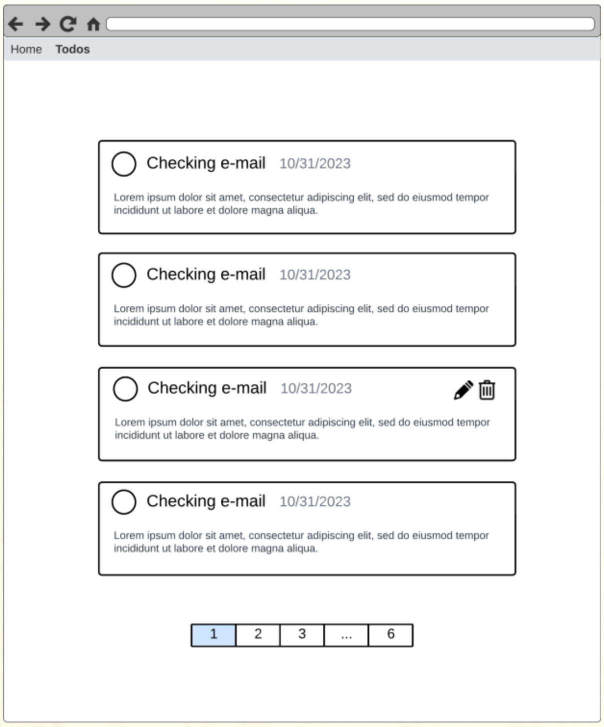
checking e-mail

Description

Lorem ipsum dolor sit omet constsectetur elit sed

Due Date *

10/31/2023



○ Checking e-mail 10/31/2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

○ Checking e-mail 10/31/2023

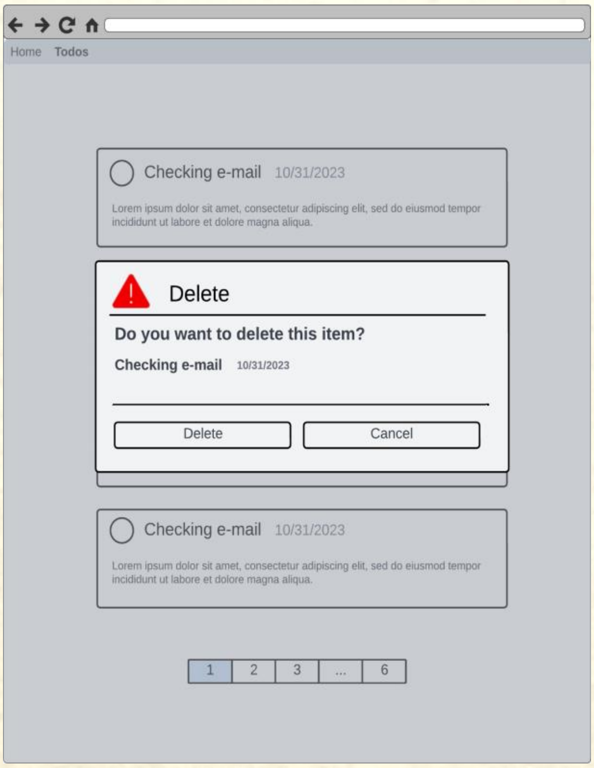
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

○ Checking e-mail 10/31/2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

○ Checking e-mail 10/31/2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



○ Checking e-mail 10/31/2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Delete

Do you want to delete this item?

Checking e-mail 10/31/2023

○ Checking e-mail 10/31/2023

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Notes:

1. You are free to style the app however you want using any tools you have learned.
2. The app must be fully responsive.
3. Add a new task form must be in Home page.
4. To-dos list must be in Todos page.
5. You can use mockapi, json-server or any fake REST API you want.
6. Todos page must have pagination (10 tasks on each page)

    - ✓ Pagination must be implemented in the client side. (You get all todos from the server and then paginate them with your code.)
    - ✓ Use queryParameters for pagination. For example : http://localhost:5500/?page=1

7. To add a new task, you must send the properties to the server.

    - ✓ id, title, description, dueDate, createdAt, updatedAt, isDone
    - ✓ use POST method

8. Values such as id, createdAt, updatedAt and isDone must not be taken from the user and must be automatically filled through your code.

    - ✓ createdAt is the time of task creation.
    - ✓ updatedAt is the time of editing a task.
    - ✓ At the time of creation, both of these values are the same.
    - ✓ It's better to send dates as timestamps to the server but it's not necessary and you can send them in any format you wish, even as a string.

9. After successfully submitting(adding) a new task, a proper toast must be shown and the form must reset.
10. By clicking on edit button, the page is redirected to home page, and the id of the selected task is set as a queryParameter(http://localhost:5500/?id=1).

- ✓ After the page is loaded, you must get the id from the URL.
- ✓ If the id queryParameter exists, details of selected task must be received from the server and used to fill the form.
- ✓ The Submit button must also change to Save.
- ✓ After successfully editing the task:

  - a proper toast must be shown.
  - updatedAt value must be changed as well.
  - the form must reset.
  - the Save button changes back to Submit button.
  - queryParameter is removed from the URL

- ✓ if server response status code is 404 (not found), the user must be redirected to a Not Found page. (Style this page however you like)
- ✓ if the id queryParameter doesn't exist, add a new task form is shown.

11. A header similar to the wireframe is necessary in all pages and navigation through pages is done using the header links.
12. By clicking on the delete button, a modal similar to the wireframe is shown, with some details of the selected todo.

- ✓ If confirmed, use DELETE method to delete the selected task from the server.
- ✓ After receiving the response from the server, the todos list must be updated.
- ✓ Page must not refresh. For example, if you're on page 3, you must stay on page 3 after deletion.
- ✓ If a page doesn't exist, redirect to not Found page.

SUBMISSION NOTES

1. Compress your homework into a single compressed file (.rar or .zip)
2. Create a repository for the homework in GitHub and give access to your TAs.
3. The name should have the following format: name_hw1_maktab99
   example: MohammadAli_Kargar_hw1_maktab85
4. Create a description word document for your questions if necessary
5. Submit your file in LMS before due date.
6. If you have any questions, feel free to ask in Rocket Chat.

Best of Luck and Happy Coding!