QUICKSTART
Over 35 Years Of Technology Training

Software Engineering Bootcamp

Project 3: Task Management Web App

Introduction

Your task is to create a fully operational web application for managing "to-do" lists using React. Optionally, you may integrate Redux for state management and React Router for navigation.

In practice, React & Redux is a popular combination of tools to use on the frontend of a web app -- - the part the user sees and interacts with. Most web-apps hinge on the users' ability to create/read/update/delete data, also known as performing "CRUD" operations.
Some examples of CRUD operations offered in popular React web apps are:

- Spotify allows users to create/read/update/delete playlists of songs.
- Facebook allows users to create/read/update/delete posts.
- What'sApp allows users to create/read/update/delete contacts

In production apps, all that data is typically stored in a database, and the web-app reads and writes data to that database by making calls through an Application Programming Interface (API) -- code that lives on the server and can communicate directly with the database.

Workflow Requirements

The following requirements are related to how you go about building your project

Planning phase - should be completed prior to beginning the development phase and/or touching any code.

Wireframes can be done on paper or using any number of widely available applications. A free one that may be useful is draw.io.

User stories
1) Write out at least three user stories

Wireframes
2) Create wireframes for each view of your app

3) Write out your app's state tree (remember the contact form will manage its own state for the form fields)
4) Write out a list of the container and presentational components you intend to use in your app

Development phase

1) Create a GitHub repository on Github.com (before you start coding)
2) Clone it to your local machine (before you start coding)
3) Make frequent commits throughout your development that are descriptive, such as "adds todos reducer" (*throughout development/coding process*)

Technical Requirements

The following requirements are related to what your **code** should contain:

1) This should be a React app based on Create React App
2) The app should contain at least two views: ***/todos*** and ***/contact***
3) When a user navigates to /todos, they should be presented with a view that:

- Display a list of the todo items
- Displays a form (text input and submit button) that allows users to add a new item to the list
- Offers a way for a task to be marked as "completed" and clearly indicates this status visually (e.g. strike-through effect)
- Offers a way for a task to be removed from the list
- Offers a way to view either
    - all todos
    - completed todos
    - incomplete todos
- When todos are added/updated/marked as complete, these changes should immediately be reflected in your internal state

4) When a user navigates to ***/contact***, they should be presented with a view that:

- Displays a contact form that displays the following fields
    - first name field
    - last name field
    - email field
    - comments field
    - Renders a the form as a controlled component such that after entering text into any of the fields, the form's state has changed

5) There should be at least 10 custom CSS rules used throughout the components of your React app
6) You must have a horizontal nav bar at the top of your site
7) You must have at least one example of content side-by-side (e.g. the "new todo" form in the example screenshot)

QUICKSTART
Over 35 Years Of Technology Training

Deliverables

1) Your user stories
2) A collection of wireframes - one for each view of your app
3) Your state tree
4) Your list of container and presentational components from the planning phase
5) Your app source code should be available for viewing in your GitHub repository
6) A ***readme.md*** file in the root project folder that contains the following information about your project:

- Your name
- Overview/description of the project
- Details on how to use it or what functionality is offered
- Technologies Used (***.html***, ***.css***)
- Ideas for future improvement (minimum of 3)

7) Your repository should contain at least 15 commits and should reflect a consistent commit history
8) Hosting on GitHub pages using the gh-pages package (https://www.npmjs.com/package/gh-pages)

Submit your GitHub link and hosting link (include everything on a word document and upload the document under Project Submission section). All deliverables should be included on GitHub and the site should be made publicly available using a hosting service.

You will be evaluated on your ability to meet both the **workflow requirements** and the **technical requirements**.

Project Grading

To pass the project, Instructor should take below criteria into consideration while grading this project and decide whether to Pass or Fail the student.

- **Functionality**
- **Robustness**
- **Creativity, styling, user experience**
- **Code quality.**
- **GitHub structure**
- **Documentation, Installation instructions, Comments**