

Assignment Eight

Please read:

- This is your eighth assignment which you need to submit on the LMS. This is an assignment on Typescript.
- Work on the questions given below and be ready with your solution in CodeSandbox. You have to submit your CodeSandbox link below on this page.
- Late Submission: The deadline to submit this assignment is 19th February 2024, 6:00PM IST.

Important Instructions:

1. Make sure that you follow the typescript rules as it will help you build your typescript muscle.
2. Do not copy from someone else as that would be cheating.

ex01: Header Component

challenge

Create a React component named `Header` that takes a single prop `title` (string) and renders it within an `<h1>` element. Define prop types for the component.

ex02: Bookmark Form Component

challenge

Your task is to create a React component named `BookmarkForm` that allows users to add bookmarks by entering a title and a URL. When the form is submitted, a new bookmark should be added to the list of bookmarks.

- The component should receive a prop named `onAddBookmark` that takes two parameters: `title` (string) and `url` (string).
- The form should include two input fields for entering the title and URL of the bookmark.
- When the form is submitted, the `onAddBookmark` callback should be invoked with the current values of `title` and `url`.
- After adding the bookmark, the `title` and `url` fields should be cleared.

ex03: Bookmark List Component

challenge

Create `BookmarkList` that displays the list of bookmarks. Each bookmark should have a "Remove" button that allows users to remove the bookmark from the list.

- The component should receive a prop named `bookmarks` which is an array of objects representing bookmarks. Each bookmark object should have properties `id` (number), `title` (string), and `url` (string).
- The component should also receive a prop named `onRemoveBookmark` that takes a single parameter `id` (number).
- When the "Remove" button is clicked, the `onRemoveBookmark` callback should be invoked with the `id` of the bookmark to be removed.

ex04: Make bookmark a link in Bookmark List

Challenge

Your goal is to make each bookmark title clickable so that when a user clicks on a bookmark, it opens the corresponding URL in a new tab.

ex05: Updating App.js

Challenge

You have already implemented the `BookmarkForm` and `BookmarkList` components. Now, your task is to integrate these components into the `App.js` file and add the necessary logic for the `handleAddBookmark` and `handleRemoveBookmark` functions.

- Set up a state variable `bookmarks` using the `useState` hook to manage the list of bookmarks. Initialize it as an empty array.
- Implement the `handleAddBookmark` function that takes `title` and `url` as parameters. Add the new bookmark to the existing list of bookmarks using the `setBookmarks` function.
- Implement the `handleRemoveBookmark` function that takes `id` as a parameter. Update the `bookmarks` state using the `setBookmarks` function with the filtered list.
- Integrate the `Header`, `BookmarkForm`, and `BookmarkList` components into the `App` component.

ex06: defining actions as unions

challenge

Define two actions as discriminated unions: `AddBookmarkAction`, and `RemoveBookmarkAction`. Each action should have a `type` property that distinguishes the action type.

ex07: creating the bookmark reducer

challenge

Create a reducer function named `bookmarkReducer` that handles the `AddBookmarkAction` and `RemoveBookmarkAction` actions. Implement the logic to add or remove bookmarks from the state.

ex08: adding initial state to reducer

challenge

In `App.js`, pass the `bookmarksReducer` as the reducer function and the `initialState` as the initial state to the `useReducer` hook.

ex10: updating actions in the app component

challenge

Modify the App component to include the new reducer-based bookmarking system. Dispatch the actions appropriately based on user interactions. Make sure to remove the `useState` code as it is not required now.

ex11: Declaring Bookmarks Context

challenge

Create a context named `BookmarksContext` that will hold the state and actions related to bookmarks. Define types for the context, including the bookmark object and the actions such as adding and removing bookmarks.

- Create a context named `BookmarksContext`
- Create a custom hook named `useBookmarksContext` that returns the context using the `useContext` hook.
- Define a `BookmarksProvider` component.
- The `BookmarksProvider` should internally use the `useReducer` hook to manage the state using the `bookmarksReducer` and an initial state of `{ bookmarks: [] }`.

ex12: implementing bookmarks context in the app

challenge

Wrap your App component with the `BookmarksProvider` to make the bookmarks context available throughout your app.

ex13: Bookmark Form with Context

challenge

Modify the `BookmarkForm` component to use the `BookmarksContext` for adding bookmarks. Instead of using the `onAddBookmark` prop, dispatch the `ADD_BOOKMARK` action to add a new bookmark to the context state.

ex14: Bookmark List with Context

challenge

Modify the `BookmarkList` component to use the `BookmarksContext` for displaying and removing bookmarks. Instead of using the `bookmarks` prop and `onRemoveBookmark` prop, use the context state and dispatch the `REMOVE_BOOKMARK` action to remove a bookmark.

ex15: clean app.js

challenge

Clean the props provided to the `BookmarkForm` and `BookmarkList`. As well as, remove the `useReducer` code from the `App.js` as it is not required now.