Create a Travel WishList App in ReactJS

Type: Projects

Skill: React

Component DevelopmentError HandlingProps and State ManagementReact HooksUI/UX Design

Medium

Develop a simple ReactJS application named "TravelWishList". This application will consist of one component - WishList. This component will render a list of travel destinations that a user wishes to visit. These locations will be passed to the component using props.

**Implement the following functionalities:**

**Initialization of WishList Component:**

* On app initialization, the WishList component should display a list of travel locations based on the initial props.
* Ensure the list is correctly rendered in the DOM.

**Props and Component Update:**

* Pass an initial set of locations to the WishList component through props.
* Modify the props by adding or removing a location.
* Use the useEffect hook to detect these changes and log a message to the console.

**Console Log on Change:**

* Log a message to the console every time the props change, showing the previous and current values of the wish list.

**Adding Locations through a Form (Bonus):**

* Create a simple form within the component to add a new location.
* When a new location is submitted, update the list and check if the useEffect hook correctly logs the change.

**Wish List Rendering:**

* Render each location passed through props.
* Ensure the list updates and rImplement the following functionalities:enders correctly when new locations are added.

**Boundary and Edge Cases:**

* Handle scenarios with an empty list, duplicate locations, and null or undefined inputs.

**UI/UX Checks:**

* Ensure the list and, if implemented, the form are visually appealing, easy to read, and user-friendly.

**You need to complete the following files:**

1. ./src/app/wish-list/WishList.js
2. ./src/App.js

**Notes:**

1. Do not change file names, class names, or method declarations.
2. Focus on the functionality; UI styling is not required for this task.

import { bottomNavigationActionClasses } from '@mui/material';

import React, { useState, useEffect } from 'react';

import '../../App.css';

const WishList = ({ initialLocations }) => {

  const [locations, setLocations] = useState(initialLocations);

  const [newLocation, setNewLocation] = useState('');

  const [error, setError] = useState('');

  useEffect(()=>{

    console.log("Locations changed now");

  },[locations])

  const addLocation = (newLocation) => {

    // check if location is not empty and does not exists alread

    if (newLocation) //true if not empty

    {

    }else{

      setError('Location is empty')

      return;

    }

      if(locations.find((x)=>newLocation==x)){

        //already found, so cannot add

        console.log("this location already found");

        setError("This location is already found. So we cannot add")

        return;

    }else{

      setLocations([...locations,newLocation]);

      setError('')

    }

    // add new location to whish list and clear erros

  };

  const removeLocation = (locationToRemove) => {

    // remove locations locations

    setLocations(locations.filter((x)=>x!=locationToRemove));

    setError('')

  };

  const handleSubmit = (e) => {

    // prevent default submistion and call add location form

    // clear new location state

    e.preventDefault();

    switch(e.target.value){

      case "Add":

        addLocation(newLocation);

        break;

      case "X":

        removeLocation(newLocation);

    }

    setNewLocation('');

  };

  return (

    <div className="wish-list-container">

      <h2>Travel Wish List</h2>

      <ul className="location-list">

        {/\* list down locations using map  \*/}

        {

          locations.map((l,i)=><li key={i}>

            {l}

          </li>)

        }

      </ul>

      {/\* display locations erros  \*/}

      {

        error

      }

      {/\* create location form and call addLocation method on submission \*/}

      Add new location

      <form onSubmit={handleSubmit}>

      Location: <input type="text" onChange={(e)=>setNewLocation(e.target.value)} />

      <input type="submit" value="Add"  />

      <input type="submit" value="X"  />

      </form>

    </div>

  );

};

export default WishList;

import React from 'react';

import { render, fireEvent } from '@testing-library/react';

import WishList from '../app/wish-list/WishList';

test('renders initial list of locations correctly', () => {

  const initialLocations = ["Paris", "Tokyo", "New York"];

  const { getByText } = render(<WishList initialLocations={initialLocations} />);

  initialLocations.forEach(location => {

    expect(getByText(location)).toBeInTheDocument();

  });

});

test('adds new location to the list', () => {

  const initialLocations = ["Paris", "Tokyo", "New York"];

  const { getByPlaceholderText, getByText } = render(<WishList initialLocations={initialLocations} />);

  const input = getByPlaceholderText('Add new location');

  const addButton = getByText('Add');

  const newLocation = 'London';

  fireEvent.change(input, { target: { value: newLocation } });

  fireEvent.click(addButton);

  expect(getByText(newLocation)).toBeInTheDocument();

});

test('removes location from the list', () => {

  const initialLocations = ["Paris", "Tokyo", "New York"];

  const { getByText, queryAllByText, queryByText } = render(<WishList initialLocations={initialLocations} />);

  const locationToRemove = "Tokyo";

  const removeButtons = queryAllByText('X');

  // Find the remove button corresponding to the location to be removed

  const removeButton = removeButtons.find(button => button.closest('li').textContent.includes(locationToRemove));

  fireEvent.click(removeButton);

  expect(queryByText(locationToRemove)).toBeNull();

});

test('does not add duplicate locations', () => {

  const initialLocations = ["Paris", "Tokyo", "New York"];

  const { getByPlaceholderText, getByText, queryByText } = render(<WishList initialLocations={initialLocations} />);

  const input = getByPlaceholderText('Add new location');

  const addButton = getByText('Add');

  const duplicateLocation = 'Paris';

  fireEvent.change(input, { target: { value: duplicateLocation } });

  fireEvent.click(addButton);

  // Expect that the duplicate location is not added to the list

  expect(queryByText(duplicateLocation)).toBeInTheDocument();

});

test('logs message to console on props change', () => {

  const initialLocations = ["Paris", "Tokyo", "New York"];

  const updatedLocations = ["Paris", "London", "New York"];

  const consoleSpy = jest.spyOn(console, 'log');

  render(<WishList initialLocations={initialLocations} />);

  expect(consoleSpy).toHaveBeenCalledWith('Previous Wish List:', initialLocations),'Current Wish List:', updatedLocations;

  // expect(consoleSpy).toHaveBeenCalledWith();

});