Build a Dynamic Itinerary using ReactJS

Type: Projects

Skill: React

Cleanup FunctionsComponent ManagementData Fetching with useEffectData PropagationDynamic Component Addition/Removal

Medium

Develop a ReactJS application named "DynamicItinerary". The application will include a parent Trip component and a child Destination component. The Trip component maintains an array of destinations for a trip, and renders each destination using the Destination component.

**Implement the following functionalities:**

**Trip Component:**

* Manages an array of destinations.
* Renders each destination using the Destination component.

**Destination Component:**

* Displays the information of a single destination.
* Use the useEffect hook to simulate fetching data on initialization.
* Implement a cleanup function in the useEffect hook to log a message to the console when the destination is removed.

**Bonus:**

* Implement functionality to dynamically add and remove destinations.
* Ensure that removing a destination triggers the cleanup function in the Destination component, logging the removal.

**Test Cases:**

* Initialization of Trip Component: On app initialization, display a list of destinations. Render each destination using the Destination component.
* Data Fetching in Destination Component: Simulate data fetching in Destination with useEffect on component initialization. Ensure that the fetched data is displayed correctly.
* Logging with Cleanup Function: Log a message to the console when a Destination component is unmounted.
* Dynamically Add Destinations (Bonus): Provide an interface in the Trip component to add new destinations. Ensure the list updates with a new Destination component for each addition.
* Dynamically Remove Destinations (Bonus): Include a remove option for each Destination. Removing a destination should update the list and trigger the cleanup function.
* Data Propagation: Ensure accurate data passing from Trip to Destination components.

**Boundary and Edge Cases:** Handle scenarios with an empty destination list, removing non-existent destinations, and adding destinations with incomplete data.

**Interaction Tests:** Test adding and removing destinations, ensuring all updates are rendered correctly.

**UI/UX Checks:** Ensure the destination list and interactive elements are user-friendly and organized.

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

1. ./src/app/destination/Destination.js
2. ./src/app/trip/Trip.js

**Notes:**

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