# React JS Tasks: Create Advanced State Management System with useReducer and Context API:

Let’s Assume, you have been tasked by your Project Manager to refactor your Quiz App using Context API.

Your Task:

1. Duplicate `src` folder to `src-no-context`.
2. Review data flow and passed props
3. Identify props drilling problem.
4. Use the Context API to fix the (very small) props drilling problem.
5. Create a new context ‘QuizContext’ with the reducer we created earlier.
6. Create a custom provider component ‘QuizProvider’ and provide all the state to the app.
7. Create a custom hook to consume state all over the application
8. Delete all unnecessary props.

Note: How you need state right in App components. This means you need to wrap the whole App into the Context (Hint: try in `index.js` or `main.js`)

Task 2: Display City Marker on Map

1. Consume `cities` props using custom hook `useCities()`
2. Map the `cities` props with Marker component.

Task 3: Interact with the Map

1. When you click city name from city list show their location in map for this use previous learning of useSearchParams() hook and get lat and lng.
2. Set center props value of map with lat and lng.
3. See if maps marker is changing or not.
4. If not, try to implement useMap() for more visit official documentation of leaflet.
5. Hint 1: create one component as <ChangeCenter position={[lat, lng]} /> inside this component use your useMap() hooks logic
6. Hint 2:   
   function ChangeCenter({position}){  
    const map = useMap()  
    map.setView(position)  
    return null  
   }
7. Use useEffect to detect lat, lng change as well and setMapPosition

Task 4: Detect click on map and navigate to form with useNavigate() hook

* 1. Use useMapEvents hook
  2. Hint 1: useMapEvent({ click: e => navigate(`form?lat=${see what goes here}&lng=${see what goes here}`) }) // comment: console ‘e’ first

Task 5: Fetch detail using provided API and fill up form

// "https://api.bigdatacloud.net/data/reverse-geocode-client?latitude=0&longitude=0"

Use this api link to fetch information based on latitude and longitude