

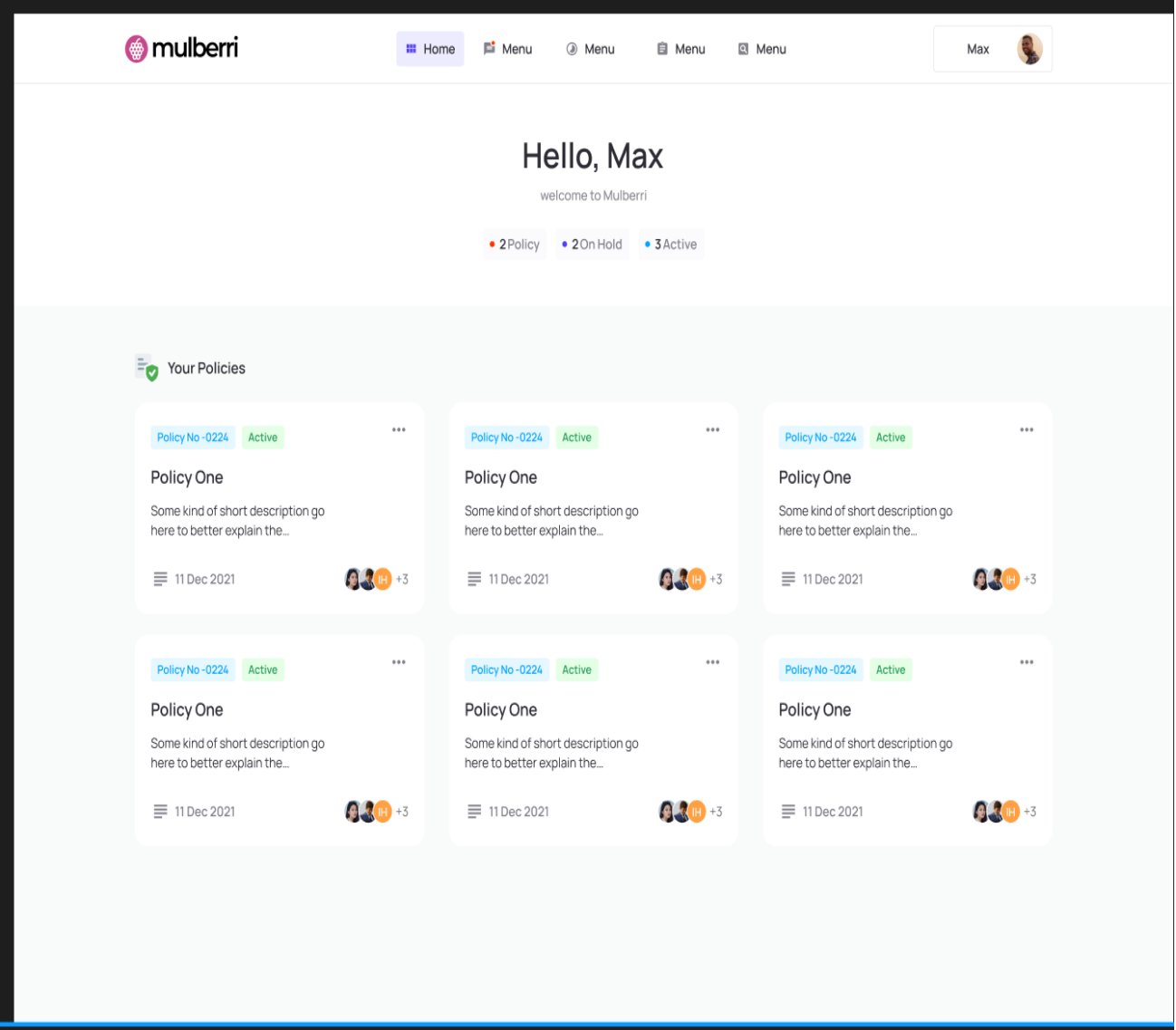
In this exercise, you are required to:

1. Build a Single Page Application (SPA) using ReactJS.
2. Ensure the application is responsive across devices: mobile ($\leq 767\text{px}$), tablet ($\geq 768\text{px}$ and $\leq 992\text{px}$), and desktop (above 992px).
3. On the landing/first screen:
 - a. Create a header that displays the logo, navigation bar (Home, Message, Theme(dark/light), Search), and the logged-in user's information.
 - b. Add a banner showing "Hello [Logged-in user name]" along with a static message "Welcome to Mulberri." Display dynamic information about the the total number of policies, count of policy statuses such as "Active," "On Hold,".
 - c. Create a list of policies (as shown in the image), each displaying its status, like "Active" or "On Hold."
 - d. Next to the "Your Policies" title, add a button labeled "Add Policy" (when clicked, a modal should open with fields to input the policy number, status, title, description, date, and a list of users)
 - e. Each policy will have an ellipsis icon. When clicked, a dropdown menu will appear with an option to "Edit." Selecting "Edit" will open the same modal used for adding a new policy, but with pre-filled data, allowing the user to update the policy details (except for the policy number).
 - f. Click on each policy to open the corresponding chat box, as shown in Screen 2
4. In the header, we have navigation options labeled "Messages." When clicked, the page redirect from landing/first screen to the second screen, where the chat corresponding to each policy will appear.
 - a. On click on corresponding policy, chat box open same as show in the Screen 2.

Skills:

- Use **ReactJS/NextJS** for the development.
- Implement **Redux** if needed for state management.
- The **UI** should be flawless and match the design closely.
- Follow **web standards** and best practices in UI development.

Screen 1:



Screen 2:

