StarTree Interview Exercise

Position: UI Developer

(Due 02/21/2021)

For this exercise, you will develop a simple, lightweight React application. Let's call the application The Timekeeper. The Timekeeper serves a single purpose - remember a particular time interval at all times.

Time Interval

The Timekeeper remembers the time interval as following properties:

- `duration`
- `endTime`
- `startTime`

The relationship between these properties is as follows:

- `duration` is a default (*DEFAULT*) value or a user input (*CUSTOM*) value **in minutes** (you get to choose the default value).
- `endTime` is always current time **in milliseconds** at the time of a user activity (see **User Activities** below).
- `startTime` is always `endTime - duration` **in milliseconds**.

User Activities

The Timekeeper recognizes following user activities:

- **Screen/Page Load**

  Whenever a user navigates to any screen/page in The Timekeeper (see **Screens/Pages** below). This also includes user reloading/refreshing an existing screen/page with the browser refresh button.

  When detected, The Timekeeper does the following:

  1. Read the `duration` from the URL query string (see **URL Query String** below).
  2. If the duration doesn't exist in the URL query string or exists but is the same as the one The Timekeeper remembers (in application state), leave the remembered `duration` as is.
  3. If the `duration` exists in the URL query string and is different from the one The Timekeeper remembers (in application state), update the

remembered `duration` and mark it as *CUSTOM* (indicating it's definitely not the initial *DEFAULT* value).
4. Capture the current time as `endTime`.
5. Calculate `startTime`.
6. Write new `duration`, `endTime` and `startTime` to the URL query string (see **URL Query String** below).

- **User Input**

  Whenever a user inputs `duration` on any screen/page (see **Screens/Pages** below).

  When detected, The Timekeeper does the following:

  1. Read the user input `duration`.
  2. Regardless of whether the user input `duration` is the same as the one The Timekeeper remembers (in application state), update the remembered `duration` and mark it as *CUSTOM* (indicating it's definitely not the initial *DEFAULT* value).
  3. Capture the current time as `endTime`.
  4. Calculate `startTime`.
  5. Write new `duration`, `endTime` and `startTime` to the URL query string (see **URL Query String** below).

Outside of the above recognized user activities, The Timekeeper will never update the remembered time interval.

URL Query String

- Whenever The Timekeeper updates the remembered time interval (in application state), it also adds updated `duration`, `endTime` and `startTime` to the URL query string as follows:

  ```
  <application_url>/<application_path>?duration=<duration_value>&endTime=<endTime_value>&startTime=<startTime_value>
  ```

- The user is not expected to access The Timekeeper with appropriate time interval properties in the URL query string. The user may choose to enter just the application URL or application URL with partial time interval properties (for example just the `startTime`) and The Timekeeper shall then add updated time interval properties to the URL query string as per the recognized user activities above.
- If the user does access The Timekeeper with appropriate time interval properties in the URL query string, The Timekeeper shall then update the remembered time interval properties (in application state) as per the recognized user activities above.
- **[BONUS]** The URL query string is maintained when navigating to different screens/pages in the application (see **Screens/Pages** below). For example, the user

may choose to append a new query string key-value pair to the URL and The Timekeeper shall make sure it always remains in the URL
- At any point of time, the user shall be able to copy The Timekeeper URL with the latest time interval properties in the query string.

Screens/Pages

The Timekeeper shall consist of three identical screens/pages:

- **ScreenA** available at `<application_url>/screenA` and also the default landing page of The Timekeeper
- **ScreenB** available at `<application_url>/screenB`
- **ScreenC** available at `<application_url>/screenC`

All three screens shall have the following:

- **Screen Title** (ScreenA or ScreenB or ScreenC)
- **Links** to navigate to other two pages
- **Display and Input** for `duration`
- **Display** for type of `duration`, *DEFAULT* or *CUSTOM*
- **Display** for `endTime`
- **Display** for `startTime`

While reusable component design is encouraged, the three screens, although identical, need to be three different components. The components used to display/generate contents on the screens however, need to be reused.

Notes

- You may assume that the user input is always valid and skip validations.
- You may add new properties to the time interval as per your convenience.
- The URL query string however shall never expose any time interval properties other than the `duration`, `endtime` and the `startTime`.
- The basic coding requirement is that The Timekeeper shall be a React application. Beyond that you are free to use any technologies/frameworks/packages to develop The Timekeeper.
- The focus of this exercise is on component based design. The Timekeeper shall consist of well designed reusable components avoiding unnecessary complexities while still satisfying all the edge cases that may arise while implementing the Time Keeper functionality.

Happy coding!