1. The states that Expo React App can enter

1) Active State: This is the normal state of the app when it's running in the foreground, and the user is actively interacting with it.
2) Background State: When the app loses focus and goes into the background, it enters the background state. In this state, the app is still running but may be paused or have limited resources allocated to it by the operating system.
3) Inactive State: In this state, the app is still visible on the screen but isn't responding to user input. This can happen when another app or system dialog is displayed on top of the app, causing it to lose focus temporarily.
4) Suspended State: When the app is in the background for an extended period and the operating system needs to reclaim resources, it may be suspended. In this state, the app is not actively running, but it's still resident in memory. When the user brings the app back to the foreground, it resumes from the suspended state.
5) Terminated State: If the operating system needs more resources and decides to terminate the app while it's in the background, it enters the terminated state. In this state, the app is no longer resident in memory, and if the user tries to open it again, it will start from scratch.
6) Offline State: In some cases, the app may lose network connectivity and enter an offline state. This can happen due to a loss of internet connection or when the device is in airplane mode. Handling offline states is crucial for apps that rely on network resources.
7) Error States: Apps can also encounter error states due to various reasons such as network errors, runtime errors, or unexpected behavior. Handling error states gracefully is essential to provide a good user experience.

2. States of LoveStorying and Happening Things

State	Why consider	What must happen
Active State	This is the normal state of the app when users are actively engaging with it.	Users should be able to navigate between different pages, read stories, create new stories, save stories, and interact with the questionnaire without any hindrance.
Background State	The app may enter the background state when users switch to another app or receive a phone call.	The app should save the current state and data, ensuring that users can seamlessly resume their activities when they return to the app.
Inactive State	Inactive state occurs when the app is visible but not receiving user input, such as when a system dialog or notification covers the app.	The app should maintain its state and data integrity. Users should be able to resume their activities once the app becomes active again.
Suspended State	The app may be suspended by the operating system to reclaim resources.	The app should gracefully handle suspension by saving its state and data. Upon resuming, users should be able to continue where they left off without data loss.
Terminated State	The app might be terminated by the operating system due to resource constraints.	The app should handle termination gracefully by persisting essential data and providing a seamless user experience when relaunched.
Offline State	Users may lose network connectivity.	The app should notify users about the loss of connectivity and provide offline capabilities where possible, such as allowing users to read

		previously saved stories or complete questionnaires offline.
Error States	Errors can occur due to various reasons, such as network issues or server errors.	The app should handle errors gracefully by providing informative error messages to users and offering options for recovery or troubleshooting.