**Feature 1: Dark Mode Toggle**

**User Story:**  
As a user, I want to be able to switch between light and dark mode at night, so I can use the app comfortably for my eyes.

**Technical Requirements:**

* Add a darkMode state in React.
* Change CSS or use a class on the body to apply the dark theme.
* Save the user setting in localStorage so it is remembered when reopening the app.

**Files Likely to be Modified:**

* App.jsx (to add toggle and state)
* app.css (to add dark mode styles)

**Potential Challenges:**

* Ensuring the theme applies to all smaller components (Buttons, Display).
* Handling colors so they are consistent in dark mode.

**Feature 2: Calculation History**

**User Story:**  
As a user, I want to see a list of recent calculations, so I can easily refer back to them without retyping.

**Technical Requirements:**

* Create an array to store recent operations in React state.
* Display the previous operations in a separate component.
* Store the data in localStorage even after closing the app.

**Files Likely to be Modified:**

* App.jsx (to manage state)
* History.jsx (new component)
* app.css (to style the operations list)

**Potential Challenges:**

* Managing the size of the history so it does not affect performance.
* Formatting the operations list clearly and readably.

**Feature 3: Priority Buttons for Operations**

**User Story:**  
As a user, I want important buttons (like +, -, ×, ÷) to be highlighted with different colors, so I can easily distinguish them during use.

**Technical Requirements:**

* Add a property for each button to specify its type (digit / operator).
* Use CSS to color the buttons based on their type.
* Ensure colors are consistent with both dark and light themes.

**Files Likely to be Modified:**

* Button.jsx (button component)
* app.css (for different colors based on type)

**Potential Challenges:**

* Handling dark and light themes with button colors.
* Making sure the colors do not reduce readability.