The screen recording of the app is attached.

I've used a redux store to make the rendering smooth.

If node\_modules packages are not installed just run **npm i**

In order to run the application please navigate to the unzipped folder in the terminal and just use **npm start** command

**Search and Filter:** You can use the search bar to search for emojis by their names. Just start typing in the search bar, and the table will dynamically update to show emojis that match your search criteria. Additionally, you can filter emojis by their categories using the dropdown menu.When you've selected a certain category then you can search in that category.

**Pagination:** The emoji table displays 10 emojis per page. If you have more than 10 emojis matching your search or filter criteria, you can use the "Next" and "Prev" buttons at the bottom of the page to navigate through the pages and see more emojis.

**Responsive Header:** On smaller screens (width less than 500px), the header will adapt to a responsive design. You'll see a hamburger menu icon on the top right corner. Clicking on the icon will reveal the search bar and filter dropdown options.

**Emoji Table**: The emojis are displayed in a sleek and modern tabular format. The table columns include "Emoji," "Name," "Category," and "Group."

**Loading Indicator:** If the data is still being fetched from the server, you'll see a loading indicator. Once the data is fetched and available, the emojis will be displayed.

**Emoji Rendering:** The emojis are rendered using their HTML codes. Each emoji can have multiple HTML codes if it has different skin tones or variations. The app will combine and display them accordingly.

**Flag Emojis:** Currently, the flag emojis are not rendered correctly due to the way they are represented in the API response. This issue is being investigated and will be fixed in a future update.

**Max Width:** The maximum width of the emoji container is set to 800px to ensure a better viewing experience on larger screens.

**Using Redux Store:**

The application uses Redux store to manage the state and keep track of the emojis data fetched from the API.

The Redux store is initialized in the store.js file using configureStore from @reduxjs/toolkit.

We have a single state emojis in the store that holds the array of emojis data.

The emojis state is populated with the data fetched from the API and stored in the store using the setEmojis action.

The emojis data is then accessed in components using the useSelector hook from react-redux, which allows us to select the desired state from the store.

**Enjoy Exploring Emojis:** Have fun exploring the vast world of emojis! The app showcases a wide variety of emojis from different categories and groups. Click on each emoji to copy it to your clipboard and use it wherever you like!