Deck of Cards

Your assignment is to code, in JavaScript or TypeScript, a set of classes that represent a deck of poker-style playing cards. (Fifty-two playing cards in four suits: hearts, spades, clubs, diamonds, with face values of Ace, 2-10, Jack, Queen, and King.)

Representing Cards

Within one of your classes, you must provide two operations:

- **shuffle()** - Shuffle returns no value, but results in the cards in the deck being randomly permuted. Please do not use library-provided "shuffle" operations to implement this function. You may use library-provided random number generators in your solution.

- **dealOneCard()** - This function should return one card from the deck to the caller. Specifically, a call to shuffle followed by 52 calls to *dealOneCard* should result in the caller being provided all 52 cards of the deck in a random order. If the caller then makes a 53rd call to *dealOneCard*, no card is dealt.

Displaying Cards

The assignment should include a web-based UI displaying cards, the *shuffle* operation, and the *dealOneCard* operation in action.

General Instructions

We have intentionally left many details of this assignment vague. You should follow the principle of least surprise in making reasonable decisions regarding the implementation.

While this is a trivial assignment, pretend that this code will become a foundational part of a new product. It should showcase the development practices you would use when working with a team of developers. Take whatever measures you feel are required for your code to meet this bar within the scope of the allotted time.

Be prepared to answer some architectural questions and questions around development decisions. We prefer assignments in modern JavaScript (ES2015+) or up-to-date TypeScript.

As for UI frameworks, Jive often works with Angular, but feel free to use other frameworks such as React or Vue.js if you are more comfortable with them.