1. Create a new Function Component called GuessingGame that renders the following:
   * A message that says "I am thinking of a number between 1 and 100. Guess the Lucky Number!"
   * A form with an input for the user's guess and a "Guess" button to submit
   * A paragraph that says "You have made 0 guesses," and displays the number of guesses the user has made
   * A paragraph that displays a message to the user indicating whether the user has guessed correctly, too high, or too low
   * A "Reset" button that resets the game by picking a new number to guess and resetting the number of guesses back to 0
2. Use the useState hook to keep track of the following data:
   * The number the user is trying to guess (lucky number)
   * The user's current guess
   * The number of guesses the user has made
   * (Optional) A message/hint to display to the user
3. Create a function within GuessingGame that will assign a new lucky for the user to guess. This function should be called if a lucky number does not exist when the game first loads, and then also any time the user clicks the Reset button.

Hint: You can use the Math.random() function to generate a random number between 1 and 20. [Math.random() documentation](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/random" \t "_blank)

1. Create a function that is called when the user clicks the "Guess" button. This function should:
   * Check if the user's guess is the same as the lucky number. If it is, display a message that congratulates the user (e.g. "You guessed the Lucky Number!")
   * If the user's guess is higher than the lucky number, display a message that says "Your guess is too high!"
   * If the user's guess is lower than the lucky number, display a message that says "Your guess is too low!"
   * Increment the number of guesses the user has made by 1
   * You might want to use another state variable to represent the message/hint that should be displayed to the user based on their guess.
2. Use Local Storage to persist the game's lucky number and the number of user guesses. If the app is refreshed, the game should continue where it left off by using the saved lucky number and displaying the total number of guesses made so far.

Hint: You can manually handle reading and writing data from/to Local Storage, or you can use the usePersistence hook to save data to Local Storage.

1. Use CSS Modules and Bootstrap Components to make your app unique and intereseting.