Project: Design a Guessing-Number App, where the user has to guess a number.
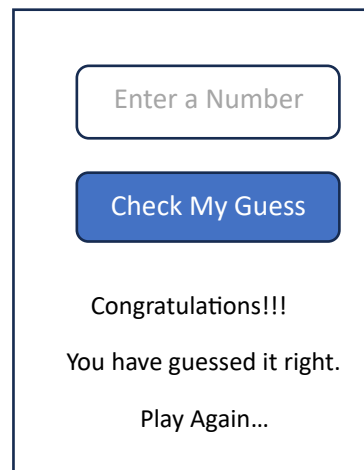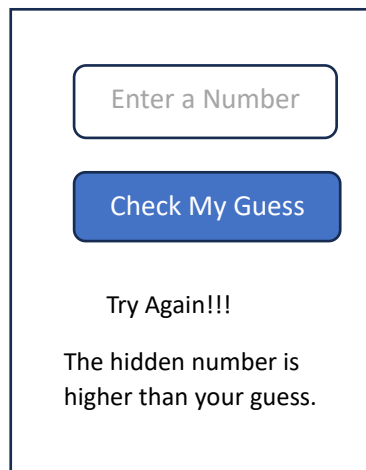
- You can use Math.random() below to generate a random number between 0 and 100.
  o Math.floor(Math.random() * 101);

- The above generated number acts as a **hidden-number** that user has to guess.

- The left-hand side diagram below shows the screen-output when the user guess was incorrect.

- The right-hand side diagram below screen shows the screen-output when the user guessed the number correctly.

Objectives:

- Creating multiple Components.

- Communicating between Parent and Child Component using 'props'.

- Implementing the 'useState' hook and understanding re-rendering of components.

Tasks: **10 points/task**

Your Guessing-Number App will consist of single-screen and two components.

1. Create a child-component (UserGuessInput), which should be responsible for rendering the Text-Input component.

2. The parent component should hold –

    a. the above UserGuessInput component
    b. a button (titled - Check My Guess), and
    c. a Text component (to render a message).

3. When the user enters a number in the text-input and clicks the button, an output message should be displayed in the Text component indicating whether the guess was correct or wrong.
    a. If the guess was wrong, give a hint to user stating if the hidden number is lower or higher than their guess.
    b. If the guess was correct, print a message outputting 'Congratulations'. Generate a new hidden-number so that the player can play again.

4. Enable the communication between parent and child component using 'useState' hook and 'props'. Pass the desired functionalities from parent to child to ensure the proper functioning of the App.

Submission:

-   Screenshots of the App for both the scenarios (correct, incorrect guess)
-   Code from App.js and UserGuessInput.js files.

Put the above code-and-screenshots in a word or pdf document, and upload the document through the submission link made available on blackboard.