Intro to JavaScript Week 6 Coding Assignment

Points possible: 100

URL to GitHub Repository:

https://github.com/lbell/bootcamp_wk6_coding_assignment

URL to Your Coding Assignment Video:

https://drive.google.com/file/d/1SX-gltKa0jFWrFZ0g-2HgeXjXgnD1gHh/view?usp=sharing

Instructions: In Visual Studio Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

For the final project you will be creating an automated version of the classic card game *WAR*. You do not need to accept any user input, when you run your code, the entire game should play out instantly without any user input.

There are many versions of the game *WAR*, but in this version there are only 2 players and you don't need to do anything special when there is a tie on a round.

Think about how you would build this project and write your plan down. Consider classes such as Card, Deck, and Player and what fields and methods they might each have. You can implement the game however you'd like (i.e. printing to the console, using alert, or some other way). The completed project should, when run, do the following:

- Deal 26 Cards to two Players from a Deck.
- Iterate through the turns where each Player plays a Card
- The Player who played the higher card is awarded a point
    - Ties result in zero points for both Players
- After all cards have been played, display the score and declare the winner.

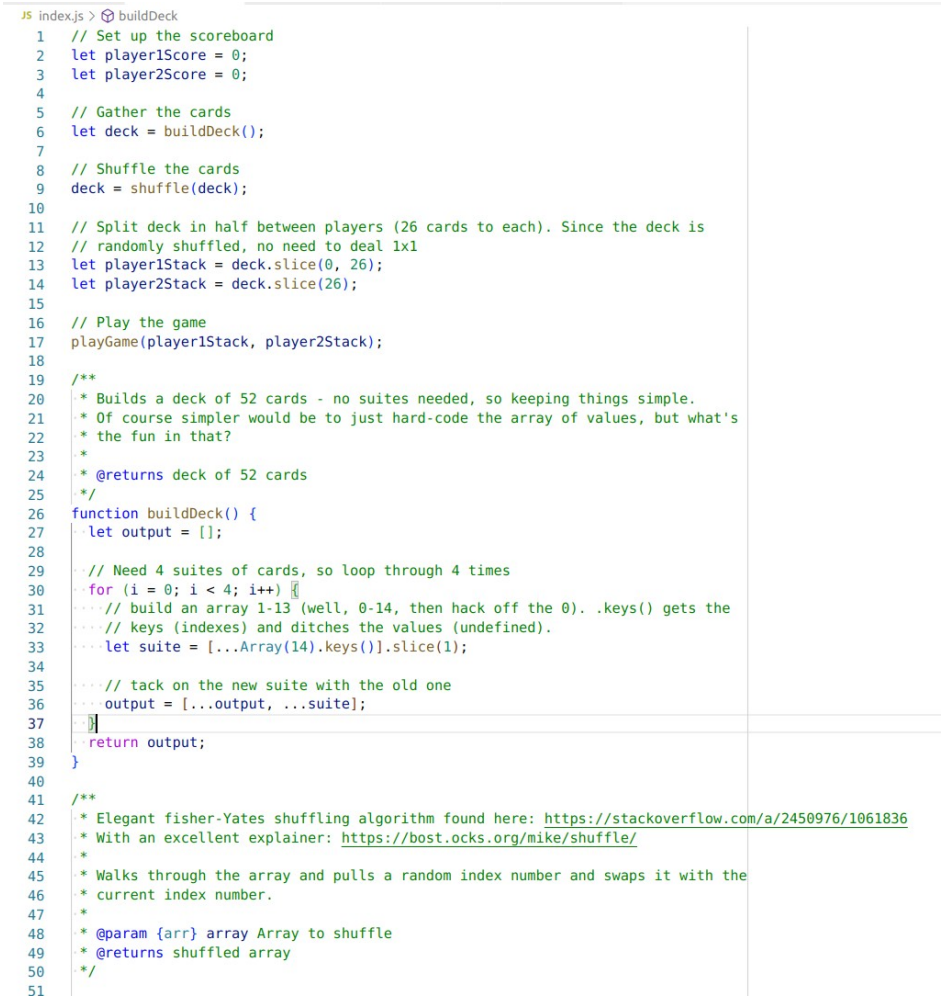
Write a Unit Test using Mocha and Chai for at least one of the functions you write.

Video Steps:

Create a video, up to five minutes max, showing and explaining how your project works with an emphasis on the portions you contributed. This video should be done using screen share and voice over. This can easily be done using Zoom, although you don't have to use Zoom, it's just what we recommend. You can create a new meeting, start screen sharing, and start recording. This will create a video recording on your computer. This should then be uploaded to a publicly accessible site, such as YouTube, Dropbox, or Google Drive. MAKE SURE THE LINK YOU SHARE IS PUBLIC or UNLISTED. If it is not accessible by your grader, your project will be graded based on what they can access. The link should be pasted in the submission text box after the GitHub repo link. REQUIRED: PUBLIC link to video, and GitHub repo link with everything listed above!

Screenshots of Code:

S



PROMINEO TECH

```
/**
 * Loops through each player's hand 26 times, calculates the score, and outputs
 * the running totals.
 *
 * @param {arr} player1Stack All of player 1's cards
 * @param {arr} player2Stack All of player 2's cards
 */
function playGame(player1Stack, player2Stack) {
  let turns = 0;
  let winner = "";

  while (turns < 26) {
    console.log("\n");

    console.log(`--- Round ${turns + 1}! ---`);

    console.log(`Player 1 card: ${player1Stack[turns]} | Player 2 card: ${player2Stack[turns]}`);
    calculateScore(player1Stack[turns], player2Stack[turns]);
    console.log(`Current Score: ${player1Score} to ${player2Score}`);

    turns++;
  }
  declareWinner();
}

/**
 * Declares the winner based on player scores.
 */
function declareWinner() {
  console.log("------------------------------");
  console.log(`FINAL SCORE: ${player1Score} to ${player2Score}`);
  if (player1Score > player2Score) {
    winner = "Player 1";
  } else if (player1Score < player2Score) {
    winner = "Player 2";
  } else {
    winner = "Tie! Everybody";
  }
  console.log(`${winner} is the winner!!!`);
}

```

index_test.js > describe("MyFunctions") callback

```
const expect = chai.expect;

describe("MyFunctions", function () {
  const unshuffledDeck = [
    1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 1
    11, 12, 13, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
  ];
  describe("#buildDeck", function () {
    it("should create a deck of cards with 1-13 repeating 4 times", function () {
      expect(buildDeck()).to.eql(unshuffledDeck);
    });
  });

  // Test the shuffle function. Note need a new array to test rather than point
  // to the original array which mutates with the shuffle.
  describe("#shuffle", function () {
    const newDeck = [
      1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
      10, 11, 12, 13, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
    ];
    it("should return an array not equal to unshuffledDeck", function () {
      expect(shuffle(newDeck)).to.not.eql(unshuffledDeck);
    });
  });
});

```

Screenshots of Running Application:

PROMINEO TECH