Intro to JavaScript Week 6 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

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:

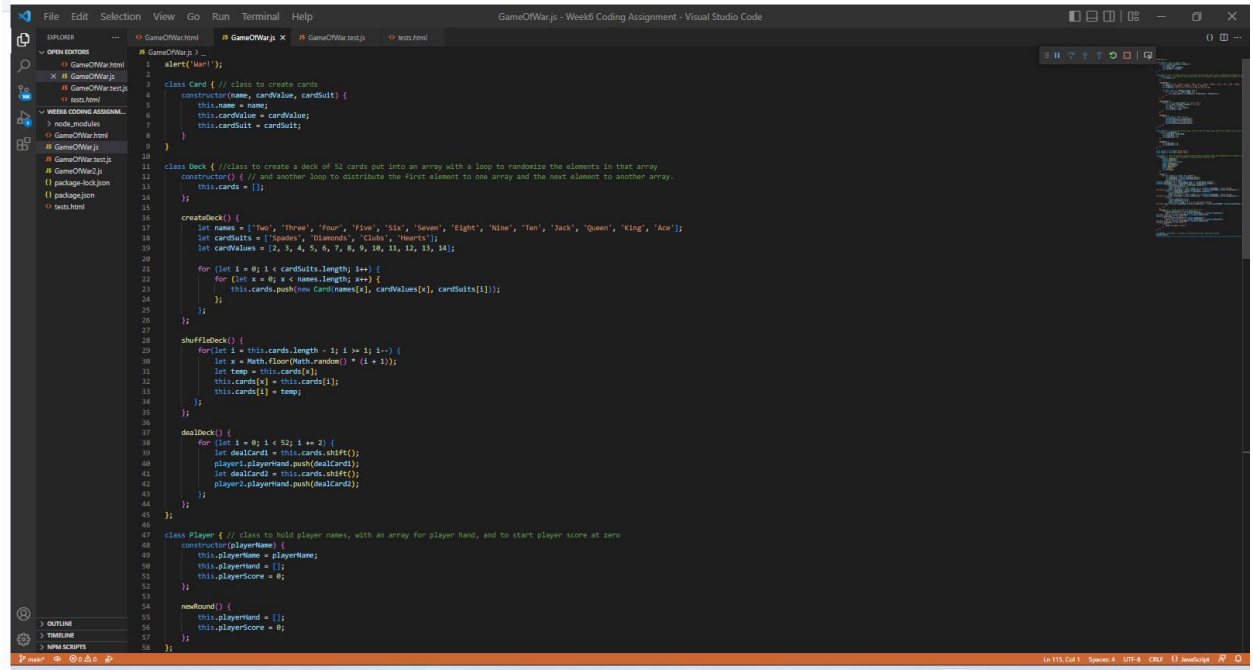
For the final project you will be creating an automated version of the classic card game *WAR*.

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 ran, do the following:

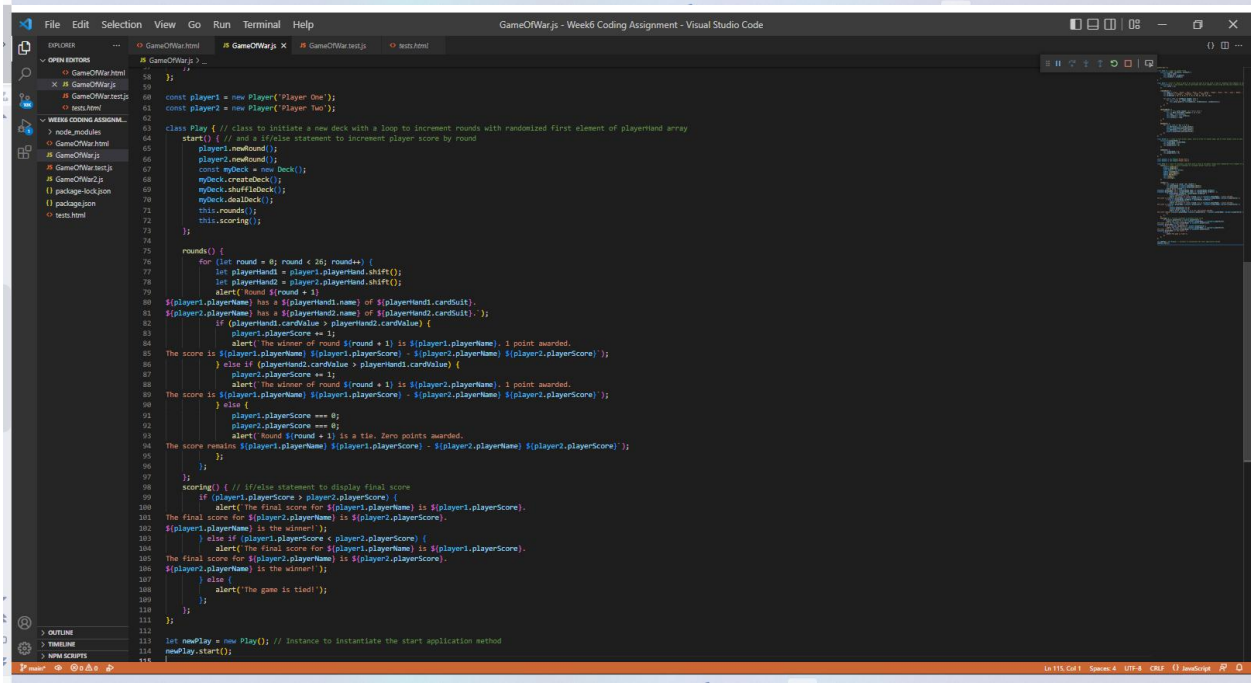
- Deal 26 Cards to two Players from a Deck.
- Iterate through the turns where each Player plays a Card
- The Player who played the higher card is awarded a point
    - Ties result in zero points for either Player
- After all cards have been played, display the score.

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

Screenshots of Code:

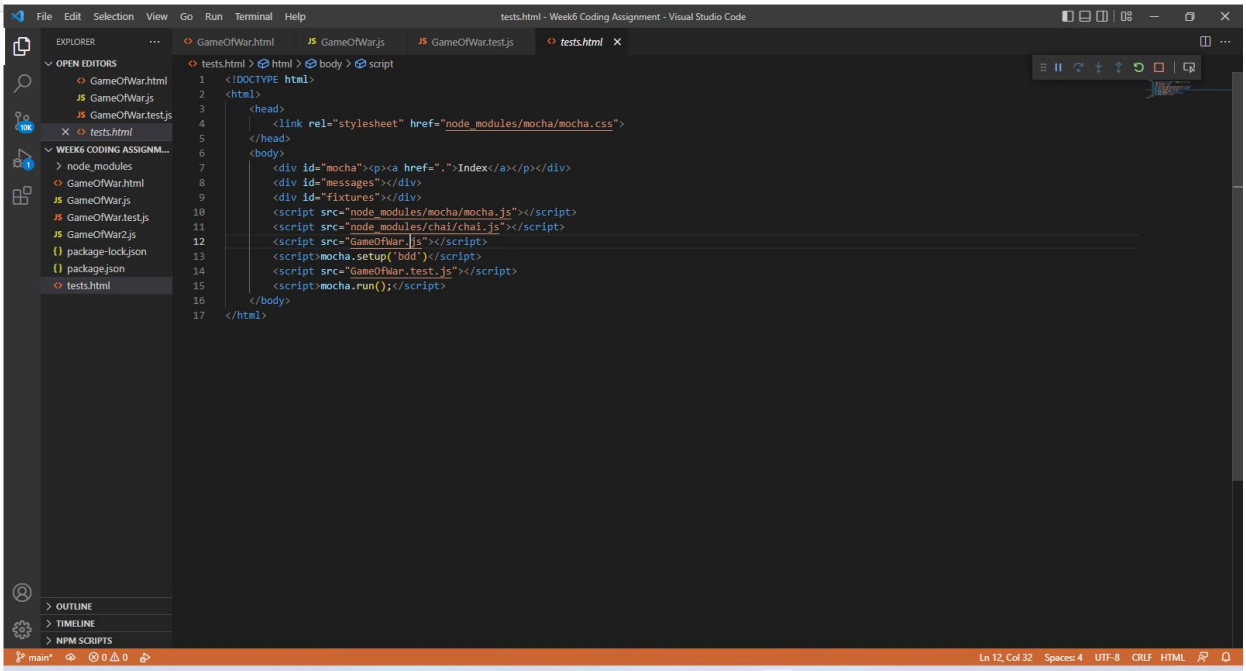


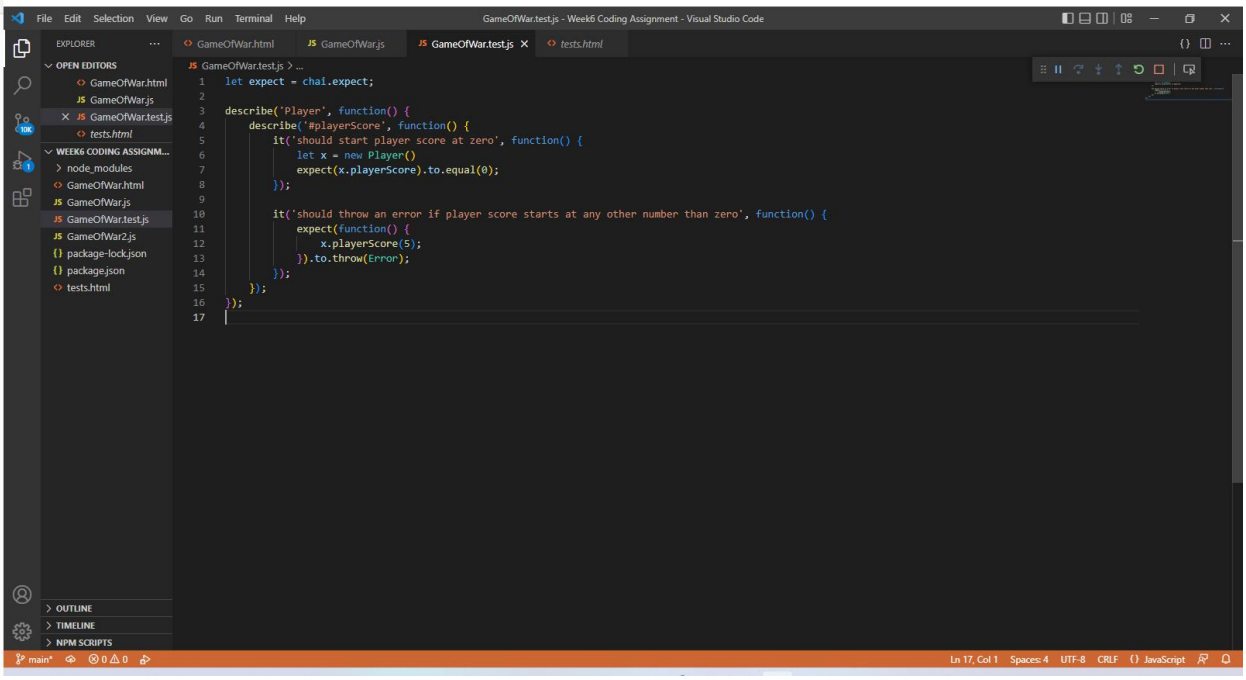
```
alert('War!');

class Card { // class to create cards
    constructor(name, cardValue, cardSuit) {
        this.name = name;
        this.cardValue = cardValue;
        this.cardSuit = cardSuit;
    }
}

class Deck { //class to create a deck of 52 cards put into an array with a loop to randomize the elements in that array
    constructor() { // and another loop to distribute the first element to one array and the next element to another array.
        this.cards = [];
    };

    createDeck() {
        let names = ['Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Jack', 'Queen', 'King', 'Ace'];
        let cardSuits = ['Spades', 'Diamonds', 'Clubs', 'Hearts'];
        let cardValues = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14];

        for (let i = 0; i < cardSuits.length; i++) {
            for (let x = 0; x < names.length; x++) {
                this.cards.push(new Card(names[x], cardValues[x], cardSuits[i]));
            };
        };
    };

    shuffleDeck() {
        for(let i = this.cards.length - 1; i >= 1; i--) {
            let x = Math.floor(Math.random() * (i + 1));
            let temp = this.cards[x];
            this.cards[x] = this.cards[i];
            this.cards[i] = temp;
        };
    };

    dealDeck() {
        for (let i = 0; i < 52; i += 2) {
            let dealCard1 = this.cards.shift();
            player1.playerHand.push(dealCard1);
            let dealCard2 = this.cards.shift();
            player2.playerHand.push(dealCard2);
        };
    };
};

class Player { // class to hold player names, with an array for player hand, and to start player score at zero
    constructor(playerName) {
        this.playerName = playerName;
        this.playerHand = [];
        this.playerScore = 0;
    };

    newRound() {
        this.playerHand = [];
        this.playerScore = 0;
    };
};
```



```
};

const player1 = new Player('Player One');
const player2 = new Player('Player Two');

class Play { // class to initiate a new deck with a loop to increment rounds with randomized first element of playerHand array
    start() { // and a if/else statement to increment player score by round
        player1.newRound();
        player2.newRound();
        const myDeck = new Deck();
        myDeck.createDeck();
        myDeck.shuffleDeck();
        myDeck.dealDeck();
        this.rounds();
        this.scoring();
    };

    rounds() {
        for (let round = 0; round < 26; round++) {
            let playerHand1 = player1.playerHand.shift();
            let playerHand2 = player2.playerHand.shift();
            alert(`Round ${round + 1}
${player1.playerName} has a ${playerHand1.name} of ${playerHand1.cardSuit}.
${player2.playerName} has a ${playerHand2.name} of ${playerHand2.cardSuit}.`);
            if (playerHand1.cardValue > playerHand2.cardValue) {
                player1.playerScore += 1;
                alert(`The winner of round ${round + 1) is ${player1.playerName}. 1 point awarded.
The score is ${player1.playerName} ${player1.playerScore} - ${player2.playerName} ${player2.playerScore}`);
            } else if (playerHand2.cardValue > playerHand1.cardValue) {
                player2.playerScore += 1;
                alert(`The winner of round ${round + 1) is ${player2.playerName}. 1 point awarded.
The score is ${player1.playerName} ${player1.playerScore} - ${player2.playerName} ${player2.playerScore}`);
            } else {
                player1.playerScore === 0;
                player2.playerScore === 0;
                alert(`Round ${round + 1) is a tie. Zero points awarded.
The score remains ${player1.playerName} ${player1.playerScore} - ${player2.playerName} ${player2.playerScore}`);
            };
        };
    };
    scoring() { // if/else statement to display final score
        if (player1.playerScore > player2.playerScore) {
            alert(`The final score for ${player1.playerName} is ${player1.playerScore}.
The final score for ${player2.playerName} is ${player2.playerScore}.
${player1.playerName} is the winner!`);
        } else if (player1.playerScore < player2.playerScore) {
            alert(`The final score for ${player1.playerName} is ${player1.playerScore}.
The final score for ${player2.playerName} is ${player2.playerScore}.
${player2.playerName} is the winner!`);
        } else {
            alert('The game is tied!');
        };
    };
};

let newPlay = new Play(); // Instance to instantiate the start application method
newPlay.start();
```

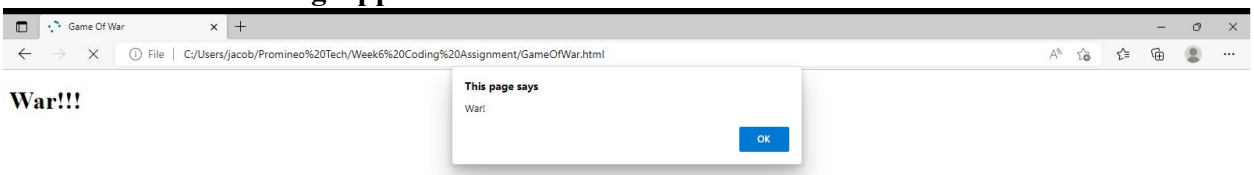
PROMINEO TECH

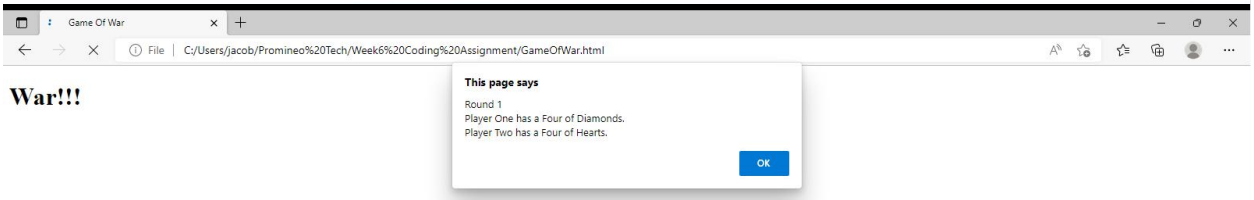




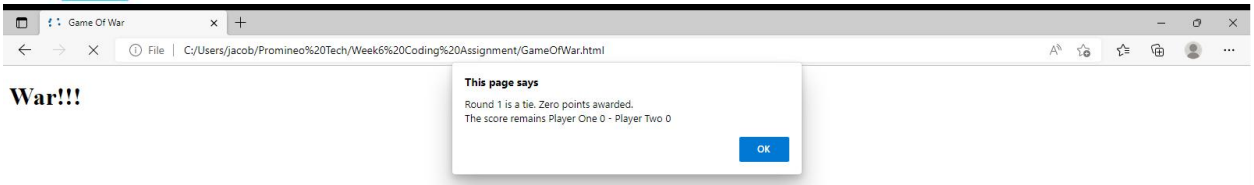
PROMINEO TECH

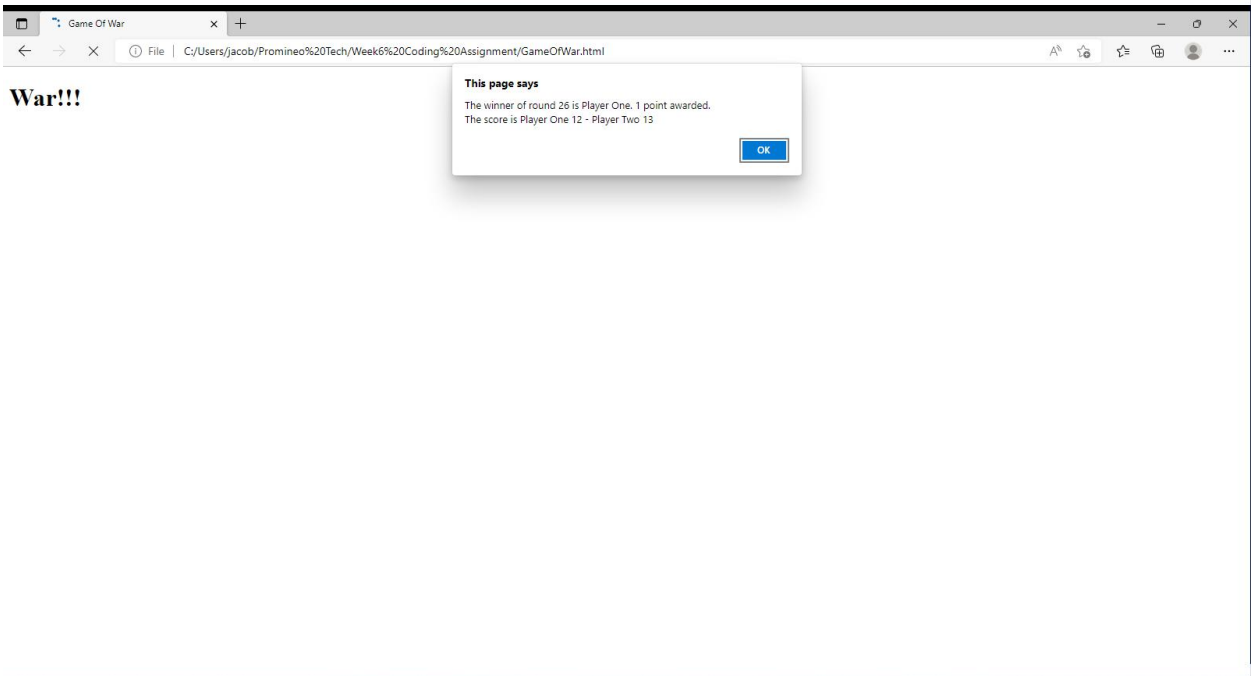
Screenshots of Running Application:

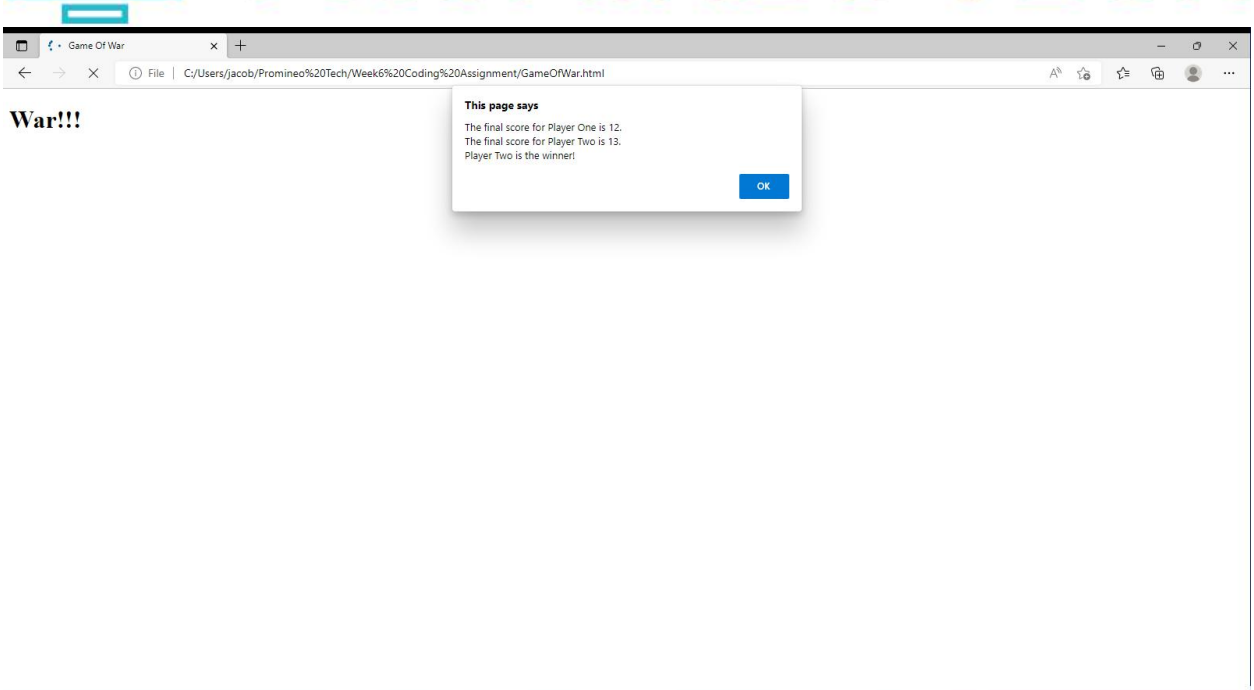


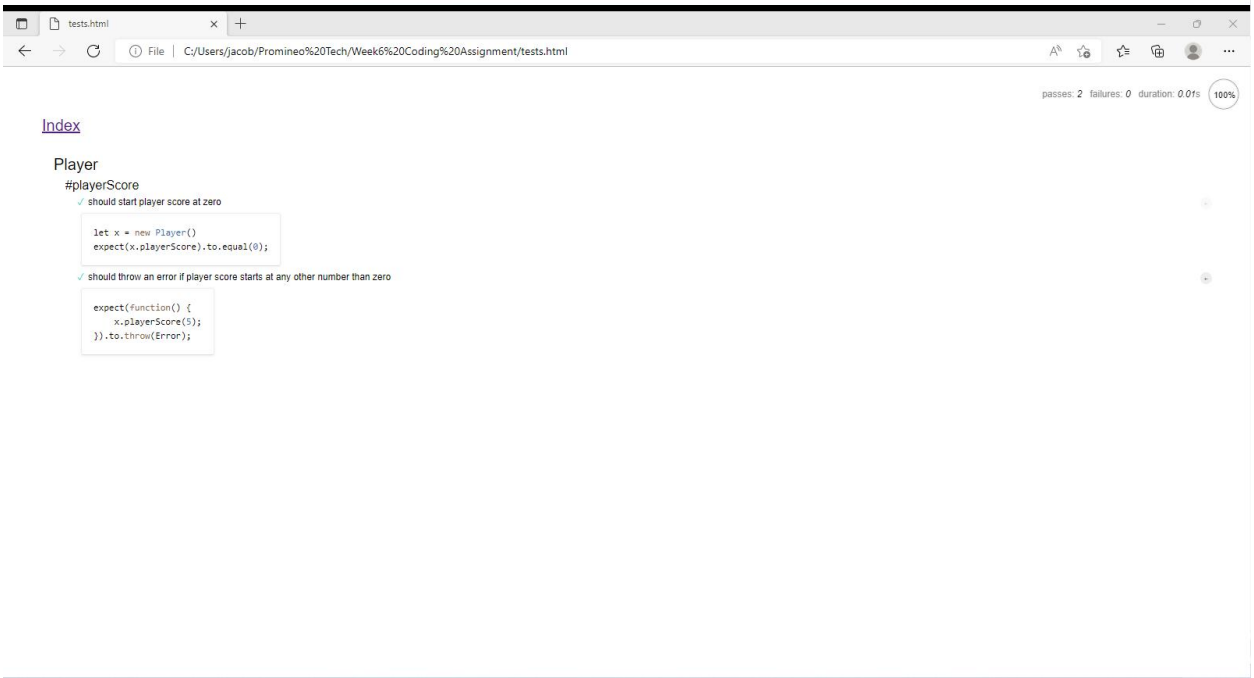



PROMINEO TECH


Game Of War
File | C:/Users/jacob/Promineo%20Tech/Week6%20Coding%20Assignment/GameOfWar.html
War!!!
This page says
Round 1 is a tie. Zero points awarded.
The score remains Player One 0 - Player Two 0
OK


Game Of War
File | C:/Users/jacob/Promineo%20Tech/Week6%20Coding%20Assignment/GameOfWar.html
War!!!
This page says
The winner of round 26 is Player One. 1 point awarded.
The score is Player One 12 - Player Two 13
OK





URL to GitHub Repository: https://github.com/JacobStuder/Promineo-Tech-Week-6-Coding-Assignment.git