

Seat Reservation Program



Part 5: Reserving the Seats

Setting up the Final Steps...

When this form is submitted a function called `processReservation()` runs.

Write the click handler and the stub for this function.

cancel and choose different seats'. The background of the page shows a grid of 180 seats arranged in 12 rows (A-L) and 15 columns. Seats 97, 98, and 99 are highlighted in pink. The grid is labeled with letters A-L on the left and right, and numbers 1-180 in the center."/>

Form Event Handler

Did you get something like this?

```
document.getElementById('confirmres').addEventListener('submit', function(event){  
    processReservation();  
    event.preventDefault();  
} );  
  
function processReservation(){  
    // process the reservation  
}
```

Setting Variables

```
function processReservation(){  
    const hardCodeRecords = Object.keys(reservedSeats).length;  
    const fname = document.getElementById('fname').value;  
    const lname = document.getElementById('lname').value;  
    let counter = 1;  
    let nextRecord = '';  
}
```

To start with, in this function, you need to know how many records are already in the reservedSeats object (or have been reserved and are in the database).

Then you need to get the first and last names of the person who reserved them, from the form.

Back to the Object

`Object.keys(reservedSeats).length;`

will tell you how many records there are already.

Next, inside the `processReservation` function, set up the `forEach` method to loop through the `selectedSeats` array.

```
var reservedSeats = {  
  record1:{  
    seat:"b19",  
    owner:{  
      fname:"Joe",  
      lname:"Smith"  
    }  
  },  
  record2:{  
    seat:"b20",  
    owner:{  
      fname:"Joe",  
      lname:"Smith"  
    }  
  },  
  record3:{  
    seat:"b21",  
    owner:{  
      fname:"Joe",  
      lname:"Smith"  
    }  
  },  
  record4:{  
    seat:"b22",  
    owner:{  
      fname:"Joe",  
      lname:"Smith"  
    }  
  }  
};
```

The ForEach Method...

Did you get something like this?

Can you add the code needed to set the class name to "r" and change the html for the selected seats to "R"?

```
function processReservation(){
  const hardCodeRecords = Object.keys(reservedSeats).length;
  const fname = document.getElementById('fname').value;
  const lname = document.getElementById('lname').value;
  let counter = 1;
  let nextRecord = '';

  selectedSeats.forEach( function(thisSeat){

    // set the class name to "r"
    // set the innerHTML for that seat <div> to "R"
    // add each record to the reservedSeats object

  } );
}
```

Setting the Class and HTML

Hopefully you got something like this.

```
selectedSeats.forEach( function(thisSeat){  
  
    document.getElementById(thisSeat).className = "r";  
    document.getElementById(thisSeat).innerHTML = "R";  
    // add each record to the reservedSeats object  
  
} );
```

Adding Each Record

`nextRecord` should be `record5` for the first time through the array. This is the number of hardcoded records plus 1.

```
selectedSeats.forEach( function(thisSeat){  
  
    document.getElementById(thisSeat).className = "r";  
    document.getElementById(thisSeat).innerHTML = "R";  
  
    nextRecord = `record${hardCodeRecords + counter}`;  
    reservedSeats[nextRecord] = {  
        seat:thisSeat,  
        owner:{  
            fname:fname,  
            lname:lname  
        }  
    };  
  
    counter++;  
  
} );
```


Clean Up...

After going through the selectedSeats array and adding the records to the reservedSeats object, you need to close the form, Empty out the selectedSeats array, and run the manageConfirmForm function again to reset its original state.

```
selectedSeats.forEach( function(thisSeat){

    document.getElementById(thisSeat).className = "r";
    document.getElementById(thisSeat).innerHTML = "R";

    nextRecord = `record${hardCodeRecords + counter}`;
    reservedSeats[nextRecord] = {
        seat:thisSeat,
        owner:{
            fname:fname,
            lname:lname
        }
    };
    counter++;
} );

// clean up...
document.getElementById('resform').style.display="none";
selectedSeats = [];
manageConfirmForm();
//You can see the resulting object in the console...
console.log(reservedSeats);
```

Entire Function

This is the entire function that processes the actual reservation.

```
function processReservation(){
  const hardCodeRecords = Object.keys(reservedSeats).length;
  const fname = document.getElementById('fname').value;
  const lname = document.getElementById('lname').value;
  let counter = 1;
  let nextRecord = '';

  selectedSeats.forEach( function(thisSeat){

    document.getElementById(thisSeat).className = "r";
    document.getElementById(thisSeat).innerHTML = "R";

    nextRecord = `record${hardCodeRecords + counter}`;
    reservedSeats[nextRecord] = {
      seat:thisSeat,
      owner:{
        fname:fname,
        lname:lname
      }
    };
    counter++;
  } );

  // clean up...
  document.getElementById('resform').style.display="none";
  selectedSeats = [];
  manageConfirmForm();
  //You can see the resulting object in the console...
  console.log(reservedSeats);
}
```

One last issue...

Once you have reserved some seats, you can actually select them again, even though they have turned red and have the 'R' inside them.

This is because when the page was loaded, these elements had the event listener added to them, and even though the class and HTML for these two elements has changed, the event listener is still there.

Let's Fix that!

A	1	2	3	4	5	6	7
B	16	17	18	R	R	R	R
C	31	32	33	34	35	36	37
D	46	47	48	49	50	51	52
E	61	62	63	64	65	66	67
F	76	R	R	79	80	81	82

The Fix:

Go back to the seatSelectionProcess function and add this if statement around the whole thing.

That will only add items to the array that DO NOT have the class “r” on them (note the !).

```
function seatSelectionProcess(thisSeat){  
    if(!document.getElementById(thisSeat).classList.contains('r')){  
        const index = selectedSeats.indexOf(thisSeat);  
  
        if( index > -1 ){  
            selectedSeats.splice(index, 1);  
            document.getElementById(thisSeat).className = "a";  
        }  
  
        else {  
            selectedSeats.push(thisSeat);  
            document.getElementById(thisSeat).className = "s";  
        }  
  
        manageConfirmForm();  
        console.log(selectedSeats);  
    }  
}
```

Summary

This project allowed you build a small application, with a visual interface that keeps track of user data while they are using it.

