

NAS TRAVELS

BUS TICKET BOOKING WEBSITE

Introduction:

The aim of this project is to design and develop a bus ticket reservation website using HTML, CSS, and JavaScript. This website will allow users to book bus tickets online, choose their preferred seats, and make payments. The website will also provide users with information on bus schedules, routes, and fares.

Features:

1. Home page: The home page will be the first page that users will see when they visit the website. It will contain information about the company and booking page.
2. Booking page: Booking page contains from , to ,date of journey and number of seats.
3. Seat selection: Users will be able to select their preferred seats on the bus. The website will show them a graphical representation of the bus layout, and they will be able to select seats by clicking on them.
4. Payment gateway: The website will have a secure payment gateway that will allow users to make payments using their net banking or G pay directly by clicking book and pay button without redirecting to other page.
5. Available class: First class and economy class are available for the users.
Users can select their preferred class.
6. Contact page: The website will have a contact page that will allow users to get in touch with the company in case of any queries or issues.

Technologies Used:

The bus ticket reservation website will be developed using HTML, CSS, and JavaScript. HTML will be used to structure the content of the website, while CSS will be used to style it. JavaScript will be used to add interactivity to the website, such as seat selection and payment process.

Conclusion:

The bus ticket reservation website will provide users with a convenient and easy way to book bus tickets online. It will be user-friendly, secure, and provide users with all the necessary information they need to make informed decisions. The use of HTML, CSS, and JavaScript will ensure that the website is responsive and works seamlessly across different devices.

[ai.html](#)

```
<!DOCTYPE html>

<html>

<head>

    <title>NAS TRAVELS</title>

    <title>Bus Ticket Booking</title>

    <link rel="stylesheet" href="style1.css">

</head>

<body><header>

    <h1>NAS TRAVELS<br>Bus Ticket Booking</h1>

    <nav>

    <ul>

        <li><a href="ai.html">Home</a></li>
```

```
        <li><a href="#About Us">About Us</a></li>
    </ul></nav>
</header>
```

```

    <form>
        <label for="from">From:</label>
        <input type="text" id="from" name="from">

        <label for="to">To:</label>
        <input type="text" id="to" name="to">

        <label for="date">Date:</label>
        <input type="date" id="date" name="date">

        <label for="seats">Number of seats:</label>
        <input type="number" id="seats" name="seats">

        <button type="button" onclick="bookTickets()">
            <a
href="C:\Users\cherian\Desktop\bus\index.html">bookTickets</a></button>
    </form>
```

```

    <div id="result"></div>
    <script src="script.js"></script><body>
```

```

    <h2>About Us</h2>
```

<h3 id="About Us">Our bus ticket booking website is committed to providing you with easy and hassle-free bus ticket bookings. We strive to offer a one-stop solution for all your bus travel needs, covering all major routes across the country. Our user-friendly platform allows you to search, compare, and book tickets within minutes. Our extensive network of bus operators ensures that you have access to a wide range of options that suit your preferences and budget. We take pride in our commitment to customer safety and comfort. All our bus operators are handpicked, and we ensure that they comply with all safety regulations. You can rest assured that you will experience a comfortable and safe journey with us. We also provide various payment options to make the booking process more convenient for you. You can pay easily using credit/debit cards, net banking, or e-wallets. Our dedicated customer support team is available 24/7 to assist you with any queries or issues you may have. In addition to booking bus tickets, we also offer various deals and discounts, making your travel experience even more affordable. We aim to provide you with a seamless and enjoyable journey, from start to finish. Choose us for your next bus journey and experience the convenience and comfort of hassle-free booking of bus tickets!

</h3>

<footer>

<p>©2023 Bus Booking website.All Rights Reserved</p>

<p>contact us</p>

<p>@9787124534</p>

<p>nas@gmail.com</p>

</footer>

</body>

</html>

[Style1.css](#)

h1 {

text-align: center;

padding-left: 500px;

```
}
```

```
body {
```

```
    font-family: cursive;
```

```
    max-height: 100%;
```

```
    overflow: auto;
```

```
    background-color: #f6f7f8;
```

```
    background-image: url("C:/Users/cherian/Desktop/bus/try.jpg");
```

```
    background-repeat: no-repeat;
```

```
    background-size: cover;
```

```
    background-position:center;
```

```
    color: hsl(0, 5%, 7%);
```

```
    overflow-x: hidden;
```

```
}
```

```
header{
```

```
    background-color: #f6f7f8;
```

```
    color: black;
```

```
    display:flex;
```

```
    justify-content:space-between;
```

```
    align-items:center;
```

```
    padding: 1em;
```

```
}
```

```
nav li{
```

```
    display: inline-block;
```

```
    margin-right: 1em;
```

```
}
```

```
nav a{
    color: black;
    text-decoration: none;
}
main{
    max-width:960px;
    margin:0 auto;
    padding: 1em;
}
h2{
    color:white;
}
h3{
    background-color: aliceblue;
}
footer{
    background-color: #1e32a5;
    color: antiquewhite;
    text-align: center;
    padding:1em;
}
```

```
form {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-top: 20px;
```

```
}
```

```
label, input {  
    margin: 10px 0;  
}
```

```
button {  
    margin-top: 20px;  
    padding: 10px 20px;  
    background-color: #4CAF50;  
    color: rgb(29, 24, 24);  
    border: none;  
    border-radius: 5px;  
    cursor: pointer;  
}
```

```
#result {  
    margin-top: 20px;  
    text-align: center;  
    font-weight: bold;  
    font-size: 20px;  
}
```

[Script.js](#)

```
function bookTickets() {  
    var from = document.getElementById("from").value;  
    var to = document.getElementById("to").value;
```

```
    var date = document.getElementById("date").value;
    var seats = document.getElementById("seats").value;

    if (from == "" || to == "" || date == "" || seats == "") {
        alert("Please fill in all fields.");
        return;
    }

    // Perform booking logic here...
}
```

[Index.html](#)

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
    <title>Simple Bus Seat Reservation Site</title>
```

```
    <link href='http://fonts.googleapis.com/css?family=Lato:400,700' rel='stylesheet'
type='text/css'>
```

```
    <link rel="stylesheet" type="text/css" href="css/jquery.seat-charts.css">
```

```
    <link rel="stylesheet" type="text/css" href="css/style.css">
```

```
    <script src="https://code.jquery.com/jquery-3.6.0.min.js" integrity="sha256-
/xUj+3OJU5yExlq6GSYGSHk7tPXikynS7ogEvDej/m4=" crossorigin="anonymous"></script>
```



```

    <script src="js/jquery.seat-charts.js"></script>
</head>

<body>
    <div class="wrapper">
        <div class="container">
            <div class="row">
                <div class="grid-50">
                    <div id="seat-map">
                        <div class="front-indicator">Bus Seat Reservation</div>
                        <h4 class="text-muted fw-bold text-center" style="padding-left:3em;
margin:.5em">From Front Row</h4>
                        <div id="bus-seat-map"></div>
                        <h4 class="text-muted fw-bold text-center" style="padding-left:3em;
margin:.5em">End of Seat Row</h4>
                    </div>
                </div>
                <div class="grid-50">
                    <div class="booking-details">

                        <form action="" method="post">

s
                            <h2>Booking Details</h2>

                            <h3> Selected Seats (<span id="counter">0</span>):</h3>
                            <ul id="selected-seats"></ul>

                            <h2>Total: <b>${<span id="total"></span></b></h2>

```

```
<button type="button" id="checkout-button"> Book and pay</button>
```

```
</form>
```

```
</div>
```

```
<div id="legend"></div>
```

```
<button id="reset-btn" type="button">Reset Bus Seat</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<script src="./js/script1.js"></script>
```

```
</body>
```

```
</html>
```

[Style.css](#)

```
html,
```

```
body {
```

```
    height: 100%;
```

```
    ;
```

```
    width: 100%;
```

```
;
}
```

```
body {
  font-family: cursive;
  max-height: 100%;
  overflow: auto;
  background-color: #ebf3f3;
  background-image: url("C:/Users/cherian/Desktop/bus/try.jpg");
  background-repeat: no-repeat;
  background-size: cover;
  background-position:center;
  color: hsl(0, 0%, 6%);
  overflow-x: hidden;
}
```

```
a {
  color: #b71a4c;
}
```

```
.front-indicator {
  margin: 0.5em 0;
  width: 100%;
  background-color: #f6f6f6;
  background-image: none;
  color: #ffffff;
  text-shadow: 2px 2px 5px #000;
```

```
font-size: 21px;
font-weight: 700;
padding: 1em 0;
text-align: center;
border-radius: 3em;
background-color: #0093E9;
background-image: linear-gradient(160deg, #0093E9 0%, #80D0C7 100%);
}
```

```
.wrapper {
  width: 100%;
  text-align: center;
}
```

```
.container {
  margin: 0 auto;
  width: 80%;
  text-align: left;
}
```

```
.booking-details {
  text-align: left;
  font-size: 12px;
  padding: 4.5em;
  margin-top: 4em;
}
```

```
.booking-details h2 {  
    margin: 25px 0 20px 0;  
    font-size: 17px;  
}
```

```
.booking-details h3 {  
    margin: 5px 5px 0 0;  
    font-size: 14px;  
}
```

```
div.seatCharts-cell {  
    color: #1a1c1e;  
    height: 4em;  
    width: 14.666666%;  
    line-height: .5em;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    font-size: 17px;  
}
```

```
div.seatCharts-seat {  
    color: #FFFFFF;  
    cursor: pointer;  
}
```

```
div.seatCharts-row {
```

```
height: 5em;
}
```

```
div.seatCharts-seat.available {
    background: rgb(103, 122, 136);
    background: radial-gradient(circle, #2a05ad 0%, rgb(76, 74, 187) 25%, rgba(76, 74, 187) 50%,
    rgba(46, 135, 196, 1) 75%, rgba(40, 121, 177, 1) 100%);
}
```

```
div.seatCharts-seat.available.first-class {
    background: rgb(178, 64, 212);
    background: radial-gradient(circle, rgb(247, 105, 176) 0%,rgb(247, 105, 176)) 25%, rgba(118,
    12, 149, 1) 50%, rgba(126, 17, 149, 1) 75%, rgba(94, 2, 121, 1) 100%;
}
```

```
div.seatCharts-seat.focused {
    background: rgb(219, 215, 215);
    background: radial-gradient(circle, rgba(219, 215, 215, 1) 0%, rgba(205, 205, 205, 1) 25%,
    rgba(195, 192, 192, 1) 50%, rgba(204, 204, 204, 1) 75%, rgba(207, 207, 207, 1) 100%);
    color: #1a1c1e
}
```

```
div.seatCharts-seat.selected {
    background-color: hwb(123 31% 35%);
    background-image: linear-gradient(45deg, hwb(123 31% 35%) 0%, hwb(123 31% 35%) 100%);
    text-shadow: 2px 2px 5px #000;
}
```

```
div.seatCharts-seat.unavailable {  
    background: rgb(252, 121, 121);  
    background: radial-gradient(circle, rgba(252, 121, 121, 1) 0%, rgba(249, 81, 81, 1) 25%,  
    rgba(231, 53, 53, 1) 50%, rgba(255, 24, 24, 1) 75%, rgba(217, 6, 6, 1) 100% );  
}
```

```
div.seatCharts-container {  
    border-right: 1px dotted #adadad;  
    padding: 20px;  
}
```

```
div.seatCharts-legend {  
    padding-left: 0px;  
    bottom: 16px;  
}
```

```
ul.seatCharts-legendList {  
    padding-left: 0px;  
}
```

```
span.seatCharts-legendDescription {  
    margin-left: 5px;  
    line-height: 30px;  
}
```

```
.checkout-button {  
    display: block;
```

```
margin: 10px 0;
font-size: 14px;
}
```

```
#selected-seats {
  min-height: 10em;
  overflow-y: scroll;
  background: #fff;
  padding: 0.5em 1em;
  overflow: auto;
}
```

```
ul#selected-seats:empty:after {
  content: "Please select Seat/s first";
  color: #6c6c6c;
  font-style: italic;
}
```

```
ul#selected-seats>li {
  color: #000;
  list-style: none;
}
```

```
.text-muted {
  color: rgb(241, 239, 239);
  text-shadow: 2px 2px 5px #000;
}
```



```
.text-center {  
  text-align: center;  
  align-items: center;  
  justify-content: center;  
}
```

```
.fw-bold {  
  font-weight: bold;  
}
```

```
.fw-bolder {  
  font-weight: bolder;  
}
```

```
.row {  
  --bs-gutter-x: 0.5rem;  
  --bs-gutter-y: 0.5rem;  
  display: flex;  
  flex-wrap: wrap;  
  margin-top: calc(-1 * var(--bs-gutter-y));  
  margin-right: calc(-.5 * var(--bs-gutter-x));  
  margin-left: calc(-.5 * var(--bs-gutter-x));  
}
```

```
.grid-50 {  
  flex: 0 0 auto;  
  width: 50%;  
}
```

```
}
```

```
div.seatCharts-space {  
    background-color: transparent;  
    color: #fff !important;  
}
```

```
li.seatCharts-legendItem {  
    margin-top: 10px;  
    font-size: 15px;  
    line-height: 5em;  
}
```

```
#checkout-button {  
    margin: 10px;  
    padding: 15px 45px;  
    text-align: center;  
    text-transform: uppercase;  
    transition: .5s;  
    background-size: 200% auto;  
    color: hsl(0, 11%, 95%);  
    border-radius: 10px;  
    display: block;  
    background-image: linear-gradient(to right, #77A1D3 0%, #79CBCA 51%, #77A1D3 100%);  
    border: none;  
    cursor: pointer;  
}
```

```
#checkout-button:hover,
```

```
#reset-btn:hover {
```

```
background-position: right center;
color: #fff;
text-decoration: none;
}
```

```
#reset-btn {
width: 100%;
margin: 10px;
padding: 15px 45px;
text-align: center;
text-transform: uppercase;
transition: .5s;
background-size: 200% auto; div.seatCharts-container {
/*min-width: 700px;*/
}
```

```
div.seatCharts-cell {

height: 16px;
width: 16px;
margin: 3px;
float: left;
text-align: center;
outline: none;
font-size: 13px;
line-height: 16px;
color: blue;
```

```
}
```

[jquery.seat-charts.css](#)

```
div.seatCharts-seat {  
    background-color: green;  
    color: white;  
    -webkit-border-radius: 5px;  
    -moz-border-radius: 5px;  
    border-radius: 5px;  
    cursor: default;  
}
```

```
div.seatCharts-seat:focus {  
    border: none;  
}
```

```
/*
```

```
.seatCharts-seat:focus {  
    outline: none;  
}
```

```
*/
```

```
div.seatCharts-space {  
    background-color: white;  
}
```

```
div.seatCharts-row {  
    height: 50px;  
}
```

```
div.seatCharts-row:after {  
    clear: both;  
}
```

```
div.seatCharts-seat.selected {  
    background-color: aqua;  
}
```

```
div.seatCharts-seat.focused {  
    background-color: #6db131;  
}
```

```
div.seatCharts-seat.available {  
    background-color: green;  
}
```

```
div.seatCharts-seat.unavailable {  
    background-color: red;  
    cursor: not-allowed;  
}
```

```
ul.seatCharts-legendList {  
    list-style: none;  
}
```

```
li.seatCharts-legendItem {  
    margin-top: 10px;  
    line-height: 2;
```

```

}

color: #fff;

border-radius: 5em;

display: block;

border: none;

cursor: pointer;

background-image: linear-gradient(to right, #06beb6 0%, #48b1bf 51%, #06beb6 100%);
}

```

[Script 1.js](#)

```

var firstSeatLabel = 1;

var booked = !!localStorage.getItem('booked') ? $.parseJSON(localStorage.getItem('booked')) :
[];

$(document).ready(function() {

    var $cart = $('#selected-seats'),

        $counter = $('#counter'),

        $total = $('#total'),

        sc = $('#bus-seat-map').seatCharts({

            map: [

                'ff_ff',

                'ff_ff',

                'ee_ee',

                'ee_ee',

                'ee____',

                'ee_ee',

                'ee_ee',

                'ee_ee',

                'ee_ee',

                'eeeeee',

```

```
],
seats: {
  f: {
    price: 75,
    classes: 'first-class', //your custom CSS class
    category: 'First Class'
  },
  e: {
    price: 50,
    classes: 'economy-class', //your custom CSS class
    category: 'Economy Class'
  }
},
naming: {
  top: false,
  getLabel: function(character, row, column) {
    return firstSeatLabel++;
  },
},
legend: {
  node: $('#legend'),
  items: [
    ['f', 'available', 'First Class'],
    ['e', 'available', 'Economy Class'],
    ['f', 'unavailable', 'Already Booked']
  ]
}
```

```

    },
    click: function() {
        if (this.status() == 'available') {

            //let's create a new <li> which we'll add to the cart items

            $('<li>' + this.data().category + ' Seat # ' + this.settings.label + ': <b>$' +
this.data().price + '</b> <a href="#" class="cancel-cart-item">[cancel]</a></li>')

                .attr('id', 'cart-item-' + this.settings.id)

                .data('seatId', this.settings.id)

                .appendTo($cart);

            /*
            * Lets update the counter and total
            *
            * .find function will not find the current seat, because it will change its status only
after return
            * 'selected'. This is why we have to add 1 to the length and the current seat price to
the total.
            */

            $counter.text(sc.find('selected').length + 1);

            $total.text(recalculateTotal(sc) + this.data().price);

            return 'selected';

        } else if (this.status() == 'selected') {

            //update the counter

            $counter.text(sc.find('selected').length - 1);

```



```

        //and total
        $total.text(recalculateTotal(sc) - this.data().price);

        //remove the item from our cart
        $('#cart-item-' + this.settings.id).remove();

        //seat has been vacated
        return 'available';

    } else if (this.status() == 'unavailable') {
        //seat has been already booked
        return 'unavailable';
    } else {
        return this.style();
    }
}
});

```

```

//this will handle "[cancel]" link clicks
$('#selected-seats').on('click', '.cancel-cart-item', function() {
    //let's just trigger Click event on the appropriate seat, so we don't have to repeat the logic
    here
    sc.get($(this).parents('li:first').data('seatId')).click();
});

```

```

//let's pretend some seats have already been booked
// sc.get(['1_2', '4_1', '7_1', '7_2']).status('unavailable');

```

```

    sc.get(booked).status('unavailable');

});

function recalculateTotal(sc) {
    var total = 0;

    //basically find every selected seat and sum its price
    sc.find('selected').each(function() {

        total += this.data().price;

    });

    return total;
}

$(function() {
    $('#checkout-button').click(function() {
        var items = $('#selected-seats li')
        if (items.length <= 0) {
            alert("Please select atleast 1 seat first.")
            return false;
        }
        var selected = [];
        items.each(function(e) {
            var id = $(this).attr('id')

```

```

        id = id.replace("cart-item-", "")
        selected.push(id)
    })
    if (Object.keys(booked).length > 0) {
        Object.keys(booked).map(k => {
            selected.push(booked[k])
        })
    }
    localStorage.setItem('booked', JSON.stringify(selected))
    alert("tickets has been booked successfully.")
    location.reload()
})
$('#reset-btn').click(function() {
    if (confirm("are you sure to reset the reservation of the bus?") === true) {
        localStorage.removeItem('booked')
        alert("Seats has been Reset successfully.")
        location.reload()
    }
})
})

```

[Jquery-seats-charts.js](#)

```

(function($) {

```

```

    //'use strict';

```

```
$.fn.seatCharts = function (setup) {
```

```
    //if there's seatCharts object associated with the current element, return it
```

```
    if (this.data('seatCharts')) {
```

```
        return this.data('seatCharts');
```

```
    }
```

```
    var fn    = this,
```

```
        seats = {},
```

```
        seatIds = [],
```

```
        legend,
```

```
        settings = {
```

```
            animate : false, //requires jQuery UI
```

```
            naming : {
```

```
                top : true,
```

```
                left : true,
```

```
                getId : function(character, row, column) {
```

```
                    return row + '_' + column;
```

```
                },
```

```
                getLabel : function (character, row, column) {
```

```
                    return column;
```

```
                }
```

```
            },
```

```
            legend : {
```

```
                node : null,
```

```
                items : []
```

```

    },
    click : function() {

        if (this.status() == 'available') {
            return 'selected';
        } else if (this.status() == 'selected') {
            return 'available';
        } else {
            return this.style();
        }

    },
    focus : function() {

        if (this.status() == 'available') {
            return 'focused';
        } else {
            return this.style();
        }

    },
    blur : function() {
        return this.status();
    },
    seats : {}

```

```

},

```

//seat will be basically a seat object which we'll when generating the map

```

seat = (function(seatCharts, seatChartsSettings) {
    return function (setup) {
        var fn = this;

        fn.settings = $.extend({
            status : 'available', //available, unavailable,
selected
            style : 'available',
            //make sure there's an empty hash if user doesn't
pass anything
            data : seatChartsSettings.seats[setup.character]
|| {}

            //anything goes here?
        }, setup);

        fn.settings.$node = $('<div></div>');

        fn.settings.$node
            .attr({
                id      : fn.settings.id,
                role    : 'checkbox',
                'aria-checked' : false,
                focusable : true,
                tabIndex : -1 //manual focus
            })
            .text(fn.settings.label)
            .addClass(['seatCharts-seat', 'seatCharts-cell',
'available'].concat(

```

with standard JSC ones

```
//let's merge custom user defined classes
```

```
fn.settings.classes,  
typeof  
seatChartsSettings.seats[fn.settings.character] == "undefined" ?  
[] :  
seatChartsSettings.seats[fn.settings.character].classes  
).join(' '));
```

```
//basically a wrapper function
```

```
fn.data = function() {  
    return fn.settings.data;  
};
```

```
fn.char = function() {  
    return fn.settings.character;  
};
```

```
fn.node = function() {  
    return fn.settings.$node;  
};
```

```
/*
```

```
* Can either set or return status depending on arguments.
```

```
*
```

```
* If there's no argument, it will return the current style.
```

```
*
```

```

* If you pass an argument, it will update seat's style
*/
fn.style = function() {

    return arguments.length == 1 ?
        (function(newStyle) {
            var oldStyle = fn.settings.style;

            //if nothing changes, do nothing
            if (newStyle == oldStyle) {
                return oldStyle;
            }

            //focused is a special style which is
            not associated with status

            fn.settings.status = newStyle !=
            'focused' ? newStyle : fn.settings.status;

            fn.settings.$node
                .attr('aria-checked', newStyle
                == 'selected');

            //if user wants to animate status
            changes, let him do this

            seatChartsSettings.animate ?

            fn.settings.$node.switchClass(oldStyle, newStyle, 200) :

            fn.settings.$node.removeClass(oldStyle).addClass(newStyle);

```



```

        return fn.settings.style = newStyle;
    })(arguments[0]) : fn.settings.style;
};

//either set or retrieve
fn.status = function() {

    return fn.settings.status = arguments.length == 1 ?
        fn.style(arguments[0]) : fn.settings.status;
};

//using immediate function to conveniently get shortcut
variables
(function(seatSettings, character, seat) {
    //attach event handlers
    $.each(['click', 'focus', 'blur'], function(index,
callback) {

        //we want to be able to call the functions
        for each seat object

        fn[callback] = function() {
            if (callback == 'focus') {
                //if there's already a focused
                element, we have to remove focus from it first

                if (seatCharts.attr('aria-
activedescendant') !== undefined) {

                    seats[seatCharts.attr('aria-activedescendant')].blur();

                }
            }
        }
    });
});

```

```

        seatCharts.attr('aria-
activedescendant', seat.settings.id);

        seat.node().focus();
    }

    /*
    * User can pass his own callback
    * and if not, use our default
    *
    * Each callback function is executed
    */
    return fn.style(typeof
seatSettings[character][callback] === 'function' ?

        seatSettings[character][callback].apply(seat) : seatChartsSettings[callback].apply(seat));

    });

    //the below will become seatSettings, character, seat
    })(seatChartsSettings.seats, fn.settings.character, fn);

    fn.node()

    //the first three mouse events are simple
    .on('click',    fn.click)
    .on('mouseenter', fn.focus)

```

function, so we have to first check if it exists

callback.

in the current seat context.

seatSettings[character][callback] === 'function' ?

seatSettings[character][callback].apply(seat) : seatChartsSettings[callback].apply(seat));

};

});

thanks to the immediate function

})(seatChartsSettings.seats, fn.settings.character, fn);

fn.node()

//the first three mouse events are simple

.on('click', fn.click)

.on('mouseenter', fn.focus)

```
.on('mouseleave', fn.blur)
```

have to know where to move the focus

```
//keydown requires quite a lot of logic, because we
```

```
.on('keydown', (function(seat, $seat) {
```

```
    return function (e) {
```

```
        var $newSeat;
```

pressed key

```
        //everything depends on the
```

```
        switch (e.which) {
```

the same event mouse click does

```
            //spacebar will just trigger
```

```
            case 32:
```

```
                e.preventDefault();
```

```
                seat.click();
```

```
                break;
```

```
            //UP & DOWN
```

```
            case 40:
```

```
            case 38:
```

```
                e.preventDefault();
```

```
            /*
```

immediate function which searches for the first "focusable" row.

```
            * This is a recursive,
```

```
            *
```

immediate function because we want a convenient access to some DOM elements

recursion because sometimes we may hit an empty space rather than a seat.

```

    * We're using
    * We're using
    *
    */
    $newSeat = (function
findAvailable($rows, $seats, $currentRow) {
    var $newRow;

    //let's
    determine which row should we move to

    if
    (!$rows.index($currentRow) && e.which == 38) {
        //if
        this is the first row and user has pressed up arrow, move to the last row

        $newRow = $rows.last();
    } else if
    ($rows.index($currentRow) == $rows.length-1 && e.which == 40) {
        //if
        this is the last row and user has pressed down arrow, move to the first row

        $newRow = $rows.first();
    } else {
        //using
        eq to get an element at the desired index position

        $newRow = $rows.eq(
```

```

//if up arrow, then decrement the index, if down increment it
$rows.index($currentRow) + (e.which == 38 ? (-1) : (+1))

);
}

//now that we
know the row, let's get the seat using the current column position
$newSeat =
$newRow.find('.seatCharts-seat,.seatCharts-space').eq($seats.index($seat));

//if the seat
we found is a space, keep looking further

return
$newSeat.hasClass('seatCharts-space') ?

findAvailable($rows, $seats, $newRow) : $newSeat;

}}($seat
//get a
reference to the parent container and then select all rows but the header

.parents('.seatCharts-container')

.find('.seatCharts-row:not(.seatCharts-header)'),

$seat
//get a
reference to the parent row and then find all seat cells (both seats & spaces)

```

```

.parents('.seatCharts-row:first')

.find('.seatCharts-seat,.seatCharts-space'),

//get a
reference to the current row

$seat.parents('.seatCharts-row:not(.seatCharts-header)')

);

//we couldn't
determine the new seat, so we better give up

if (!$newSeat.length)
{
    return;
}

//remove focus from
the old seat and put it on the new one

seat.blur();

seats[$newSeat.attr('id')].focus();

$newSeat.focus();

//update our "aria"
reference with the new seat id

seatCharts.attr('aria-
activedescendant', $newSeat.attr('id'));

```

slightly different from the one for up/down arrows.

browse the whole map using just left/right arrow, because

next row when we reach the right/left-most seat.

```
(function($seats) {
```

```
(!$seats.index($seat) && e.which == 37) {
```

has pressed left arrow and we're currently on the left-most seat

```
$seats.last();
```

```
($seats.index($seat) == $seats.length - 1 && e.which == 39) {
```

has pressed right arrow and we're currently on the right-most seat

```
$seats.first();
```

```
break;
```

```
//LEFT & RIGHT
```

```
case 37:
```

```
case 39:
```

```
e.preventDefault();
```

```
/*
```

```
* The logic here is
```

```
* User will be able to
```

```
* it will move to the
```

```
*/
```

```
$newSeat =
```

```
if
```

```
//user
```

```
return
```

```
} else if
```

```
//user
```

```
return
```

```
} else {
```

```

        //simply move one seat left or right depending on the key
        return
    $seats.eq($seats.index($seat) + (e.which == 37 ? (-1) : (+1)));
    }

    }($seat

    .parents('.seatCharts-container:first')

    .find('.seatCharts-seat:not(.seatCharts-space)'));

    if (!$newSeat.length)

        return;
    }

    //handle focus
    seat.blur();

    seats[$newSeat.attr('id')].focus();

    $newSeat.focus();

    //update our "aria"
    reference with the new seat id
    seatCharts.attr('aria-
    activedescendant', $newSeat.attr('id'));

    break;
default:
    break;

```



```

        }
    };

    })(fn, fn.node());
    // .appendTo(seatCharts.find('.' + row));

    }
})(fn, settings);

fn.addClass('seatCharts-container');

//true -> deep copy!
$.extend(true, settings, setup);

//Generate default row ids unless user passed his own
settings.namings.rows = settings.naming.rows || (function(length) {
    var rows = [];
    for (var i = 1; i <= length; i++) {
        rows.push(i);
    }
    return rows;
})(settings.map.length);

//Generate default column ids unless user passed his own
settings.naming.columns = settings.naming.columns || (function(length) {
    var columns = [];

```

```

        for (var i = 1; i <= length; i++) {
            columns.push(i);
        }
        return columns;
    })(settings.map[0].split('').length);

    if (settings.naming.top) {
        var $headerRow = $('<div></div>')
            .addClass('seatCharts-row seatCharts-header');

        if (settings.naming.left) {
            $headerRow.append($('<div></div>').addClass('seatCharts-cell'));
        }

        $.each(settings.naming.columns, function(index, value) {
            $headerRow.append(
                $('<div></div>')
                    .addClass('seatCharts-cell')
                    .text(value)
            );
        });
    }

    fn.append($headerRow);

    //do this for each map row

```

```
$.each(settings.map, function(row, characters) {
```

```
    var $row = $('<div></div>').addClass('seatCharts-row');
```

```
    if (settings.naming.left) {
```

```
        $row.append(
```

```
            $('<div></div>')
```

```
                .addClass('seatCharts-cell seatCharts-space')
```

```
                .text(settings.naming.rows[row])
```

```
        );
```

```
    }
```

```
/*
```

```
 * Do this for each seat (letter)
```

```
 *
```

* Now users will be able to pass custom ID and label which overwrite the one that seat would be assigned by getId and

```
 * getLabel
```

```
 *
```

```
 * New format is like this:
```

```
 * a[ID,label]a[ID]aaaaa
```

```
 *
```

```
 * So you can overwrite the ID or label (or both) even for just one seat.
```

```
 * Basically ID should be first, so if you want to overwrite just label write
```

it as follows:

```
 * a[,LABEL]
```

```
 *
```

```
 * Allowed characters in IDs areL 0-9, a-z, A-Z, _
```

* Allowed characters in labels are: 0-9, a-z, A-Z, _ ' ' (space)

*

*/

```
$.each(characters.match(/[a-z_]{1}(\[[0-9a-z]{0,}([0-9a-z_]+)?\])?/gi),
function (column, characterParams) {
    var matches    = characterParams.match(/([a-z_]{1})(\[[0-9a-z_
    ],+)\])?/i),

    //no matter if user specifies [] params, the character
    should be in the second element
    character      = matches[1],

    //check if user has passed some additional params to
    override id or label
    params         = typeof matches[3] !== 'undefined' ?
    matches[3].split(',') : [],

    //id param should be first
    overrideId     = params.length ? params[0] : null,

    //label param should be second
    overrideLabel  = params.length === 2 ? params[1] : null;

    $row.append(character !== '_' ?

    //if the character is not an underscore (empty space)
    (function(naming) {

        //so users don't have to specify empty objects
        settings.seats[character] = character in
    settings.seats ? settings.seats[character] : {};
```

```

                                var id = overrideId ? overrideId :
naming.getId(character, naming.rows[row], naming.columns[column]);

                                seats[id] = new seat({
                                    id      : id,
                                    label   : overrideLabel ?
                                                overrideLabel :
naming.getLabel(character, naming.rows[row], naming.columns[column]),
                                    row     : row,
                                    column  : column,
                                    character : character
                                });

                                seatIds.push(id);
                                return seats[id].node();

                                })(settings.naming) :
                                //this is just an empty space ( )
                                $('<div></div>').addClass('seatCharts-cell seatCharts-
space')

                                );

                                });

                                fn.append($row);

                                });

                                //if there're any legend items to be rendered
                                settings.legend.items.length ? (function(legend) {

```

//either use user-defined container or create our own and insert it right
after the seat chart div

```
var $container = (legend.node || $('<div></div>').insertAfter(fn))  
    .addClass('seatCharts-legend');
```

```
var $ul = $('<ul></ul>')  
    .addClass('seatCharts-legendList')  
    .appendTo($container);
```

```
$.each(legend.items, function(index, item) {  
    $ul.append(  

```

```
        $('<li></li>')
```

```
        .addClass('seatCharts-legendItem')
```

```
        .append(  

```

```
            $('<div></div>')
```

```
            //merge user defined classes with
```

our standard ones

```
            .addClass(['seatCharts-seat',
```

```
'seatCharts-cell', item[1]].concat(  

```

```
                settings.classes,
```

```
                typeof settings.seats[item[0]]
```

```
                == "undefined" ? [] : settings.seats[item[0]].classes).join(' ')
```

```
            )
```

```
        )
```

```
        .append(  

```

```
            $('<span></span>')
```

legendDescription')

```
            .addClass('seatCharts-
```

```
            .text(item[2])
```

```

        )

    );

});

    return $container;
})(settings.legend) : null;

fn.attr({
    tabIndex : 0
});

//when container's focused, move focus to the first seat
fn.focus(function() {
    if (fn.attr('aria-activedescendant')) {
        seats[fn.attr('aria-activedescendant')].blur();
    }

    fn.find('.seatCharts-seat:not(.seatCharts-space):first').focus();
    seats[seatIds[0]].focus();

});

//public methods of seatCharts
fn.data('seatCharts', {
    seats : seats,
    seatIds : seatIds,

```

```

//set for one, set for many, get for one
status: function() {
    var fn = this;

    return arguments.length == 1 ? fn.seats[arguments[0]].status() :
(function(seatsIds, newStatus) {

    return typeof seatsIds == 'string' ?
fn.seats[seatsIds].status(newStatus) : (function() {

        $.each(seatsIds, function(index, seatId) {
            fn.seats[seatId].status(newStatus);
        });

    })();

})(arguments[0], arguments[1]);

},

each : function(callback) {
    var fn = this;

    for (var seatId in fn.seats) {
        if (false === callback.call(fn.seats[seatId], seatId)) {
            return seatId;//return last checked
        }
    }

    return true;
},

node : function() {
    var fn = this;

```



```

        //basically create a CSS query to get all seats by their DOM ids
        return $('#'+ fn.seatIds.join('#'));
    },

```

```

    find : function(query) { //D, a.available, unavailable
        var fn = this;

```

```

        var seatSet = fn.set();

```

```

        //user searches just for a particular character
        return query.length == 1 ? (function(character) {
            fn.each(function() {
                if (this.char() == character) {
                    seatSet.push(this.settings.id, this);
                }
            });

```

```

            return seatSet;
        })(query) : (function() {

```

there's a dot

```

            //user runs a more sophisticated query, so let's see if

```

status

```

            return query.indexOf('.') > -1 ? (function() {
                //there's a dot which separates character and the

```

```

                var parts = query.split('.');

```

```

                fn.each(function(seatId) {

```

```

        if (this.char() == parts[0] && this.status() ==

parts[1]) {

            seatSet.push(this.settings.id, this);

        }

    });

    return seatSet;

})(); : (function() {

    fn.each(function() {

        if (this.status() == query) {

            seatSet.push(this.settings.id, this);

        }

    });

    return seatSet;

})();

})();

},

set : function set() { //inherits some methods

    var fn = this;

    return {

        seats : [],

        seatIds : [],

        length : 0,

```

```

        status : function() {
            var args = arguments,
                that = this;

            //if there's just one seat in the set and user didn't
pass any params, return current status

            return this.length == 1 && args.length == 0 ?

this.seats[0].status() : (function() {

                //otherwise call status function for each of
the seats in the set

                $.each(that.seats, function() {
                    this.status.apply(this, args);
                });
            })();
        },
        node : function() {
            return fn.node.call(this);
        },
        each : function() {
            return fn.each.call(this, arguments[0]);
        },
        get : function() {
            return fn.get.call(this, arguments[0]);
        },
        find : function() {
            return fn.find.call(this, arguments[0]);
        },
        set : function() {
            return set.call(fn);
        }
    }

```

```

        },
        push : function(id, seat) {
            this.seats.push(seat);
            this.seatIds.push(id);
            ++this.length;
        }
    };

},

//get one object or a set of objects
get : function(seatIds) {
    var fn = this;

    return typeof seatIds == 'string' ?
        fn.seats[seatIds] : (function() {

            var seatSet = fn.set();

            $.each(seatIds, function(index, seatId) {
                if (typeof fn.seats[seatId] === 'object') {
                    seatSet.push(seatId,
fn.seats[seatId]);
                }
            });

            return seatSet;
        })();
    }
}

```

```
});
```

```
return fn.data('seatCharts');
```

```
}
```

NAS TRAVELS

Bus Ticket Booking

[Home](#) [About Us](#)

From:

To:

Date:

Number of seats:

[bookTickets](#)

[About Us](#)

[bookTickets](#)

About Us

Our bus ticket booking website is committed to providing you with easy and hassle-free bus ticket bookings. We strive to offer a one-stop solution for all your bus travel needs, covering all major routes across the country. Our user-friendly platform allows you to search, compare, and book tickets within minutes. Our extensive network of bus operators ensures that you have access to a wide range of options that suit your preferences and budget. We take pride in our commitment to customer safety and comfort. All our bus operators are handpicked, and we ensure that they comply with all safety regulations. You can rest assured that you will experience a comfortable and safe journey with us. We also provide various payment options to make the booking process more convenient for you. You can pay easily using credit/debit cards, net banking, or e-wallets. Our dedicated customer support team is available 24/7 to assist you with any queries or issues you may have. In addition to booking bus tickets, we also offer various deals and discounts, making your travel experience even more affordable. We aim to provide you with a seamless and enjoyable journey, from start to finish. Choose us for your next bus journey and experience the convenience and comfort of hassle-free booking of bus tickets!

©2023 Bus Booking website. All Rights Reserved

[contact us](#)

@9787124534

nas@gmail.com

