Flight Ticket Booking

Frontend:

1. HTML
2. CSS
3. JavaScript

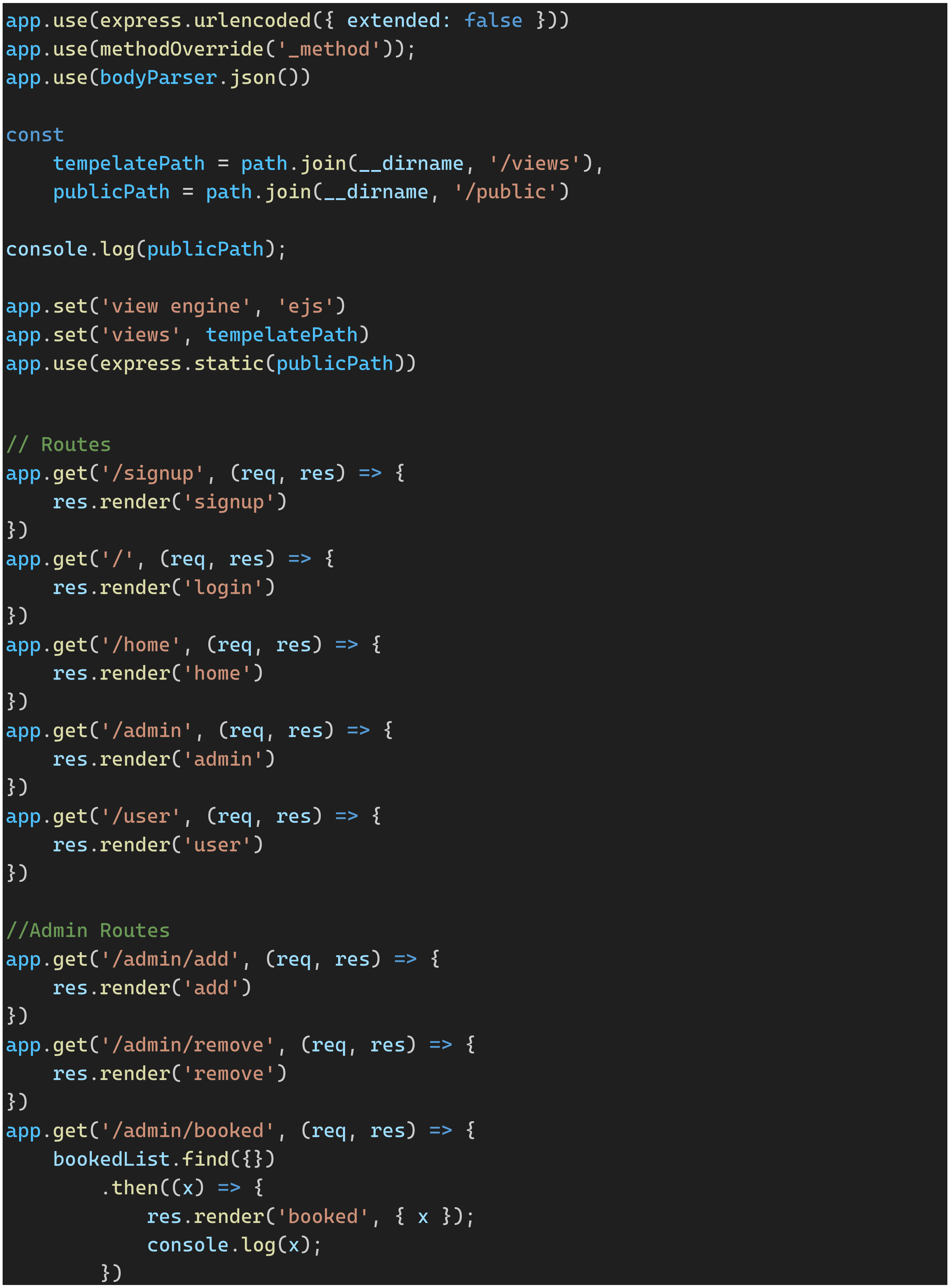
Backend :

1. NodeJS
2. Express.Js
3. MongoDB

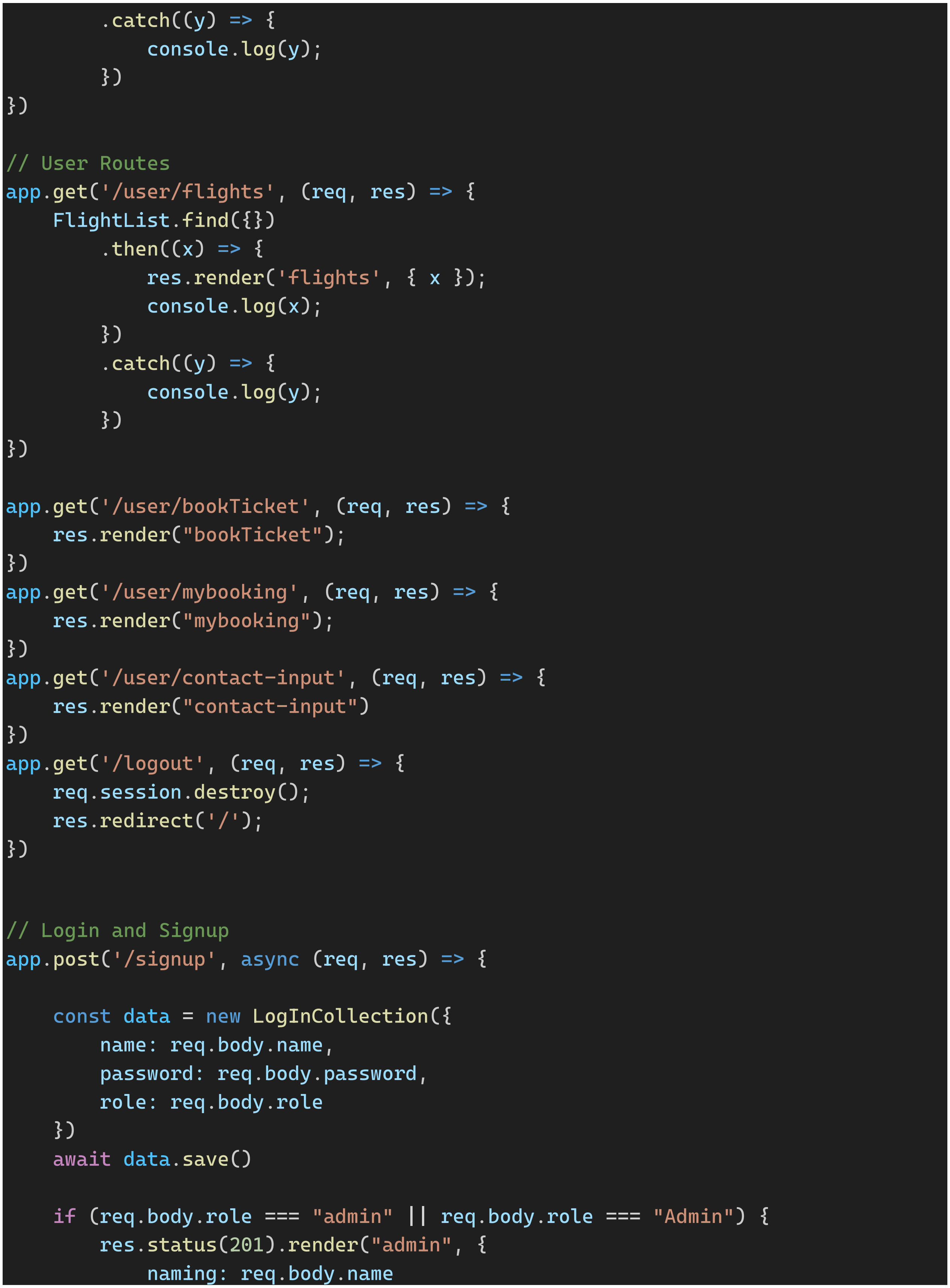
Backend Code:

app.js

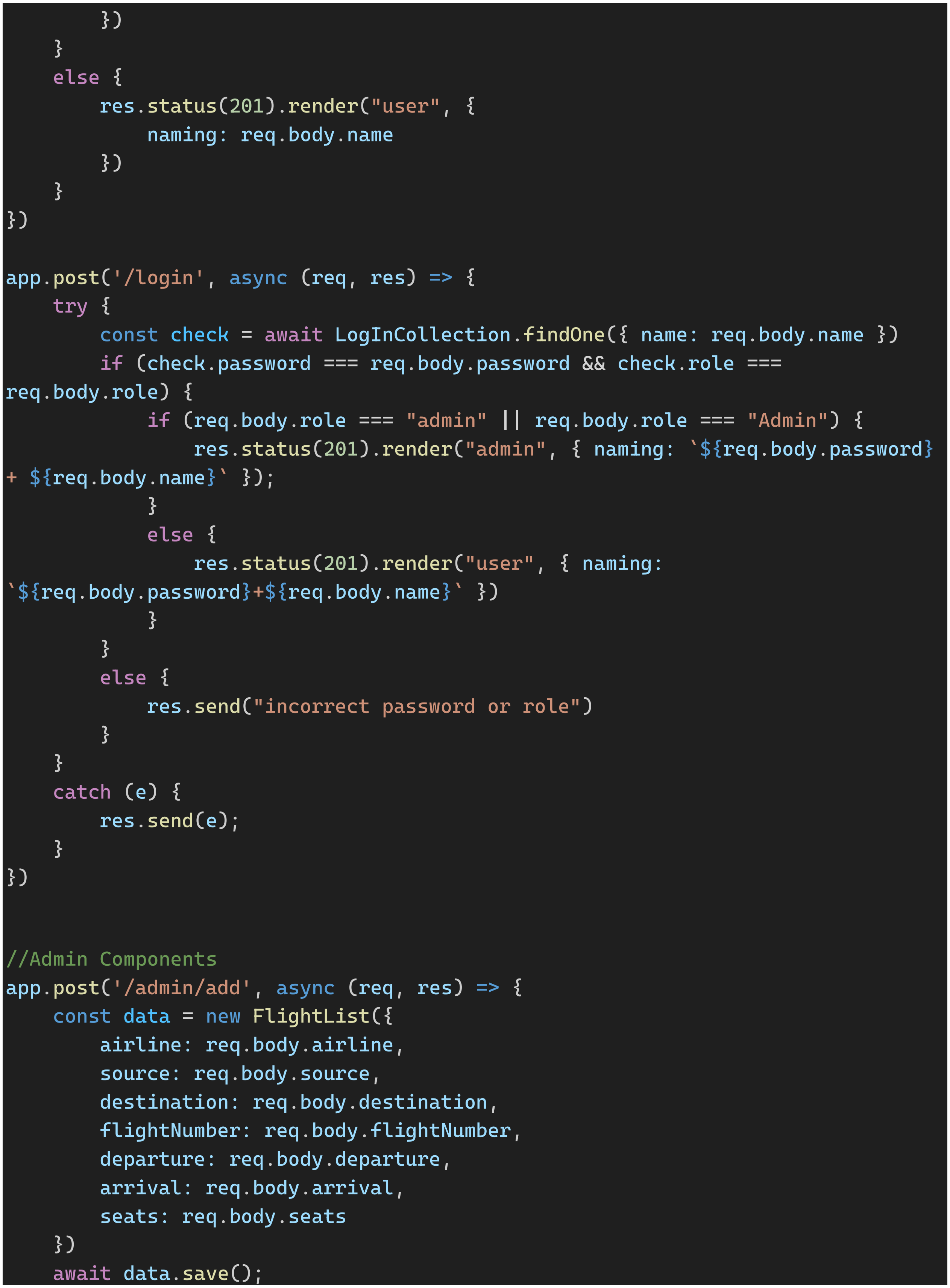
```
const
    express = require("express"),
    alert = require("alert"),
    path = require("path"),
    app = express(),
    { FlightList, LogInCollection, bookedList } = require("./mongo"),
    bodyParser = require('body-parser'),
    port = process.env.PORT || 3000,
    hostname = '127.0.0.1',
    methodOverride = require('method-override'),
    session = require('express-session'),
    flash = require('connect-flash')


// Use the method-override middleware
app.use(session({
    secret: 'flight-ticket-booking',
    saveUninitialized: true,
    resave: true
}));

app.use(flash())
app.use(express.json())
```



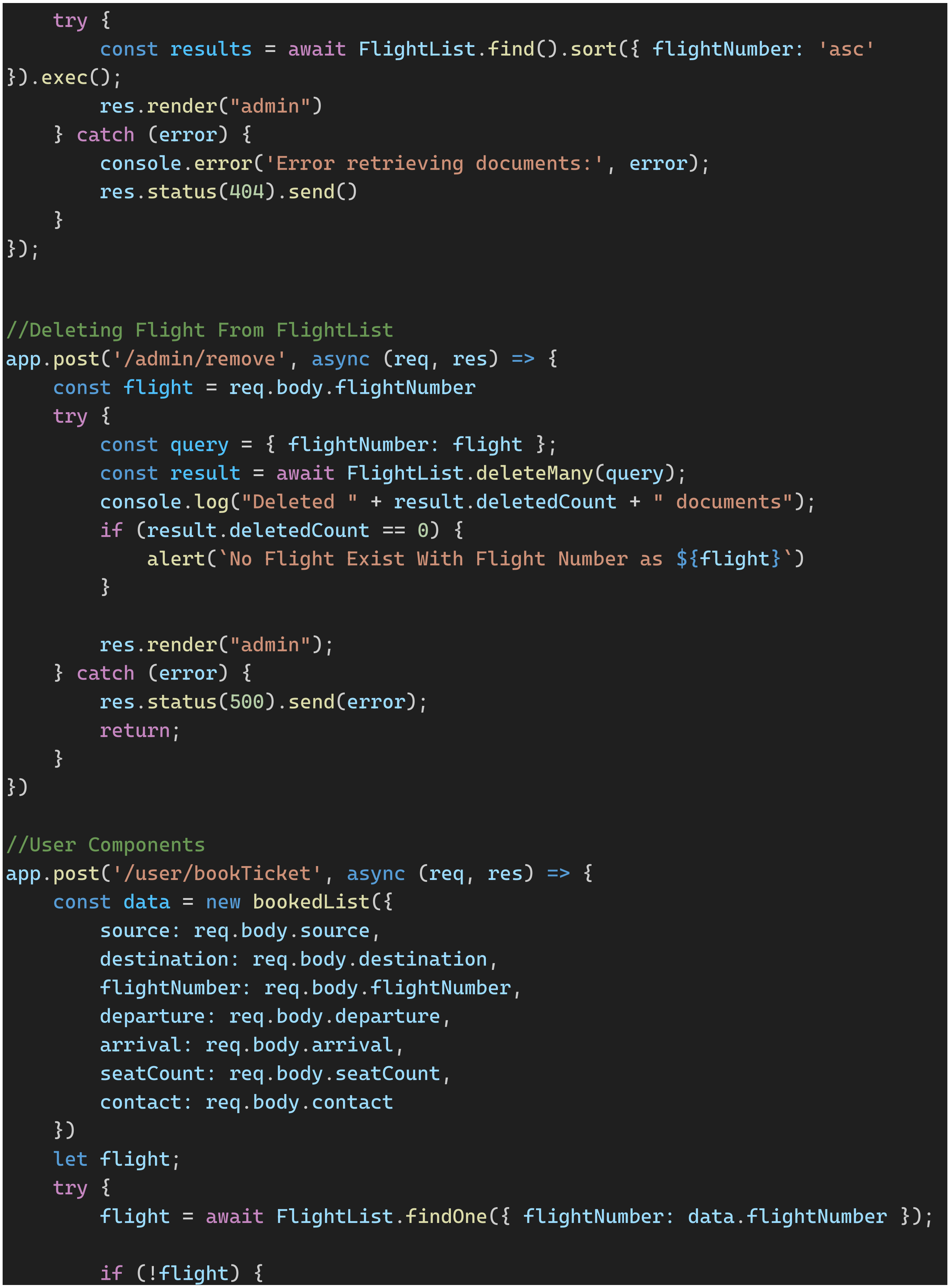
```
app.use(express.urlencoded({ extended: false }))
app.use(methodOverride('_method'));
app.use(bodyParser.json())

const
    tempelatePath = path.join(__dirname, '/views'),
    publicPath = path.join(__dirname, '/public')

console.log(publicPath);

app.set('view engine', 'ejs')
app.set('views', tempelatePath)
app.use(express.static(publicPath))


// Routes
app.get('/signup', (req, res) => {
    res.render('signup')
})
app.get('/', (req, res) => {
    res.render('login')
})
app.get('/home', (req, res) => {
    res.render('home')
})
app.get('/admin', (req, res) => {
    res.render('admin')
})
app.get('/user', (req, res) => {
    res.render('user')
})

//Admin Routes
app.get('/admin/add', (req, res) => {
    res.render('add')
})
app.get('/admin/remove', (req, res) => {
    res.render('remove')
})
app.get('/admin/booked', (req, res) => {
    bookedList.find({})
        .then((x) => {
            res.render('booked', { x });
            console.log(x);
        })
```



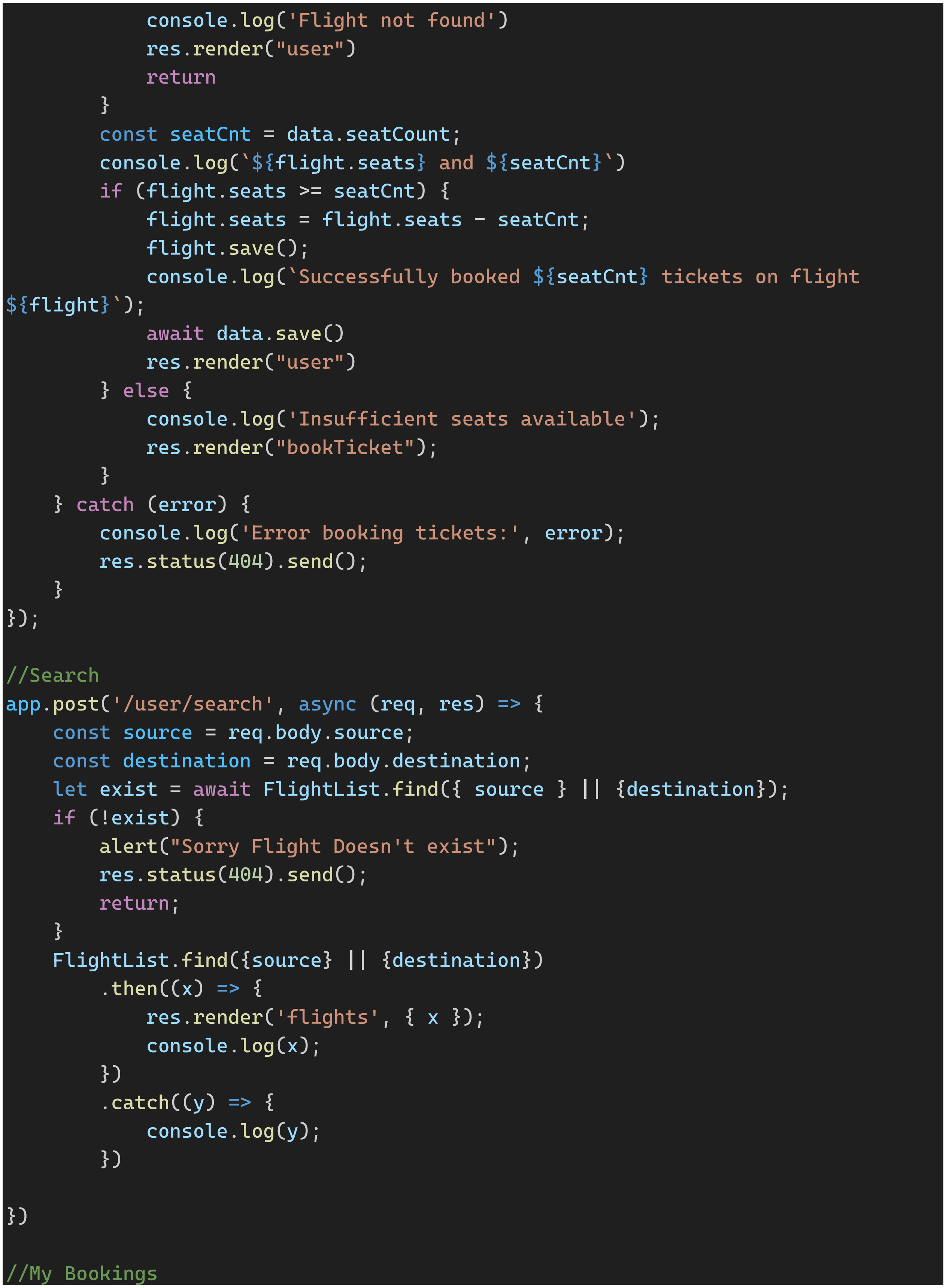
```
        .catch((y) => {
            console.log(y);
        })
})

// User Routes
app.get('/user/flights', (req, res) => {
    FlightList.find({})
        .then((x) => {
            res.render('flights', { x });
            console.log(x);
        })
        .catch((y) => {
            console.log(y);
        })
})

app.get('/user/bookTicket', (req, res) => {
    res.render("bookTicket");
})
app.get('/user/mybooking', (req, res) => {
    res.render("mybooking");
})
app.get('/user/contact-input', (req, res) => {
    res.render("contact-input")
})
app.get('/logout', (req, res) => {
    req.session.destroy();
    res.redirect('/');
})


// Login and Signup
app.post('/signup', async (req, res) => {

    const data = new LogInCollection({
        name: req.body.name,
        password: req.body.password,
        role: req.body.role
    })
    await data.save()

    if (req.body.role === "admin" || req.body.role === "Admin") {
        res.status(201).render("admin", {
            naming: req.body.name
```



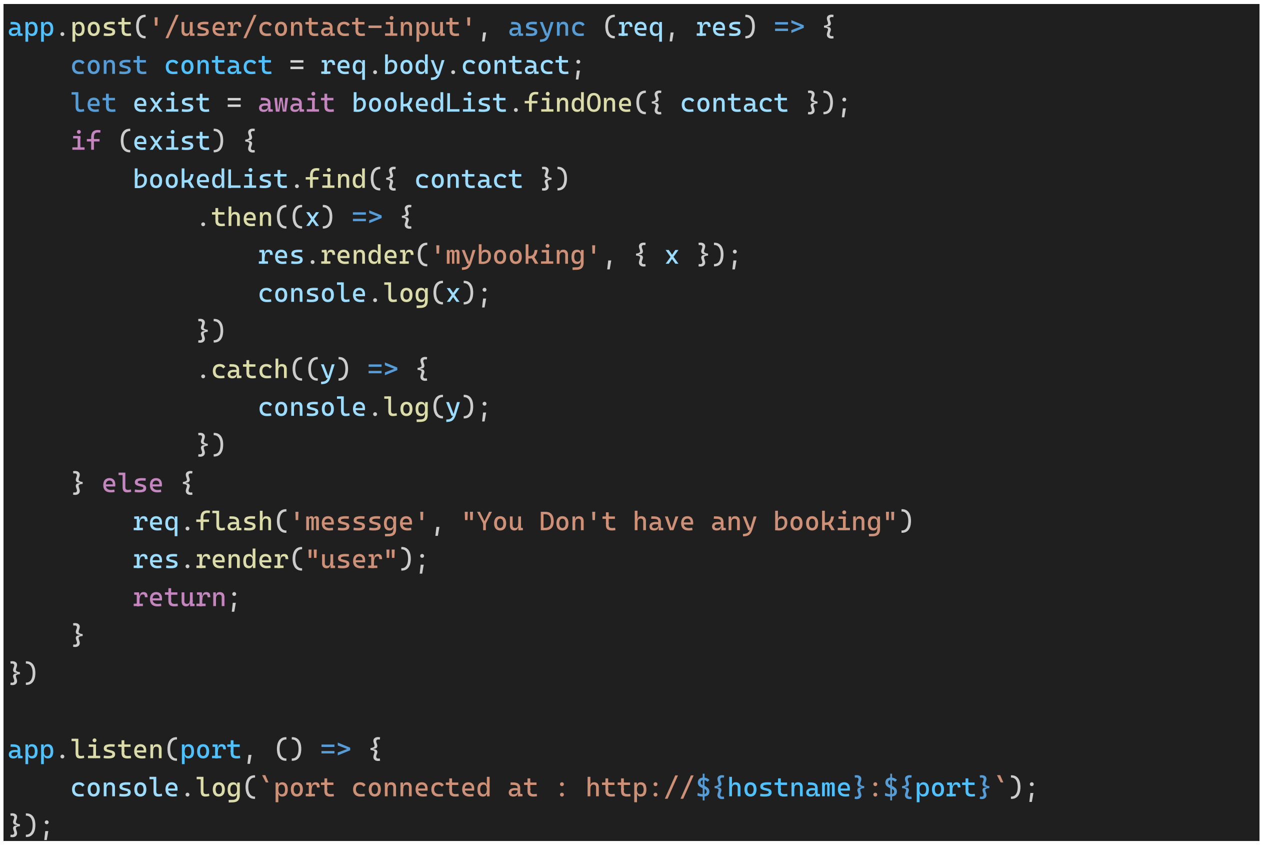
```
        })
    }
    else {
        res.status(201).render("user", {
            naming: req.body.name
        })
    }
})

app.post('/login', async (req, res) => {
    try {
        const check = await LogInCollection.findOne({ name: req.body.name })
        if (check.password === req.body.password && check.role ===
req.body.role) {
            if (req.body.role === "admin" || req.body.role === "Admin") {
                res.status(201).render("admin", { naming: `${req.body.password}
+ ${req.body.name}` });
            }
            else {
                res.status(201).render("user", { naming:
`${req.body.password}+${req.body.name}` })
            }
        }
        else {
            res.send("incorrect password or role")
        }
    }
    catch (e) {
        res.send(e);
    }
})


//Admin Components
app.post('/admin/add', async (req, res) => {
    const data = new FlightList({
        airline: req.body.airline,
        source: req.body.source,
        destination: req.body.destination,
        flightNumber: req.body.flightNumber,
        departure: req.body.departure,
        arrival: req.body.arrival,
        seats: req.body.seats
    })
    await data.save();
```



```
    try {
        const results = await FlightList.find().sort({ flightNumber: 'asc'
}).exec();
        res.render("admin")
    } catch (error) {
        console.error('Error retrieving documents:', error);
        res.status(404).send()
    }
});


//Deleting Flight From FlightList
app.post('/admin/remove', async (req, res) => {
    const flight = req.body.flightNumber
    try {
        const query = { flightNumber: flight };
        const result = await FlightList.deleteMany(query);
        console.log("Deleted " + result.deletedCount + " documents");
        if (result.deletedCount == 0) {
            alert(`No Flight Exist With Flight Number as ${flight}`)
        }

        res.render("admin");
    } catch (error) {
        res.status(500).send(error);
        return;
    }
})

//User Components
app.post('/user/bookTicket', async (req, res) => {
    const data = new bookedList({
        source: req.body.source,
        destination: req.body.destination,
        flightNumber: req.body.flightNumber,
        departure: req.body.departure,
        arrival: req.body.arrival,
        seatCount: req.body.seatCount,
        contact: req.body.contact
    })
    let flight;
    try {
        flight = await FlightList.findOne({ flightNumber: data.flightNumber });

        if (!flight) {
```

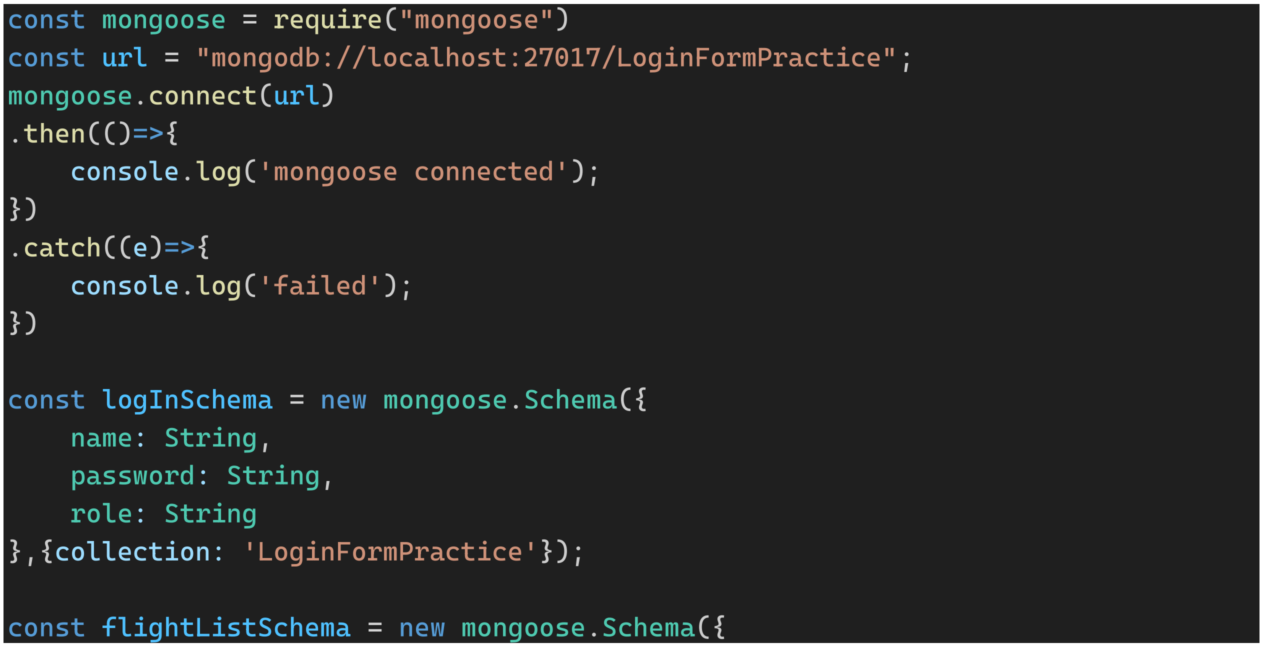


```
            console.log('Flight not found')
            res.render("user")
            return
        }
        const seatCnt = data.seatCount;
        console.log(`${flight.seats} and ${seatCnt}`)
        if (flight.seats >= seatCnt) {
            flight.seats = flight.seats - seatCnt;
            flight.save();
            console.log(`Successfully booked ${seatCnt} tickets on flight
${flight}`);
            await data.save()
            res.render("user")
        } else {
            console.log('Insufficient seats available');
            res.render("bookTicket");
        }
    } catch (error) {
        console.log('Error booking tickets:', error);
        res.status(404).send();
    }
});

//Search
app.post('/user/search', async (req, res) => {
    const source = req.body.source;
    const destination = req.body.destination;
    let exist = await FlightList.find({ source } || {destination});
    if (!exist) {
        alert("Sorry Flight Doesn't exist");
        res.status(404).send();
        return;
    }
    FlightList.find({source} || {destination})
        .then((x) => {
            res.render('flights', { x });
            console.log(x);
        })
        .catch((y) => {
            console.log(y);
        })

})

//My Bookings
```

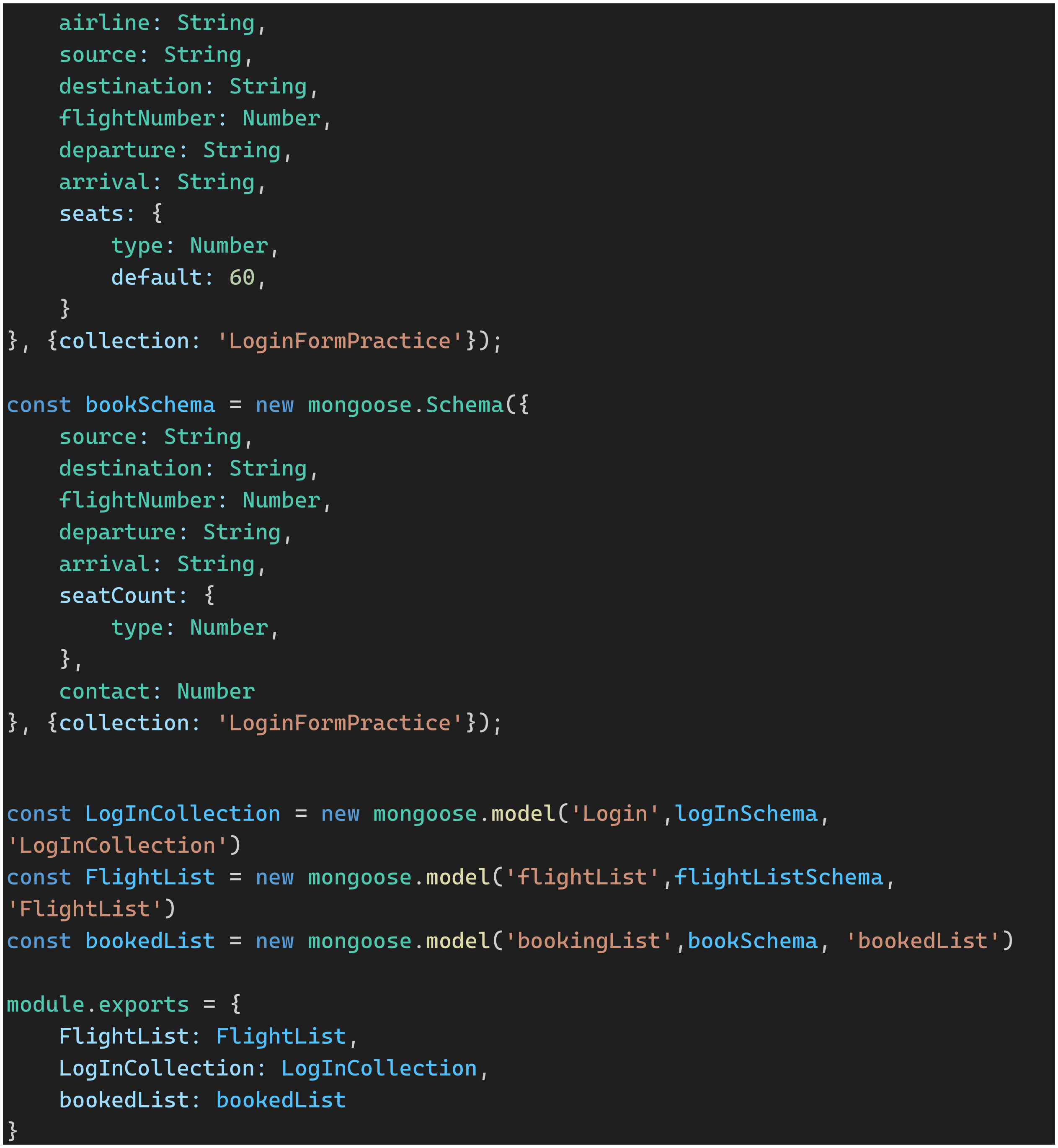


```
app.post('/user/contact-input', async (req, res) => {
    const contact = req.body.contact;
    let exist = await bookedList.findOne({ contact });
    if (exist) {
        bookedList.find({ contact })
            .then((x) => {
                res.render('mybooking', { x });
                console.log(x);
            })
            .catch((y) => {
                console.log(y);
            })
    } else {
        req.flash('messsge', "You Don't have any booking")
        res.render("user");
        return;
    }
})

app.listen(port, () => {
    console.log(`port connected at : http://${hostname}:${port}`);
});
```

mongo.js



```
const mongoose = require("mongoose")
const url = "mongodb://localhost:27017/LoginFormPractice";
mongoose.connect(url)
.then(()=>{
    console.log('mongoose connected');
})
.catch((e)=>{
    console.log('failed');
})

const logInSchema = new mongoose.Schema({
    name: String,
    password: String,
    role: String
},{collection: 'LoginFormPractice'});

const flightListSchema = new mongoose.Schema({
```



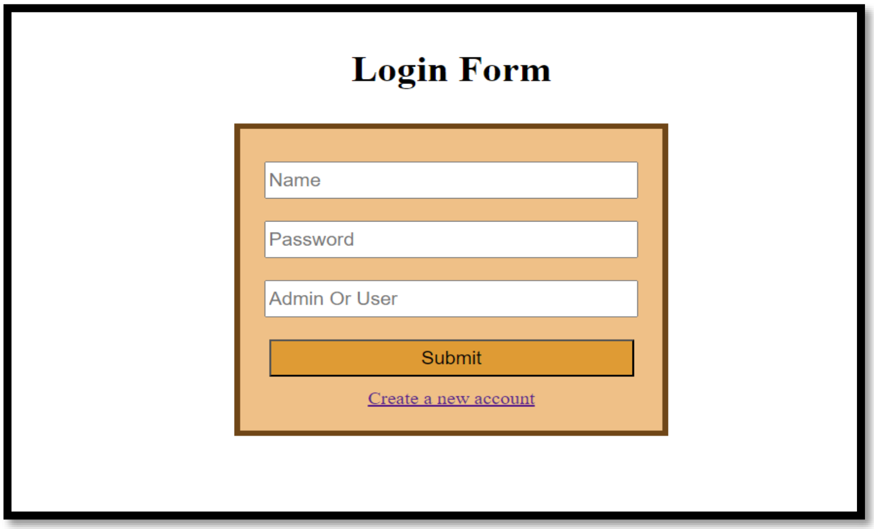
```
    airline: String,
    source: String,
    destination: String,
    flightNumber: Number,
    departure: String,
    arrival: String,
    seats: {
        type: Number,
        default: 60,
    }
}, {collection: 'LoginFormPractice'});

const bookSchema = new mongoose.Schema({
    source: String,
    destination: String,
    flightNumber: Number,
    departure: String,
    arrival: String,
    seatCount: {
        type: Number,
    },
    contact: Number
}, {collection: 'LoginFormPractice'});


const LogInCollection = new mongoose.model('Login',logInSchema,
'LogInCollection')
const FlightList = new mongoose.model('flightList',flightListSchema,
'FlightList')
const bookedList = new mongoose.model('bookingList',bookSchema, 'bookedList')

module.exports = {
    FlightList: FlightList,
    LogInCollection: LogInCollection,
    bookedList: bookedList
}
```

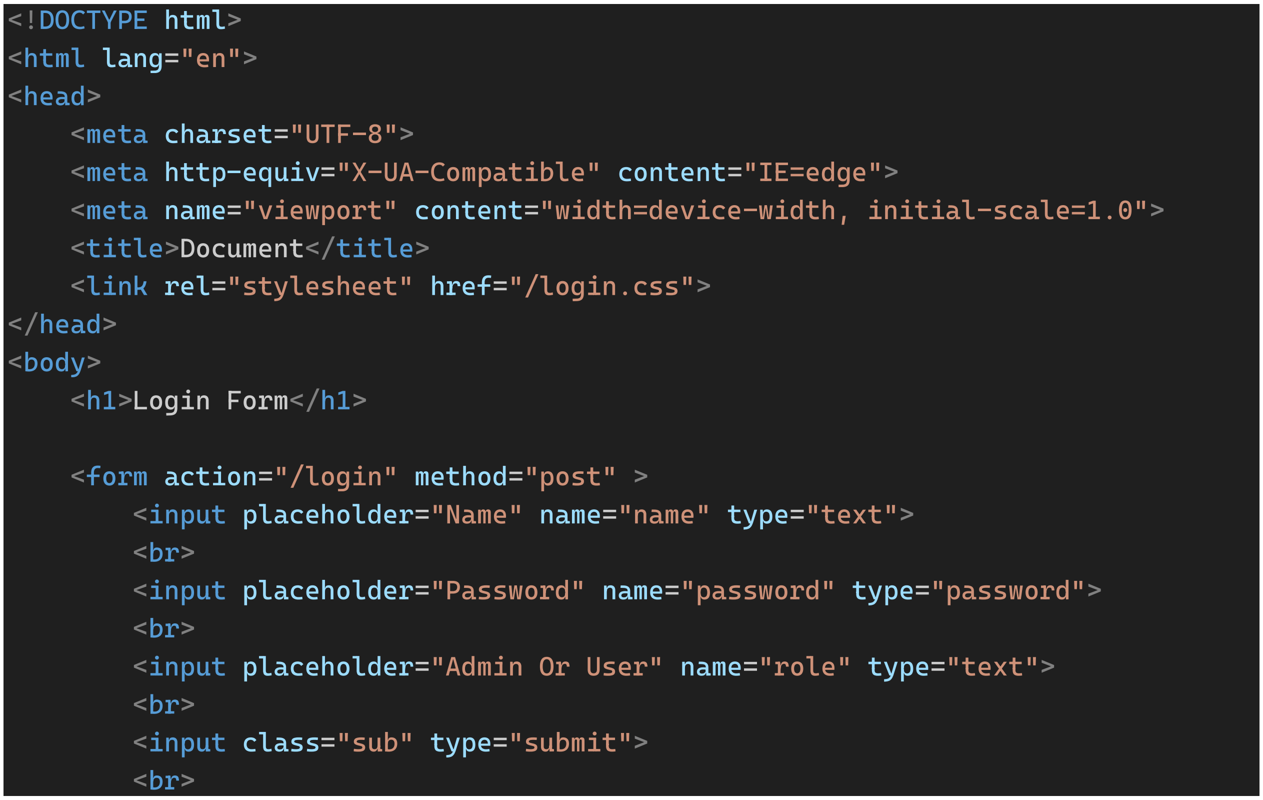
Source Code with Visualization:

1. Login and signup Page:

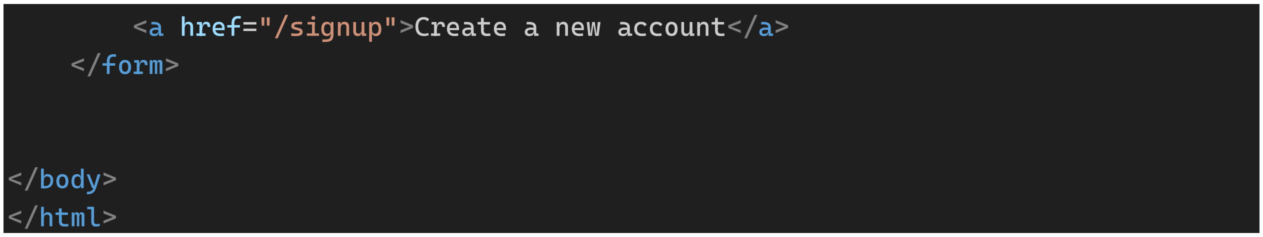
   Source code:

(a) Login.ejs



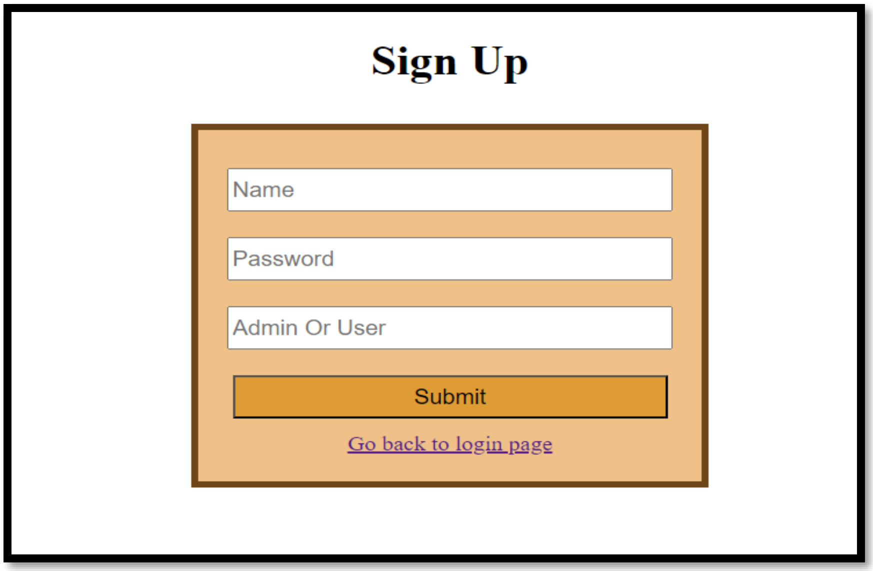


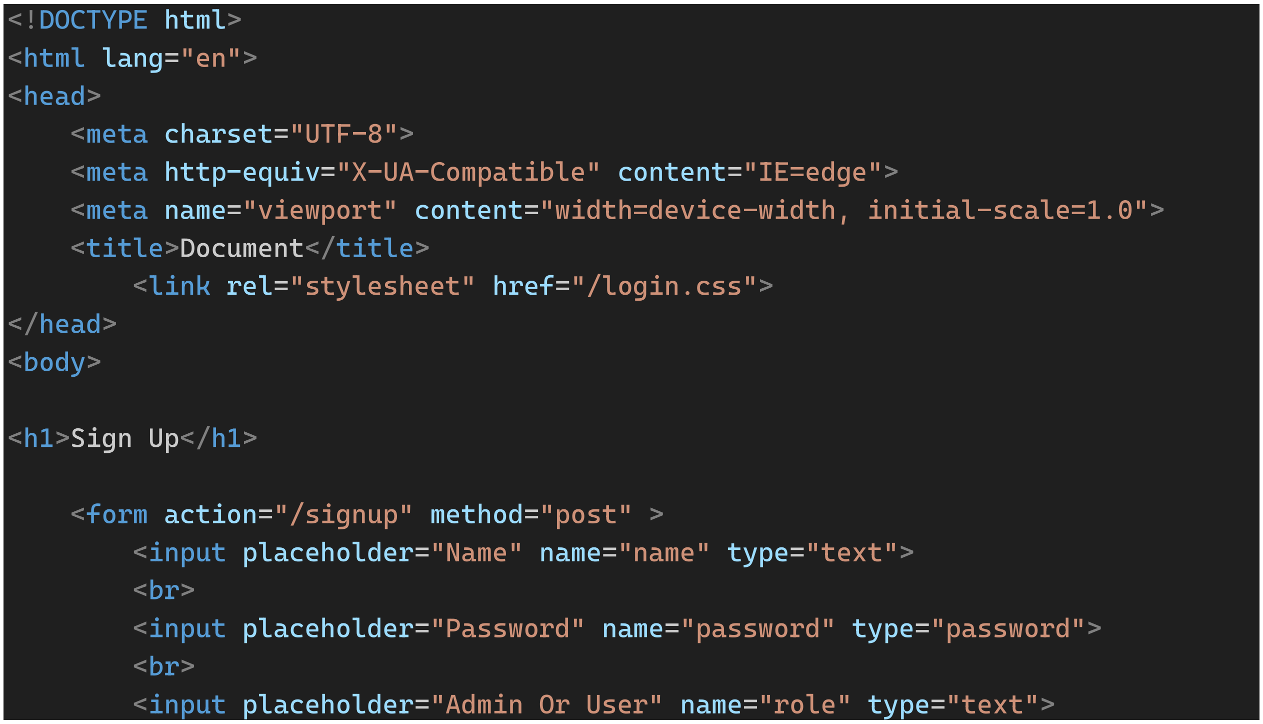
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="/login.css">
</head>
<body>
    <h1>Login Form</h1>

    <form action="/login" method="post" >
        <input placeholder="Name" name="name" type="text">
        <br>
        <input placeholder="Password" name="password" type="password">
        <br>
        <input placeholder="Admin Or User" name="role" type="text">
        <br>
        <input class="sub" type="submit">
        <br>
```



```
        <a href="/signup">Create a new account</a>
    </form>


</body>
</html>
```

(b) Signup.ejs



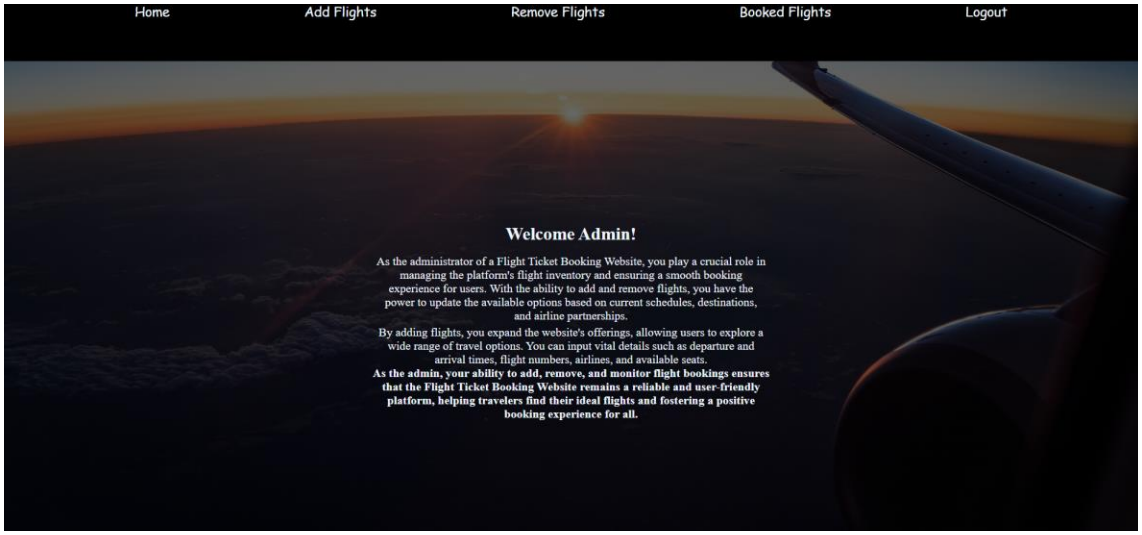


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
        <link rel="stylesheet" href="/login.css">
</head>
<body>

<h1>Sign Up</h1>

    <form action="/signup" method="post" >
        <input placeholder="Name" name="name" type="text">
        <br>
        <input placeholder="Password" name="password" type="password">
        <br>
        <input placeholder="Admin Or User" name="role" type="text">
        <br>
        <input class="sub" type="submit">
        <br>
        <a href="/">Go back to login page</a>
    </form>


</body>
</html>
```

Admin Use Cases:

1. Home Page – *admin.ejs*

Simple admin page



Welcome Admin!

As the administrator of a Flight Ticket Booking Website, you play a crucial role in managing the platform's flight inventory and ensuring a smooth booking experience for users. With the ability to add and remove flights, you have the power to update the available options based on current schedules, destinations, and airline partnerships.

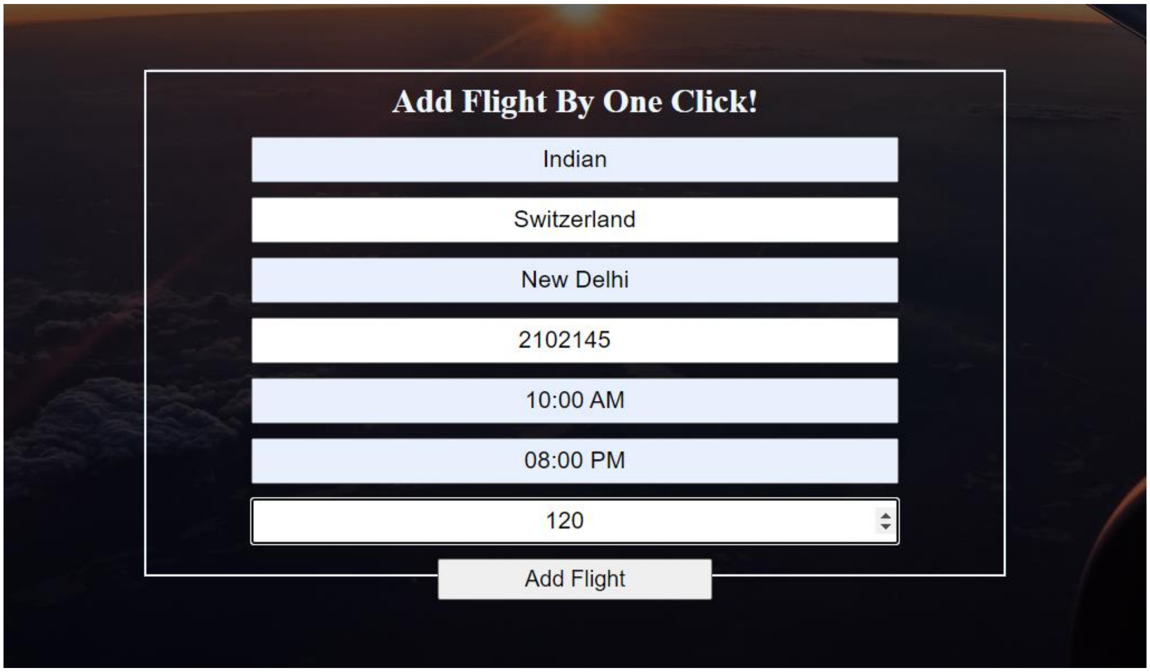
By adding flights, you expand the website's offerings, allowing users to explore a wide range of travel options. You can input vital details such as departure and arrival times, flight numbers, airlines, and available seats.

As the admin, your ability to add, remove, and monitor flight bookings ensures that the Flight Ticket Booking Website remains a reliable and user-friendly platform, helping travelers find their ideal flights and fostering a positive booking experience for all.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/admin.css">
    <title>Document</title>
</head>

<body>
```

```
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="">Home</a></li>
                <li class="item"><a href="/admin/add">Add Flights</a></li>
                <li class="item"><a href="/admin/remove">Remove
Flights</a></li>
                <li class="item"><a href="/admin/booked">Booked
Flights</a></li>
                <li class="item"><a href="/">Logout</a></li>
            </ul>
        </nav>
    </header>
    <section id="home">
        <div class="card">
            <h2 style="margin: 10px;">Welcome Admin!</h2>
            <p style="padding: 4px;">
                As the administrator of a Flight Ticket Booking Website, you
play a crucial role in managing the
                platform's flight inventory and ensuring a smooth booking
experience for users. With the ability to add
                and remove flights, you have the power to update the available
options based on current schedules,
                destinations, and airline partnerships.</p>
            <p> By adding flights, you expand the website's offerings, allowing
users to explore a wide range of travel
                options. You can input vital details such as departure and
arrival times, flight numbers, airlines, and
                available seats.
            </p>
            <b>As the admin, your ability to add, remove, and monitor flight
bookings ensures that the Flight Ticket
                Booking Website remains a reliable and user-friendly platform,
helping travelers find their ideal
                flights and fostering a positive booking experience for all.
            </b>
        </div>
    </section>
</body>

</html>
```

2. To Add Flights To FlightList Collection
   add.ejs



Indian

Switzerland

New Delhi

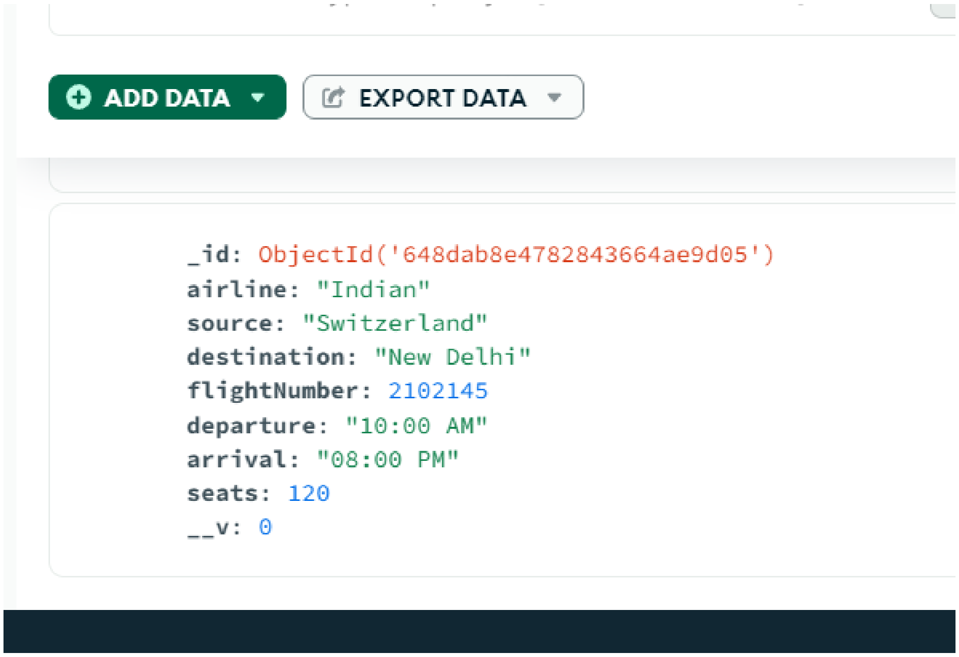
2102145

10:00 AM

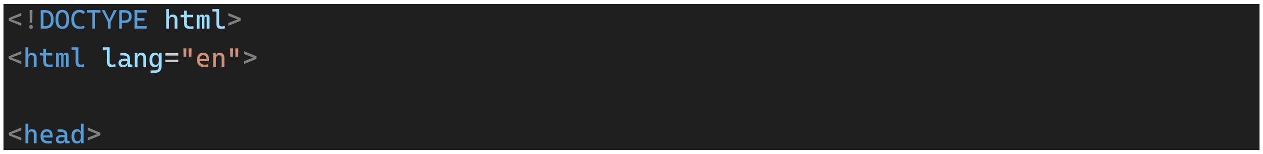
08:00 PM

120

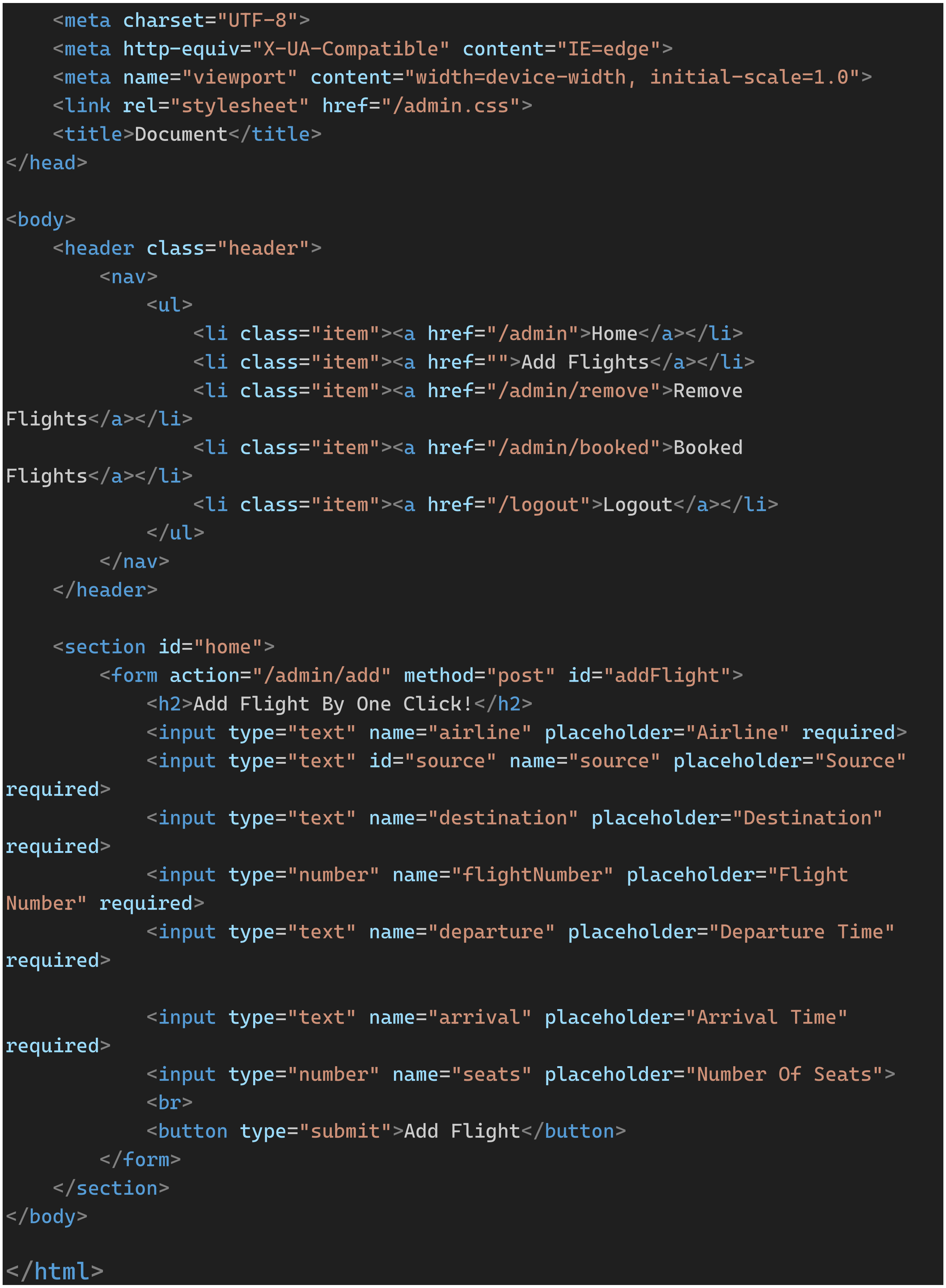
Flight is Being Added



```
_id: ObjectId('648dab8e4782843664ae9d05')
airline: "Indian"
source: "Switzerland"
destination: "New Delhi"
flightNumber: 2102145
departure: "10:00 AM"
arrival: "08:00 PM"
seats: 120
__v: 0
```

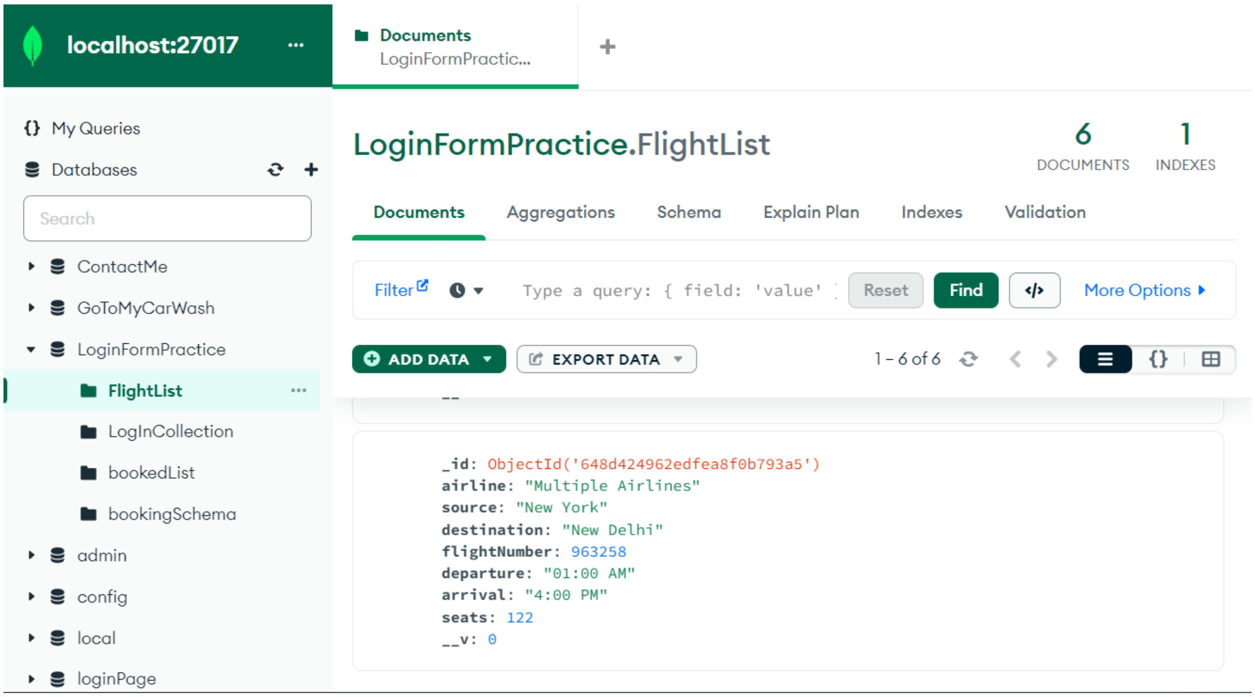


```
<!DOCTYPE html>
<html lang="en">

<head>
```



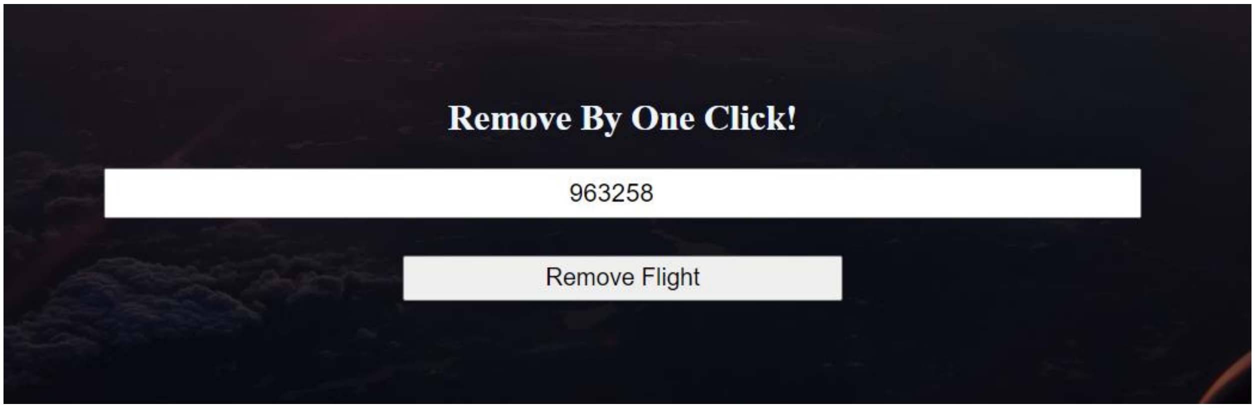
```
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/admin.css">
    <title>Document</title>
</head>

<body>
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="/admin">Home</a></li>
                <li class="item"><a href="">Add Flights</a></li>
                <li class="item"><a href="/admin/remove">Remove
Flights</a></li>
                <li class="item"><a href="/admin/booked">Booked
Flights</a></li>
                <li class="item"><a href="/logout">Logout</a></li>
            </ul>
        </nav>
    </header>

    <section id="home">
        <form action="/admin/add" method="post" id="addFlight">
            <h2>Add Flight By One Click!</h2>
            <input type="text" name="airline" placeholder="Airline" required>
            <input type="text" id="source" name="source" placeholder="Source"
required>
            <input type="text" name="destination" placeholder="Destination"
required>
            <input type="number" name="flightNumber" placeholder="Flight
Number" required>
            <input type="text" name="departure" placeholder="Departure Time"
required>

            <input type="text" name="arrival" placeholder="Arrival Time"
required>
            <input type="number" name="seats" placeholder="Number Of Seats">
            <br>
            <button type="submit">Add Flight</button>
        </form>
    </section>
</body>

</html>
```

3. Remove Flight From Database:
   remove.ejs



LoginFormPractice.FlightList

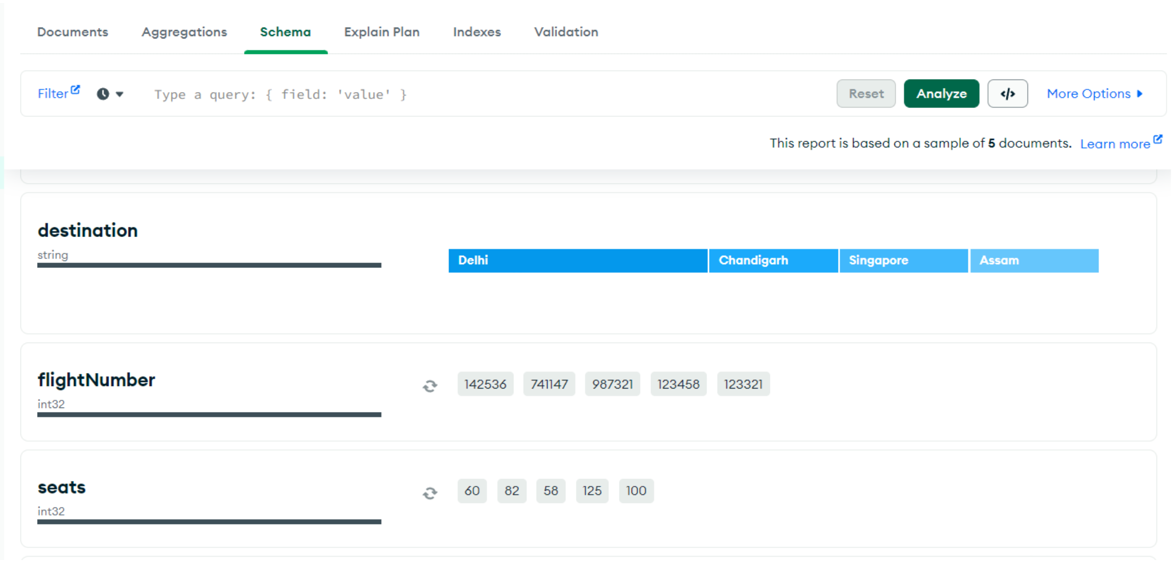
```
_id: ObjectId('648d424962edfea8f0b793a5')
airline: "Multiple Airlines"
source: "New York"
destination: "New Delhi"
flightNumber: 963258
departure: "01:00 AM"
arrival: "4:00 PM"
seats: 122
__v: 0
```



Remove By One Click!

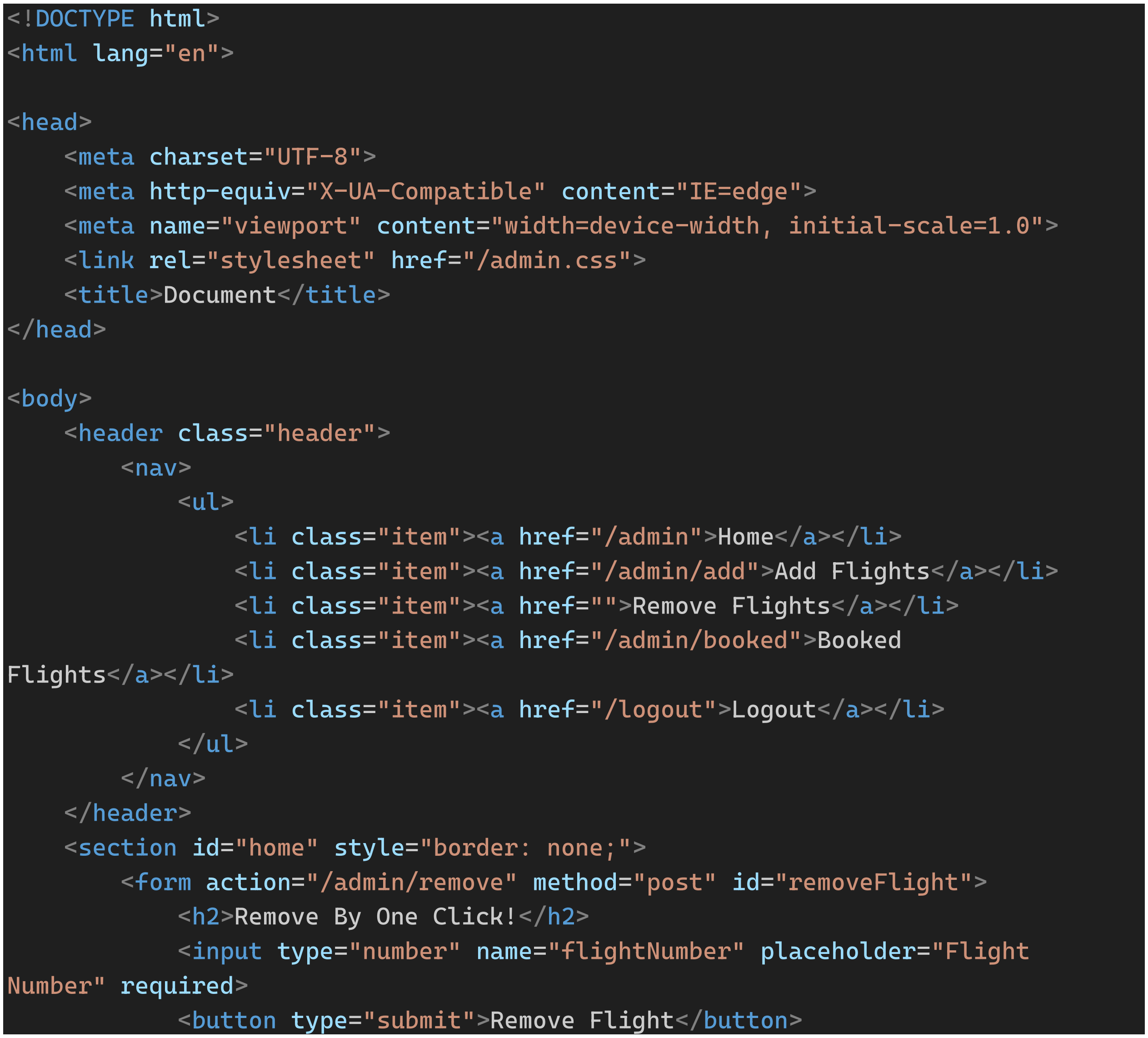
963258

Data is being deleted from database:



This report is based on a sample of **5** documents. Learn more

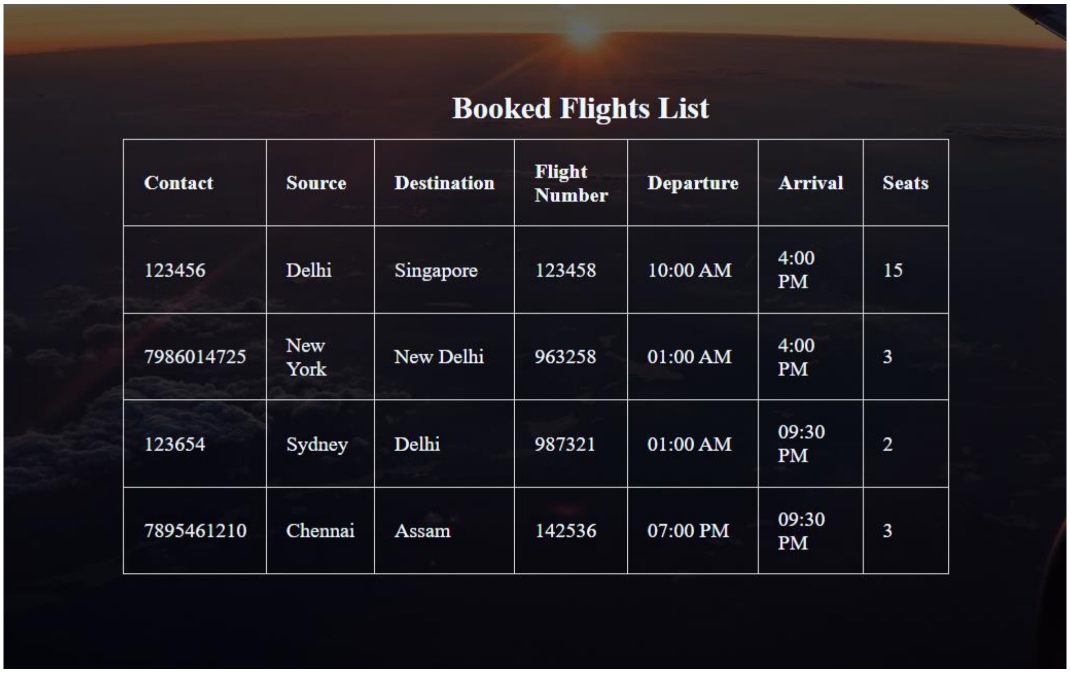
destination string	Delhi, Chandigarh, Singapore, Assam
flightNumber int32	142536, 741147, 987321, 123458, 123321
seats int32	60, 82, 58, 125, 100



```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/admin.css">
    <title>Document</title>
</head>

<body>
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="/admin">Home</a></li>
                <li class="item"><a href="/admin/add">Add Flights</a></li>
                <li class="item"><a href="">Remove Flights</a></li>
                <li class="item"><a href="/admin/booked">Booked
Flights</a></li>
                <li class="item"><a href="/logout">Logout</a></li>
            </ul>
        </nav>
    </header>
    <section id="home" style="border: none;">
        <form action="/admin/remove" method="post" id="removeFlight">
            <h2>Remove By One Click!</h2>
            <input type="number" name="flightNumber" placeholder="Flight
Number" required>
            <button type="submit">Remove Flight</button>
        </form>
        </div>
</body>

</html>
```

4. *Booked Flights:*
   booked.ejs

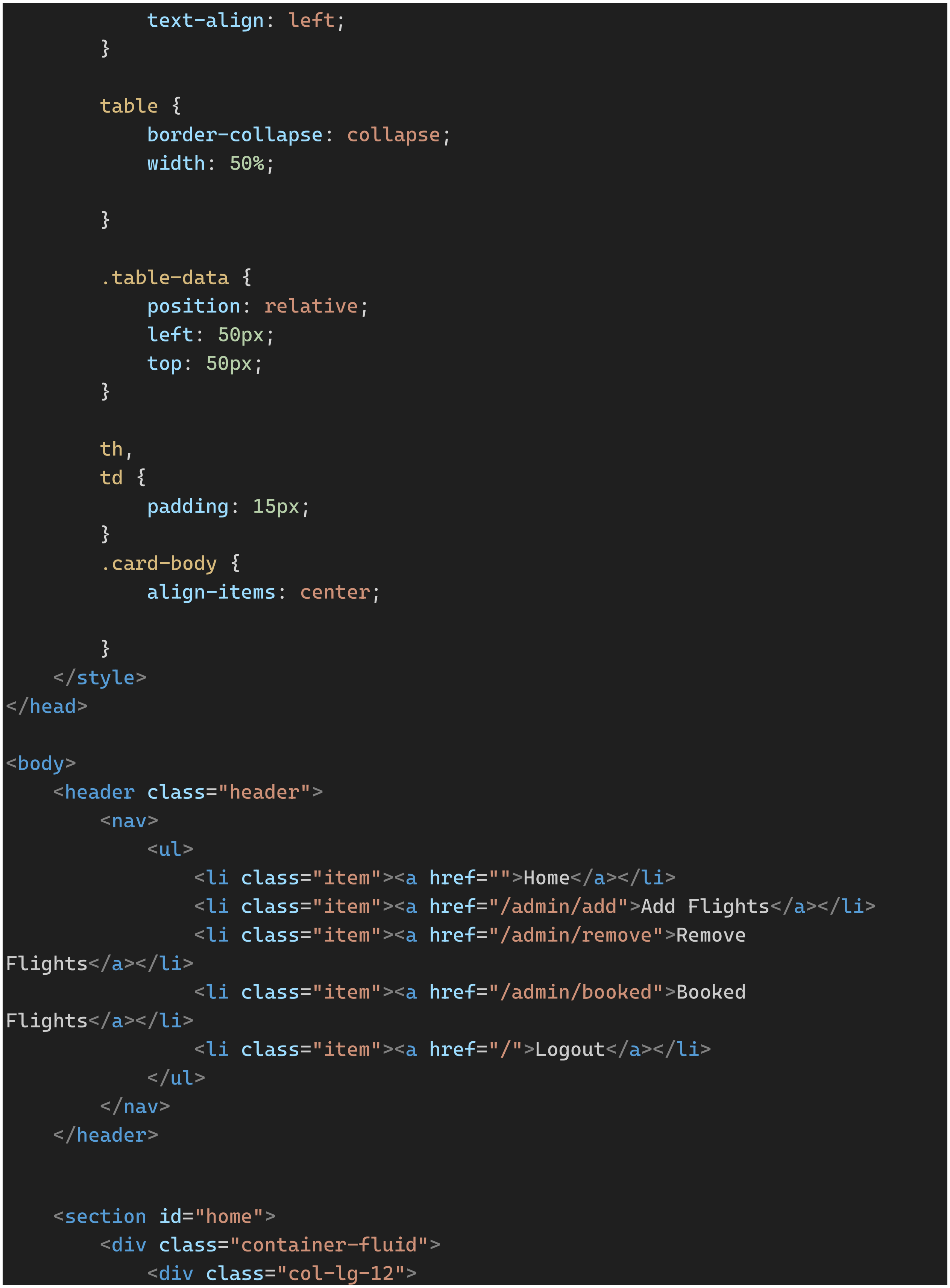
   Display's all booked flights



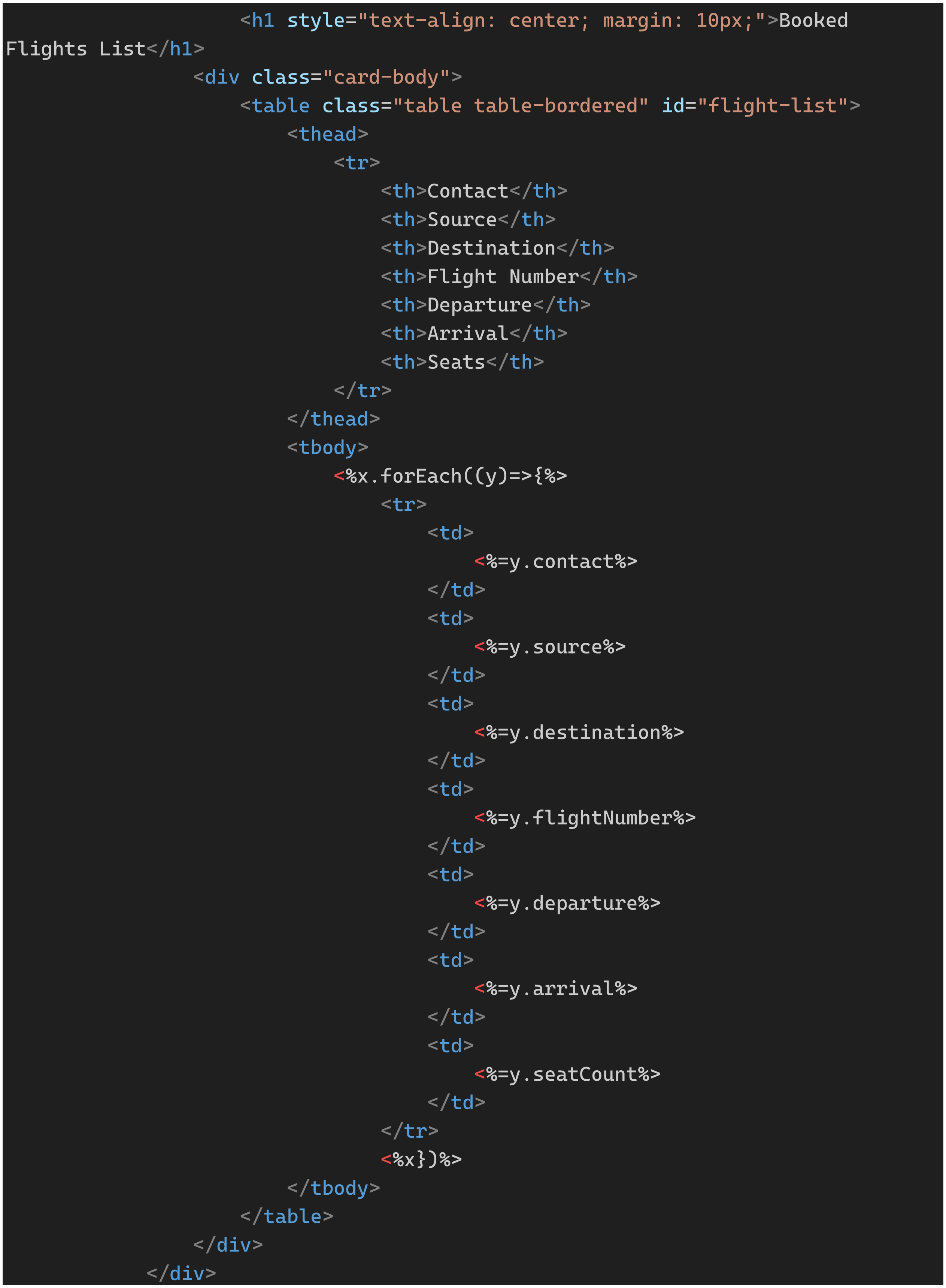
Booked Flights List

Contact	Source	Destination	Flight Number	Departure	Arrival	Seats
123456	Delhi	Singapore	123458	10:00 AM	4:00 PM	15
7986014725	New York	New Delhi	963258	01:00 AM	4:00 PM	3
123654	Sydney	Delhi	987321	01:00 AM	09:30 PM	2
7895461210	Chennai	Assam	142536	07:00 PM	09:30 PM	3

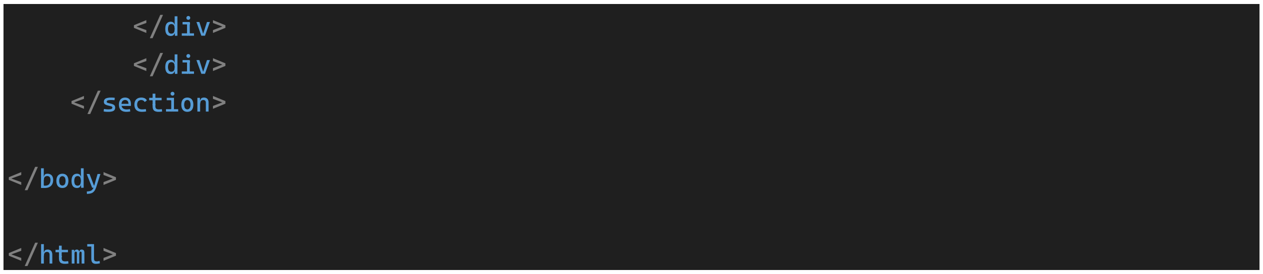
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/admin.css">
    <title>Booked Flights</title>
    <style type="text/css">
        table,
        td,
        th {
            border: 1px solid #ddd;
```



```
            text-align: left;
        }

        table {
            border-collapse: collapse;
            width: 50%;

        }

        .table-data {
            position: relative;
            left: 50px;
            top: 50px;
        }

        th,
        td {
            padding: 15px;
        }
        .card-body {
            align-items: center;

        }
    </style>
</head>

<body>
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="">Home</a></li>
                <li class="item"><a href="/admin/add">Add Flights</a></li>
                <li class="item"><a href="/admin/remove">Remove
Flights</a></li>
                <li class="item"><a href="/admin/booked">Booked
Flights</a></li>
                <li class="item"><a href="/">Logout</a></li>
            </ul>
        </nav>
    </header>


    <section id="home">
        <div class="container-fluid">
            <div class="col-lg-12">
```



```
                    <h1 style="text-align: center; margin: 10px;">Booked
Flights List</h1>
                <div class="card-body">
                    <table class="table table-bordered" id="flight-list">
                        <thead>
                            <tr>
                                <th>Contact</th>
                                <th>Source</th>
                                <th>Destination</th>
                                <th>Flight Number</th>
                                <th>Departure</th>
                                <th>Arrival</th>
                                <th>Seats</th>
                            </tr>
                        </thead>
                        <tbody>
                            <%x.forEach((y)=>{%>
                                <tr>
                                    <td>
                                        <%=y.contact%>
                                    </td>
                                    <td>
                                        <%=y.source%>
                                    </td>
                                    <td>
                                        <%=y.destination%>
                                    </td>
                                    <td>
                                        <%=y.flightNumber%>
                                    </td>
                                    <td>
                                        <%=y.departure%>
                                    </td>
                                    <td>
                                        <%=y.arrival%>
                                    </td>
                                    <td>
                                        <%=y.seatCount%>
                                    </td>
                                </tr>
                                <%x})%>
                        </tbody>
                    </table>
                </div>
            </div>
```

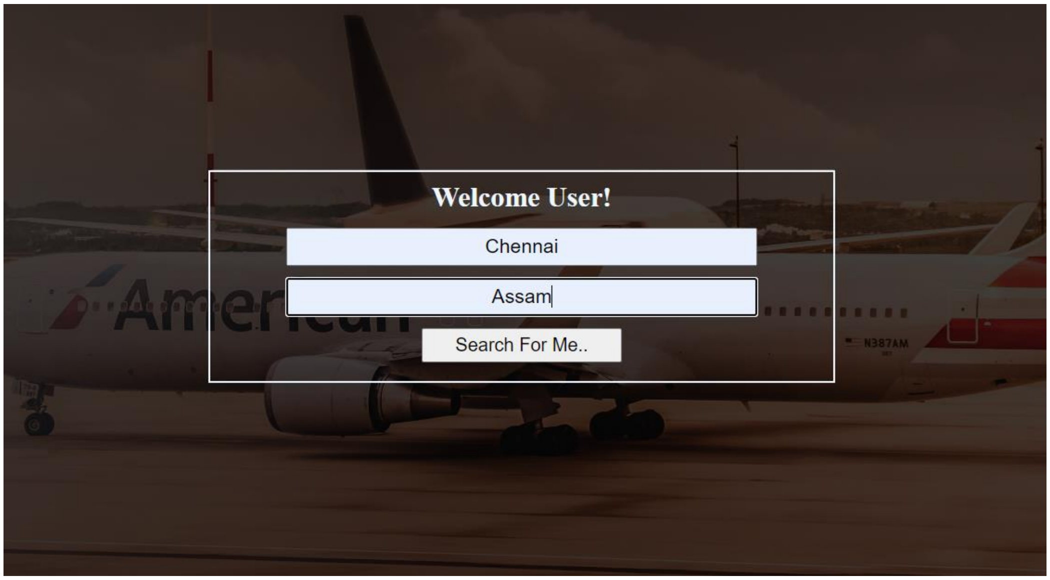


```
        </div>
        </div>
    </section>

</body>

</html>
```

User Use Cases:

1. Home Page:
   user.ejs

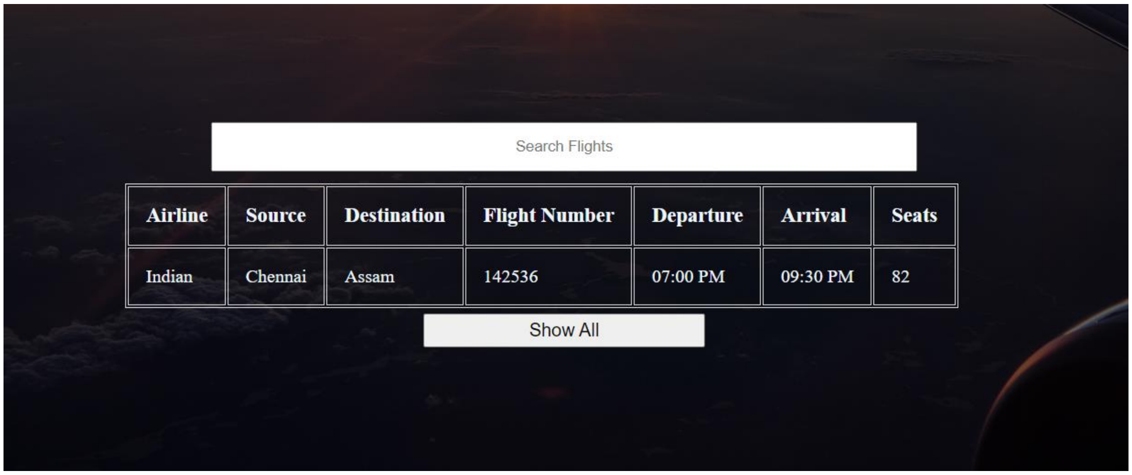
   Work as a searching option for a user.



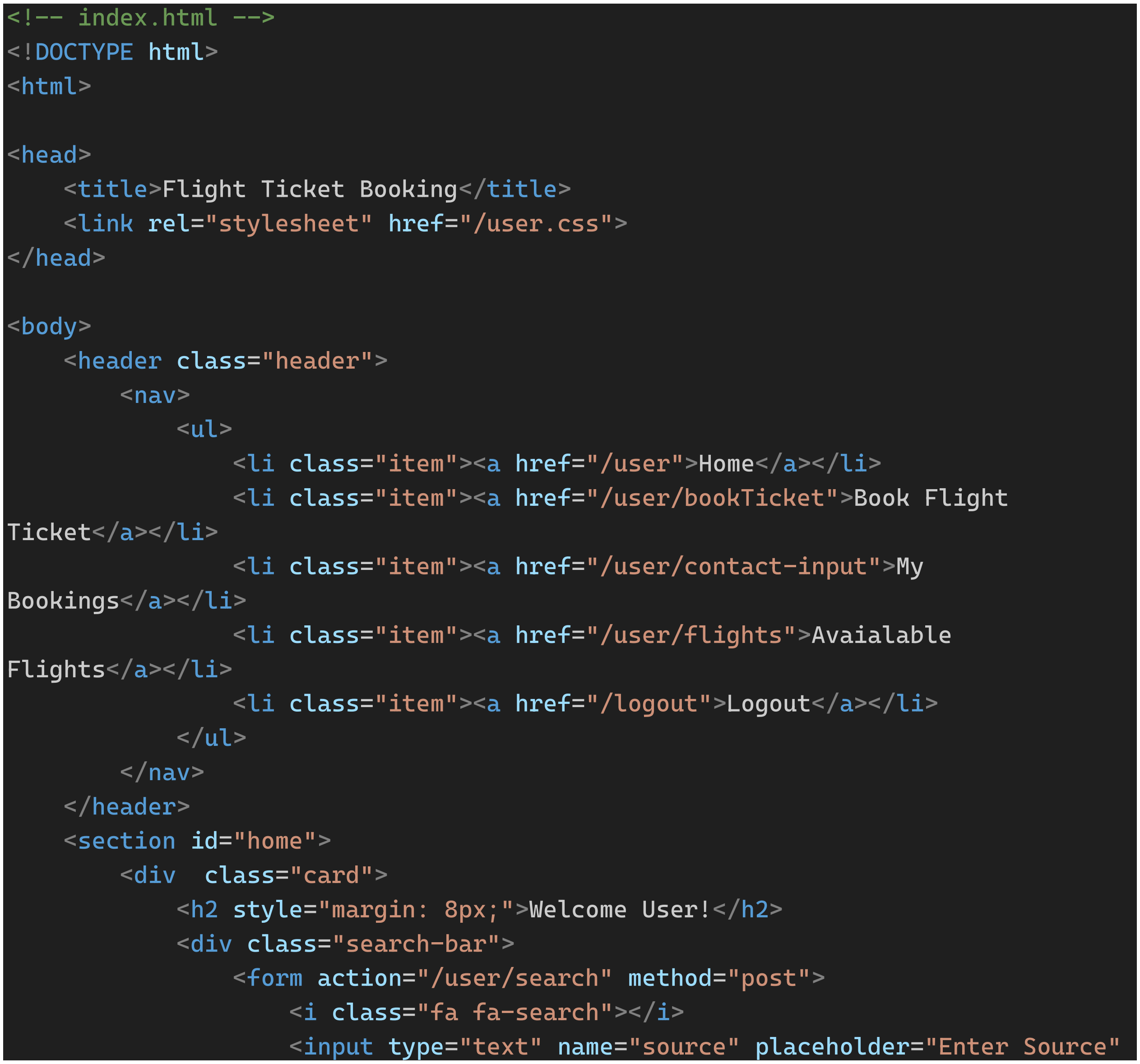
Welcome User!

Chennai

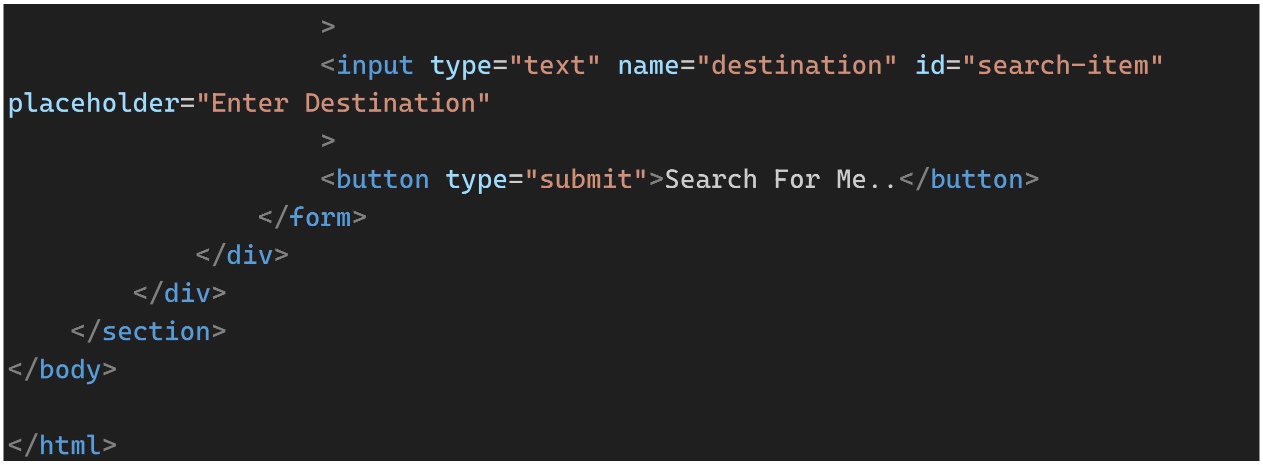
Assam



Airline	Source	Destination	Flight Number	Departure	Arrival	Seats
Indian	Chennai	Assam	142536	07:00 PM	09:30 PM	82

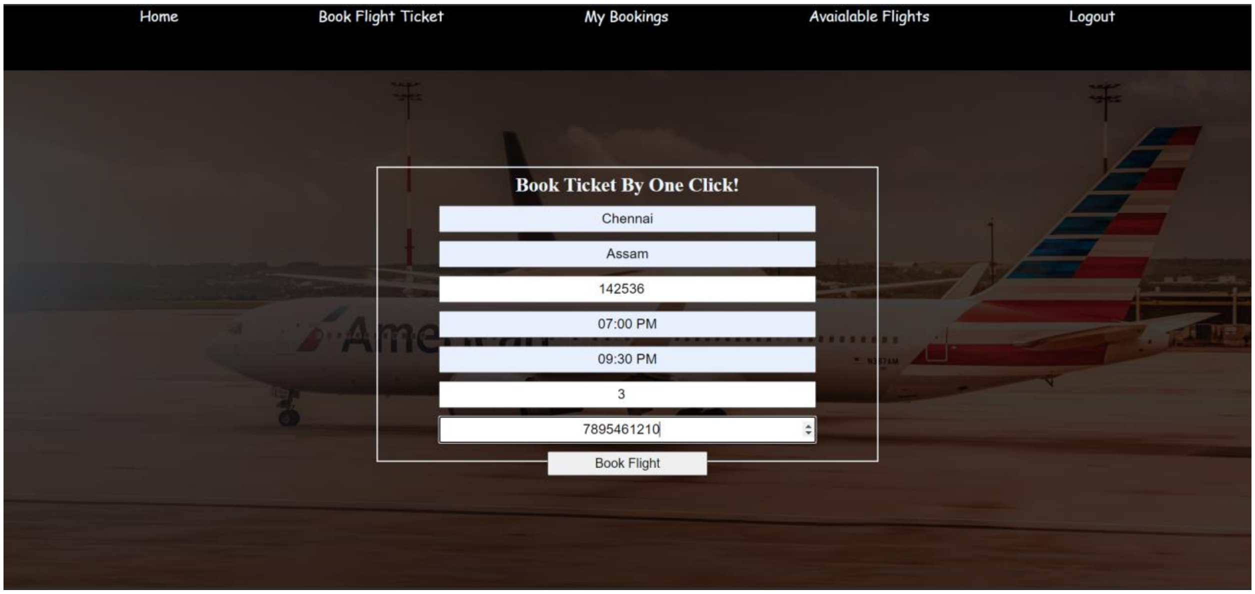


```
<!-- index.html -->
<!DOCTYPE html>
<html>

<head>
    <title>Flight Ticket Booking</title>
    <link rel="stylesheet" href="/user.css">
</head>

<body>
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="/user">Home</a></li>
                <li class="item"><a href="/user/bookTicket">Book Flight
Ticket</a></li>
                <li class="item"><a href="/user/contact-input">My
Bookings</a></li>
                <li class="item"><a href="/user/flights">Avaialable
Flights</a></li>
                <li class="item"><a href="/logout">Logout</a></li>
            </ul>
        </nav>
    </header>
    <section id="home">
        <div  class="card">
            <h2 style="margin: 8px;">Welcome User!</h2>
            <div class="search-bar">
                <form action="/user/search" method="post">
                    <i class="fa fa-search"></i>
                    <input type="text" name="source" placeholder="Enter Source"
```



```
                    >
                    <input type="text" name="destination" id="search-item"
placeholder="Enter Destination"
                    >
                    <button type="submit">Search For Me..</button>
                </form>
            </div>
        </div>
    </section>
</body>

</html>
```

2. Book Flight Ticket
   bookTicket.ejs



Book Ticket By One Click!

Chennai

Assam

142536

07:00 PM

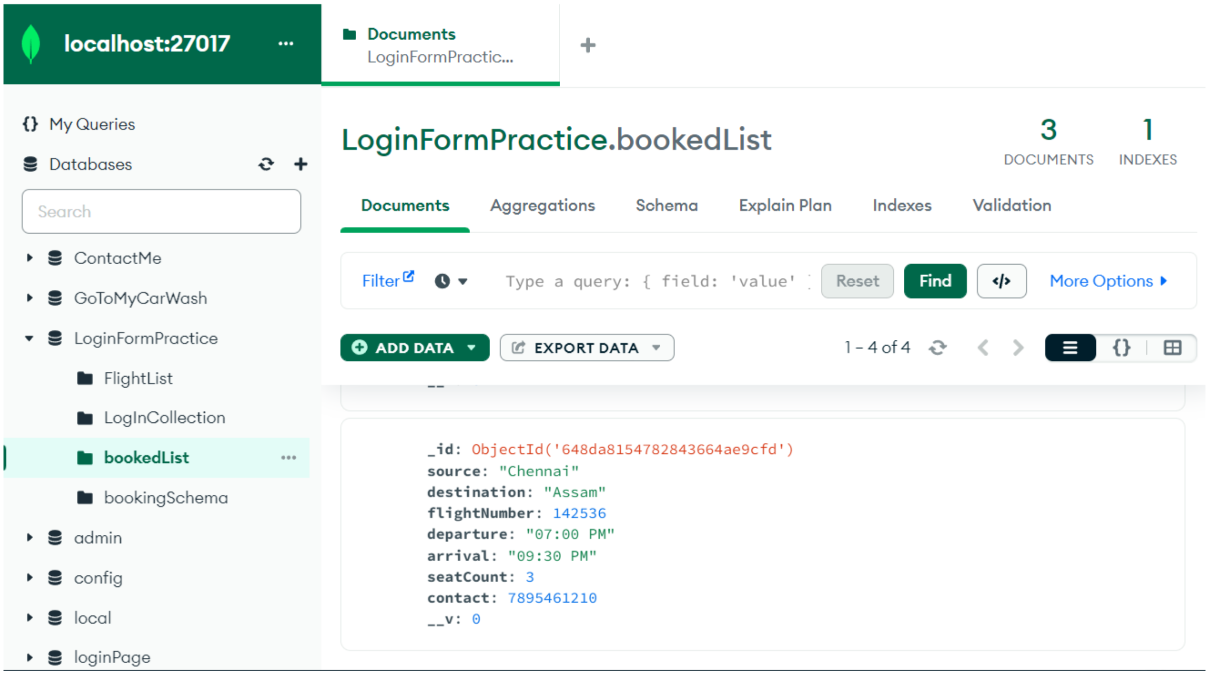
09:30 PM

3

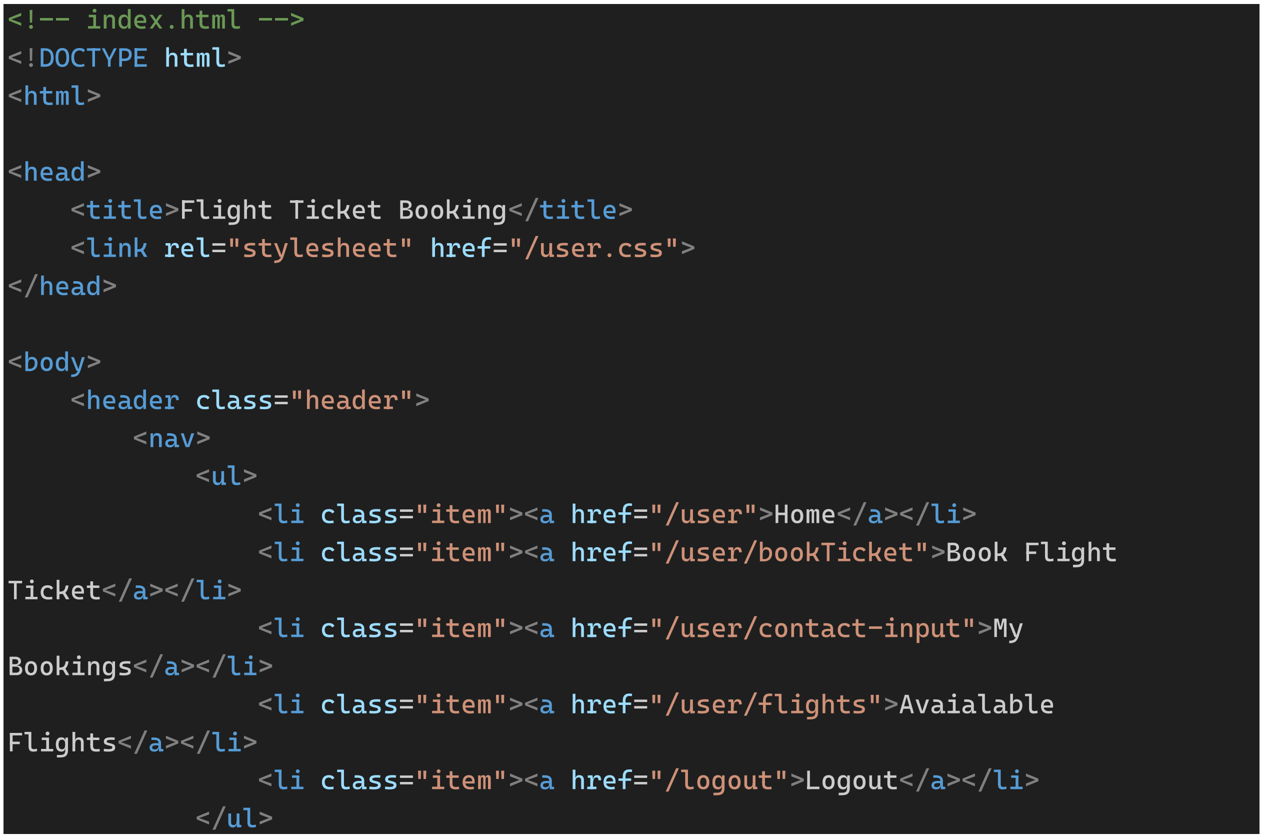
7895461210

After clicking "Book Flight", it will look for flight:

```
If (available(seats)), then {
        1. Store flight information In database
        2. Subtract the number of seats occupied
}
Else {
        Display error
}
```

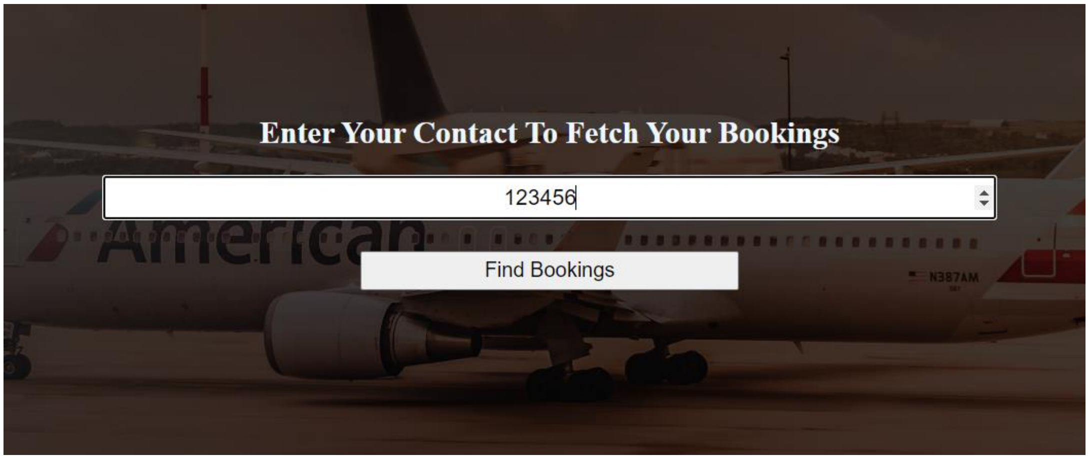


```
_id: ObjectId('648da8154782843664ae9cfd')
source: "Chennai"
destination: "Assam"
flightNumber: 142536
departure: "07:00 PM"
arrival: "09:30 PM"
seatCount: 3
contact: 7895461210
__v: 0
```



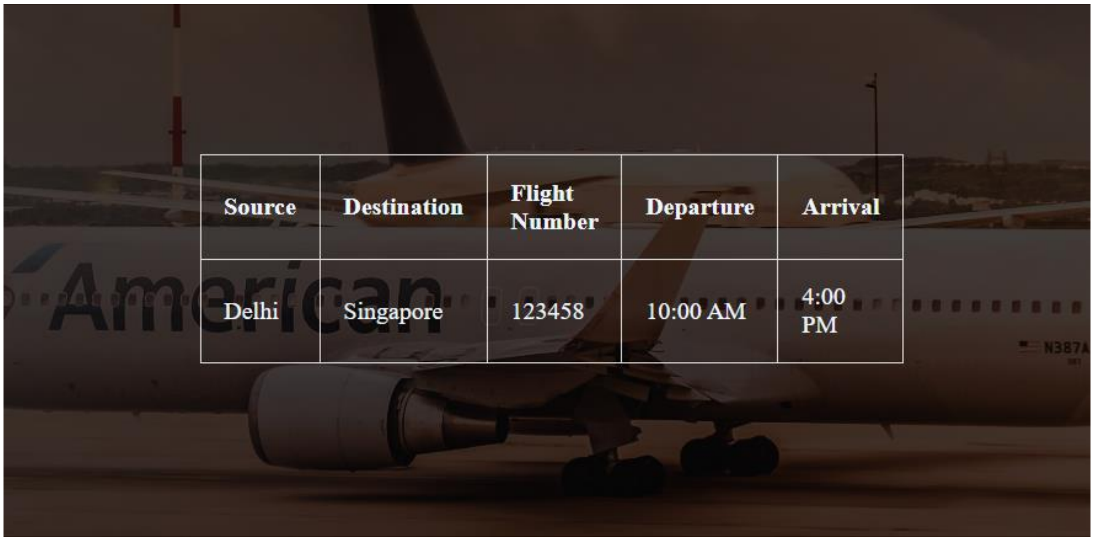
```
<!-- index.html -->
<!DOCTYPE html>
<html>

<head>
    <title>Flight Ticket Booking</title>
    <link rel="stylesheet" href="/user.css">
</head>

<body>
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="/user">Home</a></li>
                <li class="item"><a href="/user/bookTicket">Book Flight
Ticket</a></li>
                <li class="item"><a href="/user/contact-input">My
Bookings</a></li>
                <li class="item"><a href="/user/flights">Avaialable
Flights</a></li>
                <li class="item"><a href="/logout">Logout</a></li>
            </ul>
        </nav>
    </header>
    <section id="home">
        <form action="/user/bookTicket" method="post" id="bookTicket">
            <h2>Book Ticket By One Click!</h2>

            <input type="text" id="source" name="source" placeholder="Source" required>
            <br>
            <input type="text" name="destination" placeholder="Destination" required>
            <br>
            <input type="number" name="flightNumber" placeholder="Flight Number" required>
            <br>
            <input type="text" name="departure" placeholder="Departure Time" required>
            <br>
            <input type="text" name="arrival" placeholder="Arrival Time" required>
            <br>
            <input type="number" name="seatCount" placeholder="Number Of Seats Required" required>
            <br>
            <input type="number" name="contact" placeholder="Contact Number" required>
            <br>
            <button type="submit">Book Flight</button>
        </form>
    </section>
</body>
</html>
```

3. My Bookings
   mybooking.ejs



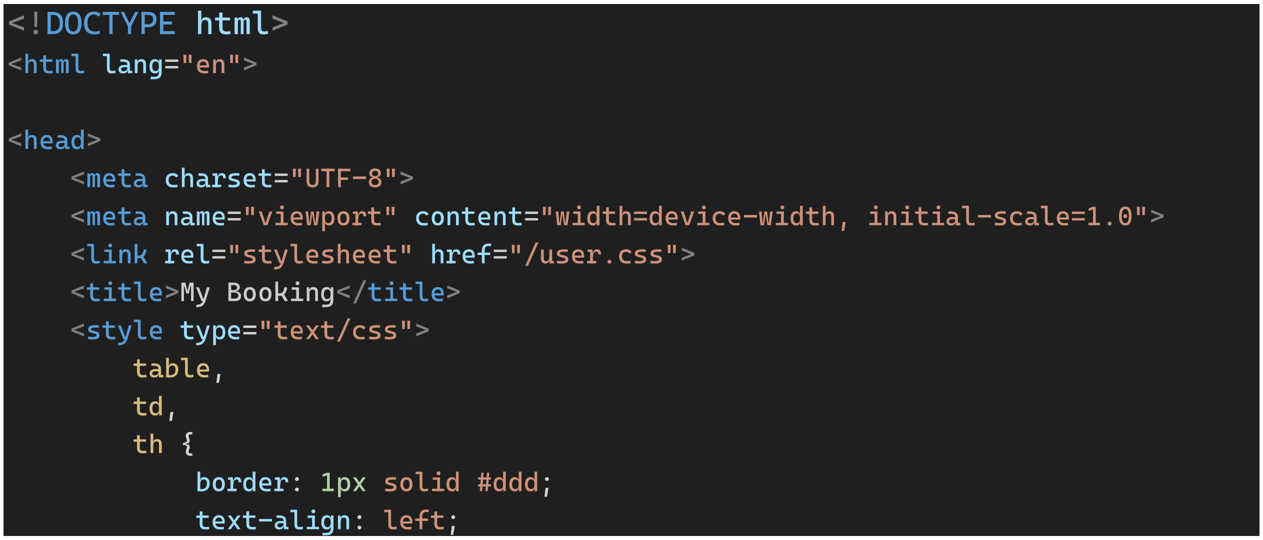
Enter Your Contact To Fetch Your Bookings

123456

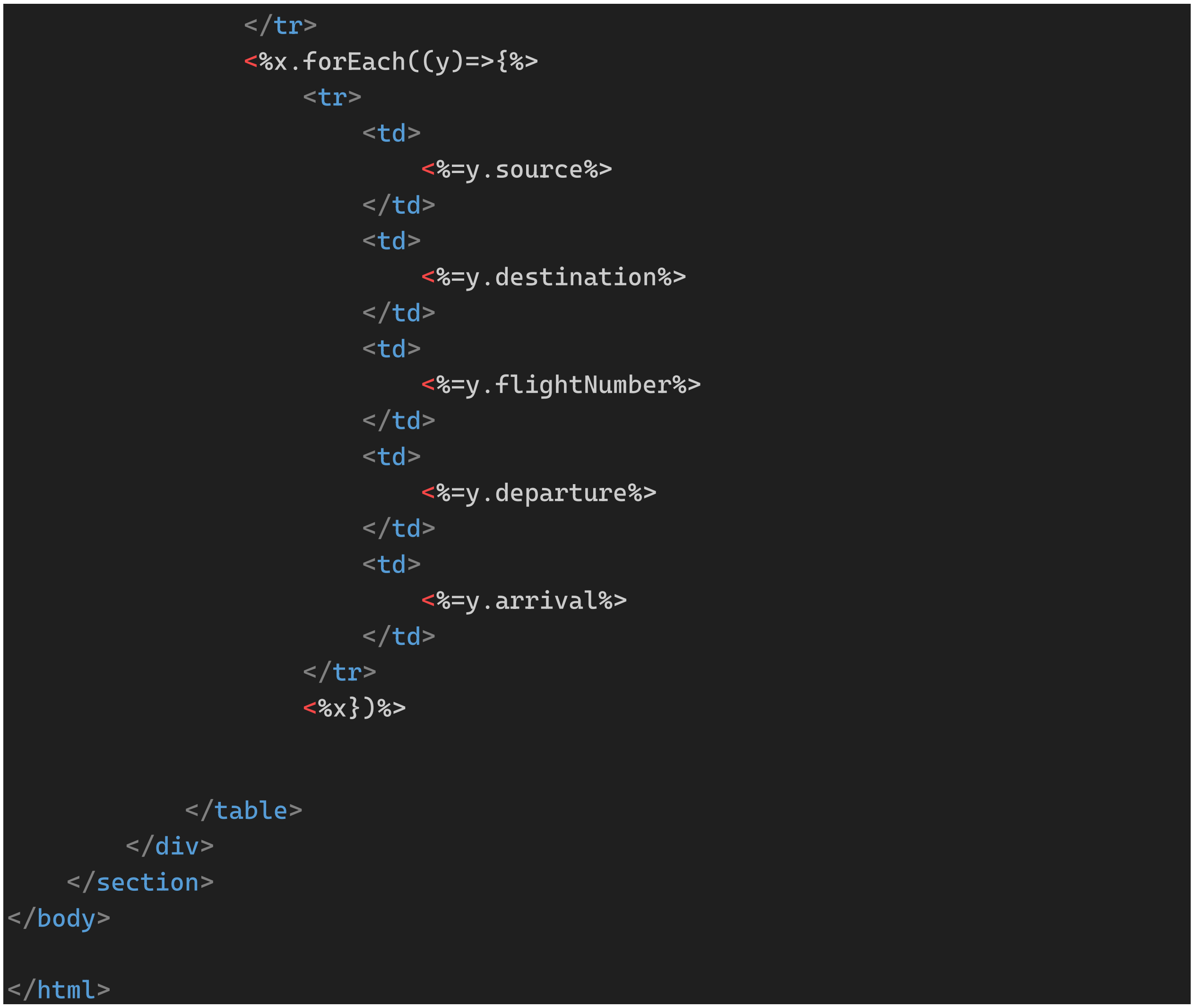


Source	Destination	Flight Number	Departure	Arrival
Delhi	Singapore	123458	10:00 AM	4:00 PM

Code:

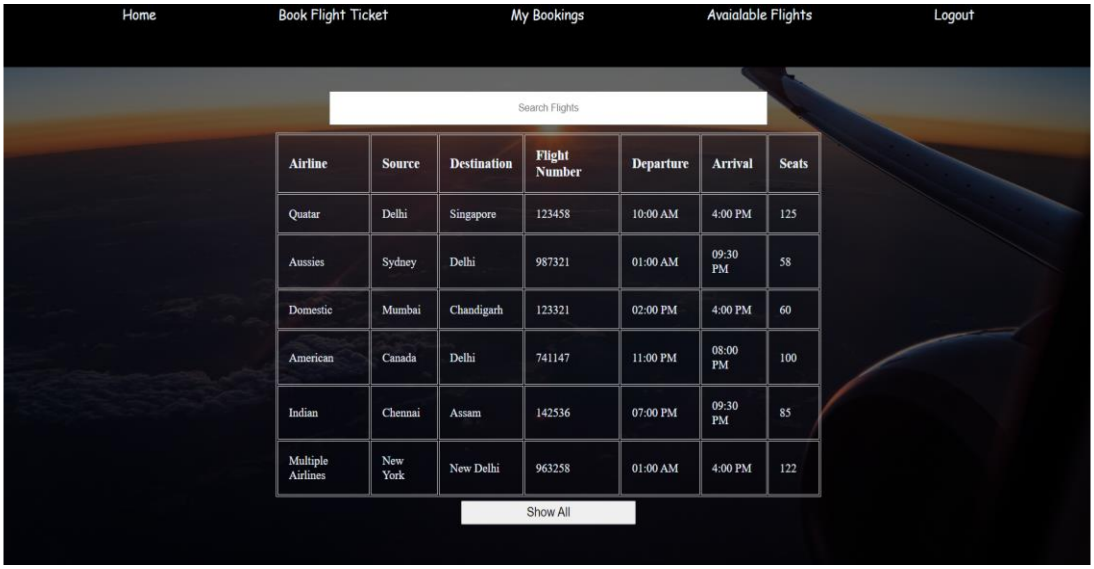


```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/user.css">
    <title>My Booking</title>
    <style type="text/css">
        table,
        td,
        th {
            border: 1px solid #ddd;
            text-align: left;
        }

        table {
            border-collapse: collapse;
            width: 50%;

        }

        .table-data {
            position: relative;
            left: 50px;
            top: 50px;
        }

        th,
        td {
            padding: 15px;
        }
    </style>
</head>

<body>
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="/user">Home</a></li>
                <li class="item"><a href="/user/bookTicket">Book Flight
Ticket</a></li>
                <li class="item"><a href="/user/contact-input">My
Bookings</a></li>
                <li class="item"><a href="/user/flights">Avaialable
Flights</a></li>
                <li class="item"><a href="/logout">Logout</a></li>
            </ul>
        </nav>
    </header>
    <section id="home">
        <div class="mybooking">
            <table>
                <tr>
                    <th>Source</th>
                    <th>Destination</th>
                    <th>Flight Number</th>
                    <th>Departure</th>
                    <th>Arrival</th>
```



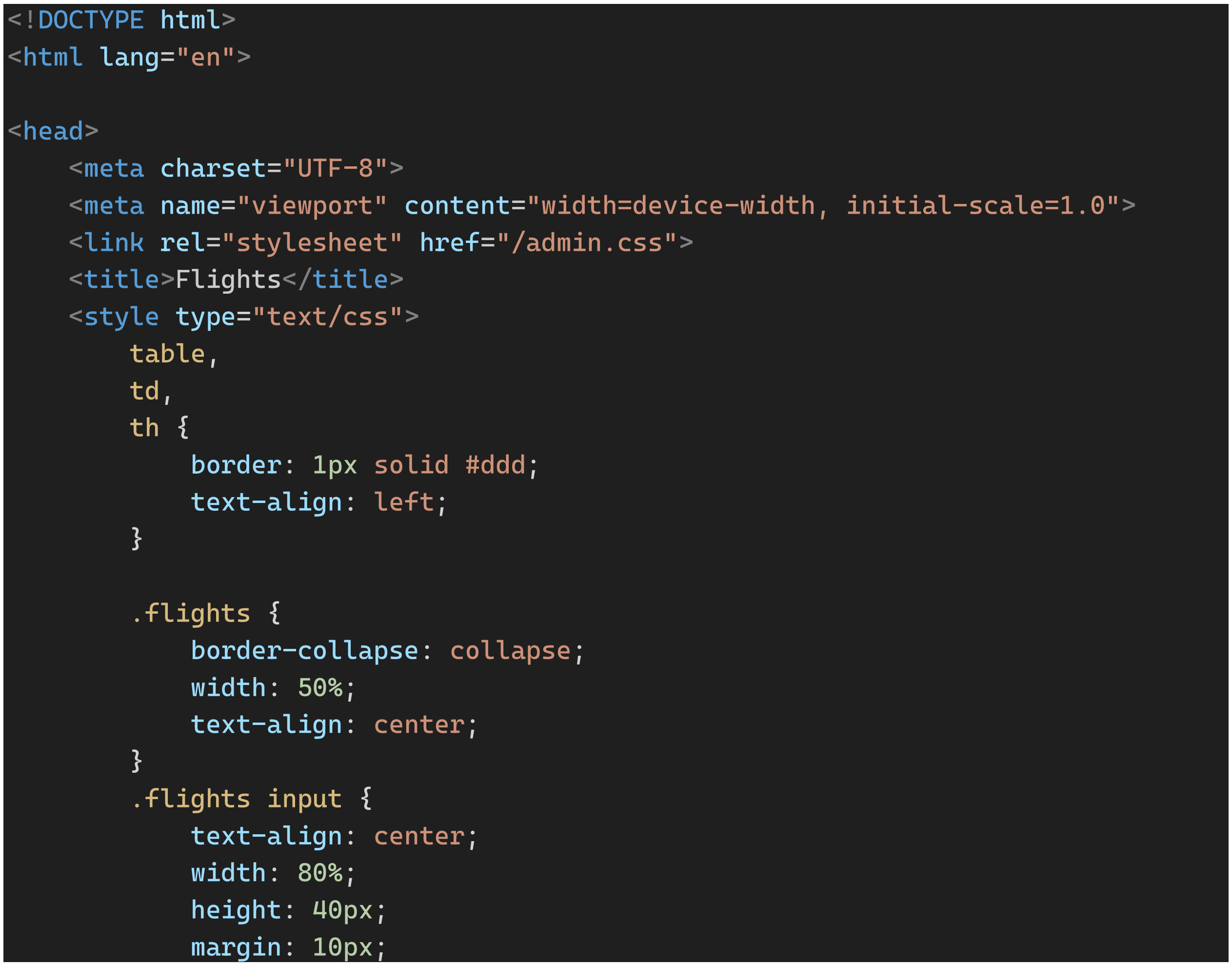
```
                </tr>
                <%x.forEach((y)=>{%>
                    <tr>
                        <td>
                            <%=y.source%>
                        </td>
                        <td>
                            <%=y.destination%>
                        </td>
                        <td>
                            <%=y.flightNumber%>
                        </td>
                        <td>
                            <%=y.departure%>
                        </td>
                        <td>
                            <%=y.arrival%>
                        </td>
                    </tr>
                    <%x})%>


            </table>
        </div>
    </section>
</body>

</html>
```

4. Available Flights
   flights.ejs

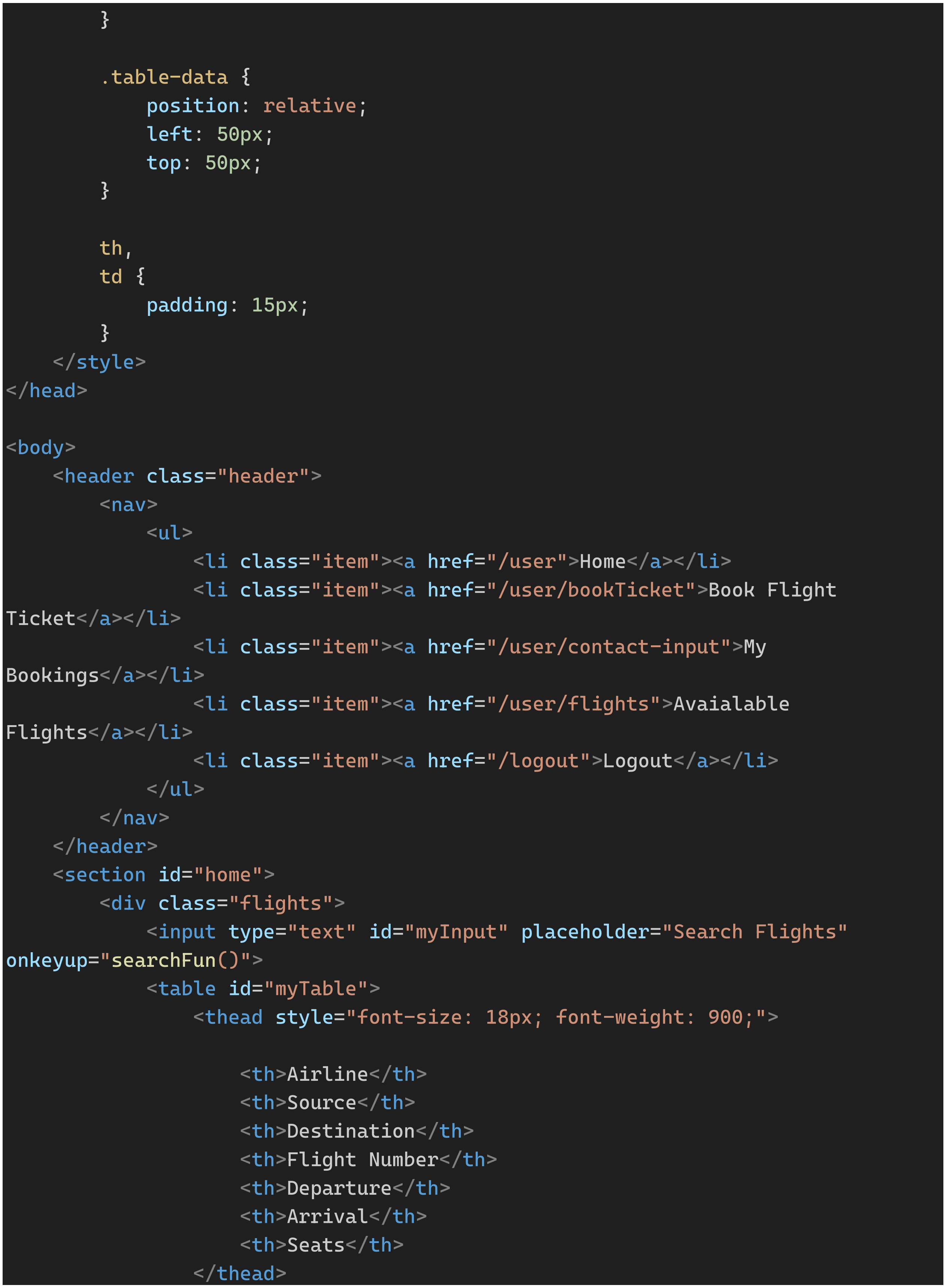


Airline	Source	Destination	Flight Number	Departure	Arrival	Seats
Quatar	Delhi	Singapore	123458	10:00 AM	4:00 PM	125
Aussies	Sydney	Delhi	987321	01:00 AM	09:30 PM	58
Domestic	Mumbai	Chandigarh	123321	02:00 PM	4:00 PM	60
American	Canada	Delhi	741147	11:00 PM	08:00 PM	100
Indian	Chennai	Assam	142536	07:00 PM	09:30 PM	85
Multiple Airlines	New York	New Delhi	963258	01:00 AM	4:00 PM	122

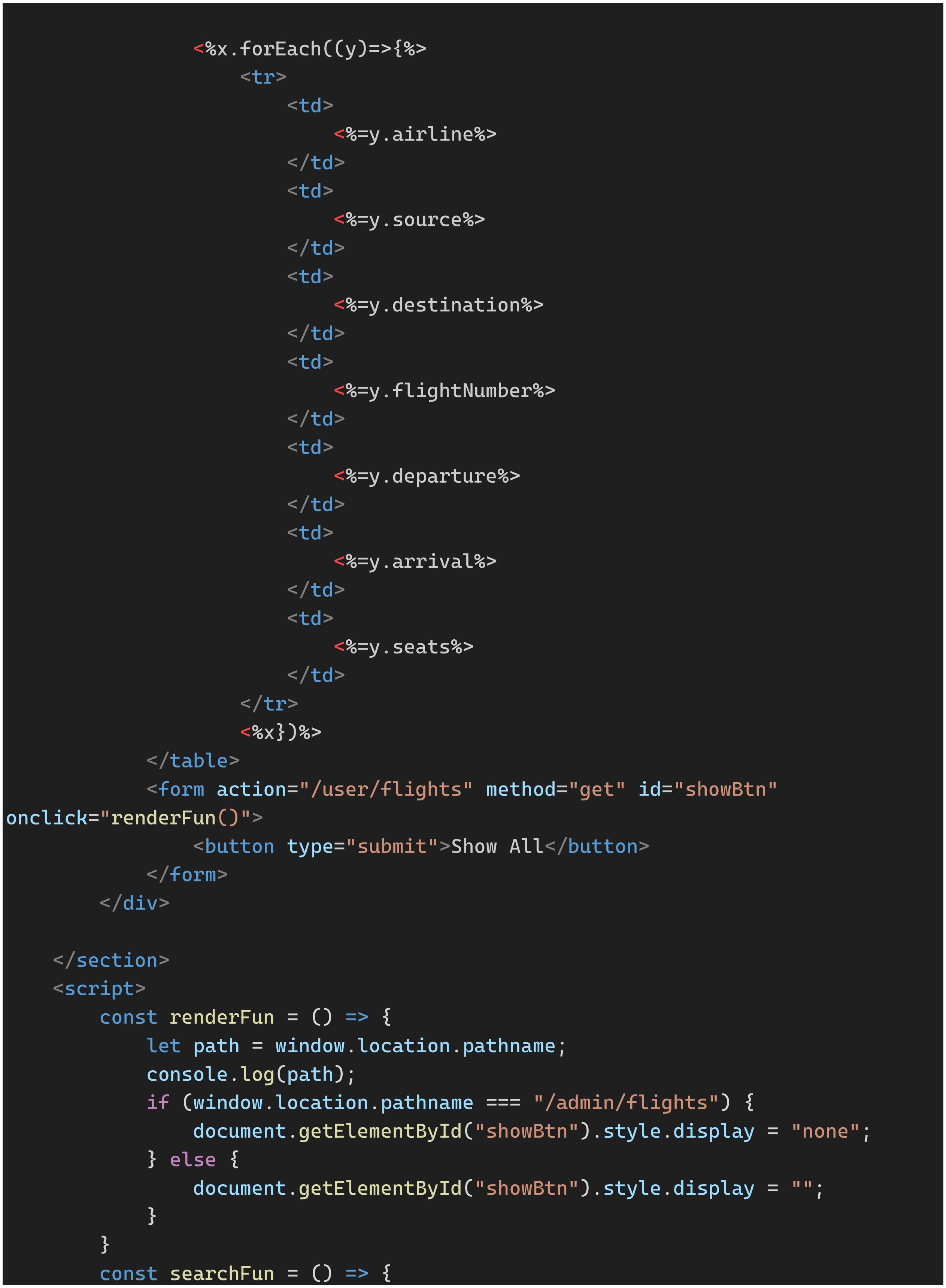
Code:



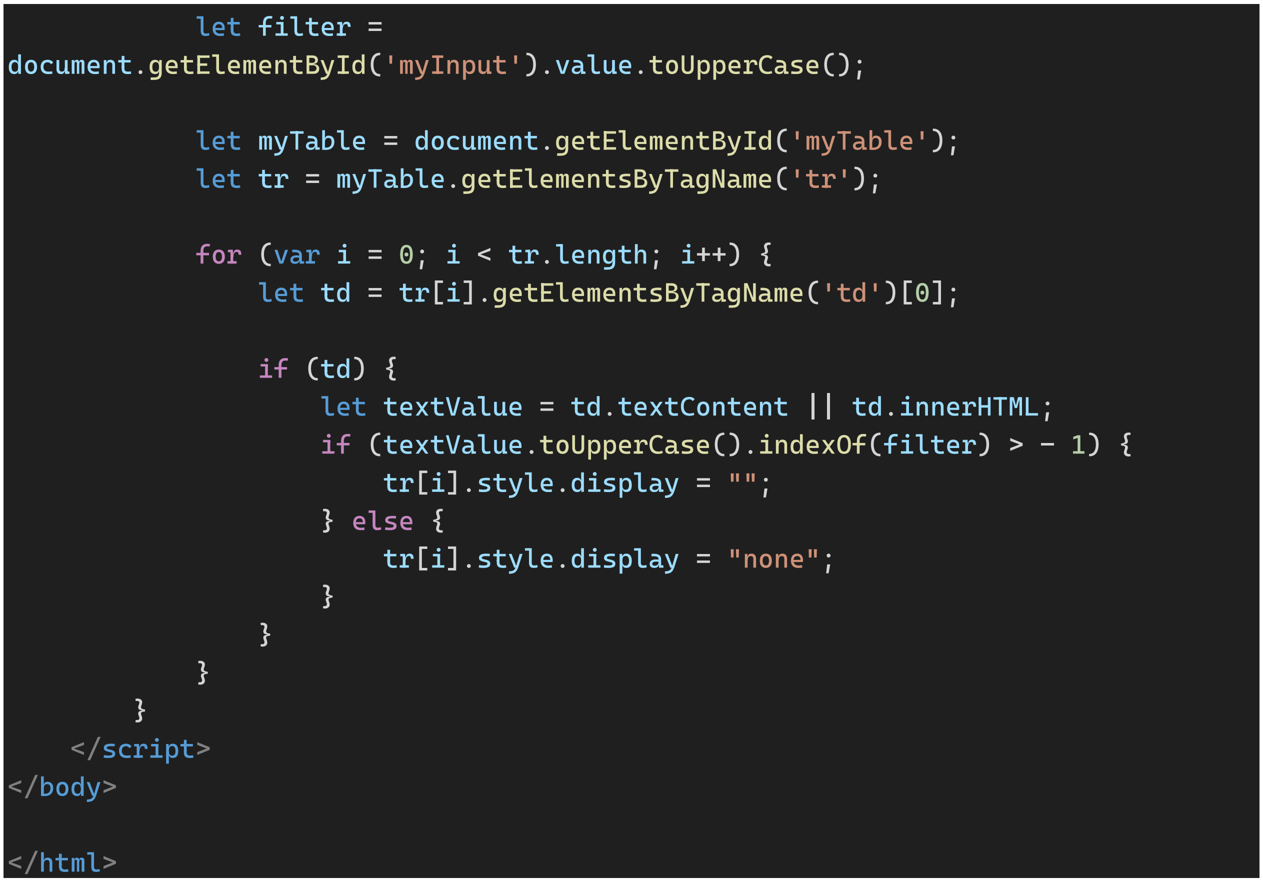
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="/admin.css">
    <title>Flights</title>
    <style type="text/css">
        table,
        td,
        th {
            border: 1px solid #ddd;
            text-align: left;
        }

        .flights {
            border-collapse: collapse;
            width: 50%;
            text-align: center;
        }
        .flights input {
            text-align: center;
            width: 80%;
            height: 40px;
            margin: 10px;
```



```
        }

        .table-data {
            position: relative;
            left: 50px;
            top: 50px;
        }

        th,
        td {
            padding: 15px;
        }
    </style>
</head>

<body>
    <header class="header">
        <nav>
            <ul>
                <li class="item"><a href="/user">Home</a></li>
                <li class="item"><a href="/user/bookTicket">Book Flight
Ticket</a></li>
                <li class="item"><a href="/user/contact-input">My
Bookings</a></li>
                <li class="item"><a href="/user/flights">Avaialable
Flights</a></li>
                <li class="item"><a href="/logout">Logout</a></li>
            </ul>
        </nav>
    </header>
    <section id="home">
        <div class="flights">
            <input type="text" id="myInput" placeholder="Search Flights"
onkeyup="searchFun()">
            <table id="myTable">
                <thead style="font-size: 18px; font-weight: 900;">

                    <th>Airline</th>
                    <th>Source</th>
                    <th>Destination</th>
                    <th>Flight Number</th>
                    <th>Departure</th>
                    <th>Arrival</th>
                    <th>Seats</th>
                </thead>
```



```
                <%x.forEach((y)=>{%>
                    <tr>
                        <td>
                            <%=y.airline%>
                        </td>
                        <td>
                            <%=y.source%>
                        </td>
                        <td>
                            <%=y.destination%>
                        </td>
                        <td>
                            <%=y.flightNumber%>
                        </td>
                        <td>
                            <%=y.departure%>
                        </td>
                        <td>
                            <%=y.arrival%>
                        </td>
                        <td>
                            <%=y.seats%>
                        </td>
                    </tr>
                    <%x})%>
            </table>
            <form action="/user/flights" method="get" id="showBtn"
onclick="renderFun()">
                <button type="submit">Show All</button>
            </form>
        </div>

    </section>
    <script>
        const renderFun = () => {
            let path = window.location.pathname;
            console.log(path);
            if (window.location.pathname === "/admin/flights") {
                document.getElementById("showBtn").style.display = "none";
            } else {
                document.getElementById("showBtn").style.display = "";
            }
        }
        const searchFun = () => {
```



```
            let filter =
document.getElementById('myInput').value.toUpperCase();

            let myTable = document.getElementById('myTable');
            let tr = myTable.getElementsByTagName('tr');

            for (var i = 0; i < tr.length; i++) {
                let td = tr[i].getElementsByTagName('td')[0];

                if (td) {
                    let textValue = td.textContent || td.innerHTML;
                    if (textValue.toUpperCase().indexOf(filter) > - 1) {
                        tr[i].style.display = "";
                    } else {
                        tr[i].style.display = "none";
                    }
                }
            }
        }
    </script>
</body>

</html>
```

Result:

Made a full stack application for flight ticket booking using HTML, CSS, JS, NodeJs, Express.Js and MongoDB.

Future Scope:

1. Separate User Database can be formed for handling unique user for each server.
2. Application can be hosted to the browser.
3. Searching Flight can be done according to Time and Date.