

FlightFinder: Navigating Your Air Travel Options

1. Database Schema (ER Diagram SQL)

```
CREATE TABLE Users (  
  user_id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(100),  
  email VARCHAR(100) UNIQUE,  
  password VARCHAR(100)  
);  
  
CREATE TABLE Airports (  
  airport_id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(100),  
  city VARCHAR(100),  
  country VARCHAR(100)  
);  
  
CREATE TABLE Flights (  
  flight_id INT PRIMARY KEY AUTO_INCREMENT,  
  flight_number VARCHAR(20),  
  departure_airport_id INT,  
  arrival_airport_id INT,  
  departure_time DATETIME,  
  arrival_time DATETIME,  
  price DECIMAL(10,2),  
  FOREIGN KEY (departure_airport_id) REFERENCES Airports(airport_id),  
  FOREIGN KEY (arrival_airport_id) REFERENCES Airports(airport_id)  
);  
  
CREATE TABLE Bookings (  
  booking_id INT PRIMARY KEY AUTO_INCREMENT,  
  user_id INT,  
  flight_id INT,  
  booking_date DATETIME,  
  status VARCHAR(50),  
  FOREIGN KEY (user_id) REFERENCES Users(user_id),  
  FOREIGN KEY (flight_id) REFERENCES Flights(flight_id)  
);
```

2. Backend Code (Node.js)

```
const express = require('express');  
const mysql = require('mysql2');  
const app = express();  
const cors = require('cors');  
app.use(cors());  
app.use(express.json());  
  
const db = mysql.createConnection({  
  host: 'localhost',  
  user: 'root',
```

```

    password: '',
    database: 'flight_finder'
  });

app.get('/flights', (req, res) => {
  const query = `
    SELECT f.flight_id, f.flight_number, a1.name AS departure, a2.name AS arrival,
           f.departure_time, f.arrival_time, f.price
    FROM Flights f
    JOIN Airports a1 ON f.departure_airport_id = a1.airport_id
    JOIN Airports a2 ON f.arrival_airport_id = a2.airport_id
  `;
  db.query(query, (err, results) => {
    if (err) res.status(500).json({ error: err });
    else res.json(results);
  });
});

app.listen(3000, () => {
  console.log('Server running on http://localhost:3000');
});

```

3. Frontend Code (HTML)

```

<!DOCTYPE html>
<html>
<head>
  <title>Flight Finder</title>
</head>
<body>
  <h1>Available Flights</h1>
  <table border="1" id="flightTable"></table>

  <script>
    fetch('http://localhost:3000/flights')
    .then(res => res.json())
    .then(data => {
      const table = document.getElementById('flightTable');
      table.innerHTML = `
        <tr>
          <th>Flight No</th><th>From</th><th>To</th>
          <th>Departure</th><th>Arrival</th><th>Price</th>
        </tr>
      `;
      data.forEach(f => {
        table.innerHTML += `
          <tr>
            <td>${f.flight_number}</td>
            <td>${f.departure}</td>
            <td>${f.arrival}</td>
            <td>${f.departure_time}</td>
            <td>${f.arrival_time}</td>
            <td>${f.price}</td>
          </tr>
        `;
      });
    });
  </script>

```

```
        `;  
    });  
});  
</script>  
</body>  
</html>
```