

Node JS and Express JS code

```
const express = require('express');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
const PORT = 3000;

// Middleware
app.use(bodyParser.json());
app.use(cors());

// In-memory data storage
let cars = [
  { id: 1, make: 'Honda', model: 'Civic', year: 2018, pricePerDay: 50, description: 'A reliable compact car' },
  { id: 2, make: 'Toyota', model: 'Camry', year: 2020, pricePerDay: 60, description: 'A spacious mid-size car' },
  { id: 3, make: 'Ford', model: 'Mustang', year: 2021, pricePerDay: 100, description: 'A powerful sports car' },
];

let bookings = [
  { id: 1, carId: 1, userId: 1, bookingDate: '2023-03-06' },
  { id: 2, carId: 2, userId: 2, bookingDate: '2023-03-10' },
];

let users = [
  { id: 1, name: 'John', email: 'john@example.com', password: '123456' },
  { id: 2, name: 'Jane', email: 'jane@example.com', password: '123456' },
];
```

```
// API endpoints

app.get('/api/cars', (req, res) => {
  res.json(cars);
});

app.get('/api/bookings', (req, res) => {
  res.json(bookings);
});

app.post('/api/bookings', (req, res) => {
  const { carId, bookingDate, userId } = req.body;

  // Check if car is available for booking on the given date

  const conflictingBooking = bookings.find(booking => booking.carId === carId &&
    booking.bookingDate === bookingDate);

  if (conflictingBooking) {
    res.status(409).json({ error: 'Car already booked on the given date' });
    return;
  }

  // Create new booking

  const newBooking = { id: bookings.length + 1, carId, bookingDate, userId };
  bookings.push(newBooking);

  res.json(newBooking);
});

app.post('/api/users', (req, res) => {
  const { name, email, password } = req.body;
```

```
// Check if user already exists with the same email
const existingUser = users.find(user => user.email === email);
if (existingUser) {
  res.status(409).json({ error: 'User with the same email already exists' });
  return;
}

// Create new user
const newUser = { id: users.length + 1, name, email, password };
users.push(newUser);

res.json(newUser);
});

// Start server
app.listen(PORT, () => {
  console.log(`Server running on port ${PORT}`);
});
```