Online Booking System for Car Rental

Team Members: Aliev Khadis, Wang Haitao, Ritu Arora

2025-02-26

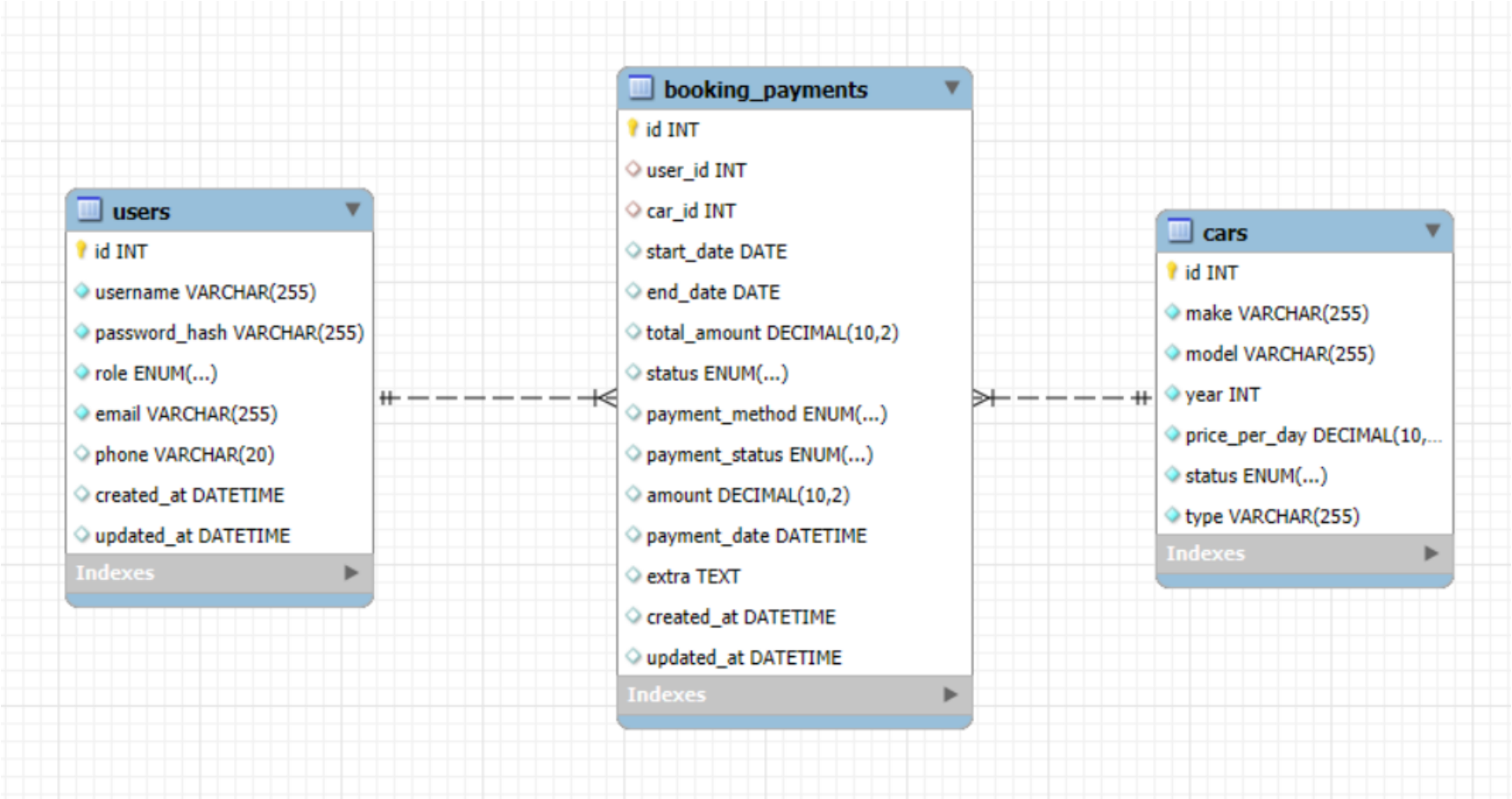
RentNDrive

- Inefficient car rental booking and management processes.
- An online booking system that simplifies car rentals for customers and streamlines management for admins.
- Ease of use for customers.
- Efficient fleet management for admins.
- Real-time booking and availability tracking.

RentNDrive: Key Features

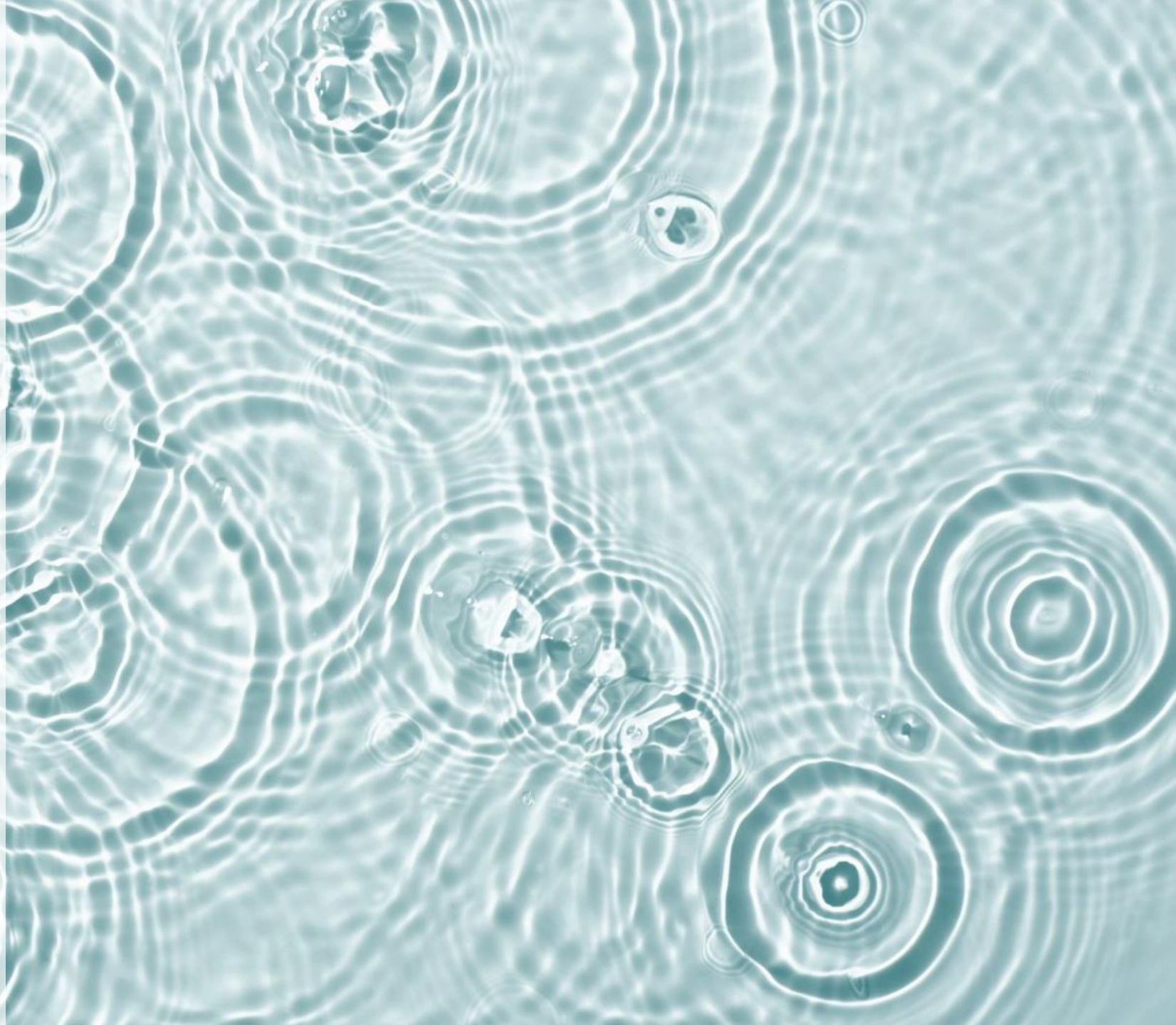
- Booking Wizard (step-by-step process)
- Car Availability Calendar (FullCalendar integration)
- User Authentication (role-based access)
- Admin Dashboard (fleet and booking management)

Entity-Relationship Diagrams



API Design Flow

- User Management: /api/login, /api/register, /api/users/{userId}
- Car Management: /api/cars (GET, POST, PUT, DELETE)
- Booking Management: /api/cars/available, /api/bookings
- Payments: /api/payments



API Design Flow

Customer Logs In

Request: POST /api/login

Response: Receives JWT token for authentication.

Customer Searches for Cars

Request: GET /api/cars/available?start_date=YYYY-MM-DD&end_date=YYYY-MM-DD

Description: Searches for available cars for specific dates (you can add optional filters such as car type, location, etc., depending on your requirements).

Customer Makes a Payment

Request: POST /api/payments

Description: Customer completes payment for the booking.

Customer Makes a Booking

Request: POST /api/bookings

Description: Customer makes a booking for a car by providing booking details like the car, rental dates, etc.

API Design Flow (contd..)

Admin Adds a New Car

Request: POST /api/cars

Description: Admin adds a new car to the fleet with details such as make, model, price, and availability.

Admin Reviews the Booking

Request: GET /api/bookings/{bookingId}

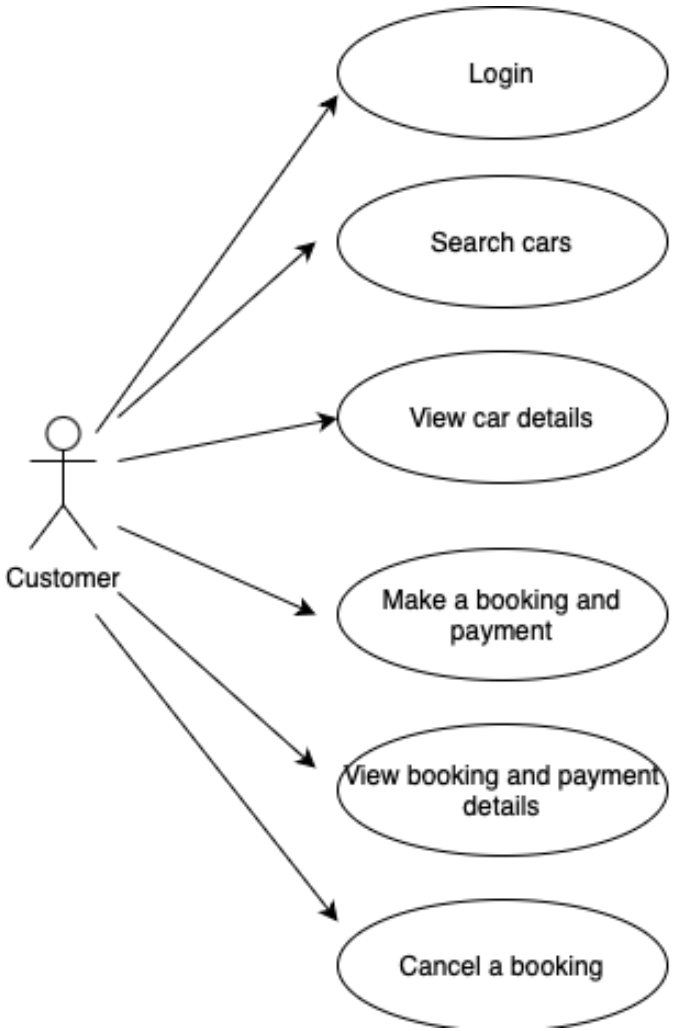
Description: Admin can review details for a specific booking.

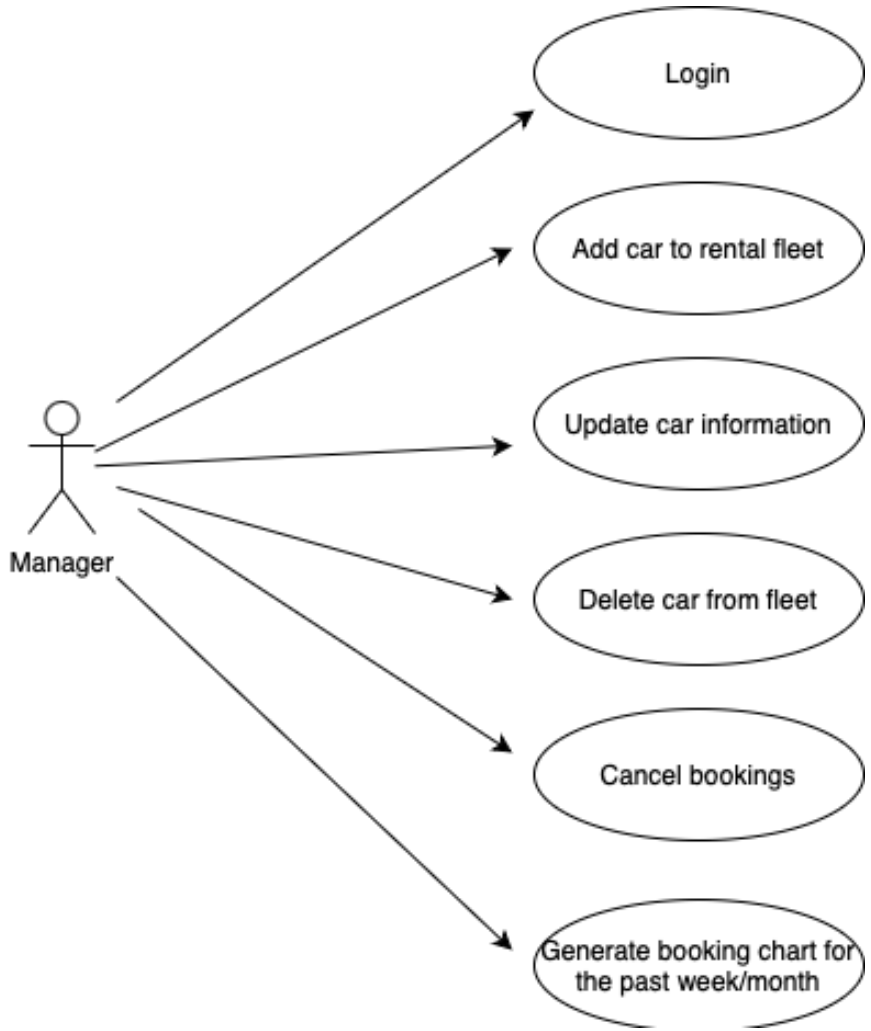
Admin Tracks Car Availability

Request: GET /api/cars/available?start_date=YYYY-MM-DD&end_date=YYYY-MM-DD

Description: Admin tracks availability of cars (This could be part of the same /available API).

Case Diagram

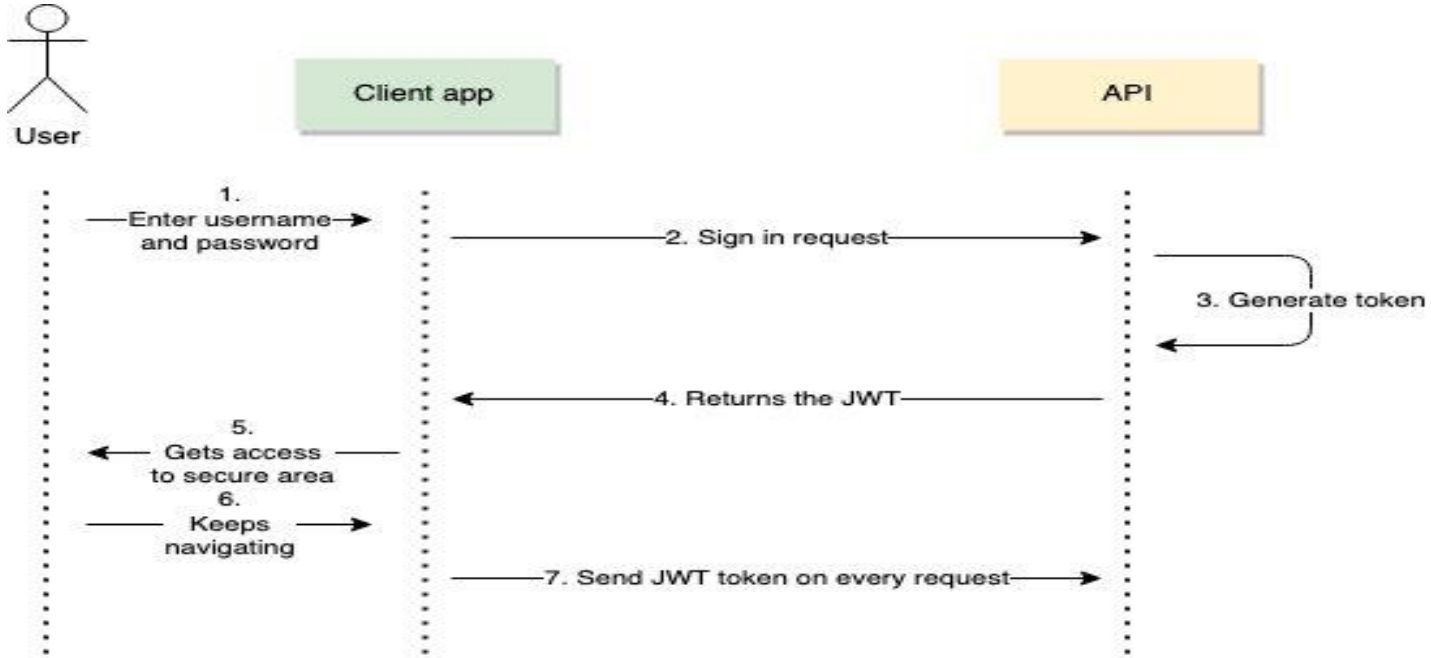




Authentication & Registration

JSON Web Tokens (JWT)

- A standard for securely transmitting information between parties as a JSON object.
- Used for authentication and authorization.
- Self-contained, carrying all necessary user information.



- Used for secure admin and user access.
- Role based access is included in the token.

Registering a New User

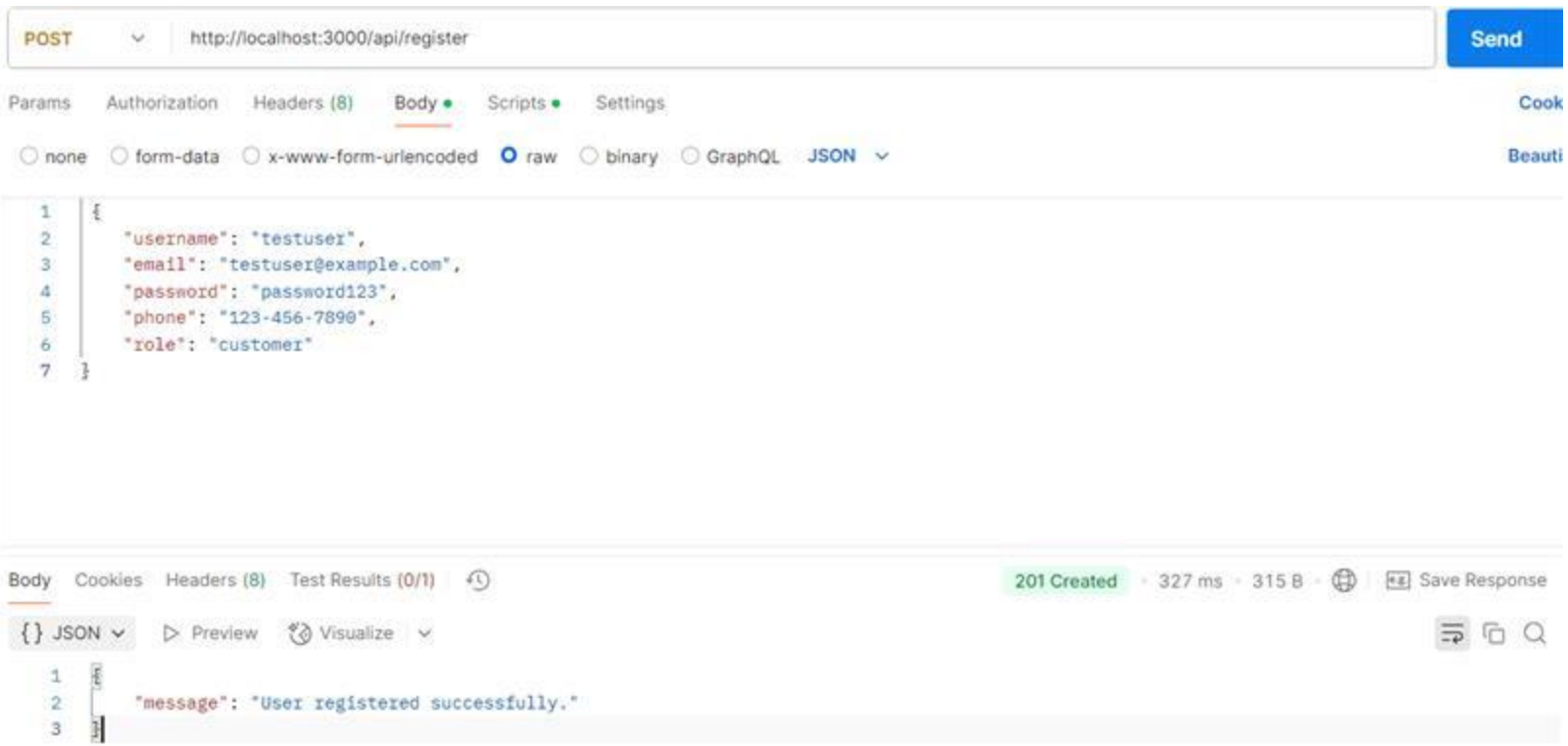
Endpoint: /api/register (POST)

Request:

- Client sends username, email, password, phone, and role (if applicable).
- Server validates input using express-validator.
- Checks for existing users (username or email).
- Hashes the password using bcryptjs.
- Stores user data in the users table.

Response:

- Success: "User registered successfully."
- Error: Validation errors or "Username/email already exists."


POST
http://localhost:3000/api/register
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cook
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beauti
{
"username": "testuser",
"email": "testuser@example.com",
"password": "password123",
"phone": "123-456-7890",
"role": "customer"
}
Body
Cookies
Headers (8)
Test Results (0/1)
201 Created
327 ms
315 B
Save Response
{ } JSON
Preview
Visualize
{
"message": "User registered successfully."
}

User Login and Token Generation

Endpoint: /api/login (POST)

Request:

- Client sends username and password.
- Server validates input.
- Checks if the user exists in the users table.
- Compares the provided password with the

stored hash using bcryptjs.

Response:

- Success: JWT token, user role, username, and id.
- Error: "Invalid username or password."

```
// Example from auth.controller.js (login function)

const jwt = require("jsonwebtoken");
const jwtConfig = require("../config/jwt");

// ... (rest of login function) ...

const token = jwt.sign(
  { id: user.id, role: user.role }, // Payload: user ID and role
  process.env.JWT_SECRET, // Secret key from environment variables
  {
    expiresIn: jwtConfig.expiresIn, // Token expiration time (e.g., '1h', '30m')
    algorithm: jwtConfig.algorithm, // Algorithm used for signing (e.g., 'HS256')
  }
);

console.log("Generated Token:", token); // Log the token for debugging

// ... (send the token in the response) ...
res.json({ token, role: user.role, username: user.username, id: user.id });
```

JWT Generation:

- Uses jsonwebtoken to create the JWT.
- Includes user ID and role in the token payload.
- Uses a secret key (JWT_SECRET) for signing.
- Sets an expiration time for the token.

Protecting Routes with JWT Verification

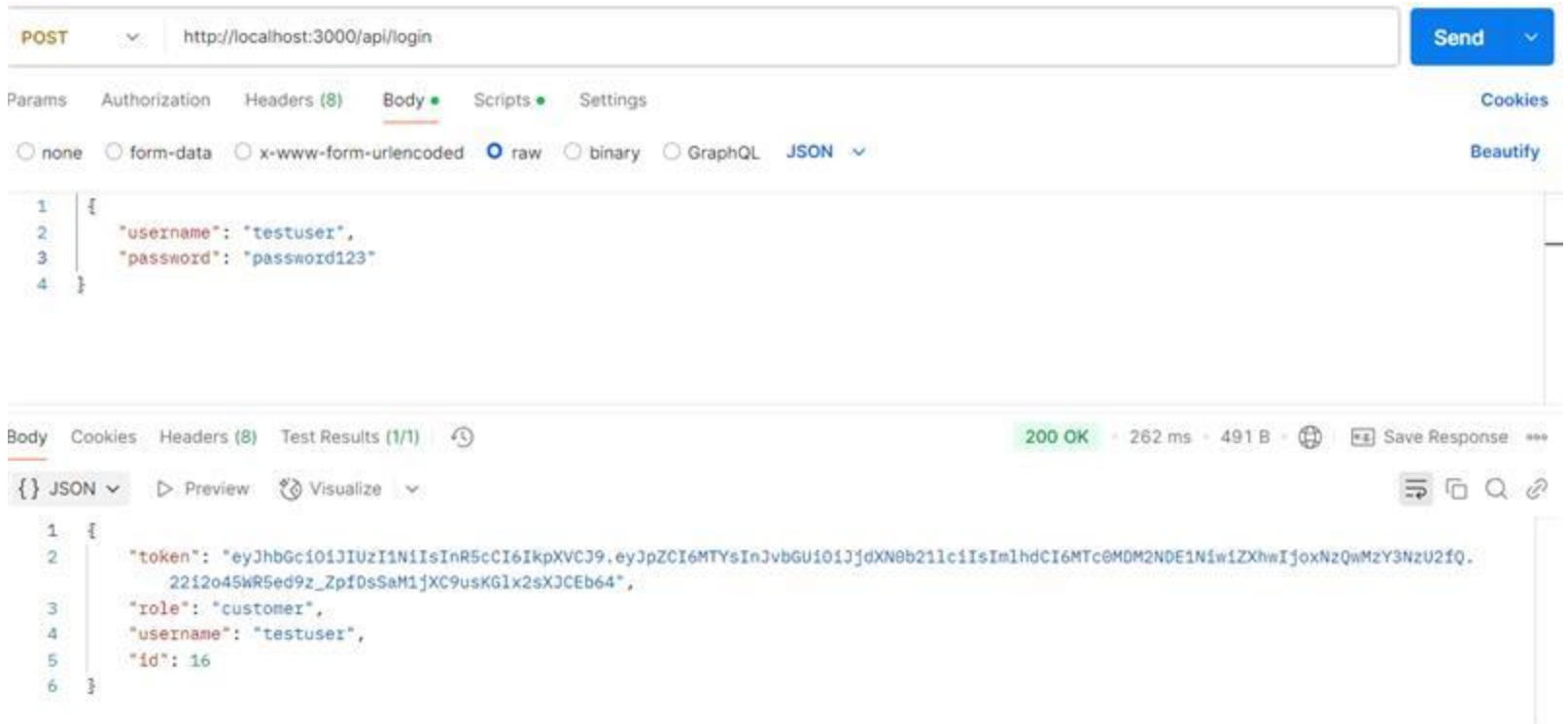
Middleware: authMiddleware.verifyToken

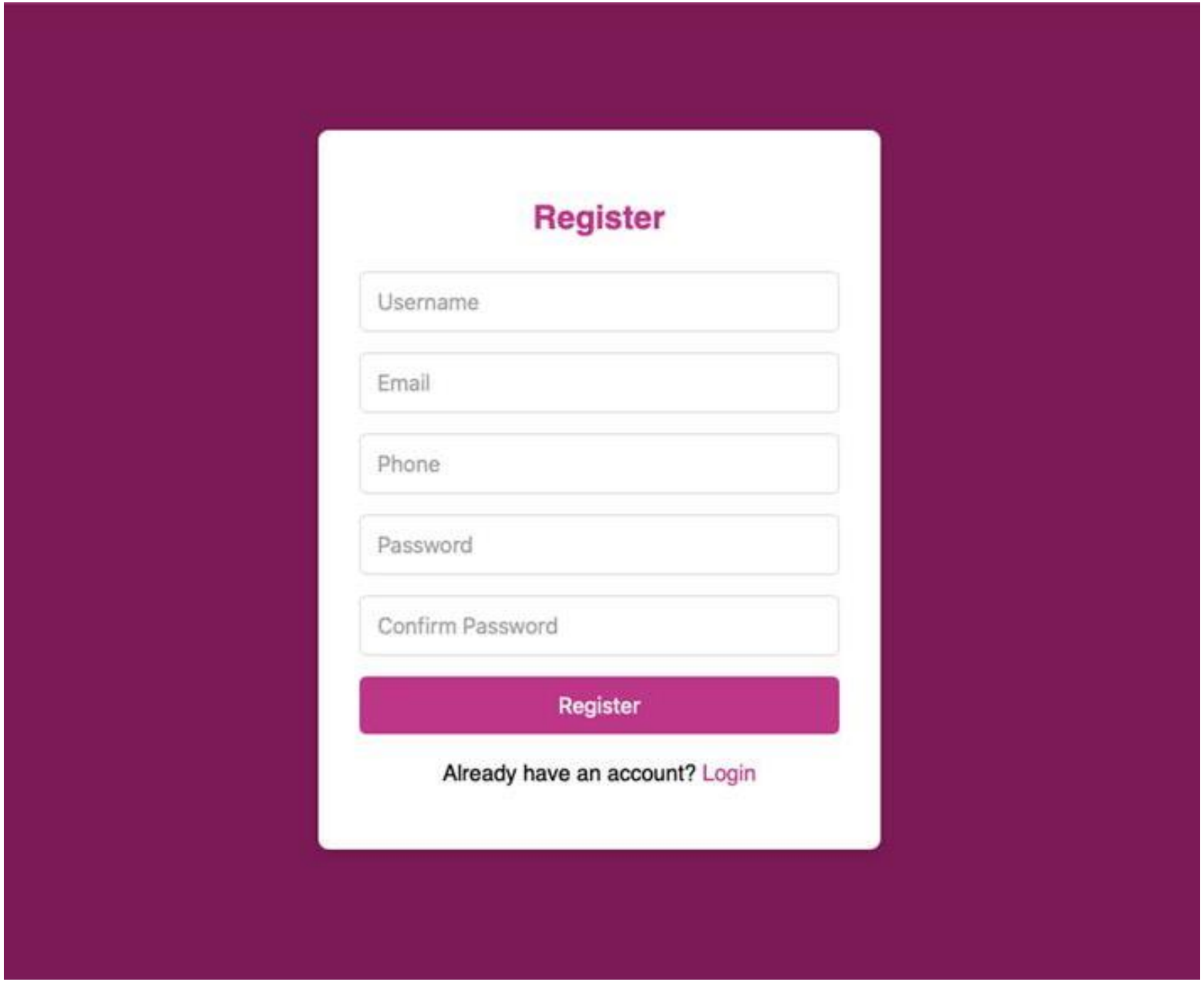
- Extracts the JWT from the Authorization header.
- Verifies the token's signature and expiration.
- Attaches user information to req.user.

Protected Routes:

- Admin routes (e.g., /api/cars) require admin role.
- User routes (e.g., /api/users/{userId}) require a valid token.
- Admin routes use router.use(protectAdminRoute) to protect all admin routes.
- The login and register routes are not protected.

```
// Example from middleware/auth.js

const jwt = require("jsonwebtoken");
const log = require("../../logger");

Tabnine | Edit | Explain
const verifyToken = (req, res, next) => {
  const token = req.header("Authorization");

  if (!token) {
    return res.status(401).json({ message: "Access denied. No token provided." });
  }

  try {
    const decoded = jwt.verify(
      token.replace("Bearer ", ""),
      process.env.JWT_SECRET
    );
    req.user = decoded; // Attach user info to the request
    next(); // Proceed to the next middleware or route handler
  } catch (error) {
    log.warn(`Token verification failed: ${error.message}`);
    res.status(400).json({ message: "Invalid token." });
  }
};

module.exports = { verifyToken };
```


POST
http://localhost:3000/api/login
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"username": "testuser",
"password": "password123"
}
Body
Cookies
Headers (8)
Test Results (1/1)
200 OK
262 ms
491 B
Save Response
JSON
Preview
Visualize
{
"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MTYsInJvbGUiOiJjdXN0b21lciIsImlhdCI6MTc0MDM2NDE1NiwiZXhwIjoxNzQwMzY3NzU2fQ.
22i2o45WR5ed9z_ZpfDsSaM1jXC9usKGlx2sXJCEb64",
"role": "customer",
"username": "testuser",
"id": 16
}


Register
Username
Email
Phone
Password
Confirm Password
Register
Already have an account? Login

RentNDrive

Car rental for any occasion

Browse an incredible selection of cars, from the everyday to the extraordinary.

Explore cars

Admin Dashbord

Dashboard
Manage Cars
Manage Bookings
Manage Customers

Total Cars
15

Total Bookings
16

Total Customers
15

Booking List

Select	ID	User ID	Car ID	Start Date	End Date	Total Amount	Status	Payment Status	Created At	Updated At
○	16	1	1	2024-01-10	2024-01-15	$250.00	confirmed	completed	2025-02-24	2025-02-24
○	17	2	2	2024-01-20	2024-01-25	$225.00	confirmed	completed	2025-02-24	2025-02-24
○	18	3	3	2024-02-01	2024-02-07	$480.00	confirmed	completed	2025-02-24	2025-02-24
○	19	4	4	2024-02-10	2024-02-15	$350.00	pending	pending	2025-02-24	2025-02-24
○	20	5	5	2024-02-20	2024-02-28	$720.00	confirmed	completed	2025-02-24	2025-02-24
○	21	6	6	2024-03-01	2024-03-05	$340.00	confirmed	completed	2025-02-24	2025-02-24
○	22	7	7	2024-03-10	2024-03-17	$525.00	pending	pending	2025-02-24	2025-02-24
○	23	8	8	2024-03-20	2024-03-25	$275.00	confirmed	completed	2025-02-24	2025-02-24
○	24	9	9	2024-04-01	2024-04-08	$420.00	confirmed	completed	2025-02-24	2025-02-24

Make Year Type Price Apply

Tesla Model 3 2022

BMW X5 2022

Mercedes C-Class 2023

Ford Mustang 2023

POST http://localhost:3000/api/bookings Send

Params Authorization Headers (9) Body Scripts Settings Coo

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beau

```
{
    "carId": 1, // Replace with actual car ID
    "startDate": "2024-01-10",
    "endDate": "2024-01-15",
    "paymentMethod": "credit_card",
    "amount": 330, // Calculate amount based on price_per_day and days
    "stripeToken": "tok_visa" // Replace with a test token from stripe.
}
```

Body Cookies Headers (8) Test Results (0/1) 201 Created 1.28 s 669 B Save Response

{} JSON Preview Visualize


```
    "id": 1,
    "user_id": 16,
    "car_id": 1,
    "start_date": "2024-01-10",
    "end_date": "2024-01-15",
    "total_amount": 330,
    "status": "confirmed",
    "payment_method": "credit_card",
    "payment_status": "completed",
    "amount": 330,
    "created_at": "2025-02-24T04:48:27.026Z",
    "updated_at": "2025-02-24T04:48:27.026Z",
    "updatedAt": "2025-02-24T04:48:27.078Z",
    "createdAt": "2025-02-24T04:48:27.029Z",
    "payment_date": "2025-02-24T04:48:27.078Z"
}
```


Postbot Runner Start Proxy Cookies Vault

Booking and payment Details

Car Information

Car Image

Make:

Year:

Type:

Price per Day:

Start Date:

yyyy-mm-dd

End Date:

yyyy-mm-dd

Extra:

Total Price: $0

Payment Information

Cardholder's Name:

Card Number:

Expiry Date:

---------, ----

CVV:

Make a Booking & Pay

Challenges

- **Test Fragility & Maintenance**

Tests break easily with small code modifications, creating additional maintenance overhead.

- **Endless Bug Fixing & Retesting**

Frequent bug generation after every code change leads to constant retesting and fixes.

Contribution from each Team Member

Khadis Aliev

- Client-side - Bookings

Haitao Wang

- Server-side – Admin dashboard
- Client-side – Admin and customer dashboards, and Payments

Ritu Arora

- Cloud Database management
- Server-side – User Registration and Authentication, Bookings and payments
- Client-side - User Registration and Authentication
- Guided team members and managed ScrumBoard and Git repository


Thank you

POST http://localhost:8080/api/register Send

Params Authorization Headers (9) Body Scripts Settings Cookie

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "username": "inuser",
    "email": "inuser@example.com",
    "password": "password123",
    "phone": "123-456-7890",
    "role": "customer"
}
```

Body Cookies Headers (8) Test Results (0/1) 201 Created 446 ms 315 B Save Response

{} JSON Preview Visualize

```
{
    "message": "User registered successfully."
}
```

POST http://localhost:8080/api/login Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Headers 8 hidden

Key	Value	Description	Bulk Edit Presets
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI...		
Key	Value	Description	

Body Cookies Headers (8) Test Results (1/1) 200 OK 234 ms 485 B Save Response

JSON Preview Visualize


```
{
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MjIsInJvbGUiOiJhZG1pbiIsImlhdCI6MTc0MDU5NDg5MywiZXhwIjoxNzQwNTk4NDkzfQ.
        J42DbZv_Zz2VANcuqErmSQv7la-xvbpU26-TxaPzzQE",
    "role": "admin",
    "username": "testadmin",
    "id": 22
}
```

GET http://localhost:8080/api/users/22 Send

Params Authorization Headers (9) Body Scripts Settings Cookies

Headers 8 hidden

Key	Value	Description	Bulk Edit Presets
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI...		
Key	Value	Description	

Body Cookies Headers (8) Test Results (1/1) 200 OK 151 ms 450 B Save Response

JSON Preview Visualize


```
{
    "id": 22,
    "username": "testadmin",
    "role": "admin",
    "email": "testadmin@example.com",
    "phone": "123-000-7890",
    "created_at": "2025-02-25T22:04:00.000Z",
    "updated_at": "2025-02-25T22:04:00.000Z"
}
```

GET http://localhost:8080/api/admin/cars

arams Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
    "username": "testadmin",
    "password": "password123"
}
```

ody Cookies Headers (8) Test Results (1/1) 200 OK

{ } JSON Preview Visualize


```
[
    {
        "id": 1,
        "make": "Toyota",
        "model": "Camry",
        "year": 2022,
        "price_per_day": "50.00",
        "status": "available",
        "type": "Sedan"
    },
    {
        "id": 2,
        "make": "Honda",
        "model": "Civic",
        "year": 2021,
        "price_per_day": "45.00",
        "status": "booked",
        "type": "Sedan"
    },
```

POST http://localhost:8080/api/admin/cars Send

Params Authorization Headers (9) Body Scripts Settings Co

none form-data x-www-form-urlencoded raw binary GraphQL JSON Bea

```
{
  "make": "Toyota",
  "model": "Camry",
  "year": "2022",
  "price_per_day": 50.00,
  "status": "available",
```

Body Cookies Headers (8) Test Results (0/1) 201 Created 161 ms 427 B Save Respons

{} JSON Preview Visualize


```
{
    "message": "Car added successfully.",
    "car": {
        "id": 17,
        "make": "Toyota",
        "model": "Camry",
        "year": "2022",
        "price_per_day": 50,
        "status": "available",
        "type": "Sedan"
    }
}
```

PUT http://localhost:8080/api/admin/cars/17 Send

Params Authorization Headers (9) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
{
  "price_per_day": 55.00
}
```

Body Cookies Headers (8) Test Results (1/1) 200 OK 277 ms 422 B Save Respo

JSON Preview Visualize

```
{
    "message": "Car updated successfully.",
    "car": {
        "id": 17,
        "make": "Toyota",
        "model": "Camry",
        "year": 2022,
        "price_per_day": 55,
        "status": "available",
        "type": "Sedan"
    }
}
```

DELETE http://localhost:8080/api/admin/cars/17 Send

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1  {
2    "price_per_day": 55.00
3  }
4
5
```

Body Cookies Headers (8) Test Results (1/1) 200 OK 192 ms 306 B Save Response

{} JSON Preview Visualize

```
1  {
2      "message": "Car deleted successfully."
3  }
```

POST http://localhost:8080/api/booking/bookings Send

Params Authorization Headers (9) Body Scripts Settings Cooki

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautif

```
{
    "carId": 15, // Replace with actual car ID
    "startDate": "2024-01-10",
    "endDate": "2024-01-15",
    "paymentMethod": "credit_card",
    "amount": 300, // Calculate amount based on price_per_day and days
    "stripeToken": "tok_visa" // Replace with a test token from stripe.
}
```

Body Cookies Headers (8) Test Results (0/1) 400 Bad Request 217 ms 339 B Save Response

{} JSON Preview Visualize

```
{
    "message": "Calculated amount does not match provided amount."
}
```

GET http://localhost:8080/api/booking/cars/available?startDate=2024-01-10&endDate=2024-01-15 Send

Params • Authorization Headers (9) Body • Scripts • Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "carId": 1, // Replace with actual car ID
    "startDate": "2025-07-10",
    "endDate": "2025-07-15",
    "paymentMethod": "credit_card",
    "amount": 300, // Calculate amount based on price_per_day and days
    "stripeToken": "tok_visa" // Replace with a test token from stripe.
}
```

Body Cookies Headers (8) Test Results (1/1) 200 OK • 163 ms • 1.59 KB • Save Response

{} JSON Preview Visualize


```
[
    {
        "id": 4,
        "make": "Chevrolet",
        "model": "Suburban",
        "year": 2020,
        "price_per_day": "70.00",
        "status": "available",
        "type": "SUV"
    },
    {
        "id": 5,
        "make": "BMW",
        "model": "X5"
```

POST http://localhost:8080/api/booking/bookings

Send

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
    "carId": 1, // Replace with actual car ID
    "startDate": "2025-07-10",
    "endDate": "2025-07-15",
    "paymentMethod": "credit_card",
    "amount": 300, // Calculate amount based on price_per_day and days
    "stripeToken": "tok_visa" // Replace with a test token from stripe.
}
```

Body Cookies Headers (8) Test Results (0/1)

201 Created 363 ms 592 B Save Response

{} JSON Preview Visualize

```
{
    "id": 33,
    "user_id": 16,
    "car_id": 1,
    "start_date": "2025-07-10",
    "end_date": "2025-07-15",
    "total_amount": 300,
    "status": "confirmed",
    "payment_method": "credit_card",
    "payment_status": "completed",
```