“System Architecture for Rent-a-Car Website”

[Frontend (Next.js)]

| | |

[Sanity CMS] [Booking API] [Payment Gateway]

| | |

[Car Data API] [Booking Data API] [Payment Processing API]

“ Workflow for Rent-a-car”

1. User Registration:

- **Action:** User signs up.
- **Process:** User submits their registration details.
- **Data Flow:** Data is stored in Sanity CMS (user data).
- **Outcome:** Confirmation is sent to the user via email.

2. Car Browsing:

- **Action:** User views available cars (Sedans, SUVs, etc.).
- **Process:** Car categories are displayed to the user.
- **Data Flow:** Frontend sends request to the Car Data API (powered by Sanity CMS) to fetch car details (name, price, availability).
- **Outcome:** Car details (model, rental price, availability) are displayed on the frontend.

3. Car Selection & Rental Duration:

- **Action:** User selects a car and specifies rental duration (e.g., 3 days, 7 days).
- **Process:** Rental price, available dates, and car model are displayed.
- **Data Flow:** The user selects a car and enters rental duration; this info is stored in Sanity CMS via API.

- **Outcome:** "Rent Now" button is activated, and rental details are passed to the checkout page.

4. Payment Process:

- **Action:** User proceeds to checkout.
- **Process:** Payment details (credit card, PayPal, etc.) are entered.
- **Data Flow:** Payment is processed via the Payment Gateway API (third-party integration).
- **Outcome:** Payment is successfully processed, and confirmation is sent to the frontend.

5. Booking Confirmation:

- **Action:** The booking is confirmed.
- **Process:** Rental details (rental duration, car model, payment status) are saved in Sanity CMS.
- **Data Flow:** Booking information (order ID, rental car, dates, total cost) is stored in Sanity CMS.
- **Outcome:** Confirmation message and booking summary are sent to the user.

6. Booking Details View:

- **Action:** User can view their rental booking details.
- **Process:** User navigates to their booking details page via the User Panel.
- **Data Flow:** The booking details are fetched from Sanity CMS (order ID, car model, rental dates, total price).
- **Outcome:** User can view all details including car rental dates, rental period, and cost.

7. User Panel (Ride Tracking):

- **Action:** User wants to track their ride (car delivery or rental status).
- **Process:** User logs in to their User Panel to view ride status.
- **Data Flow:** The User Panel fetches the ride status from the **Ride Tracking API** (could be a third-party API for real-time car tracking).
- **Outcome:** The ride status, including car delivery or rental location, is displayed to the user. If available, users can see estimated delivery time, current status, and tracking updates.

8. Shipment (Car Delivery) Tracking (if applicable):

- **Action:** User checks for real-time car delivery status.
- **Process:** Shipment status is tracked through a third-party API.
- **Data Flow:** Real-time status is fetched from the Ride Status API.
- **Outcome:** The real-time shipment status of the car is displayed on the frontend for the user.

“Detailed schema of project”

1. Car Endpoints

Get All Cars

- **Endpoint Name:** `/cars`
- **Method:** `GET`
- **Description:** Get a list of all available cars for rent

```
   [

      {

  "carId": "1",

  "model": "Tesla Model 3",

  "pricePerDay": 100,

  "carType": "Sedan",

  "availability": true

},



 ]
```

Get Specific Car Details

- **Endpoint Name:** `/cars/{carId}`
- **Method:** `GET`
- **Description:** Get details of a specific car by ID.

```
 {

"carId": "1",
  "model": "Tesla Model 3",
  "pricePerDay": 100,
  "carType": "Sedan",
  "availability": true
   }
```

---

2. User Endpoints

User Registration

- **Endpoint Name:** `/users/register`
- **Method:** `POST`
- **Description:** Register a new user.

```
{
  "email": "john@example.com",
  "password": "securepassword"
}
```

```
{
  "userId": "123",
  "email": "john@example.com"
}
```

User Login

- **Endpoint Name:** `/users/login`
- **Method:** `POST`
- **Description:** Log in a user and receive authentication details.

```
{
  "email": "john@example.com",
  "password": "securepassword"
}
```

```
{
  "userId": "123",
  "token": "jwt-token"
}
```

---

3. Payment Endpoints

Create Payment

- **Endpoint Name:** `/payments`
- **Method:** `POST`
- **Description:** Process a payment for a booking.

```
{
  "bookingId": "456",
  "amount": 500,
  "paymentMethod": "Credit Card"
}
```

```
{
  "paymentId": "789",
  "status": "Success",
  "amount": 500
}
```

Summary of API Endpoints:

1. **Car Endpoints:**
   - `GET /cars`: Get all cars.
   - `GET /cars/{carId}`: Get details of a specific car.
2. **User Endpoints:**
   - `POST /users/register`: Register a new user.
   - `POST /users/login`: Log in a user.
3. **Payment Endpoints:**
   - `POST /payments`: Process a payment for a booking.