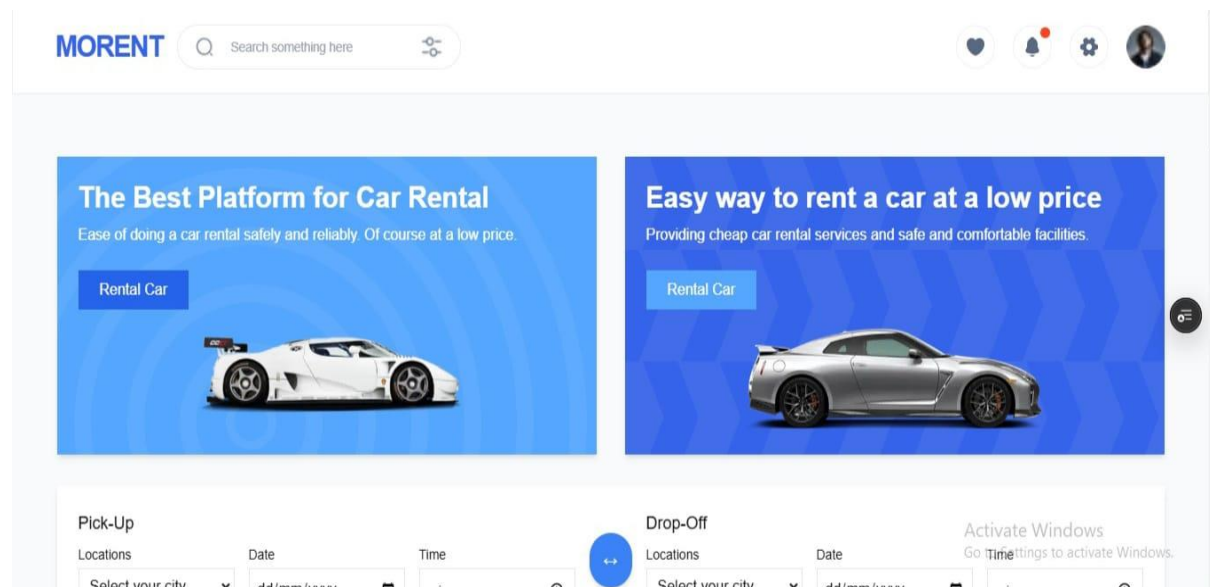


# Hackathon DAY 2

## Marketplace Technical Foundation (Rental E-Commerce)

This document outlines the end-to-end user experience for an online rental marketplace, highlighting critical touchpoints and their technical execution. Its purpose is to guide the development of a seamless, intuitive, and robust platform that meets the needs of both renters and suppliers.

### Home Page Rental E-Commerce



## Frontend Requirements

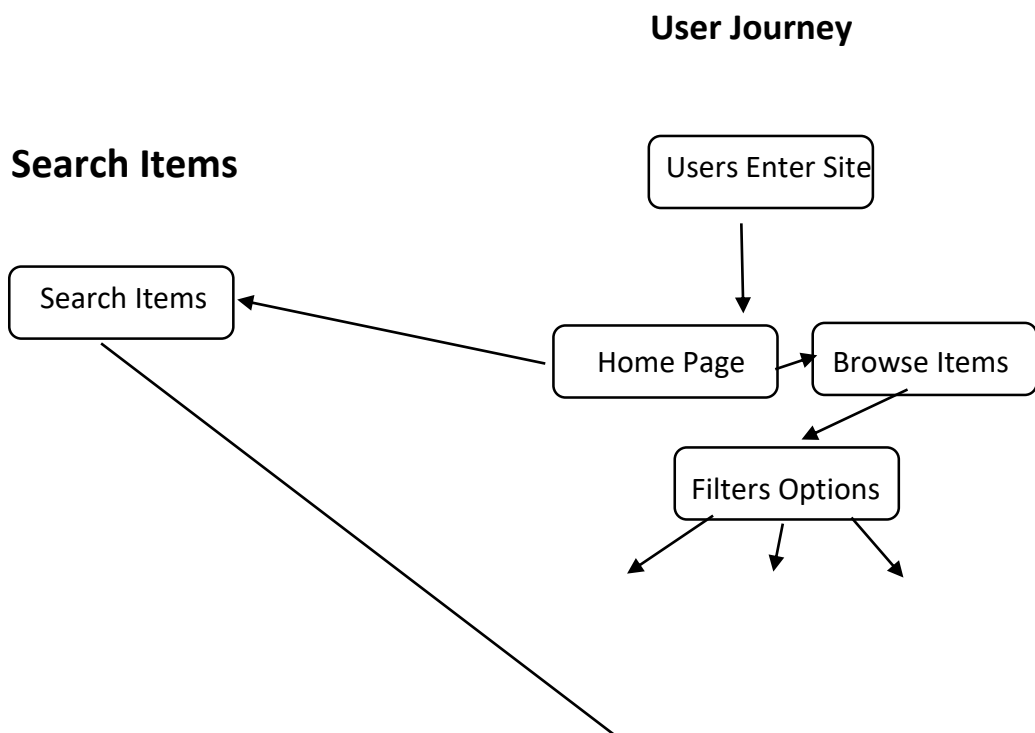
- **Responsive Design:** Ensure usability on mobile, tablet, and desktop devices.
- **Essential Pages:** Include Home, Product Listing, Product Details, Cart, and Checkout.
- **User-Friendly Interface:** Create intuitive navigation and clear call-to-action buttons.
- **Dynamic Data:** Display live product availability and rental pricing from the backend.
- **Search and Filters:** Add filters for categories, price, and rental duration with a search bar.
- **User Authentication:** Provide login, sign-up, and account management options.

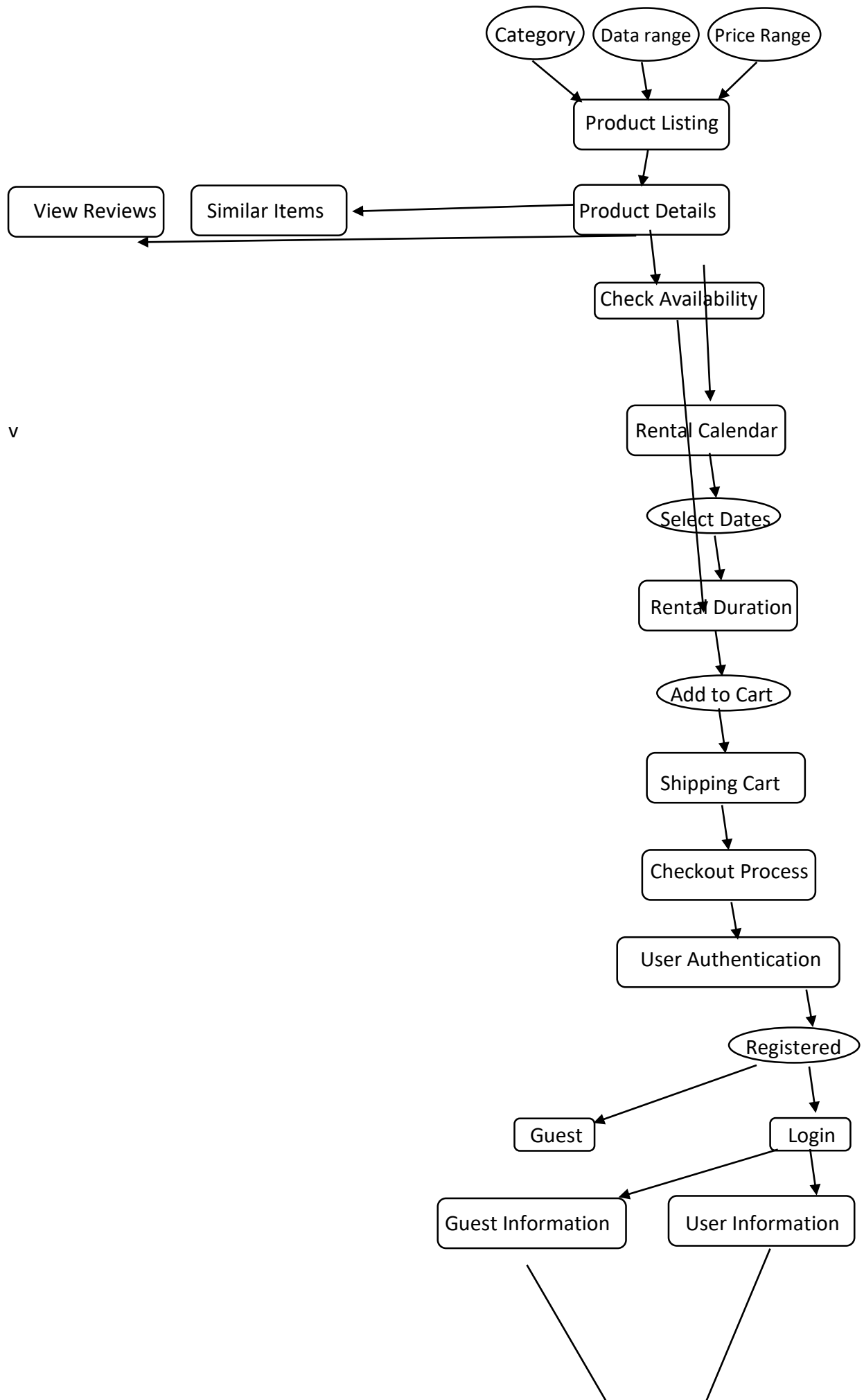
- **API Integration:** Integrate payment gateways and shipment tracking services.
- **Modern Design:** Use clean and attractive layouts for a better user experience.

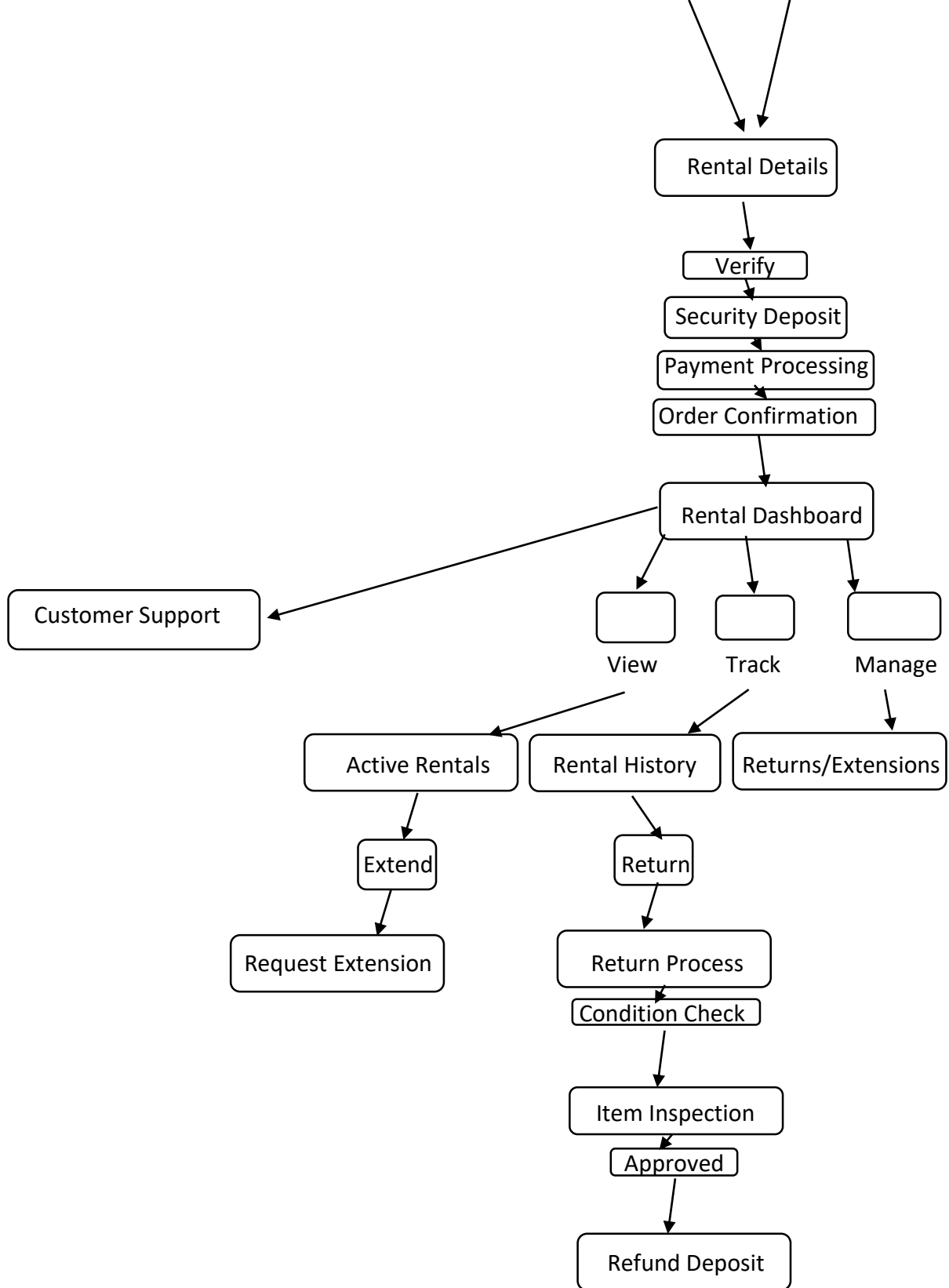
## Chart Overview:

- **Home:** User enters the site and goes to the homepage.
- **Browse:** User browses and filters products by category, date range, or price.
- **Product Selection:** User views product details, checks availability, and selects rental dates.
- **Cart & Checkout:** Items are added to the cart, and the user proceeds to checkout.
- **Rental Details:** User enters rental details.
- **Payment:** Security deposit is verified, followed by payment processing and order confirmation
- **Delivery:** User provides delivery details, and the item is shipped
- **Dashboard:** User accesses the rental dashboard to view active rentals, rental history, and manage returns or extensions.
- **Return Process:** Items are inspected for condition before refunding the security deposit.
- **Search & Review:** Users can search for items, view reviews, and find similar products.
- **Customer Support:** Dashboard provides access to support services.

## Frontend Requirements Flow Chart







## System Architecture

### Frontend:

- User interface for browsing products, placing orders, and tracking shipments.
- Responsive design for all devices.

- Dynamic product data fetched from the backend.
- Pages: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation.
- Sends API requests to fetch product listings, manage orders, and track shipments.

## **Sanity CMS:**

- Acts as the content management system and database for products, orders, and customers.
- Stores data for products (name, price, availability), orders (customer info, status), and rentals.
- Provides endpoints for frontend data retrieval.
- Integrates with third-party APIs for payment and shipment.

## **Third-Party APIs:**

- Processes secure payments for rental transactions.
- Sends payment status updates to the backend.
- Tracks the delivery status of orders.
- Sends real-time updates to the frontend via the backend.

## **Data Flow**

### **Product Browsing:**

- User requests product data via the frontend.
- Backend (Sanity CMS) serves product data dynamically.

### **Order Placement:**

- Frontend sends order details (rental duration, customer info) to the backend.
- Backend saves the order and processes payment via the Payment Gateway.

### **Shipment Tracking:**

- Backend requests order shipment details from the Shipment Tracking API.
- Real-time updates are displayed on the frontend.

## System Architecture Overview Diagram

[Frontend (Next.js)]

[Sanity CMS]

[Product Data API]

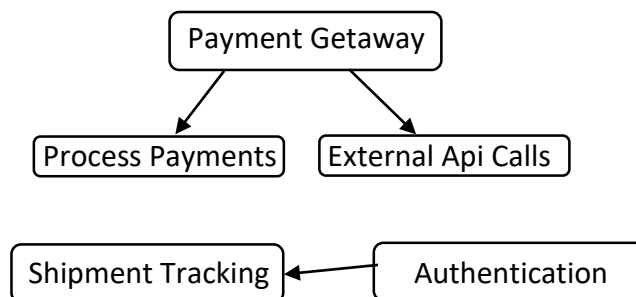
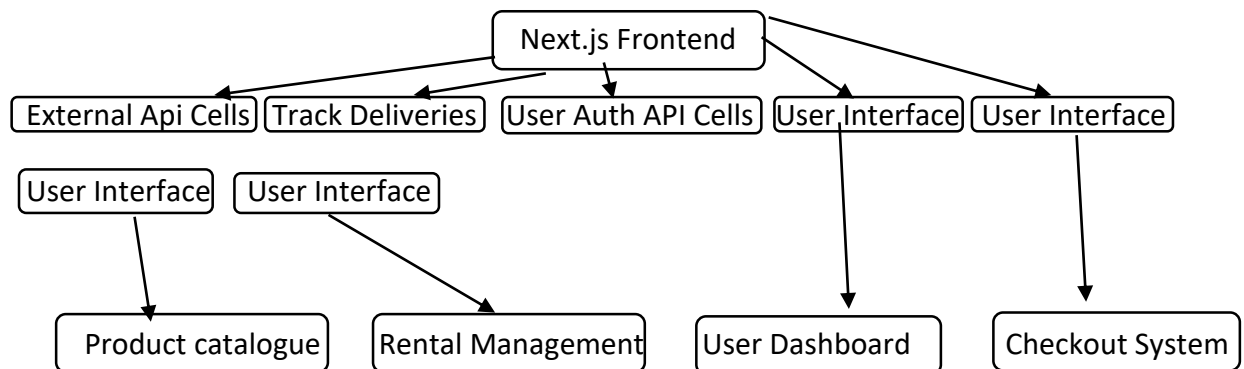
[Third-Party API]

[Shipment Tracking API]

[Payment Gateway]

## System Architecture

### Frontend layer



## ER Diagram for Rental E – Commerce

User		
String	User Id	pk
String	Name	
String	Email	
String	Phone	
String	Address	
Array	Payment methods	
Array	Rental history	

Products		
String	Name	Pk
String	Base price	
Number	Description	
String	Images	
Array	Category	
String	Images	
string	Condition	
Boolean	is Available	

Rental		
String	Rental id	Pk
String	User id	Fk
String	Product id	Fk
Date	Start date	
Date	End date	
Number	Deposit amount	
String	Status	
String	Condition Report	
Boolean	Is Active	

Order		
String	Order id	Pk
String	Rental id	Fk
Number	Total amount	
String	Payment status	
Date	Order date	
String	Shipping address	
String	Tracking info	

## Entities and Attributes

- **Product:** Attributes: id, name, price, stock, image, rental Duration.
- **Customer:** Attributes: id, name, email, phone, address.
- **Order:** Attributes: id, customer ID, product ID, rental Duration, payment Status, order Status.
- **Payment:** Attributes: id, order ID, payment Amount, payment Method, payment Status.

## Relationships

- A product can be part of multiple orders.
- One order can have multiple products.
- A customer can place multiple orders.
- Each order belongs to one customer.
- Each order has one payment.

## API Requirements

### Rental Duration and Booking:

#### Post:

Post /Api/rental-duration.

```
{  
  "product ID": 456,  
  "duration": "7 days",  
  "deposit": 500,  
  "start Date": "2025-02-01",  
  "end Date": "2025-02-08",  
  "user ID": "user123"  
}
```

#### Response:

```
{
```



```
"confirmation ID": 789,
"status": "Success",
"rental Details": {
  "total Cost": 350,
  "deposit Amount": 500,
  "start Date": "2025-02-01",
  "end Date": "2025-02-08",
  "product Info": {
    "id": 456,
    "name": "Premium Camera"
  }
}
}
```

## Rental Availability

**GET** /Api/rental-availability/{product ID}

- **Description:** Check if a product is available for specific dates.
- **Query Parameters:** start Date, end Date.

```
{
  "is Available": true,
  "available Dates": [
    {
      "start Date": "2025-02-01",
      "end Date": "2025-03-01"
    }
  ],
  "base Price": 50,
  "deposit Required": 500
}
```

```
}
```

## **Extend Rental Duration**

**PATCH** /Api/rental-duration/{rental ID}/extend

```
{
```

```
  "extension Days": 3,
```

```
  "newEndDate": "2025-02-11"
```

```
}
```

```
{
```

```
  "status": "Success",
```

```
  "updated Rental": {
```

```
    "rental ID": 789,
```

```
    "newEndDate": "2025-02-11",
```

```
    "additional Cost": 150
```

```
  }
```

```
}
```

## **Rental Condition Report**

**POST** /Api/rental/{rental ID}/condition

```
{
```

```
  "condition": "Good",
```

```
  "notes": "Minor scratch on left side",
```

```
  "images": ["image1.jpg", "image2.jpg"],
```

```
  "report Type": "pre-rental"
```

```
}
```

```
{
```

```
"report ID": "report123",  
"status": "Submitted",  
"timestamp": "2025-02-01T10:00:00Z"  
}
```

## Get Rental History

**GET** /Api/rental/history/{user ID}

```
{  
  "rentals": [  
    {  
      "rental ID": 789,  
      "product Name": "Premium Camera",  
      "start Date": "2025-02-01",  
      "end Date": "2025-02-08",  
      "status": "Active",  
      "total Paid": 350  
    }  
  ],  
  "total Rentals": 5,  
  "active Rentals": 1  
}
```

## Cancel Rental

**POST** /Api/rental/{rental ID}/cancel

```
{  
  "cancellation Reason": "Change of plans",  
}
```

```
"request Refund": true
}

{
  "status": "Cancelled",
  "refund Amount": 850,
  "refunded": "ref123",
  "cancellation Fee": 50
}
```

## Process Rental Return

**POST** /Api/rental/{rental ID}/return

```
{
  "return Date": "2025-02-08",
  "condition": "Good",
  "damage Report": null,
  "deposit Return": true
}

{
  "status": "Returned",
  "deposit Return Amount": 500,
  "processing Id": "proc123",
  "additional charges": 0
}
```

## Rental Details

**GET** /Api/rental/{rental ID}

```
{  
  "rental": {  
    "id": 789,  
    "product": {  
      "id": 456,  
      "name": "Premium Camera",  
      "category": "Photography"  
    },  
    "duration": "7 days",  
    "status": "Active",  
    "deposit": {  
      "amount": 500,  
      "status": "Held"  
    },  
    "dates": {  
      "start": "2025-02-01",  
      "end": "2025-02-08"  
    },  
    "costs": {  
      "rental Fee": 350,  
      "deposit": 500,  
      "total": 850  
    },  
    "condition Reports": {
```

```
"preRental": "report123",  
"post Rental": null  
}  
  
}  
  
}
```

**Name: Tahira Ibrahim**

**Roll no: 103987**

**Class Time: Saturday 9: 12**

**Mentor: Sir Bilal Sir Aneeq**