Technical Requirements

Frontend Requirements

- **User-Friendly Interface:**
  - A seamless and intuitive design for browsing books and navigating the platform.
- **Responsive Design:**
  - Optimize the interface for both mobile and desktop users to ensure accessibility across devices.
- **Essential Pages:**
  - **Home Page:** Showcase featured books, promotions, and categories.
  - **Product Listing Page:** Display available books with filters (genre, author, etc.).
  - **Product Details Page:** Provide detailed information about individual books, including rental terms.
  - **Cart Page:** Allow users to review their selected items before checkout.
  - **Checkout Page:** Facilitate payment and delivery options.
  - **Order Confirmation Page:** Display order summary and confirmation details post-payment.

Backend Requirements (Sanity CMS)

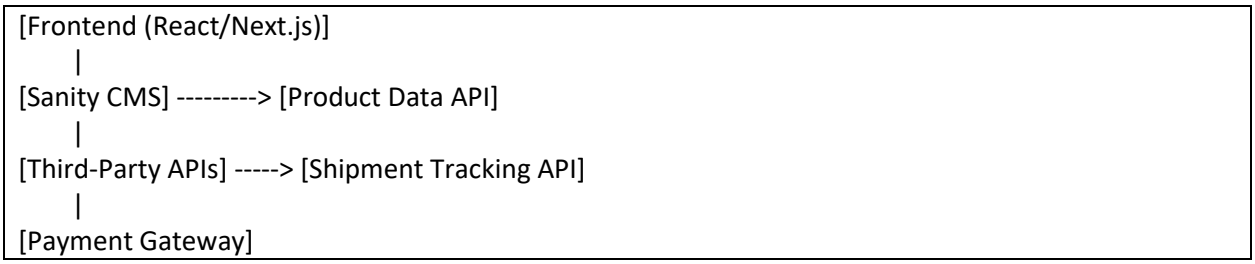
- **Sanity CMS for Data Management:**
  - Manage product inventory (books), customer profiles, and order records.
  - Create custom schemas for:
    - **Products:** Book details (name, author, genre, stock, price).
    - **Orders:** Linked to customers, products, and shipment status.
    - **Customers:** Name, contact info, rental history.
- **Schema Alignment:**
  - Design schemas that reflect the business goals established during the planning phase.

Third-Party API Integration

- **Shipment Tracking:**
  - Use APIs to track deliveries and update shipment statuses in real-time.
- **Payment Gateways:**
  - Integrate secure APIs for processing payments (e.g., Stripe, PayPal).
  - Include support for multiple payment methods (credit card, mobile wallets, etc.).
- **Additional Backend Services:**
  - Notifications: Send order confirmations and updates to customers via email or SMS.
  - Analytics: Track user interactions and sales for performance insights.

System Architecture Design

High-Level Architecture Diagram



Detailed Data Flow

1. **Browsing Products:**
   - Users visit the **Frontend** to browse available books.
   - The frontend dynamically fetches book listings and details from the **Product Data API** connected to **Sanity CMS**.
2. **Placing an Order:**
   - When users add items to the cart and proceed to checkout, the order details are sent via an API to **Sanity CMS**, where they are recorded.
3. **Shipment Tracking:**
   - After the order is processed, shipment details are retrieved using a **Third-Party Shipment Tracking API**.
   - Real-time updates on shipment status are displayed to the user on the frontend.
4. **Payment Processing:**
   - Payment information is securely transmitted to the **Payment Gateway** for processing.
   - Upon successful payment, confirmation details are returned to the user and recorded in **Sanity CMS**.

API Requirements

1. Rental Duration Management

Endpoint Name: `/rental-duration`

- **Method:** POST
- **Description:** Add rental details for a specific product.
- **Payload:**

```
{
"productId": 456,
```

```
 "duration": "7 days",

  "deposit": 500

}
```

- **Response Example:**

```
{
  "confirmationId": 789,
  "status": "Success"
}
```

2. Product Management

Endpoint Name: `/products`

- **Method:** GET
- **Description:** Fetch all available products from the backend.
- **Response Example:**

```
[

  {

    "id": 101,

    "name": "Book Title",

    "price": 150,

    "stock": 20,

    "image": "url_to_image"

  }

]
```

3. Order Management

Endpoint Name: `/orders`

- **Method:** POST
- **Description:** Create a new order and store it in the backend.
- **Payload:**

```
{
```

```
  "customerId": 123,

  "productId": 456,

  "quantity": 1,

  "rentalDuration": "7 days",

  "paymentStatus": "Paid"

}
```

- Response Example:
```
{
  "orderId": 789,
  "status": "Order Confirmed"
}
```

4. Shipment Tracking

Endpoint Name: `/shipment`

- **Method:** GET
- **Description:** Track the status of an order via a third-party API.
- **Parameters:**
  - `orderId`: The ID of the order to track.
- **Response Example:**

```
{

  "shipmentId": 987,

  "orderId": 789,

  "status": "In Transit",

  "expectedDeliveryDate": "2025-01-20"

}
```

5. Customer Management

Endpoint Name: `/customers`

- **Method:** GET
- **Description:** Retrieve customer details, including rental history.
- **Response Example:**

```
{
  "customerId": 123,
  "name": "John Doe",
  "email": "john.doe@example.com",
  "rentalHistory": [
    {
      "productId": 456,
      "rentalDuration": "7 days",
      "status": "Returned"
    }
  ]
}
```

6. Payment Management

Endpoint Name: `/payments`

- **Method:** POST
- **Description:** Process payment for a rental order.
- **Payload:**

```
{

  "orderId": 789,

  "paymentMethod": "Credit Card",

  "amount": 650

}
```

- **Response Example:**

```
{

  "transactionId": 112233,

  "status": "Success"

}
```

7. Penalty Management

Endpoint Name: `/penalties`

- **Method:** POST
- **Description:** Apply penalties for damaged or unreturned products.
- **Payload:**

```
{
  "orderId": 789,
  "penaltyReason": "Damaged Book",
  "penaltyAmount": 200
}
```

- **Response Example:**

```
{
  "penaltyId": 456,
  "status": "Penalty Applied"
}
```