System Architecture Document (Clothing Store)
1. System Architecture Document 1.1 Overview The marketplace website for your clothing store is designed to allow customers to browse and purchase premium clothing items. The system consists of several key components:

Frontend: A React-based web application providing the user interface for customers to interact with the marketplace.
Backend: A mock API that simulates the processing of data, including fetching clothing items, handling user interactions, and performing CRUD operations.
CMS: Sanity CMS to manage and store content such as clothing details, categories, and other relevant information.
Database: Integrated with Sanity to store data related to clothing items, customer profiles, and order histories.
2. Frontend Key Pages:

Homepage: A page showcasing banners, featured clothing, categories (e.g., pants, shirts), and easy navigation.
Clothing Listing: Displays a grid of clothing items with filters such as material, type (pants, shirts), category, and price range.
Clothing Details: A page showing detailed information for each clothing item (description, size, material, images).
Cart: A section to review and modify selected items before checkout.
Checkout: A user-friendly interface for finalizing orders, entering payment information, and selecting delivery options.
Order Tracking: A dedicated page to track the delivery status of placed orders.
3. Sanity CMS Key Functionalities:

Product Management: Store and manage clothing details like name, price, category (pants, shirts), material, size, images, and descriptions.
Order Management: Track customer orders, payment status, and shipping details.
4. Third-Party APIs

Payment Gateway: Integrate Stripe or PayPal for secure payment processing.
Features: Credit/debit card payment support, multi-currency support, refund and dispute management.
Shipment Tracking: Use AfterShip or similar courier APIs for real-time tracking.
Features: Real-time status updates (Shipped, Out for Delivery, Delivered), notifications for order movement.
Mock API: Simulate data and API responses without live APIs.
Features: Simulate clothing data, order statuses, and tracking updates.
5. API Structure Clothing API:

/api/clothing (GET): Fetch all clothing products.
/api/clothing/{id} (GET): Fetch a single clothing product by ID.
/api/clothing (POST): Add a new clothing product.
/api/clothing/{id} (PUT): Update clothing product details.
/api/clothing/{id} (DELETE): Delete a clothing product.
Categories API:

/api/categories (GET): Fetch all clothing categories.
/api/categories/{id} (GET): Fetch a single category by ID.
Orders API:

/api/orders (GET): Fetch all orders for a user.
/api/orders/{id} (GET): Fetch details of a single order.
/api/orders (POST): Place a new order.
/api/orders/{id}/ship (POST): Mark an order as shipped.
/api/orders/{id}/tracking (GET): Fetch tracking info for a shipped order.
Users API:

/api/users/{id} (GET): Fetch user profile details.
Authentication API:

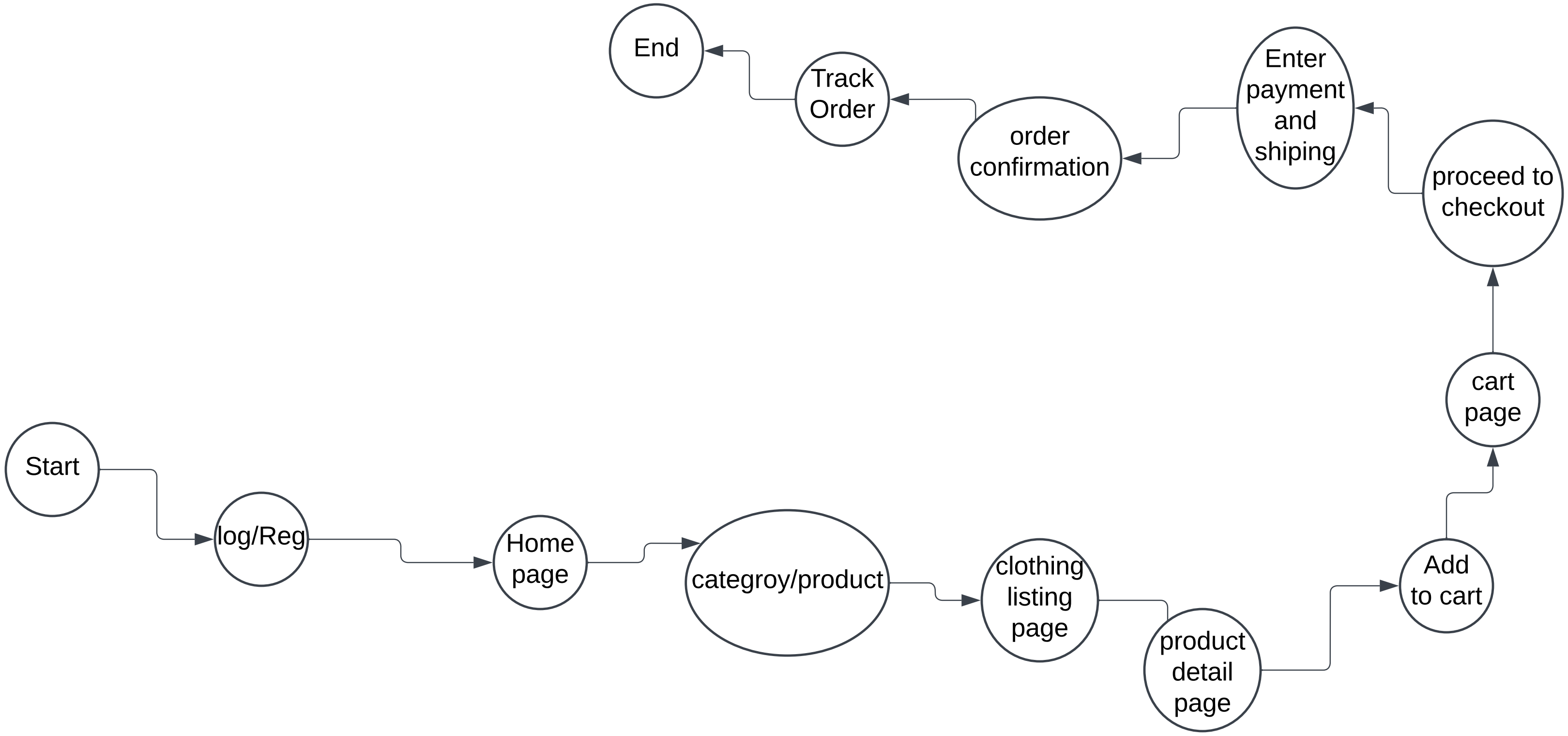
/api/auth/register (POST): User registration.
/api/auth/login (POST): User login.
Cart API:

/api/cart (GET): Fetch cart details for a user.
/api/cart/{id} (POST): Add an item to the user's cart.
/api/cart/{id} (DELETE): Remove an item from the user's cart.
6. System Workflow

Login/Signup: User initiates login or signup.
Home Page: User lands on the homepage after login or signup.
Product Page: User views a list of available clothing items.
Single Product Page: User clicks on a product to view details.
Add to Cart: User adds a product to their cart.
Checkout: User finalizes the order.
Track Order: User tracks their order status.
Sanity Schema
Clothing Schema

javascript
Copy
Edit

```
export default {
  name: 'clothing',
  title: 'Clothing',
  type: 'document',
  fields: [
    { name: 'name', title: 'Clothing Name', type: 'string', validation: Rule => Rule.required().min(3).max(50)
},
    { name: 'slug', title: 'Slug', type: 'slug', options: { source: 'name', maxLength: 96 }, validation: Rule =>
Rule.required() },
    { name: 'description', title: 'Description', type: 'text', validation: Rule => Rule.required().max(300) },
    { name: 'price', title: 'Price', type: 'number', validation: Rule => Rule.required().positive() },
    { name: 'category', title: 'Category', type: 'string', options: { list: [{ title: 'Pants', value: 'pants' }, { title:
'Shirts', value: 'shirts' }] }, validation: Rule => Rule.required() },
    { name: 'material', title: 'Material', type: 'string', validation: Rule => Rule.required() },
    { name: 'size', title: 'Size', type: 'string', validation: Rule => Rule.required() },
    { name: 'images', title: 'Images', type: 'array', of: [{ type: 'image' }], options: { hotspot: true } },
    { name: 'stock', title: 'Stock', type: 'number', validation: Rule => Rule.required().min(0) },
    { name: 'createdAt', title: 'Created At', type: 'datetime', initialValue: () => new Date().toISOString() },
  ],
};
```


Start
log/Reg
Home page
categroy/product
clothing listing page
product detail page
Add to cart
cart page
proceed to checkout
Enter payment and shiping
order confirmation
Track Order
End