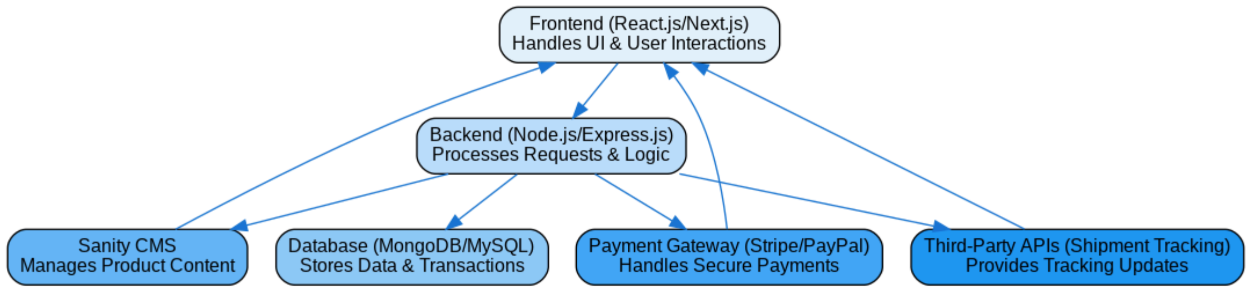
## SYSTEM ARCHITECTURE


Frontend (React.js/Next.js)
Handles UI & User Interactions
Backend (Node.js/Express.js)
Processes Requests & Logic
Sanity CMS
Manages Product Content
Database (MongoDB/MySQL)
Stores Data & Transactions
Payment Gateway (Stripe/PayPal)
Handles Secure Payments
Third-Party APIs (Shipment Tracking)
Provides Tracking Updates

## System Components & Roles:

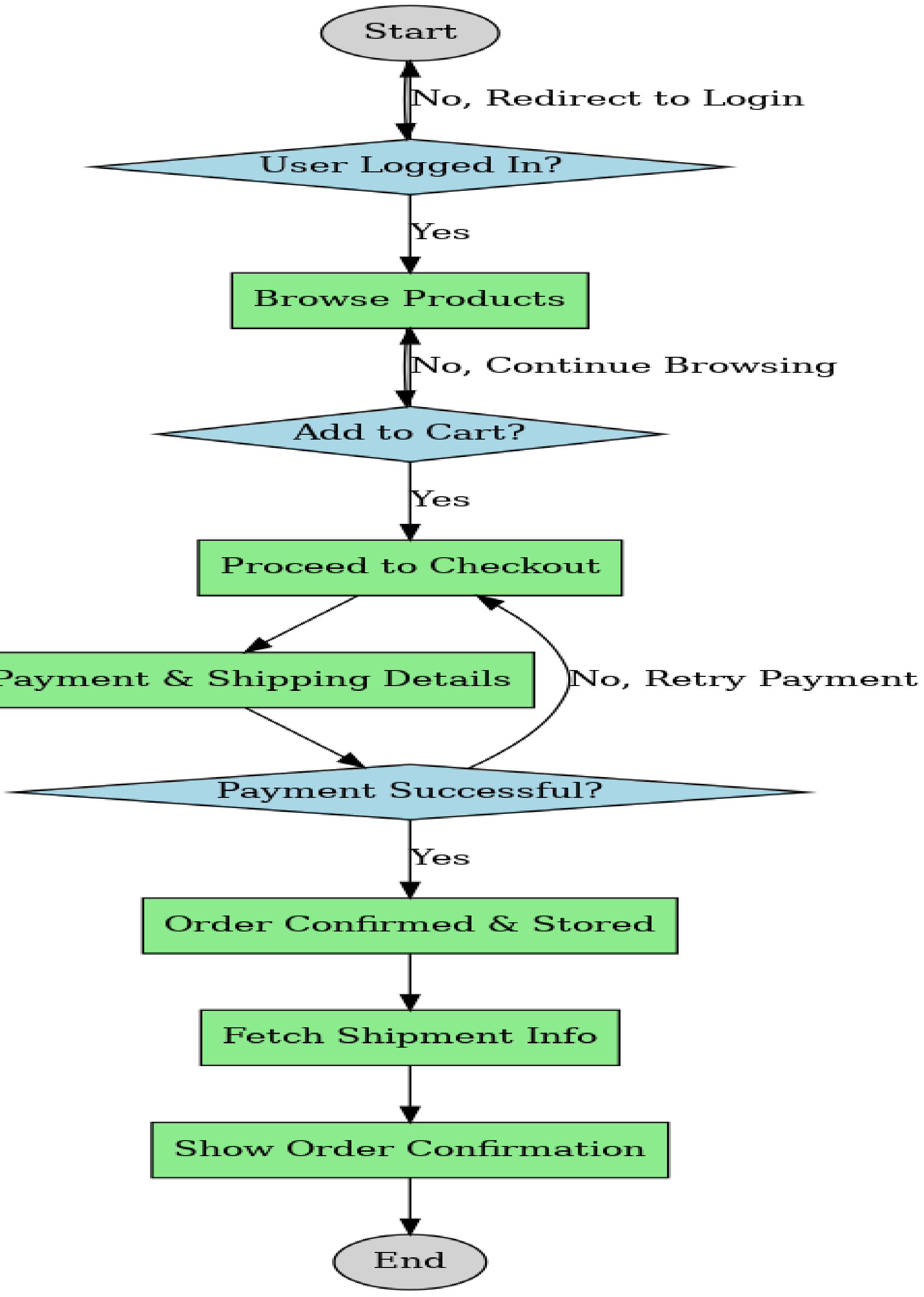
1. **Frontend (React.js/Next.js)**
    - Handles user interactions.
    - Displays product listings dynamically.
    - Sends API requests to fetch product, order, and user details.
2. **Backend (Node.js/Express.js)**
    - Manages business logic and API requests.
    - Communicates with the database and third-party services.
    - Processes orders and user authentication.
3. **Database (MongoDB/MySQL)**
    - Stores user accounts, product details, order history, and payment records.
    - Ensures data integrity and quick retrieval for frontend requests.
4. **CMS (Sanity CMS)**
    - Manages product content dynamically.
    - Allows administrators to add, update, or remove product listings.
5. **Third-Party APIs**
    - **Shipment Tracking API**: Fetches real-time delivery status.
    - **Other APIs** (e.g., tax calculation, recommendations): Enhance functionality.
6. **Payment Gateway (Stripe/PayPal)**
    - Securely processes online transactions.
    - Sends payment confirmation to the backend and updates order status.

## Workflow of Data Flow:

1. **User Registration/Login:**
    - User signs up → Data stored in the database → Confirmation sent via email.
2. **Product Browsing:**
    - User views product categories → Frontend fetches data from the CMS/API → Products displayed.
3. **Order Placement:**
    - User adds items to the cart → Proceeds to checkout → Order details saved in the database.

4. **Payment Processing:**
    - User selects payment method → Transaction processed securely → Order confirmed.
5. **Shipment Tracking:**
    - Order status fetched via Third-Party API → Displayed to the user.

## FLOWCHART



## API ENDPOINTS

| Endpoint | Method | Description | Parameters | Response Example |
|---|---|---|---|---|
| /api/products | GET | Fetch all clothing products | None | { id: 1, name: "T-Shirt" } |
| /api/products/:id | GET | Fetch a single product | id (Path) | { id: 1, name: "T-Shirt" } |
| /api/products | POST | Add a new product | name, price, category (Body) | { success: true, id: 10 } |
| /api/products/:id | PUT | Update a product | id (Path), name, price (Body) | { success: true } |
| /api/products/:id | DELETE | Delete a product | id (Path) | { success: true } |
| /api/categories | GET | Fetch all product categories | None | { categories: ["Shirts", "Pants"] } |
| /api/orders | POST | Place a new order | user_id, items (Body) | { success: true, order_id: 1001 } |
| /api/orders/:id | GET | Fetch order details | id (Path) | { order_id: 1001, status: "Shipped" } |
| /api/users | POST | Register a new user | name, email, password (Body) | { success: true, user_id: 500 } |
| /api/users/:id | GET | Fetch user profile | id (Path) | { id: 500, name: "SANA" } |