

with details like tracking ID, delivery status, payment status, shipment status, and product IDs.

Schema Design in Sanity:

Design schemas that align with business goals:

Product Schema:

fields for name, email, address & contact details.

Customer Schema:

name, email, address & contact

Order Schema:

fields for customer, product-list, payment status, shipment status.

Third-party API:

Goals: External API for advanced functionality.

Requirement:

Product API (meck API, free API)

Shipment tracking API (Shippo, Easy DHV, etc.)

Design System Architecture:

- Frontend (Next.js) \rightarrow user interface for browsing products, managing the cart, and placing orders.
Display the user interface for browsing products, managing the cart, and placing orders.
- Handles user interaction and communication with backend service via APIs.

• Sanity CMS:

- Acts as the primary backend to manage product data, customer details, and order records.
- Provides user API for the frontend to fetch and update data.

Product Data API:

Provides endpoints to fetch product listings, details, and inventory status.

Third-Party API:

Integrates services like shipping tracking and payment processing.

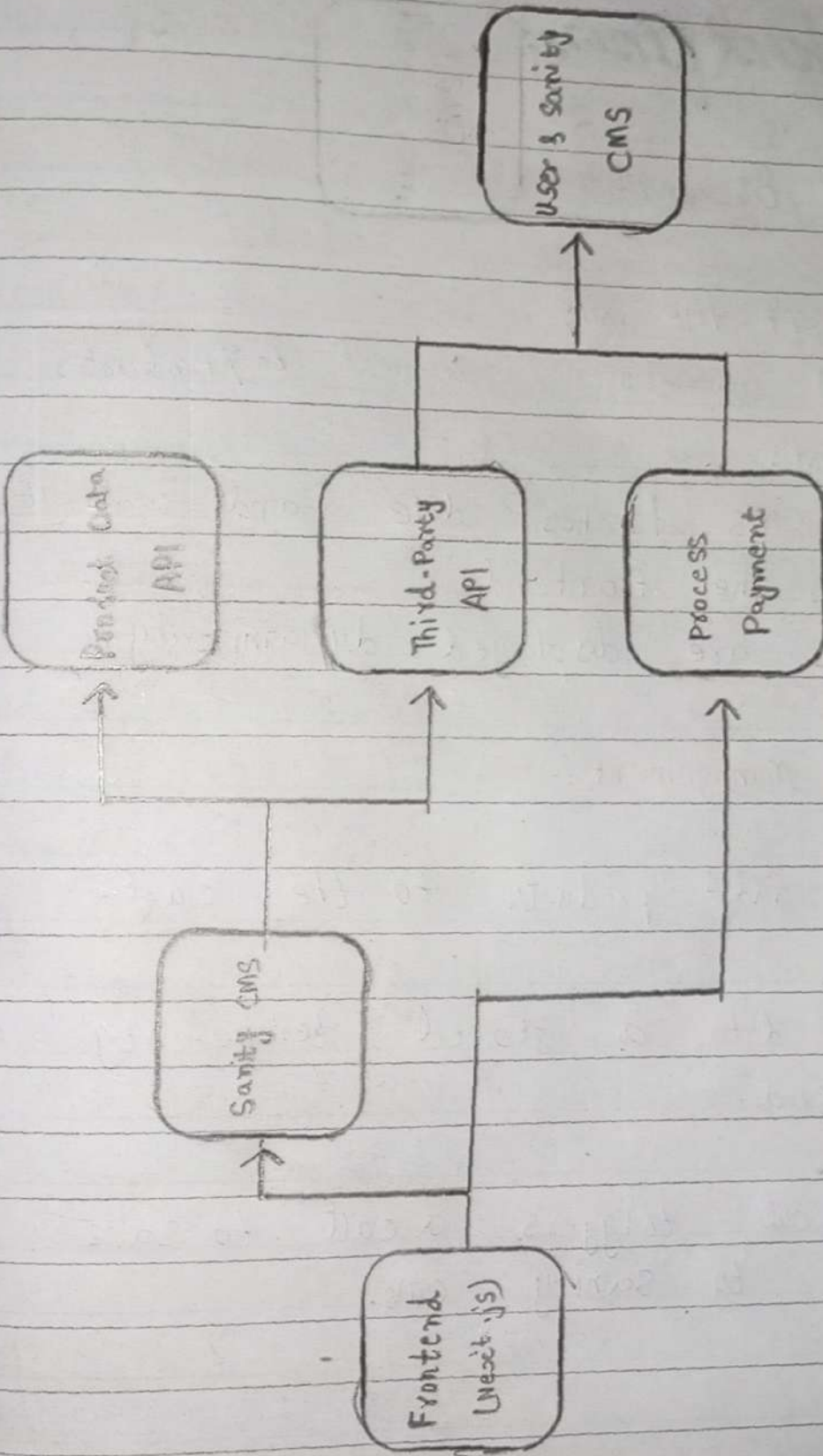
Payment Gateway:

Processes user payment securely and provides transaction confirmation.

Laiba : 00486030

Flow chart

Confirmation
Sent to user
& sanity
CMS



Workflows:

Product Brower:

user visit the site.

frontend sends a request to /Products
end point.

Sanity CMS fetches data and sends it
back to the frontend.

Products are displayed dynamically.

Cart Management :-

user adds products to the cart.

Cart data is stored temporarily on
frontend.

Checkout triggers a call to save
cart to sanity cms.

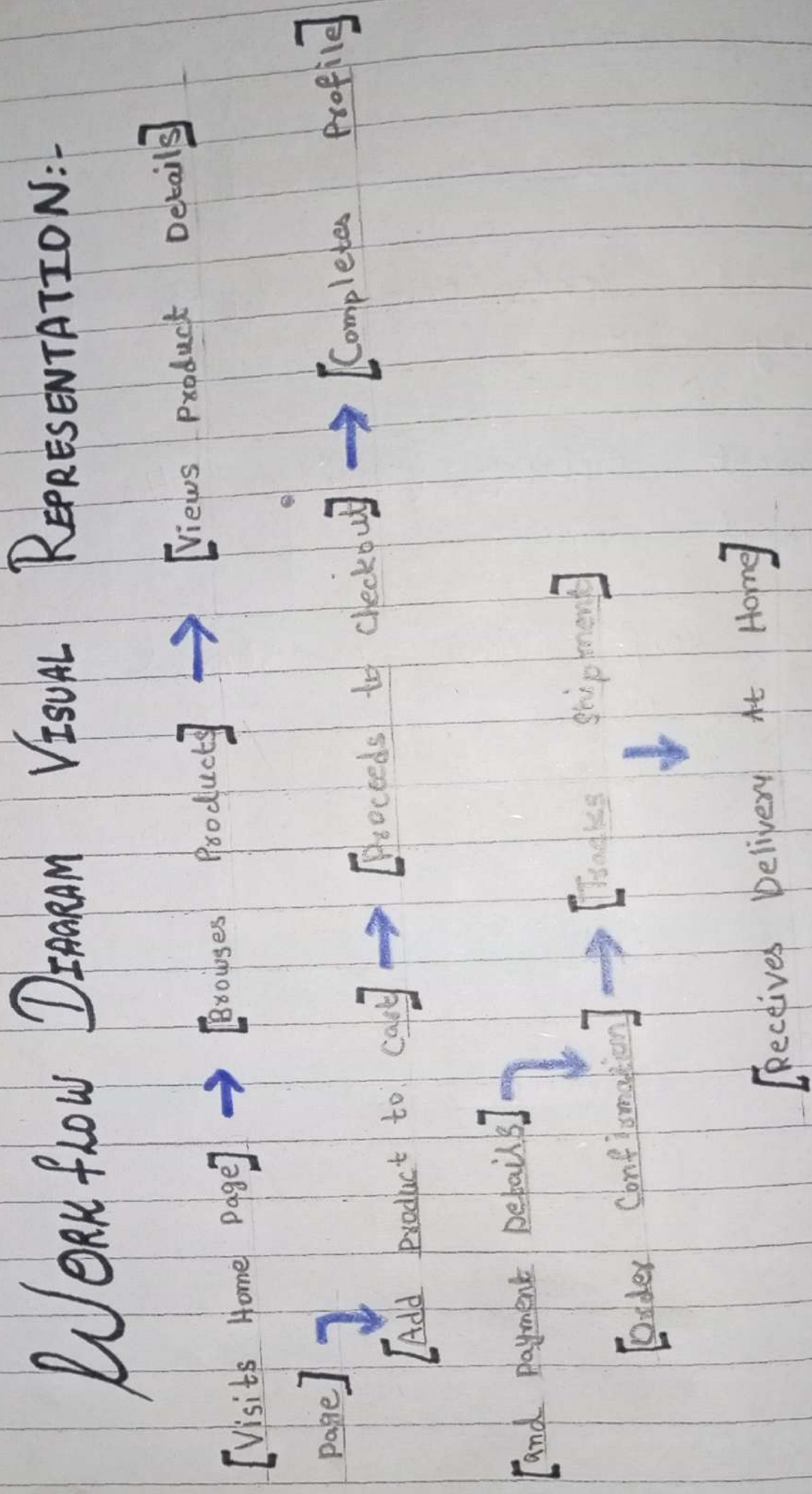
Order placement:-

user proceeds to checkout.

frontend sends order details to
Sanity CMS.

payment gateway processes the
payment securely.

Order confirmation is stored in
Sanity CMS.



Checkout Page:

collects necessary information for shipment and payment to finalize the purchase:

Order Confirmation:

provides a summary of the completed order, displaying a confirmation message if successful or a rejection message if the order fails:

Sanity CMS as Backend:

Requirements:

Product Data Management:

Sanity stores comprehensive product details, including ID, price, description, stock levels, and images.

Customer Data Management:

It maintains customer information such as name, ID, email, and addresses.

Order record:

Sanity keeps track of order

Home Page:

The homepage displays all features, a "Show more" button, benefiting inspiration, and step-by-step navigation.

Shop Page:

The shop page showcase all products, new arrivals, filter, icon, search, add-to-cart, a comparison page link, and recent blog etc.

Blog Page:

include category option and recent blogs for easy access to relevant content.

Contact Page:

The contact page provides contact details, including an email address and phone number for support.

Cart Page:

Displays selected products with their quantities, allowing users to review and adjust item before proceeding to checkout.

Define Technical Requirement:

Technical requirements are divided into three parts.

1. Frontend
2. Backend
3. Third-Party APIs.

Goals:

Build a user-friendly and attractive interface where users can easily interact with the marketplace.

Responsiveness:

Make the interface fully responsive so that it works well on all devices, allowing users to easily explore the marketplace.

User Journey Flow for an eCommerce:

This is a detailed overview of the user journey flow for an e-commerce marketplace, highlighting key steps and their technical implementation. The goal is to create a smooth and user-friendly platform.

Hackathon: 3

Lab: 00485030

Day: 2

Marketplace Technical Foundation

Elite Furnish E-Commerce:

