

Thursday Morning



Niba Khan

Hackathon Day:2

PLANNING THE TECHNICAL FOUNDATION

Frontend Requirement

- *User-Friendly interface for browsing Products:*
- *Responsive design for Mobile , tablet and desktop users*
- *Essential Pages: Home ,Product explore, product Detail, cart, checkout page, payment method, Order Confirmation and Delivery*

E-Commerce Website Work Flow:

1-Homepage:

*When a user opens the website, they'll see a **hero banner** highlighting summer and winter collections. The banner will showcase **offers, new arrivals**, and **seasonal discounts**. Below it, there will be **featured products** and **categories** like **Dresses, Accessories, and Shoes**.*

2-Product Explore:

User can explore products by:

- *Clicking on **categories pages**.*
- *Using the **search bar** to find items like **Dresses, Accessories, or Shoes**.*

3-Product Detail Page:

When a customer clicks on a product:

- *A **detail page** will open with the product's image, description, price, available sizes, and colors.*
- *Below this, there will be an **"Add to Cart" button**.*

4-Cart:

*When customers click **"Add to Cart"**, the product will be added to the cart.*

*On the **Cart Page**:*

- *Customers will see selected products, their quantities, and the total amount.*
- *They can **remove or add items** and check the updated total price.*

Options:

- *Continue Shopping*
- *Proceed to Checkout*

6-Payment Methods:

- *Credit/Debit Card*
- *Cash on Delivery*
- *Online Wallets (e.g., PayPal)*

5-Checkout Page:

On the checkout page , customers will:

- *Enter their delivery address .*
- *Select a payment method .*

7-Order Confirmation:

After confirming their order:

- *Customers will receive a confirmation message/email with an Order ID .*

8-Shipment Tracking:

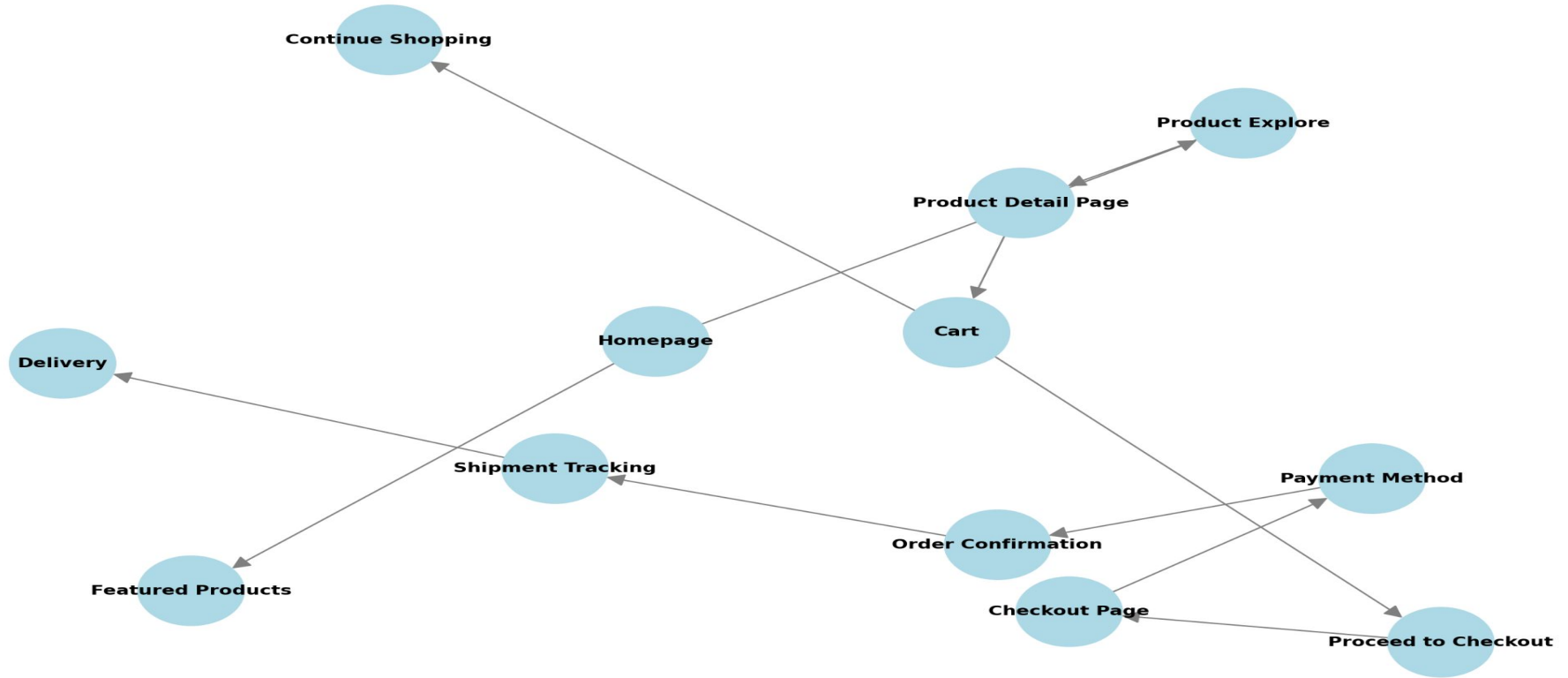
After confirming their order:

- *Customers can go to the Order History section to track their shipment in real-time through API integration.*

9-Delivery::

- *The order will be delivered to the customer's address.*

Ecommerce Website Workflow



Key Components:

Frontend (Next.js):

- *Displays the user interface for browsing products, managing the cart, and placing orders.*
- *Handles user interactions and communicates with backend services via APIs.*
- *Ensures responsiveness and seamless user experience across devices.*
- *Implements client-side validation for forms and real-time updates (e.g., cart total changes).*
-

Backend (Sanity CMS):

The backend is the part that manages the website's data. Sanity CMS is used to manage your content (products, descriptions, prices, etc.). It stores this content and provides it to the frontend whenever needed.

***Example :** When a customer sees a product on the frontend, the product data is actually coming from the backend (Sanity CMS).*

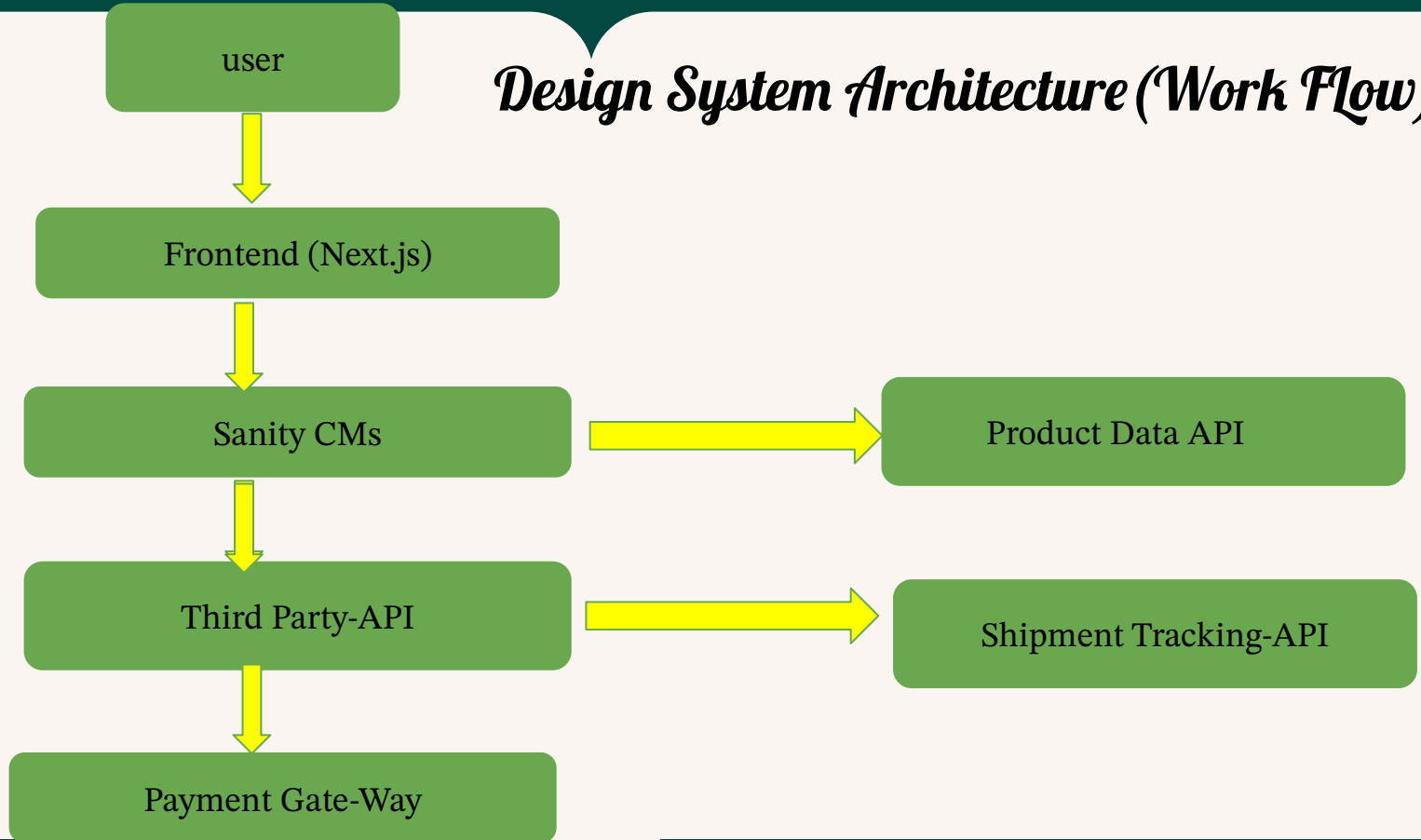
Third-Party APIs:

- *Integrates services like shipment tracking and payment processing.*
- *Connects with marketing tools for personalized product recommendations.*
- *Allows social media login integrations (e.g., Google, Facebook).*
- *Supports SMS and email notifications for order status updates.*

Payment Gateway:

Payment gateways securely handle payment information from the website and send back the status of the transaction through API responses.

Design System Architecture (Work Flow)



API EndPoints:

1-POST /api/products – Add a new product.

- Request Body: { "title": "Summer T-Shirt", "description": "Comfortable cotton t-shirt", "price": 19.99, "image": "image_url_here", "category": "Clothing" }
- Response: { "message": "Product added successfully", "productId": 1 }

2- GET /api/products – Get all products (with optional filters like category, price range).

- Query Parameters:
category=Clothing&minPrice=10&maxPrice=50
- Response: [{ "id": 1, "title": "Summer T-Shirt", "price": 19.99, "image": "image_url_here" }, ...]

3-GET /api/products/:id – Get a single product by ID.

Response: { "id": 1, "title": "Summer T-Shirt", "description": "Comfortable cotton t-shirt", "price": 19.99, "image": "image_url_here" }

4-PUT /api/products/:id – Update an existing product.

- Request Body: { "title": "Updated Summer T-Shirt", "price": 22.99 }
- Response: { "message": "Product updated successfully" }

5-DELETE /api/products/:id – Delete a product.

- Response: { "message": "Product deleted successfully" }

Sanity Schema

```
1  // schemas/product.js
2  export default {
3    name: 'product',
4    title: 'Product',
5    type: 'document',
6    fields: [
7      {
8        name: 'title',
9        title: 'Title',
10       type: 'string',
11     },
12     {
13       name: 'slug',
14       title: 'Slug',
15       type: 'slug',
16       options: {
17         source: 'title',
18         maxLength: 96,
19       },
20     },
21     {
22       name: 'description',
23       title: 'Description',
24       type: 'text',
25     },
26     {
27       name: 'price',
28       title: 'Price',
29       type: 'number',
30     },
31     {
32       name: 'image',
33       title: 'Image',
34       type: 'image',
35       options: {
36         hotspot: true,
37       },
38     },
39     {
40       name: 'category',
41       title: 'Category',
42       type: 'string',
43       options: {
44         list: [
45           { title: 'Clothing', value: 'clothing' },
46           { title: 'Accessories', value: 'accessories' },
47           { title: 'Footwear', value: 'footwear' },
48         ],
49       },
50     },
51     {
52       name: 'availableStock',
53       title: 'Available Stock',
54       type: 'number',
55     },
56   ],
57 }
```

Thank You

Thursday Morning



Niba Khan