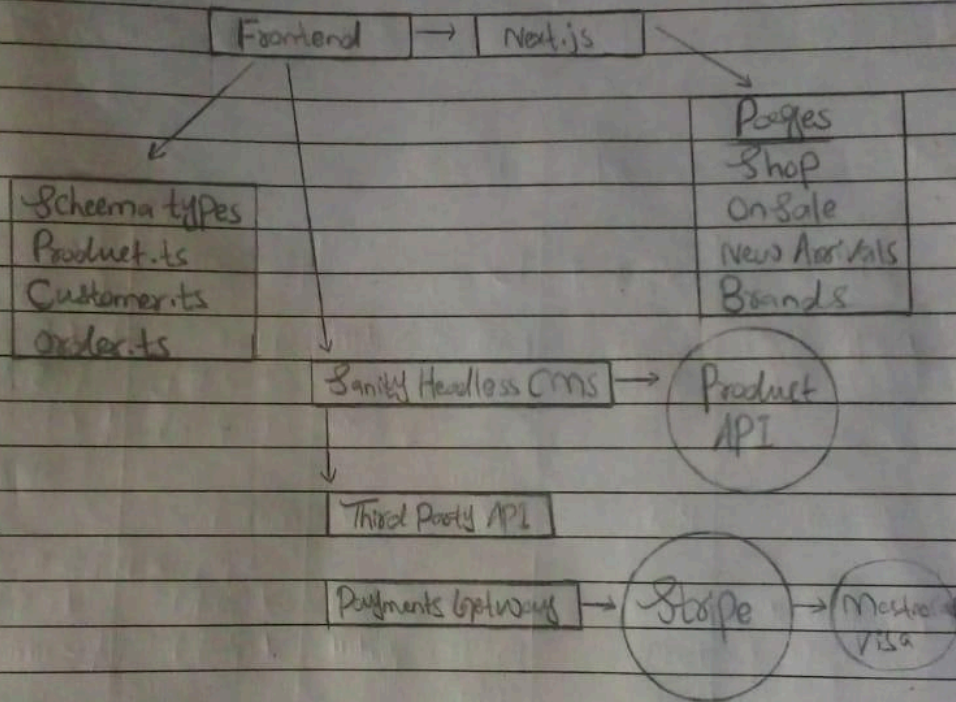


Marketplace Technical Foundation Shop.Co ECommerce

Date: _____

1) Frontend marketplace Requirements.

- User receives a notification and can leave a review.
- User-friendly interface for browsing Products.
- Responsive design for mobile and desktop users.
- Essential Pages: Shop, OnSale, new Arrivals, Brands.



Components and Roles :-

1- Frontend (Next.js) :-

1. Displays the user interface for browsing Products, managing the Carts and placing orders.
2. Handles user interactions, and communicates with backend services via APIs.

2- Sanity CMS :-

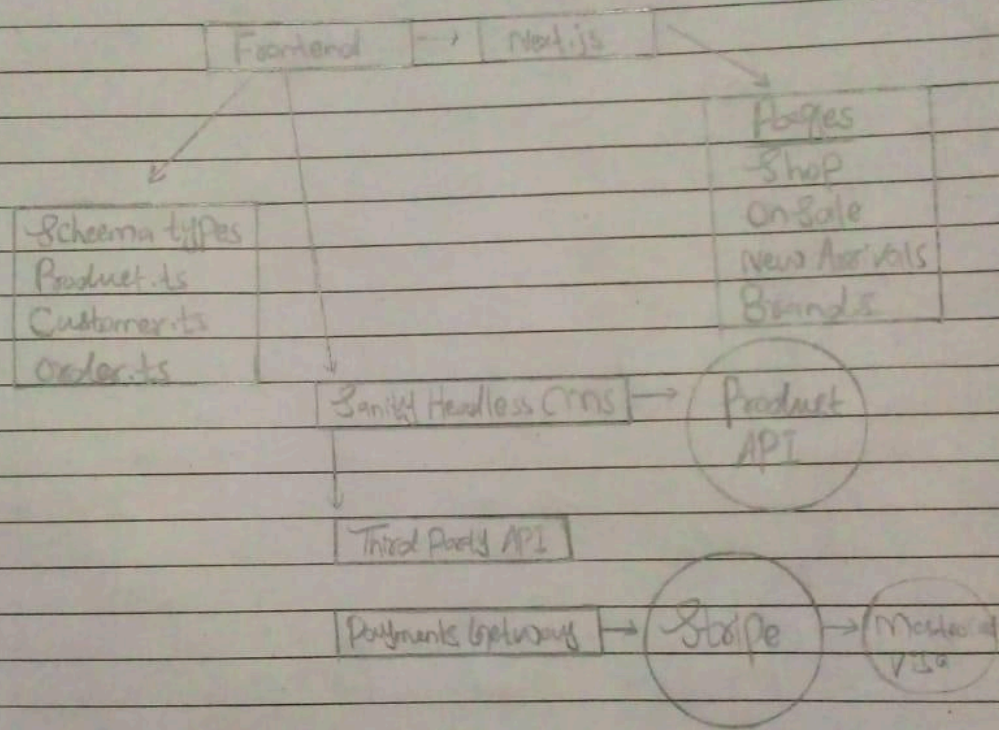
1. Acts as the primary backend to manage Product data, Customer details and order records.
2. Provides APIs for the frontend to fetch and update data.

Marketplace Technical Foundation SHOP.CO ECommerce

Date: _____

1) Frontend marketplace Requirements.

- User receives a notification and can leave a review.
- User-friendly interface for browsing Products.
- Responsive design for mobile and desktop users.
- Essential Pages: Shop, OnSale, new Arrivals, Brands.



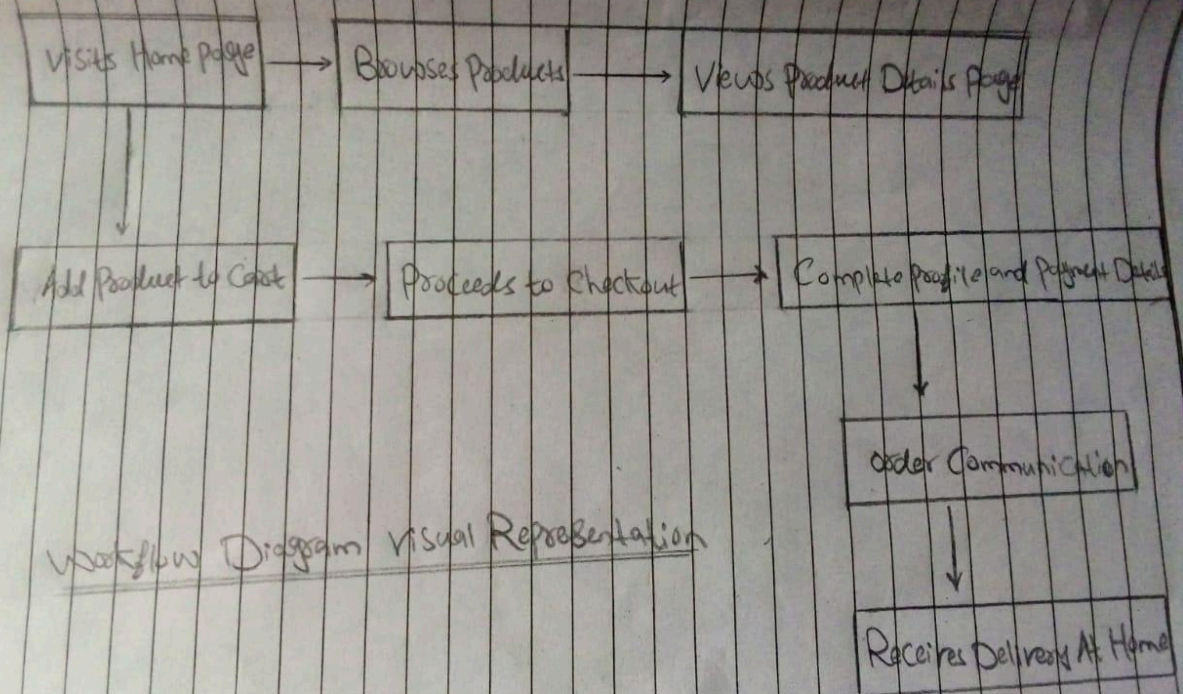
Components and Roles :-

1- Frontend (Next.js) :-

1. Displays the user interface for browsing Products, managing the Cart, and placing orders.
2. Handles user interactions and communicates with backend services via APIs.

2- Sanity CMS :-

1. Acts as the primary backend to manage Product data, Customer details and order records.
2. Provides APIs for the frontend to fetch and update data.



Workflow Diagram Visual Representation

Date:

3- Product API

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

4- Third Party APIs:-

1. Integrates services like shipment tracking and payment processing.

5. Payment Gateway:-

1. Processes user payments securely and provides transaction Confirmation.

API Endpoints

Endpoint	Method	Description	Parameters	Response Example
/api/products	Get	Fetch all products	None	{id:1, name: "t-shirt"}
/api/product/:id	Get	Fetch a single product item.	id (path)	{id:1, name: "t-shirt"}
/api/products	Post	Add a new product item.	name, price, category (body)	{success: true, id: 5}
/api/product/:id	Delete	Delete a product item.	id (path)	{success: true}
/api/product/:id	Put	update a product item.	id (path), name, price (body)	{success: true}
/api/categories	Get	Fetch all food categories	None	{categories: ["clothing"]}

Date: _____

3. Product API

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

4. Third Party APIs:-

1. Integrates services like shipment tracking and payment processing.

5. Payment Gateway:-

1. Processes user payments securely and provides transaction confirmation.

API Endpoints

Endpoint	Method	Description	Parameters	Response Example
/api/products	Get	Fetch all products	None	{id: 1, name: "t-shirt"}
/api/product/:id	Get	Fetch a single product item.	id (Path)	{id: 1, name: "t-shirt"}
/api/products	Post	Add a new product item.	name, Price, Category (body)	{Success: true, id: 5}
/api/product/:id	Delete	Delete a product item.	id (Path)	{Success: true}
/api/product/:id	Put	Update a product item.	id (Path), name, Price (body)	{Success: true}
/api/categories	Get	Fetch all product categories	None	{categories: ["Clothing"]}