Marketplace Technical Foundation

General E-Commerce Website

Overview

The furniture e-commerce website aims to provide a seamless online shopping experience for customers to browse and purchase furniture products. The website will feature a user-friendly interface, responsive design, and essential pages for a hassle-free shopping experience.

Technical Planning

Frontend

- Built using Next.js
- User-friendly interface for browsing and purchasing furniture products
- Responsive design for mobile and desktop users

Essential pages:

- **Home**: Website's main page, showcasing featured products and promotions
- **Product Listing**: Page displaying a list of products, with filtering and sorting options
- **Product Details**: Page displaying detailed information about a single product
- **Cart**: Page displaying items added to the cart, with options to update or remove items
- **Checkout**: Page facilitating the payment process, with options for shipping and payment methods
- **Order Confirmation**: Page displaying confirmation of a successful order, with order details and tracking information

Technologies: HTML, CSS, Figma

Backend

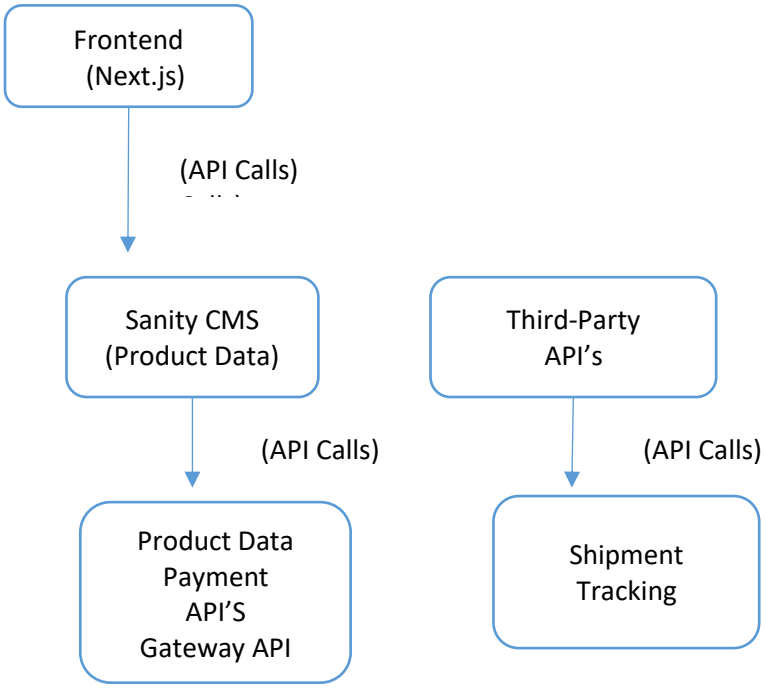
- Built using Sanity CMS
- Manages product data, customer details, and order records
- Schemas designed to align with business goals
- Technologies: Sanity CMS, Mock APIs

Third-Party APIs

- Shipment Tracking API for real-time updates on order shipping and delivery
- Payment Gateway API for secure and efficient payment processing

- Other required backend services integrated as needed

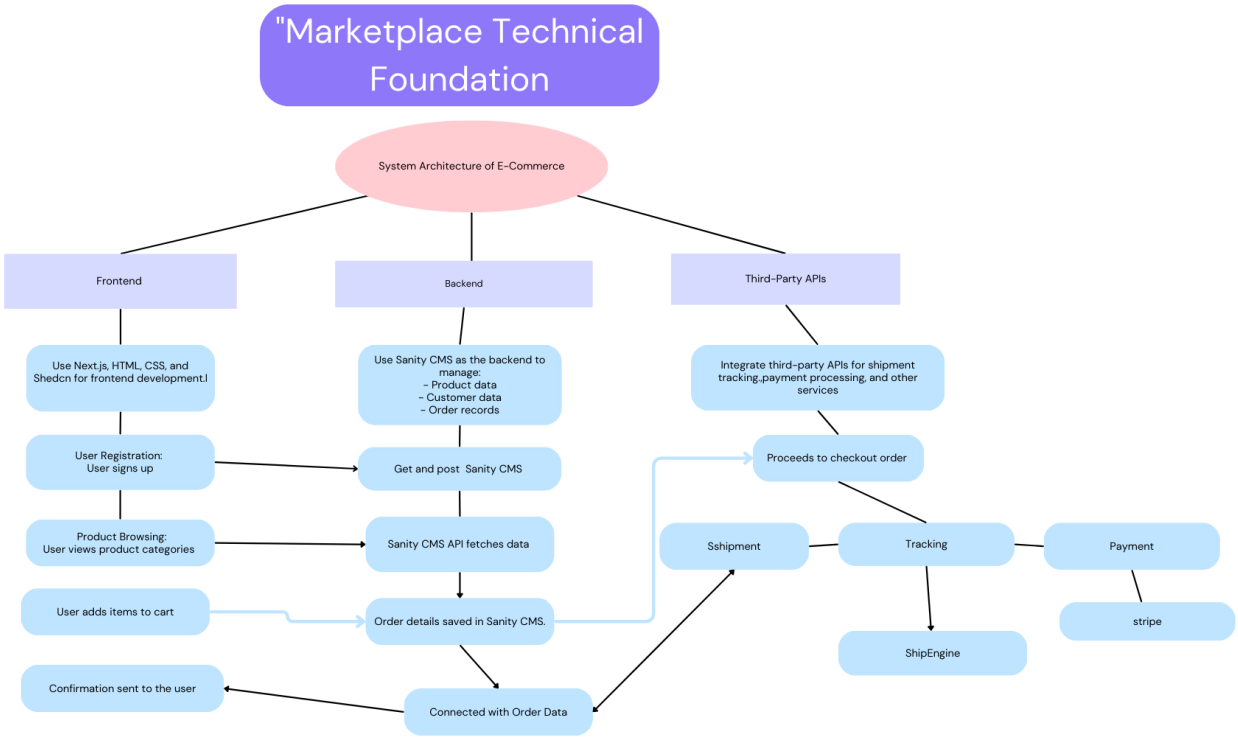
System Architecture Diagram



System Architecture Description

1. **Frontend (Next.js):** User-friendly interface for browsing products, responsive design for mobile and desktop users.
2. **Sanity CMS**: Manages product data, customer details, and order records. Acts as a single source of truth for all data.
3. **Third-Party APIs**: Provides additional functionality such as shipment tracking, payment gateway, and other required backend services.
4. **Product Data API**: Fetches product details from Sanity CMS and provides them to the frontend.
5. **Shipment Tracking API**: Fetches shipment tracking details from third-party APIs and provides them to the frontend.
6. **Payment Gateway API**: Processes payment for orders and provides payment status to frontend.

ExampleSystemArchitecture:



KeyWorkflows

1. **User Registration**: User signs up -> Data is stored in Sanity CMS -> Confirmation sent to the user.
2. **Product Browsing**: User views product categories -> Sanity CMS API fetches data -> Products displayed on frontend.
3. **Order Placement:** User adds items to cart -> Proceeds to checkout -> Order details saved in Sanity CMS.
4. **Shipment Tracking**: Order status updates fetched via Third-Party API -> Displayed to the user.

APIEndpoints

Api Name	End Points	Method	Purpose	Respond Example	Payload Example
Product API	/products	Get	Fetches all product details	{ id,name,price, Stock,image }	-
Products Detail API	/products/{id}	Get	Fetches product details by ID	{id,name,price, Stock,image,description	-

				Sizes,colours}	
Order Placement API	/orders	Post	Creates a new order	{customer_id,product_id,quantity, payment_method}	{order_id,status,total_cost }
Shipment Tracking API	/shipments/{order_id}	Get	Fetches shipment tracking details by order ID	{order_id, shipment_id , status,estimated_delivery_date}	-
Payment Gateway API	/payments	Post	Processes payment for an order	{Payment_id, Status, Transaction_id}	{order_id, shipment_id ,status,estimated_delivery_date}

AuthenticationAPI

Api Name	End Points	Method	Respond Example	Request Example
Authentication API	/api/auth/login	POST	email, password	token (JWT token), user (user details)
	/api/auth/user	GET	user (user details)	Authorization (JWT token)
Cart API	/api/cart	GET	(cart details), (cart items)	Authorization (JWT token)
	/api/cart/add	POST	Cart	product_id, quantity
Shipping API	/api/shipping/options /api/shipping/select	GET POST	(shipping options) (selected shipping details)	Option id shipping

DataSchemaDesign

Product and product Detail

- id (unique identifier): string
- name: string
- description: string
- price: number
- stock: number
- image: string (URL of the product image)
- category: string (e.g. "chair", "table", etc.)
- material: string (e.g. "wood", "metal", etc.)
- color: string (e.g. "black", "white", etc.)
-

Category

- id (unique identifier): string
- name: string
- description: string
- products: array of Product ids

Customer

- id (unique identifier): string
- name: string
- email: string
- password: string (hashed)
- address: string
- phone: string

Order

- id (unique identifier): string
- customer_id: string (foreign key referencing the Customer entity)
- order_date: date
- total: number
- status: string (e.g. "pending", "shipped", etc.)
- products: array of Product ids

Order Item

- id (unique identifier): string
- order_id: string (foreign key referencing the Order entity)
- product_id: string (foreign key referencing the Product entity)
- quantity: number
- price: number

Payment

- id (unique identifier): string
- order_id: string (foreign key referencing the Order entity)
- payment_method: string (e.g. "credit card", "paypal", etc.)
- payment_date: date
- amount: number

Shipment

- id (unique identifier): string
- order_id: string (foreign key referencing the Order entity)
- shipping_method: string (e.g. "UPS", "FedEx", etc.)
- shipping_date: date
- tracking_number: string

TechnicalRoadmap

Week 1: Planning,Design,andDevelopment

Day 1-2: PlanningandDesign

1. Define project scope and requirements
2. Conduct market research and competitor analysis
3. Create wireframes and prototypes of key pages

Day 3-4: FrontendDevelopment

1. Set up Next.js project structure and configuration
2. Develop reusable UI components using React
3. Implement responsive design and mobile-first approach

Day 5-6: BackendDevelopmentandIntegration

1. Set up Sanity CMS project structure and configuration
2. Design and implement data schemas for products and customers
3. Integrate frontend and backend using RESTful APIs

Day 7: TestingandDeployment

1. Conduct unit testing and integration testing
2. Perform cross-browser and device testing
3. Deploy application to production environment

Deliverables

- Wireframes and prototypes
- Frontend codebase
- Backend codebase
- Deployed application

AssumptionsandDependencies

- Availability of design assets and content
- Sanity CMS and Next.js documentation and support

RisksandMitigationStrategies

- Delays in design asset delivery: Regular communication with design team and flexible project planning
- Technical difficulties with Sanity CMS or Next.js: Research and documentation, community support, and contingency planning

Prepared and designed by
Kashaf Tariq