

Marketplace Technical Foundation [General E-Commerce]

Day 1 Overview

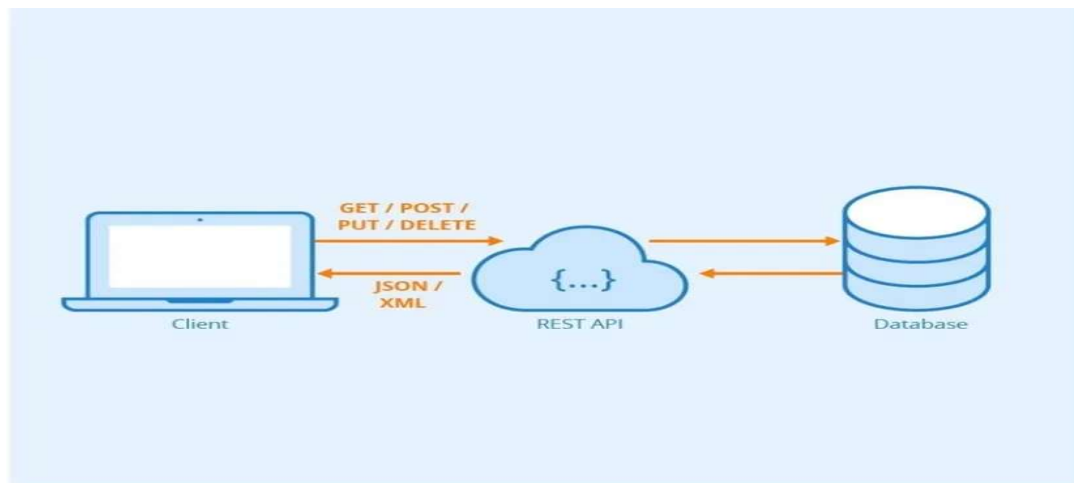
- **Marketplace Choice:** General e-commerce, Nike-themed website.
- **Business Goals:** Providing high-quality, affordable, customizable sportswear and accessories for various demographics.
- **Target Audience:** Men, women, unisex products, kids, boys, and girls.
- **Products and Services:** Shoes, sports bras, tops, t-shirts, hoodies, sweatshirts, jackets, trousers, tights, shorts, tracksuits, jumpsuits, rompers, skirts, socks, accessories, and equipment.
- **What Sets Marketplace Apart:** Speed, affordability, and customization.

Chart for Flow Recap:

Action (Button Clicked)	After Action
Add to Cart Button	Item added to cart; cart updates, option to view cart or continue shopping.
Checkout Member Button	Redirected to checkout page; member reviews order summary and fills in details.
How Would You Like to Get Your Order" Page	Member enters personal details, selects delivery preferences, confirms order.
Sign-In Button	Member logs in, redirected to dashboard/homepage, session created.
Sign-Out Button	Member logs out, redirected to homepage or login page.








🔧 Technical Foundation Overview Chart

Component	Technology/Tool	Role	Data Flow/Operations
Frontend (User Interface)	🌀 Next.js	Builds the React-based frontend	<ul style="list-style-type: none"> - Displays pages like Landing, Product, Order Summary, etc. - Makes API calls to fetch and manipulate data from the backend.
Backend (Content Management)	🔗 Sanity CMS	Headless CMS to manage content	<ul style="list-style-type: none"> - Stores and organizes product, category, and order data. - Allows retrieval and manipulation of data via API queries.
API Management	🔗 Next.js API Routes	Manages API requests and responses	<ul style="list-style-type: none"> - Defines endpoints for GET, POST, PUT, and DELETE operations. - Communicates with Sanity CMS to fetch and update data.






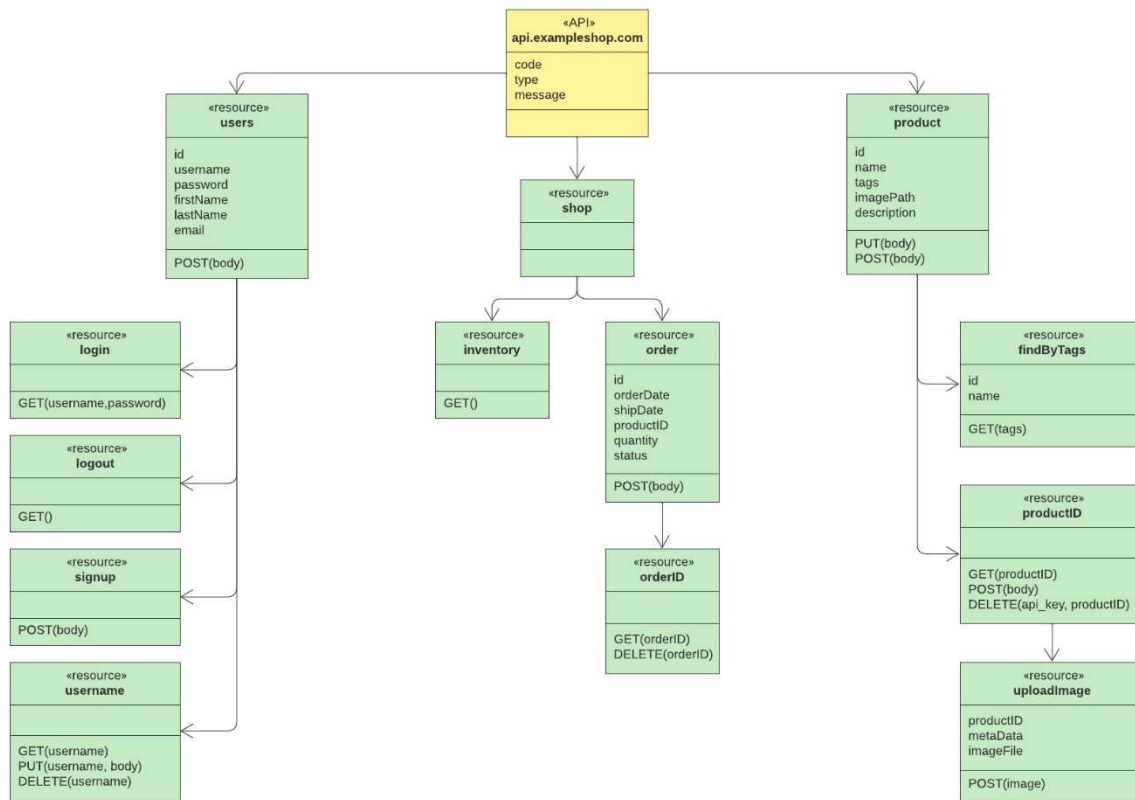
🔧 API Endpoints and Data Handling

Request Type	Endpoint	Operation	Data Flow
GET	🌐 /api/products	Fetches all product data	- Sends a request to Sanity CMS to retrieve product details.
GET	🌐 /api/orders	Fetches order data	- Retrieves order data from Sanity CMS.

Request Type	Endpoint	Operation	Data Flow
GET	 /api/users	Fetches user details	- Retrieves user information from Sanity CMS.
POST	 /api/orders	Creates a new order	- Sends new order data to the backend (Sanity CMS) for storage.
POST	 /api/products	Adds a new product	- Sends new product details to Sanity CMS for storage.
PUT	 /api/products/{id}	Updates an existing product	- Sends updated data for a specific product to Sanity CMS.
PUT	 /api/orders/{id}	Updates an existing order	- Sends updated data for a specific order to Sanity CMS.
DELETE	 /api/products/{id}	Deletes a product	- Sends a delete request to remove a product from Sanity CMS.
DELETE	 /api/orders/{id}	Deletes an order	- Sends a delete request to remove an order from Sanity CMS.

System Architecture Overview

Layer	Technology/Tool	Role
 Frontend (Client-side)	Next.js	- Renders the UI and makes requests to the API. - Displays dynamic content (products, orders, user details).
 API Layer	Next.js API Routes	- Handles requests (GET, POST, PUT, DELETE). - Retrieves or updates data from Sanity CMS.
 Backend (Server-side)	Sanity CMS	- Stores and organizes content. - Provides an API for querying and manipulating data.



🚦 User Journey Overview:

The user experience flows seamlessly through actions like adding items to the cart, checking out, and choosing delivery preferences. Login and sign-out functionalities ensure secure access, while the cart system facilitates easy management of products.

🚦 Technical Foundation:

- **Frontend (Next.js):** The **React-based frontend** handles page rendering and communicates with the backend via API calls to fetch and manipulate product and order data.
- **Backend (Sanity CMS):** **Sanity CMS**, as a headless CMS, organizes and manages content like products and orders while providing an API to access and update data.
- **API Management (Next.js API Routes):** Handles various CRUD operations for product and order data through **GET**, **POST**, **PUT**, and **DELETE** requests, enabling smooth data interaction between the frontend and backend.

API and Data Handling:

- **GET Requests:** Fetch product, order, and user data from the backend.
- **POST Requests:** Add new products or orders to the system.
- **PUT Requests:** Update existing product or order details.
- **DELETE Requests:** Remove products or orders from the system.

- **System Architecture:**
 - **Frontend** (Next.js) displays dynamic content and communicates with the backend.
 - **API Layer** (Next.js API Routes) facilitates data retrieval and updates.
 - **Backend** (Sanity CMS) stores and organizes data, offering an API for easy data access.

This cohesive architecture ensures that the e-commerce site is not only functional but also scalable and flexible, providing a smooth shopping experience for users while offering efficient content management for admins.