

Day 2: Planning the Technical Foundation

Marketplace Name: Digital Accessories Hub

Technical Requirements

1. Frontend Requirements:

- Develop a user-friendly interface using Next.js.
- Ensure responsive design for optimal viewing on mobile, tablet, and desktop.
- Include essential pages:
 - Home: Highlights featured products.
 - Product Listing: Displays all available products by category.
 - Product Details: Detailed view with descriptions, images, and pricing.
 - Cart: Shows selected items with pricing summary.
 - Checkout: Collects customer details and processes payments.
 - Order Confirmation: Displays order details and status.

2. Backend Requirements:

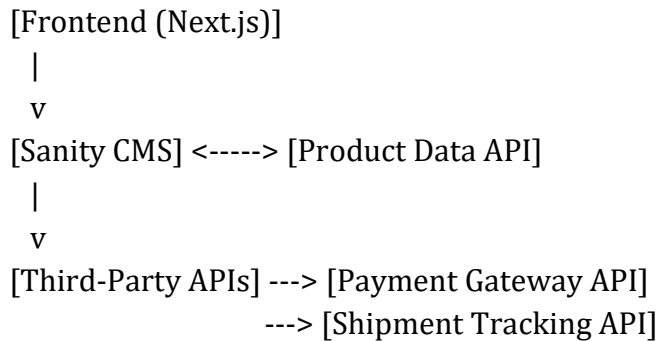
- Use Sanity CMS for:
 - Managing product details (name, price, stock, category).
 - Tracking customer orders and information.
 - Storing shipment and delivery statuses.
- Integrate third-party APIs for:
 - Payment Gateway: Secure online payments.
 - Shipment Tracking: Updates on order delivery.
 - Email Notifications: Order confirmation and updates.

3. Security Requirements:

- Implement HTTPS for secure communication.
- Use authentication for CMS access to prevent unauthorized modifications.
- Validate user input to ensure data integrity.

System Architecture

High-Level Diagram:



API Requirements

1. Product APIs:

- Endpoint: /products
- Method: GET
- Description: Retrieve a list of all products.
- Response Example:

```
[
  { "id": 1, "name": "Wireless Mouse", "price": 20 },
  { "id": 2, "name": "Gaming Keyboard", "price": 70 }
]
```

2. Order APIs:

- Endpoint: /orders
- Method: POST
- Description: Create a new order.
- Payload Example:

```
{
  "customerName": "John Doe",
  "products": [
    { "id": 1, "quantity": 2 },
    { "id": 2, "quantity": 1 }
  ],
  "paymentStatus": "Completed"
}
```
- Response Example:

```
{ "orderId": 12345, "status": "Order Placed" }
```

Sanity CMS Schemas

1. Product Schema:

```
export default {  
  name: 'product',  
  type: 'document',  
  fields: [  
    { name: 'name', type: 'string', title: 'Product Name' },  
    { name: 'price', type: 'number', title: 'Price' },  
    { name: 'category', type: 'string', title: 'Category' },  
    { name: 'tags', type: 'array', of: [{ type: 'string' }], title: 'Tags' }  
  ]  
};
```