

# Day 2 Hackathon Document: Technical Planning for FashionNest



## 1. Technical Plan Aligned with Business Goals

### Business Goals:

- Offer **affordable, high-quality women's clothing** to combat inflation-driven market challenges.
- Create a seamless shopping experience for **time-conscious and value-seeking customers**.
- Build a **user-friendly e-commerce website** with reliable delivery options and competitive pricing.

### Key Features to Achieve Goals:

- **Shopping Experience:** Intuitive product navigation, size guides, and smart product recommendations.
- **Purchase and Payments:** Secure payment methods, tax calculations, and discount handling.
- **Order Management:** Real-time tracking, easy returns, and multiple delivery preferences.

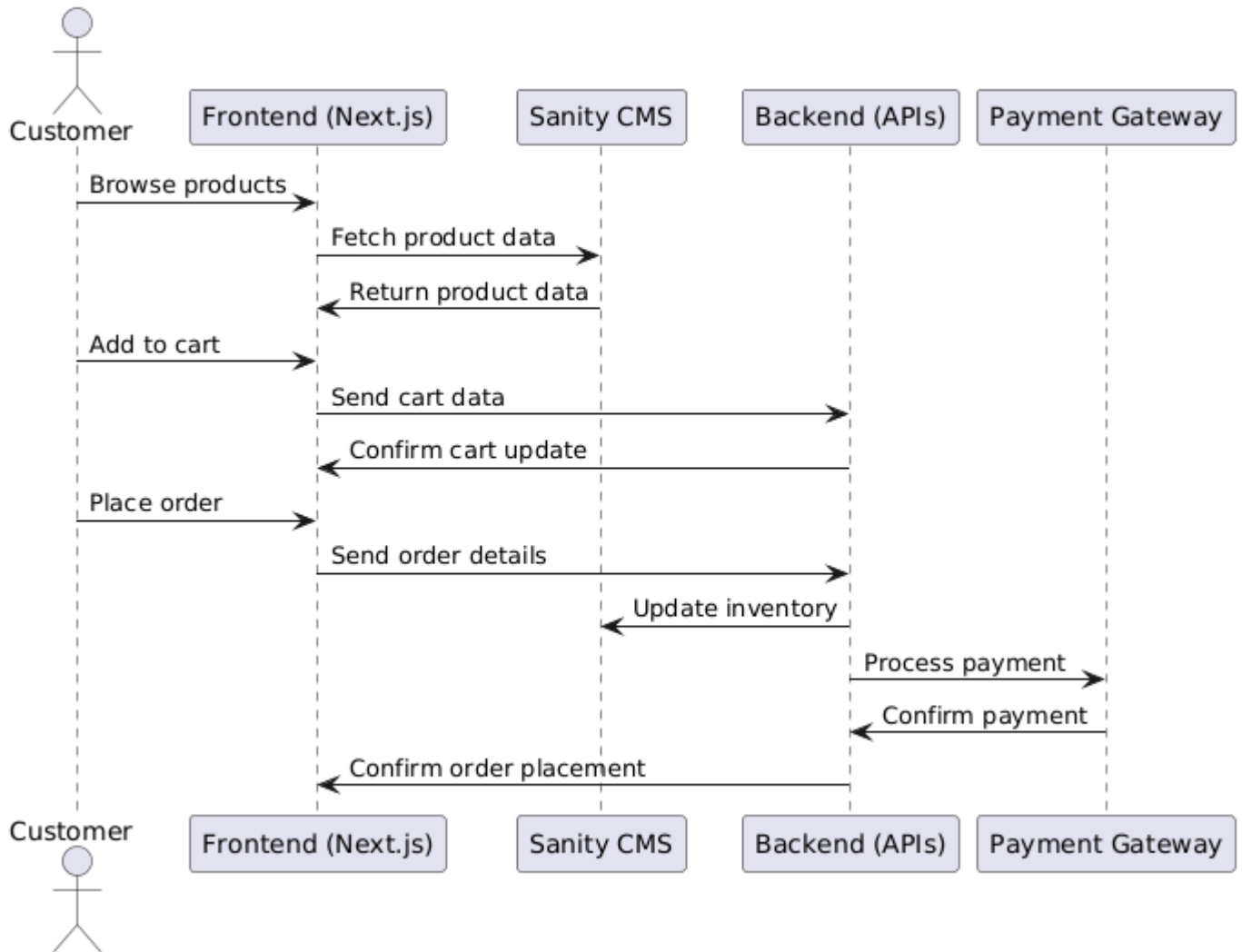
## 2. System Architecture Visualized

A **visual architecture diagram** demonstrates the interaction between:

1. **Frontend (Next.js):** Manages UI and customer-facing workflows like browsing, adding to cart, and checkout.
2. **Sanity CMS:** Handles product data, inventory management, and content updates.
3. **Third-party APIs:** Manages user authentication, order processing, and payment gateway.

### Diagram Key Workflows:

- **Product Browsing:** Fetch products from the Sanity CMS using APIs.
- **Cart Management:** Add/remove products via APIs, update UI state.
- **Order Placement:** Securely send customer and order details to the backend.



### **3. Detailed API Requirements Documented**

#### **API Endpoints:**

Endpoint	Methods	Description	Example
/api/auth	POST, GET	Handles login, registration, and session validation.	POST /api/auth {email, password} → {token}
/api/products	GET	Fetches product data from Sanity CMS.	GET /api/products?category=women → [{...}]
/api/orders	POST, GET	Places new orders, fetches order history.	POST /api/orders {cartItems} → {orderId}
/api/search	GET	Provides search and filtering options.	GET /api/search?q=dresses → [search results]

## Key Features

### Shopping Experience

- Advanced search filters for quick product discovery.
- Smart recommendations for users based on browsing history.
- Real-time inventory updates and detailed product descriptions.

### Payment Options

- Secure checkout.
- Multiple payment methods (credit/debit cards, digital wallets).

### Order Management

- Real-time order tracking.
- Hassle-free return and exchange processes.

### User Features

- Wishlist management.
- Personalized recommendations.
- Saved addresses for quick checkouts.

## 4. Sanity Schemas Drafted

### Product Schema

```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    { name: 'title', title: 'Title', type: 'string' },
    { name: 'description', title: 'Description', type: 'text' },
    { name: 'price', title: 'Price', type: 'number' },
    { name: 'category', title: 'Category', type: 'reference', to: [{ type: 'category' }] },
    { name: 'stock', title: 'Stock', type: 'number' },
    { name: 'image', title: 'Image', type: 'image', options: { hotspot: true } },
  ],
};
```

### Order Schema

```
export default {
  name: 'order',
  title: 'Order',
  type: 'document',
  fields: [
    { name: 'orderId', title: 'Order ID', type: 'string' },
    { name: 'customer', title: 'Customer', type: 'reference', to: [{ type: 'customer' }] },
    { name: 'orderItems', title: 'Order Items', type: 'array', of: [{ type: 'product' }] },
    { name: 'totalPrice', title: 'Total Price', type: 'number' },
    { name: 'status', title: 'Status', type: 'string' },
    { name: 'orderDate', title: 'Order Date', type: 'datetime' },
  ],
};
```

### Customer Schema

```
export default {
  name: 'customer',
  title: 'Customer',
  type: 'document',
  fields: [
    { name: 'name', title: 'Name', type: 'string' },
    { name: 'email', title: 'Email', type: 'string' },
    { name: 'phone', title: 'Phone', type: 'string' },
    { name: 'address', title: 'Address', type: 'text' },
    { name: 'orders', title: 'Orders', type: 'array', of: [{ type: 'order' }] },
  ],
};
```

### Category Schema

```
export default {
  name: 'category',
  title: 'Category',
```

```
type: 'document',  
fields: [  
  { name: 'name', title: 'Name', type: 'string' },  
  { name: 'description', title: 'Description', type: 'text' },  
,  
];
```

### Next Steps

- Implement the **frontend design** using Next.js.
- Integrate the **Sanity CMS** for managing product data.
- Begin developing APIs for product browsing, order placement, and cart management.