

# Comforty Complete Documentation

## ## Comforty E-Commerce Marketplace

### ### Introduction

Comforty is an innovative general e-commerce platform dedicated to providing a seamless shopping experience for high-quality furniture. Its mission is to bring comfort, style, and affordability to individuals and businesses through an intuitive, user-friendly platform. By offering a diverse range of products and services, Comforty bridges the gap between customer needs and market offerings.

**\*\*Prepared by\*\*:** Syed Muhammad Shan-e-Ali

**\*\*Participating in Hackathon\*\*:** Held on 15th, 2025

**\*\*GAIAC Slot\*\*:** Friday, 7 PM to 10 PM

### ### Table of Contents

1. Introduction
2. Project Description
3. Technical Requirements

# Comforty Complete Documentation

4. System Architecture
5. Data Schema Design
6. Workflow Diagrams
7. API Integration and Data Migration
8. Implementation Roadmap
9. Conclusion

## ### Project Description

Comforty is designed to provide an intuitive and scalable e-commerce platform. Here are the main features and roles defined:

- **\*\*Member Functionality\*\***:
- Registered members can view and purchase multiple products regardless of quantity.
- Visitors can browse and compare products but cannot make purchases.
- A "Contact Us" page enables users to reach out to the admin for support.

# Comforty Complete Documentation

## - **User Roles**:

1. **Visitors**: View and compare products.
2. **Users**: Can view, compare, and purchase products.
3. **Operators**: Manage backend operations and support.
4. **Admins**: Full privileges to add/remove products, manage users, and oversee orders.

## ### Technical Requirements

### 1. **Frontend Requirements**:

- **Framework**: Next.js with TypeScript and Tailwind CSS for responsive design.
- **Essential Pages**: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation.

### 2. **Backend Requirements**:

- **Sanity CMS**:
- Manages product data, customer profiles, and order records.
- Schema-driven content for seamless integration.

# Comforty Complete Documentation

- **Third-Party APIs**:
  - Shipment Tracking API for real-time order updates.
  - Payment Gateway for secure transactions.

## ### System Architecture

### 1. **Architecture Diagram**:

- **Frontend**: Built with Next.js, handling user interactions like browsing, adding to the cart, and checkout.
- **Backend**: Sanity CMS for managing product and order data.
- **Third-Party APIs**: Includes shipment tracking and payment gateway integrations.

### 2. **Data Flow**:

1. User interacts with the frontend.
2. Frontend fetches data from Sanity CMS.
3. APIs manage payment and shipment updates, ensuring seamless communication.

# Comforty Complete Documentation

## ### Data Schema Design

### 1. **\*\*Product Schema\*\***:

```
```javascript

export default {

  name: 'product',

  type: 'document',

  title: 'Product',

  fields: [

    { name: 'name', type: 'string', title: 'Product Name' },

    { name: 'price', type: 'number', title: 'Price' },

    { name: 'category', type: 'string', title: 'Category' },

    { name: 'stock', type: 'number', title: 'Stock' },

    { name: 'image', type: 'image', title: 'Image', options: { hotspot: true } },

    { name: 'description', type: 'text', title: 'Description' }

  ]

}
```

# Comforty Complete Documentation

...

## 2. **\*\*Order Schema\*\***:

```
```\javascript
```

```
export default {
```

```
  name: 'order',
```

```
  type: 'document',
```

```
  title: 'Order',
```

```
  fields: [
```

```
    { name: 'customerId', type: 'reference', to: [{ type: 'customer' }], title: 'Customer ID' },
```

```
    { name: 'products', type: 'array', of: [{
```

```
      type: 'object',
```

```
      fields: [
```

```
        { name: 'productId', type: 'reference', to: [{ type: 'product' }], title: 'Product ID' },
```

```
        { name: 'quantity', type: 'number', title: 'Quantity' }
```

```
      ]
```

```
    }], title: 'Products' },
```

## Comforty Complete Documentation

```
{ name: 'totalPrice', type: 'number', title: 'Total Price' },

{ name: 'deliveryAddress', type: 'string', title: 'Delivery Address' },

{ name: 'status', type: 'string', title: 'Order Status', options: { list: [

{ title: 'Pending', value: 'pending' },

{ title: 'In Progress', value: 'in_progress' },

{ title: 'Delivered', value: 'delivered' }

] } },

{ name: 'createdAt', type: 'datetime', title: 'Order Placed At' }

]

}

...
```

### ### API Integration and Data Migration

#### 1. \*\*API Overview\*\*:

- APIs are used to populate Sanity CMS with data from external sources like Shopify, Magento, WooCommerce, and WordPress.

# Comforty Complete Documentation

- APIs provided for templates enable fetching product listings, categories, and order data.

## 2. **Steps for Integration**:

- Review API documentation to identify key endpoints.
- Validate and adjust schema to match API data structure.
- Write migration scripts or manually import data using Sanity tools.
- Test API integration by rendering data in Next.js components.

## 3. **Best Practices**:

- Use `.env` files for API keys.
- Test endpoints with tools like Postman.
- Modularize utility functions for scalability.

## ### Implementation Roadmap

### 1. **Steps**:

- Sanity CMS setup



# Comforty Complete Documentation

- Frontend development with Next.js
- API integration
- Testing and deployment

## 2. **Timeline**:

- **Phase 1**: Schema Design (2 days)
- **Phase 2**: Frontend Pages (3 days)
- **Phase 3**: API Integration (2 days)

## **### Conclusion**

Comforty delivers a seamless shopping experience with modern technology. It redefines furniture e-commerce by integrating user-friendly design, efficient logistics, and secure transactions.