Comforty

Marketplace Technical Foundation

Syed Muhammad Shan-e-Ali
(GIAIC Friday 7pm)

Table of Contents

INTRODUCTION

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.

The objective of this project is to develop a general purpose e-commerce store where any kind of product can be bought from the comfort of home through the Internet. However, for implementation purpose, this paper will deal with an online shopping for Comforty is a furniture e-commerce platform offering various categories like: Wing Chair, Arm Chair, Sofas, Park Benchs etc..

An online store is a virtual store on the Internet where customers can browse the catalog and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. At that time, more information will be needed to complete the transaction. Usually, the customer will be asked to fill or select a billing address, a shipping address, a shipping option, and payment information such as Easypaisa, or Cash on Delivery. An e-mail notification is sent to the customer as soon as the order is placed.

PROJECT DESCRIPTION

- Any member can register and view available products.
- Only registered member can purchase multiple products regardless of quantity.
- Contact Us page is available to contact Admin for quries.
- There are four roles available:
  - Visitor
  - User
  - Operator
  - Admin

- Visitor can view, search, compare available products.
- User can view and purchase products.
- Operator can excess and option to add or can view everything of admin panel.
- An Admin has some extra privilege including all privilege of visitor and user.
  - ✓ Admin can add products, edit product information and add/remove product.
  - ✓ Admin can add user, edit user information and can remove user.
  - ✓ Admin can ship order to user based on order placed by sending confirmation mail.

TECHNICAL REQUIREMENTS

1. FrontEnd Requirements:

- Next.js Framework with TypeScript, Tailwind
- Responsive design for mobile and desktop
- App Routes
  - Home, Product Listing, Product Details, Cart, Checkout, Order Confirmation.

2. BackEnd Requirements:

- Sanity CMS for managing.
  - Product Data (Categories, Price, Stock)
  - Customer Data (Profiles, Order History)
  - Order Data (Products, Quantities, Status)

3. Third-Party APIs:

- Shipment Tracking API for real-time delivery status.
- Payment Gateway for secure transactions.

SYSTEM ARCHITECTURE

1. Architecture Diagram:

[Frontend: Next.js with TypeScript]
|
|----- User Interaction
| (Browse, Add to Cart, Checkout)
|
|
[Backend: Sanity CMS]
|
|------ Product Data Management
|------ Customer Data Storage
|------ Order Processing
|
[Third-Party APIs]
|
|------ Shipment Tracking API (Real-time Order Status)
|------ Payment Gateway (Secure Payments)
|

2. Data Flow:

- User interacts with the frontend.
- Frontend fetches data from Sanity CMS
- APIs handle payment and shipment update

API SPECIFICATION

1. Products API

Endpoint : /products Method : GET

Purpose : Retrieve all available products.

Request : No parameters required

Response :

```
[ {
    "id" : "101",
    "name" : "Arm Chair",
    "price" : 150,
    "category" : "Furniture",
    "stock" : 20,
    "image" : "url-to-image",
    "description" : "A comfortable arm chair",
  } same as second object
]
```

2. Orders API

Endpoint : /orders Method : POST

Purpose : Create a new order

Request Payload :

```
{"customerId" : "202",
 "Products" : [
  {"productId":"101","quantity": 2},
  {"productId" :"102","quantity" : 1}
 ],
"totalPrice" : 420,
"deliveryAddress" :"123 Main street"
}
```

Respone :

```
{
"orderId" : "98765",
"status" : "Order placed",
"estimatedDelivery":"2025-01-20",
}
```

3. Shipment Tracking API

Endpoint : /shipment-status Method : GET

Purpose : Retrieve the shipment status of an order.

Request Parameters : OrderId : The unique ID of the order

Response :

```
{ "orderId" : "98765",
  "status" : "In Transit",
  "expectedDelivery" :"2025-01-20",
  "lastUpdate": 2025-01-15 14:30:00",
  }
```

4. Customer Authentication APIs

4.1 Sign - Up API

Endpoint : /signup Method : POST

Purpose : Register a new customer

Request Payload :

```
{"name" : "Shan-e-Ali",
"email" : "smsali100@gmail.com",
"password":"secure password",
"phone" : "1234567890",
"address":"123- Orangi Town",
}
```

Response

```
{"message" : Registration Success,
"customerId":"202",
}
```

4.2 Sign - In API

Endpoint : /signin Method : POST

Purpose : Authenticate an existing customer

Request Payload :

```
{"email" : smsali100@gmail.com",
"password":"1234567890"
}
```

Response

```
{"token" : "eyJABcdMN29A!..,
"customerId":"202"
}
```

4.3 Token Validation API

Endpoint: /validate-token Method: GET

Purpose: Validate the JWT token of the user session.

Headers: • Authorization: Bearer <JWT token>

Response:

```
{ "isValid": true,
  "customerId": "202"
}
```

5. Payment Gateway API

Endpoint: /process-payment Method: POST

Purpose: Process payment for an order

Request Payload:

```
{"orderId": "98765",
 "amount": 420,
 "paymentMethod": "credit card",
 "card Details": {
  "card Number": "4111111111111",
  "expiry Date": "12/25",
  "cvv": "123",
        }
}
```

Response:

```
{"transactionID": "TXN12345",
 "status": "success",
 "amount": 420,
 "order ID": "98765",
}
```

6. API Security and Validation

1. Authentication. — JWT based authentication
2. Input validation — all incoming request-
3. Rate limit — prevent abuse of APIs
4. Error Handling — descriptive error message

DATA SCHEMA DESIGN

1. Product Schema

```
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' ,
     options:{
        list:[
          { title :'Arm Chair' ,  value :'arm_chair'},
          { title :'Desk chair' ,  value :'desk_chair'},
          {title:'Wing chair' ,  value:'wing_chair' },
          {title:'Park Bench' ,  value :'park_bench'},
          {title:'Wooden chair',  value :'wooden_chair'},
          {title:'Sofas'    ,   value:'sofas' },
              ],
          },
    },
    {name:'stock' , type:'number' , title:'Stock' , },
    {name :'image' , type:'image' , title: Image    ,
      options: { hotspot : true, },
    },
    {name:'description', type:'text'    , title:'Description',},
        ],
            };
```

2. Order Schema

```
export default {
  name: 'order', type: 'document', title: 'Order',
  fields: [
    { name: 'customerId', type: 'reference', title: 'Customer ID',
      to: [ { type: 'customer' } ],
    },
    { name: 'products', type: 'array', title: 'Products',
      of: [ { type: 'object',
              fields: [
        { name: 'productId', type: 'reference', title: 'Product ID',
          to: [ { type: 'product' } ],
        },
        { name: 'quantity', type: 'number', title: 'Quantity' },
                    ],
                },
            ],
    },
    { name: 'totalPrice', type: 'number', title: 'Total Price' },
    { name: "deliveryAddress", type: 'string', title: "Delivery Address" },
    { name: 'status', type: 'string', title: Order status,
      option: { List: [
                  { title: 'Pending', value: 'pending' },
                  { title: 'In Progress', value: 'in_progress' },
                  { title: 'Delivered', value: 'delivered' },
              ], }, },
    { name: 'createdAt', type: 'datetime', title: 'Order Place At', },
  ], },
```

3. Customers Schema

```
export default {
  name :'customer', type :'document', title :'Customer',
  fields :[
   { name:'name'    , type: 'string'   , title: 'Name Full' ,},
   { name :'email'  , type: 'string'   , title :'Email Address',},
   { name : 'password', type: 'string' , title :'Password (Hased)',},
   { name :'phone'  , type: 'string'   , title :'Phone Number',},
   { name : 'address', type:'text'     , title :'Address'  ,},
   { name :'orderHistory' , type :'array' , title :'Order History',
     of :[ { type :'reference' , to :[{type :'order'}] }],
  }, ],     };
```

4. Shipments Schema

```
export default {
  name :'shipment'   , type: 'document' ,title :'Shipment'  ,
  fields : [
   { name: 'orderId'   , type: 'reference' ,title: 'Order ID'  ,
     to : [ { type :'order' } ], },
   { name: 'status'   ,type: 'string'  ,title :'Shipment status',
     option: {  list : [
                  { title :'In Transit' , value : 'in_transit'},
                  { title :'Delivered'  , value : 'delivered' },
                  { title: 'Pending'    , value : 'pending'   },
              ], }, },
   { name :'expectedDelivery', type : 'datetime' , title :'Expected Delivery'},
   { name :'lastUpdate' ,type: 'datetime' , title :'Last Update' },
],  },
```

WORKFLOW DIAGRAMS

1. User Journey:

- Browsing → Adding to Cart → Checkout → Order Tracking

2. Order Processing:

- Confirmation Order → Payment → Shipment Update → Delivery.

IMPLEMENTATION ROADMAP

1. Steps:

- Sanity CMS setups
- Frontend development with Nextjs
- 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

Summary:

- Comforty delivers a seamless shopping experience with modern technology.