

DRY #2 FORCKATHON-3

TEHNICAL FOUNDATION STRUCTURE

To build my marketplace, we need Some technical, and foundational Structures, that help our e-commerce Site in the front-end, back-end and stuck backend.

Key Technologies

- Frontend Next.js
- Content Management System (CMS): Sanity
- Order Tracking and Shipment
- ShipEngine Authentication
- Payment Gateway
- Stripe Hosting and Deployment: Vercel

Front-end:

framework: Choose React, Nextjs, Angular, or vue.js

UI Components:

Build reusable components (eg., Forms buttons, modals).

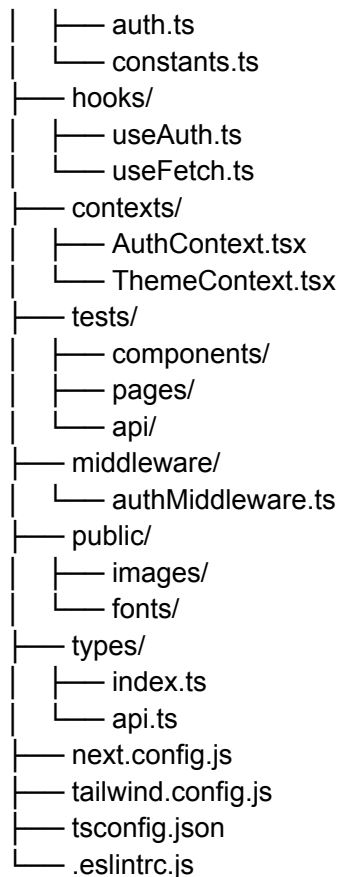
State-Management:

Implement state management with Redux, Vuen, or context API.

Routing:

use Client-side routing (React Router vue Roudner) for navigation.

```
project-root/
├── components/
│   ├── Button.tsx
│   ├── Header.tsx
│   ├── Footer.tsx
│   └── Modal.tsx
├── pages/
│   ├── api/
│   ├── auth/
│   ├── dashboard.tsx
│   ├── index.tsx
│   ├── about.tsx
│   └── contact.tsx
├── styles/
│   ├── globals.css
│   ├── tailwind.css
│   └── theme.ts
├── utils/
│   └── fetcher.ts
```



Backend Architecture:

API Design: Develop RESTfull or GraphQL APIS.

Business Logic:

Implement core business rules and logic within the backend.

Authentication:

Lise JWT, OAuth 2.0 for secure User authentication and authorization.

Frontend Requirements:

user Interface:

user-friendly interface: A clean and intuitive design that helps users easily browse products.

Responsive Design:

A design that displays well on both mobile and desktop platform.

Essential Pages:

Home page: The first page website home we'll create will be page, This will display some statics text describing the site along with dynamically.

1.Product Listing:

A Product listing is a description or display of a product that is available for sale.

What does the product listing include?

- product name
- price
- Image
- Description
- Rating & Reviews
- Filters
- Sorting option
- Product Category
- Calls to action

Product Detail :

Product details are the information about a product that is used to describe it and help customers make purchasing decisions.

Where can you find product details?

- Product details page.
- Product Packaging
- Product Catalogs

What information is include imp nat information product details?

- features
- Benefits
- Price
- Shipping option
- size
- Color
- Material
- Review

Cart:

Website that allow customers to store products before Checking out.

What does a cart do?

- Allows customers to add, remove and update the quantity.
- Make it easy for customers to edit their Cart

Checkoust:

A payment processing Page where users can confirm their I Orders.

Order Confirmation:

Confirmation message with order details and tracking information.

Key Workflows:

User Registration:

User signs up on the frontend.

Data sent to Sanity CMS via API.

Confirmation sent back to the user.

Product Browsing:

User views product categories.

Frontend calls Sanity CMS API for product data.

Products dynamically displayed.

Order Placement

User adds products to the cart and checks out.

Order details sent to Sanity CMS via API.

Order confirmation displayed to the user.

Shipment Tracking:

Order status updated using a third-party API.

Status fetched via API and shown to the user.

Sanity CMS as Backend:

What is Sanity?

Sanity is a modern "headless CMS" content management system. with Sanity you can manage your text, images, and other media with Apis.

Schema Data:

Product Schemas

- title (string)
- description (text)
- Price (number)
- Images (array of images)
- Category (reference to a category)
- stock (number)

Customer Schema

- name (string)
- email (email)
- address (object with street, city, state, Postal code)
- Order History (array of reference to order)

Order Schema:

- Order Id (string)
- Customer
- Products
- OderDate
- status
- TotalAmount

// Product Schema

```
export const productSchema = {
  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: 'images', title: 'Images', type: 'array', of: [{ type: 'image' }] },
    { name: 'category', title: 'Category', type: 'reference', to: [{ type: 'category' }] },
    { name: 'stock', title: 'Stock', type: 'number' },
    { name: 'sku', title: 'SKU', type: 'string' }
  ]
};
```

// Customer Schema

```
export const customerSchema = {
  name: 'customer',
  title: 'Customer',
  type: 'document',
  fields: [
    { name: 'name', title: 'Name', type: 'string' },
    { name: 'email', title: 'Email', type: 'email' },
    { name: 'phone', title: 'Phone', type: 'string' },
    { name: 'address',
      title: 'Address',
      type: 'object',
      fields: [
        { name: 'street', title: 'Street', type: 'string' },
        { name: 'city', title: 'City', type: 'string' },
        { name: 'state', title: 'State', type: 'string' },
        { name: 'postalCode', title: 'Postal Code', type: 'string' }
      ]
    },
    { name: 'orderHistory', title: 'Order History', type: 'array', of: [{ type: 'reference', to: [{ type: 'order' }] }] }
  ]
};
```

// Order Schema

```
export const orderSchema = {
  name: 'order',
  title: 'Order',
  type: 'document',
  fields: [
    { name: 'orderId', title: 'Order ID', type: 'string' },
```

```

{ name: 'customer', title: 'Customer', type: 'reference', to: [{ type: 'customer' }] },
{ name: 'products',
  title: 'Products',
  type: 'array',
  of: [
    {
      type: 'object',
      fields: [
        { name: 'product', title: 'Product', type: 'reference', to: [{ type: 'product' }] },
        { name: 'quantity', title: 'Quantity', type: 'number' },
        { name: 'price', title: 'Price', type: 'number' }
      ]
    }
  ]
},
{ name: 'orderDate', title: 'Order Date', type: 'datetime' },
{ name: 'status', title: 'Status', type: 'string' },
{ name: 'totalAmount', title: 'Total Amount', type: 'number' }
]
};

```

Third Party API

APIs: (Application programming interfaces)

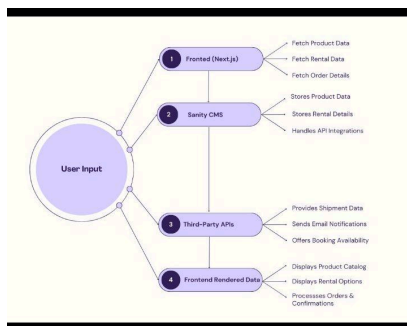
can be integrated into applications to provide access to data and services from other Sources. This can be used to integrate shipment tracking, Payment gateways, and other backend.

API Selection & Integration

- Shippo, EasyPost for Shipment tracking Stripe, paypal for payments.
- follow the API documentation for authentication (Api Keys, OAuth)

fronted-Backend coordination

- APIs return the necessary data for Frontend display (eg, shipment status, payment confirmation)
- Implement proper error handling and data validation.



API Endpoints

User Management

1. `_POST /api/auth/register_`: Register a new user.
2. `_POST /api/auth/login_`: User login.
3. `_GET /api/users/profile_`: Fetch user profile (requires authentication).
4. `_PUT /api/users/update_`: Update user details.

Product Management

1. `_GET /api/products_`: List all products.
2. `_GET /api/products/:id_`: Fetch product details by ID.
3. `_POST /api/products_`: Create a new product.
4. `_PUT /api/products/:id_`: Update product details by ID.
5. `_DELETE /api/products/:id_`: Delete a product by ID.

Order Management

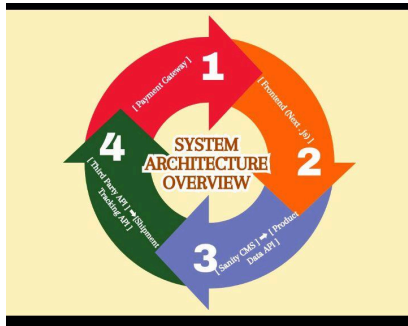
1. `_POST /api/orders_`: Create a new order.
2. `_GET /api/orders_`: List all orders.
3. `_GET /api/orders/:id_`: Fetch order details by ID.
4. `_PUT /api/orders/:id_`: Update order details by ID.
5. `_DELETE /api/orders/:id_`: Delete an Instagram order by ID.

Payment Management

1. `_POST /api/payments_`: Create a new payment.
2. `_GET /api/payments_`: List all payments.
3. `_GET /api/payments/:id_`: Fetch payment details by ID.
4. `_PUT /api/payments/:id_`: Update payment details by ID.
5. `_DELETE /api/payments/:id_`: Delete a payment by ID.

Conclusion

e-commerce website project aims to deliver a dynamic and fully functional platform focused on clothing products. By leveraging Next.js for the frontend and Sanity CMS for the backend, the platform ensures efficient product management, customer data handling, and order record maintenance. The integration of third-party APIs for shipment tracking, payment gateways, and other backend services further enhances the website's functionality, providing a seamless shopping experience.



Prepared by **SUMBUL JAWED**
slot: Sunday 2 to 5
Teacher: sir Ali Jawad.