

Day:2 task Khawaja Naqeeb Uddin Date: 17-01-25

1. Transitioning to Technical Planning

Technical Requirements:

- **Frontend Requirements:**

- Responsive and intuitive UI using Next.js, Typescript and Tailwind CSS.
- Dynamic product catalog with filtering and sorting options.
- User account management (login, registration, profile update).
- Order tracking and history.

- **Backend Requirements:**

- Sanity CMS for managing product and content data.
- Integration with third-party APIs for payment processing and shipment tracking.
- Secure authentication and authorization.

- **Third-Party APIs:**

- Payment gateway API (e.g., Stripe, PayPal).
- Shipment tracking API (e.g., Shippo, EasyPost).
- Geolocation API for delivery zone coverage.

2. Design System Architecture

System Components:

- Frontend (Next.js, Typescript with Tailwind CSS).
- Backend and Database (integrated with Sanity CMS).
- APIs (Third-party services for payment and shipment).

High-Level Diagram:

- User interacts with the frontend.
- Frontend communicates with the backend.
- Backend interacts with:
 - Sanity CMS for product data.
 - Database for orders and customers.
 - Third-party APIs for payment and shipment.

Key Workflows:

- **User Registration:**

1. User submits registration form.
2. Backend validates and stores data.
3. Confirmation email sent to user.

- **Product Browsing:**

1. User selects category or applies filters.
2. Backend fetches product data from Sanity CMS.
3. Results displayed on the frontend.

- **Order Placement:**

1. User adds products to cart and proceeds to checkout.
2. Backend calculates total price and creates an order.
3. Payment processed via third-party API.

- **Shipment Tracking:**

1. User enters order ID to track shipment.
2. Backend fetches tracking data from shipment API.
3. Tracking status displayed to user.

3. Plan API Requirements

Frontend to Backend APIs:

- **User Registration:**

- Method: POST
- Endpoint: /api/register
- Body: {

name,

email,

password,

address,

phone_number }

- **Product Browsing:**

- Method: GET
- Endpoint: /api/products
- Query Params: {

category,

price_range,

sort }

- **Order Placement:**

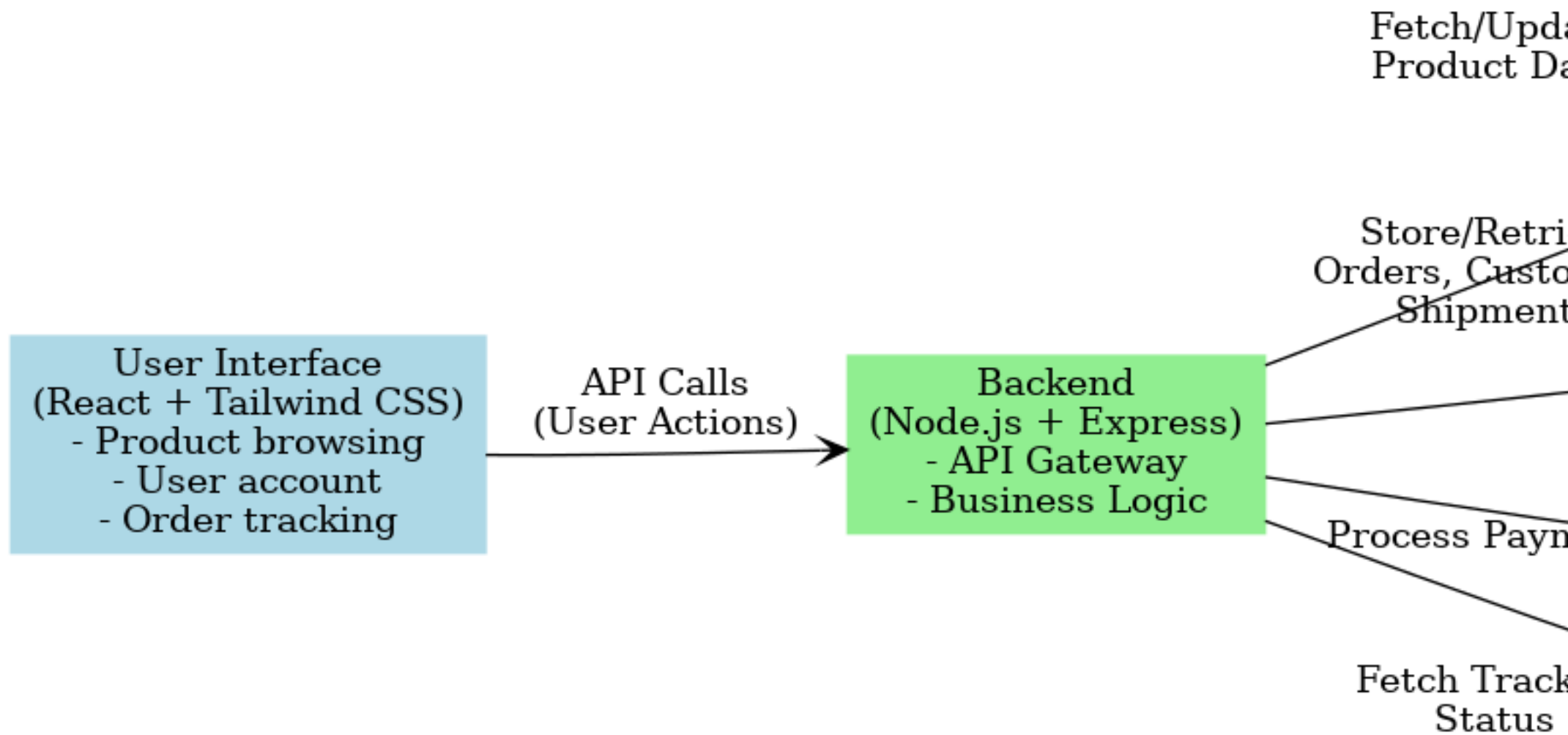
- Method: POST
- Endpoint: /api/orders
- Body: { customer_id,
- product_ids,

- total_price,
- payment_method }
- **Shipment Tracking:**
 - Method: GET
 - Endpoint: /API/track-shipment
 - Query Params: { order_id }

Backend to Third-Party APIs:

- Payment processing API integration.
- Shipment tracking API integration.
- Geolocation API for delivery zones.

Diagram of System Component Interaction



```
/product.ts
```

```
export default {
```

```
  name: 'product',
```

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

```
  title: 'Product',
```

```
  fields: [
```

```
    { name: 'name', type: 'string', title: 'Name', validation: Rule => Rule.required() },
```

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

```
    { name: 'price', type: 'number', title: 'Price', validation: Rule => Rule.required().min(0) },
```

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

```
    { name: 'images', type: 'array', title: 'Images', of: [{ type: 'image' }] },
```

```
{ name: 'stockQuantity', type: 'number', title: 'Stock Quantity', validation: Rule => Rule.min(0) },
```

```
{ name: 'sellerId', type: 'string', title: 'Seller ID' },
```

```
],
```

```
};
```

```
/Order.ts
```

```
export default {
```

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

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

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

```
  fields: [
```

```
    { name: 'customerId', type: 'string', title: 'Customer ID', validation: Rule => Rule.required() },
```



```
{ name: 'productIds', type: 'array', title: 'Products', of: [{ type: 'reference', to: [{ type: 'product' }] } ] },
```

```
{ name: 'totalPrice', type: 'number', title: 'Total Price', validation: Rule => Rule.required().min(0) },
```

```
{ name: 'orderDate', type: 'datetime', title: 'Order Date' },
```

```
{ name: 'status', type: 'string', title: 'Status', options: { list: ['pending', 'confirmed', 'shipped', 'delivered', 'canceled'] } },
```

```
],
```

```
};
```

```
/customer.ts
```

```
export default {
```

```
  name: 'customer',
```

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

```
  title: 'Customer',
```

```
fields: [  
  
  { name: 'name', type: 'string', title: 'Name', validation: Rule => Rule.required() },  
  
  { name: 'email', type: 'string', title: 'Email', validation: Rule => Rule.required().email() },  
  
  { name: 'password', type: 'string', title: 'Password', validation: Rule => Rule.required() },  
  
  { name: 'address', type: 'text', title: 'Address' },  
  
  { name: 'phoneNumber', type: 'string', title: 'Phone Number' },  
  
  { name: 'orderHistory', type: 'array', title: 'Order History', of: [{ type: 'reference', to: [{ type: 'order' }] } ] },  
  
],  
  
};
```

deliveryZone.ts

```
export default {
```

```
name: 'deliveryZone',
```

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

```
title: 'Delivery Zone',
```

```
fields: [
```

```
{ name: 'zoneName', type: 'string', title: 'Zone Name', validation: Rule => Rule.required() },
```

```
{ name: 'coverageArea', type: 'array', title: 'Coverage Area (Postal Codes)', of: [{ type: 'string' }] },
```

```
{ name: 'deliveryFee', type: 'number', title: 'Delivery Fee', validation: Rule => Rule.required().min(0) },
```

```
],
```

```
};
```

```
shipment.ts
```

```
export default {
```

name: 'shipment',

type: 'document',

title: 'Shipment',

fields: [

{ name: 'orderId', type: 'reference', to: [{ type: 'order' }], title: 'Order ID', validation: Rule => Rule.required() },

{ name: 'trackingNumber', type: 'string', title: 'Tracking Number' },

{ name: 'deliveryStatus', type: 'string', title: 'Delivery Status', options: { list: ['in transit', 'out for delivery', 'delivered'] } },

{ name: 'expectedDeliveryDate', type: 'datetime', title: 'Expected Delivery Date' },

],

};