Hackathon 3 Day 2

Planning the Technical foundation (E-commerce Marketplace)

Presented by Rida Fatima

"NIKE ECOMMERCE DYNAMIC WEBSITE"

My e-commerce website focuses on Nike products, including shoes, shorts, and clothing. It features a responsive and dynamic design built with Tailwind CSS, Next.js, and TypeScript. The website integrates Sanity for content management, along with third-party APIs for product data. Additionally, it includes a shipment tracking API and a secure payment gateway to provide a smooth and efficient shopping experience.

SETTING UP THE DEVELOPMENT ENVIRONMENT

1. Install next.js with tailwind CSS
2. Install npm
3. Install react icons
4. Install sanity CMS
5. Install PayPal for Secure payment.
6. Install Shipengin for shipping tracking.
7.Data import from sanity

Start development server: npm run dev

TECHNICAL PLANNING REQUIRMENTS

Frontend Requirments using next.js and taiwind css

- A homepage featuring dynamic product listings, promotions, and easy navigation.
- Product pages with detailed information and options to add items to the cart or wishlist.
- A shopping cart that keeps track of items with a simple, step-by-step checkout process.
- A responsive design that ensures a seamless experience across mobile, tablet, and desktop.
- A fast, smooth, and intuitive browsing experience for users.

TECHNICAL PLANNING REQUIRMENTS

Using Sanity CMS as backend

▪Set up a headless CMS to manage product data, categories, media assets, and user-generated content.

▪ Create custom schemas for products, reviews, categories, and promotions to maintain flexible content management.

▪ Integrate Sanity API with Next.js to fetch and display content dynamically on the frontend.

▪ Allow easy content editing and updates with Sanity's intuitive studio interface.

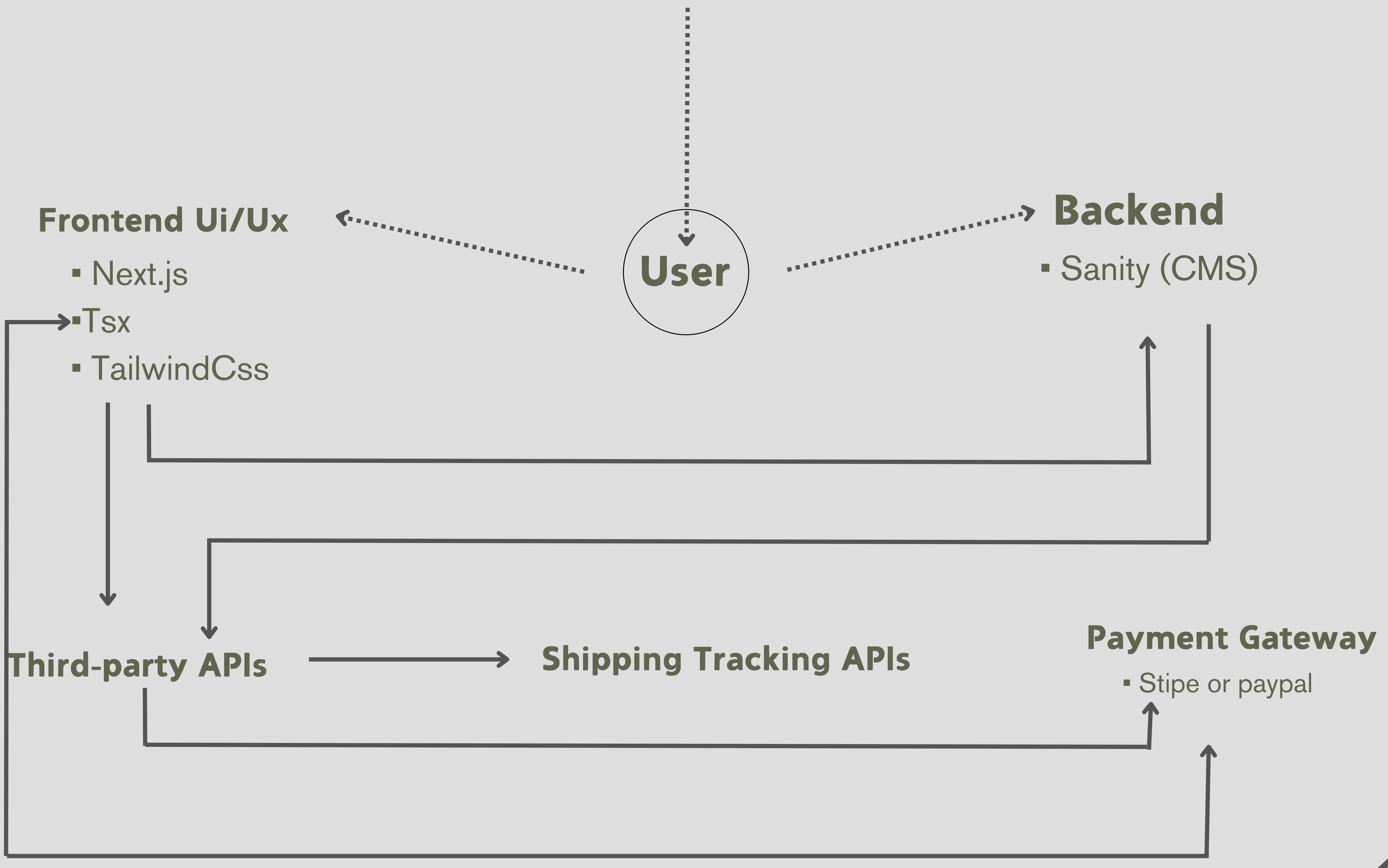
▪ Ensure a structured, scalable backend that can grow with the Marketplace.

TECHNICAL PLANNING REQUIRMENTS

Using Third-Party APIs for E-commerce

- Payment Gateway Integration: Connect to APIs like Stripe or PayPal for secure payment processing.
- Shipping & Logistics: Use APIs like ShipEngine or EasyPost for real-time shipping rates, tracking, and logistics management.
- Authentication: Integrate third-party authentication services (e.g., Google, Facebook) for user login and signup.

SYSTEM'S FLOWCHART



User

Frontend Ui/Ux

- Next.js
- Tsx
- TailwindCss

Backend

- Sanity (CMS)

Third-party APIs

Shipping Tracking APIs

Payment Gateway

- Stipe or paypal

EXPLANATION OF FLOWCHART IN THE SYSTEM

Frontend (Next.js) sends requests to Backend (API) for product data, checkout, etc.
Backend fetches content from Sanity CMS for product details, categories, and inventory.
Backend communicates with Payment Gateway (e.g., Stripe) and Shipping API (e.g., ShipEngine) for payment processing and shipment tracking.
Backend handles transactions, user authentication, and integrates with third-party services like payment and shipping APIs.
This structure shows how data flows between the Frontend, Backend, Sanity CMS, and Third -Party APIs.

API'S ENDPOINTS

• PRODUCT MANEGEMENT

- ▪GET/api/products
- ▪GET/API/products/:id
- ▪POST/api/products (sellers only)
- ▪ PUT/api/products/:id(sellers only)
- ▪ DELETE/api/products/:id (sellers only)

• ORDER MANAGEMENT

GET /api/orders
Fetch all orders
GET /api/orders/:id
Fetch details of a specific order by id
POST /api/orders
Create a new order (customer)
PUT /api/orders/:id
Update the order status (e.g., pending to shipped)
DELETE /api/orders/:id
Cancel or delete an order

API'S ENDPOINTS

• SHIPMENT MANAGEMENT

GET /api/shipments
Fetch all shipments.
GET /api/shipments/:id
Fetch details of a specific shipment by id.
POST /api/shipments
Create a new shipment record.
PUT /api/shipments/:id
Update shipment details (e.g., status, tracking number).
DELETE /api/shipments/:id
Delete or cancel a shipment record.

• PAYMENT MANAGEMENT

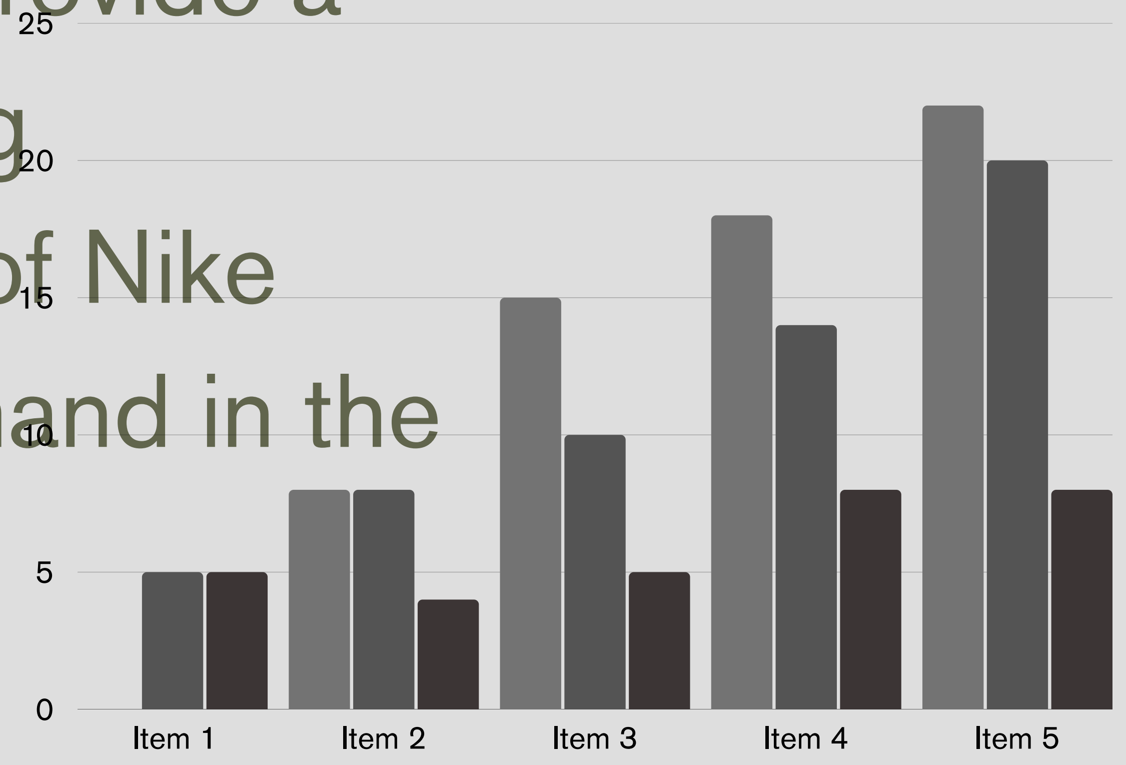
GET /api/payments
Fetch all payment records.
GET /api/payments/:id
Fetch details of a specific payment by id.
POST /api/payments
Process a new payment (e.g., for an order).
PUT /api/payments/:id
Update payment status (e.g., success, failed, refunded).
DELETE /api/payments/:id
Delete a payment record.

MY SCHEMA

```
 productSchema = {  name: 'product',  title: 'Product',  type:
'document',  fields: [    {      name: 'productName',      title:
'Product Name',      type: 'string',    },    {      name: 'category',
title: 'Category',      type: 'string',    },  {    name: 'price',    title:
'Price',    type: 'number',   },   {    name: 'inventory',   title:
'Inventory',   type: 'number'},   {      name: 'colors',      title: 'Colors',
type: 'array',      of: [{ type: 'string' }],    },    {      name: 'status',
title: 'Status',      type: 'string',    },    {      name: 'image',      title:
'Image',      type: 'image', // Using Sanity's image type for image
field      options: {        hotspot: true,      },    },    {      name:
'description',      title: 'Description',      type: 'text',    },  ],}
```

CONCLUSION

With the increasing rate of online shopping, the purpose of this platform is to provide a seamless and convenient shopping experience, offering a wide range of Nike products to meet the growing demand in the digital marketplace.



THANK YOU

Under respected Sir Ameen Alam

By Rida Fatima