Marketplace Technical Foundation

GameKicks

1. System Architecture Document

Overview

This document outlines the system architecture for the sports shoes website, detailing the interaction between the frontend, Sanity CMS, and third-party APIs.

Components

1. Frontend:

Developed using modern web technologies (Next.js with Tailwind CSS).

Responsible for rendering the user interface and handling client-side interactions.

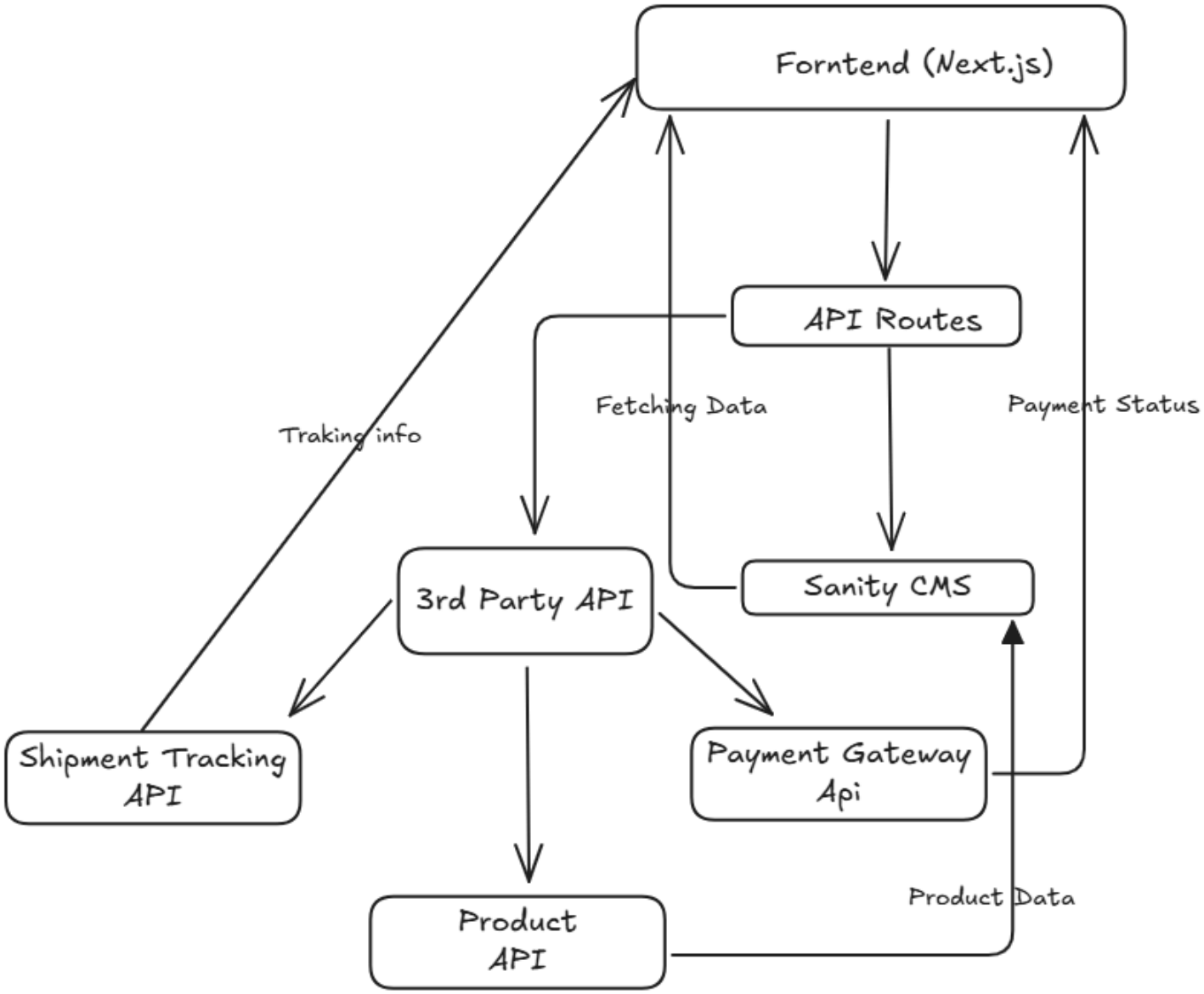
2. Sanity CMS:

Acts as the backend content management system for managing dynamic content like product catalogs, promotions, and customer reviews.

3. Third-Party APIs:

Integrates external services for specific functionalities like payment processing, order tracking, live chat, and analytics.

System Architecture Digram



Roles of Components

1. Frontend:

Handles user input and displays data fetched from Sanity CMS or third-party APIs.

Provides a seamless and responsive user experience.

2. Sanity CMS:

Stores and manages structured content for the website.

Serves as the single source of truth for all dynamic data.

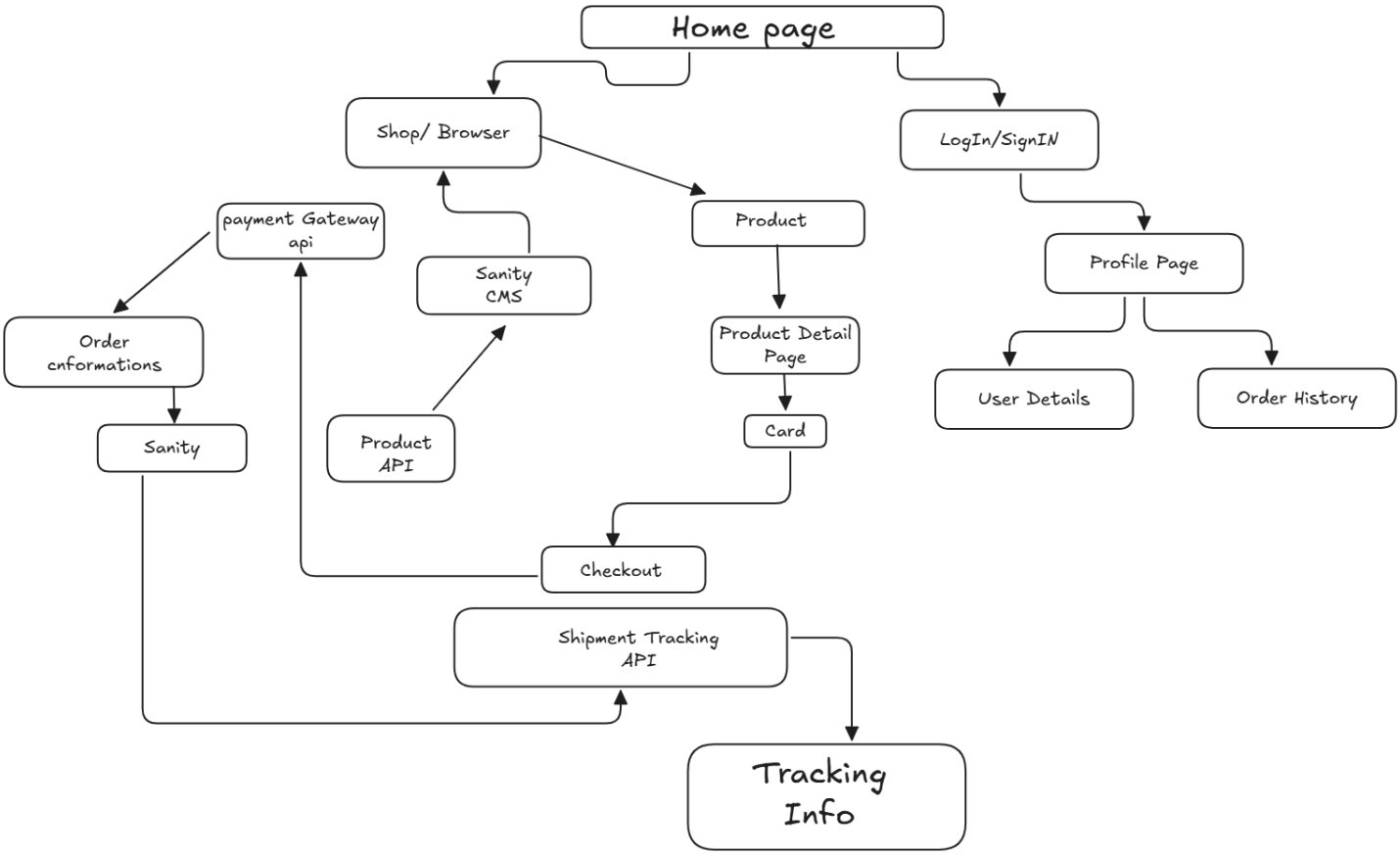
3. Third-Party APIs:

Payment APIs (e.g., Stripe): Process client payments.

Communication APIs (e.g., Twilio): Enable real-time communication.

Analytics APIs (e.g., Google Analytics): Monitor website performance and user behavior.

3.WorkFlow Digram



Technical Roadmap

1: Frontend Development

Tech Stack: React, Next.js, or Vue.js for a modern and responsive UI.

Tasks:

Set up the homepage with navigation to:

Shop/Browse section.

Filter functionality.

Login/Sign-Up.

Design reusable components:

Header, Footer, Product Cards, Forms.

Create routes for:

Product Detail Page.

Profile Page.

Cart and Checkout.

2: Backend Development

Tech Stack: Node.js with Express.js or Django/Flask.

Tasks:

Develop APIs for:

User Authentication (JWT-based secure login/sign-up).

Product Data Fetching (linked with Sanity CMS).

Order Management (Create, Read, Update, Delete operations).

Integrate Sanity CMS:

Sync product data to manage structured content efficiently.

3: Database and Content Management

Tech Stack: MongoDB, PostgreSQL, or Firebase (depending on scalability needs).

Tasks:

Design database schema:

Products, Users, Orders, Payments.

Integrate Sanity CMS for managing:

Product catalogs.

Dynamic website content.

4: Payment Gateway Integration

Tech Stack: Stripe API or PayPal SDK.

Tasks:

Securely integrate payment APIs for handling:

Card payments.

Transaction receipts.

Ensure compliance with PCI DSS standards for payment security.

5: Shipment Tracking System

Tech Stack: Shipment Tracking APIs (e.g., AfterShip or ShipEngine).

Tasks:

Develop an API for tracking orders.

Display real-time tracking information on the “Tracking Info” page.

6: Authentication and User Management

Tech Stack: Auth0, Firebase Auth, or Custom JWT-based authentication.

Tasks:

Implement secure login and registration.

Build a Profile Page for users to:

View and update details.

Access order history.

7: Analytics and Monitoring

Tech Stack: Google Analytics or Mixpanel.

Tasks:

Integrate tracking to monitor:

User behavior.

Conversion rates.

Traffic sources.

8: Deployment and Scaling

Tech Stack: AWS, Vercel, or Netlify.

Tasks:

Deploy the frontend and backend to a cloud environment.

Use CI/CD pipelines for automated deployments.

Set up autoscaling and load balancing to handle high traffic.

9: Testing and Optimization

Tech Stack: Jest, Cypress, or Postman for testing.

Tasks:

Conduct unit testing for all components.

Perform end-to-end testing for workflows (e.g., product purchase and tracking).

Optimize frontend and backend for speed and scalability.

(NIDA TARIQ)