MZS CodeWorks

Gulshan-e-Iqbal
Karachi, 65330
(000) 000 - 0000

January 16, 2025

Technical Document for SportRentHub

Overview

To create a user-friendly, efficient, and reliable online platform for renting sports equipment and gear, catering to adventure enthusiasts, casual players, and professional athletes alike..

Frontend and Backend Requirements

Frontend:

- **Framework:** Next.js
- **Styling:** Tailwind CSS / CSS-in-JS (Styled Components)
- **State Management:** React Context / Redux
- **Data Fetching:** SWR / React Query
- **Authentication:** NextAuth.js

Backend:

- **Framework:** Next.js
- **Database:** Sanity Studio / MongoDB (with Mongoose for schema)
- **CMS:** Sanity.io

- **Authentication:** JWT (JSON Web Tokens)
- **Hosting:** Vercel

Third-Party APIs:

- **Payment Gateway:** TBD
- **Shipment Tracking:** EasyPost or Shippo API
- **Email Service:** MailJet or SendGrid API

Design System Architecture

High-Level Diagram:

Key Workflows to Include

1. User Registration:

- **User signs up:** User submits registration form.
- **Data is stored in Sanity:** API call to store user data in Sanity.io.
- **Confirmation sent to the user:** Email confirmation via SendGrid/Mailgun.

2. Product Browsing:

- **User views product categories:** Frontend fetches categories.
- **Sanity API fetches data:** Sanity API returns product data.
- **Products displayed on frontend:** Rendered on the client-side.

3. Order Placement:

- **User adds items to the cart:** Items added to cart state.
- **Proceeds to checkout:** User enters shipping and payment info.
- **Order details saved in Sanity:** API call to store order details in Sanity.io.

4. Shipment Tracking:

- **Order status updates fetched via 3rd-party API:** EasyPost/Shippo API fetches tracking info.
- **Displayed to the user:** Order status and shipment tracking shown in user account.

Navigation Structure

This navigation structure provides a comprehensive and user-friendly way for visitors to explore the SportRentHub website. Each section is designed to guide users through their journey, from browsing equipment to renting and tracking their orders.

1. **Home**
    - Welcome Message
    - Featured Equipment
    - Special Offers
    - Customer Reviews
2. **Categories**
    - Skiing Gear
    - Snowboarding Gear
    - Camping Gear
    - Hiking Gear
    - Bicycles
    - Kayaks
    - Tennis Gear
    - Golf Sets
3. **Browse Equipment**
    - Search Bar
    - Filters (Category, Brand, Condition, Price Range)
    - Equipment Listings
4. **Rental Process**
    - How It Works
    - Pricing Information

- Rental Terms and Conditions

5. **My Account**
    - Profile Information
    - Order History
    - Saved Items
    - Verification Documents
6. **Cart**
    - View Cart
    - Edit Cart
    - Proceed to Checkout
7. **Checkout**
    - Shipping Information
    - Payment Information
    - Order Summary
    - Place Order
8. **Shipment Tracking**
    - Track Order Status
    - Shipment Updates
9. **Returns**
    - Return Policy
    - Initiate Return
    - Track Return Status
10. **Reviews**
    - Write a Review
    - View All Reviews
11. **Help Center**
    - FAQs
    - Contact Us
    - Support
12. **About Us**
    - Our Story
    - Meet the Team
    - Careers

13. **Blog**
    - Articles
    - Tips & Guides
    - News & Updates
14. **Legal**
    - Privacy Policy
    - Terms of Service

System Architecture Document

Describes the Overall Design and Interaction between Components:

The **SportRentHub** system is structured in a modular way to separate concerns and enhance scalability:

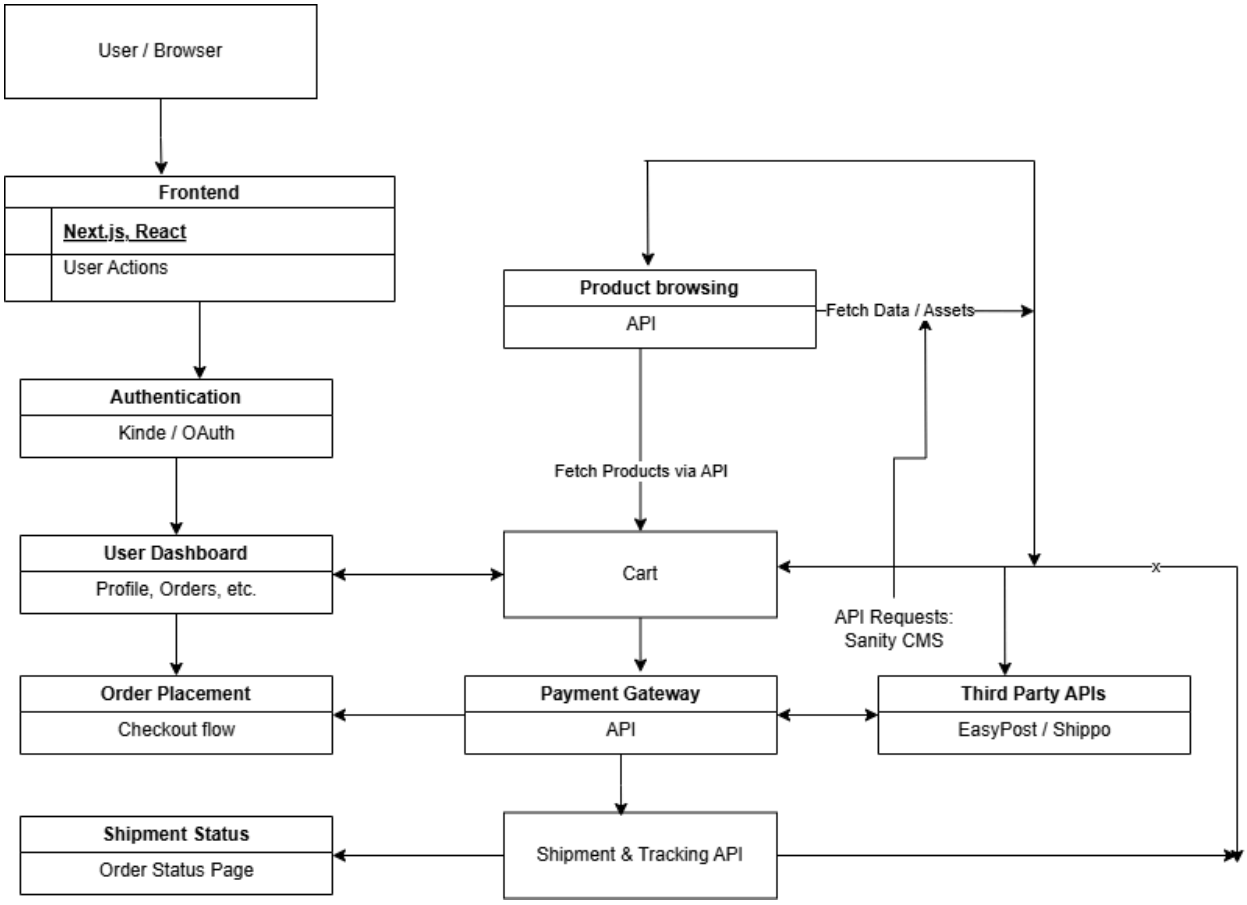
- **Client Side (Frontend):** Built using Next.js and styled with Tailwind CSS. Fetches data from the Sanity API and interacts with the backend via API routes.
- **API Server (Backend):** Handles business logic and communicates with the database (MongoDB). Also, integrates with third-party APIs for payment, email, and shipment tracking.
- **Sanity or Database (MongoDB):** Stores user information, product listings, orders, and more.
- **Sanity.io (CMS):** Manages content for products, categories, and other static data.

API Specification Document

Endpoints, Methods, Payloads, and Responses:

1. **User Registration:**
    - **Endpoint:** `/api/register`
    - **Method:** POST
    - **Payload:** `{ name, email, password, phoneNumber, address, userType }`
    - **Response:** `{ success: true, message: "User registered successfully" }`
2. **Product Browsing:**
    - **Endpoint:** `/api/products`
    - **Method:** GET
    - **Response:** `{ products: [...] }`
3. **Order Placement:**
    - **Endpoint:** `/api/orders`
    - **Method:** POST
    - **Payload:** `{ userId, cartItems, totalPrice, shippingAddress, paymentMethod }`
    - **Response:** `{ success: true, message: "Order placed successfully" }`
4. **Shipment Tracking:**
    - **Endpoint:** `/api/track`
    - **Method:** GET
    - **Params:** `trackingNumber`
    - **Response:** `{ status: "In Transit", expectedDelivery: "2025-01-20" }`

Workflow Diagram



Visualizes User Interactions and Data Flows:

1. User Registration:

   User -> API Server (/api/register) -> Sanity.io -> SendGrid

2. Product Browsing:

   User -> Sanity API (/api/products) -> Frontend

3. Order Placement:

   User -> API Server (/api/orders) -> Sanity.io -> Stripe/PayPal

4. Shipment Tracking:

User -> API Server (/api/track) -> EasyPost/Shippo

Data Schema Design

Defines Entities and Relationships for Databases or CMS:

- **User:** Stores user details including contact information and verification status.
- **Admin:** Manages platform functionalities.
- **Equipment:** Details about sports equipment, including stock and condition.
- **Rental:** Information about rentals, including start and end dates.
- **Payment:** Records of transactions.
- **Review:** User feedback for equipment.
- **Category:** Classification of equipment.
- **Penalty:** Records of penalties for delayed returns.
- **Shipment:** Details about shipments.
- **Delivery:** Status of equipment delivery.
- **Return:** Management of returned equipment.

Technical Roadmap

Outlines the Steps to Complete the Project, Milestones, and Deliverables:

1. **Initial Planning & Research:**
    - Create technical plans.
    - Define project scope.
2. **Design & Prototyping:**
    - Create wireframes and prototypes.
    - Finalize UI/UX design.
3. **Development:**
    - Set up the backend and database.

- Develop frontend components.
- Integrate Sanity.io and third-party APIs.

4. **Testing & Quality Assurance:**
    - Perform unit and integration testing.
    - Fix bugs and optimize performance.
5. **Launch & Marketing:**
    - Deploy the website.
    - Gather user feedback for continuous improvement.