**"Marketplace Technical Foundation - TokenRent"**

**Technical Documentation**

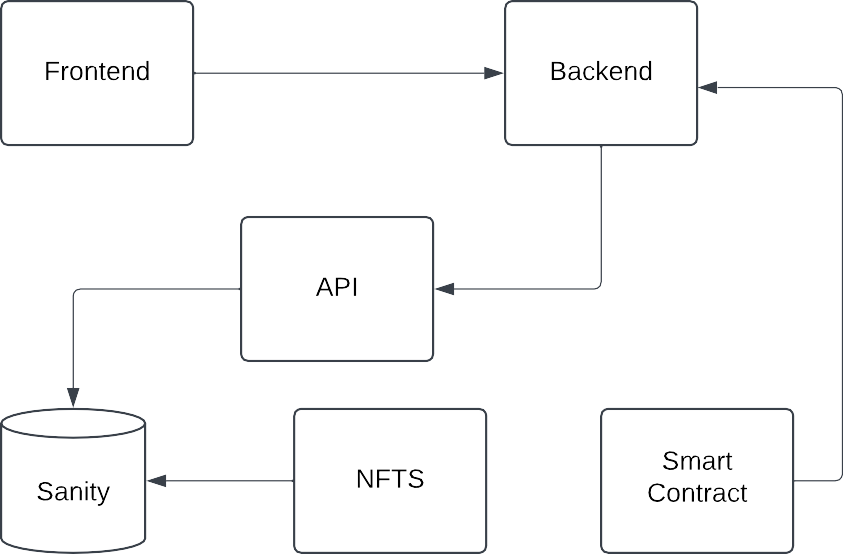
This document outlines the technical foundation for the Web3 Rental Marketplace **“TokenRent”**, focusing on system architecture, workflows, API endpoints, and project milestones. The aim is to provide a comprehensive blueprint for implementation.

**System Overview**

TokenRent integrates **blockchain technology**, **Sanity CMS**, and **third-party APIs** with a responsive frontend to deliver secure, transparent, and efficient rental services. Key components include:

* **Frontend**: Built with Next.js for dynamic user interaction. I will also Try to integrate 3D.
* **Blockchain**: Manages NFTs and smart contracts for rental agreements and escrow for security deposts.
* **Backend**: Powered by Sanity CMS to handle metadata and content management.
* **Third-Party APIs**: For payment processing, shipment tracking, and geolocation.

**System Architecture**



**Detail Work Flow Diagram**



**Core Features and Workflows**

**1. User Authentication**

* Users log in with Web3 wallets like **MetaMask** etcto access the platform securely.
* Wallet addresses are linked to user profiles stored in Sanity CMS.

**2. Product Management**

* Product data (name, description, price, NFT ID) is fetched from Sanity CMS.
* NFTs are linked to products to ensure ownership transparency.

**3. Rental Transactions**

* Smart contracts handle rental agreements, holding security deposits in escrow.
* Rental status updates are reflected on the blockchain and frontend.

**4. Payment and Shipment Integration**

* Payments are processed via a crypto gateway, updating the rental smart contract.
* Shipment tracking APIs provide real-time delivery updates for physical products.

**API Endpoints**

**1. Key API Categories**

We will need APIs for the following purposes:

1. **Product Management:** Handle product data like details, pricing, and availability.
2. **User Management:** Manage user profiles and authentication via Web3 wallets.
3. **Rental Transactions:** Process rental agreements, payments, and security deposits.
4. **NFT Integration:** Fetch data about product ownership and rental status from the blockchain.
5. **Shipment and Location Services:** Track deliveries and provide location-based services.

**User Authentication**

* **Endpoint**: /users
* **Method**: POST
* **Description**: Authenticate users via Web3 wallets.
* **Request Example**:
* {
* "walletAddress": "0x123abc",
* "name": "John Doe",
* "email": "john.doe@example.com"
* }
* **Response Example**:
* {
* "userId": 1,
* "status": "success",
* "message": "User authenticated successfully."
* }

**Product Management**

* **Endpoint**: /products
* **Method**: GET
* **Description**: Fetch a list of available products.
* **Response Example**:
* [
* {
* "id": 1,
* "name": "4K Drone",
* "price": 50,
* "availability": "Available",
* “image”: camera.png,
* "NFT\_ID": "0xabc123"
* }
* ]

**Rental Management**

* **Endpoint**: /rentals
* **Method**: POST
* **Description**: Create a new rental agreement/transaction.
* **Request Example**:
* {
* "productId": 1,
* "userId": 2,
* "rentalDuration": "3 days",
* "securityDeposite": 100,
* “paymentAmount”: 40,
* }
* **Response Example**:
* {
* "transactionId": 789,
* "status": "success",
* "message": "Rental created successfully."
* }

**NFT Metadata**

* **Endpoint**: /nfts/:id
* **Method**: GET
* **Description**: Fetch details of a specific rental transaction, including blockchain status.
* **Response Example**:
* {
* "NFT\_ID": "0xabc123",
* "productId": 1,
* "owner": "0xuser456",
* "rentalStatus": "In Use",
* "rentalHistory": [
* {
* "transactionId": 789,
* "rentedBy": "0xuser789",
* "duration": "3 days"
* }
* ]
* }

**Shipment and Location Services**

* **Endpoint**: /shipment
  + **Method**: POST
  + **Description**: Create a shipment order for a physical product.
  + **Request Example**

{

"orderId": 789,

"address": "123 Main St, City, Country",

"expectedDeliveryDate": "2025-01-18"

}

**Response Example**:

{

"shipmentId": "SHIP123",

"status": "Created",

"trackingLink": "https://shipment.example.com/track/SHIP123"

}

**Endpoint**: /geolocation

* **Method**: GET
* **Description**: Retrieve nearby available products based on user location.
* **Request Example**:

{

"latitude": 40.7128,

"longitude": -74.0060

}

**Response Example**:

[

{

"productId": 1,

"name": "4K Drone",

"distance": "2.5 miles"

},

{

"productId": 3,

"name": "Event Projector",

"distance": "5 miles"

}

]

**Milestones**

**Milestone 1: Business Planning Day**

* Define business goals, target audience, and core workflows.
* What we Develop.
* **Duration**: 1 Day

**Milestone 2: Technical Day Thinking**

* Define how the business work, there features, components ,api and how they interact with each other.
* How Smart Contract handle the rental agreement.
* How NFTS saved the digital proof of ownership of the product.
* How Sanity stores the product metadata with NFTS integration.
* How api’s look like (I created the Examples)
* **Duration**: 1 Day

**Milestone 3: Backend Development**

* Implement Sanity CMS for content management.
* Develop NFT and rental agreement smart contracts using the solidity language.
* Test the smart contracts.
* Deployments of Smart Contracts So it will use in frontend.
* **Duration**: 2 or 3 Days

**Milestone 4: Frontend Development**

* Build a responsive UI using Next.js with the integration of 3D. Also this UI is New.
* Integrate APIs and blockchain interactions.
* **Duration**: 2 or 3 Days

**Milestone 5: Testing and Debugging**

* Conduct end-to-end testing to ensure functionality.
* Resolve bugs and optimize performance.
* **Duration**: 1 Days

**Milestone 6: Deployment**

* Deploy the Frontend nextjs in Vercel.
* Perform final checks and launch.
* **Duration**: 1 Days