華南理工大學
South China University of Technology

《软件应用系统设计》 课 程 论 文

(2024-2025 学年第 1 学期)

区块链驱动的 3D 模型 NFT 交易平台

提交日期： 2025 年 1 月 1 日

学 号	姓名	专业班级	学生签名
202130670058	黄泽君	产品设计 2 班	

课程名称	软件应用系统设计	任课教师	王亮明

教师评语：

本论文成绩评定： _____分

题目 区块链驱动的 3D 模型 NFT 交易平台

学生姓名 黄泽君

摘要：论文阐述了一个 3D 模型 NFT 交易平台的构建过程，涵盖了从设计到实现的各个方面。该平台利用 React、Next.js 和 Tailwind CSS 构建了现代化的用户界面，结合以太坊区块链和智能合约实现了后端功能。通过使用 React Context API 进行全局状态管理，平台能够高效地处理用户信息和交易数据。实用函数的引入简化了与智能合约的交互，包括合约实例化、交易记录的管理和地址格式化等功能。

平台支持 3D 模型的 NFT 铸造、发布、购买、拍卖和转售功能，同时还增加了捐款、反馈、转账、社区互动等扩展功能，提供了更加全面的用户体验。经过本地测试确认所有功能正常后，项目在 Vercel 平台完成了部署。此平台展示了其在安全性、透明度和用户体验方面的优势，为 3D 模型的 NFT 交易提供了一个创新的解决方案。

关键词：非可同质化代币；3D 模型；以太坊区块链；智能合约；React 前端开发；全局状态管理

目　录

第一章 项目简介及小组分工说明

1.1 项目愿景

在国家政策的引导、技术进步的驱动以及社会需求的推动下，区块链产品的设计迎来了重要的发展机遇。本课题试图解决传统知识产权保护中小型设计难以高效获得保障的问题。专利申请流程复杂且耗时，尤其对于产品外观、界面创新或功能优化等层级较低的小设计，传统保护方式显得力不从心。通过引入区块链的非同质化代币（NFT），本研究提出一种创新的设计保护模式。借助 NFT 的去中心化、不可篡改特性，将每个小设计生成独一无二的数字化凭证并记录在链上，既能高效证明设计归属权，又为设计成果的交易与收益分配提供了新的路径。

研究期望实现以下目标：

1. 借助区块链技术，为小型设计提供一种高效、低成本的保护方式，弥补传统专利保护的不足。

2. 建立基于 NFT 的设计保护和应用标准化流程，帮助设计行业实现更便捷的创新成果认证和权益保障。

3. 探索设计保护与商业化结合的可能性，通过智能合约等技术手段，激励更多创新并促进市场化运作。

本课题不仅聚焦于优化用户体验，还为区块链技术的跨领域应用提供了实践支持。在研究过程中，通过构建科学的设计方法论并引入 NFT 技术，旨在为区块链产品的开发与知识产权保护探索新的技术路径与应用模式。

1.2 小组人员组成

本小组由个人独立完成项目，涵盖了多角色、多任务的职责需求。组员主要负责以下工作内容：

1. 项目总体规划。进行课题方向的确定、项目目标的制定以及各阶段工作计划的编排，为项目提供清晰的发展路径。

2. 设计与开发。运用双学科背景（产品设计与计算机科学技术），结合理论研究与实际需求，完成区块链产品的用户界面与交互设计，同时承担系统功能开发与实现。

3. 研究与分析。深入分析区块链技术在知识产权保护领域的可行性，研究用户需求，探索 NFT 技术的设计应用方法论。

4. 文档撰写与总结。独立完成项目设计文档的编写，包括项目愿景、技术分析、设计流程、实施细节以及最终总结。

通过个人的努力完成上述多项职责，本小组实现了从理论研究到产品设计再到成果总结的完整闭环。

第二章 需求文档

2.1 外部环境

评估外部环境中政治、经济、社会和技术因素对项目的影响：

从政治环境上说，区块链技术在中国得到了政策层面的高度关注和支持。2019 年 10 月 24 日，中共中央政治局集体学习区块链技术，习近平总书记强调区块链是核心技术创新的重要突破口，要求明确方向、加大投入，加快推动技术与产业发展。2020 年 4 月 20 日，国家发改委将区块链列为“新基建”七大领域之一，为技术发展提供政策支持和市场空间。此外，国务院发布的《十四五规划纲要》提出推动区块链与实体经济深度融合，加强基础研究和标准化建设，明确了未来的发展方向。地方政策也在逐步跟进，例如海南省在 2022 年出台的《区块链技术与产业发展条例》，为区块链技术发展提供了法律保障。

从经济环境上说，数字经济的迅猛增长、共享经济的普及以及创作者经济的崛起，为区块链应用场景提供了丰富的土壤。数字经济的持续增长为 3D 模型 NFT 市场提供了广阔的潜力。共享经济的理念在数字资产领域逐步渗透，越来越多的用户倾向于通过共享资源来降低成本和提高效率。随着创作者经济的迅速发展，越来越多的个人和小团队通过数字平台展示和变现自己的创意内容。

就社会环境而言，随着数字文化的普及和 NFT 艺术品的传播，普通用户对数字资产的接受度逐渐提高元宇宙的兴起进一步推动了虚拟资产需求的增长，3D 模型 NFT 因其个性化特点成为重要的数字资产类型。加密货币的普及提高了用户对去中心化交易和数字资产的认知与接受度，为区块链产品的市场推广提供了有力支持。同时，共享与合作文化在年轻人中流行，为产品的发展创造了良好的社会环境。

从技术环境上说，区块链技术的发展经历了 Web 1.0 的只读阶段、Web 2.0 的交互阶段，最终迈入了 Web 3.0 的去中心化阶段。在 Web 3.0 架构中，数据和身份信息由用户通过数字钱包自主管理，取代了传统的中心化平台控制模式。基于智能合约的自动化执行机制进一步提升了交易的透明性和安全性。同时，区块链技术与虚拟资产的结合，极大地拓展了数字资产的交易模式和应用场景。

2.2 用户需求

在设计 3D 模型 NFT 平台时，深入了解目标用户的特征和需求是确保平台功能设计精准化的关键。本节通过用户画像与需求分析，明确目标用户群体的核心期待与功能需求，为平台的建设提供指导。以下是平台的目标人群：

1. 3D 打印爱好者。这一群体多为技术爱好者，对 3D 打印技术有着浓厚兴趣，熟悉或拥有 3D 打印设备，热衷于探索新设计与模型资源。他们期待获取高质量、可直接打印的模型资源，学习打印技巧，并通过社区交流提升技术水平。同时，为降低打印成本，他们也希望能够找到经济实惠的模型或共享打印服务。为满足这类用户需求，平台应提供经过优化的 3D 模型 NFT，支持用户直接下载并打印；增设打印交流社区，促进用户间的经验分享；推出共享 3D 打印机功能，提升设备利用率并降低使用成本。

2. 开源分享爱好者。开源分享爱好者注重共享精神，愿意以免费或低成本方式分享自己的设计作品，同时通过他人设计获得创意启发。他们希望平台能够公平分发和传播作品，同时能通过社区的认可和经济回报获得激励。为满足这一群体的需求，平台需提供支持开源的 NFT 功能，允许用户设定授权规则；引入代币经济奖励机制，鼓励用户分享设计并促进社区合作。

3. 喜欢动手改装的人群。动手改装爱好者倾向于将设计模型进一步改装，或将其应用于实际生活场景中。他们更关注可以自定义或改装的模型资源，同时希望与志同道合者交流改装心得，甚至开展合作项目。此外，他们也对模型的实际应用案例与改装建议表现出浓厚兴趣。针对这类需求，平台应提供高可改装性的模型 NFT，并附带详细的设计文档与改装说明。

4. 个人 Studio 的设计师。这一用户群体由独立设计师或小型团队组成，他们需要一个高效的设计与资源平台来支持创意输出和商业化。他们关注能够展示并销售设计的可靠渠道，灵活的版权保护与授权机制，以及与其他设计师合作的机会。为满足这些需求，平台需提供 NFT 交易功能，帮助设计师将作品轻松变现；支持定制版权协议，保障原创设计的同时简化商业授权流程；增设设计师合作专区，促进跨领域合作与团队组建。

平台的总体战略是通过共享 3D 打印机和资源整合功能，促进用户间的资源流动与价值交换。平台不仅作为 3D 模型的交易市场，还应构建一个充满活力的社区生态，满足不同用户群体的核心需求，推动设计与区块链技术的深度融合。

2.3 产品逻辑

NFT 的生命周期设计围绕用户的上传、出售、拍卖及购买操作展开，旨在确保交易流程清晰高效，为用户提供良好的交互体验。用户在平台上上传 NFT 后，可为其设置以太坊价格和原生代币价格，并将其放入市场进行出售。在出售过程中，卖家可根据需求选择是否启用拍卖功能。若未启用拍卖，平台将提供“设置拍卖”按钮，便于卖家后续操作。当卖家选择启用拍卖功能后，该 NFT 将被转移至拍卖专区展示，并接受用户的竞标。在拍卖过程中，卖家可以实时查看各买家的出价列表，便于掌握竞标进度。当拍卖结束时，卖家将看到“结束拍卖”按钮，通过点击该按钮完成交易流程。此时，NFT 会自动转移至最高出价者的账户，且资金会结算至卖家账户。

买家可通过直接购买或参与拍卖两种方式获取 NFT。对于拍卖中的 NFT，买家可以在拍卖未结束时参与竞标。若买家的出价为当前最高价，系统将提示其为最高出价者，同时限制其再次出价，以保证出价公平性。当拍卖结束后，出价最高的买家将看到“结束拍卖”按钮，点击按钮完成交易，并将 NFT 转移至其个人账户。对于无需参与拍卖的 NFT，买家可直接支付卖家设置的价格完成购买。购买成功后，NFT 将出现在买家的“我的 NFT”页面中。

此外，买家在获取 NFT 后，若希望将其再次出售，可通过“转售”功能重新设置价格，并将 NFT 放回市场。这一功能设计为用户提供了灵活的资产流通机制，同时促进了平台交易的活跃性和多样性。通过完善的上传、出售、购买及再出售流程设计，平台为用户构建了一个高效、透明的交易系统，有助于进一步推动 NFT 市场的发展与繁荣。

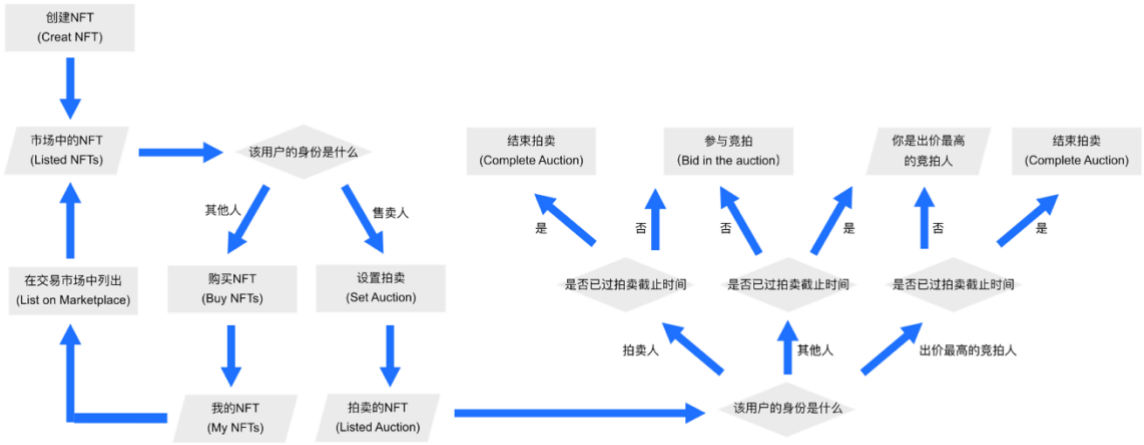

图 2-1 NFT 的生命周期

第三章 详细设计说明书

3.1 项目设置

在命令行输入“npx create-next-app”快速创建一个新的 Next.js 应用程序，删除不必要的文件并清空 CSS 样式得到简化和清理接下来，我们将探讨 Next.js 项目的实现。在 package.json 文件中，我们需要指定项目所需的各种依赖包。运行命令安装必要的库文件，然后在源代码目录下生成 .eslintrc.js、hardhat.config.js、postcss.config.js、tailwind.config.js 自定义项目的开发环境,以及编写 comp 元素、环境、代码、工具、文件的分别放置项目的各种 React 组件、Context 对象、脚本文件、实用函数和一些静态资源。

ERC20 是以太坊网络上用于定义可互换代币的一个标准[19]。这一标准由以太坊社区定义，在各种应用和服务之间实现不同代币的互操作。ERC20 标准确保代币合约遵循接口的一致性，使其具有广泛的接受度和支撑性。ERC20 标准定义了所有达到标准的代币都必须实现的一组函数和事件。以下是主要的函数和事件：

1. 总供应量 (totalSupply)：返回代币总量。
2. 余额查询 (balanceOf)：查询该地址持有的代币数量。
3. 转账 (transfer)：将一个地址的代币转移到另一个地址。
4. 授权 (approve 和 allowance)：允许第三方从你的账户中提取代币。
5. 事件 (Transfer 和 Approval)：在代币转移或授权时发出事件用于外部监听。

接下来分别介绍 3D 模型 NFT 交易平台的智能合约、上下文管理及前端部分。在实际开发过程中，这三个模块通常由三个不同的团队负责。

图 3-1 平台架构

3.2 智能合约

NFT 市场合约是最主要的职能合约，继承了 OpenZeppelin 的 ERC721URIstorage 和 ReentrancyGuard。该协议通过 ERC721 标准实现扩展，利用 Counters 库对 TokenID 和商品的销售数量进行跟踪。在部署合约时，合约的所有者被设置为合约的初始所有者。合约实现以下内容：

1. 铸造和上市 NFT：用户可以通过指定令牌 URI 和价格铸造新的 NFT 并将其列在市场上。
2. 购买和转售 NFT：已经上市的 NFT，用户可以购买并在市场上转卖。
3. 拍卖 NFT：已经列出 NFT，用户可以设置为拍卖，允许其他用户参与竞拍。在拍卖结束时，出价最高的竞拍者胜出。
4. 管理投标：安全地处理投标的放置，以及超过截止时间后完成拍卖。

安全问题是 NFT 市场合约设计中的重要考量因素。合约使用了 ReentrancyGuard 来防止资金转移操作中常见的安全漏洞——重入攻击。通过在关键函数中加入 OnlyOwner 修饰符，确保只有合约所有者才能调用该函数。此外，合约在转移资金时使用了 require 语句和检查目标地址，以确保资金的安全转移。

合约的主要模块有数据结构、事件和函数。在合约中，主要的数据结构包括 MarketItem 和 userBidding，具体如表 3-1 所示。这些数据结构通过映射存储在合约中，以便快速查找和更新相关信息。

表 3-1 NFT 市场合约的数据结构

MarketItem	数据结构存储 NFT 的详细信息，包括 NFT 的 ID、卖家地址、所有者地址、价格、拍卖价格、是否在售、拍卖状态、起止时间等。
userBidding	数据结构存储用户的竞拍信息，包含 NFT 的 ID、竞拍 ID、竞拍金额、竞拍人地址等。这些数据结构通过映射存储在合约中，以便高效地检索和更新相关资料。

合约定义了一系列事件，用于记录和触发合约中的重要操作。事件信息记录在区块链的日志（logs）中，这些日志不直接影响合约的状态，查询时效率更高。事件不仅独对开发者透明，也为用户和审计者提供了可靠的交易记录和状态追踪手段。如果你将数

据存储在合约的状态变量中，比如数组或映射，每次更新数据时都需要支付较高的 Gas 费用。合约定义的事件如表 3-2 所示，这些事件帮助客户和开发人员跟踪合约操作，并提供透明度。

表 3-2 NFT 市场合约的事件

MarketItemCreated	事件在创建市场项目时触发，记录 NFT 的 ID、卖家、所有者、价格及售出状态。
AuctionCreated	事件在创建拍卖时触发，记录卖家、价格、NFT 的 ID、起止时间。
BidCreated	事件在竞拍时触发，记录竞拍 ID、竞拍人地址、竞拍金额。
AuctionCompleted	事件在拍卖完成时触发，记录竞拍 ID、卖家、竞拍人、竞拍金额。
WithdrawBid	事件在竞拍人撤回竞拍时触发，记录竞拍 ID、竞拍人地址、竞拍金额。

合约构建了一系列函数如表 3-3 所示：

表 3-3 NFT 市场合约的函数

updateListingPrice	函数允许合约所有者更新挂牌价格。函数接受一个新的挂牌价格作为参数，并更新 listingPrice 变量。
getListingPrice	函数返回当前的挂牌价格。
createMarketItem	内部函数用于创建新的市场项目，并将其存储在 idToMarketItem 映射中。函数检查价格是否大于零，并验证支付的金额是否等于挂牌价格。
createToken	函数允许用户铸造新的 NFT，并将其添加到市场中。函数首先调用 _mint 和_setTokenURI 创建新的 NFT，然后调用 createMarketItem 将其添加到市场项目中。
resellToken	函数允许 NFT 所有者重新挂牌出售 NFT。函数更新市场项目的价格和所有者信息，并将 NFT 转移回市场合约。
createMarketSale	函数允许用户购买市场中的 NFT。函数验证支付金额是否等于 NFT 的价格，并将所有者信息更新为买家。
fetchMarketItems	函数返回所有未售出的市场项目。函数遍历 idToMarketItem 映射，

	筛选出所有未售出的项目。
fetchMyNFTs	函数返回用户购买的所有 NFT。函数遍历 idToMarketItem 映射，筛选出所有所有者为当前用户的项目。
fetchItemsListed	函数返回用户列出的所有 NFT。函数遍历 idToMarketItem 映射，筛选出所有卖家为当前用户的项目。
getContractBalance	函数返回合约当前的余额。
withdraw	函数允许合约所有者提取合约中的所有资金。
buyNFTWithToken	函数允许用户使用 ERC20 代币购买 NFT。函数更新市场项目的所有者信息，并将 NFT 转移给买家。
createAuctionListing	函数允许用户创建拍卖列表。函数创建拍卖时，我们需要设定一个起始价格、一个结束时间和一个触发拍卖创建的事件。
bid	函数允许用户竞拍 NFT。函数检查拍卖是否仍然开放，并验证竞拍金额是否高于当前价格。
completeAuction	函数允许卖家或竞拍赢家完成拍卖。
withdrawBid	函数允许竞拍人撤回竞拍。函数验证竞拍人不是最高竞拍者，并将竞拍金额退还给竞拍人。
isAuctionOpen	函数返回拍卖是否仍然开放。
isAuctionExpired	函数返回拍卖是否已经结束。
_transferFund	内部函数用于将资金转移给指定地址。
getHighestBidder	函数返回指定 Token ID 的最高竞拍人地址。
fetchMarketAuctionItems	函数返回市场中的所有拍卖项目。
fetchItemsAuctionListed	函数返回用户列出的所有拍卖项目。
getBidders	函数返回所有竞拍人的地址列表。
fetchUserBids	函数返回用户参与的所有竞拍。
fetchNFTBids	函数能够获取特定 NFT 的所有竞拍信息。

以 fetchMarketItems 函数为例，如图 3-2 所示，函数在 for 循环中遍历所有已铸造 NFT。在每次循环中检查当前 NFT 的所有者是否为合约本身（即地址 address(this)）。如

果条件成立，则表明该 NFT 尚未售出。将该 NFT 存储在 items 数组的当前位置，随后 currentIndex 递增以存储下一个未售出的 NFT。最后，函数返回该 items 数组，该数组包含了市场上所有未售出的 NFT 的详细信息。

同样地，fetchMyNFTs 函数的条件就是判断当前 NFT 的所有者（idToMarketItem[i＋1].owner）与调用函数的用户地址（msg.sender）是否相同，fetchItemsListed 函数的条件就是判断当前 NFT 的卖家（idToMarketItem[i＋1].seller）与调用函数的参与者（msg.sender）是否一致。

```
//GET ALL UNSOLD NFT

function fetchMarketItems() public view returns (MarketItem[] memory) {



    // 获取当前市场中的总物品数量

    uint itemCount = _tokenIds.current();

    // 计算未售出的物品数量 = 总物品数量 - 已售出的物品数量

    uint unsoldItemCount = _tokenIds.current() - _itemsSold.current();

    // 初始化当前索引，用于填充未售出的物品数组

    uint currentIndex = 0;



    // 创建一个大小为未售出物品数量的内存数组，用于存储未售出的物品

    MarketItem[] memory items = new MarketItem[](unsoldItemCount);



    // 遍历所有市场上的物品

    for (uint i = 0; i < itemCount; i++) {

        // 如果物品的所有者是合约本身（表示该物品未售出）
```

```
            if (idToMarketItem[i + 1].owner == address(this)) {
                // 获取当前物品的 ID
                uint currentId = i + 1;
                // 通过 ID 获取当前物品的详细信息
                MarketItem storage currentItem = idToMarketItem[currentId];
                // 将当前未售出的物品存入数组
                items[currentIndex] = currentItem;
                // 索引递增，指向下一个存储位置
                currentIndex += 1;
            }
        }

        // 返回未售出的物品数组
        return items;
    }
```

图 3-2 NFT 市场合约的 fetchMarketItems 函数

除了 NFT 市场合约，平台还通过集成多个智能合约增强了功能与用户体验。原生代币合约支持使用平台代币或以太坊（ETH）进行交易和支付；代币销售合约允许用户用以太坊购买原生代币，管理员可结束销售并提取收益；捐赠合约记录以太币捐赠信息，管理员可提取捐款；转账合约支持资金转账并记录交易详情；支持合约提供用户反馈功能并保存消息记录；社区合约创建区块链社交平台，支持账户创建、好友添加和消息发送，保障数据安全。这些合约构成了功能全面的去中心化生态系统。

3.3 实用函数和上下文管理

3.3.1 实用函数

实用函数是指在代码中用于执行常见或重复性任务的小型、通用函数。它们通常设计简洁、独立，并可在多处复用，有助于代码结构的优化。NFT 交易市场包括合约交互相关的实用函数及一些其他实用函数。

项目最主要的实用函数是与区块链技术互动的实用函数，如连接钱包、切换网络、获取用户余额，以及与智能合约交互。在合约交互的文件中定义了两个函数，FETCH_CONTRACT 是一个辅助函数，负责实例化一个以太坊合约的接口。CALLING_CONTRACT 是一个主函数，用于与用户钱包建立连接，并返回一个与智能合约交互的实例对象。这个函数主要用于确保用户的连接是安全且有效的，然后使用提供者（PROVIDER）来与合约进行交互。

此外，平台还包含多种实用函数以提升功能性与用户体验。例如，计算卖家总销售额的函数可从交易数组中返回各卖家的总额；随机字符串生成函数支持创建指定长度的字符串，用于标识或加密；地址缩短函数格式化区块链地址，仅显示前 5 位和后 4 位字符；时间转换函数可将输入时间转换为 JavaScript 的 Date 对象；转账记录函数展示所有或与用户相关的历史转账信息；社区相关函数支持用户查看群组成员、好友列表及消息记录。这些工具为应用提供了便捷性和多样化功能。

3.3.2 上下文管理

useContex 是 React 中的一个钩子（Hook），用于在函数组件中访问由 Context 提供的全局数据或状态。Context 是 React 提供的一种机制，允许我们在组件树中共享数据。在 React 的 Context API 中，图 3-3、3-4 和 3-5 展示了如何实现全局状态的管理与共享，并在各个组件中进行访问。

```
import React, { useState, useEffect } from "react";



// 创建一个用于 NFT 信息的上下文对象
```

```
export const NFTContext = React.createContext();



// 创建一个 NFTProvider 组件，用于为子组件提供上下文信息

export const NFTProvider = ({ children }) => {

  // 定义 NFT 市场的名称

  const NFT_MARKETPLACE = "DFashn DF";



  return (

    // 使用 NFTContext.Provider 提供上下文值，将 NFT_MARKETPLACE 传递给子组件

    <NFTContext.Provider value={{ NFT_MARKETPLACE }}>

      {children}  {/* 渲染子组件 */}

    </NFTContext.Provider>

  );

};
```

图 3-3 context/NFTContext.js 部分代码

```
import "../styles/globals.css";

import { NFTProvider } from "../context/NFTContext";



// 创建 Marketplace 组件，作为整个应用程序的布局

const Marketplace = ({ Component, pageProps }) => {

  return (

    // 将整个应用组件包裹在 NFTProvider 中，提供上下文
```

```
    <NFTProvider>
      <Component {...pageProps} />   {/* 渲染具体页面的组件 */}
    </NFTProvider>
  );
};


export default Marketplace;
```

图 3-4 pages/_app.js 部分代码

```
import React, { useContext } from "react";
import { NFTContext } from "../context/NFTContext";


// 创建 index 组件
const index = () => {
  // 使用 useContext Hook 从 NFTContext 中提取 NFT_MARKETPLACE
  const { NFT_MARKETPLACE } = useContext(NFTContext);


  return (
    <div>
      {/* 显示 NFT 市场名称 */}
      <h1>{NFT_MARKETPLACE}</h1>
    </div>
  );
```

```
};


// 导出 index 组件为默认导出

export default index;
```

图 3-5 pages/index.js 部分代码

在 NFTContext.js 中通过 React 的 useState 和 useEffect 钩子创建并管理一系列全局状态变量。这些变量有当前用户的以太坊账户、账户余额、NFT 市场的拍卖信息、转账历史记录、支持消息、代币销售状态、捐赠合同余额等。useEffect 钩子用于在组件挂载时初始化 NFT 市场、代币销售和捐赠合约的状态，并检查用户是否已连接他们的以太坊钱包。

Context API 在这里被用作共享状态的机制，使得这些全局状态和功能能够在整个 React 应用程序中被不同的组件访问。NFTContext 和 NFTProvider 定义了一个全局的 Context，并通过 NFTProvider 组件将 Context 中的状态和函数暴露给其子组件。

在 NFTContext 中创建合约实例，以便与智能合约进行互动。利用这些例子，合约中的各种函数可以被使用，直接在区块链网络上实现操作。此外，引入多个实用函数，为 NFT 交易市场、转账历史、支持信息和社区消息等功能提供数据和交互支持，简化了与合约的交互流程，并增强了平台的功能性和用户体验。

3.4 前端开发

3D 模型 NFT 交易平台的用户界面使用 React、Next.JS 和 TailwindCSS 三个前端框架进行构建。首先，React 作为 NFT 交易平台开发的基础库。React 是一个主要用于构建前端用户界面的 JavaScript 框架，通过组合可重复使用的组件来构建平台界面。其次，Next.js 作为交易平台的应用框架。Next.js 是 Vercel 开发的 React 框架，专为简化构建不依赖服务端渲染（SSR）应用设计。我们利用 Next.js 的 API 路由和文件路由功能实现前后端无缝集成，保证平台高效运转。最后，TailwindCSS 作为交易平台的 CSS 框架。TailwindCSS 通过提供一组预先定义的类简化了样式的编写，帮助我们快速建立响应式、现代化的用户界面，并保持代码的简洁。

在本项目的开发过程中，为了确保前端样式符合项目的具体设计需求，我们对 Tailwind CSS 进行了定制化配置。首先，通过修改 tailwind.config.js 文件，我们对 Tailwind CSS 的默认配置进行了调整，定义了项目所需的颜色、间距、字体和屏幕响应式布局等。

为了确保项目中的全局样式能够被统一管理和应用，我们在 styles/globals.css 文件中引入了所需的字体和 Tailwind 核心样式。使用@import 语句引入了外部字体库，保证了项目在不同设备和浏览器中的字体一致性。同时，文件还引入了 Tailwind CSS 的基础样式，保证所有页面都遵循基本一致的基础样式。这种集中式的样式管理方式使项目的前端结构层次清晰，更易于维护。

3.4.1 React 组件

项目没有选择传统数据库，而是通过智能合约直接与区块链交互，前端与区块链的通信通过 JavaScript 库实现，从而简化了中间数据层的设计。在前端开发方面，项目中创建了多个 React 组件，这些组件被细分为全局组件、主页组件、账户组件、管理组件、表格组件、弹窗组件和 SVG 图标组件等。全局组件用于处理应用程序的共享逻辑，如用户认证和状态管理；主页组件则展示了主要的 NFT 市场信息；账户组件涉及用户的数字资产管理和交易；管理组件用于管理和维护后台数据；表格组件用于显示复杂数据的列表，如交易记录或用户数据；弹窗组件用于交互式的提示和表单提交，而 SVG 图标组件则提供了可重用的图标元素。每个组件的职责明确，且通过 React 的组件化思想，提高了代码的复用性和可维护性。

项目的前端逻辑与智能合约的交互通过这种模块化的开发方式得到了优化，前端无需通过传统的数据库查询，就可以从区块链直接获取数据并且通过智能合约执行操作。这种架构确保了应用的透明性和可信度，使得用户的每个操作都可以被追溯。

以主页中的 3D 模型卡片为例。首先，创建了一个名为 ModelViewer 的 React 组件利用@Google/Model-Viewer 库实现 3D 模型的渲染。组件接收多个属性，包括 src（3D 模型的源地址）、alt（替代文本）、autoRotate（是否自动旋转）、cameraControls（用于控制摄像机的布尔值）以及 rotationSpeed（旋转速度）。根据传入的属性，组件将自动应用这些设置，并返回一个 model-viewer 元素来显示 3D 模型。

其次，创建了名为 NFTCard 的 React 组件用于显示 NFT 的卡片信息。组件使用了

useContext 钩子从 NFTContext 获取当前的加密货币符号，并使用 useState 钩子管理 3D 模型的是否自动旋转。在该组件中，通过 onMouseEnter 和 onMouseLeave 事件切换 autoRotate 状态，从而实现当鼠标悬停在卡片上时 3D 模型自动旋转的效果。此外，组件还展示了 NFT 的名称、价格以及卖家地址，并使用 shortenAddress 函数对卖家地址进行了简化显示。

3.4.2 React 页面

项目创建了一系列页面，包括主页、社区、创建 NFT、管理员等等，在这些页面中需要首先设置_app.js。_app.js 是 Next.js 框架中的一个特殊文件，用于自定义应用程序的根组件。通过_app.js，可以在所有页面中引入全局样式表、设置全局布局、以及保持某些组件在页面之间的一致性。它的主要作用是为整个应用提供统一的结构和功能，而无需在每个页面中重复代码。

在_app.js 中，引入了全局样式并导入 Navbar, Footer, Donate, DonateModal 这些全局组件，确保它们在应用的各个页面中都可以被使用。Navbar 组件位于页面的顶部，Footer 组件位于页面的底部。Component {...pageProps}是 Next.js 的一种模式，表示渲染当前页面的信息，这样每个页面都有对应的信息，而公共组件则在页面的头尾部分。

nft-details 是最复杂的页面之一，主要功能包括展示 NFT 的基本信息、处理购买和竞拍逻辑、以及动态加载模型视图组件。组件内部使用了多个 React Hook 来管理状态和生命周期。它从上下文中获取 NFT 相关的状态和函数，比如 buyNft 和 getSingleNFTBids，并在路由变更时加载对应的 NFT 数据。组件中还包含一个支付模态框、成功消息模态框，以及用于显示当前最高出价者和竞拍倒计时的功能。此外，它根据 NFT 的类别（图像、3D 模型或视频）选择不同的展示方式，使用 next/image 和 dynamic 从 Next.js 导入的动态组件来优化性能。

通过检查用户的账户地址和竞拍状态，系统判断是否显示不同的操作选项。也就是说，不同用户的访问该页面将呈现不同的内容。出售 NFT 的卖家在未设置拍卖时，会看到“设置拍卖”按钮；设置拍卖后，卖家可以查看各买家的出价列表。当拍卖结束时，卖家可以看到“结束拍卖”的操作按钮。其他用户可以参与竞标，在拍卖未结束时，当前出价最高的竞拍人会被告知他当前是最高出价者，且无法再出价。当拍卖结束时，出

价最高的竞拍人将能看到“结束拍卖”的按钮。

卖家（items.seller）或最高出价者（winner）中的任何一方点击“结束拍卖”，都会调用 completeAuction 函数来结束拍卖。completeAuction 函数首先检查 winner 是否为零地址（即没有赢家）。如果有赢家，继续执行拍卖结算如下所示：

1. 更新拍卖项信息：将拍卖项的所有权转移给赢家，将状态设置为已售，关闭拍卖。

2. 转移物品：通过_transfer 函数将拍卖的物品转移给赢家。

3. 清空竞标信息：清空与当前拍卖相关的所有竞标信息，包括取消最高出价者、用户出价信息以及与竞标相关的其他数据。

4. 更新拍卖计数：更新已售物品的计数和剩余竞标数量。

5. 转移资金：将竞标金额转移到卖家账户。

第四章 数据存储与管理说明书

4.1 链上存储和链下存储

在本次 NFT 市场合约设计中，我们对数据存储的方式进行了仔细的考量，主要采用了链上永久存储和链下临时存储相结合的方案。在区块链技术中，链上存储通常涉及较高的 Gas 成本，因此在设计时，我们需要对数据存储策略进行优化，以平衡存储的需求和交易成本。

链上存储是指将数据直接存储在智能合约的状态变量中，这些数据在区块链上永久存储，并且任何与之相关的操作都会产生 Gas 费用。在本合约中，所有与 NFT 相关的关键信息，如市场物品的详情、拍卖信息、竞标记录等，都采用了链上存储。这些数据通过 mapping 和 struct 的形式进行管理，确保了每个 NFT 在市场中的状态能够被正确追踪。例如，我们使用了 idToMarketItem 映射来存储每个 NFT 的详细信息，包含了 NFT 的所有者、价格、拍卖状态等信息。类似地，竞标相关的数据也被存储在 userBiddingNFT 映射中，其中包含了每个竞标者的出价记录、出价时间等信息。

虽然链上存储确保了数据的透明性和不可篡改性，但这种存储方式的缺点是 Gas 成本较高。每次对这些数据进行更新时，都需要支付一定的费用。因此，在设计时我们采取了优化措施，尽可能减少不必要的状态更新，尤其是避免频繁修改链上的数据。

为了避免因频繁更新链上数据而导致的 Gas 费用过高，我们采取了链下存储的方案，将一些不需要频繁查询或永久存储的数据放置在链下服务器上。例如，NFT 的元数据（如图片、描述等）通常存储在 IPFS 或其他分布式存储平台中，而非直接存储在链上。这不仅降低了 Gas 费用，还提高了存储效率和访问速度。通过这种方式，合约中的存储只保存与交易和拍卖相关的核心数据，如 NFT 的所有者、价格和竞标信息等，而其他信息则依赖于链下系统进行管理。

为了减少合约执行时的 Gas 成本，除了在链下存储方面的优化外，我们还在合约的逻辑设计上采取了多种优化策略。例如，在拍卖和竞标过程中，我们尽可能减少对链上数据的写入操作，而是通过计算和缓存机制将计算密集型的数据处理放到链下进行。这减少了每次竞标时对 mapping 的写入操作，从而降低了交易的 Gas 成本。此外，通过合并一些交易步骤，我们避免了多次的状态更新。例如，在 NFT 的购买和拍卖完成时，我

们通过一次性转账将资金从合约转移给卖家，从而避免了分多次执行的高 Gas 费用。

总的来说，本合约的设计在链上和链下存储之间找到了一个合理的平衡。通过合理地利用链上存储和链下存储的优势，并通过优化合约逻辑和减少不必要的数据更新，我们有效地降低了 Gas 成本，同时确保了系统的透明性和数据的不可篡改性。

4.2 IPFS 存储

IPFS（InterPlanetary File System，星际文件系统）是一种去中心化的分布式文件存储系统，旨在通过分布式网络存储和共享文件。与传统的 HTTP 协议不同，IPFS 使用内容寻址而非位置寻址来访问文件，这使得文件在全球范围内能够快速、稳定地访问。IPFS 通过将文件切割成小块，并将这些小块分布在不同的节点上，使得数据不再依赖于单个服务器，从而增强了文件的可用性和容错性。IPFS 广泛应用于去中心化应用（dApps）中，特别是在区块链和 NFT 领域，作为去中心化的存储解决方案，提供了安全、透明和高效的文件存储方式。

在我们的设计中，使用了 Pinata 作为 IPFS 的接口服务来上传文件，并将上传的文件内容存储在去中心化的 IPFS 网络中。Pinata 提供了 API 接口，通过这些接口，我们能够将文件和 JSON 元数据推送到 IPFS，并返回一个 IPFS 哈希链接，用户可以通过该链接访问存储在 IPFS 上的文件。这一设计是为了保证上传的 NFT 文件（如 3D 模型）在去中心化网络中的存储，同时避免集中化存储的风险。

在代码中，我们通过以下几个步骤实现了与 IPFS 的交互：

1. 文件上传功能：我们首先通过 useDropzone 组件实现了文件的上传功能，并对文件进行类型和大小的校验，确保文件是符合要求的 GLTF 或 GLB 格式，并且大小不超过 50MB。如果文件符合要求，调用 uploadToPinata 函数将文件上传到 Pinata。

2. Pinata 上传：在 uploadToPinata 函数中，我们使用 Axios 将文件通过 POST 请求上传到 Pinata 的 API 接口。上传成功后，Pinata 返回一个包含文件哈希（IpfsHash）的响应，我们将该哈希值拼接成一个可访问的 IPFS 文件链接，并将这个链接存储在 fileUrl 状态中，供后续使用。

3. NFT 创建：在创建 NFT 的过程中，我们将用户填写的 NFT 信息和上传的文件链接一起打包成 JSON 格式的数据，并通过同样的方式上传到 Pinata。当文件的元数据

上传成功后，我们会得到一个包含 IPFS 哈希的链接，这个链接作为 NFT 的元数据存储位置。

4. 展示与交互：在用户选择了文件后，上传的 3D 模型通过 ModelViewer 组件展示在页面上，用户可以在上传后直接查看其模型效果。这个交互方式不仅便于用户验证其上传的文件，同时通过动态加载技术提高了页面的性能。

通过这种方式，我们实现了文件的去中心化存储，确保了 NFT 的内容不会因单一服务器故障或其他集中式存储的问题而丢失或无法访问，同时也提升了 Web3 应用的透明度和可靠性。

第五章 系统安装及使用说明书

在安装和运行本系统前，请确保您已安装必要的软件工具。在 Visual Studio Code (VS Code)中打开项目文件夹，并确认本地环境中已安装 Node.js 和 npm 的指定版本，分别为 Node.js v22.3.0 和 npm v10.8.1。打开项目文件夹后，在 VS Code 的终端中输入“npm i”以安装所需的依赖包，然后在终端中输入“npm run dev”启动本地开发环境。在启动成功后，终端将显示本地开发地址为 http://localhost:3000。用户需在 Chrome 浏览器中访问该地址，以确保系统能够正确运行。

在首次访问页面时，用户需要检查浏览器是否已安装 MetaMask 插件。MetaMask 是一个常用的区块链钱包插件，用于连接去中心化应用程序（DApp）和签署相关交易。安装完成后，用户需按照提示创建或导入钱包账户，打开本地开发地址后会提示用户链接到正确的区块链网络。MetaMask 的安装和配置是系统运行的重要前提条件，未安装插件会导致交易功能无法正常使用。

系统运行后，点击导航栏的“Create”按钮，即可进入 NFT 创建页面。在该页面中，可以上传一个 3D 模型文件（支持 GLTF 或 GLB 格式），上传后模型将在界面中缓缓旋转（如图 5-1 所示）。用户需输入 NFT 的相关信息，包括名称、描述、价格（以 ETH 为单位）以及原生代币价格。完成信息填写后，点击“Create Item”按钮，MetaMask 插件会自动弹出提示支付相应的 Gas 费用。支付成功后，页面将自动跳转回主页。

回到主页后，可以看到刚刚创建的 NFT 已成功显示在界面中。当鼠标悬停在 NFT 卡片上时，该 NFT 将自动旋转（如图 5-2 所示）。切换 MetaMask 账户并刷新页面后，用户可点击新建的 NFT 进入其详情页面。在详情页面中，可以控制模型的旋转，并完成购买操作。此外，该平台还提供了黑暗模式，可通过切换主题按钮激活（如图 5-3 所示）。详细的演示视频已附在文件夹中。

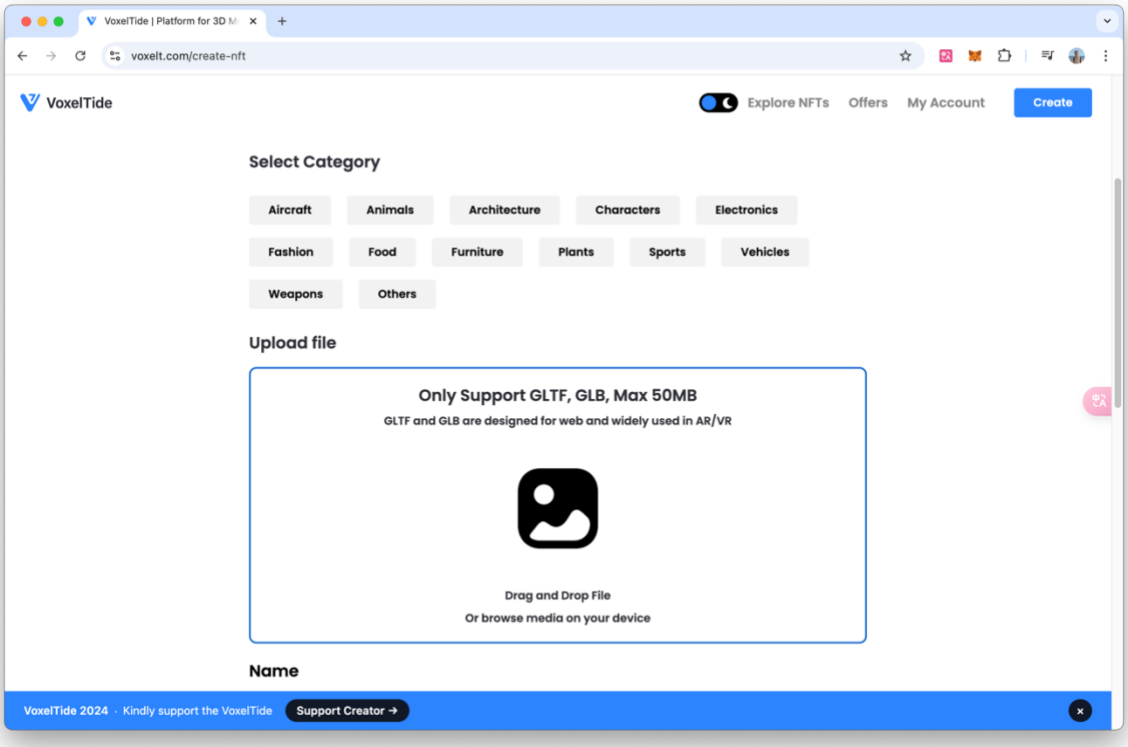

图 5-1 上传 NFT

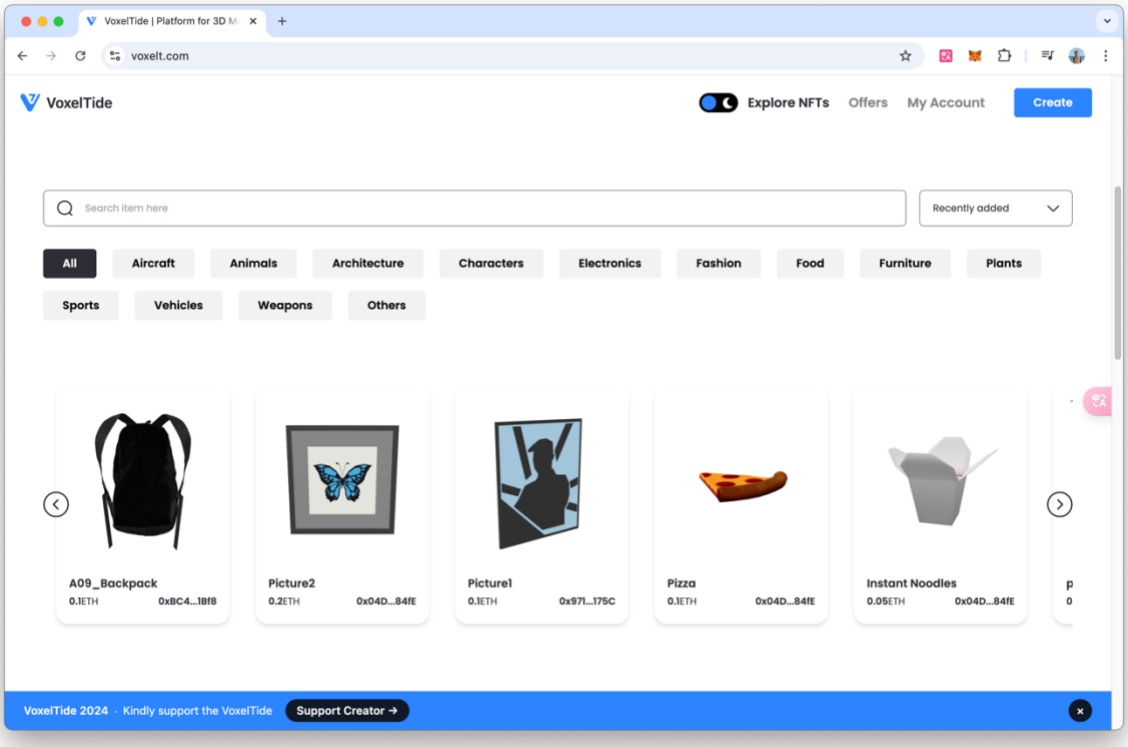

图 5-2 主页上的 NFT

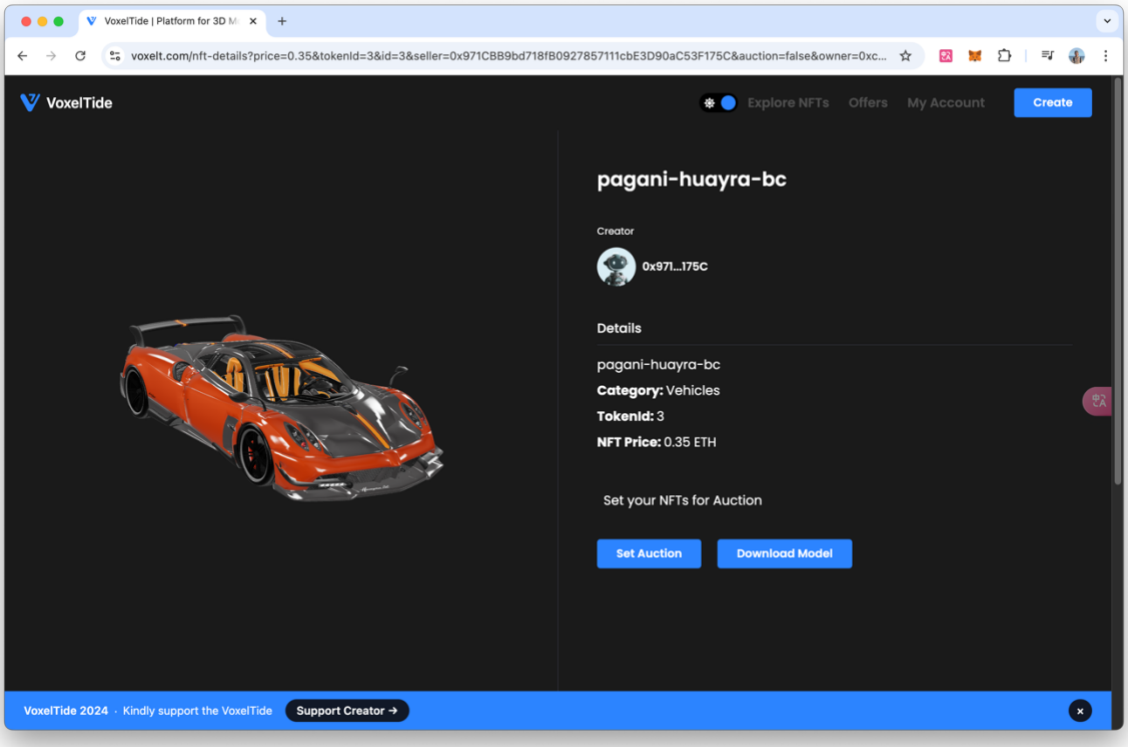
VoxelTide | Platform for 3D M
voxelt.com/nft-details?price=0.35&tokenId=3&id=3&seller=0x971CBB9bd718fB0927857111cbE3D90aC53F175C&auction=false&owner=0xc...
VoxelTide
Explore NFTs
Offers
My Account
Create
pagani-huayra-bc
Creator
0x971...175C
Details
pagani-huayra-bc
Category: Vehicles
TokenId: 3
NFT Price: 0.35 ETH
Set your NFTs for Auction
Set Auction
Download Model
VoxelTide 2024 · Kindly support the VoxelTide
Support Creator →

图 5-3 NFT 详情

参考文献

[1] Wang H, Ning H, Lin Y, et al. A survey on the metaverse: The state-of-the-art, technologies, applications, and challenges[J]. IEEE Internet of Things Journal, 2023, 10(16): 14671-14688.

[2] Hoffmann J F. Property Rights in Money (and Cryptocurrencies)[J]. European Property Law Journal, 2022, 11(3): 133-157.

[3] 李洁榆.基于区块链的知识产权共享系统设计与实现[D].南京理工大学,2023.DOI:10.27241/d.cnki.gnjgu.2023.000293.

[4] Han M, Yan M, Li J, et al. Generating uncertain networks based on historical network snapshots[C]//Computing and Combinatorics: 19th International Conference, COCOON 2013, Hangzhou, China, June 21-23, 2013. Proceedings 19. Springer Berlin Heidelberg, 2013: 747-758.

[5] Duan Z, Yan M, Cai Z, et al. Truthful incentive mechanisms for social cost minimization in mobile crowdsourcing systems[J]. Sensors, 2016, 16(4): 481.

[6] Li X, Jiang P, Chen T, et al. A survey on the security of blockchain systems[J]. Future generation computer systems, 2020, 107: 841-853.

[7] Tosh D K, Shetty S, Liang X, et al. Security implications of blockchain cloud with analysis of block withholding attack[C]//2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE, 2017: 458-467.

[8] Gilad Y, Hemo R, Micali S, et al. Algorand: Scaling byzantine agreements for cryptocurrencies[C]//Proceedings of the 26th symposium on operating systems principles. 2017: 51-68.

[9] Gangwal A, Gangavalli H R, Thirupathi A. A survey of Layer-two blockchain protocols[J]. Journal of Network and Computer Applications, 2023, 209: 103539.

[10] Zheng Z, Xie S, Dai H, et al. An overview of blockchain technology: Architecture, consensus, and future trends[C]//2017 IEEE international congress on big data (BigData congress). Ieee, 2017: 557-564.

[11] Joshi A P, Han M, Wang Y. A survey on security and privacy issues of blockchain technology[J]. Mathematical foundations of computing, 2018, 1(2).

[12] Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA)[J]. International journal of information security, 2001, 1: 36-63.

[13] Swanson T. Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems[J]. Report, available online, 2015, 28.

[14] Wood G. Ethereum: A secure decentralised generalised transaction ledger[J]. Ethereum project yellow paper, 2014, 151(2014): 1-32.

[15] Zamfir V. Introducing casper the friendly ghost[J]. Ethereum Blog, 2015.

[16] Castro M, Liskov B. Practical byzantine fault tolerance[C]//OsDI. 1999, 99(1999): 173-186.

[17] Wohrer M, Zdun U. Smart contracts: security patterns in the ethereum ecosystem and solidity[C]//2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE). IEEE, 2018: 2-8.

[18] Daulat Hussain. Pro NFT Marketplace[EB/OL]. theblockchaincoders, n.d. [2024-08-12]. https://www.theblockchaincoders.com/pro-nft-marketplace.

[19] VOGELSTELLER F. ERC20: Ethereum Token Standard[EB/OL]. https://eips.ethereum.org/EIPS/eip-20, 2015-11-19.