

Create an NFT Barter dApp

Siapa bangun proyek NFT yang seru? Di sini kita bakal bikin showcase NFT lengkap dengan fitur barter, jadi kamu bisa tukar-menukar NFT dengan user lain. Tenang, semua bakal kita pecah jadi langkah-langkah kecil biar gampang diikuti dan nggak bikin pusing. Karena level assignment ini **intermediate**, guide di bawah nggak terlalu detail jadi kamu bisa lebih mengasah skill-mu. Yuk kita mulai, pelan-pelan aja tapi tetap asik!

Contents

Prerequisites	3
Concept	3
NFT Metadata	3
transferFrom	4
Project Guide	4
Project Setup	4
Showcase the NFTs	5
Store the User's Showcase	5
Build the Barter System	6
Bonus Challenge: Barter Cross-Chain	6
Reminder	6
Documentations & References	7
Ethers.js	7
Hardhat	7
OpenZeppelin NFT (ERC-721)	7
Next.js	7
React	7
Reown (previously WalletConnect)	7
Prisma	7
PostgreSQL	7
PgAdmin	7
Tailwind CSS	7

Prerequisites

- Pemahaman dasar tentang smart contract
- Pemahaman standar ERC-721
- Pengalaman menggunakan Ethers.js atau library serupa
- Basic data management
- Pengalaman dengan JavaScript
- Fasih dengan Hardhat
- Basic CSS untuk frontend

Concept

Proyek ini bakal bikin platform simpel di mana user bisa showcase dan tukar-menukar (barter) NFT mereka. Jadi, user bisa connect wallet, lalu masukan contract address dan token ID dari NFT yang mereka punya. Platform ini bakal tampilin info lengkap dari NFT-nya, seperti nama, deskripsi, dan gambar. Info ini juga bisa disimpan ke database, biar user bisa lihat lagi kapan aja. Jadi, gampangnya, user bisa bikin galeri NFT mereka sendiri yang langsung terhubung ke blockchain.

Selain itu, ada fitur barter juga! User bisa ajukan tukar-menukar NFT dengan user lain. Cukup masukan detail contract address dan token ID dari NFT yang mau ditukar, plus NFT yang diinginkan dari user lain. Nanti, user yang nerima request bisa milih mau approve atau tolak barter-nya. Kalau disetujui, NFT langsung ditransfer otomatis di blockchain.

NFT Metadata

NFT metadata adalah informasi yang menggambarkan detail dari NFT, seperti gambar, nama, deskripsi, dan atribut khusus lainnya. Metadata ini biasanya disimpan dalam format JSON dan berisi data penting yang menggambarkan NFT secara visual dan fungsional. Contoh metadata bisa mencakup image yang berisi link gambar NFT (bisa disimpan di IPFS), name untuk menamai NFT, description untuk menjelaskan latar belakang atau cerita di balik NFT tersebut, serta attributes yang berisi karakteristik unik seperti tipe "spaceship", warna, dan berbagai kemampuan atau fitur lain yang membuat setiap NFT berbeda. Metadata ini yang akhirnya memberikan NFT nilai unik dan menarik bagi para kolektor.

```
{
  "image": "ipfs://QmaZYFB1YWJUrSHYMM23Ta2D1wR7YLYARhL5qvcsfaHngX",
  "animation_url": "ipfs://QmUHGdW8Bd41dJjxvaZQbaxJErHxux7FB9ihm9ibVQ56Y9",
  "name": "Space Doodle #3",
  "description": "Space Doodles are spaceships for your Doodle. Each Doodle in the original collection of 10,000 can launch their own personal spaceship that is generated from over 200 audio-visual traits and on-chain stats.\n\nYour Space Doodle is your Doodle and selling it will allow the new purchaser to remove the Doodle from our secure smart contract. It is highly recommended that you launch and dock your Doodles directly from https://doodles.app",
  "attributes": [
    { "trait_type": "spaceship", "value": "saucer mk4" },
    { "trait_type": "thruster", "value": "fire" },
    { "trait_type": "sector", "value": "blue" },
    { "trait_type": "nearest planet", "value": "bulbous" },
    { "trait_type": "rank", "value": 1 },
    { "trait_type": "piloting ability", "value": 1, "max_value": 8 },
    { "trait_type": "mechanical knowledge", "value": 1, "max_value": 8 },
    { "trait_type": "stamina", "value": 1, "max_value": 8 },
    { "trait_type": "bladder management", "value": 2, "max_value": 8 },
    { "trait_type": "vibe captaining", "value": 1, "max_value": 8 }
  ]
}
```

NFT metadata biasanya bisa didapatkan melalui fungsi **tokenURI** dari contract NFT dengan memasukkan **tokenId** yang sesuai. Fungsi tokenURI akan mengembalikan URL yang mengarah ke metadata tersebut dalam format JSON, yang berisi informasi seperti gambar, nama, deskripsi, dan

atribut unik lainnya. Metadata ini penting untuk menampilkan detail visual dan karakteristik dari NFT secara langsung di aplikasi.

transferFrom

Dalam standar ERC-721 untuk NFT, tidak ada fungsi **transfer** langsung seperti di ERC-20 (token biasa). Sebagai gantinya, ada **transferFrom** dan **safeTransferFrom**, dan ini ada beberapa alasan penting terkait keamanan dan fleksibilitas.

- **transferFrom**: Fungsi ini digunakan untuk mentransfer NFT dari satu alamat ke alamat lain, dengan syarat pemanggilnya memiliki izin untuk melakukannya. Fungsi ini berguna saat kamu ingin mentransfer NFT yang dimiliki oleh orang lain (misalnya melalui marketplace), asalkan kamu sudah mendapatkan persetujuan dengan fungsi **approve**. Namun, **transferFrom** tidak secara otomatis memastikan penerima dapat menerima NFT, sehingga ada risiko NFT dikirim ke contract atau alamat yang tidak bisa menanganinya.
- **safeTransferFrom**: Fungsi ini hampir sama dengan **transferFrom**, tetapi dengan lapisan keamanan tambahan. Ketika mentransfer NFT ke contract, **safeTransferFrom** akan mengecek apakah contract tersebut mendukung penerimaan NFT (misalnya, dengan mengimplementasikan **ERC721Receiver**). Jika tidak, transaksi akan dibatalkan. Ini mencegah NFT hilang atau terkunci di contract yang tidak mendukung NFT.

Alasan kenapa tidak ada **transfer** langsung di ERC-721 adalah karena standar ini mendesain NFT agar lebih aman dan fleksibel, terutama saat berurusan dengan contract. **safeTransferFrom** memastikan NFT hanya dikirim ke tempat yang aman, sementara **transferFrom** memberikan kontrol lebih pada pengguna yang mengatur izin transfer.

Feel free to try it yourself before reading the guide.

Project Guide

Guide ini bukan full guide sampai dApp jadi, melainkan hanya panduan agar project development lebih terarah. Kamu bisa styling dApp kapan pun.

Project Setup

Goal: Get a React atau Next.js app running with wallet connection.

1. Kickstart the Project

Pertama, bikin proyek baru pakai Next.js. Kita pakai **Next.js 14** biar routing-nya gampang. Kalo kamu lebih nyaman pakai React, silakan!

```
npx create-next-app@14 nft-barter
```

Quick Check: Jalankan npm run dev, pastikan app muncul di browser.

2. Install Ethers.js dan Reown (WalletConnect)

Install Ethers.js untuk berinteraksi dengan Ethereum blockchain dan Reown (sebelumnya WalletConnect) untuk connect ke berbagai wallet.

```
npm install @reown/appkit @reown/appkit-adapter-ethers ethers
```

3. Integrasi Connect Wallet

Tambahkan tombol **Connect Wallet** untuk connect wallet menggunakan MetaMask atau Reown (WalletConnect). Setelah wallet terhubung, tampilkan alamat wallet di layar. Cek documentation untuk cara integrasi Reown.

Quick Check: Pastikan wallet bisa terhubung dan alamat wallet muncul di halaman. Kalau udah muncul, kamu siap lanjut ke tahap berikutnya!

Showcase the NFTs

Goal: User bisa input **contract address** dan token ID untuk menampilkan metadata NFT.

1. ERC-721 Overview

ERC-721 adalah standar umum untuk NFT. Beberapa fungsi penting yang sering digunakan:

- **balanceOf (address)**: Menampilkan berapa banyak NFT yang dimiliki oleh alamat tertentu.
- **ownerOf (tokenId)**: Menunjukkan siapa pemilik dari NFT dengan token ID tertentu.
- **tokenURI (tokenId)**: Mengembalikan URL metadata untuk NFT.

2. Ambil Metadata NFT

Buat form atau input di mana user bisa input **contract address** dan **token ID** NFT yang mereka miliki. Gunakan **Ethers.js** untuk memanggil **tokenURI** dan ambil metadata (biasanya berupa JSON yang berisi gambar, nama, deskripsi, dan atribut unik lainnya).

Quick Check: Coba tes dengan NFT terkenal (misal BAYC) atau cari di OpenSea dan pastikan metadata muncul. Bisa tes menampilkan image dan nama NFT.

Store the User's Showcase

Goal: User bisa menyimpan NFT yang mereka tampilkan ke database dengan mapping ke alamat wallet atau username. Showcase bisa diakses oleh orang lain dan bisa di edit ulang dengan connect wallet halaman edit.

1. Setup PostgreSQL dan Prisma

Setup database seperti PostgreSQL dan PgAdmin4 untuk database, lalu install Prisma untuk ORM.

```
npm install prisma
npx prisma init
```

Buat schema untuk model **User** dan **NFTShowcase** di Prisma. Design the schema agar NFT showcase yang disusun user bisa ditampilkan ulang.

2. SaveShowcase

Simpan susunan NFT showcase-nya. Buat **dynamic routing** yang memungkinkan user untuk melihat ulang showcase-nya dengan mengakses halaman berdasarkan alamat wallet atau username. **Halaman untuk edit dan showcase (non-editable) beda route.**

Quick Check: Pastikan data NFT dapat disimpan dan diambil kembali dari database dengan benar berdasarkan user showcase.

Design the UI agar tampilan NFT showcase menarik.

Build the Barter System

Goal: Buat sistem di mana user bisa tukar-menukar NFT dengan user lain.

1. Code the Contract

Buat smart contract yang mendukung fungsi **approve** dan **transferFrom** untuk barter NFT. Pastikan contract bisa menangani barter beberapa NFT sekaligus. Setelah selesai, deploy ke jaringan testnet pilihanmu. Berikut guide untuk function dasar yang perlu ada:

- **approve()**: Memberikan izin kepada contract untuk memindahkan NFT milik user.
- **settleBarter()**: Memindahkan NFT dari satu alamat ke alamat lain setelah barter disetujui dengan **transferFrom**.
- **makeOffer()**: Membuat penawaran barter dengan memasukkan contract address dan token ID dari kedua NFT yang ingin dipertukarkan.
- **acceptOffer()**: Accept barter offer dan memicu transfer NFT dari kedua belah pihak. Sekaligus function untuk reject (tergantung parameter).
- **cancelOffer()**: Cancel barter offer yang sudah dibuat oleh user.

Testing bisa dilakukan dengan Hardhat.

2. Design the UI for reviewing barters

Agar dApp lebih usable, bikin UI untuk review barters. Berikut bagian yang perlu ada:

- Offer list: Daftar offer yang diterima dengan detail-nya. Tambahkan action buttons untuk accept atau reject.
- Barter history: Daftar transaksi barter yang tereksekusi.
- My offer: Daftar offer yang user tersebut buat. Tambahkan action button untuk cancel.

Quick Check: Pastikan semua langkah dari request hingga transfer NFT berjalan dengan benar dan aman melalui smart contract.

Bonus Challenge: Barter Cross-Chain

Kalau kamu mau tantangan lebih, coba implementasikan barter NFT cross-chain! Kamu bisa menggunakan protokol seperti **LayerZero** untuk membuat cross-chain dApp. Ini adalah tantangan yang lebih advanced, tetapi sangat menarik untuk dipelajari dan dieksplorasi.

Reminder

Pastikan semua **credential** (seperti API key atau RPC URL) disimpan dengan aman di file **.env** menggunakan library **dotenv**. Jika kamu meng-upload proyek ke GitHub atau platform lainnya, jangan lupa menambahkan **.gitignore** untuk mengabaikan file yang tidak perlu di-upload, seperti **node_modules**, **.env**, dan file sensitif lainnya. Tetap jaga keamanan datamu!

Happy coding! 🚀

Documentations & References

Berikut ini daftar dokumentasi yang perlu kamu cek untuk membantu proyek ini. Simpan baik-baik karena ini bakal jadi andalan kamu selama proses coding!

Ethers.js

Ethers.js dipakai untuk berinteraksi dengan blockchain Ethereum. Library ini bikin kerja dengan smart contract jadi jauh lebih mudah. Link: <https://docs.ethers.io>

Hardhat

Hardhat adalah environment untuk pengembangan smart contract yang sangat membantu, terutama untuk testing dan deployment. Link: <https://hardhat.org/getting-started/>

OpenZeppelin NFT (ERC-721)

OpenZeppelin menyediakan kontrak ERC-721 yang aman untuk dipakai dalam membuat NFT. Link: <https://docs.openzeppelin.com/contracts/2.x/api/token/erc721>

Next.js

Next.js adalah framework yang direkomendasikan untuk membuat UI di proyek ini. Powerful dan sudah built-in dengan banyak fitur. Link: <https://nextjs.org/docs>

React

React adalah library untuk membangun UI dengan komponen. Alternatif lain jika kamu tidak menggunakan Next.js. Link: <https://reactjs.org/docs/getting-started.html>

Reown (previously WalletConnect)

Reown digunakan untuk menghubungkan wallet secara aman, memungkinkan user untuk login dan berinteraksi dengan dApp menggunakan berbagai wallet, tidak hanya MetaMask. Ini sangat berguna untuk integrasi connect wallet di proyek kamu. Link: <https://docs.reown.com/>

Prisma

Prisma digunakan sebagai ORM untuk mengelola database, termasuk PostgreSQL yang kita pakai di proyek ini. Link: <https://www.prisma.io/docs>

PostgreSQL

PostgreSQL adalah database yang digunakan untuk menyimpan data NFT dan barter di proyek ini. Link: <https://www.postgresql.org/docs/>

PgAdmin

PgAdmin adalah alat grafis yang memudahkan pengelolaan database PostgreSQL. Link: <https://www.pgadmin.org/docs/>

Tailwind CSS

Tailwind CSS bikin styling UI jadi lebih cepat dan fleksibel. Link: <https://tailwindcss.com/docs>