

ART EUM

Vendi e acquista le tue opere



Eljon Hida mat.0522501890



INTRODUZIONE

Arteum è una DApp (Decentralized Application) che permette agli utenti di *acquistare, vendere* opere d'arte digitali sotto forma di **NFT** (Non-Fungible Tokens). La piattaforma si focalizza in particolare sulla compravendita di quadri, sfruttando la tecnologia Blockchain.

SOMMARIO



01

**TECNOLOGIE
USATE**

02

**FUNZIONALITÀ
PRINCIPALI**

03

**DESIGN
PATTERNS**

04

DEMO DApp

TECNOLOGIE 01 UTILIZZATE

The background features a complex technical illustration. It includes several interlocking gears of different sizes, some in solid black and others in light gray. There are also various geometric shapes like squares and circles, some with internal patterns, and a series of small dots arranged in a line. The overall aesthetic is clean and modern, with a focus on mechanical and technological motifs.

In questa sezione vi sarà l'elenco delle
varie tecnologie usate per permettere
lo sviluppo di Arteum

TECNOLOGIE UTILIZZATE

Tecnologia	Descrizione
Hardhat	In quanto framework di sviluppo offre tutto quello di cui si necessita oltre a offrire una rete locale che simula Ethereum
Pinata IPFS	Usato per salvarvi gli asset digitali una volta creati e messi in vendita
React	Per avere un frontend più intuitivo
Solidity & JS	Usati entrambi per il backend: il primo usato per scrivere gli Smart Contract di Arteum ed il secondo usato nel deploy e testing

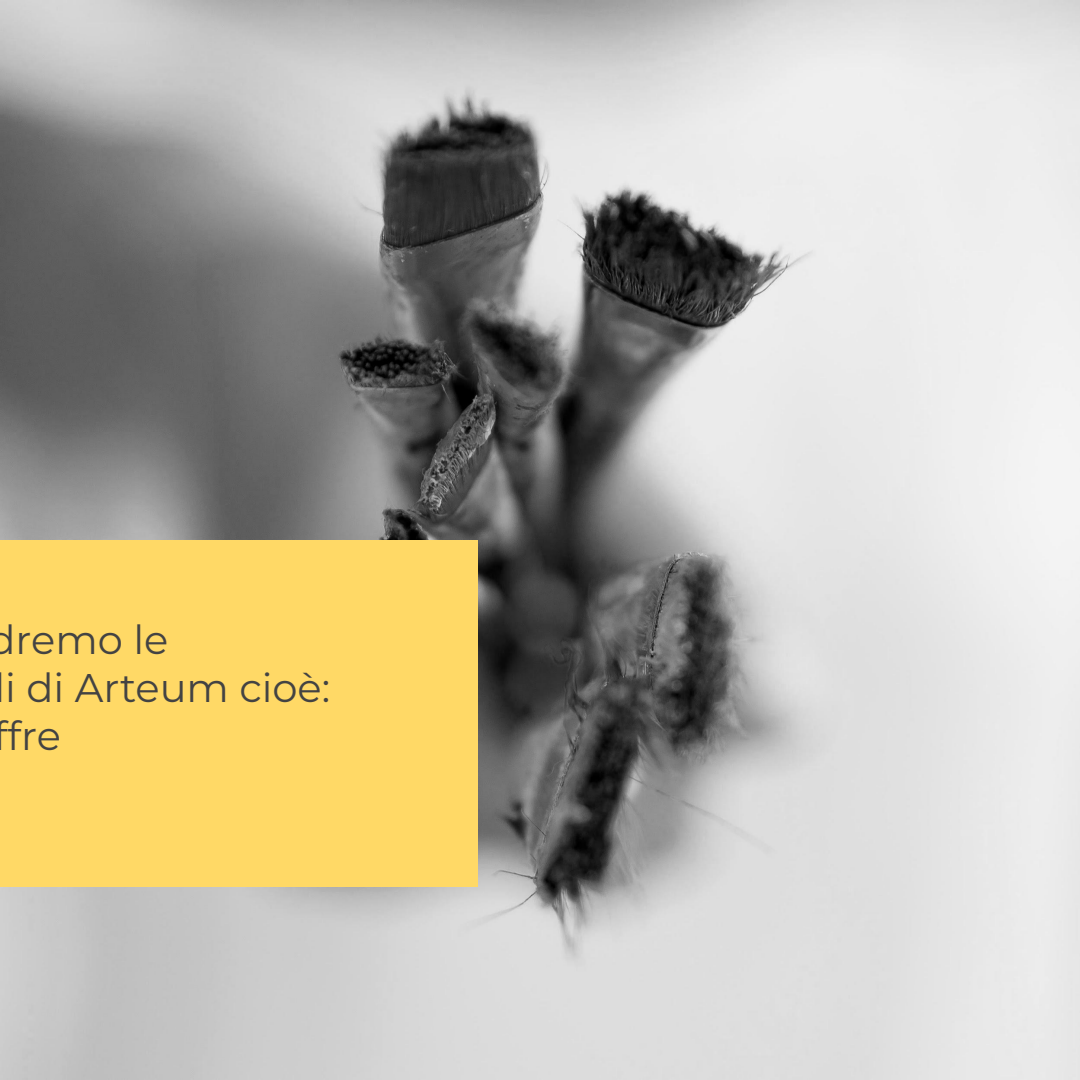
TECNOLOGIE UTILIZZATE

Tecnologia	Descrizione
Ethers.js	Usata per interagire con Hardhat permettendo il deploy, testing e connessione al wallet
Node.js	Usato per introdurre nel progetto le dipendenze necessarie al suo corretto funzionamento
Metamask	Usato per gestire, non solo le transazioni tra gli utenti, ma anche gli accessi e l'autenticazione alla piattaforma;

FUNZIONALITÀ PRINCIPALI

02

In questa sezione vedremo le
funzionalità principali di Arteum cioè:
quello che Arteum offre



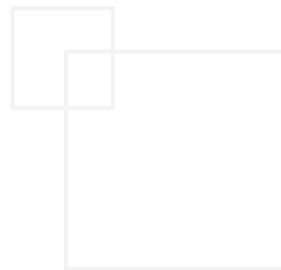
FUNZIONALITÀ PRINCIPALI

ACQUISTO NFT

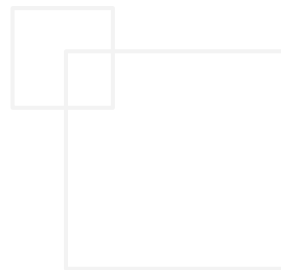
Un utente potrà interagire con la DApp solo dopo aver fatto l'accesso con Metamask, avendo così la possibilità di vedere i vari prodotti nel catalogo e acquistarli.

VENDITA NFT

Un utente potrà creare e, di conseguenza, vendere i propri NFT solo dopo aver fatto l'accesso con Metamask; gli basterà cliccare su Crea NFT e inserire le varie informazioni (nome, descrizione e prezzo) associate a quest'ultimo.



FUNZIONALITÀ PRINCIPALI



VISIONE DELLA PROPRIA LISTA NFT

Un utente potrà interagire con la DApp solo dopo aver fatto l'accesso con Metamask, avendo così la possibilità di vedere non solo i prodotti acquistati ma anche i prodotti che possiede che ha messo in vendita.

DESIGN GOAL



DESIGN GOAL

Tra gli obiettivi raggiunti abbiamo:

- **Decentralizzazione:** il sistema salva i dati degli NFT su Pinata per garantire sicurezza e decentralizzazione;
- **Semplicità d'uso:** l'interfaccia è user-friendly e offre, quindi, un'esperienza utente semplice ed intuitiva.
- **Sicurezza delle transazioni:** avendo usato alcuni design pattern si è assicurata la sicurezza nelle transazioni.

DESIGN PATTERN

03

In questa sezione vedremo i Design
Pattern implementati in Arteum



DESIGN PATTERN

CHECKS EFFECT INTERACTIONS

Come prima cosa sono stati fatti tutti i controlli, dopodiché sono stati fatti i vari aggiornamenti di stato e, solo allora, c'è stata l'interazione esterna.

```
// Acquisto di un NFT
function purchaseItem(uint _itemId) external payable nonReentrant {
    uint _totalPrice = getTotalPrice(_itemId);
    Item storage item = items[_itemId];
    require(_itemId > 0 && _itemId ≤ itemCount, "item doesn't exist");
    require(
        msg.value ≥ _totalPrice,
        "not enough ether to cover item price and market fee"
    );
    require(!item.sold, "item already sold");
    // Aggiornamento dell'item allo stato venduto
    item.sold = true;
    // Pagamento del venditore e dell'account al quale si deve il compenso
    item.seller.transfer(item.price);
    feeAccount.transfer(_totalPrice - item.price);
    // Trasferimento del NFT all'acquirente
    item.nft.transferFrom(address(this), msg.sender, item.tokenId);
}
```

DESIGN PATTERN

SECURE ETHER TRANSFER

Qui è stata usata una transfer al posto della send o call.value per vari motivi:

- Non c'era bisogno più di 2300 gas per svolgere l'operazione;
- Si voleva ricevere un feedback in caso di errore;

```
// Acquisto di un NFT
function purchaseItem(uint _itemId) external payable nonReentrant {
    uint _totalPrice = getTotalPrice(_itemId);
    Item storage item = items[_itemId];
    require(_itemId > 0 && _itemId ≤ itemCount, "item doesn't exist");
    require(
        msg.value ≥ _totalPrice,
        "not enough ether to cover item price and market fee"
    );
    require(!item.sold, "item already sold");
    // Aggiornamento dell'item allo stato venduto
    item.sold = true;
    // Pagamento del venditore e dell'account al quale si deve il compenso
    item.seller.transfer(item.price);
    feeAccount.transfer(_totalPrice - item.price);
    // Trasferimento del NFT all'acquirente
    item.nft.transferFrom(address(this), msg.sender, item.tokenId);
}
```

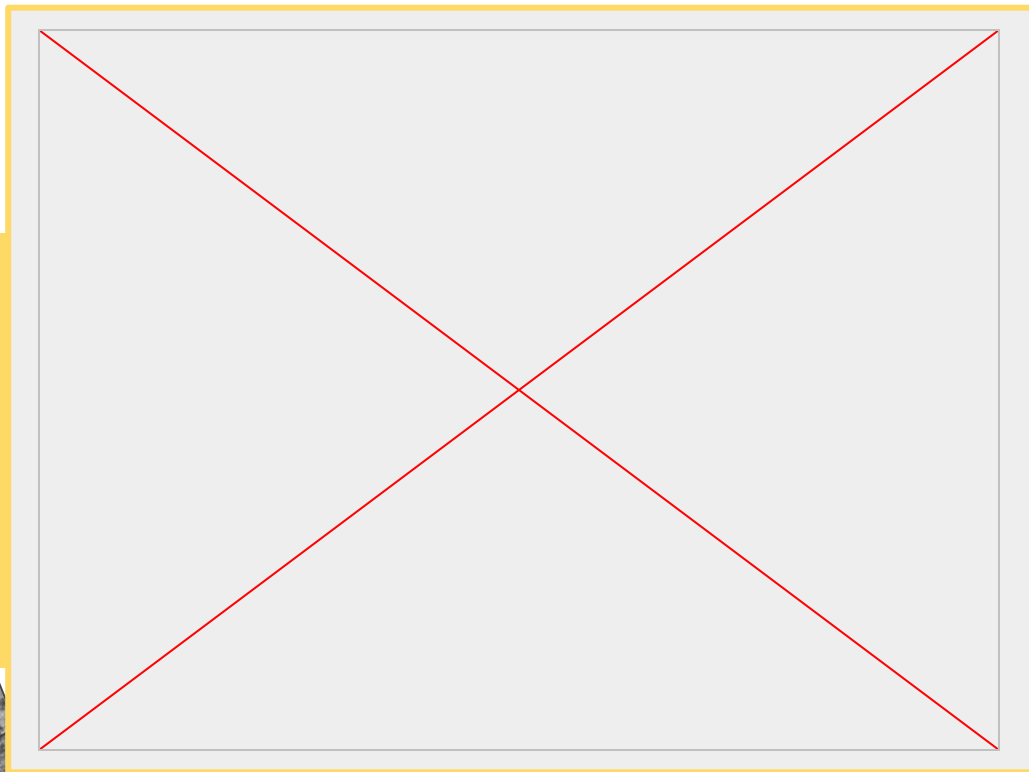
DEMO DAPP

04

In questa sezione vedremo un breve video che mostra Arteum, il suo funzionamento nonché le sue funzionalità



DEMO DAPP



**GRAZIE
PER L'ATTENZIONE**

