

## DApp marketplace

### Report

**Εργαλεία που χρησιμοποιήθηκαν:** Ganache, MetaMask, Web3.js, Remix.

#### Επισκόπηση και σύντομη περιγραφή:

Αναπτύχθηκε μια web-DApp “Summer Marketplace” που τρέχει σε τοπικό δίκτυο Ganache. Ο χρήστης συνδέεται με MetaMask, βλέπει το ενεργό address στη σελίδα και:

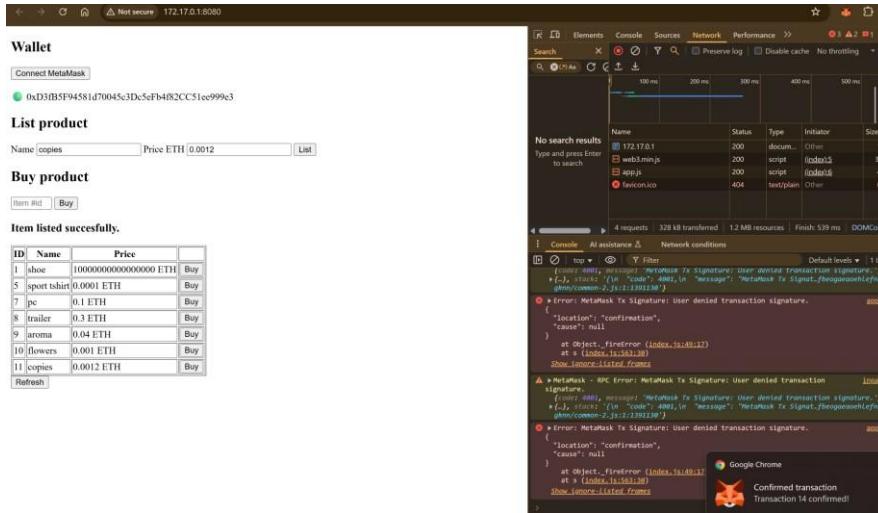
- List Product – καταχωρεί νέο προϊόν (όνομα και τιμή σε ETH) η τιμή μετατρέπεται σε wei εσωτερικά και αποθηκεύεται στο smart contract.
- Πίνακας Listings – φέρνει από το contract όλα τα διαθέσιμα items του ίδιου chain ID και τα εμφανίζει σε πραγματικό χρόνο.
- Buy – κάθε γραμμή έχει κουμπί Buy όπου ο αγοραστής υπογράφει τη συναλλαγή, στέλνει ακριβώς το ποσό σε wei και το item μαρκάρεται ως **sold**.
- Όλα τα δεδομένα (προϊόντα, πωλήσεις) μένουν στην αλυσίδα – οποιοσδήποτε λογαριασμός στο ίδιο δίκτυο βλέπει το ίδιο status. Επιβεβαίωση αυτού μέσω Ganache που φαίνονται όλα τα transactions/ Blocks.

ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS	SEARCH FOR BLOCK NUMBERS OR TX HASHES
CURRENT BLOCK 21	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK MERGE	NETWORK ID 1337	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING
BLOCK 21	MINED ON 2025-05-21 12:42:54	GAS USED 38707				<button>1 TRANSACTION</button>
BLOCK 20	MINED ON 2025-05-21 12:40:42	GAS USED 117688				<button>1 TRANSACTION</button>
BLOCK 19	MINED ON 2025-05-21 12:40:17	GAS USED 117712				<button>1 TRANSACTION</button>
BLOCK 18	MINED ON 2025-05-21 12:40:06	GAS USED 117676				<button>1 TRANSACTION</button>
BLOCK 17	MINED ON 2025-05-21 12:39:47	GAS USED 117712				<button>1 TRANSACTION</button>
BLOCK 16	MINED ON 2025-05-21 12:38:55	GAS USED 117652				<button>1 TRANSACTION</button>
BLOCK 15	MINED ON 2025-05-21 12:35:36	GAS USED 38707				<button>1 TRANSACTION</button>
BLOCK 14	MINED ON 2025-05-21 12:35:17	GAS USED 38707				<button>1 TRANSACTION</button>
BLOCK 13	MINED ON 2025-05-21 12:34:59	GAS USED 117664				<button>1 TRANSACTION</button>

To front-end είναι απλό HTML/JS με Web3.js, το back-end ένα Solidity smart-contract (SummerMarketplace.sol) που διαχειρίζεται τα listings και κάνει κάποιους ελέχγους όπως require για τιμή > 0, έλεγχος sold.

H list product φορμα πραγματοποιεί καταχώρηση νέων items (όνομα item, τιμή σε ETH

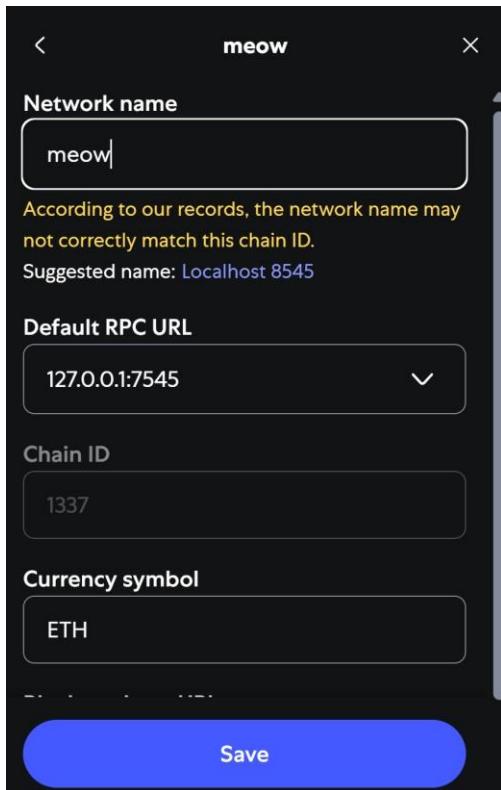
(αποθηκεύεται σε wei)), η buy product – πραγματοποιεί αγορά βάζοντας το Id του προϊόντος (item) που ενδιαφέρει, και ο πίνακας που φέρνει όλα τα items που έχουν καταχωρηθεί για το συγκεκριμένο Chain ID /contract address.



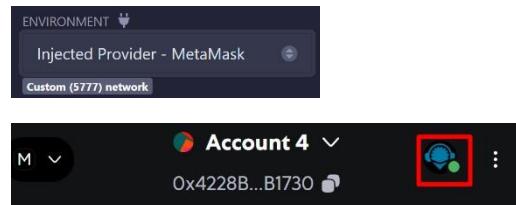
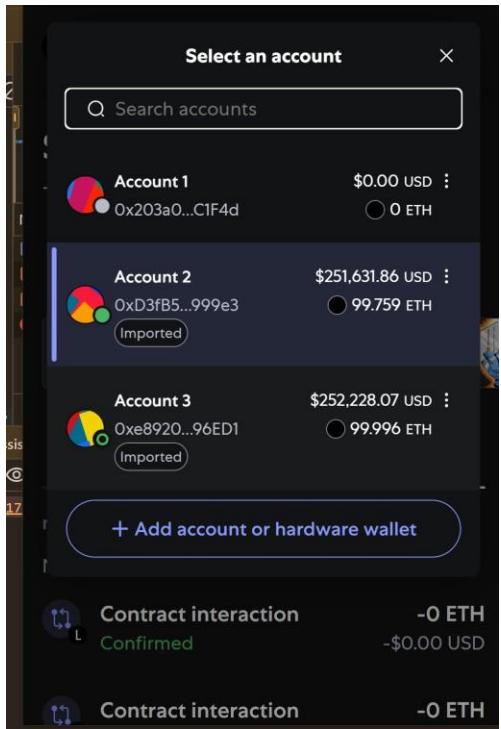
Ganache accounts (αντιγραφούμε το private key για testing του Dapp από όσα accounts θέλουμε minimum πχ. 2 για να προσομοιώσουμε αγοραπωλησία) και συνδεόμαστε στο metamask ανάλογα κάθε φορά. Ένα γενικό σχήμα θα ήταν να δημιουργήσουμε listing με account1 και αγορά με το account2. Βλέπουμε παρακάτω ανάλογα με τα transactions που έχουν πραγματοποιηθεί, να έχουν διαμορφωθεί και τα balances.

Blockchain Status								Logs		
Accounts		Blocks		Transactions		Contracts		Events		Logs
CURRENT BLOCK 21	GAS PRICE 20000000000	GAS LIMIT 6721975	HARDFORK MERGE	NETWORK ID 1337	RPC SERVER HTTP://127.0.0.1:7545	MINING STATUS AUTOMINING	WORKSPACE QUICKSTART	SAVE	SWITCH	⚙️
MNEMONIC <span>?</span>	HD PATH m44'60'0'0account_index									
aunt distance deal shoulder milk patrol flash enter talent melt either volume										
ADDRESS 0xD3FB5F94581d70045c3Dc5eFb4f82CC51ee999e3	BALANCE 99.76 ETH	TX COUNT 15			INDEX 0	🔗				
ADDRESS 0xe89206CbC4ce47e7BEDC981529Cea8A137296ED1	BALANCE 99.99 ETH	TX COUNT 6			INDEX 1	🔗				
ADDRESS 0xC22e7DBf1D1c9ED75374FfeBAC8bf9645f034D24	BALANCE 100.00 ETH	TX COUNT 0			INDEX 2	🔗				

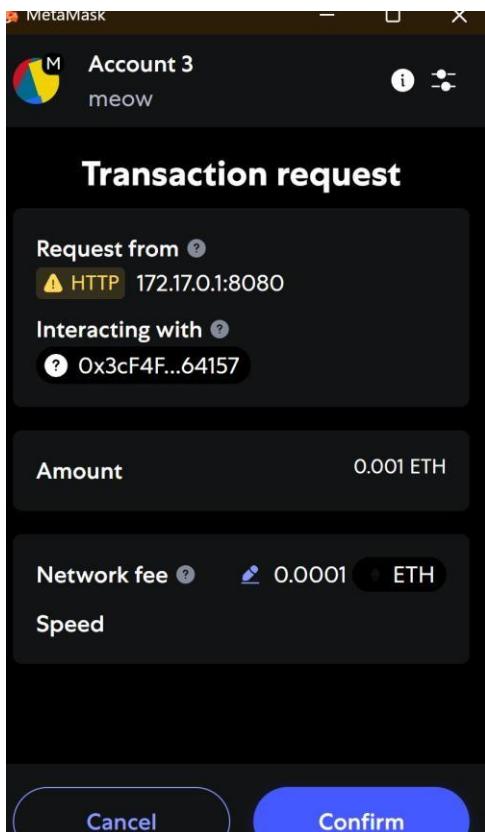
Προηγούνται τα βήματα τις διασύνδεσης του Remix με το metamask για το chain id που ενδιαφέρομαστε και το RPC URL = χρήση του test network που φτιάχαμε.



Επιλογή / Δημιουργία account



Πραγματοποιούμε ένα οποιοδήποτε transaction – interaction with Metamask – παράδειγμα αγορά :



## Αποτέλεσμα στον browser

## Wallet

Connect MetaMask

● 0xe89206CbC4ce47e7BEDC981529Cea8A137296ED1

## List product

Name \_\_\_\_\_ Price ETH \_\_\_\_\_ List \_\_\_\_\_

## Buy product

Item #id Buy

✓ bought item : 10

Μπορούμε να δούμε τα products από το remix επισης, γράφοντας το contract address στο At Address

### field (Deploy and Run transactions)

### Δομή του smart contract SummerMarketplace

**struct Product { id, name, price(wei), seller, sold }** και **mapping(uint ⇒ Product) products**

#### Καταχώρηση:

**listProduct(name, priceWei):** διαβάζει input σε ETH, το μετατρέπει σε wei (**toWei**) και καλεί **contract.methods.listProduct** (contract's function).

**require(price > 0)** → δημιουργεί προϊόν, αποθηκεύει σε wei, δίνει διαδοχικό id.

**Αγορά : buyProduct(id).send({from:user, value: prod.price})**

**buyProduct(id) payable:**

Ελέγχει ότι το item δεν έχει πουληθεί και ότι msg.value == product.price. Θέτει sold=true και μεταφέρει τα ETH στον seller όταν η συναλλαγή είναι επιτυχής.  
from:user = η διεύθυνση που πληρώνει. value: prod.price = στέλνει ακριβώς τόσα wei όσα κοστίζει το προϊόν.

Πορεία: Ο πίνακας φιλτράρει το id και το priceWei στο buyProduct, MetaMask υπογράφει, Ganache εξορύσσει, το UI ανανεώνεται (status).

**buyRow:** κουμπί αγοράς στον πίνακα με τα listings. Τα listings που εμφανίζονται όλα υπάρχουν και αν όχι το transaction θα αποτύχει. Έχουμε ήδη το priceWei. Εκτελεί την buyProducts του contract.

#### Βοηθητικά functions:

**getProduct(id)**

**totalProducts():** για χρήση από το js/ html αρχείο. Στο loadListings() εκτελεί totalProducts() και **getProduct(id)** για κάθε id.

#### Web3.js

```
async function connectWallet() {
    if (!window.ethereum) return alert("Install MetaMask");

    web3 = new Web3(window.ethereum);
    await window.ethereum.request({ method: "eth_requestAccounts" });
    user = (await web3.eth.getAccounts())[0];
    account.innerText = ` ${user}`;
    contract = new web3.eth.Contract(contractABI, contractAddress);
}
```

Challenges:

1. Internal JSON-RPC error κατά την ανάπτυξη, δεν ήταν bug του συμβολαίου αλλά εμφανιζόταν όποτε οι ρυθμίσεις MetaMask, Ganache, Remix δεν ταίριαζαν (λάθος RPC URL/chain-ID, πολύ gas, λάθος contract address , ABI στο js.
2. Gas estimate timeout: οφειλόταν στο ότι ο λογαρισμός που προσπαθούσα να κάνω deploy δεν είχε ETH, άρα η eth\_estimateGas απέτυχε.
3. Χρειάζεται redeploy στην επανεκκίνηση του Ganache αν χρειάζεται καινούργια έκδοση συμβολαίου.
4. Κάποιες φορές το MetaMask κολλούσε και δεν έκανε fetch το balance από accounts.
5. Έπρεπε να αλλάξει το ENV Version από advanced configuration στον compiler Remix σε κάτι άλλο πχ, paris, shanghai πέρα από το default Prague γιατί δεν υποστηριζόταν από Ganache.