Assignment 2

Answer 1

`0x5bD66994eb2f117f91B74aE6969Ef237eF04b7c0`

Answer 2

Ethereum is a blockchain platform designed to enable the development and execution of smart contracts and decentralized applications (dApps) without the involvement of intermediaries. It broadens the application of blockchain technology beyond just digital currencies. Smart contracts on Ethereum are self-executing scripts that activate when predefined conditions are satisfied. These contracts enhance security and streamline processes by eliminating the need for third parties. The platform operates through the Ethereum Virtual Machine (EVM), which guarantees that applications function as intended, regardless of their execution environment.

Originally, Ethereum used the Proof of Work (PoW) consensus mechanism for validating transactions but later adopted Proof of Stake (PoS) to improve energy efficiency and scalability. It also introduced gas fees, which are small payments required for executing transactions or running contracts on the network. Ethereum serves as the backbone for various advancements, including Decentralized Finance (DeFi), Non-Fungible Tokens (NFTs), and Decentralized Autonomous Organizations (DAOs).

Answer 3

I was not able to deploy the contract as it kept saying `Not enough balance, you need to buy more ETH.`

Here is the contract code:

```
1 pragma solidity ^0.8.22;
2
```

```
import {ERC721} from "@openzeppelin/contracts/token/ERC721/
    ERC721.sol";
import {ERC721URIStorage} from "@openzeppelin/contracts/token
    /ERC721/extensions/ERC721URIStorage.sol";
import {Ownable} from "@openzeppelin/contracts/access/Ownable
    .sol";

contract MyToken is ERC721, ERC721URIStorage, Ownable {
    uint256 private _nextTokenId;

    constructor(address initialOwner)
        ERC721("MyToken", "MTK")
        Ownable(initialOwner)
    {}

    function safeMint(address to, string memory uri) public
        onlyOwner {
        uint256 tokenId = _nextTokenId++;
        _safeMint(to, tokenId);
        _setTokenURI(tokenId, uri);
    }

    function tokenURI(uint256 tokenId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (string memory)
    {
        return super.tokenURI(tokenId);
    }

    function supportsInterface(bytes4 interfaceId)
        public
        view
        override(ERC721, ERC721URIStorage)
        returns (bool)
    {
        return super.supportsInterface(interfaceId);
    }
}
```

Answer 4

A wallet in blockchain acts as a tool for users to send, receive, and manage cryptocurrencies securely. It does not store the cryptocurrencies themselves but instead holds the private keys needed to access and authorize transactions

on the blockchain.

The key distinction between software and hardware wallets lies in their storage approach:

- **Software Wallets:** These are digital platforms, either apps or programs, that store private keys on devices like desktops, mobiles, or in the cloud. They are user-friendly and accessible but more prone to cyberattacks.
- **Hardware Wallets:** These are physical devices designed to keep private keys offline, providing robust security against online risks. They are less convenient but ideal for safeguarding assets over the long term.

Blockchain wallets maintain decentralization principles since they do not manage the funds or blockchain itself. Instead, they serve as tools for accessing the blockchain, with users retaining full control via their private keys.

Examples:

- Trust Wallet
- Ledger Nano S (Hardware)
- Atomic Wallet

Answer 5

The Zero Address in blockchain, commonly represented as `0x0000000000000000000000000000000000000000` in Ethereum, serves as a special placeholder address without an associated private key. This address is used for various purposes, such as removing tokens or assets from circulation or as a default value where an address is unspecified.

The private key for the Zero Address is effectively impossible to derive because of the enormous computational difficulty posed by the cryptographic hashing process and the vast key space of 2^{256} in Ethereum.

Answer 6

The Brave browser, built on the Chromium platform like Google Chrome, is known for its enhanced privacy settings, ad-blocking capabilities, and its unique Basic Attention Token (BAT) feature. Compared to Chrome, Brave is more efficient in resource usage, consuming less RAM, making it a favorite for many users.

The BAT system allows users to earn tokens by engaging with ads and other activities within the browser. These tokens can then be used to support content creators by sending them as contributions.

Answer 8

Here is the Python code to generate an Ethereum wallet:

```python
from eth_keys import keys
import os

def create_ethereum_wallet():
    random_key_bytes = os.urandom(32)
    private_key_obj = keys.PrivateKey(random_key_bytes)

    public_key_obj = private_key_obj.public_key
    eth_address = public_key_obj.to_address()

    print("Private Key:", private_key_obj)
    print("Public Key:", public_key_obj)
    print("Ethereum Address:", eth_address)

create_ethereum_wallet()
```

Answer 9

Here is the Python code to generate a contract address:

```python
from web3 import Web3
from eth_utils import to_checksum_address

def generate_contract_address(sender_address, nonce):
    w3 = Web3()
    sender_address_bytes = bytes.fromhex(sender_address[2:])
    nonce_bytes = nonce.to_bytes(32, 'big')
    data = sender_address_bytes + nonce_bytes
    contract_address_hash = w3.keccak(data)
    contract_address = contract_address_hash[-20:]
    contract_address = to_checksum_address('0x' +
       contract_address.hex())
    return contract_address

sender_address = '0x742d35cc6634c0532925a3b844bc454e4438f44e'
nonce = 5
contract_address = generate_contract_address(sender_address,
   nonce)
```

```
print("Generated Contract Address:", contract_address)

sender_address = '0xb299d668A2E852008A20eC721C1Bd1Dbd34ACBAa'
nonce = 0
contract_address = generate_contract_address(sender_address,
    nonce)
print("Generated Contract Address:", contract_address)
```