NAME: SHARON PHILIP

ROLL NO: 6961

Decentralized Music Streaming Smart Contract

BLOCKCHAIN PROJECT

MSC.IT (PART II)

NAME: Sharon Philip

ROLL NO: 6961

NAME: SHARON PHILIP

ROLL NO: 6961

INTRODUCTION

Blockchain technology is transforming various industries by enabling decentralized, transparent, and secure systems. One such industry that benefits from blockchain is music streaming. Traditional music streaming platforms are centralized, which means that they control content distribution, revenue sharing, and user data. This centralization often leads to issues like unfair revenue distribution, high intermediary fees, lack of transparency, and copyright concerns.

By integrating blockchain into music streaming, we can create a decentralized music streaming service where artists have direct control over their content, revenue is distributed fairly, and users enjoy a transparent and trust less ecosystem.

PURPOSE

The purpose of a decentralized music streaming platform is to leverage blockchain technology to create a fair, transparent, and secure music distribution ecosystem. Traditional music streaming services are controlled by centralized entities that take a sizeable portion of the revenue, leaving artists with minimal earnings and limited control over their content. Decentralized music streaming aims to empower artists and listeners by eliminating intermediaries and ensuring fair compensation for creators.

OBJECTIVE

A Decentralized Music Streaming Smart Contract aims to create a trust less, transparent, and fair ecosystem for artists, listeners, and stakeholders in the music industry. By leveraging blockchain technology, smart contracts ensure secure transactions, automated royalty distribution, and direct artist-to-fan interactions without intermediaries.

NAME: SHARON PHILIP

ROLL NO: 6961

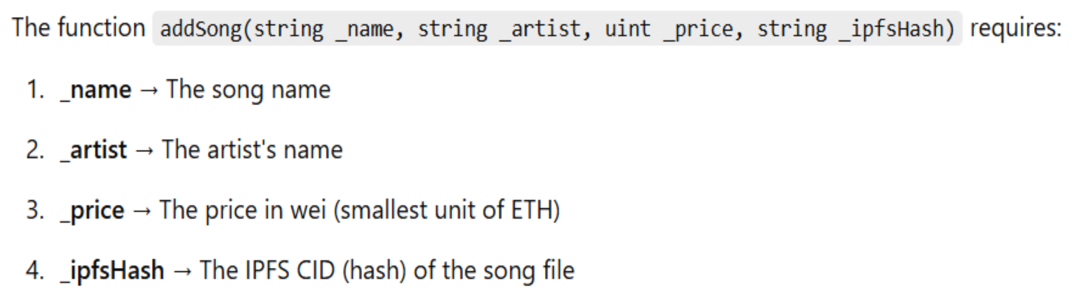
SYSTEM REQUIREMENTS

- **Hardware Requirements**: Computer with internet access

- **Software Requirements**: Remix Ethereum IDE

OPERATIONS / FEATURES

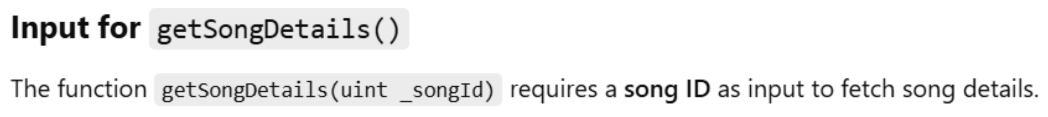
1. Inputs for add Song ()



The function `addSong(string _name, string _artist, uint _price, string _ipfsHash)` requires:

1. **_name** → The song name
2. **_artist** → The artist's name
3. **_price** → The price in wei (smallest unit of ETH)
4. **_ipfsHash** → The IPFS CID (hash) of the song file

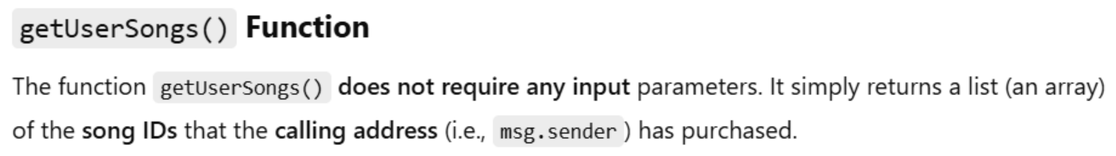
2. Input for get Song Details ()



Input for `getSongDetails()`

The function `getSongDetails(uint _songId)` requires a **song ID** as input to fetch song details.

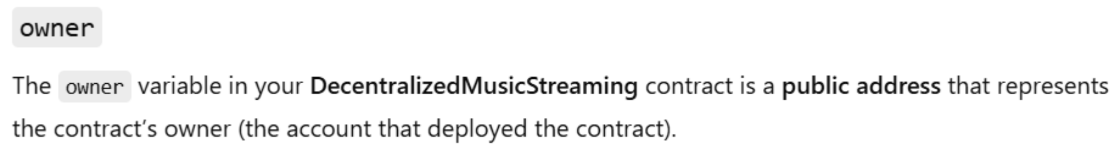
3. Get User Song



`getUserSongs()` **Function**

The function `getUserSongs()` **does not require any input** parameters. It simply returns a list (an array) of the **song IDs** that the **calling address** (i.e., `msg.sender`) has purchased.

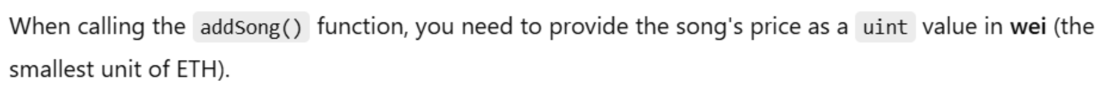
4. Owner



`owner`

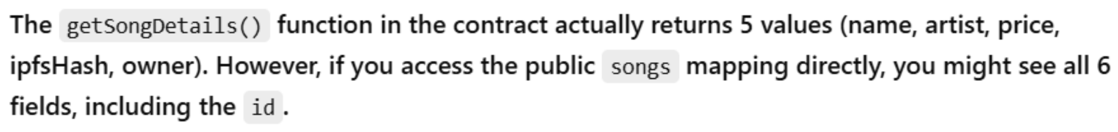
The `owner` variable in your **DecentralizedMusicStreaming** contract is a **public address** that represents the contract's owner (the account that deployed the contract).

5. Song Price



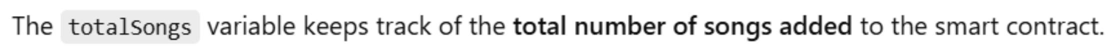
When calling the `addSong()` function, you need to provide the song's price as a `uint` value in **wei** (the smallest unit of ETH).

6. Song



The `getSongDetails()` function in the contract actually returns 5 values (name, artist, price, ipfsHash, owner). However, if you access the public `songs` mapping directly, you might see all 6 fields, including the `id`.

7. Total Song

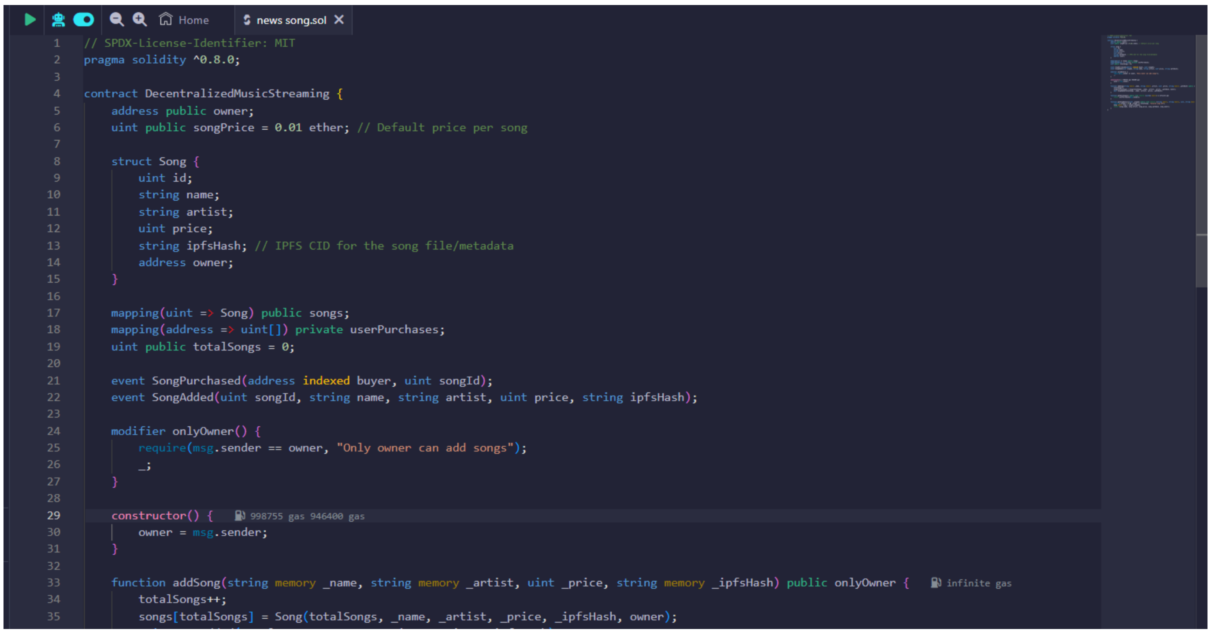


The `totalSongs` variable keeps track of the **total number of songs added** to the smart contract.

NAME: SHARON PHILIP

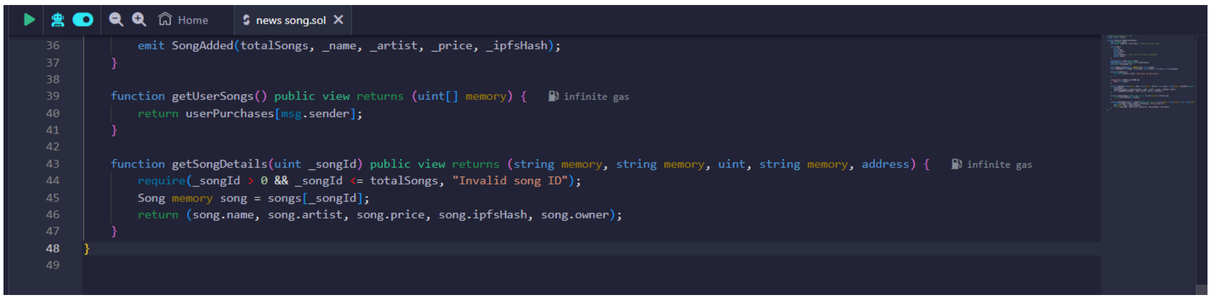
ROLL NO: 6961

CODE



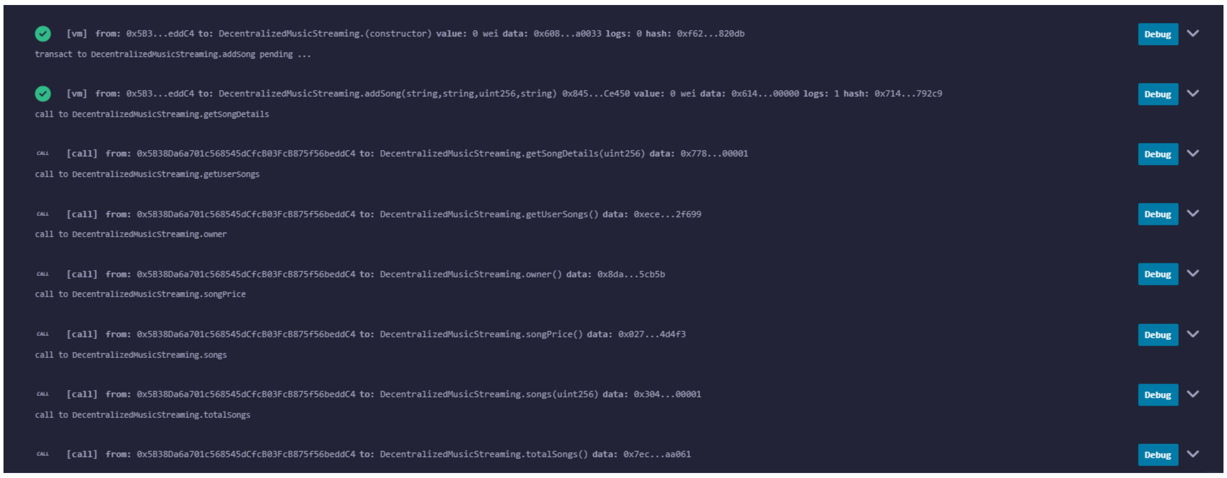
Home   news song.sol

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract DecentralizedMusicStreaming {
    address public owner;
    uint public songPrice = 0.01 ether; // Default price per song

    struct Song {
        uint id;
        string name;
        string artist;
        uint price;
        string ipfsHash; // IPFS CID for the song file/metadata
        address owner;
    }

    mapping(uint => Song) public songs;
    mapping(address => uint[]) private userPurchases;
    uint public totalSongs = 0;

    event SongPurchased(address indexed buyer, uint songId);
    event SongAdded(uint songId, string name, string artist, uint price, string ipfsHash);

    modifier onlyOwner() {
        require(msg.sender == owner, "Only owner can add songs");
        _;
    }

    constructor() {    998755 gas 946400 gas
        owner = msg.sender;
    }

    function addSong(string memory _name, string memory _artist, uint _price, string memory _ipfsHash) public onlyOwner {    infinite gas
        totalSongs++;
        songs[totalSongs] = Song(totalSongs, _name, _artist, _price, _ipfsHash, owner);
```



Home   news song.sol

```solidity
        emit SongAdded(totalSongs, _name, _artist, _price, _ipfsHash);
    }

    function getUserSongs() public view returns (uint[] memory) {    infinite gas
        return userPurchases[msg.sender];
    }

    function getSongDetails(uint _songId) public view returns (string memory, string memory, uint, string memory, address) {    infinite gas
        require(_songId > 0 && _songId <= totalSongs, "Invalid song ID");
        Song memory song = songs[_songId];
        return (song.name, song.artist, song.price, song.ipfsHash, song.owner);
    }
}

```

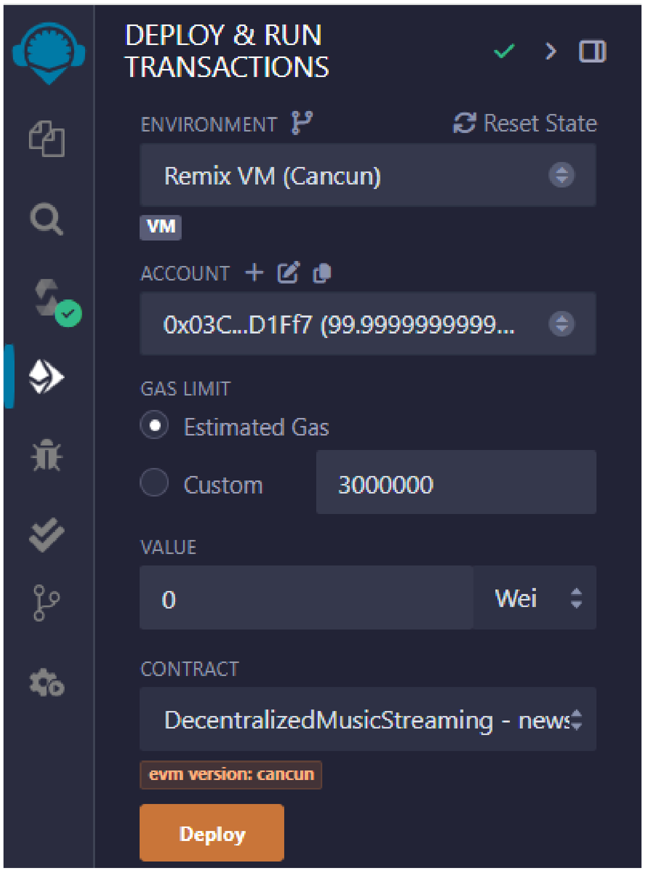


```
✓  [vm]  from: 0x5B3...eddC4 to: DecentralizedMusicStreaming.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0xf62...820db    Debug
transact to DecentralizedMusicStreaming.addSong pending ...

✓  [vm]  from: 0x5B3...eddC4 to: DecentralizedMusicStreaming.addSong(string,string,uint256,string) 0x845...Ce450 value: 0 wei data: 0x614...00000 logs: 1 hash: 0x714...792c9    Debug
call to DecentralizedMusicStreaming.getSongDetails

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.getSongDetails(uint256) data: 0x778...00001    Debug
call to DecentralizedMusicStreaming.getUserSongs

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.getUserSongs() data: 0xece...2f699    Debug
call to DecentralizedMusicStreaming.owner

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.owner() data: 0x8da...5cb5b    Debug
call to DecentralizedMusicStreaming.songPrice

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.songPrice() data: 0x027...4d4f3    Debug
call to DecentralizedMusicStreaming.songs

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.songs(uint256) data: 0x304...00001    Debug
call to DecentralizedMusicStreaming.totalSongs

CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.totalSongs() data: 0x7ec...aa061    Debug
```

NAME: SHARON PHILIP

ROLL NO: 6961

OUTPUT



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Remix VM (Cancun)

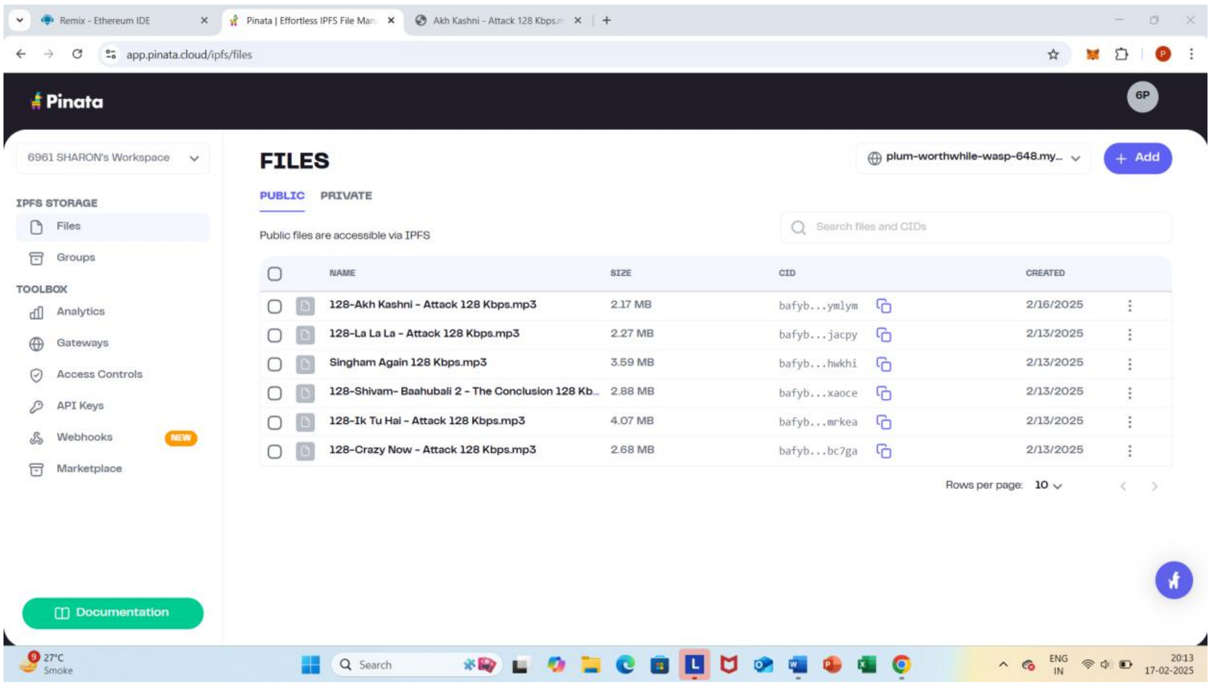
ACCOUNT: 0x03C...D1Ff7 (99.9999999999...

GAS LIMIT: Estimated Gas [✓]; Custom 3000000

VALUE: 0 Wei

CONTRACT: DecentralizedMusicStreaming - news

evm version: cancun



app.pinata.cloud/ipfs/files

FILES

6961 SHARON's Workspace

PUBLIC

Public files are accessible via IPFS

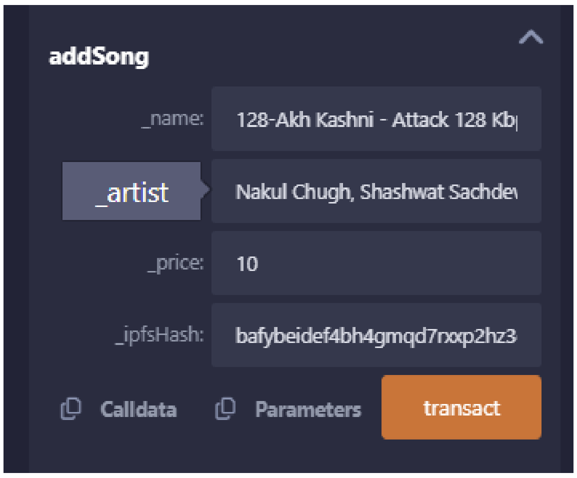
NAME	SIZE	CID	CREATED
128-Akh Kashni - Attack 128 Kbps.mp3	2.17 MB	bafyb...ymlym	2/16/2025
128-La La La - Attack 128 Kbps.mp3	2.27 MB	bafyb...jacpy	2/13/2025
Singham Again 128 Kbps.mp3	3.59 MB	bafyb...hwkhi	2/13/2025
128-Shivam- Baahubali 2 - The Conclusion 128 Kb...	2.88 MB	bafyb...xaoce	2/13/2025
128-Ik Tu Hai - Attack 128 Kbps.mp3	4.07 MB	bafyb...mrkea	2/13/2025
128-Crazy Now - Attack 128 Kbps.mp3	2.68 MB	bafyb...bc7ga	2/13/2025

Rows per page: 10

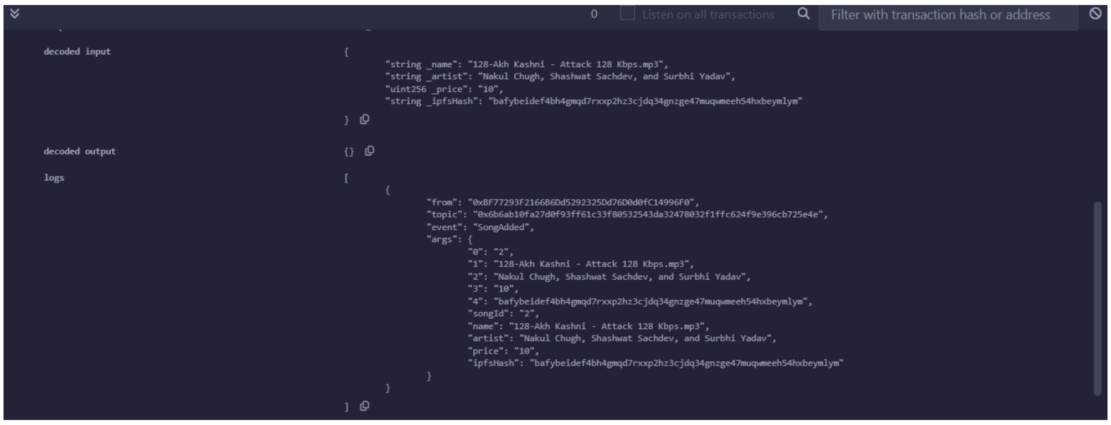
NAME: SHARON PHILIP

ROLL NO: 6961

1. **add Song ()**



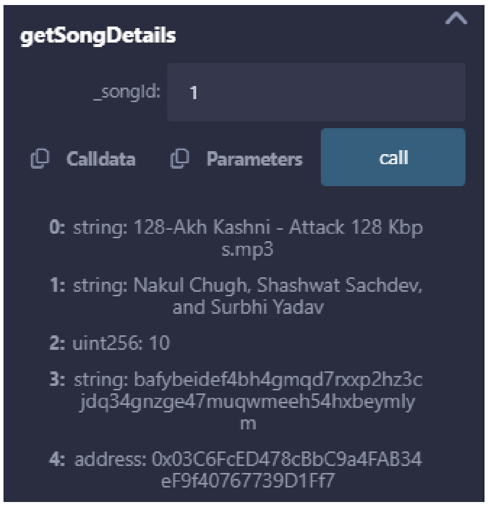
```
addSong
_name:      128-Akh Kashni - Attack 128 Kb
_artist     Nakul Chugh, Shashwat Sachde
_price:     10
_ipfsHash:  bafybeidef4bh4gmqd7rxxp2hz3
```



0
Listen on all transactions
Filter with transaction hash or address

```
decoded input      {
                          "string _name": "128-Akh Kashni - Attack 128 Kbps.mp3",
                          "string _artist": "Nakul Chugh, Shashwat Sachdev, and Surbhi Yadav",
                          "uint256 _price": "10",
                          "string _ipfsHash": "bafybeidef4bh4gmqd7rxxp2hz3cjdq34gnzge47muqwmeeh54hxbeymlym"
                   }

decoded output     {}

logs               [
                          {
                                 "from": "0xBF77293F2166B6Dd5292325Dd76D0d0fC14996F0",
                                 "topic": "0x6b6ab10fa27d0f93ff61c33f80532543da32478032f1ffc624f9e396cb725e4e",
                                 "event": "SongAdded",
                                 "args": {
                                        "0": "2",
                                        "1": "128-Akh Kashni - Attack 128 Kbps.mp3",
                                        "2": "Nakul Chugh, Shashwat Sachdev, and Surbhi Yadav",
                                        "3": "10",
                                        "4": "bafybeidef4bh4gmqd7rxxp2hz3cjdq34gnzge47muqwmeeh54hxbeymlym",
                                        "songId": "2",
                                        "name": "128-Akh Kashni - Attack 128 Kbps.mp3",
                                        "artist": "Nakul Chugh, Shashwat Sachdev, and Surbhi Yadav",
                                        "price": "10",
                                        "ipfsHash": "bafybeidef4bh4gmqd7rxxp2hz3cjdq34gnzge47muqwmeeh54hxbeymlym"
                                 }
                          }
                   ]
```

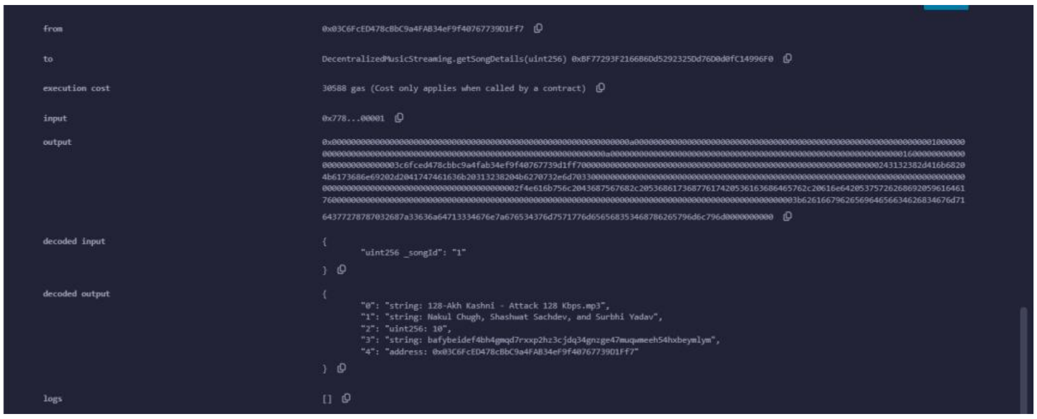
2. **get Song Details ()**



```
getSongDetails
_songId:  1

0: string: 128-Akh Kashni - Attack 128 Kbp
s.mp3
1: string: Nakul Chugh, Shashwat Sachdev,
and Surbhi Yadav
2: uint256: 10
3: string: bafybeidef4bh4gmqd7rxxp2hz3c
jdq34gnzge47muqwmeeh54hxbeymly
m
4: address: 0x03C6FcED478cBbC9a4FAB34
eF9f40767739D1Ff7
```

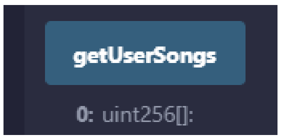
NAME: SHARON PHILIP

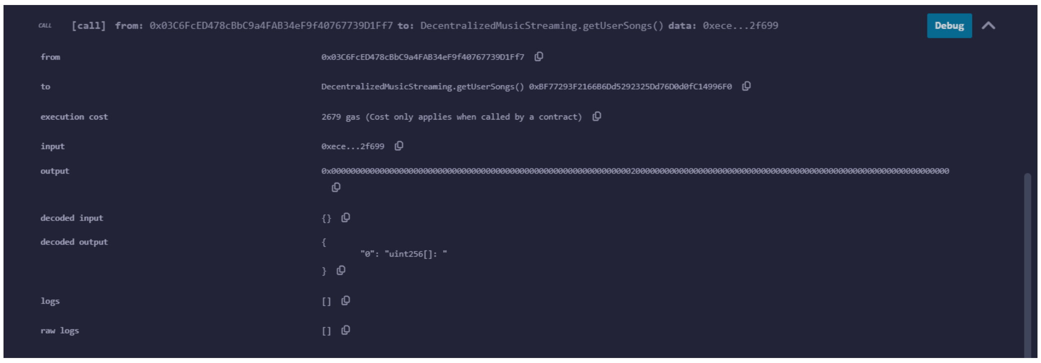
ROLL NO: 6961



```
from              0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
to                DecentralizedMusicStreaming.getSongDetails(uint256) 0xBF77293F2166B6Dd5292325Dd76D0d0fC14996F0
execution cost    30588 gas (Cost only applies when called by a contract)
input             0x778...00001
output            0x00000000000000000000000000000000000000000000000000000000000000a000000000000000000000000000000000000000000000000000000000000001000000
                  00000000000000000000000000000000000000000000000000000000000a000000000000000000000000000000000000000000000000000000000000001600000000000
                  0000000000000003c6fced478cbbc9a4fab34ef9f40767739d1ff7000000000000000000000000000000000000000000000000000000000000002431323822d416b6820
                  4b6173686e69202d2041747461636b20313238204b6270732e6d703300000000000000000000000000000000000000000000000000000000000000000000000000000000
                  00000000000000000000000000000000000000002f4e616b756c2043687567682c20536861736877617420536163686465762c20616e64205375726268692059616461
                  7600000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000003b6261667962656964656634626834676d71
                  64377278787032687a33636a64713334676e7a676534376d7571776d656568353468786265796d6c796d0000000000
decoded input     {
                          "uint256 _songId": "1"
                  }
decoded output    {
                          "0": "string: 128-Akh Kashni - Attack 128 Kbps.mp3",
                          "1": "string: Nakul Chugh, Shashwat Sachdev, and Surbhi Yadav",
                          "2": "uint256: 10",
                          "3": "string: bafybeidef4bh4gmqd7rxxp2hz3cjdq34gnzge47muqwmeeh54hxbeymlym",
                          "4": "address: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7"
                  }
logs              []
```

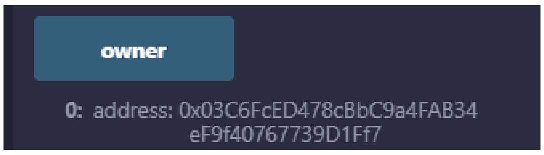
3. Get User Song

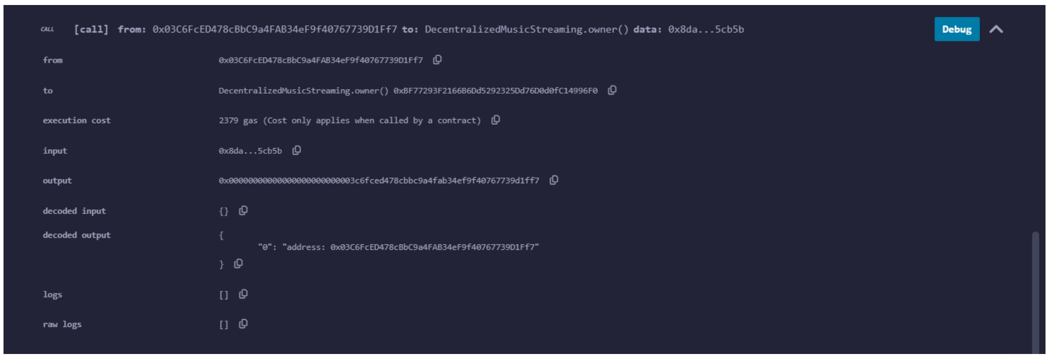




```
CALL  [call] from: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7 to: DecentralizedMusicStreaming.getUserSongs() data: 0xece...2f699
from              0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
to                DecentralizedMusicStreaming.getUserSongs() 0xBF77293F2166B6Dd5292325Dd76D0d0fC14996F0
execution cost    2679 gas (Cost only applies when called by a contract)
input             0xece...2f699
output            0x00000000000000000000000000000000000000000000000000000000000000200000000000000000000000000000000000000000000000000000000000000000
decoded input     {}
decoded output    {
                          "0": "uint256[]: "
                  }
logs              []
raw logs          []
```

4. Owner



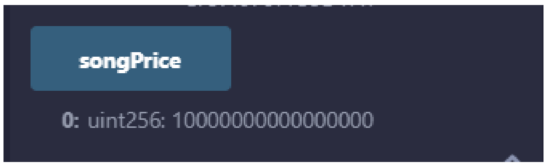


```
CALL  [call] from: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7 to: DecentralizedMusicStreaming.owner() data: 0x8da...5cb5b
from              0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
to                DecentralizedMusicStreaming.owner() 0xBF77293F2166B6Dd5292325Dd76D0d0fC14996F0
execution cost    2379 gas (Cost only applies when called by a contract)
input             0x8da...5cb5b
output            0x00000000000000000000000003c6fced478cbbc9a4fab34ef9f40767739d1ff7
decoded input     {}
decoded output    {
                          "0": "address: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7"
                  }
logs              []
raw logs          []
```

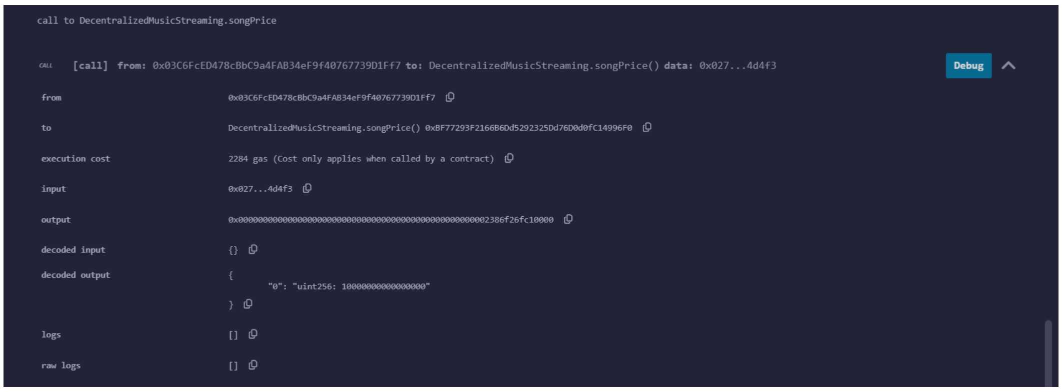
NAME: SHARON PHILIP

ROLL NO: 6961

5. Song Price

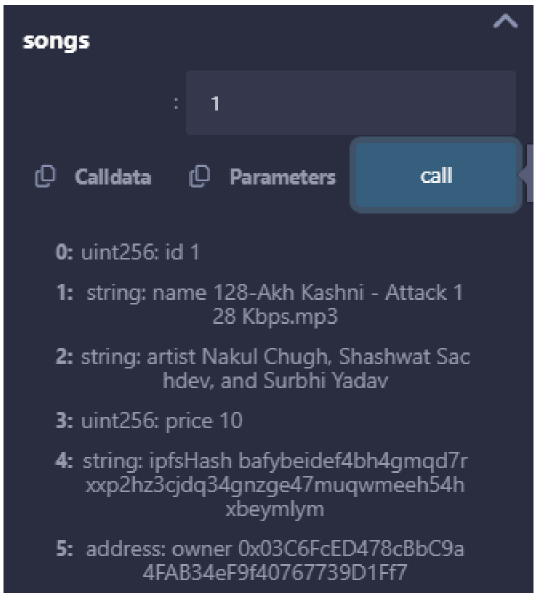


```
0: uint256: 10000000000000000
```

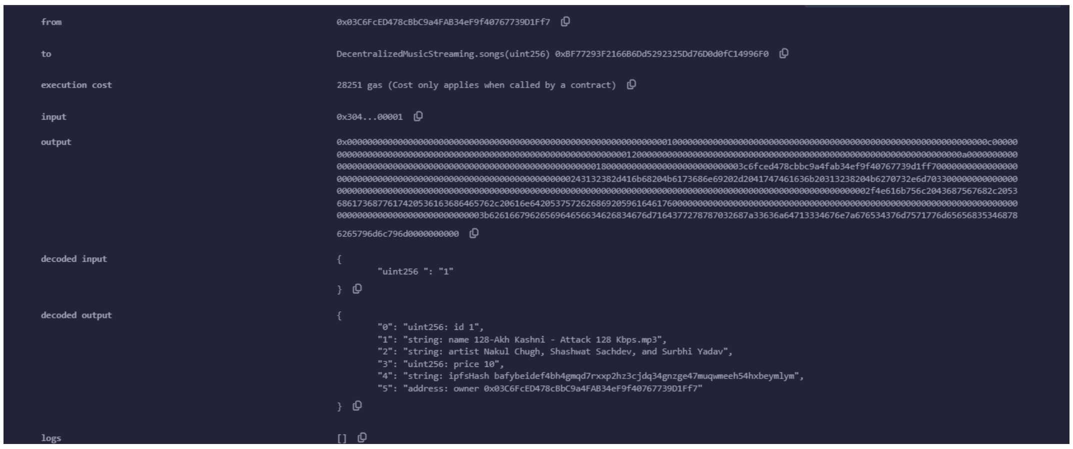


```
call to DecentralizedMusicStreaming.songPrice

CALL  [call] from: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7 to: DecentralizedMusicStreaming.songPrice() data: 0x027...4d4f3

from              0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
to                DecentralizedMusicStreaming.songPrice() 0xBF77293F2166B6Dd5292325Dd76D0d0fC14996F0
execution cost    2284 gas (Cost only applies when called by a contract)
input             0x027...4d4f3
output            0x0000000000000000000000000000000000000000000000000000002386f26fc10000
decoded input     {}
decoded output    {
                        "0": "uint256: 10000000000000000"
                  }
logs              []
raw logs          []
```

6. Song



```
: 1

0: uint256: id 1
1: string: name 128-Akh Kashni - Attack 128 Kbps.mp3
2: string: artist Nakul Chugh, Shashwat Sachdev, and Surbhi Yadav
3: uint256: price 10
4: string: ipfsHash bafybeidef4bh4gmqd7rxxp2hz3cjdq34gnzge47muqwmeeh54hxbeymlym
5: address: owner 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
```

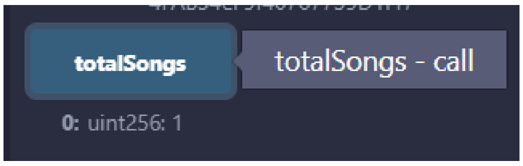


```
from              0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
to                DecentralizedMusicStreaming.songs(uint256) 0xBF77293F2166B6Dd5292325Dd76D0d0fC14996F0
execution cost    28251 gas (Cost only applies when called by a contract)
input             0x304...00001
output            0x00000000000000000000000000000000000000000000000000000000000000010000000000000000000000000000000000000000000000000000000000000000c00000
                  00000000000000000000000000000000000000000000000000000000120000000000000000000000000000000000000000000000000000000000000000a0000000000
                  000000000000000000000000000000000000000000000000000180000000000000000000000003c6fced478cbbc9a4fab34ef9f40767739d1ff70000000000000000
                  00000000000000000000000000000000000000000000002431323832d416b68204b6173686e69202d2041747461636b20313238204b6270732e6d70330000000000000
                  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000002f4e616b756c2043687567682c2053
                  686173687761742053616368646576 2c20616e64205375726268692059616461760000000000000000000000000000000000000000000000000000000000000000000
                  00000000000000000000000000003b6261667962656964656634626834676d7164377278787032687a33636a6471333467 6e7a676534376d7571776d65656835346878
                  6265796d6c796d0000000000
decoded input     {
                        "uint256 ": "1"
                  }
decoded output    {
                        "0": "uint256: id 1",
                        "1": "string: name 128-Akh Kashni - Attack 128 Kbps.mp3",
                        "2": "string: artist Nakul Chugh, Shashwat Sachdev, and Surbhi Yadav",
                        "3": "uint256: price 10",
                        "4": "string: ipfsHash bafybeidef4bh4gmqd7rxxp2hz3cjdq34gnzge47muqwmeeh54hxbeymlym",
                        "5": "address: owner 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7"
                  }
logs              []
```

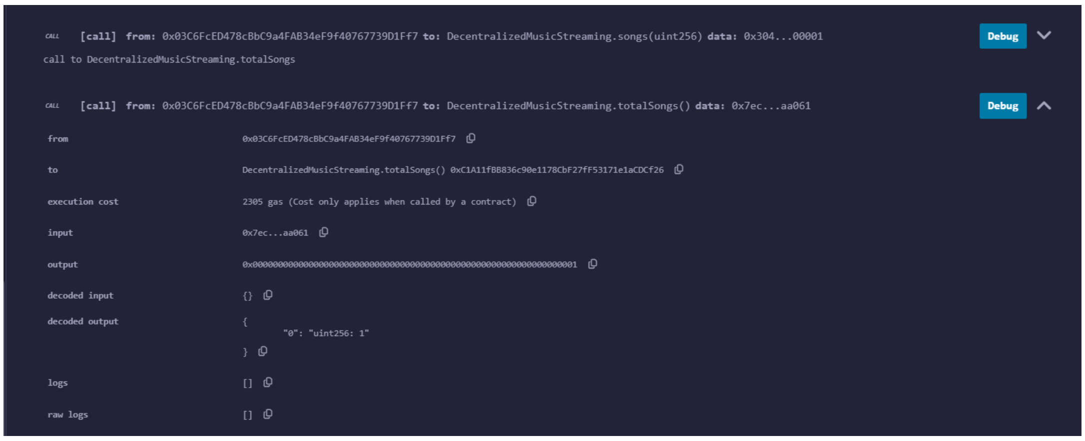
NAME: SHARON PHILIP

ROLL NO: 6961

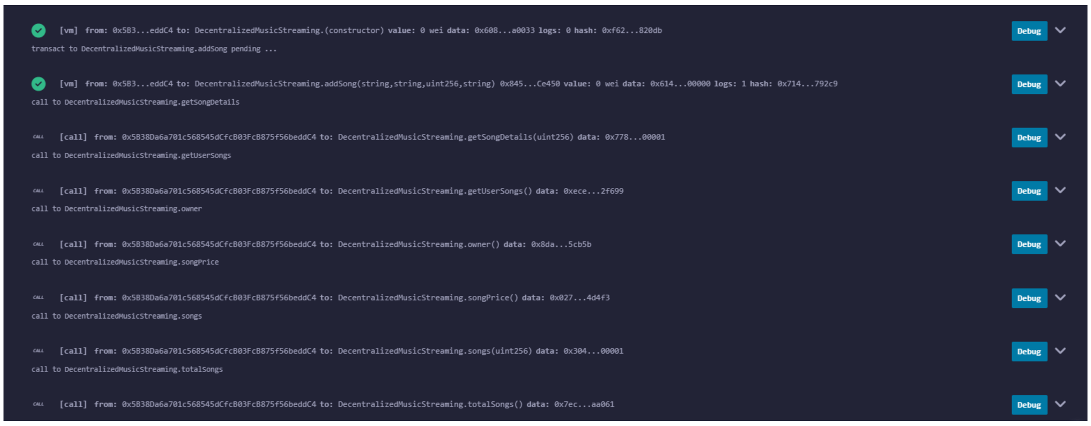
7. Total Song



```
0: uint256: 1
```



```
CALL  [call] from: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7 to: DecentralizedMusicStreaming.songs(uint256) data: 0x304...00001
call to DecentralizedMusicStreaming.totalSongs

CALL  [call] from: 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7 to: DecentralizedMusicStreaming.totalSongs() data: 0x7ec...aa061
from                 0x03C6FcED478cBbC9a4FAB34eF9f40767739D1Ff7
to                   DecentralizedMusicStreaming.totalSongs() 0xC1A11fBB836c90e1178CbF27fF53171e1aCDCf26
execution cost       2305 gas (Cost only applies when called by a contract)
input                0x7ec...aa061
output               0x0000000000000000000000000000000000000000000000000000000000000001
decoded input        {}
decoded output       {
                         "0": "uint256: 1"
                     }
logs                 []
raw logs             []
```



```
[vm] from: 0x5B3...eddC4 to: DecentralizedMusicStreaming.(constructor) value: 0 wei data: 0x608...a0033 logs: 0 hash: 0xf62...820db
transact to DecentralizedMusicStreaming.addSong pending ...
[vm] from: 0x5B3...eddC4 to: DecentralizedMusicStreaming.addSong(string,string,uint256,string) 0x845...Ce450 value: 0 wei data: 0x614...00000 logs: 1 hash: 0x714...792c9
call to DecentralizedMusicStreaming.getSongDetails
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.getSongDetails(uint256) data: 0x778...00001
call to DecentralizedMusicStreaming.getUserSongs
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.getUserSongs() data: 0xece...2f699
call to DecentralizedMusicStreaming.owner
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.owner() data: 0x8da...5cb5b
call to DecentralizedMusicStreaming.songPrice
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.songPrice() data: 0x027...4d4f3
call to DecentralizedMusicStreaming.songs
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.songs(uint256) data: 0x304...00001
call to DecentralizedMusicStreaming.totalSongs
CALL  [call] from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: DecentralizedMusicStreaming.totalSongs() data: 0x7ec...aa061
```