MOVIE TICKETING SYSTEM TOKEN

Yashwanth reddy, katam , ykatam@buffalo.edu , UB 50361872

Sri Varaha Surya Vamsi , Saripudi , srivarah@buffalo.edu , UB 50357055

Defining parameters –

1. **Symbol :** MTST

   Reason –

   It is an acronym of idea also it is easy to remember our token this way in a brief manner .

2. **Total supply :** 10000

   Reason –

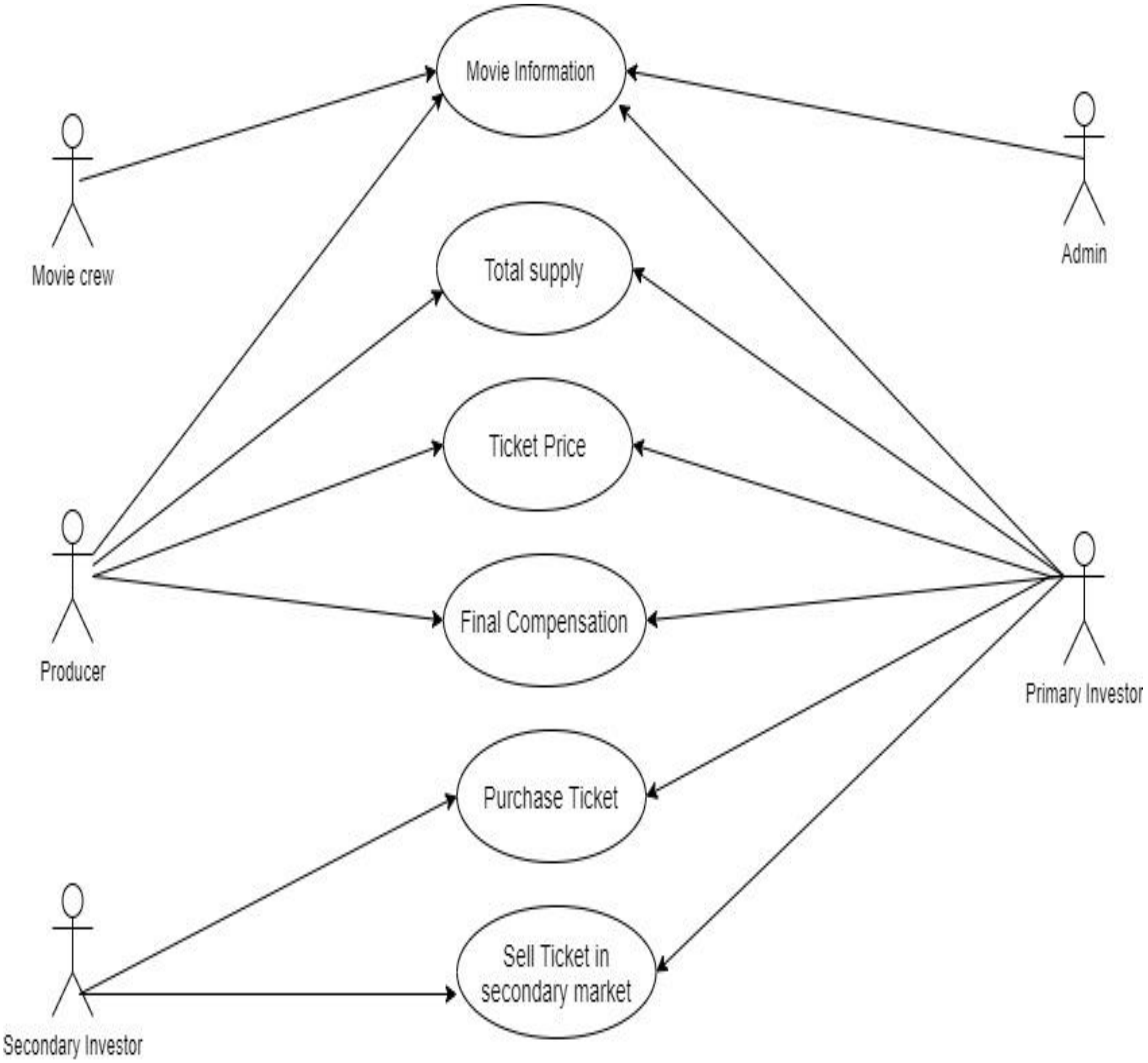
   This is kind of pilot project that we want to implement on a smaller scale first. But we are very optimistic that it can be scaled to large scale systems.

3. **Number of decimal places :** 18

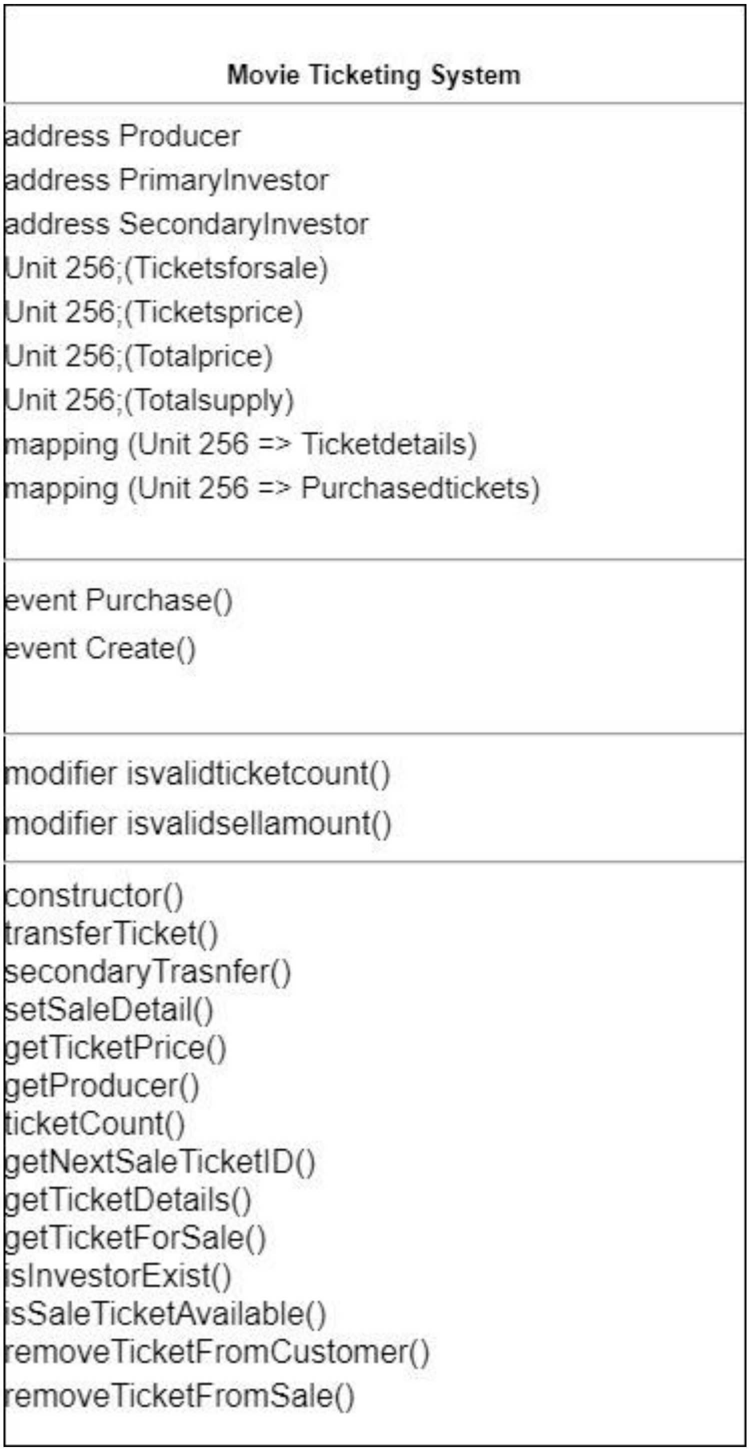
   Reason –

   It was a deliberate decision to keep the decimal places as they are available. The well thought off reason is since movie production is a costly affair, and to make it with in the reach of a small scale investor who has interest in such business avenues.

Use Case Diagram –



Contract Diagram -



Movie Ticketing System

address Producer
address PrimaryInvestor
address SecondaryInvestor
Unit 256;(Ticketsforsale)
Unit 256;(Ticketsprice)
Unit 256;(Totalprice)
Unit 256;(Totalsupply)
mapping (Unit 256 => Ticketdetails)
mapping (Unit 256 => Purchasedtickets)

event Purchase()
event Create()

modifier isvalidticketcount()
modifier isvalidsellamount()

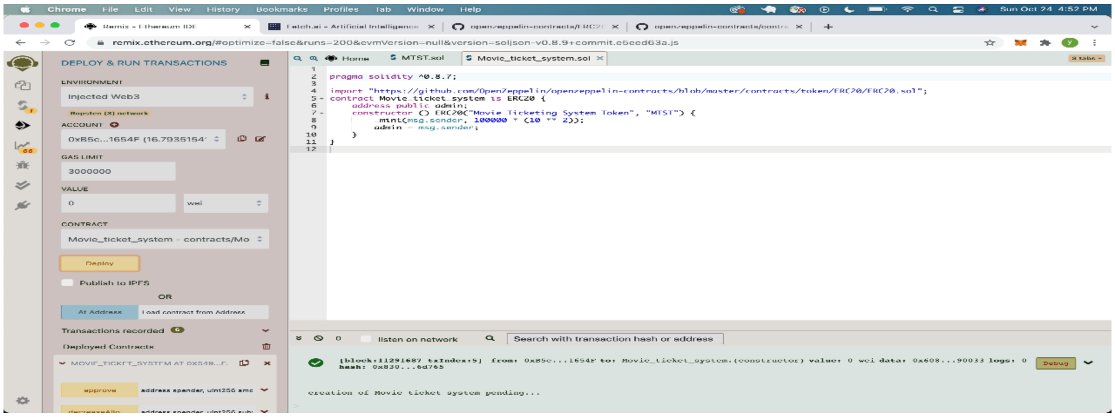
constructor()
transferTicket()
secondaryTrasnfer()
setSaleDetail()
getTicketPrice()
getProducer()
ticketCount()
getNextSaleTicketID()
getTicketDetails()
getTicketForSale()
isInvestorExist()
isSaleTicketAvailable()
removeTicketFromCustomer()
removeTicketFromSale()

Deployment and testing instructions –

1. Compile the **.sol** file in the remix.
2. First test in any inbuilt virtual environment , for example London/ berlin VM etc. The deployment should go through successfully.
3. We can see the gas fee deduction from test account. This implies successful deployment.
4. Connect the meta mask to the remix website, and ensure you have some test ether in your account, otherwise one can obtain it from ropsten faucet.
5. The deployment environment should be changed to Injected web3, the metamask address will be automatically updated in account address.
6. On tapping deploy button, a confirmation message comes to the metamask account where one has to accept/ reject the contract.
7. On accepting it, it gets deployed on to ropsten faucet, and one can see the transaction details.

Screenshots (Proof of deployment) –

1.



Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Oct 24 4:52 PM
Remix - Ethereum IDE
Fetch.ai - Artificial Intelligence
openzeppelin-contracts/ERC20
openzeppelin-contracts/contra
remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.9+commit.e5eed63a.js
DEPLOY & RUN TRANSACTIONS
Home
MTST.sol
Movie_ticket_system.sol
Deploy
At Address
Load contract from Address
approve
decreaseAllo
Debug

ENVIRONMENT: Injected Web3
Ropsten (3) network
ACCOUNT: 0xB5c...1654F (16.7935154
GAS LIMIT: 3000000
VALUE: 0 wei
CONTRACT: Movie_ticket_system - contracts/Mo
□ Publish to IPFS
OR
Transactions recorded 6
Deployed Contracts
MOVIE_TICKET_SYSTEM AT 0X548...F.
address spender, uint256 amo
address spender, uint256 sub

```
pragma solidity ^0.8.7;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20.sol";
contract Movie_ticket_system is ERC20 {
    address public admin;
    constructor () ERC20("Movie Ticketing System Token", "MTST") {
        _mint(msg.sender, 100000 * (10 ** 2));
        admin = msg.sender;
    }
}
```

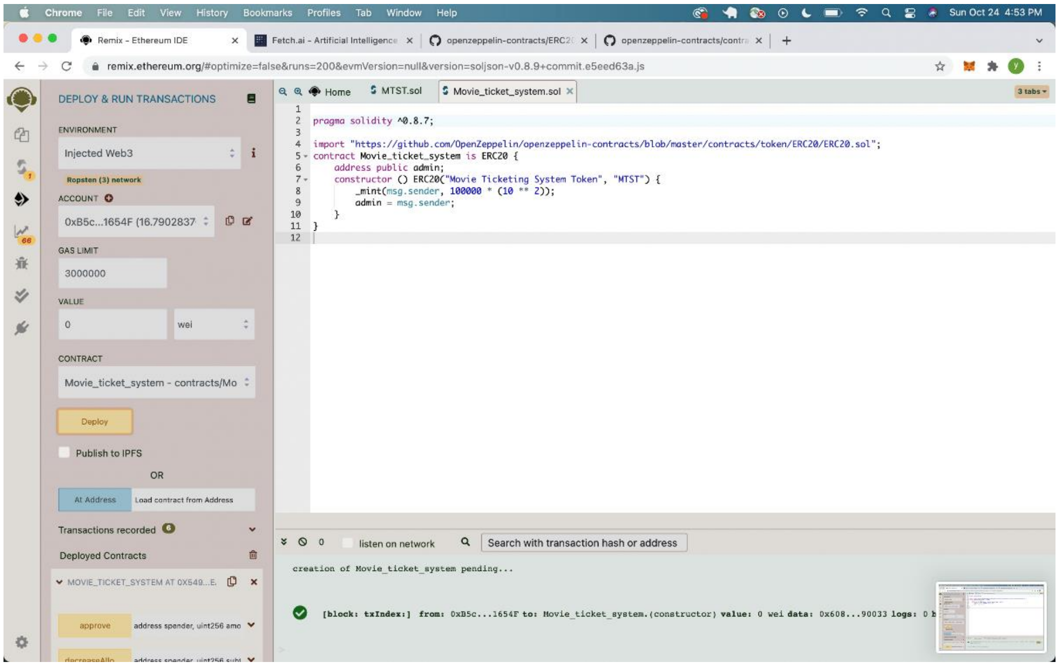
```
0   listen on network   Search with transaction hash or address

[block:11291687 txIndex:5] from: 0xB5c...1654F to: Movie_ticket_system.(constructor) value: 0 wei data: 0x608...90033 logs: 0
hash: 0x830...6d765

creation of Movie ticket system pending...
```

This screen shot is the immediate one post tapping the deployment tab after the compilation of the program. We can observe the **execution of Movie ticket_system pending…**

2.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Oct 24 4:53 PM
Remix - Ethereum IDE
Fetch.ai - Artificial Intelligence
openzeppelin-contracts/ERC2
openzeppelin-contracts/contra
remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.9+commit.e5eed63a.js
Home
MTST.sol
Movie_ticket_system.sol
3 tabs
Deploy
At Address
Load contract from Address
approve
decreaseAllo

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3

Ropsten (3) network

ACCOUNT: 0xB5c...1654F (16.7902837

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Movie_ticket_system - contracts/Mo

□ Publish to IPFS

Transactions recorded 6

Deployed Contracts

MOVIE_TICKET_SYSTEM AT 0X549...E.

approve address spender, uint256 amo

decreaseAllo address spender, uint256 subt

```

pragma solidity ^0.8.7;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/ERC20/ERC20.sol";
contract Movie_ticket_system is ERC20 {
    address public admin;
    constructor () ERC20("Movie Ticketing System Token", "MTST") {
        _mint(msg.sender, 100000 * (10 ** 2));
        admin = msg.sender;
    }
}

```

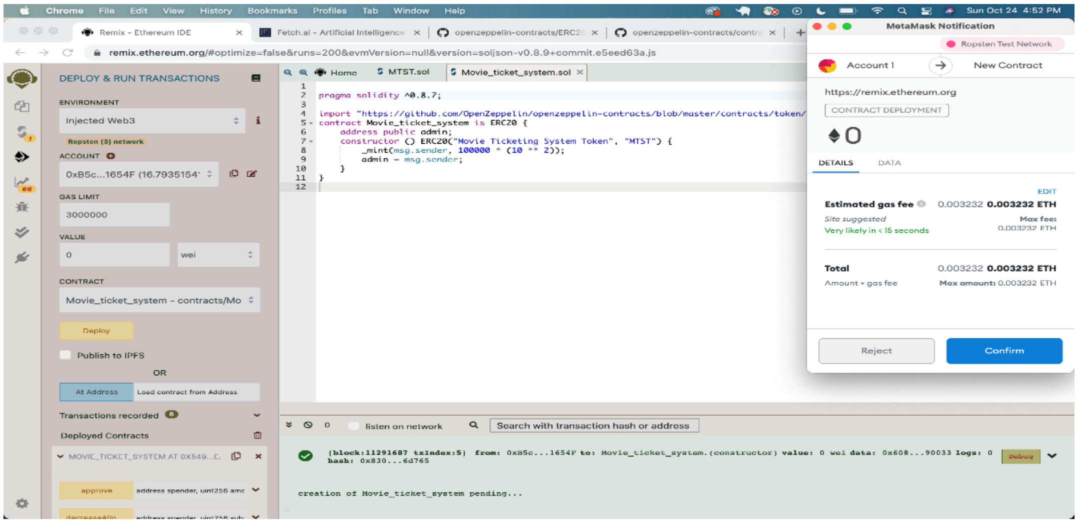
0 □ listen on network Search with transaction hash or address

```
creation of Movie_ticket_system pending...

[block: txIndex:] from: 0xB5c...1654F to: Movie_ticket_system.(constructor) value: 0 wei data: 0x608...90033 logs: 0 h
```

Success Message can be observed in the screen shot above.

3.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Sun Oct 24 4:52 PM
Remix - Ethereum IDE
Fetch.ai - Artificial Intelligence
openzeppelin-contracts/ERC2
openzeppelin-contracts/contra
remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.9+commit.e5eed63a.js
Home
MTST.sol
Movie_ticket_system.sol
Deploy
At Address
Load contract from Address
approve
decreaseAllo
Debug
DETAILS
DATA
EDIT
Reject
Confirm

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3

Ropsten (3) network

ACCOUNT: 0xB5c...1654F (16.7935154

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Movie_ticket_system - contracts/Mo

□ Publish to IPFS

Transactions recorded 0

Deployed Contracts

MOVIE_TICKET_SYSTEM AT 0X549...E.

approve address spender, uint256 amc

decreaseAllo address spender, uint256 subt

```

pragma solidity ^0.8.7;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/token/
contract Movie_ticket_system is ERC20 {
    address public admin;
    constructor () ERC20("Movie Ticketing System Token", "MTST") {
        _mint(msg.sender, 100000 * (10 ** 2));
        admin = msg.sender;
    }
}

```

MetaMask Notification

Ropsten Test Network

Account 1 → New Contract

https://remix.ethereum.org

CONTRACT DEPLOYMENT

0

Estimated gas fee 0.003232 **0.003232 ETH**

Site suggested

Very likely in < 15 seconds

Max fee: 0.003232 ETH

Total 0.003232 **0.003232 ETH**

Amount + gas fee

Max amount: 0.003232 ETH

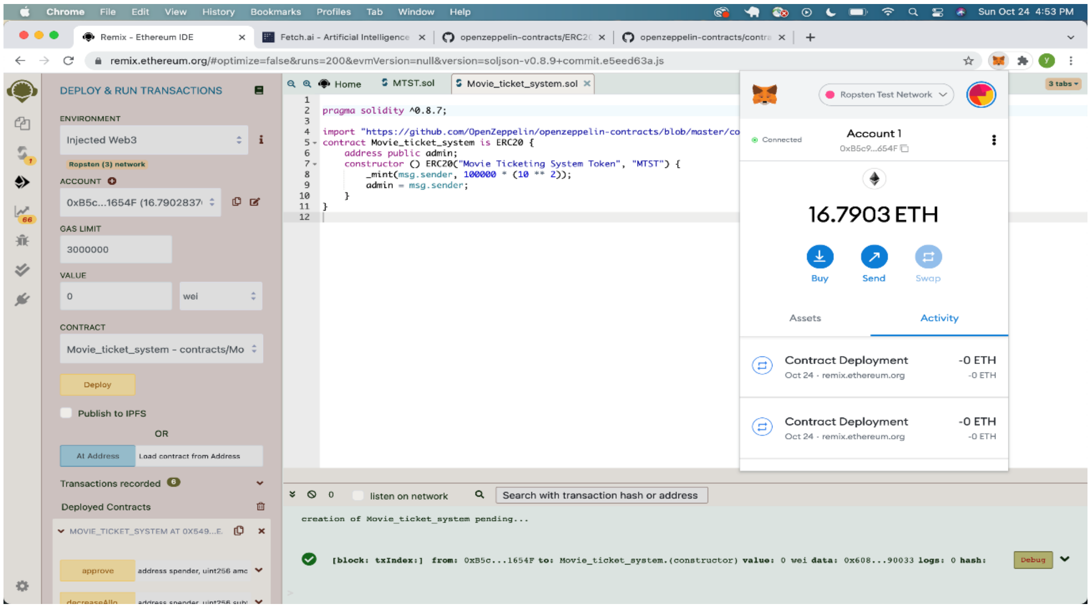
0 □ listen on network Search with transaction hash or address

```
[block:11291687 txIndex:5] from: 0xB5c...1654F to: Movie_ticket_system.(constructor) value: 0 wei data: 0x608...90033 logs: 0
hash: 0x830...6d765

creation of Movie_ticket_system pending...
```

New contract confirmation popup in my meta mask Ropsten network

4.



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.9+commit.e5eed63a.js

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Web3 (Ropsten (3) network)

ACCOUNT: 0xB5c...1654F (16.7902837

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Movie_ticket_system - contracts/Mo

Movie_ticket_system.sol

```
pragma solidity ^0.8.7;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/co
contract Movie_ticket_system is ERC20 {
    address public admin;
    constructor () ERC20("Movie Ticketing System Token", "MTST") {
        _mint(msg.sender, 100000 * (10 ** 2));
        admin = msg.sender;
    }
}
```

Ropsten Test Network

Account 1 0xB5c9...654F

16.7903 ETH

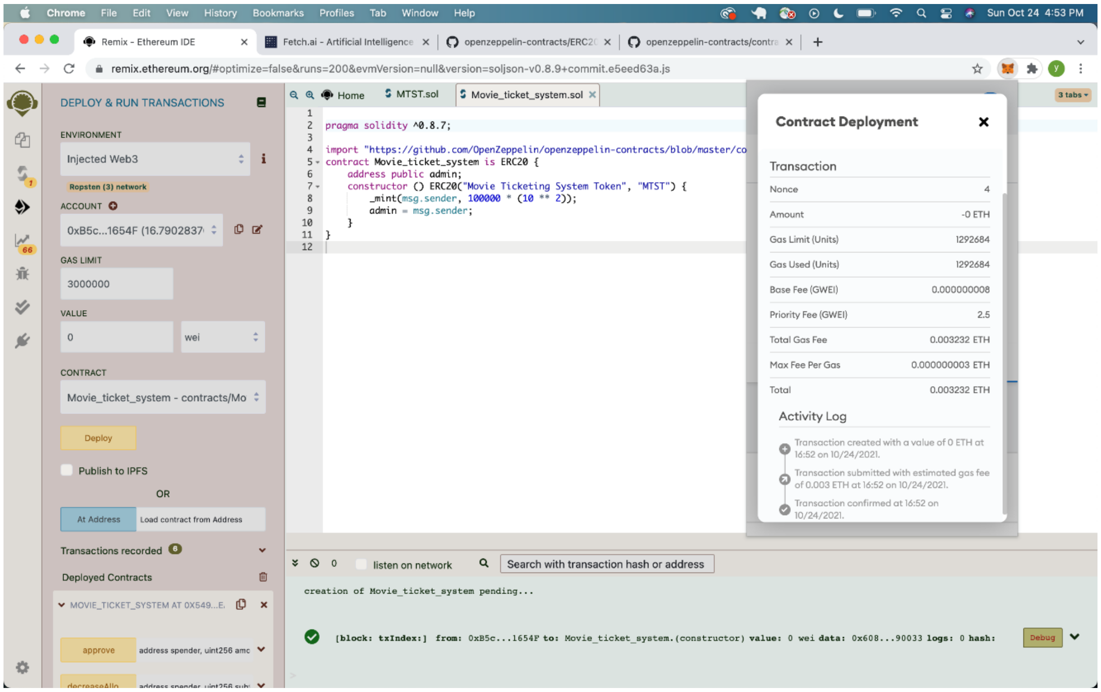
Contract Deployment Oct 24 · remix.ethereum.org	-0 ETH -0 ETH
Contract Deployment Oct 24 · remix.ethereum.org	-0 ETH -0 ETH

```
creation of Movie_ticket_system pending...

[block: txIndex:] from: 0xB5c...1654F to: Movie_ticket_system.(constructor) value: 0 wei data: 0x608...90033 logs: 0 hash:
```

Contract Deployment Complete status

5.



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.9+commit.e5eed63a.js

ENVIRONMENT: Injected Web3 (Ropsten (3) network)

ACCOUNT: 0xB5c...1654F (16.7902837

GAS LIMIT: 3000000

VALUE: 0 wei

CONTRACT: Movie_ticket_system - contracts/Mo

```
pragma solidity ^0.8.7;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/co
contract Movie_ticket_system is ERC20 {
    address public admin;
    constructor () ERC20("Movie Ticketing System Token", "MTST") {
        _mint(msg.sender, 100000 * (10 ** 2));
        admin = msg.sender;
    }
}
```

```
creation of Movie_ticket_system pending...

[block: txIndex:] from: 0xB5c...1654F to: Movie_ticket_system.(constructor) value: 0 wei data: 0x608...90033 logs: 0 hash:
```

Contract Deployment

Transaction

Nonce	4
Amount	-0 ETH
Gas Limit (Units)	1292684
Gas Used (Units)	1292684
Base Fee (GWEI)	0.000000008
Priority Fee (GWEI)	2.5
Total Gas Fee	0.003232 ETH
Max Fee Per Gas	0.000000003 ETH
Total	0.003232 ETH

Activity Log

- Transaction created with a value of 0 ETH at 16:52 on 10/24/2021.
- Transaction submitted with estimated gas fee of 0.003 ETH at 16:52 on 10/24/2021.
- Transaction confirmed at 16:52 on 10/24/2021.

Contract Deployment details