



School: ..... Campus: .....

Academic Year: ..... Subject Name: ..... Subject Code: .....

Semester: ..... Program: ..... Branch: ..... Specialization: .....

Date: .....

## Applied and Action Learning

(Learning by Doing and Discovery)

**Name of the Experiment : Token Launch – Deploying a Token Locally**

**Coding Phase : Pseudo Code/Flow Chart/Algorithm**

- Open Remix IDE.
- Import the ERC20 contract from OpenZeppelin.
- Create a constructor that sets the token name, symbol, and mints initial supply.
- Compile the contract with Solidity ^0.8.20.
- Deploy to local blockchain using account 1.
- Use functions `name()`, `symbol()`, and `totalSupply()` to verify token.

**Apparatus/Software Used:**

- OS: Windows or others.
- Remix IDE.
- Wallet: MetaMask.
- Library: OpenZeppelin ERC20

**Testing Phase:**

- Call `name()` → returns **"GuduTokens"**
- Call `symbol()` → returns **"GUD"**
- Call `totalSupply()` → returns **100000000**
- Call `balanceOf(owner)` → shows total supply in deployer's account.

## Implementation Phase: Final Output (no error)

Step 1: Open Remix IDE.

- Open Browser/Brave.
- Search Remix IDE.

Step 2: Write Smart Contract.

- Create a new file in inside of contract using .sol.
- Write code:

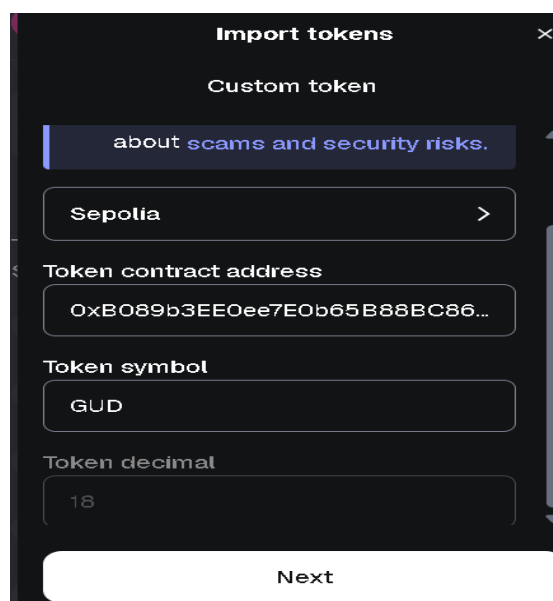
```
1 // SPDX-License-Identifier: MIT
2
3
4 pragma solidity ^0.8.0;
5
6 import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
7
8 contract MyToken is ERC20 {
9
10     constructor(string memory name,string memory symbol) infinite gas 710800 gas
11     ERC20(name,symbol){
12         _mint(msg.sender,1000000 * 10 ** decimals());
13     }
14 }
```

Step 3: Code Compile and Deploy .

- Click the Solidity compiler and compile this file.
- Click Deploy & run transaction Option .
- Deploy the write of some string name and symbol.

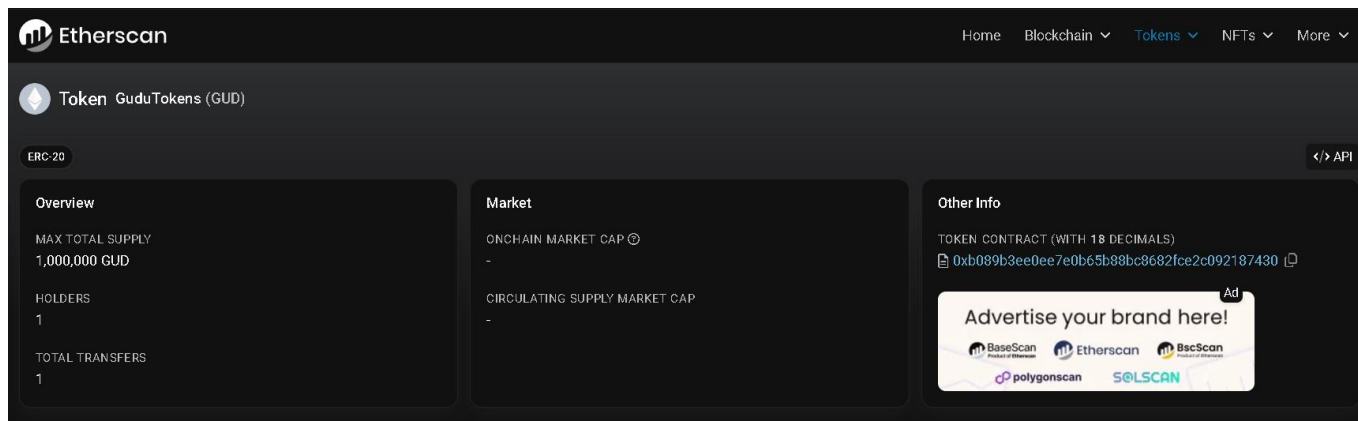
Step 4: Add Token to MetaMask.

- In MetaMask, click "**Import Tokens**".
- Enter your deployed contract address.
- Your token should appear in the wallet.



## Step 5: Check Token Details.

- `name()` → should return "GuduTokens"
- `symbol()` → should return "GUD"
- `totalSupply()` → should return 10000000.

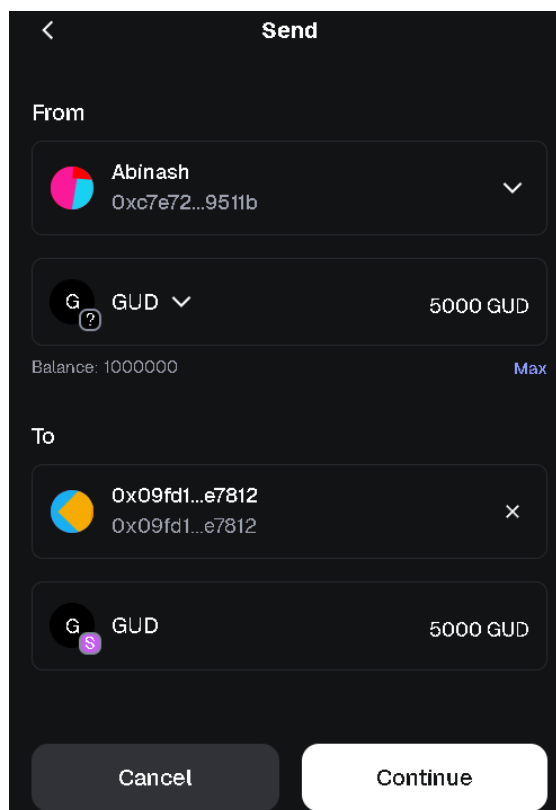


The screenshot shows the Etherscan interface for the GuduTokens (GUD) token. The page is divided into three main sections: Overview, Market, and Other Info.

- Overview:** Displays the token's contract type as ERC-20. It shows a maximum total supply of 1,000,000 GUD, 1 holder, and 1 total transfer.
- Market:** Shows the on-chain market cap and the circulating supply market cap, both currently at 0.
- Other Info:** Provides the token contract address (0xb089b3ee0ee7e0b65b88bc8682fce2c092187430) and includes an advertisement for brand promotion.

## Step 6: Transfer Token:

- Use `transfer(receiver_address, amount)` to send tokens to another account.
- Then check `balanceOf(receiver_address)` to confirm transfer.



The screenshot displays the 'Send' interface for transferring GUD tokens. It includes a 'From' section with the sender's address (Abinash, 0xc7e72...9511b) and a 'To' section with the recipient's address (0x09fd1...e7812). The transfer amount is set to 5000 GUD, and the sender's current balance is 1000000 GUD. The interface also features a 'Cancel' button and a 'Continue' button.

## Observations

- The OpenZeppelin library makes ERC20 token creation quick and secure.
- Local deployment requires no real ETH.
- Token functions behave exactly the same as on testnet/mainnet.

### ASSESSMENT

| Rubrics  | Full Mark | Marks Obtained | Remarks |
|--|-----------|----------------|---------|
| Concept  | 10        |                |         |
| Planning and Execution/<br>Practical Simulation/ Programming | 10        |                |         |
| Result and Interpretation                                    | 10        |                |         |
| Record of Applied and Action Learning                        | 10        |                |         |
| Viva   | 10        |                |         |
| <b>Total</b>   | <b>50</b> |                |         |

**Signature of the Student:**

Name :

Regn. No. :

Page No.....

**Signature of the Faculty:**

*\* As applicable according to the experiment.  
Two sheets per experiment (10-20) to be used.*

