

ETE
A Project Review Report

on

CRYPTO CURRENCY

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING



[Established under Galgotias University Uttar Pradesh Act No. 14 of 2011]

**Under The Supervision of
Ms.J. Angelin Blessy
Project Guide**

Submitted By

YASH RAJ 19SCSE1010780
SHIVAM SINGH 19SCSE1010270

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA,
December 2021**



**SCHOOL OF COMPUTING SCIENCE AND
ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA**

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled "CRYPTO CURRENCY" in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September,2021 to December, 2021, under the supervision of Ms.J. Angelin Blessy, Guide Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in project has not been submitted by us for the award of any other degree of this or any other places.

Yash Raj 19SCSE1010780

Shivam Singh 19SCSE1010270

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Ms.J.Angelin Blessy

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination Yash Raj 19SCSE1010780 and Shivam Singh 19SCSE1010270 has been held on CRYPTO CURRENCY and there work is recommended for the award of B.Tech.

Signature of Examiner(s)


Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: November, 2013
Place: Greater Noida

Abstract

A cryptocurrency is a digital currency that is secured by cryptography, which makes it nearly impossible to counterfeit. Many cryptocurrencies are decentralized networks based on blockchain technology. We can make cryptocurrency via creating a coin or token.

Creating a coin

The process of coin creation may take as little as 5 minutes. We just need to copy the code of Bitcoin, add a new variable, or even change the value of something, and that's it - we have created our blockchain and coin. However, we need to understand the code and how to change it, which requires extensive coding skills.

This option is not suitable for easy and fast way to create cryptocurrency free of charge. It requires experience in decentralized technologies or need someone who is willing to take on the role of the technology expert.

Another issue is to maintain, support, and promote the coin, as we have to create the whole logic of blockchain to launch our coin. Hiring a team of professionals to handle the task would save more time, but we would have to pay custom software development services.

Creating a token

This is a more feasible way to become a currency creator. While having complete control over the blockchain may sound like a great idea, this has certain drawbacks like increased development time, significant spending, and much more.

Fork cryptocurrency is created on top of an existing blockchain by utilizing the trust, popularity, and consensus mechanism of the underlying technology. When we build a token on top of a strong blockchain, like Ethereum, our token runs on a secure network that is protected from fraudulent attacks. Token creation is less costly in terms of money and time, as we utilize the existing decentralized architecture and implemented consensus mechanisms.

Creating a coin can be very expensive and requires extensive coding skills. For this project purpose we will be creating tokens.

Contents

Title	Page No.
Candidates Declaration	I
Acknowledgement	II
Abstract	III
Contents	IV
List of Table	V
List of Figures	VI
Acronyms	VII
Chapter 1 Introduction	1
1.1 Introduction	2
1.2 Formulation of Problem	3
1.2.1 Tool and Technology Used	
Chapter 2 Literature Survey/Project Design	5
Chapter 3 Functionality/Working of Project	9
Chapter 4 Results and Discussion	11
Chapter 5 Conclusion and Future Scope	41
5.1 Conclusion	41
5.2 Future Scope	42
Reference	43
Publication/Copyright/Product	45

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
SCSE	School of Computing Science and Engineering
BTC	BITCOIN
ETH	Ethereum
USDT	Tether

List of Table

Table No.	Table Name	Page Number
1	Ethereum, NEO and EOS	3

List of Figures

Figure No.	Name Table	Page Number
1	Ethereum, NEO and EOS	3
2	Ropsten Test Net	7
3	Ropsten Ethers	9
5	Meta Mask	11
6	Publish and Verify Contract	12
7	Adding Token	13

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering
BTC	BITCOIN
ETH	Ethereum
USDT	Tether

CHAPTER 1 INTRODUCTION

The first blockchain-based cryptocurrency was Bitcoin, which still remains the most popular and most valuable. Today, there are thousands of alternate cryptocurrencies with various functions and specifications. Some of these are clones or forks of Bitcoin, while others are new currencies that were built from scratch.

Some of the competing cryptocurrencies spawned by Bitcoin's success, known as "altcoins" include Litecoin, Peercoin, and Namecoin, as well as Ethereum, Cardano, and EOS. By Aug. 2021, the aggregate value of all the cryptocurrencies in existence is over \$1.8 trillion—Bitcoin currently represents approximately 46.5% of the total value.

TOOLS FOR CREATING A TOKEN

There are a few blockchains that provide the means for creating a token. Ethereum, NEO and EOS are the most popular tools and are relatively easy to use.

Ethereum

Ethereum became the first blockchain to offer token creation service. It provides an exceptional level of trust due to its maturity and strong position on the cryptocurrency market. All tokens built on Ethereum use the ERC-20 standard. The documentation is well written and organized, making the development process easier. A token on Ethereum can only be written in Solidity (its own programming language), but with the HTTP API you can create dApps in any language.

NEO

NEO blockchain is aimed at the smart economy and utilizes the NEP-5 standard. Unlike Ethereum, you can use almost any high-level programming language, including C#, Java, Python, and Kotlin to create your own token on top of it. HTTP API is available for interaction with the blockchain.

EOS

EOS tokens use the EOSIO. Token standard and can be created with C++ or any other language that compiles into Web Assembly. The blockchain offers great scalability, a vast number of transactions per second, and cost efficiency due to the lack of a truncation fee. The name of this blockchain often arises when discussing how to create a cryptocurrency.

The table below presents the main aspects of creating a new cryptocurrency with Ethereum, NEO and EOS.

	Ethereum	NEO	EOS
Programming language	Solidity (Ethereum's own programming language)	Almost any high-level programming language, including C#, Java, VB.Net, F#, Kotlin, Python, etc.	C++ and any language that compiles into WebAssembly (WASM)
Token standard	ERC-20	NEP-5	EOSIO.Token
Virtual Machine	Native Virtual Machine	Native Virtual Machine	Native Virtual Machine
Wallets	A lot of options	A lot of options	A lot of options
Hardware wallet	Yes	No	Yes
Consensus	PoW (proof-of-work) before the switch to PoS (proof-of-stake)	dBFT (Delegated Byzantine Fault Tolerance)	DPoS (Delegated proof-of-stake)
Primary sphere of use	Smart contracts	Smart economy, digital identity, digital assets	Smart contracts
Number of transactions per second	15 transactions/second	10,000 transactions/second	3,000+ transactions/second

CHAPTER 2 Reference Publication/ Screen Shots

LITERATURE SURVEY

Creating an ERC20 Token

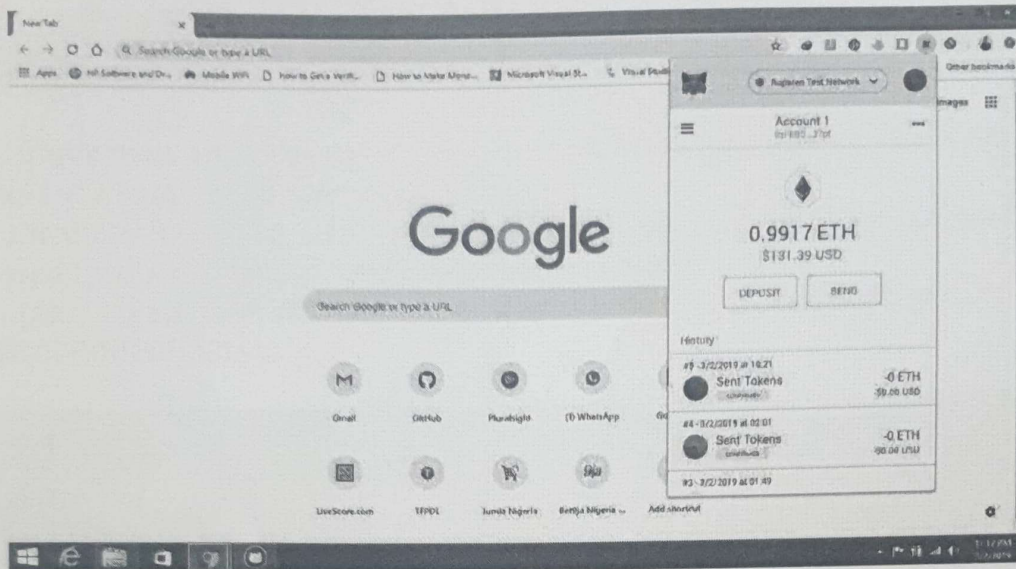
Step 1: Contract code

A token can be created as a smart contract in Ethereum. The complete source code of the contract can be found in <https://www.ethereum.org/token> location.

This code will be edited to build your own token.

Step 2: Create Ethereum wallet with MetaMask

Download MetaMask chrome extension to generate a wallet. This is going to be the owner of the smart contract. Once we download the extension, go ahead and create a new account protected by a password. Then choose "Ropsten TestNet" from the top left corner.



Step 3: Get Ropsten Ethers

We need some test Ethers in our new MetaMask account to finish the creation of the tokens.




Rinkeby Authenticated Faucet

Social network URL containing your Ethereum address...

8 peers 2649088 blocks 9.046256971665328e+56 Ethers 161578 funded

How does this work?

This Ether faucet is running on the Rinkeby network. To prevent malicious actors from exhausting all available funds or accumulating enough Ether to mount long running spam attacks, requests are tied to common 3rd party social network accounts. Anyone having a Twitter, Google+ or Facebook account may request funds within the permitted limits.

-  To request funds via Twitter, make a tweet with your Ethereum address pasted into the contents (surrounding text doesn't matter). Copy-paste the tweets URL into the above input box and fire away!
-  To request funds via Google Plus, publish a new public post with your Ethereum address embedded into the content (surrounding text doesn't matter). Copy-paste the posts URL into the above input box and fire away!
-  To request funds via Facebook, publish a new public post with your Ethereum address embedded into the content (surrounding text doesn't matter). Copy-paste the posts URL into the above input box and fire away!

You can track the current pending requests below the input field to see how much you have to wait until your turn comes:

The faucet is running invisible reCaptcha protection against bots.

Paste Google+ post URL into above text box and press "Give me Ether" button. Our account will be funded with test Ether. We can check your account balance in MetaMask wallet.

Step 4: Edit the contract code

For the decimals and total supply, we can just leave it as it is. 18 decimals mean that a coin can be spitted in 18 parts.

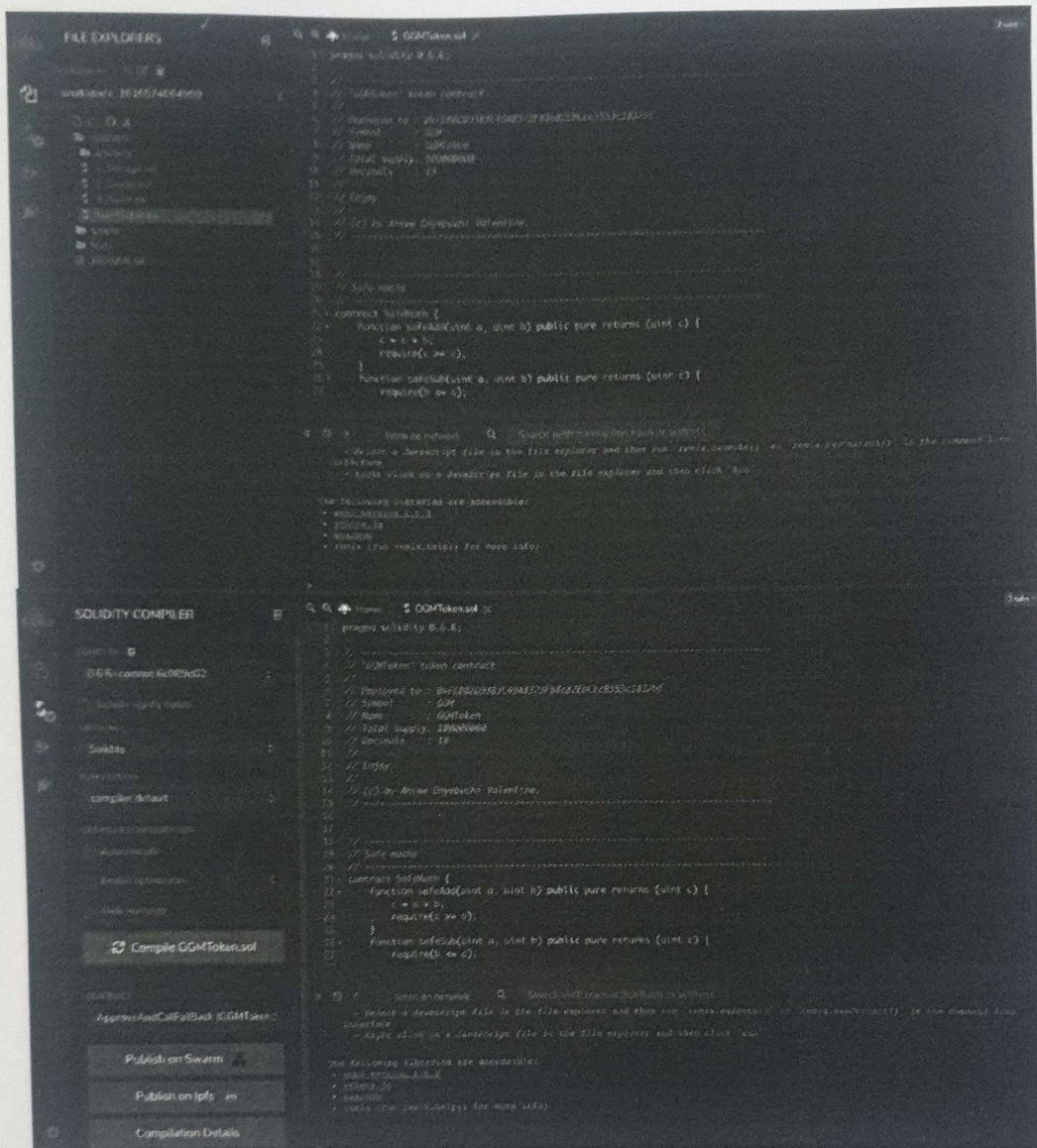
If you want to issue 100 tokens, on the total supply part you have to put 100 followed by the number of decimals that you choose.

Example if I want to emit 100 tokens, what I will put on total supply is:
100000000000000000000;

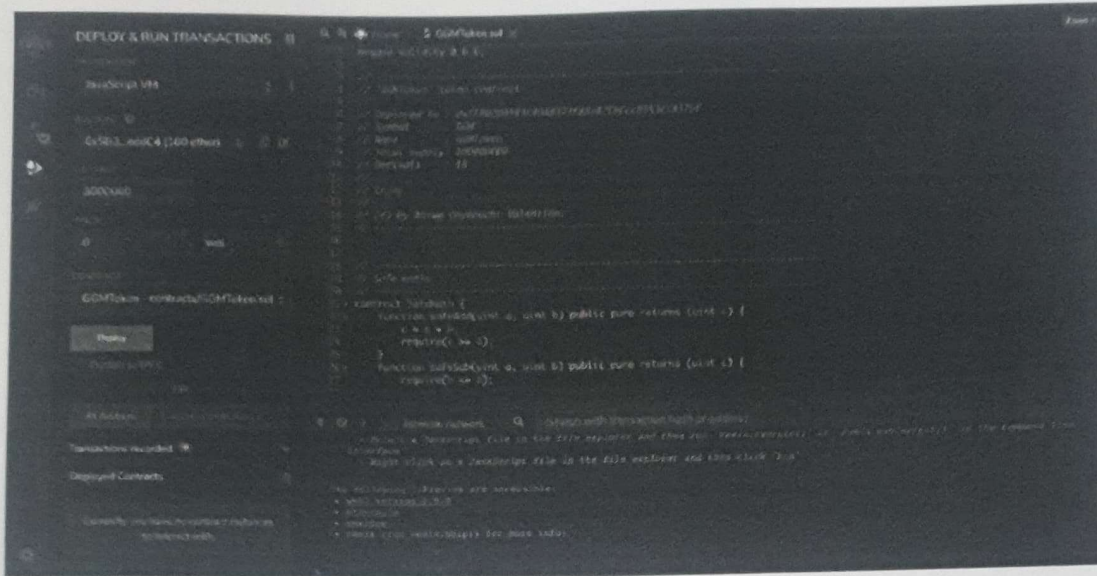
```
1 pragma solidity ^0.5.00;
2
3 // -----
4 // "GSMToken" token contract
5 //
6 // Deployed to : 0xFeB02D93B3C49A8373F8Se82EbCec8553c1837bf
7 // Symbol      : GSM
8 // Name        : GSMToken
9 // Total supply: 100000000
10 // Decimals    : 18
11 //
12 // Enjoy.
13 //
14 // (c) by Abike Dryebuchi Valentine.
15 // -----
```

Step 5: Deploy Contract Code on Remix

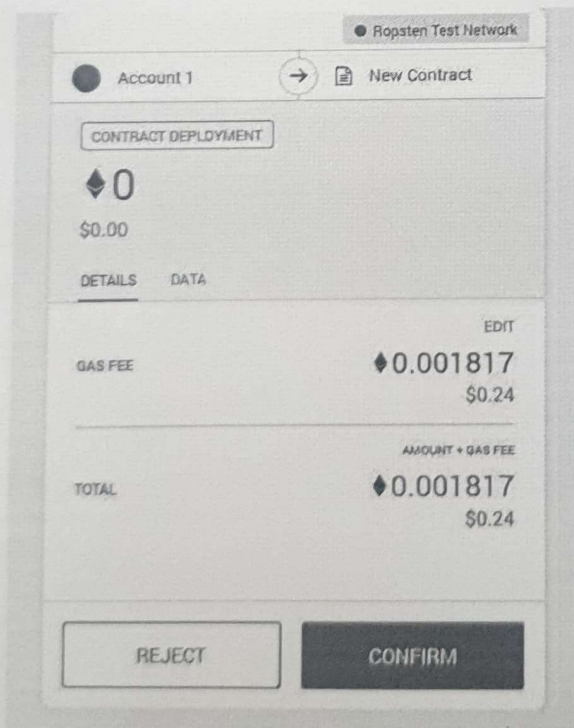
Now we have to head over Remix IDE (an online solidity compiler and debugger) and paste the code we just modified.



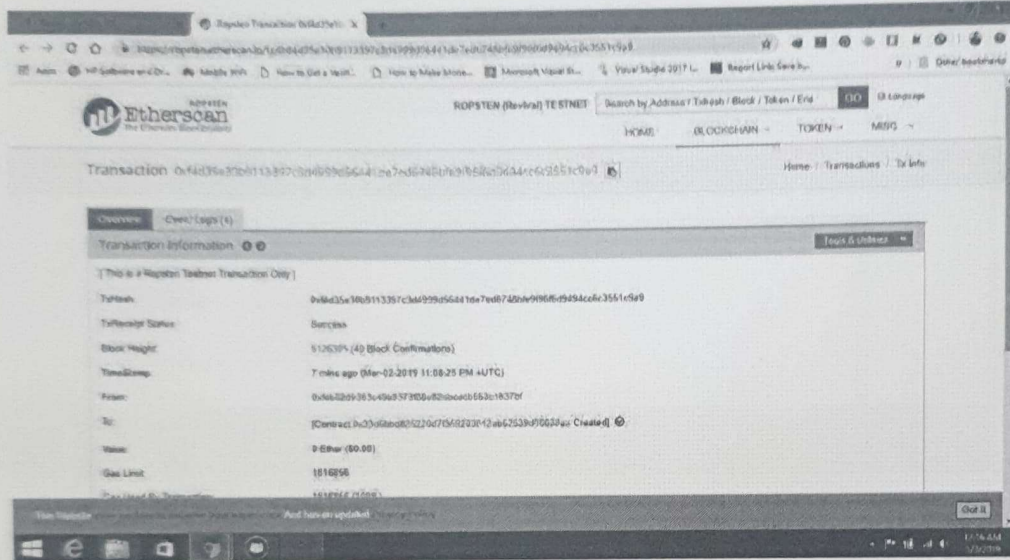
Now go to the next icon below the icon with the checkmark and click on deploy after selecting <your token name> from the drop-down list of contracts.



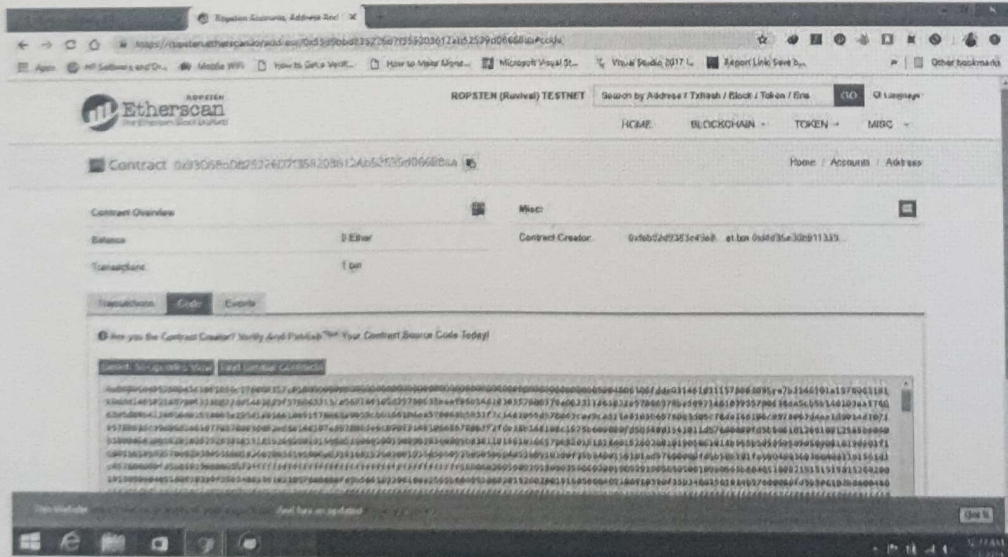
Once we hit deploy, MetaMask will prompt us to buy some test ether and submit the transaction. It looks something like this:



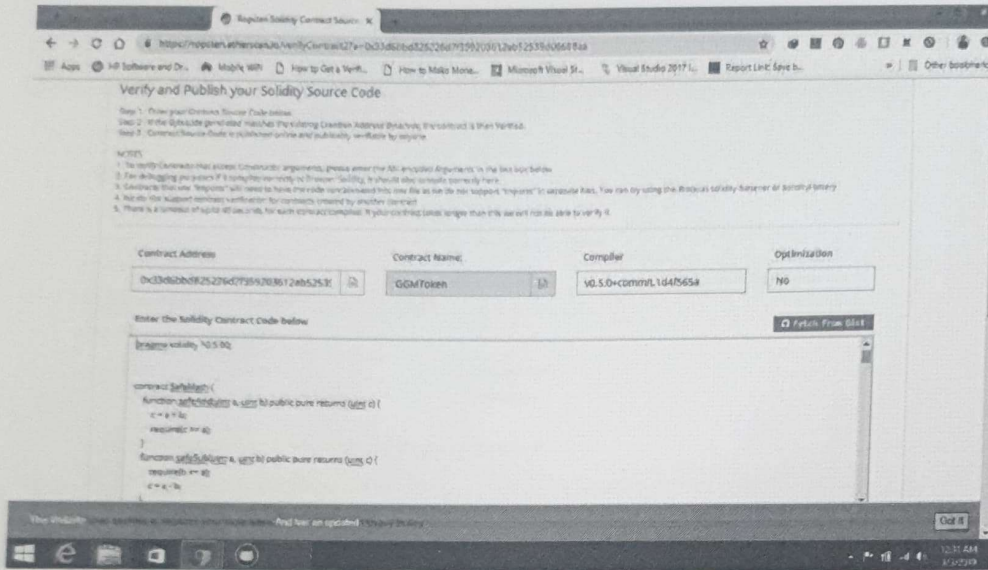
Just make sure you are on Ropsten TestNet and not on the MainNet and then hit Confirm. Now open up MetaMask again and click on the first transaction. It'll take you to Etherscan where you can observe the ongoing transaction. It may take up to 30s to confirm the transaction. Once it's confirmed it looks like the following:



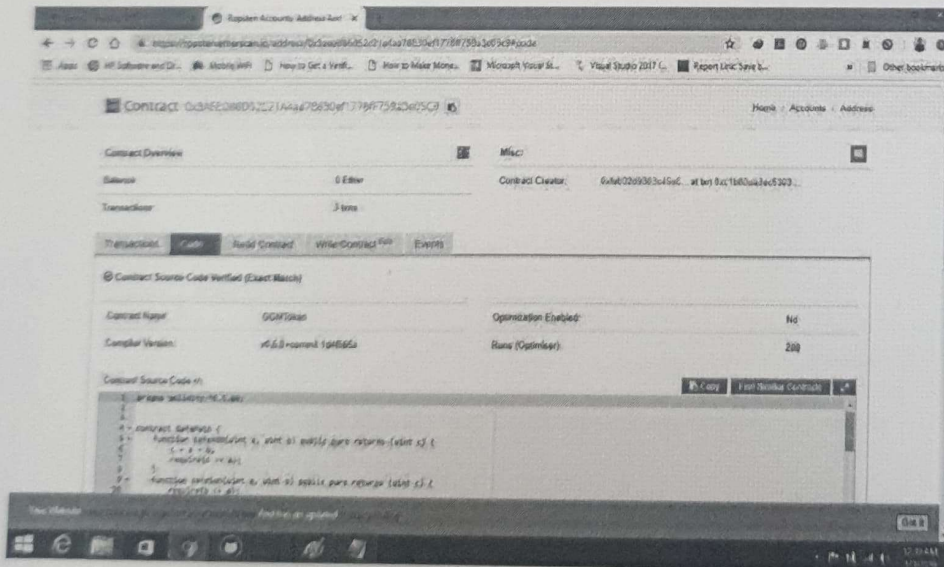
Step 6: Publish and Verify Contract
Click on contract address. On the new page click "Code".



Now click on "verify and publish" link. Once we are taken to the new page, fill up the details such as compiler version, Enable Optimizations etc and paste the solidity source we deployed earlier.

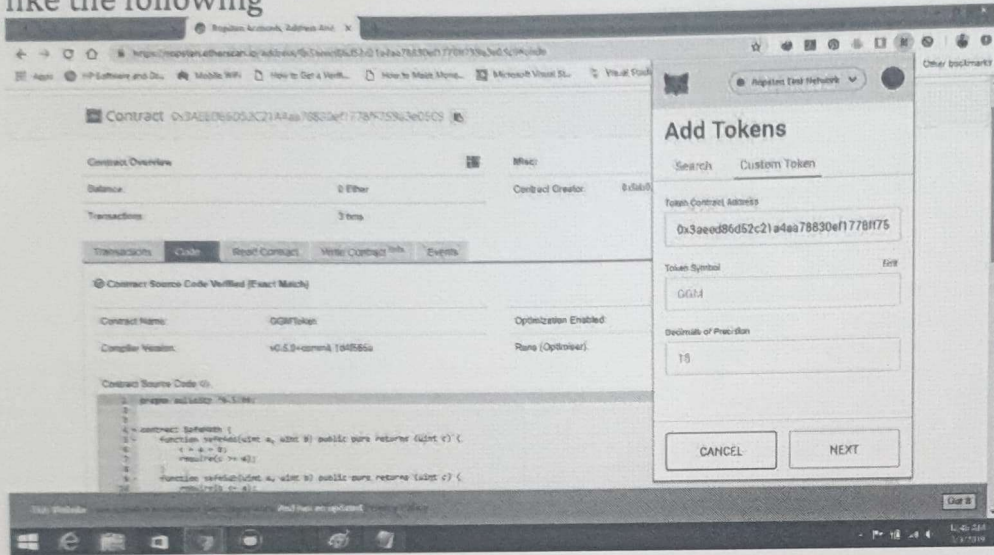


Make sure the compiler version you choose matches the one you deployed your code against in the first step. Now hit "verify and publish". If successful, it'll generate bytecode and ABI as following:



Step 7: Add token to your wallet

Now that our token has been verified, we should receive all the initial tokens (10000000) when we add it to your wallet. So, copy the contract address, go to MetaMask -> Add Token -> Custom token and paste the address. It looks like the following



Hit Next and Confirm. We should now see all the initial supply

