

Thank you for purchasing Pancake Sniper!

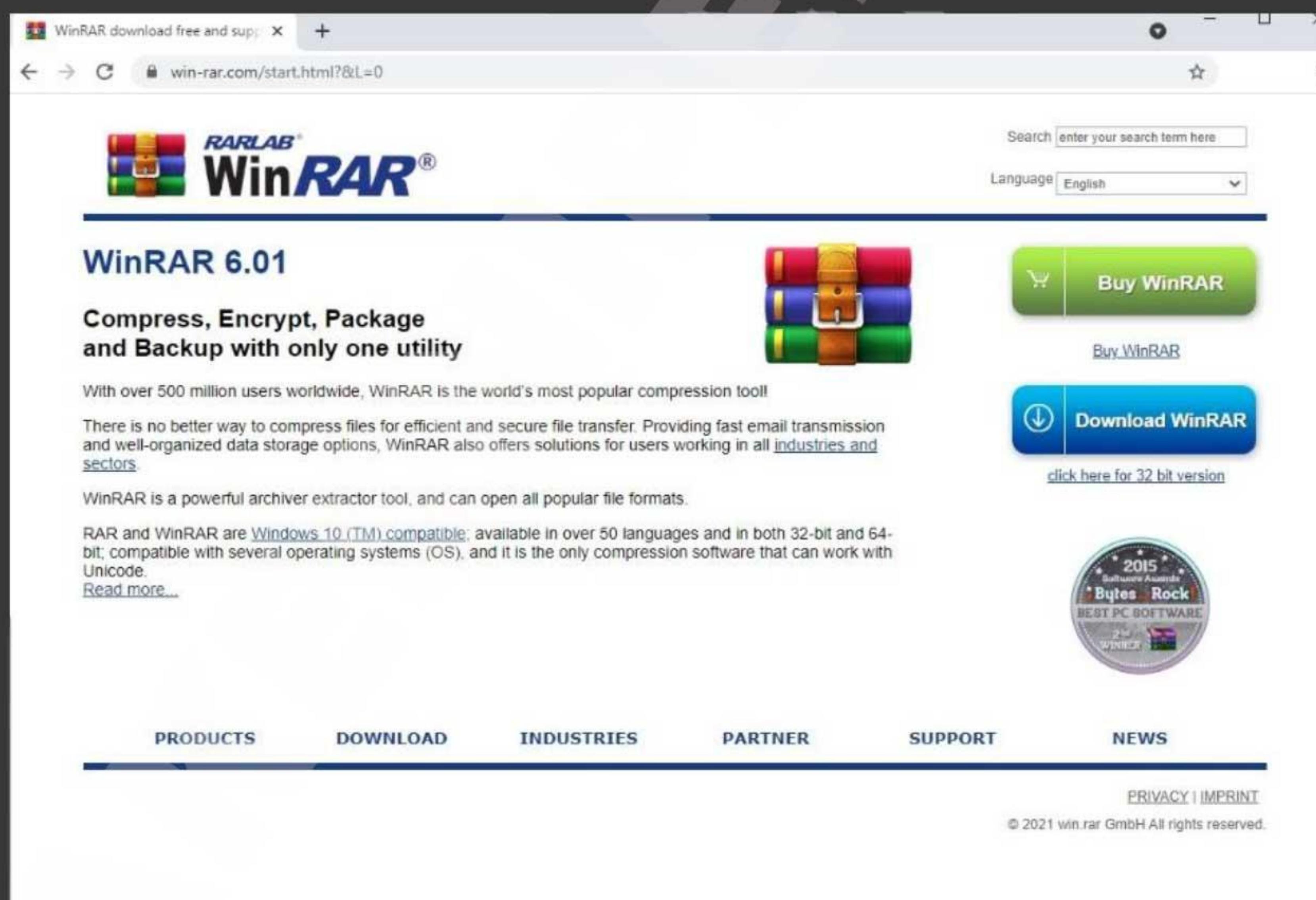
## Install Programs and Plugins

### 1. Download and install Chrome



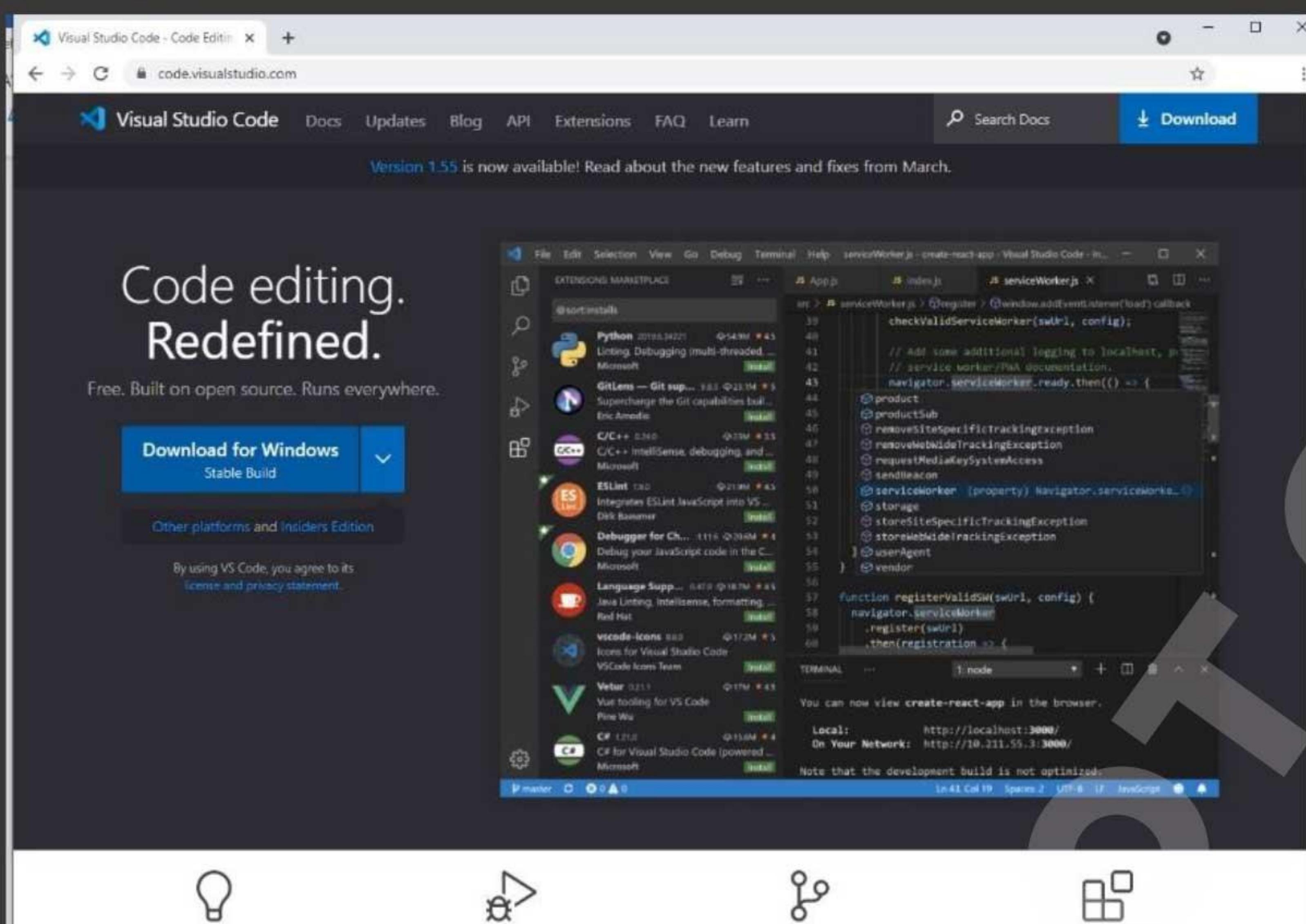
### 2. Download and install win.rar

- In the chrome search bar type [www.win-rar.com](http://www.win-rar.com)
- **Download WinRAR -> Download WinRAR**
- Follow the install steps



### 3. Download and install Visual Studio Code

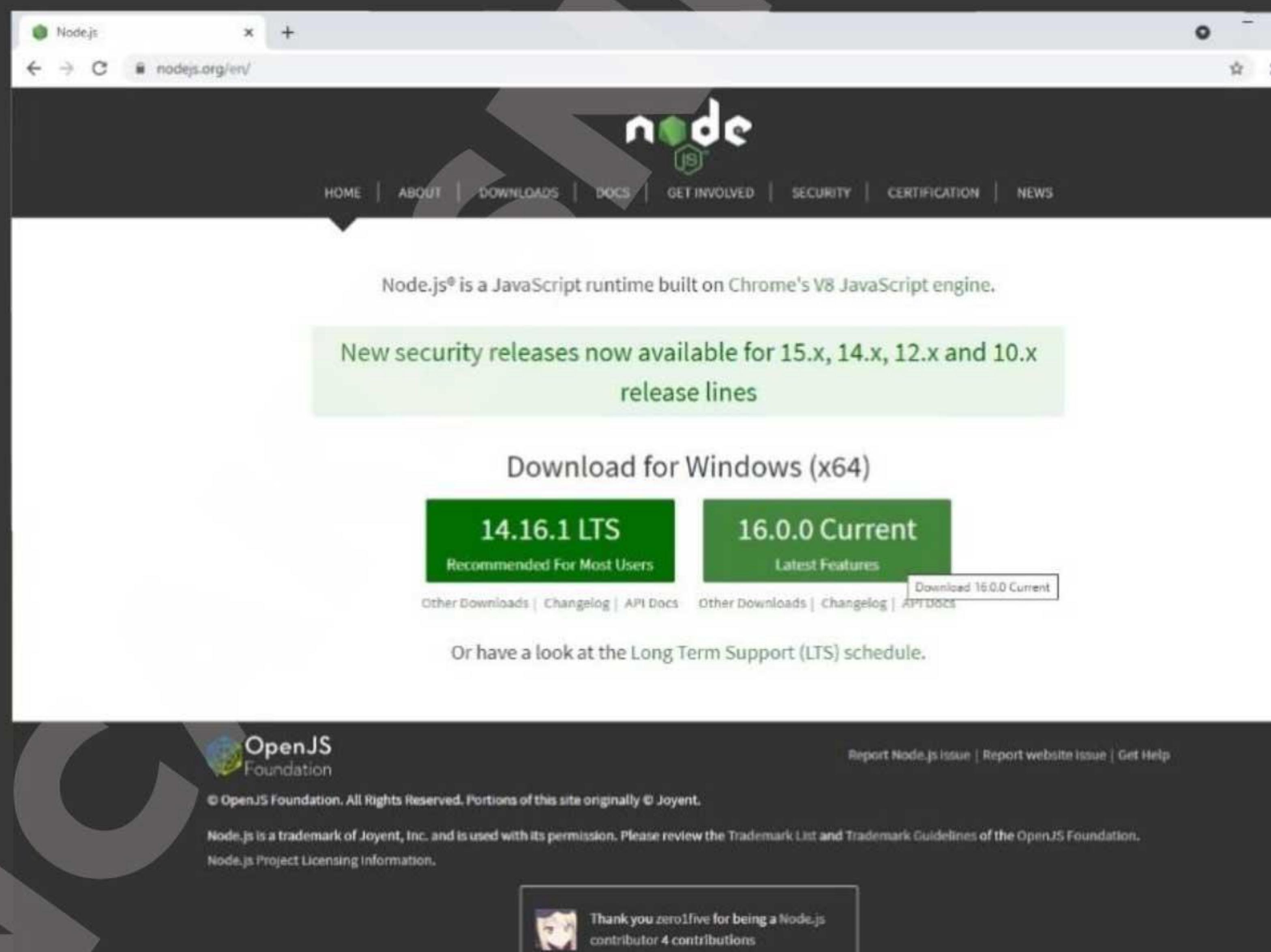
- In the chrome search bar type <https://code.visualstudio.com/>
- Press **Download for Windows**



- Install program

#### 4. Download node.js

- In the chrome search bar type [www.nodejs.org](http://www.nodejs.org)
- Press **Current**
- 
- Open **downloads** and Install plugin by clicking on **node.msi** file



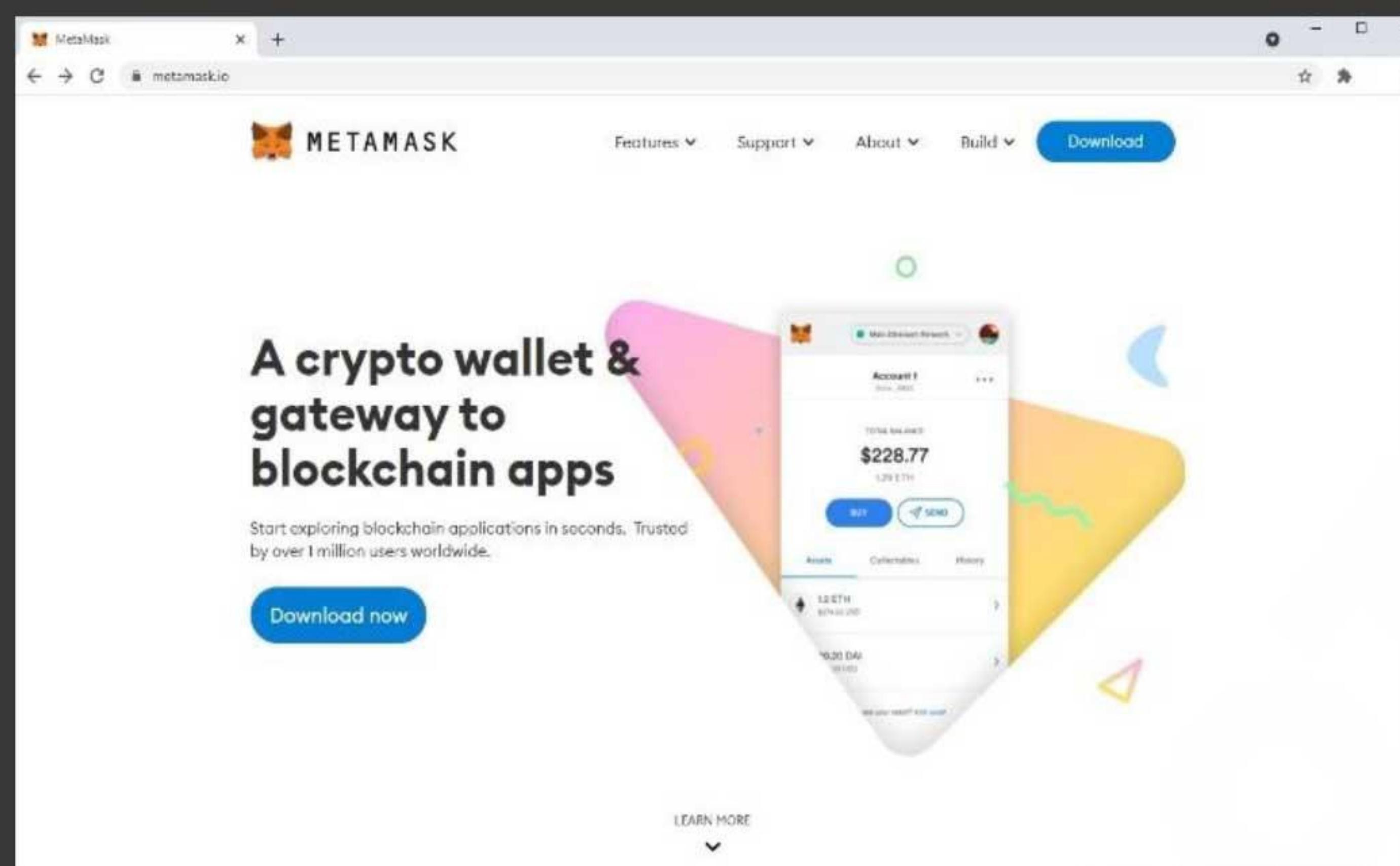
\*IMPORTANT

RESTART YOUR COMPUTER AFTER INSTALLING THESE PROGRAMS

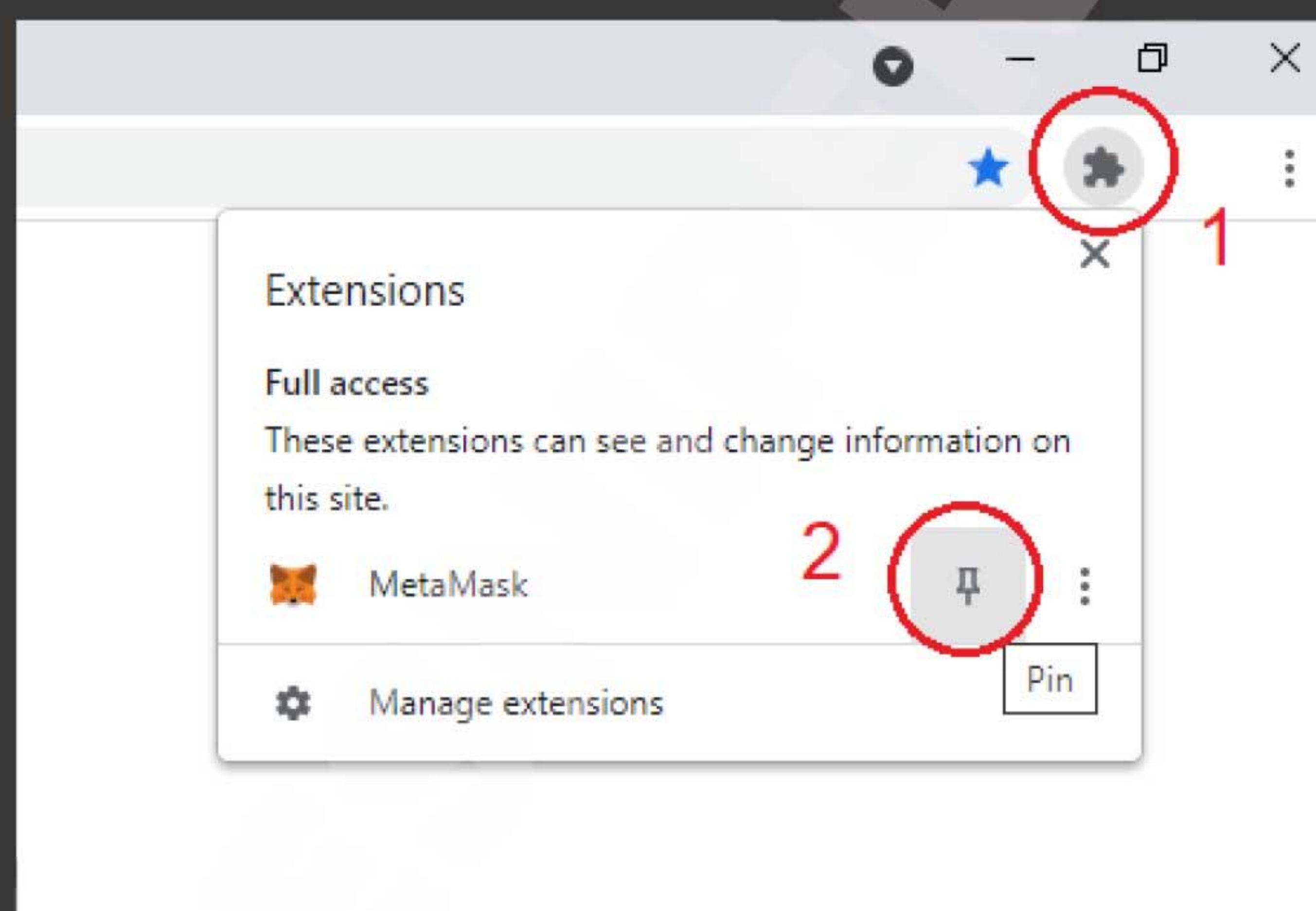
PANCAKESNIPERBOT.COM

5. Open Chrome and in the search bar type [www.metamask.io](http://www.metamask.io)

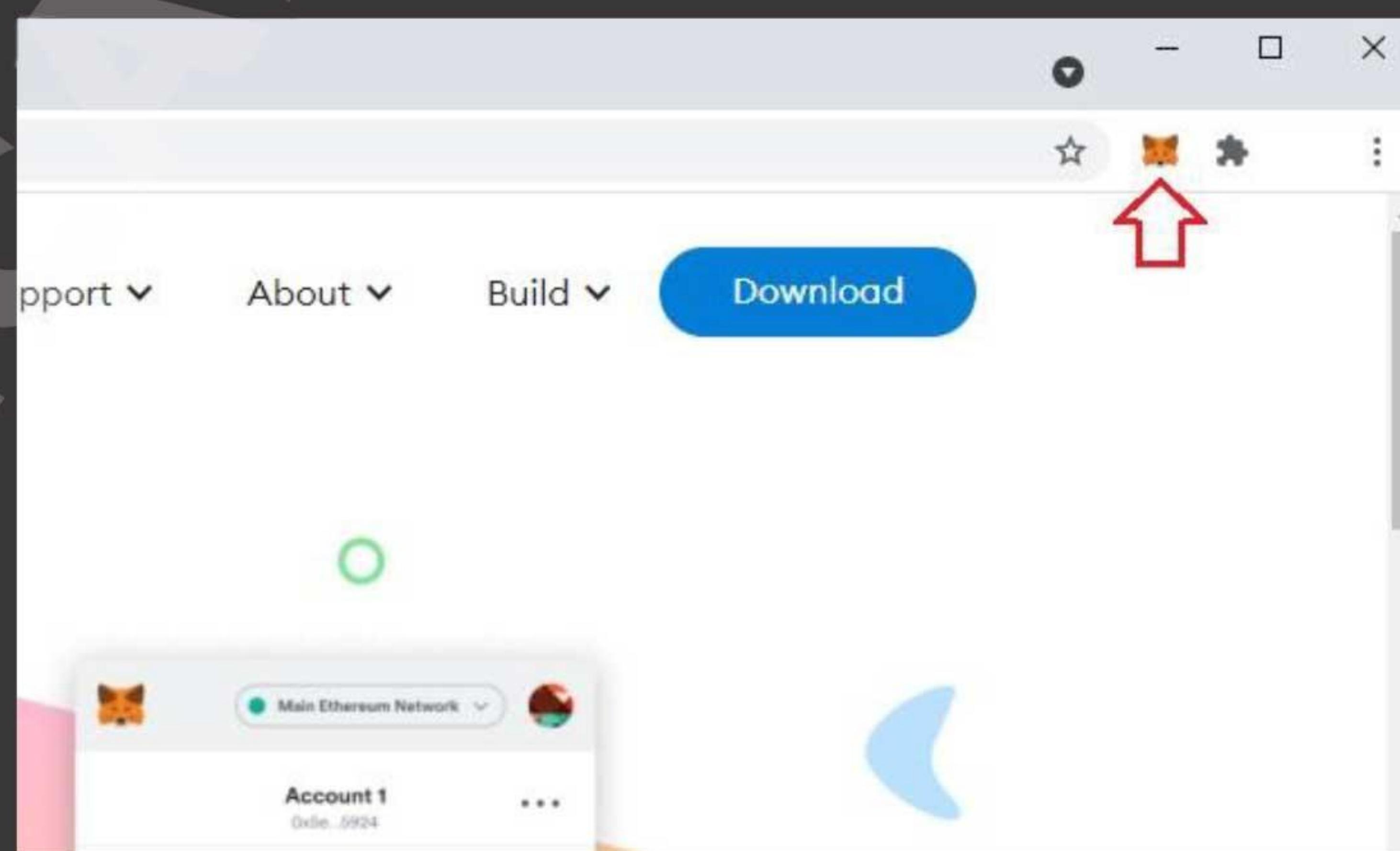
- Press: **Download** - > **install MetaMask for Chrome** – > **Add to Chrome**



- Pin MetaMask icon on your chrome browser



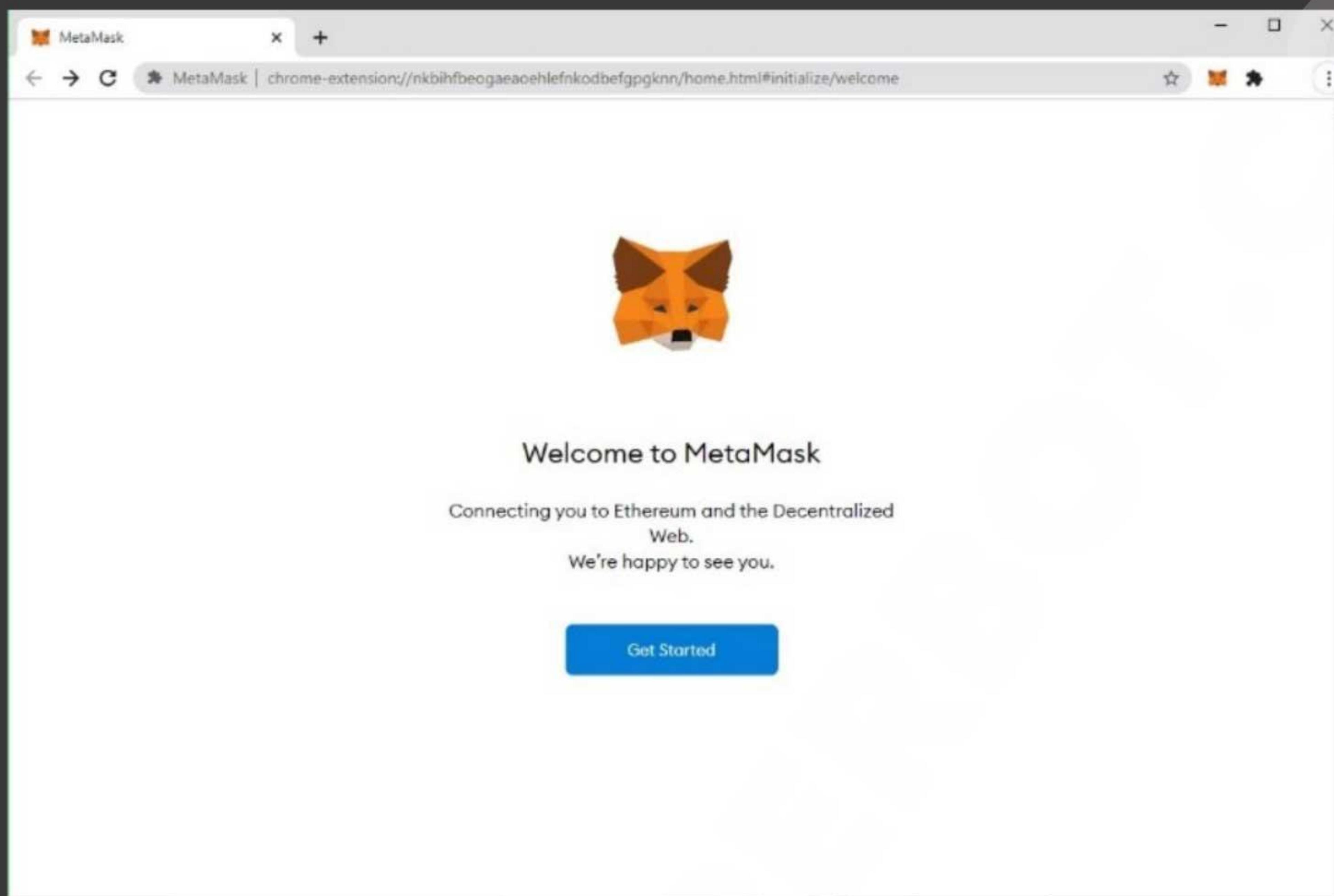
- Press on the MetaMask icon



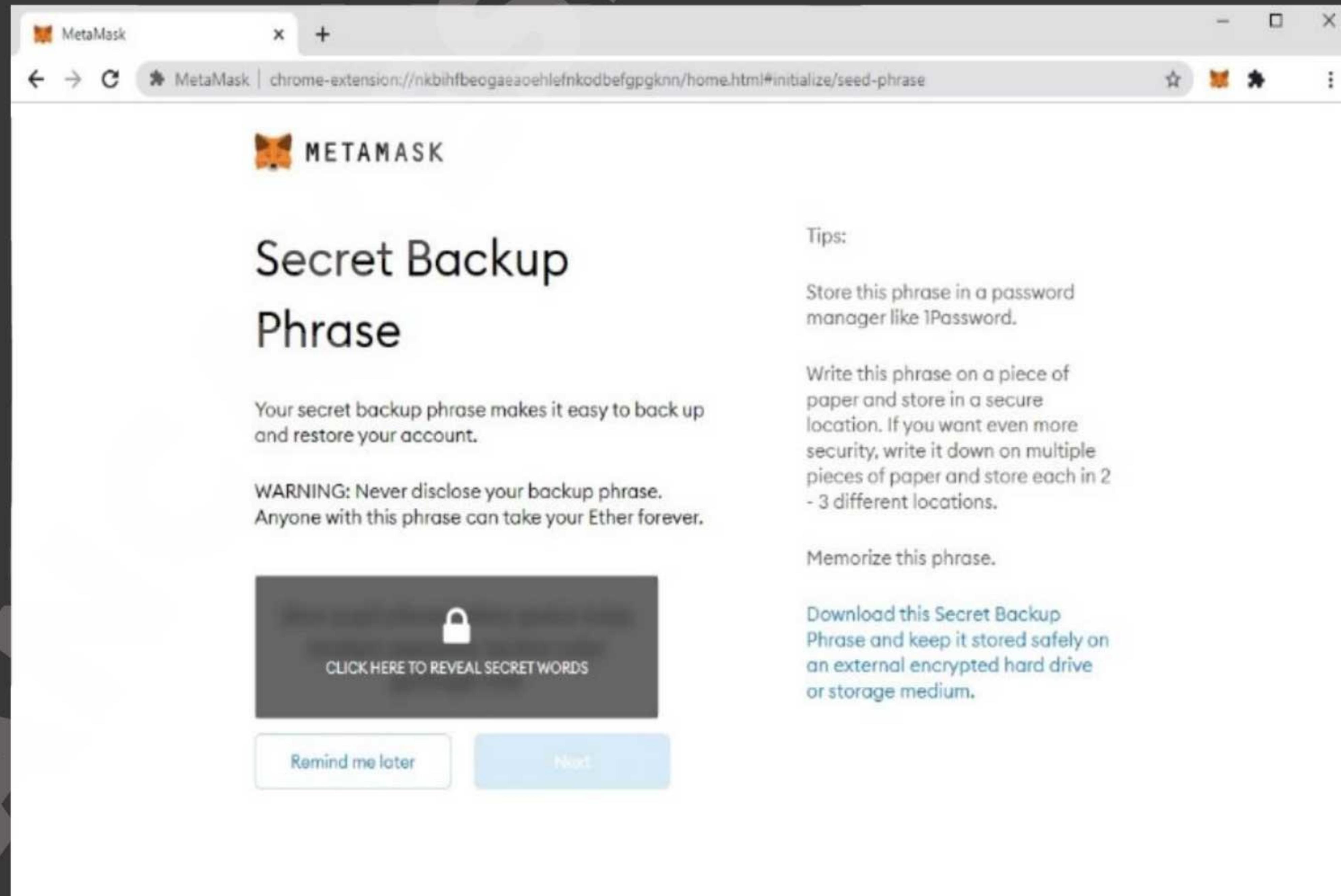
PANCAKESNIPERBOT.COM

## 6. Set up your MetaMask account

- Press **Get Started** –> **Create a Wallet** - > Create Password



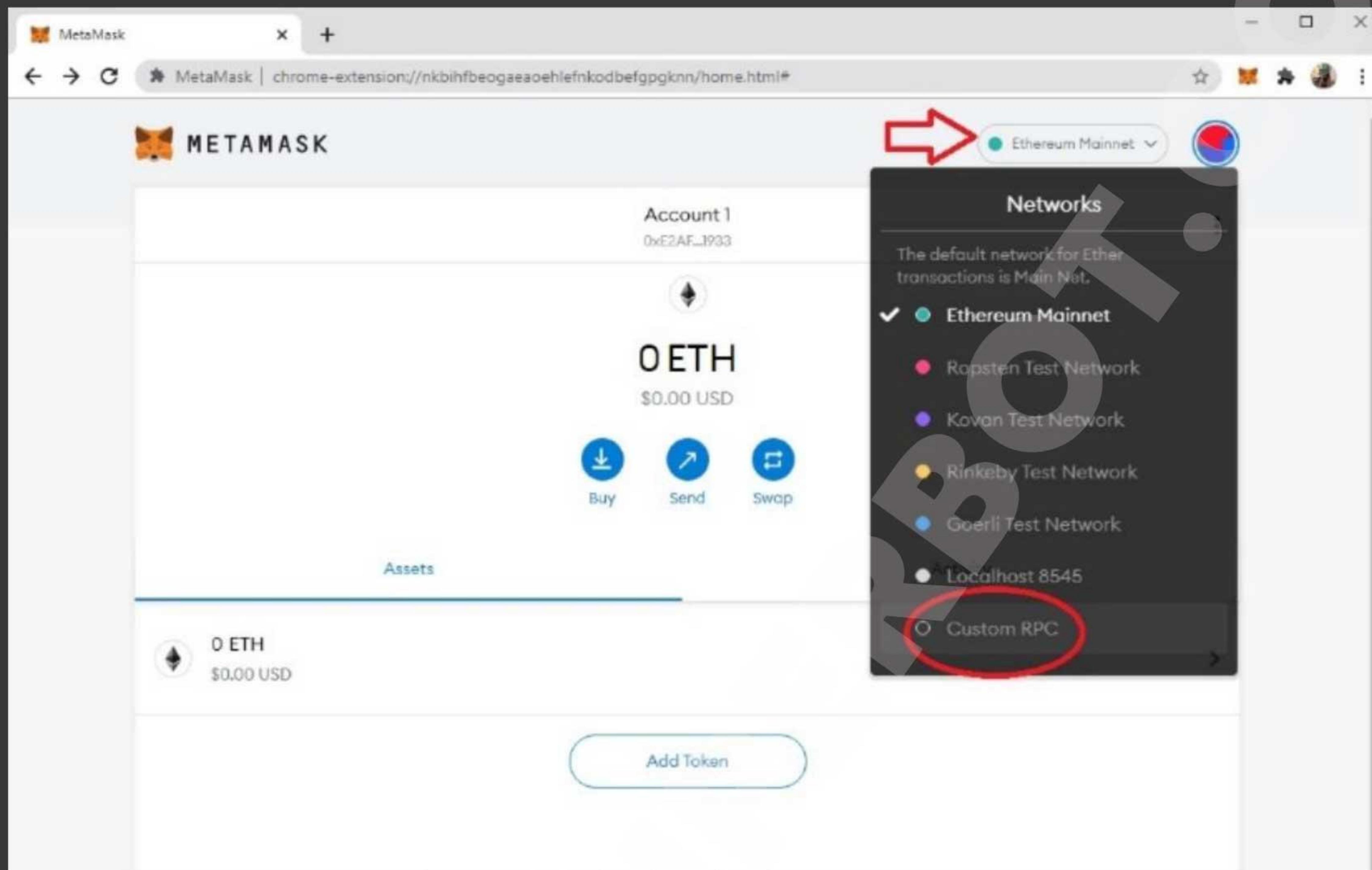
- Click on given window to receive your 12 word seed Phrase. Write down the given words on a paper and do click next to confirm the words again.



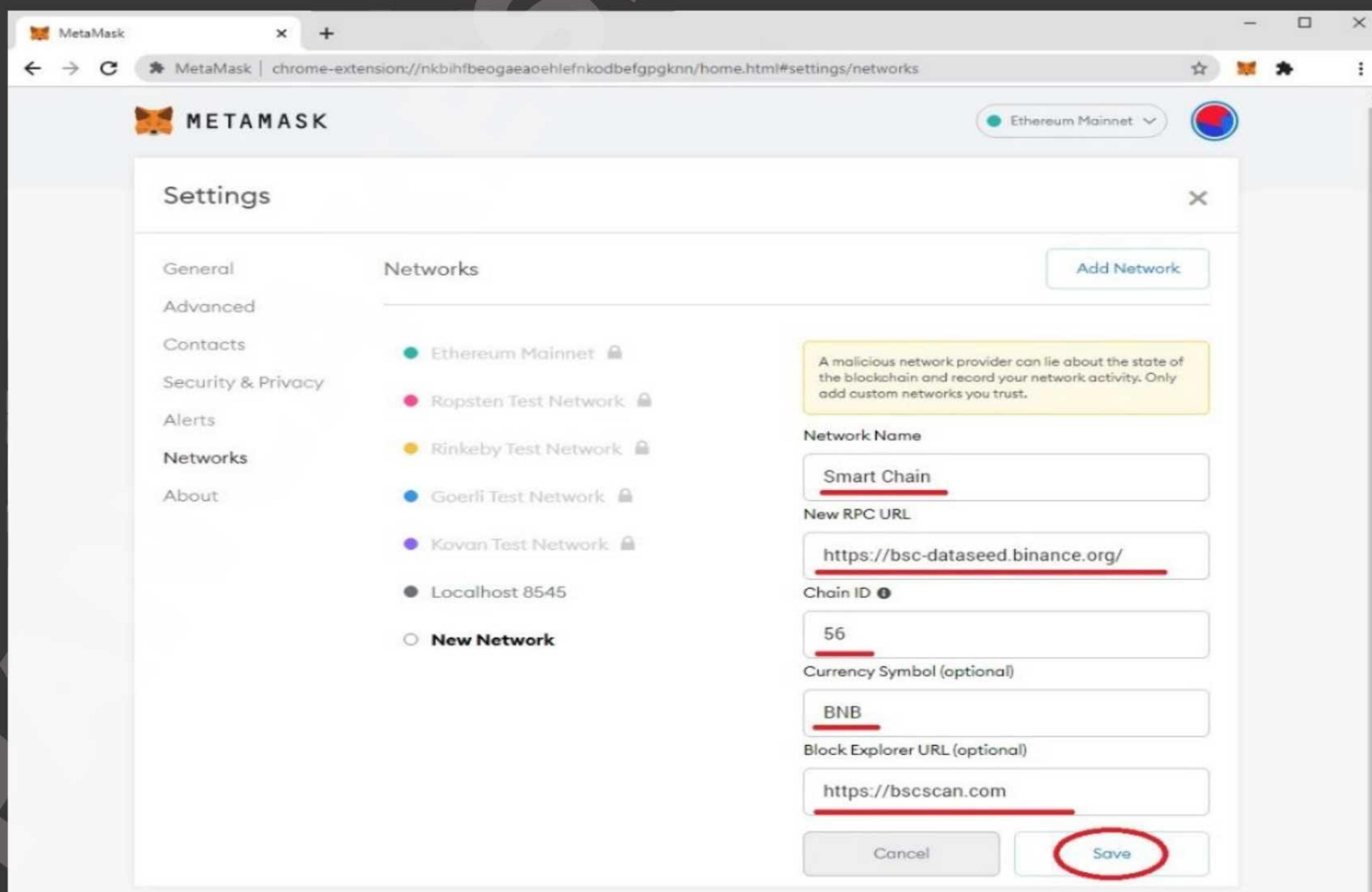
PANCAKESNIPERBOT.COM

## 7. Connect your MetaMask wallet to Binance Smart Chain

- Click on **Ethereum Mainnet** and press on **Custom RPC**



- Enter the given data bellow:



**Network Name:** Smart Chain

**New RPC URL:** <https://bsc-dataseed.binance.org/>

**ChainID:** 56

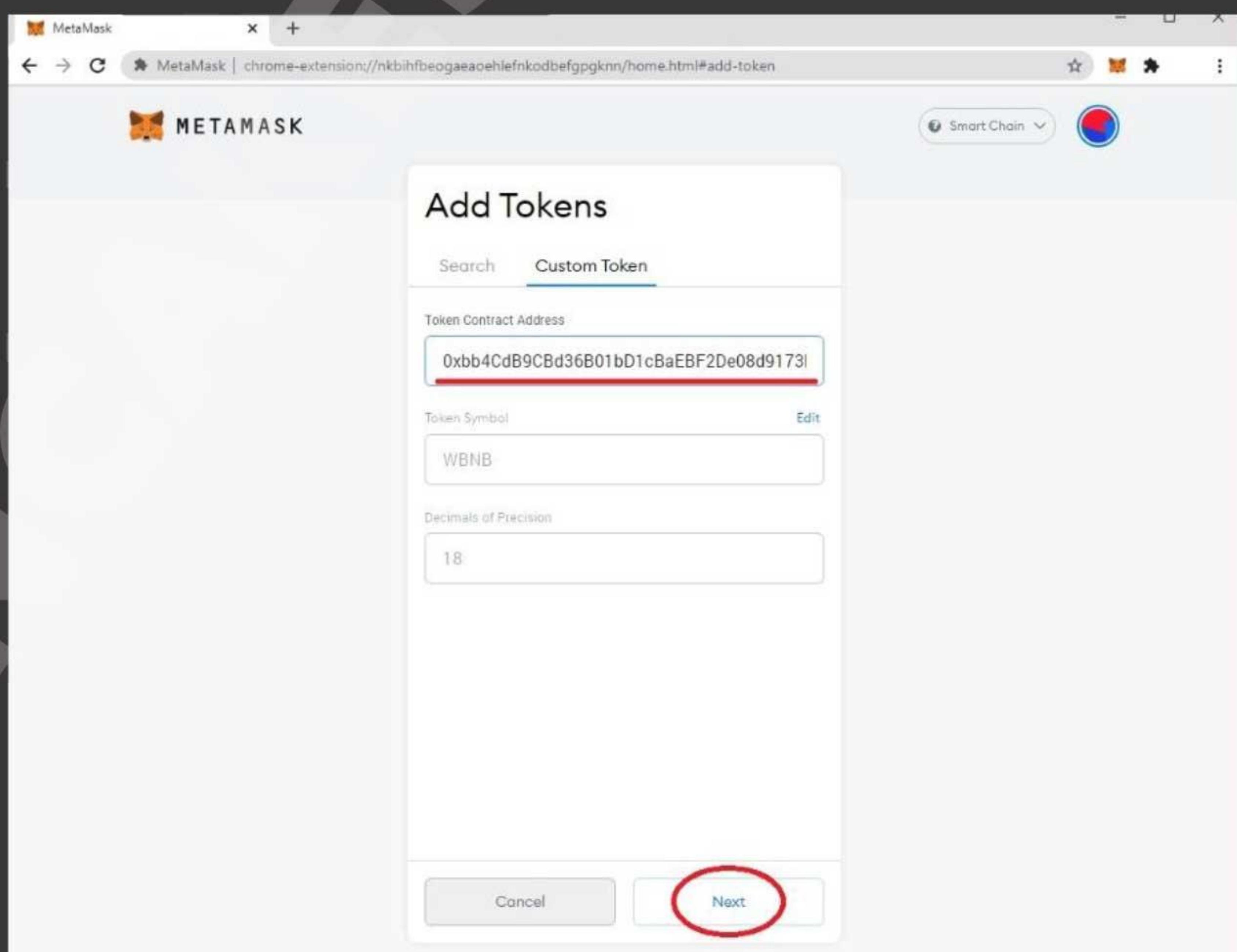
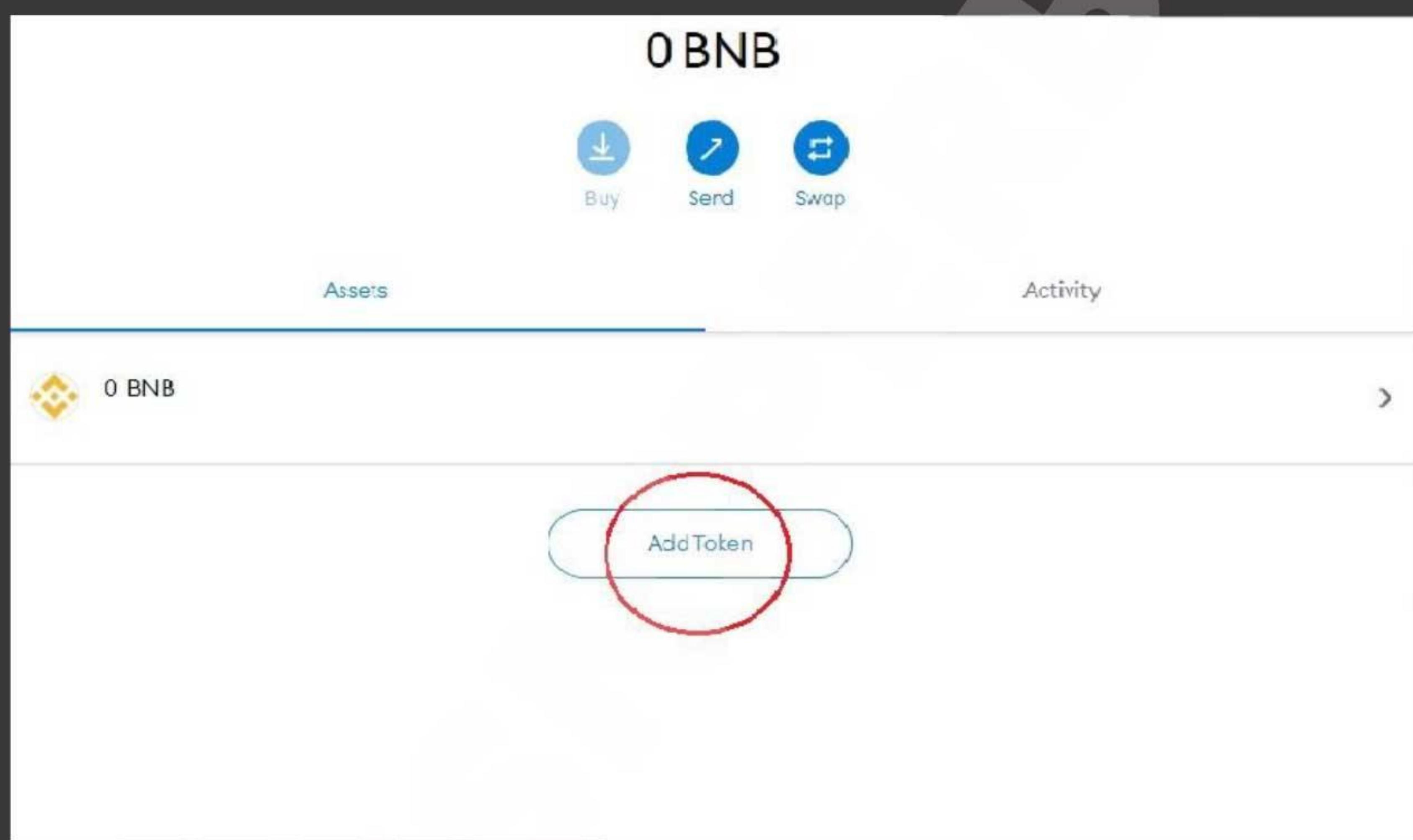
**Symbol:** BNB

**Block Explorer URL:** <https://bscscan.com>

Or copy from binance official site <https://academy.binance.com/en/articles/connecting-metamask-to-binance-smart-chain>

8. Add WBNN token name to MetaMask wallet

- Click on **Add Token - > Custom Token** and paste the WBNN address from [www.bscscan.com](https://www.bscscan.com) and

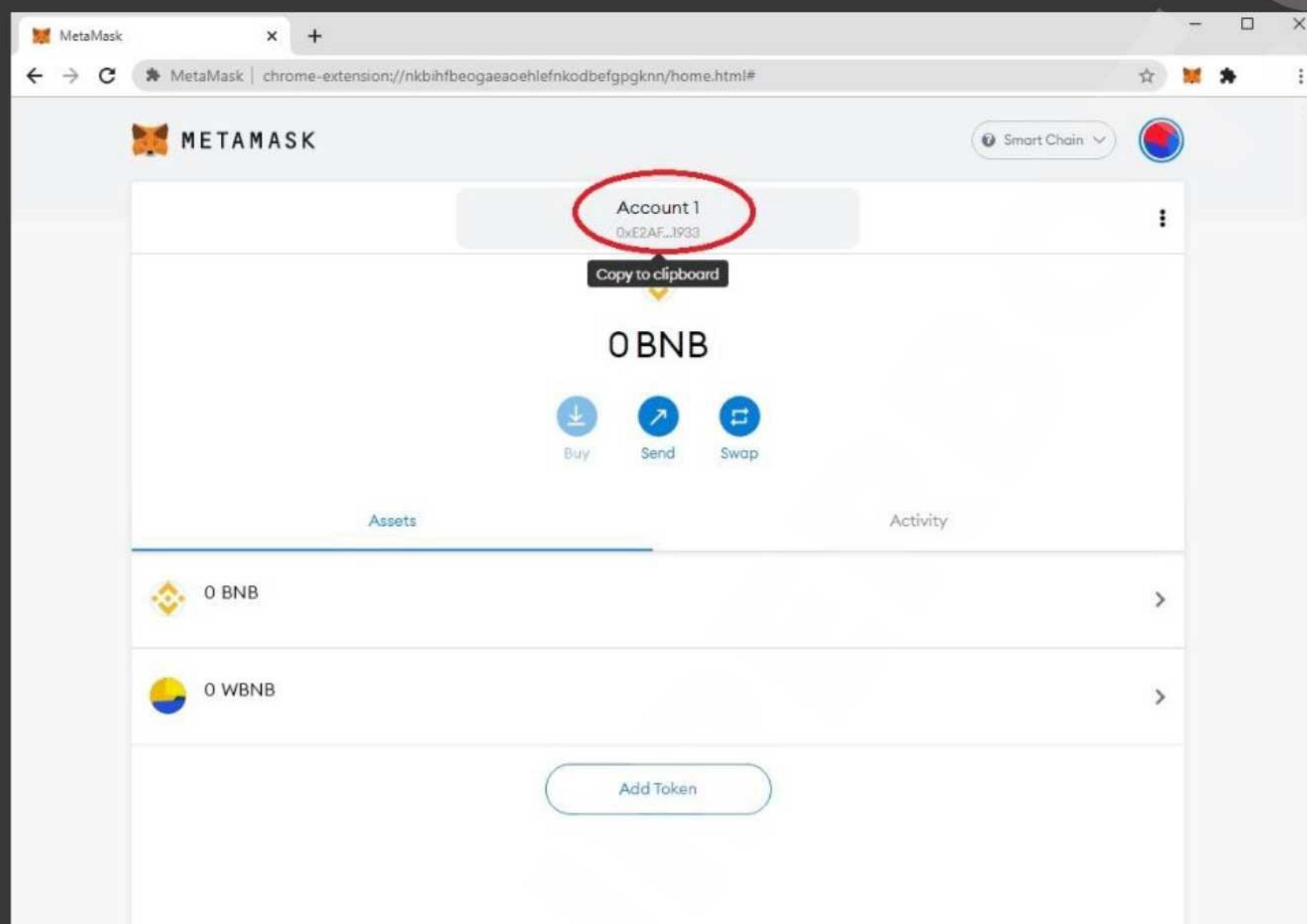


click **Next**

Link to WBNB: <https://bscscan.com/token/0xbb4Cd9CBd36B01bD1cBaEBF2De08d9173bc095c>

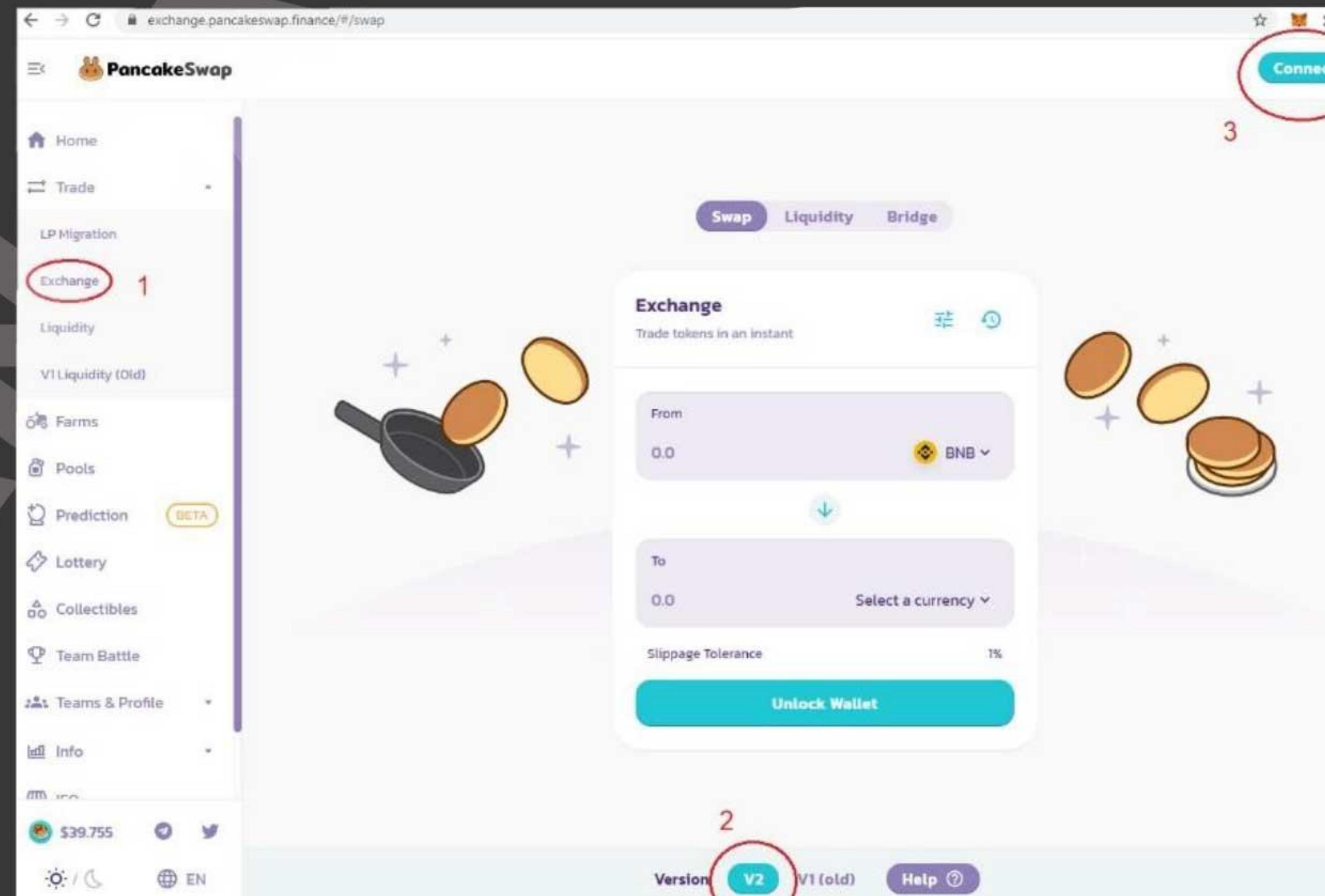
Or just copy this address: 0xbb4Cd9CBd36B01bD1cBaEBF2De08d9173bc095c

- Load BNB to your MetaMask wallet from any exchange or wallet by sending the BNB to the wallet address from the top. Click to copy

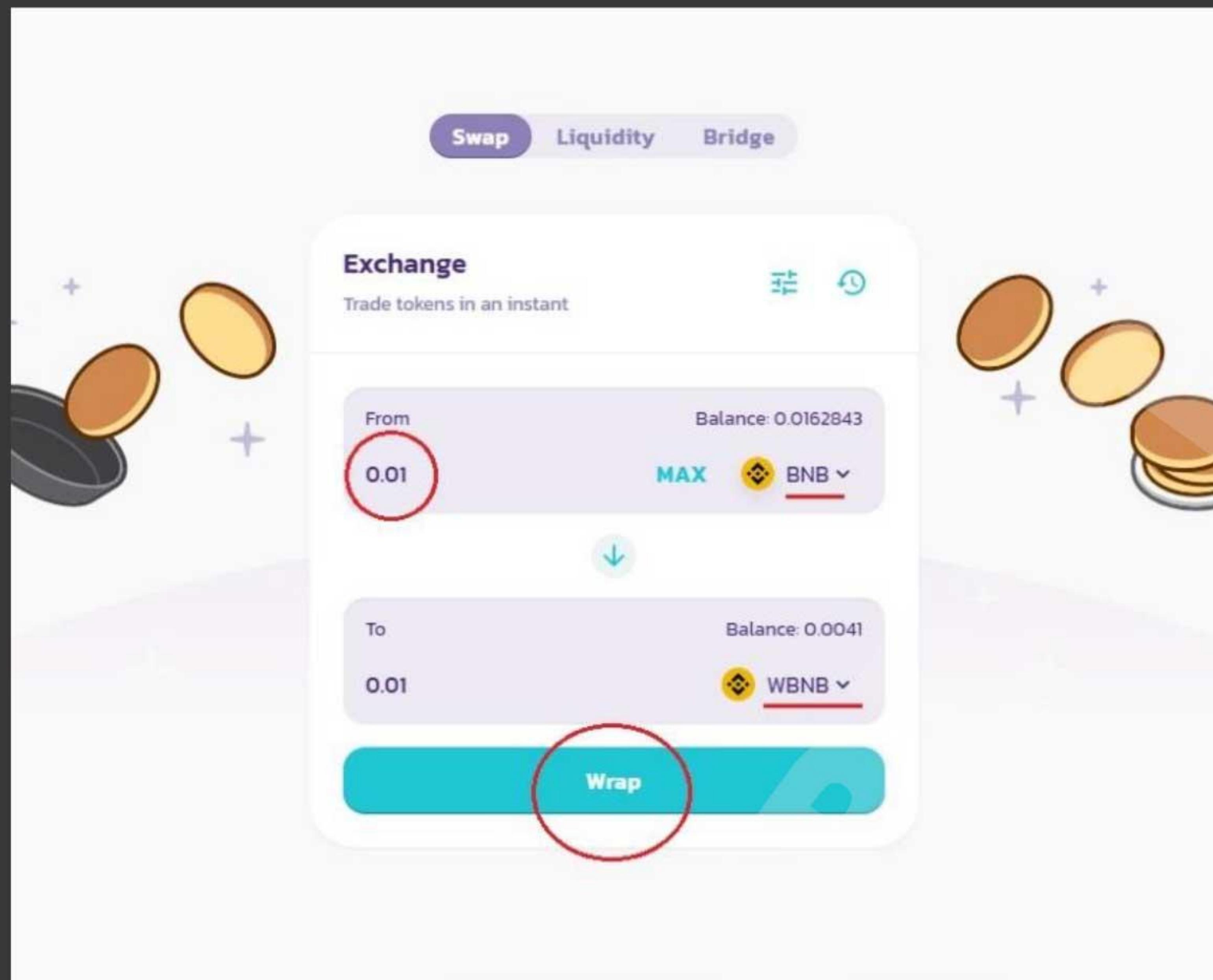


## 9. Connect MetaMask to PancakeSwap

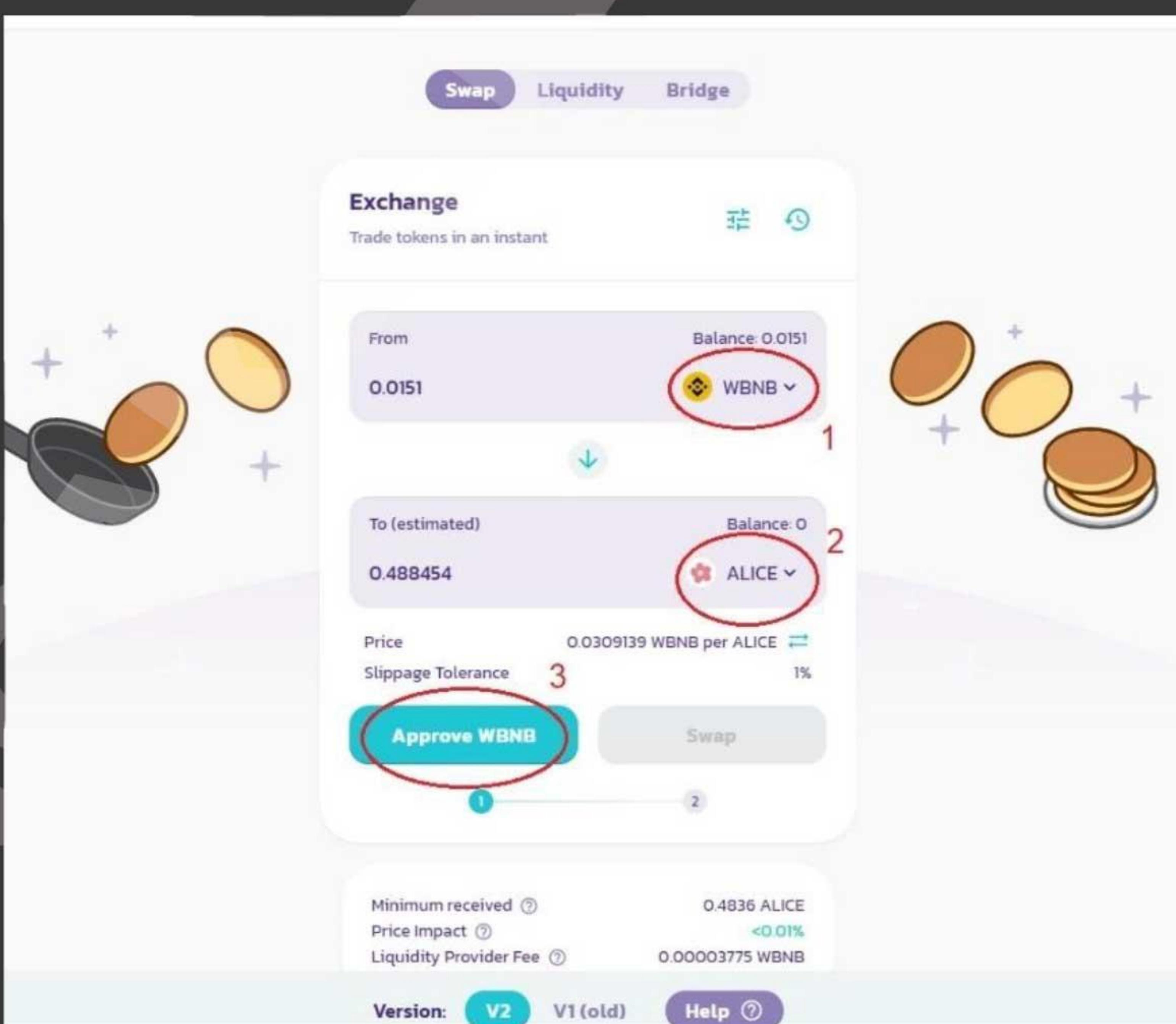
- Click **Trade -> Exchange -> V2 –> Connect**



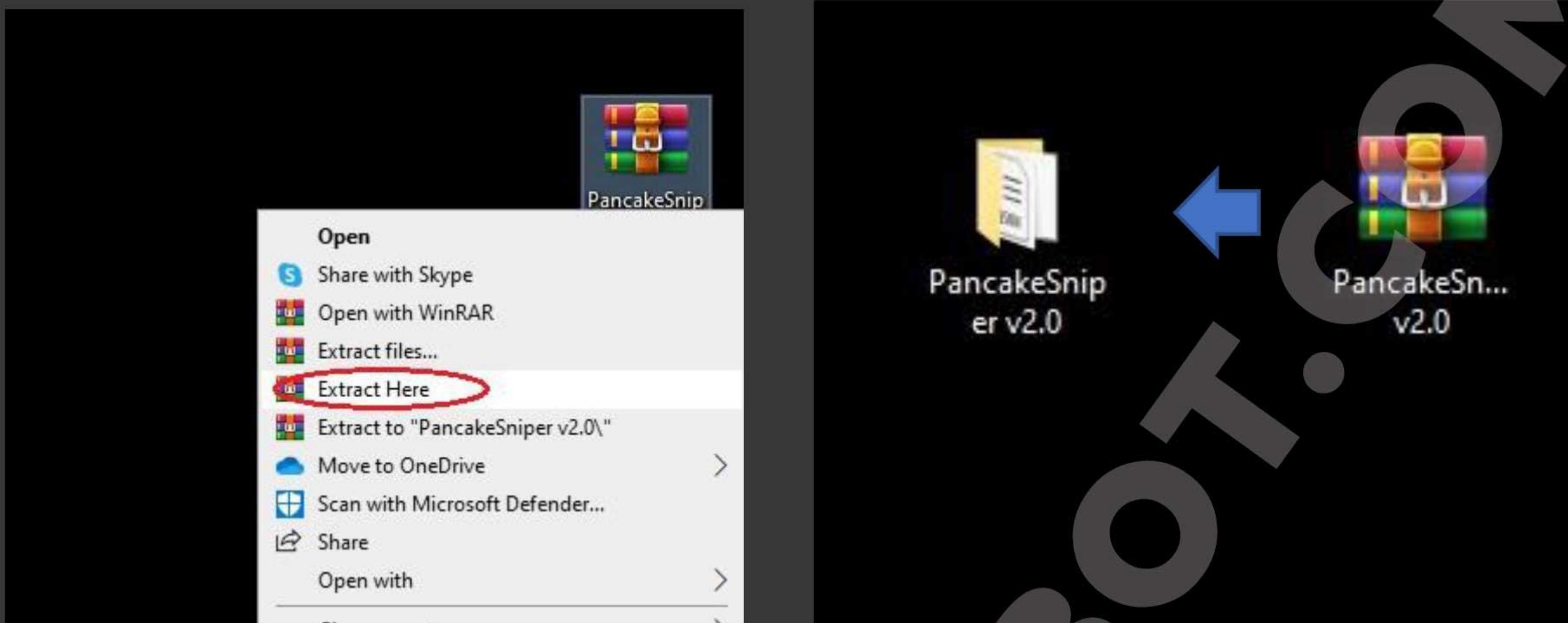
- Wrap BNB to WBNB (leave some amount of BNB to cover the tax for future snipes. Each snipe will be max of around 20 cents).



- Select WBNB and any other coin for example Alice and click on **Approve WBNB** but don't swap (this will allow to use WBNB in any transaction)

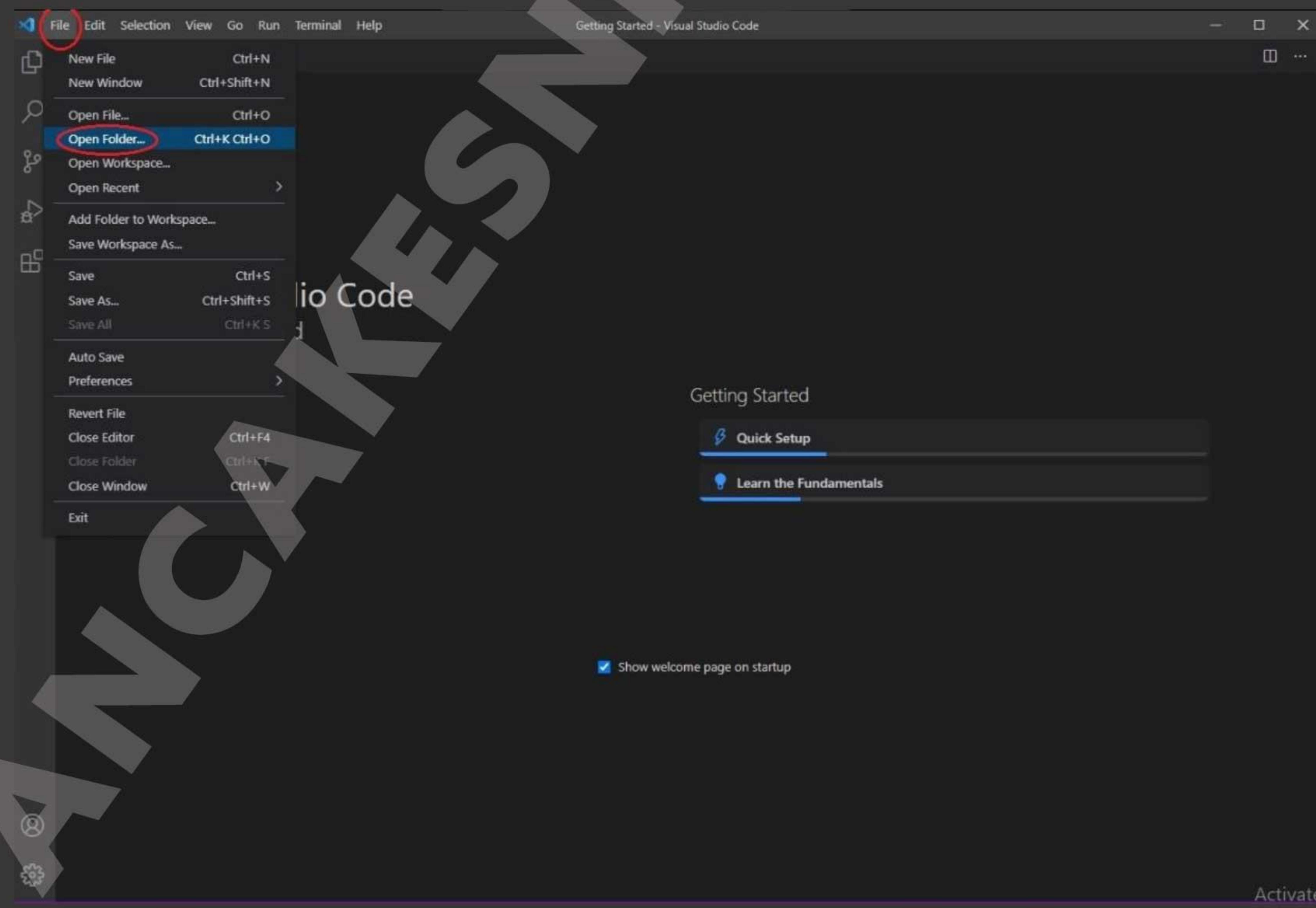


10. Unzip the purchased file using WinRAR

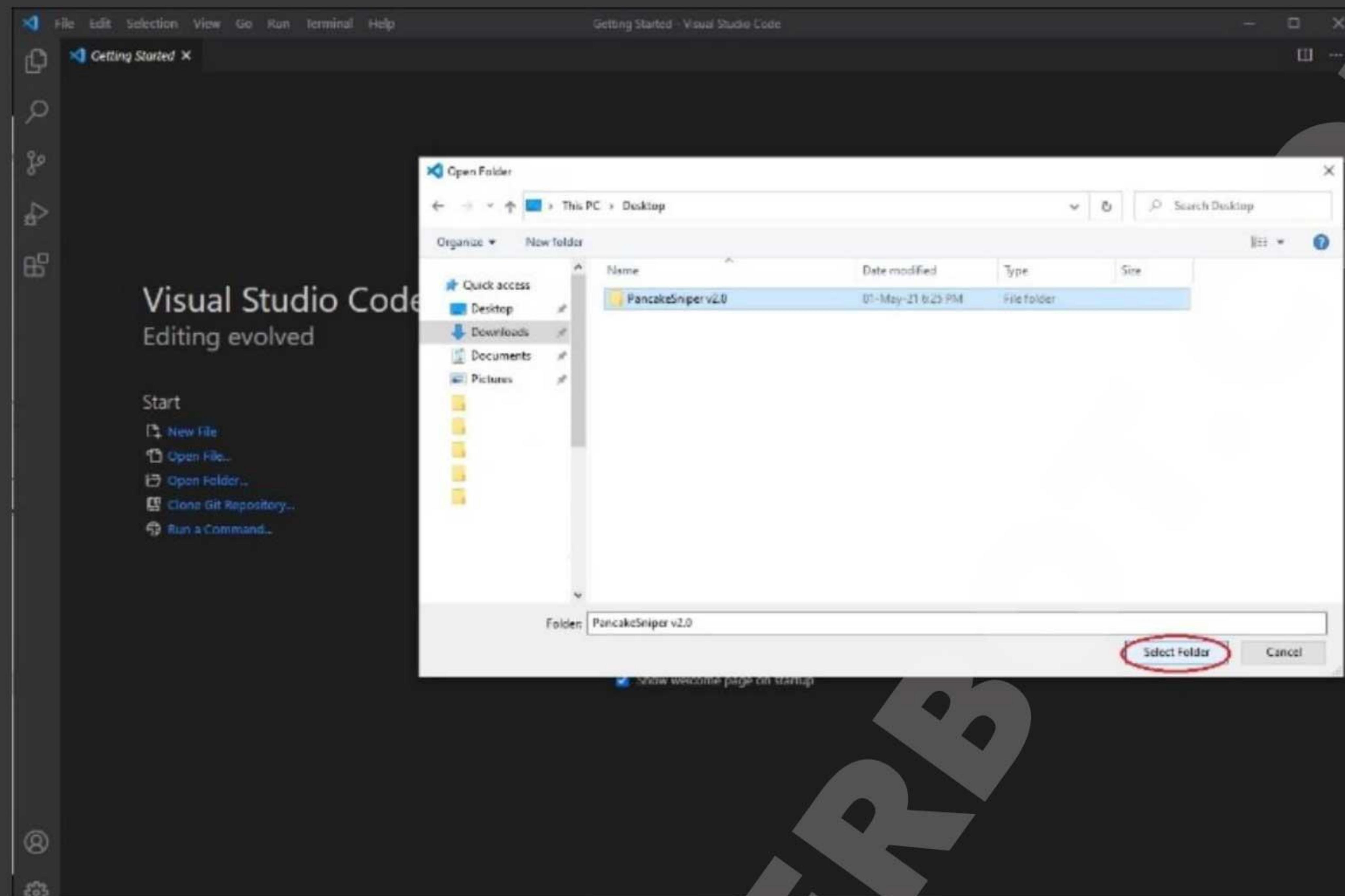


11. Open Visual Studio Code and import file

- Click **File -> Open Folder (Workspace for Mac)**



- Select unzipped folder



- Open bot.js config file

```

const ethers = require('ethers');
//ENTER YOUR DETAILS Refer to instructions!
//1. Snipe details
const SnipeID = "Enter ID of the token to snipe. Refer to instructions for formatting"
const Receive = "Enter ID of the token to buy. Usually the same as SnipeID"
const amountIn = ethers.utils.parseUnits('Enter how much WBNB to spend', 'ether')

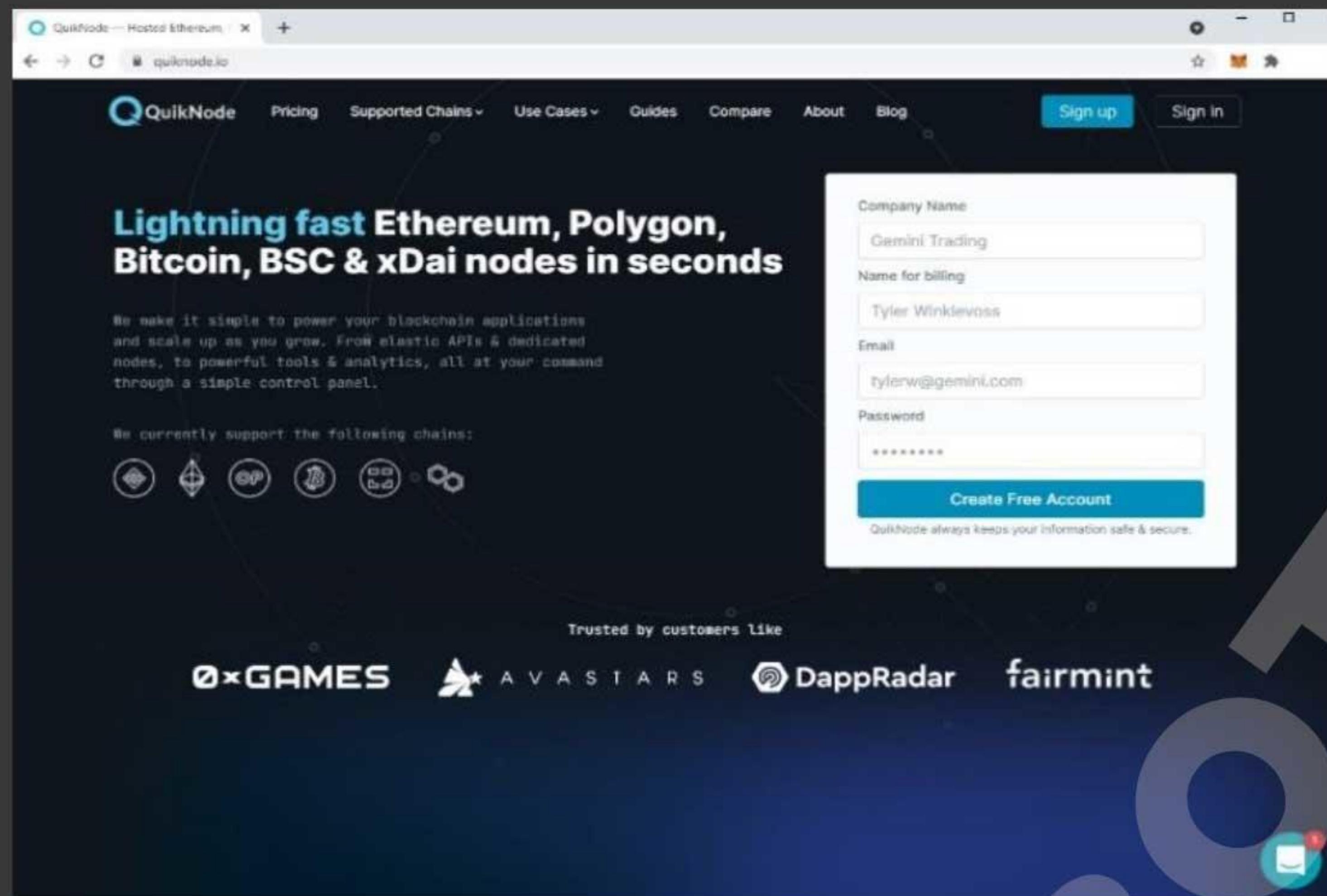
//2. Wallet / connection details
const NBS = "Enter NBS address. Refer to instructions for links"
const Seed = "Enter your wallet 12 word seed phrase"
const recipientAddress = "Enter your wallet address where to send the bought tokens"

//3. Optional settings
const Spend = "0xbba4C0B9C0d36001bD1c0aEBF20e0009123bc095c"
const routerAddress = "0x1f043C718714eb61d5aA57878854704E25624E"
const slippage = ethers.utils.parseUnits('0', 'ether')

////Done.. Do NOT change code after this!

const MethodID = "0x705d739" // function ID it searches for (one is for MEXH/WBNB pools and other is for pools t
const MethodID2 = "0xe0e33700" // function ID it searches for (one is for MEXH/WBNB pools and other is for pools
const provider = new ethers.providers.WebSocketProvider(NBS);
const wallet = ethers.Wallet.fromMnemonic(Seed);
const account = wallet.connect(provider);
provider.removeAllListeners();
const router = new ethers.Contract(
  routerAddress,
  ["function swapExactTokensForTokens(uint amountIn, uint amountOutMin, address[] calldata path, address to, uint account");
);
console.log("Connecting to the blockchain");
console.log("Starting to scan the network for a matching transaction based on the config entered");
console.log("As soon as a matching transaction has been found, it will be displayed here");
provider.on('pending', async (tx) => {
  provider.getTransaction(tx).then(function (transaction){
    if (transaction != null && transaction['data'].includes(MethodID2) && transaction['data'].includes(SnipeID) | 
      console.info(transaction)
  })
})
  
```

12. Open Chrome and go to [www.quiknode.io](http://www.quiknode.io) and register



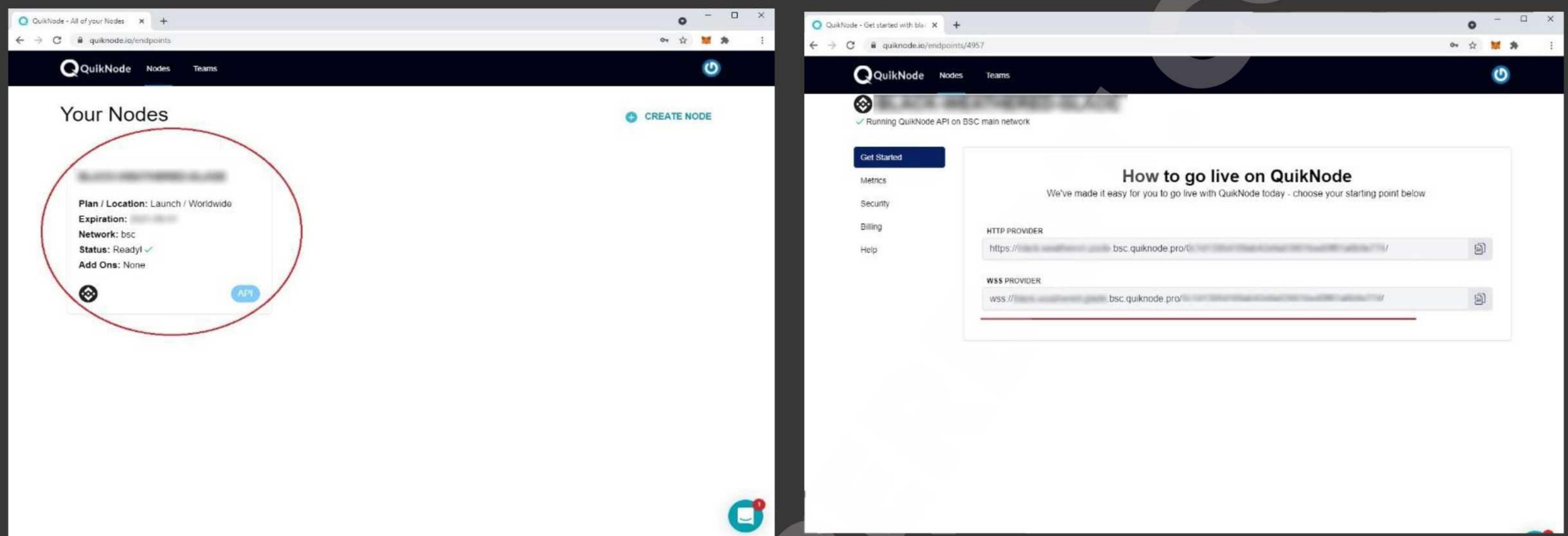
- Click on **CREATE NODE** and select a **LAUNCH** plan
- Click on **BSC** and press **CONTINUE TO BILLING**. Fill given credit card information to continue (you

The image shows three sequential screenshots of the QuikNode node creation process:

- Screenshot 1: Selecting Network**  
The user is on the 'Configure a new node' page. They have selected the 'BSC' network tab (circled in red). Other tabs shown are ETH, BTC, XDAI, POLYGON, and OPTIMISM. A 'CONTINUE TO BILLING' button at the bottom left is also circled in red.
- Screenshot 2: Last Step: Billing Info**  
The user is on the 'Last Step: Billing Info' page. It shows a 'PAY WITH CREDIT CARD' form with fields for Credit Card Number, Expiration Date, CVV, Postal Code, and Coupon Code. A 'CREATE MY NODE!' button is at the bottom. To the right, there's a summary of the 'Total Monthly Cost' (\$9.00 / mo) and a 'Plan' section for 'LAUNCH'.
- Screenshot 3: Select a Plan**  
The user is on the 'Select a plan to get started' page. It lists several plans:
  - LAUNCH**: \$9.00 / mo, Monthly calls: 300 Thousand Requests/sec: No rate limiting. A 'SELECT' button is circled in red.
  - PRO**: \$99.00 / mo, Monthly calls: 20 Million Requests/sec: No rate limiting. A 'SELECT' button is circled in red.
  - SCALE**: \$299.00 / mo, Monthly calls: 60 Million Requests/sec: No rate limiting. A 'SELECT' button is circled in red.
  - DEDICATED**: \$300.00 / mo, Monthly calls: Unlimited Requests/sec: No rate limiting. A 'SELECT' button is circled in red.
  - ENTERPRISE**: Starting at \$2k+ / mo, Monthly calls: Unlimited Requests/sec: Unlimited / second. A 'CONTACT' button is circled in red.

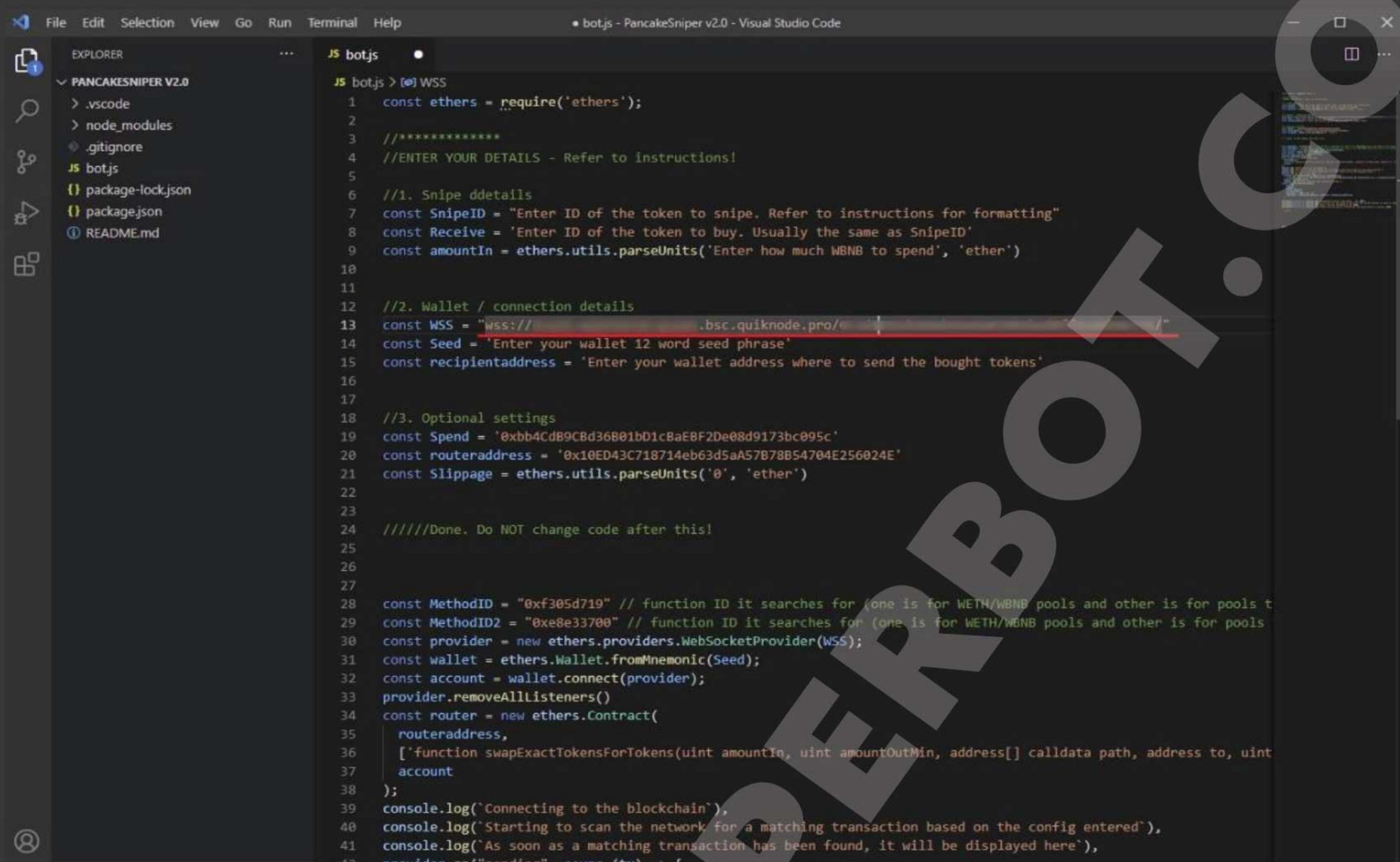
can cancel subscription after 7 days). The free plan will be enough for around 1 hour of scanning. Get the bigger plan for future scanning/sniping

- Open “Your Nodes” and copy WSS provider address



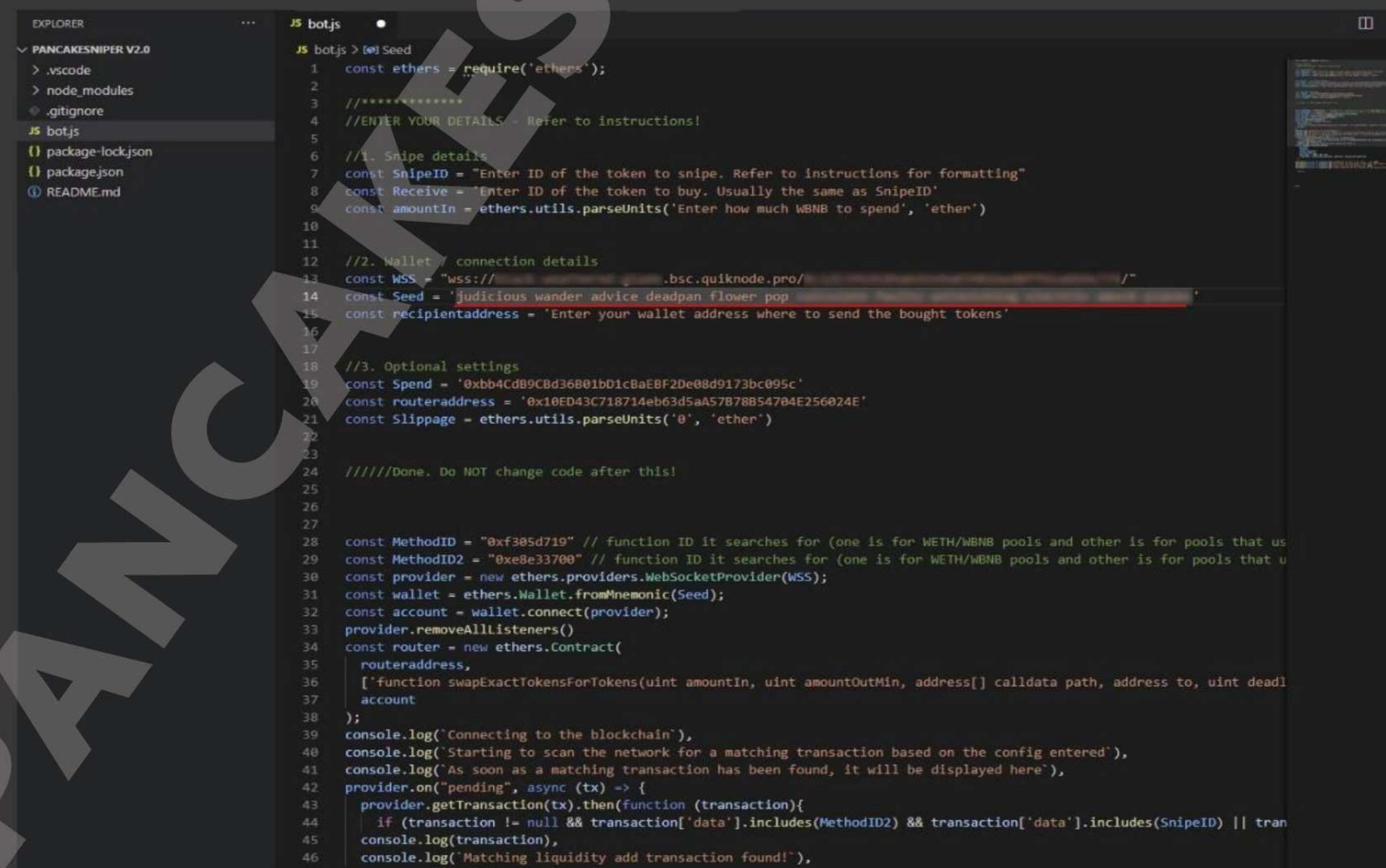
## Inserting Wallet details

1. Open Visual Studio Code and paste the address in WSS line “ ”



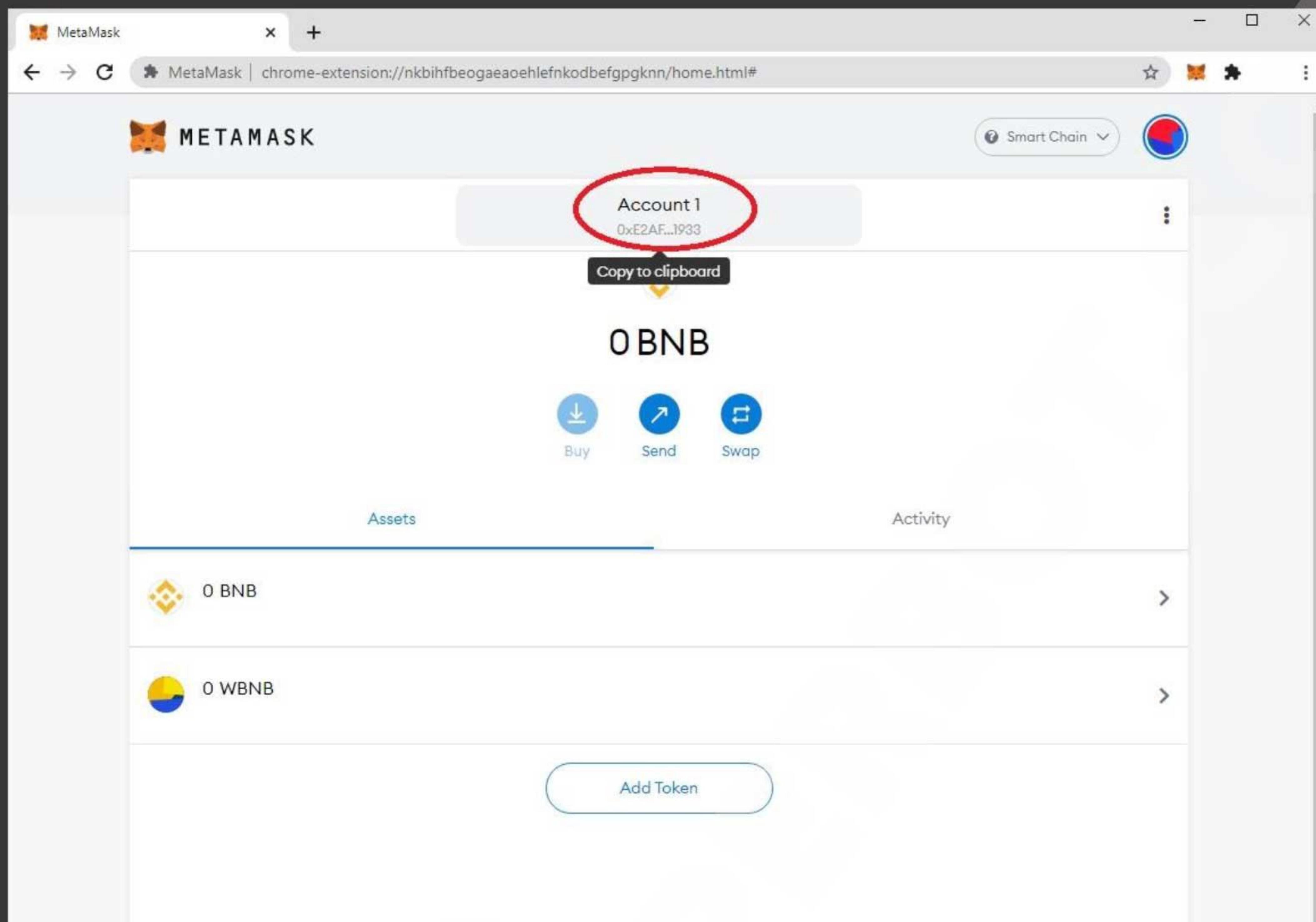
```
JS bot.js ●
JS bot.js > WSS
1 const ethers = require('ethers');
2
3 //*****
4 //ENTER YOUR DETAILS - Refer to instructions!
5
6 //1. Snipe ddetails
7 const SnipeID = "Enter ID of the token to snipe. Refer to instructions for formatting"
8 const Receive = 'Enter ID of the token to buy. Usually the same as SnipeID'
9 const amountIn = ethers.utils.parseUnits('Enter how much WBNB to spend', 'ether')
10
11
12 //2. Wallet / connection details
13 const WSS = "wss://[REDACTED].bsc.quiknode.pro/"
14 const Seed = 'Enter your wallet 12 word seed phrase'
15 const recipientaddress = 'Enter your wallet address where to send the bought tokens'
16
17
18 //3. Optional settings
19 const Spend = '0xbb4Cd89C8d36B01bD1cBaE8F2De08d9173bc095c'
20 const routeraddress = '0x10ED43C718714eb63d5aA57B78B54704E256024E'
21 const Slippage = ethers.utils.parseUnits('0', 'ether')
22
23
24 /////Done. Do NOT change code after this!
25
26
27
28 const MethodID = "0xf305d719" // function ID it searches for (one is for WETH/WBNB pools and other is for pools t
29 const MethodID2 = "0xe8e33700" // function ID it searches for (one is for WETH/WBNB pools and other is for pools
30 const provider = new ethers.providers.WebSocketProvider(WSS);
31 const wallet = ethers.Wallet.fromMnemonic(Seed);
32 const account = wallet.connect(provider);
33 provider.removeAllListeners()
34 const router = new ethers.Contract(
35   routeraddress,
36   ['function swapExactTokensForTokens(uint amountIn, uint amountOutMin, address[] calldata path, address to, uint
37   account
38 );
39 console.log(`Connecting to the blockchain`),
40 console.log(`Starting to scan the network for a matching transaction based on the config entered`),
41 console.log(`As soon as a matching transaction has been found, it will be displayed here`),
42 provider.on("pending", async (tx) => {
43   provider.getTransaction(tx).then(function (transaction){
44     if (transaction != null && transaction['data'].includes(MethodID2) && transaction['data'].includes(SnipeID) || tra
45     console.log(transaction),
46     console.log(` Matching liquidity add transaction found!`),
47   })
48 })
```

2. Enter your 12 word seed phrase (that you wrote down earlier) from your MetaMask wallet into the “const Seed” line ”. In this format – ‘word word word’



```
JS bot.js ●
JS bot.js > Seed
1 const ethers = require('ethers');
2
3 //*****
4 //ENTER YOUR DETAILS - Refer to instructions!
5
6 //1. Snipe details
7 const SnipeID = "Enter ID of the token to snipe. Refer to instructions for formatting"
8 const Receive = 'Enter ID of the token to buy. Usually the same as SnipeID'
9 const amountIn = ethers.utils.parseUnits('Enter how much WBNB to spend', 'ether')
10
11
12 //2. Wallet / connection details
13 const WSS = "wss://[REDACTED].bsc.quiknode.pro/"
14 const Seed = 'judicious wander advice deadpan flower pop'
15 const recipientaddress = 'Enter your wallet address where to send the bought tokens'
16
17
18 //3. Optional settings
19 const Spend = '0xbb4Cd89C8d36B01bD1cBaE8F2De08d9173bc095c'
20 const routeraddress = '0x10ED43C718714eb63d5aA57B78B54704E256024E'
21 const Slippage = ethers.utils.parseUnits('0', 'ether')
22
23
24 /////Done. Do NOT change code after this!
25
26
27
28 const MethodID = "0xf305d719" // function ID it searches for (one is for WETH/WBNB pools and other is for pools that us
29 const MethodID2 = "0xe8e33700" // function ID it searches for (one is for WETH/WBNB pools and other is for pools that u
30 const provider = new ethers.providers.WebSocketProvider(WSS);
31 const wallet = ethers.Wallet.fromMnemonic(Seed);
32 const account = wallet.connect(provider);
33 provider.removeAllListeners()
34 const router = new ethers.Contract(
35   routeraddress,
36   ['function swapExactTokensForTokens(uint amountIn, uint amountOutMin, address[] calldata path, address to, uint deadl
37   account
38 );
39 console.log(`Connecting to the blockchain`),
40 console.log(`Starting to scan the network for a matching transaction based on the config entered`),
41 console.log(`As soon as a matching transaction has been found, it will be displayed here`),
42 provider.on("pending", async (tx) => {
43   provider.getTransaction(tx).then(function (transaction){
44     if (transaction != null && transaction['data'].includes(MethodID2) && transaction['data'].includes(SnipeID) || tra
45     console.log(transaction),
46     console.log(` Matching liquidity add transaction found!`),
47   })
48 })
```

- Copy your wallet address from MetaMask and paste it in the “const recipientaddress” line



The screenshot shows a code editor with a dark theme. The file is named "bot.js" and is part of a project titled "PANCAKESNIPER V2.0". The code is a script for a bot, likely for interacting with a decentralized exchange. A red box highlights the line of code where the recipient address is defined: "const recipientaddress = '0xb8379'".

```

1 // *****
2 //ENTER YOUR DETAILS - Refer to Instructions!
3
4 //1. Snipe details
5 const SnipeID = "Enter ID of the token to snipe. Refer to instructions for formatting"
6 const Receive = 'Enter ID of the token to buy. Usually the same as SnipeID'
7 const amountIn = ethers.utils.parseUnits('Enter how much WBNB to spend', 'ether')
8
9 //2. Wallet / connection details
10 const WSS = "Ws://[REDACTED].bsc.quiknode.pro/[REDACTED]"
11 const Seed = 'judicious wander advice deadpan flower pop[REDACTED]1'
12 const recipientaddress = '0xb8379'
13
14 //3. Optional settings
15 const Spend = '0xbb4C089C8d36B01b01cBaEBF2De08d9173bc095c'
16 const routeraddress = '0x10ED43C718714eb63d5aA57B78B54704E256024E'
17 const Slippage = ethers.utils.parseUnits('0', 'ether')
18
19 //****Done. Do NOT change code after this!
20
21
22
23
24
25
26
27
28 const MethodID = "0xf305d719" // function ID it searches for (one is for WETH/WBNB pools and other is for pools that us
29 const MethodID2 = "0xe8e33700" // function ID it searches for (one is for WETH/WBNB pools and other is for pools that u
30 const provider = new ethers.providers.WebSocketProvider(WSS);
31 const wallet = ethers.Wallet.fromMnemonic(Seed);
32 const account = wallet.connect(provider);
33 provider.removeAllListeners();
34 const router = new ethers.Contract(
35   routeraddress,
36   ['function swapExactTokensForTokens(uint amountIn, uint amountOutMin, address[] calldata path, address to, uint deadline) external'],
37   account
38 );
39 console.log(`Connecting to the blockchain`),
40 console.log(`Starting to scan the network for a matching transaction based on the config entered`),
41 console.log(`As soon as a matching transaction has been found, it will be displayed here`),
42 provider.on("pending", async (tx) => {
43   provider.getTransaction(tx).then(function (transaction){
44     if (transaction != null && transaction['data'].includes(MethodID2) && transaction['data'].includes(SnipeID) || tra
45     console.log(transaction),
46     console.log(`Matching liquidity add transaction found`),
47     router.swapExactTokensForTokens(

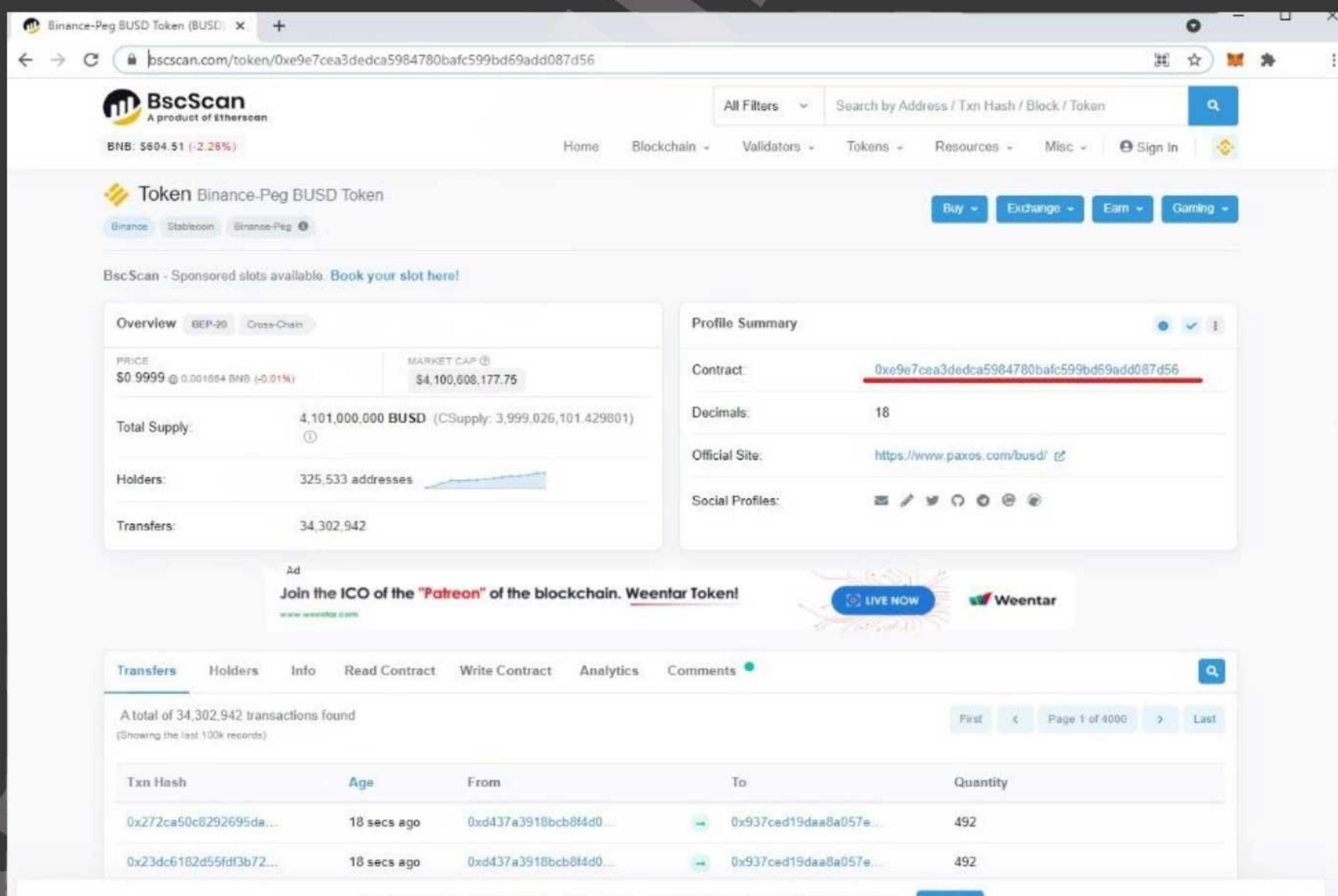
```

## Setting up Snipe details

For this example we are going to look for liquidity add transaction for BUSD and also make it buy BUSD once such a transaction is found. That is because there are not new projects currently. Use your sniping token details for a real snipe. I suggest trying first test-snipe on BUSD as well to make sure that the entered wallet details are correct and the bot works.

1. Find contract address (usually you can find address in the tokens official home page or twitter or telegram account) in this case we are going to find the BUSD contract address from [www.bscscan.com](https://www.bscscan.com)

\*NOTE check the address legitimacy in [www.bscscan.com](https://www.bscscan.com)



The screenshot shows the BscScan token details page for Binance-Peg BUSD Token. The contract address is highlighted in red as 0xe9e7cea3dedca5984780bafc599bd69add087d56. Other visible details include a price of \$0.9999, a market cap of \$4,101,000.000 BUSD, and a total supply of 4,101,000,000 BUSD. The page also displays a list of recent transfers.

- Copy and paste address in the “const Receive” line – the token to make it buy.

```
//ENTER YOUR DETAILS - Refer to instructions!  
  
//1. Snipe details  
const SnipeID = "Enter ID of the token to snipe. Refer to instructions for formatting"  
const Receive = '0xe9e7cea3dedca5984780bafc599bd69add087d56'  
const amountIn = ethers.utils.parseUnits('Enter how much WBNB to spend', 'ether')
```

2. In the “const SnipeID ” paste the same line and delete “0x” – the token for it to look for

```
//ENTER YOUR DETAILS - Refer to instructions!

//1. Snipe details
const SnipeID = "e9e7cea3dedca5984780bafc599bd69add087d56"
const Receive = '0xe9e7cea3dedca5984780bafc599bd69add087d56'
const amountIn = ethers.utils.parseUnits('Enter how much WBNB to spend', 'ether')
```

\*NOTE If the contract address has capital letters change them to small letters. Remove 0x

For example

0xE9e7cEA3dedca5984780baFC599bd69add087d56

To:

e9e7cea3dedca5984780bafc599bd69add087d56

This action is only required for SnipeID.

3. Next in the “const amountIN” enter the WBNB amount that you want to spend. **WBNB is needed for buying the token.** Its 1:1 to regular BNB. Make sure you have enough. The amount needs to be equal or less then the amount in the MetaMask wallet. Regular BNB will be needed for fees and you also need that in your wallet.

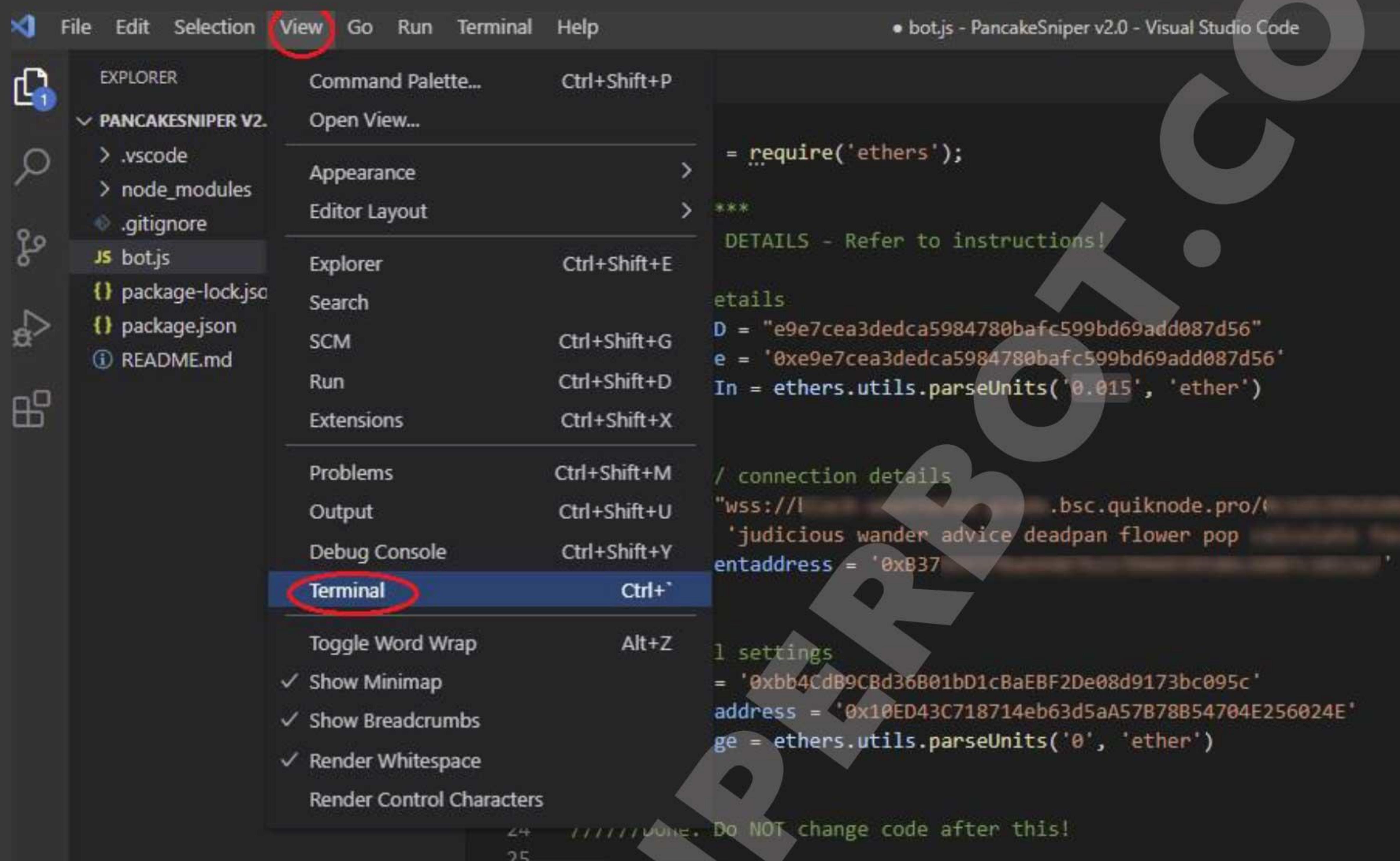
\*NOTE Do NOT change the word ‘ether’ even if sniping with WBNB. It is used for formatting.

```
//ENTER YOUR DETAILS - Refer to instructions!

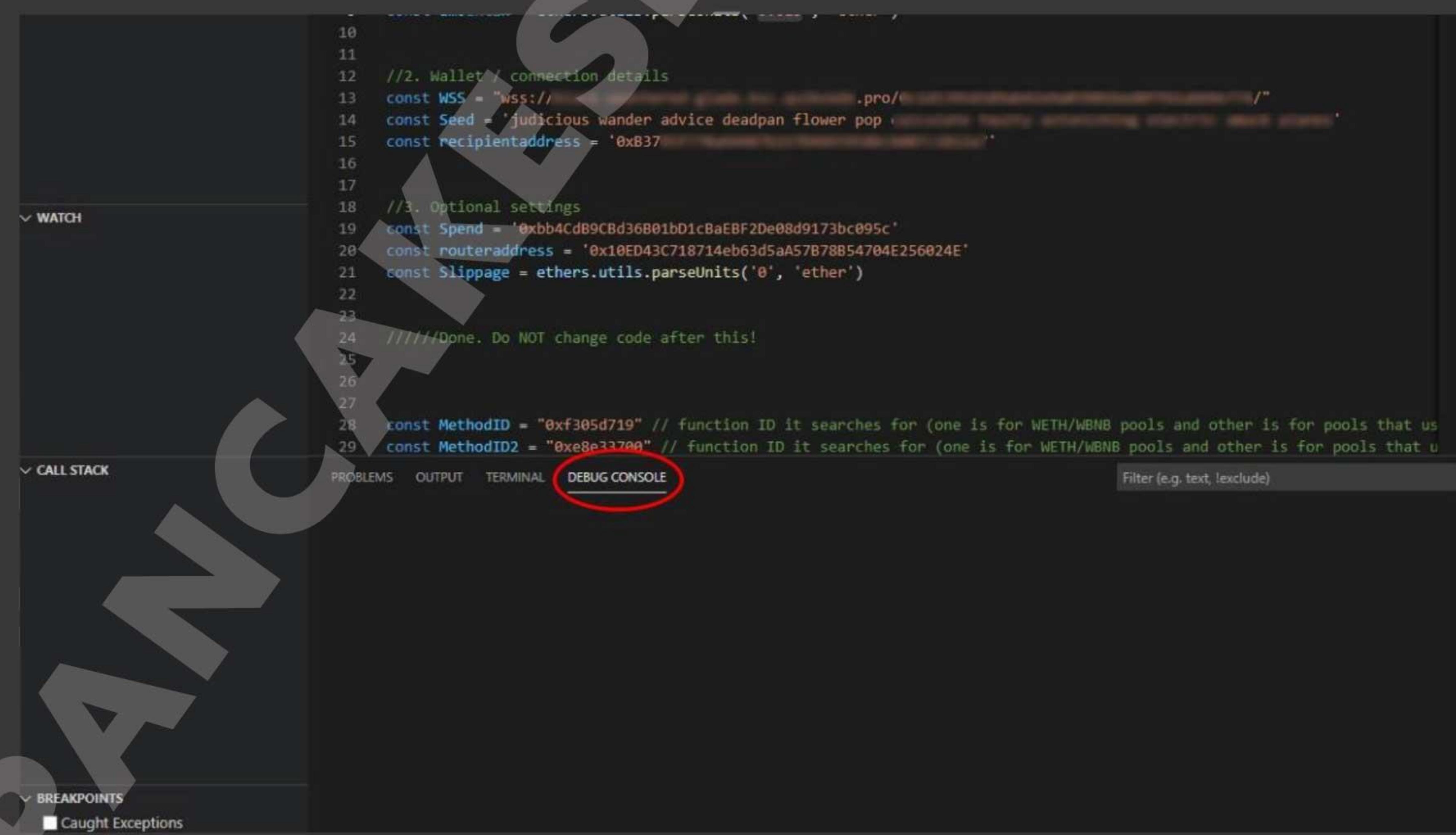
//1. Snipe details
const SnipeID = "e9e7cea3dedca5984780bafc599bd69add087d56"
const Receive = '0xe9e7cea3dedca5984780bafc599bd69add087d56'
const amountIn = ethers.utils.parseUnits('0.015', 'ether')
```

## Launching Bot

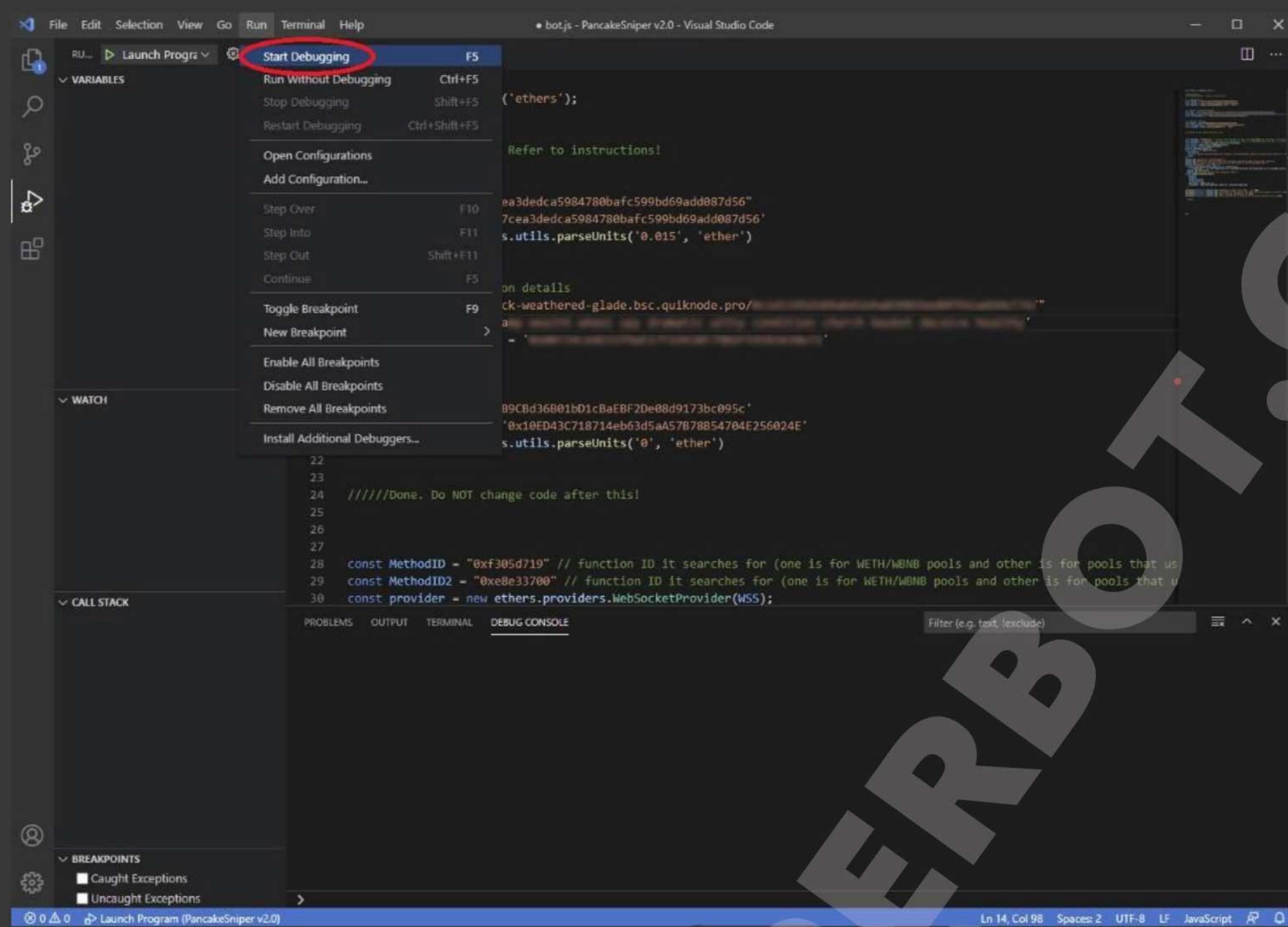
1. Click on View -> Terminal



- Click on Debug Console

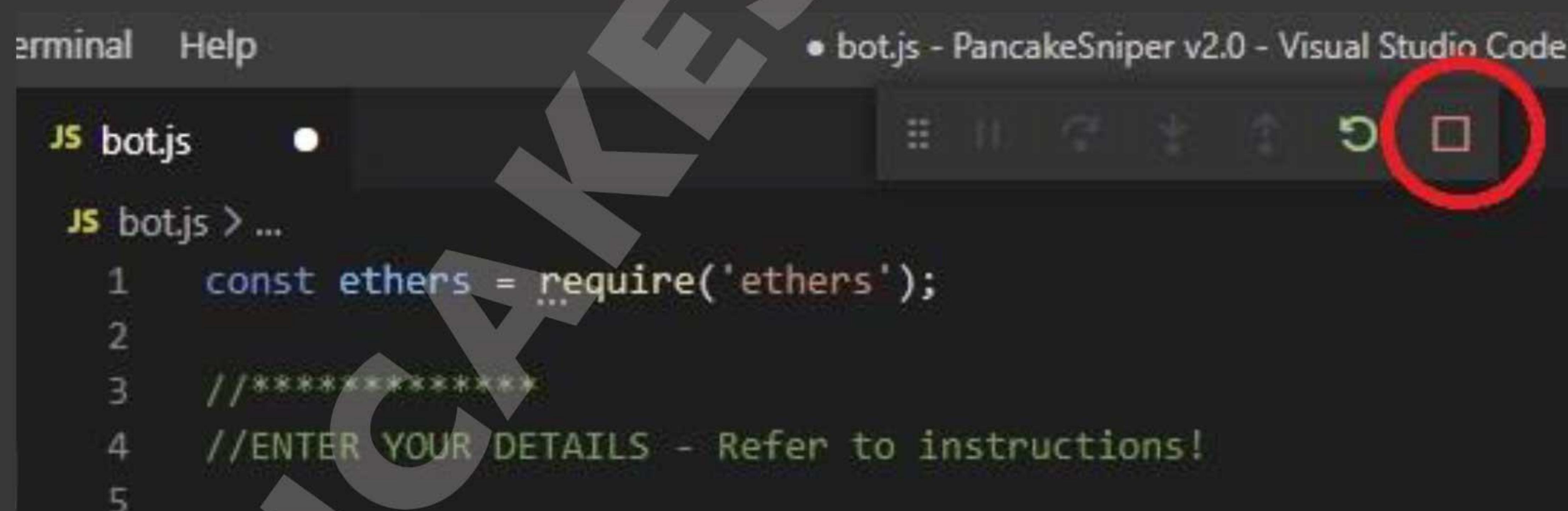


2. Click Run -> Start Debugging



It should start Sniping

- If the control bar appears on top of the page that means the bot is started
- If you want to stop the bot click the red square



3. After the Snipe check your transaction in [www.bscscan.com](https://www.bscscan.com)

- Paste your MetaMask wallet address in BscScan search bar

The screenshot shows the BscScan interface for the address 0x6e80. The top navigation bar includes links for Home, Blockchain, Validators, Tokens, Resources, Misc, and Sign In. A search bar at the top right contains the address 0x6e80. Below the header, there's a banner for a Merchant Token ICO. The main content area has two sections: 'Overview' on the left and 'More Info' on the right. The Overview section displays the balance (0.00000000 BNB), BNB Value (\$121.77), and a dropdown for Token selection. The More Info section shows a placeholder for a name tag. Below these, a table lists the latest 14 transactions from a total of 14. The table columns include Txn Hash, Block, Age, From, To, Value, and Txn Fee. The transactions show various BNB movements between the address 0x6e80 and other accounts.

Txn Hash	Block	Age	From	To	Value	Txn Fee
0x520b334bf3a37e164...	7071228	14 secs ago	0x6e80	OUT	0 BNB	0.00240343
0x11594ccc60f5e2067...	7070642	29 mins ago	0x6e80	OUT	0 BNB	0.00072
0x7febaef740a5da1c4f...	6730396	11 days 23 hrs ago	0x6e80	OUT	0 BNB	0.00022165
0x666e55854db5f2209f...	6728626	12 days 41 mins ago	0x6e80	OUT	0.21465981 BNB	0.00196877
0xaea7a1bfdf0fc6f015e...	6715989	12 days 11 hrs ago	0x6e80	OUT	0 BNB	0.00022274

Please read Most common problems (page 22) and (page 23)

## What means what

This where you edit info

```
JS bot.js > [e] Slippage
1  const ethers = require('ethers');
2
3  //*****
4  //ENTER YOUR DETAILS - Refer to instructions!
5
6  //1. Snipe details
7  const SnipeID = "e9e7cea3dedca5984780bafc599bd69add087d56"
8  const Receive = '0xe9e7cea3dedca5984780bafc599bd69add087d56'
9  const amountIn = ethers.utils.parseUnits('0.015', 'ether')
10
11
12 //2. Wallet / connection details
13 const WSS = "wss://plan-plan-plan.bsc.quiknode.pro/0c1ahd5836ty0042e9a03901bbw0abt763474/"
14 const Seed = 'fall command freezing explain leg stir command breath pretend leg lick fall'
15 const recipientaddress = '0x0000000000111111222222333333'
16
17
18 //3. Optional settings
19 const Spend = '0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c'
20 const routeraddress = '0x10ED43C718714eb63d5aA57B78B54704E256024E'
21 const Slippage = ethers.utils.parseUnits('0', 'ether')
22
23
24 /////Done. Do NOT change code after this!
25
```

### 1. Snipe details

```
6  //1. Snipe details
7  const SnipeID = "e9e7cea3dedca5984780bafc599bd69add087d56"
8  const Receive = '0xe9e7cea3dedca5984780bafc599bd69add087d56'
9  const amountIn = ethers.utils.parseUnits('0.015', 'ether')
10
```

**const SnipeID** – this is contract address without “0x” and capital letters. It used find the token you are looking to snipe. It will find the liquidity add transaction for this address

**const Receive** – this is the contract address is the token to buy when the snipe ID is found in the MEMPOOL

**const amountIN** – the amount in WBNB which you want to spend. Don’t change ‘ether’

## 2. Wallet/connection details

```
11
12 //2. Wallet / connection details
13 const WSS = "wss://plan-plan-plan.bsc.quiknode.pro/0c1ahd5836ty0042e9a03901bbw0abt763474/"
14 const Seed = 'fall command freezing explain leg stir command breath pretend leg lick fall'
15 const recipientaddress = '0x000000000011111112222222333333'
16
```

**const WSS** – this is a quiknode address that is used to connect to the blockchain network

**const Seed** – this is where you enter your seed phrase from your MetaMask wallet. It is necessary so that the bot could spend the money from your MetaMask wallet to buy tokens you snipe

**const recipientaddress** – this is the address where the bot sends the bought tokens

## 3. Optional Settings

```
18 //3. Optional settings
19 const Spend = '0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c'
20 const routeraddress = '0x10ED43C718714eb63d5aA57B78B54704E256024E'
21 const Slippage = ethers.utils.parseUnits('0', 'ether')
22
```

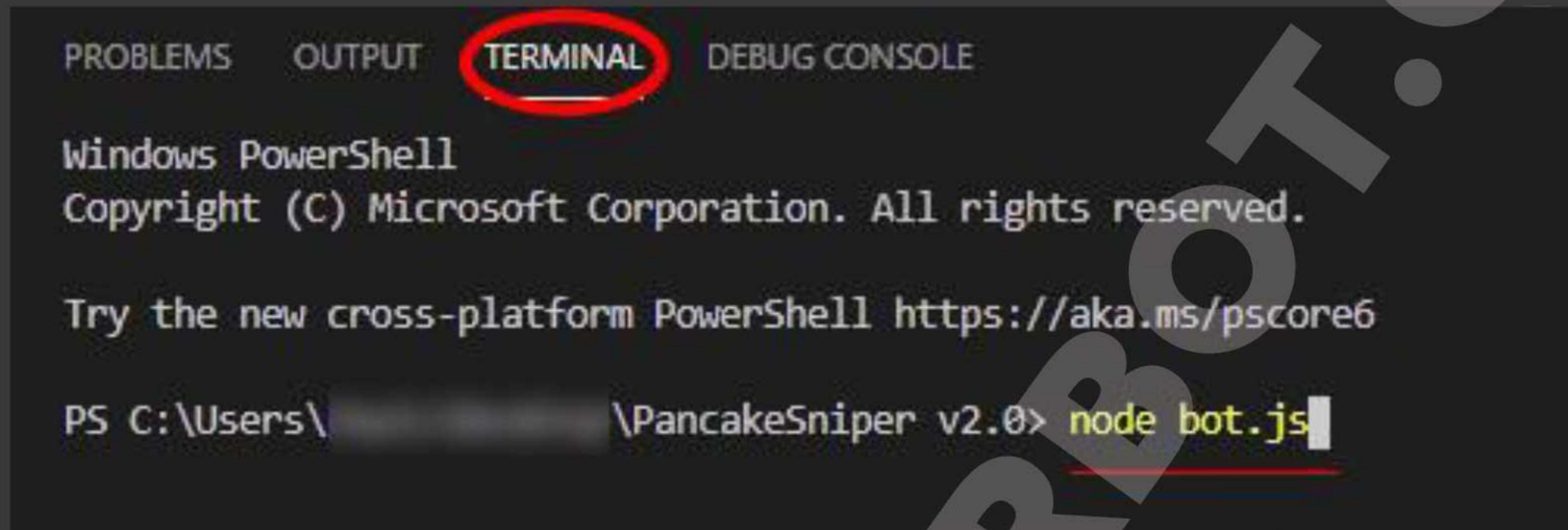
**const Spend** – coin address used for paying. In 99% cases this will be WBNB. Please do not change this address

**const routeraddress** - Pancakeswap V2 router address. If they update to V3 you can change this to the new one

**const Slippage** – actually the minimum output setting. By default set at 0. It means that the buy order will be made if you can get at least 0 of the specified tokens you are sniping with the WBNB you set at the **const amountIN**. It will buy as many tokens as possible. For example: If you set it at 85, it will only buy if you can get 85 or more of the tokens. Don't change 'ether'

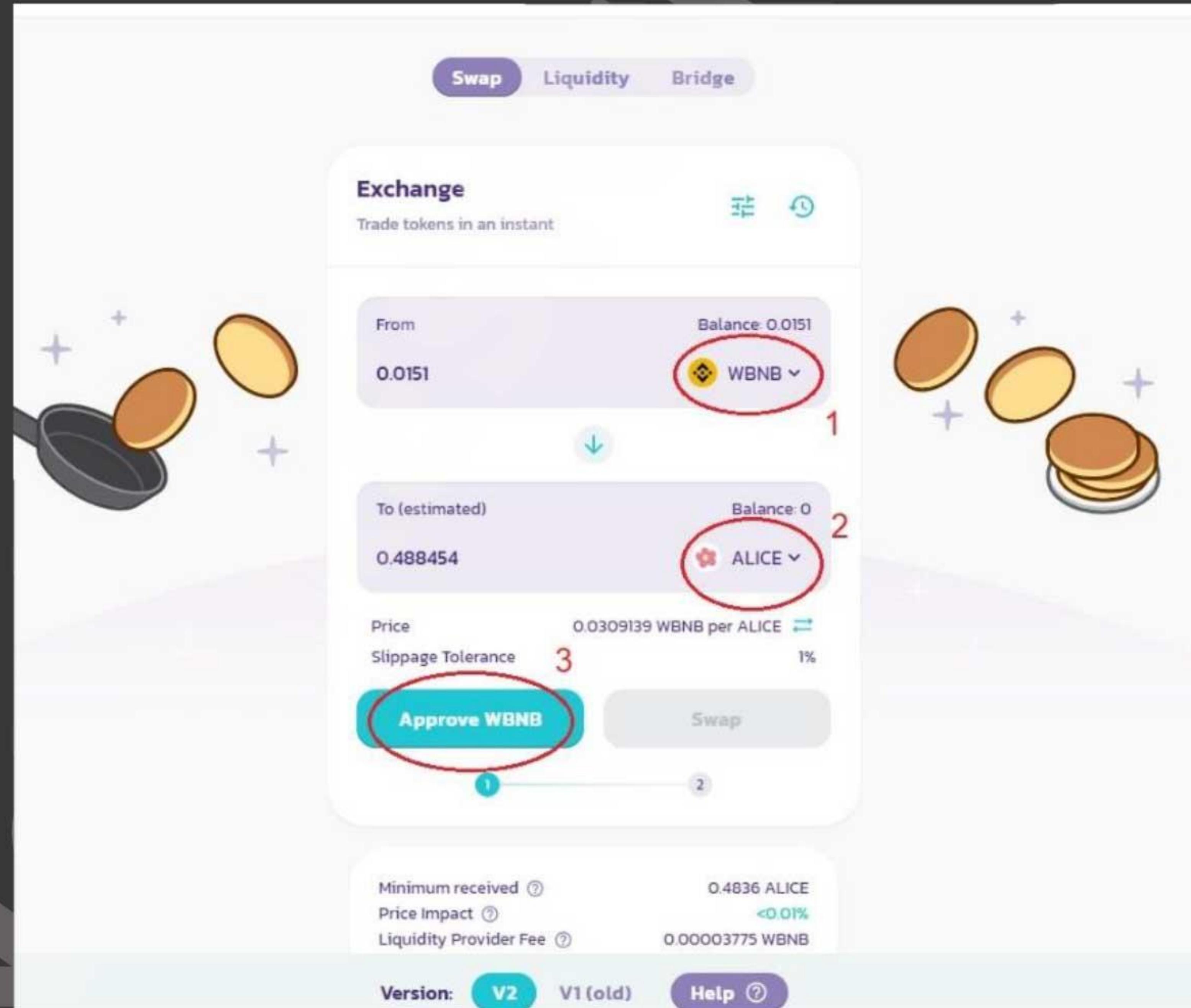
## Most common problems

1. First time launching the bot it won't start
  - o Check and make sure that every information is typed in correctly. The slightest typing mistake could cause the bot to malfunction.
  - o Click TERMINAL and write---- node bot.js



A screenshot of a Windows PowerShell terminal window. The title bar shows tabs for PROBLEMS, OUTPUT, TERMINAL (which is highlighted with a red oval), and DEBUG CONSOLE. The main area displays the PowerShell prompt and some introductory text. At the bottom, the command `node bot.js` is typed into the input field.

2. WBNB is not approved
- Bot uses WBNB for buying tokens. Make sure you have it and that its approved.

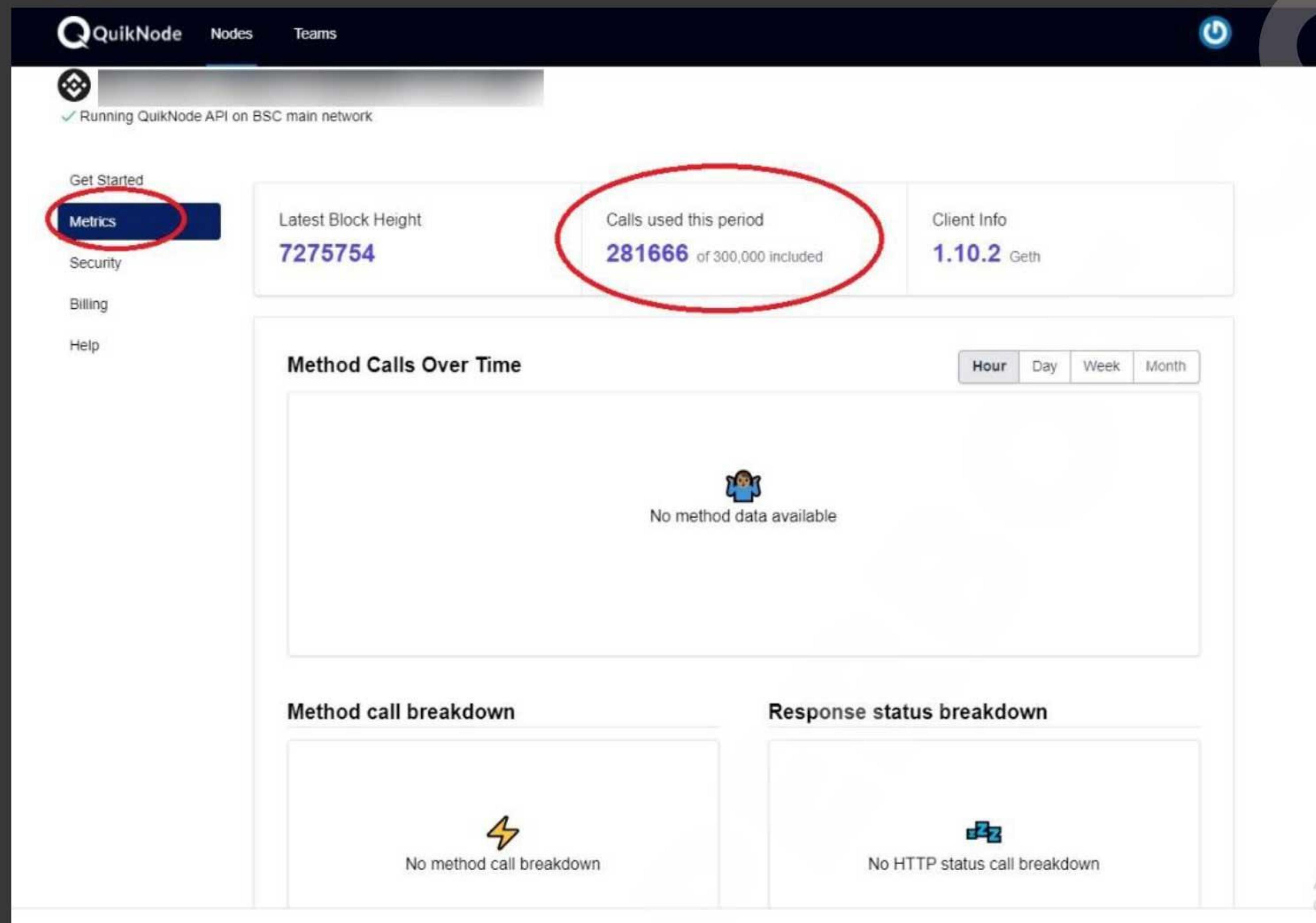


3. Use a single Metamask Wallet

Make sure that Metamask has a single wallet. Other accounts may affect the bot. For some occasion you may have to delete other accounts or uninstall Chrome and install again to clean Metamask data.

#### 4. Quicknode limit

Make sure you still have Quiknode limit left. You can check it by going into QuikNode account, open **Metrics** and see “**Calls used this period**”



#### 5. Don't ask stupid questions

“Maybe you can help me out with that in any way? I have no CreditCard”

“Can I send the code to others?”

“Does it work on windows or windows?”

“In what color should I buy my Lambo?”

“Will a vaccine affect my sniping?” etc.

# PLUS VERSION

Once you have setup the BASIC, proceed with Plus.

Here are the extra instructions for Plus version after you setup the same as basic from the included pdf instructions

Copy your wallet id.

Go inside bscscan.com and search for your wallet

Check the last OUT transaction and click the TX so it open more info

When you click Show more on the bottom, you will see the Nonce setting

There will be a number which indicates the number of the outgoing transactions from your wallet.

For example if you have made like 20 transfers from your wallet, then it will show number 20

Then go inside the bot.js file for the Plus sniper.

Scroll slightly down until you see the first SwapExactTokensforTokens (buy) function ( there will be a total of 15)

On the bottom of that function you will see setting Nonce:

If the last nonce number from bscscan is 20, then you enter 21 in the first buy function. The next will be 22. Third will be 23 and so on.

It needs this setting because it shoots the 15 times so fast that it cannot calculate whch was the last transaction. So you have to give this info manually

Make sure to not make any other transactions when you set the bot.js file up. Otherwise the count will again change

You have to do this before each Plus sniper snipe. Because the number will change when you make 15 OUT buy transactions

That is all!!!