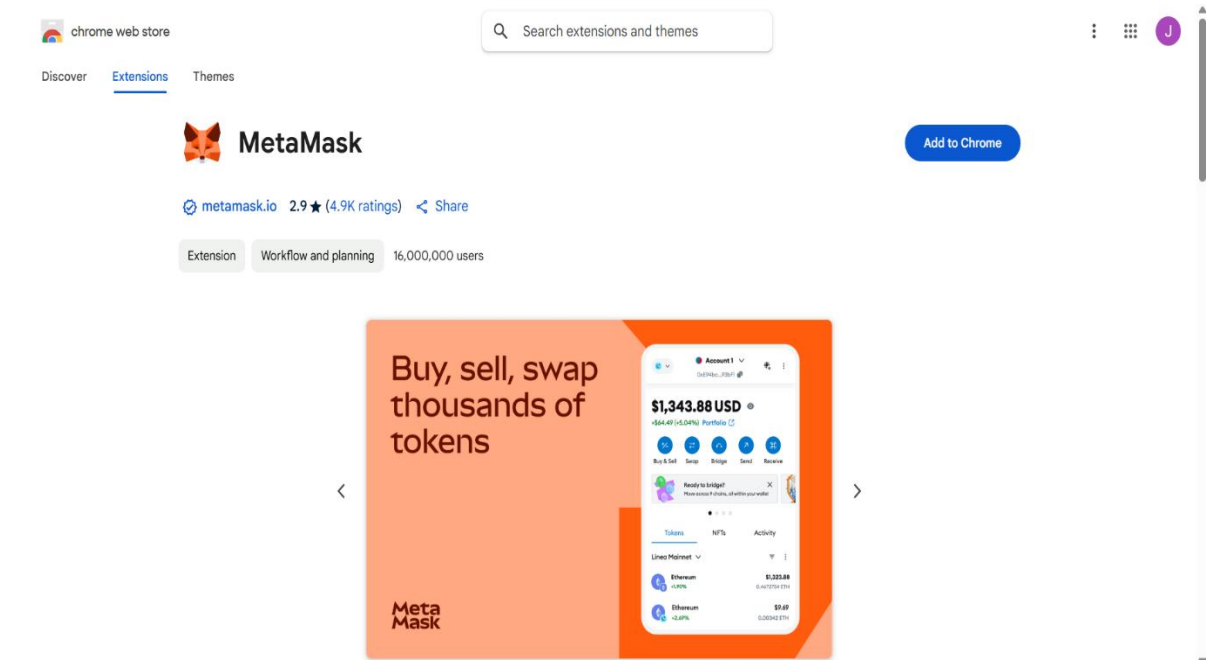
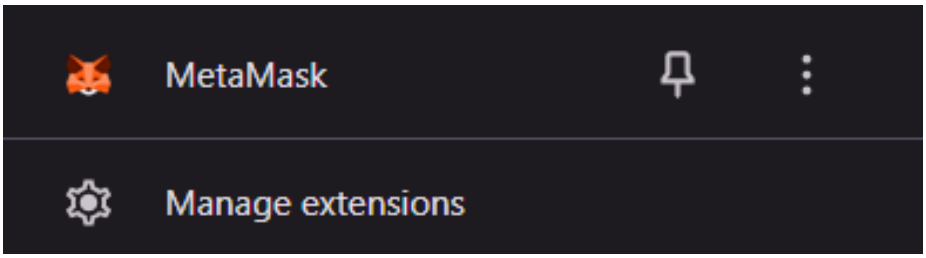
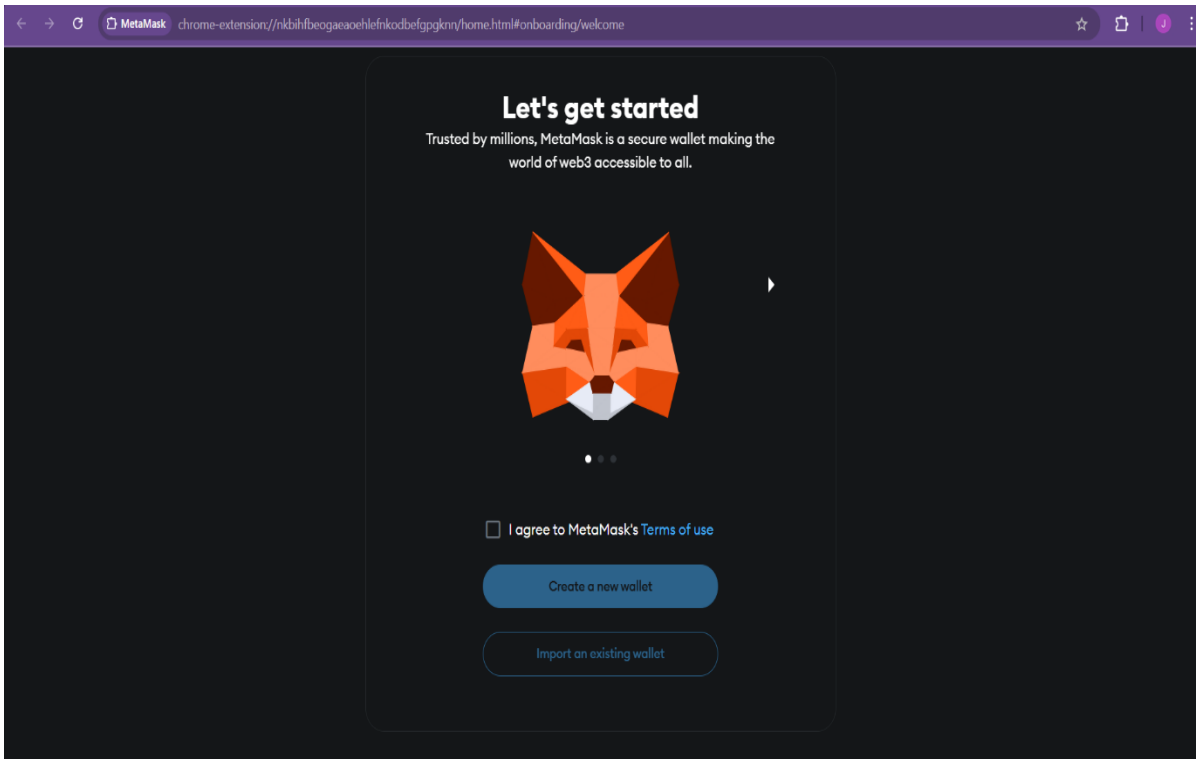
Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

Blockchain Metamask Transaction

Step 1: Add the metamask chrome Extension in the Chrome and open it.

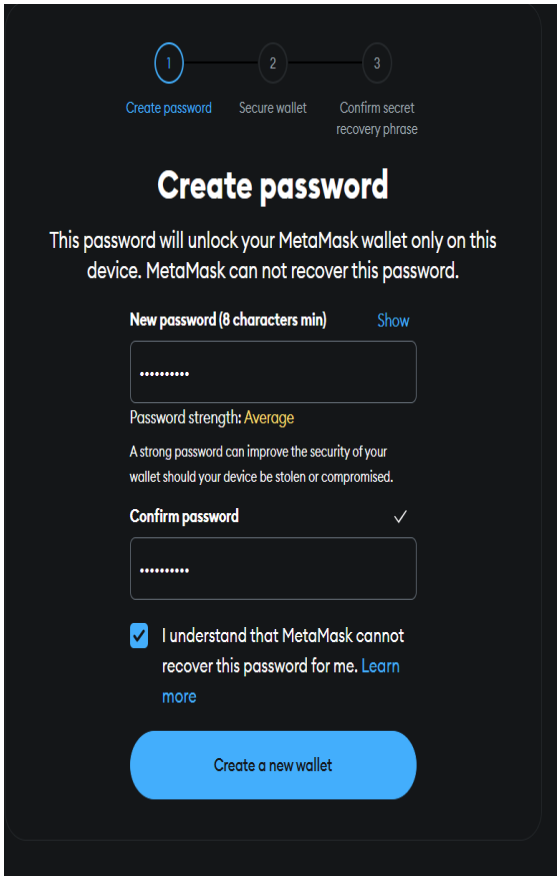


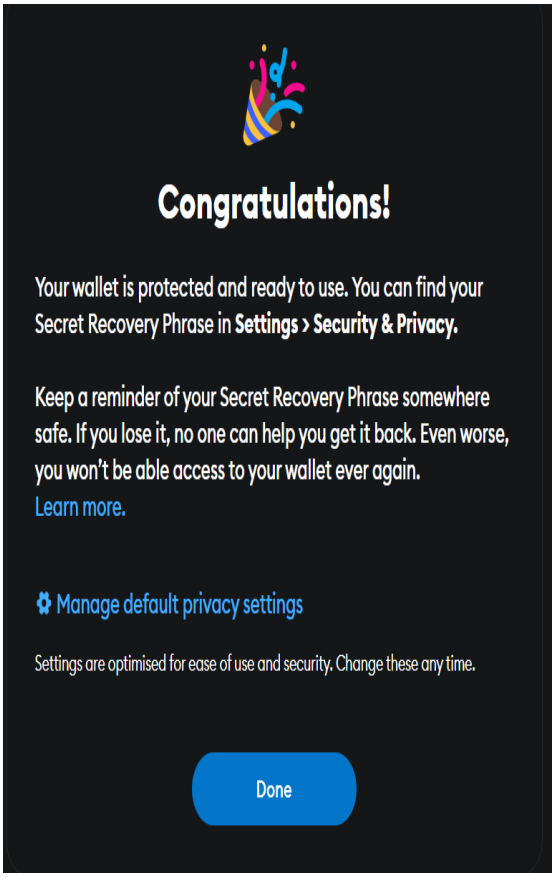




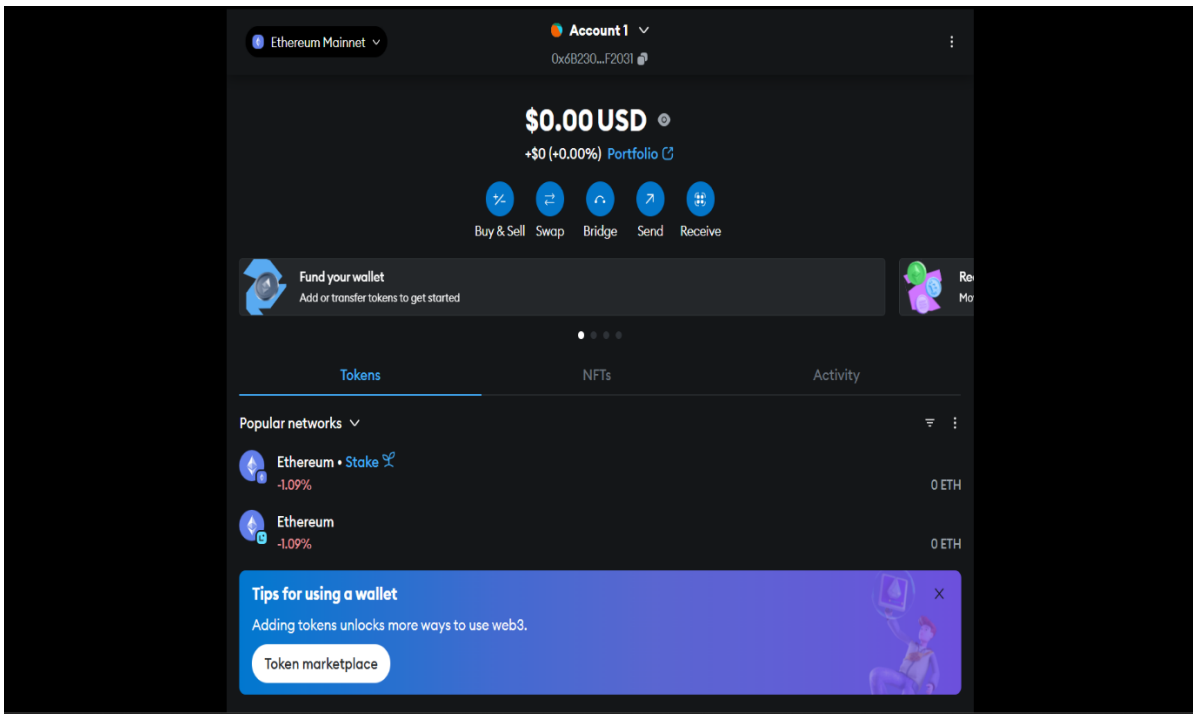
Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

Step 2: Now create Your MetaMask Account by entering all required details.



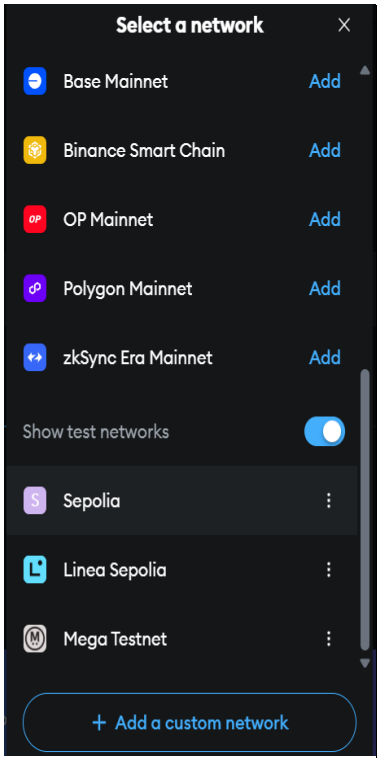


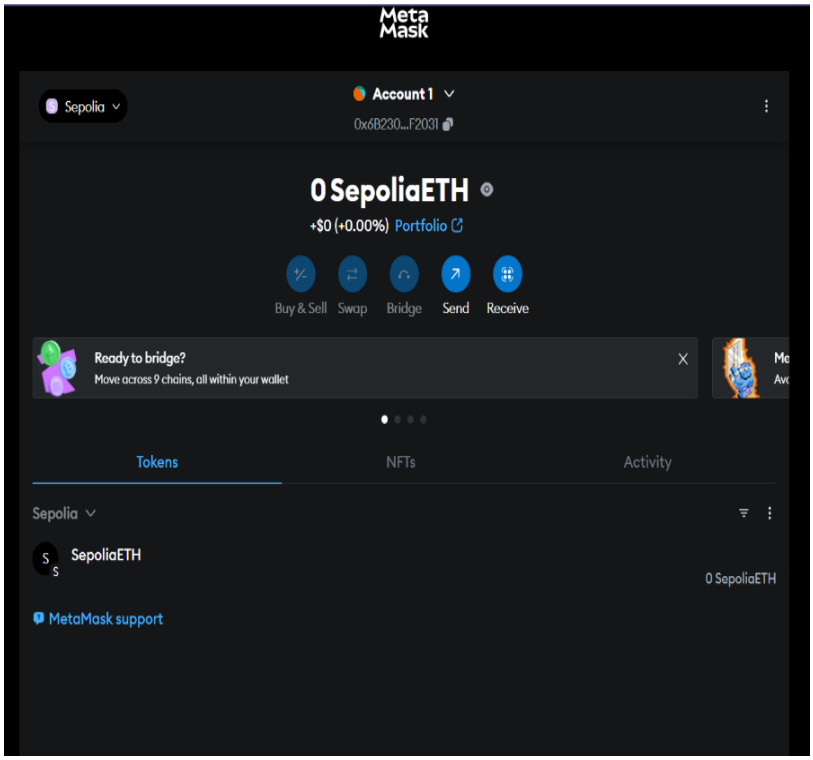
Step 3: Now MetaMask will open the dashboard of it in Ethereum Mainnet Network .



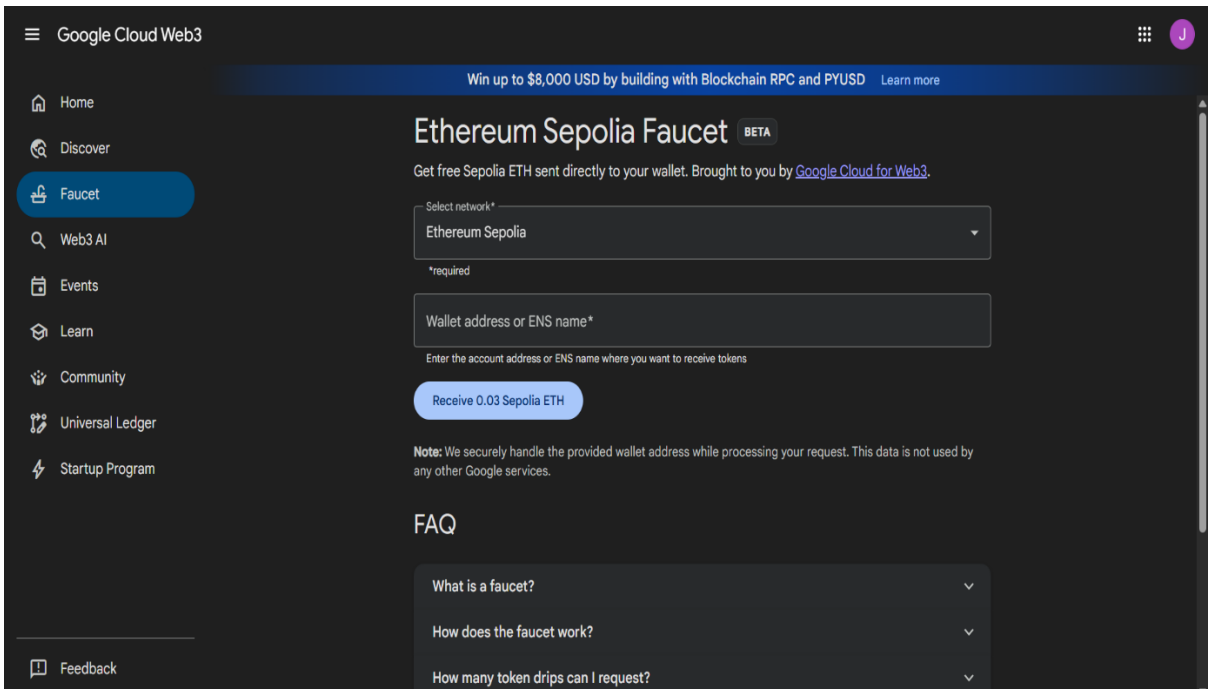
Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

Step 4: Now you have to change the network to the sepolia Network, then it will open the dashboard of the sepolia Network in MetaMask.



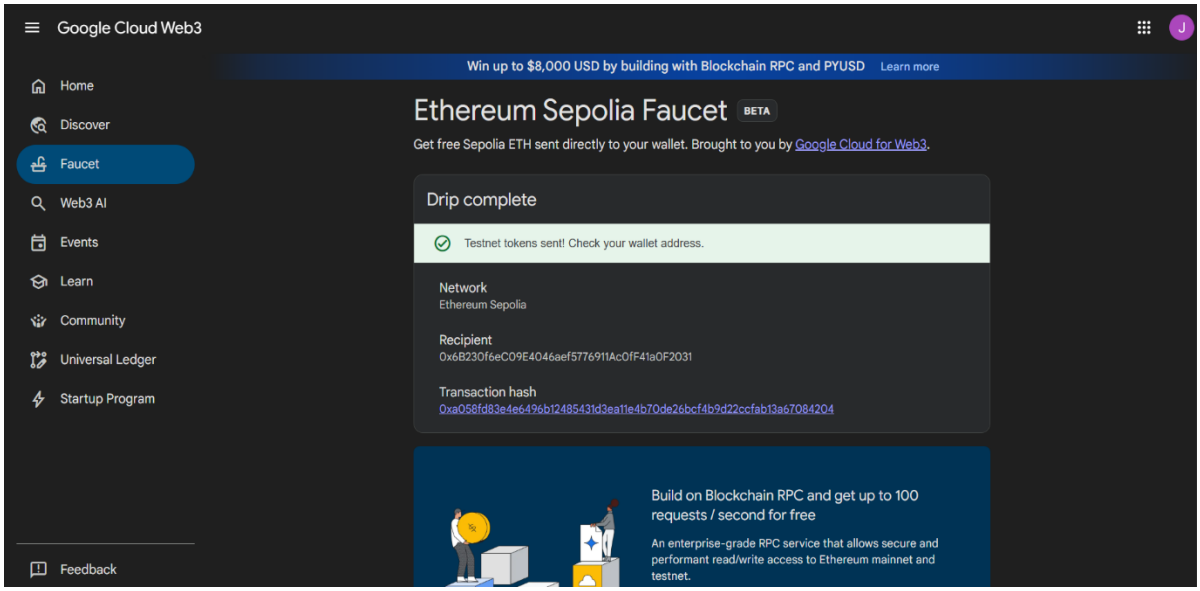


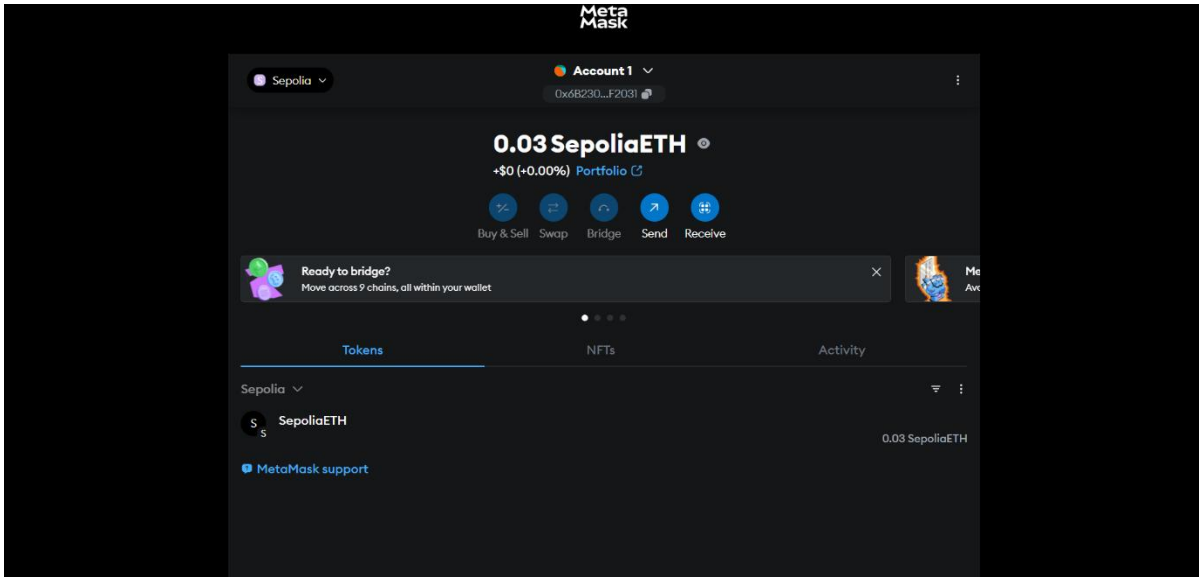
Step 5: Now you have to search for the sepolia Faucet in the browser and open the Ethereum Sepolia Faucet by google cloud to get the free Ether.



Step 6: Then you have to copy the address of your sepolia MetaMask account and paste it in the website and then click on receive 0.03 Sepolia ETH. Then it will show the confirmation of the sent ether, that we can also check in our Sepolia MetaMask account.

Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A





Step 7: We have ether in our sefolia MetaMask account, now we will Compile and deploy our smart Contract to do the transaction.

Step 8: We will now open the Remix Ethereum IDE in the browser, in that we will create a new workspace BlockChain and in that workspace we will create the contract.sol file in which our smart Contract code will be added.

Step 9: I have added the smart Contract Code on Blockchain-Based Crowdfunding Platform in the contract.sol file.

Code:

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;

contract CrowdfundingPlatform {

    struct Campaign {
```

Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

```
    uint256 id;
    address payable creator;
    uint256 goal;
    uint256 deadline;
    uint256 fundsRaised;
    bool goalReached;
    bool fundsWithdrawn;
}
mapping(uint256 => Campaign) public campaigns;
mapping(uint256 => mapping(address => uint256)) public contributions;
uint256 public campaignCount;
event CampaignCreated(
    uint256 id,
    address creator,
    uint256 goal,
    uint256 deadline
);
event ContributionReceived(uint256 id, address contributor, uint256 amount);
event FundsWithdrawn(uint256 id, address creator, uint256 amount);
event RefundIssued(uint256 id, address contributor, uint256 amount);
modifier campaignExists(uint256 _id) {
    require(campaigns[_id].id == _id, "Campaign does not exist");
    _;
}
modifier onlyCreator(uint256 _id) {
    require(
        msg.sender == campaigns[_id].creator,
        "Only creator can withdraw funds"
    );
    _;
```

Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

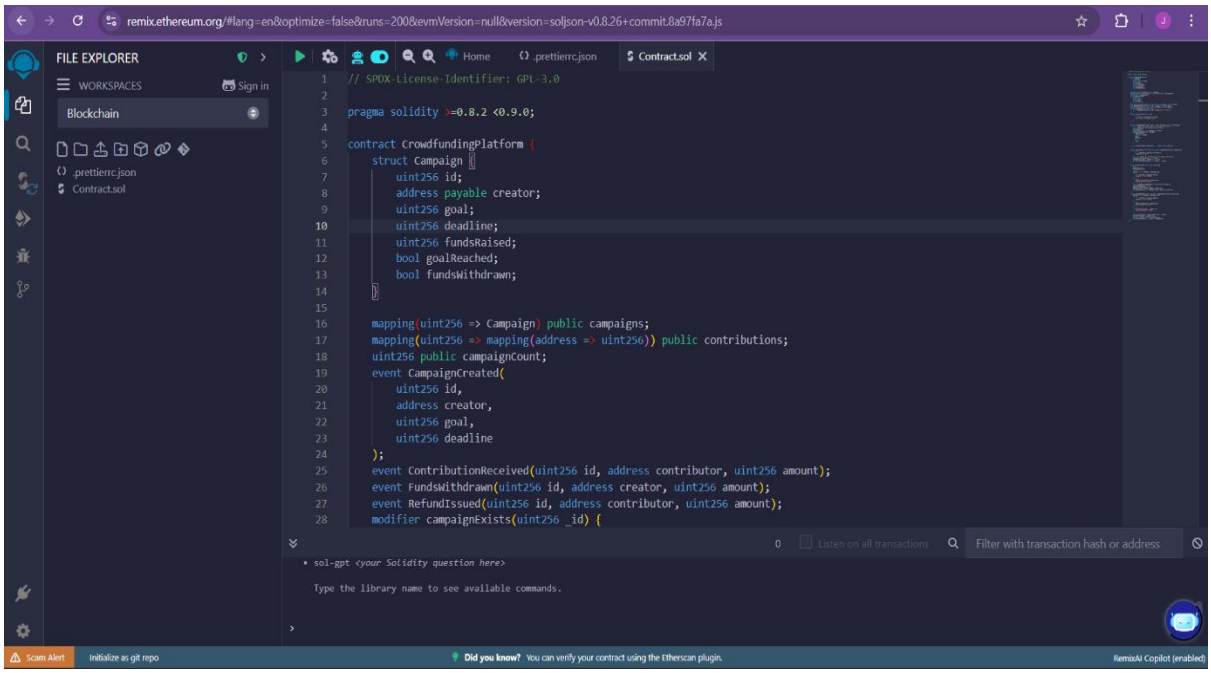
```
}
function createCampaign(uint256 _goal, uint256 _duration) public {
    require(_goal > 0, "Funding goal must be greater than zero");
    require(_duration > 0, "Duration must be valid");
    campaignCount++;
    uint256 deadline = block.timestamp + _duration;
    campaigns[campaignCount] = Campaign(
        campaignCount,
        payable(msg.sender),
        _goal,
        deadline,
        0,
        false,
        false
    );
    emit CampaignCreated(campaignCount, msg.sender, _goal, deadline);
}
function contribute(uint256 _id) public payable campaignExists(_id) {
    require(
        block.timestamp < campaigns[_id].deadline,
        "Campaign has ended"
    );
    require(msg.value > 0, "Contribution must be greater than zero");
    campaigns[_id].fundsRaised += msg.value;
    contributions[_id][msg.sender] += msg.value;
    emit ContributionReceived(_id, msg.sender, msg.value);
}
function withdrawFunds(uint256 _id)
    public
    onlyCreator(_id)
```

Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

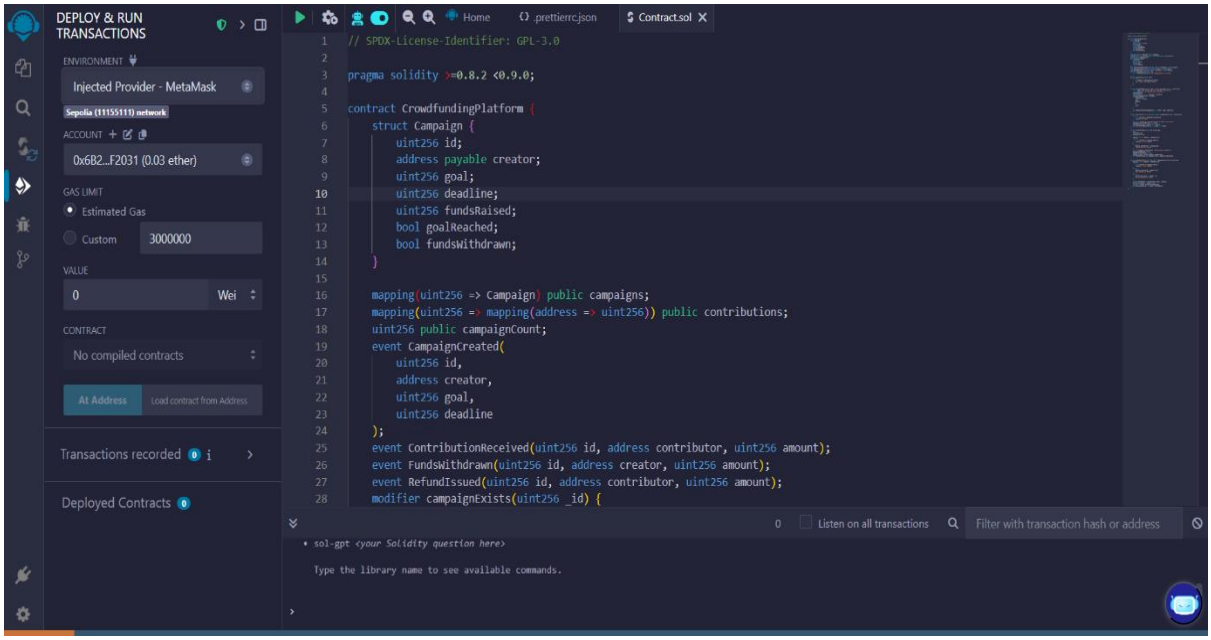
```
    campaignExists(_id)
{
    Campaign storage campaign = campaigns[_id];
    require(
        block.timestamp >= campaign.deadline,
        "Campaign is still ongoing"
    );
    require(
        campaign.fundsRaised >= campaign.goal,
        "Funding goal not reached"
    );
    require(!campaign.fundsWithdrawn, "Funds already withdrawn");
    campaign.goalReached = true;
    campaign.fundsWithdrawn = true;
    campaign.creator.transfer(campaign.fundsRaised);
    emit FundsWithdrawn(_id, campaign.creator, campaign.fundsRaised);
}
function requestRefund(uint256 _id) public campaignExists(_id) {
    Campaign storage campaign = campaigns[_id];
    require(
        block.timestamp >= campaign.deadline,
        "Campaign is still ongoing"
    );
    require(
        campaign.fundsRaised < campaign.goal,
        "Goal reached, no refunds"
    );
    require(
        contributions[_id][msg.sender] > 0,
        "No contributions to refund"
```

Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

```
        );
        uint256 refundAmount = contributions[_id][msg.sender];
        contributions[_id][msg.sender] = 0;
        payable(msg.sender).transfer(refundAmount);
        emit RefundIssued(_id, msg.sender, refundAmount);
    }
}
```

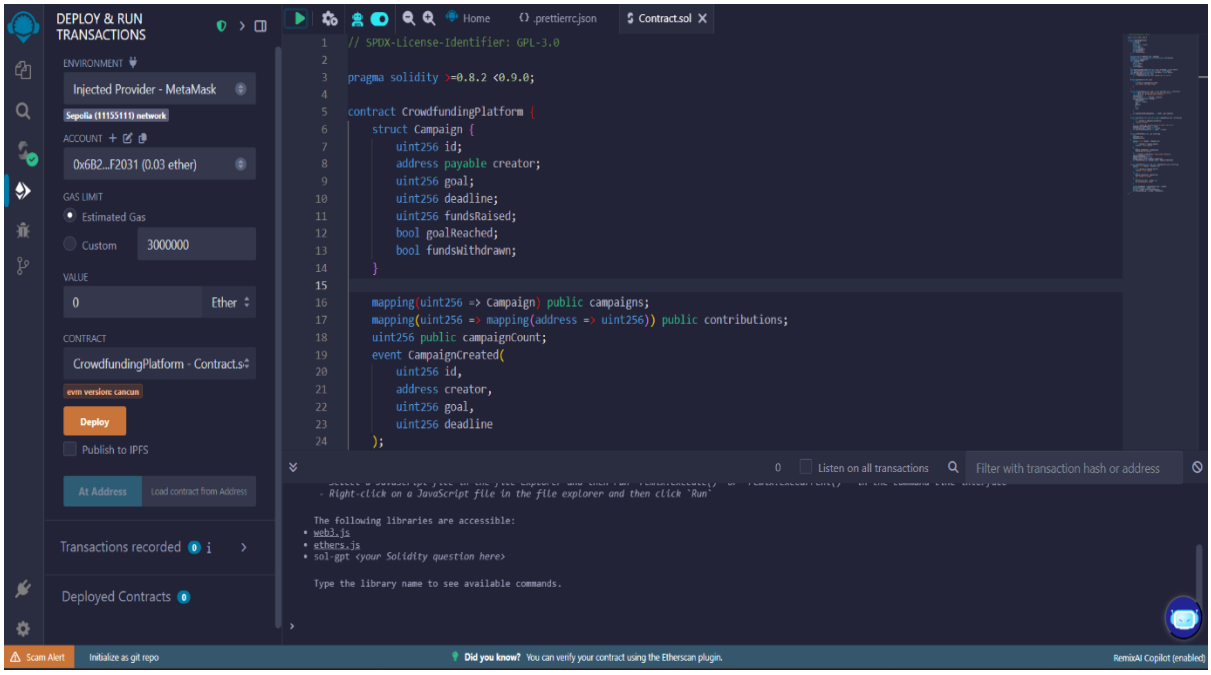


Step 10: Now open the Deploy and Run transactions in the remix ide, in the environment connect your MetaMask there.

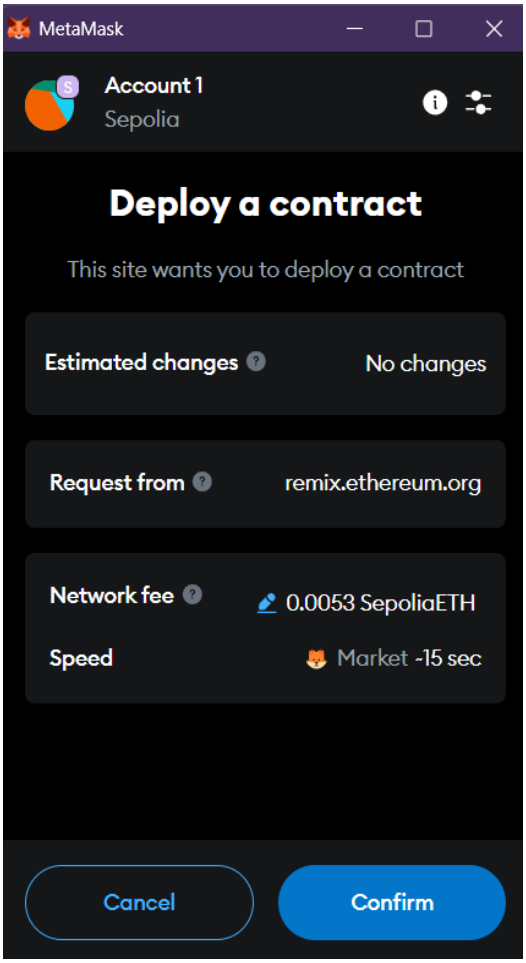


Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

Step 11: Now Compile the smart contract, after the successful compilation the file name should be displayed in the contract placeholder on the deploy and run transaction section.

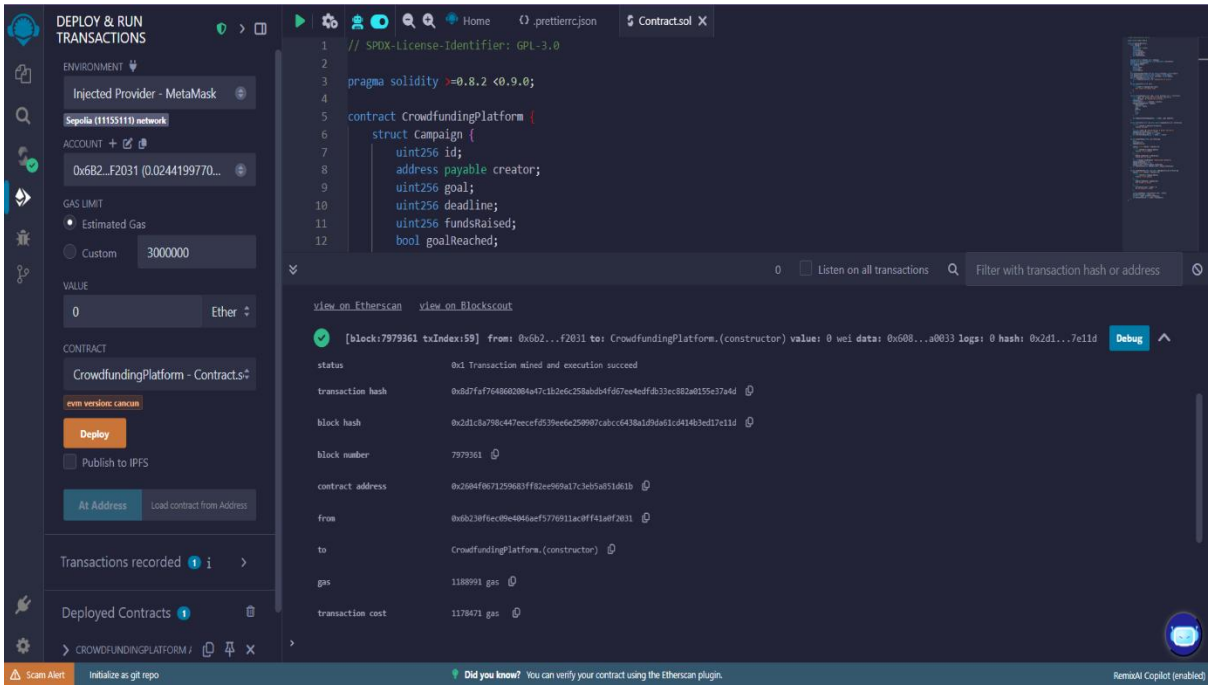


Step 12: For the Transaction, add the value in the value placeholder in the run and deploy transaction section and then click on the deploy to do the transaction.

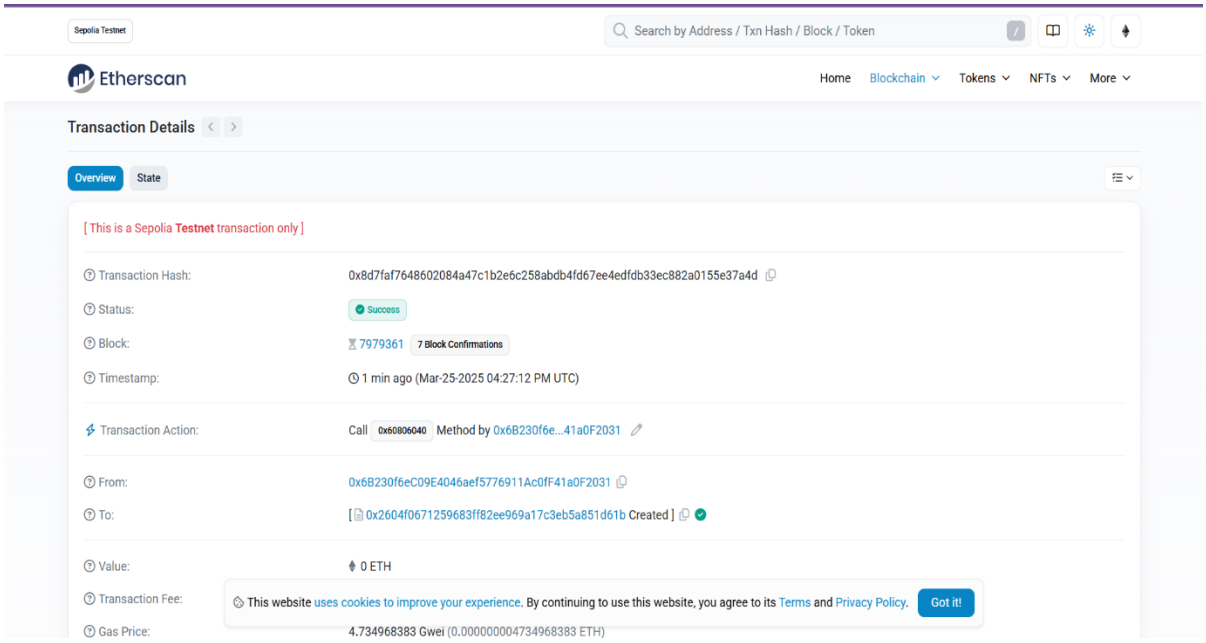


Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A

Step 13: Now when you confirm it the transaction is done and it is shown in the Remix IDE.



Step 14: We can check the details of the transaction in the Etherscan or Blockscan. I have done it in the Etherscan.



Step 15: We can also see in the MetaMask that our Sepolia ETH is also deducted and below the transaction is also shown, when we click on the transaction it will show the transaction details.

Name : Jatin Bendale
Enrollment No : 202203103510038
Div : 6-A