

Project Development Phase and Model Performance Test

Date	03 November 2023
Team ID	NM2023TMID04547
Project Name	Tracking Public Infrastructure and Toll Payments using blockchain

Model Performance Testing:


Project team shall fill the following information when working for blockchain.

S.No.	Parameter	Values	Screenshot
1.	Information gathering	Setup all the Prerequisite:	The Screenshots of information gathering are all given below.
2.	Extract the zip files	Open to vs code	The Screenshots of extract the zip file are all given below.

3.	Remix Idle Platform explorting	Deploy the smart contract code Deploy and run the transaction. Byselecting the environment - inject the MetaMask.	The Screenshots of extract the zip file are all given below.
4	Open file explorer	Open the extracted file and click onthe folder. Open src, and search for utiles. Open cmd enter commands 1.npm install 2.npm bootstrap 3.npm start	The Screenshots of extract the zip file are all given below.
5	LOCALHOST IPADDRESS	Copy the address and open it to chrome so you can see the front end of your project.	The Screenshots of extract the zip file are all given below.

1. INFORMATION GATHERING

Screenshot 1



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare

1 hour to collaborate

2-8 people recommended

➤

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

1

Team gathering

Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

2

Set the goal

Think about the problem you'll be focusing on solving in the brainstorming session.

3

Learn how to use the facilitation tools

Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

PROBLEM

Traditional infrastructure management and toll payment systems often lack transparency, making it difficult for authorities and the public to track the allocation of funds and the condition of infrastructure.

Key rules of brainstorming

To run a smooth and productive session

Stay in topic.

Encourage wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

Screenshot 2

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Person 1

Smart contracts can be used to automate toll collection and facilitate payments in real time.

Introduce a blockchain-based token system for toll payments. Users could purchase toll tokens and use them to pay for tolls.

Having this information on a transparent blockchain can help ensure accountability and efficient maintenance.

Person 2

This could include storing digital identities on the blockchain, allowing for secure and efficient verification during toll payments.

Utilize blockchain for tracking the supply chain of materials and resources used in public infrastructure projects.

This can help in ensuring the quality and origin of materials, reducing fraud, and optimizing resource allocation.

Person 3

Use blockchain to store and protect data related to infrastructure usage. This data can be used to plan future infrastructure projects, address existing infrastructure, and predict maintenance needs.

Create a blockchain-based system to track the allocation of funds for infrastructure projects.

This ensures that tax dollars and other funding sources are used efficiently and transparently.

Person 4

Implement blockchain to facilitate and record public-private partnerships for infrastructure projects.

Develop a blockchain-based feedback and complaint system for users to report issues with public infrastructure.

Utilize blockchain to record and monitor the environmental impact of infrastructure projects.

3

Group ideas

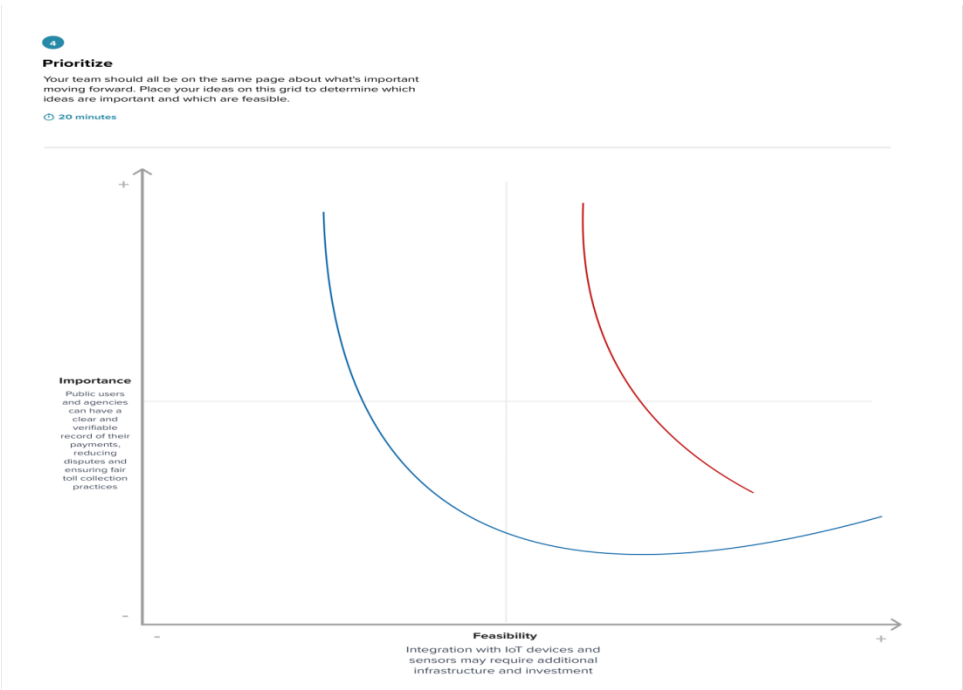
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

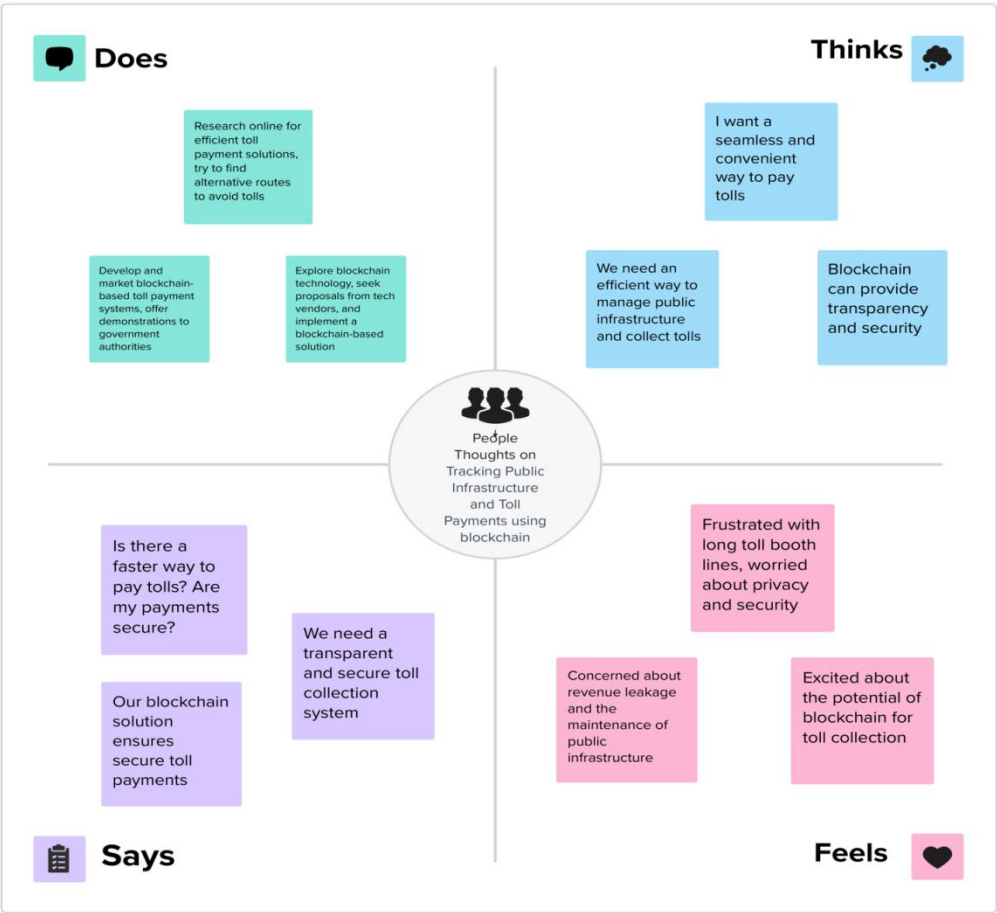
Seek funding from government grants, private investors, or a combination of both. Consider revenue-sharing models with local governments and toll collection agencies to ensure sustainable financing.

Remember that communication and collaboration among team members and stakeholders are crucial throughout the project.

Screenshot 3

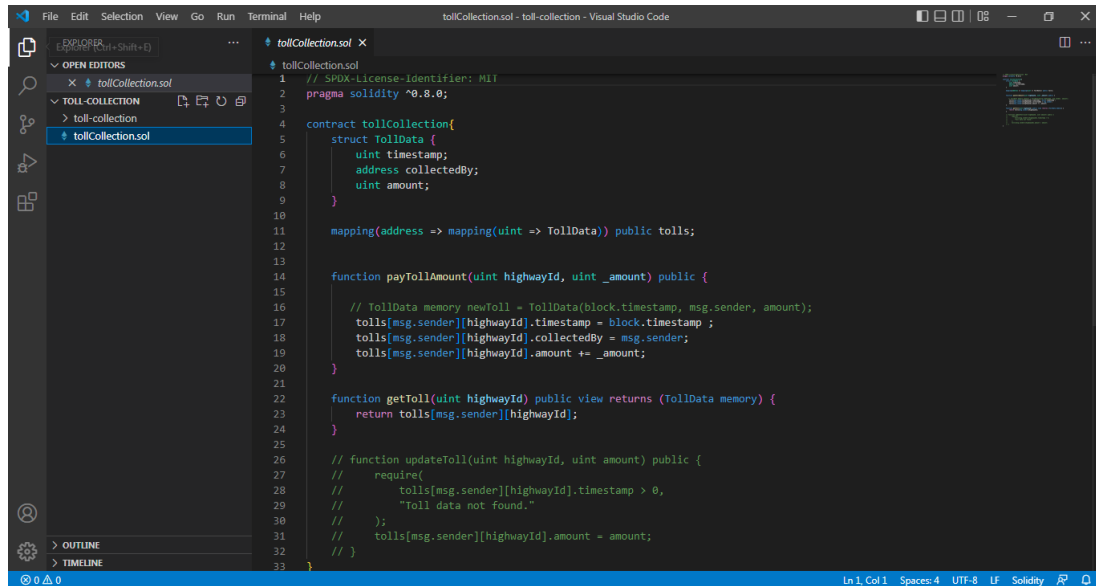


Screenshot 4



2. EXTRACT THE ZIP FILE

Screenshot 1

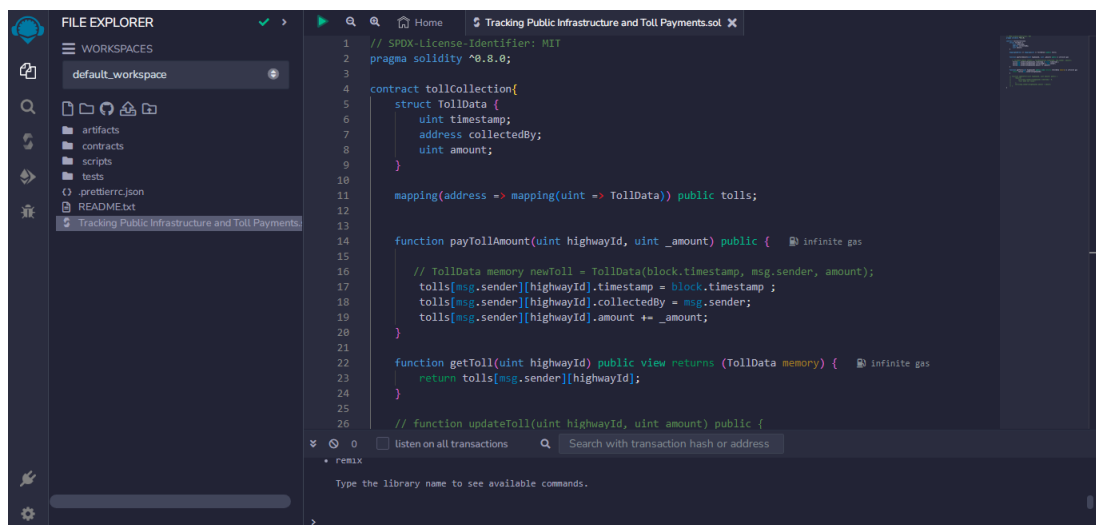


The screenshot shows the Visual Studio Code interface with a Solidity contract named `tollCollection.sol` open. The contract defines a `TollData` struct with fields `uint timestamp`, `address collectedBy`, and `uint amount`. It includes a `mapping` for tolls, a `payTollAmount` function, a `getToll` function, and an `updateToll` function. The status bar at the bottom indicates the file is at line 1, column 1, with 4 spaces, using UTF-8 encoding, and is a Solidity file.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract tollCollection{
5     struct TollData {
6         uint timestamp;
7         address collectedBy;
8         uint amount;
9     }
10
11     mapping(address => mapping(uint => TollData)) public tolls;
12
13
14     function payTollAmount(uint highwayId, uint _amount) public {
15
16         // TollData memory newToll = TollData(block.timestamp, msg.sender, amount);
17         tolls[msg.sender][highwayId].timestamp = block.timestamp ;
18         tolls[msg.sender][highwayId].collectedBy = msg.sender;
19         tolls[msg.sender][highwayId].amount += _amount;
20     }
21
22     function getToll(uint highwayId) public view returns (TollData memory) {
23         return tolls[msg.sender][highwayId];
24     }
25
26     // function updateToll(uint highwayId, uint amount) public {
27     //     require(
28     //         tolls[msg.sender][highwayId].timestamp > 0,
29     //         "Toll data not found."
30     //     );
31     //     tolls[msg.sender][highwayId].amount = amount;
32     // }
```

3. Remix Ide platform Explorting

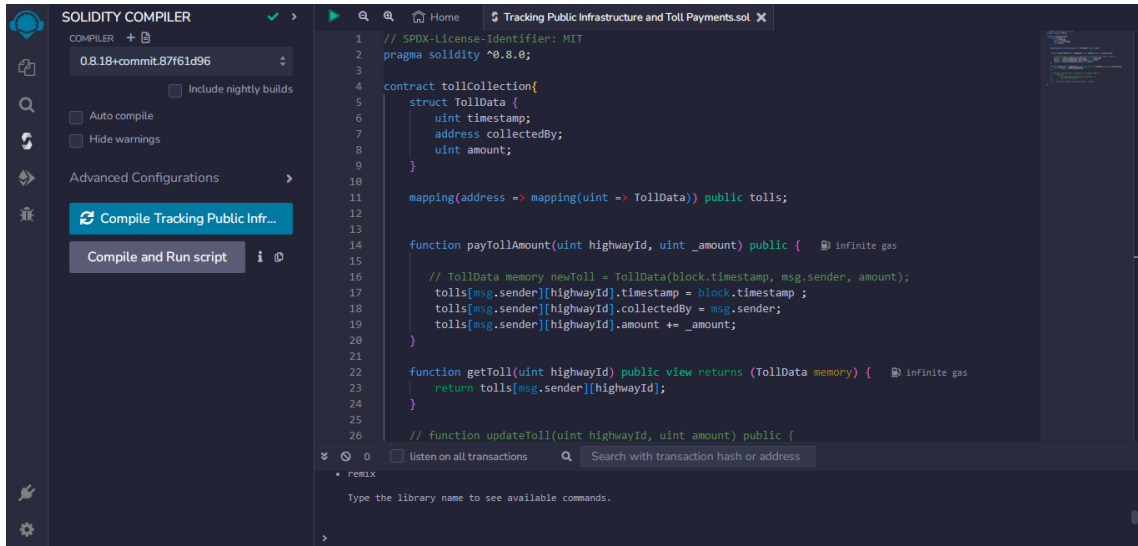
Screenshot 1



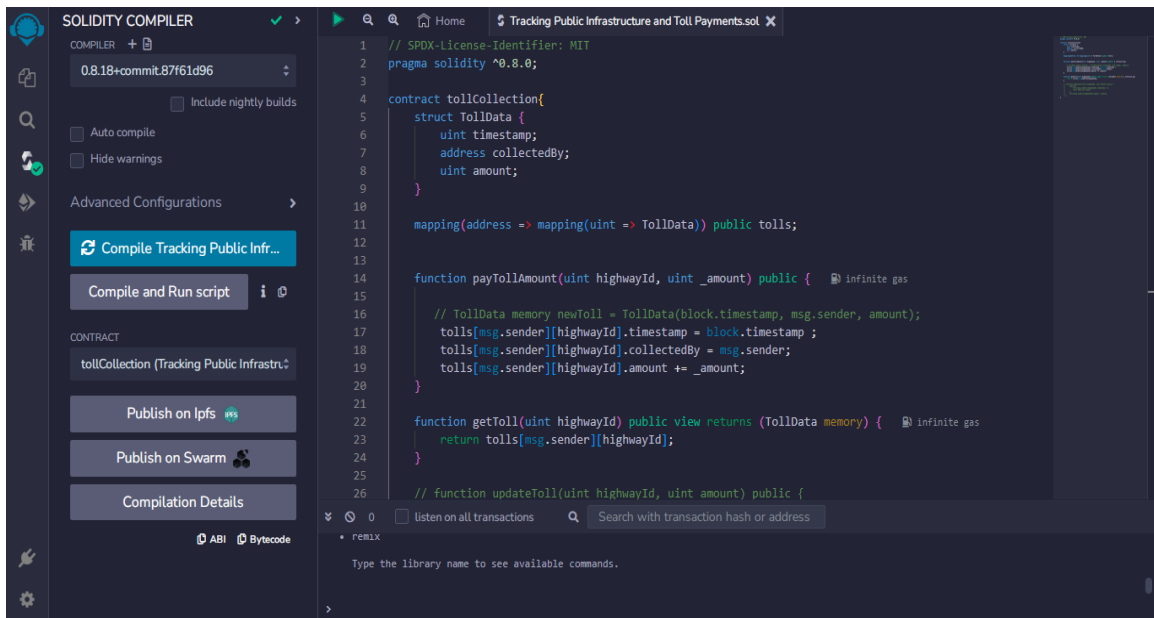
The screenshot shows the Remix IDE interface with the same Solidity contract named `Tracking Public Infrastructure and Toll Payments.sol` open. The contract code is identical to the one in the previous screenshot. The interface includes a `FILE EXPLORER` on the left, a `WORKSPACES` section, and a `REMX` section at the bottom with a search bar for transaction hash or address.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract tollCollection{
5     struct TollData {
6         uint timestamp;
7         address collectedBy;
8         uint amount;
9     }
10
11     mapping(address => mapping(uint => TollData)) public tolls;
12
13
14     function payTollAmount(uint highwayId, uint _amount) public {
15         // TollData memory newToll = TollData(block.timestamp, msg.sender, amount);
16         tolls[msg.sender][highwayId].timestamp = block.timestamp ;
17         tolls[msg.sender][highwayId].collectedBy = msg.sender;
18         tolls[msg.sender][highwayId].amount += _amount;
19     }
20
21     function getToll(uint highwayId) public view returns (TollData memory) {
22         return tolls[msg.sender][highwayId];
23     }
24
25     // function updateToll(uint highwayId, uint amount) public {
26     //     require(
27     //         tolls[msg.sender][highwayId].timestamp > 0,
28     //         "Toll data not found."
29     //     );
30     //     tolls[msg.sender][highwayId].amount = amount;
31     // }
```

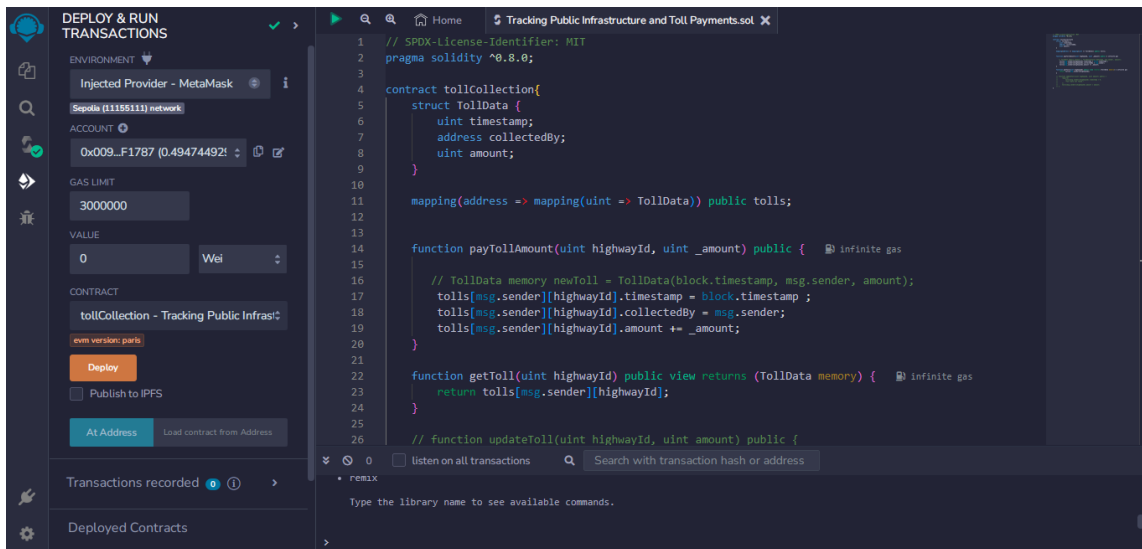
Screenshot 2



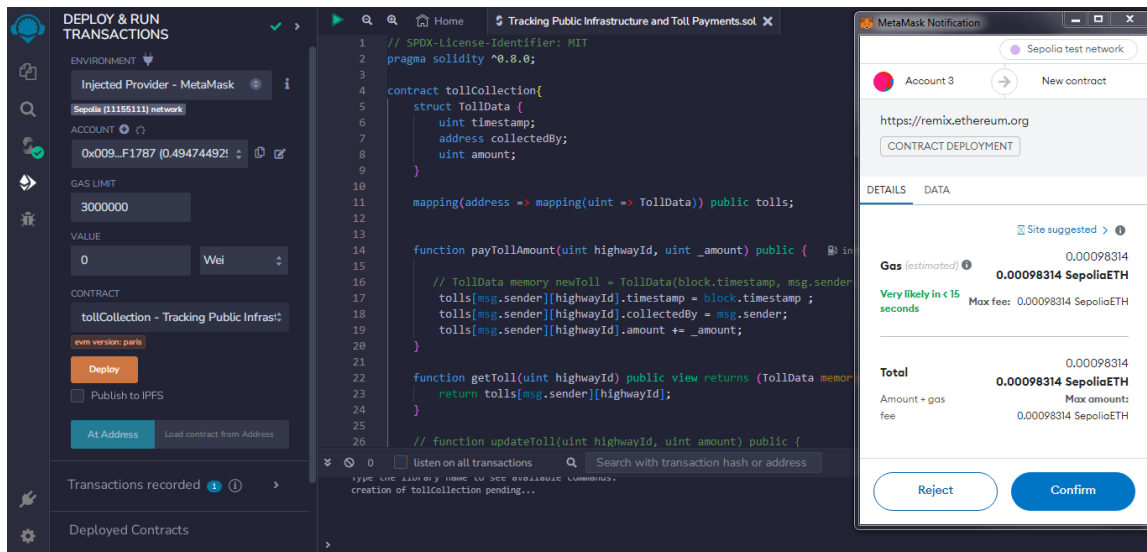
Screenshot 3



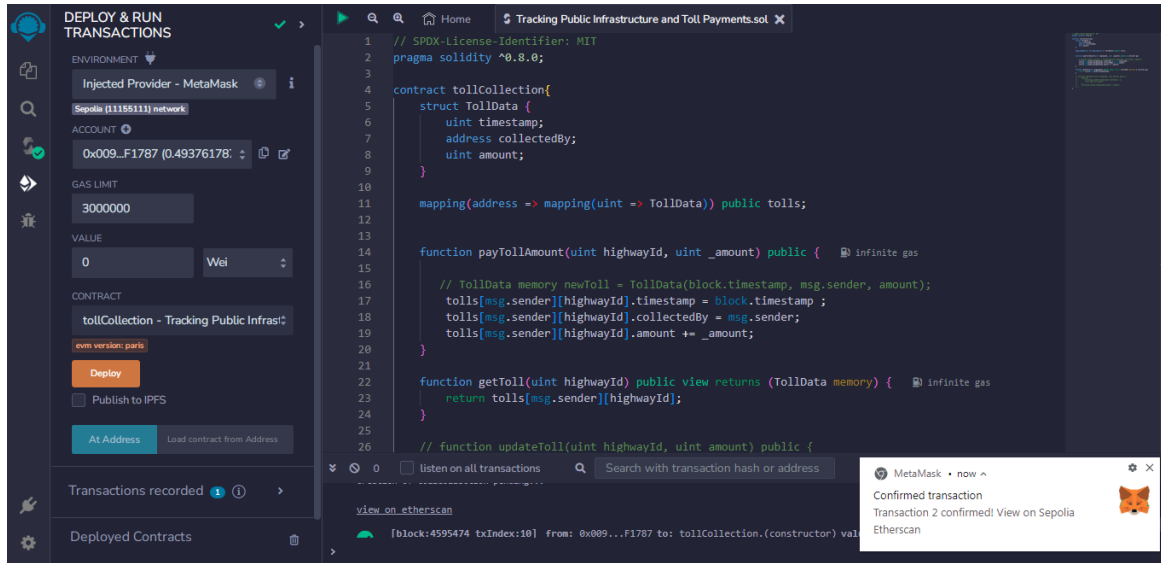
Screenshot 4



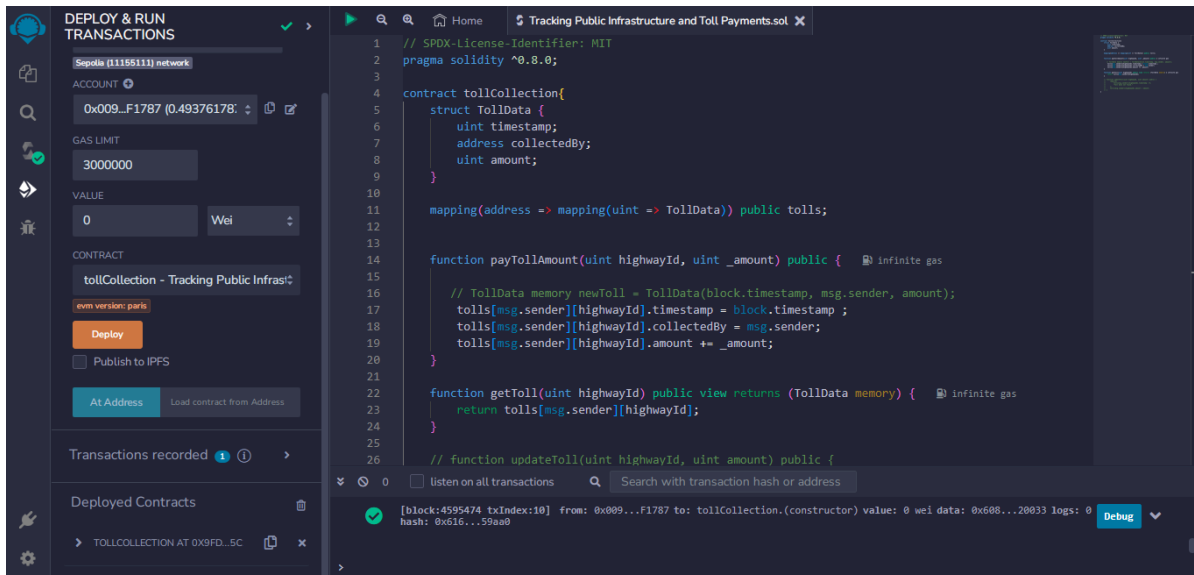
Screenshot 5



Screenshot 6

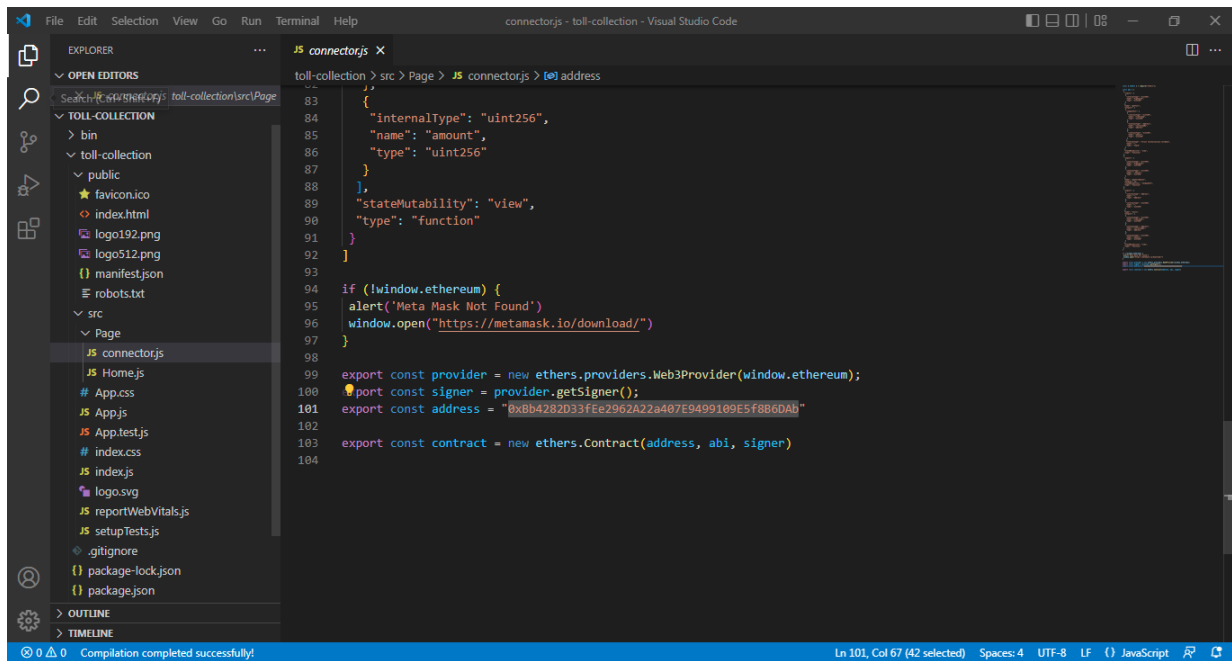


Screenshot 7

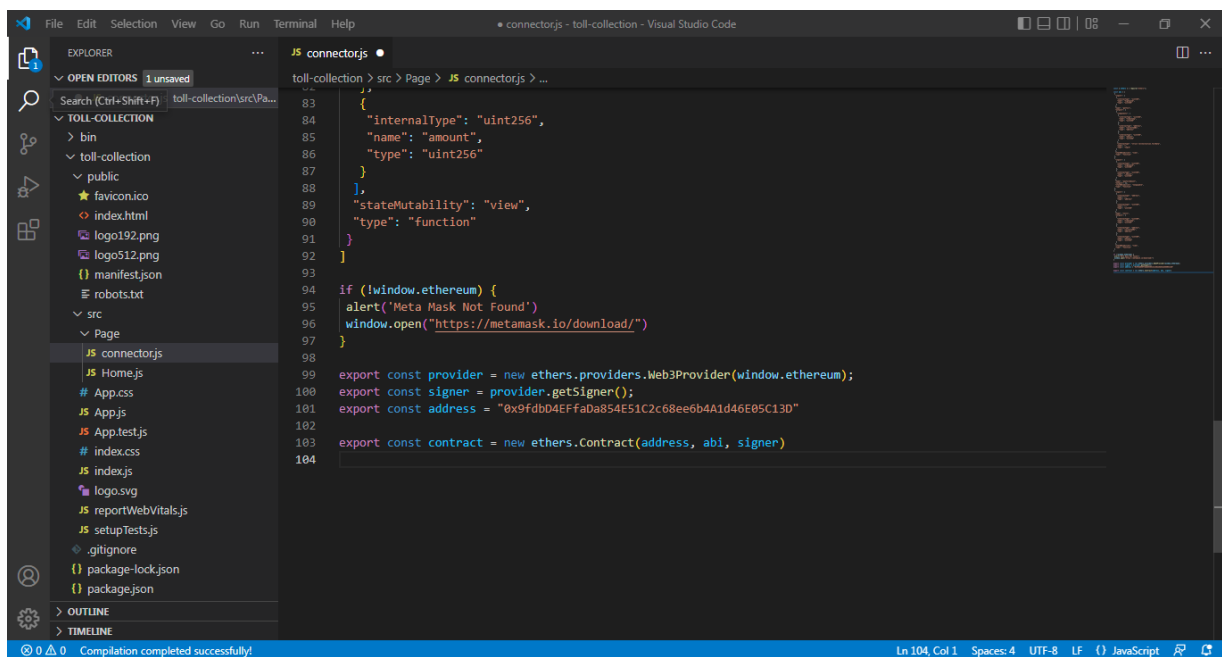


4. OPEN THE FILE EXPLORER

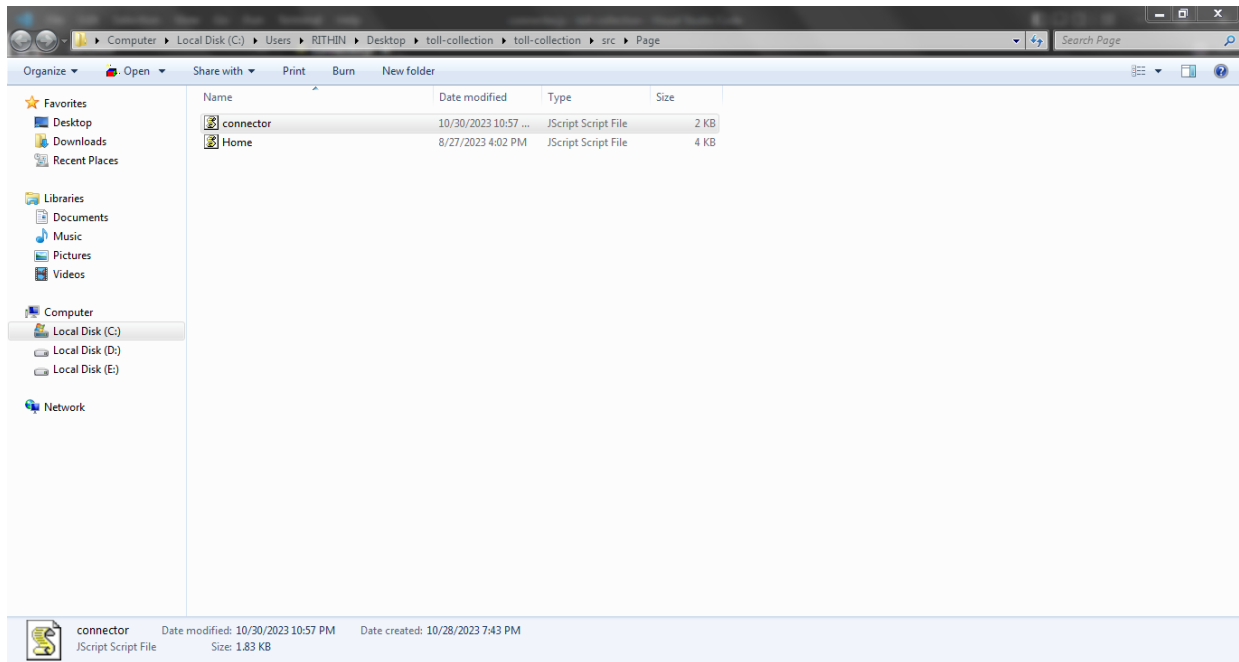
Screenshot 1



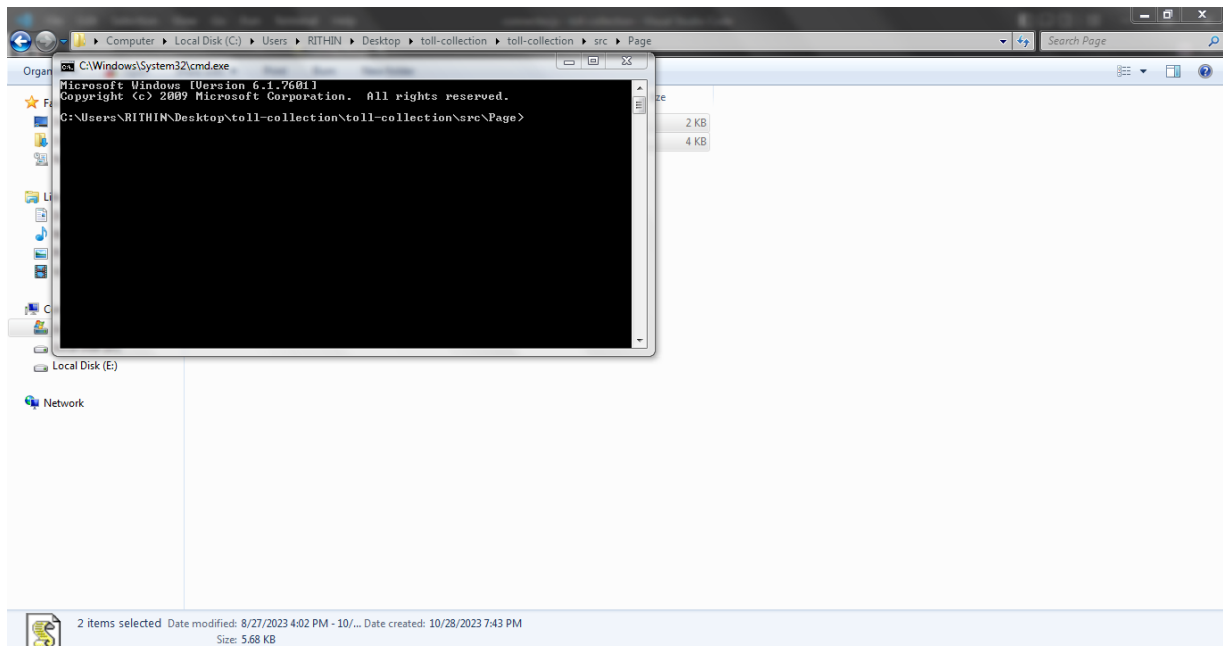
Screenshot 2



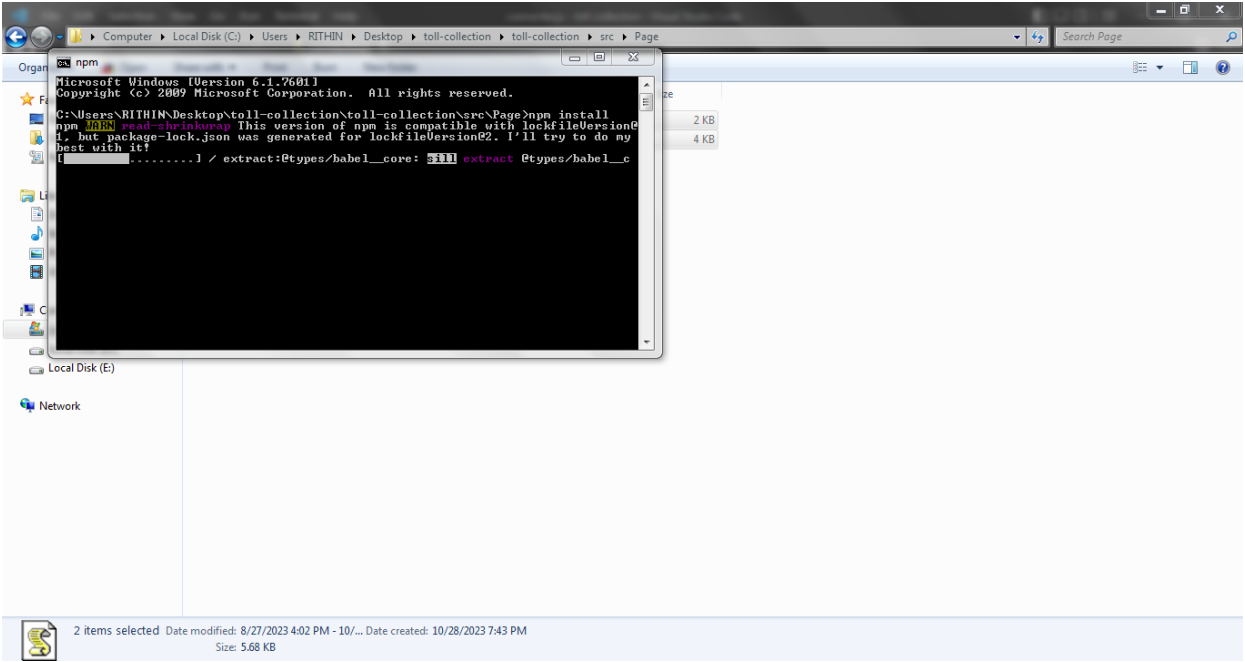
Screenshot 3



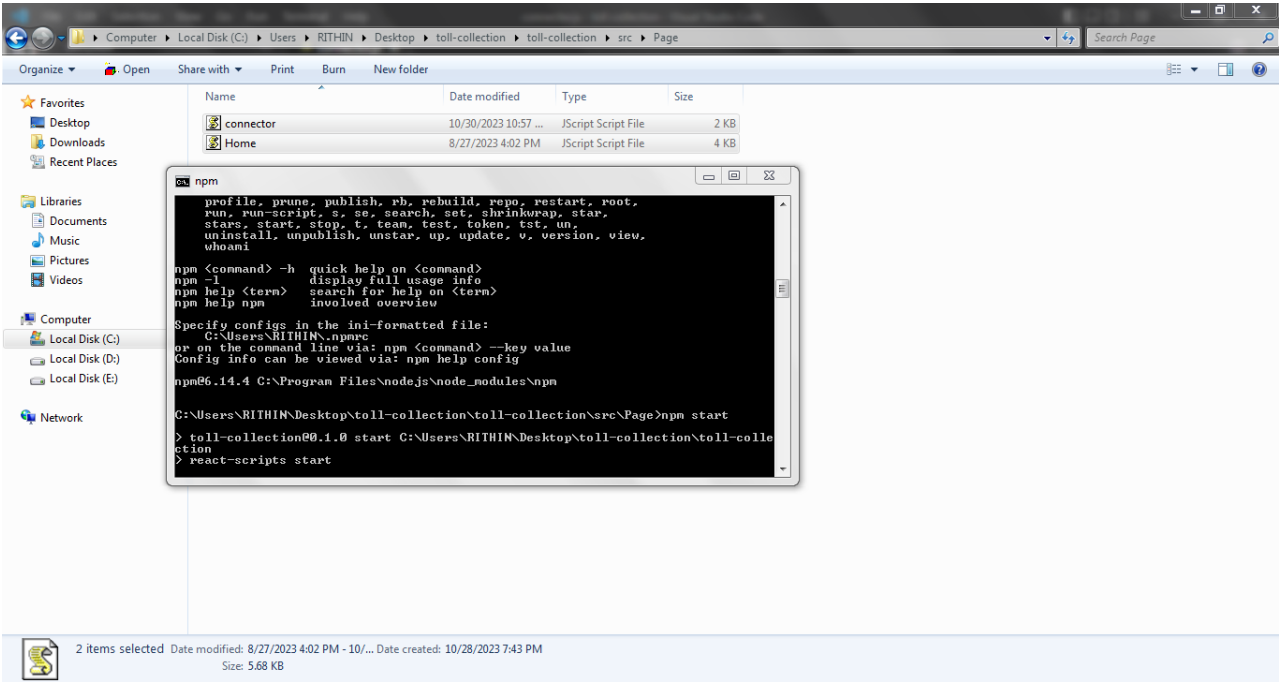
Screenshot 4



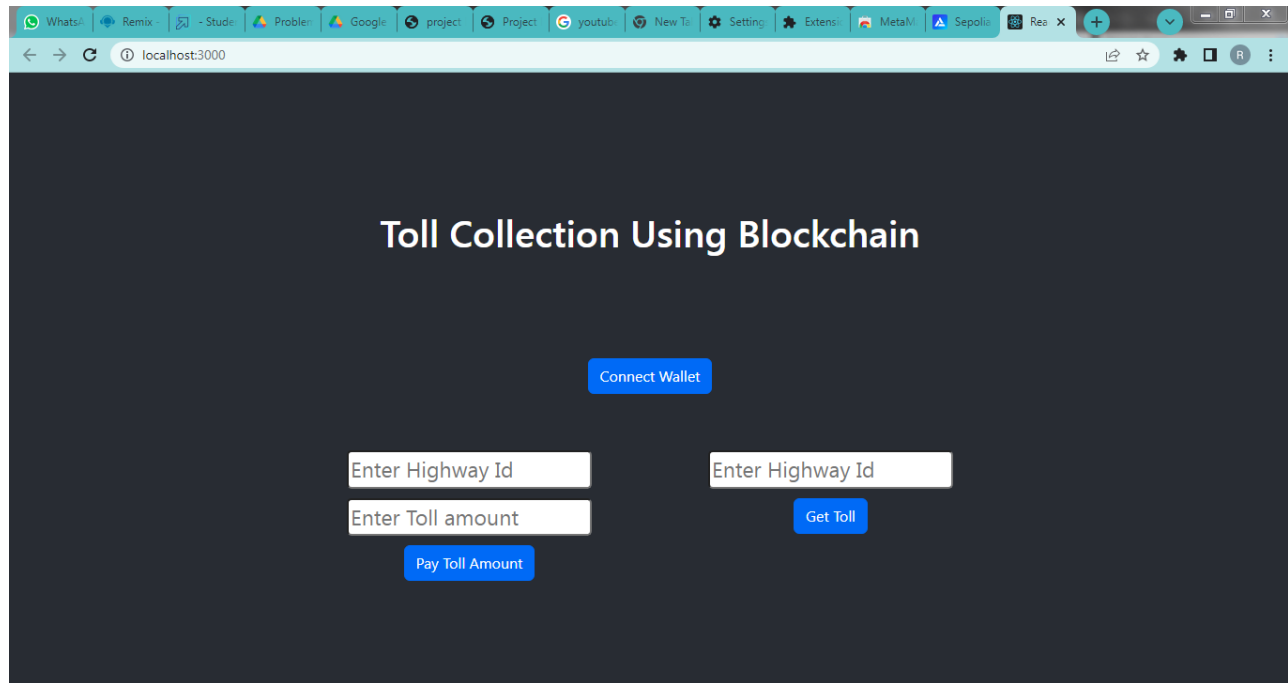
Screenshot 5



Screenshot 6



5. LOCALHOST IP ADDRESS



<http://localhost:3000>