# **PROJECT REPORT**

# **TRACKING PUBLIC INFRASTRUCTURE AND TOLL PAYMENTS USING BLOCKCHAIN**

# **1. INTRODUCTION:**

## **1.1 PROJECT OVERVIEW:**

The project, "Tracking Public Infrastructure and Toll Payments using blockchain," presents an innovative and transformative approach to revolutionizing the management of highway infrastructure and toll collection systems. In the current landscape, manual toll collection processes have been a source of frustration for commuters, leading to extended wait times at toll booths. Additionally, there is a notable lack of transparency in infrastructure maintenance and project tracking. To address these challenges, this project leverages blockchain technology, specifically the Ethereum platform, to create a comprehensive and efficient solution.

At its core, the project introduces the concept of smart contracts, which promises to enhance efficiency, transparency, and data security. This novel approach streamlines toll collection processes, reduces manual errors, and promises to elevate the overall transportation experience for users. User-friendly interfaces and enhanced data security measures ensure a seamless and secure system for all stakeholders.

Moreover, the project delves into innovative strategies to encourage the adoption of technology, including token-based rewards and the establishment of public-private partnerships. This initiative is not merely about an upgrade; it aspires to redefine infrastructure and toll collection in line with modern technological capabilities and evolving public expectations. Ultimately, the goal is to deliver a future where public infrastructure and toll payments are seamlessly integrated into the digital age, ensuring efficiency, security, and a user-centric experience for all.

## **1.2 PURPOSE:**

The purpose of the project, "Tracking Public Infrastructure and Toll Payments using blockchain," is to address the inefficiencies and challenges present in current highway infrastructure management and toll collection systems. The key objectives include:

**1. Enhancing Efficiency:** The project aims to streamline toll collection and infrastructure management processes to reduce wait times, manual errors, and overall inefficiencies, thus improving the experience for commuters and stakeholders.

**2. Ensuring Transparency:** By utilizing blockchain technology, the project seeks to provide real-time, transparent access to data related to infrastructure projects, maintenance, and toll collection, thereby enhancing trust and accountability.

**3. Data Security:** The project places a strong emphasis on data security, implementing robust encryption and backup measures to safeguard user information and transaction data, and addressing concerns about data loss.

**4. Promoting Technology Adoption:** Through user-friendly interfaces, the project aims to encourage the adoption of technology, making the transition to the new system as smooth as possible for all stakeholders.

**5. Incentivizing Innovation:** By introducing token-based rewards and exploring public-private partnerships, the project aims to stimulate innovation and accelerate the adoption of blockchain technology in the management of public infrastructure.

In summary, the purpose of this project is to transform the existing infrastructure and toll payment systems by leveraging blockchain technology to make them more efficient, transparent, and secure, while also encouraging technology adoption and fostering innovation in the transportation sector.

# **2. LITERATURE SURVEY:**

## **2.1 EXISTING SYSTEM:**

In the realm of intelligent transportation systems, the adoption of blockchain technology has paved the way for innovative solutions in toll collection and traffic optimization. The existing toll collection system, built on the foundations of blockchain technology, is a testament to the potential of this decentralized and secure approach in revolutionizing the way we collect tolls and manage traffic on our highways. The existing toll collection system serves as an example of successful blockchain integration in the intelligent transportation sector. Its emphasis on secure and efficient toll collection, data transparency, and traffic optimization aligns with the goals of our project, "Tracking Public Infrastructure and Toll Payments using

blockchain." By considering the adoption of this existing system, we aim to leverage its strengths in toll collection as a foundational element in our broader infrastructure and transportation management initiative.

## **2.2 REFERENCES:**

[1] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,”. https://bitcoin.org/bitcoin.pdf. Accessed by Jan. 29 2019

[2] Buterin V., “A next-generation smart contract and decentralized application platform. white paper”. 2014 Jan

[3] M. Singh, S. Kim, “Blockchain based intelligent vehicle data sharing frame- work,” arXiv preprint, arXiv: 1708.09721, 01 Sept. 2017.

[4] M. Singh, S. Kim, “Intelligent vehicle-trust point: reward-based intelligent vehicle communication using blockchain,” arXiv preprint, arXiv: 1707.07442, 28 Jul 2017.

[5] C. Cai, X. Yuan, and C. Wang, “Towards trustworthy and private keyword search in encrypted decentralized storage,” in Proc. IEEE 2017 International Conference on Communications (ICC), Paris, May 2017, pp. 1-7.

[6] Sharma, P.K., Moon, S.Y. and Park, J.H., “Block-VN: A Distributed Blockchain Based Vehicular Network Architecture in Smart City”. JIPS, 13(1), pp.184-195, 2017

[7] Jiang, T., Fang, H. and Wang, H., “Blockchain-based Internet of vehicles: distributed network architecture and performance analysis”. IEEE Internet of Things Journal, 2018

[8] Xu, C., Wang, K., Li, P., Guo, S., Luo, J., Ye, B. and Guo, M., “Making big data open in edges: A resource-efficient blockchain-based approach. IEEE Transactions on Parallel and Distributed Systems”, 2018.

[9] Yang, Z., Yang, K., Lei, L., Zheng, K. and Leung, V.C., “Blockchain-based decentralized trust management in vehicular networks”. IEEE Internet of Things Journal, 2018.

[10] Panescu, A.T. and Manta, V., “Smart Contracts for Research Data ˜ Rights Management over the Ethereum Blockchain Network”. Science & Technology Libraries, 37(3), pp.235-245, 2018.

## **2.3 PROBLEM STATEMENT DEFINITION:**

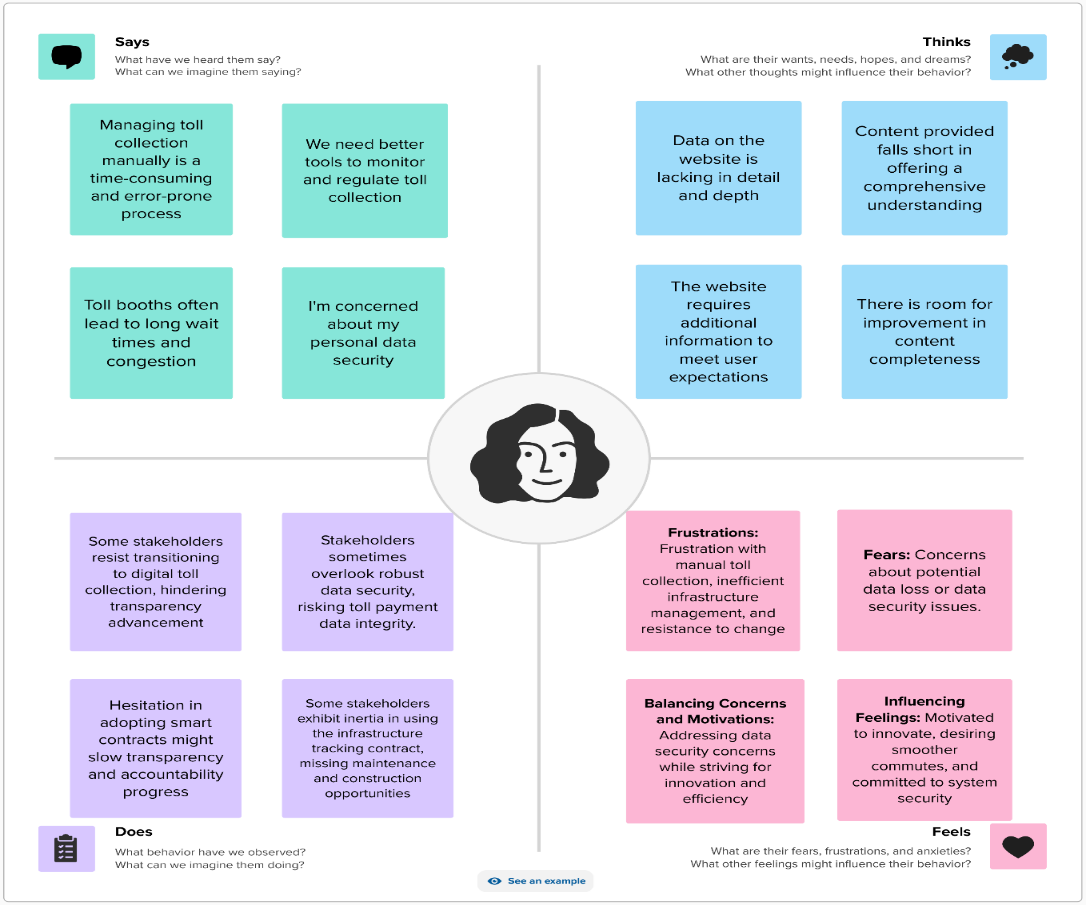
The current state of infrastructure and toll payment systems presents a myriad of challenges that hinder the efficiency and transparency of these crucial components of transportation management. Manual toll collection processes lead to extended wait times at toll booths, causing frustration among commuters and travelers. Furthermore, the lack of transparency in infrastructure maintenance and project tracking compounds the inefficiencies, eroding trust and accountability within the system. Stakeholders, including government authorities, toll operators, and the public, express valid concerns about data security and the potential for data loss, raising significant reservations about the reliability of the current system.

This complex problem necessitates an innovative and transformative solution that can streamline toll collection, enhance transparency, and ensure robust data security. At its core, the problem revolves around improving the efficiency and security of toll collection and infrastructure management while addressing stakeholder concerns about data security and resistance to technological change. The solution should aim to eliminate manual errors, reduce wait times, establish a secure and user-friendly toll payment system, and ultimately provide a seamless and efficient transportation experience for users and stakeholders.

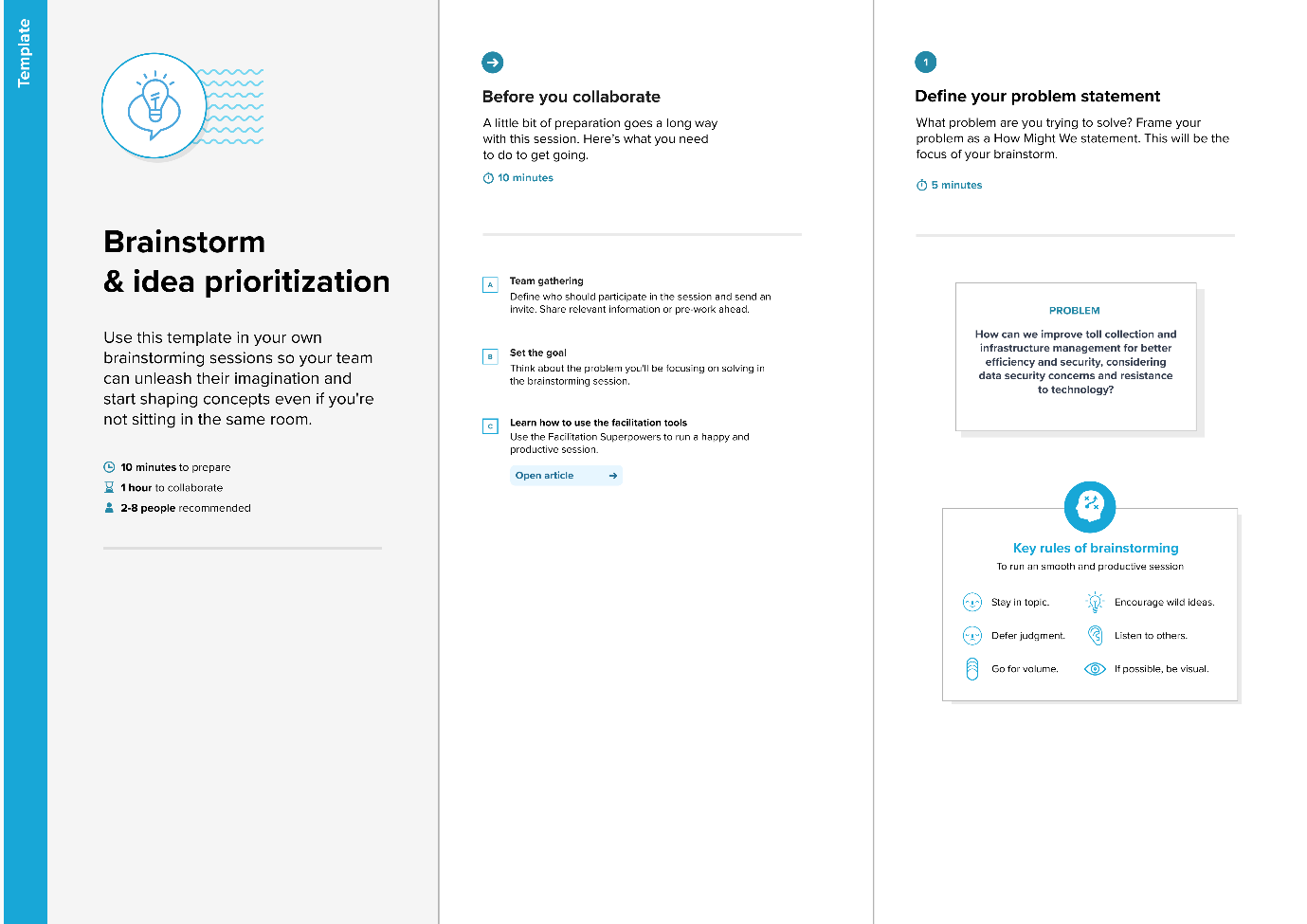
By addressing these challenges, the proposed project, "Tracking Public Infrastructure and Toll Payments using blockchain," aims to offer a comprehensive and forward-thinking response to the current issues in infrastructure and toll payment systems.

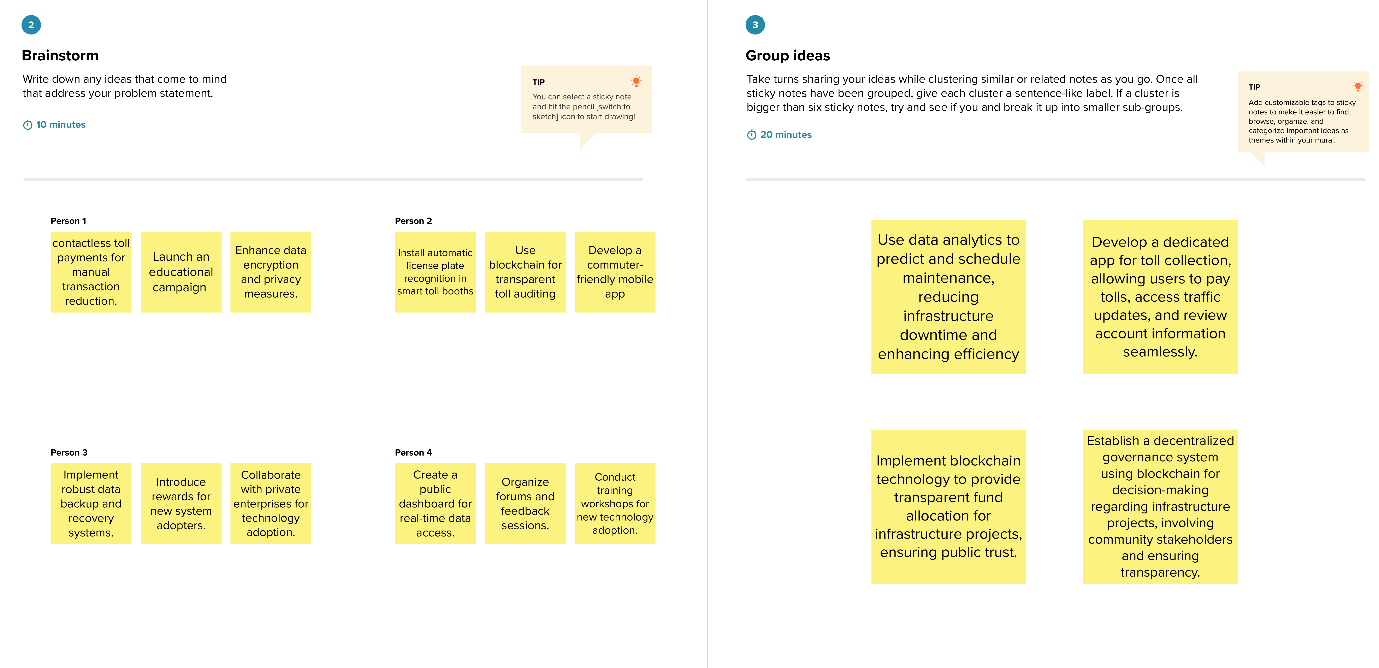
# **3. IDEATION & PROPOSED SOLUTION:**

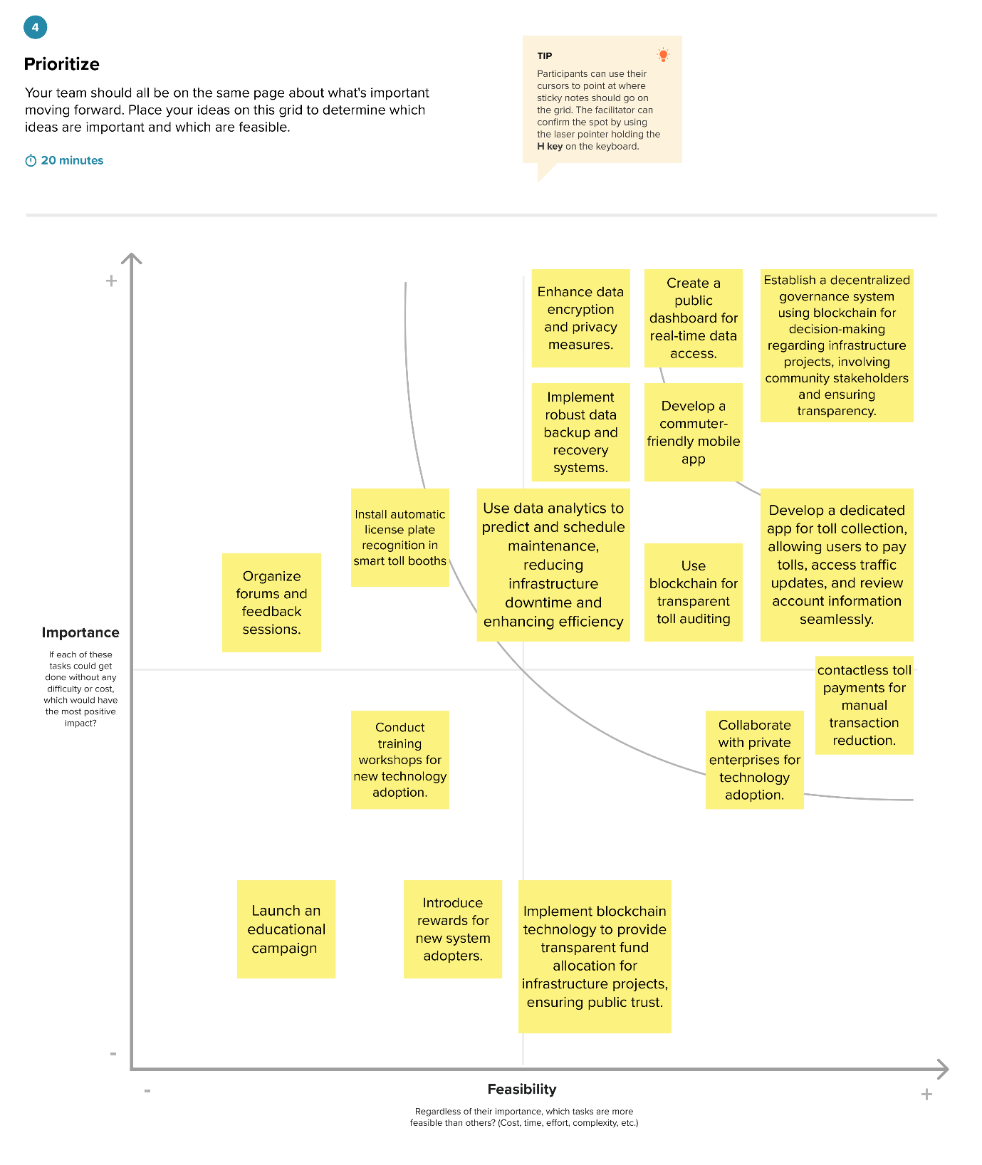
**3.1 Empathy Map Canvas:**

****

## **3.2 IDEATION & BRAINSTORMING:**







**4 REQUIREMENT ANALYSIS:**

## **4.1 FUNCTIONAL REQUIREMENT:**

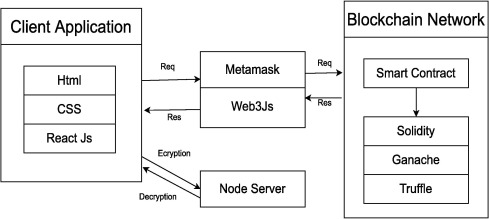
* The project's functional requirements encompass seamless toll collection, transparent infrastructure tracking, and enhanced user experience. It includes the implementation of smart contracts for automated toll payments and accurate record-keeping.
* The system must provide real-time data access through user-friendly interfaces, enabling commuters to check toll information and receive traffic updates.
* Additionally, robust data encryption and secure backup mechanisms are imperative to ensure data integrity and privacy.
* The project requires a transparent audit trail facilitated by blockchain, enabling stakeholders to monitor toll collection processes effectively. Furthermore, the introduction of token-based rewards and collaboration with private entities are essential components to incentivize user adoption and accelerate technology integration.
* These functional requirements collectively ensure a streamlined, secure, and user-focused infrastructure and toll payment system.

## **4.2 NON-FUNCTIONAL REQUIREMENT:**

* The project's non-functional requirements encompass reliability, security, and usability. It necessitates a high level of system reliability, ensuring uninterrupted toll collection operations even during peak traffic periods.
* Data security is paramount, requiring adherence to industry standards for encryption and privacy to safeguard user information. Scalability is crucial, allowing the system to handle varying transaction volumes efficiently.
* Additionally, the user interface must be intuitive and responsive, providing a seamless experience for both commuters and stakeholders.
* Compliance with regulatory standards and seamless integration with existing infrastructure are vital aspects. The system's performance must be optimized to handle concurrent users, ensuring swift response times.
* Accessibility, encompassing compatibility with various devices and assistive technologies, is also a non-functional requirement, promoting inclusivity and user satisfaction.

# **5. PROJECT DESIGN:**

* 1. **DATA FLOW DIAGRAM:**



## **USER STORIES:**

1. As a driver, I want a seamless and contactless toll payment experience to reduce the time spent at toll booths and make my commute more efficient.

2. As a highway operator, I want an intuitive dashboard that allows me to easily monitor and manage toll collection data, ensuring a smooth operation.

3. As a government authority, I want a transparent system that provides real-time updates on public infrastructure projects, enabling me to make informed decisions about infrastructure management and maintenance.

4. As a blockchain developer, I want a secure and scalable system architecture that can efficiently process a high volume of transactions, ensuring the reliability and performance of the blockchain network.

5. As a commuter, I want a system that incentivizes using less congested routes by offering rewards or discounts, reducing my travel time, and contributing to better traffic flow.

6. As a public user, I want access to real-time information about toll rates, payment history, and infrastructure maintenance, allowing me to plan my travel and finances effectively.

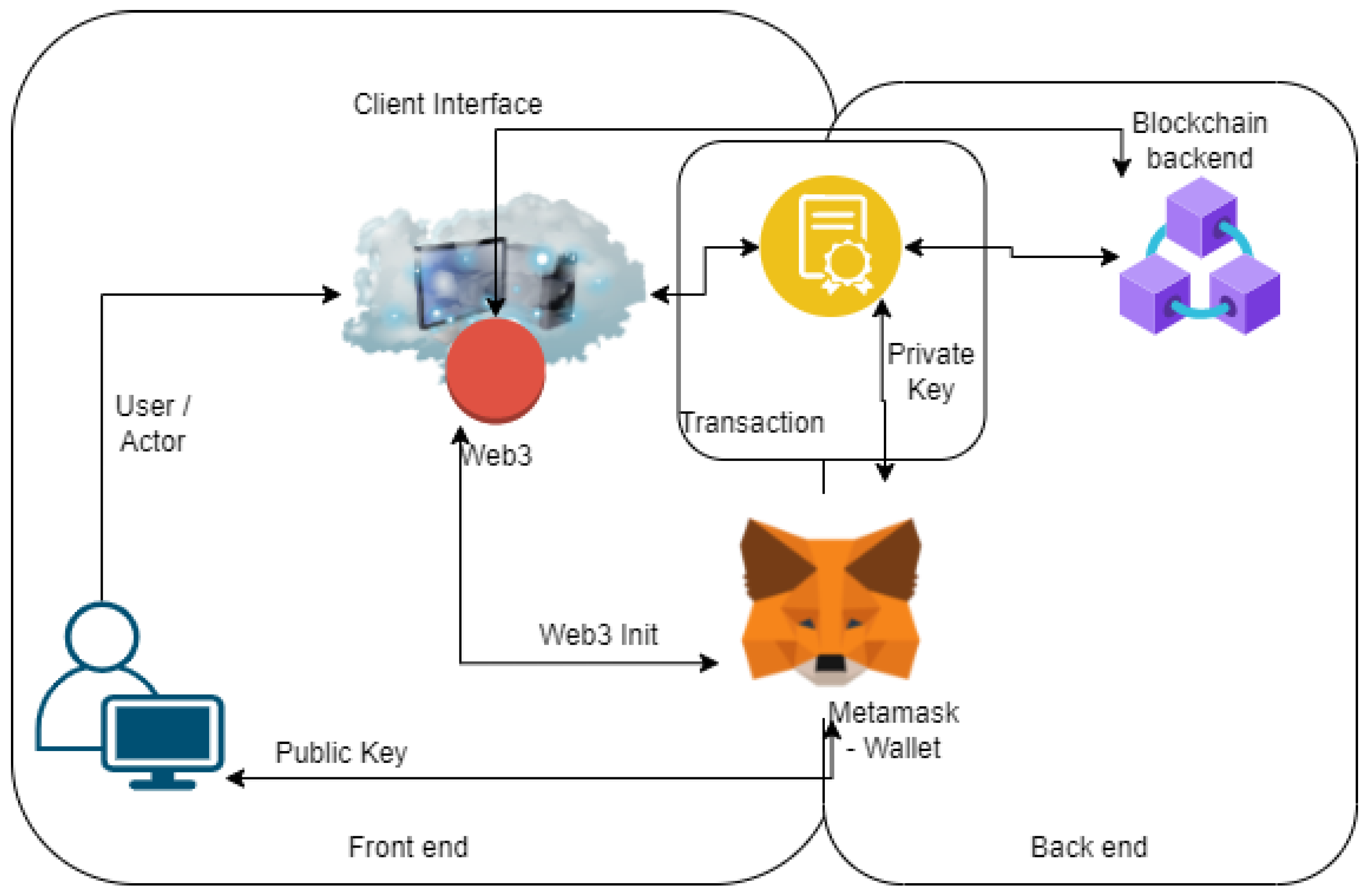
7. As a blockchain expert, I want to collaborate with the project team to ensure that the blockchain technology used is robust, secure, and complies with best practices in the industry.

8. As a toll booth operator, I want an efficient and user-friendly interface to process toll payments, reducing the chances of errors and ensuring a smooth experience for drivers.

9. As a maintenance crew member, I want a platform that provides me with the latest updates on infrastructure projects, helping me coordinate maintenance tasks effectively and ensure the safety of commuters.

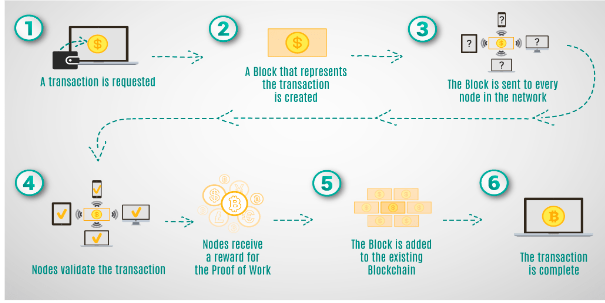
10. As a public transportation user, I want a system that encourages the use of eco-friendly and electric vehicles, potentially offering discounts or incentives to promote environmentally friendly commuting.

## **5.2 SOLUTION ARCHITECTURE:**



# **PROJECT PLANNING:**

* 1. **TECHNICAL ARCHITECTURE:**

****

## **6.2 SPRINT PLANNING & ESTIMATION:**

* The sprint planning for this project involves breaking down tasks into manageable segments, focusing on key milestones. During sprint sessions, the team will prioritize implementing smart contracts, developing user interfaces, and integrating data security measures. Estimation is based on the complexity of tasks, with more straightforward components receiving shorter sprint durations for rapid progress.
* Implementing contactless toll payments might constitute one sprint, while developing the user-friendly app could span multiple sprints. Regular review and adaptation of the sprint plan allow for dynamic adjustments based on the team's velocity, ensuring efficient progress toward project goals while maintaining flexibility to address emerging challenges and opportunities.

## **6.3 SPRINT DELIVERY SCHEDULE**

The sprint delivery schedule for this project is structured to ensure consistent progress and timely milestones. Each sprint, typically lasting two to four weeks, focuses on specific tasks and deliverables. The initial sprints concentrate on

foundational elements such as setting up the blockchain environment and developing basic smart contracts. Subsequent sprints involve iterative enhancements, including user interface refinements, data security implementation, and integration of token-based rewards. Regular sprint reviews and retrospectives allow for continuous feedback and adjustment. The project adopts an Agile approach, allowing for flexibility in adapting to changing requirements. By adhering to this sprint delivery schedule, the project ensures a systematic, incremental rollout of features, leading to a successful and efficient project completion.

# **7 CODING AND SOLUTIONING:**

**7.1 Feature 1: Toll Collection**

**Explanation**: This feature enables drivers to pay tolls and record the transaction details, including the timestamp, collector's address, and amount. Here are some code enhancements:

**1. Struct for Toll Data:** The code defines a struct named `TollData` to represent toll transaction information. This struct includes three fields: `timestamp`, `collectedBy`, and `amount`. It will store the timestamp of the transaction, the address of the collector (toll booth operator), and the amount paid by the driver.

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract TollCollection {

struct TollData {

uint timestamp;

address collectedBy;

uint amount;

}

**2. Mapping for Toll Data:** The code uses a mapping to store toll data. The mapping is a data structure that associates an address (driver's address) with another mapping that maps highway identifiers (represented by `uint`) to `TollData`. This allows the

contract to store toll data for each driver on each highway. The `tolls` mapping is of the form `mapping(address => mapping(uint => TollData))`.

// Mapping to store toll data for each driver (address) and highway (highwayId)

mapping(address => mapping(uint => TollData)) public tolls;

**3. Event for Logging Toll Payments**: An event named `TollPaid` is defined. Events are used to log important occurrences in smart contracts. In this case, it's used to log toll payments. The event logs details such as the driver's address, the highway identifier, the timestamp of the transaction, and the amount paid.

// Event to log toll payments

event TollPaid(address indexed driver, uint indexed highwayId, uint timestamp, uint amount);

**4. Function to Pay Toll Amount:** The `payTollAmount` function allows drivers to pay tolls. It takes two parameters: `highwayId` (the identifier of the highway) and `\_amount` (the toll amount to be paid). The function starts with a `require` statement to ensure that a valid amount is paid (greater than zero).

function payTollAmount(uint highwayId, uint \_amount) public {

// Ensure a valid amount is paid

require(\_amount > 0, "Amount must be greater than zero");

**5. Storage of Toll Data:** Within the function, the toll data for the driver on the specified highway is stored. The code first retrieves the `TollData` struct from the mapping, and then it updates the `timestamp`, `collectedBy`, and `amount` fields. This records the transaction details.

// Store toll data for the driver on the specified highway

TollData storage toll = tolls[msg.sender][highwayId];

toll.timestamp = block.timestamp;

toll.collectedBy = msg.sender;

toll.amount += \_amount;

**6. Event Emission:** After recording the toll payment, an event is emitted using `emit`. This logs the toll payment, making it accessible for later reference and external applications.

// Emit an event to log the toll payment

emit TollPaid(msg.sender, highwayId, block.timestamp, \_amount);

}

function getToll(uint highwayId) public view returns (TollData memory) {

return tolls[msg.sender][highwayId];

}

}

# **8. PERFORMANCE TESTING:**

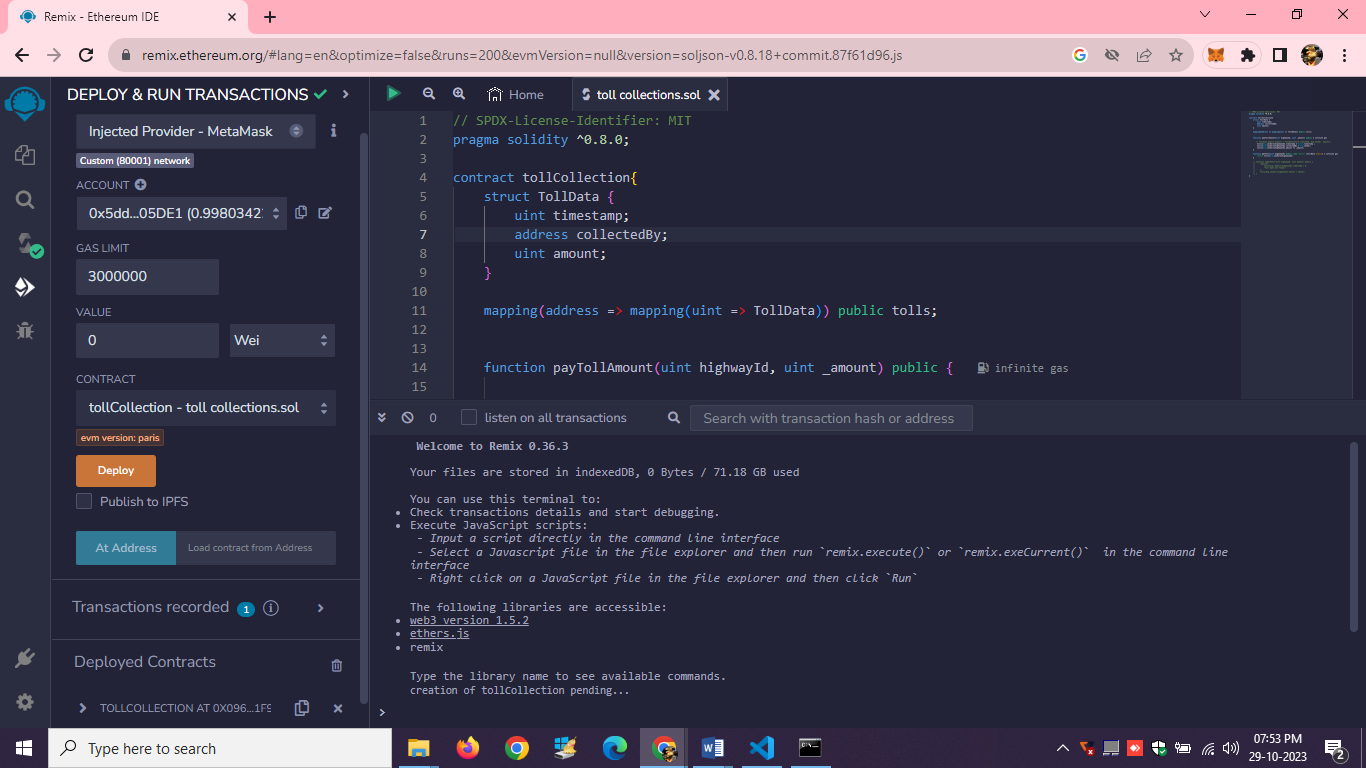
## **8.1 PERFORMANCE METRICS:**

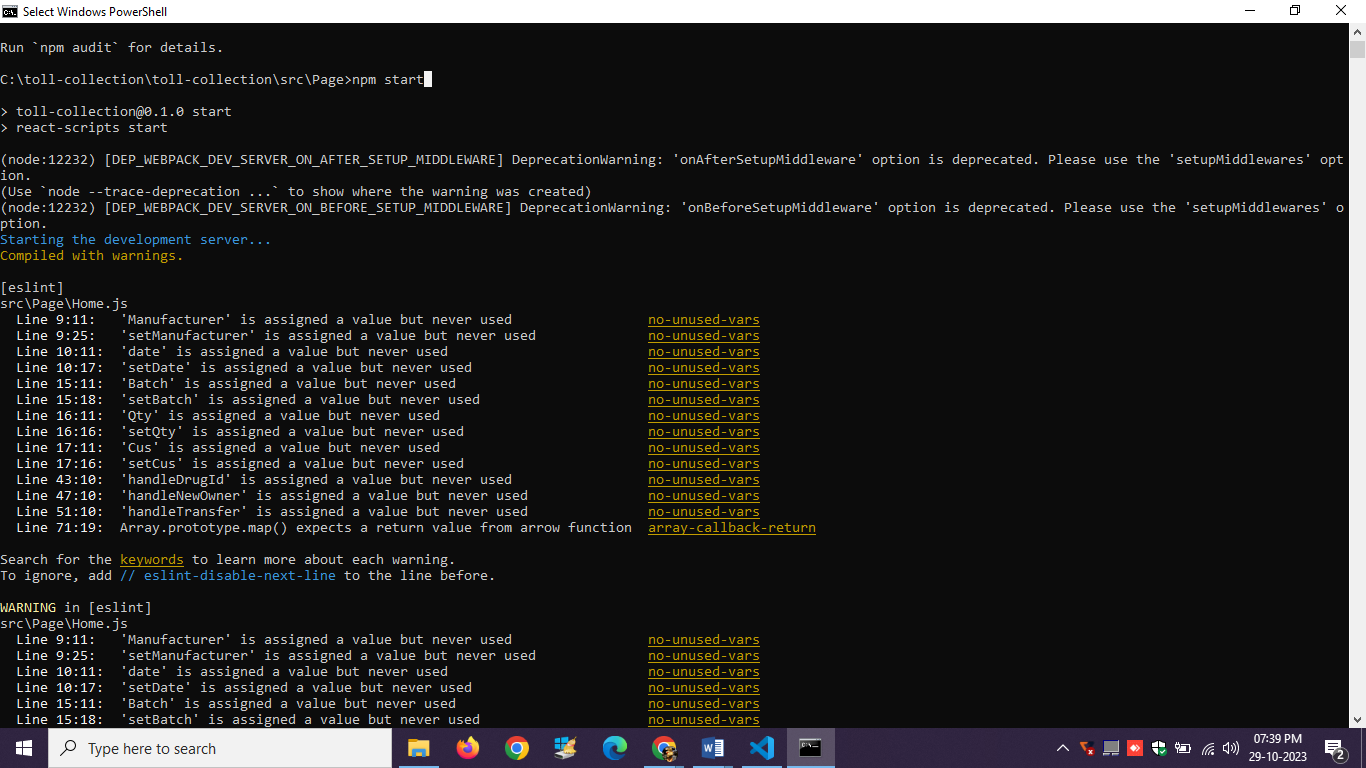
|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Parameter** | **Values** | **Screenshot** |
| 1. | Information gathering | Setup all the Prerequisite: |  |
| 2. | Extract the zip files | Open to vs code |  |

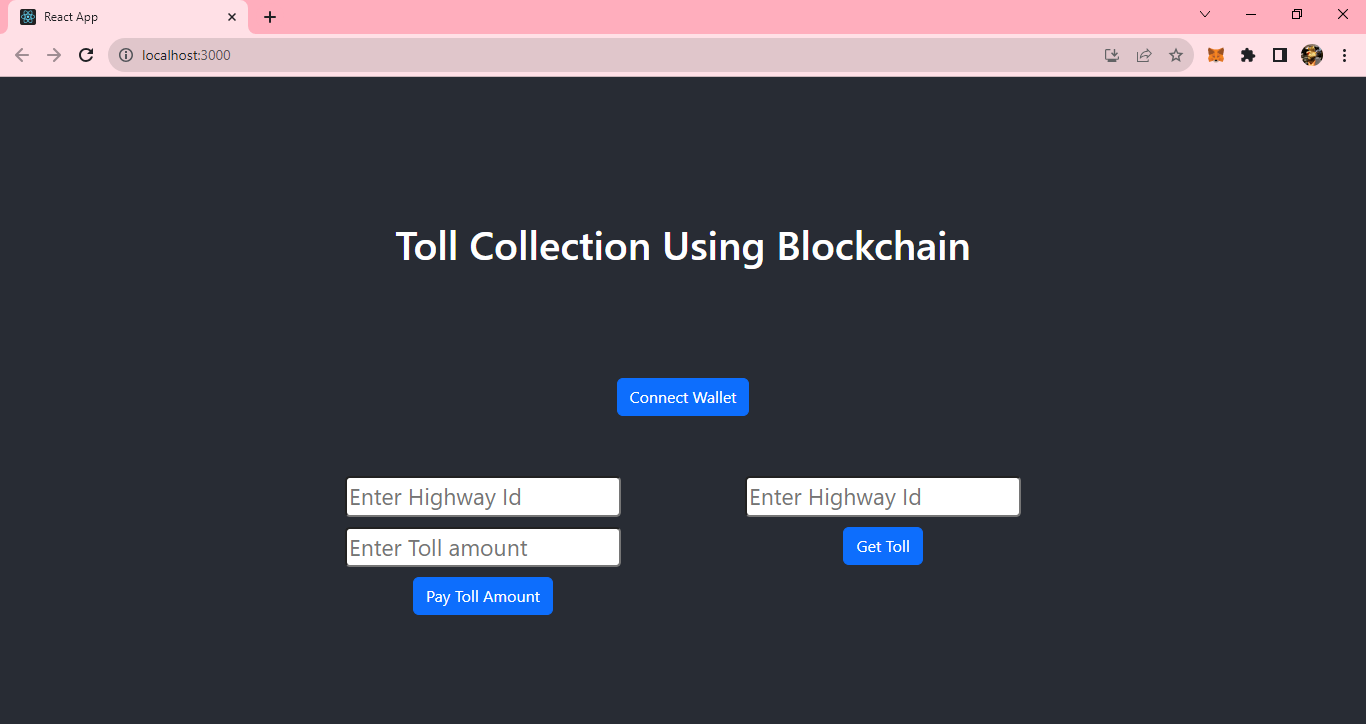
|  |  |  |  |
| --- | --- | --- | --- |
| 3. | Remix Ide platform explorting | Deploy the smart contract code  Deploy and run the transaction. By selecting the environment - inject the MetaMask. |  |
| 4 | Open file explorer | Open the extracted file and click on the folder.  Open src, and search for utiles.  Open cmd enter commands   1. npm install 2. npm bootstrap 3. npm start |  |
| 5 | {LOCALHOST IP ADDRESS | copy the address and open it to chrome so you can see the front end of your project. |  |

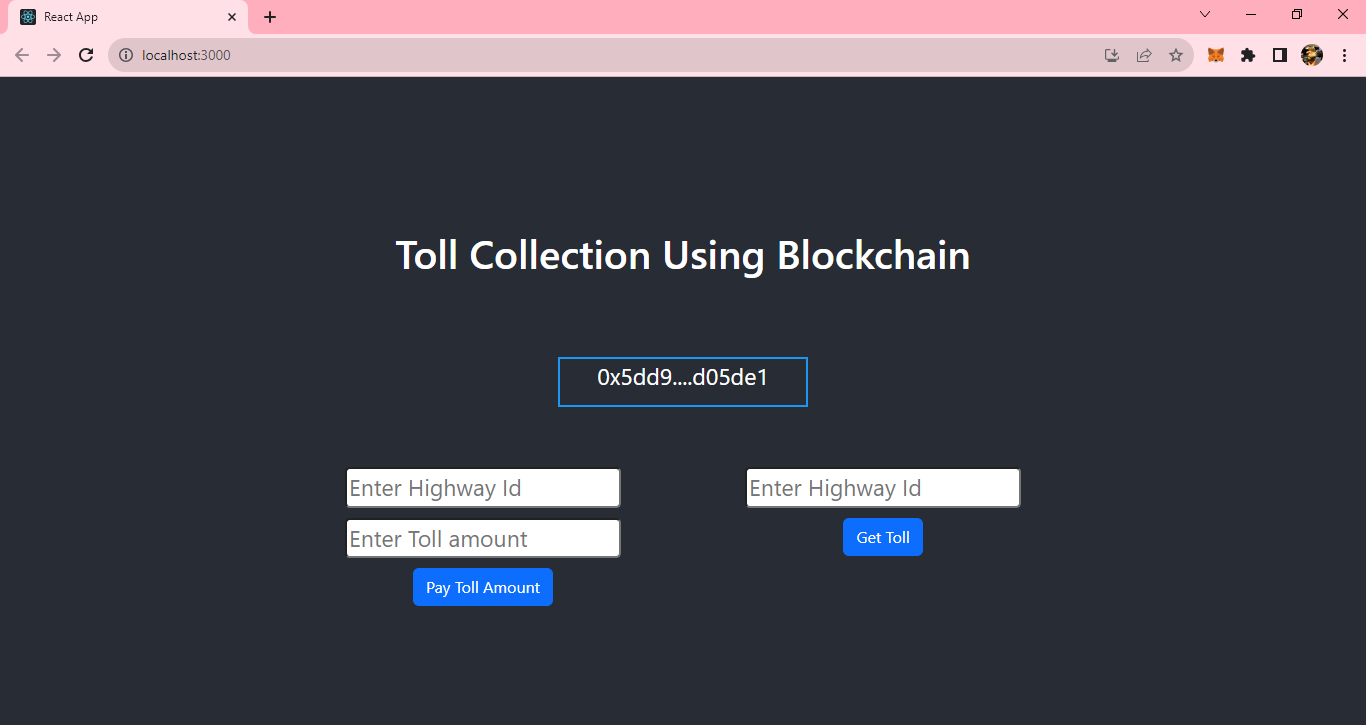
# **9. RESULT:**

## **9.1 OUTPUT SCREENSHOTS:**









# **10. ADVANTAGES & DISADVANTAGES**

## **ADVANTAGES**

* **Efficiency and Convenience:** Automating toll collection through blockchain technology reduces wait times and eliminates manual errors, providing a seamless and efficient experience for commuters.
* **Transparency and Accountability:** Blockchain's immutable ledger ensures transparent and traceable transactions, promoting accountability in toll collection and infrastructure management processes.
* **Enhanced Data Security:** Implementation of robust encryption and decentralized storage methods ensures the security and integrity of sensitive data, mitigating the risk of breaches.
* **Incentivized Adoption:** Token-based rewards incentivize users to adopt the new system, encouraging widespread acceptance and participation, thereby accelerating technology adoption.
* **Public-Private Collaboration:** Collaboration with private entities fosters innovation and investment, leveraging external expertise and resources to enhance the project's scope and impact.

## **DISADVANTAGES**

* **Initial Implementation Costs:** Setting up the blockchain infrastructure and developing user-friendly interfaces may involve substantial initial costs, posing a financial challenge to project initiation.
* **Technical Complexity:** Integrating blockchain technology requires specialized knowledge, making it essential to have skilled professionals on the team, which can be a potential limitation.
* **Regulatory Challenges**: Adhering to evolving regulations and legal frameworks related to blockchain technology can present hurdles, requiring continuous monitoring and compliance efforts.
* **User Adoption Barriers:** Users unfamiliar with blockchain technology may face challenges in understanding and adopting the new system, necessitating extensive user education and support.
* **System Scalability:** As the number of users grows, ensuring the scalability of the blockchain network and infrastructure becomes crucial to maintain optimal performance, which might pose scalability challenges in the long term.

# **11. CONCLUSION:**

In conclusion, the project "Tracking Public Infrastructure and Toll Payments using blockchain" represents a groundbreaking leap toward a more efficient, secure, and user-friendly transportation system. By leveraging the power of blockchain technology, the project addresses the longstanding challenges of manual toll collection and opaque infrastructure management. The implementation of smart contracts ensures seamless, automated toll transactions, significantly reducing wait times and errors. Moreover, the transparency embedded in blockchain promotes accountability, instilling trust among stakeholders. The introduction of user-friendly interfaces and token-based rewards not only enhances user experience but also incentivizes rapid technology adoption.

However, this project's success hinges on continuous user education and seamless integration with existing systems. Overcoming initial adoption hurdles and ensuring regulatory compliance are critical steps. Collaborations with private entities and a vigilant eye on technological advancements will be pivotal for long-term sustainability.

In essence, this initiative heralds a new era in transportation management, promising a future where commuters experience swift, secure transactions, while stakeholders benefit from transparent, accountable operations. As this project paves the way for smarter, more accessible public infrastructure management, it stands as a testament to the transformative potential of blockchain technology in shaping the future of transportation systems.

# **12. FUTURE SCOPE:**

The project's future scope is vast and promising, offering opportunities for continued innovation and growth. One avenue for expansion lies in integrating emerging technologies like the Internet of Things (IoT) devices, enabling real-time data collection on road conditions and traffic patterns. Machine learning algorithms can be incorporated to predict maintenance needs, and optimize infrastructure management. Additionally, the project could explore interoperability with other blockchain networks, facilitating seamless transactions across various platforms.

Furthermore, extending the project's reach to encompass smart city initiatives could revolutionize urban mobility. Collaborations with local authorities and public transportation systems can lead to integrated, multimodal transportation solutions. The concept of digital identities and decentralized authentication could enhance user experience, ensuring secure and personalized services.

# **13. APPENDIX:**

## **Source Code:**

**Toll Collection.sol**

// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract tollCollection{

struct TollData {

uint timestamp;

address collectedBy;

uint amount;

}

mapping(address => mapping(uint => TollData)) public tolls;

function payTollAmount(uint highwayId, uint \_amount) public {

// TollData memory newToll = TollData(block.timestamp, msg.sender, amount);

tolls[msg.sender][highwayId].timestamp = block.timestamp ;

tolls[msg.sender][highwayId].collectedBy = msg.sender;

tolls[msg.sender][highwayId].amount += \_amount;

}

function getToll(uint highwayId) public view returns (TollData memory) {

return tolls[msg.sender][highwayId];

}

// function updateToll(uint highwayId, uint amount) public {

// require(

// tolls[msg.sender][highwayId].timestamp > 0,

// "Toll data not found."

// );

// tolls[msg.sender][highwayId].amount = amount;

// }

}

**Connector.js**

const { ethers } = require("ethers");

const abi = [

 {

  "inputs": [

   {

    "internalType": "uint256",

    "name": "highwayId",

    "type": "uint256"

   }

  ],

  "name": "getToll",

  "outputs": [

   {

    "components": [

     {

      "internalType": "uint256",

      "name": "timestamp",

      "type": "uint256"

     },

     {

      "internalType": "address",

      "name": "collectedBy",

      "type": "address"

     },

     {

      "internalType": "uint256",

      "name": "amount",

      "type": "uint256"

     }

    ],

    "internalType": "struct tollCollection.TollData",

    "name": "",

    "type": "tuple"

   }

  ],

  "stateMutability": "view",

  "type": "function"

 },

 {

  "inputs": [

   {

    "internalType": "uint256",

    "name": "highwayId",

    "type": "uint256"

   },

   {

    "internalType": "uint256",

    "name": "\_amount",

    "type": "uint256"

   }

  ],

  "name": "payTollAmount",

  "outputs": [],

  "stateMutability": "nonpayable",

  "type": "function"

 },

 {

  "inputs": [

   {

    "internalType": "address",

    "name": "",

    "type": "address"

   },

   {

    "internalType": "uint256",

    "name": "",

    "type": "uint256"

   }

  ],

  "name": "tolls",

  "outputs": [

   {

    "internalType": "uint256",

    "name": "timestamp",

    "type": "uint256"

   },

   {

    "internalType": "address",

    "name": "collectedBy",

    "type": "address"

   },

   {

    "internalType": "uint256",

    "name": "amount",

    "type": "uint256"

   }

  ],

  "stateMutability": "view",

  "type": "function"

 }

]

if (!window.ethereum) {

 alert('Meta Mask Not Found')

 window.open("https://metamask.io/download/")

}

export const provider = new ethers.providers.Web3Provider(window.ethereum);

export const signer = provider.getSigner();

export const address = "0x096D2954cabe7F33FBA8e890A3fb96498D51f9E6"

export const contract = new ethers.Contract(address, abi, signer)

**Home.js**

import React, { useState } from "react";

import { Button, Container, Row, Col } from 'react-bootstrap';

import 'bootstrap/dist/css/bootstrap.min.css';

import { contract } from "./connector";

function Home() {

   const [Id, setId] = useState("");

   const [TollAmount, setTollAmount] = useState("");

   const [Manufacturer, setManufacturer] = useState("");

   const [date, setDate] = useState("");

   const [TranId, setTranId] = useState("");

   const [Owner, setOwner] = useState("");

   const [BookId, setBookId] = useState("");

   const [BookDet, setBookDet] = useState("");

   const [Batch, setBatch] = useState("");

   const [Qty, setQty] = useState("");

   const [Cus, setCus] = useState("");

   const [Wallet, setWallet] = useState("");

   const handleId = (e) => {

      setId(e.target.value)

   }

   const handleTollAmount = (e) => {

      setTollAmount(e.target.value)

   }

   const handleToll = async () => {

      try {

         let tx = await contract.payTollAmount(Id.toString(), TollAmount.toString())

         let wait = await tx.wait()

         alert(wait.transactionHash)

         console.log(wait);

      } catch (error) {

         alert(error)

      }

   }

   const handleDrugId = (e) => {

      setTranId(e.target.value)

   }

   const handleNewOwner = (e) => {

      setOwner(e.target.value)

   }

   const handleTransfer = async () => {

      try {

         let tx = await contract.transferDrugOwnership(TranId.toString(), Owner)

         let wait = await tx.wait()

         console.log(wait);

         alert(wait.transactionHash)

      } catch (error) {

         alert(error)

      }

   }

   const handleTollDetailsId = (e) => {

      setBookId(e.target.value)

   }

   const handleDrugDetails = async () => {

      try {

         let tx = await contract.getToll(BookId.toString())

         let arr = []

         tx.map(e => {

            arr.push(e)

         })

         console.log(tx);

         setBookDet(arr)

      } catch (error) {

         alert(error)

         console.log(error);

      }

   }

   const handleWallet = async () => {

      if (!window.ethereum) {

         return alert('please install metamask');

      }

      const addr = await window.ethereum.request({

         method: 'eth\_requestAccounts',

      });

      setWallet(addr[0])

   }

 return (

  <div>

   <h1 style={{ marginTop: "30px", marginBottom: "80px" }}>Toll Collection Using Blockchain</h1>

       {!Wallet ?

          <Button onClick={handleWallet} style={{ marginTop: "30px", marginBottom: "50px" }}>Connect Wallet </Button>

          :

          <p style={{ width: "250px", height: "50px", margin: "auto", marginBottom: "50px", border: '2px solid #2096f3' }}>{Wallet.slice(0, 6)}....{Wallet.slice(-6)}</p>

       }

   <Container>

    <Row>

     <Col style={{marginRight:"100px"}}>

      <div>

       <input style={{ marginTop: "10px", borderRadius: "5px" }} onChange={handleId} type="number" placeholder="Enter Highway Id" value={Id} /> <br />

      <input style={{ marginTop: "10px", borderRadius: "5px" }} onChange={handleTollAmount} type="number" placeholder="Enter Toll amount" value={TollAmount} /> <br />

       <Button onClick={handleToll} style={{ marginTop: "10px" }} variant="primary"> Pay Toll Amount</Button>

      </div>

     </Col>

             <Col >

                <div style={{ margin: "auto" }}>

                   <input style={{ marginTop: "10px", borderRadius: "5px" }} onChange={handleTollDetailsId} type="number" placeholder="Enter Highway Id" value={BookId} /><br />

                   <Button onClick={handleDrugDetails} style={{ marginTop: "10px" }} variant="primary">Get Toll</Button>

                   {BookDet ? BookDet?.map(e => {

                      return <p>{e.toString()}</p>

                   }) : <p></p>}

                </div>

             </Col>

   </Row>

   </Container>

  </div>

 )

}

export default Home;

**Project Demo Link:**

**Demo video link:** <https://youtu.be/hacYcMeoNc8>

https://drive.google.com/drive/folders/1TT-rF\_Ee1ocsMhKm557h3nZlNazItXkp?usp=drive\_link