|  |  |
| --- | --- |
|  | **DEPARTMENT OF COMPUTER ENGINEERING** |

**Collaborative Project Report**

|  |  |
| --- | --- |
| Semester | B.E. Semester VII – Computer Engineering |
| Subject | Blockchain |
| Subject Professor In-charge | Prof. Swapnil S. Sonawane |
| Assisting Teachers | Prof. Swapnil S. Sonawane |

|  |  |
| --- | --- |
| Roll Number | Name of Students |
| 21102A0003 | Omkar Patil |
| 21102A0005 | Pranav Redij |
| 21102A0006 | Sahil Pokharkar |
| 21102A0014 | Deep Salunkhe |

**Name of the Project:**

**HardHat-Todo**

**Project Details:**

**Project Title: HardHat-Todo**

**Overview**

The Decentralized Todo List Application allows users to manage their tasks using Ethereum smart contracts. This project demonstrates how blockchain technology can be used to create a transparent and tamper-proof application for task management.

**Technologies Used**

* **Frontend**: React, JavaScript, CSS
* **Blockchain**: Ethereum
* **Smart Contracts**: Solidity
* **Libraries**: ethers.js, Web3Provider
* **Development Tools**: Hardhat ,MetaMask

**Features**

1. **Add Todo**: Users can create new tasks, which are stored on the Ethereum blockchain.
2. **View Todos**: Users can view all their tasks fetched from the smart contract.
3. **Complete Todo**: Users can mark tasks as completed, which updates the task state in the smart contract.
4. **Error Handling**: The application includes error handling for interactions with the blockchain, providing feedback to users when errors occur.

**Smart Contract**

* **Contract Name**: Todo
* **Functions**:
  + createTask(string content): Adds a new task to the list.
  + getAllTasks(): Retrieves all tasks stored in the smart contract.
  + toggleTaskCompletion(uint index): Toggles the completion status of a task.

**User Interface**

* The user interface is simple and intuitive, allowing users to add and view their todos easily. It includes input fields for new tasks and buttons to complete them.

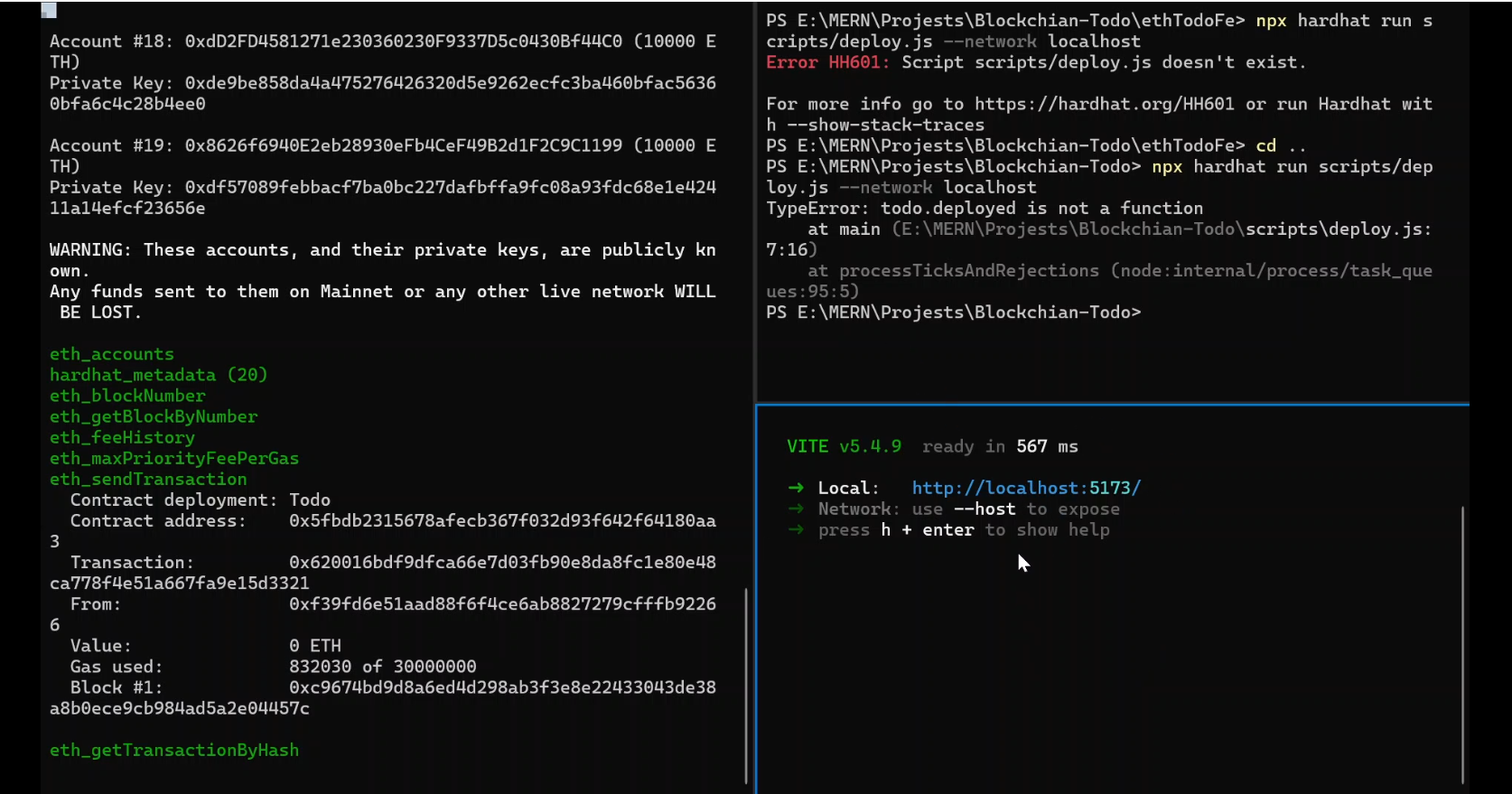
**Challenges Faced**

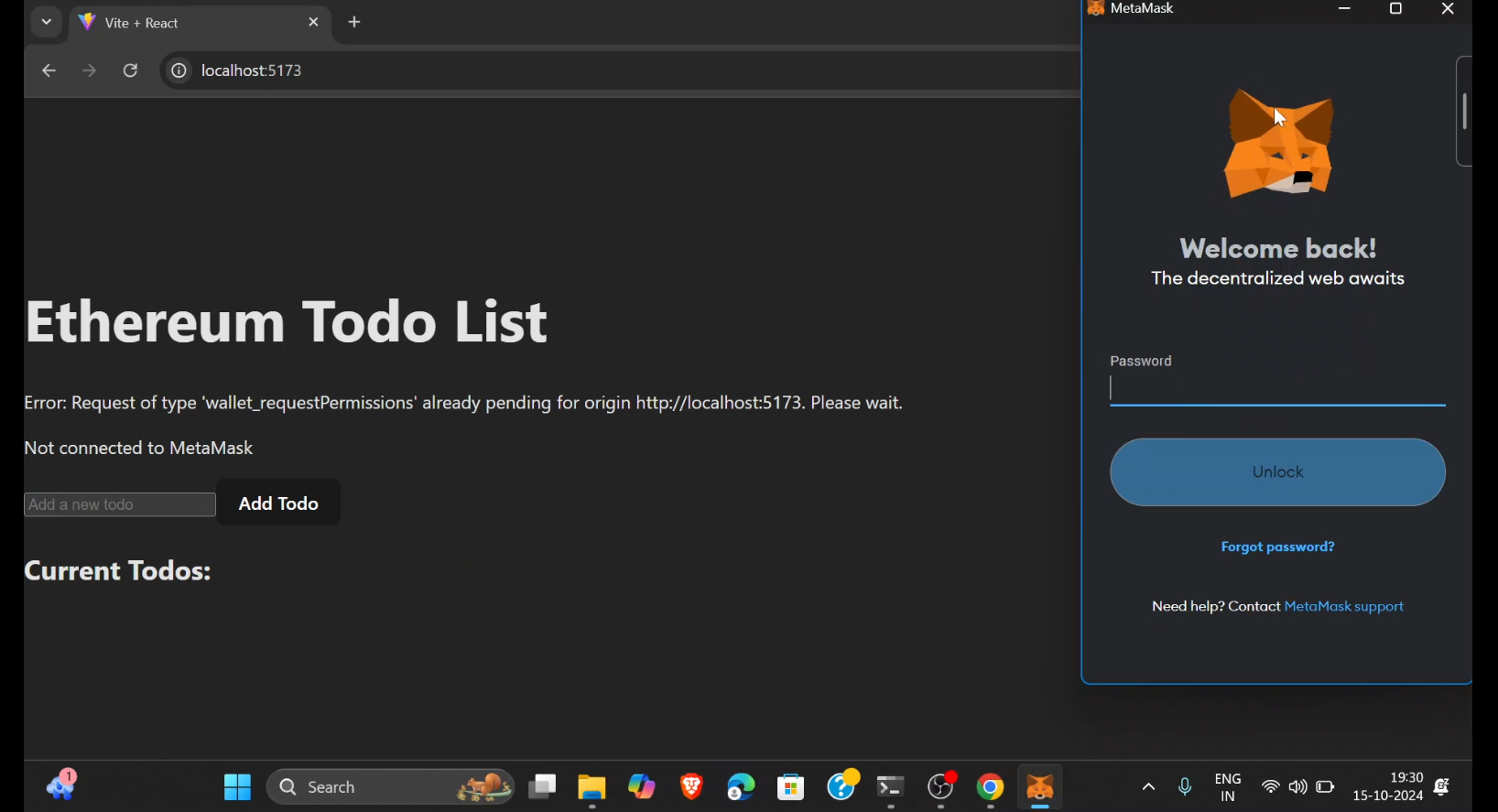
* **Blockchain Interaction**: Understanding how to interact with Ethereum smart contracts using ethers.js posed some initial challenges.
* **Handling Asynchronous Operations**: Managing state and asynchronous calls effectively, especially when fetching data from the blockchain.
* **Error Handling**: Implementing comprehensive error handling for user interactions with the smart contract.

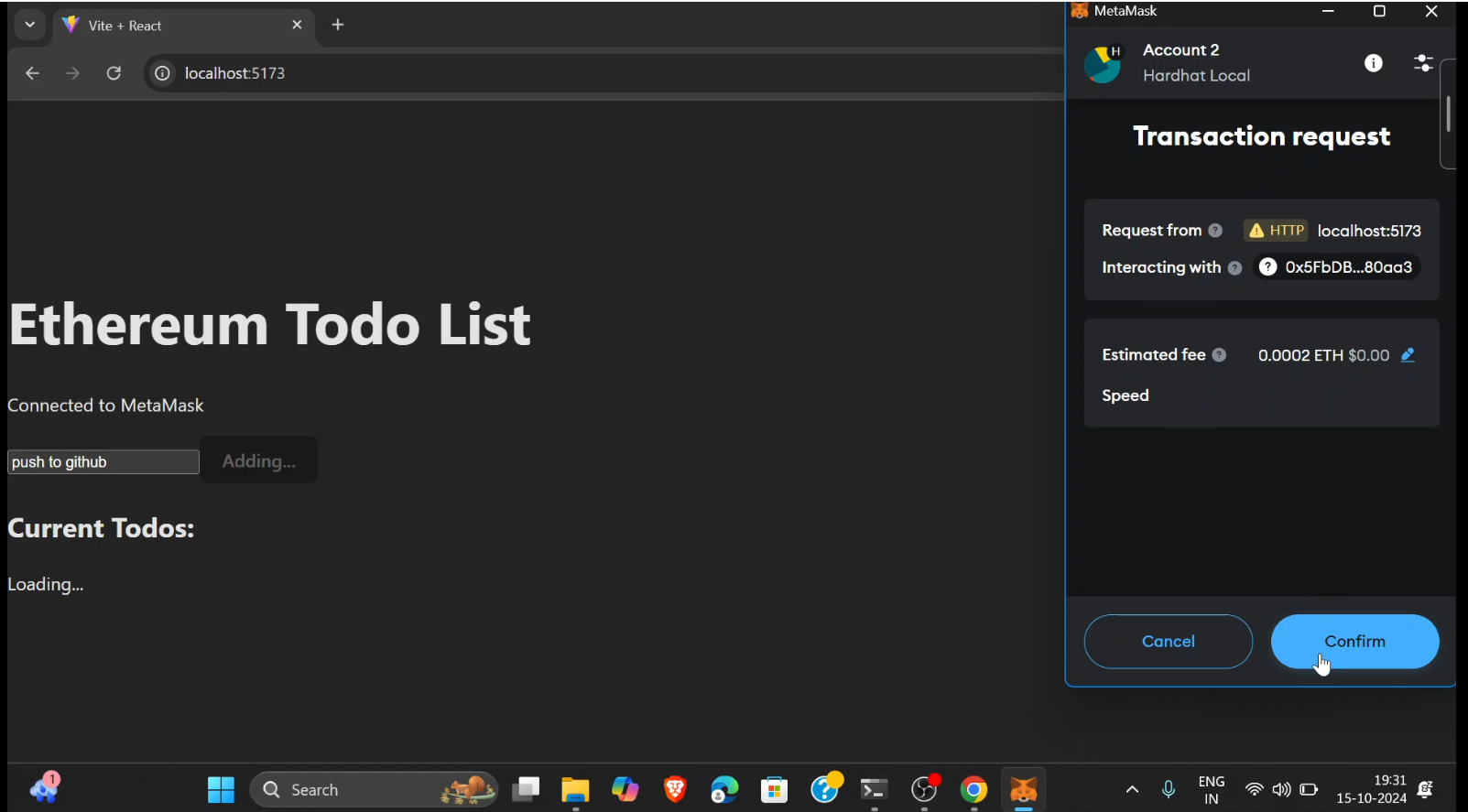
**GitHub Repository Link (Public):**

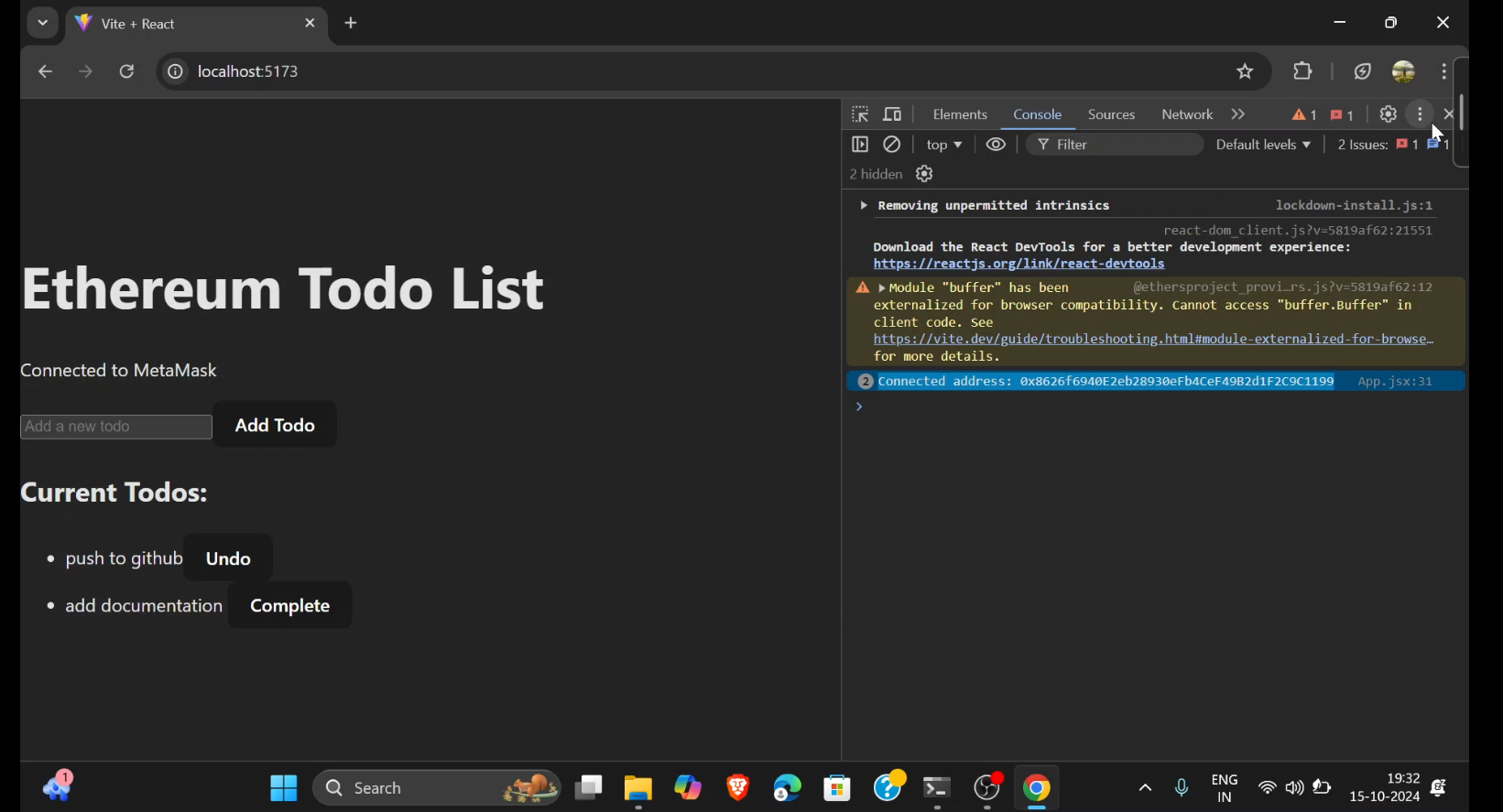
[**https://github.com/deepsalunkhee/HardHat-Todo**](https://github.com/deepsalunkhee/HardHat-Todo)

**Output Screenshots:**

****

****

****

****