**Voting application with Blockchain**

**Professional Certificate Program in Blockchain by IIT Kanpur**

**Rajendra kumar,** [**rkumar084@gmail.com**](mailto:rkumar084@gmail.com)

**Introduction**

This document provides details of implementation of the capstone project of online voting application using ethereum blockchain. The scope is to design, implement and test necessary back-end calls for the voting application.

**Implementation**

Smart contracts for the backend is implemented using Solidity and the source code is provided in the below embedded document. Below are the programming languages and tools which are used for implementation

* Solidity
* RemixIDE
* Ganache

Important constructs of Solidity which are used in the implementation

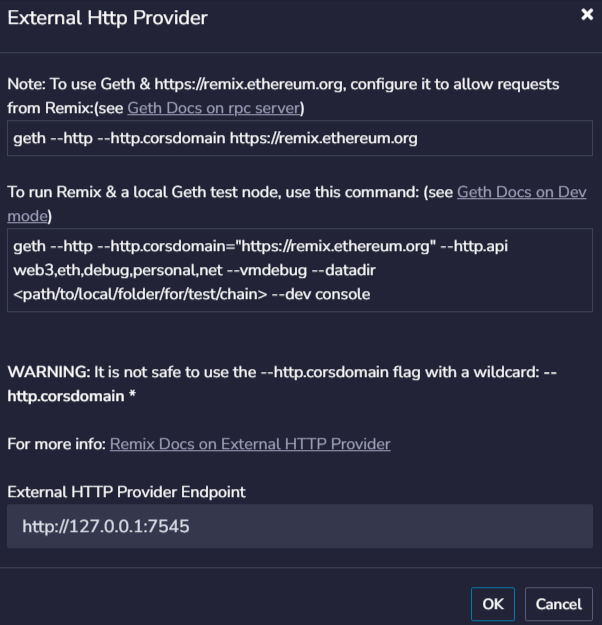
* Smart contracts
* Error handling using *require*
* Mapping variables
* Dynamic arrays
* Custom modifiers
* User defined data types
* Enums



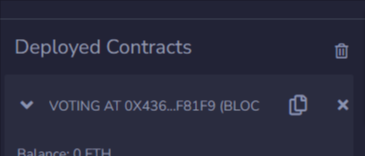
**Configuration and deployment**

Remix IDE is configured to connect to Ganache running locally as shown in the screenshot. Smart contract is compiled and deployed on to the local ethereum blockchain.

Configuring Ganache to Remix

****

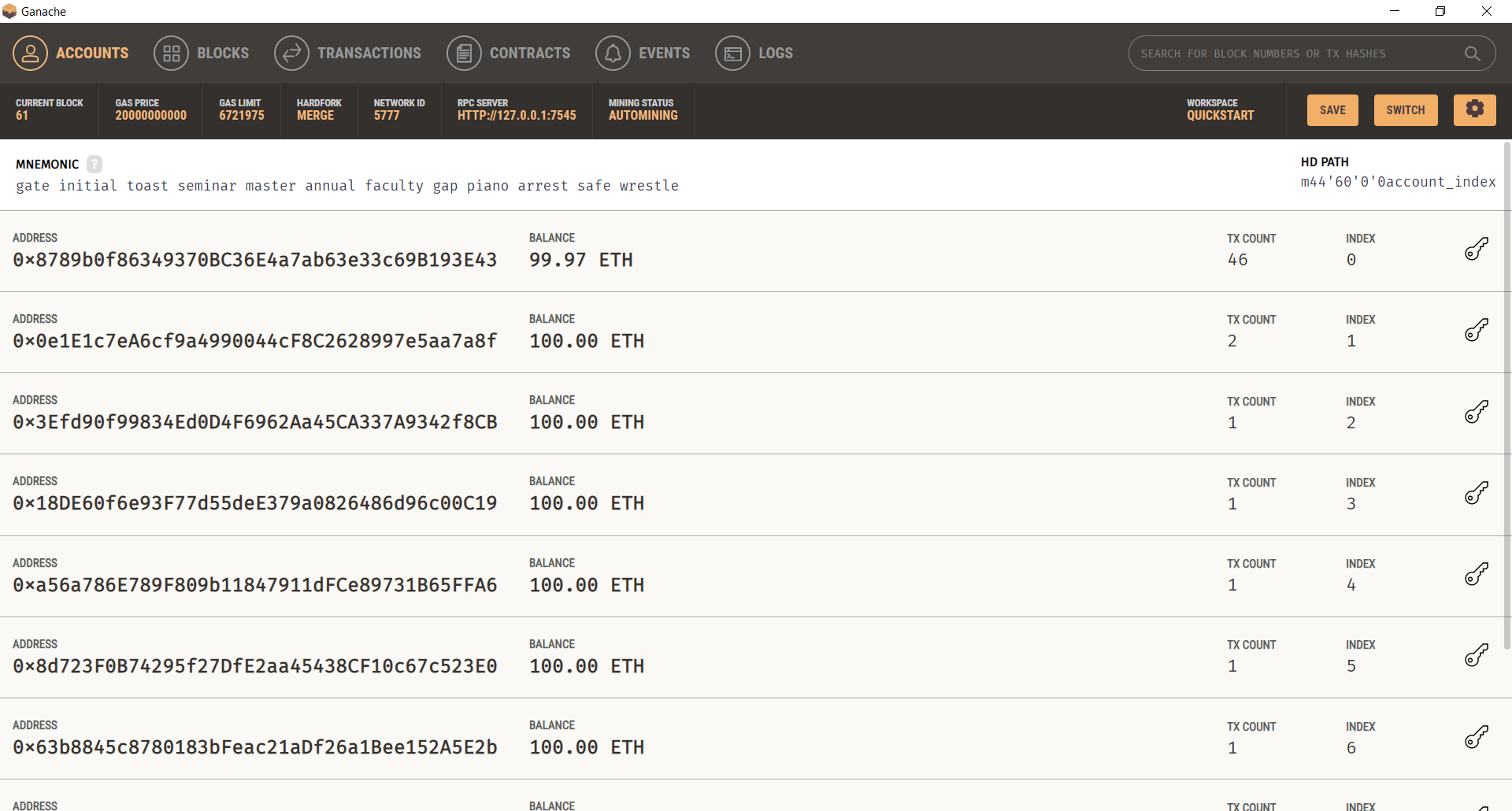
Address of the smart contract after deployment

****

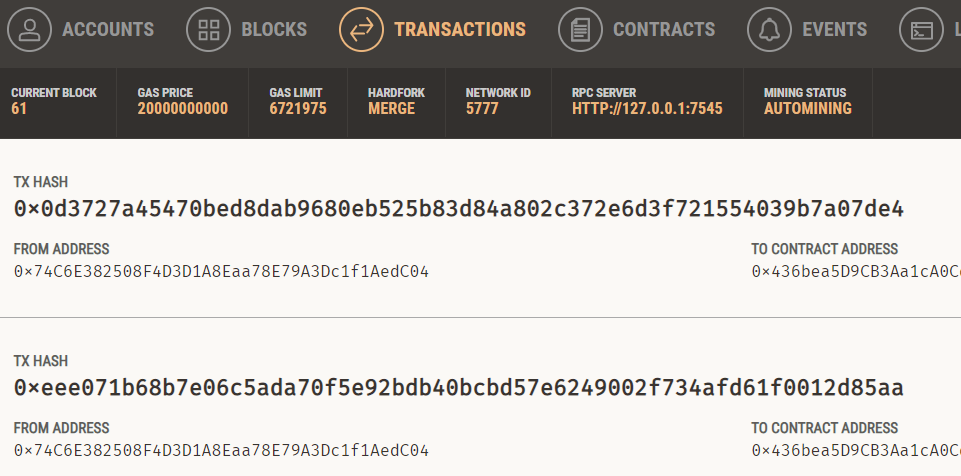
**Testing**

Remix IDE is used for connecting to the local Ganache network and calls to smart contract methods are initiated from Remix only. Addresses for test data are taken from the local ethereum block chain. One of the addresses is hardcoded as admin who is responsible for conducting elections and have certain privileged operations to perform. The sender address to test scenarios relating to admin operations, delegation and voting. Below are few screenshots showing test data, transactions, and blocks created after testing.

Admin account and other Accounts taken for testing



Transactions and Contract address



Blocks created for the transactions performed.

