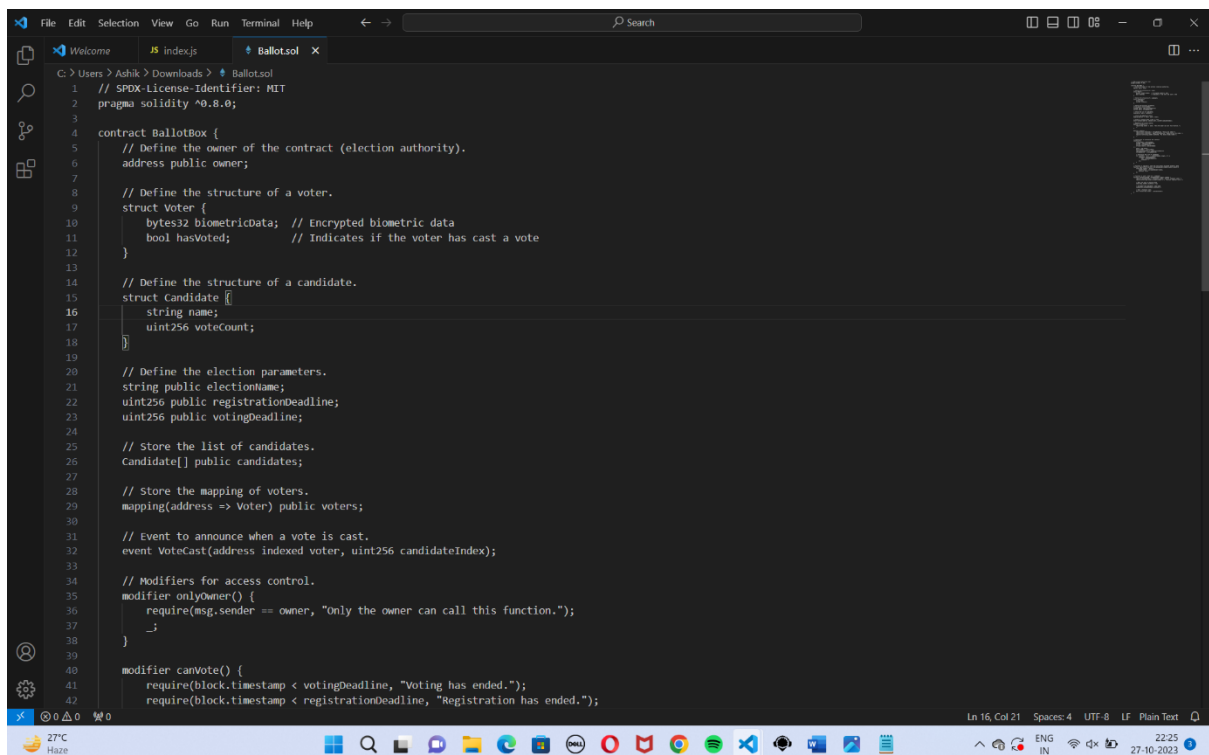


DOWNLOAD THE ZIP FILE OF THE PROJECT AND EXTRACT IT AND OPEN IN VS CODE

DATE	20 SEPTEMBER 2023
TEAM ID	NM2023TMID02213
PROJECT NAME	BIOMETRIC SECURITY SYSTEM FOR VOTING PLATFORM
MAXIMUM MARKS	4 MARKS



```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract BallotBox {
5     // Define the owner of the contract (election authority).
6     address public owner;
7
8     // Define the structure of a voter.
9     struct Voter {
10         bytes32 biometricData; // Encrypted biometric data
11         bool hasVoted; // Indicates if the voter has cast a vote
12     }
13
14     // Define the structure of a candidate.
15     struct Candidate {
16         string name;
17         uint256 voteCount;
18     }
19
20     // Define the election parameters.
21     string public electionName;
22     uint256 public registrationDeadline;
23     uint256 public votingDeadline;
24
25     // Store the list of candidates.
26     Candidate[] public candidates;
27
28     // Store the mapping of voters.
29     mapping(address => Voter) public voters;
30
31     // Event to announce when a vote is cast.
32     event VoteCast(address indexed voter, uint256 candidateIndex);
33
34     // Modifiers for access control.
35     modifier onlyOwner() {
36         require(msg.sender == owner, "Only the owner can call this function.");
37         _;
38     }
39
40     modifier canVote() {
41         require(block.timestamp < votingDeadline, "Voting has ended.");
42         require(block.timestamp < registrationDeadline, "Registration has ended.");
43     }
```

