

### PROGRAM:

```
App = {

  web3Provider: null,
  contracts: {},
  account: '0x0',
  hasVoted: false,

  init: function() {
    return App.initWeb3();
  },

  initWeb3: function() {
    // TODO: refactor conditional
    if (typeof web3 !== 'undefined') {
      // If a web3 instance is already provided by Meta Mask.
      App.web3Provider = web3.currentProvider;
      web3 = new Web3(web3.currentProvider);
    } else {
      // Specify default instance if no web3 instance provided
      App.web3Provider = new
Web3.providers.HttpProvider('http://localhost:7545');
      web3 = new Web3(App.web3Provider);
    }

    return App.initContract();
  }
}
```

```
},
```

```
initContract: function() {
```

```
  $.getJSON("Election.json", function(election) {
```

```
    // Instantiate a new truffle contract from the artifact
```

```
    App.contracts.Election = TruffleContract(election);
```

```
    // Connect provider to interact with contract
```

```
    App.contracts.Election.setProvider(App.web3Provider);
```

```
    App.listenForEvents();
```

```
    return App.render();
```

```
  });
```

```
},
```

```
// Listen for events emitted from the contract
```

```
listenForEvents: function() {
```

```
  App.contracts.Election.deployed().then(function(instance) {
```

```
    // Restart Chrome if you are unable to receive this event
```

```
    // This is a known issue with Metamask
```

```
    // https://github.com/MetaMask/metamask-extension/issues/2393
```

```
    instance.votedEvent({}, {
```

```
      fromBlock: 0,
```

```
      toBlock: 'latest'
```

```
}).watch(function(error, event) {  
  console.log("event triggered", event)  
  // Reload when a new vote is recorded  
  App.render();  
});  
});  
},
```

```
render: function() {
```

```
  //document.write('<script type="text/javascript" src="../../lol.js"></script>');  
  var electionInstance;  
  //console.log(names[0]);  
  var loader = $("#loader");  
  var content = $("#content");  
  //var request=require('request/request.js');
```

```
  loader.show();
```

```
  content.hide();
```

```
  // Load account data
```

```
  web3.eth.getCoinbase(function(err, account) {
```

```
    if (err === null) {
```

```
      App.account = account;
```

```
$("#accountAddress").html("Your Account: " + account);  
}  
});
```

```
// Load contract data
```

```
App.contracts.Election.deployed().then(function(instance) {  
    electionInstance=instance;  
    // for(var j=1;j<=names.length;j++){  
    // return electionInstance.addCandidate(names[j-1]);  
    // }  
    return electionInstance.candidatesCount();  
}).then(function(candidatesCount) {  
    // var candidatesResults = $("#candidatesResults");  
    // candidatesResults.empty();  
    var voteList=[];  
    var sum=0;  
    var candidatesSelect = $('#candidatesSelect');  
    candidatesSelect.empty();
```

```
    var candidateOption1="<option disabled selected value>select</option>";  
    candidatesSelect.append(candidateOption1);  
    //if(sessionStorage.getItem("session")==="1"){  
    for (var i = 1; i <= candidatesCount; i++) {  
        electionInstance.candidates(i).then(function(candidate) {
```

```

    var id = candidate[0];
    var name = names[id-1];
    var voteCount = candidate[2];
    voteList[id-1]=voteCount;

    //electionInstance.addCandidate(name);

    // Render candidate Result
    // var candidateTemplate = "<tr><td>" + name + "</td><td>" + voteCount +
"</td></tr>"
    // candidatesResults.append(candidateTemplate);

    // Render candidate ballot option
    var candidateOption = "<option value='" + id + "' >" + name + "</ option>"
    candidatesSelect.append(candidateOption);
});
}

//sessionStorage.setItem("session",0);
//}

for (var i = 1; i <= candidatesCount; i++) {
    electionInstance.candidates(i).then(function(candidate) {
        var id = candidate[0];
        var voteCount = candidate[2];
        voteList[id-1]=voteCount;
        //sum=sum+voteCount;
    });
}

```

```
});  
}
```

```
var ctx = $('#myChart')[0].getContext('2d');
```

```
var myChart = new Chart(ctx, {  
  type: 'bar',  
  data: {  
    labels: names,  
    datasets: [{  
      label: '# of Votes',  
      data: voteList,  
      backgroundColor:"",  
      borderColor:"blue",  
      borderWidth: 1  
    }]  
  },  
  options: {  
    scales: {  
      yAxes: [{  
        ticks: {  
          beginAtZero: true
```

```
    }  
  }  
}  
}  
});
```

```
var ctx = $('#myChart1')[0].getContext('2d');
```

```
// //var sum=0;
```

```
// console.log(sum);
```

```
var myChart = new Chart(ctx, {  
  type: 'pie',  
  data: {  
    labels: ["Voted", "Did not vote"],  
    datasets: [  
      {  
        fill: true,  
        backgroundColor: [  
          'green',  
          'pink'],  
        data: [sum,3-sum],
```

```
//Notice the borderColor
    borderColor:  ['green', 'pink'],
    borderWidth: [2,2]
  }
]
},
options: {
  title: {
    display: true,
    text: 'Voter Turnout',
    position: 'top'
  },
  rotation: -0.7 * Math.PI
}
});

return electionInstance.voters(App.account);
}).then(function(hasVoted) {
  // Do not allow a user to vote
  if(hasVoted) {
    $('#form').hide();
  }
  loader.hide();
```



```
        content.show();
    }).catch(function(error) {
        console.log(error);
    });
},
```

```
showInfo:function(){
    var candidateId=parseInt($('#candidatesSelect').val());
    var candidateInfo=$('#candidateInfo');
    candidateInfo.empty();

    candidateInfo.append("Name:"+names[candidateId-1]+<br>"+
    "Party:"+party[candidateId-1]+<br>"+
    "Criminal Cases pending:"+criminal[candidateId-1]+
    "<br>"+
    "Age:"+age[candidateId-1]+<br>"+
    "Assets:"+asset[candidateId-1]+<br>"+
    "Liabilities:"+liab[candidateId-1]+<br>");

    },
```

```
castVote: function() {
    var candidateId = $('#candidatesSelect').val();
    App.contracts.Election.deployed().then(function(instance) {
        return instance.vote(candidateId, { from: App.account });
    }).then(function(result) {
        // Wait for votes to update
        $('#content').hide();
    });
}
```

```
    $("#loader").show();  
  }).catch(function(err) {  
    console.error(err);  
  });  
}  
};
```

```
$(function() {  
  sessionStorage.setItem("session","1");  
  $(window).load(function() {  
    App.init();  
  });  
});
```