Developing User Interface for your Application using truffle and webpack:

Remember Truffle compiles your contracts, deploys them and generate necessary files for web interface.

Webpack:

At its core, *webpack* is a *static module bundler* for modern JavaScript applications. When webpack processes your application, it recursively builds a *dependency graph* that includes every module your application needs, then packages all of those modules into one or more *bundles*.

For more information please visit https://webpack.js.org/concepts/

We will use webpack 2.2.0.

Inside your project Voting:

$ cd Voting

Install truffle locally

$npm init

$ npm install truffle –save-dev

$npm install webpack@2.2.0

We are not going to build a very fancy project UI. It will only contain an html file(index.html) along with a javascript file(app.js).

Create app and js directories

$ mkdir -p app/js

then app.js file

$ touch app/js/app.js

Create index.html with your favorite editor and write the following code in it:

```
<!doctype html>
<html lang="en" dir="ltr">
<head>
<title>Voting App</title>
<meta charset="utf-8">




</head>
<body>
<div>
<script src="js/app.js"></script>
 Vote<input name="Candidate" type="text" placeholder="" />
    <button id="vote">Now</button>.
</div>
<div id="candidate1"></div>
```

```
<div id="candidate2"></div>



</body>
</html>
```

We need few more packages for this UI to work:

web3: Is used to communicate with ethereum node or deployed smart contract from JavaScript or web application.

Truffle-contract: Instantiate compiled contracts

blue-bird: Promise library for JavaScript

jquery:*jQuery* is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is DOM manipulator.

Install them all:

$ npm install web3 truffle-contract bluebird jquery –save

Install file-loader to port index.html to app.js

$ npm install file-loader –save-dev

Now Open ./app/js/app.js file in your favorite editor and add:

```
const Promise = require("bluebird");
const truffleContract = require("truffle-contract");
const $ = require("jquery");
// Not to forget our built contract
const voteJson = require("../../build/contracts/Voting.json");
require("file-loader?name=../index.html!../index.html");



// Supports Mist, and other wallets that provide 'web3'.
if (typeof web3 !== 'undefined') {
    // Use the Mist/wallet/Metamask provider.
    window.web3 = new Web3(web3.currentProvider);
} else {
    // Your preferred fallback.
    window.web3 = new Web3(new Web3.providers.HttpProvider('http://localhost:8545'));
}
```

You can copy and paste this section of the code. It just act as library to help you promisify web3. You should also download utils.js from moodle. It returns transaction receipt after transaction is successfully mined.

```
web3.eth.getTransactionReceiptMined = require("./utils.js");


function sequentialPromise(promiseArray) {
    const result = promiseArray.reduce(
        (reduced, promise, index) => {
            reduced.results.push(undefined);
            return {
                chain: reduced.chain
                    .then(() => promise)
                    .then(result => reduced.results[ index ] = result),
                results: reduced.results
            };
        },
        {
            chain: Promise.resolve(),
            results: []
        });
    return result.chain.then(() => result.results);
}
sequentialPromise([
    Promise.resolve(web3.eth), Promise.resolve({ suffix: "Promise" })
]).then(console.log);
web3.eth.getAccountsPromise = function () {
    return new Promise(function (resolve, reject) {
```

```
        web3.eth.getAccounts(function (e, accounts) {
            if (e != null) {
                reject(e);
            } else {
                resolve(accounts);
            }
        });
    });
};
```

Here you add Main part of your code:

Prepare your voting contract:

```
const Vote = truffleContract(voteJson);
Vote.setProvider(web3.currentProvider);
```

```
window.addEventListener('load', function() {
  //vote is button id, votefunc is the function that will be executed when button is pressed
      $("#vote").click(votefunc);
      let deployed;
      var candidate1_votes;
```

```
    var candidate2_votes;
    return Vote.deployed()
      .then(_deployed => { //deploing the contract
        deployed = _deployed;
      //   calling Clist function to return list of candidates in an array candidatelist
      //then showing candidates with their number of votes in the html page
        return _deployed.Clist.call().then(candidatelist=>{
          _deployed.totalVotesFor.call(candidatelist[0]).then(votenumer1=>{
            $("#candidate1").html("First Candidate is " + web3.toAscii(candidatelist[0])+" With total number of votes "+votenumer1);
          });
          _deployed.totalVotesFor.call(candidatelist[1]).then(votenumer2=>{
            $("#candidate2").html("First Candidate is " + web3.toAscii(candidatelist[1])+" With total number of votes "+votenumer2);
          });



        })

      });
});

//Function that will be executed when button is pressed
const votefunc = function() {
```

```
};
```

Just make sure you use 0.18.0 version of web3. Edit package.json and make sure you have :

"web3": "^0.18.0"

Edit webpack.config.js and its content should look like this:

```
module.exports = {
    entry: "./app/js/app.js",
    output: {
        path: __dirname + "/build/app/js",
        filename: "app.js"
    },
    module: {
        loaders: []
    }
};
```

Lets compile and deploy our contract first:

open a new terminal and

make sure testrpc is running:

$testrpc

return to first terminal and:

$./node_modules/.bin/truffle migrate

Bundle all your webfiles:

$./node_modules/.bin/webpack

Eye ball it:

while being inside Voting directory execute:

$php -S 0.0.0.0:8000 -t ./build/app/

Now open web browser and inside address bar write down:

localhost:8000