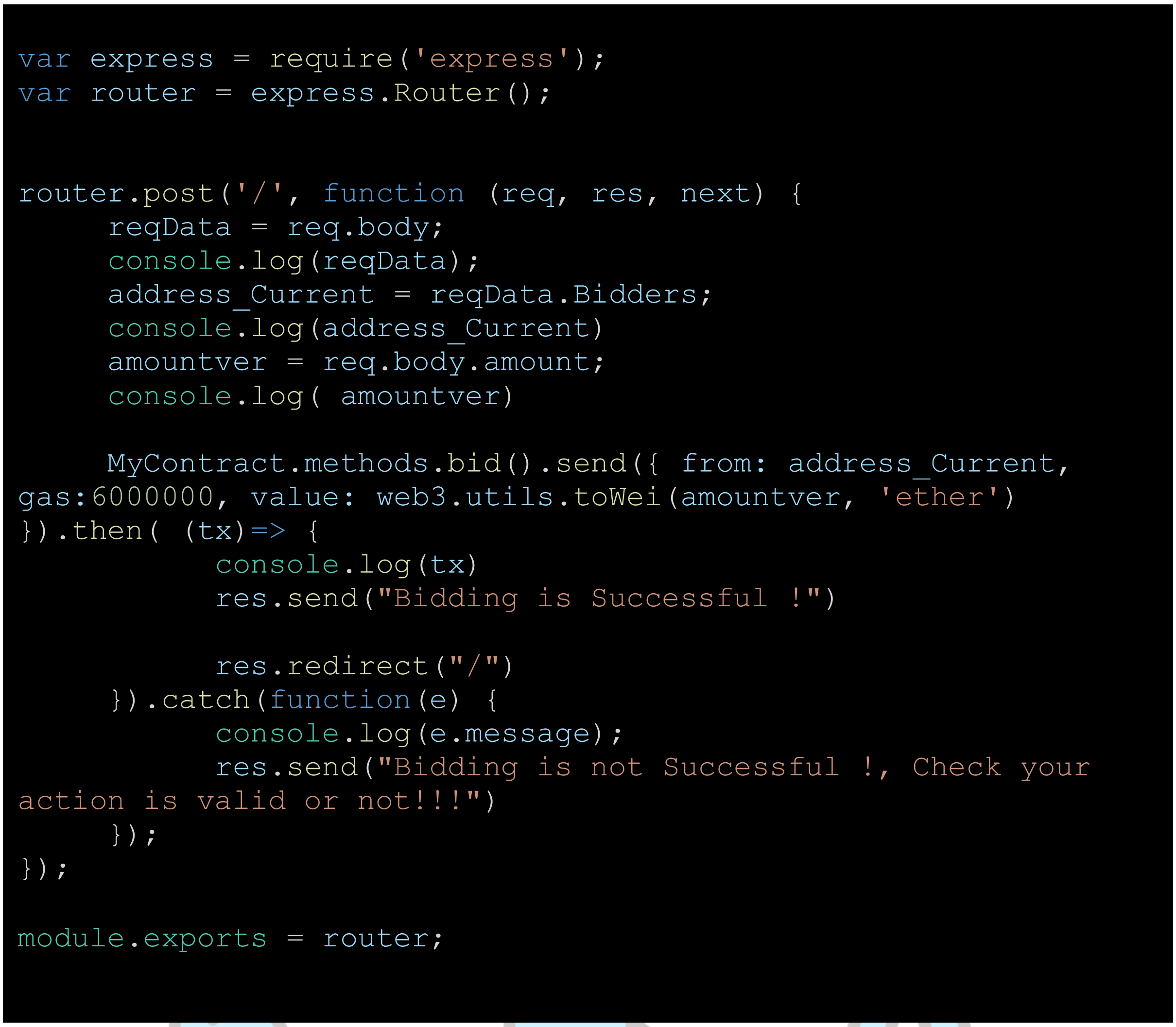
Auction Dapp Part3

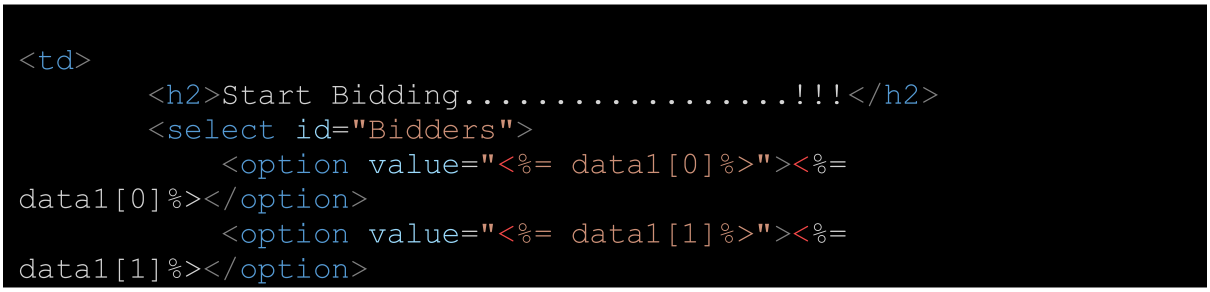
We completed our get auction details functionality, next we need to complete other functionalities for bidding, get bid value, cancel an auction, and withdraw for enable bidders to get their ether back.

- Next we need to create routes for implement bidding functionality, **bidFor.js.**



```
var express = require('express');
var router = express.Router();


router.post('/', function (req, res, next) {
     reqData = req.body;
     console.log(reqData);
     address_Current = reqData.Bidders;
     console.log(address_Current)
     amountver = req.body.amount;
     console.log( amountver)

     MyContract.methods.bid().send({ from: address_Current,
gas:6000000, value: web3.utils.toWei(amountver, 'ether')
}).then( (tx)=> {
           console.log(tx)
           res.send("Bidding is Successful !")

           res.redirect("/")
     }).catch(function(e) {
           console.log(e.message);
           res.send("Bidding is not Successful !, Check your
action is valid or not!!!")
     });
});

module.exports = router;
```

- Include corresponding view element on **auction.ejs** (add this part after the div closing of auction details)



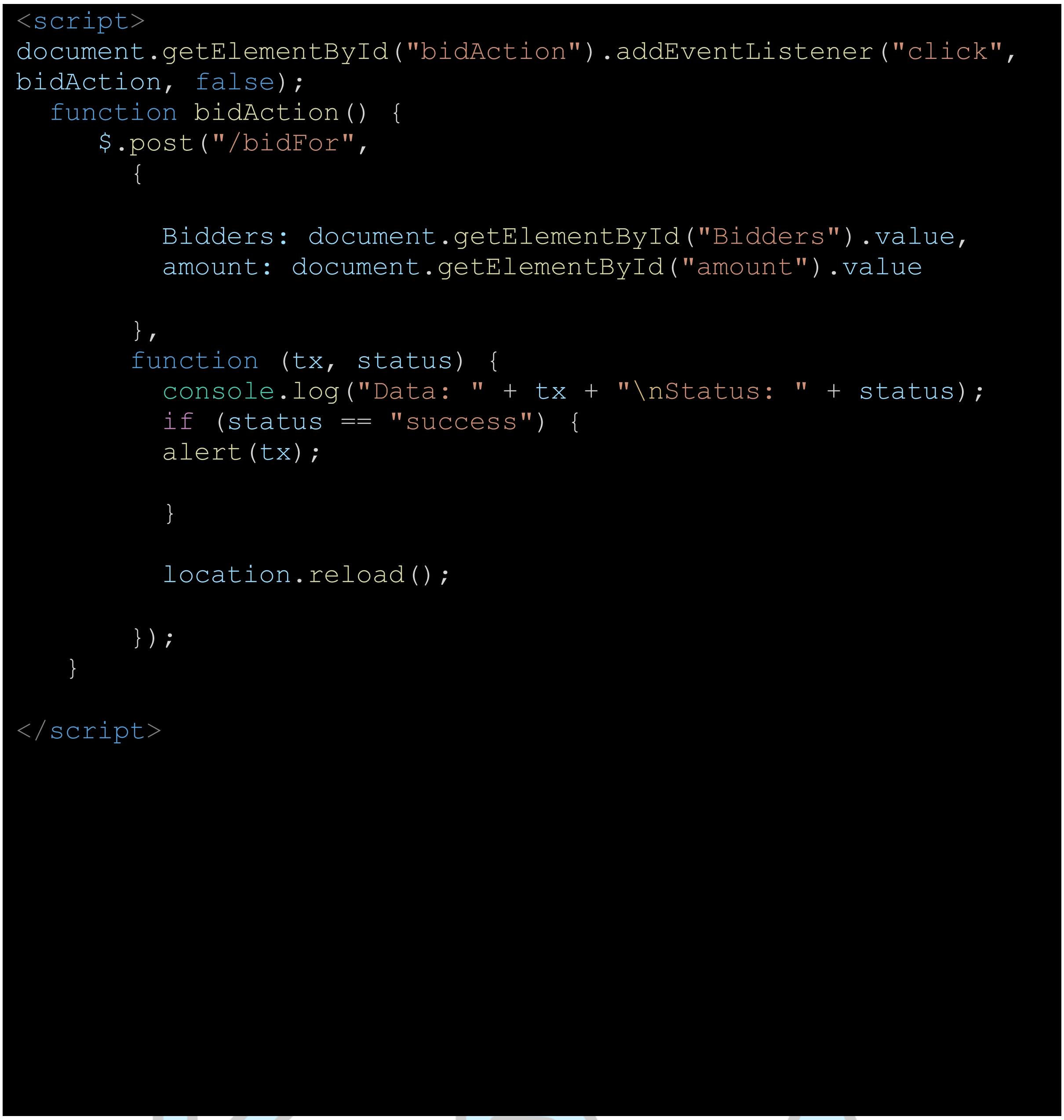
```
<td>
       <h2>Start Bidding..................!!!</h2>
       <select id="Bidders">
           <option value="<%= data1[0]%>"><%=
data1[0]%></option>
           <option value="<%= data1[1]%>"><%=
data1[1]%></option>
```

```
           <option value="<%= data1[2]%>"><%=
data1[2]%></option>
           <option value="<%= data1[3]%>"><%=
data1[3]%></option>
           <option value="<%= data1[4]%>"><%=
data1[4]%></option>
         </select>
         <input id="amount" name="amount" type="text"
placeholder="Amount (wei)">
         <input type="submit" id="bidAction" value="Bid">

         <br>
         <br>
```

- Include following script references

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.
min.js"></script>
```

- Include script for trigger bid function from front end



```
<script>
document.getElementById("bidAction").addEventListener("click",
bidAction, false);
  function bidAction() {
     $.post("/bidFor",
       {

         Bidders: document.getElementById("Bidders").value,
         amount: document.getElementById("amount").value

       },
       function (tx, status) {
         console.log("Data: " + tx + "\nStatus: " + status);
         if (status == "success") {
         alert(tx);

         }

         location.reload();

       });
   }

</script>
```

- Next we need to create routes for implement get bid value functionality, **getBid.js.**

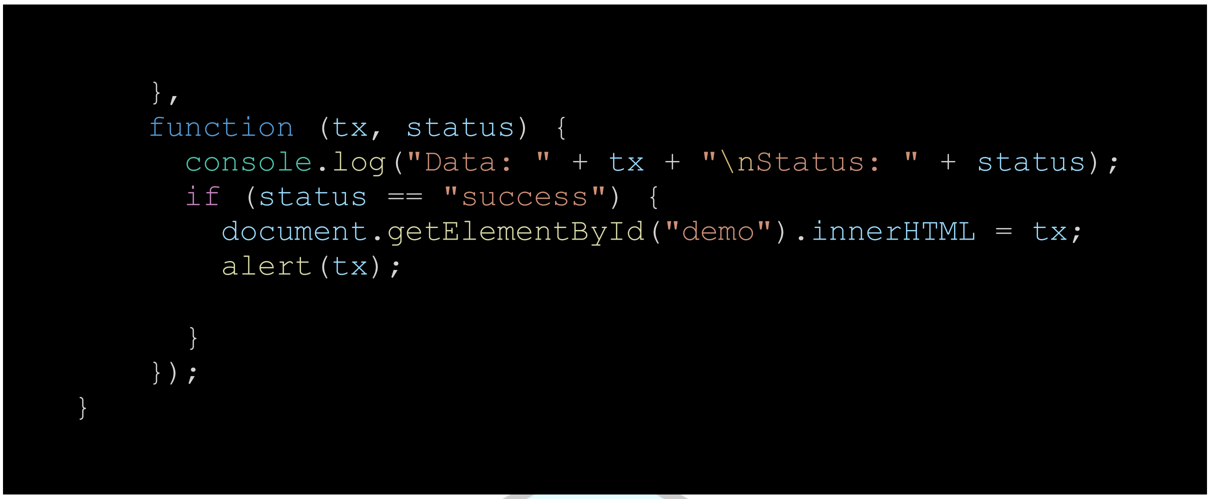
```
var express = require('express');
var router = express.Router();

router.post('/', function (req, res, next) {
 reqData = req.body;
 console.log(reqData);
 addressver = req.body.address;
 MyContract.methods.bids(addressver).call({ from: addressver
}).then(function (data) {
   console.log("Bid amount is: ", data);
   res.send(data);
 })
});

module.exports = router;
```

- Include corresponding view element on **auction.ejs** (add this part after the br element)

```
     <p class="stylename">Check your Bid value here:</p>
     <input id="address" name="address" type="text"
placeholder="Address">
     <input type="submit" id="bidValue" value="Bid Value" />
     <br>
     <br>
```

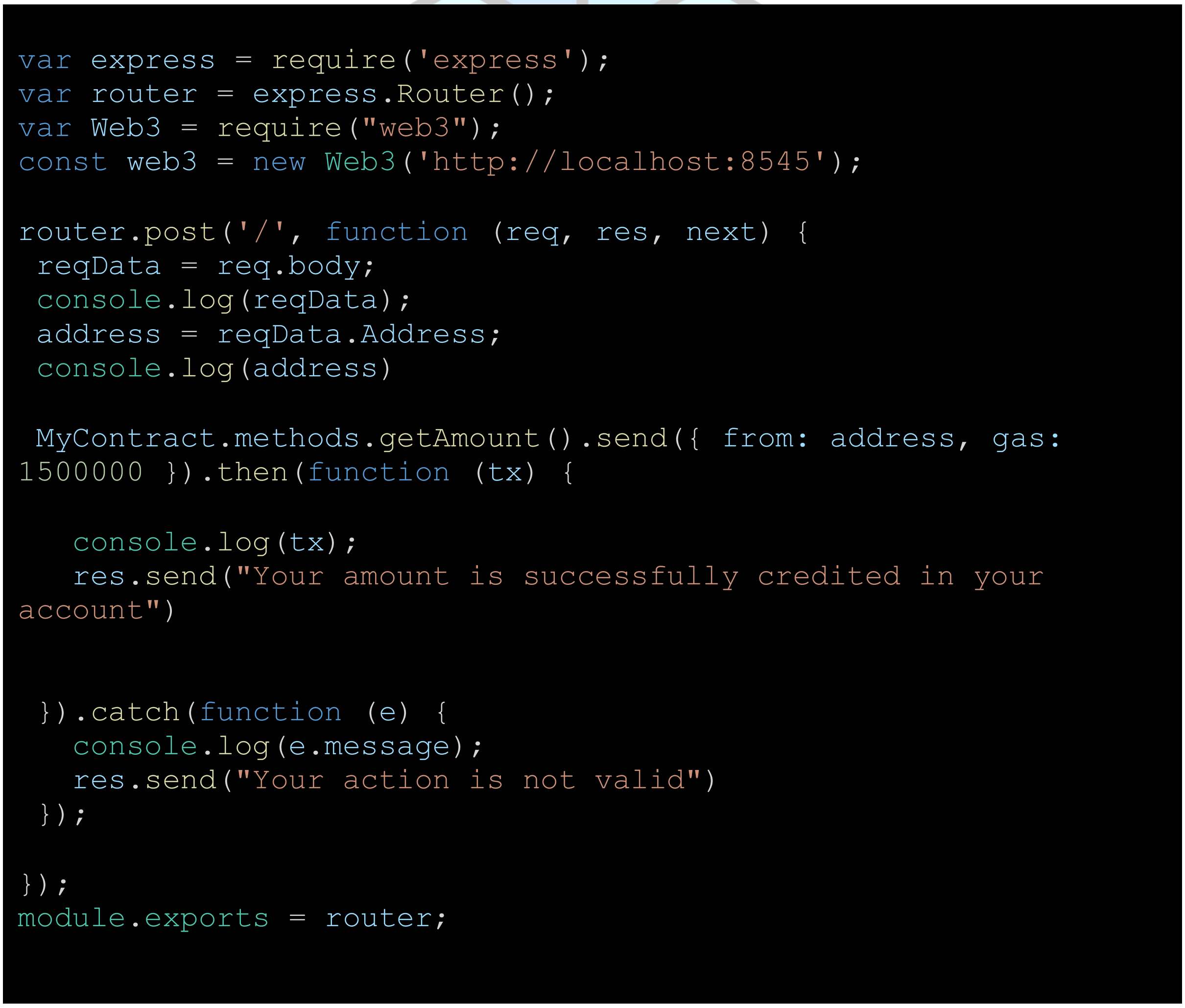
- Include script for trigger get bid function from front end

```
document.getElementById("bidValue").addEventListener("click",
bidValue, false);


  function bidValue() {
     $.post("/getBid",
       {

         address: document.getElementById("address").value
```



```
       },
       function (tx, status) {
         console.log("Data: " + tx + "\nStatus: " + status);
         if (status == "success") {
           document.getElementById("demo").innerHTML = tx;
           alert(tx);

         }
       });
   }
```

- Next we need to create routes for implement bidding functionality, **getBidBack.js.**



```
var express = require('express');
var router = express.Router();
var Web3 = require("web3");
const web3 = new Web3('http://localhost:8545');

router.post('/', function (req, res, next) {
 reqData = req.body;
 console.log(reqData);
 address = reqData.Address;
 console.log(address)

 MyContract.methods.getAmount().send({ from: address, gas:
1500000 }).then(function (tx) {

   console.log(tx);
   res.send("Your amount is successfully credited in your
account")


 }).catch(function (e) {
   console.log(e.message);
   res.send("Your action is not valid")
 });

});
module.exports = router;
```

- Include corresponding view element on **auction.ejs** (add this part after the br element)

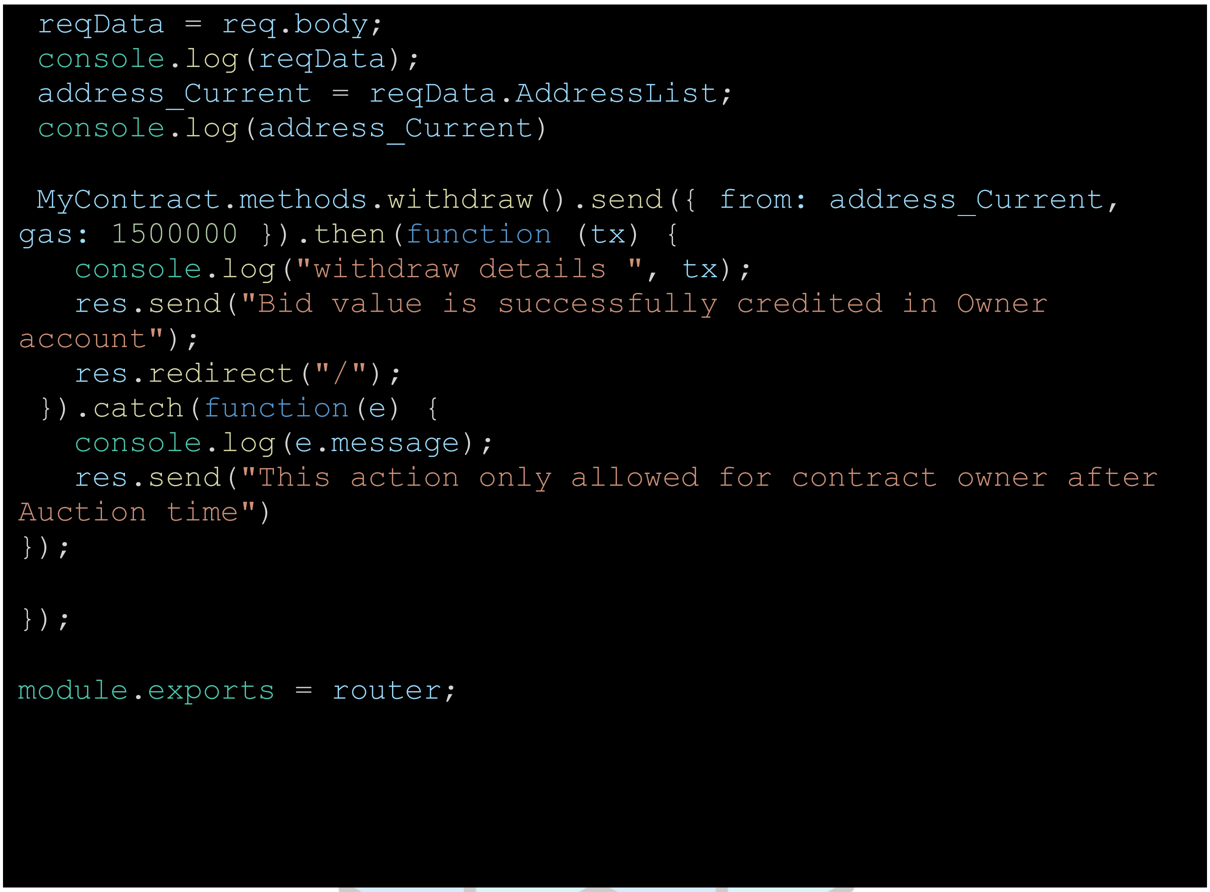
```
<p class="stylename">Get Your bids (For loosers!!!):</p>
     <select id="AddressList1">
       <option value="<%= data1[0]%>"><%= data1[0]%></option>
       <option value="<%= data1[1]%>"><%= data1[1]%></option>
       <option value="<%= data1[2]%>"><%= data1[2]%></option>
       <option value="<%= data1[3]%>"><%= data1[3]%></option>
       <option value="<%= data1[4]%>"><%= data1[4]%></option>
     </select>
     <input type="submit" id="getBidBack" value="Get Your Bids
" />

     <br>
     <br>
```

- Include script for trigger getBidBack function from front end

```
document.getElementById("getBidBack").addEventListener("click",
getBidBack, false);

function getBidBack() {
     $.post("/getBidBack",
       {
         Address: document.getElementById("AddressList1").value

       },
       function (tx, status) {
         console.log("Data: " + tx + "\nStatus: " + status);
         if (status == "success") {
           alert(tx);
         }
       });
   }
```

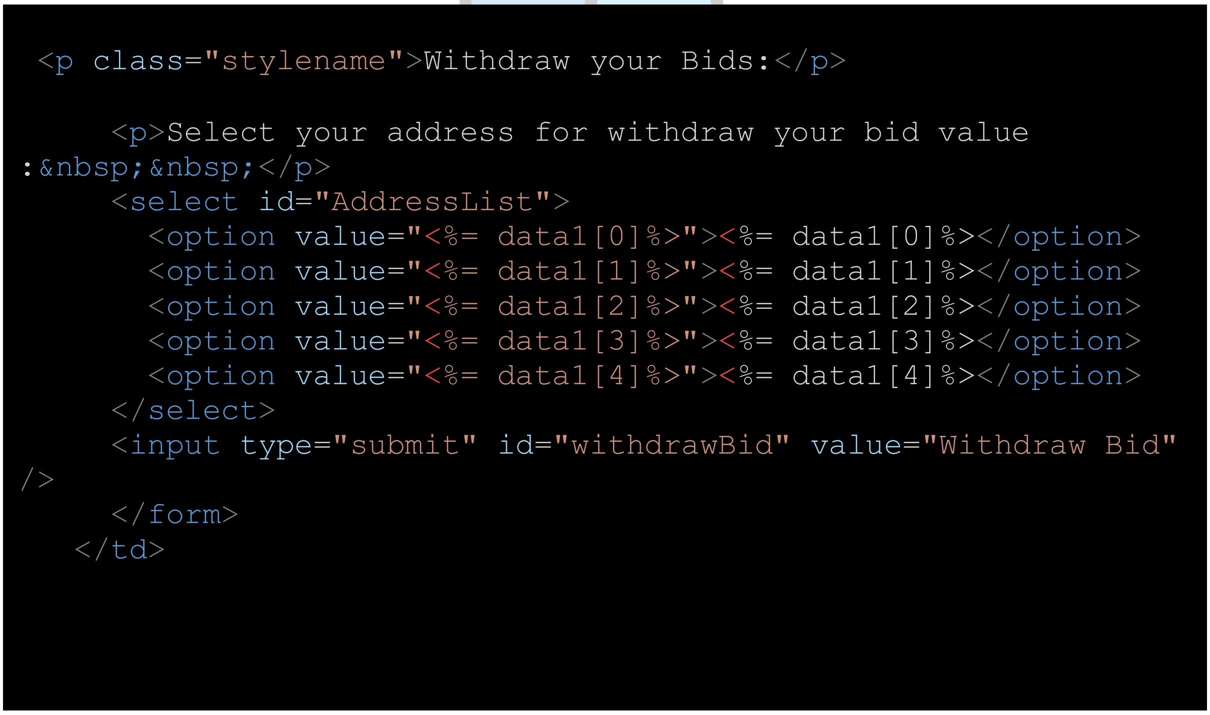
- Next we need to create routes for implement withdraw functionality **withdrawBid.js.**

```
var express = require('express');
var router = express.Router();

router.post('/', function (req, res, next) {
```



```
 reqData = req.body;
 console.log(reqData);
 address_Current = reqData.AddressList;
 console.log(address_Current)

 MyContract.methods.withdraw().send({ from: address_Current,
gas: 1500000 }).then(function (tx) {
   console.log("withdraw details ", tx);
   res.send("Bid value is successfully credited in Owner
account");
   res.redirect("/");
 }).catch(function(e) {
   console.log(e.message);
   res.send("This action only allowed for contract owner after
Auction time")
});

});

module.exports = router;
```

- Include corresponding view element on **auction.ejs** (add this part after the br element)



```
 <p class="stylename">Withdraw your Bids:</p>

     <p>Select your address for withdraw your bid value
:&nbsp;&nbsp;</p>
     <select id="AddressList">
       <option value="<%= data1[0]%>"><%= data1[0]%></option>
       <option value="<%= data1[1]%>"><%= data1[1]%></option>
       <option value="<%= data1[2]%>"><%= data1[2]%></option>
       <option value="<%= data1[3]%>"><%= data1[3]%></option>
       <option value="<%= data1[4]%>"><%= data1[4]%></option>
     </select>
     <input type="submit" id="withdrawBid" value="Withdraw Bid"
/>
     </form>
   </td>
```

- Include script for trigger withdraw function from front end

```
document.getElementById("withdrawBid").addEventListener("click"
, withdrawBid, false);

function withdrawBid() {
     $.post("/withdrawBid",
       {

         AddressList:
document.getElementById("AddressList").value


       },
       function (tx, status) {
         console.log("Data: " + tx + "\nStatus: " + status);
         if (status == "success") {
           alert(tx);
 }

       });
   }
```

- Next step is to add the routes to the middleware chain. We do this in app.js.

```
     var bidForRouter = require('./routes/bidFor');
     var getBidRouter = require('./routes/getBid');
     var getBidBackRouter = require('./routes/getBidBack');
     var withdrawouter = require('./routes/withdrawBid');
```

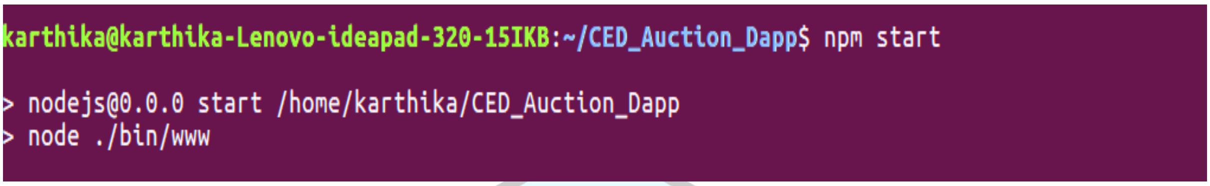
- Next we need to add our (previously imported) route-handling code to the request handling chain

```
app.use('/bidFor', bidForRouter);
app.use('/getBid', getBidRouter);
app.use('/getBidBack', getBidBackRouter);
app.use('/withdrawBid', withdrawouter);
```

for running our web application use following command

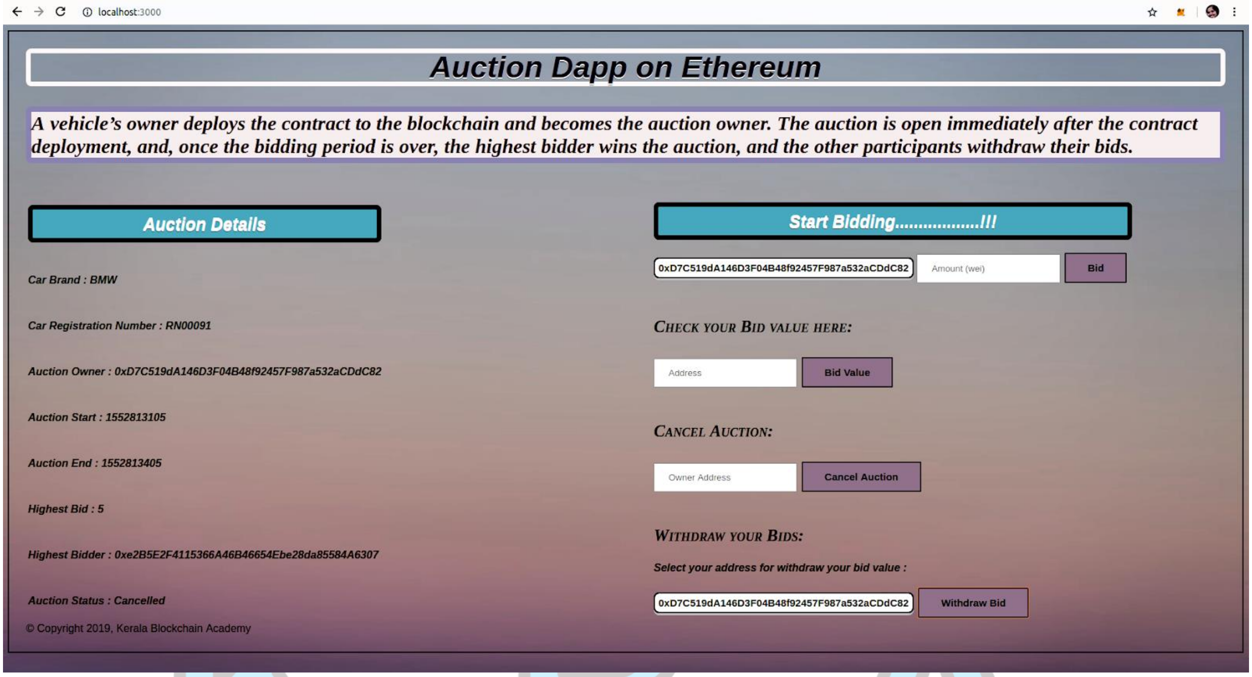
```
$ npm start
```

Output



```
karthika@karthika-Lenovo-ideapad-320-15IKB:~/CED_Auction_Dapp$ npm start

> nodejs@0.0.0 start /home/karthika/CED_Auction_Dapp
> node ./bin/www
```

Navigate to http://localhost:3000/



Auction Dapp on Ethereum

A vehicle’s owner deploys the contract to the blockchain and becomes the auction owner. The auction is open immediately after the contract deployment, and, once the bidding period is over, the highest bidder wins the auction, and the other participants withdraw their bids.

Auction Details

Car Brand : BMW

Car Registration Number : RN00091

Auction Owner : 0xD7C519dA146D3F04B48f92457F987a532aCDdC82

Auction Start : 1552813105

Auction End : 1552813405

Highest Bid : 5

Highest Bidder : 0xe2B5E2F4115366A46B46654Ebe28da85584A6307

Auction Status : Cancelled

Start Bidding..................!!!

0xD7C519dA146D3F04B48f92457F987a532aCDdC82

CHECK YOUR BID VALUE HERE:

CANCEL AUCTION:

WITHDRAW YOUR BIDS:

Select your address for withdraw your bid value :

0xD7C519dA146D3F04B48f92457F987a532aCDdC82