FRONT END

```
<html>

<head>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.m
in.css" integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
        crossorigin="anonymous">

  <!-- <title>APMC</title> -->
  <style>
    body {
    background-color: rgb(25, 53, 73);
    text-align: left;
    color: white;
    font-family: Arial, Helvetica, sans-serif;
    }



.container2{
    position: relative;
    margin-top: 100px;
    margin-left: 180px;

}


.dropbtn {
  background-color: #f35f6b;
  color: rgb(244, 244, 244);
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;
}


/* Dropdown button on hover & focus */
.dropbtn:hover, .dropbtn:focus {
```

```
  background-color: #2980B9;
}

/* The container <div> - needed to position the dropdown content */
.dropdown {
  position: relative;
  display: inline-block;
}

/* Dropdown Content (Hidden by Default) */
.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 150px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

/* Links inside the dropdown */
.butt {
    display: flex;
    gap: 22px;
    flex-direction: row-reverse;
}

.dropdown-content p {
  color: black;
  padding: 10px 5px 2px 5px;
  text-decoration: none;
  display: block;
  cursor: pointer;

}
.button1 {
    width: 162px;
    height: 56px;
    text-align: center;
    background-color: rgb(104, 202, 13);
}

.container1{
```

```
    margin-top: 10px;
    align-content: center;
}
.label {
    display: inline-block;
    width: 140px;
}

/* Change color of dropdown links on hover */
.dropdown-content a:hover {background-color: #ddd;}

/* Show the dropdown menu (use JS to add this class to the .dropdown-
content container when the user clicks on the dropdown button) */
.show {display:block;}

</style>
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"
integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8="
    crossorigin="anonymous"></script>
    <script charset="utf-8" src="https://cdn.ethers.io/scripts/ethers-
v4.min.js" type="text/javascript"></script>

</head>

<body>

  <script>

    /* When the user clicks on the button,
toggle between hiding and showing the dropdown content */
function myFunction() {
  document.getElementById("myDropdown").classList.toggle("show");
}

// Close the dropdown menu if the user clicks outside of it
window.onclick = function(event) {
  if (!event.target.matches('.dropbtn')) {
    var dropdowns = document.getElementsByClassName("dropdown-content");
    var i;
    for (i = 0; i < dropdowns.length; i++) {
      var openDropdown = dropdowns[i];
      if (openDropdown.classList.contains('show')) {
```

```
        openDropdown.classList.remove('show');
      }
    }
  }
}
  </script>
<div class = "container">

    <br></br>
      <div class="container1">
        <div class="form-group">

            <!-- uint journey_dist;
            uint rem_battery;
            uint battery_capacity;
            uint free_charging; -->

          <div class="row">
            <div class="col offset-md-4 col-md-4">
                <h1>Charging Stations</h1>
                <br></br>
              <label for="x_c" class="label">X-
coordinate:      </label>
              <input type="text" id="x"><br></br>
              <label for="y_c">Y-
coordinate:           <
/label>
              <input type="text"  id="y"><br></br>
              <!-- <label for="distance">Journey
distance:   </label>
              <input type="text"  id="d"><br></br> -->
              <label for="battery_c">Battery
capacity:     </label>
              <input type="text" id="bc"><br></br>
              <label for="battery_r">Remaining battery: </label>
              <input type="text" id="rb"><br></br>
              <!-- Enter 1 for free charging else 0 -->
              <label for="fast_c">Fast
charging:        </label>
              <input type="text" id="fc">
              <br></br>
              <div class="butt">
```

```
            <button class="btn dropbtn button1"
onclick="our_optimal()"> Optimal Solution </button>
            <br>
            <div class="form-group">
              <div class="dropdown" float="left">
                  <button onclick="myFunction()" class="btn
dropbtn">User Preference</button>
                  <div id="myDropdown" class="dropdown-content">
                    <p id="d" onclick="user_optimal(d)">Distance</p>
                    <p id="c" onclick="user_optimal(c)">Cost</p>
                  </div>
                </div>
          </div>
        </div>
        </div>
      </div>

</div>


  <script>
    window.ethereum.enable()
    var provider = new
ethers.providers.Web3Provider(web3.currentProvider, 'ropsten');

    //change this address to that of bank contract
    var bankContractAddress =
"0x2eBFc3EF701B2DA1C29618B9b0c470131E78eF40";
    let bankContractABI = [
    {
        "inputs": [],
        "name": "cal_distance",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "coordinates",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
```

```
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "x1",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "y1",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "bc1",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "rb1",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "fc1",
                "type": "uint256"
            }
        ],
        "name": "final_fun",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "init",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
```

```
        "inputs": [],
        "name": "init_stat",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "init1",
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "x_coo",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "y_coo",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "bc",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "rb",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "fc",
                "type": "uint256"
            }
        ],
        "name": "initial",
```

```
        "outputs": [],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "sortByCost",
        "outputs": [
            {
                "components": [
                    {
                        "internalType": "string",
                        "name": "c_name",
                        "type": "string"
                    },
                    {
                        "internalType": "uint256",
                        "name": "cost_c",
                        "type": "uint256"
                    }
                ],
                "internalType": "struct charging_stations.cost_arr[]",
                "name": "",
                "type": "tuple[]"
            }
        ],
        "stateMutability": "nonpayable",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "sortByDist",
        "outputs": [
            {
                "components": [
                    {
                        "internalType": "string",
                        "name": "d_name",
                        "type": "string"
                    },
                    {
                        "internalType": "uint256",
```

```
                    "name": "dist_d",
                    "type": "uint256"
                }
            ],
            "internalType": "struct charging_stations.dist_arr[]",
            "name": "",
            "type": "tuple[]"
        }
    ],
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "inputs": [],
    "name": "sortOptimal",
    "outputs": [
        {
            "components": [
                {
                    "internalType": "string",
                    "name": "o_name",
                    "type": "string"
                },
                {
                    "internalType": "uint256",
                    "name": "op_cost",
                    "type": "uint256"
                }
            ],
            "internalType": "struct charging_stations.optimal[]",
            "name": "",
            "type": "tuple[]"
        }
    ],
    "stateMutability": "nonpayable",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
```

```
                "type": "uint256"
            }
        ],
        "name": "arr",
        "outputs": [
            {
                "internalType": "string",
                "name": "name",
                "type": "string"
            },
            {
                "internalType": "uint256",
                "name": "x",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "y",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "d",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "cost",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "fast_charging",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
```

```
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "cost_array",
    "outputs": [
        {
            "internalType": "string",
            "name": "c_name",
            "type": "string"
        },
        {
            "internalType": "uint256",
            "name": "cost_c",
            "type": "uint256"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "f",
    "outputs": [
        {
            "internalType": "string",
            "name": "cs",
            "type": "string"
        },
        {
            "internalType": "uint256",
            "name": "final_cost",
            "type": "uint256"
        }
    ],
```

```
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "getstations",
        "outputs": [
            {
                "components": [
                    {
                        "internalType": "string",
                        "name": "name",
                        "type": "string"
                    },
                    {
                        "internalType": "uint256",
                        "name": "x",
                        "type": "uint256"
                    },
                    {
                        "internalType": "uint256",
                        "name": "y",
                        "type": "uint256"
                    },
                    {
                        "internalType": "uint256",
                        "name": "d",
                        "type": "uint256"
                    },
                    {
                        "internalType": "uint256",
                        "name": "cost",
                        "type": "uint256"
                    },
                    {
                        "internalType": "uint256",
                        "name": "fast_charging",
                        "type": "uint256"
                    }
                ],
                "internalType": "struct charging_stations.stations[]",
                "name": "",
```

```
            "type": "tuple[]"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [],
    "name": "getstations1",
    "outputs": [
        {
            "components": [
                {
                    "internalType": "string",
                    "name": "o_name",
                    "type": "string"
                },
                {
                    "internalType": "uint256",
                    "name": "op_cost",
                    "type": "uint256"
                }
            ],
            "internalType": "struct charging_stations.optimal[]",
            "name": "",
            "type": "tuple[]"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "optimal_array",
    "outputs": [
        {
```

```
                "internalType": "string",
                "name": "o_name",
                "type": "string"
            },
            {
                "internalType": "uint256",
                "name": "op_cost",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [],
        "name": "pp",
        "outputs": [
            {
                "internalType": "uint256",
                "name": "x",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "y",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "battery_capacity",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "rem_battery",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "fast_charge",
                "type": "uint256"
            }
```

```
        ],
        "stateMutability": "view",
        "type": "function"
    },
    {
        "inputs": [
            {
                "internalType": "uint256",
                "name": "",
                "type": "uint256"
            }
        ],
        "name": "preference",
        "outputs": [
            {
                "internalType": "string",
                "name": "name",
                "type": "string"
            },
            {
                "internalType": "uint256",
                "name": "x",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "y",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "d",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "cost",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "fast_charging",
```

```
            "type": "uint256"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "preference1",
    "outputs": [
        {
            "internalType": "string",
            "name": "o_name",
            "type": "string"
        },
        {
            "internalType": "uint256",
            "name": "op_cost",
            "type": "uint256"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "r_dist",
    "outputs": [
        {
            "internalType": "string",
```

```
            "name": "d_name",
            "type": "string"
        },
        {
            "internalType": "uint256",
            "name": "dist_d",
            "type": "uint256"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "tc",
    "outputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "stateMutability": "view",
    "type": "function"
},
{
    "inputs": [
        {
            "internalType": "uint256",
            "name": "",
            "type": "uint256"
        }
    ],
    "name": "v",
    "outputs": [
        {
```

```
                "internalType": "uint256",
                "name": "cc_v",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "ct_v",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "cw_v",
                "type": "uint256"
            },
            {
                "internalType": "uint256",
                "name": "cf_v",
                "type": "uint256"
            }
        ],
        "stateMutability": "view",
        "type": "function"
    }
]

    provider.listAccounts().then(function (accounts) {
        console.log("Charging Station running...")
      signer = provider.getSigner(accounts[0]);
      bankContract = new ethers.Contract(bankContractAddress,
bankContractABI, signer);
    })


    async function user_optimal(e) {
        x = $("#x").val();
        y = $("#y").val();
        bc = $("#bc").val();
        rb = $("#rb").val();
        fc = $("#fc").val();
        var t = [];
        var z = await bankContract.final_fun(x, y, bc, rb, fc);
```

```
        if(e.id == "d"){

            var p = await bankContract.r_dist(0);
            var q = await bankContract.r_dist(1);
            var r = await bankContract.r_dist(2);
            var s = await bankContract.r_dist(3);
            //alert(`Station     Distance\n${p.d_name}    ${p.dist_d}\n
    ${q.d_name}     ${q.dist_d}\n       ${r.d_name}     ${r.dist_d}\n
    ${s.d_name}     ${s.dist_d}`)
            setTimeout(function() {
                alert(`Stations ranking based on
distance.\nStation    -    Distance\n${p.d_name}         -    ${p.dist_d}\n$
{q.d_name}         -    ${q.dist_d}\n${r.d_name}         -    ${r.dist_d}\n$
{s.d_name}         -    ${s.dist_d}`)
            }, 35000);


            // alert(t);
        }else if(e.id = "c"){
            var p = await bankContract.cost_array(0);
            var q = await bankContract.cost_array(1);
            var r = await bankContract.cost_array(2);
            var s = await bankContract.cost_array(3);
            //alert(`Station     Cost\n${p.c_name}    ${p.cost_c}\n
${q.c_name}     ${q.cost_c}\n       ${r.c_name}     ${r.cost_c}\n
${s.c_name}     ${s.cost_c}`)
            //await
alert(`Station    -    Cost\n${p.c_name}         -    ${p.cost_c}\n${q.c_nam
e}         -    ${q.cost_c}\n${r.c_name}         -    ${r.cost_c}\n${s.c_nam
e}         -    ${s.cost_c}`)
            setTimeout(function() {
                alert(`Stations ranking based on
cost.\nStation    -    Cost\n${p.c_name}         -    ${p.cost_c}\n${q.c_nam
e}         -    ${q.cost_c}\n${r.c_name}         -    ${r.cost_c}\n${s.c_nam
e}         -    ${s.cost_c}`)
            }, 35000);
        }
    }

    async function our_optimal(){
        x = $("#x").val();
        y = $("#y").val();
```

```
        bc = $("#bc").val();
        rb = $("#rb").val();
        fc = $("#fc").val();
        var z = await bankContract.final_fun(x, y, bc, rb, fc);
        var p = await bankContract.f(0);
        var q = await bankContract.f(1);
        var r = await bankContract.f(2);
        var s = await bankContract.f(3);
        setTimeout(function(){
            alert(`Stations ranking based on distance, cost and
time.\nStation   -   Optimal
Cost\n${p.cs}        -   ${p.final_cost}\n${q.cs}        -   ${q.final_c
ost}\n${r.cs}        -   ${r.final_cost}\n${s.cs}        -   ${s.final_c
ost}`)
        }, 35000);

        //await alert(`Station   -   Optimal
Cost\n${p.cs}        -   ${p.final_cost}\n${q.cs}        -   ${q.final_c
ost}\n${r.cs}        -   ${r.final_cost}\n${s.cs}        -   ${s.final_c
ost}`)
    }



  </script>
</body>

</html>
```