

CCT College Dublin

Assessment Cover Page

Module Title:	Cross Platform Development
Assessment Title:	Theory and practical
Lecturer Name:	Dr. Muhammad Iqbal
Student Full Name:	Wallace Esteves de Carvalho Caio Machado de Oliveira
Student Number:	2020326 2020351
Assessment Due Date:	20/05/2022
Date of Submission:	20/05/2022

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

Design cross-platform	2
Code.....	2
Index.html.....	3
Script.js.....	3
Style.css	5
Package.json	5
Unit testing using Jest.....	6
Participation.....	6
Video Link.....	6
GitHub Link.....	6

Design cross-platform

We were asked to develop an electricity bill calculator that will give us the output with VAT included and without VAT. In order to make the design for this assignment we have decided to use one of the projects that was given in class by our lecture Dr. Muhammad Iqbal. In order to start the application, you have to do a few steps for the application to work properly.

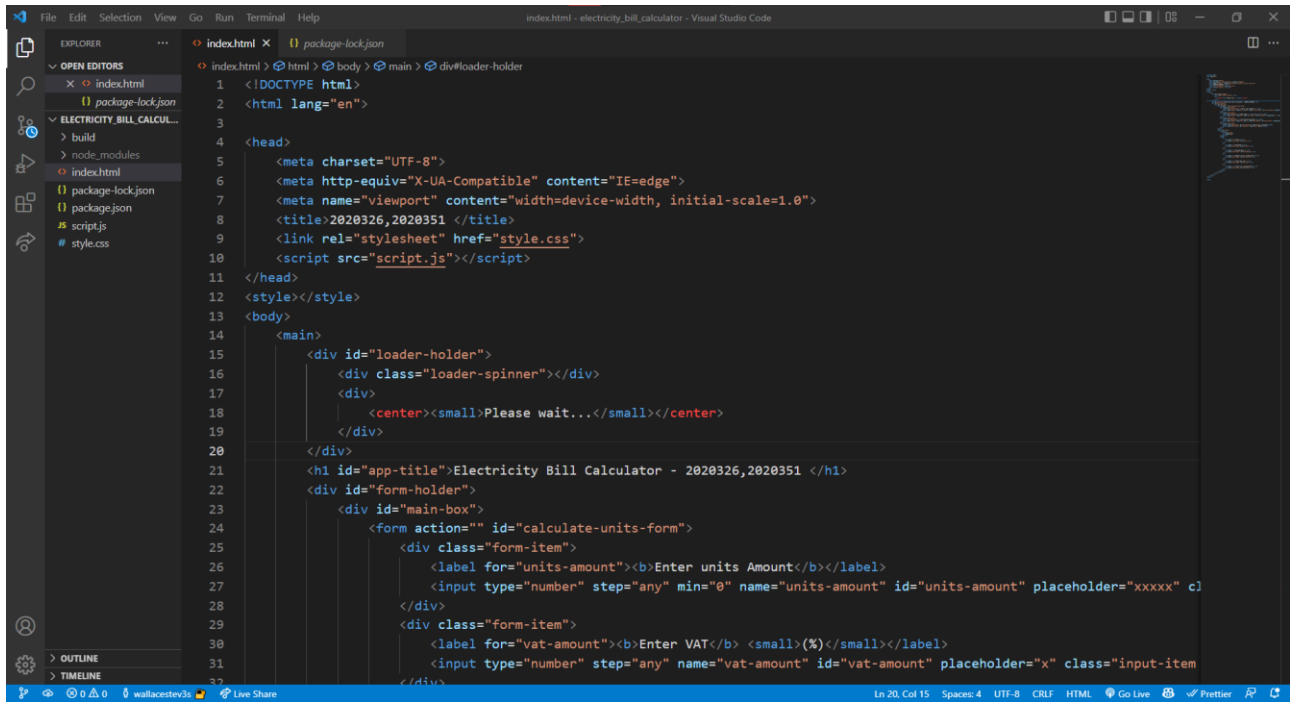
Desktop application using CMD windows command prompt (Run as an administrator) or git bash shell.

1. `npm install`
2. `npm install -g nw`
3. `npm install async@2.1.4 --save`
4. `npm install nwbuild@0.0.5 --save`
5. `npm install osenv@0.1.4 --save`
6. `nw` – to start the application

Code

Coding explanation will be available on [Video](#).

Index.html



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>2020326,2020351 </title>
9   <link rel="stylesheet" href="style.css">
10  <script src="script.js"></script>
11 </head>
12 <style></style>
13 <body>
14   <main>
15     <div id="loader-holder">
16       <div class="loader-spinner"></div>
17       <div>
18         <center><small>Please wait...</small></center>
19       </div>
20     </div>
21     <h1 id="app-title">Electricity Bill Calculator - 2020326,2020351 </h1>
22     <div id="form-holder">
23       <div id="main-box">
24         <form action="" id="calculate-units-form">
25           <div class="form-item">
26             <label for="units-amount"><b>Enter units Amount</b></label>
27             <input type="number" step="any" min="0" name="units-amount" id="units-amount" placeholder="xxxxx" c1
28           </div>
29           <div class="form-item">
30             <label for="vat-amount"><b>Enter VAT</b> <small>(<b>%</b></small></label>
31             <input type="number" step="any" name="vat-amount" id="vat-amount" placeholder="x" class="input-item
32           </div>
33         </form>
34       </div>
35     </div>
36   </main>
37 </body>
38 </html>
```

Script.js

```

1 window.start_loader = function() {
2   const loader = document.getElementById('loader-holder')
3   loader.style.display = 'flex';
4 }
5 window.end_loader = function() {
6   const loader = document.getElementById('loader-holder')
7   loader.style.display = 'none';
8 }
9
10 window.onload = function() {
11   setTimeout(() => {
12     end_loader()
13   }, 500)
14
15   const unitForm = document.getElementById('calculate-units-form')
16   unitForm.addEventListener('submit', function(e) {
17     e.preventDefault()
18     start_loader();
19     const unitsAmount = document.getElementById('units-amount').value;
20     const vat = document.getElementById('vat-amount').value;
21     const PayableDays = document.getElementById('days-amount').value;
22     var monthly = 0,
23         vatCost = 0,
24         total = 0,
25         totalInterest = 0,
26         billWoVat = 0;
27     billWoVat = (parseFloat(unitsAmount) * 0.20 + (parseFloat(PayableDays) * 0.04));
28     vatCost = parseFloat(billWoVat) * (parseFloat(vat) / 100);
29     total = parseFloat(billWoVat) + parseFloat(vatCost);
30     totalInterest = parseFloat(billWoVat) + parseFloat(vatCost);
31     setTimeout(() => {
32       document.getElementById('units').textContent = parseFloat(unitsAmount).toLocaleString("en-US", { style: "decimal"

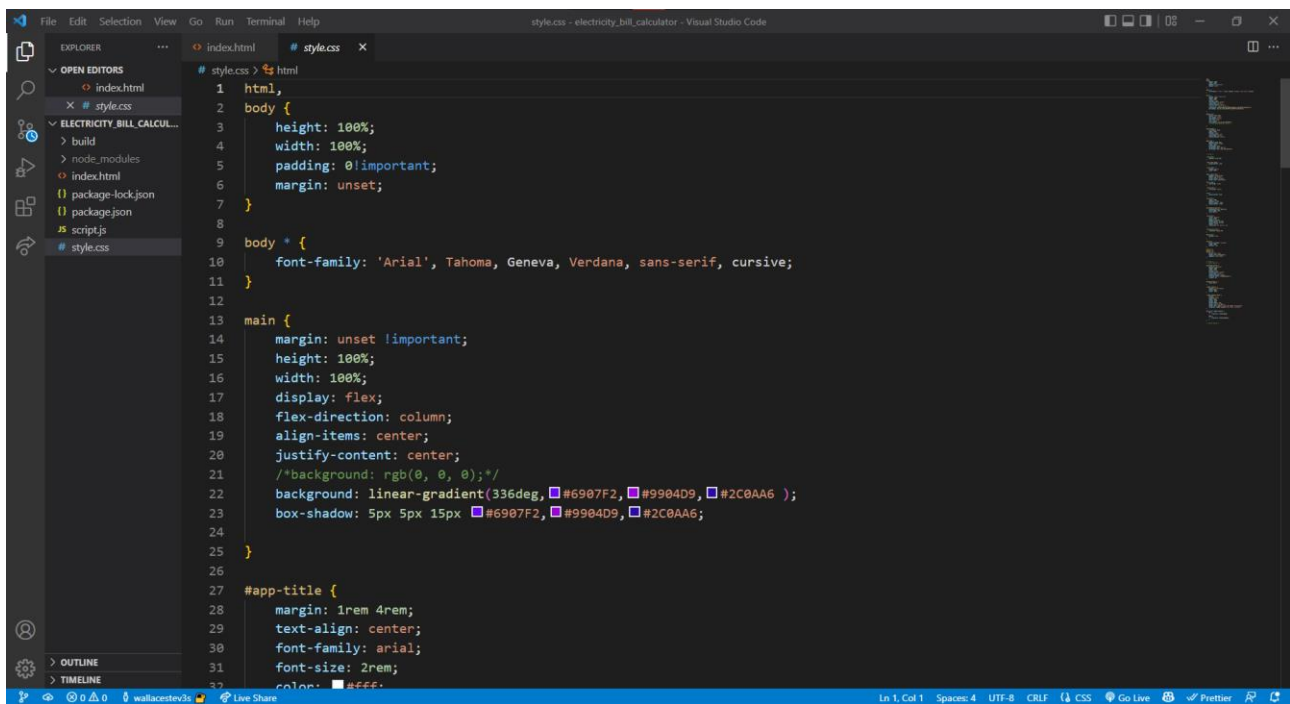
```

```

28   vatCost = parseFloat(billWoVat) * (parseFloat(vat) / 100);
29   total = parseFloat(billWoVat) + parseFloat(vatCost);
30   totalInterest = parseFloat(billWoVat) + parseFloat(vatCost);
31   setTimeout(() => {
32     document.getElementById('units').textContent = parseFloat(unitsAmount).toLocaleString("en-US", { style: "decimal"
33     document.getElementById('vat-rate').textContent = parseFloat(vat).toLocaleString("en-US", { style: "decimal", ma
34     document.getElementById('billing-days').textContent = parseFloat(PayableDays).toLocaleString("en-US", { style: "
35     document.getElementById('bill-wo-vat').textContent = parseFloat(billWoVat).toLocaleString("en-US", { style: "dec
36     document.getElementById('vat-cost').textContent = parseFloat(vatCost).toLocaleString("en-US", { style: "decimal"
37     document.getElementById('total-bill').textContent = parseFloat(totalInterest).toLocaleString("en-US", { style: "
38     document.getElementById('result').style.display = 'table';
39     document.getElementById('reset-btn').style.display = 'block';
40     end_loader()
41   }, 500)
42 })
43
44 unitForm.addEventListener('reset', function(e) {
45   start_loader();
46   setTimeout(() => {
47     document.getElementById('units').textContent = ""
48     document.getElementById('vat-rate').textContent = ""
49     document.getElementById('billing-days').textContent = ""
50     document.getElementById('bill-wo-vat').textContent = ""
51     document.getElementById('vat-cost').textContent = ""
52     document.getElementById('total-bill').textContent = ""
53     document.getElementById('result').style.display = 'none';
54     document.getElementById('reset-btn').style.display = 'none';
55     end_loader()
56   }, 500)
57 })
58 }
59

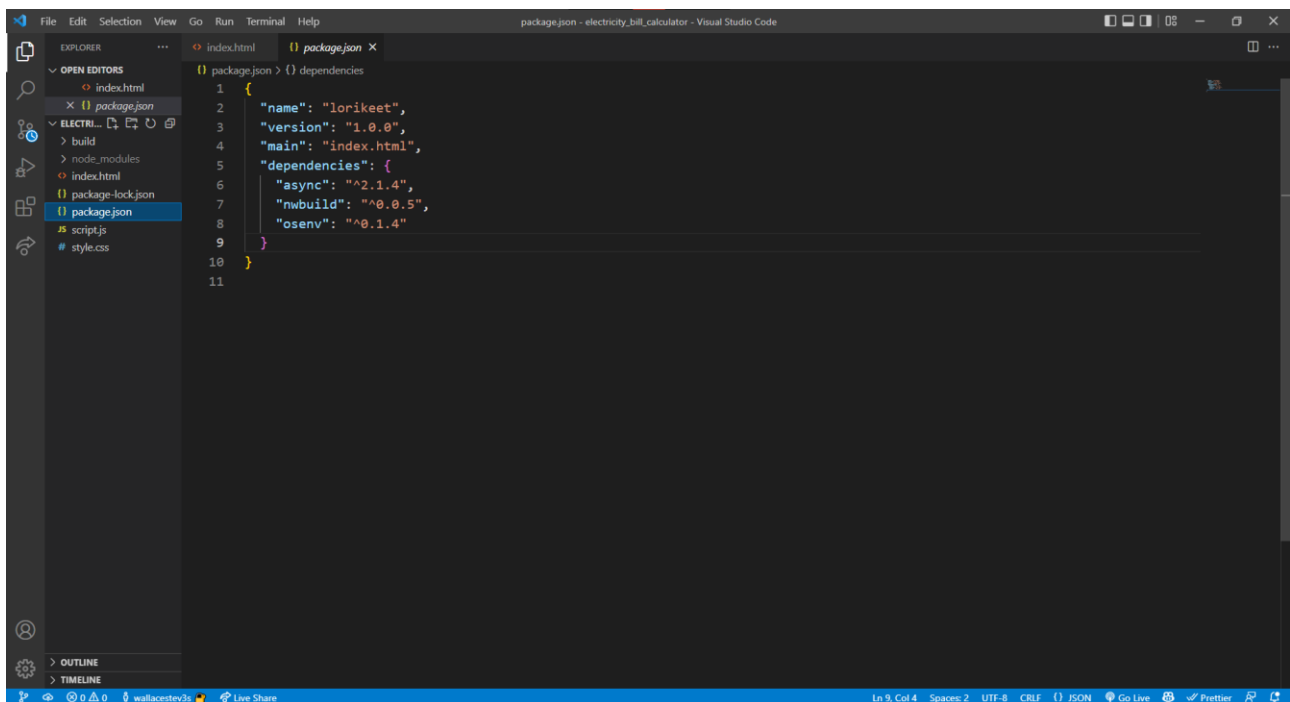
```

Style.css



```
1 html,
2 body {
3   height: 100%;
4   width: 100%;
5   padding: 0 !important;
6   margin: unset;
7 }
8
9 body * {
10  font-family: 'Arial', Tahoma, Geneva, Verdana, sans-serif, cursive;
11 }
12
13 main {
14   margin: unset !important;
15   height: 100%;
16   width: 100%;
17   display: flex;
18   flex-direction: column;
19   align-items: center;
20   justify-content: center;
21   /*background: rgb(0, 0, 0);*/
22   background: linear-gradient(336deg, #6907F2, #9904D9, #2C0AA6 );
23   box-shadow: 5px 5px 15px #6907F2, #9904D9, #2C0AA6;
24 }
25
26
27 #app-title {
28   margin: 1rem 4rem;
29   text-align: center;
30   font-family: arial;
31   font-size: 2rem;
32   color: #fff;
```

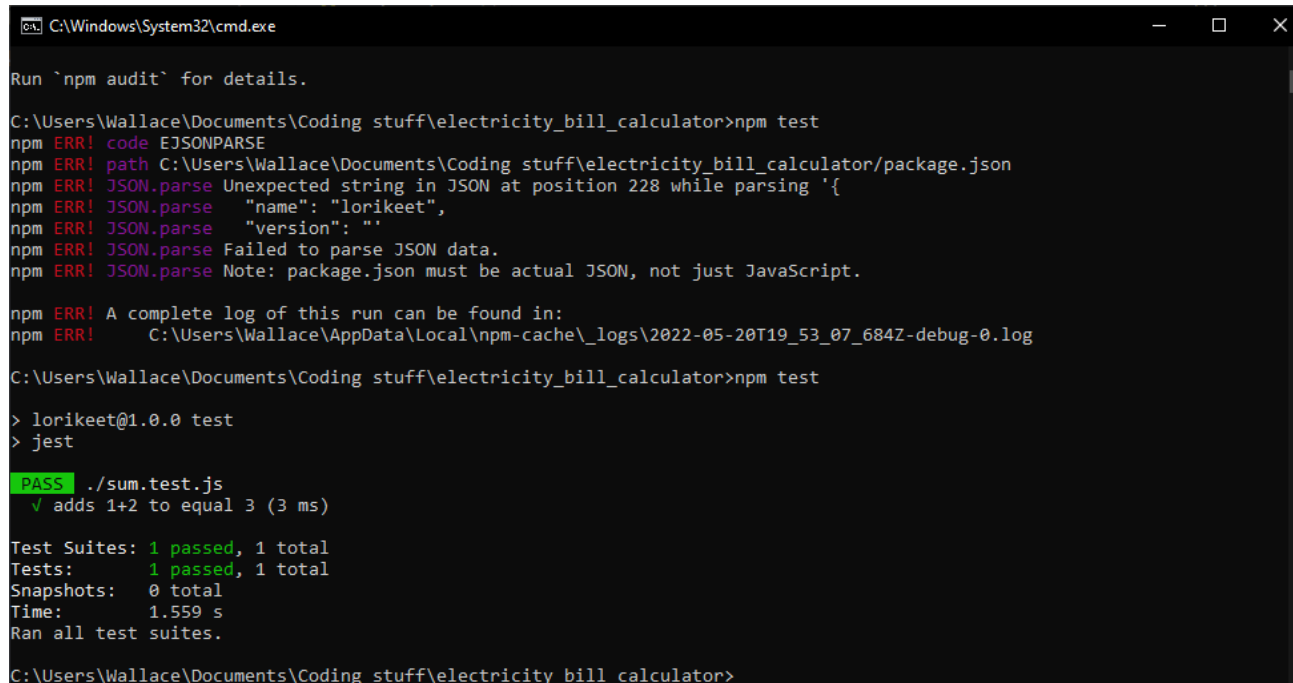
Package.json



```
1 {
2   "name": "lorikeet",
3   "version": "1.0.0",
4   "main": "index.html",
5   "dependencies": {
6     "async": "^2.1.4",
7     "nwbuild": "^0.0.5",
8     "osenv": "^0.1.4"
9   }
10 }
11
```

Unit testing using Jest

We were able to find an application to do the test called “Jest”, unfortunately we couldn’t do a proper test with our calculator because we didn’t have the knowledge to use our coding into the test application. But we did a simple test using two files on it “sum.js” and “sum.test.js”. We wish we were able to complete this test because it’s important to ensure that our code is working properly and won’t have any problems.



```
C:\Windows\System32\cmd.exe

Run `npm audit` for details.

C:\Users\Wallace\Documents\Coding stuff\electricity_bill_calculator>npm test
npm ERR! code EJSONPARSE
npm ERR! path C:\Users\Wallace\Documents\Coding stuff\electricity_bill_calculator/package.json
npm ERR! JSON.parse Unexpected string in JSON at position 228 while parsing '{
npm ERR! JSON.parse   "name": "lorikeet",
npm ERR! JSON.parse   "version": ""
npm ERR! JSON.parse Failed to parse JSON data.
npm ERR! JSON.parse Note: package.json must be actual JSON, not just JavaScript.

npm ERR! A complete log of this run can be found in:
npm ERR!     C:\Users\Wallace\AppData\Local\npm-cache\_logs\2022-05-20T19_53_07_684Z-debug-0.log

C:\Users\Wallace\Documents\Coding stuff\electricity_bill_calculator>npm test

> lorikeet@1.0.0 test
> jest

PASS ./sum.test.js
  ✓ adds 1+2 to equal 3 (3 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:  0 total
Time:        1.559 s
Ran all test suites.

C:\Users\Wallace\Documents\Coding stuff\electricity_bill_calculator>
```

Participation

Wallace Esteves

I was responsible for editing the report and CSS file and also some of the index.html file, me and Caio worked together to come up with a solution for the math operation.

Caio Machado

Since Caio is a genius in math he was responsible for the math operation, the functions on the “script.js” and “index.html”. He also makes the video for the code explanation.

Video Link

<https://drive.google.com/file/d/1pvO2loCQPP2D5F2irvfgKBNAeUcW7058/view>

GitHub Link

<https://github.com/wallacestev3s/electricity-bill-calculator>