

Build your own web app

Why?

- NOT to become a software engineer
- Show you basics of how web apps work
- Have more meaningful discussion about web applications (understand the lingo)
- Give you confidence to learn more/ explore about code!
- Give you something real to take home!

Setup

Download and install:

- Atom (<https://atom.io/>)
- Zeit Now (<https://zeit.co/download>)
- Download Google-Drive folder (tinyurl.com/my-parking-app)

Setup

Open the folder using Atom.

Open the HTML file using Google Chrome.

**What you're going to
build**

How we're building this

1. Input box
2. Convert to X and Y
3. Find nearest 10 car parks
4. Styling the web page

Suggested workflow

1. I'll write a small code snippet
2. I'll run the snippet to show you what it does
3. I'll explain the changes
4. You type out and try out the changes on your laptops!

Displaying some text


```
<body>  
  hello  
</body>
```

```
<body>  
  hello there!  
</body>
```

```
<body>  
  hello<br>  
  there!  
</body>
```

```
<body>  
  <input>  
</body>
```

```
<body>  
  <div>  
    <input>  
  </div>  
</body>
```

```
<body>  
  <div>  
    <input placeholder="Enter a postcode">  
  </div>  
</body>
```



```
<body>
  <div>
    <input placeholder="Enter a postcode">
  </div>
</body>
```

```
<script>
  alert("Hello!")
</script>
```



```
<body>  
  <div>  
    <input placeholder="Enter a postcode">  
  </div>  
</body>
```

```
<script>  
  console.log("Hello!")  
</script>
```

```
<head>  
  <script src="https://unpkg.com/vue"></script>  
</head>
```

```
<body>  
  <div>  
    <input placeholder="Enter a postal code">  
  </div>  
</body>
```

```
<script>  
  console.log("Hello!")  
</script>
```

```
<head>  
  <script src="https://unpkg.com/vue"></script>  
</head>
```

```
<body>  
  <div>  
    <input placeholder="Enter a postal code">  
  </div>  
</body>
```

```
<script>  
  new Vue()  
</script>
```

```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>

<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postal code">
  </div>
</body>

<script>
  var config = {
    el: "#myParkingApp"
  }

  new Vue(config)
</script>
```

```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>

<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode">
      {{ postcode }}
    </div>
  </body>

  <script>
    var config = {
      el: "#myParkingApp",
      data: {
        postcode: "123456"
      }
    }

    new Vue(config)
  </script>
```

```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>

<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode">
      {{ postcode }}
    </div>
  </body>

  <script>
new Vue({
  el: "#myParkingApp",
  data: {
    postcode: ""
  }
})
</script>
```

We're at our first milestone!

Next step, converting the postcode into x and y .

carpark.html

```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>

<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode">
    {{ postcode }}
  </div>
</body>

<script src="carpark-logic.js"></script>
```


carpark-logic.js

```
var config = {  
  el: "#myParkingApp",  
  data: {  
    postcode: ""  
  }  
}
```

```
new Vue(config)
```

carpark.html

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode">
    {{ postcode }}
    {{ postcodeX }}
    {{ postcodeY }}
  </div>
</body>
```

carpark-logic.js

```
var config = {  
  el: "#myParkingApp",  
  data: {  
    postcode: "",  
    postcodeX: "111",  
    postcodeY: "222"  
  }  
}
```

carpark.html

```
<head>
  <script src="https://unpkg.com/vue"></script>
</head>

<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}<br>
  </div>
</body>

<script src="carpark-logic.js"></script>
```

carpark.html

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="postcode = 275983">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}<br>
  </div>
</body>
```

carpark.html

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}<br>
  </div>
</body>
```

carpark-logic.js

```
var config = {  
  el: "#myParkingApp",  
  data: {  
    postcode: "",  
    postcodeX: "",  
    postcodeY: ""  
  },  
  methods: {  
    searchPostcode: function() {  
      this.postcodeX = "333"  
      this.postcodeY = "444"  
    }  
  }  
}
```

How can we convert a postcode into x and y ?

We can "phone a friend"!

carpark.html

```
<head>  
  <script src="https://unpkg.com/vue"></script>  
  <script src="https://unpkg.com/axios/dist/axios.min.js"></script>  
</head>
```

carpark-logic.js

```
searchPostcode: function() {  
  axios.get("https://developers.onemap.sg/commonapi/search", {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  })  
}
```

carpark-logic.js

```
searchPostcode: function() {  
  axios.get("https://developers.onemap.sg/commonapi/search", {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  }).then(response => {  
    console.log(response)  
  })  
}
```

carpark-logic.js

```
searchPostcode: function() {  
  axios.get("https://developers.onemap.sg/commonapi/search", {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  }).then(response => {  
    console.log(response.data)  
  })  
}
```

carpark-logic.js

```
searchPostcode: function() {  
  axios.get("https://developers.onemap.sg/commonapi/search", {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  }).then(response => {  
    console.log(response.data.results[0])  
  })  
}
```

carpark-logic.js

```
searchPostcode: function() {  
  axios.get("https://developers.onemap.sg/commonapi/search", {  
    params: {  
      searchVal: this.postcode,  
      returnGeom: "Y",  
      getAddrDetails: "N"  
    }  
  }).then(response => {  
    var searchResult = response.data.results[0]  
    this.postcodeX = searchResult.X  
    this.postcodeY = searchResult.Y  
  })  
}
```

More than halfway done now!

Next, get the nearest car parks and display them.

carpark.html

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">

    <div v-for="carpark in carparks">
      {{ carpark }}
    </div>
  </div>
</body>
```


carpark-logic.js

```
data: {  
  postcode: "",  
  postcodeX: "",  
  postcodeY: "",  
  carparks: []  
}
```

carpark-logic.js

```
methods: {
  searchPostcode: function() {
    axios.get("https://developers.onemap.sg/commonapi/search", {
      params: {
        searchVal: this.postcode,
        returnGeom: "Y",
        getAddrDetails: "N"
      }
    }).then(response => {
      var unwrappedResults = response.data.results[0]
      this.postcodeX = unwrappedResults.X
      this.postcodeY = unwrappedResults.Y
      this.getNearestCarparks()
    })
  },

  getNearestCarparks: function() {
    this.carparks = ["carpark1", "carpark2", "carpark3"]
  }
}
```

Recall:

```
searchPostcode: function() {  
  axios.get("https://developers.onemap.sg/commonapi/search", {  
    ...  
  }).then(response => {  
    ...  
  })  
}
```

Recall:

```
searchPostcode: function() {  
  axios.get("https://developers.onemap.sg/commonapi/search", {  
    ...  
  }).then(response => {  
    ...  
  })  
}
```

I wish we could do:

```
getNearestCarparks: function() {  
  axios.get('https://getcarparks.com/').then(carparks => ...)  
}
```

carpark.html

<head>

...

</head>

<body>

...

</body>

<script src="carpark-helpers.js"></script>

<script src="carpark-logic.js"></script>

carpark-logic.js

```
getNearestCarparks: function() {  
    getCarparkList(this.postcodeX, this.postcodeY)  
        .then(carparks => {  
  
        })  
    }  
}
```

carpark-logic.js

```
getNearestCarparks: function() {  
  getCarparkList(this.postcodeX, this.postcodeY).then(carparks => {  
    console.log(carparks)  
  })  
}
```

carpark-logic.js

```
getNearestCarparks: function() {  
  getCarparkList(this.postcodeX, this.postcodeY).then(carparks => {  
    this.carparks = carparks  
  })  
}
```


All done with the logic.

Now we're going to make things look nicer!

carpark.html

```
<body>
  <div id="myParkingApp">
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}

    <div v-for="carpark in carparks">
      {{ carpark }}
    </div>
  </div>
</body>
```

carpark.html

```
<body>
  <div id="myParkingApp">
    <h1>Nearest Carpark Availability</h1>
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}

    <div v-for="carpark in carparks">
      {{ carpark }}
    </div>
  </div>
</body>
```

carpark.html

```
<body>
  <div id="myParkingApp">
    <h1>Nearest Carpark Availability</h1>
    <input placeholder="Enter a postcode"
      v-model="postcode"
      v-on:keyup.enter="searchPostcode">
    {{ postcode }}<br>
    {{ postcodeX }}<br>
    {{ postcodeY }}

    <div v-for="carpark in carparks">
      {{ carpark }}
    </div>
  </div>
</body>
```

carpark.html

```
<div v-for="carpark in carpark">  
  {{ carpark.distance }}<br>  
  {{ carpark.address }}<br>  
</div>
```

carpark.html

```
<div v-for="carpark in carpark">  
  Distance: {{ carpark.distance }}m <br>  
  Address: {{ carpark.address }}<br>  
</div>
```

carpark.html

```
<div v-for="carpark in carpark">
  Distance: {{ carpark.distance }}m<br>
  Address: {{ carpark.address }}<br>
  Total Lots: {{ carpark.total_lots }} <br>
  Lots Available: {{ carpark.lots_available }} <br>
  <br>
</div>
```

Now the content looks the same, but the styling of the content is different!

Recall how we did styles like background-color, text-align

Recall how we did styles like background-color, text-align

Doing all the styling like that makes it difficult to read and change our styles

Just like our Javascript code, we can put all our styles in a separate file!

carpark.html

```
<head>
  <script src="https://unpkg.com/vue"></script>
  <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
  <link rel="stylesheet" type="text/css" href="carpark.css" />
</head>
```

carpark.css

```
body {  
    background-color: lightsteelblue;  
    text-align: center;  
}
```

```
input {  
    width: 50%;  
    font-size: 20px;  
}
```

carpark.html

```
<body>
  <div id="myParkingApp">
    <div id="inputBackground">
      <h1>Nearest Carpark Availability</h1>
      <input placeholder="Enter a postcode"
        v-model="postcode"
        v-on:keyup.enter="searchPostcode">
      <br>
    </div>

    <div id="carparkResults" v-for="carpark in carpark">
      Distance: {{ carpark.distance }}m<br>
      Address: {{ carpark.address }} <br>
      Total Lots: {{ carpark.total_lots }} <br>
      Lots Available: {{ carpark.lots_available }} <br>
      <br>
    </div>
  </div>
</body>
```

carpark.html

```
<head>  
  <script src="https://unpkg.com/vue"></script>  
  <script src="https://unpkg.com/axios/dist/axios.min.js"></script>  
  <link rel="stylesheet" type="text/css" href="carpark-style.css" />  
</head>
```

Deploying your app to the internet!

That's all folks :)