Npm install -g vue@next

Installing CLI

Npm install -g @vue/cli

Vue app using command

Vue create projectname

Options API

Name :

Data() {}

Methods: {}

Template :

Composition API – setup() {

Data declaration,

Function declaration

}

e.g this is normal js function which does not have reactive data in it

function sum()

{

Var val1 = 9;

Var val2 = 8;

Var sum = val1 + val2; // 17

Val2 = 12;

Console.log(sum); //17

}

Using composition API we can deal with reactive data in vue app.

Use of v-bind – it is used assign dynamic values to html attributes.

v-html –

{{}} – interpolation – data binding - it renders plain text

To output real HTML we have to make use v-html directive.

Event Handling

* In Vue you can make use of v-on direrctive for listening to DOM event.
* For button v-on:click or you can make use of shorthand @click

EventModifiers

.stop – stopPropogation

.prevent – preventDefault

.capture

.self

.once

.passive

KeyModifier - @keyup, @keydown @keypress

Styling with CSS and Bootstrap

Classes active and text-danger should be applied dynamically depending on some scenarios

e.g active : this.isActive && !this.error

‘text-danger’ : this.error

Here to use this expression we have to make use of computed property on a vue instance.

To use bootstrap 3 in vue 3

Npm install bootstrap-vue bootstrap –save

Module 5 - Rendering Conditionally contents and list

v-if, v-else, v-else-if directive are used for conditional rendering

conditional rendering is also possible using v-show

difference in v-show and v-if,v-else,v-if-else – v-show is always rendered and it remains in DOM

and it toggle display css property to none if condition is false

v-if real conditional rendering that means element is rendered only when condition is satisfied or true else not .

v-if has higher toggle cost, v-show has higher initial render cost. If you want to toggle something frequently then use v-show else if rarely if you want to toggle something then use v-if.

**Module 3 : Methods & Event Handling**

* Event binding

<https://v3.vuejs.org/guide/events.html>

<https://v3.vuejs.org/guide/events.html#listening-to-events>

* Event & Methods

<https://v3.vuejs.org/guide/events.html#method-event-handlers>

Methods in inline handlers

<https://v3.vuejs.org/guide/events.html#methods-in-inline-handlers>

Multiple event handlers

<https://v3.vuejs.org/guide/events.html#multiple-event-handlers>

Event Modifiers

<https://v3.vuejs.org/guide/events.html#event-modifiers>

Key Modifiers

<https://v3.vuejs.org/guide/events.html#key-modifiers>

Videos :

<https://www.vuemastery.com/courses/intro-to-vue-3/event-handling-vue3>

* Computed Properties & Watchers

[https://v3.vuejs.org/guide/computed.html#computed-properties](https://v3.vuejs.org/guide/computed.html)

* Computed Properties Vs Methods

<https://v3.vuejs.org/guide/computed.html#computed-caching-vs-methods>

* Computed Properties Vs Watchers

<https://v3.vuejs.org/guide/omputed.html#computed-vs-watched-property>

* v-on & shorthands

https://v3.vuejs.org/api/directives.html#v-on

**Module 4 : Styling with CSS and Bootstrap:**

Dynamic Styling with Inline Styles & Adding CSS Classes Dynamically

<https://v3.vuejs.org/guide/class-and-style.html#binding-inline-styles>

<https://michaelnthiessen.com/dynamically-add-class-name/>

Classes & Computed Properties

<https://www.w3resource.com/vue/class-and-style-bindings.php>

Videos :

<https://www.youtube.com/watch?v=EukZKT5JpZQ>

Dynamic Classes: Array Syntax

<https://michaelnthiessen.com/dynamically-add-class-name/>

Videos :

<https://www.youtube.com/watch?v=HITR6m2lGPA>

Styling using Bootstrap

<https://www.techiediaries.com/vue-bootstrap/>

<https://www.positronx.io/easy-method-to-add-bootstrap-to-vue-cli-project/>

Videos :

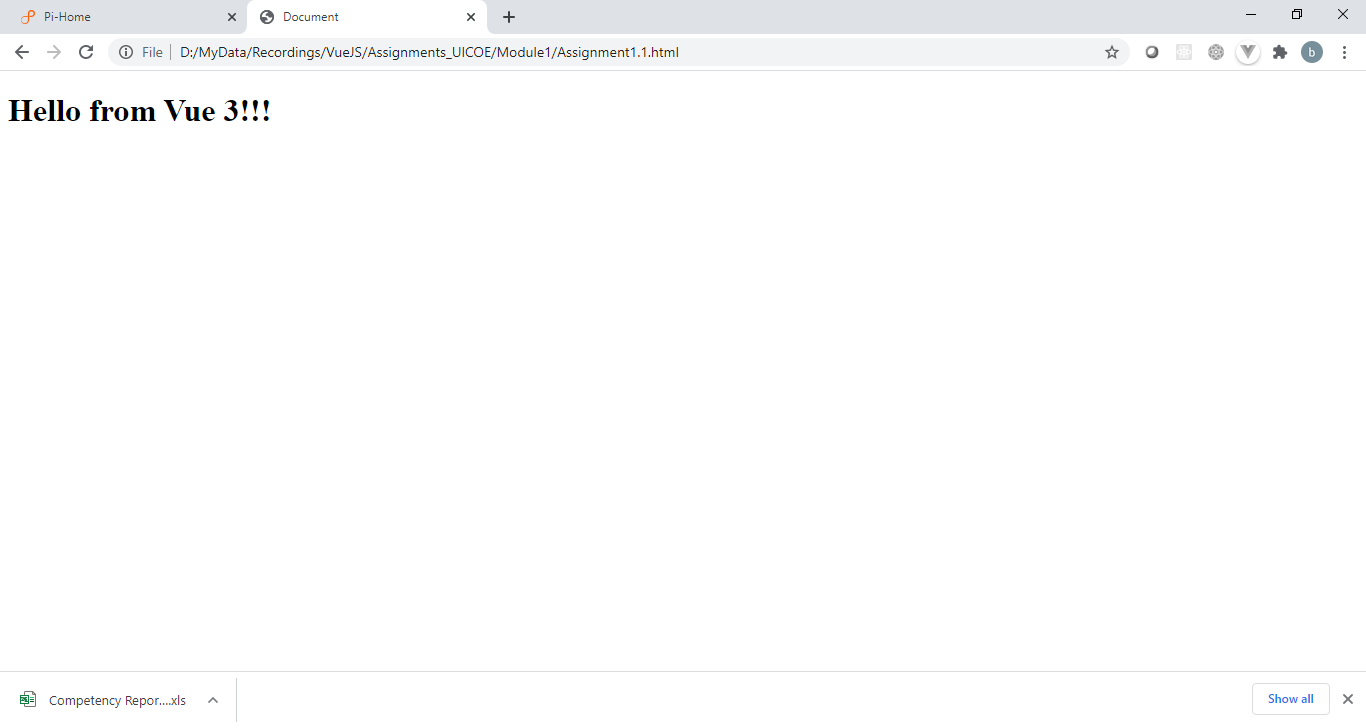
<https://www.positronx.io/easy-method-to-add-bootstrap-to-vue-cli-project/>

Assignments :

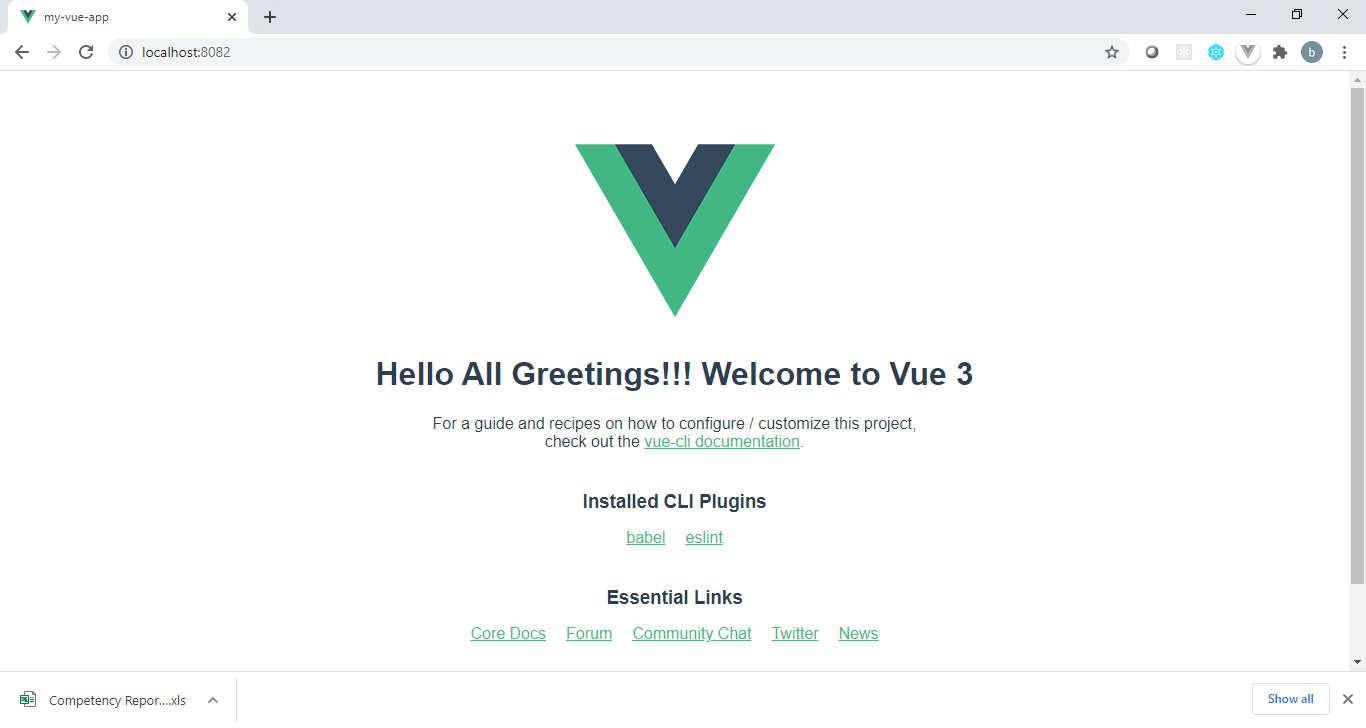
Assignments

**Module 1 : Introduction To Vue.js**

**1.1** Create a Vue instance that will generate below view. Use CDN links for Vue 3 library in html.

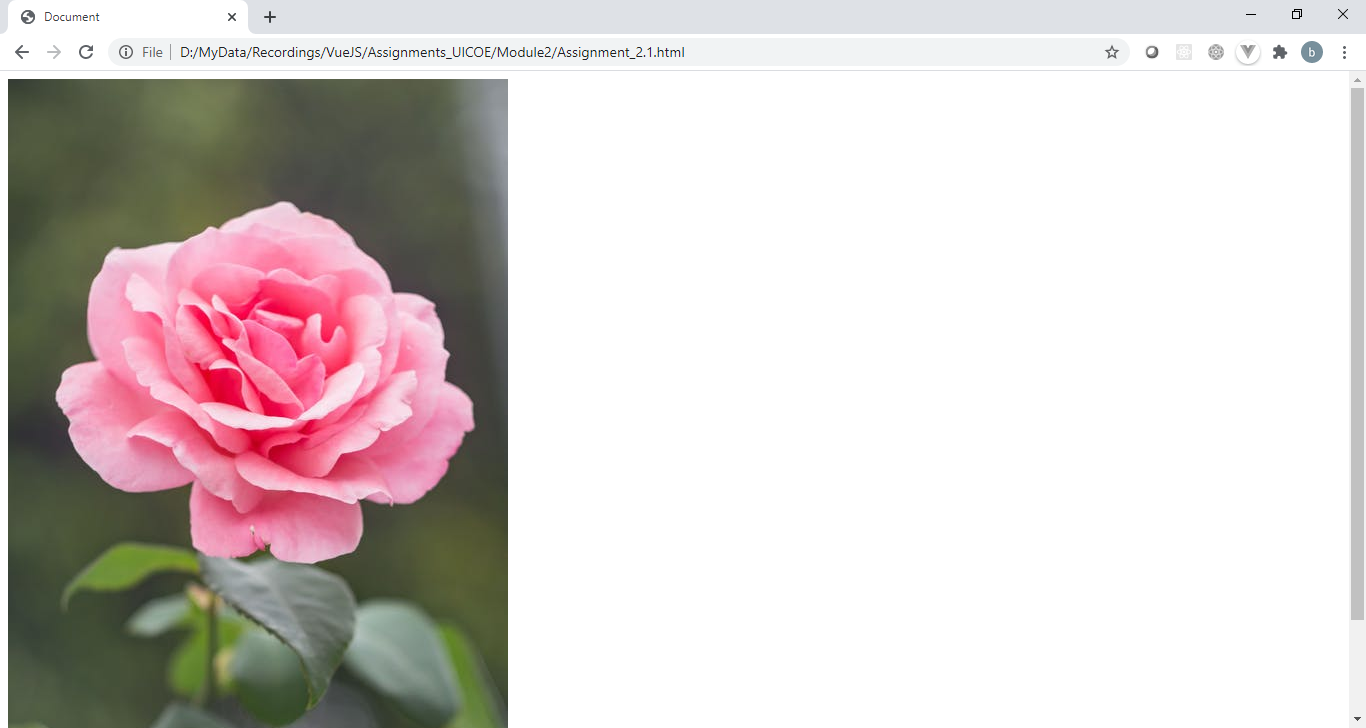


**1.2** Create a Vue 3 app using Vue CLI 4, that will generate below output.

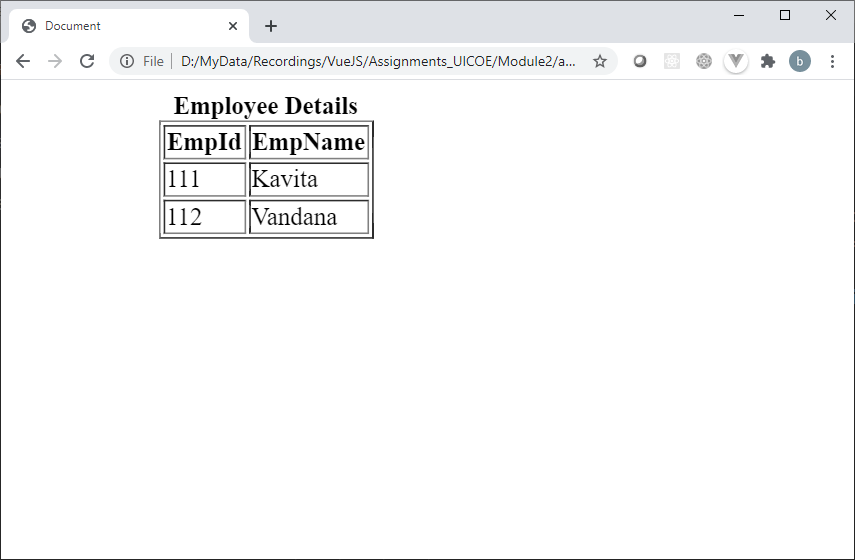


**Module 2 :** **DOM Interaction With Vue 3**

**2.1** Create a Vue app instance with data, url. In the template of it, bind the url from the Vue instance to the src attribute of an html element <img> and generate below output.



2.2 Create a Vue instance with data mytable. Data mytable holds html in it i.e table of employees. Bind the property mytable to the span in a div with id, app in such a manner that it should generate below view.



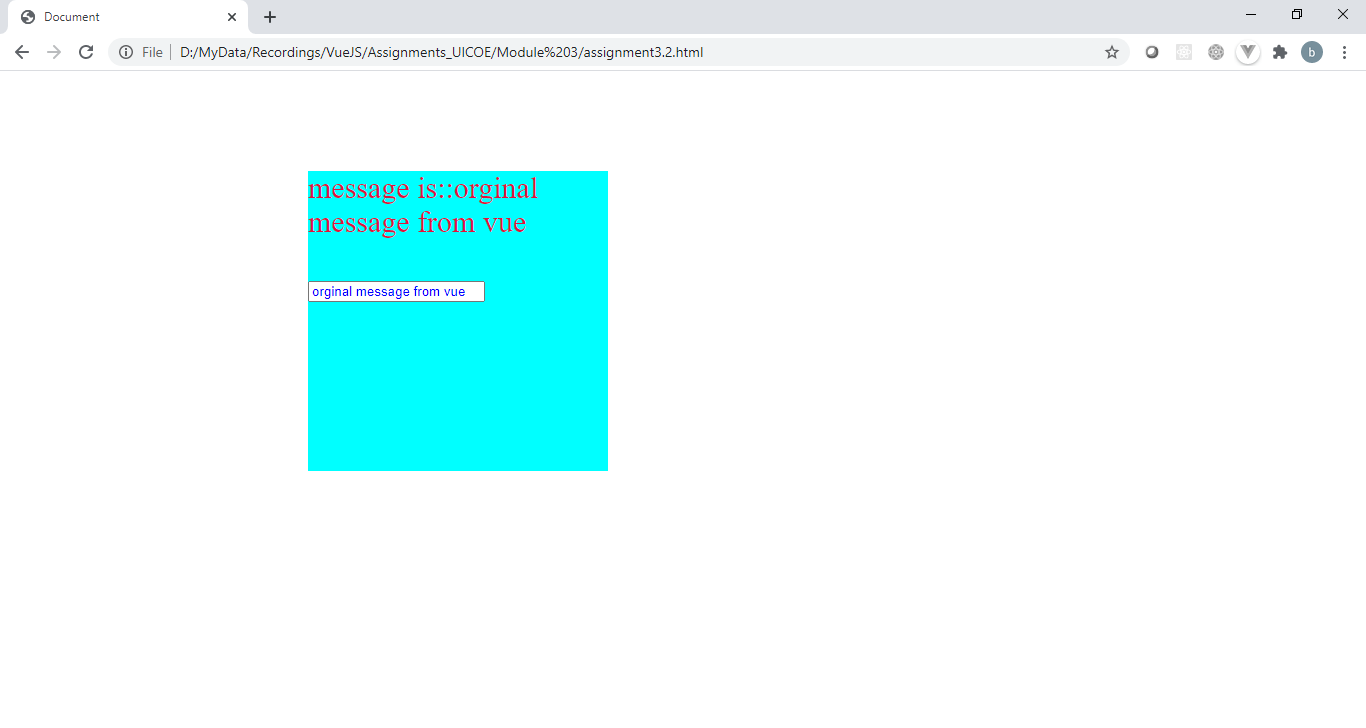
**Module 3:** **Methods & Event Handling**

3.1 Generate below output by creating a Vue 3 instance.

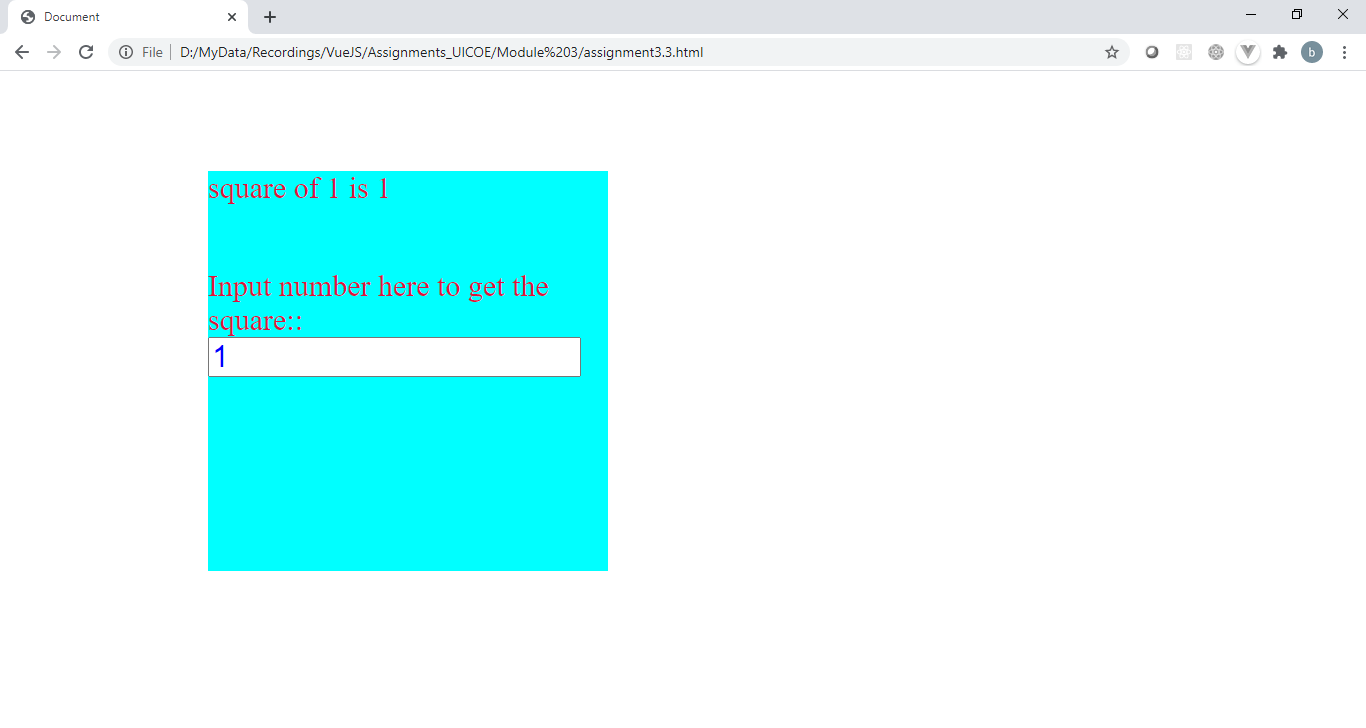


When the button, get will be clicked it should multiply the shown number by 2 on each click and the multiplied value should be displayed number::value. Value of number must be a binded property from vue instance.

3.2 Create a Vue 3 instance that will generate below output i.e. data message and text field binded with the message. On change of the value in the text field, data message of Vue3 instance should be updated using event handler updateMessage() and the changes must be reflected on UI.

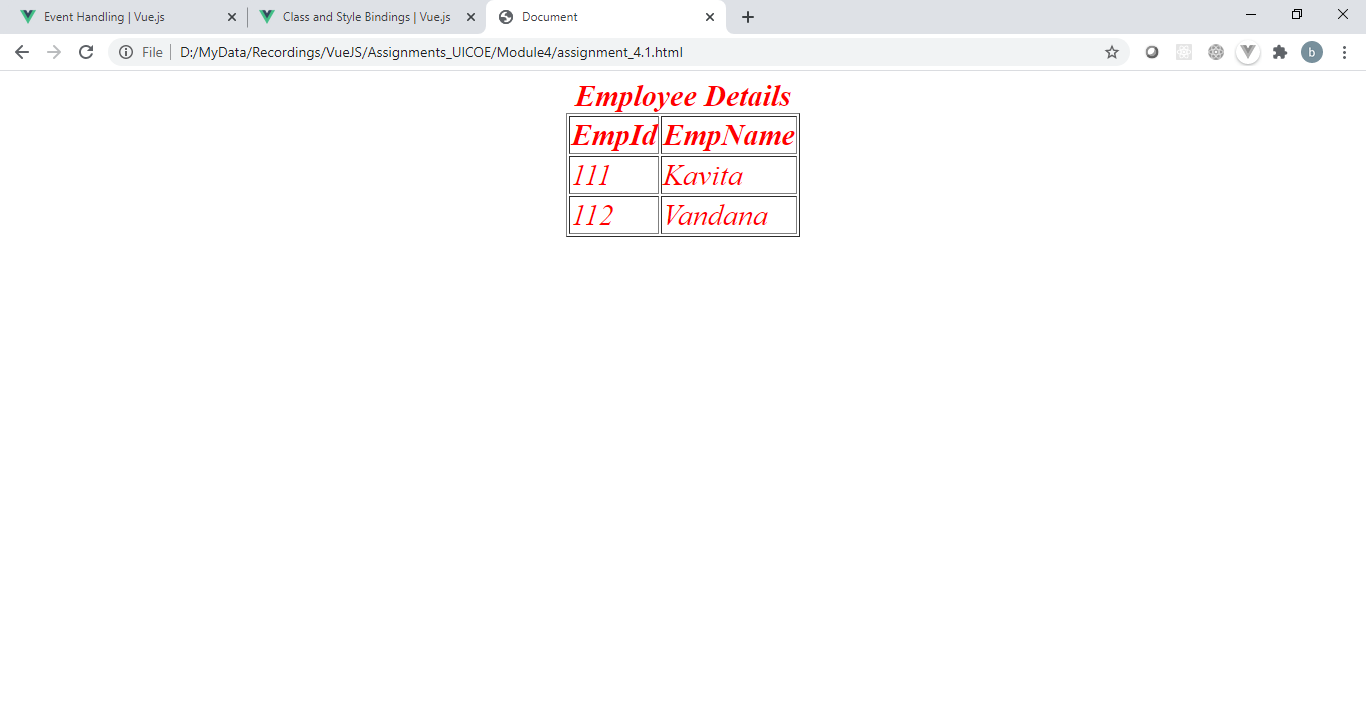


3.3 Create a Vue instance with data square and number. When rendered should generate UI as given below i.e. text field which accepts number and above it square of the entered number shown. On every input the entered value must be stored in data number of Vue instance and square of it should get stored in the data store of it.

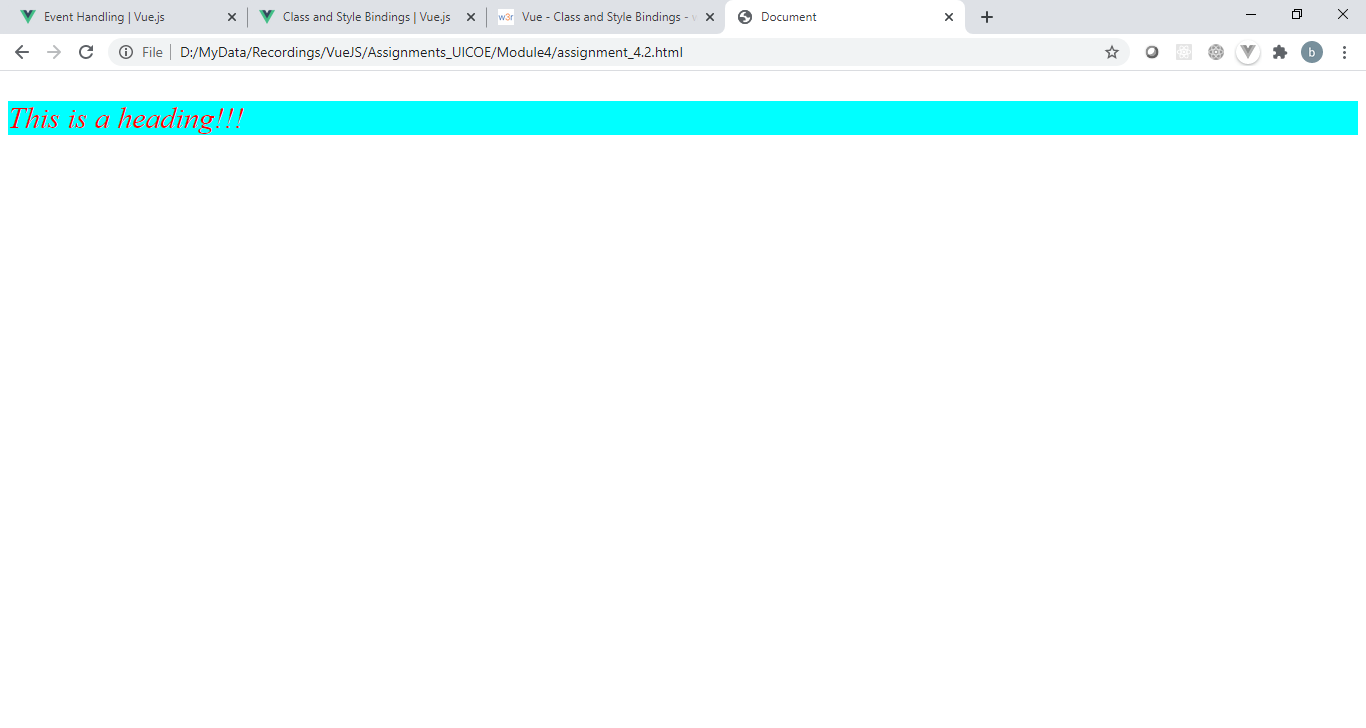


**Module 4 :** **Styling with CSS and Bootstrap**

4.1 Create a Vue 3 app instance that will generate below given output using v-html directive and css styling.



4.2 Create a Vue app instance that will generate below given output using Object Styling i.e. background color and text color should be applied using object styling.



4.3 Create a vue application using Vue CLI 4 and apply the styling using bootstrap to generate a view

as given below. Also it will add one flower image in it on render of HelloWorld component as shown below.

