

SIT120 Practical

Week 6 – Vue.js Framework (Handling User Inputs)



GOALS:

- Gain better understanding of Vue.js Framework and its different components.
- Understand Handling User Inputs, Form Input Bindings and their components in depth
- How binding and Modifiers work?

Task 1: Using V-model for handling user inputs

When user types in input, V-model will automatically accommodate that value in local data. How can we do this? Lets try to implement the following using V-model.

Form User

Name:

Password:

Email:

Task 2: Checkbox in your project

Visit and understand

<https://v2.vuejs.org/v2/guide/forms.html#Checkbox>

Multiple checkboxes, bound to the same Array:

```
<input type="checkbox" id="jack" value="Jack" v-model="checkedNames">
<label for="jack">Jack</label>
<input type="checkbox" id="john" value="John" v-model="checkedNames">
<label for="john">John</label>
<input type="checkbox" id="mike" value="Mike" v-model="checkedNames">
<label for="mike">Mike</label>
<br>
<span>Checked names: {{ checkedNames }}</span>
```

HTML

```
new Vue({
  el: '...',
  data: {
    checkedNames: []
  }
})
```

JS

☐ Jack ☐ John ☐ Mike
Checked names: []

Apply and implement similar concept in your project.

Task 3: Dynamic options rendering using v-for



Visit and understand <https://v2.vuejs.org/v2/guide/forms.html#Select>

Select

Single select:

```
<select v-model="selected">
  <option disabled value="">Please select one</option>
  <option>A</option>
  <option>B</option>
  <option>C</option>
</select>
<span>Selected: {{ selected }}</span>
```

```
new Vue({
  el: '...',
  data: {
    selected: ''
  }
})
```

B Selected: B

Implement dynamic rendering with v-for in your project.

Additional information: Using Modifiers in your project



1. Do you want to use **lazy modifiers** to sync after change events? Think and Discuss with peers

```
<!-- synced after "change" instead of "input" -->  
<input v-model.lazy="msg">
```

2. Do you need to trim whitespace from user inputs automatically?

```
<input v-model.trim="msg">
```

3. What about automatic typecasting? Will that be useful for your project?

```
<input v-model.number="age" type="number">
```