

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

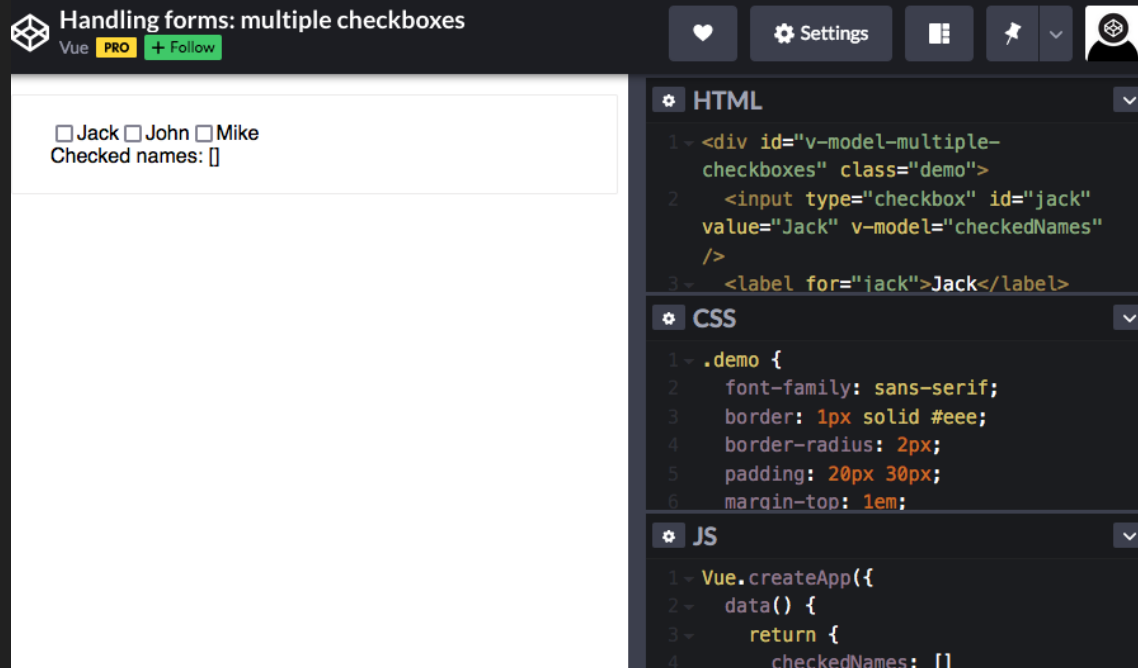
Name:

Password:

Email:

Task 2: Checkbox in your project

Visit and understand <https://codepen.io/team/Vue/pen/bGdmoyj>



Apply and implement similar concept in your project.

Task 3: Dynamic options rendering v-for

Visit and understand <https://codepen.io/team/Vue/pen/abORVZm>

Handling forms: select with dynamic options
Vue PRO + Follow

One ▾ Selected: A

```
HTML
1 <div id="v-model-select-dynamic"
  class="demo">
2 <select v-model="selected">
3   <option v-for="option in
  options"
  v-bind:value="option.value">
```

```
CSS
1 .demo {
2   font-family: sans-serif;
3   border: 1px solid #eee;
4   border-radius: 2px;
5   padding: 20px 30px;
6   margin-top: 1em;
```

```
JS
1 Vue.createApp({
2   data() {
3     return {
4       selected: 'A',
5       options: [
```

Implement dynamic rendering with v-for in your project.

Task 4: 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">
```