COS30043
Interface Design and Development

SWINBURNE UNIVERSITY OF TECHNOLOGY

Lecture 6 – Form and Validation

2022 – Semester 1

KNOW ING

Contents

Form Validation:

– Using HTML 5 Form Attributes

– Using VueJS

– Using Vuetify

HTML 5 Form Validation Attributes

- **required**: Specifies whether a form field needs to be filled in before the form can be submitted.
- **minlength and maxlength**: Specifies the minimum and maximum length of textual data (strings)
- **min and max**: Specifies the minimum and maximum values of numerical input types
- **type**: Specifies whether the data needs to be a number, an email address, or some other specific preset type.
  - <input type=“date” …
  - <input type=“time”…
  - <input type=“email” …
- **pattern**: Specifies a regular expression that defines a pattern the entered data needs to follow.

HTML 5 Form Validation Example

```
<form  method="post"
action="https://mercury.swin.edu.au/it000000/formtest.php" >

<p>First Name:<input type="text" name="fName"
      required minlength="5" maxlength="8" ></p>
<p>Age:<input type="number" name="age"   min="10"
      max="50" ></p>
<p>Date:<input type="date" name="date"  ></p>
<p>Post Code:<input type="text" name="postcode"
      pattern="\d{4}" ></p>

<p><input type="submit" value="submit"></p>
</form>
```

Contents

Form Validation:

- Using HTML 5 Form Attributes
- Using VueJS
- Using Vuetify

VueJS Form Validation

- If you want VueJS completely control over the validations
  - use form’s novalidate attribute to disable the HTML 5 validations supported by modern browsers

    Example: <form novalidate >

VueJS Form Validation – HTML

```
<div id="app">
  <form @submit="checkForm“ method="post“
     action="https://mercury.swin.edu.au/it000000/formtest.php" novalidate >
  <div v-if="errors.length">
    <p>Please correct the following error(s):</p>
    <ul> <li v-for="error in errors">{{ error }}</li> </ul>
  </div>
```

Error message

```
  <p>
    <label for="fName">First Name</label>
    <input type="text" name="fName" id="fName" v-model="fName"/>
  </p>
```

Text input

```
  <p><input type="submit" value="submit">      </p>
```

Submit button

```
 </form>
</div>
```

VueJS Form Validation – Vue JS

```
const app = Vue.createApp({
     data() {   return {  fName: ‘’,   errors: [ ]    } },

     methods: {
         checkForm: function(e) {
                    this.errors = [ ];   var result = true;
                    if ( !this.fName) {
                          this.errors.push("First name required")
                          result =   false;
                    }
                    if (!result)
                          e.preventDefault();  //prevent form submission
         }
     }
})
app.mount('#app')
```

Contents

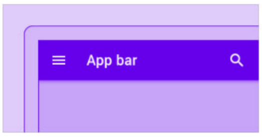
Form Validation:

- – Using HTML 5 Form Attributes
- – Using VueJS
- – Using Vuetify

Vuetify

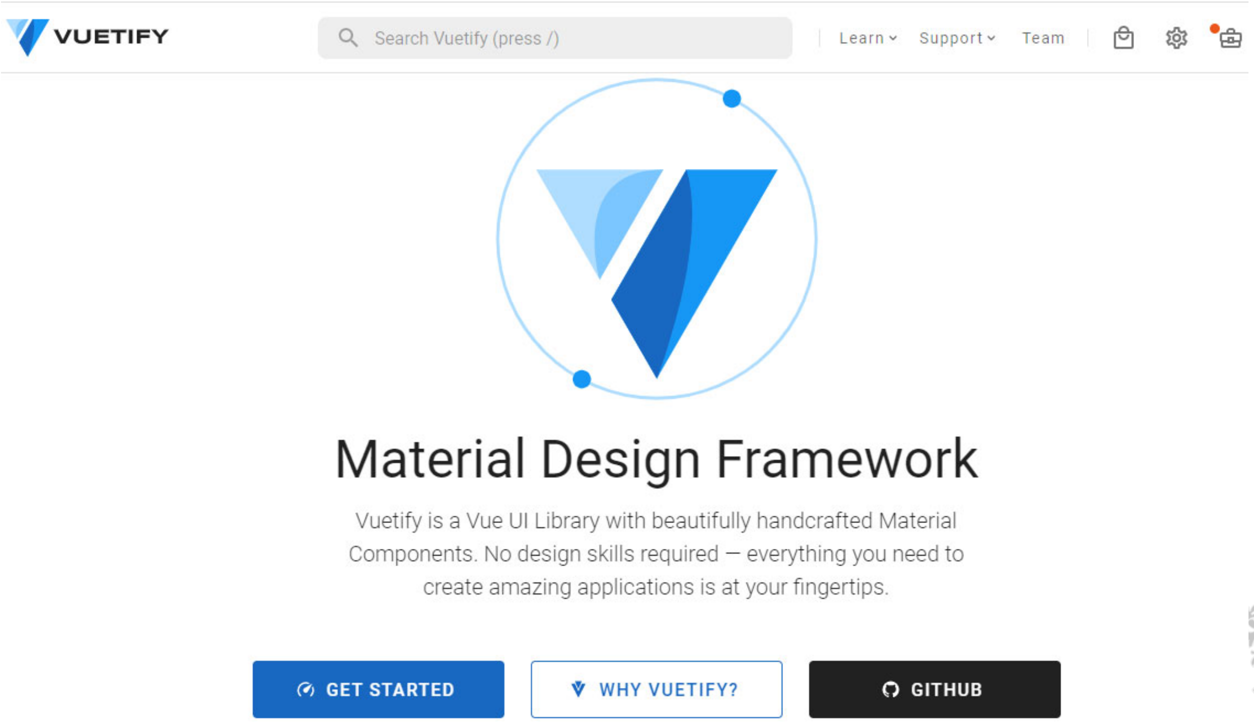
- Complete UI framework built on top of Vue.js
- Vuetify is a Vue UI Library with Material design components

- **Material design (https://material.io/) was created by Google to help teams build high-quality digital experiences for Android, iOS, Flutter, and the web.**
- **Material Design is inspired by the physical world and its textures, including how they reflect light and cast shadows. Material surfaces reimagine the mediums of paper and ink.**
- **There are different components for creating user interface, such as cards, floating buttons, app bar, text fields…**



Vuetify

- https://vuetifyjs.com/



Material Design Framework

Vuetify is a Vue UI Library with beautifully handcrafted Material Components. No design skills required — everything you need to create amazing applications is at your fingertips.

Vuetify - Installation

- To use Vuetify, in the <head> section, use <link> link to
  - Google font
  - Material design icon
  - Vuetify css
- Use <script> link to
  - vue.js
  - vuetify.js

The current released version of Vuetify does not support Vue 3. Support for Vue 3 will come with the release of Vuetify v3.

We can download Vuetify version 3 beta from

https://github.com/vuetifyjs/vuetify/releases/tag/v3.0.0-beta.0

Template for VueJS with Vuetify

```
<!DOCTYPE html>
<html>
<head>
<link href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900"
        rel="stylesheet">
<link href="https://cdn.jsdelivr.net/npm/@mdi/font@6.x/css/materialdesignicons.min.css"
        rel="stylesheet">
<link href="js/vuetify-v3.0.0-beta.0.min.css" rel="stylesheet">
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
    <div id="app">
        <v-app>
            <v-main>
                <v-container>Hello world</v-container>
            </v-main>
        </v-app>
    </div>
```

The v-app component is a mandatory wrapper required for your application to work properly.

Template for VueJS with Vuetify (continued)

```
        <script src="js/vue.js"></script>
        <script src="js/vuetify-v3.0.0-beta.0.min.js"></script>

        <script>
            const app = Vue.createApp({ })
            const vuetify = Vuetify.createVuetify( )

            app.use(vuetify)
            app.mount('#app')
        </script>

    </body>
    </html>
```

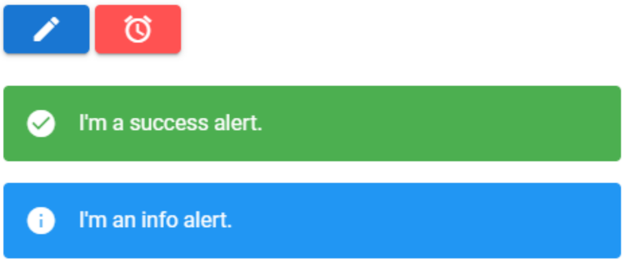
Vuetify Example

```
<v-container fluid>
        <v-btn  color="primary" >
                <v-icon>mdi-pencil</v-icon>  <!-- mdi: material design icon -->
        </v-btn>

        <v-btn   color="error" >
                <v-icon>mdi-alarm</v-icon>
        </v-btn>
</v-container>

<v-container fluid>
        <v-alert type="success">
                I'm a success alert.
        </v-alert>

        <v-alert type="info">
                I'm an info alert.
        </v-alert>
</v-container>
```



I'm a success alert.

I'm an info alert.

Vue JS Form Validation

- We can validate forms using custom Vue logic
- We can also use Vuetify to build custom rules on each component to validate forms
- We can also use Vue frameworks like Vee-validate and vuelidate for form validation

Using Vuetify for form validation

- Vuetify has a component **v-form** which makes it easy to add validation to form inputs.
- All the input components has a *rules* prop which accepts a mixed array of types *function, boolean* and *string* .

Vuetify From Validation Example (HTML)

```
<v-form ref="myForm" method="post" v-model="valid"
action="http://mercury.swin.edu.au/it000000/formtest.php">

<v-text-field name="firstName" v-model="firstName"
      :rules="nameRules" label="First Name"  required >
</v-text-field>
<v-text-field name="email" v-model="email" type="email"
      :rules="emailRules" label="Email"  required >
</v-text-field>

<v-btn  type="submit" :disabled="!valid" color="success"
      @click="validate">Submit
</v-btn>
</v-form>
```

Vuetify From Validation Example (JS)

```
const vuetify = Vuetify.createVuetify( )
const app = Vue.createApp({
     methods: {
        validate() {this.$refs.myForm.validate() } // validate all inputs
     },
data: () => ({
      valid: true, firstName: '', email: ‘’,
      nameRules: [
        v => !!v || 'Name required',
        v => (v && v.length <= 10) || 'Name must be less than 10 characters'
      ],

     emailRules: [
        v => !!v || 'E-mail is required',
        v => /[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$/.test(v) || 'E-mail
must be valid',
      ]
})
})
app.use(vuetify)
app.mount('#app')
```

SWINBURNE UNIVERSITY OF TECHNOLOGY

WHAT’S NEXT?
- API 1

KNOW ING