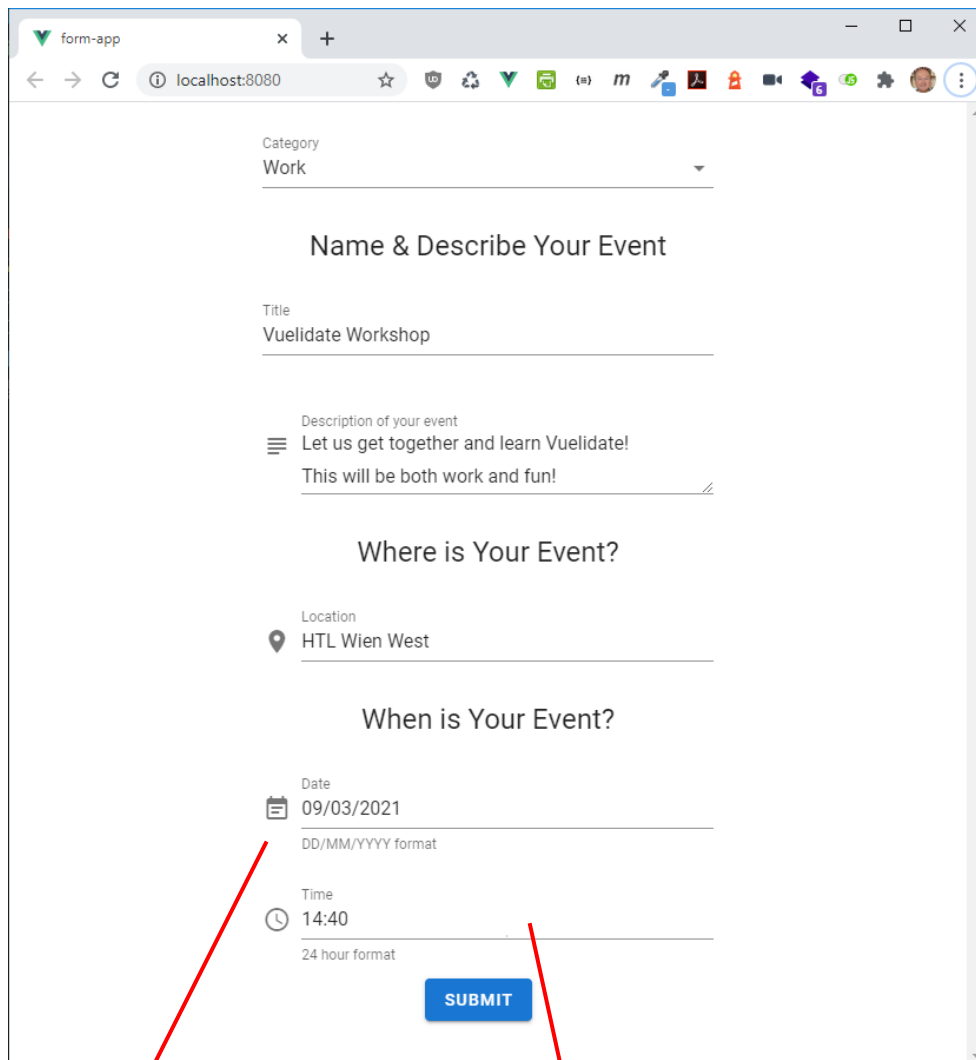
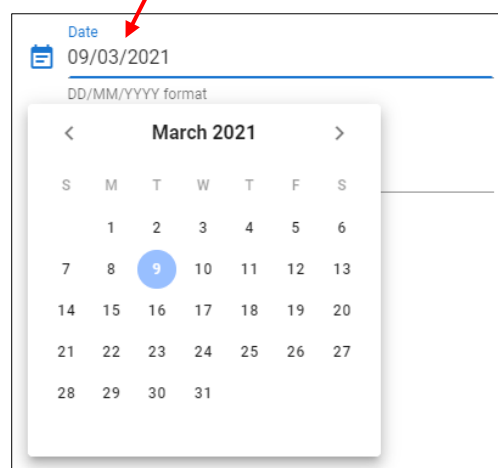
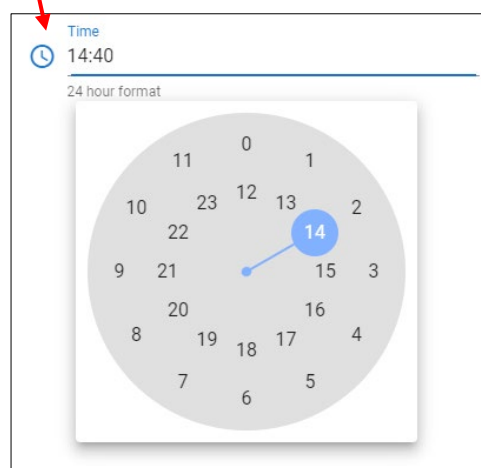


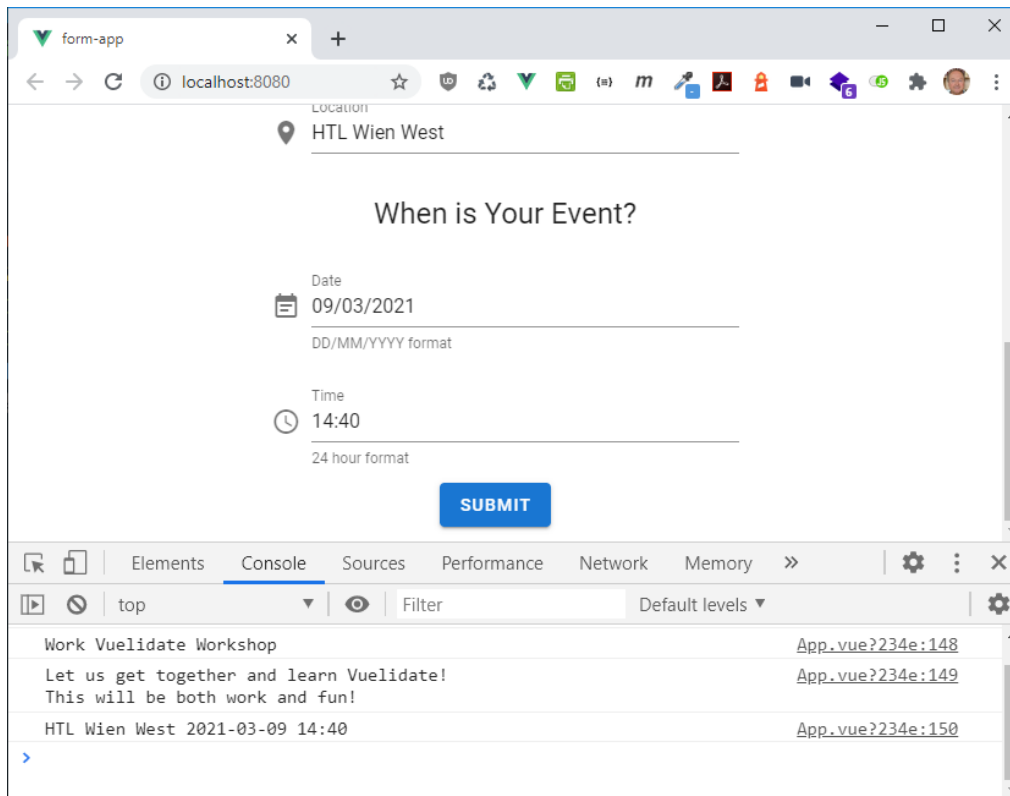
In diesem Arbeitsblatt gehen wir von einem bereits erstellten Formular aus. Du kannst zur Übung aber auch selbst das Formular mit Vuetify nachbauen!

1. Entpacke die Beispielapplikation **form-app 2 - Start.zip** und installiere die fehlenden Module.

Dein Ergebnis sollte so aussehen:

In der Console werden alle Werte, die eingegeben wurden, zu Testzwecken ausgegeben:



form-app

localhost:8080

Location  
HTL Wien West

When is Your Event?

Date  
09/03/2021  
DD/MM/YYYY format

Time  
14:40  
24 hour format

SUBMIT

Work Vuelidate Workshop App.vue?234e:148  
Let us get together and learn Vuelidate! App.vue?234e:149  
This will be both work and fun!  
HTL Wien West 2021-03-09 14:40 App.vue?234e:150

2. Installiere Vuelidate wie im Arbeitsblatt 1 beschrieben.

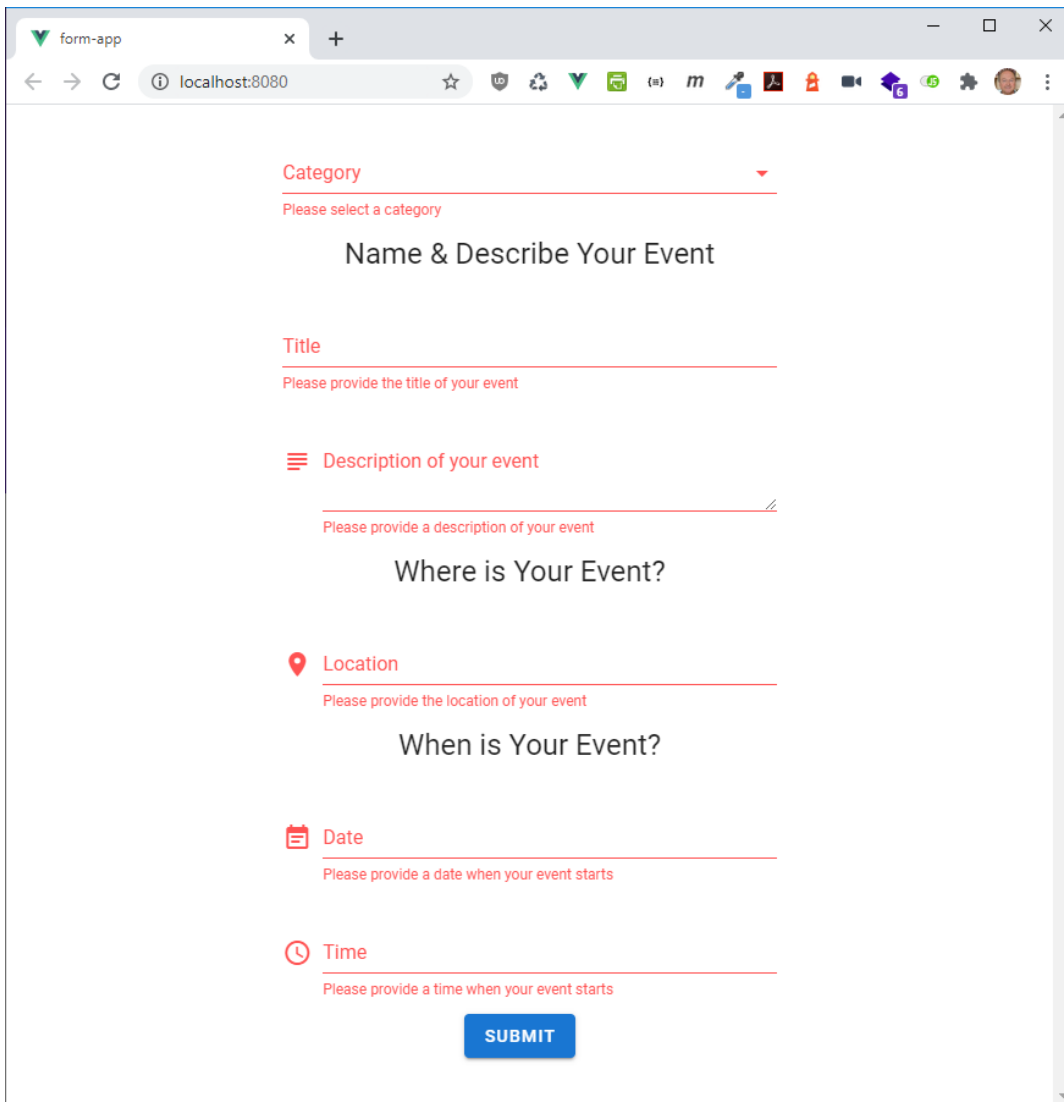
3. Importiere den Validator **required** in App.vue. Definiere das Property **validatons** und stelle sicher, dass folgende Eingaben required sind: **time**, **date**, **title**, **location**, **description**, **category**.

4. Erstelle nun für die Eingabefelder den nötigen Code, um den Fehlerfall abzufangen, dass nichts eingegeben wurde. Das bedeutet unter anderem, den **blur** Event abzufangen, mit **\$touch** das Property **\$dirty** zu setzen und computed Properties mit Fehlermeldungen zu definieren. Beispiel für computed Property **dateErrors**:

```
dateErrors() {
  let errors = [];
  if (!this.$v.date.$dirty) return errors;
  if (!this.$v.date.required) errors.push('Please provide a date when your event starts');
  return errors;
},
```

Hinweis Bei **date** und **time** sollte der **blur** Event beim **v-text-field** sein, nicht bei **v-menu**!

```
<v-text-field
  :error-messages="dateErrors"
  v-model="dateFormatted"
  label="Date"
  hint="DD/MM/YYYY format"
  prepend-icon="mdi-calendar-text"
  persistent-hint
  readonly
  v-bind="attrs"
  @blur="date = parseDate(dateFormatted); $v.date.$touch();"
  v-on="on"
></v-text-field>
```



form-app

localhost:8080

Category

Please select a category

### Name & Describe Your Event

Title

Please provide the title of your event

Description of your event

Please provide a description of your event

### Where is Your Event?

Location

Please provide the location of your event

### When is Your Event?

Date

Please provide a date when your event starts

Time

Please provide a time when your event starts

**SUBMIT**

5. Füge nun auch den Check für den Submit Button hinzu. Im Falle eines Fehlers irgendwo im Formular, soll der Button **disabled** sein. Dazu verwenden wir `$v.$invalid`.

```
<v-btn :disabled="$v.$invalid" @click="submit" class="mr-4 primary">submit</v-btn>
```

`date`, `time` und `category` müssen nicht weiter überprüft werden. Sie können syntaktisch nicht inkorrekt sein.

Allerdings wäre folgendes zusätzlich wünschenswert:

- `title` sollte mindestens 5 Stellen haben und mit einem Buchstaben beginnen, sonst nur Buchstaben, Ziffern und dazwischen liegenden Leerzeichen bestehen!
- `description` sollte mindestens 10 Stellen haben
- `Location` sollte mindestens 5 Stellen haben

Um die Mindestlänge sicher zu stellen können wir den Validator `minLength` verwenden (siehe Arbeitsblatt 1).

6. Implementiere die Checks für die Mindestlänge.

Um zu überprüfen, ob das erste Zeichen ein Buchstabe ist, schreiben wir einen Customvalidator. Das ist einfach eine simple Funktion, die `true` oder `false` zurückgibt, je nachdem, ob die Bedingung erfüllt ist oder nicht.

Zum Beispiel:

```
const startsWithLetter = (value) => (value && value[0].match(/[a-z]/i) ? true : false);
```

Diese können wir genau so wie die eingebauten Validators verwenden.

```
title: { required, minLength: minLength(5), startsWithLetter },
```

7. Baue diesen Check ein. Achtung, da nur die erste Fehlermeldung aus dem Array angezeigt wird, kommt es auf die Reihenfolge an! Also zuerst auf **required** checken!

8. Wir schreiben ebenfalls einen Customvalidator für den Check, dass nur **title** nur aus Buchstaben, Ziffern und dazwischen liegenden Leerzeichen bestehen soll.

```
const alphaNumSpace = (value) => (value && value.match(/^[a-z\d\s]+$/i) ? true : false);
```

Baue auch diesen Check ein.

Was aber, wenn der Benutzer nur Leerzeichen eingibt? Wird da nicht unser Längencheck ausgehebelt? Ja. Deswegen geben wir zu **v-model** den Modifier **trim** hinzu, der entfernt alle vorangehenden und nachfolgenden Leerzeichen!

Beispiel:

```
<v-text-field
  :error-messages="titleErrors"
  label="Title"
  v-model.trim="title"
  @blur="$v.title.$touch()"
>
</v-text-field>
```



Nun sollten alle Checks implementiert sein!

9. Erinnerst du dich noch an das Negativbeispiel aus dem Arbeitsblatt 1? Füge nun eine Checkbox hinzu, die angehakt werden muss, bevor das Formular abgeschickt werden kann.

```
<v-row justify="center">
  <v-col col="12" sm="6">
    <v-checkbox
      v-model="termsAndConditions"
      :errorMessages="termsAndConditionsErrors"
      label="I accept the terms and conditions for HTL events"
    ></v-checkbox>
  </v-col>
</v-row>
```

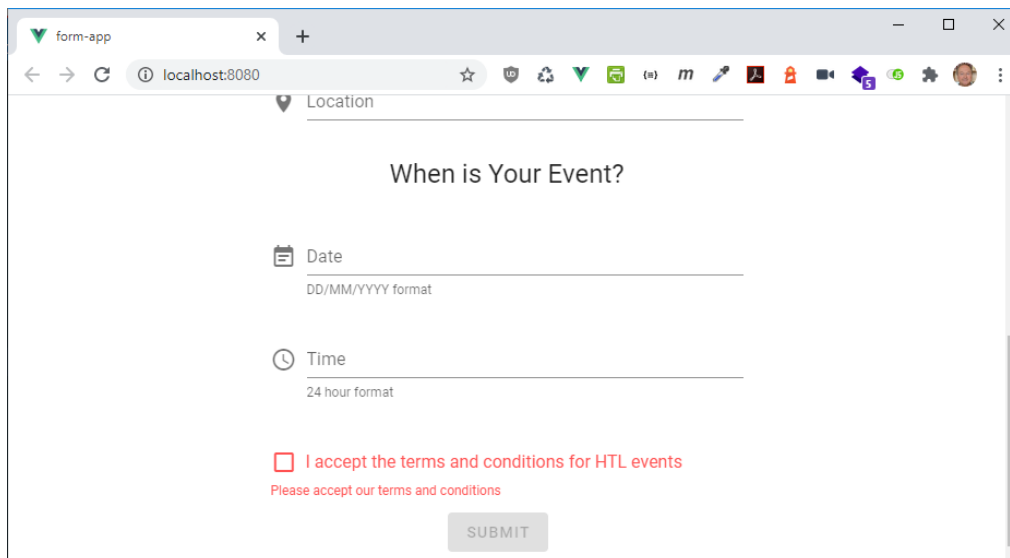
Hinweis: Leider feuert bei der Checkbox nicht der Blur Event. Wenn du auch schon beim select/deselect der Checkbox eine Fehlermeldung bringen möchtest, dann verwende einen Watcher!

```
watch: {
  // ...
  termsAndConditions() {
    this.$v.termsAndConditions.$touch();
  },
},
```

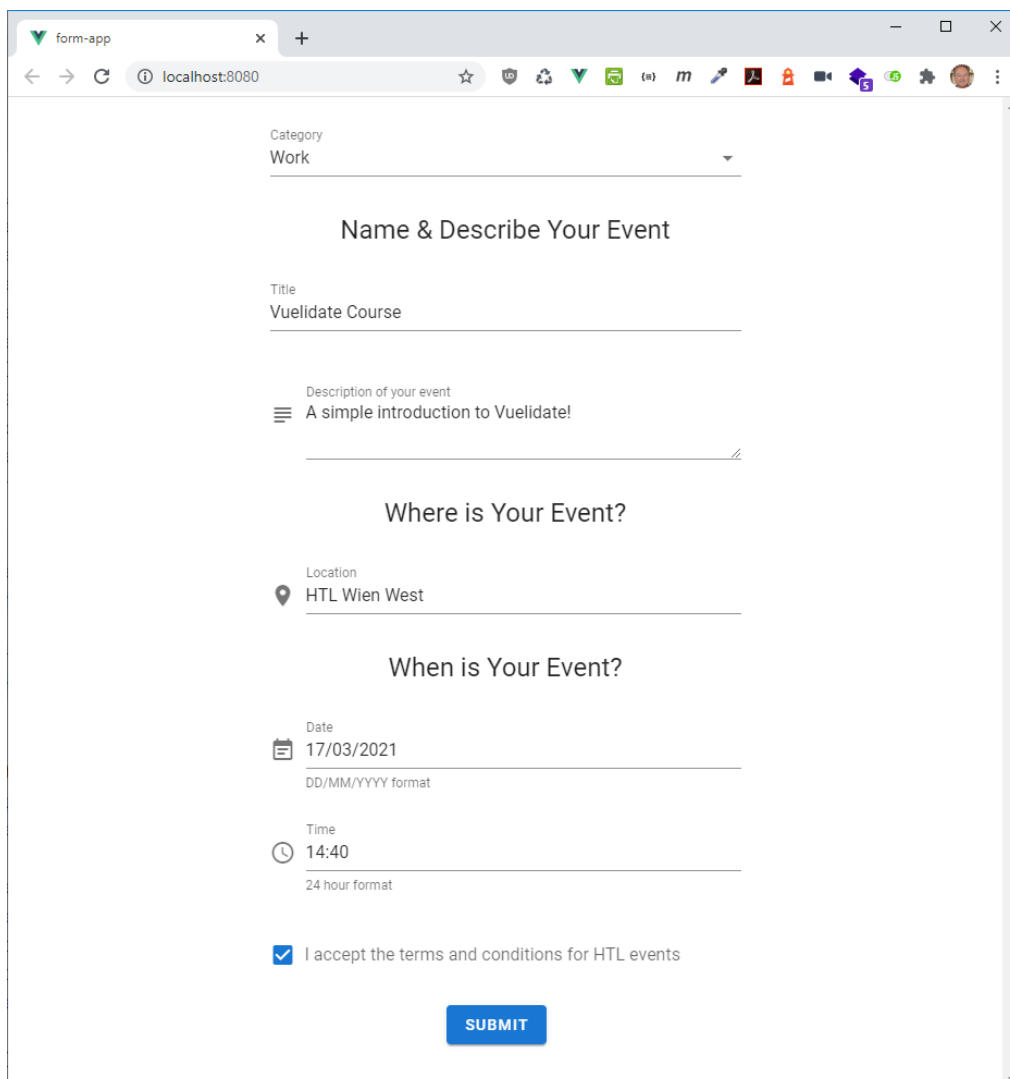
Die Checkfunktion, die prüft ob die Checkbox angehakt ist:

```
const checked = (value) => value === true;
```

Wenn du alles implementiert hast, sollte es so aussehen:



The screenshot shows a web browser window with the address bar at localhost:8080. The page title is 'form-app'. The form is titled 'When is Your Event?'. It contains three input fields: 'Location' (with a location pin icon), 'Date' (with a calendar icon and 'DD/MM/YYYY format' hint), and 'Time' (with a clock icon and '24 hour format' hint). Below these fields is a checkbox labeled 'I accept the terms and conditions for HTL events' with the text 'Please accept our terms and conditions' underneath. At the bottom is a grey 'SUBMIT' button.



The screenshot shows the same web browser window, but the form is now filled out. The 'Category' dropdown is set to 'Work'. The form title is 'Name & Describe Your Event'. The 'Title' field contains 'Vuelidate Course'. The 'Description of your event' field contains 'A simple introduction to Vuelidate!'. The 'Location' field contains 'HTL Wien West'. The 'Date' field contains '17/03/2021' and the 'Time' field contains '14:40'. The checkbox 'I accept the terms and conditions for HTL events' is now checked. The 'SUBMIT' button is now blue.