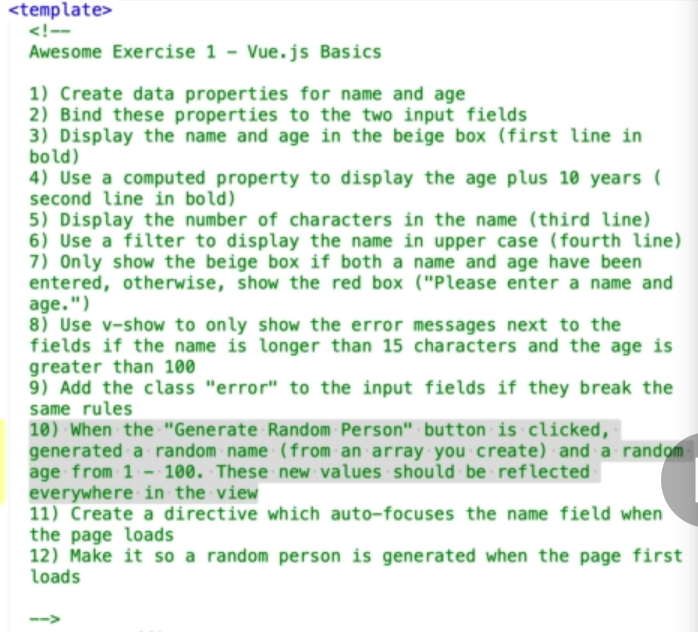
**<https://www.udemy.com/course/quasarframework/>**

**Exercise:**

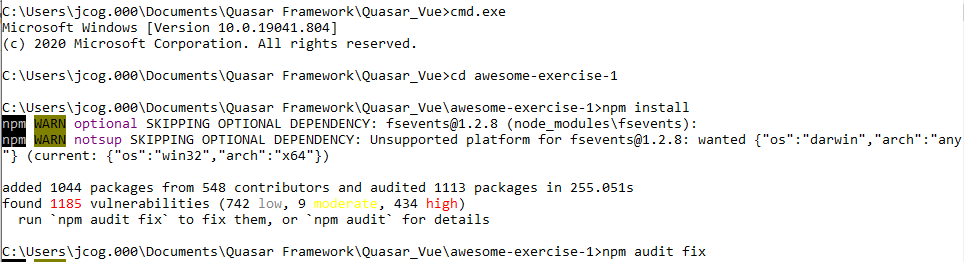
**Assignment:**



**Setup:**

**Open a cmd file and cd into the awesome-exercise-1 folder:**

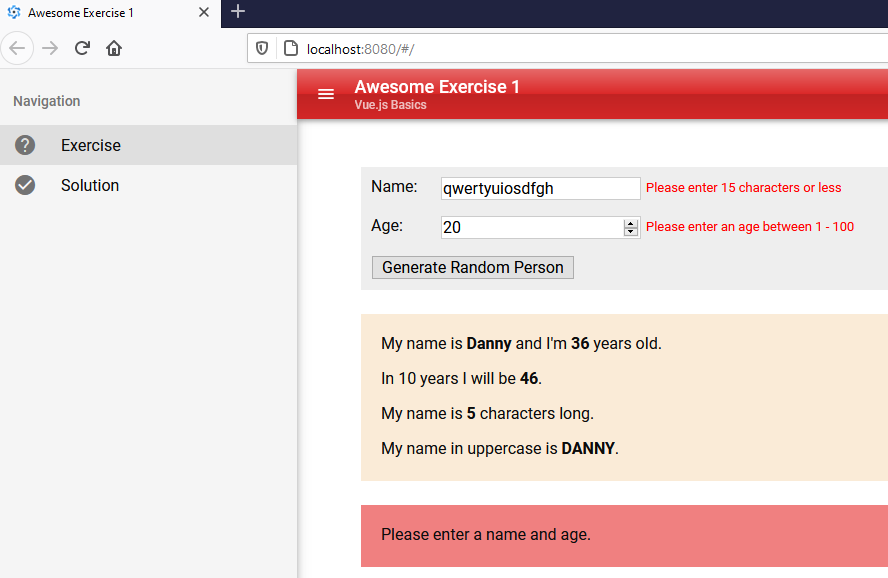
**Then execute npm install, followed by npm audit fix:**



**Then execute quasar dev to start up app:**



**Application boots up in Chrome:**



1) Create data properties for name and age



<script>

  export default {

    data() {

      return {

        name: 'joe',

        age: 64,

      }

    }

  }

</script>

2) Bind these properties to the two input fields

<input v-model="name" type="text"/>

<input v-model="age" type="number"/>

3) Display the name and age in the beige box (first line in bold)

 <p>My name is <b>{{name}}</b> and I'm <b>{{age}}</b> years old.</p>

4) Use a computed property to display the age plus 10 years (second line in bold)

 },

    computed: {

      addTenToAge() {

// using Number(this.age) forces addition with the

// increment/decrement on the input

        let newAge = Number(this.age) + 10;

        return newAge;

      },

<p>In 10 years I will be <b>{{this.addTenToAge}}</b>.</p>

5) Display the number of characters in the name (third line)

<p>My name is <b>{{ name.length }}</b> characters long.</p>

6) Use a filter to display the name in upper case (fourth line)

 methods: {

    },

    filters: {

      nameToUppercase(value){

         return value.toUpperCase();

       }

    }

  }

<p>My name in uppercase is <b>{{ name | nameToUppercase }}</b>.</p>

7) Only show the beige box if both a name and age have been entered, otherwise,

show the red box ("Please enter a name and age.")

<div v-if="name.length > 1" class="description q-mb-lg">

      <p>My name is <b>{{name}}</b> and I'm <b>{{age}}</b> years old.</p>

      <p>In 10 years I will be <b>{{this.addTenToAge}}</b>.</p>

      <p>My name is <b>{{ name.length }}</b> characters long.</p>

      <p>My name in uppercase is <b>{{ name | nameToUppercase }}</b>.</p>

 </div>

 <div v-if="name.length < 1" class="no-details" >

     <p >Please enter a name and age.</p>

 </div>

8) Use v-show

to only show the error messages next to the fields if the name is longer than 15 characters and the age is greater than 100

  <label v-show="name.length > 15" class="error">

Please enter 15 characters or less</label>

  <label v-show="age < 2 || age > 99" class="error">

Please enter an age between 1 - 100</label>

9) Add the class "error" to the input fields if they break the same rules

<input v-bind:class="{error : name.length > 15}" v-model="name" type="text"/>

<input v-bind:class="{error : age < 2 || age > 99}" v-model="age" type="number"/>

 10) When the "Generate Random Person" button is clicked, generated a random

name (from an array you create) and a random age from 1 - 100.

These new values should be reflected everywhere in the view

<button @click="generateRandomPerson">Generate Random Person</button>

data() {

      return {

        name: 'joe',

        age: 64,

        randomNames: ['roberto', 'billy', 'daniel'],

      }

    },

    computed: {

      addTenToAge() {

        let newAge = Number(this.age) + 10;

        return newAge;

      },

    },

    methods: {

      generateRandomPerson() {

       this.generateName();

       this.generateAge();

      },

      generateName() {

        return this.name =

this.randomNames[Math.round(Math.random()\*this.randomNames.length)];

      },

      generateAge() {

        return this.age =

Math.floor( 2 + Math.random()\*97);

      }

    },

11) Create a directive which auto-focuses the name field when the page loads

<input v-autofocus

v-bind:class="{error : name.length > 15}"

          v-model="name" type="text"/>

directives: {

       autofocus: {

         inserted(el) {

           el.focus()

         }

       }

     },

  }

</script>

  12) Make it so a random person is generated when the page first loads

mounted() {

      this.$refs.rName = this.generateName();

      this.$refs.rAge = this.generateAge();

    }

  }

</script>

<label>Name:</label>

        <input ref="rName"

<label>Age:</label>

        <input ref="rAge"