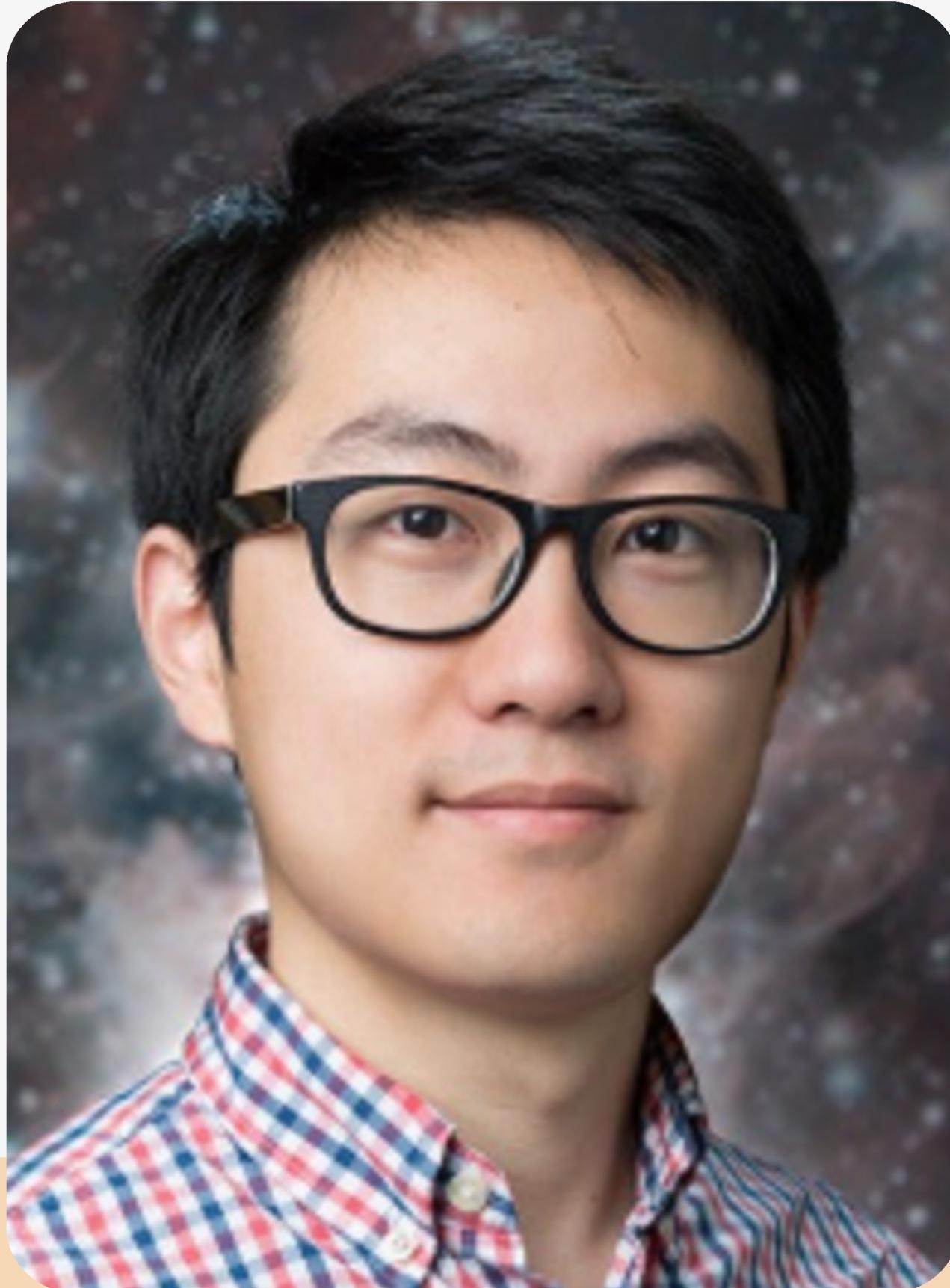


VUEJS

Presentation by Nhóm 12



Evan You

- Là một Chuyên gia Công nghệ Sáng tạo tại Google.
- Nhà phát triển cốt lõi tại Meteor.
- Từ năm 2016 làm việc toàn thời gian cho framework Vue.js.

ABOUT AUTHOR

Tạo một ứng dụng Vue (Vue application instance)

```
Vue.createApp({  
  data : function(){  
    return{  
      }  
    },  
  }).mount('#')
```

Data - Property

```
data() {
  return {
    id = 0,
    message: "Xin chào, đây là một chuỗi dữ liệu!",

    user: {
      name: "Nguyễn Văn A",
      age: 25,
      email: "nguyenvana@example.com",
    },

    students_name: ["An", "Bình", "Cường", "Dũng"],

    students_info: [
      { name: "Nam", age: 18, class: "12A1" },
      { name: "Hoa", age: 17, class: "11B2" },
      { name: "Hùng", age: 19, class: "12C3" },
      { name: "Linh", age: 18, class: "12A1" }
    ],
  };
},
```

Interpolations - Text

```
<div id="app">
  <h1> {{ message }} </h1>
</div>

<script>
  Vue.createApp({
    data: function () {
      return {
        message: "Welcome to Vuejs"
      }
    },
  }).mount('#app');
</script>
```

Directives - V-bind

```
<div id="app">
  
  <!-- <img src = {{ imageSrc }} alt="Dynamic Vue Image" /> --> sai
</div>

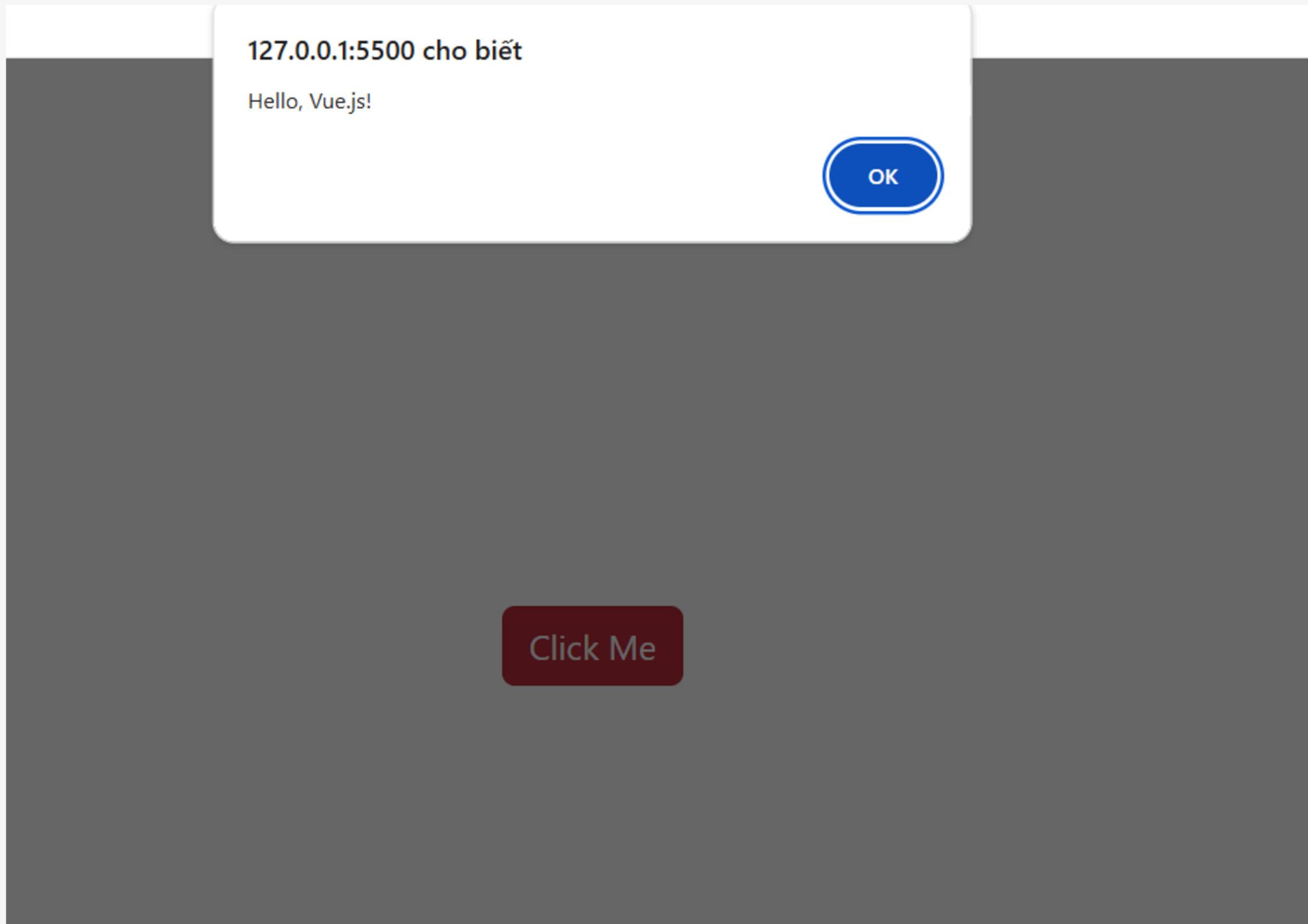
<script>
Vue.createApp({
data() {
  return {
    imageSrc: "https://via.placeholder.com/150"
  };
}
}).mount('#app');
</script>
```

Directives - V-on - Methods

```
<div id="app" class="d-flex justify-content-center align-items-center vh-100">
  <button class = "btn btn-danger" v-on:click="sayHello">Click Me</button>
</div>

<script>
Vue.createApp({
  methods: {
    sayHello() {
      alert("Hello, Vue.js!");
    }
  }
}).mount('#app');
</script>
```

Directives - V-on - Methods

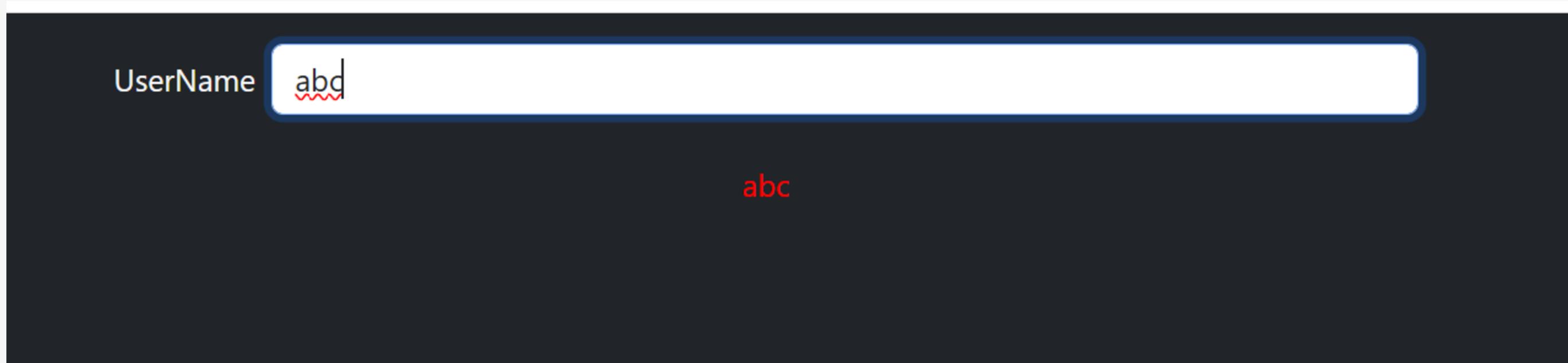


Directives - V-on - Methods

```
<div id="app" class="bg-dark p-3 rounded text-center text-light vh-100">
  <div class="d-flex justify-content-center align-items-center">
    <label class="me-2"> UserName </label>
    <input type="text" class="form-control w-50" v-on:input = "handleInput($event)">
  </div>

  <br>
  <p style = "color:red"> {{inputText}} </p>
</div>
<script>
  Vue.createApp({
    data() {
      return {
        inputText: '',
      };
    },
    methods: {
      handleInput(event) {
        this.inputText = event.target.value;
      },
    },
  }).mount('#app');
</script>
```

Directives - V-on - Methods



Directives - V-model - Methods

```
<div id="app" class="bg-dark p-3 rounded text-center text-light vh-100">
  <div class="d-flex justify-content-center align-items-center">
    <label class="me-2"> UserName </label>
    <input type="text" class="form-control w-50" v-model="inputText">
  </div>

  <br>
  <p style="color:red">{{ inputText }}</p>
</div>

<script>
  Vue.createApp({
    data() {
      return {
        inputText: '',
      };
    },
  }).mount('#app');
</script>
```

Watchers & Computed Properties



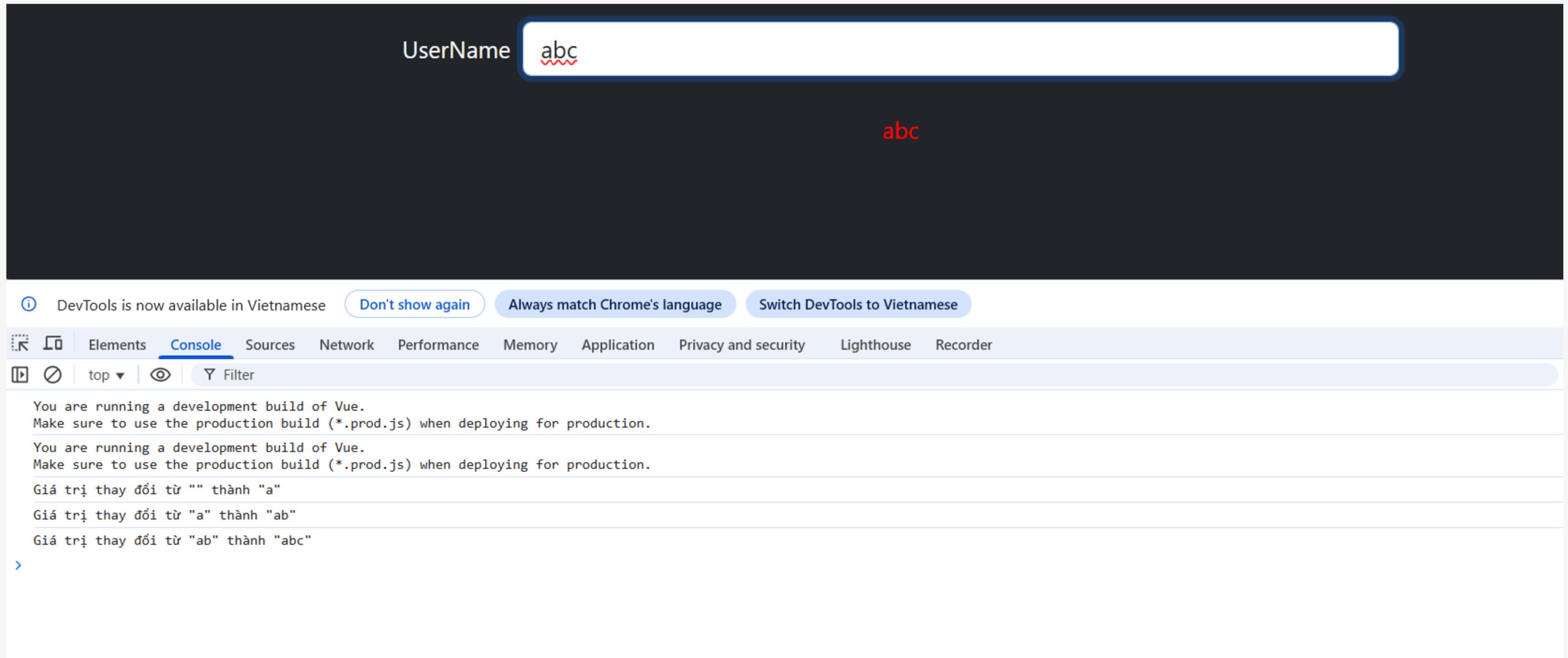
Watchers

```
<div id="app" class="bg-dark p-3 rounded text-center text-light vh-100">
  <div class="d-flex justify-content-center align-items-center">
    <label class="me-2"> UserName </label>
    <input type="text" class="form-control w-50" v-model="inputText">
  </div>

  <br>
  <p style="color:red">{{ inputText }}</p>
</div>

<script>
  Vue.createApp({
    data() {
      return {
        inputText: '',
      };
    },
    watch: {
      inputText(newValue, oldValue) {
        console.log(`Giá trị thay đổi từ "${oldValue}" thành "${newValue}"`);
        if (newValue.length > 10) {
          alert("Input quá dài! Vui lòng nhập ít hơn 10 ký tự.");
        }
      },
    },
  }).mount('#app');
</script>
```

Watchers



The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. At the top, there is a search bar with the placeholder 'UserName' and a red wavy underline over the word 'abc'. Below the search bar, the word 'abc' is displayed in red text. The main console area contains the following log entries:

- You are running a development build of Vue.
Make sure to use the production build (*.prod.js) when deploying for production.
- You are running a development build of Vue.
Make sure to use the production build (*.prod.js) when deploying for production.
- Giá trị thay đổi từ "" thành "a"
- Giá trị thay đổi từ "a" thành "ab"
- Giá trị thay đổi từ "ab" thành "abc"

At the bottom left of the console area, there is a blue arrow pointing to the right.

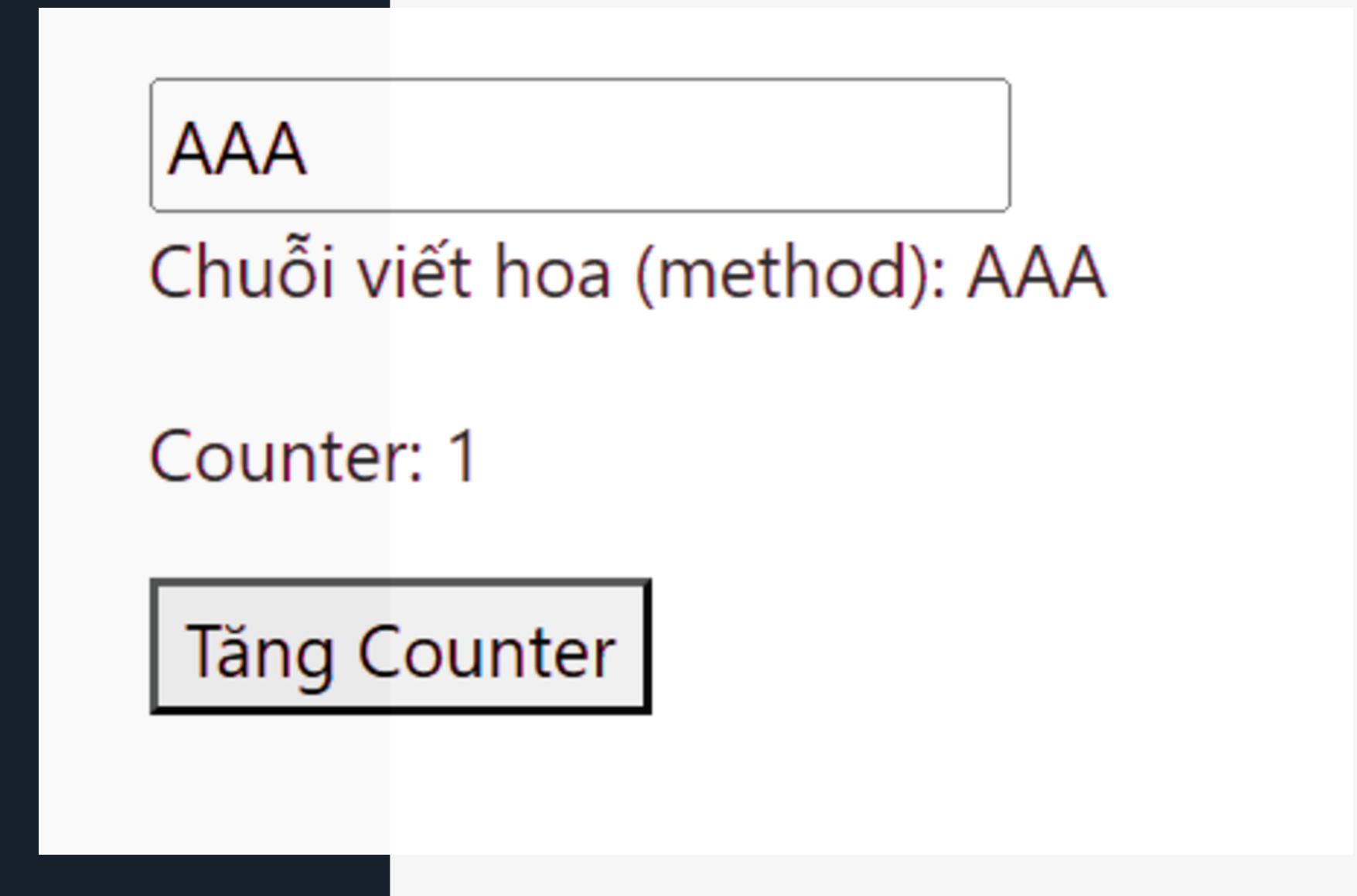
At the very top of the DevTools window, there are several status messages in Vietnamese:

- DevTools is now available in Vietnamese
- [Don't show again](#)
- [Always match Chrome's language](#)
- [Switch DevTools to Vietnamese](#)

Computed Properties

```
<script src="https://unpkg.com/vue@3"></script>
<div id="app" class="m-4">
  <input v-model="message" placeholder="Nhập chuỗi">
  <p>Chuỗi viết hoa (method): {{ getUppercaseMessage() }}</p>
  <p>Counter: {{ counter }}</p>
  <button @click="increaseCounter">Tăng Counter</button>
</div>

<script>
Vue.createApp({
  data() {
    return {
      message: '',
      counter: 0
    };
  },
  methods: {
    getUppercaseMessage() {
      console.log("Method được gọi");
      return this.message.toUpperCase();
    },
    increaseCounter() {
      this.counter++;
    }
  }
}).mount('#app');
</script>
```



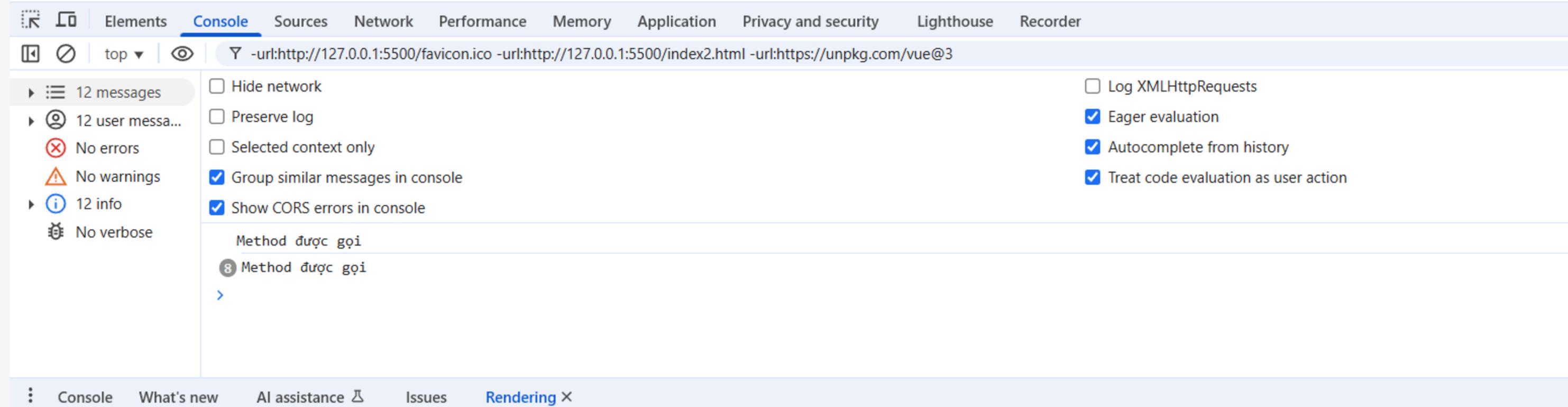
Computed Properties

Nhập chuỗi

Chuỗi viết hoa (method):

Counter: 8

Tăng Counter



The screenshot shows a browser developer tools interface with the 'Console' tab selected. The console output area displays the following content:

```
top
- url: http://127.0.0.1:5500/favicon.ico
- url: http://127.0.0.1:5500/index2.html
- url: https://unpkg.com/vue@3
```

On the left, there is a sidebar with message filtering options:

- 12 messages
- 12 user messages
- No errors
- No warnings
- 12 info
- No verbose

On the right, there are several checkboxes with their current states:

- Hide network
- Preserve log
- Selected context only
- Group similar messages in console
- Show CORS errors in console
- Log XMLHttpRequests
- Eager evaluation
- Autocomplete from history
- Treat code evaluation as user action

Below the checkboxes, two lines of text are shown:

Method được gọi

Method được gọi

At the bottom of the console, there are tabs: 'Console' (selected), 'What's new', 'AI assistance', 'Issues', and 'Rendering' (highlighted).

Computed Properties

```
<script src="https://unpkg.com/vue@3"></script>
<div id="app" class="m-4">
  <input v-model="message" placeholder="Nhập chuỗi">
  <p>Chuỗi viết hoa (computed): {{ uppercaseMessage }}</p>
  <p>Counter: {{ counter }}</p>
  <button @click="increaseCounter">Tăng Counter</button>
</div>

<script>
Vue.createApp({
  data() {
    return {
      message: "",
      counter: 0
    };
  },
  computed: {
    uppercaseMessage() {
      console.log("Computed được gọi");
      return this.message.toUpperCase();
    }
  },
  methods: {
    increaseCounter() {
      this.counter++;
    }
  }
}).mount('#app');
</script>
```

Nhập chuỗi

Chuỗi viết hoa (computed):

Counter: 0

Tăng Counter

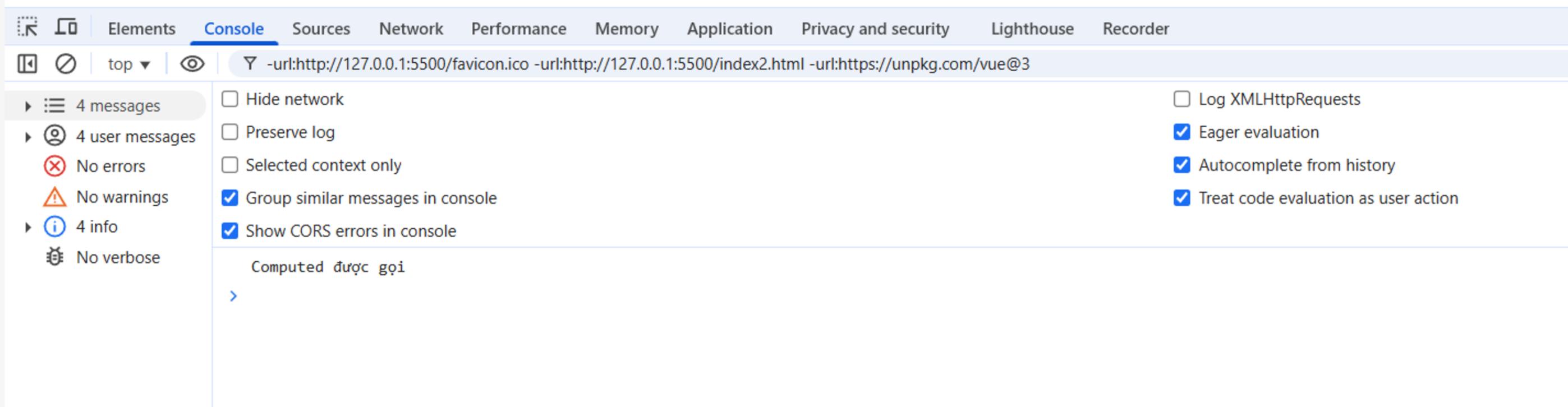
Computed Properties

Nhập chuỗi

Chuỗi viết hoa (computed):

Counter: 10

Tăng Counter



The screenshot shows a browser's developer tools with the 'Console' tab selected. The console output is as follows:

```
top ▾ -url:http://127.0.0.1:5500/favicon.ico -url:http://127.0.0.1:5500/index2.html -url:https://unpkg.com/vue@3
▶ 4 messages
▶ 4 user messages
✖ No errors
⚠ No warnings
▶ 4 info
✖ No verbose
Computed được gọi
>
```

The 'Console' tab has the following configuration options:

- Hide network
- Preserve log
- Selected context only
- Group similar messages in console (checked)
- Show CORS errors in console (checked)
- Log XMLHttpRequests
- Eager evaluation (checked)
- Autocomplete from history (checked)
- Treat code evaluation as user action (checked)

Style Binding

```
<div id="app">
  <div :style="obj">
    Đây là ví dụ style binding với obj.
  </div>
  <button @click="toggleStyle">Thay đổi màu và kích thước</button>
</div>

<script>
  Vue.createApp({
    data() {
      return {
        obj: {
          color: "blue",
          fontSize: "18px",
        },
      };
    },
    methods: {
      toggleStyle() {
        this.obj.color = this.obj.color === "blue" ? "red" : "blue";
        this.obj.fontSize = this.obj.fontSize === "18px" ? "24px" : "18px";
      },
    },
  }).mount("#app");
</script>
<script src="https://unpkg.com/vue@3"></script>
```

Đây là ví dụ style binding với obj.

Thay đổi màu và kích thước

Conditional Rendering - V-if

```
<div id="app" class="m-3">
  <h1>Ví dụ về v-if</h1>
  <button @click="toggleMessage">Bật/Tắt thông điệp</button>
  <p v-if="showMessage">Chào mừng bạn đến với Vue.js!</p>
  <p v-else>Thông điệp đã bị ẩn.</p>
</div>

<script>
  Vue.createApp({
    data() {
      return {
        showMessage: true,
      };
    },
    methods: {
      toggleMessage() {
        this.showMessage = !this.showMessage;
      },
    },
  }).mount("#app");
</script>
```

Ví dụ về v-if

Bật/Tắt thông điệp

Chào mừng bạn đến với Vue.js!

Conditional Rendering - V-show

```
<div id="app" class="m-3">
  <h1>Ví dụ về v-show</h1>
  <button @click="toggleMessage">Bật/Tắt thông điệp</button>
  <p v-show="showMessage">Chào mừng bạn đến với Vue.js!</p>
  <p v-show="!showMessage">Thông điệp đã bị ẩn.</p>
</div>

<script>
  Vue.createApp({
    data() {
      return {
        showMessage: true,
      };
    },
    methods: {
      toggleMessage() {
        this.showMessage = !this.showMessage;
      },
    },
  }).mount("#app");
</script>
```

Ví dụ về v-show

Bật/Tắt thông điệp

Chào mừng bạn đến với Vue.js!

Directives - V-for

```
<div id="app">
  <h1>Danh sách học sinh</h1>
  <ul>
    <li v-for="(student, index) in students" :key="index">
      {{ index + 1 }}. {{ student }}
    </li>
  </ul>
</div>

<script>
  Vue.createApp({
    data() {
      return {
        students: ["An", "Bình", "Cường", "Dũng"],
      };
    },
  }).mount("#app");
```

Danh sách học sinh

- 1. An
- 2. Bình
- 3. Cường
- 4. Dũng

Thank
You

