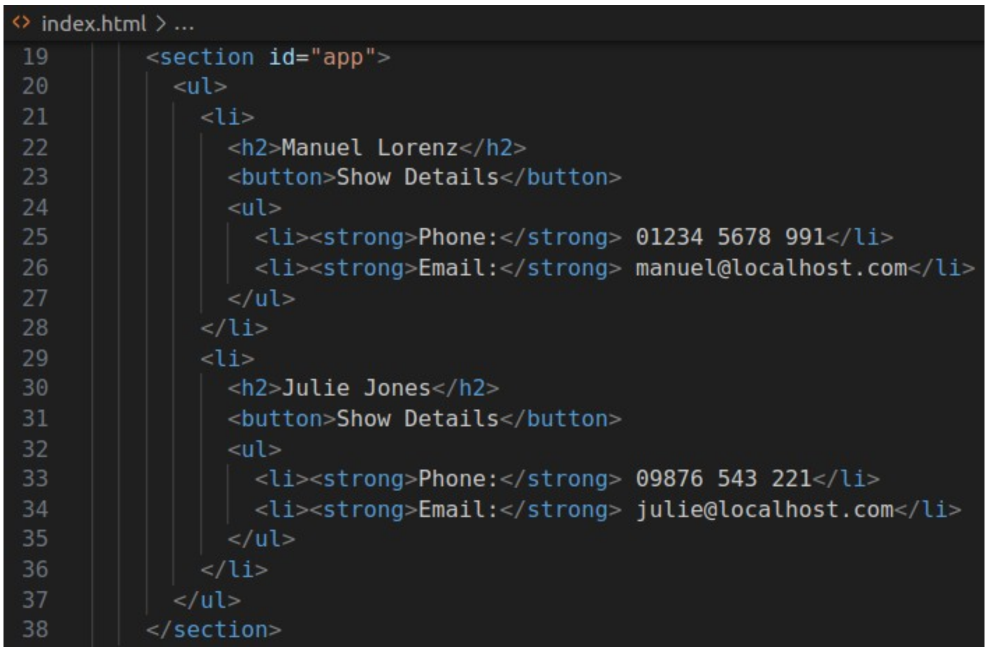
# SECTION 06 - INTRODUCING COMPONENTS

## Index

**- Understanding the problem**

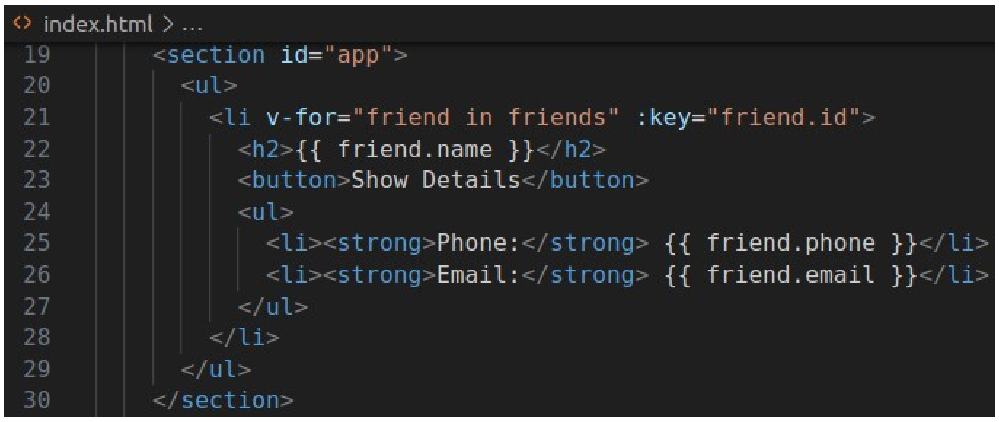
- Starting with this HTML code where the *li* block are repeated:



index.html > ...

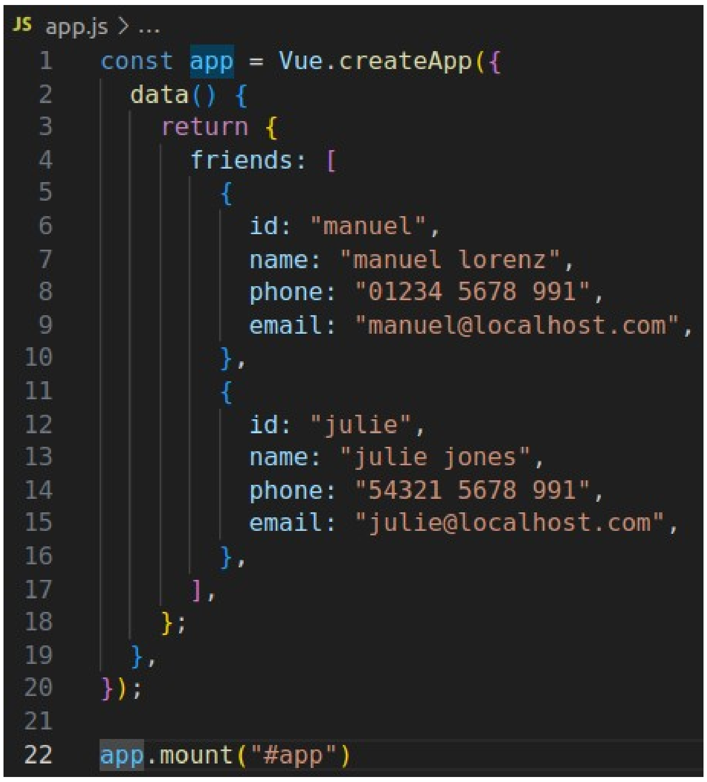
```html
<section id="app">
  <ul>
    <li>
      <h2>Manuel Lorenz</h2>
      <button>Show Details</button>
      <ul>
        <li><strong>Phone:</strong> 01234 5678 991</li>
        <li><strong>Email:</strong> manuel@localhost.com</li>
      </ul>
    </li>
    <li>
      <h2>Julie Jones</h2>
      <button>Show Details</button>
      <ul>
        <li><strong>Phone:</strong> 09876 543 221</li>
        <li><strong>Email:</strong> julie@localhost.com</li>
      </ul>
    </li>
  </ul>
</section>
```

- We can loop an array with that content to avoid duplicated code:



index.html > ...

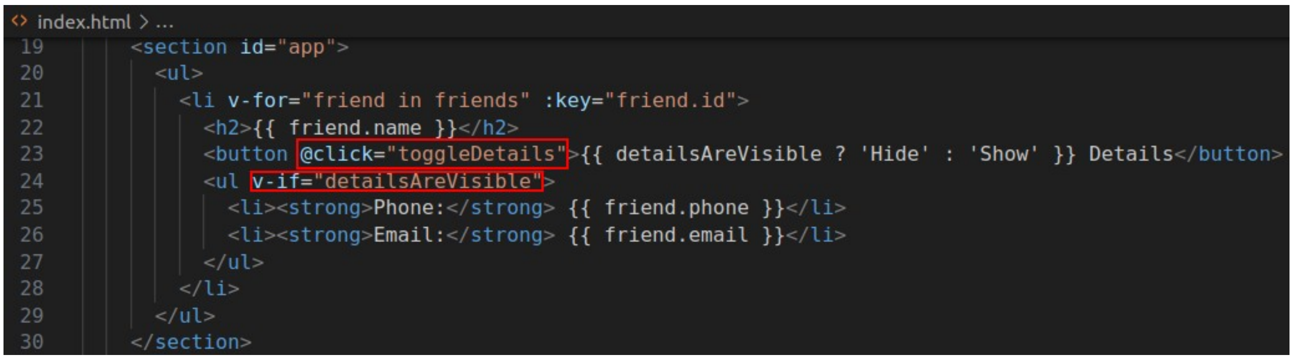
```html
<section id="app">
  <ul>
    <li v-for="friend in friends" :key="friend.id">
      <h2>{{ friend.name }}</h2>
      <button>Show Details</button>
      <ul>
        <li><strong>Phone:</strong> {{ friend.phone }}</li>
        <li><strong>Email:</strong> {{ friend.email }}</li>
      </ul>
    </li>
  </ul>
</section>
```



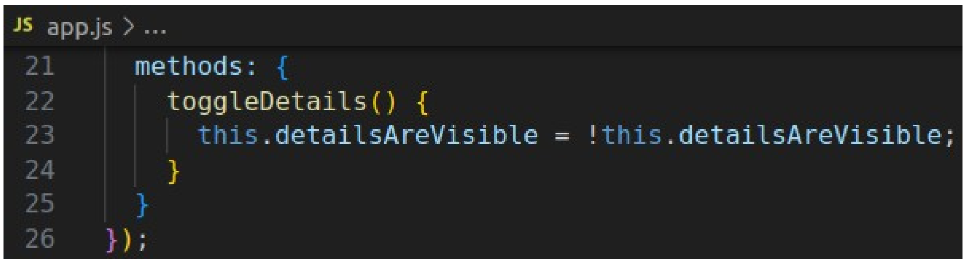
```js
const app = Vue.createApp({
  data() {
    return {
      friends: [
        {
          id: "manuel",
          name: "manuel lorenz",
          phone: "01234 5678 991",
          email: "manuel@localhost.com",
        },
        {
          id: "julie",
          name: "julie jones",
          phone: "54321 5678 991",
          email: "julie@localhost.com",
        },
      ],
    };
  },
});

app.mount("#app")
```

- If we want to add functionality to the *Show Details* button, we do the following:



```html
<section id="app">
  <ul>
    <li v-for="friend in friends" :key="friend.id">
      <h2>{{ friend.name }}</h2>
      <button @click="toggleDetails">{{ detailsAreVisible ? 'Hide' : 'Show' }} Details</button>
      <ul v-if="detailsAreVisible">
        <li><strong>Phone:</strong> {{ friend.phone }}</li>
        <li><strong>Email:</strong> {{ friend.email }}</li>
      </ul>
    </li>
  </ul>
</section>
```



```js
  methods: {
    toggleDetails() {
      this.detailsAreVisible = !this.detailsAreVisible;
    }
  }
});
```

- The issue is that when clicking the button, both parts toggle and not the only one in which I click. This is because the button always points at the same method. Every button for every item gets the same event handler.

- There is work arounds. For example we can pass an id to the method.

- This issue can be solved with components.

**- Introducing components**

- Components are great if you have certain HTML blocks which you reuse in different parts. Also if you have a certain functionality that should be enclosed in that HTML block and that should be specific to that HTML block.

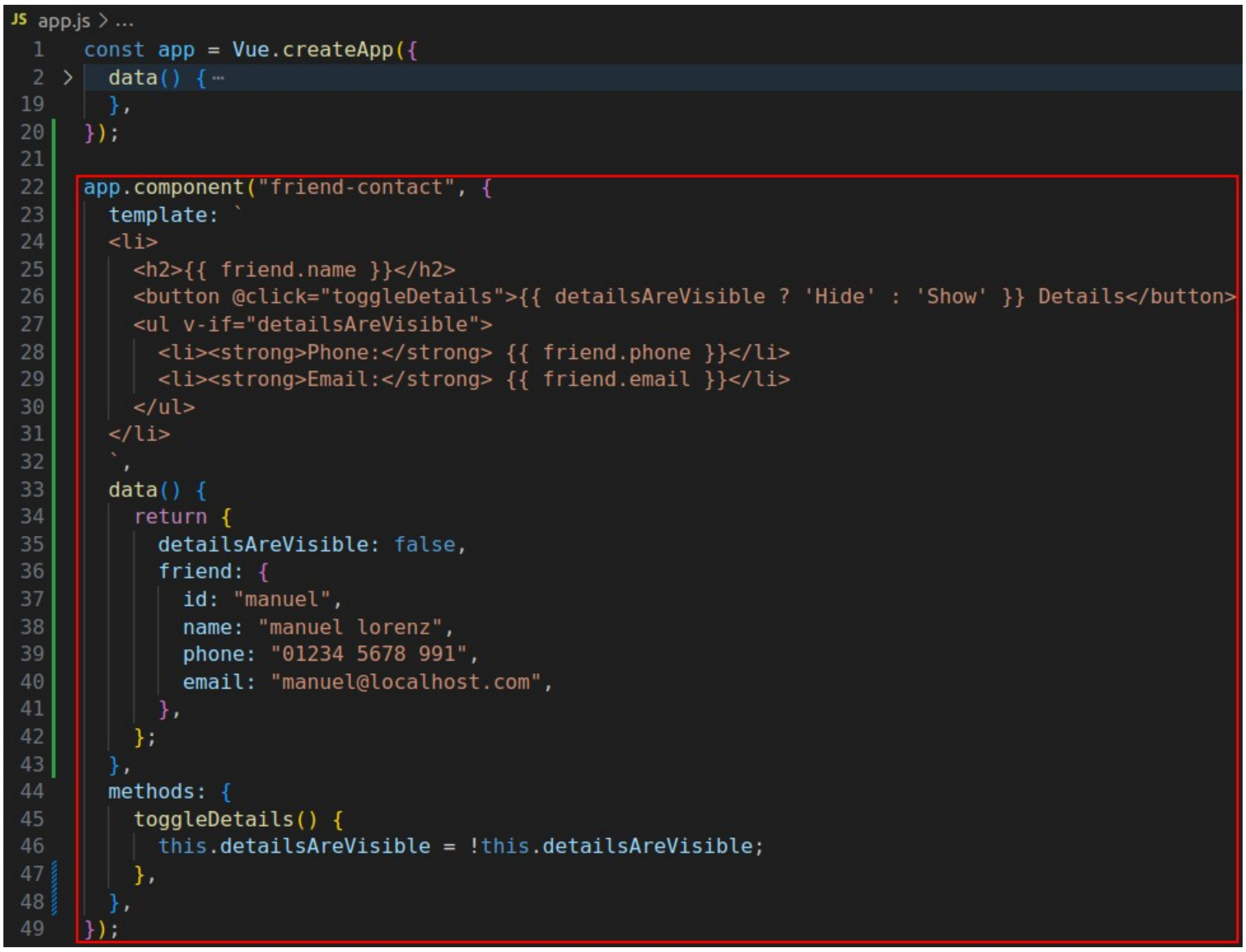
- Components can also be great to splitting your big application into multiple smaller chunks.

- *app.component()* tells Vue that we want to create a component.

- A component is basically like a custom HTML element so you have to define your own HTML tag.

- A Vue component is essentially just another Vue app, just an app that belongs to another app. So it is an app connected to our main app.

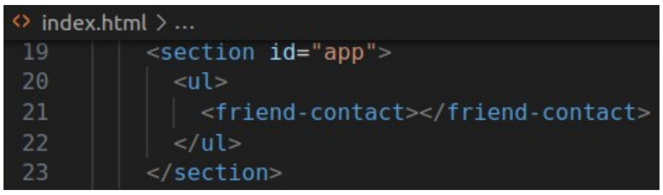
- This is a first approach to components defining a template within it:



app.js > ...

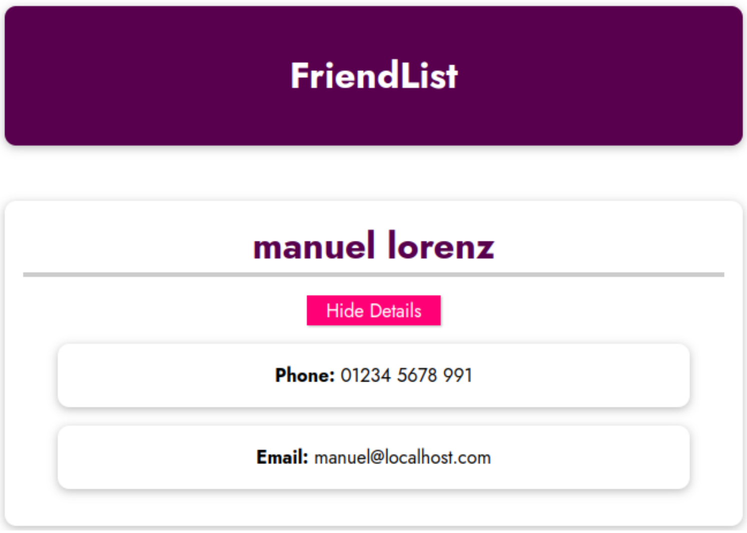
```js
const app = Vue.createApp({
  data() {…
  },
});

app.component("friend-contact", {
  template: `
  <li>
    <h2>{{ friend.name }}</h2>
    <button @click="toggleDetails">{{ detailsAreVisible ? 'Hide' : 'Show' }} Details</button>
    <ul v-if="detailsAreVisible">
      <li><strong>Phone:</strong> {{ friend.phone }}</li>
      <li><strong>Email:</strong> {{ friend.email }}</li>
    </ul>
  </li>
  `,
  data() {
    return {
      detailsAreVisible: false,
      friend: {
        id: "manuel",
        name: "manuel lorenz",
        phone: "01234 5678 991",
        email: "manuel@localhost.com",
      },
    };
  },
  methods: {
    toggleDetails() {
      this.detailsAreVisible = !this.detailsAreVisible;
    },
  },
});
```



index.html > ...

```html
<section id="app">
  <ul>
    <friend-contact></friend-contact>
  </ul>
</section>
```



**- Multiple Vue apps vs multiple components**

- You can use Vue.js to control parts of (possibly multiple HTML) pages OR you use it to build so-called *Single Page Applications* (SPAs).

- If you control multiple, independent parts of HTML pages, you will often work with multiple Vue apps (i.e. you create multiple apps by calling createApp() more than once).

- On the other hand, if you're building a SPA, you typically work with just one *root app* (i.e. createApp() is only used once in your entire codebase) and you build up a user interface with multiple components.

- You absolutely are allowed to also use components in cases where you have multiple Vue apps but you typically won't use multiple Vue apps if you build one big connected user interface. This is because Vue apps are independent from each other, so they can't really communicate with each other. You might find "hacks" to make it work but there's no great "official" way of sharing data between apps. For instance updating something in app A in case something happens in app B, etc.

- Components do offer certain communication mechanisms that allow you to exchange data between them. Hence you can build one connected UI if you work with one root app that holds multiple components.