How to transfer the data from child to parent

and from parent to another child?

What we have : User, UserForm and UserTable

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

Figure 1UserTable

Figure 2User and User Table

1. User and UserTable

When click on add it will show the dialog to UserTable to input the data. Therefore we start from add button:

**<q-btn dense icon="add" color="primary" round class="q-mb-sm" @click="handleAdd" >**

When they click on the add button it will called function : handleAdd

**handleAdd() { this.dialog = true;},** then it will open the UserTable :

<q-dialog

      v-model="dialog"

      persistent

      :maximized="true"

      transition-show="slide-up"

      transition-hide="slide-down"

    >

      <q-card class="bg-white">

        <q-bar>

          <q-space />

          <q-btn

            dense

            flat

            icon="close"

            class="text-white bg-red"

            @click="dialog = false"

            round

          >

            <q-tooltip class="bg-white text-primary">Close (ESC)</q-tooltip>

          </q-btn>

        </q-bar>

**<user-form @batch-import="handleBatchImport" @submit="handleSubmit" />**

      </q-card>

    </q-dialog>

1. UserTable

A screenshot of a computer

AI-generated content may be incorrect.

After they fill the form they will click on the submit button :

**<div class="q-ma-md text-center">**

**<q-btn**

**:color="updateDoc ? 'info' : 'primary'"**

**:label="updateDoc ? 'Update' : 'Submit'"**

**@click="handleSubmit"**

**>**

**<q-tooltip class="bg-white text-primary">(Ctrl + S)</q-tooltip>**

**</q-btn>**

**</div>**

**Explain: <q-tooltip class="bg-white text-primary">(Ctrl + S)</q-tooltip>** is the quasar mouse over then it will show the message

When they click submit will called function : **handleSubmit**

**handleSubmit() {**

**this.$emit("submit", { doc: this.form, source: "manual" });**

**if (this.form.password === this.confirm && this.form.password != null) {**

**if (this.reset) {**

**this.form.reset = this.reset;**

**}**

**}**

**if (this.updateDoc) {**

**// code update**

**}**

**this.$emit("close");**

**},**

**Explain :  this.$emit("submit", { doc: this.form, source: "manual" });**

**-this.$emit :** is the vue that use for transfter the data to parent UserForm: and they emit

object: { doc: this.form, source: "manual" } : one is from that they fill and the source is string manual.

ចឹងយើងបោះទៅនឹងយើងយកវាទៅដាក់នៅកន្លែងណាដែលយើងចង់អោយ វាទទួលទិន្នន័យដោយត្រូវដាក់ Our custom event យើងនឹង **@submit**

* In the User we have dialog to keep the UserTable component :

**<user-form @batch-import="handleBatchImport" @submit="handleSubmit" />**

* **In User Table:** កន្លែងដែលយើងចាប់យក **Data**

**A screenshot of a computer

AI-generated content may be incorrect.**



In Vue.js, props stands for "properties" — they are a mechanism for passing data from a parent component to a child component.

props: {

users: {

type: Array, // The expected data type is an array

default: null, // If no value is passed, it defaults to null

required: true // This prop must be provided by the parent

}

}

Therefore we can get the data from UserForm to the UserTable. To display the data we use v-for :

**<tbody>**

**<tr v-for:="(user, index) in users" :key="index">**

**<td class="text-left">{{ index+1 }}</td>**

**<td class="text-left">{{user.username}}</td>**

**<td class="text-left">{{user.email}}</td>**

**<td class="text-left">{{user.phone}}</td>**

**<td class="text-left">{{user.role}}</td>**

**<td class="text-left">**

**<q-btn**

**color="red"**

**dense**

**icon="delete"**

**round**

**@click="handleDelete(user.\_id)"**

**/>**

**<q-btn**

**color="info"**

**dense**

**icon="edit"**

**class="q-ml-sm"**

**round**

**@click="habndleEdit(user)"**

**/>**

**</td>**

**</tr>**

**</tbody>**