**Documentation**

Example: Crud Application

1. Manage Employee Details

We have three texboxes to enter - the First Name, Last Name and Address. There is an add button, which adds the values entered in the textboxes in a table format with a delete button.

The table format is created using components. The click button interacts with the parent component using the emit event to delete the elemet from the array. The values entered are stored in the array and the same are shared with the child component using the **prop** property.



<html>

<head>

<title>VueJs Instance</title>

<script type = "text/javascript" src = "js/vue.js"></script>

</head>

<body>

<style>

#databinding{

padding: 20px 15px 15px 15px;

margin: 0 0 25px 0;

width: auto;

}

span, option, input {

font-size:20px;

}

.Table{

display: table;

width:80%;

}

.Title{

display: table-caption;

text-align: center;

font-weight: bold;

font-size: larger;

}

.Heading{

display: table-row;

font-weight: bold;

text-align: center;

}

.Row{

display: table-row;

}

.Cell{

display: table-cell;

border: solid;

border-width: thin;

padding-left: 5px;

padding-right: 5px;

width:30%;

}

</style>

<div id = "databinding" style = "">

<h1>Employee Details</h1>

<span>First Name</span>

<input type = "text" placeholder = "Enter First Name" v-model = "fname"/>

<span>Last Name</span>

<input type = "text" placeholder = "Enter Last Name" v-model = "lname"/>

<span>Address</span>

<input type = "text" placeholder = "Enter Address" v-model = "addr"/>

<button v-on:click = "showdata" v-bind:style = "styleobj">Add</button>

<br/>

<br/>

<customercomponent

v-for = "(item, index) in custdet"

v-bind:item = "item"

v-bind:index = "index"

v-bind:itr = "item"

v-bind:key = "item.fname"

v-on:removeelement = "custdet.splice(index, 1)">

</customercomponent>

</div>

<script type = "text/javascript">

Vue.component('customercomponent',{

template : '<div class = "Table"><div class = "Row" v-bind:style = "styleobj"><div class = "Cell"><p>{{itr.fname}}</p></div><div class = "Cell"><p>{{itr.lname}}</p></div><div class = "Cell"><p>{{itr.addr}}</p></div><div class = "Cell"><p><button v-on:click = "$emit(\'removeelement\')">X</button></p></div></div></div>',

props: ['itr', 'index'],

data: function() {

return {

styleobj : {

backgroundColor:this.getcolor(),

fontSize : 20

}

}

},

methods:{

getcolor : function() {

if (this.index % 2) {

return "#FFE633";

} else {

return "#D4CA87";

}

}

}

});

var vm = new Vue({

el: '#databinding',

data: {

fname:'',

lname:'',

addr : '',

custdet:[],

styleobj: {

backgroundColor: '#2196F3!important',

cursor: 'pointer',

padding: '8px 16px',

verticalAlign: 'middle',

}

},

methods :{

showdata : function() {

this.custdet.push({

fname: this.fname,

lname: this.lname,

addr : this.addr

});

this.fname = "";

this.lname = "";

this.addr = "";

}

}

});

</script>

</body>

</html>

1. Manage Currency Converter

We have created a currency converter that converts one value of currency to the selected value of other currency. We have created two dropdowns of currency. When we enter the amount to convert in the textbox, the same is displayed below after conversion. We are using the computed property to do the necessary calculation for currency conversion.

<html>

<head>

<title>VueJs Instance</title>

<script type = "text/javascript" src = "js/vue.js"></script>

</head>

<body>

<style>

#databinding{

padding: 20px 15px 15px 15px;

margin: 0 0 25px 0;

width: auto;

background-color: #e7e7e7;

}

span, option, input {

font-size:25px;

}

</style>

<div id = "databinding" style = "">

<h1>Currency Converter</h1>

<span>Enter Amount:</span><input type = "number" v-model.number = "amount" placeholder = "Enter Amount" /><br/><br/>

<span>Convert From:</span>

<select v-model = "convertfrom" style = "width:300px;font-size:25px;">

<option v-for = "(a, index) in currencyfrom" v-bind:value = "a.name">{{a.desc}}</option>

</select>

<span>Convert To:</span>

<select v-model = "convertto" style = "width:300px;font-size:25px;">

<option v-for = "(a, index) in currencyfrom" v-bind:value = "a.name">{{a.desc}}</option>

</select><br/><br/>

<span> {{amount}} {{convertfrom}} equals {{finalamount}} {{convertto}}</span>

</div>

<script type = "text/javascript">

var vm = new Vue({

el: '#databinding',

data: {

name:'',

currencyfrom : [

{name : "USD", desc:"US Dollar"},

{name:"EUR", desc:"Euro"},

{name:"INR", desc:"Indian Rupee"},

{name:"BHD", desc:"Bahraini Dinar"}

],

convertfrom: "INR",

convertto:"USD",

amount :""

},

computed :{

finalamount:function() {

var to = this.convertto;

var from = this.convertfrom;

var final;

switch(from) {

case "INR":

if (to == "USD") {

final = this.amount \* 0.016;

}

if (to == "EUR") {

final = this.amount \* 0.013;

}

if (to == "INR") {

final = this.amount;

}

if (to == "BHD") {

final = this.amount \* 0.0059;

}

break;

case "USD":

if (to == "INR") {

final = this.amount \* 63.88;

}

if (to == "EUR") {

final = this.amount \* 0.84;

}

if (to == "USD") {

final = this.amount;

}

if (to == "BHD") {

final = this.amount \* 0.38;

}

break;

case "EUR":

if (to == "INR") {

final = this.amount \* 76.22;

}

if (to == "USD") {

final = this.amount \* 1.19;

}

if (to == "EUR") {

final = this.amount;

}

if (to == "BHD") {

final = this.amount \* 0.45;

}

break;

case "BHD":

if (to == "INR") {

final = this.amount \*169.44;

}

if (to == "USD") {

final = this.amount \* 2.65;

}

if (to == "EUR") {

final = this.amount \* 2.22;

}

if (to == "BHD") {

final = this.amount;

}

break

}

return final;

}

}

});

</script>

</body>

</html>

