#৫

<div id="root">

<ul>

<li>

<a :class="{ 'active' : activeClass == 'link1'}" :href="link1.url" :title="link1.title"> {{link1.text}} </a>

</li>

<li>

<a :class="{ 'active' : activeClass == 'link2'}" :href="link2.url" :title="link2.title"> {{link2.text}} </a>

//v-bind:atribute অথবা :atribute এভাবে এইচটিএমএল এট্রিভিউট বাইন্ড করা হয়। :atribute টিকে বলে সর্টহেন্ড বাইন্ড

</li>

<br>

<br>

<p v-html="html"></p> //এইচটিএমএল বাইন্ডিং

</ul>

</div>

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

<script type="text/javascript">

var app = new Vue({

el: "#root",

data: {

link1: { text: "Facebook",

title: "Go To Facebook",

url: "http://facebook.com"

},

link2: {text: "Youtube",

title: "Go To Youtubr",

url: "http://youtube.com" },

activeClass: "link1",

html: '<a href="http://www.rimonbd.com"> About Me </a>'

}

});

</script>

#6

<div id="root">

<h1>Skills</h1> <hr>

<ul>

<li v-for="(skill, i) in skills">

{{i+1}}. {{ skill.name }} ({{skill.experience}})

</li>

</ul>

</div>

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

<script type="text/javascript">

var app = new Vue({

el: "#root",

data: {

skills: [

{name: "HTML", experience: 5},

{name: "CSS", experience: 4},

{name: "JS", experience: 4},

{name: "php", experience: 3},

{name: "java", experience: 3},

{name: "nodejs", experience: 1}

]

}

});

</script>

#7: Conditional rendering

<div id="root">

<h1>Skills</h1> <hr>

<ul>

<li v-for="(skill, i) in skills">

<p v-if="skill.experience > 3"> {{i+1}}. {{ skill.name }} ({{skill.experience}}) </p>

</li>

</ul>

</div>

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

<script type="text/javascript">

var app = new Vue({

el: "#root",

data: {

skills: [

{name: "HTML", experience: 5},

{name: "CSS", experience: 5},

{name: "JS", experience: 4},

{name: "php", experience: 3},

{name: "java", experience: 6},

{name: "nodejs", experience: 2}

]

}

});

</script>

#8: Event and methods

<div id="root">

<h1>Skills</h1> <hr>

<ul>

<li v-for="(skill, i) in skills">

<p>{{skill}} <button @click="removeSkill(i)">x</button></p>

</li>

</ul>

<input type="text" placeholder="Enter skill" v-model="newSkill">

<button v-on:click="addSkill()"> Add </button>

</div>

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

<script type="text/javascript">

var app = new Vue({

el: "#root",

data: {

newSkill: "",

skills: ["HTML", "CSS", "JS"]

},

methods:{

addSkill: function(){

this.skills.push(this.newSkill);

this.newSkill = "";

},

removeSkill: function(i){

this.skills.splice(i, 1);

}

}

});

</script>

#9 Computed Properties

<div id="root">

<h1>All Payments</h1> <hr>

<ul>

<li v-for="(p, i) in payments">

<p> {{p.name}} -> {{p.amount}} </p>

</li>

</ul>

<strong>

Total Amount: {{ totalAmount }}

</strong>

<!-- <h1>Unpaid Payments</h1> <hr>

<ul>

<li v-for="(p, i) in unPaidPayments">

<p>

{{p.name}} -> {{p.amount}}

</p>

</li>

</ul>

-->

</div>

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

<script type="text/javascript">

var app = new Vue({

el: "#root",

data: {

payments: [

{name: "houseRent", amount: 15000, paid: false },

{name: "gas", amount: 700, paid: true },

{name: "electricity", amount: 1000, paid: false },

{name: "water", amount: 800, paid: true },

{name: "net", amount: 1200, paid: false }

]

},

computed: {

unPaidPayments: function(){

var unpaid = this.payments.filter(function(payment){

return payment.paid;

});

return unpaid;

},

totalAmount: function(){

var total = 0;

for(var i=0; i< this.payments.length; i++){

total += this.payments[i].amount;

}

return total;

}

}

});

</script>

#10: Filter

<div id="root">

{{name | lowercase}} <br>

{{profession | uppercase | lowercase}} <br>

Today is {{today | bdFormat}}

</div>

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

<script type="text/javascript">

Vue.filter("lowercase", function(s){

return s.toLowerCase();

});

Vue.filter("bdFormat", function(s){

return s.getUTCDate() + "-" + (s.getUTCMonth()+1) + "-" + s.getUTCFullYear();

});

var app = new Vue({

el: "#root",

data: {

name: "Mamunur Rashid Rimon",

profession: "Software Engineer",

today: new Date()

},

computed: {

upperCasedName: function(){

return this.name.toUpperCase();

}

},

filters:{

uppercase: function(s){

return s.toUpperCase();

}

}

});

</script>

</body>

#11: Components

<div id="root">

<panel title="This is panel title">

This is panel content1 This is panel content This is panel content This is panel content

</panel>

</div>

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

<script type="text/javascript">

Vue.component("panel", {

template: `

<div class="panel" v-if="show">

<div class="panelTitle">

{{title}}

<div class="close" @click="show = false">x</div>

</div>

<div class="panelContent">

<p>

<slot></slot>

</p>

</div>

</div>

`,

props: ['title', 'content'],

data: function(){

return {

show: true

}

}

});

var app = new Vue({

el: "#root",

data: {

}

});

</script>

</body>

**#12: Component communications and custom events**

<div id="root">

<button @click="showLogin = true">Login</button>

<modal v-if="showLogin" @onsave="showLogin = false" @oncancel="showLogin = false">

<template slot="header">

Login Now

</template>

<template slot="body">

<h1>Enter your name</h1>

<input type="text" name="">

</template>

</modal>

</div>

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

<script type="text/javascript">

Vue.component("modal", {

template: `

<div class="modal">

<div class="modalContent">

<div class="modalHeader">

<slot name="header"></slot>

</div>

<div class="modalBody">

<slot name="body"></slot>

</div>

<div class="modalHeader">

<button @click="$emit('onsave')">Save</button>

<button @click="$emit('oncancel')">Cancel</button>

</div>

</div>

</div>

`,

props: ["modalTitle"],

data: function(){

return {

show: true

}

}

});

var app = new Vue({

el: "#root",

data: {

showLogin: false

}

});

</script>

</body>

## #13 Practical component example

**components.js:**

var Event = new Vue();

Vue.component("tabs", {

template: `

<div class="tab">

<ul>

<li v-for="tab in tabs" :class="{'active': tab.selectedNow}" @click="makeEvent(tab.name)">

{{tab.name}}

</li>

</ul>

<div class="details">

<slot></slot>

</div>

</div>

`,

props: ["activeTab"],

data: function(){

return {

tabs: []

}

},

mounted: function(){

this.tabs = this.$children;

//console.log(this.tabs);

},

methods: {

makeEvent: function(s){

Event.$emit('tabchange', s);

}

}

});

Vue.component("tab", {

template: `<div v-if="selectedNow">

<slot></slot>

</div>

`,

props: ['selected', "name"],

data: function(){

return {

selectedNow: false

}

},

created: function(){

var currentVue = this;

currentVue.selectedNow = currentVue.selected;

Event.$on("tabchange", function(s){

if(s == currentVue.name){

currentVue.selectedNow = true;

} else{

currentVue.selectedNow = false;

}

});

}

});

**index.html:**

<div id="root">

<tabs>

<tab name="Tab1" :selected="true">

Tab1 content... <button>btn1</button>

</tab>

<tab name="Tab2">

Tab2 content...

</tab>

<tab name="Tab3">

Tab3 content...

</tab>

</tabs>

<div class="tab">

<ul>

<li class="active">

Tab1

</li>

<li class="">

Tab2

</li>

<li class="">

Tab3

</li>

</ul>

<div class="details">

<div> Tab1 content... <button>btn1</button></div>

<div></div>

<div></div>

</div>

</div>

</div>

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

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

<script type="text/javascript">

var app = new Vue({

el: "#root",

data: {

activeTabName: "Tab1"

}

});

</script>

</body>

</html>

## #14: Transition Animation

<div id="root">

<button @click="show = !show">Toggle</button>

<transition name="bar">

<div class="foo" v-show="show"> Hello World </div>

</transition>

<h1>Skills</h1> <hr>

<input type="text" placeholder="Enter skill" v-model="newSkill">

<button v-on:click="addSkill()"> Add </button>

<br><br><br>

<ul>

<transition-group name="bar">

<li v-for="(skill, i) in skills" :key="i">

<p>{{skill}} <button @click="removeSkill(i)">x</button></p>

</li>

</transition-group>

</ul>

</div>

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

<script type="text/javascript">

var app = new Vue({

el: "#root",

data: {

show: false,

newSkill: "",

skills: ["HTML", "CSS", "JS"]

},

methods:{

addSkill: function(){

this.skills.push(this.newSkill);

this.newSkill = "";

},

removeSkill: function(i){

this.skills.splice(i, 1);

}

}

});

</script>

</body>

</html>

## #15 Routing with vue-router

**.htaccess:**

RewriteEngine on

RewriteCond %{REQUEST\_FILENAME} !-d

RewriteCond %{REQUEST\_FILENAME} !-f

RewriteRule . index.html [L]

**index.html:**

<div id="root">

<ul class="menu">

<li> <a href="#home"> Home </a></li>

<li> <a href="#about">About </a> </li>

<li> <a href="#contact"> Contact </a></li>

</ul>

<br>

<ul class="menu">

<li> <router-link to="home"> Home </router-link> </li>

<li> <router-link to="about"> About </router-link> </li>

<li> <router-link to="contact"> Contact </router-link> </li>

</ul>

<div class="main">

<div class="foo">

<router-view>

</router-view>

</div>

</div>

</div>

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

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

<script type="text/javascript">

var Home = {

template: `

<div>

<h1>Welcome </h1>

<p> This is home page content </p>

</div> `

};

var About = {

template: `

<div>

<h1>About </h1>

<p>This is about page content </p>

</div> ` `

};

var Contact = {

template: `

<div>

<h1>Contact </h1>

<p> This is contact page content </p>

</div> `

};

var routes = [

{ path: '/home', component: Home },

{ path: '/about', component: About },

{ path: '/contact', component: Contact }

];

var router = new VueRouter({

routes: routes,

mode: 'history',

base: '/router-view/'

});

var app = new Vue({

router: router

}).$mount('#root');

</script>

</body>