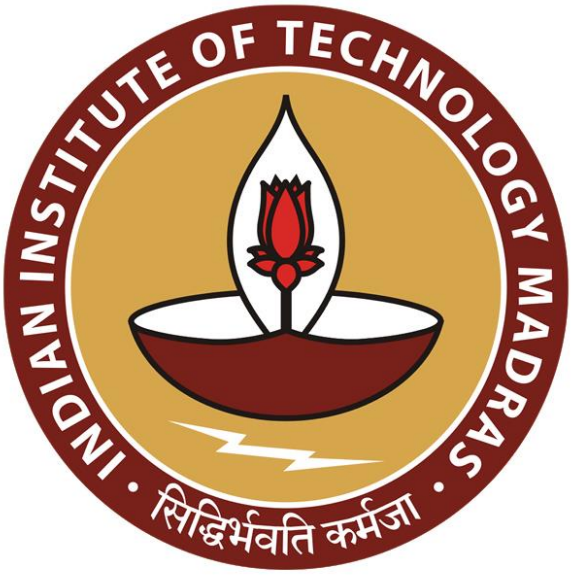
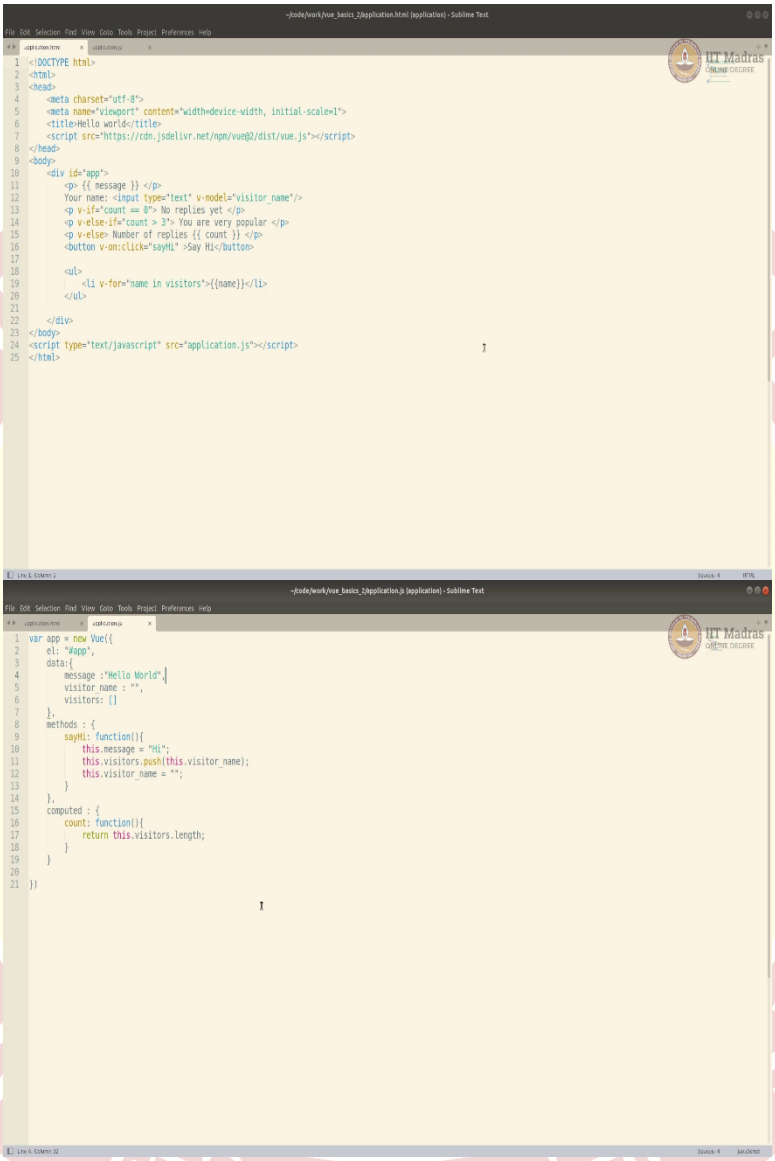


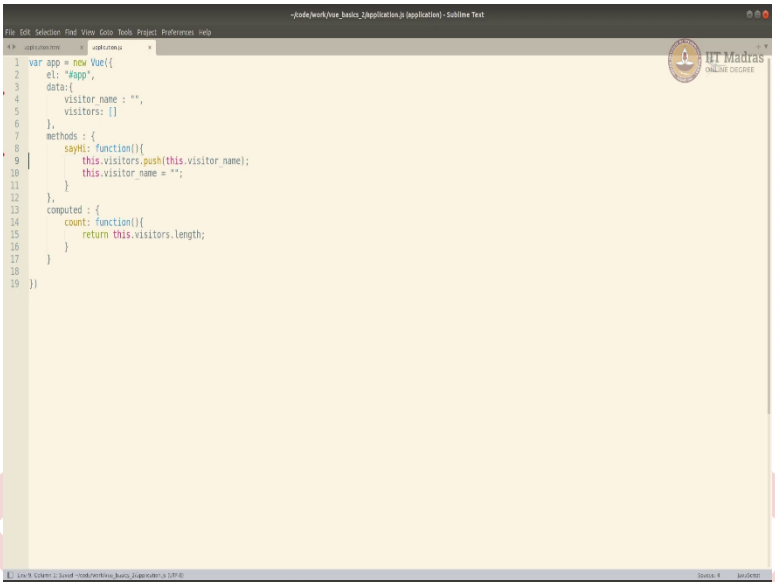
IIT Madras

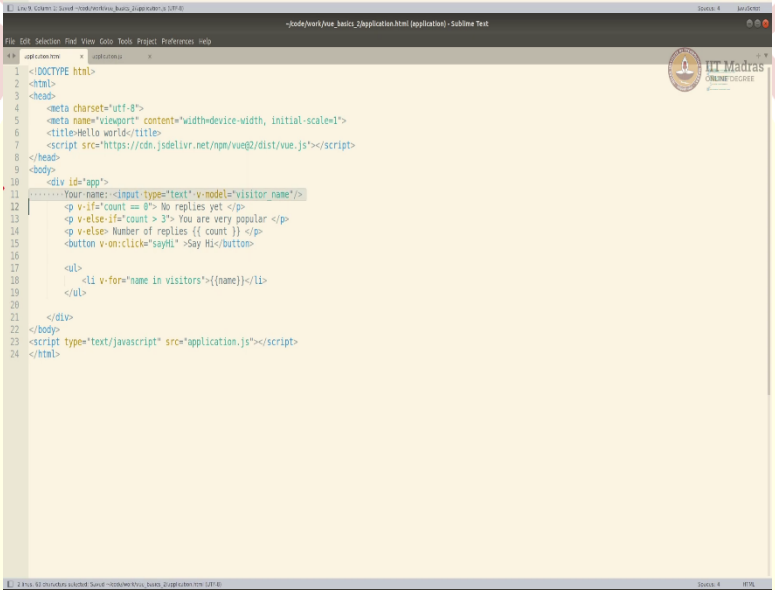
ONLINE DEGREE

Modern Application Development - II
Professor Thejesh G N
Software Consultant
Indian Institute of Technology, Madras
Components in VueJS

(Refer Slide Time: 00:14)



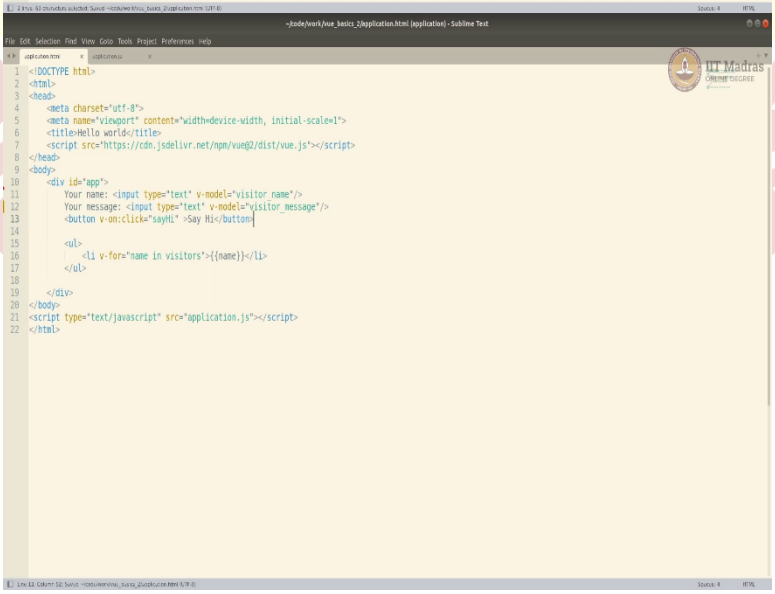

~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
var app = new Vue({
el: "#app",
data:{
visitor_name : "",
visitors: []
},
methods : {
sayHi: function(){
this.visitors.push(this.visitor_name);
this.visitor_name = "";
}
},
computed : {
count: function(){
return this.visitors.length;
}
}
})
IIT Madras
ONLINE DEGREE


~/code/work/vue_basics_2/application.html (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello world</title>
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>

Your name: <input type="text" v-model="visitor_name"/>
 No replies yet
<p v-else-if="count > 3"> You are very popular
<p v-else> Number of replies {{ count }}
<button v-on:click="sayHi" >Say Hi</button>

<li v-for="name in visitors">{{name}}

</body>
<script type="text/javascript" src="application.js"></script>
</html>
IIT Madras
ONLINE DEGREE


~/code/work/vue_basics_2/application.html (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello world</title>
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>

Your name: <input type="text" v-model="visitor_name"/>
Your message: <input type="text" v-model="visitor_message"/>
<button v-on:click="sayHi" >Say Hi</button>

<li v-for="name in visitors">{{name}}

</body>
<script type="text/javascript" src="application.js"></script>
</html>
IIT Madras
ONLINE DEGREE

~/code/work/vue_basics_2/application.js (application) - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

application.html | application.js

```
var app = new Vue({
    el: "#app",
    data:{
        visitor_name : "",
        visitor_message : "",
        visitors: []
    },
    methods : {
        sayHi: function(){
            this.visitors.push(this.visitor_name);
            this.visitor_name = "";
        }
    },
    computed : {
        count: function(){
            return this.visitors.length;
        }
    }

})
```

Line 6, Column 9; Saved ~/code/work/vue_basics_2/application.js (UTF-8) — Spaces: 4 — JavaScript

~/code/work/vue_basics_2/application.js (application) - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

application.html | application.js

```
var app = new Vue({
    el: "#app",
    data:{
        visitor_name : "",
        visitor_message : "",
        messages: []
    },
    methods : {
        sayHi: function(){
            this.messages.push( { "visitor_name": this.visitor_name,  "visitor_message": this.visitor_message });
            this.visitor_name = "";
            this.visitor_message = ""
        }
    },
    computed : {
        count: function(){
            return this.messages.length;
        }
    }

})
```

Line 17, Column 41; Saved ~/code/work/vue_basics_2/application.js (UTF-8) — Spaces: 4 — JavaScript

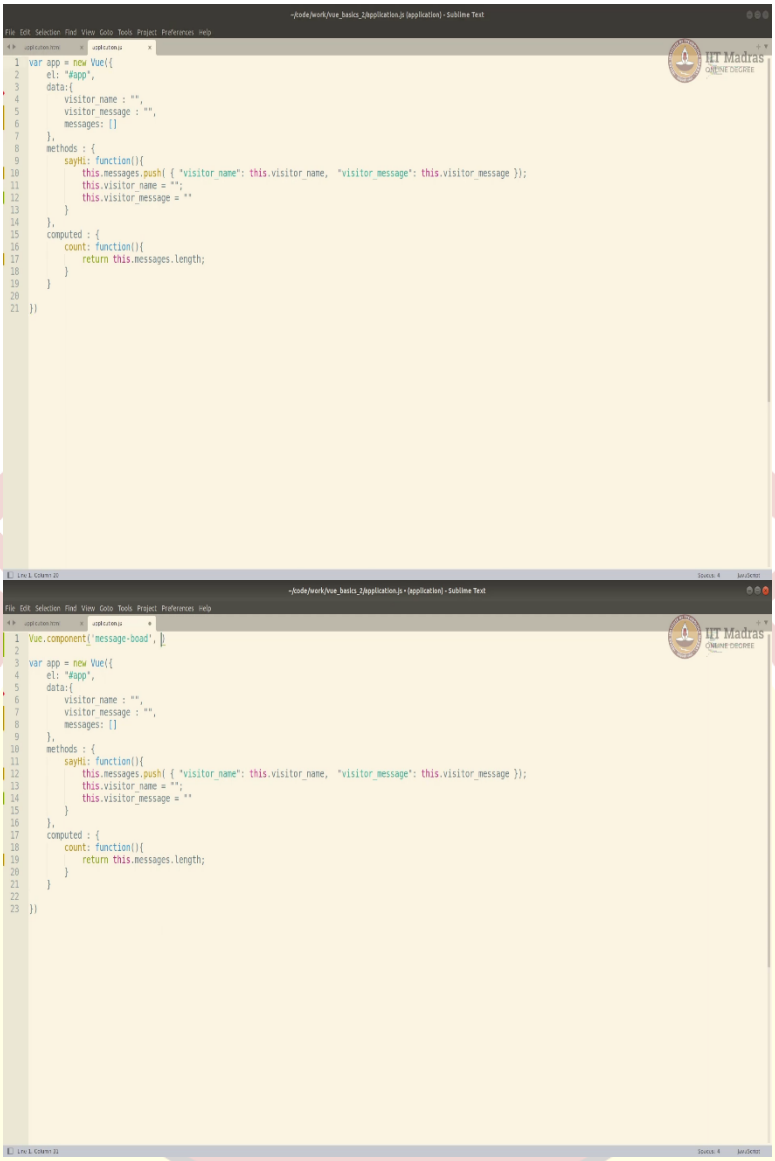
~/code/work/vue_basics_2/application.html (application) - Sublime Text

File Edit Selection Find View Goto Tools Project Preferences Help

application.html | application.js

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello world</title>
    <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>
    <div id="app">
        Your name: <input type="text" v-model="visitor_name"/>
        Your message: <input type="text" v-model="visitor_message"/>
        <button v-on:click="sayHi" >Say Hi</button>

        <ul>
            <li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}</li>
        </ul>

    </div>
</body>
<script type="text/javascript" src="application.js"></script>
</html>
```

Line 17, Column 14; Saved ~/code/work/vue_basics_2/application.html (UTF-8) — Spaces: 4 — HTML



Welcome to the Modern Application Development II screencast. In this screencast, we will continue to experiment with our app, and componentize it, and see, how we can improve it by using various features of Vue. For this, you will need an editor. I am using Sublime Text here. As you can see, you need a browser. I have Firefox here, I have already installed the Vue, add-on, so that we can experiment with it. And, yeah, you can see here.

Let us start. To begin with the we will remember doing this app where every step can leave a name, and say hi, and then it adds up and shows the things. I have continued to use the same application, you can see the code here, this is a HTML part, this is a JS part, and this is same file that I have opened here. And if you can enter your name, say hi, it shows your name here and number of replies. Let us improve this a little bit.

You know, what I will do is, I will remove this message this top message we do not require. And I am going to remove this. So, I am going to add another attribute incoming attribute

called visitor message. Let us say we want to live name and message. And then, let us assume that we do not need to say any of this, we will just say, hi. And the say hi message is there, then visitor name, visitor message.

Let us call these messages, let us store all the messages visitor leaves, along with their name and actual message. So, this total message, we will make it plural, messages will have when the hi called it will store both the name and the message. So, let us call this attributor as visitor name and then no visitor messages, visitor message. This will store this dot visitor message and then we will empty that, we will empty this, we will this. And yeah, let this be there. I think this will not become visitors dot length become messages dot count.

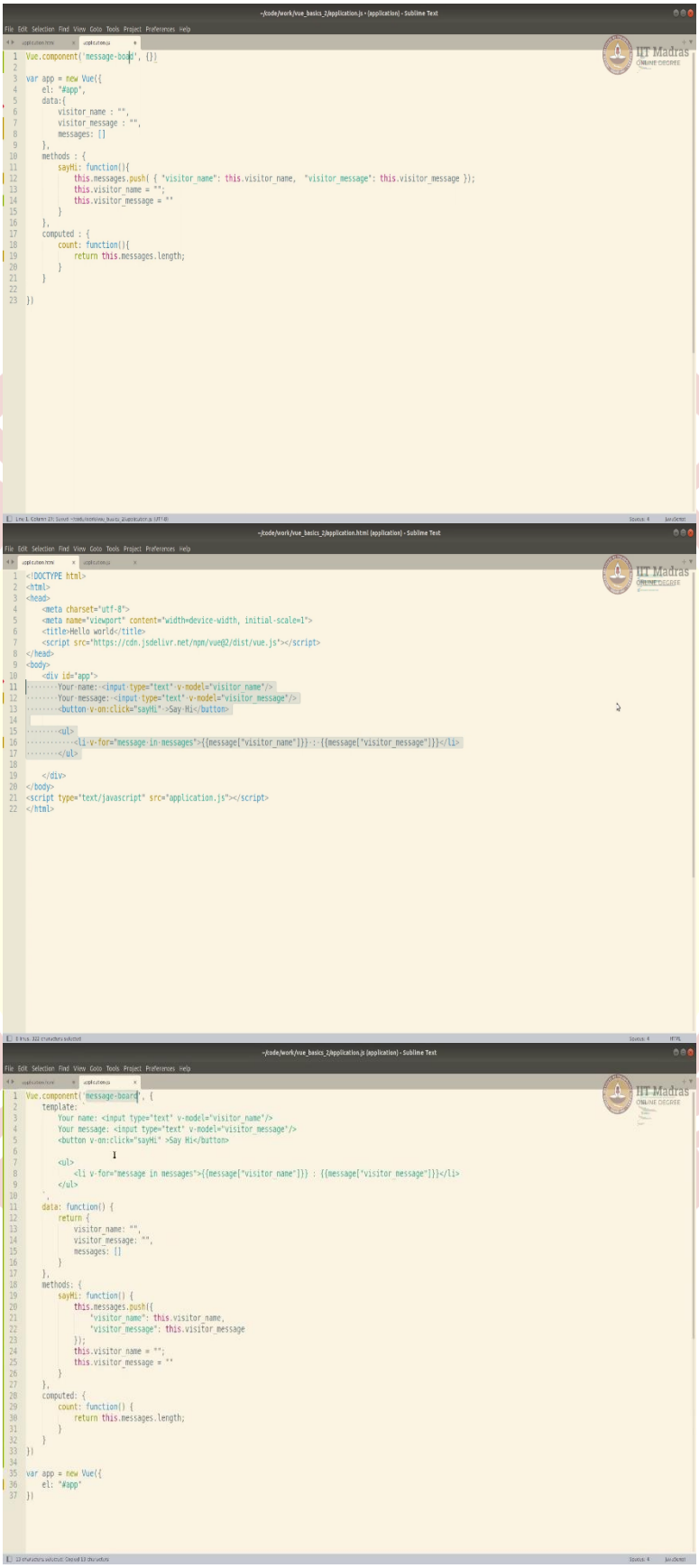
Okay, here let us just check once, messages. I am adding to messages when the hi is called. Okay, here, I think we are iterating through visitors, we will not be iterating through visitors anymore we will be iterating through messenger and each one called a message. And we can print message here, but we will not print the object we will print the this one visitor name and what message he left?

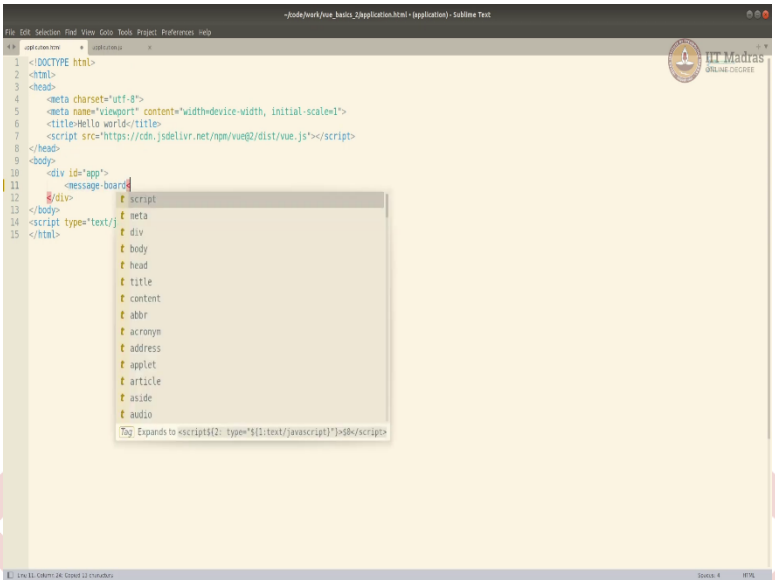
That will be message open bracket, this is just attribute. I think this should work. Let us just check. Reload, so there it is. Your name, let us say Tej welcome, say hi. So, it is showing Tej and welcome.

Now, let us assume. I mean, this is not really like a production level application, but let us assume that there are three folks who are hosting this website, and you want three books or three guest books where folks can leave a message for three folks. For example, there is let us say Tej's, messages board, Raj's messages board and Fatima's message board. And we want the same thing appear thrice, so that all of them can receive like a message here. It is a simple thing. We are not storing anywhere, it is just a display, but we let us say we have three boards, that would make you want to repeat all of this thrice, and write logic for that and all of that. The easier way to do that would be to convert this into a component.

Now, it is very easy to convert the Vue application into a component. We will have to just define a new Vue component. That can be done very easily, by calling Vue dot component. Like the way we have a new Vue, we will call Vue component. Now Vue component actually takes two important attributes. One is a name, what you want to call the name to be, we want to call it message board. Let us called message board and then it takes optional parameters, just like our app.

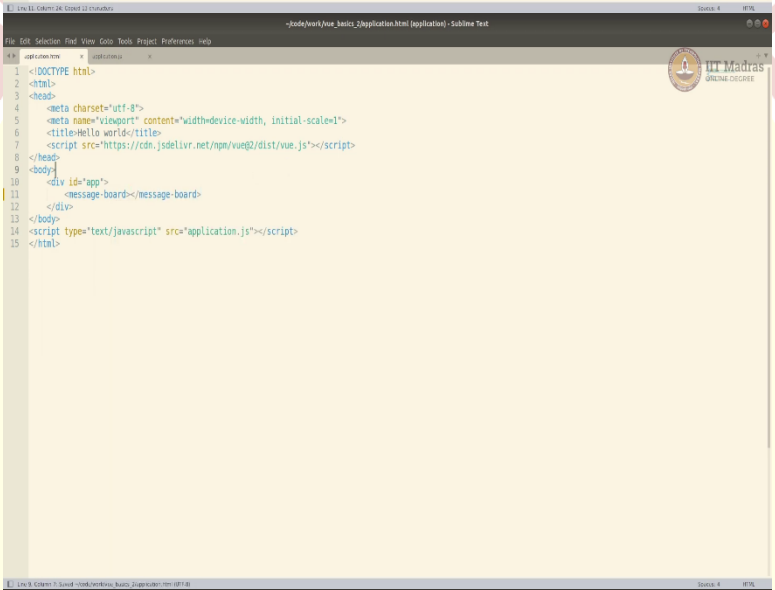
(Refer Slide Time: 05:57)




~/code/work/vue_basics_2/application.html • (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello world</title>
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>

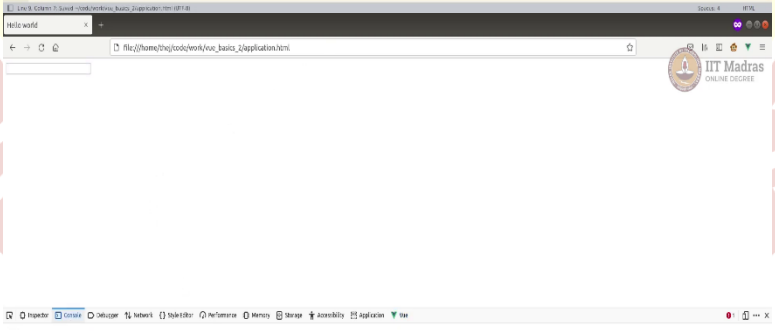
<message-board

</body>
<script type="text/j
</html>
script
meta
div
body
head
title
content
abbr
acronym
address
applet
article
aside
audio
Tag Expands to <script$(2: type="${1:text/javascript}")>$0</script>
IIT Madras
ONLINE DEGREE

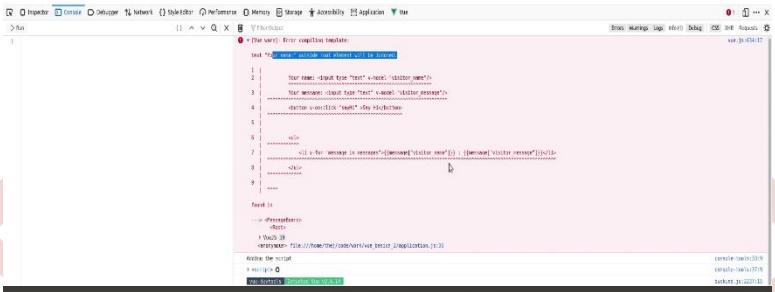

~/code/work/vue_basics_2/application.html (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello world</title>
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>

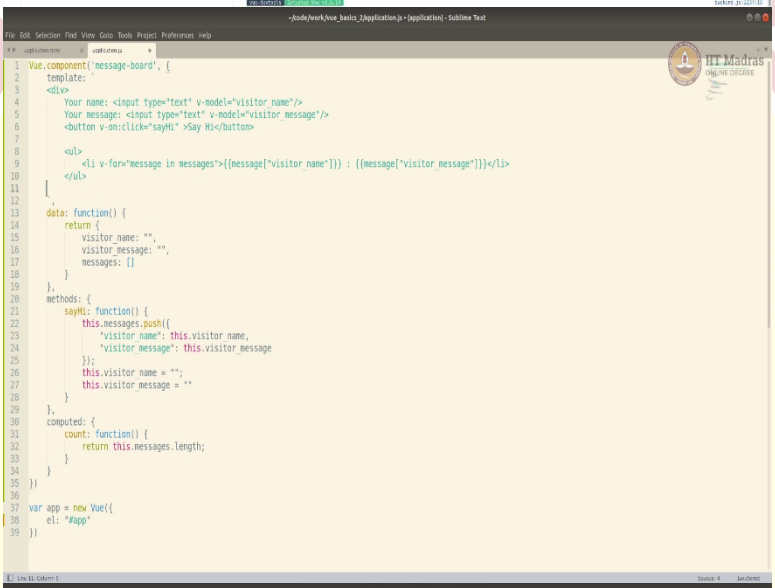
<message-board></message-board>

</body>
<script type="text/javascript" src="application.js"></script>
</html>
IIT Madras
ONLINE DEGREE


Hello world
file:///home/tinaj/code/work/vue_basics_2/application.html
IIT Madras
ONLINE DEGREE


Hello world
file:///home/the/code/work/vue_basics_2/application.html

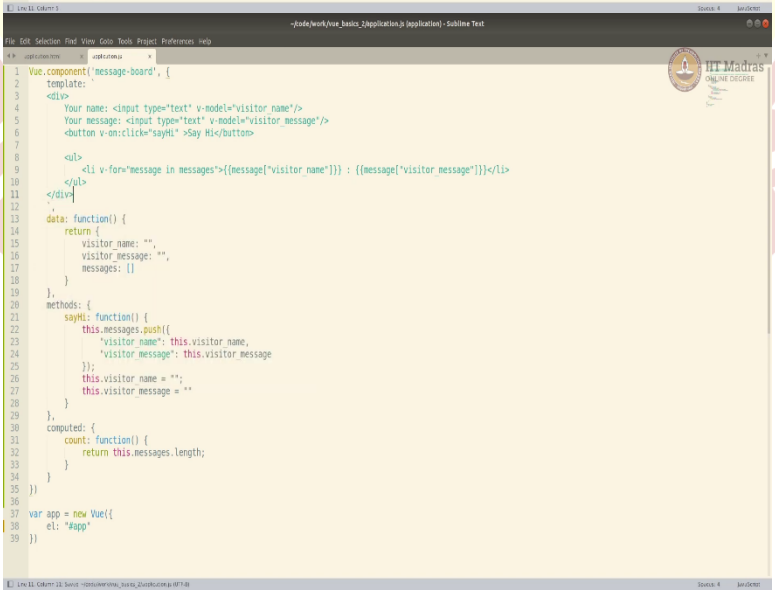



~/code/work/vue_basics_2/application.js • (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
Vue.component('message-board', {
template: `

Your name: <input type='text' v-model="visitor_name"/>
Your message: <input type='text' v-model="visitor_message"/>
<button v-on:click="sayHi" >Say Hi</button>

<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

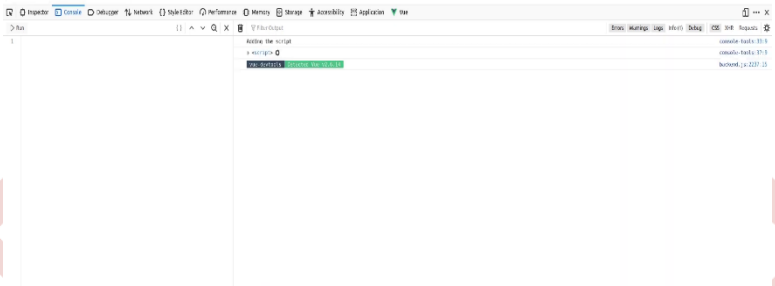
`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app"
})
Line 11, Column 5
Tab Size: 4
JavaScript

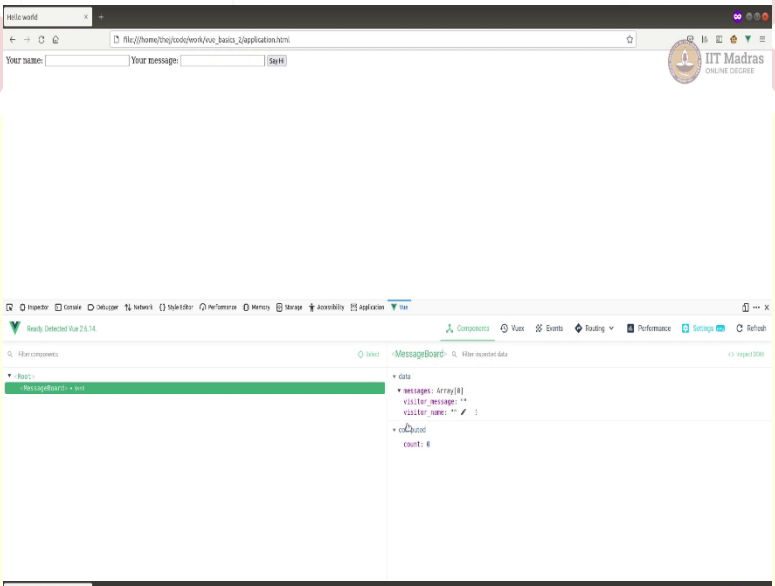

~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
Vue.component('message-board', {
template: `

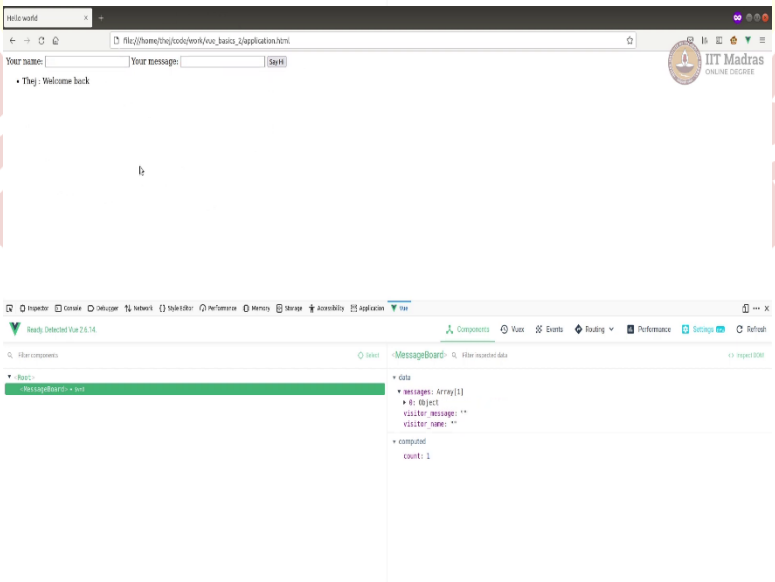
Your name: <input type='text' v-model="visitor_name"/>
Your message: <input type='text' v-model="visitor_message"/>
<button v-on:click="sayHi" >Say Hi</button>

<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app"
})
Line 11, Column 13
Tab Size: 4
JavaScript







Now, Vue component is easy, we can just do it by just define the Vue component by calling Vue dot component, component. It takes two attributes, the first one being the name, and the

second is set of attributes. Let us say what do you want to call it, let us, you want you want to call it message board because we want each, each one to get a message board, like Tej, Supriya and Fatima to get a message board each, so we can call it message board. Board, now it takes input parameters, just like our Vue app. Now, our Vue app is doing almost, you know, single message board tasks, we have to move all of these into our view component.

Let us do that and see what happens. Okay, we are going to copy the same thing, remove this and paste it here. I am going to format it, so everything remains same, but here, we are now going to use data as an attribute, but data will actually become a function, which returns the data, because we want to keep a set of data separate for each component. Also, it will become a function which returns the data, you can return this. Just formatting a bit.

Now, we have converted data into a function this remains same, this remain same, everything else remains same. Now let us see how to use this. This is what we have defined, here, but the only thing that we missed is how we are going to write the content of the component, which is the template, which is actually here, this part.

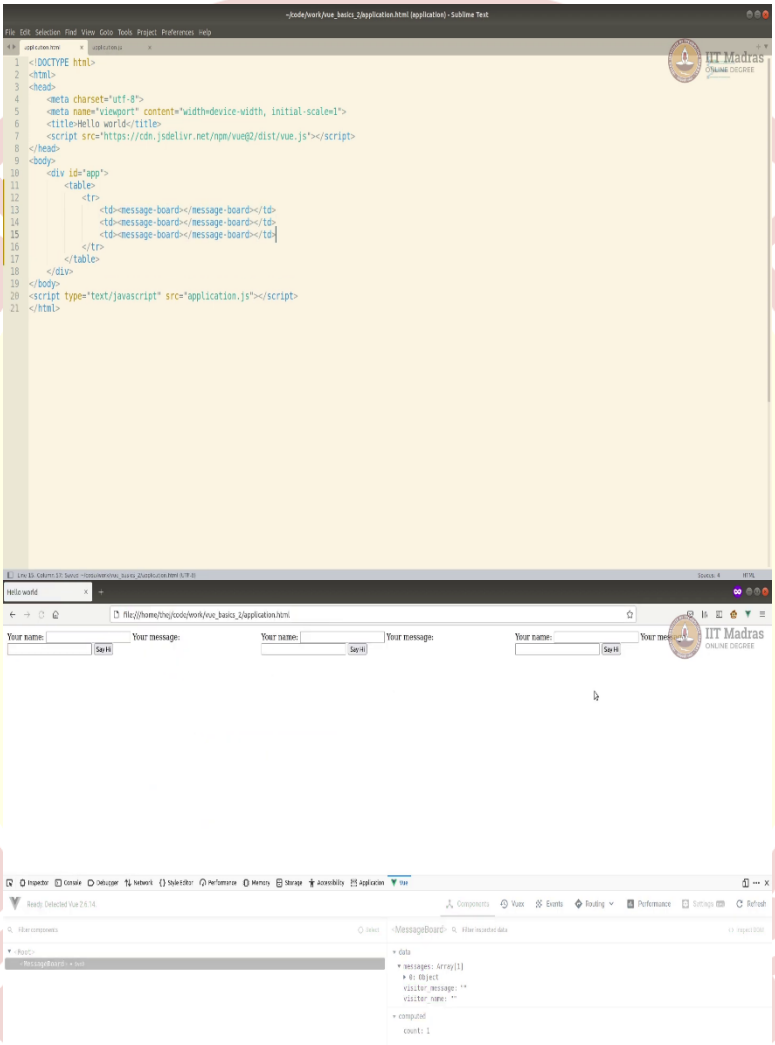
So, let us define a template. Template is defined by define the attribute template, use backtick to put component details, and we could just because each message board will contain these details. So, we are going to take this and put this into like our component because this whole thing is going to become component so the whole thing, we are going to move into a template.

Okay, so we have moved into a template. Now, our whole application you kind of componentized into component. Now we can use this component multiple times, unlike our application, which is only one. So, let us see how to use it, and that's quite easy. That you can do by just calling message. What was the name of the component? Message Board?

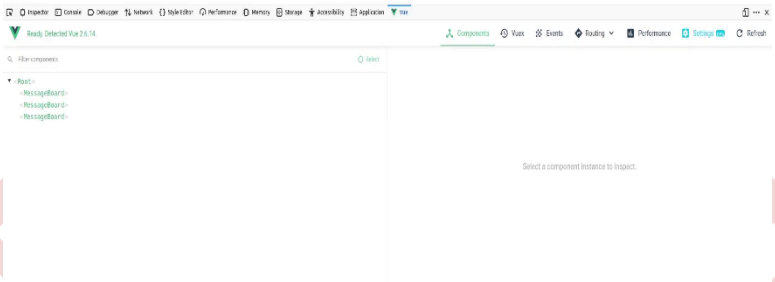
Message board and slash message board. Okay, let us refresh and see. I have kept things on. Okay, there seems to be some error. Let us see what is the error? Okay, so now it is throwing can error can read the error here. Text your name is outside the root element will be ignored. Basically, there is no root element, there are many parallel elements, there is no root element, because a component can have only one root element and that one root element should be there. We do not have a root element here. So, we are going to just add a div as our root element.

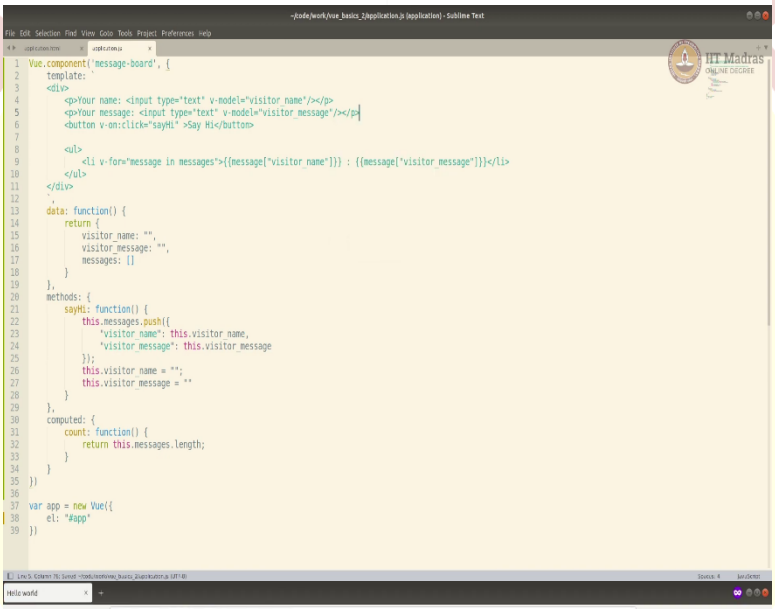
Okay, let us add div as a root element. This should fix the template. Let us refresh this. Okay, error is gone. Now we can see if you actually click on view here and you can see there is root and there is message board. If you click on message board, you can see its attribute and its functions and stuff like that. Now you can give a name, and let us say welcome back, I say, hi, so it shows here.

(Refer Slide Time: 10:57)




Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Your name:
Your message:
Say Hi
IIT Madras
ONLINE DEGREE

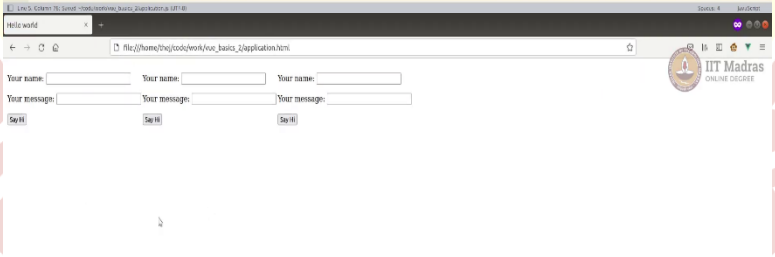

Ready. Detected Vue 2.6.14.
Components
Vuex
Events
Routing
Performance
Settings
Refresh
<Root>
<MessageBoard>
<MessageBoard>
<MessageBoard>
Select a component instance to inspect.

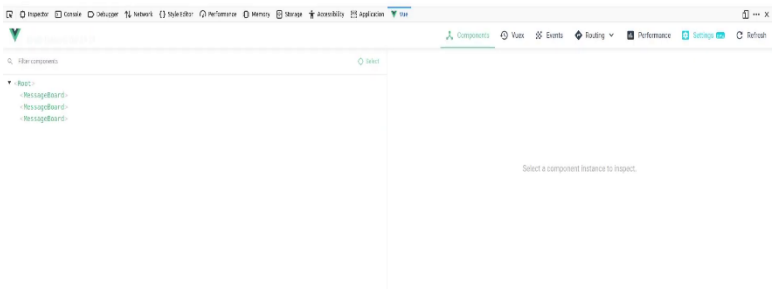

~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
Vue.component('message-board', {
template: `

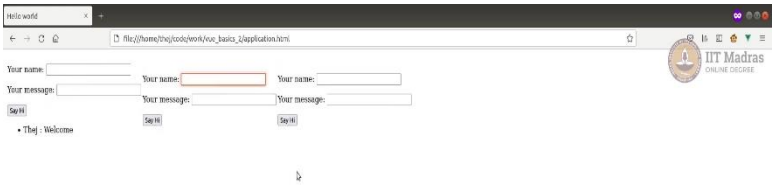
Your name: <input type="text" v-model="visitor_name"/>
Your message: <input type="text" v-model="visitor_message"/>
<button v-on:click="sayHi" >Say Hi</button>

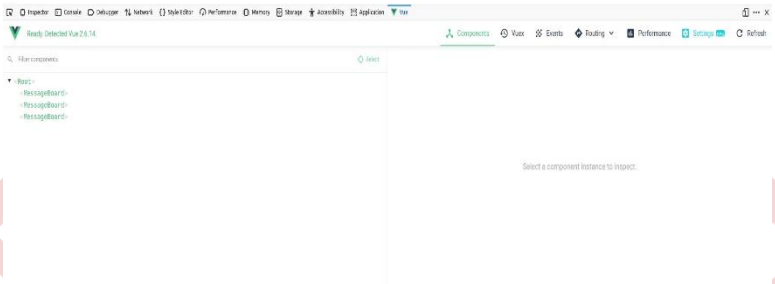
<li v-for="message in messages">{{message["visitor_name"]}} : {{message['visitor message"]}}

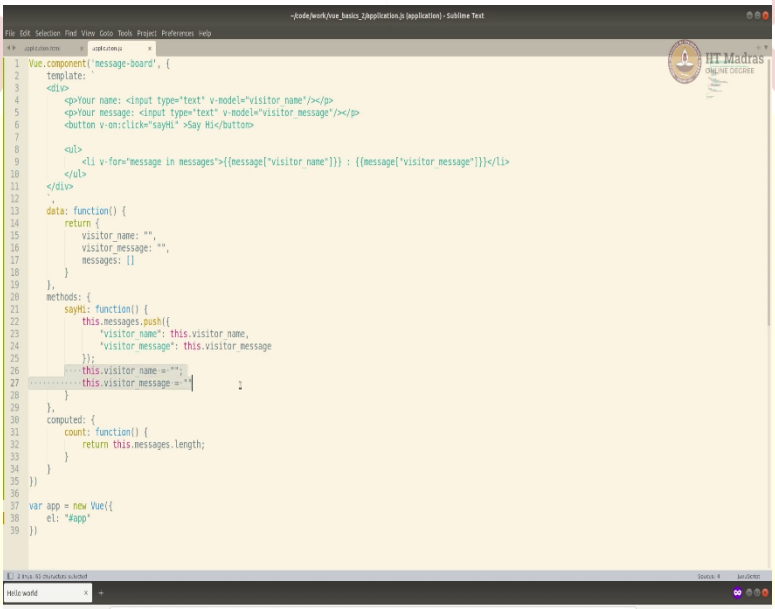
`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app"
})
IIT Madras
ONLINE DEGREE


Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Your name:
Your message:
Say Hi
IIT Madras
ONLINE DEGREE


Components
Vuex
Events
Routing
Performance
Settings
Refresh
<Root>
<MessageBoard>
<MessageBoard>
<MessageBoard>
Select a component instance to inspect.


Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Your name:
Your message:
Say Hi
• Thej : Welcome
Your name:
Your message:
Say Hi
Your name:
Your message:
Say Hi

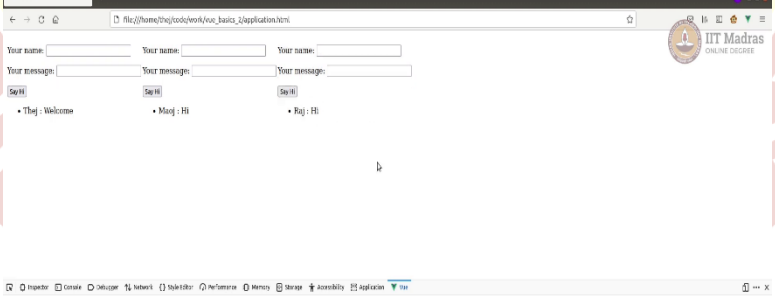

Ready. Detected Vue 2.6.14.
Components
Vuex
Events
Routing
Performance
Settings
Refresh
<Root>
<MessageBoard>
<MessageBoard>
<MessageBoard>
Select a component instance to inspect.

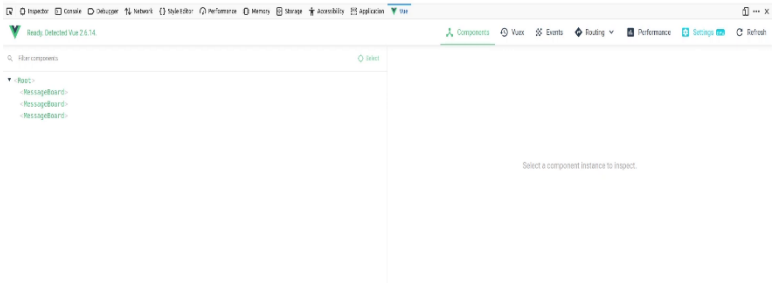

~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
Vue.component('message-board', {
template: `

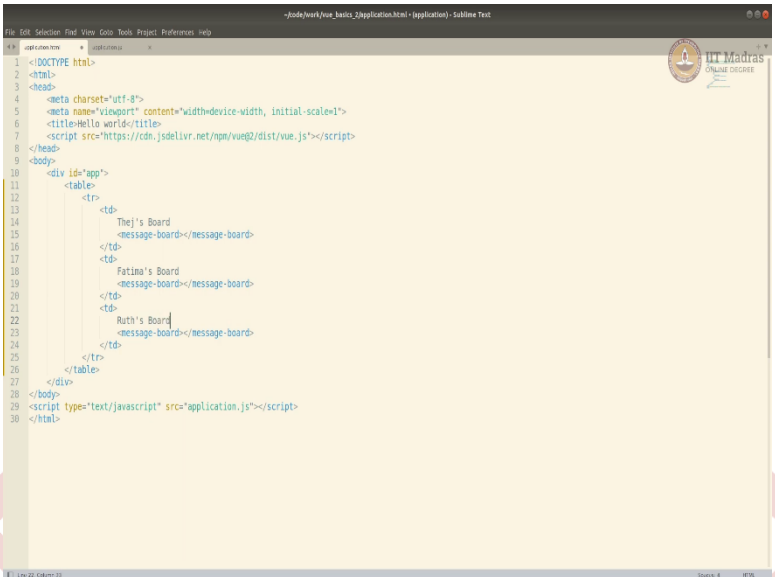
Your name: <input type="text" v-model="visitor_name"/>
Your message: <input type="text" v-model="visitor_message"/>
<button v-on:click="sayHi" >Say Hi</button>

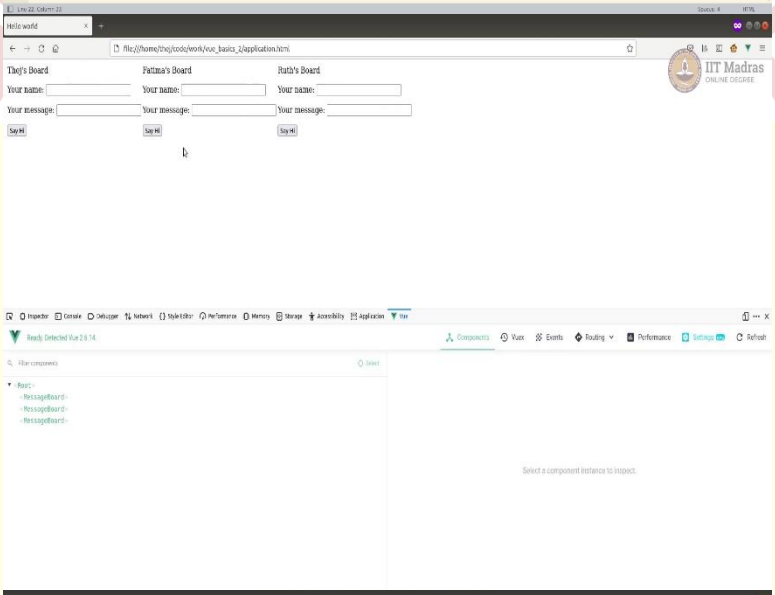
<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor message"]}}

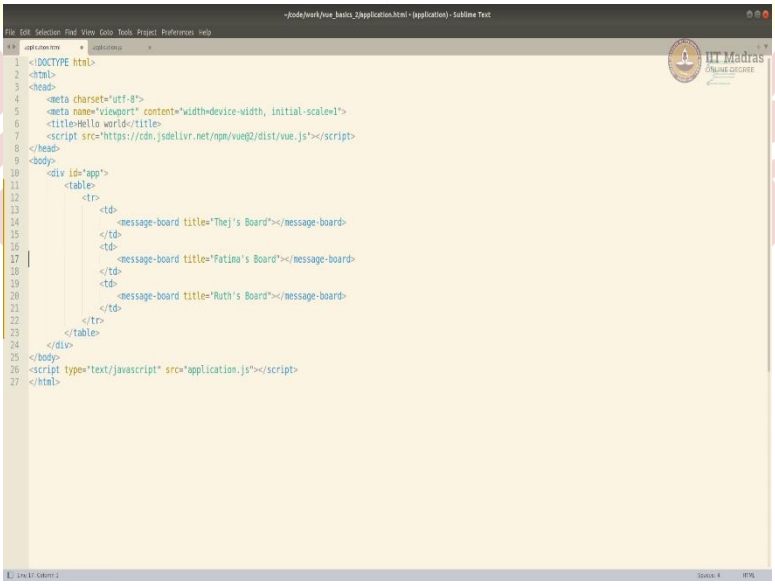
`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app"
})


Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Your name:
Your message:
Say Hi
• Thej : Welcome
Your name:
Your message:
Say Hi
• Manoj : Hi
Your name:
Your message:
Say Hi
• Raj : Hi

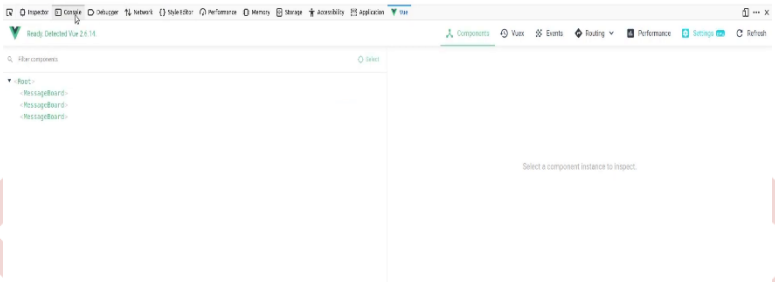

Ready. Detected Vue 2.6.14.
Components
Vuex
Events
Routing
Performance
Settings
Refresh
<Root>
<MessageBoard>
<MessageBoard>
<MessageBoard>
Select a component instance to inspect.

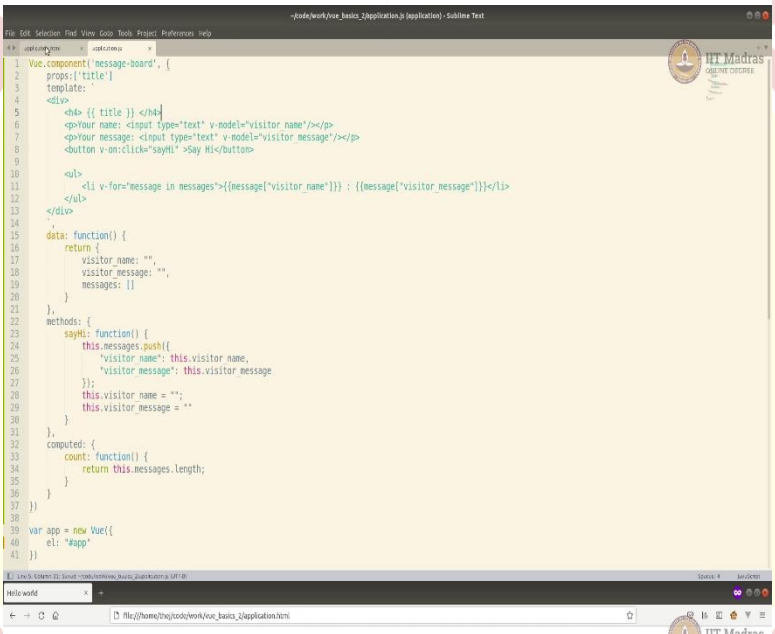

~/code/work/vue_basics_2/application.html • (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
IIT Madras
ONLINE DEGREE
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Hello world</title>
7 <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
8 </head>
9 <body>
10
11 <table>
12 <tr>
13 <td>
14 Thej's Board
15 <message-board></message-board>
16 </td>
17 <td>
18 Fatima's Board
19 <message-board></message-board>
20 </td>
21 <td>
22 Ruth's Board
23 <message-board></message-board>
24 </td>
25 </tr>
26 </table>
27
28 </body>
29 <script type="text/javascript" src="application.js"></script>
30 </html>
Line 22, Column 22
Source: 4
HTML


Hello world
file:///home/thej/code/work/vue_basics_2/application.html
IIT Madras
ONLINE DEGREE
Thej's Board
Your name:
Your message:
Say Hi
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Vue
Ready. Detected Vue 2.6.14.
Components
Vuex
Events
Routing
Performance
Settings
Refresh
Filter components
Select
<Root>
<MessageBoard>
<MessageBoard>
<MessageBoard>
Select a component instance to inspect.


~/code/work/vue_basics_2/application.html • (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
IIT Madras
ONLINE DEGREE
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <title>Hello world</title>
7 <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
8 </head>
9 <body>
10
11 <table>
12 <tr>
13 <td>
14 <message-board title="Thej's Board"></message-board>
15 </td>
16 <td>
17 <message-board title="Fatima's Board"></message-board>
18 </td>
19 <td>
20 <message-board title="Ruth's Board"></message-board>
21 </td>
22 </tr>
23 </table>
24
25 </body>
26 <script type="text/javascript" src="application.js"></script>
27 </html>
Line 17, Column 2
Source: 4
HTML


Hello world
file:///home/tinaj/code/work/vue_basics_2/application.html
Your name:
Your message:
Say Hi


Components
Vuex
Events
Routing
Performance
Settings
Refresh
Select a component instance to inspect.

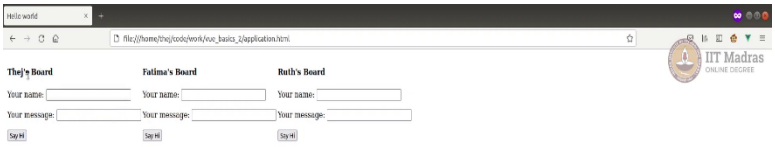

~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
Vue.component('message-board', {
props:['title']
template:
`

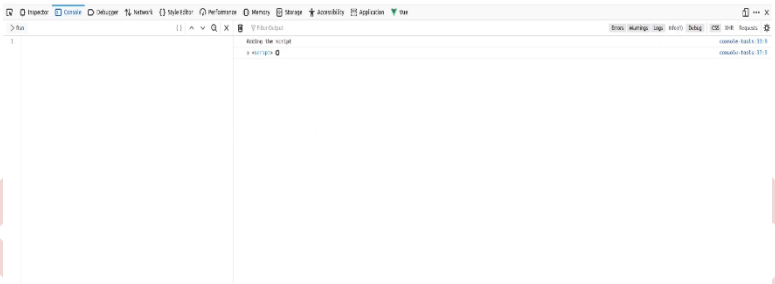
<h4> {{ title }} </h4>
Your name: <input type="text" v-model="visitor_name"/>
Your message: <input type="text" v-model="visitor_message"/>
<button v-on:click="sayHi" >Say Hi</button>

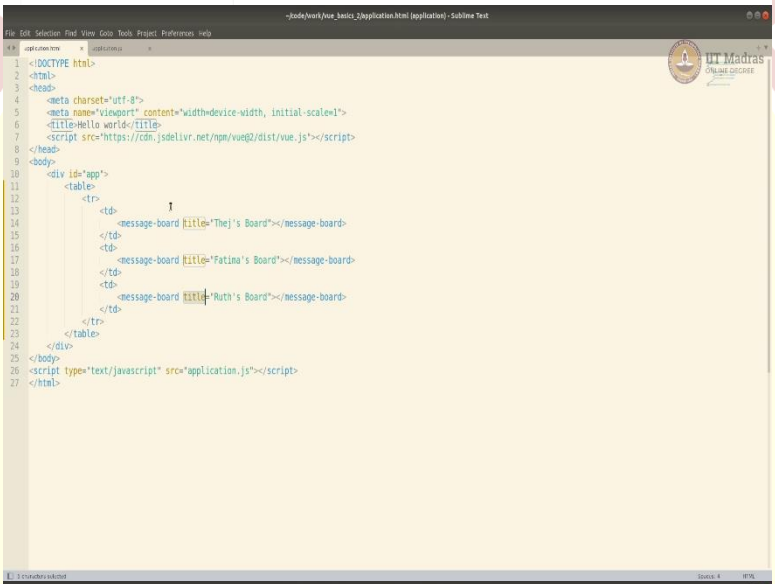
<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app"
})
Hello world
file:///home/tinaj/code/work/vue_basics_2/application.html


Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Vue

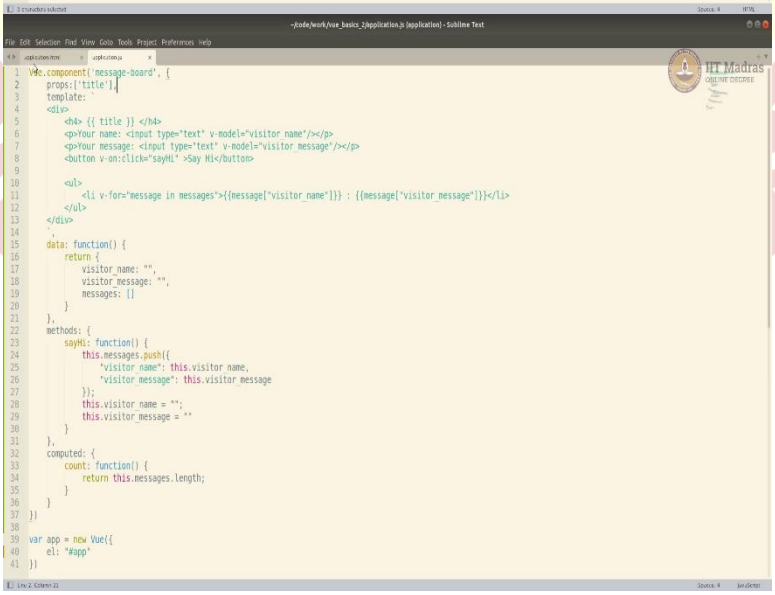

Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Thej's Board
Your name:
Your message:
Say Hi
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi




~/code/work/vue_basics_2/application.html (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello world</title>
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>

<table>
<tr>
<td>
<message-board title="Thej's Board"></message-board>
</td>
<td>
<message-board title="Fatima's Board"></message-board>
</td>
<td>
<message-board title="Ruth's Board"></message-board>
</td>
</tr>
</table>

</body>
<script type="text/javascript" src="application.js"></script>
</html>
1 character selected
Source: 4
HTML


~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
Vue.component('message-board', {
props:['title'],
template:

<h4> {{ title }} </h4>
Your name: <input type="text" v-model="visitor name"/>
Your message: <input type="text" v-model="visitor message"/>
<button v-on:click="sayHi" >Say Hi</button>

<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor name,
"visitor_message": this.visitor message
});
this.visitor_name = "";
this.visitor_message = ""
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app"
})
Line 2, Column 22
Source: 4
JavaScript

Now, the whole purpose of doing this was to have three message borders three independent message boards. Let us add them, I am just going to make it simple table to add them as like columns. You could use divs and you know, rows and columns using the frameworks that we have learned before, using our standard framework, bootstrap, you can create rows and columns and then make them that way, but I am for a quick use, I am just going to make it into a table.

A simple table with one row and column, three columns, each column containing a message board. So, I can see message boards here. So, you can see three message boards. Let us make it little more. You can do any styling here, you can do what all the formatting that you want to add. So, I am just going to do like a P to bring them into next line. You can add your styles and other classes and other things. Let us refresh this. So, I can see that now there behind there is a p.

Let us say Tej, welcome back for this let us say Fatima's board and this is the then say, hi, so it is reloading, that is because it is clearing the things, not reloading actually, clearing the things. Let us say this one Manoj also says, hi, but to this board, and Raj again says, hi, to Fatima's board. But now we want to add like a name for each of the one name of the board. We can do that by just adding for example a name here. We could probably like let us do, add a name here for each of the board. I will show you a better way of doing it but essentially let us start I would say your had Tej's board, Fatema's board, Ruth's board. So, you can see here which board is whose.

But I really do not want to do this. I want it to be part of our definitions of part of our board. I wanted to send a message along with this to actually our component saying this is the board name, that you can do by sending a property or you can do, for example, say title of the board and say Tej's board instead of doing here we are going to send it and make it part of the component.

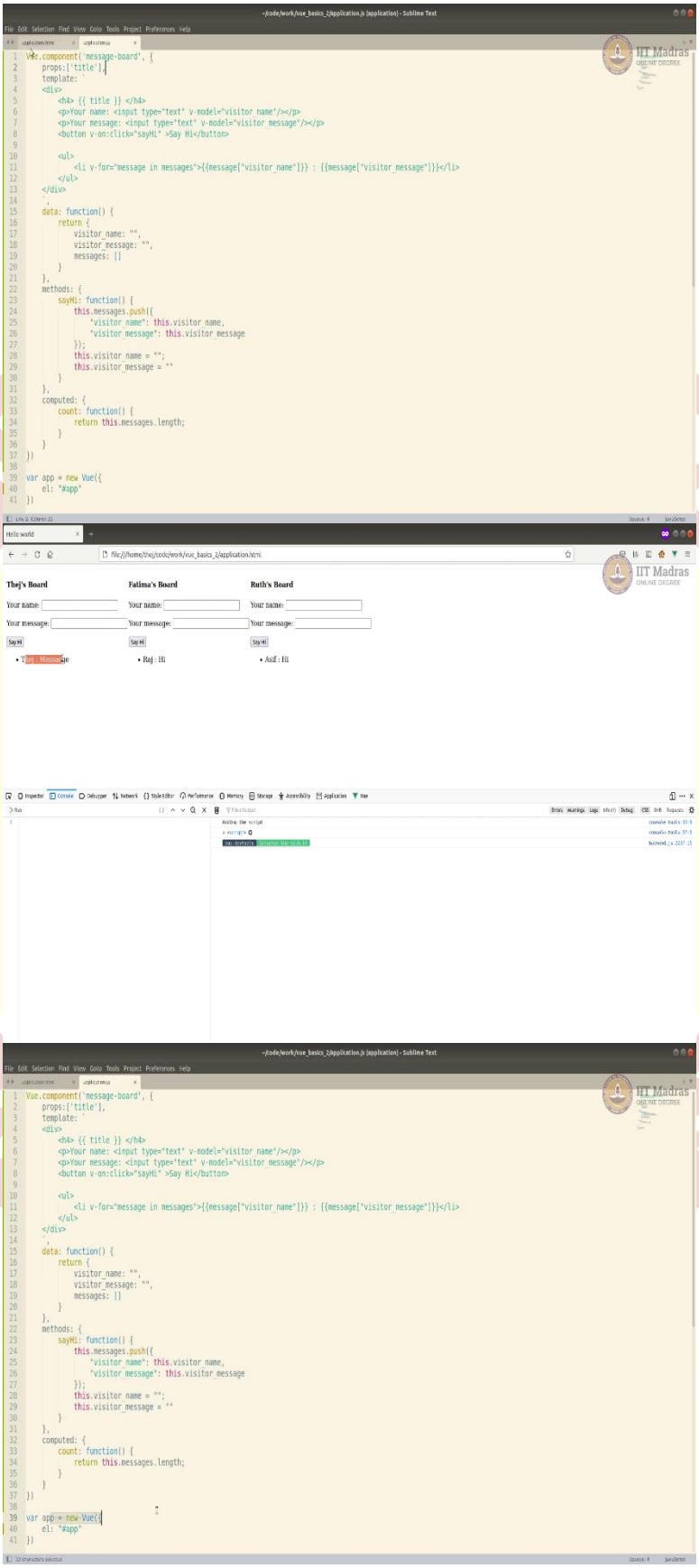
I will show you how to, how you will use title inside the template. Let us do this first. Fatima. So, I am going to remove that here, remove that here. I am going to refresh right and you can see nothing happens because we are sending it as title here, but we are not using it here. Now to use it, you have to define properties or they are called PROPS. You can for simple ones, you can just define it an arrays an array and we want to remove. We want to receive title, and we want to bring the title here.

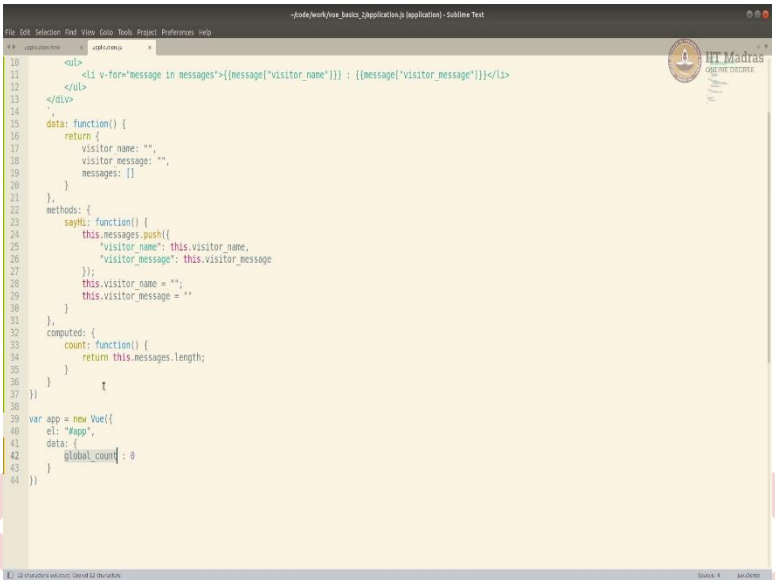
Now I am going to do like a h4, or h3, whatever. And then, like our regular property we can use here, and then you can save. Now you can see that you are sending the property here to the message port, which is defined, and we are using that property instead the message board. Okay, there seems to be something wrong. Oh, okay. Syntax error. Okay. So, let us refresh.

Okay, now we can see, you can see the name of the board, using the property. So, that is how you actually send values from the main app into the component. I mean, we just showed it as a title, but this is a way to send details or values, or some data from the main app into the component. You can send many things. You could probably send some style elements, for example, or it could send other properties like size, for example, how big or how small, you want the whole component to be things like that. And then you can use that here to format it or to make it the way you want, it. Basically, can send all kinds of data from here to here using properties.

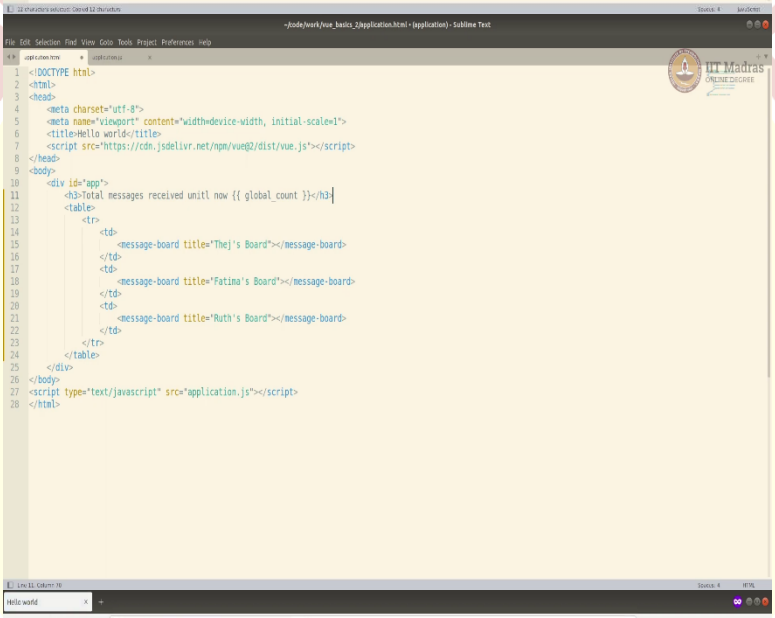
You can expand it to, you know, add some default values, and add some property types, like whether it is a string, integer, etc. but here, I am using simple property. You can explore that by yourself.

(Refer Slide Time: 17:04)

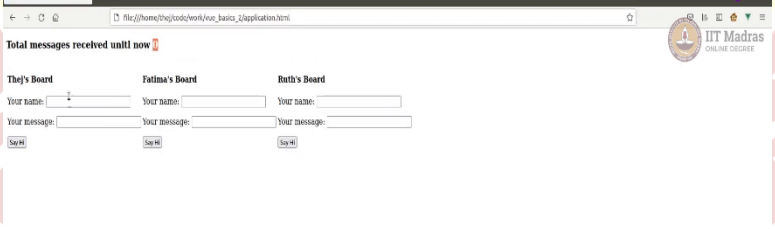


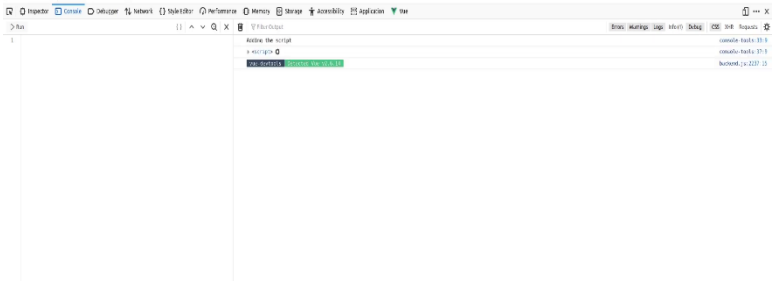


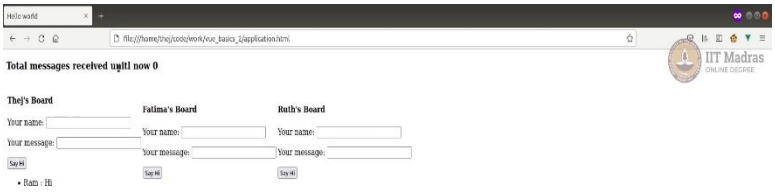
```
        <ul>
            <li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}</li>
        </ul>
    </div>
    `,
    data: function() {
        return {
            visitor_name: "",
            visitor_message: "",
            messages: []
        }
    },
    methods: {
        sayHi: function() {
            this.messages.push({
                "visitor_name": this.visitor_name,
                "visitor_message": this.visitor_message
            });
            this.visitor_name = "";
            this.visitor_message = ""
        }
    },
    computed: {
        count: function() {
            return this.messages.length;
        }
    }
})

var app = new Vue({
    el: "#app",
    data: {
        global_count : 0
    }
})
```

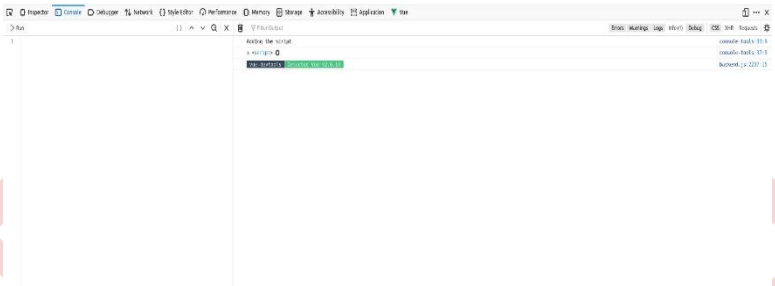


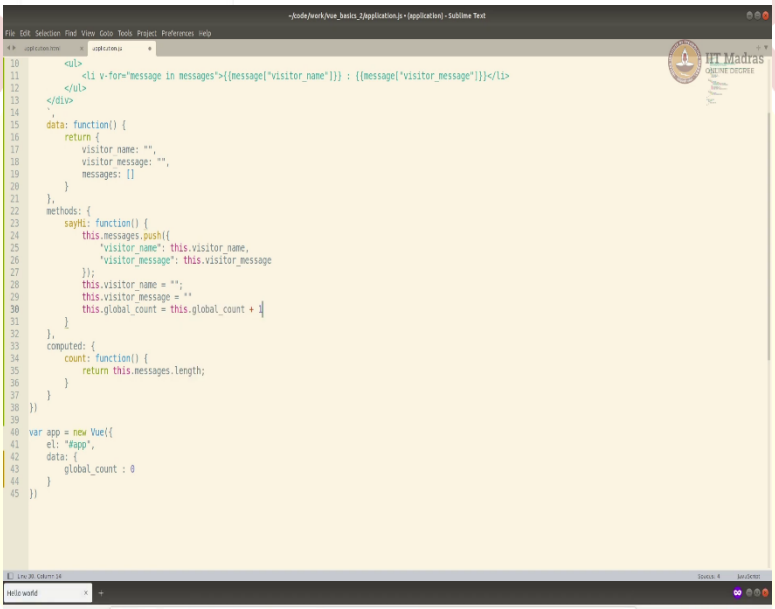
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello world</title>
    <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>
    <div id="app">
        <h3>Total messages received unitl now {{ global_count }}</h3>
        <table>
            <tr>
                <td>
                    <message-board title='Thej's Board"></message-board>
                </td>
                <td>
                    <message-board title="Fatima's Board"></message-board>
                </td>
                <td>
                    <message-board title="Ruth's Board"></message-board>
                </td>
            </tr>
        </table>
    </div>
</body>
<script type="text/javascript" src="application.js"></script>
</html>
```





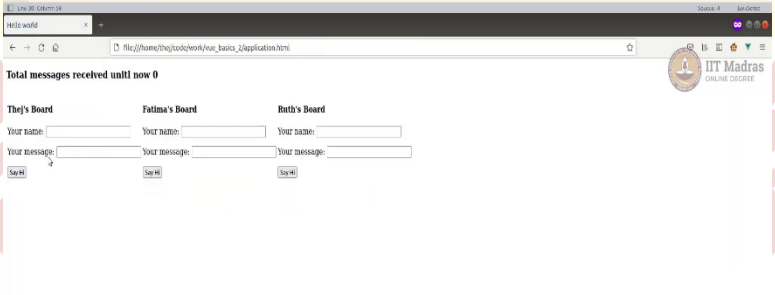

Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Total messages received until now 0
Thej's Board
Your name:
Your message:
Say Hi
• Ram : Hi
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi

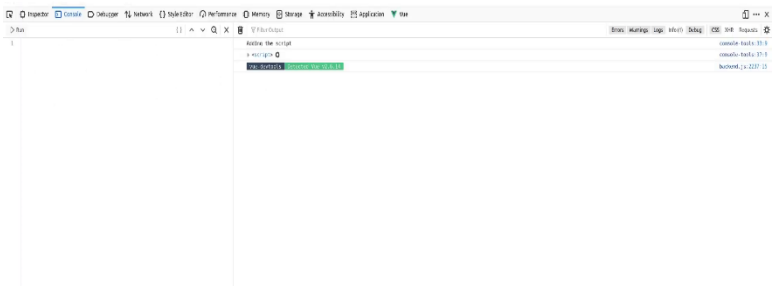



~/code/work/vue_basics_2/application.js • (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js

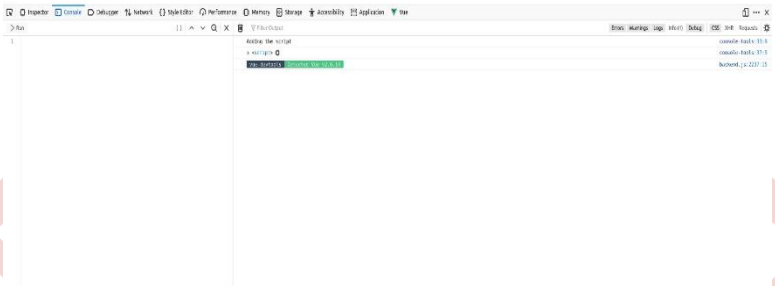
<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

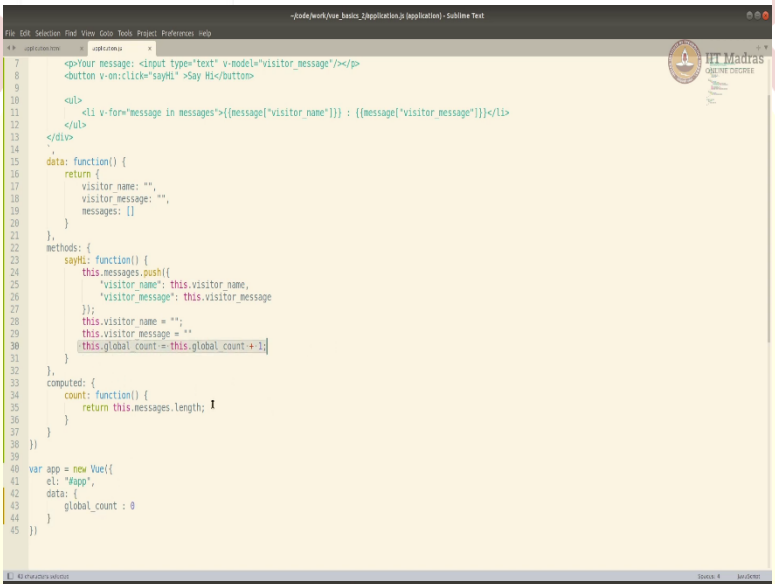
`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
this.global_count = this.global_count + 1
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app",
data: {
global_count : 0
}
})


Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Total messages received until now 0
Thej's Board
Your name:
Your message:
Say Hi
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi



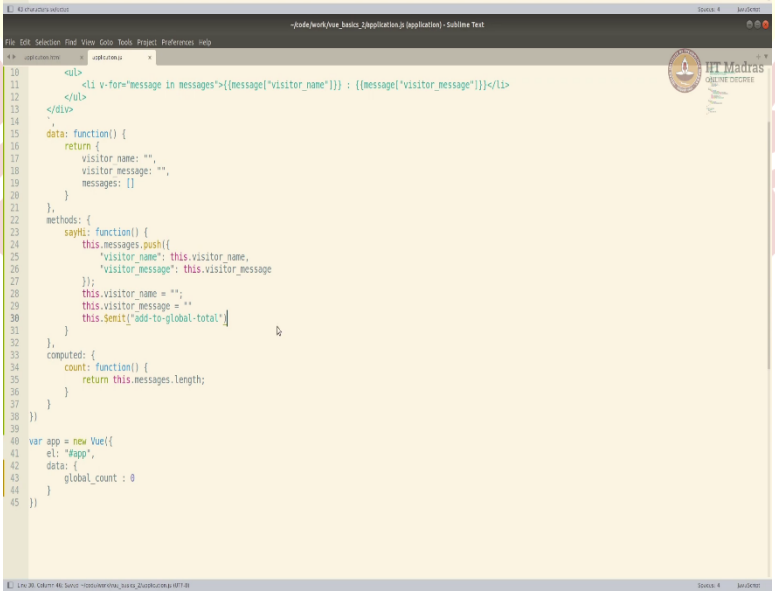

Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Total messages received until now 0
Thej's Board
Your name:
Your message:
Say Hi
• Ram : Hi
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi




~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
Your message: <input type="text" v-model="visitor_message"/>
<button v-on:click="sayHi" >Say Hi</button>

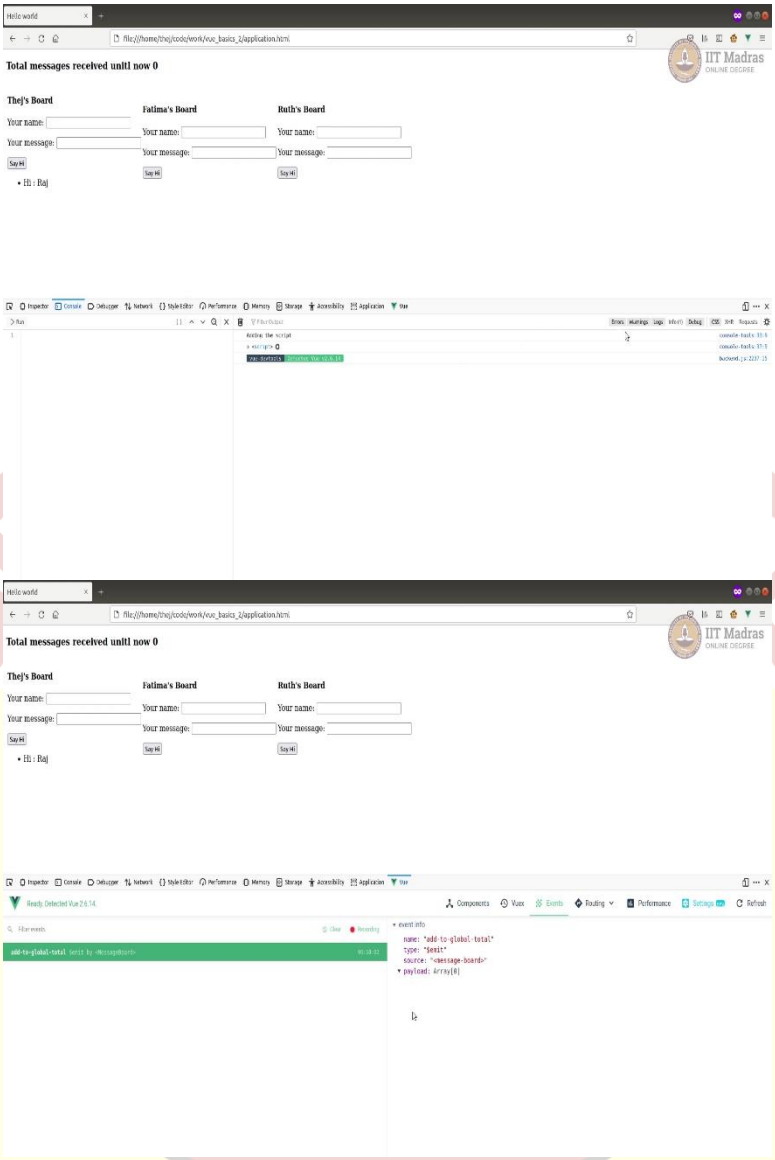
<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
this.global_count = this.global_count + 1;
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app",
data: {
global_count : 0
}
})


~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js

<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
this.$emit("add-to-global-total")
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app",
data: {
global_count : 0
}
})



Now, I want to do one more thing. Now we have added, like a message board, which works independently. Here is one, here, Raj says, hi. Asif, says, hi. Now, I want to show a global counter that says, how many messages were received in total? Currently, we know one message was here, one message is here, one message here. And if you see in our component we have computed, but it is at that individual level, that individual level, it is not the total overall application's total message received.

There is no way to count do that currently. Well, we could do if we could pass on while, when somebody says, hi, if you could tell that, we have received a message back to app saying you have received a message, and then we could count upon show them the total message, but there is no, currently, there is no way to send that back. We will see how to do that. We can do that by emitting the event.

I will show you how to emit an event from here to do that. You cannot really do for example, if I define data here, like we defined, which is a global app-level data, let us say global count is to 0. And then, I can use that here global or let us say total messages issued until now. I mean, you can do this let us say what happens. Let us say this is, it is 3 just to give some significance. It can be any tag.

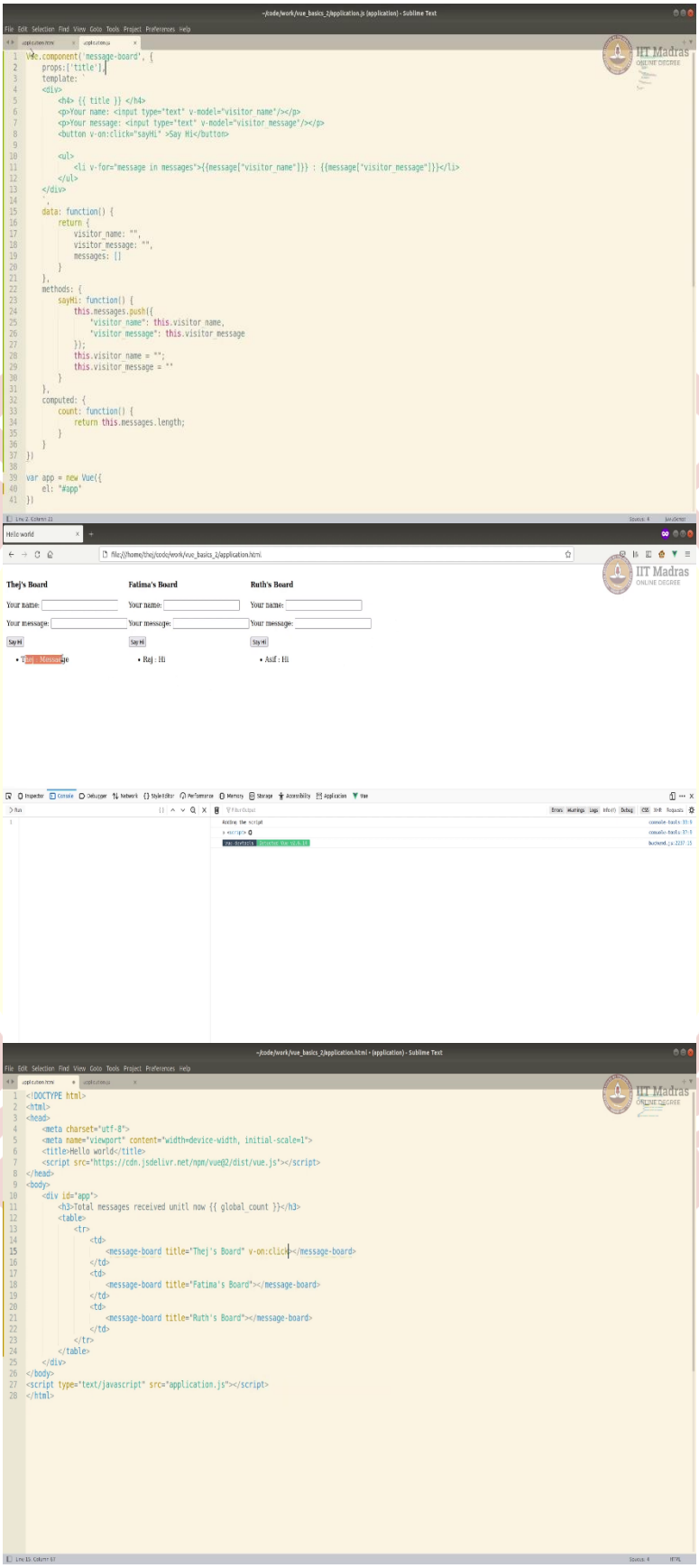
And then if I do, did I save it? Yes. I have saved it and let me refresh, so it shows 0. And now if I do Ram, hi, it is not incrementing because we have not incremented the global count. Now to increment the global count like we did before, this dot global count is equal to this dot global count plus 1. If we do that, let us see what happens.

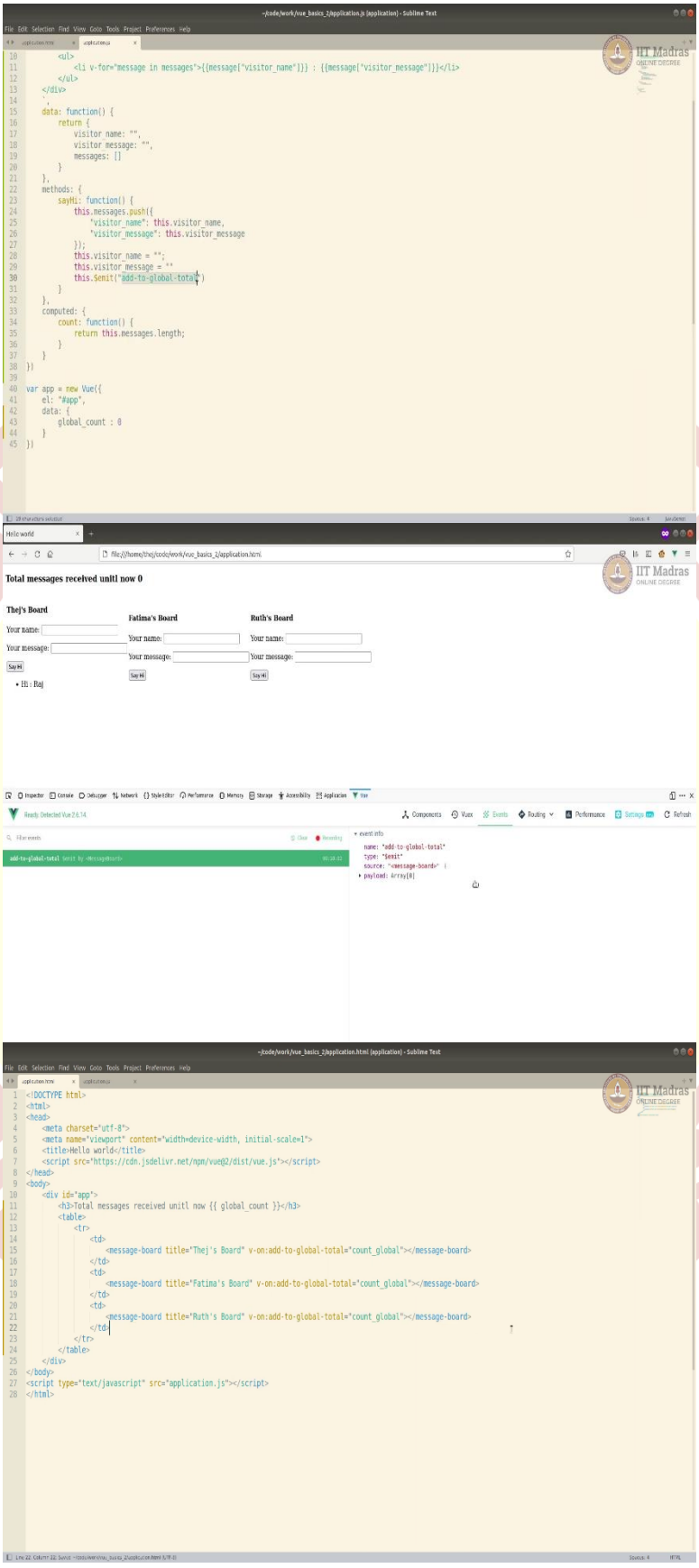
Okay, now, hi, Ram, hi. So now nothing happened. Because if you see, this dot global count does not exist here on this, it exists on the app. So, this nothing happens for this. Basically, there is no way to reach the main app using this. So, to reach that app, what we are going to do is, we are going to emit an event. What we are going to do is, we are going to call instead of this we are going to say, this dot dollar emit. This is how you raise an event. What is the name of the event that you want to raise? We want to raise add to global total.

So, this is the event we want to raise. Let us see what happens now when we do this. We are now, hi, Raj. And if I say hi, it did, hi, but if you go to your view, if you go to event, you can see that there is an event. The event name was add to global total, and it got emitted and the message was message, the source of the emit was a message board, this message board or any message board and the payload was array.

We are not currently worried about the payload, we just know that the event was raised and, but nothing was done. Because like any event, once the event is emitted, you need to handle it.

(Refer Slide Time: 22:06)



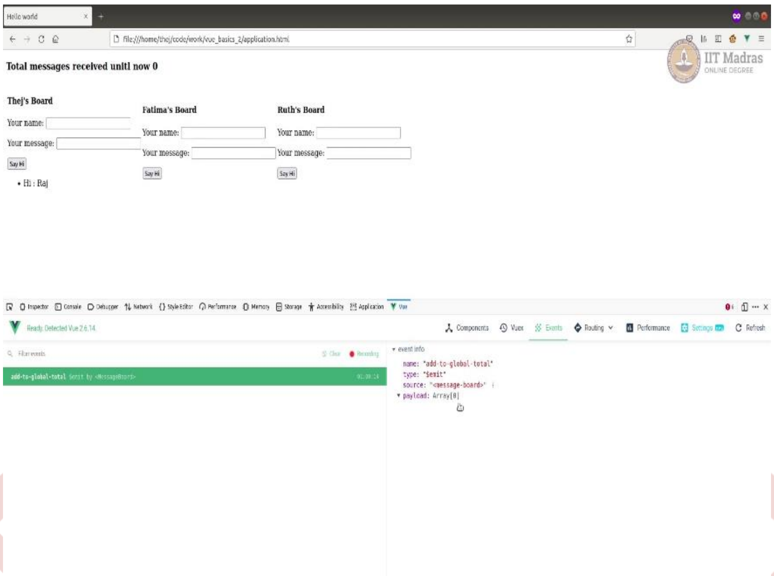

~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
IIT Madras
ONLINE DEGREE

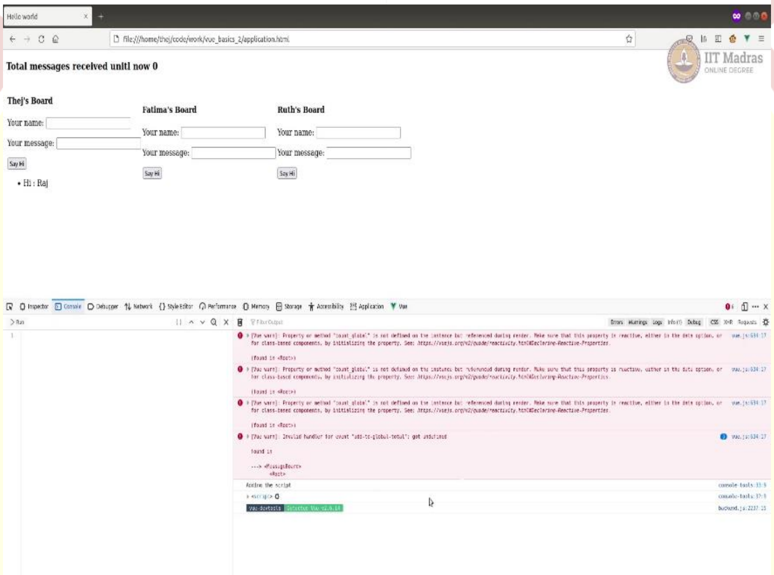
<li v-for="message in messages">{{message["visitor_name"]}} : {{message["visitor_message"]}}

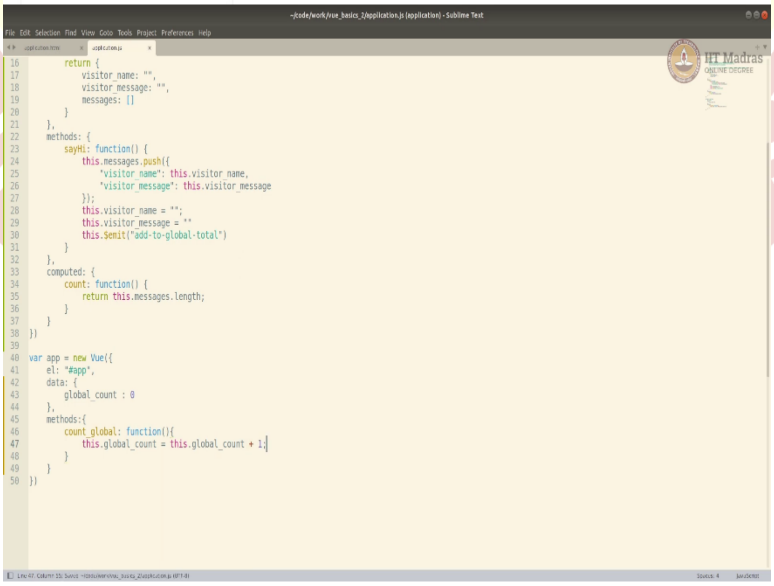
`,
data: function() {
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
this.$emit("add-to-global-total")
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app",
data: {
global_count : 0
}
})
Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Total messages received unitl now 0
Thej's Board
Your name:
Your message:
Say Hi
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi
Thej : Raj
Components
Vuex
Events
Routing
Performance
Settings
Refresh
Ready. Detected Vue 2.6.14.
Filter events
Clear
Recording
add-to-global-total $emit by <MessageBoard>
event info
name: "add-to-global-total"
type: "$emit"
source: "<message-board>"
payload: Array[0]
~/code/work/vue_basics_2/application.html (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
application.html
application.js
IIT Madras
ONLINE DEGREE
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Hello world</title>
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>

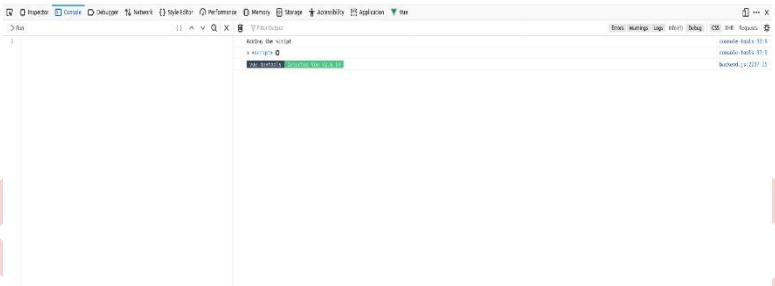
<h3>Total messages received unitl now {{ global_count }}</h3>
<table>
<tr>
<td>
<message-board title="Thej's Board" v-on:add-to-global-total="count_global"></message-board>
</td>
<td>
<message-board title="Fatima's Board" v-on:add-to-global-total="count_global"></message-board>
</td>
<td>
<message-board title="Ruth's Board" v-on:add-to-global-total="count_global"></message-board>
</td>
</tr>
</table>

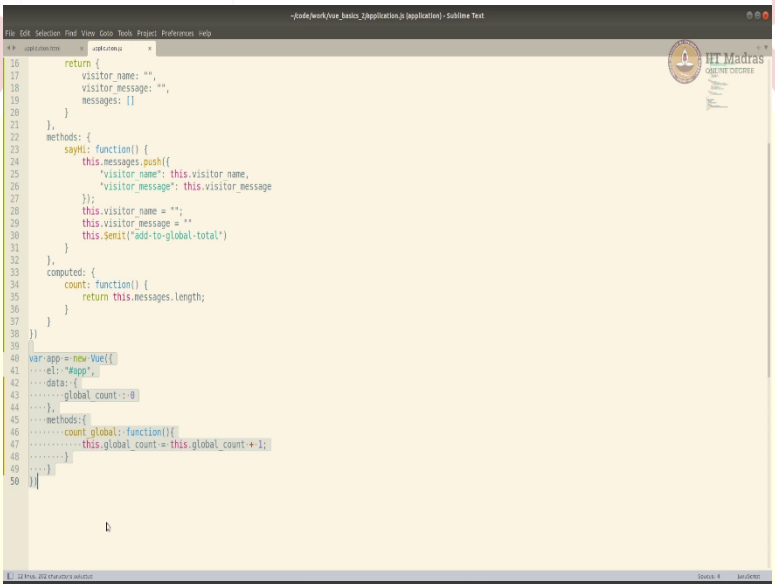
</body>
<script type="text/javascript" src="application.js"></script>
</html>

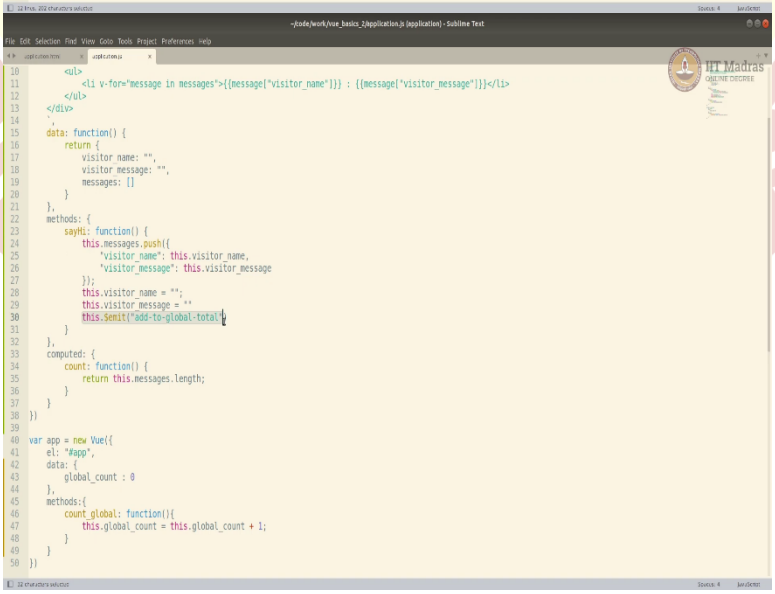

Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Total messages received until now 0
Thej's Board
Your name:
Your message:
Say Hi
• Hi : Raj
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi
IIT Madras
ONLINE DEGREE
name: "add-to-global-total"
type: "$emit"
source: "<message-board>"
payload: Array[0]


Hello world
file:///home/thej/code/work/vue_basics_2/application.html
Total messages received until now 0
Thej's Board
Your name:
Your message:
Say Hi
• Hi : Raj
Fatima's Board
Your name:
Your message:
Say Hi
Ruth's Board
Your name:
Your message:
Say Hi
IIT Madras
ONLINE DEGREE


~/code/work/vue_basics_2/application.js (application) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
return {
visitor_name: "",
visitor_message: "",
messages: []
}
},
methods: {
sayHi: function() {
this.messages.push({
"visitor_name": this.visitor_name,
"visitor_message": this.visitor_message
});
this.visitor_name = "";
this.visitor_message = ""
this.$emit("add-to-global-total")
}
},
computed: {
count: function() {
return this.messages.length;
}
}
})
var app = new Vue({
el: "#app",
data: {
global_count : 0
},
methods:{
count_global: function(){
this.global_count = this.global_count + 1;
}
}
})







Now, we did learn in the last thing, we could add an event handler by V on, by calling this V on. So, we can do the same thing. We can call we can use V on and handle the event. Just

give me a second, let us try something V on. It is not actually V colon it is V dash on colon. And what we want, what is the event, we usually used to do, click because we always handle click generally. But here, the event name is not click, event name is you know is add to global total. That is the name of the event, you can see here. Name of the event is add to global total.

So that is the same event name we are going to use it here, and then, we are going to call a function count global. This is a function that is going to handle this event. We are going to do the same thing for this too and for this too. Now, let us see what happened if I call this. I am going to refresh this.

Say hi, say Raj. Say hi. Event was raised, but nothing happened. It got raised, but no one handled it. In the console if you see, there is a event raised but nothing, no count global function to handle it. We have not defended function at all. So now we will just define this function. Where do we need to define this function? Because it is getting called emitted on the top of the event it is going to be defined at the app level, so we are going to define it here.

So, we are going to define like before, add an attribute called methods within that we will add a function, the function name will be count global and function. This function will not take any input, it will just increment this. This dot global count is equal to this dot global count plus 1.

Just checking all the brackets are closed. Now, I think it should work. Let us see whether it really works. Hi, not hi, Raj says, Hi see it got incremented here 1 globally. Tej says, Hi, Ruth says, Hi. See, this is how you actually pass the event. You do not have to just pass the event, you can also pass data along with the event. You can check out the details of emit on how to send the data and how to receive the data here and do the data.

For example, if you have a global pool of messages that you want to display on the side, not only individually, but also together, you can do that, I mean, just giving an example. But if I had a cart, you could add multiple products into the same single cart. Each cart could have, each for example, catalog could have different sections, but there could be one total catalog for the whole application. And that is how you send data back. Now, in this case, from a component back to the main app.

That is it for today. What you learnt is, how to create a component, how to handle component templates, how to send properties to the component from the main app to the component and

use that property. Then how to actually send emit the event and use that event to get data and update the main app. That is all in today's screencast, See you next time. Thank you.