

EE4216 Lab 3 – Reactive Web App Design

You are required to submit a lab report to explain the results of each checkpoint with supplementary information such as screenshots and source code as appropriate. A public URL of your website must be provided for each checkpoint for the grader to verify the results.

Objectives:

- Learn to create reactive UI with Vue
- Learn to integrate networking functions into Vue

Unlike server-side rendered apps that need expensive roundtrips, reactive web quickly reacts to any user interaction. The pressure on the backend servers is also decreased, making these apps scale much better under heavy loads, all leading to a smoother user experience.

Dynamic data table allows information to be passed and manipulated dynamically. In this lab, you will be guided to build a dynamic data table step by step using Vue. At start, you are given a Vue base app that contains a complete CSS and a script of the Vue root instance as follow:

<pre>1 /* style of table */ 2 table { 3 text-align: left; 4 font-family: 'Open Sans', sans-serif; 5 width: 500px; 6 border-collapse: collapse; 7 border: 2px solid #444777; 8 margin: 10px; 9 } 10 11 table th { 12 background: #444777; 13 color: #FFF; 14 padding: 5px; 15 min-width: 30px; 16 } 17 18 table td { 19 padding: 5px; 20 border-right: 2px solid #444777; 21 } 22 23 table tbody tr:nth-child(2n) { 24 background: #D4D8F9; 25 }</pre>	<pre>1 /* Vue root instance */ 2 var app = new Vue({ 3 el: '#app', 4 data: { 5 columns: { 6 id: 'ID', 7 name: 'Full Name', 8 phone: 'Phone', 9 }, 10 rows: [11 { id: 1, name: "Test Name 1", phone: '305-917-1301'}, 12 { id: 2, name: "Test Name 2", phone: '210-684-8953'}, 13 { id: 3, name: "Test Name 3", phone: '765-338-0312'}, 14] 15 } 16 });</pre>
---	--

Task 1 - Creating a Vue Table Template

From the script above, you can see two sets of test data – **columns** and **rows** – are used to define the structure and content of the table shown below. You are asked to create this HTML table using Vue template. In this task, you do not need to modify the given CSS and JS.

1. Mount your Vue app instance to a <div> container in your HTML.
2. Use **v-for** directive to iterate through the columns and rows array to generate table header and body respectively.

ID	Full Name	Phone
1	Test Name 1	305-917-1301
2	Test Name 2	210-684-8953
3	Test Name 3	765-338-0312

Checkpoint 1:

Provide a public URL to demonstrate your result. Explain your Vue template code briefly.

Task 2 – Adding Dynamic Content to Table

In the previous task, the rows of the table are preset for testing purpose. Now you are asked to load the table with dynamic user data from a remote API endpoint. In this task, you only need to modify the JS script file. User data can be obtained from this endpoint:

<https://jsonplaceholder.typicode.com/users>

1. Write a function to asynchronously fetch the user data from the above endpoint and update the table rows.
2. Hook the function up to the Vue instance's lifecycle such that the user data is automatically loaded into the table before the table is displayed.

ID	Full Name	Phone
1	Leanne Graham	1-770-736-8031 x56442
2	Ervin Howell	010-692-6593 x09125
3	Clementine Bauch	1-463-123-4447
4	Patricia Lebsack	493-170-9623 x156
5	Chelsey Dietrich	(254)954-1289
6	Mrs. Dennis Schulist	1-477-935-8478 x6430
7	Kurtis Weissnat	210.067.6132
8	Nicholas Runolfsdottir V	586.493.6943 x140
9	Glenna Reichert	(775)976-6794 x41206
10	Clementina DuBuque	024-648-3804

The content of the table is loaded from *JsonPlaceholder*.

Checkpoint 2:

Provide a public URL to demonstrate your result. Explain your modified JavaScript code briefly.

Task 3 – Manipulating Data in Table

In this task, you are asked to add a sorting function to the table. When a column header is first clicked, the table will be sorted by that column in ascending order. Subsequent clicks will toggle the sorting order inversely.

1. Write a function to sort an 2D array (table rows) by a given key (column).
2. Attach the sort function to the Vue instance.
3. Use `v-on:click` directive to trigger the sort function.
4. Use `v-bind:class` directive to append a small arrow (up and down) to the header to indicate the current ordering. The arrow can be made with this CSS trick:

<https://css-tricks.com/snippets/css/css-triangle/>.

ID	Full Name ▲	Phone
5	Chelsey Dietrich	(254)954-1289
10	Clementina DuBuque	024-648-3804
3	Clementine Bauch	1-463-123-4447
2	Ervin Howell	010-692-6593 x09125
9	Glenna Reichert	(775)976-6794 x41206
7	Kurtis Weissnat	210.067.6132
1	Leanne Graham	1-770-736-8031 x56442
6	Mrs. Dennis Schulist	1-477-935-8478 x6430
8	Nicholas Runolfsdottir V	586.493.6943 x140
4	Patricia Lebsack	493-170-9623 x156

ID ▼	Full Name	Phone
10	Clementina DuBuque	024-648-3804
9	Glenna Reichert	(775)976-6794 x41206
8	Nicholas Runolfsdottir V	586.493.6943 x140
7	Kurtis Weissnat	210.067.6132
6	Mrs. Dennis Schulist	1-477-935-8478 x6430
5	Chelsey Dietrich	(254)954-1289
4	Patricia Lebsack	493-170-9623 x156
3	Clementine Bauch	1-463-123-4447
2	Ervin Howell	010-692-6593 x09125
1	Leanne Graham	1-770-736-8031 x56442

The table is sorted by the name ascendingly (left) and by the ID descending (right).

Checkpoint 3:

Provide a public URL to demonstrate your result. Explain your code in the HTML, JavaScript and CSS files in detail respectively.

- END -