Faculty of Science, Engineering and Technology

Interface Design and Development

Pass Task 5.1: Creating Components

Overview

Components enables you to create modular codes for your web application. In this task, you will create a status posting web application that uses components to insert a status into the timeline. As there is no persistent storage, the timeline will initially be empty. The latest status will always be displayed on top pushing down previous status. Each status will also contain a delete button that will enable the user to delete the specific status.

Purpose: Learn how to use components.

Task: Create a web app that allows users to post status on a timeline, and each timeline will have its own delete button to delete the message.

Time: This task should be completed in your tutorial and submitted for feedback before the start of week 7.

Resources:

- Lecture notes #3, #4 and #5

Submission Details

You must submit the following files:

- Status posting source code (mypost.html).
- Status posting component source code (appmypost.js).
- Screenshot of the web app.

Please submit the mypost.html and appmypost.js as one zip file and submit the screenshot as a separate file.

Make sure that your task has the following in your submission:

- The status posting web application is HTML5 compliant.
- Demonstrates understanding in using the VueJS framework.
- Demonstrates use of VueJS compoments.

Instructions

Implement the status posting web app. In this web application, you will need to create a Vue component to insert/delete a status into/from an array of status. Status is inserted as the first element of the array, while each status will have a button that allows it to be deleted.

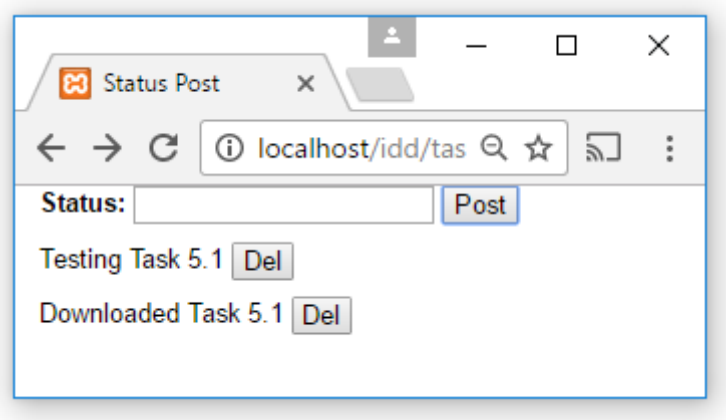
1. Start by creating a new HTML file in an editor (eg. VS Code, Brackets).

2. Implement the basic outline of an VueJS web app with the appropriate scripts.

Note: It is a good practice to write the component in a separate file.

```
<script src="js/appmypost.js"></script>
```

3. The HTML file will have
   - User input for the user to enter their status followed by a post button
   - Status view that shows the message posted followed by a delete button



Hint: Use v-for to list the status, which can be in a div, list or table. Use “index” to identify which status to delete, by using it as a parameter for the delete method.

4. Create the appmypost.js (a sample template in the next page).
5. Your web app should now be complete. Make sure you test it on the browser to make sure that it works as you expect.

Web App: **appmypost.js**

```
const app = Vue.createApp({  })

app.component('app-mypost', // indicating the component tag
{
    // defining data to be used in the component
    data:function(){
         return{
              statPosts:[],
              strStatus:''
         }
    },
    // define the template for the component
    template:
    // your code here
         ,
    // defining the methods for add and remove status
messages
    methods:{
         add:function(status){
              //push status into statPosts array
         },
         remove:function(index){
              //delete status from statPost using index
         }
    }
});
app.mount('#app')
```