Show of hands, history, layout, Vue instance, data binding and events, displaying lists, error handling, posting

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Nova Code Camp Todo</title>

    <script src="//unpkg.com/vue"></script>

    <script src="//unpkg.com/axios/dist/axios.min.js"></script>

    <link rel="stylesheet" href="//unpkg.com/bulma/css/bulma.css">

    <style>

      .strikethrough {

        text-decoration: line-through;

      }

      .table th, .table td {

        width: 30%;

      }

    </style>

  </head>

  <body>

    <section class="section">

      <div class="container">

        <h1 class="title is-1 has-text-grey">Nova Code Camp Todo</h1>

        <p class="subtitle has-text-light-grey">Let's get organized!</p>

        <hr/>

        <div id="app">

          <div class="columns">

            <div class="column">

              <h3 class="title is-3"><em>Todo Items</em></h2>

              <table class="table is-fullwidth">

                <thead>

                  <tr>

                    <th>Title</td>

                    <th>Description</td>

                    <th></td>

                  </tr>

                </thead>

                <tbody>

                  <tr v-for="(item, index) in items">

                    <td class="has-text-weight-bold is-size-4" :class="{ 'has-text-grey-lighter strikethrough': item.done, 'has-text-primary': !item.done }">

{{ item.title }}

</td>

                    <td style="white-space: pre;">{{ item.description }}</td>

                    <td>

                      <button v-on:click="item.done = true" v-if="!item.done" class="button is-success">Done!</button>

                      <button v-on:click="items.splice(index, 1)" class="button is-danger">Remove</button>

                    </td>

                  </tr>

                </tbody>

              </table>

            </div>

            <div class="column">

              <h3 class="title is-3"><em>Add An Item</em></h2>

              <div class="field">

                <label class="label">Title</label>

                <div class="control">

                  <input type="text" class="input" v-model="title" :class="{ 'is-danger': titleEmpty }">

                  <p v-if="titleEmpty" class="help is-danger">A title is required.</p>

                </div>

              </div>

              <div class="field">

                <label class="label">Description</label>

                <div class="control">

                  <textarea class="textarea" v-model="description"></textarea>

                </div>

              </div>

              <button class="button is-primary" v-on:click="add" :class="{ 'is-loading': loading }">

Add Item

</button>

            </div>

          </div>

        </div>

      </div>

    </section>

  </body>

</html>

<script>

  new Vue({

    el: '#app',

    data: {

      items: [],

      title: null,

      description: null,

      titleEmpty: false,

      loading: false

    },

    methods: {

      add: function() {

        if(this.title === null) {

          this.titleEmpty = true;

          return;

        }

        // Create and add the new item

        var item = {

          title: this.title,

          description: this.description,

          done: false

        };

        //this.items.push(item);

        // Reset the entry state

        //this.title = null;

        //this.description = null;

        //this.titleEmpty = false;

        // Send to the server

        this.loading = true;

        var self = this;

        axios.post('/', item)

          .then(function (response) {

            self.items.push(item);

            self.title = null;

            self.description = null;

            self.titleEmpty = false;

            self.loading = false;

          })

          .catch(function (error) {

            console.log(error);

            self.loading = false;

          });

      }

    }

  });

</script>

public class TodoItem

{

public string Title { get; set; }

public string Description { get; set; }

}

public class HomeController : Controller

{

public IActionResult Index()

{

return View();

}

[HttpPost]

public IActionResult Index([FromBody] TodoItem item)

{

Thread.Sleep(2000);

Console.WriteLine($"Title: {item.Title}, Description: {item.Description}");

return Ok();

}

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}