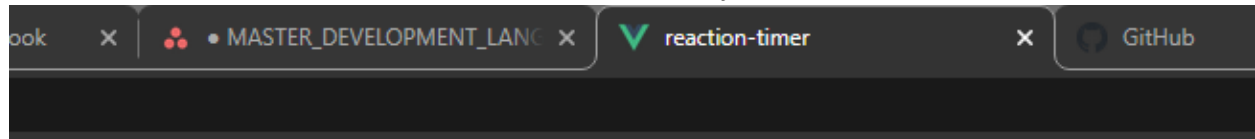


These are other notes on View from the Reaction-Timer Project



Ninga Reaction Timer

Play

Reaction Time: 440 ms

Too Slow .. lol

GitHub Repo:

<https://github.com/lionel5116/VueReactionTimerProject.git>

Tip: Another tip on emitting methods

Tip: Passing props (just a refresher)

Tip: Adding Bootstrap to Vue

Tip: Adding Routing in Vue 3

Tip: Refactoring the app for the routing and nav

Tip: Deployment and routing

Tip: Vuex (State Management) in Vue

Tip: Using axios in vue

Tip: Running unit tests with jest an vue tools

Tip: Another tip on emitting methods

If you look below, we are emitting a method that we will name 'end' in the parent component. Notice how we are passing data along with it.

```
22 console.log('component mounted')
23 setTimeout(() => {
24   this.showBlock = true;
25   this.startTimer();
26   console.log(this.delay)
27 }, this.delay)
28 },
29 updated() {
30   console.log('component updated')
31 },
32 unmounted(){
33   console.log('component unmounted')
34 },
35 methods: {
36   startTimer(){
37     //increment reaction time by 10ms with set interval
38     this.timer = setInterval(() =>{
39       this.reactionTime += 10
40     }, 10);
41   },
42   stopTimer(){
43     clearInterval(this.timer)
44     console.log(this.reactionTime)
45     this.$emit('end',this.reactionTime)
46   },
47 }
48 }
49 </script>
50
51 <style>
52 .block {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

DONE Compiled successfully in 350ms

App running at:
- Local: http://localhost:8080/
- Network: http://10.241.8.165:8080/

Now over in our parent component:

Notice below how we are making reference to the emitter by its name ('end')

We are pointing it to the method "endgame"

In the method, we are specifying an argument as a parameter for the function. When created the emitter, we passed in some data, so what that means is that the method create automatically receives that data as it's parameter.

EXPLORER

REACTION-TIMER

- node_modules
- public
- src
 - assets
 - components
 - MyBlock.vue
 - MyResults.vue
 - App.vue
 - main.js
- .gitignore
- ~\$ewNotes-ReactionTimerProject.docx
- babel.config.js
- jsconfig.json
- package-lock.json
- package.json
- README.md
- ViewNotes-ReactionTimerProject.docx
- vue.config.js

src > App.vue > {} "App.vue" > template > p

```
3 <button
4   @click="start"
5   :disabled="isPlaying"
6 >Play</button>
7 <MyBlock
8   v-if="isPlaying"
9   :delay="delay"
10  @end="endGame"/>
11 <p>Reaction Time: {{score}}</p>
12 </template>
13
14 <script>
15
16 import MyBlock from './components/MyBlock.vue'
17
18 export default {
19   name: 'App',
20   components: {
21     MyBlock,
22   },
23   data() {
24     return {
25       isPlaying: false,
26       delay: null,
27       score: null
28     }
29   },
30   methods: {
31     start(){
32       this.delay = 2000 + Math.random() * 5000
33       //the component will 'mount' when it becomes visible in the DOM
34       this.isPlaying = true;
35     },
36     endGame(reactionTime){
37       this.score = reactionTime;
38       this.isPlaying = false;
39     }
40   }
41 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

DONE Compiled successfully in 350ms

App running at:

- Local: <http://localhost:8080/>
- Network: <http://192.168.1.105:8080/>

Tip: Passing props (just a refresher)
We declare props

```
Help MyResul
... App.vue M MyResults.vue U X MyBlock.vue U
src > components > MyResults.vue > {} "MyResults.vue" > script
1 <template>
2   <div>
3     <p>Reaction Time: {{score}} ms</p>
4   </div>
5 </template>
6
7 <script>
8 export default {
9   name: 'MyResults',
10  props: ['score'],
11  data() {
12    return {
13
14    }
15  },
16 }
17 </script>
18
19 <style>
20
21 </style>
```

Then to receive the prop from the parent:

```

src > ✓ App.vue > {} "App.vue" > template
5 |         :disabled= !isPlaying
6 |     >Play</button>
7 |
  |     You, 4 minutes ago • Uncommitted changes
8 |     <MyBlock
9 |         v-if="isPlaying"
10 |         :delay="delay"
11 |         @end="endGame"/>
12 |
13 |     <MyResults
14 |         v-if="showResults"
15 |         :score="score"
16 |     />
17 |
18 |     <!--<p v-if="showResults">Reaction Time: {{score }} ms</p>-->
19 |
20 | </template>
21 |
22 | <script>
23 |
24 | import MyBlock from './components/MyBlock.vue';
25 | import MyResults from './components/MyResults.vue';
26 |
27 | export default {
28 |     name: 'App',
29 |     components: {
30 |         MyBlock,
31 |         MyResults
32 |     },
33 |     data() {
34 |         return {
35 |             isPlaying: false,
36 |             delay:null,
37 |             score:null,
38 |             showResults: false
39 |         }
40 |     },
41 |     methods: {
42 |         start(){
43 |             this.delay = 2000 + Math.random() * 5000

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

GITLENS

Tip: Adding Bootstrap to Vue

In vue 3 , you have to use bootstrap 5

You can install bootstrap the same way (regular npm) as any other node.js project

<https://getbootstrap.com/docs/5.0/components/navbar/>

```
npm i bootstrap@5.0.1
```

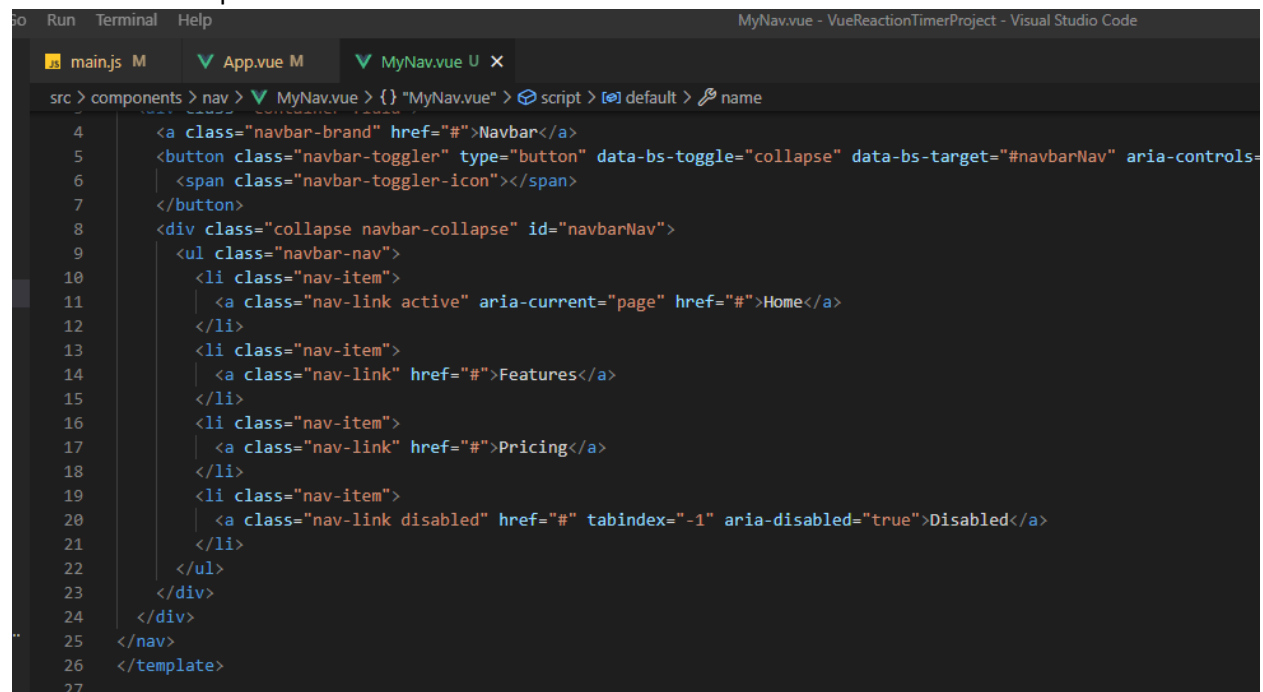
```
npm install --save @popperjs/core
```

Then In main.js

```
import "bootstrap/dist/css/bootstrap.min.css"
```

```
import "bootstrap"
```

Create a nav component

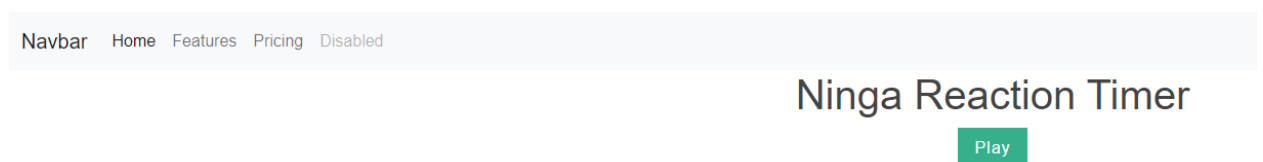
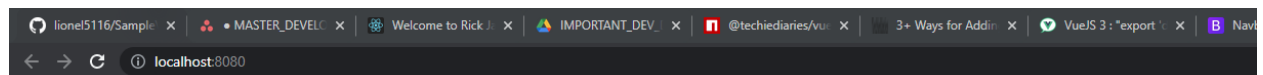


```
MyNav.vue - VueReactionTimerProject - Visual Studio Code
src > components > nav > MyNav.vue > {} "MyNav.vue" > script > default > name
4   <a class="navbar-brand" href="#">Navbar</a>
5   <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls=
6     <span class="navbar-toggler-icon"></span>
7   </button>
8   <div class="collapse navbar-collapse" id="navbarNav">
9     <ul class="navbar-nav">
10      <li class="nav-item">
11        <a class="nav-link active" aria-current="page" href="#">Home</a>
12      </li>
13      <li class="nav-item">
14        <a class="nav-link" href="#">Features</a>
15      </li>
16      <li class="nav-item">
17        <a class="nav-link" href="#">Pricing</a>
18      </li>
19      <li class="nav-item">
20        <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
21      </li>
22    </ul>
23  </div>
24 </div>
25 </nav>
26 </template>
27
```

```
src > App.vue > {} "App.vue" > template > MyBlock
You, 4 minutes ago | 2 authors (p00149021@houstonisd.org Mag17615@7 and others)
1 <template>
2 <MyNav />
3 <h1>Ninga Reaction Timer</h1>
4 <button
5   @click="start"
6   :disabled="isPlaying"
7 >Play</button>
8
9 <MyBlock
10   v-if="isPlaying"
11   :delay="delay"
12   @end="endGame"/>
13
14 <MyResults
15   v-if="showResults"
16   :score="score"
17 />
18
19 <!--<p v-if="showResults">Reaction Time: {{score }} ms</p-->
20
21 </template>
22
23 <script>
24
25 import MyBlock from './components/MyBlock.vue';
26 import MyResults from './components/MyResults.vue';
27 import MyNav from './components/nav/MyNav.vue';
28
29 export default {
30   name: 'App',
31   components: {
32     MyBlock,
33     MyResults,
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

Module Warning (from ./node_modules/postcss-loader/dist/cjs.js):
Warning



You can then use bootstrap anywhere in your application. Just refer to the bootstrap docs

Outline buttons

In need of a button, but not the hefty background colors they bring? Replace the default modifier classes with the `.btn-outline-*` ones to remove all background images and colors on any button.



```
<button type="button" class="btn btn-outline-primary">Primary</button>
<button type="button" class="btn btn-outline-secondary">Secondary</button>
<button type="button" class="btn btn-outline-success">Success</button>
<button type="button" class="btn btn-outline-danger">Danger</button>
<button type="button" class="btn btn-outline-warning">Warning</button>
<button type="button" class="btn btn-outline-info">Info</button>
<button type="button" class="btn btn-outline-light">Light</button>
<button type="button" class="btn btn-outline-dark">Dark</button>
```

[Copy](#)

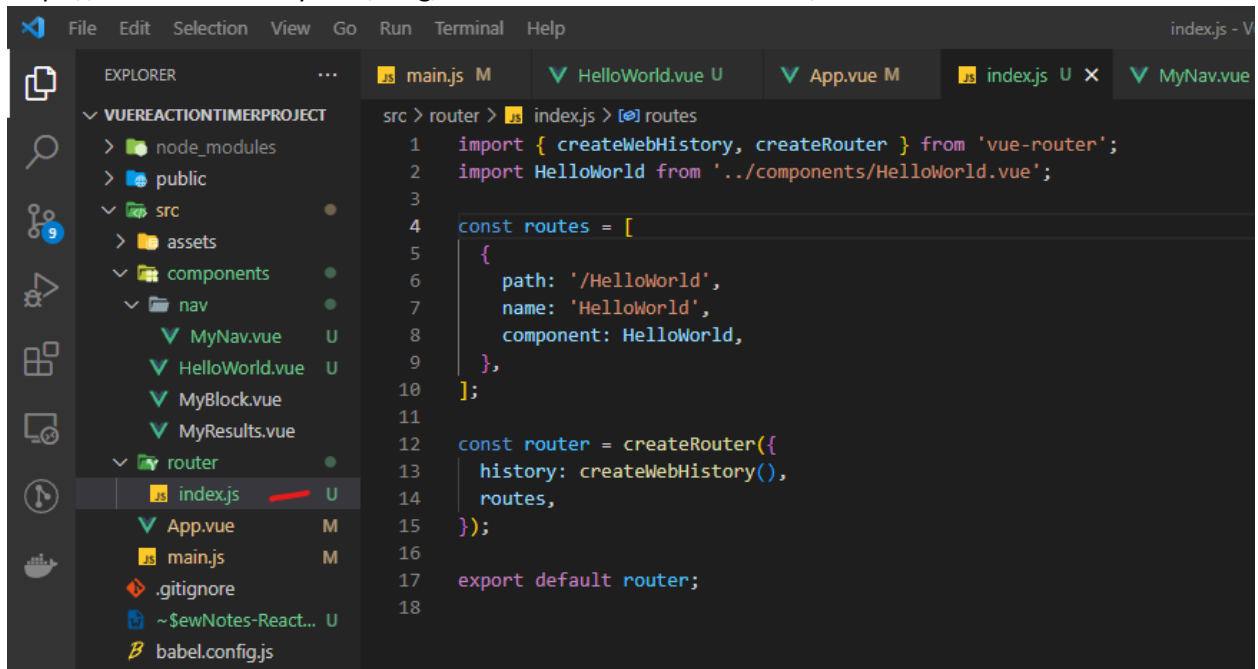
```
src > App.vue > {} "App.vue" > style
You, 1 second ago | 2 authors (p00149021@houstonisd.org Mag17615@7 and others)
1 <template>
2   <MyNav />
3   <h1>Ninga Reaction Timer</h1>
4   <button
5     class="btn btn-outline-info"
6     @click="start"
7     :disabled="isPlaying"
8   >Play</button>
9
10  <MyBlock
11    v-if="isPlaying"
12    :delay="delay"
13    @end="endGame"/>
14
15  <MyResults
```


Ninga Reaction Timer

Play

Tip: Adding routing in Vue 3

<https://www.vuemastery.com/blog/vue-router-a-tutorial-for-vue-3/>



The screenshot shows a VS Code editor interface for a project named 'VUEREACTIONTIMERPROJECT'. The Explorer sidebar on the left shows the file structure, including 'src', 'components', and 'router'. The 'router' folder is expanded, showing 'index.js' selected. The main editor area displays the code in 'index.js', which sets up a Vue Router instance with a single route for '/HelloWorld'. The code is as follows:

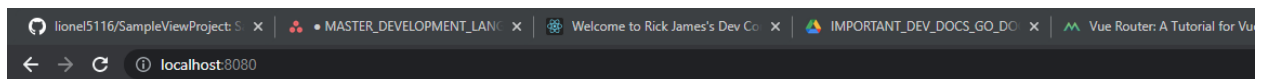
```
src > router > index.js > [⌘] routes
1  import { createWebHistory, createRouter } from 'vue-router';
2  import HelloWorld from '../components/HelloWorld.vue';
3
4  const routes = [
5    {
6      path: '/HelloWorld',
7      name: 'HelloWorld',
8      component: HelloWorld,
9    },
10 ];
11
12 const router = createRouter({
13   history: createWebHistory(),
14   routes,
15 });
16
17 export default router;
```

In main.js

```
main.js M X HelloWorld.vue U App.vue M index.js U MyNav.vue U
src > main.js
You, 19 minutes ago | 2 authors (You and others)
1 import { createApp } from 'vue'
2 import App from './App.vue'
3 import router from './router/index';
4
5
6 import "bootstrap/dist/css/bootstrap.min.css"
7 import "bootstrap"
8
9
10 createApp(App).use(router).mount('#app')
11
```

In your Custom Nav Component

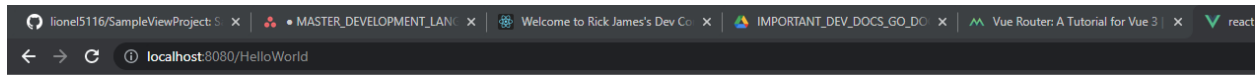
```
Go Run Terminal Help MyNav.vue - VueReactionTimerProject - Visual Studio Code
main.js M HelloWorld.vue U App.vue M index.js U MyNav.vue U X
src > components > nav > MyNav.vue > {} "MyNav.vue" > template > div > nav.navbar.navbar-expand-lg.navbar-light.bg-light
1 <template>
2 <div>
3 <nav class="navbar navbar-expand-lg navbar-light bg-light">
4 <div class="container-fluid">
5 <a class="navbar-brand" href="#">Navbar</a>
6 <button
7   class="navbar-toggler"
8   type="button"
9   data-bs-toggle="collapse"
10  data-bs-target="#navbarNav"
11  aria-controls="navbarNav"
12  aria-expanded="false"
13  aria-label="Toggle navigation"
14 >
15 <span class="navbar-toggler-icon"></span>
16 </button>
17 <div class="collapse navbar-collapse" id="navbarNav">
18 <ul class="navbar-nav">
19 <li class="nav-item">
20 <a class="nav-link active" aria-current="page" href="/">Home</a>
21 </li>
22 <li class="nav-item">
23 <router-link to="/HelloWorld" class="nav-link">HelloWorld</router-link>
24 </li>
25 </ul>
26 </div>
27 </div>
28 </nav>
29 </div>
30 <router-view />
31 </template>
32
33 <script>
```



Navbar Home HelloWorld

Ninga Reaction Timer

Play



Navbar Home HelloWorld

I AM RICK JAMES!!!!

Hello World Test SFC

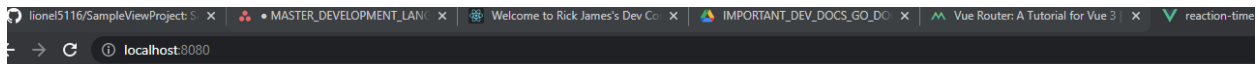
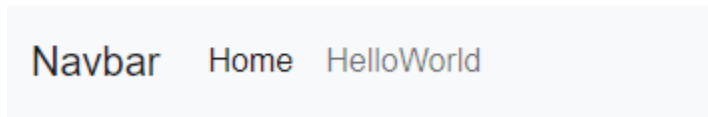
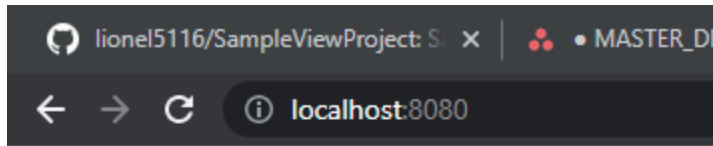
Ninga Reaction Timer

Play

For the home link

```
<div>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Navbar</a>
      <button
        class="navbar-toggler"
        type="button"
        data-bs-toggle="collapse"
        data-bs-target="#navbarNav"
        aria-controls="navbarNav"
        aria-expanded="false"
        aria-label="Toggle navigation"
      >
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link active" aria-current="page" href="/">Home</a>
          </li>
          <li class="nav-item">
            <router-link to="/HelloWorld" class="nav-link">HelloWorld</router-link>
          </li>
        </ul>
      </div>
    </div>
  </nav>
</div>
```

You don't need a route



Navbar Home HelloWorld

Ninga Reaction Timer

Play

Tip: Refactoring the app for the routing and navigation
I refactored the app to correct the navigation

```
Run Terminal Help App.vue - VueReactionTimerProject
main.js HelloWorld.vue App.vue M MyGame.vue U index.js M MyNav.v

src > App.vue > {} "App.vue" > template > MyNav
You, 2 minutes ago | 2 authors (p00149021@houstonisd.org Mag17615@7 and others)
1 <template>
2 <MyNav /> You, 56 minutes ago • Add Nav- Bootstrap and routing ...
3 </template>
4
5 <script>
6 import MyNav from './components/nav/MyNav.vue';
7
8 export default {
9   name: 'App',
10  components: {
11    MyNav
12  },
13
14 }
15 </script>
16
17 <style>
18 #app {
19   font-family: Avenir, Helvetica, Arial, sans-serif;
20   -webkit-font-smoothing: antialiased;
21   -moz-osx-font-smoothing: grayscale;
22   text-align: center;
23   color: #444;
24   margin-top: 60px;
25 }
26 /*
27 button {
28   background: #0faf87;
```

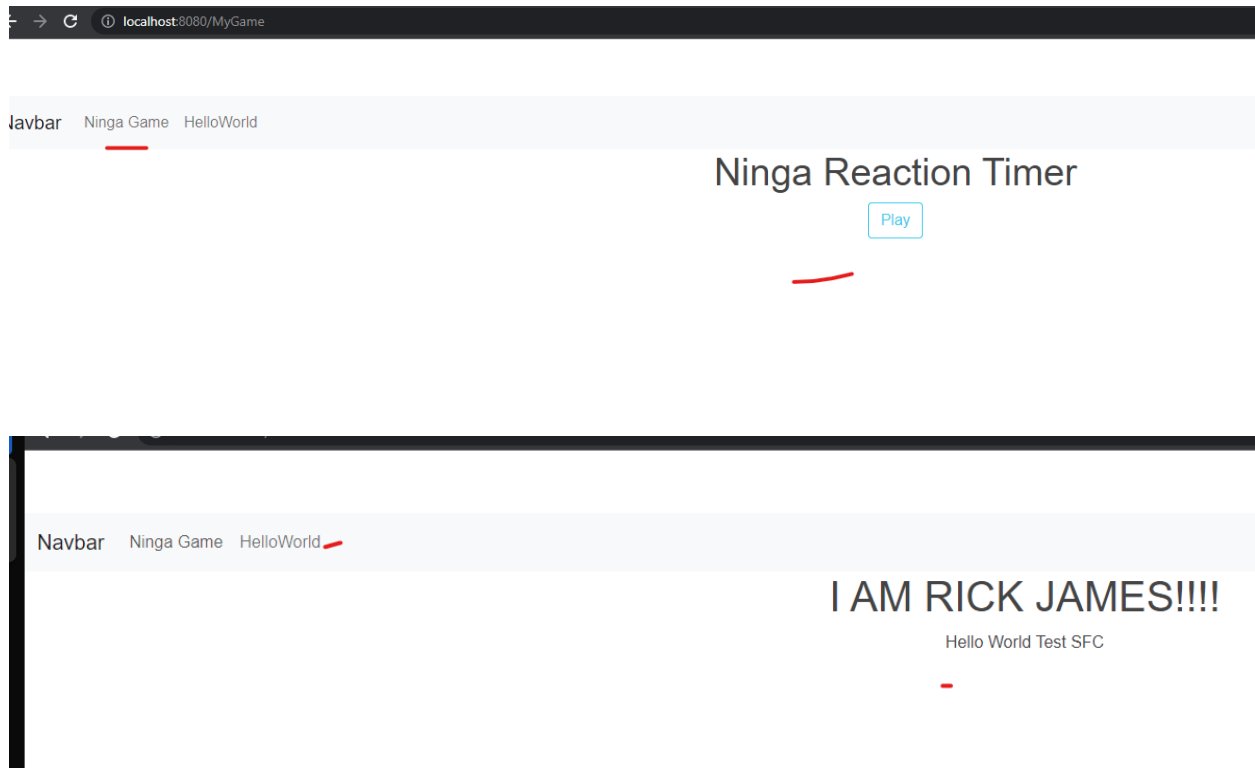
```
File Edit Selection View Go Run Terminal Help MyGame.vue - VueReactionTimerProject - Visual Studio Code
EXPLORER
VUEREACTIONTIMERPROJECT
  node_modules
  public
  src
    assets
    components
      nav
        MyNav.vue M
        HelloWorld.vue
        MyBlock.vue
        MyGame.vue U
        MyResults.vue
    router
      index.js M
      App.vue M
      main.js
  .gitignore
  ~$ewNotes-React... U
  babel.config.js
  jsconfig.json
  package-lock.json
  package.json

src > components > MyGame.vue > {} "MyGame.vue" > script > default
1 <template>
2 <h1>Ninga Reaction Timer</h1>
3 <button
4   class="btn btn-outline-info"
5   @click="start"
6   :disabled="isPlaying"
7 >Play</button>
8
9 <MyBlock
10   v-if="isPlaying"
11   :delay="delay"
12   @end="endGame"/>
13
14 <MyResults
15   v-if="showResults"
16   :score="score"
17 />
18 </template>
19
20 <script>
21 import MyBlock from './MyBlock.vue';
22 import MyResults from './MyResults.vue';
23 export default {
24   name: 'MyGame',
```

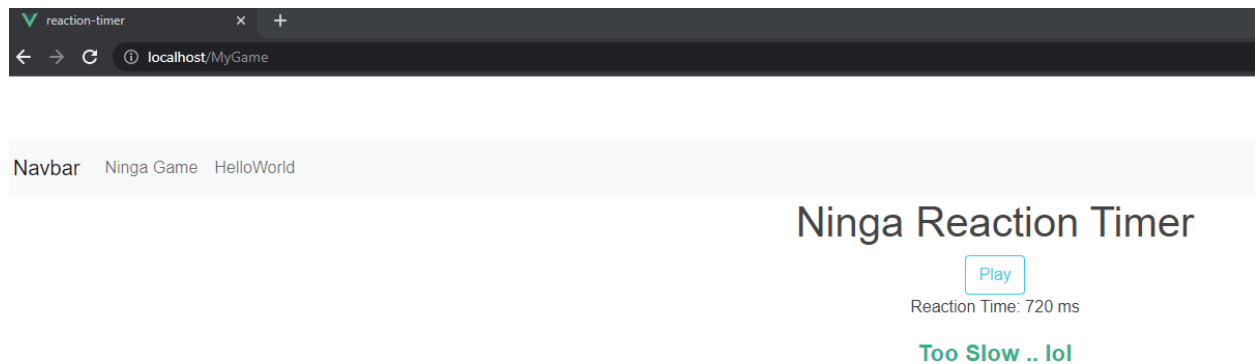
```
main.js HelloWorld.vue App.vue M MyGame.vue U index.js M X Myt
src > router > index.js > routes
You, 1 second ago | 1 author (You)
1 import { createWebHistory, createRouter } from 'vue-router';
2 import HelloWorld from '../components/HelloWorld.vue';
3 import MyGame from '../components/MyGame.vue';
4
5 const routes = [
6   {
7     path: '/HelloWorld',
8     name: 'HelloWorld',
9     component: HelloWorld,
10   },
11   {
12     path: '/MyGame',
13     name: 'MyGame',
14     component: MyGame,
15   },
16 ];
17
18 const router = createRouter({
19   history: createWebHistory(),
20   routes,
21 });
22
23 export default router;
24
```

localhost:8080

Navbar Ninga Game HelloWorld



Tip: Deployment and routing
When I deployed the app, the routing worked



Also the addition to vue.config also fixed the deployment issue with the paths as well


```
Go Run Terminal Help vue.config.js - VueReactionTimerProject - Visual Studio Code
main.js x HelloWorld.vue App.vue Release Notes: 1.69.0 vue.config.js x MyGame.vue
vue.config.js > ...
p00149021@houstonisd.org Mag17615@7, 2 days ago | 1 author (p00149021@houstonisd.org Mag17615@7)
1 const { defineConfig } = require('@vue/cli-service')
2 module.exports = defineConfig({
3   transpileDependencies: true,
4   publicPath: ''
5 })
6
```

Tip: Vuex (State Management) in Vue

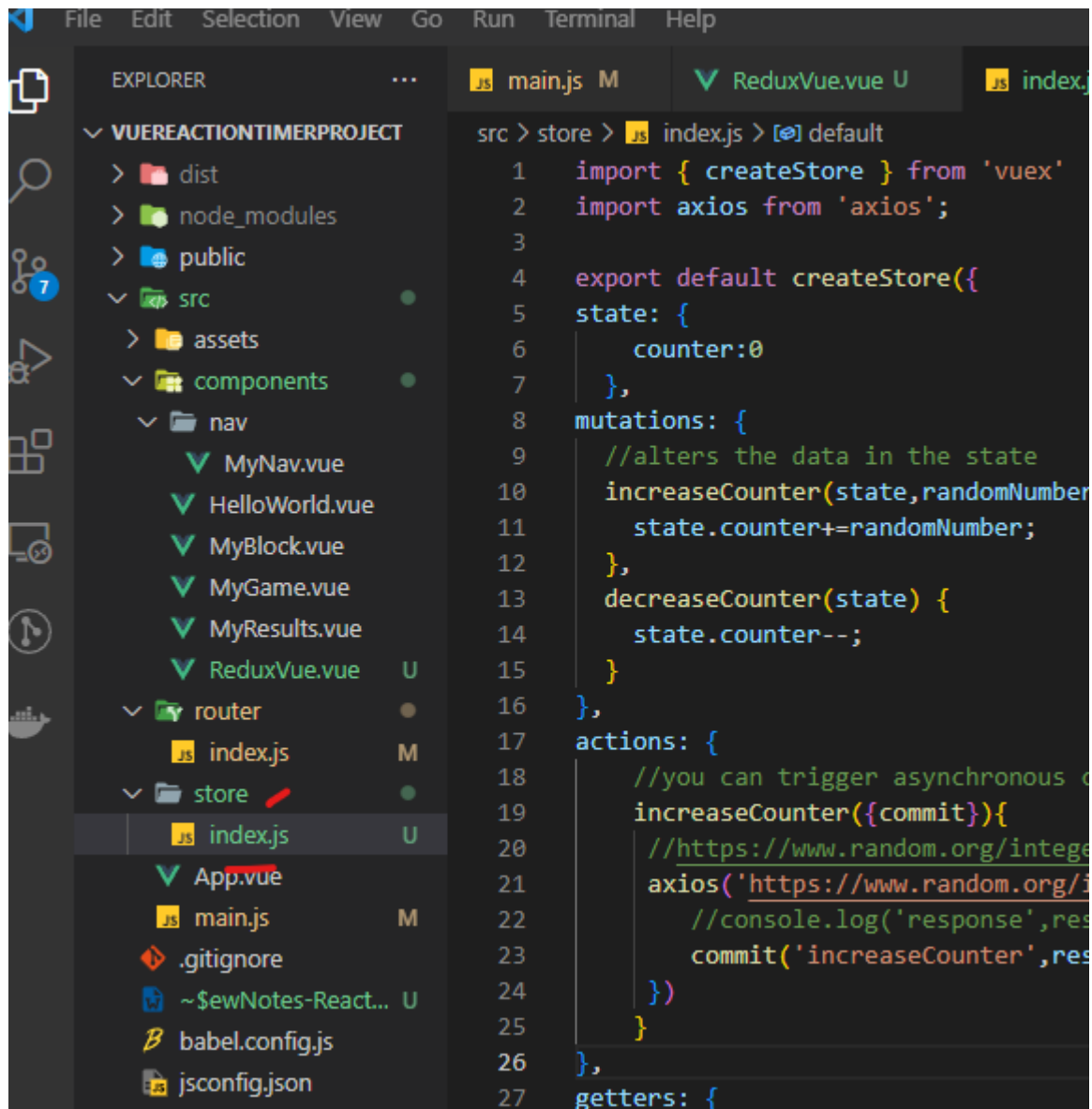
This came from this video

<https://www.youtube.com/watch?v=nFh7-HfODYY>

To add vuex to your project

npm install vuex@next --save

Then we can create our store



To add our store, we make a reference in our main.js file

```
src > JS main.js
You, 30 minutes ago | 2 authors (You and others)
1  import { createApp } from 'vue'
2  import App from './App.vue'
3  import router from './router/index';
4  import store from './store';
5
6
7  import "bootstrap/dist/css/bootstrap.min.css"
8  import "bootstrap"
9
10
11 createApp(App) You, 30 minutes ago • Uncommitted changes
12   .use(router)
13   .use(store)
14   .mount('#app')
15
```

Now every component in our application has access to our store

We will demonstrate two types of actions in our store

```
main.js M  ReduxVue.vue U X  index.js U
src > components > ReduxVue.vue > {} "ReduxVue.vue" > script > default > name
1  <template>
2    <div>
3      
4      <div class="counter">
5        {{ $store.state.counter }}
6      </div>
7    </div>
8    <div class="buttons">
9      <button @click="$store.dispatch('increaseCounter')">+</button>
10     <button @click="$store.commit('decreaseCounter')">-</button>
11   </div>
12 </template>
13
14 <script>
15 export default {
16   name: 'ReduxVue'
17 }
18 </script>
19
20 <style>
21 div {
22   margin-bottom: 10px;
23 }
24 .counter {
25   font-size: 80px;
26 }
27 .buttons button {
28   font-size: 40px;
29   width: 100px;
30   margin: 0 10px;
31 }
32 </style>
```

commit: allows to fire an action in your store

dispatch: allows you to fire an action in our store

To fetch data from our store we use the `$store.state.<property>`

```
main.js M  ReduxVue.vue U  index.js U x
src > store > index.js > default
1  import { createStore } from 'vuex'
2  import axios from 'axios';
3
4  export default createStore({
5    state: {
6      counter:0
7    },
8    mutations: {
9      //alters the data in the state
10     increaseCounter(state,randomNumber) {
11       state.counter+=randomNumber;
12     },
13     decreaseCounter(state) {
14       state.counter--;
15     }
16   },
17   actions: {
18     //you can trigger asynchronous code here (api) - dispatch actions
19     increaseCounter({commit}){
20       //https://www.random.org/integers/?num=10&min=1&max=6&col=1&base=10&format=plain&rnd=new
21       axios('https://www.random.org/integers/?num=1&min=1&max=6&col=1&base=10&format=plain&rnd=new').then(response => {
22         //console.log('response',response)
23         commit('increaseCounter',response.data)
24       })
25     }
26   },
27   getters: {
28     //you can change or filter some data in your state
29   },
30   modules: {
31     //allows you break up your vuex store
32   }
33 })
```

Like stated above to change some state in our store, we can use the mutations section to fire a method the update state.

The actions section allows us to fire asynchronous code (api) and change a state's property (Tip: Using axios in vue)

npm install axios

Tip: Running unit tests with jest and vue tools

Website for unit test examples

<https://v2.vuejs.org/v2/cookbook/unit-testing-vue-components.html?#Why-test>

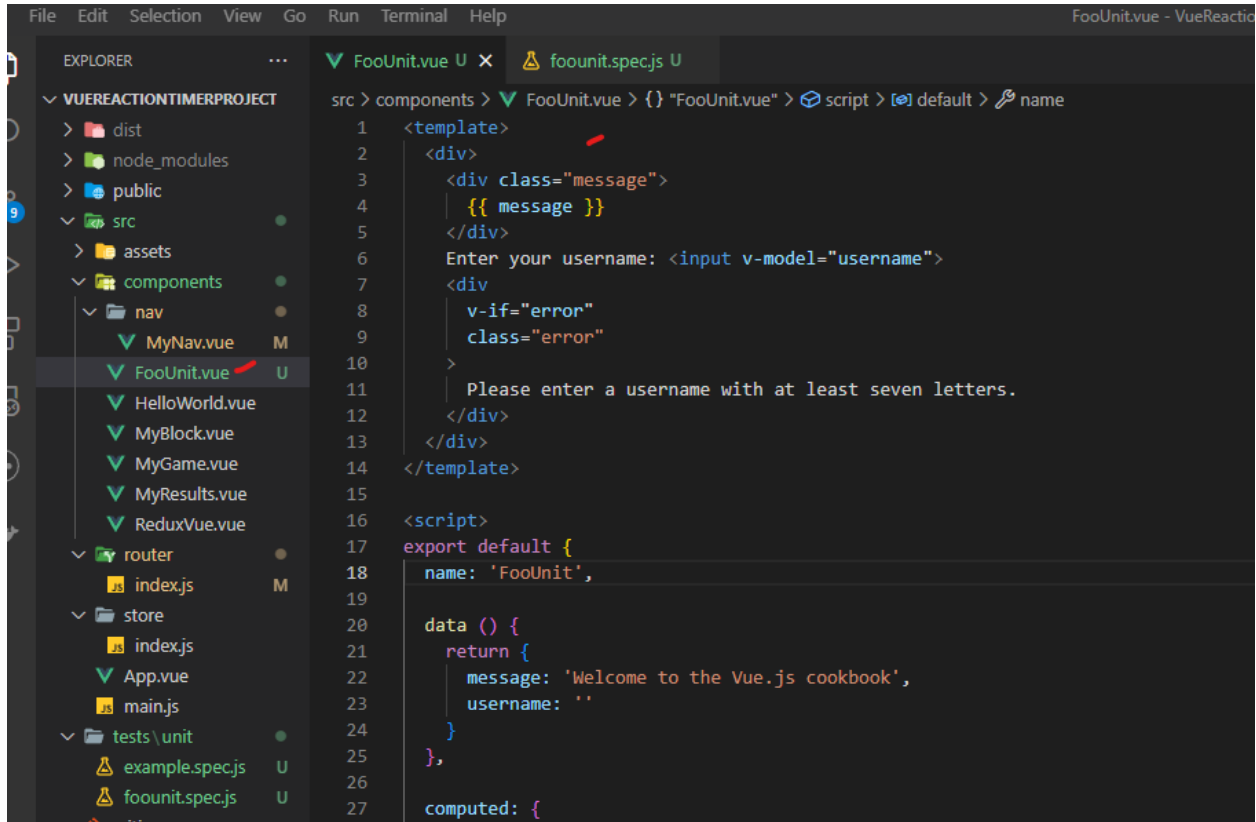
Run these commands to set up unit testing

<https://v1.test-utils.vuejs.org/installation/#using-vue-test-utils-with-jest-recommended>

```
vue add unit-jest
```

```
npm install --save-dev @vue/test-utils
```

I created a component to unit test



The screenshot shows a code editor with two files open: `FooUnit.vue` and `foounit.spec.js`. The `FooUnit.vue` file contains a template with a message display, a username input field, and an error message. The `foounit.spec.js` file contains a Jest unit test for the component.

```
1 <template>
2   <div>
3     <div class="message">
4       {{ message }}
5     </div>
6     Enter your username: <input v-model="username">
7     <div
8       v-if="error"
9       class="error"
10    >
11      Please enter a username with at least seven letters.
12    </div>
13  </div>
14 </template>
15
16 <script>
17 export default {
18   name: 'FooUnit',
19
20   data () {
21     return {
22       message: 'Welcome to the Vue.js cookbook',
23       username: ''
24     }
25   },
26
27   computed: {
```

I added some test code

```
to Run Terminal Help foounit.spec.js - VueReactionTimerProject - Visual Studio Code
V FooUnit.vue U foounit.spec.js U X
tests > unit > foounit.spec.js > describe('Foo') callback > it('renders a message and responds correctly to user input') callback
1 import { shallowMount } from '@vue/test-utils'
2 import FooUnit from '../../src/components/FooUnit.vue';
3
4 describe('Foo', () => {
5   it('renders a message and responds correctly to user input', () => {
6     const wrapper = shallowMount(FooUnit, {
7       data() {
8         return {
9           message: 'Hello World',
10          username: ''
11        }
12      }
13    })
14
15    // see if the message renders
16    expect(wrapper.find('.message').text()).toEqual('Hello World')
17
18    // assert the error is rendered
19    expect(wrapper.find('.error').exists()).toBeTruthy()
20
21    // update the `username` and assert error is no longer rendered
22    // wrapper.setData({ username: 'lionel5' })
23    // expect(wrapper.find('.error').exists()).toBeFalsy()
24  })
25 })
```

Ran the test command to test one component
npm run test:unit foounit

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS
PS C:\DevProjects\Programming\vueProjects\VueReactionTimerProject> npm run test:unit foounit

> reaction-timer@0.1.0 test:unit C:\DevProjects\Programming\vueProjects\VueReactionTimerProject
> vue-cli-service test:unit "foounit"

PASS tests/unit/foounit.spec.js
  Foo
    ✓ renders a message and responds correctly to user input (27 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.135 s
Ran all test suites matching /foounit/i.
PS C:\DevProjects\Programming\vueProjects\VueReactionTimerProject> 
```