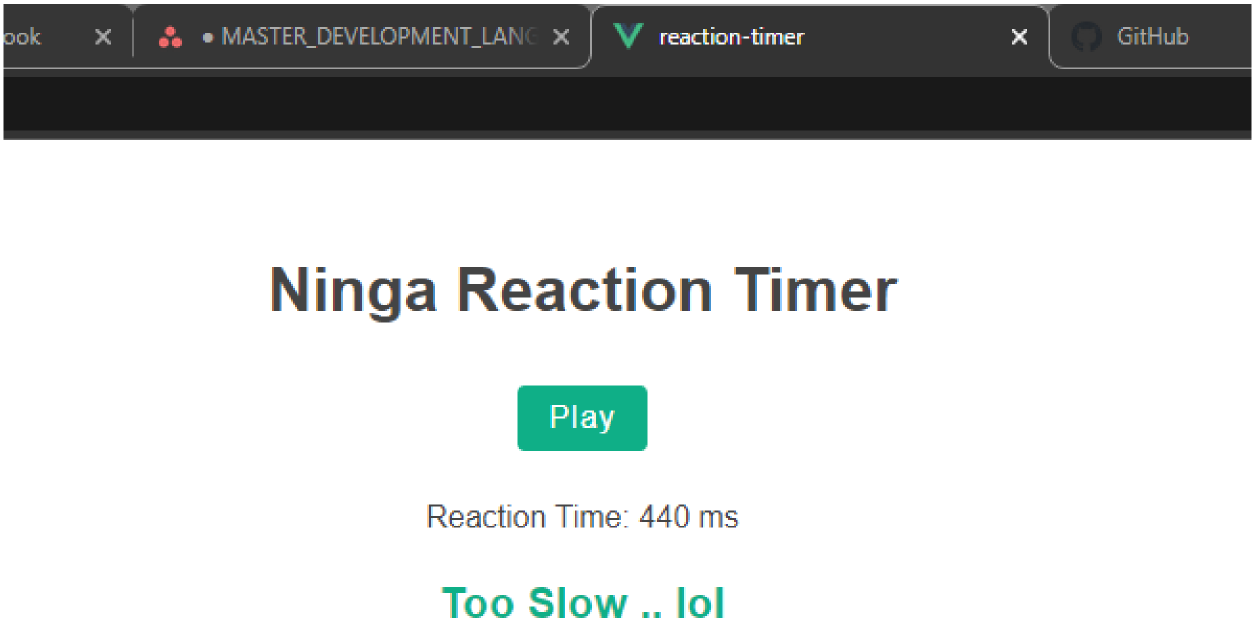
These are other notes on View from the Reaction-Timer Project



Ninga Reaction Timer

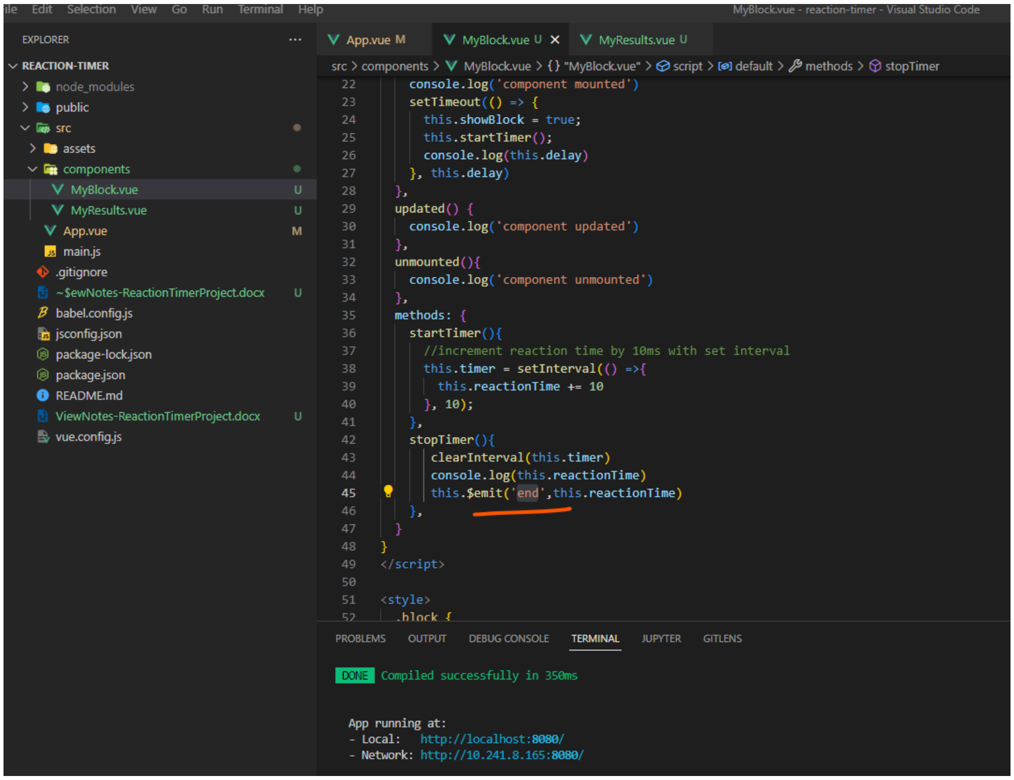
Reaction Time: 440 ms

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

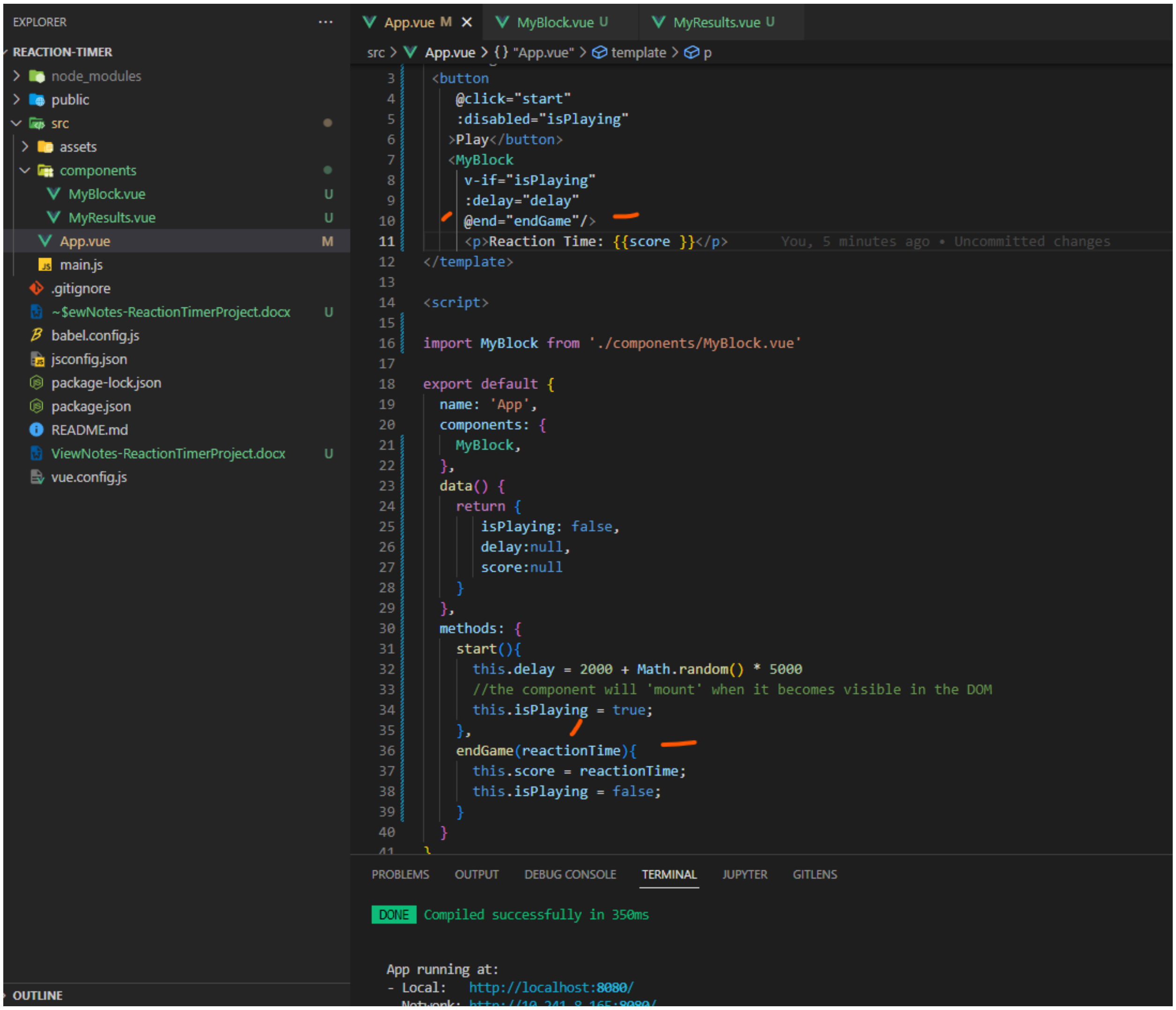
**Tip**: Another tip on emitting methods
If you look below, we are emitting a method that we will name ‘end’ in the parent component. Notice how we are passing data along with it.



```
    console.log('component mounted')
    setTimeout(() => {
      this.showBlock = true;
      this.startTimer();
      console.log(this.delay)
    }, this.delay)
  },
  updated() {
    console.log('component updated')
  },
  unmounted(){
    console.log('component unmounted')
  },
  methods: {
    startTimer(){
      //increment reaction time by 10ms with set interval
      this.timer = setInterval(() =>{
        this.reactionTime += 10
      }, 10);
    },
    stopTimer(){
      clearInterval(this.timer)
      console.log(this.reactionTime)
      this.$emit('end',this.reactionTime)
    },
  }
}
</script>

<style>
  .block {
```

```
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.

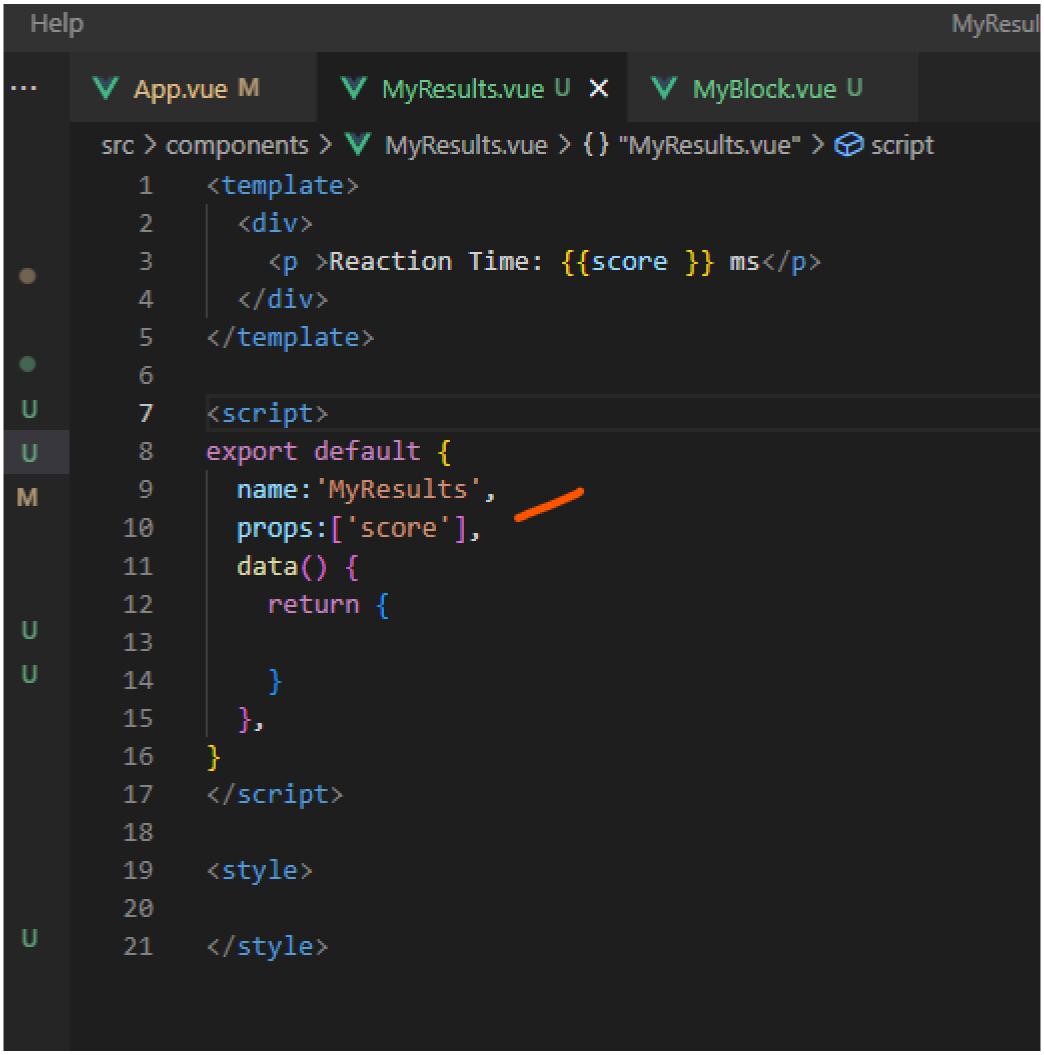


```
  <button
    @click="start"
    :disabled="isPlaying"
   >Play</button>
   <MyBlock
     v-if="isPlaying"
     :delay="delay"
     @end="endGame"/>
     <p>Reaction Time: {{score }}</p>        You, 5 minutes ago • Uncommitted changes
</template>

<script>

import MyBlock from './components/MyBlock.vue'

export default {
  name: 'App',
  components: {
    MyBlock,
  },
  data() {
    return {
      isPlaying: false,
      delay:null,
      score:null
    }
  },
  methods: {
    start(){
      this.delay = 2000 + Math.random() * 5000
      //the component will 'mount' when it becomes visible in the DOM
      this.isPlaying = true;
    },
    endGame(reactionTime){
      this.score = reactionTime;
      this.isPlaying = false;
    }
  }
}
```

```
DONE Compiled successfully in 350ms


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

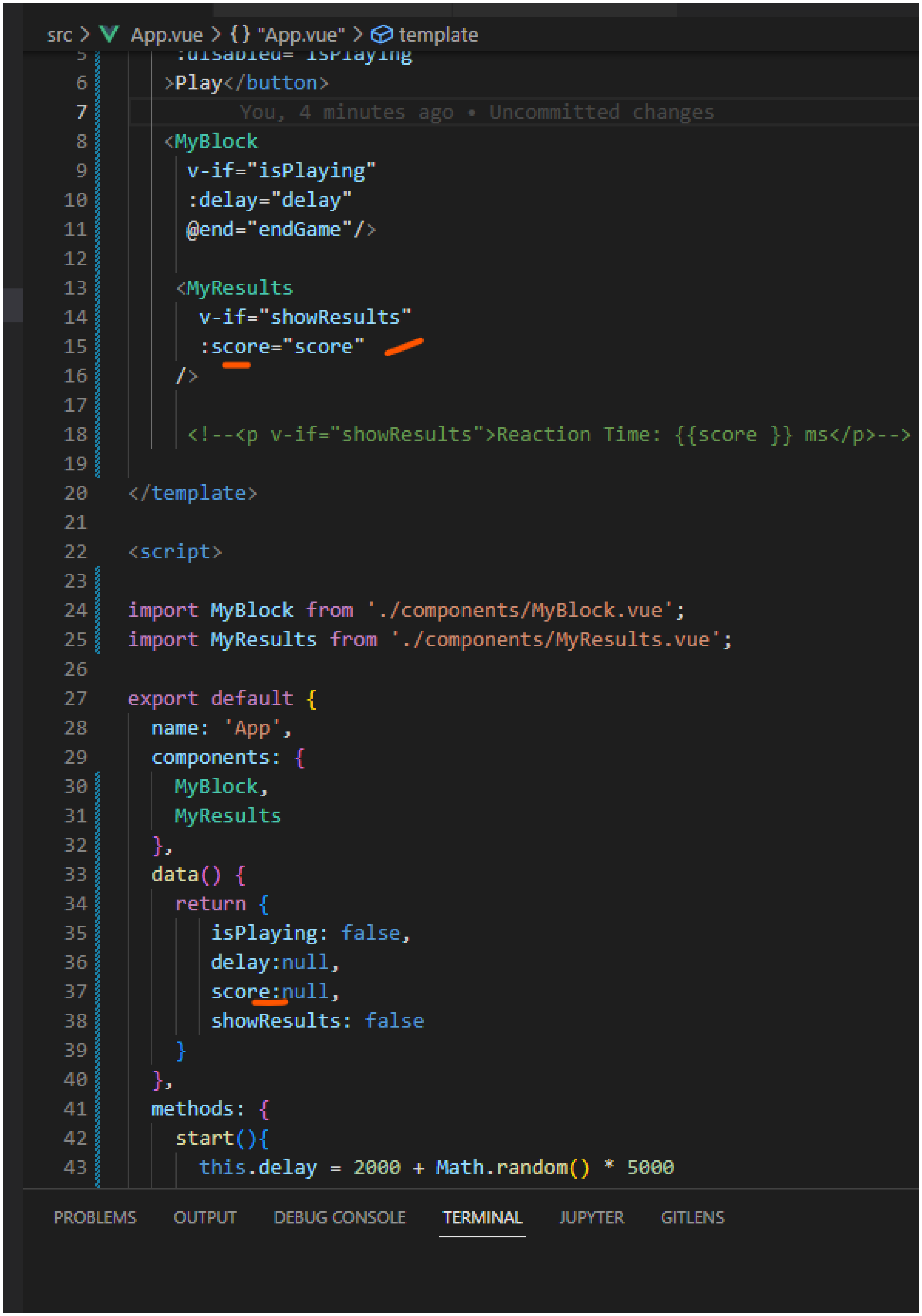
**Tip**: Passing props (just a refresher)

We declare props


Help
MyResul
App.vue M
MyResults.vue U
MyBlock.vue U
src > components > MyResults.vue > {} "MyResults.vue" > script

```
<template>
  <div>
    <p >Reaction Time: {{score }} ms</p>
  </div>
</template>

<script>
export default {
  name:'MyResults',
  props:['score'],
  data() {
    return {

    }
  },
}
</script>

<style>

</style>
```

Then to receive the prop from the parent:



```
    :disabled="isPlaying"
   >Play</button>
        You, 4 minutes ago • Uncommitted changes
   <MyBlock
     v-if="isPlaying"
     :delay="delay"
     @end="endGame"/>

    <MyResults
      v-if="showResults"
      :score="score"
    />

     <!--<p v-if="showResults">Reaction Time: {{score }} ms</p>-->

</template>

<script>

import MyBlock from './components/MyBlock.vue';
import MyResults from './components/MyResults.vue';

export default {
  name: 'App',
  components: {
    MyBlock,
    MyResults
  },
  data() {
    return {
       isPlaying: false,
       delay:null,
       score:null,
       showResults: false
    }
  },
  methods: {
    start(){
      this.delay = 2000 + Math.random() * 5000
```

**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/

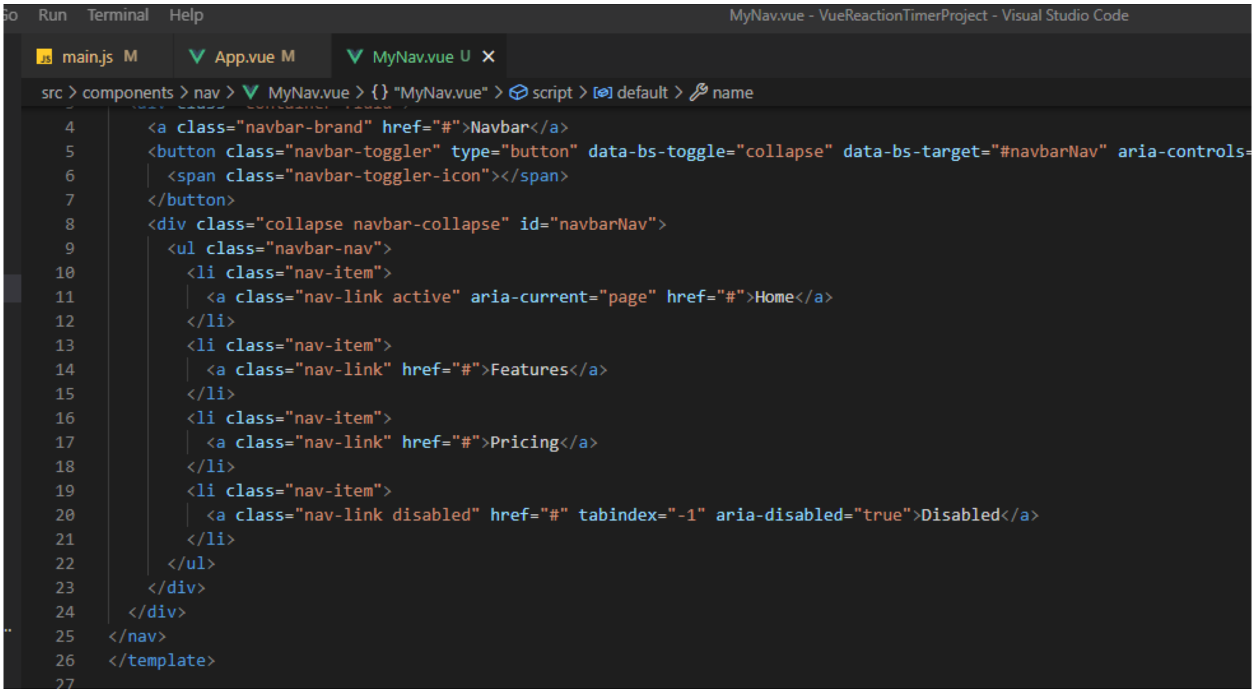
npm i bootstrap@5.0.1

npm install --save @popperjs/core

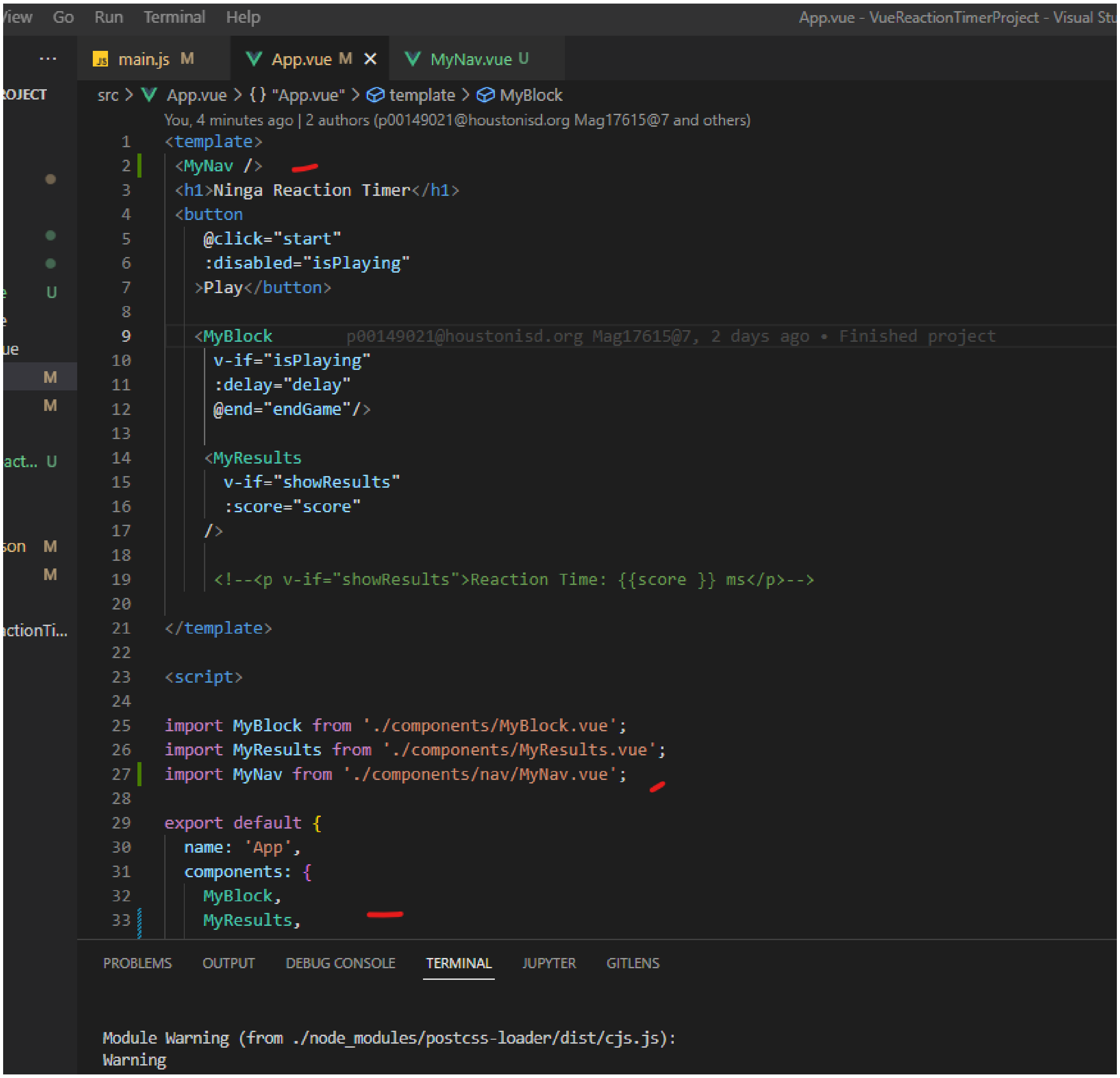
Then In main.js

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

Create a nav component

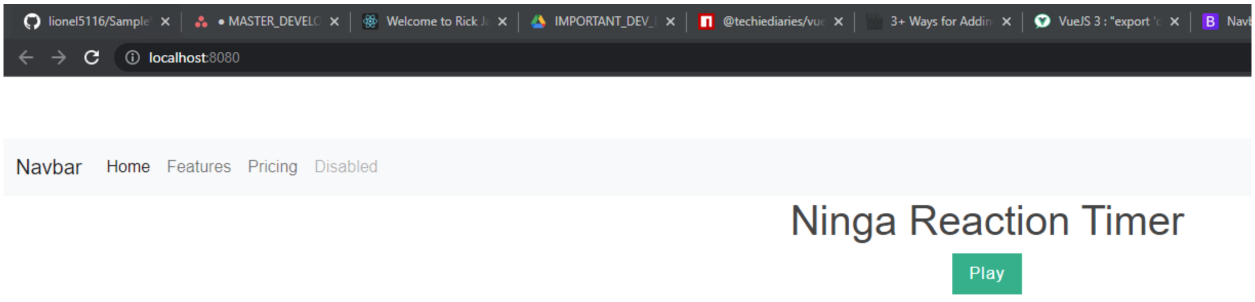

Go Run Terminal Help
MyNav.vue - VueReactionTimerProject - Visual Studio Code
main.js M
App.vue M
MyNav.vue U
src > components > nav > MyNav.vue > {} "MyNav.vue" > script > default > name

```
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls=
      <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">
          <a class="nav-link" href="#">Features</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Pricing</a>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
</template>
```

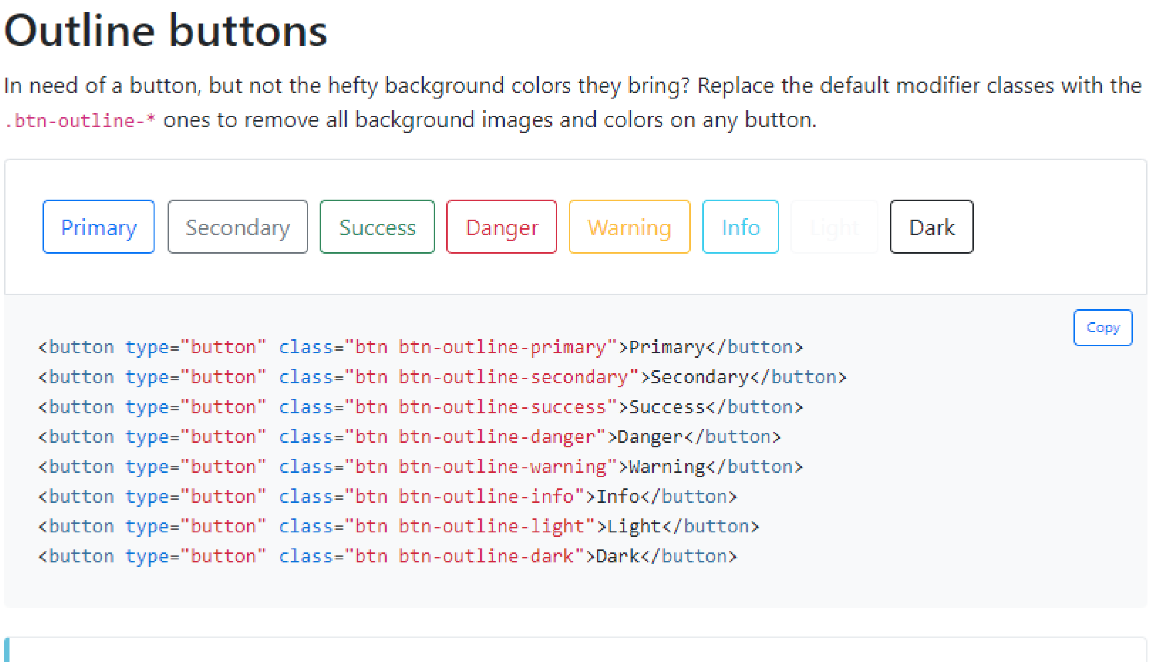

View Go Run Terminal Help
App.vue - VueReactionTimerProject - Visual Stu
main.js M
App.vue M
MyNav.vue U
src > App.vue > {} "App.vue" > template > MyBlock
You, 4 minutes ago | 2 authors (p00149021@houstonisd.org Mag17615@7 and others)
p00149021@houstonisd.org Mag17615@7, 2 days ago • Finished project
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
GITLENS

```
<template>
 <MyNav />
 <h1>Ninga Reaction Timer</h1>
 <button
    @click="start"
    :disabled="isPlaying"
   >Play</button>

   <MyBlock
     v-if="isPlaying"
     :delay="delay"
     @end="endGame"/>

    <MyResults
      v-if="showResults"
      :score="score"
    />

     <!--<p v-if="showResults">Reaction Time: {{score }} ms</p>-->

</template>

<script>

import MyBlock from './components/MyBlock.vue';
import MyResults from './components/MyResults.vue';
import MyNav from './components/nav/MyNav.vue';

export default {
  name: 'App',
  components: {
    MyBlock,
    MyResults,
```

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



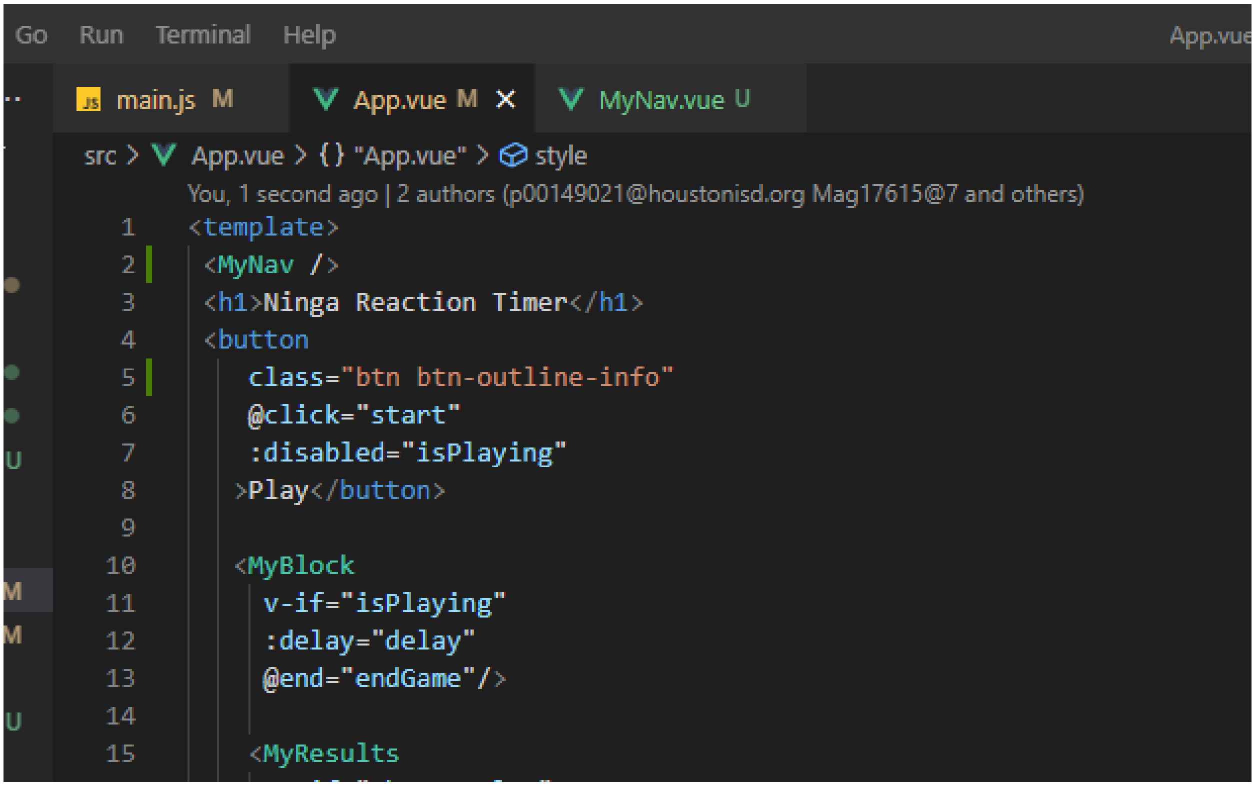
You can then use bootstrap anywhere in your application. Just refer to the bootstrap docs



# Outline buttons

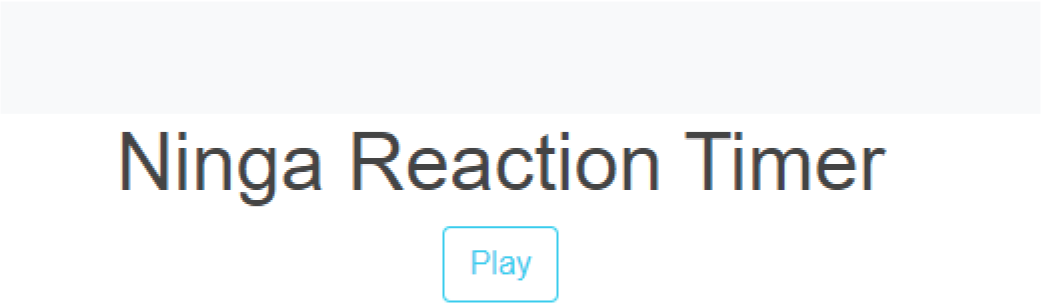
In need of a button, but not the hefty background colors they bring? Replace the default modifier classes with the `.btn-outline-*` ones to remove all background images and colors on any button.

```html
<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>
```

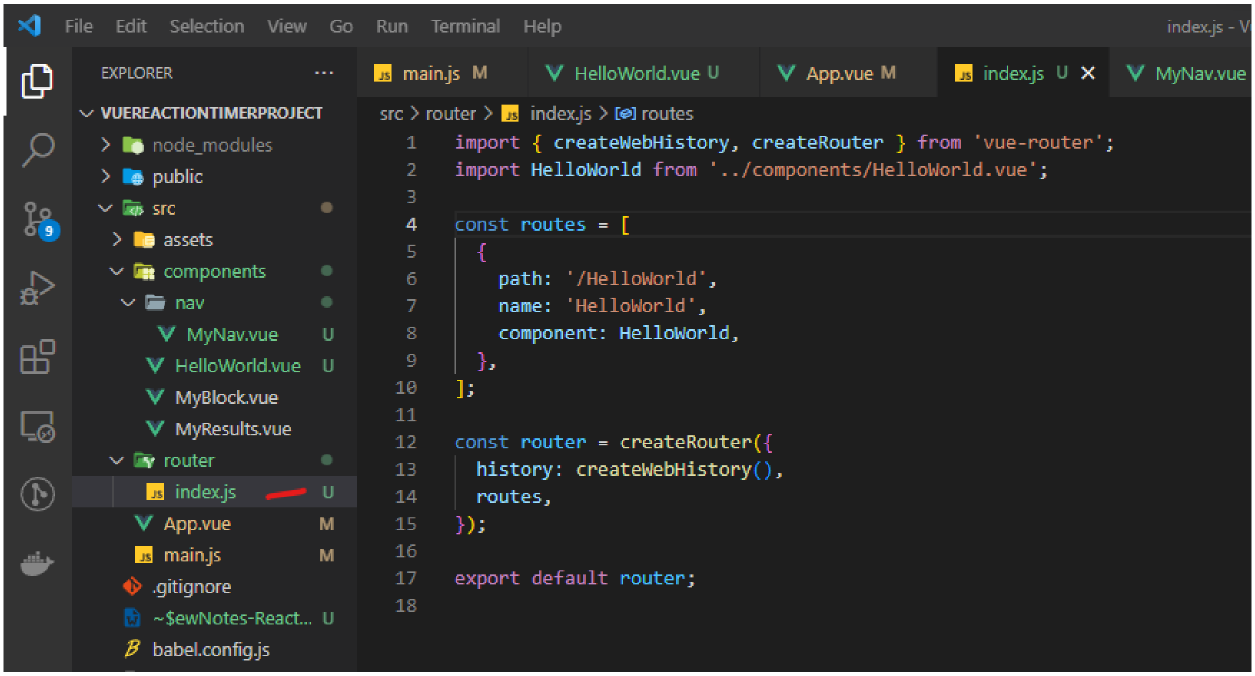


Go
Run
Terminal
Help
App.vue
main.js M
App.vue M
MyNav.vue U
src > App.vue > {} "App.vue" > style
You, 1 second ago | 2 authors (p00149021@houstonisd.org Mag17615@7 and others)

```html
<template>
 <MyNav />
 <h1>Ninga Reaction Timer</h1>
 <button
   class="btn btn-outline-info"
   @click="start"
   :disabled="isPlaying"
  >Play</button>

  <MyBlock
    v-if="isPlaying"
    :delay="delay"
    @end="endGame"/>

   <MyResults
```

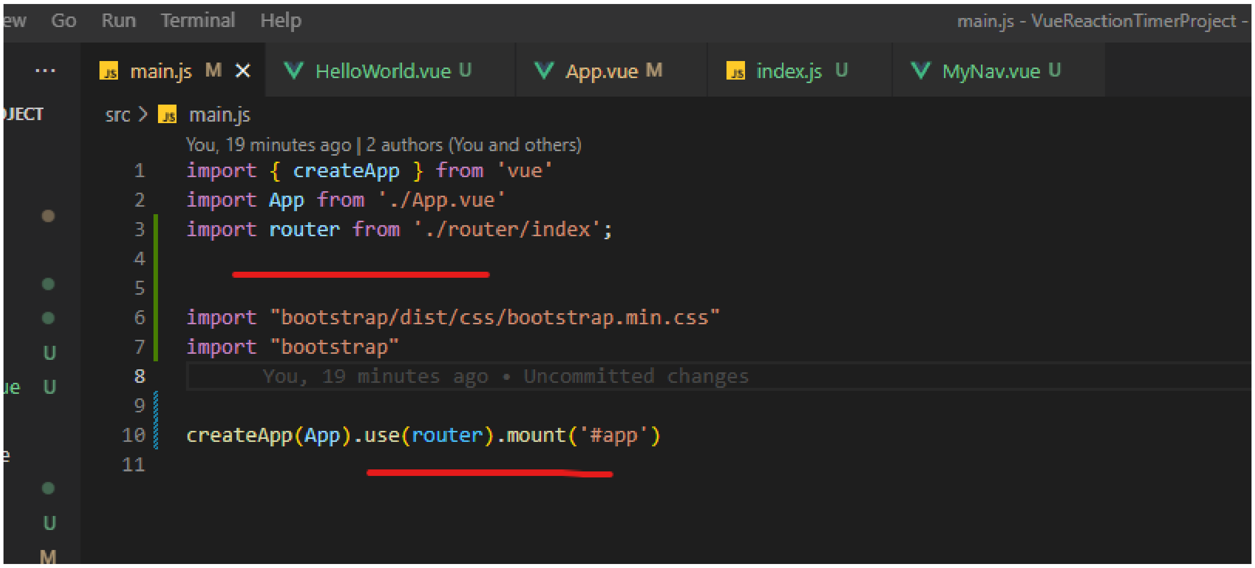


Tip: Adding routing in Vue 3
https://www.vuemastery.com/blog/vue-router-a-tutorial-for-vue-3/



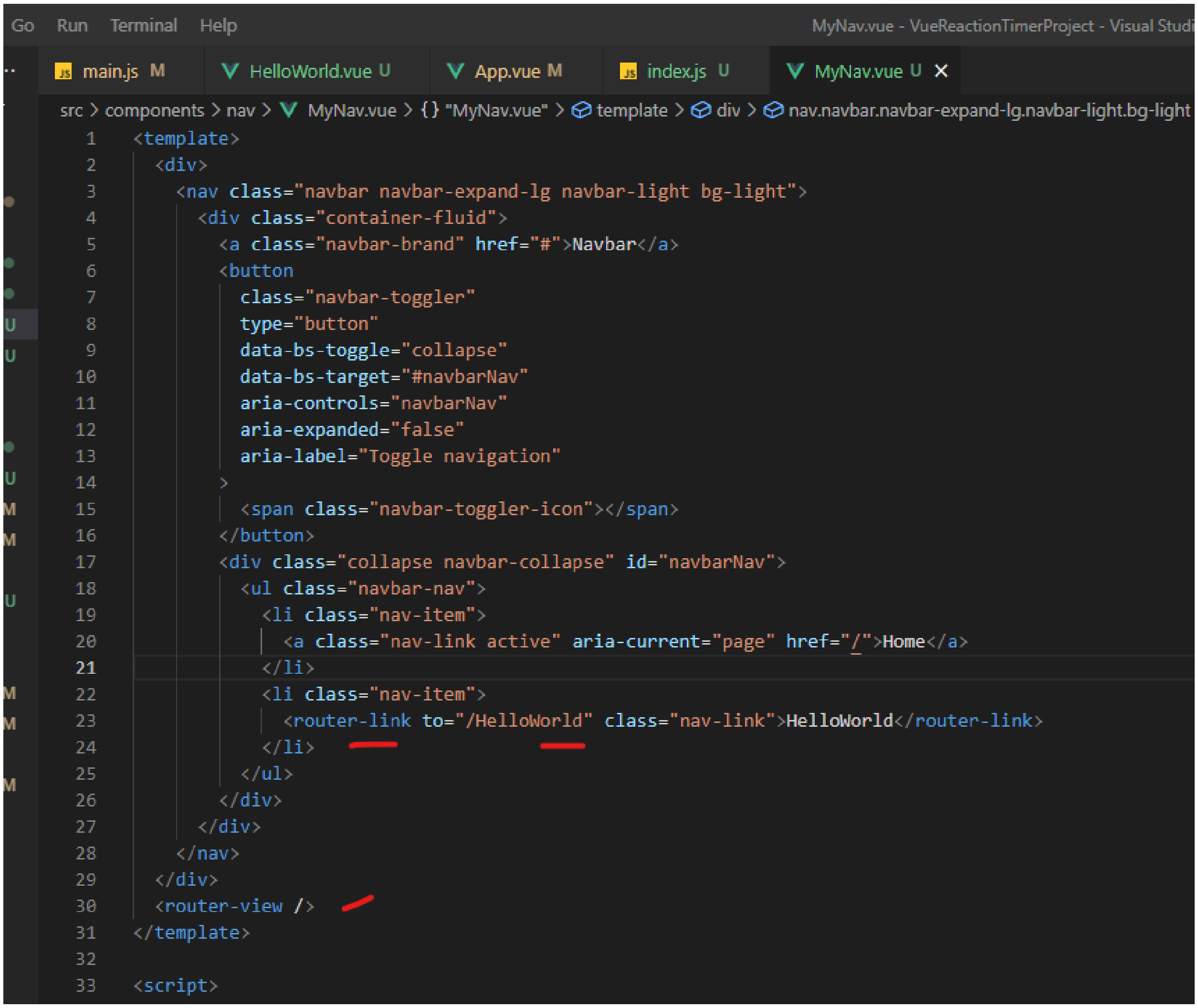
```
import { createWebHistory, createRouter } from 'vue-router';
import HelloWorld from '../components/HelloWorld.vue';

const routes = [
  {
    path: '/HelloWorld',
    name: 'HelloWorld',
    component: HelloWorld,
  },
];

const router = createRouter({
  history: createWebHistory(),
  routes,
});

export default router;
```

In main.js

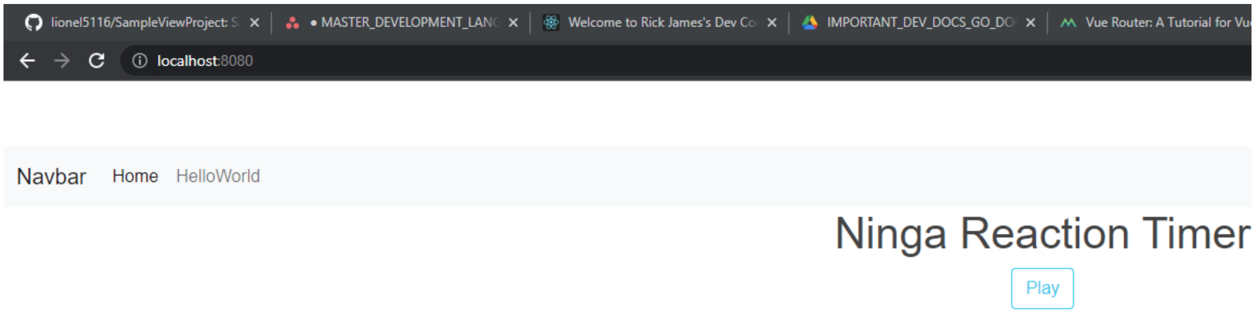


```
import { createApp } from 'vue'
import App from './App.vue'
import router from './router/index';


import "bootstrap/dist/css/bootstrap.min.css"
import "bootstrap"
        You, 19 minutes ago • Uncommitted changes

createApp(App).use(router).mount('#app')
```

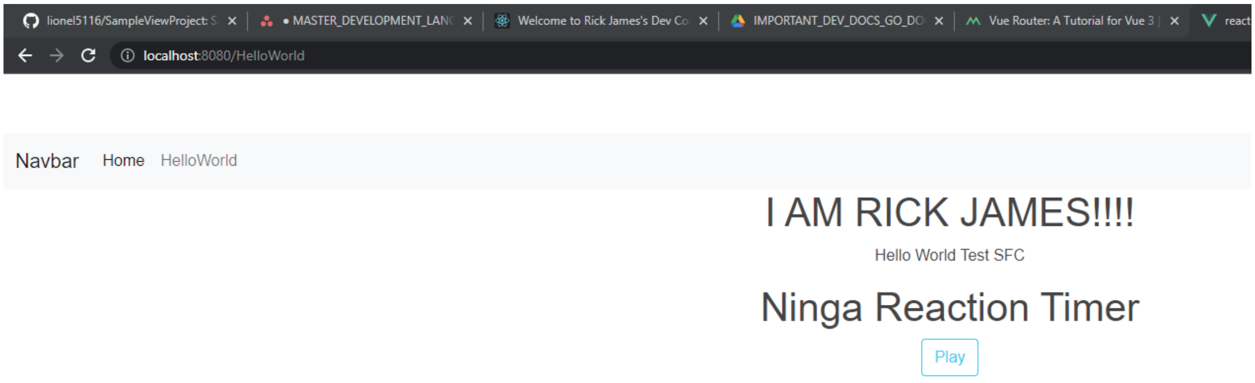
In your Custom Nav Component


Go Run Terminal Help
MyNav.vue - VueReactionTimerProject - Visual Studi
main.js M
HelloWorld.vue U
App.vue M
index.js U
MyNav.vue U
src > components > nav > MyNav.vue > {} "MyNav.vue" > template > div > nav.navbar.navbar-expand-lg.navbar-light.bg-light

```
<template>
  <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>
  <router-view />
</template>

<script>
```


lionel5116/SampleViewProject: S
• MASTER_DEVELOPMENT_LANG
Welcome to Rick James's Dev Co
IMPORTANT_DEV_DOCS_GO_DO
Vue Router: A Tutorial for Vu
localhost:8080
Navbar
Home
HelloWorld
Play

Ninga Reaction Timer

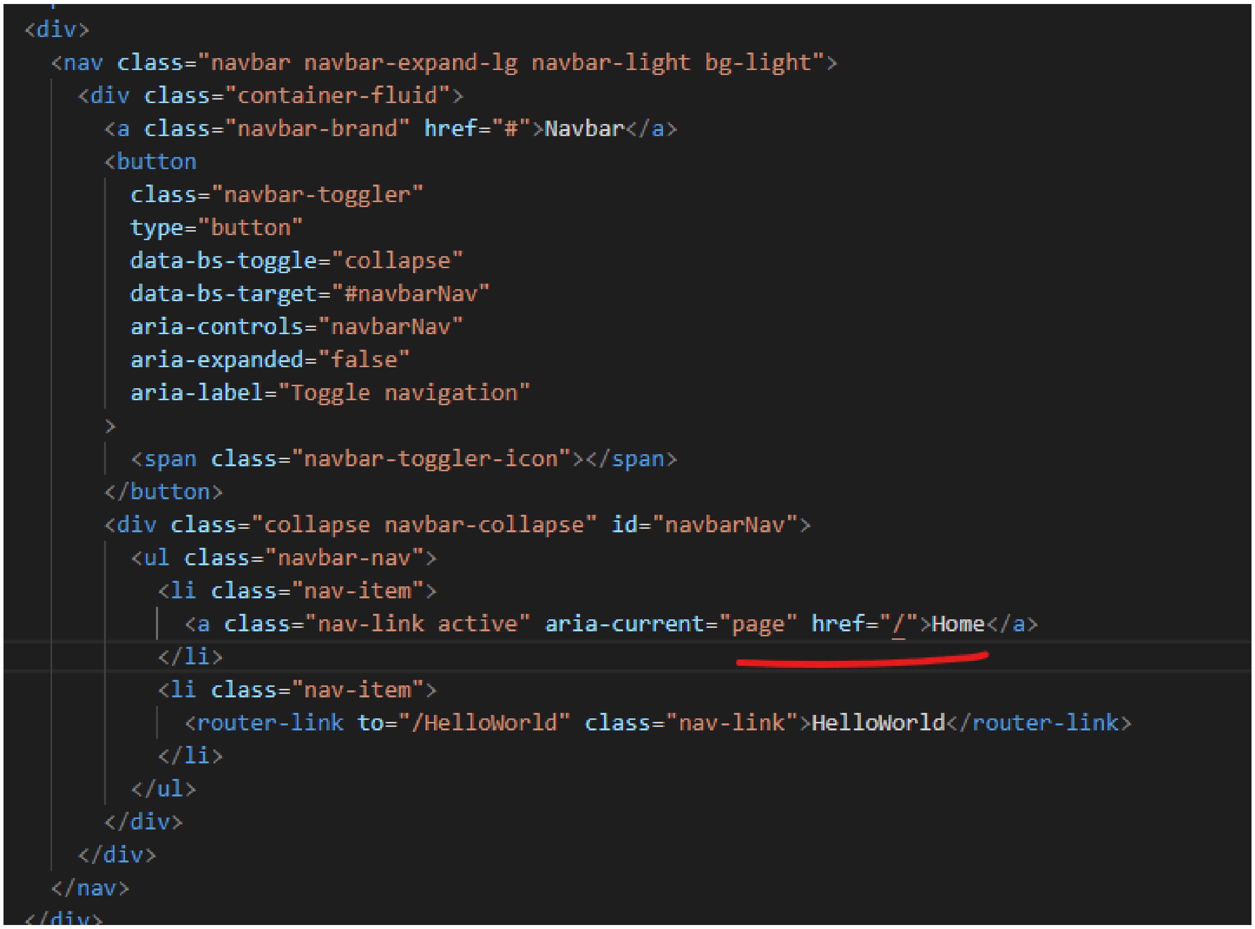

lionel5116/SampleViewProject: S
● MASTER_DEVELOPMENT_LANC
Welcome to Rick James's Dev Co
IMPORTANT_DEV_DOCS_GO_DO
Vue Router: A Tutorial for Vue 3 |
react
localhost:8080/HelloWorld
Navbar
Home
HelloWorld
Play

I AM RICK JAMES!!!!

Hello World Test SFC

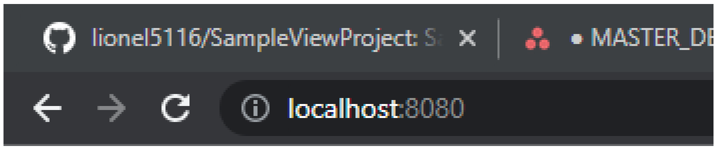
Ninga Reaction Timer

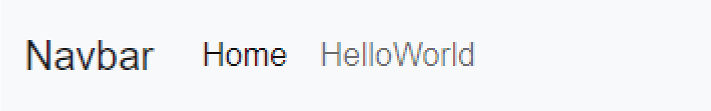
For the home link

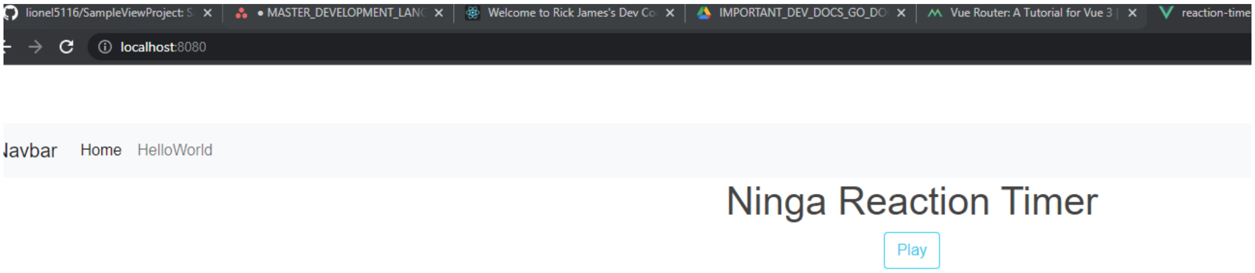


```
<div>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
      <a class="navbar-brand" href="#">Navbar</a>
      <button
        class="navbar-toggler"
        type="button"
        data-bs-toggle="collapse"
        data-bs-target="#navbarNav"
        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>
```

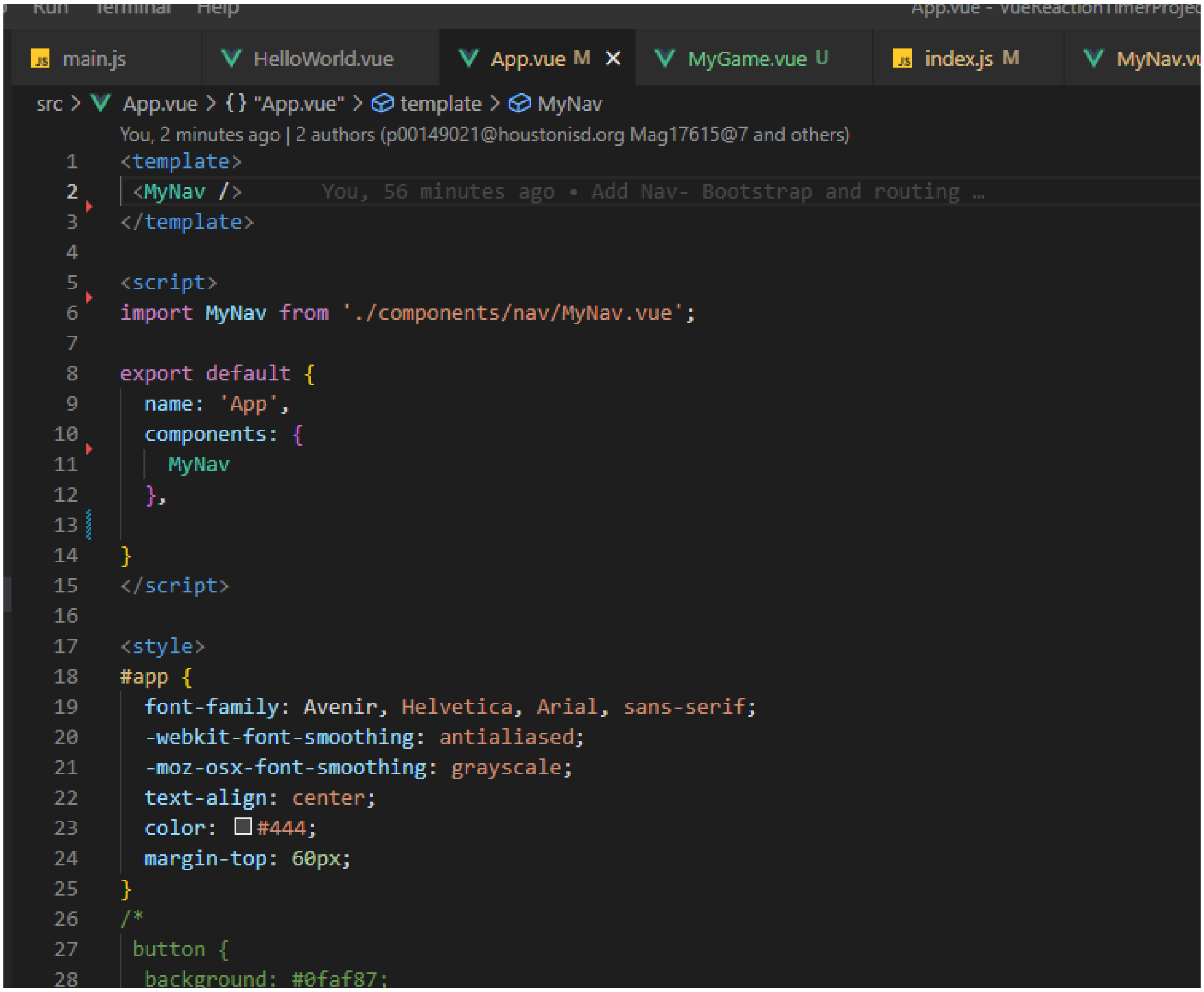
You don’t need a route



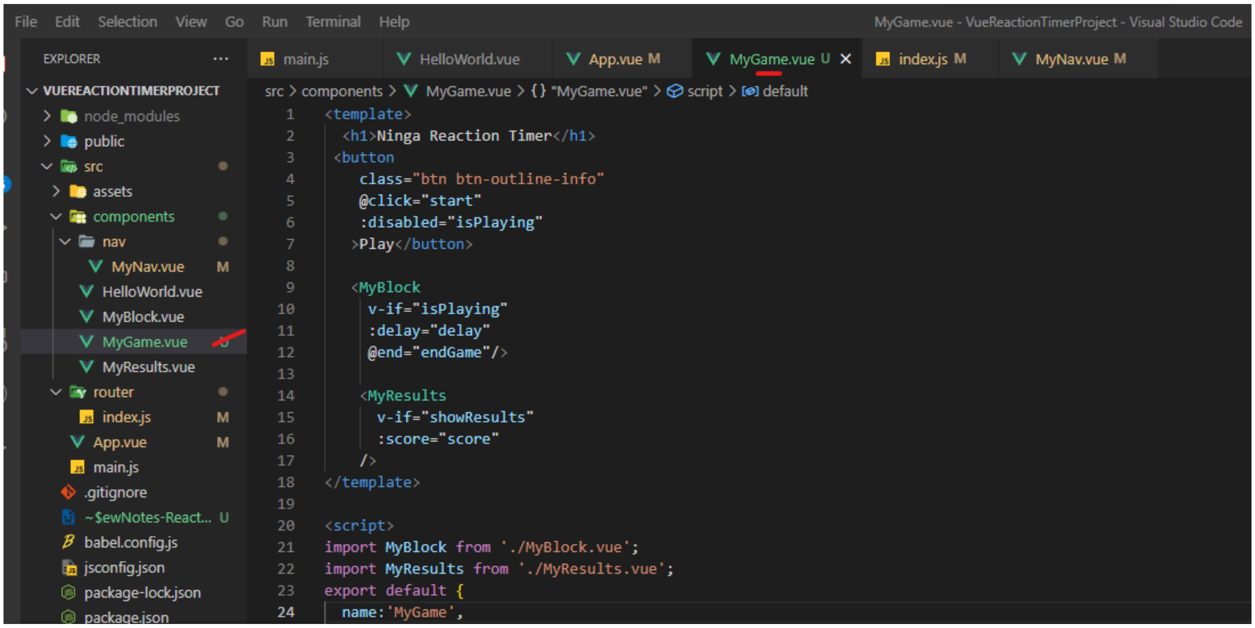




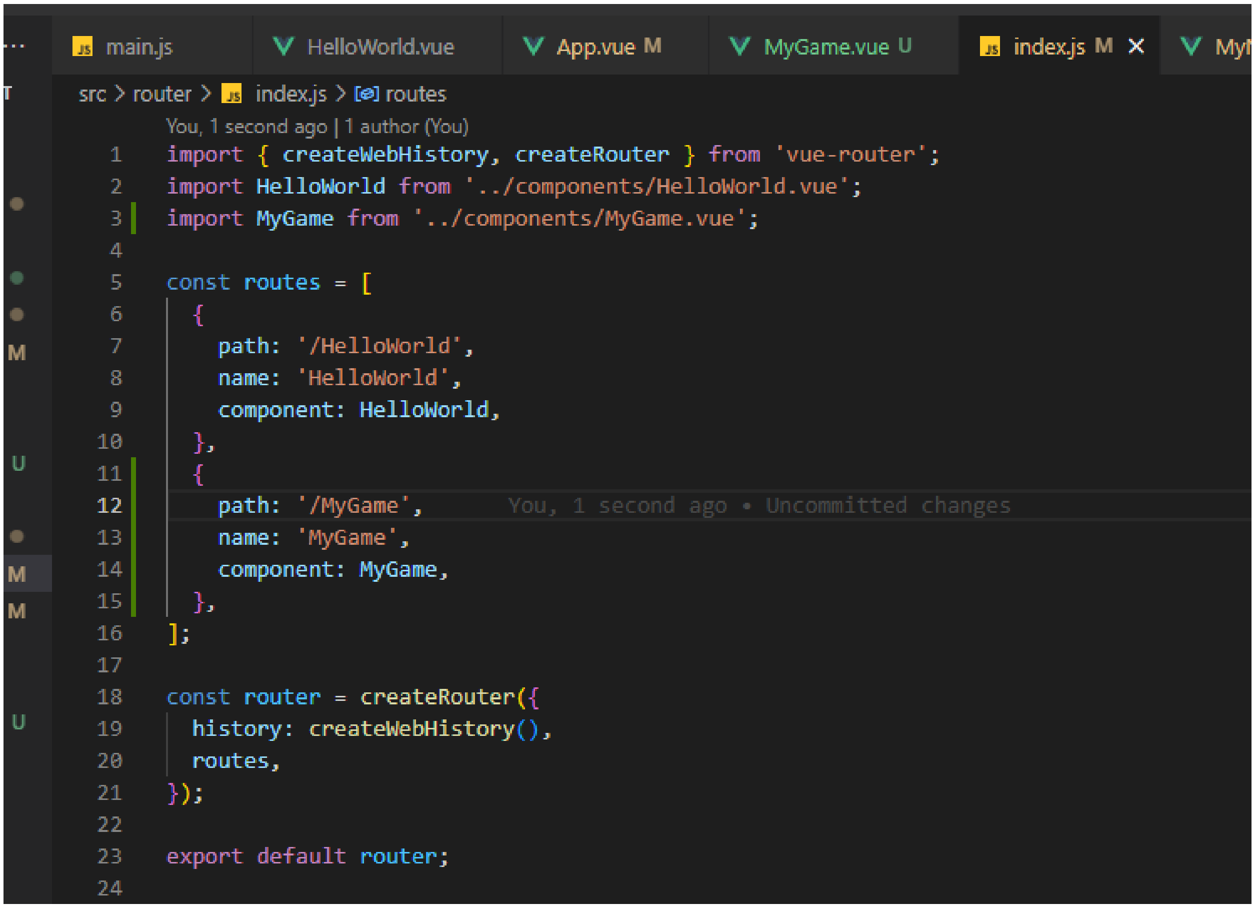
Tip: Refactoring the app for the routing and nav
I refactored the app to correct the navigation


Run Terminal Help
App.vue - VueReactionTimerProjec
main.js
HelloWorld.vue
App.vue M ×
MyGame.vue U
index.js M
MyNav.vu
src > App.vue > {} "App.vue" > template > MyNav
You, 2 minutes ago | 2 authors (p00149021@houstonisd.org Mag17615@7 and others)
You, 56 minutes ago • Add Nav- Bootstrap and routing …

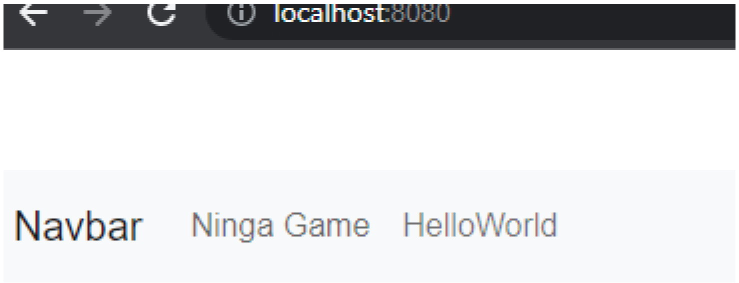
```
<template>
 <MyNav />
</template>

<script>
import MyNav from './components/nav/MyNav.vue';

export default {
  name: 'App',
  components: {
    MyNav
  },

}
</script>

<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #444;
  margin-top: 60px;
}
/*
 button {
  background: #0faf87;
```

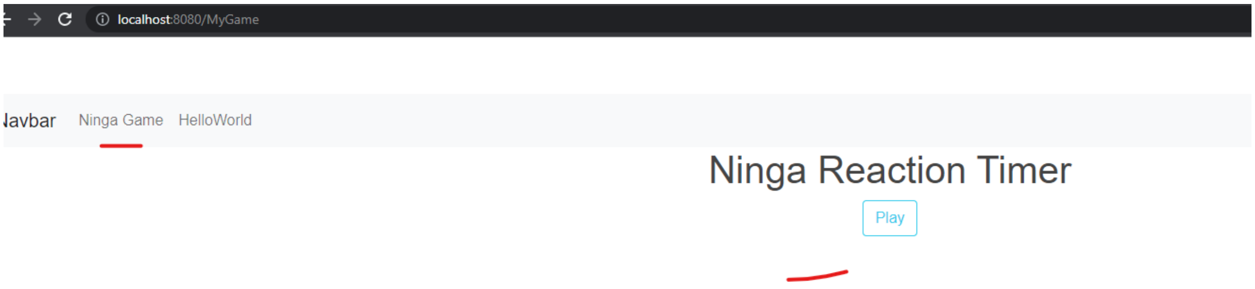


```
<template>
  <h1>Ninga Reaction Timer</h1>
 <button
    class="btn btn-outline-info"
    @click="start"
    :disabled="isPlaying"
   >Play</button>

   <MyBlock
    v-if="isPlaying"
    :delay="delay"
    @end="endGame"/>

    <MyResults
     v-if="showResults"
     :score="score"
    />
</template>

<script>
import MyBlock from './MyBlock.vue';
import MyResults from './MyResults.vue';
export default {
  name:'MyGame',
```

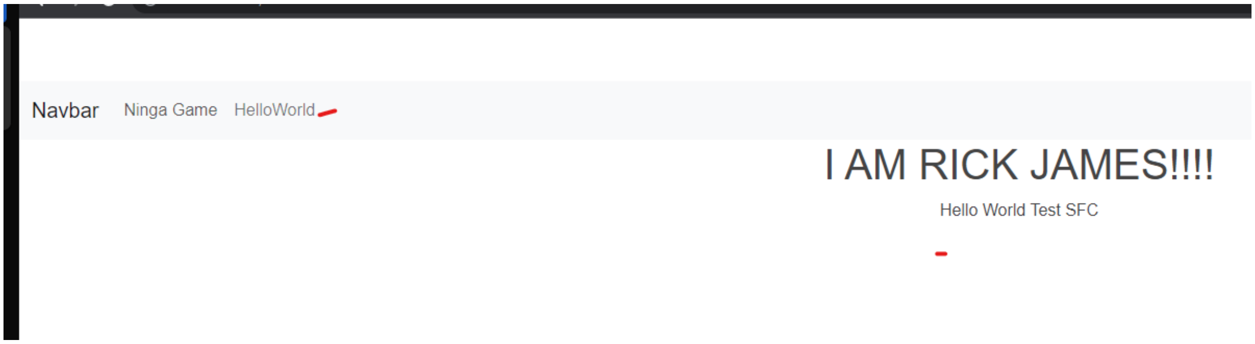

main.js
HelloWorld.vue
App.vue M
MyGame.vue U
index.js M
MyI
src > router > index.js > routes
You, 1 second ago | 1 author (You)
You, 1 second ago • Uncommitted changes
M
U
M
M
U

```js
import { createWebHistory, createRouter } from 'vue-router';
import HelloWorld from '../components/HelloWorld.vue';
import MyGame from '../components/MyGame.vue';

const routes = [
  {
    path: '/HelloWorld',
    name: 'HelloWorld',
    component: HelloWorld,
  },
  {
    path: '/MyGame',
    name: 'MyGame',
    component: MyGame,
  },
];

const router = createRouter({
  history: createWebHistory(),
  routes,
});

export default router;

```





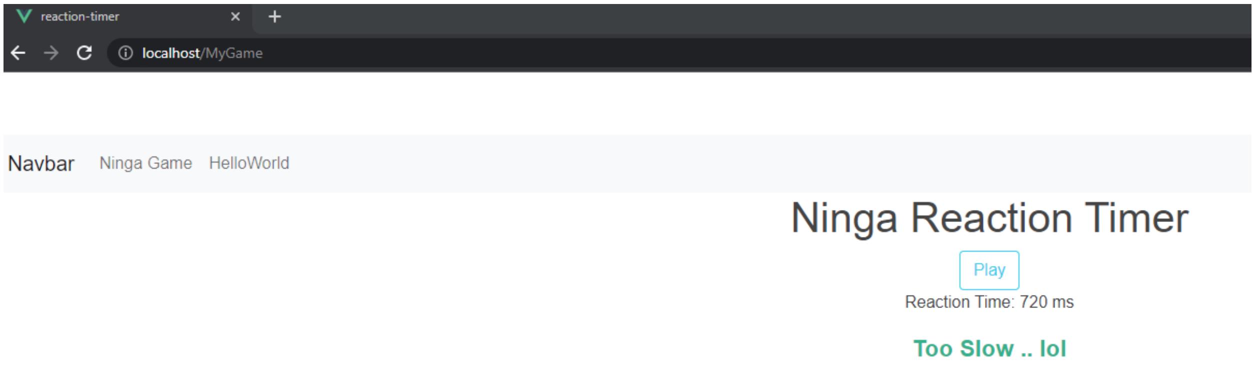
Ninga Reaction Timer



I AM RICK JAMES!!!!

Hello World Test SFC

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

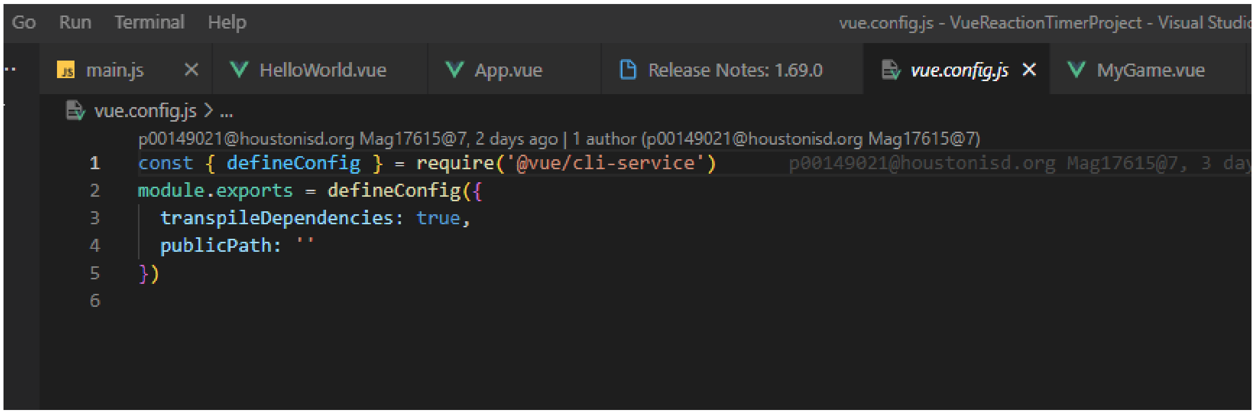


Ninga Reaction Timer

Reaction Time: 720 ms

Too Slow .. lol

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

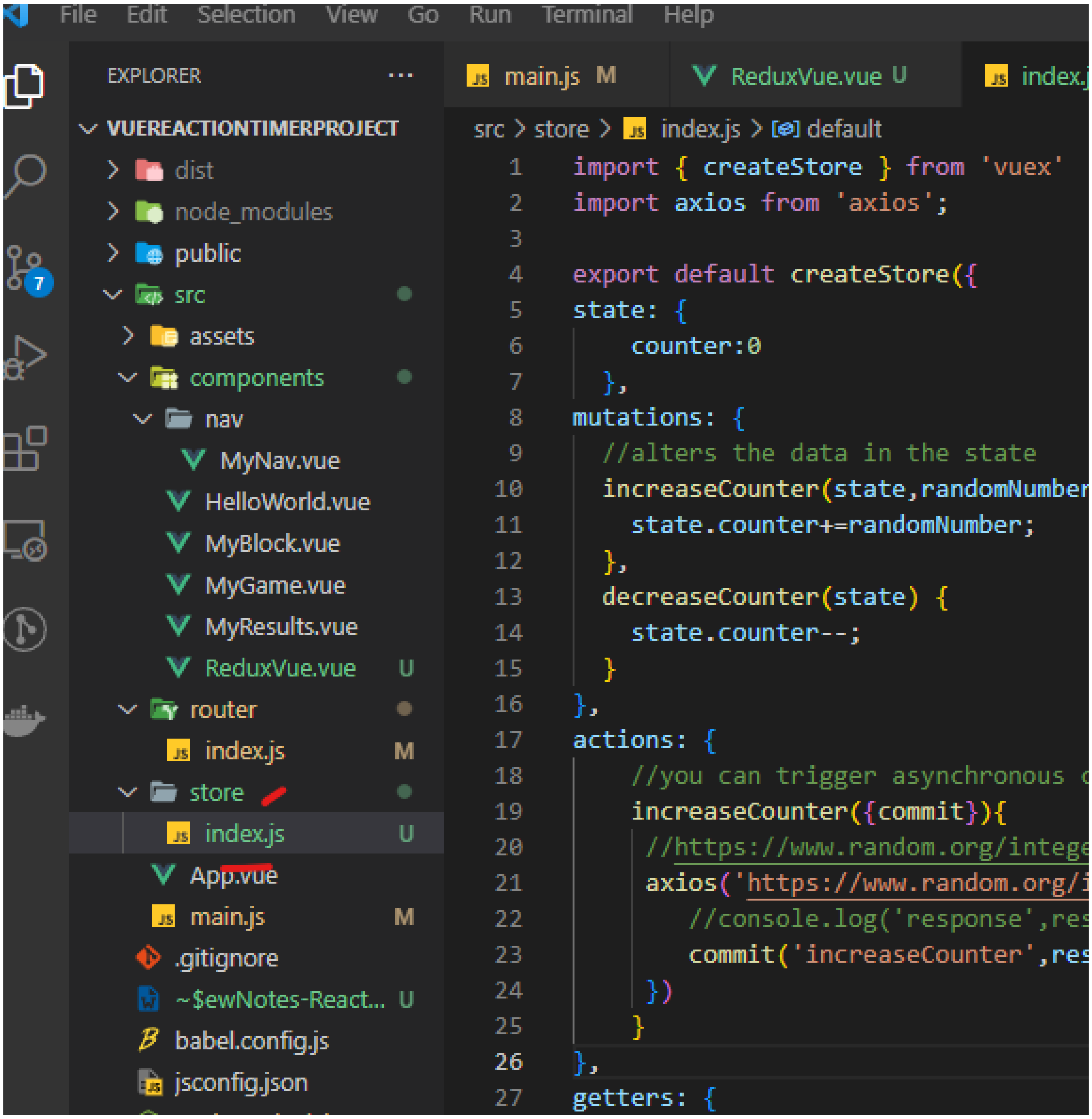

Go
Run
Terminal
Help
vue.config.js - VueReactionTimerProject - Visual Studi
main.js
HelloWorld.vue
App.vue
Release Notes: 1.69.0
vue.config.js
MyGame.vue
vue.config.js > ...
p00149021@houstonisd.org Mag17615@7, 2 days ago | 1 author (p00149021@houstonisd.org Mag17615@7)
p00149021@houstonisd.org Mag17615@7, 3 day

```
const { defineConfig } = require('@vue/cli-service')
module.exports = defineConfig({
  transpileDependencies: true,
  publicPath: ''
})

```

Tip: Vuex (State Management) in Vue
This came from this video
https://www.youtube.com/watch?v=nFh7-HfODYY

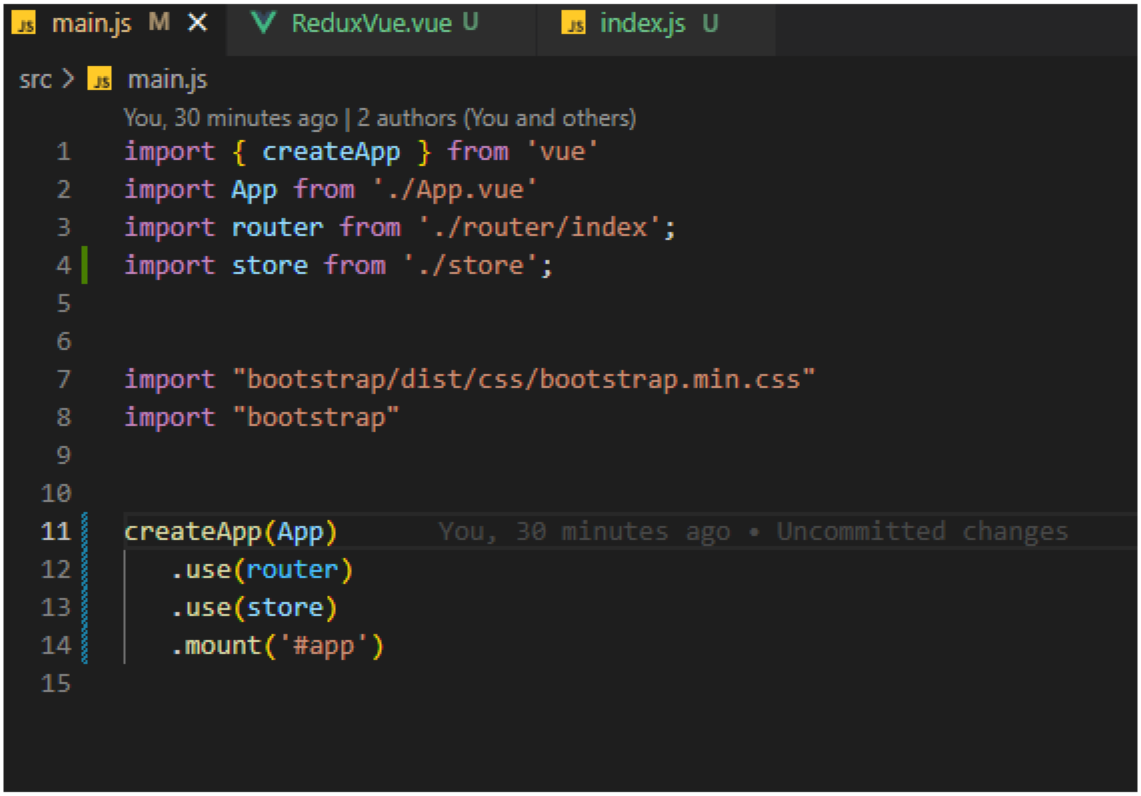
To add vuex to your project
npm install vuex@next –save

Then we can create our store



```
import { createStore } from 'vuex'
import axios from 'axios';

export default createStore({
state: {
    counter:0
  },
mutations: {
  //alters the data in the state
  increaseCounter(state,randomNumber
    state.counter+=randomNumber;
  },
  decreaseCounter(state) {
    state.counter--;
  }
},
actions: {
    //you can trigger asynchronous c
    increaseCounter({commit}){
     //https://www.random.org/intege
     axios('https://www.random.org/i
        //console.log('response',res
        commit('increaseCounter',res
     })
    }
},
getters: {
```

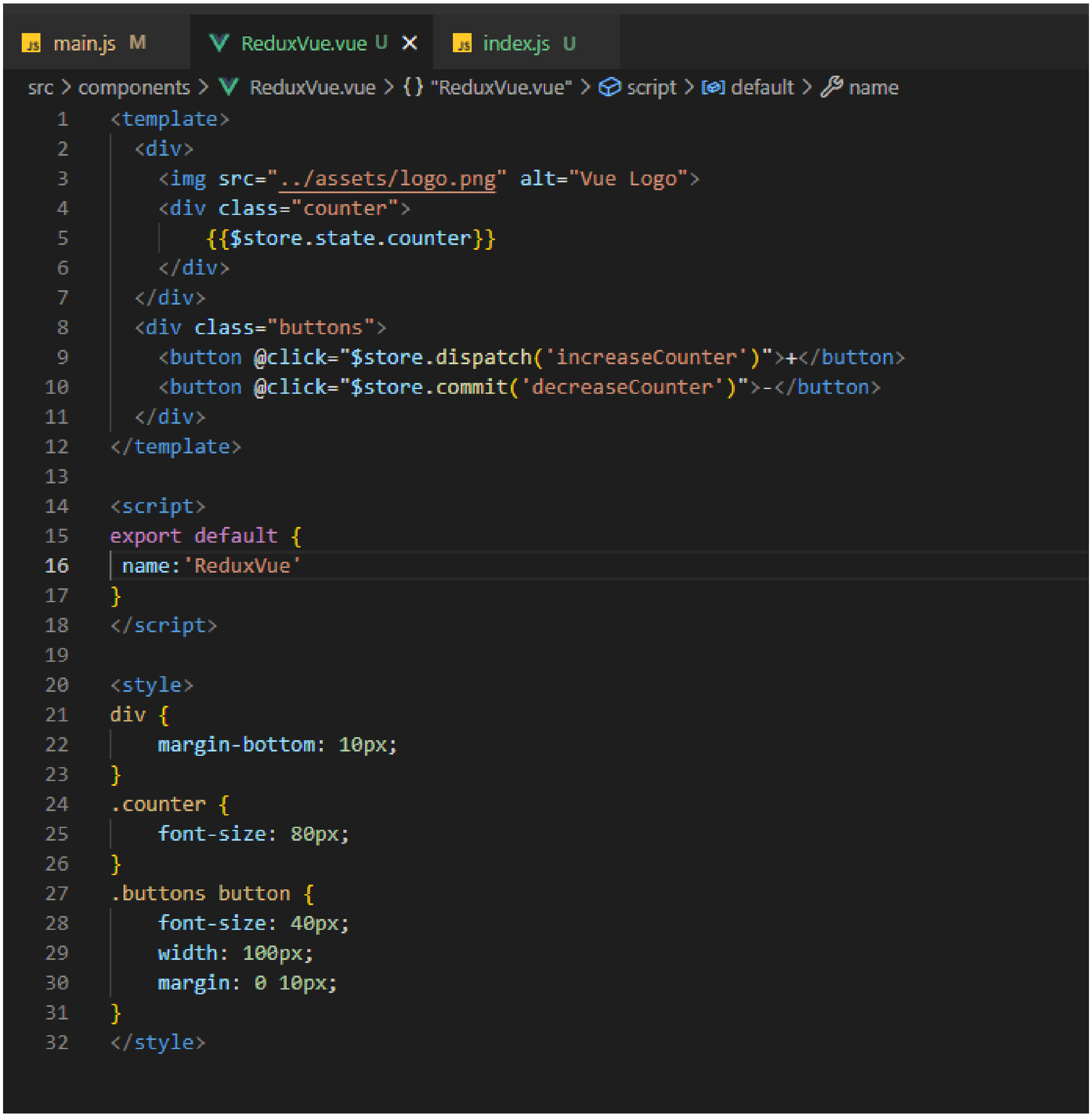
To add our store, we make a reference in our main.js file


main.js M
ReduxVue.vue U
index.js U
src > main.js
You, 30 minutes ago | 2 authors (You and others)
You, 30 minutes ago • Uncommitted changes

```
import { createApp } from 'vue'
import App from './App.vue'
import router from './router/index';
import store from './store';


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


createApp(App)
    .use(router)
    .use(store)
    .mount('#app')

```

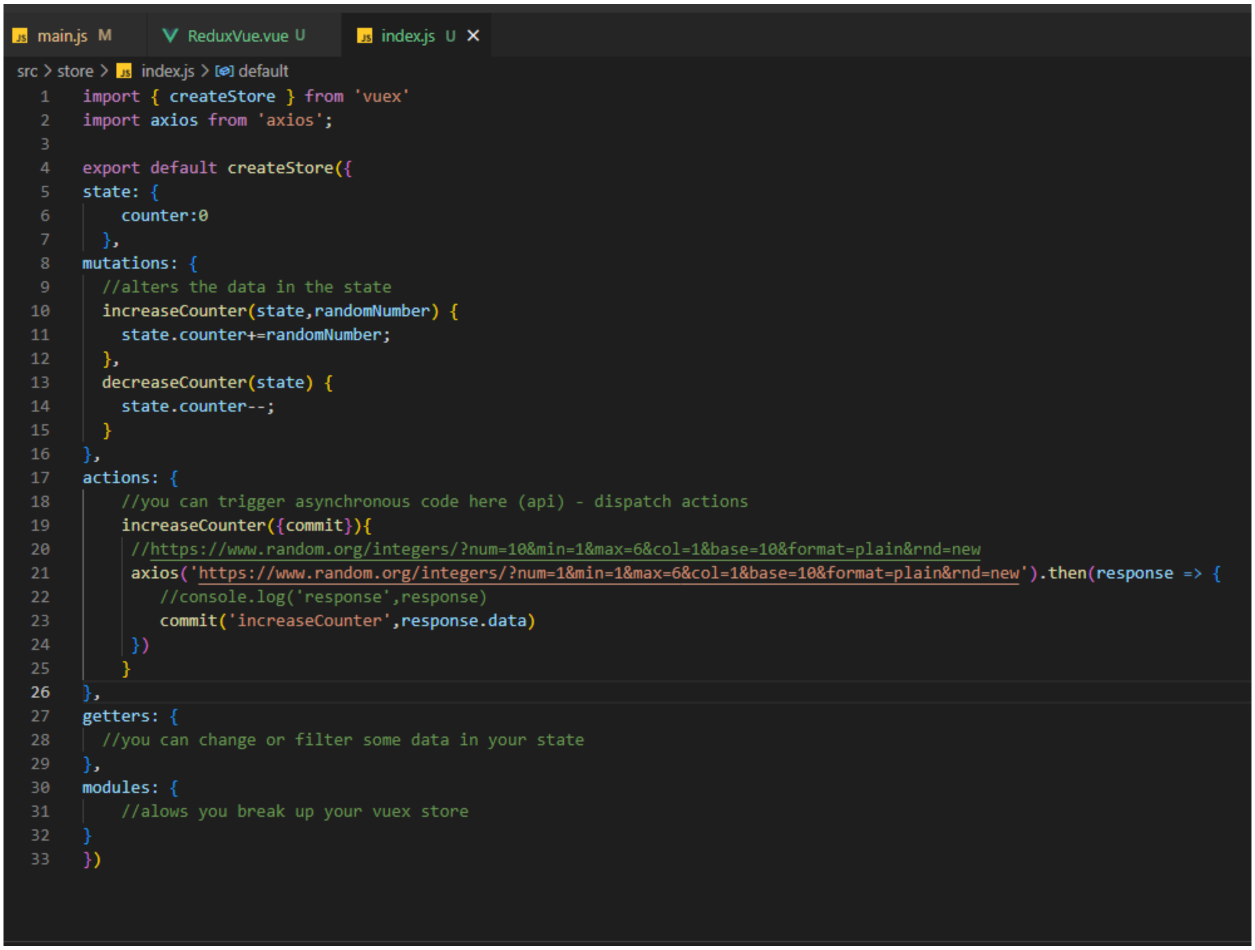
Now every component in our application has access to our store

We will demonstrate two types of actions in our store


main.js M
ReduxVue.vue U
index.js U
src > components > ReduxVue.vue > {} "ReduxVue.vue" > script > default > name

```
<template>
  <div>
    <img src="../assets/logo.png" alt="Vue Logo">
    <div class="counter">
        {{$store.state.counter}}
    </div>
  </div>
  <div class="buttons">
    <button @click="$store.dispatch('increaseCounter')">+</button>
    <button @click="$store.commit('decreaseCounter')">-</button>
  </div>
</template>

<script>
export default {
 name:'ReduxVue'
}
</script>

<style>
div {
    margin-bottom: 10px;
}
.counter {
    font-size: 80px;
}
.buttons button {
    font-size: 40px;
    width: 100px;
    margin: 0 10px;
}
</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>



```
import { createStore } from 'vuex'
import axios from 'axios';

export default createStore({
state: {
    counter:0
  },
mutations: {
  //alters the data in the state
  increaseCounter(state,randomNumber) {
    state.counter+=randomNumber;
  },
  decreaseCounter(state) {
    state.counter--;
  }
},
actions: {
    //you can trigger asynchronous code here (api) - dispatch actions
    increaseCounter({commit}){
     //https://www.random.org/integers/?num=10&min=1&max=6&col=1&base=10&format=plain&rnd=new
     axios('https://www.random.org/integers/?num=1&min=1&max=6&col=1&base=10&format=plain&rnd=new').then(response => {
        //console.log('response',response)
        commit('increaseCounter',response.data)
     })
    }
},
getters: {
  //you can change or filter some data in your state
},
modules: {
    //alows you break up your vuex store
}
})
```

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 an vue tools

Website for unit test examples

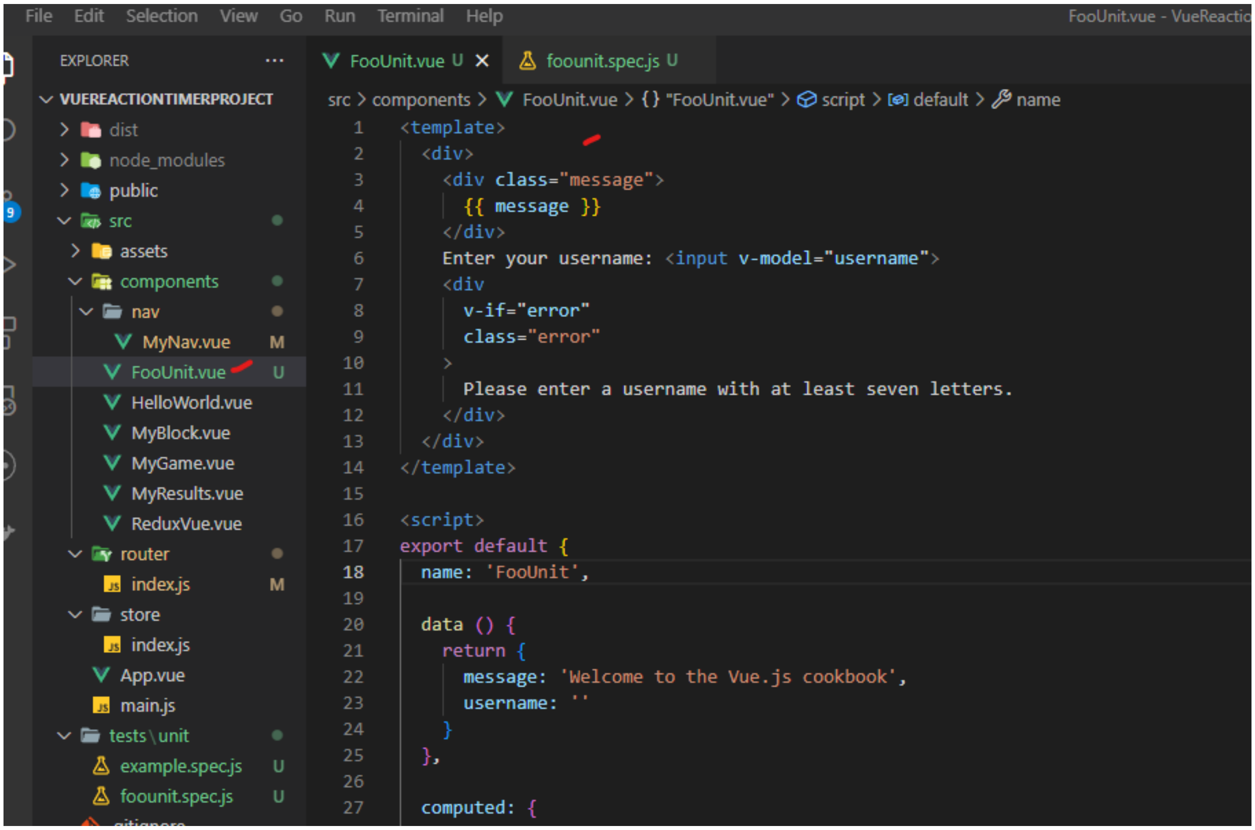
https://v2.vuejs.org/v2/cookbook/unit-testing-vue-components.html?#Why-test

Run these commands to set up unit testing

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

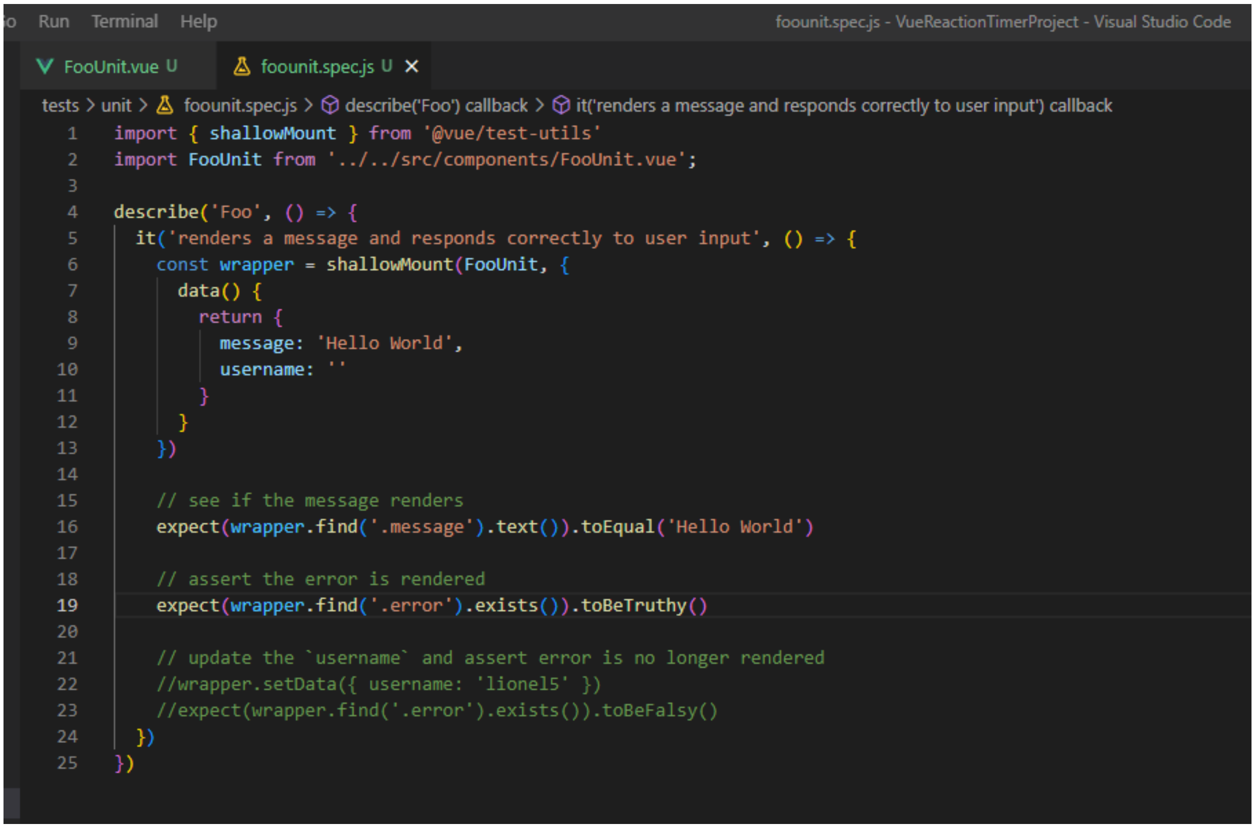
```
vue add unit-jest
npm install --save-dev @vue/test-utils
```

I created a component to unit test



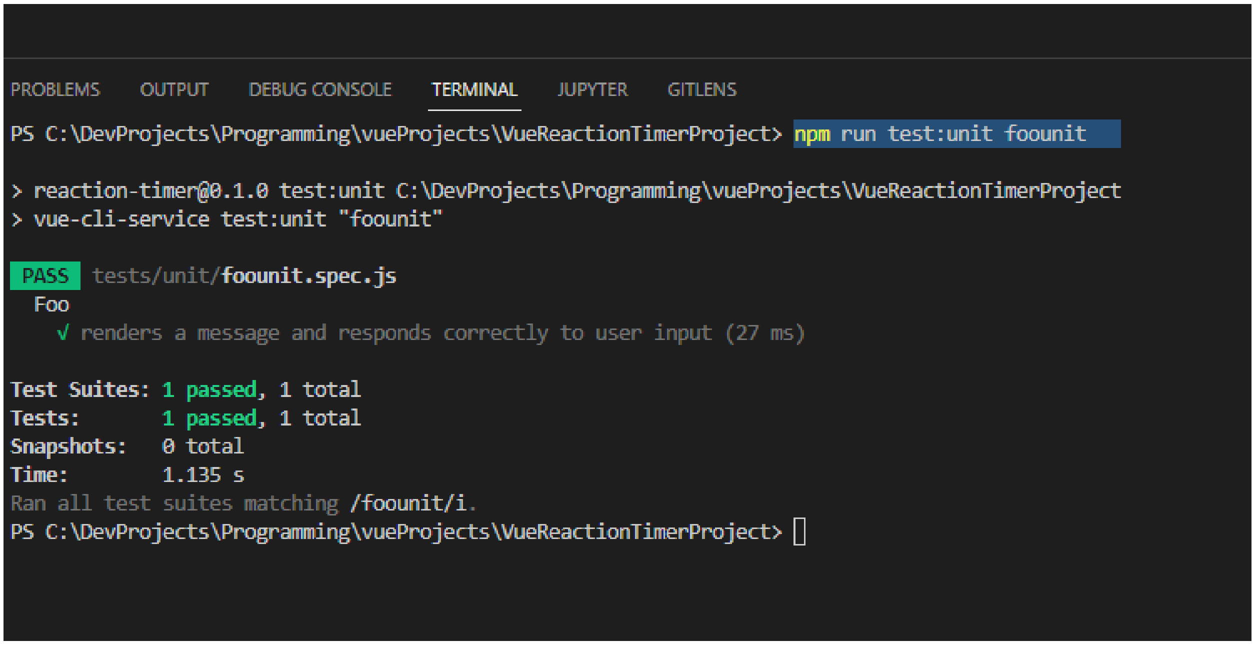
```
<template>
  <div>
    <div class="message">
      {{ message }}
    </div>
    Enter your username: <input v-model="username">
    <div
      v-if="error"
      class="error"
    >
      Please enter a username with at least seven letters.
    </div>
  </div>
</template>

<script>
export default {
  name: 'FooUnit',

  data () {
    return {
      message: 'Welcome to the Vue.js cookbook',
      username: ''
    }
  },

  computed: {
```

I added some test code


o Run Terminal Help
foounit.spec.js - VueReactionTimerProject - Visual Studio Code
FooUnit.vue U
foounit.spec.js U ×
tests > unit > foounit.spec.js > describe('Foo') callback > it('renders a message and responds correctly to user input') callback

```
import { shallowMount } from '@vue/test-utils'
import FooUnit from '../../src/components/FooUnit.vue';

describe('Foo', () => {
  it('renders a message and responds correctly to user input', () => {
    const wrapper = shallowMount(FooUnit, {
      data() {
        return {
          message: 'Hello World',
          username: ''
        }
      }
    })

    // see if the message renders
    expect(wrapper.find('.message').text()).toEqual('Hello World')

    // assert the error is rendered
    expect(wrapper.find('.error').exists()).toBeTruthy()

    // update the `username` and assert error is no longer rendered
    //wrapper.setData({ username: 'lionel5' })
    //expect(wrapper.find('.error').exists()).toBeFalsy()
  })
})
```

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



```
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> 
```