https://blog.logrocket.com/how-to-use-vue-3-typescript/
https://vuejsexamples.com/vueboard-admin-dashboard-built-vue-3-tailwind-css-and-typescript/
https://www.youtube.com/watch?v=JfI5PlSLr9w

Vue3JS, Pinia — https://delldigital.udemy.com/course/vue-js-v3-super-fast-course-from-zero-to-advanced-web-development — https://github.com/xinle1030/Vue-JS-3-Complete---Including-Composition-API-and-Pinia/tree/master
Vue3JS with Typescript, Pinia — https://delldigital.udemy.com/course/vuejs-3-the-composition-api/

Install Vue3JS
1. Install Node JS
2. Install Vue cli — npm install -g @vue/cli
3. Update Vue cli — npm update -g @vue/cli

Create Vue Project
1. vue create <project-name>
2. manually selected features — Babel, Typescript, Router, Vuex, CSS Processors, Linter/Formatter

Create Vue Project with Vite — Unlike Vue CLI, Vite doesn't rely on Webpack. Instead, it has its own development server that leverages native ES modules directly in the browser. Vite utilizes Rollup for the build process, which results in faster performance compared to other methods.
1. Install vite — npm install -g vite
2. Create vite project — npm creare vite
3. To run vite project — npm i
vite

Use Vue router
1. Install router — npm install vue-router
2. cretae a new file at ./src called router.ts
3. At ./src/router.ts
4. In main.ts

```
import { createRouter, createWebHistory } from "vue-router"
import { router } from "./router"
createApp(App).use(store).use(router).mount('#app')
```

5. Create router in ./src/router.ts
* beforeEnter is to use as navigation guard

```
export const router = createRouter({
  history: createWebHistory(),
  routes: [
    {
      path: "/",
      component: Home
    },
    {
      path: "/posts/new",
      component: NewPost,
      beforeEnter: () => {
        const usersStore = useUsers();

        if (!usersStore.currentUserId) {
          return {
            path: "/"
          }
        }
      }
    },
    {
      path: "/posts/:id/edit",
      component: EditPost
    },
    {
      path: "/posts/:id",
      component: ShowPost
    }
  ]
})
```

6. Create each page in ./src/views folder
7. Import component in ./src/router.ts
8. Use <RouterView /> in App.vue to render component based on which route we are in now
9. Use router-link in navbar to navigate to diff route link

```
<router-link to="/">Home</router-link>
```

10. To use router in other file for redirect

```
import { useRouter } from 'vue-router';
const router = useRouter();
router.push("/")
```

Pinia — State Management System to build large application with VueJS
All the state and reactive variables live inside a store, acting as a single source of truth in the application
Store will then feed all the components that need those variables

1. Install Pinia — npm i pinia

2. In main.js, import pinia and make the app to use pinia

```
import "./assets/reset.css";
import "./assets/main.css";

import { createApp } from "vue";
import { createPinia } from 'pinia';

import App from "./App.vue";

const pinia = createPinia();
const app = createApp(App);
app.use(pinia);
app.mount("#app");
```

3. Create ./src/stores folder

4. Create a store js file under the folder

```
import { defineStore } from "pinia";

export const useTasksStore = defineStore("store_id", () => {})
```

5. Follow naming convention — use<StoreName>Store

6. Set up store using Option API / Composition API — always need to return everything from the store that we want to use from the outside including state variable, computed variable and method

7. To use the store from other file

```
import { useTasksStore } from "@/stores/tasksStore.js";
const store = useTasksStore();
```

"@ means src folder"

To run a ts file in src

1. Run """npx ts-node-esm src/<pathname>""" in the terminal

For authentication and authorization

1. Instal necessary packages — npm i cookie-parser @types/cookie-parser jsonwebtoken @types/jsonwebtoken express-session @types/express-session --include=dev

For frontend to configure backend route so we dont have to purposely mention backend route as http://localhost:8000 in the FE code — Use server proxy to make both FE and BE have the same origin on localhost:3000 - have everything run on same port

1. In vite.config.ts

```
export default defineConfig({
  plugins: [vue()],
  server: {
    proxy: {
      "^/api/.*": {
        target: "http://localhost:8000",
        changeOrigin: true,
        rewrite: (path) => {
          const p = path.replace(/^\/api/, "");
          return p;
        },
      },
    },
  },
});
```

2. Now we can just fetch from "api/posts" instead of "http://localhost:8000/api/posts" in FE code

Deploy to production

1. Build an optimised production bundle — npm run build
   Build file can be found under <project folder>/dist/

2. Use Digital Ocean

3. Configure static server with NginX which is a popular http server
   Install NGINX on your machine
   Copy all the build files under /dist into the default place where nginx lives — cp -r dist/* ?usr/share/nginx/html
   Configure nginx.conf at project root folder

Testing Module

Previously, this course used Jest + Vue Test Utils for testing. This works well for smaller apps
But now-days I recommend either Vitest or Cypress.

Tute: https://www.youtube.com/watch?v=nLBwVOJDT1I
Codebase: It uses the same code base as this course. The final code can be viewed as a diff here. https://github.com/lmiller1990/vuejs-composition-api-v3/compare/master...testing

1. Install vitest — npm install @vue/test-utils jsdom vitest --include=dev

2. Configure vite.config.ts

```
/// <reference types="vitest" />
import { defineConfig } from "vite";
import vue from "@vitejs/plugin-vue";

export default defineConfig({
  plugins: [vue()],
test: {globals:true, environment: 'jsdom'},
  server: {
    proxy: {
      "^/api/.*": {
        target: "http://localhost:8000",
        changeOrigin: true,
        rewrite: (path) => {
```

```
            const p = path.replace(/^\/api/, "");
            return p;
          },
        },
      },
    },
  },
});
```

3. For each component, create a file <component>.spec.ts

4. Inside <component>.spec.ts, put following code

```
import { mount } from "@vue/test-utils"
import { describe, it } from "vitest"

describe("componentName", () => {it(testName", () => {  // do smt  })  })
```

5. Run test

At project root folder terminal, run "npx vitest"