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=Jfl5PlSLr9w

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

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"