Faire un clone du repository git :  
<https://github.com/alex8088/electron-vite-decorator-example>

Ce proje test une bonne base mais il est en solid et non en vue  
Il faut passer le projet en vue avec els manip suivante :  
  
1) : remplacer le code de main.tsx par celui-ci

import { createApp } from 'vue';

import App from './App.vue';

createApp(App).mount('#root');

2) : Renomer le fichier « App.tsx » en « App.vue » et remplacer le code par celui-ci

<template>

  <div class="container">

    <Versions />

    <!-- ... rest of your code ... -->

  </div>

</template>

<script>

import Versions from './components/Versions.vue';

export default {

  components: {

    Versions,

  },

};

</script>

<style>

/\* Styles go here \*/

</style>

3) : Renomer le fichier « Version.tsx » en « Version.vue » et remplacer le code par celui-ci

<template>

  <ul class="versions">

    <li class="electron-version">Electron v{{ versions.electron }}</li>

    <li class="chrome-version">Chromium v{{ versions.chrome }}</li>

    <li class="node-version">Node v{{ versions.node }}</li>

    <li class="v8-version">V8 v{{ versions.v8 }}</li>

  </ul>

</template>

<script>

export default {

  data() {

    return {

      versions: window.electron.process.versions,

    };

  },

};

</script>

4) : Pour le moment le projet ne fonctionne par car il faut faire les install et les config suivante donc faire l’installe de vue avec la commande suivante  
npm install vue

5) :Remplacer le code dans tsconfig.json par celui-ci

{

  "files": [],

  "references": [{ "path": "./tsconfig.node.json" }, { "path": "./tsconfig.web.json" }],

  "compilerOptions": {

    "target": "esnext",

    "module": "esnext",

    "moduleResolution": "node",

    "jsx": "preserve",

    "strict": true,

    "esModuleInterop": true,

    "skipLibCheck": true,

    "forceConsistentCasingInFileNames": true,

    "allowJs": true,

    "resolveJsonModule": true,

    "isolatedModules": true,

    "noEmit": true,

    "jsxImportSource": "vue",

    "types": ["vite/client", "vue"]

  },

  "include": ["src/\*\*/\*.ts", "src/\*\*/\*.d.ts", "src/\*\*/\*.tsx", "src/\*\*/\*.vue"],

  "exclude": ["node\_modules"]

}

6) :Remplacer le code dans electron.vite.config.ts par celui-ci

import { resolve } from 'path';

import { defineConfig, externalizeDepsPlugin, swcPlugin } from 'electron-vite';

import Vue from '@vitejs/plugin-vue';

export default defineConfig({

  main: {

    plugins: [externalizeDepsPlugin(), swcPlugin()]

  },

  preload: {

    plugins: [externalizeDepsPlugin()]

  },

  renderer: {

    resolve: {

      alias: {

        '@renderer': resolve('src/renderer/src')

      }

    },

    plugins: [Vue()]

  }

})

7) : Mettre a jours la version de vite avec la commande suivante

npm install vite@latest --save-dev

8) : Installer le pluging « @vitejs/plugin-vue » avec la commande suivante

npm install @vitejs/plugin-vue --save-dev

9) : Lancer le serveur avec la commande   
npm run dev

10) : Plus qu’à changer le nom du projet dans le package.json a la ligne ‘’ name ‘’ : ’’nom-du-projet’’

11) : Installer scss pour pouvoir l’utiliser dans les template vue voici la commande

npm install sass