**Tutorial 10: Introduction to Vue CLI**

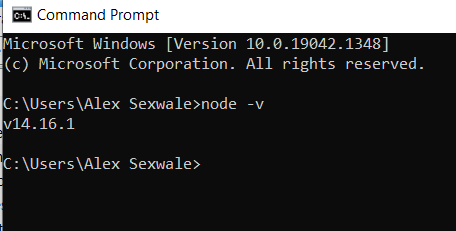
We will no longer be using the CDN. We will no longer manually creating our HTML, CSS and JavaScript files. We will be using the CLI (Command Line Interface) going forward.

Benefits of Vue CLI:

* Use Modern JavaScript features
* Provided us with a live-reload dev server
* Optimize our code for production

**Step 1:** Install [Node.js](https://nodejs.org/en/) on your computer to be able to use the **NPM** (Node Package Manager) to be able to install the Vue CLI. Click the download button to get the latest version. Run the installation and install the software to be able to get started.

To check whether Node is installed open up your terminal or command prompt and enter the command **node –v**. node-v checks the node version that is currently installed on your computer.



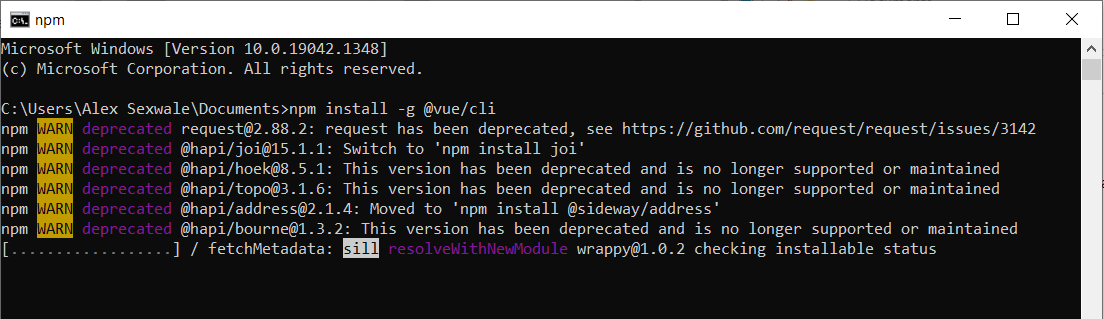
If the version number does not come up. Try to reinstall Node on your computer and check the node version again using **node –v**.

**Step 2:** Install the Vue CLI globally on your computer, type in your terminal or command prompt the following command:

**npm install –g @vue/cli**

**-g** refers to installing a package globally, accessible anywhere on the drive that it is being installed on.

**@vue/cli** is the package we are installing which is Vue.

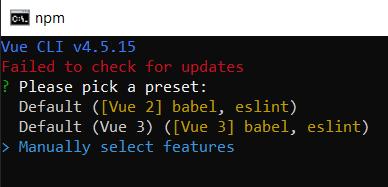


Make sure you are in the correct file path of where you want to create your projects in your terminal or command prompt.

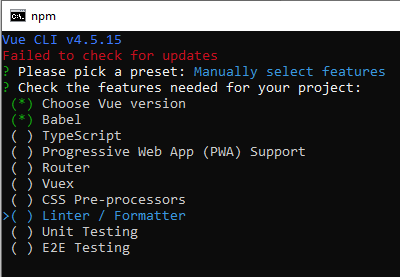
To create a new Vue project type in the command **vue create *\*project name\****

Example: **vue create modal-project**

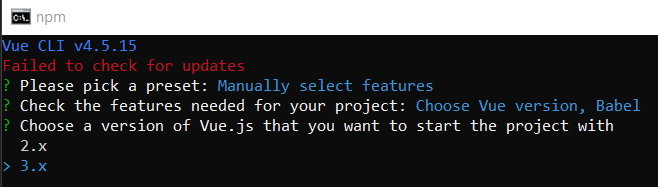
Select “Manually select features” option and press enter.



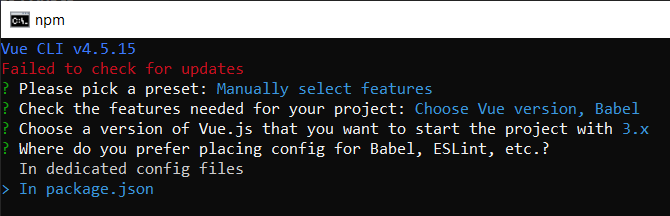
Deselect “Linter / Formatter” and leave “Choose Vue version” and “Babel” selected and press enter.



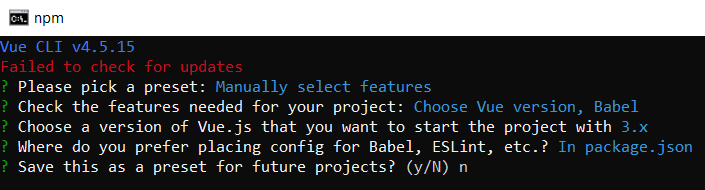
Select “3.x” and press enter.



Select “In package.json” and press enter.

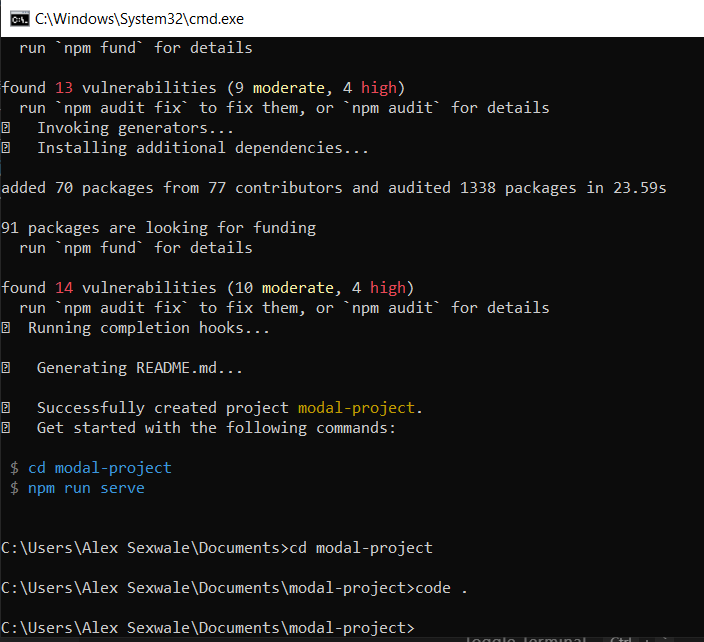


Type “n” and press enter



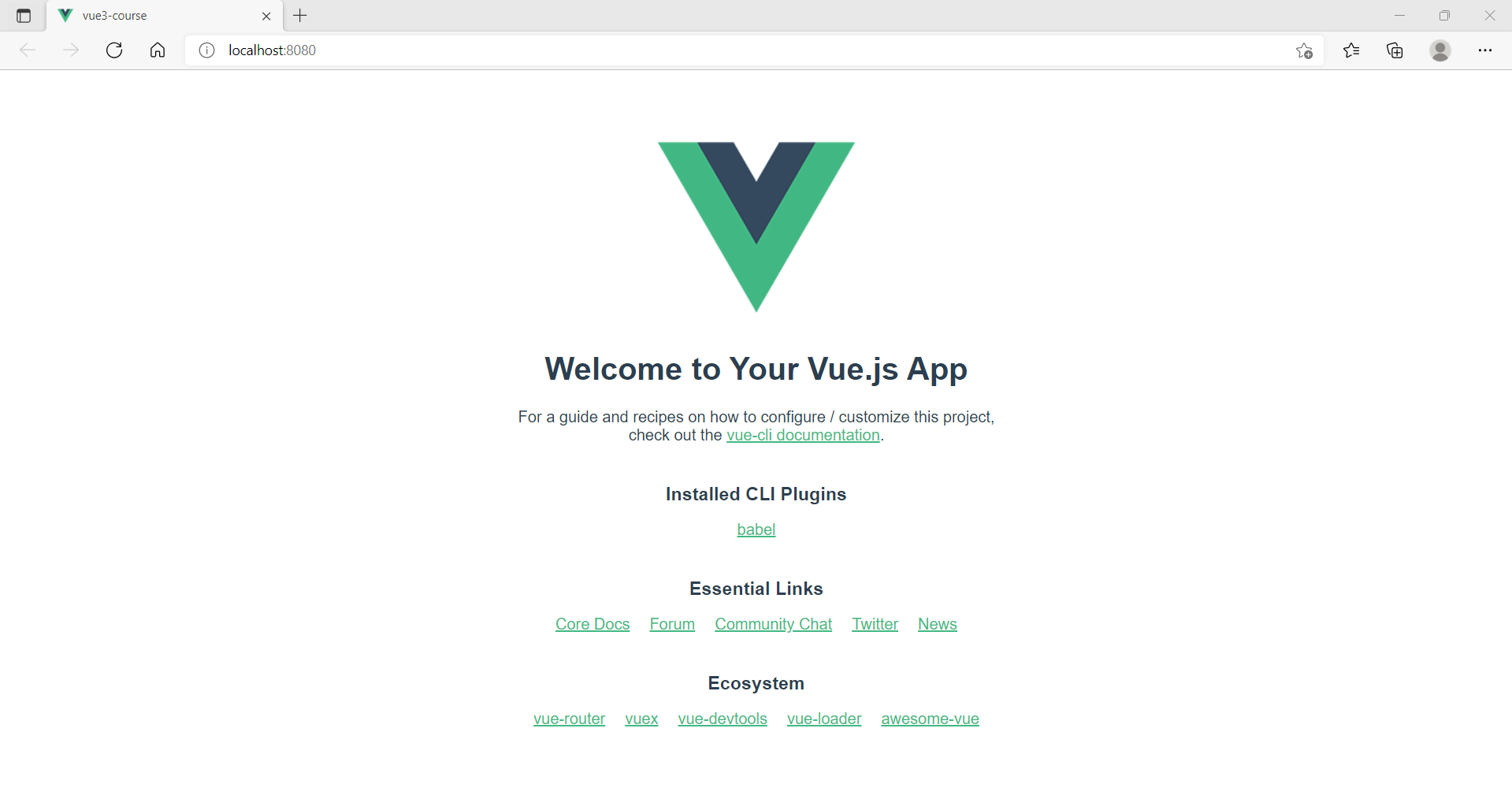
Change the directory by typing cd **modal-project** in the terminal or command prompt you are on.

Open the project using VS code, type **code .** in the terminal or command prompt in the directory of the project.



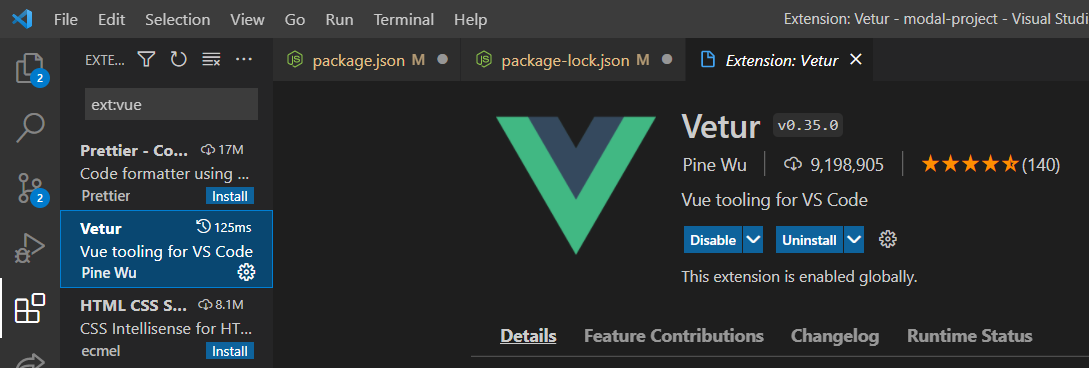
In order to run the project in the terminal or command prompt, make sure you are in the correct file path and type the command **npm run serve**. The project will open a new port, localhost://8080 which will allow you to view the website on your browser.

**Output:**



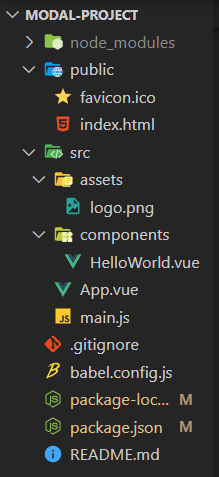
**Exploring the Vue File Structures**

Make sure the Vetur extension is installed on VS Code to be able to have a good colour combination when viewing the Vue syntax. Sometimes the syntax can be in plain white and will be difficult to follow.

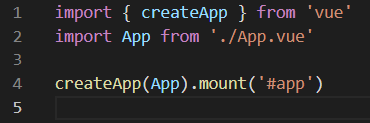


As previously discussed Vue is used to create single page applications, meaning only one HTML page is created and all the Vue components are rendered in that single HTML page. The rest of the pages will be created using the vue file extension.

Folder structure:



The **main.js** file starts the application. It takes the **App component** and passes it in as the root component to **createApp** and is then mounting it to the **dom** in an id called app.



Every Vue file represents a Vue component and has the .vue file extension at the end.

Every Vue component can contain three different parts.

Firstly, an HTML template inside a **template** tag, the HTML code is not written in the HTML file.

Secondly, a **script** tag where we export a component object.

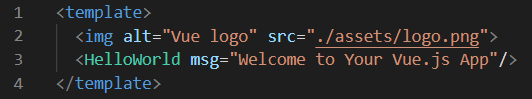
Lastly a **style** tag that can style any component that we apply to the template.

The **script** tags and style tags are optional but every component must have a template.

Example of Vue component:



**Templates**

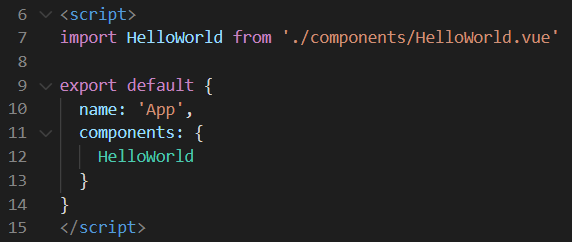


We create the template in the component template rather than the index.html file because we might have many different components with different templates and may not show all at the templates at the same time. Furthermore, there could be many components and can get messy really quickly.

Each component has its own template at the top then when we render that component in the dom, its template is injected into the dom.

The root tag must always be a template tag. Inside you can organise your template the way need it to be in any way.

**Script**



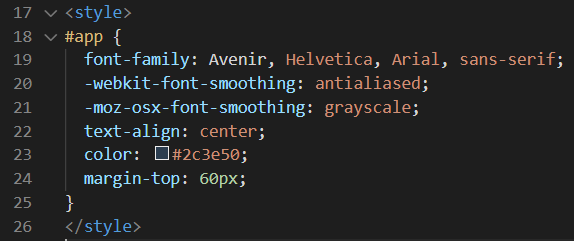
We can import other components which makes it easier for us to manage our code and reduces the amount of code we need to write as we are able to reuse the components in multiple places. We can also import other files such as JavaScript files which we can use inside the component and/or other node packages that we may have installed and what to include in the specific components (we will go in-depth in later tutorials).

We can export the current component which will us to import the component in other components and/or call the component in JavaScript files like main.js (which is used to render the component in index.html file).

The export default object represents the root component. It can contain keys such as data, methods, computed, etc.

Providing a name is optional, but it provides the component name and when wanting to call the component in the template tag of another component, it would be useful to provide the component a name that represents what the component is for especially if you create Vue files with same names.

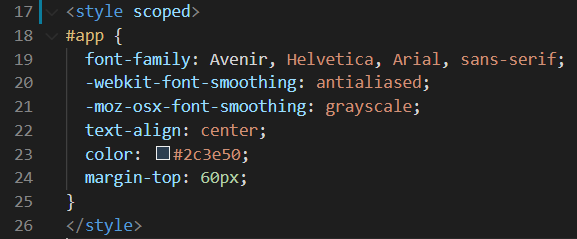
**Style**



The style tag is global meaning you can style any id, class or tag.

We can scope them so the styling only refers to this template meaning that is no longer global.

Look at example below (**line 17**).



From this tutorial, Tutorial-10, onwards, when you clone the repository, make sure to open up the terminal or command prompt and type out the command **npm install** to make sure the dependencies are install. otherwise the Vue project will not run. You are required to have a **node\_modules** folder which is where the dependencies run from which is not a file that will be found on the **Github** repository.

Additional resources can be found on the [Vue CLI](https://cli.vuejs.org/) website.