Unit 5 - Vue.js - Node.js

Classwork: The classwork for this unit should be saved in a new folder:
your_repo/unit_5/vuejs_nodejs/
Homework: Homework should be saved here: **your_repo/homework/unit_5/**

*** Day 01 ***

What is a component in Vue.js?

Components in Vue.js are like modules in Python. You will be able to create a component, which will be a piece of your HTML code and you will be able to reuse it multiple times (for example for a list of data), and even if you use it only once, it will be useful to separate the code to keep it clean.

But for that, we will need to deploy our **Vue.js app** with **Node.js,** which will **compile our Vue.js app in JavaScript.**

Let's install **npm**, a package manager for Node.js and is included as a recommended feature when you install the Node.js. We are going to use Python to install npm because there is a python package to help us create a npm virtual env like we do for python.

Install npm

1. Create a python virtual environment
2. Activate the virtual env
3. `pip install nodeenv` (this package will help to create a nodejs virtual environment, like we have with python)
4. Install nodejs 20.11.1 (the latest LTS(long-term) version at this time), anywhere you want.
   nodeenv --node=20.11.1 --prebuilt env_node_20.11.1
5. Deactivate your python => deactivate
6. Activate your node env
   source env_node_20.11.1/bin/activate
7. Done!

Install Vue.js

Go here for info: https://vuejs.org/guide/quick-start.html

We are going to create a Vue app via script. You need to run this command on the folder you are going to have your project.

```
npm create vue@latest
```

Select No for everything except the last 2 options: ESLint and Prettier

Go to the new folder created by Vue.js and type:

```
npm install
```

Npm is like the pip in Python.

Vue.js (via the script npm create vue@latest), has created a **package.json**. Here is where you set up the js packages you want to use. This is similar to the requirement file we use in Python.

In package.json, we have the Vue.js package as dependency. We also have some devDependencies, we have some packages here to check the syntax and formatting. There is a package named vite. **Vite will be used to create a front-end server.** Any modification you will do, it is going to be refreshed immediately, like the live server we used previously.

Npm is going to install the packages in the **node_modules** folder. You can check, there are a lot of packages there.

One advice in case you use google drive (or OneDrive for microsoft): do not install npm package on this type of drive because it is going to take a long time to sync (there are a lot, a lot of small files). Like the python environment, too. But in the python environment you can install it anywhere you want, activate it, and install packages as you want. With nodejs, you do not have this flexibility, so be aware. Nodejs is going to install a node_modules folder in the same directory as your js project. So annoying. I searched on how to have this node_modules outside of the projects, but did not find anything.

Once npm has installed the packages, we can **run the vite server**, by running:

npm run dev

It is going to tell you in which port it is running. Mine is running at http://localhost:5173/

In the package.json, there is a **scripts section,** and we see that the **dev command is calling vite**. Vite is going to look at the vite.config.js and will use the file index.html as an entry point for the html. I am not very familiar with vite yet. By the way, vite is a French word that means fast.

Vue js has been setting up things for us.

The index.html page is calling to the src/main.js
In the src folder, you have your vue js app.
In the main.js you have the vuejs app and it is mounted.

Rename the folder src, as src_vuejs, and let's create a new folder src.

Copy the main.js file from src_vuejs to src. Remove the line about the CSS.

Create an App.vue in the folder src.
And let's copy the hello-world example (https://vuejs.org/examples/#hello-world):

```
<!--
Say Hello World with Vue!
-->

<script setup>
import { ref } from 'vue'

// A "ref" is a reactive data source that stores a value.
// Technically, we don't need to wrap the string with ref()
// in order to display it, but we will see in the next
// example why it is needed if we ever intend to change
// the value.
const message = ref('Hello World!')
</script>

<template>
  <h1>{{ message }}</h1>
</template>
```

Save it, you should see the new page refreshed automatically on your page served by the vite server. (http://localhost:5173/ for me)

I let you discover the other examples: https://vuejs.org/examples/