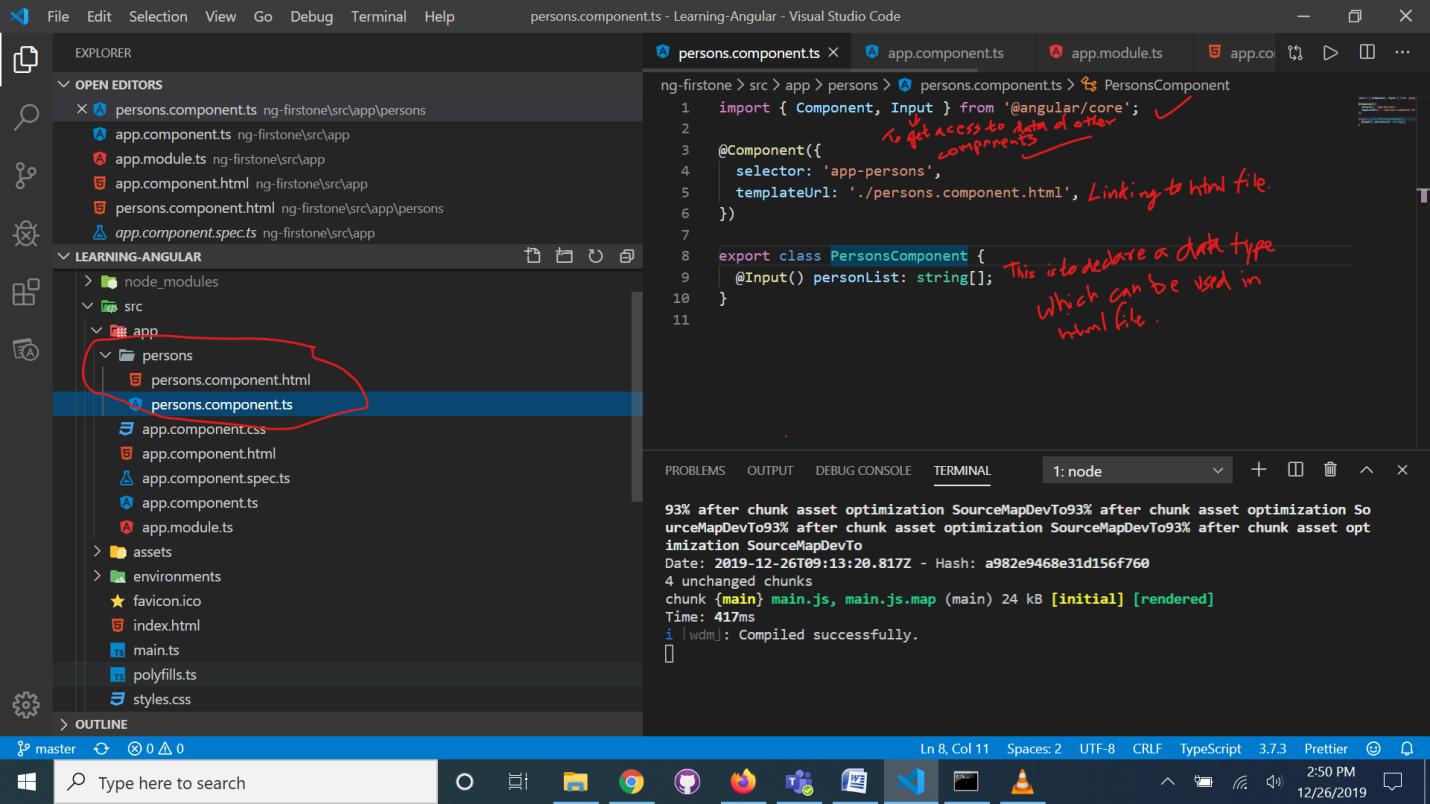
To add a new component, make a new component and html file, preferably inside a new folder.

Every component has selector field which assigns it a name which can be used as a tag in the html file.

The html file of that component is then connected with the app component by mentioning the new component’s tag in its(app component) html file which in turn has a selector in its own component.ts, which is called in index.html !

1.New folder -> Make new.component.ts and new.component.html

2. New.Component.ts -> add @component and add selector name -> add export class with any name which will be imported by app.module.ts

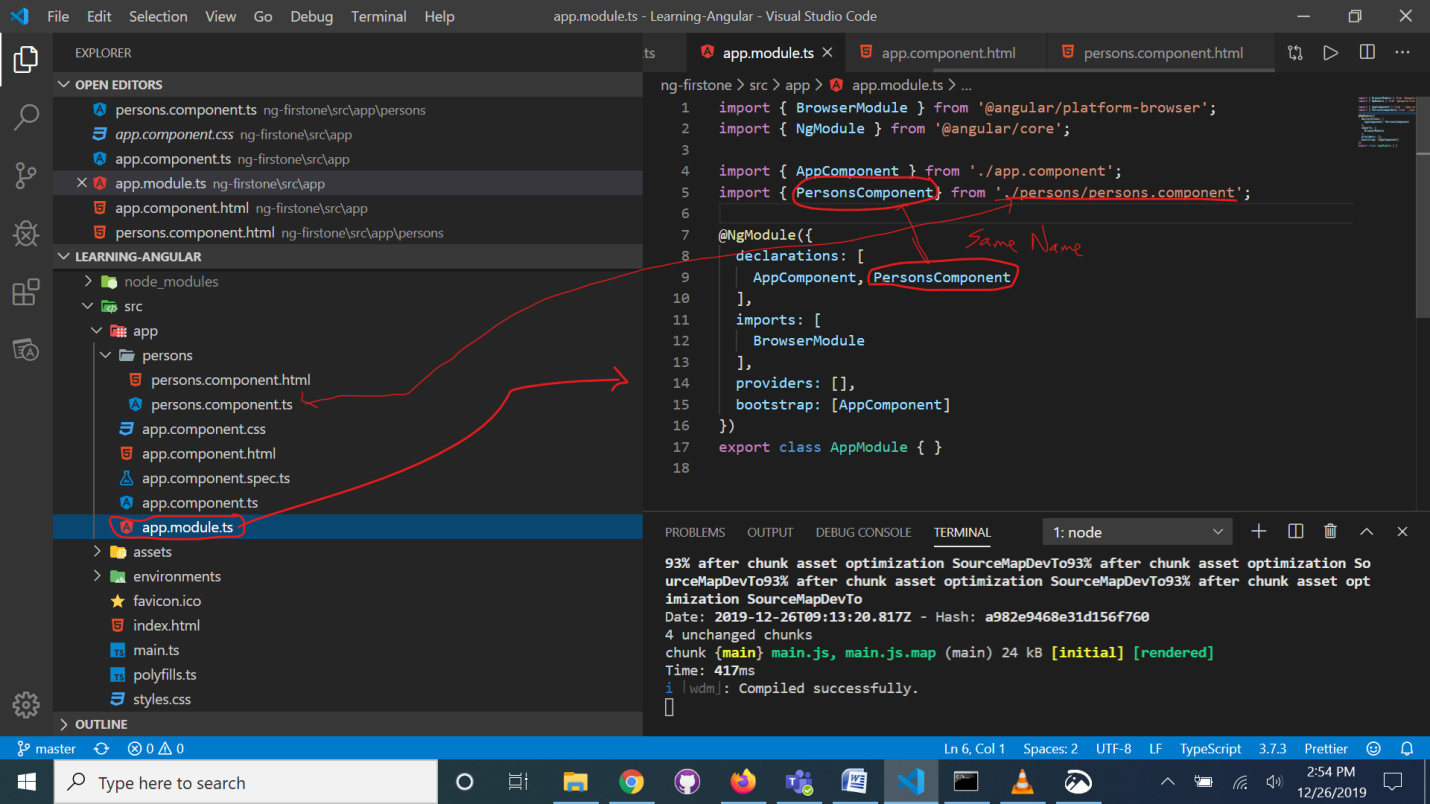


*Remember the name used in export statement is used in all other files for referencing this file.*

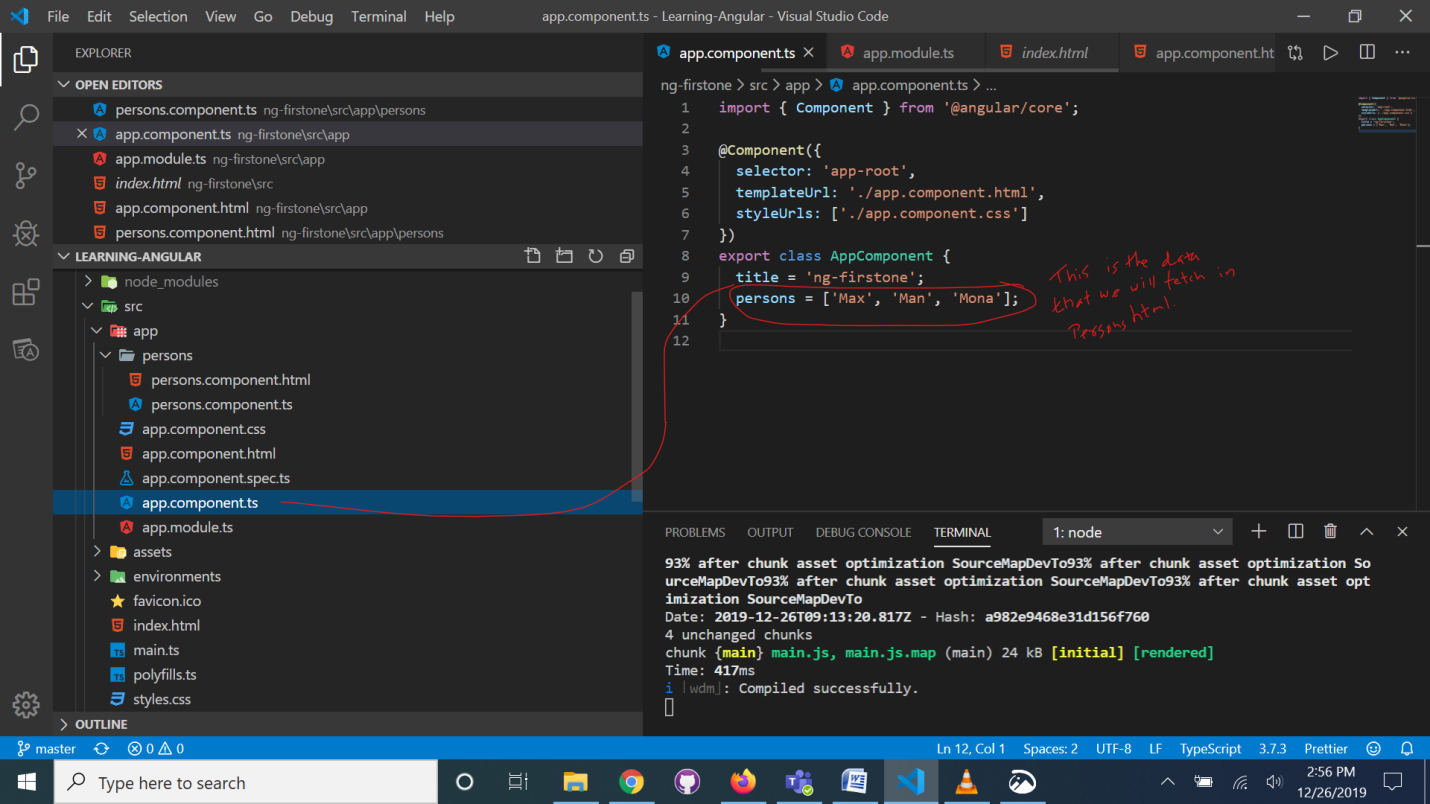
*Here we chose PersonComponent.*

*Also, we declared the personList as the input, so this variable will be given some input of string array type in the html page.*

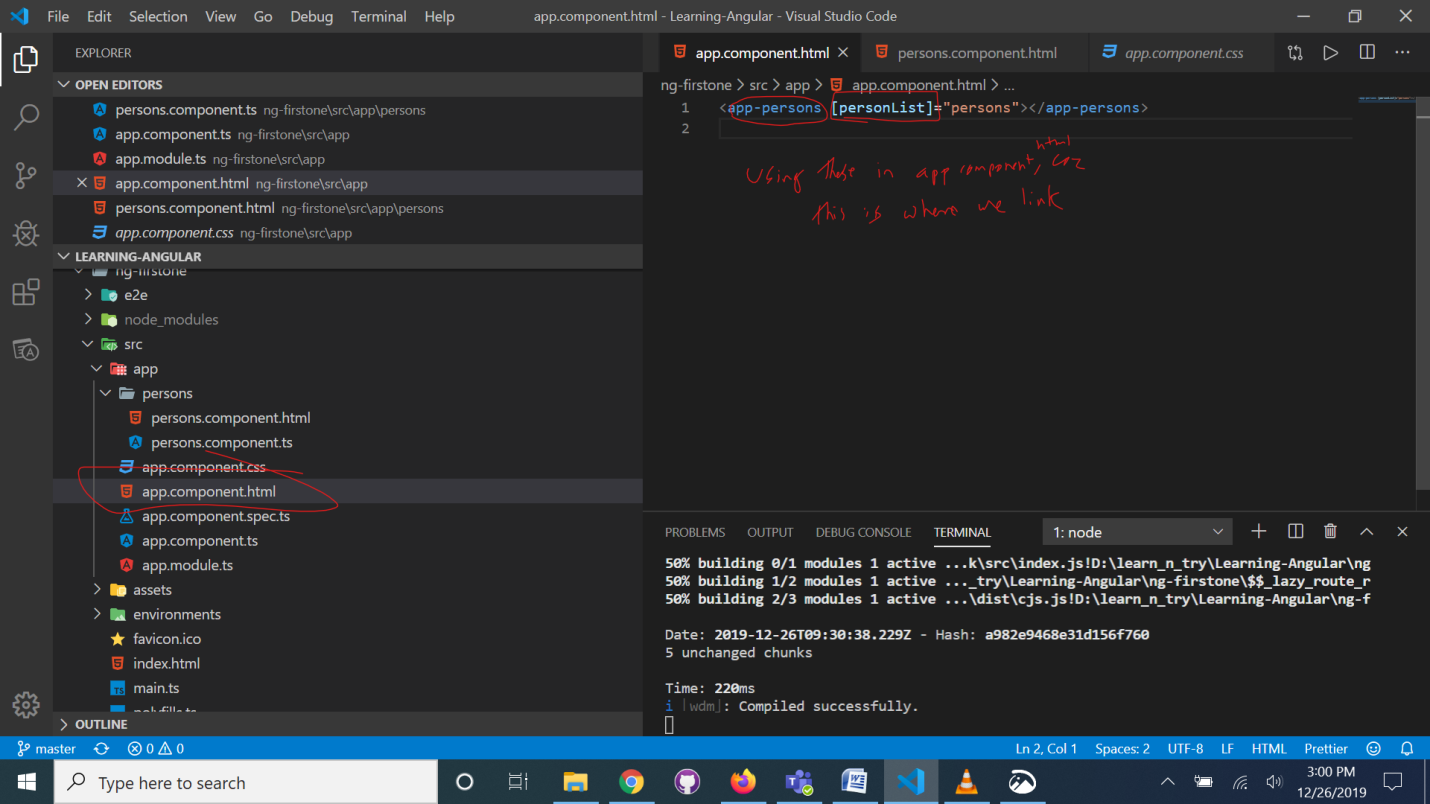
3. App.module.ts-> import the component and add its name in declaration.



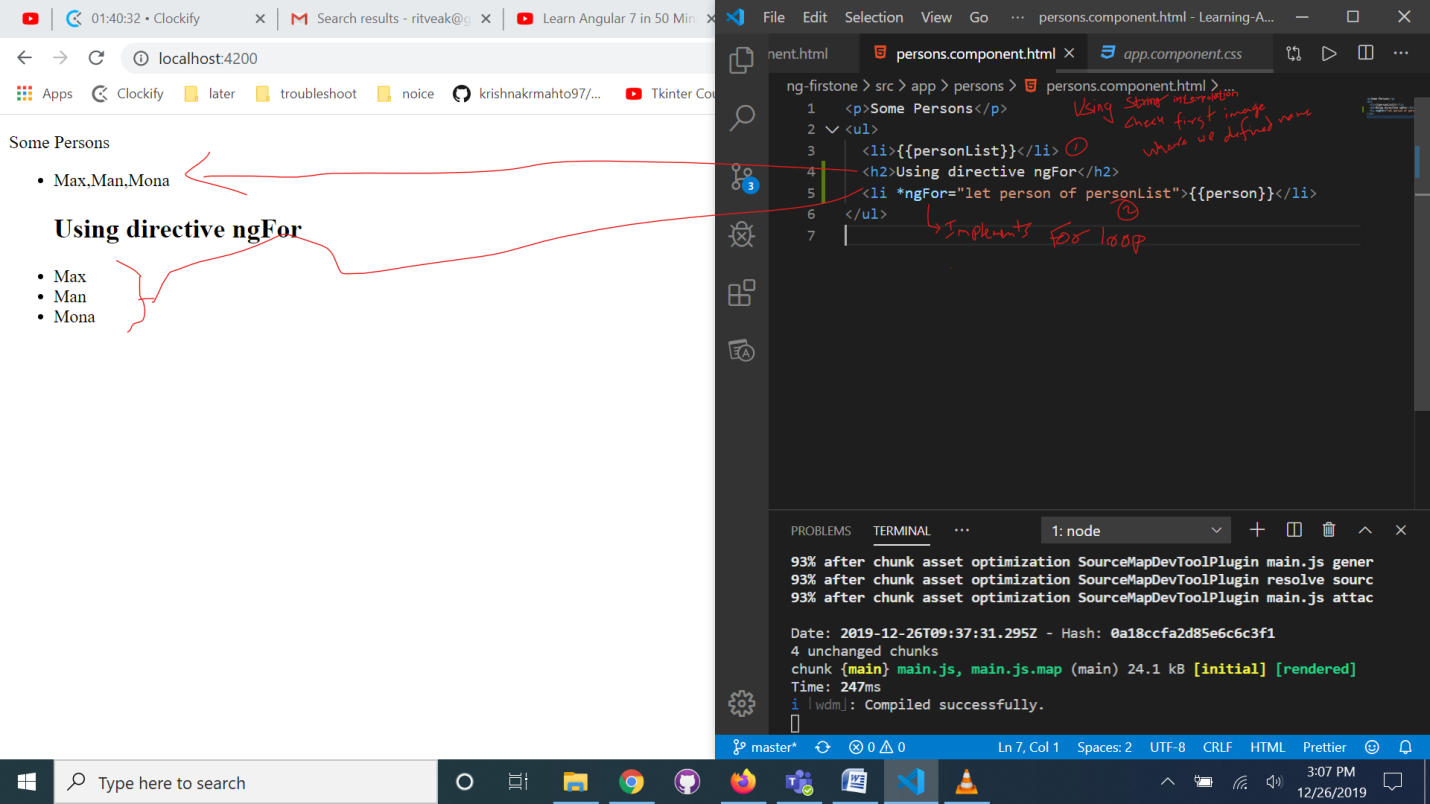
4. Either declare a data in app.component’s export statement or get data from http req, either way it is kept in export section.



5. Use the new component’s tag in app.component.html *(The html file of that component is then connected with the app component by mentioning the new component’s tag in its(app component) html file which in turn has a selector in its own component.ts, which is called in index.html !)*

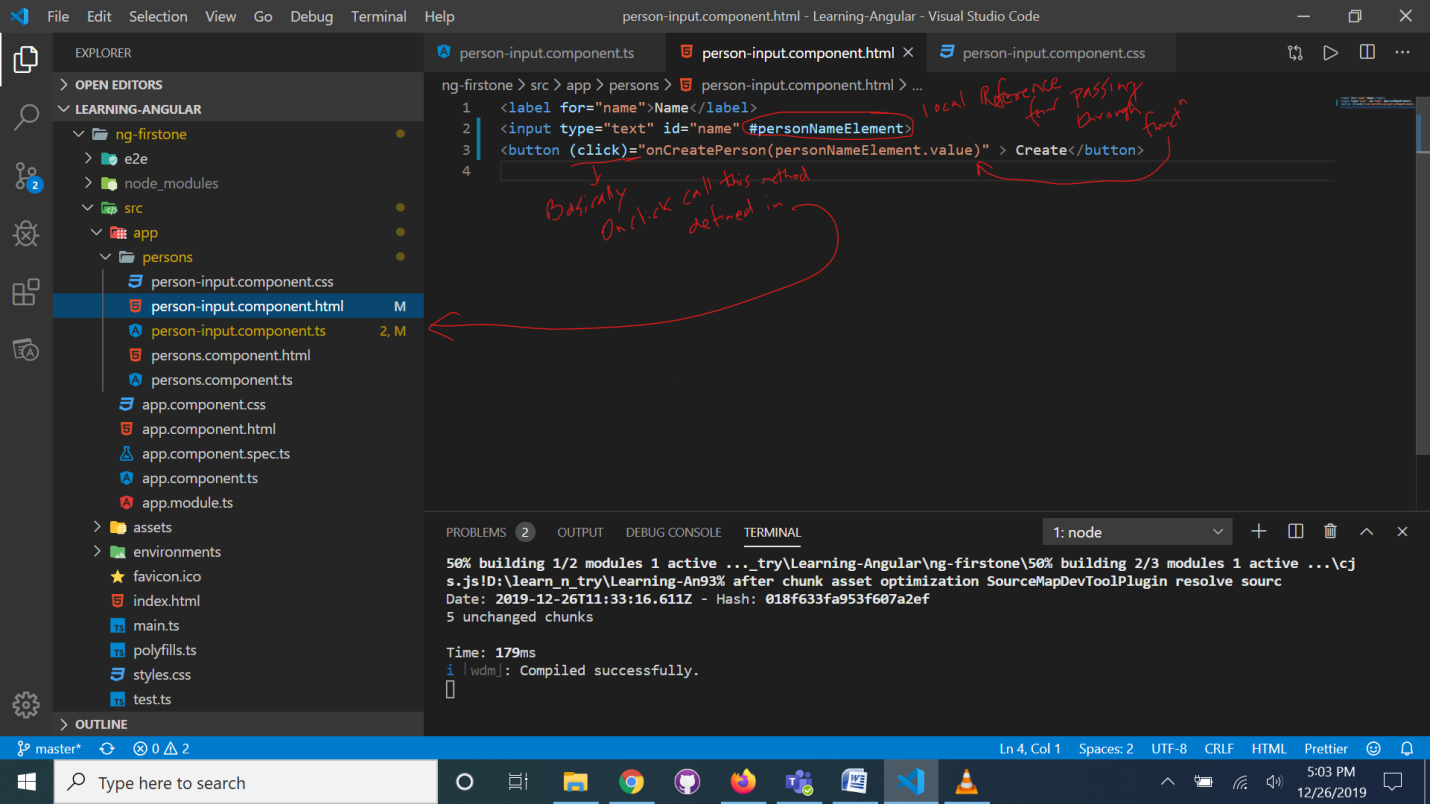


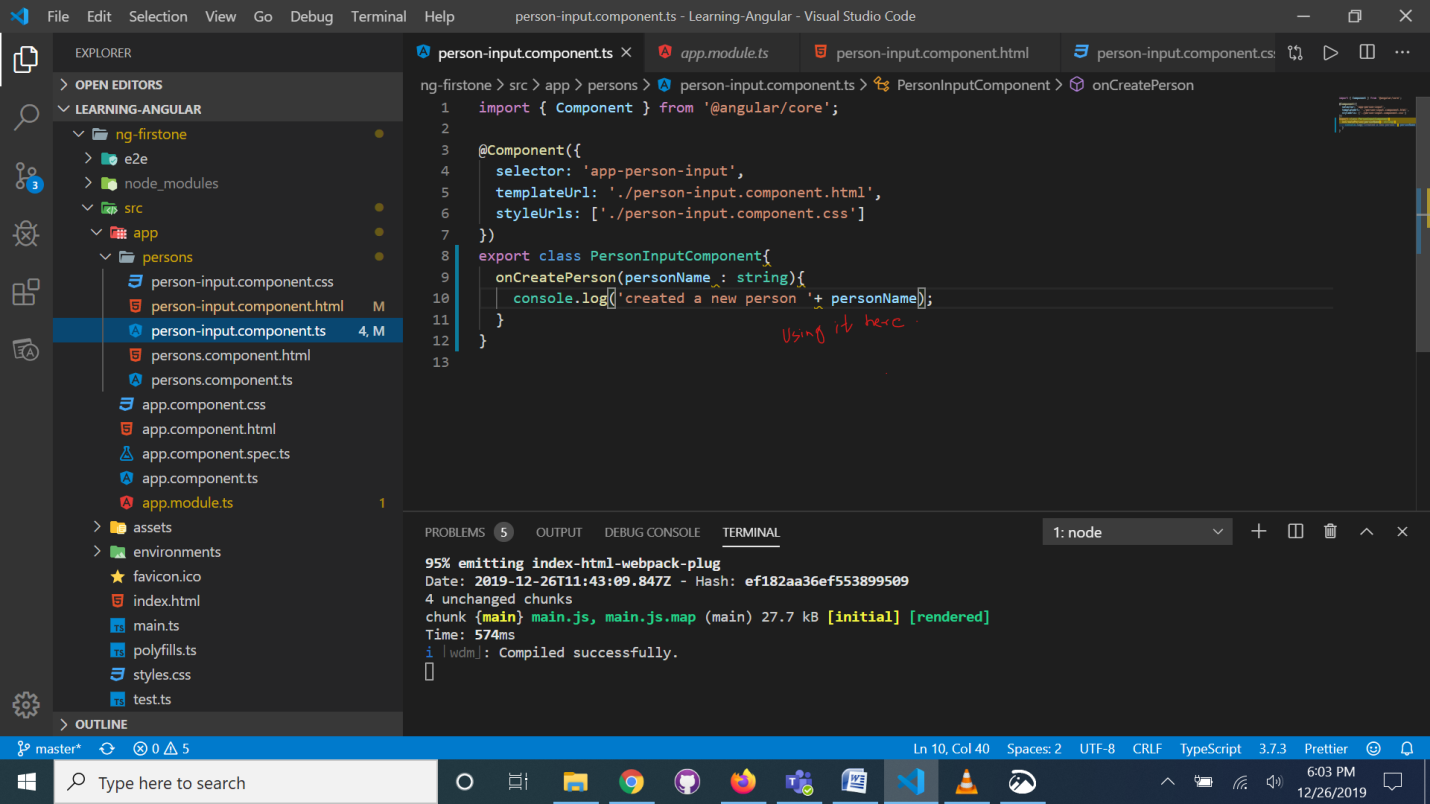
6. Using ngFor we can make a for loop fro displaying elements of array.



TAKING INPUT AND ADDING IT TO THE LIST THAT WE HAVE:

Make three new files (.ts, .html, .css)and connect them as we did earlier(by importing and declaring in app.module.ts):





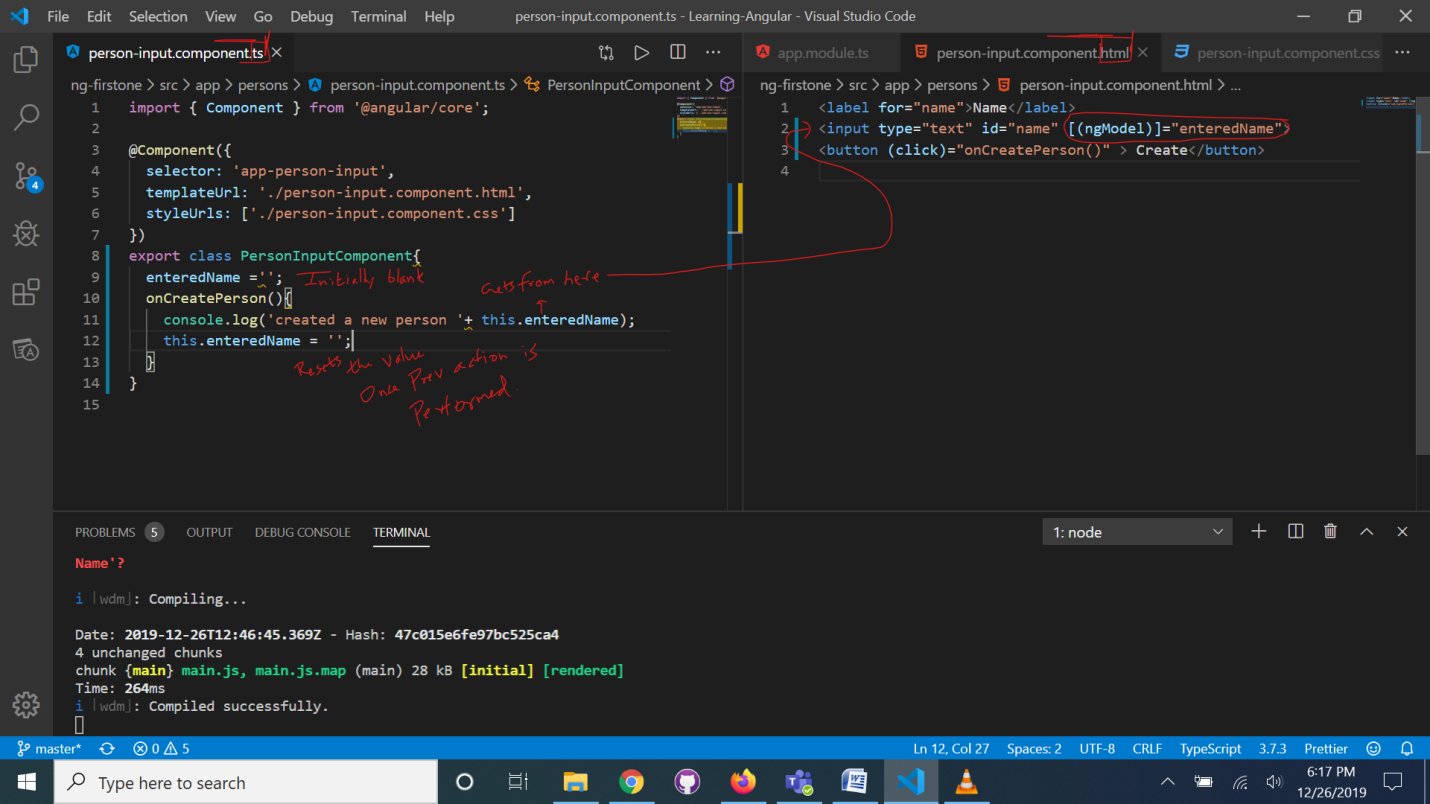
Property binding -> [] eg [personList]

Event binding -> () eg(click)

String interpolation -> {} eg {{person}}

Two way Binding->[()] eg [(ngModel)]

Basically we can change the value of the variable by entering a value and we can reset the value by our code.



For our own property binding we used Input as a decorator.  
For our own event binding we use output as a decorator.

