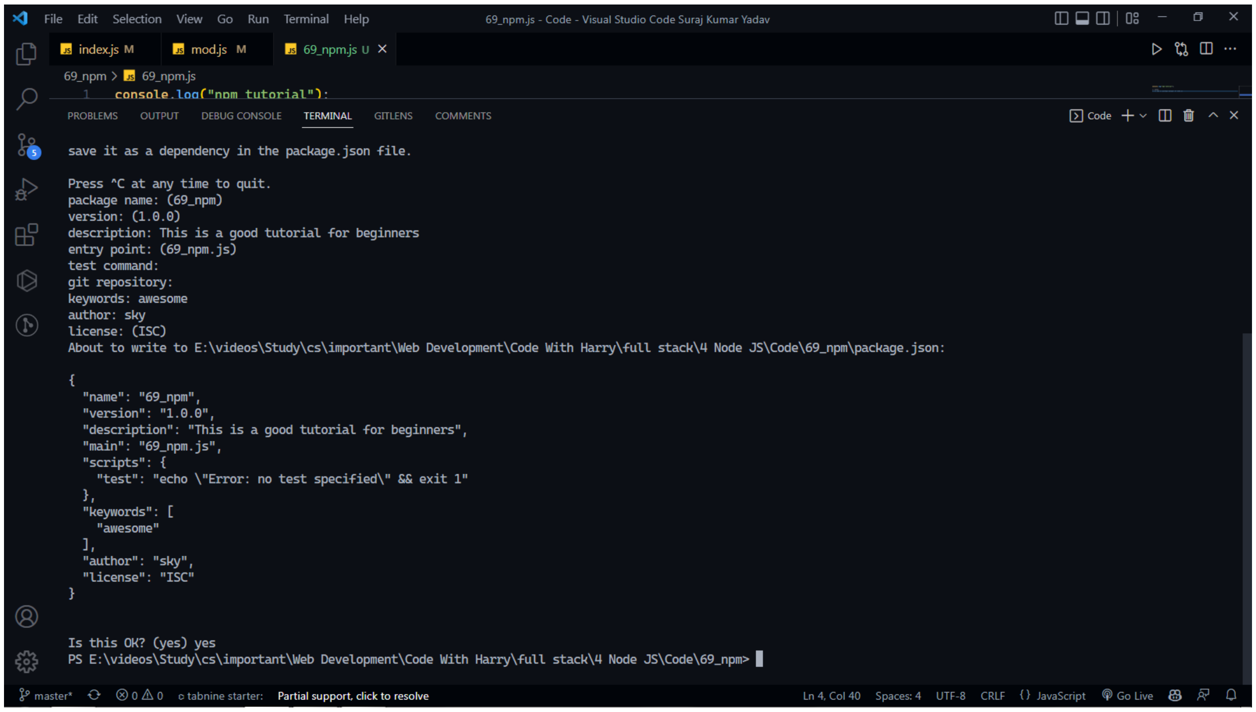
To get version of:

1. npm: npm --version
2. node: node --version
   in terminal

Type “npm init” in terminal and fill the details



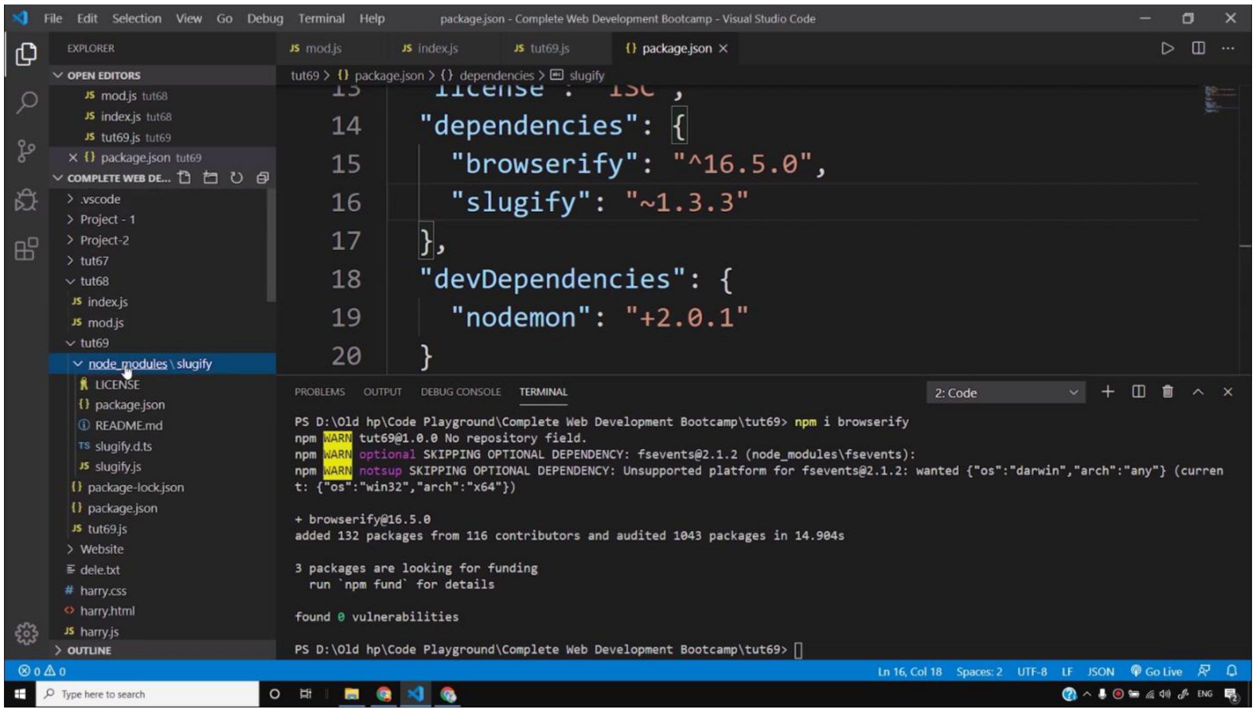
```
console.log("npm tutorial");
```

```
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (69_npm)
version: (1.0.0)
description: This is a good tutorial for beginners
entry point: (69_npm.js)
test command:
git repository:
keywords: awesome
author: sky
license: (ISC)
About to write to E:\videos\Study\cs\important\Web Development\Code With Harry\full stack\4 Node JS\Code\69_npm\package.json:

{
  "name": "69_npm",
  "version": "1.0.0",
  "description": "This is a good tutorial for beginners",
  "main": "69_npm.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "awesome"
  ],
  "author": "sky",
  "license": "ISC"
}


Is this OK? (yes) yes
PS E:\videos\Study\cs\important\Web Development\Code With Harry\full stack\4 Node JS\Code\69_npm>
```

Alternatively, we can any other details

- ➔ “npm init” initializes folder as node package
- ➔ If we install something on node, it will automatically add it to dependencies (inside newly created json file)
- ➔ After installing a package, it will be stored in a folder inside the folder we are working on it named node_modules.
- ➔ **To install a module → “npm install packageName”**
  - o We can simple write “i” in place of “install”
- ➔ **To uninstall a module → “npm uninstall packageName”**
- ➔ If we mistakenly delete the node_modules folder, we simple have to write “npm install” in terminal, it will automatically install all the dependencies present in the json file
- ➔ **To install a particular version of a module → “npm install packName@1.2.3”, where 1.2.3 is version. version is in the form of major.minor.patch**
  - o Patch count increase when bug is fixes or something like that
  - o Minor count increase when new features are added without removing old features
  - o Major count increase when major changes occur like old function is replaced by new function i.e. old one is deprecated

- ➔ It may be possible that package we are installing is also depend on some other package
- ➔ **To install a module as dev dependency → “npm install packName --save-dev”**
    - ○ **This package will only work at development stage not at production stage**
- ➔ **To globally install a package → “npm intall packName --global”**
- ➔ Run a Js file using a package → “packName filename”
- ➔ **To see version of any package: npm view packName version**



```
  "dependencies": {
    "browserify": "^16.5.0",
    "slugify": "~1.3.3"
  },
  "devDependencies": {
    "nodemon": "+2.0.1"
  }
```

```
PS D:\Old hp\Code Playground\Complete Web Development Bootcamp\tut69> npm i browserify
npm WARN tut69@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.2: wanted {"os":"darwin","arch":"any"} (curren
t: {"os":"win32","arch":"x64"})

+ browserify@16.5.0
added 132 packages from 116 contributors and audited 1043 packages in 14.904s

3 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

PS D:\Old hp\Code Playground\Complete Web Development Bootcamp\tut69> 
```

By default symbol before version is ^, but if we change it to ~, it will install any patch released to it i.e. any smaller update see above page for detail.

There is nothing like +, it is misprinted, it should be >, it will install latest version

Packages to install:

1. **Nodemon**
2. **Express**
3. **Slugify**
4. **browerify**

Nodemon → as file changes it will automatically run again that file to implement new change → i.e. refresh every time as file changes.