1. First install VS Code, simply next 🡪 next …. Basic installation process
2. Then install node JS

To check whether you have npm and node js installed or not, type the below in your command prompt

|  |
| --- |
| npm -v  node -v |

1. Then install angular by using the below command

|  |
| --- |
| npm install -g @angular/cli |

1. To create new Angular Project, go to your desired location and type this in CMD or Power Shell

|  |
| --- |
| ng new <<APP NAME>> |

1. Now go inside the app folder

|  |
| --- |
| cd <<APP NAME>> |

1. To check whether angular server is running or not,

|  |
| --- |
| ng serve –open |

1. To stop angular server, press Ctrl + C 🡪 Y 🡪 Enter.
2. In this tutorial we have used Angular material library so, to add angular material library,

|  |
| --- |
| ng add @angular/material |

1. To create a component,

|  |
| --- |
| ng g c components/list |

Where **components** is folder name inside the project 🡪 src 🡪 app, and list is component name, by creating one more folder gets created inside the components folder which contains

* List.component.css
* List.component.html
* List.component.spec.ts
* List.component.ts

1. Similarly create two more components named,

|  |
| --- |
| ng g c components/create  ng g c components/edit |

1. To open the project in VS Code,

|  |
| --- |
| Go 1 step back by using 🡪 **cd..**  Now type “**code .**” it will open your app in the VS code |

1. “app.module.ts”, this is the root module, to import the Router Module

|  |
| --- |
| import {RouterModule, Routes} from '@angular/router';  and to define routes, I have used the below, change it accordingly to your project  const routes: Routes = [  { path: 'create', component: CreateComponent },  { path: 'edit/:id', component: EditComponent },  { path: 'list', component: ListComponent },  { path: '', redirectTo: 'list', pathMatch: 'full' }  ]; |

1. To activate Router module, add in imports section

|  |
| --- |
| Imports: 🡪 RouterModule.forRoot(routes) |

1. App.component.html is the main template where we can define how our components HTMLs should render.

**Bug Track Service: -**

**NOTE: You should give the node.js project name in SMALL LETTERS only.**

1. Go 1 step back in CMD using **cd..** and then make the Bug Track Service folder using **mkdir BugTrackService** command.
2. Now go inside the directory by using **cd .\BugTrackService**
3. Now install dependencies into the BugTrackService,
   1. To install babel (it converts your code into JavaScript that **Node.js** can run and make you happy),

|  |
| --- |
| npm install --save-dev babel-cli babel-preset-env |

* 1. Next add the “.babelrc” file, by right click on BugTrackService 🡪 New File 🡪 “.babelrc”

**Also we can try “New-Item ‘.babelrc’”**

* 1. Now install babel.watch by

|  |
| --- |
| npm install --save-dev babel-watch |

* 1. Now install **express**, “npm install express”
  2. Now install **mongoose library**, “npm install mongoose”
  3. Now install **cors** package, “**npm install cors**”
  4. Now add server.js file and then write the code and to check whether node.js server is running or not type “**npm run dev**”, when you run “localhost:4000” in browser then you’ll see its running there too.

1. To run mongo db use “mongod” command.
2. In mongo DB we create collections how we have tables in SQL Server, the below bold name is the collections name which gets created automatically.

|  |
| --- |
| export default mongoose.model(**'Bug'**, Bug); |

1. To check whether the RESP APIs are working correctly or not we can use a tool called **Postman**, it’s a tool where we can type the URLs and it’ll return the data.
2. So, assume you closed your PC, and again when you open your computer, BAM…!!!! How to check all the above things again?
   1. Run mongo DB, using “**mongod**” command, if you get any error see what you are getting, mostly you’ll get as “**data/db**” not found so create that in the drive where Mongo DB is installed and now it’ll work.
   2. To run node JS server, use command by navigating to Service “**npm run dev**”.
   3. Now you are good to test your APIs through Postman app.

Part 3 of Tutorial:

1. Create a service and through dependency injection we can use it.
2. To create go to BugTrackApp through command line and type the following command where g stands for generate, s service followed by the name of the service.

|  |
| --- |
| Ng g s Bugs |

1. Jkckjhsdk

DELETE SERVICE IS NOT WORKING