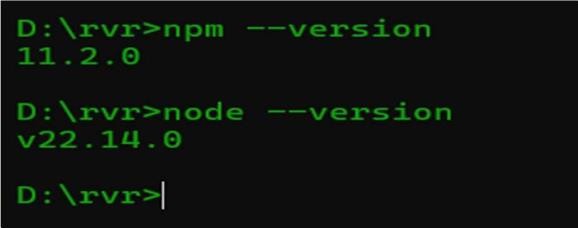
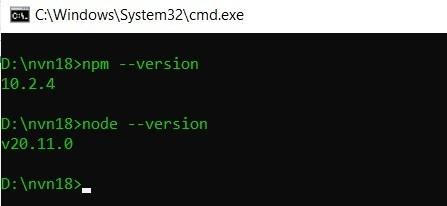


# CS325(JOEL02)::R-20::FULLSTACKDEVELOPMENT



**Lab1:**

**CreateaNode.JSenvironmentwithnodeandnpmutilitiescommandsandtocheckand test the node environment with Node.js Console module.**

* **Step1:installationofNode.jsenvironmentNode**
  1. **DownloadNode.js:**
     1. VisittheofficialNode.jswebsiteusingyour webbrowser.
     2. On the homepage, you'll typically find download links for the latest version of Node.js. If you need a specific version, you might need to visit the "Previous releases" section.
     3. Choose the appropriate installer for your operating system. Node.js provides installers for various platforms including Windows, macOS, and Linux.
  2. **InstallNode.js**:
     1. Oncetheinstaller isdownloaded, locatethedownloadedfile(usually inyour "Downloads" folder).

Followtheinstallationinstructionsspecifictoyouroperatingsystem:

**Windows:**

Double-clickthedownloadedinstaller file.

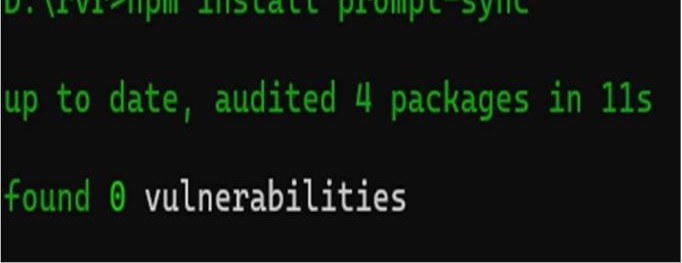
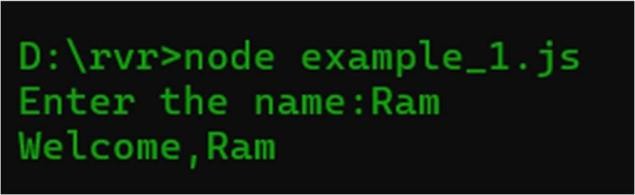
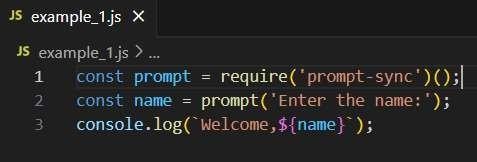
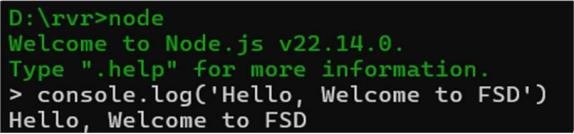
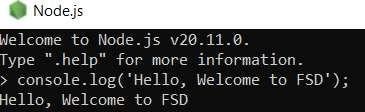
Follow the setup wizard instructions. You can generally accept the defaultsettings, butensurethatthe"npmpackagemanager"optionis selected during installation.

* 1. **VerifyInstallation**:

Afterinstallation,openaterminalorcommandprompt.

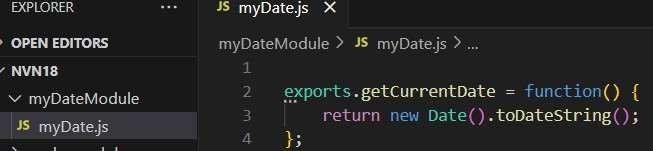
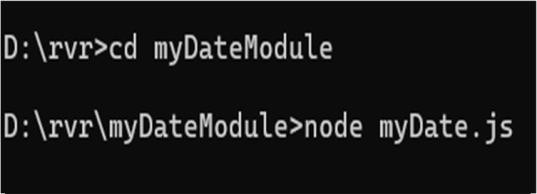
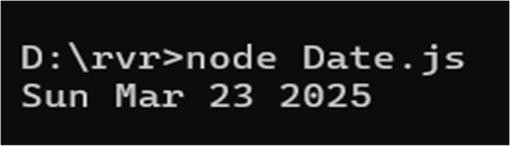
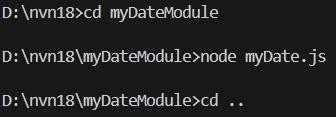
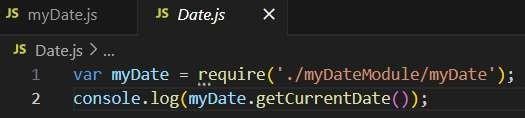
Typethefollowingcommandtoverifythat Node.js and npm areinstalled correctly:

ThesecommandswilldisplaytheversionsofNode.jsandnpminstalledonyour system. If you see version numbers for both, the installation was successful.



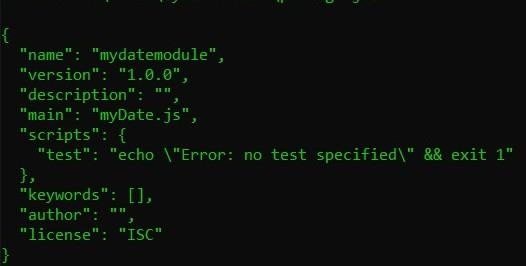
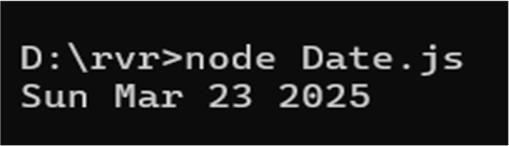
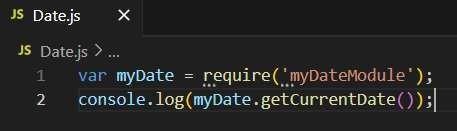
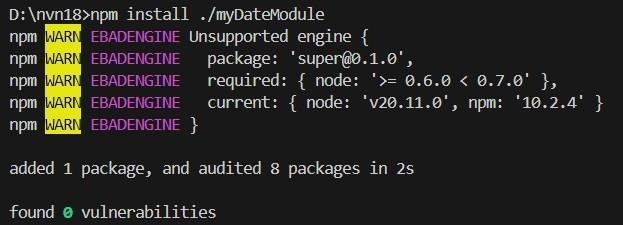
* **Step2:TestthroughthenodeREPLshellcommands**
* **OpenTerminal/CommandPrompt:**Openyourterminalorcommand prompt.
* **CheckNode.jsInstallation**:Typenode-vandpress Enter tocheckifNode.js is installed correctly. It should display the version number.
* **OpenNode.js REPL**:Typenodeandpress Enter toopentheNode.js REPL (Read-Eval-Print Loop).
* **TestNode.jsCommands**:YoucannowtestJavaScriptcommandsdirectlyin the REPL.
* **Step-3:installprompt-syncmoduleusingnpmutility.**
* Installprompt-sync:Inyourterminalor commandprompt,type:
* ​
* **Step-4:Testandchecktheprompt-syncwithconsoleModuleApplication**

# Lab2:



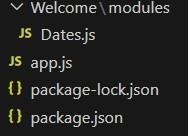
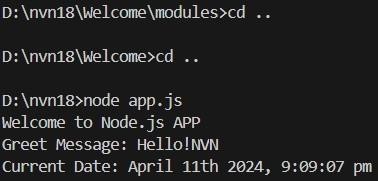
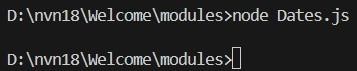
**Create a custom Date module using exports keyword Node module by using npm commands andtodetermineanddisplaycurrentNode.JSWebservertimeanddate.**

* **Step1:CreateNodePackageModule myDate()usingnodeutilitieswithout package.json file**
  1. **Create aDirectory**:Createa directorywhereyou want tostoreyour custom Node module. You can name it something like myDate Module.
  2. **Create myDate.jsFile**:Insidethedirectory,createa JavaScriptfilenamed myDate.js.
  3. **DefinetheModule**:InmyDate.js,defineyourcustomDatemodule.
  4. **ExporttheModule**:Usetheexports keywordtomaketheget Current Date function accessible outside the module.
  5. **UsingtheModule**:Now,youcanusethis moduleinother Node.jsfilesby requiring it:



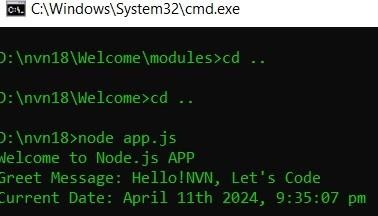
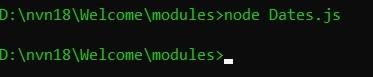
* **Step-2:CreatetheNodePackageModule myDate()usingwithpackage.jsonfile directives like version, name,bin,etc.,**
* Initializea newnpmproject withnpminitThiswillcreatea package.json file.
* Inthepackage.jsonfile,youcanspecifydirectiveslikeversion,name,bin,etc
* **Step–3:Alsoinstallcreatedpackagedmoduleusingnpmutility**
  1. To install the module locally, you can use npm install <folder>. For example,ifyour moduleisinafoldernamedmydate,youwouldusenpm install ./myDateModule.
  2. Tousethemoduleinyour code, youcannowuserequire('myDateModule').

# Lab3:



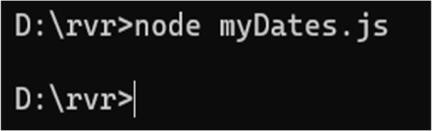
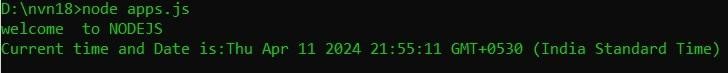
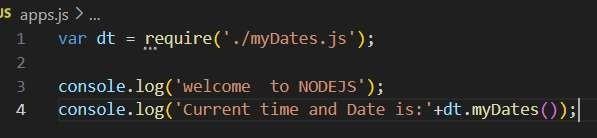
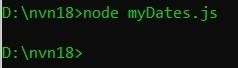
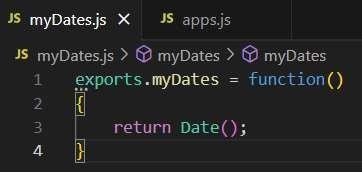
**CreateNodeJSApplicationwithFolderstructureusing npmutilities anddevelopone application to display “welcome Node JS APP” Greet message**

* **Step-1:WithVisualStudioCodeAPPFramework(Anyother)**
  1. Create a Folder named Welcome and Create the another Folder inside the WelcomenamedModules, insidetheModulescreatethefilenamedDates.js in Vs Code.
  2. Createtheapp.js fileat OutsidetheWelcomeFolder andrun theprogram using the node .
* **Step–2:WithoutVisualStudioCodeAPPFramework**



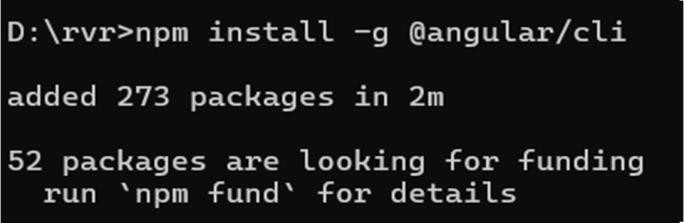
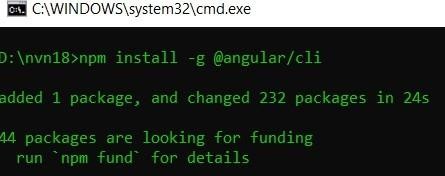
1. Createa DirectorynamedWelcomeandCreatetheanother Directoryinsidethe Welcome named Modules , inside the Modules create the file named Dates.js without Vs Code.
2. OutsidetheWelcome DirectoryCreatetheapp.jsFiletorunthemainprogram**:**

* **Step–3:AlsoAccesstheCustommyDateModule.**



* 1. Createa filenamedmyDates.js,wherethiscodeexportsa functionmyDates.js that returns current date and time.
  2. **Importthe myDates.jsusingtherequireinthe apps.js file:**

# Lab4:

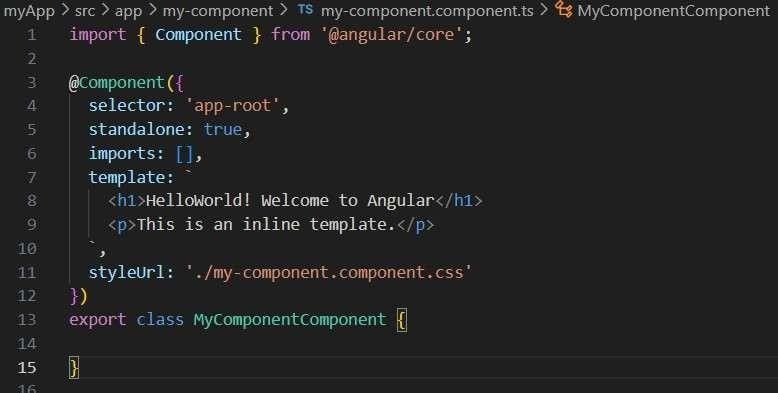


**CreateAngularCLIApplicationswithdifferentcomponentconfigurationsteps using different @Angular ng module utilities at CLI environment.**

* **Step-1:ClasscomponentAngularapp**

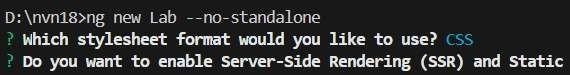
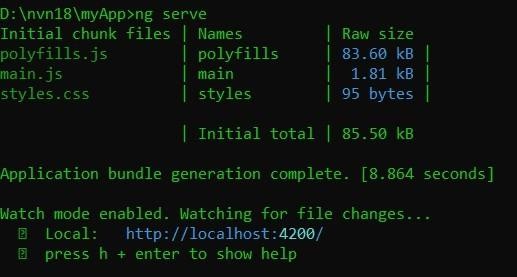
1. Firstinstalltheangularjsusingthecommandpromptasshowinthebelow**:**
2. Createa newAngular ApplicationnamedmyApp withthefollowing syntax**:**
3. CreateanewComponentwithnameofthe myComponent:

* **Step – 2: Define Inline selector component in Angular HelloWorldapp with root element**
  1. Open my-component.component.ts in your Angular application and by default the selector value will be ‘app-component’ but change the selector property to 'app-root'. Your my-component.component.ts file should look like this:
* **Step–3:DefineInlinetemplatecomponentinAngularHelloWorldappwith HTML elements**



* 1. Inmy-component.component.ts, changethetemplateUrlpropertytotemplate and define your HTML elements inline. Your my-component.component.ts file should look like this:
* **Step–4:DefineInlineStylecomponent inAngular HelloWorldapptostyle the color of the message.**

1. Inmy-component.component.ts, changetheStyleUrlpropertytostylesand define your CSS elements inline. Your my-component.component.ts file should look like this **:**
2. Runtheapplicationusingthefollowingcommand:

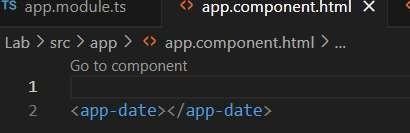
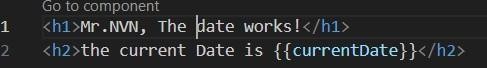
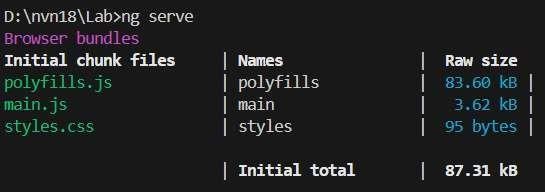
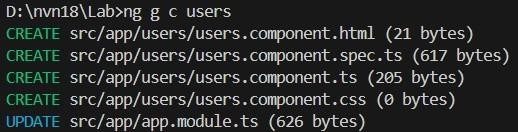


# Lab5:

**CreateAngularCLIApplications usingAngularClasscomponentconstructorsand objects and different variable initialization.**

* **Step–1:CreateDateClassConstructorwithcurrentDateinClassComponent**
  1. **Create anewAngularproject:**youcancreatea newAngular projectby running:
  2. **Create a new component:** Navigate into your new project directory and generate a new component. For example, if we want to create a component named date, we would run:
  3. **UpdatetheComponentClass**:Openthedate.component.tsfileandupdate it as follows:

After theupdationofthisfile,printtheCurrentDateinthe date.component.hmtl file as follows:



Atthispoint, thenewlygeneratedcomponentdateiscompleted, torun this application , add the <app-date> selector in the app.component.html

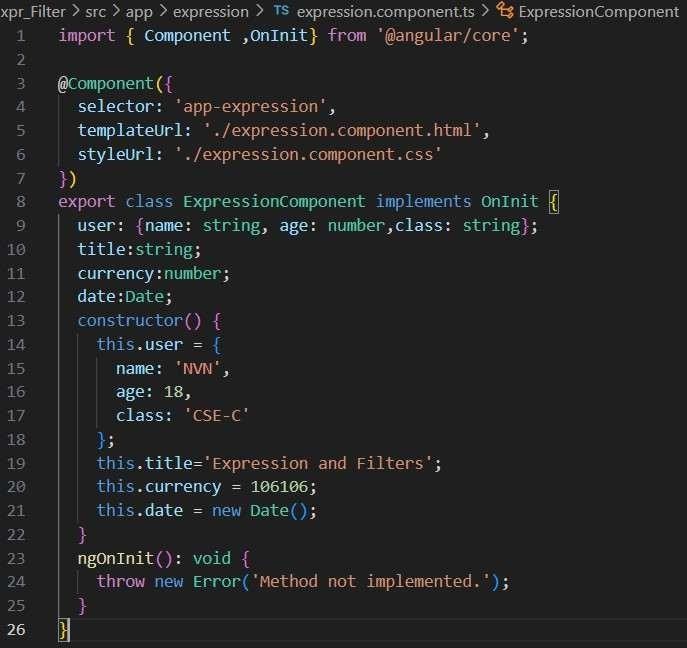
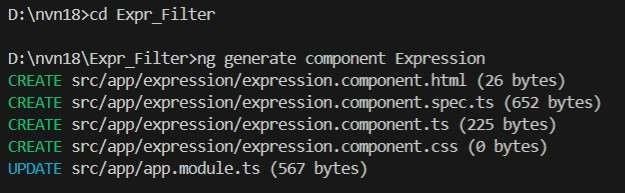
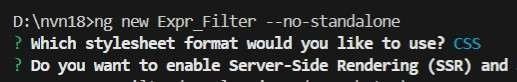
* 1. **RuntheApplication:** Toruntheapplication,usethefollowingcommand:
* **Step–2:ByusingSelector,templateURLandstyleURL Externalcomponent configurations demonstrate the constructor with different objects**

1. **Create anewcomponent:**Navigateintoyour newproject directoryand generate a new component. Here we created the userDetails component:
2. **Updatethecomponentclass:updatetheusers.component.tsfile,like this:**



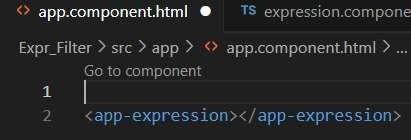
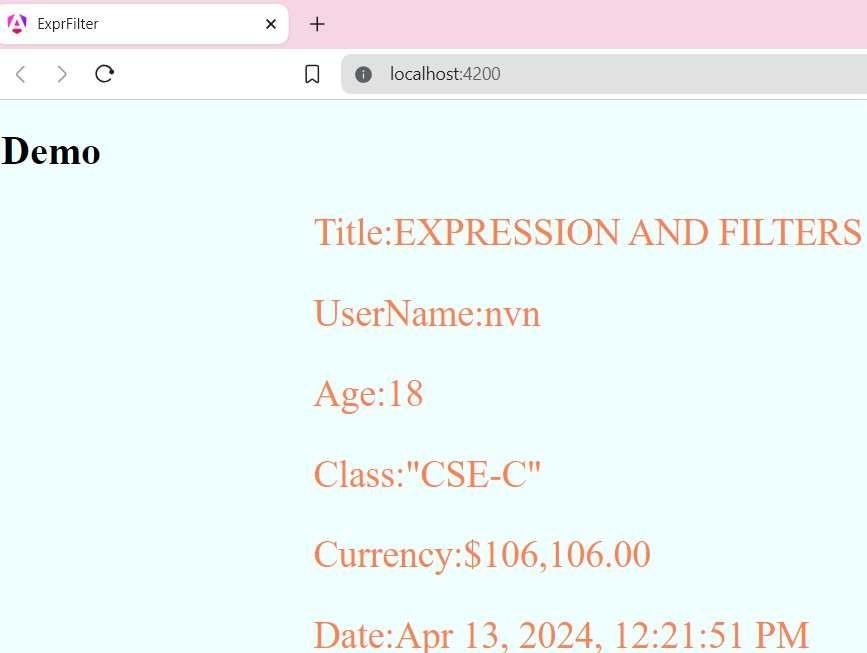
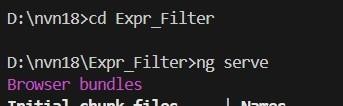
1. **Createthe htmlfile:**updatetheusers.component.htmlfileintheusers folder**.**
2. **RuntheApplication:** Now, ToRuntheApplication,usethefollowing command:

# Lab6:

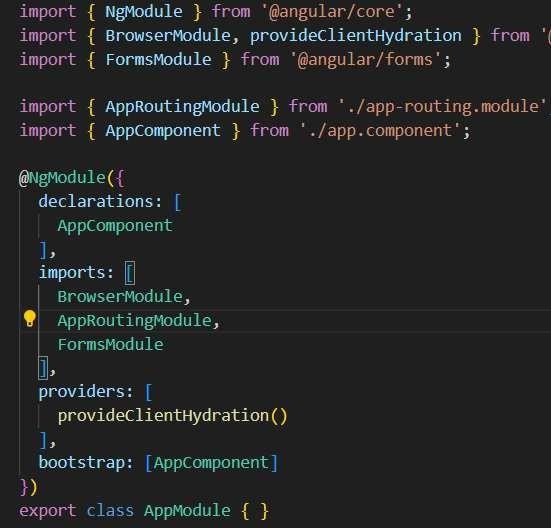
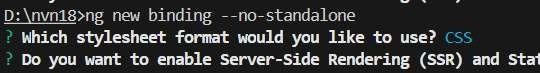


CreateAngularCLIApplications usingAngular ExpressionsandFilterstodemonstratethe one App.

* **CreatedifferentAngularExpressionsinClassComponent**
* **AlsoSpecifywithDifferentAngularpipesorfilterstodemonstrateeachfilter with Angular expression**
  1. Createtheangular applicationwithnameof Expr\_Filter, withthefollowing command :
  2. Createthenewcomponentfor theExpr\_Filter Applicationwithname of Expression , with the following command :
  3. Writetheexpressionandfilters codeintheexpression.component.tsand bindthosevaluesintheexpress.component.htmlfiles,thecodefollowsas:



1. Toexecutethepipes , writethefollowingcodeinthe expression.component.html in the following way:
2. Beforethatlinktheexpression.component.htmlfiletoapp.component.html main file , in the following way:
3. Executetheangularapplication, withthefollowingcommand:



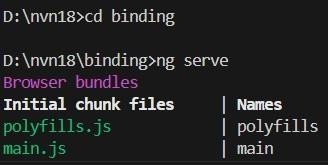
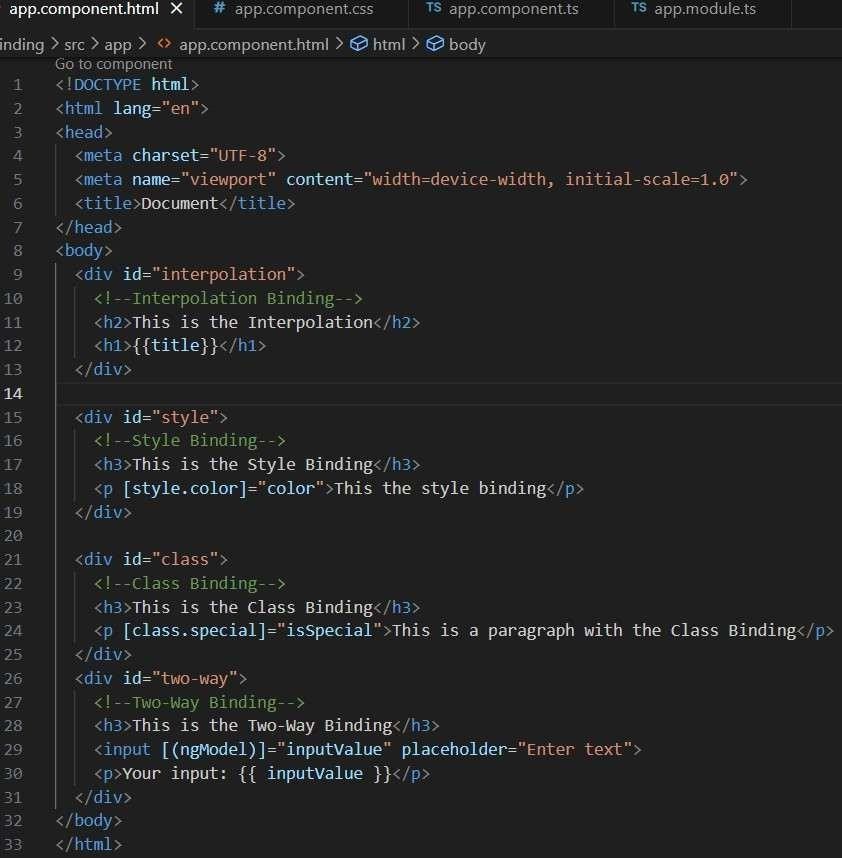
# Lab7:

**CreateAngularCLIApplications usingDataBindingdemonstrateeachbindingtype with form elements.**

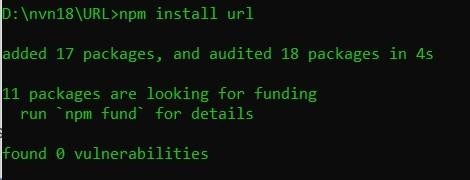
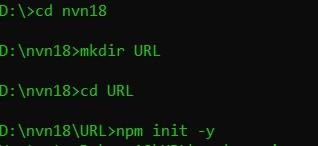
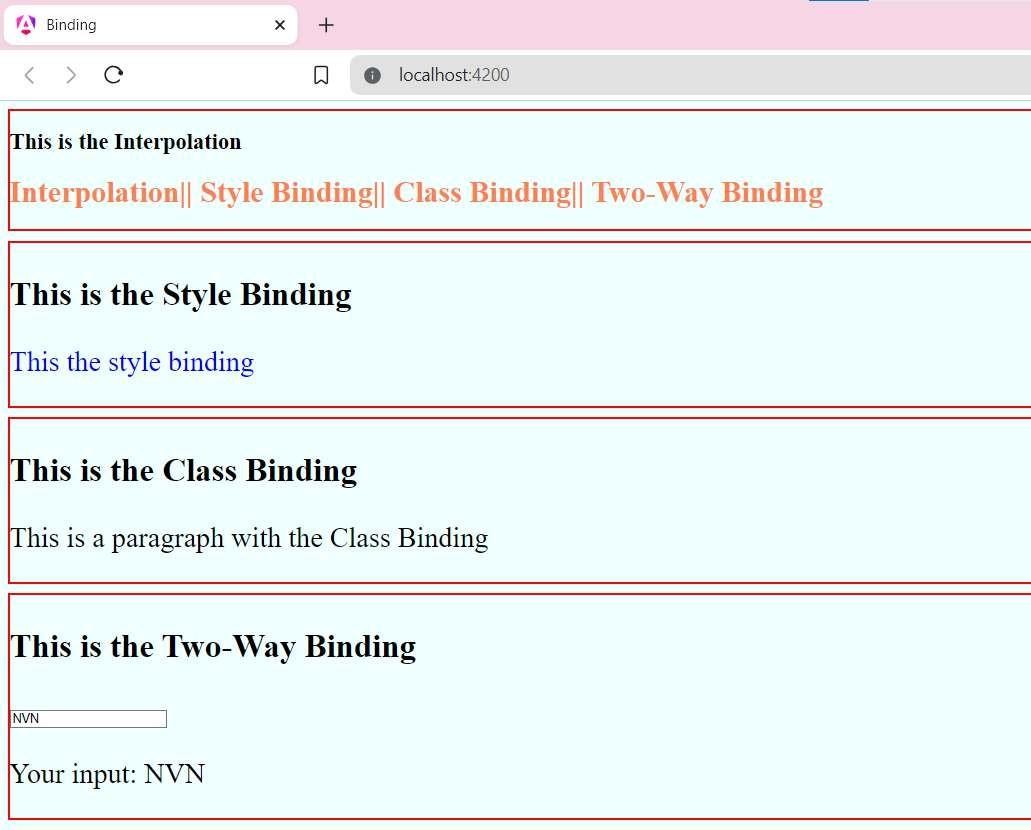
* **InterpolationBinding.**
* **StyleBinding**
* **ClassBinding.**
* **Two–waybinding.**

1. **Createtheangularapplicationwithnamebinding,inthe followingway:**
2. **Intheapp.component.tsfilemakethefollowingchangestodothebinding process:**
3. **Allthebinding moduleswillwork,excepttheTwo-Waybinding,so make the changes in the app.module.ts file.**

**Intheapp.module.tsfile, importtheForms Module inthat file, inthe following way:**



1. **Intheapp.component.html,writethefollowingcodetobindingthe elements into webpage:**
2. **ToRuntheApplicationdothefollowing:**



# Lab8:

Create Node.js Application using URL module to decompose URL Components with urlStr = 'http://user:pass@host.com:80/resource/path?query=string#ha”

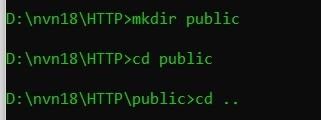
* ResolvingtheURLComponentswithurl.parse()andurl.format()methods
* AlsotoResolvingtheURL usingurl.resolve();

1. **Create aNewNode.jsProject:**Openyour terminalandcreatea newdirectory for your project. Then navigate into that directory and initialize a new Node.js project by running:
2. **InstallRequiredDependencies:**Sincewe'llbeusingbuilt-inNode.js modules, there are no external dependencies to install.
3. **Create the Node.js Script:** Create a new JavaScript file, named **url.js** such as app.js,inyour project directory. Thisfilewill containthecodetodecomposeand resolve URLs.

**Writethecodeinthefollowingways:**



1. **RuntheApplication:Runthe url.jsapplication, inthefollowingway:**

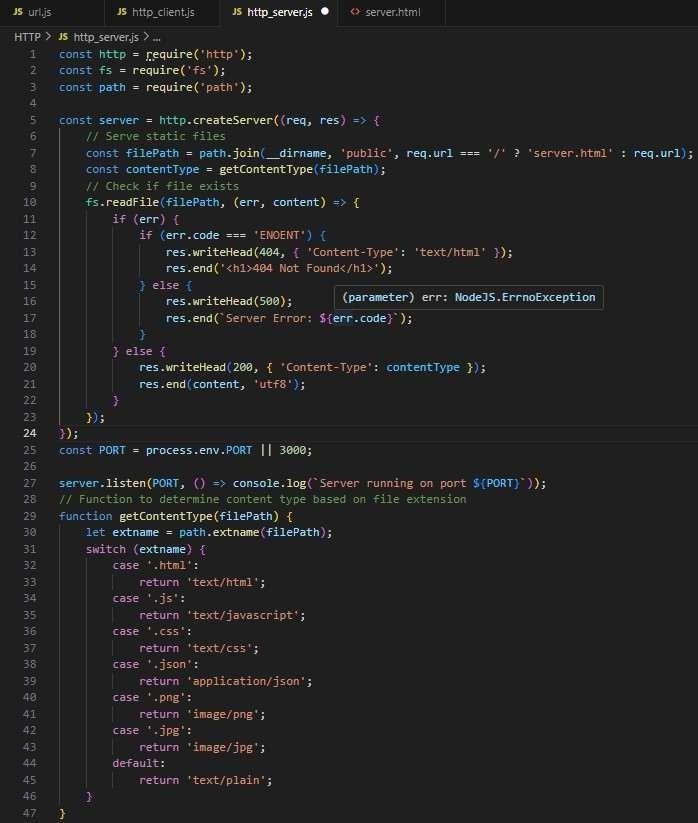


# Lab9:

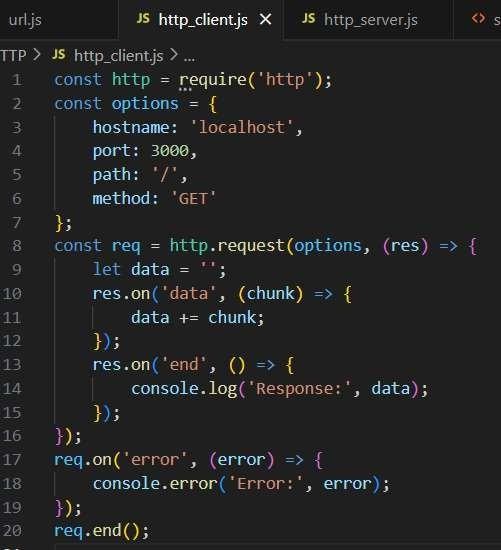
ImplementingHttpServer andHttpClientusinghttpnode.js moduleanddemonstratethe Http Client/server Application.

* CreateHttpStaticserver filesdatausingstaticfiles.
* DefineHttpRequest/HttpResponseobjects.
  1. **Createthe necessaryfiles**:Createa newdirectoryfor your projectandcreate the http\_server.js and http\_client.js files. Also, create a public directory and place server.html file in it. You can put some basic HTML content in server.html.
  2. Createthehttp\_server.jsfilesandhttp\_client.jsfilesandwritethefollowingcode Mentioned below and at the same time create the server.html
  3. **Runtheserver:**Opena terminal,navigatetoyour projectdirectory,andrun node http\_server.js. You should see the message ‘Server is listening on port 3000’.
  4. **Testtheserver:**Opena webbrowser andgoto[http://localhost:3000.](http://localhost:3000/) Youshould see the content of your index.html file.
  5. **Runthe client**:Ina newterminal window(ortab), navigateto your project directory and run node http\_client.js. You should see the content of your server.htmlfileprintedintheterminal. This is theresponsefromtheserver.

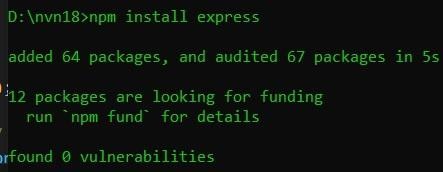
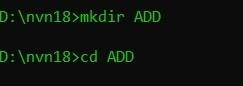
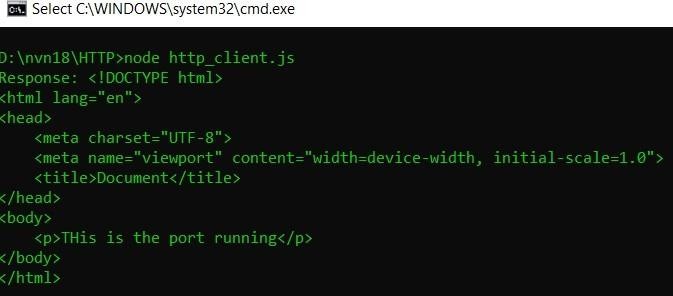
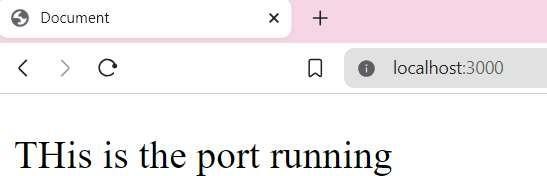
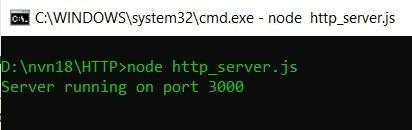
**http\_server.js:**



**http\_client.js:**



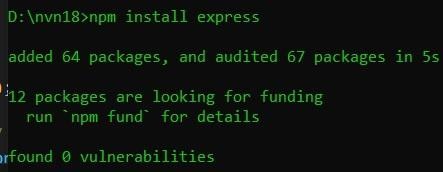
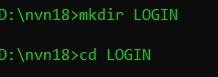
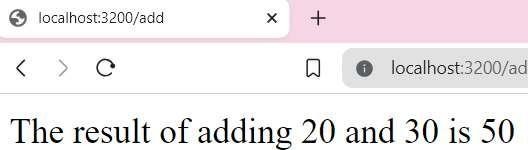
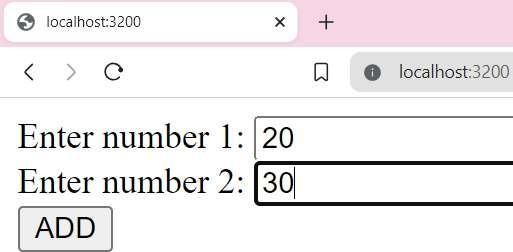
**Server.html:**



# Lab10.

CreateSimpleArithmeticOperations Formwithdifferentforminput elementsN1andN2 text components and ADD button component.

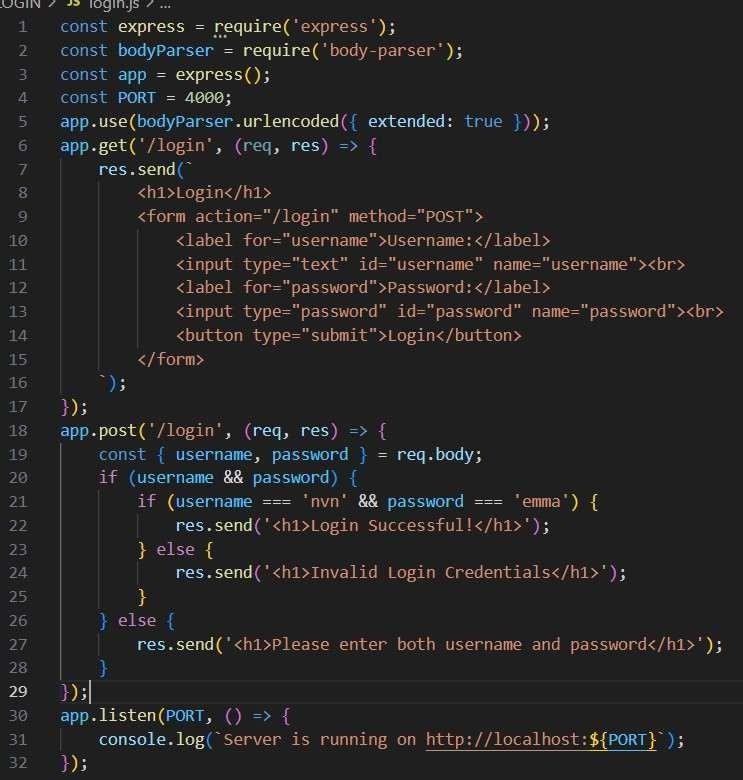
* ProvideExpressServerwithlistenport:3000
* UseExpress.userouteand URLPattern‘/add’
* ProvidedifferentroutingconfigurationseitherPOST orGET
  1. CreatethedirectoryADDandinsidethedirectorycreatethenodeapplication with name of add.js .
  2. WeimportExpress andthebody-parsermiddlewareforparsingformdata.
  3. Wecreateaninstance oftheExpressapplication.
  4. Wesettheportto 3000.
  5. WeusebodyParser.urlencoded()middlewaretoparseURL-encodedformdata.
  6. Wedefinea routefor servingtheHTMLform(/), whichcontainstwoinputfields for numbers (N1 and N2) and a submit button (ADD).
  7. Wedefinea routefor handlingPOSTrequeststo/add. Whentheformis submitted, this route extracts the numbers from the form data, adds them together, and sends the result as a response.
  8. Weoptionallydefinea routefor handlingGETrequeststo/add.Thisrouteis similar to the POST route but expects the numbers to be passed as query parameters instead of form data.
  9. Westarttheserverandlistenonport3000.
  10. Thecodewillbefollowedas:



# Lab11:

CreateSimpleLoginformPageApplicationusingExpressJSModule:

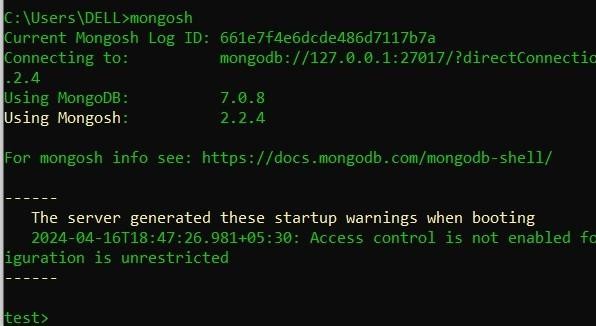
* ProvideExpressServerwithlistenport:4000with URLPattern‘/login’
* Displaytheloginformwithusername,password,andsubmit buttononthescreen.
* Userscaninput thevaluesontheform.
* Validatetheusernameandpasswordenteredbytheuser.
* DisplayInvalidLoginCredentialsmessagewhentheloginfails.
* Showasuccessmessagewhenloginis successful.
  1. CreatethedirectorynamedLOGIN andinsidethedirectory createthefilenamed login.js
  2. WeimportExpress andthebody-parsermiddlewareforparsingformdata.
  3. Wecreateaninstance oftheExpressapplication.
  4. Wesettheportto 4000.
  5. WeusebodyParser.urlencoded()middlewaretoparseURL-encodedformdata.
  6. Wedefinea routefor servingtheloginform(/login). This routedisplaysa form with input fields for username and password, along with a submit button.
  7. Wedefinea routefor handlingPOSTrequests to/login.Thisroutereceivesthe submitted form data, validates the username and password, and sends an appropriate response:
  8. Ifbothusernameandpasswordareprovidedandmatchtheexpectedvalues(in this case, 'admin' and 'password'), it sends a success message.
  9. Ifeitherusernameorpasswordis missing,itsendsamessagepromptingtheuser to enter both.
  10. Iftheprovidedusernameor passwordisincorrect,itsendsanerror message indicating invalid credentials.
  11. Westarttheserverand listen onport4000.
  12. Thecodefollowsas:



**Login.js:**



ram



# Lab12:

**CreateSimpleMongDBServerwithmongodconfigurationdataandalso manage Mongoshell using mongosh :**

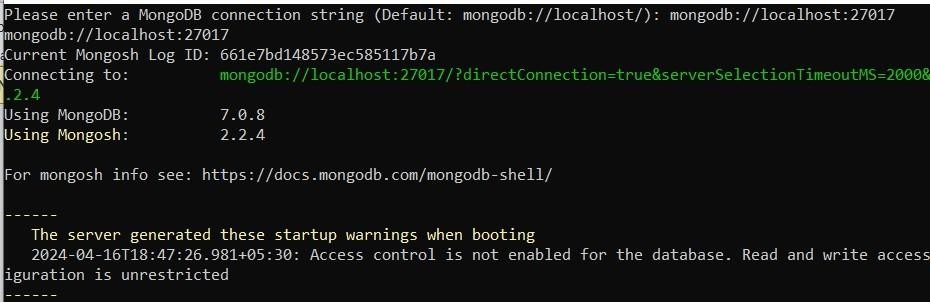
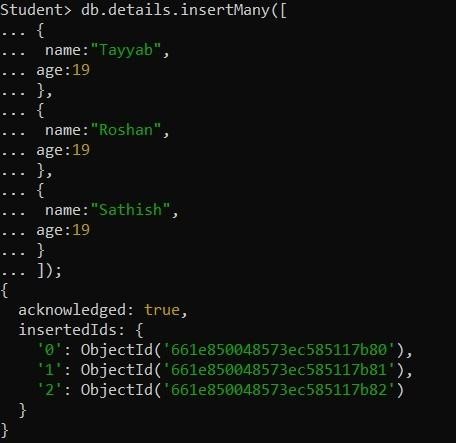
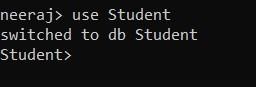
* **CreatesimplestudentdocumentDatabase**
* **Insertonestudentrecordinmongosh**
* **Updateanddeleteonedocumentinmongosh**
* **AlsotoperformconnectionfromMongoDBtonode.jsdriverconnectionstring**

**Step 1:** Install MongoDB First, you need to install MongoDB on your machine.You can download it from the official MongoDB website. After downloading, follow the instructions to install it**.**

**Step 2:** Start MongoDB Server You can start the MongoDB server byrunning the mongod command in your terminal. This will start the MongoDB server on the default port 27017.

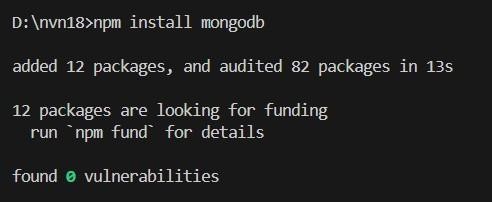
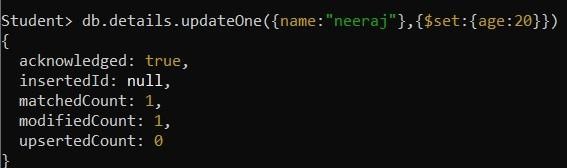
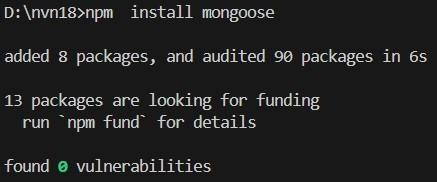
**Step-3:** Connect to MongoDB Server using Mongoshell Open a new terminal window and connect to the MongoDB server using the mongo command. This will start the MongoDB shell (mongosh).

Step – 4: Create the DataBase named the Student in the mongosh shell using the following command. Before creating the database first establish the connection with the MongoDB shell and place the localhost address in the mongoshell .



**Step–5**:Insertthevaluesintothedocumentsunder thedatabasenamedStudent:

**Step-6:**Updatethevaluesinthedocumentsnameddetailsunder thedatabase Student.



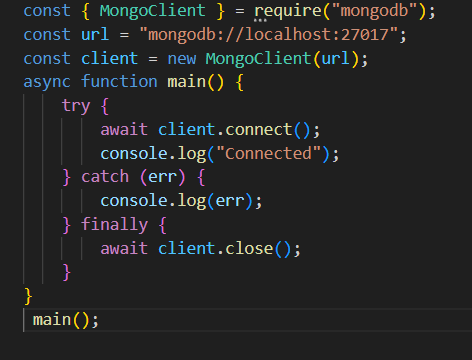
**Step-7:**Deletetheanyvalues inthedetailsintheDatabasenamedStudent:

**Step–8:**ConnecttheMongoDBserver toNodejsApplication,firstyou needto install the mongodb in the nodejs using the npm

YoucanusetheMongoClientobjecttoconnecttoyourMongoDBserver:

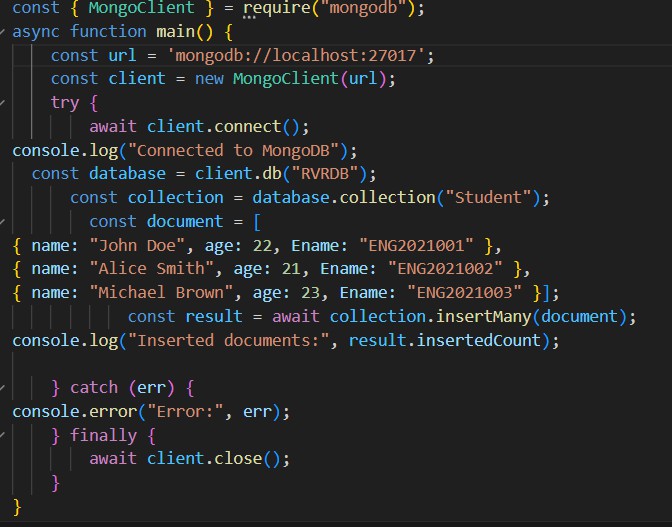
1. Createthemongoo.jsapplicationandwritethefollowingcode
2. Makesurethatthemongoserverisrunningandmaketheconnections
3. Printallthevaluesinthedatabase.





**OUTPUT**:

C:\Users\rvr>nodeapp.js Connected

**InsertthedatausingMongoClientobject:**

**OUTPUT:**

**C:\Users\rvr>nodeapp.js Connected to MongoDB Inserted documents: 3**