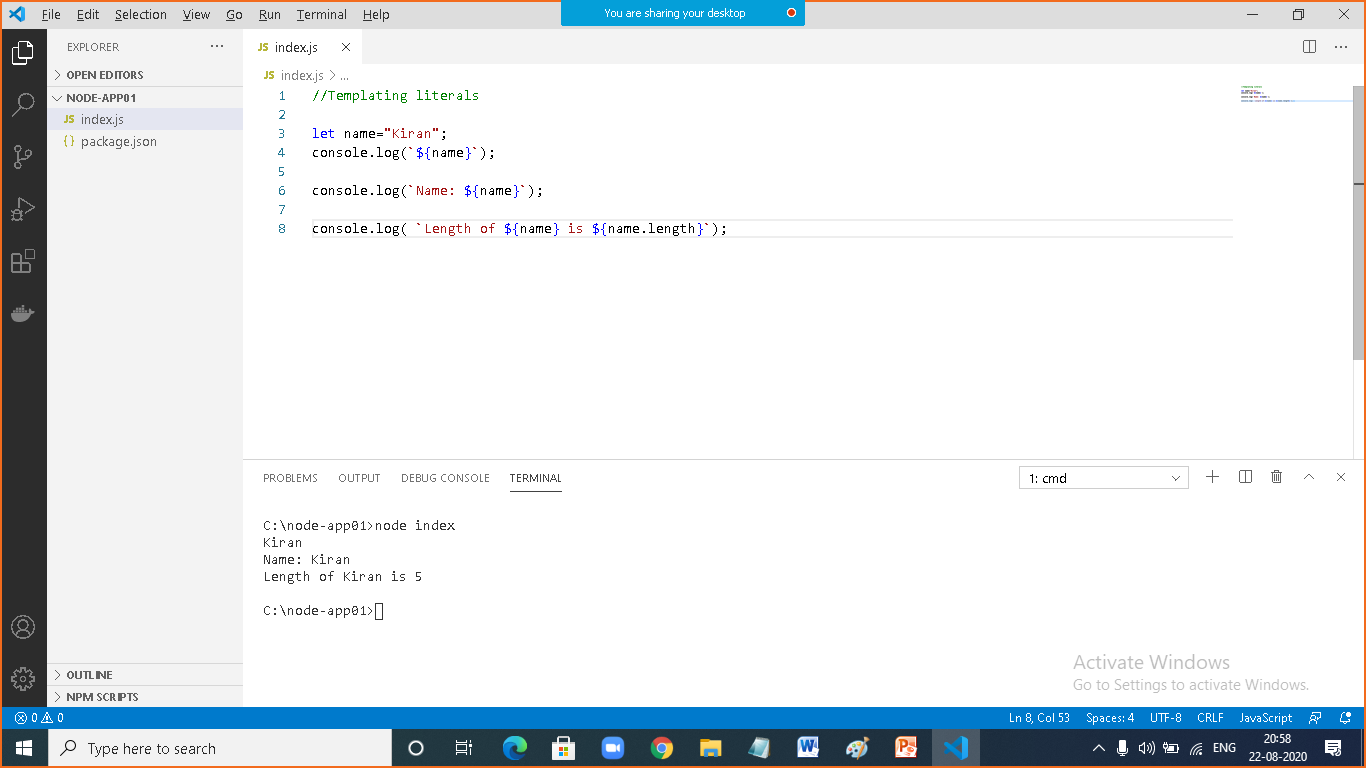
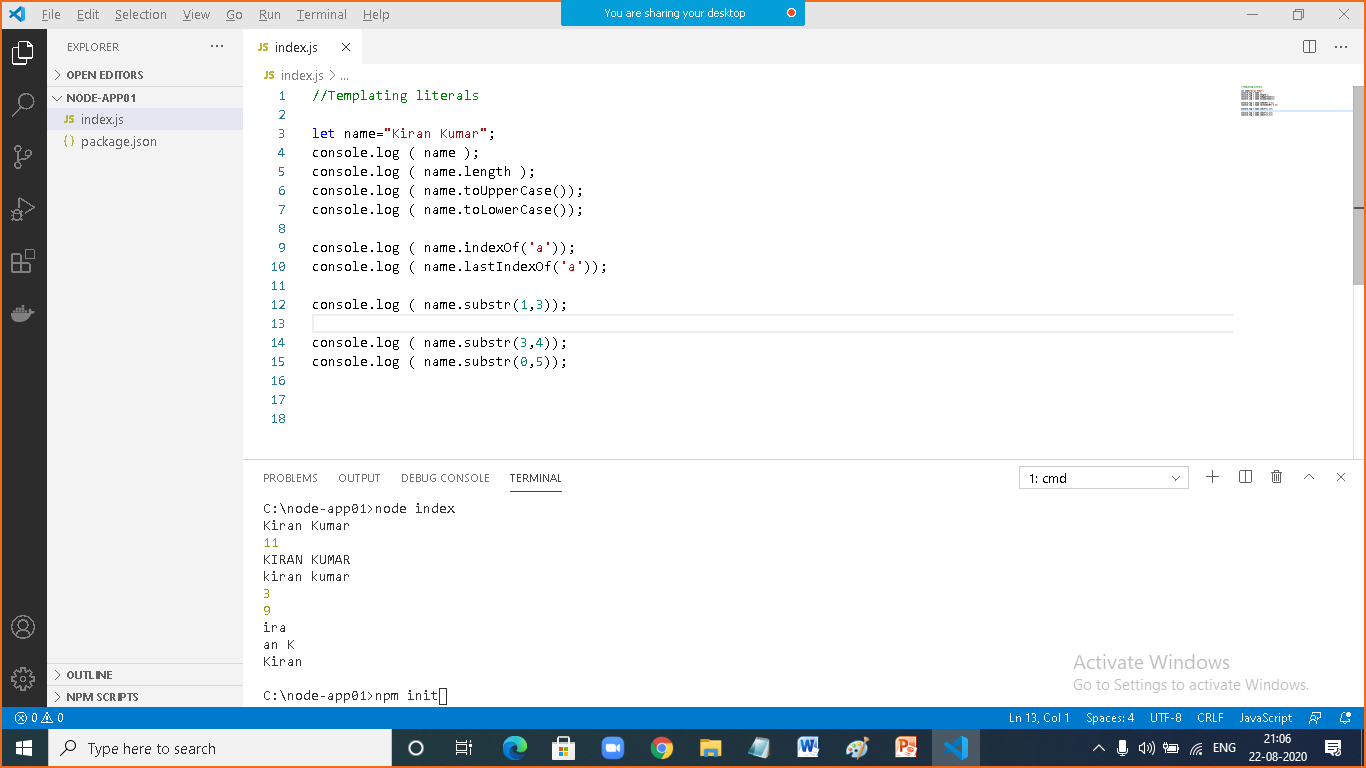


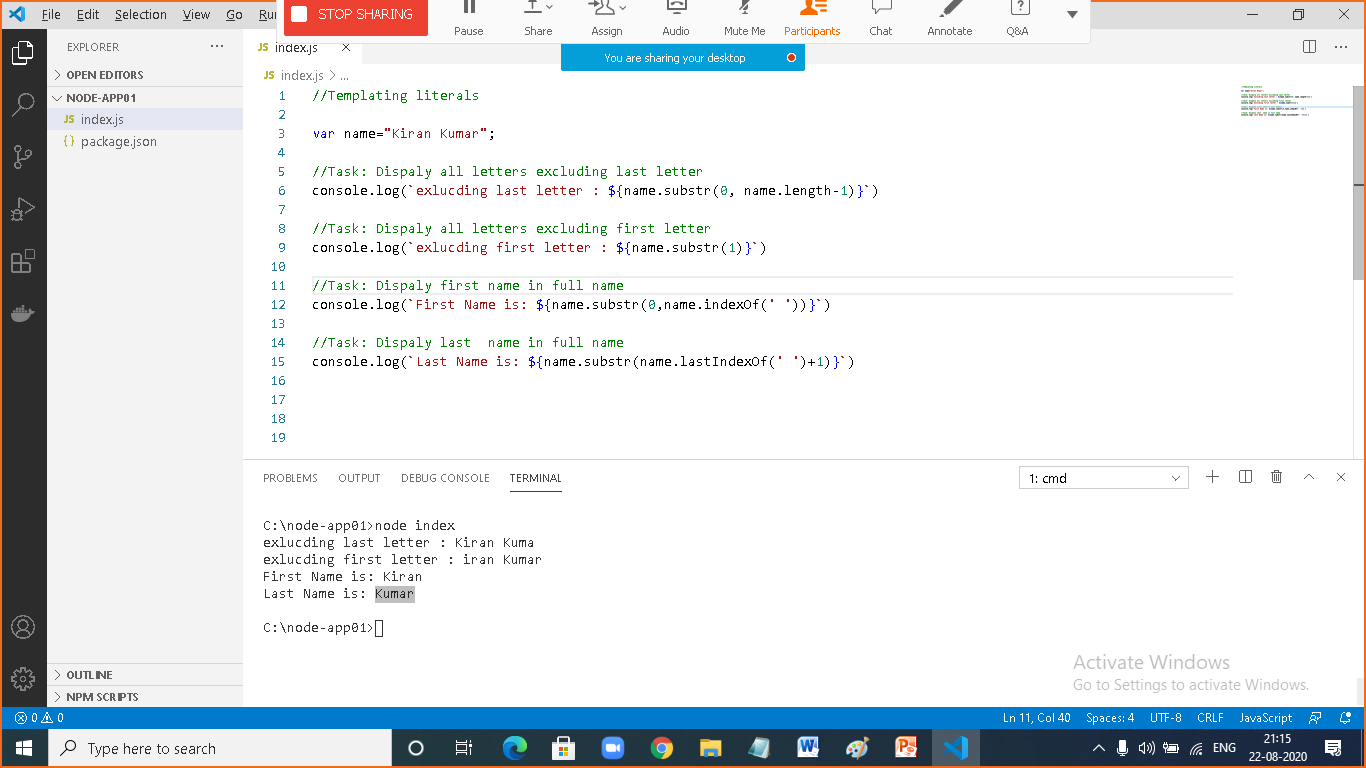
|  |  |
| --- | --- |
|  |  |
|  | var v1=100;  console.log("v1= "+v1+", data type is "+ typeof v1);  var v2="Kiran";  console.log("v2= "+v2+", data type is "+ typeof v2);  var v3=true;  console.log("v3= "+v3+", data type is "+ typeof v3);  var v4;  console.log("v4= "+v4+", data type is "+ typeof v4); |



|  |  |
| --- | --- |
|  |  |
|  | //Templating literals  let name="Kiran";  console.log(`${name}`);  console.log(`Name: ${name}`);  console.log( `Length of ${name} is ${name.length}`); |

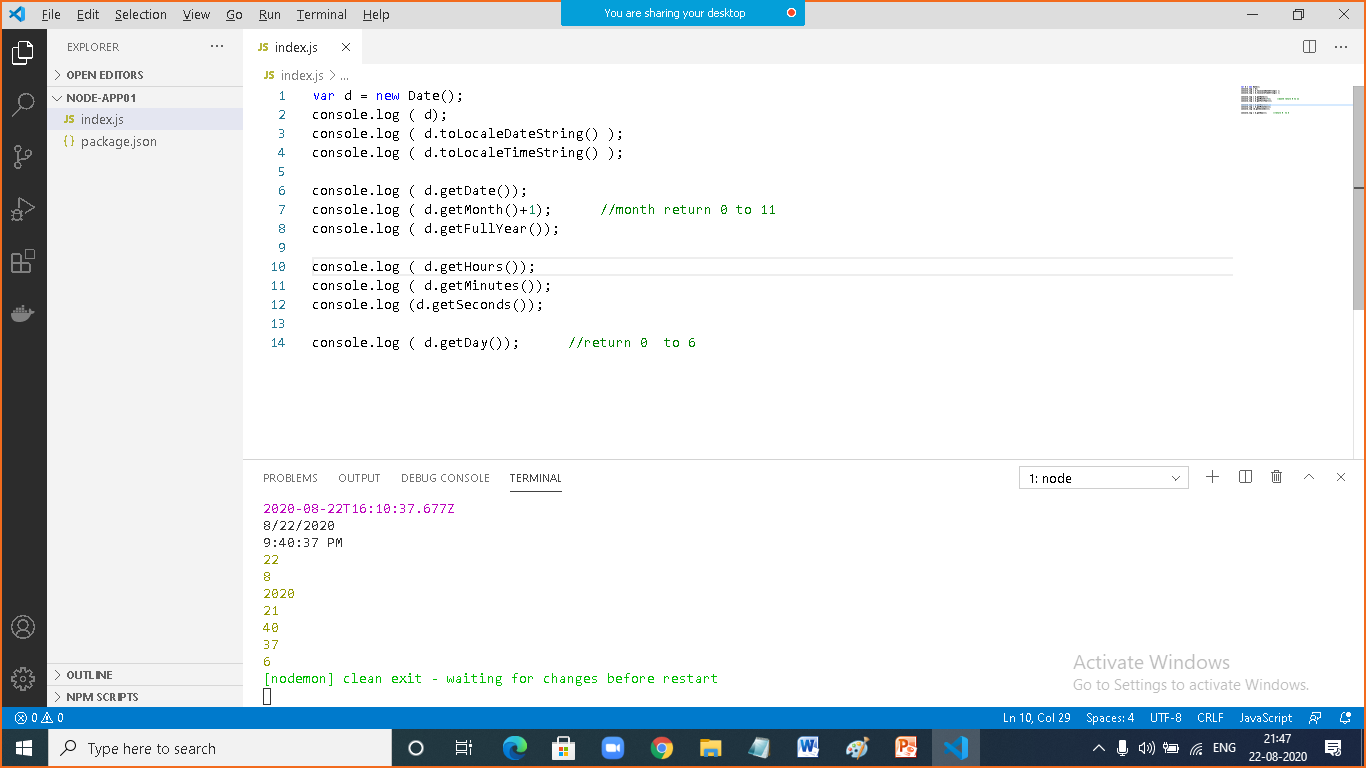


|  |  |
| --- | --- |
|  |  |
|  | //Templating literals  let name="Kiran Kumar";  console.log ( name );  console.log ( name.length );  console.log ( name.toUpperCase());  console.log ( name.toLowerCase());  console.log ( name.indexOf('a'));  console.log ( name.lastIndexOf('a'));  console.log ( name.substr(1,3));  console.log ( name.substr(3,4));  console.log ( name.substr(0,5)); |



|  |  |
| --- | --- |
|  |  |
|  | //Templating literals  var name="Kiran Kumar";  //Task: Dispaly all letters excluding last letter  console.log(`exlucding last letter : ${name.substr(0, name.length-1)}`)  //Task: Dispaly all letters excluding first letter  console.log(`exlucding first letter : ${name.substr(1)}`)  //Task: Dispaly first name in full name  console.log(`First Name is: ${name.substr(0,name.indexOf(' '))}`)  //Task: Dispaly last name in full name  console.log(`Last Name is: ${name.substr(name.lastIndexOf(' ')+1)}`) |

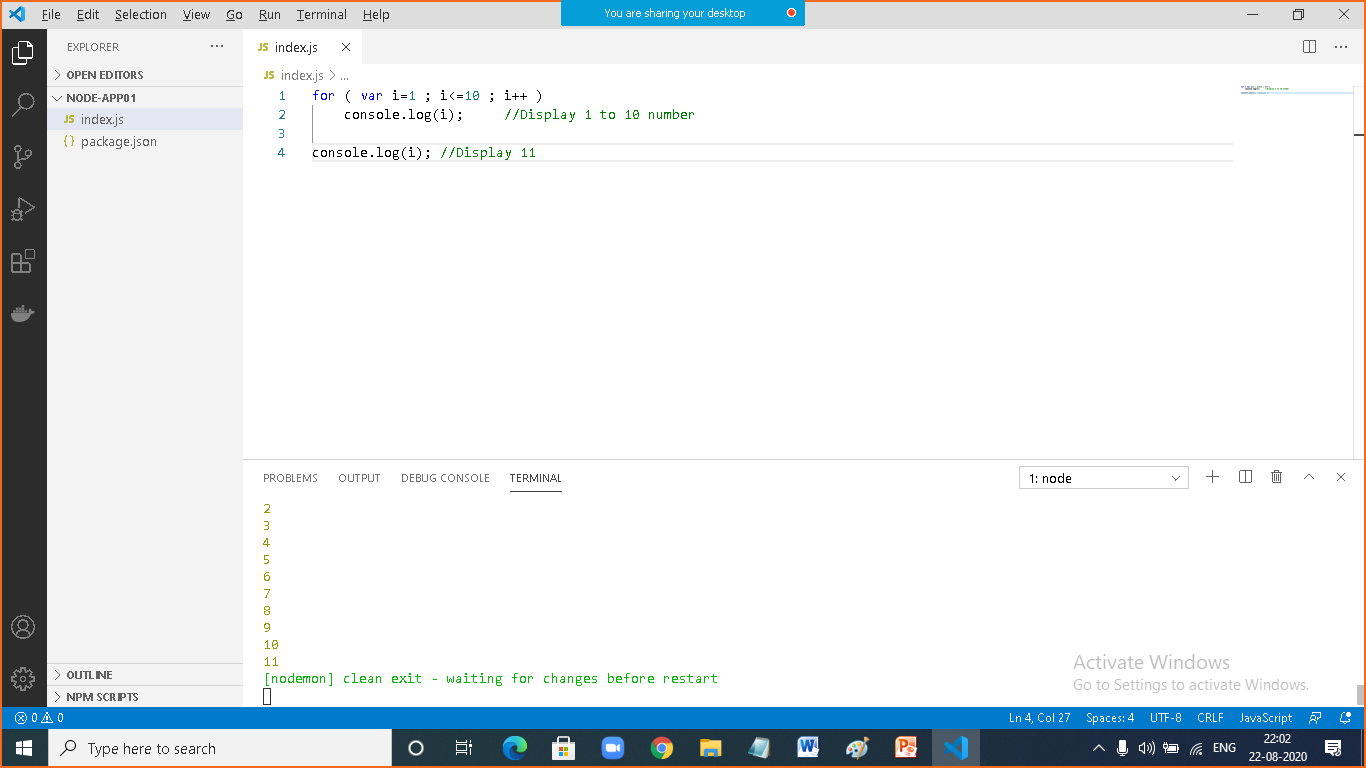
|  |  |
| --- | --- |
|  | **nodemon** |
|  | This is an external module, which is used to run the javascript file in watch mode  Any changes made in the source will effect in output  To install: npm install -g nodemon |

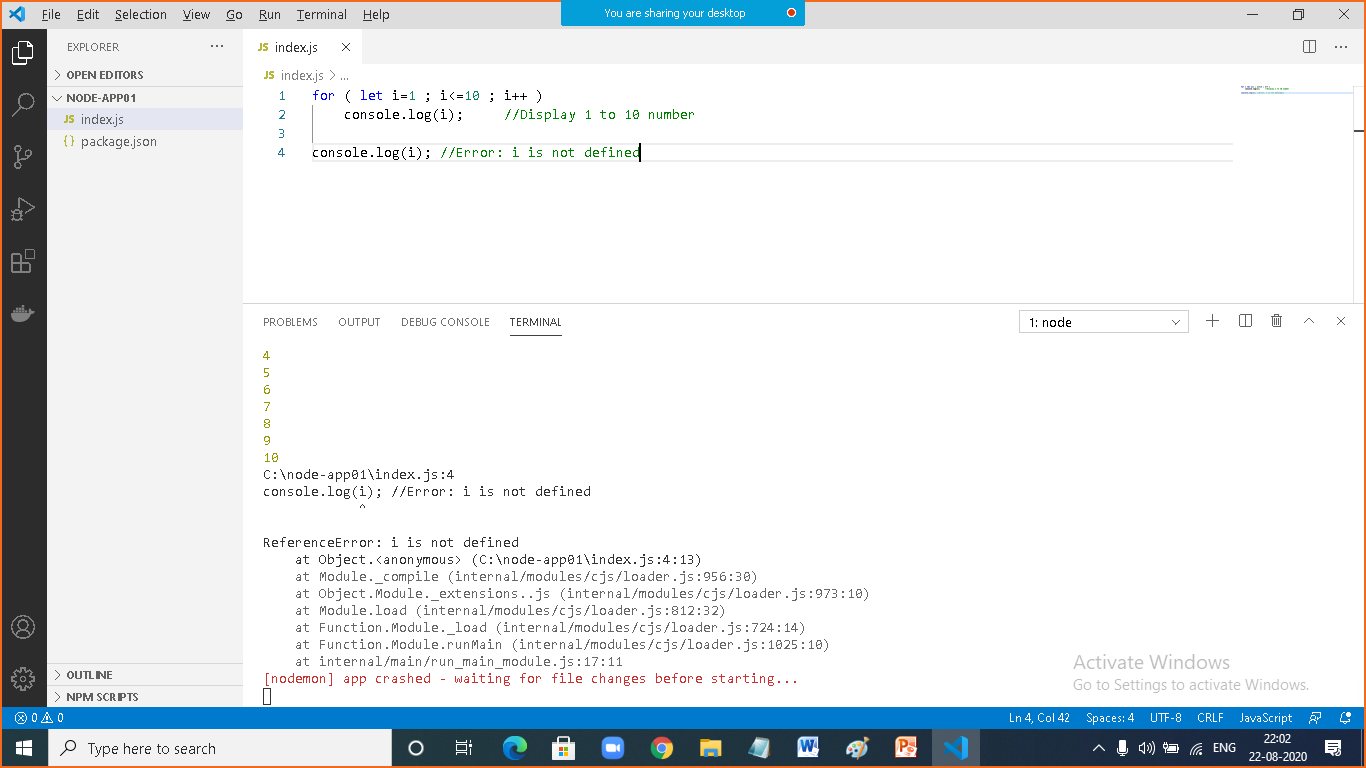


|  |  |
| --- | --- |
|  |  |
|  | var d = new Date();  console.log ( d);  console.log ( d.toLocaleDateString() );  console.log ( d.toLocaleTimeString() );  console.log ( d.getDate());  console.log ( d.getMonth()+1); //month return 0 to 11  console.log ( d.getFullYear());  console.log ( d.getHours());  console.log ( d.getMinutes());  console.log (d.getSeconds());  console.log ( d.getDay()); //return 0 to 6 |

|  |  |
| --- | --- |
|  |  |
|  | //Display wishes based on time  var hour = new Date().getHours();  if ( hour<12 )  console.log("Good Morning!");  else if ( hour<16)  console.log("Good Afternoon!");  else console.log("Good Evening!");  var day =new Date().getDay()+1;  console.log("Today is: ");  switch (day ){  case 1:  console.log("Sunday");  break;  case 2:  console.log("Monnday");  break;  case 3:  console.log("Tuesday");  break;  case 4:  console.log("Wednesday");  break;  case 5:  console.log("Thursday");  break;  case 6:  console.log("Friday");  break;  case 7:  console.log("Saturday");  break;  }  if ( day==1 || day==7)  console.log("Today is weekend, enjoying with friends");  else  console.log("Today is weekday and busy with office works"); |

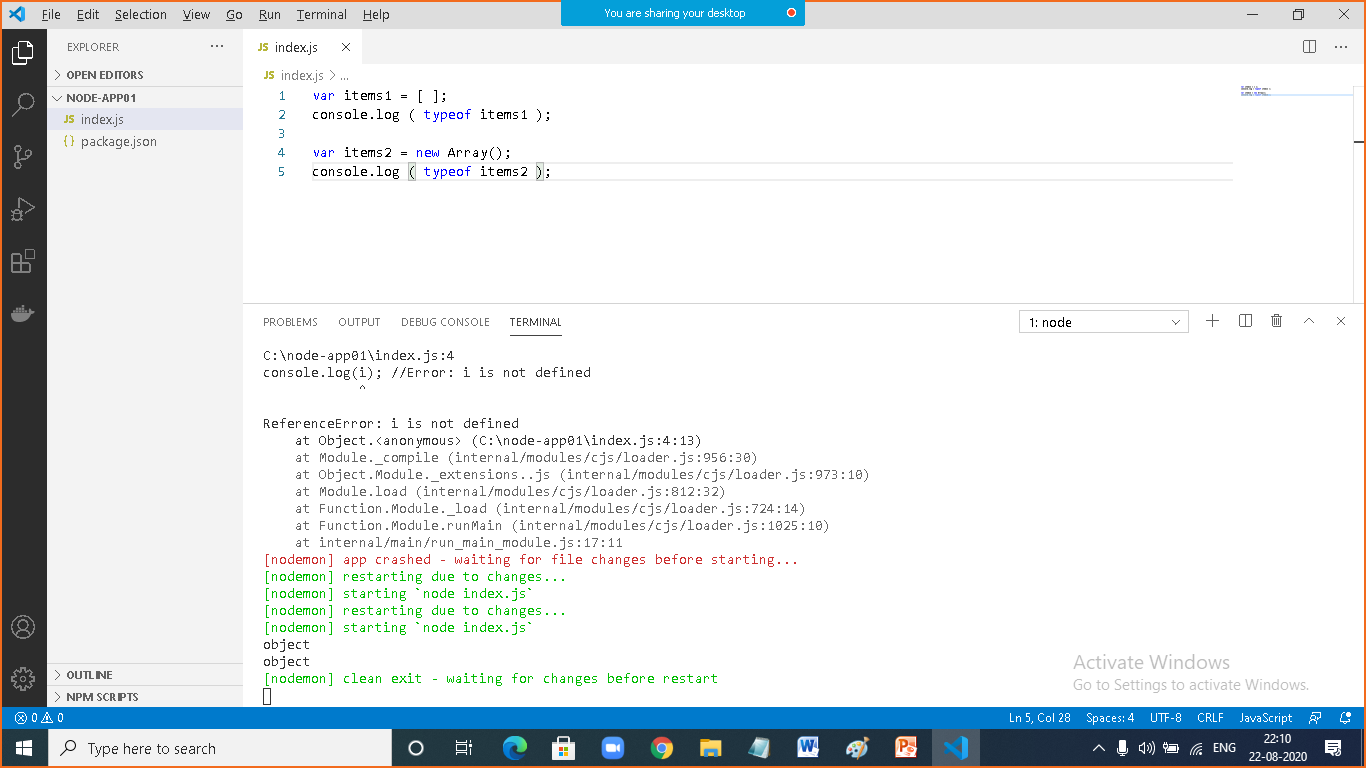
|  |  |
| --- | --- |
|  |  |
|  | var i=1;  while ( i<=10){  console.log(i);  ++i;  } |
|  | var i=1;  do  {  console.log(i);  ++i;  }while ( i<=10); |
|  | for ( var i=1 ; i<=10 ; i++ )  console.log(i); |

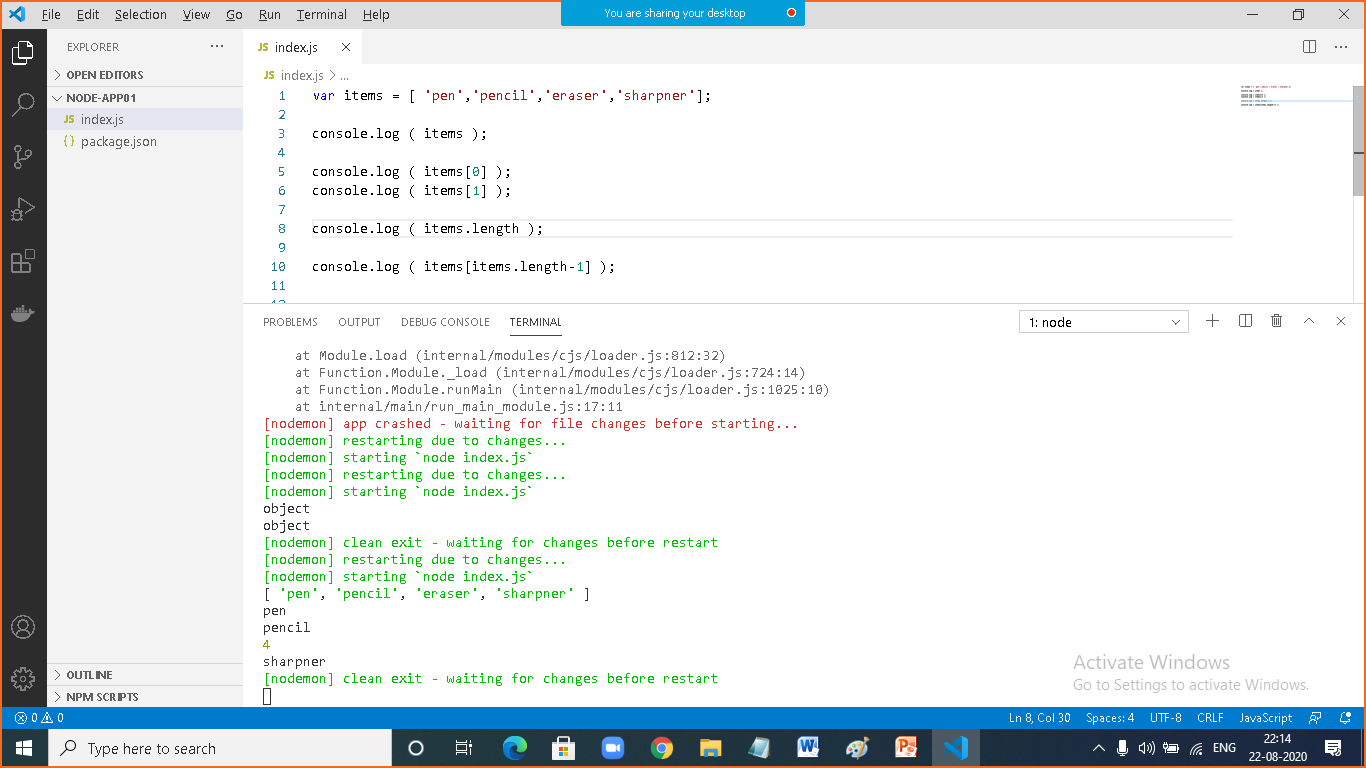


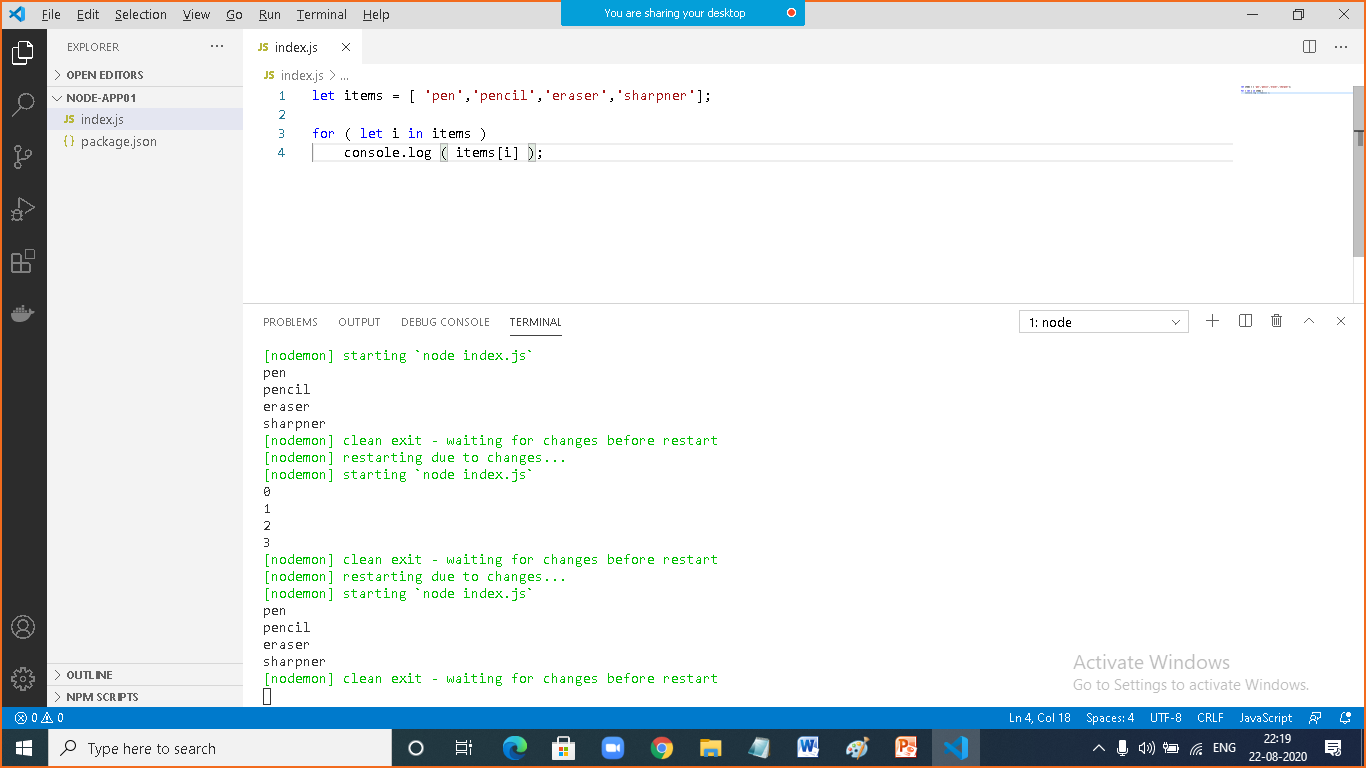


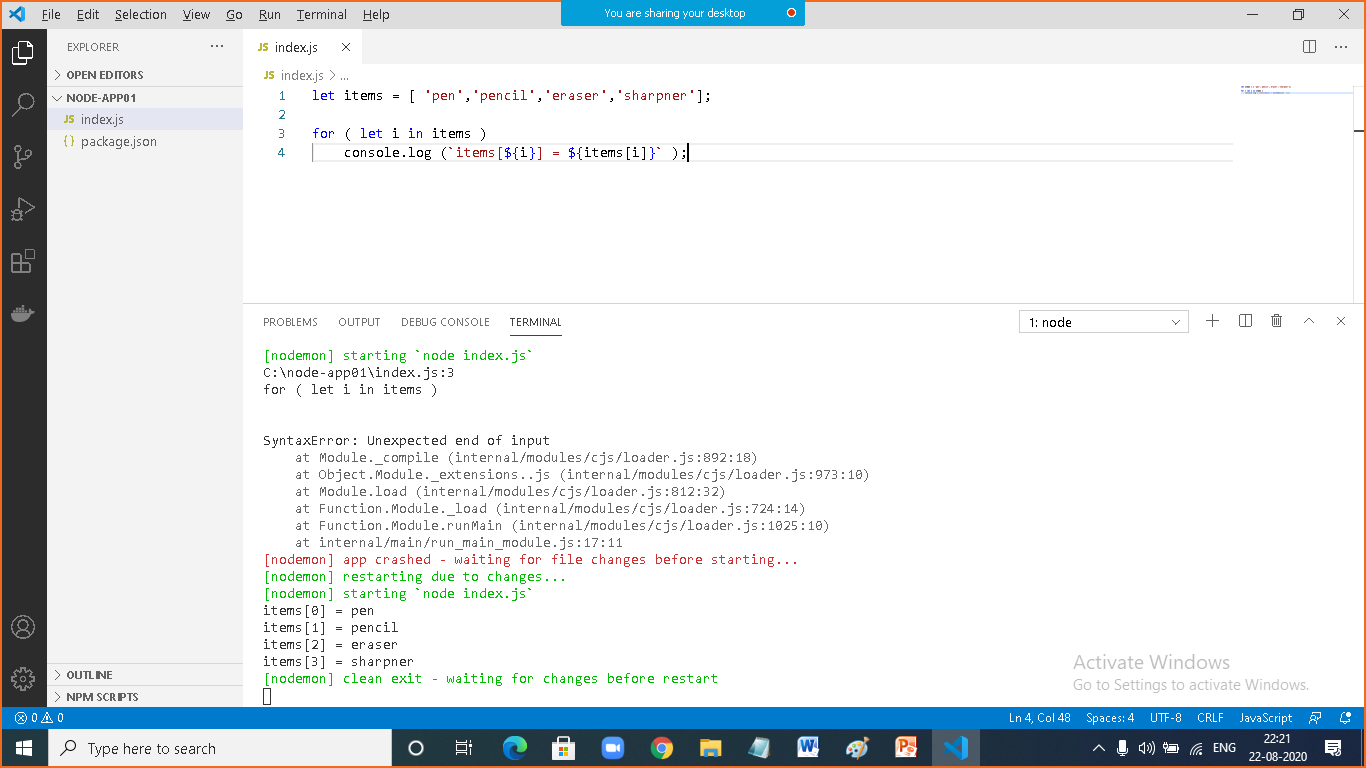
|  |  |
| --- | --- |
|  | array |
|  | It is used to store multiple items within single identification  All items will be stored in homogenous order  Array is an index based collection |

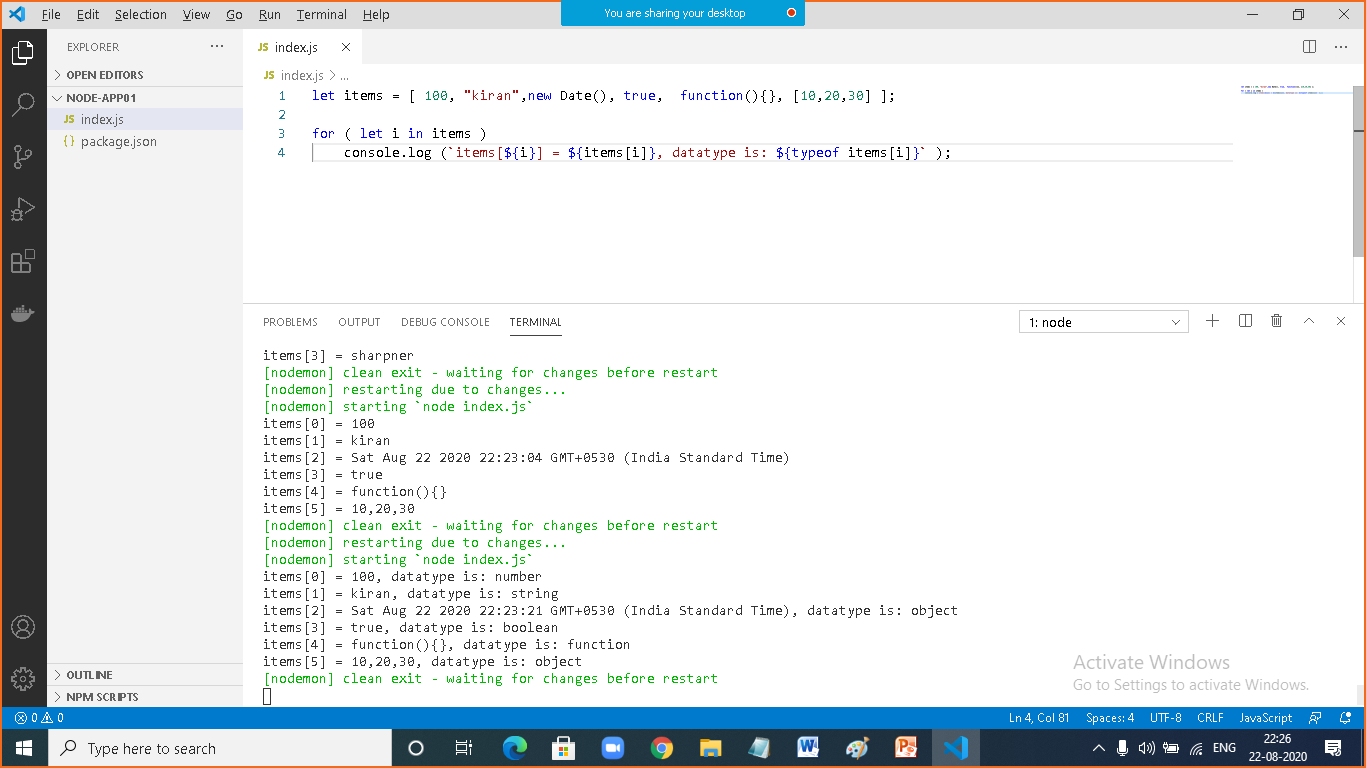
|  |  |
| --- | --- |
|  | Javascript array |
|  | Used to unlimited items  Each and every time can be different type  All UI Technologies uses Array in the form of Collections |



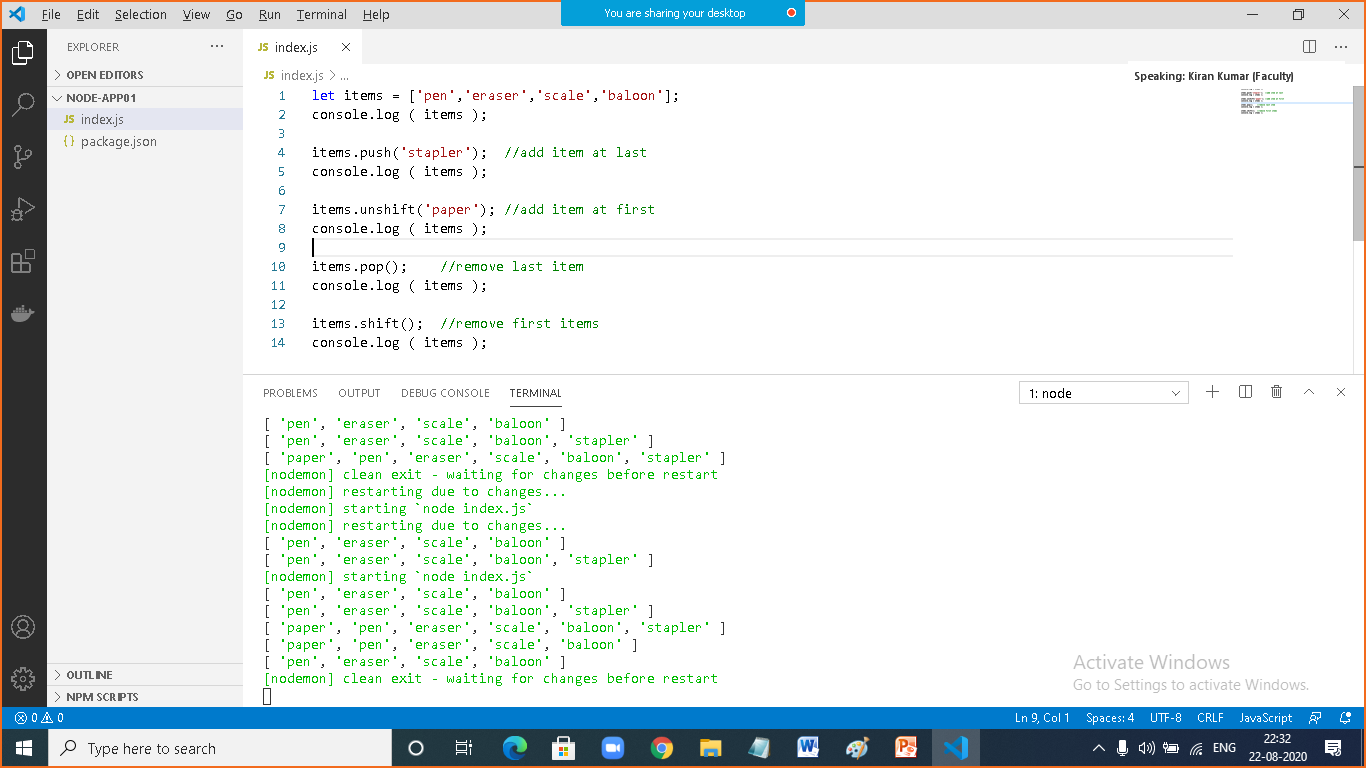






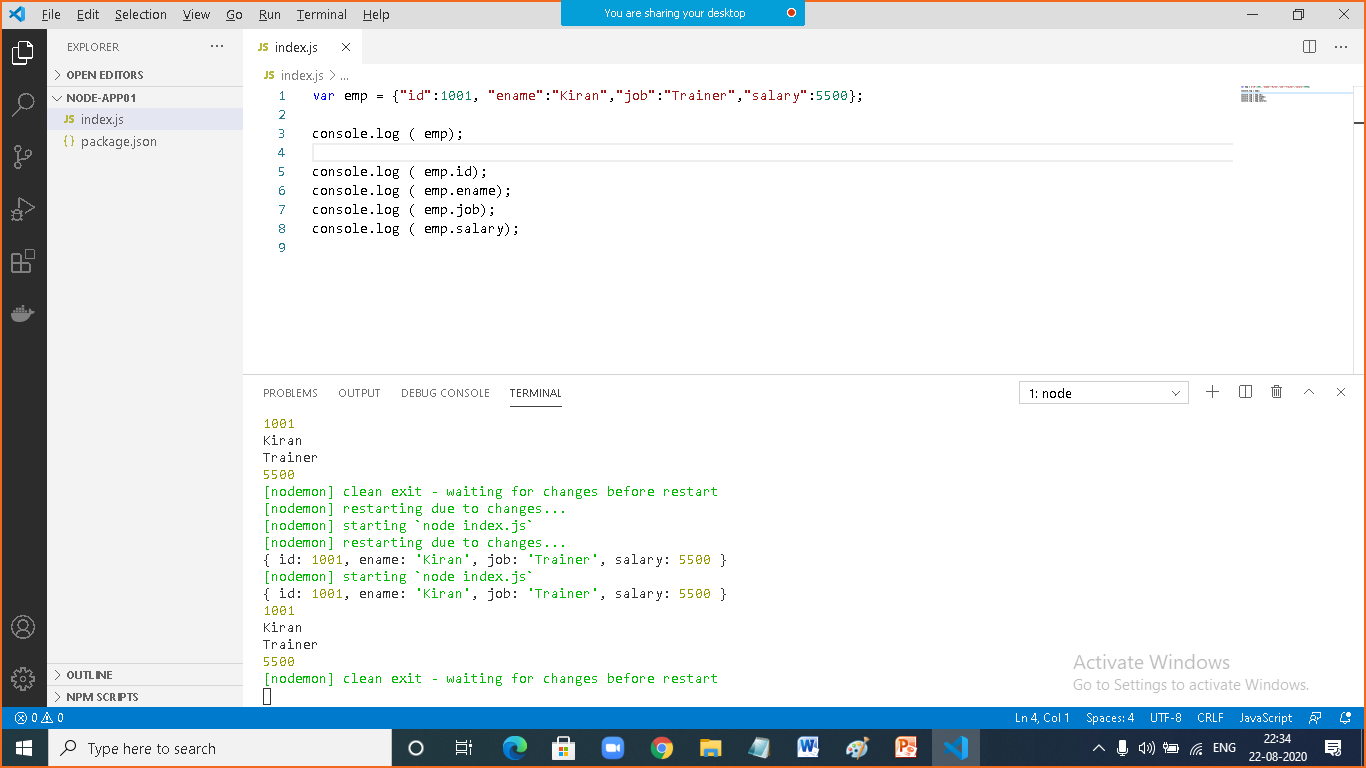


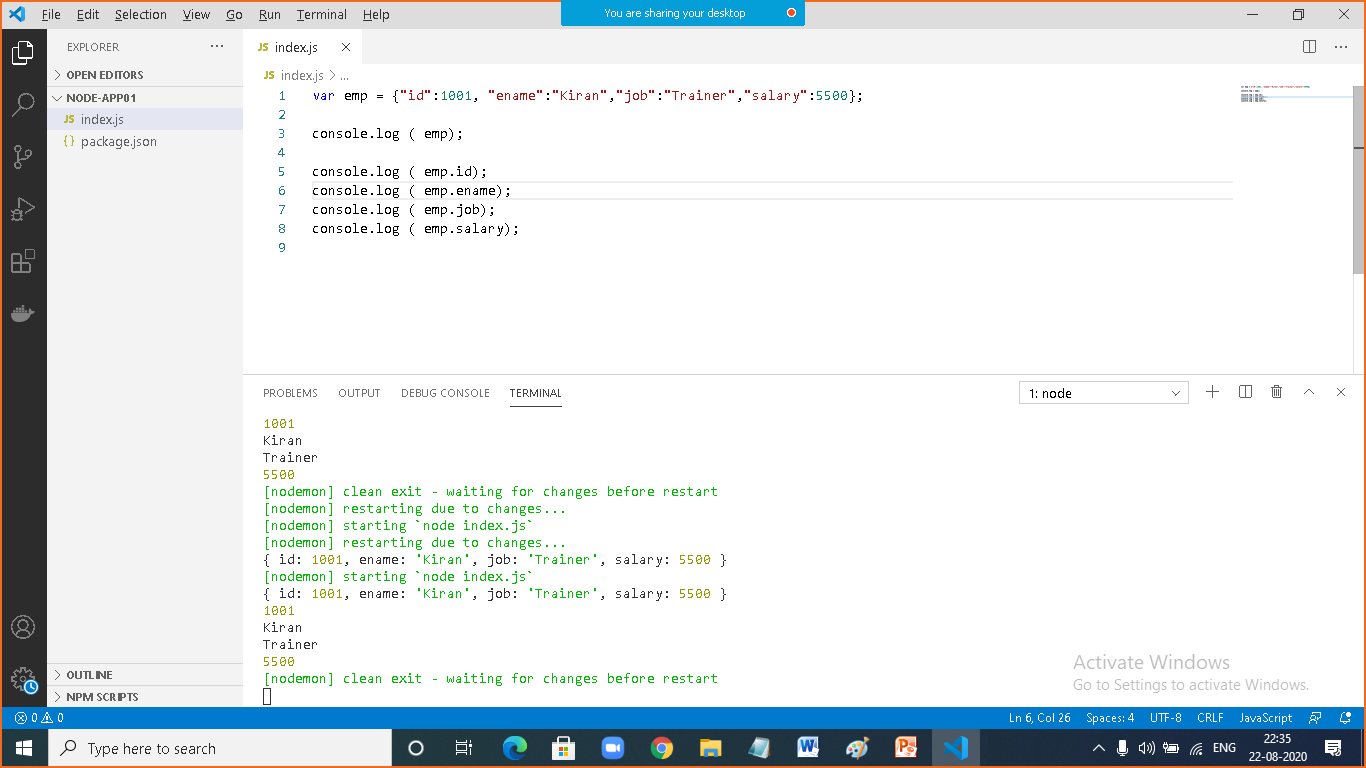
|  |  |
| --- | --- |
|  |  |
|  | let items = [ 100, "kiran",new Date(), true, function(){}, [10,20,30] ];  for ( let i in items )  console.log (`items[${i}] = ${items[i]}, datatype is: ${typeof items[i]}` ); |



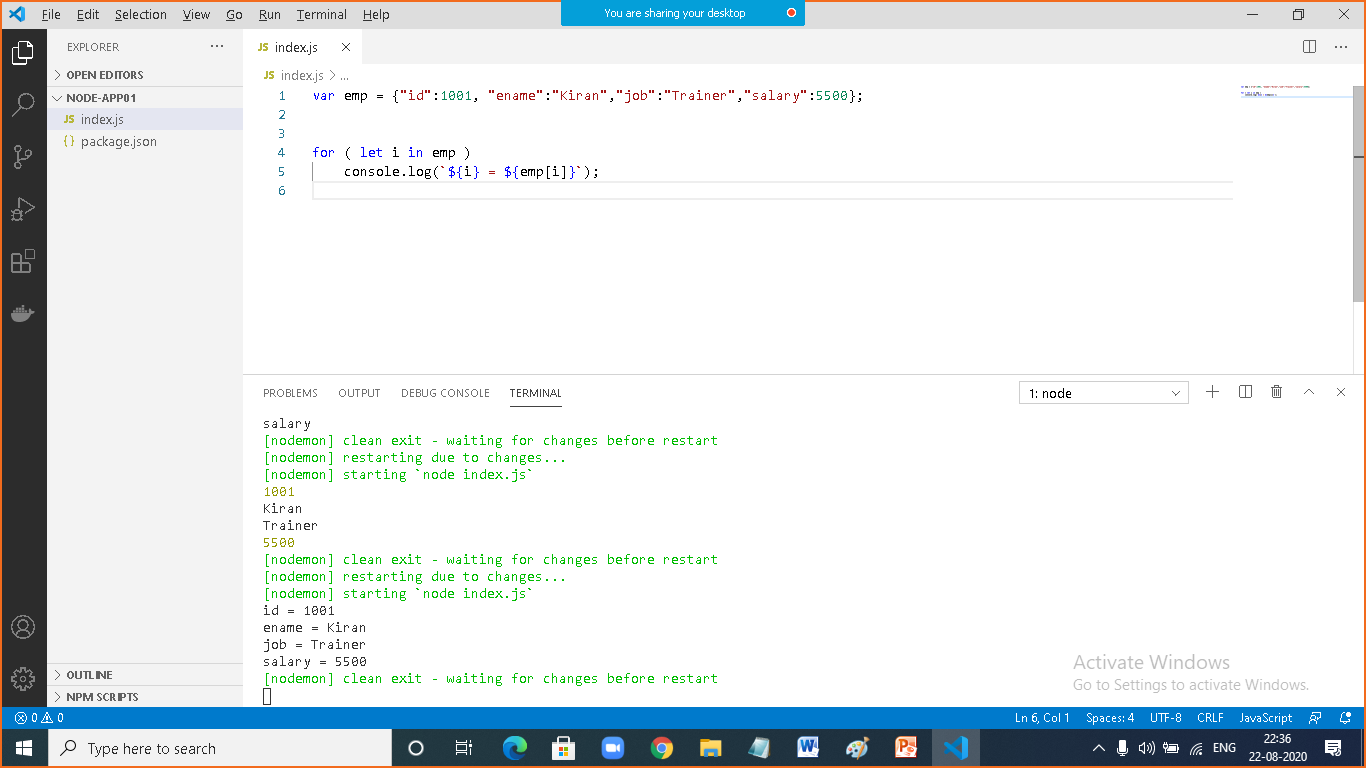
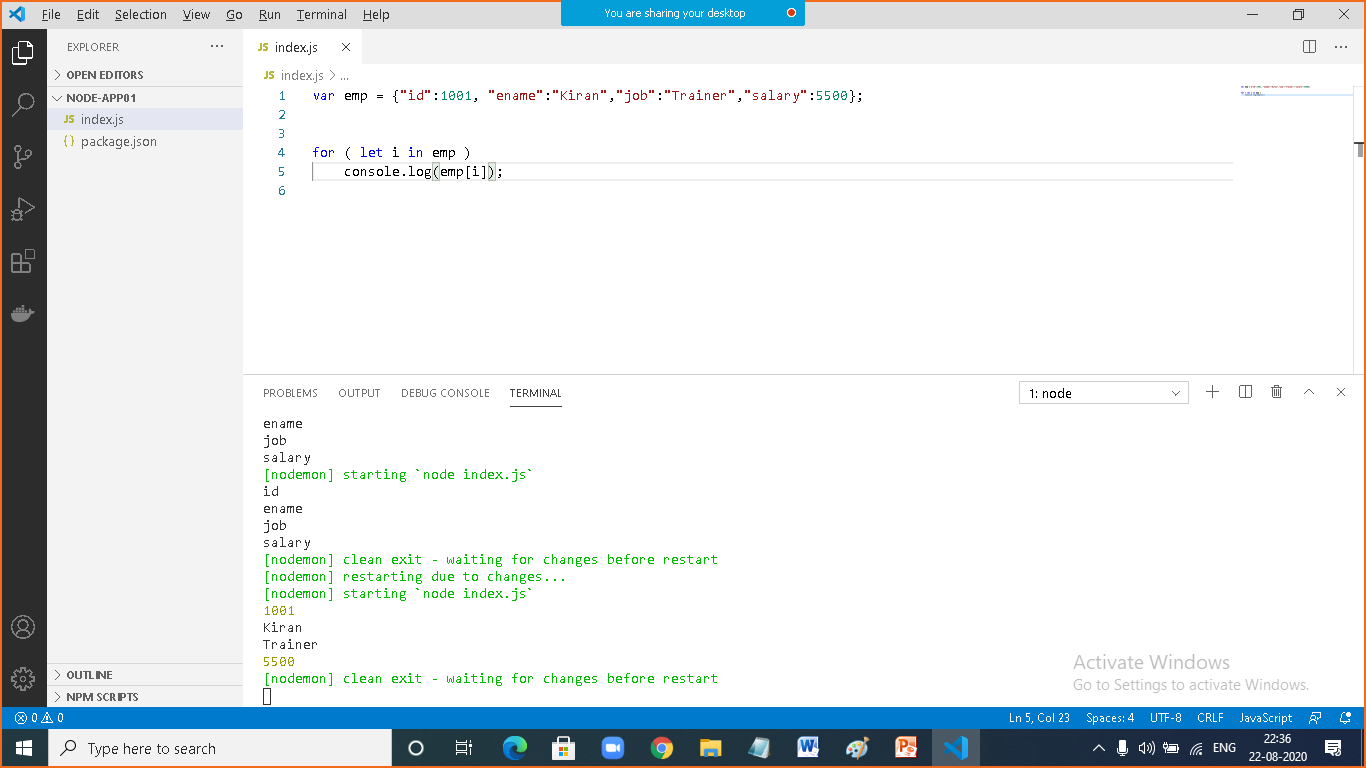
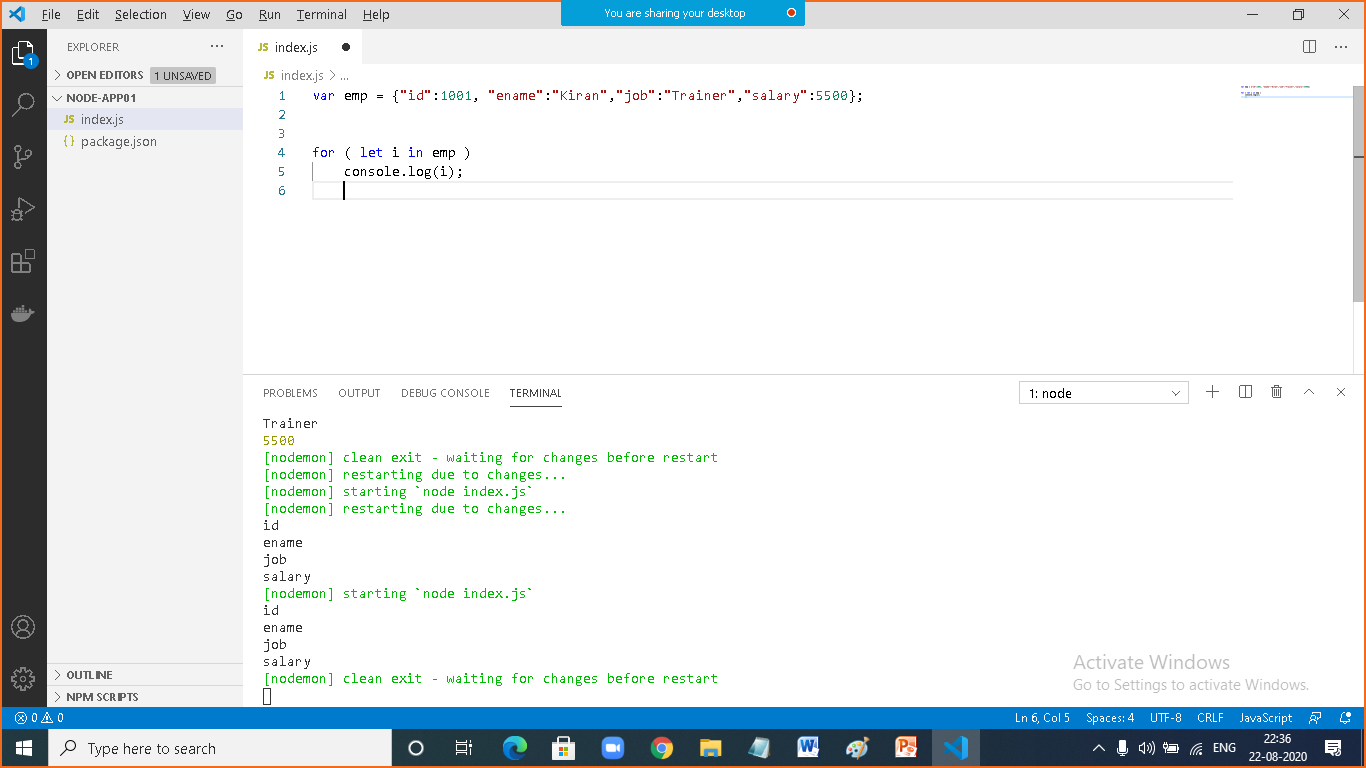
|  |  |
| --- | --- |
|  |  |
|  | let items = ['pen','eraser','scale','baloon'];  console.log ( items );  items.push('stapler'); //add item at last  console.log ( items );  items.unshift('paper'); //add item at first  console.log ( items );  items.pop(); //remove last item  console.log ( items );  items.shift(); //remove first items  console.log ( items ); |

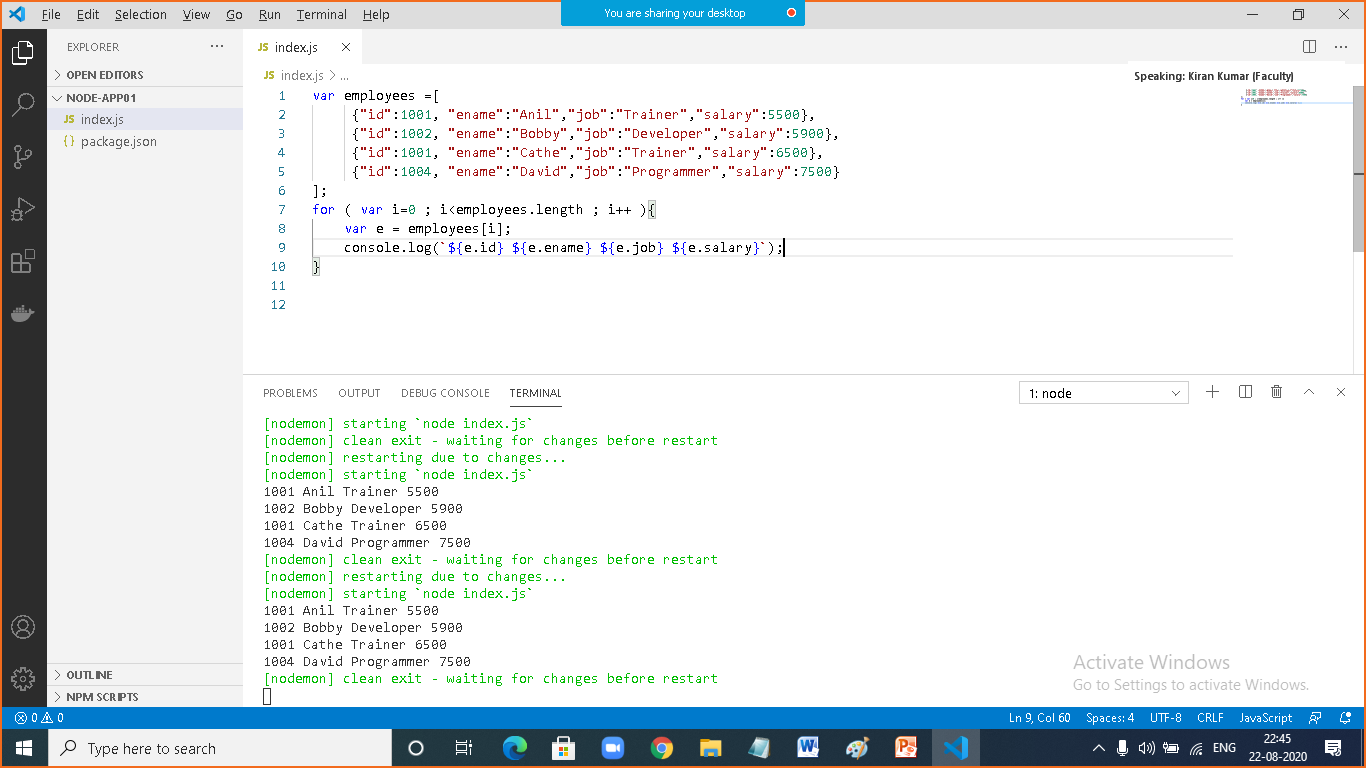
|  |  |
| --- | --- |
|  | json |
|  | JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. |





|  |  |
| --- | --- |
|  |  |
|  | var emp = {"id":1001, "ename":"Kiran","job":"Trainer","salary":5500};  console.log ( emp);  console.log ( emp.id);  console.log ( emp.ename);  console.log ( emp.job);  console.log ( emp.salary); |





|  |  |
| --- | --- |
|  |  |
|  | var employees =[  {"id":1001, "ename":"Anil","job":"Trainer","salary":5500},  {"id":1002, "ename":"Bobby","job":"Developer","salary":5900},  {"id":1001, "ename":"Cathe","job":"Trainer","salary":6500},  {"id":1004, "ename":"David","job":"Programmer","salary":7500}  ];  for ( var i=0 ; i<employees.length ; i++ ){  var e = employees[i];  console.log(`${e.id} ${e.ename} ${e.job} ${e.salary}`);  } |

|  |  |
| --- | --- |
|  |  |
|  | var employees =[  {"id":1001, "ename":"Anil","job":"Trainer","salary":5500},  {"id":1002, "ename":"Bobby","job":"Developer","salary":5900},  {"id":1003, "ename":"Cathe","job":"Trainer","salary":6500},  {"id":1004, "ename":"David","job":"Programmer","salary":7500}  ];  employees.unshift( {"id":1000, "ename":"Ashok","job":"Developer","salary":9900});  employees.push( {"id":1005, "ename":"Fernandez","job":"Developer","salary":6800});  for ( let e of employees)  console.log(`${e.id} ${e.ename} ${e.job} ${e.salary}`); |

|  |  |
| --- | --- |
|  | **Understand module** |
|  | A Module is a container file, contains code which can be exported  How to create a module?  Create a file with .js extension  How to export source?  Module.exports  How to import?  require() |

