

**Ex.No. 1****CALCULATOR NODE****Execution Steps**

**1. Open visual studio code click file -> Open Folder -> select or create folder**

**2. In explorer panel click create new file give file name as division.js and type the following code and click save**

```
function division(a, b) {  
    console.log("The division of " + a + " & " + b + " is : " + a / b);  
}  
module.exports = division;
```

**3. In explorer panel click create new file give file name as multiplication.js and type the following code and click save**

```
function multiplication(a, b) {  
    console.log("The multiplication of " + a + " & " + b + " is : " + a * b);  
}  
module.exports = multiplication;
```

**4. In explorer panel click create new file give file name as subtraction.js and type the following code and click save**

```
function subtraction(a, b) {  
    console.log("The subtraction of " + a + " & " + b + " is : " + (a - b));  
}  
module.exports = subtraction;
```

**5. In explorer panel click create new file give file name as sum.js and type the following code and click save**

```
function sum(a, b) {  
    console.log("The sum of " + a + " & " + b + " is : " + (a + b));  
}  
module.exports = sum;
```

**6. In explorer panel click create new file give file name as main.js and type the following code and click save**

```
var moment = require("moment");  
var sum = require("./sum");  
var subs = require("./subtraction");
```

```
var mult = require("./multiplication");  
var div = require("./division");  
var firstOperand = +process.argv[2];  
var secondOperand = +process.argv[3];  
console.log("Today is: " + moment().format("dddd, MMMM Do YYYY, h:mm:ss a"));  
sum(firstOperand, secondOperand);  
subs(firstOperand, secondOperand);  
mult(firstOperand, secondOperand);  
div(firstOperand, secondOperand);
```

### **7. Click Terminal->new Terminal and type node division.js**

node multiplication.js

node sum.js

node subtraction.js

node main.js firstnumber secondnumber

**8. If you found any error like module not found then type npm i module name to install it then type node main.js firstnumber secondnumber**

## **Ex.No. 2                      CHAIN MIDDLEWARE TO CREATE A TIME SERVER**

### **Execution Steps**

**1. Open visual studio code click file -> Open Folder -> select or create folder**

**2. In explorer panel click create new file give file name as main.js and type the following code and click save**

```
var express = require("express");  
var app = express();  
// Chaining middleware. A Time server  
app.get(  
  "/now",  
  (req, res, next) => {  
    req.time = new Date().toString();  
    next();  
  },  
  (req, res) => {  
    res.json({ time: req.time });  
  })  
);
```

```
}  
);  
  
app.listen(process.env.PORT || 3000);
```

**3. Click Terminal->new Terminal and type node main.js**

**4. To run the program go to the browser and type**

**localhost:3000/now**

**to exit server running press Ctrl+C**

### **Ex.No.3 GET ROUTE PARAMETER INPUT FROM THE CLIENT**

#### **Execution Steps**

**1. Open visual studio code click file -> Open Folder -> select or create folder**

**2. In explorer panel click create new file give file name as main.js and type the following code and click save**

```
var express = require("express");  
var app = express();  
// Get input from client - Route parameters  
app.get("/:word/echo", (req, res) => {  
  res.json({ echo: req.params.word });  
});  
app.listen(process.env.PORT || 3000);
```

**3. Click Terminal->new Terminal and type node main.js**

**4. To run the program go to the browser and type**

**localhost:3000/:angular/echo**

**to exit server running press Ctrl+C**

### **Ex.No.4 GET QUERY PARAMETER INPUT FROM THE CLIENT**

**Execution Steps**

**1. Open visual studio code click file -> Open Folder -> select or create folder**

**2. In explorer panel click create new file give file name as main.js and type the following code and click save**

```
var express = require("express");
```

```
var app = express();
```

```
// Get input from client - Query parameters
```

```
// /name?first=<firstname>&last=<lastname>
```

```
app.route("/name").get((req, res) => {
```

```
  res.json({ name: `${req.query.first} ${req.query.last}` });
```

```
});
```

```
app.listen(process.env.PORT || 3000);
```

**3. Click Terminal->new Terminal and type node main.js**

**4.To run the program go to the browser and type**

**localhost:3000/name?first=Welcome&last=SRM**

**to exit server running press Ctrl+C**

**Ex.No.5****SHOPPING LIST API****Execution Steps**

**1. Open visual studio code click file -> Open Folder -> select or create folder**

**2. In explorer panel click create new file give file name as item.js and type the following code and click save**

```
class Item {
```

```
  constructor(name, price) {
```

```
    this.name = name;
```

```
    this.price = price;
```

```
    this.id = Item.id;
```

```
    Item.list.push(this);
```

```
    Item.id++;
```

```
}  
  
static update(id, data) {  
  let foundItem = Item.list.find((v) => v.id === id);  
  foundItem.name = data.name;  
  foundItem.price = data.price;  
  return foundItem;  
}  
  
static find(id) {  
  return Item.list.find((v) => v.id === id);  
}  
  
static remove(id) {  
  let foundIdx = Item.list.findIndex((v) => v.id === id);  
  Item.list.splice(foundIdx, 1);  
}  
}  
  
Item.id = 1;  
Item.list = [];  
module.exports = Item;
```

**3. In explorer panel click create new file give file name as main.js and type the following code and click save**

```
const express = require("express");  
const app = express();  
const morgan = require("morgan");  
const bodyParser = require("body-parser");  
  
const Item = require("./item");  
  
app.use(morgan("tiny"));  
app.use(bodyParser.urlencoded({ extended: false }));  
app.use(bodyParser.json());  
  
app.get("/items", (req, res) => {  
  return res.json(Item.list);  
});
```

```
app.post("/items", (req, res) => {  
  let newItem = new Item(req.body.name, req.body.price);  
  return res.json(newItem);  
});
```

```
app.get("/items/:id", (req, res) => {  
  let foundItem = Item.find(+req.params.id);  
  return res.json(foundItem);  
});
```

```
app.patch("/items/:id", (req, res) => {  
  let foundItem = Item.update(+req.params.id, req.body);  
  return res.json(foundItem);  
});
```

```
app.delete("/items/:id", (req, res) => {  
  Item.remove(+req.params.id);  
  return res.json("Removed");  
});
```

```
// catch 404 and forward to error handler  
app.use((req, res, next) => {  
  var err = new Error("Not Found");  
  err.status = 404;  
  next(err);  
});
```

```
// error handlers
```

```
// development error handler
```

```
// will print stacktrace
```

```
if (app.get("env") === "development") {  
  app.use((err, req, res, next) => {  
    res.status(err.status || 500);
```

```
res.send({
  message: err.message,
  error: err,
});
});
}
// production error handler
// no stacktraces leaked to user
app.use((err, req, res, next) => {
  res.status(err.status || 500);
  res.send({
    message: err.message,
    error: {},
  });
});

app.listen(process.env.PORT || 3000, () => {
  console.log("Server is listening on port 3000");
});
```

**3. Click Terminal->new Terminal and type node item.js and node main.js**

**4. To run the program install Postman API select post method, type localhost:3000/items type key and value for insertion then click send**

**Select Get method, type localhost:3000/items and click send to view the inserted details**

**Select patch method, type localhost:3000/items/itemnumber and click send to search the details**

**Select delete methods and type localhost:3000/items/itemnumber and click send to delete the details**

**Install Mongodb by clicking leaf symbol in explorer panel and connect by typing mongodb://localhost:27017**

**Ex.No. 6                      Create a Model – Using model.find()**

**Execution Steps**

**1. Click create new file type the file name as config.js and type the below code**

```
// Creating schema

var mongoose = require("mongoose");
mongoose.connect("mongodb://localhost:27017/Sample", {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

var Schema = mongoose.Schema;
var PersonSchema = new Schema({
  name: { type: String, required: true },
  age: Number,
  favoriteFoods: [{ type: String, unique: true }],
});
var Person = mongoose.model("Person", PersonSchema);
var arrayOfPeople = [
  { name: "Frankie", age: 74, favoriteFoods: ["Taco"] },
  { name: "Sol", age: 76, favoriteFoods: ["Roast chicken", "Pizza"] },
  { name: "Robert", age: 78, favoriteFoods: ["Burger"] },
];
// Defining and Using model.find()
var findPeopleByName = function (personName) {
  Person.find({ name: personName }, function (err, personFound) {
    if (err) return console.log(err);
    done(personFound);
  });
};

findPeopleByName("Robert");
```

**2. Click Terminal->New Terminal->type npm i mongoose after it is runned type node config.js****Ex.No. 7****Create a Model – using model.findOne()****Execution Steps****1. Click create new file type the file name as config.js and type the below code**



```
// Creating schema
```

```
var mongoose = require("mongoose");
mongoose.connect("mongodb://localhost:27017/Sample", {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
```

```
var Schema = mongoose.Schema;
var PersonSchema = new Schema({
  name: { type: String, required: true },
  age: Number,
  favoriteFoods: [{ type: String, unique: true }],
});
```

```
var Person = mongoose.model("Person", PersonSchema);
```

```
var arrayOfPeople = [
  { name: "Frankie", age: 74, favoriteFoods: ["Taco"] },
  { name: "Sol", age: 76, favoriteFoods: ["Roast chicken", "Pizza"] },
  { name: "Robert", age: 78, favoriteFoods: ["Burger"] },
];
```

```
// Defining and using model.findOne()
```

```
var findOneByFood = function (food) {
  Person.findOne({ favoriteFoods: food }, (err, data) =>
    err ? done(err) : done(data)
  );
};
```

```
findOneByFood("Pizza");
```

**2. Click Terminal->New Terminal->type npm i mongoose after it is runned type node config.js**

**Ex.No. 8****Create a Model – Find , Edit , Save****Execution Steps****1. Click create new file type the file name as config.js and type the below code**

```
// Creating schema
var mongoose = require("mongoose");
mongoose.connect("mongodb://localhost:27017/Sample", {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

var Schema = mongoose.Schema;
var PersonSchema = new Schema({
  name: { type: String, required: true },
  age: Number,
  favoriteFoods: [{ type: String, unique: true }],
});

var Person = mongoose.model("Person", PersonSchema);

var arrayOfPeople = [
  { name: "Frankie", age: 74, favoriteFoods: ["Taco"] },
  { name: "Sol", age: 76, favoriteFoods: ["Roast chicken", "Pizza"] },
  { name: "Robert", age: 78, favoriteFoods: ["Burger"] },
];

// Defining and calling findEditThenSave function
var findEditThenSave = function (personId) {
  var foodToAdd = "hamburger";
  Person.findById({ _id: personId }, function (err, data) {
    if (err) {
      return done(err);
    } else {
      data.favoriteFoods.push(foodToAdd);
      data.save((err, data) => (err ? done(err) : done(data)));
    }
  });
}
```

```
}  
});  
};
```

```
findEditThenSave("6019640b4c99351684dc8561");
```

**2. Click Terminal->New Terminal->type npm i mongoose after it is runned type node config.js**

## **Ex.No. 9                                  Create a Model – Delete**

### **Execution Steps**

**1. Click create new file type the file name as config.js and type the below code**

```
// Creating schema  
var mongoose = require("mongoose");  
mongoose.connect("mongodb://localhost:27017/Sample", {  
  useNewUrlParser: true,  
  useUnifiedTopology: true,  
});
```

```
var Schema = mongoose.Schema;  
var PersonSchema = new Schema({  
  name: { type: String, required: true },  
  age: Number,  
  favoriteFoods: [{ type: String, unique: true }],  
});
```

```
var Person = mongoose.model("Person", PersonSchema);
```

```
var arrayOfPeople = [  
  { name: "Frankie", age: 74, favoriteFoods: ["Taco"] },
```

```
{ name: "Sol", age: 76, favoriteFoods: ["Roast chicken", "Pizza"] },
{ name: "Robert", age: 78, favoriteFoods: ["Burger"] },
];
```

```
varremoveManyPeople = function () {
varnameToRemove = "Robert";
Person.remove({ name: nameToRemove }, function (error, data) {
error ?done(error) : done(data);
});
};
removeManyPeople("Robert");
```

**2. Click Terminal->New Terminal->type npm i mongoose after it is runned type node config.js**

### Ex.No. 10

### Queries

#### Execution Steps

**1. In the explorer panel right click on localhost:27017 connected->click add database ->right click on database->click add collection->right click on collection ->click insert document in that type the details ->click view documents for updations**

**2. In the mongodb shell type the queries and get the results**

**a. Write a MongoDB query to display all the documents in the collection restaurants**

**Query:**

```
db.restaurants.find(); //restaurants is the collection
```

**b. Write a MongoDB query to display next 5 restaurants after skipping first 5 restaurants in the borough Bronx.**

**Query:**

```
db.rest.find({ borough: "Bronx" }).skip(5).limit(5);
```

**c. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a grade point 'A' not belonging to the borough Brooklyn. The document must be displayed according to cuisine in descending order.**

**Query:**

```
db.rest
.find({
  $and: [
```

```
{ cuisine: { $ne: "American " } },  
{ "grades.grade": "A" },  
{ borough: { $ne: "Brooklyn " } } },  
],  
}))  
.sort({ cuisine: -1 });
```

**d. Write a MongoDB query to find the restaurant Id, name, borough, and cuisine for those restaurants which do not belong to the borough Staten Island or Queens or Brooklyn or Bronx.**

**Query:**

```
db.rest.find(  
{ borough: { $nin: ["Staten Island", "Queens", "Bronx", "Brooklyn"] } },  
{ _id: 0, restaurant_id: 1, name: 1, borough: 1, cuisine: 1 }  
);
```

**e. Write a MongoDB query arrange the name of restaurants in ascending order along with all the columns.**

**Query:**

```
db.rest.find({}, { _id: 0, name: 1 }).sort({ name: 1 });
```

**f. Write a MongoDB query to find the restaurant name, borough, longitude, and latitude and cuisine for those restaurants which contain ‘Mad’ as the first three letters of its name.**

**Query:**

```
db.rest.find(  
{ name: { $regex: /^Mad.* / } },  
{ _id: 0, name: 1, borough: 1, "address.coord": 1, cuisine: 1 }  
);
```

**Ex.No.11****Navigation Menu****Execution Steps:**

1. Open Visual studio code click open folder, create or select folder
2. Select terminal new terminal, in the terminal type `npm i -g @angular/cli` to install angular
3. In the terminal type `ng new filename --no-standalone`, select the line highlighted in blue color, press y and wait for some time
4. In the terminal type `cd filename`
5. In the explorer panel click filename->src->app.component.html and the type the below code

```
<div id="main">
<nav class="{ { active} }" (click)="$event.preventDefault()">
<a href="#" class="home" (click)="active='home'">Home</a>
<a href="#" class="projects" (click)="active='projects'">Projects</a>
<a href="#" class="services" (click)="active='services'">Services</a>
<a href="#" class="contact" (click)="active='contact'">Contact</a>
</nav>

<p *ngIf="!active">Please click a menu item</p>
<p *ngIf="active">You chose <b>{ { active} }</b></p>
</div>
```

6. In the explorer panel click filename->src->app.component.css and the type the below code

```
* {
margin: 0;
padding: 0;
}

body {
font: 15px/1.3 "Open Sans", sans-serif;
color: #5e5b64;
text-align: center;
}
```

a,

```
a:visited {  
outline: none;  
color: #389dc1;  
}
```

```
a:hover {  
text-decoration: none;  
}
```

```
section,  
footer,  
header,  
aside,  
nav {  
display: block;  
}
```

```
/*-----  
The menu  
-----*/
```

```
#main {  
text-align: center;  
}
```

```
nav {  
display: inline-block;  
margin: 60px auto 45px;  
background-color: #5597b4;  
box-shadow: 0 1px 1px #ccc;  
border-radius: 2px;  
}
```

```
nav a {  
display: inline-block;
```

```
padding: 18px 30px;
color: #fff !important;
font-weight: bold;
font-size: 16px;
text-decoration: none !important;
line-height: 1;
text-transform: uppercase;
background-color: transparent;

-webkit-transition: background-color 0.25s;
-moz-transition: background-color 0.25s;
transition: background-color 0.25s;
}
```

```
nav a:first-child {
border-radius: 2px 0 0 2px;
}
nav a:last-child {
border-radius: 0 2px 2px 0;
}
```

```
nav.home .home,
nav.projects .projects,
nav.services .services,
nav.contact .contact {
background-color: #e35885;
}
```

```
p {
font-size: 22px;
font-weight: bold;
color: #7d9098;
}
```

```
p b {
```



```
color: #ffffff;
display: inline-block;
padding: 5px 10px;
background-color: #c4d7e0;
border-radius: 2px;
text-transform: uppercase;
font-size: 18px;
}
```

**7. In the explorer panel click filename->src->app.component.ts and the type the below code, change the active:string to active:string=null**

```
import { Component } from '@angular/core';
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  active: string;
}
```

**8. In the explorer panel click filename->src->app.module.ts and the type the below code**

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
```

```

    ],
    providers: [],
    bootstrap: [AppComponent]
  })
  export class AppModule { }

```

**9. Save all the files and type ng serve in the terminal wait for some time then select the blue color highlighted text then press Ctrl+Click then your code is running you see the output**

**10. Press Ctrl+C to come out from running of server**

## Ex.No.12

## Inline Editor

### Execution Steps:

- 1. Open Visual studio code click open folder, create or select folder**
- 2. Select terminal new terminal, in the terminal type npm i -g @angular/cli to install angular**
- 3. In the terminal type ng new filename --no-standalone, select the line highlighted in blue color, press y and wait for some time**
- 4. In the terminal type cd filename**
- 5. In the explorer panel click filename->src->app.component.html and the type the below code**

```

<div id="main" (click)="hideTooltip()">
<div class="tooltip" (click)="$event.stopPropagation()" *ngIf="showtooltip">
<label for="text"></label>
<input name="text" type="text" [(ngModel)]="value" />
</div>

<p (click)="toggleTooltip($event)">{{ value }}</p>
</div>

```

- 6. In the explorer panel click filename->src->app.component.css and the type the below code**

```

* {

```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
body {
```

```
font: 15px/1.3 "Open Sans", sans-serif;
```

```
color: #5e5b64;
```

```
text-align: center;
```

```
}
```

```
a,
```

```
a:visited {
```

```
outline: none;
```

```
color: #389dc1;
```

```
}
```

```
a:hover {
```

```
text-decoration: none;
```

```
}
```

```
section,
```

```
footer,
```

```
header,
```

```
aside,
```

```
nav {
```

```
display: block;
```

```
}
```

```
/*-----
```

```
    The edit tooltip
```

```
-----*/
```

```
.tooltip {
```

```
background-color: #5c9bb7;
```

```
background-image: -webkit-linear-gradient(to bottom, #5c9bb7, #5392ad);
```

background-image: -moz-linear-gradient(to bottom, #5c9bb7, #5392ad);

background-image: linear-gradient(to bottom, #5c9bb7, #5392ad);

box-shadow: 0 1px 1px #ccc;

border-radius: 3px;

width: 290px;

padding: 10px;

position: absolute;

left: 50%;

margin-left: -150px;

top: 80px;

}

.tooltip:after {

/\* The tip of the tooltip \*/

content: "";

position: absolute;

border: 6px solid #5190ac;

border-color: #5190ac transparent transparent;

width: 0;

height: 0;

bottom: -12px;

left: 50%;

margin-left: -6px;

}

.tooltip input {

border: none;

width: 100%;

line-height: 34px;

border-radius: 3px;

box-shadow: 0 2px 6px #bbb inset;

text-align: center;

font-size: 16px;

```
font-family: inherit;  
color: #8d9395;  
font-weight: bold;  
outline: none;  
}
```

```
p {  
font-size: 22px;  
font-weight: bold;  
color: #6d8088;  
height: 30px;  
cursor: default;  
text-align: center;  
}
```

```
p b {  
color: #ffffff;  
display: inline-block;  
padding: 5px 10px;  
background-color: #c4d7e0;  
border-radius: 2px;  
text-transform: uppercase;  
font-size: 18px;  
}
```

```
p:before {  
content: "📖";  
display: inline-block;  
margin-right: 5px;  
font-weight: normal;  
vertical-align: text-bottom;  
}
```

```
#main {  
height: 300px;
```

```
position: relative;
padding-top: 150px;
}
```

**7. In the explorer panel click filename->src->app.component.ts and then type the below code, select tsconfig.json file in explorer panel then type**

**“noImplicitAny”:false,**

**“noImplicitThis”:false,**

```
import { Component } from '@angular/core';
```

```
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
```

```
export class AppComponent {
  showtooltip = false;
  value = 'Edit me.';
```

```
hideTooltip = function () {
  this.showtooltip = false;
}
```

```
toggleTooltip = function (e) {
  e.stopPropagation();
  this.showtooltip = !this.showtooltip;
}
}
```

**8. In the explorer panel click filename->src->app.module.ts and then type the below code**

```
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
```

```
import { BrowserModule } from '@angular/platform-browser';
```

```
import { AppComponent } from './app.component';
```

```
@NgModule({  
  declarations: [  
    AppComponent  
  ],  
  imports: [  
    BrowserModule,  
    FormsModule  
  ],  
  
  providers: [],  
  bootstrap: [AppComponent]  
})
```

```
export class AppModule { }
```

**9. Save all the files and type ng serve in the terminal wait for some time then select the blue color highlighted text then press Ctrl+Click then your code is running you see the output**

**10. Press Ctrl+C to come out from running of server**

### **Ex.No.13**

### **Order Form**

#### **Execution Steps:**

- 1. Open Visual studio code click open folder, create or select folder**
- 2. Select terminal new terminal, in the terminal type npm i -g @angular/cli to install angular**
- 3. In the terminal type ng new filename --no-standalone, select the line highlighted in blue color, press y and wait for some time**
- 4. In the terminal type cd filename**

**5. In the explorer panel click filename->src->app.component.html and the type the below code**

```
<form>
<h1>Services</h1>
<ul>
<li *ngFor="let service of services" (click)="toggleActive(service)"
[class]="{ active:service.active}">
    {{ service.name }} <span>{{ service.price | currency }}</span>
</li>
</ul>
<div class="total">
    Total: <span>{{ total() | currency }}</span>
</div>
</form>
```

**6. In the explorer panel click filename->src->app.component.css and the type the below code**

```
@import url(https://fonts.googleapis.com/css?family=Cookie);
```

```
* {
margin: 0;
padding: 0;
}
body {
font: 15px/1.3 "Open Sans", sans-serif;
color: #5e5b64;
text-align: center;
}
a,
a:visited {
outline: none;
color: #389dc1;
}
```



```
a:hover {
text-decoration: none;
}
section,
footer,
header,
aside,
nav {
display: block;
}
/*-----
    The order form
-----*/
form {
background-color: #61a1bc;
border-radius: 2px;
box-shadow: 0 1px 1px #ccc;
width: 400px;
padding: 35px 60px;
margin: 50px auto;
}

form h1 {
color: #fff;
font-size: 64px;
font-family: "Cookie", cursive;
font-weight: normal;
line-height: 1;
text-shadow: 0 3px 0 rgba(0, 0, 0, 0.1);
}

formul {
list-style: none;
color: #fff;
font-size: 20px;
```

```
font-weight: bold;
```

```
text-align: left;
```

```
margin: 20px 0 15px;
```

```
}
```

```
formul li {
```

```
padding: 20px 30px;
```

```
background-color: #e35885;
```

```
margin-bottom: 8px;
```

```
box-shadow: 0 1px 1px rgba(0, 0, 0, 0.1);
```

```
cursor: pointer;
```

```
}
```

```
formul li span {
```

```
float: right;
```

```
}
```

```
formulli.active {
```

```
background-color: #8ec16d;
```

```
}
```

```
div.total {
```

```
border-top: 1px solid rgba(255, 255, 255, 0.5);
```

```
padding: 15px 30px;
```

```
font-size: 20px;
```

```
font-weight: bold;
```

```
text-align: left;
```

```
color: #fff;
```

```
}
```

```
div.total span {
```

```
float: right;
```

```
}
```

**7. In the explorer panel click filename->src->app.component.ts and then type the below code, select tsconfig.json file in explorer panel then type**

“noImplicitAny”:false,

“noImplicitThis”:false,

```
import { Component } from '@angular/core';
```

```
@Component({
```

```
  selector: 'app-root',
```

```
  templateUrl: './app.component.html',
```

```
  styleUrls: ['./app.component.scss']
```

```
})
```

```
export class AppComponent {
```

```
  services = [
```

```
    {
```

```
      name: 'Web Development',
```

```
      price: 300,
```

```
      active: true
```

```
    },
```

```
    {
```

```
      name: 'Design',
```

```
      price: 400,
```

```
      active: false
```

```
    },
```

```
    {
```

```
      name: 'Integration',
```

```
      price: 250,
```

```
      active: false
```

```
    },
```

```
    {
```

```
      name: 'Training',
```

```
      price: 220,
```

```
      active: false
```

```
    }  
  ];
```

```
  toggleActive = function (s) {
```

```
    s.active= !s.active;
```

```
};

total = function () {
  let total = 0;

  this.services.forEach(element => {
    if (element.active) {
      total += element.price;
    }
  });
  return total;
}}
```

**8. In the explorer panel click filename->src->app.module.ts and the type the below code**

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

**9. Save all the files and type ng serve in the terminal wait for some time then select the blue color highlighted text then press Ctrl+Click then your code is running you see the output**

**10. Press Ctrl+C to come out from running of server**

**Execution Steps:**

- 1. Open Visual studio code click open folder, create or select folder**
- 2. Select terminal new terminal, in the terminal type `npm i -g @angular/cli` to install angular**
- 3. In the terminal type `ng new filename --no-standalone`, select the line highlighted in blue color, press y and wait for some time**
- 4. In the terminal type `cd filename` then type `ng g pipe search`**
- 5. In the explorer panel click filename->src->app.component.html and the type the below code**

```
<div>
<div class="bar">
<input type="text" [(ngModel)]="searchString" placeholder="Enter your search terms" />
</div>
<ul>
<li *ngFor="let i of items | searchFor:searchString">
<a [href]="i.url">
<img [src]="i.image" />
</a>
<p>{{ i.title }}</p>
</li>
</ul>
</div>
```

- 6. In the explorer panel click filename->src->app.component.css and the type the below code**

```
* {
margin: 0;
padding: 0;
}

body {
font: 15px/1.3 "Open Sans", sans-serif;
color: #5e5b64;
```

```
text-align: center;
```

```
}
```

```
a,
```

```
a:visited {
```

```
outline: none;
```

```
color: #389dc1;
```

```
}
```

```
a:hover {
```

```
text-decoration: none;
```

```
}
```

```
section,
```

```
footer,
```

```
header,
```

```
aside,
```

```
nav {
```

```
display: block;
```

```
}
```

```
/*-----
```

```
    The search input
```

```
-----*/
```

```
.bar {
```

```
background-color: #5c9bb7;
```

```
background-image: -webkit-linear-gradient(to bottom, #5c9bb7, #5392ad);
```

```
background-image: -moz-linear-gradient(to bottom, #5c9bb7, #5392ad);
```

```
background-image: linear-gradient(to bottom, #5c9bb7, #5392ad);
```

```
box-shadow: 0 1px 1px #ccc;
```

```
border-radius: 2px;
```

```
width: 400px;
```

```
padding: 14px;
```

```
margin: 45px auto 20px;
```

```
position: relative;
```

$$\}$$

```
.bar input {
```

```
background: #fff no-repeat 13px 13px;
```

background-image:

url( w5kb3dzLiB4bXBNTTpJbnN0YXV5jZUIEPSj4bXAuaWlkOkU5NEY0RTIFMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBliB4bXBNTTpEb2N1bWVudElEPSj4bXAuZGlkOkU5NEY0RTIGMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBIj4gPHhtcE1NOkRlcm12ZWRGcm9tIHNOUmVmOmIuc3RhbmNlSUQ9InhtcC5paWQ6RTk0RjRFOUMxMDg3MTFFMzlfMTNcQUFDMzJGOTJDNUeIiHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RTk0RjRFOUQxMDg3MTFFMzlfMTNcQUFDMzJGOTJDNUeIiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmlmOmIjERj4gPC94OnhtcG1ldGE+IDw/eHBhY2tldCBibmQ9InliPz4DjA/RAAABK0IEQVR42pTSQUdEURjG8dOY0TqmPkGmRcqYD9CmzZAWJRHVRiA0iFYtM6uofYaiEW2SRJtEi9YxIk1p07ZkWsww0v/wNByve7vm5ee8M+85zz1jbt9Os+WiGkYdYxjCOx5wgFeXUHmtBSzpcCGa+5BJTCjEP+0nKWAT8xqe4ArPGEEVC1hHEbs2oBwdXkM7mj/JLZrad437sCGHOfUtcziutuYu2v8XUFF/4f6vMK/YgAH1HxkBYV60AR31gxkBYd6xAeF3VzMCwvzOBpypX8V4yuFRzX2d2gD/15yjH4fYQEnzkj4fae5rJulF2sMXVrAsaTWtRFu4Osb+1jEDT71/ZveyhouTch2fINQL9hKefKjuYFfuznXWzXMTabyrvfyIV3M4vhXgAEAUMs7K0J9UJAAAAAASUVORK5CYII=);

border: none;

width: 100%:

```
line-height: 19px;
```

padding: 11px 0;

```
border-radius: 2px;
box-shadow: 0 2px 8px #c4c4c4 inset;
text-align: left;
font-size: 14px;
font-family: inherit;
color: #738289;
font-weight: bold;
outline: none;
text-indent: 40px;
}
```

```
ul {
list-style: none;
width: 428px;
margin: 0 auto;
text-align: left;
}
```

```
ul li {
border-bottom: 1px solid #ddd;
padding: 10px;
overflow: hidden;
}
```

```
ul li img {
width: 60px;
height: 60px;
float: left;
border: none;
}
```

```
ul li p {
margin-left: 75px;
font-weight: bold;
```



```
padding-top: 12px;  
color: #6e7a7f;  
}
```

**7. In the explorer panel click filename->src->app.component.ts and then type the below code, select tsconfig.json file in explorer panel then type**

```
“noImplicitAny”:false,  
“noImplicitThis”:false,
```

```
import { Component } from '@angular/core';
```

```
@Component({  
  selector: 'app-root',  
  templateUrl: './app.component.html',  
  styleUrls: ['./app.component.scss'],  
})  
export class AppComponent {  
  searchString;  
  items = [  
    {  
      url: 'https://tutorialzine.com/2013/07/50-must-have-plugins-for-extending-twitter-bootstrap/',  
      title: '50 Must-have plugins for extending Twitter Bootstrap',  
      image: 'https://tutorialzine.com/media/2013/07/featured_4.jpg'  
    },  
    {  
      url: 'https://tutorialzine.com/2013/08/simple-registration-system-php-mysql/',  
      title: 'Making a Super Simple Registration System With PHP and MySQL',  
      image: 'https://tutorialzine.com/media/2013/08/simple_registration_system.jpg'  
    },  
    {  
      url: 'https://tutorialzine.com/2013/08/slideout-footer-css/',  
      title: 'Create a slide-out footer with this neat z-index trick',  
      image: 'https://tutorialzine.com/media/2013/08/slide-out-footer.jpg'  
    },  
    {
```

```

url: 'https://tutorialzine.com/2013/06/digital-clock/',
title: 'How to Make a Digital Clock with jQuery and CSS3',
image: 'https://tutorialzine.com/media/2013/06/digital_clock.jpg'
  },
  {
url: 'https://tutorialzine.com/2013/05/diagonal-fade-gallery/',
title: 'Smooth Diagonal Fade Gallery with CSS3 Transitions',
image: 'https://tutorialzine.com/media/2013/05/featured.jpg'
  },
  {
url: 'https://tutorialzine.com/2013/05/mini-ajax-file-upload-form/',
title: 'Mini AJAX File Upload Form',
image: 'https://tutorialzine.com/media/2013/05/ajax-file-upload-form.jpg'
  },
  {
url: 'https://tutorialzine.com/2013/04/services-chooser-backbone-js/',
title: 'Your First Backbone.js App – Service Chooser',
image: 'https://tutorialzine.com/media/2013/04/service_chooser_form.jpg'
  }
];
}

```

**8. In the explorer panel click filename->src->app.module.ts and the type the below code**

```

import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { SearchForPipe } from './search-for.pipe';

@NgModule({
  declarations: [
    AppComponent,
    SearchForPipe

```

```

    ],
    imports: [
      BrowserModule,
      FormsModule
    ],
    providers: [],
    bootstrap: [AppComponent]
  })
  export class AppModule { }

```

**9. In the explorer panel click filename->src->search.pipe.ts and the type the below code change the below code `var result=[]`; to `varresult:string[]=[]`;**

```

import { Pipe, PipeTransform } from '@angular/core';

@Pipe({
  name: 'searchFor'
})
export class SearchForPipe implements PipeTransform {

  transform(arr, searchString) {
    if (!searchString) {
      return arr;
    }

    var result = [];

    searchString = searchString.toLowerCase();

    // Using the forEach helper method to loop through the array
    arr.forEach(function (item) {
      if (item.title.toLowerCase().indexOf(searchString) !== -1) {
        result.push(item);
      }
    });
  }
}

```

return result;

}}

10. Save all the files and type `ng serve` in the terminal wait for some time then select the blue color highlighted text then press `Ctrl+Click` then your code is running you see the output

11. Press `Ctrl+C` to come out from running of server