

Assignment 10

Question 1:

1. Create the application which contains one service named as Arithmetic.

Arithmetic service contains two methods named as Add and Sub.

Both of this methods accepts two integers and perform addition and subtraction respectively.

We have to create one child component named as Demo under app component which uses Arithmetic service using Dependency Injection.

That methods from the service add and sub should be called from Demo component by passing some hardcoded values. And display the result of addition and subtraction inside Demo component.

Assignment 10

The screenshot shows the VS Code interface with the following details:

- EXPLORER** pane: Shows the project structure with files like `app.component.ts`, `app.component.html`, `demo.component.html`, etc.
- OPEN EDITORS** pane: Shows the current editors open, including `app.component.ts` and `app.component.html`.
- EDITOR** pane (Content):

```
<app-demo></app-demo>
<router-outlet></router-outlet>
```

The screenshot shows the VS Code interface with the following details:

- EXPLORER** pane: Shows the project structure with files like `app.component.ts`, `app.component.html`, `demo.component.html`, etc.
- OPEN EDITORS** pane: Shows the current editors open, including `app.component.ts` and `app.component.html`.
- EDITOR** pane (Content):

```
(alias) class ArithmeticService
import ArithmeticService

'ArithmeticService' is declared but its value is never
read. ts(6133)

Quick Fix... (⌘.)
import { ArithmeticService } from './arithmetic.service';

@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet, DemoComponent],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css'],
})
export class AppComponent {
  title = 'Assignment_10_1_Again';
}
```

Assignment 10

demo.component.html — Assignment_10_1_Again

EXPLORER ... app.component.ts 2 \leftrightarrow app.component.html \leftrightarrow demo.component.html X \triangleright \square

OPEN EDITORS

- ✓ TS app.compo... 2 \leftrightarrow app.component....
- ✗ \leftrightarrow demo.componen...
- ✓ TS demo.comp... 2
- TS arithmetic.s... 1

ASSIGN... \square \square \square \square

src

```
src > app > demo >  $\leftrightarrow$  demo.component.html > h2
Go to component
1 <h2>Assignment 10.1</h2>
2 <h2>Addition of add(10,20) is : {{ add(10, 20) }}</h2>
3 <h2>Subtraction of sub(10,20) is : {{ sub(10, 20) }}</h2>
4
```

demo.component.ts — Assignment_10_1_Again

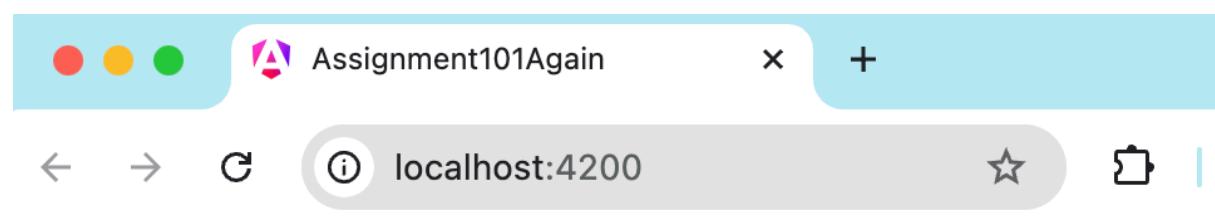
... mo.component.html \leftrightarrow TS demo.component.ts 2 X \leftrightarrow TS arithmetic.service.ts 1 \triangleright \square

```
src > app > demo > TS demo.component.ts > DemoComponent > sub
1 import { Component } from '@angular/core';
2 import { ArithmeticService } from '../arithmetic.service';
3
4 @Component({
5   selector: 'app-demo',
6   standalone: true,
7   imports: [],
8   templateUrl: './demo.component.html',
9   styleUrls: ['./demo.component.css'],
10 })
11 export class DemoComponent {
12   constructor(private arithmeticservice: ArithmeticService) {}
13
14   public add(num1: number, num2: number): number {
15     return this.arithmeticservice.Add(num1, num2);
16   }
17
18   public sub(num1: number, num2: number): number {
19     return this.arithmeticservice.Sub(num1, num2);
20   }
21 }
22
```

Assignment 10

```
arithmetic.service.ts — Assignment_10_1_Again
· mo.component.html    TS demo.component.ts 2    TS arithmetic.service.ts 1 × ▾
src > app > TS arithmetic.service.ts > ArithmeticService > Sub
1   import { Injectable } from '@angular/core';
2
3   @Injectable({
4     providedIn: 'root',
5   })
6   export class ArithmeticService {
7     constructor() {}
8
9     Add(Num1: number, Num2: number): number {
10       return Num1 + Num2;
11     }
12
13     Sub(Num1: number, Num2: number): number {
14       return Num1 - Num2;
15     }
16   }
17
```

Output:



Assignment 10.1

Addition of add(10,20) is : 30

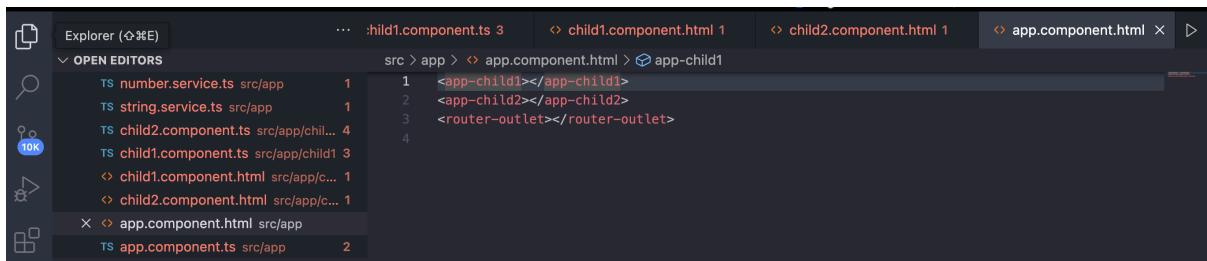
Substraction of sub(10,20) is : -10

Assignment 10

Question 2:

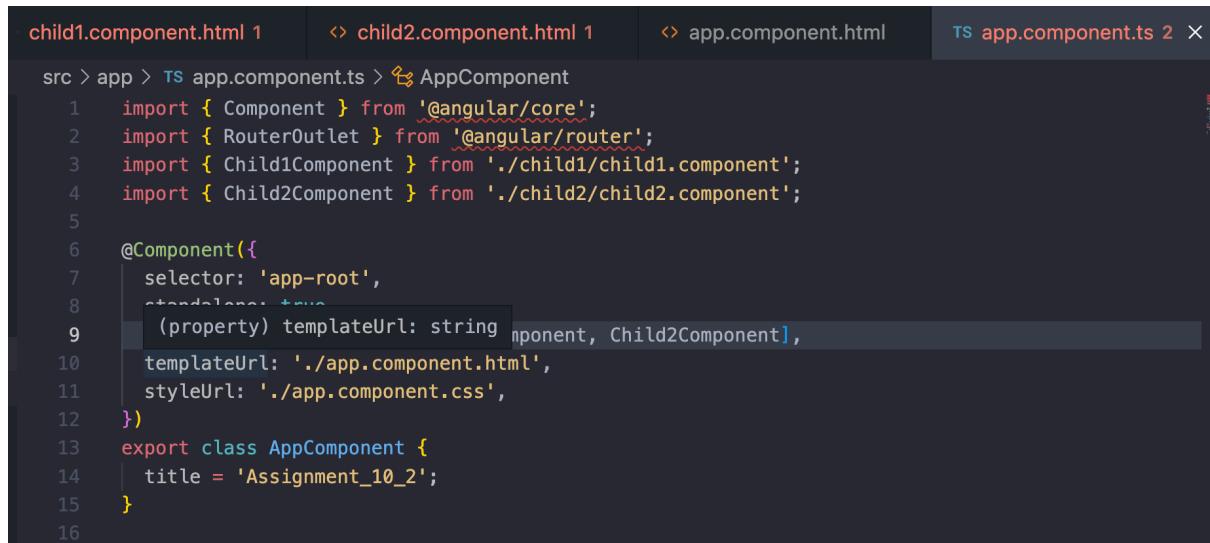
2. Create the application which contains two services named as Number and String.
Number service contains one method named as ChkPrime() which accepts number and check whether that number is prime or not.
String service contains one method named as CountCapital() which counts number of capital characters and return its count.
We have to create two child component named as Child1 & Child2 under app component
Child1 uses Number service & Child2 uses String service using Dependency Injection.
Call both the methods from the respective components by passing some hardcoded values and display the result.

Assignment 10

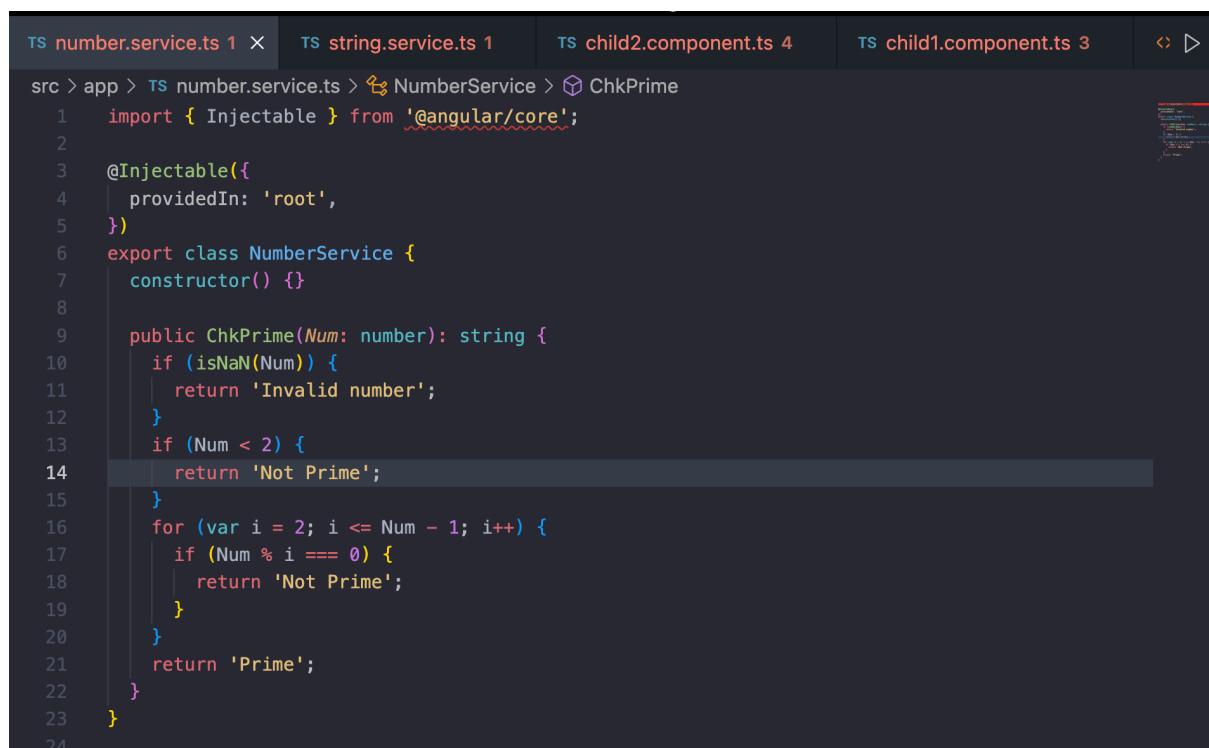


The screenshot shows the VS Code interface with the Explorer view open. The sidebar on the left has icons for file, folder, search, and refresh. The main area displays a list of open editors:

- TS number.service.ts src/app
- TS string.service.ts src/app
- TS child2.component.ts src/app/child...
- TS child1.component.ts src/app/child1
- ▷ child1.component.html src/app/c...
- ▷ child2.component.html src/app/c...
- ▷ app.component.html src/app
- TS app.component.ts src/app



```
src > app > TS app.component.ts < AppComponent
1 import { Component } from '@angular/core';
2 import { RouterOutlet } from '@angular/router';
3 import { Child1Component } from './child1/child1.component';
4 import { Child2Component } from './child2/child2.component';
5
6 @Component({
7   selector: 'app-root',
8   templateUrl: './app.component.html',
9   styleUrls: ['./app.component.css'],
10 })
11 export class AppComponent {
12   title = 'Assignment_10_2';
13 }
14
15
16
```



```
src > app > TS number.service.ts > NumberService > ChkPrime
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root',
5 })
6 export class NumberService {
7   constructor() {}
8
9   public ChkPrime(Num: number): string {
10     if (isNaN(Num)) {
11       return 'Invalid number';
12     }
13     if (Num < 2) {
14       return 'Not Prime';
15     }
16     for (var i = 2; i <= Num - 1; i++) {
17       if (Num % i === 0) {
18         return 'Not Prime';
19       }
20     }
21     return 'Prime';
22   }
23 }
```

Assignment 10

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files in the current workspace, including `number.service.ts`, `string.service.ts`, `child2.component.ts`, `child1.component.ts`, `app.config.ts`, `app.routes.ts`, `number.service.spec.ts`, `string.service.spec.ts`, and `string.service.ts`.
- Code Editor:** The active file is `string.service.ts`. The code implements a `CountCapital` service that counts uppercase letters in a given string.

```
src > app > TS string.service.ts 1
1 import { Injectable } from '@angular/core';
2 import { UpperCasePipe } from '../../../../../node_modules/@angular/common/index';
3 import { numberAttribute } from '../../../../../node_modules/@angular/core/index';
4
5 @Injectable({
6   providedIn: 'root',
7 })
8 class StringService {
9   constructor() {}
10
11   public CountCapital(Context: string): number {
12     var Cnt: number = 0;
13     for (var i = 0; i < Context.length; i++) {
14       if (Context[i] >= 'A' && Context[i] <= 'Z') {
15         Cnt++;
16       }
17     }
18
19     return Cnt;
20   }
21 }
22
```

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files in the current workspace, including `number.service.ts`, `string.service.ts`, `child2.component.ts`, `child1.component.ts`, `child1.component.html`, `child1.component.css`, `child1.component.spec.ts`, `app.component.ts`, `app.component.html`, `app.component.css`, `app.component.spec.ts`, `app.config.ts`, and `app.routes.ts`.
- Code Editor:** The active file is `child1.component.html`. It contains HTML code for a child component that calls a function from the Number service.

```
src > app > child1 > child1.component.html > div.Child1 > h2
1 <div class="Child1">
2   <h2 style="color: brown">
3     Child1 Component calling function from Number service
4   </h2>
5   <h2>check if Prime or Not</h2>
6   <input type="text" [(ngModel)]="num" />
7   <h2>{{ chkPrime(num) }}</h2>
8   <br />
9 </div>
10
```

Assignment 10

OutPut :

The screenshot shows a code editor with multiple tabs at the top: string.service.ts, child2.component.ts, child1.component.ts (selected), and child1.com. The child1.component.ts tab has a red 'X' icon. The code in the editor is as follows:

```
src > app > child1 > child1.component.ts > ...
1 import { Component } from '@angular/core';
1 import { CommonModule } from '@angular/common';
4 import { FormsModule } from '@angular/forms';
3
1 import { NumberService } from '../number.service';
5
1 @Component({
1   selector: 'app-child1',
1   templateUrl: './child1.component.html',
2   styleUrls: ['./child1.component.css'],
1 })
13 export class Child1Component {
14   public num: number = 0;
15   constructor(private numberserviceobj: NumberService) {}
16
17   public chkPrime(num: number): string {
18     return this.numberserviceobj.ChkPrime(this.num);
19   }
20 }
21
```

The screenshot shows a code editor with multiple tabs at the top: child1.component.ts, child1.component.html (selected), child2.component.html, and app.component.ts. The child1.component.html tab has a red 'X' icon. The code in the editor is as follows:

```
src > app > child2 > child2.component.html > div.Child2 > h2
1 <div class="Child2">
2   <h2 style="color: brown">
3     | Child2 Component calling function from String service
4   </h2>
5   <h2>Check the Capital Letters count in provided string</h2>
6   <input type="text" [(ngModel)]="str" />
7   <h2>{{ countCapital(str) }}</h2>
8   <br />
9 </div>
10
```

Assignment 10

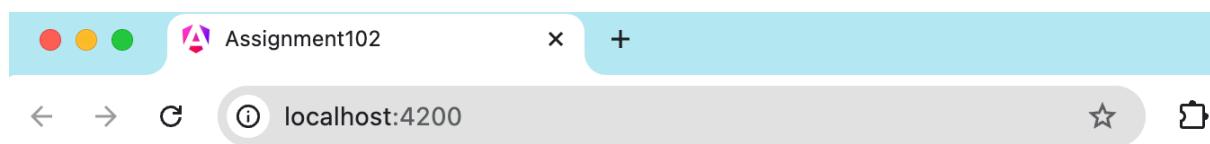


Child2 Component calling function from String service

Check the Capital Letters count in provided string

HelloZ

2



Child2 Component calling function from String service

Check the Capital Letters count in provided string

Assignment 10

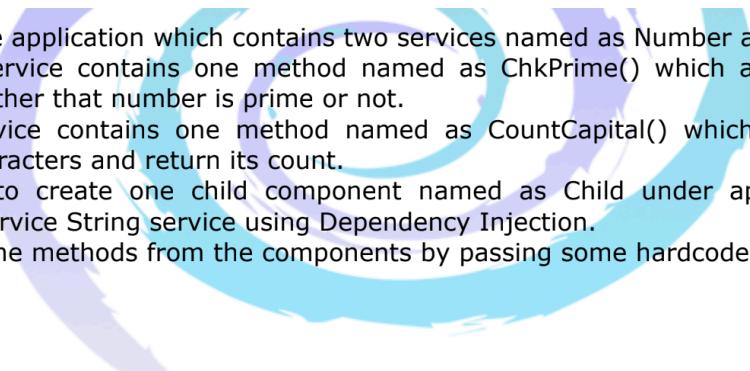
The screenshot shows a web browser window with the title "Assignment102". The address bar displays "localhost:4200". The page content is divided into two main sections:

- Child1 Component calling function from Number service**
check if Prime or Not
An input field contains the value "0".
Not Prime
- Child2 Component calling function from String service**
Check the Capital Letters count in provided string
An input field is empty.
0

Assignment 10

Question 3:

3. Create the application which contains two services named as Number and String.
Number service contains one method named as ChkPrime() which accepts number and check whether that number is prime or not.
String service contains one method named as CountCapital() which counts number of capital characters and return its count.
We have to create one child component named as Child under app component uses Number service String service using Dependency Injection.
Call both the methods from the components by passing some hardcoded values and display the result.

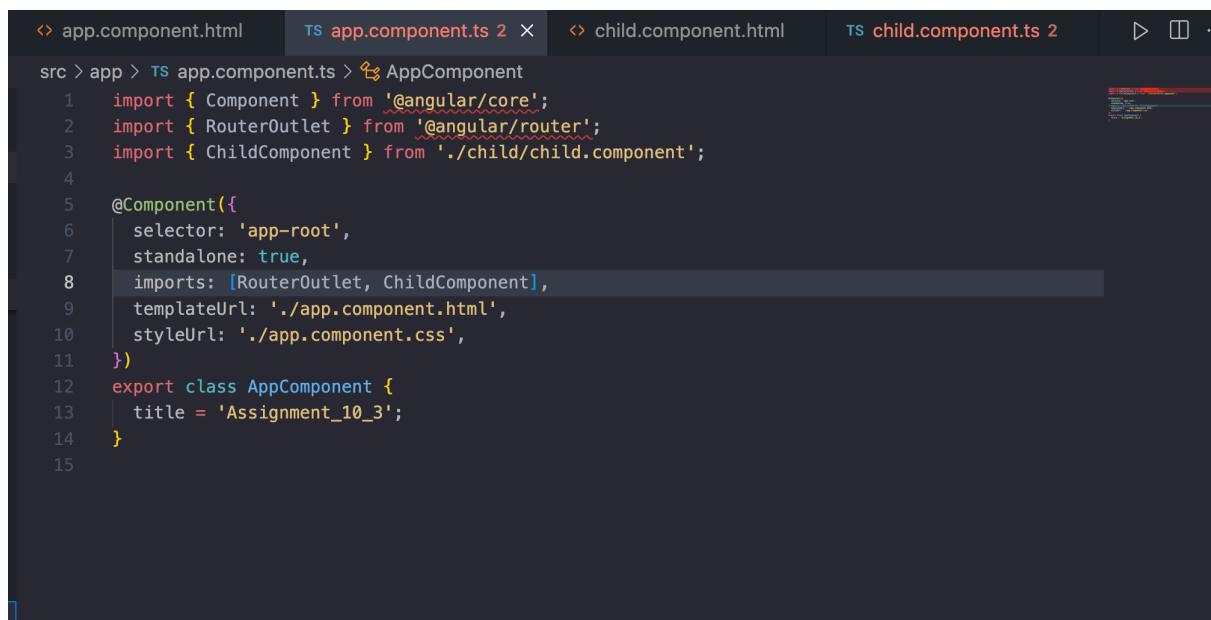


The screenshot shows a code editor window titled "app.component.html — Assignment_10_3". The Explorer sidebar on the left lists files and folders. In the main editor area, the file "app.component.html" is open, showing the following code:

```
src > app > app.component.html > router-outlet
1 <app-child></app-child>
2 <router-outlet></router-outlet>
3
```

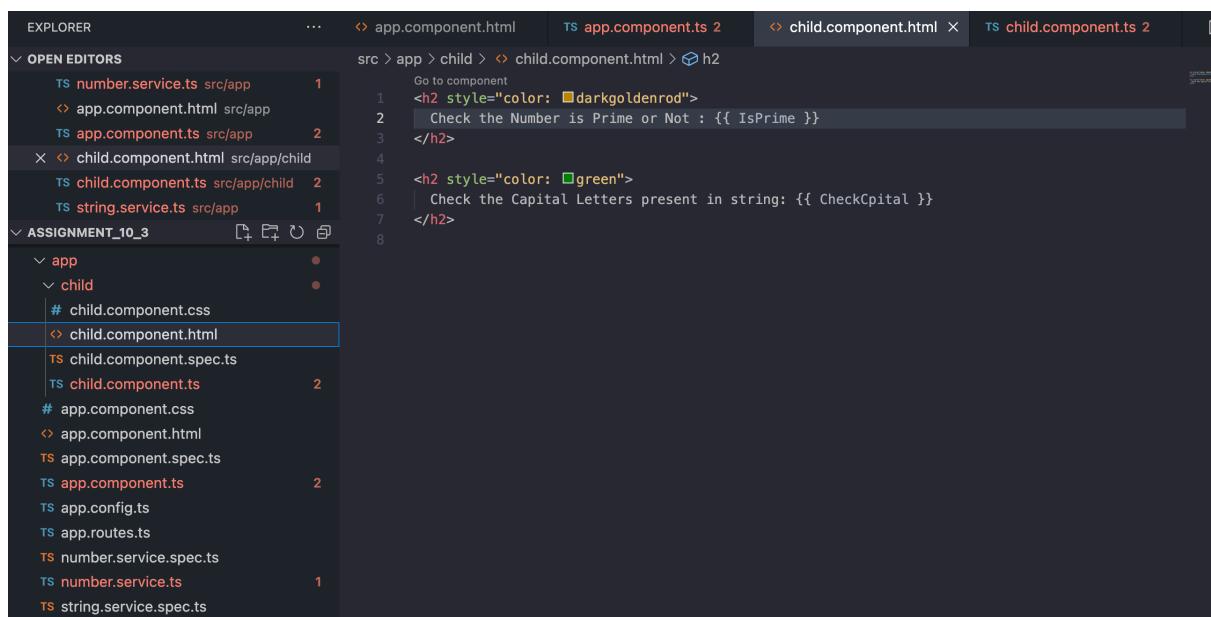
The "OPEN EDITORS" section shows other files like "number.service.ts", "app.component.ts", and "child.component.ts" are open. The "ASSIGNMENT_10_3" folder contains files such as "app", "child", "child.component.css", "child.component.html", "child.component.spec.ts", "child.component.ts", "app.component.css", "app.component.html", "app.component.spec.ts", "app.component.ts", "app.config.ts", "app.routes.ts", "number.service.spec.ts", "number.service.ts", and "string.service.spec.ts".

Assignment 10



The screenshot shows a code editor with the following tabs: app.component.html, TS app.component.ts 2, child.component.html, and TS child.component.ts 2. The active tab is app.component.ts. The code is as follows:

```
src > app > TS app.component.ts > AppComponent
1 import { Component } from '@angular/core';
2 import { RouterOutlet } from '@angular/router';
3 import { ChildComponent } from './child/child.component';
4
5 @Component({
6   selector: 'app-root',
7   standalone: true,
8   imports: [RouterOutlet, ChildComponent],
9   templateUrl: './app.component.html',
10  styleUrls: ['./app.component.css'],
11})
12 export class AppComponent {
13   title = 'Assignment_10_3';
14 }
15
```



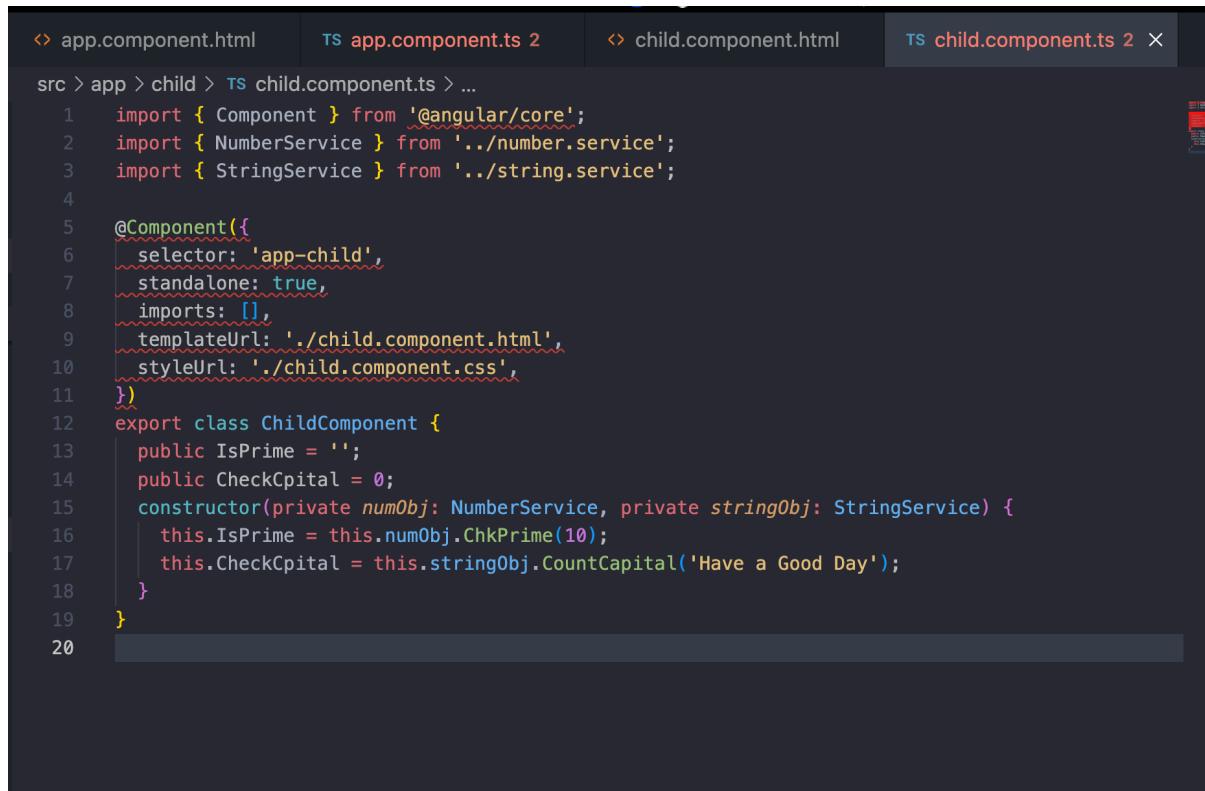
The screenshot shows a code editor with the following tabs: app.component.html, TS app.component.ts 2, child.component.html, and TS child.component.ts 2. The active tab is child.component.ts. The code is as follows:

```
src > app > child > child.component.html > h2
1 Go to component
2 <h2 style="color: darkgoldenrod">
3 | Check the Number is Prime or Not : {{ IsPrime }}
4 | </h2>
5 <h2 style="color: green">
6 | Check the Capital Letters present in string: {{ CheckCpital }}
7 | </h2>
8
```

The Explorer sidebar on the left shows the project structure:

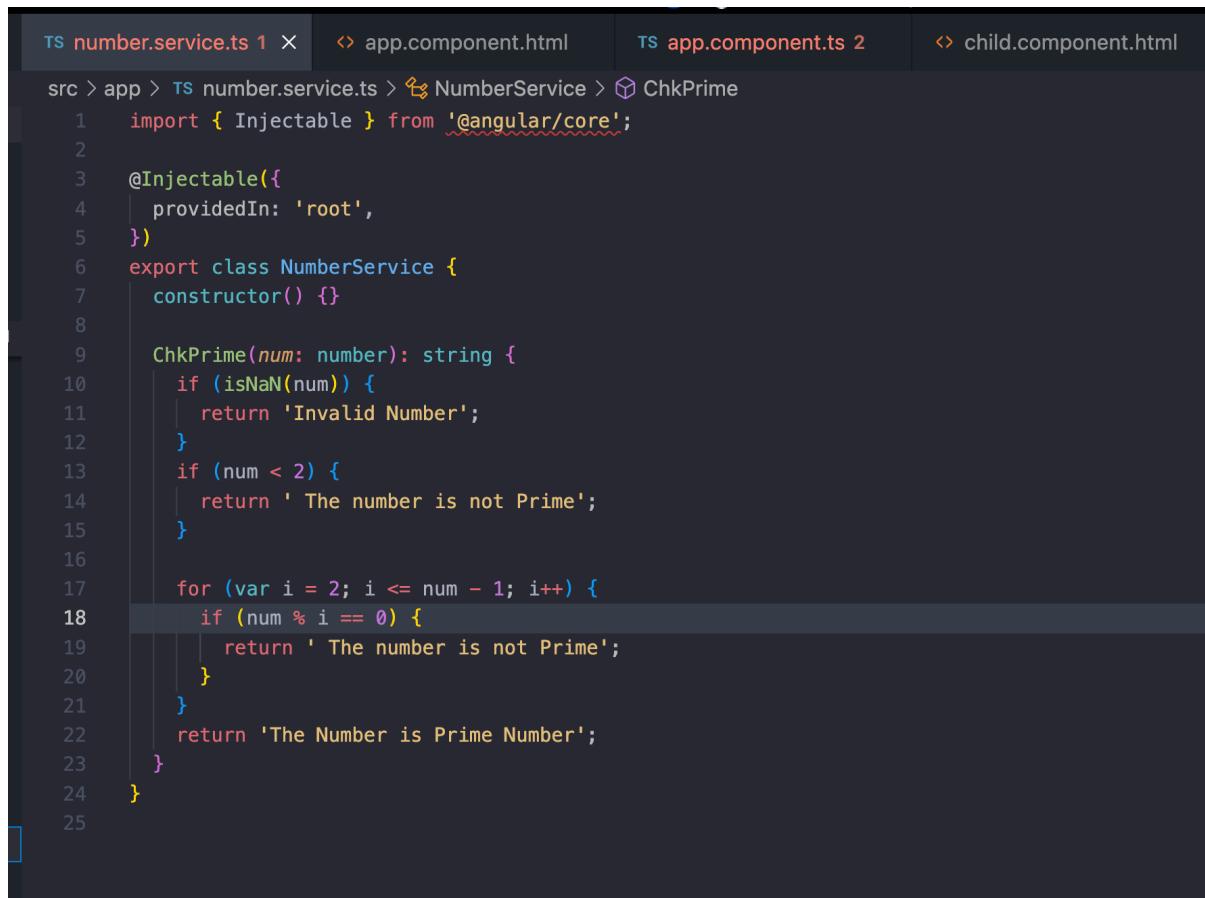
- OPEN EDITORS:
 - TS number.service.ts src/app
 - TS app.component.html src/app
 - TS app.component.ts src/app
 - TS child.component.html src/app/child
 - TS child.component.ts src/app/child
 - TS string.service.ts src/app
- ASSIGNMENT_10_3:
 - app
 - child
 - # child.component.css
 - child.component.html
 - TS child.component.spec.ts
 - TS child.component.ts
 - # app.component.css
 - TS app.component.html
 - TS app.component.spec.ts
 - TS app.component.ts
 - TS app.config.ts
 - TS app.routes.ts
 - TS number.service.spec.ts
 - TS number.service.ts
 - TS string.service.spec.ts

Assignment 10



The screenshot shows a code editor with multiple tabs open. The active tab is 'child.component.ts' (version 2). The code defines a component named 'ChildComponent' with the selector 'app-child'. It imports 'NumberService' and 'StringService' from their respective services. The component has properties 'IsPrime' and 'CheckCpital'. In the constructor, it initializes 'IsPrime' to the value returned by 'ChkPrime(10)' and 'CheckCpital' to the value returned by 'CountCapital('Have a Good Day')'.

```
src > app > child > child.component.ts > ...
1 import { Component } from '@angular/core';
2 import { NumberService } from '../number.service';
3 import { StringService } from '../string.service';
4
5 @Component({
6   selector: 'app-child',
7   standalone: true,
8   imports: [],
9   templateUrl: './child.component.html',
10  styleUrls: ['./child.component.css'],
11})
12 export class ChildComponent {
13   public IsPrime = '';
14   public CheckCpital = 0;
15   constructor(private numObj: NumberService, private stringObj: StringService) {
16     this.IsPrime = this.numObj.ChkPrime(10);
17     this.CheckCpital = this.stringObj.CountCapital('Have a Good Day');
18   }
19 }
20
```



The screenshot shows a code editor with multiple tabs open. The active tab is 'number.service.ts' (version 1). The code defines a service named 'NumberService' with a single method 'ChkPrime'. This method checks if a given number is prime. It first checks if the number is NaN or less than 2. If so, it returns an error message. Otherwise, it loops from 2 to num-1, checking if num % i == 0. If it finds such a divisor, it returns 'The number is not Prime'. If no divisors are found, it returns 'The Number is Prime Number'.

```
src > app > number.service.ts > NumberService > ChkPrime
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root',
5 })
6 export class NumberService {
7   constructor() {}
8
9   ChkPrime(num: number): string {
10   if (isNaN(num)) {
11     return 'Invalid Number';
12   }
13   if (num < 2) {
14     return ' The number is not Prime';
15   }
16
17   for (var i = 2; i <= num - 1; i++) {
18     if (num % i == 0) {
19       return ' The number is not Prime';
20     }
21   }
22   return 'The Number is Prime Number';
23 }
24 }
```

Assignment 10

```
TS app.component.ts 2      <> child.component.html      TS child.component.ts 2      TS string.service.ts 1 X
c > app > TS string.service.ts > StringService > CountCapital
1   import { Injectable } from '@angular/core';
2
3   @Injectable({
4     providedIn: 'root',
5   })
6   export class StringService {
7     constructor() {}
8
9     public CountCapital(str: string): number {
10       var cnt: number = 0;
11
12       for (var i = 0; i < str.length; i++) {
13         if (str[i] >= 'A' && str[i] <= 'Z') {
14           cnt++;
15         }
16       }
17       return cnt;
18     }
19   }
20
```

OutPut:



Check the Number is Prime or Not : The number is not Prime

Check the Capital Letters present in string: 3

Assignment 10

Assignment 10