ICP5

Name: Aqdus Charolia
Email: aacy5x@mail.umkc.edu
Github: https://github.com/Aqdusc/WebDevCourse/tree/main/Web_Development/ICP5

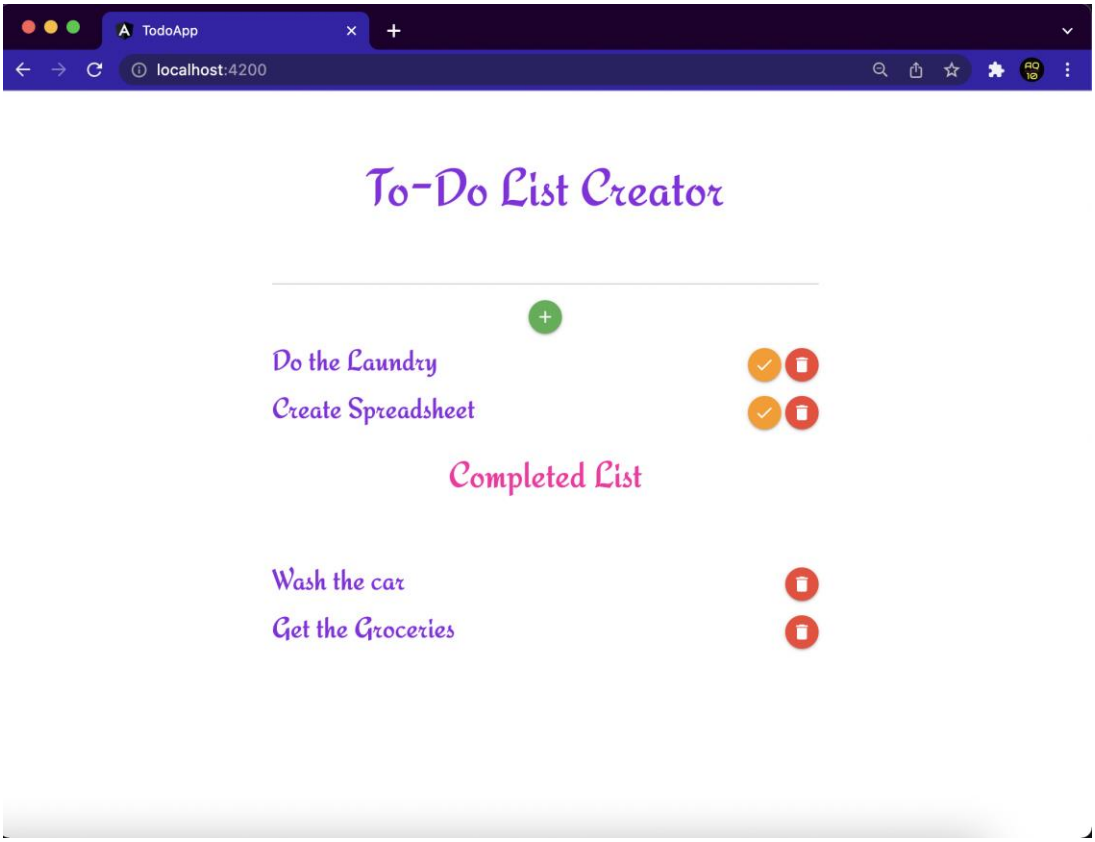
Name: Affan Charolia
Email: aacbb8@mail.umkc.edu
Github: https://github.com/Affancharolia/WebDevCourse/tree/main/Web_Development/ICP5

Aim: To use Angular Typescript to develop a to-do list and a timer.

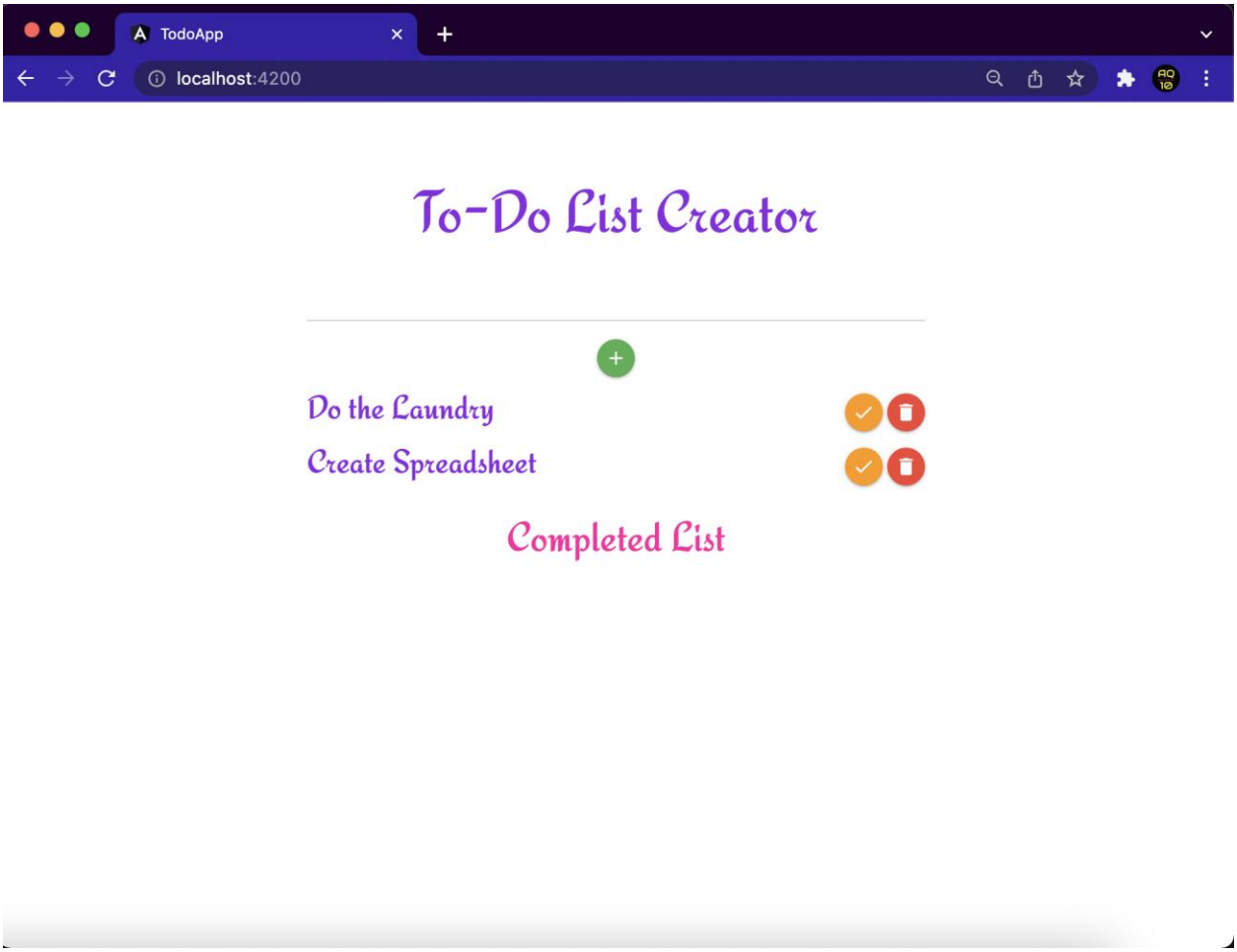
Explanation/Screenshots:

Part 1 To-do list: Created a basic to-do list webapp using angular that takes the input and displays the tasks, when the task is completed you click it and it moves to completed list and then you can delete the task.

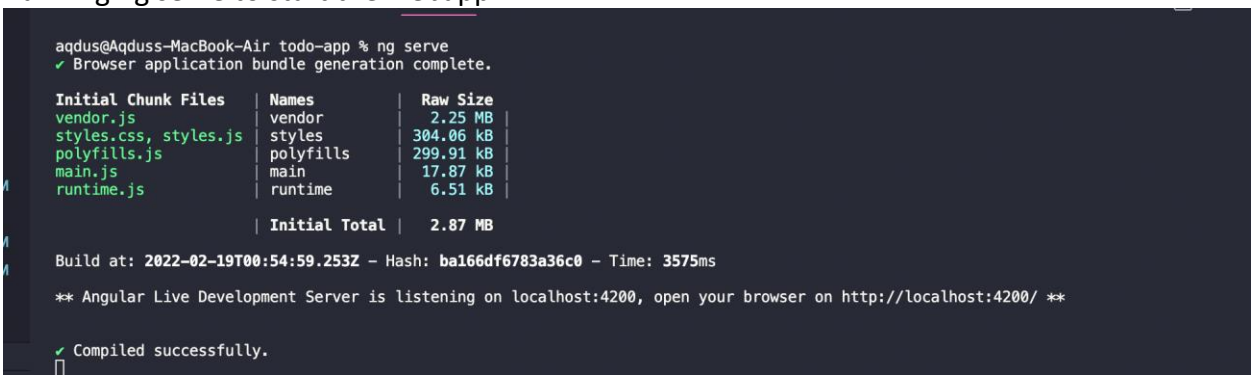
App Screenshot:



After Deleting the Wash the car and get the groceries list.



Running ng serve to start the webapp



Index.html file

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>TodoApp</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Adding the font theme from google -->
  <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Redressed">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <!-- Adding the bootstrap css files to the index.html file -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"
     integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
     integrity="sha384-
ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
  <style>
    /* Creating the general css features to be added to the website */
    body{
    font-family: "Redressed", sans-serif;
    color:blueviolet;
    font-size: 35px;
    }
  </style>
    </head>
<body>
  <!-- Calling the app.component.html file where the main html and css is made -->
  <app-root></app-root>
</body>
</html>
```

The app.component.html file

```
<!-- here the main html file is created -->
<div class="container">
  <h1>To-Do List Creator</h1>
  <!-- Here we use typescript that creates a from where you enter the text and adds to the
  the to-do list -->
```

```
 <form class="todo-form" #f="ngForm" (ngSubmit)="onSubmit(f)">
   <div>
    <!-- Creating the input field which calls the ngModel typescript function which add the input
    to the to-do list -->
    <input type="text" id="title" name="title" ngModel>
    <!-- After the text has been entered and the add button pressed the to-do is added -->
    <button type="submit" class="btn-floating waves-effect waves-light green"><i class="material-
icons">add</i></button>
   </div>
 </form>

<!-- This part of the html deals with the checking of the to-do and deleting it if completed  -->
 <ul class="todo-list">
   <!-- Here we can recieve every to-do that is obtained
    from the form by running a loop and adding it to the list.  -->
   <li  *ngFor="let todo of todos">
    <!-- Here we check if the to-do is not complete we give the option of checking it else we delete it
-->
    <div  *ngIf="!this.todo.complete">
      <div>{{this.todo.title}}</div>
      <div>
        <button (click)="onComplete(this.todo.id)" class="btn-floating waves-effect waves-light
orange"><i class="material-icons">check</i></button>
        <button (click)="onDelete(this.todo.id)" class="btn-floating waves-effect waves-light red"><i
class="material-icons">delete</i></button>
      </div>
    </div>
   </li>

   <h3 style="color: rgba(255, 0, 157, 0.959);">Completed List</h3>
   <br>
   <li  *ngFor="let todo of todos">
    <!-- Here is the to-do is checked complete it is moved to the to-do complete portion where it can
only be deleted -->
    <div  *ngIf="this.todo.complete">
      <div>{{this.todo.title}}</div>
      <div>
        <button (click)="onDelete(this.todo.id)" class="btn-floating waves-effect waves-light red"><i
            class="material-icons">delete</i></button>
      </div>
    </div>
```

```
    </li>
  </ul>
</div>
```

App.component.ts

```
// this is our main typescript file where we create the functionality of the webapp and also create the
//objects and variables to store and display the data and the operations to be performed on the input
data
// and the to-do list

//importing all the documents from libraries and angular modules
import { Component } from '@angular/core';
import { NgForm } from '@angular/forms';
import { Guid } from 'guid-typescript';// importing the guid-typescript after installing it
import { from } from 'rxjs';
import { Todo } from 'src/models/todo.model';//importing the to-do class created in the models file
@Component({
  selector: 'app-root', //here the selector points to this template app.component.ts file
  templateUrl: './app.component.html',// points to the .html file where the html file is written
  styleUrls: ['./app.component.css']// points to the .css file created
})
//exporting the class Appcomponet
//here we start creating and initializing all the variables to be used in the webapp
//create an array in which all the to-do are added
//have initialized with 2 basic tasks.
export class AppComponent {
  todos: Todo[] = [
    new Todo(Guid.create(),'Wash Car',false),
    new Todo(Guid.create(), 'Buy Groceries', false),

  ]
//here when the form is submitted a new to-do is added to the list
  onSubmit(form: NgForm){
    let todo=new Todo(Guid.create(),form.value.title, false);
    this.todos.push(todo);
    form.resetForm();// we reset the form for new input task to add
  }
// here the oncomplete function moves the completed task to the completed list
  onComplete(id: Guid){
    let todo=this.todos.filter(x=>x.id===id)[0];
    todo.complete=true;
```

```
  }
// here on detleting the completed task the function deletes the task form the list by using the id
  onDelete(id: Guid){
    let todo=this.todos.filter(x=>x.id===id)[0];
    let index=this.todos.indexOf(todo,0);
    if(index>-1){
      this.todos.splice(index,1);// finally we perform splice operation
    }
  }
}
```

Model.todo.ts

```
//here in todo.model.ts i have imported Guid for marking each task input to an id when inputted
import { Guid } from "guid-typescript";
//creating the class todo with the variables in the constructor
export class Todo{
    constructor(
        public id:Guid,// for each task an id is assigned
        public title: string, //for each task the input is string
        public complete:boolean //for completion we create a boolean to check if task is completed.
    ){}
}
```

App.component.css

```
/* creating the basic css layout and adding to the app.component.html file */

.container{
    padding: 3% 10%;
    margin: 0 auto;
    text-align: center;
}

.todo-list> li>div{
    display: flex;
    flex-direction: row;
    justify-content: space-between;
}
```

```
.todo-list >li{
    margin-top: 5px;
}

.todo-list button{
    margin-left: 5px;
}
```

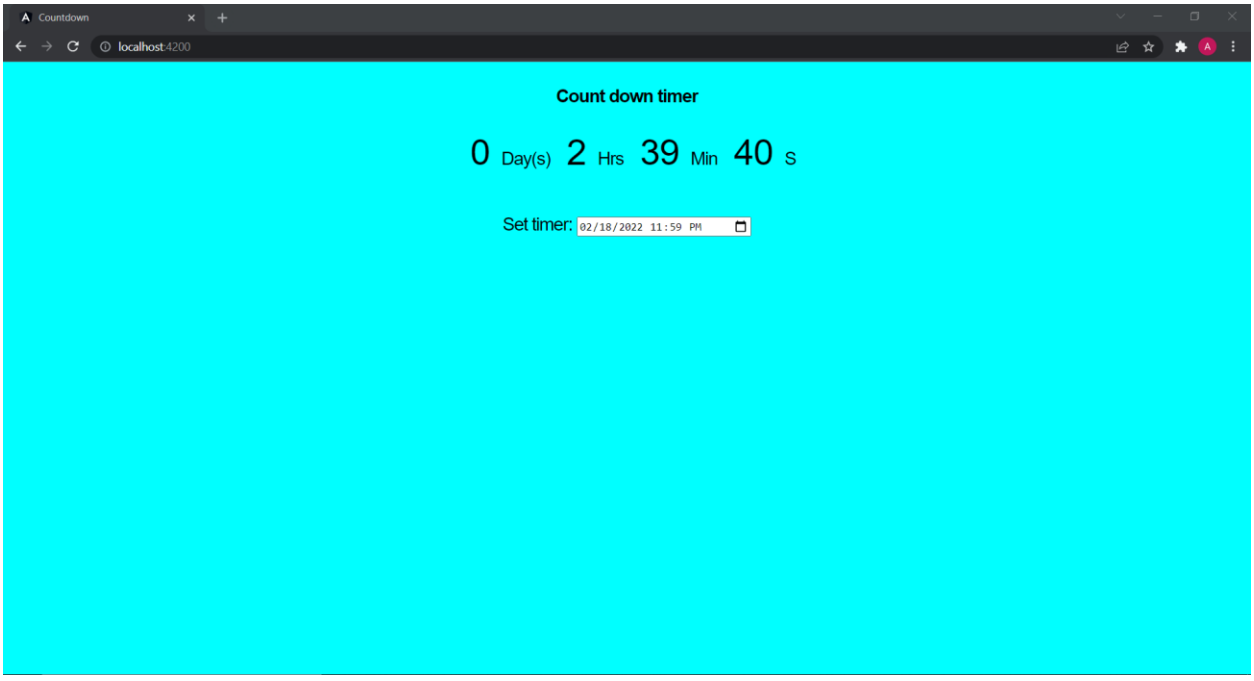
Hence in this fashion the to-do list was created.

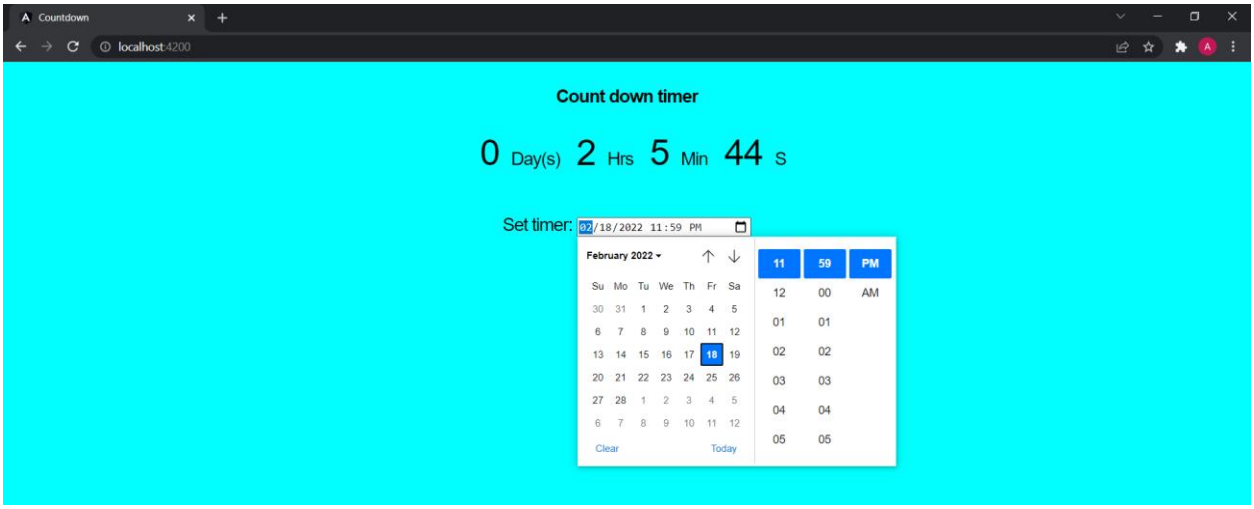
For all the styles and icons the following files were installed by ng command.

```
1. "./node_modules/material-icons/iconfont/material-icons.css",
2. "./node_modules/materializecss/dist/css/materialize.min.css"
```

Part 2 Count Down Timer: Created a basic count down timer which when it gets an input from the user starts to count in a decrementing manner.

App Screenshot:





Html:

We use the datetime-local type in input tag to get the date time selected by user.

component.html M | index.html M | styles.css | count-down.component.css U | count-down.component.html U | count-down.component.ts U

src > app > count-down > count-down.component.html > div.home > div.timer > div.give > label.Set

Go to component

```
<div class="home">
    <div class="timer">
        <h4> Count down timer </h4>
        <span id="days"> {{day}} </span>Day(s)
        <span id="hours"> {{hrs}} </span>Hrs
        <span id="minutes"> {{mins}} </span>Min
        <span id="seconds"> {{secs}} </span>S
    <div class="give">
    <label class="Set">
        Set timer:
        <input type="datetime-local" [(ngModel)]="dateTime">

    </label>
    </div>

    </div>
</div>
```

CSS:

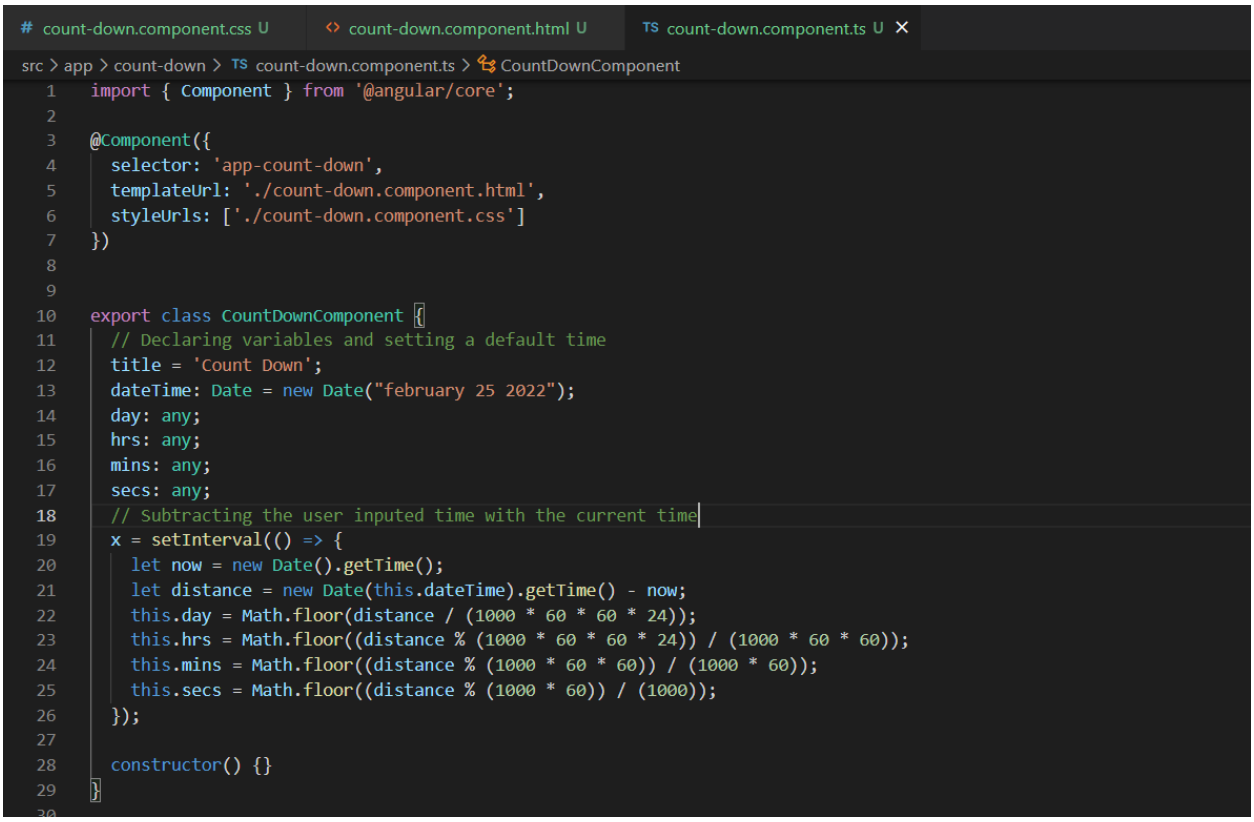
component.html M | index.html M | styles.css | count-down.component.css U | count-down.component.html U | count-down.component.ts U

src > app > count-down > count-down.component.css > .give

```
.timer {
  text-align: center;
  font-family: Arial, sans-serif;
  font-size: 1.4em;
  letter-spacing: -1px;
}
.timer span {
  font-size: 2em;
  margin: 0 3px 0 15px;
}

.give {
  margin: 50px;
}
```

Type Script:


count-down.component.css U
count-down.component.html U
TS count-down.component.ts U
src > app > count-down > TS count-down.component.ts > CountDownComponent
1 import { Component } from '@angular/core';
2
3 @Component({
4 selector: 'app-count-down',
5 templateUrl: './count-down.component.html',
6 styleUrls: ['./count-down.component.css']
7 })
8
9
10 export class CountDownComponent {
11 // Declaring variables and setting a default time
12 title = 'Count Down';
13 dateTime: Date = new Date("february 25 2022");
14 day: any;
15 hrs: any;
16 mins: any;
17 secs: any;
18 // Subtracting the user inputed time with the current time
19 x = setInterval(() => {
20 let now = new Date().getTime();
21 let distance = new Date(this.dateTime).getTime() - now;
22 this.day = Math.floor(distance / (1000 * 60 * 60 * 24));
23 this.hrs = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
24 this.mins = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
25 this.secs = Math.floor((distance % (1000 * 60)) / (1000));
26 });
27
28 constructor() {}
29 }