Anusha Nelluri – anhx2@umsystem.edu

GitHub Link - https://github.com/NelluriAnusha/Demo_Remote/tree/main/Webpart/ICP5

Achyuth Kumar Valeti – avgh3@umsystem.edu

GitHub Link - https://github.com/AchyuthValeti/Demo_Remote/tree/main/Webpart/ICP5

ICP5 (Angular)

Introduction:

Angular is an open-source framework that allows us to create dynamic and single-page applications (SPAs) for both desktop & mobile. It creates apps with Typescript and HTML. Angular is written entirely in Typescript.

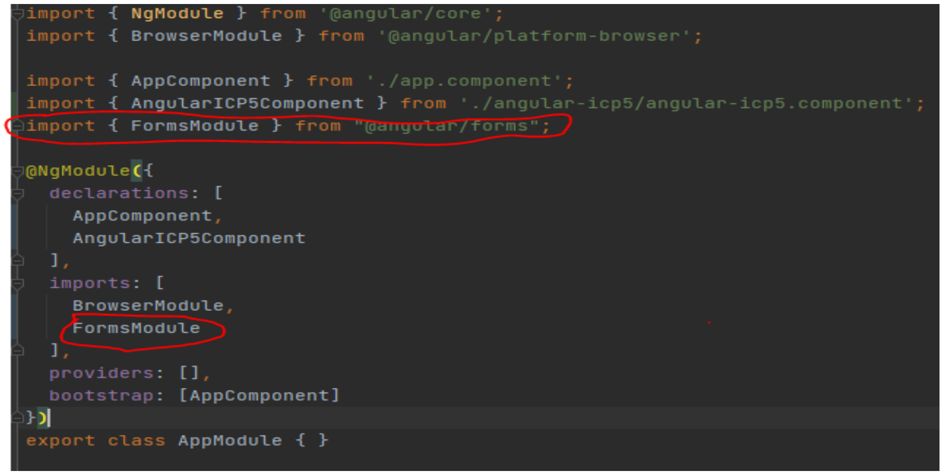
The importance of Angular and its components (components, string interpolation, property binding, event, and two-way data binding, Ng Modules, and directives) are discussed in this assignment.

Tasks:

Installation:

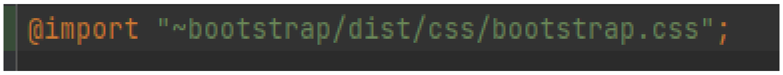
We have installed Node.js & Angular CLI in our machines and created a new project/application using ‘ng new’ command. Angular CLI handles various libraries configuration and initialization.

- Here, we are going to develop the following two applications using angular elements.
  1. To-do application
  2. Countdown Timer
- Imported ‘FormModule’ for form implementation in ‘app.module.ts’ file as below.



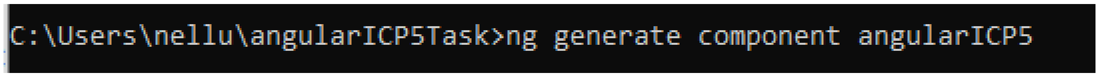
```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { AngularICP5Component } from './angular-icp5/angular-icp5.component';
import { FormsModule } from "@angular/forms";

@NgModule({
  declarations: [
    AppComponent,
    AngularICP5Component
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

- We have also used Bootstrap which makes our applications responsive as well.
- In newly created application, updated ‘style.css’ file with bootstrap details as below and executed ‘npm install bootstrap’ command on command prompt.



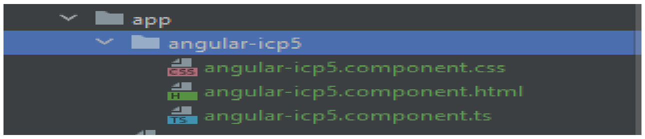
```
@import "~bootstrap/dist/css/bootstrap.css";
```

- Created a component ‘angular-icp5’ which shows a UI element. We can have a lot of these components on a single web page. Each component maintains a section of the page and independent of each other. We can also have both child & parent components for component.
  Executed the following command on command prompt to create a component:



```
C:\Users\nellu\angularICP5Task>ng generate component angularICP5
```

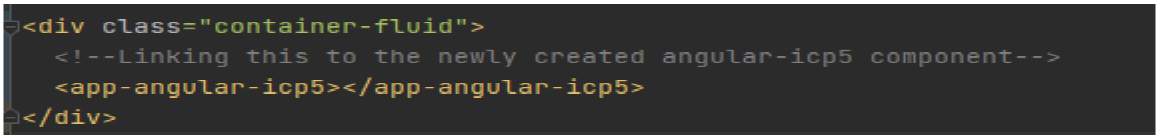
- After the component creation is done, the following files can be seen in ‘app’ folder.



```
app
  angular-icp5
    angular-icp5.component.css
    angular-icp5.component.html
    angular-icp5.component.ts
```

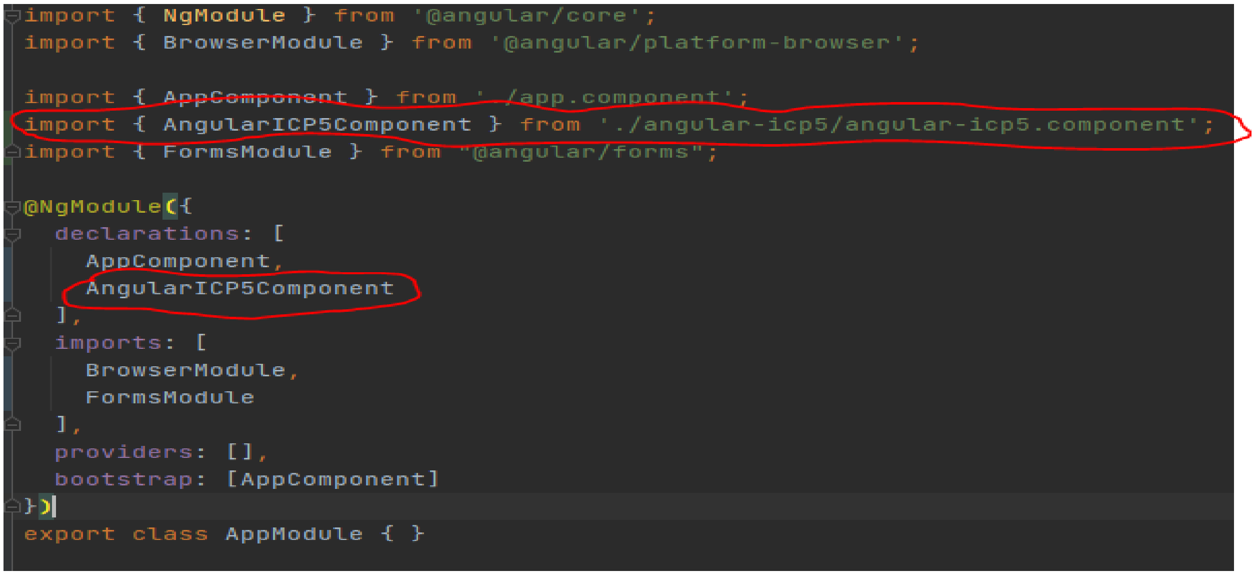
- ‘app.module.ts’ & ‘app.component.html’ files are updated with respective component changes.

app.component.html code:



```
<div class="container-fluid">
  <!--Linking this to the newly created angular-icp5 component-->
  <app-angular-icp5></app-angular-icp5>
</div>
```

app.module.ts:

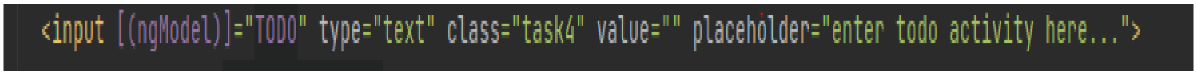


```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { AngularICP5Component } from './angular-icp5/angular-icp5.component';
import { FormsModule } from "@angular/forms";

@NgModule({
  declarations: [
    AppComponent,
    AngularICP5Component
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Note: Newly created 'angular-icp5.component.html', 'angular-icp5.component.css' & 'angular-icp5.component.ts' files are implemented individually for two Tasks and merged the code into a single file to get the output in a single web application.

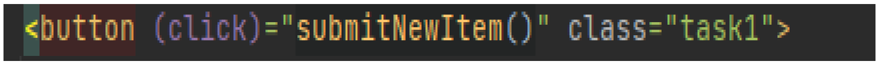
1.To-do application:

- We created a basic ToDO web application using Angular and its elements. The main purpose of this ToDo application is to remind us of important tasks. We just add the tasks and then delete them after they are completed.
- We used HTML, CSS, & TypeScript and followed the below steps for implementation.
  1. Firstly, we used different tags of HTML like <div>, <input>, <button> and so on for creating this application.
  2. For 'input' tag, we used 'ngModel' directive to bind the data which implements two-way binding in Angular forms. 'ngModel' is part of FormsModule and that is the reason we have installed FormsModule into the ngModule.
  3. We bind the 'TODO' property to the input element. And we also used the event binding as it updates the property whenever input changes.



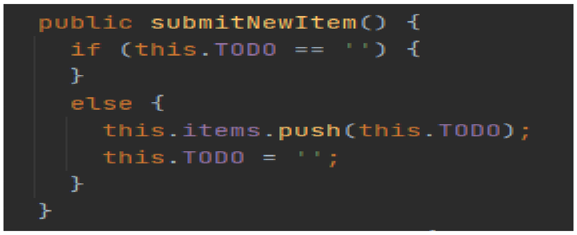
```
<input [(ngModel)]="TODO" type="text" class="task4" value="" placeholder="enter todo activity here...">
```

  4. Added a button with 'click' event which is declared with event binding syntax i.e., bracket symbol (). This event name is assigned to a function 'submitNewItem()' and it will be called when the button is clicked.



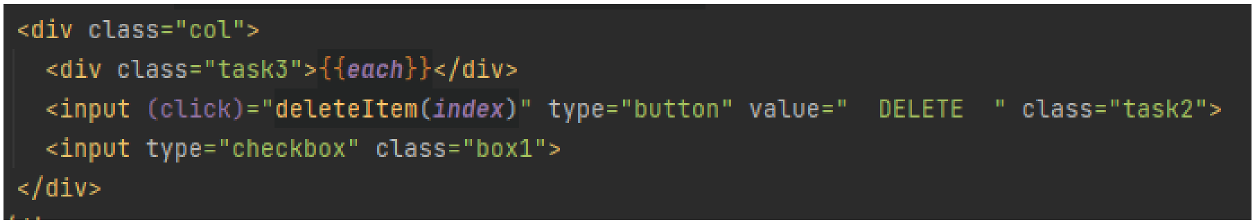
```
<button (click)="submitNewItem()" class="task1">
```

  5. In component class, we defined the method for handling the task i.e., adding new ToDo tasks to the list.



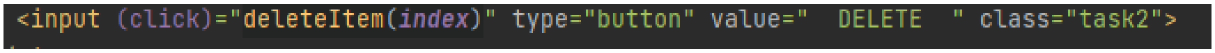
```
public submitNewItem() {
  if (this.TODO == '') {
  }
  else {
    this.items.push(this.TODO);
    this.TODO = '';
  }
}
```

  6. Used '*ngFor' to display each item for the specified collection on the webpage. It is a structural directive that makes our job easy is to iterate over array or list and so on.
  7. Here, we used interpolation to display the data like '{{each}}'.



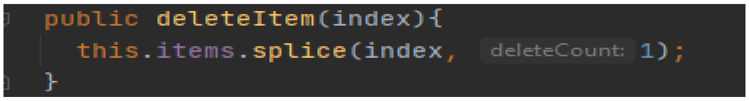
```
<div class="col">
  <div class="task3">{{each}}</div>
  <input (click)="deleteItem(index)" type="button" value="  DELETE  " class="task2">
  <input type="checkbox" class="box1">
</div>
```

8. Added one more ‘input’ type button to delete the item from ToDo list. When we clicked on delete button the respective item will be deleted from the list.



```
<input (click)="deleteItem(index)" type="button" value="  DELETE  " class="task2">
```

9. Implemented the definition to delete the item and added the same in component class of typescript file. Used ‘splice()’ to delete the item as below.

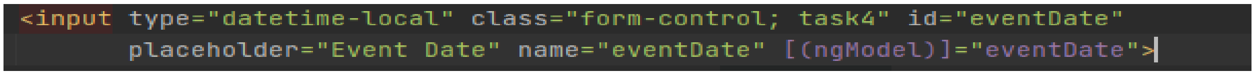


```
public deleteItem(index){
  this.items.splice(index, deleteCount: 1);
}
```

2.Countdown Timer:

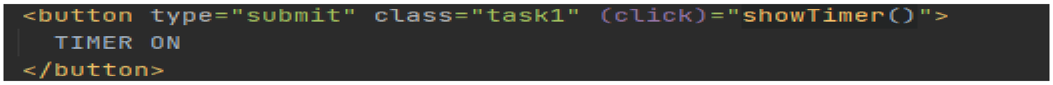
- We developed a countdown timer application using Angular and its elements. The main purpose of this application is to display the months, days, hours, minutes, and seconds of a user-entered event in a continuously decrementing way.
- We used HTML, CSS, & TypeScript and followed the below steps for implementation.

1. Used different html tags for developing this web application like <div>, <p> and so on.
2. Used <input type="datetime-local"> which is used to pick the date & time. It displays year, month, day & time on the web page. Also, we used ‘ngModel’ to bind the ‘eventDate’ (selected item deadline) to the ‘input’ element.



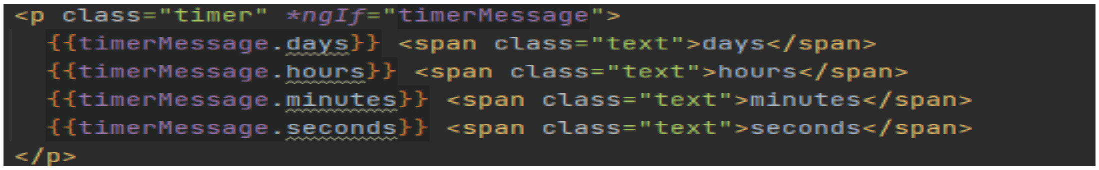
```
<input type="datetime-local" class="form-control; task4" id="eventDate"
       placeholder="Event Date" name="eventDate" [(ngModel)]="eventDate">
```

3. Added a button with a click event and it is assigned to a function ‘showTimer()’. When we clicked on this button, the function ‘showTimer()’ will be called.



```
<button type="submit" class="task1" (click)="showTimer()">
  TIMER ON
</button>
```

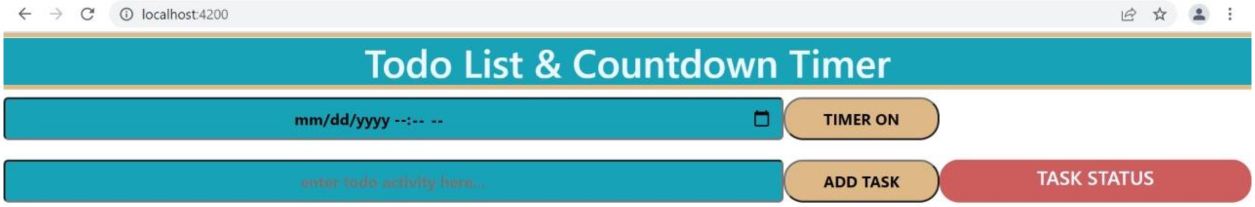
4. Implemented the definition of this function in component class of TypeScript file.
5. Using getTime() method, stored the current date & time into one variable and also stored the selected item time into another variable. Finally, we found the difference between both countdowndate and current date and stored the result into a one variable.
6. Added the functionality that timer will be updated with a time interval of one second.
7. In HTML file, *ngFor used to display the specific time for that task on the webpage and used interpolation to print the Timer data in days, hours, minutes & seconds.



```
<p class="timer" *ngIf="timerMessage">
  {{timerMessage.days}} <span class="text">days</span>
  {{timerMessage.hours}} <span class="text">hours</span>
  {{timerMessage.minutes}} <span class="text">minutes</span>
  {{timerMessage.seconds}} <span class="text">seconds</span>
</p>
```

Output Pages for all the Tasks:

Add the List of tasks and the Start the timer:

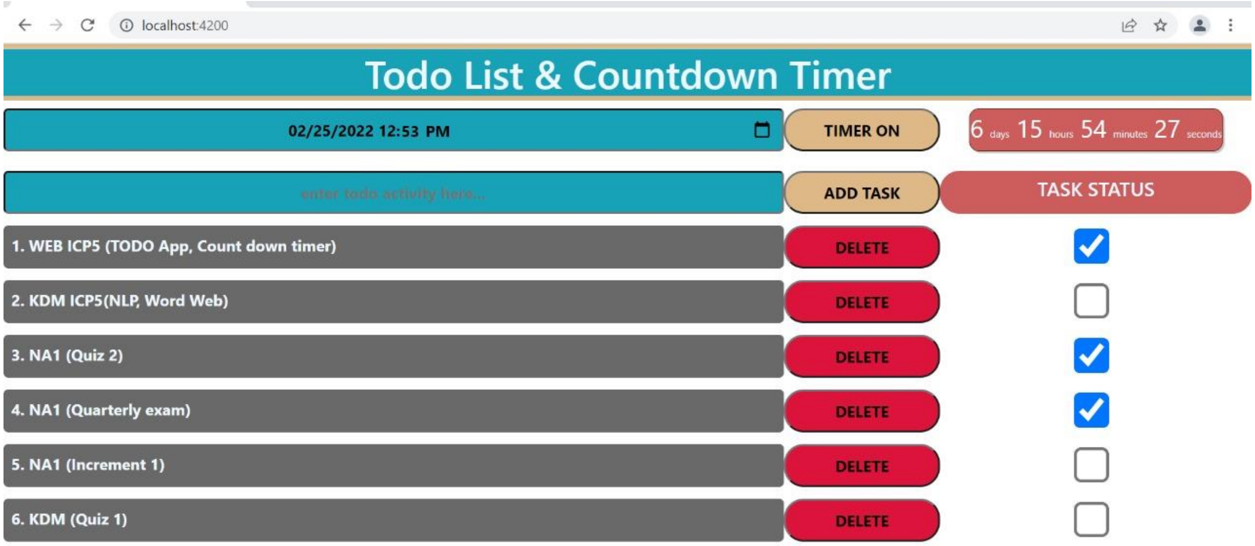


Todo List & Countdown Timer

mm/dd/yyyy --:-- --

enter todo activity here...

Output showing Timer and ToDo list:



Todo List & Countdown Timer

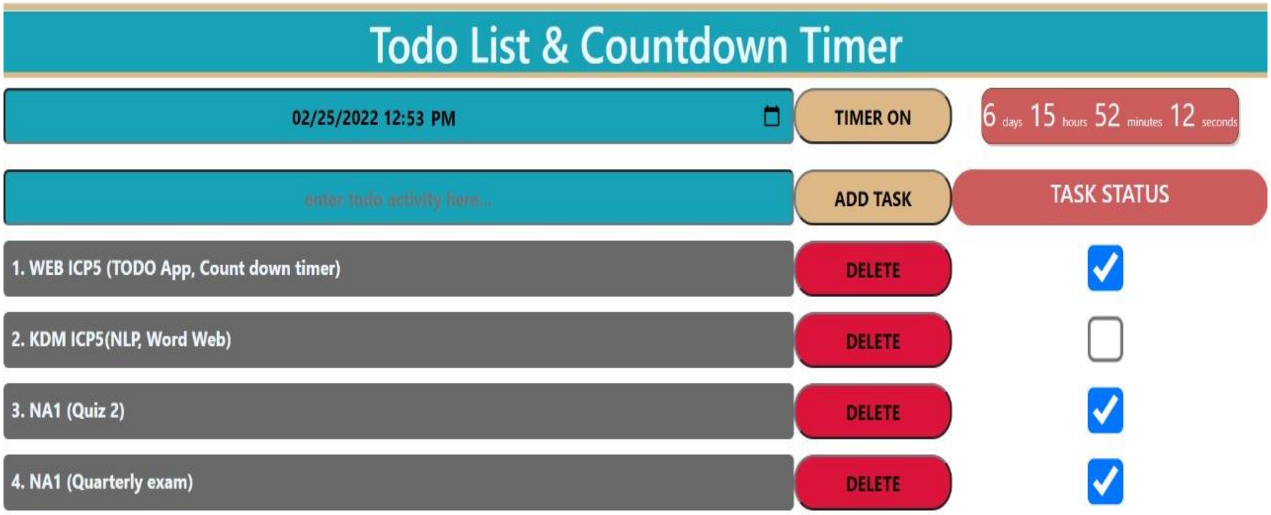
02/25/2022 12:53 PM

6 days 15 hours 54 minutes 27 seconds

enter todo activity here...

1. WEB ICP5 (TODO App, Count down timer) ☑
2. KDM ICP5(NLP, Word Web) □
3. NA1 (Quiz 2) ☑
4. NA1 (Quarterly exam) ☑
5. NA1 (Increment 1) □
6. KDM (Quiz 1) □

Deleted Tasks output:



Todo List & Countdown Timer

02/25/2022 12:53 PM

6 days 15 hours 52 minutes 12 seconds

enter todo activity here...

1. WEB ICP5 (TODO App, Count down timer) ☑
2. KDM ICP5(NLP, Word Web) □
3. NA1 (Quiz 2) ☑
4. NA1 (Quarterly exam) ☑

Contribution:

We have contributed equally.

Conclusion:

In this ICP, we have learned Angular and its elements and developed a web application using the same.

Challenges:

We have not faced any major challenges while doing the assignment.