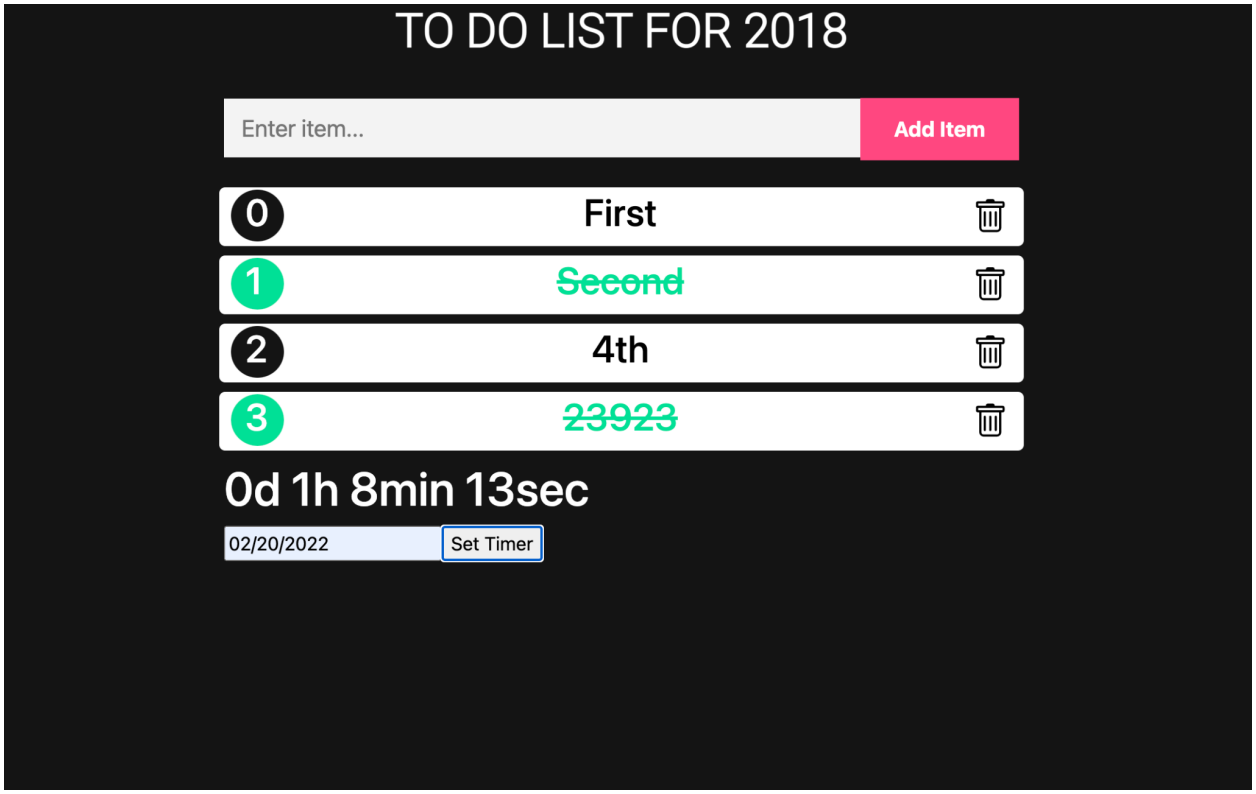
Repo: https://github.com/Irvin-Solis/CSEE-5590/tree/main/Source/Lesson5

In ICP 5, we learned about the Angular framework and Typescript. For the first part of the ICP, we were to create a TODO list using the angular framework. To tackle this task, I created a bit of HTML to form how an Item would look on the UI. I accomplished this using bootstrap and CSS. I then used typescript to show the functionality of an Item. I created an Item class and exported that into my main app.component.ts, That way I could create Items and delete Items. I then used some angular to create a function to iterate thru my items and update the HTML with the required tags. I then tackled the completing and Item and deleting an Item. With completing, the Item class holds a bool to represent if it's completed. I created a function to check on the status of the bool and CSS to change depending on the status of it. I also added an onclick function to change the status of the bool, and in doing so update the UI with it. With Deleting, I filter the Item array using the index of the Item and delete the corresponding item. On Adding an Item I created an Input Form to let users name the Item and add it to the list. I Just created another Item Object and added it to the array. The UI would update accordingly.
For the 2nd Part, creating a timer was a bit easier. I created another component and named it timer. With the functionality, I created a Date Object for the date the user inputs and a date object to represent the time right now. I then did some math to figure out the difference and set an interval function to update the difference as time goes on. A variable is updates as well and that is what is being presented.


TO DO LIST FOR 2018
Enter item...
Add Item
0
First
1
Second
2
4th
3
23923
0d 1h 8min 13sec
02/20/2022
Set Timer

```
  // define list of items
  title = 'TO DO LIST FOR 2018'
  inputItem = ""
  items: Item[];

  constructor() { }

  ngOnInit() {
    this.items = [
      {
        content: 'First',
        completed: false
      },
      {
        content: 'Second',
        completed: true
      },
    ]
  }

  // Write code to push new item
  submitNewItem() {
    this.items.push({
      content: this.inputItem,
      completed: false
    });

    this.inputItem =""
  }

  // Write code to complete item
  completeItem(id) {
    this.items.map((v, i) =>{
      if (i == id) v.completed = !v.completed;

      return v;
    })
  }

  // Write code to delete item
  deleteItem(id) {
    this.items = this.items.filter((v, i) => i != id);
  }

```

```
    <div class="main-title">
      <h1>{{title}}</h1>
    </div>
    <div class="box">
      <div class="container">
        <div class="row">
          <div class="col-12">
            <form (submit)="submitNewItem()">
              <input
                type="text"
                name="input"
                placeholder="Enter item..."
                class="item-input"
                [(ngModel)]="inputItem"
                />
                <input type="submit" value="Add Item" class="item-submit" >
            </form>
          </div>
        </div>
        <div class="items" *ngFor="let item of items; let i = index;">
          <div class="row rectangle">
            <div class="col-1 number circle {{(item.completed ? 'c-done' : '')}}">
              <h2>{{ i }}</h2>
            </div>
            <div class="col-4 title {{(item.completed ? 'done' : '')}}" (click)="completeItem(i)">
              <h2>{{ item.content }}</h2>
            </div>
            <div class="col-1 delete" (click)="deleteItem(i)">
              <svg xmlns="http://www.w3.org/2000/svg" width="30" height="50" fill="black" class="bi bi-trash" viewBox="0 0 16 16">
                <path d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5zm2.5 0a.5.5 0 0 1 .5.5v6a.5.5 0 0
                <path fill-rule="evenodd" d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1
              </svg>
            </div>
          </div>
        </div>
        <app-timer></app-timer>
      </div>
    </div>
  </body>
```