Ex. No: 4	Angular based App creation
09.08.2023	

Aim:

To Create an App using ANGULAR with Components, Binding, and Services usage.

Algorithm:

1. Setup angular using the ng serve command
2. Create all the required components.
3. Organize the app structure.
4. Implement the services that are needed.
5. Define component HTML templates with data binding to display dynamic content
6. Enable component communication using input/output properties and event binding.
7. Apply CSS styles to components, optimize for performance, and deploy the app.

Program:

Component code:

```
import { Component, Input, Output, EventEmitter } from "@angular/core";
import { Item } from "../item";


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

export class ItemComponent {

  editable = false;

  @Input() item!: Item;
  @Output() remove = new EventEmitter<Item>();

  saveItem(description: string) {
    if (!description) return;
    this.editable = false;
    this.item.description = description;
  }
}
```

Service code:

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

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

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

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

app.component.ts:

```
import { Component } from "@angular/core";
import { Item } from "./item";

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

export class AppComponent {
  title = "todo";
  filter: "all" | "done" = "all";
  allItems = [
    { description: "eat", done: true },
    { description: "sleep", done: false },
    { description: "play", done: false },
    { description: "laugh", done: false },
  ];
  get items() {
    if (this.filter === "all") {
      return this.allItems;
    }
    return this.allItems.filter((item) =>
      this.filter === "done" ? item.done : !item.done
```

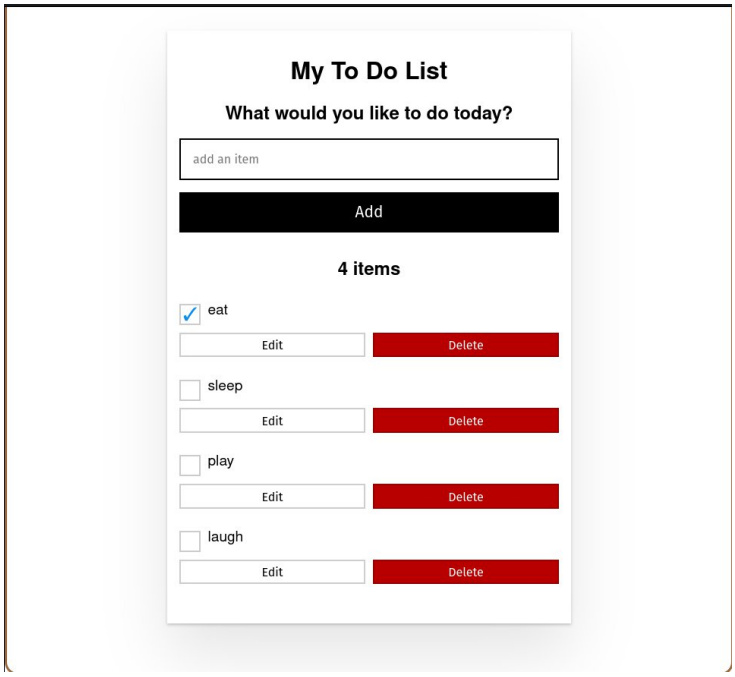
```
    );
  }
  addItem(description: string) {
    this.allItems.unshift({
      description,
      done: false
    });
  }
  remove(item: Item) {
    this.allItems.splice(this.allItems.indexOf(item), 1);
  }
}
```

Component html code:

```
<div class="main">
  <h1>My To Do List</h1>
  <label for="addItemInput">What would you like to do today?</label>

  <input
   #newItem
   placeholder="add an item"
   (keyup.enter)="addItem(newItem.value); newItem.value = ''"
   class="lg-text-input"
    id="addItemInput" />

  <button class="btn-primary" (click)="addItem(newItem.value)">Add</button>


  <h2>
    {{items.length}}
    <span *ngIf="items.length === 1; else elseBlock">item</span>
    <ng-template #elseBlock>items</ng-template>
  </h2>

  <ul>
    <li *ngFor="let i of items">
      <app-item (remove)="remove(i)" [item]="i"></app-item>
    </li>
  </ul>

</div>
```

Output:

Github Link: https://github.com/AsHtrich/Web_tech2023



Result:
Therefore, we've successfully created a simple react app.