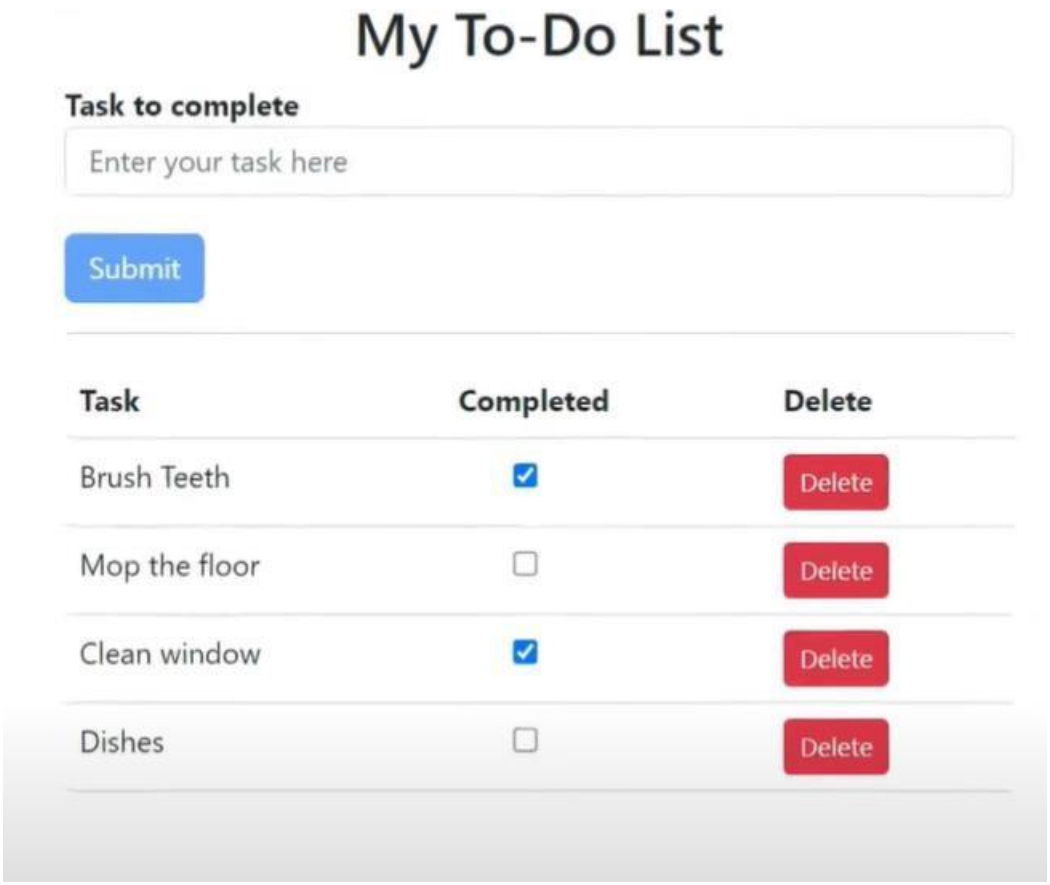
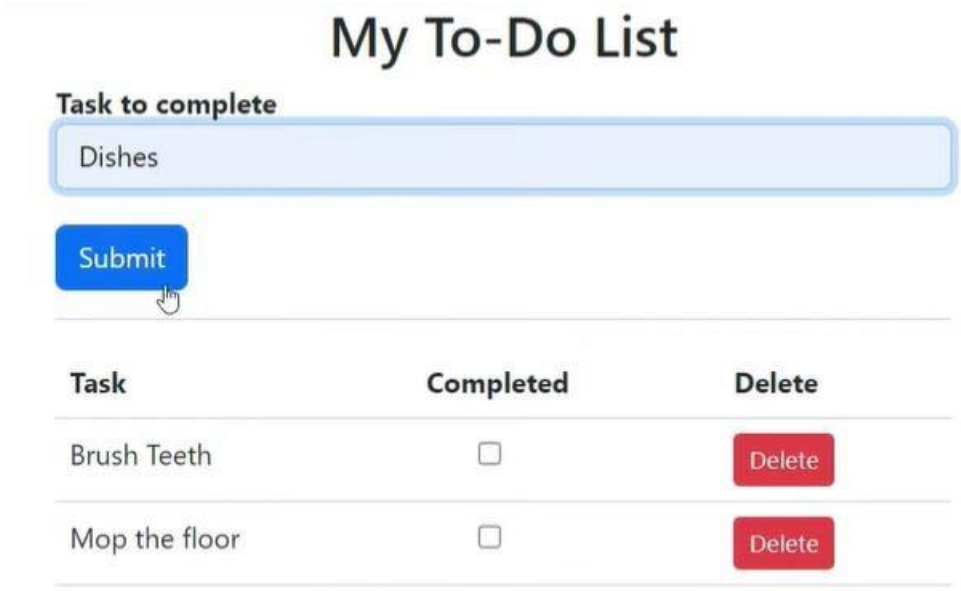
WEBTECH LAB EXERCISE 4

Adhyuth Charan

21011102008

Output Screenshots:





Source Code:

Index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Todo</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
</head>
<body>
  <app-root></app-root>
</body>
</html>
```

main.ts

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';
```

```
if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

app.component.spec.tss

```
import { TestBed } from '@angular/core/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [
        AppComponent
      ],
    }).compileComponents();
  });

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'todo'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('todo');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content
span')?.textContent).toContain('todo app is running!');
  });
});
```

app.component.tss

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

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

app.module.tss

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

import { AppComponent } from './app.component';
import { TodolistComponent } from './todolist/todolist.component';

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

todolist.component.css

```
h1 {
    text-align: center;
```

```
    font-size: 2rem;
}

#submitButton {
    margin-top: 20px;
}

#myCheckbox {
    margin-left: 30px;
}
```

todolist.component.html

```
<div class="container">

    <h1>To-Do Application</h1>

    <hr>
    <form (ngSubmit)="onSubmit(taskForm)" #taskForm="ngForm">
        <div class="form-group">
            <label for="task">Task</label>
            <input type="text" class="form-control" id="task"
placeholder="Enter task" ngModel name="task" required>
            <small *ngIf="taskForm.invalid && taskForm.dirty"
id="errorMessage" class="form-text text-danger">Required
                field</small>
        </div>

        <button [disabled]="taskForm.invalid" id="submitButton" type="submit"
class="btn btn-primary">Submit</button>
    </form>

    <hr>

    <table class="table">
        <thead>
            <tr>
                <th scope="col">Task</th>
                <th scope="col">Completed</th>
                <th scope="col">Delete</th>
            </tr>
        </thead>
        <tbody>
            <tr *ngFor="let t of taskArray; index as i">
                <td>{{t.taskName}}</td>
                <td><input (change)="onCheck(i)" id="myCheckbox"
type="checkbox"></td>
```

```
                <td><button class="btn btn-danger btn-sm"
(click)="onDelete(i)">Delete</button></td>
            </tr>

        </tbody>
    </table>
</div>
```

Todolist.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { TodolistComponent } from './todolist.component';

describe('TodolistComponent', () => {
  let component: TodolistComponent;
  let fixture: ComponentFixture<TodolistComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ TodolistComponent ]
    })
    .compileComponents();

    fixture = TestBed.createComponent(TodolistComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

todolist.component.ts

```
import { Component, OnInit } from '@angular/core';
import { NgForm } from '@angular/forms';

@Component({
  selector: 'app-todolist',
  templateUrl: './todolist.component.html',
```

```
  styleUrls: ['./todolist.component.css']
})
export class TodolistComponent implements OnInit {
  taskArray = [{ taskName: 'Brush teeth', isCompleted: false }];

  constructor() { }

  ngOnInit(): void {
  }

  onSubmit(form: NgForm) {
    console.log(form);

    this.taskArray.push({
      taskName: form.controls['task'].value,
      isCompleted: false
    })

    form.reset();
  }

  onDelete(index: number) {
    console.log(index);

    this.taskArray.splice(index, 1);
  }

  onCheck(index: number) {
    console.log(this.taskArray);

    this.taskArray[index].isCompleted = !this.taskArray[index].isCompleted;
  }

}
```