1. Create A Tooltip Directive

Solution:

1. Drag Drop Directive

Solution:

***Directive Code***

import { Directive, ElementRef, Renderer2, HostListener } from '@angular/core';

@Directive({

selector: '[appDraggable]'

})

export class DraggableDirective {

private isDragging = false;

private startX = 0;

private startY = 0;

private offsetX = 0;

private offsetY = 0;

constructor(private el: ElementRef, private renderer: Renderer2) {

this.renderer.setStyle(this.el.nativeElement, 'position', 'absolute');

this.renderer.setStyle(this.el.nativeElement, 'cursor', 'grab');

}

@HostListener('mousedown', ['$event'])

onMouseDown(event: MouseEvent): void {

this.isDragging = true;

this.startX = event.clientX;

this.startY = event.clientY;

this.offsetX = this.el.nativeElement.offsetLeft;

this.offsetY = this.el.nativeElement.offsetTop;

this.renderer.setStyle(this.el.nativeElement, 'cursor', 'grabbing');

event.preventDefault();

}

@HostListener('mousemove', ['$event'])

onMouseMove(event: MouseEvent): void {

if (!this.isDragging) return;

const moveX = event.clientX - this.startX;

const moveY = event.clientY - this.startY;

const newLeft = this.offsetX + moveX;

const newTop = this.offsetY + moveY;

this.renderer.setStyle(this.el.nativeElement, 'left', `${newLeft}px`);

this.renderer.setStyle(this.el.nativeElement, 'top', `${newTop}px`);

}

@HostListener('mouseup')

@HostListener('mouseleave')

onMouseUp(): void {

this.isDragging = false;

this.renderer.setStyle(this.el.nativeElement, 'cursor', 'grab');

}

}

***app.module***

@NgModule({

declarations: [DraggableDirective],

exports: [DraggableDirective],

})

export class SharedModule {}

***HTML***

<div appDraggable

style="width: 100px; height: 100px; background-color: lightblue;">

Drag Me!

</div>