

Nama : Faris Adlan Aminudin (1731710041)

Source Code : https://github.com/farisdX/FarisAdlanA-1731710041-Project_Four

```
recipe.service.ts × shopping-list.service.ts × recipe-detail.component.html × recipe-detail.component.ts ×
1 import { Injectable, EventEmitter } from '@angular/core';
2
3 @Injectable({providedIn: 'root'})
4 export class RecipeService {
5
6   recipeSelected = new EventEmitter<Recipe>();
7
8   recipes: Recipe[] = [
9
10     new Recipe( {name: 'Burger', desc: 'Whataburger is an American privately held regional fast food restaurant chain, based in San Antonio, Texas.'},
11       new Ingredient( {name: 'meal', amount: 1},
12         new Ingredient( {name: 'tomato', amount: 2}
13       )
14     ),
15
16     new Recipe( {name: 'Dream meal', desc: 'At Dream Dinners you make homemade meals for your family in our store, then freeze, thaw and cook.'},
17       new Ingredient( {name: 'rice', amount: 1},
18         new Ingredient( {name: 'meal', amount: 3},
19         new Ingredient( {name: 'egg', amount: 1}
20       )
21     )
22   );
23
24   getRecipe(){
25     return this.recipes.slice();
26   }
27
28   addIngredientsShoppingList(ingredients: Ingredient[]){
29     this.slsService.addIngredients(ingredients);
30   }
31
32   constructor( private slsService: ShoppingListService ) { }
33
34 }
```

```
recipe-list.component.ts × recipe.service.ts × shopping-list.service.ts × recipe-detail.component.html ×
1 import { Component, OnInit, EventEmitter, Output } from '@angular/core';
2 import { Recipe } from '../recipe.model';
3 import { RecipeService } from '../recipe.service';
4
5 @Component({
6   selector: 'app-recipe-list',
7   templateUrl: './recipe-list.component.html',
8   styleUrls: ['./recipe-list.component.css']
9 })
10 export class RecipeListComponent implements OnInit {
11
12   // @Output() recipeWasSelected = new EventEmitter<Recipe>();
13
14   recipes: Recipe[];
15
16   constructor(private recipeService: RecipeService) { }
17
18
19   ngOnInit() {
20     this.recipes = this.recipeService.getRecipe();
21   }
22
23   // onRecipeSelected(recipe: Recipe) {
24   //   this.recipeWasSelected.emit(recipe);
25   // }
26
27 }
28 }
```

```
ce.ts × recipe-detail.component.html × recipe-detail.component.ts × shopping-list.component.ts × recipe.comp

1  import { Component, OnInit } from '@angular/core';
2  import { Recipe } from '../recipe.model';
3  import { RecipeService } from '../recipe.service';
4  import { Injectable } from '@angular/core';
5
6  @Component({
7    selector: 'app-recipe',
8    templateUrl: './recipe.component.html',
9    styleUrls: ['./recipe.component.css'],
10   providers:[RecipeService]
11 })
12 export class RecipeComponent implements OnInit {
13   selectedRecipe: Recipe;
14
15   constructor(private recipeService: RecipeService) { }
16
17   ngOnInit() {
18     this.recipeService.recipeSelected
19       .subscribe(
20         generatorOrNext: (recipe:Recipe)=>{
21           this.selectedRecipe= recipe;
22         }
23       )
24   }
25 }
26
27
28
```

```
g-list.component.ts × recipe.component.ts × recipe-list.component.spec.ts × recipe.model.ts × recipe-item.component.ts ×

1  import ...
4
5  @Component({
6    selector: 'app-recipe-item',
7    templateUrl: './recipe-item.component.html',
8    styleUrls: ['./recipe-item.component.css']
9  })
10 export class RecipeItemComponent implements OnInit {
11
12   @Input() recipe: Recipe;
13
14   // @Output() recipeSelected = new EventEmitter<void>();
15
16   constructor(private recipeService:RecipeService) { }
17
18   ngOnInit() {
19
20   }
21
22   onSelect() {
23     // this.recipeSelected.emit();
24     this.recipeService.recipeSelected.emit(this.recipe);
25   }
26 }
27
```

```
component.ts x recipe-list.component.spec.ts x recipe.model.ts x recipe-item.component.ts x recipe-list.component.html x
1 <div class="row">
2   <div class="col-xs-12">
3     <button class="btn btn-success">Recipe</button>
4     <hr>
5   </div>
6   <div>
7   </div>
8 </div>
9 <div class="row">
10  <div class="col-xs-12">
11    <app-recipe-item
12      *ngFor="let recipeEl of recipes"
13      [recipe]="recipeEl">
14    </app-recipe-item>
15  </div>
16 </div>
```

```
component.spec.ts x recipe.model.ts x recipe-item.component.ts x recipe-list.component.html x recipe.component.html x
1 <div class="row">
2   <div class="col-md-5">
3     <app-recipe-list></app-recipe-list>
4   </div>
5   <div class="col-md-7">
6     <app-recipe-detail
7       *ngIf="selectedRecipe; else infotext"
8       [recipe]="selectedRecipe"
9     ></app-recipe-detail>
10    <ng-template #infotext>
11      <p> Please Select Recipe</p>
12    </ng-template>
13  </div>
14 </div>
15
```

```
shopping-list.service.ts × shopping-list.component.ts ×
1 import { Ingredient } from '../shared/ingredient.model';
2 import { Component, OnInit } from '@angular/core';
3 import { ShoppingListService } from '../shopping-list/shopping-list.service';
4
5 @Component({
6   selector: 'app-shopping-list',
7   templateUrl: './shopping-list.component.html',
8   styleUrls: ['./shopping-list.component.css']
9 })
10 export class ShoppingListComponent implements OnInit {
11
12   ingredients: Ingredient[]
13   // = [
14   //   new Ingredient('Apples', 20),
15   //   new Ingredient('tomato', 10),
16   // ];
17
18   constructor(private slsService: ShoppingListService) { }
19
20   ngOnInit() {
21     this.ingredients = this.slsService.getIngredient();
22     this.slsService.ingredientChange
23       .subscribe(
24         generatorOrNext: (Ingredient: Ingredient[]) => {
25           this.ingredients = Ingredient;
26         }
27       )
28   }
29   // onIngredientAdded(ingredient: Ingredient) {
30   //   this.ingredients.push(ingredient);
31   // }
32 }
33
```

```
shopping-list.service.ts × shopping-list.component.ts ×
1 import ...
2
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class ShoppingListService {
8   ingredientChange = new EventEmitter<Ingredient[]>();
9
10   private ingredients: Ingredient[]
11   = [
12     new Ingredient( name: 'Apples', amount: 20),
13     new Ingredient( name: 'tomato', amount: 10),
14   ];
15   constructor() { }
16
17   getIngredient(){
18     return this.ingredients.slice();
19   }
20
21   addIngredient(ingredient: Ingredient){
22     this.ingredients.push(ingredient);
23     this.ingredientChange.emit(this.ingredients.slice());
24   }
25
26   addIngredients(ingredients : Ingredient[]){
27     this.ingredients.push(...ingredients);
28     this.ingredientChange.emit(this.ingredients.slice())
29   }
30
31
32
33
```

```
shopping-list.service.ts × app.module.ts × shopping-list.component.ts ×
1  import ...
15
16  @NgModule({
17    declarations: [
18      AppComponent,
19      HeaderComponent,
20      ShoppingListComponent,
21      ShoppingEditComponent,
22      RecipeComponent,
23      RecipeListComponent,
24      RecipeDetailComponent,
25      RecipeItemComponent,
26      FooterComponent,
27      DropdownDirective
28    ],
29    imports: [
30      BrowserModule,
31      FormsModule
32    ],
33    providers: [ShoppingListService],
34    bootstrap: [
35      AppComponent
36    ]
37  })
38  export class AppModule { }
39
```

```
shopping-list.service.ts × app.module.ts × shopping-edit.component.ts × shopping-list.component.ts ×
1  import ...
4
5  @Component({
6    selector: 'app-shopping-edit',
7    templateUrl: './shopping-edit.component.html',
8    styleUrls: ['./shopping-edit.component.css']
9  })
10 export class ShoppingEditComponent implements OnInit {
11   @ViewChild('nameInput') nameRef: ElementRef;
12   @ViewChild('amountInput') amountRef: ElementRef;
13   // @Output() ingredientAdded = new EventEmitter<Ingredient>();
14   constructor(private slsService: ShoppingListService) { }
15
16   ngOnInit() {
17   }
18   addItem() {
19     const ingName = this.nameRef.nativeElement.value;
20     const ingAmount = this.amountRef.nativeElement.value;
21     const newIngredient = new Ingredient(ingName, ingAmount);
22     this.slsService.addIngredient(newIngredient);
23     // this.ingredientAdded.emit(newIngredient);
24   }
25
26 }
27
```

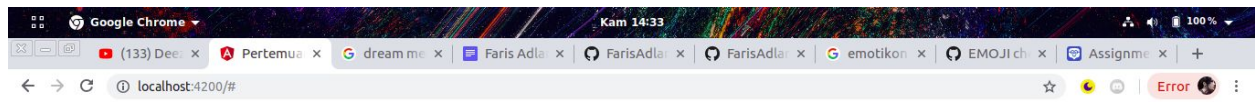


```
shopping-list.service.ts × app.module.ts × shopping-edit.component.ts × shopping-list.component.html ×
1 <div class="row">
2   <div class="col-xs-10">
3     <app-shopping-edit></app-shopping-edit>
4     <hr>
5     <ul class="list-group">
6       <a href="#" class="list-group-item"
7         style="..."
8         *ngFor="let Ingredient of ingredients">
9         {{Ingredient.name}} {{Ingredient.amount}}
10      </a>
11    </ul>
12  </div>
13 </div>
14
```

```
shopping-list.service.ts × app.module.ts × shopping-edit.component.ts × recipe-detail.component.html ×
17 <button class="btn btn-primary dropdown-toggle">
18   Manage Recipe<span class="caret"></span>
19 </button>
20 <ul class="dropdown-menu">
21   <li> <a (click)="onAddShoppingList()" style="cursor:pointer" > To Shopping</a></li>
22   <li> <a href="#"> Edit Recipe</a></li>
23   <li> <a href="#"> Delete Recipe</a></li>
24 </ul>
25 </div>
26 </div>
27 </div>
28 <div class="row">
29   <div class="col-xs-12">
30     {{recipe.description}}
31   </div>
32 </div>
33 <div class="row">
34   <div class="col-xs-12">
35     <!--Ingredient-->
36     <ul class="list-group-item">
37       <li
38         class="list-group-item"
39         *ngFor="let ingredient of recipe.ingredient">
40         {{ingredient.name}}-{{ingredient.amount}}
41       </li>
42     </ul>
43   </div>
44 </div>
```

```
t.service.ts × app.module.ts × shopping-edit.component.ts × recipe-detail.component.html × recipe-detail.component.ts ×
1 import { Component, OnInit, Input } from '@angular/core';
2 import { Recipe } from '../recipe.model';
3 import { RecipeService } from '../recipe.service';
4
5 @Component({
6   selector: 'app-recipe-detail',
7   templateUrl: './recipe-detail.component.html',
8   styleUrls: ['./recipe-detail.component.css']
9 })
10 export class RecipeDetailComponent implements OnInit {
11   @Input() recipe: Recipe;
12   constructor(private recipeService: RecipeService) { }
13
14   ngOnInit() {
15   }
16
17   onAddShoppingList(){
18     this.recipeService.addIngredientsShoppingList(this.recipe.ingredient);
19   }
20
21 }
22
```

Hasil :



Recipe Book

Recipes Shopping List Manage

Name	Amount
bread	3

Add Delete Clear

Apples(20)
tomato(10)
bread(3)
bread(3)
bread(3)

