launch_code

Codergirl – Frontend

Unit 2 - Class 4

December 9, 2020

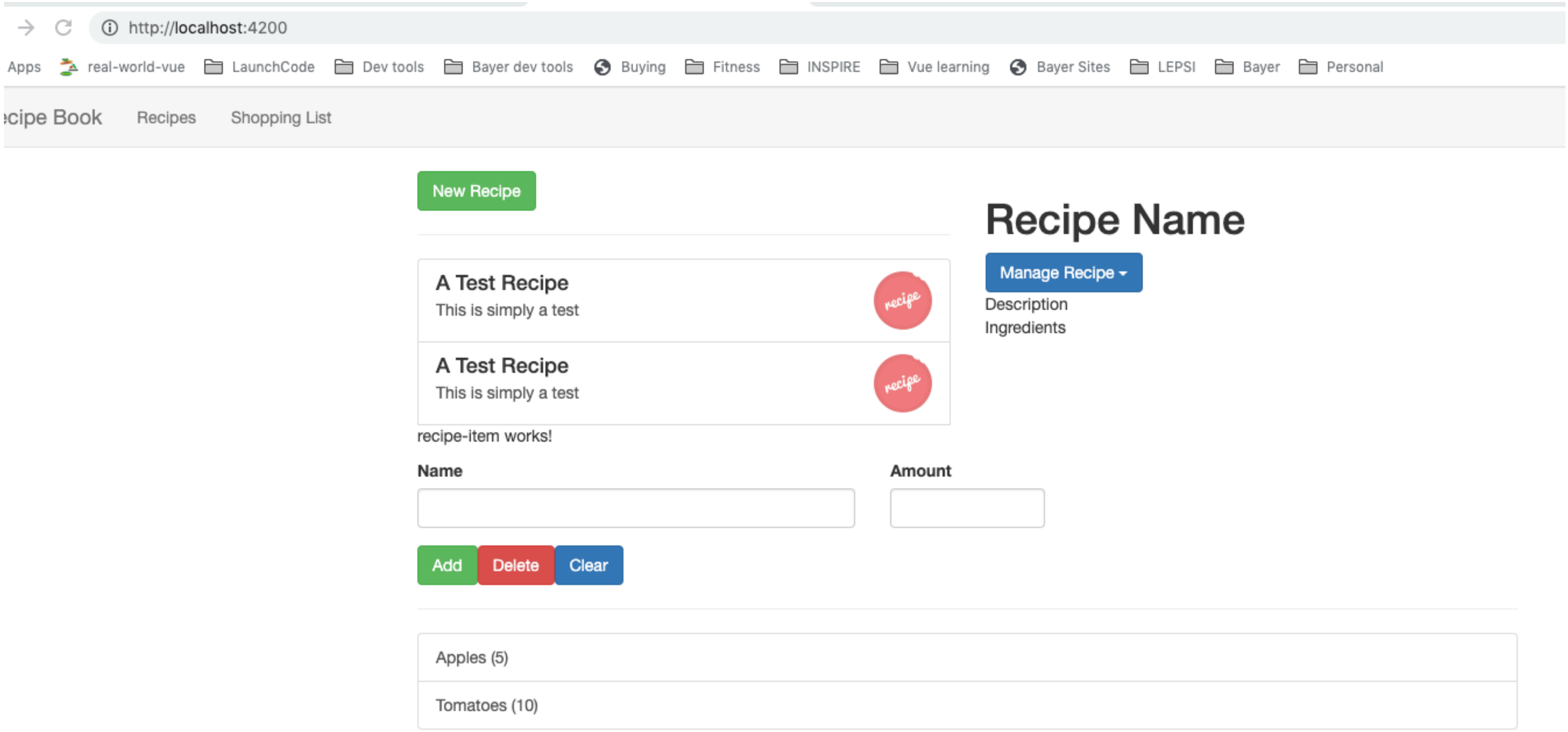
Agenda

Section 6

1. Adding Navigation with ngIf and Event Binding. (85)
2. Passing Recipe Data with Property Binding. (86)
3. Passing Data with Event and Property Binding (87)
4. Allowing the User to Add Ingredients to the Shopping List (89)

1. Adding Navigation with ngIf and Event Binding. (85)

app – we want to conditional logic to only load one of the two sections between recipes and shopping based on header clicked (emit event)



A Test Recipe
This is simply a test

A Test Recipe
This is simply a test

recipe-item works!

Recipe Name

Description
Ingredients

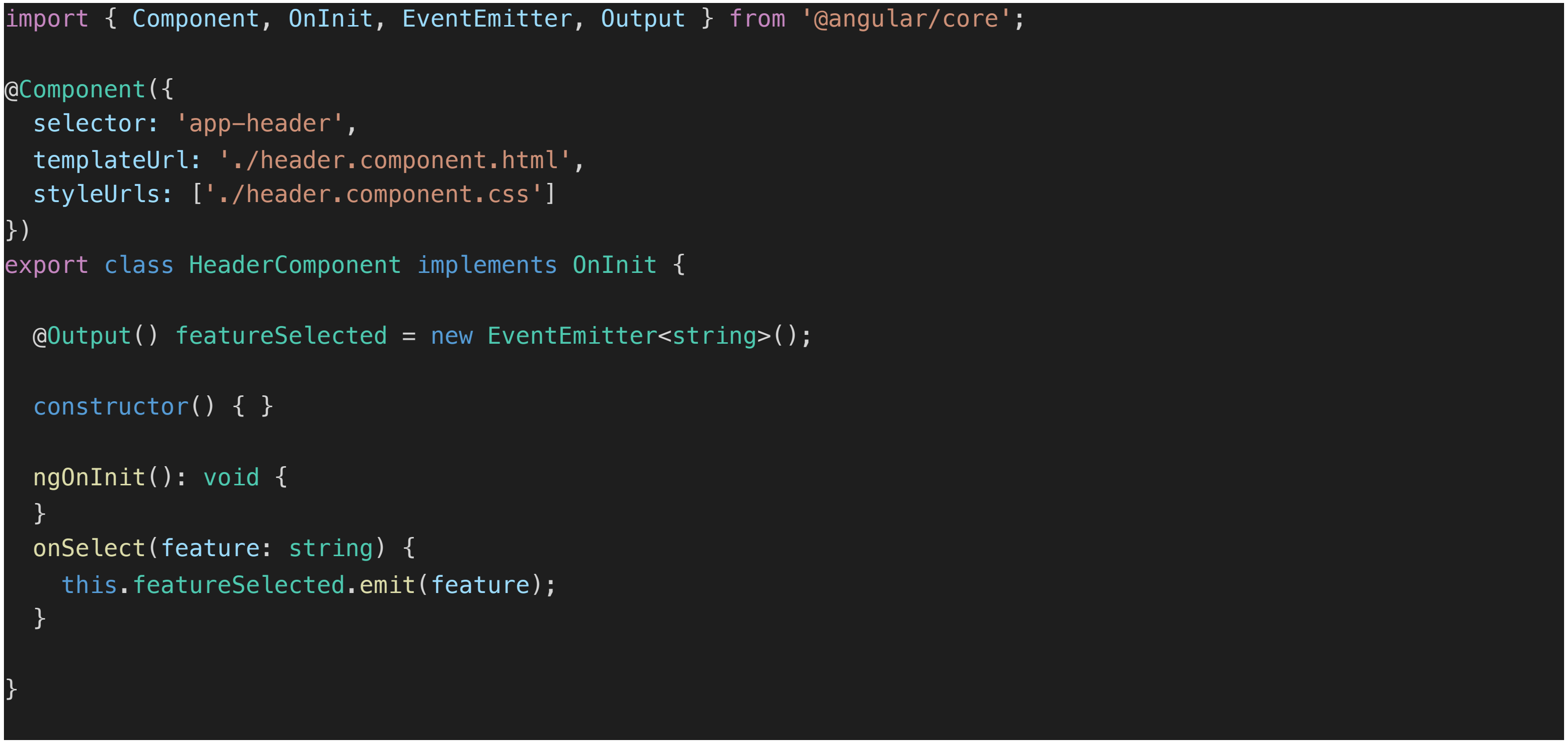
Apples (5)

Tomatoes (10)

1. Adding Navigation with ngIf and Event Binding. (85)

Breakout rooms to work on the solution – 15 mins

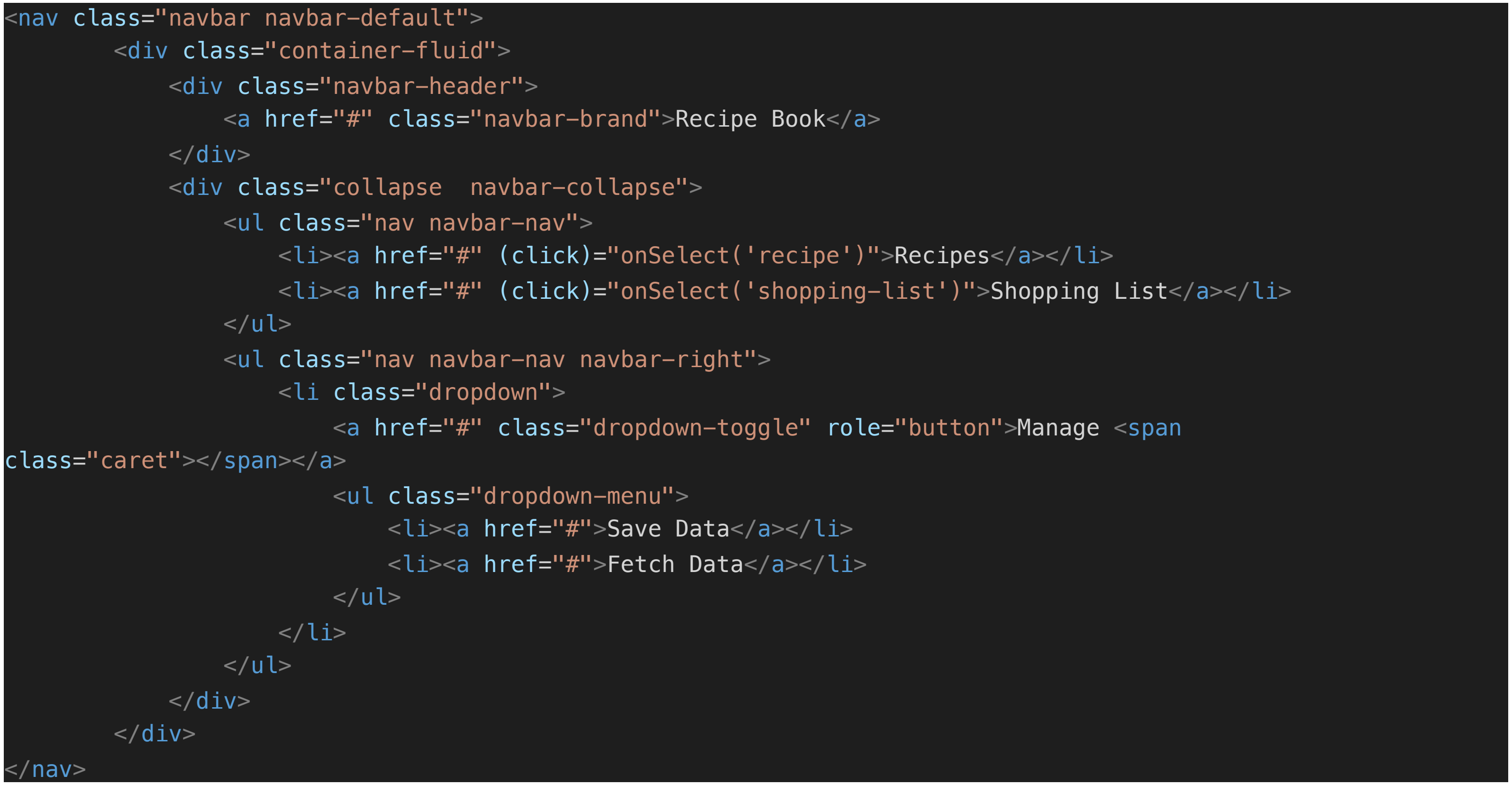
1. Adding Navigation with ngIf and Event Binding. (85)



```typescript
import { Component, OnInit, EventEmitter, Output } from '@angular/core';

@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.css']
})
export class HeaderComponent implements OnInit {

  @Output() featureSelected = new EventEmitter<string>();

  constructor() { }

  ngOnInit(): void {
  }
  onSelect(feature: string) {
    this.featureSelected.emit(feature);
  }

}
```

header-component.ts

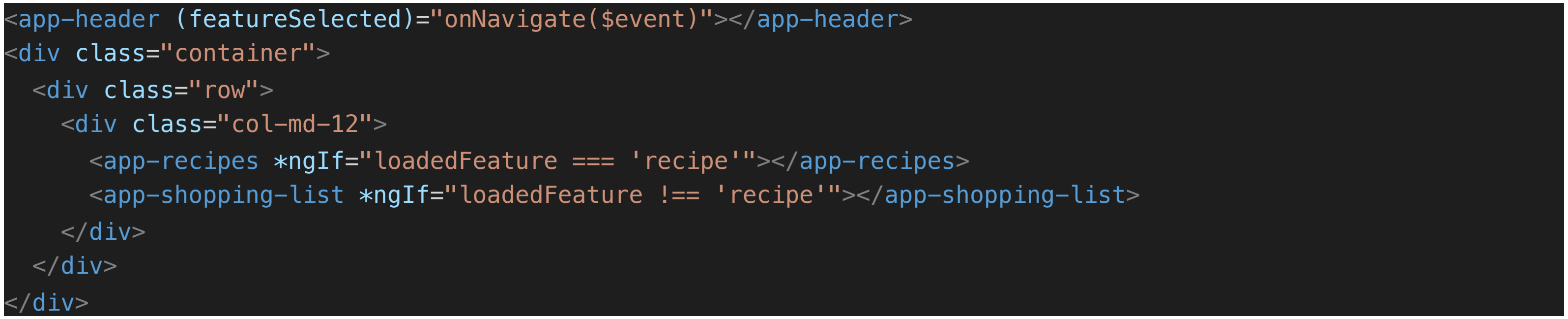
1. Adding Navigation with ngIf and Event Binding. (85)



```
<nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <a href="#" class="navbar-brand">Recipe Book</a>
            </div>
            <div class="collapse  navbar-collapse">
                <ul class="nav navbar-nav">
                    <li><a href="#" (click)="onSelect('recipe')">Recipes</a></li>
                    <li><a href="#" (click)="onSelect('shopping-list')">Shopping List</a></li>
                </ul>
                <ul class="nav navbar-nav navbar-right">
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle" role="button">Manage <span
class="caret"></span></a>
                        <ul class="dropdown-menu">
                            <li><a href="#">Save Data</a></li>
                            <li><a href="#">Fetch Data</a></li>
                        </ul>
                    </li>
                </ul>
            </div>
        </div>
</nav>
```

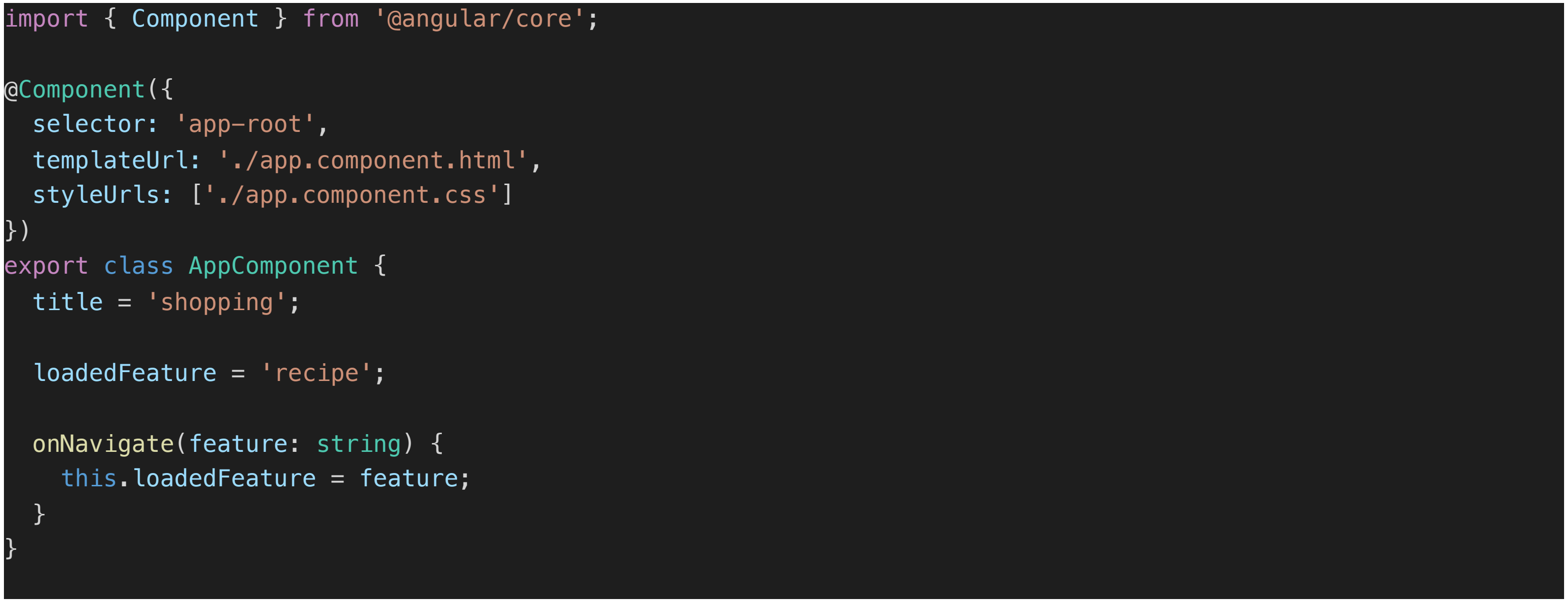
header-component.html

1. Adding Navigation with ngIf and Event Binding. (85)



```html
<app-header (featureSelected)="onNavigate($event)"></app-header>
<div class="container">
  <div class="row">
    <div class="col-md-12">
      <app-recipes *ngIf="loadedFeature === 'recipe'"></app-recipes>
      <app-shopping-list *ngIf="loadedFeature !== 'recipe'"></app-shopping-list>
    </div>
  </div>
</div>
```

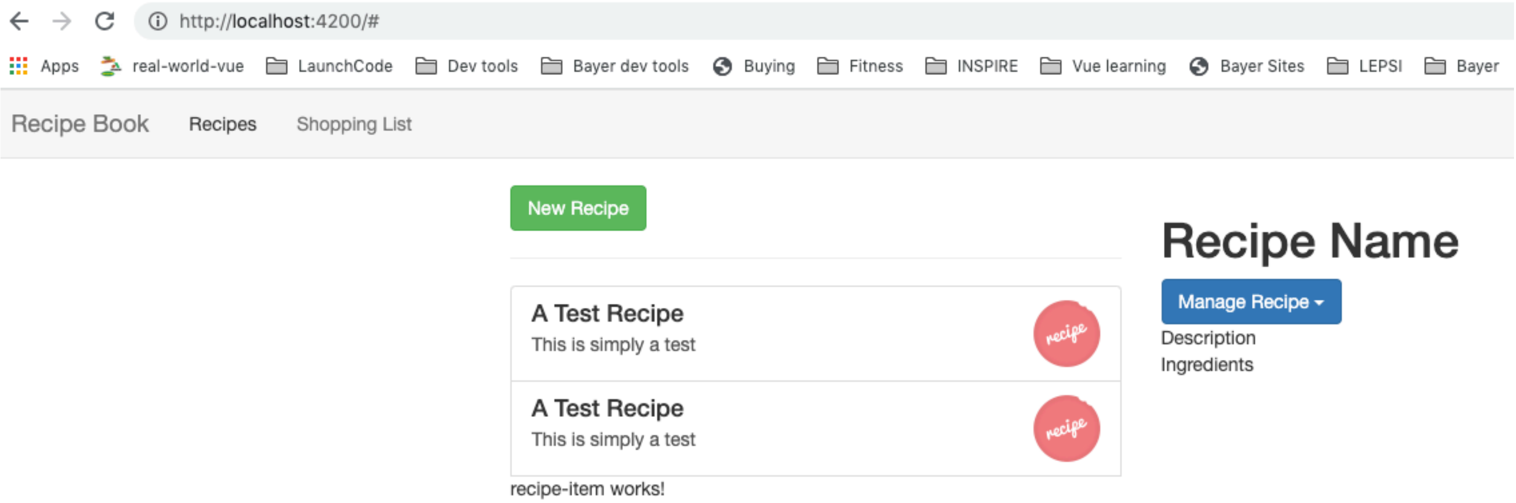
app-component.html



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

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

  loadedFeature = 'recipe';

  onNavigate(feature: string) {
    this.loadedFeature = feature;
  }
}
```

app-component.ts

1. Adding Navigation with ngIf and Event Binding. (85)



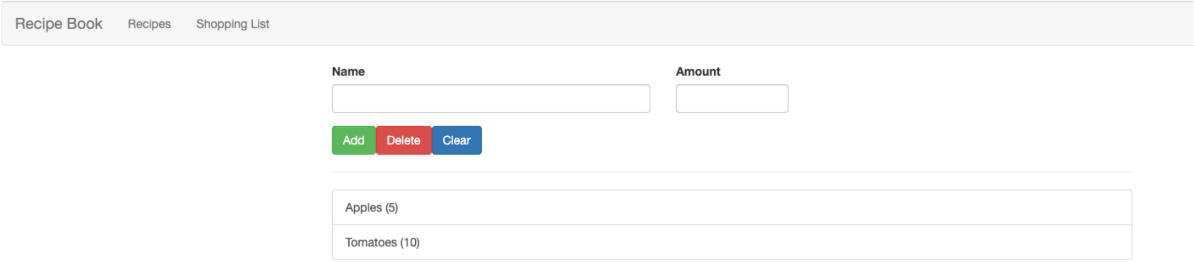
A Test Recipe
This is simply a test

A Test Recipe
This is simply a test

recipe-item works!

Recipe Name

Description
Ingredients



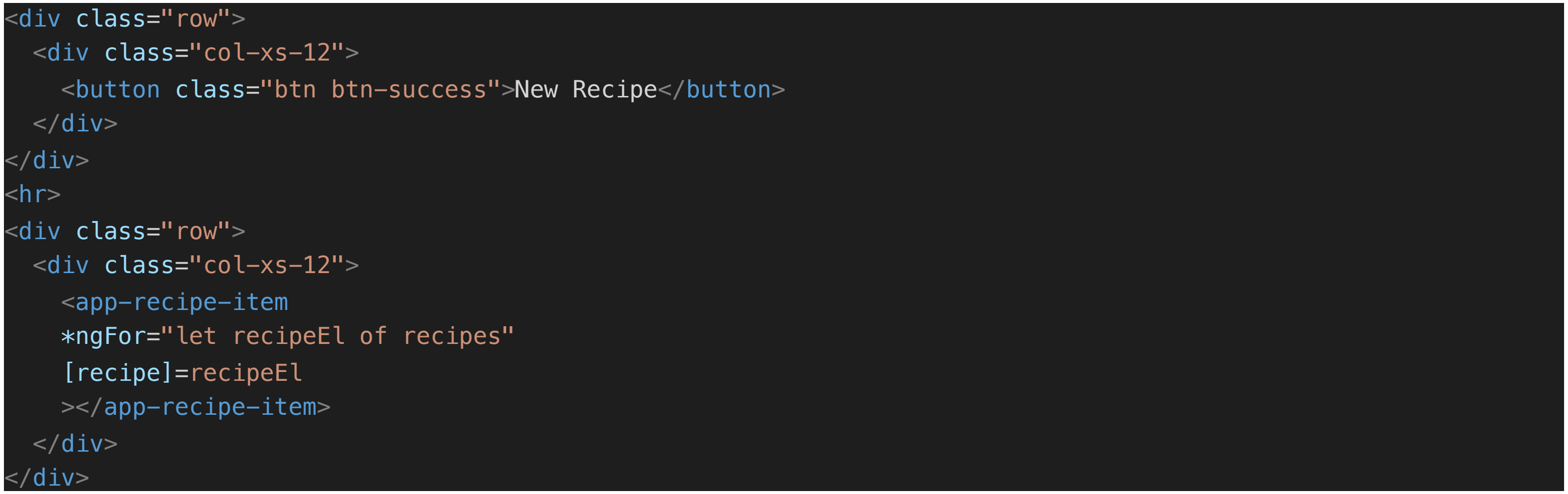
Apples (5)
Tomatoes (10)

2. Passing Recipe Data with Property Binding. (86)

app – we want to move for loop from recipe –item and move it to the recipe-list and pass down single recipe item to recipe-item component and make that work.

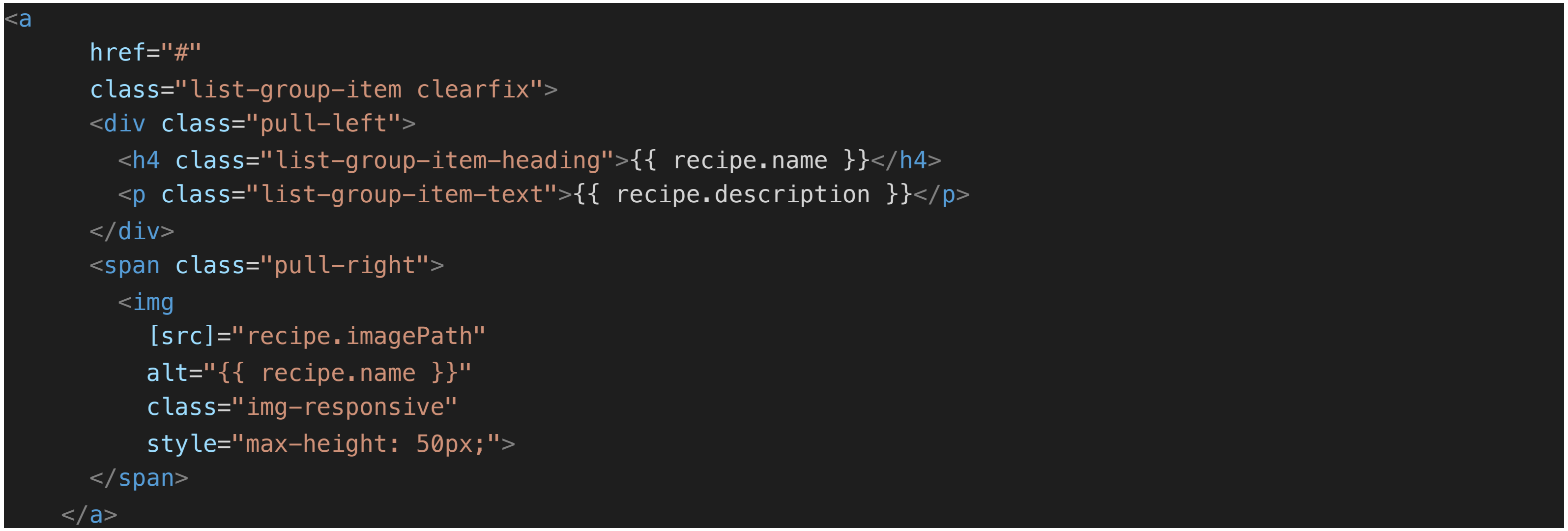
Breakout room – 15 mins

2. Passing Recipe Data with Property Binding. (86)



```
<div class="row">
  <div class="col-xs-12">
    <button class="btn btn-success">New Recipe</button>
  </div>
</div>
<hr>
<div class="row">
  <div class="col-xs-12">
    <app-recipe-item
    *ngFor="let recipeEl of recipes"
    [recipe]=recipeEl
    ></app-recipe-item>
  </div>
</div>
```

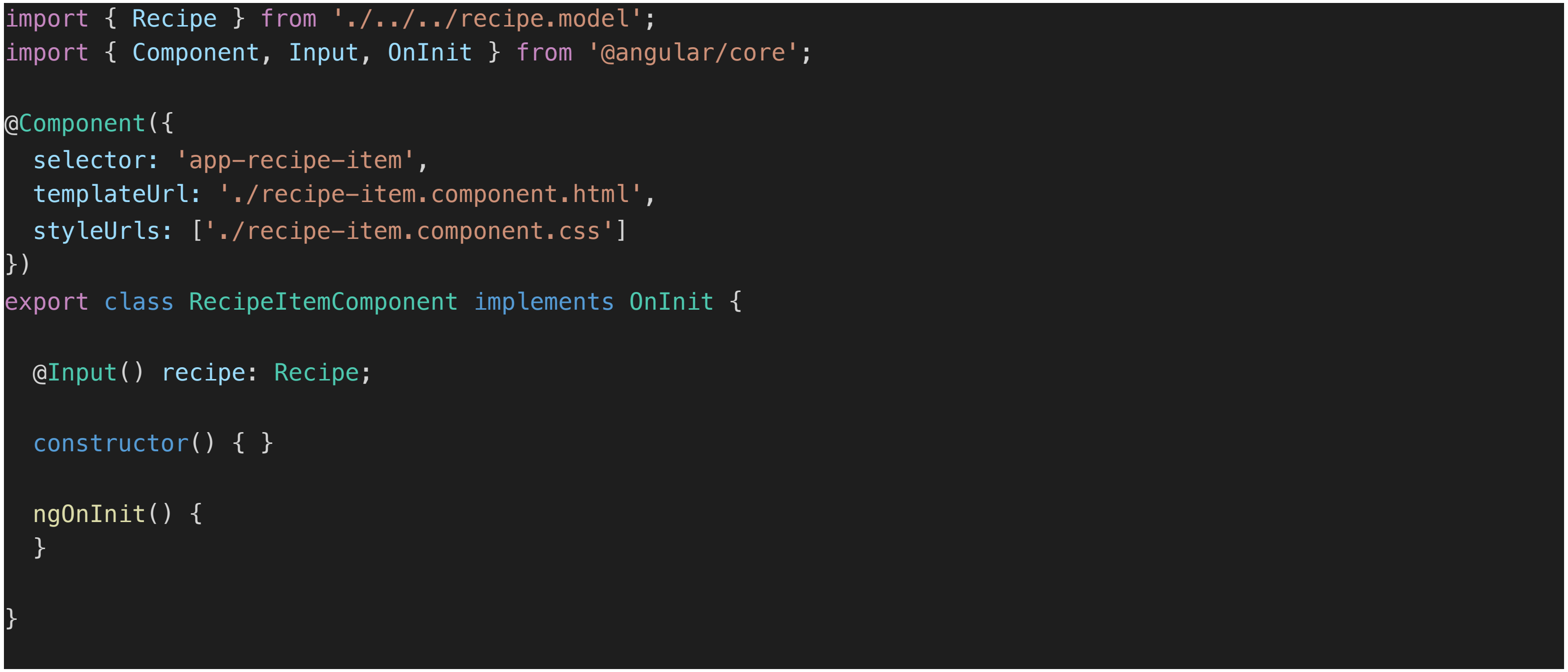
recipe-list.component.html



```
<a
      href="#"
      class="list-group-item clearfix">
      <div class="pull-left">
        <h4 class="list-group-item-heading">{{ recipe.name }}</h4>
        <p class="list-group-item-text">{{ recipe.description }}</p>
      </div>
      <span class="pull-right">
        <img
          [src]="recipe.imagePath"
          alt="{{ recipe.name }}"
          class="img-responsive"
          style="max-height: 50px;">
      </span>
    </a>
```

recipe-item.component.html

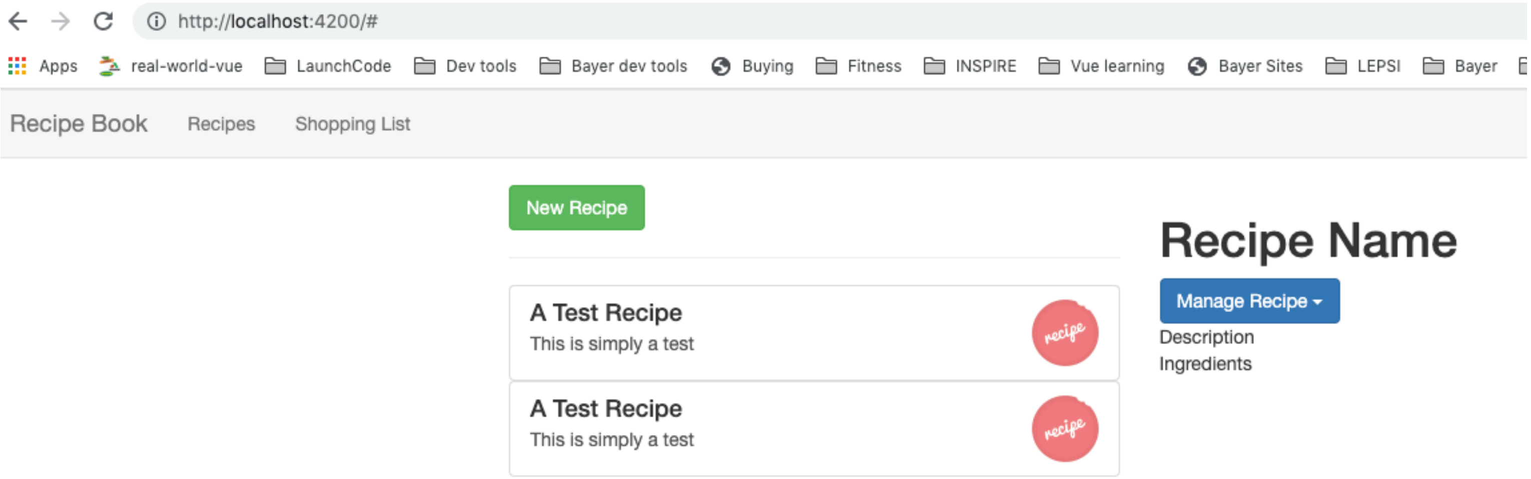
2. Passing Recipe Data with Property Binding. (86)



```
import { Recipe } from './../../recipe.model';
import { Component, Input, OnInit } from '@angular/core';

@Component({
  selector: 'app-recipe-item',
  templateUrl: './recipe-item.component.html',
  styleUrls: ['./recipe-item.component.css']
})
export class RecipeItemComponent implements OnInit {

  @Input() recipe: Recipe;

  constructor() { }

  ngOnInit() {
  }

}
```

recipe-item.component.ts

2. Passing Recipe Data with Property Binding. (86)



A Test Recipe
This is simply a test

A Test Recipe
This is simply a test

Recipe Name

Description
Ingredients

3. Passing Data with Event and Property Binding (87)

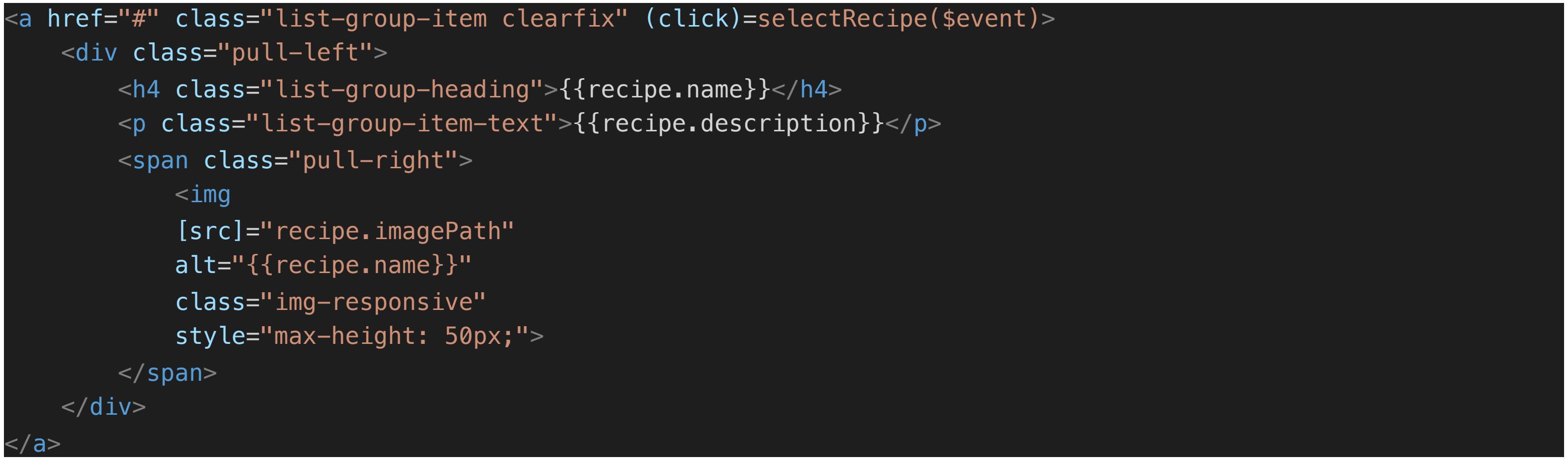
Goal: Click on each recipe item to load recipe details

Emit event on recipe item that it was clicked (click listener for recipe item), to get event to recipe detail to pass down which recipe was clicked to component

Property and Event binding.

Breakout room : 20 mins

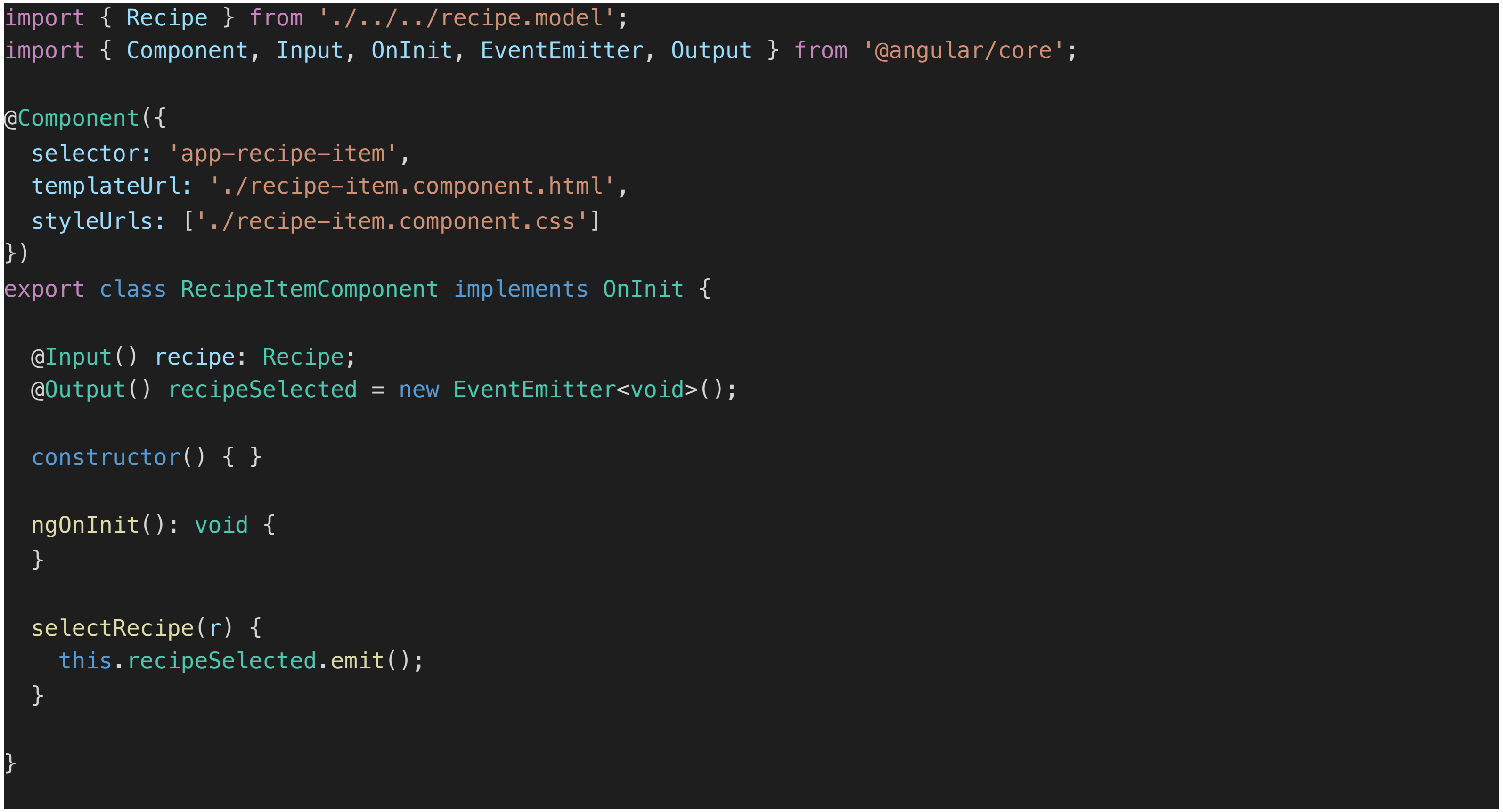
3. Passing Data with Event and Property Binding (87)



```
<a href="#" class="list-group-item clearfix" (click)=selectRecipe($event)>
    <div class="pull-left">
        <h4 class="list-group-heading">{{recipe.name}}</h4>
        <p class="list-group-item-text">{{recipe.description}}</p>
        <span class="pull-right">
            <img
            [src]="recipe.imagePath"
            alt="{{recipe.name}}"
            class="img-responsive"
            style="max-height: 50px;">
        </span>
    </div>
</a>
```

recipe-item.component.html

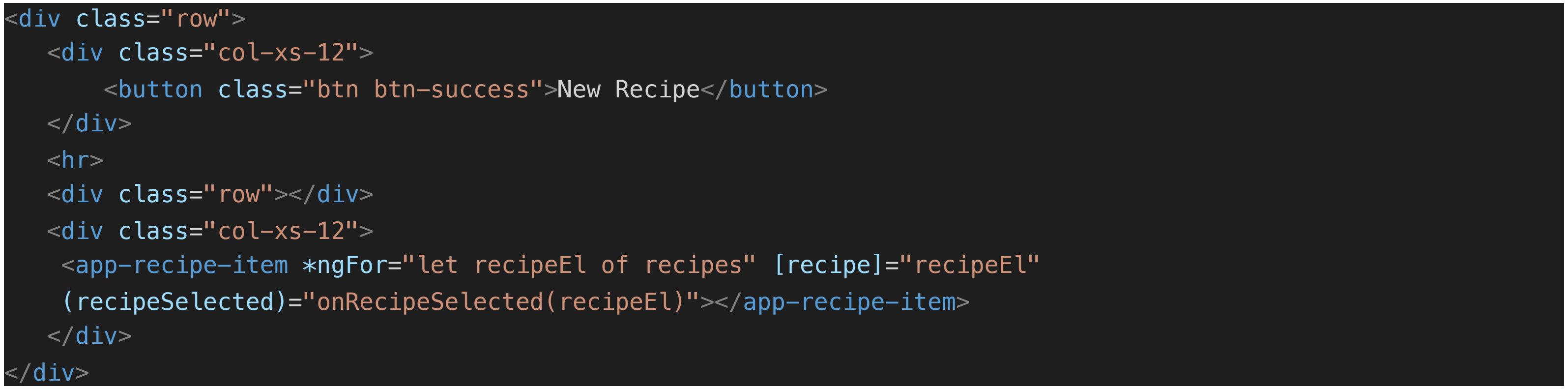
3. Passing Data with Event and Property Binding (87)



```typescript
import { Recipe } from './../../recipe.model';
import { Component, Input, OnInit, EventEmitter, Output } from '@angular/core';

@Component({
  selector: 'app-recipe-item',
  templateUrl: './recipe-item.component.html',
  styleUrls: ['./recipe-item.component.css']
})
export class RecipeItemComponent implements OnInit {

  @Input() recipe: Recipe;
  @Output() recipeSelected = new EventEmitter<void>();

  constructor() { }

  ngOnInit(): void {
  }

  selectRecipe(r) {
    this.recipeSelected.emit();
  }

}
```

recipe-item.component.ts

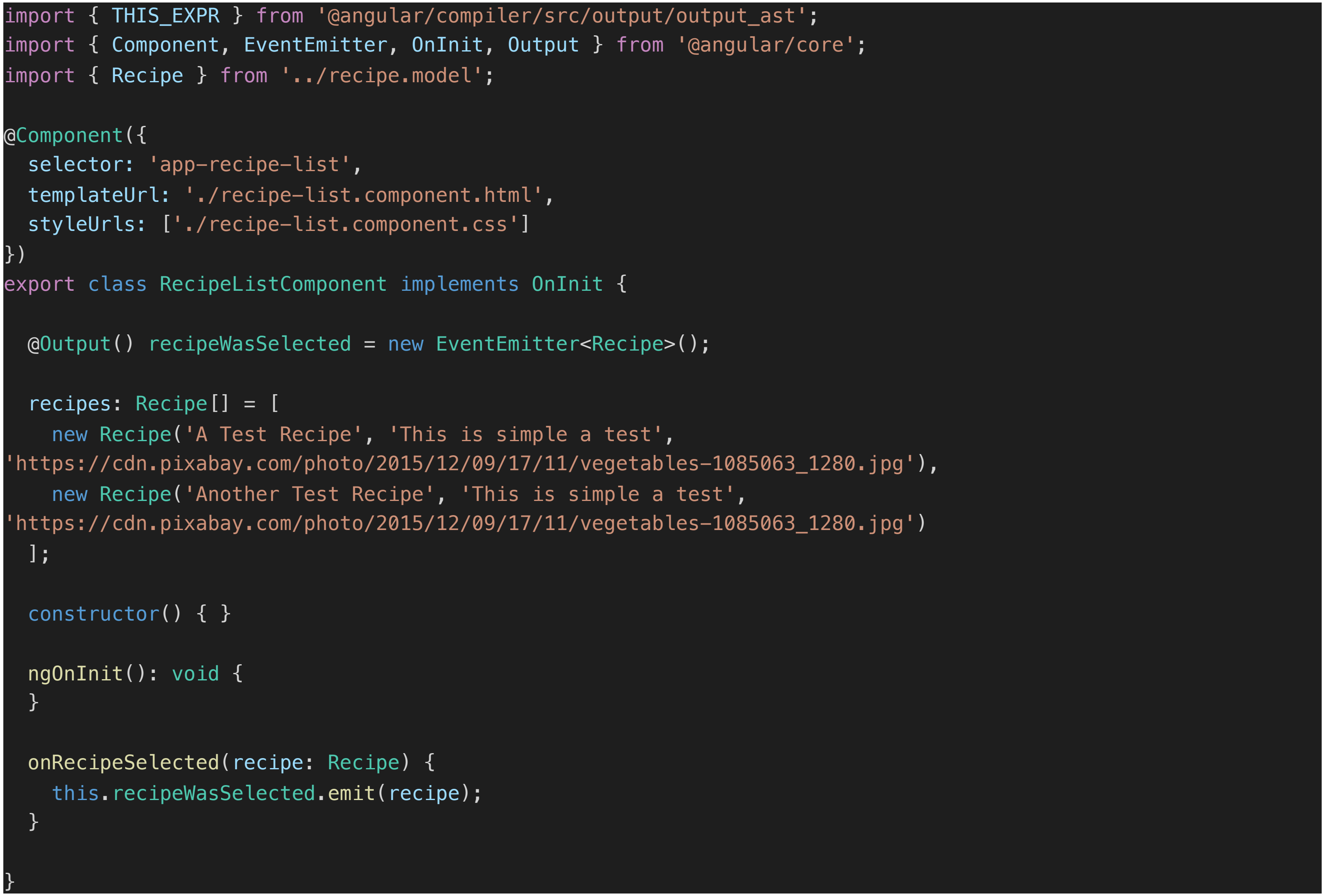
3. Passing Data with Event and Property Binding (87)



```html
<div class="row">
   <div class="col-xs-12">
       <button class="btn btn-success">New Recipe</button>
   </div>
   <hr>
   <div class="row"></div>
   <div class="col-xs-12">
    <app-recipe-item *ngFor="let recipeEl of recipes" [recipe]="recipeEl"
    (recipeSelected)="onRecipeSelected(recipeEl)"></app-recipe-item>
   </div>
</div>
```

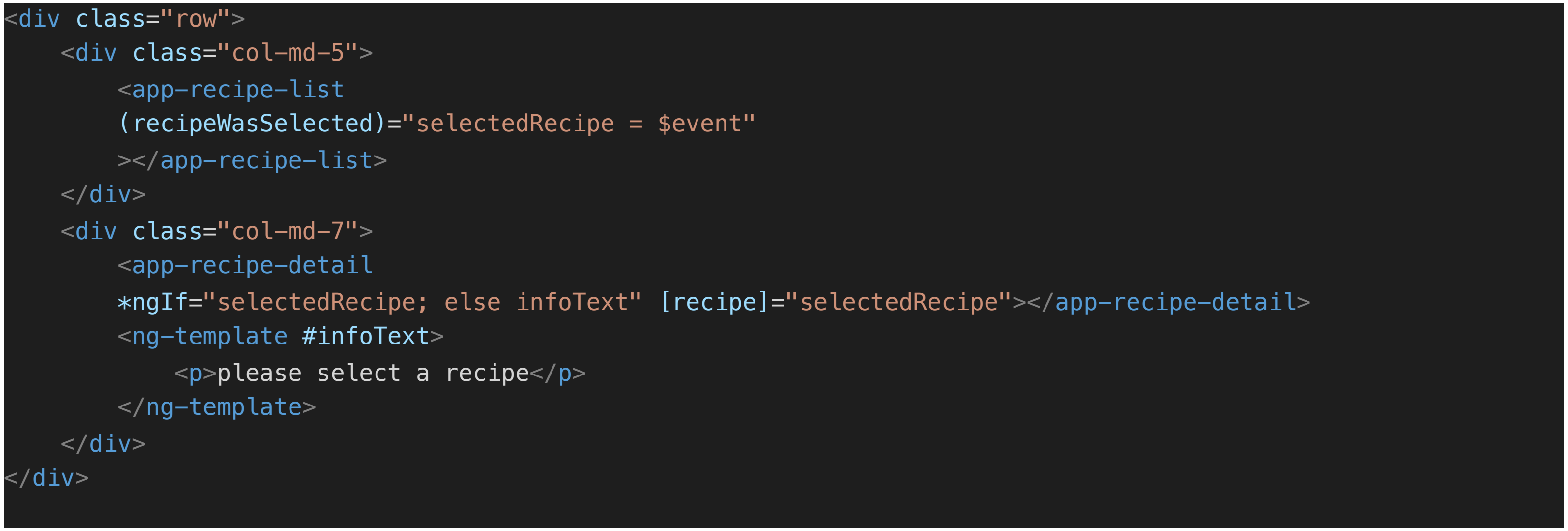
recipe-list.component.html

3. Passing Data with Event and Property Binding (87)



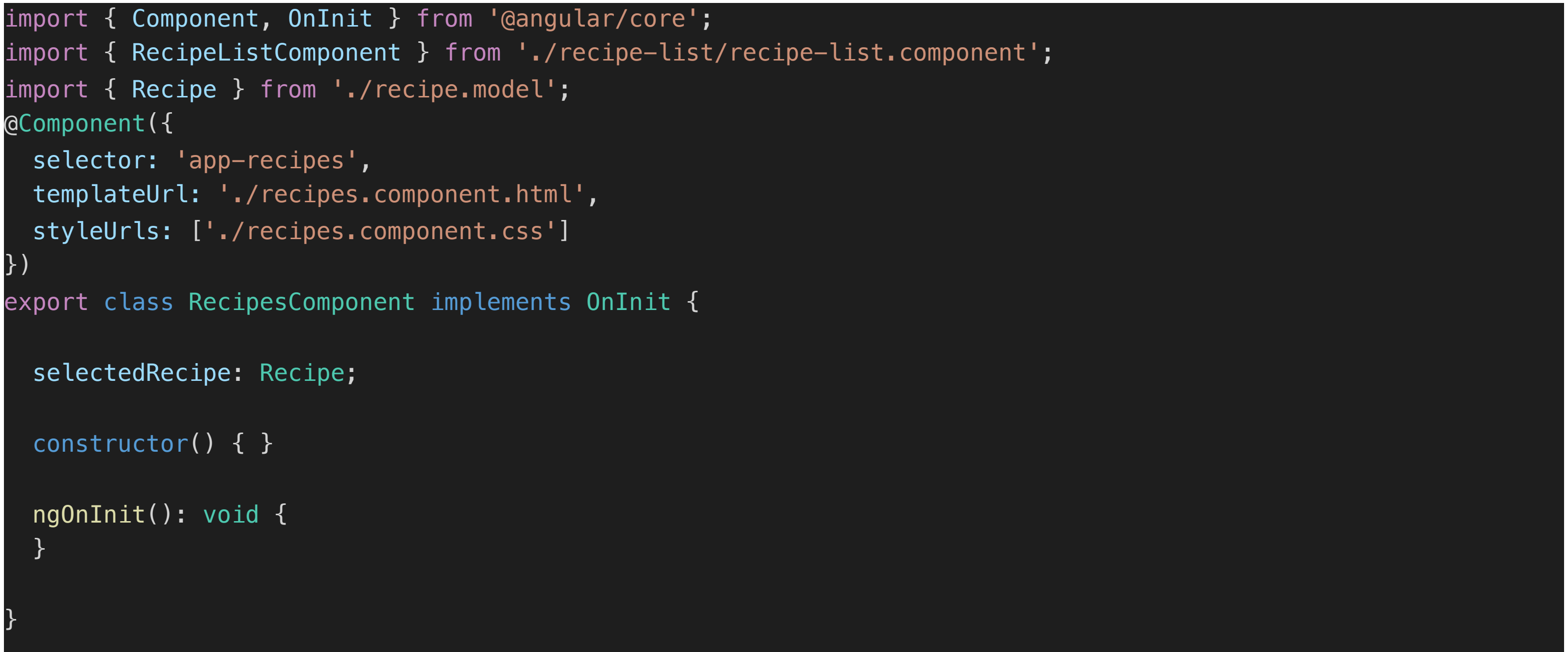
```
import { THIS_EXPR } from '@angular/compiler/src/output/output_ast';
import { Component, EventEmitter, OnInit, Output } from '@angular/core';
import { Recipe } from '../recipe.model';

@Component({
  selector: 'app-recipe-list',
  templateUrl: './recipe-list.component.html',
  styleUrls: ['./recipe-list.component.css']
})
export class RecipeListComponent implements OnInit {

  @Output() recipeWasSelected = new EventEmitter<Recipe>();

  recipes: Recipe[] = [
    new Recipe('A Test Recipe', 'This is simple a test',
'https://cdn.pixabay.com/photo/2015/12/09/17/11/vegetables-1085063_1280.jpg'),
    new Recipe('Another Test Recipe', 'This is simple a test',
'https://cdn.pixabay.com/photo/2015/12/09/17/11/vegetables-1085063_1280.jpg')
  ];

  constructor() { }

  ngOnInit(): void {
  }

  onRecipeSelected(recipe: Recipe) {
    this.recipeWasSelected.emit(recipe);
  }

}
```

3. Passing Data with Event and Property Binding (87)



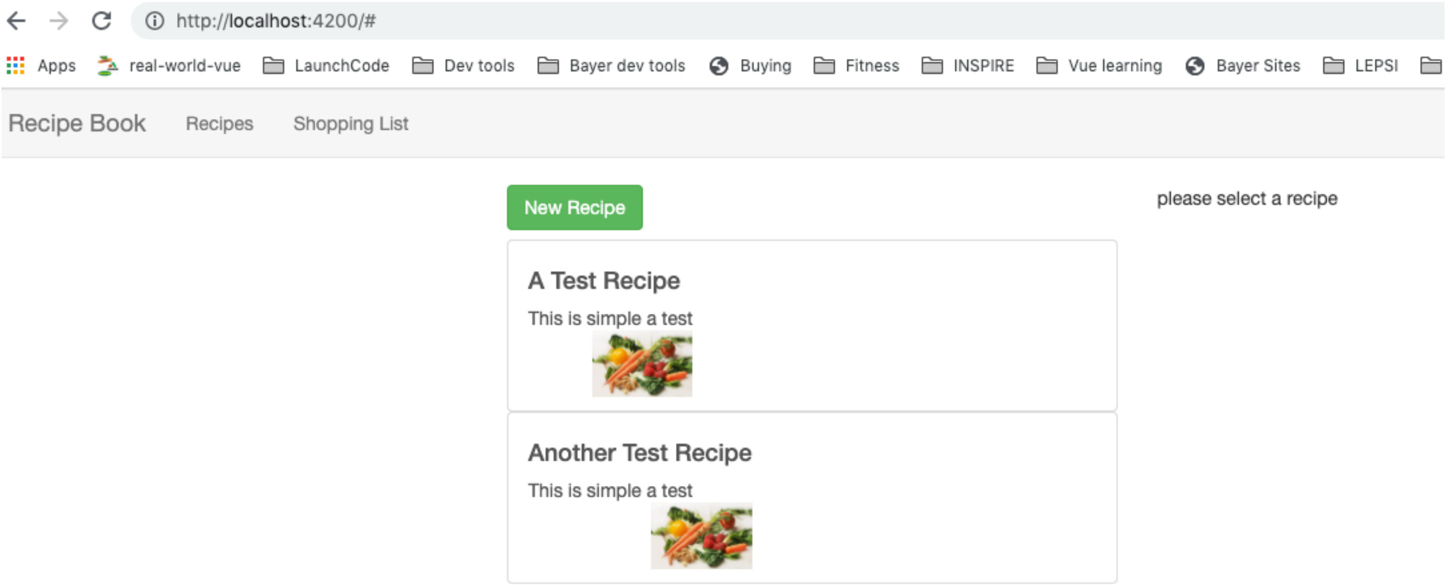
```html
<div class="row">
    <div class="col-md-5">
        <app-recipe-list
        (recipeWasSelected)="selectedRecipe = $event"
        ></app-recipe-list>
    </div>
    <div class="col-md-7">
        <app-recipe-detail
        *ngIf="selectedRecipe; else infoText" [recipe]="selectedRecipe"></app-recipe-detail>
        <ng-template #infoText>
            <p>please select a recipe</p>
        </ng-template>
    </div>
</div>
```

recipe-component.html



```typescript
import { Component, OnInit } from '@angular/core';
import { RecipeListComponent } from './recipe-list/recipe-list.component';
import { Recipe } from './recipe.model';
@Component({
  selector: 'app-recipes',
  templateUrl: './recipes.component.html',
  styleUrls: ['./recipes.component.css']
})
export class RecipesComponent implements OnInit {

  selectedRecipe: Recipe;

  constructor() { }

  ngOnInit(): void {
  }

}
```

3. Passing Data with Event and Property Binding (87)



please select a recipe

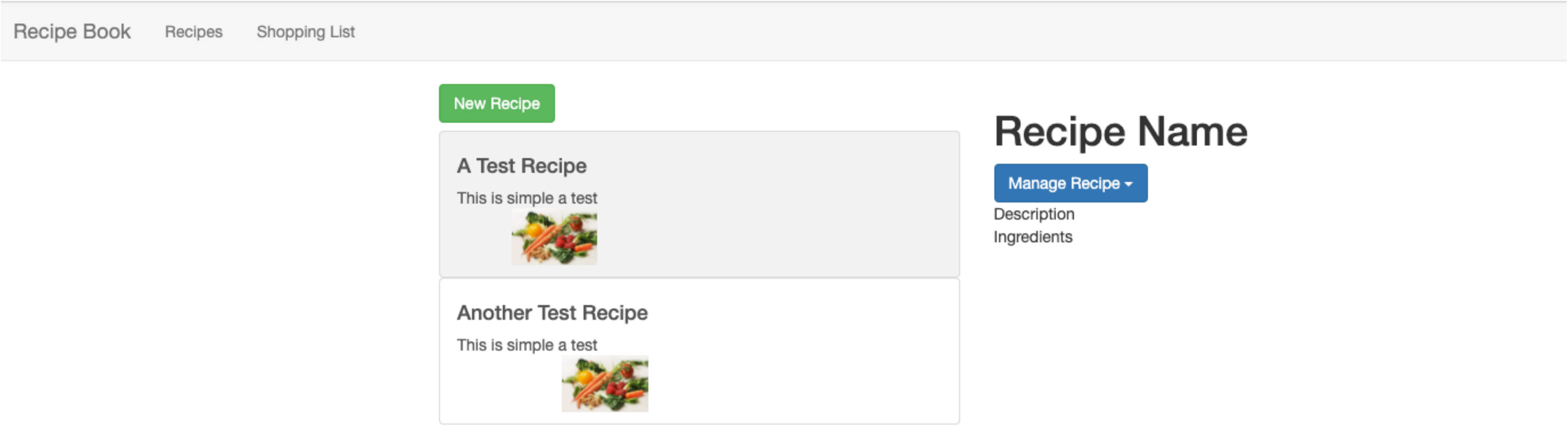
A Test Recipe

This is simple a test

Another Test Recipe

This is simple a test

3. Passing Data with Event and Property Binding (87)



A Test Recipe

This is simple a test

Another Test Recipe

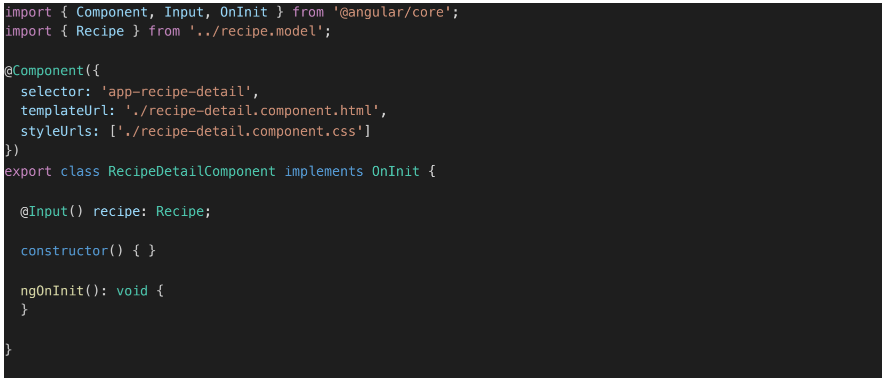
This is simple a test

Recipe Name

Description

Ingredients

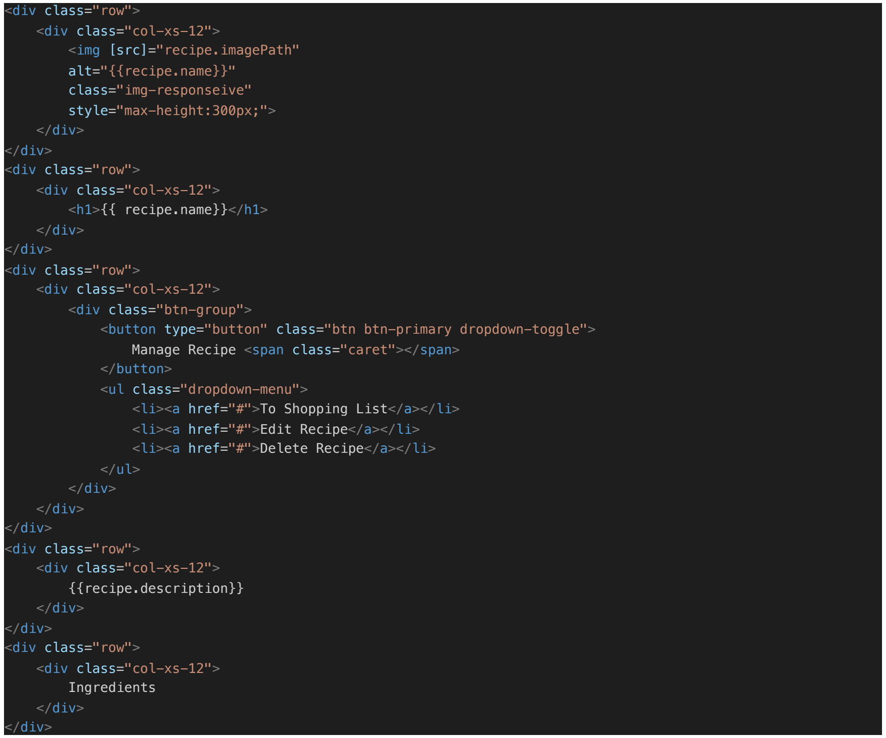
3. Passing Data with Event and Property Binding (87)



```
import { Component, Input, OnInit } from '@angular/core';
import { Recipe } from '../recipe.model';

@Component({
  selector: 'app-recipe-detail',
  templateUrl: './recipe-detail.component.html',
  styleUrls: ['./recipe-detail.component.css']
})
export class RecipeDetailComponent implements OnInit {

  @Input() recipe: Recipe;

  constructor() { }

  ngOnInit(): void {
  }

}
```

recipe-detail-component.ts

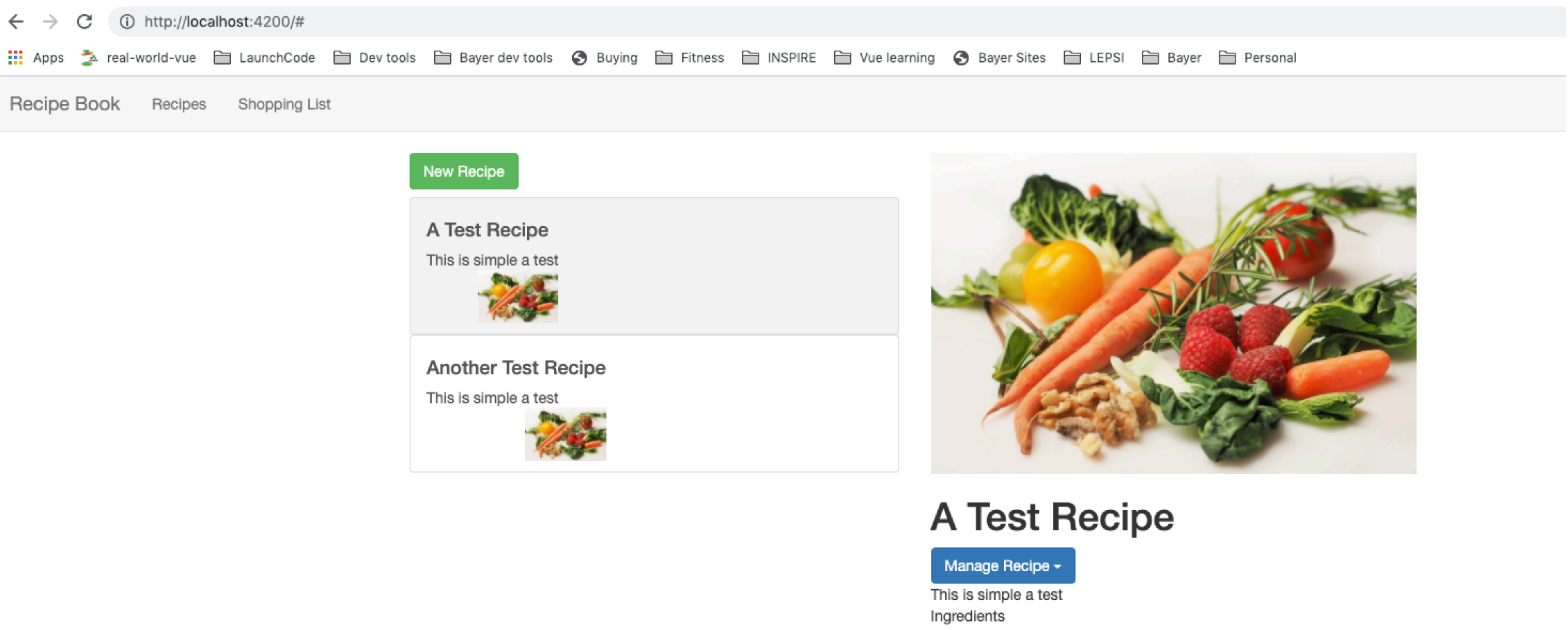
3. Passing Data with Event and Property Binding (87)



```html
<div class="row">
    <div class="col-xs-12">
        <img [src]="recipe.imagePath"
        alt="{{recipe.name}}"
        class="img-responseive"
        style="max-height:300px;">
    </div>
</div>
<div class="row">
    <div class="col-xs-12">
        <h1>{{ recipe.name}}</h1>
    </div>
</div>
<div class="row">
    <div class="col-xs-12">
        <div class="btn-group">
            <button type="button" class="btn btn-primary dropdown-toggle">
                Manage Recipe <span class="caret"></span>
            </button>
            <ul class="dropdown-menu">
                <li><a href="#">To Shopping List</a></li>
                <li><a href="#">Edit Recipe</a></li>
                <li><a href="#">Delete Recipe</a></li>
            </ul>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-xs-12">
        {{recipe.description}}
    </div>
</div>
<div class="row">
    <div class="col-xs-12">
        Ingredients
    </div>
</div>
```

recipe-detai-componentl.html

3. Passing Data with Event and Property Binding (87)



A Test Recipe

This is simple a test

Another Test Recipe

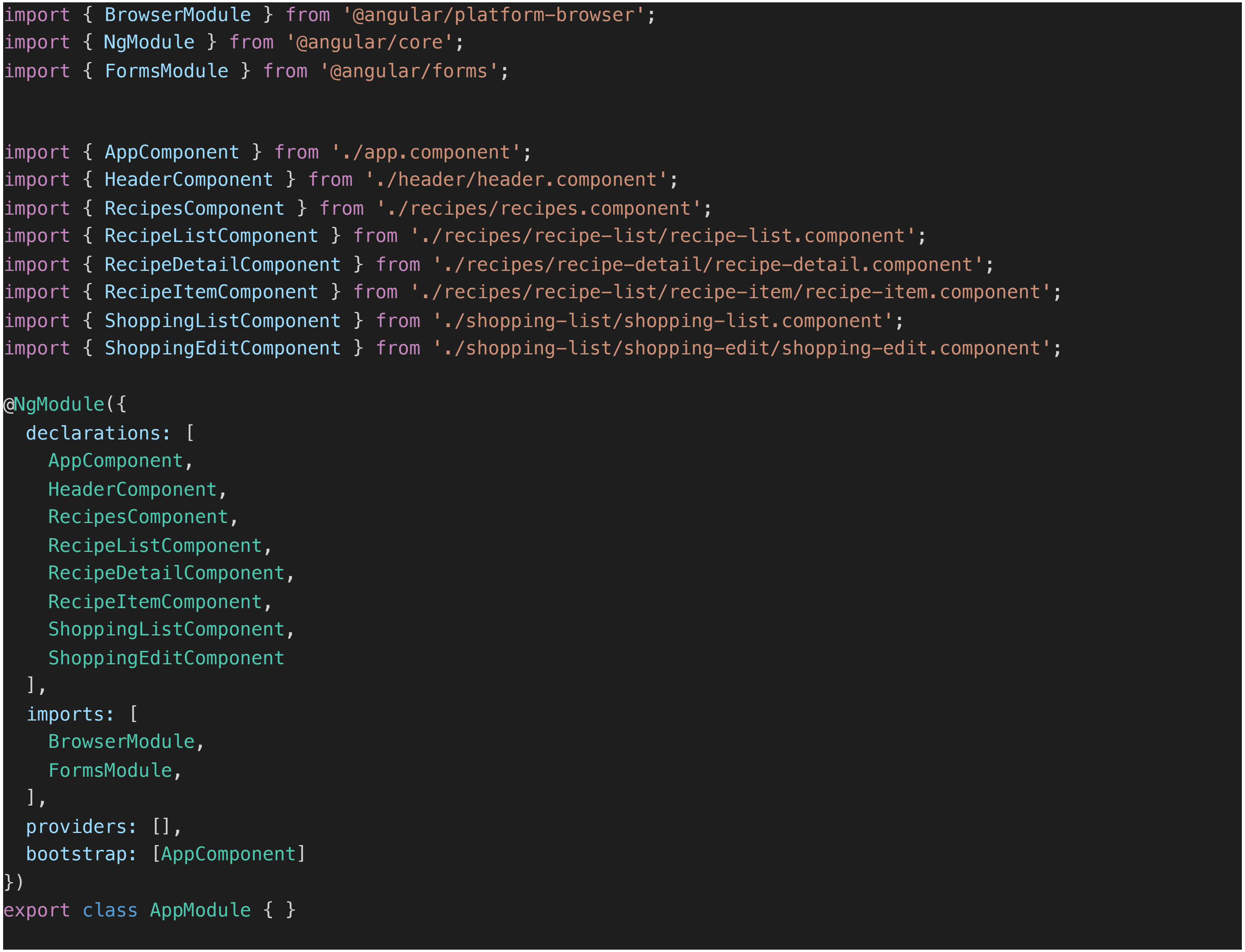
This is simple a test

A Test Recipe

This is simple a test

Ingredients

Add FormsModule to AppModule



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


import { AppComponent } from './app.component';
import { HeaderComponent } from './header/header.component';
import { RecipesComponent } from './recipes/recipes.component';
import { RecipeListComponent } from './recipes/recipe-list/recipe-list.component';
import { RecipeDetailComponent } from './recipes/recipe-detail/recipe-detail.component';
import { RecipeItemComponent } from './recipes/recipe-list/recipe-item/recipe-item.component';
import { ShoppingListComponent } from './shopping-list/shopping-list.component';
import { ShoppingEditComponent } from './shopping-list/shopping-edit/shopping-edit.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    RecipesComponent,
    RecipeListComponent,
    RecipeDetailComponent,
    RecipeItemComponent,
    ShoppingListComponent,
    ShoppingEditComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

4. Allowing the User to Add Ingredients to the Shopping List (89)

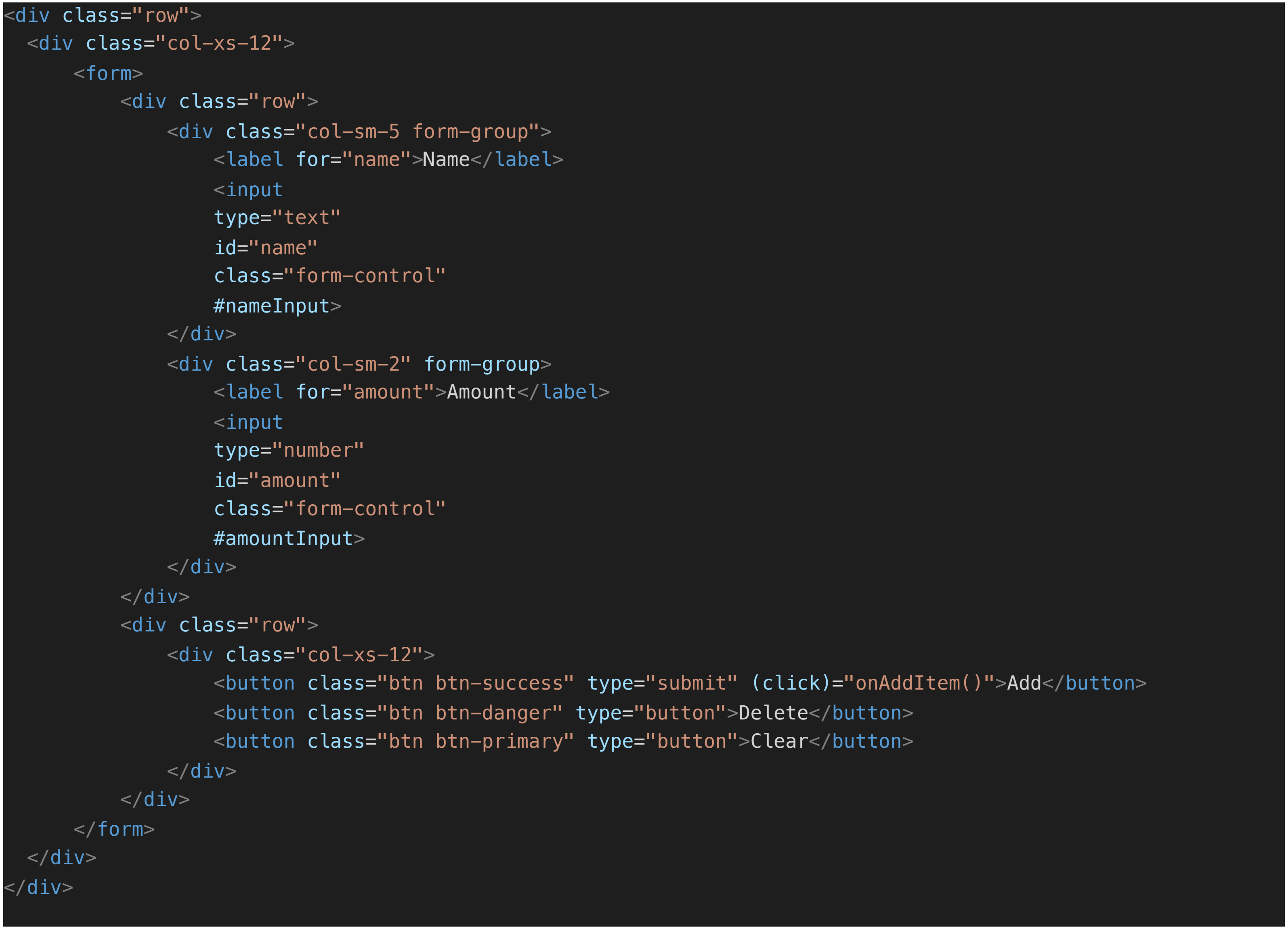
localReference for nameInput and amountInput

Pass them as argument or addViewChild

Emit event

Breakout room – 20 mins.

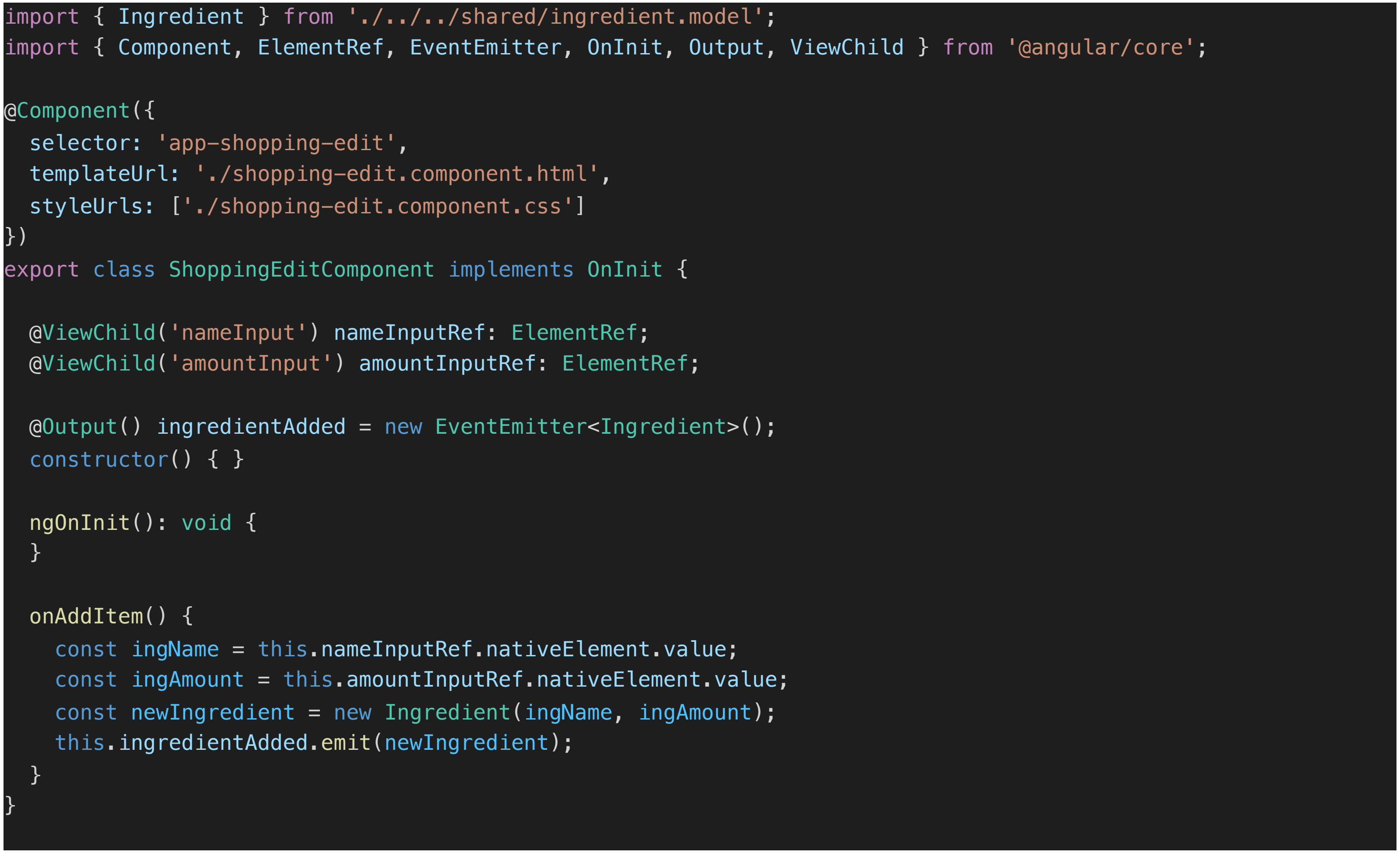
4. Allowing the User to Add Ingredients to the Shopping List (89)



```
<div class="row">
  <div class="col-xs-12">
      <form>
          <div class="row">
              <div class="col-sm-5 form-group">
                  <label for="name">Name</label>
                  <input
                  type="text"
                  id="name"
                  class="form-control"
                  #nameInput>
              </div>
              <div class="col-sm-2" form-group>
                  <label for="amount">Amount</label>
                  <input
                  type="number"
                  id="amount"
                  class="form-control"
                  #amountInput>
              </div>
          </div>
          <div class="row">
              <div class="col-xs-12">
                  <button class="btn btn-success" type="submit" (click)="onAddItem()">Add</button>
                  <button class="btn btn-danger" type="button">Delete</button>
                  <button class="btn btn-primary" type="button">Clear</button>
              </div>
          </div>
      </form>
  </div>
</div>
```

shopping-edit.component.html

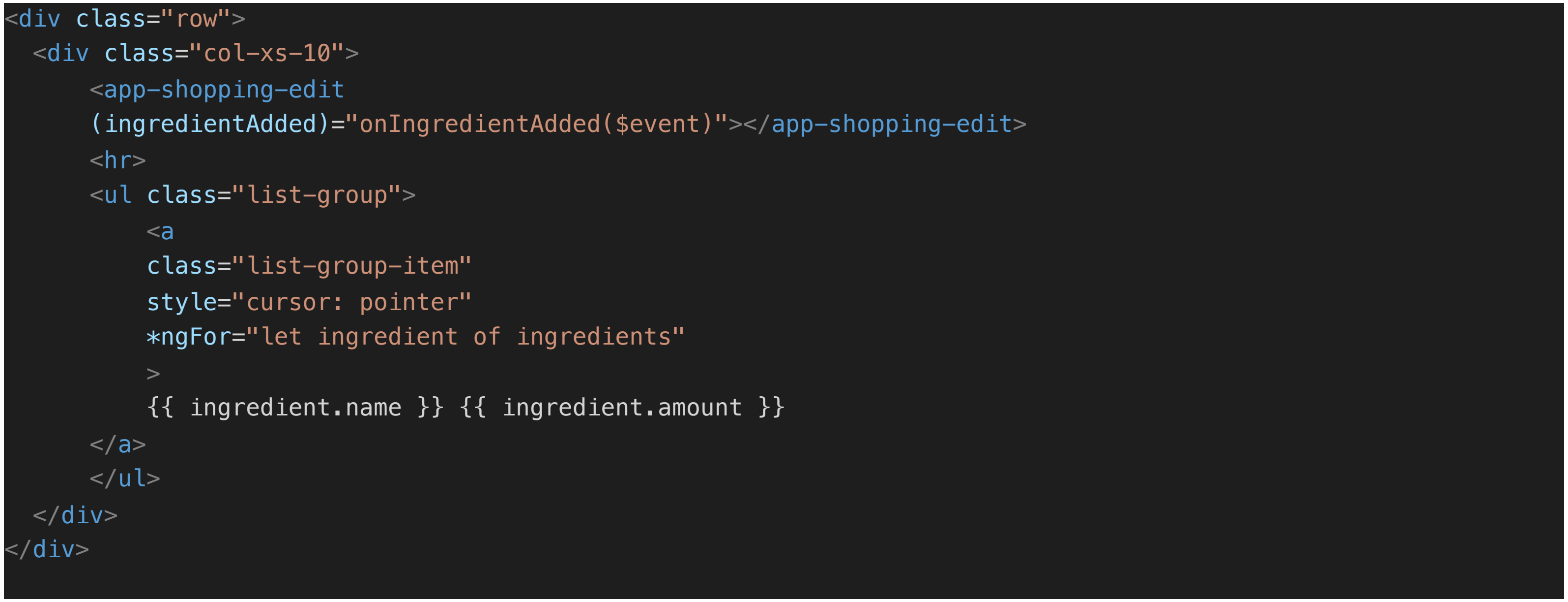
4. Allowing the User to Add Ingredients to the Shopping List (89)



```
import { Ingredient } from './../../shared/ingredient.model';
import { Component, ElementRef, EventEmitter, OnInit, Output, ViewChild } from '@angular/core';

@Component({
  selector: 'app-shopping-edit',
  templateUrl: './shopping-edit.component.html',
  styleUrls: ['./shopping-edit.component.css']
})
export class ShoppingEditComponent implements OnInit {

  @ViewChild('nameInput') nameInputRef: ElementRef;
  @ViewChild('amountInput') amountInputRef: ElementRef;

  @Output() ingredientAdded = new EventEmitter<Ingredient>();
  constructor() { }

  ngOnInit(): void {
  }

  onAddItem() {
    const ingName = this.nameInputRef.nativeElement.value;
    const ingAmount = this.amountInputRef.nativeElement.value;
    const newIngredient = new Ingredient(ingName, ingAmount);
    this.ingredientAdded.emit(newIngredient);
  }
}
```

shopping-edit.component.ts

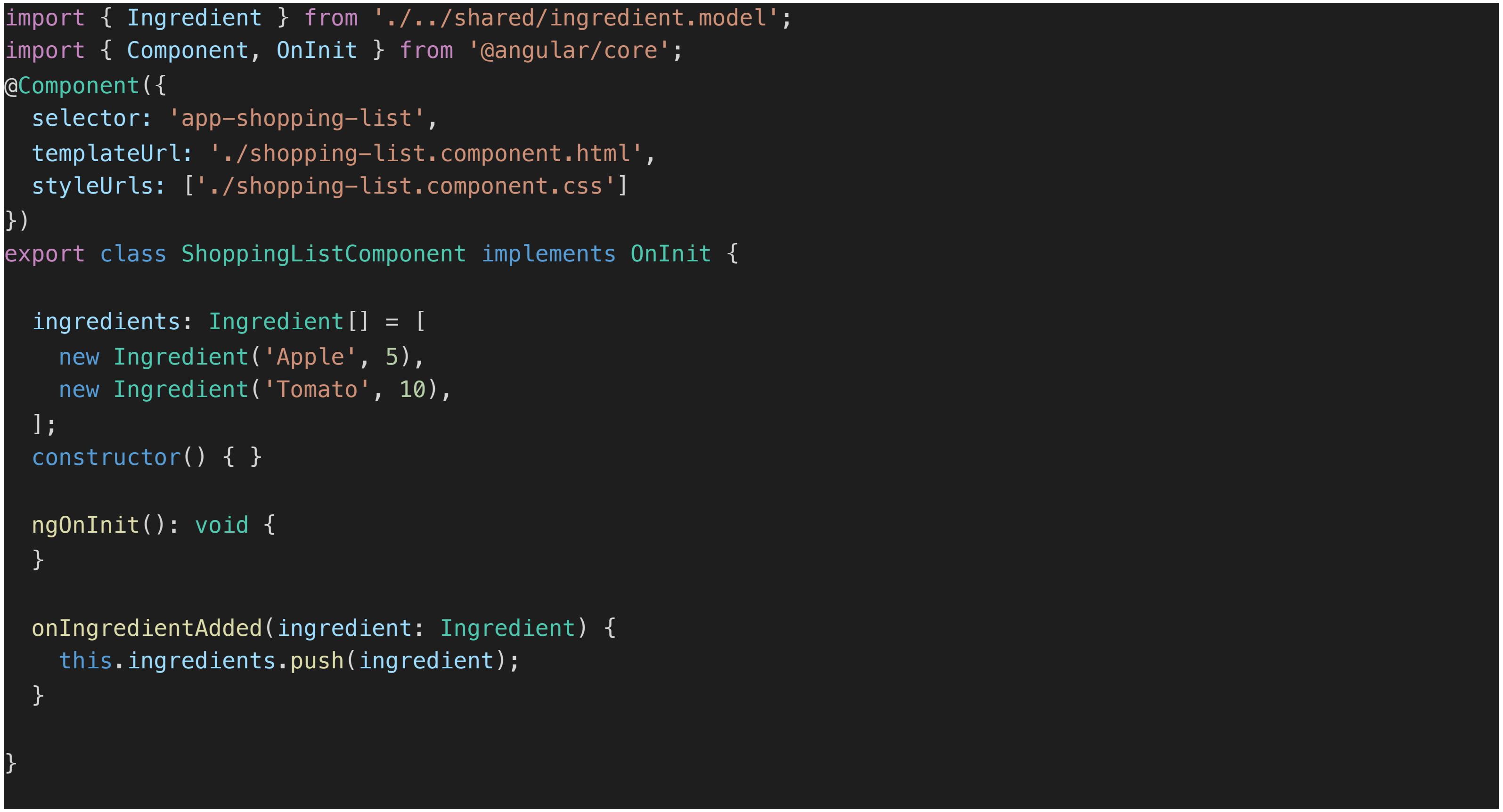
4. Allowing the User to Add Ingredients to the Shopping List (89)



```html
<div class="row">
  <div class="col-xs-10">
      <app-shopping-edit
      (ingredientAdded)="onIngredientAdded($event)"></app-shopping-edit>
      <hr>
      <ul class="list-group">
          <a
          class="list-group-item"
          style="cursor: pointer"
          *ngFor="let ingredient of ingredients"
          >
          {{ ingredient.name }} {{ ingredient.amount }}
      </a>
      </ul>
  </div>
</div>
```

shopping-list.component.html

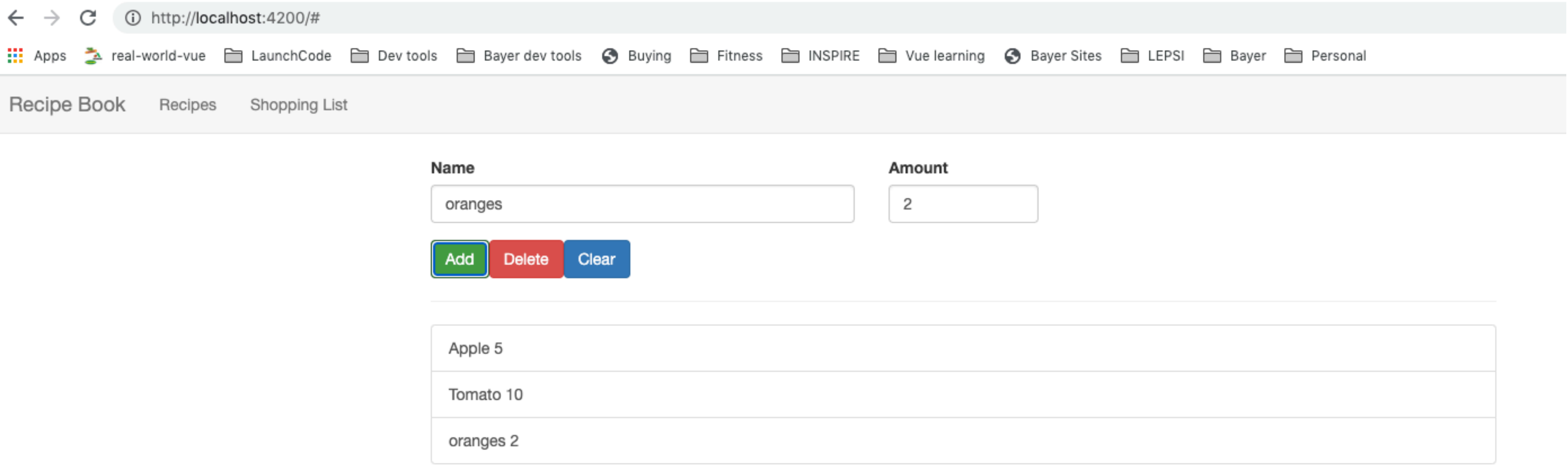
4. Allowing the User to Add Ingredients to the Shopping List (89)



```
import { Ingredient } from './../shared/ingredient.model';
import { Component, OnInit } from '@angular/core';
@Component({
  selector: 'app-shopping-list',
  templateUrl: './shopping-list.component.html',
  styleUrls: ['./shopping-list.component.css']
})
export class ShoppingListComponent implements OnInit {

  ingredients: Ingredient[] = [
    new Ingredient('Apple', 5),
    new Ingredient('Tomato', 10),
  ];
  constructor() { }

  ngOnInit(): void {
  }

  onIngredientAdded(ingredient: Ingredient) {
    this.ingredients.push(ingredient);
  }

}
```

shopping-list.component.ts

Finished app…so far!



Name: oranges

Amount: 2

- Apple 5
- Tomato 10
- oranges 2

Questions?