1. \*\*Set up your Angular project:\*\*

- Use Angular CLI to create a new project:

```bash

ng new recipe-book

cd recipe-book

```

2. \*\*Create Components:\*\*

- Create components for the main recipe book, recipe list, recipe details, and any other necessary components:

```bash

ng generate component recipe-book

ng generate component recipe-list

ng generate component recipe-details

```

3. \*\*Create a Service:\*\*

- Generate a service to handle recipe-related functionalities. Use Angular CLI:

```bash

ng generate service recipe

```

4. \*\*Update the Recipe Service:\*\*

- Update the `recipe.service.ts` file to manage recipe-related operations. Here's a simplified example:

```typescript

// recipe.service.ts

import { Injectable } from '@angular/core';

import { BehaviorSubject } from 'rxjs';

@Injectable({

providedIn: 'root',

})

export class RecipeService {

private recipesSubject = new BehaviorSubject<Recipe[]>([]);

recipes$ = this.recipesSubject.asObservable();

addRecipe(recipe: Recipe): void {

const currentRecipes = this.recipesSubject.value;

const newRecipes = [...currentRecipes, recipe];

this.recipesSubject.next(newRecipes);

}

getRecipes(): Recipe[] {

return this.recipesSubject.value;

}

}

export interface Recipe {

name: string;

ingredients: string[];

instructions: string;

}

```

5. \*\*Update the Recipe List Component:\*\*

- Update the `recipe-list.component.ts` file to use the Recipe Service and display the list of recipes:

```typescript

// recipe-list.component.ts

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

import { RecipeService, Recipe } from '../services/recipe.service';

@Component({

selector: 'app-recipe-list',

templateUrl: './recipe-list.component.html',

styleUrls: ['./recipe-list.component.css'],

})

export class RecipeListComponent implements OnInit {

recipes: Recipe[] = [];

constructor(private recipeService: RecipeService) {}

ngOnInit() {

this.recipes = this.recipeService.getRecipes();

this.recipeService.recipes$.subscribe((recipes) => {

this.recipes = recipes;

});

}

}

```

6. \*\*Update the Recipe List Component HTML:\*\*

- Update the `recipe-list.component.html` file to display the list of recipes:

```html

<!-- recipe-list.component.html -->

<div>

<h2>Recipe List</h2>

<ul>

<li \*ngFor="let recipe of recipes">

<a [routerLink]="['/recipe', recipe.name]">{{ recipe.name }}</a>

</li>

</ul>

</div>

```

7. \*\*Use the Components in the Recipe Book:\*\*

- Update the `recipe-book.component.html` file to use the Recipe List component and add functionality for adding recipes:

```html

<!-- recipe-book.component.html -->

<div>

<app-recipe-list></app-recipe-list>

<div>

<label for="recipeName">Enter Recipe Name:</label>

<input type="text" id="recipeName" [(ngModel)]="newRecipeName" />

<button (click)="addRecipe()">Add Recipe</button>

</div>

</div>

```

- Update the `recipe-book.component.ts` file:

```typescript

// recipe-book.component.ts

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

import { RecipeService, Recipe } from '../services/recipe.service';

@Component({

selector: 'app-recipe-book',

templateUrl: './recipe-book.component.html',

styleUrls: ['./recipe-book.component.css'],

})

export class RecipeBookComponent {

newRecipeName: string;

constructor(private recipeService: RecipeService) {}

addRecipe(): void {

if (this.newRecipeName) {

const newRecipe: Recipe = {

name: this.newRecipeName,

ingredients: [],

instructions: '',

};

this.recipeService.addRecipe(newRecipe);

this.newRecipeName = '';

}

}

}

```

Make sure to add `RouterModule` to your `app.module.ts` to enable routing. Add the following import statements and update the imports array:

```typescript

// app.module.ts

import { RouterModule, Routes } from '@angular/router';

import { RecipeBookComponent } from './recipe-book/recipe-book.component';

import { RecipeDetailsComponent } from './recipe-details/recipe-details.component';

const routes: Routes = [

{ path: 'recipe/:name', component: RecipeDetailsComponent },

{ path: '', component: RecipeBookComponent },

];

@NgModule({

// ...

imports: [

// ...

RouterModule.forRoot(routes),

],

// ...

})

```