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

- Use Angular CLI to create a new project:

```bash

ng new expense-tracker

cd expense-tracker

```

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

- Create components for the main expense tracker, expense list, and any other necessary components:

```bash

ng generate component expense-tracker

ng generate component expense-list

```

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

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

```bash

ng generate service expense

```

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

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

```typescript

// expense.service.ts

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

import { BehaviorSubject } from 'rxjs';

@Injectable({

providedIn: 'root',

})

export class ExpenseService {

private expensesSubject = new BehaviorSubject<number[]>([]);

expenses$ = this.expensesSubject.asObservable();

addExpense(amount: number): void {

const currentExpenses = this.expensesSubject.value;

const newExpenses = [...currentExpenses, amount];

this.expensesSubject.next(newExpenses);

}

}

```

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

- Update the `expense-list.component.ts` file to use the Expense Service:

```typescript

// expense-list.component.ts

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

import { ExpenseService } from '../services/expense.service';

@Component({

selector: 'app-expense-list',

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

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

})

export class ExpenseListComponent implements OnInit {

expenses: number[] = [];

constructor(private expenseService: ExpenseService) {}

ngOnInit() {

this.expenseService.expenses$.subscribe((expenses) => {

this.expenses = expenses;

});

}

}

```

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

- Update the `expense-list.component.html` file to display the list of expenses:

```html

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

<div>

<h2>Expense List</h2>

<ul>

<li \*ngFor="let expense of expenses">{{ expense }}</li>

</ul>

</div>

```

7. \*\*Use the Components in the Expense Tracker:\*\*

- Update the `expense-tracker.component.html` file to use the Expense List component and add functionality for adding expenses:

```html

<!-- expense-tracker.component.html -->

<div>

<app-expense-list></app-expense-list>

<div>

<label for="expenseAmount">Enter Expense Amount:</label>

<input type="number" id="expenseAmount" [(ngModel)]="newExpenseAmount" />

<button (click)="addExpense()">Add Expense</button>

</div>

</div>

```

- Update the `expense-tracker.component.ts` file:

```typescript

// expense-tracker.component.ts

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

import { ExpenseService } from '../services/expense.service';

@Component({

selector: 'app-expense-tracker',

templateUrl: './expense-tracker.component.html',

styleUrls: ['./expense-tracker.component.css'],

})

export class ExpenseTrackerComponent {

newExpenseAmount: number;

constructor(private expenseService: ExpenseService) {}

addExpense(): void {

if (this.newExpenseAmount) {

this.expenseService.addExpense(this.newExpenseAmount);

this.newExpenseAmount = null;

}

}

}

```

Make sure to add FormsModule to your `app.module.ts` for the ngModel to work. Add the following import statements and update the imports array:

```typescript

// app.module.ts

import { FormsModule } from '@angular/forms';

@NgModule({

// ...

imports: [

// ...

FormsModule,

],

// ...

})

```