**USE HTML5’S BUILT-IN WEB STORAGE TO PERSIST STATE IN ANGULAR**

*Uses forms and services.*

1. Use Web Storage to persist state in Angular
   1. Use JavaScripts’ localStorage
   2. We can’t call it directly like in Vanilla JavaScript. Instead we will create a service wrapper that encapsulates the localStorage functionality.
2. localStorage data is stored in key-value pairs. It has four main functions, which we will implement in our service wrapper:
   1. localStorage.setItem('key', 'value')- stores an item in the browser storage
   2. localStorage.getItem('key')- reads the value for the specified key item from the browser storage.
   3. localStorage.removeItem('key') - removes/deletes the item with the specified key from the browser storage.
   4. localStorage.clear() - removes all the items in the storage
3. First, create a new application
   1. ng new local-storage-app –defaults
4. Create the service wrapper called local-storage

***local-storage-service.ts***

import { Injectable } from '@angular/core';@Injectable({  
 providedIn: 'root'  
})export class LocalStorageService {  
 constructor() { }  
 public setItem(key: string, value: string) {  
 localStorage.setItem(key, value);  
 }  
   
 public getItem(key: string){   
 return localStorage.getItem(key)  
 } public removeItem(key:string) {  
 localStorage.removeItem(key);  
 } public clear(){  
 localStorage.clear();   
 }}

1. To use the local storage service in any component in the application, import the service and inject it via dependency injection in the constructor.
2. Create a user interface that allows you to enter and store a person’s details, query, and delete stored data via the key. This will interact with the service.

Graphical user interface, application

Description automatically generated

1. The code simply gets user input and uses two-way data binding to get the values in the component. Clicking any of the buttons will trigger the necessary functions in the component.

***app.component.html***

<h4> Add person </h4>  
 <table>  
 <tr>  
 <td>Firstname</td>  
 <td> <input [(ngModel)]="firstname" name="firstname" type="text"></td>  
 </tr> <tr>  
 <td>Lastname</td>  
 <td> <input [(ngModel)]="lastname" name="lastname" type="text"></td>   
 </tr> <tr>   
 <td>Age</td>   
 <td> <input [(ngModel)]="age" name="age" type="number"></td> </tr><tr> <td></td> <td><button (click)="addPerson()">Add</button></td></tr>  
</table><h4> Get person </h4>  
<input [(ngModel)]="queryKey" name="queryKey" type="text"><br>  
<button (click)="getPerson(queryKey)"> Get </button><h4>Result</h4>  
{{queryResult}}<h4> Delete person</h4>  
 <input [(ngModel)]="removeKey" name="removeKey" type="text"> <br>  
 <button (click)="deletePerson(removeKey)"> Delete </button><h4> Reset store </h4>  
<button (click)="reset()"> Reset </button>

1. Import the *FormsModule*and add it to the imports array, of the main module file i.e *app.module.ts*. We need the *FormsModule*because we are using two-way data binding.
2. In the ***app.component.ts*** file, there are properties for the person’s details that are used for two-way data binding. There is a person object which is JSON stringified to be stored in the browser storage.
3. Then save all pages, and serve up the application. Fill out information, obtain it in the next form, and delete it.

***app.component.ts***

import { Component } from '@angular/core';  
import { LocalStorageService } from './local-storage.service';@Component({  
 selector: 'app-root',  
 templateUrl: './app.component.html',  
 styleUrls: ['./app.component.css']  
})export class AppComponent {  
 title = 'local-storage-app';  
 constructor( private localStorageService: LocalStorageService) {} public firstname: string = '';  
 public lastname: string = '';  
 public age: number = null; public queryKey: string;  
 public removeKey: string; public queryResult: string; public person = {firstName: this.firstname, lastName: this.lastname, age: this.age }public addPerson(){  
 this.person.firstName = this.firstname  
 this.person.age = this.age;  
 this.person.lastName = this.lastname;  
 this.localStorageService.setItem(this.firstname, JSON.stringify(this.person))  
} public getPerson(nameToRemove:string){  
 this.queryResult = this.localStorageService.getItem(nameToRemove);  
}public deletePerson(nameToDelete:string){  
 this.localStorageService.removeItem(nameToDelete);  
}public reset(){  
 this.localStorageService.clear();  
}}