

Pokemon Trainer Registration Form

Name:

Age:

Address:

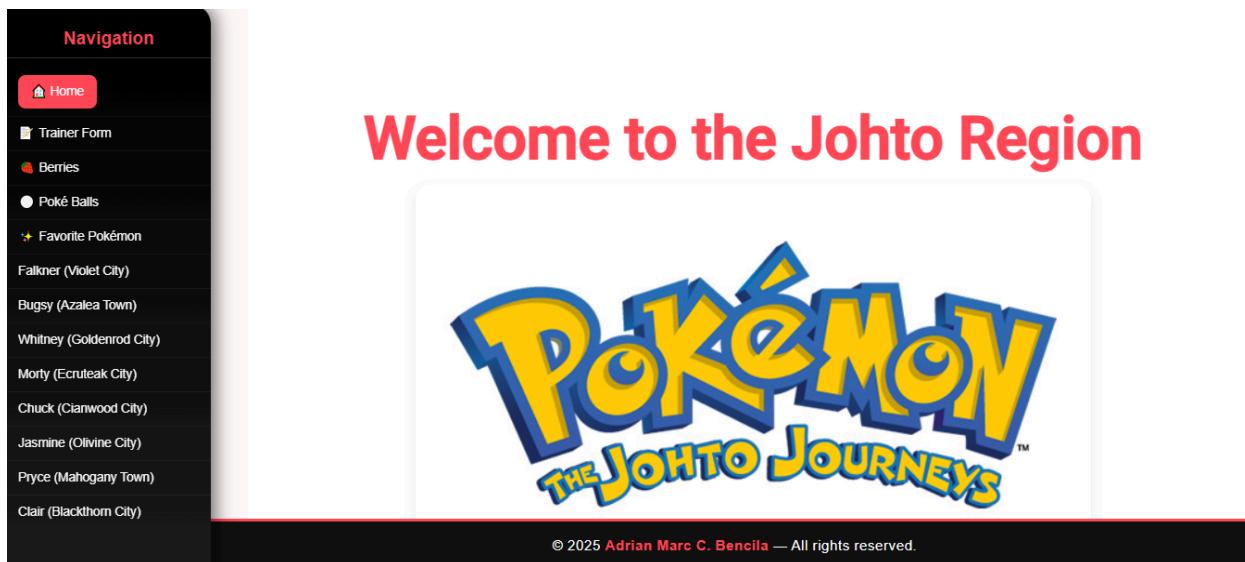
Favorite Pokemon:

Pokemon Trainer List:

Name: Pikachu
Age: 4
Address: pokemon
Favorite Pokemon: pichu

Submit

© 2025 Adrian Marc C. Bencila — All rights reserved.



Navigation

- Home
- Trainer Form
- Berries**
- Poké Balls
- Favorite Pokémon
- Falkner (Violet City)
- Bugsy (Azalea Town)
- Whitney (Goldenrod City)
- Morty (Ecruteak City)
- Chuck (Cianwood City)
- Jasmine (Olivine City)
- Pryce (Mahogany Town)
- Clair (Blackthorn City)

JOHTO REGION BERRIES

Berry	Name	Effect
	Oran Berry	Restores 10 HP
	Sitrus Berry	Restores 25% HP
	Lum Berry	Cures any status condition
	Cheri Berry	Cures paralysis
	Chesto Berry	Wakes up a sleeping Pokéémon

© 2025 Adrian Marc C. Bencila — All rights reserved.

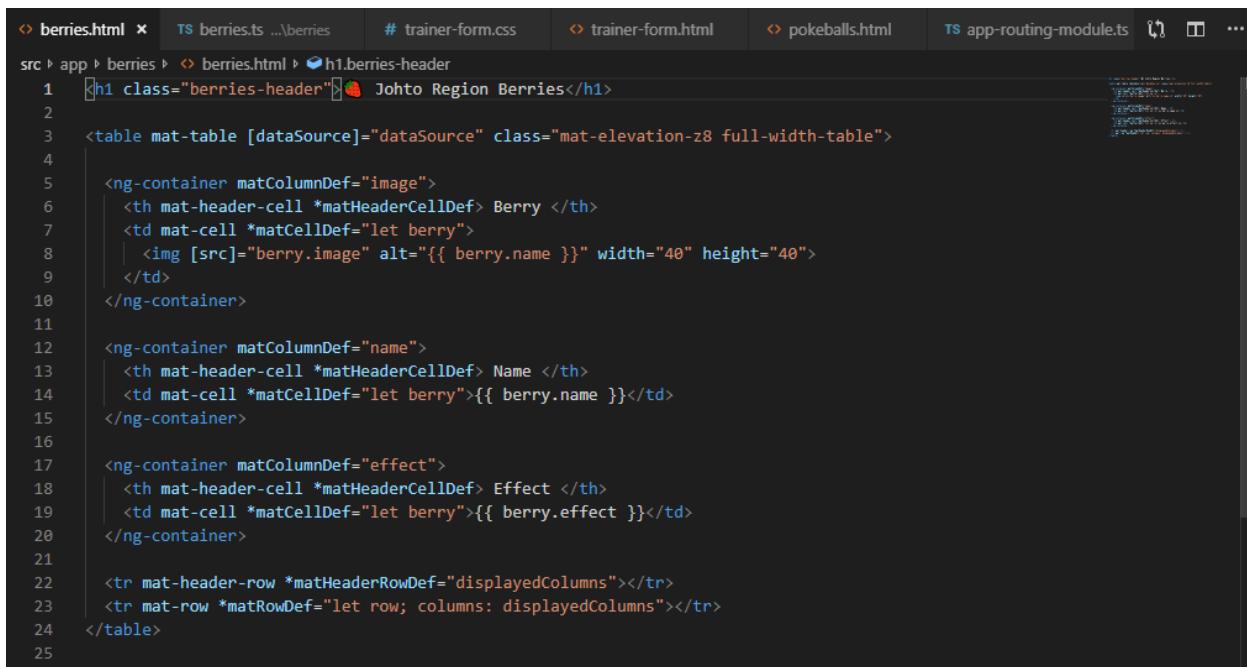
Navigation

- Home
- Trainer Form
- Berries**
- Poké Balls**
- Favorite Pokémon
- Falkner (Violet City)
- Bugsy (Azalea Town)
- Whitney (Goldenrod City)
- Morty (Ecruteak City)
- Chuck (Cianwood City)
- Jasmine (Olivine City)
- Pryce (Mahogany Town)
- Clair (Blackthorn City)

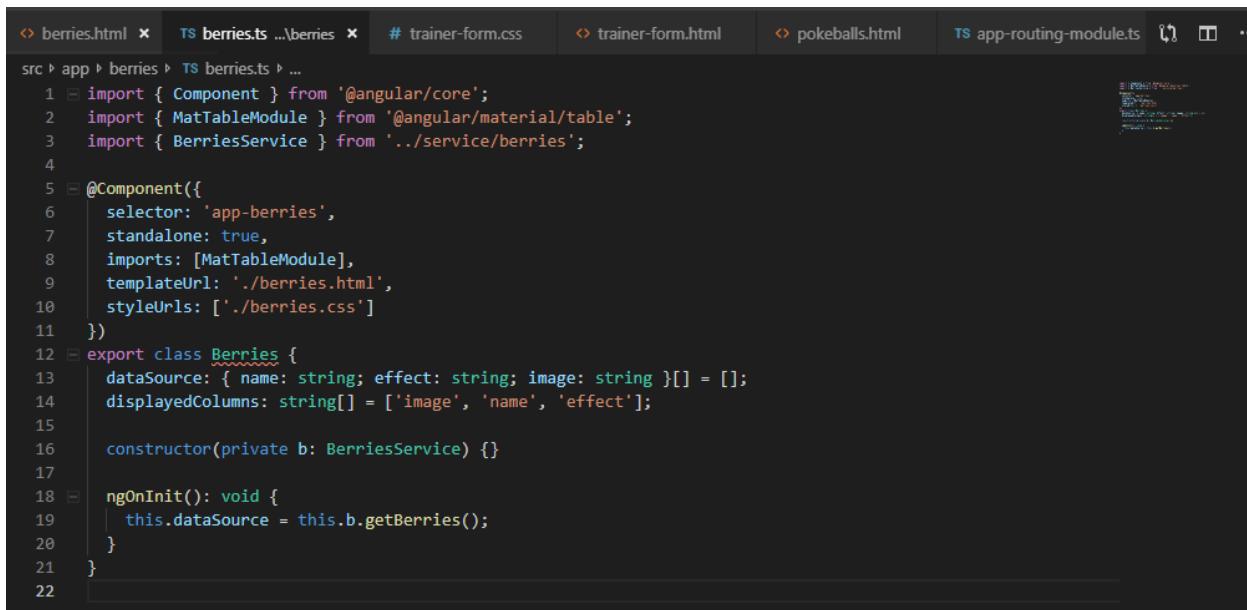
POKÉ BALLS COLLECTION

Poké Ball	Name	Type
	Poké Ball	Standard
	Great Ball	Improved
	Ultra Ball	High Performance
	Master Ball	Legendary
	Premier Ball	Special

© 2025 Adrian Marc C. Bencila — All rights reserved.



```
<h1 class="berries-header"> Johto Region Berries</h1>
<table mat-table [dataSource]="dataSource" class="mat-elevation-z8 full-width-table">
  <ng-container matColumnDef="image">
    <th mat-header-cell *matHeaderCellDef> Berry </th>
    <td mat-cell *matCellDef="let berry">
      <img [src]="berry.image" alt="{{ berry.name }}" width="40" height="40" />
    </td>
  </ng-container>
  <ng-container matColumnDef="name">
    <th mat-header-cell *matHeaderCellDef> Name </th>
    <td mat-cell *matCellDef="let berry">{{ berry.name }}</td>
  </ng-container>
  <ng-container matColumnDef="effect">
    <th mat-header-cell *matHeaderCellDef> Effect </th>
    <td mat-cell *matCellDef="let berry">{{ berry.effect }}</td>
  </ng-container>
  <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
  <tr mat-row *matRowDef="let row; columns: displayedColumns"></tr>
</table>
```



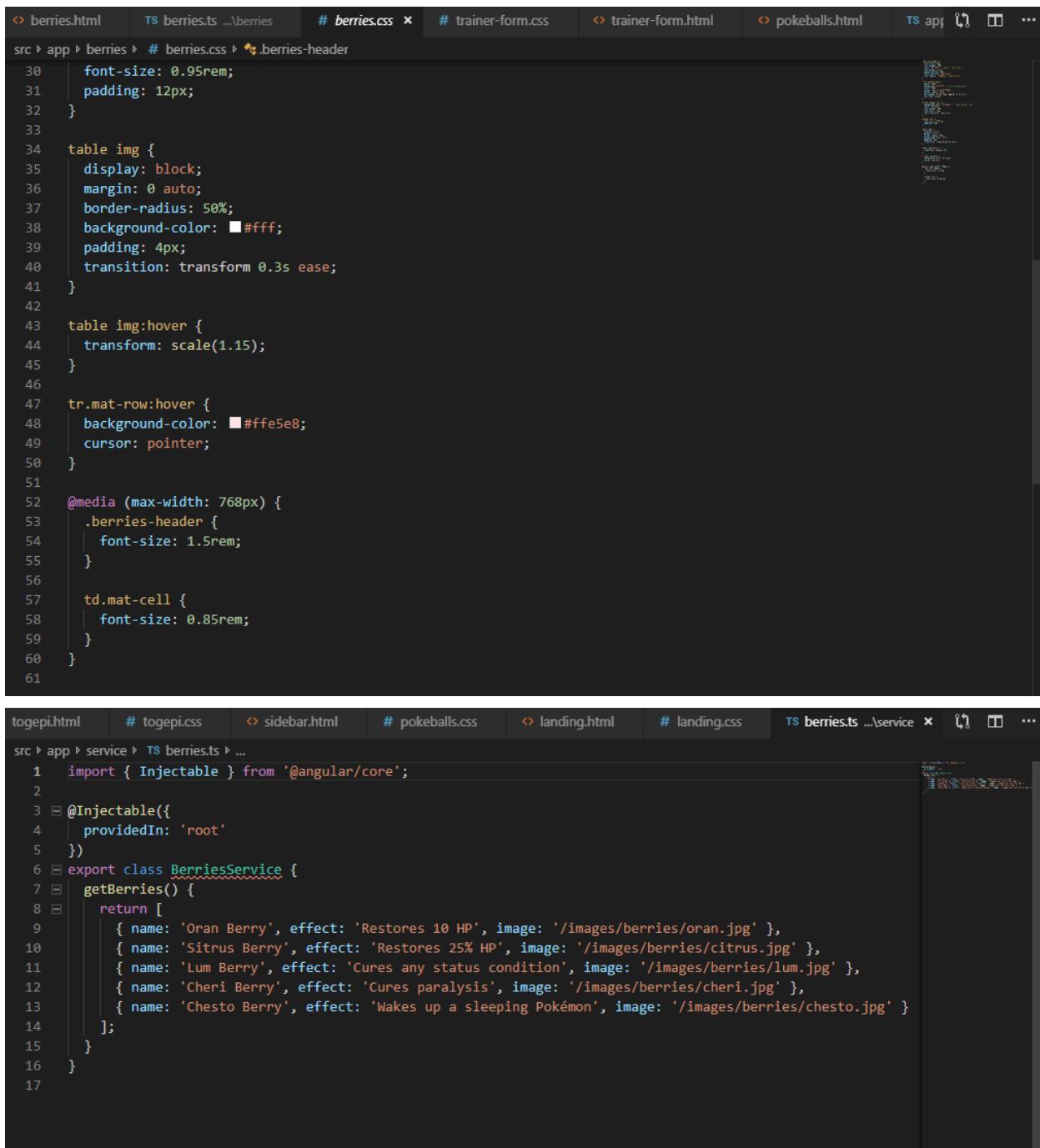
```
import { Component } from '@angular/core';
import { MatTableModule } from '@angular/material/table';
import { BerriesService } from '../service/berries';

@Component({
  selector: 'app-berries',
  standalone: true,
  imports: [MatTableModule],
  templateUrl: './berries.html',
  styleUrls: ['./berries.css']
})
export class Berries {
  dataSource: { name: string; effect: string; image: string }[] = [];
  displayedColumns: string[] = ['image', 'name', 'effect'];

  constructor(private b: BerriesService) {}

  ngOnInit(): void {
    this.dataSource = this.b.getBerries();
  }
}
```

```
src > app > berries > #berries.css > *.berries-header
1  .berries-header {
2    font-size: 2rem;
3    font-weight: 700;
4    color: #ff4655; /* pastel red accent */
5    text-align: center;
6    margin-bottom: 20px;
7    text-transform: uppercase;
8    font-family: 'Poppins', sans-serif;
9  }
10
11 .full-width-table {
12   width: 100%;
13   background: #ffffff; /* white background */
14   color: #0000;
15   border: 2px solid #ff4655;
16   border-radius: 12px;
17   box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
18   overflow: hidden;
19 }
20
21 th.mat-header-cell {
22   background-color: #ffe5e8; /* light pastel red */
23   color: #ff4655;
24   font-weight: 600;
25   font-size: 1rem;
26   text-transform: uppercase;
27 }
28
29 td.mat-cell {
30   font-size: 0.95rem;
31   padding: 12px;
32 }
```



The screenshot shows a code editor with two tabs open:

- berries.css**: A CSS file containing styles for a table and its rows. It includes a media query for screens up to 768px, adjusting font sizes and colors.
- berries.ts**: A TypeScript file defining a service named BerriesService. It has a single method getBerries() which returns an array of berry objects. Each berry object has properties: name, effect, and image.

```
berries.css
font-size: 0.95rem;
padding: 12px;
}

table img {
display: block;
margin: 0 auto;
border-radius: 50%;
background-color: #fff;
padding: 4px;
transition: transform 0.3s ease;
}

table img:hover {
transform: scale(1.15);
}

tr.mat-row:hover {
background-color: #ffe5e8;
cursor: pointer;
}

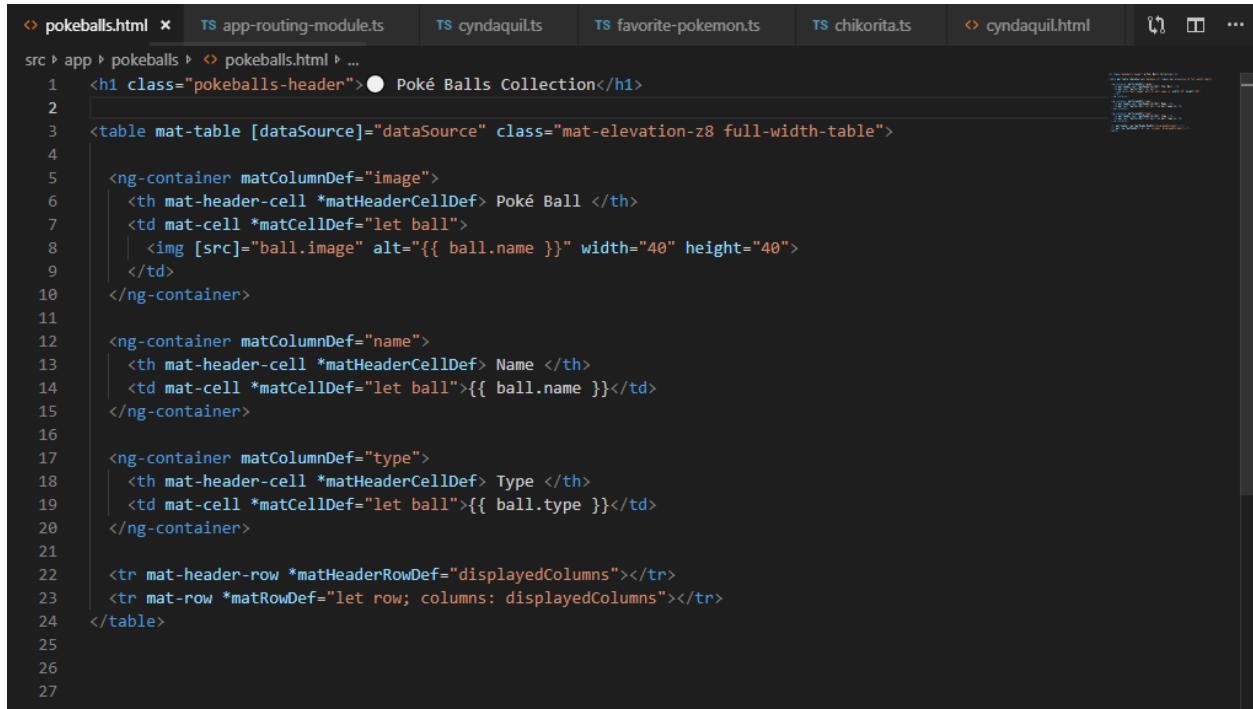
@media (max-width: 768px) {
.berries-header {
font-size: 1.5rem;
}

td.mat-cell {
font-size: 0.85rem;
}
}

berries.ts
import { Injectable } from '@angular/core';

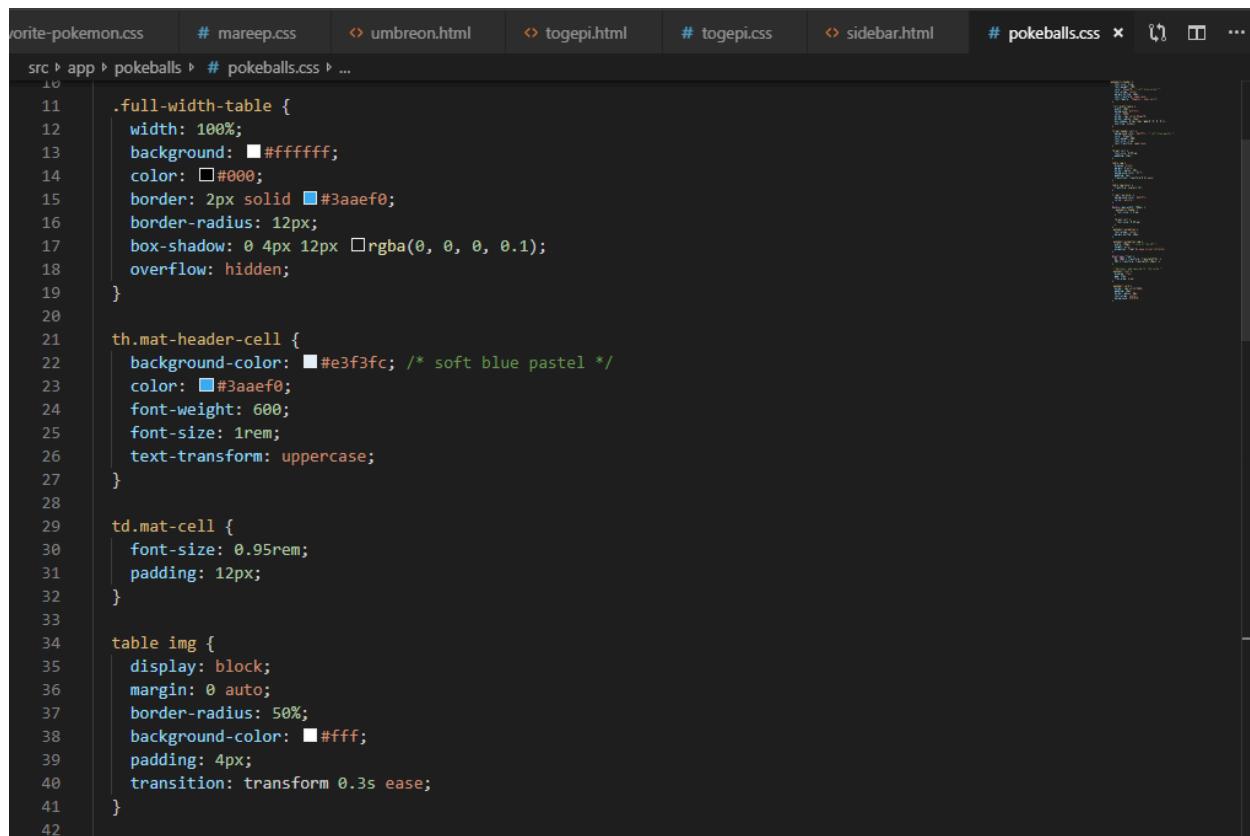
@Injectable({
providedIn: 'root'
})
export class BerriesService {
getBerries() {
return [
{ name: 'Oran Berry', effect: 'Restores 10 HP', image: '/images/berries/oran.jpg' },
{ name: 'Sitrus Berry', effect: 'Restores 25% HP', image: '/images/berries/citrus.jpg' },
{ name: 'Lum Berry', effect: 'Cures any status condition', image: '/images/berries/lum.jpg' },
{ name: 'Cheri Berry', effect: 'Cures paralysis', image: '/images/berries/cheri.jpg' },
{ name: 'Chesto Berry', effect: 'Wakes up a sleeping Pokémon', image: '/images/berries/chesto.jpg' }
];
}
}
```

```
# togepi.css x sidebar.html # pokeballs.css landing.html # landing.css ts berries.ts ...\\service ts pokeballs.ts x ...  
src > app > service > TS pokeballs.ts > ...  
1 import { Injectable } from '@angular/core';  
2  
3 @Injectable({  
4   providedIn: 'root'  
5 })  
6 export class PokeballsService {  
7   getPokeballs() {  
8     return [  
9       { name: 'Poké Ball', type: 'Standard', image: '/images/pokeballs/standard.jpg' },  
10      { name: 'Great Ball', type: 'Improved', image: '/images/pokeballs/great.jpg' },  
11      { name: 'Ultra Ball', type: 'High Performance', image: '/images/pokeballs/ultra.jpg' },  
12      { name: 'Master Ball', type: 'Legendary', image: '/images/pokeballs/master.jpg' },  
13      { name: 'Premier Ball', type: 'Special', image: '/images/pokeballs/premier.jpg' }  
14    ];  
15  }  
16}  
17  
  
togeapi.css x sidebar.html # pokeballs.css landing.html # landing.css ts berries.ts ...\\service ts pokeballs.ts x ...  
src > app > pokeballs > TS pokeballs.ts > ...  
1 import { Component } from '@angular/core';  
2 import { MatTableModule } from '@angular/material/table';  
3 import { PokeballsService } from '../service/pokeballs';  
4  
5 @Component({  
6   selector: 'app-pokeballs',  
7   standalone: true,  
8   imports: [MatTableModule],  
9   templateUrl: './pokeballs.html',  
10  styleUrls: ['./pokeballs.css']  
11})  
12 export class Pokeballs {  
13   dataSource: { name: string; type: string; image: string }[] = [];  
14   displayedColumns: string[] = ['image', 'name', 'type'];  
15  
16   constructor(private p: PokeballsService) {}  
17  
18   ngOnInit(): void {  
19     this.dataSource = this.p.getPokeballs();  
20   }  
21}  
22
```



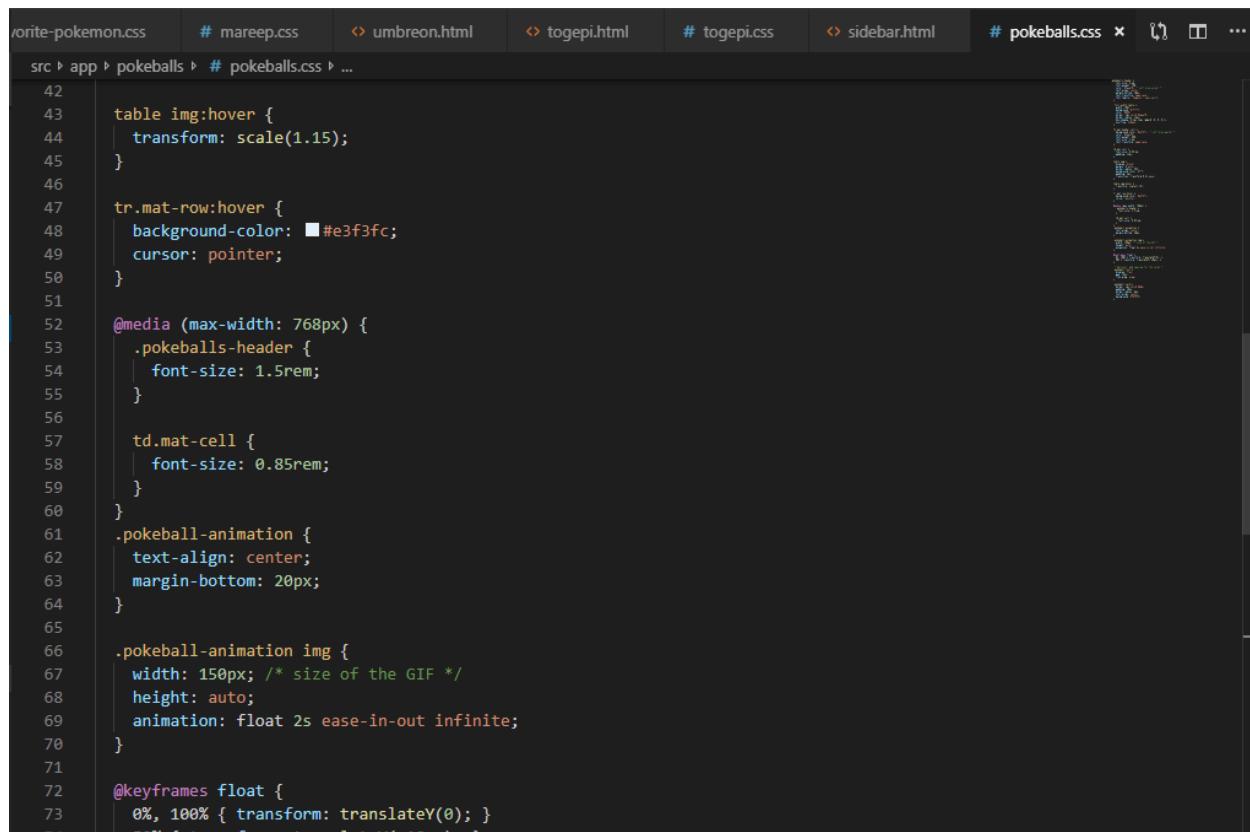
The screenshot shows a code editor with the file 'pokeballs.html' open. The code is an Angular template for a table displaying Poké Balls. It includes a header, three columns ('image', 'name', 'type'), and two rows for header and data. The 'image' column displays a placeholder image for each ball, and the 'name' and 'type' columns show the ball's name and type respectively. The code uses Angular's structural directives (*ng-container) and structural styles (matColumnDef) to define the columns and their data source.

```
<h1 class="pokeballs-header">Poké Balls Collection</h1>
<table mat-table [dataSource]="dataSource" class="mat-elevation-z8 full-width-table">
  <ng-container matColumnDef="image">
    <th mat-header-cell *matHeaderCellDef> Poké Ball </th>
    <td mat-cell *matCellDef="let ball">
      <img [src]="ball.image" alt="{{ ball.name }}" width="40" height="40">
    </td>
  </ng-container>
  <ng-container matColumnDef="name">
    <th mat-header-cell *matHeaderCellDef> Name </th>
    <td mat-cell *matCellDef="let ball">{{ ball.name }}</td>
  </ng-container>
  <ng-container matColumnDef="type">
    <th mat-header-cell *matHeaderCellDef> Type </th>
    <td mat-cell *matCellDef="let ball">{{ ball.type }}</td>
  </ng-container>
  <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr>
  <tr mat-row *matRowDef="let row; columns: displayedColumns"></tr>
</table>
```



The screenshot shows a code editor interface with a dark theme. The top navigation bar includes tabs for 'vorite-pokemon.css', '# mareep.css', 'umbreon.html', 'togeipi.html', '# togeipi.css', 'sidebar.html', '# pokeballs.css', and an ellipsis (...). The main editor area displays the following CSS code:

```
src > app > pokeballs > # pokeballs.css > ...
11 .full-width-table {
12   width: 100%;
13   background: #ffffff;
14   color: #000;
15   border: 2px solid #3aaef0;
16   border-radius: 12px;
17   box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
18   overflow: hidden;
19 }
20
21 th.mat-header-cell {
22   background-color: #e3f3fc; /* soft blue pastel */
23   color: #3aaef0;
24   font-weight: 600;
25   font-size: 1rem;
26   text-transform: uppercase;
27 }
28
29 td.mat-cell {
30   font-size: 0.95rem;
31   padding: 12px;
32 }
33
34 table img {
35   display: block;
36   margin: 0 auto;
37   border-radius: 50%;
38   background-color: #fff;
39   padding: 4px;
40   transition: transform 0.3s ease;
41 }
42 }
```



A screenshot of a code editor showing a CSS file named 'pokeballs.css'. The file contains CSS rules for styling a table and applying a GIF animation to an image element. The code includes media queries for different screen widths and various CSS properties like transform, background-color, and font-size.

```
src > app > pokeballs > # pokeballs.css > ...
42 table img:hover {
43   | transform: scale(1.15);
44 }
45
46 tr.mat-row:hover {
47   | background-color: #e3f3fc;
48   | cursor: pointer;
49 }
50
51 @media (max-width: 768px) {
52   .pokeballs-header {
53     | font-size: 1.5rem;
54   }
55
56   td.mat-cell {
57     | font-size: 0.85rem;
58   }
59 }
60
61 .pokeball-animation {
62   | text-align: center;
63   | margin-bottom: 20px;
64 }
65
66 .pokeball-animation img {
67   | width: 150px; /* size of the GIF */
68   | height: auto;
69   | animation: float 2s ease-in-out infinite;
70 }
71
72 @keyframes float {
73   0%, 100% { transform: translateY(0); }
74   50% { transform: translateY(-15px); }
75 }
```