

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.

WATER POKÉMONFIRE POKÉMONGRASS POKÉMON

**Totodile**



Water

HP: 50  
Attack: 65  
Defense: 64  
Speed: 43

- Revert -

- Evolve -

**Croconaw**



Water

HP: 65  
Attack: 80  
Defense: 80  
Speed: 55

- Revert -

- Evolve -

**Feraligatr**



Water

HP: 85  
Attack: 105  
Defense: 100  
Speed: 78

- Revert -

- Evolve -

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.

The image displays three vertically stacked mobile application screens, each showing a different stage of the Charmander evolution line. Each screen has a blue header bar with three tabs: 'WATER POKEMON' (with a water drop icon), 'FIRE POKEMON' (with a fire icon, which is highlighted in yellow), and 'GRASS POKEMON' (with a leaf icon). The main content area is a light orange gradient background.

**Top Screen (Charmander):**

- Name:** Charmander
- Type:** Fire
- HP:** 39
- Attack:** 52
- Defense:** 43
- Speed:** 65

**Middle Screen (Charmeleon):**

- Name:** Charmeleon
- Type:** Fire
- HP:** 58
- Attack:** 64
- Defense:** 55
- Speed:** 80

**Bottom Screen (Charizard):**

- Name:** Charizard
- Type:** Fire/Flying
- HP:** 78
- Attack:** 84
- Defense:** 75
- Speed:** 100

Each screen also features a 'Reset' button at the bottom left and an 'Evolve' button at the bottom right.

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.

WATER POKÉMON   FIRE POKÉMON   GRASS POKÉMON



**Chikorita**  
  
Grass  
HP: 45  
Attack: 49  
Defense: 65  
Speed: 45  
[Revert](#) [Evolve](#)

WATER POKÉMON   FIRE POKÉMON   GRASS POKÉMON



**Bayleef**  
  
Grass  
HP: 60  
Attack: 62  
Defense: 80  
Speed: 60  
[Revert](#) [Evolve](#)

WATER POKÉMON   FIRE POKÉMON   GRASS POKÉMON



**Meganium**  
  
Grass  
HP: 80  
Attack: 82  
Defense: 100  
Speed: 80  
[Revert](#) [Evolve](#)

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.

The screenshot shows a code editor interface with two tabs open: `fire-pokemon.html` and `fire-pokemon.ts`.

**fire-pokemon.html:**

```
src > app > fire-pokemon > fire-pokemon.html
1  <div class="pokemon-container">
2    <div class="pokemon-card" [class.evolved]="isEvolved">
3      <h2>{{ currentPokemon.name }}</h2>
4
5      <img [src]="currentPokemon.image" [alt]="currentPokemon.name" class="pokemon-img" />
6
7      <div class="type-label">
8        <span *ngFor="let type of currentPokemon.typing">{{ type }}</span>
9      </div>
10
11     <div class="stats">
12       <p>HP: {{ currentPokemon.hp }}</p>
13       <p>Attack: {{ currentPokemon.attack }}</p>
14       <p>Defense: {{ currentPokemon.defense }}</p>
15       <p>Speed: {{ currentPokemon.speed }}</p>
16     </div>
17
18     <button (click)="revert()" [disabled]="currentIndex === 0">← Revert</button>
19     <button (click)="evolve()" [disabled]="currentIndex === pokemonList.length - 1">Evolve →</button>
20   </div>
21 </div>
22
```

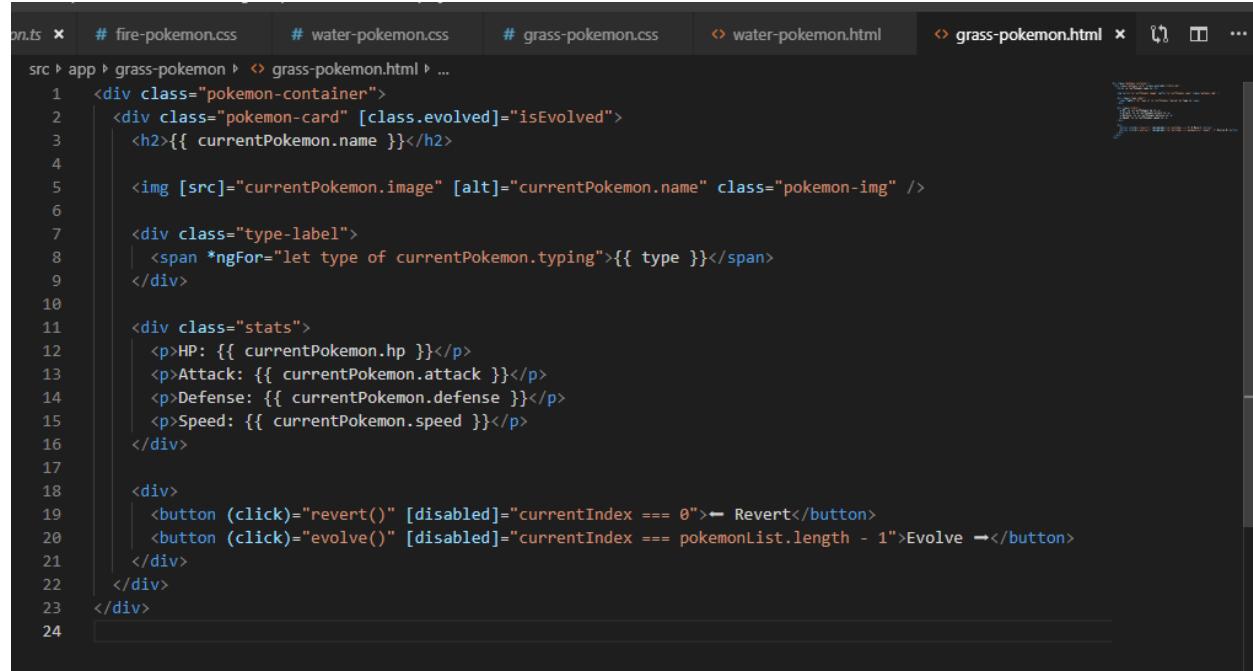
**fire-pokemon.ts:**

```
src > app > fire-pokemon > fire-pokemon.ts
1  import { Component, OnInit } from '@angular/core';
2  import { CommonModule } from '@angular/common';
3  import { FirePokemonService } from '../services/fire-pokemon';
4  import { Pokemon } from '../models/pokemon.model';
5
6  @Component({
7    selector: 'app-fire-pokemon',
8    standalone: true,
9    imports: [CommonModule],
10   templateUrl: './fire-pokemon.html',
11   styleUrls: ['./fire-pokemon.css']
12 })
13 export class FirePokemon implements OnInit {
14   pokemonList: Pokemon[] = [];
15   currentIndex = 0;
16   isEvolved = false;
17
18   constructor(private fireService: FirePokemonService) {}
19
20   ngOnInit(): void {
21     this.pokemonList = this.fireService.getPokemonList();
22     this.updateEvolvedState();
23   }
24
25   get currentPokemon(): Pokemon {
26     return this.pokemonList[this.currentIndex];
27   }
28
29   evolve(): void {
30     if (this.currentIndex < this.pokemonList.length - 1) {
31       this.currentIndex++;
32       this.updateEvolvedState();
33     }
34   }
35 }
```

A tooltip message at the top right of the editor window says: "This workspace has extension recommendations." with buttons for "Install All" and "Show Recommendations".

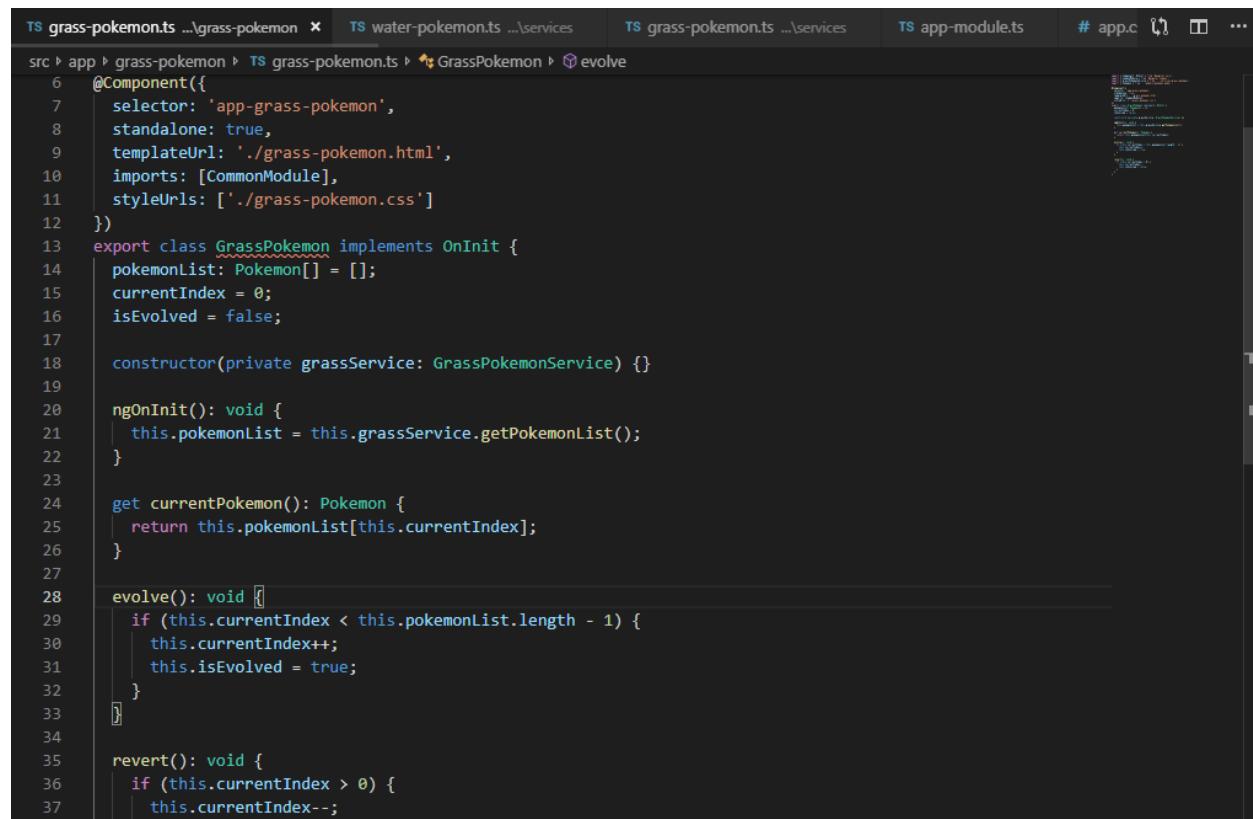
# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.



A screenshot of a code editor showing a component template. The file is named 'grass-pokemon.html'. The template contains a structure for displaying a Pokemon card, including a header with the current Pokemon's name, an image, type information, stats (HP, Attack, Defense, Speed), and buttons for reverting or evolving the Pokemon.

```
src > app > grass-pokemon > grass-pokemon.html > ...
1  <div class="pokemon-container">
2    <div class="pokemon-card" [class.evolved]="isEvolved">
3      <h2>{{ currentPokemon.name }}</h2>
4
5      <img [src]="currentPokemon.image" [alt]="currentPokemon.name" class="pokemon-img" />
6
7      <div class="type-label">
8        <span *ngFor="let type of currentPokemon.typing">{{ type }}</span>
9      </div>
10
11      <div class="stats">
12        <p>HP: {{ currentPokemon.hp }}</p>
13        <p>Attack: {{ currentPokemon.attack }}</p>
14        <p>Defense: {{ currentPokemon.defense }}</p>
15        <p>Speed: {{ currentPokemon.speed }}</p>
16      </div>
17
18      <div>
19        <button (click)="revert()" [disabled]="currentIndex === 0">← Revert</button>
20        <button (click)="evolve()" [disabled]="currentIndex === pokemonList.length - 1">Evolve →</button>
21      </div>
22    </div>
23  </div>
24
```

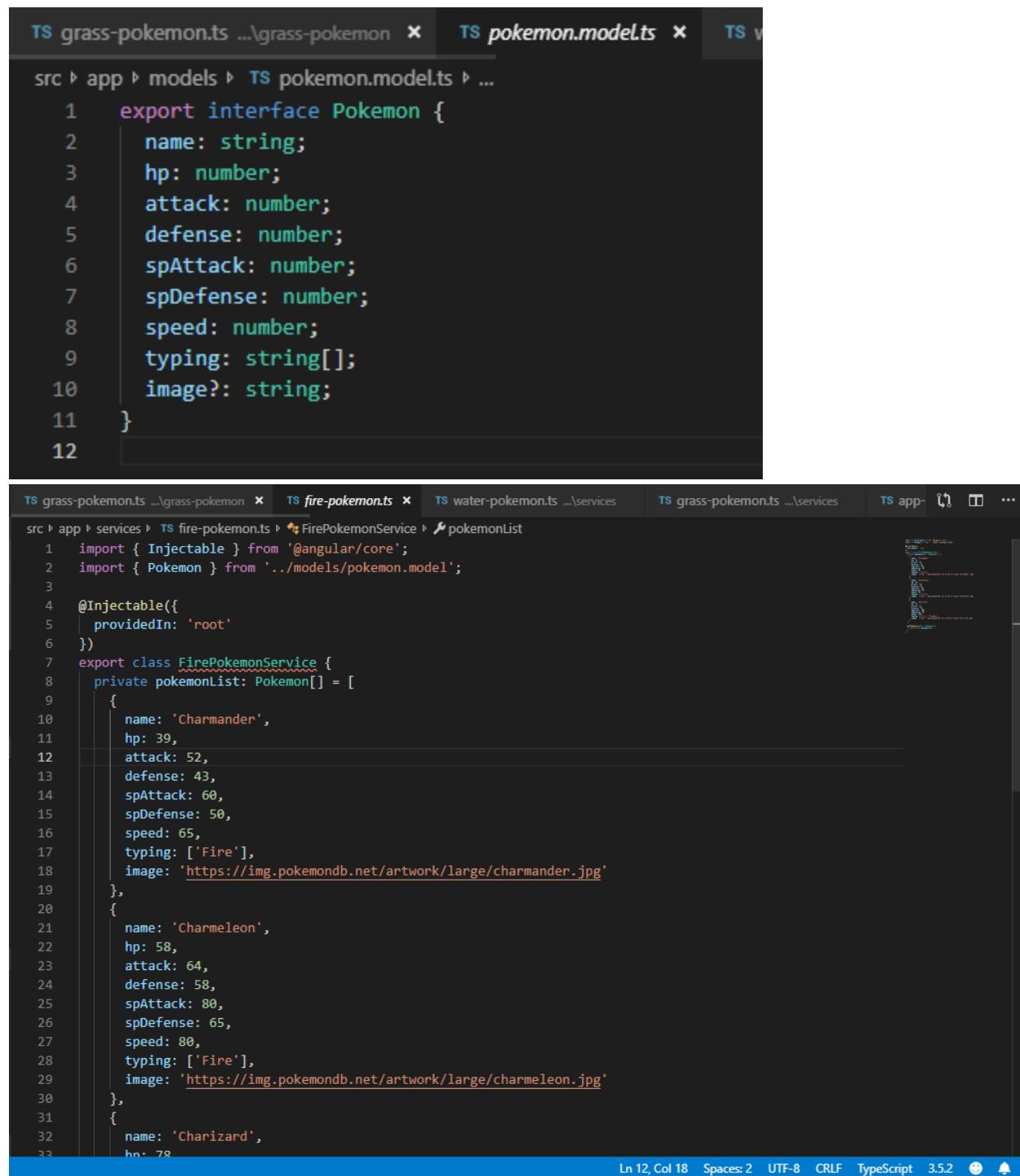


A screenshot of a code editor showing a component class. The file is named 'grass-pokemon.ts'. It defines a component 'app-grass-pokemon' with a template URL pointing to 'grass-pokemon.html'. The class 'GrassPokemon' implements 'OnInit' and has methods for getting the current Pokemon, evolving it, and reverting it.

```
src > app > grass-pokemon > grass-pokemon.ts > GrassPokemon > evolve
6  @Component({
7    selector: 'app-grass-pokemon',
8    standalone: true,
9    templateUrl: './grass-pokemon.html',
10   imports: [CommonModule],
11   styleUrls: ['./grass-pokemon.css']
12 })
13 export class GrassPokemon implements OnInit {
14   pokemonList: Pokemon[] = [];
15   currentIndex = 0;
16   isEvolved = false;
17
18   constructor(private grassService: GrassPokemonService) {}
19
20   ngOnInit(): void {
21     this.pokemonList = this.grassService.getPokemonList();
22   }
23
24   get currentPokemon(): Pokemon {
25     return this.pokemonList[this.currentIndex];
26   }
27
28   evolve(): void {
29     if (this.currentIndex < this.pokemonList.length - 1) {
30       this.currentIndex++;
31       this.isEvolved = true;
32     }
33   }
34
35   revert(): void {
36     if (this.currentIndex > 0) {
37       this.currentIndex--;
```

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.



```
TS grass-pokemon.ts ...\\grass-pokemon ✘ TS pokémon.model.ts ✘ TS w
src \\ app \\ models \\ TS pokémon.model.ts \\ ...
1 export interface Pokémon {
2   name: string;
3   hp: number;
4   attack: number;
5   defense: number;
6   spAttack: number;
7   spDefense: number;
8   speed: number;
9   typing: string[];
10  image?: string;
11 }
12

TS grass-pokemon.ts ...\\grass-pokemon ✘ TS fire-pokemon.ts ✘ TS water-pokemon.ts ...\\services ✘ TS grass-pokemon.ts ...\\services ✘ TS app- ⌂ ⌄ ...
```

```
src \\ app \\ services \\ TS fire-pokemon.ts \\ FirePokemonService \\ pokémonList
1 import { Injectable } from '@angular/core';
2 import { Pokémon } from '../models/pokémon.model';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class FirePokemonService {
8   private pokémonList: Pokémon[] = [
9     {
10       name: 'Charmander',
11       hp: 39,
12       attack: 52,
13       defense: 43,
14       spAttack: 60,
15       spDefense: 50,
16       speed: 65,
17       typing: ['Fire'],
18       image: 'https://img.pokemondb.net/artwork/large/charmander.jpg'
19     },
20     {
21       name: 'Charmeleon',
22       hp: 58,
23       attack: 64,
24       defense: 58,
25       spAttack: 80,
26       spDefense: 65,
27       speed: 80,
28       typing: ['Fire'],
29       image: 'https://img.pokemondb.net/artwork/large/charmeleon.jpg'
30     },
31     {
32       name: 'Charizard',
33       hp: 78,
```

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.

The image displays two screenshots of a code editor interface, likely from Visual Studio Code, showing parts of a Pokemon application. Both screenshots show a dark-themed interface with multiple tabs open at the top.

**Top Screenshot:** The active tab is 'TS fire-pokemon.ts'. The code defines a service named 'FirePokemonService' with a private array 'pokemonList' containing one element: Charizard. The element has properties like name ('Charizard'), hp (78), attack (84), defense (78), spAttack (109), spDefense (85), speed (100), typing ('Fire', 'Flying'), and image URL ('https://img.pokemondb.net/artwork/large/charizard.jpg'). It also includes a method 'getPokemonList()' that returns the array.

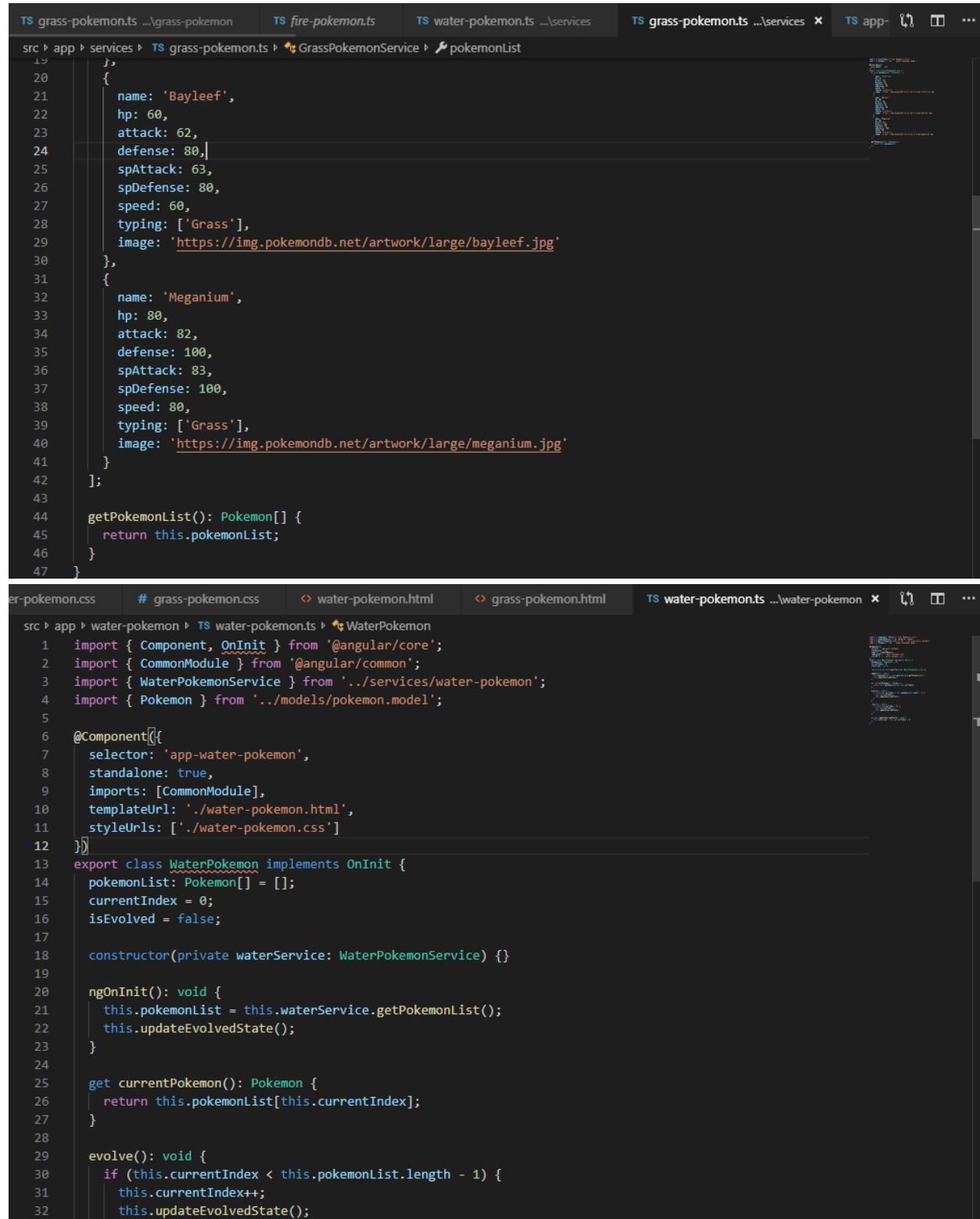
```
30     },
31   {
32     name: 'Charizard',
33     hp: 78,
34     attack: 84,
35     defense: 78,
36     spAttack: 109,
37     spDefense: 85,
38     speed: 100,
39     typing: ['Fire', 'Flying'],
40     image: 'https://img.pokemondb.net/artwork/large/charizard.jpg'
41   }
42 ];
43
44 getPokemonList(): Pokemon[] {
45   return this.pokemonList;
46 }
47 }
48 
```

**Bottom Screenshot:** The active tab is 'TS grass-pokemon.ts'. The code defines a service named 'GrassPokemonService' with a private array 'pokemonList' containing three elements: Chikorita, Bayleef, and Meganium. Each element has properties like name, hp, attack, defense, spAttack, spDefense, speed, typing, and image URL. The 'GrassPokemonService' class also includes a method 'getPokemonList()' that returns the array.

```
1 import { Injectable } from '@angular/core';
2 import { Pokemon } from '../models/pokemon.model';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class GrassPokemonService {
8   private pokemonList: Pokemon[] = [
9     {
10       name: 'Chikorita',
11       hp: 45,
12       attack: 49,
13       defense: 65,
14       spAttack: 49,
15       spDefense: 65,
16       speed: 45,
17       typing: ['Grass'],
18       image: 'https://img.pokemondb.net/artwork/large/chikorita.jpg'
19     },
20     {
21       name: 'Bayleef',
22       hp: 60,
23       attack: 62,
24       defense: 80,
25       spAttack: 63,
26       spDefense: 80,
27       speed: 60,
28       typing: ['Grass'],
29       image: 'https://img.pokemondb.net/artwork/large/bayleef.jpg'
30     },
31     {
32       name: 'Meganium',
33       hp: 80
34     }
35   ]
36 }
```

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.



The screenshot shows a code editor with two tabs open, both displaying TypeScript code related to a Pokemon application.

**Top Tab:** grass-pokemon.ts

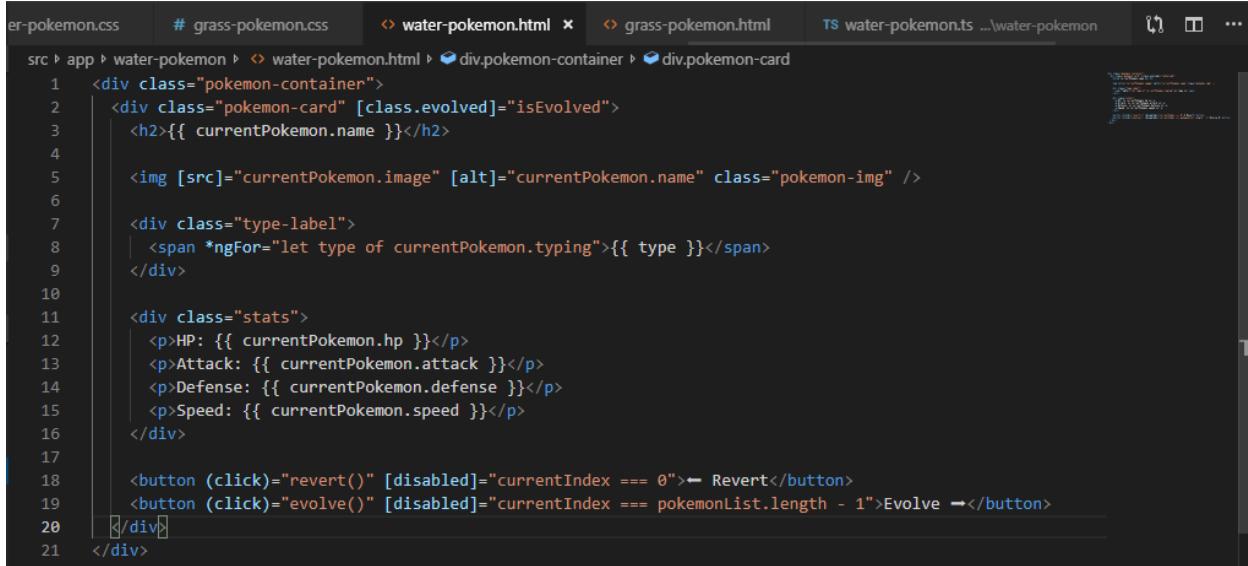
```
19     },
20     {
21       name: 'Bayleef',
22       hp: 60,
23       attack: 62,
24       defense: 80,
25       spAttack: 63,
26       spDefense: 80,
27       speed: 60,
28       typing: ['Grass'],
29       image: 'https://img.pokemondb.net/artwork/large/bayleef.jpg'
30     },
31   {
32     name: 'Meganium',
33     hp: 80,
34     attack: 82,
35     defense: 100,
36     spAttack: 83,
37     spDefense: 100,
38     speed: 80,
39     typing: ['Grass'],
40     image: 'https://img.pokemondb.net/artwork/large/meganium.jpg'
41   }
42 ];
43
44 getPokemonList(): Pokemon[] {
45   return this.pokemonList;
46 }
47 }
```

**Bottom Tab:** water-pokemon.ts

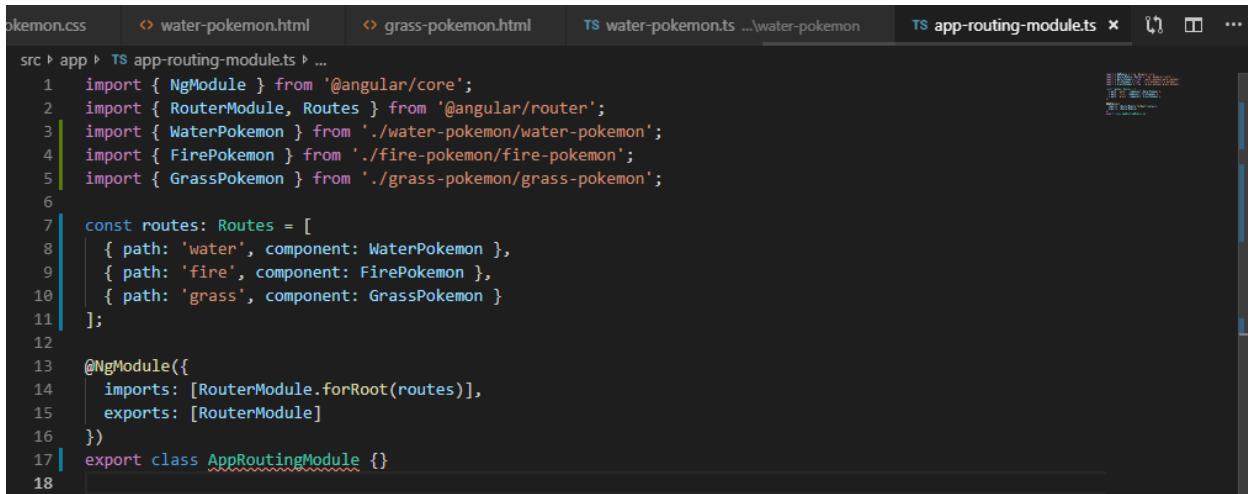
```
1 import { Component, OnInit } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { WaterPokemonService } from '../services/water-pokemon';
4 import { Pokemon } from '../models/pokemon.model';
5
6 @Component({
7   selector: 'app-water-pokemon',
8   standalone: true,
9   imports: [CommonModule],
10  templateUrl: './water-pokemon.html',
11  styleUrls: ['./water-pokemon.css']
12 })
13 export class WaterPokemon implements OnInit {
14   pokemonList: Pokemon[] = [];
15   currentIndex = 0;
16   isEvolved = false;
17
18   constructor(private waterService: WaterPokemonService) {}
19
20   ngOnInit(): void {
21     this.pokemonList = this.waterService.getPokemonList();
22     this.updateEvolvedState();
23   }
24
25   get currentPokemon(): Pokemon {
26     return this.pokemonList[this.currentIndex];
27   }
28
29   evolve(): void {
30     if (this.currentIndex < this.pokemonList.length - 1) {
31       this.currentIndex++;
32       this.updateEvolvedState();
33     }
34   }
35
36   updateEvolvedState() {
37     const evolvedPokemon = this.pokemonList[this.currentIndex];
38     if (evolvedPokemon) {
39       this.isEvolved = true;
40     } else {
41       this.isEvolved = false;
42     }
43   }
44 }
```

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.



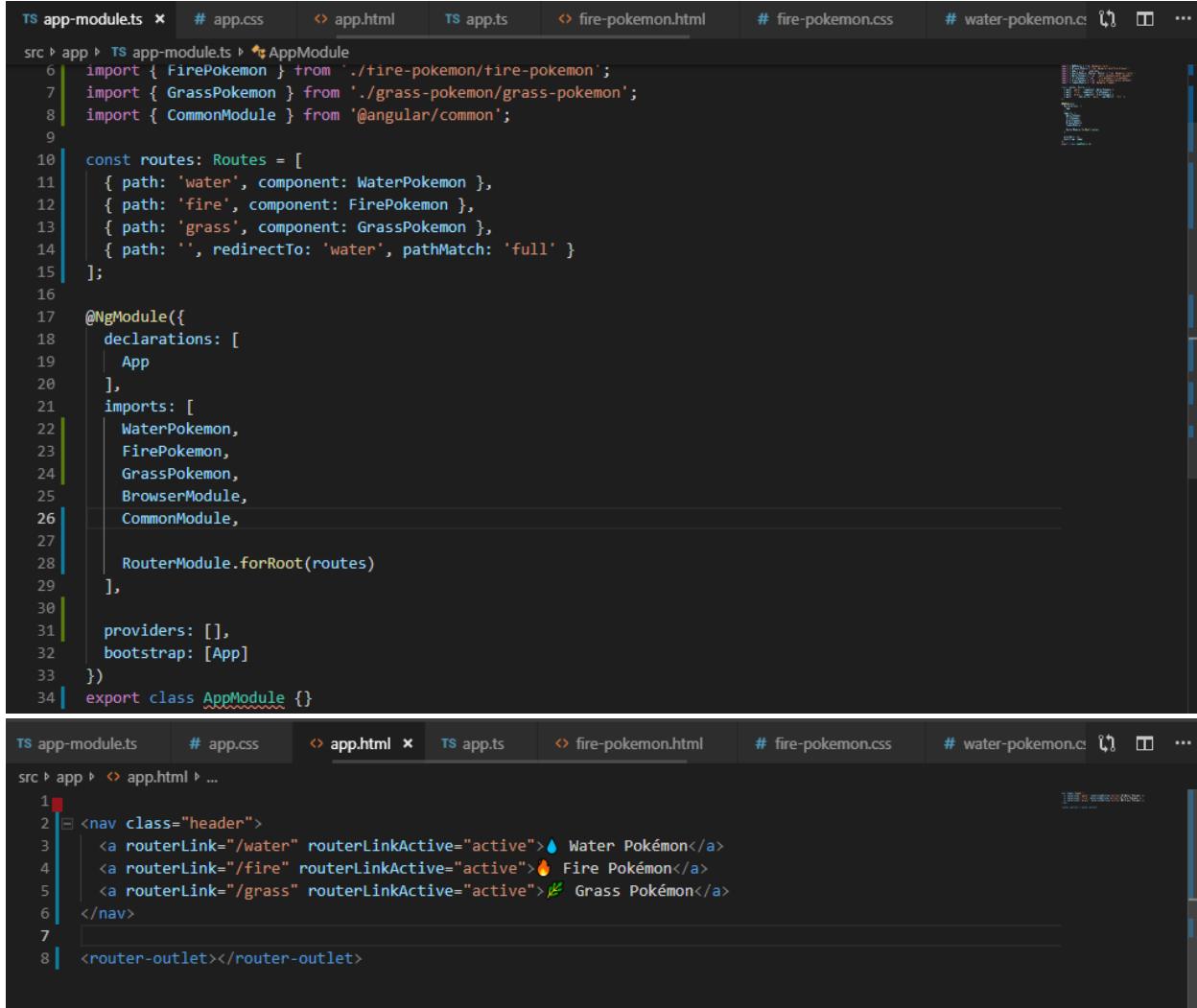
```
src > app > water-pokemon > water-pokemon.html < grass-pokemon.html
1  <div class="pokemon-container">
2    <div class="pokemon-card" [class.evolved]="isEvolved">
3      <h2>{{ currentPokemon.name }}</h2>
4
5      <img [src]="currentPokemon.image" [alt]="currentPokemon.name" class="pokemon-img" />
6
7      <div class="type-label">
8        <span *ngFor="let type of currentPokemon.typing">{{ type }}</span>
9      </div>
10
11      <div class="stats">
12        <p>HP: {{ currentPokemon.hp }}</p>
13        <p>Attack: {{ currentPokemon.attack }}</p>
14        <p>Defense: {{ currentPokemon.defense }}</p>
15        <p>Speed: {{ currentPokemon.speed }}</p>
16      </div>
17
18      <button (click)="revert()" [disabled]="currentIndex === 0">← Revert</button>
19      <button (click)="evolve()" [disabled]="currentIndex === pokemonList.length - 1">Evolve →</button>
20    </div>
21  </div>
```



```
src > app > TS app-routing-module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { WaterPokemon } from './water-pokemon/water-pokemon';
4  import { FirePokemon } from './fire-pokemon/fire-pokemon';
5  import { GrassPokemon } from './grass-pokemon/grass-pokemon';
6
7  const routes: Routes = [
8    { path: 'water', component: WaterPokemon },
9    { path: 'fire', component: FirePokemon },
10   { path: 'grass', component: GrassPokemon }
11 ];
12
13 @NgModule({
14   imports: [RouterModule.forRoot(routes)],
15   exports: [RouterModule]
16 })
17 export class AppRoutingModule {}
```

# APPDEV1 – Introduction to Application Development

BENCILA, ADRIAN MARC C.



The image shows a screenshot of a code editor with two tabs open: `app.module.ts` and `app.html`.

**app.module.ts:**

```
ts app-module.ts # app.css app.html ts app.ts fire-pokemon.html # fire-pokemon.css # water-pokemon.css ...  
src > app > TS app-module.ts > AppModule  
6 import { FirePokemon } from './fire-pokemon/fire-pokemon';  
7 import { GrassPokemon } from './grass-pokemon/grass-pokemon';  
8 import { CommonModule } from '@angular/common';  
9  
10 const routes: Routes = [  
11   { path: 'water', component: WaterPokemon },  
12   { path: 'fire', component: FirePokemon },  
13   { path: 'grass', component: GrassPokemon },  
14   { path: '', redirectTo: 'water', pathMatch: 'full' }  
15 ];  
16  
17 @NgModule({  
18   declarations: [  
19     App  
20   ],  
21   imports: [  
22     WaterPokemon,  
23     FirePokemon,  
24     GrassPokemon,  
25     BrowserModule,  
26     CommonModule,  
27  
28     RouterModule.forRoot(routes)  
29   ],  
30  
31   providers: [],  
32   bootstrap: [App]  
33 })  
34 export class AppModule {}
```

**app.html:**

```
ts app-module.ts # app.css app.html ts app.ts fire-pokemon.html # fire-pokemon.css # water-pokemon.css ...  
src > app > app.html > ...  
1 <nav class="header">  
2   <a routerLink="/water" routerLinkActive="active">💧 Water Pokémon</a>  
3   <a routerLink="/fire" routerLinkActive="active">🔥 Fire Pokémon</a>  
4   <a routerLink="/grass" routerLinkActive="active">🌿 Grass Pokémon</a>  
5 </nav>  
6  
7 <router-outlet></router-outlet>
```