

- 1. Lépés:** elindítjuk a xamppot, és létrehozuk az adatbázist. Az adatbázis kódja az *adatbazis.txt* fájlban van. Felül az *SQL* földre kattintva csak másoljuk be a kódot és indítsuk el.
- 2. Lépés:** a *backend* mappából indítsuk el a cmd-t, és node index-el indítjuk el a szerveret (jelenleg *index.js* a fő fájl neve, de ez bármi lehet)
- 3. Lépés:** a főmappából (*viragbolt_forras*) indítsuk el a cmd-t, nyissuk meg a Visual Studio code-t és hozzunk létre egy angular projektet
 - *code .*
 - *ng new frontend --no-strict --standalone false*
- 4. Lépés:** a frontend mappába belépve hozzuk létre a componenseket és a servicet
 - *cd frontend*
 - *ng generate component components/index*
 - *ng generate component components/aruk*
 - *ng generate component components/megrendeles*
 - *ng generate service services/adatok*
- 5. Lépés:** indítsuk el a projektet
 - *ng serve*
- 6. Lépés:** a visual studio code-ba belépve először másoljuk át a meglévő kódokat a reszponzív weboldalból
 - a nevenincsbt-ből másoljuk ki a *sunflower.jpg*-t és a *fokep.jpg*-t a frontend public mappájába
 - a nevenincsbt *style.css*-t másold bele a frontend/src *styles.css*-be, viszont a *#nyito*-ba a background-image rosszul van meghívva, javítsd át *url("/fokep.jpg")*;
 - a nevenincsbt *.html* fájljait másold át a megfelelő *component.html* fájlba (például: *index.html* – *index.component.html*), de nem kell a teljes kód,

csak az, ami a <body> tagen belül van (a <script> sem kell!). Arra figyelj, hogy minden fájl alján van egy email cím (nevenincsb@gmail.com), aminél a @-ot nem tudja lekezelni a html, cseréld ki @-re

7. Lépés: az *app-routing.module.ts* fájlba hozzuk létre az útvonalakat

```
viragbolt_forras - app-routing.module.ts

1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3
4 import { IndexComponent } from '../components/index/index.component';
5 import { ArukComponent } from '../components/aruk/aruk.component';
6 import { MegrendelesComponent } from '../components/megrendeles/megrendeles.component';
7
8 const routes: Routes = [
9   { path: "", component: IndexComponent },
10  { path: "aruk", component: ArukComponent },
11  { path: "megrendeles", component: MegrendelesComponent}
12 ];
13
14 @NgModule({
15   imports: [RouterModule.forRoot(routes)],
16   exports: [RouterModule]
17 })
18
19 export class AppRoutingModule { }
```

8. Lépés: az *app.component.html* fájlba hozzunk létre egy alap html kódot, amelybe másoljuk bele a bootstrap oldaláról a css fájl meghívását és készítsünk egy <router-outlet> taget a body-n belül.

```
viragbolt_kesz - app.component.html

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha2/dist/css/bootstrap.min.css"
8     rel="stylesheet" integrity="sha384-aFq/bzH65dt+w6FI2ooMVUpc+21e0SRygnTpmBvdBgSdnuTN7QbdgL+OapgHtvPp"
9     crossorigin="anonymous">
10  <title>Document</title>
11 </head>
12
13 <body>
14   <router-outlet></router-outlet>
15 </body>
16
17 </html>
```

9. Lépés: Lépés: a létrehozott componens.html fájlokba módosítsuk az `` tageket, hogy azok a megfelelő fájlokat hívják meg.

- `` → ``
- `` → ``
- `` → ``

10. Lépés: az *app.module.ts* fájlba hívjuk meg a szükséges modulokat

```
viragbolt_kesz - app.module.ts

1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { IndexComponent } from './components/index/index.component';
7 import { ArukComponent } from './components/aruk/aruk.component';
8 import { MegrendelesComponent } from './components/megrendeles/megrendeles.component';
9
10 import { HttpClientModule } from '@angular/common/http';
11 import { CommonModule } from '@angular/common';
12 import { FormsModule } from '@angular/forms';
13
14 @NgModule({
15   declarations: [
16     AppComponent,
17     IndexComponent,
18     ArukComponent,
19     MegrendelesComponent
20   ],
21   imports: [
22     BrowserModule,
23     AppRoutingModule,
24
25     HttpClientModule,
26     CommonModule,
27     FormsModule
28   ],
29   providers: [],
30   bootstrap: [AppComponent]
31 })
32 export class AppModule { }
```

11. Lépés: készítsük el a service-t az *adatok.service.ts* fájlba (services mappán belül)

```
viragbolt_forras - adatok.service.ts

1  import { Injectable } from '@angular/core';
2
3  import { Observable } from 'rxjs';
4  import { HttpClient } from '@angular/common/http';
5
6  @Injectable({
7    providedIn: 'root'
8  })
9  export class AdatokService {
10
11    private url = "http://localhost:3000/api/flowers";
12
13    constructor(private http:HttpClient) { }
14
15    get(): Observable<any>{
16      return this.http.get<any>(this.url);
17    }
18  }
```

12. Lépés: ezután a components/aruk mappában az *aruk.component.ts* kell elkészítenünk a service-t felhasználva

```
viragbolt_forras - aruk.component.ts

1  import { Component } from '@angular/core';
2
3  import { AdatokService } from '../../services/adatok.service';
4
5  @Component({
6    selector: 'app-aruk',
7    standalone: false,
8    templateUrl: './aruk.component.html',
9    styleUrls: ['./aruk.component.css']
10 })
11
12 export class ArukComponent {
13
14   viragok: any[] = [];
15
16   constructor(private adatok:AdatokService) {}
17
18   ngOnInit(){
19     this.lekeres();
20   }
21
22   lekeres(){
23     this.adatok.get().subscribe((adat) => {
24       this.viragok = adat;
25     });
26   }
27 }
```

13. Lépés: utolsó lépésként, csak egy `*ngFor` direktívát használva kell az *aruk.component.html* fájlba meghívni a megfelelő adatokat (a `<h2>Vetőmagjaink:</h2>` után csak egy `<div>` -et hagyjunk meg!)

```
viragbolt_forras - aruk.component.html
1  <main class="container">
2    <div class="row">
3      <h2>Vetőmagjaink:</h2>
4      <div *ngFor="let virag of viragok" class="col-lg-4 mt-4 arukep">
5        <h4>{{virag.nev}}</h4>
6        <a href="megrendeles"></a>
8      </div>
9    </div>
10 </main>
```

Új rész:

14. Lépés: az *adatok.service.ts*-ben készítsünk egy változót, amelyben eltároljuk majd kiválasztott kép id-ját és egy eljárást, ami majd ezt átadja neki

```
viragbolt_kesz - adatok.service.ts

1  import { Injectable } from '@angular/core';
2  import { Observable } from 'rxjs';
3  import { HttpClient } from '@angular/common/http';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class AdatokService {
9
10     private url = "http://localhost:3000/api/flowers";
11     constructor(private http: HttpClient) { }
12
13     get(): Observable<any> {
14         return this.http.get<any>(this.url);
15     }
16
17     id: number = 0;
18
19     id_mentes(kapottid: number) {
20         this.id = kapottid;
21     }
22
23 }
```

15. Lépés: az *aruk.component.ts*-ben ehhez készítsünk egy eljárást, ami a service-nek átadja az id-t és átvált a másik lapra

```
viragbolt_kesz - aruk.component.ts

1  import { Component } from '@angular/core';
2  import { AdatokService } from '../services/adatok.service';
3
4  import { Router } from '@angular/router';
5
6  @Component({
7    selector: 'app-aruk',
8    standalone: false,
9    templateUrl: './aruk.component.html',
10   styleUrls: ['./aruk.component.css']
11 })
12 export class ArukComponent {
13
14   viragok: any[] = [];
15   constructor(private adatok: AdatokService, private router: Router) { }
16
17   ngOnInit() {
18     this.lekeres();
19   }
20
21   lekeres() {
22     this.adatok.get().subscribe((adat) => {
23       this.viragok = adat;
24     });
25   }
26
27   kivalasztott(id: number) {
28     this.adatok.id_mentes(id);
29     this.router.navigate(["megrendeles"]);
30   }
31
32 }
```


16. Lépés: az *aruk.component.html*-ben az `` taget írjuk át egy `div`-re, amelyre hívjuk meg a kiválasztott eljárást gombnyomásra

```
viragbolt_kesz - aruk.component.html

1  <main class="container">
2    <div class="row">
3      <h2>Vetőmagjaink:</h2>
4      <div *ngFor="let virag of viragok" class="col-lg-4 mt-4 arukep">
5        <h4>{{virag.nev}}</h4>
6        <div (click)="kivalasztott(virag.id)">
7          
8        </div>
9      </div>
10   </div>
11 </main>
```

17. Lépés: *adatok.service.ts*-ben hozzuk létre az függvényt a get végpont számára, ami id alapján fogja az adatokat lekérni.

```
viragbolt_kesz - adatok.service.ts

1  import { Injectable } from '@angular/core';
2  import { Observable } from 'rxjs';
3  import { HttpClient } from '@angular/common/http';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class AdatokService {
9
10     private url = "http://localhost:3000/api/flowers";
11     constructor(private http: HttpClient) { }
12
13     get(): Observable<any> {
14         return this.http.get<any>(this.url);
15     }
16
17     id: number = 0;
18
19     id_mentes(kapottid: number) {
20         this.id = kapottid;
21     }
22
23     getid(): Observable<any> {
24         return this.http.get<any>(`${this.url}/${this.id}`);
25     }
26
27 }
```

18. Lépés: ezt használjuk fel a *megrendeles.component.ts*-hez

```
viragbolt_kesz - megrendeles.component.ts

1  import { Component } from '@angular/core';
2  import { AdatokService } from '../services/adatok.service';
3
4  @Component({
5    selector: 'app-megrendeles',
6    standalone: false,
7    templateUrl: './megrendeles.component.html',
8    styleUrls: ['./megrendeles.component.css']
9  })
10 export class MegrendelesComponent {
11
12     viragok: any[] = [];
13     rendelt_darab: number = 0;
14
15     constructor(private adatok: AdatokService) { }
16
17     ngOnInit() {
18         this.lekeres();
19     }
20
21     lekeres() {
22         this.adatok.getid().subscribe((adat) => {
23             this.viragok = adat;
24         });
25     }
26
27 }
```

19. Lépés: *ngFor direktívával jelenítsük meg az adatokat a *megrendeles.component.html*-be. A feladat azt kéri, hogy a mennyiségnél ne lehessen többet megadni a készletnél, így az input max-nak a virág készletével kell megegyeznie.

```
viragbolt_kesz - megrendeles.component.html
1 <main *ngFor="let virag of viragok" class="container">
2   <h2>{{virag.nev}} ({{virag.kategoria.nev}})</h2>
3   <div class="row">
4     <div class="col-md-6">
5       
6     </div>
7     <div class="col-md-6">
8       <p>{{virag.leiras}}</p>
9       <form>
10        <p class="text-center"><span id="ar">Ár: {{virag.ar}}Ft</span></p>
11        <div>
12          <p class="text-center">
13            <label for="mennyiseg">Mennyiség:</label>
14            <input type="number" name="mennyiseg" id="mennyiseg" min="1" max="{{virag.keszlet}}" value="1"/>
15          </p>
16          <p class="text-center">
17            <button class="btn btn-warning btn-lg">Megrendelem</button>
18          </p>
19        </div>
20      </form>
21    </div>
22  </div>
23 </main>
```

20. Lépés: A megrendelésnek csak akkor kell megjelenni, ha van a növényből készleten, ezt egy *ngIf-el tudjuk szabályozni

```
viragbolt_kesz - megrendeles.component.html
1 <form>
2   <p class="text-center"><span id="ar">Ár: {{virag.ar}}Ft</span></p>
3   <div *ngIf="virag.keszlet>0; else nincs">
4     <p class="text-center">
5       <label for="mennyiseg">Mennyiség:</label>
6       <input type="number" name="mennyiseg" id="mennyiseg" min="1" max="{{virag.keszlet}}" value="1" />
7     </p>
8     <p class="text-center">
9       <button class="btn btn-warning btn-lg">Megrendelem</button>
10    </p>
11  </div>
12  <ng-template #nincs>
13    <p class="text-center">Jelenleg nincs a termékből készleten, keresse fel oldalunkat később!</p>
14  </ng-template>
15 </form>
```

21. Lépés: A feladat vége a rendelés funkció, amihez szükséges még egy módosítás a put végponttal, ezt is az *adatok.service.ts*-ben kell megírni

```
viragbolt_kesz - adatok.service.ts

1  import { Injectable } from '@angular/core';
2  import { Observable } from 'rxjs';
3  import { HttpClient } from '@angular/common/http';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class AdatokService {
9
10     private url = "http://localhost:3000/api/flowers";
11     constructor(private http: HttpClient) { }
12
13     get(): Observable<any> {
14         return this.http.get<any>(this.url);
15     }
16
17     id: number = 0;
18
19     id_mentes(kapottid: number) {
20         this.id = kapottid;
21     }
22
23     getid(): Observable<any> {
24         return this.http.get<any>(`${this.url}/${this.id}`);
25     }
26
27     put(id: number, keszlet: any): Observable<any> {
28         return this.http.put<any>(`${this.url}/${id}`, keszlet)
29     }
30
31 }
```

22. Lépés: a *megrendeles.component.ts*-ben létre kell hozni egy rendelés eljárást. Ilyenkor ugye a megkapott adatot ki kell vonni a jelenlegi készletből, majd ugye erre a maradékra kell módosítani a készletet. Mielőtt viszont ezt elindítanánk kell egy if, ami megvizsgálja, hogy nehogymint mínuszba menjen a készlet. Ha minden jó, módosítja az adatot és visszatér az aruk lapra.

```
viragbolt_kesz - megrendeles.component.ts

1  rendeles(id: number) {
2      const virag = this.viragok.find(v => v.id === id);
3
4      const eredeti_keszlet = virag.keszlet;
5
6      const modositott = {
7          keszlet: eredeti_keszlet - this.rendelt_darab
8      };
9      if(modositott.keszlet<0)
10     {
11         alert('Nincs elegendő a raktáron!')
12     }
13     else{
14         this.adatok.put(id, modositott).subscribe({
15             next: () => {
16                 alert("Sikeres rendelés!");
17                 this.router.navigate(["aruk"]);
18             },
19             error: () => {
20                 alert("Nem sikerült a rendelés!");
21             }
22         });
23     }
24 }
```

23. Lépés: Utolsó lépésként, csak annyi kell, hogy ezt az eljárást meghívjuk a megrendelés gombra és az input mezőnek adjuk be, hogy onnan olvassa be az adatot a *megrendeles.component.ts*

```
viragbolt_kesz - megrendeles.component.html
1 <div *ngIf="virag.keszlet>0; else nincs">
2   <p class="text-center">
3     <label for="mennyiseg">Mennyiség:</label>
4     <input type="number" name="mennyiseg" id="mennyiseg" min="1" max="{{virag.keszlet}}"
5       value="1" [(ngModel)]="rendelt_darab" />
6   </p>
7   <p class="text-center">
8     <button class="btn btn-warning btn-lg" (click)="rendeles(virag.id)">Megrendelem</button>
9   </p>
10 </div>
```