

- 1. Lépés:** elindítjuk a xamppot, és létrehozuk az adatbázist. Az adatbázis kódja az *adatbazis.txt* fájlban van. Felül az SQL fülre 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
(`nevenincsbt@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.

9. 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_forras - 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
33 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>
```