Allasinterjú

**frontend**

A frontend Angular 18-ban készült standalone komponensekkel.

A standalone-n azért érezzük fontosnak kifejezni, mert a project struktúra lényegesen változik a két alkalmazás között. Hosszas értekezés után döntöttünk a standalone mellett mert ekkora projectnél ezt tünt célravezetőbbnek.

Alapvetően 3 különböző „fajta” file-t különböztetünk meg **services**, **dtos**, **components**.

**services**:

A services-t igyekeztünk logika alapján szétválasztani. Ez segít az olvashatóságban, megértésben és a skálázhatóságban.

Ezek veszik fel a kapcsolatot a backenddel.

Törekedtünk az 1 component -> 1 service megvalósitására.

**dtos** (Data Transfer Object):

Alapvető céljuk egy egységes mód a kommunikációra.

Gyakorlatilag interface-k amelyek azokat az információkat tartalmazzák amit a backendnek küldünk/fogadunk.

**components**:

Az oldalon 2 fajta component-et különbözetünk meg

commons:

Ezek azok, amelyek több lapra is felkerülnek

Segítenek az egységesítésben, a gyorsabb kódolásban és a skálázhatóságban.

Ezek állnak egy ts és egy css fileból.

Html nem készül hozzájuk mivel, rövid lenne hozzájuk és az átláthatóságban és gondolkozásban segít ez a megoldás.

Css azért készül hozzájuk, mert a rövid html kód ellenére a css sokszor hosszú, ezért az jobb külön file-ban.

rendesek:

Ezek azok, amelyek megjelennek az oldalt, ezek között történik a navigáció.

Ezek állnak ts, html és css fileból.

**components**

commons

**<navbar>**

* home gomb, ha nem home-on van
* Oldalcím
* Ha be van jelentkezve, „sign out” és „profil gomb”
* Ha nincs jelentkezve „sign in”

**<job-card>**

* cég neve
* munka neve
* munka címe
* DtoJobShort-ot tartalmaz
* kattintható: navigálja az adott munka komponensére.

rendesek

**<home>**

* <navbar>
* <job-card>-ok
* Tartalmaz egy DtoJobShort tömböt

**<job application>**

* <navbar>
* **placeholder**

**<profile >**

* <navbar>
* **placeholder**

**(Innen folytatni miután átbeszeltük)**

**dtos**

* DtoJobShort
* **Egy masikat megbeszélni ami a joboknak az osszes adatat tartalmazza**

**services**

job-application

 getJobs(): Observable<List< DtoJobShort>> {

   return Observable<List< DtoJobShort>>

  }

 getJob(jobId : number ): Observable< DtoJob> {

   return Observable< DtoJob>

  }

getAppliedJobs(userId : number): Observable<List< DtoJobShort>> {

   return Observable<List< DtoJobShort >>

  } // returnoli azokat a munkakat amkire a user jelentkezett

getAdvertisedJobs(userId : number): Observable<List< DtoJobShort>> {

   Observable<List< DtoJobShort>>

  } // returnoli azokat a munkakat amiket a cég hirdetett

getUser (userId : number): Observable < DtoUser> {

   return Observable < DtoUser>

  } // returnoli a sima munkavallalok adatait

postNewJob(newJob : DtoJob):{

http.post()

} // egy post, ami a body kuld egy új munkát,ezeket elmenti az adatbazisba.