

Baranya Megyei Szakképzési Centrum
Baranya Megyei SzC Radnóti Miklós Közgazdasági
Technikum

Vizsgaremek

Készítették: Pópe Dániel
és Kapus Benjámin

Pécs

2023

Baranya Megyei Szakképzési Centrum
Baranya Megyei SzC Radnóti Miklós Közgazdasági
Technikum

Szakma megnevezése: Szoftverfejlesztő és –tesztelő

A szakma azonosító száma: 5 0613 12 03

Vizsgaremek

Szakikereső

Készítették: Pópé Dániel
és Kapus Benjámín

Pécs

2023

Tartalomjegyzék

I. Bevezető, a feladat rövid ismertetése	4
II. A felhasználói dokumentáció	5
1. A program általános specifikációja	5
2. Rendszerkövetelmények	5
• Hardver követelmények	6
• Szoftver követelmények	6
3. A program telepítése	6
4. A program használatának a részletes leírása	7
III. A fejlesztői dokumentáció	12
1. Témaválasztás indoklása	12
2. Az alkalmazott fejlesztői eszközök	12
3. Tervezési módszer	12
4. Adatmodell leírása	13
5. Részletes feladat-specifikáció, algoritmusok	17
6. Forráskód	18
7. Tesztelési dokumentáció	18
8. Továbbfejlesztési lehetőségek	18
IV. Összegzés	19

I. Bevezető, a feladat rövid ismertetése

A vizsgaremekhez több projektötletünk volt, viszont a túlzott bonyolultság és a követelmények alapján szépen kiszűrtük az ötleteket és végül erre a témára esett a választásunk.

A mai világban az emberek törekednek a technológiák fejlesztésére, a dolgok praktikusabbá tételére. Ennek egyik hátránya, hogy minden egyre bonyolultabb és összetettebb lesz, ezért mindenhez szükség van szaktudásra vagy speciális eszközökre. A dolgok elhasználoznak, elavulnak, vagy tönkre mennek.

Bizonyára mindannyian, vagy legalábbis a családban, ismerettségi körben előfordult, hogy felfogadtunk valamilyen szakembert. Ezt sok keresés, kevés választási lehetőség és akár több hét, hónap várakozás előzte meg. Emellé természetesen hozzátartozik az is, hogy a szakember milyen munkát végzett. Sajnos gyakran hallunk, megbízhatatlan, munkáját rosszul elvégző emberekről, akik nagyobb kárt okoztak, mint amennyi hasznót. Ezekre a problémákra szeretnénk megoldásként ezt a projektet.

A fejlesztés során rengeteg új ismeretet szereztünk, mint például a kód lefutásakor keletkezett hibák megoldásáról. Emellett több olyan része is volt a projektnek, amivel az iskolában nem találkoztunk és ezért saját magunknak kellett rájönni, megkeresni a megoldást. Összességében egy jó „tanulópénz” volt ez a projekt.

A projekttel kapcsolatban felmerült az, hogy publikus és elérhető legyen mindenki számára az interneten, viszont a tapasztalatlanság és a tőke hiánya miatt arra a döntésre jutottunk, hogy erre nem kerül sor. A projekt viszont teljes mértékben tovább fejleszthető, ötletekre a későbbiekben kitérek.

A feladatok elosztását alapvetően két részre osztottuk: frontend és backend. Az ötletelési, tervezési szakasznál jobbnak találtuk, hogy teljesen együtt dolgozzuk ki az adott feladatot. Az adatbázis-táblák is így jöttek létre. Ezt követően a kinézettel és az azokhoz tartozó, böngészőben megvalósítható funkciókkal Pópe Dániel foglalkozott. Az adatbázis eljárások készítésével és az adatbázis kapcsolat létrehozásával Kapus Benjámin dolgozott. Természetesen végig kommunikáltunk és ahol tudtunk besegítettünk a másoknak.

A projektmunka során több projektszervezési programot is használtunk. Például Trello-t az ötletek rendszerezéséhez, GitHub-ot a projekt tárolására és verziókezelésére.

II. A felhasználói dokumentáció

1. A program általános specifikációja

A program egy online platform, mely lehetővé teszi szakemberek és megrendelők találkozását és kapcsolatba kerülését. A felhasználók különböző szakmákban dolgozó hirdetők között válogathatnak, és bérelhetnek fel szakembert például Ács, Asztalos, Autószerelő, Bádogos, Biztonság technika, Burkoló, Csőhálózat-szerelő, Festő-mázoló-tapétázó, Gáz-szerelő, Hő-és-hangszigetelő, Hűtő-klíma-hőszivattyú, Kazánszerelő, Kertész, Kőműves, Klíma-szerelő, Nyílászáró-szerelő, Szobafestő, Tetőfedő, Villany-szerelő, Víz-szerelő, Vízkútfúró, Vízszigetelő. területen.

A hirdetések tartalmazzák a hirdető által vállalt területeket, ezzel megkönnyítve a felhasználók dolgát a megfelelő szakember kiválasztásában. A hirdetők megadhatják a foglalkozásukat, ezzel új megrendelőket szerezhetnek és bővíthetik ügyfélkörüket.

A platform lehetőséget ad a megrendelőknek arra, hogy értékeljék a hirdetők munkáját, ezáltal segítve a többi felhasználót a megbízható szakemberek kiválasztásában. A megrendelők megbízásokat adhatnak a hirdetőknek, így gyorsabban és hatékonyabban tudják megtalálni a megfelelő szakembert.

Az üzenetküldési funkció lehetővé teszi a felhasználók közötti kommunikációt, így még könnyebbé válik a kapcsolatfelvétel és az együttműködés. A platform továbbá biztonságos és megbízható környezetet kínál mind a szakemberek, mind a megrendelők számára.

Az alkalmazás felhasználóbarát és könnyen használható, így bárki számára elérhető. A program célja, hogy segítsen a szakembereknek megtalálni az ügyfeleiket és a megrendelőknek megbízható és kiváló minőségű szolgáltatást nyújtó szakembereket találni. Az oldal működtetői mindent megtesznek azért, hogy a felhasználók elégedettek legyenek az alkalmazással

2. Rendszerkövetelmények

• *Hardver követelmények*

Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Memory	2 GB minimum, 4 GB recommended		
Screen resolution	1280x1024 or larger		
Application window size	1024x680 or larger		
Internet connection	Required		

• *Szoftver követelmények*

	Windows requirements	Mac requirements	Linux requirements
Operating system	Windows 8 or later	macOS High Sierra 10.13 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

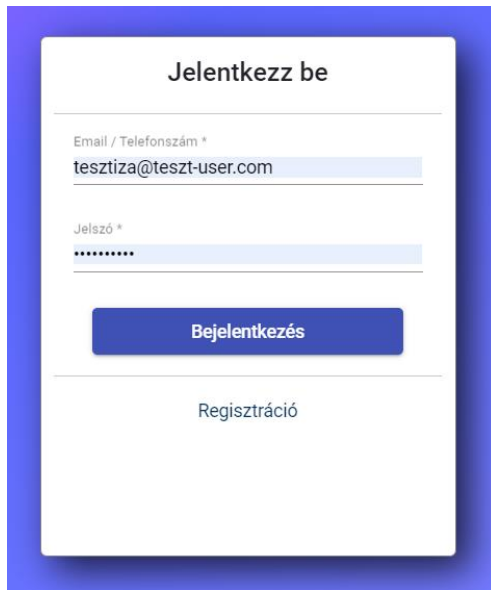
Xampp 7.4.30, Wildfly 26.1.1-Final, JDK 16, JRE 7, Node-js 16.0, Angular 15.0

3. A program telepítése

A program böngészőben fut, nem kell telepíteni semmit sem, csak az előző pontban említett szoftvereket.

4. A program használatának a részletes leírása

Ha először érkezünk az oldalra a bejelentkezési oldal jelenik meg.



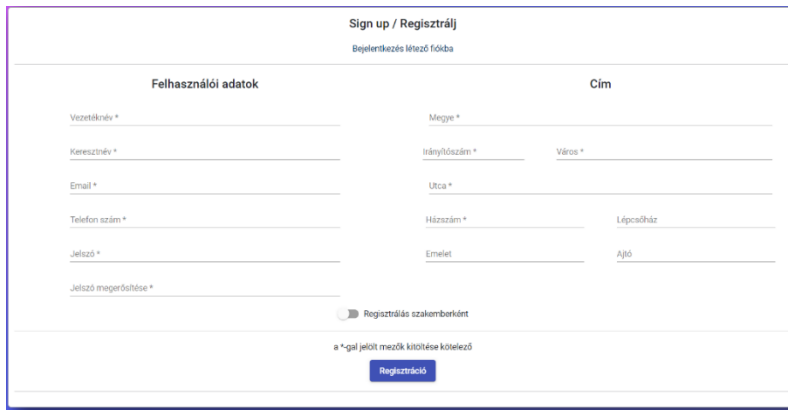
The screenshot shows a login form titled "Jelentkezz be". It has two input fields: "Email / Telefonszám *" with the value "tesztiza@teszt-user.com" and "Jelszó *" with masked characters. Below the fields is a blue button labeled "Bejelentkezés". At the bottom, there is a link labeled "Regisztráció".

1. ábra Bejelentkezési felület

Ha van felhasználói fiók, akkor a felső beviteli mezőbe az email vagy telefonszám, és az alsó beviteli mezőbe a megfelelő jelszó beírásával a bejelentkezés gombra kattintva bejelentkezhetünk. Ha az adatok nem megfelelőek akkor egy sikertelen bejelentkezés feliratú felugró ablak tájékoztat erről. Sikeres bejelentkezés esetében az oldal átnavigál a Főoldalra. Ha nincs felhasználói fiók az oldalhoz akkor a regisztráció felíratra kattintva létrehozható a Form megfelelő kitöltésével, Ha mégis

rendelkezünk aktív felhasználói fiókkal a bejelentkezés létező fiókba szövegre kattintva, visszavigálhatunk a bejelentkezési felületre.

Regisztrációnál kétfajta lehetőség van, létrehozható megrendelői fiók vagy szakember fiók.



The screenshot shows a registration form titled "Sign up / Regisztrálj" with the subtitle "Bejelentkezés létező fiókba". It is divided into two main sections: "Felhasználói adatok" and "Cím". The "Felhasználói adatok" section includes fields for "Vezetéknév *", "Keresztnév *", "Email *", "Telefon szám *", "Jelszó *", and "Jelszó megerősítése *". The "Cím" section includes fields for "Megye *", "Irányítószám *", "Város *", "Ulica *", "Házszám *", "Emelet", "Lépcsőház", and "Ajtó". There is a checkbox labeled "Regisztrálás szakemberként" and a note "a *gal jelölt mezők kitöltése kötelező". At the bottom is a blue button labeled "Regisztráció".

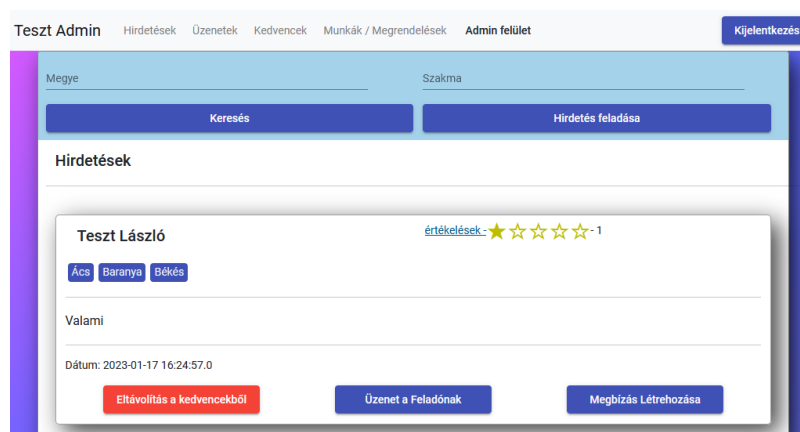
2. ábra Regisztrációs felület

A megrendelői fiók nem tud hirdetéseket létrehozni, és nem fogadhat megrendeléseket. A felhasználónak meg kell adnia a vezetéket, és keresztnévét lehetőleg

200 karakternél nem hosszabb, az e-mail címét, amelyre egy megerősítő link kerül majd kiküldésre, a telefonszámát, egy jelszót amit az oldalhoz való bejelentkezésnél használ, ez legalább 8 karakter hosszúságú kis és nagybetűket, egy szám, és speciális karaktert tartalmaz, a lakhelyének címét a megyét egy listából választhatja ki, a városának irányítószámát (Ez csak szám karakter lehet), városának nevét, utca nevét, és házszámát, emeletes ház esetében lépcsőház, emelet, és ajtó megadása is lehetséges. A típusok között a regisztrálás szakemberként kapcsoló rákattintásával válthatunk. Szakemberi fiók

létrehozásakor, egy 22 foglalkozást tartalmazó listából kiválasztható a foglalkozás. Szakemberi fiók létrehozása esetén lehetőség van Cég regisztrálására is, ez a cég regisztrálása kapcsoló bekapcsolásával érhető el. Itt a cég adatokat beírva, cégnév, adószám (Ez csak szám lehet) és telephely címe kitöltésével tehető meg.

Ha minden adatot helyesen adtunk meg a regisztráció gomb megnyomásával létrehozható a felhasználói fiók. Ha valamilyen adat nem megfelelő akkor az adott beviteli mező alatt jelzi egy piros szöveg a problémát. Ha a regisztráció sikeres akkor a felhasználó által megadott email-címre kiküldésre kerül egy aktivációs link. Az email-ben érkezett kódra rányomva a fiók aktiválásra kerül, és utána be lehet jelentkezni vele. A Főoldalra lépve az oldal tetején a navigációs sáv található. bal felső sarokban a felhasználó neve található, mellette Hirdetések, Üzenetek, Kedvencek, Munkák / Megrendelések, Kijelentkezés, és Admin Felhasználó esetében egy Admin felület, ezekre kattintva navigálhatunk az weblap megfelelő oldalára.



3. ábra Navigációs sáv és a hirdetés oldal

Alap esetben a Hirdetések oldal fogad, ahol lehetőségünk van a feladott hirdetések között böngészni, ilyenkor semmilyen szűrési feltétel nincs ezért az összes hirdetés megjelenik, a hirdetések fölötti

sávban lehetőségünk van szűrési feltételeket beállítani, szűrhetünk megyére, foglalkozásra, vagy mindkettőre, ebben az esetben csak a szűrési feltételeknek megfelelő hirdetések jelennek meg, ha nincs a keresési feltételeknek megfelelő opciókkal rendelkező hirdetés a „Nincs hirdetés ilyen kategóriában” felirat jelenik meg. Szakember esetén egy új hirdetés feladása gombra kattintva lehetősége van feladni egy hirdetést, amivel a többi felhasználó később interakciót végezhet. A hirdetés feladása egy párbeszédablakon történik, ahol meg kell adnia, hogy a szakember, mely foglalkozásához kapcsolódva szeretné létrehozni a hirdetést, mely területekről fogad megrendelést, és egy leírást, amivel szimpatikusabbá teheti magát a megrendelők számára. Egy hirdetés csak akkor jelenik meg, ha egy adminisztrátor jóváhagyta.

A hirdetések tartalmazzák a hirdető nevét, Egy szakmát, amivel a feladó rendelkezik, területeket, ahonnan a hirdető megrendeléseket fogad. Egy értékelési részt, ahol megjelenik, hogy más felhasználók mennyire elégedettek a hirdetés feladójának munkájával vizuális (csillagokkal jelölve) és átlagpont pont értékben (Ha egy hirdető korábban még nem értékelték „nincs” felirat látható). Felhasználóként hozzáadhatunk hirdetések a kedvencekhez, már meglévő kedvencet törölhetünk, üzenetet írhatunk a hirdetés feladójának, amely átnavigál az üzenetek oldalra, vagy megbízást adhatunk számára, ez egy dialógus ablakban történik, ahol a megrendelő leírja milyen ügyben szeretné a hirdető segítségét kérni, ezt elolvasva a szakember könnyebben képes árajánlatot adni. Abban az esetben, ha a hirdetés feladója tekinti meg a hirdetését lehetősége van Törölni, vagy szerkeszteni azt.



4. ábra Értékelés

Az értékelések feliratra kattintva átnavigálásra kerül az Értékelések oldalra. Az értékelések oldalon

megtekinthető, a már korábban a hirdetővel dolgoztató felhasználók értékelése, amin szerepel a felhasználó neve, általa adott pontszám, és az indoklás, hogy segítse a későbbi megrendelőket a hirdető munkájának megbízhatóságában.

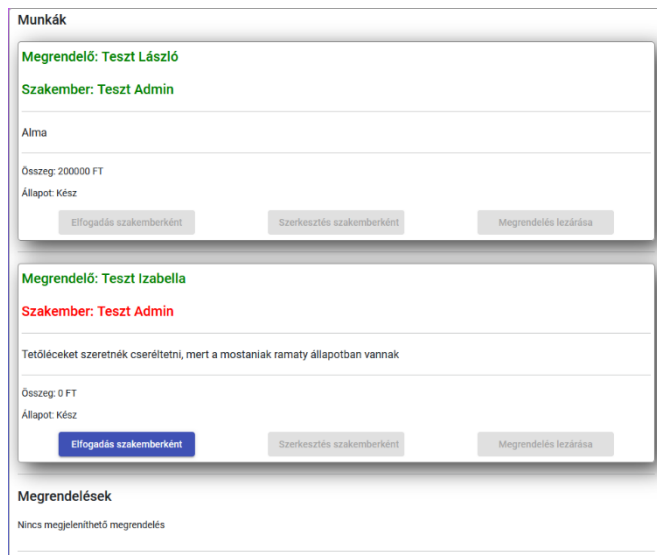
Az üzenetek oldalon a már korábban beszélgetést folytatott felhasználók jelennek meg, itt a megfelelő chatfejre kattintva betöltődnek a felhasználóval váltott korábbi üzenetek, és lehetőség van új üzenet küldésére. Ha egy felhasználótól érkezett olyan üzenet, amit még nem néztünk meg egy piros mezőben lévő szám jelez.



5. ábra Üzenetek oldal

A chateknél az üzenetpartner neve jelenik meg és kattintással válthatuk közöttük, ennek következtében betöltődnek a korábban kapott, valamint küldött szövegek. Üzenetet az üzenet mezőbe való gépeléssel, majd a küldés gomb lenyomásával küldhetünk

A kedvencek oldalon lehetőség van a már korábban kedvencekhez adott hirdetések megtekintése, illetve eltávolítása. Abban az esetben, ha töröltünk egy kedvencet, a hirdetés megmarad az oldal frissítéséig, megelőzve a véletlen eltávolítást.



5. ábra Munkák és megrendelések

ezt a munka elfogadása gombra kattintva tehetik meg. Ha a munka elkészült vagy valamelyik fél nem elégedett lezárhatja a munkát, így a későbbiekben már nem lehet szerkeszteni, és a megrendelőnek lehetősége nyílik értékelni a szakembert az értékelés írása gombra kattintva.

Az értékelő leírhatja a szakemberek kapcsolatos észrevételeit és csillagokkal értékelheti ezzel segítve a többi felhasználót. Egy értékelés akkor jelenik meg, ha egy Admin jóváhagyta.

A Profil oldalon megjelennek a felhasználó különböző adatai, megrendelői felhasználó esetén az elérhetőség, és a regisztrációnál megadott cím. A szerkesztés felirattal az adatok módosíthatóak egy párbeszédablak segítségével, ahova a jelenlegi adataink betöltésre kerülnek, a szerkesztés gombra kattintva az adatok módosulnak és

A munkák és megrendelések oldalon megjelennek más felhasználóknak adott megrendelések, vagy a megrendelőktől kapott munkák. Lehetőség van egy megrendelés szerkesztésére. A megrendelő szerkesztheti a szakembernek adott munka leírását, a Szakember pedig a munka összegét módosíthatja. A munka akkor érvényes, ha mindkét fél elfogadta a másik fél ajánlatát

bezáródik az ablak. Mégse gombra kattintva bezárásra kerül az ablak és a változtatások érvénytelenek lesznek.

Szakember fiók esetében megjelennek a foglakozások, ahol a korábban felvett

6. ábra Szakember Profil oldala

foglalkozás jelenik meg, és a szerkesztés opcióval adható hozzá új szakma, vagy lehet törölni a már meglévőt. A regisztrált cég is megjelenik, ami szintén módosítható, illetve, ha nem regisztráltunk céggel utólag létrehozhatunk a

cég regisztrálása opcióval, illetve törölhetjük a meg lévő cég törlése opcióval. Ha nem vagyunk elégedettek a szolgáltatásokkal a felhasználói fiókot is törölhetjük a fiók törlése opcióval.

Admin felületen 3 opció vehető igénybe Felhasználók, elfogadásra váró hirdetések, és elfogadásra váró értékelések.

A felhasználók oldalon egy táblázatban kilistázásra kerülnek a felhasználók ID, név, email-cím és státusz alapján, az Inaktív felhasználó nem rendelkezik még érvényesített email-címmel, illetve lehet Offline, vagy Online, valamint lehetőség van a felhasználó törlésére

Az elfogadásra váró hirdetések oldalon megjelenik minden olyan hirdetés, amely jelenleg jóváhagyásra vár, és az adminisztrátor elutasíthatja ebben az esetben a hirdetés törlésre kerül, vagy elfogadhatja a hirdetést.

Az elfogadásra váró értékelések oldalon hasonlóképpen megjelennek az elfogadásra váró értékelések és jóváhagyás vagy elutasítás lehetséges.

III. Fejlesztői dokumentáció

A kód írásakor törekedtünk a „clean code” elveinek megfelelni. Ennek fényében a lehető legegyszerűbb, legrövidebb és legjobb megoldásokat alkalmaztuk. A sorok, függvények köré megjegyzéseket is tettünk a még gyorsabb megértés érdekében. Az eredményen látszik, hogy könnyen átlátható és megérthető kódot sikerült készítenünk.

1. Témaválasztás indoklása

A téma kiválasztásának fő szempontjai természetesen a vizsgaremek elvárásai voltak, viszont fontosnak tartottuk, hogy hozzánk közel álló, ismert problémára, feladatra találjunk megoldást. Ezért is sikerült egy életszerű problémára megoldást nyújtó oldalt létrehozni.

Adatbázist is használ, mivel a felhasználók, a hirdetések és sok más adatot el kell tárolnunk. Ezekhez az adatokhoz rengeteg rész tartozik: A regisztrációtól kezdve a hirdetések lekérésén át az értékelések írásáig bármi. Vannak alap és különleges funkciók. Tehát sok lehetőség és elképzelés volt. Ezekre a függvényekre könnyen rálehetett építeni a RESTful komponenseket is.

Ezen kívül volt rendelkezésre álló megjeleníthető tartalom is, mint például a hirdetések, kedvencek, profil adatok. Ezekhez az elemekhez könnyű volt a megjelenítést kialakítani, ami különböző méretű eszközökön is jól látható és áttekinthető.

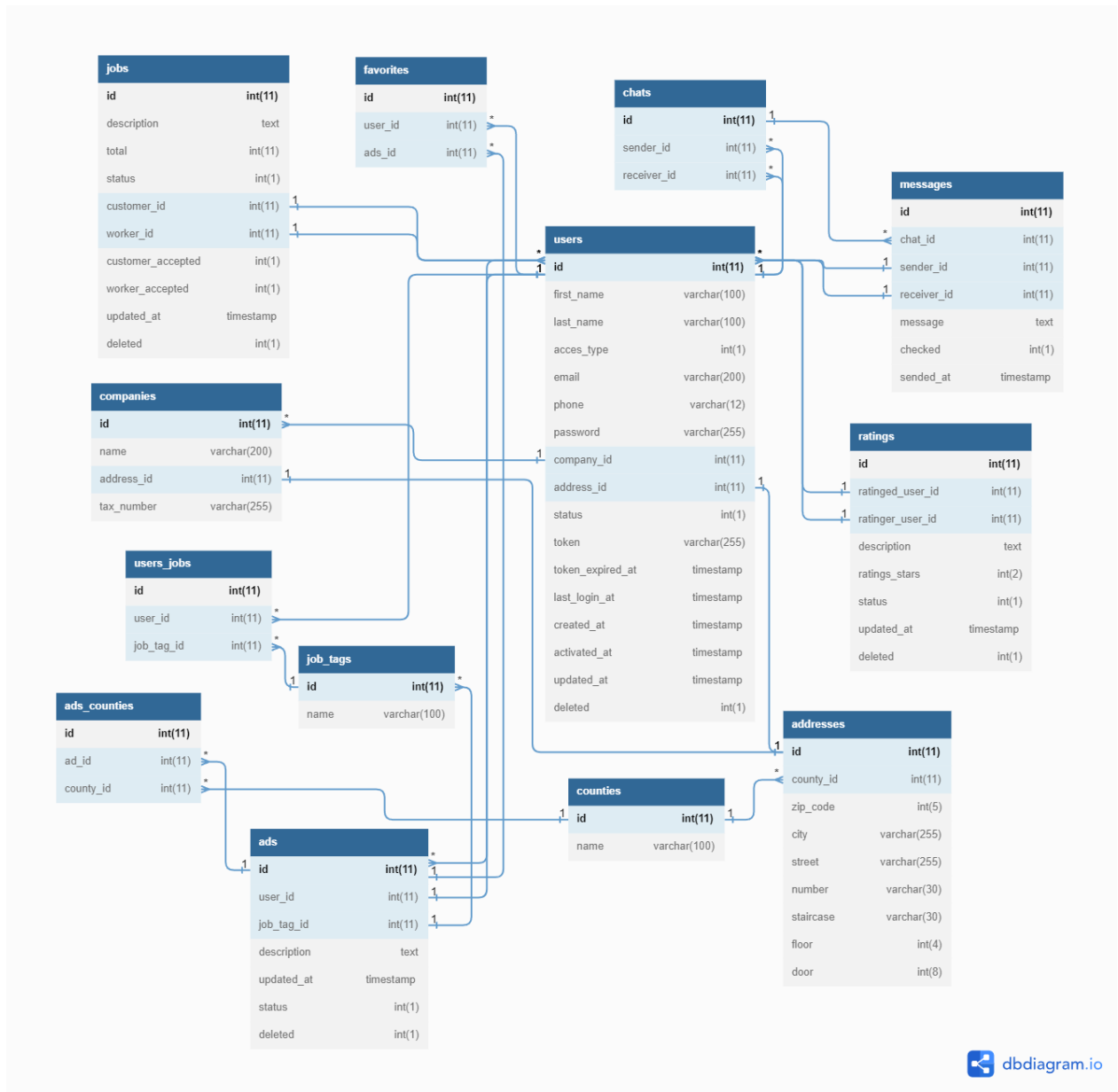
2. Az alkalmazott fejlesztői eszközök

- Programnyelvek
 - Adatbázis: MySQL
 - Frontend: Angular
 - Backend: Java
- Fejlesztői környezet
 - Adatbázis: Xampp (PHPMyAdmin), dbdiagram.io
 - Frontend: Visual Studio Code, PostMan, Paint
 - Backend: Netbeans, PostMan, WildFly szerver
 - Mindegyik: Git, GitHub Desktop

3. Tervezési módszer

4. Adatmodell leírása

Az adatbázis tervezéséhez a dbdiagram nevű, adatbázismodell-diagram készítő oldalt választottuk. Ezzel a diagram szerkesztővel könnyen átalakíthattuk és rendezhettük a táblákat, oszlopokat és kapcsolatokat. Az 1. ábrán látható a teljes adatbázis az adattáblák összekapcsolásával.



7. ábra Adatbázis adatbázismodell-diagramja

Az alábbiakban részletezem az 1. ábra diagramjának a részeit és az adatbázis szerkezetét.

Mindenhol próbáltunk egyértelmű, egyszerű és rövid elnevezéseket használni. Az adatbázist „szakemberkereso”-nek neveztük el, így egyértelmű, hogy ehhez a projekthez

tartozik. A táblák nevei az angol megfelelője annak, amit benne tárolunk (többszámban), mint például a hirdetéseknek „ads” és a felhasználóknak „users”. Amennyiben több a többhöz kapcsolat van az adatok között ott kapcsolótáblát alkalmaztunk. Ezeknek a nevét a két összekapcsolt tábla alkotja egy alsó vonallal elválasztva. Több helyen is előfordulnak azonos oszlopnevek, de csak ha ugyanazt a szerepet töltik be. Úgy gondoltuk, hogy ezeket nem írjuk le külön-külön minden táblához, hanem itt az elején részletezzük őket.

Természetesen mindenhol található elsődleges kulcs, amit „id”-nak neveztünk el. Ez a mező szám típusú és maximum 11 karakter hosszú lehet. A táblák közötti kapcsolatokat, ha egy másik táblában szerepelt idegen kulcsként, általában a tábla neve egyesszámban, egy alsó vonal és az „id” részek alapján neveztük el. Például „ad_id”.

A következő mező a „status”, ami szám típusú és csak 1 karakter lehet. Az adatbázisunknak két formája van:

- A „users” táblában 3 értéke lehet a következő párosításban:
 - „-1”, ha a felhasználó regisztrált az oldalon, viszont a megerősítő E-mail-ben a linket még nem nyitotta meg.
 - „0”, amennyiben az előző pontban említett linket megnyitva aktiválta az E-mail címét, viszont az oldalra ebben a pillanatban nincs bejelentkezve a felhasználó, tehát „offline”.
 - „1” akkor lehet, ha megerősítette a regisztrációját és éppen bejelentkezve van a felhasználó, „online” állapotban van.
 - Ha egyszer megerősítette a regisztrációt, az utolsó két érték között változhat a „status” tulajdonság. Ezt a „login” és „logout” (bejelentkezés - kijelentkezés) tárolt eljárások biztosítják.
- A „ratings”, „ads” táblákban két érték lehet: „0” és „1”. Egy értékelés vagy hirdetés létrehozásakor, vagy módosításakor alapértelmezettként a „0” értékre állítjuk, ami azt jelenti, hogy nincs ellenőrizve, elfogadva. Ezt egy admin jogosultságú felhasználó tudja elfogadni, miután megnézte és nem talált kifogásolható tartalmat benne. Miután elfogadta a „acceptRating” vagy „acceptAd” eljárást meghívva „1”-esre változik az érték.
- A „jobs” táblában bár értékek terén nem változik az előző ponthoz képest, jelentésben egy kicsit eltér. Itt a „0”-hoz a nincs kész az „1”-hez pedig a

kész jelentés van rendelve. A szakember tudja kész állapotba helyezni a „changeJobStatus” procedúrával.

Egy másik gyakran használt tulajdonság a „deleted”. Ennek a szerepe, hogy tudjuk törölve lett-e az adott sor. A „0” a nem törölt, az „1” a törölt.

A „jobs” táblában mindkét fél el tudja fogadni a „szerződést”. „worker_accepted” és „customer_accepted” mezőnevekkel vannak ellátva. Két értéke lehet: „0”, ami azt jelenti, hogy nem fogadta el a feltételeket és az „1” amikor el van fogadva.

Időbélyegeket is használunk, például az „updated_at”, ami timestamp típusú.

Az első és egyben a legfontosabb tábla a „users”. Ebben a felhasználók adatait tároljuk. A nevét két részben, vezetéknévre és keresztnévre lebontva. Mindkettőt szöveg formában tároljuk és legfeljebb 100-100 karakter lehet. Szöveges formában tároljuk még az E-mail címet 200 és a telefonszámot 12 karakterig. A felhasználó jogosultságát az „access_type” tulajdonság alapján kezeli az adatbázis, a backend és a frontend is. 1 karakteres szám formájú lehet: „0” – Általános felhasználó, „1” – Szakember és „2” – Admin. A felhasználó jelszava SHA256-al lett titkosítva, ami ma az egyik legbiztonságosabb titkosítási algoritmus. Ezt szöveggént 255 karakterig tároljuk. A regisztrációhoz és elfelejtett jelszóhoz az adatbázisba eltárolunk egy generált tokent. Ez 255 karakterből állhat. Tartozik ehhez egy lejáratási időbélyeget tartalmazó mező is a „token_expired_at”. Ezeknek alapértelmezettként „NULL” az értékük, viszont, ha a hozzá tartozó funkciót használják, akkor egyedi értékeket kapnak. Az értékek csak megerősítésig, vagy adott ideig elérhetőek. Több időbélyeget is tárolunk, mint például a „last_login_at”, a „created_at” és az „updated_at”.

A „users” tábla „company_id” egy egyed id-ja a „companies” táblából. Az „address_id” pedig az „addresses” tábla egy sorára hivatkozik. Ennek a táblának az id-ja más táblákkal is kapcsolatban van: „jobs - customer_id, - worker_id”, „ads - user_id”, „users_jobs - user_id”, „messages - sender_id, - receiver_id”, „chats - sender_id, - receiver_id”, „ratings - ratinged_user_id, - rater_user_id” és a „favorites - user_id”.

A „jobs” két felhasználó közötti munkaszerződéseket tárolja. Itt található egy „description”, leírás, ami text típusú, egy „total”, összeget jelent és szám. 11 karakter lehet. A többi mezőt már leírtuk korábban és az előző kapcsolatokhoz képest sincs más.

Az „ads” táblában is csak a fent említett tulajdonságok vannak. Egy új kapcsolat viszont van, a „job_tag_id”, ami a „job_tags” tábla id-t jelenti.

A „companies” táblában előkerül néhány új egyedtípus. A „name” szöveges és maximum 200 karakter lehet, ebben a cég nevét tároljuk. Van egy „tax_number”, ami az adószámot jelöli és legfeljebb 255 karakter hosszú szövegből állhat. Kapcsolati téren az „addresses” tábla id mezőjével van összefüggésben.

Az üzeneteket a „messages” adattáblában helyeztük el. Itt a „message” szöveg és a „checked” 1 karakteres szám, ami ha a címzett felhasználó nem látta 0, ha látta 1. Kapcsolatai a következők: „chat_id” a „chats” tábla azonosítójával („id”), „sender_id” és „receiver_id” a „users” „id” tulajdonságát köti össze.

Az értékelések pedig a „ratings”-ben kerültek eltárolásra. Itt egy leírás („description”) szöveggént és egy csillagértékelés („rating_stars”) számként található.

A „counties” és a „job_tags” egyszerű táblák, csak egy 200 karakter hosszú szöveges név („name”) található bennük.

A „users_jobs”, a „chats”, az „ads_counties” és a „favorites” kapcsolótáblák. Több a többhöz összefüggés miatt kellett külön táblákba rendezni. Ezeken a táblákon kívül egy a többhöz kapcsolatot használunk, 1-2 kivétellel.

Az adatbázis tervezésénél minden normalizálási szinten átmentünk, így normalizált állapotban van az adatbázis. Például az „addresses” tábla több adatot tárol, ami nem mindig kell a felhasználó lekéréséhez és így nem is redundáns. A „chats” és „messages” tábla „sender_id” és „receiver_id” mezői direkt szedik meg a normalizálás szabályait, mert így sokkal egyszerűbb az üzenetekhez a felhasználók adatait megjeleníteni. A fejlesztés során az adatbázis szerkezetéhez nem nyúltunk, nem módosítottuk. Pár alkalommal a mezők hosszát változtattuk csak kis mértékben.

Az adatbázis kezelő oldalt jelszóval védjük és csak pár admin jogosultságú felhasználót hoztunk létre. Csak adatbázisból, manuálisan lehet ilyen felhasználókat létrehozni, ez is a biztonságot szolgálja. A jelszó titkosított és a frontenden és a backenden is ellenőrizzük a felhasználó jogosultságát, így szinte lehetetlen bejutni az adatbázisba. A GDPR-t egy felugró ablakban fogadtatjuk el a felhasználóval.

5. Részletes feladat-specifikáció, algoritmusok

A projektben minden függvény megtalálható Angular-ban és Java-ban is a megfelelő formában. Most a backend részét mutatjuk be.

Az első és legalapvetőbb függvény a regisztráció, amit a felhasználók először látnak az oldalon. Ezt „createUser” és „createUserWorker” néven találjuk meg. A kettő nagyon hasonlít egymásra, csak az információk mennyiségében van különbség. Amelyik felhasználó szakemberként regisztrál, a foglalkozását és a cég adatait is megadhatja. Statikusan meghívható és publikus metódusok. Azért ezeket választottuk, mert így a legegyszerűbb, a legkönnyebb és a leggyorsabb. A bemenő paraméter egy „Users” adattáblával megegyező, entitás modell osztály példánya. Először egy 40 karakter hosszú, véletlen karakterekből álló tokenet generálunk. Ezután a jelszót felülírjuk a jelszó titkosított változatával, amit a DigestUtils sha256Hex függvénnyel érünk el. Ezt követően megadjuk egyesével az adatokat és meghívjuk a megfelelő tárolt eljárást. Ha nem futott eddig semmi hibára, akkor meghívunk egy úgynevezett „registrationEmail” nevű függvényt, ami a megadott E-mail címre küld egy linket a fiók aktiválásához. A procedúrából visszakapott eredményt, ami a felhasználó id-ja, átalakítjuk Integer típusúvá és visszaadjuk.

Egy másik fontos függvény a „createAd”, ami hirdetést hoz létre. Szerkezetileg nagyon hasonlít a regisztrációhoz, de pár dolgot módosítanunk kellett. Bemenő paraméterként „Ads” entitás osztályból létrehozott Object-et kap meg. Itt is átadjuk az adatokat és lefuttatjuk a MySQL parancsot eljárás formájában. Az adatokat ellenőrizzük is, hogy az adatbázisba beszúrt új rekordban ne legyenek rossz sorra való hivatkozások. Erre egy példa a 8. ábrán látható sor, ahol a „JobTags” osztályban lévő metódust meghívva azonosító alapján visszaadja a rekordot, ha talált olyan adatsort.

```
spq.setParameter("job_tag_id_in", JobTags.getJobTagById(ad.getJobTagId()).getId());
```

8. ábra Példa Java kódban adat ellenőrzésre

A „createRating” az értékelések írásához készített függvény. Ez is követi az előző szerkezeteket. „Ratings” entitás osztály példánya a bemenő paramétere. Az adatok beírása után ezt az eljárást is lefuttatjuk, viszont ennek nincs eredménye. A függvényt ezért void-ra állítottuk és nincs is visszatérési értéke.

Mindegyik függvényben van hibakezelés és hibadobás is, azért hogy ne kapjon a felhasználó számára értelmezhetetlen hibákat.

6. Forráskód

7. Tesztelési dokumentáció

8. Továbbfejlesztési lehetőségek

A projekt tervezésénél és a megvalósítás közben is számos ötlet született, amik a jövőben a program fejlesztéséhez hozzájárulhatnak. Volt, amelyiket meg szerettük volna valósítani, de közös megegyezés alapján végül nem került bele ez a funkció.

Ilyen volt például a szakember profiljában a referencia munkáknak a képként való feltöltése. Ezeket a képeket akár a hirdetésekhez is odalehetne tenni, így a felhasználók a keresés közben egyből láthatnák, milyen munkát végez az adott szakember. Természetesen nem az összes képet, hanem csak egy pár darabot a szemléltetés miatt. Ha a felhasználó szeretné megnézni a többi referenciát is, itt a hirdetésnél lehetne egy gomb, hivatkozás, ami a szakember profiljához visz és ott az összes képet végig lehetne nézni. Terveztünk ehhez a funkcióhoz egy biztonsági részt is, amely annyiból állt volna, hogy a feltöltött képek átmennének egy vizsgálaton, amit egy admin jogosultságú felhasználó végezne. A vizsgálat lényege, hogy valóban témához, munkához kapcsolódó képek kerüljenek feltöltésre. Ilyen szűrés egyébként a projektben más résznél is megtalálható, például az értékeléseknél. Ott a szöveget ellenőrizzük, hogy kulturált és értelmes megjegyzések kerüljenek fel az oldalra és az értékelt szakember arculatába.

Egy másik feladat lehetne az, hogy a cégeket még mélyebben, komolyabban vegyük. Gondolunk itt arra, hogy több adatot kéne megadni, több ember is tartozhatna egy céghez. Akár úgy is megvalósítható lehetne ez a rész, hogy egy új jogosultság kerülne be a rendszerbe, egy úgynevezett „cég” felhasználó. Ez a felhasználó lehetne a cég tulajdonosa, igazgatója, vezetője vagy akár egy megbízott alkalmazott is. A legtöbb dolgot használhatná az oldalon. Például adhatna fel komplett hirdetést, amiben akár egy egész munkafolyamatot elvállalhatnának (építkezés: betonozás, szigetelés, berendezés...) és ehhez több alkalmazott és munkakör is tartozhatna. Valamint az alkalmazott szakembereket akár a vállalkozáson keresztül is lehetne regisztrálni, így ha új alkalmazottat vesz fel a cég, egyből regisztrálhatná az oldalon is. Ez a rész akár a támogatáson és a használaton keresztül a jövedelem és a fenntartás költségeit is állhatná.

A harmadik fejlesztési lehetőség lenne a legösszetettebb, legbonyolultabb és leghosszabb. Ez az ötlet az oldalon való fizetést tenné lehetővé. Tehát a munka befejeztével a teljes összeget, előleget vagy akár közben, több részletben való kifizetést is lehetővé tenné. Természetesen ehhez több biztonsági lépést is megkérné lépni, de ez sok átverést is kiküszöbölne. Például előleg kifizetése után eltűnik a szakember, vagy éppen a munka megvalósítását követően a megrendelő nem fizet a nem megfelelő munkavégzésre hivatkozva. A fizetéseket nyomon lehetne követni, így a „nem jött meg a pénz”, vagy a „rossz számlára utalta” problémákat is meg lehetne oldani. Emellett a munkaszerződésben lekérné fektetni a fizetéstől való elállásnak a feltételeit is. Készpénzes fizetésnél pedig annyival egyszerűbb lenne ki, hogy miután megtörtént a fizetés, egy tableten, vagy más eszközön aláírná mindkét fél és ezzel bizonyítanák, hogy megtörtént a megbeszéltek összeg kifizetése.

IV. Összegzés

Szerintem elmondhatom, hogy sok új dologgal találkoztunk, megismertük a fejlesztés különböző részeit és ezek által fejlődöttünk. Mivel csapatban kellett dolgozni, így a kommunikációs készségeink is fejlődtek és a feladatok felosztásában is szereztünk némi gyakorlatot, ami a jövőben a hasznunkra válik. A témát szerencsére ismertük és nem is nehéz, ezért inkább a „hogyan csináljuk” kérdés okozott fennakadást. Persze ezen is gyorsan tovább leptünk és így egy nagyon jó weboldalt sikerült készítenünk.

Irodalomjegyzék, forrásmegjelölés

<https://material.angular.io/> - Gomb, Input, Legördülő menü, Ikon, Táblázat, Párbeszédablak, Kapcsoló.

<https://cssgenerator.org/box-shadow-css-generator.html>

<https://cssgenerator.org/gradient-css-generator.html>

<https://getbootstrap.com/docs/4.0/components/navbar/>

<https://www.baeldung.com/jax-rs-spec-and-implementations>

<https://www.baeldung.com/cors-in-jax-rs>