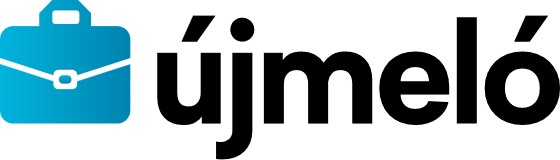
|  |  |
| --- | --- |
| A képen embléma, szöveg, szimbólum, Védjegy látható  Automatikusan generált leírás | BMSzC Bolyai János Műszaki Technikum és Kollégium |



Vizsgaremek  
dokumentáció

Készítette:  
Major Dominik Ábel

Budapest, 2024

Tartalomjegyzék

[I. Bevezetés 5](#_Toc164275248)

[1. Téma választás 5](#_Toc164275249)

[II. Fejlesztői dokumentáció 5](#_Toc164275250)

[2. Fejlesztői környezet 5](#_Toc164275251)

[a) Visual Studio Code 5](#_Toc164275252)

[b) Git (GitHub) 5](#_Toc164275253)

[c) XXAMP 6](#_Toc164275254)

[d) PhpMyAdmin 6](#_Toc164275255)

[3. Fejlesztői hardver 6](#_Toc164275256)

[4. Programozási nyelvek 6](#_Toc164275257)

[5. Könyvtárszerkezet 7](#_Toc164275258)

[a) Frontend 7](#_Toc164275259)

[b) Backend 7](#_Toc164275260)

[6. Látványtervek 8](#_Toc164275261)

[a) Látványterv 9](#_Toc164275262)

[b) Élő kép 9](#_Toc164275263)

[c) Konklúzió 10](#_Toc164275264)

[7. Adatbázis 10](#_Toc164275265)

[a) „users” tábla 11](#_Toc164275266)

[b) „companies” tábla 12](#_Toc164275267)

[c) „category” tábla 12](#_Toc164275268)

[d) „locations” tábla 12](#_Toc164275269)

[e) „applications” tábla 13](#_Toc164275270)

[f) „advertisement” tábla 13](#_Toc164275271)

[8. Legfontosabb algoritmusok 14](#_Toc164275272)

[a) Regisztráció menete 14](#_Toc164275273)

[b) Bejelentkezés menete 15](#_Toc164275274)

[c) Álláshirdetés létrehozásának menete 15](#_Toc164275275)

[d) Dinamikus órabér képzés 16](#_Toc164275276)

[9. Tesztelési dokumentáció 16](#_Toc164275277)

[a) Bejelentkezés 16](#_Toc164275278)

[b) Regisztráció 17](#_Toc164275279)

[c) Új álláshirdetés létrehozása 17](#_Toc164275280)

[d) Álláshirdetés törlése 19](#_Toc164275281)

[e) Kezdőlap 19](#_Toc164275282)

[a) Végpontok tesztelése 20](#_Toc164275283)

[10. Továbbfejlesztés 20](#_Toc164275284)

[a) Egyedi jelentkezés minden álláshirdetéshez 20](#_Toc164275285)

[b) Admin felhasználó 20](#_Toc164275286)

[c) Jelszó frissítése 21](#_Toc164275287)

[III. felhasználói dokumentáció 22](#_Toc164275288)

[1. Kezdőlap 22](#_Toc164275289)

[2. Álláshirdetés oldal 23](#_Toc164275290)

[3. Regisztráció 24](#_Toc164275291)

[4. Bejelentkezés 25](#_Toc164275292)

[5. Vezérlő panel 26](#_Toc164275293)

[a) Profil 26](#_Toc164275294)

[b) Álláshirdetések 27](#_Toc164275295)

[c) Jelentkezők 28](#_Toc164275296)

[d) Álláshirdetés szerkesztése 29](#_Toc164275297)

# Bevezetés

## Téma választás

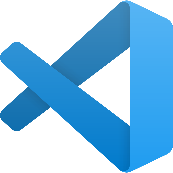
Témámnak egy álláshirdetési platformot elkészítését választottam. Azért ezt a témát választottam mert egy ilyen típusú weboldalhoz minden webfejlesztési képesség szükséges a tervezéstől az átadásig. Sokat böngésztem ilyen oldalakat, hogy inspirációt szerezzek, és hogy megfigyeljem mik a legfontosabb elemek az ilyen típusú weboldalakon.

# Fejlesztői dokumentáció

## Fejlesztői környezet

Számomra nagyon fontos, hogy olyan fejlesztői környezetet teremtsek magam körül, amiben kényelmes a kódolás. Ezáltal már egy kiforrott környezetben tudok fejleszteni.

### Visual Studio Code

A Visual Studio Code egy ingyenesen elérhető, nyílt forráskódú fejlesztői környezet, amelyet a Microsoft fejlesztett ki. Számomra a program legfőbb előnye, hogy rengeteg kiegészítő tartozik hozzá, amivel végteleníthetőek a funkciók. Sokat használtam a beépített terminál funkcióját is, amivel sokkal könnyebben dolgoztam együtt például a git-tel.

### Git (GitHub)

A képen Közúti jelzőtábla, szimbólum, tábla látható

Automatikusan generált leírásA git egy verziókezelő technológia, aminek segítségével parancsok segítésével tölthetjük fel a projekt fájljait valamilyen távoli szerverre. Én a GitHub-ot használtam vele. Azért volt ez nagyon hasznos hiszen két gépen dolgoztam az otthoni asztali PC-men, illetve a laptopomon így minden részfeladat után feltöltöttem a távoli szerverre a jelenlegi állapotot és a másik gépen fejlesztés előtt pedig letöltöttem a legfrissebb állapotot. És ott folytathattam, ahol előzőleg abbahagytam.

### XXAMP

A képen szimbólum, Grafika, clipart, rajzfilm látható

Automatikusan generált leírásA XXAMP egy nyíltforráskódú fejlesztői eszköz, aminek segítségével lokálisan tudtam futtatni az adatbázisomat így sokkal hatékonyabb és biztonságosabban zajlott a fejlesztés mintha egy élő adatbázison lett volna.

### A képen közlekedés, vízijármű, vitorlás hajó, hajó látható Automatikusan generált leírásPhpMyAdmin

A PhpMyAdmin-t használtam az adatbázisom eléréséhez és szerkesztéséhez. Itt teszteltem minden „INSERT INTO” és „SELECT” utasítást, amit később implementáltam a programomba.

## Fejlesztői hardver

A fejlesztést két eszközről végeztem, a laptopomról és az asztali gépemről. Ezeknek a hardvere a következő:

|  |  |  |
| --- | --- | --- |
|  | **Laptop** | **Asztali gép** |
| **Processzor** | I7-11850H | I5-7400K |
| **Memória** | 32GB | 16GB |
| **Videókártya** | RTX A2000 | GTX 1060 3GB |

## Programozási nyelvek

A programozási nyelvek tekintetében a domináló a Javascript, aminek különböző formáit vettem be a fejlesztés során. A backend programozást NodeJS-ben végeztem míg a Frontend programozást natív Javascript-ben. Emellett természetesen HTML-t és CSS-t. Az elosztás nagyjából ilyen:

|  |  |
| --- | --- |
| **JavaScript** | 40% |
| **HTML** | 30% |
| **CSS** | 30% |

## Könyvtárszerkezet

A könyvtárszerkezetem két fő mappából épül fel a „frontend” és a „backend” mappa. Erre azért van szükség, hogy jól elkülöníthetőek legyenek a külön fajta kódok.

### Frontend

A frontend mappán belül két mappa van az „assets” és a „views”. Az „assets” mappában van az összes css fájl, javascript fájl és a képek, ezeknek megfelelően „css”, „js” és „images” mappa. A „js” mappán belül „connection” és „function” mappa. A „function” mappa tárol minden olyan „js” kiterjesztésű fájlt, amiben olyan funkció van írva, amit több oldalon is használok. A „connection” mappában olyan javascript fájlok vannak, amik kapcsolatot teremtenek a „backend”-del. Az „assets”-en belül van még egy „uploads” mappa, ami tartalmazza az összes olyan fájlt, amit a felhasználó feltölthet ezért külön szortírozva három mappára „ml”, „cv” és „logo”. A másik fő mappa az „assets”-en kívül a „views”. Ebben a mappában van az összes „html” fájl és azon belül is külön vannak választva az authentikációhoz szükséges „html” fájlok és a kezelőfelülethez szükséges fájlok.

### Backend

A backend mappa legfőbb fájlja a „server.js”, ez tartalmazza az egész backendnek a magját, vagy ha úgy nézzék ez az elosztó. Külön fájlban van még a „db.js” és a „jwt.js” A „db.js” tartalmazza az adatbázishoz való csatlakozást. A „jwt.js”-ben van két függvény, amely ellenőrzi, illetve létrehozza az authentikációhoz szükséges kulcsot. Itt van még a .env kiterjesztésű fájl, amely a környezeti változókat tartalmazza. A backend egyik legfontosabb mappája a „routes”, amely az összes végpontot kezelő fájlt tartalmazza.

## Látványtervek

A frontend fejlesztés megkezdését egy hosszadalmas látványterv készítés előzte meg. A látványterveket a Figma nevű programban készítettem el. Mindig 100%-os tervet csinálok így a hibák 90%-a már a tervezés alatt kibukik. Minden egyes oldal minden egyes állapotát megterveztem asztali és mobil nézetben. Ha több lett volna az idő akkor természetesen több breakpoint-ra lett volna szükség. Még a látványtervezés előtt építettem fel az újmeló arculat alapjait, azaz a színeket, betűtípusokat. A logót is én készítettem Adobe Illustrator-ban.

Logó két változatban:



Itt látható, hogy mennyi látványterv is készült:

A képen képernyőkép, Multimédiás szoftver látható

Automatikusan generált leírás

Most pedig bemutatnám a kezdőlapot először látványterven aztán pedig élőben:

### Látványterv

A képen szöveg, Webhely, Weblap, Online hirdetés látható

Automatikusan generált leírás

### Élő kép

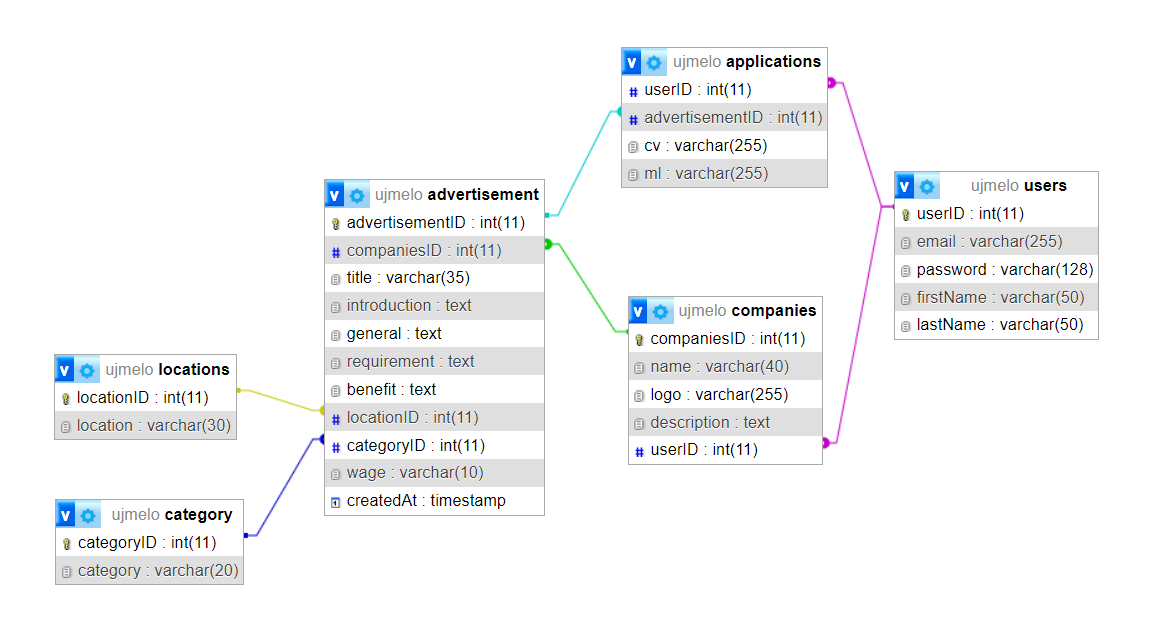
A képen szöveg, képernyőkép, szoftver, Webhely látható

Automatikusan generált leírás

### Konklúzió

Látható, hogy a két kép megegyezik, csupán a háttérnek a színei mások, ez azért van, mivel ez egy folyamatosan változó színátmenetes háttér. Rengeteg előnye van, ha a fejlesztés előtt először minden egyes oldalt megtervezünk.

## Adatbázis

Adatbázisnak „MySQL” adatbázist választottam, mivel ebben vagyok a legjáratosabb. Ez egy relációs adatbázis, amit könnyű használni, illetve futtatni. Az adatbázisom neve „ujmelo”, a dump-ját a backend mappában tárolom.

### „users” tábla

Ennek a táblának a bemutatásával kezdeném mivel ez az a tábla, ami a felhasználóval legelsőnek lép kapcsolatba. Öt mezőt tartalmaz, amik a következőek:

|  |  |
| --- | --- |
| **userID\*** | Ez a mező tartalmazza az összes felhasználónak az azonosítóját. Ez az érték alapján lesz a felhasználó beazonosítható. Ez az elsődleges kulcs. |
| **email** | Ez a mező tartalmazza a felhasználók email címeit, amire akkor van például szükség, amikor levelet küld a rendszer egy jelentkező beérkezésekor. |
| **password** | Ez a mező tartalmazza a hash-elt jelszavakat. |
| **firstName** | Ez a mező tartalmazza a keresztneveket. |
| **lastName** | Ez a mező tartalmazza a vezetékneveket. |

### „companies” tábla

Ez a tábla tartalmazz a céges információkat. A regisztrációnál a céges regisztráció csak egy opció így nem tartozik minden felhasználóhoz cég.

|  |  |
| --- | --- |
| **companiesID\*** | Ez a mező tartalmazza a cégeknek az azonosítójukat. Ez az elsődleges kulcs. |
| **name** | Ez a mező tartalmazza a cég neveket. |
| **logo** | Ez a mező tartalmazza a céglogóknak a fájl neveit. |
| **description** | Ez a mező tartalmazza a cég leírásokat. |
| **userID\*** | Ez a mező tartalmazza azokat a userID-kat, amikhez tartoznak az adott cégek. Ez egy idegenkulcs, ami a „users” tábla „userID” referenciája. |

### „category” tábla

Ez a tábla tartalmazza az összes kategóriát, amik elérhetők a weboldalon.

|  |  |
| --- | --- |
| **categoryID\*** | Ez a mező tartalmazza a kategóriáknak az azonosítójukat. Ez az elsődleges kulcs. |
| **category** | Ez a mező tartalmazza a kategóriákat. |

### „locations” tábla

Ez a tábla tartalmazza az összes magyarországi város neveit.

|  |  |
| --- | --- |
| **locationID\*** | Ez a mező tartalmazza a városoknak az azonosítójukat. Ez az elsődleges kulcs. |
| **location** | Ez a mező tartalmazza a városok neveit. |

### e) „applications” tábla

Ez a tábla tartalmazza az jelentkezést az állásokra.

|  |  |
| --- | --- |
| **userID\*** | Ez a mező tartalmazza a jelentkező azonosítóját. Ez egy idegenkulcs, ami a „users” tábla „userID” referenciája. |
| **advertisementID\*** | Ez a mező tartalmazza, hogy a felhasználó melyik állásra jelentkezett. Ez egy idegenkulcs, ami az „advertisements” tábla „advertisementID” referenciája. |
| **cv** | Ez a mező tartalmazza a jelentkezéskor feltöltött önéletrajz fájl nevét. |
| **ml** | Ez a mező tartalmazza a jelentkezéskor feltöltött motivációs levél fájl nevét. |

### f) „advertisement” tábla

Ez a tábla tartalmazza az álláshirdetések összes adatait. Ez a tábla rendelkezik a legtöbb mezővel.

|  |  |
| --- | --- |
| **advertisementID\*** | Ez a mező tartalmazza az álláshirdetések azonosítóját. Ez a mező az elsődleges kulcs. |
| **companiesID\*** | Ez a mező tartalmazza a céges azonosítóját, így lehet beazonosítani, hogy melyik céghez tartozik az adott hirdetés. Ez egy idegenkulcs, ami a „companies” tábla „comapniesID” referenciája. |
| **title** | Ez a mező tartalmazza az álláshirdetések címeit. |

|  |  |
| --- | --- |
| **introduction** | Ez a mező tartalmazza az álláshirdetések rövid bevezetőit. |
| **general** | Ez a mező tartalmazza az álláshirdetések általános leírásait. |
| **requirement** | Ez a mező tartalmazza az álláshirdetések elvárásait. |
| **benefit** | Ez a mező tartalmazza az álláshirdetések juttatásait. |
| **locationID\*** | Ez a mező tartalmazza az álláshirdetések településeinek azonosítóit. Ez egy idegen kulcs, ami a „location” tábla „locationID” referenciája. |
| **categoryID\*** | Ez a mező tartalmazza az álláshirdetések kategóriáinak azonosítóit. Ez egy idegenkulcs, ami a „category” tábla „categoryID” referenciája. |
| **wage** | Ez a mező tartalmazza az álláshirdetések órabéreit. |
| **createdAt** | Ez a mező tartalmazza azt, hogy mikor hozták létre az álláshirdetést. |

## Legfontosabb algoritmusok

### Regisztráció menete

A regisztráció az egyik legfontosabb algoritmus egy összetett webalkalmazásban, ezért gondoltam, hogy kiemelem ennek a menetét szakmai szemmel.

1. A register.html-en oldalon van az a form ami kezeli a regisztrációt. Itt van a submit button is ami triggereli a „register()” függvényt.
2. A függvény először elmenti az inputokat változókba majd POST metódussal elküldi ezeket a /auth/register végpontra.
3. A szerver oldalon először egy validációs függvénynek kell megfelelni. Minden mezőt validálok Express Validátorral. Pár példa erre:

* Email: isEmail() Meg kell felelnie az email követelményeinek tehát kell benne legyen @ karakter előtte utána betűk és a végén valamilyen domain.
* Policy: Be pipálva kell legyen az „Elfogadom az adatvédelmi nyilatkozatot.

1. Utána kerül sor a rendes regisztráció függvényre. Ha volt hiba a validáció során ezt visszaküldöm a kliens oldalra. Majd ezt a tömböt megjeleníti a kliens oldalon a mezők alatt.
2. Ha sikeres a validáció akkor, ha cégként regisztrált a felhasználó Multer segítségével feltölti a szervert a céglogóját.
3. Ezután pedig következik az adatbázisba való feltöltés.

### Bejelentkezés menete

Ha regisztráció van akkor egyértelműen kell lennie bejelentkezésnek is így most azt fogom bemutatni.

1. A login.html oldalon van az a form ami a bejelentkezést kezeli, a login.js-ben pedig a kliens oldali kódja. Submit gomb nyomására elmenti változókba a jelszót és az email címet majd ezt elküldi a szervernek.
2. A login szerveroldali kódja is először validálja a bemeneti mezőket, ebben az esetben, hogy ne legyenek üresek.
3. Két segédfüggvény a bejelentkezéshez:

A getUserByEmail függvény ellenőrzi, hogy melyik felhasználó tartozik egy adott email címhez. A comparePassword pedig azt ellenőrzi, hogy a beírt jelszó, illetve a bycrypt-el titkosított jelszó megegyezik-e.

1. Utána a login függvény, ha hiba van a validációval visszaküldi kliens oldalra a hibákat.
2. Ezután ellenőrzi a login függvény, hogy létezik-e ezzel az email címmel felhasználó, ha nem hibát visszaküldi, majd ugyanez a jelszóval.
3. Ha létezik ez a felhasználó akkor elkészíti a tokent a jwt.js és cookie-ba elmenti „access-token” néven.

### Álláshirdetés létrehozásának menete

Az álláshirdetésekkel foglalkozó weboldalak leglényegesebb funkciója értelemszerűen az álláshirdetések létrehozása.

1. Az vezérlőpanelen belül az advertisements.html azaz az „Álláshirdetések” menüponton található oldalon van egy „Új álláshirdetés” gomb.
2. Ezt megnyomja lefut az a kliens oldali kód, amely fetch-el a /advertisement/create-new-ra ahol létrejön egy új rekord az advertisements táblában.
3. Majd, ha sikeres a létrehozás akkor elirányít a kliens oldal ennek az álláshirdetésnek a szerkesztésére.
4. Az advertisment/create-new végponton először is azzal kell foglalkozni, hogy melyik céghez tartozik az adott hirdetés, ez a tokenből kinyerenő userID-ből állapítható meg.
5. Ezután már csak egy adatbázis feltöltés történik.

### Dinamikus órabér képzés

A kezdőlapon az órabér szűrő dinamikusan töltődik fel. Ez egy érdekes funkció ezért szeretném bemutatni, hogy hogyan is működik ezeknek a tartományoknak a képzése.

Ez a függvény lényege, hogy a hirdetések órabéreinek kinyerése után generál 3 órabér tartományt a következő módon:

X = (maximum – minimum) / 3

Ezután a tartományok a következőek:

1. (Minimum) – (Minimum + x)
2. (Minimum + x) – (Minimum + 2 \* x)
3. (Minimum + 2 \* x) – (Maximum)

## Tesztelési dokumentáció

### Bejelentkezés

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset neve** | **Bevitt adat** | **Elvárt eredmény** | **Sikeresség** |
| Sikeres bejelentkezés | Megfelelő email és jelszó | Sikeres bejelentkezés, átirányítás a /home oldalra | Sikeres |
| Hibás adatokkal sikertelen bejelentkezés | Hibás email vagy jelszó | Hibaüzenet megjelenése: "Hibás email cím vagy jelszó." | Sikeres |
| Üres mezőkkel sikertelen bejelentkezés | Üres email és jelszó mezők | Hibaüzenet megjelenése: "Az email cím mező nem lehet üres.", "A jelszó mező nem lehet üres." | Sikeres |
| Megfelelő email és üres jelszóval sikertelen bejelentkezés | Megfelelő email, üres jelszó | Hibaüzenet megjelenése: "A jelszó mező nem lehet üres." | Sikeres |

### Regisztráció

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset neve** | **Bevitt adat** | **Elvárt eredmény** | **Sikeresség** |
| Sikeres regisztráció | Megfelelő adatok | Sikeres regisztráció, átirányítás a /login oldalra | Sikeres |
| Hibás email formátum | Hibás email formátum | Hibaüzenet megjelenése: "Érvénytelen email cím." | Sikeres |
| Két jelszó nem egyezik | Két jelszó különböző | Hibaüzenet megjelenése: "A két jelszó nem egyezik meg." | Sikeres |
| Üres mezők regisztrációkor | Hiányzó mezők | Hibaüzenet megjelenése az üres mezők miatt. | Sikeres |
| Adatvédelmi nyilatkozat nincs elfogadva | Hiányzó mezők | Hibaüzenet megjelenése az üres mezők miatt. | Sikeres |
| Hiányzó keresztnév | Hiányzó mezők | Hibaüzenet megjelenése az üres mezők miatt. | Sikeres |
| Hiányzó vezetéknév | Hiányzó mezők | Hibaüzenet megjelenése az üres mezők miatt. | Sikeres |
| Hiányzó jelszó | Hiányzó mezők | Hibaüzenet megjelenése az üres mezők miatt. | Sikeres |

### Új álláshirdetés létrehozása

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset neve** | **Bevitt adat** | **Elvárt eredmény** | **Sikeresség** |
| Üres inputok | - | Hibás mezők | Sikeres |
| Hiányos inputok | Csak a cím mező kitöltve | Minden mező hibás | Sikeres |
| Sikeres feltöltés | Minden mező helyesen kitöltve. | Átirányítás a /controlPanel/advertisements oldalra | Sikeres |

### Álláshirdetés törlése

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset neve** | **Bevitt adat** | **Elvárt eredmény** | **Sikeresség** |
| Álláshirdetés törlése | - | Az adott álláshirdetés törlése és átirányítás a /controlPanel/advertisements oldalra | Sikeres |

### Kezdőlap

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset neve** | **Bevitt adat** | **Elvárt eredmény** | **Sikeresség** |
| Álláshirdetések megjelennek | - | Az adott álláshirdetés törlése és átirányítás a /controlPanel/advertisements oldalra | Sikeres |
| Álláshirdetések szűrése település alapján | Település kiválasztás | Kizárólag az a település jelenik meg | Sikeres |
| Álláshirdetések szűrése kategória alapján | Kategória kiválasztás | Kizárólag az a kategória jelenik meg | Sikeres |
| Álláshirdetések szűrése órabér alapján | Órabér kiválasztás | Kizárólag az az órabér sáv jelenik meg | Sikeres |
| Álláshirdetések szűrése kulcsszó alapján | Kulcsszó kiválasztás | Kizárólag az a kulcsszó jelenik meg | Sikeres |
| Álláshirdetések rendezése legtöbbet fizető szerint | Legtöbbet fizető rendezés kiválasztás | Legtöbbet fizető az első | Sikeres |
| Álláshirdetések megtekintése | - | Átirányít a /advertisement/view/:id oldalra | Sikeres |
| Álláshirdetések szűrése és rendezése visszaállítása | - | Visszaállítja a szűrőket | Sikeres |

### Végpontok tesztelése

|  |  |  |  |
| --- | --- | --- | --- |
| **Teszteset neve** | **Bevitt adat** | **Elvárt eredmény** | **Sikeresség** |
| calculateWageRanges | - | Visszaad egy json listát | Sikeres |
| getAdvertisement | - | Visszaad egy json listát | Sikeres |
| getAllAdvertisements | - | Visszaad egy json listát | Sikeres |
| getAllCategories | - | Visszaad egy json listát | Sikeres |
| getAllLocations | - | Visszaad egy json listát | Sikeres |
| getApplicants | - | Visszaad egy json listát | Sikeres |
| getUsedCategories | - | Visszaad egy json listát | Sikeres |
| getUsedLocations | - | Visszaad egy json listát | Sikeres |

## Továbbfejlesztés

Egy ilyen típusú webalkalmazásban rengeteg funkció lehetne, ami mind kényelmesebbé tenné az állások hirdetését vagy keresését. Én ezekből a funkcióból a leglényegesebbeket készítettem el azonban ez idő alatt kialakult egy kép a fejemben, hogy talán mik a leghasznosabbak.

### Egyedi jelentkezés minden álláshirdetéshez

Talán a leghasznosabb tulajdonság egy ilyen webalkalmazásban az, hogy minden állásfeltöltő egyedileg be tudja állítani, hogy milyen mezőt kérnek az álláshirdetők. Például, ha lényeges valamilyen álláshoz a beszélt nyelvek akkor azt jelentkezéskor a felhasználó egyből be tudja állítani.

### Admin felhasználó

Jó fejlesztés lenne az is, ha lenne admin mód a webalkalmazásban, több funkciót is lehetne adni neki. Például, hogy az álláshirdetéseket megjelenés előtt az adminnak kellene jóváhagynia.

### Jelszó frissítése

Ha esetleg egy felhasználó elfelejti a jelszavát akkor lehetne egy „elfelejtett jelszó” funkció az oldalon.

# felhasználói dokumentáció

## A képen szöveg, képernyőkép, Webhely, Weblap látható Automatikusan generált leírásKezdőlap

## A képen szöveg, képernyőkép, szoftver, Weblap látható Automatikusan generált leírásÁlláshirdetés oldal

## A képen szöveg, képernyőkép, Betűtípus, diagram látható Automatikusan generált leírásRegisztráció

## A képen szöveg, képernyőkép, Betűtípus, Márka látható Automatikusan generált leírásBejelentkezés

## Vezérlő panel

### A képen szöveg, képernyőkép, diagram, Betűtípus látható Automatikusan generált leírásProfil

### A képen szöveg, képernyőkép, Weblap, szoftver látható Automatikusan generált leírásÁlláshirdetések

### A képen szöveg, képernyőkép, Betűtípus, szoftver látható Automatikusan generált leírásJelentkezők

### A képen szöveg, képernyőkép, Betűtípus, diagram látható Automatikusan generált leírásÁlláshirdetés szerkesztése