Tanuló(k) neve: Farkas Levente, Markó Bence Mendbajr, Parraghy Dániel József

Képzés: nappali munkarend

Szak: 5 0613 12 03 Szoftverfejlesztő és tesztelő technikus

**A záródolgozat címe:**

**Bazár Bunker**

Konzulens: Bólya Gábor

Beadási határidő: 2024.04.15

Győr, 2023. 04.12

**Módos Gábor**igazgató

**Konzultációs lap**

|  |  |  |  |
| --- | --- | --- | --- |
|  | A konzultáció | | Konzulens aláírása |
| ideje | témája |
| 1. | 2024.02.15. | Témaválasztás és specifikáció |  |
| 2. | 2024.03.14. | Záródolgozat készültségi fokának értékelése |  |
| 3. | 2024.04.12. | Dokumentáció véglegesítése |  |

**Tulajdonosi nyilatkozat**

Ez a dolgozat a saját munkánk eredménye. Dolgozatunk azon részeit, melyeket más szerzők munkájából vettünk át, egyértelműen megjelöltük.

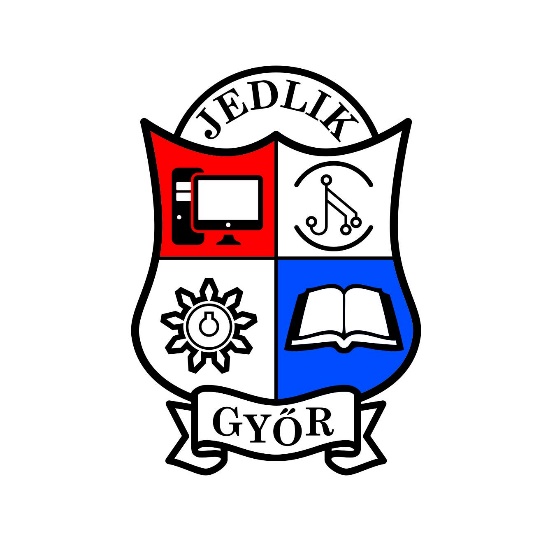
Ha kiderülne, hogy ez a nyilatkozat valótlan, tudomásul vesszük, hogy a szakmai vizsgabizottság a szakmai vizsgáról kizár minket és szakmai vizsgát csak új záródolgozat készítése után tehetünk.

Győr, 2024. április 15.

tanuló aláírása

tanuló aláírása

tanuló aláírása

****

Győri Szakképzési Centrum Jedlik Ányos Gépipari és Informatikai Technikum és Kollégium



Farkas Levente – Parraghy Dániel – Markó Bence Mendbajr

Győr 2024

Tartalom

1. Bevezetés 6

1.1 A probléma és megoldása 6

1.2 A záródolgozat publikus elérhetősége 6

1.3 Teszt felhasználói belépési adatok 6

2. A záródolgozat készítésének folyamata 6

2.1 Projektszervezés, tervezés 6

2.2 Csapatmunka, feladatelosztás 7

2.2.1 Kommunikációs csatornák 7

2.3 A projekt jövője 8

3. Adatbázis 8

3.1 Az adatbázis logikai struktúrája 8

3.2 Az adatbázis tábláinak felépítése 8

4. Backend 10

4.1 Fejlesztői környezet / nyelv 10

4.2 API dokumentáció 10

4.2.1 AdController 10

4.2.3 EmailController 13

4.2.4 UploadController 14

4.2.5 UserController 14

4.2.6 SupportController 18

4.3 Backend tesztek: 18

5. Frontend 19

5.1 Nyelvválasztás: React.js 19

5.2 Design nyelv: CSS 20

5.3 JavaScript 21

5.4 WebDesign: 21

5.5 Oldalak: 22

5.5.1 WelcomePage 22

5.5.2 Hirdetések 22

5.5.3 Hirdetés Részletek 23

5.5.4 Bejelentkezés 24

5.5.5 Regisztráció 25

5.5.6-7-8 Elfelejtett jelszó, Elfelejtett jelszó bejelentkezés, Új jelszó megadása 25

5.5.9 Új hirdetés 27

5.5.10 Profil oldal 27

5.5.11 Profilképek 29

5.5.12 Saját hirdetések 29

5.5.13 Support 30

5.5.14 AdminPage 30

6. Natív mobilapplikáció felhasználásának bemutatása 32

6.1 Fejlesztési környezet 32

6.2 Az első indítás 32

6.3 Támogatás 33

6.3.1 Bejelentkezés nélkül 33

6.3.2 Bejelentkezést követően 33

6.4 Felhasználói profil kezelése és a hozzátartozó beállítások 34

6.4.1 Bejelentkezés 34

6.4.2 Új felhasználó regisztrálása 35

6.4.3 Elfelejtett jelszó visszaállítása 35

6.4.4 Felhasználói profil módosítása 36

6.5 Keresési lehetőségek a hirdetések között 38

6.5.1 Kategóriák szerinti szűrés a főoldalról 38

6.5.2 Paraméteres keresés 38

6.6 Hirdetések böngészése, csoportosítása 39

6.6.1 Lista nézet 39

6.6.2 Részletes nézet 39

6.6.3 Kapcsolatfelvétel az eladóval 40

6.6.4 Hirdetés hozzáadása a „Kedvencekhez” 40

6.6.5 Kedvencek elérése 41

6.7 Saját hirdetés feltöltése, szerkesztése 41

6.7.1 Feltöltés menete 42

6.7.2 Fényképek feltöltése a hirdetéshez 42

6.7.3 Feltöltött hirdetés szerkesztése 43

6.7.4 Hirdetés törlése 43

# 1. Bevezetés

## 1.1 A probléma és megoldása

Szinte minden tanuló tanulmányi karrierje során legalább egyszer előfordul, hogy egy adott kötelező olvasmányt, vagy szükséges felszerelést nem tud fellelni még az iskolai vagy egyetemi könyvtárban vagy régiségboltokban sem. Az ilyen helyzetekre kíván megoldást nyújtani záródolgozatunk, a „Bazár Bunker”, mely egy specifikusan a tanulóknak fejlesztett adás-vétel oldal. Így a nebulóknak soha többé nem kell hosszú órákig kutatni a várost egy-egy kötelező olvasmányért, mert projekt munkánkkal percek alatt fellelheti az áhított tárgyat.

## 1.2 A záródolgozat publikus elérhetősége

Záródolgozatunkat az „Open-Source” szellemiségében készítettük, ebből adódóan programunk minden sora elérhető csapatunk GitHub oldalán, az alábbi linken.A linken megtalálható projektünk minden összetevője.

Github:  
<https://github.com/farkaslevente/14AA-D-Bazarbunker>

## 1.3 Teszt felhasználói belépési adatok

# 2. A záródolgozat készítésének folyamata

## 2.1 Projektszervezés, tervezés

Záródolgozatunk témáját késéssel választottuk meg, mivel az eredeti ötletünkről egy kis idő után kiderül, hogy túlmutatott az akkori tudásunkon. Ennek ellenére a folyamat kedvező ütemben haladt. Elsőként természetesen a záródolgozat témája került megalkotásra, majd nem sokkal később ezt követte adatbázisunk elemeinek felépítése. Miután ezek az alapvető feladatok befejeződtek a csapat elkezdett a megbeszélt leosztásnak megfelelően dolgozni, mindenki a maga-maga ágazatával.

## 2.2 Csapatmunka, feladatelosztás

Csapatmunkánk során úgy gondoltuk, hogy minden tag (jelen esetben 3) záródolgozatunk egy-egy ágazatát fejleszti, így született meg az alábbi feladatleosztás:

**Farkas Levente**

* Natív mobilapplikáció fejlesztése és tesztelés
* grafikai feladatok (lógók készítése, mobil applikáció design alkotása stb.)
* dokumentáció mikromenedzselés

**Markó Bence Mendbajr**

* Backend oldali fejlesztés és tesztelés
* Szerver menedzselés
* Adatbázis menedzselés

**Parraghy Dániel József**

* Frontend oldali fejlesztés és tesztelés
* weboldal design megalkotás
* Közös munka menedzselés

### 2.2.1 Kommunikációs csatornák

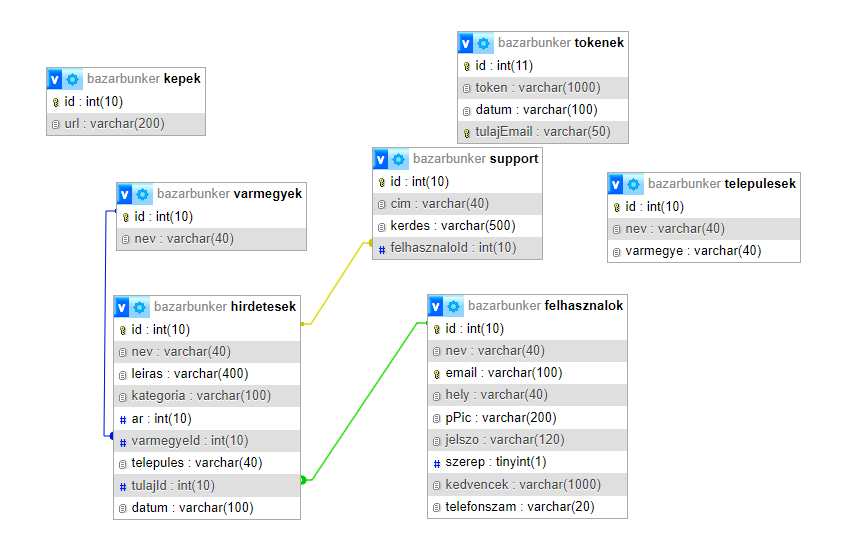
Munkánk során több fajta kommunikációt is alkalmaztunk, valamint minden újdonságról értesítettük egymást. Ezt nagyban megkönnyítette, hogy fizikailag egy helyen tartózkodtunk, valamint egy csoportban voltunk az iskolában töltött idő alatt. Amennyiben már nem az iskolában, hanem otthon „home office-ban” folytattuk a munkát hétvégente, vagy délutánonként a kommunikáció a Facebook Messenger szolgáltatáson, egy közös csoportban zajlott. Itt kerültek megosztásra a fontosabb információk, a dokumentáció egyes darabja, egymás ellenőrzése és még sok más tevékenység, mely elkerülhetetlen egy csoportos feladat során.

## 2.3 A projekt jövője

A záródolgozat jövője egyenlőre még zavaros, folyamatos mozgásban van, azonban az esetleges további fejlesztés esetén mindenképpen megvalósítanánk az eredeti ötlet részeként is megálmodott lehetőséget, hogy az eladó és a potenciális vevő a különböző külső lehetőségeken túl rendszerünkön belül is válthasson privát üzeneteket, így egyszerűsítve az adás vétel menetét.

# 3. Adatbázis

## 3.1 Az adatbázis logikai struktúrája



## 3.2 Az adatbázis tábláinak felépítése

1. **Felhasználók**

* id: elsődleges azonosító
* nev: felhasználó neve, publikus megtekinthető
* email: felhasználó email-címe, mely a bejelentkezéshez elengedhetetlen
* hely: felhasználó tartózkodási helye, melyet a hirdetésnél jelenítünk meg
* pPic: felhasználó profilképe, mely publikusan elérhető
* szerep: felhasználó jogait definiáló változó, alapértelmezetten 0
* kedvencek: felhasználó által mentett hirdetések
* telefonszam: felhasználó telefonszáma, melyen keresztül kapcsolatba léphetnek a felhasználók

2. **Tokenek**

* id: elsődleges azonosító
* token: kiosztott token egésze
* datum: a token létrehozásának ideje
* tulajEmail: a tokenhez tartozó felhasználó email címe

3. **Hirdetések**

* id: elsődleges azonosító
* nev: a hirdetés címe, mely alapján keresni is lehet
* leiras: a hirdetés leírása, melyben megemlíthetők még fontos információk
* kategoria: a hirdetés kategóriája, ami alapján kereshetnek rá a felhasználók
* ar: a hirdetett termék ára, melyre szűrni is lehet
* varmegyeId: a településhez kapcsolódó vármegye kulcsa
* telepules: a termék tartózkodási helye, meghirdetés helyszíne
* tulajId: a hirdetés feladójának kulcsa
* datum: a hirdetés feladásának időpontja

4. **Vármegyék**

* id: elsődleges kulcs
* nev: a vármegye teljes neve

5. **Települések**

* id: elsődleges kulcs
* nev: település teljes neve
* varmegye: a település mely vármegyéhez tartozik

6. **Képek**

* id: elsődleges kulcs,
* url: a kép forrása

7. **Support**

* id: elsődleges kulcs
* cim: a feltett kérésnek vagy kérdésnek címe
* kerdes: hosszasabb kifejtése a problémának
* felhasznaloId: a feltevő felhasználónak kulcsa

# 4. Backend

## 4.1 Fejlesztői környezet / nyelv

A Node.js az egyik hanem legnépszerűbb JavaScript környezet egyszerűsége és hatékonysága végett. Lehetővé teszi Javascript kódok futtatását szerver oldalon, így a fejlesztők könnyen és gyorsan tudnak olyan webes alkalmazásokat alkotni, melyek kódja a böngészőn kívül fut. Mindezt teszi egy egységes nyelven, hogy a lehetséges ütközések száma a minimális legyen.

## 4.2 API dokumentáció

### 4.2.1 AdController

Elérési útvonal: /ads/

**getAds:**

• Kérés típusa: HttpGet

• Paraméter: nincs

• Jogosultság: Admin, User

• Feladat: kilistázza az összes hirdetést

• Válasz: JSON

• Hiba: üres tömbD

**postAds:**

• Kérés típusa: HttpPost

• Paraméter: Ad objektum

• Jogosultság: Admin, User

• Feladat: Új hirdetés hozzáadása

• Válasz: 200 OK

• Hiba: 500 Internal server error

**editAds:**

• Kérés típusa: HttpPut

• Paraméter: Ad objektum

• Jogosultság: Admin, User

• Feladat: Saját meglévő hirdetés módosítása

• Válasz: 200 OK

• Hiba: 500 Internal server error

**deleteAds:**

• Kérés típusa: HttpDelete

• Paraméter: Id

• Jogosultság: Admin, User

• Feladat: Meglévő hirdetés törlése

• Válasz: 200 OK

• Hiba: 500 Internal server error

**modifyAds:**

• Kérés típusa: HttpPut

• Paraméter: Ad objektum

• Jogosultság: Admin

• Feladat: Meglévő hirdetés módosítása

• Válasz: 200 OK

• Hiba: 500 Internal server error

**removeAds:**

• Kérés típusa: HttpPut

• Paraméter: Ad objektum

• Jogosultság: Admin

• Feladat: Meglévő hirdetés módosítása

• Válasz: 200 OK

• Hiba: 500 Internal server error

***4.2.2 AuthController***

Elérési útvonal: /

**register:**

• Kérés típusa: HttpPost

• Paraméter: User objektum

• Jogosultság: Admin, User

• Feladat: Új fiók létrehozása

• Válasz: 200 OK

• Hiba: 500 Internal server error

400 Bad Request

403 Forbidden

**login:**

• Kérés típusa: HttpPost

• Paraméter: User objektum

• Jogosultság: Admin, User

• Feladat: Bejelentkezés meglévő fiókba

• Válasz: JSON

• Hiba: 500 Internal server error

400 Bad Request

401 Unathorized

**resetPassword:**

• Kérés típusa: HttpPost

• Paraméter: Email cím

• Jogosultság: Admin, User

• Feladat: Jelszó visszaállítása

• Válasz: 200 OK

• Hiba: 500 Internal server error

404 Not found

**authorizeReset:**

• Kérés típusa: HttpPost

• Paraméter: Visszaállító kód, Email cím

• Jogosultság: Admin, User

• Feladat: Fiókhitelesítés

• Válasz: 200 OK

• Hiba: 500 Internal server error

401 Unathorized

### 4.2.3 EmailController

Elérési útvonal: /email

**subscribe:**

• Kérés típusa: HttpGet

• Paraméter: Üres tömb

• Jogosultság: Admin, User

• Feladat: Feliratkozás hírlevélre

• Válasz: 200 OK

• Hiba: 500 Internal server error

### 4.2.4 UploadController

elérési útvonal: /pictures/

**upload:**

• Kérés típusa: HttpPost

• Paraméter: Kép

• Jogosultság: Admin, User

• Feladat: Kép feltöltése

• Válasz: 200 OK

• Hiba: 500 Internal server error

**uploadedPictures:**

• Kérés típusa: HttpGet

• Paraméter: Üres tömb

• Jogosultság: Admin, User

• Feladat: Feltöltött képek lekérése

• Válasz: 200 OK

• Hiba: 500 Internal server error

### 4.2.5 UserController

Elérési útvonal: /users/

**getUsers:**

• Kérés típusa: HttpGet

• Paraméter: Üres tömb

• Jogosultság: Admin

• Feladat: Összes felhasználó lekérése

• Válasz: JSON

• Hiba: 500 Internal server error

**getUserById:**

• Kérés típusa: HttpGet

• Paraméter: Id

• Jogosultság: Admin, User

• Feladat: Egy felhasználó lekérdezése ID alapján

• Válasz: JSON

• Hiba: 500 Internal server error

**putUsers:**

• Kérés típusa: HttpPut

• Paraméter: User objektum

• Jogosultság: Admin, User

• Feladat: Meglévő fiók adatainak változtatása

• Válasz: 200 OK

• Hiba: 500 Internal server error

**deleteUsers:**

• Kérés típusa: HttpDelete

• Paraméter: Id

• Jogosultság: Admin, User

• Feladat: Fiók törlése

• Válasz: 200 OK

• Hiba: 500 Internal server error

**editUsers:**

• Kérés típusa: HttpPut

• Paraméter: User objektum

• Jogosultság: Admin

• Feladat: Felhasználó összes adatának változtatása

• Válasz: 200 OK

• Hiba: 500 Internal server error

**removeUsers:**

• Kérés típusa: HttpDelete

• Paraméter: userId

• Jogosultság: Admin

• Feladat: Felhasználó törlése

• Válasz: 200 OK

• Hiba: 500 Internal server error

**changePicture:**

• Kérés típusa: HttpPost

• Paraméter: Kép

• Jogosultság: Admin, User

• Feladat: Profilkép megváltoztatása

• Válasz: 200 OK

• Hiba: 500 Internal server error

**updateFavourites:**

• Kérés típusa: HttpPost

• Paraméter: Hirdetés ID

• Jogosultság: Admin, User

• Feladat: Hirdetés hozzáadása kedvencekhez

• Válasz: 200 OK

• Hiba: 500 Internal server error

**removeFavourites:**

• Kérés típusa: HttpPost

• Paraméter: Hirdetés ID

• Jogosultság: Admin, User

• Feladat: Hirdetés eltávolítása kedvencekből

• Válasz: 200 OK

• Hiba: 500 Internal server error

**newPassword:**

• Kérés típusa: HttpPost

• Paraméter: Jelszó

• Jogosultság: Admin, User

• Feladat: Jelszóváltoztatás

• Válasz: 200 OK

• Hiba: 500 Internal server error

**support:**

• Kérés típusa: HttpPost

• Paraméter: Support objektum

• Jogosultság: Admin, User

• Feladat: Kérdés feltétele fejlesztői csapat felé

• Válasz: 200 OK

• Hiba: 500 Internal server error

### 4.2.6 SupportController

elérési útvonal: /support

**getSupport:**

• Kérés típusa: HttpGet

• Paraméter: Üres tömb

• Jogosultság: Admin

• Feladat: Kérdések lekérése

• Válasz: JSON

• Hiba: 500 Internal server error

**deleteSupport:**

• Kérés típusa: HttpDelete

• Paraméter: ID

• Jogosultság: Admin

• Feladat: Kérdés törlése

• Válasz: 200 OK

• Hiba: 500 Internal server error

## 4.3 Backend tesztek:

A backend oldali teszteket unit tesztek végzik el, melyhez segítségül vettük a jest könyvtárat. Minden teszt véletlenszerű adatokkal történik így elkerülve az esetleges ütközéseket, valamint egy olyan adatbázisba kerülnek az adatok mely csak egy másolata az igazinak és a tesztek végeztével az törlődik.

# 5. Frontend

## 5.1 Nyelvválasztás: React.js

A React.Js egy ingyenes és nyílt forráskódú front-end JavaScript-könyvtár a felhasználói felületek, és UI-komponensek létrehozásához. A nyelv számára hatalmas pozitívum a Meta csoport támogatása, ezzel együtt az egyéni fejlesztők is felkarolták.

Elkészítéséhez nagy mennyiségű tutorial anyagot lehet találni a Youtube, GitHub és StackOverFlow felületein is, így bárki számára könnyen megtanulható. Emellett határtalan mennyiségű, nagy segítséget nyújtó VsCode extension áll a fejlesztők részére.

 A React nyelv továbbá rendelkezik sok előre beépített[[1]](#footnote-1), telepítendő komponenssel, melyek tovább emelik a felhasználói színvonalat, legyen szó fejlesztői vagy a kész weblapot használó felhasználó élményekről.

Továbbá fontos megjegyezni, hogy szolgáltatásunk megfelelő működéséhez eleng

edhetetlen a következő parancs kiadása, mely telepíti a szükséges elemeket.

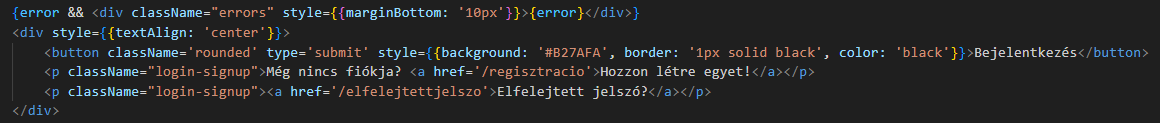
## 5.2 Design nyelv: CSS

Cascading Style Sheets - Lépcsőzetes stíluslapok

* A CSS leírja, hogyan kell a HTML-elemeket megjeleníteni a képernyőn, papíron vagy más adathordozón
* A CSS rengeteg munkát takarít meg. Egyszerre több weboldal elrendezését is vezérelheti
* A külső stíluslapok CSS-fájlokban tárolódnak

****A CSS-t a weboldalak stílusának meghatározására használják, beleértve a tervezést, az elrendezést és a megjelenítés variációit a különböző eszközökhöz és képernyőméretekhez.

Néha azonban egyszerűbb egyből a HTML elemnek adni egy design paramétert:



Ezzel a megoldással kisebb elemeket, melyeknek nem szeretnénk saját design osztályt létrehozni, egyből alakítani tudunk. Plusz pozitívum, hogy ezen paraméterek felülírják bármilyen más CSS osztály hatását, így biztosak lehetünk abban, hogy a megfelelő design kerül az elemre.

Ezen két megoldással, illetve a **Bootstrap** osztályainak helyenkénti alkalmazásával készült az oldal design-ja.

## 5.3 JavaScript

A **JavaScript**, amelyet gyakran JS- nek is szoktak rövidíteni, a web programozási nyelve és alaptechnológiája, a HTML és a CSS mellett. A webhelyek 99%-a JavaScriptet használ a kliens oldalon a weboldal viselkedéséhez. A JavaScript egy magas szintű, gyakran éppen időben lefordított nyelv, amely megfelel az ECMAScript szabványnak. Dinamikus gépeléssel, prototípus-alapú objektumorientációval és első osztályú funkciókkal rendelkezik. A Javascript a világ legnépszerűbb programozási nyelve.

A Javascript többek között lehetővé teszi, hogy lekorlátozzuk bizonyos „route”-okat, így például, a nem bejelentkezett felhasználó nem fér hozzá az „/ujhirdetes” oldalt.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

## 5.4 WebDesign:

Az oldal design-ja átesett egy teljes újragondoláson, mikor **Vue.js** -ről, **React.js** -re változtattuk meg a FrontEnd nyelvet. A design célja az volt, hogy egy könnyen navigálható, a számítógépektől távol álló felhasználók számára is könnyen használható legyen, barátságos környezetet biztosítson. A weboldalon animációk nem fellelhetőek a hardverkímélés miatt. A weboldal design-ja teljes mértékben reszponzív, így számítógépen és bármilyen kijelző nagyságú eszközön, mobiltelefonon is teljesen működőképes.

## 5.5 Oldalak:

### 5.5.1 WelcomePage

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírásEz az oldal jelenik meg, mikor betöltjük a weblapot. Illetve bejelentkezett felhasználók számára ez az oldal nyit lehetőség a hírlevelünkre történő feliratkozásra.

### 5.5.2 Hirdetések

A képen szöveg, képernyőkép, szoftver, Weblap látható

Automatikusan generált leírásAz oldal betöltéskor lekéri az adatbázisban lévő összes hirdetést, és megjeleníti ezeket egy „Card” komponens segítségével. Ha egy felhasználó bejelentkezik, akkor saját hirdetéseit nem látja ezen az oldalon, a félreértés elkerülése, és a felhasználói élmény javítása érdelében. Az oldal továbbá biztosít való idejű szűrést a betöltött hirdetéseken az alábbi paraméterek alapján:

* Megnevezés (hirdetés megnevezése)
* Minimum és Maximum ár
* Vármegye
* Kategória

Továbbá ezeket a szűrőket egy gombnyomással alaphelyzetbe állíthatjuk.

A hirdetéseken elhelyezett „Részletek” gombra kattintva megtekinthetjük a hirdetés részleteit.

### 5.5.3 Hirdetés Részletek

A képen szöveg, képernyőkép, irodaszerek látható

Automatikusan generált leírásAz oldal megjeleníti a hirdetés részleteit (megnevezés, ár, település, hirdető neve, telefonszáma, leírás). Továbbá a hirdetéshez feltöltött képeket egy „AdDetailsCarousel” komponens segítségével jelenítjük meg. Mindemellett a bejelentkezett felhasználó itt tudja felvenni a „Kedvencek” közé a hirdetést.

### 5.5.4 Bejelentkezés

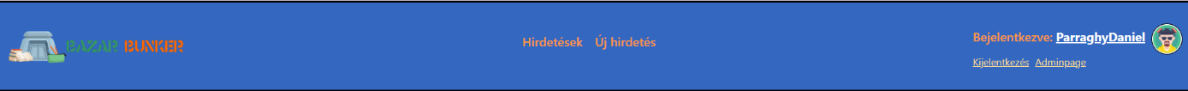
Az oldal egy form-ot tölt be, melynek helyes kitöltése után a Welcome oldalra navigál minket a szoftver, hiba esetén pedig visszajelzést kap a felhasználó.

A képen szöveg, képernyőkép, Operációs rendszer, Betűtípus látható

Automatikusan generált leírásA képen szöveg, képernyőkép látható

Automatikusan generált leírásErről az oldalról tudunk tovább haladni a „regisztráció” és „elfelejtett jelszó” oldalakra.

Bejelentkezés után a felhasználó számára elérhetővé válnak az eddig zárolt oldalak, és lezárttá válik többek között a „Bejelentkezés” oldal is.

Továbbá a „NavBar” komponens is megváltozik, hogy vizuális megerősítést nyújtson a felhasználó számára a bejelentkezés sikerességéről[[2]](#footnote-2).

### 5.5.5 Regisztráció

A képen szöveg, képernyőkép, szám, Betűtípus látható

Automatikusan generált leírásA regisztáláskor a felhasználónak kötelező megadnia ezen adatokat. Hibaelhárítást végeznek a HTML elemek paraméterei. Tovább egy egyszerű függvény a „Település” mező kezdőbetűjét nagyra állítja.

### 5.5.6-7-8 Elfelejtett jelszó, Elfelejtett jelszó bejelentkezés, Új jelszó megadása

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásEzen az oldalon a felhasználó megadhatja, hogy milyen email címre küldje az oldal a 8 karakteres kódot.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA „Tovább” gombra kattintva ez várja a felhasználót. Itt alap értékként megkapja az előző oldalon megadott email címet:

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásItt a felhasználó gyakorlatilag újra bejelentkezik, majd egyből az „Új jelszó megadás” oldalra küldjük:

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásItt a felhasználó megad egy új jelszót, és saját profiljára kerül át:

### A képen szöveg, képernyőkép, Betűtípus, szám látható Automatikusan generált leírás5.5.9 Új hirdetés

Bejelentkezés után megjelenik a felhasználó számára a lehetőség, hogy új hirdetést hozzon létre. Betöltéskor az oldal lekéri az adatbázisban szereplő vármegyék és települések adatait. Kategóriát előre beállított lehetőségekből választhatunk.

Az összes mező kitöltése kötelező, hogy a szűrések megfelelő módon tudjanak működni.

Kép(ek) feltöltése szintén kötelező, annak érdekében, hogy részletesebbé tegyük a hirdetéseket.

### 5.5.10 Profil oldal

A képen szöveg, képernyőkép, Emberi arc, szoftver látható

Automatikusan generált leírásBetöltéskor lekéri az adatbázisból a felhasználó adatait. Továbbá az oldal sok további lehetőséget biztosít a felhasználó számára. A felhasználó tudja saját adatait szerkeszteni, kedvenc hirdetéseit is meg tudja tekinteni, és ha úgy döntene a felhasználó, hogy szolgáltatásainkat nem szeretné továbbra is igénybe venni, akkor törölheti fiókját is.

A képen szöveg, Emberi arc, képernyőkép, rajzfilm látható

Automatikusan generált leírásA képen szöveg, képernyőkép, tervezés látható

Automatikusan generált leírásHelyben történő, egyszerű adatváltoztatást biztosítunk, ezzel is a felhasználó szabadságát helyezzük előtérbe. Illetve a már korábban említett „Új jelszó” oldalra innen is el tud látogatni a felhasználó.

A kedvenc hirdetések, illetve a felhasználó törlésének megerősítését mind a **react-modal** segítségével tesszük.

A képen szöveg, képernyőkép, Betűtípus, Weblap látható

Automatikusan generált leírásA kedvenc hirdetések modal linkeket jelenít meg a kedvencek közé helyezett hirdetésekhez, amiket a felhasználó adataiból nyer ki.

### 5.5.11 Profilképek

A képen képernyőkép, Emberi arc, rajzfilm látható

Automatikusan generált leírásEzen oldal az adatbázisban fellelhető, előre beállított „avatar” képek közül tud választani a felhasználó. Különösen jó ez a megoldás, hiszen könnyedén tudunk az adatbázisban megadni új képeket, ezzel tovább bővítve a választékot.

### 5.5.12 Saját hirdetések

A képen szöveg, képernyőkép, Betűtípus, diagram látható

Automatikusan generált leírásA felhasználó számára megjeleníti az általa feltöltött hirdetések minimális információit. Ha a felhasználó így sem biztos, hogy melyik hirdetés, melyik, a „Megnevezés” oszlopba kattintva megtekintheti részletesen a hirdetést.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA szerkesztő modal, lehetővé teszi, hogy a felhasználó a hirdetés bármely elemének változtatását[[3]](#footnote-3). Továbbá lehetőség nyílik a hirdetések törlésére is.

### A képen szöveg, képernyőkép, Betűtípus, szám látható Automatikusan generált leírás5.5.13 Support

Lehetővé teszi a felhasználó számára, hogy közvetlenül az adminokkal tudjon beszélgetést kezdeményezni. A beérkezett kérdések az adatbázisban láthatóak az adminok számára, így azokra hamar válasz fog érkezni.

### 5.5.14 AdminPage

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA képen szöveg, szám, sor, Betűtípus látható

Automatikusan generált leírásA képen szöveg, Betűtípus, szám, sor látható

Automatikusan generált leírásA képen szöveg, Betűtípus, sor, szám látható

Automatikusan generált leírásAz egyetlen oldal, amely nem 100 százalékban reszponzív, mivel csak admin felhasználók láthatják. Lehetővé teszi ezen adminok számára az **összes** felhasználó és hirdetés szerkesztését. Megjeleníti továbbá a beérkezett support kérdéseket, illetve a felhasználó kezeléshez használt **jwt** tokeneket. Természetesen, az összes adat törlésére is lehetőséget biztosít az oldal.

Mindezen feladatok elvégzéséhez, segítséget nyújtanak modal komponensek.

A képen szöveg, képernyőkép, Betűtípus, szám látható

Automatikusan generált leírásA képen szöveg, képernyőkép, képernyő, Betűtípus látható

Automatikusan generált leírásÚgy gondoljuk, hogy bizonyos FrontEnd tesztelések futtatása nélkülözhetetlen az oldal megfelelő működésének megbízhatóságáról. Erre a feladatre a **SeleniumIDE** Google Chrome bővítményt használtuk, mellyel a legérzékenyebb folyamtokat teszteltük:

* Elfelejtett jelszó
* Elfelejtett jelszó bejelentkezés
* A képen szöveg, képernyőkép, Betűtípus látható

  Automatikusan generált leírásBejelentkezés
* Új jelszó
* Bejelentkezés

# 6. Natív mobilapplikáció felhasználásának bemutatása

## 6.1 Fejlesztési környezet

Applikációnkat az iskolában tanított és használt, Visual Studio 2022 fejlesztői környezetben alkottuk, a Microsoft által fejlesztett .NET MAUI alapokra építve. A MAUI alapvetően egy multiplatform C# alapú relatív új nyelv (hiszen mindössze 2022 májusában jelentették be a megjelenését), mely szinte minden, a piacon elérhető platformra fejlesztési lehetőséget nyújt. Megjelenítésre a platform azt XAML (Extensible Application Markup Language) nyelvet alkalmazza, melyet kiegészít a C#, mint a számításokhoz használt programozási nyelv. Alkalmazásunk alapvetően az MVVM (Model View ViewModel) szerkezetet használja, ezzel nagyban leegyszerűsítve a fájlszerkezet értelmezését, s a kód felhasználását.

## 6.2 Az első indítás

A betöltő képernyő után a felhasználót az alkalmazás kezdő oldala fogadja. Applikációnk alapvetően akár felhasználói fiókkal, akár anélkül is ugyan úgy használható. Ugyan természetesen vannak a bejelentkezésnek előnyei (mint például a kedvenc hirdetések elmentése, vagy hirdetések regisztrálása), azonban fiók nélkül az alkalmazás ugyan úgy felhasználható.

## 6.3 Támogatás

Amennyiben a felhasználónak segítségre van szüksége natív mobilalkalmazásunk felhasználásában nincs más dolga, mint megnyitni a „Támogatás” oldalt, amely két megjelenési formával rendelkezik, a felhasználó aktuális státusza alapján.

### 6.3.1 Bejelentkezés nélkül

Alkalmazásunk egyik célja, hogy felhasználóinak nyitva hagyja az ajtót a bejelentkezés mentes használat felé. Ennek egy következményeként született meg egy útmutató, mely ebből a dokumentációból készült el, amely elnavigálja a felhasználót az alkalmazás alapvető funkciói között, mint például a bejelentkezés, vagy a keresés menete. Mivel ez a felhasználási útmutató ezen dokumentációt használja fel, mint forrás minden alapvető funkció, amelyet egy felhasználó bejelentkezés nélkül használni tud megtalálható benne.

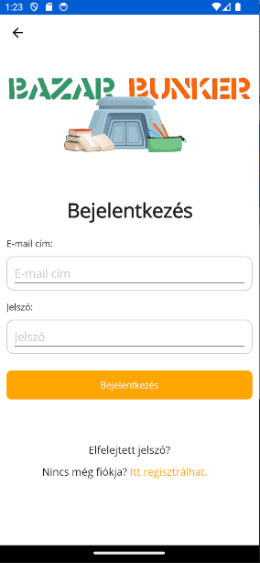
### 6.3.2 Bejelentkezést követően

Miután a felhasználó bejelentkezettegy megújult, megváltozott „Támogatás” oldal fogadja a felhasználót, ahol újabb opciók válnak elérhetővé. Bejelentkezést követően a felhasználó már feltehet konkrét kérdéseket, vagy akár fel is iratkozhat rendszerünk hírlevelére, így elkerülve azt, hogy bármilyen jövőbeli fejlesztésünkről lemaradjon.

## 6.4 Felhasználói profil kezelése és a hozzátartozó beállítások

### 6.4.1 Bejelentkezés

Mint az látható az applikáció megnyitásakor 3 fő menüpont jelenik meg a képernyő alján. Ezek az alábbiak (balról jobbra):

* Bejelentkezés
* Főoldal
* Támogatás

Bejelentkezésre a bal oldali ikon megnyomása után van lehetőség, amely után a felhasználó 3 irányba fordulhat:

* Bejelentkezés
* Regisztráció
* Amennyiben a felhasználó elfelejtette a fiók jelszavát, igényelhet egy újat

Bejelentkezéshez a felhasználó fiók email címét, valamint jelszavát kell megadni az oldalon. Helytelen bejelentkezés esetén a hibát az applikáció egy hibaüzenet formájában jelzi.

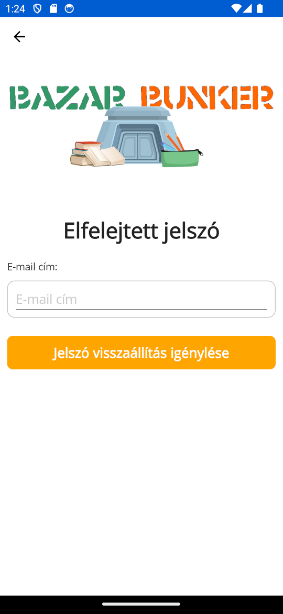
### 6.4.2 Új felhasználó regisztrálása

regisztráció esetén a következő adatokat kell megadni a leendő felhasználónak:

* Felhasználónév
* Felhasználói e-mail
* Felhasználói jelszó
* Felhasználó vármegyéje, majd települése
* Felhasználói telefonszám

Amennyiben a felhasználó nem ad meg minden adatot a rendszer egy hibaüzenettel jelzi a problémát, amíg a felhasználó nem ad meg minden szükséges adatot.

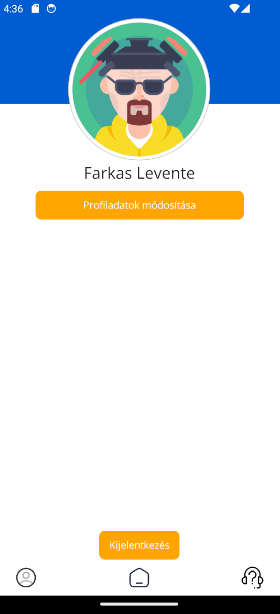
### 6.4.3 Elfelejtett jelszó visszaállítása

Mindenkivel előfordulhat, hogy olykor-olykor elfelejti jelszavait. Alkalmazásunk erre az eshetőségre is felkészült. A már említett bejelentkezés oldalon fellelhető egy szöveg („Elfelejtett jelszó?”), melyre koppintva az alkalmazás felnyitja egy újabb oldalát, amely lehetőséget nyújt a felhasználónak jelszava megváltoztatására. Ez a folyamat a mobil alkalmazásban 3 lépésből áll, azonban nem csak az alkalmazásban fog azonosítást nyerni a felhasználó személyazonossága. A folyamat első lépéseként a felhasználónak meg kell adnia a regisztráláshoz használt email címet. Ezt követően az applikáció egy újabb oldalon bekéri az időközben a megadott email címre elküldött 8 karakteres biztonsági kódot. Amennyiben ez a kód helyes, az alkalmazás egy újabb ablakot nyit, amelyben a felhasználó megadhatja az új jelszavát. Miután ezt megtette, a felhasználó bejelentkezhet az alkalmazásba az újonnan beállított jelszavával.

### 6.4.4 Felhasználói profil módosítása

Bejelentkezést követően a felhasználót már egy kissé megváltozott, kibővített Főoldal fogadja. Immár a következő lehetőségek nyíltak meg a felhasználó előtt:

* Profilbeállítások
* Saját Hirdetések
* Főoldal
* Támogatás
* Kedvenc hirdetések

Mivel bejelentkezett, lehetősége nyílt a profiladatok módosítására, melyet a profilt jelző ikon megnyomásával érhet el.

Alapvetően a következő felület fogadja a felhasználót. A felhasználónak itt lesz lehetősége kijelentkezni a fiókjából, valamint itt lehetséges a profiladatok esetleges módosítása, melyhez a „Profiladatok módosítása” gombra kell koppintani. Ezentúl sok oldalhoz hasonlóan erről a képernyőről is elérhető a főoldal, valamint a támogatás oldala, így szinte bárhonnan vissza lehet jutni a legfontosabb oldalakra.

gomb megnyomásakor megkezdődik a szerkesztés folyamata. A profil szinte összes adatát módosíthatja a felhasználó beleértve a profilképét is. A felhasználó a változtatásokat a „Profiladatok frissítése” gombbal véglegesítheti, azonban ez korlátok nélkül, bármikor újból módosítható. Ha a felhasználó mégsem kívánja módosítani adatait egy „swipe” mozdulattal a képernyő széléről visszavonhatja ezt a tevékenységet. Ha a felhasználó a profilképét szeretné kicserélni mindössze az ezt a funkciót jelző, a felhasználó név mellet lévő ikonra kell koppintania.

Ezen gomb lenyomásakor a felhasználó átjut egy új oldalra, ahol a csapatunk által előre meghatározott profilkép csomagból választhat a tetszésének megfelelő ikont. Az általunk szolgáltatott profilképek segítségével a felhasználó személyes ízléséhez igazíthatja virtuális profilját, melyet más felhasználók is láthatnak majd a hirdetések megtekintése során, ezzel kicsit személyesebbé téve a kapcsolatot a az alkalmazás és felhasználó között. Kiválasztást követően az applikáció visszanavigál a profil oldalra, azonban ebben az esetben a módosítások elvesznek. Azonban, ha a felhasználó meggondolja magát és mégsem módosítaná a profilképét, a már említett „swipe” mozdulattal visszajuthat a profiladatok módosításához, ahol a már megadott változtatások bekerülnek a megfelelő mezőkbe.

## 6.5 Keresési lehetőségek a hirdetések között

### 6.5.1 Kategóriák szerinti szűrés a főoldalról

Alkalmazásunkban hirdetések megtekintésére, keresésére több lehetőség is létezik, az, hogy melyiket használja a felhasználó csak rajta áll. Alapvetően a hirdetéseket 6 fő kategóriába soroltuk, melyeket a főoldalon 1-1 színes kép, egyes esetekben feliratok is jelölnek. Ezek a kategóriák a következőek:

* Általános iskolásoknak
* Középiskolásoknak
* Egyetemistáknak
* Kötelező olvasmány
* Írószerek
* Kiegészítők

Minden hirdetés regisztrálásakor kötelező megadni az adott hirdetéshez leginkább illő kategóriát, s ennek a kötelezettségnek köszönhetően minden hirdetés megjelenik az adott kategóriára való szűrés esetén.

### 6.5.2 Paraméteres keresés

Amennyiben a felhasználó nem csupán kategória alapján kíván keresni, az alkalmazás tartalmaz paraméteres keresést is. Ehhez a funkcióhoz a felhasználónak mindössze az applikáció főoldalon (a záródolgozat logója mellett) található kereső gombra kell koppintania. Ezt követően megjelenik a keresést segítő oldal, amelyen a felhasználó több különböző paramétert, kritériumot megadhat. A felhasználó a hirdetések között kutathat az áhított hirdetés címe, kategóriája, valamint a hirdetés helye (vármegye vagy akár település szintjén), illetve ára alapján. Az utóbbi esetében a felhasználó megadhatja a keresési ársáv minimum és maximum értékét.

## 6.6 Hirdetések böngészése, csoportosítása

### 6.6.1 Lista nézet

Miután a felhasználó rákeres a kritériumainak megfelelő hirdetésekre azok egy lista nézetben jelennek meg, melyben megjelennek blokkonként az egy-egy hirdetéshez tartozó feltöltött képek közül egy, valamint a hirdetés legfontosabb adatai, jellemzői. Hasonlóan jelennek meg az alkalmazásban a felhasználó kedvenc, valamint saját hirdetései. A hirdetés fejlécén a következő adatok láthatóak:

* A hirdetés címe
* A hirdetés ára
* A hirdetés Kategóriája
* A hirdetés helye (település szinten)
* A hirdetés feltöltésének dátuma

### 6.6.2 Részletes nézet

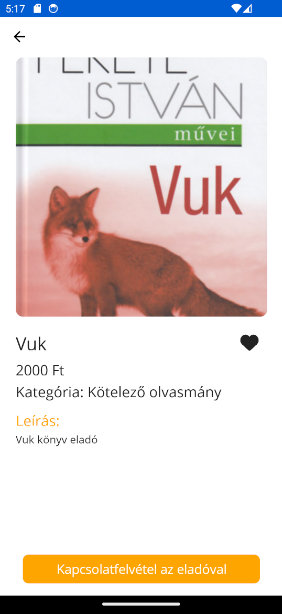
****Ezt a képernyőt akkor látja a felhasználó amikor kiválaszt egy számára szimpatikus hirdetést a lista nézetben. Itt megtekinthető az összes a hirdetéshez feltöltött fénykép, a hirdetés teljes leírása, valamint a kiválasztott hirdetés további adatai is. Ezen túl ezen a felületen láthatják egymás elérhetőségeit a felhasználók, amely külső lehetőséget biztosít a kapcsolatlétesítésre, email, telefon vagy akár sms formájában is. Továbbá ez az oldal az alkalmazás azon része, amelyen a felhasználó elmentheti a hirdetést a kedvencei közé, amely a főoldalról bármikor megtekinthető.

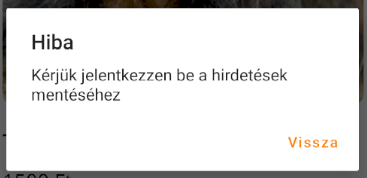
### 6.6.3 Kapcsolatfelvétel az eladóval

Ha egy hirdetés felkelti a felhasználó figyelmét lehetősége van az eladó adatainak megtekintésére. Ebben az esetben a felhasználónak nincs más dolga a cél eléréshez, mint rákoppintani a „Kapcsolatfelvétel az eladóval” gombra, ami után az eladó elérési adatai, valamint itt kap szerepet a profilképek egyénisége is. A következő adatok válnak láthatóvá a felugró ablakban:

* a hirdető neve
* a hirdetés lokációja
* a hirdető email címe/elektronikus elérhetősége
* a hirdető telefonos elérhetősége

### 6.6.4 Hirdetés hozzáadása a „Kedvencekhez”

Amennyiben az adott hirdetmény elnyerte a felhasználó érdeklődését, s úgy döntene, hogy később még fel kívánja keresni az adott hirdetést, applikációnk ehhez is kínál megoldást. A „Részletes nézet” segítségével a felhasználó mindössze egy koppintással elmentheti magának az adott tartalmat, illetve, ha meggondolná magát ugyan ilyen könnyedséggel vissza is vonhatja ezt a lépését. Abban az esetben, ha a hirdetést a felhasználó már hozzáadta kedvenceinek listájához a hirdetés képe/képei alatti szív ikon telítetté válik, ellenkező esetben az ikon külalakja látható csak.

Azonban ez a funkció már bejelentkezéshez kötött, így abban az esetben, ha a felhasználó nem rendelkezik fiókkal rendszerünkben az alkalmazás jelzi számára, hogy a funkció használatához legyen kedves bejelentkezni.

### 6.6.5 Kedvencek elérése

Miután az alkalmazás használója bejelentkezett a létrehozott fiókjába a főoldalról elérhetővé válik a kedvencek „mappája”, mint az a bejelentkezést követően a főoldal jobb alsó sarkában látható.

Az ikon koppintása után egy, a Lista nézethez hasonló lapon megjelennek a felhasználó által megjelölt „kedvenc” hirdetések.

## 6.7 Saját hirdetés feltöltése, szerkesztése

Természetesen adok-veszek tematikájú applikáció révén szó nem csak megtekinteni lehet a különböző hirdetéseket, hanem akár saját már megunt, vagy éppen nem használt tárgyait, tanulmányaihoz használt eszközeit, könyveit is képes meghirdeti a felhasználó. Ehhez magától érthetően ahhoz, hogy ezt a felhasználó megtehesse be kell jelentkeznie rendszerünkön keresztül egy fiókba.

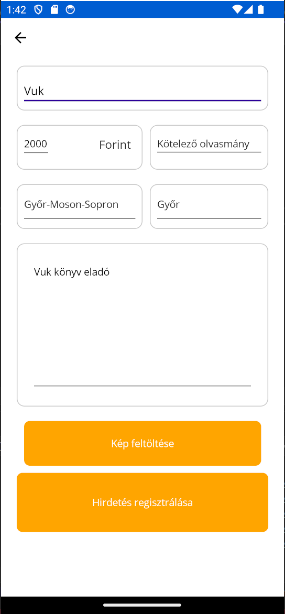
Amikor úgy dönt a felhasználó, hogy megosztja hirdetését az alkalmazás más használóival egy hirdetés feltöltése rendkívül hasonló megoldást alkalmaz a keresés menetével. A leendő hirdetőnek mindössze pár fontos adatot kell megadnia. Melyek a következőek:

* a hirdetés címe
* a hirdetés ára
* a hirdetés kategóriája
* a hirdetés vármegyéje
* a hirdetés települése
* a hirdetés leírása

Természetesen, ha a felhasználó úgy kívánja akár képekkel is kiegészítheti, színesítheti hirdetését, melynek nincs darabszámra vonatkozó limitációja.

### 6.7.1 Feltöltés menete

A feltöltés megkezdéséhez a felhasználónak nincs más dolga, mint a főoldalra navigálni, majd onnan a lenti ikonok közül a feltöltést szimbolizáló elemre koppintani.

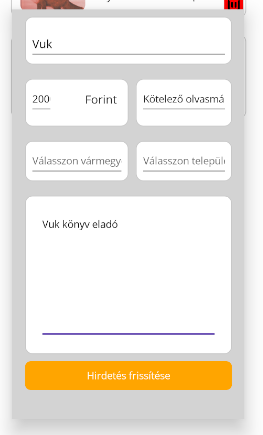
Ezt követően az alkalmazás automatikus átirányul a tematikusan a felhasználó hirdetéseinek kialakított oldalra, ahonnan megkezdődhet a feltöltés. A feltöltést az „Új hirdetés” gomb indítja meg, majd ezután az alkalmazás egy újabb oldalra irányul át. Ezen az oldalon nyílik lehetősége a felhasználónak megadni a hirdetése legfontosabb adatait, melyeket fentebb a „Saját hirdetés feltöltése, szerkesztése” pont alatt megemlített a dokumentáció, illetve ezen az oldalon nyílik lehetőség a hirdetéshez tartozó képek feltöltésére is. Ha a felhasználó megadott minden szükséges adatot a feltöltés véglegesítéséhez mindössze a „Hirdetés regisztrálása” gombra kell koppintania, s a hirdetése felkerül rendszerünk adatbázisába.

### 6.7.2 Fényképek feltöltése a hirdetéshez

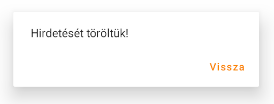
Amennyiben a felhasználó fényképek(et) kíván feltölteni hirdetéséhez, mindössze a „Kép feltöltése” gombra kell koppintania, s előjön a sokak által jól ismert galéria, amelyben egyesével kiválasztani a kívánt képeket, melyek ezután automatikusan felkerülnek rendszerünkbe, hozzárendelve az adott hirdetéshez, illetve az adott felhasználóhoz.

### 6.7.3 Feltöltött hirdetés szerkesztése

Magától érthetődő, hogy ha a felhasználónak van lehetősége feltölteni tartalmakat, akkor azokat szerkeszteni és törölni is képes kell, hogy legyen. Alkalmazásunk ezt a kérdést egy felhasználó barát, egyszerű módon teszi meg. Miután a felhasználó belép a hirdetéseit tartalmazó oldalra[[4]](#footnote-4) láthatja, hogy minden feltöltött hirdetés rendelkezik 2 gombbal, ami csak és kizárólag ezen hirdetések sajátja, melyek a szerkesztés és törlés feladatát látják el, mint az ikonjukból is látható.

Abban az esetben, ha a hirdetés tulajdonosa úgy gondolja, hogy változtatna hirdetése bármely[[5]](#footnote-5) tulajdonságán, mindössze a szerkesztés ikonjára kell koppintania, amely elővezet egy előugró „popup” ablakot, ami tartalmazza az adott hirdetés adatait, így egyszerűen és egyértelműen átírható bármilyen adat.

### 6.7.4 Hirdetés törlése

Egy hirdetés törlése még egyszerűbb, mint az említett hirdetés szerkesztése. Mindössze a felhasználónak annyi dolga van, hogy rákoppant a „kuka” ikonra, mely után a tulajdonost alkalmazásunk még egyszer megerősítésként megkérdezi, hogy biztosan törölni kívánja-e a hirdetést, s ha ezt a tulajdonos megerősítette, a rendszer a következő üzenettel jelzi a sikeres törlést.

1. A React modal egy olyan UI (User Interface) elem, melyet általában felugró ablakokhoz alkalmaznak [↑](#footnote-ref-1)
2. Fontos, hogy az „Adminpage” link csak az admin szerepű felhasználóknak jelenik meg. [↑](#footnote-ref-2)
3. A hirdetéshez feltöltött képeket nem lehet szerkeszteni, ezt a funkciót később szeretnénk hozzáadni az oldalhoz. [↑](#footnote-ref-3)
4. lásd feltöltés menete [↑](#footnote-ref-4)
5. Sajnálatos módon a hirdetések képeinek szerkesztését, törlését nem sikerült még elérni, esetleges jövőbeli fejlesztések részét képezni [↑](#footnote-ref-5)