



Debreceni SZC Mechwart András Gépipari és Informatikai Technikum  
14EKC

Szoftverfejlesztő és -tesztelő projektfeladat

# Kovács Mentor- Webalkalmazás Dokumentáció

Készítette:

Kacsora Adrienn

Dobrán Dániel

Bihari Csaba

2026

## Tartalomjegyzék

1. Bevezetés .....	4
1.1 A téma ismertetése: .....	4
1.2 A témaválasztás indoklása:.....	4
1.3 Szakmai célkitűzés: .....	4
1.4 Célközönség .....	5
2. Csapat és szerepek .....	6
2.1 Feladatmegosztás .....	6
2.2 Munkafolyamat és verziókezelés.....	6
3. Fejlesztői dokumentáció .....	6
3.1 A fejlesztéshez szükséges indoklás .....	6
3.2 Fejlesztőkörnyezet (hardver és szoftverek) ismertetése, és a választás indoklása .....	7
3.2.1 Hardver környezet.....	7
3.2.2 Szoftver környezet.....	7
3.2.3 Telepített csomagok (Requirements).....	8
3.3 A kialakított adatszerkezet, adatbázis bemutatása .....	9
3.3.1 Az adatbázis kapcsolatokat az alábbi ábra szemlélteti: .....	9
3.3.2 A fő modellek:.....	10
3.4 A program tipikus, egyedi, vagy érdekesebb algoritmusainak bemutatása.....	10
3.4.1 Egyedi Django Management Parancsok .....	11
3.4.2 Parancsok használata összefoglalva.....	13
3.5 Tesztelés leírása, tesztdokumentáció .....	14
3.5.1 Hardver/szoftver követelmények: .....	14
3.5.2 Teszt felhasználók .....	14
3.5.3 Tesztelési esetek .....	15
3.5.4 Tesztelési eredmények összegzése .....	17
3.6 Fejlesztési lehetőségek ismertetése .....	17
4. Felhasználói dokumentáció.....	18
4.1 A program céljának és lényegesebb funkcióinak összefoglalása.....	18
4.2 Szükséges hardvereszközök és szoftverek felsorolása.....	18
4.3 Telepítés és indítás lépéseinek ismertetése .....	18
4.4 Az alkalmazás részletes bemutatása .....	19
4.4.1 Kezdőoldal (Főoldal) .....	19

4.4.2 Regisztrációs oldal .....	22
4.4.3 Bejelentkezési oldal .....	24
4.4.4 Foglalási oldal (Fő funkció) .....	25
4.4.5 Sikeresnek ítélt foglalási oldal.....	27
4.4.6 Profil oldal.....	28
4.4.7 Admin oldal.....	28
4.4.8 Kilépés.....	29
4.5 Hibajelzések magyarázata .....	31
5. Összefoglalás .....	32
5.1 Irodalomjegyzék: .....	32

# 1. Bevezetés

## 1.1 A téma ismertetése:

A Kovász Mentor egy webalapú időpontfoglaló alkalmazás, amely lehetővé teszi a házi pékáruk iránt érdeklődőknek, hogy személyre szabott oktatási szolgáltatásokat foglaljanak le. Az alkalmazás a kovással való sütés művészetének oktatásához kapcsolódik, amely manapság egyre nagyobb népszerűségnek örvend. A rendszer összeköti az oktatókat és a tanulni vágyó felhasználókat egy online platformon keresztül, ahol könnyedén böngészhetik a kínált szolgáltatásokat, regisztrálhatnak, bejelentkezhetnek és szabad időpontokat foglalhatnak.

## 1.2 A témaválasztás indoklása:

A kovászos kenyérfőzés oktatására épülő weboldallal szeretnénk kiszolgálni a tudatos táplálkozás és a kézműves hagyományok iránti igényt. Egy ilyen platform nemcsak az egészségtudatos vásárlókat szólítja meg, hanem a stresszmentes, alkotó hobbira vágyókat is, miközben az online időpontfoglaló funkció professzionális és kényelmes ügyfélményt biztosít.

A szoftverfejlesztés és tesztelés tanulása keretében ez a téma gyakorlati alkalmazást nyújt a webfejlesztés alapjaira, beleértve a backend (Django), adatbázis-kezelés (SQLite), API-készítés (Django REST Framework) és frontend integrációt. A projekt vizsgamunka része, így alkalmas a tanult ismeretek demonstrálására.

## 1.3 Szakmai célkitűzés:

Egyrészt egy működőképes webalkalmazás létrehozása, amely támogatja a felhasználói regisztrációt, bejelentkezést, szolgáltatások böngészését és időpontfoglalást. Az alkalmazásnak biztonságosnak, felhasználóbarátnak és könnyen bővíthetőnek kell lennie, hogy a jövőben további funkciók (pl. fizetés, email értesítések) hozzáadhatóak legyenek. Másrészt a projekt célja a tanult ismeretek alkalmazása és elmélyítése, valamint a dokumentációs készségek fejlesztése.

## 1.4 Célközönség

Az alkalmazás két fő felhasználói csoportra lett tervezve:

- Felhasználók: akik szeretnék időpontot foglalni egy adott oktatásra.
- Adminisztrátorok: akik megtekintik, kezelik az időpontokat és az összes foglalást.

## 2. Csapat és szerepek

### 2.1 Feladatmegosztás

A projektet három fős csapat végezte. A feladatmegosztás a következőképpen alakult:

- Dobrán Dániel: Backend, adatmodell tervezés, API-k, tesztelés
- Kacsora Adrienn: Frontend sablonok, felhasználói űrlapok, CSS, tesztelés
- Bihari Csaba: Dokumentáció, tesztelés

Minden feladat elvégzése során a csapattagok átnézték egymás munkáját (code review jelleggel), és a végső integrációs lépéseket közösen hajtották végre.

### 2.2 Munkafolyamat és verziókezelés

A fejlesztés során a Github verziókezelő rendszerét használtuk a változások nyomon követésére. Hoztunk létre brancheket, melyeket pull requestet követően mergeltünk a fő ágba rövid commitok kíséretében.

## 3. Fejlesztői dokumentáció

### 3.1 A fejlesztéshez szükséges indoklás

Előzetes kutatás során számos hasonló megoldás létezik az interneten (pl. Calendly, Acuity Scheduling, Setmore), azonban ezek:

- Drága, fizetős szolgáltatások vagy az ingyenes változatok limitált funkciókkal rendelkeznek
- Nem testreszabhatóak az adott feladat specifikus igényei szerint
- Adatbiztonság és adatvédelmi kérdések merülhetnek fel
- Nem biztosítják a teljes körű kontrollálhatóságot a fejlesztő és az adminisztrátor számára

Az általunk fejlesztett megoldás ezekben az aspektusokban tér el:

- Nyílt forráskódú és testreszabható
- Teljes felügyelet az adatok és a biztonsági intézkedések felett
- Ingyenes megvalósítás
- Skálázható és bővíthető architektúra

## 3.2 Fejlesztőkörnyezet (hardver és szoftverek) ismertetése, és a választás indoklása

### 3.2.1 Hardver környezet

A projekt fejlesztése során háromfős csapatban dolgoztunk, és mindannyian hasonló számítógépeket használtunk. A fejlesztés otthoni környezetben történt, amely biztosította a szükséges erőforrásokat az alkalmazás hatékony futtatásához és teszteléséhez. Az ehhez megfelelő minimális hardver követelmények:

- Processzorok: Intel/AMD multi-core processzorok
- RAM: Minimum 8 GB
- Tárolóhely: 500 MB szabad terület
- Hálózat: Internetkapcsolat, szükséges a Python csomagok letöltéséhez

### 3.2.2 Szoftver környezet

Fejlesztéshez szükséges szoftverek listája, indoklással:

- Windows 10/11: Operációs rendszer
  - o Asztali gépen a legtöbb felhasználó ezt használja
  - o Kompatibilis a többi fejlesztői szoftverrel
- Python 3.11: A fő programozási nyelv
  - o Ingyenes, nyílt forráskódú
  - o Egyszerű, olvasható szintaxis
  - o Gyors fejlesztés lehetősége
  - o Számos előre megírt (ready-made) kódkönyvtára van, amely speciálisan webalkalmazások fejlesztésére készült
  - o Tanulási görbe nem meredek, így hatékonyabb a csapatmunka
- Django 5.1.7: Webkeretrendszer
  - o Kifejezetten webalkalmazások fejlesztésére tervezett Python framework
  - o Beépített autentikáció és engedélykezelés rendszer
  - o ORM (Object-Relational Mapping) a könnyű adatbázis kezeléshez
  - o Admin felület automatikus generálása
  - o Kiváló dokumentáció és nagy közösség
  - o Gyors prototípus-fejlesztés lehetősége

- Biztonság (beépített CSRF, SQL injection védelem)
- Django REST Framework 3.16.0: API fejlesztéshez
  - Könnyű API fejlesztés
  - Serialization automatika
  - Beépített JSON formátum támogatás
  - Verziókezelés támogatása
- SQLite (beépített Django-val): Adatbázis
  - Fájl-alapú
  - Nem igényel külön szerver telepítést
  - Adatok könnyedén hordozhatók
  - Ideális fejlesztési célokra
  - Fejlesztéshez tökéletes,
- Git (2.40): Verziókezelés
  - Változások követése
  - Visszaállítás lehetősége
  - Csapatmunka támogatása
- Visual Studio Code 1.85: Kódszerkesztő
  - Ingyenes
  - Bővíthető Python, Django és HTML/CSS támogatással
  - Beépített Git integrációval és debugger-rel

A választás indoklása összességében: a technológiák szabadon elérhetőek, jól dokumentáltak így alkalmasak a projekt megvalósítására.

### 3.2.3 Telepített csomagok (Requirements)

A projektünk több külső programkönyvtárra épül, ezeket a Python ökoszisztémájából töltöttük le:

- asgiref==3.8.1
- Django==5.1.7
- django-cors-headers==4.7.0
- djangorestframework==3.16.0

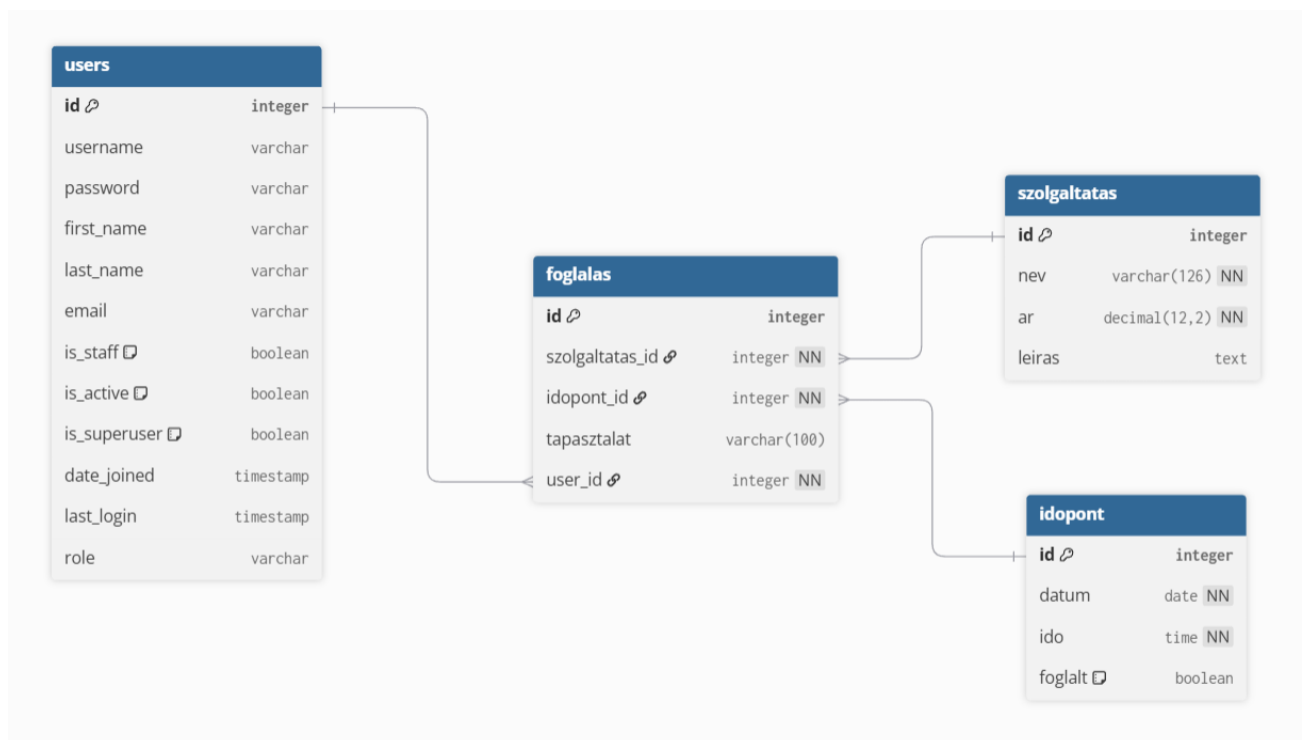


- et\_xmlfile==2.0.0
- numpy==2.2.6
- openpyxl==3.1.5
- pandas==2.3.1
- python-dateutil==2.9.0.post0
- pytz==2025.2
- six==1.17.0
- sqlparse==0.5.3
- typing\_extensions==4.13.0
- tzdata==2025.2

### 3.3 A kialakított adatszerkezet, adatbázis bemutatása

Az alkalmazás adatbázisát SQLite kezeli, amely egy könnyű, fájl-alapú relációs adatbázis. A db.sqlite3 fájl tartalmazza az összes adatot, és a Django ORM biztosítja a Python objektumok és adatbázis táblák közötti leképezést. Az adatbázis szerkezetét a Django migrations kezeli, amely verziókövetéssel biztosítja a változások követhetőségét.

#### 3.3.1 Az adatbázis kapcsolatokat az alábbi ábra szemlélteti:



Ábra 1: Adatbázis

Ez a szerkezet biztosítja az adatok integritását és könnyíti a lekérdezéseket.

### 3.3.2 A fő modellek:

- User (Felhasználó):

A Django által biztosított, beépített modell, amely a felhasználók adatait tárolja. Így tudjuk, ki jelentkezett be, és milyen jogosultságai vannak.

- Szolgáltatás:

Leírás a szolgáltatásról, ez tárolja a szolgáltatás árát, nevét, például "Kenyérfőzés" vagy "Készletkészítés".

- Időpont:

Az elérhető időpontok kezelése, ez a naptár, ez alkotja a rendszer gerincét. Itt látjuk, mely időpontok léteznek, és melyek vannak már lefoglalva, illetve biztosítja, hogy ugyanazon időpontban ne lehessen másik időpontot foglalni.

- Foglálás:

Ez a táblázat köti össze a felhasználókat, a szolgáltatásokat és az időpontokat: ki, mikor, milyen szolgáltatásra foglalt.

### 3.4 A program tipikus, egyedi, vagy érdekesebb algoritmusainak bemutatása

- Szabad időpontok szűrése: A foglalás view-ban az időpontokat szűrjük, hogy csak a jövőbeli szabad időpontok jelenjenek meg, legalább 2 órával előfoglalással. Algoritmus: Aktuális idő + 2 óra, majd szűrés dátum és idő alapján.
- Foglálás ellenőrzése: A foglal view-ban ellenőrizzük, hogy az időpont szabad-e, és ha van, akkor legalább 2 órával későbbi-e. Ha nem, hibaüzenet.
- API szabad időpontok: SzabadIdopontokApi view: Lekérdezi a foglalt=False időpontokat, serializálja és visszaadja JSON-ban.
- Regisztráció validáció: A regisztracio view-ban használjuk a Django password\_validation-t és egyedi ellenőrzéseket (pl. felhasználónév egyediség).

- Bejelentkezés és logout: Django beépített autentikáció használata, de kiegészítve üzenetekkel.

### 3.4.1 Egyedi Django Management Parancsok

Az alkalmazás három egyedi management parancsot tartalmaz az adminisztráció megkönnyítésére.

#### 1. `igenerate` – Időpontok automatikus generálása

Funkció: Automatikusan időpontokat generál a rendszerbe a következő napokra, melynek alapértelmezett beállítása van, de igény szerint manuálisan paraméterezhető.

Működés alapértelmezetten:

- Időintervallum: 10 napra előre (mai naptól számítva)
- Munkaórák: 10:00-20:00 között
- Gyakorisága: 3 óránként (10:00, 13:00, 16:00, 19:00)
- Hétvége kezelése: Szombatot és vasárnapot kihagyja (nem generál hétvégékre)
- Duplikáció védelem: elkerüli az ismétlődő időpontokat

Használata: `python manage.py igenerate`

Példa kimenet:

- 2026-01-06 (hétfő): 10:00, 13:00, 16:00, 19:00
- 2026-01-07 (kedd): 10:00, 13:00, 16:00, 19:00
- ... (10 munkanap)

Működés opcionális beállításokkal:

- `napok`: hány napra generáljon
- `lepeskoz`: hány óránként generáljon
- `kezdet`: kezdő dátum (YYYY-MM-DD formátumban)

Futtatási példa és kimenete:

```
python manage.py igenerate --napok 20 --lepeskoz 2 --kezdet 2026-03-02
```

Ezzel a paraméterekkel 20 napra, 2 óránként generálna időpontokat 2026.03.02-től.

## 2. idelete – Szabad időpontok törlése

Funkció: Kitörli a még le nem foglalt, jövőbeli időpontokat a mai naptól kezdve. Új időbeosztás készítésénél hasznos.

Működés:

- Lekérdezi az összes időpontot
- Ellenőrzi, hogy az időpont dátuma nagyobb vagy egyenlő a mai dátummal
- Csak a szabad időpontokat törli, foglalt időpontokat nem érinti

Használat: `python manage.py idelete`

## 3. mytask – Múltbéli foglalások archiválása

Funkció: Lementi a múltbéli foglalásokat Excel fájlba, majd törli őket az adatbázisból.

Működés:

- Lekérdezi az összes foglalást
- Kigyűjti azokat, amelyek dátuma korábbi a mai dátumnál
- Excel fájlba menti az alábbi adatokkal: email cím, dátum, időpont, szolgáltatás neve, ár
- Az Excel mentési helye: `backend/excelek/ÉÉÉÉ-HH-NN.xlsx` formátumban
- Törli a múltbéli időpontokat és foglalásokat az adatbázisból

Használat: `python manage.py mytask`

Automatizálás:

Ajánlott naponta automatikusan futtatni Windows Feladatütemezővel (Task Scheduler):

Időpont: Minden nap éjjel (például 01:00)

Batch fájl: `elmultidopontoktorlese.bat`

Előnyök:

- Adatok megőrzése archiválás céljából
- Adatbázis méretének karbantartása

- Tárhely optimalizálás
- Adatelemzés lehetősége Excel-ben

Excel fájl példák:

- backend/excelek/2026-01-05.xlsx
- backend/excelek/2026-01-06.xlsx

#### 4. stat – Grafikon készítés

Grafikont készít az excelek mappában levő excel táblázatokból, ahol az X tengelyen a dátum, Y tengelyen az aznapi összbevétel szerepel.

Működés:

- Átnavigál az excelek mappába
- Megnyitja az exceleket egyesével, azokon belül összeadja az Ár oszlopot, majd elmenti Y koordinátaként
- Az excel nevét elmenti X koordinátaként (A .xlsx kiterjesztés nélkül)
- A Matplotlib könyvtár segítségével ábrázolja a grafikont.

#### 3.4.2 Parancsok használata összefoglalva

Telepítés utáni ajánlott sorrend:

- python manage.py idelete – Szabad időpontok törlése
- python manage.py igenerate – Első időpontok generálása
- python manage.py stat – Grafikont generál az excelekből (Y tengely = bevétel, X tengely = dátum)
- python manage.py runserver – Szerver indítása

Karbantartási rutin (hetente vagy havonta):

- python manage.py igenerate – Új időpontok generálása
- python manage.py mytask – Múltbéli foglalások archiválása

Automatizálás Windows-on:

A mytask parancs futtatható automatikusan az elmultidopontoktorlese.bat fájjal, amelyet a Windows Feladatütemező napi rendszerességgel indíthat.

Megjegyzés: Ezek az egyedi parancsok a Django BaseCommand osztályból származnak, így ugyanúgy működnek, mint a beépített Django parancsok (például runserver, migrate).

### 3.5 Tesztelés leírása, tesztdokumentáció

A tesztelés manuális volt, különböző környezetekben:

#### 3.5.1 Hardver/szoftver követelmények:

- Processzor: Intel i5/i7 vagy AMD Ryzen 5/7
- Memória (RAM): Minimum 8GB
- Képernyőfelbontások, amelyeket teszteltünk:
  - Asztali gép: 1920x1080 pixel (Full HD)
  - Laptop: 1366x768 pixel (HD)
  - Tablet: 768x1024 pixel (iPad méret)
  - Telefon:
    - 390x844 pixel - iPhone 14/15 méret
    - 360x800 pixel - Samsung Galaxy S23 méret
    - 375x667 pixel - iPhone SE
- Operációs rendszer: Windows 10, Windows 11.
- Böngészők:
  - Google Chrome 120+ verzió
  - Mozilla Firefox 121+ verzió
  - Microsoft Edge 120+ verzió

#### 3.5.2 Teszt felhasználók

A különböző funkciók teszteléséhez különböző jogosultságok kellenek. Az adminnak több mindent szabad csinálni, ezért a teszteléshez két különböző jogosultságú felhasználót hoztunk létre:

1. Admin (Adminisztrátor) felhasználó:

- Email: admin@localhost.local
- Jelszó: AdminTest123
- Jogosultságok: Mindent csinálhat (foglalások törlése, új szolgáltatások hozzáadása stb.)

## 2. Normál (Átlagos) felhasználó:

- Email: teszt@localhost.local
- Jelszó: TestUser123
- Jogosultságok: Csak a saját foglalásait kezelheti

### 3.5.3 Tesztelési esetek

Itt néhány tesztet részletesen leírunk: mit csinált a tesztelő, mi volt a várt eredmény, és mi történt valójában.

#### Teszt 1: Sikeres regisztráció

Célja, hogy új felhasználó tud-e regisztrálni.

Lépések:

1. Megnyitottuk a regisztrációs oldalt
2. Kitöltöttük az űrlapot:
  - Felhasználónév: teszt\_ujfelhasznalo\_2025
  - Email: ujfelhasznalo@test.hu
  - Jelszó: ValidPass123!
  - Jelszó megerősítés: ValidPass123!
3. Rákattintottunk a "Regisztráció" gombra

Várható eredmény: Sikeresen létrejön az új felhasználó, átirányítás a sikeres regisztráció oldalra.

Aktuális eredmény: Működik – Az új felhasználó létrejött az adatbázisban, sikeres oldal jelent meg.

#### Teszt 2: Foglalt felhasználónévvel való regisztráció

Célja, hogy a rendszer nem engedi meg, ha valaki olyan felhasználónevet választ, ami már létezik.

Lépések:

1. Megnyitottuk a regisztrációs oldalt
2. Kitöltöttük az űrlapot egy már létező felhasználónévvel:
  - Felhasználónév: teszt (ez már létezik!)
  - Email: ujfelhasznalo2@test.hu
  - Jelszó: ValidPass123!
  - Jelszó megerősítés: ValidPass123!
3. Rákattintottunk a "Regisztráció" gombra

Várható eredmény: Hibaüzenet jelenik meg: "A felhasználónév már foglalt"

Aktuális eredmény: Működik – A megfelelő hibaüzenet jelent meg, nem engedte a regisztrációt.

- Email: ujfelhasználó@test.hu
- Jelszó: ValidPass123!

Várható eredmény: Hibaüzenet jelenik meg: "A felhasználónév már foglalt"

Aktuális eredmény: Működik – A megfelelő hibaüzenet jelent meg, nem engedte a regisztrációt.

A teljes tesztelés további 18 tesztből állt, amelyek mind sikeresen lefutottak. A dokumentáció terjedelme miatt ezeket itt nem részletezzük a fentebb leírt részletességgel, de minden funkció (foglalás, törlés, bejelentkezés, API, böngésző kompatibilitás, mobilnézet) tesztelve lett:

- Regisztráció: Helyes adatokkal sikerül, duplikált felhasználónévvel hiba.
- Bejelentkezés: Helyes jelszóval sikerül, helytelennel hiba.
- Foglalás: Szabad időponttal sikerül, foglalttal vagy túl korai időponttal hiba.
- API: GET /app/api/szabadidopontok/ visszaadja a szabad időpontokat JSON-ban.
- Hibakezelés: Hiányzó adatoknál a megfelelő hibaüzenetek jelennek meg.



- Teszt felhasználó: Felhasználónév: testuser, jelszó: testpass123 (admin panelen létrehozva).
- Teljesítmény: Kis adatbázissal (néhány rekord) gyors, nagyobbal is elfogadható lenne.

### 3.5.4 Tesztelési eredmények összegzése

- Összes teszt: 20 darab
- Sikeres tesztek: 20 darab
- Sikertelen tesztek: 0 darab
- Sikerességi arány: 100%

Következtetés: Minden tesztelt funkció megfelelően működik. Az alkalmazás stabil, különböző eszközökön és böngészőkben egyaránt használható. A regisztráció, bejelentkezés, foglalás, törlés, API, valamint a reszponzív megjelenés mind megfelelően működnek különböző böngészőkben (Chrome, Firefox, Edge) és eszközökön (laptop, mobil, tablet). Az alkalmazás minden esetben helyesen működött, hibák esetén megfelelő üzenetekkel reagált.

### 3.6 Fejlesztési lehetőségek ismertetése

- Az alkalmazás jelenleg teljes mértékben működőképes, de számos fejlesztési lehetőség van, amelyek még jobbra tennék. Ebben a részben néhány ötletet mutatunk be, amelyeket a jövőben meg lehetne valósítani.
- Email értesítés: amikor valaki lefoglal egy időpontot, automatikusan kapjon egy megerősítő emailt. Hasonlóképpen arról is, ha valaki törli a foglalást.
- SMS értesítés: email értesítéshez hasonló okokból, de nehezebb figyelmen kívül hagyni.
- Fizetés integráció: Stripe vagy PayPal hozzáadása az árakhoz.
- Több nyelv: Django i18n hozzáadása magyar-angol váltáshoz.
- Csoportos foglalás: Több ember is tudjon egyszerre foglalni ugyanarra az időpontra (pl. tanfolyamokat lehetne szervezni, csoportos kedvezményt lehetne adni)
- Google Naptár integráció: A lefoglalt időpontok automatikusan megjelenjenek az admin Google Naptárában.
- Értékelési rendszer: Az oktatás után a felhasználók értékelhessék a szolgáltatást (1-5 csillag, szöveges véleményezés)

## 4. Felhasználói dokumentáció

### 4.1 A program céljának és lényegesebb funkcióinak összefoglalása

A „Kovász Mentor” egy internetes oldal (webalkalmazás) időpontfoglaló rendszerrel a házi készítésű pékáruk szerelmeseinek. Célja, hogy személyre szabott oktatást nyújtson a kovással való sütésről. Főbb funkciók: regisztráció, bejelentkezés, szolgáltatások böngészése, időpontfoglalás, profilkezelés.

### 4.2 Szükséges hardvereszközök és szoftverek felsorolása

- Hardver: Számítógép vagy mobileszköz internet-hozzáféréssel.
- Szoftver: Modern webböngésző (Chrome, Firefox, Edge), JavaScript engedélyezve.

### 4.3 Telepítés és indítás lépéseinek ismertetése

- Töltse le a projektet: `git clone https://github.com/Adri959/teszt5555`
- A projekt mappájában nyisson egy parancssort
- Hozza létre a virtuális környezetet: `'python -m venv .'`
- Aktiválja a virtuális környezetet: `'Scripts\activate'`
- Telepítse a függőségeket: `'pip install -r requirements.txt'`
- Navigáljon a backend mappába: `'cd backend'`
- Futtassa a migrációkat: `'python manage.py migrate'`
- Indítsa el a szerveret: `'python manage.py runserver'`
- Nyissa meg a böngészőben: `http://127.0.0.1:8000/`

## 4.4 Az alkalmazás részletes bemutatása

### 4.4.1 Kezdőoldal (Főoldal)



Ábra 2: Kezdőlap

Az alkalmazás kezdőlapján jól láthatóan szerepel az alkalmazás neve és egy rövid ismertető. A felületről közvetlenül indítható az oktatásra történő jelentkezés, továbbá elérhetőek a Belépés és a Regisztráció gombok. A fejlécben elhelyezett felső navigációs sáv segítségével a weboldal releváns szekciói érhetőek el, lefelé görgetve a felület ugyanígy a megfelelő tartalmi egységekhez navigál, melyek a következők.

Az oktató rövid bemutatkozása:



### Bemutatkozás

András vagyok. A kenyérfőzési utam egy emésztési problémából indult, ami miatt el kellett hagynom a bolti pékárukat. Eleinte nehezen szántam rá magam az otthoni sütésre, és be kell vallanom, több volt a kudarc, mint a siker. De nem adtam fel — tanultam, kísérleteztem, és a kenyerek egyre jobbak lettek.

Később beiratkoztam egy pék képzésre, ahol fantasztikus csapat gyűlt össze. Ebben a közegben igazán fellángolt bennem a kovász-szenvedély, és hamarosan világossá vált: ez nem csupán hobbi, hanem hivatás. Ma már saját pékséget vezetek, és időről időre megnyitom a műhely ajtaját, hogy másokkal is megoszthassam az évek során megszerzett tapasztalataimat.

Ábra 3: Bemutatkozás

Miért előnyös a kovással való sütés:

## Miért kovással?

- Elérhetővé teszi az értékes tápanyagokat
  - Könnyebben emészthető
- Alacsonyabb a glikémiás indexe
  - Prebiotikumokat tartalmaz
- Hosszabb az eltarthatósága
  - Adalékanyag-mentes



Ábra 4: Miért kovással?

Bővebb információk a szolgáltatásról:

## Oktatás

barátságos légkör

hasznosítható tudás

gyakorlás és sikerélmény

válaszok a kérdéseidre

120 perc

12 000 Ft



### Témakörök:

- Kováskészítés & etetés
- Gyúrás, dagasztás, hajtogatás
- Kelesztés, formázás, sütés
- Alapanyagok összeválogatása
- Speciális eszközök & praktikák
- Gyakori hibák & megoldások

Az órákat általában egy rövid **elméleti** résszel kezdjük, hogy megértsd a kovász és a kenyértészta működését, így a tanultakat nem kizárólag recepteket követve, hanem rugalmasan is alkalmazhatod.

Majd a **gyakorlati** részben végigkísérlek a kenyérfőzés teljes folyamatán: minden lépést meg fogok mutatni, el fogok magyarázni, és segíteni foglak a profi és magabiztos mozdulatok elsajátításában.

Az magánórák tempóját és tematikáját a tudásszintedhez és **az igényeidhez igazítom**, hogy pontosan azt kaphasd, amire szükséged van. Az a célom, hogy minden egyes óra a fejlődésedet szolgálja.

Ábra 5: Oktatás

Felhasználói vélemények:



Ábra 6: Vélemények

Az időpontfoglaláshoz szükséges regisztráció, illetve bejelentkezés. A lap tetején lévő „Jelentkezés” gomb ide navigál:

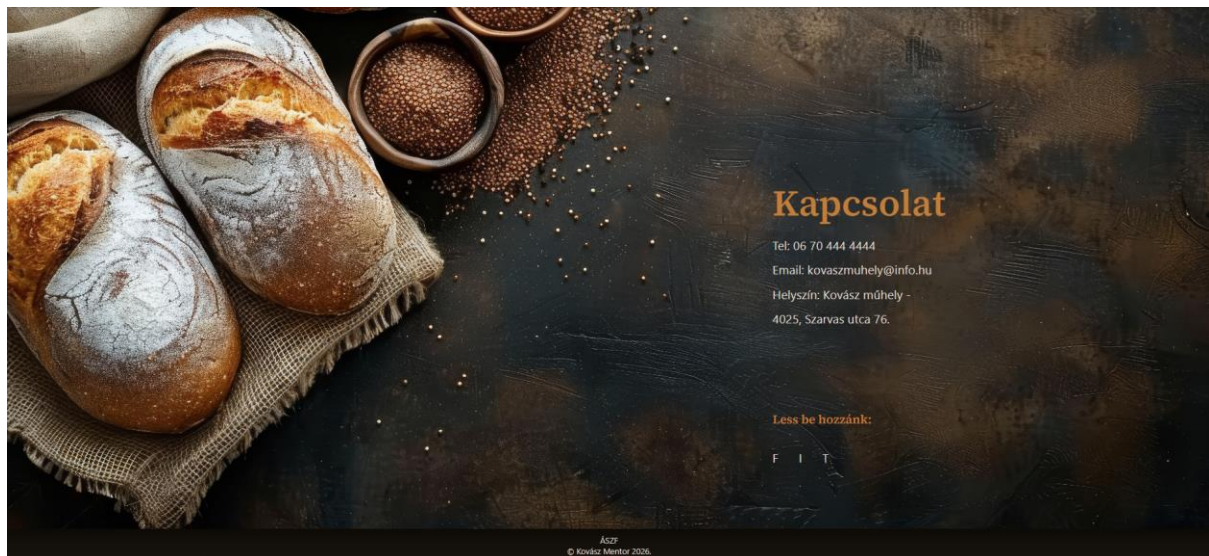


Ábra 7: Jelentkezés



- „Regisztráció” gomb: Erre kattintva új felhasználó fiók hozható létre.
- „Belépés” gomb: Erre kattintva beléphetsz a meglévő fiókkal.

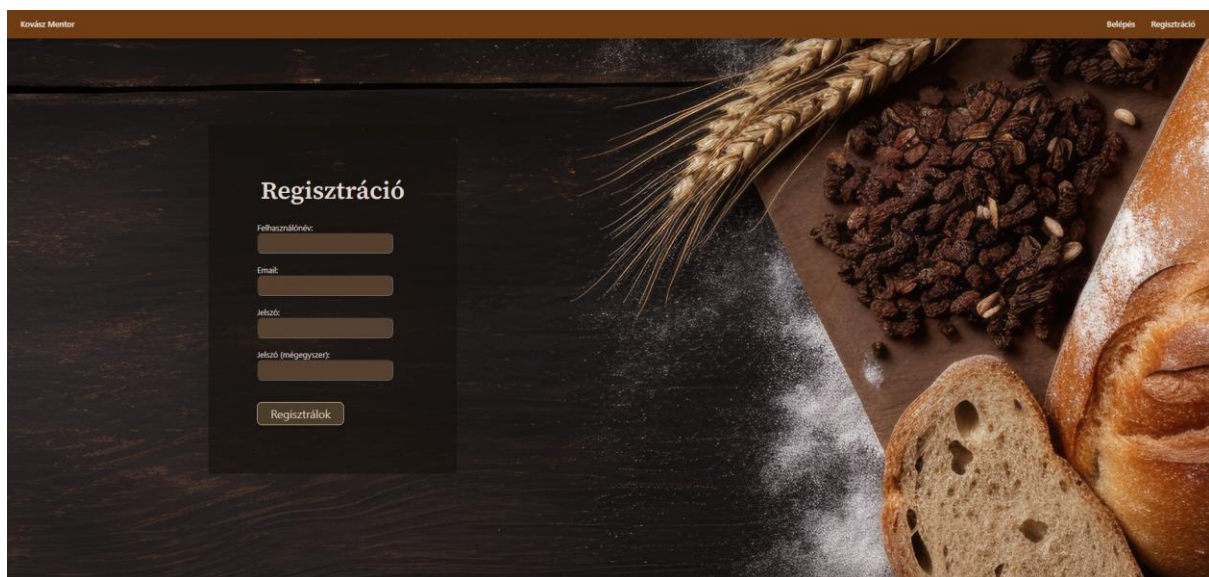
Kapcsolati rész, az összes elérhetőséggel:



Ábra 8: Kapcsolat

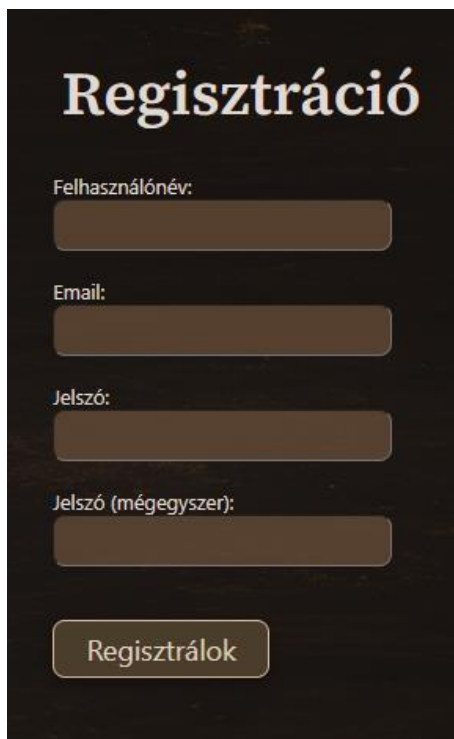
#### 4.4.2 Regisztrációs oldal

Az alkalmazás használatához először regisztrálnod kell.



Ábra 9: Regisztráció

Kitöltendő mezők:



The image shows a registration form titled "Regisztráció" (Registration) in a stylized font. Below the title, there are four input fields with labels: "Felhasználónév:" (Username), "Email:", "Jelszó:" (Password), and "Jelszó (még egyszer):" (Password (confirm)). Each field is represented by a dark brown rounded rectangle. At the bottom of the form is a button labeled "Regisztrálok" (Register).

Ábra 10: Regisztráció mezők

#### 1. Felhasználónév

- Hossz: minimum 1, maximum 150 karakter
- Nem lehetnek azonos felhasználónevek
- Példa: "kovaszmentor2026"

#### 2. Email cím

- Érvényes email cím formátum (pl. "email@example.com")
- Nem lehetnek azonos email címek
- Ezt fogja használni a bejelentkezésre is

#### 3. Jelszó

- Minimum 8 karakter hosszú
- Nem állhat csak számokból, vagy csak betűkből
- Nyilvánosan nem látható, helyette pontok jelennek meg

- Példa: "KovaszMentor2026!"

#### 4. Jelszó megerősítése

- Ugyanaz, mint amit az előző mezőben írtunk
- Az alkalmazás ellenőrzi, hogy megegyezik-e

Hibajelzések és kezelésük:

- "A felhasználónév már foglalt" - Válassz másik felhasználónevet!
- "Az email cím már foglalt" - Használj másik email címet!
- "A jelszavak nem egyeznek!" - Biztosítsd, hogy mindkét jelszó mező azonos legyen!
- "A jelszó túl rövid", "A jelszónak szükséges számokat tartalmaznia stb." - Válaszd meg a jelszót az útmutató szerint!

A regisztráció után automatikusan átkerülsz a foglalási oldalra.

#### 4.4.3 Bejelentkezési oldal

Ha már regisztrálva vagy, a bejelentkezési oldalon tudsz belépni.

Kitöltendő mezők:



Belépés

Email:

Jelszó:

Belépek

[Regisztráció](#)

Ábra 11: Belépési adatok



1. Email cím: Az a cím, amelyet a regisztrációnál megadtál.

2. Jelszó: Az a jelszó, amelyet a regisztrációnál állítottál be.

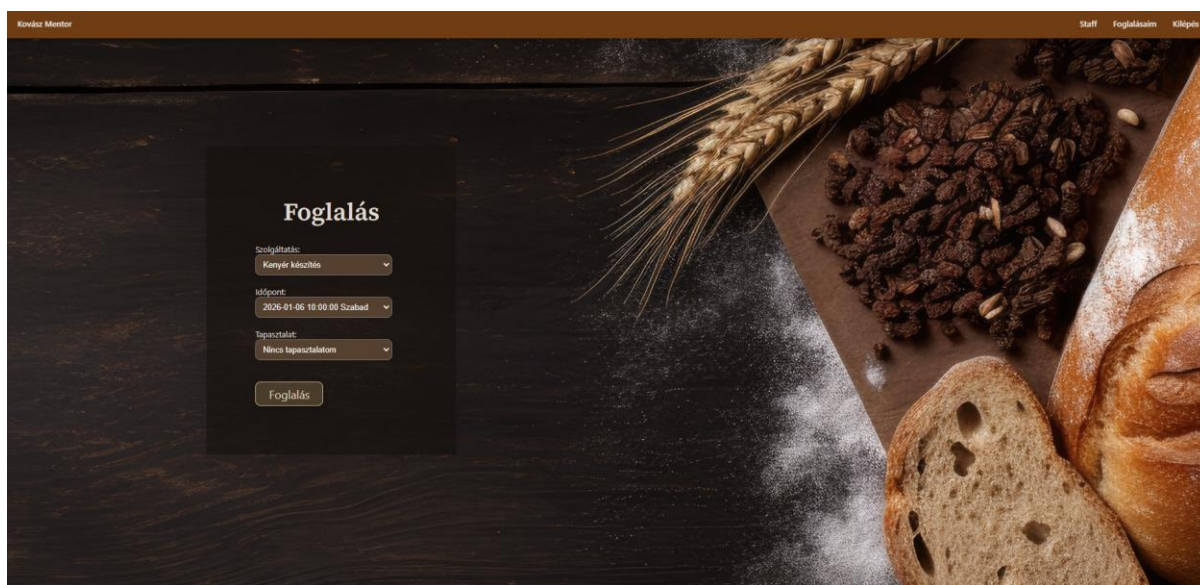
Hibajelzések:

- "Hibás bejelentkezési adatok" - Ellenőrizd, hogy helyesen írtad-e az email és jelszó kombinációt!

A sikeres bejelentkezés után a foglalási oldalra jutsz.

#### 4.4.4 Foglalási oldal (Fő funkció)

Ez az alkalmazás legfontosabb oldala. Itt tudod megtekinteni az elérhető időpontokat és a foglalást végrehajtani.



Ábra 12: Foglalás

A foglaláshoz szükséges adatok megadása legördülő menüből:



The image shows a dark-themed booking form titled "Foglalás" in a large, light-colored serif font. Below the title, there are three dropdown menus, each with a label and a light-colored button containing the selected option and a downward arrow. The first dropdown is labeled "Szolgáltatás:" and has "Kenyér készítés" selected. The second is labeled "Időpont:" and has "2026-01-15 19:00:00 Szabad" selected. The third is labeled "Tapasztalat:" and has "Nincs tapasztalatom" selected. At the bottom of the form is a light-colored button with the text "Foglalás".

Ábra 13: Foglalás kiválasztása

Az oldal felépítése:

### 1. Szolgáltatások választása

- Elérhető szolgáltatások listája jelenik meg: "Kenyérkészítés", "Kovászkészítés" stb.
- Válaszd ki azt, amelyre szeretnél időpontot foglalni

### 2. Elérhető időpontok

- Az adott szolgáltatásra elérhető szabad időpontok
- Felépítés: Dátum (év-hó-nap) + Időpont (óra:perc)
- Csak azok az időpontok jelennek meg, amely legalább 2 órával később vannak (ha mai napról van szó)
- Korábbi vagy már foglalt időpontok nem jelennek meg

### 3. Időpont kiválasztása

- Válassz egy időpontot az elérhető listából

- Kattints az adott időpontra (vagy válassz egy rádió gomb/checkbox segítségével)

#### 4. Tapasztalat mező

- Válaszd ki, hogy mennyi releváns tapasztalattal rendelkezel

#### 5. "Foglalás" gomb

- Kattints erre a gombra a foglalás végrehajtásához

#### 4.4.5 Sikeresnek ítélt foglalási oldal

A foglalás után ezt az oldalt látod, amely megerősíti a sikeres foglalást, majd visszavezet a kezdőoldalra vagy a profil oldalra.



Ábra 14: Sikeres foglalás

Hibás foglalás esetén:

- Hibaüzenet jelenik meg (pl. "Az időpont legalább 2 órával későbbi kell legyen!")
- Az oldal nem változik meg, újra próbálkozhatsz

#### 4.4.6 Profil oldal

A bejelentkezett felhasználó a saját foglalásait itt tekintheti meg.



Ábra 15: Foglalásaim

Az oldal tartalma:

Egy táblázat a felhasználó összes foglalásaival:

- Dátum és időpont - Szolgáltatás neve
- "Törlés" gomb - Erre kattintva törölhető a foglalás

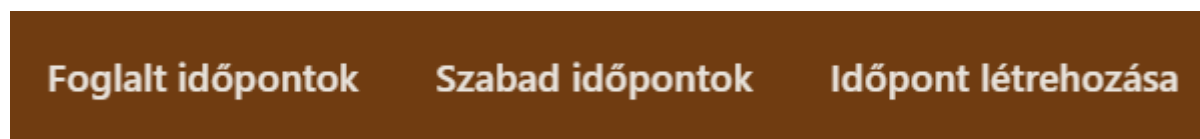
Törlés után:

- A foglalás törlésre kerül, az időpont felszabadul
- Visszakerülsz a profil oldalra

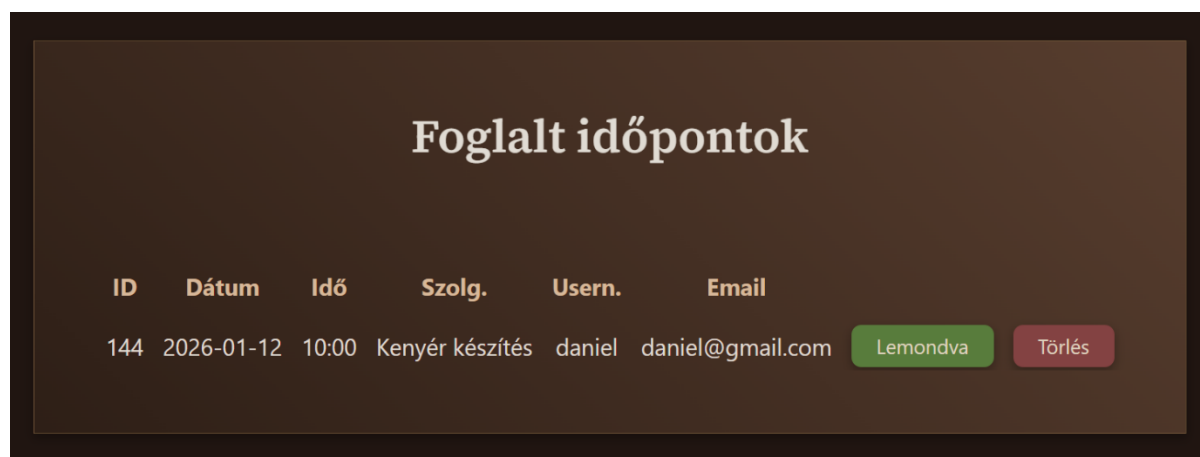
#### 4.4.7 Staff oldalak

Ezek az oldalak csak Staff jogosultságú felhasználók számára érhetőek el, ha a Staff gombra kattintanak.

A navbar megváltozik, és három új menüpont érhető el.



[Foglalt időpontok oldal](#)



Itt megtekinthető az összes foglalás.

- Lemondva gomb: Törli a foglalást, az időpont újra elérhetővé válik.
- Törlés gomb: Törli a foglalást és az időpontot is.

[Szabad időpontok oldal](#)



Itt megtekinthetők és törölhetők a még nem foglalt időpontok.

- Összes időpont törlése gomb: Törli az összes jövőbeli szabad időpontot.
- Törlés gomb: Törli az adott időpontot.

The image shows two side-by-side form panels with a dark brown background and light brown text. The left panel is titled 'Időpont hozzáadás' (Add time slot) and contains input fields for 'Dátum:' (Date) with the value '2026-01-10' and 'Idő:' (Time) with the value '21:00'. Below these is a 'Create' button. The right panel is titled 'Több időpont generálása' (Generate multiple time slots) and contains input fields for 'Hány napra:' (For how many days) with the value '10', 'Lépésköz:' (Interval) with the value '3', and 'Kezdeti dátum:' (Start date) with the value '2026-01-10'. Below these is a 'Generate' button.

Időpont hozzáadása form:

- Egy darab szabad időpontot kreál, a formátum a placeholdernek megfelelő kell legyen. (dátumnál: ÉÉÉÉ-HH-NN, időnél: ÓÓ-PP)

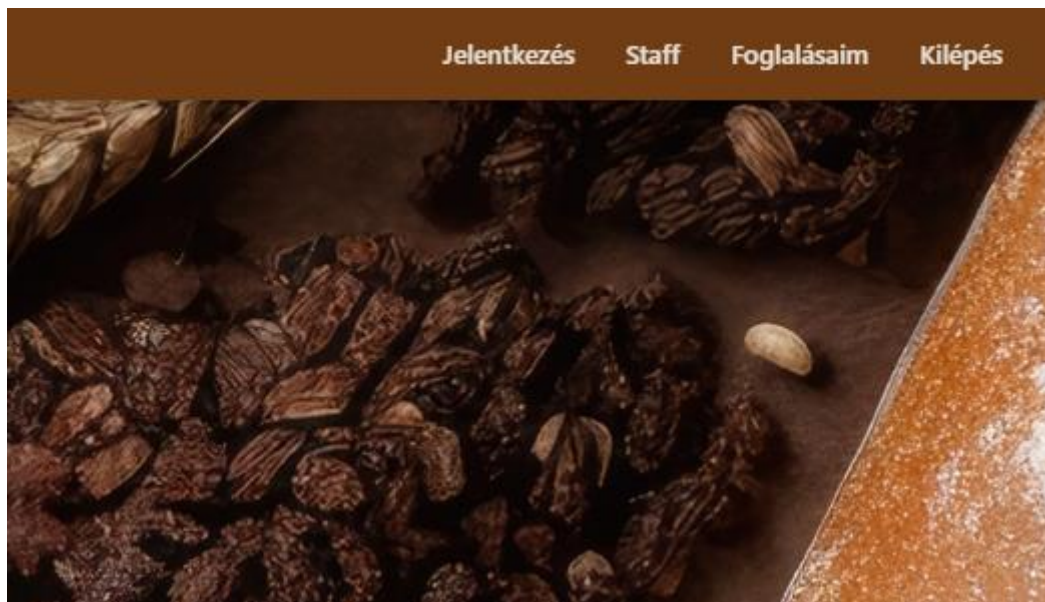
Több időpont generálása form:

Generate gomb:

- 10 napra előre 3 óránként (10 és 20 óra között) generál időpontokat a mai naptól kezdve.
- Hogy hány napra, milyen lépésközzel és melyik naptól kezdve generáljon opcionálisan módosítható. (Ahol nincs megadva érték, ott az alapértelmezett érték lesz)

#### 4.4.8 Kilépés

A böngészés végeztével a jobbfelső sarokban lévő „Kilépés” gombra kattintva lehet kijelentkezni:



Ábra 16: Kilépés

Ennek hatására visszavigálunk a kezdő oldalra.

#### 4.5 Hibajelzések magyarázata

- | Hibajelzés | Oka | Megoldás |
- | "A felhasználónév már foglalt" | Ez a felhasználónév már létezik | Válassz másik felhasználónevet |
- | "Az email cím már foglalt" | Már regisztrált valaki ezzel az emaillel | Használj másik email címet |
- | "A jelszavak nem egyeznek!" | A jelszó és megerősítése nem azonos | Biztosítsd, hogy mind kettő ugyanaz |
- | "A jelszó túl rövid" | Kevesebb mint 8 karakter | Legalább 8 karaktert használj |
- | "Hibás bejelentkezési adatok" | Hibás email vagy jelszó | Ellenőrizd a helyességét
- | "Sikertelen foglalás! Az időpont legalább 2 órával..." | Túl közeli időpontot választottál mai napra | Válassz 2 órán túli vagy holnap időpontot |
- | "404 Not Found" | Az oldal nem létezik | Ellenőrizd az URL-t |

## 5. Összefoglalás

Az elkészült munka sikeresen megvalósítja a célkitűzéseket: működő webalkalmazás időpontfoglalással. A legnagyobb kihívás az időpontgenerálás és a nyitóképernyő megformázása volt, de sokat tanultunk Django-ból és webfejlesztésből. A program hasznosulhat oktatási célokra, és továbbfejleszthető kereskedelmi használatra.

### 5.1 Irodalomjegyzék:

- Django dokumentáció: <https://docs.djangoproject.com/en/5.1/>
- Django REST Framework: <https://www.django-rest-framework.org/>
- Python hivatalos oldal: <https://www.python.org/>
- Péter Miklós weblapja: <https://miklospeter.wordpress.com/>
- Infojegyzet.hu dokumentáció útmutató:  
<https://infojegyzet.hu/webszerkesztes/dokumentacio/>
- W3Schools: <https://www.w3schools.com/>
- HTML, CSS, JavaScript referencia: <https://developer.mozilla.org/>
- Dbdiagram: <https://dbdiagram.io/home>