**Dokumentáció: dineR**

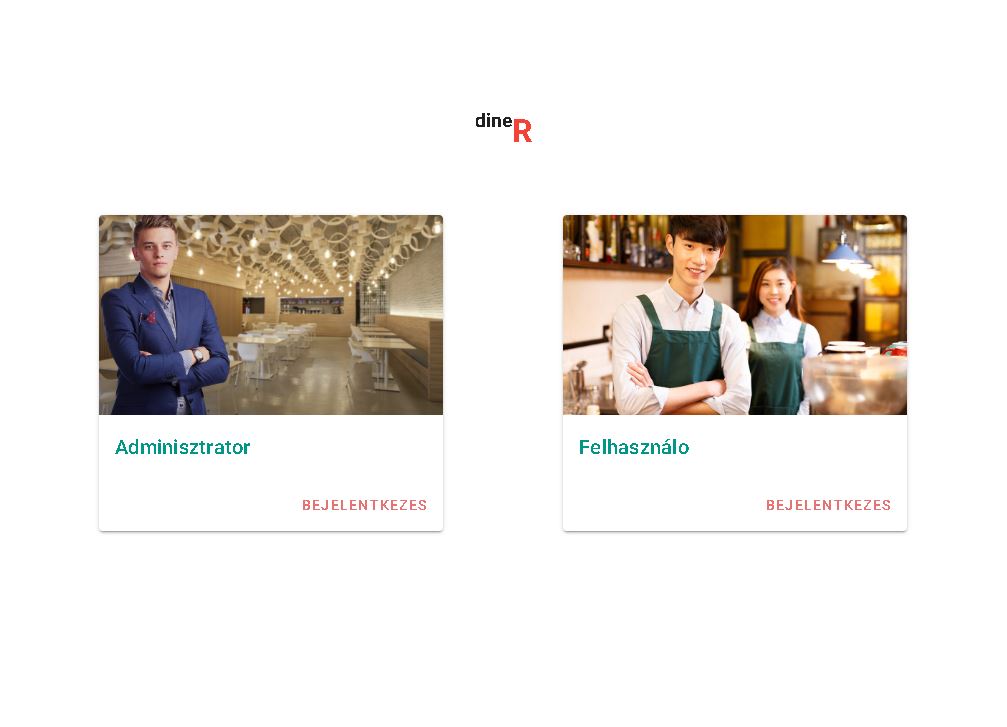


**Bevezető:**

A dineR applikáció az olyan vendéglátóiparban tevékenykedő kisebb éttermek csoportját célozza meg, ahol még nem találtak informatikai megoldást a mindennapi feladatok szimulációjára. Különösképp a felszolgálókra és az ő feladatkörükre fókuszáltam az applikáció tervezése és írása közben. A cél, hogy a hagyományos rendelésfelvételt felváltsa egy valamelyest innovatív megoldás, amely rugalmasan tud alkalmazkodni fejlett életviteleinkhez.

**Front-end:**

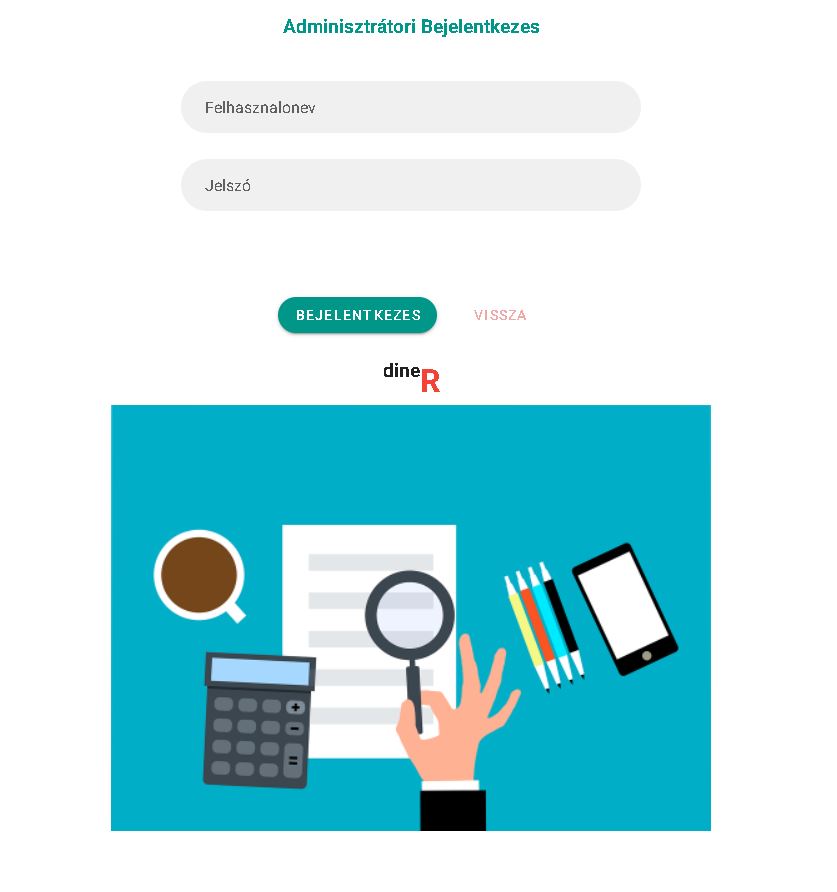
**Bejelentkező-kezdőoldal**



Az applikáció természetéből adódóan két fő nézetre oszlik, adminisztrátor és felhasználói nézetre. Ezen nézetek külön bejelentkezést igényelnek. Az egyes nézetek bejelentkezési képernyőjéhez a Bejelentkező-kezdőoldalon keresztül jut el a felhasználó / adminisztrátor. Mindkét útvonalhoz a Bejelentkezés gombbal lehet eljutni.

**Adminisztrátori Bejelentkezés**

Az Adminisztrátori Bejelentkezés képernyőn nyílik lehetősége az adminisztrátoroknak a bejelentkezésre.



A bejelentkezési képernyő validációval van ellátva, mindkét mező kötelező mező, valamint a Jelszó mező legalább 4 karakter hosszúságú. A validációt a Felhasználói nézet bejelentkezési nézete is használja.

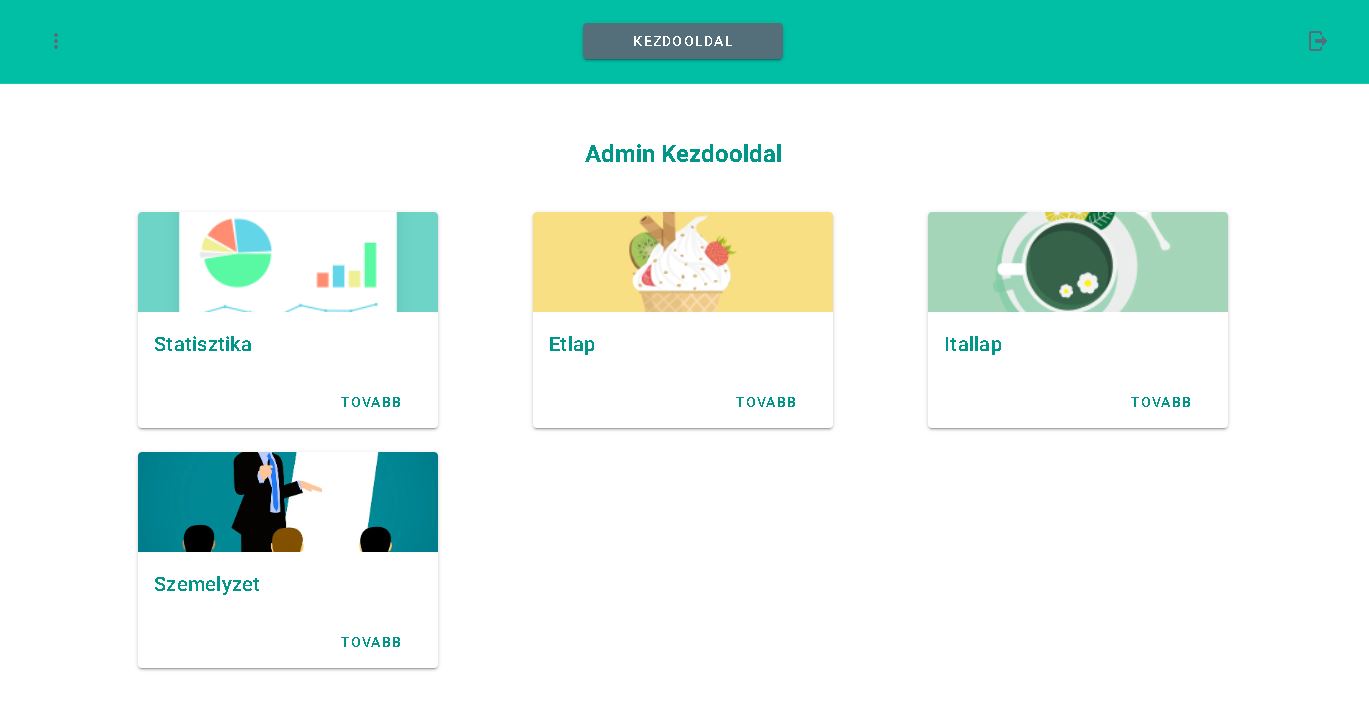




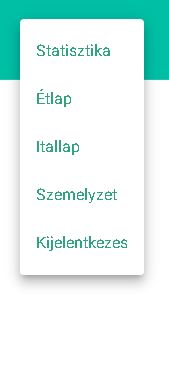
A bejelentkezési kísérlet esetleges sikertelenségéről visszajelzést kapunk.

**Adminisztrátori Kezdőoldal**

Az adminisztrátori nézetbe való bejelentkezés után az adminisztrátori kezdőoldalon 4 gyorsgomb mellett egy menü gomb, egy kijelentkezés gomb, és egy kezdőoldal gomb áll rendelkezésünkre.



A menüből bármely oldalon elérhető közvetlenül az összes többi oldal az adminisztrátori nézetek közül.



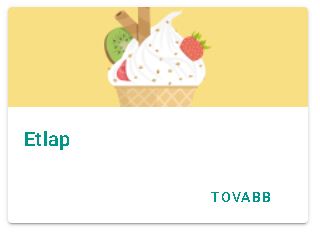
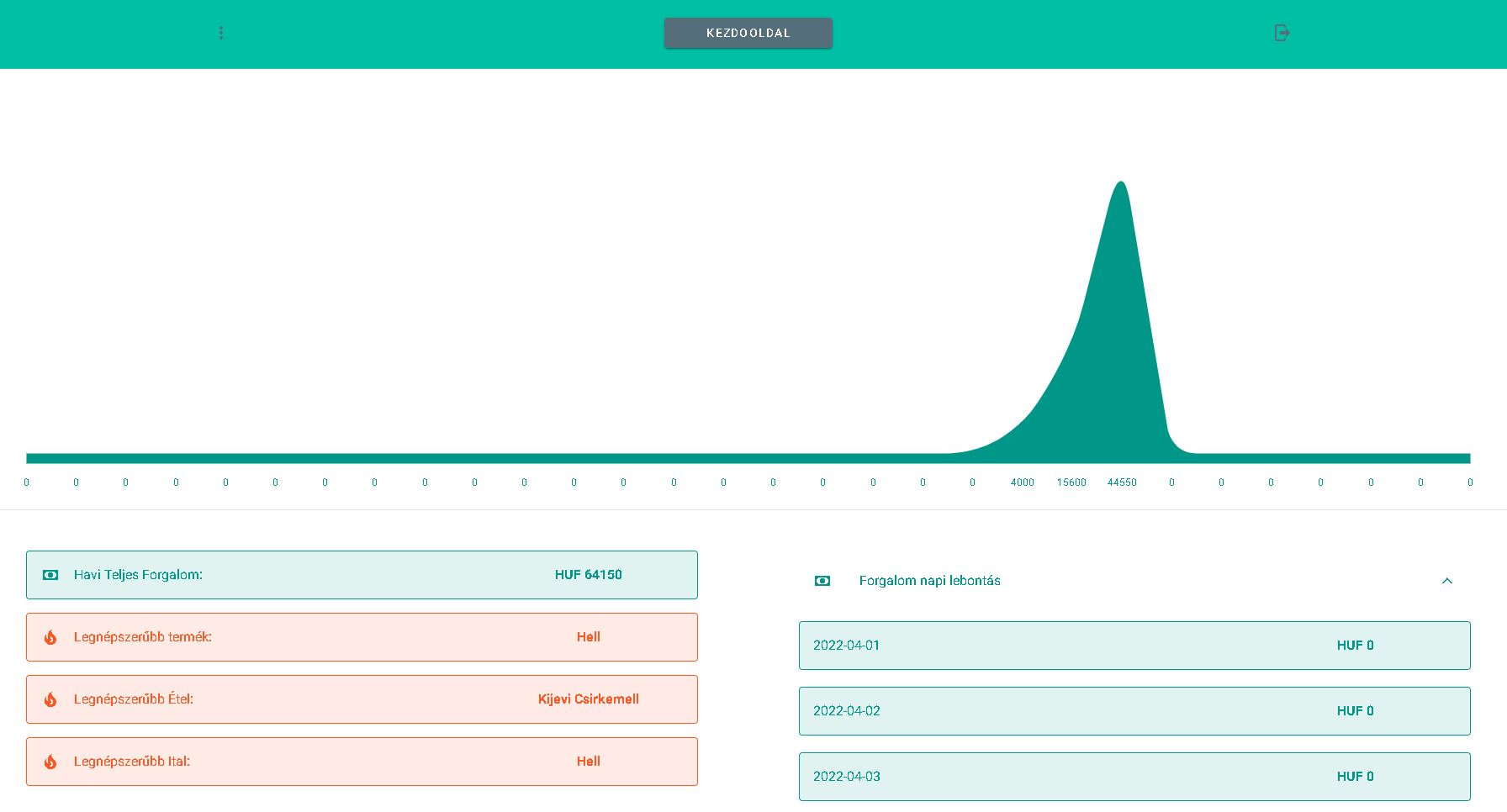
A kijelentkezés gomb segítségével visszakerülünk a Bejelentkező-Kezdőoldalra, ahol nézetet válthatunk vagy újra bejelentkezhetünk.



Az első gyorsgomb egy statisztikai nézetre vezet bennünket a Tovább gombra kattintva.

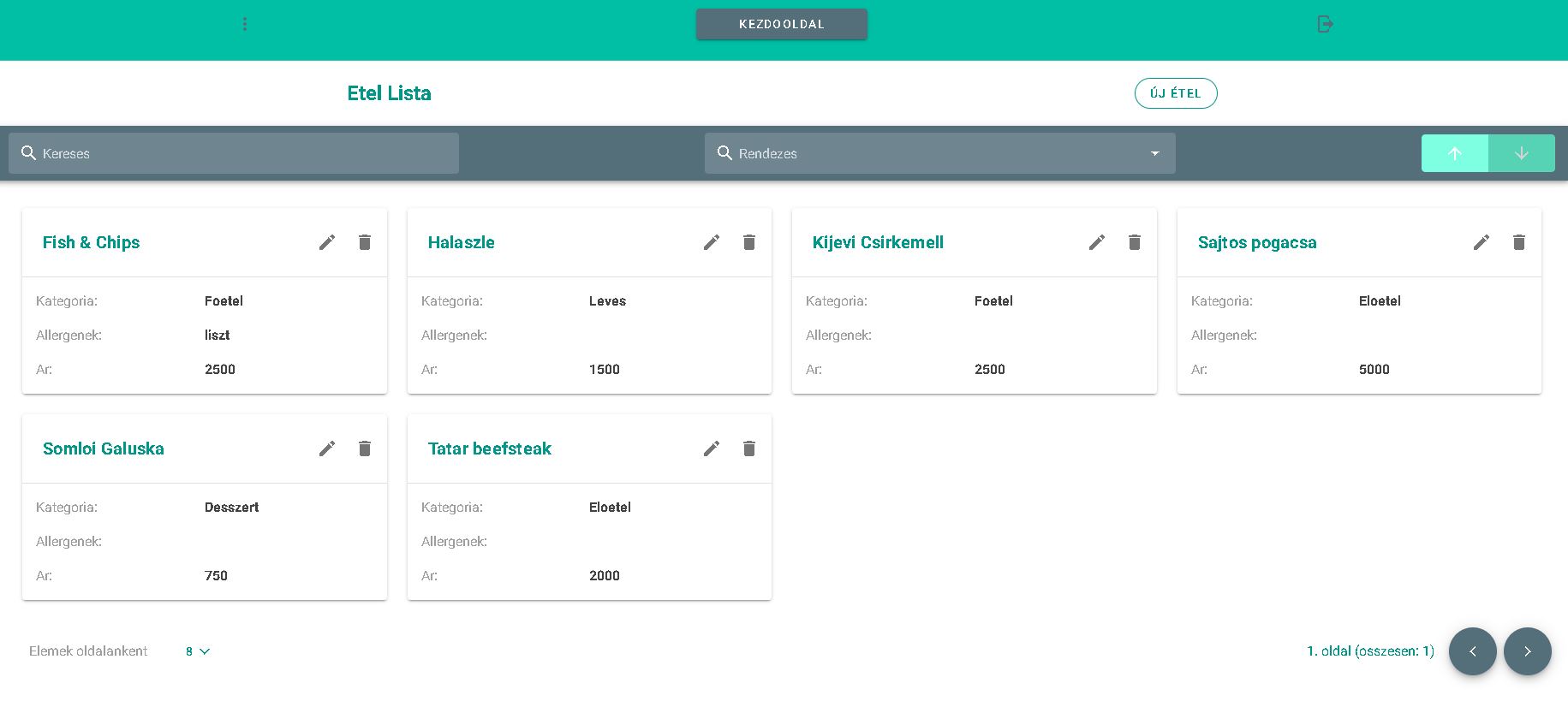
Az adminisztrátori statisztikai nézeten egy grafikon ad visszajelzést a havi bevételek alakulásáról. Erről a Forgalom napi lebontás legördülő gomb lenyitásával listaszerű nézetet kapunk a napi bevételekről.

Továbbá elérhető a havi teljes forgalom összege, valamint a legnépszerűbb   
termékek.  
  
  
  
  
**Adminisztrátori statisztikai oldal**



A következő gyorsgomb a Tovább gombra kattintva az étlaphoz visz.

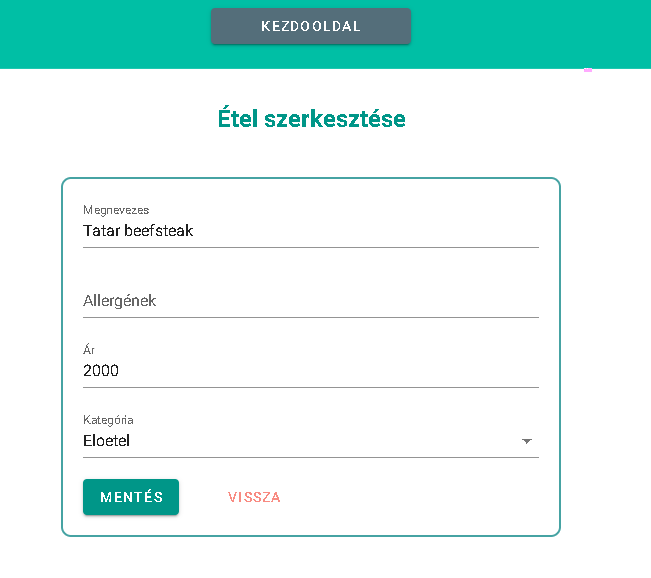
Az étlapon elérhető az összes menüben szereplő étel, melyekhez hozzáadhatunk új ételt, vagy szerkeszthetünk, törölhetünk meglévőt.  
  
  
  
  
  
  
  
  
  
**Adminisztrátori Étlap lista**





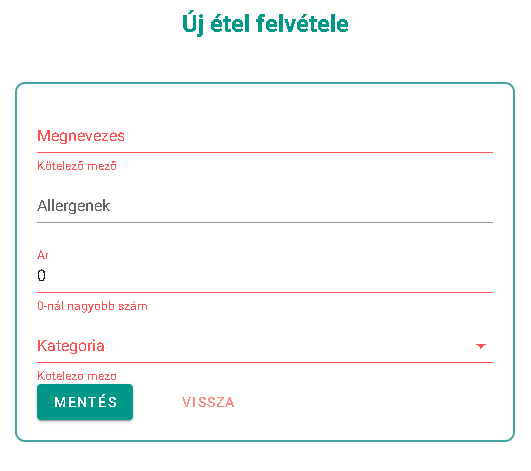
Az ételek kártyák formájában tekinthetőek meg az étel listában, amely a szerkesztés (ceruza) ikonra kattintva szerkesztői nézetet hoz elő, ahol minden részletet látni fogunk a bejegyzésről, olyanokat is, amelyek a listanézetben nem elérhetőek.



**Étel Szerkesztés**  
  
  
  
Szerkesztés után a Mentés gombbal tudjuk a változtatásokat érvényesíteni, vagy a vissza gombbal mehetünk vissza a lista nézetbe.

**Étel hozzáadása étlaphoz**  
Az új étel gomb nyitja meg a létrehozó nézetet, ahol új ételt vehetünk fel. C:\Users\Szőke Lajos\AppData\Local\Microsoft\Windows\INetCache\Content.Word\newFood.jpg

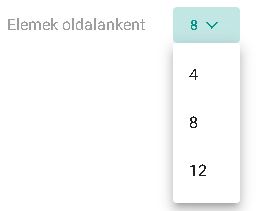
Az új étel felvétele, valamint a meglévő szerkesztése is validációhoz kötött.



Valamennyi listanézet kereshető & szűrhető & rendezhető. Erre a kövezkező gombok állnak rendelkezésre:

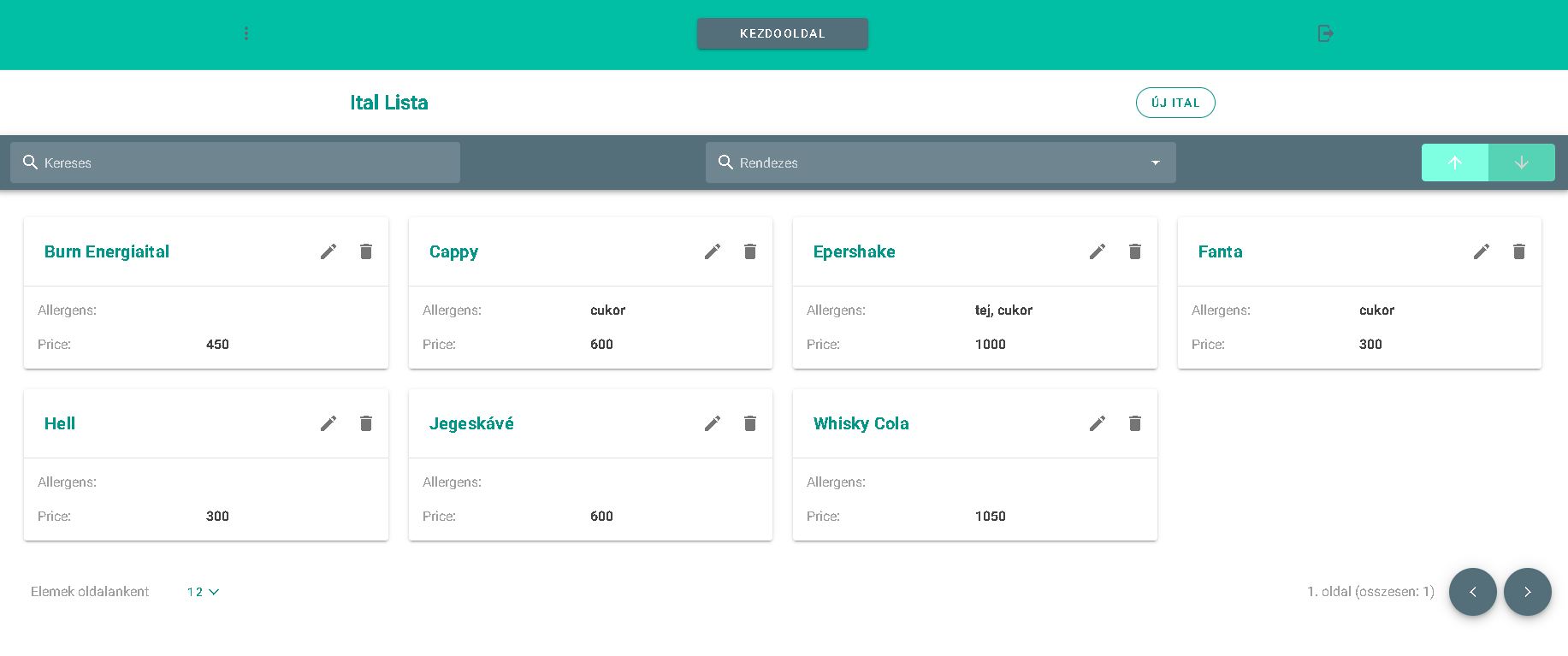
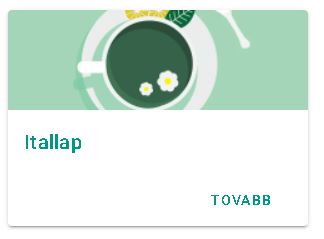


Továbbá az egy oldalon visszaadott elemek számát tudjuk manipulálni, és a létrehozott lapozórendszeren keresztül tudunk lapozni.



**Adminisztrátori Itallap lista**

Az italok listanézetben ugyanazok a lehetőségek elérhetőek mint az étlapok listanézetben.

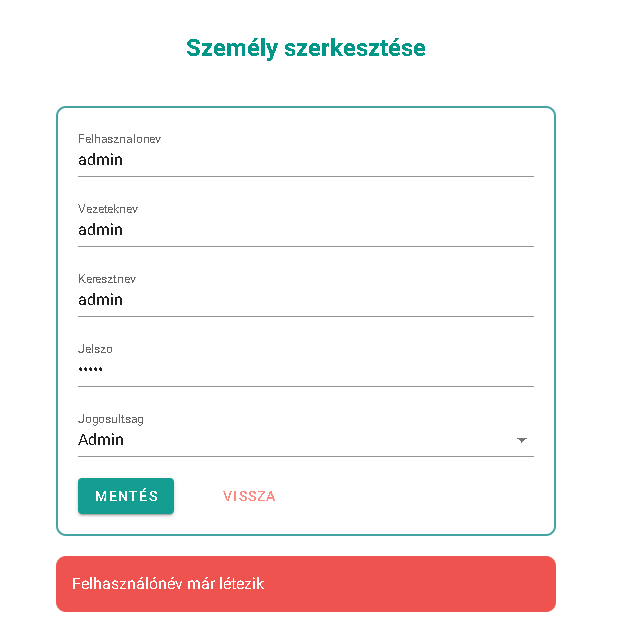


**Adminisztrátori Személyzet lista**

Az adminisztrátorok és felhasználók listáját a Személyzet gyorsgombon keresztül, vagy a menün keresztül is elérhetjük. A lista hasonlóan épül fel, mint az előző listák. A szerkesztési nézetében egy lényeges különbség van az előző listákhoz képest.:



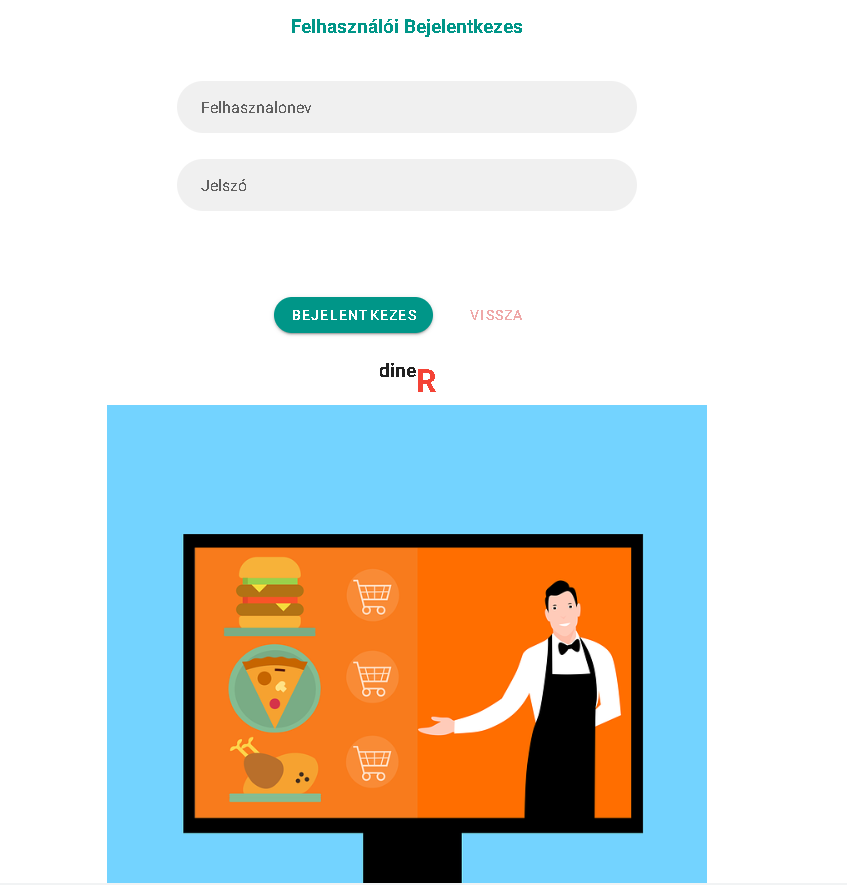
Mentés előtt figyelembe vesszük, hogy létezik-e már hasonló felhasználónév a rendszerben. Ha igen, arról kapunk visszajelzést.  
**Személy szerkesztése**



A felhasználói nézetbe a Bejelentkezési Kezdőképernyőnél a Felhasználó kártyán kell a Bejelentkezésre kattintanunk.  
  
**Felhasználói bejelentkezés**



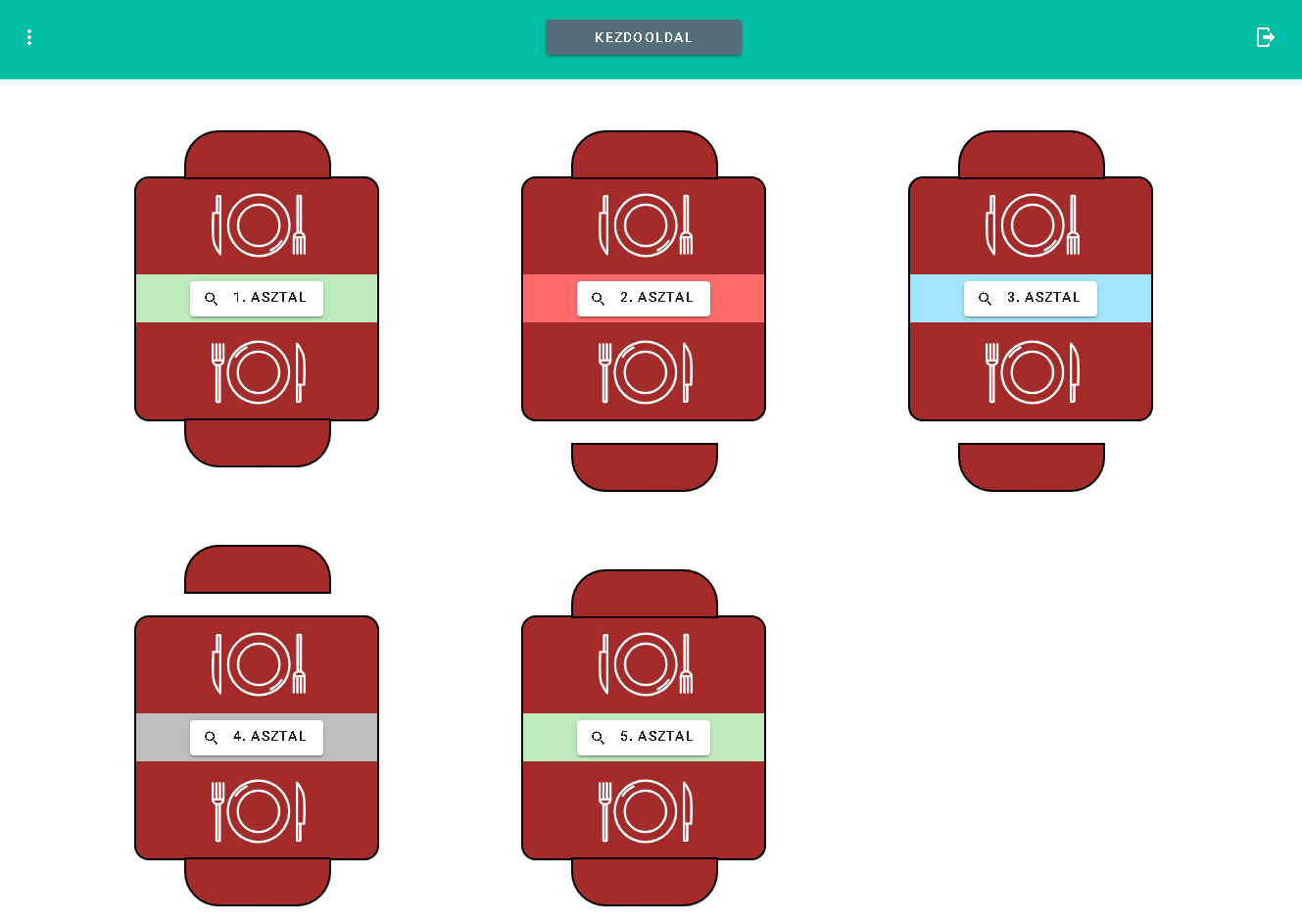
A bejelentkezésre kattintva megnyílik a felhasználókra testreszabott bejelentkezési képernyő.



Validáció és visszajelzés a bejelentkezés sikerességéről hasonlóképpen történik, mint az adminisztrátori bejelentkezésnél.

**Felhasználói Kezdőoldal & Asztalok nézet**

Bejelentkezés után a kezdőoldalon folyamatosan követhetőek az asztalok & székek állapota.



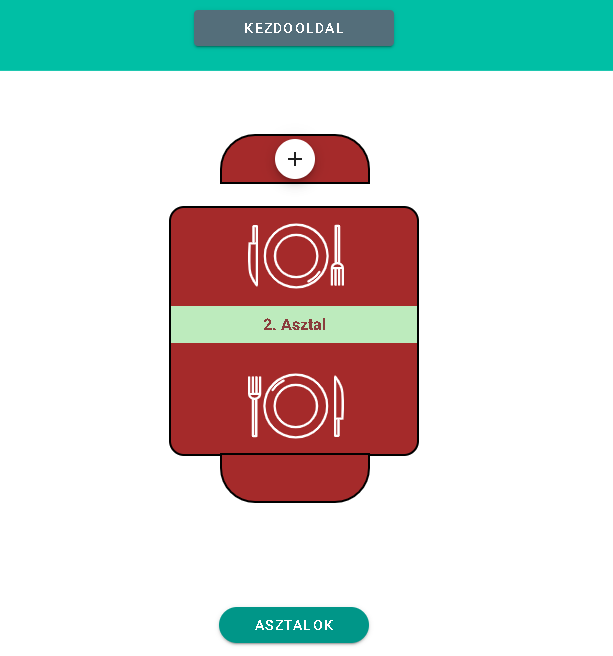
Az asztalok állapotát a terítő színe árulja el. 4 színkódot különböztetünk meg:

* szabad (az asztal készen áll rendelésfelvételre), színkódja: zöld
* rendelésre vár (az asztal tud további rendeléseket fogadni, már meglévő rendelése megtekinthető), színkódja: vörös
* fizetésre vár (az asztalhoz megérkezett minden rendelése, készen áll a fizetésre, számla megtekinthető & fizethető), színkódja: kék
* fizetett (az asztal rendezte a számláját, tisztítása szükséges), színkódja: szürke

A székek nyitott állapotát az asztaltól való távolsága mutatja.

 Egy asztal megnyitása a nagyító gomb segítségével történik. Ezt használva jutunk el egy asztal nézetébe, ahol minden vele kapcsolatos tevékenység lehetségessé válik. Az asztalok nézet korlátozott, ott csak és kizárólag a székek állapotát tudjuk változtatni, ha azok nincsenek zárolt állapotban. Minden további tevékenységhez az asztal nézetet kell használnunk a nagyító gombbal.

**Asztal nézet**

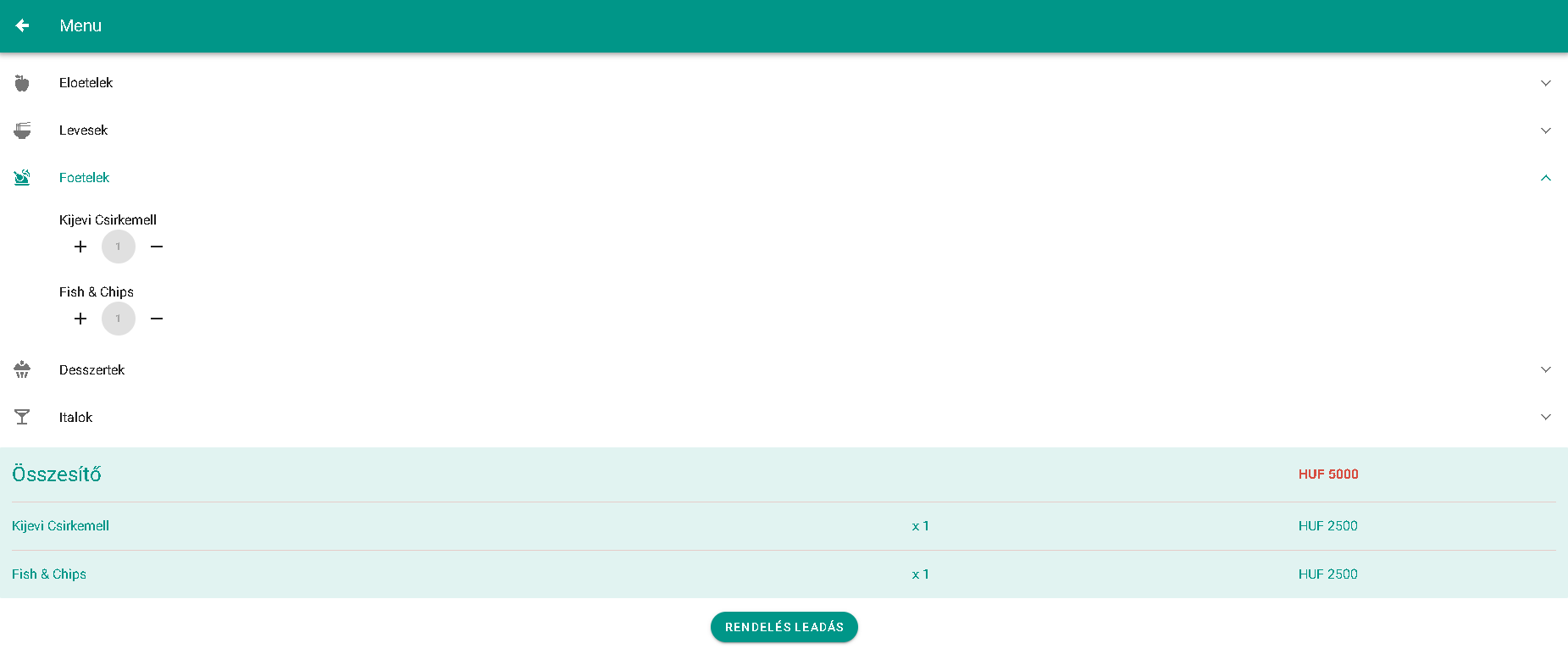


Az asztal nézetben, bármely nyitott állásban lévő székből megnyithatjuk az étlap & itallap menüt. Ugyanitt az Asztalok gomb visszavisz minket az asztalok nézethez.

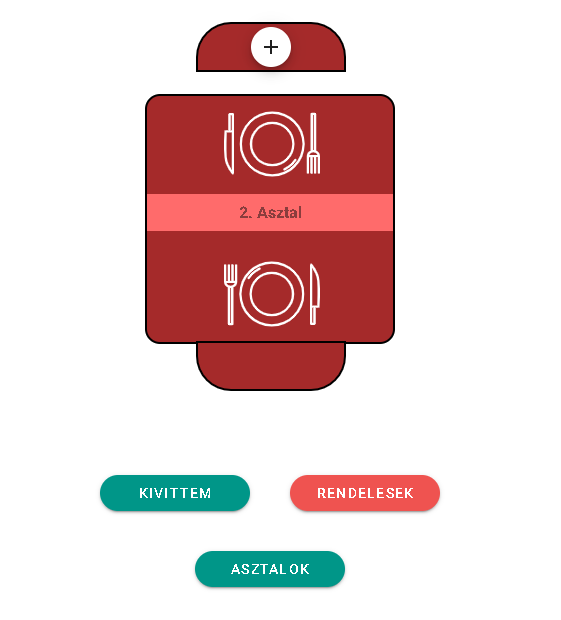
**Étlap & Itallap menü**

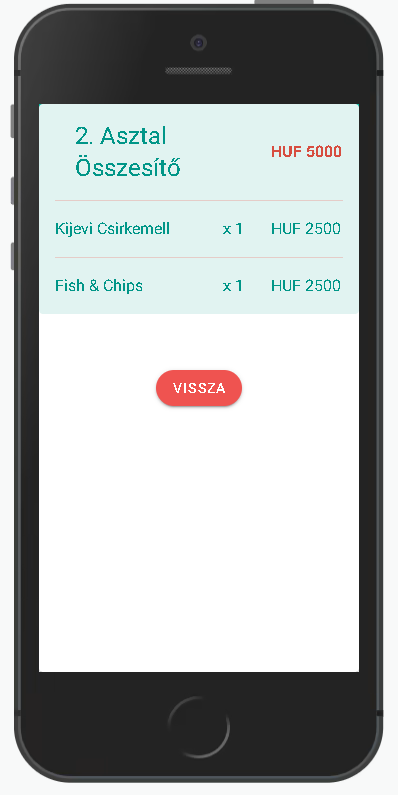
Az étlap & itallap menü megnyitása után lehetőségünk nyílik rendelések leadására.

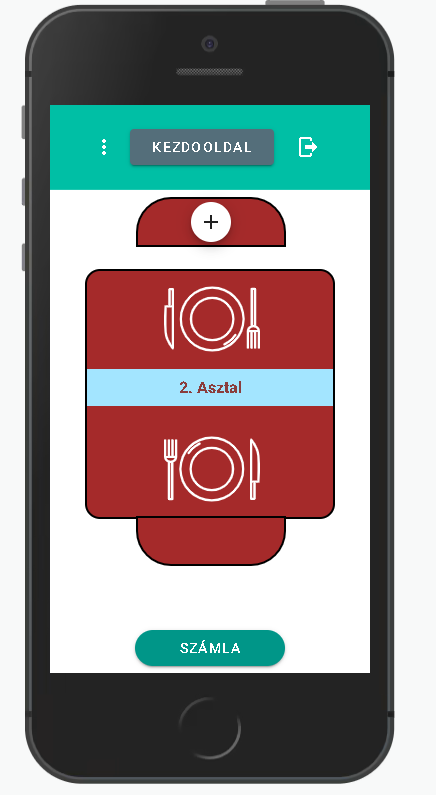
Amint egy ételt a kosárba helyezünk a  gomb segítségével, frissül az összesítő komponens, így folyamatos visszajelzést kapunk a végleges rendelés összegéről. A  gomb a kosárban lévő étel számát csökkenti. Az étel számot a két gomb között láthatjuk, valamint az összesítőben.

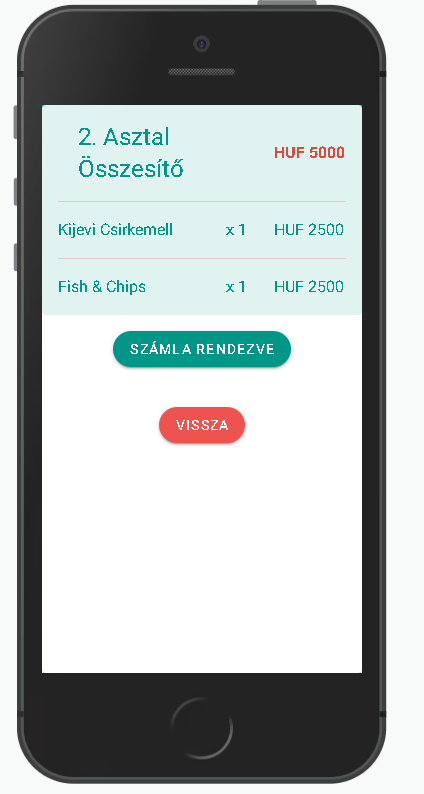
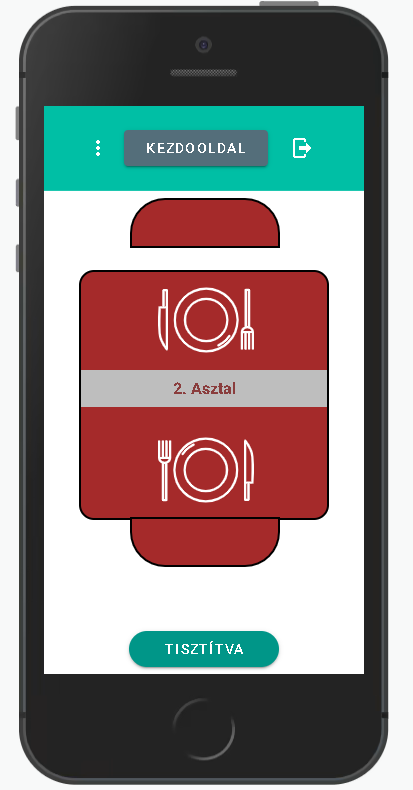


A rendelés leadás gomb csak akkor elérhető, ha legalább egy termék van a kosárban. A gomb használata után újra az asztal nézetbe kerülünk, ahol a terítő színe vörösre változik, ezzel jelezve, hogy az asztal vár a rendelésére.

**Rendelések**

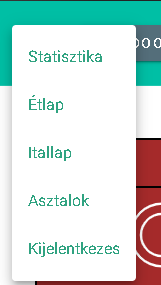
Megjelenik ezzel egyidejűleg a Rendelések gomb, amellyel az asztal összes leadott rendelését tekinthetjük meg, valamint a Kivittem gomb.

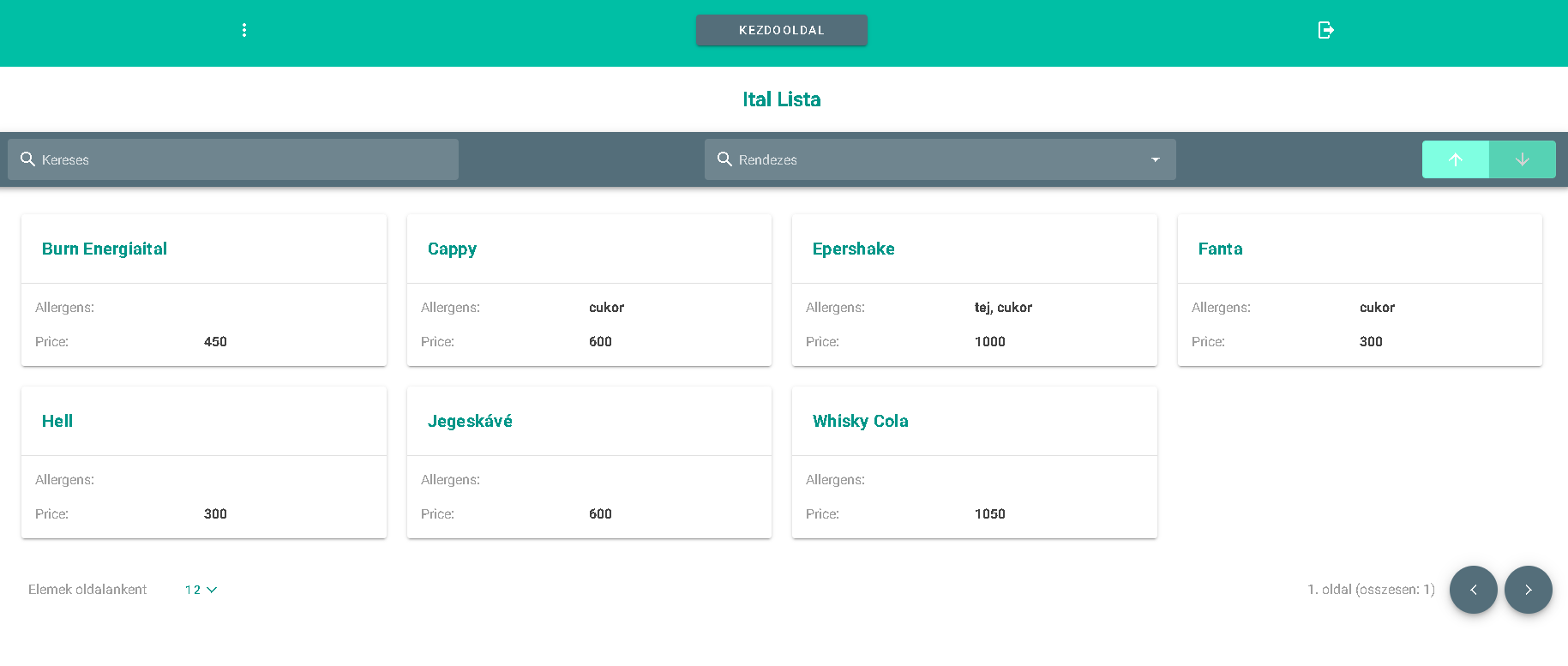
A Kivittem gomb megváltoztatja az asztal státuszát, a terítő színkódja kékre változik, megjeleníti a Számla gombot, amely lehetőség et biztosít a számla kiegyenlítésére.   
**Számla**

A számla gomb a számla nézetet fogja megnyitni. Itt véglegesíthetjük a számla kifizetését, vagy visszamehetünk az asztal nézetbe.  
  
  
  
  
A számla kifizetésének véglegesítése után a terítő színe szürke lesz.   
   
A tisztítva gomb lenyomásakor az asztal és székei visszakerülnek eredeti állapotukba, a terítő zöld lesz, az asztal újra készen áll vendég fogadására.

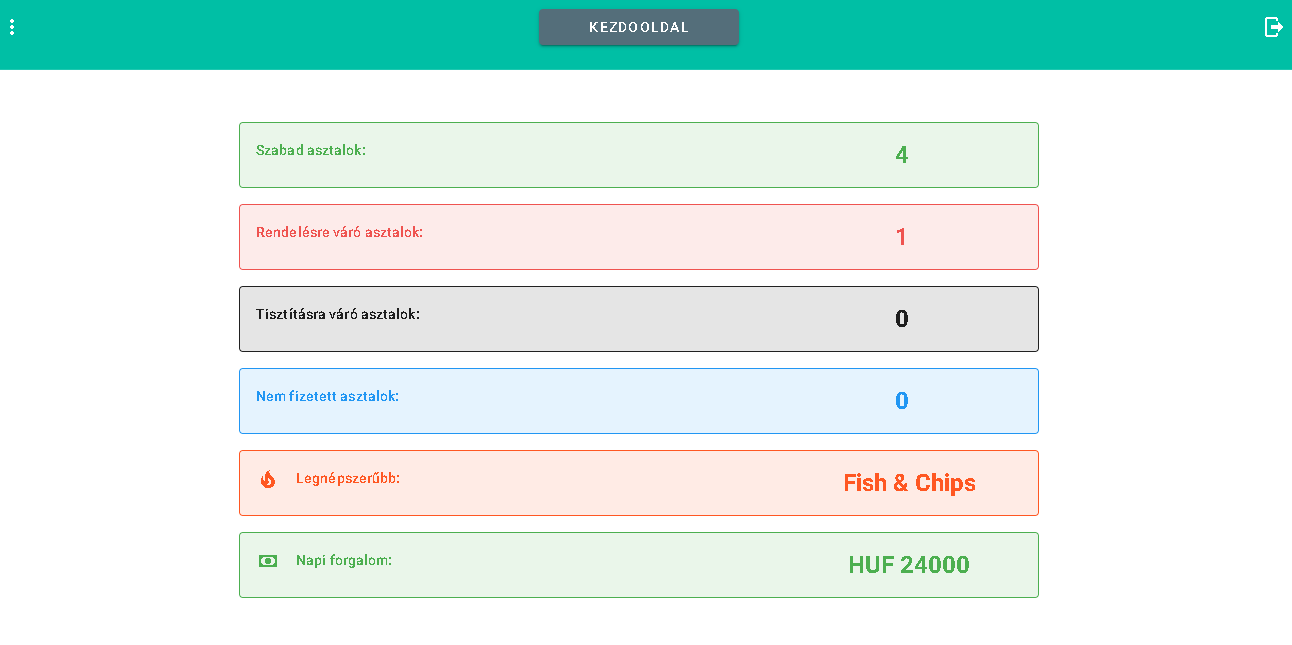
A székek bizonyos helyzetekben zárolva lesznek, mozgásra képtelenek. Például akkor, ha van egy rendelés leadva az asztalnál és csak egy szék van nyitott állapotban. Ilyenkor nem tudjuk betolni a széket. Hasonlóképpen nem tudjuk nem tiszta asztaloknál bárhogyan mozgatni a székeket.

A felhasználói nézeten is elérhető a főmenü, amelyen az étlap & itallap listát csak olvasási módban meg tudjuk nyitni, valamint felhasználónak szánt statisztikai oldalt is elérhetünk.

**Felhasználói Itallap & Étlap lista**  
  
  
  


A csak olvasási módot a szerkesztési gombok eltüntetésével oldottuk meg.

**Felhasználói Statisztika nézet**



A statisztikai oldalon folyamatos visszajelzést kapunk a szabad, rendelésre váró, tisztításra váró, és nem fizetett asztalok számáról, valamint a legnépszerűbb ételről és napi forgalomról.

A bejelentkezési felületeken kívül valamennyi oldal alapvető védelemmel van ellátva, így csökkentve a bejelentkezés nélküli tevékenységeket:



**Back-end**

**Adatbázis szerkezet:**

TÁBLA: asztalok

MEZŐK:

id [INT, AUTO INCREMENT, UNIQUE]

alsoSzekFoglalt [BOOL]

alsoSzekZarolva [BOOL]

felsoSzekFoglalt [BOOL]

felsoSzekZarolva [BOOL]

rendelesAllapot [STRING, VARCHAR(50)]

aktivRendeles [INT]

EXTRA INFORMÁCIÓ:

A tábla célja az asztalok és székek folyamatos állapot követése. Boolean értékekkel tájékozódunk arról, hogy egy-egy szék jelenleg foglalt-e, vagy zárolva van-e. Az asztalok rendelkeznek rendelés állapottal. A mező értéke egy kell, hogy legyen az alábbi 4 közül: **idle, ordered, delivered, paid**Az aktivRendeles mezőben tartjuk nyilván, hogy mi a jelenlegi rendelési azonosító ami az asztalhoz tartozik. Egy asztal szabad ’idle’ állapotban 0-as aktivRendeles mezővel rendelkezik, máskülönben egy a rendeles táblában létező id-val

TÁBLA: rendeles

MEZŐK:

id [INT, AUTO INCREMENT, UNIQUE]

asztalAzonosito [INT FOREIGN KEY asztalok.id]

letrehozva[DATE]

fizetve[BOOL]

visszavonva[BOOL]

EXTRA INFORMÁCIÓ:

A rendelések rögzítése statisztikai és állapotkövetési célokból szükséges. Egy-egy rendelés egyedi azonosítóval rendelkezik, amihez tartozhat egy vagy több étel rendelés. A hozzátartozó étel rendelést a rendelesreszlet táblából olvashatjuk ki.

TÁBLA: rendelesreszlet

MEZŐK:

id [INT, AUTO INCREMENT, UNIQUE]

rendelesAzonosito [INT FOREIGN KEY rendeles.id]

asztalAzonosito [INT FOREIGN KEY asztalok.id]

etelAzonosito [INT FOREIGN KEY etlap.id]

etelDarabszam [INT]

EXTRA INFORMÁCIÓ:

Minden megrendelt étel egy rekordként jelenik meg a táblán. A rekordban rögzítjük a darabszámot.

TÁBLA: etlap

MEZŐK:

id [INT, AUTO INCREMENT, UNIQUE]

nev [STRING, VARCHAR(255)]

kategoria [STRING, VARCHAR(50)]

allergenek [STRING, VARCHAR(255)]

ar [INT]

EXTRA INFORMÁCIÓ:

Az étlap táblán keresztül tesszük lehetővé az adminisztrátornak az étlap kezelést, valamint innen tápláljuk be az adatokat a dineR applikáció adminisztrátori és felhasználói nézeteibe.

TÁBLA: felhasznalok

MEZŐK:

id [INT, AUTO INCREMENT, UNIQUE]

felhasznalonev [STRING, UNIQUE, VARCHAR(255)]

jelszo [STRING, VARCHAR(20)]

vezeteknev [STRING, VARCHAR(255)]

keresztnev [STRING, VARCHAR(255)]

admin [BOOL]

EXTRA INFORMÁCIÓ:

A felhasználók táblában hozzuk létre az adminisztrátorokat és felhasználókat, valamint ezen tábla segítségével lehetséges bejelentkezni az applikációba. A felhasznalonev egyedi, nem szereplhet kétszer ugyanaz a felhasznalonev a táblában.

**API szerkezet**

**BASE URL: http://localhost:3000**

Metódus: GET

Végpont: /food/list

Eredmény: Étlap lista

Eredmény szerkezet:

[

{

"id": 91,

"nev": "Somloi Galuska",

"kategoria": "Desszert",

"allergenek": "",

"ar": 750

},

{

"id": 92,

"nev": "Almás pite",

"kategoria": "Desszert",

"allergenek": "",

"ar": 1000

}

]

Metódus: POST

Végpont: /food/create

Fogadott Body paraméterek:

- name (VARCHAR(255))

- category ("Ital" vagy "Eloetel" vagy "Leves" vagy "Foetel" vagy "Desszert")

- allergens (VARCHAR(255))

- price (INT)

Eredmény: Étel / Ital objektum létrehozása az etlap táblában

Eredmény szerkezet:

[

{

"id": 93,

"nev": "Sajttorta",

"kategoria": "Desszert",

"allergenek": "Tej",

"ar": 850

}

]

Metódus: PUT

Végpont: /food/edit/:id

Fogadott URL paraméterek: id (Meglévő étel egyedi azonosítója)

Fogadott Body paraméterek:

- name (VARCHAR(255))

- category ("Ital" vagy "Eloetel" vagy "Leves" vagy "Foetel" vagy "Desszert")

- allergens (VARCHAR(255))

- price (INT)

Eredmény: Létező étel / ital objektum megváltoztatása az etlap táblában

Eredmény szerkezet:

[

{

"id": 93,

"nev": "Sajttorta",

"kategoria": "Desszert",

"allergenek": "Tej",

"ar": 900

}

]

Metódus: DELETE

Végpont: /food/remove/:id

Fogadott URL paraméterek: id (Meglévő étel egyedi azonosítója)

Eredmény: Létező étel / ital objektum törlése az etlap táblából

Eredmény szerkezet:

{

"success": "true"

}

Metódus: GET

Végpont: /users/list

Eredmény: Adminisztrátori / Felhasználói lista

Eredmény szerkezet:

[

{

"id": 37,

"felhasznalonev": "admin",

"jelszo": "admin",

"vezeteknev": "admin",

"keresztnev": "admin",

"admin": 0

},

{

"id": 71,

"felhasznalonev": "szokes",

"jelszo": "szokes",

"vezeteknev": "Szoke",

"keresztnev": "Sandor",

"admin": 1

}

]

Metódus: POST

Végpont: /users/create

Fogadott Body paraméterek:

- felhasznalonev (VARCHAR(255))

- jelszo (VARCHAR(20))

- vezeteknev (VARCHAR(255))

- keresztnev (VARCHAR(255))

- admin (0 vagy 1 , ahol a 0 az adminisztrátor, az 1 a felhasználó)

Eredmény: Adminisztrátor / Felhasználó objektum létrehozása a felhasznalok táblában

Eredmény szerkezet:

[

{

"id": 85,

"felhasznalonev": "TestUser",

"jelszo": "TestUser",

"vezeteknev": "TEST",

"keresztnev": "USER",

"admin": 1

}

]

Metódus: PUT

Végpont: /users/edit/:id

Fogadott URL paraméterek: id (Meglévő adminisztrátor / felhasználó egyedi azonosítója)

Fogadott Body paraméterek:

- felhasznalonev (VARCHAR(255))

- jelszo (VARCHAR(20))

- vezeteknev (VARCHAR(255))

- keresztnev (VARCHAR(255))

- admin (0 vagy 1 , ahol a 0 az adminisztrátor, az 1 a felhasználó)

Eredmény: Létező Adminisztrátor / Felhasználó objektum megváltoztatása a felhasznalok táblában

Eredmény szerkezet:

[

{

"id": 85,

"felhasznalonev": "TestUser",

"jelszo": "TestUser",

"vezeteknev": "TEST",

"keresztnev": "USER",

"admin": 1

}

]

Metódus: DELETE

Végpont: /users/remove/:id

Fogadott URL paraméterek: id (Meglévő adminisztrátor / felhasználó egyedi azonosítója)

Eredmény: Adminisztrátor / Felhasználó objektum létrehozása a felhasznalok táblában

Eredmény szerkezet:

{

"success": "true"

}

Metódus: POST

Végpont: /login

Fogadott Body paraméterek:

- felhasznalonev (VARCHAR(255))

- jelszo (VARCHAR(20))

- jogosultsag (0 vagy 1 , ahol a 0 az adminisztrátor, az 1 a felhasználó)

Eredmény: Adminisztrátor / Felhasználó autentikálása

Eredmény szerkezet:

{

"success": "true"

}

Metódus: GET

Végpont: /tables/list

Eredmény: Asztalok és hozzátartozó székek jelenlegi állapota

Eredmény szerkezet:

[

{

"id": 1,

"alsoSzekFoglalt": 0,

"alsoSzekZarolva": 0,

"felsoSzekFoglalt": 0,

"felsoSzekZarolva": 0,

"rendelesAllapot": "idle",

"aktivRendeles": 0

},

{

"id": 2,

"alsoSzekFoglalt": 0,

"alsoSzekZarolva": 0,

"felsoSzekFoglalt": 0,

"felsoSzekZarolva": 0,

"rendelesAllapot": "idle",

"aktivRendeles": 0

},

{

"id": 3,

"alsoSzekFoglalt": 0,

"alsoSzekZarolva": 0,

"felsoSzekFoglalt": 0,

"felsoSzekZarolva": 0,

"rendelesAllapot": "idle",

"aktivRendeles": 0

},

{

"id": 4,

"alsoSzekFoglalt": 0,

"alsoSzekZarolva": 0,

"felsoSzekFoglalt": 0,

"felsoSzekZarolva": 0,

"rendelesAllapot": "idle",

"aktivRendeles": 0

},

{

"id": 5,

"alsoSzekFoglalt": 0,

"alsoSzekZarolva": 0,

"felsoSzekFoglalt": 0,

"felsoSzekZarolva": 0,

"rendelesAllapot": "idle",

"aktivRendeles": 0

}

]

Metódus: PUT

Végpont: /tables/:id/:state

Fogadott URL paraméterek: id (az asztal egyedi azonosítója), state ("idle" vagy "ordered" vagy "delivered" vagy "paid", ahol "idle" a szabad, "ordered" a rendelt, "delivered" a kiszolgált, "paid" a fizetett)

Eredmény:

- Az id-hoz tartozó asztal rendelesAllapotanak megváltozása az asztalok táblában, a state paraméternek megfelelően.

- "paid" state paraméter esetén az aktiveRendeles lenullázása az asztalok táblában és fizetve mező megváltoztatása a rendeles táblában.

Eredmény szerkezet:

{

"success": "true"

}

Metódus: PUT

Végpont: /tables/:id/chair/:pos/:state/:lock

Fogadott URL paraméterek: id (az asztal egyedi azonosítója), pos(0 vagy 1, ahol a 0 a felső szék, 1 az alsó szék), state (0 vagy 1, ahol 0 a szabad, 1 a foglalt), lock (0 vagy 1, ahol 0 a nem zárolt, 1 a zárolt)

Eredmény:

- Az id-hoz tartozó asztal, pos-hoz tartozó szék státuszának és zárolásának megváltozása

Eredmény szerkezet:

{

"fieldCount": 0,

"affectedRows": 1,

"insertId": 0,

"serverStatus": 2,

"warningCount": 0,

"message": "(Rows matched: 1 Changed: 1 Warnings: 0",

"protocol41": true,

"changedRows": 1

}

Metódus: POST

Végpont: /tables/:id/chair/:pos/state/:state/

Fogadott URL paraméterek: id (az asztal egyedi azonosítója), pos(0 vagy 1, ahol a 0 a felső szék, 1 az alsó szék), state (0 vagy 1, ahol 0 a szabad, 1 a foglalt)

Eredmény:

- Az id-hoz tartozó asztal, pos-hoz tartozó szék státuszának megváltozása

Eredmény szerkezet:

{

"fieldCount": 0,

"affectedRows": 1,

"insertId": 0,

"serverStatus": 2,

"warningCount": 0,

"message": "(Rows matched: 1 Changed: 1 Warnings: 0",

"protocol41": true,

"changedRows": 1

}

Metódus: POST

Végpont: /tables/:id/chair/:pos/lock/:state/

Fogadott URL paraméterek: id (az asztal egyedi azonosítója), pos(0 vagy 1, ahol a 0 a felső szék, 1 az alsó szék), state (0 vagy 1, ahol 0 a nem zárolt, 1 a zárolt)

Eredmény:

- Az id-hoz tartozó asztal, pos-hoz tartozó szék zárolásának megváltozása

Eredmény szerkezet:

{

"fieldCount": 0,

"affectedRows": 1,

"insertId": 0,

"serverStatus": 2,

"warningCount": 0,

"message": "(Rows matched: 1 Changed: 1 Warnings: 0",

"protocol41": true,

"changedRows": 1

}

Metódus: GET

Végpont: /tables/:id/orders

Fogadott URL paraméterek: id (az asztal egyedi azonosítója)

Eredmény:

- Az id-hoz tartozó asztal rendelési adatai

Eredmény szerkezet:

[

{

"etelDarabszam": 2,

"etelAzonosito": 11

},

{

"etelDarabszam": 2,

"etelAzonosito": 83

}

]

Metódus: POST

Végpont: /orders/create/:tableId

Fogadott URL paraméterek: tableId (az asztal egyedi azonosítója)

Fogadott Body paraméterek: [{"orderCount": 2,"orderId": 11},{"orderCount": 2,"orderId": 83}] (névtelen lista, amelyben névtelen objektumokok találhatóak, kulcs érték párokkal, minden névtelen objektumnak két kulcsot kell tartalmaznia, orderCount és orderId, ahol az orderCount a rendelés darabszáma, az orderId pedig az étel/ital egyedi azonosítója az etlap táblából)

Eredmény:

- Az id-hoz tartozó asztal rendelési adatai

Eredmény szerkezet:

[

{

"etelDarabszam": 2,

"etelAzonosito": 11

},

{

"etelDarabszam": 2,

"etelAzonosito": 83

}

]

Metódus: GET

Végpont: /stat/day

Eredmény:

- napi statisztika az összforgalomról és a legnépszerűbb ételről

Eredmény szerkezet:

{

"sum": 38600,

"star": {

"id": 90,

"price": 1500,

"name": "Halaszle",

"category": "Leves"

}

}

Metódus: GET

Végpont: /stat/month

Eredmény:

- havi statisztika, összforgalom, napi forgalom, legnépszerűbb termék, étel, ital

Eredmény szerkezet:

{

"sumPerDay": [

{

"letrehozva": "2022-04-01",

"sum": 0

},

{

"letrehozva": "2022-04-02",

"sum": 0

},

{

"letrehozva": "2022-04-03",

"sum": 0

},

{

"letrehozva": "2022-04-04",

"sum": 0

},

{

"letrehozva": "2022-04-05",

"sum": 0

},

{

"letrehozva": "2022-04-06",

"sum": 0

},

{

"letrehozva": "2022-04-07",

"sum": 0

},

{

"letrehozva": "2022-04-08",

"sum": 0

},

{

"letrehozva": "2022-04-09",

"sum": 0

},

{

"letrehozva": "2022-04-10",

"sum": 0

},

{

"letrehozva": "2022-04-11",

"sum": 0

},

{

"letrehozva": "2022-04-12",

"sum": 0

},

{

"letrehozva": "2022-04-13",

"sum": 0

},

{

"letrehozva": "2022-04-14",

"sum": 0

},

{

"letrehozva": "2022-04-15",

"sum": 0

},

{

"letrehozva": "2022-04-16",

"sum": 0

},

{

"letrehozva": "2022-04-17",

"sum": 0

},

{

"letrehozva": "2022-04-18",

"sum": 0

},

{

"letrehozva": "2022-04-19",

"sum": 0

},

{

"letrehozva": "2022-04-20",

"sum": 0

},

{

"letrehozva": "2022-04-21",

"sum": 4000

},

{

"letrehozva": "2022-04-22",

"sum": 15600

},

{

"letrehozva": "2022-04-23",

"sum": 53050

},

{

"letrehozva": "2022-04-24",

"sum": 38600

},

{

"letrehozva": "2022-04-25",

"sum": 0

},

{

"letrehozva": "2022-04-26",

"sum": 0

},

{

"letrehozva": "2022-04-27",

"sum": 0

},

{

"letrehozva": "2022-04-28",

"sum": 0

},

{

"letrehozva": "2022-04-29",

"sum": 0

},

{

"letrehozva": "2022-04-30",

"sum": 0

}

],

"sum": 111250,

"star": {

"id": 64,

"price": 300,

"name": "Hell",

"category": "Ital"

},

"starFood": {

"id": 83,

"price": 2500,

"name": "Kijevi Csirkemell",

"category": "Foetel"

},

"starDrink": {

"id": 64,

"price": 300,

"name": "Hell",

"category": "Ital"

}

}

**Felhasználási útmutató:**

A dineR backendje NodeJS és ExpressJS valamint MySQL segítségével tud működni. A NodeJS telepítése előzetesen meg kell, hogy történjen a számítógépen, valamint szükséges egy futó MySQL szerver. A MySQL szerveren importálni kell a csatolt adatbazis.sql file-t ami létrehozza az adatbázist, vagy az adatbazis.sql fájl SQL utasításait végre kell hajtani, hogy legyen egy kész adatbázis, mielőtt az a backend API-t elindítjük. Minden más backend dependenciát a package.json file tartalmaz. Az API-ban rögzítve van a localhost 3000-es portja, valamint az ’etterem’ nevű adatbázis, root nevű felhasználó jelszó nélküli hozzáférése. Ezeket igény esetén felül lehet írni, arra viszont ügyelni kell, hogy akkor a MySQL szerveren hasonló néven jöjjön létre az adatbázis és létezzen a felhasználó.

Frontenden a dineR applikáció a VueJS keretrendszerre támaszkodik, melynek további dependenciái, mint például a Vuetify, Vuex és Vue Router a frontendhez tartozó package.json file fog ismertetni.

Ha minden készen áll, az API-t el kell indítani a következő paranccsal: node API.js

Ezután az applikáció használatba vehető localhoston, a node által kiosztott porton, általában 8080.