

RÉKA

Rendelés, Értékesítés, Készlet, Adminisztráció

Algoritmus

Készítette:
Csuka Tamás József
Paller Péter Décse
2025.11.07.

Tartalomjegyzék

1. Bejelentkezés	3
1.1. Cél / Feladat megfogalmazása.....	3
1.2. Bemenet.....	3
1.3. Kimenet	3
1.4. Működés leírása (szövegesen)	3
1.5. Pszeudokód	3
1.6. API.....	4
1.7. Kép (minta)	5
2. Partner cégek kilistázása	6
2.1. Cél / Feladat megfogalmazása.....	6
2.2. Bemenet.....	6
2.3. Kimenet	6
2.4. Működés leírása (szövegesen)	6
2.5. Pszeudokód	6
2.6 API.....	7
2.7. Kép (minta)	7
3. Rendelés leadás.....	8
3.1. Cél / Feladat megfogalmazása.....	8
3.2. Bemenet.....	8
3.3. Kimenet	8
3.4. Működés leírása (szövegesen)	8
3.5. Pszeudokód	9
3.6. API.....	9
3.7. Kép (minta)	10
4. Rendelések kiírása	11
4.1. Cél / Feladat megfogalmazása.....	11
4.2. Bemenet.....	11
4.3. Kimenet	11
4.4. Működés leírása (szövegesen)	11
4.5. Pszeudokód	12
4.6. API.....	12
4.7. Kép (minta)	13
5. Tartalom megjelenítése	14

5.1. Cél / Feladat megfogalmazása.....	14
5.2. Bemenet.....	14
5.3. Kimenet	14
5.4. Működés leírása (szövegesen)	14
5.5. Pszeudokód.....	14
5.6. API.....	15
5.7. Kép (minta)	16
6. Fuvar levél készítés	17
6.1. Cél / Feladat	17
6.2. Bemenet.....	17
6.3. Kimenet	17
6.4. Működés leírása (szövegesen)	17
6.5.Pszeudokód.....	18
6.6. API.....	19
7. Keresés	20
7.1. Cél / Feladat megfogalmazása.....	20
7.2. Bemenet.....	20
7.3. Kimenet	20
7.4. Működés leírása (szövegesen)	20
7.5. Pszeudokód.....	20
7.7. Kép (minta)	21

1. Bejelentkezés

1.1. Cél / Feladat megfogalmazása

Az algoritmus célja, hogy a rendszerbe történő bejelentkezés során ellenőrizze a felhasználó által megadott felhasználónevet és jelszót, valamint érvényesítse a mezők kitöltését és hosszát. A bejelentkezés csak akkor történik meg, ha:

- mindkét mező ki van töltve,
- az adatok megengedett hosszúságúak,
- a felhasználónév szerepel az adatbázisban,
- a megadott jelszó megegyezik az adatbázisban tárolt jelszóval.

1.2. Bemenet

- fnev – felhasználónév (szöveg, max. 100 karakter)
- jelszo – jelszó (szöveg, max. 100 karakter)
- Felhasznalo – adatbázis tábla, amely tartalmazza a regisztrált felhasználókat

1.3. Kimenet

- Sikeres bejelentkezés üzenet, ha az adatok helyesek
- Hibaüzenetek, ha bármelyik feltétel nem teljesül:
- „Adja meg a felhasználónevet vagy a jelszót.”
- „A felhasználónév vagy a jelszó nem megengedett hosszúságú.”
- „Hibás felhasználónév!”
- „Hibás jelszó!”

1.4. Működés leírása (szövegesen)

- A rendszer bekéri a felhasználótól a felhasználónevet és a jelszót a megfelelő beviteli mezőkben.
- A program ellenőrzi, hogy a mezők nincsenek-e üresen, és hogy a megadott adatok hossza nem haladja-e meg a megengedett 100 karaktert.
- Az algoritmus lekérdezi az adatbázisból a megadott felhasználónévhez tartozó jelszót.
- Ha a felhasználónév nem található meg az adatbázisban, a rendszer hibaüzenetet ad.
- Ha a felhasználónév létezik, akkor összehasonlítja a megadott jelszót az adatbázisban szereplővel.
- Amennyiben egyeznek, megtörténik a bejelentkezés; különben hibaüzenet jelenik meg.

1.5. Pszeudokód

ALGORITMUS bejelentkezés

VÁLTOZÓK:

String: fnev := ""

```
String: jelszo := ""
Adatbázis tábla: Felhasznalo
Adat: talalat
Bool: userOK := false
ciklus amíg (IGAZ)
    ciklus, amíg NEM(fnev <> ÜRES ÉS jelszo <> ÜRES ÉS
    fnev.hosszúság <= 100 és jelszo.hosszúság <= 100)
        BE: fnev [normál szövegdoz]
        BE: jelszo [jelszó szövegdoz]
        HA fnev = ÜRES VAGY jelszo = ÜRES:
            KIÍR "Adja meg a felhasználónevet vagy a jelszót."
        HA fnev.hosszúság > 100 VAGY jelszo.hosszúság > 100:
            KIÍR: "A felhasználónév vagy a jelszó nem megengedett
            hosszúságú"
    ciklus vége
    talalat := LEKERDEZ("SELECT jelszo FROM Felhasznalo WHERE nev
    = fnev")
    HA talalat = ÜRES:
        KIÍR: "Hibás felhasználónév!"
        CONTINUE
    HA jelszo = talalat
        userOK := true
        BREAK
    KÜLÖNBEN
        KIÍR: "Hibás jelszó!"
ciklus vége
ALGORITMUS vége
```

1.6. API

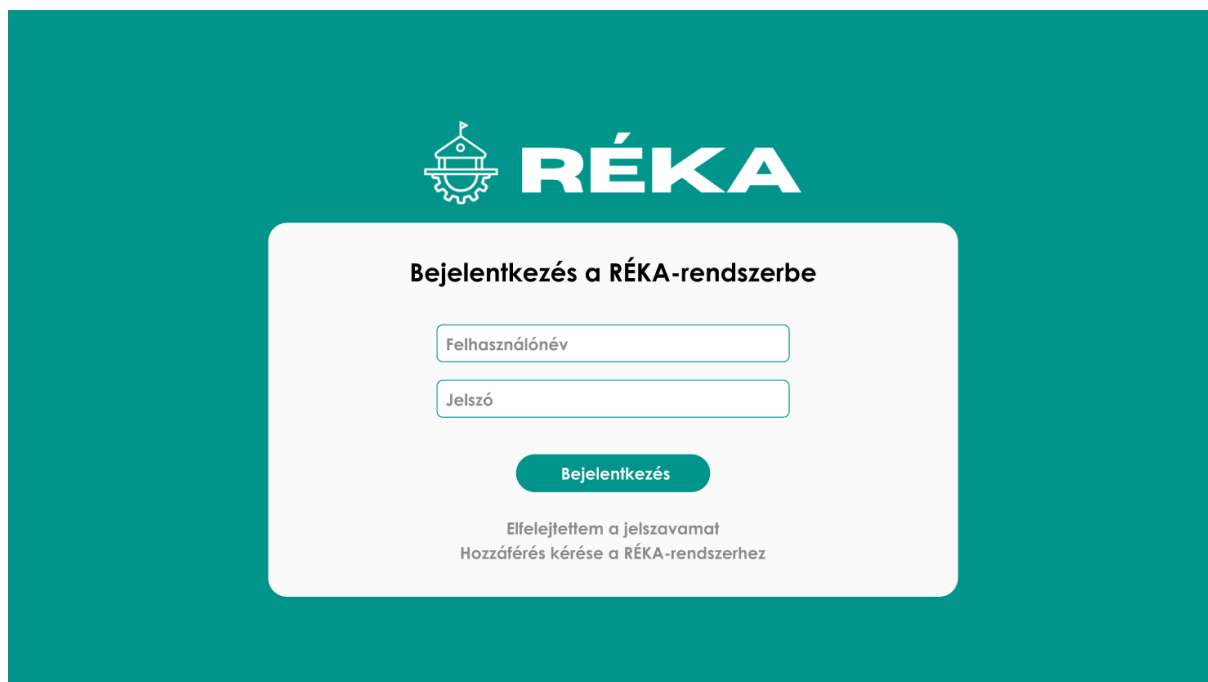
- Cél: Felhasználó azonosítása felhasználónév és jelszó alapján.
- HTTP módszer: POST
- Bemenet (JSON):

```
{
  "fnev": "string",
  "jelszo": "string"
}
```
- Leírás: Ellenőrzi, hogy a felhasználó létezik-e, és hogy a jelszó helyes-e.
- Sikeres válasz:

```
{
  "status": "ok",
  "message": "Sikeres bejelentkezés"
}
```
- Hibás válasz:

```
{
  "status": "error",
  "message": "Hibás felhasználónév vagy jelszó"
}
```

1.7. Kép (minta)



The image shows a login interface for the RÉKA system. At the top, the RÉKA logo is displayed. Below it, the title "Bejelentkezés a RÉKA-rendszerbe" is centered. There are two input fields: "Felhasználónév" (Username) and "Jelszó" (Password). A green "Bejelentkezés" (Login) button is positioned below the fields. At the bottom, there are two links: "Elfelejttem a jelszavamat" (I forgot my password) and "Hozzáférés kérése a RÉKA-rendszerhez" (Request access to the RÉKA system).

RÉKA

Bejelentkezés a RÉKA-rendszerbe

Felhasználónév

Jelszó

Bejelentkezés

[Elfelejttem a jelszavamat](#)
[Hozzáférés kérése a RÉKA-rendszerhez](#)

2. Partner cégek kilistázása

2.1. Cél / Feladat megfogalmazása

Az algoritmus célja, hogy kilistázza egy adott cég partnereit az adatbázisból. A felhasználó (aki egy cég alkalmazottja) azonosítójából kiindulva meghatározza a felhasználó cégének azonosítóját, majd lekérdezi az összes olyan céget, amellyel ez a cég partnerkapcsolatban áll — akár vevőként, akár eladóként. A kimenet egy lista a partner cégekről, alapvető adataikkal (név, adószám, e-mail stb.).

2.2. Bemenet

sajat_id – a bejelentkezett felhasználó azonosítója (egész szám)

2.3. Kimenet

- A partner cégek listája, amely tartalmazza a partnerek nevét, adószámát, elérhetőségeit.
- Képernyőn vagy táblázatos formában jeleníthető meg.

2.4. Működés leírása (szövegesen)

- Az algoritmus a bejelentkezett felhasználó (sajat_id) alapján lekéri annak a cégnek az azonosítóját, amelyhez az illető tartozik.
- Ezután az adatbázis Partnerseg táblájából lekérdezi az összes olyan rekordot, ahol a cég eladóként vagy vevőként szerepel.
- A lekérdezésekből az algoritmus összegyűjti a partner cégek adatait a Ceg táblából.
- Végül megjeleníti a partnerek listáját (név, adószám, e-mail stb.).

2.5. Pszeudokód

```
Algoritmus: listaz_partner_cegek
  Változók:
    Integer: saját_id
    Integer: saját_ceg_id
    Adat lista: partner_ceg
  saját_ceg_id = SELECT id FROM Ceg WHERE CONTAINS (alkalmazott,
'saját_id')
  partnerseg_lista = SELECT elado FROM Partnerseg WHEREOR vevo =
saját_ceg_id
Algoritmus tartalom_megjelenitese
ALGORITMUS vége
```

2.6 API

- Cél: A bejelentkezett felhasználó cégének partnerlistáját adja vissza.
- HTTP metódus: GET
- Paraméter: saját_id (integer)
- Leírás: A felhasználóhoz tartozó céget és annak partnereit listázza.
- Kimenet:

```
[
  {
    "id": 2,
    "nev": "Minta Kft.",
    "adoszam": "12345678-1-12",
    "email": info@mintaceg.hu
  }
]
```

2.7. Kép (minta)



RÉKA Tóth Réka

Összecsuk Kezdőlap Statisztika Adatok Áruház Raktárkezelés Beállítások Cégem Beérkezett rend. **Partnerségek** Lenti tesztem

Partnerségek kezelése

+ Új eladói partnerség felvétele

Eladói partnerségek

Partner neve	Fizetési mód	Fizetési idő	
Fundaluka Kft.	Banki átutalás	3 nap	 
Rácz István e.v	Készpénz	5 nap	 

Vevői partnerségek

Partner neve	
Drinks Kft.	
Snacky Bt.	

3. Rendelés leadás

3.1. Cél / Feladat megfogalmazása

Az algoritmus célja, hogy a rendszerben egy vevőcég egy partner (eladó) cég termékeiből rendelést tudjon leadni. Az algoritmus lekéri a partner termékeit, megjeleníti azokat a felhasználónak, majd a kiválasztott tételeket rögzíti a Rendeles és RendelesTétel táblákban.

3.2. Bemenet

- partner_ceg – az eladó cégek listája az adatbázisból
- termék_lista – a kiválasztott partner céghez tartozó termékek listája
- saját_ceg – a bejelentkezett (vevő) cég adatai
- N – a partner_ceg lista indexe, amely meghatározza, melyik partnerhez készül a rendelés
- rendelt_termek[3×M] – egy 3 oszlopos mátrix, ahol az oszlopok:
 - o szállítási cím
 - o termék azonosító
 - o rendelt mennyiség

3.3. Kimenet

- Új bejegyzés a Rendeles táblában
- Több új bejegyzés a RendelesTétel táblában
- A rendelés státusza: „beérkezett”

3.4. Működés leírása (szövegesen)

- Az algoritmus lekéri a kiválasztott partner cég által kínált termékeket a Termek táblából.
- Megjeleníti ezeket a termékeket a felhasználó számára, hogy kiválaszthassa, mit szeretne megrendelni.
- A rendszer új rendelést hoz létre a Rendeles táblában, amely tartalmazza:
 - o a partnerkapcsolat azonosítóját,
 - o a rendelés dátumát (mai dátum),
 - o a rendelés státuszát („beérkezett”),
 - o és a vevő szállítási címét.
- Ezután a felhasználó által kiválasztott tételeket beilleszti a RendelesTétel táblába, összekapcsolva az új rendelés azonosítójával.
- A folyamat végén a rendelés minden tétele rögzítésre kerül az adatbázisban.

3.5. Pszeudokód

Algoritmus: rendeles_leadasa

VÁLTOZÓK:

Adat lista: partner_ceg

Adat lista: termék_lista

Adat lista: saját_ceg

Integer: N

String mátrix [2xM] nagyságú: rendelelt_termek [id,mennyiség]

FOLYAMAT:

```
termek_lisat = SELECT * FROM Termek WHERE tulajdonos =  
partner_ceg[N].id
```

Algoritmus: Tartalom megjelenítése

```
INSERT INTO Rendeles (partnerseg, datum, status, sz_cim)  
VALUES (SELECT id FROM Partnerseg WHERE vevo = saját_ceg.id  
AND elado = partner_ceg.id, MAI_DATUM, "beérkezett",  
sajat.cim)
```

ciklus tetel IN rendelt_termek

```
INSERT INTO Rendeles_tetel (rendeles_id, termék_id, mennyiség  
VALUES ((SELECT id FROM Rendeles WHERE partnerseg = (SELECT  
id FROM Partnerseg WHERE vevo = saját_ceg.id AND elado =  
partner_ceg.id)AND datum = maidatum), tetel[1], tetel[2])
```

ciklus vége

ALGORITMUS vége

3.6. API

- Cél: Új rendelés létrehozása egy partnercég termékeire.
- HTTP metódus: POST
- Bemenet:

```
{  
  "vevo_ceg_id": 1,  
  "elado_ceg_id": 2,  
  "rendelt_termek":  
    [  
      {  
        "termek_id": 5,  
        "mennyiség": 10  
      }  
    ]  
}
```

- Kimenet:

```
{  
  "status": "ok",  
  "message": "Rendelés sikeresen leadva",  
  "rendeles_id": 24  
}
```

3.7. Kép (minta)

 **RÉKA**

Kovács István

Kezdőlap


Áruház

Beállítások

Cégem

Leadott rendelések

Kosár

Termék neve	Partner	Mennyiség	Ár
Chio Chips Hagymás-tejfölös 60g (10db karton)	Snacky Bt.	— 8 +	21000 Ft 
Chio Chips Hagymás-tejfölös 60g (10db karton)	Snacky Bt.	— 8 +	21000 Ft 

Kosár 3

Kosár összértéke: 52000 Ft Partner: Snacky Bt.

Rendelés leadása

4. Rendelések kiírása

4.1. Cél / Feladat megfogalmazása

Az algoritmus célja, hogy egy adott céghez tartozó rendeléseket lekérdezze és megjelenítse az adatbázisból. A lekérdezés történhet szűrési feltételek alapján (pl. rendelő, állapot, beérkezés dátuma), vagy ezek hiányában az adott cég összes rendelését listázza.

4.2. Bemenet

- saját_cég – az aktuálisan bejelentkezett vállalat adatai (lista vagy rekord formában)
- rendelő – a megrendelő azonosítója vagy neve (szöveg típus)
- állapot – a rendelés állapota, pl. „függőben”, „teljesítve” (szöveg típus)
- be_erk – a rendelés beérkezésének dátuma (dátum típus)

4.3. Kimenet

- Az aktuális céghez tartozó rendelések listája az adatbázisból.
- A lekérdezés eredménye lehet:
 - o Szűrt lista, ha megadásra kerültek a feltételek (rendelő, állapot, dátum).
 - o Teljes lista, ha nincs szűrés.

4.4. Működés leírása (szövegesen)

- Az algoritmus megvizsgálja, hogy a felhasználó adott-e meg szűrési feltételeket (rendelő, állapot, beérkezési dátum).
- Ha igen, akkor a rendelések táblából csak azokat az adatokat kéri le, amelyek megfelelnek a feltételeknek:
 - o a rendelés partnerkapcsolata az aktuális céghez tartozik,
 - o a szállítási cím egyezik a rendelőével,
 - o a rendelés állapota és dátuma megfelel a megadott értékeknek.
- Ha nincs megadva szűrés, akkor az algoritmus lekéri az összes rendelést, amely az adott céghez (eladóhoz) kapcsolódik.
- Az eredményként kapott lista kiírásra kerül vagy megjelenik a felhasználói felületen.

4.5. Pszeudokód

Algoritmus rendelesek

VÁLTOZÓK:

Adat lista: saját_ceg

String: rendelo

String: allapot

Date: be_erk

FOLYAMAT:

```
HA rendelo <> ÜRES VAGY allapot <> ÜRES VAGY be_erk <> ÜRES:  
  SELECT * FROM rendelesek WHERE partnerseg = (SELECT id  
    FROM partnerseg WHERE elado = saját_ceg.id) AND sz_cim =  
    rendelo.id AND status = allapot AND datum = be_erk
```

KÜLÖNBEN

```
  SELECT * FROM rendelesek WHERE partnerseg = (SELECT id  
    FROM partnerseg WHERE elado = saját_ceg.id)
```

Algoritmus tartalom_megjelenites

VÉGEALGORITMUS

4.6. API

- Cél: Rendelések listázása állapot, rendelő vagy dátum szerint.
- HTTP módszer: GET
- Paraméterek: allapot, rendelo, datum (opcionális)
- Kimenet:

```
[  
  {  
    "id": 12,  
    "datum": "2025-10-30",  
    "status": "beérkezett",  
    "partner": "Minta Kft."  
  }  
]
```

4.7. Kép (minta)

 **RÉKA**

Összecsuk

Kezdőlap

Statistika

Adatok

Áruház

Raktárkezelés

Beállítások

Cégem

Beérkezett rend.

Leadott rendelések

Lenti tesztelem

Beérkezett rendelések

Rendelési azon.	Megrendelő	Rendelési dátum	
AA000006	Fundaluka Kft.	ma, 8:51	Megnyitás
AA000005	Apacuka Kft.	ma, 7:51	Megnyitás
AA000004	Rácz István e.v.	szept. 8., 19:01	Megnyitás
AA000003	OpikArt Kft.	szept. 8., 16:30	Megnyitás
AA000002	Jókai András e.v.	szept. 8., 12:27	Megnyitás
AA000001	LDP Bt.	szept. 8., 11:44	Megnyitás

5. Tartalom megjelenítése

5.1. Cél / Feladat megfogalmazása

Az algoritmus célja, hogy a RÉKA-rendszer oldalain betöltse és megjelenítse a megfelelő tartalmakat az adatbázisból egy API-hívás segítségével.

5.2. Bemenet

- Adott aloldal megnyitása

5.3. Kimenet

- Az adott aloldalhoz kapcsolódó adatok megjelenítése HTML DOM segítségével.
- Hiba esetén a képernyőre a "Nem sikerült lekérni az adatokat" üzenet kerül kiírása, ami a konzolon is megjelenik.

5.4. Működés leírása (szövegesen)

- Az algoritmus a RÉKA-rendszer legtöbb oldalán lefuttatásra kerül az oldal megnyitásakor. A betöltés közben egy „Betöltés...” szöveg jelenik meg a képernyőn (addig, amíg a loading bool igaz értékre van állítva), és létrejön az oldal nevével megegyező nevű üres string és egy üres 'error' string is.
- A töltés során az algoritmus egy GET kérést küld a megfelelő API-útvonalra.
- Sikeresen válasz esetén, az oldal nevével megegyező nevű üres string-be bekerül a válasz tartalma, majd az oldal formátumának megfelelően a string tartalma megjelenik a képernyőn HTML DOM segítségével. A loading bool értéke hamis lesz és a betöltés szövege pedig elrejtésre kerül.
- Hibás válasz esetén, az 'error' string-be kerül a hibaüzenet, majd a hibaüzenet szövege kiírásra kerül a képernyőn a HTML DOM segítségével. A loading bool értéke hamis lesz és a betöltés szövege pedig elrejtésre kerül.

5.5. Pszeudokód

ALGORITMUS `tartalomMegjelenites`

VÁLTOZÓK:

String: `rendelesek` := ""

Bool: `tolt` := HAMIS

String: `hibauz` := ""

FOLYAMAT:

`tolt` = IGAZ

`hibauz` = ""

PRÓBÁLD MEG:

KÜLDJ GET kérést az `"/api/orders"` címre

HA a válasz sikeres:

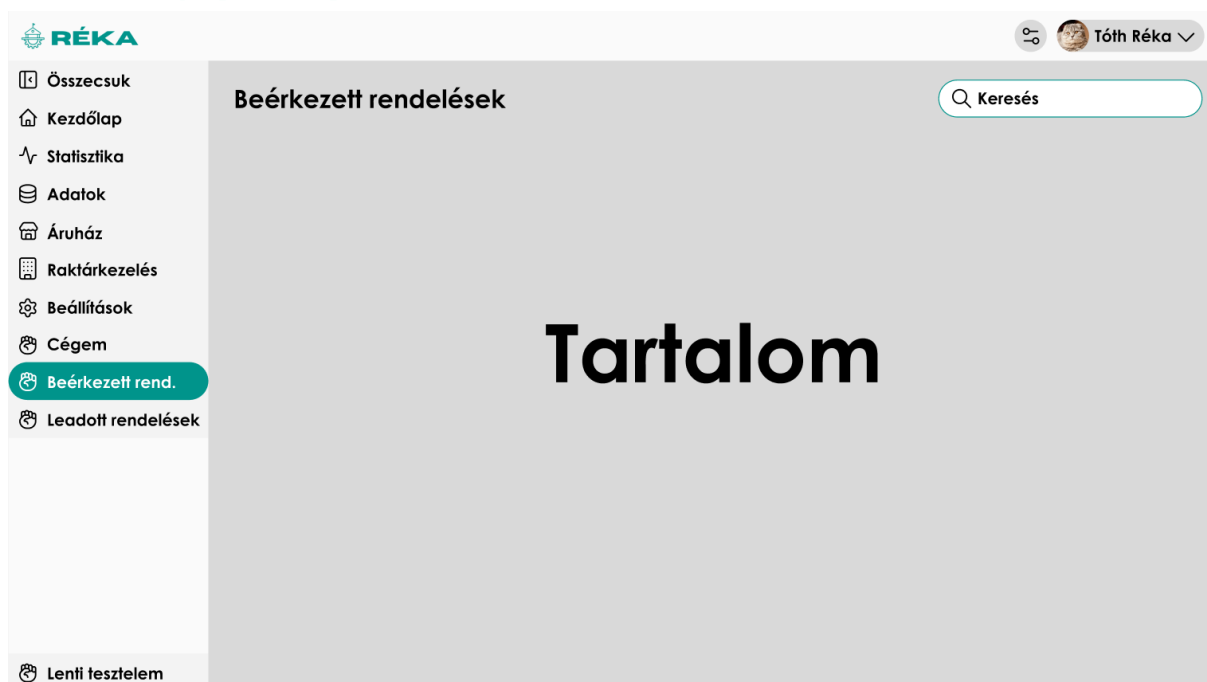
```
rendelesek = válaszbán kapott adatok (tömb
             objektumokkal)
HA hiba történt:
    console.log(hiba)
    hibauz = "Hiba történt: " + hiba
VÉGÜL:
    tolt = HAMIS
HA tolt = IGAZ
    KIÍR: "Betöltés..."
KÜLÖNBEN
    KIÍR: ""
HA hibauz != ""
    KIÍR: hibauz
KÜLÖNBEN:
    KIÍR: rendelesek (megfelelően módon)
```

5.6. API

- Cél: Rendelések megjelenítése a frontenden.
- HTTP metódus: GET
- Leírás: Betölti az /api/orders adatait és megjeleníti a DOM-ban.
- Kimenet:

```
[
  {
    "rendeles_id": 1,
    "datum": "2025-11-01",
    "status": "teljesítve"
  }
]
```


5.7. Kép (minta)



6. Fuvarlevél készítés

6.1. Cél / Feladat

Az algoritmus célja, hogy egy adott rendelés azonosító alapján automatikusan létrehozzon egy fuvarlevelet, amely tartalmazza:

- az eladó és vevő adatait,
- a rendeléshez tartozó termékek listáját,
- minden tétel nettó, áfa és bruttó értékét,
- többféle áfa kulcs esetén az áfa bontást,
- a rendelés összesített nettó és bruttó értékét.

A fuvarlevél kimenete lehet képernyőn megjelenített vagy fájlba mentett dokumentum.

6.2. Bemenet

rendeles_id – egész szám, a rendelés azonosítója.

6.3. Kimenet

Fuvarlevél, amely tartalmazza:

- Eladó és vevő adatait.
- Rendelés azonosítót és dátumot.
- Tételeket nettó, ÁFA és bruttó bontásban.
- ÁFA-kulcs szerinti összesítést.
- Összes nettó és bruttó végösszeget.

6.4. Működés leírása (szövegesen)

- A felhasználó megadja a rendelés azonosítót.
- Az algoritmus lekérdezi az adott rendeléshez tartozó adatokat a Rendeles, Partnerseg, Ceg és Termek táblákból.
- A rendelés tételeinek adatait a RendelesTetel és Termek táblákból olvassa be.
- Az összesítéshez minden tételre kiszámolja:
 - o a nettó értéket ($\text{ár} \times \text{mennyiség}$),
 - o a számított áfa értéket ($\text{nettó} \times \text{afa_kulcs} / 100$),
 - o a bruttó értéket ($\text{nettó} + \text{áfa}$).
- Az algoritmus összegzi a teljes rendelés nettó és bruttó értékeit, valamint áfa kulcsenként is összesíti az áfa összegeket.
- A kimenetként generált fuvarlevél tartalmazza az eladó és vevő adatait, a tételeket, az áfa bontást, valamint az összesített értékeket.

6.5.Pszudokód

ALGORITMUS: Fuvarlevel_Keszites

VÁLTOZÓK:

```
Integer: rendelésid, tetel_szam
Adat lista: rendelés_adat, partnerseg_adat, eladó_adat,
            vevo_adat
Adat: rendelési_hely
Lista: tetelek_lista
Adat lista: tetel
Adat mátrix: aradat [M,6] [tetel_nev, mennyiség, netto,
                        afa_kulcs, afa_ertek, brutto]
Lista [halmaz]: aktual_afa_kulcs
Integer mátrix: afa_fiz [M,2] [afakulcs, ertek]
Integer: ossz_netto:= 0, ossz_brutto:= 0, afa_ertek:= 0
```

FOLYAMAT:

```
BE: rendelés_id
rendelés_adat := LEKERDEZ("SELECT * FROM Rendeles WHERE id
=rendelés_id")
partnerseg_adat := LEKERDEZ("SELECT * FROM Partnerseg WHERE
id = rendelés_adat.partnerseg")
eladó_adat := LEKERDEZ("SELECT * FROM Ceg WHERE id =
partnerseg_adat.elado")
vevo_adat := LEKERDEZ("SELECT * FROM Ceg WHERE id =
partnerseg_adat.vevo")
rendelési_cím := LEKERDEZ("SELECT telephely_cím FROM
Felhasználó WHERE id = rendelés_adat.sz_cím")tetelek_lista
= LEKERDEZ(" SELECT Termek.nev, Termek.ar, Termek.afa_kulcs,
RendelesTetel.mennyiség FROM RendelesTetel Inner JOIN Termek
ON RendelesTetel.termek_id = Termek.id WHERE
RendelesTetel.rendelés_id = rendelés_adat.id")
KIÍR "-----"
KIÍR "          F U V A R L E V É L          "
KIÍR "-----"
KIÍR "Eladó: ", eladó_adat.nev
KIÍR "Adószám: ", eladó_adat.adoszam
KIÍR "Cím: ", eladó_adat.cím
KIÍR "-----"
KIÍR "Vevő: ", vevo_adat.nev
KIÍR "Adószám: ", vevo_adat.adoszam
KIÍR "Cím: ", vevo_adat.cím
KIÍR "-----"
KIÍR "Rendelés azonosító: ", rendelés_adat.id
KIÍR "Dátum: ", rendelés_adat.datum
KIÍR "Szállítási cím: ", rendelési_cím
KIÍR "-----"
KIÍR "Termék | Mennyiség | Nettó ár | ÁFA% | Bruttó ár"
CIKLUS amíg tetel IN tetelek_lista
```

```
KIÍR tetel.nev, " | ", tetel.mennyiseg, " | ", nettó, " |  
    ", tetel.afa_kulcs, "% | ", (tetel.ar * tetel.mennyiseg)  
    * (1 + (tetel.afa_kulcs/100))  
Integer: ii:= 1  
ossz_netto:= ossz_netto + tetel.mennyiseg * tetel.ar  
CIKLUS amíg afa_fiz[ii,1] <> aradat[i,4]  
    ii:= ii+1  
CIKLUS vége  
HA ii <= afa_fiz.oszlophossz:  
    afa_fiz[ii,2]:= afa_fiz[ii,2] + (tetel.ar *  
        tetel.mennyiseg) * (tetel.afa_kulcs/100)  
    ossz_bruttó:= ossz_brutto + (tetel.ar *  
        tetel.mennyiseg) * (1 + (tetel.afa_kulcs/100))  
KÜLÖNBEN  
    afa_fiz <= új sor  
    afa_fiz[ii+1,1]:= tetel.afa_kulcs  
    afa_fiz[ii+1,2]:= afa_fiz[ii+1,2] + (tetel.ar *  
        tetel.mennyiseg) * (tetel.afa_kulcs/100)  
    ossz_bruttó:= ossz_brutto + (tetel.ar *  
        tetel.mennyiseg) * (1 + (tetel.afa_kulcs/100))  
CIKLUS vége  
KIÍR "-----"  
KIÍR "Összes nettó: ", ossz_netto  
KIÍR (kurzor sorvégén marad) "Összes áfa":  
    CIKLUS amíg i := 1 i < afa_fiz.oszlophossz  
        KIÍR afa_fiz[i,1], "%: ", afa_fiz[i,2]  
    CIKLUS vége  
KIÍR "Összes bruttó: ", ossz_brutto  
ALGRITMUS vége
```

6.6. API

Cél: Rendeléshez tartozó fuvarlevél generálása nettó, ÁFA és bruttó árakkal.

HTTP metódus: GET

Paraméter: rendeles_id (integer)

Kimenet:

```
{  
    "rendeles_id": 24,  
    "datum": "2025-11-04",  
    "osszes_netto": 3500,  
    "osszes_brutto": 4310  
}
```

7. Keresés

7.1. Cél / Feladat megfogalmazása

- RÉKA-rendszer egyes oldalain keresés az oldalakon lévő táblázatokban, listáiban vagy div-eiben.

7.2. Bemenet

- Keresendő kifejezés
- Az oldalon található táblázat/lista/megfelelő div

7.3. Kimenet

- A keresőben megadott kifejezéssel egyező vagy hasonló elemek megjelenítése a táblázatban vagy listában, a nem egyező elemek pedig elrejtésre kerülnek.
- Ha nincs a keresésnek megfelelő elem, akkor a „Nincs találat” nevű szöveg jelenik meg az oldalon.

7.4. Működés leírása (szövegesen)

- A keresendő kifejezés, a táblázat/lista/azonos osztályú div elemek és a nincs találat elem megszerzése HTML DOM-mal.
- Ékezetek eltávolítása a keresendő kifejezésből függvénnel és kisbetűsítés.
- Táblázat/lista/div-ek összes sorának megszerzése, majd azok kisbetűsítése és ékezetmentesítése.
- A megadott részek összehasonlítása a keresett kifejezéssel, a nem egyező találatok elrejtése. Egyezés esetén a 'visibleCount' értéke eggyel növekszik.
- Amennyiben a visibleCount értéke 0, a táblázat/lista/divek teljes elrejtése és a 'Nincs találat' felirat megjelenítése. Ha az érték nem 0, a keresésnek megfelelő elemek megjelenítése a táblázatban/listában/div-ekben.

7.5. Pszeudokód

ALGORITMUS kereses (itt egy táblázatban)

VÁLTOZÓK:

```
String: keres := ""
String: tartalom := ""
String: filter := ""
String: nincsTalalat := ""
Integer: lathatoSzaml := 0
Bool: talalat := HAMIS
String: nevCella := ""
String: varosCella := ""
```

FOLYAMAT:

```
content = document.getElementById("tabla").innerHTML
tartalom = document.getElementById("keresoText").innerHTML
```

```
nincsTalalat = document.getElementById("nincsTalalat").
    innerHTML
FÜGGVÉNY ekezetEltavolit(str)
    VISSZAAD str.normalize('NFD').replace(/[\u0300
        \u036f]/g, '');
ESEMÉNY input
    filter = ekezetEltavolit(keres.toLowerCase())
    tartalom.osszesKivalaszt('tbody tr');
    lathatoSzaml = 0
    CIKLUS osszes sorig
        talalat = HAMIS
        nevCella = ekezetEltavolit(row.cells[0]
            .textContent.toLowerCase());
        varosCella = ekezetEltavolit(row.cells[2]
            .textContent.toLowerCase());
        HA filter ∈ nevCella VAGY filter ∈ varosCella
            talalat = IGAZ
        HA talalat == IGAZ
            visibleCount++
        KÜLÖNBEN
            row.classList.toggle('hidden');
    CIKLUS vége
    HA lathatoSzaml == 0:
        tartalom.style.display = 'none';
        nincsTalalat.style.display = 'block';
    KÜLÖNBEN:
        tartalom.style.display = 'table';
        nincsTalalat.style.display = 'none';
```

7.7. Kép (minta)

 **Keresés**