

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

# INFORMATIKAI ISMERETEK

## EMELT SZINTŰ GYAKORLATI VIZSGA

**minden vizsgázó számára**

**2023. október 20. 8:00**

Időtartam: 240 perc

Pótlapok száma	
Tisztázati	
Piszkozati	

Jelölje be az Ön által választott  
programozási nyelvet!

(Csak egy nyelvet jelölhet meg!)

Java ☐

C# ☐

**OKTATÁSI HIVATAL**

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Fontos tudnivalók

A vizsgán használható eszközök: a vizsgázó számára kijelölt számítógép, papír, toll, ceruza, lepecsételt jegyzetlap.

A feladatlap belső oldalain és a pótlapon készíthet jegyzeteket, ezeket a vizsga végén be kell adni, de tartalmukat nem fogják értékelni.

A feladatokat tetszőleges sorrendben oldhatja meg.

Javasoljuk, hogy a feladatokat először olvassa végig, utána egyenként oldja meg az egyes részfeladatokat!

A forrásfájlokat a vizsgakönyvtárban találja.

Felhívjuk a figyelmet a gyakori mentésre, és feltétlenül javasoljuk a mentést minden esetben, mielőtt egy másik feladat megoldásába kezd.

Vizsgadolgozatát a feladatlapon található azonosítóval megegyező nevű vizsgakönyvtárba kell mentenie. A vizsga végén ellenőrizze, hogy minden megoldás a megadott könyvtárban van-e, mert csak ezek értékelésére van lehetőség! Ellenőrizze, hogy a beadandó állományok olvashatók-e, mert a nem megnyitható állományok értékelése nem lehetséges!

A programozási feladatnál a program csak abban az esetben értékelhető, ha a vizsgázó létrehozta a választott programozási környezetnek megfelelő forrásállomány(oka)t a vizsgakönyvtárban, és az tartalmazza a részfeladatok megoldásához tartozó forráskódot.

Az adatbázis-fejlesztés feladatnál az egyes részfeladatok megoldását adó SQL kódokat kell elmentenie. A feladatban megadott állományba mentett SQL kódok kerülnek csak értékelésre.

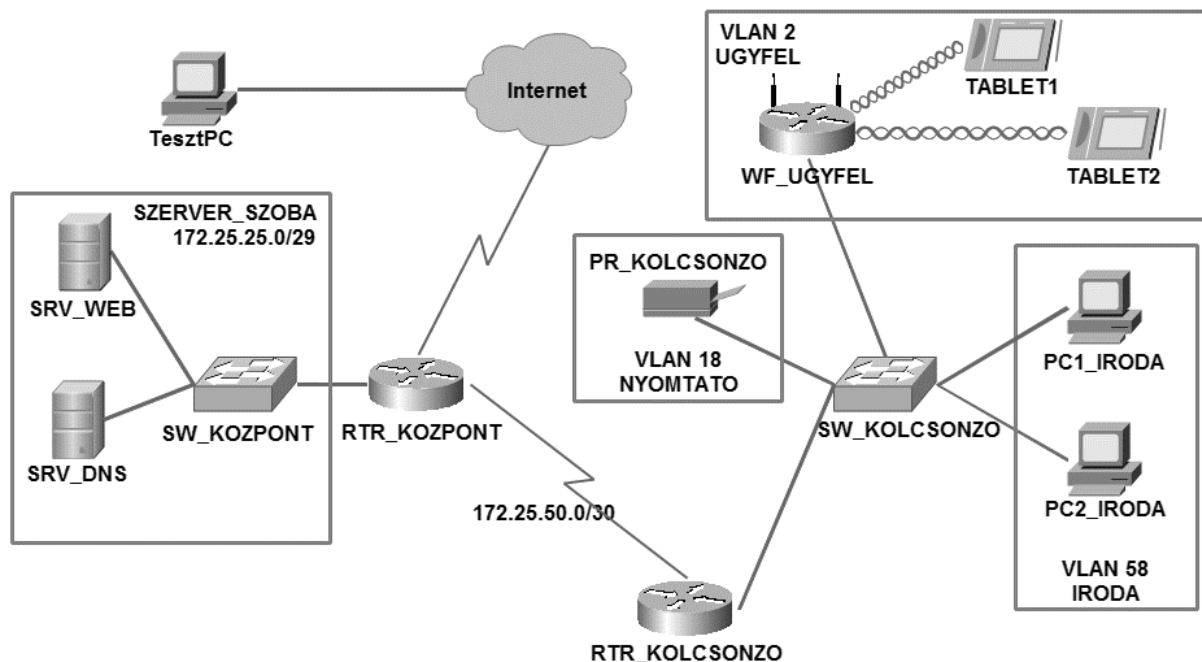
Amennyiben számítógépével műszaki probléma van, jelezze a felügyelő tanárnak! A jelzés ténye és a megállapított hiba jegyzőkönyvezésre kerül. A kiesett idővel a vizsga ideje hosszabb lesz. Amennyiben a hiba mégsem számítógépes eredetű, a javító tanár értékeléskor köteles figyelembe venni a jegyzőkönyv eseteírását. (A rendszergazda nem segítheti a vizsgázót a dolgozat elkészítésében.)

## 1. feladat

40 pont

### Autókölcsonzó

Egy újonnan nyíló autókölcsonzó szeretné a kiépítendő hálózat megfelelő működését letesztelni még mielőtt hozzáfognának a megvalósítási munkálatokhoz. Ehhez egy szimulációs szoftverben az eszközöket már elhelyezték a logikai topológián. Az Ön feladata, hogy az alábbi követelményeknek megfelelően konfigurálja a topológián található eszközöket, hogy az autókölcsonzó cég végre tudja hajtani az általuk tervezett tesztelést.



1. Töltse be az `autokolcsonzo.pkt` állományt a szimulációs programba! A teszhálózat már tartalmazza az autókölcsonzó összes hálózati eszközét és az internet szimulálására szolgáló eszközöket. Ez utóbbiak már beállításra kerültek. Az autókölcsonzó hálózatában található eszközök részleges konfigurációval már rendelkeznek, Önnek csak a feladatokban leírt módosításokat kell elvégeznie.
2. A kölcsönzésre szolgáló irodában külön alhálózatokat kell kialakítania az irodai alkalmazottak számítógépeinek (PC1\_IRODA, PC2\_IRODA), az irodában használt nyomtatónak (PR\_KOLCSONZO), és az ügyfelek számára használható vezeték nélküli hozzáféréshez (WF\_UGYFEL). Ezért az RTR\_KOLCSONZO forgalomirányító helyi hálózatában a **192.168.88.0/24** privát címtartományt három alhálózatra kell bontania a táblázatban szereplő igények figyelembevételével:

VLAN azonosító száma	VLAN neve	Igényelt IP-címek száma
58	IRODA	<b>58</b>
18	NYOMTATO	<b>3</b>
2	UGYFEL	<b>2</b>

Az `ipcimzes.txt` fájlban a példához hasonló módon rögzítse számolásának eredményét!

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Ha nem tudja elvégezni az alhálózatszámítást, akkor a továbbiakban a következő IP-címekkel dolgozzon:

VLAN azonosító száma	VLAN neve	Hálózat cím	Netmaszk
58	IRODA	<b>10.88.58.0</b>	<b>255.255.255.128</b>
18	NYOMTATO	<b>10.88.18.0</b>	<b>255.255.255.224</b>
2	UGYFEL	<b>10.88.2.0</b>	<b>255.255.255.240</b>

- Az RTR\_KOLCSONZO forgalomirányító helyi hálózathoz csatlakozó interfészén az előző feladatban meghatározott három alhálózat számára alakítson ki alinterfészeket! Az alinterfész száma minden esetben egyezzen meg a használt VLAN azonosító számával! Az alinterfészekre állítsa be a megfelelő hálózat utolsó kiosztható IP-címét!
- Az SW\_KOLCSONZO kapcsolón hozza létre a VLAN 58-at, a VLAN 18-at és a VLAN 2-t, majd a fenti táblázat alapján állítsa be a VLAN-oknak az IRODA, a NYOMTATO és az UGYFEL nevet!
- Az SW\_KOLCSONZO kapcsoló megfelelő portjainak konfigurálásával érje el, hogy a kliensekhez csatlakozó portok hozzáférési portok legyenek és a megfelelő VLAN-ba kerüljenek!
- Az SW\_KOLCSONZO kapcsolón, ahol szükséges, használjon trónk portot, hogy a hálózat működőképes legyen!
- A WF\_UGYFEL vezeték nélküli forgalomirányító Internet portja számára statikusan állítsa be a VLAN 2 hálózatának első kiosztható IP-címét, a megfelelő alapértelmezett átjárót és DNS kiszolgálót (SRV\_DNS szerver IP-címe)!
- A PR\_KOLCSONZO nyomtató számára statikusan állítsa be a VLAN 18 hálózatának első kiosztható IP-címét, a megfelelő alapértelmezett átjárót és DNS kiszolgálót (SRV\_DNS szerver IP-címe)!
- Az IRODA VLAN (VLAN 58) számára hozzon létre egy DHCP hatókört az RTR\_KOLCSONZO forgalomirányítón:
  - Biztosítsa az összes szükséges paraméter átadását!
  - A kliensek kapják meg a DNS kiszolgáló IP-címét is: SRV\_DNS szerver IP-címe!
  - A hálózat utolsó 5 darab címét ne oszthassa ki a DHCP kiszolgáló!
 Állítsa be az IRODA VLAN számítógépeit (PC1\_IRODA, PC2\_IRODA) a dinamikus IP-cím használatához!
- Az SW\_KOZPONT kapcsoló számára felügyeleti célból IP-címet kell beállítania. A kapcsolónak az adott hálózatban kiosztható első IP-címet állítsa be! Állítson be a kapcsolónak megfelelő alapértelmezett átjárót is!
- Az SW\_KOZPONT kapcsolón a privilegizált módot védő jelszó a **ABC789** legyen!
- Az SW\_KOZPONT kapcsoló első 5 virtuális vonalán állítsa be, hogy távolról csak SSH protokollal lehessen elérni az eszközt! Használjon helyi hitelesítést a vonalakon! A szükséges felhasználó neve **admin**, jelszava **ABCpassword** legyen! Állítsa be, hogy az eszköz domain neve **autoberles.hu** legyen! Engedélyezze az SSH 2-es verzióját! Használjon hozzá 1024 bites kulcsot!
- Az SW\_KOZPONT kapcsoló SRV\_WEB és SRV\_DNS szerverekhez csatlakozó portjain engedélyezzen portvédelmet úgy, hogy a jelenleg csatlakozó szerverek fizikai címét

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

statikusan vegye fel a konfigurációba! Ha más fizikai című eszköz használja a portot, akkor a port kapcsoljon le!

14. Az IPv4-es forgalom irányításához OSPF protokollt használnak a következők szerint:

- Az `RTR_KOZPONT` forgalomirányítón már korábban létrehozták és megfelelően beállították az OSPF folyamatot (szükséges hálózatok hirdetése, passzív interfészek megadása)!
- Az `RTR_KOLCSONZO` forgalomirányítón hozzon létre OSPF folyamatot, majd hirdesse az összes közvetlenül csatlakozó hálózatot! Ügyeljen rá, hogy a hálózatok ugyanabban a területben legyenek hirdetve, mint az `RTR_KOZPONT` forgalomirányítón!
- Az `RTR_KOLCSONZO` forgalomirányítón a forgalomirányítási információk küldésére nem használt (al)interfészeket állítsa be passzívnak!
- Keresse meg, hogy melyik forgalomirányítón létezik már alapértelmezett útvonal! Ugyanezen a forgalomirányítón futó OSPF folyamat kiegészítésével érje el, hogy a másik forgalomirányító is megtanulja az alapértelmezett útvonalat!
- Ellenőrizze, hogy kialakult-e szomszédsági viszony a forgalomirányítók között!

15. A `WF_UGYFEL` vezeték nélküli forgalomirányító belső (LAN) IP-címének állítsa be a **192.168.111.0/25** hálózat utolsó kiosztható címét!

16. A `WF_UGYFEL` vezeték nélküli forgalomirányító DHCP szolgáltatását konfigurálja úgy, hogy a csatlakoztatott kliensek a **192.168.111.1 – 192.168.111.100** közötti IP-címeket kaphassák meg, és megkapják a megfelelő DNS kiszolgáló IP-címét (`SRV_DNS` szerver IP-címe) is!

17. A `WF_UGYFEL` vezeték nélküli forgalomirányítón állítsa át az SSID értékét **UGYFEL**-re és állítson be WPA2 hitelesítést AES titkosítással és **ABC123def** jelszóval!

18. Csatlakoztassa a vezeték nélküli klienseket (`TABLET1`, `TABLET2`) a `WF_UGYFEL` vezeték nélküli forgalomirányítóhoz! Az esetlegesen felmerülő hibákat hárítsa el!

19. Az `RTR_KOZPONT` forgalomirányítón állítson be statikus NAT szolgáltatást, amellyel biztosítja, hogy az `SRV_WEB` szerver kívülről (a `TesztPC`-ről) a **45.78.42.6/29** IP-címmel legyen elérhető!

20. Az `RTR_KOZPONT` forgalomirányítón állítson be dinamikus túlterheléses címfordítást (PAT), amellyel biztosítja, hogy a `VLAN 58` és a `VLAN 2` hálózatokból származó csomagok forráscíme az `RTR_KOZPONT` forgalomirányító külső (internethez csatlakozó) interfészének IP-címére forduljon le! A szükséges hozzáférési lista pontosan két bejegyzésből álljon, és mindkét bejegyzés egy-egy VLAN (`VLAN 58` illetve `VLAN 2`) számára engedélyezze a címfordításban való részvételt!

21. A két forgalomirányítón és a két kapcsolón mentse el a konfigurációt, hogy azok újraindítás után is megőrizze a beállításokat!

**A hálózat működését a következőképpen tesztelheti:**

- A VLAN hálózatok számára az internet elérését és a dinamikus túlterheléses címfordítás működését tesztelheti a vezeték nélküli kliensekről (`TABLET1`, `TABLET2`) vagy az irodai számítógépekről (`PC1_IRODA`, `PC2_IRODA`) a webböngészőbe írt **www.net.hu** URL címmel vagy a **89.54.7.2** IP-címmel.
- Az autókölcsönző weboldalának elérését és a statikus NAT működését a `TesztPC` webböngészőjébe írt **www.autoberles.hu** URL címmel tesztelheti.
- Az `SW_KOZPONT` kapcsoló SSH elérésének működőképességét a belső hálózat bármely eszközéről tesztelheti.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## 2. feladat

40 pont

### Bingó

A bingó egy népszerű játék, melyet sokféle módon játszanak. Ebben a feladatban a kártyákon egy 5x5-ös táblázatban 24 szám található és középen egy üres cella. A játék azzal kezdődik, hogy minden játékos kap egy kártyát, amelyen előre megadott számok szerepelnek. A játékmester egyesével húz ki 1 és 75 közötti számokat, a játékosok pedig megjelölik a saját kártyáikon a taláataikat. Ha a kártyán a kihúzott számok közül 5 egymás mellett vízszintesen, függőlegesen vagy átlósan szerepel, akkor a játékos nyer, ha hangosan elsőként BINGÓ-t kiált. Ebben a feladatban több játékos kártyáival kell feladatokat megoldania. A kártyákon található számokat a játékosok egyedi neveivel azonosított szöveges állományokban tároltuk. Megoldásában vegye figyelembe a következőket:

B I N G O				
7	25	44	57	62
15	22	40	50	70
11	30	FREE SPACE	46	74
2	28	37	55	68
10	27	39	59	75

- *Megoldását elkészítheti saját osztály definiálása és alkalmazása nélkül is, de úgy az nem lesz teljes értékű.*
- *A képernyőre írást igénylő feladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 4. feladat)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírások is elfogadottak.*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti.*
- *A program megírásakor az állományokban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon.*

A feladat forrás mappájában megtalálható `txt` kiterjesztésű szöveges állományok tartalmazzák mátrix-szerűen a kártyákon található számokat. A számokat pontosvesszővel választottuk el, a középső üres („Joker”) cellát egy „X” karakter jelöli, mely cellát minden játékos automatikusan találatnak tekinthet. Például `Marika.txt`:

```
2;28;43;59;74
7;25;38;53;69
1;22;X;47;70
12;23;33;57;65
15;29;39;52;67
```

1. Készítsen **konzolos alkalmazást** a következő feladatok megoldására, melynek projektjét `Bingo` néven mentse el!
2. Készítsen saját osztályt `BingoJatekos` azonosítóval, melynek adatait alkalmasság legyenek egy játékos nevének és kártyájának tárolására, találatainak jelölésére és a további feladatok megoldására. Az osztály konstruktora inicializálja az adatokat!

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

3. Olvassa be és tárolja a forrásállományokban kódolt játékosok neveit és kártyáit (az állomány neve egyben a játékos neve is) és tárolja el egy olyan adatszerkezetben, ami a további feladatok megoldására alkalmas! A választott adatszerkezet feltöltésekor használja a `BingoJatekos` osztály példányait! Legfeljebb 100 játékos (forrásállomány) lehet. Megoldásához **felhasználhatja** a `nevek.text` szöveges állományt, melyben soronként a feldolgozandó forrásállományok neveit gyűjtöttük ki. Ha csak egy állományt tud beolvasni és tárolni, akkor az `Andi.txt` állománnyal folytassa a feladatok megoldását.
4. Határozza meg és írja ki a képernyőre a játékosok számát!
5. Készítsen `SorsoltSzamotJelol` azonosítóval metódust (függvényt) a `BingoJatekos` osztályba, melynek segítségével a paraméterben átadott számot „**megjelöli**” a játékos kártyáját kódoló adatszerkezetben, ha a kisorsolt szám létezik.
6. Készítsen `BingoEll` azonosítóval jellemzőt vagy metódust (függvényt), melynek segítségével ellenőrizheti, hogy a játékos kártyáján **megjelölt** (kihúzott) számok teljesítik-e a bevezetőben ismertetett nyerési (BINGÓ) feltételt. A kódtag térjen vissza igaz értékkel, ha a játékosnak bingója van, hamissal, ha nincs.
7. Véletlenszerűen, az ismétlések elkerülésével állítsa elő a számhúzások értékeit 1-75 tartományban mindaddig, míg **egy vagy több** játékosnak bingója nem lesz. A megoldásában felhasználhatja az előző feladatokban elkészített metódusokat. A kihúzott számokat sorszámokkal írja a képernyőre! Ügyeljen a minta szerinti igazításra!
8. Jelenítse meg a lehetséges nyertes/nyertesek neveit és kártyáit a képernyőn! A megjelölt (kihúzott) számok és a „Joker, X” cella jelenjen csak meg táblázatos formában a kártyán, a ki nem húzott számokat „0” számjegy jelölje! A mátrix megjelenítésekor ügyeljen a minta szerinti igazításra!
9. Készítsen **grafikus alkalmazást** `BingoGUI` néven, melynek segítségével egy bingo kártyát tud előállítani és menteni!

A grafikus alkalmazásban a következő feladatokat végezze el:

- a. Alakítsa ki a felhasználói felületet a következő minta szerint! Állítsa be az alkalmazás címsorában megjelenő „*Bingo*” feliratot és a beviteli mező alapértelmezett értékét (`bingo.txt`)! A beviteli mezőben az állomány nevét tudja megadni a mentéshez!



*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- b. Ha a „Kártya generálása” parancsgombra kattintunk akkor a következőket hajtsa végre:
- Hozzon létre futási időben 5x5db beviteli mezőt, ha korábban már léteztek, akkor az újak létrehozása előtt törölje a régi mezőket, vagy azok tartalmát!
  - Töltse fel véletlenszerűen ismétlődések nélkül az oszlopokban lévő beviteli mezőket a következő értéktartományokból:
    - 1. oszlop: 1-15,
    - 2. oszlop: 16-30,
    - 3. oszlop: 31-45,
    - 4. oszlop: 46-60,
    - 5. oszlop: 61-75
  - A középső bevitelimező alapértelmezett értéke „X” legyen és oldja meg, hogy a benne lévő tartalmat ne lehessen szerkeszteni!
  - Mátrix-szerűen jelenítsen meg a beviteli mezőket az alkalmazásablak üres területén a következő minta szerint!

Bingo

Kártya generálása

9	23	35	57	72
13	27	33	48	66
7	22	X	53	64
15	18	40	51	63
12	17	41	47	69

bingo.txt

Mentés

- c. Ha a „Mentés” parancsgombra kattintunk, akkor mentse szöveges állományba a megadott néven a beviteli mezők tartalmát a következő minta szerint! Az adatokat pontosvesszővel válassza el! Például a `bingo.txt` állomány:

```
9;23;35;57;72
13;27;33;48;66
7;22;X;53;64
15;18;40;51;63
12;17;41;47;69
```



--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

### Konzolos alkalmazás minták:

```
4. Feladat: Játékosok száma: 11

7. feladat: Kihúzott számok
1.->38 2.->5 3.->37 4.->20 5.->11 6.->21 7.->45 8.->56 9.->40 10.->65
11.->22 12.->32 13.->48 14.->59 15.->54 16.->23 17.->62 18.->60 19.->29 20.->4
21.->30 22.->35 23.->57 24.->44 25.->15 26.->43 27.->47 28.->3

8. Feladat: Lehetséges nyertes(ek):
Edit
0 0 0 0 0
0 0 44 54 0
3 30 X 57 62
0 0 32 0 0
0 0 0 0 0
```

A fenti mintában Editnek a középső sorból húztak ki 4 számot, így a „Joker, X” cellával bingója van.

```
4. Feladat: Játékosok száma: 11

7. feladat: Kihúzott számok
1.->74 2.->62 3.->66 4.->14 5.->48 6.->28 7.->35 8.->59 9.->71 10.->6
11.->58 12.->68 13.->24 14.->12 15.->27 16.->45 17.->51 18.->31 19.->54 20.->23
21.->19 22.->50 23.->2 24.->65 25.->20 26.->30 27.->57 28.->44 29.->52

8. Feladat: Lehetséges nyertes(ek):
Anna
0 0 31 57 62
0 0 0 52 0
0 0 X 50 71
0 24 35 0 0
14 0 0 54 0
Gabi
0 0 45 51 0
0 24 0 54 0
0 0 X 48 65
2 0 0 52 62
0 28 0 58 74
```

A fenti mintában Annának átlósan húztak ki 4 számot, így a „Joker, X” cellával bingója van. Gabinak az utolsó előtti oszlopból húzták ki mind az öt számot, így neki is bingója van.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

### 3. feladat

40 pont

#### Hamburger

A következő feladatban egy hamburgerekre specializálódott étterem weboldalán kell részfeladatokat elvégeznie.

A feladat két részből áll:

- A Boston Burger szolgáltatásait népszerűsítő, publikus weboldal véglegesítése („Weboldal kódolása” feladatrész)
- Az étterem online rendeléseit tároló adatbázison lekérdezések írása az étterem tulajdonosai számára.

A webalkalmazás főbb elemei már rendelkezésre állnak a vizsgakönyvtár

3\_feladat\hamburger mappájában. Csak a feladatokban felsorolt állományokat módosítsa!

A weboldal már tartalmaz hivatkozást a 3\_feladat\hamburger\web\assets\css mappán belül található `bootstrap.css` fájlra, ami a Bootstrap 4 keretrendszer stíluslapja. Szintén rendelkezésre állnak a Bootstrap 4 rendszer teljeskörű működéséhez szükséges JavaScript fájlok is (`bootstrap.js`, `jquery-3.4.1.min.js`).

A webszerver indításához a Windows parancssorában tegye aktuálissá a vizsgakönyvtár 3\_feladat\hamburger könyvtárát, majd futtassa az **npm run start** parancsot! A webszerver indítása után az étterem nyilvános oldala a `http://localhost:8080` URL-en keresztül érhető el. Ügyeljen rá, hogy a parancssor ablakát a feladat megoldása közben **ne zárja be** és **ne jelöljön benne ki semmit**, mert az a webszerver futását megállítja!

Amennyiben munkája közben a mappa tartalmában véletlenül olyan módosításokat végezne, ami után a kiszolgáló nem működik megfelelően, akkor a helyreállításhoz a mappa eredeti tartalmát megtalálja a forrásállományok között található `hamburger.zip` tömörített állományában.

#### Weboldal kódolása

Az alábbi utasításoknak megfelelően végezze el a weboldal fejlesztését. Ügyeljen rá, hogy az oldal a módosítások után is megtartsa a rezponzív viselkedését.

A feladatok megoldásához a következő állományokat kell módosítania:

- 3\_feladat\hamburger\web\index.html
- 3\_feladat\hamburger\web\hamburger.css
- 3\_feladat\hamburger\web\hamburger.js

A munkája végén ezeket az állományokat a feladatlap „Az elkészült munka beadása” részben megadottak szerint a vizsgakönyvtár 3\_feladat\beadott nevű mappájába kell másolnia!

Az egyedi formázási beállításokat a `hamburger.css` stílusállományban, a kódolási feladatokat pedig a `hamburger.js` JavaScript fájlban végezze el!

1. Helyezzen el megfelelő hivatkozást, hogy az oldal megnyitásakor megjelenjen az oldal favicon-ja! A link típusú elem attribútumai:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

rel	icon
type	image/x-icon
href	assets/favicon.ico

- Helyezzen el hivatkozást a `hamburger.css` stíluslapra úgy, hogy az felülírja az összes többi CSS stílusfájl beállításait!
  - Állítsa be a dokumentumban használt betűtípust és betűméretet az alábbiak szerint:
    - Az alapértelmezett betűtípus az egész dokumentumban „Open Sans” legyen, tartalék betűtípusnak pedig a sans-serif betűtípust adja meg.
    - A `h1` és `h2` elemek betűtípusa „Dancing Script” legyen, cursive tartalékkal.
    - Az alapértelmezett betűvastagság az egész dokumentumban 600 legyen.
  - Alakítsa ki a weboldal navigációs részét az alábbiak szerint:
    - „Navigáció” részben hozza létre az alábbi menüelemeket:
      - Kezdőlap
      - Menü
      - Rólunk
      - Asztalfoglalás
    - Ehhez a „Menüelemek” szöveget cserélje ki egy négyelemű számozatlan listára, a lista külső HTML-eleme (`<ul>`) `navbar-nav` és `mx-auto`, a listaelemek pedig `nav-item` osztálykijelölővel legyenek formázva!
      - Az egyes listaelemekhez hozzon létre hivatkozásokat, melyek közül az első az aktuális weboldal tetejére, a továbbiak pedig sorrendben a `menu`, `rolunk`, és `asztalfoglalas` dokumentum szintű azonosítókra mutatnak!
      - A hivatkozások a `nav-link` osztálykijelölővel legyenek megformázva!
  - A böngészője fejlesztői eszközeinek a segítségével állapítsa meg az oldal fejlécében található „Rendelés” gomb háttérszínét! Állítsa be, hogy a `btn1` osztály háttérszíne is ez a szín legyen! Érje el, hogy minden esetben felülírja ez a beállítás az ezzel az osztállyal rendelkező elemek háttérszínét!
- Ha nem tudja megállapítani a háttérszínt, használja a piros színt a `btn1` osztály háttérszínének.*
- A menü szekcióban található, `ajanlo-1` és `ajanlo-2` div elemek esetén érje el beépített Bootstrap osztályok és reszponzív rács segítségével, hogy a kártyák közepes és annál nagyobb képernyőn egymás mellett, kettő oszlopban jelenjenek meg!
  - A Menü kártyái között az egyik elem még csak ideiglenes helykitöltő információkkal van ellátva. Módosítsa a tartalmát a `Forrás/húsimádó.txt` fájlban leírtak alapján!
  - Helyezzen el hivatkozásokat a következő fájlokra ebben a sorrendben a weboldal végén!
    - `assets/js/custom.js`
    - `hamburger.js`

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

9. A `hamburger.js` fájlban hozzon létre egy olyan JavaScript kódot, ami a weboldal `asztalfoglalas` szekciójában található információkat (név, telefonszám, e-mail cím, vendégek száma, időpont) elküldi a webszervernek, ha a „FOGLALÁS BEKÜLDÉSE” parancsgombra kattintunk!

A REST-kérés paraméterei:

- Kérés típusa: POST
- URL: `/api/foglalas`
- Elküldött adat típusa: JSON
- Elküldött adat szerkezete:

```
{  
  name:String, (név, pl. „Kovács Andrea”)  
  phone:String, (telefonszám, pl. „+36201234567”)  
  email:String, (e-mail cím, pl. „kovacs.andrea@mail.com”)  
  seats:number, (vendégek száma, pl. „2”)  
  datetime:date, (foglalás dátuma, pl. „2021.11.24 20:00”)  
}
```
- Választípus: JSON
- Válaszüzenet sikeres küldés esetén: `{feldolgozva:boolean}`  
(pl. `{feldolgozva:true}`)

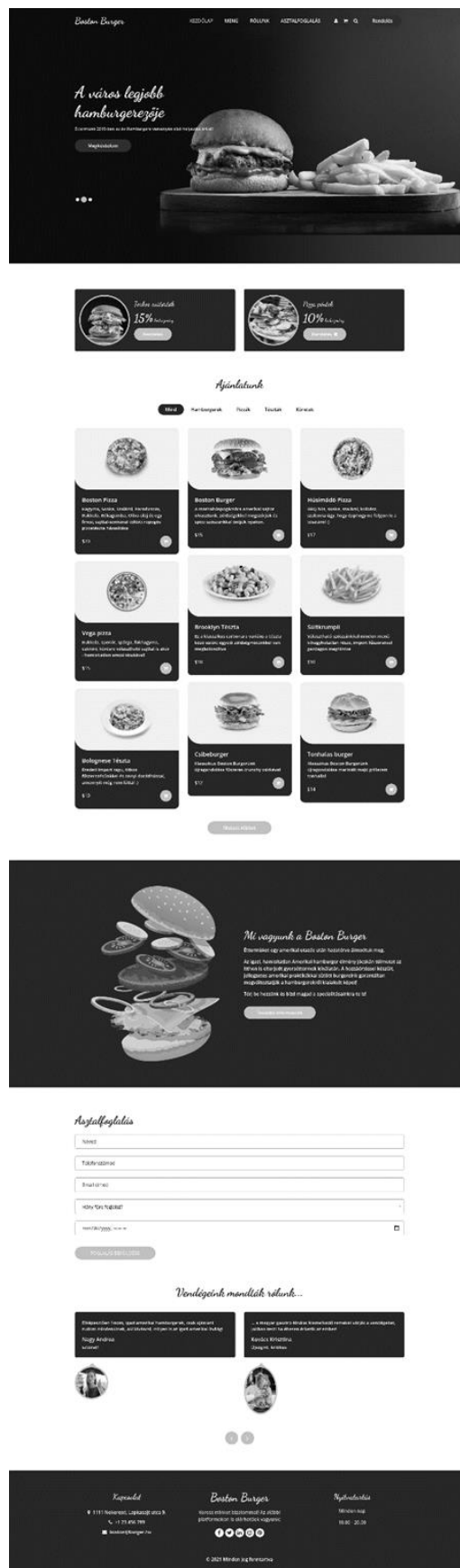
Amennyiben valamelyik mező értéke üres, akkor a „Kérlek, minden mezőt tölts ki az űrlapon!” üzenetet jelenítse meg egy felugró ablakban, és ne küldje be az űrlap adatait a szervernek!

Amennyiben a küldés sikeres (200-as státuszkód), törölje a weboldalon nevet, telefonszámot és e-mail címet, majd egy felugró ablakban jelenítse meg az alábbi üzenetet: „Foglalását regisztráltuk!”

Amennyiben nem tudja kiolvasni a megfelelő adatokat az űrlapból, akkor a példaként feltüntetett információkkal küldje el a kérést a szerver felé.

*A feladat a következő oldalon folytatódik*

**Minta:** (A megoldás szövegének tagolása felbontástól függően eltérhet a képen láthatótól.)



*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## Adatbázis-kezelés

A hamburgerező adatbázisa egy tetszőleges nap összes rendeléséhez kapcsolódó információt tartalmazza. A vezetőség számára kimutatásokat kell készítenie az adatok elemzéséhez.

Az Ön feladata az SQL lekérdezések elkészítése és beillesztése a vizsgakönyvtárban található **3\_feladat\hamburger\lekerdezések\lekerdezések.sql** fájlba. A munkája végén ezt az állományt a feladatlap „Az elkészült munka beadása” részében megadottak szerint a **3\_feladat\beadott** nevű mappájába át kell másolnia! Figyeljen rá, hogy az állomány szerkezetét ne módosítsa (pl. ne törölje a \*\*\* karaktereket tartalmazó elválasztó sorokat), mert ellenkező esetben az adminisztrációs oldal nem lesz képes megjeleníteni a lekérdezések eredményét. Az SQL parancsokat több sorba is törtheti.

Az elkészített lekérdezéseit a <http://localhost:8080/admin> oldal megnyitásával tesztelheti. Továbbá itt ellenőrizheti a webszerver és a MySQL adatbázis elérhetőségét is.

Ügyeljen arra, hogy a lekérdezésben pontosan a kívánt mezők és mezőnevek szerepeljenek, és felesleges mezőt ne jelenítsen meg!

Az adatbázis a következő táblákat tartalmazza:

### felhasznalo

id	Egész szám, egy vendég azonosítója, <b>PK</b>
nev	Szöveg, a felhasználó teljes neve (egyedi érték)

### etelkategoria

id	Egész szám, egy ételkategoria azonosítója, <b>PK</b>
nev	Szöveg, az ételkategoria neve (egyedi érték)

### menutétel

id	Egész szám, egy étel azonosítója a menüben, <b>PK</b>
nev	Szöveg, a rendelhető étel teljes neve
ar	Egész szám, egy rendelhető étel bruttó ára
etelkatId	Egész szám, az ételkategoria azonosítója, <b>FK</b>

### rendeles

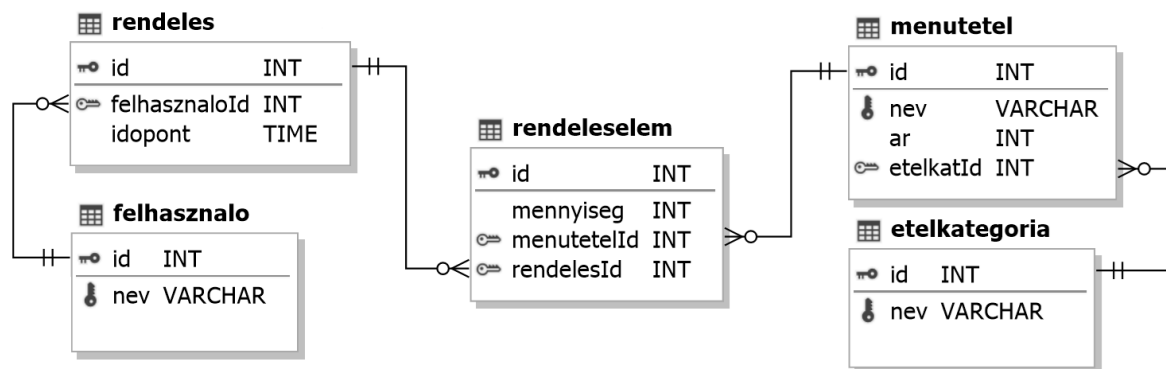
id	Egész szám, egy rendelés azonosítója, <b>PK</b>
felhasznaloId	Egész szám, a rendelést feladó felhasználó azonosítója, <b>FK</b>
idopont	A rendelés időpontja, „ÓÓ:PP” formátumban (24 órás)

### rendeleselem

id	Egész szám, a rendelés kategória azonosítója, <b>PK</b>
mennyiseg	Egész szám, a rendelt tétel darabszáma
menutetelId	Egész szám, a rendelt menütétel azonosítója, <b>FK</b>
rendelesId	Egész szám, a rendelés azonosítója, <b>FK</b>

Az elsődleges kulcsokat **PK**-val, az idegenkulcsokat **FK**-val jelöltük.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--



10. Hozzon létre a lokális SQL serveren `hamburger` néven adatbázist! Az adatbázis alapértelmezett rendezési sorrendje a magyar szabályok szerint legyen! Állítsa be az UTF-8 kódolást alapértelmezettnek az adatbázis létrehozásánál! **(10. feladat)**
11. Forrás mappában található `adattabazis.sql` állomány tartalmazza a táblákat létrehozó és az adatokat a táblákba beszűrő SQL parancsokat! Futtassa az `adattabazis.sql` parancsfájlt a `hamburger` adatbázisban!
12. Listázza ki az összes olyan menütételt, amely pontosan 2500 forintba kerül! Jelenítse meg az összes oszlopát az adattáblának! **(12. feladat)**
13. Az adatbázisban a „Grill pizza 32cm” forintban megadott ára 2300 helyett hibásan 23000 összeggel került tárolásra. SQL parancs segítségével módosítsa a tétel árát a helyes összegre! **(13. feladat)**
14. Állapítsa meg a kilenc, legtöbb rendelést leadó felhasználó nevét! A lekérdezésben jelenítse meg a rendelések darabszámát „rendeles\_db” és a felhasználó nevét „vendeg\_nev” álnevekkel (oszlopfejléccel)! *Feltételezheti, hogy a 9. és a 10. helyen nem alakult ki holtverseny.* **(14. feladat)**
15. Adjon hozzá a `menutetel` táblához egy új bejegyzést a következő adatokkal: **(15. feladat)**
  - `nev:` Boston Tészta
  - `ar:` 2200
  - `etelkatId:` 3
16. Listázza ki Balázs Flóra összes rendelésének összes menütételét! A lekérdezésben a „rendeles\_azon” oszlopban jelenítse meg a rendelés azonosítóját, a „rendeleselem\_azon” oszlopban jelenítse meg a rendeleselem azonosítóját, a „menutetel\_neve” oszlopban pedig a rendeleselemhez tartozó menütétel nevét! **(16. feladat)**
17. Állapítsa meg, hogy melyik felhasználó fizetett a legtöbbet az összes megrendelése után!

Egy rendelés teljes ára a következők szerint határozható meg:

**alapdíj = mennyiség \* tétel ára**

**osszeg = alapdíj + jutalék**

ahol a jutalék az ételszállító cég 10%-os jutaléka, melyet az alapdíjra számolnak fel.

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

A megoldásban jelenítse meg a felhasználó nevét („felhasznalo\_nev”) és a végösszeget („osszesen”)! *Feltételezheti, hogy nem alakult ki holtverseny.* **(17. feladat)**

### Az elkészült munka beadása

Munkájának befejezése után másolja át az alábbi állományokat a vizsgakönyvtár beadott mappájába!

- 3\_feladat\hamburger\web\index.html
- 3\_feladat\hamburger\web\hamburger.css
- 3\_feladat\hamburger\web\hamburger.js
- 3\_feladat\hamburger\lekerdezések\lekerdezések.sql

**Munkája a „Weboldalak kódolása és adatbázis-kezelés” feladatnál csak a beadott mappában található állományok tartalma alapján lesz értékelve! Amennyiben más fájlokat is létrehozott vagy módosított, azok a javítás során nem lesznek figyelembe véve.**



--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

témakörök	a feladat sorszáma	pontszám	
		maximális	elért
Hálózati ismeretek	1.	<b>40</b>	
Programozás	2.	<b>40</b>	
Weboldalak kódolása, adatbázis-kezelés	3.	<b>40</b>	
<b>A gyakorlati vizsgarész pontszáma</b>		<b>120</b>	

\_\_\_\_\_  
dátum

\_\_\_\_\_  
javító tanár

	pontszáma <b>egész számra</b> kerekítve	
	elért	programba beírt
Számítógépen megoldott gyakorlati feladatok		

\_\_\_\_\_  
dátum

\_\_\_\_\_  
dátum

\_\_\_\_\_  
javító tanár

\_\_\_\_\_  
jegyző