

Azonosító
jel:

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

ÉRETTSÉGI VIZSGA • 2021. október 22.

**INFORMATIKAI
ISMERETEK**

**EMELT SZINTŰ
GYAKORLATI VIZSGA**

2021. október 22. 8:00

Időtartam: 240 perc

Pótlapok száma	
Tisztázati	
Piszkozati	

Jelölje be az Ön által választott
programozási nyelvet!
(Csak egy nyelvet jelölhet meg.)

Java ☐

C# ☐

EMBERI ERŐFORRÁSOK MINISZTERIUMA

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

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.)

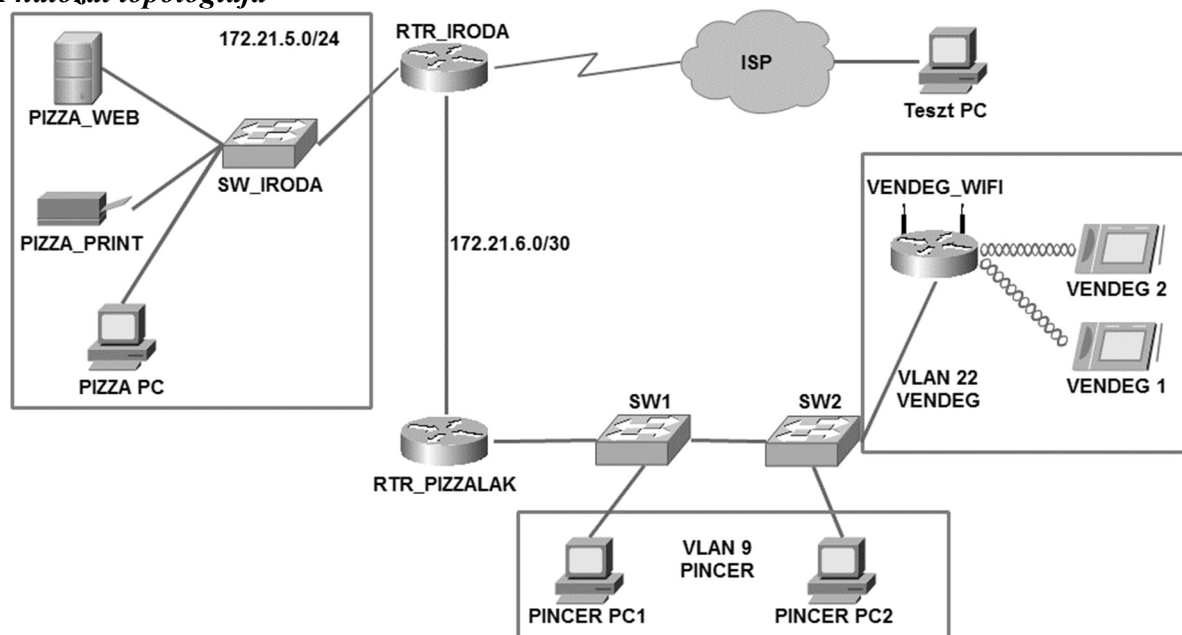
1. feladat

40 pont

PIZZANET

Egy újonnan nyíló pizzázó számítógépes hálózatának tervezésével bízták meg. A tervezés első fázisában a logikai topológiát már elkészítette egy kollégája, illetve néhány konfigurációs lépés is elvégzésre került a szimulált logikai topológián. Az Ön feladata, hogy a megadott tervek és elváráslista alapján szimulációs programmal elkészítse a pizzázó teszhálózatát.

A hálózat topológiája



Beállítások

- Töltse be a pizzanet.pkt állományt a szimulációs programba! A teszhálózat már tartalmazza a vállalat összes hálózati eszközét és az internet szimulálására szolgáló eszközöket. Ez utóbbiak már beállításra kerültek. A vállalati eszközök részleges konfigurációval már rendelkeznek, Önnek csak a feladatokban leírt módosításokat kell elvégeznie.
- Az RTR_PIZZALAK forgalomirányító helyi hálózatában a **10.20.30.0/24** privát címtartományt szeretnék használni. A két VLAN számára VLSM használatával a lehető leghatékonyabban alakítson ki alhálózatokat a következő IP-cím igények figyelembevételével:

VLAN azonosítója	VLAN neve	Igényelt IP-címek száma
9	PINCER	28
22	VENDEG	2

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

A feladat a következő oldalon folytatódik.

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

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

VLAN száma	VLAN neve	Hálózat cím	Netmaszk
9	PINCER	172.23.9.0	255.255.255.192
22	VENDEG	172.23.10.0	255.255.255.224

- Az RTR_PIZZALAK forgalomirányítón hozza létre a szükséges alinterfészeket úgy, hogy az alinterfész száma egyezzen meg a használt VLAN azonosító számával! Az alinterfészekre állítsa be a megfelelő hálózat utolsó kiosztható IP-címét!
- Az SW2 kapcsolón hozza létre a VLAN 9-et és a VLAN 22-t, és a fenti táblázat alapján állítsa be a VLAN-oknak a PINCER és a VENDEG nevet! (Az SW1 kapcsolón már léteznek a szükséges VLAN-ok.)
- Az SW1 és az SW2 kapcsolók megfelelő portjainak konfigurálásával érje el, hogy a kliensekhez csatlakozó portok hozzáférési portok legyenek, és a megfelelő VLAN-ba kerüljenek!
- Az SW1 és az SW2 kapcsolókon, ahol szükséges, használjon trónk portot, hogy a hálózat működőképes legyen!
- A VENDEG WIFI vezeték nélküli forgalomirányító Internet portja számára statikusan állítsa be a megfelelő hálózat első kiosztható IP-címét, a megfelelő alapértelmezett átjárót és DNS kiszolgálót (4.4.4.4)!
- A PINCER VLAN számára már létezik egy DHCP hatókör az RTR_PIZZALAK forgalomirányítón. Egészítse ki a meglévő DHCP konfigurációt az alábbi figyelembevételével:
 - Biztosítsa az összes szükséges, jelenleg hiányzó paraméter átadását!
 - Az utolsó 2 címet ne oszthassa ki a DHCP kiszolgáló!
 Állítsa be a PINCER VLAN számítógépeit a dinamikus IP-cím használatához!
- A PIZZA_WEB szervernek statikusan állítsa be a megfelelő hálózat első kiosztható IP-címét, a megfelelő alapértelmezett átjárót és DNS kiszolgálót: **4.4.4.4!**
- Az SW_IRODA kapcsoló PIZZA_WEB szerverhez csatlakozó portján engedélyezzen portvédelmet úgy, hogy csak a jelenleg csatlakozó szerver számára legyen engedélyezve a kapcsolat. A portvédelem beállításakor a PIZZA_WEB szerver fizikai címét statikusan rögzítse! Ha más fizikai című eszköz használja a portot, akkor a beérkező keret ne kerüljön továbbításra, de a port ne kapcsoljon le!
- Az SW_IRODA kapcsoló legalább első 5 virtuális vonalán állítsa be, hogy távolról telnet protokollal lehessen elérni és menedzselni az eszközt! Ehhez vegye figyelembe a következőket:
 - Állítsa be az alapértelmezett felügyeleti VLAN számára a használt hálózat utolsó előtti IP-címét, és adja meg a megfelelő alapértelmezett átjárót!
 - A privilegizált módot védő jelszó a **pizzairoda** legyen!
 - A telnet elérés jelszava **pizzairoda** legyen!
- Az RTR_IRODA forgalomirányítón vegyen fel az internet felé vezető alapértelmezett statikus IPv4 útvonalat! A megadásnál használja a következő ugrás IP-címét, mely a csatlakozó hálózat első kiosztható IP-címe!

A feladat a következő oldalon folytatódik.

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

13. Az IPv4-es forgalom irányításához RIPv2 protokollt használnak. Állítsa be mindkét forgalomirányítón a RIPv2 protokollt a következők szerint:
 - a. Mindkét forgalomirányítón hirdesse az összes közvetlenül csatlakozó hálózatot, kivéve az RTR_IRODA forgalomirányítón! Itt az internet (ISP) felé menő hálózatot ne hirdesse!
 - b. Állítsa passzívra azon (al)interfészeket, amelyekhez LAN-ok csatlakoznak, és nincsenek forgalomirányításra használva!
 - c. Érje el, hogy a forgalomirányítók ne az osztályos hálózatokat, hanem a használt alhálózatokat hirdessék!
 - d. Az RTR_IRODA forgalomirányítón futó RIPv2 folyamat kiegészítésével érje el, hogy a másik forgalomirányító is megtanulja az alapértelmezett útvonalat!
14. A VENDEG WIFI belső hálózatához tartozó IP-címét állítsa át a 192.168.44.254/24-es címre! A vezeték nélküli eszköz DHCP szolgáltatását konfigurálja úgy, hogy a csatlakoztatott kliensek a 192.168.44.1 – 192.168.44.60 közötti IP-címeket kaphassák meg, és megkapják a megfelelő DNS kiszolgáló IP-címét (4.4.4.4) is!
15. A VENDEG WIFI vezeték nélküli eszközön állítsa át az SSID értékét **PIZZANET**-re, és állítson be WPA2 hitelesítést **szeretemapizzat** jelszóval!
16. Csatlakoztassa a vezeték nélküli klienseket (VENDEG 1, VENDEG 2) a VENDEG WIFI eszközhöz!
17. Az RTR_IRODA forgalomirányítón állítson be statikus NAT szolgáltatást, amellyel biztosítja, hogy a PIZZA_WEB szerver kívülről (a Teszt PC-ről) a 187.54.20.6 IP-címmel legyen elérhető!
18. Az RTR_IRODA forgalomirányítón dinamikus portalapú címfordítást (PAT) kell konfigurálni az RTR_PIZZALAK VLAN hálózatai és a PIZZA PC kliens számítógép számára. A címfordítás konfigurációja részben már elkészült. Feladata, hogy a használt – már létező – hozzáférési listát egészítse ki a két VLAN hálózatával, majd az érintett interfészek esetén állítsa be a címfordítás irányát!
19. A PIZZA_WEB szerver számára az internet irányából IPv6-os elérést is kell biztosítani. Ehhez az RTR_IRODA forgalomirányítón engedélyezze az IPv6-os forgalomirányítást, majd az alábbiak szerint állítson be IPv6-címeket:
 - a. Serial0/0/0 interfészének globális IPv6 cím: **3300::2/64**
 - b. GigabitEthernet0/1 interfészének globális IPv6 cím: **2055:21::1/64**, link-local cím: **FE80::1**
20. Az RTR_IRODA forgalomirányítón vegyen fel alapértelmezett IPv6 útvonalat az internet (ISP) irányába a **3300::1** következő ugrás cím használatával!
21. A PIZZA_WEB szervernek állítsa be a **2055:21::50/64** IPv6-címet! Az alapértelmezett átjáró a forgalomirányító link-local címe legyen!
22. A forgalomirányítókra és a kapcsolókra mentse el a konfigurációt, hogy azok újraindítás után is megőrizzék a beállításokat!

A hálózat működését a következőképpen tesztelheti:

- Az internet elérését és a dinamikus túlterheléses címfordítás működését tesztelheti a belső hálózat kliens gépeiről a webböngészőbe írt **www.internet.hu** URL-címmel vagy a **4.4.4.4** IP-címmel.
- A pizzázó weboldalának elérését és a statikus NAT működését a Teszt PC webböngészőjébe írt **www.pizzalak.hu** URL-címmel tesztelheti.
- A pizzázó weboldalának IPv6-os elérését a Teszt PC webböngészőjébe írt **2055:21::50** címmel tesztelheti.

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

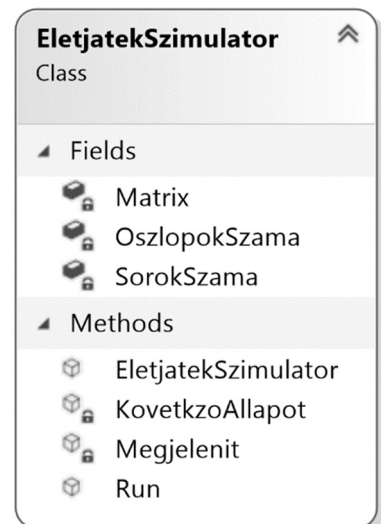
2. feladat

40 pont

Életjáték szimulátor¹

Az életjátékot John Horton Conway, a Cambridge-i Egyetem matematikusa találta ki. Játékként való megnevezése megtévesztő lehet, mivel „nullszemélyes” játék. A „játékos” szerepe mindössze annyi, hogy megad egy kezdőalakzatot, és azután csak figyeli az eredményt. Matematikai szempontból az ún. **sejtautomaták** közé tartozik. A játék egyes lépéseinek eredményét számítógép számítja ki. Ebben a feladatban egy életjáték szimulátort kell készítenie.

- Készítsen **konzolos alkalmazást** a következő feladatok megoldására, melynek projektjét Eletjatek néven mentse el!
- Készítsen saját osztályt EletjatekSzimulátor azonosítóval, melynek kód- és adattagjainak azonosítóit és láthatósági szintjét az osztálydiagram szemlélteti! A privát adattagokat egy lakat szimbólum különbözteti meg a publikusaktól. Ékezetes azonosítókat is készíthet, illetve azokat kis- és nagybetűkkel is kezdheti.
- Az EletjatekSzimulátor osztály konstruktora kapja paraméterül a mátrix méretét egész típusú változóknak!
- A konstruktor állítsa be a paraméterek értékeivel az adattagok (OszlopokSzama, SorokSzama) értékeit! Ugyancsak a konstruktorban inicializálja a Matrix azonosítójú, egész típusú mátrixot (kétdimenziós tömböt) technikai okokból a megadott méretnél **kettővel több sorral és oszloppal**, majd tölts fel úgy véletlenszerűen 0 vagy 1 értékekkel a „belső cellákat”, hogy a „külső cellák” (szélső sorok és szélső oszlopok) 0 értékűek legyenek mind a négy oldalon!
- Készítse el a megjelenítést végző metódust! A metódus a mátrixot jelenítse meg úgy a konzolon, hogy a „külső cellákat” az „X” karakterrel, a „belső cellákat” 0 érték esetén szóközzel, 1 érték esetén „S” (**sejt**) karakterrel jelölje a minta szerint!
- Kódolja a következő állapotot (kört) meghatározó metódust a következők szerint:
 - A mátrixban lévő **1-es értékeket sejteknek** nevezzük. Egy cella környezete a hozzá legközelebb eső maximum 8 cella (tehát a cellához képest „átlósan” elhelyezkedő cellákat is figyelembe vesszük). Egy sejt vagy üres cella szomszédjai a környezetében lévő sejtek (sejtszomszédok).



```

XXXXXXXXXXXX
XS  S S SSSX
XSSSSSSS SSX
XS  SSSSS  X
X  SS      X
X SSS S    X
XSSSSSSSS X
XS SSSS SS X
X  SS  S SX
XS      S X
X SS SSSSS X
XXXXXXXXXXXX
    
```

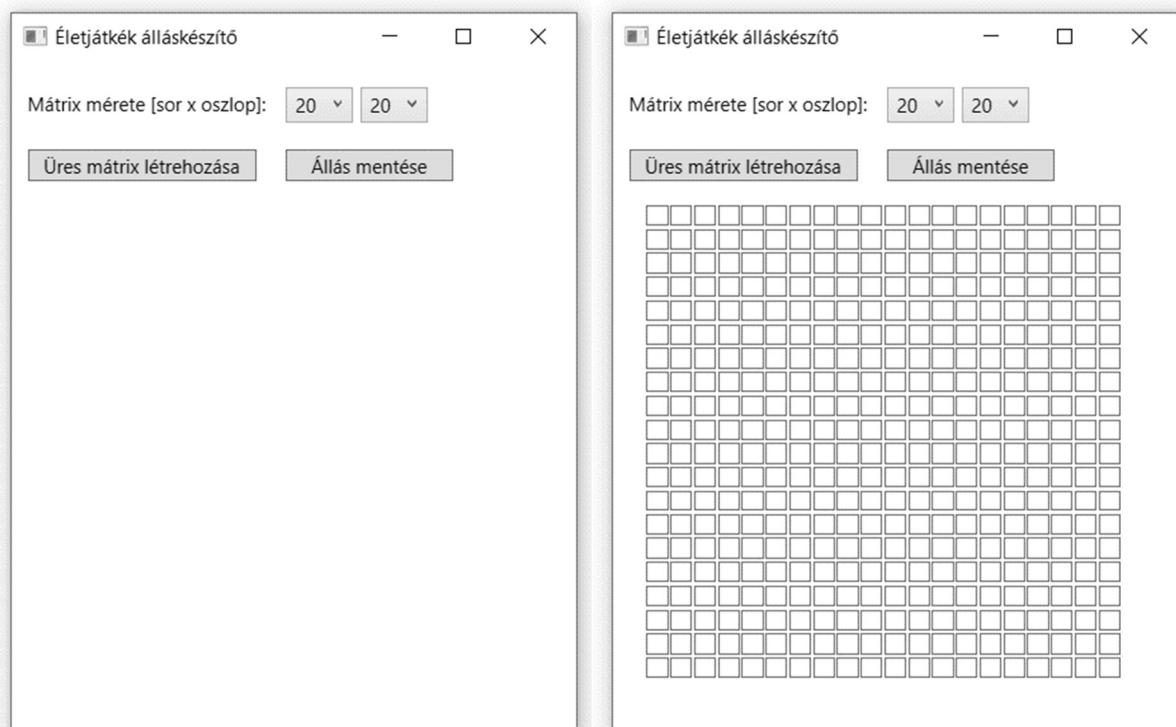
A feladat a következő oldalon folytatódik.

¹ Forrás: <https://hu.wikipedia.org/wiki/Életjáték>

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

- Egy sejtrel vagy üres cellával egy körben a következő három dolog történhet:
 - A sejt túléli a kört (cella értéke 1 marad), ha két vagy három szomszédja van.
 - A sejt elpusztul (cella értéke 0 lesz), ha kettőnél kevesebb (elszigetelődés) vagy háromnál több (túlnépesedés) szomszédja van.
 - Új sejt születik minden olyan üres (0 értékű) cellában, melynek környezetében pontosan három sejt található.
 - Minden egyes „belső cellára” a sejtszomszédok számának meghatározása után rendre alkalmazza a fenti három szabályt! A sejtszomszédok számának meghatározása az előző állapotú mátrixon történjen, azaz az új állapotot kódoló mátrix értékeit csak a módszer végén töltsse vissza a *Matrix* adattagba!
7. A `Run()` azonosítójú módszerben jelenítse meg a mátrixot, határozza meg a következő kör állapotát a privát módszerek hívásával, majd várakozzon 500 ezredmásodpercet a program végrehajtása!
 8. A főprogramban hozzon létre egy *EletjatekSzimulator* példányt 10x10-es mérettel a minta szerint, majd ismételten hívja a `Run()` módszerét egy billentyű leütéséig!
 9. Készítsen **grafikus alkalmazást**, melynek a projektjét *EletjatekGUI* néven mentse el!

A grafikus alkalmazásban a következő feladatokat végezze el:



- a. Alakítsa ki a felhasználói felületet a fenti, bal oldali minta szerint! Az alkalmazás címsorában megjelenő felirat „Életjáték álláskészítő” legyen! A mátrix méretét a legördülő listák segítségével lehessen kiválasztani, melyek értékeit töltsse fel az alkalmazás indulásakor 5–20 számokkal! Oldja meg, hogy mindkét listában a 20-as érték legyen az alapértelmezett!

A feladat a következő oldalon folytatódik.

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

- b. Az „Üres mátrix létrehozása” feliratú parancsgomb lenyomása után hozzon létre programjával jelölőnégyzet (CheckBox) példányokat mátrixszerűen elrendezve a kiválasztott méretnek megfelelően a fenti, jobb oldali minta szerint! Ha korábban már volt létrehozott jelölőnégyzet-mátrix a felhasználói felületen, akkor a létrehozás előtt törölje azt!
- c. Az „Állás mentése” parancsgomb lenyomása után hozzon létre szöveges állományt, mely a mátrix sorait kódolja! Az állomány azonosítója például `Eletjatek_5x6.txt` legyen, ahol a számértékek a mátrix méretének megfelelően változnak! Az üres jelölőnégyzeteket 0-val, a kipipált állapotúakat 1-gyel jelölje a következő minta szerint!

Mátrix mérete [sor x oszlop]: 5 6

Üres mátrix létrehozása

Állás mentése

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

➔

Eletjatek_5x6.txt

0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	1	1	1	0
0	1	0	0	1	0

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

3. feladat

40 pont

Parkolóház

A következő feladatban egy parkolóház weboldalán kell részfeladatokat elvégeznie.

A feladat két részből áll:

- A parkolóház szolgáltatásait népszerűsítő, publikus weboldal véglegesítése („Weboldal kódolása” feladatrész)
- A parkolóház bérlői adatbázisához lekérdezések elkészítése a parkolóház tulajdonosai számára („Adatbázis-kezelés” feladatrész)

A webalkalmazás főbb elemei már rendelkezésére állnak a vizsgakönyvtár 3_feladat\parkolohaz mappájában. Csak a feladatokban felsorolt állományokat módosítsa!

A webszerver indításához a Windows parancssorában tegye aktuálissá a vizsgakönyvtár 3_feladat\parkolohaz nevű mappáját, majd futtassa az **npm run start** parancsot! A webszerver indítása után a parkolóház 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, melyek után a kiszolgáló nem működik megfelelően, akkor a helyreállításhoz a mappa eredeti tartalmát megtalálja a forrásállományok között található parkolohaz.zip tömörített állományában.

Weboldal kódolása

Az alábbi utasításoknak megfelelően végezze el a nyitólap fejlesztését! Ügyeljen rá, hogy az oldal a módosítások után is megtartsa a reszponzív viselkedését!

A feladat megoldása során a következő állományokat kell módosítania:

- 3_feladat\parkolohaz\web\index.html
- 3_feladat\parkolohaz\web\parkolohaz.css
- 3_feladat\parkolohaz\web\parkolohaz.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.

A formázási beállításokat a parkolohaz.css stílusállományban, a kódolási feladatokat pedig a parkolohaz.js JavaScript fájlban végezze el!

1. Helyezzen el megfelelő hivatkozást, hogy a böngészőben megnyitva megjelenjen az oldal faviconja! A link típusú elem attribútumai:

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

2. Helyezzen el hivatkozást a parkolohaz.css stíluslapra úgy, hogy az felülírja az összes többi CSS stílusfájl beállításait!
3. Állítsa be, hogy a böngésző címsorában megjelenő cím „Parkolóház” legyen!

A feladat a következő oldalon folytatódik.

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

4. Alakítsa ki a weboldal navigációs részét az alábbiak szerint:
 - a. A „*Navigáció*” részben hozza létre az alábbi menüelemeket:
 - Szolgáltatások
 - Rólunk
 - KapcsolatEhhez a „*Menüelemek*” szöveget cserélje ki egy háromelemű számozatlan listára, a lista külső HTML-eleme () navbar-nav és ml-auto, a listaelemek pedig nav-item osztálykijelölővel legyenek formázva!
 - b. Az egyes listaelemekhez hozzon létre hivatkozásokat, melyek sorrendben a szolgáltatások, rolunk és kapcsolat dokumentum szintű azonosítókra mutatnak!
 - c. A hivatkozások a nav-link, rounded és js-scroll-trigger osztálykijelölőkkel legyenek megformázva!
5. Böngészője fejlesztői eszközeinek segítségével állapítsa meg, hogy az „*Üzenet küldése*” feliratú gombnak mi a háttérszíne! A bg-primary osztály módosításával érje el, hogy a „*Fejléc*” résznek és a „*Rólunk*” szekciónak ugyanez a szín legyen a háttérszíne!
6. A „*Szolgáltatások*” szekcióban érje el beépített Bootstrap osztályok és reszponzív rács segítségével, hogy a közepes és annál nagyobb méretű kijelző esetén a három szolgáltatások-item osztályazonosítójú elem azonos szélességgel, egy sorban jelenjen meg!
7. A „*Szolgáltatások felugró kártyái*” megjegyzéssel jelzett szekcióban az első két szolgáltatás kártyára kattintás esetén felugró ablak HTML-kódját találja. Ennek mintájára hozzon létre még egy kártyát, melyet a forras mappában található szolgáltatás-felugro-tartalom.txt állományban található információkkal alakítson ki!
8. A rolunk-felirat azonosítójú címsort formázza a következő CSS tulajdonságok beállításával:
 - a. A betűméret 42 pixel legyen!
 - b. Állítson be egy 1 pixel széles, folytonos stílusú, fehér színű keretet!
 - c. A szöveg legyen középre igazítva!
 - d. A szöveg körül állítson be 10 pixel belső margót!
9. A rolunk azonosítójú szekcióban található szövegbekezdések közül az egyik megjelenése hibás, eltér a másik kettőtől. Keresse meg a hibát, és javítsa ki!
10. Helyezzen el hivatkozást a parkolohaz.js fájlra a weboldal végén!

A feladat a következő oldalon folytatódik.

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

11. A `parkolohaz.js` fájlban hozzon létre egy olyan JavaScript kódot, ami a weboldal kapcsolat szekciójában található információkat (név, e-mail-cím, telefonszám, üzenet) elküldi a webszervernek, ha az „Üzenet küldése” parancsgombra kattintunk!
- a. A REST-kérés paraméterei:
- Kérés típusa: POST
 - URL: `/api/kapcsolat`
 - Elküldött adat típusa: JSON
 - Elküldött adat szerkezete:

```
{
  name:String, (név, pl. „Kovács Andrea”)
  email:String, (e-mail-cím, pl. „kovacs.andrea@mail.com”)
  phone:String, (telefonszám, pl. „+36201234567”)
  message:String (üzenet, pl. „2 hónapra helyet szeretnék foglalni”)
}
```
 - Választípus: JSON
 - Válaszüzenet sikeres küldés esetén: `{feldolgozva:boolean}` (pl. `{feldolgozva:true}`)
- b. Amennyiben valamelyik mező értéke üres, akkor a „Kérjük, minden mezőt töltsön ki az űrlapon!” üzenetet jelenítse meg egy felugró ablakban, és ne küldje be az űrlap adatait a szervernek!
- c. Amennyiben a küldés sikeres (200-as státuszkód), törölje a weboldalon a beírt adatokat, és egy felugró ablakban jelenítse meg az alábbi üzenetet:
„Munkatársunk hamarosan keresni fogja önt!”

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

12. A világon 2020-ban átlagosan percenként 150 elektromos autót gyártottak le. Készítsen egy olyan JavaScript programkódot, amely a `samlalo` azonosítójú `` elembe 0-ról indulva fél másodpercenként egy 1-gyel növelt számot illeszt be!
- a. A számláló a weboldal betöltődésekor induljon el!
- b. A fél másodperces intervallumhoz használhatja a `setInterval()` függvényt:
`setInterval(functionName, interval);`
ahol:
- a `functionName` a meghívandó JavaScript függvény azonosítója,
 - az `interval` pedig a függvény ismételt meghívásának az időköze ezredmásodpercben (1000 ezredmásodperc = 1 másodperc).

A feladat a következő oldalon folytatódik.

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

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

ZSÁK UTCAI PARKOLÓHÁZ

SZOLGÁLTATÁSOKRÓLUNKKAPCSOLAT



PARKOLÓHÁZ A VÁROS SZÍVÉBEN

Vigyázzunk autójára!

SZOLGÁLTATÁSOK



RÓLUNK

A Zsák utcai parkolóház 1995-ben épült, kifejezetten az új városrész megőrzésére és az autós forgalmára adott válaszként.

Célunk, hogy könnyen megközelíthető, kényelmesen használható parkolóházat biztosítsunk Önnek, hogy mindig biztonságban érezhesse autóját!

Hosszú és rövid távra is bérelhet nálunk parkolóhelyet, legyen szó akár autóról, akár motorokról. Bővebb információért írjon nekünk az alábbi űrlap segítségével!

KAPCSOLAT

Kérjük, töltsse ki az alábbi jelentkezési lapot, és egy munkatársunk hamarosan felveszi Önnel a kapcsolatot.

Név

E-mail cím

Telefonszám

Mennyi időre és hány parkolóhelyet szeretne bérelni?

Üzenet küldése

EGYÉB ELÉRHETŐSÉGEINK:

TUDTA?



Amíg ezen a weboldalon tartózkodott, összesen 416 elektromos autót gyártottak a világban! Mi 15 elektromos töltőállomással várjuk elektromos autótól rendelkező ügyfeleinket.

A részletekről érdeklődjön a jelentkezési lapon!

Zsák utcai parkolóház ©2021

A feladat a következő oldalon folytatódik.

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

Adatbázis-kezelés

A parkolóház 2020-as évi adminisztrációját tartalmazza az adatbázisunk. Az Ön feladata az SQL lekérdezések elkészítése és beillesztése a vizsgakönyvtárban található 3_feladat\parkolohaz\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 kell másolnia. Az adatbázis-kezelési feladatok javítása során ennek az állománynak a tartalma lesz csak értékelve.

Az üzletvezetés számára is készült egy adminisztrációs oldal, amin keresztül a parkolóház bérletével kapcsolatos információkat és statisztikákat lehet megjeleníteni. Az oldalt a böngészőjében a <http://localhost:8080/admin> URL-en keresztül nyithatja meg. Az adatokat a rendszer adatbázisából kérdezi le a szerveralkalmazás, ami már fel van készítve arra, hogy a 3_feladat\parkolohaz\lekerdezések\lekerdezések.sql fájlból kiolvassa a megfelelő SQL utasítást, és ennek megfelelően szolgáltatson adatokat az adminisztrációs oldal számára. A megoldásait ennek megfelelően ellenőrizheti a böngészőjében a <http://localhost:8080/admin> URL megnyitásával is. Itt ellenőrizheti a szerver és az adatbázis elérhetőségét is.

A következő adatbázis-kezelési feladatok megoldására elkészített SQL parancsokat a lekerdezések.sql állományba illessze be a feladatok végén zárójelben jelölt elem értékeként! 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 törheti.

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

Az adatbázis a következő táblákat tartalmazza:

jarmutipus

id	Egész szám, a járműtípus azonosítója, PK
nev	Szöveg, a járműtípus megnevezése

parkolohely

id	Egész szám, a parkolóhely egyedi azonosítója, PK
jarmutipusId	A jármű típusának azonosítója, FK
emeletId	A parkolóhelyhez tartozó emelet azonosítója, FK
felirat	Szöveg, a parkolóhely felirata
szelesseg	Lebegőpontos szám (double), a parkolóhely szélessége méterben

emelet

id	Egész szám, az emelet azonosítója, PK
nev	Szöveg, az emelet megnevezése

foglalas

id	Egész szám, a foglalás egyedi azonosítója, PK
parkolohelyId	A foglaláshoz tartozó parkolóhely azonosítója, FK

A feladat a következő oldalon folytatódik.

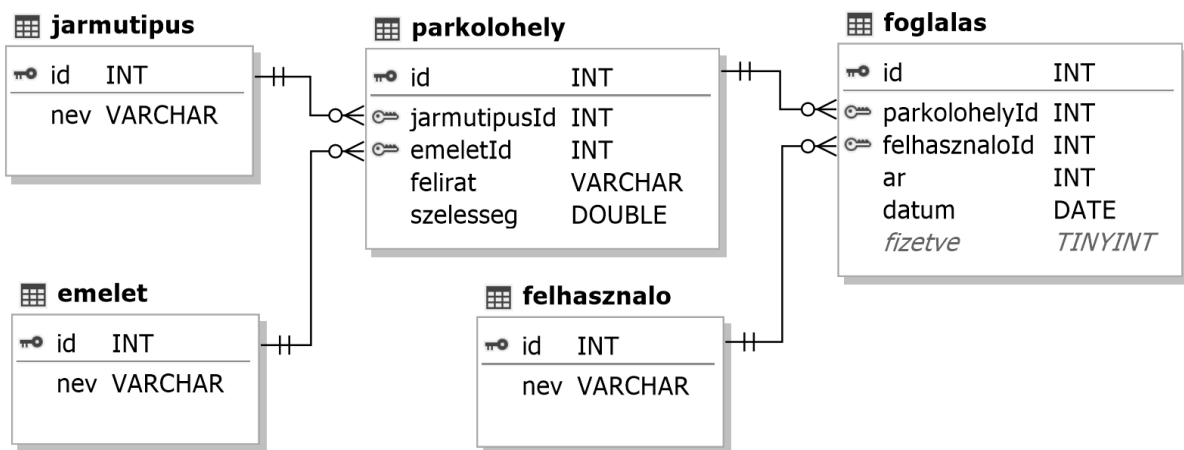
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

felhasznaloId	A foglaláshoz tartozó felhasználó azonosítója, FK
ar	A foglalásért fizetendő összeg
datum	Dátum/idő, a foglalás dátuma
fizetve	Logikai, a felhasználó kifizette-e már a foglalás árát.

felhasznalo

id	Egész szám, a felhasználó egyedi azonosítója, PK
nev	Szöveg, a felhasználó neve

Az elsődleges kulcsokat PK-val, az idegen kulcsokat FK-val jelöltük.



- Hozzon létre a lokális SQL serveren parkolohaz néven adatbázist! Az adatbázis alapértelmezett rendezési sorrendje a magyar szabályok szerinti legyen! Állítsa be az UTF-8 kódolást alapértelmezettnek az adatbázis létrehozásánál!
- A forras mappában található adatbazis.sql állomány tartalmazza a táblákat létrehozó és az adatokat a táblákba beszűrő SQL parancsokat! Futtassa az adatbazis.sql parancsfájlt a parkolohaz adatbázisban!
- Írja ki, összesen hány darab, pontosan 1,5 méter széles parkolóhely található a parkolóházban! A mező neve masfeles_parkolohelyek_szama legyen!
- Vegyen fel egy új járműtípust „Elektromos gépjármű” névvel, és módosítsa a „P001” feliratú parkolóhely járműtípusát is erre! A módosítást úgy készítse el, hogy az „Elektromos gépjármű” járműtípus azonosítóját az utasítás futtatásakor állapítsa meg!
- Kovács Anna vendégünk csak néhány napra foglalt magának helyet. Listázza ki a hozzá tartozó foglalásokat az adatbázisból! A lekérdezés során a datum és ar mezőket jelenítse meg! A lekérdezés során a felhasználó kiválasztását a „Kovács Anna” név felhasználásával tegye!
- Állapítsa meg, hogy melyik motorkerékpáros helyet bérlő felhasználó rendelkezik a legnagyobb tartozással! Lekérdezésében használja fel, hogy a motorkerékpár járműtípus neve a jarmutypus táblában „Motorkerékpár”! A lekérdezésben a nev és szumma_tartozas oszlopneveket használja! Az számít a rendszerben tartozásnak, ahol a foglalás fizetve értéke hamis. Megoldását úgy készítse el, hogy azonos nevű felhasználók esetén ne kerüljön összeadásra a tartozásuk! Feltételezheti, hogy nem alakult ki holtverseny a legtöbb tartozást felhalmozók között!

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

Az elkészült munka beadása:

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

- 3_feladat\parkolohaz\web\index.html
- 3_feladat\parkolohaz\web\parkolohaz.css
- 3_feladat\parkolohaz\web\parkolohaz.js
- 3_feladat\parkolohaz\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.	40	
Programozás	2.	40	
Weboldalak kódolása, adatbázis-kezelés	3.	40	
A gyakorlati vizsgarész pontszáma		120	

dátum

javító tanár

	pontszáma egész számra kerekítve	
	elért	programba beírt
Számítógépen megoldott gyakorlati feladatok		

dátum

dátum

javító tanár

jegyző