

INFORMATIKA ISMERETEK
EMELTSZINTŰ GYAKORLATI VIZSGA
MINTAFELADATOK

1. Hálózati ismeretek

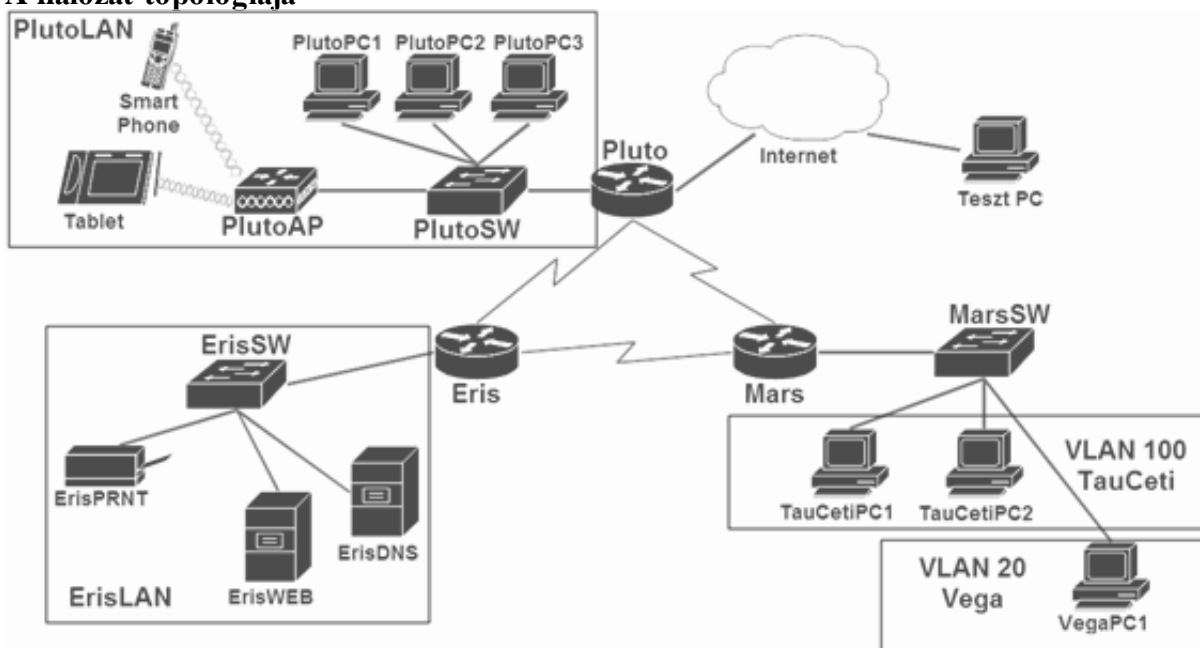
40 pont

Planet-NET

40 pont

Egy terjeszkedés alatt álló vállalat hálózatának tervezésével bízták meg. A vállalat jelenleg három telephellyel rendelkezik. Feladata, hogy a megadott tervek alapján szimulációs programmal elkészítse a vállalat teszhálózatát.

A hálózat topológiája



Beállítások

1. Töltse be a Planet_NET.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.
2. A Mars hálózatában a 10.20.30.0/24 privát címtartományt szeretnék használni. A két VLAN számára a lehető leghatékonyabban alakítson ki alhálózatokat a következő IP-cím igények figyelembevételével:

VLAN száma	VLAN neve	Igényelt IP-címek száma
100	TauCeti	100
20	Vega	20

Az ipcimzes.txt fájlban a példához hasonló módon rögzítse számolásának eredményét! Ha nem tudja elvégezni az alhálózatszámolást, akkor a továbbiakban a következő IP-címekkel dolgozzon:

VLAN száma	VLAN neve	Hálózat cím	Netmaszk
100	TauCeti	10.0.50.0	255.255.255.0
20	Vega	10.0.60.0	255.255.255.128

3. A Mars forgalomirányítón hozza létre a szükséges alinterfészeket úgy, hogy az alinterfész száma egyezzen meg a használt VLAN számával! Az alinterfészekre állítsa be a megfelelő hálózat első kiosztható IP-címét!
4. A VegaPC1-nek statikusan állítsa be a megfelelő hálózat utolsó kiosztható IP-címét, a megfelelő alapértelmezett átjárót és DNS kiszolgálót (ErisDNS)!

-
5. A `MarsSw` kapcsolón hozza létre a VLAN 20-at és állítsa be a `Vega` nevet! (A VLAN 100 már létezik.)
 6. A `MarsSW` kapcsoló megfelelő portjainak konfigurálásával érje el, hogy a kliensekhez csatlakozó portok hozzáférési portok legyenek és a megfelelő VLAN-ba kerüljenek!
 7. A `MarsSW` kapcsoló forgalomirányítóhoz csatlakozó portját állítsa be trónk módúra!
 8. A `TauCeti` VLAN számára hozzon létre egy DHCP hatókört a `Mars` forgalomirányítón:
 - a. Biztosítsa az összes szükséges paraméter átadását!
 - b. A DNS szolgáltatást az `ErisDNS` szerver biztosítja!
 - c. Az első 10 címet ne oszthassa ki a DHCP kiszolgáló!Állítsa be a `TauCeti` VLAN számítógépeit a dinamikus IP-cím használatához!
 9. Az IPv4-es forgalom irányításához OSPF protokollt használnak, melyet a `Pluto` és az `Eris` forgalomirányítón már konfiguráltak. Az Ön feladata a `Mars` forgalomirányítón az OSPF protokoll beállítása:
 - a. Használja a 10-es folyamatazonosítót!
 - b. Hirdesse az összes közvetlenül csatlakozó hálózatot a 0-ás területben!
 - c. A forgalomirányítási információk küldésére nem használt (a)interfészeket állítsa be passzívnak!
 10. A `Pluto` forgalomirányítón vegyen fel az Internethez vezető alapértelmezett statikus útvonalat! A megadásnál használja a kimenő interfészt! A `Pluto` forgalomirányítón futó OSPF folyamat kiegészítésével érje el, hogy a többi forgalomirányító is megtanulja az alapértelmezett útvonalat!
 11. A `PlutoAP` vezeték nélküli eszközön állítsa át az SSID értékét **PLUTO**-ra és állítson be WPA2 hitelesítést **plutowifi** jelszóval!
 12. Csatlakoztassa a vezeték nélküli klienseket (SmartPhone, Tablet) a `PlutoAP` eszközhöz!
 13. A `plutoi` kliensek a `Pluto` forgalomirányítón futó DHCP szolgáltatástól kapnának IP-címet, de a szolgáltatás konfigurációja jelenleg hibás. Hárítsa el a DHCP szolgáltatás konfigurációjában található hibát, és kérjen IP-címet a klienseknek!
 14. Az `Eris` forgalomirányítón a privilegizált módot védő jelszó az **erisspass** legyen!
 15. Az `Eris` forgalomirányító első 5 virtuális vonalán állítsa be, hogy távolról csak SSH protokollal lehessen elérni az eszközt! Használjon helyi hitelesítést a vonalakon! A szükséges felhasználó neve **erisadmin**, jelszava **erisspass** legyen! Állítsa be, hogy az eszköz domain neve `eris.net` legyen! Engedélyezze az SSH 2-es verzióját! Használjon hozzá 1024 bites kulcsot!
 16. Normál hozzáférési lista használatával érje el, hogy az első 5 virtuális vonalon keresztül csak az `Eris` hálózatából (192.168.50.0/26) lehessen elérni az eszközt!
 17. A `Pluto` forgalomirányítón állítson be statikus NAT szolgáltatást, amellyel biztosítja, hogy az `ErisWEB` szerver kívülről (a `Teszt PC`-ről) a 22.33.44.3 IP-címmel legyen elérhető!
 18. A `Pluto` forgalomirányítón állítson be PAT szolgáltatást, amely a forgalomirányító internet felőli interfészének címére fordítja a teljes belső hálózatból az internet felé menő csomagok IP-címét!
 19. Az `Eris`-en található szerverek számára IPv6-os elérést is kell biztosítani. A `Pluto` forgalomirányítón már beállításra kerültek a megfelelő IPv6-címek. Az `Eris` forgalomirányítón állítson be IPv6-címeket:
 - a. `Serial0/0/1` interfészének globális IPv6 cím: 2040:40::2/64
-

b. GigbitEthernet0/1 interfészének globális IPv6 cím: 2050:50::1/64, link-local cím: FE80::1

20. A Pluto eszközön vegyen fel alapértelmezett IPv6 útvonalat az Internet irányába a 2030:30::1 következő ugrás cím használatával!
21. A forgalomirányítók közé az IPv6 forgalom irányítására konfiguráljon RIPng protokollt RNG névvel! Érje el, hogy az Eris forgalomirányító megtanulja az alapértelmezett útvonalat a szomszédjától!
22. Az ErisWEB szervernek állítsa be a 2050:50::50/64 IPv6-címet! Alapértelmezett átjárója a forgalomirányító link-local címe legyen!
23. Az ErisPRNT nyomtató és az ErisDNS szerver SLAAC segítségével jussanak IPv6-címhez!

Hálózat működésének tesztelése:

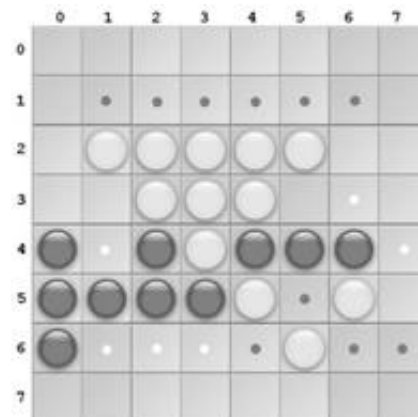
- A forgalomirányítás működik a Mars forgalomirányítón, a routing táblában jelennek meg OSPF-től tanult bejegyzések.
- A TauCeti VLAN és a Vega VLAN kliens gépeiről elérhető a www.net.hu (7.7.7.70) és közben a PAT működik.
- A TESZT PC-ről elérhető a www.eris.web (22.33.44.3) és közben a statikus NAT működik.
- Az Eris forgalomirányító az Eris hálózatából elérhető SSH kapcsolattal.

2. Programozás**40 pont**

TUDNIVALÓK ÉS JAVASLATOK FELKÉSZÍTŐ, ILLETVE JAVÍTÓ TANÁROK SZÁMÁRA

Ebben a feladatban a vizsgázónak konzolos és/vagy grafikus alkalmazást kell készítenie a tanult programozási nyelv (Java vagy C#) és fejlesztői környezet felhasználásával.

A reversi játékot általában 8×8 mezőből álló négyzetácsos táblán játsszák. Ebben a feladatban a tábla sorait és oszlopa it is **0-tól 7-ig** azonosítjuk az ábra szerint. A játékot legjobb olyan korongokkal játszani, amelyek két oldala különböző színű (feladatunkban kék és fehér). A két játékos felváltva rakja le korongjait. A soron következő játékos csak olyan helyre rakhat, ahol meg tudja **fordítani** az ellenfél legalább egy korongját. Ez úgy lehetséges, hogy az éppen letett korong és a játékos másik korongja között egyenes vonalban vízszintesen, függőlegesen vagy átlósan kizárólag csak az ellenfél egy vagy több korongja található. Az egyes játékosok következő lehetséges lépéseit az ábrában kisebb körökkel szemléltetjük. Például a kék (sötét) játékos az 1;5 (sor;oszlop) mezőre azért rakhat szabályosan, mert a 4;2 mezőn lévő korongjával közrefog átlósan 2 db fehér korongot, így azok megfordulnak. A fehér játékos azért nem rakhat a 3;5 mezőre, mert nincs olyan korongja, amivel kizárólag csak kék (sötét) korongokat fogna közre, így erre a mezőre lépve nem tudna fordítani. Az ábra a forrás állomány adataival készült, így tanulmányozása a megoldáshoz segítséget nyújthat. A továbbiakban a reversi játékkal kapcsolatos feladatokat kell megoldania.



A megoldás során vegye figyelembe a következőket:

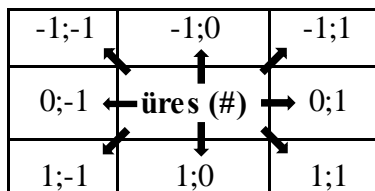
- *A program készítése során törekedjen az objektum orientált (OOP) megoldásra, amire a feladatsor ajánlásokat is tartalmaz. Amennyiben a programot ilyen módon nem tudja elkészíteni, akkor a feladatokat saját osztály létrehozása nélkül is megoldhatja, de így kevesebb pontot ér a megoldása. Ebben az esetben, ha a feladat jellemző vagy metódus létrehozását kéri, akkor Önnek saját alprogramot (függvényt, eljárást) kell készítenie, amely **paramétereken keresztül** kommunikál a hívó programmal!*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges input adatok mellett is megfelelően működjön!*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti!*

1. Készítsen **grafikus** felhasználói felületű programot a következő feladatok megoldására, amelynek a forráskódját ReversiGUI néven mentse el!
2. Hozzon létre saját osztályt Tabla azonosítóval és definiáljon benne egy karakter típusú mátrixot (kétdimenziós tömböt) Allas azonosítóval, melyben egy játék pillanatnyi állását tudja majd tárolni! A mátrix sorai és oszlopai 0-tól 7-ig legyenek indexelve!
3. Készítse el az osztály konstruktorát, ami a következő feladatokat hajtja végre:
 - a. Inicializálja az Allas mátrixot 8×8-as mérettel!
 - b. Feltölti az Allas mátrixot a „#”, „K” és „F” karakterekkel egy szöveges állományból. A feldolgozandó szöveges fájl nevét a konstruktor paramétereként adjuk át! A feladat megoldásához használandó allas.txt állomány 8 sora, soronként 8 karakterrel tárolja egy játék állását. A tábla üres mezőit a „#” karakter, a játékosok korongjait a „K” (kék) és „F” (fehér) karakterek kódolják.
 - c. A konstruktort tetszőlegesen tovább bővítheti a későbbi feladatok megoldásához!
4. Hozzon létre egy Tabla típusú osztálypéldányt (objektumot), melynek a konstruktora az allas.txt forrás állomány nevét kapja aktuális paraméterként feldolgozásra!

5. Készítsen a grafikus tervező használatával és/vagy a `Megjelenit` metódus kódolásával az **1. minta** szerint grafikus felhasználói felületet, ami megjeleníti az `Allas` mátrixban eltárolt játék állását! A `Megjelenit` metódust a `Tabla` osztályban helyezze el! Színes mintákat a program futásáról a `minta*.jpg` állományokban talál! A grafikus felület kialakításához úgy válasszon osztályokat, hogy a 8-10. feladatok megoldhatók legyenek!
6. Definiáljon a `Tabla` osztályban metódust a következő algoritmus kódolásával! (Ha nem a `Tabla` osztályban kódolja a metódust, akkor az `Allas` mátrix is a függvény paramétere legyen!)

```
Függvény VanForditas(jatekos: Karakter, sor, oszlop,
                    iranySor, iranyOszlop: Egész): Logikai
    Változó aktSor, aktOszlop: Egész
    Változó ellenfel: Karakter
    Változó nincsEllenfel: Logikai
    aktSor:=sor + iranySor
    aktOszlop:=oszlop + iranyOszlop
    ellenfel:='K'
    Ha (jatekos='K') akkor
        ellenfel:='F'
    Elágazás vége
    nincsEllenfel:=igaz
    Ciklus amíg (aktSor>0 és aktSor<8 és
    aktOszlop>0 és aktOszlop<8 és t[aktSor, aktOszlop]=ellenfel)
        aktSor:=aktSor + iranySor
        aktOszlop:=aktOszlop + iranyOszlop
        nincsEllenfel:=hamis
    Ciklus vége
    Ha (nincsEllenfel vagy aktSor<0 vagy aktSor>7 vagy
    aktOszlop<0 vagy aktOszlop>7 vagy
    t[aktSor, aktOszlop]<>jatekos) akkor
        Térj vissza hamis
    Elágazás vége
    Térj vissza igaz
    Függvény vége
```

A metódus a megadott játékos, megadott lépését, megadott irányban határozza meg, hogy az adott irányban történhet-e fordítás. Az `iranySor` és `iranyOszlop` paraméterek a következők szerint határozzák meg a feltételezett fordítás irányát:

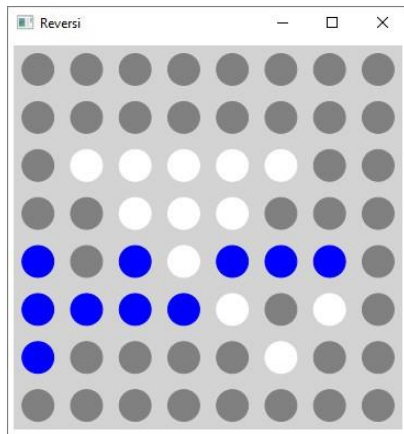


7. Készítsen a `Tabla` osztályban logikai értékkel visszatérő metódust `SzabalyosLepes()` azonosítóval, ami meghatározza egy megadott játékos megadott lépéséről, hogy szabályos lépés vagy nem szabályos lépés! Szabályosnak tekintünk egy lépést, ha a megadott cella üres és a nyolc irány valamelyikéből (lásd előző feladat) a megadott játékosal történhet fordítás. Megoldásában használja fel a korábban elkészített `VanForditas()` metódust is!

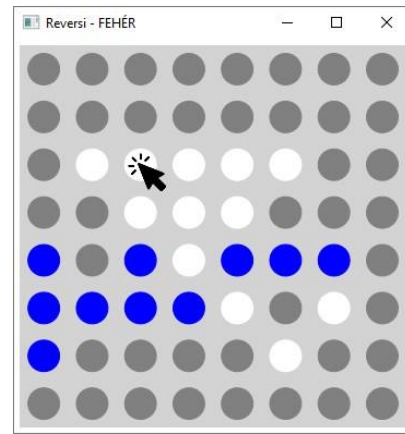
8. A program indulásakor a kék vagy fehér korongokra kattintva lehessen kiválasztani a játékban következő játékost! Az így kiválasztott játékos „színe” kerüljön az ablak címsorába a **2. mintának** megfelelően! Ha indulás után üres (szürke) mezőre kattintunk, akkor ne történjen semmi!
9. Miután sikeresen kiválasztottunk egy játékost az előző feladatban leírtak szerint, határozzuk meg egy üres (szürke) mezőre kattintva, hogy az előzőleg kiválasztott játékos lépése szabályos-e erre a mezőre! Ezt az ablak címsorában jelezzük a **3. és 4. mintának** megfelelően! Megoldáshoz használja fel a SzabalyosLepes() metódust!
10. Legyen lehetőség minden üres cella ellenőrzésére és új játékos (szín) választására a fenti módszerek megismétlésével!

Minták:

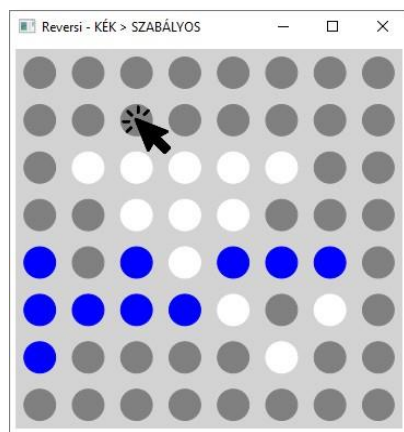
1. minta



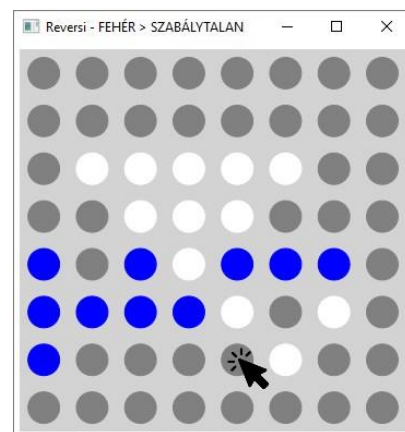
2. minta



3. minta



4. minta



Forrás:

<http://mek.oszk.hu/00000/00056/html/138.htm> (2017.06.10)

3. Csudijó étterem

40 pont

A következő feladatban egy étterem komplex webalkalmazásának fejlesztésében kell részfeladatokat elvégeznie. Az Ön feladata két részből áll:

- a nyilvánosságnak szánt weboldal véglegesítése („Weboldalak kódolása” feladatrész);
- adatbázis lekérdezések elkészítése az üzletvezetés által használt adminisztrációs oldal számára („Adatbázis-kezelés” feladatrész).

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

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 forrás állományok között található csudijo.zip tömörített archív állományában.

Figyelmeztetés: a teljes archívum kitömörítése több percig is eltarthat!

A webszerver indításához a Windows parancssorában tegye aktuálissá a vizsgakönyvtár 3_feladat\csudijo nevű mappáját, majd futtassa az `npm run start` parancsot.

Weboldalak kódolása rész

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

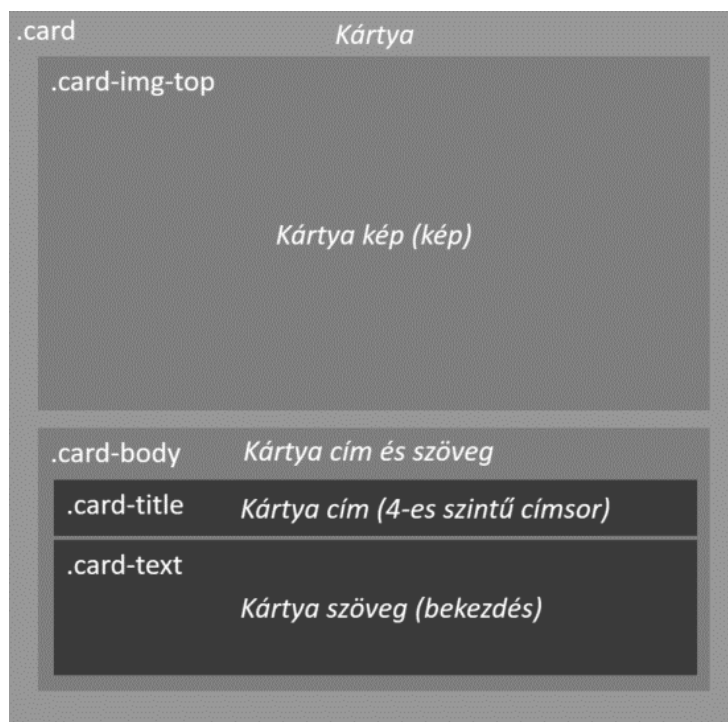
A munkája végén ezeket az állományokat a feladatlapon „Az elkészült munka” beadása részben megadottak szerint a munkakönyvtárának beadott mappájába kell másolnia!

A formázási beállításokat a csudijo.css stílusállományban végezze el, lehetőleg úgy, hogy az új szelektorok létrehozása a stílusállomány végén történjen!

1. Nyissa meg az index.html fájlt! Állítsa be, hogy a böngésző címsorában megjelenő cím „Csudijó Étterem” legyen!
 2. Helyezzen el hivatkozást a csudijo.css stíluslapra! Ügyeljen rá, hogy a csudijo.css stíluslap formázásai felülírják a bootstrap.min.css formázási beállításait.
 3. Az egész dokumentumra vonatkozóan állítsa be, hogy a sormagasság 1,8 értékű legyen.
 4. Alakítsa ki a weboldal navigációs részét az alábbiak szerint:
 - a. A „Navigáció” szekcióban hozza létre az alábbi menüelemeket: „Rólunk”, „Étlap”, „Legnépszerűbb”, „Vendégkönyv”, „Kapcsolat”. Ehhez a „Menüelemek” szöveget cserélje ki egy ötelemű 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 nav-link osztályba sorolt hivatkozásokat, melyek sorrendben a rolunk, etlap, legnepyszerubb, vendegkonyv, kapcsolat horgonyra mutatnak!
 5. A fejléc azonosítókijelölő bővítésével módosítsa a fejléc megjelenését! Állítsa be háttérképpnek a kepek/etterem.jpg képet! A háttérkép ismétlődés nélkül, teljes
-

mértékben fedje le az elemet úgy, hogy a kép teljes egészében látszódjon! A szöveg fehér színnel jelenjen meg!

6. A „Fejlécszöveg” feliratot cserélje ki az 1-es szintű címsorban szereplő „Csudijó Étterem” címre, és az ezt követő „Ahol az ízek életre kelnek” szövegű motto osztályba sorolt bekezdésre! A motto azonosítókijelölő létrehozásával formázza a bekezdést úgy, hogy a betűmérete legyen fele az egyes szintű címsorhoz beállított betűméretnek!
7. Szűrja be a kepek/csicseri.jpg képet a „Rólunk” szekcióban a bekezdés alá! Ha a kép fölé visszük az egeret, vagy a kép valamiért nem jeleníthető meg, akkor mindkét esetben a „Csicseri patak” szöveg jelenjen meg! Formázza úgy a képet a csudijó.css fájlban létrehozott új bejegyzéssel, hogy maximális szélessége ne haladja meg a szülő elem szélességét, de ez a beállítás csak a rolunk tagkijelölőjú szekcióban található képekre vonatkozzon!
8. Állítsa be a dokumentum bekezdéseiben a betűvastagságot! Állapítsa meg böngészőjének fejlesztői eszköze segítségével, hogy mennyi az aktuális vastagság, és állítson be ennél 100 egységgel kisebb értéket új elemszelektor létrehozásával!
9. Az „Étlap” szekcióban a kettes címsor alatt szűrjon be az ábrának megfelelő szerkezetű négy darab kártyát! Ahol nincs külön jelölve a HTML-elem típusa, ott használjon keretet (<div>)! Formázza az egyes elemeket az ábrán megadott Bootstrap osztálykijelölőkkel! A kártyákat a kartya.txt állományban megadott tartalommal alakítsa ki!
10. A kártyák számára hozzon létre reszponzív viselkedést biztosító rácsot! Közepes és nagy kijelző méret esetén két sorban, soronként két-két kártyát, míg kicsi és extra kicsi kijelzőjú mobil eszközök esetén négy sorban, soronként egy-egy kártyát jelenítsen meg! A megjelenítések közötti váltás a Bootstrap 4 rács alapértelmezett 768 px-es töréspontján történjen.



11. Helyezzen el hivatkozást a csudijó.js JavaScript fájlra weboldal végén!
12. A csudijó.js fájlban hozzon létre olyan JavaScript kódot, ami a szerverről Ajax kéréssel lekérdezi, hogy melyik étel volt a legnépszerűbb az elmúlt évben, és a válaszüzenetben kapott étel nevét betölti a legnépszerűbb tagkijelölővel rendelkező bekezdésbe! A megfelelő szerver oldali végpont az alábbi jellemzőkkel bír:

-
- Kérés típusa: GET
 - URL: /api/legnepszerubb
 - Választípus: JSON
 - Válasz adat: {etelNev}

Amennyiben nem tudja elvégezni az adatlekérést a szerverről, a „LECSÓ KOLBÁSZCSIPSSZEL” szöveget töltsse be a HTML oldal betöltődését követően JavaScript kód segítségével a legnepszerubb tagkijelölővel rendelkező bekezdésbe.

13. Írjon olyan kódot, ami a weboldal „Vendégkönyv” szekciójában beírt bejegyzést elküldi a szervernek.

- Kérés típusa: POST
- URL: /api/vendegkonyv
- Elküldött adat típusa: JSON
- Elküldött adat: {bejegyzes}
- Választípus: JSON
- Válasz adat: {bejegyzes}

Amennyiben a küldés sikeres, törölje a weboldalon a bejegyzést és egy felugró ablakban jelenítse meg az alábbi üzenetet: „Köszönjük a bejegyzését!”.

Adatbázis-kezelés rész

Az adatbázisunk az étteremben rendelhető termékeket és a 2019-es év februári rendeléseit tartalmazza. Az Ön feladata az SQL lekérdezések elkészítése és beillesztése a munkakönyvtárájában található

3_feladat/csudijo/lekerdezések/lekerdezések.sql fájlba. A munkája végén ezt az állományt a feladatlap „Az elkészült munka” beadása részében megadottak szerint a beadott mappába 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 az étteremmel kapcsolatos információkat és statisztikákat lehet megjeleníteni. Az oldalt a böngészőjében a <http://localhost:8000/admin> URL-en keresztül nyithatja meg.

Az adatokat az éttermi rendszer adatbázisából kérdezi le a szerveralkalmazás, ami már fel van készítve, hogy a csudijo/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:8000/admin> URL megnyitásával is.

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

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

Az adatbázis a következő táblákat tartalmazza:

termek

id	Egész szám, az étlapon vagy itallapon szereplő termék sorszáma, PK
nev	Szöveg, a termék megnevezése
egysegar	Egész szám, a termék bruttó eladási ára
etel	Egész szám:
kategoriaId	A termék kategóriájának azonosítója, FK

kategoriak

id	Egész szám, a kategória azonosítója, PK
nev	Egész szám, a kategória megnevezése

rendelesek

id	Egész szám, a rendelési tétel azonosítója, PK
mennyiseg	Decimális érték
afakulcs	Decimális érték, az ÁFA-kulcs százaléktétele
datum	Dátum/idő, a rendelés dátuma

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



-
14. Hozzon létre a lokális SQL szerveren `csudijo` néven adatbázist! Az adatbázis alapértelmezett rendezési sorrendje a magyar szabályok szerinti legyen! Állítsa be az UTF-8 kódolást alapértelmezettnek az adatbázis létrehozásánál! (**14. feladat:**)
15. Az adatbázis mappában található `adatbázis.sql` állomány tartalmazza a táblákat létrehozó és az adatokat a táblába beszűrő SQL parancsokat! Futtassa az `adatbázis.sql` parancsfájlt a `csudijo` adatbázisban!
16. A `termek` tábla `etel` mezőjében 1-es érték jelzi, ha az adott tétel étel típusú. Írassuk ki, hány étel szerepel az étlapon! A számított mező neve `etelek-szama` legyen! (**16. feladat:**)
17. Listázza ki az összes desszert kategóriába tartozó termék nevét és árát! Rendezze a desszerteket névsorba! A lekérdezésben használja az alábbi kategórianevet: „Desszertek” (**17. feladat:**)
18. Adjon hozzá a termékekhez egy új desszertet: (**18. feladat:**)
- | | |
|--------------|----------------------|
| nev: | Málnahabos pohárkrém |
| egysegar: | 1090 |
| kategoriaId: | 6 |
| etel: | 1 |
19. Írassa ki a februárban eladott mennyiség alapján a három legnépszerűbb étel nevét és az eladott mennyiséget! A számított mező neve `mennyisege` legyen! (**19. feladat:**)
20. 2019-ben a helyben készülő és helyben is fogyasztott ételek ÁFA-kulcsa 5%, míg az elvitelre kért ételek, valamint az italok ÁFA-kulcsa 27%. A rendelések táblában minden tétel mellett fel van tüntetve a **bruttó** (ÁFA-val növelt) eladási ár, illetve, hogy mekkora ÁFA-kulccsal történt az eladás (5%-os ÁFA-kulcs esetén 0,05, 27%-os ÁFA-nál pedig 0,27).
Határozza meg a teljes **bruttó** árbevételt és a teljes bevétel ÁFA tartalmát! A számított mezők neve `brutto` és `afa` legyen! Az ÁFA tartalom meghatározásához használja az alábbi segédletet:
- $$\text{Nettó_ár} = \text{Bruttó_ár} / (1 + \text{ÁFA_kulcs})$$
- Például 1270 Ft bruttó ár és 27% ÁFA-kulcs esetén a nettó ár:
- $$1270 \text{ Ft} / (1 + 0,27) = 1000 \text{ Ft}$$
- $$\text{ÁFA_tartalom} = \text{Bruttó_ár} * (1 - 1 / (1 + \text{ÁFA_kulcs}))$$
- Kerítse a számított értékeket egész számra a `ROUND` függvény segítségével az alábbi szintaxist használva: `ROUND(érték, 0)` (**20. feladat:**)
-

Csudijó Étterem - Adminisztrációs oldal

16. Ételek száma az étlapon	
etelek-szama	
50	

17. Desszertek	
nev	ar
Csokoládé szuflé	990
Málnahabos pohárkrém	1090
Palacsinta	300
Túrógombóc háziasan édes tejföllel, áfonyával	1390

19. A három legnépszerűbb étel februárban	
nev	menyiseg
Bögrében sült paradicsomos tészta	207
Lecsó kolbászcipsszel	201
Palacsinta	118

20. Nettó árbevétel és befizetendő ÁFA	
brutto	afa
4291897	481225

99. Vendégkönyv bejegyzések	
bejegyzes	datum
Remek hely, kiváló vedéglátás, mindenkinek ajánlom!	2019-05-04T22:55:02.000Z

Az elkészült munka beadása

21. Munkájának befejezése után másolja az alábbi állományokat a munkakönyvtárának 3_feladat\beadott mappájába!

- csudijo\web\index.html
- csudijo\web\csudijo.css
- csudijo\web\csudijo.js
- csudijo\lekerdezések\lekerdezések.sql

Munkája 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.

**INFORMATIKA ISMERETEK
EMELTSZINTŰ GYAKORLATI VIZSGA**

**TUDNIVALÓK ÉS JAVASLATOK FELKÉSZÍTŐ, ILLETVE
JAVÍTÓ TANÁROK SZÁMÁRA**

A vizsgázóknak egy komplex webalkalmazásban kell feladatokat elvégezniük. Ennek elsősorban szemléletformáló célja van. A feladatok mindegyike megoldható az érettségi követelményrendszerben előírt weblapszerkesztési, JavaScript és adatbázis-kezelési ismeretekkel. A szerveroldali alkalmazást, valamint az adminisztrációs weblapot nem kell a vizsgázóknak módosítaniuk.

- a. A vizsgázó a maximális pontszámot még abban a szélsőséges esetben is meg tudja szerezni, ha a szerveroldali alkalmazás nem fut. Az Ajax kérések teszteléséhez természetesen szükség van a működő szerverre.
 - b. „Weboldalak kódolása” rész
 - Ebben a részben a vizsgázóknak Bootstrap 4 keretrendszer segítségével már előzetesen elkészített weblapot kell véglegesíteniük. Ennek során HTML szerkesztési feladatokat, meglévő Bootstrap 4 stílusok alkalmazását, valamint saját stílusok létrehozását és alkalmazását kell elvégezniük.
 - A vizsgázóknak ügyelnie kell rá, hogy a szerkesztés során ne sérüljön a weboldal responzív működése.
 - c. „Adatbázis-kezelés” rész
 - Az SQL lekérdezéseket a vizsgázók a korábbi érettségi feladatoknak megfelelően elkészíthetik és tesztelhetik pl. phpMyAdmin segítségével.
 - Az elkészült lekérdezéseket a `lekerdezések.sql` fájlba kell beilleszteniük a megfelelő feladat sorszáma alá. A javítás során csak ennek a fájlnak a tartalma lesz értékelve. A szerver oldali alkalmazás fel van készítve, hogy ebből a fájlból kiolvassa a megfelelő lekérdezéseket, és ennek megfelelő adatokat szolgáltatson az adminisztrációs weboldal számára. A vizsgázóknak vigyáznia kell, hogy ne módosítsák a fájl belső szerkezetét (pl. ne töröljék az elválasztó „***” karaktereket), mert ellenkező esetben a szerver nem tudja feldolgozni a lekérdezéseket. Ebben az esetben az adminisztrációs oldalon hibaüzenet jelenik meg. Mindez nem befolyásolja a vizsgázó munkájának értékelését.
-

**INFORMATIKA ISMERETEK
EMELTSZINTŰ GYAKORLATI VIZSGA**

**KIEGÉSZÍTŐ RENDSZERGAZDAI ÚTMUTATÓ
A MINTAFELEADATSOR WEBOLDALAK KÓDOLÁSA ÉS
ADATBÁZIS-KEZELÉS
FELADATRÉSZÉHEZ**

A weboldalak kódolása és adatbázis-kezelés részben egy komplex webalkalmazás fejlesztésében kell a vizsgázóknak feladatokat elvégezniük. A készen kapott Node.js alapú szerveroldali alkalmazás MySQL adatbázisból fog dinamikusan adatokat szolgáltatni, emellett a statikus webtartalmat (a vizsgázó által módosítandó nyitóoldalt és a szintén készen kapott adminisztrációs oldalt) is ez a rendszer szolgálja ki.

A vizsgakörnyezet előkészítése:

- a. Az adathordozó Forrás mappáját tartalmával együtt másolja a vizsgázó munkakönyvtárába.
- b. A szoftverlista előírásainak megfelelően előzetesen már telepítenie kellett a Node.js környezetet. Ellenőrizze a Windows parancssorában kiadott `node -v` parancs segítségével, hogy a szoftverlistának megfelelő verziójú Node.js van-e telepítve a gépen.
- c. A vizsgakönyvtár Forrás\3. Csudijo\web mappában levő csudijo.zip fájlt csomagolja ki a vizsgakönyvtár 3_feladat könyvtárában. A kicsomagolás után a szerveroldali alkalmazás állományai a munkakönyvtár 3_feladat\csudijo mappájában lesznek elérhetők.
- d. A teszteléshez a Windows parancssorában tegye aktuálissá a vizsgakönyvtár 3_feladat\csudijo mappáját, majd az `npm run start` parancs segítségével indítsa el a szerveralkalmazást. A `http://localhost:8000` URL segítségével tesztelheti a szerveralkalmazás működését.
- e. Indítsa el a XAMPP csomagban található Apache és a MySQL komponenseket, és a `http://localhost/phpmyadmin` URL segítségével ellenőrizze a phpMyAdmin működését is. Ellenőrizze azt is, hogy a MySQL szerver alapértelmezett root felhasználójának jelszó nélküli elérése is működik. Ehhez a Windows parancssorában tegye aktuálissá a XAMPP csomag `mysql/bin` mappáját, majd adja ki a `mysql -u root` parancsot.
- f. Tesztelje az adminisztrációs oldal elérését is a `http://localhost:8000/admin` URL segítségével.
- g. Hozzon létre egy beadott nevű mappát a vizsgakönyvtár 3_feladat mappájában.

A weboldalak kódolása és adatbázis-kezelés részben a vizsgázó munkája a beadott mappában található fájlok alapján lesz értékelve, így a vizsga befejezését követően menteni csak ezt a mappát kell.
