

Gépkocsi kölcsönzés nyilvántartás

Készítsünk alkalmazást, amely gépkocsik kölcsönzését tartja nyilván.

1. Hozzunk létre egy adatbázist, amely két táblát tartalmaz:

Cars

id:	szám, automatikus számozás, elsődleges kulcs
producer:	szöveg(20), az autó gyártója, pl. Opel, Suzuki
type:	szöveg(30), az autó típusa, pl. Corsa, Swift
vintage:	szám, az autó forgalomba helyezésének éve
fuel:	szöveg(10), az üzemanyag típusa, pl. benzin, dízel

Tenants

id:	szám, automatikus számozás, elsődleges kulcs
name:	szöveg(80), a bérlő neve
address:	szöveg(200), a bérlő lakcíme (irsz,város,utca,házzámezgyben)
regCostumer:	tinynt, 0 vagy 1 attól függően, hogy törzsvásárló vagy sem

A táblákat töltsük fel 4-5 teszt rekorddal értelemszerűen!

2. Hozzunk létre egy header állományt, mely az alábbi menüpontokat tartalmazza:

- Home - valamilyen üdvözlő szöveggel fogadja az új látogatókat
- Cars - összes autó listája
- All tenants - összes bérlő listája
- Regular costumers - törzsvásárlók listája

A header állomány minden esetben jelenjen meg az oldalon!

3. Hozzuk létre a `_layout.php`-t és a menüpontokhoz tartozó feldolgozó és megjelenítő oldalakat. Csak a `_layout.php` tartalmazza a teljes *HTML* fájlt! Az alábbi view-k nem írhatnak header állományokat!

- A) **homeView.php**: Adjon egy üdvözlő szöveget az új látogatóknak!
- B) **cars.php**: Adjon egy listát az összes autó összes adatáról!
- C) **carsView.php**: Táblázatos formában jeleníti meg a kapott adatokat! Egy legördülő listába gyűjtsük ki a gyártmányokat, és kiválasztva egyet, csak a feltételnek megfelelő autókat listázza! *Legyen lehetőség a módosításra, törlésre!* Ha a felhasználó a módosít gombra kattint, az oldal dobjon át a `modifi.php`-ra, a kiválasztott autó azonosítójával együtt.
- D) **modifyCar.php**: a kiválasztott autó adatait állítja elő.
- E) **modifyCarsView.php**: A kiválasztott autó új adatainak feltöltését szolgáló űrlap. Alapértelmezetten az autó adatait jeleníti meg.
- F) **allTenant.php**: Adjon egy listát az összes bérlő összes adatáról.
- G) **allTenantView.php**: A kapott adatokat táblázatos formában jeleníti meg. Legyen lehetőség a módosításra! Módosítás gombra kattintva dobjon át a `modifyTenant.php`-ra.
- H) **modifyTenant.php**: lekéri a kiválasztott bérlő adatait és átadja a view-nak.
Ha elküldésre kerül a `modifyTenantView.php` űrlap, update-eli az adatbázist, majd visszairányít a `Modify Tenant` oldalra.
- I) **modifyTenantView.php**: A kiválasztott bérlő adatait jeleníti meg alapértelmezetten! Submit után az adatokat a **modifyTenant.php** fájl dolgozza fel a loopolást elkerülendő.
- J) **regularCostumers.php**: A törzsvásárlók nevét és címét adja át a view-nak, név szerint növekvő rendben.
- K) **regularCostumersView.php**: Csak a törzsvásárlók nevét és címét jeleníti meg.

A kékkel jelölt oldalak nem elérhetőek közvetlenül a menüsávból!

A feldolgozó fájlok csak adatokat kezelnek, írnak-olvasnak az adatbázisba, a megjelenítő oldalak csak megjelenítik az átadott adatokat html-be ágyazva. Itt már nincsen logika/feldolgozás.

A view-k esetében kötelező az az alapelve követése, hogy csak azt írjuk ki php-val, ami dinamikus!

Pl.: Ez hibás:

```
<?php
if($akarmi)
{
    echo '<table>
        <tr>
            <td>column</td>
            <td>{data}</td>
        </tr></table>';
}
?>
```

Ez jó:

```
<?php if($akarmi): ?>
  <table>
    <tr>
      <td>column</td>
      <td><?=$data?></td>
    </tr>
  </table>
<?php endif ?>
```

Az elkészült program legyen esztétikus, átlátható elrendezésű!