

# Buszjáratok és vizsgálataik

## Gyakorló feladat

Ön egy buszjáratokat és a kötelező ellenőrzéseikre vonatkozó adatokat nyilvántartó webes alkalmazás egyes moduljainak elkészítésére kapott megbízást.

Feladata, hogy a program meghatározott moduljait megvalósítsa adatbázis támogatással.

**Tábla: busz**

<b>id</b>	szám	a buszjárat azonosítója, elsődleges kulcs
indulo	szöveg(30)	az induló állomás
cel	szöveg(30)	a célállomás
menetido	szám	A járat menetideje percben
alacsony	szám	Alacsonypadlós vagy nem (0 vagy 1)

**Tábla: szerviz**

<b>id</b>	szám	a felülvizsgálat azonosítója, elsődleges kulcs, <b>auto_inc</b>
<i>buszID</i>	szám	a busz azonosítója, <i>idegen kulcs</i>
datum	dátum	a felülvizsgálat ideje
megjegyzés	szöveg(100)	a vizsgálat eredménye
engedelyezve	szám	további futás engedélyezése: 0 vagy 1

**Néhány tesztadat az adatbázis használatához:**

id	indulo	cel	menetido	alacsony
32	Örs vezér tere	Árpád híd	42	1
276	Örs vezér tere	Cinkota garázs	32	1
7	Rákosporzasztó	Etele tér	62	0

id	buszID	datum	megjegyzes	engedelyezve
1	1	2018-01-12	minden rendben	1
2	2	2018-01-12	lengőkar hiba	0
3	3	2018-01-12	minden rendben	1
4	1	2018-02-13	ajtók nem záródnak	0
5	2	2018-03-14	minden rendben	1

## Feladata:

Készítsen programot, mely csatlakozik az adatbázishoz és a lentebb részletezett funkciókat valósítja meg!  
Megjegyzés: a busz tábla azonosítóját is **auto\_increment**-re állítsuk.

### 1. Adatbázis létrehozása

Csatlakozzon az adatbázis-kiszolgálóhoz saját azonosítójával és készítse el az adatbázisban a **busz** és a **szerviz** táblákat!

### 2. A program létrehozása

Használja a forrásban kapott alap rendszert és nevezze el a projektet buszjáratok néven, majd ezen belül alkossa meg a programegységeket, kihasználva a rendszer szerkezetét!

**Integrálja az alábbi programmodulokat!**

**A) Az alkalmazás az alábbi menüpontokat tartalmazza:**

- *Kezdőlap*
- *Összes buszjárat*
- *Új buszjárat*
- *Új vizsgálata*

**B) Kezdőlap**

Tetszőleges tartalommal formázza meg!

**C) Összes buszjárat**

- Helyezzen el egy címet: “Menetrendek”
- A betöltődés után táblázatos elrendezésben jelenítse meg a **busz tábla adatait, melyben a rögzített buszok minden adata** szerepeljen!
- Készítsen a táblázatnak fejléce is!
- A táblázatban legyen egy adatok feliratú nyomógomb, melynek hatására a táblázatban **kiválasztott buszhoz tartozó vizsgálati időpontokat** jelenítse meg egy új oldalon (**Vizsgálatok**)!
- A táblázat buszjáratainak adatát egy **nyomógombra** kattintva legyen lehetőségünk egy **új oldalon módosítani**.

**D) Vizsgálatok**

- Az előző feladatban részletezett oldalon kiválasztott buszjáratához tartozó vizsgálati adatok jelenjenek meg!
- Adj meg, hány vizsgálatot végeztek az adott buszjáraton!

## E) Buszjárat módosítása

- A C) feladatban kiválasztott **buszjárat adatainak módosítása**.
- Sikertelen, vagy sikertelen művelet esetén **tájékoztasson** az oldal!
- **Sikeres módosítást követően** az összes buszjárat táblázata jelenjen meg!

## F) Új buszjárat rögzítése

- A menüpontra kattintva legyen lehetőségünk új buszjáratot rögzíteni!
- Végezzen hibaellenőrzést, törődjön a kötelezően kitöltendő mezőkkel!
- Rögzítés után az összes buszjárat adata jelenjen meg!

## G) Új vizsgálat rögzítése

- Vizsgálatot **csak létező buszjárhoz** rögzíthetünk

## MEGOLDÁS

### 1. Táblák létrehozása és adatok beszúrása az adatbázisba

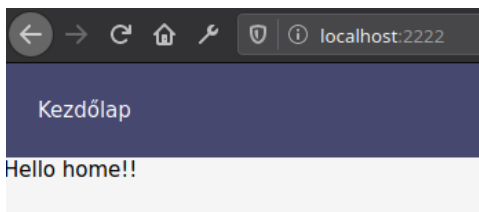
```
1 CREATE TABLE `busz` (  
2   `id` int AUTO_INCREMENT PRIMARY KEY,  
3   `indulas` varchar(30),  
4   `cel` varchar(30),  
5   `menetido` smallint,  
6   `alacsony` tinyint CHECK (`alacsony` IN (0, 1))  
7 )DEFAULT CHARSET = UTF8;  
8  
9 CREATE TABLE `szerviz`(  
10  `id` smallint AUTO_INCREMENT PRIMARY KEY ,  
11  `buszID` int,  
12  `datum` date,  
13  `megjegyzes` varchar(100),  
14  `engedelyezve` tinyint CHECK (`engedelyezve` IN (0, 1)),  
15  FOREIGN KEY( `buszID`) REFERENCES `busz`(`id`) ON DELETE SET NULL ON UPDATE CASCADE  
16 )DEFAULT CHARSET = UTF8;  
17  
18  
19 INSERT INTO `busz` VALUES  
20 (1,'Őrs vezér tere','Árpád híd',42,1),  
21 (2,'Őrs vezér tere','Cinkota garázs',32,1),  
22 (3,'Rákosszorvasztó','Etele tér',62,0);  
23  
24 INSERT INTO `szerviz` VALUES  
25 (1,1,'2018-01-12','minden rendben',1),  
26 (2,2,'2018-01-12','lengőkar hiba',0),  
27 (3,3,'2018-01-12','minden rendben',1),  
28 (4,null,'2018-01-13','ajtók nem záródnak',0),  
29 (5,null,'2018-01-14','minden rendben',1);
```

## 2. Modulok elkészítése

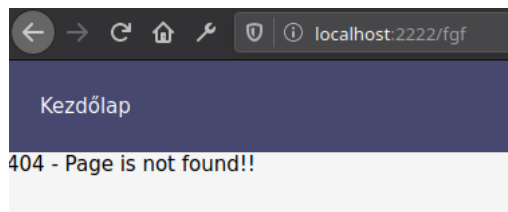
A kapott keretet használjuk:

A rendszer tartalmazza a kész fájlstruktúrát. Az .htaccess fájlnak köszönhetően az url-rewrite megtörténik minden nem Style mappában lévő fájl/mappa hívása esetén, amit az index.php kap meg. Az url szétszedése a *ProcedurálisMVC.pdf*-ben taglalt módon történik, melynek köszönhetően meghívásra kerül a bejegisztrált kontroller, ami vagy az üres HomeController, vagy a NotFoundController. A kontrollerek a controllers.php-ban laknak. Ide fogjuk majd felvenni a saját kontrollereinket az egyes modulok kiszolgálásához. A kontrollerek hívják a view függvényt, ami a megkapott adatokat átadja a \_layout.php-nak és a beinkludolt nézeteknek(view-k).

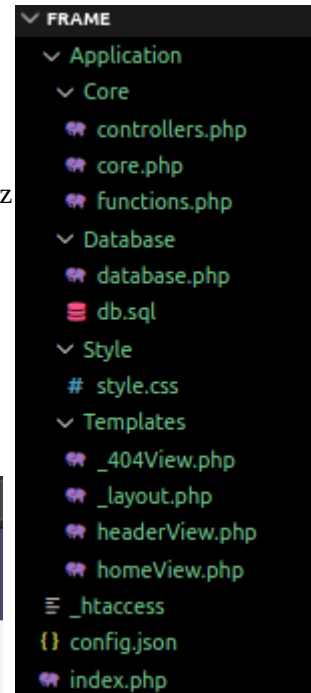
A nyers Frame így néz ki, ezt fogjuk kiegészíteni:



Van home kontroller



Nincs fgf kontroller



Még mielőtt bármit tennénk, állítsuk be az index.php-ban az APPROOT konstans értékét!!

```
// winsql.vereb.dc/diakXX/vizsgafeladat/index.php
// APPROOT = vizsgafeladat
```

### A) Menüpontok

Létrehozzuk a feladatban specifikált menüpontokat a headerView.php-ban. A Kezdőlap már létezik, adjuk hát hozzá az “összes buszjárat”, “új buszjárat” és az “új vizsgálat” menüpontokat!

```
Application > Templates > headerView.php > ...
1 <header class="header">
2   <a href="<?= APPROOT ?>/">Kezdőlap</a>
3   <a href="<?= APPROOT ?>/allBus">Összes buszjárat</a>
4   <a href="<?= APPROOT ?>/newBusForm">Új buszjárat</a>
5   <a href="<?= APPROOT ?>/newTestForm">Új vizsgálat</a>
6 </header>
```

Nagyon fontos, hogy a href értéke mindig az APPROOT mappával kiegészítve legyen meghívva!!

## B) Kezdőlap

A formázás itt tetszőleges, esztétikus legyen.

## C) Összes buszjárat

Első körben elkészítjük a fejlécben megadott allBus kontrollert és felregisztráljuk a core.php-ban. Regisztráció:

```
56 addRoute('/allBus', 'allBusController');  
57
```

Az addroute hozzáadja a routes tömbhöz a '/allBus' mintát és a hozzá tartozó allBusControllert. Ezt fogja a routing függvény kikeresni és meghívni, amikor a fejlécben az **Összes busz** gombra kattintunk.

*Helyezzen el egy címet: "Menetrendek"* - Az egyes kontrollereink megkapják a konfigurációt beolvasó getConfig és az adatbázis kapcsolatért felelős getConnection függvényt. Itt bevezetünk egy új kontrollert, az **errorController**-t, amelyet hiba esetén írunk ki a felhasználónak. Ha az errorController-t hívjuk, utána befejeződik az alkalmazás: **die**.

```
Application > Core > controllers.php > PHP Intelephense > newTestFormController  
51 function allBusController()  
52 {  
53     $config = getConfig( CONFPATH );  
54     $pdo    = getConnection( $config );  
55  
56     if(!$pdo)  
57     {  
58         errorController();  
59         die;  
60     }  
61  
62     $buses = getAllBus( $pdo );  
63  
64  
65     view([  
66         "title" => "- Menetrendek",  
67         'view'   => 'allBus',  
68         'buses' => $buses  
69     ]);  
70 }
```

```
198 function errorController()  
199 {  
200     view([  
201         "title" => "- Something is wrong while page loaded",  
202         'view'   => '_error'  
203     ]);  
204 }
```

A view függvény minden esetben tömböt vár, aminek tartalmazni kell egy view és egy title változót. Előbbi a megjelenítendő nézet nevét jelöli, utóbbi az ablakfülbe kiírandó címet adja meg.

- A betöltődés után táblázatos elrendezésben jelenítse meg a **busz tábla adatait, melyben a rögzített buszok minden adata szerepeljen!**

A 62. sorban létrehozuk a **buses** tömböt, ezt fogja az allBusesView.php megjeleníteni. Ehhez a kontrollerünk egy getAllBus nevű függvényt hív meg, ami megkapja a \$pdo referenciát és visszaadja az adatbázisból a `busz` tábla tartalmát. Akkor is tömbbel tér vissza, ha üres a reláció!

**Az adatbáziskezelő függvényeinket a Database/database.php fájlba írjuk!**

```

28  /**
29  * @param PDO $pdo
30  *
31  * @return array
32  */
33  function getAllBus( PDO $pdo )
34  {
35      $smt = $pdo->prepare('SELECT * FROM `busz`');
36
37      try
38      {
39          if (!$smt->execute())
40          {
41              throw new RuntimeException($smt->errorInfo()[2]);
42          }
43
44          return $smt->fetchAll();
45      }
46      catch (RuntimeException $e)
47      {
48          return [];
49      }
50  }

```

GetAllBus: Lekéri az adatbázisból a busz tábla adatait és vissza tér vele. Ha valami hiba üti fel a fejét, akkor üres tömbbel tér vissza.

A controllers.php 65. sorában meghívásra kerül a **view** függvény és megkapja a címet, a nézetet és a buszok tömbjét.

```

43  /**
44  * A view függvény a kapott adatokat kibontja és átadja a layoutnak.
45  * A $datas-nak tartalmaznia kell egy 'view' és egy 'title' kulcsot!
46  */
47  function view($datas)
48  {
49      extract($datas);
50      unset($datas);
51      require_once APPPATH.'Templates/_layout.php';
52  }

```

A 47. sorban lévő extract függvény a paraméterül kapott tömböt “**kicsomagolja**”, vagyis a benne lévő kulcs-érték pároknak új változókat hoz létre, az unset-el pedig törli a \$datas tömböt. Ezek után meghívja a

`_layout.php`-t, amely a kapott view értéke alapján meghívja a kontrollerhez tartozó nézetet. Ebben az esetben az `allBusView.php`-t.

A létrehozott `allBusView.php`

```
Application > Templates > allBusView.php > ...
1  <table class="allbus-table">
2      <tr>
3          <th>Azonosító</th>
4          <th>Indulás</th>
5          <th>cel</th>
6          <th>Menetidő</th>
7          <th>Alacsony</th>
8          <th>Adatok</th>
9          <th>Módosít</th>
10     </tr>
11     <?php foreach($buses as $bus): ?>
12         <tr>
13             <?php for($i=0;$i<count($bus) / 2; $i++): ?>
14                 <td><?= $bus[$i] ?></td>
15             <?php endforeach ?>
16             <td>
17                 <a href="<?= APPROOT ?>/testes/<?= $bus[0] ?>" class="btn btn-grn">Lekér</a>
18             </td>
19             <td>
20                 <a href="<?= APPROOT ?>/modifyBusForm/<?= $bus[0] ?>" class="btn btn-rng">Gyerünk</a>
21             </td>
22         </tr>
23     <?php endforeach ?>
24 </table>
```

- Készítsen a táblázatnak fejléce is!

A 2. és 10. sorok között létrehozzuk a statikus fejléce a táblázatnak. Ez a része mindig meg fog jelenni attól függetlenül, hogy lesz-e a táblázatnak tartalma, vagy sem.

A külső **foreach** ciklussal bejárjuk a **\$buses** tömböt, míg a belső **for** ciklussal csak a tömb indexeit járjuk végig és írjuk ki a kimenetre.

- A táblázatban legyen egy adatok feliratú nyomógomb, melynek hatására a táblázatban **kiválasztott buszhoz tartozó vizsgálati időpontokat** jelenítse meg egy új oldalon (**Vizsgálatok**)!
- A táblázat buszjáratainak adatát egy nyomógombra kattintva legyen lehetőségünk egy új oldalon módosítani.

A két gomb a 17. és a 20. sorban lett elhelyezve. Mindkét anchor tag megkapta a maga url-jét használva az `APPROOT` konstans és a css formázást. Az adatok feliratot ebben az esetben **lekér**-re cseréltem.

**Az url-hez hozzá kell fűzni az adott buszjárat azonosítóját!!**



A megjelenés:

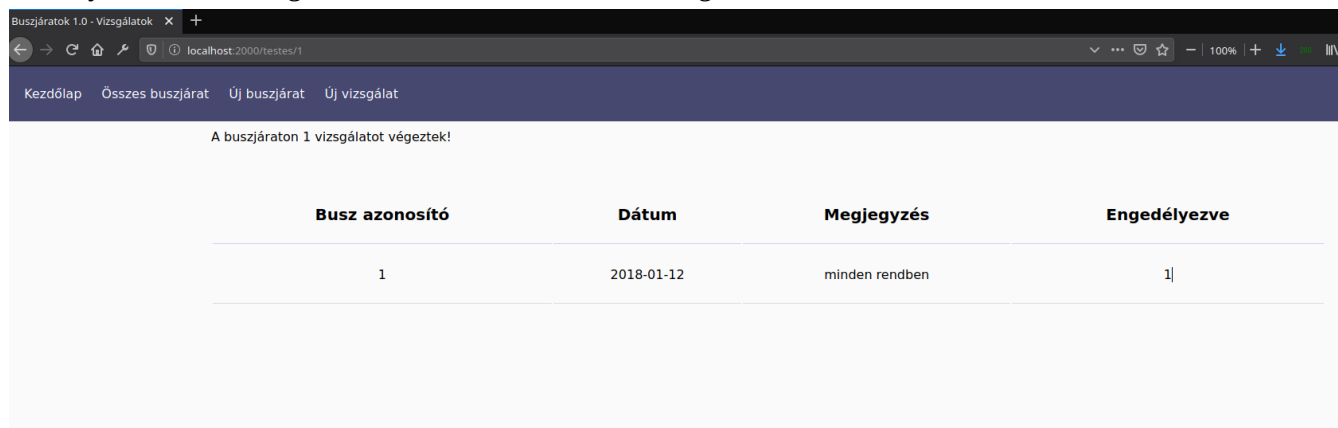


Azonosító	Indulás	cel	Menetidő	Alacsony	Adatok	Módosít
1	Őrs vezér tere	Árpád híd	42	1	Lekér	Gyerünk
2	Őrs vezér tere	Cinkota garázs	32	1	Lekér	Gyerünk
3	Rákosborzasztó	Etele tér	62	0	Lekér	Gyerünk

## D) Vizsgálatok

- Az előző feladatban részletezett oldalon kiválasztott buszjárhoz tartozó vizsgálati adatok jelenjenek meg!

Ha rányomunk a Lekér gombra, a link átdob minket a Vizsgálatok oldalra:



Busz azonosító	Dátum	Megjegyzés	Engedélyezve
1	2018-01-12	minden rendben	1

Itt meg kell említeni a mintaillesztés, vagy matching egy tulajdonságát, név szerint a minta-csoportokat. Ebben az esetben átadjuk az url-el az id-t is, vagyis a kapott url így néz ki:

<http://localhost:2000/testes/1>

Ekkor a `preg_match` függvény lehetőséget nyújt arra, hogy egy tömbbe kigyűjtse a csoportokba szedett mintákat. Ezt úgy tehetjük meg, hogy adunk neki egy harmadik paramétert, amibe automatikusan beleszűri az egyes csoportok értékeit, mint egy tömbbe.

```
preg_match($pattern, $cleanedUri, $matches)
```



Itt a \$pattern a minta, a cleanedUri a tisztított url és a \$matches lesz az a tömb, melybe kigyűjti az adatokat.

Pl.. A mintánk *testes/(\d+)*

```
array (size=2)
  0 => string '/testes/1' (length=9)
  1 => string '1' (length=1)
```

Ebben az esetben a 'testes/' részminta után van egy tag, melyet úgy tudunk megkülönböztetni a minta többi részétől, hogy zárójelbe tesszük. A \d bármilyen számkaraktereket helyettesít. Ekkor a preg\_match függvény az '1'-et egy külön kulcs alá teszi a \$matches tömbben.

Ezeket a csoportokat el is nevezhetjük, még pedig úgy, hogy a zárójelbe egy kérdőjelet teszünk, aztán kicsit html-es móddal <név> relációsjelek közé írjuk a csoport nevét.

Pl.. A mintánk *testes/(<id>\d+)*

```
array (size=3)
  0 => string '/testes/1' (length=9)
  'id' => string '1' (length=1)
  1 => string '1' (length=1)
```

Amint látjuk, az 1-es már kétszer szerepel és ahogy szeretnénk volna, az egyik esetben megkapta az **id** nevet. Ez alapján odaadhatjuk az adatokat a kontrollerünknek, majd lekérhetjük vele a megadott sort az adatbázisból.

```

Application > Core > functions.php > ...
27 function routing($cleanedUri)
28 {
29     global $routes;
30
31     foreach($routes as $pattern => $controller)
32     {
33         if(preg_match($pattern, $cleanedUri, $matches))
34         {
35             $controller($matches);
36             return;
37         }
38     }
39     notFoundController();
40 }
41

```

Különlegesség, hogy a PHP megengedi azt, hogy több paraméterrel lehet függvényt hívni, mint amennyit vár, de kevesebbet már hibát dob. Nem vettük észre, de az allBusController-nek nem adtunk át adatokat, mégis hibátlanul működik a program.

```

72 function testesController($datas)
73 {
74     $id = $datas['id'];
75
76     $config = getConfig( CONFPATH );
77     $pdo     = getConnection( $config );
78
79     if(!$pdo)
80     {
81         errorController();
82         die;
83     }
84
85     $testes = getTestes( $pdo, $id );
86
87
88     view([
89         "title"           => "- Vizsgálatok",
90         'view'            => 'testes',
91         "testes"          => $testes,
92         "numOfTestes"     => count($testes)
93     ]);
94 }

```

TestesController: A kapott adatból kiolvassuk az id-t és ez alapján kérjük le az adatbázisból a buszjáratokhoz tartozó vizsgálatokat a 85. sorban a `getTestes` függvénnyel.

A view-nak átadott értékek

- cím
- nézet neve
- a vizsgálatok - `$testes`
- és az alpont utolsó feladatában említett **‘Adja meg, hány vizsgálatot végeztek az adott buszjáraton!’** utasításnak megfelelő számérték, ami a lekérdezett vizsgálatok `count()` natív függvénnyel történő megszámlálása.

```
Application > Database > database.php > ...
52  /**
53   * @param PDO $pdo connection object
54   * @param $id bus Id
55   *
56   * @return array
57   */
58  function getTestes( PDO $pdo, $id ):array
59  {
60      $smt = $pdo->prepare('SELECT * FROM `szerviz` WHERE buszId = :id');
61
62      $smt->bindParam(':id', $id);
63
64      try
65      {
66          if (!$smt->execute())
67          {
68              throw new RuntimeException($smt->errorInfo()[2]);
69          }
70
71          return $smt->fetchAll();
72      }
73      catch (RuntimeException $e)
74      {
75          return [];
76      }
77  }
```

A `getTestes` függvény

A megszokott útvonalon eljutunk a `testesView.php`-hoz, ami az alábbi módon néz ki:

```

Application > Templates > testesView.php > ...
1 <div class="all-test-container">
2 |   A buszjáraton <?= $numOfTestes ?> vizsgálatot végeztek!
3 </div>
4 <table class="allbus-table">
5 |   <tr>
6 |       <th>Busz azonosító</th>
7 |       <th>Dátum</th>
8 |       <th>Megjegyzés</th>
9 |       <th>Engedélyezve</th>
10 |   </tr>
11 |   <?php foreach($testes as $test): ?>
12 |       <tr>
13 |           <?php for($i=1;$i<count($test) / 2; $i++): ?>
14 |               <td><?= $test[$i] ?></td>
15 |           <?php endfor ?>
16 |       </tr>
17 |   <?php endforeach ?>
18 </table>
19

```

Az 1-3. sorban kiírásra kerül a vizsgálatok száma 'numOfTestes', a 4.től pedig a táblázat, szintén foreach/for ciklusszerkezettel.

Busz azonosító	Dátum	Megjegyzés	Engedélyezve
1	2018-01-12	minden rendben	1

## E) Buszjárat módosítása

- A C) feladatban kiválasztott **buszjárat adatainak módosítása.**
- Sikertelen művelet esetén **tájékoztasson** az oldal!
- Sikeres módosítást követően az összes buszjárat táblázata jelenjen meg!

Ha az összes buszjárat táblázatának Módosít mezőjében található gombjára kattintunk, a link átdob minek a

például a [/modifyBusForm/2](#) oldalra. Láthatjuk, hogy itt is adatot adunk át az url-ben, mégpedig egy azonosítót. Ezt az adatot a kontroller regisztrációnál ismét névvel látjuk el és a kontrollerben így hívkozunk majd rá.

60

```
addRoute('/modifyBusForm/(?<id>\d+)', 'modifyBusFormController');
```

Mivel itt már adatbázis módosítás is lesz, ezért a funkciót két részre bontjuk. Híváskor csak egy form jelenik meg, módosításkor pedig átirányítás történik majd, hogy az adatokat kitöröljük a böngészőből. A formot a modifyBusFormView.php jeleníti meg, amit a modifyBusFormController szolgál ki.

```
Application > Core > controllers.php > PHP Intelephense > modifyBusController
132 function modifyBusFormController($datas)
133 {
134     $id = $datas['id'];
135
136     $config = getConfig( CONFPATH );
137     $pdo     = getConnection( $config );
138
139     if(!$pdo)
140     {
141         errorController();
142         die;
143     }
144
145     $bus = getBusById( $pdo, $id );
146
147     if(!$bus)
148     {
149         errorController();
150         die;
151     }
152
153     view([
154         "title"      => "- Az Ön szolgálatában",
155         'view'       => 'modifyBusForm',
156         "id"         => $bus['id'],
157         "indulas"    => $bus['indulas'],
158         "cel"        => $bus['cel'],
159         "menetido"   => $bus['menetido'],
160         "low"        => $bus['alacsony'],
161     ]);
162 }
```

A 134-től 145. sorig ismerős az eljárás. A 145. sorban lekérjük az adott buszjárat adatait azonosító alapján, majd leellenőrizzük, hogy kaptunk-e adatot. Ha nincs meghívjuk az errorControllert.

A view-nak itt már elég sok adatot adunk át:

- oldal címe
- nézet neve
- busz adatai.

A nézet kódja és képe:

```
Application > Templates > newBusFormView.php > form
1  <form action="<?= APPROOT ?>/newBus" method="POST">
2      <fieldset class="modify-bus-form" >
3          <legend>Új buszjárat felvitele</legend>
4
5          <label for="">Indulás</label>
6          <input type="text" name="indulas" value="<?= $indulas ?>" autofocus required>
7
8          <label for="">cel</label>
9          <input type="text" name="cel" value="<?= $cel ?>" required>
10
11          <label for="">Menetidő</label>
12          <input type="number" name="menetido" min="0" value="<?= $menetido ?>" required>
13
14          <label for="">Alacsony</label>
15          <input type="number" name="low" max="1" min="0" value="<?= $low ?>" required>
16
17          <input type="submit" value="Submit" class="btn btn-grn">
18          <input type="hidden" name="id" value="<?= $id ?>">
19      </fieldset>
20 </form>
```

A kód. Látható, hogy az url-ben kapott azonosítót itt rejtett mezőben adjuk tovább – **18.** sor – és POST metódust használunk.

### Buszjárat módosítása

Indulás

Órs vezér tere

cel

Cinkota garázs

Menetidő

32

Alacsony

1

Submit

## Megjelenés

Láthatjuk, hogy minden egyes inputmezőbe beleírjuk a megfelelő adatot. Gondoskodunk róla, hogy a form minden mezőjét kötelező legyen kitölteni - **required** – és az első mező autofocus tulajdonságot kap.

Az űrlap minden esetben adatokkal lesz elküldve, így azt már php-ból ebben a kivételes esetben már nem kell leellenőrizni. Elküldés esetén a `modifyBusController` hívódik meg.

61

```
addRoute('/modifyBus', 'modifyBusController');
```



```

Application > Core > controllers.php > ...
103
164 function modifyBusController($datas)
165 {
166     $bus    = $_POST;
167
168     $config = getConfig( CONFPATH );
169     $pdo    = getConnection( $config );
170
171     if(!$pdo)
172     {
173         errorController();
174         die;
175     }
176
177     $bus = updateBus( $pdo, $bus );
178
179     if(!$bus)
180     {
181         view([
182             "title" => "- Something went wrong while page is loaded",
183             'view' => '_error'
184         ]);
185     }
186     else
187     {
188         view([
189             "title" => "- Modify successful",
190             'view' => 'successfulModify'
191         ]);
192     }
193
194
195     header('refresh: 2; url= '.APPROOT.'/allBus');
196 }

```

Itt a kapott adatokat lementjük egy \$bus nevű változóba, szokás szerint kapcsolatot nyitunk az adatbázissal, leellenőrizzük, hogy sikerült-e, majd a 177. sorban frissítjük a busz tábla megfelelő rekordját az új adatokkal.

```

Application > Database > database.php > PHP Intelephense > insertNewBus
146  /**
147   * @param PDO $pdo connection object
148   * @param $bus datas of a bus
149   *
150   * @return boolean
151   */
152  function updateBus( $pdo, $bus ): bool
153  {
154      extract($bus);
155
156      $smt = $pdo->prepare( 'UPDATE `busz`
157                          SET `indulas`    = :indulas,
158                              `cel`        = :cel,
159                              `menetido`   = :menetido,
160                              `alacsony`   = :low
161                          WHERE id = :id');
162
163      $smt->bindParam(':id', $id);
164      $smt->bindParam(':indulas', $indulas);
165      $smt->bindParam(':cel', $cel);
166      $smt->bindParam(':menetido', $menetido);
167      $smt->bindParam(':low', $low);
168
169      try
170      {
171          if (!$smt->execute())
172          {
173              throw new RuntimeException($smt->errorInfo()[2]);
174          }
175
176          return true;
177      }
178      catch (RuntimeException $e)
179      {
180          return false;
181      }
182  }

```

A kapott \$bus tömböt kicsomagoljuk (opcionális, lehet indexel is hivatkozni, de ez kényelmesebb és átláthatóbbkódot eredményez), megkreáljuk a statement-et, beparaméterezzük és logikai értékkel térünk vissza.

Mivel minden esetben vissza kell jeleznünk a felhasználónak a művelet sikeressége, vagy sikertelensége végett, a modifyBusController-ben a módosítás visszatérési értéke szerint fogunk meghívni két nézetet. Az első az \_errorView.php, ha sikertelen, a második esetben a successfulModifyView.php-t, ha sikeres, majd a 195. sorban a fejlécbe egy eddig még nem tanult **refresh** paramétert helyezünk el. Ez hasonlatos a Location paraméterhez annyi különbséggel, hogy itt időzítést is meg lehet adni, míg a Location azonnali redirect-el jár. Ebben az esetben az oldal újratöltődik 2 másodperc múlva, hogy a felhasználónak legyen ideje elolvasni a státusz üzenetet.

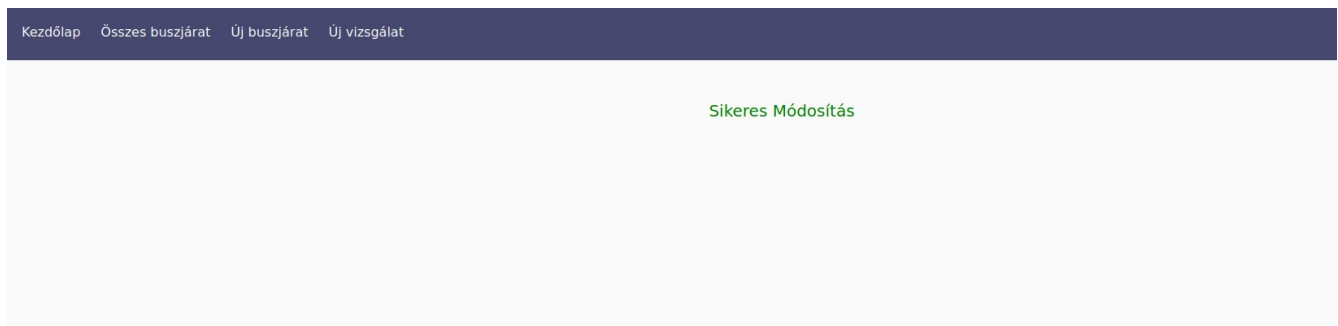
A két nézet:

```
Application > Templates > _errorView.php > ...  
1 <main class="error-cont">  
2 |   Sorry!! Something went wrong!<br>  
3 |   Please try again it later!  
4 </main>
```

**\_errorView.php** Bár már más kontrollereknél is használtuk, most jelenik meg először a leírásban.

```
Application > Templates > successfulModifyView.php > ...  
1 <div class="succ-mod-container">Sikeres Módosítás</div>
```

**successfulModifyView.php**



## F) Új buszjárat rögzítése

- A menüpontra kattintva legyen lehetőségünk új buszjáratot rögzíteni!
- Végezzen hibaellenőrzést, törődjön a kötelezően kitöltendő mezőkkel!
- Rögzítés után az összes buszjárat adata jelenjen meg!

A screenshot of a web application interface showing a form for adding a new bus route. The navigation bar is the same as in the previous image. The main content area is light gray and contains a form titled "Új buszjárat felvitele". The form has four input fields: "Indulás" (Start), "Cél" (Destination), "Menetidő" (Travel time), and "Alacsony" (Low). Each field has a small downward arrow icon next to it. Below the fields is a green "Submit" button.

Kontrollerek regisztrációja:

```
63 addRoute('/newBusForm', 'newBusFormController');
64 addRoute('/newBus', 'newBusController');
```

A megszokott módon megjelenítés és feldolgozás, külön-külön kontrollerekkel.

## A form-hoz tartozó kód

```
Application > Core > controllers.php > PHP Intelephense > newBusController
10
11 function newBusFormController()
12 {
13     view([
14         "title"      => "- Új buszjárat felvitele",
15         'view'       => 'newBusForm'
16     ]);
17 }
18 }
```

```
Application > Templates > newBusFormView.php > ...
1 <form action="<?= APPROOT ?>/newBus" method="POST">
2     <fieldset class="modify-bus-form" >
3         <legend>Új buszjárat felvitele</legend>
4
5         <label for="">Indulás</label>
6         <input type="text" name="indulas" autofocus required>
7
8         <label for="">Cél</label>
9         <input type="text" name="cel" required>
10
11         <label for="">Menetidő</label>
12         <input type="number" name="menetido" min="0" required>
13
14         <label for="">Alacsony</label>
15         <input type="number" name="low" max="1" min="0" required>
16
17         <input type="submit" value="Submit" class="btn btn-grn">
18     </fieldset>
19 </form>
20
```

Mivel az alacsony mező logikai értéket vehet csak fel, célszerű a number max értékét 1-re, minimum értékét 0-ra és minden mezőt kötelezően kitöltendőre állítani.

A feldogozáshoz tartozó kód:

```
19 function newBusController()
20 {
21
22     $bus = $_POST;
23
24     $config = getConfig( CONFPATH );
25     $pdo     = getConnection( $config );
26
27     if (!$pdo)
28     {
29         errorController();
30         die;
31     }
32
33     $insert = insertNewBus( $pdo, $bus );
34
35     if ($bus)
36     {
37         header('Location: '.APPROOT.'/allBus');
38     }
39     else
40     {
41         errorController();
42     }
43 }
```

Ha sikerül az insertálás, átlépünk az **Összes busz** oldalra, különben **error**.

## G) Új vizsgálat rögzítése

- Vizsgálatot **csak létező buszjárat**hoz rögzíthetünk

at Új vizsgálat

## Új vizsgálat fül

Az új vizsgálat modul kontrollerei:

```
66     addRoute('/newTestForm', 'newTestFormController');
67     addRoute('/newTest', 'newTestController');
68
```

Szokás szerint: egy form, ahogy a felhasználó az új adatokat adhatja meg és a feldolgozás, melyben az adatbázis művelet zajlik a végén átirányítással.

```
90 function newTestFormController()
91 {
92     $config = getConfig( CONFPATH );
93     $pdo     = getConnection( $config );
94
95     if(!$pdo)
96     {
97         errorController();
98         die;
99     }
100
101     view([
102         "title" => "- Új vizsgálat hozzáadása",
103         'view'   => 'newTestForm',
104         "ids"    => getBusIds($pdo)
105     ]);
106 }
```

A feladat szerint 'Vizsgálatot **csak létező buszjárat**hoz rögzíthetünk', ezért le kell kérnünk az adatbázisból a létező buszjárat azonosítóit. Ezt a **getBusIds** függvénnyel hajtjuk végre.

```
Application > Database > database.php > PHP Intelephense > insertNewTest
79 function getBusIds(PDO $pdo)
80 {
81     $smt = $pdo->prepare("SELECT `id` FROM `busz`");
82
83     try
84     {
85         if (!$smt->execute())
86         {
87             throw new RuntimeException($smt->errorInfo()[2]);
88         }
89
90         return $smt->fetchAll();
91     }
92     catch (RuntimeException $e)
93     {
94         return null;
95     }
96 }
```



A nézet. Ha van az adatbázisban buszjárat, megjelenik a form, különben értesítjük a felhasználót a hiányosságról.

```
Application > Templates > newTestFormView.php > ...
1  <?php if(count($ids) > 0): ?>
2      <form action="<?= APPROOT ?>/newTest" method="POST">
3          <fieldset class="modify-bus-form" >
4              <legend>Új vizsgálat felvitele</legend>
5
6              <label for="">Buszjárat azonosító</label>
7              <select name="id" id="">
8                  <?php foreach($ids as $id): ?>
9                      <option value="<?=$id['id']?>"><?=$id['id']?></option>
10                 <?php endforeach ?>
11             </select>
12             <br>
13             <label for="">Dátum</label>
14             <input type="date" name="date" required>
15
16             <label for="">Megjegyzés</label>
17             <input type="text" name="note" min="0" required>
18
19             <label for="">Engedélyezve</label>
20             <input type="number" name="allowed" max="1" min="0" required>
21
22             <input type="submit" value="Submit" class="btn btn-grn">
23         </fieldset>
24     </form>
25 <?php else: ?>
26     Sajnáljuk, nincs buszjárat regisztrálva a rendszerünkben.
27 <?php endif ?>
```

A szerviz táblába négy adatot kell beinzertálnunk: buszID, datum, megjegyzes és engedélyezve. A buszID a legördülő-menü tételeiből választható csak, a datum adatát egy **dátum**, míg az engedélyezve adatot egy **number** típusú inputtal adhatja meg a felhasználó. Ne felejtsük el a number input **min-max** attribútumait beállítani!! A form a newTest controllernek küldi el az adatait.



The screenshot shows a web browser window with the address bar displaying 'localhost:2000/newTestForm'. The browser's navigation bar includes back, forward, and search icons, along with a star icon for bookmarks and a zoom level of 100%. The page has a dark blue header with white text links: 'Kezdőlap', 'Összes buszjárat', 'Új buszjárat', and 'Új vizsgálat'. The main content area is white and features a form titled 'Új vizsgálat felvitele' (New Exam Submission). The form contains the following fields and controls:

- 'Buszjárat azonosító' (Bus route identifier): A dropdown menu showing the value '4'.
- 'Dátum' (Date): A text input field with a placeholder 'éééé . hh . nn .'.
- 'Megjegyzés' (Remarks): A text input field.
- 'Engedélyezve' (Approved): A text input field with a small dropdown arrow on the right side.
- 'Submit': A green button with white text.

**newTestController:**

```
Application > Core > controllers.php > PHP Intelephense > newBusFormController
96 function newTestController()
97 {
98     $test    = $_POST;
99
100    $config = getConfig( CONFPATH );
101    $pdo    = getConnection( $config );
102
103    if(!$pdo)
104    {
105        errorController();
106        die;
107    }
108
109    $result = insertNewTest( $pdo, $test );
110
111    if(!$result)
112    {
113        errorController();
114        die;
115    }
116
117    header('Location: '.APPROOT.'/allBus');
118 }
```

Siker esetén átirányítás, különben `errorController`.

