

# Vasúti menetrend: HF specifikáció

*Jójárt Bence, GFHSCH, 2017.04.18.(módosítva 04.24.)*

## A feladat rövid bemutatása

Az általam választott feladat egy vasúti menetrend nyilvántartó. A rendszer tárolja az egyes vonatokat (azonosító, melyik vasútvonalhoz tartozik), az egyes állomásokat (azonosító, név), a köztük lévő kapcsolat pedig az, hogy az adott vonat megáll -e az adott állomáson, és ha igen, akkor mikor, illetve az adott állomás induló- vagy végállomása e az adott járatnak.

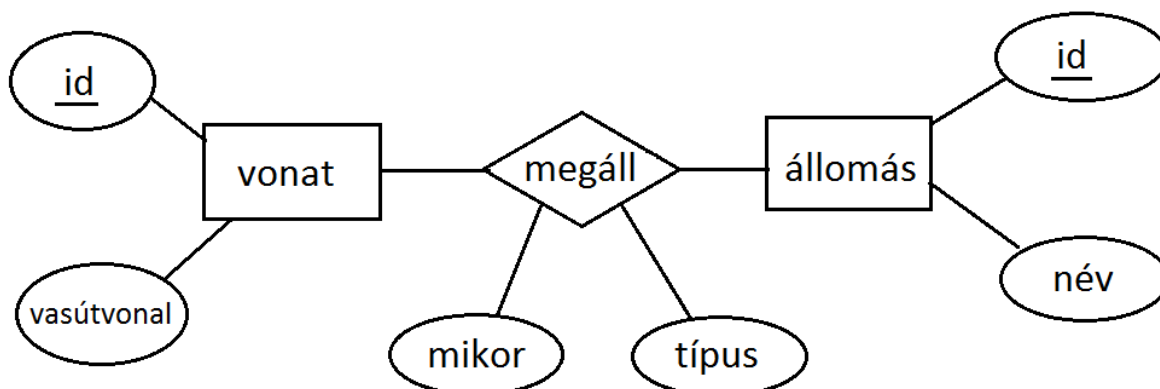
## A főoldalról elérhető oldalak és az azokon elérhető funkciók listája

- Főoldal: a főoldal statikus HTML oldal, innen érhetők el a különböző menüpontok: menetrend kereső, állomások böngészése, új járat felvétele.
- Menetrend kereső: a felhasználó beírja a kiinduló és a célállomást, és ennek megfelelően a rendszer ugyanazon az oldalon kilistázza, hogy melyik járatokkal tud közlekedni a két pont között.
- Új járat felvétele: a felhasználó ennél a menüponttal teljesen új járatot vehet fel a rendszerbe. Itt először csak a kezdő- és a végállomást kell megadnia, illetve a hozzájuk tartozó indulási és érkezési időket, illetve annak a vasútvonalnak a számát, amelyen a járat közlekedik.
- Új állomás felvétele: ez az oldal kilistázza felsorolásszerűen az összes már meglévő állomást, illetve a felhasználó új állomásokat vehet fel a rendszerbe.
- Járatok módosítása: ez az oldal kilistázza az összes járatot. Ezek közül a felhasználó választhat, hogy melyiket akarja módosítani, vagy törölni. Törlés esetén az adott járat azonnal kitörlődik. Módosításkor egy olyan oldalra kerül, ahol módosíthatja az általa kiválasztott járatot.
- Adott járat módosítása: a felhasználó hozzáadhat új köztes megállóhelyeket a járatokhoz, vagy akár módosíthatja is a járatok egyes megállóihoz tartozó megállási időpontokat, sőt, akár törölheti is azokat. A járatok induló- és végállomása fix, azokat nem lehet törölni vagy az idejüket módosítani.

## Adatbázis séma

Az adatbázisban az egyes vonatokat és az egyes állomásokat tároljuk. A köztük lévő kapcsolótábla mondja meg, hogy mikor jár az adott vonat az egyes állomáson (ez egy több-több kapcsolat). Az egyes entitások attribútumai:

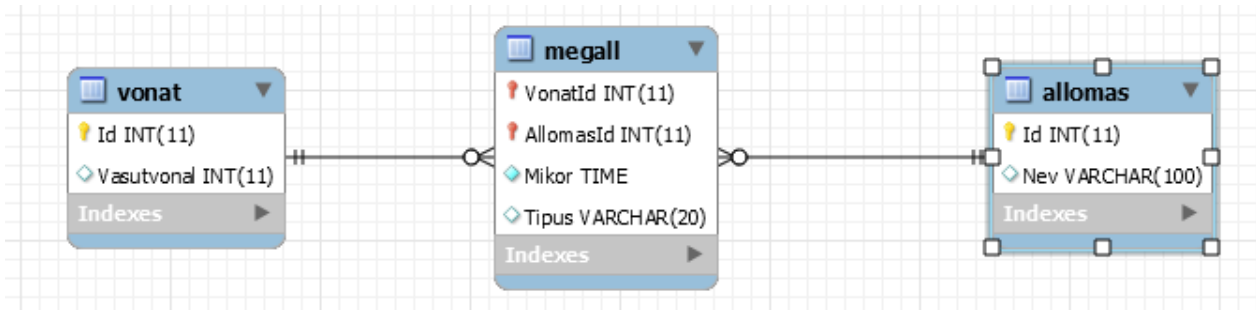
- Vonat: Id, Vasútvonal
- Állomás: Id, Név
- Megáll: VonatId, ÁllomásId, Mikor, Típus (a típus azt mondja meg, hogy az adott megálló induló-, köztes-, vagy végállomás)



# Fejlesztői dokumentáció

## Az adatbázis

Elsőként az adatbázis relációs sémáját készítettem el, a specifikációban bemutatott módon.



Minden Vonat egy egy járatnak felel meg. Tartozik hozzá egy vasútvonal szám, ez 0-nál nagyobb egész. A Vonatot az Id-ja azonosítja, mely egy AUTO INCREMENT elsődleges kulcs. Az Állomásoknak hasonlóan van egy neve, és egy AUTO INCREMENT elsődleges kulcs Id-ja. A Vonatok megállnak az Állomásokon, ezt a több-több kapcsolatot a Megáll tábla modellezi. Itt természetesen külön-külön külső kulcsok, és együtt elsődleges kulcsok a Vonat és az Állomás Id-ja. A megálló típusa mondja meg, hogy induló állomásról, köztes megállóról, vagy végállomásról beszélünk. A TIME típusú mikor attribútum pedig azt mondja meg, hogy mikor áll meg (vagy indul, vagy beér) az adott vonat az adott állomáson.

## Felhasználói felület (amit a felhasználó lát a weboldal szolgáltatásaiból)

Az **alkalmazás főoldala** (index.php) egy statikus html oldal. A fejléc, a lábléc, és a menü minden oldalon ugyanaz, az oldalanként változó tartalom a menütől jobbra jelenik meg. Itt ez egy nagy kép egy vonatról. A fejlécen a bal felül lévő képre, vagy a VASÚTI MENETREND címre kattintva visszajutunk a főoldalra. A menüben pedig navigálhatunk a különböző oldalak között.



A **menetrend kereső** oldalon (kereso.php) a felhasználó legördülő listából kiválasztja, hogy honnan akar menni, hova akar menni, és ezt a tervezés gombbal elküldi az oldalnak. Az válaszul indulási idő szerint rendezve kiírja a járatokat, hogy melyikkel tud utazni (jelen esetben a példa kedvéért 3 olyan járatot töltöttem fel, ami megfelel a Dunaújvárosból Kelenföldre utazó felhasználónak).



**VASÚTI MENETREND**

- [Menetrend kereső](#)
- [Új járat felvétele](#)
- [Új állomás felvétele](#)
- [Járatok módosítása](#)

**KERESÉS A MENETRENDBEN**

Honnan?

Hova?

Vasútvonal	Indulás	Mikor indul	Érkezés	Mikor érkezik
42	Dunaújváros Induló állomás	03:31:00	Kelenföld Köztes megálló	05:06:00
42	Dunaújváros Induló állomás	04:30:00	Kelenföld Köztes megálló	06:06:00
42	Dunaújváros Induló állomás	05:30:00	Kelenföld Köztes megálló	07:06:00

Készítette: Jójárt Bence, GFHSCH

Az **új járat felvétele** oldalon (ujjarat.php) a felhasználó teljesen új járatot adhat hozzá az adatbázishoz. Ehhez meg kell adnia a vasútvonal számát, egy induló állomást (itt is legördülő listából), és hogy mikor indul onnan a vonat, és még a vonat célállomását, és hogy mikor ér be oda. A köztes megállót később tudja majd hozzáadni, a járatok módosítása menüpont alatt. A felhasználónak természetesen arra is oda kell figyelnie, hogy a vonat nem indulhat korábban, mint ahogy beér, és a kiinduló és a célállomás nem lehet azonos. Ha sikeres az új járat létrehozása, arról a felhasználó visszajelzést kap. Ha valamiért nem sikerült, arról is kap hibaüzenetet.



**VASÚTI MENETREND**

- [Menetrend kereső](#)
- [Új járat felvétele](#)
- [Új állomás felvétele](#)
- [Járatok módosítása](#)

**ÚJ JÁRAT FELVÉTELE A RENDSZERBE**

Vasútvonal száma?

Induló állomás?

Mikor indul?

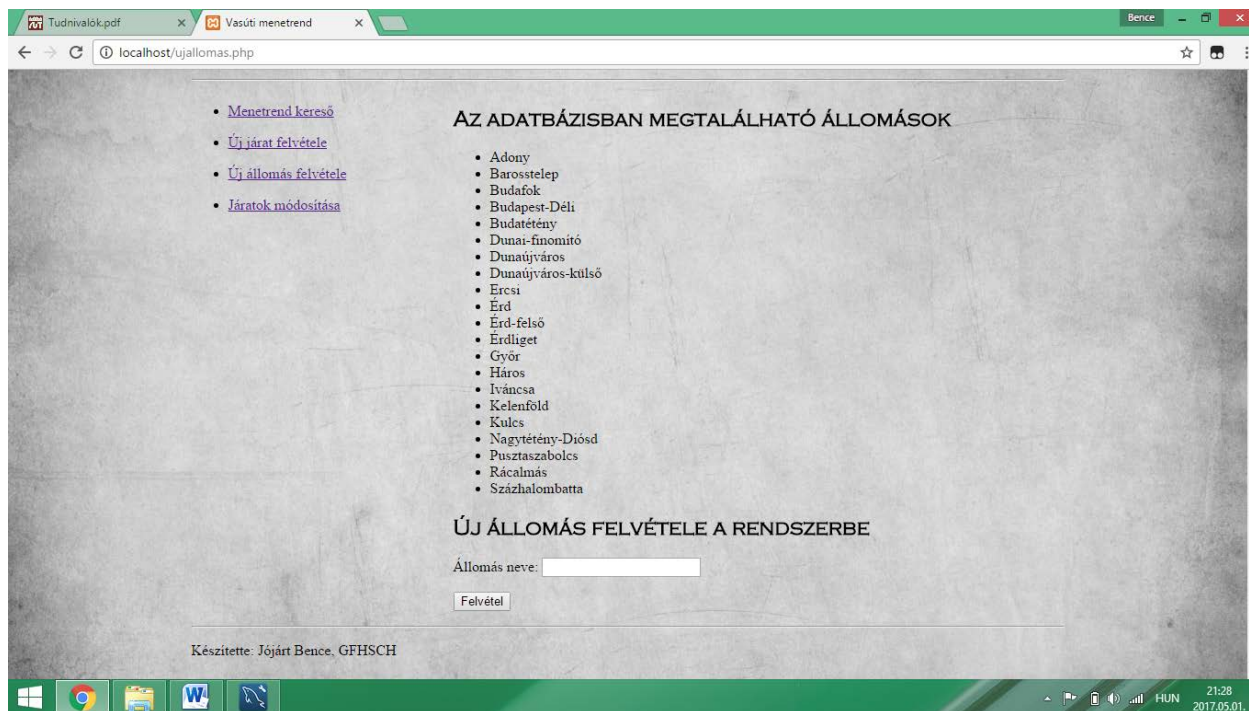
Végállomás?

Mikor érkezik?

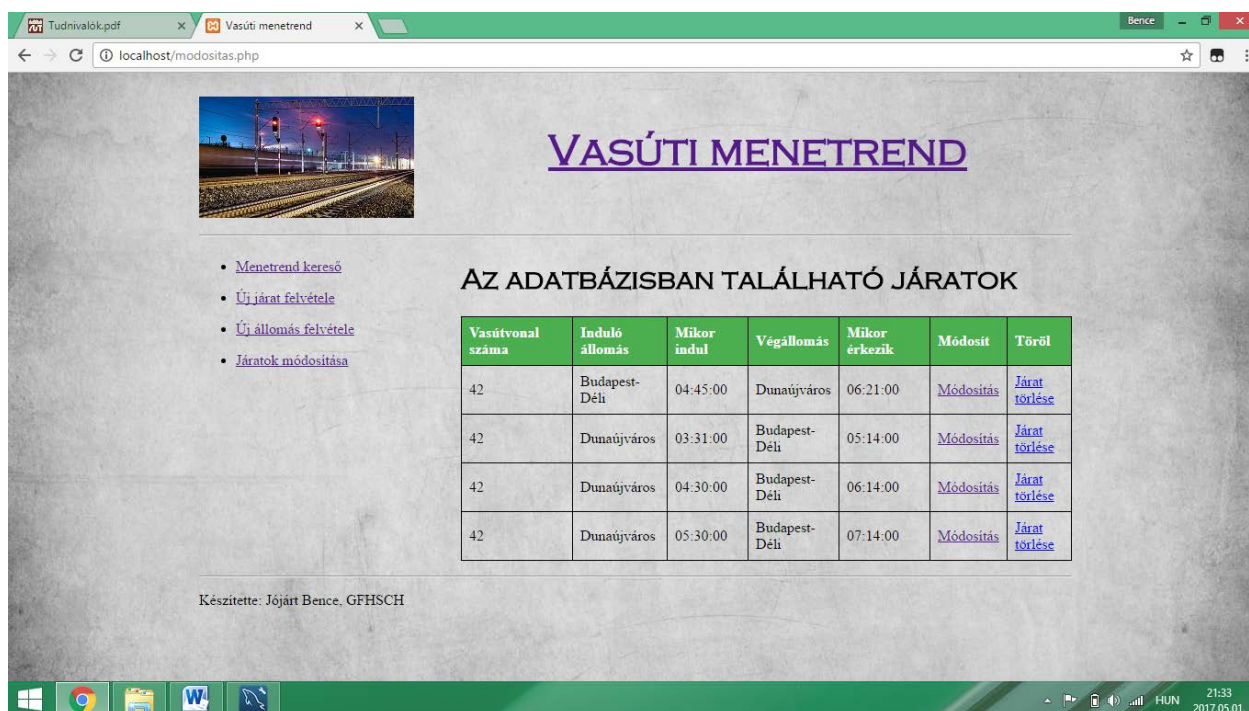
Készítette: Jójárt Bence, GFHSCH



**Új állomást** is fel lehet venni az adatbázisba, ezt az 'Új állomás felvétele' menüpontban lehet megtenni (ujallomas.php). Itt először kilitázza az oldal a már meglévő állomásokat, ha a felhasználó esetleg olyat akarna felvenni, ami már létezik. Az oldal alján tudunk új állomást felvenni. Azonban ha a felhasználó mégis olyan állomást akarna hozzáadni, ami már létezik a rendszerben, akkor hibaüzenetet kap, és nem jön létre még egy ugyanolyan nevű állomás. Arra figyelni kell még, hogy az állomásban nem lehet szóköz karakter, mert a mysql injection elleni védelem miatt az ilyen állomás nem fog felvevődni.



A **járatok módosítása** menüponttal elérhető oldal (modositas.php) kilitázza az összes járatot, induló állomás szerint rendezve, azok kezdő-és végállomásával, ill. azt hogy mikor indul és mikor ér be a céljához a vonat, és még a vasútvonal számát is. Minden járatot lehet módosítani, vagy törölni. Ha a járat törlésére kattintunk, akkor a teljes járat azonnal kitörlődik (és a hozzá kapcsolódó Megáll rekordok is). Ha a módosításra kattintunk, ott pedig egy új oldal töltődik be, az adott járat bővebb információival. A példában nézzük meg a 3:31-kor Dunaújvárosból induló járatot.




Ekkor bejön egy új oldal, ahol az **adott járat** kapcsolatban látunk infókat (jaratmodosit.php?id=1). Az URL id=1 részében látjuk, hogy az 1-es id-jű vonatról van itt szó. Ezen az oldalon ki van listázva a járat összes megállója, asszerint rendezve, hogy mikor áll meg ott. A köztes megállóknál módosíthatjuk, hogy mikor álljon ott meg, de természetesen ennek továbbra is az indulási idő és a végállomásra beérkezési idő közé kell esnie, különben a rendszer nem fogja engedni a módosítást. Megállókat törölhetünk is, de csak a köztes megállókat. Az oldal alján új köztes megállót is hozzáadhatunk a járatunkhoz, de a megállási időre továbbra is fennállnak az eddigi feltételek. Ha módosítottunk valamit, vagy megállót adtunk hozzá a járatához, akkor ugyanerre az oldalra jutunk vissza, hogy ne kelljen újra megkeresnünk a járatot.

Tudnivalók.pdf

Vasúti menetrend

Bence

localhost/jaratmodosit.php?id=1



Menetrend kereső

Új járat felvétele

Új állomás felvétele

Járatok módosítása

A KIVÁLASZTOTT JÁRAT ADATAI:

Vasútvonal száma: 42

Állomás	Megáll	Típus	Megállási idő módosítása	Töröl
Dunaiújváros	03:31:00	Induló állomás		
Rácalmás	03:38:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Kulcs	03:42:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Adony	03:47:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Pusztaszabolcs	03:56:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>

Windows

Chrome

File Explorer

Word

Taskbar icons

System tray

21:40

2017.05.01.

Tudnivalók.pdf

Vasúti menetrend

Bence

localhost/jaratmodosit.php?id=1

Barosstelep	04:52:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Budatény	04:55:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Háros	04:58:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Budafok	05:00:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Kelenföld	05:06:00	Köztes megálló	Új idő: --:-- <input type="button" value="Módosít"/>	<a href="#">Megálló törlése</a>
Budapest-Déli	05:14:00	Végállomás		

ÚJ KÖZTES MEGÁLLÓ HOZZÁADÁSA A JÁRATHOZ

Állomás neve? Adony

Mikor áll meg ott? --:--

Hozzáadás

Készítette: Jójárt Bence, GFHSCH

Windows

Chrome

File Explorer

Word

Taskbar icons

System tray

21:40

2017.05.01.

## Az elkészített állományok felépítése, funkciói (hogyan működik az alkalmazás szerver oldalon)

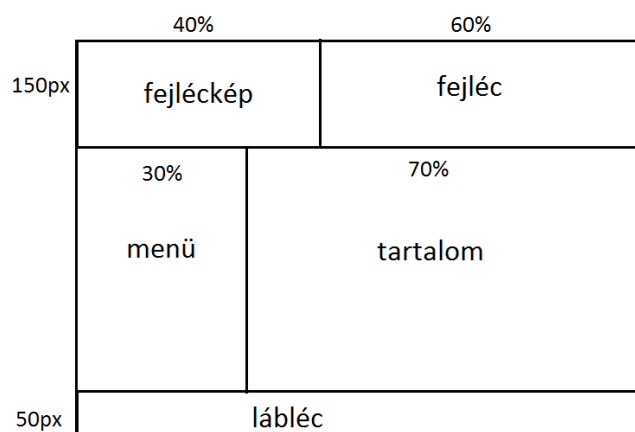
A teljes alkalmazás 12 darab php kiterjesztésű fájlból, és egy css fájlból áll. Minden php oldal szerkezete a következőképpen néz ki: be van include-olva a header.php, a menu.php, majd jön az oldal tartalma (div id=tartalom-ként), és legvégül a footer.php. Ez azért van így, mert a fejléc, a menü és a lábléc minden oldal esetén teljesen ugyanaz.

Elsőként a **header.php**-t mutatom be. Itt található az oldal címe (Vasúti menetrend), és egy kép a bal oldalon. Ezeket href-ekkel láttam el, hogy rájuk kattintva a főoldalra jussunk vissza. A header.php további fontos funkciókat is ellát: itt include-oltam be a theme.css fület, hogy az oldalak meg legyenek formázva. Hogy működjenek az ékezetes betűk, azt is itt állítom be, http meta, header() függvény és mysqli parancs formájában. Valamint a MySQL szerverhez is itt kapcsolódunk.

A következő a **menu.php**. Ez egy eléggé egyszerű és rövid állomány, 4 href-ből áll listába rendezve, amiken keresztül eljuthatunk az előző részben bemutatott menüpontokhoz, amik rendelkeznek a különböző funkciókkal.

A minden oldalon megtalálható állományok közül a végső a **footer.php**. Ennek a fő funkciója az, hogy itt bontódik a header.php-ban felépített MySQL kapcsolat, ill. a feladatkiírásnak megfelelően itt van a nevem és a Neptun-kódom.

Ezen három állományhoz minden oldalon tartozik egy-egy <div> blokk, ill. a minden állományban megtalálható tartalomhoz is tartozik egy <div>. A **theme.css** file egy stíluslap, ez az állomány felelős az oldal stílusáért, formázásáért. Az oldalelrendezést a következőképpen alakítottam ki:



Ezek után következzenek a felhasználó által látott állományok bemutatása, amik tulajdonképpen „csak” a tartalom részben térnek el egymástól. A főoldal az **index.php**, itt a tartalom csak egy kép egy vonatról.

A menüpontok szerint sorban haladva jön a **kereso.php**. A legördülő listákból kiválasztható állomások már egy MySQL lekérdezés eredményei. A keresés során a felhasználó egy POST kérést küld a szervernek, amit szerveroldalon feldolgozunk adatbázis-lekérdezés formájában. Az útvonal-kereső lekérdezés lényege az, hogy összeillesztjük önmagával a magáll táblát a vonatid szerint, majd az így kapott rekordokat szűrjük a kiinduló- és a célállomásra. A mysqli\_real\_escape\_string() függvényt minden ilyen lekérdezésben alkalmazza a teljes alkalmazás, hogy védjük az adatbázist az esetleges felhasználói támadásoktól. Ha a kiinduló és a célállomás ugyanaz, akkor hibaüzenetet küldünk. Egyébként pedig egy táblázatban kilistázzuk a MySQL által visszaadott eredményeket, a mysqli\_fetch\_array() függvény használatával.

A következő menüpont az új járat felvétele, az ennek megfelelő php állomány az **ujjarat.php**. Itt vehet fel a felhasználó új járatokat az adatbázisba. Itt megint egy http POST kéréssel dolgozunk, az induló és a végállomás

Készítette: Jójárt Bence, GFHSCH, 2017.05.02.



esetén továbbra is legördülő listából lehet választani, az időket pedig time típusú input mezőkkel kérjük be. Szerver oldalon beérkezik a POST kérés. Itt először hibákat vizsgálunk (induló állomás nem lehet egyenlő a végállomással, induló idő kisebb legyen mint az érkezési idő). Majd ha minden rendben van, beszúrunk az adatbázisba egy új járatot (Vonat egyedpéldányt), és még hozzá két Megáll rekordot a kiinduló és a végállomással. Mivel a Vonatnál az id auto\_increment, ennek lekéréséhez a mysqli\_insert\_id() függvényt használjuk. Ha sikeres a járat létrehozása, erről a felhasználónak is küldünk üzenetet.

Következik az **ujallomas.php** (itt lehet új állomást felvenni). Itt először 'Az adatbázisban megtalálható állomások' alcím alatt egy lekérdezés után kiíratjuk az állomásokat, majd az 'Új állomás felvétele a rendszerbe' alcím alatt a felhasználó itt is egy POST kérést küldhet a szerver felé. Ha minden jó, azaz nem volt még olyan állomás, amit a felhasználó létre akart hozni, akkor azt beírjuk az adatbázisba, majd 1,5 sec után az oldal frissül, és most már az éppen hozzáadott állomás is ott lesz a felsorolásban.

Az utolsó menüpont a Járatok módosítása (**modositas.php**). Itt első körben a szerver kilistázza az összes létező járatot (úgy, mint a kereso.php-ban, csak szűrés nélkül), hogy a felhasználó tudjon közülük választani, hogy melyiket akarja törölni vagy módosítani. Az egyes járatok mellett elhelyezett törlés és módosítás gombok pedig egy http GET kérés formájában egy-egy új oldalra visznek, ahol is az URL-ben lesz benne a kérdéses járat id-ja.

Ha egy járatot törölni akarunk, akkor a **jarattorol.php** oldalra jutunk. Ez az oldal a felhasználó számára nem is látható, mert egy headerként kiadott Location: utasítással visszaugrunk a modositas.php-ra, ha az oldal elvégezte a törlést. A törlés pedig a következőképpen zajlik: az oldal a kapott GET kérésből kinyeri a törölni kívánt járat id-ját. Majd először kitörli a hozzá kapcsolódó Megáll rekordokat, végül pedig magát a járatot.

Ha pedig módosítani akarunk egy járatot, akkor a **jaratmodosit.php** oldalra fog jutni a felhasználó. Itt is, mivel egy GET kéréssel jutunk ide, az URL-ben ott lesz a járat id-ja. Ez az oldal két funkcióval is bír. Először: kilistázzuk a járat megállóit, időrendben, táblázatszerűen. Majd a köztes megállók mellett ott lesz egy-egy input, hogy megváltoztathassuk, hogy mikor álljon meg ott a járat, ez egy POST kérést jelent. Így eljuthatunk a megallomodosit.php oldalra, aminek pedig egy GET kéréssel továbbítjuk az id-t. Emellett, minden köztes megálló mellett van egy töröl gomb is, így a megallotorol.php oldalra jutunk, az id itt is GET kéréssel jut el az állomáshoz. Az oldal másik funkciója pedig a következő: itt tudunk új köztes megállót hozzáadni a járhoz. Szokás szerint egy POST kérést dolgoz fel az oldal, ezután hibákat vizsgálunk (a megállási idő jó-e, nem áll –e meg még ott a vonat), majd ha minden jó, egy MySQL Query paranccsal beírjuk a megállót az adatbázisba, és legvégül frissül az oldal, és ugyanez az oldal jön be a már módosított értékekkel, hogy további módosításokhoz ne kelljen újra megkeresni a járatot.

A fentiekben leírtak szerint egy-egy megálló megállási idejének módosítása egy POST és egy GET kérést is küld a **megallomodosit.php** állomáshoz. Ezt az oldal feldolgozza, majd ha az új megállási idő az indulási és az érkezési idő közé esik, update-eljük az adott Megáll rekordot. Végül módosítás után azonnal visszaugrunk a jaratmodosit.php-ra, azzal a vonat id értékkel, amelyik járatval eddig is dolgoztunk.

Ha egy megállót törölünk, akkor pedig a **megallotorol.php**-re ugrunk. Itt is egy GET kéréssel kapjuk meg a törölni kívánt megállás vonatid-ját és állomásid-ját, az oldal kitörli az ennek megfelelő megáll rekordot, majd a módosításhoz hasonlóan visszaugrunk a jaratmodosit.php-ra. Ez egyébként azt is jelenti, hogy a felhasználó a megallomodosit.php-ból és a megallotorol.php-ból semmit se lát, hiszen azok meghívódás után gyorsan elvégzik a dolgukat, majd azonnal visszatérnek a hívó oldalhoz.