**Záródolgozat**

****

**Somodi Dániel**

**Szoftverfejlesztő és -tesztelő**

**5/13/A**

**Kecskeméti SZC  
Kandó Kálmán Technikum**

**2024**

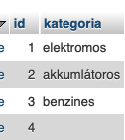
**Tartalomjegyzék**

**Témaválasztás**

Amiért ezt a témát választottam, az az, hogy apám xy év óta foglalkozik kerti gépek kölcsönzésével, így személyesen láthattam ennek fejlődését és hogy a közelünkbe élők mennyire használják ki ezt. És ennek hatására készítettem el egy weboldalt, hogy így minél többen halljanak az apám vállalkozásáról, és ennek egy átdolgozott, 2.0változatát készítettem el. Ami sokkal felhasználóbarátabb, egyszerűbb lett, és kapott egy külön keresőt is.

**Fejlesztői dokumentáció**

1. Adatbázis

A weboldal mögötti adatbázist én készítettem és frissítem ha érkezne hozzánk egy új gép vagy valami ár változás történt. Az adatbázisom két egyszerű táblából áll, az egyik egy sima kategória rész ahol mind a négy kategóriának van egy azonosítója. 

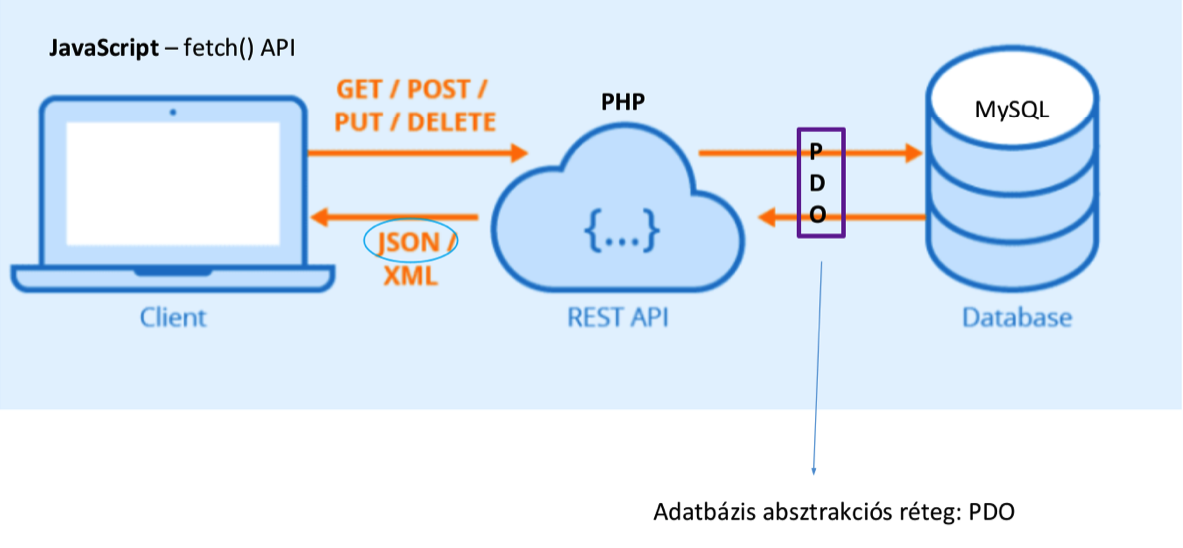
A második pedig maga ahol a gépek adatait tárolom mint pld.: gép megnevezése, leírása, bérleti díj.



a kettő tábla össze van kötve így egyszerűbb kezelni a kategória részt mint új adat létrehozásánál, mint lekérdezésnél.

A leiras részben minden sor után tettem egy “*<br>*”, hogy így biztosan új sorba kezdődjenek az adatok és rendezetebben jelenik meg így a weboldalon.

2. REST API

A REST API-k arra szolgálnak, hogy könnyen kommunikáljunk az interneten keresztül. Alapelveik közé tartozik az erőforrások használata, az egyszerű működés és az állapot nélküli kapcsolat. Amikor tervezünk egy REST API-t, fontos figyelembe venni az erőforrásokat, az URL-eket, a HTTP módszereket, a reprezentációkat és a státuszkódokat. Összességében a REST API-k segítenek a könnyű és hatékony internetes kapcsolattartásban.

3. Szerver oldali API

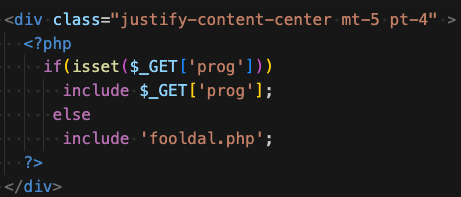
Szerveroldali API-k segítenek az alkalmazásoknak adatokat kérni és változtatni a szerveren. Két fő típusa van: RESTful és SOAP API-k.

Fontos a biztonság, gyorsaság és érthető dokumentáció. Az API-k segítik az alkalmazások közötti hatékony kommunikációt és adatcserét. Ezt alkalmaztam én is és jelenítettem meg a saját adatbázisomból weboldalra az adatokat.

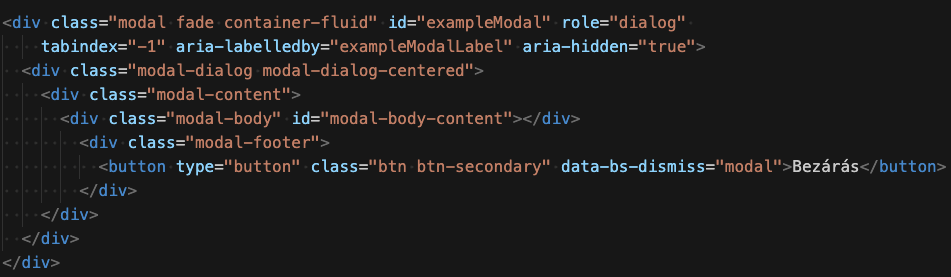


A fő oldalnál minden adatot kikértem az adatbázisból hisz tudtam, hogy minden kelleni fog a fő oldalamra.

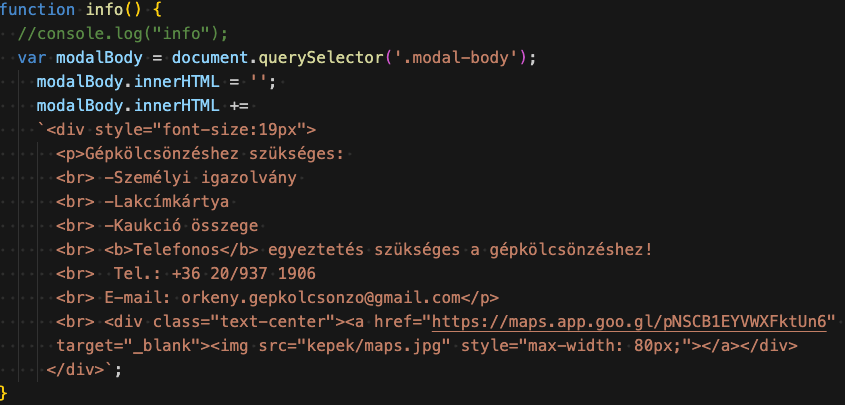
4. Fő keret

Az index.php kódom egy egyszerű navbar, footert tartalmaz, es egy felugró ablakot megvalósító modalt. Ezeket a [getbootstrap.com](http://getbootstrap.com) -ról használtam fel, és formáztam át tetszőlegesre. Ezek között van az a php kód rész ami megvalósítja h milyen adatok hogyan is néznek ki. 

A három fő elem közül kiemelném a modalt ami javascripten keresztül kétféle módon kerül a tartalma feltöltésre, attól függően hogy egy gépre kattint a felhasználó vagy az információs ikonra.

maga a *modal*: 

innen a *modal-body* osztály kerül feltöltésre minden kattintás után a megfelelő azonosítóval ellátott gép adataival, vagy az információs résszel.

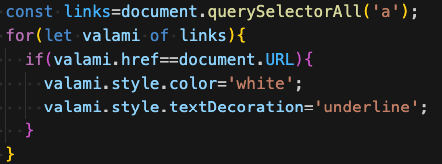
az információs részt feltöltő script:

Itt látható, hogy minden újbóli function meghívásnál, kiüríti a *modal-body* osztályt, hogy így biztosan csak a megadott adatok legyenek benne.

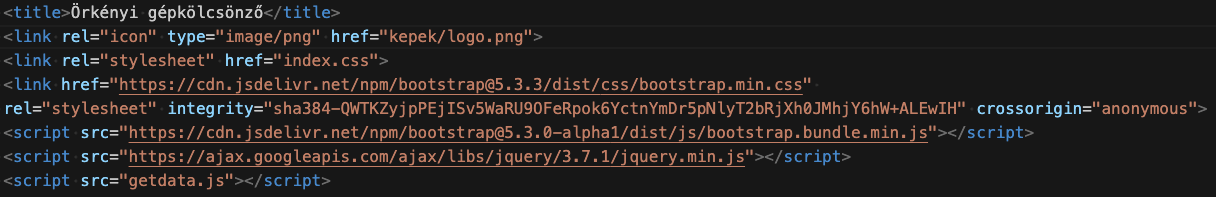
A navbar-omba megtalálható egy Főoldal menüpont és a három kategória menüpontja valamint az információs gomb, kereső oldalra átirányító gomb is.

navbar kód részlet:

ezeken egy kisebb stílus formázást tartalmaz mi azért felel, hogy az aktív menüpont elem aláhúzott és folytonosan fehér legyen

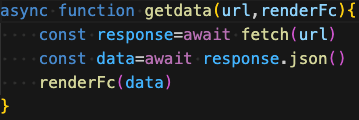
ezért felelős részlet:

5. Linkek, scriptek

Az egyik legfőbb része, hogy minden segítségeket használtam még amik segítették a formázást, vagy hogy egyáltalán fel tudjam dolgozni a gépek adatait

Itt használok két stílust, az egyik saját kezűleg kezelt, itt pár dolgot próbáltam ki, pár helyezést, vagy melyik stílusba nézne ki jobban, a másik pedig a [getbootstrap](https://getbootstrap.com/docs/5.3/getting-started/introduction/) oldalon található jelenleg legújabb 5.3as verzióját használtam. Itt állítottam át a böngésző fülön lévő ikont is.

Még három scriptet használtam fel, az egyik egy *async function* ami magába foglalja az adatok beolvasását.



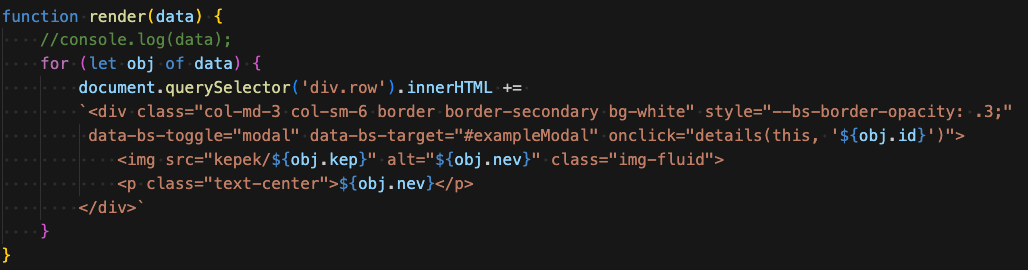
Van benne egy *ajax* script is ami felel a kereső működésért, a harmadik pedig is a bootstrap script-ét tartalmazza.

6. Főoldal

A fooldal.php -ba alkottam meg magát a fő részét az oldalnak ahol működik az adatok betöltése és kezelése, html csak egy minimális van benne, pont hogy egy *container-fluid* és egy *row* osztályt kapott két *div* objektum.

Maga a *javascript*-be kicsit több tartalom van viszont, itt használom fel a külön fájlba megírt *getdata function* -t, és olvasom be az annak megfelelő szerver mapa alatt lévő php fájlt amivel egy parancs során lehívom a kellő adatokat.

*getdata* használat közben:

Ezután minden adatot a *render function* -be felhasználtam, hogy aztán innen töltsem fel a weboldal tartalmát:

Itt egy kártya kinézetbe töltöttem és formáztam meg a gépek képeit és alattuk a gép neveit, majd *onclick* -el oldottam meg hogy a *modal* megfelelő helyen, azonosító alapján jó adatokkal legyenek feltöltve.

Itt látható, hogy bootstrap flexbe igazítottam, hogy fluid legyen és alkalmazkodjon a képernyőhöz.

Ezután írtam meg a *details* *function* -t, ami feltöltötte a modalt az azonosító alapján lévő adatokkal.

Itt is látható, hogy a *modal-body* osztályba raktam az adatokat, és hogy minden újra hívásnál törli az előző adatait.

7. Többi menüpont szortírozás

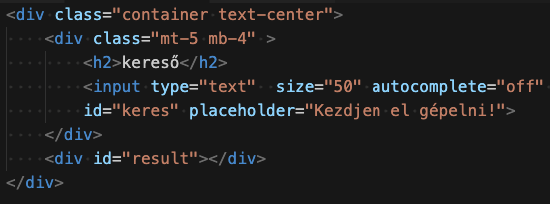
A fő oldal menüponton kívül ott van a három eltérő kategória ami mögött hasonló szerver oldali api van csak ott külön szűrés van a kategóriára pld.:

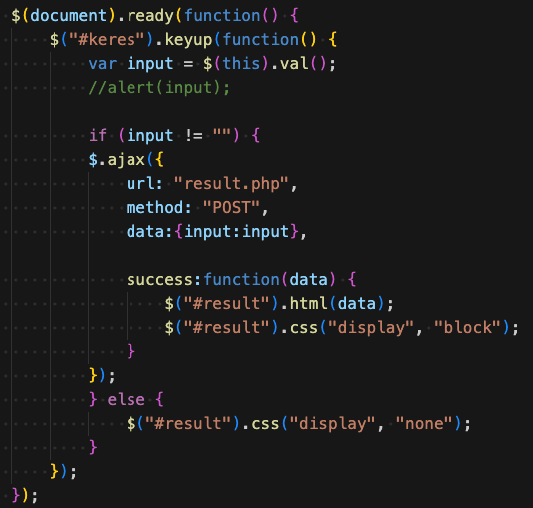
A feltöltés is hasonló csak a *getdata* a megfelelő szerver mappában lévő fájlt olvassa be.

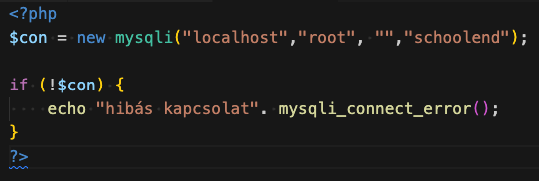
8. Kereső oldal

A kereső megalkotása kicsit hosszadalmasabb volt de szerencsére [WebsiteGeek](https://www.youtube.com/watch?v=Yggrlux69MQ&list=WL&index=5) videója alapján könnyebben sikerült megalkotni.

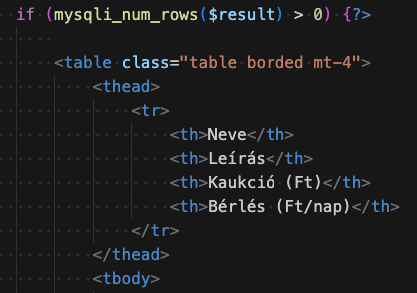
Ez egy egyszerű javascript kód, ami egy kereső mezőre reagál. Minden egyes billentyű lenyomásnál egy AJAX kérést indít egy PHP fájlra, hogy frissítse a keresési eredményeket. Ha a keresőmező üres, az eredmények elrejtődnek.

Itt látható egy egyszerű megoldás a kinézetre. és pár beállítás a mezőre: 

Ezután pedig jön a script mögé:

Itt pedig az össze kapcsolódás a szerver oldalról:

Ezek mellé még tartozik egy külön php fájl ami megjeleníti a megfelelő adatokat amikre sikeres a keresés:

Ez ellenőrzi, hogy megfelelőek-e a keresőbe található szókkal és azokat megjeleníti egy táblázatban:

Ha pedig valami elír a felhasználó vagy olyan adat nem található az adatbázisban azt pirosan kiírja.