Jegyzőkönyv

Webtechnológiák

Féléves feladat

**Készítette**: Garay Gabriel

**Neptunkód**: GJ2N7R

**Dátum**: 2023. 12. 09.

Tartalomjegyzék

[Feladat leírása 3](#_Toc153048214)

[Felhasznált technológiák 4](#_Toc153048215)

[HTML 5](#_Toc153048216)

[CSS 6](#_Toc153048217)

[Javascript, Jquery 7](#_Toc153048218)

[Ajax, Json 8](#_Toc153048219)

[Kiegészítés: PHP 9](#_Toc153048220)

# **Feladat leírása**

A feladat egy komplex nyilvántartó rendszer készítése tetszőleges témakörben. Lehessen a nyilvántartott termékeket kilistázni. Termék felvitele előtt legyen egy ellenőrzés, hogy minden adat helyes-e.

A cél igényes stílus kialakítása, és a kurzus keretein belül tanult technológiák használata.

# **Felhasznált** **technológiák**

A kötelezően felhasznált technológiák a **HTML, CSS, Javascript** és **Jquery, Ajax** és **Json.**

Ezen felül egy problémát PHP kóddal oldottam meg, amit részletesen kifejtek a későbbiekben.

## **HTML**

A weboldal struktúrájának és elemeinek megvalósításához a HTML jelölő nyelvet használtam. Három aloldal található a weboldalon belül. Ezek sok azonos elemmel rendelkeznek:

* mindnek van egy fejléce a weboldal tetején a címmel és a három lista elemmel, melyek az aloldalakra mutatnak,
* két oldalt egy-egy, mesterséges intelligenciával generált kozmetikai kép,
* középen a tartalomnak kihelyezett lekerekített blokk,
* az oldalak alján a lábléc

A három aloldal a következő:

* A weboldal leírása (*index.html*)
  + Egy gyors áttekintés az oldal működéséről és a navigációs elemekről.
* A termékek listázása (*list.html*)
  + A már bevitt termékek listázására van lehetőség, egyszerre egy termék megjelenítésével. Az éppen kilistázott termék törlése és módosítása egy kiegészítő funkció, utóbbi ellátva formátumkezeléssel.
* Termék hozzáadása (*add.html*)
  + Új termék hozzáadása típus- és formátumkezeléssel ellátva. Hozzáadáskor egy mező sem maradhat üresen. **Termék neve**, **gyártó** és a **típus** lehet bármilyen típusú. A **szavatosság**nak csak ’YYYY-MM-DD’ bemenet lehetséges, illetve a **készletmennyiség**, **ár** és **kedvezményes ár** mezők szigorúan csak számot fogadnak el.

## **CSS**

Ezzel a technológiával a weboldal kinézetét alakítottam. Az összes aloldal egy **style.css** nevű fájlból kapja a kinézetét.

Ez alól kivétel a fejléc menüpontjainak **animáció**ja, mely minden aloldal esetén egy saját CSS fájlból kerül megvalósításra. Az animációval az éppen megnyitott aloldalhoz tartozó menüpont szövege egy fél másodpercre megnagyobbodik.

## **Javascript, Jquery**

**Javascript** valósítja meg az oldalak funkcióit. Ennek egyszerűsítésére használatos a **Jquery** library, hiszen megkönnyíti az eseménykezelést, dom elemek manipulációját, animációt és az ajax használatát.

Két helyen van jelentőssége a projekt szempontjából:

* Termékek listázása:
  + Az oldal megnyitásakor beolvassa a **products.json** fájlt, amiben a termékek vannak tárolva.
  + Termékek között a nyilakkal lehet navigálni.
  + Ezen az oldalon van lehetőség az épp megjelenített termék törlésére és módosítására is. A törlés egy gombnyomással történik, de a módosítás ennél bonyolultabb. Első lenyomásnál válnak a mezők módosíthatóvá, a gomb pedig a ’Mentés’ szövegre vált. Amennyiben a módosítás utáni adatok megfelelnek a kritériumoknak (a típus megfelelő, és nem maradt üresen egy mező sem), a mentés gomb megnyomásával módosítja a json fájlt.
* Termék hozzáadása:
  + Egy **űrlap**ban található táblázat mezőinek kitöltése után (nem maradhat üres mező) a ’Hozzáadás’ gombbal történik az adatok mentése Json fájlba.
  + Üres mező esetén a program figyelmeztet a bemeneti mezőben, hogy nem maradhat üres. Helytelen típusú adatnál pedig felugrik egy ablak, amiben a ezzel kapcsolatban figyelmeztet. Utóbbi kettőnél a helytelenül megadott adat mezője piros színű keretet is ölt.

## **Ajax, Json**

Az **Ajax** hívások lehetővé teszik adatok olvasását és küldését anélkül, hogy frissíteni kéne az oldalt. A **Json** pedig objektumok tárolására alkalmas.

A Json-t a javascriptnél említett módon használtam, a termékek tárolására objektumokként.

Ezt a fájlt Ajax-szal frissítettem egy termék adatainak módosításánál, törlésénél, illetve új termék bevitelekor is ezzel továbbítottam az **űrlap**ból kapott adatokat a Json fájlnak.

## **Kiegészítés: PHP**

A **PHP** egy szerveroldali programozó nyelv, mellyel szerver-oldali funkciókat tudunk végezni. Ebben az esetben az Ajax minden **POST** (módosítás, hozzáadás, törlés) kérést egy php fájllal egészítettem ki, mivel szervert használtam a fájlok tárolására és a weboldal megvalósítására.

Termék felvitelekor Ajax **POST** kéréssel meghívtam a **process.php** fájlt, amiben az elküldött mezők adatait vizsgáltam meg. Amennyiben az egyik mező üres, hibát jelez vissza az ajaxnak, ezt pedig javascripttel megjelenítettem a felhasználónak az **űrlap**on.

Termék módosításnál - amennyiben a bevitt adatok helyesek – és törlésnél az ajax **POST** kérés a **modify.php** fájllal küldi tovább a szerverre a módosított Json fájlt.