Nagy Konrád Márk

q1yheo | 2024 május 1.

Jegyzőkönyv

Webtechnológia alapjai | Munkahelyi adatbázis

**Tartalomjegyzék**

* 1. index.html
* 2. style.css
* 3. navbar.html
* 4. user-list.html
* 5. user-form.html
* 6. car-list.html
* 7. car-form.html
* 8. stats.html
* 9. workplaceSafety.html
* 10. products.json
* 11. ajax.js
* 12. video.js
* 13. main.js
* 14. index.js

**Bevezetés**

A feladat leírása:

* Én egy munkahelyi adatbázisnak a weboldalát terveztem meg, ami több dologra is képes. Először is van egy navigációs fül, ahol lehet a különböző html oldalak között ugrálni. Az alapértelmezett oldalon a munkahelyi dolgozók jelennek meg táblázatos formában. Az „Új felhasználó” oldalon egy űrlap jelenik meg, aminek segítségével tudjuk bővíteni a dolgozók listáját. Van egy „Céges autók listája” oldal is, ahol hasonló módon, mint az első oldalon, céges autók vannak táblázatosan nyilvántartva. Plusz funkció az oldalon a videó és az azt vezérlő gombok. Az „Új céges autó” fül hasonlóan egy űrlap, mint már arról írtam. A „Statisztika” menüpont egy új oldalra visz, melyben statisztikai kimutatásokat láthatunk különféle szempontok alapján. A „Munkavédelmi szabályzat” fül pedig felhozza a json file-ból kinyert adatokat táblázatos formába.

1. Feladat: index.html
   1. Ezen az oldalon van a szíve a projektnek. A head részben található a jquery importálása és a css file is. Az ajax számára is itt vannak a scriptek.
   2. A body részben van a weboldal elején látható hullámzó animáció kódja.
2. Feladat: style.css
   1. Itt található minden kozmetikai formázás és animáció. Az előbb említett hullámzás itt kel életre.
3. Feladat: navbar.html
   1. Ez egy amolyan elosztó oldal. Itt van beállítva, hogy mely oldalra menjen a felhasználó tovább, ha rákattint egy menü pontra. Ezek szépítését kódon belüli css-el oldottam meg. Például a „Céges autók listája” gombra kattintva átkerülünk az adott oldalra.
4. Feladat: user-list.html
   1. Ez az alapértelmezetten megjelenő oldal. Itt látható a táblázat, melyben a munkavállalók adatai vannak. Ennek a mikéntjét majd a js fileoknál fogom taglalni.
5. Feladat: user-form.html
   1. Ez egy űrlap, melynek segítségével adatokat tudunk felvinni a user-list.html-be. Található itt szöveges beviteli mező, rádió gomb, legördülő lista és checkbox is.
6. Feladat: car-list.html
   1. Ez a céges autóknak egy táblázat. Hasonló, mint a munkavállalók táblázata. Extraként ezen az oldalon található a videó, melyet gombok segítségével lehet vezérelni. Ellehet indítani, megállítani és hasonlók.
7. Feladat: car-form.html
   1. Ez lényegében ugyanaz, mint a másik űrlap, csak az autókat lehet vele bővíteni.
8. Feladat: stats.html
   1. Ezen az oldalon különféle statisztikai adatokat jelenítek meg diagrammok segítségével.
9. Feladat workplaceSafety.html
   1. Ez egy munkavédelmi szabályzat, amely json fileból került be a táblázatba ajax segítségével. Írtam rá egy fetch kódót is, ami egy alternatív megoldása a feladatnak. Ez nincs felhasználva a végső változatban, de szívesen megmutatom tanárúrnak.
10. Feladat: products.json
    1. Ez a kis adatbázis adja a „Munkavédelmi szabályzat”-ban található táblázat tartalmát.
11. Feladat: ajax.js
    1. Itt a nagyon egyszerű módja található json file olvasásra és felhasználásra. Igazából egy for ciklus segítségével végig megyünk az adatokon és beírjuk őket a megfelelő helyre.
12. Feladat: video.js
    1. Itt találhatóak a függvények, melyek a videó manipulálását teszik lehetővé.
13. Feladat: main.js
    1. A fő mozgatórugója a projektnek. Menjünk lépésről lépésre:
       1. Van kettő változó tömb, melyben a táblázatokhoz szükséges adatok vannak eltárolva. Ez a users és cars tömb.
       2. Ezután következik az initialize() funkció. Ez a navbarból betölti a user-listet. Ez a főképernyő, ami indításkor megjelenik. Ilyenkor egy kis animáció is lejátszódik, mely segítségével a menüpontok kicsiről nagyra nőnek.
       3. loadUserList(): Ennek segítségével feltöltjük egy for ciklus segítségével a munkavállalók táblázatát.
       4. loadCarList(): Ez pedig a Céges autók táblázatát tölti fel.
       5. loadUserForm(): Ez az űrlapot hozza elő, amely segítségével új felhasználókat adhatunk a munkavállalókos táblázathoz.
       6. loadCarForm(): Az autós űrlapot hozza elő.
       7. loadWorkplaceSafety(): A munkavédelmis lapot hozza be.
       8. loadStats(): A statisztikai oldalt hozza be. Itt vannak legenerálva a diagrammok.
       9. checkUser(): Ennek segítségével tudunk hibát kiíratni, ha rossz értéket vittünk be a munkavállalós űrlapon. Először konstansok láthatók, amelyek az ellenőrizni kívánt mezőt jelzik. Van egy logikai változó, ami alapból igazra van állítva. Kezdődhet a vizsgálat. If segítségével megnézem, hogy az adott kritériumnak megfelel-e a kapott beviteli érték, ha nem, abban az esetben a logikai változó értéke hamisra változik, a beviteli mező kerete piros lesz és kiír egy hibaüzenetet. Miután az összes ellenőrzés megtörtént, a függvény visszatérési értéke a logikai változó értéke lesz.
       10. saveUser(): E függvény segítségével menthető el az űrlap tartalma a táblázatba. Először van a checkUser() segítségével ellenőrzi, hogy van-e hiba, ha van, akkor addig nem ment, amíg fenn áll. Ezután push-olja az adatokat a megfelelő helyre. A legvégén pedig a átdob minket a user-list oldalra, ahol egyből láthatjuk munkánk eredményét.
       11. checkCar(): Ugyanaz mint a checkUser(), csak az autós űrlapra.
       12. saveCar(): Ugyanaz mint a saveUser(). csak az autós oldalra.
       13. Végül pedig van egy initialize() függvény.
14. Feladat: index.js
    1. Ennek segítségével tudunk egy saját port-ot létrehozni.