JEGYZŐKÖNYV

Web technológiák 1

Féléves feladat/beadandó

Weblap készítés

Készítette: **Simkó Levente**

Neptunkód: **YG5YAO**

Dátum: **2024. 01. 24.**

**Feladat:**

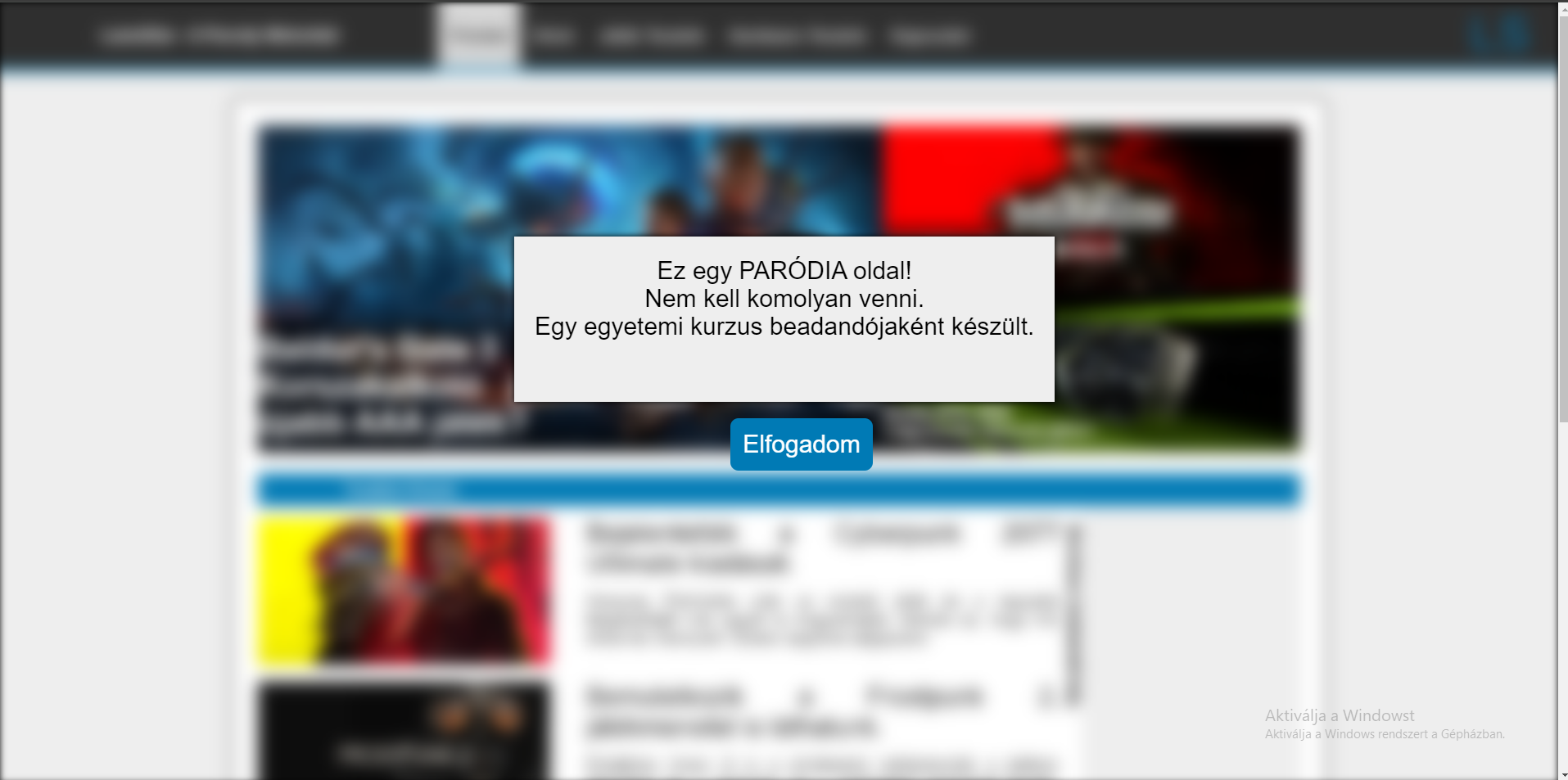
Mivel nem jutott semmi olyan kifejezetten érdekes dolog az eszembe amit egy weblapra fel lehetne tenni, így úgy döntöttem, csinálok egy weboldalt ami egy már meglévő weblapot parodizál ki. Leszögezném semmi bajom nincs a weblappal, sőtt. Az általam választott weblap egy ismertebb gamereknek szóló weblap, ahol játékokkal, számítógépekkel és egyéb IT témában jelennek meg cikkek, legyen az hír, teszt, stb. Dönthettem volna úgy, hogy egyszerűen kimásolom a kódját a weboldalnak és úgy újracsinálom minimális változtatással, de annak semmi értelme nem lenne mert nem lenne elfogadva, mint saját weboldal és nem is gyakorlok a fejlesztéssel, így hát úgy döntöttem, hogy kisebb-nagyobb eltéréseket belerakva igyekszem újra kreálni a weboldalt, úgy hogy saját dolgokat rakok bele, viszont a paródia miatt bizonyos dolgokat meg megtartok.

**Minden weboldal:**

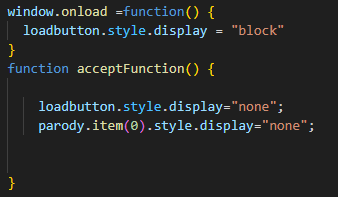
Minden oldalon található egy navigációs menü és egy gomb amin a ha lentekerünk a weboldalon akkor jelenik meg és az adott weboldal tetejére vissza görget ha rányomunk.

**Főoldal:**

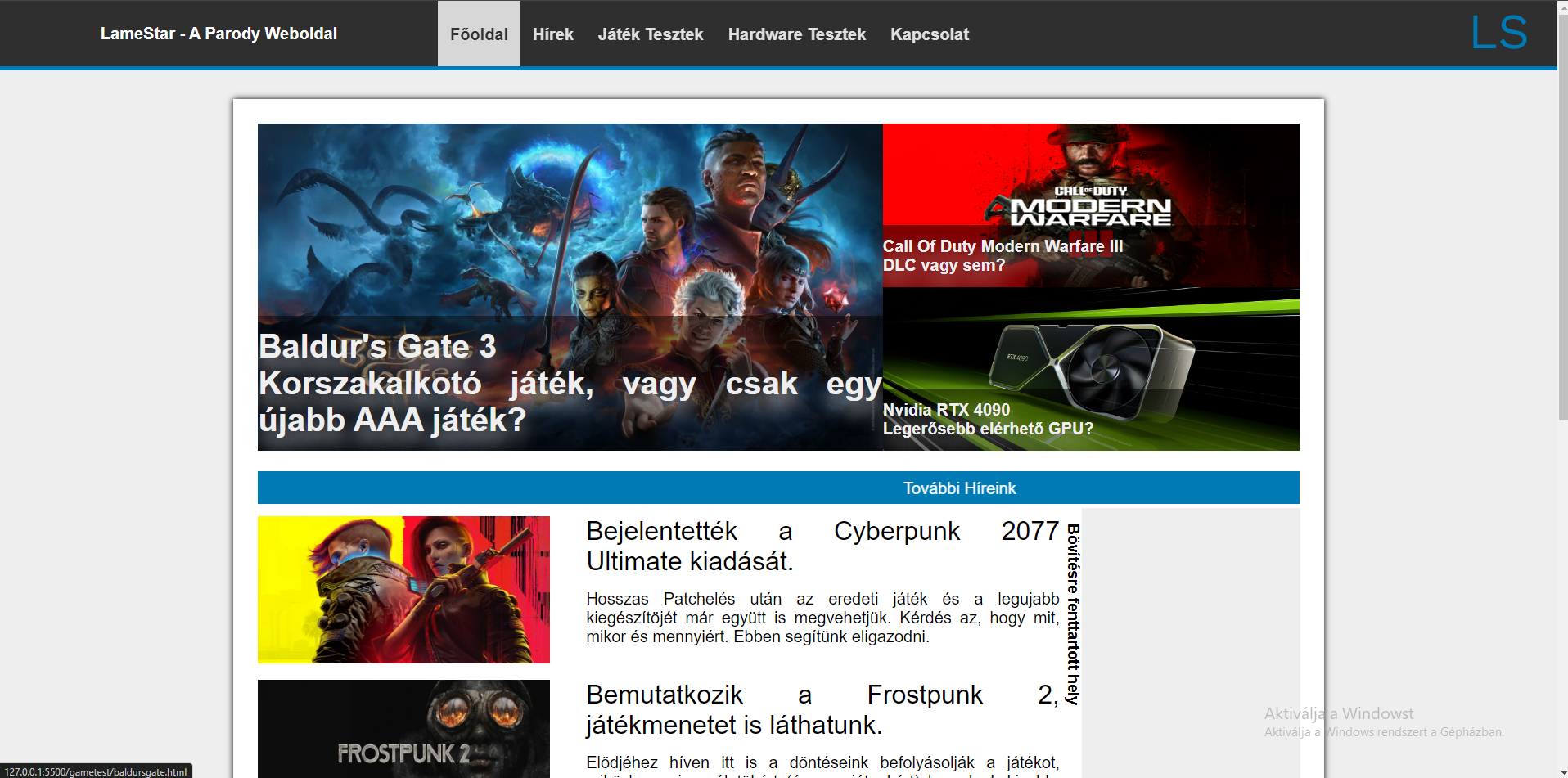
Mivel ez egy paródia, így alapból azzal találkozik szembe az ember mikor a főoldalra betölt, hogy erre figyelmeztető rész felugrik, és a háttér nem olvasható egészen addig míg az elfogad gombra rá nem kattint. Erre az ötletet a sok helyen elérhető adblock kikapcsolását kérő ablakok adták az ötletet.



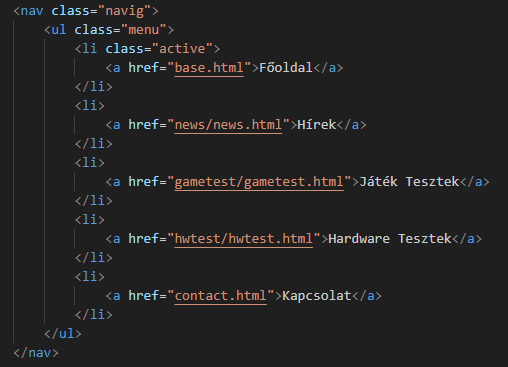
Fontos, hogy ez minden egyes alkalommal előjön a főoldalon, mert nem tárolok el semmit sütikben. A az elfogad gombra nyomva ez megszűnik és olvasható válik a weboldal. Ezt úgy érem el, hogy amikor betölt az oldal megjelenik a elfogad gomb illetve a parody div ami a blurt és a felhívást tartalmazza, majd az elfogad gombra rá van kötve egy script ami kattintásra lefut. Ekkor a gombot és a parody div-et elrejti a felhasználó elől és a weboldal olvashatóvá válik. Ezért a lentebbi kódrészlet felel, az viszont itt nem látható, hogy lekéri a weblapról az adott class-al vagy id-val rendelkező elemeket és azt elnevezem változóként.

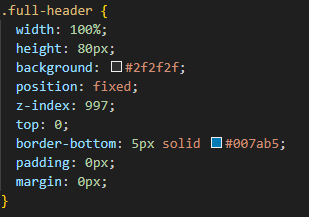


Ha elfogadtuk a weboldalon található figyelmeztetést akkor az alábbi kép jön velünk szembe. Itt már a felület olvasható és a linkek is kattinthatóvá válnak a weblapon.

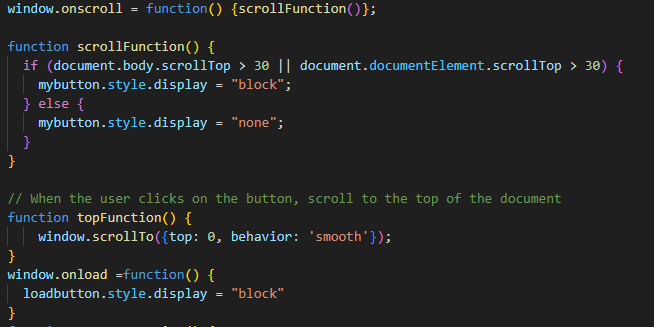


Haladjuk fentről lefelé a weboldalon. Első dolog ami szembe jön velünk az a menü rendszer. Itt választhat a felhasználó milyen típusú híreket szeretne olvasni, vagy merre szeretne navigálni. Az egészért alapvetően egy táblázat felel, aminek egy sora és 3 oszlopa van. A linkek amik a többi weboldalra visznek a menüben pedig egy nem sorba rendezett lista felel. Ezt CSS-sel módosítva kapjuk meg azt a kezelőfelületet ami látható fent is. Fontosnak tartom kitérni arra, hogy attól függően melyik navigálás menüt választja az ember az más színnel jelenik meg mint a többi. Ezt egy active nevű class-sal érem el, ugyanis fontosnak tartottam azt, hogy a felhasználónak valamilyen visszajelzés legyen. Ezen felül ha valamelyik linkre ráviszem az egeret akkor annak megváltozik a színe, fehérről kékre. A lentebb található kép egy részlet a menüért felelő kód részletből. Ezt követő kép pedig a CSS-sel fog foglalkozni szintén a menün belül, ugyanis ez a menü mindig a weboldal tetején található annak érdekében, hogy ne kelljen mindig felgörgetni a weblapon, ha navigálni szeretnénk azon.



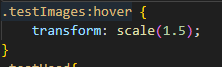
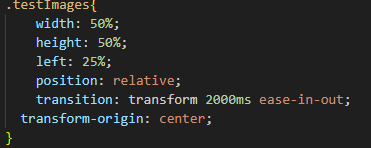


Ezt követően a weboldal tényleges része következik. Ez tartalmazza a híreket. Legfelül van a kiemelt hír, ami két részből áll, az egyik a főhír a bal oldalon, a jobb oldalon két kisebb hír található. Ezeknél a link kicsit animálva van, így felhívja a figyelmet azokra. (Legalábbis Chromeon működik az animáció.) Lentebb haladva találjuk a további híreket a weboldalon. Ezek közül egyikhez se kötődik rendes weboldal, csak a elrendezés mutatják be. Tőle jobbra pedig egy további bővítési lehetőségeknek fentartott hely található. A gombért ami a weboldal tetejére görget az alábbi kódsor felel.



**Tesztek:**

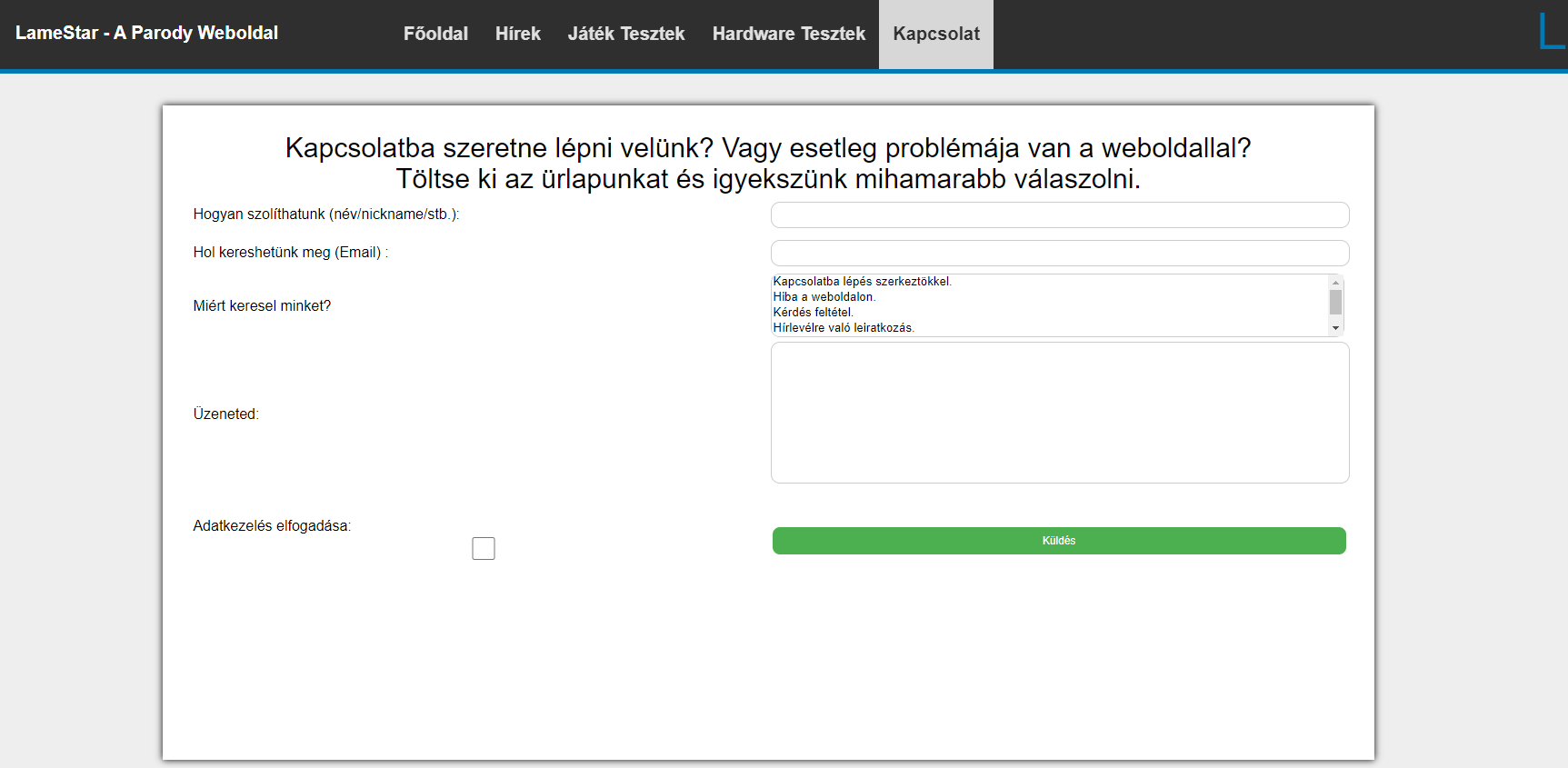
Mivel a navigációs menüben választható Hírek és a két tesztekkel foglalkozó oldal hasonló. így azokat nem részletezném. A tesztekkel foglalkozó tényleges weboldalak viszont érdekesek egy picit, ugyanis azokban egy adott classal ellátott képek újraméretezik magukat, ha valaki ráviszi az egeret.



Ezekért a testImage classban található transition felel ami meg a hover esetén transformot hajtja végre, így a képet 1.5-szeresére növeli.

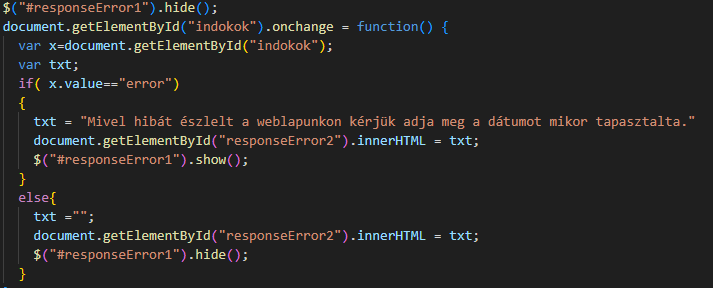
Mivel nem volt időm teljes teszteket írni és kedvem se így random töltöszöveg van benne.

**Kapcsolat:**

****

Formkezelésre azt találtam ki, mi lenne, ha egy olyat csinálnék, hogy kapcsolatba tudsz kerülni a weboldal fejlesztőivel/szerkesztőjével. Mivel vannak olyan weboldalak ahol nincs publikusan kiírva email cím, hanem egy űrlapot használnak arra, hogy a felhasználó kapcsolatba tudjon kerülni az üzemeltetőkkel, így ezt oldottam meg. Ezen a felületen lehetőség van, megadni hogyan szólítsuk az embert valamint, hogy mi az email címe amivel kapcsolatba tudunk vele lépni és hogy mi miatt szeretne kapcsolatba lépni és üzenetet is írnia kell mert senki se szeretni azt, hogy például problémát jelentenek be a weboldallal kapcsolatban de azt, hogy pontosan mi az azt nem lehet tudni.

Minden egyes mezőben meg kell jelölni valamit, ellenkező esetben a weboldal figyelmeztetést dob fel a felhasználó felé. Ha a felhasználó a hiba a weboldalon menüpontot választja akkor weblap megkéri, hogy válasszon dátumot is mikor érzékelte a hibát, ezzel lehetőséget adva a fejlesztőknek, hogy tudják követni, ha egy hibát kijavítanak, akkor az későbbiekben előfordul-e valami más miatt.



$("#validate").click(function () {

var nameValid= false;

var emailValid= false;

var messageValid= false;

var reasonValid = false;

var contactName=$("#contactName").val();

var gdpr = document.getElementById("GDPR");

var contactEmail=$("#contactEmail").val();

var contactMessage=$("#contactMessage").val();

var reason =$("#indokok").val();

$("#contactName").css('border','1px solid #ccc');

$("#contactEmail").css('border','1px solid #ccc');

$("#contactMessage").css('border','1px solid #ccc');

$("#indokok").css('border','1px solid #ccc');

if(!gdpr.checked){

alert("Fogadja el a a Adatkezeést!");

return;

}

if(contactName==""){

alert("Név megadása kötelezö.");

$("#contactName").css('border','2px solid #ff0000');

return;

}

else{

nameValid= true;

}

if(reason==""){

alert("Adjon meg indokot.");

$("#indokok").css('border','2px solid #ff0000');

return;

}

else{

reasonValid=true;

}

if(contactEmail==""){

alert("Kapcsolatba lépési emailcím megadása kötelező.");

$("#contactEmail").css('border','2px solid #ff0000');

return;

}

else{

emailValid=true;

}

if(contactMessage==""){

alert("Nincs üzenet.");

$("#contactMessage").css('border','2px solid #ff0000');

return;

}

else{

messageValid=true;

}

if(nameValid && reasonValid && emailValid && messageValid && gdpr.checked) {

var txt;

if (confirm("Biztos el akarja küldeni?")) {

txt = "Elküldtük az üzenetét.";

} else {

txt = "Semmi nem történt.";

}

document.getElementById("response").innerHTML = txt;

}

});

Mint a fenti kód részletből látszik (tudom nem szép de ha képet csinálok rosszabb lenne) ellenőrzöm, hogy valamilyen értéket adott meg a felhasználó és ha minden részhez tartozik érték akkor engedem csak “elküldeni” azt.