**1. beadandó feladat (2022.12.16 23.55-ig)**

Ebben a beadandó feladatban egy statikus weboldalt kell készítenie, a webalkalmazás több oldalból áll, dinamikus eszközök és külső könyvtárak közül csak a megengedett eszközöket használhatja fel. A feladat célja egy bemutatkozó oldal elkészítése, valamint egy oldal készítése minta alapján (szerkezet, tagolás, stílusjegyek megválasztása).  A beadandó témája egy bemutatkozó oldal elkészítése.

* Technológiák: HTML5, CSS3, Bootstrap 5

A beadandó feladat megoldása során a stílusra és szerkezetre vonatkozóan ahol nincs külön leírás, ott az elvárás, hogy az oldal szerkezete logikus és a szabványnak megfelelő legyen kialakítva, a stílus pedig esztétikus, valamint jól strukturált. Ezeknek a feladatoknak a megoldása során Bootstrap-et használhat.

A lenti oldalak legyenek egymásból elérhetőek menüpontokon keresztül. Az oldalnak reszponzívaknak kell lenni! A beadandó feladatban az alábbi részeket kell teljesíteni:

1. Önéletrajz oldal:
   1. Szakmai önéletrajz az elvárt formai követelményeknek megfelelően (iskolák felsorolva, pontokba szedve stb.), egy kép beszúrása. A szakmai önéletrajz alrészei: alapadatok, szakmai tapasztalat, iskolák, nyelvismeret stb. Az egyes alrészekhez ezen az oldalon könyvjelzők segítségével el lehet ugrani.
   2. Az oldalról pdf formátumban legyen letölthető az elkészített europass önéletrajzuk. Ha nincs, akkor el kell készíteni: <https://europass.hu/europass/docs/1>
   3. Az oldalon legyen olyan link is, amely segítségével e-mailt lehet küldeni Önöknek küldeni az alapértelmezett levelezőből e-mail cím beírása nélkül. A levél tárgya legyen: Webprogramozás I.
2. Órarend oldal: Az egyetemi órarend táblázatos formában felírva (nappali esetén az egy heti, levelező esetén a féléves órarend). Távoktatási tagozaton itt táblázatos formában azt kell megjeleníteni, hogy az aktuális félévben milyen tárgyakat vettek fel (tárgykód, tárgynév, kreditérték) és a táblázatban legalsó sorában legyen összegezve a felvett kreditek száma.
3. Naptár oldal: 2022. december hónap havi nézetét kell felépíteni (pl. Windows naptár havi nézet), az első és utolsó héten átcsúszó napok betűszíne legyen piros, háttérszíne szürke, ezzel is jelezve, hogy ezek a napok már nem képezik részét a hónapnak.
4. Képgaléria: Kategorizált képgaléria, amely legalább 20 képet tartalmaz, legalább 2 kategóriára bontva. Minden képnek van címe, sorszáma és miniatűrje. A galériában arra is legyen lehetőség, hogy egyszerre az összes képet megtekintsük. Elvárás az elegáns megjelenítés. JavaScript képnézegetőt nem lehet alkalmazni! (Nem különszedve 2/3 html oldalra - JS vagy CSS segítségével külön megjelenítés)
5. Videógyűjtemény: Videók gyűjteményét tartalmazó oldal. Elvárás, hogy minden videó lejátszható legyen helyben, tartozzon hozzá cím és leírás (pl., mint a YouTube lista). Egy oldalon 5 videó jelenjen meg és legalább két oldalból álljon (lapozási funkció megvalósítása). Az elemek lapozása statikusan történik, dinamikus eszközöket nem lehet használni. Youtube videók esetén használható beágyazás az egyes videóknál!
6. Sakktábla oldal: Ez a feladatrész a mintának való HTML és CSS felépítést vizsgálja, így ebben a feladatban az általános kereten (fejléc, menüsor, lábléc) kívül, azaz a konkrét tartalmi részben nem használhat Bootstrap elemeket. A méreteknek arányaiban kell megegyezni, a színek esetén pedig a mintának megfelelőket kell használni (esetlegesen Gimp vagy Paint programokkal vizsgálható a színkód). A feladat lényege a minta lehetőség szerinti pontos lemásolása lenne, így ennek megfelelően kapnak majd pontot.



Az értékelés az alábbi szempontok alapján valósul meg:

* A HTML oldal szerkezete és felépítése: A javítás előfeltétele a szabványnak megfelelő oldal készítése (HTML Validátor nem ad vissza Error típusú eredményt). Ebben a pontban az egyes logikai egységek kialakítása, kezelése, egymásba ágyazása kerül vizsgálatra.
* CSS szerkezete és felépítése: A javítás előfeltétele a szabványnak megfelelő stíluslap készítése (CSS Validátor nem ad vissza Error típusú eredményt). Ebben a pontban az egyes logikai egységek kialakítása, kezelése, egymáshoz való viszonya kerül vizsgálatra, valamint, hogy mennyire sikerült egységes megjelenítést meghatározni és ezt mennyire duplikációmentes, átlátható kódbázissal. A vizsgálat kiterjed arra is, hogy mennyire újrahasznosítható és módosítható könnyen a kódbázis.
* Egységek szétválasztása: A HTML, CSS  állományok jól elkülöníthetőek egymástól, a készítettek kódbázisok külön-külön újrahasznosíthatóak.
* Bootstrap használata: Boostrap professzionális alkalmazása, az oldalak reszponzívak, Bootstrap használata helyes, a Bootstrap szabályainak felülírása helyesen történik meg.

**2. beadandó feladat (2022.12.16 23.55-ig)**

Ebben a beadandó feladatban egy API szolgáltatással kell dolgoznia. Az API-val json adatformátumban történik az adatcsere, az Ön feladata, hogy az API-ban biztosított szolgáltatások web alkalmazáson keresztül használhatóak legyenek valamint, hogy a weboldal az eredményt feldolgozza és visszajelzést adjon a felhasználónak a leírásnak megfelelően. A megoldás során AJAX kéréseket használunk (XMLHttpRequest és fetch egyaránt elfogadott, de figyeljünk a hibakezelésre).

* API elérése: <https://reqres.in/api/users>
* API dokumentációja (paraméterezés, visszatérés, elérés): <https://reqres.in/>
* Kezelendő http metódusok: GET, POST, PUT vagy PATCH (választható), DELETE
* Technológiák: HTML5, CSS3, JavaScript (ECMAScript 2020), Bootstrap 5 (opcionális)

A beadandó feladat megoldása során a stílusra és szerkezetre vonatkozóan nincs külön leírás, egyetlen elvárás, hogy az oldal szerkezete logikus és a szabványnak megfelelő legyen, a stílus pedig esztétikus, valamint jól strukturált. A feladat megoldása során Bootstrap használata nem kötelező, de használható.

1. Az oldal indulásakor kérjük le a felhasználók listáját, és jelenítsük meg azt táblázatos formában (jelenjen meg minden adata a felhasználónak (teljes név, azonosító), illetve az avatarja képképként). Minden sor végén készítsünk egy üres oszlopot is, ide fognak kerülni a funkciógombok (törlés, frissítés, stb.). Ügyeljünk arra, hogy amíg nem állnak készen adatok, addig is tájékoztassuk a felhasználót, hogy az adatok lekérése folyamatban van és várakozni kell.
2. Figyeljünk arra, hogy az eredmény több oldalra bontva jelenik meg, így legyen lapozható a listánk. Megoldandó feladat, hogy tudjuk változtatni az egy oldalon megjelenő elemek számát is. Nyilván ekkor a listák mérete és az oldalak száma is dinamikusan változni fog.
3. Egy-egy felhasználó teljes nevére kattintva a listából jelenjen meg a kiválasztott felhasználó összes adata (egyedi GET kérés alapján) névjegykártya formátumban a lista alatt. Kezeljük azt az esetet is, ha már nem létezik a felhasználó.
4. A funkció gombot között helyezzünk el egy Törlés gombot, amely megfelelő paraméterrel meghívja a DELETE metódust, és felugró ablakban tájékoztat a sikeres végrehajtásról (státuszkód alapján tudjuk eldönteni). Amennyiben a végrehajtás sikerül töltsük újra a listát (*a törlés igazából nem fog megtörténni, csak a kód alapján tudjuk, hogy helyes volt*).
5. A funkció gombot között helyezzünk el egy Frissítés gombot, amely megjelenít egy egyszerű űrlapot (akár felugró ablakban), ahol a felhasználó tudja módosítani a szerkeszthető adatokat, majd megerősítés után küldjük el a PUT/PATCH kérést (csak az egyikkel kell megoldani). Ha sikerült a frissítés (státuszkód alapján) erről előugró ablakban tájékoztassuk a kezelőt, majd töltsük újra a felhasználói listát (*a frissítés igazából nem fog megtörténni, csak a kód alapján tudjuk, hogy helyes volt*).
6. Az oldalon legyen lehetőség új felhasználó hozzáadására is, ehhez a táblázat utolsó sora legyen egy megfelelő űrlap (szerkeszthető mezőkkel) és a funkciógombok oszlopában legyen egy mentés gomb, amely beküldi a megfelelő adatokat. Az adatbeküldés előtt hajtson végre ellenőrzést:
   1. A felhasználó neve:
      1. csak az angol ábécé kis és nagybetűit tartalmazhatja, valamint számjegyeket
      2. legalább 5 karakter,
      3. első karaktere csak betű lehet
   2. Munkakör:
      1. csak az angol ábécé kisbetűit tartalmazhatja, valamint szóköz karaktert
      2. legalább 5 karakter
      3. első karakter csak betű lehet

Sikeres végrehajtás esetén felugró ablakba írjuk ki a létrehozás idejét és a felhasználó azonosítóját (amiket válaszban kapunk) és töltsük vissza az adatokat. Hiba esetén logoljuk a hibát a konzolra.

Az értékelés az alábbi szempontok alapján valósul meg:

* A HTML oldal szerkezete és felépítése: A javítás előfeltétele a szabványnak megfelelő oldal készítése (HTML validátor nem ad vissza Error típusú eredményt). Ebben a pontban az egyes logikai egységek kialakítása, kezelése, egymásba ágyazása kerül vizsgálatra, valamint hogy a kialakított szerkezetek mennyire támogatják a CSS-el való formázást, valamint a JavaScriptből való DOM műveleteket.
* CSS szerkezete és felépítése: A javítás előfeltétele a szabványnak megfelelő stíluslap készítése (CSS variátor nem ad vissza Error típusú eredményt). Ebben a pontban az egyes logikai egységek kialakítása, kezelése, egymáshoz való viszonya kerül vizsgálatra, valamint, hogy mennyire sikerült egységes megjelenítést meghatározni és ezt mennyire duplikációmentes, átlátható kódbázissal. A vizsgálat kiterjed arra is, hogy mennyire újrahasznosítható és módosítható könnyen a kódbázis.
* JavaScript szerkezete és működése: A javítás során a Clean Code elvek betartása, egységes elnevezési konvenciók, jól strukturált kódok vizsgálata történik meg. Milyen mértékben sikerült egységes kódot készíteni, amely máshol is újrahasznosítható. Az egyes algoritmusok működése helyes-e, kiterjed-e minden lehetséges esetre, valamint az elkészített algoritmusok mennyire átláthatóak és hatékonyak.
* Egységek szétválasztása: A HTML, CSS és JS állományok jól elkülöníthetőek egymástól, a készítettek kódbázisok külön-külön újrahasznosíthatóak, pl. (az API lekérés és JavaScript objektummá alakítása nem függ össze azzal, hogy majd hogyan jelenítjük meg). A DOM kezelés esetén is, hogy ne legyen szoros függés a JavaScript állományok szempontjából a HTML vonatkozásában (pl. nem az API-t lekérő egység jeleníti meg a tartalmat az oldalon annak szerkezetének ismeretében, hanem ez a függvény visszaadja és egy másik egység dolgozza fel).