**WEB-programozás I.**

**Bevezetés**

A projektet pár hiányában két GitHub fiókkal dolgoztam ki. A fő fiók (SDani0210) hozta létre a repót, a második fiókot (Durrot) kollaborátorként hozzáadtam. A commitokat mindkét fiókkal végzem.

A GitHub repó elérhetősége: <https://github.com/SDani0210/webprog-hazi_NN8T44>

Az oldal elérési linkje:

1. **Feladat**

A honlap kiinduló állománya az **index.html**, amelyet egységes szerkezet alapján építettem fel. Az oldal minden változatán megtalálható a fejléc (header), a vízszintes navigációs menüsor (nav), az oldalsáv (aside), a tartalmi rész (main – div-content) és a lábléc (footer). A menüsorban az aktuálisan kiválasztott menüpont külön stílussal van kiemelve. A láblécben a saját nevemet és Neptun kódomat tüntettem fel.

A lábléc elhelyezését úgy oldottam meg, hogy mindig az oldal alján maradjon, még akkor is, ha kevés tartalom jelenik meg. A stílusokat a **style.css** fájl tárolja, így az oldal minden része egységes megjelenést kapott.

**Táblázat menü (képek kellenek)**

A „Táblázat” menüpontban egy űrlap segítségével hoztam létre egy dinamikus CRUD-kezelő táblázatot JavaScript segítségével. Az adatok név, életkor, város és telefonszám szerint kerülnek rögzítésre és jelennek meg a táblázatban.

* A felületen minden mezőre építettem validációs szabályokat:
* A mezők nem lehetnek üresek
* A név és város mezők maximális hossza 30 karakter
* A telefonszám maximum 15 karakter
* Az életkor csak 0–120 közötti szám lehet

A táblázatban szereplő sorokra szerkesztési (✏️) és törlési (🗑️) lehetőség is van. Ha a felhasználó egy sort szerkeszt, a hozzáadás gomb ikonja pipára vált, és a korábban megadott adatokat frissíti, nem új sort hoz létre.

A felhasználók listája valós időben szűrhető bármely mező alapján (kivéve a gombok oszlopát), és az oszlopfejlécek kattinthatók, így a táblázat bármely oszlop szerint rendezhető növekvő vagy csökkenő sorrendbe.

Az alkalmazás kinézetét Bootstrap segítségével készítettem el, és színekkel valamint hover-hatással is jeleztem a rendezhető oszlopokat.

**HTML5 API**

Ebben a feladatrészben egyetlen HTML oldalon belül valósítottam meg a következő HTML5 API technológiákat.

* Web Storage: A localStorage használatával figyelem, hányszor töltöttem be az oldalt. A szám növekszik, és a látogató visszajelzést kap.
* Web Worker: Egy stopperóra valósult meg, amely tized- és ezredmásodperces pontossággal működik. A start gomb futtatja a háttérszálat, a stop leállítja, majd a start gomb újraindítja (nem nullázza).
* Geolocation API: A böngésző engedélykérés után kiírja a felhasználó szélességi és hosszúsági pozícióját. Hiba esetén egyértelmű visszajelzés történik.
* Drag and Drop API: Egy papír galacsin képet lehet behúzni egy szemetes kép ikonjába. A sikeres húzás után egy szöveges üzenet jelenik meg. A képek a projekt img/ mappájából származnak.
* Canvas API: Az egér mozgatásával rajzolni lehet a vásznon. A clear gombbal a felület törölhető.
* SVG: Egy valódi radioaktivitás veszélyt jelző piktogram készült el három külön forgatott, íves, kifelé szélesedő fekete lapáttal, középen fekete körrel és sárga háttérrel.
* Server-Sent Events (SSE): Ez a rész egyelőre hiányzik, mert a feladat nem engedi a szerveroldali technológiák használatát.

A példák mindegyike interaktív, stílusos, és Bootstrap-pel egységes dizájnba illeszkedik. A teljes rész az html5.html fájlban valósult meg.