Budapesti Gépészeti SZC
Eötvös Loránd Technikum

Műszaki informatikus
54 481 05

Vadász Dávid

2024

Záródolgozat

Füstérzékelő

Vadász Dávid

Budapest

2024

Budapest Gépészeti SZC

Eötvös Loránd
Technikum

Műszaki informatikus
54 481 05

Készítette

Vadász Dávid

Konzulens

Molnár József

KONZULENSI NYILATKOZAT HELYE

TANULÓI NYILATKOZAT

Alulírott …… műszaki informatikus tanuló kijelentem, hogy ezt a záró dolgozatot meg nem engedett segítség nélkül, saját magam készítettem csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző (k), cím, magyar nyelvű tartalmi kivonat, készítés éve, konzulens (ek) neve) a BGéSZC Eötvös Loránd Technikum nyilvános hozzáférésű elektronikus formában, a munka teljes szövegét pedig az iskola belső hálózatán keresztül (vagy hitelesített felhasználó számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik.

Kelt: Budapest, 2024 március 07.

……..

[Név]

TARTALOMJEGYZÉK

BEVEZETÉS

A dolgok internete (IoT) olyan elektronikai eszközöket vagy intelligens eszközöket jelent, melyek képesek felismerni lényegi információt és ezt továbbítani internetes hálózaton, hogy további eszközökkel kommunikáljanak vagy feldolgozhassák ezeket az adatokat adott célokra. Ezek az okos eszközök beépített szenzorok vagy érzékelők segítségével képesek adatokat gyűjteni. Az IoT a modern világban folyamatosan terjedő technológia, mellyel újabb és újabb módszereket hozunk létre életünk megkönnyítésére, és az online tér terjesztésére. Ilyen eszközök közé tartozik például az okosóra, okosriasztó, okosfüggöny, okosfűtésvezérlés, a záródolgozat témája, az okos füstérzékelő, és még sok más eszköz.

Köszönetnyilvánítás

Szeretném megköszönni osztálytársamnak és egyben csapattársamnak Szűcs Erik Dánielnek a záródolgozatban való aktív és kreatív részvételét és segítségét a megvalósításban.

Külön ki szeretném emelni Molnár József tanárurat mint a záródolgozat konzulensét a projektben való segítségnyújtásért, véleménynyilvánításért és ellenőrzésért, illetve Szénásy Zsolt tanárurat a füstérzékelő modelljéhez való tervezésnél nyújtott segítségért.

VÁLASZTOTT TÉMA INDOKLÁSA

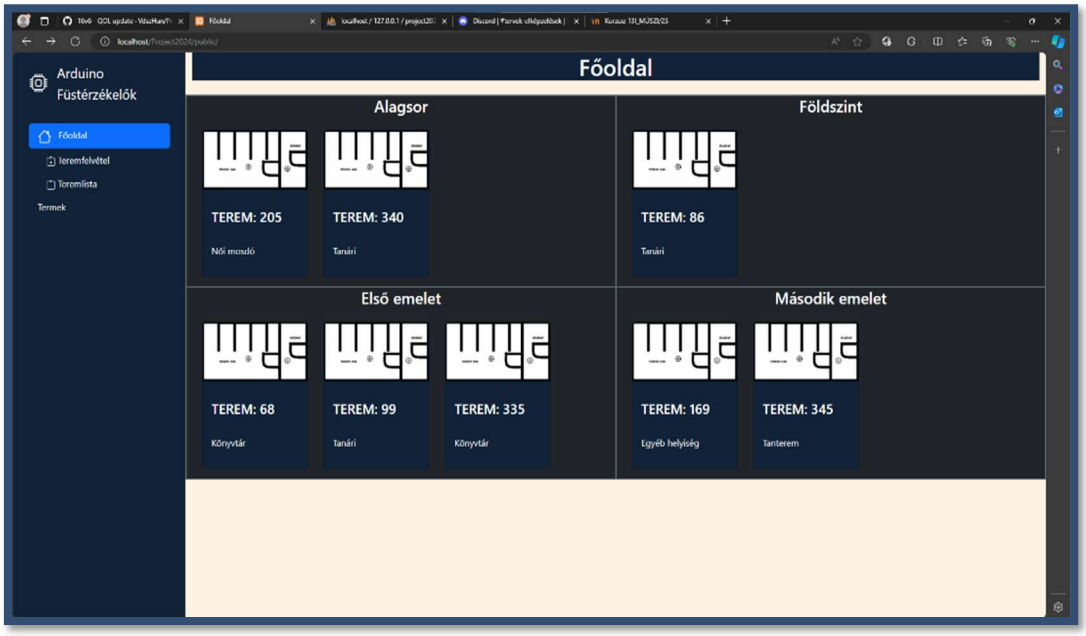
Öt évnyi informatikai tanulmányaim során a programozással történő fejlesztés és tervezés az, amiben kiemelkedőnek érzem magam, weboldalak tervezésével, programok kódolásával, módosításával szeretek foglalkozni, ennek köszönhetően programtervezőnek/fejlesztőnek tervezek továbbtanulni, illetve elhelyezkedni a jövőben.

A programozás felé való érdeklődésem és szeretetem, illetve a technikum érdekében való cselekvés miatt választottam ezt a témát. A projekt több ötlet és tény összevetéséből született, az Arduino mikrokontrollerek és szenzorok, a laravel, php és adatbázisok összekötése, illetve legvégül a technikumban fennálló problémából, az épületen belül való tiltott dohányzás.

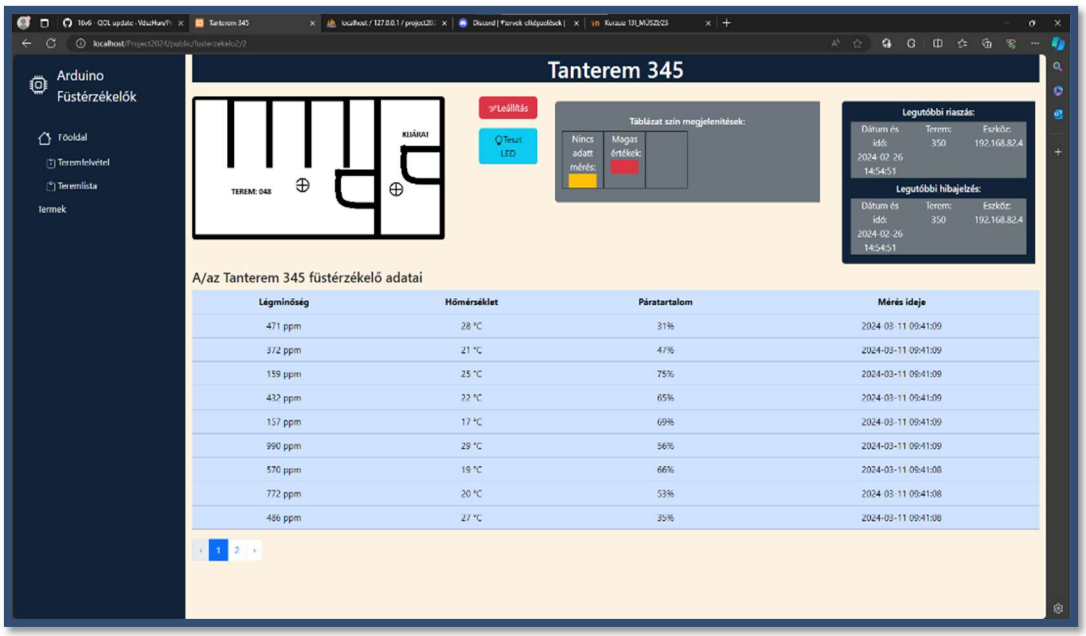
Az összevetésből megszületett a füstérzékelő ötlet, ami az elkövetkezendő hónapokban fokozatosan meg lett alkotva és továbbfejlesztve. Egy Arduino Wi-Fi modul, melyre egy hőmérséklet és páratartalommérő, levegőminőségmérő és jelző LED-ek kerültek, ezek köré pedig egy modell lett tervezve, majd 3D nyomtatva. Továbbá egy laravellel kialakított php webszerver, melyen valós időben lehet figyelni a szenzorok által mért adatokat, a mért adatok és termek pedig adatbázisba vannak feltöltve külön táblákban.

A füstérzékelő tanári kar által használható webes felület, ahol folyamatosan figyelhető, hogy történik-e valahol dohányzás a technikum épületén belül, melyet a kihelyezett füstérzékelők mérnek és töltenek fel. Az Arduino kódban csak a hálózat nevét és jelszavát kell módosítani attól függően, hogy a technikum melyik forgalomirányítóját éri el a kihelyezendő füstérzékelő, maga a csatlakozás dinamikusan történik a hálózatra (DHCP), lefoglal magának egy szabad IP címet az eszköz.

Termeket a webes felületen lehet felvenni, módosítani és törölni az igények alapján, azaz dinamikusan bővíthető a webes felület.



1. ábra: A Főoldal nézete fejlesztés közben, a Microsoft Edge böngészőben



2. ábra: Egy tanterem nézete fejlesztés közben, a Microsoft Edge böngészőben

TÉMA KIFEJTÉSE

Fejlesztői környezet bemutatása

A záródolgozat fejlesztése során több program, keretrendszer és könyvtár is igénybe lett véve a célok eléréséhez.

A webes felület fejlesztéséhez a **Visual Studio Cod**e programot használtam, az füstérzékelő fejlesztéséhez az **Arduino IDE** programot vettem igénybe, a füstérzékelő modelljéhez pedig a **SolidWork**s program volt felhasználva. A webszerver működtetéséhez és adatbázis kezeléshez pedig szükség volt az **XAMPP** nevű programra. A webes felület teszteléséhez és megtekintéséhez a **Google Chrome** és **Microsoft Edge** böngészőket használtam.

A kódolásban rengeteget segített nekem Molnár tanár úr által leadott órai anyagok a php és arduino programozásról, illetve szintén a tanárúrtól származó online segédanyagok, sokat tanultam a webszerverek készítéséről és a mikrokontrollerek, szenzorok sokféle felhasználásáról a tanárúrnak köszönhetően melyet itt alkalmazhattam.

Problémák megoldásánál a w3school, a stackoverflow és a github oldalak segítettek, az interneten nap mint nap oszjták meg az emberek az általuk tapasztalt programozási problémákat, melyekre szinte mindig találni megoldásokat ezeken az oldalakon.

A záródolgozat, és minden ahhoz tartozó kód, dokumentáció, modellek, tervek, képek, stb. feltöltésre került a saját **GitHub** profilomon generált tárolóba (**Repository**-ba), mely a záródolgozat fejlesztése során folyamatosan kapta a frissítéseket, módosításokat (**Commit**-okat). A GitHub által beépített **Commits** funkciónak köszönhetően visszanézhető minden változás, amin keresztülment a projekt a létrehozásától kezdve.

A projekt linkje: https://github.com/VdszHun/Project2024

A záródolgozatban felhasznált eszközök

A webes felület megtervezéséhez szükség volt az ingyenes **Laravel** keretrendszerre, mellyel PHP alapú webszervereket lehet fejleszteni, ennek az alkalmazásához pedig szükség volt a **Composer** nevű PHP függőség kezelő telepítésére, mely szintén ingyenes.

A web fejlesztése a Microsoft által forgalmazott, **Visual Studio Code 1.87.0**-ás változatában történt, ahol a Composert és Laravelt telepíteni kellett ezzel a paranccsal a programban található terminálon keresztül: **composer** global require laravel/installer. Továbbá a programon belül telepítésre került a **Laravel Blade Snippets** és **Laravel Snippets** kiegészítők, melyek szintaxisok és prefixek kiemelésével, egyszerűbb alkalmazásával segítették a kódolást, illetve a **Thunder Client** kiegészítő, mellyel API-on keresztüli adatok feltöltését lehet tesztelni.

Az okoseszköz fejlesztéséhez az **Arduino IDE 2.3.2**-es változata volt használva, azon belül telepítésre került az **esp8266**-os nevű board kezelő, mellyel a Wi-Fi modulra lehet kódot írni, **Adafruit Unified Sensor** és **DHT sensor library** nevű könyvtárak, melyek szükségesek voltak a szenzorokhoz való kód készítéséhez. A webszerver és adatbázis kezeléshez az **XAMPP Control Panel v3.3.0**-as változata volt alkalmazva.

A füstérzékelő modelljének megtervezéséhez a **SolidWorks 2023** volt használva, a modell pedig a technikumban lett kinyomtatva egy 3D nyomtatóval.

Az okoseszközhöz felhasznált alkatrészek:

- 1db ESP8266 Wemos - NodeMCU 1.0 (ESP-12E modul)
- 1db USB-C kábel
- 1db DHT11 Temperature & Humidity Sensor
- 1db MQ-135 Gas Sensor
- 2db Arduino LED
- 1db 3kΩ, és 1db 2kΩ ellenállás
- Vezetékek
- Egyedi, a technikum által nyomtatott 3D modell

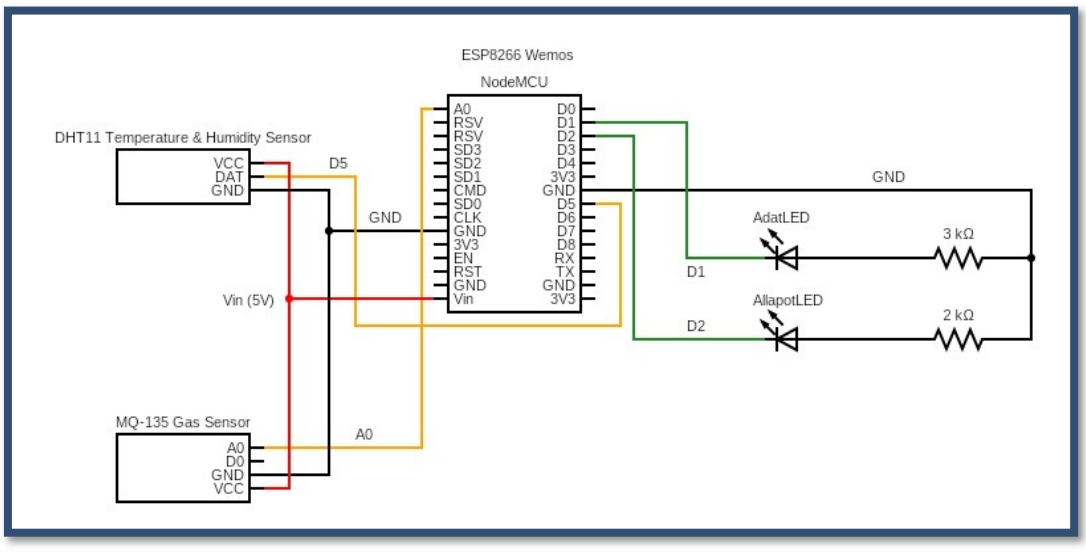
RENDSZER BEMUTATÁSA

Rendszertervrajz

A füstérzékelőt kétféleképpen is meg kellett tervezni, készült egy áramköri nézet és modellnézet is az eszközről. Az alább látható áramköri rajz fontos szerepet játszik a technikum érdekében, hisz ha több füstérzékelőt szeretnének kihelyezni a jövőben e rajz alapján lehet összekötni az eszközöket, LED-eket és ellenállásokat az adott feladatokra ellátott vezetékekkel, nagyon fontos figyelembe venni a vezetékekhez kapcsolódó csatlakozások neveit, hisz ha nem megfelelően vannak, bedugva zárlatossá tehetik a rendszert, ami tönkre is teheti az alkatrészeket, vagy elégetheti az ellenállásokat.

Az eszközön felhasznált csatlakozások jelentései:

- D1, D2, D5: Digitális csatlakozás
- A0: Analóg csatlakozás
- GND: Földelés
- Vin: 5 Voltos áramellátás
- VCC: Áramfogadás
- DAT: Adat csatlakozás, digitális csatlakozáshoz kell kötni

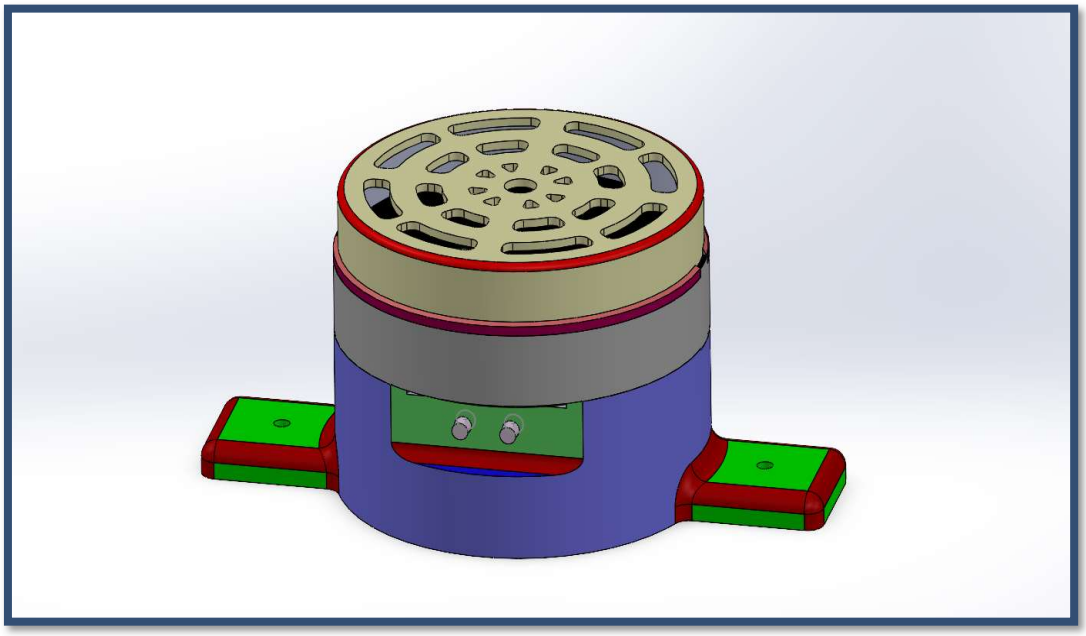


3. ábra: A füstérzékelő áramköri nézete a Circuit Diagram weboldalon

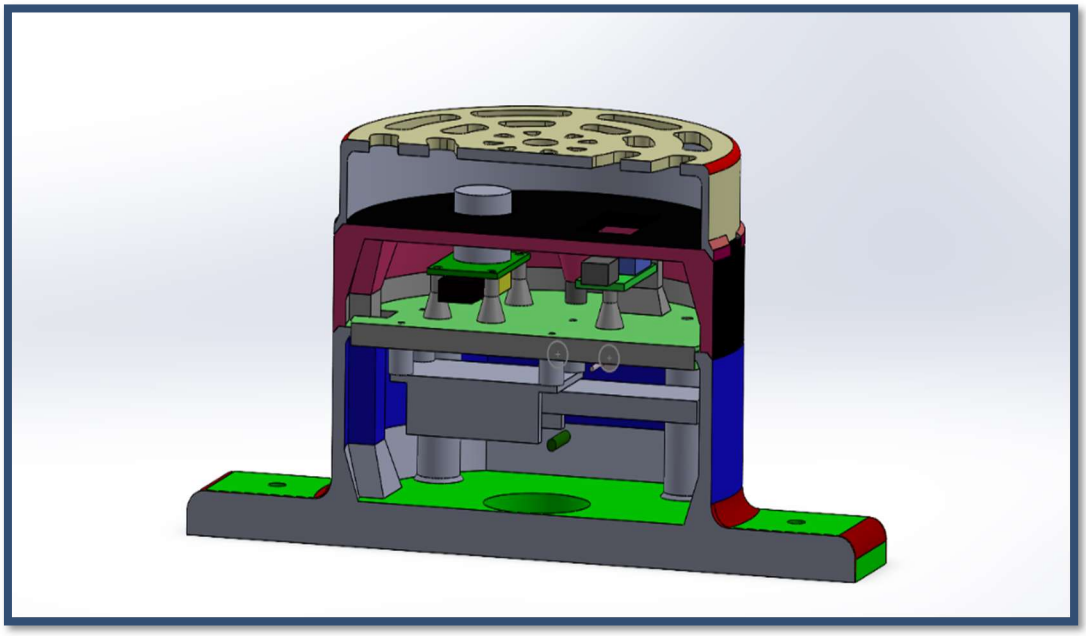
Az eszközről szintén készült egy 3D modell, melyet a technikum kinyomtathat a saját 3D nyomtatójával bármennyiszer. A modell tartalmazza a tok darabjait, illetve a lemodellezett Wi-Fi modult, szenzorokat, LED-eket és ellenállásokat. A végleges modell, és minden egyes alkatrésze külön-külön megtalálható a projekt **Project2024\public\3dmodel** mappájában. Szintén itt megtalálható egy robbantott nézet a modellről, mp4 fájlformátummal, a robbantott nézet megmutatja a model szét és összeszerelését lépésenként egy rövid videóban, melyet a későbbiekben lehet alapul venni további eszközök összeszerelésénél és javításánál.

A modell tervezésekor figyelembe lett véve

- **a modell egyszerű összerakhatósága**: a modell darabokból és csavarokból áll össze, melyeket könnyen össze lehet rakni.
- **az eszközök elhelyezhetősége**: minden felhasznált szenzor és eszköz mérete le lett mérve milliméter pontosan, hogy a modellben is precízen el lehessen helyezni őket.
- **a LED-ek kivezetése**: a LED-ek ki vannak vezetve a modell oldalára, hogy bárki láthassa az általuk küldött jelzéseket.
- **az esetleges plafonra szerelés**: a modell oldalán található szárnyaknak köszönhetően az eszköz akár plafonra is helyezhető.
- **szűrő, ami eléri a szenzorokat**: a modell tetején lévő szűrőn keresztül bejut a levegő a szenzorokhoz, így a szenzorok sikeresen végre tudják hajtani a méréseket. Mivel a modell több emeletből áll, így ha esetleg valamilyen folyadék éri az eszközt legfeljebb a szűrőn tud átjutni, megvédve az alkatrészeket sérüléstől.
- **a vezetékek törés és szakadásvédelme**: a modell belülről tágas és több szintből áll, így a vezetékek nem sérülhetnek.
- **tápegység bevezetése**: a modell alja úgy lett tervezve, hogy a Wi-Fi modulra lehessen helyezni valamilyen tápegységet. Alapesetben a Wi-Fi modul USB-C kábellel kap áramellátást, viszont biztos, hogy ez nem lenne kivitelezhető mindenhol, így más módszerek alkalmazásához lett hagyva hely a modellben.



4. ábra: A füstérzékelő 3D modellje, a SolidWorks 2023-ban



5. ábra: A füstérzékelő félbevágott 3D modellje, a SolidWorks 2023-ban

A program felépítése

A webnézethez és az Arduinohoz felhasznált programkódok külön lettek bontva a nagy mennyiségük miatt, az alábbi alfejezetben ezeket emelem ki, a végén pedig a projektben használt adatbázis működése is megjelenik. A webnézet sok mappából áll, a dokumentációban azokat emeltem ki melyekben tettem módosításokat.

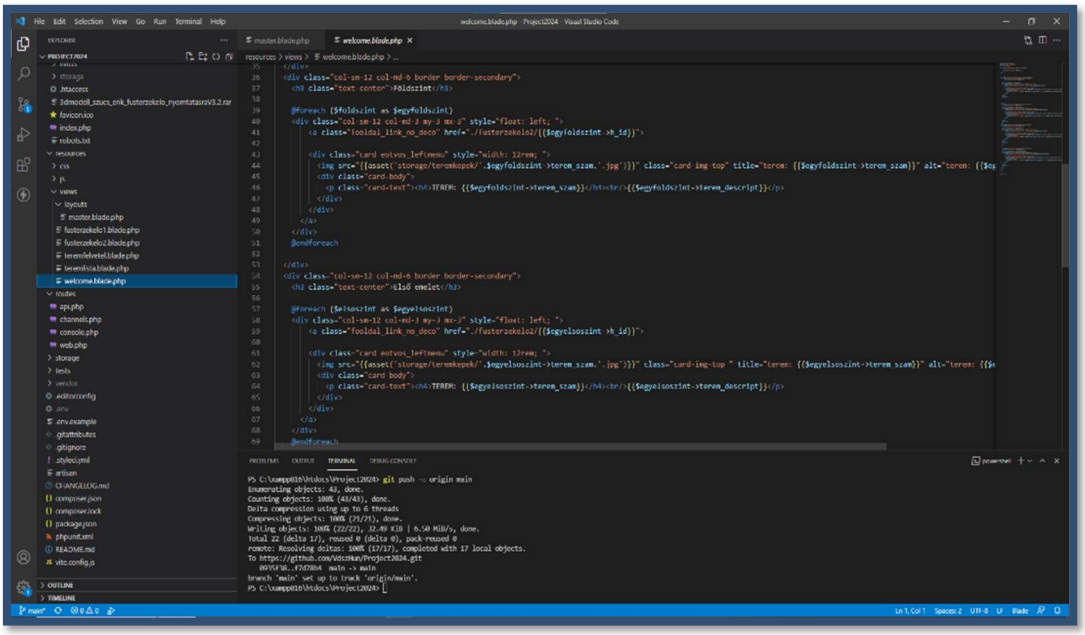
Views mappa: a views mappában találhatóak azok a fájlok, amiket a webnézet használatakor láthatunk. A mappán belül elhelyezkedik egy további mappa, **layouts** névvel, azon belül pedig a **master.blade.php** nevű fájl. A master a weboldal alapja, a neve is utal rá, a. **blade.php pedig a fájlkiterjesztés**, mely a többi views mappán belül megjelenő fájlon is megtalálható. Maga a master fájl a főoldal tartalmának egy részét teszi ki, a másik része a **welcome** fájlban található, viszont ez nem jelenti azt, hogy másik oldalra kellene kattintani a teljes főoldal nézéséhez, mert egybe jelenik meg a master és a welcome. A master fájl szintén tartalmaz olyan elemeket, amik minden oldalon megjelennek, például a baloldalon található menü sáv, mellyel ugrálhatunk a webfelület különböző lapjain, vagy az adott oldalak címe.

A **fusterzekelo2** nevű fájlban található az a programkód, amit a weboldalon akkor láthatunk, ha megnyitunk egy termet.

A **teremfelvetel** nevű fájlban a teremfelvétel nevű fül nézete és funkciói vannak kódolva. Ezen a nézeten vehetünk fel termeket a weboldalra, ahol megadhatjuk a terem számát, szintjét, az elhelyezendő eszköz IP címét, a terem típusát (pl. Tanári), illetve egy képet csatolhatunk a teremhez, mely a későbbiekben ahhoz a teremhez lesz csatolva mindenhol.

A **teremlista** fájlban a teremlista fül nézete, és az ahhoz tartozó működés található. Ezen a nézeten egy táblázatos listát láthatunk az adatbázisban létező termekről, melyen módosíthatjuk a felvételnél megadott adatokat, vagy akár törölhetünk is termeket.

A **welcome** fájlban, mint feljebb is említve a főoldal további tartalmai találhatóak, ebben a fájlban a főoldal négy emeletre való tagolása helyezkedik el, melynek köszönhetően a webnézeten a termek a szintjük alapján vannak elhelyezve.



6. ábra: Részlet a welcome.blade.php-ból, a Visual Studio Code programban

Routes mappa: Ebben a mappában két fájl került módosításra, a **web.php** és **api.php** fájlok. A web.php fájlban a projekt webes útvonalait adhatjuk meg, melyek működésében a controllereknek van nagy szerepe. Az api.php-val pedig az adatok feltöltéséhez lett létrehozva egy útvonal, melynek a működése szintén controllerben lett megoldva.

Controllers mappa: A controllerek útválasztásnál, adatfolyásnál, adatbáziskapcsolatnál, logikai feladatoknál, és más fontos, mélyre gyökerező területeken nyújtanak segítséget. A projektben minden oldalnak van egy saját controllerje amik segítik az adott oldalnézet megfelelő működését, kapcsolatát az adatbázissal, az értékek pontosságát, kapcsolatot az eszközzel, stb.

A projektben használt controllerekben az alábbi feladatok és funkciók találhatóak meg: útvonalak megfelelő működése, átirányítása, weboldali kapcsolása az eszköznek, vagy a LED-jeinek a tesztelése, a főoldalon a termek sorrendbe helyezése szintek alapján, adatfeltöltés és annak a validációja (a feltöltendő adatok ellenőrzése a követelmények

alapján), az adattáblázatok színekre bontása kritériumok alapján (például túl magas érték esetében piros legyen az adott cella).

Models mappa: A model-ek az adatbázis és a php közötti kommunikációban segítenek, olyanok, mint egy híd, elvégzik a logikai és adatkezelési feladatokat a két fél között. A projektben minden táblának (vagy másképp nézve migrációs fájlnak) van egy model fájl párja is ebben a mappában, mely elvégzi a feljebb említett feladatokat, és még másokat. A modeleken belül meg van adva, hogy melyik táblához kapcsolódnak, mi az egyedikulcsuk, védést kapnak, az időbélyegek pedig tiltva vannak.

Providers mappa: A providers mappában csak az **AppServiceProvider.php** fájl került módosításra, egy sor kóddal, mely lehetővé teszi a projektben a paginálás funkciót, a paginálással táblázatokba helyezhetünk léptéket, ez arra megoldás, hogyha már sok adat található egy táblázatban feloszlhatjuk oldalakra a megjelenését.

Migrations mappa: a migrations mappában az adatbázis létrehozásához segítő fájlok találhatóak, mindegyik migrációs fájl egy táblát hoz létre a projekt adatbázisában, ha már létezik a tábla, az adott tábla nem fogja újra legenerálni. Létezik migrációs fájl ezekhez a táblákhoz: **helyszinek, meresek**. Lejjebb az adatbázisos alfejezetben részletesen le van írva mindegyik tábla feladata.

A projekt első generálásakor alap migrációs fájlokat is tartalmazott a kód, azok törlésre kerültek, mivel a projektben hasznuk nincs.

Seeders mappa: A seederek az adatbázist előregenerált adatokkal való feltöltésére szolgáló fájlok, lehet rájuk gondolni úgy, mint egy script fájlra, ami bizonyos lépéseket hajt végre futtatáskor. A projektbe két seeder fájlt hoztam létre, az egyik a helyszín táblát tölti fel véletlenszerű adatokkal, pontosabban kiválaszt a program egy terem szintet, egy terem számot, egy terem típust, és végül egy véletlenszerűen generált egyedi IP címet. Ez az egész egy négyszer ismétlődő for loopban helyezkedik el, ami azt jelenti, hogy minden egyes seeder futtatáskor, melynek a parancsa php artisan db:seed lefut ez a parancssorozat 4-szer, generálva 4 termet szinttel, számmal, típussal, és mindegyiknek különböző IP címmel. A másik létrehozott seeder pedig 30 véletlenszerű mérést hoz létre, egyedikulcsok alapján az első 4 eszköznek, a generált adatok levegőminőség, hőmérséklet, páratartalom és hibakód. A két seedernél meg kell hívni az adott táblák modeljeit a sikeres lefuttatáshoz.

Szintén a Seeder mappában helyezkedik el a DatabaseSeeder fájl, melyben a feljebbi seederek vannak meghívva.

A seedereknek a projektben nincs kulcsfontossága, könnyebítés érdekében hoztam létre őket, zavarónak találtam, hogy ahányszor ürítettem az adatbázist, vagy töröltem, vagy új helyen hoztam létre a projektet újra és újra be kellett szúrni adatokat manuálisan. A megírt seederekkel viszont egy parancs futtatásával le lehet rövidíteni sok munkát.

Public mappa: **A public mappában helyezkednek el a teljes projekthez felhasznált eszközök**, például a 3D modell fájljai, az Arduino kód, CSS fájl, a dokumentáció, képek és még más egyebek.

A **CSS** fájl egyes HTML elemek kinézetén vagy tulajdonságán módosít, például ebben a fájlban lett beállítva a weboldalon található egyedi színek, melyek megegyeznek néhány, a technikum hivatalos weboldalán található színnel.

Egyéb fájl(ok) és mappák: A projektben található egy **.env** nevű fájl, env fájlokban konfigurációs beállításokat, környezeti változókat és érzékeny információt szoktunk tárolni. A fájlban módosításra került két változó:

- **DB_HOST=localhost**
- **DB_DATABASE=project2024**

A host változóval az adatbázisnak azt mondjuk meg, hogy milyen IP címen találja az adatbázist, localhost ebben az esetben, mely a helyi számítógépet jelenti, a database változóval pedig a nevét változtatjuk az adatbázisnak.

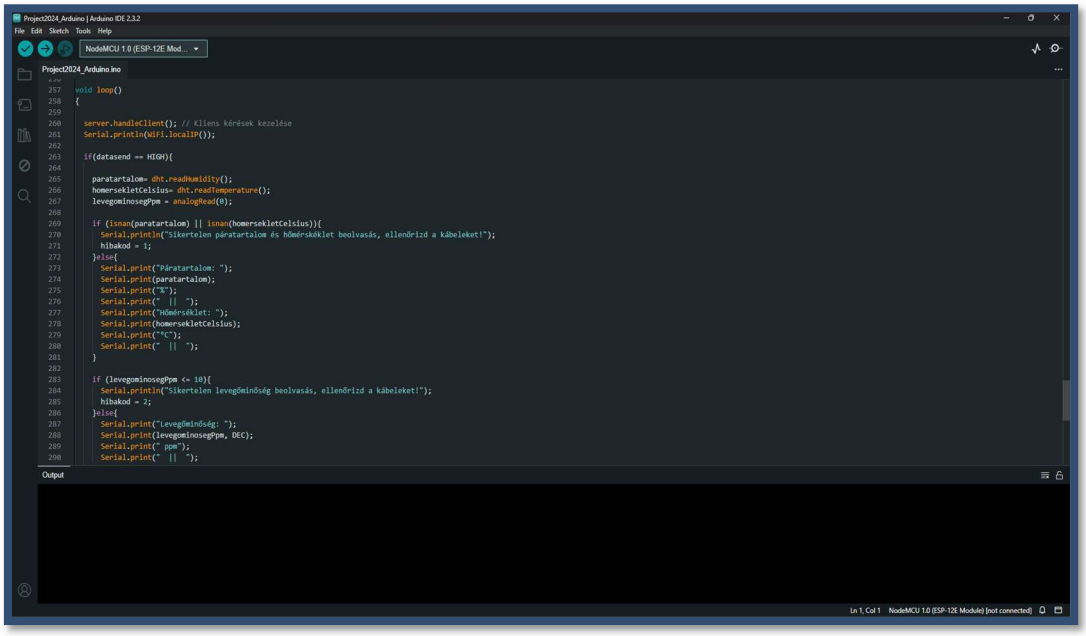
A projektben található a storage mappa, ezen belül helyezkedik el az **app/public/teremkepek** útvonal, melybe a termekhez feltöltött képek találhatóak.

Arduino kód: Az Arduino kód egyetlen fájlból áll. A kód elejére leírtam, hogy milyen könyvtárak és board kezelő szükséges az eszköz sikeres működéséhez, ezt követi bizonyos könyvtárak meghívása, változók deklarálása, és webszerver kapcsolat beállítása. A kód következő részletében a webszerveren található eszköz kapcsoló és LED tesztelő működését biztosító kódrészletek találhatóak. Mindezek után következik a kód setup részlete, az a parancssorozat, ami az eszköz felállásakor egyszer fut le, ide tartozik például a hálózatra kapcsolódás, vagy a csatlakozások bekapcsolása. A setup után helyeztem el az adatküldés kódrészletét, mellyel a webszerveren adott teremhez tölti fel

az eszköz a mért adatokat. A hibakódok jelzése is külön részlegetek kapott, minden egyes hibatípusnak van saját jelzése:

Hibakód száma	Hibakód jelentése	Hibakód jelzése
0	Minden megfelel	ÁllapotLED: Folyamatos AdatLED: 1 rövid villogás
1	Sikertelen páratartalom és hőmérsékletbeolvasás	ÁllapotLED: 1 rövid, 1 hosszú villanás
2	Sikertelen levegőminőségbeolvasás	ÁllapotLED: 1 rövid, 2 hosszú villanás
3	Sikertelen IP címzés	ÁllapotLED: 1 rövid, 3 hosszú villanás
4	Sikertelen adatküldés az adatbázisba	ÁllapotLED: 2 rövid, 3 hosszú villanás

A kód legvégén pedig a loop található, a setupot követő végtelen ciklus. ebben a kódrészletben 6 másodpercenként az eszköz megpróbál mérni a szenzoraival, figyelembe véve bármilyen fellépő hibát, majd megpróbálja elküldeni a mért adatokat az adatbázisba, illetve kiiratja őket az Arduino IDE soros portjára is (serial port). A soros portra való kiirítás arra hasznos, hogy megtudjuk az eszköz megfelelően működik-e mielőtt el lenne helyezve a kijelölt helyére, illetve ide kerül feltöltésre az eszköz saját IP címe.



7. ábra: Részlet az Arduino kódból, az Arduino IDE-ben

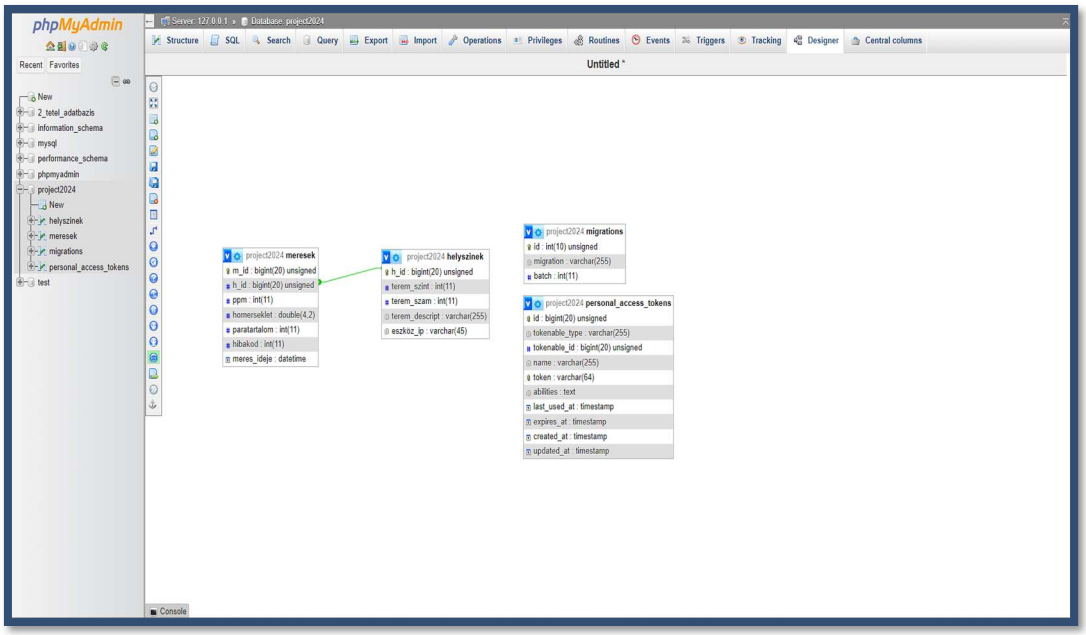
Adatbázis: Az adatbázis két táblával lett bővítve azokon kívül, amit a rendszer alapból legenerál az adatbázis működéséhez. Ezek a helyszínek és a mérések táblák. A helyszínek táblában a felvett termeket tároljuk el, a tábla tartalmazza:

- **h_id**: A tábla egyediazonosítója
- **terem_szint**: A terem szintje a technikum szintjei alapján
- **terem_szam**: A terem száma a technikum ajtóira kihelyezett számok alapján
- **terem_descript**: A terem típusa az adott terem használata alapján, például tanári, előadó, labor, tanterem, férfi mosdó, stb.
- **eszköz_ip**: A teremhez tartozó füstérzékelő IP címe. Ezt a címet akkor tudhatjuk meg, hogyha sikeresen felcsatlakozott a füstérzékelő az adott iskolai hálózatra és kiiratja az IP címét.

A mérések táblában az eszközök által mért adatokat tároljuk el, a tábla tartalmazza:

- **m_id**: A tábla egyediazonosítója
- **h_id**: A helyszínek tábla azonosítója, amely itt idegenkulcsként funkcionál, ezzel az értékkel lehet összekötni, hogy az adott eszköz melyik teremhez tartozik.

- **ppm**: a mért levegőminőség értéke, egész számként tárolva
- **homerseklet**: a mért hőmérséklet értéke, két tizedesjegyű törtszámként tárolva
- **paratartalom**: a mért páratartalom értéke százalékban tárolva
- **hibakod**: Arduinóból érkező esetleges hibakód száma
- **meres_ideje**: Adatfeltöltéskor elmentett időpont, év-hónap-nap óra:perc:másodperc felosztásban



8. ábra: Az adatbázis tervezői nézete, melyen megfigyelhető az idegenkulcsos összekötés, az XAMPP által futtatott phpMyAdmin oldalon

Az adatbázisba többféleképpen is lehet feltölteni adatot. Alapesetben a füstérzékelő tölt fel adatokat 6 másodpercenként. Szintén lehet feltölteni adatot manuálisan a phpMyAdmin nézeten belül, vagy akár a seederekkel is, végső módszerként pedig a Visual Studio Code-ban lévő Thunder Client kiegészítővel.

Adatokat törölni lehet manuálisan, vagy a weboldal teremlista nézetében, teljes termeket, szintén van mód adatok módosítására ugyanezeken a helyeken.

SAJÁT VÉLEMÉNY

A technikumban töltött 5 évem alatt számtalanszor mentem be úgy mosdóba, hogy dohány szagát éreztem, gyakran nyomát se találva annak, hogy ki okozhatta, ez a tény pedig bosszantó, mivel a dohányfüst káros hatású az emberi szervezetre, és így aki bemegy a mosdóba ki van téve a károsodásnak ugyanúgy mint aki dohányzik, ráadásként a technikum házirendje szerint tilos az épületen belül dohányozni a feljebbi ok miatt is, és a cigaretta általi tűzveszély miatt, nehéz számomra megérteni, hogy a dohányzók szünetek ideje alatt miért nem hagyják el az épületet elvégezni a dolgukat, majd térnek vissza, hisz azzal nem ártanak másnak, nem kerül be egy zárt légtérbe a füst. A projekt megalkotásával reményt látok abban, hogy változás menjen keresztül ezeken a személyeken, hisz a fizikai érzékelő nem riaszt, így a dohányzók nem sejthetik az érzékelő jelenlétét, nagyobb eséllyel elkapva őket és elnyerve méltó büntetésüket, ösztönözve őket, hogy gyűjtsék össze az erőt arra, hogy egyszerűen kisétáljanak az épületből. A weboldal felhasználóbarátnak lett tervezve, így a tanári kar könnyen megszokhatná a használatát, mely akár az eszköz gyors elterjedését is eredményezhetné a technikumban. Arra is látok esélyt, hogy esetleg a projekt inspiráljon más oktatási intézményeket is hasonló eszközök tervezésére és alkalmazására. Örülök, hogy részt vehettem egy projekt fejlesztésében, amivel többszáz diák, és a tanári kar mindennapjain könnyíthetek.

Szintén örülök a tapasztalatnak és élménynek, amit a záródolgozat készítése alatt szereztem, hisz ezt a tudást vihetem magammal tovább felsőoktatásba, vagy majd a munka világába, mint jövőbeli programozó.

Továbbfejlesztési lehetőségek

Fontos fejlesztési ötletnek tartom a jobb szenzorok alkalmazását a mostaniakhoz viszonyítva, a jelenlegi szenzorok pár méteres hatótávolsággal rendelkeznek, így kell, körülbelül 15 másodperc mire eljutna a dohányfüst az érzékelési zónához. Modernebb és fejlettebb szenzorok hamarabb és távolabbról érzékelnék a füstöt, hamarabb jelezve a webfelületen a fellépő gondot. A jobb szenzorok érdekében a modellen is tovább kellene fejleszteni moduláris módszerre, hogy többfajta szenzor is bekerülhessen

ugyanarra a helyre, például a gépházakba is többfajta alaplapot lehet behelyezni a csavarfoglalatok alapján.

Létezik az esély arra, hogy a dohányzók megpróbálnának sérülést okozni vagy teljesen tönkretenni a kihelyezett eszközöket a saját érdekükben, így a modell továbbfejlesztését is fontosnak tartom, például egy rácsrendszer bevezetését, vagy egy továbbfejlesztett szűrőt, ami tovább csökkenti a folyadékok általi esetleges kárt, esetleg még a falak tovább vastagítását rázkódásvédelemhez.

IP cím beállítása infravörös távvezérléssel, egy infravörös jeladóval és távirányítóval, illetve az Arduino kód továbbfejlesztésével kivitelezhető lenne az, hogy ne a programkódban kelljen átírni IP címet, hanem egy infravörös távirányítón lévő gombokkal lehessen azt beállítani, lecsökkentve a beállítási időt statikus hálózatoknál.

Globális kapcsoló az összes füstérzékelőnek a webfelelületen. A nap minden szakában nincs szükség az eszközök működésére, és a felesleges futásuk csak csökkentené az élettartamukat, illetve feltöltené az adatbázist sok felesleges adatokkal, amiket utána törölni kéne. Egy globális kapcsolóval fel és le lehetne kapcsolni az összes kihelyezett eszközt, megoldva ezeket a felmerülő problémákat.

Mért adatok törlése termenként vagy globálisan. Habár a helyszínek táblát csak mi tudjuk bővíteni a mérések tábla folyamatosan növekedne méretileg, lelassítva idővel a webfelület működését, és egyre több tárhelyt venne igénybe. A weboldalra lehetne tervezni olyan gombot vagy felületet ahol termenként lehetne törölni a mért adatokat, vagy globálisan. Akár egy automatikus rendszer is bevezethető lenne, ami például törölne minden adatot, ami több mint X nap idős.

Hűtés bevezetése az eszközbe. Jelenleg a modellnek nincs hűtés tervezve, a benne található szenzorok, elektronikai eszközök használat során idővel felmelegednek, és ez csökkenti az élettartamukat, akár egy egyszerű kis ventilátor is sokat javíthatna az eszköz élettartamán, persze megfelelő tervezéssel, hisz figyelembe kéne venni, hogy a mérni kívánt levegő bejusson, az eszközök általi melegebb levegő pedig kifelé, mint egy számítógépházban.

DOKUMENTÁCIÓ – 4. FEJEZET

A kód egyszerűsítése, javítása, minőségének növelése. Nincs olyan programkód, amin ne lehetne javítani, fejleszteni, szebbé tenni, egyszerűbbé tenni, ahogy a technológia fejlődik, úgy jelennek meg újabb és egyszerűbb módok a programozásban is.

A weboldalon található színek módosíthatósága preferencia alapján. Ha esetleg a projekt túljutna a technikumon, lehetne továbbfejleszteni úgy a weboldalt, hogy bármely más iskola is használhassa a webnézetet olyan színbeli beállításokkal amilyeneket ők szeretnének, jelenleg a webnézet a technikum színeire lett mintázva.

Mobilbarát weboldal. A jelenlegi weboldal számítógépes nézetre lett tervezve és mobilnézeten használhatatlan, ha igény lenne rá a mobilnézet támogatottá válna.

ÖSSZEFOGLALÁS

Konklúzió

Eredmények bemutatása

IRODALOMJEGYZÉK

A záródolgozatomban használt források:

- https://laravel.com/
- https://getcomposer.org/
- https://www.arduino.cc/
- https://www.circuit-diagram.org/
- https://www.apachefriends.org/hu/index.html
- https://www.infojegyzet.hu/webszerkesztes/zarodolgozatmintak/
- https://www.w3schools.com/
- https://getbootstrap.com/
- https://stackoverflow.com/
- https://github.com/
- https://www.solidworks.com/
- https://color.adobe.com/create/image
- https://hu.wikipedia.org/wiki/Kezd%C5%91lap

ÁBRAJEGYZÉK

A CSATOLT MELLÉKLETEK JEGYZÉKE

1. melléklet: A Budapesti Gépészeti SZC Eötvös Loránd Technikum logója
   Forrás: https://eotvosszki.hu/

MELLÉKLETEK

1. melléklet: A Budapesti Gépészeti SZC Eötvös Loránd Technikum logója