## 

Nagy-Pál Olivér

Gabóczi Ádám

Papp Bence

Sátoraljaújhely, 2024. május 21.

## 

## **Tartalomjegyzék**

I. Bevezetés

A. A fogászati rendelő projekt célja

B. A fogászati egészség fontossága

II. Szolgáltatásaink

A. Általános fogászati kezelések

B. Esztétikai fogászat

C. Fogszabályozás

III.

A. Fenntartható gyakorlatok

B. Zöld kezdeményezések

IV. Árak és Finanszírozás

A. Kezelési költségek

B. Biztosítási lehetőségek

V. A fogászati egészség népszerűsítése a közösségben

A. Oktatási programok

B. Résztvétel közösségi eseményeken

VI. Jogi és Etikai Szempontok a Fogászatban

A. Páciensjogok

B. Etikai normák

VII. Összefoglalás és Jövőbeli Tervek

A. Projekt eredményei

B. Jövőbeli fejlesztési tervek

## 

## 

## **A témánk neve Betegnyilvántartó!**

**I. Bevezetés**

A fogászati egészség kiemelkedő fontossággal bír az egyének általános jólétében. A mosoly, az étkezési képesség és az önbizalom meghatározó elemei mindennapi életünknek. A projektünk célja, hogy bemutassa a modern és magas színvonalú fogászati rendelőnket, ahol nem csupán a fogászati problémákra koncentrálunk, hanem a pácienseink teljes körű egészségét is szem előtt tartjuk.

**A) A Fogászati Rendelő Projekt Célja:**

Ebben a bevezető fejezetben röviden áttekintjük a projekt céljait és azt, hogyan kívánjuk elérni azokat. Megismertetjük az olvasóval a fogászati rendelőnk filozófiáját és azt, hogyan különbözünk más hasonló intézményektől.

**B) A Fogászati Egészség Fontossága:**

Részletezzük, miért létfontosságú a fogászati egészség a teljes testi egészség szempontjából. Bemutatjuk a fogászati problémák lehetséges hatásait, és hangsúlyozzuk a megelőzés és a rendszeres ellenőrzések fontosságát.

Megismerhetjük az orvosokat, akik a modern diagnosztikai eszközök segítségével megfelelően látják el a pácienseket! A rendelőnkben törekedünk a környezettudatosságra, illetve fenntartható gyakorlatokat is tartunk a jövő fogorvosai számára!

Végül a projekt végén láthatjuk, hogy a jövőbeli terveink és kilátásaink milyen céllal fognak megvalósulni!

# **Rendszerkövetelmény felhasználói dokumentációja:**

Az alábbiak szükségesek ahhoz, hogy a Windows 10-re frissítsük PC-nket vagy táblagépünket:

1. **Legújabb operációs rendszer:** Győzödjünk meg arról, hogy a legújabb verziót futtatjuk, vagyis a Windows 7 SP1-et vagy Windows 8.1 Update-et.

* Állapítsuk meg, melyik verzió fut a számítógépen,
* Ha szükségünk van a legújabb verzióra, akkor töltsük le a Windows 7 SP1-et vagy Windows 8.1 Update-et.

1. **Processzor:** 1 gigahertzes (GHz) vagy gyorsabb processzor vagy SoC
2. **RAM:** 1 gigabájt (GB) 32 bites rendszerhez vagy 2 GB 64 bites rendszerhez.
3. **Merevlemez-terület:** 16 GB 32 bites operációs rendszerhez vagy 20 GB 64 bites operációs rendszerhez
4. **Videókártya:** A DirectX 9-es vagy újabb verziójával kompatibilis, WDDM 1.0 illesztőprogrammal
5. **Kijelző:** 800 X 600

## Funkciók használatának bemutatása:

# 

# Az adatbázis felépítése

A táblázat felépítése:

A táblázat sorokból és oszlopokból áll.

A sorokat rekordoknak (magyarul bejegyzéseknek) hívjuk. Minden rekord egy személy, tárgy, dolog tulajdonságait tartalmazza.

A rekordokban lévő adatokat mezőknek hívjuk, amik táblázatos elrendezésben oszlopokat alkotnak. A mezők neve kerül az oszlop tetejére fejlécnek.

Egy sor és egy oszlop metszéspontja a cella. Több szomszédos cella együtt egy tartományt alkot.

Időnként a táblázatot 90 fokkal elforgatva használjuk, amikor a sorok kapnak fejlécet, és az oszlopok tartalmazzák az adatbázis elemeit. Kereszt táblázatról akkor beszélünk, amikor a sorok és az oszlopok is szempontokat tartalmaznak. Az adatok tárolására ez kevésbé alkalmas, ezért már létező adatbázisokból készült kimutatásokra használjuk inkább.

A relációs adatbázisok általában nem egyetlen relációból, táblából állnak, hanem több tábla alkot egy adatbázist. Azt, hogy egy adatbázist hány táblára bontunk szét, vagy hány táblát fogunk össze egy adatbázisba, már az adatbázis megtervezésekor eldöntjük. A táblákra bontásnál az attribútumok közötti kapcsolat jelentősen befolyásolja, hogy mely oszlopok kerülnek egy táblába.

A relációs modellnek megfelelõen az adatok a relációs adatbázisokban relációkban tárolódnak, amelyeket a felhasználó táblák formájában érzékel. Minden reláció egyedekbõl (tuple, rekord) és jellemzõkbõl (mezõ) épül fel.

**Az adatbázis tervezésének 7 javasolt lépése van:**

**1. Követelménykezelés:** itt határozzuk meg az adatbázis célját. Gondoljuk át milyen információkhoz szeretnénk majd jutni az adatbázisból. Tudnunk kell, hogy melyek azok az adatok amelyeket tárolnunk kell az egyedről.

**2. Egyedek, táblák meghatározása:** az összegyűjtött adatok rendszerezése után információrendszerbe kell őket szervezni. Az információrendszer egyedekkel foglalkozik. Az egyedek tárolása fizikailag egy táblában történik. A tábla soraiba (rekordjaiba) kerülnek az egyedpéldányok, a rekord mezőibe (oszlopokba) pedig az attribútumok. Minden adatot csak egy táblában tároljunk. Erre azért van szükség, hogy a későbbi módosításkor csak egy helyen kelljen frissíteni az adatokat. Az egy adott témára vonatkozó adatok egy táblában tároljuk.

**3. Mezők, attribútumok meghatározása:** itt tervezzük meg konkrétan a táblákat. Feladatunk meghatározni a táblákat felépítő mezőket.

**Az attribútumokat többféleképpen osztályozhatjuk:**

a) egyszerű, azaz tovább nem bontható, valamint összetett. Az összetett attribútum több egyszerű értékből áll.

**b) egyértékű:** minden előfordulásnál csak egy értéket vehet fel. A többértékű minden előfordulásnál több értéket is felvehet.

c) a tárolt attribútum értékeit az adatbázis tárolja. A származtatott értéke más attribútumok alapján határozható meg.

**4. Azonosítók meghatározása:** a táblákban tárolt adatokat egyértelműen kell azonosítani. Elsődleges kulcsra minden olyan táblában szükség van, amelynek rekordjait egyenként szeretnénk azonosítani. Az elsődleges kulcs olyan azonosító, amelynek értékei nem ismétlődhetnek az adott táblában. Az elsődleges kulcsnak fontos szerepe van a relációs adatbázisokban. Segítségével növelhetjük a hatékonyságot, gyorsítja a keresést és az adatok összegyűjtését. Az Accessben háromféle elsődleges kulcs alkalmazható:

**1) számláló típusú:** ez a leggyakrabban alkalmazott. Ilyenkor létre kell hozni egy Számláló típusú mezőt. Az Access minden egyes új rekord számára egyedi sorszámot generál.

**2) egy mezőből álló elsődleges kulcs:** a kulcs nem számláló típusú, ha nem tartalmaz egyetlen ismétlődő értéket sem (például adószám esetén).

**3) több mezőből álló elsődleges kulcs:** ilyen kulcsot több mező felhasználásával képezünk. Erre akkor kerül sor, ha egyetlen mező egyediségét sem tudjuk biztosítani.

**5. Kapcsolatok meghatározása:** a táblák rekordjait kapcsoljuk össze az elsődleges kulcsmezők segítségével. A kapcsolat 2 egyed összetartozását jelenti. A kapcsolat számosságát három csoportba oszthatjuk: az egyedviszonyokat az előző fejezetek egyikében már tárgyaltuk, azért itt most tényleg csak felsorolás történik.

**1. Egy az egyhez kapcsolat.**

**2. Egy a többhöz kapcsolat.**

**3. Több a többhöz kapcsolat.**

**6. Ellenőrzés:** A mezők, táblák és kapcsolatok megtervezése után meg kell nézni a tervet, hogy nem maradt-e benne hiba. A kezdeti stádiumban könnyebb az adatbázis tervét módosítani, mint amikor már fel van töltve adatokkal.

**7. Adatbevitel:** a szükséges javítások elvégzése után vigyük be az adatokat a már létező táblákba. Kialakíthatjuk továbbá a többi objektumot. Van lehetőség űrlapok, jelentések és lekérdezések készítésére!

# Az adatbázis követelményei:

A phpMyAdmin felülete teljes egészében böngészőn alapul, szükség lesz egy webszerverre -

például: Apache, nginx, IIS - a phpMyAdmin fájlok telepítéséhez. Továbbá szükség van egy böngészőre, amely a Bootstrap 5.2.0-ás számú verzióját támogatja!

A használathoz egy PHP 7.2.5 vagy újabb, támogatással rendelkező PHP könyvtárra van szükség (SPL) kiterjesztéssel, kivonattal, ctype-al, valamint JSON-támogatással (session).

A phpMyAdmin Windows rendszeren történő beszerzésének legegyszerűbb módja egy, a harmadik féltől származó termékek használata, amely tartalmazza a phpMyAdmin-t egy adatbázissal és webszerverrel, például egy XAMPP-al együtt.

Nem alkalmaz semmilyen speciális biztonsági módszert az adatbázis-kiszolgáló! Az engedélyek megadása a kiszolgáló weboldalán, a Felhasználók oldala menüben használható.

# Az adatbázis tervezési folyamata:

1. **Az adatbázis céljának meghatározása:**   
   Ez felkészíti a további lépésekre.
2. **A szükséges információk megkeresése és rendszerezése:**   
   Gyűjtsön össze minden információt, amelyet rögzíteni szeretne az adatbázisban, például a termékek nevét és rendelési számát.
3. **Az információk táblákra osztása:**   
   Az információt ossza fel nagyobb entitásokra vagy témákra (például Termékek és Rendelések). Ezután minden témából tábla lesz
4. **Információs elemek oszlopokká alakítása:**   
   Döntsük el, milyen információkat szeretnénk tárolni az egyes táblákban. Minden elem mezővé alakul, és a táblában oszlopként fog megjelenni. Egy Alkalmazottak nevű táblában például a Vezetéknév és a Felvétel dátuma mezők szerepelhetnek.

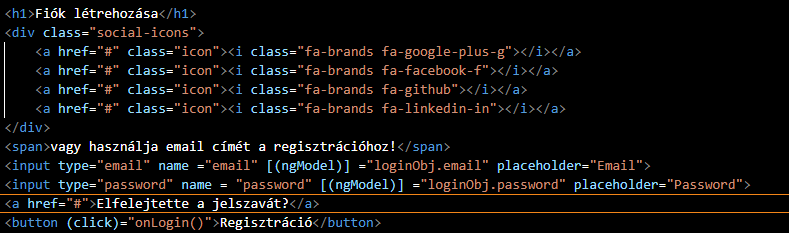
* **Elsődleges kulcsok megadása**   
  Határozza meg az egyes táblák elsődleges kulcsát. Az elsődleges kulcsok az egyes sorokat egyedileg azonosító oszlop. Elsődleges kulcs lehet például a Termékazonosító vagy a Rendelésazonosító mező.
* **Táblakapcsolatok beállítása**   
  A táblákat áttekintve döntse el, hogyan kapcsolódnak az egyes táblák adatai a többi tábla adataihoz. Ha egyértelművé szeretné tenni a táblák kapcsolatát, létrehozhat új mezőket a táblákban, vagy létrehozhat új táblákat.
* **A terv finomítása**   
  Győződjön meg róla, hogy hibáktól mentes a terv. Hozza létre a táblákat, és adjon hozzá néhány rekordnyi mintaadatot. Ellenőrizze, hogy a táblák a kívánt eredményeket adják-e. Szükség esetén módosítsa a tervet.
* **A normalizációs szabályok alkalmazása**   
  Az adatnormalizációs szabályokat alkalmazva ellenőrizze, hogy a táblák szerkezete megfelelő-e. Módosítsa a táblákat, ha szükséges.

## Regisztráció, bejelentkezés, kijelentkezés :

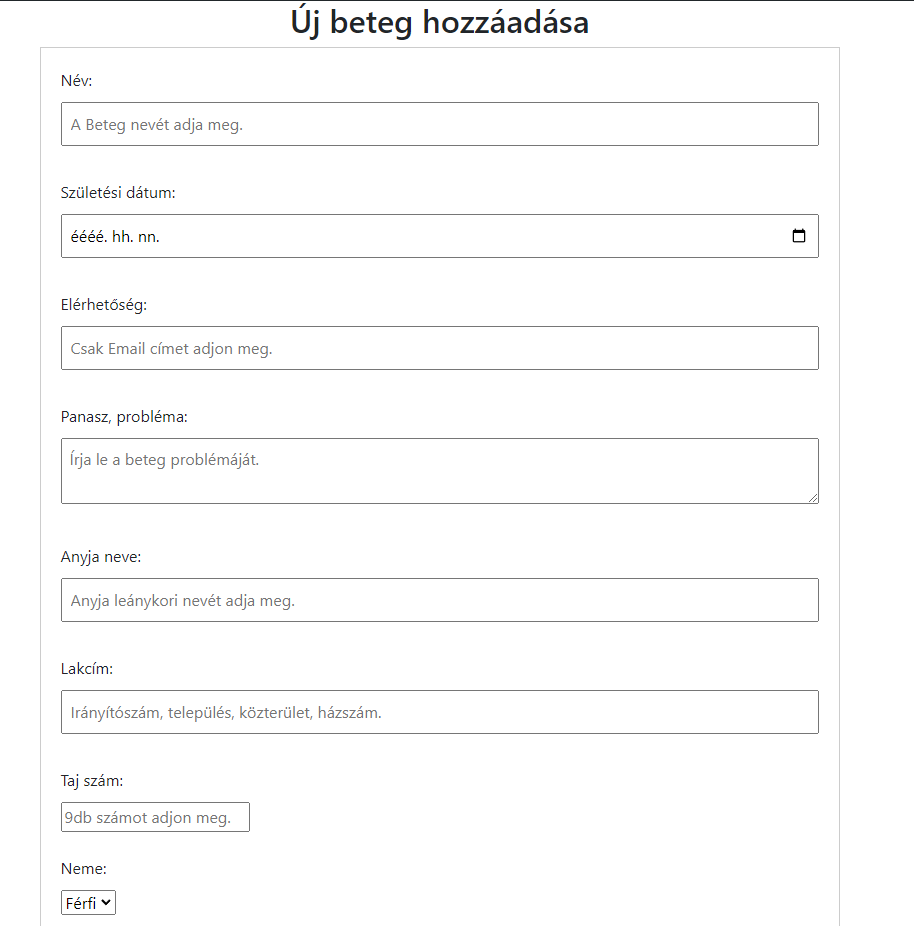
**Bevezetés:**

Verzió: 1.0

A regisztráció elkészítése során figyelembe került egy olyan szempont, hogy a felület az orvosok számára lett kialakítva, amelyben van lehetőség a személyes profil létrehozására! A regisztráció során ha az orvos a bejelentkezésnél elfelejtette a jelszavát, akkor lehetőség van egy új jelszó kérésére! Egy, a már meglévő profil esetén egy “Bejelentkezés” gomb került kialakításra, ahol a fogorvosok megadhatják a személyes adataikat a betegnyilvántartó használata előtt! Ez a dokumentáció leírja a bejelentkezési felületet, amelyet Angular keretrendszerrel lett létrehozva! A felület kialakításához az app.component.html, és a hozzátartozó stílusokat tartalmazó app.component.css fájlokat használtuk. A dokumentáció a 2024.04.03-i verzióra vonatkozik.



Az alábbi kódrészlet a bejelentkezés folyamatát, valamint az új jelszó kérést mutatja be!



Betegnyilvántartó Frontend Dokumentáció

A betegnyilvántartó frontend alkalmazás egy Angular keretrendszerben készült alkalmazás, amely lehetővé teszi új betegek felvételét, a már regisztrált betegek szerkesztését és törlését. Az alkalmazás továbbá lehetővé teszi a betegek adatainak helyi tárolását a böngésző localStorage-jében.

Betegnyilvántartó Funkciók

Betegfelvétel:

Az alkalmazásban elérhető funkciók közül az egyik legfontosabb a betegfelvétel. Az űrlap kitöltésével könnyen és gyorsan lehet új betegeket felvenni az alkalmazásba. A betegfelvétel során a felhasználóknak meg kell adniuk bizonyos alapvető információkat a betegről, hogy azokat később könnyen kezelni és nyomon követni lehessen.

A következő adatokat kell megadni a beteg felvételéhez:

Név: A beteg teljes neve.

Születési dátum: A beteg születési ideje, amely fontos információ a beteg kezeléséhez és nyomon követéséhez. A születési dátum alapján az orvosok megállapíthatják a beteg életkorát, ami kulcsfontosságú lehet a diagnózisban és a kezelési terv kidolgozásában. Például bizonyos betegségek kockázata vagy a kezelés hatékonysága függhet az életkortól. Emellett a születési dátum alapján lehetőség van az évfordulók, rendszeres vizsgálatok vagy oltások emlékeztetőinek beállítására az egészségügyi rendszerben.

Elérhetőség: Az elérhetőségi adatok rögzítése fontos lépés a betegnyilvántartásban, mivel lehetővé teszi az orvosok számára, hogy könnyen kapcsolatba lépjenek a beteggel. Az elérhetőségi adatok lehetnek telefonos számok, e-mail címek vagy akár postai címek is, amelyek révén az egészségügyi szakemberek tájékoztatást kaphatnak a beteg állapotáról, további vizsgálatokról vagy kezelésekről.

A telefonszámok és e-mail címek gyors és hatékony kommunikációt tesznek lehetővé az orvosok és a betegek között. Például egy sürgős esetben az orvosok telefonon keresztül értesíthetik a beteget azonnali intézkedésekről vagy sürgős konzultációkról. Az e-mail címek pedig lehetővé teszik az írásos kommunikációt, például vizsgálati eredmények vagy kezelési útmutatók küldését.

Fontos, hogy az elérhetőségi adatokat bizalmasan és biztonságosan kezeljük, hogy megfeleljenek az adatvédelmi előírásoknak és a betegek biztonságának. Az alkalmazásnak megfelelő biztonsági intézkedéseket kell bevezetnie az ilyen érzékeny adatok védelme érdekében, például titkosítási technológiák vagy hozzáférési korlátozások segítségével.

Az elérhetőségi adatok rögzítése és rendszerezése segít az egészségügyi szolgáltatóknak abban, hogy hatékonyan kommunikáljanak a betegekkel, gondoskodjanak róluk és biztosítsák, hogy a betegek számára mindig elérhető legyen az egészségügyi ellátás és támogatás.

Panasz, Probléma: A beteg panaszai vagy problémái, amelyek alapján az orvosok diagnózist és kezelési tervet készíthetnek.

Anyja neve: A beteg anyjának neve, ami egy fontos azonosító adat.

Lakcím: A beteg állandó lakcíme, amely fontos lehet például postai levelezés vagy otthoni látogatások során.

TAJ szám: A beteg egészségügyi azonosítószáma, amely egyértelműen azonosítja a beteget az egészségügyi rendszerben.

Nem: A beteg nemének rögzítése az egyik alapvető információ a betegnyilvántartásban, amely fontos szerepet játszik a beteg kezelése és gondozása során. A beteg nemének megadása lehetővé teszi az egészségügyi szakemberek számára, hogy figyelembe vegyék a nem-specifikus kockázati tényezőket, valamint az esetleges biológiai különbségeket a kezelés során.

Az egyes nemeknek sajátos egészségügyi kockázataik és igényeik vannak, ezért fontos, hogy az egészségügyi rendszer lehetőséget biztosítson a betegek nemének nyilvántartására. Például bizonyos betegségek vagy egészségügyi állapotok előfordulási gyakorisága különbözhet a nemek között, így az orvosoknak és a kezelési tervet kidolgozó szakembereknek figyelembe kell venniük ezeket az eltéréseket a megfelelő diagnózis és kezelés érdekében.

Az alkalmazásban a beteg nemének rögzítése lehetővé teszi az egészségügyi szakemberek számára, hogy személyre szabott és hatékony kezelési tervet dolgozzanak ki a betegek számára. Például bizonyos gyógyszerek hatékonysága vagy mellékhatásai eltérőek lehetnek a nemek között, ezért fontos, hogy az orvosok ezeket az információkat figyelembe vegyék a kezelés során.

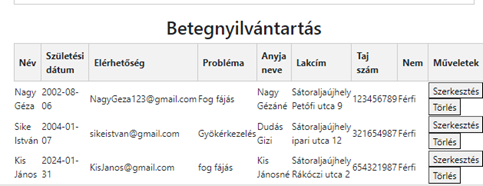
Az alkalmazásban a beteg nemének rögzítése általában egy egyszerű legördülő menü vagy választógombok segítségével történik, amely lehetővé teszi a felhasználók számára, hogy könnyen kiválasszák a megfelelő lehetőséget. Fontos, hogy az alkalmazás biztosítsa a nem-bináris vagy egyéb identitásokat is lehetőséget a megadásra, hogy minden beteg számára inkluzív és tiszteletteljes legyen az egészségügyi ellátás.

Az egészségügyi rendszerben a betegek nemének rögzítése fontos lépés a betegek egészségének és jólétének megőrzése érdekében, és segíti az egészségügyi szakembereket abban, hogy hatékonyan kezeljék és támogassák a betegeket az egészségügyi problémáik megoldásában.

Ezek az adatok lehetővé teszik az orvosok számára, hogy hatékonyan kezeljék és nyomon kövessék a betegeket az alkalmazásban. A betegfelvétel során rögzített adatok segíthetnek a pontos diagnózisban és a hatékony kezelési terv kidolgozásában, elősegítve ezzel a betegek gyorsabb felépülését és egészségük javítását. Az űrlap egyszerű és felhasználóbarát felépítése segíti a felhasználókat a gyors és pontos adatbevitelben, ezáltal minimalizálva a hibalehetőségeket és növelve az alkalmazás hatékonyságát és felhasználói elégedettségét.

Felhasznált Technológiák

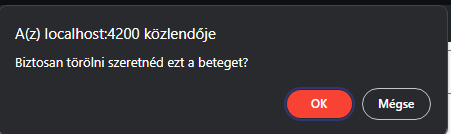
Az Angular egy alkalmazástervező keretrendszer és fejlesztői platform hatékony és kifinomult egyoldalas alkalmazások létrehozásához. Ezek az Angular dokumentumok segítenek az Angular keretrendszer és fejlesztői platform elsajátításában és használatában, az első alkalmazástól a bonyolult egyoldalas alkalmazások optimalizálásáig a vállalatok számára. Az oktatóanyagok és útmutatók letölthető példákat tartalmaznak a projektek elindításához.

Beteg adatainak szerkesztése

A már regisztrált betegek adatait lehet szerkeszteni.

Az adatok szerkesztése a megfelelő beteg sorában található "Szerkesztés" gombra kattintva történik.

Beteg adatainak törlése

A betegtörlés funkció célja az alkalmazásból már regisztrált betegek eltávolítása. A funkció előtt a felhasználót megerősítő kérelem jelenik meg, hogy biztosan törölni szeretné-e a beteget. Ez segít megelőzni véletlen törléseket, és lehetővé teszi a felhasználó számára, hogy megerősítse a döntését. A betegtörlés gondosan megvalósított mechanizmusának köszönhetően biztosítjuk a betegek adatainak védelmét és a felhasználói élményt.

CSS

Az alkalmazásban a betegek adatait egy JavaScript tömbben tároljuk, amely lehetővé teszi az adatok dinamikus kezelését és manipulálását. Az adatokat a böngésző localStorage-jében is tároljuk, így biztosítva, hogy a felhasználók adatai ne vesszenek el a böngészési unkamenetek között. Ezáltal a felhasználók számára kényelmes és zökkenőmentes élményt nyújtunk.

Az alkalmazás frontend részét egy űrlap segítségével valósítjuk meg. Az űrlap beküldésekor a JavaScript kód validálja az adatokat, hogy biztosítsa az adatok megfelelőségét és érvényességét. Ez fontos lépés az adatok megbízhatóságának és konzisztenciájának biztosításához. A megfelelő adatokkal frissítjük a betegek táblázatát és a localStorage-t, hogy az alkalmazás mindig naprakész legyen és a legfrissebb adatokat jelenítse meg a felhasználóknak.A betegek táblázatát dinamikusan frissítjük azáltal, hogy figyelemmel kísérjük az űrlap beküldését, új betegek hozzáadását, meglévő betegek szerkesztését vagy törlését. Ez a megközelítés lehetővé teszi az alkalmazás számára, hogy valós időben reagáljon a felhasználói interakciókra, és azonnal frissítse az adatokat a felhasználók számára.

## Használt csoportmunka eszközök bemutatása:

**Trello:** A Trello egy web alapú, Kanban stílusú listakészítő alkalmazás. A felhasználók különböző oszlopokkal hozhatják létre a feladat táblákat, és áthelyezhetik a feladatokat közöttük. Az oszlopok általában olyan feladat állapotokat tartalmaznak, mint a Teendő, Folyamatban, Elkészült. A Trello egy olyan projektmenedzsment szoftver és együttműködést segítő eszköz, amelyben könnyen szervezhetünk és követhetünk feladatokat, projekteket és teendőket egy virtuális táblán. Hasznos lehet egyéneknek és csapatoknak egyaránt, hogy átláthatóan szervezzék meg a munkájukat, együttműködjenek és kövessék a feladatok előrehaladását.



**GitHub:** A GitHub egy fejlesztői platform, amely lehetővé teszi a fejlesztők számára kódjuk létrehozását, tárolását, kezelését és megosztását. [Git](https://en.wikipedia.org/wiki/Git) szoftvert használ, amely biztosítja a Git plusz [hozzáférés-](https://en.wikipedia.org/wiki/Access_control) [vezérlés elosztott verziókezelését](https://en.wikipedia.org/wiki/Distributed_version_control), [hibakövetést](https://en.wikipedia.org/wiki/Bug_tracking_system), [szoftverfunkciókéréseket](https://en.wikipedia.org/wiki/Software_feature), [feladatkezelőt](https://en.wikipedia.org/wiki/Task_management), [folyamatos integrációt](https://en.wikipedia.org/wiki/Continuous_integration) és [wikiket](https://en.wikipedia.org/wiki/Wiki) minden projekthez. Székhelye Kaliforniában található[,](https://en.wikipedia.org/wiki/California) 2018 óta [a Microsoft](https://en.wikipedia.org/wiki/Microsoft) leányvállalata.

****

**phpMyAdmin:** A phpMyAdmin támogatja a MySQL-kompatibilis adatbázisokat.

**Követelmények:**

MySQL 5.5 vagy újabb,

MariaDB 5.5 vagy újabb verzió.

**Szolgáltatások:**

* adatbázisok, táblák oszlopok és indexek létrehozása, másolása, eldobása, átnevezése és módosítása,
* több eredményhalmaz megjelenítése tárolt eljárásokon és lekérdezéseken keresztül,
* InnoDB táblák és idegen kulcsok támogatása,
* [PDF](https://docs.phpmyadmin.net/hu/latest/glossary.html#term-pdf) ábrák készítése az adatbázis elrendezéséről,
* InnoDB táblák és idegen kulcsok támogatása,
* a szerver, az adatbázisok és táblák karbantartása a kiszolgáló beállításai szerint.



**Apache MySQL:** Az Apache webszerver a világ legnépszerűbb webszerverei közé tartozik. Jól dokumentált, aktív felhasználói közösséggel rendelkezik, és az internet történetének nagy részében széles körben használták, ami nagyszerű választás a webhelyek tárolására.



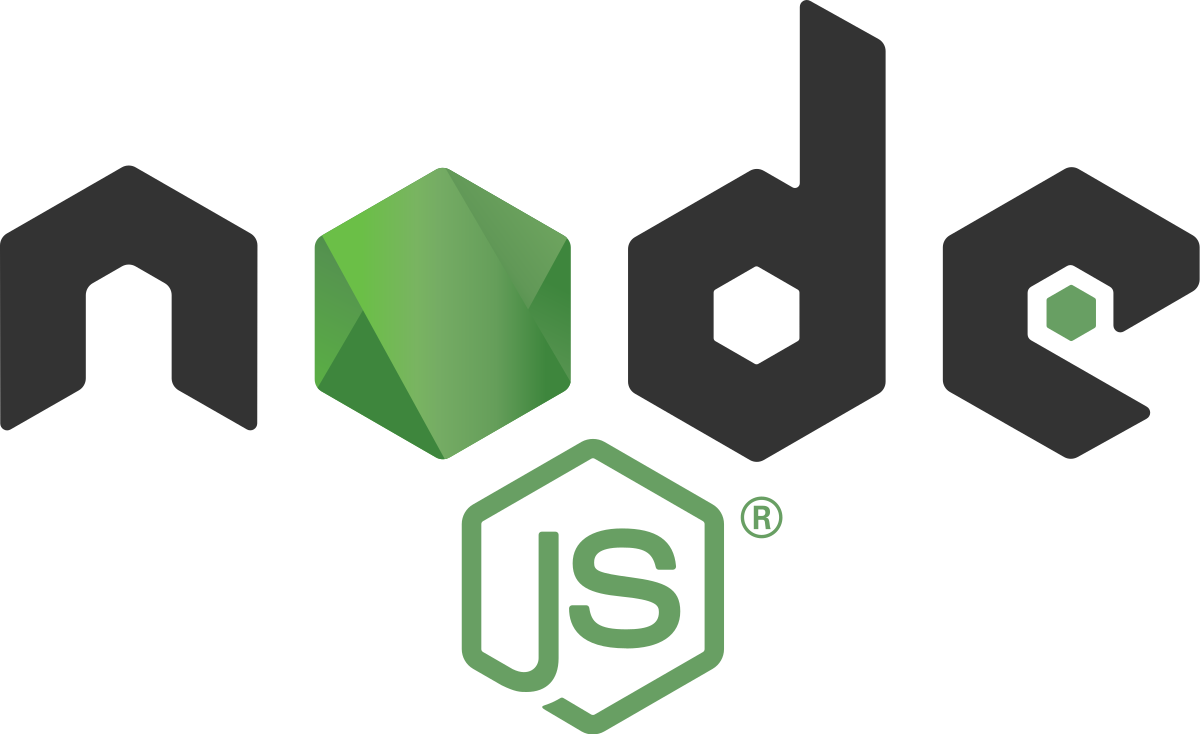
## 

## Fejlesztéshez használt technológiák, szoftverek:

**SceneBuilder:** A JavaFX Scene Builder egy vizuális elrendezési eszköz, amellyel a felhasználók gyorsan, kódolás nélkül tervezhetik meg a JavaFX alkalmazások felhasználói felületét. A felhasználók áthúzhatják a felhasználói felület összetevőit egy munkaterületre, módosíthatják tulajdonságaikat, stíluslapokat alkalmazhatnak.

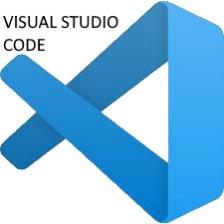


**Node.js:** A Node.js egy nyílt forráskódú, többplatformos JavaScript futási környezet és könyvtár webalkalmazások futtatásához az ügyfél böngészőjén kívül. A fejlesztők a Node.js-t használják szerveroldali webalkalmazások létrehozásához, és ez tökéletes adatintenzív alkalmazáshoz, mivel aszinkron eseményt használ.





**XAMPP Control Panel:** A projekt során a 3.3.0-ás verzió került felhasználásra. A XAMPP egy teljesen ingyenes, könnyen telepíthető Apache disztribúció MariaDB, PHP és Perl komponenseket tartalmazva. A XAMPP egy integrált rendszer, mely a webes alkalmazások készítését, tesztelését és futtatását oldja meg. Óriási előnye, hogy egyetlen egy csomagban tartalmazza az összes elengedhetetlen eszközt, amely szükséges a webes alkalmazások készítéséhez, így nem kell egyesével összevadászni őket az internetről, illetve telepíteni. Ez az előnye, hogy minden szoftver egyetlen egy telepítőbe van csomagolva, rengeteg időt spórolhat meg számunkra.

**Visual Studio Code:** A Visual Studio Code egy könnyűsúlyú, de erőteljes fejlesztői környezet, amelyet a webfejlesztők gyakran használnak. Segítségekkel kiterjesztésekkel és gyorsan működő debuggolási funkciókkal rendelkezik, amelyek segítik a hatékony fejlesztést. A projekt során az 1.85.0 számú verzió került felhasználásra.

**HTML (HyperText Markup Language):** Az alapvető építőköve a weboldalaknak. Szöveges tartalmak strukturálására, linkek, képek és egyéb elemek beillesztésére használjuk. A fogorvosi rendelő oldalon fontos az információk tiszta és könnyen érthető megjelenítése, amit a HTML segítségével érhetünk el.



**CSS (Casvading Style Sheets):** felel a weboldal design-ért és megjelenéséért. A fogorvosi rendelő oldalon keresztül a CSS segítségével szabályozzuk a színeket, elrendezést, stílust, hogy vonzó és professzionális kinézetű legyen.



**Javascript:** A Javascript az interaktivitásért és dinamizmusért felelős. Az oldalon elhelyezett űrlapok, interaktív elemek, és egyéb dinamikus funkciók létrehozásához alkalmazzuk.

**Angular:** Az Angular egy olyan keretrendszer amely segít az egyoldalas alkalmazások (SPA) készítésében. Ezek az alkalmazások gyorsabbak és simábbak lehetnek a hagyományos weboldalakkal szemben. Az Angular segítségével könnyen kezelhetjük az adatokat, és egy összetett, dinamikus felhasználói élményt hozhat létre.



**Paint.Net:** A Paint.Net egy ingyenes grafikai szerkesztő alkalmazás, amelyet kifejezetten a Windows operációs rendszerhez terveztek. Bár nem rendelkezik a Photoshop összetettségével vagy a GIMP kiterjedt funkcióival, a Paint. Net könnyen kezelhető felhasználói felülete és számos alapvető és középszintű képszerkesztési eszköze miatt nagyon népszerű.

# 

# Irodalomjegyzék

A jegyzékben felhasznált weboldalak, valamint a hozzáférések a pontos dátumokkal és az időpontokkal:

A felhasználói dokumentációban használt weboldalak:

Oracle.com

https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html

2023.11.24: 08 óra 15 perc

Wikipedia.org

https://hu.wikipedia.org/wiki/Trello

2023.11.24: 08 óra 10 perc

phpmyadmin.net

https://www.phpmyadmin.net

2023.11.24: 8 óra 22 perc

Oracle.com

https://www.oracle.com/mysql/what-is-mysql/

2023.11.24: 8 óra 32 perc

Simplilearn.com

<https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs>

2023.11.24: 8 óra 52 perc

DigitalOcean.com

https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-22-04

2023.11.24: 09 óra 05 perc

NekoMajin42.io

<https://nekomajin42.github.io/bevinfo/pages/database/index.html>

2023.11.26 17 óra 02 perc

RS1.sze.hu

<http://rs1.sze.hu/~takacsg/oktatas/lev/GKLB_INTM010-Adatbazisok/SQL_olvasson.pdf>

2023.11.26 17 óra 31 perc

Tudasbazis.sulinet.hu

<https://tudasbazis.sulinet.hu/hu/szakkepzes/informatika/adatbazis-kezeles/adatbazis-kezeles-alapfogalmai/alapfogalmak-tabla-rekord-mezo-egyed-attributum>

2023.11.26 17 óra 33 perc

Aries.EKTF.hu

<https://aries.ektf.hu/~hz/pdf-tamop/pdf-xx/Radvanyi-hdbms-HU.pdf>

2023.11.26 17 óra 39 perc

XAMPP magyarázat

<https://www.apachefriends.org/hu/index.html>

2023.12.05 8 óra 18 perc

Paint.NET:

<https://www.getpaint.net/>

2023.12.16 17 óra 15 perc

Support.Microsoft.com

<https://support.microsoft.com/hu-hu/topic/az-adatb%C3%A1zisok-tervez%C3%A9s%C3%A9nek-alapjai-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5>

2024.01.26. 08 óra 15 perc

Foglaljorvost.hu

<https://foglaljorvost.hu/fogorvos/budapest>

2024.01.31. 08 óra 11 perc

Apollonia.hu

https://apollonia.hu/

2024.01.31. 08 óra 45 perc

MediCoverDental.hu

<https://medicoverdental.hu/kapcsolat/>

2024.01.31. 09 óra 03 perc

Neak.GOV.hu

<https://www.neak.gov.hu/pfile/file?path=/letoltheto/altfin_dok/altfin_virt_dok2/Alapellatas/Fogkodlista_2020.03.15.01-tol&inline=true>

2024.02.01. 08 óra 20 perc

SubaDental.com

<https://subadental.com/hu/szolgaltatasaink/parodontologia-fogagybetegsegek>

2024.02.08. 08 óra 27 perc

phpMyAdmin dokumentációja, bevezetője

<https://docs.phpmyadmin.net/hu/latest/require.html#database>

2024.02.09. 11 óra 05 perc

<https://docs.phpmyadmin.net/hu/latest/intro.html>

2024.02.09. 11 óra 10 perc

DrBorbathMosolykozpont.hu

<https://www.drborbathmosolykozpont.com/pdf/dr_med_dent_mom.pdf>

2024.02.13. 08 óra 11 perc

Websterdds.com

<https://websterdds.com/12-types-of-dental-specialties/>

2024.02.16. 08 óra 21 perc

Wikipedia.org

[**https://en.wikipedia.org/wiki/GitHub**](https://en.wikipedia.org/wiki/GitHub)

2024. 02. 27. 08 óra 55 perc

Webiskola.hu

<https://webiskola.hu/php-ismeretek/php-apache-xampp-letoltese-telepitese/>

2024.02.28 12 óra 45 perc

kiszervezettmarketing.hu

<https://kiszervezettmarketing.hu/online-marketing/trello/>

2024.02.28 12 óra 52 perc

wikipedia.org

<https://en.wikipedia.org/wiki/MySQL>

2024.02.28 13 óra 00 perc

online.visual-paradigm.com

<https://online.visual-paradigm.com/diagrams/solutions/free-online-diagram-maker/>

2024.03.05 09 óra 20 perc

phpMyAdmin dokumentációja

[Üdvözöljük a phpMyAdmin dokumentációjában! — phpMyAdmin 5.2.1 dokumentáció](http://localhost/phpmyadmin/doc/html/index.html)

2024.03.18 12 óra 24 perc

Regisztrációs felület

<https://www.youtube.com/watch?v=5QM_GK-pFpM>

2024.03.19 11 óra 28 perc

Regisztráció ingyenesen felhasználható kódrészlete

<https://codepen.io/NTIAKO/pen/wvOqjxm>

2024.03.19 11 óra 31 perc

medicoverdental.hu

<https://medicoverdental.hu/kezeleseink/>

2024.03.22 08 óra 50 perc

angular.io

<https://angular.io/guide/what-is-angular>

2024.03.27. 09 óra 05 perc

W3Schools.com

<https://www.w3schools.com/bootstrap4/tryit.asp?filename=trybs_navbar_color&stacked=h>

2024.04.04 11 óra 34 perc

Frontend és Backend közötti kommunikáció kódja

<https://www.youtube.com/watch?v=leMy-yvU9tY>

2024.04.08 09 óra 39 perc

angular.io

https://angular.io/guide/http-setup-server-communication

2024.04.08 09 óra 49 perc

A W3 schools placeholderei

<https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_input_placeholder>

2024.04.05 11 óra 14 perckor