

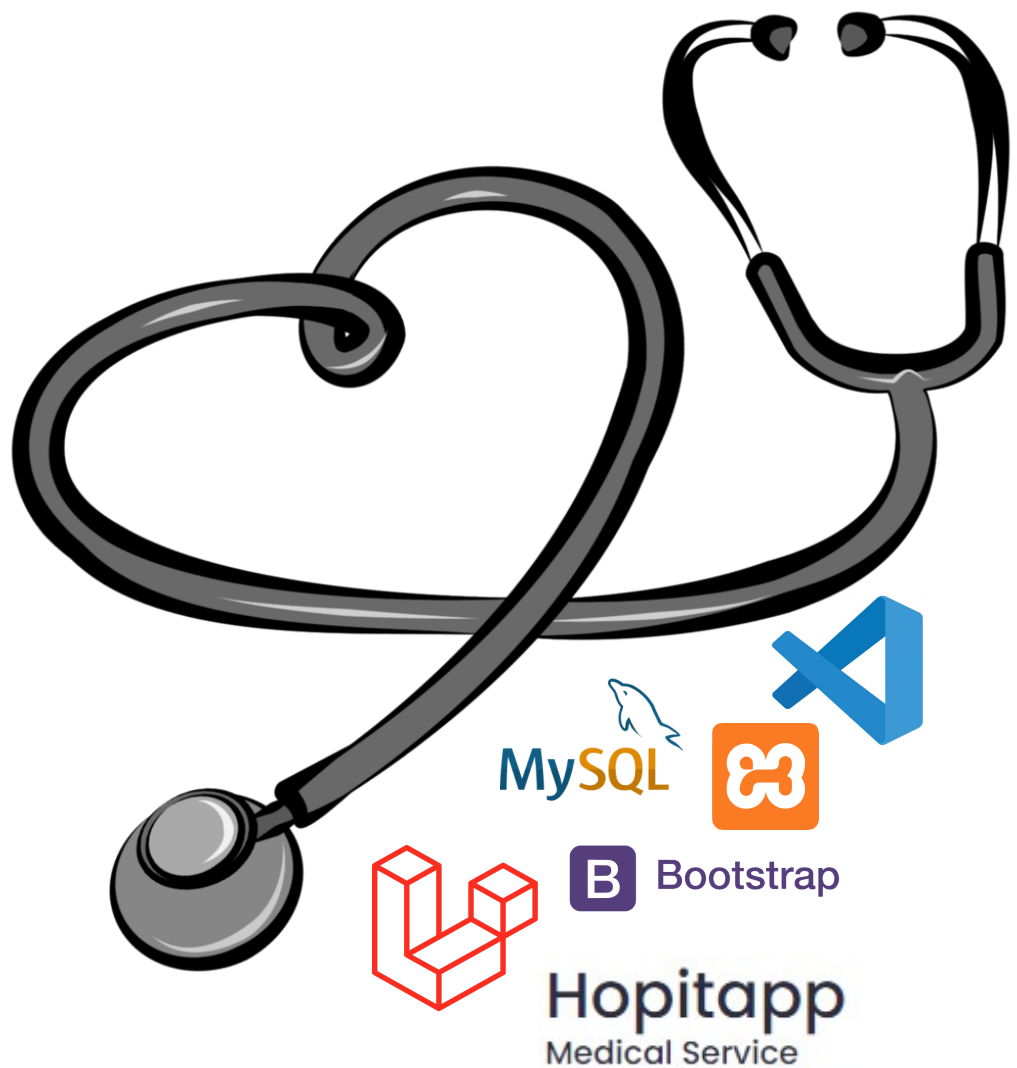
Soltész Tamara

SZOFTVERFEJLESZTŐ ÉS -TESZTELŐ TECHNIKUS

VVSzC Gépipari és Informatikai Technikum

Vizsgaremek

2023



Tartalom

Összefoglaló	3
Program	4
Laravel	4
Bootstrap	5
Adatbázis (MySQL)	6
Adatbázis felépítése	7
Osztályok és metódusok	9
Model	9
View	10
Controller	11
Fejlesztési lehetőségek	13
Felhasználói dokumentáció	14
Vendégeknek	14
Adminoknak	17
GitHub elérhetőség	19
Hivatkozások	20

Összefoglaló

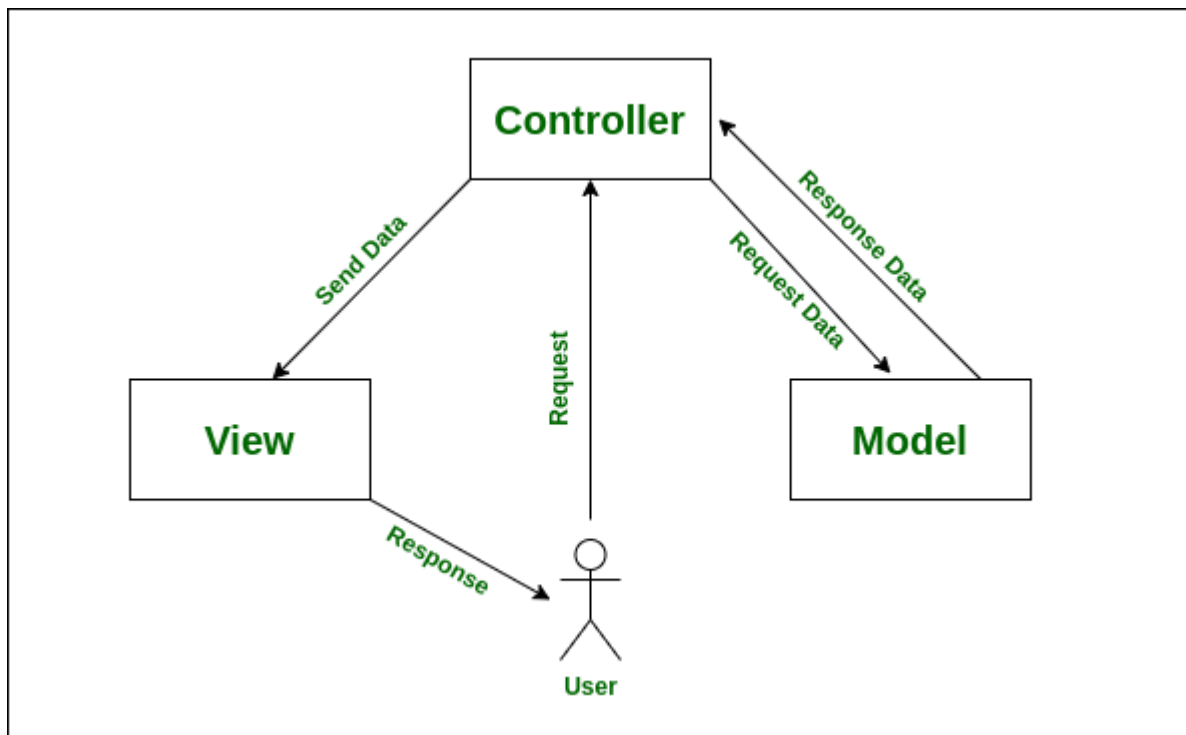
A Hopitapp egy webes felületű, kórházi időpontfoglaló alkalmazás, amely Laravel-ben készült.

Az oldalt 3 szemszögből lehet megtekinteni: bejelentkezés nélkül, bejelentkezve, mint sima felhasználó és bejelentkezve, mint admin user. A három felület közül csak az utóbbi tér el a többitől, ugyanis az admin felhasználó erősebb jogkörrel rendelkezik, például adatokat tud törölni a honlapon keresztül. A hozzá tartozó színskema is más, mint amit a többi felhasználó láthat, ugyanis míg az oldal bárki által megtekinthető része világos témájú, addig az admin felület sötét megjelenésben pompázik. Ezekhez az oldalakhoz más profiloknak nincsen hozzáférése.

A regisztrált és nem bejelentkezett megtekintés közötti különbség nem ennyire erős, de látható. Először is ha nem vagyunk bejelentkezve, a fejlécen egy lenyíló menü helyett két gomb helyezkedik el, a "login" és a "register", valamint nem lehet időpontfoglalást intézni.

Program

Az alkalmazás követi a már jól ismert MVC (Model-Controller-View) alapelveit. A nézetek adatokkal való kommunikációja egy közbeépített osztály a vezérlő segítségével zajlik. Így könnyen elkülöníthetők az adatok, és egy esetleges fejlesztés esetén nem kell az egész rendszert újra írni, hanem csak az adott területhez tartozó részt.



MVC programstruktúra^[1]

Laravel^[2]

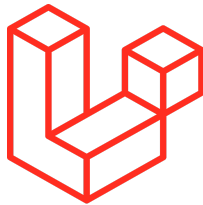
Laravel egy népszerű nyílt forráskódú PHP keretrendszer (framework), amelyet webalkalmazások és weboldalak fejlesztésére használnak. A Laravel célja, hogy a fejlesztőknek könnyű és hatékony eszközt nyújtson a modern, robusztus webalkalmazások készítéséhez.

A Laravel keretrendszer számos hasznos funkciót és eszközt tartalmaz, például:

- **Eloquent ORM:** Egy erőteljes adatbázis-kezelő réteg, amely lehetővé teszi az adatbázis táblák könnyű műveleteit objektumokkal és nyelvi kifejezésekkel.
- **Routing:** Rugalmas útvonalkezelő rendszer, amely segítségével könnyen definiálható az alkalmazás URL-útvonalai.
- **Middleware:** Köztes rétegek, amelyeket a HTTP kérések feldolgozása során alkalmazhatunk, például hitelesítés, naplózás vagy gyűjtés céljából.
- **Blade:** Egy intuitív sablon motor, amely segítségével könnyen készíthetsz és rendszerezhetsz HTML-sablonokat.

- **Authentikáció és hitelesítés:** Beépített eszközök az autentikáció és hitelesítés kezelésére
- **Artisan:** Egy parancssori eszköz, amely segítségével gyorsan és hatékonyan végrehajthatók a gyakori fejlesztési feladatok.

A program a framework legújabb, 10.20.2 verziójával készült, ami a php 8.2.4-et kezeli.



Bootstrap^[3]

Bootstrap egy ingyenes és nyílt forráskódú front-end keretrendszer, amelyet webfejlesztéshez és webdizájnhoz használnak. Ezt a keretrendszert arra tervezték, hogy segítse a fejlesztőket az egyszerűsített, hatékony és egységes módon történő webalkalmazások és weboldalak készítésében. A Bootstrapot eredetileg a Twitter fejlesztette.

A Bootstrap egy sor előre elkészített stíluslapokat és JavaScript komponenst tartalmaz, amelyeket könnyen hozzá lehet adni a weboldalakhoz és webalkalmazásokhoz.

Néhány jellemző tulajdonsága a Bootstrapnek:

- **Responsive tervezés:** A Bootstrap lehetővé teszi, hogy a weboldalak és webalkalmazások automatikusan alkalmazkodjanak a különböző eszközök képernyőméretéhez és típusához, például asztali számítógépekhez, táblagépekhez és okostelefonokhoz.
- **Könnyű testreszabhatóság:** A Bootstrap lehetővé teszi a stíluslapok és komponensek testreszabását.
- **Fejlesztőbarát:** Az API-dokumentáció és a széles körű közösség segíti a fejlesztőket a Bootstrap használatában és problémák megoldásában.

Bootstrap használata a webfejlesztésben időt és erőforrásokat takaríthat meg, mivel sok alapvető funkciót és dizájnt előre tartalmaz. Ez lehetővé teszi a fejlesztők számára, hogy gyorsabban fejlesszenek web alkalmazásokat és weboldalakat, miközben biztosítják a minőséget és az egyenletes kinézetet.

A projekt a 4.0 verziójú Bootstrapet használja.



Adatbázis (MySQL)^[4]

A program az adatokat egy MySQL adatbázisban tárolja és éri el.

MySQL egy nyílt forráskódú relációs adatbázis-kezelő rendszer, amelyet széles körben használnak az adatok tárolására és kezelésére a webalkalmazásokban és más szoftverrendszerekben. MySQL a világ egyik legnépszerűbb adatbázis rendszere, és számos alkalmazás és webhely alapját képezi. Itt vannak néhány fő tulajdonsága és jellemzője:

- **Relációs adatbázis:** MySQL egy relációs adatbáziskezelő rendszer, amely azt jelenti, hogy az adatokat táblákban tárolja, és azok között relációk vannak. Ez lehetővé teszi az adatok strukturált tárolását és hatékony lekérdezésüket.
- **Nyelv és SQL-támogatás:** MySQL támogatja a SQL nyelvet, amely segítségével lekérdezéseket és műveleteket lehet végrehajtani az adatokon. A SQL az adatbázisrendszerek közös lekérdező nyelve.
- **Teljesítmény és skálázhatóság:** MySQL nagyon gyors és hatékony adatbázis, és lehetővé teszi a skálázhatóságot, hogy nagyobb terheléssel is megbirkózzon. Alkalmas kisméretű weboldaltól nagyvállalati alkalmazásokig terjedő projektekhez.
- **Platformfüggetlenség:** MySQL többféle platformon futtatható, például Linux, Windows, macOS.
- **Közösség és támogatás:** MySQL széles körű közösségi támogatással rendelkezik, így könnyű segítséget és dokumentációt találni a problémák megoldásához és fejlesztéshez.



Az adatbázis felépítése

id	name	email	speciality	image
6	Dr Laura Mayer	lm@hopitapp.com	Cardiologist	1696510349.png
7	Dr John Harrington	jh@hopitapp.com	Gynecologist	1696510431.png
8	Dr Juan Carlos	jc@hopitapp.com	Dentist	1696510519.png
10	Dr Monica Lyn	ml@hopitapp.com	Surgeon	1696510671.png

A Doctors tábla elemei

A Doctors tábla az intézményben dolgozó orvosok neveit és elérhetőségeit tartalmazza.

- **id:** az orvos egyedi azonosítója
- **name:** az orvos teljes neve
- **email:** az orvos elérhetősége
- **speciality:** az orvos szakterülete
- **image:** az orvos fényképét tartalmazó fájlnek a neve. Ezek a képek a \Hopitapp\public\doctorimage mappába kerülnek bele feltöltés után

id	name	description	image
4	Dental service	Tooth whitening, orthodontic treatment	1696511701.PNG
5	Surgery	General, plastic surgery	1696511765.PNG
6	Cardiology	Everything with heart	1696511885.PNG

A Service tábla elemei

A Service tábla a szolgáltatásokat tartalmazza.

- **id:** a szolgáltatás egyedi azonosítója
- **name:** a szolgáltatás neve
- **description:** a szolgáltatás leírása
- **image:** a szolgáltatáshoz tartozó kis ikon fájl neve Ezek a képek a \Hopitapp\public\serviceimage mappába kerülnek bele feltöltés után

id	name	email	phone	address	usertype
2	admin	admin@admin.com	123456789	admin st 1	1
3	nonadmin	nonadmin@nonadmin.com	987654321	nonnononon	0

A Users tábla elemei

A Users tábla a regisztrált felhasználókat tartalmazza.

- **id:** a felhasználó azonosítója
- **name:** a felhasználó teljes neve
- **email:** a felhasználó email címe
- **phone:** a felhasználó telefonszáma helyett
- **address:** a felhasználó lakcíme
- **usertype:** a felhasználó típusa. Ha 0, akkor páciens; ha 1 akkor admin jogosultság körű. (Az ok, amiért nem bináris értékű a mező az, hogy esetleges fejlesztés esetén az orvos felhasználókat is lehessen külön kategóriába sorolni)
- **password:** A felhasználói profilhoz tartozó jelszó titkosítva

Az adatbázissal való kommunikációt a Laravel keretrendszer biztosítja, már beépített metódusokkal.

Ezek mellett a vizsgaremek elkészítéséhez még a XAMPP^[5] nevű programot használtam, mely egy szabad és nyílt forrású platformfüggetlen webszerver-szoftvercsomag, amelynek legfőbb alkotóelemei az Apache webszerver, a MariaDB (korábban a MySQL) adatbázis-kezelő, valamint a PHP és a Perl programozási nyelvek értelmezői (végrehajtó rendszerei).^[6]

A kód írását Visual Studio Code-ban végeztem.^[7]

A két különböző felület mintája két különböző template segítségével készült el.^{[8][9]}

Osztályok és metódusok

Model

Laravelben a model osztály felel az adatok megjelenítésére, illetve az adatbázissal való kapcsolattartásra.

Ezeket a `"php artisan make:model [modelnév] -m"` paranccsal lehet elkészíteni, és társul hozzájuk egy migrációs fájl is, amellyel az adatbázisban létre lehet hozni a modellhez tartozó táblát, valamint a hozzá tartozó mezőket. Ha elkészültek a fájlok akkor a `"php artisan migrate"` paranccsal tudjuk futtatni. A projektben 4 model található. User: a felhasználók adataihoz, Service a szolgáltatásokhoz, Doctor az orvosok adataihoz, illetve az Appointment, ami a foglalásokhoz tartozik.

```
database > migrations > 2023_10_04_145822_create_doctors_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('doctors', function (Blueprint $table) {
15               $table->id();
16               $table->string('name')->nullable();
17               $table->string('email')->nullable();
18               $table->string('speciality')->nullable();
19               $table->string('image')->nullable();
20
21               $table->timestamps();
22           });
23       }
24
25       /**
26       * Reverse the migrations.
27       */
28       public function down(): void
29       {
30           Schema::dropIfExists('doctors');
31       }
32  };
33
```

A Doctor modellhez tartozó migrációs fájl

View

A nézetekhez tartozó fájlok a \Hopitapp\resources\views mappában találhatóak.

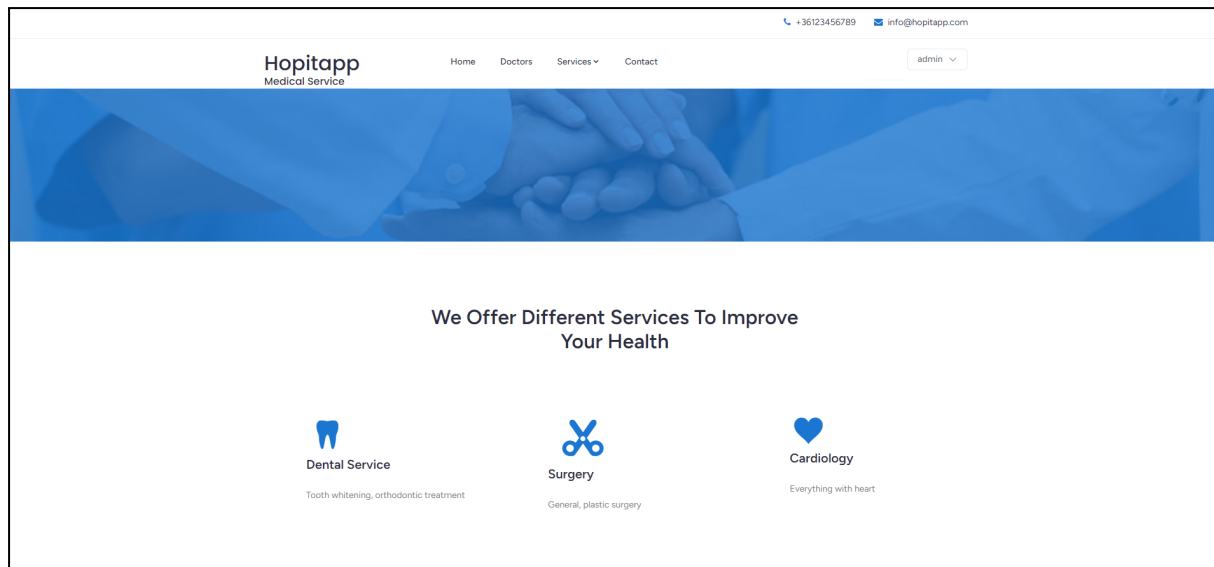
- **welcome.blade.php:** Ő mutatja meg az oldal fő felületét (Home)
- **doctors.blade.php:** A kórházban dolgozó alkalmazottakat mutatja
- **services.blade.php:** Az elérhető szolgáltatások megjelenítéséért felelős

```
resources > views > admin > services.blade.php
1 <div class="row">
2   <div class="col-12 grid-margin">
3     <div class="card">
4       <div class="card-body">
5         <h4 class="card-title">Services</h4>
6         <div class="table-responsive">
7           <table class="table">
8             <thead>
9               <tr>
10                <th>
11
12                </th>
13                <th> Image </th>
14                <th> Name </th>
15                <th> Description</th>
16              </tr>
17            </thead>
18            <tbody>
19              @foreach ($services as $service)
20                <tr>
21                  <td>
22                    <div class="form-check form-check-muted m-0">
23                      <label class="form-check-label">
24                        <input type="checkbox" class="form-check-input">
25                      </label>
26                    </div>
27                  </td>
28                  <td>
29                    
30                  </td>
31                  <td> {{ $service->name }} </td>
32                  <td> {{ $service->description }} </td>
33                </tr>
34              @endforeach
35            </tbody>
36          </table>
37        </div>
38      </div>
39    </div>
40  </div>
41</div>
```

A service.blade.php tartalma

- **appointment.blade.php:** Időpont foglalást segíti, viszont ez a felület csak regisztrált felhasználóknak érhető el
- **contact.blade.php:** A kórház elérhetőségeit mutatja

Az adminhoz tartozó oldalak az admin almappában találhatóak, ezek azok a felületek, melyen valamilyen admin joghoz kötött műveletet lehet elvégezni.



A service.blade.php megjelenése

Controller

A vezérlőhöz tartozó fájlok a `\Hopitapp\app\Http\Controllers` mappában találhatóak.

Két fő fájl van, az egyikben az adminhoz tartozó műveletek, míg a másikban a vendégekhez tartozó metódusok vannak. Minden metódus le van kezelve a jogosultság körének megfelelően.

Ezekben a fájlokban történik az adatok és a nézetek összekapcsolása, a keretrendszer beépített metódusával, valamint az adatbázis elérése.

AdminController:

A controller minden egyes metódusa ellenőrzi, hogy a bejelentkezett felhasználónak tényleg van-e jogköre az adott oldal hozzáféréséhez, ha nincs erről tájékoztatja, vagy bejelentkezésre kötelezi.

Ilyen metódusok például:

adddoctor(): Ha szeretnénk további, az intézményben dolgozó orvosokat adni a rendszerhez, ez a felület elérésére szolgál

uploaddoctor(Request \$request): Az előző felülethez tartozó POST metódus, mely az adatokat a rendszerbe menti

deletedoc(\$id): Lehetővé teszi az adott orvos adatainak törlését az adatbázisból és így a felületről

```
1 reference | 0 overrides
140 public function uploaddoctor(Request $request)
141 {
142
143
144
145
146     if (Auth::id()) {
147
148         if (Auth::user()->usertype == '1') {
149
150             $doctor = new doctor;
151
152
153             $image = $request->file;
154
155             $imagename = time() . '.' . $image->getClientOriginalExtension();
156             $request->file->move('doctorimage', $imagename);
157
158             $doctor->image = $imagename;
159
160             $doctor->name = $request->name;
161
162             $doctor->email = $request->email;
163
164             $doctor->speciality = $request->speciality;
165
166             $doctor->save();
167
168             return redirect()->back();
169         } else {
170
171             $doctors = doctor::all();
172             $services = Service::all();
173
174             return view('nopermission');
175
176         }
177     }
178
179     } else {
180         return view('auth.login');
181     }
182 }
183
184
```

AdminController::uploaddoctor() metódusa

HomeController

A controllerben található metódusok lehetővé teszik egyes al-oldalak elérését. Alapvetően mindenki számára elérhetőek, kivétel ez alól a foglalást biztosító felület. Illetve a `redirect()` metódust, ahol ha admin felhasználók vagyunk az admin felületet érhetjük el, ha bejelentkezett felhasználók akkor a sima címdalt, illetve ha még nem regisztráltunk akkor a bejelentkezési felületet.

```
16 public function redirect(){
17
18     $doctors= doctor::all();
19     $services= Service::all();
20
21     if (Auth::id()){
22
23         if (Auth::user()->usertype=='0'){
24             return view('welcome',compact('doctors','services'));
25         }
26         else{
27             $appointments=appointment::all();
28             return view('admin.show_appointments',compact('appointments'));
29         }
30     }
31     else{
32         return redirect()->back();
33     }
34 }
35
36
```

A controller és a view összekötését a `\Hopitapp\routes\web.php` fájl biztosítja.

Fejlesztési lehetőségek

Az alkalmazás jelenleg csak időpontok foglalására alkalmas, de sok lehetőség van benne.

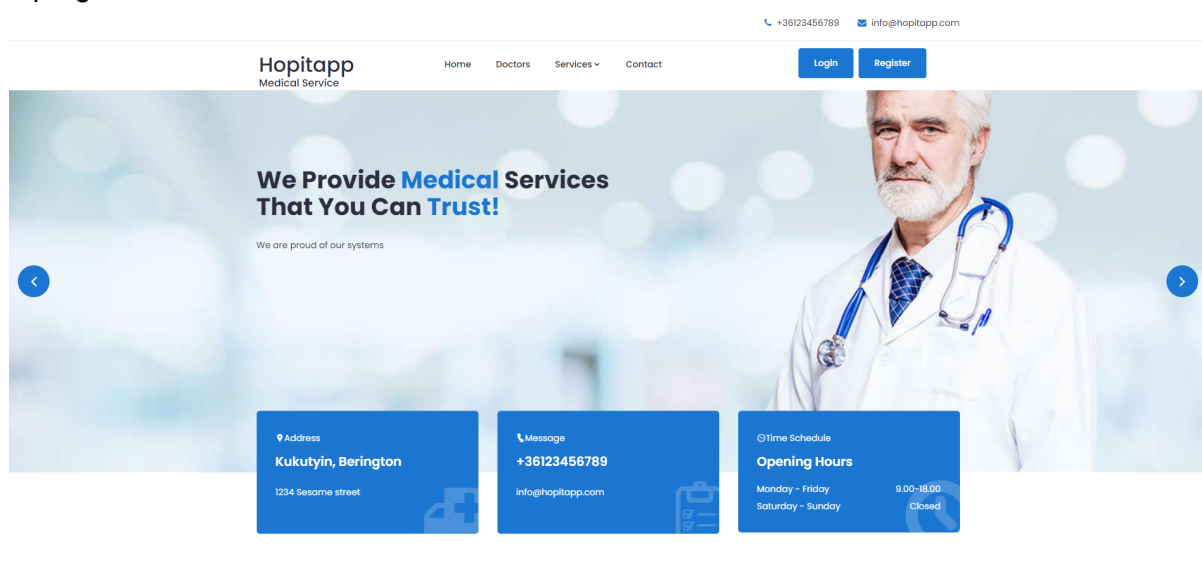
Lehetne fejleszteni a foglalás menetét, például ha egy adott orvoshoz nincs szabad időpont, akkor ajánlana egy másik, a területen dolgozó specialistát. Ha valaki lemond egy foglalást az oldalon, azonnal nyomon lehetne követni. Illetve ha van más, aki az adott orvoshoz foglalt időpontot kaphatna értesítést erről, abban az esetben, ha egy későbbi alkalomra szól a foglalása.

De nem csak időpontfoglalásra kell gondolni, akár még egy közvetlen üzenő felületet is ki lehetne alakítani az orvosok és páciensek között.

Felhasználói dokumentáció

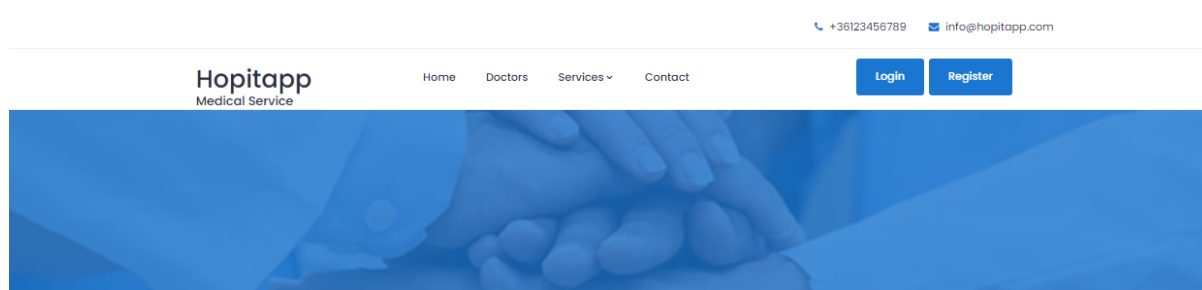
Vendégeknek

A program elsői elindulásakor a következő felület látható:



A lap lentebb görgetésével lehet látni a szolgáltatásokat, az itt dolgozó orvosokat, illetve információkat.

A menüpontok használatával szintén ezeket az információkat lehet elérni, csak egy másik oldal betöltésével.



Our Doctors



They are always calm because have a lot of patients



Cardiologist

Dr Laura Mayer

lm@hopitapp.com



Gynecologist

Dr John Harrington

jh@hopitapp.com



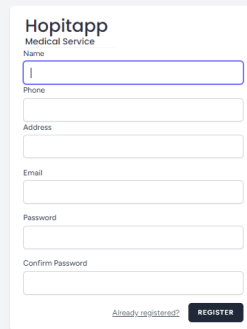
Dentist

Dr Juan Carlos

jc@hopitapp.com



A "Service" egy legördülő menü, amire ha rákattintunk a szolgáltatásokat lehet megtekinteni, míg ha a legördülő menüben lévő "Book Appointment"-et választjuk, a foglalásunkat lehet intézni. Ez csak bejelentkezett felhasználónak érhető el. Ha nem bejelentkezett felhasználóként jelenítjük meg, akkor a regisztrációs felületre visz át. Ez az oldal a "Register" menüre kattintva is elérhető.



The registration form for Hopitapp Medical Service is displayed. It includes input fields for Name, Phone, Address, Email, Password, and Confirm Password. A "REGISTER" button is located at the bottom right, and a link "Already registered?" is positioned just above it.

Hopitapp
Medical Service

Name

Phone

Address

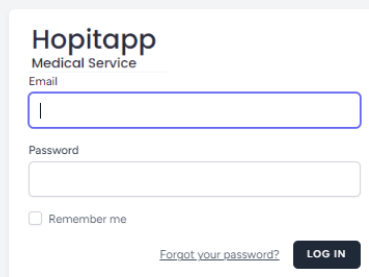
Email

Password

Confirm Password

[Already registered?](#) **REGISTER**

Az adataink megadása után tudunk is regisztrálni, vagy ha már van létező felhasználói fiókunk, akkor az ahhoz való bejelentkező felületet az "Already registered" felíratra kattintva érhetjük el.



The login form for Hopitapp Medical Service is displayed. It includes input fields for Email and Password. A "LOG IN" button is located at the bottom right, and a link "Forgot your password?" is positioned just above it. There is also a "Remember me" checkbox.

Hopitapp
Medical Service

Email

Password

☐ Remember me

[Forgot your password?](#) **LOG IN**

Bejelentkezés után (ha nincs admin jogosultságunk) újra a nyitó felület jelenik meg, és most már elérjük az időpontfoglalás felületét.

+36123456789

info@hopitapp.com

Hopitapp
Medical Service

HomeDoctorsServices▼Contact

nonadmin ▼

Get Your Appointment

Nonadmin

Nonadmin@Nonadmin.Com

Dr Laura Mayer (Cardiologist) ▼

Book An Appointment

Éééé. Hh. Nn.

Address

Kukutyin, Berington

1234 Sesame street

Message

+36123456789

info@hopitapp.com

Time Schedule

Opening Hours

Monday - Friday
Saturday - Sunday

9.00-18.00
Closed

Itt az orvos és a dátum kiválasztásával tudunk időpontot foglalni. Ha nem adunk meg időpontot, akkor csak jelezzük az admin felhasználó felé igényünket és majd ő ad nekünk egy időpontot az ő felületén.

A “Contact” menüpont alatt a kórház elérhetőségei találhatóak.

+36123456789

info@hopitapp.com

Hopitapp
Medical Service

HomeDoctorsServices▼Contact

nonadmin ▼

Contact

Address

Kukutyin, Berington

1234 Sesame street

Message

+36123456789

info@hopitapp.com

Time Schedule

Opening Hours

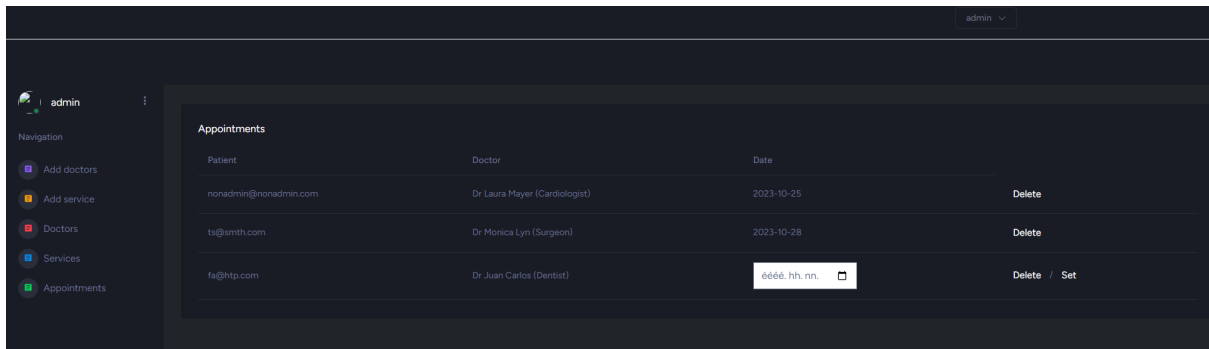
Monday - Friday
Saturday - Sunday

9.00-18.00
Closed

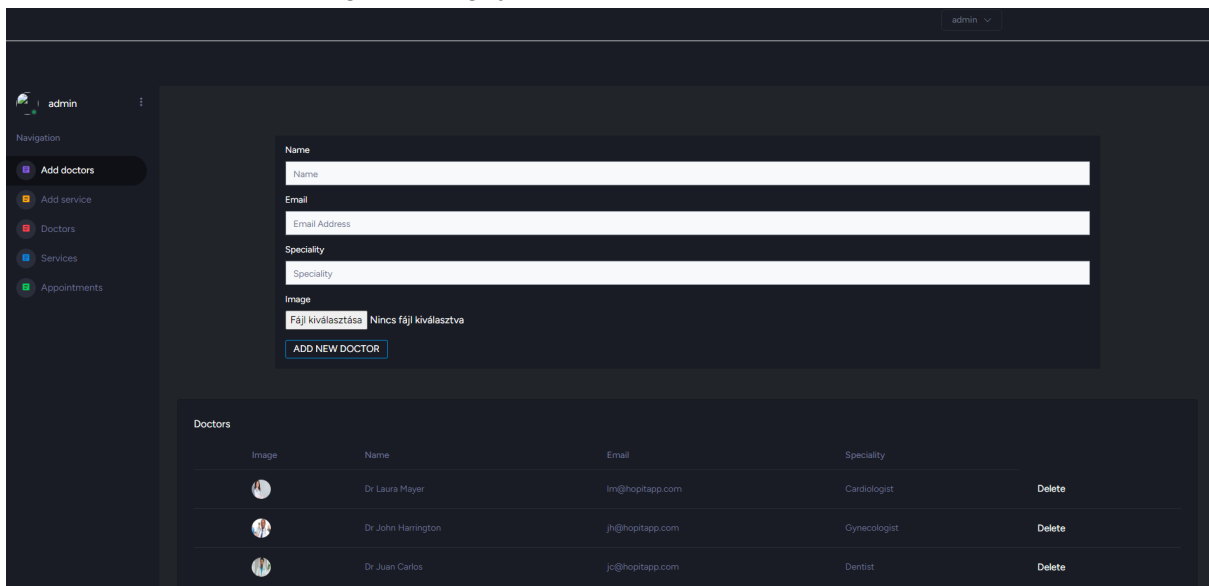
Természetesen a jobb felső sarokban lévő felhasználóneveinkre kattintva ki tudunk jelentkezni a rendszerből.

Adminoknak

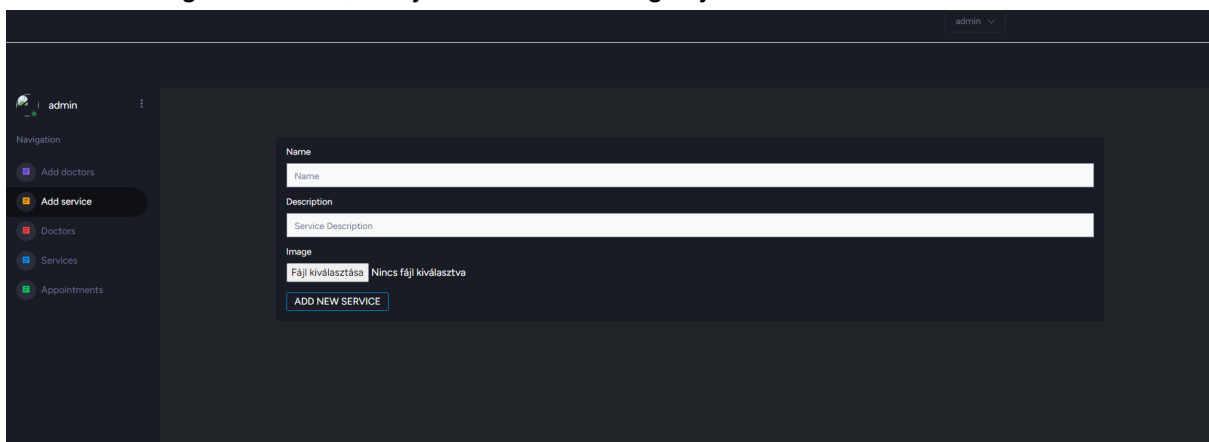
Bejelentkezés után a következő felület fogadja az admin jogosultsággal rendelkező felhasználókat:



Elsőre a foglalt időpontok oldalát látjuk, ahol törölhetjük egy adott illetőnek az időpontját, vagy ha nem foglalt le ténylegesen időpontot, be tudjuk állítani azt az időpont kiválasztásával és a "Set" gomb megnyomásával.

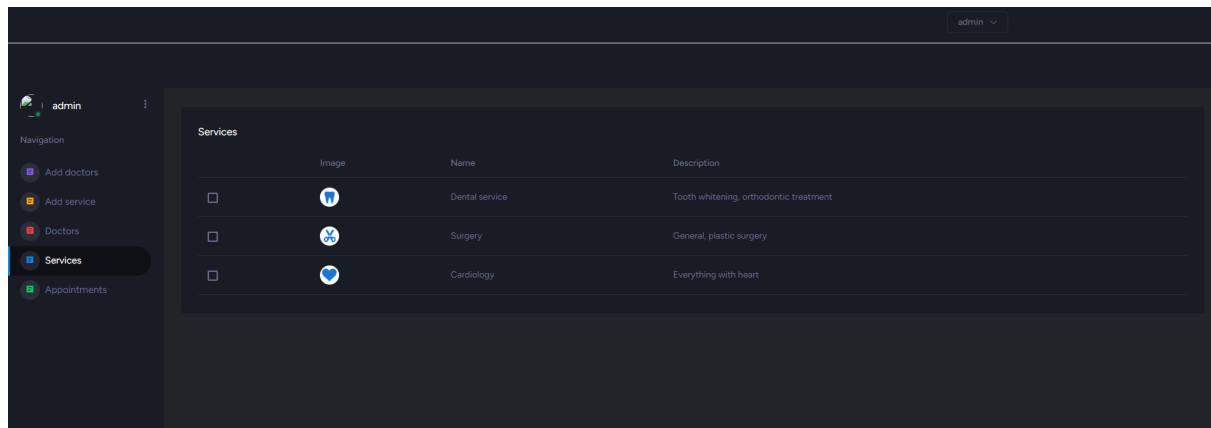


Ha új orvost szeretnénk felvenni a rendszerbe, az "Add doctor" menüpontban tehetjük meg. Az adatok megadása után, az új doktor adatai meg is jelennek a lent lévő táblázatban.



Ha új szolgáltatást szeretnénk hozzáadni, akkor az "Add service" menüpont a számunkra megfelelő választás.

Ha meg is szeretnénk nézni eddigi szolgáltatásainkat akkor azt a “Services” menüben tehetjük meg.

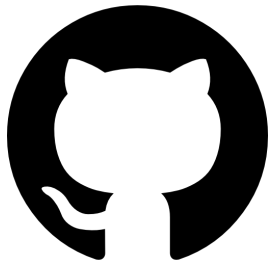


A Doctor menüpont alatt azt a táblázatot tudjuk megtekinteni, amit az “Add doctor” menüpontban is láthattunk.

Természetesen admin módban is megtekinthetjük a website vendég verzióját. Ehhez beírunk egy a vendég módhoz tartozó linket és regisztrált felhasználóként böngészgethetjük. Abban az esetben, ha vissza szeretnénk lépni admin módba, a jobb felső sarokban lévő legördülő menüből kiválasztjuk a “Home” oldalt és újra a foglalt időpontok felületét láthatjuk. Admin módban is a jobb felső sarokban található a kijelentkezéshez tartozó menüpont.

GitHub elérhetőség

Az alkalmazás elérhető [GitHubon](#).



A kiexportált adatbázisban (hopitapp.sql) több felhasználható található.

Az admin oldal teszteléséhez szükséges felhasználó adatai:

Email: admin@admin.com

Jelszó: admin123456789

A vendég oldal egyik felhasználójának adatai:

Email: nonadmin@nonadmin.com

Jelszó: nonadmin123456789

Hivatkozások

- [1] <https://media.geeksforgeeks.org/wp-content/uploads/20230927120218/mvc.png>
- [2] <https://laravel.com/>
- [3] <https://getbootstrap.com/>
- [4] <https://www.mysql.com/>
- [5] <https://www.apachefriends.org/hu/index.html>
- [6] <https://hu.wikipedia.org/wiki/XAMPP>
- [7] <https://code.visualstudio.com/>
- [8] <https://www.free-css.com/free-css-templates/page296/mediplus-lite>
- [9] <https://github.com/BootstrapDash/corona-free-dark-bootstrap-admin-template>
- [10] <https://github.com/tsoltesz/stgtv2023>