

Vas VÁRMegyei Szakképzési Centrum  
Horváth Boldizsár   
Közgazdasági és Informatikai  
TECHNIKUM

A 5 0613 12 03 számú Szoftverfejlesztő és –tesztelő vizsgaremek

**A szoftveralkalmazás dokumentációja**

Készítették:

Földessi Zoltán valentin

Hajba Martin

Nagy botond

**Szombathely**

**2025**

# Bevezetés

## Program célja (Frontend):

Célunk egy versenyképes weboldal létrehozása volt a maga műfajában. Körülnézve, kutakodva nem találtunk olyan autószereléssel kapcsolatos weboldalt, ami mindannyiunknak tetszett volna. Mindhármunkat érdekelnek az autók, azoknak működése, szerelése, karbantartása.

Fontos szempont volt számunkra, hogy az oldal letisztult, könnyen átlátható, felhasználóbarát legyen, modern megjelenésével kiemelkedjen a konkurens oldalak közül, illetve, hogy a látogatók gyorsan megtalálják a számukra szükséges információkat, legyen szó időpontfoglalásról, szolgáltatásaink bemutatásáról vagy akár kapcsolatfelvételről. Törekedtünk arra, hogy a weboldal ne csak informatív, hanem bizalmat ébresztő és barátságos is legyen.

Program célja (Asztali):

Célja a szervízhez tartozó adatok nyilvántartása, ezzel segítve a dolgozók munkáját és az ügyfelek kéréseit.

## Célközönség:

(Frontend)

Azon ügyfelek bevonzása, akik autóvásárlás, műszaki vizsgáztatás, állapotfelmérés, autó javítás előtt állnak és szeretnének teljeskörű, megbízható számlaképes autószerelő vállalkozást megbízni.

(Asztali)

Járműszervizek, akik szeretnék munkájukat gördülékenyebbé, könnyebben átláthatóvá tenni.

## Betűszavak és meghatározások:

GYIK: Gyakran ismételt kérdések

SMS: Short message service

# Program általános leírása

## Követelmények, tervezés és a rendszer jellemzői – megvalósítási terv

**Követelmények**

 **React**:

* react
* react-dom

 **Dátumkezeléshez**:

* date-fns

 **Ikonokhoz**:

* react-icons

 **Material UI komponensekhez (gomb, tooltip.)**:

* @mui/material
* @emotion/react
* @emotion/styled

 **Email küldéshez**:

* emailjs/browser

 S**liderhez**:

* react-slick
* slick-carousel

 **GYIK (FAQ) komponenshez**:

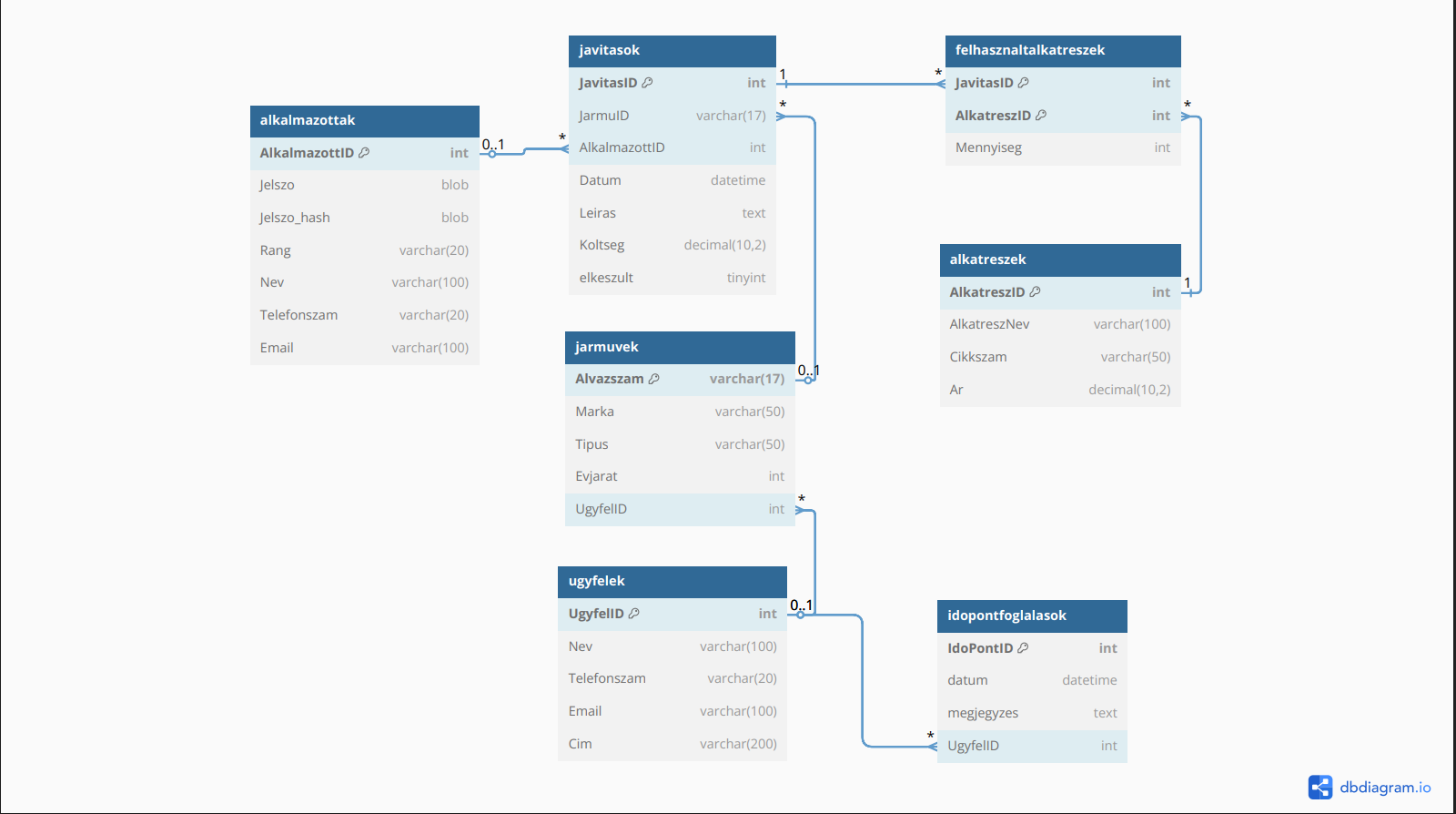
* react-faq-component

npm install react react-dom date-fns react-icons @mui/material @emotion/react @emotion/styled emailjs/browser react-slick slick-carousel react-faq-component

## Tervezés – feladatok felosztása

|  |  |
| --- | --- |
| **Feladat:** | **Készítette:** |
| Adatbázis | Hajba Martin, Nagy Botond |
| Entity relationship diagram | Hajba Martin |
| ASP.NET Web Application / Unit Test | Hajba Martin |
| Asztali alkalmazás | Nagy Botond |
| Weboldal | Földessi Zoltán |
| Dokumentáció | Hajba Martin, Földessi Zoltán, Nagy Botond |

## Adatbázis ER modell:

****

. ábra: ER-Modell

Tárolja az ügyfelek adatait, járműveiket, javítási munkákat, alkalmazottakat, alkatrészeket és időpontfoglalásokat. A rendszer célja, hogy nyomon kövesse a szervizelési folyamatokat, alkatrészhasználatot és ügyfélkapcsolatokat.

## Táblák és Mezők

1. Alkalmazottak

Mezők:

* AlkalmazottID (Primary Key): Egyedi azonosító.
* Jelszo: Titkosított vagy bináris formátumban tárolt jelszó (Salt)
* Jelszo\_hash: Hash-elt jelszó
* Rang: Alkalmazott rangja (pl. "szerelő", "művezető").
* Nev, Telefonszam, Email: Személyes adatok.

2.Alkatrészek

Mezők:

* AlkatreszID (Primary Key): Egyedi azonosító.
* AlkatreszNev: Alkartész neve (pl. "Légfilter").
* Cikkszam: Egyedi cikkszám.
* Ar: Alkartész ára (HUF).

3.Felhasznált alkatrészek

Mezők:

* JavitasID (Foreign Key → javitasok.JavitasID): A javítás azonosítója.
* AlkatreszID (Foreign Key → alkatreszek.AlkatreszID): Felhasznált alkatrész azonosítója.
* Mennyiseg: Felhasznált mennyiség.
* Kapcsolat: Többértelmű kapcsolótábla a javitasok és alkatreszek között.

4. Időpontfoglalások

Mezők:

* IdoPontID (Primary Key): Egyedi azonosító.
* datum: Foglalt dátum és idő.
* megjegyzes: Egyéb információk (pl. "Olajcsere").
* UgyfelID (Foreign Key → ugyfelek.UgyfelID): Ügyfél azonosítója.

5. Járművek

Mezők:

* Alvazszam (Primary Key): Egyedi alvázszám (17 karakter).
* Marka, Tipus, Evjarat: Jármű adatai.
* UgyfelID (Foreign Key → ugyfelek.UgyfelID): Tulajdonos azonosítója.

6. Javítások

Mezők:

* JavitasID (Primary Key): Egyedi azonosító.
* JarmuID (Foreign Key → jarmuvek.Alvazszam): Javított jármű.
* AlkalmazottID (Foreign Key → alkalmazottak.AlkalmazottID): Felelős alkalmazott.
* Datum: Javítás időpontja.
* Leiras, Koltseg: Javítás részletei és költsége.
* elkeszult: Státusz (0: folyamatban, 1: kész).

7. Ügyfelek

Mezők:

* UgyfelID (Primary Key): Egyedi azonosító.
* Nev, Telefonszam, Email, Cim: Ügyfél adatai.

Kapcsolatok (Foreign Keys)

* felhasznaltalkatreszek → javitasok és alkatreszek.
* idopontfoglalasok → ugyfelek.
* jarmuvek → ugyfelek.
* javitasok → jarmuvek és alkalmazottak.

## JarmuSzerviz Web API

Ez a dokumentáció a JarmuSzerviz ASP.NET Web API működését magyarázza el. A rendszer egy üzlet adatkezelését támogatja, beleértve az ügyfeleket, járműveket, alkalmazottakat, javításokat, időpontfoglalásokat és alkatrészeket. A dokumentáció célja, hogy egyértelműen leírja a kontrollerek működését, és segítsen a fejlesztőknek gyorsan integrálni az API-t.

Projekt Struktúra és Swagger UI

A Web API, Swagger segítségével dokumentálja önmagát. A Swagger UI lehetővé teszi:

* Végpontok interaktív tesztelését.
* Kérés-válasz formátumok vizualizálását.

**Általános Működési Elv**

Az összes Controller ugyanazon logikát követi:

* CRUD műveletek (Create, Read, Update, Delete) végrehajtása.
* RESTful végpontok használata (GET, POST, PUT, DELETE).
* Adatbázis kommunikáció Entity Framework keretrendszerrel.
* Swagger UI integráció a manuális teszteléshez.
* Minden végpont standard HTTP státuszkódokkal válaszol (pl. 200 OK, 404 Not Found).

**Közös Szerkezet Minden Controllerben**

|  |  |  |  |
| --- | --- | --- | --- |
| Metódus | Útvonal | Művelet | Példa |
| **GET** | /api/{entitás} | Összes elem listázása. | /api/alkatreszek |
| **GET** | /api/{entitás}/{id} | Egy elem lekérése ID alapján. | /api/jarmuvek/ABC123 |
| **POST** | /api/{entitás} | |  | | --- | |  |   Új elem hozzáadása. | /api/ugyfelek |
| **PUT** | /api/{entitás}/{id} | Elem frissítése. | /api/javitasok/3 |
| **DELETE** | /api/{entitás}/{id} | Elem törlése. | /api/idopontok/5 |

**Jelszókezelés**

A rendszerben a jelszavak titkosítva és biztonságosan tárolódnak. A folyamat részletei:

Jelszó Generálás és Hash-elés

Amikor egy alkalmazott létrejön vagy frissül, a következő lépések történnek:

CreatePasswordHash Metódus

A Security/ApiValidator.cs osztály felelős a hash és salt generálásáért.

1.2 Működés Lépésről Lépésre

Salt generálása:

* A rendszer egy véletlenszerű 512 bites salt-ot hoz létre a HMACSHA512 algoritmussal.
* A salt az alkalmazott Jelszo mezőjébe kerül.

Hash készítése:

* A felhasználó által megadott jelszót a rendszer összekombinálja a salttal.
* A kombinációt a HMACSHA512 algoritmus hashel, és az eredmény az alkalmazott Jelszo\_hash mezőjébe kerül.

A képen szöveg, szoftver, Multimédiás szoftver, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

## Jelszó Ellenőrzés

## Amikor egy alkalmazott bejelentkezik, a rendszer a következőt hajtja végre:

## VerifyPasswordHash Metódus

## Működés

## A rendszer kiveszi a salt-ot az alkalmazott Jelszo mezőjéből.

## A felhasználó által megadott jelszót újra hash-eli a salt segítségével.

## Összehasonlítja az új hash-t az adatbázisban tárolt Jelszo\_hash értékkel.

## Ha egyezik: sikeres bejelentkezés. Ha nem: hiba.

A képen szöveg, képernyőkép látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

## Felület vázlatos megtervezése:

Tervezéskor fő célunk a letisztultság, átláthatóság, felhasználóbarát környezet megtervezése volt. Minden komponenst előre vázlatszerűen megterveztünk, amiken megvalósítás közben változtattunk.

# Segédoldalak, információs felületek, linkek:

Programunk megvalósításához több helyről is vettünk segítségeket, képeket ezzel gyorsítva, megkönnyítve a folyamatokat.

Ezek között volt:

* Az órán megtanult tudás alkalmazása.
* Kódok újrahasznosítása.
* Ingyenesen felhasználható képek, ikonok, betűstílusok.

linkek:

* <https://www.youtube.com/watch?v=RWz23UKXdAk&t=912s> Naptár
* <https://www.youtube.com/watch?v=OV8MVmtgmoY> Flip card
* <https://www.svgbackgrounds.com/> Hátterek
* <https://www.npmjs.com/package/react-slick> Slider
* <https://coolors.co/db324d-e8e1ef-7a9e7e-22181c-d16014> Szín paletták
* <https://fonts.google.com/?authuser=0> Betű stílus
* <https://react-icons.github.io/react-icons/> Ikonok
* <https://www.npmjs.com/package/react-faq-component> GYIK

# Komponenseinek technikai leírása

## Használatának rövid bemutatása:

Weboldalunk reactban készült, megvalósitása egy OnePage-ben formájában jött létre, "react": "^19.1.0" -es verziót használ.

### 1.Főoldal

Az oldal tetején megtalálható a szervizünk logója, egy navigációs menü mely elvezet a további oldalakra. Menüsorunk reszponzív, alatta található pedig egy slider rövid kis szövegekkel.

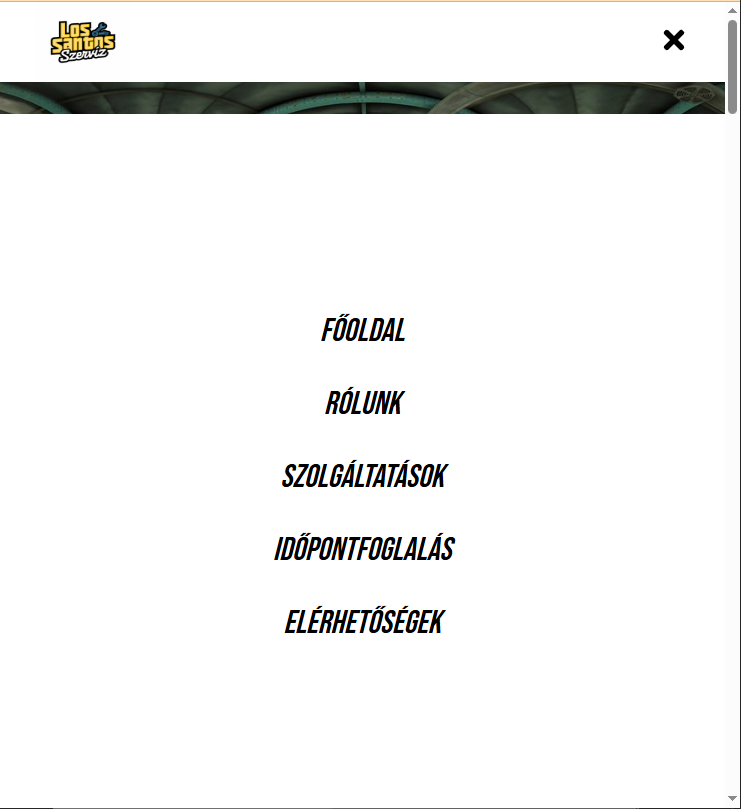
Szükséges hozzá:

* import "slick-carousel/slick/slick.css";
* import "slick-carousel/slick/slick-theme.css";
* import { FaBars, FaTimes } from "react-icons/fa";
* import Slider from "react-slick";

-npm install react-slick slick-carousel react-icons



. ábra: Főoldal képernyőkép



. ábra: Főoldal reszponzív képernyőkép

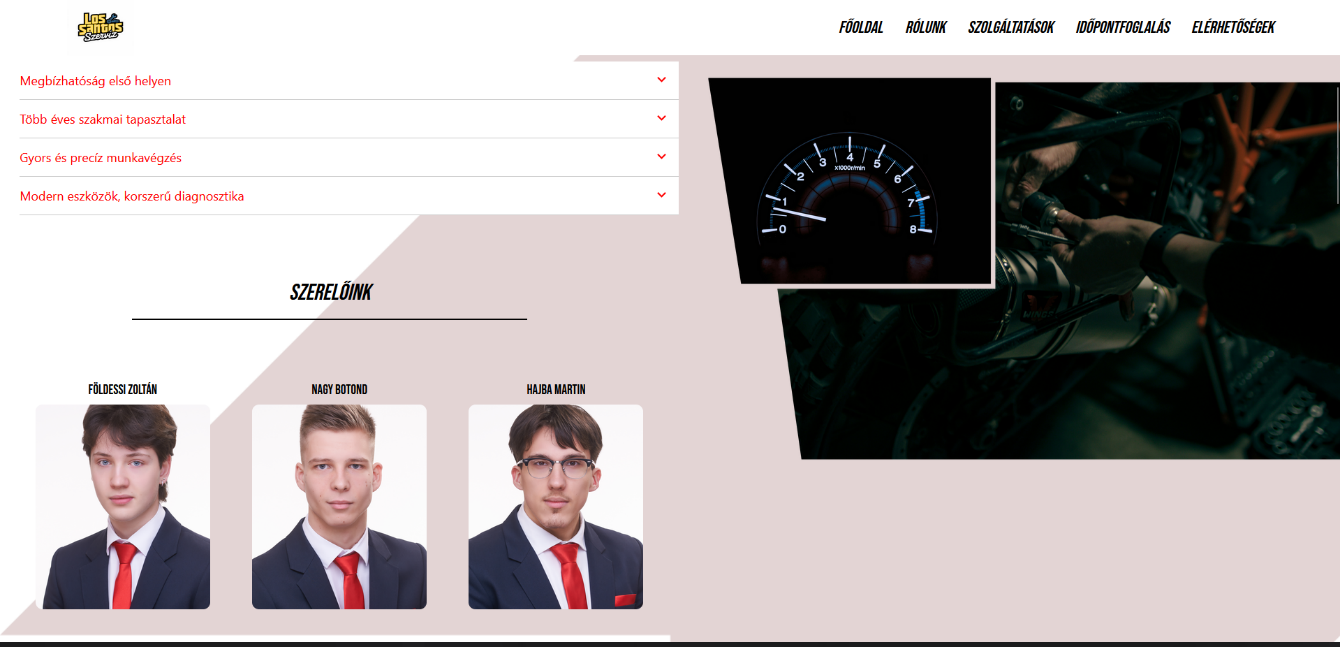
### 2.Rólunk oldal

A rólunk oldalon megtalálható egy GYIK beépített komponens, amit mi rövid bemutatkozásra használtunk ezzel sarkallva, ösztönözve az embereket arra, hogy minket válasszanak. Egy általunk szerkesztett kép, illetve szerelőinknek rövid fényképes bemutatkozása kártyák formájában valósult meg. Reszponzivitás érdekében 1000px szélesség után az oldalon található kép eltűnik, ezzel helyet adva a GYIK komponensnek.

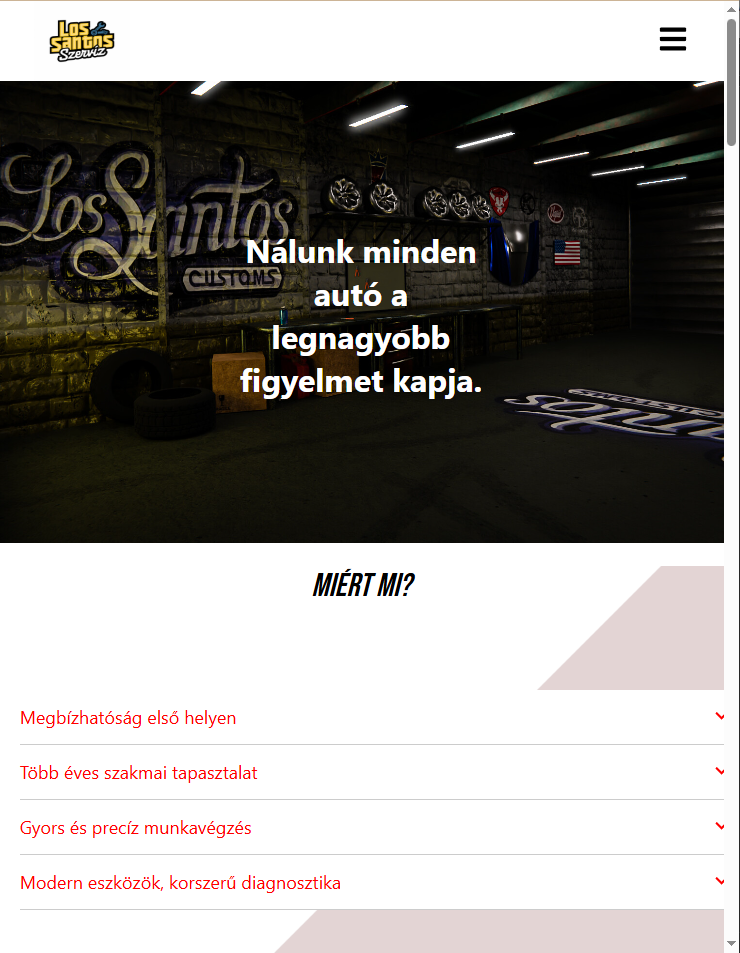
Szükséges hozzá:

* import Faq from "react-faq-component";
* npm install react-faq-component

Háttér forrása: https://www.svgbackgrounds.com/



. ábra: Rólunk oldal képernyőkép



. ábra: Rólunk reszponzív képernyőkép

. Rólunk reszponzív képernyőkép

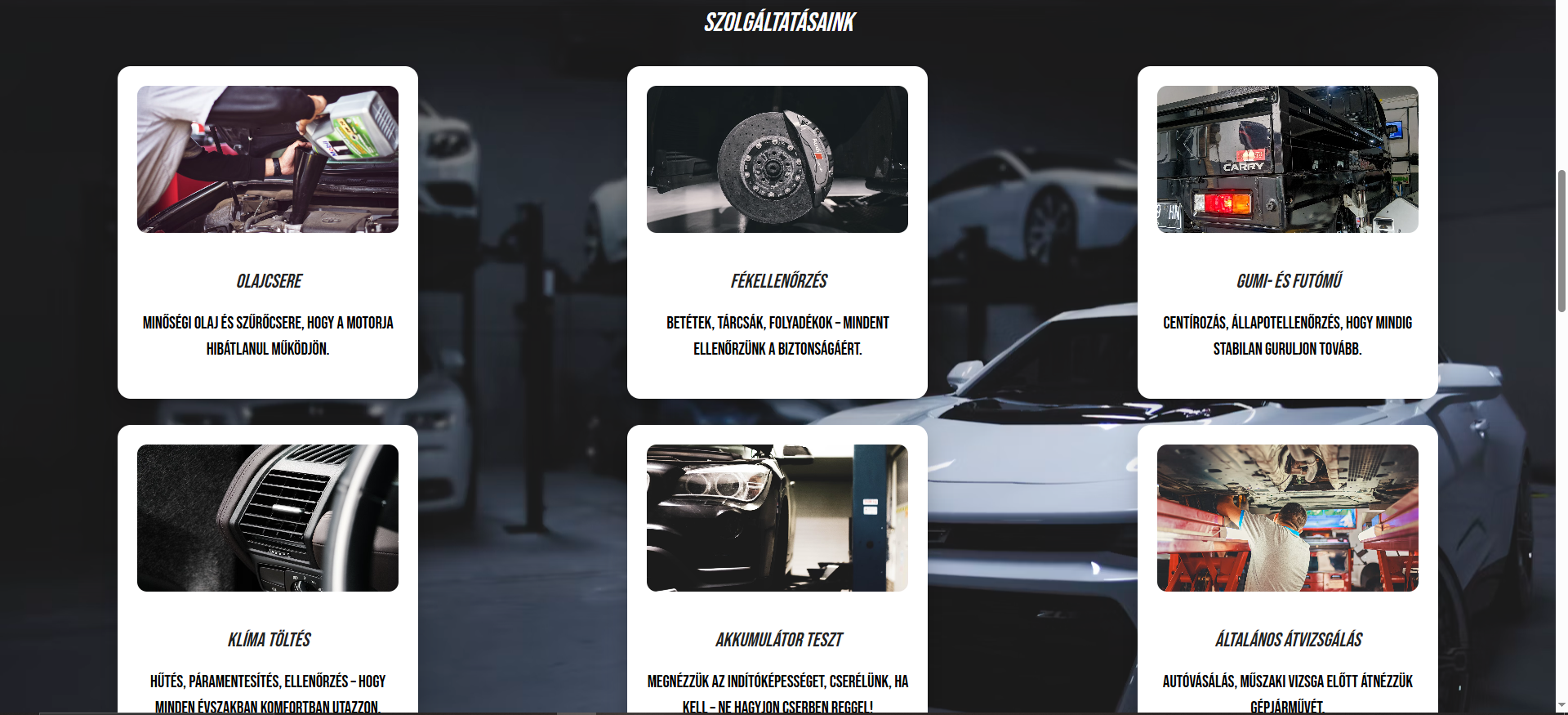


. ábra: Rólunk reszponzív képernyőkép

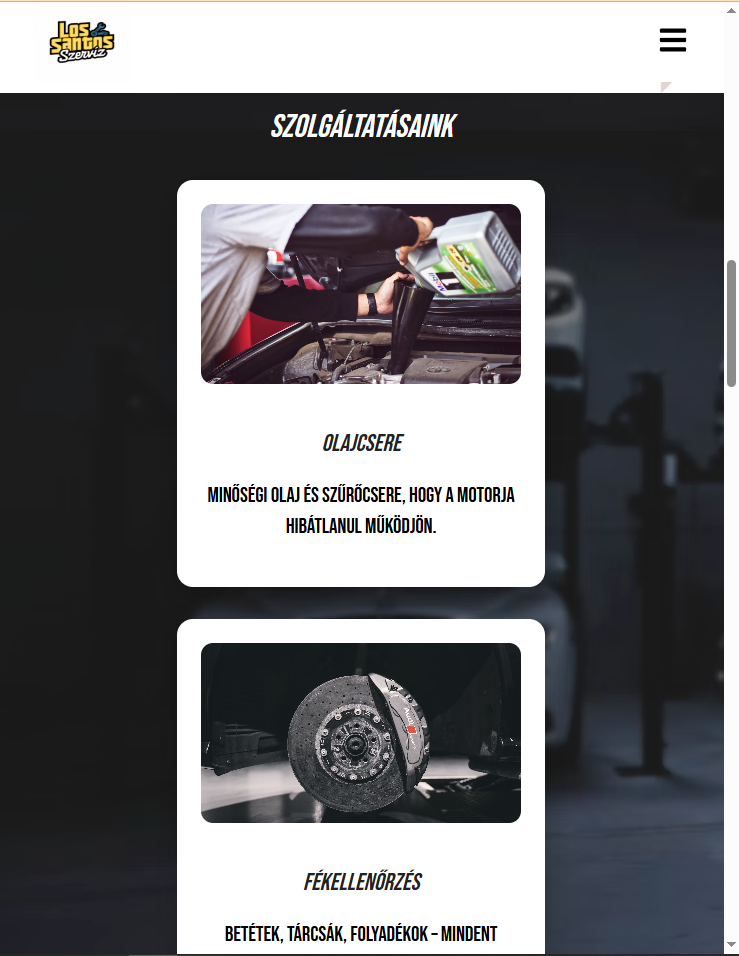
### 3.Szolgáltatások oldal

Szolgáltatásainknál röviden leírjuk az alap szolgáltatásokat kártyákba téve, amik tartalmaznak kis animációt. Az oldal háttere, illetve a kártyákba illesztett képek Stock fotók, ezekkel próbálva felkelteni a figyelmet.

Képek forrása: <https://unsplash.com/?authuser=0>



. ábra: Szolgáltatások oldal képernyőkép



. ábra: Szolgáltatások reszponzív képernyőkép

### 4.Időpontfoglalás oldal

Időpont foglalása történhet személyesen, telefonon, vagy akár weboldalon keresztül is.

A naptár alatt található egy kérdőjel ikon, mely magyarázattal szolgál az időpontfoglalás tekintetében, mindenkinek egyértelművé téve a foglalás menetét. A dátum megadása után hasonló módszerrel lehet kiválasztani az órát, mely után az űrlapon kérjük az ügyfél teljes nevét, lakcímét, telefonszámát. A telefonszám, e-mail cím ellenőrizve van, hogy biztosan jó formátumot írt-e be az ügyfél. Ezután a kívánt szolgáltatást lehet kiválasztani, legördülő lista formájában.

Mivel nem minden szolgáltatásra elegendő a 2 óra munkaóra, található egy egyéb fül a szerelés típusánál, melyre kattintva az ügyfél tájékoztatást kap arra vonatkozóan, hogy szükséges minket felkeresni az oldalon feltüntetett elérhetőségeken.

Amennyiben sikeres volt az időpontfoglalás, küldünk egy visszaigazoló e-mailt feltüntetve a kívánt szolgáltatást és az időpontot.

A date-fns csomag segítségével tudjuk kezelni a dátumokat (formázás, számolás), date-fns/locale pedig segít a hónapok magyar lokalizációjában.

Az emailjs/browser segít a frontendről való e-mail küldésben API segítségével.

Az oldalhoz tartozik még egy fetch.js fájl, melyben kettő fetch metódus (két POST) mellyel a felhasználó által megadott adatokat felvisszük az adatbázisunkba.

Szükséges hozzá:

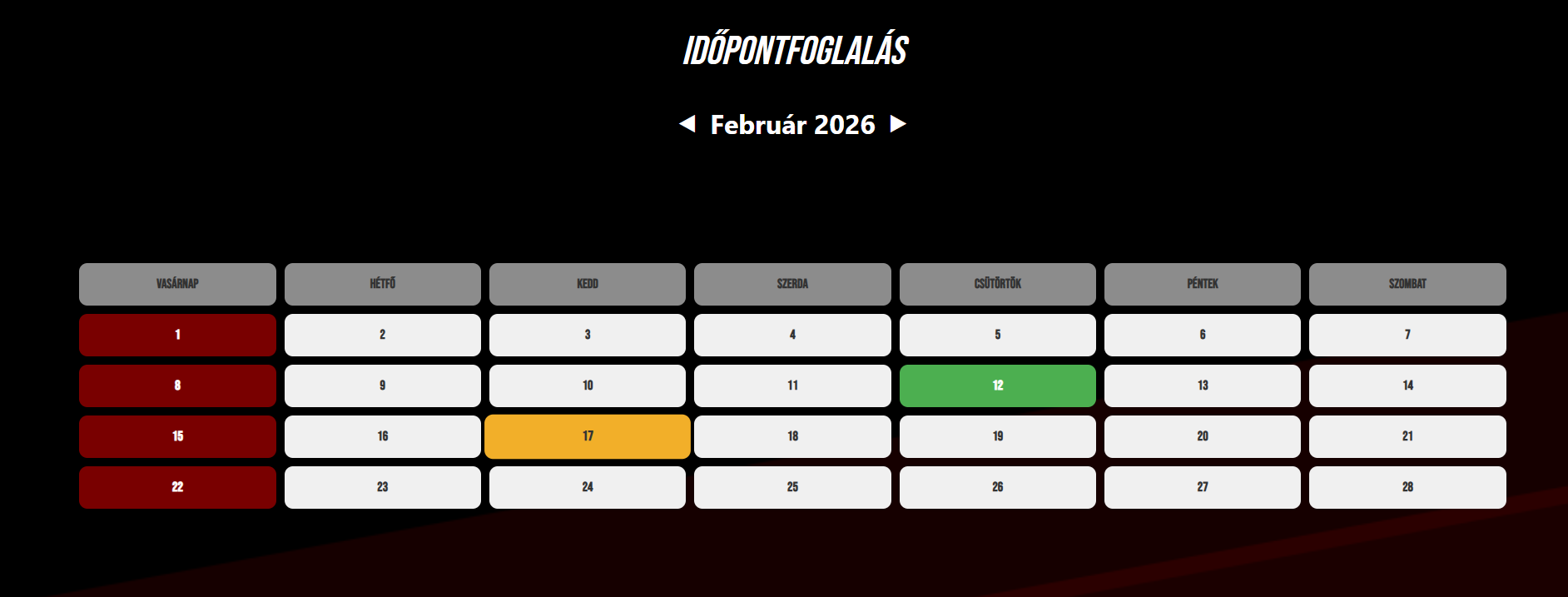
* import { hu } from 'date-fns/locale';
* import { RxQuestionMarkCircled } from "react-icons/rx";
* import Tooltip from '@mui/material/Tooltip';
* import Button from '@mui/material/Button';
* import { MdEmail } from "react-icons/md";
* import { FaUser } from "react-icons/fa";
* import { FaScrewdriverWrench } from "react-icons/fa6";
* import { createUgyfel, createIdopont } from './fetch';
* import { FaPhoneSquareAlt } from "react-icons/fa";
* import { FaHouseUser } from "react-icons/fa";
* import emailjs from '@emailjs/browser';
* import {
* eachDayOfInterval,
* endOfMonth,
* format,
* startOfMonth,
* getDay,
* isSameDay,
* addMonths,
* subMonths
* } from 'date-fns';
* npm install react-icons date-fns @mui/material @emotion/react @emotion/styled @emailjs/browser

Háttér forrása: <https://www.svgbackgrounds.com/>

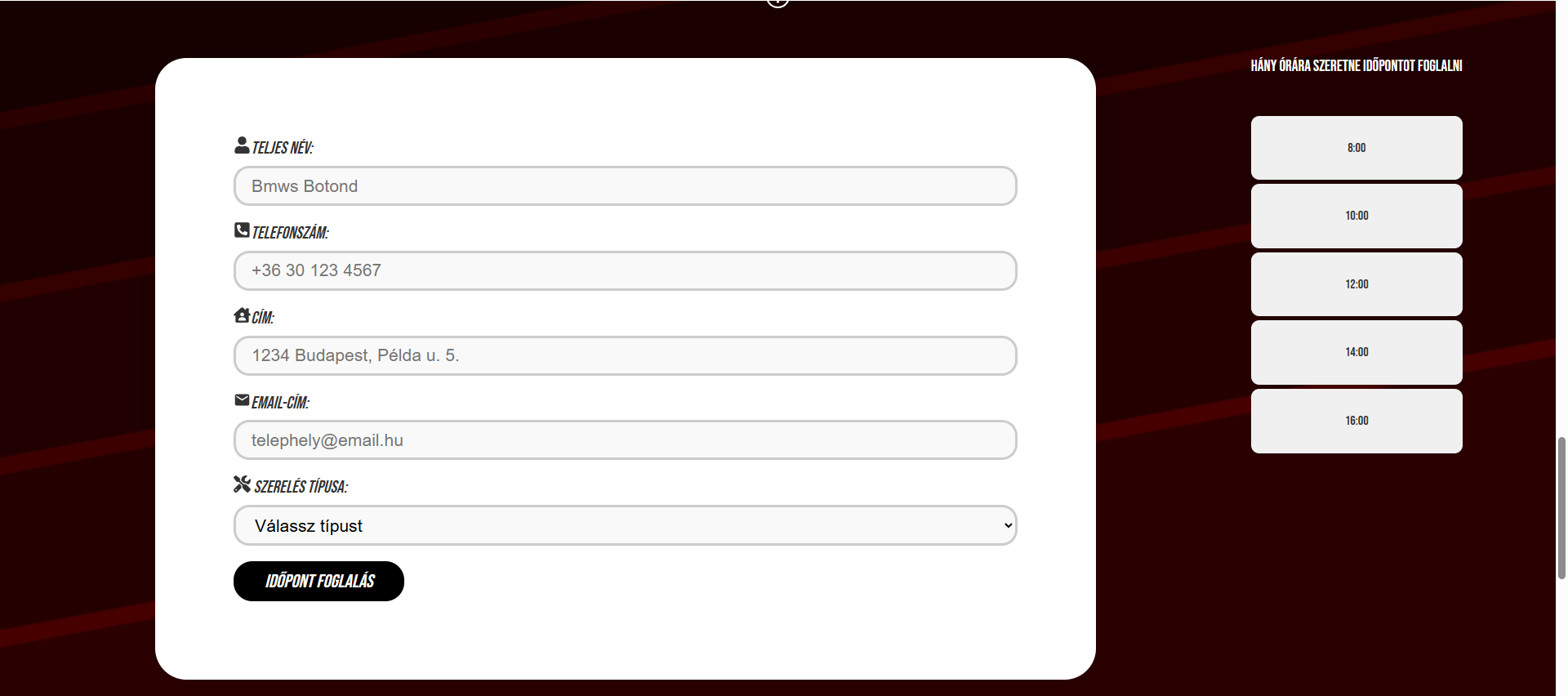
Naptár forrása: <https://www.youtube.com/watch?v=RWz23UKXdAk&t=912s>



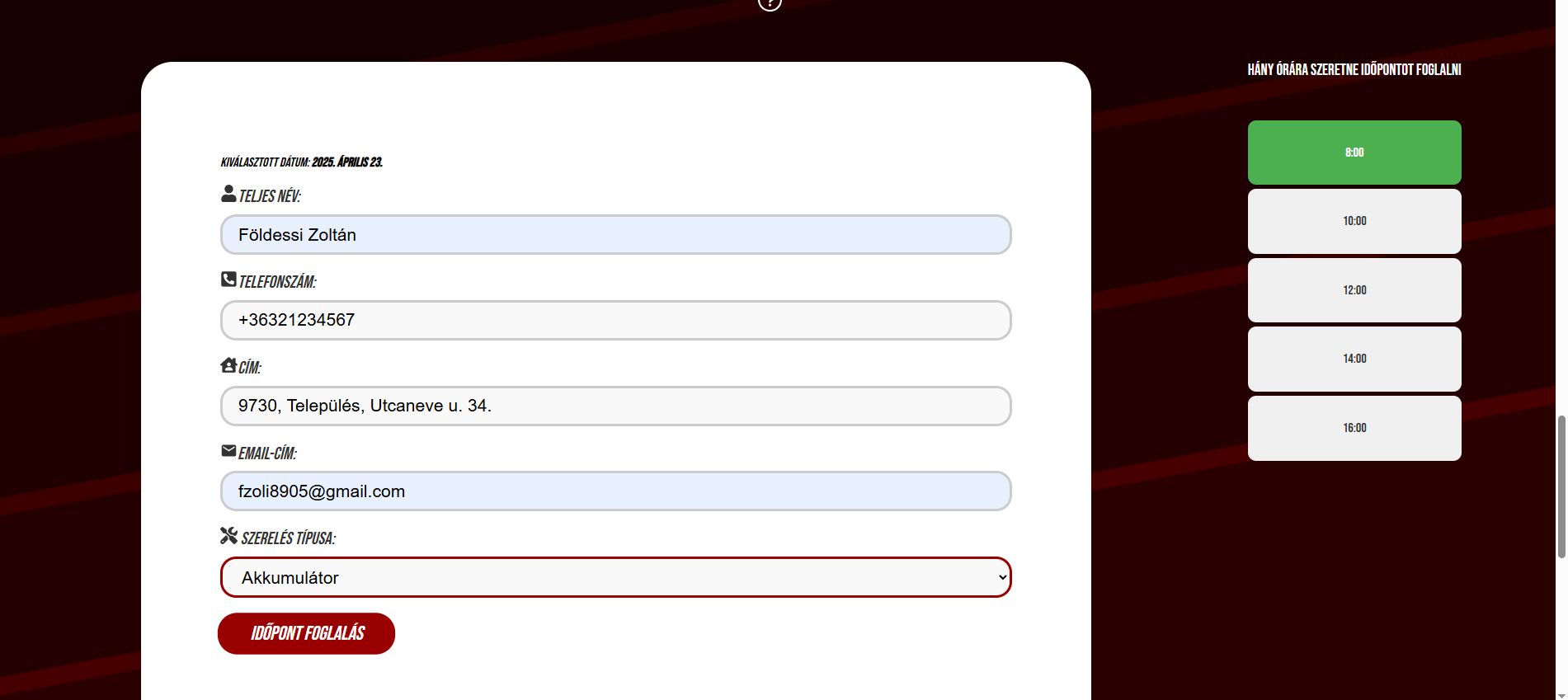
. ábra: Időpontfoglalás1 képernyőkép



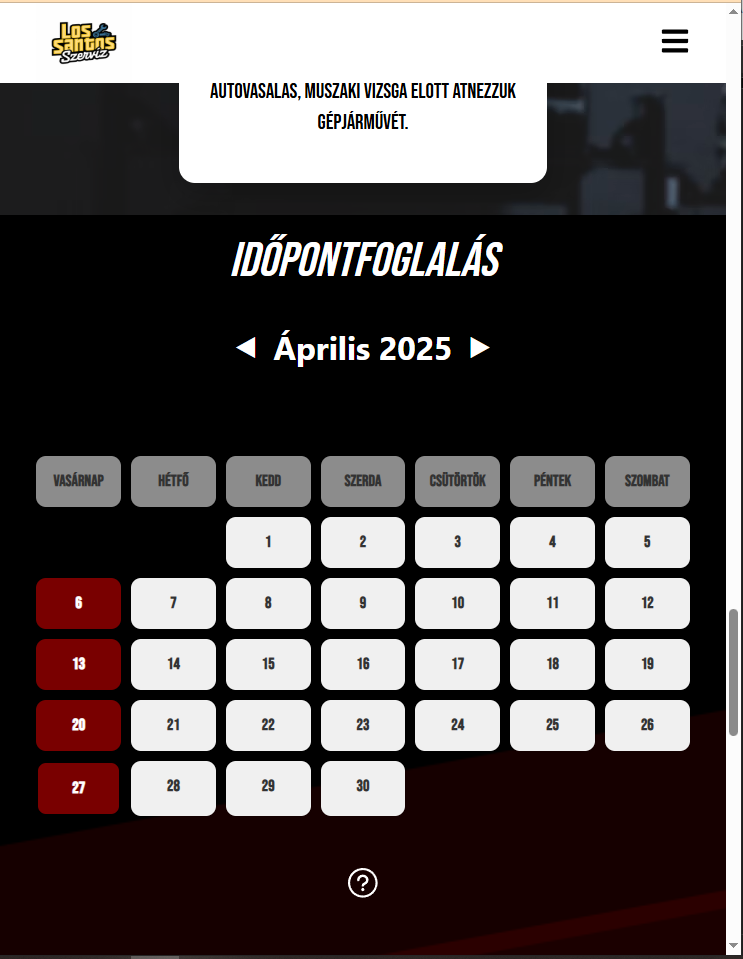
. ábra: Időpontfoglalás2 képernyőkép



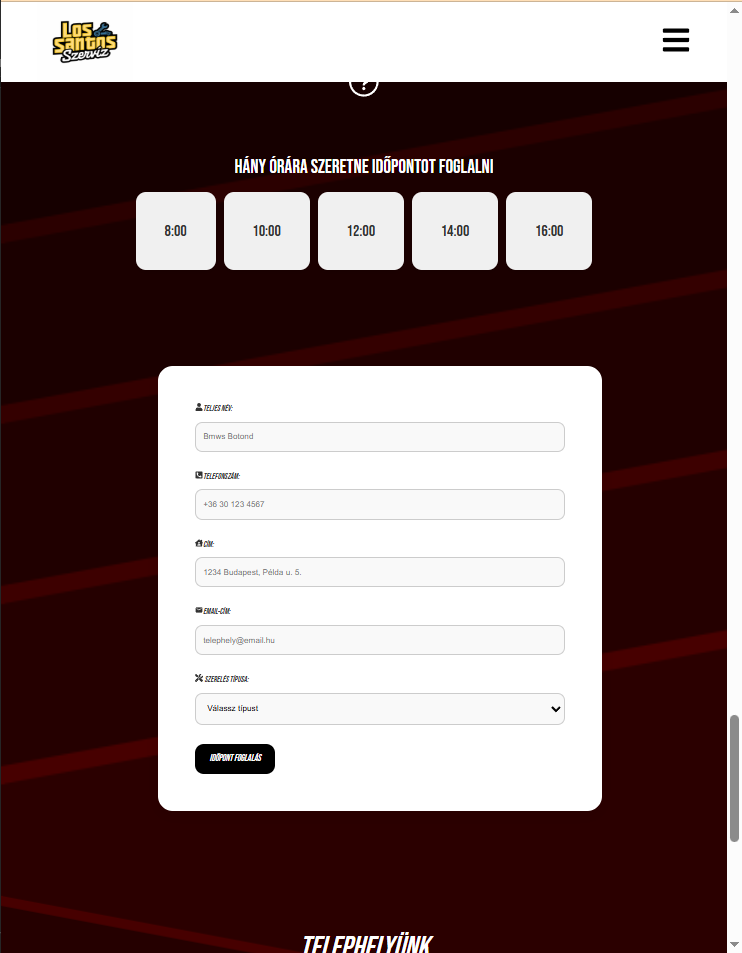
. ábra: Időpontfoglalás űrlap1 képernyőkép



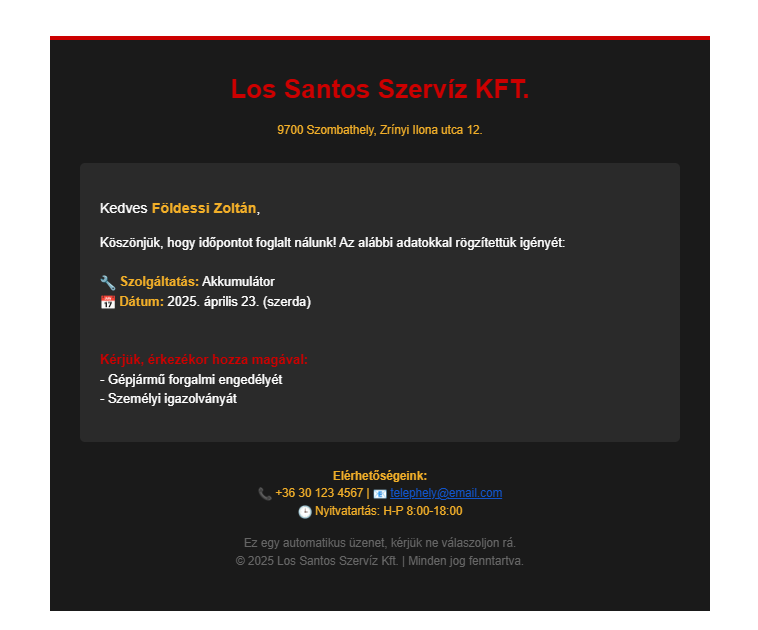
. ábra: Időpontfoglalás űrlap2 képernyőkép



. ábra: Időpontfoglalás reszponzív képernyőkép



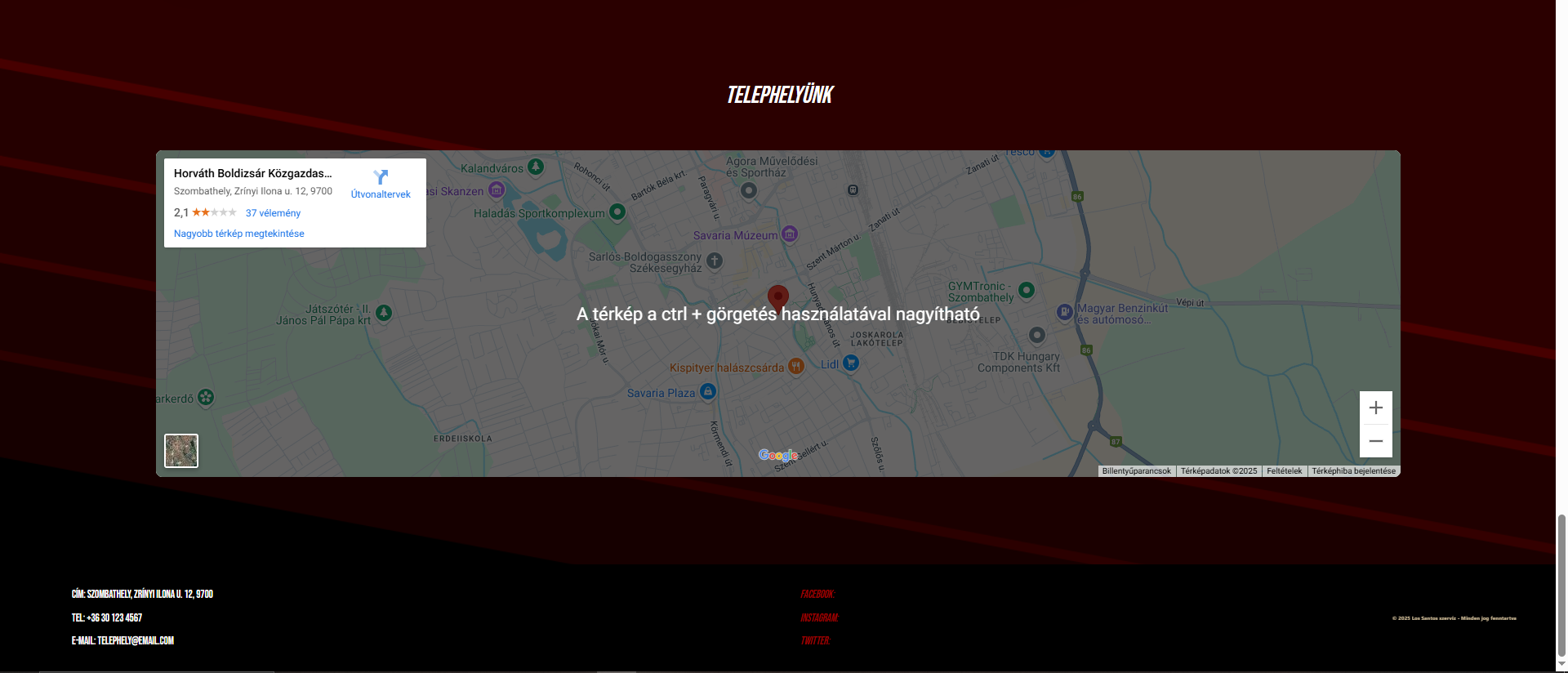
. ábra: Időpontfoglalás űrlap reszponzív képernyőkép



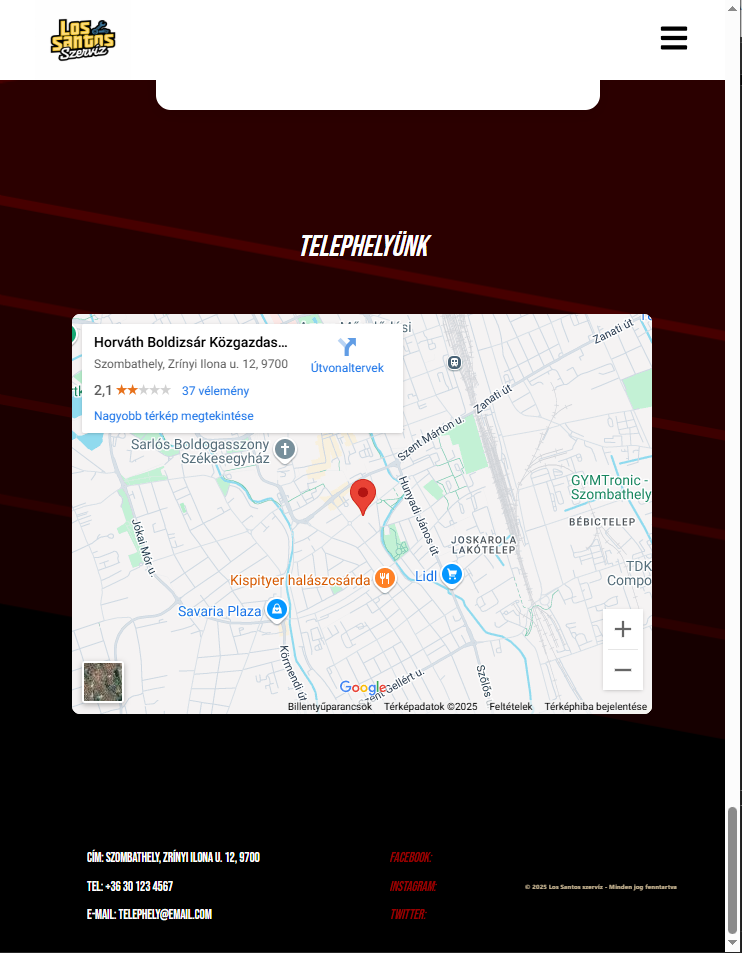
. ábra: Visszaigazoló E-Mail képernyőkép

### 5.Footer

Weboldalunk alján megtalálható autószerelő műhelyünk elérhetősége, illetve elhelyezkedése, melyet Google Maps-al szemléltetünk az ügyfelek számára, megkönnyítve a tájékozódást.



. ábra: Elérhetőségek képernyőkép



. ábra: Elérhetőségek reszponzív képernyőkép

**Tesztek**

**ASP.NET teszt**

Ez a dokumentáció a **Jarmuszerviz** projekt kontrollereinek egységtesztjeit írja le. A tesztek az **MSTest** keretrendszert és a **Moq** könyvtárat használják a mockolt adatbázis-kontextus létrehozásához. Minden kontrollerhez tartozik egy tesztosztály, amely lefedi a CRUD műveleteket és a hibakezeléseket.

**Általános struktúra**

* **Mockolt kontextus**: Az adatbázis műveleteket egy Mock<JarmuSzervizContext> és Mock<DbSet<T>> segítségével szimulálják.
* **TestInitialize**: Minden tesztosztályban a SetUp metódus inicializálja a mock objektumokat és a kontrollert.
* **Metódusok**: Minden tesztmetódus egy adott funkcionalitást ellenőriz, például:
  + Get\_ReturnsAllX: Ellenőrzi, hogy az összes entitás visszaadásra kerül-e.
  + Get\_WithId\_ReturnsCorrectX: Ellenőrzi, hogy egy adott ID-val rendelkező entitás helyesen visszaadásra kerül-e.
  + Post\_InvalidRequest\_ReturnsBadRequest: Ellenőrzi, hogy érvénytelen kérés esetén hibát dob-e a rendszer.
  + Delete\_ExistingX\_ReturnsNoContent: Ellenőrzi a törlés sikerességét és az adatbázis változásait.

**Főbb tesztek**

1. **AlkalmazottakControllerTests**
   * **Get\_ReturnsAllAlkalmazottak**: Ellenőrzi, hogy a Get metódus visszaad-e 2 alkalmazottat.
   * **Post\_InvalidRequest\_ReturnsBadRequest**: Üres név és jelszó esetén BadRequest eredményt vár.
2. **AlkatreszekControllerTests**
   * **Delete\_ExistingAlkatresz\_ReturnsNoContent**: Ellenőrzi, hogy a törlés meghívja-e a Remove és SaveChanges metódusokat.
3. **FelhasznaltAlkatreszekControllerTests**
   * **Put\_InvalidRequest\_ReturnsBadRequest**: Ellenőrzi, hogy eltérő ID-k esetén BadRequest válasz érkezik-e.
4. **IdopontFoglalasokControllerTest**
   * **Delete\_ExistingFoglalas\_ReturnsNoContent**: Ellenőrzi a státuszkódot és az adatbázis módosítását.
5. **JarmuvekControllerTest**
   * **Post\_ValidRequest\_ReturnsCreated**: Ellenőrzi, hogy egy új jármű létrehozása Created státuszt eredményez.
6. **JavitasokControllerTests**
   * **Put\_InvalidRequest\_ReturnsBadRequest**: Ellenőrzi, hogy eltérő ID-val történő frissítés hibát jelez-e.
7. **UgyfelekControllerTest**
   * **Delete\_NonExistingUgyfel\_ReturnsNotFound**: Ellenőrzi, hogy nem létező ügyfél törlése NotFound-ot ad vissza.

# Továbbfejlesztési lehetőségek

Weboldal:

A meglévő időpontfoglalási rendszert továbbfejleszthetjük, hogy az ügyfelek valós időben lássák a még elérhető, valamint a már foglalt időpontokat, lehetőséget teremtve arra, hogy módosítani lehessen a foglalásokat. E-mailes és SMS értesítésekkel emlékeztethetjük a felhasználókat a közelgő foglalásaikra, így csökkenthetjük a lemondásokat, mely által növelhetjük a felhasználói elégedettséget. A felhasználók véleményének, elégedettségének, értékelésének integrálása az oldalra segíthet növelni a bizalmat a potenciális ügyfelek körében.

Kérdőívek beépítése, mellyel teljes körű képet kapva maximalizálni lehetne az ügyfelek elégedettségét.

Tartalom

[Bevezetés 1](#_Toc196683866)

[Program célja (Frontend): 1](#_Toc196683867)

[Célközönség: 1](#_Toc196683868)

[Betűszavak és meghatározások: 1](#_Toc196683869)

[Program általános leírása 2](#_Toc196683870)

[Követelmények, tervezés és a rendszer jellemzői – megvalósítási terv 2](#_Toc196683871)

[Tervezés – feladatok felosztása 3](#_Toc196683872)

[Adatbázis ER modell: 4](#_Toc196683873)

[Táblák és Mezők 4](#_Toc196683874)

[JarmuSzerviz Web API 7](#_Toc196683875)

[Jelszó Ellenőrzés 9](#_Toc196683876)

[Amikor egy alkalmazott bejelentkezik, a rendszer a következőt hajtja végre: 9](#_Toc196683877)

[VerifyPasswordHash Metódus 9](#_Toc196683878)

[Működés 9](#_Toc196683879)

[1. A rendszer kiveszi a salt-ot az alkalmazott Jelszo mezőjéből. 9](#_Toc196683880)

[2. A felhasználó által megadott jelszót újra hash-eli a salt segítségével. 9](#_Toc196683881)

[3. Összehasonlítja az új hash-t az adatbázisban tárolt Jelszo\_hash értékkel. 9](#_Toc196683882)

[4. Ha egyezik: sikeres bejelentkezés. Ha nem: hiba. 9](#_Toc196683883)

[Felület vázlatos megtervezése: 10](#_Toc196683884)

[Segédoldalak, információs felületek, linkek: 10](#_Toc196683885)

[Komponenseinek technikai leírása 11](#_Toc196683886)

[Használatának rövid bemutatása: 11](#_Toc196683887)

[1.Főoldal 11](#_Toc196683888)

[2.Rólunk oldal 12](#_Toc196683889)

[3.Szolgáltatások oldal 15](#_Toc196683890)

[4.Időpontfoglalás oldal 17](#_Toc196683891)

[5.Footer 22](#_Toc196683892)

[Továbbfejlesztési lehetőségek 25](#_Toc196683893)

**Ábrajegyzék**

[1. ábra: ER-Modell 3](#_Toc196672349)

[2. ábra: Főoldal képernyőkép 9](#_Toc196672350)

[3. ábra: Főoldal reszponzív képernyőkép 10](#_Toc196672351)

[4. ábra: Rólunk oldal képernyőkép 11](#_Toc196672352)

[5. ábra: Rólunk reszponzív képernyőkép 11](#_Toc196672353)

[6. ábra: Rólunk reszponzív képernyőkép 12](#_Toc196672354)

[7. ábra: Szolgáltatások oldal képernyőkép 13](#_Toc196672355)

[8. ábra: Szolgáltatások reszponzív képernyőkép 14](#_Toc196672356)

[9. ábra: Időpontfoglalás1 képernyőkép 16](#_Toc196672357)

[10. ábra: Időpontfoglalás2 képernyőkép 16](#_Toc196672358)

[11. ábra: Időpontfoglalás űrlap1 képernyőkép 17](#_Toc196672359)

[12. ábra: Időpontfoglalás űrlap2 képernyőkép 17](#_Toc196672360)

[13. ábra: Időpontfoglalás reszponzív képernyőkép 18](#_Toc196672361)

[14. ábra: Időpontfoglalás űrlap reszponzív képernyőkép 19](#_Toc196672362)

[15. ábra: Visszaigazoló E-Mail képernyőkép 20](#_Toc196672363)

[16. ábra: Elérhetőségek képernyőkép 21](#_Toc196672364)

[17. ábra: Elérhetőségek reszponzív képernyőkép 22](#_Toc196672365)