Szegedi Szent Benedek School Of Business Technikum

Informatika és távközlés

Szoftverfejlesztő és tesztelő technikus

506131203

Torta Webshop

A vizsgaremek készítői:

Veres Szabolcs 2/14 Szft

Tasic Nenad 2/14 Szft

Suti Zsolt 2/14 Szft

A témavezető neve:

Bodrogi Péter Róbert

Szeged, 2023

1. TARTALOMJEGYZÉK

2. Dokumentáció

Projekt neve: Torta Webshop

Projekt felelősök: Veres Szabolcs, Tasic Nenad, Suti Zsolt

Kapcsolattartó email: sabolc120@gmail.com, s.zsili98@gmail.com, neca.tasic03@gmail.com

Képzés neve: Szoftverfejlesztő és tesztelő

Webshop főtevékenysége: Torták megrendelése

Webshop altevékenysége: Tortáink népszerűsítése

3. Záródolgozat témája

Egy Webshop bemutatása, ahol a tortákat tudjuk népszerűsíteni és egy működő adatbázisból tudjuk őket értékesíteni.

Témaválasztás okai:

- Olyan témát akartunk, amit kortól függetlenül el tudunk adni bárkinek
- Könnyű bővíteni (további sütemények hozzáadása)
- Megnyerő desingt lehet neki adni, ezáltal több vásárlót tudunk elérni

Tervezendő program funkciói:

- Regisztráció
- Bejelentkezés (Admin és vásárló jogosultság)
- Kosárba helyezés és kivétel
- Kosár tartalmának követése
- Oldalak közötti navigáció

A webshopunk minden olyan embernek ajánlanánk, akik szeretik az édességeket és torta rendelés előtt állnak bármilyen eseményre.

4. Torta Áruház bemutató

Célunk az volt, hogy képesek legyünk létrehozni egy valós életbeli webshopot, mint egy csapat tapasztalatlan diák.
Elkezdtünk kutatni és utána nézni milyen technológiákat kell majd használnunk, és a készülődés is elkezdődött. Végül egyhangúan arról döntöttünk, hogy kiválasztunk egy közkedvelt és finom dolgot, amivel elérhetjük az emberek pusztán a látvánnyal. A webshopban a modern kávézói hangulatot raktuk bele, mint dizájn és a menüt is ugyanígy próbáltuk meg kitalálni.

A képek mind egy szálig AI(Aritifical Intelligence) által generált képek, amik Szabi ötletei voltak és végűl hatalmas siker lett. A webshop tapasztalatok hiánya miatt nem tükrözi teljes mértékben egy valódi webshop professzionalitását, tehát a biztonságon nem volt nagy hangsúly ugyanez vonatkozik a távolabbi API megoldásokra, gondolunk most arra hogy egy diák projektben nyilvánvalóan nem rakunk be fizetési lehetőségeket.

5. Projekt céljai és funkció

Funkcionalitás, metódusok

- Külön tortalap a menüben elérhető tortáknak
- Torták kosárba helyezése, mennyiségek változtatása
- Adminisztrációs oldal a rendelések kezelésé, törlése
- Rendelések leadása, rendelési állapotok változtatása
- Kizárólag bejelentkezés után lehessen rendelni
- Kizárólag adatok megadása után lehessen rendelni

Dizájn, User Experience

- Modern kávézói hangulat és atmoszféra
- Kivételes, inkább ínyenc étterem mint sem normális
- Modern kávézói hangulat mellet télies hangulat is
- Ügyfél orientáció, könnyű kezelhetőség
- Törekvés a modern dizájnra, kevés szövegre és megfelelő hangulatra

6. Architektúra és Technológiák

Frontend, kliens oldal

A kiválasztott frontend framework az Angular lett ami egy Typescript alapú framework, rendkívül népszerű az Single Page Applications-nek köszönhetően, ennek több oka is van

- Nagyon egyszerű kezelhetőség, professzionális megközelítés annak irányába hogy minél hatékonyabb és egyszerűbb legyen weboldalt modulokra feldarabolni ami igazán megkönnyíti a felelős dolgát.
- Typescript alapú nyelv, ami egy jobb JavaScript, és a legtöbb frontend framework kizárólag JavaScriptet használ míg az Angular az Typescriptet használ, ami egy biztosabb és tágabb JavaScript, röviden annyit jelent hogy több lehetőség van vele mint más frontend frameworkkel.

Backend, a szerver oldal

A kiválasztott backend framework az egy Java alapú framework, Spring Boot, rendkívül népszerű annak köszönhetően hogy rengeteg funkciót biztosít bármiféle nehezebb teendő nélkül, ennek is több oka van miért esett erre a választás

- Kiváló felhasználói kezelési függőség, magyarul kiegészítő ami kizárólag Spring Boot-ban elérhető, ennek köszönhetően a projekt legnehezebb része már nem is volt annyira nehéz.
- Java alapú framework ami az egyik legerősebb és legjobb lehetőségeket nyújtó programozási nyelv és egyben a C# őse is.

7. Fejlesztési folyamatok és munkálatok

Frontend, kliens oldal-Veres Szabolcs, Suti Zsolt
El kellett érnünk, hogy minél több lehetőség legyen a magas színvonalú dizájn miatt, ezért szükség volt egy ehhez megfelelő frameworkre, emellet különböző technológiákat is kellett használnunk mint például..

- Stable Diffusion - Annak az AI-nak a neve aminek köszönhetjük a magas színvonalú képeinket, anélkül hogy aggódnunk kelljen a szerzői jogok miatt, vagy több órát eltölteni azzal hogy saját képeket próbáljunk készíteni
- Először utána kellett nézni annak hogy egyáltalán mit akarunk elérni, gyakoroltuk az OpenAI képgeneráló technológiáját hogyan érhetjük el azt hogy minél stílusosabb és odaillő képeket tudjunk készíteni, emellet kiválasztottuk a színsémát is.

Backend, szerver oldal-Veres Szabolcs, Tasic Nenad
Mivel olyan dolgok voltak a követelményekben mint felhasználói kezelés, kellett találnunk valami módot arra hogy egy tapasztalatlan diák is képes legyen egy ilyen komplett rendszert megírni

- Spring Boot tökéletes választás lett, mivel volt külön beépített lehetőség erre
- Emellet, tökéletesen passzolt az adatbázissal.

Adatbázis, adatkezelés-Veres Szabolcs, Tasic Nenad
Egy minél egyszerűbb adatbázist próbáltunk választani amihez volt már tapasztalatunk és ebbe a képbe bele illett a MySQL/phpmyadmin

8. Rendszerkomponensek, főbb metódusok

Felhasználói rendszer kezelése

Itt olyan dolgokat kellett beállítani mint a jogkörök, jelszó titkosítás.

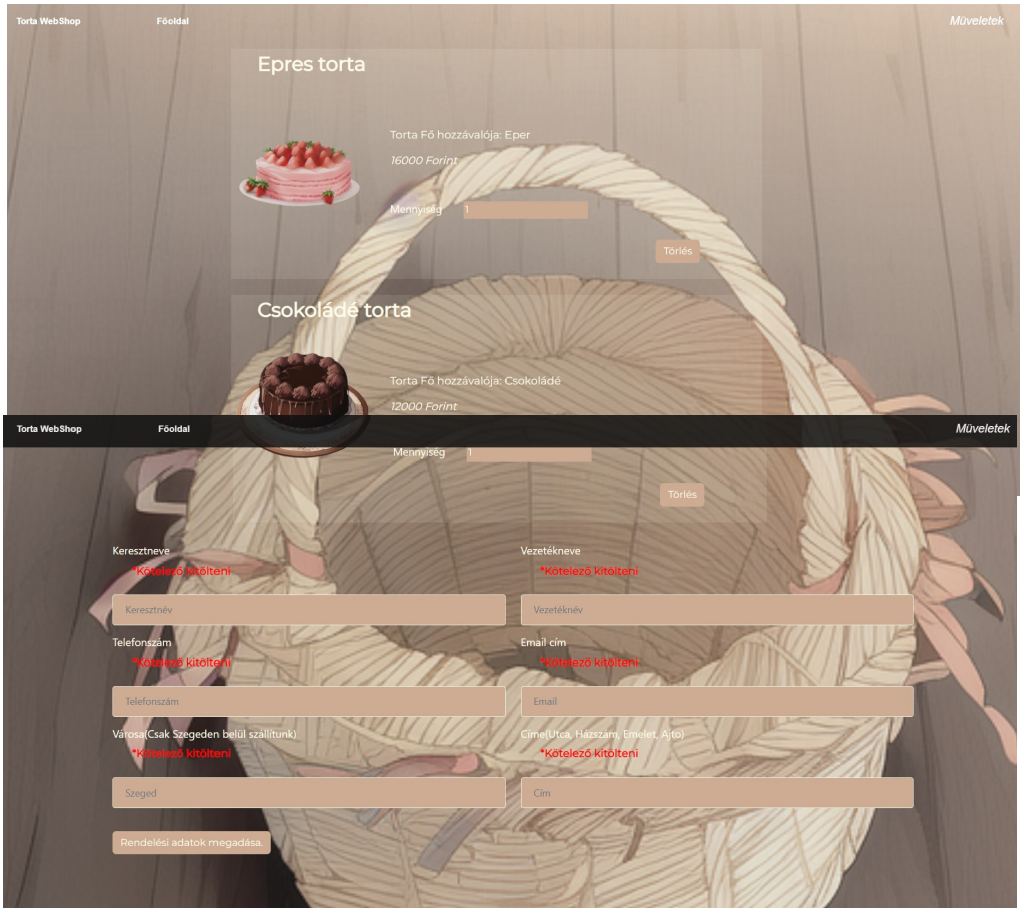
- Jogkörök: Ezek az adatok a frontendben lettek kezelve, hogy ki mit tud megtenni attól függően hogy kinek van ADMIN rangja, és kinek van kizárólag USER jogköre. Azonban a backend-ből kiindulva mindenki “user”-ként kezdd, amit kizárólag adatbázison keresztül lehet változtatni, ennek az egésznek annyi volt a titka hogy egy lokális adattárolót használjunk a böngészőben, aminek annyi a lényege hogy bejelentkezéskor a felhasználó jogköre el mentődik egy belső raktárba a böngészőn belül amit tudunk használni ellenőrzésként, például ha a felhasználónak kizárólag “USER” jogköre van,akkor a frontendben beállított filter nem fogja tovább engedni az adminisztrációs oldalra hanem egyszerűen csak vissza dobja a főoldalra a kíváncsiskodó felhasználót aki az URL szerkesztésével próbált rosszalkodni.
- Jelszó titkosítása: Ez egy elképesztően hasznos funkció ami elérhető Spring Boot-ban, a jelszót már regisztráció után rögtön titkosításra kerül az adatbázisban, amit szinte az adatbázis adminisztrátor sem képes megfejteni vagy kitalálni, mert lényegében csak ennyit lát:
  $2a$10$XIoRKmFbTfKjXlW1schkUOb.utcqpxjx6s3JkNaRLih…

Rendszerkomponensek, főbb metódusok 2.

Kosárba helyezés, kosár szerkesztése

Erre már nem volt külön kisegítő könyvtár a Spring Boot-ban ezért el mondhatjuk ez volt a legnehezebb része az egésznek a rendelések leadása után..

- Ahogyan minden eggyes tortának az adata az adatbázisban el van tárolva, ezt is úgy szintén el kellet tárolni, és azt is hogy mikor van szó rendelésről és mikor csak kosár termékről.
- Ennek a táblának 11 oszlopa van amibe bele tartozik a torta adatai, felhasználó adatai, a kosárba rakott torta mennyisége, az hogy rendelésről vagy kosár termékről van-e szó ami röviden egy True or False, és arról hogy milyen torta is van a felhasználó kosarában (A torta azonosítója) itt 2 Foreign key-t kellet használnunk.
- Azt hogy hogyan rakjuk be ezeket az adatokat egy táblába a kosárra kattintáskor annyiból állt hogy a torta lapon elérhető adatokat nyertük ki, és az éppen bejelentkezett felhasználó adatait nyertük ki..(Torta neve, ára, felhasználónév, torta azonosító…stb) mivel ez a rengeteg adat lekérdezésre kerül mikor kiválasztunk egy tortát és meglátogatjuk annak a tortának a lapját.
- Akkor, még azt is el kellett érnünk hogy a kosarat szerkeszteni lehessen, ezért valójában csak 2 szerkesztési lehetőség volt, ami a torta mennyiségének a változtatása volt, vagy pedig a torta teljes mértékben kitörlése a kosár táblából az adott felhasználó számára.



Rendszerkomponensek, főbb metódusok 3.

Rendelés leadása, lépései.

A rendelés leadásának több lépése is van, és itt sokat kellet trükközni is, ez a lépés volt a legnehezebb az egész projektben és főképpen itt voltak elakadások..

- Mi sem tudtuk eddig, azt, hogy több form-ot egyszerre nem lehet elküldeni a kiszolgáló felé, ezzel az volt a hatalmas probléma hogy a Kosár oldalon megjelent torták mind egy-egy form-ba tartoztak ezért külön erre is ki kellett találni valamit, ha esetleg nem egy fajta tortát hanem több fajtát szeretne az ügyfél egyszerre rendelni, ezt kizárólag az Angular segítségével tudtuk megoldani, ami egy annyira összettett lépés volt hogy erről külön nem lesz bejegyzés a dokumentációban.
- Ahogyan már említettem, a kosár táblában van egy True or False érték is, ezt az értéket kellett szerkeszteni vagyis “UPDATE”-elni a rendelés leadásakor, ez azért volt fontos, hogy miután egy rendelés le lett adva akkor az adatbázisban is legyen jelezve hogy már nem csak egy kosár termékről van szó hanem egy valódi rendelésről.
- Emellet, értelemszerűen hogy kövessük a valós életbeli példákat arra is szükség volt hogy a felhasználótól elkérjük a szállítási információkat, és az elérhetőségi adatokat mint a telefonszámot és az email-címet, ezeket az adatokat külön kötelező volt megadni, ha bármi is kimaradt, arra a weboldal felhívta a figyelmét a felhasználónak, addig amíg nem lett megadva minden szükséges adat addig nem lehetett leadni a rendelést.

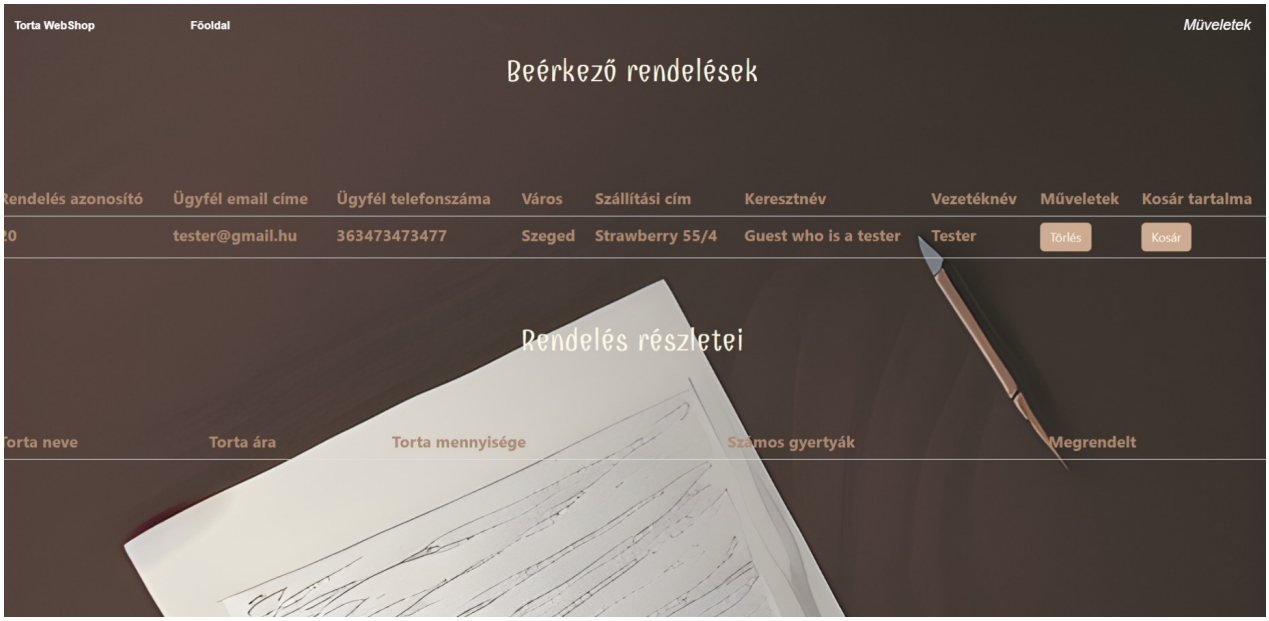
```
3 usages
@Autowired
private OrderDetailsRepo orderDetailsRepo;

1 usage
@Autowired
private CakeBasketRepo cakeBasketRepo;
1 usage  Sabolc120
@Override
public OrderDetails completeOrderPost(
                                    Long userId, String firstName, String lastName, String customerPhoneNumber, String customerEm
                                    String address, String userName) {
    OrderDetails orderDetails = new OrderDetails();
    UserModel userModel = userRepo.findById(userId).orElse( other: null);
    orderDetails.setCustomerPhoneNumber(customerPhoneNumber);
    orderDetails.setCustomerEmail(customerEmail);
    orderDetails.setFirstName(firstName);
    orderDetails.setLastName(lastName);
    orderDetails.setUserModel(userModel);
    orderDetails.setCity("Szeged");
    orderDetails.setAddress(address);
    orderDetails.setUserName(userName);
    return orderDetailsRepo.save(orderDetails);

}

1 usage  Sabolc120
@Override
public List<OrderDetails> getOrders() { return orderDetailsRepo.findAll(); }
```

9. Rendszerkomponensek, főbb metódusok

Adminisztrációs felület a személyzet számára

Szükségünk volt egy külön oldalra ahol a kosár tartalmát/rendeléseket tudtuk figyelemmel követni, ezt is igazán érdekesen oldottuk meg…

- Elsősorban, nem hagyhattuk hogy egy “USER” jogkörű felhasználó képes legyen ezt az oldalt látni, éppen ezért egyszerűen mondva ha bármilyen felhasználó meg is találja a megfelelő elérési útvonalat, azt vissza rúgjuk a főoldalra.
- Ezen az oldalon rengeteg adat elérhető, lényegében a már említett 11 oszloppal rendelkező adatbázis tábla kerül lekérdezésre. Itt kettő opciónk van, az adott kosár “entry” vagyis bejegyzés tartalmának megnézése hogy milyen termékek is vannak abban a kosárban, a példa kedvéért, Bence a kosarába helyezett 2 tortát, Dávid pedig 3 tortát, akkor ez 2 külön bejegyzést ölel fel, és ennek a kettő felhasználónak a kosár tartalmát mindenestül le lehet kérdezni ezen az oldalon, a másik itteni lehetőség az a “Delete”.



Tesztelés, Minőségbiztosítás

A tesztelésre használt program

Egy automatikus tesztet választottunk ami Seleniumot használ ami egy C# alapú tesztelő "bot". Lényege annyi, hogy automatikus hajt végre különböző műveleteket amiket mi írtunk meg előre, ebben az esetben a "tesztelőnk" regisztrál, bejelentkezik, kijelentkezik..és más egyéb dolgokat tesz mint amit egy alap felhasználó tenne, több oka is van annak hogy ezt választottuk.

- Első és legfontosabb oka az volt hogy egyikünk sem ismert más féle tesztelő programot
- Egyszerűbb volt egy botot készíteni ami egy felhasználó viselkedését utánozza mint sem egy manuális tesztelést végezni

Minőségbiztosítás, megbízhatóság

Mint már említettem, a projekt nem egy valós biznisz alapú projekt hanem egy iskolai vizsga, éppen ezért nem volt annyira nagy hangsúly a biztonságon vagy pedig az adatok kezelésén, azonban a példa kedvéért beleraktunk néhány érdekességet

- Admin és Felhasználó jogkörök külön-külön engedélyekkel, azért hogy egy általános fel használó ne tudjon kotorászni a rendelések és a felhasználói adatok között.
- Adatbázisban jelszó titkosítás, ez egy bónusz amit főképpen Spring Boot-nak köszönhet tünk.
- Kivételkezelés, a weboldalon lehet nem szembetűnő azonban vannak dolgok amiket nem lehet megtenni annak az érdekében hogy ne legyenek váratlan hibák.

```
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Support.UI;
using SeleniumExtras.WaitHelpers;
using IJavaScriptExecutor = OpenQA.Selenium.IJavaScriptExecutor;
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        // Set the path to the ChromeDriver executable
        ChromeOptions options = new ChromeOptions();
        using (IWebDriver driver = new ChromeDriver(@"E:\Opera_letöltések\chromedriver_win32\chromedriver.exe"))
        {
            WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));


            //Elmegy a főoldalra
            driver.Navigate().GoToUrl("http://localhost:4200/");
            driver.Manage().Window.Maximize();
            Thread.Sleep(2000);

            //Rá kattint a menübuttonre
            IWebElement menuButton = driver.FindElement(By.Id("menu-btn"));
            wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.Id("menu-btn")));
            menuButton.Click();

            //Menüből rákattint a regisztrációs buttönre
            IWebElement regLink = driver.FindElement(By.Id("reg"));
            wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.Id("reg")));
            regLink.Click();
```

10.Adatbázis, és annak a működése

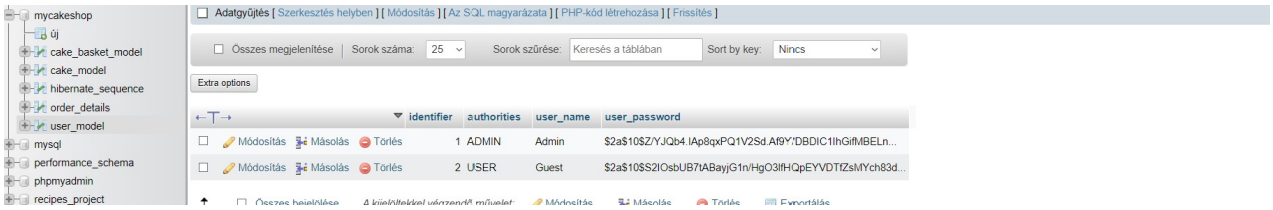
Használt program

Mint program, az XAMPP-ot használtunk azon belül pedig a MySQL phpmyadmint, ennek is nagyon egyszerű oka volt, mindannyiunk ehhez értett a legjobban és már mindannyian dolgoztunk vele.

Táblák az adatbázisban

Az projektünkhöz 4-5 táblát használtunk összesen, ezekbe tartoznak a következők:

- User_model - Lényegében itt vannak eltárolva a regisztráltak
- Order_details - Itt vannak eltárolva a kapcsolattartási adatok, és a felhasználó adatai mikor rendelést ad le.
- Cake_model - Itt vannak azok a torták amiket a weboldalon lehet látni, mivel azok mind egy adatbázisból vannak lekérdezve.
- Cake_basket_model - A legfontosabb táblázat az összes közül, valójában itt van mind a felhasználó adatai és a kosarának a tartalma is mind a torta adataival, ez a tábla kerül lekérdezésre mikor a rendeléseket kezeljük.



Adatbázis, és annak a működése 2.

Adatbázis tábla kapcsolatok

- Cake_model - Egyedülálló tábla, nem rendelkezik foreign key-el.
- Order_details - Már nem egy egyedülálló tábla, rendelkezik egy foreign key-el ami a felhasználó azonosítóra mutat
- Cake_basket_model - Rendelkezik kettő foreign key-el, az egyik a torta azonosítójára mutat rá, a másik pedig a felhasználói fiókra.
- User_model - Egyedülálló tábla, próbáltuk a felhasználói adatokat mint a jelszavat semmivel sem keverni azért hogy csak egyszer kelljen lekérdezni az egész session alatt.

Adatbázis lekérdezések, backend

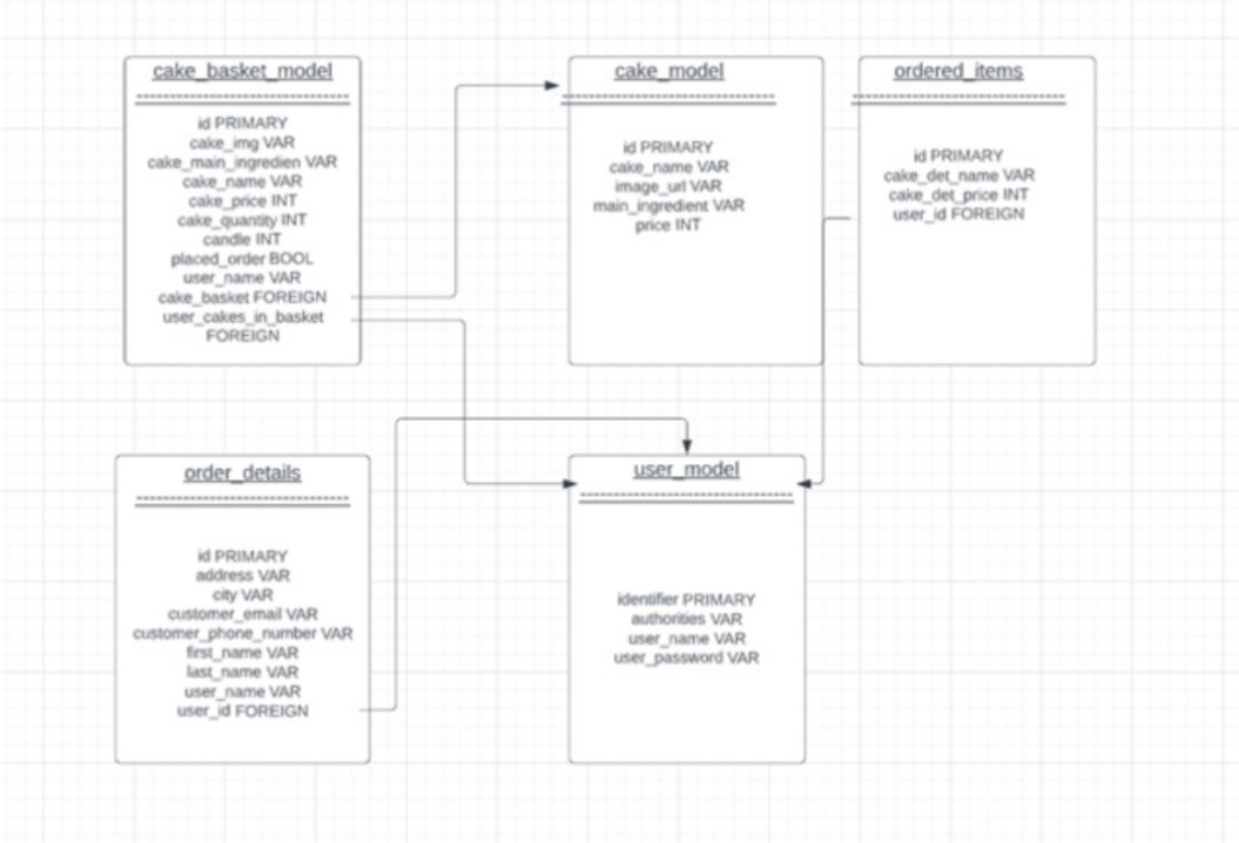
Itt natív lekérdezéseket használtunk, más szóval nem a beépített ORM-ot (Object Relational Mapping) ami annyit takar hogy a querryket mi írtuk meg nem pedig automatizáltunk, ennek is több oka volt:

- A Spring Boot-ban rendelkezésre álló automatikus lekérdezések nem voltak elég részletesek a projekt követelményeivel szemben
- Döntenünk kellett hogy mindent le kérdezünk és a bonyolultabb módokat választjuk-e vagy pedig mi magunk intézzük el a magunk módján
- Túlzottan sok volt a hiba lehetőség, nem volt elegendő redundancia és rugalmasság az automatikus lekérdezésekben.

```
1 usage  Sabolc120
@Query(value = "SELECT * FROM cake_basket_model WHERE cake_basket = :cake_basket AND user_cakes_in_basket = :user_cakes_in_basket
CakeBasketModel findByCakeModelId(@Param(value="cake_basket")
                                  Long cake_basket, @Param(value="user_cakes_in_basket") Long user_cakes_in_basket);

no usages  Sabolc120
@Query(value = "SELECT * FROM cake_basket_model WHERE cake_basket = :cake_basket AND user_cakes_in_basket = :user_cakes_in_basket
CakeBasketModel findByCakeModel_Id(@Param(value="cake_basket")
                                   Long cake_basket, @Param(value="user_cakes_in_basket") Long user_cakes_in_basket);

1 usage  Sabolc120
@Query(value = "SELECT * FROM cake_basket_model WHERE id = :id AND user_cakes_in_basket = :user_cakes_in_basket", nativeQuery = t
CakeBasketModel findById(@Param(value="id")
                         Long id, @Param(value="user_cakes_in_basket") Long user_cakes_in_basket);

no usages  Sabolc120
@Query(value = "SELECT * FROM cake_basket_model WHERE user_cakes_in_basket = :user_cakes_in_basket", nativeQuery = true)
List<CakeBasketModel> findAllById(@Param(value="user_cakes_in_basket") Long userId);

1 usage  Sabolc120
@Query(value = "SELECT * FROM cake_basket_model WHERE user_name = :user_name", nativeQuery = true)
List<CakeBasketModel> findAllByUserName(@Param(value="user_name") String userName);
}
```

Adatbázis relációk, adatbázis térkép

Adatbázis relációs térkép Egy ingyenes programot használtunk ennek elkészítésére, mivel a projekt követelményeiben szerepelt.



11. Verziókezelők

Projektmunka során használt verziókezelő programok A projekt teljes életciklusa alatt Github-ot használtunk, a github tökéletes választás lett ide a vetélytársaihoz képest amelynek több oka is van.

- Github desktop alkalmazás ami megkönnyíti a github használatát
- Mindig onnan tudtuk folytatni ahol az egyikünk abba hagyta a projektet
- Rengeteg fájl formátum lehetőség, könnyedén fel tudtuk rakni akár az egész adatbázist bármiféle probléma nélkül.

12. Felhasználói dokumentáció, kézikönyv

Használati útmutató a webshophoz A webshopot nagyon egyszerű használni, teljesen egyforma egy hagyományos webshoppal amit az interneten lehet találni.

● Elsősorban, ahhoz hogy vásárolni lehessen bármit is a menüből, szükséges előtte regisztrálni majd pedig bejelentkezni. Erre azért volt szükség, hogy az adminisztrációs oldalon könnyen szemmel lehessen követni a rendeléseket, és ne legyenek felhasználói fiók nélküli rendelések.

● Amint tudjuk milyen tortát szeretnénk, egyszerűen rá kattintunk a főoldalon található menüből az egyik tortára, miután betöltött a tortalap szükséges de nem kötelező kiválasztani a gyertyát hogy milyen számot használjunk, tehát ha a gyertyák száma 16-ra van állítva, akkor egy 16-os számmal fog jönni a torta házhoz, amennyibben alapértéken van hagyva tehát 0-án, akkor azt úgy vesszük hogy nincsen szükség gyertyákra.

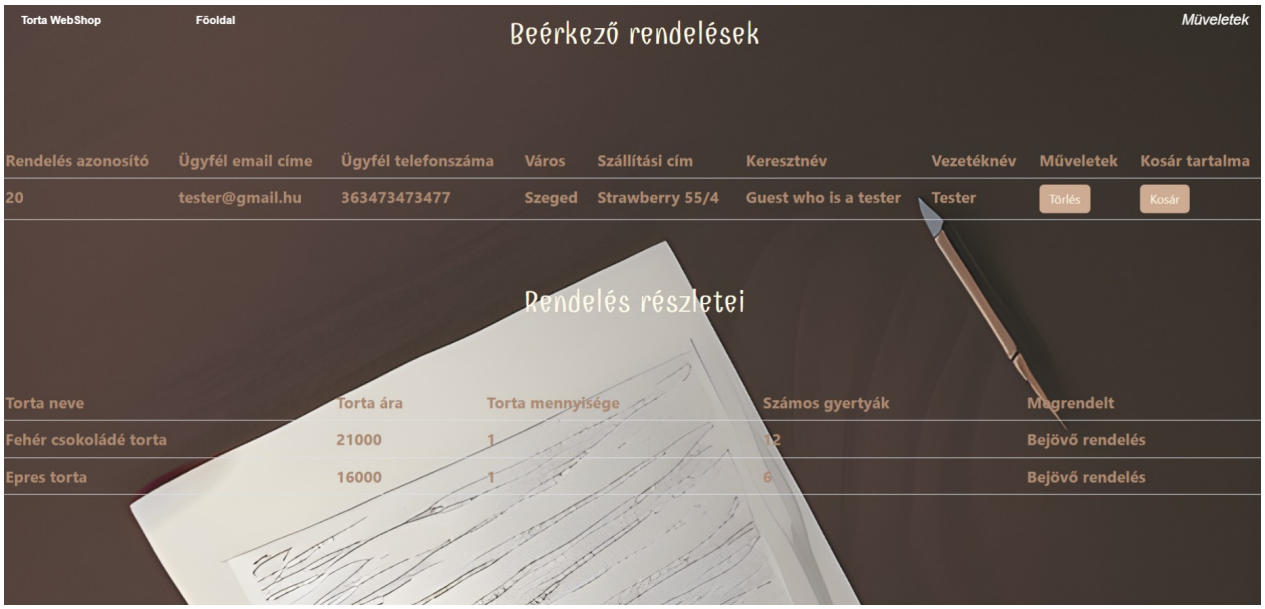
● Miután a kosár gombra kattintunk, kinyitjuk a navigációs sávot ahol lesz egy olyan szekció hogy “Kosaram”. Itt lesznek az eddigi kosárba tett torták ahol majd lehet a mennyiséget változtatni, vagy akár kosár elemet törölni.

● Ezután már csak meg kell adnunk a kapcsolattartói adatainkat, szállítási címet, és miután mind ez megfelel utána le lehet adni a rendelést.

Felhasználói dokumentáció, kézikönyv 2.

Felhasználói útmutató dolgozók számára Rendelkezésre biztosítottunk az adatbázisban egy alap felhasználót amennyibben valaki tesztelni szeretné és megnézni az adminisztrációs oldalt.

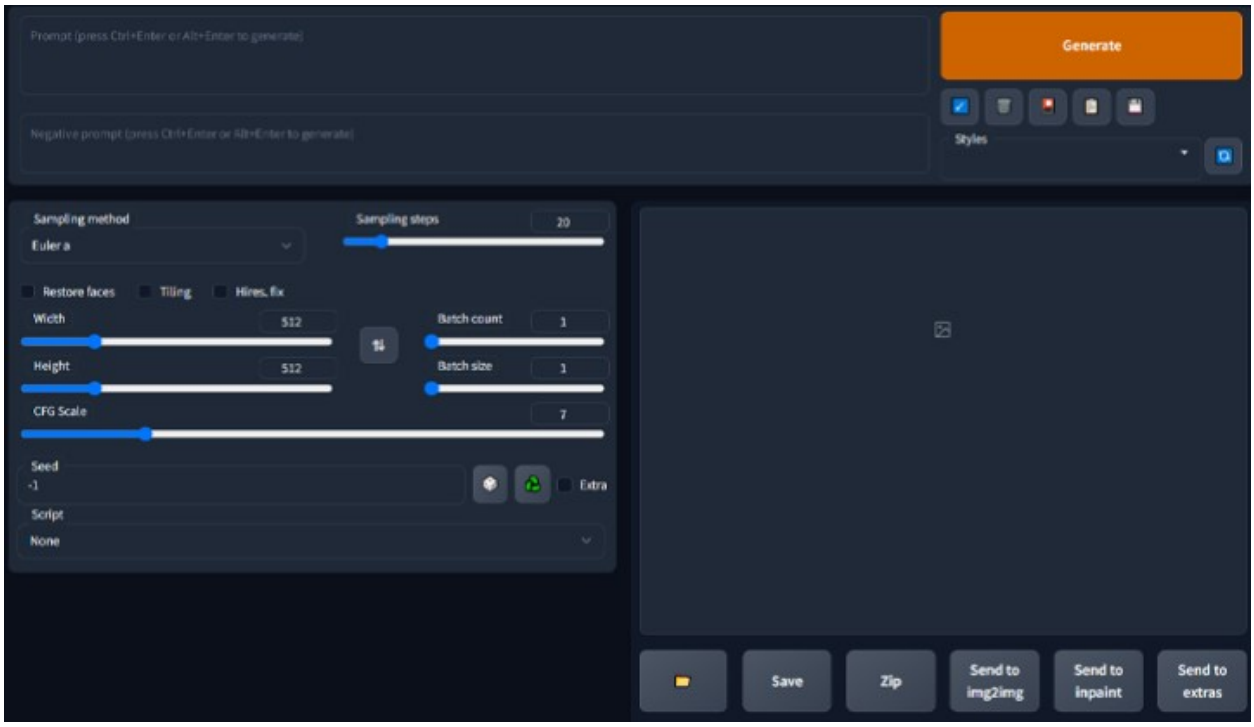
- Először is, be kell jelentkezni az adott fiókkal, például:
- Felhasználónév: Admin, Jelszó: Admin
- Ezután ha sikeres volt a bejelentkezés, látogassuk meg a következő URLT-:localhost:4200/getOrders
- Itt ezen az oldalon látnunk kellene az éppen bejövő rendeléseket.



13. Stable diffusion, a projekthez használt “művész”

Ebben a szekcióban, arról lesz leírás hogyan használtuk a és szereztük meg a “művész”-ünk.

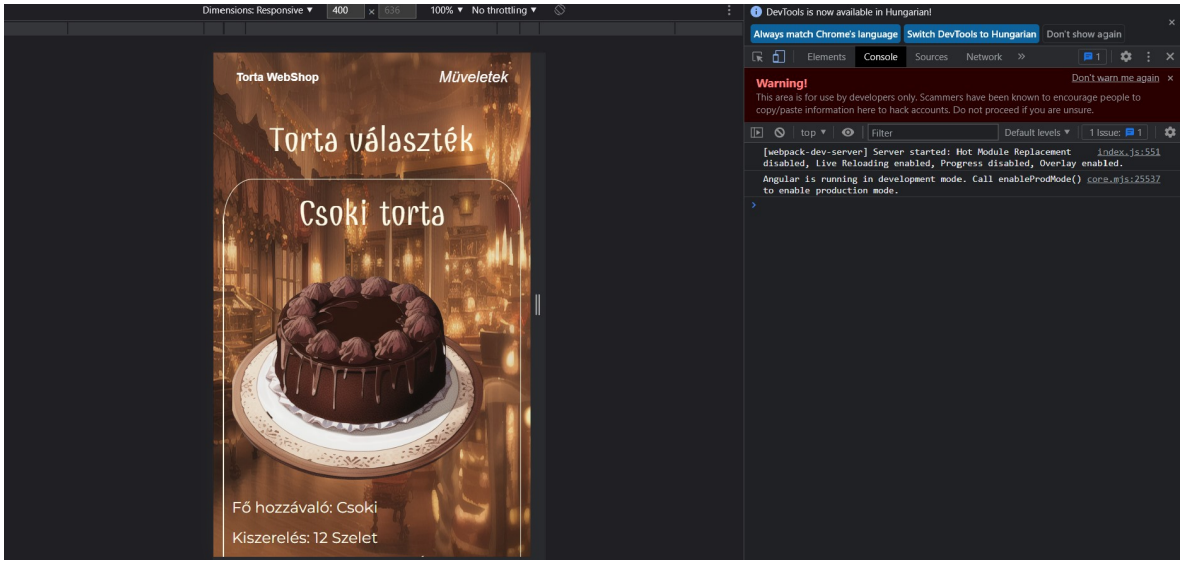
- Ez egy nyíltan elérhető program, azonban van egy fontos követelménye ami annyi hogy egy legalább 4GB Video Memóriával rendelkező videokártya szükséges hozzá.
- Mikor egy AI képet készít, annak szüksége van egy forrásra, ez a forrás lehet egy album madarakról, fákról, városokról, bármiről, mi egy olyan csomagot használtunk amit “Anything”-nek hívnak, ahogy a neve is mondja bármit lehet vele készíteni.
- Rajzfilm stílusú/mesevilág stílusú képeket lehet vele készíteni, és tökéletesen illet a mi tervünkhöz.
- Annyi probléma volt a “művész”-el, hogy szükséges volt egy szoftver ami a felbontást korrigálta, alapesetben maximum egy 512x512-es képet lehet készíteni, ha a weboldalunkhoz illő 1920x1080 pixeles képet szerettünk volna, ahhoz szükség lett volna egy darab egy millió forintos videókártyára, ezért külön ingyenes szoftvereket kellett használnunk hogy a kívánt felbontást elérjük, ezért trükközni kellet vele egy kicsit, és ezeknek az ingyenes programoknak köszönhető hogy a megfelelő felbontásba tudtuk felhasználni ezeket a képeket a projektünkhöz.

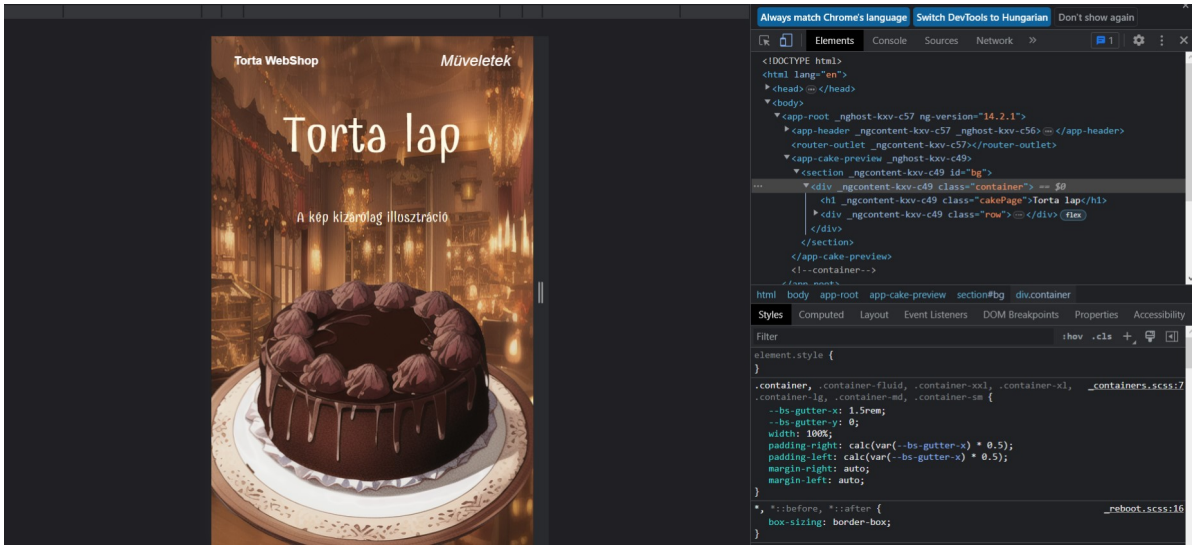


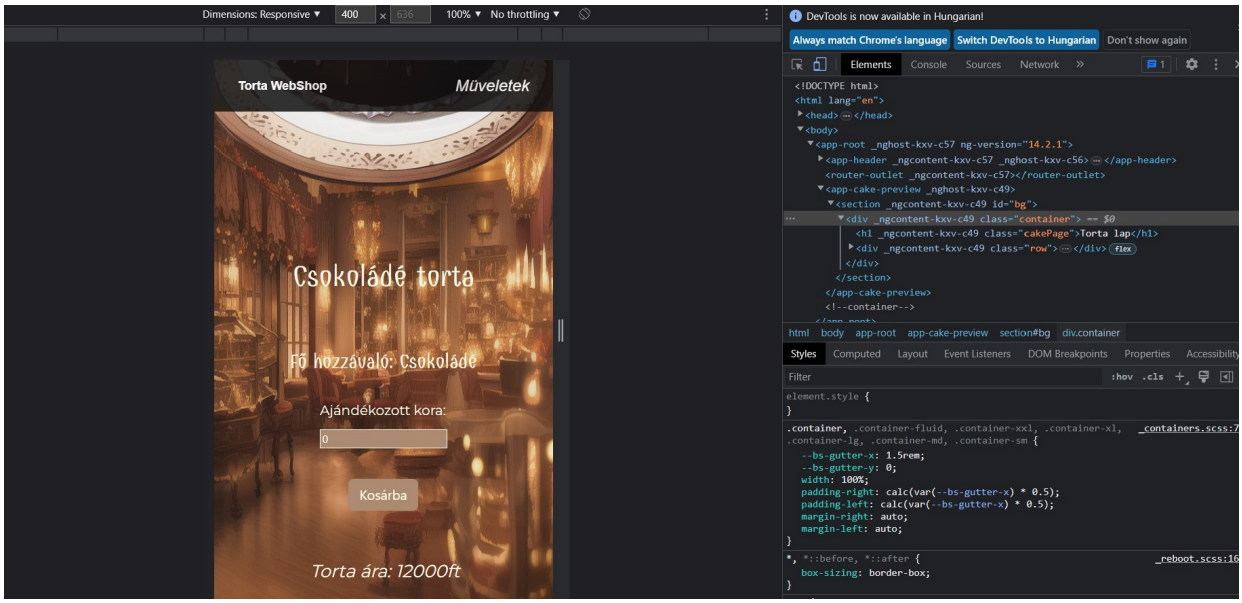
14. Mobil nézet a mobil felhasználók számára, és egyéb más felhasználók számára

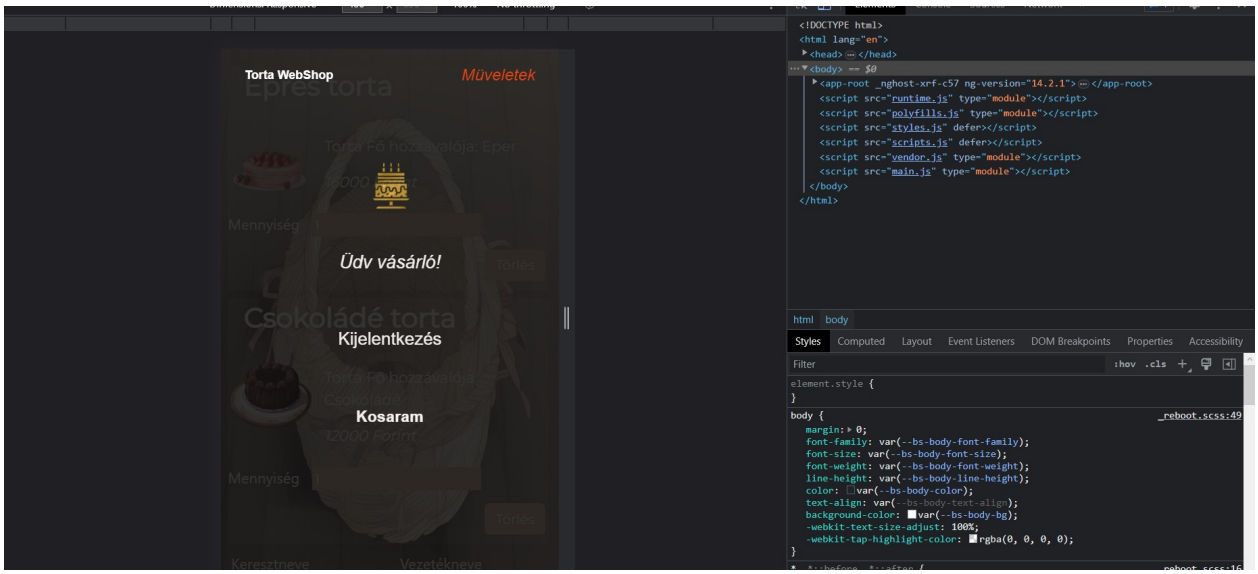
Mobilra, más eszközökre. Próbáltunk egy optimális mobilnézetet találni azok látogatóink számára akik telefonról látogatjuk meg a weboldalunk, azonban van pár probléma ami nem került javításra a nagyon komplex és összetett dizájn miatt.

- A mobil nézet többé-kevésbé tökéletes, azonban a webshop nem tudd megfelelni minden különböző felbontásnak sajnálatosan, aminek annyi az oka hogy a használt bootstrap könyvtár gyakran nem volt elég hatékony, ezért a saját kezünkbe kellett venni az irányítást.
- A weboldal Full-Hd(1920x1080) felbontásra van kitalálva, azonban mint követelmény rendelkezik mobil nézettel is, ezen a kettő nézeten kívül nem tudjuk garantálni a tökéletes felhasználói élményt
- Kompromisszumokat kellett kötni a nagy színvonalú dizájnért cserébe, ami itt az lett hogy nem minden felbontásban tökéletes a weboldalunk, amit nem lenne lehetetlenség javítani azonban mint diákmunka nem voltak ilyen elvárások felénk.






Torta WebShop
Müveletek
Csokoládé torta
Fő hozzávaló: Csokoládé
Ajándékozott kora:
0
Kosárba
Torta ára: 12000ft


Torta WebShop
Müveletek
Üdv vásárló!
Kijelentkezés
Kosaram

15. fogalomtár

Artificial Intelligence - Mesterséges intelligencia
API - Szerver amin a lekérdezések vannak és kommunikál az adatbázissal
User Experience - Felhasználói tapasztalat a termék használatakor Ügyfél orientáció- Ügyfélközpontú, ügyfél elégedettség központú Typescript - JavaScript alapú frontend programozási nyelv
Framework - Egy hatalmas könyvtár valós életbeli projektekhez C# - Microsoft által fejlesztett programozási nyelv
Frontend- A weboldal kliens oldala amit a felhasználó lát
Backend - A weboldal “kábelezése” amit csak a fejlesztő lát, a szerver.
User, Admin - Jogkörök, és különböző engedélyek minden felhasználónak.
URL - Elérési útvonal a böngésző sávjában
Foreign key- Olyan adatbázis tábla amely össze van kapcsolva másikkal
Form- Kitölthető mező webes felületen
True Or False- Igaz hamis értékek, amely alapján a program dönt Update - Egy meglévő “dolog” frissítése, szerkesztése
Bot - Előre beprogramozott robot ami csak egy algoritmus,automata
XAMPP- Egy szoftvergyűjtemény különböző szerveroldali műveletekhez
ORM- Előre beépített adatbázis kezelő könyvtár szerver oldalon Reláció- Táblák kapcsolata egymással,azoknak a kommunikációja.

16. Projekt összefoglalása

Nagyon röviden, egy Spring Boot backenddel és egy Angular frontenddel rendelkező projektet készítettünk el erre a vizsgára, nagyon sok új dolgot próbáltunk ki közben és új eszközökhöz is kellet nyúlnunk, megtanulnunk azért hogy egy épkézláb projektet képesek legyünk összerakni.
A projekt nem releváns egy valós életbeli környezetben mint egy valódi webshop és mikor dolgoztunk is rajta ezzel tisztában voltunk végig, mivel a követelmény kezdetektől fogva nem az volt hogy egy magánszemély részére készítsünk el egy webshopot, egy projekt amiben minden szükséges dolog benne van ami ahhoz kellene, az már professzionális és szakmabeli munka lenne teljesen, nem pedig egy csapatnyi diák munkája.
Tortákat választottunk ki mint témakör a webshophoz, a stílust pedig modern-re és letisztultra terveztük meg azért hogy a lehető legközelebb álljon az elképzelésekhez.

17.Továbbfejlesztési lehetőségek

- Biztonsági funkciók kiterjesztése valós életbeli felhasználása
- Torták hozzáadása és törlése
- Fizetési lehetőségek hozzáadása
- Responzivitás fejlesztése
- Több lehetőség biztosítása a tortalapon pl: egyedi díszítés, nyomtatható ostya fotók
- Formára készített torták
- Kínálat fejlesztése: Sütemények, péksütemények, kávé
- Kedvezmények
- Házhozszállítás
- Élő ügyfélszolgálat