Készítette:

Hézső károly ádám  KOVÁCS TIBOR

tribitrabi Alkatrészek Dokumentáció

Tartalomjegyzék

[I. Feladatspecifikáció 2](#_Toc197885750)

[II. Alkalmazott szoftverek és technológiák 3](#_Toc197885751)

[III. A projektmunka telepítése 7](#_Toc197885752)

[IV. A felület funkciói, felépítése és használata 9](#_Toc197885753)

[V. Api végpontok 15](#_Toc197885754)

## 

## Feladatspecifikáció

A projekt célja egy korszerű, felhasználóbarát autóalkatrész webáruház és adminisztrációs felület létrehozása, amely hatékonyan szolgálja ki a vásárlói és üzleti igényeket. Az autóalkatrészek piaca folyamatosan növekvő trendet mutat, hiszen az autótulajdonosok számára elengedhetetlen a rendszeres karbantartás és az alkatrészek cseréje.

Egy webáruház lehetővé teszi, hogy a vásárlók kényelmesen böngésszenek és rendeljenek alkatrészeket anélkül, hogy személyesen fel kellene keresniük egy üzletet. Az adminisztrációs felület biztosítja a vállalkozás számára a termékek kezelését és az árak beállítását.

A választásunkat tovább erősítette az a piaci igény, hogy az online vásárlási élmény gyors és intuitív legyen. Az általunk alkalmazott technológiák – React a frontendhez, Express a szerveroldali logikához, MySQL az adatbáziskezeléshez és Winform az adminisztrációs felülethez – biztosítják a skálázhatóságot, megbízhatóságot és a könnyű bővíthetőséget.

Ez a projekt nem csupán technológiai kihívás, hanem egy valós üzleti igényekre épülő megoldás, amely hosszú távon fenntartható, innovatív és felhasználóközpontú rendszert biztosít a vásárlók és az üzemeltetők számára.

## Alkalmazott szoftverek és technológiák

2.1 React

Egy népszerű JavaScript könyvtár, amelyet dinamikus és komponens-alapú webalkalmazások fejlesztésére használnak. Gyors és hatékony UI létrehozást tesz lehetővé a virtuális DOM segítségével.

A blue atom symbol with a black background

AI-generated content may be incorrect.

2.2 Express és NodeJs

Az Express egy minimalista és rugalmas Node.js webkeretrendszer, amely gyors szerveroldali alkalmazások fejlesztését teszi lehetővé. Egyszerű útvonalkezelést, middleware támogatást és API fejlesztést biztosít.

A NodeJs egy JavaScript futtatási környezet, amely lehetővé teszi szerveroldali alkalmazások fejlesztését. Nem blokkoló I/O műveleteket használ, így kiváló teljesítményű és skálázható rendszerek fejlesztésére alkalmas.

A logo for a company

AI-generated content may be incorrect.

2.3 MySQL

Egy nyílt forráskódú, relációs adatbázis-kezelő rendszer, amely nagy mennyiségű adat hatékony kezelésére és lekérdezésére alkalmas. Gyakran használják webalkalmazások backendjében.A logo of a database

AI-generated content may be incorrect.

2.4 Bootstrap

Egy népszerű CSS keretrendszer, amely előre definiált stílusokat és komponenseket biztosít, így gyors és reszponzív weboldalak fejlesztését teszi lehetővé.

A white letter on a purple square

AI-generated content may be incorrect.

2.5 WinForms

A Windows Forms egy Microsoft keretrendszer, amely lehetővé teszi asztali alkalmazások fejlesztését a .NET környezetben. Egyszerű UI-elemekkel gyors fejlesztést biztosít.

A purple hexagon with a white background

AI-generated content may be incorrect.

2.6 Visual Studio Code

Egy könnyű és gyors kódszerkesztő, amely támogatja számos programozási nyelvet. Beépített terminál, Git-integráció és bővítmények teszik népszerűvé.

A blue logo with a cross

AI-generated content may be incorrect.

2.7 Visual Studio

Egy professzionális fejlesztői környezet, amely elsősorban .NET és C++ alkalmazásokhoz készült. Robusztus hibakeresési eszközökkel és széleskörű integrációkkal rendelkezik.

A purple ribbon in a shape of a triangle

AI-generated content may be incorrect.

2.8 GitHub

Egy népszerű verziókezelő platform, amely Git alapú projektek kezelésére szolgál. Lehetővé teszi a fejlesztők számára a kód tárolását, közös munkát és verziókövetést, így hatékonyan kezelhetőek a fejlesztési folyamatok.

A black background with a black square

AI-generated content may be incorrect.

2.9 Trello

Egy vizuális projektmenedzsment eszköz, amely kártyaalapú rendszeren keresztül segíti a feladatok nyomon követését és szervezését. Kiváló választás csapatmunkához, mivel könnyen átláthatóvá teszi a fejlesztési és egyéb munkafolyamatokat.

A blue and black logo

AI-generated content may be incorrect.

2.10 Figma

Egy modern tervezési eszköz, amely elsősorban UI/UX dizájnok létrehozására használható. Online együttműködést biztosít, így a csapatok valós időben dolgozhatnak a terveken, gyorsítva a fejlesztési és design iterációkat.

A black background with white spots

AI-generated content may be incorrect.

## A projektmunka telepítése

A legelső lépés hogy a GitHub Repository-t be clone-ozzuk egy újonnan létrehozott mappába a git clone (link) segítségével (cmd-ben):

A screenshot of a computer

AI-generated content may be incorrect.

Utána a Xampp segítségével elindítjuk az Apache és a MySQL modult ezek nélkül nem fog futni a szerver(Az Apache csak akkor kell ha meg akarjuk nyitni az adatbázisunkat a böngészőben)

A screenshot of a computer program

AI-generated content may be incorrect.

Amint ez megtörtént megnyitjuk a backend nevű mappát és nyitunk egy cmd-t. A cmd-be beírjuk a következőt: node server.js. (a pont nem kell a végére) Ezzel elindítjuk az express szerverünket ami az egész projekt alapja.

Ezután megkeressük a a frontend mappát amiben van egy reactFrontend nevű mappa azt megnyitjuk és nyitunk egy cmd-t. A cmd-be beírjuk a következőt: npm run dev. Ezzel elindítottuk a weboldalt. Még itt a cmd ben az o betű megnyomásával meg tudjuk nyitni a böngészőben az oldalt.

## A felület funkciói, felépítése és használata

1. Asztali alkalmazás

Ezen a felületen lehet kezelni az alkatrészeket. Ezt a felületet csak egy előre generált felhasználóval lehet elérni A screenshot of a computer

AI-generated content may be incorrect.

* 1. Alkatrészek

Bejelentkezés után megjelenik a kezelőfelület ahol lehet hozzáadni, törölni és módosítani az alkatrészeket amik majd megjelennek a webáruházban.A screenshot of a computer

AI-generated content may be incorrect.

Itt szükséges megadni az azonosítóját a nevét az árát és a mennyiségét egy adott alkatrésznek. A termék hozzáadása gombra kattintva hozzáadja az alkatrészt az adatbázishoz és ezt egy felugróablak is jelzi. A termék törléséhez ki kell választani az adott sort amit ki szeretnénk törölni ezt a bal oldalon található nyíl segítségével lehet megtenni. Módosításhoz meg kell adni a módosítandó termék id-ját és utána minden adatot akkor is ha valamelyik adat nem újult meg.

* 1. Webes felület

A webes felületen lehet elvégezni a vásárlást. Vásárláshoz mindenféleképpen be kell legyen jelentkezve a felhasználó. A weboldal a főoldalt tölti be először ahol három kép váltakozik és alatta megtalálható négy napi ajánlat amik teljesen véletlenszerűen váltakoznak naponta.A screenshot of a car made of parts

AI-generated content may be incorrect.

Bejelentkezéshez és regisztrációhoz a navigációs sávban található log in gombot kell megnyomni. Regisztráláskor meg kell adni egy felhasználónevet egy jelszót és egy email címet. Ha ezek megvannak és az oldal kiírta hogy sikeres a regisztráció akkor a regisztráció melletti gomb segítségével lehet bejelentkezni. Bejelentkezéskor elég csak a felhasználónevet és jelszót megadni.

A screenshot of a computer

AI-generated content may be incorrect.

Ha ez sikeresen megtörtént és az oldal kiírta hogy sikeres a bejelentkezés akkor a felhasználói profilt fogja megnyitni ahol lehet látni a bejelentkezési adatokat. Itt lehet a jelszót módosítani vagy a profilt törölni.A screenshot of a computer

AI-generated content may be incorrect.

Ezek után a Shop gombot kell megnyomni a navigációs sávban hogy az áruházba kerüljünk. Itt megjelenik az összes alkatrész ami az adatbázisban van. A keresősáv segítségével könnyen meglehet találni azt az alkatrészt amit keresünk. A keresősáv dinamikus tehát ha csak egy betűt is beírunk már csak azok az alkatrészek Vannak listázva amik tartalmazzák azt az adott betűt.

A screenshot of a computer

AI-generated content may be incorrect.

A kártyákon megtalálható gombbal lehet az alkatrészeket a kosárba helyezni.

A Cart gomb segítségével a navigációs sávban eljutunk a kosárhoz ahol látjuk az eddig kosárba rakott termékeket.A screenshot of a computer

AI-generated content may be incorrect.

Itt lehet a mennyiséget növelni vagy a kosárból törölni. A proceed to checkout gomb segítségével eljutunk a ‘kasszához’.A screenshot of a computer

AI-generated content may be incorrect.

Itt meg kell adni a fizetési módot és a szállítási adatokat. Ezek az adatok később megjelennek a rendelés követés oldalon. A place order gomb megnyomásával leadjuk a rendelést.A screenshot of a computer

AI-generated content may be incorrect.

Ha ezt látjuk és ráfrissítünk az oldalra megjelenik a rendeléskövetés gomb a navigációs sávon.A screenshot of a computer

AI-generated content may be incorrect.

Itt látható a rendelés azonosítója a dátuma a fizetési mód és a szállítási cím. Ezt csak az a felhasználó láthatja aki leadta a rendelést

## Api végpontok

**Alkatrészek**

**1. Alkatrészek lekérése**

* **Végpont:** /parts
* **Metódus:** GET
* **Leírás:** Az összes alkatrész lekérése az adatbázisból.
* **Hibakezelés:**
  + Az éppen aktuális hibát kiírja egy felugró ablakban

**2. Új alkatrész hozzáadása**

* **Végpont:** /parts
* **Metódus:** POST
* **Leírás:** Új alkatrész hozzáadása az adatbázishoz.
* **Paraméterek:**
  + name (String, kötelező) – Az alkatrész neve.
  + price (Number, kötelező) – Az alkatrész ára.
  + stock (Number, kötelező) – Az alkatrész raktárkészlete.
* **Hibakezelés:**
  + 400 Bad Request – Ha egy kötelező mező hiányzik.
  + 500 Internal Server Error – Ha a mentés sikertelen.

**3. Felhasználó regisztrációja**

* **Végpont:** /register
* **Metódus:** POST
* **Leírás:** Új felhasználó regisztrációja és mentése az adatbázisba.
* **Paraméterek:**
  + username (String, kötelező) – A felhasználó egyedi neve.
  + password (String, kötelező) – A felhasználó jelszava.
  + email (String) – A felhasználó E-mail címe
* **Hibakezelés:**
  + Ha nem sikerül az adatbázisba berakni az adatokat azt egy felugróablakban jelzi az oldal
  + 403 – Ha a felhasználónév már létezik.
  + 201 Success – Ha a mentés sikeres.

**4. Felhasználó bejelentkezése**

* **Végpont:** /login
* **Metódus:** POST
* **Leírás:** A felhasználó bejelentkezése és JWT token generálása.
* **Paraméterek:**
  + username (String, kötelező) – A felhasználó neve.
  + password (String, kötelező) – A felhasználó jelszava.
* **Hibakezelés:**
  + Ha valamelyik kötelező mező hiányzik.
  + 409 Conflict – Ha a bejelentkezési adatok nem helyesek.
  + Ha a szerverhiba miatt nem lehet bejelentkezni.

**5. Felhasználói profil lekérése**

* **Végpont:** /profile
* **Metódus:** GET
* **Leírás:** A bejelentkezett felhasználó profiljának lekérése.
* **Hibakezelés:**
  + 500  – Ha a felhasználói adatokkal gond van
  + 404 – Ha a felhasználó nem atlálható.

**6. Jelszó módosítása**

* **Végpont:** /profile/password
* **Metódus:** PUT
* **Leírás:** A bejelentkezett felhasználó jelszavának módosítása.
* **Paraméterek:**
  + newPassword (String, kötelező) – Az új jelszó.
* **Hibakezelés:**
  + 400 Bad Request – Ha a jelszó nem felel meg a követelményeknek.
  + 500 Internal Server Error – Ha a módosítás sikertelen.

**7. Felhasználó törlése**

* **Végpont:** /profile
* **Metódus:** DELETE
* **Leírás:** A felhasználó törlése az adatbázisból.
* **Hibakezelés:**
  + 404 – Ha a felhasználó nem található.
  + 500 Internal Server Error – Ha a törlés sikertelen.

**8. Alkatrész törlése**

* **Végpont:** /parts/:id
* **Metódus:** DELETE
* **Leírás:** Egy alkatrész törlése az id alapján.
* **Paraméterek:**
  + id (String, kötelező) – Az alkatrész egyedi azonosítója.
* **Hibakezelés:**
  + 404 Not Found – Ha az alkatrész nem létezik.
  + 500 Internal Server Error – Ha a törlés sikertelen.

**9. Új rendelés létrehozása**

* **Végpont:** /checkout
* **Metódus:** POST
* **Leírás:** Egy új rendelés létrehozása és mentése az adatbázisba.
* **Paraméterek:**
  + paymentData (String, kötelező) – A fizetéshez szükséges adatok.
  + shippingData (String, kötelező) – A szállítási cím és információk.
* **Hibakezelés:**
  + 500 Internal Server Error – Ha a rendelés mentése sikertelen.

**10. Legutóbbi rendelés lekérése**

* **Végpont:** /myorder
* **Metódus:** GET
* **Leírás:** A bejelentkezett felhasználó legutóbbi rendelésének lekérése.
* **Paraméter:** Nincs.
* **Hibakezelés:**
  + 404 – Ha a felhasználónak nincs rendelése.
  + 500 Internal Server Error – Ha a rendelés lekérése sikertelen.

**11. Alkatrész módosítása**

* **Végpont:** /parts/:id
* **Metódus:** PUT
* **Leírás:** Egy meglévő alkatrész adatainak (név, ár, készlet) frissítése az adatbázisban.
* **Paraméterek:**
  + id (Integer, kötelező) – A módosítandó alkatrész egyedi azonosítója (URL paraméterként).
  + name (String, kötelező) – Az új alkatrész neve.
  + price (Integer, kötelező) – Az új ár.
  + stock (Integer, kötelező) – Az új raktárkészlet.
* **Hibakezelés:**
  + 404 Not Found – Ha a megadott azonosító nem létezik az adatbázisban.
  + 500 Internal Server Error – Ha a módosítás során szerverhiba lép fel.
  + 200 OK – Sikeres módosítás esetén visszajelzést küld az új adatokkal.