# RideHub Dokumentáció

A RideHub egy olyan szolgáltatás, amely jelenlegi pillanatban Békés illetve Hajdú-Bihar vármegye lakosainak könnyű és egyszerű utazási módot nyújt.

Szeretjük az egyszerű, letisztult és minimalista designt, ezért a webalkalmazásunkat is ilyen formában igyekeztünk személyre szabni. Emellett szerettünk volna felhasználóbaráttá formázni. Nem tettünk bele figyelemelterelő elemeket, igyekeztünk a lényeget a főképbe helyezni.

Azért ezt a témát választottuk mert a vezetés nyugalmat, élvezetet illetve kikapcsolódást nyújt és ha valakinek az a munkája amit szeret, az már igazából szinte nem is munkának tekinthető, hanem szórakozásnak.

Ezért kitaláltuk, hogy miközben úgymond szórakozunk, segítünk az embereken azzal kapcsolatban hogy egyszerűen , gyorsan és hatékonyan , de ami még ennél is fontosabb, biztonságban érjék el úti céljaikat.

Az ötlet az UBER szolgáltatásból merült fel, ami hazánkban még nem igazán terjedt el, főleg a vidéki területeken. Igyekszünk több árkategóriát és minőséget biztosítani. Ha valaki gyorsabb vagy igényesebb megjelenést szeretne, vannak drágább és jobb kategóriás autók, viszont akinek nem fontos a megjelenés, vagy csak szimplán szeretne inkább kevesebbet költekezni nálunk erre is van lehetőség.

# Fejlesztői dokumentáció

### Fejlesztői környezet

Fejlesztői környezetek amiket mi használtunk:

* Visual Studio Code
* Trae

Hardver:

Egy Lenovo ThinkCentre M80s típusú számítógépet használtunk a fejlesztés nagy részében, amely a következő specifikációkkal rendelkezik:

* Processzor: Intel Core I5-10400
* Memória: 8GB

Az alábbi operációs rendszereken fejlesztettük a webalkalmazást:

* Windows 10 pro
* Windows 11 Pro

Futtatási környezet

* Node.js
* PHP

Verziókezelés

A verziókezeléshez Git-et használtunk, a kódot pedig GitHub-on tároljuk.

### Keretrendszer

A webalkalmazásunkat Laravel és Inertia keretrendszerben fejlesztettük. Ez egy FullStack keretrendszer, ami lehetővé teszi a Frontend és Backend egyidejű fejlesztését a kódbázisban. Azért választottuk ezt a keretrendszert mert erről hallottunk a legtöbbet és úgy gondoltuk ez javunkra válhat.

### Adatbázis

A projectünkhöz SQLite-ot használtunk az adatok tárolására. Az adatbázisban 6 tábla található, amelyek itt találhatók:

🏎️ cars – Autók

Oszlop Típus Leírás

id tinyint(4) Egyedi azonosító

img varchar(26) Kép neve vagy URL

cartype varchar(20) Autó típusa (pl. sedan)

price mediumint(9) Ár

license\_plate varchar(7) Rendszám

status tinyint(4) Állapot (pl. elérhető)

color varchar(5) Szín

👨‍✈️ drivers – Sofőrök

Oszlop Típus Leírás

id tinyint(4) Egyedi azonosító

picture varchar(0) Profilkép (nem definiált hosszal)

pictureP varchar(13) Jogosítvány képe

name varchar(16) Sofőr neve

description varchar(1) Rövid leírás

status tinyint(4) Elérhetőség, státusz

💬 feedbacks – Visszajelzések

Oszlop Típus Leírás

id varchar(0) Egyedi azonosító (hibás típus!)

user\_id varchar(0) Felhasználóra mutató hivatkozás

rating varchar(0) Értékelés

comment varchar(0) Szöveges megjegyzés

📦 orders – Rendelések

Oszlop Típus Leírás

id tinyint(4) Egyedi azonosító

from varchar(16) Kiindulási hely

to varchar(10) Célállomás

car\_id tinyint(4) Autóra való hivatkozás

price varchar(6) Ár

driver\_id tinyint(4) Sofőr azonosítója

start int(11) Indulási idő (timestamp)

end int(11) Érkezési idő (timestamp)

duration tinyint(4) Időtartam (percben?)

status tinyint(4) Rendelés állapota

🏙️ settlements – Települések

Oszlop Típus Leírás

id smallint(6) Egyedi azonosító

name varchar(21) Település neve

👤 users – Felhasználók

Oszlop Típus Leírás

id tinyint(4) Egyedi azonosító

name varchar(16) Név

role varchar(6) Szerepkör (pl. admin)

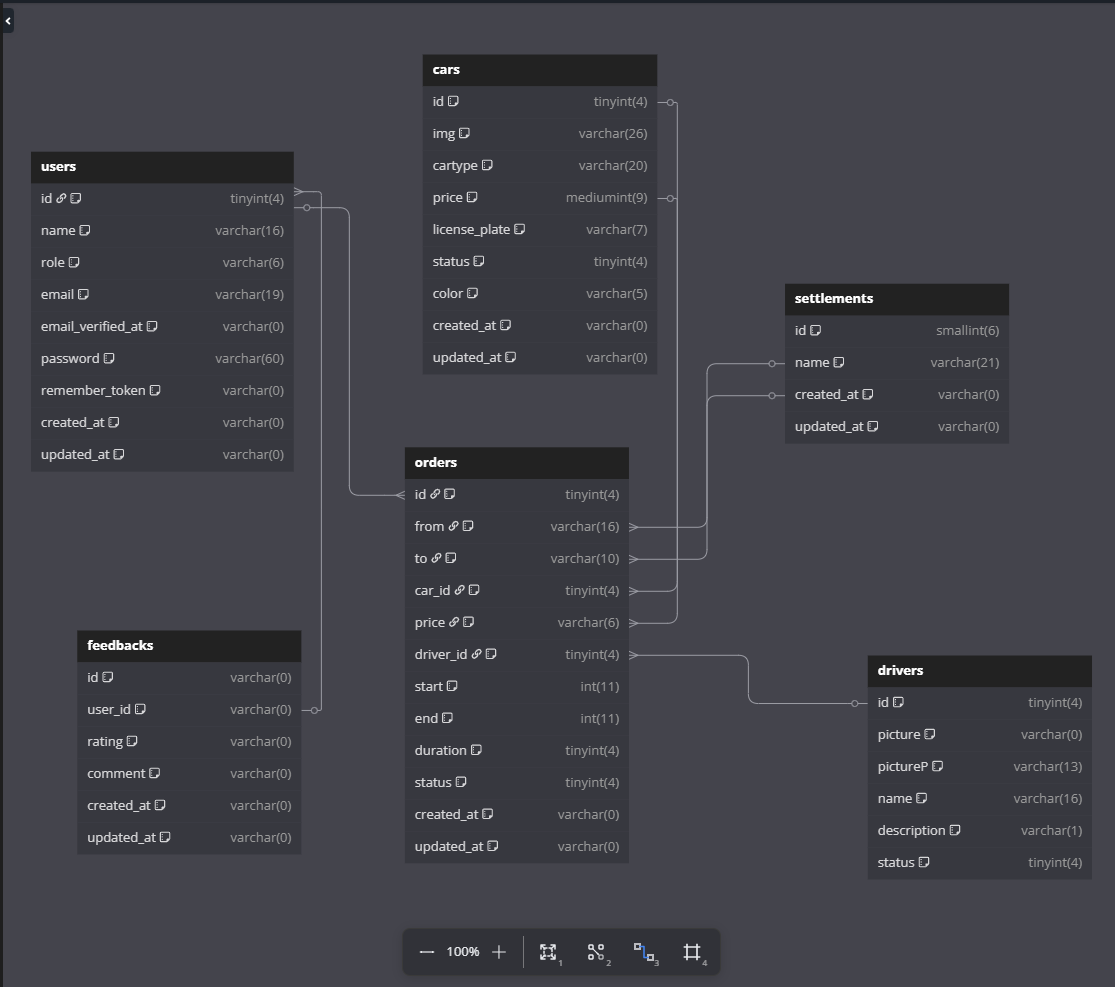
email varchar(19) Email cím

email\_verified\_at varchar(0) Email megerősítés ideje

password varchar(60) Jelszó (hash)

remember\_token varchar(0) Token

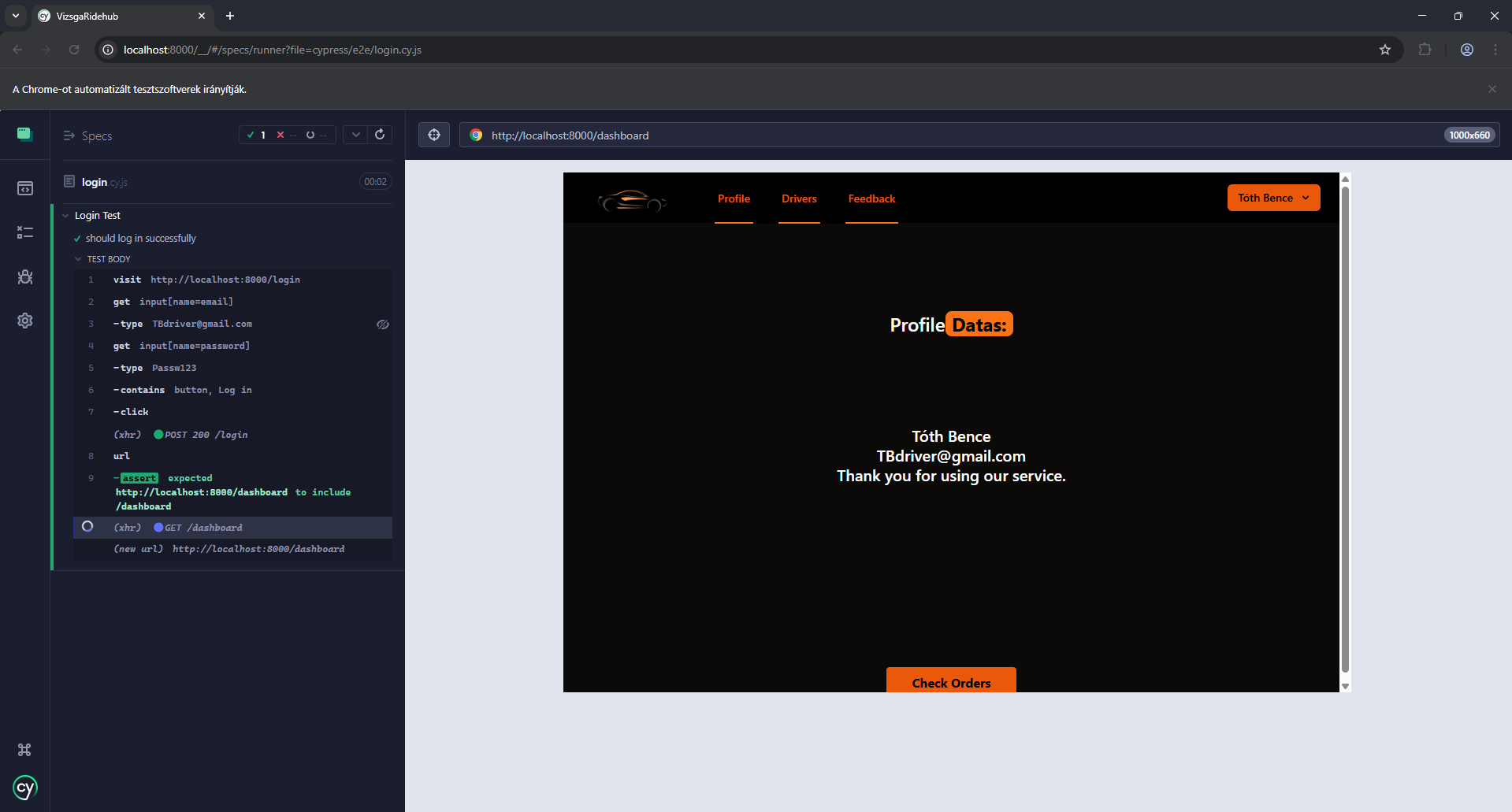
Az adatbázist az alább látható adatbázis terv alapján készítettük el.



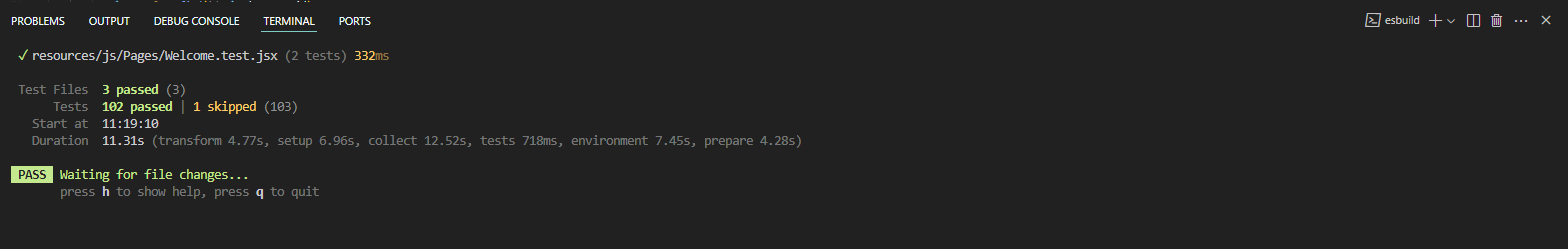
### Tesztelés

Vitest, React Testing Library illetve Cypress

Egy példa a Cypress tesztelésre:



Vitest tesztelés:



### Tailwind CSS

A projektünkben a Tailwind CSS-t használjuk a stílus kezeléséhez. A Tailwind egy utility-first CSS keretrendszer, amivel gyorsan és konzisztensen lehet a felhasználói felületeket felépíteni előre definiált osztályokkal.

### Cél és Funkciók

Ahogy a bevezetésben is megemlítettük, a RideHubot egy gyors és hatékony fuvarszolgáltatásként szeretnénk alkalmazni. Olyan emberek számára szeretnénk biztonságos közlekedést biztosítani, akik nem rendelkeznek autóval vagy nem szeretnének tömegközlekedéssel közlekedni és szeretnének a lehető leghamarabb az úti céljaikba érni.

A funkciók egyszerűek és könnyen észrevehetők. Viszont jelenleg csak angolul elérhető, úgyhogy egy minimális angol tudást igényel, de áttekinthető a rendelési felület ezért aki nem érti a nyelvet az is könnyen eligazodhat rajta.

* A sofőrök és az autók oldala megtekinthető bejelentkezés nélkül is
* A rendelési felület, a visszajelzés és a fuvarok csak bejelentkezés után megjeleníthető
* Lehetőség van szöveges és csillagos értékelést leadni
* A rendelés csak akkor adható le ha az indulási és érkezési városok, a sofőr illetve az autó is ki van választva
* A leadott fuvarokat a profil felületen lehet megtekinteni, de kezelni őket (elindítani és lezárni) csak az admin joggal rendelkező sofőrök tudják

### Szükséges eszközök

Az alkalmazás eléréséhez szükség van egy olyan eszközre, amely rendelkezik operációs rendszerrel, legyen ez telefon, táblagép, notebook, laptop, számítógép vagy akár okos televízió is. Minden készüléken teljes mértékben optimalizált és reszponzív az alkalmazásunk.

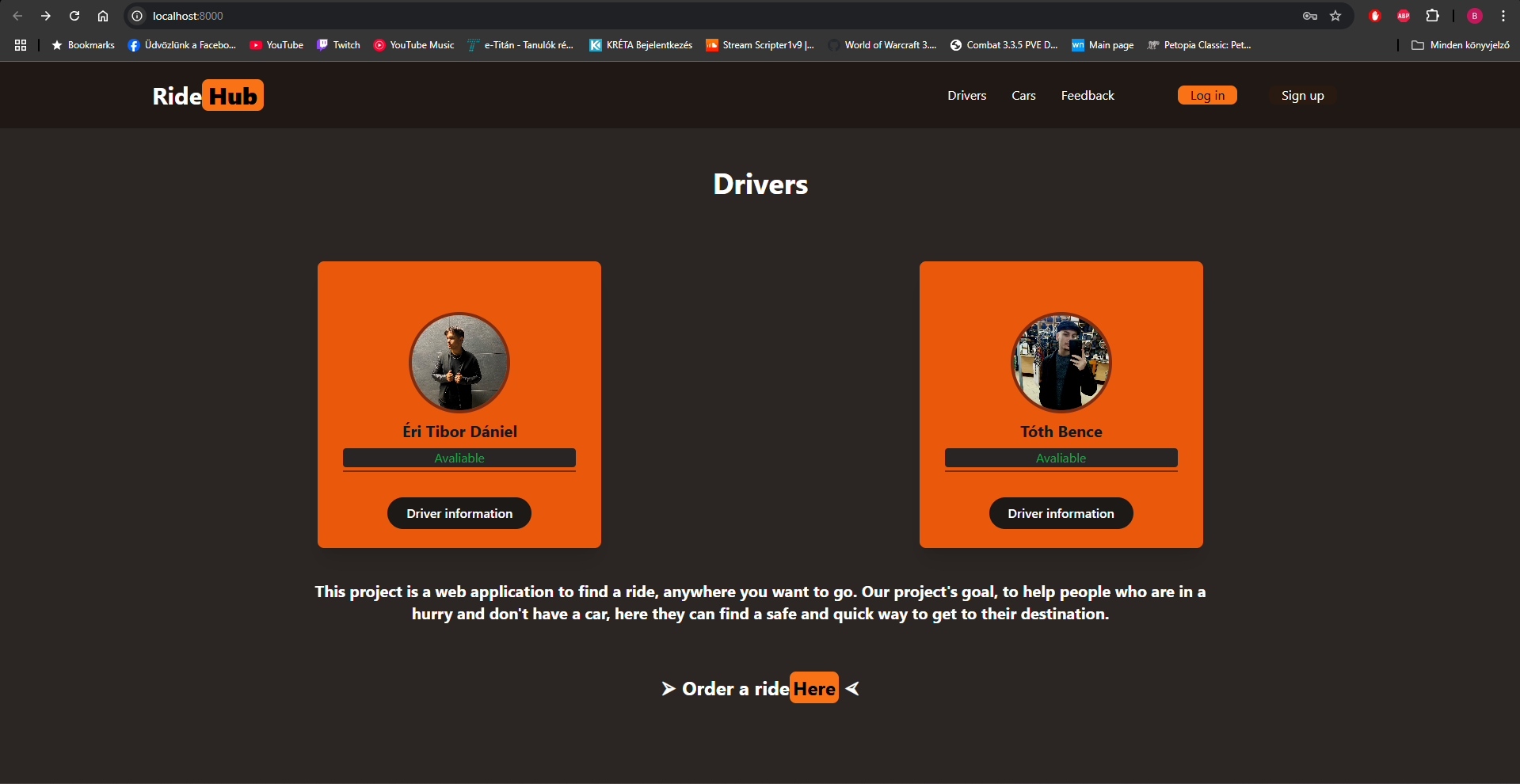
Illetve szükség van egy webböngészőre is ahol meg tudjuk nyitni az alkalmazást.

#### Böngészők ahol teszteltük az alkalmazást

* Google Chrome
* Arc
* Microsoft Edge
* Mozilla Firefox
* Opera

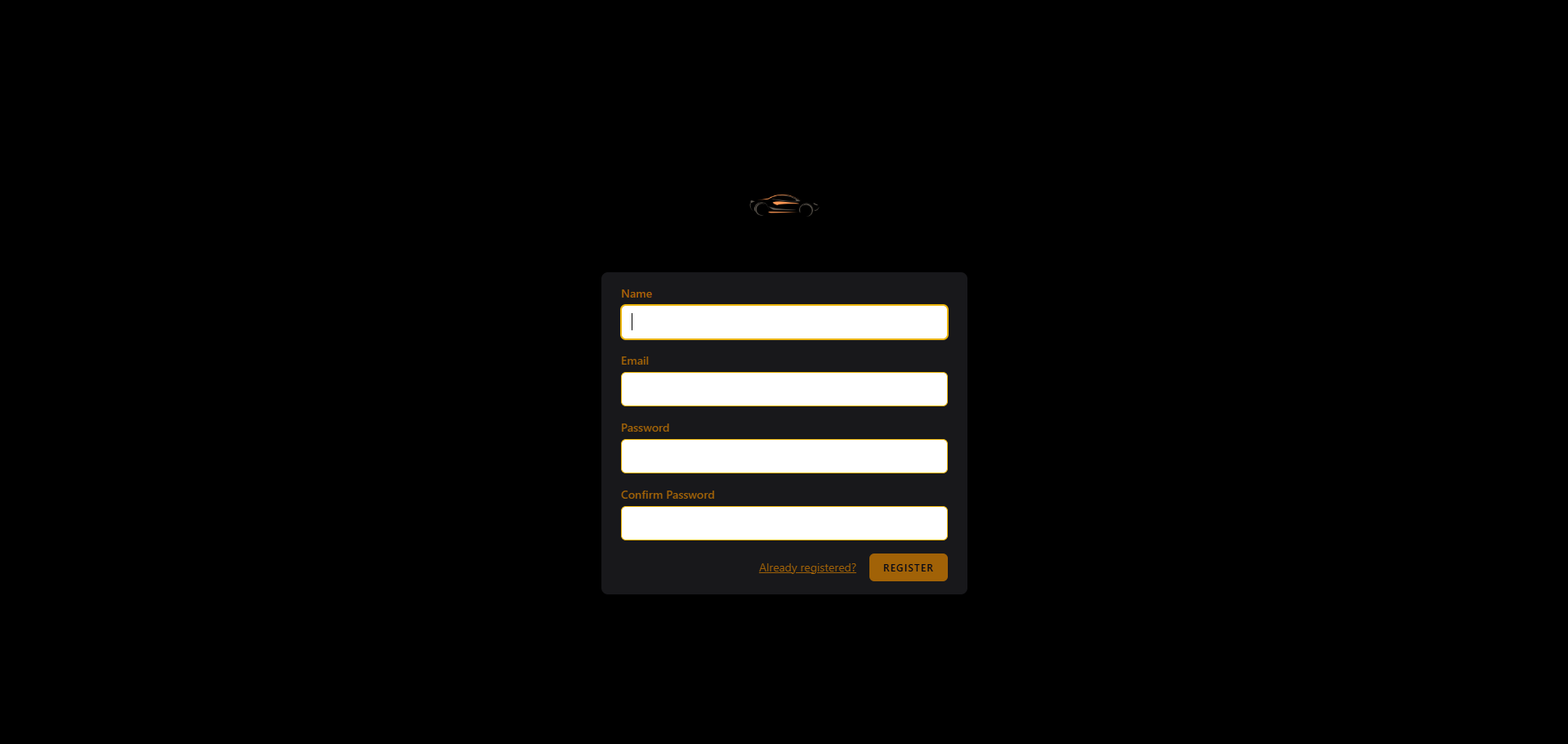
## Felhasználói leírás

A főoldalra lépve egy rövid leírást láthatunk az alkalmazásról és a sofőröket figyelhetjük meg



A regisztáció gombra kattintva ezt a felületet kapjuk

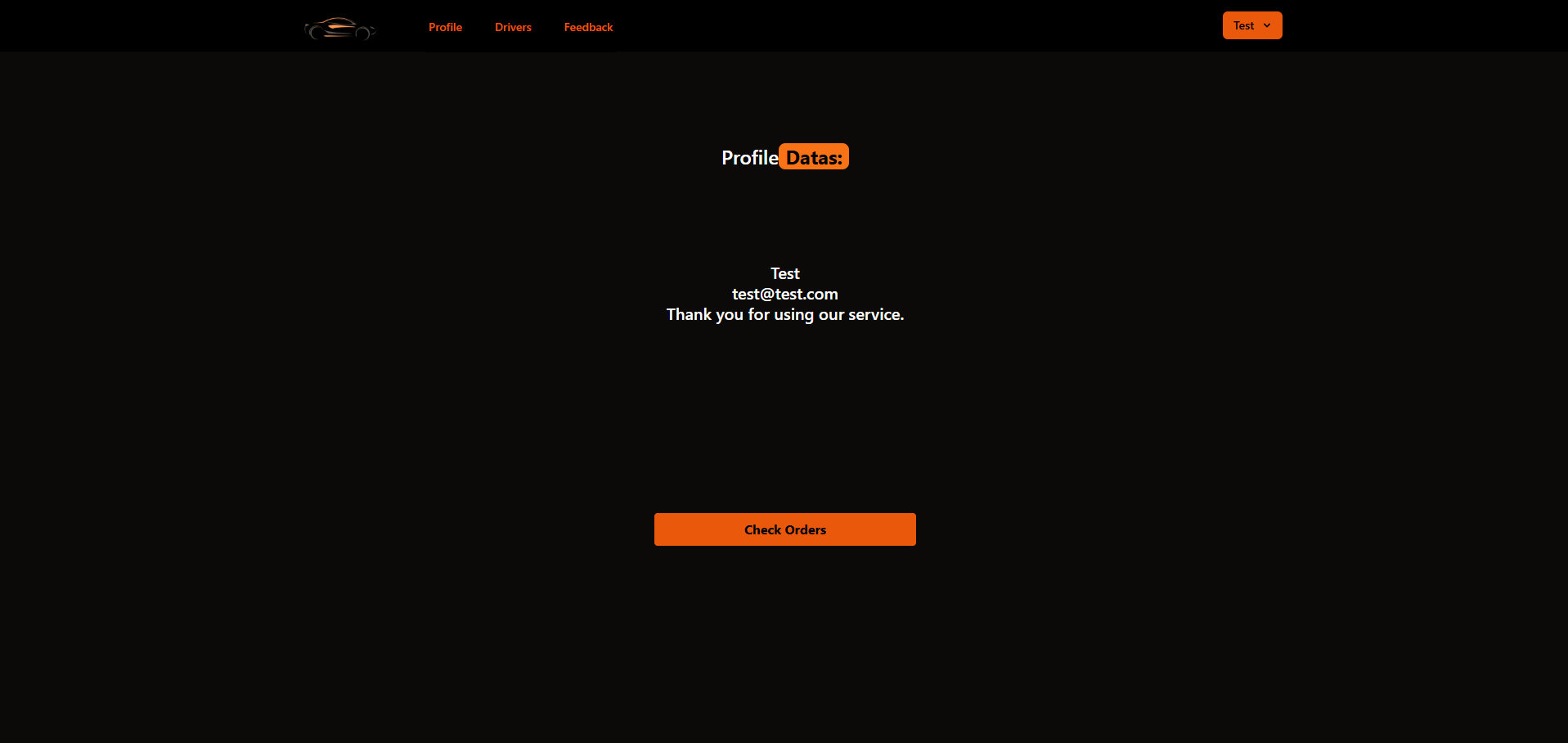
A regisztáció könnyen elvégezhető, ha mégsem szeretnénk regisztálni az autó cionra kattintva vissza térhetünk a főőoldalra.



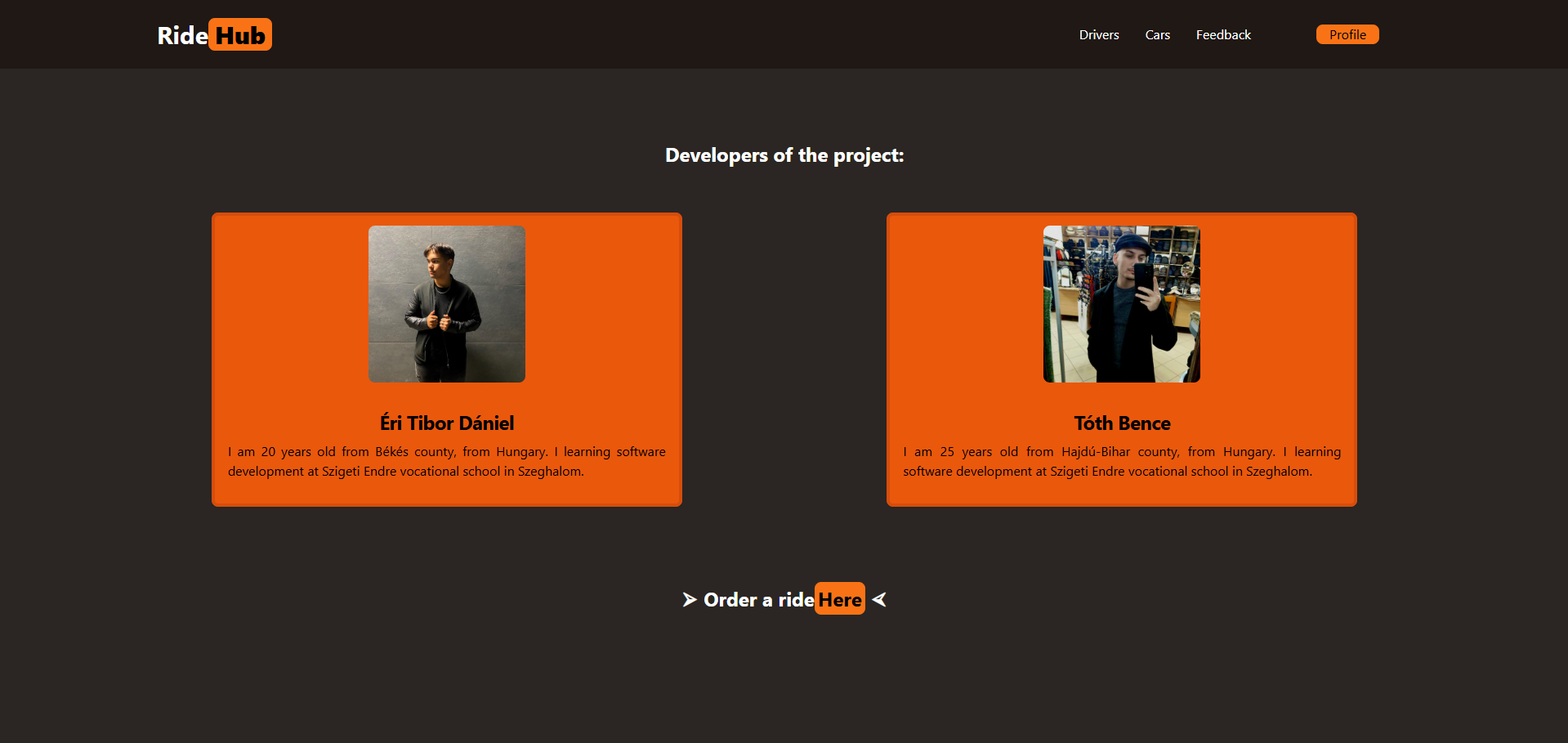
A regisztráció után bejelentkezésre már nincs szükség automatikusan a profilra dob, innen vagy az autó iconra a főoldalért vagy a többi lehetőségre kattintva navigálódhatunk el.

[Kép helye: Képernyőkép 2025-04-15 092556.png]

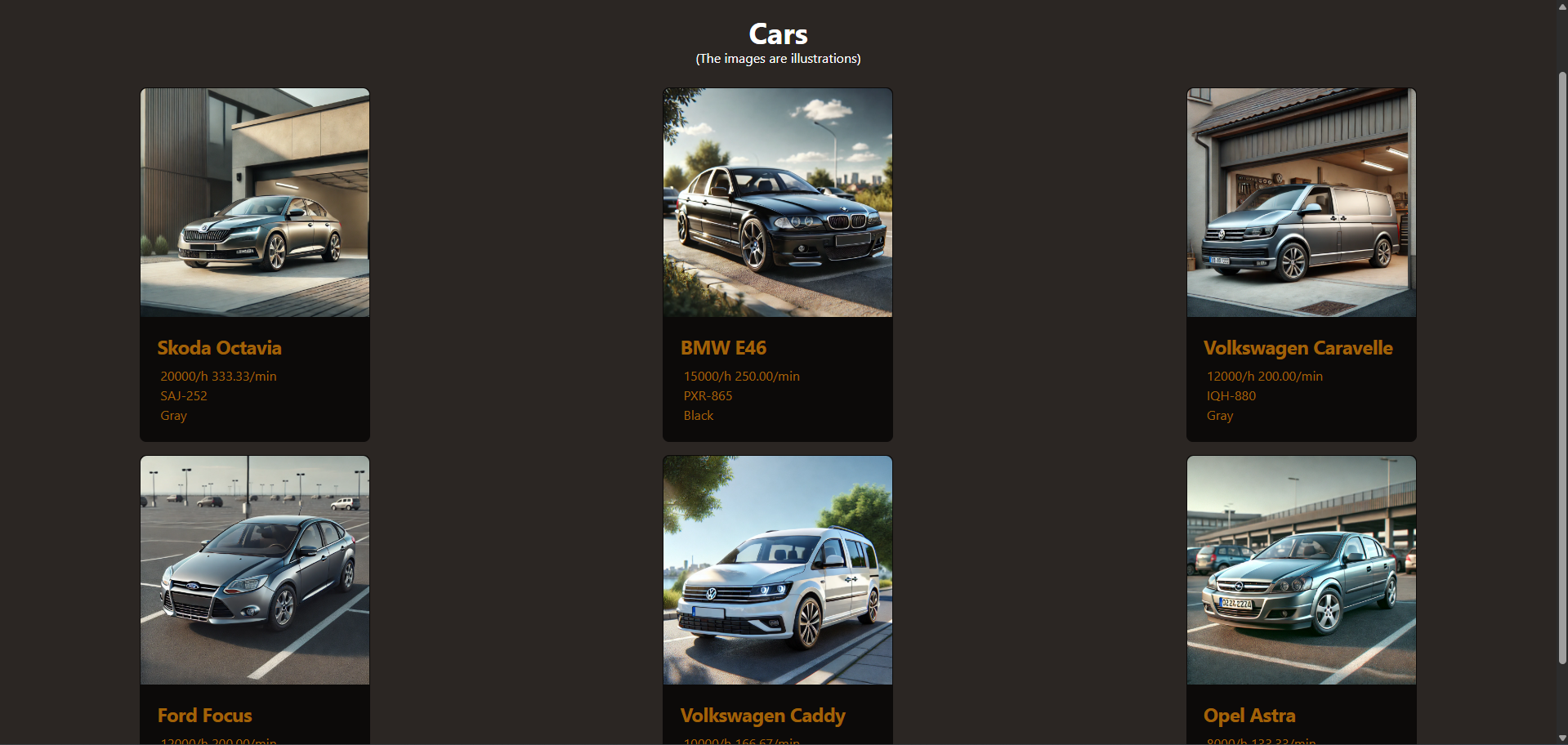
A főoldalra visszatérve nézzük meg mit láthatunk a többi menüpontban



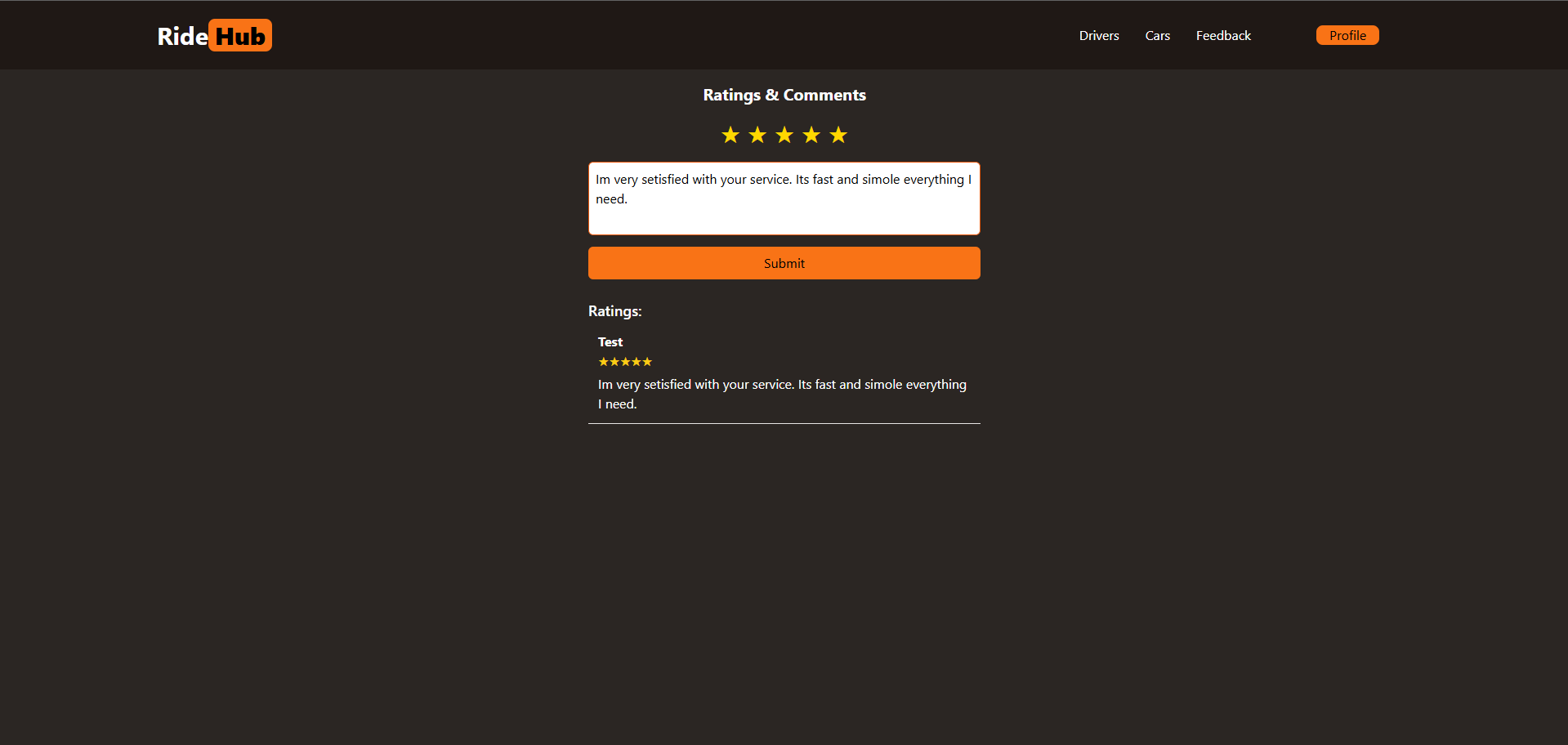
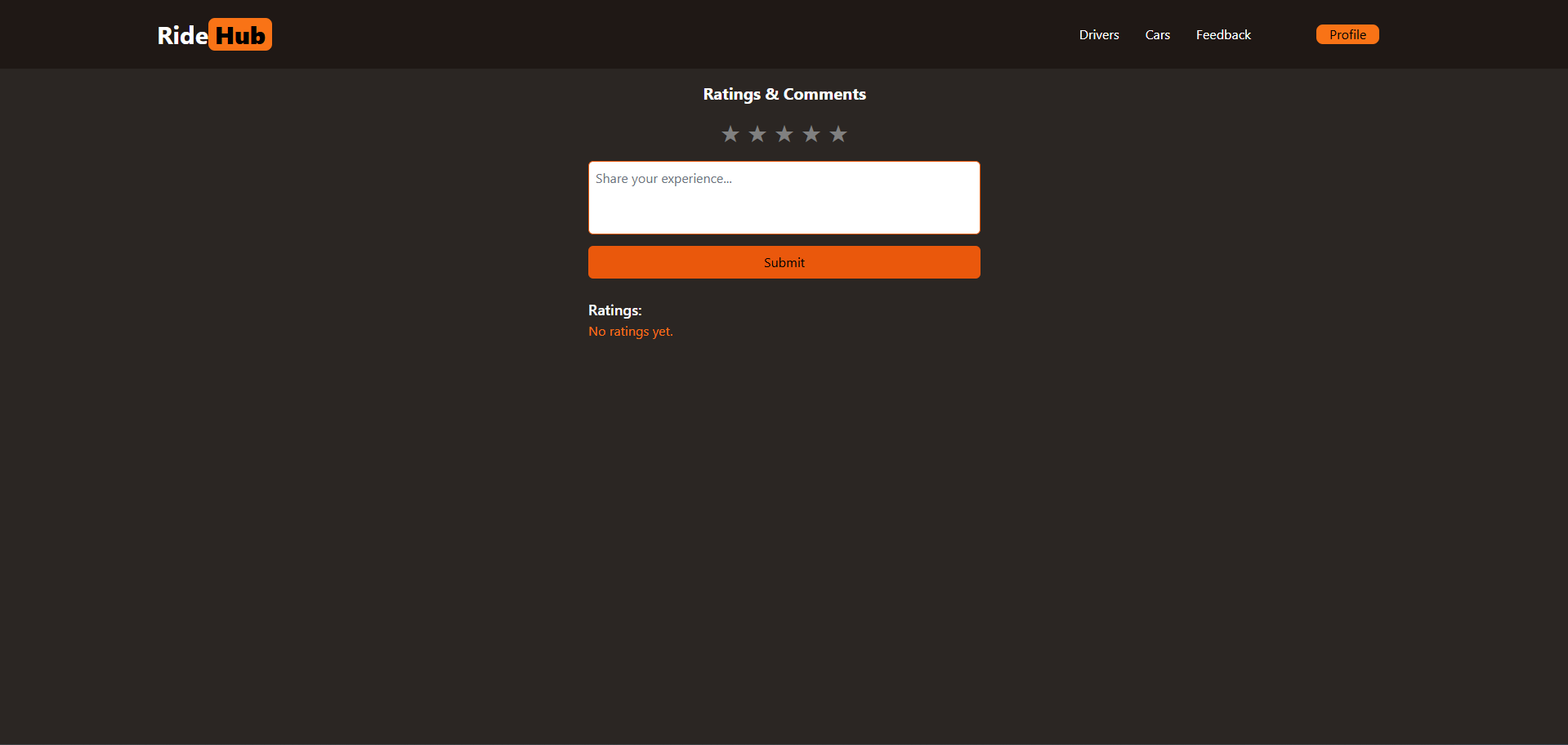
A "Drivers"-nél a sofőrökről tudhatunk meg több információt



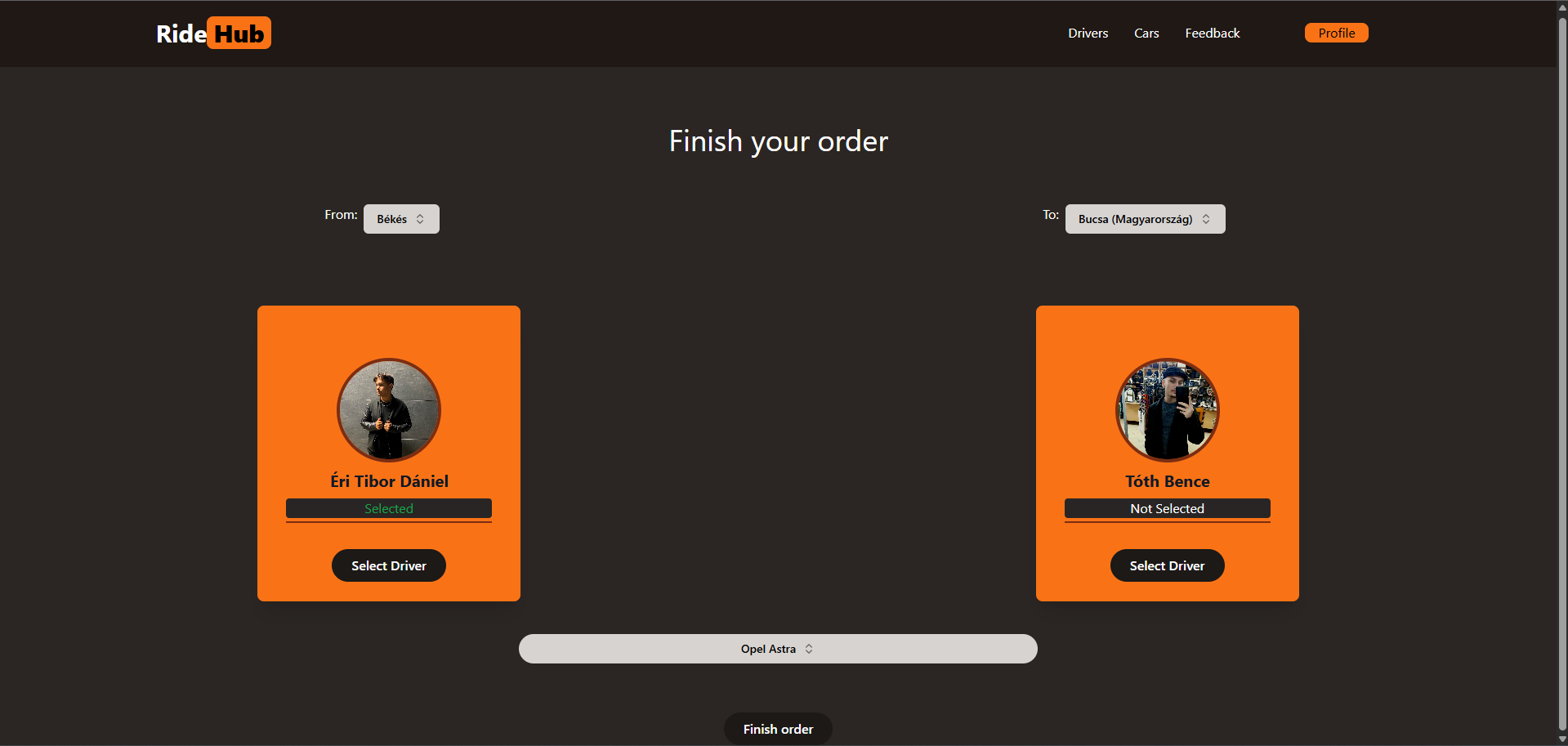
A "Cars" menüpont alatt a cégünkben működő járművekről tudhatunk meg többet pl. Ár, Szín valamint hogy milyen modell.



A "Feedback"-nél vissza jelzést küldhetünk és olvashatunk az alkalmazásról



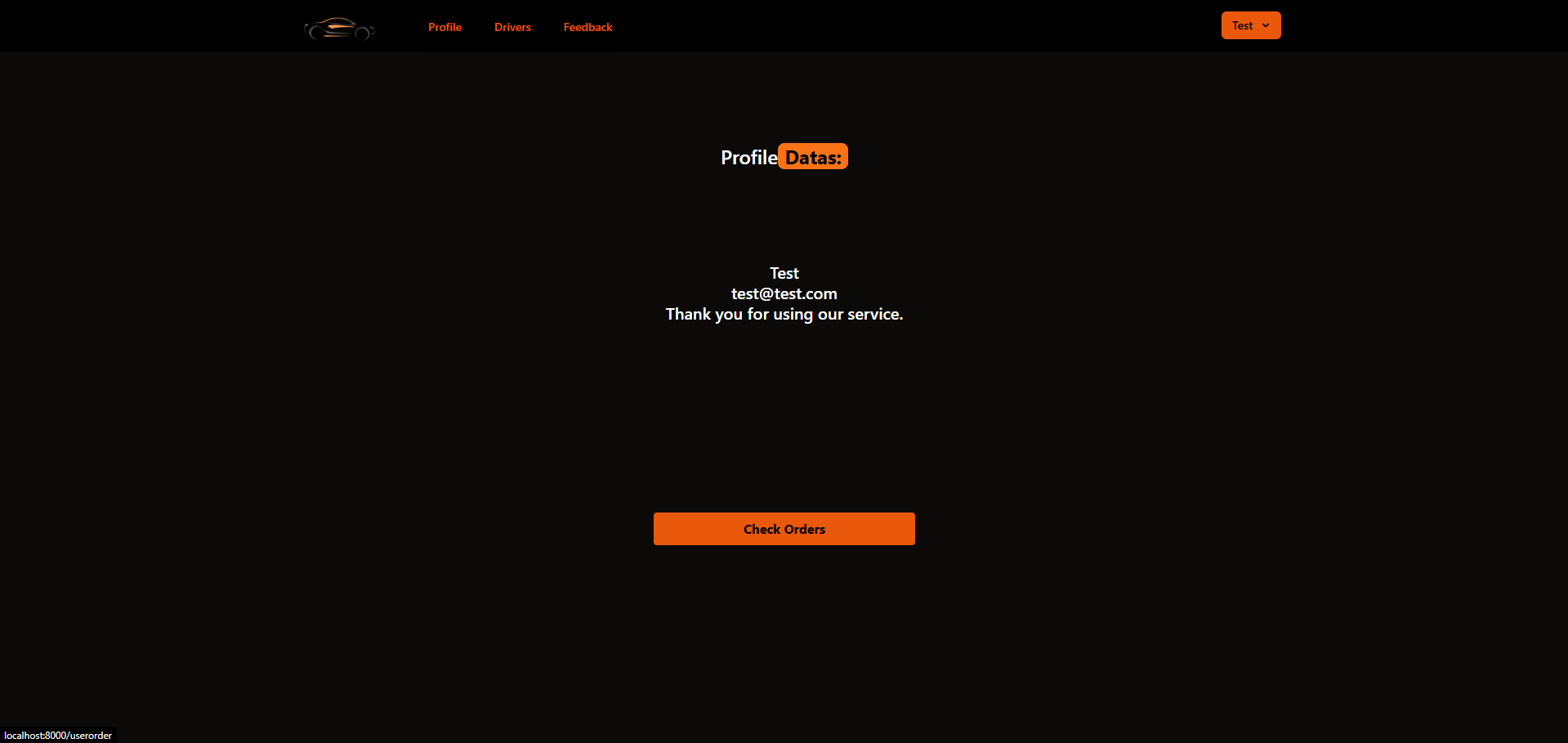
Majd a rendelési felület amely a főoldalról és a "drivers" oldalról érhetunk el a "Order a ride here" gomb megnyomásával érhetünk el

Itt választhatjuk ki azt a 4 dolgot ami a rendeléshez szükséges:

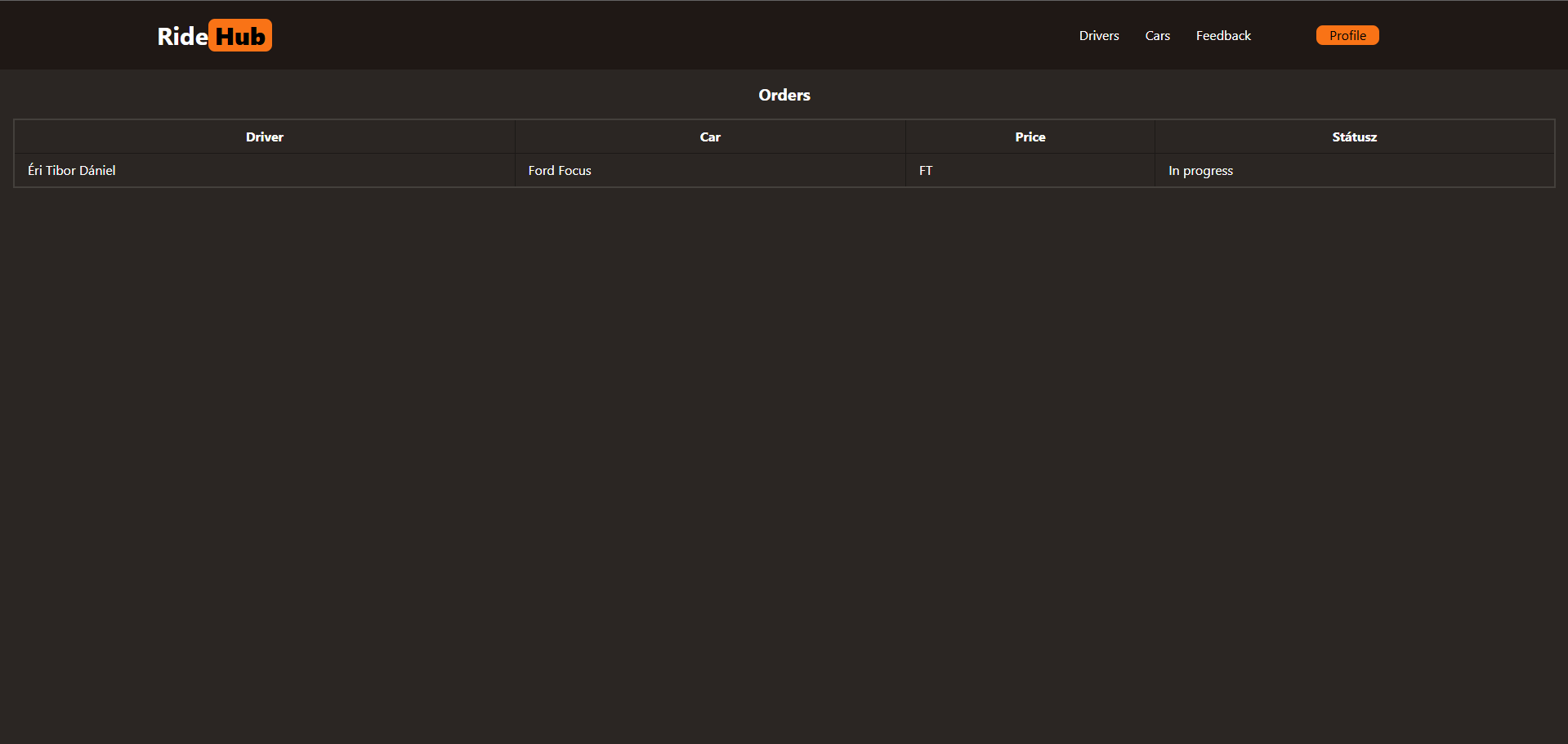
* Az a települést ahonnan indulni szeretnénk
* Azt a települést ahová menni szeretnénk
* A sofőrt akivel menni szeretnénk
* És az autót.

A "Finish Order" gombal elküldhetjük a rendelés.

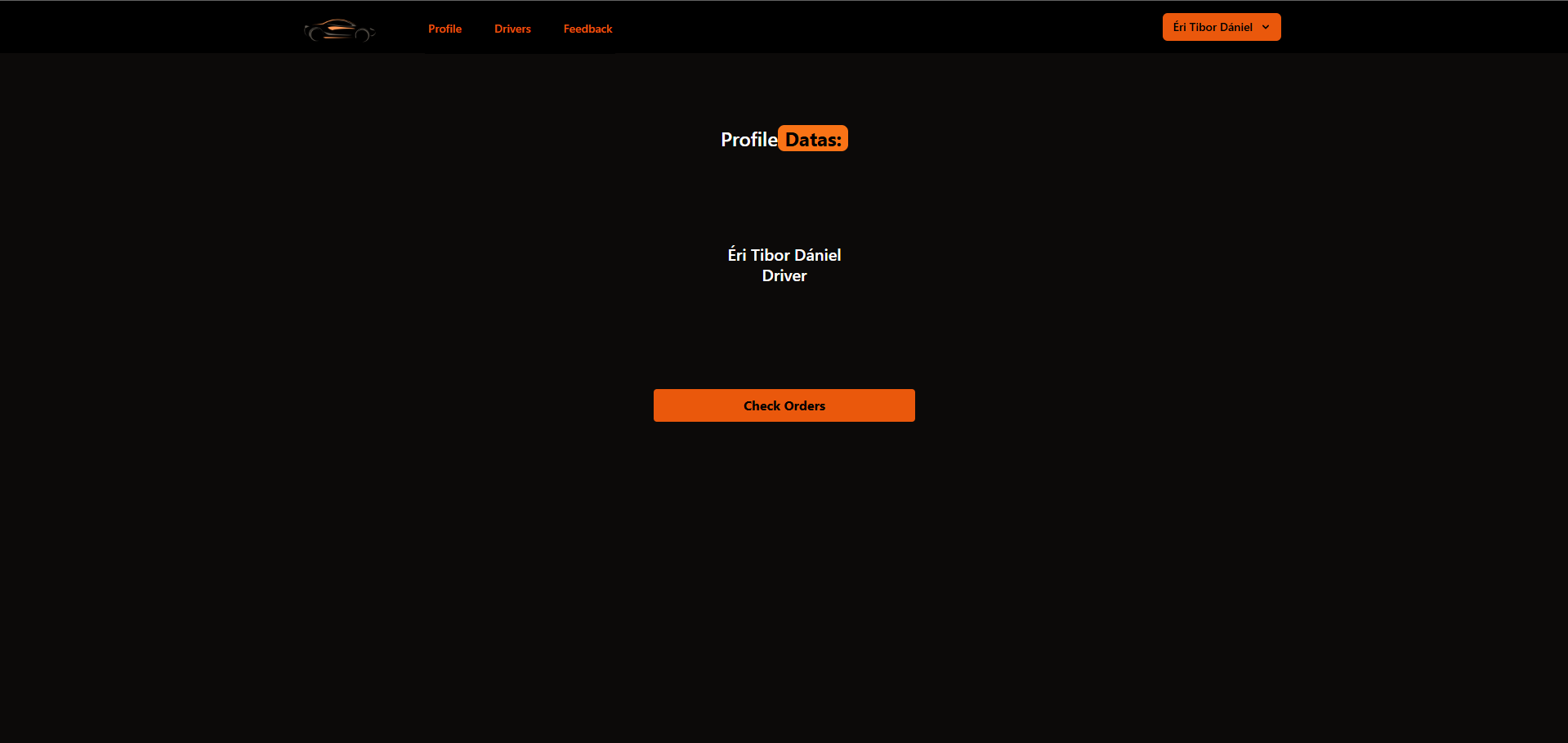
A profil oldalra visszamenve kattintsunk a check orders gombra



A gombra kattintva láthatjuk a rendeléseket

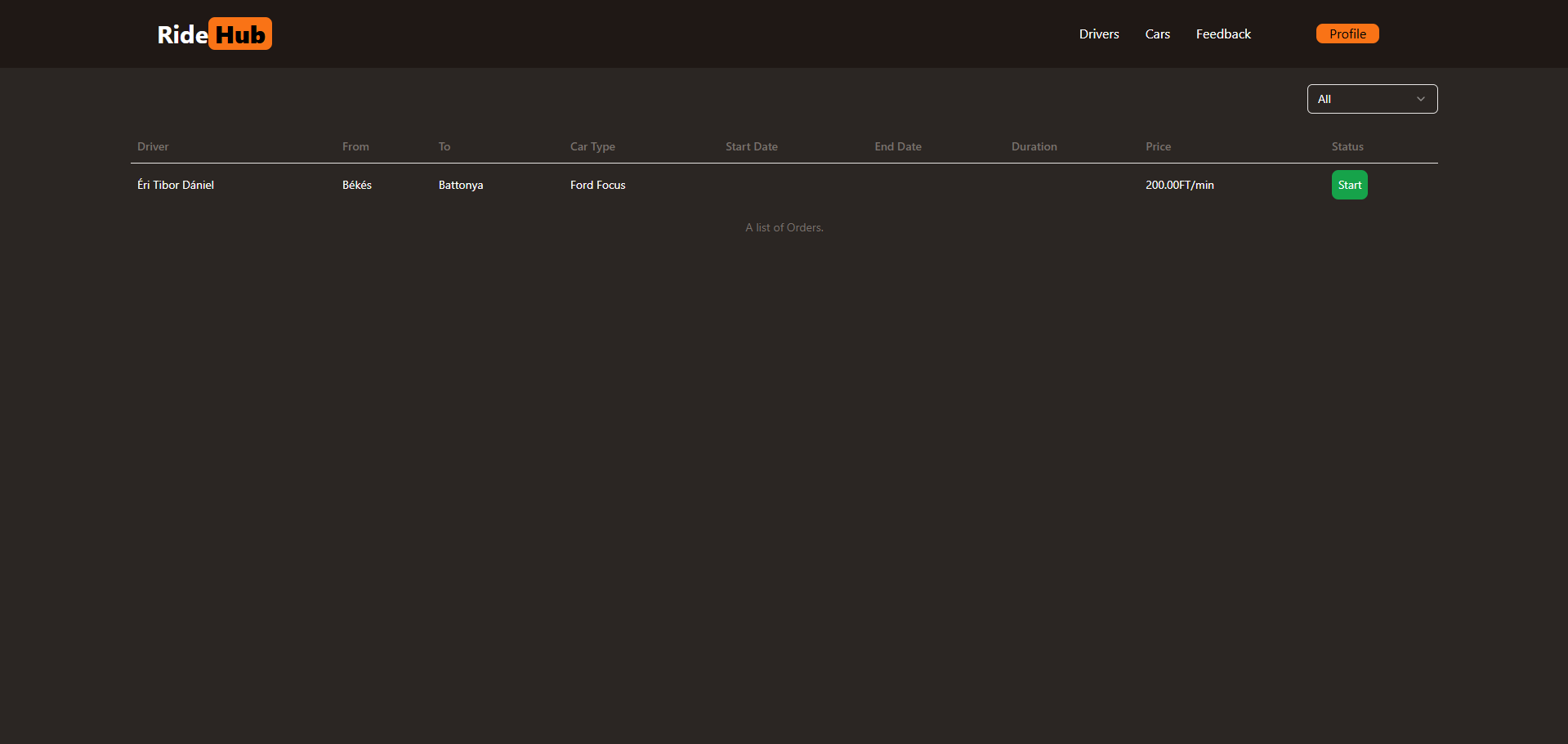


A sofőr oldalról:

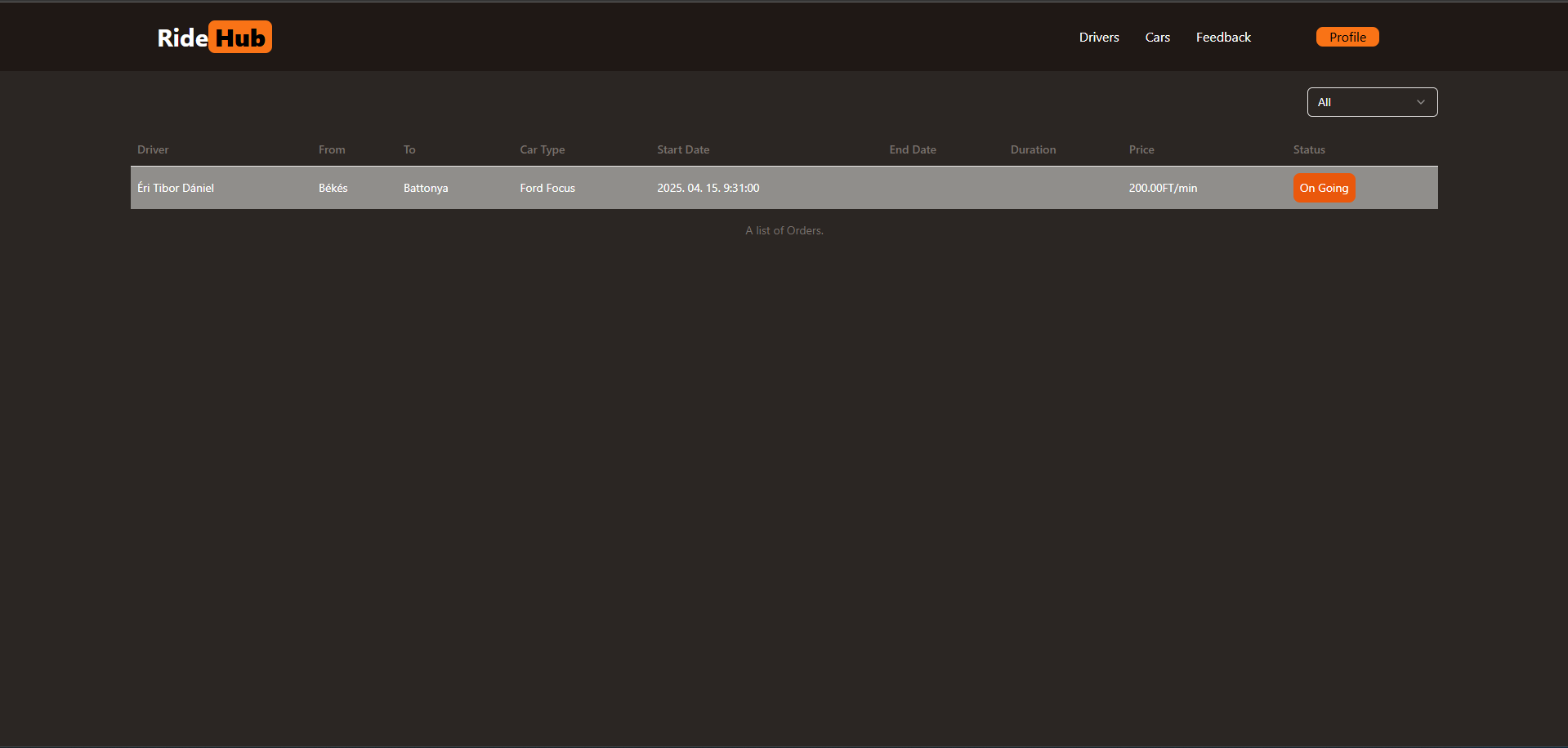


A "Check Orders" gombra kattintva láthatjuk a beérkezett rendeléseket

A Start gombra kattintva jelezzük hogy a fuvar elkezdődött



Még nem végeztünk ezt látjuk majd, Az "On Goingra" majd a "Finish"-re kattintva végzünk a fuvarral.



Ha végeztünk láthatjuk az árat is amely az időből és az autó árából számolódik ki

