

## **Feladat: Autó bérlet webes applikáció**

### **1. szerepről (felhasználó, sofőr, főnök)**

#### **főnök:**

- látja a rendeléseket, tudja azokat törölni
- látja a sofőröket, tudja őket törölni, hozzáadni és ideiglenesen kivenni (szabadság)
- látja a járműveket, tudja törölni és hozzáadni

#### **sofőr:**

- látja a saját feladatait

#### **felhasználó:**

- regisztrálhasson
- látja az éppen aktuális sofőröket és járműveket
- tudja szerkeszteni a profilját
- tudja töltölni a saját rendeléseit

### **2. USE-case funkció lista**

#### **Regisztráció**

POST: /registration

bemenő paraméter: username, password, email

kimenő paraméter: "Success message", "Error message"

#### **felhasználó adatainak megadása**

POST: /save/id

bemenő paraméter: teljes név, telefonszám, jogosítvány kategória, jogosítvány lejárata

kimenő paraméter: "Success alert", "Error alert"

#### **felhasználó adatainak szerkesztése**

POST: /save/id

bemenő paraméter: teljes név, telefonszám, jogosítvány kategória, jogosítvány lejárata

kimenő paraméter: "Success alert", "Error alert"

### **Sofőrök lekérdezése**

GET: /drivers

bemenő paraméter: null

kimenő paraméter: sofőrök listázva

### **Járművek lekérdezése**

GET: /products

bemenő paraméter: null

kimenő paraméter: járművek listázva

### **Bérlés**

POST: /Rent

bemenő paraméter: id, kezdeti-vég dátum, választott autó/sofőr

kimenő paraméter: "Succes alert", "Error alert"

### **Bérlés törlése**

DELETE: /rentDelete/RentID

bemenő paraméter: null

kimenő paraméter: "Succes alert", "Error alert"

### **Sofőr saját munkáinak lekérdezése**

GET: /getDriverRents/driverId

bemenő paraméter: null

kimenő paraméter: rendelések listázva

### **Főnök látja a rendeléseket**

GET: /AdminRents

bemenő paraméter: null

kimenő paraméter: rendelések listázva

### **Rendelések törlése**

DELETE: /rentDelete/RentID

bemenő paraméter: null

kimenő paraméter: "Succes alert","Error alert"

### **Főnök látja a járműveket**

GET: /AdminVehicles

bemenő paraméter: null

kimenő paraméter: járművek listázva

### **Járművek törlése**

DELETE: /vehicleDelete/vehicleId

bemenő paraméter: null

kimenő paraméter: "Succes alert","Error alert"

### **Járművek hozzáadása**

POST: /addNewCar

bemenő paraméter: márka, model, évjárat, alváz szám, ár/nap, üzemanyag típus, rendszám, szín

kimenő paraméter: "Succes alert","Error alert"

### **Főnök látja a sofőröket**

GET: /AdminDrivers

bemenő paraméter: null

kimenő paraméter: sofőrök listázva

### **Sofőrök törlése**

DELETE: /driverDelete/driverId

bemenő paraméter: null

kimenő paraméter: "Succes alert","Error alert"

### **Sofőr hozzáadása**

POST: /addNewCar

bemenő paraméter: név, nem, jogosítvány kat. , username, password, email

kimenő paraméter: "Succes alert", "Error alert"

### **3.Magas szintű rendszerterv**

backend: node

frontend: react

adatbázis: mySQL

böngésző követelmény: legfrissebb Chrome, Firefox, Safari

### **4. SRCEENEK**

-külön mellékelve leírással

### **5. ADATBÁZIS MODELLJE**

costumer {

id (pk)(int),

user\_id (fk)(int),

Fullname (varchar),

License\_category (varchar),

License\_expiration (text),

Phone (varchar)

}

```
drivers {
id (pk)(int),
  user_id (fk)(int),
  name (varchar),
  sex (varchar),
  license_category (varchar),
  available (tinyint)
}

products {
id (pk)(int),
  brand (varchar),
  model (varchar),
  chassisNumber (text),
  plateNumber (text),
  fuel (varchar),
  color (varchar),
  rentprice (int),
  year (int),
  image (longtext)
}

rate {
id(pk)(int),
  user_id (fk)(int),
  car_id (fk)(int),
  car_rate (int),
  driver_id (fk)(int),
  driver_rate (int)
}
```

```
rent {  
  id(pk)(int),  
  user_rent_id (fk)(int),  
  start_date (date),  
  end_date (date),  
  car_id (fk)(int),  
  driver_id (fk)(int)  
}  
  
users {  
  id (pk)(int),  
  user_id (int),  
  username (varchar),  
  password (varchar),  
  email(vvarchar)  
}
```

## **6. Alkalmazások kiválasztása**

Visual Studio Code (node, react), XAMPP(MySQL)

## 7.ROUTING (szöveges leírása)

app.post("/registration") adatok begépelése után, gombra kattintás és ellenőrzést követően, egy JSON formátumu üzenetet kapunk vissza

app.post("/save/id") profil oldalon adatok megadása után és gombnyomást követően egy alert-et kapunk vissza (JSON formátumu szöveg)

app.get("/drivers") menüpontra kattintásnál lekérdezzük a sofőröket és azokat listázva látjuk majd

app.get("/products") menüpontra kattintásnál lekérdezzük a járműveket és azokat listázva látjuk majd (JSON formátumu)

app.post("/Rent") adatok megadása után, gombnyomást és ellenőrzést követően az adatok felkerülnek adatbázisba és egy alertet kapunk vissza (JSON formátumu szöveg)

app.delete("/rentDelete/RentID") felhasználó a saját profilján törölheti gombnyomásra a rendeléseit, alertet kapunk vissza (JSON formátumu szöveg)

app.get("/getDriverRents/driverId") sofőrök saját munkáinak lekérdezése, JSON formátumu tömb jön vissza

app.get("/AdminRents") admin oldalon az összes bérlés lekérdezése, JSON formátumu tömb jön vissza

app.delete("/rentDelete/RentID") gombnyomásra törölük az adott rendelést, egy alertet kapunk vissza

app.get("/AdminVehicles") admin oldalon az összes jármű lekérdezése, JSON formátumu tömb jön vissza

app.delete("/vehicleDelete/vehicleId") gombnyomásra törölük az adott járművet, egy alertet kapunk vissza

app.post("/addNewCar") adatok beírása és gombnyomást követően felküldjük adatbázisba az adatokat és egy alertet kapunk vissza

app.get("/AdminDrivers") admin oldalon az összes sofőr lekérdezése, JSON formátumu tömb jön vissza

app.delete("/driverDelete/driverId") gombnyomásra törölük az adott sofőrt, egy alertet kapunk vissza

app.post("/addNewDriver") adatok beírása és gombnyomást követően felküldjük adatbázisba az adatokat és egy alertet kapunk vissza

## 8.Middleware(algoritmusok)

nincs middleware

## **9.Implementáció**

Forráskód mellékelve

## **10.Tesztelés**

Backend - Postman tesztek (mellékelve)

React - tesztek a projekten belül a Teszt mappában

## **11.Végfelhasználói dokumentáció**

Felhasználói dokumentáció néven mellékelve