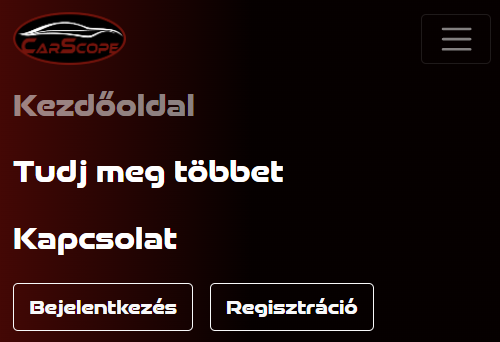
**Frontend/Design dokumentáció**

A weboldal html5, a weblapok tartalmának és megjelenésének leírására szolgáló szabványos programozási nyelven lett írva. Az oldalak formázása és kinézete .css fájlokkal és a bootstrap különböző osztályaival van megvalósítva.

Különböző megjelenítésekhez és validálásokhoz pedig a vue.js keretrendszer van alkalmazva. A public/js mappában találhatók azok a fájlok melyek ezeket a metódusokat és függvényeket tartalmazzák.

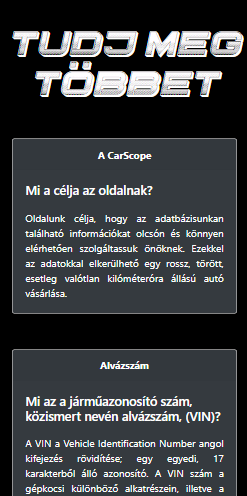
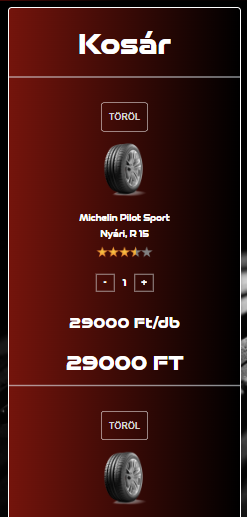
Az oldal reszponzivitása bizonyos helyeken bootstrappel van megvalósítva, mint például a navigációs sáv is.

**Mobilnézet**

A tudj meg többet/termékek felületen a kártyák elhelyezése is a bootstrap columns osztályjelölőjével lett formázva. Ennek a segítségével különböző felbontásokra be lehet állítani, hogyan ossza fel az adott elemeket az adott képernyőméreten. Összesen 12 oszlopra van felosztva a képernyő és különböző képernyő méreteken meglehet adni, hogy egy adott elem hány oszlopot foglalhat el ebből a 12 oszlopból.  
<div class="col-12 col-md-6 col-lg-4 col-xl-3 col-xxl-2">

A mobilnézetnél az addig használt „inline” stílusok átlettek alakítva vertikális irányú stílusokká, azzal a céllal, hogy a felhasználónak mobiltelefonon is könnyen olvashatóak legyenek a szövegek, valamint a gombok linkek és lenyíló listák kezelése is sokkal kényelmesebb így mobil méretű felbontásokon.

Bizonyos megoldások éppen ezért nagyobb felbontásokon elérhetőek csak, mint például a kalkulátor oldalon található 180°-kal elforduló kép.



Mobilnézeten és egyéb felbontásokon a megfelelő megjelenítéshez a .css-fájlokban a @media („felbontás”)-al van formázva. Ezzel megadhat egy médialekérdezést és egy CSS-blokkot, amelyek csak akkor vonatkoznak a dokumentumra, ha a médialekérdezés megadott felbontása megegyezik az oldal pillanatnyi felbontásával, amelyen a tartalmat használják. Paraméternek egy minimum és maximum értéket is meg lehet adni egyszerre.

@media (min-width: 941px) and (max-width: 1200px) {

    .fizetesgomb {

        margin-left: 30px;

    }

}

A kosárnál a fizetés véglegesítése és az adatkezelési tájékoztató is a bootstrap modal ablakjával lett megvalósítva. A véglegesítéskor a felhasználó által megadott adatok megfelelését az input mezőben elhelyezett „pattern” is ellenőrzi, ha nem megfelelő a beírt adat a mező színe pirosra vált.

pattern="[ a-zA-ZÀ-ÿ\u00f1\u00d1\u0171\u0170\u0150 ^-,]\*"



**Főoldal/.js fájlok**

A főoldalon a kettő lenyíló listát egy „watcher” vue.js szolgáltatás ellenőrzi, ami csak abban az esetben futtatja a hozzá tartozó függvényt, ha a figyelt elem tulajdonsága megváltozik.

watch: {},

Jelen esetben a „select” mezőbe elhelyezett „v-modell” direktíva segítségével tudjuk a lenyíló lista választott átadni a figyelő függvénynek, valamint biztosítani a két irányú adatközlést. mezőjét

<select v-model="selected" id="cusSelectbox" name="options">

A már előre deklarált változók itt a feltételnek megfelelően értéket kapnak, amit utána a vue.js „v-if” direktívájával megvizsgálunk.

selected(value)

      {

        if(value == "R")

        {

          this.valasztott = "R";

          this.country\_valasztott = "";

        }

        if(value == "A")

        {

          this.valasztott = "A";

          this.country\_valasztott = "";

        }

      },

Ennek megfelelően jelennek meg vagy tűnnek el például az országok zászlói vagy jelenik meg egy újabb lenyíló lista.

<div v-if="valasztott == 'R' &&   
country\_valasztott != ''" class="info">

A „v-if” direktíva egy feltétellel rendelkező elem megjelenítési, CSS tulajdonságának átváltására szolgál. Ha a feltétel igaz, akkor láthatóvá teszi, máskülönben láthatatlanná teszi azt.

A vizsgálni kívánt tartományra hivatkozni kell a „mount „paranccsal, amit vagy id-hez, class-hoz vagy html elemhez tudunk csatolni.

app.mount('#cars')

**Termékek/rendelés**

A termékek felületen egy termék hozzáadásakor, egy bootstrap modal jelenik meg, melyre egy autómata bezáró függvény van meghívva, amit a „setTimeout” függvény segítségével időzíthetünk.

function timer(obj){

    setTimeout(function() {

        $('#exampleModal').modal('hide');

    }, 800);

};

A rendelés leadásakor pedig a felugró ablakon a bekért mezők értékét a megadott „pattern”-en kívül a rendelés.js fájlban található függvények is ellenőrzik.

watch: {

    postalcode(value) {

      if (!/\D/.test(value) == true && value.length == 4 && value.replace(' ','') != '') {

        this.irsz\_validate = true;

      } else {

        this.irsz\_validate = false;

      }

    },

},

Itt szintén a „watcher” figyelő segítségével nézzük meg a bevitt mező értékét valamint, hogy az megfelelően lett -e kitöltve.

A kosárnál a rendelés véglegesítésénél pedig az általános szerződési feltételek ablak megnyitása után, ha az elfogad gombra kattintunk akkor a rendelés „modal”-on levő „checkbox” autómatikusan el lesz fogadva a „gomb elfogad” metódus hatására.

methods: {

    gombelfogad() {

      this.isChecked = 'checked'

      this.afsz\_validate = true;

    }

  },