0

WEB Alkalmazás

Fejlesztői napló

eCommerceSyncBundle | 2024. 03. 03.

Tartalomjegyzék

* [Technikai információ.](#asd) 2.-3.

* [Header rész.](#header) 4.
* [Csomaginformációs oldal.](#cso) 5.-6.
* [Adatbázisinformációs oldal.](#db) 6.-8.

* [Csapatinformációs oldal.](#csap) 8.-10.
* [Kapcsolat oldal.](#kapcs) 10-11.
* [Letöltési információ.](#letinf) 11.-12.
* [Operációs rendszer választás.](#opv) 12.
* [Footer rész.](#footer) 12.

* [Vélemények/Vélemény írása oldal](#velemenyek). 13.

* [Adatbázis.](#pdb) 14-15.

* [MediaQuery](#mq). 15.

* [Továbbfejlesztési lehetőségek](#tfl). 16.

* [Miért jött létre?](#miert)/[Csapattagok és feladatköreik](#csapattagokfeladat) 16.

* [Tesztek](#test) 16.-17.

* [Források](#f). 17.

[Technikai információ](#elso)

1. Project információk
2. Vue.Js 3(Composation API)
3. CSSGrid
4. FlexBox
5. Tailwindcss(CSS Freamwork)
6. Bootstrap(icons)
7. CSS animations
8. CSS Variables
9. MediaQuery(Bövebben a 15.Oldalon)
10. Json-server(Bövebben a 14.Oldalon)

Miután letöltöttük GitHub-ról a .zip fájlt a kitömörítés után a rendszer termináljában ki kell adnunk az npm install (vagy npm i) parancsot a szükséges node-moduls telepítéséhez.

A project elindításához szükséges elindítanunk magát a projectet, ami localhost-on fut (a parancs npm run dev) az adatok megjelenítéséhez pedig el kell indítanunk a json-servert (a parancs npx json-server syncbundle.json).

**Megjegyzés:** A json-server csak ennél a projectnél szerepel ezen a néven.

A Vue.Js egy úgynevezett SPA(single-page application) technologiát használ amely lehetővé teszi hogy minden tartalmat egyetlen oldalra töltsünk be ahelyett hogy a teljes oldalt újra kellene tölteni minden alkalommal ha új szakaszra lépünk. Az SPA-k az AJAX(Asynchronous JavaScript and XML) nevű technológiát használják, hogy szükség szerint kis adatbiteket töltsenek be a szerverről. Ez azt jelenti, hogy csak a legutóbbi frissítés óta megváltozott tartalom töltődik be így jelentősen csökken az átvitt adatok mennyisége.

Van még néhány előnye: mivel minden tartalom egy menetben töltődik, sokkal gyorsabb és egyszerűbb a frissítés. Ez különösen a JavaScript könyvtárak használatakor igaz. Mint mindennek az SPA-nak is megvannak az előnyei és hátrányai.

**Előnyök:**

* gyorsaság és reakciókézség
* jobb felhasználói élmény
* könnyebb fejlesztés

**Hátrányok:**

* JavaScript függőség
* SEO optimalizálás
* keresési előzmények

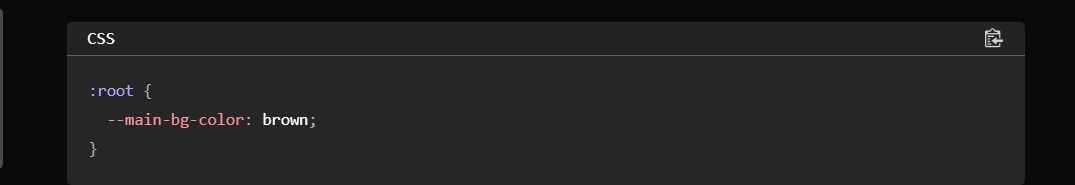
A mai világban alapvető fontosságú a weboldalak esztétikus megjelenése ehhez a HTML és CSS leírónyelveket használjuk. A Grid System (rácsszerkezet-szisztéma) a CSS leírónyelv egy eszköze, ami segítségével egy tetszőleges rácsszerkezetet hozhatunk létre a GRID-et nem minden böngésző támogatja mégis elterjedt a használata.

A Flexbox az előbbiekhez hasonlóan a weboldalon lévő elemek pozicionálására szolgál bár használata nem mindig praktikus viszont vannak olyan specifikus problémák, amiket ’elegánsabban” lehet vele megoldani.

A Tailwindcss egy CSS keretrendszer, amely lehetővé teszi a fejlesztők számára, hogy CSS stílusokat hozzanak létre egyéni CSS írása nélkül. CSS-szabályok definiálása helyett a segédszabályokat közvetlen a HTML-elemekre alkalmazhatja, ez a megközelítés gyorsítja a fejlesztési folyamatot.

A BootStrap ikonok egy olyan segédkönyvtár, amely SVG állományokat tartalmaz.

CSS animációk segítségével interaktívvá tehetjük a weboldalunk 100% CSS használatával néhány CSS animáció: @keyframe, animation-name, animation-duration stb.

Elkerülhetetlen a CSS használata az interneten megjelent tartalmak számára. A változók használata egy olyan lehetőség, amellyel „pofásabbá” tehetjük a munkánk ahogy a neve mutatja a kódunk különböző szintjein meghatározhatunk egy változót, aminek értéket adunk így a későbbiekben elég csak erre a változóra hivatkoznunk. Használata a következő képen látszik: 

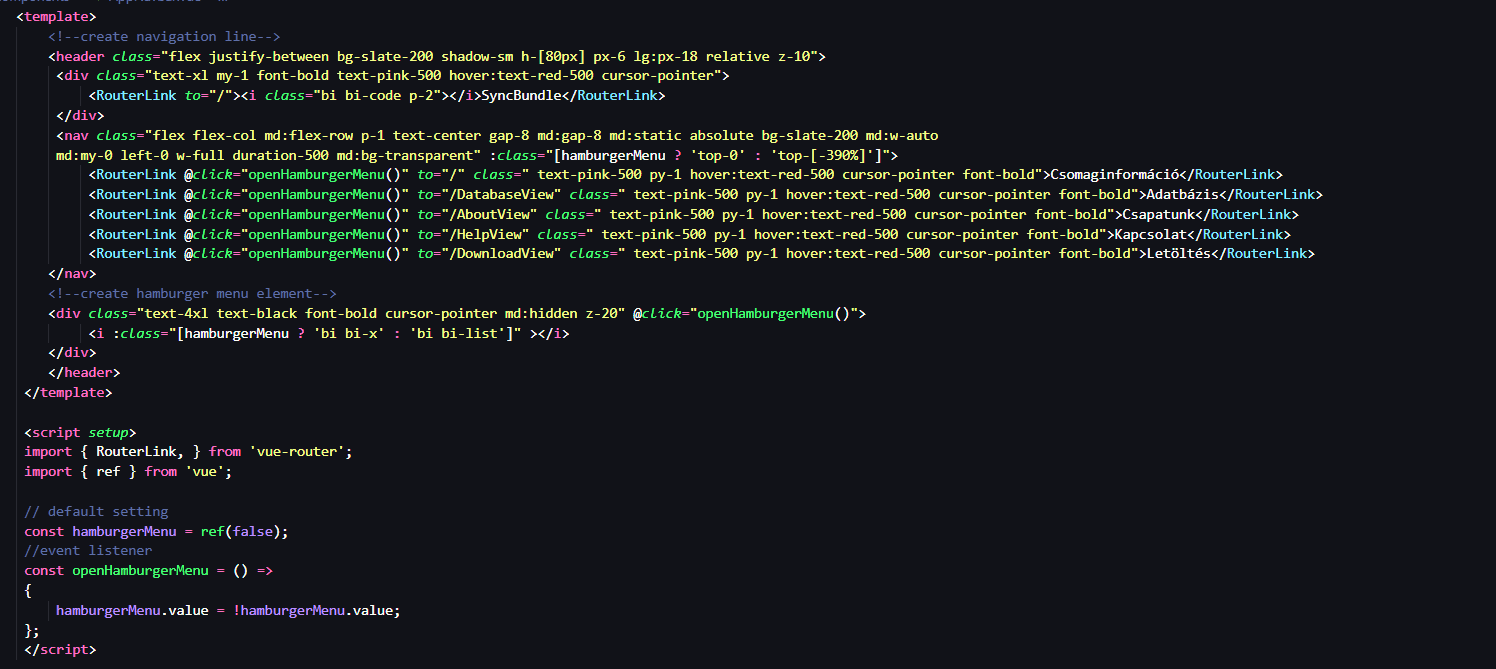
MediaQuery-k segítségével tehetjük responsiv viselkedésűvé oldalunkat, azaz minden kijelzőn más-más elrendezést adhatunk az elemeknek. Használata a következő képen látszik:

A képen szöveg, képernyőkép, Betűtípus, sor látható

Automatikusan generált leírás

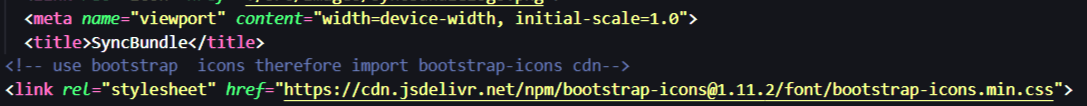
[Header rész](#masodik)

A Header rész tartalmazza a navigációt, ez a rész display: flex-re van állítva az ezen belüli nav tag szintén display: flex amely alapértelmezés szerint col elrenezésű viszont md méretnél(768px) átvált row-ra ez biztosítja a hamburger menü fentről lefelé lenyíljon ezáltal a kisebb kijelzőkön is élvezhető megjelnést add.



Ehhez a részhez Tailwindcss stílusok lettek felhasználva, melyek open sorce jellegűek ezáltal biztosított az ingyenes elérés és a szabad felhasználás, azonban kisebb egyedisége van az így készült elemeknek.

Ezen kívül a kódban szerepel egy icon melyet cdn segítségével használok a BootStrap oldaláról.



A megjelenése közvetlenül a projectnév részlete mellett található.

A képen Betűtípus, Grafika, fehér, tervezés látható

Automatikusan generált leírás A „< >” jelek egy <i> tag-ben helyezkednek el.

[Csomaginformációs oldal](#harmadik)

Az oldalon megjelenő tartalom responsive viselkedéséért CSS Grid beállítások felelnek melyek töréspontjai a tartalomhoz mérten lettek beállítva a megfelelő olvasási élményhez igazodva. Továbbá az oldalon található képek egy külön mappában lettek elhelyezve (images mappa), ezen kívül a megjelenő adatokat a csomagról egy json-server szolgáltatja.

Az oldalon látható képek 768px alatt eltűnnek ez egy egyszerű display:none beállítás.

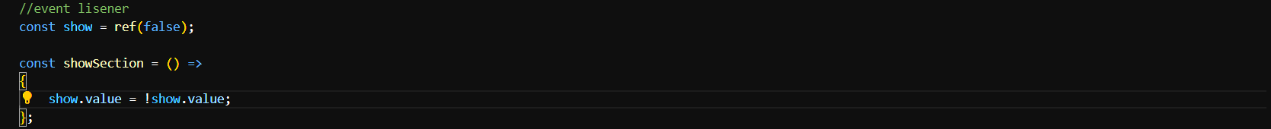


A „Miért jött létre?” oldalrészlet mellet egy BootStrap ikon található, amit a következő képen láthatunk.

A képen szimbólum, csavarkulcs, tervezés látható

Automatikusan generált leírás

Ez az ikon click eseményre reagál és megjelenik a hozzá tartozó szöveg, ez a kódrészlet lentebb látható.



Mivel a WordPress Plugin (Bővítmény) és a WordPress Theme (Téma) információk összetartoznak ezért a nagyobb kijelzőkön egymás mellett helyezkednek el viszont 650px-nél egymás alá kerülnek. Ez grid-column és grid-row segítségével érhető el. A következő kódrészlet bemutatja a beállítást.

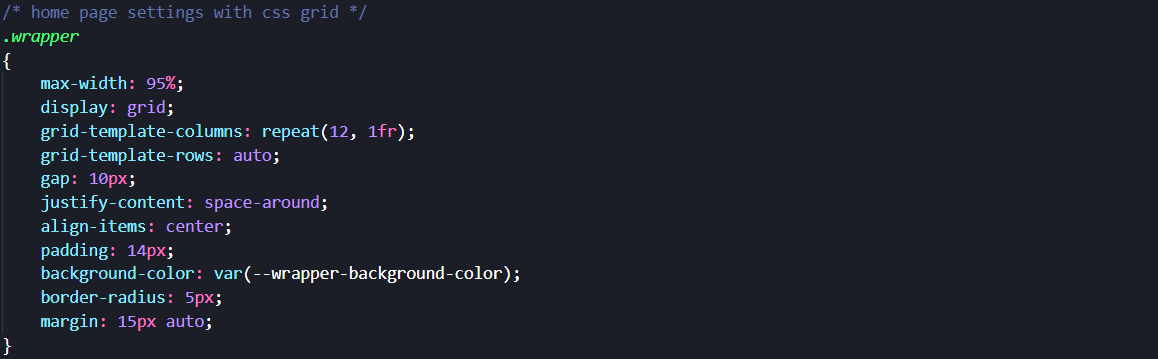


A következő kepén látható az oldal(ak) SEO beállítása, a Searc Engine Optimalization azaz a kereső optimalizálás nagyon fontos szempont egy honlap számára egyes felmérések arra jutottak, hogy az emberek az első legfeljebb az első 3 találatot nézik meg ezért fontos, hogy weblapinkon belegyen állítva a SEO. Persze itt különbség van a fizetett és nem fizetett tartalmak közt! A következő képen m3gtekinthető a kód.

A képen képernyőkép, Betűtípus, szöveg látható

Automatikusan generált leírás

Az lentebb látható képen láthatóak a szülő „wrapper” beállításai CSS Grid-el.



[Adatbázis oldal](#negyedik)

Ez az oldal mutatja be az alkalmazás által használt adatbázist és a hozzá tartozó jellemzőket az oldal responsive viselkedésű ennek érdekében CSS Grid-et használ. Az itt látható adatok egy json-serverről érkeznek melynek a kódja a következő képen lesz látható.



A megjelenő tartalmak animáltak az oldal elején található adatbázis bemutatása rész két összetevőből épül fel ez azt jelenti, hogy a bemutató szöveg egy animáció segítségével cserélhető ennek köszönhetően kevesebb tartalom megjelenése lehetséges az oldalon egy adott pillanatban.

A következő képeken bemutatom az animációhoz tartozó kódokat.

JavaScript



HTML



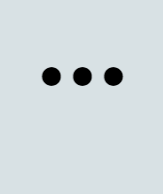
CSS



Az imént szemléltetett kódok mindegyike újra felhasználható, szabadon alakítható az aktuális projekthez mérten. Viszont nem szabad megfelejtkeznünk a változónevek átevezéséről az áltatunk használt nevekre.

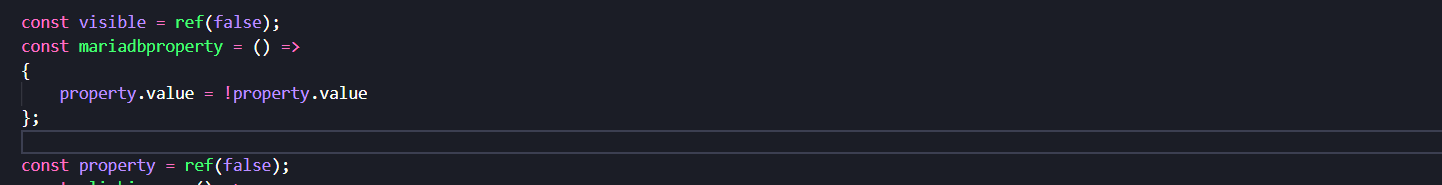
 A képen látható BootStrap ikonra kattintva elindul az animáció.

A jellemzők az adatbázissal kapcsolatban egy <ul> tag-ben helyezkednek el, aminek az <li> tag-ei tartalmaznak egy <button> elemet ez az, ami közvetlenül felel ez animációért és itt található egy <strong> elemben a tulajdonság neve. Az első tulajdonságnév mellett látható BootStrap ikon felelős az animáció aktiválásáért, ami a következő képen látható.



Az ikonra kattintva megjelennek a jellemző tulajdonságok leírásai a következő képeken láthatóak a beállítások.

JavaScript



HTML

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

Az imént szemléltetett kódok mindegyike újra felhasználható, szabadon alakítható az aktuális projekthez mérten. Viszont nem szabad megfelejtkeznünk a változónevek átevezéséről az áltatunk használt nevekre.

Csapatinformációs oldal

A fejlesztői csapatot bemutató oldal nem tartalmaz animációt csak a CSS Grid beállításokat mivel a csapat 3 tagja (Pernyész Dániel, Tesszáry Péter, Bagi Péter)3 külön-külön lévő „dobozban” kerülnek bemutatásra, ami tartalmaz egy képet és egy rövid bemutatkozást.

A következő képeken látható a teljes CSS kód az oldalra szabottan.

A képen szöveg, képernyőkép, szoftver, Multimédiás szoftver látható

Automatikusan generált leírás

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

A képen képernyőkép, szöveg látható

Automatikusan generált leírás

A képen szöveg, képernyőkép, Multimédiás szoftver, szoftver látható

Automatikusan generált leírás

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

A képen szöveg, képernyőkép látható

Automatikusan generált leírás

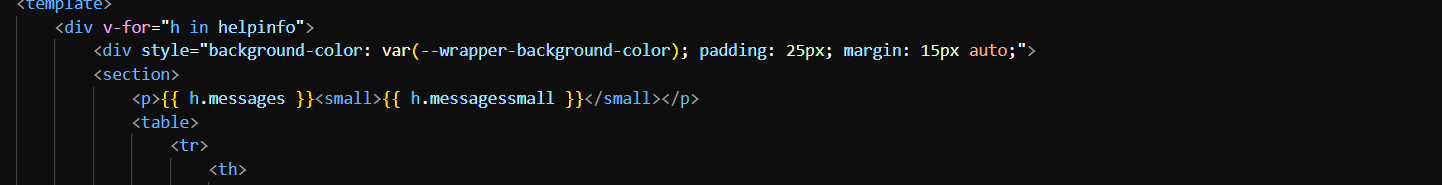
A Grid beállításai az oldalra szabottak és minden kijelzőn esztétikusan jelennek meg.

Az adatok ennél az oldalnál is egy json-szerverről jönnek.



[Kapcsolati oldal](#hat)

Az oldal első része CSS Grid használatával van elhelyezve a „wraper” tetején ez a rész csak információs szerepet tölt be és minden kijelzőméretnél ott helyezkedik el.



Ezután következik egy table tag ami display:flexre van állítva az ezt követő <tr> tag szintén display:flex amit közre zár a display:flex <th> ez biztosítja hogy minden kijelzőméreten középen legyen a táblázat.Ez a táblázat tartalmazza a csapattagok neveit ami egy <a> tag-ben található a kiválasztott névre kattintva megnyílik az e-mail kliensünk és azonnal írhatük üzenetet az <a> tag-ben megadott „mailto” property segítségével.

A következő képeken láthatóak a HTML rész és a hozzá tartozó CSS beállítás.

A képen szöveg, képernyőkép, Betűtípus látható

Automatikusan generált leírás

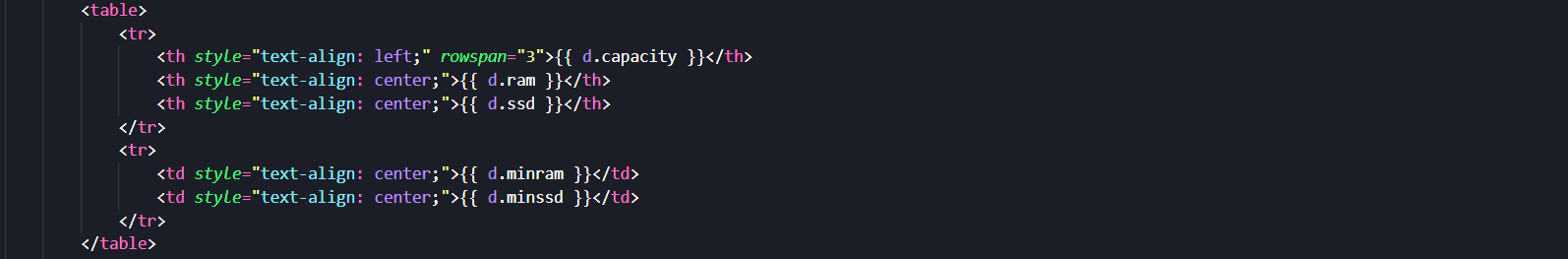
A képen képernyőkép, szöveg, Multimédiás szoftver látható

Automatikusan generált leírás

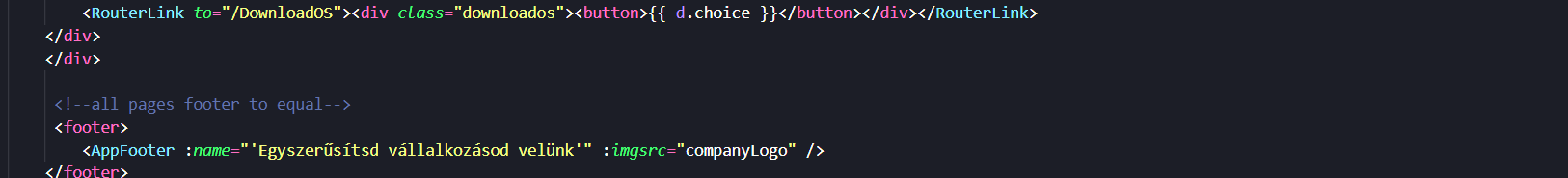
Továbbá az oldalon található adatok egy json-serverről jönnek az előbbiekben ismertetett módon.

[Letöltési információk oldal](#het)

A letöltés oldalon található a szoftvercsomag gépigényére vonatkozó információk egy táblázatban foglalnak helyet, amelynek első része a „Gépigény” három sort foglal el ez a sorok összevonásával érhető el (rowspan).



Az ezalatt található „Letöltés” gomb egy <RouterLink> amely átnavigál az operációs rendszer választása oldalra, amely alapértelmezés szerint nem látható.

wss

[Operációs rendszer választás](#nyolc)

A megfelelő operációs rendszer letöltésére szolgáló oldal akkor jelenik meg mikor a letöltés oldalon lévő „Letöltés” feliratú gombra kattintunk. Az oldalon felsorolt operációs rendszer változatok egy-egy <RoterLink> elemben vannak elhelyezve, amik kattintás után a főoldalra irányítanak vissza. A kódok az alábbiakban tekinthetők meg.



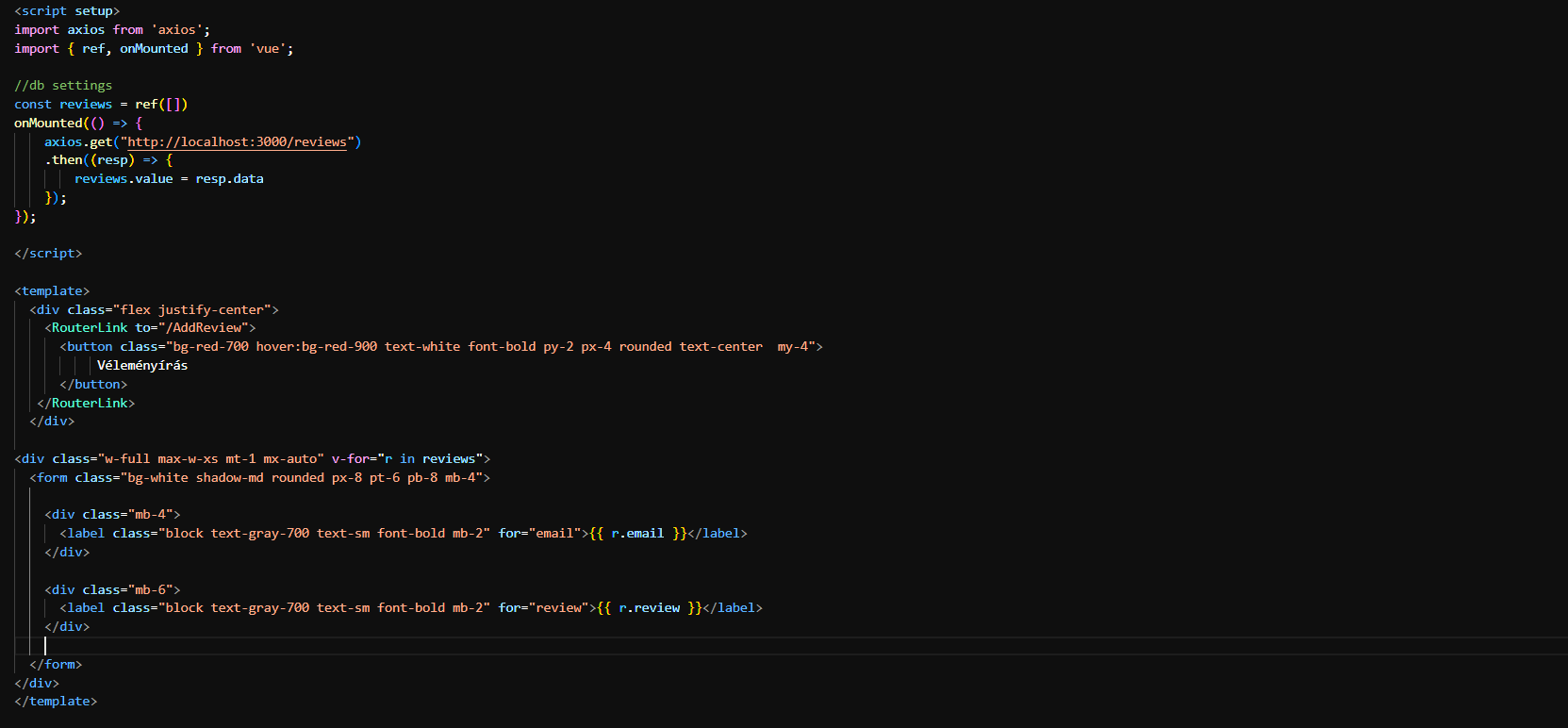
[Footer rész](#kilenc)

A Footer rész minden oldalon azonosan megjelenik a lábléc tartalmazza a logót és egy rövid szöveget a responsitvitás érdekében FlexBox-al lett igazítva amelyet Tailwindcss segítségével használok. A lenti képeken megtalálható a Footer rész beállítása.

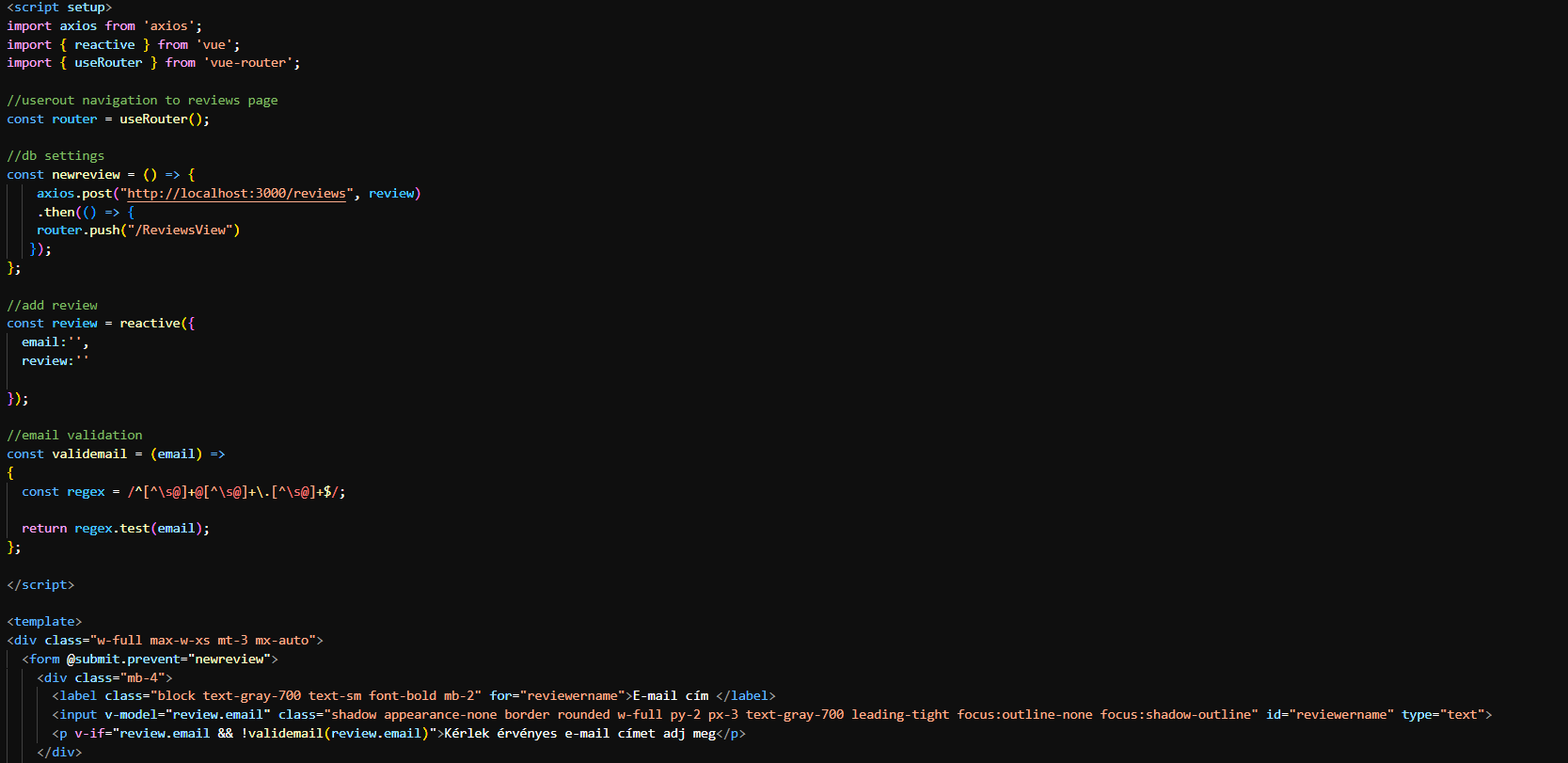


[Vélemények/Vélemény írás oldal](#tiz)

A két oldal felépítése azonos Tailwindcss alkalmazásával készültek a véleményeket megjelenítő oldal kódja alább látható.



Látható a „Véleményírás” oldalra navigáló <RouterLink> ami előhozza az oldalt, amin szabadon írhatunk véleményt.



Az alkalmazott Tailwindcss által adott könnyítés ellenére jól látható hogy szükséges egy kisebb JavaScript kódrészt írni hogy működőképes legyen a funkció, továbbá validációs beállítások hozzáadása is szükséges mivel csak helyes e-mail cím megadása után lesz látható a form ahová a véleményt lehet írni.

A „Közzétesz” gomb azután aktiválódik miután szabályos az e-mail cím és a véleményeknek kialakított rész nem üres a lenti képen látható az a kódrészlet, amivel ezt el tudjuk érni.

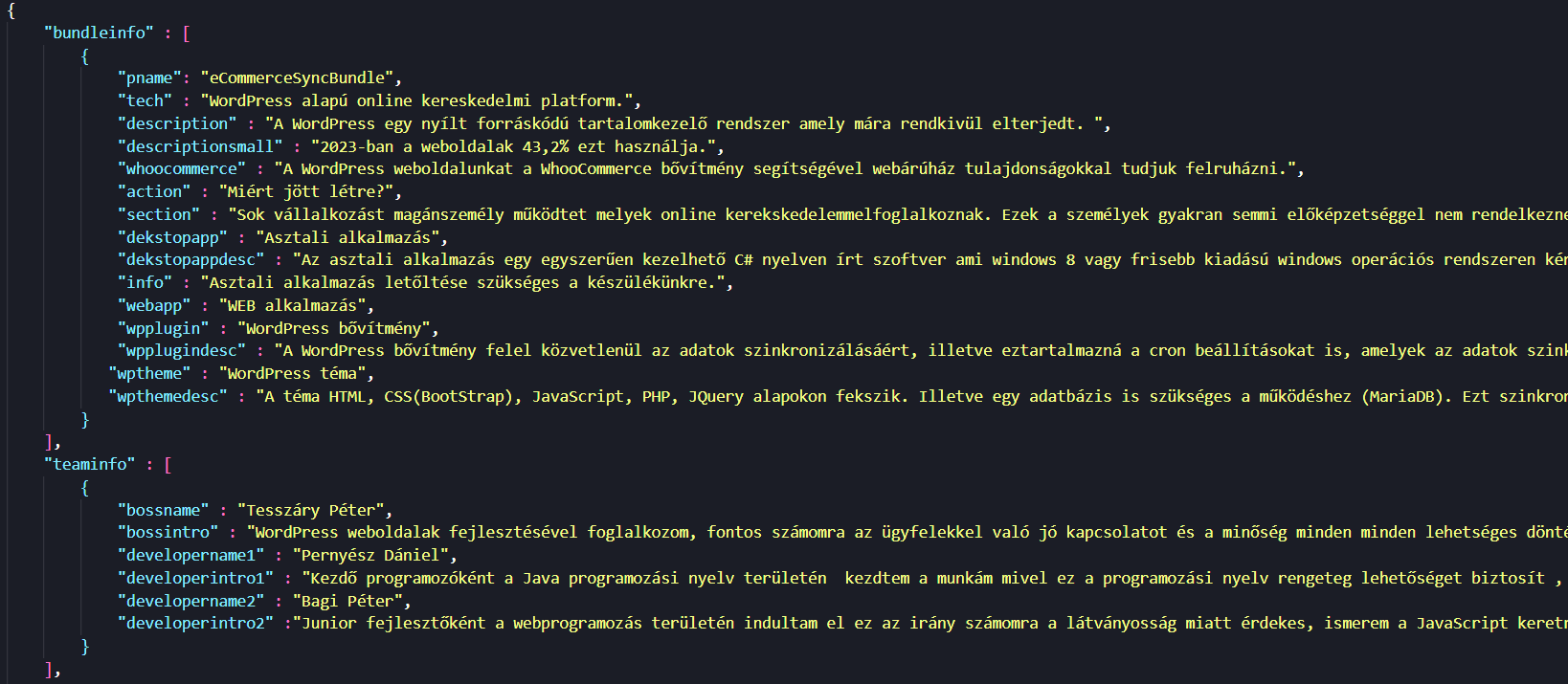
A képen képernyőkép, szöveg, Betűtípus látható

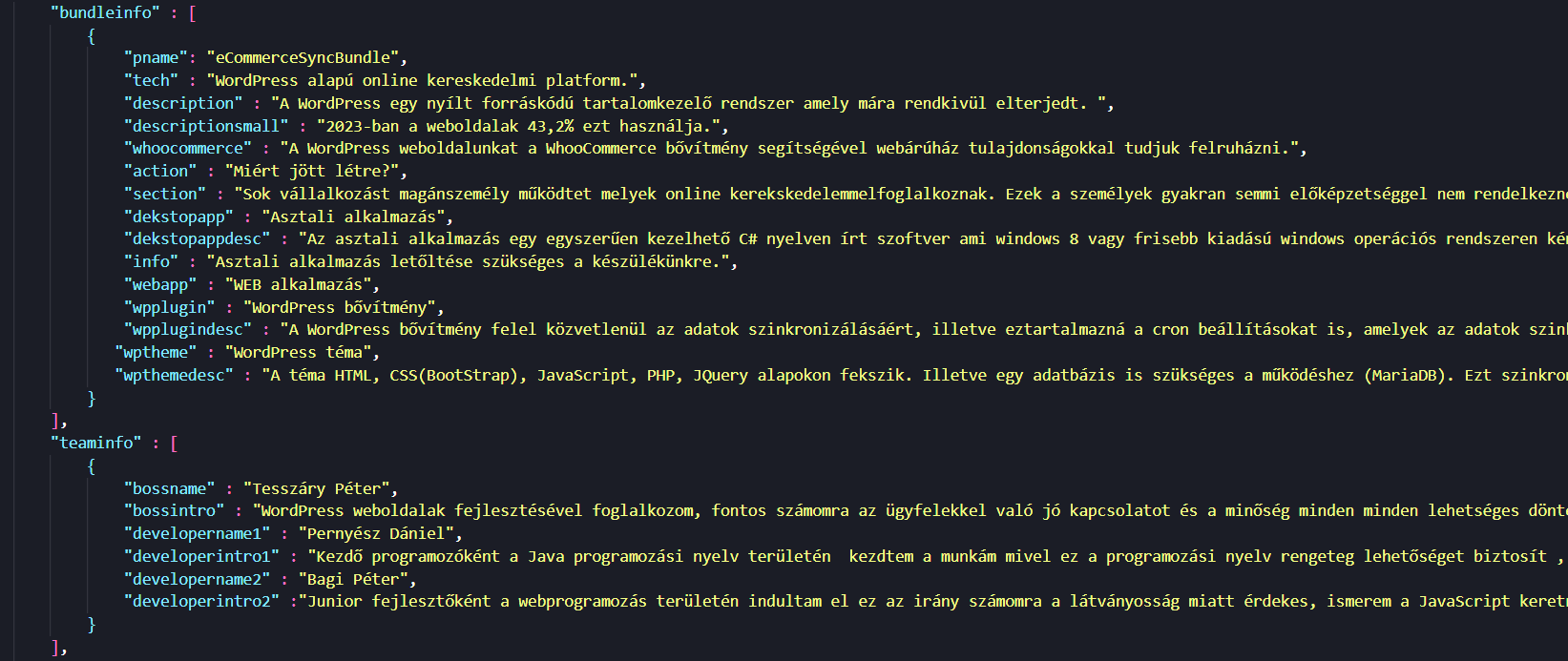
Automatikusan generált leírás

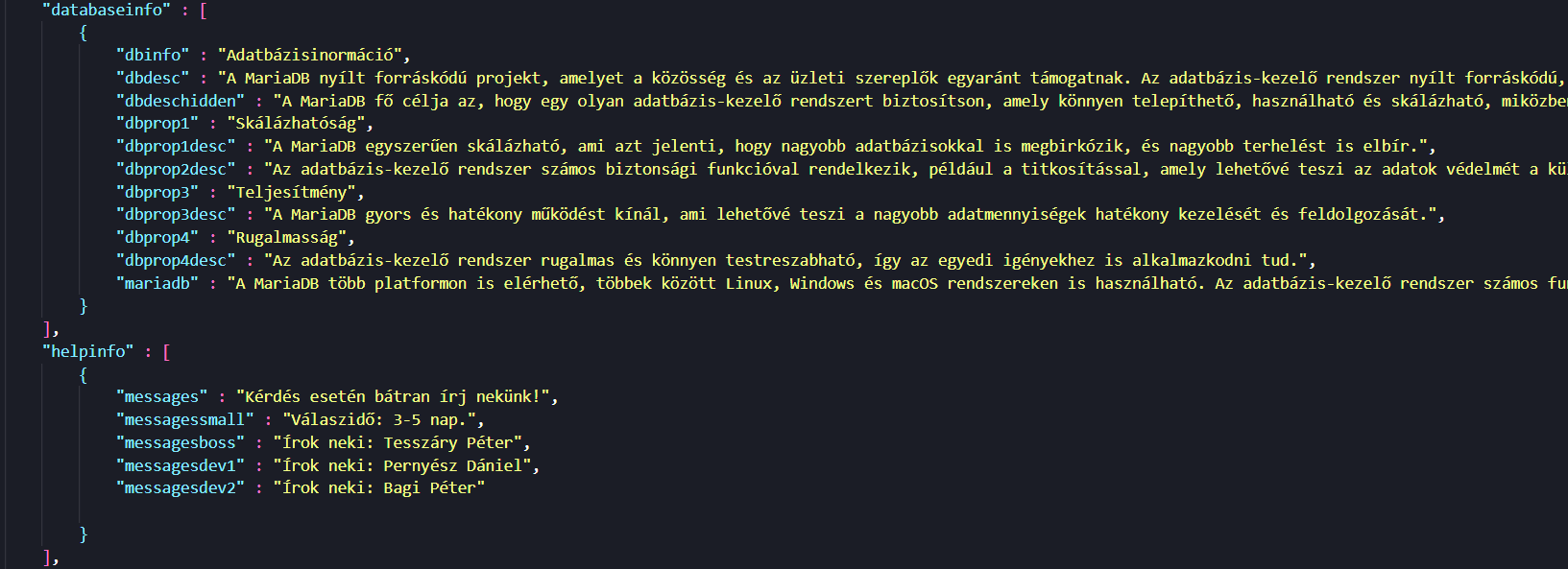
[Adatbázis](#tizenegy)

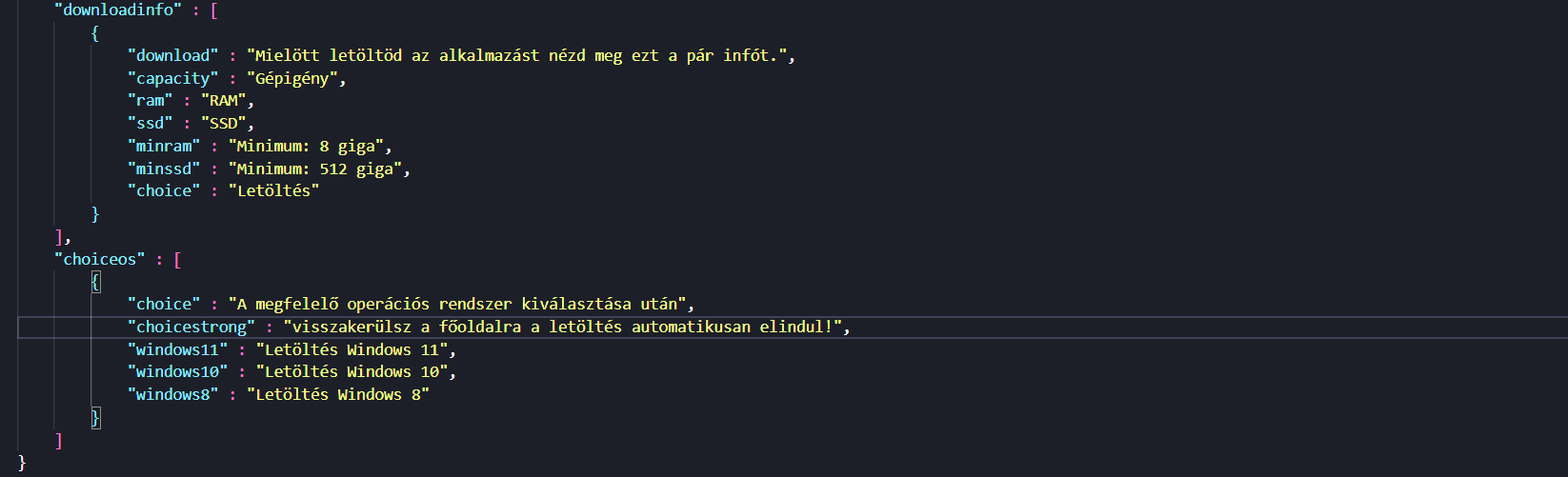
JSON(JavaScript Object Notation) használatával készült az adatbázis ez a kiszolgáló egy Node.js eszköz amely egy RESTful API-t szimulál ezáltal a Frontend fejlesztők könnyen létrehozhatnak API-kat. Fontos hogy az „API” felé küldött kérések csak szimulációk viszont minden HTTP kérésre megfelelő választ kapunk ez ideális a Frontend fejlesztők gyors munkavégzéséhez.

A file szintakszisa:









[MediaQuery](#tizenketto)

Az oldalak mindegyike MediaQuery segítségével igazodik az aktuális kijelzőhöz a mobilefirst elvet alkalmazva először mobiltelefon kijelzőkön lett tesztelve a kinézet a töréspontok az adott tartalom igényeinek megfelelően lettek kialakítva.

A következő képek megmutatják az oldalaknál alkalmazott töréspontokat.

****

****

****

Saját MediaQuery írása lassítja a fejlesztés folyamatát, de a manapság népszerű CSS keretrendszerek túlzott használata miatt a WEB-es felületen megjelenő tartalmak kevésbé egyediek.

[Továbbfejlesztési lehetőségek](#tizenharom)

A WEB alkalmazás fejlesztési lehetőségei széles körűek erre néhány példa.

* SQL adatbázis kapcsolatot
* operációs rendszerek listáját ki bővíteni
* Csapat oldalra telefon esetleg fax elérhetőség hozzáadása
* Adatbázis oldalhoz kódrészlet képek hozzáadása
* Kód újjászervezésé
* Teljeskörű tesztelés

[Miért jött létre?](#tizennegy)

Manapság sok webshop-ot magánszemély üzemeltet ezért a megírt programmal szerettünk volna nekik segítséget nyújtani számukra.

[Csapattagok és feladataik](#nyolc)

Tesszáry Péter, Pernyész Dániel: Asztali alkalmazás fejlesztése C# nyelven, WordPress Téma és Bővítmény.

Bagi Péter: Weboldal fejlesztése Vue.Js segítségével.

[Tesztek](#esetek)

Funkcióteszteket végeztem a Vitest segítségével, ami egy modern tesztelési lehetőség Frontendes funkciók vizsgálatára. Habár ezt a technikát minden node.js alapú keretrendszer támogatja a beállítások projectenként eltérhetnek.

Az alábbiakban az általam használt teszteseteket mutatom be.

A képen szöveg, képernyőkép, szoftver, számítógép látható

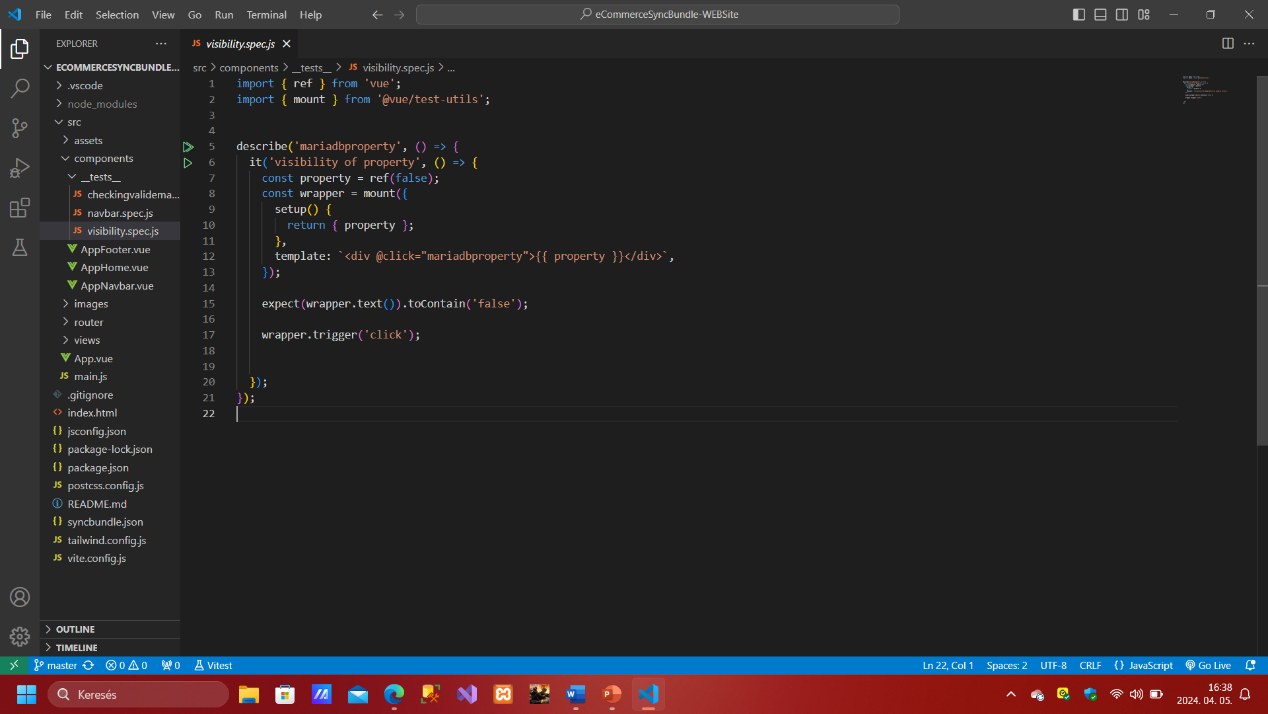
Automatikusan generált leírás

Ez a teszteset a navbar viselkedését a navigáció pontosságát figyeli.

A képen szöveg, képernyőkép, szoftver, számítógép látható

Automatikusan generált leírás

A fenti a teszteset azt figyeli, hogy a szükséges változás megtörténik-e a helyes e-mail cím beírása után, azaz a <p> tagba írt figyelmeztetés eltűnik e.



A teszteset vizsgálja, hogy az adatbázis tulajdonságok megfelelően jelenek-e meg, ehhez létre kellett hozni egy teszt template-t.

# [Források](#tizenot)

[Vue.js - The Progressive JavaScript Framework | Vue.js (vuejs.org)](https://vuejs.org/)

[Bevezetés a Vue.js keretrendszerbe | ITHub](https://ithub.hu/blog/post/Bevezetes-a-Vuejs-keretrendszerbe)

[CSS animáció készítése - WEBiskola](https://webiskola.hu/css-ismeretek/css-animacio-animation-keszitese/#:~:text=Kezd%C3%A9snek%20n%C3%A9zz%C3%BCk%20meg%2C%20hogy%20milyen%20anim%C3%A1ci%C3%B3%20tulajdons%C3%A1gok%20l%C3%A9teznek,5%20animation-iteration-count%20%28Az%20anim%C3%A1ci%C3%B3%20ism%C3%A9tl%C5%91d%C3%A9s%C3%A9nek%20sz%C3%A1ma.%29%20Tov%C3%A1bbi%20elemek)

[json-server - npm (npmjs.com)](https://www.npmjs.com/package/json-server?activeTab=versions)

[Reszponzív weboldal készítése (Viewport és CSS Media Query) - WEBiskola](https://webiskola.hu/css-ismeretek/reszponziv-weboldal-keszitese-viewport-css-media-query/)

[A CSS Flexbox bemutatása példákon keresztül, kódokkal együtt! | Gremmédia (gremmedia.hu)](https://gremmedia.hu/css-flexbox-bemutatasa-peldakon-keresztul)

[Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.](https://tailwindcss.com/)

[Tailwind CSS for Beginners: A Step-by-Step Guide - DEV Community](https://dev.to/akashakki/tailwind-css-for-beginners-a-step-by-step-guide-3gff#:~:text=Tailwind%20CSS%20is%20a%20CSS%20framework%20that%20provides,development%20process%20and%20offers%20great%20flexibility%20for%20customization.)

[Single-Page Applications: A Comprehensive Guide - KeyCDN](https://www.keycdn.com/blog/single-page-application)

[Kódolj okosabban - css változók - Online Webdesign Tanfolyam - Szabadúszó webdesigner kurzus mentorálással](https://webdesigntanfolyam.com/kodolj-okosabban-css-valtozok/)

[A CSS GRID bemutatása-részletes példákkal, kódokkal együtt! (gremmedia.hu)](https://gremmedia.hu/css-grid-bemutatasa-peldakkal-kiserve)

[Bootstrap Icons · Official open source SVG icon library for Bootstrap (getbootstrap.com)](https://icons.getbootstrap.com/)