Bevezetés:

Manapság 2022-ben és az elmúlt években egyre nagyobb teret hódítottak az online vásárlási felületek és az IT szféra fejlődésének köszönhetően egyre jobban arra felé hajlunk, hogy amit csak lehet, azt valamilyen online felületen végezzük, ez az ügyintézéstől fogva a meetingek és találkozók megtartásán keresztül egészen a vásárlásig terjedt ki. Ezért döntöttünk úgy mi sem állunk a változás újtába és a növekvő igénynek megfelelve mi ezen a területen szeretnénk létrehozni átfogó porjektet, amely teljesen életképes a ma, azaz 2022 igényeinek, teljes mértékben megfelel és kielégítő felhasználói élményt nyújt.

Fejünkben többféle remek ötlet is megfordult, példának okáért egy futárszolgálat weboldala, online kaszinó ötlete és egy mezőgazdasági videómegosztó platform is, de végül egy webáruház gondolata tűnt a leg ideálisabbnak és a leg kézenfekvőbbnek.

A webshop tárgyát/témáját illetően több lehetőség is megfogalmazódott;

Például:

* Bútorbolt
  + Indoklás:
    - A környezetünkben több irányultságú cég található, a projektet megpróbáltunk volna náluk értékesíteni
  + Elhatározás:
    - Végül is úgy döntöttünk, hogy más ötleteket is megvizsgálunk
* „OnlySelfies” online tartalommegosztó oldal
  + Indoklás:
    - Magas kereslet a hasonló monotizálható tartalommegosztó oldalak iránt
  + Elhatározás:
    - Ezt a praktikát etikátlannak találtuk és így az ötletet elvetettük
* Online kaszinó
  + Indoklás
    - Az online tartalomgyártók promócióiknak köszönhetően, ez az iparág világszerte nagy teret hódított az elmúlt pár év alatt.
  + Elhatározás
    - A piac telítettsége miatt más ötlet mellett döntöttünk

Specifikáció:

Vésősoron egy webáruház megvalósítása mellett döntöttünk. Mivel napjainkban az online vásárlás nagy teret hódított, ezért kézenfekvőnek tűnt, hogy mi is egy webshopot készítsünk. Véleményünk szerint ez a projekt kellően összetett és a modern igényekhez mérten aktuális ahhoz, hogy a vizsga követelményeinek megfeleljen.

A webshop megvalósítását illetően több lehetőség is nyitva állt előttünk:

Ami a backendet illeti mindenképpen a jelenleg rendelkezésre álló technológiákat figyelembe véve, csakis kizárólag, keretrendszerek használata jöhetett szóba, a natív programnyelven írt backend ötletét elvetettük, a szükségtelen bonyolultsága miatt.

Továbbá szükségünk van keretrendszerekre illetve azok alkalmazására, hiszen ezeknek a segítségével, jóval gyorsabban dolgozhatunk, valósíthatjuk meg az adott funkcionalitást.

Ezen felül, gondoljunk csak a böngésző kompatibilitásra: ez a probléma nem csupán CSS esetén merülhet fel. A megírt kódunk nem egyformán fog működni minden böngésző, illetve azok verziói alatt.

Mindezek eredményeként a mai modern webfejlesztésben gyakorlatilag mindenki keretrendszereket, és könyvtárakat is használ.

Elsősorban két lehetőséget vettünk fontolóra: A PHP alapú Laravel framework-öt és a JavaScript alapú Express.js –t.

-

-

-

Ami a Laravel-t illeti mindenképpen előnyt jelentett, hogy maga a framework nagyon jól dokumentált és napjainkban továbbra is széles körben elterjedt és nagy népszerűségnek örvend.

A Laravel erősen az MVC modellen alapul, ami számunkra szintén pozitívum volt, mert, elsősorban ilyen keretrendszereket használtunk a múltban.

Viszont jelenleg a JavaScript alapú rendszerekben több tapasztalattal rendelkeztünk ezért választottuk az Express.js-t.

-

-

-

Az Express.js –re esett végül a választásunk.

A framework szintén az MVC modellt követi, defőleg az miatt döntöttünk az express mellett, mert ez egy JavaScript alapú framework melynek syntax-át jelentősan jobban kedveltünk a PHP-val szemben. AZ express szintén egy elterjedt megoldás a backendek terén és nekünk is több tapasztalatunk van ennek a megoldásnak a használatában és nagyobb óraszámban is tanultuk, bár az utóbbi nemfeltétlenül volt szempont mivel funkcióiban mind a két framework használatával megtudjuk csinálni ugyanazt a projektet.

A frontend terén is több lehetőséget vettünk fontolóra:

A Typescript alapú Angular és a Javascript alapú Vue.js lehetőségét mérlegeltük. Egyedül natív kódolással hasonló okok miatt most sem szerettünk volna dolgozni, mindenképpen most is az imént említett lehetőségekből szerettünk volna választani és natív Javascript kódot csak akkor szerettünk volna alkalmazni, amikor olyan dolgot szerettünk volna megvalósítani, amelyet a framework nem tartalmaz.

Az Angular bár napjainkban egy elég elterjedt megoldás, a környezetünkben is elismert cégek programoznak benne, mi személyes preferencia alapján a Vue.js mellett döntöttünk.

Szeretnénk megjegyezni, hogy bár most nem az Angulárra esett a választásunk, de mindenképpen megéri most a közös projekt mellett és a későbbiekben is, nagy hangsúlyt fektetni rá, hiszen a jelenlegi ismereteink alapján elmondhatjuk, hogy sok lehetőség van benne. A Typescript bár egyesek mondhatják, hogy nehezebb nyelv, mint a Javascript, mindenképpen megéri a bele fektetett időt, melyet később majd a személyes gyakorló projektek által igenis bele fogunk fektetni.

A frontedet Vue.js-ben írjuk!

A Vue.js egy **nagyon könnyen megtanulható, és használható JavaScript könyvtár**. Alacsony Learning Curve-vel rendelkezik.

A Vue.js-t annak ellenére, hogy könnyebben elsajátítható, rengeteg feladatra könnyedén fel lehet használni. Kifejezetten szeretik startupok is használni, valamint kiváló párosítást alkot [Laravel](https://gremmedia.hu/edukacio/bejegyzes/laravel-keretrendszer-mit-erdemes-tudni-rola-tenyleg-legnepszerubb-keretrendszer)keretrendszerrel.

Igaz most ezt az előnyét nem fogjuk kihasználni.

Szintén pozitívumnak könyveltük el, hogy a Vue.js-ben minden eddiginél könnyebben tudtunk teszt adatokat felvenni olyan struktúrában, amilyenben majd már élesben fogja kapni az adatokat és így a kezdetektől fogva koncentrálni tudtunk egy dinamikus frontend létrehozására.

Ami az adatbázis típusát illeti, szintén két lehetőség közül választottunk: a MySQL és a NoSQL között.

Bár a MySQL tűnhet az egyértelmű megoldásnak, a NoSQL lehetőségét komolyan fontolóra vettük és végül az mellett is döntöttünk.

Erre a döntésre nem voltak hasonló indokaink, mint az előzőekben, hiszen mind a két megoldás teljesen valid és széles körben használt megoldás.

Az adatstruktúrák jelentősen különböznek, de mind a két megoldás teljesen átlátható tehát ez nemvolt szempont. A NoSQL az adatokat BSON formátumban tárolja majd küldés előtt alakítja át JSON-be, ez miatt az adatforgalom gyorsabb, bár ez az előny leginkább nagyobb adathalmazoknál mutatkozik meg, a mi projektünknél ez az előny elhanyagolható.

Végsősoron a döntésünket személyes preferencia alapján hoztuk meg.