

Baptista Szeretetszolgálat EJSZ Széchenyi István Gimnáziuma És Technikuma

Vizsgaremek

Készítették:

Bruder Botond
Novográdecz Milán László

Pécs

2025

Baptista Szeretetszolgálat EJSZ Széchenyi István Gimnáziuma És Technikuma

Szakma megnevezése: Szoftverfejlesztő és tesztelő

Szakma azonosító száma: 506131203

Vizsgaremek

LuxHorizon

Készítették:

Bruder Botond

Novográdecz Milán László

Tartalomjegyzék

Készítették:	1
Bevezetés	4
Feladat rövid ismertetése	4
Cél	4
Jövőbeli tervek	4
Használt technológiák	4
• Programozási nyelvek	5
• Adatbázis	5
• Egyéb	5
Adatbázis	5
Táblák:	5
Elvégzett tesztek	6
Tesztelési fajták	6

Jelentőség	7
Frontend részletes leírása	7
Elrendezés:.....	7
Felhasználói felület (UI) elemek és reszponzivitás:	8
Felhasználói interakciók:.....	8
Felhasználói élmény (UX) elvek:	9
Kiemelkedő frontend megoldások:	9
Program működésének a részletes leírása	9
1. Regisztráció:	9
2. Bejelentkezés:.....	10
3. Autók keresése és szűrése:	10
4. Autó kiválasztása és részletek megtekintése:	10
5. Foglalás:	10
6. Visszajelzés a foglalásról:	11
Backend részletes leírása	11
Adatbázis-kezelés részletesen:.....	11
Autók keresése:	12
Felhasználói munkamenetek (Sessions):	12
Hibakezelés:.....	12
Fejlesztési technológiák	13
• HTML és CSS	13
• JavaScript.....	13
• Bootstrap	13
• PHP	13
• MySQL.....	14
• PHPUnit	14
Csapatmunka	14
• Bruder Botond	14
• Novográdecz Milán	14
Kommunikáció és projektmenedzsment:	14
Verziókövetés:	15
Kihívások és megoldások:	15
Kódminőség biztosítása:	15
Frontend és backend összehangolása:	15
Lezárás	15

Bevezetés

A sebesség és az exkluzív autók világa régóta vonzza a rajongók széles táborát. Projektünk erre a szenvedélyre építve egy egyedülálló online platformot hoz létre, amely lehetővé teszi a sportautók kedvelőinek, hogy Magyarország izgalmas versenypályáin élhessék át a vezetés páratlan élményét. A weboldalunk célja, hogy áthidalja a szakadékot az álomautók és a valós vezetési élmény között, biztonságos és professzionális környezetet teremtve a bérléshez.

Ez a dokumentáció részletesen bemutatja a szoftverfejlesztési folyamatot, a projekt technikai hátterét, az alkalmazott technológiákat, valamint a minőségbiztosítási eljárásokat. Célunk, hogy átfogó képet nyújtsunk a projekt minden aspektusáról, a tervezéstől kezdve a megvalósításon át a tesztelésig. A következőkben feltárjuk a rendszer architekturáját, a frontend és backend komponensek működését, az adatbázis felépítését, valamint az elvégzett tesztek eredményeit.

Feladat rövid ismertetése

A projekt célja egy olyan weboldal létrehozása, amely sportautók bérletét kínálja Magyarország különböző versenypályáin. A platform elsődleges célközönsége a sportautók iránt érdeklődő és a vezetés élményét versenypályán megtapasztalni vágyó felhasználók. A weboldal fő funkciója az autók bérletének egyszerű és átlátható biztosítása.

Cél

A projekt végső célja egy országosan elismert és megbízható sportautó-kölcsönző platform kiépítése, amely a magyar versenypályákon kínál egyedülálló lehetőséget a sportautók szerelmeseinek. A projekt hátterében az autók iránti mély szenvedély áll.

Jövőbeli tervek

A jövőben tervezzük a weboldal funkcionalitásának bővítését további felhasználóbarát funkciók bevezetésével. Ilyen lehet például a pálya foglalási rendszer integrálása, instruktorok bevonása a vezetési élmény fokozása érdekében, valamint közösségi funkciók kialakítása a felhasználók interakciójának elősegítésére. Célunk egy folyamatosan fejlődő és a felhasználói igényekhez igazodó platform létrehozása.

Használt technológiák

A projekt fejlesztése során az alábbi technológiákat alkalmaztuk:

- **Programozási nyelvek:** HTML, CSS, JavaScript, Bootstrap, PHP
- **Adatbázis:** MySQL
- **Egyéb:** Bootstrap (a reszponzív design és az egységes megjelenés biztosítására)

Adatbázis

Táblák:

A projekt adatainak tárolására MySQL adatbázist használtunk. Az adatbázis az alábbi főbb táblákból áll:

- **categories:** Az autó kategóriáit tárolja (pl. sportautó, szuperautó). Tartalmazza a kategória azonosítóját (category_id), nevét (name) és leírását (description).
- **brands:** Az autó márkáit tárolja (pl. Ferrari, Porsche). Tartalmazza a márka azonosítóját (brand_id), nevét (name), logójának URL-jét (logo_url) és leírását (description).
- **cars:** Az elérhető autókat tárolja. Tartalmazza az autó azonosítóját (car_id), a márkáját (brand_id - hivatkozik a brands táblára), a kategóriáját (category_id - hivatkozik a categories táblára), modelljét (model), motorjának típusát (engine), lóerejét (hp), végsebességét (top_speed), sebességváltójának típusát (gears), meghajtását (drive), gyorsulását (acceleration), képének URL-jét (image_url), napi bérleti díját (rent_price), valamint a különböző lapszámhoz tartozó bérleti díjakat (lap_price_1, lap_price_3, lap_price_5, lap_price_10) és az elérhetőségét (available).
- **tracks:** A bérelhető versenypályákat tárolja. Tartalmazza a pálya azonosítóját (track_id), nevét (name), hosszát (length), leírását (description), képének URL-jét (image_url), a pálya elrendezésének URL-jét (layout_url) és a galéria elérhetőségét (gallery).
- **users:** A regisztrált felhasználók adatait tárolja. Tartalmazza a felhasználó azonosítóját (user_id), nevét (name), e-mail címét (email), jelszavának hash-ét (password_hash), regisztrációjának időpontját (created_at) és utolsó bejelentkezésének időpontját (last_login).
- **bookings:** A leadott foglalásokat tárolja. Tartalmazza a foglalás azonosítóját (booking_id), a felhasználó azonosítóját (user_id - hivatkozik a users táblára), az autó azonosítóját (car_id - hivatkozik a cars táblára), a pálya azonosítóját (track_id - hivatkozik a tracks táblára), a bérlet kezdetének dátumát (start_date), a bérlet végének dátumát (end_date), a teljes árat (total_price),

a foglalás típusát (booking_type - enum: 'rental', 'track_day'), a megtett körök számát (laps), a foglalás státuszát (status - enum: 'pending', 'confirmed', 'completed', 'cancelled') és a foglalás létrehozásának időpontját (created_at).

Kapcsolatok:

Az egyes táblák között a következő kapcsolatok állnak fenn:

- Egy márka (brands) több autóhoz (cars) tartozhat (one-to-many).
- Egy kategóriához (categories) több autó (cars) tartozhat (one-to-many).
- Egy felhasználó (users) több foglalást (bookings) hozhat létre (one-to-many).
- Egy autó (cars) több foglalásban (bookings) szerepelhet (one-to-many).
- Egy versenypályán (tracks) több foglalás (bookings) történhet (one-to-many).

Ez a struktúra biztosítja a szükséges adatok hatékony tárolását és lekérdezését a weboldal működéséhez.

Elvégzett tesztek

A projekt minőségének biztosítása érdekében különböző típusú tesztek végeztünk a fejlesztés során. Különös hangsúlyt fektettünk a felhasználói felület (frontend) tesztelésére és a backend komponensek egységtesztjeire.

Tesztelési fajták

- **Manuális frontend tesztek:** Összesen 90 darab manuális tesztet hajtottunk végre a weboldal frontendjén. Ezek a tesztek lépésről lépésre dokumentáltak lettek egy Excel táblában, lefedve a különböző felhasználói interakciókat, a vizuális elemek helyességét, a responzivitást és az alapvető funkcionalitást (pl. navigáció, űrlapok működése, autók listázása, bérleti folyamat). A részletes tesztjegyzőkönyvek biztosítják a tesztek átláthatóságát és a reprodukálhatóságot.
- **PHPUnit tesztek:** A backend logikájának tesztelésére PHPUnit keretrendszert használtunk. Ezek az egységtesztek az egyes PHP osztályok és metódusok elkülönített működését vizsgálták, biztosítva a kód helyességét és robusztusságát. A tesztek lefedték az adatbázis-interakciókat, az üzleti logikai szabályokat és az API végpontok alapvető működését.

Jelentőség

Az elvégzett tesztek kulcsfontosságúak voltak a projekt sikeres megvalósítása szempontjából. A manuális frontend tesztek lehetővé tették a felhasználói élmény szempontjából kritikus hibák és inkonzisztenciák feltárását, biztosítva egy felhasználóbarát és intuitív felületet. A PHPUnit tesztek pedig a backend kód megbízhatóságát és stabilitását garantálták, csökkentve a későbbi éles környezetben felmerülő hibák kockázatát. A tesztelés során szerzett tapasztalatok segítettek a kód finomításában és a projekt minőségének jelentős javításában.

Ezzel lényegében a teljes dokumentáció vázlatát lefedtük a megadott információk alapján. Mit gondolsz, van még valami, amit fontosnak tartasz hozzáadni vagy részletezni? Esetleg a bevezetőt szeretnéd még finomítani? Ha így megfelel, a következő lépés lehet a teljes dokumentáció egybefüggő szöveggé formálása és a stilisztikai egységesítése.

Frontend részletes leírása

Elrendezés:

A weboldal felhasználói felülete letisztult és intuitív kialakításra törekszik, hogy a sportautók iránt érdeklődők könnyedén megtalálják a számukra legmegfelelőbb bérleti lehetőséget. A főbb oldalak és azok funkcionalitása a következők:

- **Főoldal (Index Page):** A weboldal központi kiindulópontja, ahol a felhasználók áttekintést kaphatnak a kínált autógyártókról. Az oldalon a különböző autógyártók logói vagy kiemelt képei jelennek meg. A felhasználó egy adott gyártóra kattintva megtekintheti az adott márka összes elérhető sportautóját.
- **Autógyártó Oldala:** Ezen az oldalon az adott gyártó összes bérelhető sportautójának listája található. Az autók megjelenítése "card layout" formában történik, ahol minden kártya tartalmazza az autó legfontosabb adatait (pl. modell, kép, alapár) és egy gombot a részletesebb adatok megtekintéséhez.

- **Autó Részletei Oldal:** A kiválasztott autó típusának dedikált oldala. Itt a felhasználó részletes információkat találhat az autóról, beleértve a főbb műszaki tulajdonságait (pl. motor, lóerő, végsebesség, gyorsulás), képgalériát és a bérleti lehetőségeket. Az oldal kiemelt részén található a foglalási rendszer, ahol a felhasználó megadhatja a bérlet dátumát, a versenypályát, a kívánt körök számát, és azonnal láthatja a bérlet összegzését. Ezen az oldalon egy gomb is található, amely átirányítja a felhasználót az adott autó hivatalos gyártói weboldalára.
- **Versenypályák Összesítő Oldala:** Ez az oldal listázza az összes elérhető magyar versenypályát, ahol a sportautók bérelhetők. A pályák megjelenítése itt is "card layout" formában történik, tartalmazva a pálya nevét, egy képet és egy gombot a részletes adatok megtekintéséhez.
- **Versenypálya Részletei Oldal:** A kiválasztott versenypálya részletes bemutatása. Az oldalon megtalálható a pálya leírása, hossza, elrendezése, valamint egy képgaléria a pályáról. Ezen az oldalon is elhelyeztünk egy gombot, amely a pálya hivatalos weboldalára vezet.

Felhasználói felület (UI) elemek és reszponzivitás:

A weboldal felhasználói felületének kialakításakor a letisztultság és az átláthatóság voltak a fő szempontok. A navigáció egyszerű és intuitív, a fontos információk könnyen megtalálhatók. A "card layout" alkalmazása mind az autók, mind a versenypályák listázásakor biztosítja a strukturált és áttekinthető megjelenítést különböző képernyőméretekken. A Bootstrap keretrendszer teljes körű kihasználásával valósítottuk meg a reszponzív design-t, így a weboldal optimálisan jelenik meg asztali számítógépeken, táblagépeken és okostelefonokon is. Különböző Bootstrap grid layoutokat és komponenseket használtunk az egyes oldalak elrendezéséhez, biztosítva a tartalom megfelelő igazodását a különböző nézetekhez.

Felhasználói interakciók:

A felhasználói interakciók kezelése natív JavaScripttel és PHP-val történik. A frontend oldalon a JavaScript felelős a dinamikus elemek működéséért, a hover effektekért, a színek változásáért és az azonnali visszajelzésekért (pl. űrlapok validálása). A gombokra történő kattintások eseménykezelői JavaScriptben kerültek implementálásra, amelyek szükség esetén AJAX kéréseket indítanak a PHP backend felé az adatok lekérdezéséhez vagy a foglalások feldolgozásához. Az űrlapok validálása részben a frontend oldalon JavaScripttel történik a gyors visszajelzés érdekében, a végső validáció és az adatok feldolgozása pedig a PHP backend feladatkörébe tartozik.

Felhasználói élmény (UX) elvek:

A felhasználói élmény tervezése során az alábbi elveket követtük:

- **Egyszerű navigáció:** A felhasználók könnyen és gyorsan eljuthassanak a kívánt információhoz.
- **Átlátható információ:** A lényeges adatok (pl. autó tulajdonságai, bérleti feltételek, árak) egyértelműen és strukturáltan jelenjenek meg.
- **Gyors visszajelzés:** A rendszer azonnali visszajelzést adjon a felhasználói interakciókra (pl. sikeres foglalás, űrlaphiba).
- **Letisztultság:** A vizuális megjelenés ne legyen zavaró, a hangsúly a tartalom legyen.
- **Szép színek:** A weboldal színvilága kellemes és a sportautók dinamikus világához illeszkedjen.

Kiemelkedő frontend megoldások:

A weboldal frontendjén számos interaktív elemet alkalmaztunk a felhasználói élmény fokozására. A hover effektek és a színek változása a különböző elemeken (pl. kártyák, gombok) vizuális visszajelzést adnak a felhasználói interakciókra. Az autó részletei oldalon a képgaléria dinamikus megjelenítése és a foglalási rendszer interaktív árösszegezése emelhető ki. A foglalási folyamat során a dátum és a körök számának megadásakor a rendszer azonnal frissíti a bérlet összegét, átláthatóvá téve a költségeket a felhasználó számára.

Ez a részletes leírás a frontendről jelentősen növeli a dokumentáció terjedelmét. Mit gondolsz, ez így megfelelő részletességű, vagy van még valami, amit érdemes lenne hozzáadni vagy másképp megfogalmazni?

Program működésének a részletes leírása

A weboldalunkon a sportautók bérlete a magyar versenypályákon regisztrációhoz és bejelentkezéshez kötött folyamat. A felhasználói élményt a regisztrációtól a sikeres foglalásig végigkísérő lépések a következők:

1. Regisztráció:

A weboldal használatához új felhasználóknak először regisztrálniuk kell. A regisztrációs folyamat során a felhasználónak meg kell adnia egy egyedi felhasználónevet, egy érvényes e-mail címet és egy legalább 8 karakter hosszú jelszót. A frontend oldalon azonnali ellenőrzés történik az e-mail cím formátumára és a jelszó hosszára. Amennyiben a megadott adatok nem felelnek meg a követelményeknek, a rendszer azonnali hibaüzenetet jelenít meg, segítve a

felhasználót a helyes adatok megadásában. A sikeres regisztrációt követően a felhasználó bejelentkezhet a weboldalra.

2. Bejelentkezés:

A regisztrált felhasználók az e-mail címük és jelszavuk megadásával jelentkezhetnek be a weboldalra. A rendszer ellenőrzi a megadott e-mail címet és jelszót.

Amennyiben a megadott e-mail cím nem található a regisztrált felhasználók között, vagy a jelszó helytelen, a rendszer megfelelő hibaüzenetet jelenít meg. A sikeres bejelentkezést követően a felhasználó hozzáférhet a weboldal összes funkciójához, beleértve az autók böngészését és a foglalások kezelését. A bejelentkezés tehát elengedhetetlen a weboldal funkcióinak eléréséhez.

3. Autók keresése és szűrése:

A bejelentkezést követően a felhasználók többféleképpen kereshetnek a bérelhető sportautók között. A főoldalon a gyártók listájából kiindulva eljuthatnak az adott márka összes autójához. Az autókat tartalmazó oldalakon lehetőség van szűkíteni a találatokat különböző paraméterek alapján, mint például:

- **Márka:** A felhasználó kiválaszthat egy vagy több kedvenc márkát.
- **Modell:** Lehetőség van konkrét modellek keresésére.
- **Lóerő:** Az autók teljesítményére lehet szűrni egy adott tartományon belül.
- **Motor:** A motor típusa (pl. benzines, elektromos) alapján is lehet szűrni.
- **Végsebesség:** A maximális sebesség alapján lehet szűkíteni a listát.
- **Sebességi fokozatok száma:** A sebességváltó típusára vonatkozó szűrési lehetőség.
- **Hajtás:** A meghajtás típusa (pl. összkerékhajtás - AWD, hátsókerékhajtás - RWD) alapján is lehet szűrni.
- **Gyorsulás (0-100 km/h):** A gyorsulási érték alapján is lehet szűkíteni a találatokat.

4. Autó kiválasztása és részletek megtekintése:

A keresési eredmények között a felhasználó kiválaszthat egy érdeklődő autót, amelyre kattintva megnyílik az autó részletes oldala. Ezen az oldalon a felhasználó megtekintheti az autó összes fontos tulajdonságát, képeit és a bérleti lehetőségeket.

5. Foglalás:

Az autó részletei oldalon található a foglalási rendszer. A felhasználónak a következő adatokat kell megadnia a foglaláshoz:

- **Helyszín (versenypálya):** A legördülő menüből ki kell választania a kívánt versenypályát.
- **Dátum:** A naptárból ki kell választania a bérlet kívánt dátumát.
- **Körök száma:** Meg kell adnia, hogy hány kört szeretne megtenni az adott pályán az adott autóval.

A megadott adatok alapján a rendszer azonnal kiszámolja a bérlet összegét, figyelembe véve az autó bérleti díját és a választott körök számát. A "Foglalás" gombra kattintva a rendszer ellenőrzi, hogy az adott autó a kiválasztott időpontban és pályán elérhető-e.

6. Visszajelzés a foglalásról:

A foglalás elküldését követően a felhasználó egy oldalon belüli üzenetben kap visszajelzést a foglalás sikerességéről. Amennyiben az adott autó a kiválasztott időpontban már foglalt, a rendszer hibaüzenetet jelenít meg, és a felhasználónak lehetősége van másik időpontot vagy autót választani. Sikeres foglalás esetén a rendszer rögzíti a foglalást, és a felhasználó a profiljában megtekintheti a részleteket.

Backend részletes leírása

A weboldalunk backendjének alapját a PHP programozási nyelv képezi, amely a szerveroldali logika oroszlánrészéért felelős. A rendszer a MySQL adatbázissal szoros együttműködésben biztosítja a dinamikus tartalom megjelenítését és a felhasználói interakciók kezelését.

Adatbázis-kezelés részletesen:

A natív SQL lekérdezések használata a backendben azt jelenti, hogy a PHP kód közvetlenül kommunikál az adatbázis-szerverrel. Ez magában foglalja az SQL utasítások (SELECT, INSERT, UPDATE, DELETE) dinamikus generálását a PHP változók és a felhasználói bemenetek alapján. Például, amikor egy felhasználó egy adott márkára szűr az autók listájában, a PHP kód egy olyan SQL SELECT lekérdezést épít fel, amely a WHERE záradékban tartalmazza a kiválasztott márka azonosítóját. Hasonlóképpen, egy új foglalás létrehozásakor a PHP kód egy INSERT utasítást generál, amely a felhasználó által megadott adatokat (autó azonosító, pálya azonosító, dátum, körök száma stb.) illeszti be a bookings táblába.

A natív SQL használatának előnye a nagyfokú kontroll az adatbázis-műveletek felett és a potenciálisan jobb teljesítmény bizonyos esetekben, mivel nincs egy ORM réteg többletterhelése. Ugyanakkor nagyobb odafigyelést igényel a fejlesztők részéről a biztonságos SQL lekérdezések írása érdekében, elkerülve az SQL injection támadásokat. A projekt során gondosan ügyeltünk a felhasználói bemenetek validálására és a lekérdezések paraméterezésére a biztonság garantálása érdekében.

Autók keresése:

A bejelentkezést követően a felhasználók a bérelhető sportautók között a főoldalon található gyártók listájából választhatnak. Egy adott gyártóra kattintva megtekinthetik az adott márka összes elérhető sportautóját. Jelenleg a rendszer nem kínál további szűrési lehetőségeket a márkán kívül. A felhasználók a gyártó oldalán láthatják az adott márka összes elérhető modelljét.

Felhasználói munkamenetek (Sessions):

A PHP session mechanizmusa lehetővé teszi a szerver számára, hogy információkat tároljon egy adott felhasználóval kapcsolatban a különböző oldallátogatások között. Amikor egy felhasználó sikeresen bejelentkezik, a PHP elindít egy új munkamenetet. A szerver generál egy egyedi munkamenet-azonosítót, amelyet elküld a felhasználó böngészőjének egy cookie-ban. A böngésző ezt a cookie-t automatikusan elküldi a szervernek minden további kérésnél. A szerver a cookie-ban található azonosító alapján azonosítja a felhasználóhoz tartozó munkamenet-adatokat, amelyek a szerver memóriájában vagy egy fájlban tárolódnak. Ebben a munkamenetben tárolhatók a felhasználó bejelentkezési állapota, felhasználói azonosítója és egyéb, a felhasználói élményhez vagy a biztonsághoz szükséges információk. A kijelentkezéskor a munkamenet megsemmisül.

Hibakezelés:

A robusztus hibakezelés elengedhetetlen egy stabil webalkalmazás működéséhez. A backend PHP kódunkban kiterjedten használjuk a try-catch blokkokat a potenciálisan problémás műveletek (például adatbázis-lekérdezések, fájlkezelés) körül. Ha egy kivétel (hiba) történik a try blokkban lévő kód végrehajtása során, a vezérlés azonnal átkerül a catch blokkba. A catch blokkban lehetőségünk van a hiba naplózására (például egy log fájlba), egy felhasználóbarát hibaüzenet generálására a frontend számára, vagy akár egy adott hiba típusához tartozó speciális hibakód visszaadására. A hibakódok segíthetnek a frontendnek a hiba okának pontosabb azonosításában és a megfelelő visszajelzés megjelenítésében a felhasználó számára. Például egy sikertelen adatbázis-lekérdezés esetén egy 500-as HTTP

státusz kód és egy részletesebb hibaüzenet kerülhet visszaadásra a frontendnek a fejlesztés során, míg éles környezetben egy általánosabb hibaüzenet jelenhet meg a felhasználónak a biztonság érdekében.

Fejlesztési technológiák

A projekt során gondosan megválasztott technológiák kulcsszerepet játszottak a weboldal sikeres megvalósításában, biztosítva a felhasználóbarát felületet, a hatékony működést és a megbízható adatkezelést.

- **HTML és CSS:** A weboldal alapját a strukturált HTML szemantikus elemekkel való felépítése képezte, amely biztosítja a tartalom logikus elrendezését és a keresőmotorok számára való optimális indexelhetőséget. A CSS stíluslapok feleltek a weboldal vizuális megjelenéséért, a letisztult design és a felhasználóbarát megjelenés megteremtéséért. Különös figyelmet fordítottunk a CSS szabályok átláthatóságára és karbantarthatóságára. A felhasználóbarátság szempontjai vezéreltek mind a HTML struktúra, mind a CSS stílusok kialakításakor.
- **JavaScript:** A JavaScript a frontend oldalon dinamikus funkcionalitás biztosításáért volt felelős. Főbb feladatai közé tartozott az adatok aszinkron lekérése a backend API-ről AJAX kérések segítségével, a felhasználói interakciók (például gombnyomások) kezelése és néhány vizuális effekt (például hover effektek, színek változása) megvalósítása. A JavaScript kódunk moduláris felépítésre törekedett a jobb karbantarthatóság érdekében.
- **Bootstrap:** A Bootstrap keretrendszert a reszponzív design gyors és hatékony megvalósítása, valamint a gazdag UI komponens készlete miatt választottuk. A Bootstrap grid rendszerének köszönhetően a weboldalunk különböző képernyőméreteken is optimálisan jelenik meg. Számos előre elkészített komponenst (például navigációs sáv, gombok, kártyák) használtunk fel, amelyek jelentősen felgyorsították a fejlesztési folyamatot és egységes megjelenést biztosítottak. A Bootstrap által kínált ikonok szintén fontos szerepet játszottak a felhasználói felület vizuális gazdagításában és a funkciók érthetőbbé tételében.
- **PHP:** A PHP a szerveroldali logika megvalósításának fő eszköze volt. A PHP szkriptek kezelték a frontendről érkező kéréseket, kommunikáltak a MySQL adatbázissal a PHPMyAdmin által létrehozott közvetlen kapcsolaton keresztül, és generálták a válaszokat a frontend számára (leggyakrabban

JSON formátumban). A projekt a PHP legújabb elérhető verzióját használta a legújabb funkciók és biztonsági javítások kihasználása érdekében.

- **MySQL:** A MySQL relációs adatbázis-kezelő rendszert a megbízhatósága és a PHP-val való jó kompatibilitása miatt választottuk a projekt adatainak tárolására. Az adatbázis sémájának tervezésekor a hatékony adatelérés és a kapcsolatok integritásának biztosítása voltak a fő szempontok.
- **PHPUnit:** A backend kód minőségének biztosítása érdekében unit tesztek írtunk PHPUnit keretrendszer segítségével. Ezek a tesztek az egyes PHP függvények és osztályok elkülönített működését vizsgálták, biztosítva a kód helyességét és a várt eredményeket különböző bemenetek esetén. A unit tesztek segítettek a kód korai szakaszában feltárni a hibákat és megakadályozni azok tovaterjedését a rendszer más részeire.

Csapatmunka

A projekt megvalósításában egy kétfős fejlesztői csapat vett részt:

- **Bruder Botond:** Frontend fejlesztő és tesztelő. Fő feladata a weboldal felhasználói felületének kialakítása, a responzivitás biztosítása és a frontend funkcionalitásának tesztelése volt.
- **Novográdecz Milán:** Backend fejlesztő és tesztelő. Fő feladata a szerveroldali logika implementálása, az adatbázis-kezelés és a backend komponenseinek tesztelése volt.

Feladatok megosztása:

A feladatok megosztása a csapattagok szakértelme és érdeklődése mentén történt. Bruder Botond a felhasználói felület tervezésére és implementálására, valamint a felhasználói interakciók tesztelésére koncentrált. Novográdecz Milán a háttérben zajló folyamatokért, az adatbázis-kapcsolatokért és a szerveroldali üzleti logikáért felelt, emellett a backend komponenseit is tesztelte. Bár a szerepek specializáltak voltak, a projekt sikere érdekében szoros együttműködésre törekedtünk.

Kommunikáció és projektmenedzsment:

A csapaton belüli kommunikáció folyamatos és közvetlen volt. A felmerülő kérdéseket, problémákat és ötleteket rendszeresen megbeszéltük, biztosítva a közös megértést és a hatékony haladást. Bár dedikált projektmenedzsment eszközt nem használtunk, a feladatok nyomon követésére minimálisan a GitHub projektkezelési funkcióit vettük igénybe.

Verziókövetés:

A verziókövetésre a Git-et használtuk, amely lehetővé tette a kód változásainak nyomon követését, a párhuzamos fejlesztést és a biztonságos együttműködést. Különböző ágakat hoztunk létre az új funkciók fejlesztéséhez és a hibajavításokhoz, majd ezeket integráltuk a fő ágba a stabil verziók megőrzése érdekében.

Kihívások és megoldások:

A projekt során számos kihívással szembesültünk, amelyek főként a komplex funkcionalitás megvalósításából és a technikai részletek finomításából adódtak. Ezeket a kihívásokat a csapat rengeteg beletett órával, közös ötleteléssel és folyamatos önfejlesztéssel sikerült leküzdenie. A problémák megoldásában kulcsszerepet játszott a proaktív kommunikáció és a csapattagok elkötelezettsége a projekt sikere iránt.

Kódminőség biztosítása:

A kódminőség biztosítása érdekében informális kódáttekintéseket tartottunk, ahol a csapattagok átnézték egymás kódját a hibák felderítése és a legjobb gyakorlatok betartása érdekében. Emellett a következetes kódolási stílusra is nagy hangsúlyt fektettünk.

Frontend és backend összehangolása:

A frontend és a backend fejlesztésének összehangolása iteratív módon történt. Először a frontend alapvető elemei kerültek kialakításra, majd erre építve fejlesztettük a backend funkcionalitást. A két terület közötti szoros kommunikáció és a jó közös hang elengedhetetlen volt a zökkenőmentes integrációhoz és a weboldal egységes működéséhez.

Lezárás

A jelen dokumentáció részletesen bemutatta a sportautó-bérlési weboldal fejlesztésének folyamatát a koncepciótól a tesztelésig. A projekt célja egy felhasználóbarát és átlátható platform létrehozása volt, amely lehetővé teszi a sportautók kedvelőinek, hogy Magyarország versenypályáin élhessék át a vezetés élményét.

A dokumentáció áttekintette a projekt háttérét és célkitűzéseit, a felhasznált technológiákat, az adatbázis felépítését, a frontend és backend részletes működését, az elvégzett tesztelési folyamatokat, valamint a fejlesztői csapatmunkát.

A fejlesztés során a felhasználói élmény központi szerepet játszott. A letisztult design, az egyszerű navigáció és az átlátható foglalási folyamat mind azt a célt szolgálták, hogy a sportautók iránt érdeklődők könnyen és élvezetesen használhassák a weboldalt. A rezponzív kialakítás biztosítja a platform elérhetőségét különböző eszközökön.

A backend robusztus működése és a hatékony adatbázis-kezelés elengedhetetlen volt a weboldal funkcionalitásának biztosításához. A frontend és a backend szoros együttműködése tette lehetővé a dinamikus tartalom megjelenítését és a felhasználói interakciók zökkenőmentes kezelését.

Az elvégzett manuális és unit tesztek kulcsfontosságúak voltak a projekt minőségének biztosításában, lehetővé téve a hibák korai felismerését és javítását. A fejlesztői csapat elkötelezett munkája és a hatékony kommunikáció elengedhetetlen volt a projekt sikeres megvalósításához.

Bár a projekt elérte a kitűzött célokat, a jövőbeli tervek között szerepel a weboldal további fejlesztése új funkciókkal és a felhasználói felület finomításával. Célunk, hogy egy folyamatosan fejlődő és a felhasználói igényekhez igazodó platformot hozzunk létre a sportautó-bérlés piacán Magyarországon.

Reméljük, hogy ez a dokumentáció átfogó képet nyújt a projekt minden aspektusáról.