BMSZC Bláthy Ottó Titusz Informatikai Technikum

1032. Budapest, Bécsi út 134.

1. Záró dolgozat

DevNest

Konzulens tanár: Készítette:

Péter Szabó Richárd Török Máté, Kuhn Kincső, Lugosi Bálint

Tartalom

[1 Bevezetés 3](#_Toc165983406)

[1.1 Feladat leírás 3](#_Toc165983407)

[1.1.1 Általános ismertetés 3](#_Toc165983408)

[2 Felhasználói dokumentáció 5](#_Toc165983409)

[2.1 A program általános specifikációja 5](#_Toc165983410)

[2.2 Rendszerkövetelmények 6](#_Toc165983411)

[2.2.1 Hardver követelmények 6](#_Toc165983412)

[2.2.2 Szoftver követelmények 6](#_Toc165983413)

[2.3 3. A program telepítése 7](#_Toc165983414)

[2.4 A program használatának a részletes leírása 10](#_Toc165983415)

[2.5 GYÍK, FAQ (Gyakran Ismételt Kérdések) 11](#_Toc165983416)

[3 Fejlesztői dokumentáció 12](#_Toc165983417)

[3.1 Az alkalmazott fejlesztői eszközök 12](#_Toc165983418)

[3.1.1 Adatmodell leírása 13](#_Toc165983419)

[3.2 Részletes feladatspecifikáció, algoritmusok 15](#_Toc165983420)

[3.3 Tesztelési dokumentáció 19](#_Toc165983421)

[4 Összegzés 22](#_Toc165983422)

[5 Felhasznált irodalom 25](#_Toc165983423)

# Bevezetés

## Feladat leírás

### Általános ismertetés

A programozási nyelvek tanulása az online térben rendkívül releváns és különösen érdekes téma számomra. Ennek az az oka, hogy az online oktatási lehetőségek rohamosan növekednek és egyre elérhetőbbé válnak az emberek számára világszerte. Ezen túlmenően a programozás egy olyan készség, amelynek mesteri szintű elsajátítása számos előnnyel járhat a mindennapi életben és a szakmai karrierben egyaránt.

Az online programozási nyelv tanulását választottam, mert hiszek abban, hogy a digitális technológia és a számítástechnika alapvető ismeretek, amelyek nélkülözhetetlenek a jövő munkaerőjének számára. Az online térben elérhető tanfolyamok és források lehetővé teszik, hogy bárki, függetlenül attól, hogy hol él vagy milyen háttérrel rendelkezik, hozzáférjen ezekhez az ismeretekhez. Ezáltal az online tanulás lehetővé teszi a rugalmas tanulási módot, amely alkalmazkodik az egyéni időbeosztáshoz és tempóhoz.

Ezenkívül az online programozási nyelv tanulása lehetővé teszi az interaktív tanulási élményeket, amelyek gyakorlati készségeket és problémamegoldó képességeket fejlesztenek ki. Az online kurzusok gyakran tartalmaznak valós projekteket és feladatokat, amelyek segítenek a tanulóknak alkalmazni az elméleti tudást a valós életből vett helyzetekben. Emellett az online közösségek és fórumok lehetővé teszik a tanulók számára, hogy egymást segítsék, együtt dolgozzanak és inspirálják egymást.

Végül, az online programozási nyelv tanulásának választása azért is fontos számomra, mert hozzájárul a személyes és szakmai fejlődésemhez. A digitális készségek és a programozás iránti tudás és érdeklődés növelése nemcsak segít abban, hogy versenyképesebb legyek a munkaerőpiacon, hanem lehetőséget teremt arra is, hogy innovatív projektekbe és kezdeményezésekbe keveredjek bele.

Összességében tehát az online programozási nyelv tanulása egy olyan téma, amely kiterjed a digitális tanulás, a szakmai fejlődés és az innováció területeire, és személy szerint is érdekel, hogy hogyan lehet kihasználni a digitális technológia nyújtotta lehetőségeket a tudás megszerzésére és a személyes fejlődésre.

# Felhasználói dokumentáció

## A program általános specifikációja

A projekt egy online tanulási platform, ahol rugalmasan tanulhatsz különböző programozási és fejlesztői nyelveket. Amint az oldalra lépsz, észre fogod venni, hogy be kell jelentkezned vagy regisztrálnod kell. A bejelentkezés két típusú felhasználót különböztet meg: tanulók és oktatók. Ez a megkülönböztetés a háttérben történik, így a diákok nem tudnak oktatóként belépni, és fordítva.

De mi is a különbség a két típusú felhasználó között? Az oktatók létrehozhatnak iskolákat, kurzusokat, regisztrálhatnak diákokat és más oktatókat is. A tanulók pedig választhatnak, hogy melyik nyelvet szeretnék tanulni, és ez a választás megjelenik a profiljukon, így mindig nyomon követhetik, hol tartanak az adott nyelv tanulásában. Emellett módosíthatják személyes adataikat, és egy naptárban láthatják, hogy az adott nap tanultak-e vagy sem.

A kurzusok három szinten épülnek fel: kezdő, középhaladó és haladó. Mindegyik szint egyre nehezebb tananyagot és kérdéseket kínál. Általában 2-4 oldal tananyagot követ egy ellenőrző kérdéssor, amelyet legalább 60%-os eredménnyel kell teljesíteni ahhoz, hogy továbbléphess a következő szintre. Ha úgy érzed, hogy egy adott nyelv nem annyira tetszik, bármikor újrakezdheted azt, és a régi nyelvekben elért teljesítményed el van mentve a profilodban, így nem kell újra kezdened, ha már egyszer megcsináltad.

A projekt elindítása előtt fontos megérteni és megfelelően felkészülni a rendszerkövetelményekre, hogy biztosítsuk az optimális működést és teljesítményt. A következőkben részletesen áttekintem a rendszerkövetelményeket a projekt szempontjából:

## Rendszerkövetelmények

### Hardver követelmények

* Szerver:

Megfelelő processzor teljesítmény: Ajánlott legalább Quad-core vagy annál magasabb.

Memória (RAM): Minimum 4 GB, azonban a többi felhasználó és a növekvő adatbázis mérete miatt ajánlott legalább 8 GB.

Tárhely: Legalább 50 GB SSD tárhely ajánlott az adatbázis és a fájlok tárolásához.

* Adatbázis szerver:

A projekt adatbázisához valamilyen relációs adatbázis-szerver szükséges, például MySQL, PostgreSQL vagy SQL Server.

Megfelelő processzor és memória: Az adatbázis méretétől és a lekérések gyakoriságától függően ajánlott erős hardver.

### Szoftver követelmények

Operációs rendszer:

A projektet hostoló szerveren futó operációs rendszer támogatása szükséges. Általában Linux (például Ubuntu), Windows vagy más Unix-alapú rendszerek használhatók.

Web szerver:

Apache vagy Nginx web szerver ajánlott. A Laravel tökéletesen működik mindkettőn.

PHP támogatás: Legalább PHP 7.4 vagy annál újabb verzió szükséges, de a legfrissebb PHP verziót javasolt használni a biztonsági és teljesítménybeli javulások érdekében.

Adatbázis szoftver:

MySQL, PostgreSQL vagy SQL Server adatbázis-kezelő szoftver szükséges, attól függően, hogy melyiket preferálod vagy melyik áll rendelkezésre a szerveren.

Egyéb szükséges szoftverek:

Composer: A Laravel projekt függőségeinek kezelésére.

Node.js és npm: Frontend fájlok (pl. JavaScript, CSS) kezelésére, például Laravel Mix használatához.

Git: Verziókezeléshez és fejlesztési folyamat könnyebb kezeléséhez.

## 3. A program telepítése

Egy Laravel projekt elindításához számos eszközre van szükség, például egy webszerverre, adatbázis-szerverre, valamint a Laravel projekt függőségeinek kezelésére szolgáló eszközökre. A következőkben részletesen át fogom tekinteni, hogyan lehet ezeket az eszközöket telepíteni, és hogyan lehet elindítani egy Laravel projektet.

Telepítés lépésről lépésre:

Xampp:

1. Nyisd meg a [XAMPP letöltési oldalát](https://www.apachefriends.org/download.html) a böngésződben.
2. Válaszd ki a megfelelő operációs rendszert:
3. A letöltési oldalon található legördülő menüben válaszd ki az operációs rendszerednek megfelelő verziót. Például Windows, macOS vagy Linux.
4. Kattints a megfelelő verzió mellett található "Download" vagy "Letöltés" gombra a letöltés megkezdéséhez.
5. A böngésződ letölti a XAMPP telepítőjét. Ez néhány percet igénybe vehet attól függően, hogy milyen gyors a kapcsolatod.
6. Miután letöltődött, kattints rá a letöltött fájlra, hogy elindítsd a telepítőt. A Windows-on ez egy .exe fájl lehet.
7. Indítsd el a telepítőt a letöltés után. Ha Windows-ot használsz, valószínűleg megjelenik egy ablak, amely engedélyt kér a telepítéshez. Kattints az "Igen" vagy "Futtatás" gombra.
8. Kövesd az útmutatót:
9. A telepítő egy egyszerű lépéseken keresztül vezet végig. Általában elfogadhatod az alapértelmezett beállításokat. A telepítő felajánlja a telepítést a C:\xampp mappába, de ha szeretnél, változtathatsz a telepítési helyen.
10. Miután befejeződött a telepítés, a XAMPP készen áll a használatra. A telepítő általában egy lehetőséget is kínál a XAMPP indítására.
11. Indítsd el a XAMPP vezérlőpultját:
12. Miután telepítetted a XAMPP-ot, elindíthatod a vezérlőpultot, ahol bekapcsolhatod az Apache és MySQL szervereket. Keress a Start menüben vagy az alkalmazások között a XAMPP mappában, majd indítsd el a "XAMPP Control Panel" nevű alkalmazást.
13. Ellenőrizd az Apache és MySQL szerverek státuszát:
14. A XAMPP vezérlőpulton látni fogod az Apache és MySQL szervereket. Győződj meg róla, hogy mindkettő zöld színű (vagyis fut).

Composer:

Nyisd meg a [Composer letöltési oldalát](https://getcomposer.org/download/) a böngésződben.

A letöltési oldalon találsz két módot a Composer telepítéséhez: Windows-hoz vagy macOS/Linux-hoz. Válaszd ki az operációs rendszerednek megfelelő lehetőséget.

Kattints a megfelelő linkre vagy gombra a letöltés megkezdéséhez. Például, ha Windows-t használsz, kattints a "Windows Installer" linkre.

Miután letöltődött a telepítő fájl, kattints rá annak elindításához. A böngésződ letöltések mappájában találhatod.

A telepítő elindítása után a telepítő varázsló vezet végig a telepítési folyamaton. Az alapértelmezett beállítások elfogadása általában megfelelő.

Várj, amíg a telepítés befejeződik. A Composer telepítése után a rendszered már használhatja a Composer parancssoros interfészét.

Miután a telepítés befejeződött, nyiss egy parancssort (a Windows-on például a Command Prompt-ot vagy PowerShell-t) és futtasd a composer --version parancsot. Ha a telepítés sikeres volt, meg kell jelenítenie a Composer verzióját.

Node.js-t és npm-et:

Nyisd meg a [Node.js letöltési oldalát](https://nodejs.org/en/download/) a böngésződben.

A letöltési oldalon találsz különböző verziókat a Node.js-hez. Az ajánlott verzió a LTS (Long Term Support) verzió, mivel ez a stabil verzió és hosszú távú támogatást kap. Válassz ki egy megfelelő verziót az operációs rendszerednek megfelelően.

Kattints a letöltési gombra a kiválasztott verzió mellett. A letöltés elindul, és a böngésződ letöltések mappájába kerül a telepítő fájl.

Miután letöltődött a telepítő fájl, kattints rá annak elindításához. A telepítő varázsló megjelenik.

Kövesd a telepítő varázsló lépéseit. Az alapértelmezett beállítások általában megfelelőek.

Miután a telepítés befejeződött, nyiss egy parancssort (a Windows-on például a Command Prompt-ot vagy PowerShell-t) és futtasd a következő parancsokat:

node --version

npm --version

Ezek a parancsok megjelenítik a telepített Node.js és npm verzióit. Ha megjelenik a verziószám, akkor sikeresen telepítetted őket.

A Laravel projekt mappájában futtasd a composer install parancsot, hogy a Composer telepítse a projekt függőségeit.

Állítsd be a .env fájlt:

Másold a .env.example fájlt .env-re: cp .env.example .env.

A xamppon nyomj ra a mysql sorban az admin gombra vagy meny rá a <http://localhost/phpmyadmin/> oldalra

Hozz létre egy adatbazist bármilyen néven

Állítsd be az adatbázis kapcsolatot és egyéb konfigurációs beállításokat a .env fájlban.

Tölsd le a hozzáadott könyvtárakat:

Futtasd az npm install parancsot a node könyvtárak letöltéséhez.

Generálj alkalmazáskulcsot:

Futtasd a php artisan key:generate parancsot a Laravel alkalmazáskulcs generálásához.

Migrációk futtatása:

Futtasd a php artisan migrate:fresh --seed parancsot az adatbázis táblák migrálásához.

Szerver indítása:

Futtasd a php artisan serve parancsot a beépített fejlesztői szerver indításához majd egy másik terminálba az npm run dev parancsot.

Készen is vagyunk a <http://127.0.0.1:8000/> oldalon készen vár a projekt.

## A program használatának a részletes leírása

Regisztráció és Bejelentkezés

Az Online Tanulási Rendszer használatához először regisztrálnod kell, vagy ha már rendelkezel fiókkal, be kell jelentkezned.

Regisztráció:

Látogasd meg az oldalunkat és kattints a "Regisztráció" gombra.

Töltsd ki az űrlapot az alapvető adatokkal, mint például a felhasználónév, e-mail cím és jelszó.

Kapsz egy emailt a sikeres regisztrációdhoz és beléptetünk a rendszerbe.

Bejelentkezés:

Ha már rendelkezel fiókkal, egyszerűen kattints a "Bejelentkezés Tanulóként" gombra.

Add meg az email címedet és jelszavak, majd kattints a "Bejelentkezés" gombra.

Elfelejtett jelszó estén a bejelentkezés gomb után a „Forgot password” gombra kattintva majd az email címed megadása után kapsz egy emailt, amibe egy linkre kattintva megváltoztathatod a jelszavad.

2. Fiók Kezelése

Miután sikeresen bejelentkeztél, elérheted a felhasználói profilod, ahol számos lehetőség áll rendelkezésedre.

Profil szerkesztése:

A "Profil" menüpont alatt szerkesztheted a személyes adataidat, mint például a neved, e-mail címed vagy felhasználói neved.

Itt láthatod, hogy milyen nyelvekbe kezdtél bele és azokba milyen eredmény/szintet értél el.

3. Kurzusok és Tanulás

Az Online Tanulási Rendszerben számos kurzus áll rendelkezésedre, amelyek segítségével fejlesztheted programozási képességeidet.

Kurzusok böngészése:

Keresd meg a számodra érdekes kurzusokat a "Learn" menüpont alatt.

Nézd meg a kurzusok leírását és válaszd ki azt, amelyik érdekel.

Tanulási folyamat:

A kurzusok három szinten állnak rendelkezésedre: kezdő, középhaladó és haladó.

Minden szint után kvíz teszteli a megszerzett tudásodat.

4. Tanulási Napló és Naptár

A rendszer lehetővé teszi számodra, hogy nyomon kövesd a tanulási előrehaladásodat.

Ha sikeresen teljesitesz egy tesztet akkor a calendarba zoldel megjelenik az adott kurzus nevével egy sor. Ha sikertelen akkor sárgával. Bármennyi tesztet csinálhatsz egy nap.

Nyelvváltás és újrakezdés:

Bármikor megváltoztathatod a tanulási nyelvet vagy újrakezdheted a tanulást egy másik nyelvvel.

Mentett Előrehaladás:

A rendszer automatikusan menti a teljesített kurzusokat és eredményeket, így nem veszíted el a tanulmányaidat.

## GYÍK, FAQ (Gyakran Ismételt Kérdések)

Hányszor lehet egy tesztet újra próbálni?

Mi történik, ha minden szintet teljesítek?

Mi történik, ha befejeztem minden kurzust?

# Fejlesztői dokumentáció

## Az alkalmazott fejlesztői eszközök

A program fejlesztése során számos különböző szoftveres eszközt használtam, hogy hatékonyan tudjam megvalósítani a tervezett funkcionalitásokat és létrehozzam a dokumentációt. Az alábbiakban felsorolom ezeket az eszközöket, valamint röviden ismertetem, hogy milyen célokra használtam őket:

Laravel: A program fő keretrendszere, amelynek segítségével az alkalmazás logikai szerkezetét és adatkezelését megvalósítottam. A Laravel a PHP egyik kiváló keretrendszere, amely gyors fejlesztést tesz lehetővé és számos beépített funkcióval rendelkezik, mint például az Eloquent ORM.

PHP: Az alkalmazás logikai részének megvalósításához használtam. A PHP egy széles körben elfogadott szkriptnyelv a webfejlesztés területén, amely lehetővé teszi a dinamikus weboldalak készítését.

JavaScript: A felhasználói felület interaktivitásának és dinamizmusának biztosításához használtam JavaScriptet. Ez a nyelv elengedhetetlen a modern webalkalmazások kliensoldali logikájának implementálásához.

Blade: A Laravel beépített sablonmotorja, amely lehetővé teszi a dinamikus tartalom készítését. A Blade segítségével könnyen és hatékonyan generálhattam HTML-t a PHP kódjaimból.

HTML és CSS: Az alkalmazás felhasználói felületének kialakításához és stílusozásához használtam HTML-t és CSS-t. Az HTML a weboldalak strukturális elemeinek megadására szolgál, míg a CSS a megjelenésükért felel.

XAMPP: Egy könnyen telepíthető, többplatformos szerverkörnyezet, amely tartalmazza az Apache HTTP szervert, a MySQL adatbázis-kezelőt és az OpenSSL-t. XAMPP-t használtam a fejlesztési környezet létrehozásához és az alkalmazásom helyi teszteléséhez.

MySQL: Az adatbázis-kezelő rendszer, amelyben az alkalmazásom adatbázisait tároltam. A MySQL nagyon elterjedt és jól dokumentált relációs adatbázis-kezelő rendszer, amely hatékony adatkezelést tesz lehetővé.

jQuery: Egy könnyű és gyors JavaScript könyvtár, amely segít az általános feladatok, mint például az eseménykezelés vagy az AJAX hívások megvalósításában. jQuery-t használtam az alkalmazásomhoz tartozó dinamikus funkciók fejlesztése során.

DataTables: Egy jQuery alapú plugin, amely lehetővé teszi a HTML táblázatok kényelmes és interaktív működését. DataTables-t használtam az adatok tabuláris formában történő megjelenítéséhez és kezeléséhez az alkalmazásban.

FullCalendar: Egy JavaScript alapú naptárkomponens, amely lehetővé teszi a felhasználók számára az események kezelését és megjelenítését naptári formában. FullCalendar-t használtam az alkalmazásom naptár funkciójának megvalósításához.

Az általam használt modulok és eszközök jogszerű felhasználása rendkívül fontos szempont a szoftverfejlesztés során. Minden eszköz és modul esetében gondoskodtam arról, hogy megfelelően licencezzem vagy engedélyezzem a használatukat, és betartottam azokat a felhasználási feltételeket, amelyeket a készítőik előírtak. A legtöbb esetben a modulokhoz és eszközökhöz tartozó dokumentációban részletesen le van írva, hogy milyen módon lehet őket használni, és milyen feltételekkel. Ennek alapján igyekeztem biztosítani, hogy minden felhasznált komponens teljes mértékben jogszerűen legyen beépítve és használva az alkalmazásban.

### Adatmodell leírása

Az alkalmazás adatbázisát MySQL segítségével hoztuk létre, amelynek struktúráját és kapcsolatait alább részletezem. A táblák és kapcsolatok leírása mellett egy szöveges leírást is biztosítok az adatbázis fő adatszerkezeteiről.

Adattáblák és kapcsolatok:

users: Ez a tábla tartalmazza az alkalmazás felhasználóinak adatait, mint például a felhasználónév, jelszó, email cím stb.

calendar: A naptárt reprezentáló tábla, amelyben tároljuk az egyes felhasználókhoz rendelt eseményeket. A tábla tartalmazza az események id-jét, nevét, kezdési és befejezési időpontját, valamint a hozzájuk kapcsolódó felhasználó azonosítóját.

courses: Az oktatási kurzusokat tároló tábla, amelyben rögzítjük a kurzusok nevét, leírását, típusát és az oktató iskoláját.

course\_data: A kurzusokhoz kapcsolódó adatok tárolására szolgáló tábla, amely tartalmazza a kurzusokhoz tartozó további információkat.

course\_quizzes: A kurzusokhoz kapcsolódó tesztek tárolására szolgáló tábla, amelyben rögzítjük a tesztek nevét, leírását és a hozzájuk tartozó kurzus azonosítóját.

course\_user: A kurzusok és felhasználók közötti kapcsolatot tároló tábla. Ebben rögzítjük, hogy mely felhasználó mely kurzusokhoz van hozzárendelve.

prog\_langs: A programozási nyelveket tároló tábla, amelyben a különböző programozási nyelvek neveit és típusait jegyezzük fel.

schools: Az oktatási intézményeket tároló tábla, amelyben rögzítjük az egyes iskolák nevét és egyéb releváns információkat.

types: Az adatbázisban használt különböző típusokat tároló tábla, például a kurzusok típusát.

Kapcsolatok:

A calendar tábla és a users tábla között egy-egy kapcsolat áll fenn, a felhasználókhoz rendelt eseményeket az azonosító alapján kötjük össze.

A courses és types táblák között egy-több kapcsolat áll fenn, egy típus több kurzushoz is tartozhat.

A course\_data, course\_quizzes, prog\_langs és schools táblák egy-több kapcsolatban állnak a courses táblával, egy kurzushoz több adat, teszt, programozási nyelv és iskola is tartozhat.

Szöveges leírás:

Az adatbázis fő célja az oktatási kurzusok és az azokhoz kapcsolódó események és információk kezelése, valamint az alkalmazás felhasználóinak azonosítása és a felhasználókhoz kapcsolódó adatok tárolása. A felhasználók azonosítása és a kurzusokhoz való hozzáférés biztosítása érdekében a users és courses táblák között kapcsolat áll fenn, valamint a course\_user tábla segítségével tároljuk az egyes felhasználókhoz rendelt kurzusokat. Az események kezelése a calendar táblában történik, amely tartalmazza az eseményeket és a hozzájuk tartozó felhasználókat. Az egyes kurzusokhoz kapcsolódó további információkat, teszteket, programozási nyelveket és iskolákat a megfelelő táblákban tároljuk, és ezeket kapcsolatokkal kötjük össze a courses táblával. Ezáltal az adatbázis struktúrája támogatja az alkalmazás fő funkcionalitásait és lehetővé teszi a szükséges adatok hatékony kezelését.

## Részletes feladatspecifikáció, algoritmusok

A show metódus felelős az egy adott kurzus megjelenítéséért az alkalmazásban. A következőkben részletesen leírom a metódus működését és paramétereit, valamint a visszatérési értékeket.

Metódus: show

Paraméterek:

$course: Egy Course objektum, amely az adott kurzust reprezentálja.

Visszatérési értékek:

A metódus HTML nézetet ad vissza, amely tartalmazza az adott kurzus adatait és esetleges további információkat.

Működés:

A metódus kezdetén ellenőrzi, hogy van-e $\_GET változó, ha igen, akkor vizsgálja annak értékét.

Ha a $\_GET['sikerult'] változó értéke "false", akkor a felhasználónak nem sikerült valamilyen tevékenység, ezért egy új eseményt hoz létre a calendar táblában a DB::table("calendar")->insert() metódussal. Az esemény címe a kurzus neve, a kezdési és befejezési időpontja pedig az aktuális dátum.

Ha a $\_GET['sikerult'] változó értéke nem "false", akkor további lépések következnek.

Ellenőrzi, hogy van-e $\_GET["elozocourse"] változó. Ha van, akkor a $\_GET["elozocourse"] értéke alapján lekérdezi az előző kurzus adatait a courses táblából, és létrehoz egy új eseményt a calendar táblában az előző kurzus nevével és zöld háttérszínnel.

Ha van $\_GET['level'] és $\_GET['basecourseid'] változó, akkor a course\_user táblában frissíti a felhasználó kurzuson belüli szintjét (completed) a megadott értékkel.

Ellenőrzi, hogy a kurzus útvonala (c\_route) benne van-e a megadott croutearray tömbben. Ha nincs, akkor visszaadja a kurzus megjelenítésére szolgáló HTML nézetet.

Ellenőrzi, hogy a felhasználó már regisztrált-e a kurzusra a course\_user táblában. Ha nem, akkor új rekordot szúr be a course\_user táblába a felhasználóhoz és a kurzushoz tartozó adatokkal, majd növeli a kurzust választó felhasználók számát a prog\_langs táblában.

Ha a felhasználó már regisztrált a kurzusra, akkor ellenőrzi, hogy milyen szinten van a kurzuson. Attól függően, hogy a kurzus szintje, a megfelelő HTML nézetet adja vissza.

Végül, ha a kurzus útvonala benne van a megadott croutearray tömbben, akkor a megfelelő HTML nézetet adja vissza a kurzus megjelenítésére.

Metódus: checkInputData

Paraméterek:

Nincsenek bemeneti paraméterek a metódusnak.

Visszatérési értékek:

A metódus egy JSON választ ad vissza, amely tartalmaz egy takenornot nevű boolean értéket vagy hibaüzenetet, attól függően, hogy a bemeneti adatok megfelelnek-e a követelményeknek.

Működés:

Az első lépés a bemeneti adat lekérése a $\_POST globális tömbből. A $inputValue változóban tárolódik az átküldött adat.

Lekérdezi az összes felhasználót az adatbázisból a User::all() metódussal, és eltárolja őket a $users változóban.

Az alapértelmezett érték a takenornot változóban false. Ez az érték azt fogja jelenteni, hogy a felhasználónév még nem foglalt.

Ellenőrzi, hogy a bemeneti adat nem üres és legalább 5 karakter hosszú-e.

Ha a feltételek teljesülnek, akkor végigmegy a felhasználók listáján a foreach ciklus segítségével. Ha talál olyan felhasználót, akinek a felhasználóneve megegyezik az átküldött értékkel, akkor a takenornot változó értékét true-ra állítja és megszakítja a ciklust (break parancs).

Ha a felhasználónév már foglalt, akkor a metódus egy JSON választ ad vissza a takenornot értékével, hogy true-ra állítja.

Ha a bemeneti adat üres, vagy rövidebb, mint 5 karakter, akkor egy hibaüzenetet ad vissza JSON formátumban, amely jelzi, hogy a felhasználónév túl rövid.

Végül a metódus visszaadja a JSON választ a klienstől érkezett kéréssel kapcsolatban.

Ez a metódus felelős a felhasználók által megadott felhasználónevek egyediségének ellenőrzéséért az alkalmazásban. Ellenőrzi, hogy a megadott felhasználónév már foglalt-e vagy sem, és a megfelelő választ adja vissza a kliensnek JSON formátumban.

Útvonal: /calendar

Metódus: GET

Paraméterek:

Nincsenek bemeneti paraméterek a metódusnak.

Visszatérési értékek:

Az útvonal egy HTML nézetet ad vissza, amely tartalmazza a felhasználó naptárának adatait.

Működés:

Az útvonalhoz tartozó anonim függvényben lekéri az aktuális felhasználó azonosítóját a Auth::user()->id segítségével, majd eltárolja a $userid változóban.

Lekéri a felhasználóhoz tartozó naptáradatokat a calendar táblából a felhasználó azonosítója alapján a DB::table('calendar')->where("user\_id", $userid)->get() lekérdezéssel. Az eredményt eltárolja a $calendardata változóban.

Visszaadja a calendar/index nézetet a $calendardata változóval egy asszociatív tömbben átadva, hogy a nézet hozzáférjen a naptár adatokhoz.

A name('calendar') metódussal nevet ad az útvonalnak, amely lehetővé teszi az útvonal elérését a kód más részeiből név alapján.

Ez az útvonal felelős a felhasználó naptárának megjelenítéséért az alkalmazásban. Lekéri a felhasználóhoz tartozó naptáradatokat az adatbázisból, majd egy HTML nézetet ad vissza, amely tartalmazza ezeket az adatokat a felhasználó számára.

Ez a nézet felelős a felhasználó naptárának megjelenítéséért az alkalmazásban. A naptár megjelenítéséhez a FullCalendar könyvtárat használja.

A calendar/index nézetben egy div elem van, amely az id attribútummal rendelkezik és calendar értéket kap. Ebben a div elemben jelenik meg a naptár.

Az inline script részben inicializálódik a FullCalendar, és a naptár megjelenítéséhez használja a kapott $calendardata változót. Ez a változó tartalmazza a felhasználóhoz tartozó naptáradatokat, amelyeket előzetesen JSON formátumba alakítottunk.

## Tesztelési dokumentáció

1. Felhasználói Regisztráció Teszteset:

Leírás:

Ebben a tesztesetben egy új felhasználó regisztrál az online tanulási rendszerbe.

Teszteset:

Felhasználó kattint a "Regisztráció" gombra az otthoni oldalon.

Kitölti az űrlapot a szükséges adatokkal (felhasználónév, e-mail, jelszó stb.).

Megnyomja a "Regisztráció" gombot.

Ellenőrzi az e-mail fiókját a regisztráció megerősítéséhez kapott üzenetért.

Várható Eredmények:

Sikeres regisztráció esetén a felhasználó bejelentkeztetésre kerül a rendszerbe.

A felhasználó kap egy üzenetet a jobb felső sarokban egy felrúgó ablakon.

Hibák:

Hibás adatok megadása esetén megjelenik egy figyelmeztető üzenet.

2. Tanuló Bejelentkezés Teszteset:

Leírás:

Ebben a tesztesetben egy tanuló bejelentkezik az online tanulási rendszerbe.

Teszteset:

A felhasználó megadja az email címét és jelszavát a bejelentkezési oldalon.

Megnyomja a "Bejelentkezés" gombot.

Várható Eredmények:

Sikeres bejelentkezés esetén a felhasználó bejelentkezett állapotba kerül és egy felugró ablakba kiírja a felhasználó nevét és hogy sikeresen bejelentkezett.

Hibák:

Hibás email vagy jelszó esetén hibaüzenet jelenik meg.

Ha a rendszer nem elérhető, akkor megjelenik egy általános hibaüzenet.

3. Kurzus Kvíz Teszteset:

Leírás:

Ebben a tesztesetben egy tanuló részt vesz egy kurzus kvízen.

Teszteset:

A tanuló kiválaszt egy kurzust a kurzusok közül.

Elolvassa a tananyagot és eljut a kvízig.

Kitölti a kvíz kérdéseit és megnyomja az "Ellenőrzés" gombot.

Várható Eredmények:

Ha a tanuló elért legalább 60%-ot a kvízen, akkor egy zöld üzenet jelenik meg ami a következő szintre irányit.

Ha nem érte el a 60%-ot, akkor megjelenik egy piros üzenet ami újra kezdi a kurzust.

Hibák:

Nem tölti ki az összes tesztkérdést akkor hiba üzenetbe visszajelzést kap.

Ha a rendszer nem tudja menteni az eredményeket, akkor megjelenik egy általános hibaüzenet.

Hibák Felsorolása:

Az adatok validációja során nem kezeli megfelelően a hibás adatokat.

A regisztráció megerősítésekor az e-mail nem mindig érkezik meg.

Néhány esetben a rendszer instabil, és nem minden esetben menti el az eredményeket.

Tesztelési Módszerek:

A fentebb leírt teszteseteket fehér doboz és fekete doboz módszerrel is teszteltük. A fehér doboz módszer során az alkalmazás belső működését és struktúráját vizsgáltuk, míg a fekete doboz módszer során a felhasználói interfészt és funkciókat teszteltük anélkül, hogy a belső működésbe mélyedtünk volna.

# Összegzés

1. További Interaktív Funkciók:

Hanganyagok: Az egyes tananyagokhoz hanganyagokat lehetne kapcsolni, hogy még inkább megkönnyítsük a tanulók számára a tanulást.

Kód Részletek: A kurzusokban strukturáltan megjeleníthetnénk a kód részleteket, hogy a tanulók könnyebben követhessék az oktatást és gyakorolhassák a kódírást.

2. Bővített Képességek és Tanulási Tartalom:

Haladó Kurzusok: Az alap- és középhaladó szintű kurzusok mellett haladó szintű kurzusokat is be lehetne vezetni a rendszerbe, hogy még több lehetőséget nyújtsunk a tanulóknak.

Új Programozási Nyelvek: Folyamatosan bővíthetnénk a rendelkezésre álló programozási nyelvek listáját a piaci igényeknek megfelelően.

3. Felhasználói Élmény Fokozása:

Jobb Navigáció és Felhasználói Interfész: Folyamatosan fejleszthetjük a felhasználói felületet és a navigációt, hogy még egyszerűbben és kényelmesebben használhassák a felhasználók a rendszert.

Személyre Szabott Ajánlások: Algoritmusok segítségével személyre szabott kurzusajánlásokat kínálhatunk a felhasználóknak a tanulási igényeiknek megfelelően.

4. Piaci Trendek és Elvárások:

Mobilitás: Figyelembe kell venni a mobil eszközökön való használatot és egy esetleges mobilalkalmazás fejlesztését is.

Adatvédelem és Biztonság: Fontos az adatvédelem és a felhasználók biztonsága, ezért folyamatosan figyelnünk kell az adatvédelmi szabályozásokra és azoknak megfelelően kell alakítani a rendszert.

5. Tanulási Teljesítmény Elemzése:

Felhasználói Statisztikák: Bővíthetjük a rendszert olyan funkciókkal, amelyek lehetővé teszik a felhasználók számára, hogy nyomon kövessék saját tanulási teljesítményüket és fejlődésüket.

Tanári Visszajelzés: Képzők és oktatók számára lehetőséget biztosíthatunk a tanulók teljesítményének értékelésére és visszajelzésére, hogy még hatékonyabb legyen a tanulási folyamat.

6. Kiterjesztett Valóság (AR) és Virtuális Valóság (VR) Támogatás:

Interaktív Tanulási Élmény: AR és VR technológiák segítségével még interaktívabbá tehetjük a tanulási élményt, például virtuális laboratóriumokkal vagy kiterjesztett valósággal támogatott kódolási gyakorlatokkal.

7. Közösségi Funkciók:

Tanulói Fórumok: Létrehozhatunk tanulói fórumokat, ahol a felhasználók megoszthatják tapasztalataikat, kérdéseket tehetnek fel és egymást segíthetik a tanulás során.

Közös Projekt lehetőség: Lehetőséget adhatunk a felhasználóknak, hogy közösen dolgozzanak projektjeiken, így gyakorolva a valós életbeli együttműködést és fejlesztve a problémamegoldó képességeiket.

8. Adaptív Tanulási Algoritmusok:

Személyre Szabott Tanulási Terv: Az adaptív tanulási algoritmusok segítségével személyre szabott tanulási tervet kínálhatunk a felhasználóknak, figyelembe véve az egyéni tanulási stílusokat és erősségeket.

9. Piaci Versenyképesség és Eladhatóság:

Innováció és Kreativitás: Folyamatosan figyelemmel kell kísérnünk a piaci trendeket és az ügyfél visszajelzéseket annak érdekében, hogy innovatív funkciókat és tartalmakat fejlesszünk ki, amelyek megkülönböztetnek minket a versenytársaktól.

Felhasználói Elégedettség és Retenció: A felhasználói élmény javítása és a rendszer folyamatos fejlesztése kulcsfontosságú a felhasználók elégedettségének és hosszú távú elkötelezettségének biztosításához.

# Felhasznált irodalom

Elektronikus Dokumentum:

OpenAI. "GPT-3.5 Model Documentation". Letöltve: OpenAI hivatalos weboldaláról, https://openai.com/gpt-3/.