

**SAPIENTIA ERDÉLYI MAGYAR TUDOMÁNYEGYETEM
MAROSVÁSÁRHELYI KAR,
INFORMATIKA SZAK**



SAPIENTIA
ERDÉLYI MAGYAR
TUDOMÁNYEGYETEM

Lustra avagy tudományos folyóiratok értékelése kategóriái
alapján

DIPLOMADOLGOZAT

Témavezető:
Farkas Csaba,
Egyetemi tanár

Végzős hallgató:
Fekete Dániel

2023

UNIVERSITATEA SAPIENTIA DIN CLUJ-NAPOCA
FACULTATEA DE ȘTIINȚE TEHNICE ȘI UMANISTE,
SPECIALIZAREA INFORMATICĂ



UNIVERSITATEA
SAPIENTIA

Evaluare revistelor științifice în funcții de categorie

LUCRARE DE DIPLOMĂ

Coordonator științific:
Farkas Csaba,
Profesor universitar

Absolvent:
Fekete Dániel

2023

**SAPIENTIA HUNGARIAN UNIVERSITY OF
TRANSYLVANIA
FACULTY OF TECHNICAL AND HUMAN SCIENCES
COMPUTER SCIENCE SPECIALIZATION**



SAPIENTIA
HUNGARIAN UNIVERSITY
OF TRANSYLVANIA

Scientific journal rankings according to categories

BACHELOR THESIS

Scientific advisor:
Farkas Csaba,
Full Professor

Student:
Fekete Dániel

2023

UNIVERSITATEA „SAPIENTIA” din CLUJ-NAPOCA
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș
Programul de studii: Informatică

Viza facultății:

LUCRARE DE DIPLOMĂ

Coordonator științific:
Dr. Farkas Csaba

Candidat: Fekete Dániel
Anul absolvirii: 2022

a) Tema lucrării de licență:

Evaluarea revistelor științifice în funcție de categorie (Scimago)

b) Problemele principale tratate:

În lucrare, scopul a fost implementarea unui site web responsabil cu clasificarea diferitelor reviste științifice. Pentru aceasta, s-au utilizat limbajele de programare PHP și JavaScript. Lucrările științifice sunt salvate într-o bază de date dedicată, cu care backend-ul site-ului comunică.

c) Desene obligatorii:

- UML Use Case Diagram
- Grafice despre structura codului
- Desene despre design-ul aplicației

d) Softuri obligatorii:

Un software este implementat ca un site web în PHP, în care utilizatorul poate căuta reviste, poate vizualiza datele ale revistelor și poate lista revistele în funcție de categorii unice. În plus, există posibilitatea de a importa lista actuală în sistem în fiecare an.

e) Bibliografia recomandată:

[1] Falagas, M. E., Kouranos, V. D., Arencibia-Jorge, R., & Karageorgopoulos, D. E. (2008). Comparison of SCImago journal rank indicator with journal impact factor. *The FASEB journal*, 22(8), 2623-2628.

[2] Dr. Hornyacsek Júlia. *A tudományos kutatás elmélete és módszertana. Nemzeti Közzolgálati Egyetem Hadtudományi és Honvédtisztképző Kar, Budapest, 2014.*

f) Termene obligatorii de consultații: săptămânal

g) Locul și durata practicii: Universitatea „Sapientia” din Cluj-Napoca,
Facultatea de Științe Tehnice și Umaniste din Târgu Mureș, sala / laboratorul 413

Primit tema la data de: 20.06.2021

Termen de predare: 07.07.2023

Semnătura Director Departament



Semnătura responsabilului
programului de studiu



Semnătura coordonatorului



Semnătura candidatului



Declarație

Subsemnatul/a FEKETE DANIEL, absolvent(ă) al/a
specializării INFORMATICA, promoția 2022...
cunoscând prevederile Legii Educației Naționale 1/2011 și a Codului de etică și deontologie profesională a Universității Sapientia cu privire la furt intelectual declar pe propria răspundere că prezenta lucrare de licență/proiect de diplomă/disertație se bazează pe activitatea personală, cercetarea/proiectarea este efectuată de mine, informațiile și datele preluate din literatura de specialitate sunt citate în mod corespunzător.

Localitatea, TG. MUREȘ
Data: 16.06.2023

Absolvent
Semnătura Fekete

Kivonat

Dolgozatom témája egy felhasználóbarát webhely, amely több weblapon, különböző módszereket alkalmazva kimutatja a keresett folyóiratok rangját kategóriái alapján, felhasználva a SCImago Journal Rank (SJR) adatbázisát és mérőszámait. Ezen kereséseket ugyanakkor, kinyomtathatjuk PDF-formátumban viszont az oldal is elmenti egy íróhoz csatolva, amelyekben tovább keresgélhetünk, összehasonlítva a kereséseket is. Ezen folyóiratokat szerkeszthetjük, pontosabban hozzáadhatunk olyan fontos információkat, amelyeket a felhasznált adatbázis nem tartalmaz, mint például az akadémiai év, amelyben kiadták, vagy a tudományos cikk címe. Mindezek mellett az adatbázisunkban még nem szereplő folyóiratokat hozzá tudjuk adni az adatbázishoz. Az adatok frissességéről egy általam készített bővítmény gondoskodik, amely negyedévente frissíti az adatbázist az online elérhető adatokkal. A webhelyen levő adatok eléréséhez a látogatónak kötelessége regisztrálni ahhoz, hogy tudjon keresni, továbbá új adatokat adhasson hozzá.

Ahhoz, hogy szinten tudjunk maradni a fejlődés gyorsaságával, muszáj jelen lennünk a digitális életben. Ehhez a legrövidebb út a webhely: bárki számára, bárholnan elérhető, és böngészhető.

A továbbiakban bemutatom, lépésről lépésre egy webhely kialakulását. Prezentálom a módszereket, amelyeket alkalmaztam a webhely kivitelezésére. Emellett szó fog esni az adatokról, amelyeket feldolgoztam és használtam a rangsorolás kialakítására, a technológiákról amelyeket felhasználtam ahhoz, hogy meg tudjam valósítani a webhely céljait. Majd bemutatom az értékelési szempontot, amely alapján a webhely kialakítja a tudományos folyóiratok rangsorát. Ezt követően megnézzük, hogy hogyan néz ki a webhely felhasználói-, és fejlesztői szemszögből is.

Végezetül, megnézzük, hogy hogyan válik a webhely mindenki számára elérhetővé.

Rezumat

Tema tezei mele este dezvoltarea unui site web ușor de utilizat, care utilizează metode de afișare a clasamentului revistelor științifice căutate în funcție de categorie, în ordine crescătoare sau descrescătoare. Căutările se bazează pe informațiile din baza de date SCImago Journal Rank (SJR). Scimago Journal Rank este un indicator bibliometric, ce utilizează o metodă complexă de calcul care ia în considerare nu numai numărul de citări primite de un articol sau de o revistă, ci și importanța revistelor care fac aceste citări. Rezultatele căutărilor pot fi tipărite în format PDF, iar pagina poate fi salvată ulterior pentru a compara căutările. Aceste documente pot fi editate ulterior, deoarece nu conțin toate informațiile necesare în baza de date utilizată, cum ar fi anul universitar în care a fost publicat sau titlul articolului științific. Acesta permite adăugarea revistelor care nu se regăsesc în prezent în baza noastră de date. Pentru a asigura actualizarea datelor, folosesc o extensie creată de mine, care este actualizată trimestrial cu cele mai recente informații disponibile online. Pentru a avea permisiunea de a accesa datele de pe site, utilizatorii trebuie să se înregistreze pentru a putea efectua căutări și pentru a adăuga informații noi.

Pentru a fi în pas cu evoluția și a ne adapta la schimbări, este vital să fim prezenți în mediul digital. Cel mai rapid mod pentru a avea acces acest lucru este prin intermediul site-ului nostru web, acesta fiind disponibil pentru oricine, de oriunde, și este conceput pentru a fi ușor de utilizat.

În prima parte a lucrării voi prezenta pas cu pas construirea siteului web, prezentând metodele pe care le-am utilizat pentru dezvoltarea acestuia. Prin urmare, voi face referire și la datele pe care le-am prelucrat și utilizat pentru a crea clasamentul, după care voi include și tehnologiile pe care le-am folosit pentru a atinge obiectivele site-ului. Pe lângă asta, în lucrarea mea voi aminti criteriile de evaluare folosite pentru a dezvolta baza de cunoștințe a site-ului și clasamentul revistelor științifice. Voi analiza aspectul site-ului atât din punctul de vedere al utilizatorului, cât și din perspectiva dezvoltatorului.

În ultima parte a lucrării mele, voi investiga cum site-ul este accesibil și ușor de utilizat pentru toți vizitatorii.

Abstract

In today's rapidly evolving digital landscape, the need to stay connected and informed has become more crucial than ever. The accessibility and usability of websites play a significant role in facilitating efficient information retrieval and knowledge sharing. Therefore, the primary focus of this thesis is to investigate the creation of a website that is user-friendly and intuitive. This website aims to provide extensive rankings of academic journals using the SCImago Journal Rank (SJR) database and its associated metrics. We can edit scientific journals, or more precisely, we can add important information that is not included in the database used, such as the academic year in which it was published, or the title of the scientific article. The search results can be saved as PDF files for convenient access later on. Furthermore, visitors also have the opportunity to contribute to the website's database by adding new journals, thereby helping to expand the range of available information. In order to gain access to the data on the website, visitors are requested to register, enabling them to perform searches and contribute new information.

The primary objective of this thesis is to present a step-by-step guide to building a website that is easily accessible to users worldwide, ensuring a seamless browsing experience. Throughout this research, I will explore a variety of methodologies and techniques implemented during the development phase, offering insights into the underlying technologies that were utilized. Furthermore, this thesis aims to shed light on the evaluation criteria employed to establish the journal ranking system within the website. By analyzing and processing relevant data, the ranking methodology will be delineated, providing users with a reliable and informed basis for their academic endeavors.

Next, I will proceed to outline the assessment standards employed in the construction of the website's knowledge base, specifically pertaining to the ranking of scientific journals. Subsequently, I will examine the website's design and functionality from both the user's and developer's perspectives.

Finally, I will thoroughly investigate the measures implemented to ensure the website's accessibility for all users, regardless of their abilities

Tartalomjegyzék

1. Bevezető	11
2. Programok, technológiák bemutatása	13
2.1. Adatok	13
2.2. Felhasznált technológiák az adatok feldolgozására	13
2.2.1. Excel [Mic16]	13
2.2.2. WP All Import [Sof22]	14
2.2.3. Advanced Custom Fields [Del22]	14
2.3. Felhasznált szoftverek a webhely létrehozására	14
2.3.1. WordPress [Wor22]	14
2.3.2. Enfold Téma [Kri22]	14
2.4. Felhasznált technológiák a rangsorolásra	14
2.4.1. HTML5 [WHA08]	14
2.4.2. SCSS [Nat22]	15
2.4.3. Node-sass [Ope22]	15
2.4.4. PHP [PHP22]	15
2.4.5. JavaScript [Net11]	15
2.4.6. jQuery [jQu22]	15
2.4.7. jsPDF [Par22]	16
2.4.8. PHP Storm [Jet21]	16
2.4.9. GitHub	16
2.5. Felhasznált egyéb technológiák	16
2.5.1. FacetWP [Fac21]	16
2.5.2. Ultimate Member [Ult23]	16
3. Rendszer specifikációja	17
3.1. Rendszerkövetelmények	17
3.1.1. Funkcionális követelmények	17
3.1.2. Nem funkcionális követelmények	18
4. Webhely szerkezete	19
4.1. Webhely backend	19
4.2. Webhely frontend	20
5. Folyóiratok értékelése	25
5.1. Értékelési rendszer	25

6. A szoftver bemutatása	27
6.0.1. Az űrlapok létrehozása	27
6.1. ISSN szerinti keresés	28
6.1.1. jQuery	28
6.1.2. PHP	32
6.1.3. Folyóirat rangjának megtalálása kategóriánként	34
6.2. További funkcionalitások	35
6.3. PDF generálása jsPDF segítségével	37
6.4. Szoftver befejezése	37
6.5. Telepítés a webszerverre	38
Összefoglaló	39
Ábrák jegyzéke	40
Táblázatok jegyzéke	41
Irodalomjegyzék	43
Függelék	44
F.1. A PHP Storm felülete	44
F.2. A WordPress felülete	45

1. fejezet

Bevezető

Napjainkban a tudományos fejlődés fénykorát éljük, nap mint nap újabb és újabb tudományos csodákról olvashatunk, amelyek eredményei nélkül el sem tudnánk képzelni az életünk. A tudomány nem konstans vagy végleges, hanem folyamatosan fejlődő. Gyarapítása tudományos kutatások, és további tanulmányok által válhat lehetővé. Ebből kifolyólag nem mindegy, hogy mit olvasunk, vagy miből inspirálódunk. [Jú14]

Jelen dolgozatomban, ezen fontos folyóiratokat igyekeztem rangsorolni egyenként, hogy a szerzőik, esetleg az olvasóik egy könnyed keresés után azonnal tudják, hogy mely kategóriában melyik volt az elmúlt évek legjobb értékelését szerző folyóirata. A felhasználók számára létrehoztam egy nyomtatási lehetőséget is, amelyet a jsPDF könyvtárral valósítottam meg, melynek köszönhetően a felhasználó PDF-et generálhat a kereséséből, amit letölthet a saját eszközére. Ráadásul egy külön adatbázisba lementem az összes webhelyen történt keresést, a felhasználó által beírt szerző alapján. Továbbá, ha egy adott folyóiratot nem találnak, akkor annak az alapvető információival kibővíthető az adatbázisunk.

A választott és megvalósított programokkal, rendszerekkel arra törekedtem, hogy egy olyan teljes működő webhelyet tudjak létrehozni, amely gyors választ ad, felhasználóbarát, és könnyen elérhetővé tehető bárki számára. Emiatt esett a választásom egy WordPress webhely létrehozására. Ezen szoftver segítségével egyszerűen létrehozható egy webhely, amelyet saját ízlésem szerint alakíthatok, könnyedén kezelhetek nagy mennyiségű adatot, amelyekkel ezáltal gördülékenyen lehet dolgozni.

Felmerülhet bennünk a kérdés, hogy mi is az SJR, és hogyan is működik az értékelési rendszere. Az SJR-t az úgynevezett Scimago Lab fejlesztette ki, amely a spanyolországi Granada városában található Granada Egyetem egyik kutatócsoportjából áll. Az úgynevezett SJR mutató a tudományos folyóiratok tudományos befolyásának mérőszáma, amely figyelembe veszi mind az egyéb folyóiratok által kapott idézések számát, mind pedig az idézések forrásául szolgáló folyóiratok fontosságát, presztízsét. Egy folyóirat SJR-mutatója egy numerikus érték, amely az adott folyóiratnak az előző három év során közzétett dokumentumra jutó súlyozott idézések átlagos számát mutatja egy adott évben. A magasabb SJR-mutató értékek a folyóirat nagyobb presztízsét jelzik. Ezen numerikus érték kiszámításához a fejlesztői a Google PageRank™ algoritmusát és a Scopus® adatbázisát használták fel. [sci23]

Az SJR nagy adathalmazával a WordPress rendszerben sem a legegyszerűbb dolgozni. A nagy mennyiségű adatt miatt egy pontos és gyors keresést alkalmaztam, az Ajax-Search-t. Ez kis mennyiségű adatokkal dolgozik a háttérben, emiatt a weblapot nem kell újratölteni minden egyes keresés alkalmával. A keresés megavólitásához jQuery-t, és PHP-t applikáltam.

A webhely megjelenítését egy úgynevezett Enfold WordPress Témában valósítottam meg. A téma segítségével a weboldalak alapvető felépítését hoztam létre, mint például a fejléct, továbbá az alapvető betűtípust és színeket állítottam be. A weboldal fejlécében megjelenő oldalakat HTML-ben írtam, ezeket SCSS-el (Sassy CSS) formáltam és dizájnoltam. A webhely biztonságának érdekében egy egyszerűbb megoldást alkalmaztam, az Ultimate Member-t, amely egy ingyenesen elérhető, nyílt forráskódú bővítmény a WordPresshez.

A dolgozat további részében ismertetem az általam használt szoftvereket, és előnyeiket. Továbbá bemutatom a sikeresen létrehozott kereséseket és megjelenítéseket.

2. fejezet

Programok, technológiák bemutatása

2.1. Adatok

Dolgozatomban az SJR adatbázisával dolgoztam, amelyet a webhelyükről töltöttem le. [SJR](#) Az adatok egy CSV, azaz pontosvesszővel tagolt formátumban elérhetőek. Ezen információkat egy bővítmény segítségével sikerült a weboldalam adatbázisához illesztnem. A bővítmény hiányosságai miatt fel kellett dolgoznom az adatokat. Pontosabban mezők típusainak sokaságát kellett megváltoztatnom ahhoz, hogy a megfelelőképpen tudjam tárolni az adatokat.

Az adatok aktuálításának érdekében létrehoztam egy saját bővítményt, amely egy PHP fájlból áll. A bővítmény egy a WordPress oldalon futó, az Unix-alapú operációs rendszerekben alkalmazott úgynevezett Cron Job segítségével negyedévente letölti az SJR weboldaláról a táblázatot CSV-formátumban, amelyben kicseréli a vesszőket pontos vesszőkre, és az újonnan létrejött dokumentumot átadja az importálásra használt bővítménynek, amely a továbbiakban feltölti az adatbázisban még nem létező folyóiratokat.

Az évek alapján feltöltött adatokat úgynevezett posztokként tároltam. Minden tudományos folyóirat egy posztnak felel meg, amelynek sajátos adatai vannak. A folyóiratok adatait tároló adatbázist az Advanced Custom Fields nevezetű bővítmény segítségével hoztam létre. Ezeket az adatokat az importálásból kapja meg az adatbázis. A posztok dinamikusak, könnyen változtathatóak, ugyanazon sablont használva megjeleníthetőek.

2.2. Felhasznált technológiák az adatok feldolgozására

2.2.1. Excel [[Mic16](#)]

A Microsoft Excel segítségével dolgoztam fel a CSV adatokat. Néhány numerikus adatot tartalmazó mezőben a vesszőket ki kellett cserélnem pontos vesszőkre. Másként nem fogadta el a WP All Import bővítmény INT típusúnak a mezők értékeit.

2.2.2. WP All Import [[Sof22](#)]

A bővítmény segítségével egyszerűen lehet feltölteni adatokat a webhelyünkre. Csak feltöltjük a CSV vagy XML típusú fájlokat, kiválasztjuk, hogy mihez szeretnénk hozzárendelni - a dolgozatom esetében ezek a posztok voltak - és a feltöltött fájl mezőit hozzárendeljük egyenként az általunk kívánt mezőkhöz. Az importálás formáját le lehet menteni sablonként, amelyet újra és újra fel lehet használni. Ennek köszönhetően tudtam biztosítani a saját bővítményemben is az új adatok helyes feltöltését.

2.2.3. Advanced Custom Fields [[Del22](#)]

Egy újabb bővítményt kellett alkalmaznom ahhoz, hogy a posztjaimnak speciális egyedi mezőjei és metaadatai lehessenek. Emiatt egy általam már ismert, népszerű bővítményt használtam, amellyel hozzáadtam a szükséges mezőket a posztként tárolt tudományos folyóiratokhoz, és a lementett keresésekhez is egyaránt.

2.3. Felhasznált szoftverek a webhely létrehozására

2.3.1. WordPress [[Wor22](#)]

A webhely legfőbb alkotóeleme, amely tartalmazza az összes bővítményt és posztot, a WordPress szoftver. A hivatkozásokban elérhető linken megtalálható WordPress szoftver nyílt forráskódú és ingyenesen használható. A használatához szükségeltetik egy domain név, és webtárhely. Az általam készített webhely egy Brassóban található webtárhelyen van tárolva, és egy fejlesztői domainhez van kötve. [F.2.1](#)

Azért választottam ezt a szoftvert, mert a diplomadolgozatom estében a legfontosabb, hogy könnyedén, és bárholnan elérhető legyen a webhely. Emellett az eredményeket egy jól kialakított felületen lehet gyorsan kezelni és megjeleníteni.

2.3.2. Enfold Téma [[Kri22](#)]

A WordPress webhelyek könnyed arculatváltásának érdekében úgynevezett témákat lehet hozzárendelni az adott webhelyhez. A dolgozatomhoz egy általam már megvásárolt témát használtam. A témával fontosabb kinézeti elemeket lehet megváltoztatni amely kihat vagy az egész webhelyre, vagy csak egyedi weboldalakra.

2.4. Felhasznált technológiák a rangsorolásra

2.4.1. HTML5 [[WHA08](#)]

A HTML, avagy a hiperszöveges jelölőnyelv (HyperText Markup Language) a weboldalak fő alkotóeleme. Olyan szimbólumokat és szövegeket tartalmaz, amelyek a megjelenítő programnak leírják, hogyan is kell megjeleníteni illetve feldolgozni az adott állomány tartalmát. A HTML állomány három részre bontható, a Dokumentum Típus Definícióra, a fejlécre, és a törzsrre. A dolgozatomban csak a törzsben alkalmaztam, az általam kívánt megjelenítés eléréséhez.

2.4.2. SCSS [[Nat22](#)]

Az SCSS egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Az SCSS teljes mértékben kompatibilis a CSS szintaxisával, miközben támogatja a Sass teljes erejét. Lehetővé teszi olyan funkciók használatát, amelyek még nem részei a szélesebb CSS szabványnak, és jobb munkafolyamatokat biztosít a stíluslapok karbantartásához. A webhelyen a SCSS segítségével változtattam a weboldalak stílusán.

2.4.3. Node-sass [[Ope22](#)]

A Node-sass egy Node.js könyvtár, amely lehetővé teszi, hogy a .scss fájlokat natív módon, hihetetlen sebességgel és automatikusan css-re fordíthassuk. Minden SCSS fájl mentésekor a beállított mappában kigenerálja a tömörített CSS fájlt. Ennek köszönhetően a webhely könnyedén egy fájlban megtalálja a teljes weboldal dizájnját, amelyet szépen strukturált, több fájlból építettem fel.

2.4.4. PHP [[PHP22](#)]

A PHP egy általános szerveroldali szkriptnyelv dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható. A kódot a webszerver PHP feldolgozómodulja értelmezi, ezzel dinamikus weboldalakat hozva létre. Továbbá az Ajax egyik alkotó eleme, hiszen a JavaScript a szerver oldali PHP fájlokat hívja meg, amelyek esetemben feldolgozzák és visszatérítik, lementik vagy kicserélik a megfelelő adatot. [[wik07](#)]

2.4.5. JavaScript [[Net11](#)]

JavaScript egy dinamikus szkriptnyelv, amelyet weboldalak interaktivitásának és dinamizmusának megvalósítására használnak. Az egyik legelterjedtebb és legfontosabb programozási nyelv a webfejlesztés területén.

JavaScript kliensoldali nyelvként működik, vagyis a böngészőben fut, és lehetővé teszi az interaktív elemek, dinamikus tartalom és aszinkron műveletek kezelését a weboldalon. Ez a nyelv felelős azért, hogy weboldalaink élővé váljanak és felhasználói interakcióra reagáljanak.

2.4.6. jQuery [[jQu22](#)]

A jQuery egy nyílt forráskódú JavaScript könyvtár, amely leegyszerűsíti a webes alkalmazások létrehozását és navigálását. A jQuery különösen a HTML dokumentumobjektum-modell (DOM) manipulációját, az aszinkron JavaScript és XML (Ajax) és az eseménykezelést egyszerűsíti. A jQuery emellett a CSS-tulajdonságok manipulálásával JavaScript-funkciókat is tartalmaz, hogy a weboldal elemeinek olyan effektusokat adjon, mint például a be- és kikapcsolás. A jQuery széles körben használt JavaScript-könyvtár és több ezer felhasználó által készített bővítmény támogatja. [[jQu19](#)]

A dolgozatomban kihagyhatatlan volt a használata, hiszen oldalújrátöltés nélkül szerettem volna keresni a tudományos folyóiratok között. Erre Ajax keresést alkalmaztam,

amely a gomb megnyomására kérést küld és meghív egy PHP függvényt. A PHP függvény által visszatérített adatot is ezáltal jelenítettem meg. Továbbá különböző DOM elemek eltüntetésére, vagy megjelenítésére is használtam, mint ahogy a gombok megnyomásának a figyelésére is.

2.4.7. jsPDF [[Par22](#)]

Az jsPDF egy JavaScript könyvtár, amely lehetővé teszi a PDF-dokumentumok dinamikus generálását a böngészőben. A könyvtár széles körben használható a webfejlesztésben, különösen olyan alkalmazásokban vagy weboldalakban, ahol szükség van a felhasználói adatok PDF formátumban történő exportálására vagy dokumentumok dinamikus létrehozására.

2.4.8. PHP Storm [[Jet21](#)]

A PhpStorm egy szabadalmaztatott, platformokon átívelő IDE azaz integrált fejlesztőkörnyezet PHP-hez, amelyet a csehországi JetBrains cég készített. A PhpStorm PHP-, HTML- és jQuery-szerkesztőt kínál, amely a PHP- és jQuery-kódhoz menet közbeni kód-elemzést, hibamegelőzést és automatikus refaktorálást biztosít.

A PhpStorm Java nyelven íródott. A felhasználók bővíthetik az IDE-t a PhpStorm-hoz készített bővítmények telepítésével, vagy írhatnak saját bővítményeket. [F.1.1](#)

2.4.9. GitHub

A GitHub egy ingyenes használható verziókezelő platform, amely egyaránt egy szociális hálózat is. A fejlesztők megoszthatják másokkal a kódjaikat, esetleg jelenthetnek hibákat a fejlesztők számára (Issues), sőt akár javíthatnak is rajta (Pull Request), amelyet az eredeti publikáló könnyedén hozzáadhat a tárolójához (Repository).

2.5. Felhasznált egyéb technológiák

2.5.1. FacetWP [[Fac21](#)]

A FacetWp bővítményt a folyóiratok listázásánál alkalmaztam. Ennek segítségével elsősorban behívtam az egyedi felépítésű listázást és létrehoztam két legördülő menüt, hogy a felhasználó rendezhesse a listát különböző szempontok alapján.

2.5.2. Ultimate Member [[Ult23](#)]

Az Ultimate Member egy WordPress bővítmény, amelyet főként felhasználói profilok kialakítására szolgál. Ez a bővítmény lehetővé teszi, hogy könnyedén hozzassunk létre és tesztre szabhassunk felhasználói regisztrációs és bejelentkezési rendszereket. A bővítmény szükséges volt ahhoz, hogy az adatbázist biztonságossá tegyem a rosszakaró látogatók ellen.

3. fejezet

Rendszer specifikációja

3.1. Rendszerkövetelmények

A dolgozatom következő részében bemutatom a rendszer követelményeit, funkcionális- és nem funkcionális igényeit az általam létrehozott webhelynek.

A webhely rendszerkövetelményei:

1. A webhely további működtetésének érdekében nincs szükség emberi erőforrásokra, hiszen minden automatikusan implementálva lett.
2. Hardveres erőforrásokra viszont szükség van: kell egy webes szerver, ahonnan elérhetővé lehet tenni.
3. A webhely elérhetőségéhez és működéséhez stabil internetkapcsolatra van szükség.
4. Fontos szempont a webhely biztonsága és védettsége. Ez magába foglalja a megfelelő tűzfalak, SSL tanúsítványok és más biztonsági intézkedések beállítását a webhely védelmére a kiberbűnözők és támadásaik ellen.

3.1.1. Funkcionális követelmények

A webhely fejlesztése alatt arra törekedtem, hogy egy újonc számára is érthető legyen a használata. A fejlécen elhelyezkedő menüpontokkal könnyedén lehet navigálni az oldalak között. A fejléc minden oldalon megjelenik, ezért egyszerűen lehet oldalt váltani.

Két fajta weboldal található meg a webhelyen:

1. **Prezentációs oldalak:** ezeken az oldalakon, a felhasználó csak nézelődni tud. Az általa kiválasztott tudományos folyóiratokat, vagy azok információit jelenítettem meg. A megjelenítést sablonok létrehozásával oldottam meg, amelyek a rendelkező információk alapján épülnek fel.
2. **Kereső oldalak:** Ezeken az oldalakon üres mezők és gombok találhatóak. A felhasználó az üres mezőt kitöltheti az általa keresett folyóirat nevével, vagy úgynevezett ISSN - egyedi kódjával - , továbbá a lementett keresésekben való keresés esetén kitöltheti egy szerző nevével, akadémiai évvel, vagy a folyóirat értékelésével, majd a "Search" gombra kattintva megjelennek a találatok ugyan azon a weboldalon. Az

ISSN keresés azért különlegesebb, mert az ott megjelenő második mezőt akár több egyedi számmal is kitöltheti a felhasználó. A weboldal legkevesebb egy, legfeljebb nyolc folyóiratot tud megjeleníteni, kategóriái alapján értékelve. Esetleg lehetősége nyílik a saját nevét, vagy a folyóiratok szerzőinek nevét is kitölteni az első mezőben, amely így a névvel fogja megjeleníteni a folyóiratok értékelését. A mentett keresésekben való keresés esetében a weboldal megjelenít minden olyan szerzőt, amelynek volt legalább egy, a keresett évben kiadott folyóirat az elsőként említett keresésében, vagy a keresett ranggal megegyező rangú folyóirata.

3.1.2. Nem funkcionális követelmények

A használhatóság és teljesítmény szempontjából igyekeztem a fejlettebb, és gyorsabb technológiákat alkalmazni. Elsősorban a weboldalak betöltésére figyeltem. A sebesség fokozásában arra törekedtem, hogy ne keljen sok adatot betöltenem ahhoz, hogy a weboldal fő látványos elemei látszódjanak. A weboldalak betöltési sebességét megvizsgáltam, oldalanként tízszer, és az oldalak töltésének átlaga, kevesebb mint két másodperc.

Emellett odafigyeltem, hogy a weboldal rendelkezzen minél több ember számára szükséges funkciókkal. Továbbá elérhető legyen bárki számára, akinek van internet kapcsolata. Viszont egy beléptető rendszerrel láttam egy a webhelyet, hogy megelőzzem az esetleges visszaéléseket a webhely szolgáltatásaival szemben.

Az oldal karbantarthatóságát az UpdraftPlus - Backup/Restore bővítmény biztosítja. Ezen bővítmény minden nap készít egy biztonsági mentést a webhely adatbázisáról, henteente egyszer a fájlok összességéről, majd azt lementi a felhőbe. Ha bármely pillanatban a webhely megadja magát, akkor egy kattintással vissza lehet hozni a legutóbbi, még működő verziót. Ennek köszönhetőn a webhely biztosítani tudja, hogy az év minden napján a felhasználók rendelkezésére állhasson. [Upd22]

A webhely biztonságát az SSL biztosítja, mely titkosított kapcsolatot hoz létre a webszerver és a böngésző között, és ez teszi URL előtagja elé a "HTTPS"-t.

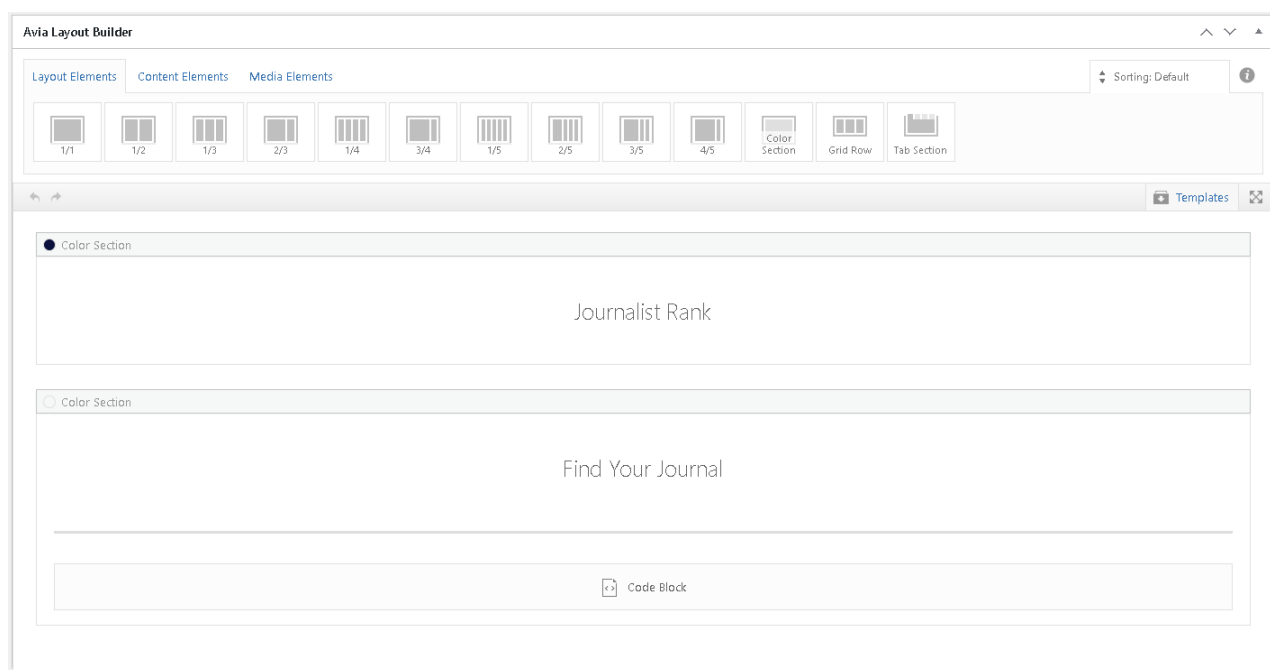
4. fejezet

Webhely szerkezete

4.1. Webhely backend

A következőkben elsősorban bemutatom a szoftvert vizuálisan, ahhoz hogy elősegítsem a szoftver felépítésének megértését.

A weboldalak a WordPress backend-jében lettek létrehozva. Itt építettem fel az oldalakat a következő képpen:



4.1. ábra. Az Enfold téma oldal szerkesztő bővítménye.

A fenti ábrán látszik, hogy két címet jelenítettem meg - az első cím háttere színes - és egy kód részt, ahová behívtam egy általam megírt űrlapot. Mindhárom kereső oldal hasonlóan néz ki, ahogyan a listázó oldal is.

A WordPress témában, ha módosításokat szeretnénk alkalmazni, ahhoz elsősorban létre kell hozni egy child témát. Ezt közvetlen a szülő téma mellett kell megtenni. A neve a következőképpen kell kinézzen: *szülőneve-child*. Minden amit én hoztam létre itt

található. A dolgozatomban esetében, az enfold-child mappa tartalmazza az én szoftvereimet, ez a mappa elérhető a GitHub-on is.

A tudományos folyóiratokat egyenként megjelenítő oldalt én építettem fel PHP, és HTML használatával. Itt lényegében a felhasználó a posztokat jelenítheti meg egyenként, az adataikkal együtt. A megjelenítés az saját sablonomat alkalmazza, amely megtalálható a következő elérési útvonalon: `enfold-child\single-post.php`

4.2. Webhely frontend

Mostmár, hogy tudjuk, hogyan is épülnek fel a háttérben a weboldalak szerkezetei, tekintsünk körül, hogy mit is lát a felhasználó, amikor meglátogatja az oldalt. Négy oldal megjelenítését mutatnám be.

Ez első és egyben fő oldal egy keresés, ahol a felhasználó egy , vagy akár nyolc folyóíratra kereshet rá. Ha a keresőmotor talál valamit, akkor megjeleníti a folyóiratok címét, kategóriáit, azok értékelését, és a legjobb értékelést, amelyet valamely kategóriája kapott. Amennyiben a folyóiratok egy szerző által keletkeztek, beírhatja a szerző nevét is, amely megjelenik a folyóiratok felett, ezt a keresést a webhely automatikusan lementi, amely továbbnavigálva a "Searches" fülön jeleníthető meg.

Négy véletlenszerű ISSN-t írtam be, a saját nevemmel ellátva. A következő eredmény született:

SJR

TITLE ISSN

Fekete Dániel

You can check at least one, up to 6 Journals, separate with -> ; For example: 12345678;98765432;12344321

00079235;14710072;00426822;17055105

Check

Fekete Dániel

Ca-A Cancer Journal for Clinicians	Hematology D1 Oncology D1	Top Rank: D1
Nature Reviews Molecular Cell Biology	Cell Biology D1 Molecular Biology D1	Top Rank: D1
Virology	Virology Q2	Top Rank: Q2
International Journal of Numerical Analysis and Modeling	Numerical Analysis Q2	Top Rank: Q2

4.2. ábra. Az ISSN keresés, négy egyedi kóddal, és a nevemmel ellátva.

A második oldal egy folyóirat egyedi oldala. Ezen oldal minden folyóiratnál azonos, csak a folyóirat információi változnak.

SJR

TITLEISSNQ

MMWR Recommendations and Reports
Publisher: Centers for Disease Control and Prevention (CDC)

SJR:
25.045

SUBJECT AREA AND CATEGORY:
Your rank D1 in Epidemiology category.
Your rank D1 in Health (social science) category.
Your rank D1 in Health Information Management category.
Your rank D1 in Health, Toxicology and Mutagenesis category.
Your rank D1 in Medicine (miscellaneous) category.

Personal Best:
D1

H-Index:
148

ISSN:
10575987, 15458601

Country:
United States

4.3. ábra. Folyóiratok megjelenítése.

Eme oldalakra a cím keresőről, vagy a folyóiratok listázásáról lehet eljutni. A keresett folyóirat nevét beírva, megjeleníti azon találatokat, amelyek címében szerepel a mezőbe írt szöveg. Majd megjeleníti azokat egy táblázatban, amelyek teljes címére kattintva eljuthatunk az egyedi oldalára.

A harmadik oldal a listázás. Amikor a felhasználó ide érkezik, a legjobb 40 folyóiratot látja kategóriától függetlenül az SJR értéke alapján. Itt a felhasználó kiválaszthat egyedi kategóriákat és rendezési szempontokat. Az oldal mindig 40 folyóiratot fog megjeleníteni. A cím háttére a folyóirat rangja alapján változik.

A következő képen kiválasztottam a **Small Animals** kategóriát, és rendeztem a rangsor alapján :

Theriogenology		
SJR: 0.685	H Index: 137	Top-rank: D1
Veterinary Clinics of North America – Small Animal Practice		
SJR: 0.684	H Index: 73	Top-rank: D1
Journal of Feline Medicine and Surgery		
SJR: 0.645	H Index: 66	Top-rank: Q1
Journal of Small Animal Practice		
SJR: 0.626	H Index: 71	Top-rank: Q1
Porcine Health Management		
SJR: 0.615	H Index: 21	Top-rank: Q1
Journal of the American Animal Hospital Association		
SJR: 0.47	H Index: 69	Top-rank: Q2

4.4. ábra. Folyóiratok listázása.

Az utolsó ábrán bemutatom a lementett keresések közötti keresést:

RANK CHECKER

AccountSearchesISSN SearchTitle SearchJournals List

SEARCH IN OUR DATABASE

d1

Search

FEKETE DANIEL

Quarterly Journal of Economics	Max Rank: D1	Academic Year not set
Nature Reviews Molecular Cell Biology	Max Rank: D1	Academic Year not set
Ca-A Cancer Journal for Clinicians	Max Rank: D1	Academic Year not set

FARKAS CSABA

Nonlinear Analysis: Real World Applications	Max Rank: D1	Academic Year not set
---	--------------	-----------------------

4.5. ábra. A szerzők által keresett D1-es kategóriába tartozó folyóiratok

Ezen az oldalon rákereshetünk, egy adott rangsorolásra, egy szerző nevére, vagy akár egy akadémiai évszámra is. Az adatbázisba csak egyszer mentődik le minden egyes keresés, dátummal ellátva. Így a webhely adminisztrátorai még azt is láthatják, hogy melyik szerző mikor, és mely folyóiratokra keresett rá.

5. fejezet

Folyóiratok értékelése

5.1. Értékelési rendszer

Ahogy a dolgozatomban már többször is említettem, a folyóiratok értékelése volt a fő cél. Felmerülhet bennünk a kérdés, hogy hogyan értékeltém a folyóiratokat. Ebből a szemszögből könnyű dolgom volt, hiszen ezt már a SCImago Jorunal Rank megtette helyettem. Egy folyóirat az idézések száma és az idéző lap presztízse alapján kapott egy értéket. Én a tudományos folyóiratok úgynevezett Q-indexére voltam kíváncsi. Ezt már az SJR is kiszámolta előttem. Viszont én kategóriákra bontva akartam rangsorolni.

A Q-index a folyóiratokat négy negyedbe sorolja be.

Index	Érték
Q1	0-25%
Q2	25-50%
Q3	50-75%
Q4	75-100%

5.1. táblázat. Q-index értékelési rendszere.

A dolgozatomban bevezettem még egy rangsorolási indexet, a D1-et. Ezt a mutatót, a legjobb 10%-ba tartozó folyóiratok kategóriái kapták meg.

Tehát én az SJR által már kiszámolt érték felhasználásával csökkenő sorrendbe állítottam a folyóiratokat a kategóriáik alapján, és megnéztem, hogy a sorban hányadik helyen állnak, ebből létrehozva egy százalékos értéket. Ezután egy szoftver, segítségével számoltam ki a folyóirat rangját.

Index megállapításáért felelős függvény:

```
function get_rank($value){  
    if ($value > 0 && $value <= 10) {  
        return 'D1';  
    }  
    if ($value > 10 && $value <= 25) {  
        return 'Q1';  
    }  
    if ($value > 25 && $value <= 50) {  
        return 'Q2';  
    }  
    if ($value > 50 && $value <= 75) {  
        return 'Q3';  
    }  
    if ($value > 75 && $value <= 100) {  
        return 'Q4';  
    }  
}
```

6. fejezet

A szoftver bemutatása

6.0.1. Az űrlapok létrehozása

Az előző fejezetben (4.1) említettem meg a behívott űrlapokat. Ezeket az egyesített functions.php-ban hoztam létre, úgynevezett "shortcode" formátumban. Egy WordPress függvény segítségével hozzárendeltem a webhelyhez.

```
add_shortcode('issn_input', 'issn_input_shortcode');

function issn_input_shortcode()
{
    $atts = '<div class="form-container-issn">';
    $atts .= '<form method="post" action="">';
    $atts .= '<label for="input-field-author">Author Name: </label>';
    $atts .= '<input id="input-field-author" type="text">';
    $atts .= '<label for="input-field-data-holder-issn">You can check at least
        one, up to 6 Journals, separate with -> ; For example: 12345678;
        98765432; 12344321 </label>';
    $atts .= '<input id="input-field-data-holder-issn" pattern="^[0-9,]*$"
        type="text">';
    $atts .= '</form>';
    $atts .= '<button type="button" name="submit"
        class="search-on-pressed-issn" >Check</button>';
    $atts .= '</div>';
    $atts .= '<div id="response_holder_issn">';
    $atts .= '</div>';
    return $atts;
}
```

Az eljárás lényege, hogy egy stringbe beleírom a teljes felépítését az űrlapnak. Amikor meghívódik, akkor a string tartalmát megjeleníti a weboldalon. Ezen függvényből két darab van, hiszen két különbözőt kellett létrehoznom a két keresés eltérő mezőjei miatt.

6.1. ISSN szerinti keresés

A dolgozatomban ezen részében bemutatom az általam készített kódokat, kódrészleteket. Ezen szoftvereket PHP és jQuery nyelven írtam, PHP STORM IDE legfrissebb változatában. [PHP22]

A dinamikus keresés okozta a legtöbb nehézséget számomra. Kihívás volt elkészíteni, hiszen még sosem hívtam segítségül az Ajax-ot, ráadásul ebben az Ajaxban van egy másik Ajax meghívás is, amelyre azért volt szükség, hogy le tudjam menteni a keresést. A dinamikus keresés érdekében egy hatékony megoldást kellett választanom, amelynek köszönhetően az Ajaxra esett a választásom elsősorban. A megvalósítás eléggé összetett. A keresés több függvényből épül fel. A dinamikussága a háttérben történő keresésében rejlik. A keresés elindításához egy kiváltó okra van szükség, amely esetemben egy gombnyomásnak felel meg. A megvalósításához jQuery-t használtam. Ennek a módszernek köszönhetően nem kell az oldalnak újra töltődnie ahhoz, hogy a keresett folyóiratokat megjelenítse a látogató számára.

6.1.1. jQuery

A script a következőképpen néz ki:

```
1  jQuery('.search-on-pressed-issn').on('click', function () {
2      /**Getting data from input field, and Posts ISSN **/
3      let issn_input = document.getElementById('input-field-data-holder-issn').value;
4      let author_input = document.getElementById('input-field-author').value;
5      var output;
6      //clear old data
7      $('#response_holder_issn').html("");
8
9      /** Ajax search **/
10     $.ajax({
11         type: 'POST',
12         url: rank_checker.admin_url,
13         dataType: 'json',
14         data: {
15             'action': 'get_issn_data_fun',
16             'issn_input': issn_input
17         },
18         success: function (data) {
19             output = [];
20
21             if (Array.isArray(data[0].succes_data)) {
22
23                 if (author_input !== '') {
24                     jQuery.ajax({
25                         url: rank_checker.admin_url,
26                         type: 'POST',
27                         data: {
28                             action: 'save_search',
29                             author: author_input,
30                             journals: data[0].succes_data,
31                         },
32                         success: function (response) {
```

```

33         console.log(response)
34         output = '<h6>' + response.data + '</h6>';
35         $('#error-handler').append(output);
36         $('#error-handler').show();
37         setTimeout(function () {
38             $('#error-handler').hide();
39         }, 5000);
40     },
41     error: function (xhr, status, error) {
42         // Handle error response
43         output = '<h2>' + error + '</h2>';
44         $('#error-handler').append(output);
45         $('#error-handler').show();
46         setTimeout(function () {
47             $('#error-handler').hide();
48         }, 5000);
49     }
50 });
51 }
52
53 $.each(data[0].succes_data, function (i) {
54     $data_single = data[0].succes_data;
55     output += '<div id="issn_holder">' + '<div class="title_holder">' +
56         $data_single[i].title + '</div>';
57     output += '<div class="issn_holder_data">';
58     $.each($data_single[i].ranking, function (j) {
59         output += '<div class="issn_holder_data_columns">' + $data_single[i]
60             .ranking[j].response_cat + ' ' + '<span class="bold">' +
61             $data_single[i].ranking[j].response_cat_rank + '</span>' + '</div>';
62     });
63     output += '</div>';
64     output += '<div class="top_rank_holder">' + '<span>Top Rank:</span>'
65         + '<span class="bold">' + $data_single[i].max_rank + '</span>' + '
66         </div>' + '<button onclick="editJournal(event)" id="journal_edit"
67         data_postID="' + $data_single[i].post_id + '" ></button>' + '</div>';
68
69     //checking if has custom data saved
70     if ($data_single[i].title_of_article || $data_single[i].number_of_editors
71         || $data_single[i].academic_year) {
72         output += '<div class="added-data">';
73         if ($data_single[i].title_of_article) {
74             output += '<span class="all-data"> Article Title: <span class="
75                 bold">' + $data_single[i].title_of_article + '</span> </span>
76                 ';
77         }
78         if ($data_single[i].number_of_editors) {
79             output += '<span class="all-data"> Number of Editors: <span
80                 class="bold">' + $data_single[i].number_of_editors + '</span>
81                 > </span>';
82         }
83         if ($data_single[i].academic_year) {
84             output += '<span class="all-data"> Academic Year: <span class="
85                 bold">' + $data_single[i].academic_year + '</span> </span>';

```

```

74         }
75         output += '</div>';
76     }
77     });
78 }
79 if (data[0].failed_issn !== null) {
80     $.each(data[0].failed_issn, function (i) {
81         $data_single_issn = data[0].failed_issn;
82         output += '<div id="issn_holder" class="failed">' + '<span>' +
            $data_single_issn[i].failed_issn + ' not found in our database, add this
            by clicking the + on the top right corner' + '</span>' + '<button
            onclick="addJournal(event)" id="journal_add" data_issn="' +
            $data_single_issn[i].failed_issn + '>+</button>' + '</div>';
83     });
84 }
85
86 if (!!author_input) {
87     author_output = [];
88     author_output += '<div id="author_holder">' + '<span>' + 'Author: ' + '</
89         span>' + '<span>' + author_input + '</span>' + '</div>';
90     $('#response_holder_issn').append(author_output);
91 }
92 $('#response_holder_issn').append(output);
93 },
94 error: function (data) {
95     output = '<h2>' + data.responseJSON.data + '</h2>';
96     $('#response_holder_issn').append(output);
97 }
98 });
99 });

```

Nagyon fontos, hogy a Script csak azután töltődik be, amiután a teljes oldal betöltődött, ezt az első sorban kezelem le.

Az első sorban látható a megfelelő ID-val rendelkező gomb megnyomásának figyelése. Amint ez megtörtént, belép a függvénybe.

A következő sorokban lementem egy-egy változóba a felhasználó által kitöltött mezők adatait. Majd deklarálom a kimenetet és üressé teszem a megjelenítésre használt szekciót, amelyre szükség van ahhoz, hogy egy újabb keresés esetén tüntesse el az előző keresési eredményeket.

A 10. sortól kezdődik az Ajax search. Ide fentről lefelé haladva a következő adatokat kellet megadnom:

1. A kiszolgálótól várt adatok típusa. Én adatot kértem, emiatt POST változót kaptam. Ahhoz, hogy a továbbiakban megmondhassam a függvénynek, hogy hogyan cselekedjen sikeres, vagy sikertelen válasz esetén, meg kellet adnom ezt az adatot.
2. Az URL az egyetlen kötelező paraméter. Ez a karakterlánc tartalmazza azt a címet, amelyre a kérést el kell küldeni. A dolgozatomban ezt előre deklaráltam. A rank_checker változónak az 'admin-ajax.php' karakterlánc értéket adtam.

3. Az adat típusát JSON-re állítottam, hogy könnyebben tudjak adatot visszatéríteni.
4. Az adatnál két paramétert használtam:
 - (a) Az első a függvény amelyet meghív.
 - (b) A második pedig egy változó, amelyet továbbít a meghívott függvény számára. Esetemben ez a mezőbe írt adatanak felel meg.

Ha a kitöltött adatok megfelelőek, meghívja az előbb említett `get_issn_data_fun` függvényt. A meghívott függvény válasza alapján jár el a továbbiakban.

A meghívott függvényben kétféle választ tudunk adni:

1. sikeres válasz esetén, elsősorban ellenőrzöm hogy a szerzői mező üres-e. Amennyiben nem, hozzá csatolja a weboldalon létrehozott eddig üres HTML szekcióhoz. Továbbá a 23.- sortól látható a következő Ajax meghívás, amely lementi a szerzőhöz rendelve a keresett folyóiratokat. A script által meghívott `save_search` nevű PHP függvénynek hasonlóképpen az előzőhöz két fajta visszatérítési értéke lehet, amelynek célja a hibakezelés és a látogató informálása. Lényegében egy öt másodpercig tartó üzenetet jelenít meg, amely a mentés sikerességét tartalmazza.

Az 53. sorban látható szintaxis végigmegy a visszatérített adatokon. Esetünkben egy több dimenziós tömbön. Ezen végighaladva első sorban kiveszi a címet és a kimeneti karakterlánchoz csatolja a HTML elemekkel együtt. Majd újra végigmegy minden kategóriáján az adott folyóiratnak és hozzácsatolja a kimeneti karakterlánchoz - HTML-ben felépítve - a kategóriáit, a rangját, és a legjobb rangját is. Mindemellett a 64. sortól kezdődően megnézi, hogy más felhasználó által van-e lementett információ az egyedi mezőkben, amennyiben talál, azokat is megjeleníti. Az ISSN alapján sikeresen megtalált folyóiratokhoz hozzárendel egy gombot a jobb felső sarokban, amely megkattintásával egy olyan űrlap jelenik meg, amely által hozzáadhatunk ezekhez az előbb már említett egyedi mezőkhöz új adatokat. A sikertelen ISSN keresés esetében is megjelenik egy hasonló gomb ugyanazon a helyen, amely megkattintásával egy másik űrlap jelenik meg, amely lehetőséget biztosít egy a webhely adatbázisában még nem szereplő folyóirat hozzáadásához. Ezeket az űrlapokat hasonlóan építettem fel, mint a már említett [6.0.1](#) fejezetben. És ezek is egy-egy Ajax lekérdezéssel működnek.

2. sikertelen válasz esetén hozzárendeli a sikertelenség okát az üres szekcióhoz, mint a beágyazott Ajax keresésnél. A cél a látogató informálása azzal kapcsolatosan, hogy létezik-e az adatbázisban a keresett ISSN, vagy valami hiba lépett fel a függvény futtatása alatt.

6.1.2. PHP

A továbbiakban bemutatom az előző függvény által meghívott függvényt, amely visszatéríti a feldolgozott adatokat a jQuery számára.

Fontos megemlítenem, hogy ahhoz, hogy az egyéni Ajax végpontokat lehessen kezelni úgynevezett webhook-okra van szükség. Azért van szükség többre, mert egyenként egy-egy fél számára engedélyezik a használatát. Vagy csak az Admin felhasználók számára, vagy csak a jogtalanok számára. A webhook-okat közvetlen a válaszadó függvény előtt deklaráltam. Ezekben a használandó függvényem nevét kell megadnom a "wp_ajax"/"wp_ajax_nopriv" előtagokkal, és azt a nevet amelyre a jQuery hívásban hivatkozom.

```
1 add_action('wp_ajax_nopriv_get_issn_data_fun', 'check_multiple_issn');
2 add_action('wp_ajax_get_issn_data_fun', 'check_multiple_issn');
```

Deklarálás után, már csak a függvényemre van szükség, amely feldolgozza az adatokat.

A feldolgozó egység:

```
1 function check_multiple_issn()
2 {
3     $searched_issns = $_POST['issn_input'];
4     if (strlen($searched_issns) < 4) {
5         wp_send_json_error('Too short ISSN, minimum numbers: 4', 404);
6     } elseif (strlen($searched_issns) > 60) {
7         wp_send_json_error('Too long ISSN, maximum numbers: 60', 404);
8     } else {
9         $issn_holder = explode(";", $searched_issns);
10        foreach ($issn_holder as $issn) {
11            $issn_without_space = str_replace(' ', '', $issn);
12            $args = array(
13                'post_type' => 'post',
14                'posts_per_page' => -1,
15                'meta_query' => array(
16                    array(
17                        'key' => 'issn',
18                        'value' => $issn_without_space,
19                        'compare' => 'LIKE'
20                    )
21                )
22            );
23            $journal_found = new WP_Query($args);
24            if ($journal_found->have_posts()) {
25                foreach ($journal_found->get_posts() as $post) {
26                    $categoryes = get_the_category($post->ID);
27                    $response_title = get_the_title($post->ID);
28                    $data_min = [];
29                    $flag = 101;
30                    foreach ($categoryes as $category) {
31                        $response_cat = $category->name;
32                        $response_cat_rank = get_rank(get_the_order_by_sjr($category->slug,
33                            $post->ID) * 100);
34                        $data_min[] = array(
35                            "response_cat" => $response_cat,
```



```

35         "response_cat_rank" => $response_cat_rank
36     );
37     $valuenumber = get_the_order_by_sjr($category->slug, $post->ID) *
        100;
38
39     if ($valuenumber < $flag) {
40         $flag = $valuenumber;
41     }
42 }
43 $response_max_rank = get_rank($flag);
44 $data[] = array(
45     "title" => $response_title,
46     "max_rank" => $response_max_rank,
47     "ranking" => $data_min
48 );
49 }
50 }
51 }
52 if ($data) {
53     echo json_encode($data);
54     die;
55 } else {
56     wp_send_json_error('No Search Reults Found', 404);
57 }
58 }
59 }

```

A függvény harmadik sorában egy változóba eltárolom a mező értékét, amelyet a jQuery küldött. Majd a további sorokban megvizsgálom a hosszát, ha nem elég hosszú, vagy túl hosszú, akkor sikertelen üzenetet küldök vissza.

A továbbiakban feltételezem, hogy az adat helyes-e, amely beérkezett. Ezután eltárolom egy változóban a feldarabolt beérkező adatot. Ennek a módszernek köszönhetően tudok egyszerre többet is megvizsgálni. Majd egyenként veszem mindenik feldarabolt adatot. Eltávolítom az üres szöközőket, hogy a további folyamatokba ez ne zavarjon bele. Ezután egy WordPress lekérdezés segítségével az összes folyóiratot átvizsgálva megkeresem azt amely ISSN értéke tartalmazza a feldarabolt beérkező adatot. Ez a WordPress lekérdezés a MySQL adatbázisának a gyors lekérdezésére szolgál.

A hatodik sorban létrehozok egy objektumot, amelybe beleillesztem a megtalált, és kiszűrt folyóiratokat. Amennyiben ez minden esetben üres, a legvégső sikertelen üzenet fog továbbítani a jQuery számára, amely jelzi a felhasználónak, hogy nem talált ehhez hasonló adatot a folyóiratok között.

Hogyha az objektum nem üres legalább egy esetben, akkor előkészítem a szükséges adatokat az előző jQuery függvény számára egy adat nevű változóba. Egy tömbbe illeszttem a folyóirat címét, egy tömbbe egyenként a kategóriáit, és az értékelését, és legutolsó sorban a legjobb értékelését.

Abban az esetben, ha a tömb nem üres, visszaküldöm az adatokat a kiíró egység számára, és kilépek a teljes függvényből.

A harminckettedik sorban láthatunk két függvény hívást. Az első már bemutattam, amely a folyóiratok rangját határozza meg ([5.1](#)). A második függvény megállapítja a

megadott kategória és a folyóirat egyedi száma alapján, hogy abban a kategóriában milyen értékelése van.

6.1.3. Folyóirat rangjának megtalálása kategóriánként

```
1 function get_the_order_by_sjr($cat_input, $post_ID_input)
2 {
3     $args = array(
4         'post_type' => 'post',
5         'posts_per_page' => -1,
6         'category_name' => $cat_input,
7         'meta_query' => [
8             'orderby_query' => array(
9                 'key' => 'sjr',
10            )
11        ],
12        'orderby' => array(
13            'orderby_query' => 'DESC',
14        ),
15    );
16
17    $query = new WP_Query($args);
18    $posts_found = $query->found_posts;
19
20    if ($query->have_posts()) {
21        $counter = 1;
22        foreach ($query->get_posts() as $post) {
23            //echo '<pre>' . var_dump($post) . '</pre>';
24
25            if ($post->ID === $post_ID_input) {
26                //echo $posts_found;
27                return $counter / $posts_found;
28            } else {
29                $counter++;
30            }
31            //echo '<strong>' . $query->found_posts . '</strong>';
32        }
33
34    } else {
35        return 'We didnt get any of other in this Category';
36    }
37
38 }
```

Ez a függvény hasonlóképpen használja a WordPress lekérdezéseket. Azonban itt a lekérdezés minden hasonló kategóriájú folyóiratot kiválaszt, és csökkenő sorrendbe rendezi őket a SCimago Jorunal Rank által meghatározott SJR érték alapján. Majd megkeresi, hogy hányadikként találta meg az adott folyóiratot az egyedi azonosítója alapján. Majd elosztja a helyezését az összes ugyanazon kategóriában levő folyóiratok számával.

Máris megvan minden ahhoz, hogy megtaláljuk a keresett folyóiratot, és az értékelését a kategóriái alapján.

6.2. További funkcionalitások

A webhelyen működő további funkcionalitásokat mind hasonló módszerrel kellett megoldanom, mint amelyeket már korábban bemutattam. Vagyis a cím szerinti keresés, az új folyóiratok hozzáadása, a már meglévők szerkesztése, vagy a lementett keresésekben történő keresést is Ajax segítségével valósítottam meg. Ezek jQuery része nem sokban különbözik a már bemutatottól. A lényeges különbség a meghívott függvény nevében, és a visszaérkezett adat feldolgozásában látható. A PHP részek nagyban eltérnek, hiszen teljesen más adatokat kell visszatéríteniük.

Egy érdekesebb funkciót ellátó függvényeket mutatnék be, amely a keresett folyóiratokat menti le hozzárendelve a szerzőhöz. A kód a következőképpen néz ki:

```
1 function save_search_ajax_handler()
2 {
3     if ( isset($_POST['author']) && isset($_POST['journals'])) {
4         $author = $_POST['author'];
5         $journals = $_POST['journals'];
6
7         // Check if the search query already exists as a post title
8         $existing_post = get_page_by_title($author, OBJECT, 'searches');
9         if ($existing_post) {
10             // Get the existing post ID
11             $post_id = $existing_post->ID;
12
13             // Check if journals repeater field exists
14             if (!get_field('searched_journals', $post_id)) {
15                 add_journals_to_repeater($journals, $post_id);
16                 wp_send_json_success('Search saved successfully');
17             }
18
19             // Check if the new journal data already exists in the repeater field
20             $existing_journal = get_field('searched_journals', $post_id);
21             $existing_journal_ids = array_column($existing_journal, 'the_journal');
22
23             $new_journals = array_filter($journals, function ($journal) use ($existing_journal_ids)
24             {
25                 return !in_array($journal['post_id'], $existing_journal_ids);
26             });
27
28             if (empty($new_journals)) {
29                 wp_send_json_success('Search already exists in our database');
30             } else {
31                 foreach ($new_journals as $journal) {
32                     $journal_exists = false;
33
34                     foreach ($existing_journal as $existing) {
35                         if ($existing['the_journal'] === $journal['post_id']) {
36                             $journal_exists = true;
37                             break;
38                         }
39                     }
40                 }
41             }
42         }
43     }
44 }
```

```

41         if (!$journal_exists) {
42             add_row('searched_journals', array(
43                 'max_rank' => $journal['max_rank'],
44                 'the_journal' => $journal['post_id'],
45                 'date_of_addition' => date('Y-m-d H:i:s')
46             ), $post_id);
47         }
48     }
49
50     wp_send_json_success('Search saved successfully');
51 }
52 } else {
53     // Create a new post if the search query doesn't exist
54     $post_data = array(
55         'post_title' => $author,
56         'post_type' => 'searches',
57         'post_status' => 'publish'
58     );
59
60     $post_id = wp_insert_post($post_data);
61
62     if (is_wp_error($post_id)) {
63         wp_send_json_error('Error saving search');
64     } else {
65         add_journals_to_repeater($journals, $post_id);
66         wp_send_json_success('Search saved successfully');
67     }
68 }
69 } else {
70     wp_send_json_error('Invalid request');
71 }
72 }

```

Ezt a függvényt a már említett [6.1.1](#) Ajax hívásban benne lévő Ajax lekérés hívja meg, amely feltétele, hogy legyen a szerzői mező kitöltve a keresésben. A függvény a következőképpen menti le a kereséseket:

1. A harmadik sorban ellenőrzi, hogy érkezik-e megfelelő adat, amely a szerzőt, és a keresett folyóiratok ISSN számát jelenti. Amennyiben érkezik, akkor ezeket az adatokat elmenti egy-egy változóba.
2. A nyolcadik sortól a függvény felülvizsgálja, hogy nem-e létezik már ez a szerző az adatbázisban, amennyiben nem létezik létrehozza, ellenkező esetben pedig lekéri az egyedi azonosítóját. Ezután azt is megvizsgálja, hogy van-e létrehozva számára a megfelelő mező, amelyet a későbbiekben fel kell töltenie, ha nincs ezt is hozzáadja.
3. A huszadik sorban végigmegy az új keresés folyóiratain és megvizsgálja a létezésüket. Ha nincs, akkor meghív egy egyszerű függvényt amely hozzáilleszti az keresés folyóiratit a szerzőhöz. Ez a függvény valójában megegyezik a harmincadik sortól kezdődő ciklussal.
4. Ha netalán már léteznének a mentett keresések között a folyóiratok, akkor felfrissíti az adatait.

5. Majd az utolsó sorokban a függvény lefutása alapján visszatérít egy üzenetet amelyet megjeleníték a felhasználónak a mentés állapotával kapcsolatosan.

6.3. PDF generálása jsPDF segítségével

Ahhoz, hogy a felhasználók PDF-ben is le tudják menteni a keresésüket a jsPDF könyvtárat alkalmaztam. Elsősorban letöltöttem GitHubról, és hozzáadtam a szükséges fileokat a megfelelő helyre, majd behívtam a projektembe. Ezután a következő scripttel oldottam meg a kigenerálását:

```
1      $('#pdfButton').click(function () {
2
3          var style = document.createElement('style');
4          style.innerHTML = '#journal_add,#journal_edit {display: none;}';
5
6          // Append the <style> tag to the <head> element
7          document.head.appendChild(style);
8
9          // Remove the <style> tag after 4 seconds
10         setTimeout(function () {
11             style.parentNode.removeChild(style);
12         }, 4000);
13
14         window.jsPDF = window.jspdf.jsPDF;
15
16         let srcwidth = document.getElementById('response_holder_issn').scrollWidth;
17         let pdf = new jsPDF('p', 'pt', 'a4');
18
19         pdf.html(document.getElementById('response_holder_issn'), {
20             html2canvas: {
21                 scale: 600 / srcwidth
22             },
23             callback: function () {
24                 window.open(pdf.output('dataurlnewwindow'));
25             }
26         });
27
28     });
```

A megfelelő azonosítójú gombot megkattintva, elsősorban eltüntettem azokat a gombokat amelyeket nem szeretnék a PDF-hez hozzáadni, majd behívom a jsPDF könyvtár fő függvényét a megfelelő paraméterekkel ahhoz, hogy kigenerálja A4-es formátumba a megjelölt egyedi azonosítóval rendelkező HTML div-ben szereplő elemeket, majd az egészet megnyitom egy új böngésző fülben.

6.4. Szoftver befejezése

Amiután a fent bemutatott függvények működőképes állapotba kerültek, kihagyhatlan volt a tesztelésük. A tesztelési fázis alatt kerültek a fenti állapotba. Minden lehetséges bemeneti adat variációt megpróbáltam minden függvény esetében.

A tesztelési fázis után csinosítottam, és elrendeztem őket, hogy a felhasználó számára érthető, és könnyen átlátható legyen.

6.5. Telepítés a webserverre

Amint a webhely a kész állapotát elérte, egy utolsó lépés maradt hátra, HTTP protokolon keresztül mindenki számára elérhetővé tenni.

A webhelyet a következő címen lehet elérni: <https://dev.imok.biz/dani-vizsga/>, létrehoztam egy felhasználót a könnyebb elérés érdekében, amelynek a felhasználóneve: testuni, és jelszava: university2023. A webhely elkészítésének a fázisait a következő GitHub linken lehet elérni: <https://github.com/fnascha/SJR-Rank-Checker>

Összefoglaló

Dolgozatomban egy webhely megvalósításával foglalkoztam, amelyet különböző programozási technológiák, szoftverek, és bővítmények segítségével valósítottam meg. Először ismertettem az adathalmazt, amelyet feldolgoztam, és a technológiákat, amelyeket alkalmaztam a webhely létrehozásához, és a célom eléréséhez. Majd megvizsgáltuk a rendszer specifikációit, funkcionális-, és nem funkcionális követelményeit. A továbbiakban bemutattam a webhelyet fejlesztői-, és felhasználói szemszögből is. Megmagyaráztam az általam újonnan tanult elméletet, amely szükségeltetik egy hasonló projekt létrehozásában. Következőekben megvizsgáltuk, hogy hogyan is értékelik a folyóiratokat, és bővítettük az értékelési skálát. Mindezek mellett bemutattam egy kódrészletet, amelyet felhasználtam az értékelések kiszámolásában. Ezek után részletesen is megnéztük hogy milyen szoftvereket alkottam a legnagyobb kihívást jelentő keresésekkel. A tapasztalatlanságom következtében nagy kihívásokkal járt véghez vinni az Ajax metódust, de azt hiszem kijelenthetem, hogy rengeteget tanulhattam, és legfőképp megérthettem az összetettebb aszinkron webes alkalmazások létrehozását. A függvényeket nemcsak megnéztük, hanem részletes leírást adtam a felépülésükről. A következő lépésben az annyira nem látványos, viszont a webhelyet látványossá tevő dizájnolást csak megemlítettem, a teszteléssel együtt. Majd a kész munkának publikussá tételét is lejegyeztem, elérhetőséget adva az olvasóknak.

Ábrák jegyzéke

4.1. Az Enfold téma oldal szerkesztő bővítménye.	19
4.2. Az ISSN keresés, négy egyedi kóddal, és a nevével ellátva.	21
4.3. Folyóiratok megjelenítése.	22
4.4. Folyóiratok listázása.	23
4.5. A szerzők által keresett D1-es kategóriába tartozó folyóiratok	24
F.1.1 Javascript/PHP/HTML/SCSS/CSS-Szerkesztő.	44
F.2.1 Backend.	45

Táblázatok jegyzéke

5.1. Q-index értékelési rendszere.	25
--	----

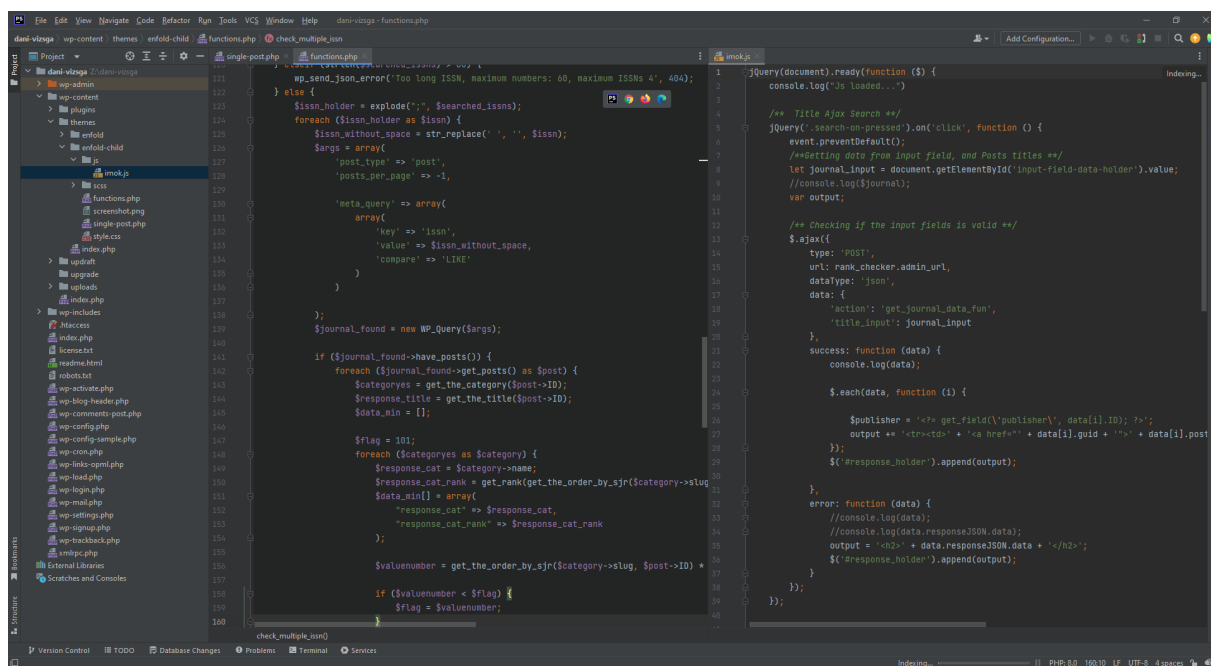
Irodalomjegyzék

- [Del22] Delicious Brains. *Advanced Custom Fields*, 2022. Available at <https://www.advancedcustomfields.com/>.
- [Fac21] FacetWP, LLC. *FacetWP*, 2 march 2021. Available at <https://facetwp.com/>.
- [Jet21] JetBrains. *PHP Storm*, 2 June 2021. Available at www.jetbrains.com/phpstorm/.
- [jQu19] Definition of jquery, March 2019.
- [jQu22] The jQuery Team. *jQuery*, 12 may 2022. Available at <https://jquery.com/>.
- [Jú14] Dr. Hornyacsek Júlia. *A tudományos kutatás elmélete és módszertana*. Nemzeti Közszerolálati Egyetem Hadtudományi és Honvédtisztképző Kar, Budapest, 2014.
- [Kri22] Kriesi. *Enfold - Responsive Multi-Purpose Theme*, 2022. Available at <https://themeforest.net/item/enfold-responsive-multipurpose-theme/4519990>.
- [Mic16] Microsoft. *Microsoft Excel*, 2016. Available at <https://www.microsoft.com/en-us/excel>.
- [Nat22] Natalie Weizenbaum, Chris Eppstein. *SCSS*, 24 February 2022. Available at <https://sass-lang.com/>.
- [Net11] Netscape Communications Corporation. *JavaScript*, 22 March 2011. Available at <https://www.javascript.com/>.
- [Ope22] OpenJS Foundation. *Node.js*, 2022. Available at <https://nodejs.org/en/>.
- [Par22] Parallax. *jsPDF*, 2022. Available at <https://www.npmjs.com/package/jspdf>.
- [PHP22] PHP Group. *PHP*, 2022. Available at <https://www.php.net/>.
- [sci23] Scimago journal rank-wikipedia, 14 January 2023.
- [Sof22] Soflyy. *WP All Import Pro*, 2022. Available at <https://www.wpallimport.com/>.
- [Ult23] Ultimate Member. *Ultimate Member*, 2023. Available at <https://ultimatemember.com/>.

- [Upd22] Updraft WP Software Ltd. *UpdraftPlus - Backup/Restore*, 2022. Available at <https://updraftplus.com/>.
- [WHA08] WHATWG. *HTML5*, 22 January 2008. Available at <https://html.com/>.
- [wik07] Php-wikipedia, 25 November 2007.
- [Wor22] WordPress Foundation. *WordPress*, 2022. Available at <https://wordpress.org/>.

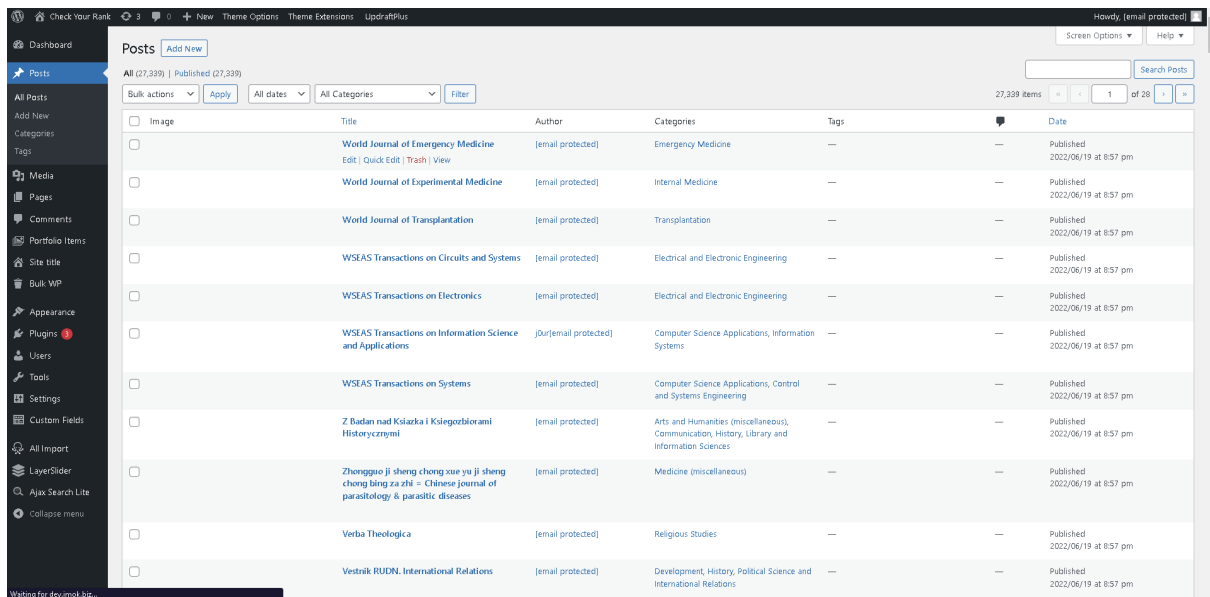
Függelék

F.1. A PHP Storm felülete



F.1.1. ábra. Javascript/PHP/HTML/SCSS/CSS-Szerkesztő.

F.2. A WordPress felülete



The screenshot shows the WordPress admin dashboard. The sidebar on the left contains navigation links: Dashboard, Posts, All Posts, Add New, Categories, Tags, Media, Pages, Comments, Portfolio Items, Site title, Bulk WP, Appearance, Plugins (with a red update indicator), Users, Tools, Settings, Custom Fields, All Import, LayerSlider, Ajax Search Lite, and Collapse menu. The top bar shows the user 'Howdy, ' and site statistics: 'Check Your Rank', '3', '0', '+ New', 'Theme Options', 'Theme Extensions', and 'UpdateWPPlus'. The main content area is titled 'Posts' and shows 'All (27,339) | Published (27,339)'. It includes filters for 'Bulk actions', 'Apply', 'All dates', 'All Categories', and 'Filter'. A search bar and 'Search Posts' button are also present. The posts list table has columns: Image, Title, Author, Categories, Tags, and Date. The first post is 'World Journal of Emergency Medicine' by '' in the 'Emergency Medicine' category, published on 2022/06/19 at 8:57 pm. Other posts include 'World Journal of Experimental Medicine', 'World Journal of Transplantation', 'WSEAS Transactions on Circuits and Systems', 'WSEAS Transactions on Electronics', 'WSEAS Transactions on Information Science and Applications', 'WSEAS Transactions on Systems', 'Z badań nad Księżką i Księgobiorami Historycznymi', 'Zhongguo ji zhong chong xue ye ji zhong chong bing za zhi - Chinese Journal of parasitology & parasitic diseases', 'Verba Theologica', and 'Vestnik RUDN. International Relations'.

Image	Title	Author	Categories	Tags	Date
<input type="checkbox"/>	World Journal of Emergency Medicine Edit Quick Edit Trash View		Emergency Medicine	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	World Journal of Experimental Medicine		Internal Medicine	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	World Journal of Transplantation		Transplantation	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	WSEAS Transactions on Circuits and Systems		Electrical and Electronic Engineering	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	WSEAS Transactions on Electronics		Electrical and Electronic Engineering	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	WSEAS Transactions on Information Science and Applications		Computer Science Applications, Information Systems	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	WSEAS Transactions on Systems		Computer Science Applications, Control and Systems Engineering	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	Z badań nad Księżką i Księgobiorami Historycznymi		Arts and Humanities (miscellaneous), Communication, History, Library and Information Sciences	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	Zhongguo ji zhong chong xue ye ji zhong chong bing za zhi - Chinese Journal of parasitology & parasitic diseases		Medicine (miscellaneous)	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	Verba Theologica		Religious Studies	—	Published 2022/06/19 at 8:57 pm
<input type="checkbox"/>	Vestnik RUDN. International Relations		Development, History, Political Science and International Relations	—	Published 2022/06/19 at 8:57 pm

F.2.1. ábra. Backend.