

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

**Automatické indexování webových stránek
mezinárodního odborného časopisu s podporou
bezbariérového webu**

Michail Utjaganov

© 2022 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Michail Utjaganov

Informatika

Název práce

Automatické indexování webových stránek mezinárodního odborného časopisu s podporou bezbariérového webu

Název anglicky

Automatic indexing of the inter journal pages of an international professional magazine with the support of a barrier-free website

Cíle práce

Cílem práce je webová aplikace pro poskytování citačních informací z mezinárodního odborného časopisu SLOVJANI.info, která umožní indexování uvedených publikací a jejich vlastností do crawler vyhledávače odborné literatury Google Scholar. Aplikace bude také dostupná pro nevidomé osoby a osoby se zrakovým postižením.

Metodika

Aplikace bude vytvořena podle standardů softwarového inženýrství pomocí PHP a MySQL. Na dokumentaci projektu bude použit UML. Aby byla aplikace vhodná pro zrakově hendikepované, bude vytvořena podle standardů Web Content Accessibility Guidelines 2.1.

Doporučený rozsah práce

40-80 stran

Klíčová slova

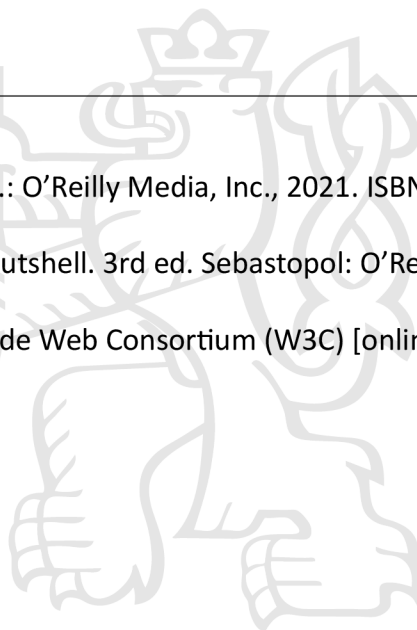
Google Scholar; crawlers; indexing; bezbariérový web

Doporučené zdroje informací

NIXON, Robin. Learning PHP, MySQL & JavaScript. 6th ed. S.l.: O'Reilly Media, Inc., 2021. ISBN 9781492093824.

SPAINHOUR, Stephen a ECKSTEIN, Robert. Webmaster in a nutshell. 3rd ed. Sebastopol: O'Reilly Media, Inc., 2002. ISBN 9780596003579.

Web Content Accessibility Guidelines (WCAG) 2.1. World Wide Web Consortium (W3C) [online]. 2018. Dostupné z: <https://www.w3.org/TR/WCAG21/>



Předběžný termín obhajoby

2021/22 LS – PEF

Vedoucí práce

doc. Ing. Vojtěch Merunka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 1. 3. 2022

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 7. 3. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 15. 03. 2022

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Automatické indexování webových stránek mezinárodního odborného časopisu s podporou bezbariérového webu" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2022

Poděkování

Rád bych touto cestou poděkoval doc. Ing. Vojtěchu Merunkovi, Ph.D. za jedinečnou příležitost vytvořit bakalářskou práci na zajímavé téma a za všechnu jeho pomoc při její tvorbě. Dále bych chtěl poděkovat Ing. Ivaně Kantnerové za znalosti které jsem od ní získal, v této práci byly nedocenitelné.

Automatické indexování webových stránek mezinárodního odborného časopisu s podporou bezbariérového webu

Abstrakt

Tato bakalářská práce se zabývá tvorbou webové aplikace pro mezinárodní odborný časopis SLOVJANI.INFO, která automaticky vytváří metadata pro indexování roboty vyhledávače Google Scholar a je přístupná dle pravidel WCAG 2.1. V teoretické části jsou vysvětleny použité technologie a programy, včetně jejich role v tvorbě webové aplikace. V praktické části je na začátku zobrazen a popsán model databáze a webové aplikace. Po modelech je vysvětlena použitá databáze, jak byla zprovozněna a upravena. Následně jsou popsány jednotné prvky, které tvoří základ všech stránek, po čemž jsou popsány individuální stránky, čemu slouží a v ní obsažené uživatelské funkcionality. U funkcí je popsáno, jaké řešení byla navrhována, jejich vyhodnocení a odůvodnění zvolené řešení. Dále jsou popsány pravidla směrnice WCAG 2.1 a úpravy které byly provedeny pro jejich dodržení a responzivní design. Ve výsledku a diskuzi je popsána provedena validace aplikace a testování aplikace na přístupnost.

Klíčová slova: Google Scholar; crawlers; indexing; bezbariérový web; HTML; CSS; PHP; MySQL

Automatic indexing of the inter journal pages of an international professional magazine with the support of a barrier-free website

Abstract

This bachelor thesis is about creating web application for international professional journal SLOVJANI.INFO, which automatically creates metadata for indexing by Google Scholar robots, and is accessible according to rules of WCAG 2.1. The theoretical part explains applied technologies and softwares, including their role in creation of web application. Beginning of the practical part shows and explains models of database and web application. After models follows explanation of used database, how it was put into operation and altered. Next is description of uniform elements, which form foundation of all pages, after which follows description of individual pages, their role and included user functionality. Solutions for functions are described and evaluated, according to which selected solution is chosen. Afterwards is explanation of WCAG 2.1 guideline rules and adjustments which were made for better accessibility and responsive design. In results and discussion is description of validation of web application and testing of accessibility.

Keywords: Google Scholar; crawlers; indexing; barrier-free web; HTML; CSS; PHP; MySQL

Obsah

1 Úvod.....	12
2 Cíl práce a metodika	13
2.1 Cíl práce	13
2.2 Metodika	13
3 Teoretická východiska	14
3.1 SQL a MySQL	14
3.2 HTML	14
3.3 CSS.....	15
3.4 PHP	15
3.5 WCAG 2.1.....	15
3.6 Google Scholar.....	16
3.7 Crawling a indexing	16
3.8 Využitý software	16
3.8.1 WebRatio	16
3.8.2 XAMPP.....	17
3.8.3 Notepad++	17
3.8.4 MySQL Workbench.....	17
3.8.5 Google Chrome.....	17
3.8.6 NonVisual Desktop Access	18
4 Vlastní práce	19
4.1 Model	19
4.2 Databáze	21
4.3 Jednotné prvky	22
4.3.1 Cascading Style Sheets	23
4.3.2 Jazyky	24
4.3.3 Vyhledávání	25
4.4 content.php - Stránka článků.....	26
4.4.1 Stránkování	27
4.4.2 Pořadí	28
4.4.3 Údaje článku	29
4.5 Search.php - Stránka vyhledávání.....	30
4.5.1 Zabezpečení slabin.....	31
4.5.2 Vyhledávání	33
4.6 Article.php - Stránka podrobnosti článku	34
4.6.1 Detaily článku	34
4.6.2 Metadata.....	35

4.6.3	Tlačítka stáhnutí.....	36
4.7	Robots.txt	37
4.8	Sitemap.php.....	38
4.9	Přístupnost.....	39
4.9.1	Vnímatelnost	39
4.9.2	Použitelnost.....	42
4.9.3	Pochopitelnost.....	43
4.9.4	Robustnost	44
4.10	Responzivní design	44
5	Výsledky a diskuse	46
5.1	Validace.....	46
5.2	Kontrola přístupnosti.....	46
5.3	Kontrola responzivního designu	47
6	Závěr.....	48
7	Seznam použitých zdrojů	49

Seznam obrázků

Obrázek 1 – UML model databáze	19
Obrázek 2 – IFML model webové aplikace vytvořen ve WebRatio	20
Obrázek 3 – ER Diagram databáze vygenerován v MySQL Workbench	21
Obrázek 4 – Ukázka PHP kódu pro vytvoření připojení k MySQL databázi	22
Obrázek 5 – Jednotný základ stránek	22
Obrázek 6 – Ukázka CSS tříd metodiky BEM	23
Obrázek 7 – Ukázka HTML kódu pro připojení CSS souboru.....	23
Obrázek 8 – Ukázka kódu pro připojení jazykového PHP souboru a vypsání textu	25
Obrázek 9 – Ukázka stránky článků	26
Obrázek 10– Ukázka PHP kódu pro načtení a výpočet offsetu	28
Obrázek 11 – Příkaz pro načtení článků	29
Obrázek 12 – Ukázka stránky po vyhledání termínu „Interslavic“	30
Obrázek 13– Ukázka nezabezpečeného SQL příkazu v PHP	31
Obrázek 14 – Ukázka zabezpečeného SQL příkazu v PHP pomocí připraveného příkazu	31
Obrázek 15 – Ukázka nesprávného a správného způsobu napsání znaku < a > v HTML ..	32
Obrázek 16 – Ukázka neošetřeného a ošetřeného výpisu hodnoty v HTML přes PHP	32
Obrázek 17 – Ukázka kódu pro načtení a ošetření vyhledávané hodnoty v PHP	33
Obrázek 18 – Ukázka stránky článku	34
Obrázek 19 – Ukázka metadat v zdrojovém kódu stránky	35
Obrázek 20 – Obsah souboru robots.txt.....	37
Obrázek 21 – Ukázka obsahu sitemap.php	38
Obrázek 22 – Ukázka definování formátu obsahu a vypsání XML objektu	39
Obrázek 23 – Ukázka sekcí stránky	40
Obrázek 24 - Ukázka HTML kódu a CSS tříd pro nastavitelnou velikost textu	41
Obrázek 25 – Meta element pro umožnění responzivního zobrazení	44
Obrázek 26 - Ukázka gridbox s uvedenými grid-area	45
Obrázek 28 – Seznam vstupů a sekcí stránky v NVDA	46
Obrázek 29 – Zobrazení stránky v Google Chrome na Android zařízení	47

Seznam tabulek

Tabulka 1 - Vytvořená metadata.....	36
-------------------------------------	----

Seznam použitých zkratk

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

PHP – PHP: Hypertext Preprocessor

SQL – Structured Query Language

WCAG – Web Content Accessibility Guidelines

URL – Uniform Resource Locator

SEO – Search Engine Optimization

UML – Unified Modeling Language

IMFL – Interaction Flow Modeling Language

BEM – Block Element Modifier

ARIA – Accessible Rich Internet Applications

1 Úvod

Vznik internetu změnil, jakým způsobem vyhledáváme informace. Dřív aby se člověk mohl dozvědět věci o konkrétním tématu, bylo nutné jít do knihovny nebo knihkupectví pro knihu, nebo koupit časopis na dané téma v trafikách. Dnes lze získat takové informace jednoduchým požadavkem ve webových vyhledávačích. Ale pod nespočtelným počtem výsledku, který nám takový vyhledávání může nabídnou se jednoduše ztrácí profesionální a odborné zdroje mezi zdroji, který mohou být neoborné, nekvalitní a někdy dokonce i nepravdivý. Tímto se ztrácí určitá úroveň kvality, která byla zaručená profesionálním tiskem publikací. Z tohoto důvodu vznikl specializovaný vyhledávač Google Scholar. Tento vyhledávač umožňuje vyhledávat zdroje které dodržuje stanovené podmínky. Aby bylo možné najít publikace v tomto vyhledávači, je nutné zveřejňovat informace o nich, a to buď ve formátu metadat ve webové stránce, nebo veřejně dostupný PDF soubor který dodržuje stanovené podmínky formátování údajů. Přítomnost v tomto vyhledávači umožňuje vyšší šanci, že vaše publikace bude nalezena těmi, kdo by se o ní mohli zajímat, obzvlášť ti, kteří hledají profesionální a odborný zdroj.

Mezi těmi, co by mohly takové zdroje vyhledávat mohou být i lidé omezený zdravotním postižením. Ať už zrakové, sluchové nebo tělesné, tyto postižení omezují možnosti interakce uživatele s webovými stránky. Aby mohli lidé tímto způsobem omezení taky mít možnost využít těchto webových stránek, vznikl standard Web Content Accessibility Guideline, který umožnil vývojářům webových stránek a aplikaci pochopit, co můžou udělat, aby byla jejich tvorba víc přístupná těmto lidem.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této bakalářské práce je vytvořit webovou aplikaci pro mezinárodní odborný časopis SLOVJANI.INFO, která umožní indexování článků do vyhledávače Google Scholar. Tato aplikace bude vytvořena dle standardů Web Content Accessibility Guidelines 2.1, aby byla přístupná osobám se zdravotním postižením.

2.2 Metodika

Nejdříve bude vytvořen model databáze a webové aplikace. Model databáze bude vytvořen v UML, aby byla zobrazena relace mezi daty. Následně bude vytvořen IMFL model aplikace, pomocí čeho bude definována struktura a interakce uživatele s webovou aplikací. Jakmile je určena struktura webové aplikace, připraví se lokální instance MySQL databáze s daty, která bude rozšířena prvky potřebnými pro tuto webovou aplikaci. Následovat bude vytvoření samotné webové aplikace pomocí PHP, HTML a CSS. Aplikace bude vytvořena tak aby byla responzivní a dodržovala požadavky směrnice WCAG 2.1. Na závěr bude webová aplikace a validována a testována na přístupnost pomocí nástrojů Google Lighthouse a Axe DevTools.

3 Teoretická východiska

Pro vývoj této webové aplikace byly využity různé technologie a programy, které jsou v této kapitole popsány. U technologií bude vysvětlena krátce jejich historie, funkce a vlastnosti. U programů bude vysvětlena jejich funkce a role v tvorbě této webové aplikace.

3.1 SQL a MySQL

První relační databáze byla vytvořena v roce 1970 výzkumníkem IBM Edgarem Frank Coddem. Tehdy jazyk pro tuto databázi byl znám jako SEQUEL – Structured English Query Language. Tento jazyk prošel několika standardizacemi a později byl přejmenován na SQL – Structured Query Language (1). Z toho jazyka vzniklo široký počet různých systémů pro správu relačních databází, jako Microsoft SQL Server, Oracle SQL a MySQL, každý se svými unikátními funkcemi, ale základním principem stejný. MySQL je jednou z relativně novějších relačních databázových systémů. MySQL je volně distribuovaný open source systém pro správu relačních databází, patřící společnosti Oracle Corporation. Výhodou MySQL je že je jedním z jednodušších systémů k nainstalování, výkonově méně náročných a datově menších databázových systému, díky čemu lze bez obtíží provozovat jednoduchou databázi i na slabším hardwaru (2 str. 3). Pro tento databázový systém existuje software jménem MySQL Workbench, který nabízí jednoduchý a srozumitelný grafický prostředí pro práci s MySQL databázemi.

3.2 HTML

HyperText Markup Language (HTML) vznikl v Evropské organizaci pro jaderný výzkum jako odpověď na potřebu po autorském jazyku a distribučním systému pro tvorbu a sdílení multimediálních integrovaných elektronických dokumentů přes internet (3 str. 3). Od této doby prošel tento značkovací jazyk změnami a standardizací, až na dnes využívaný standard HTML 5. HTML dokument se skládá z prvků, neboli elementů, který jsou začaty a zakončený pomocí příslušného značky, s doplňitelnými prvky nebo textem mezi značky, závisle podle typu prvku (3 str. 16).

3.3 CSS

První verze Cascading Style Sheets vznikla v roce 1996, jejíž nejnovější standardizovaná verze je CSS3. Tento jazyk pro stylování umožňuje definovat vzhled stránky. Funguje tak, že určíme selektor, který může být HTML element ke kterému se tento styl přiřadí, přiřazený ID elementu, nebo třída která je následně v HTML kódu v atributu „class“ přiřazena k zvolenému elementu. Po selektoru následují deklarace, pomocí kterých definujeme hodnotu vlastností, jako například typ fontu, barva pozadí, dokonce i pozice elementu. Díky tomuto jazyku lze velice flexibilně změnit vzhled a strukturu stránky podle potřeby (4).

3.4 PHP

PHP: Hypertext Preprocessor je skriptovací jazyk, který umožňuje tvorbu dynamického obsahu webové stránky. Pomocí tohoto jazyku lze jednoduše vytvořit komunikaci s SQL databázemi, a vytvářet stránky s obsahem načtený z databázi. Syntaxe tohoto jazyku jsou inspirovány z programovacích jazyků jako C, Java a Perl. Původně tento jazyk vytvořil v roce 1994 Rasmus Lerdorf, jako nástroj pro sledování počtu návštěvníků na jeho online resumé, tehdy tento jazyk byl znám jako „Personal Home Page Tools“. Rasmus v červnu 1995 publikoval zdrojový kód, a od té doby prošel tento jazyk mnoha změnami (5). Nejnovější verze tohoto jazyku byla publikována 25 listopadu 2021, jako PHP 8.1. V rámci této bakalářské práce se pracuje s verzí PHP 7.4.5.

3.5 WCAG 2.1

Aby mohli vývojáři webových aplikací a stránek vědět, co můžou udělat, aby jejich stránky byly víc přístupné, vytvořila skupina Web Accessibility Initiative konsorcia World Wide Web směrnici Web Content Accessibility Guidelines. Tato směrnice obsahuje doporučení, pomocí kterých je možný udělat stránku víc přístupnou pro osoby nejen se zrakovým postižením, ale i se sluchovým postižením a omezenou schopností pohybu. WCAG 2.1 je rozšířením směrnice WCAG 2.0 která byla publikována prosinci roku 2008. Kromě podmínek tato směrnice obsahuje doporučení, jak lze tyto podmínky dodržet i za jakých okolností může dojít k jejich nedodržení (6).

3.6 Google Scholar

Google Scholar je specializovaný webový vyhledávač, určený pro vyhledávání naučné literatury. Tento vyhledávač eviduje a umožňuje vyhledávání podle bibliografických údajů, jako autor, instituce a víc. Evidují se zde citace, což umožňuje autorovi sledovat jaký jiný autoři citovala jeho dílo. Publikace jsou zahrnuty do vyhledávání pomocí robotu jako „crawler“ a „parser“ (7).

3.7 Crawling a indexing

Pro automatizované nacházení stránek a jejich zahrnutí do vyhledávání byly vytvořeny specializované roboty. Konkrétně Google Scholar využívá dva typy robotu – „crawler“ a „parser“. Crawler nachází a prohledává přes všechny dostupné webové stránky a PDF soubory, při čem sbírá jejich základní údaje jako titulek, pro zobrazení když bude daná stránka ve výsledcích vyhledávání (8). Oproti tomu parser sbírá podrobné bibliografické údaje publikace. Nejvhodnější je tyto údaje uvádět pomocí meta značek v elementu „head“ stránky, ale parser je také schopen údaje zpracovat z PDF souborů a textu stránky, pokud jsou dodrženy stanovené formátové podmínky (9). V této aplikaci budou bibliografická data vytvořena ve meta značkách.

3.8 Využitý software

Při tvorbě této bakalářské práce byly využity různé softwary, pomocí kterých byly navrženy, vytvořeny nebo testovány různé prvky webové aplikace.

3.8.1 WebRatio

Webratio je software pro navrhování a vytváření webových a mobilních aplikací pomocí low-code technologie. Tento program umožňuje vytváření modelu webových aplikací podle standardu Interaction Flow Modeling Language, kterým se definuje obsah a interakce uživatele ve front-end aplikaci (10). Pomocí tohoto programu byl navržen model webové aplikace.

3.8.2 XAMPP

XAMPP je volně dostupná open source distribuce Apache web serveru, obsahující PHP, MariaDB a Perl. Tato distribuce umožňuje jednoduché zprovoznění lokálního Apache serveru s různými doplňky, ideální pro tvorbu a testování webových aplikací (11). Tento program byl využit pro lokální testování webové aplikace.

3.8.3 Notepad++

Notepad++ je volně dostupný textový editor. Výhodou tohoto textového editoru je že podporuje kolem 80 různých programovacích jazyků, včetně HTML, CSS, PHP a SQL, a umožňuje tím přehledné barevné zvýraznění syntaxe (12). V tomto programu byl napsán PHP, HTML a CSS kód webové aplikace.

3.8.4 MySQL Workbench

MySQL Workbench je grafický program od společnosti Oracle pro práci s MySQL databázemi. Tento software nabízí nástroje pro data modelování, SQL vývoj, administraci serveru a víc (13). Tento program byl použit pro vytvoření lokální instance databáze a programování SQL kódu.

3.8.5 Google Chrome

Google Chrome je webový prohlížeč vytvořený společností Google v roce 2008, a je jedním z nejrozšířenějších webových prohlížečů (14). V prosinci roku 2021 tento prohlížeč tvořil 63 % všech používaných prohlížečů (15). Největší výhodou tohoto prohlížeče z vývojářského hlediska je dostupnost nástrojů pro ulehčení tvorby webové stránky, jako možnost zobrazení zdrojového kódu, simulace zobrazení stránky na mobilním zařízení a testování a optimalizování stránky pomocí nástroje Lighthouse. Dále jsou v tomto prohlížeči dostupný uživateli vytvořené rozšíření, včetně Axe DevTools, nástroje pro testování přístupnosti webové stránky. Tento prohlížeč je dostupný i na mobilních Android zařízeních. Tento prohlížeč byl použit pro testování webové aplikace, včetně jeho Android verze pro testování kompatibility s mobilními zařízeními.

3.8.6 NonVisual Desktop Access

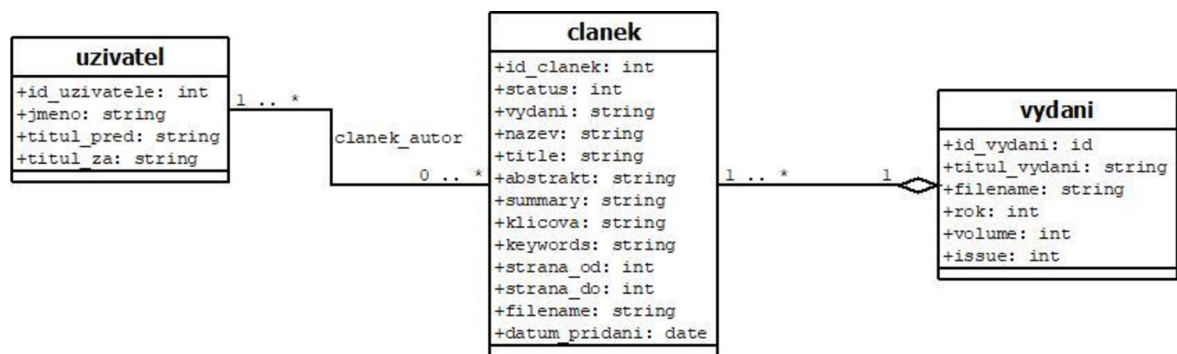
NonVisual Desktop Access (NVDA) je volně dostupná čtečka obrazovky pro zrakově postižené, umožňující převedení textového obsahu do zvukového využitím technologie „text-to-speech“. Tento program podporuje přes 50 různých jazyků a umožňuje čtení obsahu programů jako Google Chrome, Microsoft Office, e-mailový klienty a přehrávače hudby (16). Pomocí tohoto programu byla testována srozumitelnost a funkčnost prvků webové aplikace pro čtečky obrazovky.

4 Vlastní práce

Prvním krokem pro tvorbu webové aplikace bylo určení struktury a funkcí pomocí vytvoření modelu webové aplikace. Následně byla vytvořena lokální instance databáze časopisu, u který se pomocí UML modelu popsaly vztahy mezi daty. Jakmile byla databáze zprovozněna, následovala tvorba samotné webové aplikace po částech, nejdřív jednotný základ všech stránek, potom konkrétně stránky pro výpis článků, vyhledávání článků a podrobnosti článku. Následně byly tyto stránky upraveny, aby dodržovali směrnici WCAG 2.1 a měli responzivní design pro mobilní zařízení.

4.1 Model

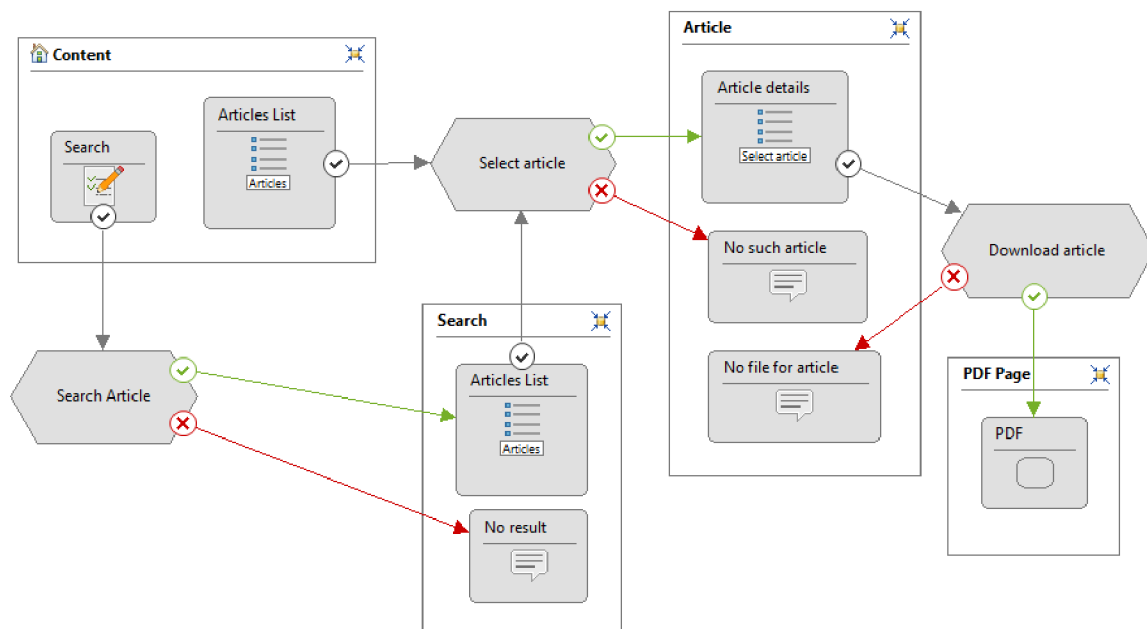
Obrázek 1 – UML model databáze



Zdroj: Vlastní zpracování, 2022

Pro zobrazení vztahu mezi tabulkou databáze byl vytvořen UML model, který obsahuje tabulky a jejich atributy potřebné pro webovou aplikaci. Kromě importovaných tabulek model obsahuje taky nově vytvořenou tabulku „vydani“ potřebnou pro tuto webovou aplikaci.

Obrázek 2 – IFML model webové aplikace vytvořen ve WebRatio



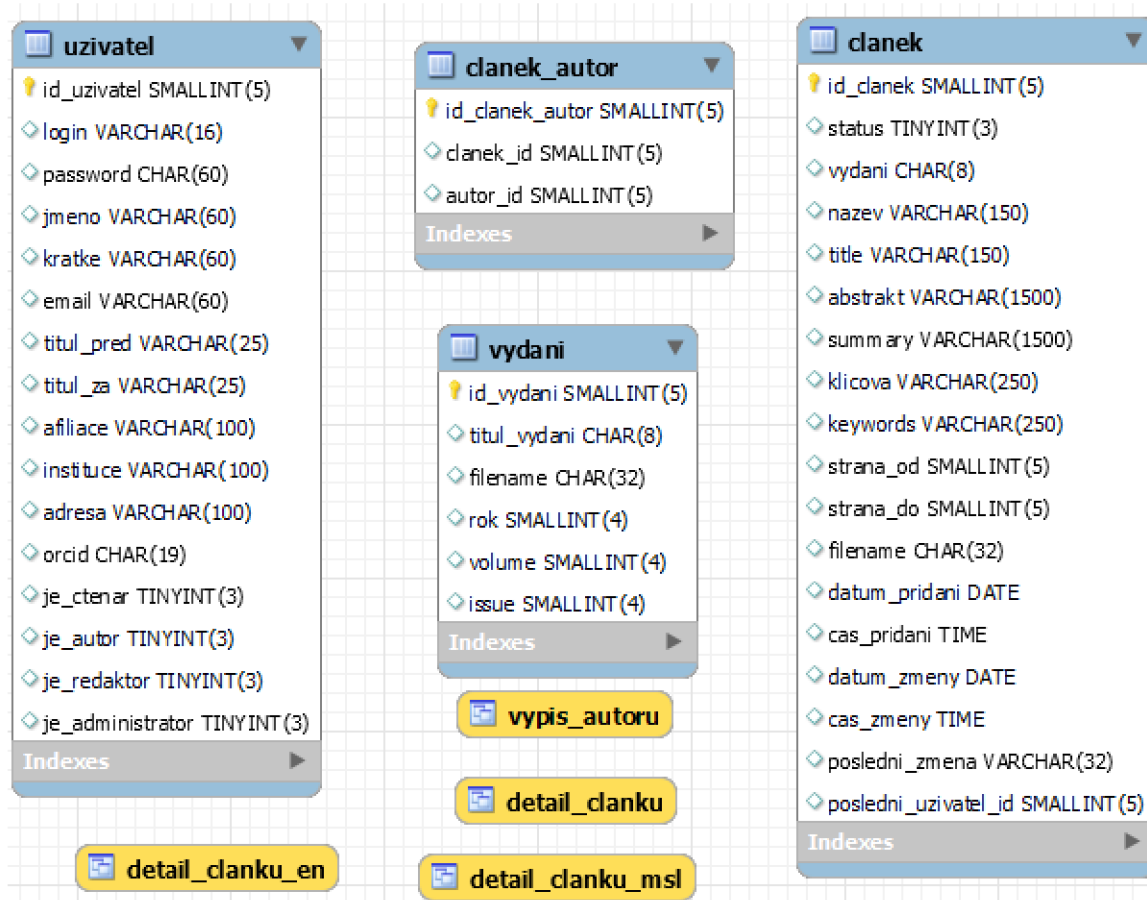
Zdroj: Vlastní zpracování, 2022

Model samotné webové aplikace byl vytvořen v IFML pomocí programu WebRatio. Jsou zde definovány stránky aplikace, prvky, způsoby interakce uživatele a jejich výsledky. Hlavní stránkou této webové aplikace je stránka Content, kde uživateli bude zobrazen seznam článků. Uživatel kromě zvolení článku bude mít možnost konkrétně vyhledat požadovaný článek. Vyhledání přesune uživatele na stránku Search, kde bude buď uživateli zobrazeny odpovídající články nebo upozornění že žádný článek nebyl nalezen. Stejně jako na stránce Content i zde si může uživatel zvolit konkrétní článek, čímž se přesune na stránku Article. Zde budou zobrazeny údaje článku a možnost stáhnout jeho PDF soubor. Pokud bude článek dostupný ve formátu PDF, bude možné ho stáhnout, pokud ne tak uživatel bude upozorněn že PDF soubor není dostupný.

4.2 Databáze

U této webové aplikace se pracuje už s existující databází, která byla exportována do .sql souboru pro importování do lokálně hostované databáze, aby bylo možné s ní pracovat bez ovlivnění běhu hlavní stránky. Jelikož na této databázi závisí další stránky, jakékoliv úpravy do existující struktury jsou nežádoucí. Tato databáze nemá explicitně definované vztahy mezi tabulky neboli cizí klíče.

Obrázek 3 – ER Diagram databáze vygenerován v MySQL Workbench



Zdroj: Vlastní zpracování, 2022

Aby bylo možné získat data v praktických formách pro webovou aplikaci, aniž by PHP kód obsahoval dlouhé příkazy SELECT, bylo vytvořeno několik pohledu, neboli VIEW. Pohledy umožňují vytvořit defacto virtuální tabulku, která místo toho aby zobrazovala vlastní data, zobrazuje data načtená z jiných tabulek podle nastavených podmínek. Pohled se vytváří pomocí funkce CREATE VIEW, kde je nastaveno jméno pohledu a následně

napsán SELECT příkaz který definuje obsah pohledu. Pro webovou aplikaci byly vytvořeny 4 pohledy – „detail_clanku_msl“, „detail_clanku_en“, „vypis_autoru“ a „detail_clanku“.

Obrázek 4 – Ukázka PHP kódu pro vytvoření připojení k MySQL databázi

```
$db = new mysqli($host, $user, $password, $dbname, $port, $socket);
```

Zdroj: Vlastní zpracování, 2022

Pro připojení k databázi bylo využito existující řešení pro připojení. V složce includes je soubor „connect.php“, ve kterém je vytvořené připojení k databázi v objektově orientovaném stylu. Tento soubor je následně pomocí funkce „include“ zahrnut do stránky. Tento způsob připojení umožňuje nutnost změnit údaje pouze v jednom souboru v případě změny parametrů připojení.

4.3 Jednotné prvky

Obrázek 5 – Jednotný základ stránek



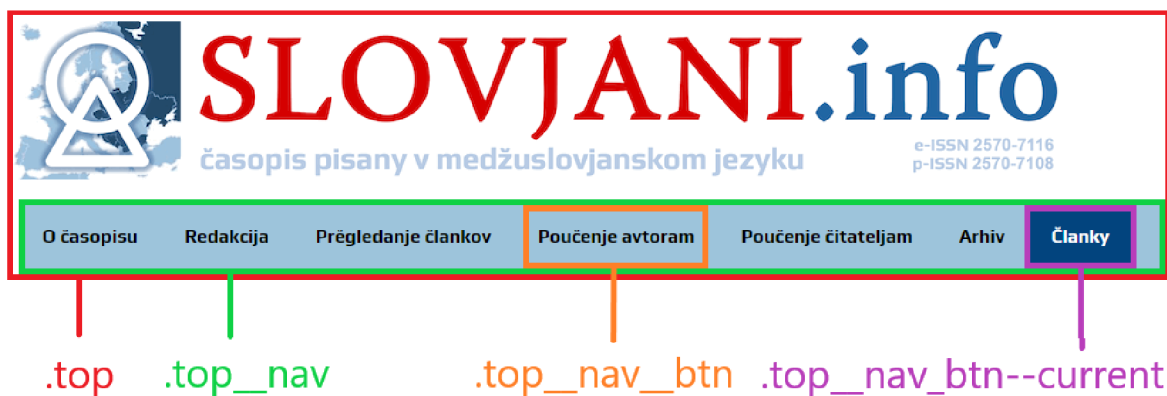
Zdroj: Vlastní zpracování, 2022

Všechny stránky této aplikace mají jeden stejný základ, tvořený zápatím a hlavicí, kde může uživatel pomocí tlačítek zvolit ve jazyk stránky a vyhledávání, kde může vyhledávat konkrétní články podle vlastností. Dále zde může najít navigační lištu, logo

časopisu SLOVJANI.INFO a tlačítko pro přeskočení k hlavnímu obsahu stránky. Po hlavici následuje hlavní obsah stránky, který bude popsán individuálně u každé stránky. Na konci stránky je zápatí, kde lze najít tlačítka pro změnu velikosti fontu, tlačítko pro vrácení na vrch stránky a logo WCAG 2.1 který označuje že tato stránka dodržuje směrnici na úrovni AAA.

4.3.1 Cascading Style Sheets

Obrázek 6 – Ukázka CSS tříd metodiky BEM



Zdroj: Vlastní zpracování, 2022

CSS bylo napsáno podle metodiky BEM. Touto metodikou je možné vytvořit přehlednější kód, kde je CSS třída pojmenována nejdřív názvem bloku, po kterém následuje jméno buď další bloku nebo elementu rozdělený dvojitým podtržítkem. Následně pokud upravujeme samotný element, jeho modifikátory jsou odděleny čárkou (17).

Obrázek 7 – Ukázka HTML kódu pro připojení CSS souboru

```
<link rel="stylesheet" href="styles/stylecontent.css">
```

Zdroj: Vlastní zpracování, 2022

Samotný soubor CSS se připojuje v HTML pomocí elementu „link“ v hlavě HTML kódu, obsahující atribut „rel“, který definuje, jaký vztah tento soubor má k současnému dokumentu. Dále element obsahuje atribut „href“, kde se odkazuje na lokaci souboru CSS (4 str. 8). Aby byla dodržena struktura souboru existující stránky, CSS soubor je uložen ve složce „styles“.

4.3.2 Jazyky

Údaje o člancích jsou dostupný ve dvou jazycích – angličtině a mezislovanštině. Z tohoto důvodu je vhodné umožnit volbu uživateli, v jakým jazyce požaduje, aby byla stránka a články zobrazeny. Jednou z možností, jak vytvořit webovou aplikaci ve více jazycích, je vytvoření oddělné stránky pro každý jazyk. Pokud ale je rozdíl pouze v textu a struktura stránky je vždy totožná, toto řešení je nepraktické. Vhodnějším řešením je vytvoření dynamické stránky, která zobrazuje obsah v požadovaném jazyku.

Uživatel má k dispozici na vrcholu stránky tlačítka pro volbu jazyka. Po stisknutí požadovaného jazyku je stránka znovu načtená s nastavenou variabilní pro jazyk pomocí metody GET. Tato variabilní je načtena a zkontrolována, zdali obsahuje platnou hodnotu „en“ pro anglický jazyk nebo „msl“ pro mezislovanský. V případě že tato variabilní není nastavena (například při prvním otevření stránky) nebo obsahuje neplatnou hodnotu, načítá se výchozí hodnota „msl“. Tlačítko pro jazyk, který má uživatel v dané chvíli zvolen je deaktivován a zašedlý.

Aby bylo možné udržet požadovaný jazyk i na dalších stránkách, je nutné někam tento zvolený jazyk uložit. Jednou z možností je vytvoření takzvaných „cookies“. Cookies umožňují stránce uložit krátký textový řetězec v paměti prohlížeče uživatele, které lze nastavit aby byly ponechány i po vypnutí a znovuotevření prohlížeče. Do těchto cookies je možné uložit různá data, jako například zvolený design stránky, nebo uložené položky (18 str. 193). Ale jedním z jejich nejčastějších použití je pro sledování online aktivity uživatele, aby bylo možné vytvořit reklamní obsah odpovídající online aktivitě uživatele. Z tohoto důvodu v Evropské unii vznikl v roce 2002 zákon, který požaduje po stránkách, aby získalo uživatelský souhlas pro vytvoření cookies (19). Jelikož není nutné, aby byl zvolený jazyk dlouhodobě uložen, a nemáme žádný další data k uložení, tak aby nebylo nutné vytvořit prvky pro požádání souhlasu od uživatele pro cookies, bylo zvoleno ukládání dat pomocí „session“. Session funguje tak, že server webové aplikace přiřadí prohlížeči unikátní identifikátor, který je uložen do paměti prohlížeče, podobně jako cookies. Podle tohoto identifikátoru se následně na straně serveru ukládají a načítají požadované hodnoty. Tento identifikátor existuje jenom dokud uživatel nezavře prohlížeč, a jelikož neobsahuje nic víc

než unikátní identifikátor, neumožňuje sledování aktivit konkrétního uživatele, počítá se jako nezbytný cookies a není nutné u session požadovat souhlas od uživatele (18 str. 197)

Jakmile je uživatelem zvolený jazyk, je nutné načíst odpovídající obsah. První využitá metoda byla vytvoření textového souboru, který obsahoval na každém řádku jednu část textu. Tento textový soubor byl načten pomocí metody „file“. Tato metoda umožňuje načíst obsah textového souboru do pole, kde jeden prvek odpovídá jednomu řádku textu (20). Toto řešení se však prokázalo jako nepraktické, jelikož jediná možnost, jak procházet tímto polem je pomocí číselného klíče. Aby bylo možné načítat text pomocí přehlednějšího klíče, je nutné mít pole s definovaným klíčem ke každé hodnotě. Z tohoto důvodu byl vytvořen pro každý jazyk PHP soubor, který obsahuje stejnojmenné pole a klíče, ale text obsažený v poli odpovídá příslušnému jazyku. Tyto soubory jsou pojmenovány identicky, ale obsahují na konci zkratku příslušného jazyku. Aby bylo možné tyto soubory a jejich pole využít v jiných souborech, existuje v PHP funkce „include“. Pomocí této funkce je kód obsažený v zahrnutém souboru proveden jako kdyby byl na řádku kde je include, a variabilní v něm jsou dále zděděny tímto souborem. (18 str. 57)

Obrázek 8 – Ukázka kódu pro připojení jazykového PHP souboru a vypsání textu

```
include 'Languages/content_' . $language . '.php';  
echo '<h1>' . $content["PAGE_TITLE"] . '</h1>';
```

Zdroj: Vlastní zpracování, 2022

Spojením stejných vlastností jména a lokace PHP souboru s hodnotou pro jazyk je soubor s polem textového obsahu zahrnut do dané stránky, kde následně kdykoliv kdy je potřeba text, je zavolána tabulka „\$content“ s klíčem příslušného textu.

4.3.3 Vyhledávání

Funkcí vyhledávání je umožnit uživateli najít konkrétní články pomocí krátkého textu, který je použit pro nalezení shody v titulech, klíčových slovech a jménech autorů. Aby mohl uživatel definovat co hledá, na všech stránkách je dostupný formulář s textovým polem a tlačítkem pro odeslání.

Po stisknutí tlačítka pro odeslání je obsah textového pole odeslána jako variabilní „search“. Tato variabilní může být odeslána pomocí PHP dvěma metodami, GET nebo POST. Metoda GET je ideální pro hodnoty, které určují, co je zobrazeno uživateli. Tato metoda ukládá variabilní a jejich hodnoty do URL. Výhodou této metody je že pokud uživatel znovu načte stejný URL nebo stránku obnoví, hodnoty budou ponechány a bude zobrazen stejný obsah. Proto je tato metoda vhodná pro definování filtrů a vyhledávání. Oproti tomu metoda POST ukládá variabilní do těla HTTP žádosti. Variabilní není vidět v URL, a pokud uživatel stránku obnoví či znovu otevře stejný URL, tyto variabilní jsou ztraceny. Tato metoda je vhodná pro odeslání hodnot, které jsou využity hlavně v back-end a neovlivňují přímo zobrazený obsah, jako osobní údaje při registraci a přihlášení (18 str. 174). Jelikož hodnota vyhledávání ovlivňuje obsah zobrazený uživateli a je ideální, aby při obnovení stránky tato hodnota zanechána, po odeslání je uživatel přesunut na stránku vyhledávání a hodnota je odeslána pomocí metody GET. Aby mohl uživatel vědět pomocí jakých vlastností může vyhledávat, jsou v atributu „placeholder“ formuláře uvedeny vlastnosti které jsou prohledány.

4.4 content.php - Stránka článků

Obrázek 9 – Ukázka stránky článků



The screenshot shows the website SLOVJANI.info, which is a journal published in the Slavic language. The page displays a list of articles under the heading "Articles". The first article is titled "Coronavirus scepticism, fanaticism and morbid religiosity" by Job Getcha of Telmessos, published in Issue 2020 (1). The summary states: "English original from September 30, 2020, written by prof. Job of Telmessos who is an Archbishop of the Ecumenical Patriarchate in Constantinople and the Permanent Representative of the Ecumenical Patriarchate to the World Council of Churches and the Dean of the Institute for Orthodox Theology Higher Studies at Chambésy, Switzerland." The keywords are: "orthodoxy; religiosity; COVID-19; fanaticism". The same article is listed again below.

Zdroj: Vlastní zpracování, 2022

Funkcí této hlavní stránky je výpis všech článků časopisu SLOVJANI.INFO, aby mohl uživatel prolistovat a otevřít podrobnosti požadovaných článků. Pro přehlednější zobrazení je maximální možný počet článků na jedné stránce omezen na 10 článků, a uživatel má na konci stránky k dispozici tlačítka pro postupování po stránkách. Podrobnosti zobrazené uživateli odpovídají jazyku, který má zvolený. Uživatel má možnost nastavit pořadí článků buď od nejnovějšího nebo nejstaršího.

4.4.1 Stránkování

Pro lepší přehlednost výpisu článků je praktické omezit maximální počet článků na jedné stránce, a umožnit uživateli prolistování. Toto lze provést několika způsoby. První je omezit počet zobrazených stránek přes PHP, tím že omezíme počet vypsaných článků z pole výsledku příkazu SQL. Toto řešení má nevýhodu, že zbytečně načítáme všechny řádky i když potřebujeme jen konkrétní omezený počet, což způsobuje zbytečnou zátěž na databázi. Z tohoto důvodu je vhodnější omezit počet požadovaných článků přímo v SQL příkazu, čehož můžeme dosáhnout pomocí klauzule LIMIT. Tato klauzule umožňuje optimalizovat načítání dat pomocí omezení počtu načtených řádků a určení prvního načteného řádku (21).

Příkaz pro výpis článků je omezen na 10 řádků začínaje od řádku určený hodnotou variabilní „\$offset“. Hodnota této variabilní se určuje podle stránky, na které uživatel je a maximálního počtu článků na jednu stránku. Aby mohl uživatel zvolit stránku, jsou na konci hlavní části stránky tlačítka. K dispozici je deaktivovaný tlačítko současné stránky, aby mohl uživatel doplnitelně k titulu stránky vědět na které stránce se momentálně nachází. Dále jsou k dispozici tlačítka první, poslední, tři následující a předešlý stránky. Tlačítka pro zvolení první a poslední stránky jsou nastavena tak, aby se nevytvářeli, pokud se uživatel nachází 4 stránky od začátku nebo konce. Pomocí těchto tlačítek je požadovaný číslo stránky odesláno pomocí metody GET do URL stránky jako variabilní a stránka znovu načtena.

Obrázek 10– Ukázka PHP kódu pro načtení a výpočet offsetu

```
$page=1;
if(!empty($_GET["page"]) && is_numeric($_GET["page"]) && $_GET["page"]>0)
{
    if($_GET["page"] > $lastpage) {
        $page=$lastpage;
    }else{
        $page=floor($_GET["page"]);
    }
}
$offset = ($page - 1) * 10;
```

Zdroj: Vlastní zpracování, 2022

Při načtení stránky je zkontrolováno, jestli je definována hodnota pro stránku. Pokud je definována a je to číslo které je větší než 0 a menší než maximální možná stránka, hodnota je načtena do variabilní pro stránku. Maximální možná stránka se vypočítává pomocí rozdělení počtu článku, získaný s SELECT příkázem s funkcí COUNT, a maximálním počtem článku na stránku. Pokud nejsou podmínky hodnoty stránky dodrženy, nebo je hodnota prázdná, zůstane výchozí první stránka. Následně se pomocí této hodnoty vypočítá offset, který je následně zahrnut v SELECT příkazu jako číslo prvního načteného řádku.

4.4.2 Pořadí

Aby si mohl uživatel zvolit, jestli chce aby zobrazené články byly seřazeny od nejnovějšího nebo nejstaršího, na vrcholu hlavní části stránky je dostupný výběrový menu. V menu jsou dostupné možnosti na pořadí, kterou jakmile uživatel zvolí, dojde k události, která provede nastavený skript v atributu „onChange“. V tomto případě atribut obsahuje „this.form.submit()“, což okamžitě odešle formulář, a stránka se znovu načte s pořadím definovaným metodou GET. Variabilní pro pořadí je načtena, její obsah zkontrolován, jestli obsahuje platnou hodnotu ASC nebo DESC, a následně je načten do variabilní pro pořadí. Pokud je variabilní prázdná nebo obsahuje neplatnou hodnotu, využije se výchozí hodnota ASC. Parametry ASC nebo DESC lze určit v SQL příkazech s klauzulí ORDER BY pořadí načtených řádků. (2 str. 151). Tabulka článků obsahuje sloupec datum_pridani, který u článku z roku 2020 a dřív obsahuje jeden stejný neplatný datum, z toho důvodu je výpis doplnitelně seřazen podle jména vydání, který obsahuje rok a číslo vydání.

4.4.3 Údaje článku

Obrázek 11 – Příkaz pro načtení článků

```
$command = $db->prepare("SELECT * from detail_clanku_ " . $language . " order by datum_pridani " . $order . ", vydani " . $order . " limit ?, 10");  
$command->bind_param('i', $offset);  
$command->execute();  
$result = $command->get_result();
```

Zdroj: Vlastní zpracování, 2022

Výpis článku z databáze se provádí pomocí jednoho ze dvou pohledu, anglického nebo mezislovanského. Který pohled je použit záleží na nastaveném jazyku stránky. Oba pohledy obsahují stejný počet stejně pojmenovaných sloupců, který obsahují údaje v příslušném jazyku a pro případ že nebudou dostupný údaje v daném jazyku, alternativní sloupce obsahující údaje z druhého jazyku. Oba pohledy mají podobný název, rozlišují se v záponě, kde je určen jazyk. Tato zápona je připojena k názvu tabulky v připraveném příkazu, a následně je definováno pořadí a který řádky jsou požadovány podle stránky. Hodnoty pro jazyk a pořadí musí být vloženy přímo do příkazu, a ne přes funkci „bind_param“. Důvodem je že to nejsou data, ale parametry. Ale jelikož tyto variabilní jsou ošetřeny, aby mohli obsahovat pouze stanovené hodnoty, nedochází tu k bezpečnostní hrozbě.

Společně s těmito údaji se vypisuje i jméno autorů, které jsou načteny pomocí pohledu „vypis_autoru“. Jelikož jsou tituly a jména uložena v oddělném sloupci, je nutné tyto řádky sjednotit. Toho lze dosáhnout pomocí funkce „CONCAT_WS“, kde jsou definovány které hodnoty jsou spojeny, s jakým oddělovačem. V tomto pohledu se spojují sloupce „titul_pred“, „jmeno“ a „titul_zat“, rozdělený mezerou. Jelikož ale ne všichni autoři mají titul, docházelo zde k případům, kde jména bez titulů měla na začátku mezeru. Aby tyto zbytečné mezery byly odstraněny, obsah tohoto sloupce doplnitelně prochází funkcí „TRIM“, která odstraní zbytečné mezery na začátku a konci. Všechny údaje článku jsou následně uvedeny ve svém elementu „article“ s odkazem na stránku článku. Údaje jsou rozpoloženy v tomto elementu pomocí CSS modulu „Gridbox“.

4.5 Search.php - Stránka vyhledávání

Obrázek 12 – Ukázka stránky po vyhledání termínu „Interslavic“

Search result: interslavic

Order from:

On the question of the Interslavic language orthography

Author: Nikolay Vladimirovich Kuznetsov MA in history

Issue: 2020 (1)

Summary: This article describes various interslavic orthographies and alphabets. Its goal is to start a new discussion on this topic.

Keywords: Interslavic language; Cyrillic; Latin; orthography; Vuk's alphabet

Interslavic slang and the processes of its appearance

Author: Anton Vladimirovich Saprykin

Issue: 2020 (1)

Summary: This article is an analysis of the reasons for the appearance of slang in the languages of the world, a list of modern slangs of the Interslavic language of the Discord channel, as well as a list of examples of the use of slang, some invented for the article, some of the examples used are taken from oral speech.

Keywords: Interslavic slang; slang Discord channel; slang Telegram channel; slang of the Interslavic language community; colloquial Interslavic language; oral Interslavic language; oral speech of the Interslavic language

Editor's foreword

Author: assoc. prof. Vojtěch Merunka Ph.D.

Issue: 2019 (2)

Summary: This text refers to the 30th anniversary of the fall of communism; also about the new template of our journal; as well as questions and issues related to our Interslavic orthography (spelling).

Keywords: 30th anniversary of the fall of communism; SLOVJANI.info new journal template; news; Interslavic orthography (spelling)

Zdroj: Vlastní zpracování, 2022

Funkce této stránky je zobrazení článku odpovídající požadavkům vyhledávání. Uživatel se může dostat na tuto stránku napsáním požadovaného termínu a buď kliknutí tlačítka pro odeslání formuláře nebo stisknutím tlačítka ENTER. Na vrcholu hlavní části stránky je uvedeno uživateli, jaký termín vyhledává. Články jsou vypsány stejným způsobem jako na stránce Content.php, ale v tomto případě nejsou stránkovány. Vyhledávání je provedeno přes data odpovídající jazyku stránky zvolený uživatelem.

4.5.1 Zabezpečení slabin

Jelikož není uživatel omezen tím, co napíše do vyhledávání, vzniká zde bezpečnostní slabina ve formě SQL injection a risk interpretování speciálních HTML znaků.

SQL injection je bezpečnostní slabina, kdy hodnota v SQL příkazu je interpretována jako příkaz s potenciálně škodlivým efektem. Lze tímto způsobem napáchat škodu v databázi, zobrazit data ke kterým nemá mít uživatel přístup, a v některých případech i dokonce obejít zabezpečení, například u přihlašování (18 str. 292).

Obrázek 13– Ukázka nezabezpečeného SQL příkazu v PHP

```
$result=$db->query("SELECT * FROM clanky WHERE titul = " . $_GET["search"] . "');
```

Zdroj: Vlastní zpracování, 2022

Pokud by uživatel vyhledal „2020 (1)“, nezabezpečený příkaz by správně našel články tohoto vydání. Ale pokud by uživatel napsal „; DROP TABLE clanky --“, došlo by k smazání tabulky článků, jelikož je hodnota ve vyhledávání interpretována jako příkaz. Pro zabezpečení této slabiny existuje v PHP „Prepared statement“, neboli připravený příkaz. U těchto příkazů je možné určit v jakých místech bude variabilní hodnota a definovat její formát. Tímto způsobem nemůže dojít k jejímu interpretování jako kód.

Obrázek 14 – Ukázka zabezpečeného SQL příkazu v PHP pomocí připraveného příkazu

```
$command = $db->prepare("SELECT * from claneek where vydani = ?");  
$command->bind_param('s', $_GET["search"]);  
$command->execute();
```

Zdroj: Vlastní zpracování, 2022

Pomocí znaku otazníku jsou v připravených příkazech označena pozice hodnoty, u které je následně pomocí „bind_param“ definován nejdřív formát příslušným písmenem,

například „s“ pro „string“, a následně samotná hodnota. Příkaz je proveden pomocí „execute“. Tento příkaz nemůže interpretovat hodnotu jinak než jako požadovanou hodnotu, a tímto je zabezpečen proti SQL injection. (18 str. 293).

Obrázek 15 – Ukázka nesprávného a správného způsobu napsání znaku < a > v HTML

```
<h1><example></h1>  
<h1>&lt;example&gt;</h1>
```

Zdroj: Vlastní zpracování, 2022

Přítomnost speciálních HTML znaku které nejsou „escaped“ může způsobit nesprávné zobrazení stránky. Charaktery jako <, > a & mají speciální roli v HTML. Znaky větší a menší než označují začátek a konec značek v HTML, ampersand označuje začátek HTML entity. Jejich přítomnost v textu může způsobit nesprávnou interpretaci HTML kódu, proto když jsou použity v prostém textu, musí být zaměněny jejich HTML entitou, tvořenou znakem ampersand s jejich názvem nebo Unicode číslem zakončený středníkem. (3 str. 22)

Obrázek 16 – Ukázka neošetřeného a ošetřeného výpisu hodnoty v HTML přes PHP

```
$example = "<example>";  
echo "<h1>" . $example . "</h1>";  
echo "<h1>" . htmlspecialchars($example) . "</h1>";
```

Zdroj: Vlastní zpracování, 2022

Aby nebylo nutné vytvářet složité řešení pro úpravu variabilních obsahující text s těmito znaky, existuje v PHP funkce „htmlspecialchars“. Tato funkce změní charaktery, který mají speciální roli v interpretaci za jejich HTML entitu, a tím umožní vypsání hodnoty bez toho aby mohli způsobit potíže při interpretaci (22).

4.5.2 Vyhledávání

Obrázek 17 – Ukázka kódu pro načtení a ošetření vyhledávané hodnoty v PHP

```
$search = null;
if(!empty($_GET["search"]))
{
    $search = $_GET["search"];
}
$searchvalue= "%" . $search . "%";
$search = htmlspecialchars($search);
```

Zdroj: Vlastní zpracování, 2022

Nacházení článku odpovídající hledanému termínu se provádí pomocí operátoru LIKE v příkazu SELECT na pohledu odpovídající současně zvolenému jazyku stránky. LIKE funguje tak že vyhledává shodu mezi definovaným textovým řetězcem a řetězcem obsažený v zvoleném sloupci tabulky. Vyhledávaný textový řetězec může být doplněn takzvanými „wildcard“ charaktery, který umožňují flexibilní vyhledávání. V MySQL jsou dostupné dva tyto charaktery, % který označuje žádný a víc jakýkoliv znak a _ který označuje přesně jeden libovolný znak (2 str. 145). Jelikož pro vyhledávání z bezpečnostních důvodů používáme připravený příkaz, tyto znaky musí být doplněný k vyhledávanému termínu předem, protože pokud je v připraveném příkaze otazník označující parametr obklopen těmito znaky, připravený příkaz ho není schopný rozpoznat jako pozici pro hodnotu. Připravený příkaz je navrhnout tak aby prohledával pomocí této metody po shodách v titulech, klíčových slovech a jménech autorů v pohledu zvoleného jazyka. Jelikož jména autoru jsou zahrnuta v oddělné tabulce, je příkaz navržen tak aby vypsal články jejichž ID bylo nalezeno ve vnořeném SELECT příkazu provedený na pohledu „vypis_autoru“.

4.6 Article.php - Stránka podrobnosti článku

Obrázek 18 – Ukázka stránky článku

Medžuslovjanský sleng i procesy jeho pojavjenja

English title:

Interslavic slang and the processes of its appearance

Author:

Anton Vladimirovič Saprykin

Issue:

2020 (1)

Summary (Interslavic):

Tutój članok jest analizuju povodov pojavjenja slengov v jazykah světa, spisom dnešnih slengov medžuslovjanskógo jezyka v Diskord kanalu, a takože spisom primérov koristanja slengov, někotóre priméry koristanja sūt napisanými za članok, a někotóre sūt stenografijeju ustnógo govora.

Summary:

This article is an analysis of the reasons for the appearance of slang in the lan-guages of the world, a list of modern slangs of the Interslavic language of the Discord channel, as well as a list of examples of the use of slang, some invented for the article, some of the examples used are taken from oral speech.

Keywords (Interslavic):

medžuslovjanský sleng; sleng Diskord kanala; sleng Telegram kanala; sleng občiny medžuslovjanskógo jezyka; razgovorný jezyk; razgovorný medžuslovjanský jezyk; ustný medžuslovjanský jezyk; ustný govor medžuslovjanskógo jezyka

Keywords:

Interslavic slang; slang Discord channel; slang Telegram channel; slang of the Interslavic language community; colloquial Interslavic language; oral Interslavic language; oral speech of the Interslavic language

No article PDF available

Download issue PDF
Pages: 25-34

Zdroj: Vlastní zpracování, 2022

Funkce této stránky je zobrazení všech vlastností článku a vytvoření metadat pro indexování Google Scholar. Uživatel se může dostat na tuto stránku kliknutím odkazu článku na stránce článků a vyhledávání. Kromě údajů o článku v obou jazycích jsou na této stránce dostupná tlačítka pro stáhnutí PDF souboru samostatného článku nebo vydání který tento článek obsahuje.

4.6.1 Detaily článku

Výpis podrobnosti o článku se provádí pomocí SQL pohledu „detail_clanku“, který kromě dat z tabulky „clanky“ vypisuje doplnitelná data o vydání z nové tabulky „vydani“, která je připojena pomocí JOIN. Pomocí WHERE je omezen pohled na vypisování článku se hodnotou 3 ve sloupci „status“, aby bylo možné vypsat pouze zveřejněné články, stejně jako ostatní pohledy. Taktéž se vypisuje jména autoru článku, který se provádí pomocí pohledu „vypis_autoru“. Tyto pohledy jsou obsaženy v připravených dotazech, ve kterých

se definuje identifikátor článku. Tento identifikátor je načten pomocí metody GET, a následně zkontrolován že obsahuje platnou numerickou hodnotu. V případě že uživatel nastaví nečíselnou hodnotu do této variabilní, je upozorněn že požádal neplatnou hodnotu. Všechny vypsané řádky jsou obsaženy v podmínce „if“, který kontroluje, pokud vypisovaný řádek nemá data. Pokud tento údaj je prázdný, není pro něj vytvořen řádek, aby nedocházelo k zobrazování prázdných řádků.

4.6.2 Metadata

Obrázek 19 – Ukázka metadat v zdrojovém kódu stránky

```
...<!DOCTYPE html> == $0
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" href="images/icons/favicon.ico">
    <link rel="stylesheet" href="styles/stylecontent.css">
    <script src="scripts/responsive.js"></script>
    <title>Article: До питання о правописах междусловянского языка</title>
    <meta name="description" content="Article: До питання о правописах междусловянского языка">
    <meta name="citation_title" content="До питання о правописах междусловянского языка">
    <meta name="citation_publication_date" content="2020">
    <meta name="citation_author" content="Nikolay Vladimirovich Kuznetsov">
    <meta name="citation_journal_title" content="SLOVJANI.info">
    <meta name="citation_issn" content="2570-7116">
    <meta name="citation_volume" content="5">
    <meta name="citation_issue" content="1">
    <meta name="citation_firstpage" content="5">
    <meta name="citation_lastpage" content="13">
    <meta name="citation_pdf_url" content="https://slovjani.info/archive/2020-1.pdf">
  </head>
  <body class="text--multiplier--1">
    <header class="top">...</header>
    <main id="maincontent" class="center">...</main>
    <footer class="foot">...</footer>
  </body>
</html>
```

Zdroj: Vlastní zpracování

Důležitou funkcí pro tuto stránku je vytvoření metadat pro indexování Google Scholar. Metadata jsou uložena v elementu „head“, kde jsou vytvořeny elementy „meta“ obsahující atribut „name“ pro značku typu údaje a atribut „content“ pro samotný údaj. V metadatech jsou vypsané údaje v mezislovanštině. Pro výpis údajů byly využity Highwire Press značky.

Tabulka 1 - Vytvořená metadata

Highwire Press značka	Obsah
citation_title	Titul článku, načtený z pohledu
citation_publication_date	rok publikace vydání, načteno z pohledu
citation_author	jména autorů dle požadavků Google Scholar bez titulů, načteno z pohledu
citation_journal_title	název časopisu, SLOVJANI.info
citation_issn	ISSN časopisu, 2570-7116
citation_volume	ročník vydání, načteno z pohledu
citation_issue	číslo vydání, načteno z pohledu
citation_pdf_url	odkaz na PDF článku nebo vydání, načteno z pohledu
citation_firstpage	První stránka, načteno z pohledu
citation_lastpage	Poslední stránka, načteno z pohledu

Zdroj: Vlastní zpracování, 2022

4.6.3 Tlačítka stáhnutí

Na dolní části podrobnosti článku jsou dostupná dvě tlačítka, jedno pro stáhnutí samotného PDF souboru článku a druhý pro PDF souboru vydání. Tlačítka pro stáhnutí odkazují na adresu vytvořenou spojením adresy archivu souborů a jména souboru z pohledu. Tlačítko pro vydání obsahuje stránky, na kterých lze tento článek najít. Jelikož ne všechny samostatné články mají pro sebe vytvořený PDF soubor, provádí se kontrola, jestli hodnota pro jméno souboru není prázdná. Pokud není prázdná, tlačítko odkáže na její lokaci v archivu. Pokud ale je prázdná, tlačítko je deaktivováno a upozorňuje že tento článek ještě není dostupný.

4.7 Robots.txt

Pro dosažení lepší optimalizace pro vyhledávače je přítomný soubor robots.txt. V tomto textovém souboru lze definovat pravidla pro roboty, jako například crawlery, ke kterým stránkám, složkám či souborům mají přístup a kterým ne. Tyto instrukce se hlavně používají pro vyhnutí přetížení stránky požadavky, nikoliv pro skrytí stránky před vyhledávači (23).

Obrázek 20 – Obsah souboru robots.txt

```
User-agent: *  
Allow: /content.php  
Allow: /article.php  
Disallow: /search.php  
Disallow: /languages/
```

```
Sitemap: https://slovjani.info/sitemap.php
```

Zdroj: Vlastní zpracování

Robots.txt funguje způsobem, že první se definuje pomocí „User-agent:“ pro který roboty pravidla platí. Lze zde napsat buď konkrétní roboty, jako „Googlebot“, což omezí přístup robotům Google, nebo „*“, což označuje až na výjimky všechny roboty. Následně se definují pravidla. Syntaxe pravidel rozpoznávaný Googlem jsou „allow:“, čímž povolujeme přístup ke souborům, nebo naopak „disallow:“, čímž zakážeme přístup. Taktéž lze v tomto souboru definovat lokaci souboru mapy stránky, a to pomocí „sitemap:“. Mezitím co v povolení a zakázání se uvádí lokální adresa, v mapě stránky musí být uvedena URL adresa (24). Jelikož stránka SLOVJANI.INFO už má existující soubor robots.txt, tyto pravidla byly do tohoto souboru doplněny.

4.8 Sitemap.php

Obrázek 21 – Ukázka obsahu sitemap.php

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>https://slovjani.info/content.php</loc>
  </url>
  <url>
    <loc>https://slovjani.info/article.php?article=1</loc>
  </url>
  ...
</urlset>
```

Zdroj: Vlastní zpracování

Funkce mapy stránky je umožnění lepšího crawlingu stránek, obzvláště když je jich vysoký počet. Tento soubor může být formátu XML, RSS nebo prostý text. Nejvhodnější je vytvoření pomocí XML, jelikož kromě adresy stránky lze zde uvést doplňitelné údaje, jako poslední datum úpravy stránky, jak často se mění či prioritu stránky v rámci domény (25).

Jelikož se jedná o mapu dynamické webové aplikace, pro tuto webovou aplikaci byla zvolena XML mapa vytvořená pomocí PHP. První je vytvořen objekt reprezentující XML dokument pomocí funkce „SimpleXMLElement“, ve kterém je definován XML formát s elementem „urlset“. Tento element má atribut „xmlns“ který uvádí URL adresu jmenného prostoru neboli „namespace“. Jelikož jeden název elementu může označovat víc věcí, jmenný prostor nám umožňuje definovat co tento element má reprezentovat (18 str. 269). Následně je do elementu „urlset“ nejdříve manuálně vložena pomocí „addChild“ URL adresa pro stránku článků, následně pomocí spojení URL adresy stránky článku a ID článků načtených přes připravený SQL příkaz v cyklu „while (\$row = \$result->fetch_assoc())“ vloženy všechny adresy článků.

Obrázek 22 – Ukázka definování formátu obsahu a vypsání XML objektu

```
Header('Content-type: text/xml');  
echo $xml->asXML();
```

Zdroj: Vlastní zpracování

Jamile „urlset“ obsahuje všechny stránky a jejich odkazy, předtím než je cokoliv vypsáno je nutné definovat, že bude zobrazen obsah formátu XML. Toho lze dosáhnout pomocí funkce „header“ obsahující atribut „Content-type“, kde definujeme že se jedná o XML (18 str. 189). Nakonec pomocí funkce „asXML“ je mapa vypsána ve formátu XML.

4.9 Přístupnost

Jedním z důležitých aspektů této webové aplikace je, aby byla přístupná osobám se zdravotním postižením. Podle směrnic WCAG 2.1 byla aplikace upravena a rozšířena funkcemi, aby dodržovala úroveň AAA přístupnosti. Samotná směrnice přístupnosti se dělí na 4 principy, který se zabývají každý svým aspektem vlastností webové stránky. Nebudou zde zmíněny všechny směrnice, ale jen hlavní sekce a ty, kvůli kterým byly provedeny změny kódu nebo přidány prvky.

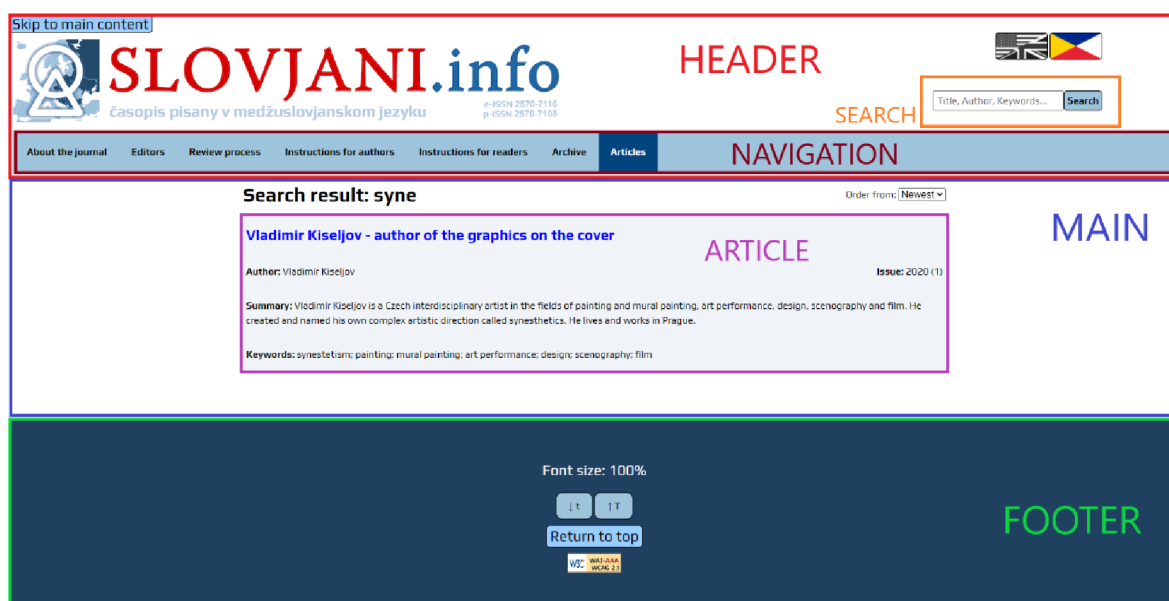
4.9.1 Vnímatelnost

První princip je vnímatelnost, tento princip se zabývá tím, že jsou informace a struktura stránky prezentována tak aby byl uživatel schopen porozumět obsahu. Sekce 1.1 směrnice se zabývá textovou alternativou. U obsahů, který jsou netextového formátu, jako například obrázky, je nutné vytvořit alternativní název, aby byl uživatel používající čtečku obrazovky schopen pochopit co má tento obsah zobrazovat (26). Příkladem je logo na vrcholu stránky, který obsahuje HTML atribut „alt“, který popisuje že jde o logo časopisu SLOVJANI.INFO. Další příkladem jsou tlačítka pro volbu jazyka. Jelikož bylo designově zvoleno že tlačítka budou obsahovat obrázek vlajek příslušného jazyka místo textu, je nutné těmto tlačítkům doplnit informaci pomocí „aria-label“. Aria-label je atribut do kterého můžeme napsat název nebo funkci elementu pro čtečky obrazovky v případech kdy text obsažený v elementu může nedostatečně vysvětlovat funkci elementu nebo nevysvětlovat

vůbec. Doplnitelně k aria-label, který by měl obsahovat krátký popis, lze tyto údaje doplnitelně rozšířit v atributu „aria-description“ (27).

Sekce 1.2 směrnice se zabývá video a audio obsahem stránek, který se však této webové aplikace netýká, jelikož takový obsah zde není (26).

Obrázek 23 – Ukázka sekcí stránky



Zdroj: Vlastní zpracování, 2022

Sekce 1.3 směrnice se zabývá přizpůsobitelností obsahu, aby bylo obsah možný zobrazit alternativními způsoby ale vždy byla udržena jeho smysluplnost a struktura. Směrnice 1.3.1 požaduje, aby byla stránka správně strukturována pomocí HTML sekcí (26). Pomocí těchto sekcí jsou všechny stránky této aplikace rozděleny na 3 hlavní části. První je hlavička „header“ která zahrnuje prvky které jsou dostupné na všech stránkách, jako je logo, vyhledávání, volba jazyku, a navigační lištu, která je sama obsažená doplnitelně v podsekcí „nav“. Hlavní obsah stránky je zahrnut do sekce „main“, kde doplnitelně na stránce článků a vyhledávání jsou zobrazené články zahrnutý každý do své podsekcce „article“. Poslední je sekce „footer“ která označuje zápatí stránky.

Směrnice 1.3.2 požaduje, aby byl pořadí obsahu smysluplné, čehož je dosaženo pomocí seřazení a označení obsahů pomocí sekcí. Směrnice 1.3.3 požaduje, aby bylo možné zobrazit stránku na výšku i na šířku bez ztráty konzistentnosti obsahu, což je velice důležité,

aby bylo možné webovou stránku zobrazit i na mobilních zařízeních (26). Webová aplikace je navrhována tak aby byla plně zobrazitelná i na mobilních zařízeních, a obsah je podle toho schopen se strukturou přizpůsobit.

Sekce 1.4 směrnice se zabývá přehledností obsahu, aby bylo možné ho bez obtíží vizuálně pochopit. Směrnice 1.4.1 požaduje, aby nebyly používány pouze barvy jako způsob pro předání informací (26). Jediný případ, kde k tomu dochází je u tlačítek pro volbu jazyka, kde je tlačítko momentálně zvoleného jazyka zašedlý. Aby byla tato směrnice dodržena, zašedlý tlačítko doplnitelně nevytváří rámeček, když je nad ním kurzor, a kurzor nemění svůj design, čímž je uživateli doplnitelně zobrazeno že tlačítko je deaktivované. Směrnice 1.4.3 a 1.4.6 požadují, aby byl udržen poměr 7:1 v kontrastu mezi textem a jeho pozadím (26). Kontrast byl testován pomocí nástroje WebAIM Contrast Checker, kde všechny kombinace použitých barev pozadí a textu vycházeli v poměru vyšším než 7:1.

Obrázek 24 - Ukázka HTML kódu a CSS tříd pro nastavitelnou velikost textu

HTML:

```
<body class="text--multiplier--<?php echo $fontsize;?>">
<h1 class="text--size--3 list__title">Header 1</h1>
```

CSS:

```
.text--multiplier--2{
    --multiplier: 1.25;
}
.text--size--3{
    font-size: calc(200% * var(--multiplier));
}
```

Zdroj: Vlastní zpracování, 2022

Směrnice 1.4.4 požaduje, aby bylo možné změnit velikost fontu až na 200% původní velikosti bez narušení struktury stránky (26). Velikost fontu je možné měnit ve většině prohlížečů, ale doplnitelně byla v této aplikaci vytvořena možnost zvětšit velikost fontu pomocí tlačítek dostupných ve zápatí stránky. Tyto tlačítka pomocí PHP mění CSS třídu „text--multiplier--[hodnota 1-4]“, která je nastavena v třídě těla HTML. Tato CSS třída obsahuje variabilní pro zvětšení textu ve skocích po 25 %. Tato variabilní je následně zahrnuta do tříd „font--size--1“ která označuje font velikosti 100 %, „font--size--2“ pro 150 % a „font--size--3“ pro 200 %, ve kterých se pomocí CSS funkce „calc“ provádí výpočet požadované velikosti textu. Velikost fontu se následně definuje u individuálních textů, například

nejdůležitější text stránky bude v elementu „<h1>“ a bude mít přiřazenou třídu pro největší velikost fontu „font-size-3“.

4.9.2 Použitelnost

Hlavním bodem tohoto principu je, aby byly všechny prvky designu a navigace použitelné. Z toho důvodu první sekce 2.1 této směrnice se zabývá tím, aby bylo možné projít celou stránku za pomoci klávesnice, aniž by mohlo kdekoliv dojít k chycení v „pasti“, odkud by uživatel nemohl postupovat klávesnicí dál po stránce a musel by z tohoto důvodu stránku znovu otevřít (26). Webová aplikace byla zkontrolována, aby všechny prvky byly dostupné přes klávesnici a nikde nebylo možné uvíznout. Sekce 2.2 směrnice požaduje, aby nebyl na stránce žádný časem omezený obsah, který nikde v této webové aplikaci není. Taktéž v aplikaci nikde není obsah či prvek, který by mohl blikat víc než 3krát za vteřinu, tím dodržuje pravidla směrnice 2.3 která se zabývá obsahem který by mohl zavinit epileptický záchvat (26).

Sekce 2.4 směrnice požaduje, aby byly uživateli dostupné prvky, který by mohli zjednodušit navigaci po webové stránce a udělat jí víc srozumitelnou. Prvním tímto prvkem je tlačítko pro přeskočení k hlavnímu obsahu jako první dostupný element při pohybu po stránce pomocí klávesnice, kvůli směrnici 2.4.1 která požaduje, aby bylo dostupné tlačítko pro přeskočení na všech stránkách opakující bloků obsahu (26). V této aplikaci toto tlačítko umožňuje přeskočit vrchol stránky a přesune uživatele hned na hlavní část stránky označený atributem „id=“maincontent““. Taktéž dostupné je podobné tlačítko v zápatí stránky, který naopak uživatele vrátí na vrchol stránky. Podle požadavků směrnice 2.4.2 obsahují všechny stránky aplikace v elementu „head“ titulek stránky v elementu „title“, který obsahuje kromě titulu na stránce článku číslo stránky, stránce vyhledávání požadovaný termín a stránce článku název zobrazeného článku. Směrnice 2.4.7 požaduje, aby bylo uživateli pohybující se po stránce pomocí klávesnice zvýrazněn prvek na kterém momentálně je (26). Toho je možné dosáhnout změnou vzhledu prvku pomocí dynamické pseudotřídy „:focus“, který umožňuje změnit design prvku který je momentálně objektem pozornosti klávesnice, podobně jako pseudotřída „:hover“, která umožňuje změnit design prvku když je nad ní kurzor myši. Tento prvek potom může mít jiný design nastavený v CSS (3 str. 248).

Sekce 2.5 směrnice se zabývá zjednodušením interakce s prvky stránky jinými vstupními zařízení než klávesnice, jako je myš a dotykový monitor. Nejdůležitější úpravou dle směrnice 2.5.5 bylo nastavení pomocí CSS minimální velikosti všech tlačítek v aplikaci na 44x44 pixelů. Jedinou výjimkou dle pravidel je tlačítko pro odeslání vyhledávání, jelikož jeho velikost je určena velikostí textu.

4.9.3 Pochopitelnost

Cílem tohoto principu je, aby byly informace a operace uživatelského rozhraní pochopitelné. Sekce 3.1 směrnice se zabývá čitelností a směrnice 3.1.1 a 3.1.2 požadují, aby byl definován jazyk stránky a elementu s odlišným jazykem než samotná stránka (26). Jazyk stránky se definuje pomocí atributu „lang“ ve elementu „<html>“, obsahuje standardizovaný ISO 639-1 jazykový kód. Zde dochází k unikátnímu problému. ISO kód pro anglický jazyk je „en“ (28), ale stránka je taktéž dostupná v mezislovanštině, moderní jazyk, který ještě nemá přiřazený ISO kód. Z tohoto důvodu bylo zvoleno dočasné označení „msl“, který nereprezentuje žádný jiný jazyk. Zobrazený obsah stránek odpovídá uživateli zvolenému jazyku, až na výjimku na stránce podrobnosti článku, kde jsou zobrazeny údaje v angličtině i mezislovanštině. Z tohoto důvodu údaje mají taktéž individuálně definovaný jazyk. Směrnice 3.1.3, 3.1.4 a 3.1.5 požadují, aby text stránky neobsahoval neobvykle termíny jako idiomy a žargony, a pokud se používají zkratky, aby byly dostupné u nich jejich význam (26). Samotná webová aplikace takové termíny ani zkratky nevyužívá, ale mohou se objevit ve zobrazených člancích.

Sekce 3.2 se zabývá předvídatelností, neboli že uživatel může očekávat kde se některý element objeví. Směrnice 3.2.3 požaduje, aby byly navigační prvky přes celou webovou aplikaci stejný pořadí (26). Jelikož toto není kompletně nezávislá aplikace, ale součástí domény SLOVJANI.INFO, vzhled navigační lišty je navržen tak aby byl identický existujícímu vzhledu stránky, ale kód je plně nový a originální, napsaný tak aby tato lišta dodržovali všechny požadavky očekávané standardem přístupnosti.

4.9.4 Robustnost

Cílem tohoto principu je, aby obsah byl dostatečně robustní, aby byl dostupný pro co nejvíce prohlížečů, zařízení a podporných technologií. Tento princip se skládá z jedné sekce o kompatibilitě. Nejdůležitější směrnice této sekce 4.1.1 která požaduje, aby byl kód aplikace validní (26). V rámci výsledků a diskuzí bude provedena validace všech kódu pomocí validačních nástrojů.

4.10 Responzivní design

Webová aplikace je designována, aby bylo možné ji zobrazit jak na počítačích, tak i na mobilních zařízeních bez jakékoliv ztráty obsahu. Prvním krokem, jak umožnit lepší vykreslování obsahu na mobilních zařízeních je pomocí meta značky „viewport“.

Obrázek 25 – Meta element pro umožnění responzivního zobrazení

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Zdroj: Vlastní zpracování, 2022

Tato meta značka umožňuje nastavit aby šířka zobrazení stránky odpovídala šířce zařízení pomocí vlastnosti „width=device-width“, a výchozí zvětšení stránky pomocí „initial-scale“, čímž se definuje že 1 Viewport pixel se rovná 1 CSS pixelu (29). Další klíčový způsob, kterým lze zlepšit zobrazení webové aplikace na mobilních zařízeních je použití CSS modulů „flex“ a „grid“ pro definování struktury stránky. Modul „flexbox“ umožňuje definovat způsob rozpoložení obsahu podél jedné osy, definovat jakým směrem je obsah poskládán, velikost prostoru mezi a způsob jakým jsou srovnány (4 str. 563). Modul „gridbox“ oproti tomu umožňuje definovat rozpoložení obsahu ve dvoudimenzionálním prostoru, pomocí definování prostoru vlastností „grid-template-areas“, který obsahuje mapu rozpoložení obsahu podle jmen určených vlastností „grid-area“ v třídě elementu (4 str. 655).

Pozice elementů sekcí obsažených v těle se provádí pomocí flexboxu. Tento flexbox má nastavenou vlastnost výšky na „100vh“, což určuje že bude využito 100% výšky zobrazovací plochy, a zarovnání „justify-content“ na „space-between“, který definuje že případný prostor pro zaplnění mezi elementy bude pouze mezi nimi, nikoliv na začátku a konci. Pomocí těchto

vlastností je zaručeno že pokud nebude mít stránka dostatek obsahu pro zaplnění obrazovky na celou výšku, přesto zůstane zápatí stránky na konci. Další příklad využití flexbox je u navigační lišty, u kterého jsou pomocí tohoto modulu zarovnány tlačítka na levou stranu stránky, a v případě že nebude dostatek prostoru je zobrazit v jedné řadě.

Obrázek 26 - Ukázka gridbox s uvedenými grid-area

Who and how is learning the Czech language today		<i>list_article_title</i>	
Author: Vit Benešovský	<i>list_article_author</i>	<i>list_article_edition</i>	Issue: 2020 (1)
Summary: The experience of a teacher the Czech language in Prague		<i>list_article_summary</i>	
Keywords: learning the Czech language; motivation to the learning Slavic languages		<i>list_article_keywords</i>	

Zdroj: Vlastní zpracování, 2022

Samotný vrchol stránky a podrobnosti článku jsou strukturovány za pomoci gridboxu. V něm obsažené prvky mají definovanou pozici v mapě „grid-template-areas“, kterou lze pomocí „media queries“ změnit podle velikosti obrazovky. Media queries umožňují definovat jaké CSS třídy mají být použity, pokud je obrazovka konkrétního typu nebo velikosti (4 str. 20). Aplikace má CSS tímto způsobem nastavenou mapu gridboxu a vlastnosti některých jiných tříd v případě, když je obrazovka na šířku menší než 1250 pixelů.

Navigační lišta je designována tak aby v případě, že obrazovka je na šířku menší než 1250 pixelů, pomocí kombinace CSS media queries a JavaScriptu se promění v „hamburger menu“, neboli vertikální menu, který uživatel může otevřít a zavřít. Díky tomu se vyhýbáme zbytečně dlouhému vrcholu stránky na vertikálních obrazovkách. Pomocí CSS je nastaveno, že pokud je obrazovka široka méně než 1250 pixelů, je směr lišty změněn z řádku na sloupec a tlačítka zneviditelněna pomocí doplňné CSS třídy obsahující „display:none“. Naopak zviditelněný je tlačítko pro menu, který po stisknutí provede JavaScript funkci, která odstraní CSS třídu, který tlačítka dělá neviditelný, a naopak při druhém stisknutí tuto třídu navrátí. Tato funkce taktéž mění atribut „aria-label“ aby tlačítko správně čtečkám obrazovky uvádělo, jestli tlačítko otevře menu nebo naopak zavře.

5 Výsledky a diskuse

Na závěr byla provedena validace kódu, kontrola přístupnosti a responzivního designu pomocí dostupných nástrojů.

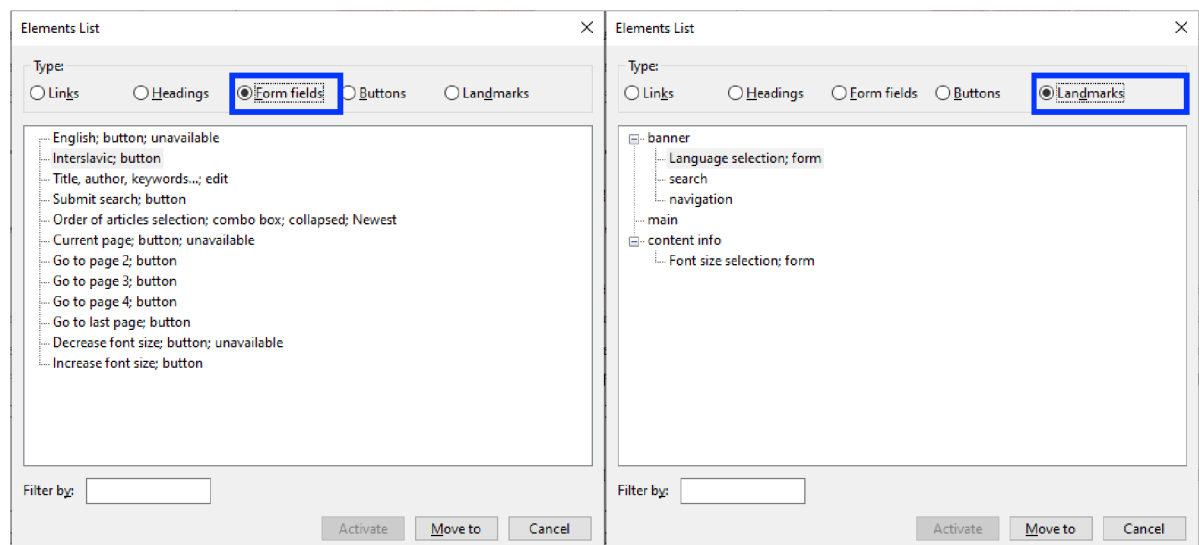
5.1 Validace

HTML kód byl validován pomocí W3C Markup Validation Service. Při validaci bylo nalezeno několik překlepů, který byly opraveny a výsledkem je HTML kód všech stránek validní. Stejným způsobem byl validován CSS kód pomocí W3C CSS Validation Service, kde nebyly nalezeny žádné chyby.

5.2 Kontrola přístupnosti

Kontrola přístupnosti byla provedena kombinací použití nástrojů Google Lighthouse a Axe DevTools a procházení stránky s čtečkou obrazovky NVDA. Jediné chyby, který byly nalezeny pomocí Google Lighthouse a Axe DevTools bylo že atribut „aria-description“, jehož funkcí je podrobněji vysvětlit roli prvku, byl označen jako neplatný ARIA atribut. Důvod k tomuto označení je že „aria-description“ je novým atributem, přítomný v ARIA 1.3 který je ještě ve vývoji a je ve fázi návrhu, mezitím co poslední přijatý ARIA 1.2 tento atribut ještě neobsahuje (27).

Obrázek 27 – Seznam vstupů a sekcí stránky v NVDA



Zdroj: Vlastní zpracování, 2022

Během procházení stránky s programem NVDA bylo zkontrolováno že všechny prvky mají pochopitelný označení. Tento program byl schopen přečíst informace v atributu „aria-description“. Čtečka je schopná přečíst obsah v angličtině, ale v současné době ještě neexistuje čtečka pro mezislovanštinu. Přesto je stránka schopná uvádět stejné informace v mezislovanštině, pro případ že tento jazyk bude v budoucnosti integrován do těchto programů.

5.3 Kontrola responzivního designu

Obrázek 28 – Zobrazení stránky v Google Chrome na Android zařízení



Zdroj: Vlastní zpracování, 2022

Kontrola responzivního designu byla provedena pomocí nástrojů v Google Chrome pro simulaci obrazovky mobilních zařízení. Při kontrole nebyla nalezena žádná ztráta obsahu nebo struktury. Doplnitěně byl design testován přímo na mobilním Android zařízení taktěž v prohlížeči Google Chrome, kde taktěž nebyl nalezen žádný problém.

6 Závěr

Výsledkem této bakalářské práce je funkční webová aplikace pro mezinárodní odborný časopis SLOVJANI.INFO, která vytváří metadata pro vyhledávač Google Scholar a dodržuje podmínky směrnice WCAG 2.1 na úrovni AAA, aby byla přístupná osobám se zdravotním postižením.

V teoretické části byly popsány technologie a programy použité při vývoji této webové aplikace, včetně jejich rolí při tvorbě této aplikace. V praktické části byla aplikace navržena pomocí IFML ve WebRatio, vytvořena lokální instance MySQL databáze časopisu, a podle modelu pomocí PHP, HTML a CSS vytvořena samotná aplikace. U funkcí stránky bylo popsáno, jakými metodami bylo možné je dosáhnout a jejich výhody a nevýhody, podle kterých byl odůvodněn a zvolen konkrétní způsob řešení. Zvolená metoda byla vysvětlena, jak funguje. Funkce vyhledávání byla ošetřena, aby byla zabezpečená proti bezpečnostním hrozbám SQL injection. Následně byla aplikace upravená a rozšířená funkcemi pro dosažení úrovně AAA směrnice WCAG 2.1. Byla provedena úprava aplikace, aby byl design responzivní a kompatibilní s mobilními zařízení, při čemž bylo využito JavaScript. Na závěr byl kód validován pomocí W3C nástrojů a přístupnost byla testována pomocí nástrojů Google Lighthouse, Axe DevTools a programu NVDA.

Webová aplikace je nasazená a dostupná na adrese „<https://slovjani.info/content.php>“.

7 Seznam použitých zdrojů

1. History of SQL. Oracle Corporation [Online]. [cit. 2022-3-3]. Dostupné z: https://docs.oracle.com/cd/B19306_01/server.102/b14200/intro001.htm.
2. TAHAGHOGHI, Seyed M.M. a Hugh E. WILLIAMS. *Learning MySQL*. Sebastopol, CA: O'Reilly, [2007]. ISBN 978-0-596-00864-2.
3. MUSCIANO, Chuck a Bill KENNEDY. *HTML & XHTML: The definitive guide*. Sixth edition. Sebastopol, CA: O'Reilly, [2007]. ISBN 978-0-596-52732-7.
4. MEYER, Eric A. a Estelle WEYEL. *CSS: The definitive guide : visual presentation for the web*. Fourth edition. Sebastopol, CA: O'Reilly, [2018]. ISBN 978-1-449-39319-9.
5. History of PHP. The PHP Group [Online]. [cit. 2022-3-2]. Dostupné z: <https://www.php.net/manual/en/history.php.php>.
6. Web Content Accessibility Guidelines (WCAG) 2.1. W3C [Online]. [cit. 2022-27-2]. Dostupné z: <https://www.w3.org/TR/WCAG21/>.
7. About Google Scholar. Google [Online]. [cit. 2022-3-4]. Dostupné z: <https://scholar.google.com/intl/cs/scholar/about.html>.
8. Crawl Guidelines. Google [Online]. [cit. 2022-3-4]. Dostupné z: <https://scholar.google.com/intl/cs/scholar/inclusion.html#crawl>.
9. Indexing Guidelines. Google [Online]. [cit. 2022-3-4]. Dostupné z: <https://scholar.google.com/intl/cs/scholar/inclusion.html#indexing>.
10. Scope, IFML: The Interaction Flow Modeling Language. Object Management Group, Inc. [Online]. [cit. 2022-3-3]. Dostupné z: <https://www.ifml.org/scope/>.
11. About - Apache Friends. Apache Friends [Online]. [cit. 2022-3-3]. Dostupné z: <https://www.apachefriends.org/about.html>.
12. HO, Don. Getting started | Notepad++ User Manual. Notepad++ [Online]. [cit. 2022-3-3]. Dostupné z: <https://npp-user-manual.org/docs/getting-started/>.
13. MySQL Workbench. Oracle Corporation [Online]. [cit. 2022-3-3]. Dostupné z: <https://www.mysql.com/products/workbench/>.
14. HOSCH, William L.. Chrome, Internet browser. Britannica [Online]. [cit. 2022-3-3]. Dostupné z: <https://www.britannica.com/technology/Chrome>.
15. VAILSHERY, Lionel Sujay. Web Browsers - Statistics & Facts. Statista [Online]. [cit. 2022-3-1]. Dostupné z: <https://www.statista.com/topics/5684/web-browsers/>.
16. NV Access | About NVDA. NV Access [Online]. [cit. 2022-3-3]. Dostupné z: <https://www.nvaccess.org/about-nvda/>.

17. Quick start / Methodology / BEM. Yandex [Online]. [cit. 2022-3-11]. Dostupné z: <https://en.bem.info/methodology/quick-start/>.
18. TATROE, Kevin, Peter MACINTYRE a Rasmus LERDORF. *Programming PHP*. Third edition. Sebastopol, CA: O'Reilly, [2013]. ISBN 978-1-449-39277-2.
19. KOCH, Richie. Cookies, the GDPR, and the ePrivacy Directive. Proton Technologies [Online]. [cit. 2022-2-12]. Dostupné z: <https://gdpr.eu/cookies/>.
20. PHP - File. The PHP Group [Online]. [cit. 2022-2-12]. Dostupné z: <https://www.php.net/manual/en/function.file.php>.
21. MySQL - LIMIT. Oracle Corporation [Online]. [cit. 2022-2-6]. Dostupné z: <https://dev.mysql.com/doc/refman/5.6/en/limit-optimization.html>.
22. PHP - htmlspecialchars. The PHP Group [Online]. [cit. 2022-2-19]. Dostupné z: <https://www.php.net/manual/en/function htmlspecialchars.php>.
23. Introduction to robots.txt. Google [Online]. [cit. 2022-3-4]. Dostupné z: <https://developers.google.com/search/docs/advanced/robots/intro>.
24. Create a robots.txt file. Google [Online]. [cit. 2022-3-4]. Dostupné z: <https://developers.google.com/search/docs/advanced/robots/create-robots-txt>.
25. Sitemaps XML format [Online]. [cit. 2022-3-4]. Dostupné z: <https://www.sitemaps.org/protocol.html>.
26. How to Meet WCAG (Quick Reference). W3C [Online]. [cit. 2022-2-27]. Dostupné z: <https://www.w3.org/WAI/WCAG21/quickref/>.
27. Accessible Rich Internet Applications (WAI-ARIA) 1.3. W3C [Online]. [cit. 2022-2-27]. Dostupné z: <https://w3c.github.io/aria/>.
28. Codes for the Representation of Names of Languages. Library of Congress [Online]. [cit. 2022-2-27]. Dostupné z: https://www.loc.gov/standards/iso639-2/php/English_list.php.
29. Using the viewport meta tag to control layout on mobile browsers. Mozilla Corporation [Online]. [cit. 2022-2-28]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTML/Viewport_meta_tag.