

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky

Webová aplikace na hodnocení uživatelů a kanálů portálu YouTube

Ondřej Matějka

Bakalářská práce  
2022

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2021/2022

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Ondřej Matějka**  
Osobní číslo: **I19117**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Téma práce: **Webová aplikace na hodnocení uživatelů a kanálů portálu YouTube**  
Zadávající katedra: **Katedra informačních technologií**

## Zásady pro vypracování

Práce se zabývá analýzou a realizací webové aplikace na hodnocení uživatelů a streamovaných kanálů videoportálu YouTube. Mezi hodnotící kritéria patří například vhodnost pro děti a mladistvé, a proto aplikace může posloužit i jako souhrn informací rodičům, co jejich děti sledují. Aplikace taktéž využívá originální YouTube Data API (v3) a vytváří automaticky obecný přehled o lidech s širokým vlivem na zmíněném portále. Ucelenou recenzi ale tvoří hlavně samotní uživatelé aplikace pomocí hodnotící číselné stupnice u jednotlivých kategorií a možností přidávání komentářů k tvůrcům videí. Technologie pro realizaci zahrnují HTML 5, CSS3, Javascript ES6, jQuery, PHP 8/Laravel, SQL s MySQL databází. Zmíněna je historie portálu, jeho dnešní specifikace, použité technologie a jejich implementace.

Rozsah pracovní zprávy: **min. 30 stran**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

LAURENČÍK, Marek. Tvorba www stránek v HTML a CSS. Praha: Grada Publishing, 2019. Průvodce (Grada). ISBN 978-80-271-2241-7.  
HOPKINS, Callum. PHP okamžitě. Brno: Computer Press, 2014. ISBN 978-80-251-4196-0.  
LACKO, Ľuboslav. SQL: hotová řešení. Brno: Computer Press, 2003. K okamžitému použití (Computer Press). ISBN 80-7226-975-5.

Vedoucí bakalářské práce: **Ing. Jan Panuš, Ph.D.**  
Katedra informačních technologií

Datum zadání bakalářské práce: **17. prosince 2021**  
Termín odevzdání bakalářské práce: **13. května 2022**

**Ing. Zdeněk Němec, Ph.D.** v.r.  
děkan

L.S.

**Ing. Jan Panuš, Ph.D.** v.r.  
vedoucí katedry

V Pardubicích dne 28. února 2022

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 17. 1. 2022

Ondřej Matějka

## **PODĚKOVÁNÍ**

Nejvíce chci poděkovat vedoucímu bakalářské práce panu Ing. Janu Panušovi, Ph. D. za vlídnost a ochotu, dále pak své rodině, která mi poskytla zázemí a podporu pro tvorbu práce a celkově možnost studovat Univerzitu Pardubice.

## **ANOTACE**

Práce se zabývá analýzou a realizací webové aplikace na detailní hodnocení uživatelů a kanálů video-portálu YouTube. Mezi hodnotící kritéria patří například vhodnost pro děti a mladistvé, a proto aplikace může posloužit i jako souhrn informací rodičům, co jejich děti sledují. Aplikace taktéž využívá originální YouTube Data API (v3) a vytváří automaticky obecný přehled o lidech s širokým vlivem na zmíněném portále. Ucelenou recenzi ale tvoří hlavně samotní uživatelé aplikace pomocí hodnotící číselné stupnice u jednotlivých kategorií a možností přidávání komentářů k tvůrcům videí. Technologie pro realizaci zahrnují HTML 5, CSS3, Javascript ES6, jQuery, PHP 8/Laravel, SQL s MySQL databází. Zmíněna je historie portálu, jeho dnešní specifikace, použité technologie a jejich implementace.

## **KLÍČOVÁ SLOVA**

webová aplikace, YouTube, chytré hodnocení, žebříček influencerů, webový server Apache, HTML, CSS, JavaScript, jQuery, YouTube Data API (v3), MySQL, SQL, php, Laravel, Ubuntu server

## **TITLE**

Webová aplikace na hodnocení uživatelů a kanálů portálu YouTube.

## **ANNOTATION**

The bachelor thesis deals with the analysis and implementation of a web application for detailed evaluation of users and channels of the YouTube video portal. The evaluation criteria includes, for example, suitability for children and adolescents, so the application can also serve as a summary of information for parents to know, what their children are watching. The application also uses the original YouTube Data API (v3) and automatically creates a general overview of people with a wide influence on the portal. However, a comprehensive review is mainly made up by the users of the application themselves by evaluating the rating numeric scale for individual categories and the possibility of adding comments to video creators. Implementation technologies include HTML 5, CSS3, Javascript ES6, jQuery, PHP 8 / Laravel, SQL with MySQL database. The history of the portal, its current specifications, the technologies used and their implementation are mentioned.

## **KEYWORDS**

web application, YouTube, smart ratings, influencers ranking, web server Apache, HTML, CSS, JavaScript, jQuery, YouTube Data API (v3), MySQL, SQL, php, Laravel, Ubuntu server

# OBSAH

Seznam obrázků .....	9
Seznam tabulek .....	10
Seznam zkratk .....	11
Úvod.....	12
1 Internetový server YouTube.....	13
1.1 Historie .....	14
1.2 Statistiky .....	16
2 Konkurenční nástroje .....	17
2.1 Social Blade .....	17
2.2 Speakrj.....	18
3 Analýza sentimentu .....	18
4 Klasifikace obsahu elektronických medií Kijkwijzer .....	19
5 Použité technologie .....	21
5.1 Protokol http .....	21
5.2 PHP.....	25
5.3 Framework Laravel.....	28
5.4 Jazyk HTML a kaskádové styly .....	30
5.5 MySQL databáze a jazyk SQL .....	32
5.6 JavaScript.....	34
6 Návrh aplikace.....	34
6.1 Požadavky.....	34
6.1.1 Funkční požadavky .....	34
6.1.2 Nefunkční požadavky.....	36
6.2 Databázový návrh .....	37
6.2.1 Entitně-vztahový diagram (ERD).....	37
6.2.2 ERD věty .....	38
6.2.3 Relační model.....	39
6.2.4 Logování změn profilů.....	40
6.3 Příklad užití.....	41
6.4 Specifikace případů užití a uživatelské rozhraní .....	42
6.4.1 Příklad užití UC1 .....	42
6.4.2 Příklad užití UC2.....	43
6.4.3 Příklad užití UC3.....	43
6.4.4 Příklad užití UC4.....	44
6.4.5 Příklad užití UC5.....	45
6.4.6 Příklad užití UC6.....	46
6.4.7 Příklad užití UC7.....	46
6.4.8 Příklad užití UC8.....	46
6.4.9 Příklad užití UC9.....	48
6.4.10 Příklad užití UC10.....	48

6.4.11	Případ užití UC11 .....	49
6.4.12	Případ užití UC12 .....	50
6.4.13	Případ užití UC13 .....	50
6.5	Vývoj aplikace, nástroje a služby .....	51
6.6	Nasazení aplikace a instalace nezbytných součástí .....	52
6.7	Testování a optimalizace aplikace .....	56
Závěr	.....	57
Použitá literatura	.....	59
Přílohy	.....	62



## SEZNAM OBRÁZKŮ

Obrázek 1: Oficiální podoba portálu YouTube vydaného v roce 2005 [4] .....	14
Obrázek 2: Ukázka použití analýzy sentimentu nástroje Neticle [9].....	19
Obrázek 3: Značky systému Kijkwijzer [11] .....	20
Obrázek 4: Diagram životního cyklu spojení HTTP/1.0 a HTTP/1.1 [12].....	21
Obrázek 5: Formát paketu a segmentu TCP [13] .....	22
Obrázek 6: Struktura HTTP požadavku a odpovědi, zdroj – vlastní .....	23
Obrázek 7: Zpracování skriptu PHP [20] .....	27
Obrázek 8: Zpracování a cachování kódu PHP [21].....	28
Obrázek 9: Ukázka základní struktury HTML, zdroj – vlastní .....	31
Obrázek 10: ERD model aplikace, zdroj – vlastní.....	38
Obrázek 11: Relační model aplikace, zdroj – vlastní .....	39
Obrázek 12: Případy užití webové aplikace, zdroj – vlastní.....	41
Obrázek 13: Návrh žebříčku, zdroj – vlastní .....	42
Obrázek 14: Návrh přihlášení a registrace, zdroj – vlastní.....	43
Obrázek 15: Návrh profilu, zdroj – vlastní .....	45
Obrázek 16: Návrh hodnocení, zdroj – vlastní .....	47
Obrázek 17: Návrh tvorby profilu, zdroj – vlastní.....	49
Obrázek 18: Návrh vyhledávání, zdroj – vlastní .....	51

## SEZNAM TABULEK

Tabulka 1: Přehled nejzajímavějších statistik YouTube [1] .....	17
Tabulka 2: Směrovací tabulka CRUD Laravel [23] .....	29

## SEZNAM ZKRATEK

UI	User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
480p	480 pixelů ve vertikálním rozlišení s progresivním skenováním
1080p	1080 pixelů ve vertikálním rozlišení s progresivním skenováním
CSS	Cascading Style Sheets
JS	JavaScript
DOM	Document Object Model
APT	Advanced Packaging Tool
MVC	Model-View-Controller
API	Application Programming Interface
ORM	Object-Relational Mapper
CRUD	Create, Read, Update, Delete

## ÚVOD

Internet se stal naším denním chlebem. Pracujeme, vzděláváme se, komunikujeme nebo se jen bavíme přes tuto globální síť. Člověk se v takovém neomezeném virtuálním prostoru může relativně rychle ztratit. Platforma YouTube patří mezi přední servery pro nahrávání, streamování a sledování videí. Mezi mladými lidmi bychom těžce hledali takové jedince, kteří zmíněný název neznají, nebo ho dokonce pravidelně nepoužívají. Pokud odevzdáme svůj čas konkrétním tvůrcům a vydavatelům videí na YouTube třeba prostřednictvím funkce „Odebírat“, měli bychom vědět, jestli je to moudrá volba. Čas je drahocenná věc a vždy je dobré zhodnotit, jak s ním nakládáme nebo jak sním nakládají ostatní, či naše děti, a jestli je takový obsah vůbec vhodný pro dotyčné oči.

YouTube se rozhodl odstranit negativní hodnocení ze stránek videí ve formě palce dolů. Na to rychle zareagovala komunita a vytvořila plugin pro prohlížeče „Return YouTube Dislike“, který hrubě nahrazuje funkcionalitu poměru mezi kladným a záporným hodnocením. Tvůrci videí navíc mohou sami ovlivňovat sekci komentářů. Autor přišel proto s nápadem na vytvoření aplikace, která by poskytovala obecný přehled o sledovaných kanálech na video-platformě, spočítala statistiky z dostupných dat, vyhodnotila a umožnila poskytnout sdílení názorů druhých lidí. Prostředí by mělo být co nejjednodušeji dostupné, proto byla zvolena možnost webových technologií.

Úkolem je automaticky vytvořit veřejnému video-kanálu profil, zajistit průběžnou aktualizaci dat a zobrazovat zpracované informace. Zajímavé budou zejména průměrné zhlédnutí videí, jejich hodnocení na straně serveru YouTube, počet odběratelů kanálu a uživatelské komentáře pod příspěvky. Hodnotit půjde i ze strany aplikace pomocí komentářů a stupnice, a to pouze jednou pro uživatele, který se musí autentizovat emailovou adresou a heslem. Své názory však může měnit. Hodnotící kritéria budou reprezentována stupnicí od jedné do pěti.

Autorovým cílem je aplikaci vytvořit a otestovat na vzorku českých tvůrců v rámci praktické části. V teoretické úrovni se zaměří nejdříve na samotný portál YouTube, uvede historii a statistická data využití služby na internetu. Zajímavé budou taktéž podobné nástroje dostupné online pro sledování souhrnných dat kanálů, do kterých se podrobně nahlédne. Následně se určí funkční a nefunkční požadavky pro vývoj aplikace. Vymezí a přiblíží se použité protokoly (HTTP/2, SSL/TLS), programovací jazyky (PHP 8, JavaScript), jazyky webových stránek (HTML a kaskádové styly), datová vrstva (MySQL, SQL) a nadstavbové frameworky (Laravel 8.x). Autor vytvoří databázový model, navrhne uživatelské rozhraní, určí se způsob

vyhodnocení dat, nastíní se, jak bude vypadat realizace a lokální vývoj aplikace, její následné nasazování na hostující server s veřejnou doménou. Aplikaci nakonec autor odladí a otestuje.

## 1 INTERNETOVÝ SERVER YOUTUBE

Služba YouTube umožňuje jakémukoliv uživateli nahrávat videa na jejich server. Média lze následně upravit, zveřejnit, sdílet, komentovat nebo třeba hodnotit palcem nahoru. Jejím aktuálním majitelem a správcem k roku 2022 je americký gigant Google Inc. Uvádí se, že je to druhá nejnavštěvovanější stránka světa hned po nejrozšířenějším webovém vyhledávači Google. Ač je využívání služeb zdarma, Google inkasuje nemalé výdělky z reklam vkládané do přehrávání videí. [1]

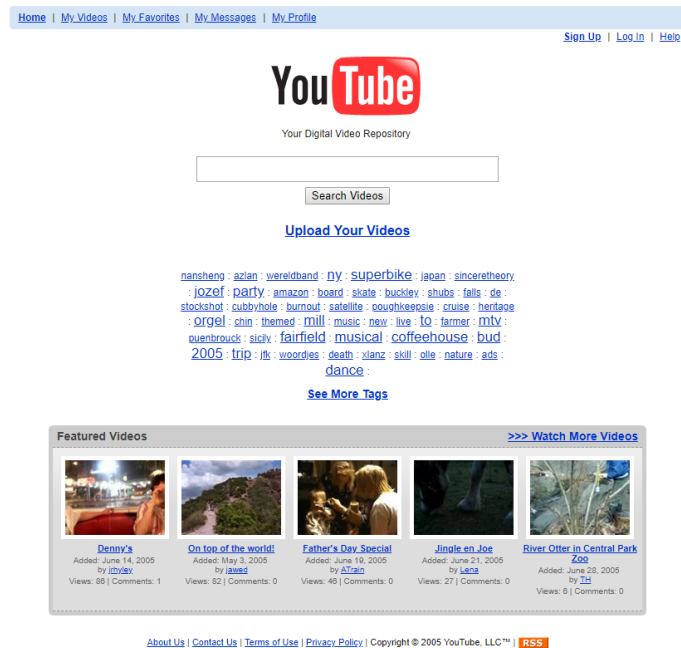
YouTube však není jen o legračních videích. Uživatelé po celém světě vytváří různé druhy obsahu. Server rozlišuje nahraná díla do mnoha kategorií: Film & Animation, Autos & Vehicles, Music, Pets & Animals, Sports, Short Movies, Travel & Events, Gaming, Videoblogging, People & Blogs, Comedy, Entertainment, News & Politics, Howto & Style, Education, Science & Technology, Nonprofits & Activism, Movies, Anime/Animation, Action/Adventure, Classics, Comedy, Documentary, Drama, Family, Foreign, Horror, Sci-Fi/Fantasy, Thriller, Shorts, Shows, Trailers. [2]

Služba se neustále vyvíjí, a tak nezůstává jen u stálých funkcí, ale rozšiřuje je. YouTube Music poskytuje prostředí pro poslech různorodé hudby, podporuje vytváření playlistů a jejich doporučování. Konkuruje tak celosvětově známé službě Spotify. YouTube Music spatřilo poprvé světlo světa v listopadu roku 2015. Největší transformaci prodělal v roce 2018, kdy přibyla webová platforma, umělá inteligence, off-line stahování a poslech na pozadí. YouTube Music pomalu vytlačí paralelně běžící sesterskou službu Google Play Music. [3]

S hudební platformu se také na trh dostavil YouTube Premium jako měsíční předplatné služeb YouTube. Ten zahrnuje YouTube Music umožňující pouštění hudby a videí na pozadí na přenosných zařízeních. Obsah lze taktéž předběžně stáhnout a přehrávat bez připojení k internetu.

## 1.1 Historie

Vznik jména YouTube nebyl nijak složitý. Jak je zřejmé, jedná se o akronym dvou slov. „You“ vyjadřuje způsob tvorby videí, tedy autoři tvorby jsou samotní uživatelé, nikoliv server samotný, a „Tube“ je starší výraz pro televizi. Při registraci domény se YouTube brzy stal předmětem žaloby společnosti, která vlastnila web [www.utube.com](http://www.utube.com), nicméně společnost s žalobou nepochodila.



Obrázek 1: Oficiální podoba portálu YouTube vydaného v roce 2005 [4]

Služba YouTube byla založena v roce 2005 pracovníky společnosti PayPal Chadem Hurley, Stevem Chenem a Jawedem Karimem. Startup obdržel na začátku investici ve výši 11,5 milionů dolarů od společnosti Sequoia Capital. První video, které se na portálu objevilo, neslo název „Já v zoo“ od zakladatele Karima. V prosinci po testovací fázi byl web oficiálně spuštěn a prvním videem, které získalo více než milion zhlédnutí, byla reklama od společnosti Nike. Prvním záměrem byly typy videí prezentující uživatele sami sebe. Brzy se přešlo na nápad obecnějších médií a server se raketově začal rozrůstat.

Rok 2006 přináší další novinky. Mezi nimi jsou skupiny (možnost seskupování videí). Uživatelé si také mohou přizpůsobovat svůj profil. YouTube omezil obsah na pouhých deset minut. Portál přichází s možností role „Directors“ pro tvůrce vlastního obsahu zdarma v reakci

na desetiminutový limit. Na videa lze reagovat a nahrávací funkce jsou dostupné o pro mobilní telefony. Již v této době Google kupuje YouTube za 1,65 miliardy amerických dolarů. Platforma měla zhruba 65 zaměstnanců.

V roce 2007 přichází pro tvůrce partnerský program umožňující vydělávat na nahraných videích. Přidává se podpora pro vícejazyčnost. Objevují se reklamní bannery přímo ve videích ve spodní části přehrávače s poloprůhledným motivem.

YouTube přidává podporu videí s rozlišením 480p v roce 2008. Objevují se i nástroje pro analýzu videí. Funkce Audioswap umožňuje autorům přidat licencovanou hudbu do tvorby.

V roce 2009 se na platformu přidává americký Kongres (občané mají poprvé možnost vidět dění v Kongresu) a Vatikán. Cena Peabody Award byla získána za vynikající výsledky v internetových médiích. YouTube zavádí službu hudebních videí VEVO. Ke konci roku si taktéž uživatelé užívají videa již v rozlišení 1080p a server umí rozpoznávat automaticky řeč.

Možnost půjčit si celovečerní film přišla v roce 2010. Hodnocení palcem se objevilo a nahradilo starý systém hvězd. Video se již mohou objevit na serveru v rozlišení 4k. YouTube vytvořil službu LIVE poskytující uživatelům funkci streamování živého obsahu.

YouTube Analytics přináší pokročile a chytré nástroje pro sledování výkonu kanálů s rokem 2011. Zjistit lze, kdo sleduje nahraná videa, jak dlouho a popularitu pro strategii zpeněžení.

YouTube se sloučil s Google Video a je na něm živě vysílána prezidentská debata a olympiáda. Na platformě se také v tento rok (2012) objevilo dodnes známé video „Gangnam Style“, které ukázalo, co je to pojem virální s miliardou zhlédnutí.

Rok 2013 ukázal, že se na serveru objeví miliarda unikátních návštěvníků měsíčně.

YouTube zakládá službu Red (budoucí Premium), která umožňuje odstranění reklam z platformy v 2015. Zrodila se hudební platforma YouTube Music společně se službou Gaming. Následující rok se objevuje obecná služba předplatného.

V roce 2019 lze přidat do videa dvě reklamy a v dalším roce se pomalu odebírá možnost kategorií. [3]

Rok 2021 odebral možnost negativního hodnocení medií a přidává funkci Shorts pro krátké příběhy jako mají konkurenční sociální služby (např. TikTok).

## 1.2 Statistiky

YouTube je druhá nejnavštěvovanější stránka světa. Zdaleka nejvíce uživatelů pochází z Indie (467 milionů), následují Spojené státy americké a Indonésie. 54 % uživatelů jsou muži. Vztaheno k USA, 95 % obyvatel v rozmezí 18-29 let používá YouTube. Skupina nad nimi do 49 let má pouze o čtyři procenta méně. Zhruba polovina starých lidí (65 let a více) je také aktivních na platformě. Na serveru zužitkují uživatelé 25 % z celkového síťového mobilního provozu. Aktivních uživatelů je aktuálně okolo 2,6 miliardy. Facebook jich stále eviduje víc zhruba o 300 milionů. Více než 70 % uživatelů je ovlivněno predikčním algoritmem na nabízení okruhu zájmů, na které se pak lidé koukají. 20 % uživatelů zastavuje video, pokud je obsah nezaujal do 10 vteřin. Nejsledovanějším videem se stalo video „Baby Shark“ od Pinkfong s 10,3 miliardy zhlédnutí. „T-Series“ je nejodebíranější kanál na platformě s 210 miliony odběrateli. 37 milionů kanálů je registrováno na serveru. YouTube je dostupný v 80 různých jazycích. Pětina času strávené na službě pochází z mobilních zařízení. Samotná služba vygenerovala roční zisk pro rok 2021 28,8 miliard amerických dolarů se zhruba 30% nárůstem. Jimmy Donaldson (Mr. Beast) vydělal v roce 2021 nejvíce peněz na platformě s odhadem 54 milionů dolarů. Vlivní tvůrci s pěti tisíci odběrateli inkasují průměrně za video 315 USD, pokud počet sledujících přesáhne 500 tisíc, může se jednat o částku necelých čtyř tisíc dolarů. S více než 122 miliony aktivními uživateli denně se zkoukne více než miliarda hodin videí. Průměrný strávený čas vychází na 18 minut na uživatele. [1]



Tabulka 1: Přehled nejzajímavějších statistik YouTube [1]

Celkový počet aktivních uživatelů za měsíc	2,6 miliard
Celkový počet denně aktivní uživatelů	Více než 122 milionů
Průměrný čas strávený denně na YouTube	18 minut
Počet zhlédnutých videí za den	Více než jedna miliarda
Počet hodin videí nahraných za jednu minutu	500

## 2 KONKURENČNÍ NÁSTROJE

### 2.1 Social Blade

Social Blade byl založen v roce 2008. Služba samotná umožňuje agregovat data ze sociálních sítí. Podporované sociální sítě jsou: YouTube, Twitch, Facebook, Instagram, Twitter, TikTok, Dailymotion, Odysee, Trovo, Mixer, DLive, StoryFire. Data portál přetaví na statistické tabulky a grafy. Historie sledovaných profilů osobností sahá až 3 roky nazpět. Server poskytuje čtyři možnosti předplatného a prémiových služeb. Služba umožňuje také veškerá zpracovaná data poskytovat dále pomocí vlastního REST API.

Populární na platformě je funkce pro živé sledování některých ukazatelů v reálném čase jako například čítač počtu odběratelů na serveru YouTube. Prohlédnout si lze také různé srovnávací žebříčky tvůrců dle země, kategorií obsahu tvorby nebo v rámci časového úseku. Nejsledovanějšími aspekty jsou sledovanost, odběratelé a odhadované zisky. Informace si lze přehledně v tabulce a grafu s jejich časovým vývojem změn. Tvůrce je možné porovnat.

Služba disponuje nejen webovým rozhraním, ale i aplikacemi pro mobilní zařízení, které lze získat v obchodech Google Play a App Store, K dispozici jsou také rozšíření pro webové prohlížeče Firefox a Google Chrome, která umožňují zobrazit rychle a přehledně informace o uživateli na zobrazeném profilu konkrétní sociální sítě. Za zmínku stojí i poskytovaný nástroj finanční kalkulačky pro výpočet odhadovaných zisků na platformě YouTube. [5]

## 2.2 Speakrj

Speakrj má obdobné ambice jako Social Blade, ale jejich funkce nejsou tak pokročilé. Server podporuje jen 4 sociální sítě: Instagram, Twitter, YouTube, Facebook a zobrazuje jen základní údaje a přehled o profilech tvůrců.

Službu vede digitální poradce marketingového světa Rajat Jain. Vytváří nástroje a nabízí konzultace pro moderní marketing založený například na automatických procesech a procesech podpořených vědeckými metodami a důkazy. [6]

## 3 ANALÝZA SENTIMENTU

Analýza sentimentu je technika pro získání emočního rozpoložení a postoje autora přispívajícího v nejčastěji textové podobě. Díky strojovému učení dnešní doby umožňuje relativně spolehlivě analyzovat názory uživatelů zejména v marketingu k určitým službám a produktům. Analýza sentimentu tak vyhodnotí velké množství neuspořádaných dat levně a efektivně. Výsledky umožňují odhalit a reagovat na vznikající problém se zanedbatelným zpožděním. Jelikož jsou data vytvářena lidmi, kteří mohou reagovat iracionálně, musí se k analýze přistupovat opatrně a s odstupem.

Způsoby použití analýzy jsou takřka omezené jen naší představivostí. Analyzovaný text můžeme klasifikovat obecně jako pozitivní, neutrální a negativní. Nic nám ale nebrání rozšířit stupnici na více úrovní pro přesnější specifikaci hodnocení. Více zajímavější může být informace získaná z textu o emoci, jako je například vztek, spokojenost nebo smutek. Analyzovat lze také zájem zákazníka nebo jeho naléhavost.

Nejčastějším způsobem, jak se analýza realizovala, bylo použití lexikonů, slovníků, které umožňují ohodnocovat číselnou známkou dílčí slova a fráze v textu. Problémů bohužel nastává hned několik. Na Zemi existuje mnoho kultur mluvících odlišným jazykem. Celkový počet vyskytujících se jazyků se odhaduje na tisíce [7]. Implementace vícejazyčného analyzátoru je tedy vysoce náročná. Jazykový klasifikátor musí nejdřív určit, o jaký jazyk se jedná, a následně se přepnout do potřebného vyhodnocovacího jádra. Dalším problémem může být expresivita, kreativita a významný výskyt proměnlivého slangu v jazyce. [8]

Obrázek 2: Ukázka použití analýzy sentimentu nástroje Neticle [9]

Obrázek 2 názorně ukazuje příklad analýzy komentáře pod hudebním videem. Správně odvodil pozitivitu textu, v kterém našel 3 slovníkové shody. Analyzátor zachytil i dvě téma, kterého se text týká. Nicméně se zde ukazuje i omezenost a problematika nástroje. Ač se v komentáři nenachází významově nic negativního, nástroj označil frázi se slovem „strašně“ jako zápornou a odečetl 2 body od výsledné známky i přes to, že v tomto kontextu se jedná o pozitivní vyjádření, jak moc chytlavá je to písnička. Analyzátor vrací prostřednictvím API odpověď ve formátu JSON, kde můžeme najít jak souhrn hodnocení, tak konkrétní data o nalezených shodách a jejich vyhodnocení.

## 4 KLASIFIKACE OBSAHU ELEKTRONICKÝCH MEDIÍ KIJKWIJZER

Dnešní svět médií přímo žije. Audiovizuální obsah se na nás hrne ze všech stran. Může se jednat o filmy, seriály, videa, počítačové a konzolové hry. Mnoho těchto vlivů může negativně působit zvláště na děti. Samozřejmě každý případ je ojedinělý v rámci individuální povahy dítěte a jeho věku. Důležitá je v tomto období role rodičů, kteří nastolují vhodnou výchovu. Jako kandidát

pro nástroj ulehčující volbu sledování audiovizuálního obsahu může být nizozemský Kijkwijzer.

Kijkwijzer je klasifikační systém mediálních dokumentů. Slovo vychází ze spojení slova „kijk“, které znamená dívat se, a slova „kajkwajzr“, překladem rádce. Nizozemsko tak zasadilo semínko v roce 2001 pro jednotný hodnotící systém výroby audiovizuálního obsahu. Systém se uchytil pro televizní stanice, videohry, kina a DVD distribuci. Kijkwijzer informuje zodpovědné osoby, zda je příslušné dílo potenciálně škodlivé pro mladší účastníky. K tomu využívá symboly pro věkové kategorie a značky označující důvod. [10]



Obrázek 3: Značky systému Kijkwijzer [11]

Symboly věkového označení se rozdělují do sedmi kategorií:

- AL – vhodné pro všechny,
- 6 – škodlivé pro děti do šesti let,
- 9 – škodlivé pro děti do devíti let,
- 12 – škodlivé pro děti do dvanácti let,
- 16 – škodlivé pro děti do šestnácti let,
- 18 – škodlivé pro děti do osmnácti let.

Značky pro opodstatnění věkového limitu se označují jako:

- Násilí – označené pěstí,
- Strach – označené pavoukem,
- Sex – označené chodidly nad sebou,
- Hrubý jazykový projev – označené křikem,
- Zneužívání drog a alkoholu – označené jehlou,
- Diskriminace – označené panáčkem před kontrastním davem.

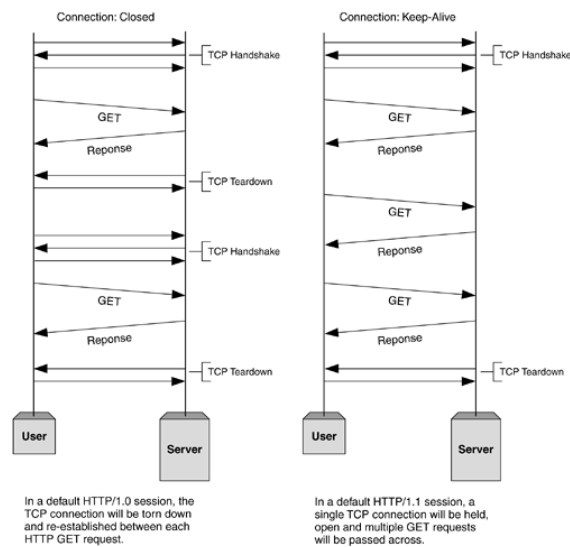
Oficiální webové stránky [kijkwijzer.nl](http://kijkwijzer.nl) již poskytují rozsáhlou databázi filmů, her, televizi, DVD společně s klasifikací obsahu pomocí svého systému nebo systému PEGI. Titul lze zde jednoduše nalézt společně s vysvětlením, proč je jeho obsah příslušně klasifikován. Kijkwijzer využívá vlastního dotazníku, který se distribuuje relevantním divákům. Odpovědi se pak dle algoritmu vyhodnotí a dílo se označí.

## 5 POUŽITÉ TECHNOLOGIE

### 5.1 Protokol http

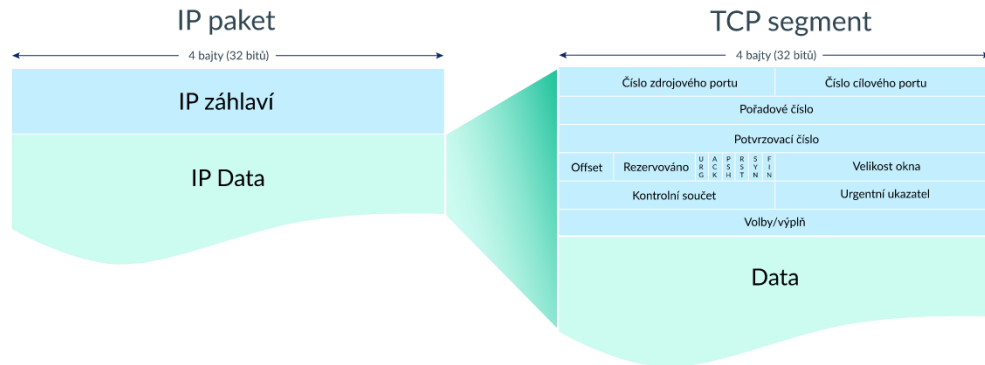
#### 5.1.1 O protokolu http

Zkratka internetového protokolu http vychází z anglického výrazu „Hypertext Transfer Protocol“ a umožňuje komunikaci mezi webovým serverem a klientem. V dnešní době se hojně využívá pro stahování hypertextových dokumentů reprezentující webové stránky. Lze tedy říct, že se jedná o jeden z nejrozšířenějších a nejpobulárnějších protokolů jednadvacátého století. Oblíbenost získal s jeho jednoduchostí a nenáročností. A i přes tyto aspekty dokáže poskytnout komplexní nástroj pro bohatou komunikaci napříč celým internetem. Kromě hypertextových dokumentů umí vyžádat od serveru v podstatě jakýkoliv soubor či data.



Obrázek 4: Diagram životního cyklu spojení HTTP/1.0 a HTTP/1.1 [12]

Protokol TCP transportní vrstvy umožňuje spolehlivé spojení stran. Protokol odstraňuje nedostatky přenosu mechanismy pro zamezení ztráty paketů, špatného pořadí, duplicity nebo poškození.



Obrázek 5: Formát paketu a segmentu TCP [13]

Obrázek 5 ukazuje veškerá data hlavičky, která jsou přenášena. První zařízení posílá paket s daty a druhé zařízení vždy přenos potvrzuje zvýšením potvrzujícího čísla o délku obdržených dat. Tento mechanismus tak umožňuje detekci chyb v přenosu.

Jak je vidět na Obrázek 4, uživatel nejdříve naváže TCP spojení transportní vrstvy. Klient zasílá segment SYN (Synchronize Sequence Number) s počátečním náhodným sekvenčním číslem. Server odpovídá segmenty SYN, také s náhodným číslem, a ACK (Acknowledgement), který potvrzuje počáteční pořadové číslo odeslané klientem zvětšené o jedničku. Uživatel dokončuje proces poslední fází odeslání segmentu ACK s pořadovým číslem serveru a přičtenou jedničkou. Nyní jsou obě strany synchronizovány a navazují spolehlivé spojení pro přenos dalších dat. Toto vyjednávání se označuje jako Třicestný HandShake TCP/IP. [13], [14 str. 70]

Samotná komunikace http není šifrovaná. V dnešní době je již silně nedoporučeno používat nešifrované spojení. Řešením pro webové služby je komunikace přes vrstvu TLS, kde se pak protokol označuje jako HTTPS. Proces se označuje jako TLS handshake a probíhá hned po TCP spojení. Nejčastěji se využívá algoritmu výměny klíče RSA. Klient posílá serveru iniciační zprávu „hello“. Ta obsahuje TLS verzi, šifrovací sady a náhodný řetězec bytů (client random). Server odpovídá zprávou „server hello“, kde se nachází SSL certifikát, vybraná šifrovací sada a další náhodná posloupnost bytů. Klient na základě obdrženého certifikátu ověřuje identitu serveru s certifikační autoritou, aby zjistil, jestli komunikuje opravdu se správným cílem, a ne

například útočníkem. Klient odesílá další náhodná data a šifruje je veřejným klíčem serveru, který obdržel v rámci certifikátu SSL. Oba účastníci generují relační klíč z náhodných dat klienta a serveru. Měli by dojít ke stejnému výsledku. Oba si posílají zašifrovanou zprávu „finished“ a potvrzují správnost klíče. Komunikace nyní přechází do symetrického šifrování pomocí vypočteného relačního klíče. [15]

Na aplikační vrstvě už je následně v režii protokol http komunikující standartně na portu 80, nebo 443. Protokol je na mechanismu klient-server, tedy klient posílá vždy dotaz a server mu relevantně odpovídá. Pokud verze http (1.1 a větší) podporuje funkci HTTP pipelining, může se odesílat více dotazů paralelně naráz, aniž by se čekalo na odpověď přechozího dotazu.

```
1. ===Request===
2. GET / HTTP/1.1
3. Host: google.com
4.
5. ===Response===
6. HTTP/1.1 200 OK
7. Date: Sat, 02 Apr 2022 15:13:37 GMT
8. Expires: -1
9. Cache-Control: private, max-age=0
10. Content-Type: text/html; charset=UTF-8
11.
12. <!doctype html>...
```

Obrázek 6: Struktura HTTP požadavku a odpovědi, zdroj – vlastní

Obrázek 6 ukazuje jednoduchou strukturu komunikace serveru a klienta přes protokol http. Řádek 2 požadavku se označuje jako „Start line“. Začíná metodou, za kterou následuje cesta, na kterou se v rámci URI cílí. Řádek ukončuje protokol s verzí. Následuje seznam hlaviček (obecné, hlavičky požadavku, hlavičky reprezentací). „Host“ určuje cílový server. Tělo žádosti často není potřeba specifikovat. Webový server nám následně odpovídá stavovým řádkem, kde najdeme protokol s verzí, návratový kód a textová reprezentace stavu. Následují hlavičky (obecné, hlavičky odpovědi, hlavičky reprezentací). Odpověď završuje tělo zprávy, které je odděleno prázdným řádkem. Tělo tu taktéž není povinné.

Stavové kódy lze rozdělit do následujících skupin:

- 100–199 pro informační rozsah,

- 200–299 pro indikaci úspěšného požadavku,
- 300–399 pro oznámení přesměrování požadavku,
- 400–499 pro chybový stav klientského požadavku,
- 500–599 pro chybový stav při zpracování na serveru.

Ukončení spojení TCP probíhá přes mechanismus čtyřcestného handshaku. První strana posílá aktivní žádost o uzavření spojení FIN s náhodným číslem a přechází do stavu „FIN\_WAIT 1“. Server potvrzuje pomocí segmentu ACK s obdržným číslem větším o jedničku a přechází do stavu „CLOSE\_WAIT“ pasivního uzavření. Žadatel mění stav na „FIN\_WAIT2“. Nyní se spojení nachází v jednostranně uzavřeném režimu, kde od stanice, která poslala „FIN“, se již neočekává přenos dat. Druhá stanice odesílá „FIN“ s náhodným číslem a přechází do stavu „LAST\_ACK“ a první po obdržení do stavu „TIME\_WAIT“ a posílá poslední potvrzovací segment druhé straně. Spojení se uzavřelo. [13]

### 5.1.2 Historie

Tim Berners-Lee zahájil vývoj protokolu http verze 0.9 v roce 1989. O dva roky později, tedy v roce 1991, byla zveřejněna a specifikována v podobě uvedené na stránce <https://www.w3.org/Protocols/HTTP/AsImplemented.html>. Jednalo se opravdu o primitivní protokol, který nedisponoval hlavičkami. Obsahoval pouze jednu metodu GET. Taková konstrukce přinášela často nedostatky. Například absence specifikace hlavičky „Content-length“ neumožňovala sledovat progres stahovaného obsahu, a tak uživatel nevěděl, jaký poměr objemu dat vůči celku obdržel.

V roce 1996 se s verzí 1.0 již do protokolu zaintegrovaly hlavičky, které umožňují serveru a klientovi přidávat dodatečné informace do požadavků a opovědí HTTP, které se hned po stavovém řádku přidávají každá na nový řádek ve formátu jména hlavičky (nerozlišují se malá a velká písmena) a hodnoty oddělené dvojtečkou. Protokol již není limitován pouze na hypertextové dokumenty díky hlavičce „Content-Type“, a tak lze například přenášet média, skripty nebo soubory se styly. K metodě GET přibyly metody HEAD a POST. Spojení se okamžitě ukončuje po odpovědi, proto se pro každý dotaz otevírá neustále nové spojení pomocí třicestného handshaku TCP. Tato síťová strategie je nicméně hodně náročná na výkon. Verze protokolu byla uveřejněna v dokumentech RFC pod číslem 1945.



Počátek roku 1997 přinesl verzi protokolu 1.1 specifikovanou v článku RFC 2068 a následně její aktualizace vyšla v roce 1999 v RFC 2616. Největším přínosem oproti starším verzím je řešení perzistentního spojení. Hlavička „Keep-Alive“ zajistí stálost, možnost použít vícekrát jedno spojení nebo třeba zřetězení. Společně s parametry „timeout“ a „max“ lze specifikovat časový limit pro spojení nebo maximální počet požadavků. Protokol taktéž umožňuje blokové přenosy, kompresi s dekompresí, dohody o obsahu přenosu, rychlejší odezvy, podporu pro virtuální servery a úspory šířky pásma přidáním podpory pro mezipaměť. Tato verze se dodnes využívá nejčastěji, přestože trpí nadále dost velkými neduhy. Podporovaných metod je již osm a sestávají z: GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT a OPTIONS.

Verze 2 označovaná jako HTTP/2 byla specifikovaná v roce 2015 v článku RFC 7540 a RFC 7541 pro kompresi hlaviček. Základní kameny jako stavové kódy, metody, hlavičky nebo URI protokolu byly zachovány. Přešlo se na novou, binární vrstvu. Komunikační schéma se ale s předchůdcem nemění. Aplikační logika nasazených starších webových aplikací tak zůstává kompatibilní. HTTP/2 využívá výhod komprimace hlaviček, multiplexingu, server-push a prioritizace zdrojů. [12]

Třetí generace je právě ve vývoji. Jejím hlavním cílem je záměna protokolu TCP za UDP s cílem vyšší rychlosti. Jelikož se již pracuje s multiplexem, u TCP spojení se pakety formují v řadě tak, jak byly odeslány, může docházet k blokování připojení, pokud dojde ke ztracení paketu. Protokol Quick UDP Internet Connections (QUIC) vytvoří více spojení mezi klientem a serverem, a tak zamezí zmíněnému problému a čekání. [16]

## 5.2 PHP

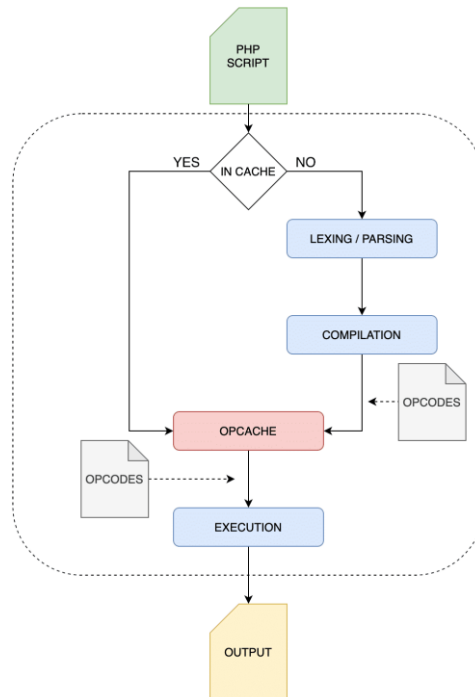
Programovací skriptovací jazyk PHP (Hypertextový preprocesor) nachází pochopení a využití již od roku 1994, kdy poprvé Rasmus Lerdorf položil základní kámen ve formě binárních nástrojů Common Gateway Interface (CGI) napsané v programovacím jazyce C. V následujícím roce byl jeho kód zveřejněn, a tak se mohla celá komunita začít podílet na opravách chyb a dalším vylepšení. Od té doby PHP prošel dlouhým vývojem. [17]

PHP stále patří mezi nejpobulárnější webové jazyky, i když je vytlačován značnou konkurencí, jako je Python, Java nebo jazyky nad C. Jazyk poskytuje příjemné prostředí pro rychlý vývoj a snadné učení společně s mnoha knihovnamy, které PHP obsahuje již od instalace nebo je možné další doinstalovat jednoduše později. Kombinuje jak procedurální, tak objektový přístup s příjemnou moderní syntaxí obdobou jazyků Perl, C nebo Java. [18 str. 13-15]

Hypertextový preprocesor lze stáhnout a nainstalovat samostatně, ale často se využívá již přichystaných balíčků MAMP (Mac, Apache, MySQL, PHP), WAMP (Windows, Apache, MySQL, PHP) a LAMP (Linux, Apache, MySQL, PHP), které dle operačního systému poskytují webový server, databázovou vrstvu a podporu skriptovacího jazyka. Oficiální stránky php.net poskytují podrobnou dokumentaci ve více než osmi jazycích, kterou lze prohlížet přímo na stránkách nebo je možné si ji stáhnout. [19 str. 18], [18 str. 13-15]

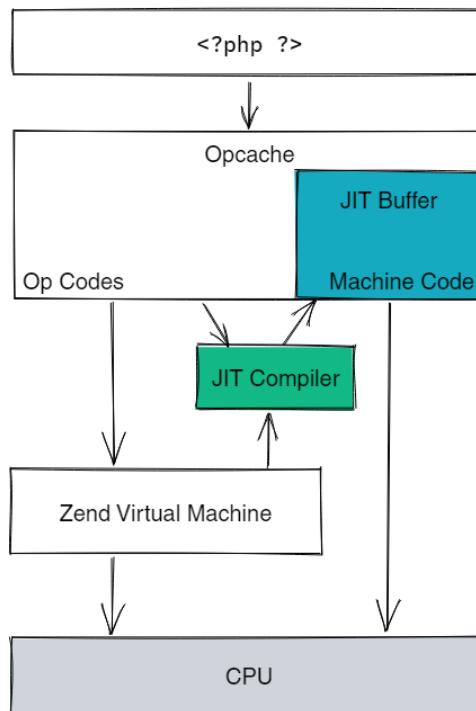
Rozšíření lze přidat manuálně z depozitáře PECL (PHP Extension Community Library), na systému Windows stáhneme požadovanou knihovnu s koncovkou .dll a vložíme do složky „PHP\ext“. Knihovnu přidáme do příslušného konfiguračního souboru php.ini pro načtení a restartujeme službu. Funkci phpinfo() můžeme použít pro zjištění úspěšného načtení. Pokud bychom knihovnu nenašli, musíme se podívat do souboru se záznamy. Knihovny lze také získat pomocí správce balíčků PEAR (PHP Extension and Application Repository). Využit lze taktéž linuxových balíčkovacích systémů, jako je APT. V dnešní době se nicméně stal nejpobulárnější správcem knihoven Composer. Umožňuje instalovat balíčky, pracovat se závislostmi, automaticky začleňovat a načítat balíčky a poskytuje aktualizace. Composer využívají nejmodernější frameworky jako například Laravel a Symfony.

Skript PHP lze spustit přímo na počítači s instalací jazyka, ale častěji ho využívá samotný webserver pro dynamicky řešené stránky. Webový server dostane požadavek od uživatele a vyhodnotí, zdali je odpověď potřeba interpretovat přes PHP procesor, nejjednodušeji podle koncovky souboru, na který je požadavek zaslán. Výsledná odpověď serveru se posílá zpět do prohlížeče, nejčastěji ve formátu HTML, kde se data vykreslí.



Obrázek 7: Zpracování skriptu PHP [20]

PHP 8 využívá jádra Zend, kde se kód spouští ve virtuálním stroji. Kromě nových možností zápisu kódu a zkracování, přichází osmá verze hlavně s kompilátorem JIT (Just In Time). Vylepšuje se tak rychlost interpretace společně s již implementovanými funkcemi Opcache a Opcodes, které zavádějí do paměti bytecode a není potřeba celý kód pokaždé kompilovat, ale jen spouštět. JIT umožňuje vytvořit cache s přímo strojovým kódem pro procesor. [20]



Obrázek 8: Zpracování a cachování kódu PHP [21]

### 5.3 Framework Laravel

Laravel framework je postaven nad jazykem PHP s otevřeným zdrojovým kódem a pod svobodnou licencí MIT (Massachusetts Institute of Technology). Vyvíjí se od roku 2012 a v dnešní době je jeden z nejrozšířenějších a nejpoužívanějších frameworků v rámci daného jazyka. Od osmého února 2022 je nejnovější verze číslo 9, která vyžaduje minimálně verzi PHP 8. Jeho jádro se zakládá na obdobně slavném frameworku Symfony. Ač samotné programování v hypertextovém preprocesoru není nijak obtížné, Laravel vývoj webových aplikací zřetelně usnadňuje a hlavně zpřehledňuje. S širokou komunitou a podrobnou oficiální dokumentací není jakákoliv tvorba ve frameworku náročná. [22]

Laravel využívá softwarové architektury MVC (Model-View-Controller).

- Kontrolér zpracovává a vyhodnocuje aplikační logiku. Typicky je spojován se směrováním. Přebírá parametry od požadavků, pracuje s databází nebo se souborovým

systemem, validuje data, a nakonec předává režii pohledům, které zpracují poskytnuté informace do šablon a vrací kód HTML.

- Model reprezentuje datový objekt. Laravel používá systém Eloquent, který pracuje jako ORM (Object-Relational Mapper). Eloquent mapuje databázové tabulky na objekty a stejně tak k nim jde následně přistupovat. Není zapotřebí dotazovacího jazyka pro práci s daty, stačí volat příslušně implementované metody nad objektem, které si umí poradit i v zřetěžené podobě a převést volání do dotazu na databázi.
- Pohledy poskytuje šablonovací systém Blade. Ten umožňuje do uživatelem předpřipravených šablon vkládat proměnné, cyklit, podmiňovat nebo skládat šablony z dílčích částí apod. Výsledkem je kompletně vygenerovaný HTML kód zaslaný do prohlížeče. Pro ušetření režie a rychlejší odezvu je zpracovaný výsledek uložen do cache.

Další důležitou funkcí, kterou Laravel poskytuje, je směrování (routing). Uživatel zasílá požadavek na server, ten dle příslušné adresy předá obsluhu kontroléru, který žádost zpracuje s případnými parametry. Obslužnou rutinu nemusí definovat jen kontrolér, ale i například anonymní funkce. Záznam ve směrovací tabulce tvoří dále metoda požadavku http. Lze vytvořit i jednoduše cestu pro mnoho metod navázanou na takzvaný kontrolér zdrojů (Resource Controller). Ten obsahuje implicitně sedm metod: index, create, store, show, edit, update, destroy, které jsou navázány automaticky na předpřipravený systém směrovacích cest a tvoří tak rozhraní CRUD (Create, Read, Update, Delete). [23]

Tabulka 2: Směrovací tabulka CRUD Laravel [23]

Metoda	URL adresa	Metoda	Jméno cesty
GET	/comment	index	comment.index
GET	/comment/create	create	comment.create
POST	/comment	store	comment.store
GET	/comment/{id}	show	comment.show
GET	/comment/{id}/edit	edit	comment.edit
PUT/PATCH	/comment/{id}	update	comment.update
DELETE	/comment/{id}	destroy	comment.destroy

Ještě, než požadavek http projde ke kontroléru, může ho zkontrolovat a popřípadě zastavit middleware. Jedná se o vrstvu, která vstupuje implicitně do požadavku a vykonává nad ním určité akce. Registraci vlastního middleware lze vytvořit a napojit na směrovací cestu nebo skupinu cest. Příkladem použití může být autentizace uživatele webové aplikace, kde nechceme, aby nepřihlášený uživatel neměl možnost přístupu k neoprávněným cestám.

Laravel lze tvořit a ovládat nejen přes prostředí editoru, ale i přes rozhraní příkazové řádky. K tomu slouží nástroj Artisan. Artisan umožňuje sledovat stav softwarové struktury, vygenerovat mnoho tříd z předpřipravených šablon (např. kontrolér, databázovou migraci, směrovací cestu nebo uvést aplikaci do režimu údržby). Framework lze obohatit o knihovny pomocí nástroje Composer, který umožňuje instalovat a spravovat uživatelské balíčky PHP zveřejněné v depozitáři [24 str. 1-6].

## 5.4 Jazyk HTML a kaskádové styly

HTML vychází z anglického výrazu „HyperText Markup Language“. Jedná se o webový jazyk definující strukturovaný dokument, kterému rozumí všechny webové prohlížeče umožňující jeho vizuální interpretaci. Konsorcium W3C spravuje další vývoj jazyka HTML. Samotný jazyk se často pojí se styly CSS poskytující design výsledné prezentaci a Javascriptem, který umožňuje dynamickou práci s elementy dokumentu. [25 str. 15]

Základní kámen jazyka HTML položil v roce 1991 Tim Berners-Lee ve svém WWW projektu v CERN pro sdílení informací a výsledků různých vědeckých výzkumů napříč celým světem. Nápad se stal tak populární, že byla rychlá poptávka po nových funkcionalitách. HTML verze 2.0 byla specifikována ve článku RFC 1866 následujícího roku a přidala možnost pracovat s formuláři. Vývoj se snaží striktně zachovat kompatibilitu s předchůdci. Verze 3.0 vylepšuje výslednou vizuální podobu dokument a uživatel je již schopen psát matematické vzorce a tabulky. Společenství W3C převzalo zodpovědnost za správu standartu a uvádějí verzi 3.2. Rok 1997 přinesl verzi 4.0 s jednotnou podporou skriptů, objektů a rámečku. Brzo potom se objevil bratr jazyka, významný XML pro strukturovanou práci s daty. Nejmodernější podoba hypertextu se označuje jako HTML5. [26]

HTML dokumenty lze prohlížet přímo na osobním počítači z lokálního souborového systému. Většinou se soubory označují koncovkou .html. Užitečné mohou být pro generování dokumentace nebo reportů, které mají příjemný a přehledný zjev, a tak se v nich dobře čte. Nicméně největším přínosem je umístění na dokumentů přímo na webový server pro možnost sdílet informace napříč širokou sítí. Samotný soubor se skládá z tagů a atributů. Tagy ohraničují elementy na stránce a atributy specifikují přesnější stav a chování. Sémantiku tvoří celkem okolo 110 tagů. Ty mohou být rozděleny podle toho, jestli jsou párové a nepárové. Základní struktura dokumentu obsahuje označení typu, hlavičku a tělo.

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title>Titulek celé stránky</title>
5.   </head>
6.   <body>
7.     <h1>Nejvyšší nadpis</h1>
8.     <p>Odstavec textu</p>
9.   </body>
10. </html>
```

Obrázek 9: Ukázka základní struktury HTML, zdroj – vlastní

CSS (Cascading Style Sheets) umožňují samotný HTML dokument vizuálně přizpůsobit našim představám. I mnoho jiných technologií používá standart pro stylizaci prvků. Kaskádové styly navrholo také konsorcium W3C a nyní se o něj stará skupina CSS Working Group. Styly se definují přes pravidla, které vymezují skupinu deklarací. Pravidlo se skládá ze selektoru a bloku. V bloku se vyskytují vlastnosti s hodnotou oddělenou dvojtečkou. Jednotlivé vlastnosti se kategorizují do modulů. Jelikož se deklarace můžou přepisovat s posloupným tokem dokumentu, může se využít označení hodnoty „!important“, která zamezuje přepsání hodnoty jinou možnou pozdní deklarací se stejným identifikátorem. [27]

Mezi základní selektory patří [28]:

- \* - pro všechny elementy,
- #id – pro element s příslušným atributem id,

- `.class` – pro elementy s příslušnou třídou,
- `element` – pro přesně definované elementy
- `element.class` – pro konkrétní elementy obsahující zmíněnou třídu,
- `element .class` – pro všechny elementy s konkrétní třídou uvnitř specifikovaného typu elementu,
- `element, .class` – pro všechny typy specifikovaného elementu a elementy s označenou třídou,
- `element > .class` – pro všechny elementy se specifickou třídou, kteří mají jako rodiče specifikovaný element,
- `element+element` – pro první element následující po elementu,
- `element~element` – pro všechny elementy následující po elementu,
- `[atribut=hodnota]` – pro všechny elementy se specifickým atributem a hodnotou.

Jelikož CSS neumožňuje mnoho programovacích praktik, existují například preprocesorové skriptovací jazyky Sass a Less. Ty podporují proměnné, cykly, vnořování, funkce s argumenty nebo dědičnost. Vytvářejí tak přehlednou strukturu, kterou lze jednodušeji spravovat. Nevýhodou setrvává nutnost kompilovat kód do stylů CSS po každé úpravě.

## 5.5 MySQL databáze a jazyk SQL

MySQL je populární relační databázový systém pro menší až střední projekty. Databáze je rozšířena celosvětově bez ohledu na přesný záměr. Na trhu se vyskytuje již od poloviny devadesátých let dvacátého století. Není omezená na operační systém, tak ji lze nainstalovat snadno na skoro libovolnou platformu. Jedná se o svobodný software pod licencí GPL, ale může být i v komerčním provedení. Je vyvíjena společností Oracle Corporation v jazycích C a C++. Její aktuální verze se označuje jako osmá řada. Databáze vyniká jednoduchostí, rychlostí a širokou možností implementace. MySQL se skládá ze schémat obsahujících tabulky (sloupce a řádky), pohledy, trigger, procedury apod. [29]

První vrstvou serveru jsou síťové služby pro obsluhu databáze. Konkrétním spojením je přiřazeno separátní vlákno a jednotlivé instance se snaží server udržet naživu vůči žadatelům. Na druhé vrstvě probíhá rozebírání a analýza dotazů společně s vyrovnávací pamětí a rozhraním



pro databázový engine. Poslední vrstvu tvoří ukládací engine samotný, s kterým se komunikuje pomocí API.

Výchozí úložný engine pro MySQL je InnoDB, který je nejvíce používán s podporou transakcí, referenční integrity, zamykání řádků a zotavení. MyISAM je druhý nejpoužívanější. Jeho využití najde hlavně webové uložení. Zamykání provádí až na tabulkové úrovni a nedisponuje transakčním systémem. Oba umějí pracovat s fulltextovým vyhledáváním.

Mezi další úložné engine patří [29]:

- CSV,
- Memory,
- Archive,
- Merge,
- Federated,
- Example,
- Blackhole.

MySQL používá populární, standardizovaný jazyk SQL (Structured Query Language) pro dotazování na databázi. Tento způsob je implementován u téměř veškerých relačních databází. Organizace ANSI (American National Standards Institute) s Mezinárodní organizací pro normalizaci (ISO) se stará o standardy jazyka. SQL se může vyskytovat v mnoha verzích, a tak je nutné si dát pozor, k jaké databázi se pojí.

Dotazovací příkazy lze rozdělit do čtyř základních kategorií [30 str. 144], [31]:

- DDL (Data Definition Language) – patří sem příkazy jako CREATE, ALTER, DROP, COMMENT, RENAME, TRUNCATE,
- DML (Data Manipulation Language) – obsahuje příkazy SELECT, INSERT, DELETE, UPDATE, CALL, LOCK TABLE, MERGE, EXPLAIN PLAN,
- DCL (Data Control Language) – spadá sem GRANT a REVOKE,
- TCL (Transaction Control Language) – transakční příkazy COMMIT, ROLLBACK, SAVEPOINT, SET TRANSACTION.

## 5.6 JavaScript

JavaScript je jeden z nejrozšířenějších skriptovacích jazyků. Jeho využití není limitováno platformou. ECMAScript je pojmenování pro jeho standardizovanou verzi vycházející z příslušné asociace spravující jazyk. Ač samotný název může přinášet zmatečnost, JS a Java nemají kromě syntaktické rodiny nic společného. Hojně se dnes vyskytuje na webových stránkách, kde skripty obsluhují příslušné elementy dokumentu HTML a dávají souboru život. Webový prohlížeč na straně uživatele se následně stará o jeho interpretaci. Uplatnění se ale může rozšířit i na stranu serverových strojů. Jedno ze softwarových řešení může být Node.js poskytující rychlé a spolehlivé prostředí pro vývoj nejčastěji webových aplikací založené na reálném čase. Disponuje i podporou například pro operace nad souborovým systémem, JS interpretovaný prohlížečem si takové operace nemůže dovolit. Využití můžeme najít i v operačních systémech pro automatizaci dávkovými skripty. [32]

Nad Javascriptem se zrodilo mnoho aplikačních rámců. Mezi ty nejpoužívanější patří React, Angular a Vue.js. Ty se nejčastěji využívají pro webové aplikace a práci s DOM. Ulehčují vývoj a zpřehledňují kód. Zvláště pro rozsáhlejší projekty se stávají aplikační rámce JS nezbytností. Za zmínku stojí i nadstavba TypeScript, která rozšiřuje funkcionalitu JS o statické typování, moduly, objektový přístup a mnoho jiného. TypeScript je kompatibilní s čistým Javascriptem a nutno zmínit, že tento vyšší kód se musí kompilovat do JS.

## 6 NÁVRH APLIKACE

### 6.1 Požadavky

#### 6.1.1 Funkční požadavky

- FP1) Hlavní obrazovka bude zobrazovat žebříček tvůrců s nejlepšími výsledky.
- FP2) Jedna stránka žebříčku poskytne zobrazení deseti tvůrců formou paginace.
- FP3) Půjde přepínat mezi žebříčky s hodnocením, počty odběratelů, průměrnými zhlédnutími, průměrnými počty palců na videu a průměrnými reakcemi u videa ve formě komentářů.

- FP4) Žebříček zobrazuje pořadové číslo, jméno tvůrce a příslušnou informaci.
- FP5) Žebříček filtruje pouze aktivní profily.
- FP6) Pro všechny stránky bude nad hlavním obsahem zobrazena lišta s logem, vyhledávací funkcí a odkazem na sekci účtu.
- FP7) Vyhledávání bude hledat zadaný výraz v indexovaných jménech profilů tvůrců. V případě shody se aplikace přepne na požadovaný profil.
- FP8) Neautentizovaný uživatel uvidí v účtové sekci dva formuláře.
- FP9) Jeden formulář umožňuje přihlášení uživatele přes emailovou adresu a heslo.
- FP10) Druhý formulář zajišťuje registraci uživatele a vyžaduje přezdívku, emailovou adresu a heslo.
- FP11) Server validuje data pro registraci následujícím způsobem: IP adresa požadavku musí být unikátní v databázi registrovaných uživatelů, přezdívka je povinná v rozsahu délky tří až šestnácti znaků. Ta se musí skládat z alfanumerických znaků a musí být unikátní, nezabraná. Email musí být validního formátu, který se v databázi ještě nenachází. Heslo vyžaduje délku šesti až třicet dva znaků.
- FP12) Uživatelské heslo musí být uloženo v nečitelné bezpečné formě.
- FP13) Přihlašovací žádost ověří správné údaje společně s kontrolou, jestli je účet aktivní.
- FP14) Nový uživatel obdrží implicitně roli obyčejného uživatele.
- FP15) Obrazovka po úspěšném přihlášení poskytuje změnu hesla, seznam hodnocení a odhlášení.
- FP16) Uživatel s rolí administrátora uvidí navíc odkaz na administraci.
- FP17) Administrace obsahuje správu uživatelů a profilů tvůrců.
- FP18) Správa uživatelů zahrnuje vyhledávání přes přezdívku, umožňuje změnu role, změnu stavu účtu (aktivní/neaktivní) a smazání účtu.
- FP19) Administrátor uvidí po najetí myši na přezdívku uživatele v seznamu informací o příslušné emailové adrese účtu a adrese IP.
- FP20) Správa kanálů zahrnuje vyhledávání přes jméno tvůrce kanálu.
- FP21) Administrace kanálů umožňuje měnit stav profilu (aktivní/neaktivní) a mazání profilů.
- FP22) Administrace poskytuje funkci přidání profilu s automaticky agregovanými informacemi pro tvůrce na serveru YouTube přes modální okno.

- FP23) Přidávání tvůrce probíhá přes prvotní vyhledání jména kanálu, které vrátí ikonu kanálu a ID kanálu. Administrátor musí výsledek hledání potvrdit a tím se vytvoří lokální profil. Proces a data probíhají přes YouTube Data API v3.
- FP24) Profil tvůrce ukazuje profilový obrázek použitý na kanále YouTube společně s popisem, počtem sledujících, celkovým počtem zhlédnutí videí na kanále a počet autorem nahraných dostupných videí.
- FP25) Pod jménem profilu se zobrazí grafická lišta znázorňující celkové hodnocení ve formě procentuální hodnoty z nejlepšího možného hodnocení určující šířku barevné lišty, která je vizualizována gradientem od sytě červené po sytě zelenou barvu.
- FP26) Profil obsahuje blok s průměrnými statistickými daty (počet palců, komentářů, zhlédnutí) z posledních deseti autorem vydaných videí.
- FP27) Se statistickými daty na profilu vyhodnocené ze vzdáleného rozhraní se zobrazí i značky kijkwijzer: „vhodné od věku“ se zobrazí příslušná značka s největším počtem hodnocení v lokálním systému a dále četnost hodnotících značek „násilí“, „strach“, „sex“, „vulgární mluva“, „diskriminace“, „alkoholové/drogové zneužívání“.
- FP28) Poslední blok zobrazuje všechny komentáře od uživatelů od nejnovějšího data.
- FP29) Počet zobrazovaných komentářů najednou může být pro přehlednost pouze 5 s využitím strategie paginace.
- FP30) Komentář zobrazuje autora, celkové hodnocení, text komentáře, datum změny.
- FP31) Pokud je uživatel přihlášen jako administrátor, jednotlivé komentáře je možnost mazat.
- FP32) Přihlášenému uživateli se ještě zobrazí blok s možností komentování a hodnocení se vstupními poli pro textový komentář, celkové hodnocení 1-5 pomocí radio-button a nepovinné prvky: radio-button pro značku věkové vhodnosti obsahu a checkboxy označující obsah typu „násilí“, „strach“, „sex“, „vulgární mluva“, „diskriminace“, „alkoholové/drogové zneužívání“.
- FP33) Profily se budou aktualizovat po jednom dni.
- FP34) Aplikace bude zaznamenávat změny dat profilu.

### 6.1.2 Nefunkční požadavky

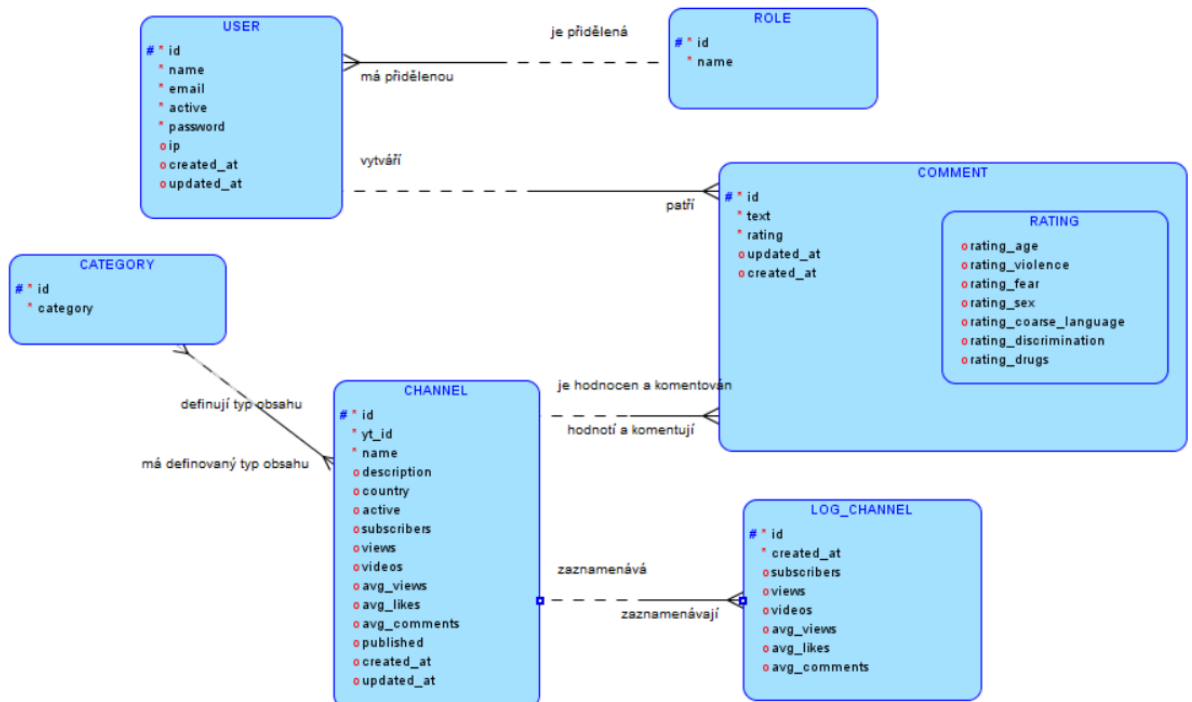
- NFP1) Webová aplikace poběží na linuxovém systému.
- NFP2) Systém bude veřejně dostupný.
- NFP3) Server bude používat veřejnou IPv4 adresu.

- NFP4) Serverovou adresu a její překlad bude zajišťovat doménové jméno.
- NFP5) Správu serveru umožní protokol SSH.
- NFP6) Apache 2.4. bude poskytovat webový server.
- NFP7) Komunikace mezi klientem a serverem bude probíhat na http/2.
- NFP8) Komunikaci bude chránit vrstva SSL s certifikátem od společnosti Let's Encrypt.
- NFP9) Dynamický obsah bude zajišťovat PHP 8.
- NFP10) Preprocesorový framework zastane Laravel 8.x.
- NFP11) Aplikace bude dodržovat architekturu model-view-controller.
- NFP12) Webové skripty zajistí VanillaJS, jQuery a jQuery UI.
- NFP13) Relační databáze MySQL 8 bude poskytovat datovou vrstvu pro perzistentní ukládání dat.
- NFP14) Databáze bude splňovat metodiku třetí normální formy.
- NFP15) Databáze bude mít nastaveny integritní omezení.
- NFP16) Databáze bude dodržovat jmenné konvence Snake case.
- NFP17) Pro databázi bude vytvořen jeden účet pro přístup aplikace.
- NFP18) Databázový účet se bude moct připojit pouze z lokální sítě.
- NFP19) Prezentace bude moderní a příjemná.
- NFP20) Dokumenty budou psány v HTML 5.
- NFP21) Grafiku utvoří kaskádové styly CSS3.
- NFP22) Ikony pro ovládací prvky poskytne služba Font Awesome.
- NFP23) Značkování obsahu musí být ve vizuální formě kijklwizer.
- NFP24) Aplikace bude mít vlastní ikonu a logo.
- NFP25) Systém poskytne základní dvě role: uživatel, administrátor.
- NFP26) Systém použije YouTube Data API v3 se základním klíčem.
- NFP27) Aplikace zajistí nepřesázení denního limitu požadavků na API se stropem deset tisíc.
- NFP28) Aplikace musí být optimalizovaná pro minimálně deset tisíc požadavků za den.
- NFP29) Přístup k aplikaci musí být příjemný i na mobilním zařízení.
- NFP30) Aplikace bude řešit základní bezpečnostní problematiku webových stránek.

## 6.2 Databázový návrh

### 6.2.1 Entitně-vztahový diagram (ERD)

ERD je nástroj softwarového inženýrství pro vizualizaci a modelování dat. Používá se nejčastěji pro koncept relační databáze. Diagram zobrazuje entity se sledovanými stavy a jejich vzájemné vztahy. Obrázek 10 ukazuje diagram modelované webové aplikace.



Obrázek 10: ERD model aplikace, zdroj – vlastní

## 6.2.2 ERD věty

Jazyk ERDish pomáhá překlenout problémy s komunikací mezi klientem a vykonavatelem a definovat pravidla v rámci byznysových pravidel v entitně-relační formě. Skládá se z konstantního slova „každý“, které začíná větu. Následuje název zdrojové entity s volitelností (může/musí) společně se slovesem definující vztah. Věta pokračuje kardinalitou a názvem druhé entity. Věty se tvoří obousměrně.

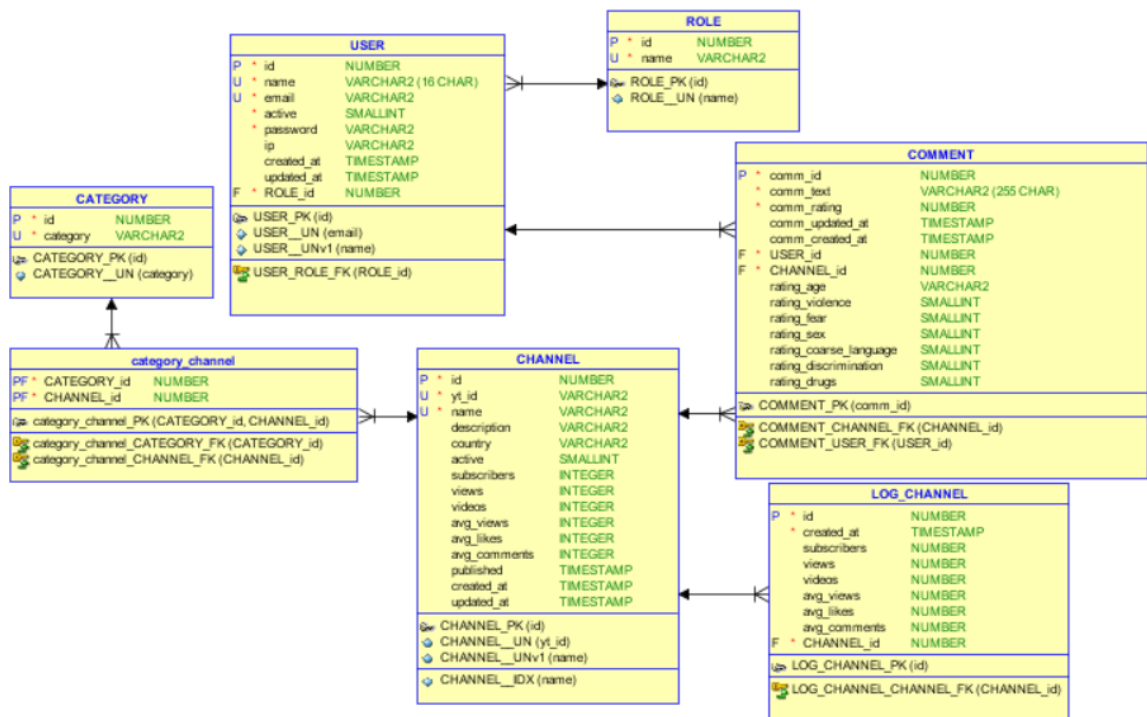
ERDish v případě webové aplikace vypadá takto:

- Každý uživatel musí mít přidělenou právě jednu roli.
- Každá role může být přidělena k několika uživatelům.
- Každý uživatel může vytvářet několik komentářů.

- Každý komentář musí patřit právě jednomu uživateli.
- Každý komentář musí hodnotit a komentovat právě jeden kanál.
- Každý kanál může být hodnocena a komentována několika komentáři.
- Každý kanál musí mít definovaný obsah několika kategoriemi.
- Každý kanál může být zaznamenán v několika ložích.
- Každý log musí zaznamenávat právě jeden kanál.
- Každá kategorie můžou definovat typ obsahu několika kanálů.

### 6.2.3 Relační model

Relační model zobrazuje možnou datovou implementaci abstraktního modelu například pro databázi. Obrázek 11 ukazuje relační model webové aplikace. Entity transformované na tabulky získaly cizí klíče podle definovaných vztahů. Subtyp hodnocení se začlenil do super typu tabulky komentářů a spojily se atributy. Relace M:N se rozpadla na spojovací tabulku kategorií a kanálů s kompozitním klíčem cizích spojovaných tabulek. Číselné identifikátory získají sekvenci pro auto-inkrementaci.



Obrázek 11: Relační model aplikace, zdroj – vlastní

Uživatelé mají integritní omezení pro přezdívku a emailovou adresu, které musejí být unikátní. Stejně tak role společně s kategoriemi se nesmí v databázi duplikovat. Kanál ukládá jedinečný identifikátor z YouTube, a i jméno profilu musí být unikátní. Jelikož se přes toto jméno vyhledává, je na sloupci aplikován index. Uživatelská přezdívka je omezena na 16 znaků a komentáře omezeny na 255 znaků.

Výsledný seznam použitých tabulek vypadá takto:

- **CATEGORIES** – Definované kategorie obsahu portálu YouTube.
- **COMMENTS** – Uživatelské komentáře k příslušným profilům tvůrců obsahující hodnocení.
- **CHANNEL\_CATEGORIES** – Spojová tabulka pro přiřazení vícero kategorií pro profilovou stránku.
- **CHANNELS** – Tabulka obsahuje samotné profily tvůrců.
- **LOG\_CHANNELS** – Tabulka pro uchování historických dat při změně údajů v tabulce *CHANNELS*.
- **ROLES** – Tabulka pro seznam dostupných uživatelských rolí.
- **USERS** – Tabulka s daty účtů.

Za číselníky považujeme dvě tabulky: **ROLES** a **CATEGORIES**, jelikož slouží jako neměnné výčtové typy.

#### 6.2.4 Logování změn profilů

Statistická data bude ukládat automaticky databázový spouštěč aplikovaný na tabulku „CHANNELS“. Po jakékoliv aktualizaci tabulky se záznam s hodnotami přidá do tabulky „LOG\_CHANNELS“, kde se data budou postupně agregovat a nepřepisovat. Data v budoucnu mohou sloužit jako vizuální výstup pro daný profil formou grafu zobrazující časový průběh sledovaných aspektů.

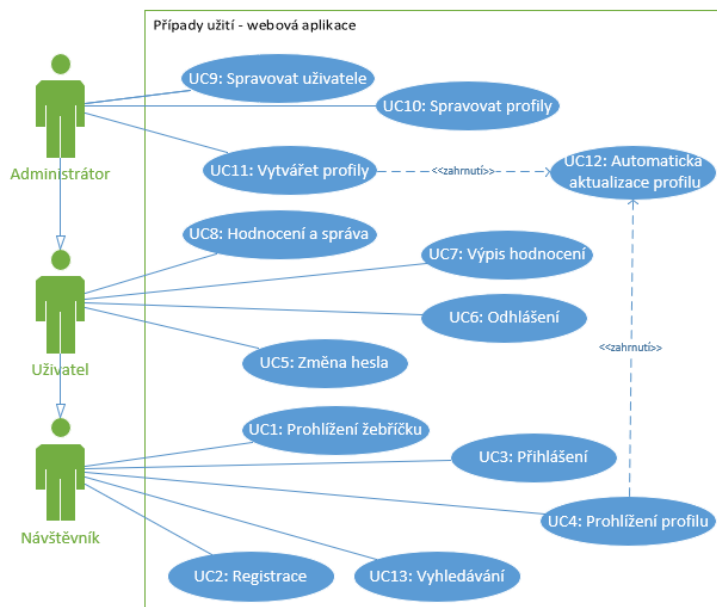


Databázový spouštěč vypadá takto:

```
1. CREATE TRIGGER `trigger_log_channel` AFTER UPDATE ON `channels` FOR EACH ROW as
2. INSERT INTO log_channels (channel_id, created_at, subscribers, views, videos, avg_views,
   avg_likes, avg_comments)
3. VALUES (OLD.id, now(), NEW.subscribers, NEW.views, NEW.videos, NEW.avg_views,
   NEW.avg_likes, NEW.avg_comments);
4.
```

### 6.3 Příklad užití

Use case diagram se používá pro vizualizaci obecné funkčnosti informačního systému. Diagram ukazuje, co se od systému bude požadovat a co bude umět. Samotná implementace a chování systému nás nezajímá. Jedná se nejčastěji o prvotní formu návrhu, aby se strany shodli na představě funkčnosti. Diagram se nejčastěji skládá z aktérů, systému, případů užití a vztahů mezi objekty.



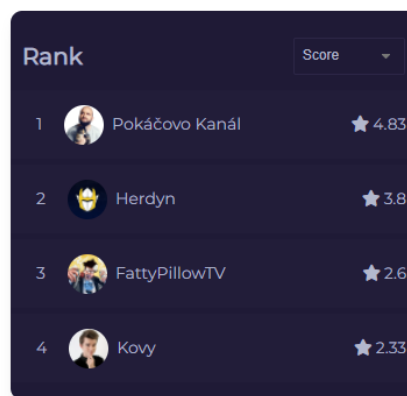
Obrázek 12: Případy užití webové aplikace, zdroj – vlastní

Obrázek 12 vyjadřuje specifické užití webové aplikace. Diagram ukazuje, že v systému budou hrát roli tři aktéři, kteří mají možnost vyvolat 13 zobecněných případů užití.

## 6.4 Specifikace případů užití a uživatelské rozhraní

### 6.4.1 Příklad užití UC1

- **Název:** Prohlížení žebříčku.
- **Popis:** Aktéři si prohlíží žebříček, který srovnává profily tvůrců dle sledovaného kritéria.
- **Aktéři:** návštěvník, uživatel, administrátor.
- **Podmínky vzniku:** vstup aktéra na hlavní stránku s cestou „/“ nebo kliknutím na logo aplikace.
- **Základní tok:**
  1. Aktér vstoupí na stránku.
  2. Systém nabídne žebříček tvůrců s výchozím parametrem lokálního skóre.
  3. Pokud uživatel klikne na položku, zobrazí se profil tvůrce.
  4. Pokud uživatel změní položku filtru výběrového menu, automaticky se žebříček transformuje.
- **Návrh UI:**

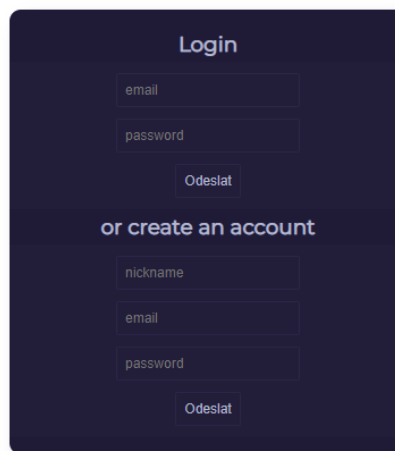


Obrázek 13: Návrh žebříčku, zdroj – vlastní

- **Dokončení:** opuštění stránky.

## 6.4.2 Příklad užití UC2

- **Název:** Registrace.
- **Popis:** Aktéři si vytváří uživatelský účet pro možnost hodnocení.
- **Aktéři:** návštěvník.
- **Podmínky vzniku:** Aktér není přihlášen.
- **Základní tok:**
  1. Aktér vstoupí na uživatelskou stránku.
  2. Systém zkontroluje, jestli není aktér přihlášen, jinak podstrčí přehled účtu.
  3. Aktér zadává přezdívku, emailovou adresu a heslo a odesílá formulář.
  4. Systém validuje vstupní data.
  5. Pokud vstupy neprochází validátorem, vrátí se příslušná nesplněná podmínka a tok končí.
  6. Systém vkládá nový účet do databáze.
- **Návrh UI:**



Obrázek 14: Návrh přihlášení a registrace, zdroj – vlastní

- **Dokončení:** Systém zapíše nový účet do databáze.

## 6.4.3 Příklad užití UC3

- **Název:** Přihlášení.
- **Popis:** Aktéři se přihlašují ke svému stávajícímu účtu.

- **Aktéři:** návštěvník.
- **Podmínky vzniku:** Aktér není přihlášen.
- **Základní tok:**
  1. Aktér vstoupí na uživatelskou stránku.
  2. Systém zkontroluje, jestli není aktér přihlášen, jinak podstrčí přehled účtu.
  3. Aktér zadává emailovou adresu a heslo účtu a odesílá formulář.
  4. Systém kontroluje status účtu a přihlašovací údaje v databázi.
  5. Pokud vstupy neprochází validátorem, vrátí se příslušná nesplněná podmínka a tok končí.
  6. Systém přihlašuje identifikovaný účet.
- **Dokončení:** nastavení dočasného relačního klíče.

#### 6.4.4 Příklad užití UC4

- **Název:** Prohlížení profilu.
- **Popis:** Aktéři si prohlíží konkrétní lokální profil tvůrce videí.
- **Aktéři:** návštěvník, uživatel, administrátor.
- **Podmínky vzniku:** vstup aktéra na adresu url se jménem profilu.
- **Základní tok:**
  1. Aktér vstoupí na profilovou stránku.
  2. Systém zkontroluje poslední aktualizaci profilu.
  3. Pokud jsou data více než den staré, systém vyvolá případ automatické aktualizace profilů.
  4. Systém vrací profilovou stránku.
- **Návrh UI:**



Obrázek 15: Návrh profilu, zdroj – vlastní

- **Dokončení:** opuštění stránky.

#### 6.4.5 Příklad užití UC5

- **Název:** Změna hesla.
- **Popis:** Aktéři si mění heslo ke svému uživatelskému účtu.
- **Aktéři:** uživatel, administrátor.
- **Podmínky vzniku:** Aktér musí být přihlášen.
- **Základní tok:**
  1. Aktér vstoupí na stránku účtu.
  2. Aktér kliká na ikonku klíče.
  3. Systém zobrazuje formulář se vstupním polem pro nové heslo.

4. Uživatel vyplňuje a odesílá formulář.
  5. Systém validuje požadavek.
  6. Pokud vstupy neprochází validátorem, vrátí se příslušná nesplněná podmínka a tok končí.
  7. Otisk nového hesla se aktualizuje v databázi.
- **Dokončení:** aktualizace hesla příslušného účtu v databázi.

#### 6.4.6 Příklad užití UC6

- **Název:** Odhlášení.
- **Popis:** Aktéři se odhlašují z navázané relace.
- **Aktéři:** uživatel, administrátor.
- **Podmínky vzniku:** Aktér musí být přihlášen.
- **Základní tok:**
  1. Aktér vstoupí na stránku účtu.
  2. Aktér kliká na odkaz s textem odhlášení.
  3. Systém ničí relační klíč a vrací stránku přihlášení a registrace.
- **Dokončení:** zrušení existující relace.

#### 6.4.7 Příklad užití UC7

- **Název:** Výpis hodnocení.
- **Popis:** Aktéři si prohlíží seznam všech svých udělených hodnocení.
- **Aktéři:** uživatel, administrátor.
- **Podmínky vzniku:** Aktér musí být přihlášen.
- **Základní tok:**
  1. Aktér vstoupí na stránku účtu.
  2. Aktér kliká na ikonku bubliny komentáře.
  3. Systém zobrazuje tabulku všech hodnocení od nejnovějšího data.
- **Dokončení:** opuštění stránky.

#### 6.4.8 Příklad užití UC8

- **Název:** Hodnocení a správa.
- **Popis:** Aktéři komentují a hodnotí dostupné tvůrce s možností změny a odebrání interakce.
- **Aktéři:** uživatel, administrátor.
- **Podmínky vzniku:** Aktér musí být přihlášen.
- **Základní tok:**
  1. Aktér vstoupí na stránku profilu.
  2. Systém kontroluje autentizaci uživatele.
  3. Pokud je uživatel přihlášen:
    - i. Systém zobrazuje hodnotící formulář.
    - ii. Systém kontroluje existenci uživatelského hodnocení.
    - iii. Pokud komentář od uživatele pro příslušný profil existuje, systém vyplňuje data z databáze pro úpravu a zobrazuje možnost smazání.
    - iv. Pokud aktér klikne na ikonu šipky, zobrazí se další rozšířené možnosti hodnocení.
    - v. Pokud aktér klikne na tlačítko odeslat, je komentář s hodnocením upraven, či vytvořen v databázi.
    - vi. Pokud aktér klikne na odkaz s textem odstranit, komentář je z databáze odebrán.

- **Návrh UI:**

The image shows a dark-themed user interface for rating a YouTuber. At the top, it says 'Rate Youtuber'. Below that, there's a 'Total score:' section with five radio buttons labeled 'One', 'Two', 'Three', 'Four', and 'Five'. The 'Five' button is selected. Underneath is a 'Suitable for age:' section with four radio buttons: '16+', '12+', '6+', and 'All'. The 'All' button is selected. The 'Content Categories:' section has several checkboxes: 'Violence', 'Fear', 'Sex', 'Coarse language', 'Discrimination', and 'Drugs/alcohol abuse', all of which are currently unchecked. At the bottom, there's a text input field containing the text 'Trefnější, satirické a vtipné písničky v česku nenajdeš'. Below the input field are two buttons: 'Submit comment' and 'Delete'.

Obrázek 16: Návrh hodnocení, zdroj – vlastní

- **Dokončení:** aktualizace dat komentáře v databázi.

#### 6.4.9 Příklad užití UC9

- **Název:** Spravovat uživatele.
- **Popis:** Aktéři si zobrazují a mění role a status uživatelských účtů.
- **Aktéři:** administrátor.
- **Podmínky vzniku:** Aktér musí být autorizován jako administrátor.
- **Základní tok:**
  1. Aktér vstoupí na stránku administrace uživatelů.
  2. Systém kontroluje příslušnou autorizaci uživatele, jinak tok končí.
  3. Systém zobrazuje seznam uživatelů s možností změny statusu, role a smazání.
  4. Když aktér zadá výraz do vyhledávacího pole a zmáčkne tlačítko lupy, systém vrátí uživatele s přezdívkou obsahující vstupní výraz.
  5. Když aktér klikne na ikonu fajfky vedle položky, systém aktualizuje data příslušného účtu v databázi.
  6. Když aktér klikne na ikonu odpadní koše vedle položky, systém smaže data příslušného účtu v databázi.
- **Dokončení:** aktualizace uživatelských účtů v databázi.

#### 6.4.10 Příklad užití UC10

- **Název:** Spravovat profily.
- **Popis:** Aktéři si zobrazují místní profily, mění jejich status a mají možnost mazat.
- **Aktéři:** administrátor.
- **Podmínky vzniku:** Aktér musí být autorizován jako administrátor.
- **Základní tok:**
  1. Aktér vstoupí na stránku administrace profilů.
  2. Systém kontroluje příslušnou autorizaci uživatele, jinak tok končí.
  3. Systém zobrazuje seznam profilů s možností změny statusu a smazání.
  4. Když aktér zadá výraz do vyhledávacího pole a zmáčkne tlačítko lupy, systém vrátí profil se jménem obsahující vstupní výraz.
  5. Když aktér klikne na ikonu fajfky vedle položky, systém aktualizuje data příslušného profilu v databázi.

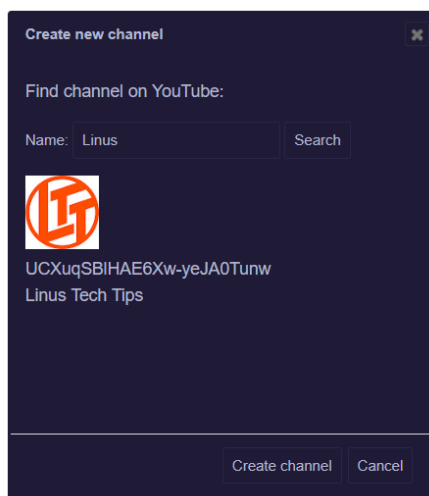


6. Když aktér klikne na ikonu odpadní koše vedle položky, systém smaže data příslušného profilu v databázi.

- **Dokončení:** aktualizace místních profilů v databázi.

#### 6.4.11 Příklad užití UC11

- **Název:** Vytvářet profily.
- **Popis:** Aktéři vytváří místní profily tvůrců.
- **Aktéři:** administrátor.
- **Podmínky vzniku:** Aktér musí být autorizován jako administrátor.
- **Základní tok:**
  1. Aktér vstoupí na stránku administrace profilů.
  2. Systém kontroluje příslušnou autorizaci uživatele, jinak tok končí.
  3. Aktér kliká na ikonu symbolu plus.
  4. Systém otevírá modální okno.
  5. Aktér zadává jméno tvůrce, pod kterým vystupuje na portálu YouTube a kliká na tlačítko s textem hledat.
  6. Systém vrací výsledek z dotazu na YouTube API.
  7. Když aktér klikne na tlačítko s textem vytvořit kanál, vyvolá se případ automatické aktualizace profilů se vstupním parametrem identifikátoru kanálu.
- **Návrh UI:**



Obrázek 17: Návrh tvorby profilu, zdroj – vlastní

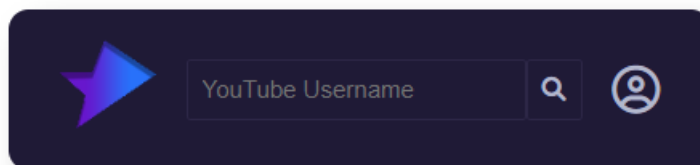
- **Dokončení:** vytvoření profilu v databázi.

#### 6.4.12 Příklad užití UC12

- **Název:** Automatická aktualizace profilu.
- **Popis:** Aktéři vyvolávají implicitně funkci pro aktualizaci dat místních profilů.
- **Aktéři:** návštěvník, uživatel, administrátor.
- **Podmínky vzniku:** Implicitní vyvolání v případě vytvoření profilu, nebo v případě zobrazení profilu.
- **Základní tok:**
  1. Systém si vyžádá data z YouTube API pro příslušný kanál (popis, země, odběratelé, zobrazení, videa).
  2. Pokud je dotaz úspěšný:
    - i. Pokud místní profil neexistuje, vytvoří se.
    - ii. Přijatá data se zapíší k profilu.
- **Dokončení:** synchronizace dat z YouTube API s místní databází.

#### 6.4.13 Příklad užití UC13

- **Název:** Vyhledávání.
- **Popis:** Aktéři vyhledávají místní profil.
- **Aktéři:** návštěvník, uživatel, administrátor.
- **Podmínky vzniku:** žádné.
- **Základní tok:**
  1. Aktér zadá výraz do vstupního pole a klikne na ikonu lupy.
  2. Systém vyhledá profil přes atribut jména.
  3. Pokud je vyhledávání úspěšné, je aktér přesměrován na stránku profilu prvního výsledku, jinak je vráceno ohlášení neúspěchu.
- **Návrh UI:**



Obrázek 18: Návrh vyhledávání, zdroj – vlastní

- **Dokončení:** opuštění stránky.

## 6.5 Vývoj aplikace, nástroje a služby

Po nadefinování požadavků se v softwaru Microsoft Visio Professional 2021 vytvořily diagramy UML. SQL Developer Data Modeler od Oracle pomohl s modelováním diagramů databáze. Autor využil studijních licencí k získání profesionálního vývojového prostředí PhpStorm od společnosti JetBrains. Software poskytuje nejen efektivní editor pro psaní kódu s širokou podporou jazyků, formátování, napovídání a zvýrazňování syntaxe, ale i nástroje pro práci se soubory v projektu a jejich prohlížení a pro správu databázového systému pomocí zabudovaného klienta. Návrh a tvorba grafických podkladů probíhala v Adobe Photoshop 2020.

Webový server byl nasazen a vyvíjen na již stávající pronajaté řešení virtuálního privátního serveru na platformě Contabo na programu CLOUD VPS S se čtyřmi jádry a osmi gigabajty operační paměti v německém Norimberku. 64bitový operační systém Ubuntu 18.04 zajišťuje rychlé a stabilní prostředí.

Použitá doména druhého řádu s TLD Evropské unie je registrována u poskytovatele Wedos a proprietárními servery DNS. Doména je napojena na německý server pomocí záznamu typu A, tedy pomocí IPv4 adresy hostitele.

K ovládní serveru se využilo protokolu SSH přes knihovni podporu aplikace Terminal pro operační systémy Windows. Napojení na vzdálený server umožňuje samotný PhpStorm s konzolovým přístupem a správou souborového systému a synchronizací změn v reálném čase přes protokol SFTP. Díky SSH lze tunelovat i do lokální databáze serveru s bezpečnostními restrikcemi na přímé, vzdálené připojení.

## 6.6 Nasazení aplikace a instalace nezbytných součástí

Aplikace vyžaduje webový server s podporou skriptovacího jazyka PHP 8 a databázový systém MySQL 8. Nevyžaduje se konkrétní operační systém, aplikace by měla běžet bez omezení na různých platformách. Následující postup popisuje implementaci na linuxové operační systémy typu Debian.

Nejdříve aktualizujeme depozitář a systémové balíčky:

```
1. sudo apt update
2. sudo apt -y upgrade
```

Přidáme depozitář pro preprocessorové balíčky:

```
1. sudo add-apt-repository ppa:ondrej/php
```

Začleníme depozitář do systému:

```
1. sudo apt update
```

Nainstalujeme webový server Apache2:

```
1. sudo apt install apache2
2. sudo systemctl start apache2
```

Ujistíme se, že služba Apache2 bude spuštěna automaticky při startu systému:

```
1. sudo systemctl enable apache2
```

Přidáme podporu pro http/2:

```
1. sudo a2dismod mpm_prefork
2. sudo a2enmod mpm_event setenvif proxy_fcgi
3. sudo a2enmod http2
```

Nainstalujeme PHP a všechny přidružené knihovny:

```
1. sudo apt install php php-fpm php-mbstring php-cli php-bcmath php-json php-xml php-zip
   php-pdo php-common php-mysql php-tokenizer
2. sudo systemctl start php-fpm
3. sudo systemctl enable php-fpm
4. sudo a2dismod php
5. sudo a2enconf php-fpm
```

Připravíme si databázový server MySQL:

```
1. sudo apt install mysql-server
2. sudo systemctl start mysql.service
3. sudo systemctl enable mysql.service
```

Pro finální a plnou konfiguraci databáze zadáme:

```
1. sudo mysql_secure_installation
```

Jsme tázáni, jestli chceme zavést bezpečnostní komponentu, která zamezí vytvoření slabého hesla. Pokud zvolíme ano (y), skript nás nabádá k volbě úrovně politiky síly hesla. Nabízí tři varianty od nuly do dvou:

- **0** – Velikost hesla musí mít nejméně osm znaků.
- **1** – Kromě minimálně osmi znaků musí heslo obsahovat speciální znaky.
- **2** – K výše zmíněným pravidlům se přidává i slovníkový soubor.

Dle požadavků zvolíme požadovanou úroveň zabezpečení.

Skript zakončuje průvodce výzvou pro zadání hesla hlavního databázového uživatele „root“. Zvolíme heslo dle své potřeby.

Následně vytvoříme přístupový účet do databáze pro webovou aplikaci:

```
1. sudo mysql -u root -p
2. CREATE USER 'web_user'@'localhost' IDENTIFIED WITH authentication_plugin BY 'password';
3. CREATE DATABASE web_app;
4. GRANT ALL PRIVILEGES ON web_app.* TO 'web_user'@'localhost';
5. FLUSH PRIVILEGES;
6. QUIT;
```

Jméno účtu a heslo zvolte libovolně dle vlastních představ společně se jménem databáze. Do zvolené databáze spustíme dotazy (příloha B) na vytvoření potřebných tabulek:

```
1. mysql -u web_user -p web_app < ratetuber.sql
```

Nyní založíme samotnou složku pro soubory aplikace:

```
1. cd /var/www/html
2. mkdir web_app
```

Cestu k projektu si můžeme pojmenovat libovolně.

Přepneme se do složky aplikace, kam nakopírujeme příložený archiv (příloha A) a rozbálíme ho, přidáme podporu balíčkovacího systému Composer a nainstalujeme všechny závislosti:

```
1. cd web_app
2. tar -xzf ratetuber.tar.gz
3. curl -sS https://getcomposer.org/installer | php
4. sudo mv composer.phar /usr/local/bin/composer
5. sudo chmod +x /usr/local/bin/composer
6. composer dump-autoload
7. composer install
```

Pro složku projektu musíme přiřadit příslušná práva pro správný chod aplikace, aby webový server mohl správně fungovat a interpretovat soubory:

```
1. sudo chown -R www-data:www-data /var/www/html/web_app
2. sudo chmod -R 775 /var/www/html/web_app/storage
```

Nastavíme správné přístupové údaje do databáze v souboru „.env“ přes direktiva:

```
1. DB_CONNECTION=mysql
2. DB_HOST=localhost
3. DB_PORT=3306
4. DB_DATABASE=web_app
5. DB_USERNAME=web_user
6. DB_PASSWORD=password
```

Posledním krokem je přidání virtuálního hosta do webového serveru Apache2. Vytvoříme konfigurační soubor web\_app.conf následovně:

```
1. sudo nano /etc/apache2/sites-available/web_app.conf
```

Do souboru vložíme:

```
1. <VirtualHost *:80>
2. ServerName example.com
3. ServerAlias www.example.com
4. ServerAdmin admin@example.com
5. DocumentRoot /var/www/html/web_app/public
6. <Directory /var/www/html/web_app>
7. AllowOverride All
8. </Directory>
9. ErrorLog ${APACHE_LOG_DIR}/error.log
10. CustomLog ${APACHE_LOG_DIR}/access.log combined
11. Protocols h2 http/1.1
12. </VirtualHost>
```

a změníme „ServerName“ a „ServerAlias“ dle námi vlastněné domény a „ServerAdmin“ na kontakt administrátora.

Aplikujeme konfigurační soubor takto:

```
1. sudo a2enmod rewrite
2. sudo a2ensite web_app.conf
3. sudo systemctl restart apache2
```

Aplikace by již měla fungovat. Pro zabezpečení komunikace nám chybí ssl/tls vrstva. Pomocť nám může následující balíček:

```
1. sudo apt install certbot python3-certbot-apache
2. sudo certbot --apache
```

Spuštěný průvodce umožní získat SSL certifikát a přepnout virtuálního hosta na protokol HTTPS. Budeme vyzváni k zadání emailové adresy pro upozornění například konce platnosti certifikátu. Následně musíme odsouhlasit podmínky služby. Další fáze nám nabízí odebírat novinky organizace, které můžeme odmítnout. Nejdůležitější sekce přichází nyní, kde si vybereme ze seznamu naší nakonfigurovanou stránku detekovanou v nastavení Apache2. Poslední fáze se dotazuje na omezení užívání pouze zabezpečeného webu. Vybereme „2“ – přesměrování na adresu s certifikátem. Naše nastavení je kompletní. Důležité je dbát na

obnovování certifikátu, kde proces probíhá obdobně, popřípadě lze využívat funkci cron pro automatickou obnovu.

## 6.7 Testování a optimalizace aplikace

Do aplikace byly začleněny čtyři profily českých tvůrců videí na portálu YouTube. Pro vytvoření se využila lokálně implementovaná funkce pro automatickou tvorbu profilů pomocí YouTube Data API v3 přes administraci. Byla oslovena skupina mladých lidí, kteří měli za úkol relevantně vyzkoušet všechny případy užití a otestovat bezpečnost vstupů a stabilitu systému webové aplikace dostupné na: <https://app.tejkys.net/ratetuber/>. Několik dobrovolníků se zúčastnilo a splnilo požadovaná kritéria.

Pro lepší uživatelský zážitek bylo díky zpětné vazbě účastníků opraveno několik překlepů v zobrazujících se textových popiskách a zpřehledněno interpretování žebříčku na hlavní stránce pomocí přidání hodnot kritérií přímo do položky žebříčku.

Program OWASP ZAP 2.11.1 s více než dva tisíce požadavky potvrdil absenci kritických chyb a zranitelností jako: SQL Injection, Cross Site Scripting (XSS), Cross-Site Request Forgery (CSRF). Varoval před možností injektování rámců dokumentu HTML. Tento typ útoku byl následně ochráněn přidáním hlavičky „X-FRAME-OPTIONS "SAMEORIGIN"“ na straně serveru pro odpovědi na HTTP.



## ZÁVĚR

Hlavním cílem práce bylo vytvoření funkční webové aplikace pro zobrazování přehledných a aktualizovaných informací o vlivných tvůrcích publikující svoji tvorbu na platformě YouTube prostřednictvím profilů. Mezi důležité prvky patřila možnost hodnotit tvorbu a sdílet svůj názor s ostatními. Hlavní cíl práce byl splněn.

Bakalářská práce v teoretické části přiblížila detailněji průřez vývoje populární platformy YouTube a její nynější podobu způsobu využití. Zmíněna byla pro představu statistická data ohledně návštěvnosti a provozu. Další kapitola se zabírala problémem, jak sledovat informace a průběžné a dynamické statistiky ke konkrétním uživatelům s již existujícími nástroji. Pozornosti neunikla ani analýza sentimentu pro ohodnocování textů přes slovníkové výrazy. Jedna ze zvolených možností, Kijkwijzer, přiblížila detailnější způsob řešení ohodnocování médií, kterým se autor inspiroval v praktické části. Následoval obecný přehled technologií, kterých se realizace webové aplikace dotkla. Nebyl opomenut komunikační protokol http, skriptovací jazyk pro dynamicky generovaný obsah, zasazení PHP do vyšší struktury v podobě aplikačního rámce Laravel pro snadnější a strukturovanou práci na rozsáhlejších projektech s jednoduchou a přehlednou správou. Pro definici, vizualizaci a stylizaci elementů byly uvedeny jazyky HTML a CSS. Zmíněné technologie zastřešil JavaScript pro podporu dynamického chování na klientské straně a práci s DOM.

Praktická část se zaměřila na samotný návrh a vývoj webové aplikace. Byly stanoveny funkční a nefunkční požadavky, které jsou od aplikace vyžadovány. Další část představila databázový model ERD společně s modelem fyzickým, vztahy mezi entitami, kardinalitu a parcialitu, požadované atributy s datovými typy, integritní omezení a zaznamenávání změn. Microsoft Visio v další kapitole poskytl vizuální podobu diagramu případů užití v jazyce UML pro lepší pochopení funkcionality systému ve vztahu k aktérům. Jednotlivé případy užití byly pak detailněji specifikovány i s vizuální podobou. Autor uvedl i vývojová prostředí a nástroje pro tvorbu webové aplikace a převážně by chtěl vyzdvihnout chytré prostředí PhpStorm pro tvorbu webového obsahu. Díky pluginu pro příslušný aplikační rámec a profesionálnímu editoru se jednoduše píše aplikační kód, rychle formátuje soubor, našeptávač relevantně napomáhá a linkuje potřebné třídy. Další kapitola popisuje nasazení a instalaci na linuxových serverech s balíčkovacím systémem APT. Na závěr byla veřejná adresa webové aplikace poskytnuta dobrovolníkům, kteří si prostředí vyzkoušeli a penetračně otestovali. Zpětná vazba byla pozitivní jak z ovládacího hlediska, tak z hlediska vizuálního.

Webová aplikace poskytuje ještě velký prostor pro rozšíření a vylepšení funkcionalit. Příkladem mohou být předpřipravená sběrná data vhodná pro interpretaci časového vývoje sledovaných parametrů v podobě grafu. Taktéž zavedení emailového serveru by pomohlo se zabezpečením systému a rozšíření uživatelských funkcí například pro obnovení hesla. Požadavky na přidání profilu by mohli vytvořit samotní uživatelé a administrátoři by žádosti schvalovali, nebo by se proces zautomatizoval úplně. Aplikace by také mohla poskytovat více jazykových mutací pro pokrytí širšího spektra potenciálních uživatelů. Laravel poskytuje předpřipravený systém pro multijazyčnost. Filtrování žebříčku profilů by šlo rozšířit i o země a další. Originálním tvůrcům by se mohl nalinkovat příslušný profil po autorizaci a mohli by reagovat na názory fanoušků. Služba „Return YouTube Dislike“ by šla začlenit do aplikace pro relevantnější data. Nápadů je opravdu hodně a po delším provozu se nabídnou určitě další.

## POUŽITÁ LITERATURA

1. **ASLAM, Salman.** YouTube by the Numbers: Stats, Demographics & Fun Facts. *Omnicores Agency*. [online] 14. Únor 2022. [cit. 1. Březen 2022] Dostupné z: <https://www.omnicoreagency.com/youtube-statistics/>.
2. **PRATHAP, Dinesh Gowtham.** youtube api video category id list. *GitHub Gist*. [online] 1. Září 2016. [cit. 1. Březen 2022] Dostupné z: <https://gist.github.com/dgp/1b24bf2961521bd75d6c>.
3. **MCFADDEN, Christopher.** YouTube's History and Its Impact on the Internet. *Interesting Engineering*. [online] 3. Červenec 2020. [cit. 1. Březen 2022] Dostupné z: <https://interestingengineering.com/youtubes-history-and-its-impact-on-the-internet>.
4. YouTube in 2005. *Web Design Museum*. [online] © 2022. [cit. 1. Březen 2022] Dostupné z: <https://www.webdesignmuseum.org/gallery/youtube-2005>.
5. **HOLCOMBE, Jeremy.** What is Social Blade and What Can it Do for Your Small Business? *GreenGeeks*. [online] © 2008–2022. [cit. 1. Březen 2022] Dostupné z: <https://www.greengeeks.com/blog/social-blade-for-your-small-business/>.
6. **JAIN, Rajat.** Rajat Jain - Top Digital Marketing Consultant. *speakrj*. [online] © 2021. [cit. 1. Březen 2022] Dostupné z: <https://www.speakrj.com/>.
7. **MICHELLE.** OW MANY LANGUAGES ARE THERE IN THE WORLD? THE NUMBER MAY SURPRISE YOU (2021). *The Intrepid Guide*. [online] 1. Leden 2022. [cit. 1. Březen 2022] Dostupné z: <https://www.theintrepidguide.com/how-many-languages-are-there-in-the-world/>.
8. Sentiment Analysis: A Definitive Guide. *MonkeyLearn*. [online] © 2022. [cit. 1. Březen 2022] Dostupné z: <https://monkeylearn.com/sentiment-analysis/>.
9. Live demo Neticle text analysis. *Text Analysis API*. [online] © 2022. [cit. 1. Březen 2022] Dostupné z: <https://neticle.com/textanalysisapi/en/analyze-texts-with-neticle-Semantic-API>.

10. **VANĚK, Aleš.** Kijkwijzer - nizozemský systém pro klasifikaci obsahu elektronických médií. *Ikaros*. 2002, sv. ročník 6, č. 3. Dostupné z: <http://ikaros.cz/node/10882>. ISSN 1212-5075.
11. **KIJKWIJZER.** *Nederlandse Vereniging Van Bioscopen En Filmtheaters*. [online] © 2019. [cit. 1. Duben 2022] Dostupné z: <https://www.denvbf.nl/dossiers/kijkwijzer/>.
12. **GAMAGE, Thilina Ashen.** Evolution of HTTP — HTTP/0.9, HTTP/1.0, HTTP/1.1, Keep-Alive, Upgrade, and HTTPS. *Medium*. [online] 17. Listopad 2017. [cit. 1. Duben 2022] Dostupné z: <https://medium.com/platform-engineer/evolution-of-http-69cfe6531ba0>.
13. **FOX, Pamela.** Transmission Control Protocol (TCP). *Khan Academy*. [online] © 2022. [cit. 1. Duben 2022] Dostupné z: <https://cs.khanacademy.org/computing/informatika-pocitace-a-internet/x8887af37e7f1189a:internet/x8887af37e7f1189a:tcp-protokol/a/transmission-control-protocol--tcp>.
14. **POLLARD, Barry.** *HTTP/2 in Action*. New York: y Manning Publications Co. , 2019. ISBN: 9781617295164.
15. What is a TLS handshake? *Cloudflare*. [online] © 2022. [cit. 1. Duben 2022] Dostupné z: <https://www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/>.
16. **ČEGAN, Lukáš.** Protokol HTTP. *Moodle*. [online] 4. Leden 2022. [cit. 1. Duben 2022] Dostupné z: <https://moodle.upce.cz/moodle/mod/page/view.php?id=284737>.
17. History of PHP. *PHP*. [online] © 2001-2022. [cit. 1. Duben 2022] Dostupné z: <https://www.php.net/manual/en/history.php.php>.
18. **HOPKINS, Callum.** *PHP okamžitě*. Brno: Computer Press, 2014. ISBN 978-80-251-4196-0.
19. **NIXON, Robin.** *Learning PHP, MySQL & JavaScript*. Sebastopol: O'Reilly Media, Inc., 2018. ISBN: 978-1-491-97891-7.
20. **DANIELE, Carlo.** What's New in PHP 8 (Features, Improvements, and the JIT Compiler). *Kinsta*. [online] 30. Březen 2022. [cit. 1. Duben 2022] Dostupné z: <https://kinsta.com/blog/php-8/>.

21. PHP 8.0: JIT. *PHP.Watch*. [online] © 2018-2022. [cit. 1. Duben 2022] Dostupné z: <https://php.watch/versions/8.0/JIT>.
22. Laravel. *VIP Trust*. [online] © 2022. [cit. 1. Duben 2022] Dostupné z: <https://viptrust.com/technologie/ostatni/laravel>.
23. **Code, Chris on**. Simple Laravel CRUD with Resource Controllers. *DigitalOcean*. [online] 15. Zář 2020. [cit. 1. Duben 2022] Dostupné z: <https://www.digitalocean.com/community/tutorials/simple-laravel-crud-with-resource-controllers>.
24. **DOCKINS, Kelt**. *Design Patterns in PHP and Laravel*. New York: Apress Media, 2017. ISBN: 978-1-4842-2450-2.
25. **LAURENČÍK, Marek**. *Tvorba www stránek v HTML a CSS*. Praha: Grada Publishing, 2019. ISBN 978-80-271-2241-7.
26. **KOSEK, Jirka**. Historie a vývoj HTML. *htmlguru*. [online] © 2022. [cit. 1. Duben 2022] Dostupné z: <http://htmlguru.cz/uvod-historie.html>.
27. CSS: Cascading Style Sheets. *MDN Web Docs*. [online] 2. Květen 2022. [cit. 1. Květen 2022] Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
28. CSS Selector Reference. *W3Schools*. [online] © 1999-2022. [cit. 1. Duben 2022] Dostupné z: [https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp).
29. What Is MySQL And Why It Is Used? *Software Testing Help*. [online] 5. Květen 2022. [cit. 5. Květen 2022] Dostupné z: <https://www.softwaretestinghelp.com/what-is-mysql/>.
30. **JUBA, Salahaldin, VANNAHME, Achim a VOLKOV, Andrey**. *Learning PostgreSQL*. Packt Publishing Ltd.: Birmingham, 2015. ISBN 978-1-78398-918-8.
31. SQL | DDL, DQL, DML, DCL and TCL Commands. *GeeksForGeeks*. [online] 30. Zář 2021. [cit. 1. Duben 2022] Dostupné z: <https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/>.
32. **ČÁPKA, David**. Lekce 1 - Úvod do JavaScriptu. *ITnetwork.cz*. [online] 17. Duben 2013. [cit. 1. Duben 2022] Dostupné z: <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>.

## **PŘÍLOHY**

Příloha A – Archív zdrojových kódů webové aplikace	63
Příloha B – DDL SQL pro MySQL databázi	64

## **PŘÍLOHA A – ARCHÍV ZDROJOVÝCH KÓDŮ WEBOVÉ APLIKACE**

Příloha obsahuje zdrojové soubory webové aplikace v komprimovaném balíčku gzip v souboru `ratetuber.tar.gz`.

## **PŘÍLOHA B – DDL SQL PRO MYSQL DATABÁZI**

Příloha obsahuje sadu příkazů SQL pro vytvoření potřebných tabulek v souboru ratetuber.sql.