



Zadání bakalářské práce

Název:	Tvorba pluginu pro správu partnerů na webu
Student:	Šimon Melkes
Vedoucí:	Ing. Tomáš Vondra, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Webové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

CATVUSA je nezisková organizace propagující Českou Republiku v USA. Nadace se chystá spustit větší množství fundraisingových kampaní. K tomu potřebuje být schopná měnit informace o partnerech na svých stránkách. Loga a popisy partnerů potřebuje zadavatel dokázat jednoduše měnit v administraci webu a určit, na jakých místech se budou zobrazovat. Zobrazování by mělo být řešeno dynamicky pomocí prvků šablon systému Wordpress a data by měla být dostupná přes API pro mobilní a televizní aplikace.

1. Seznámit se s CMS Wordpress a tvorbou pluginů
2. Navrhnout ve spolupráci se zadavatelem práce administrační rozhraní pro správu partnerů
3. Implementovat plugin, který bude dynamicky a s proměnlivým pořadím zobrazovat bannery a popisy partnerských organizací ve všech vyhrazených prostorech na stránkách na základě jejich vztahu k obsahu
4. Navrhnout a implementovat API, které umožní partnery zobrazovat v mobilních aplikacích
5. Plugin i API řádně otestovat s využitím jednotkových testů a uživatelského testování funkčnosti obou částí
6. Vytvořit uživatelskou příručku k vytvořené práci v angličtině

Bakalářská práce

TVORBA PLUGINU PRO SPRÁVU PARTNERŮ NA WEBU

Šimon Melkes

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Tomáš Vondra, Ph.D.
16. května 2024

Chtěl bych poděkovat především svému vedoucímu Ing. Tomášovi Vondrovi, Ph.D. za skvělé vedení práce. Také bych chtěl poděkovat zadavateli Johnu Honnorovi za všechny konzultace ohledně práce a za možnost zapojení se do projektu CATVUSA. Také děkuji své rodině za podporu při psaní. V neposlední řadě děkuji i účastníkům uživatelského testování.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen "Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 16. května 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Šimon Melkes. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Melkes Šimon. *Tvorba pluginu pro správu partnerů na webu*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Abstrakt

Cílem této bakalářské práce je vytvořit plugin pro Wordpress pro neziskovou televizní stanici Czech-American TV, jejíž činností je propagace České republiky v USA. Nezisková organizace provozující televizní stanici potřebuje mít možnost jednoduchým způsobem spravovat zobrazování marketingových/fundraisingových kampaní a tím si zajistit prostředky pro své fungování. Hlavním přínosem práce je poskytnutí nástroje, pomocí kterého lze jednoduše přidávat obsah partnerů na web televizní stanice. V textu je popsána analýza požadavků uživatele, ze které se vycházelo při návrhu a implementaci pluginu.

Klíčová slova plugin, Wordpress, PHP, API, Czech-American TV, CATVUSA, správa partnerů, CMS, implementace, marketing

Abstract

The aim of this bachelor thesis is to create a Wordpress plugin for Czech-American TV, a non-profit television station whose activity is to promote the Czech Republic in the USA. The non-profit organization running the TV station needs to be able to manage the display of marketing/fundraising campaigns in a simple way and thus secure funds for its operation. The main contribution of the work is to provide a tool that can be used to easily add partner content to the TV station's website. An analysis of user requirements is described in the text, which was the basis for the design and implementation of the plugin.

Keywords plugin, Wordpress, PHP, API, Czech-American TV, CATVUSA, partners management, CMS, implementation, marketing

Obsah

Poděkování	i
Prohlášení	ii
Abstrakt	iv
Úvod	1
1 Cíle práce	2
2 Teorie	3
2.1 Webové technologie	3
2.2 Wordpress	4
2.3 API	9
3 Analýza	11
3.1 Představení webu	11
3.2 Partneři	12
3.3 Dřívější implementace	14
3.4 Aktuální stav	15
3.5 Požadovaný cíl	17
3.6 Specifikace požadavků	17
3.7 Model případů užití	18
4 Návrh	22
4.1 Navrhovaná funkcionalita	22
4.2 Principy	22
4.3 Navrhovaná architektura	23
5 Implementace	25
5.1 Architektura	25
5.2 Partner	28
5.3 Ukládání dat	32
5.4 Zobrazování partnerů	33
5.5 Integrace do stránky	35
5.6 API implementace	36
5.7 Jednotkové testování	37

Obsah	vi
5.8 Uživatelské testování	37
6 Diskuze	39
7 Závěr	40
Obsah přiloženého média	44
Přílohy	45
Seznam zkratk	57

Seznam obrázků

3.1	Homepage webu CATVUSA	11
3.2	Banner partnera	12
3.3	Náhledy na umístění partnerů využívající banner	13
3.4	Náhledy na umístění partnerů, kteří nevyužívají banneru	14
3.5	Sidebar s vloženým HTML	16
3.6	Template	17
3.7	Use case diagram	21
4.1	Schéma návrhu architektury pluginu	24
5.1	Formulář partnera	29
5.2	Relationship formulář	30
5.3	Formulář pro nahrání klipu	30
5.4	Partner story fields	31
5.5	Partner story s obsahem pro novou stránku	31
5.6	Partner story s odkazem	32
5.7	Sidebar uvnitř templaty	35
5.8	Vložení shortcodu do sidebaru	36
5.9	Registraci API endpointů	37
1	Broadcast partner	45
2	Post partner	46
3	Single-broadcast partner	46
4	Page partner	47
5	Slider partner	48
6	Our partner	49
7	Partner stories	49
8	Integrace do stránky	50
9	Ukázka změn provedených v souboru singular.php	51
10	Administrace v pluginu Smart slider	52
11	Skupina meta fields v nastavení pluginu ACF	53
12	Nastavení post meta field v pluginu ACF	54
13	Export skupiny meta fields v pluginu ACF	55

Seznam tabulek

5.1	Uživatelské testování	38
-----	---------------------------------	----

Seznam výpisů kódu

2.1	Funkce pro registraci post type	6
2.2	Funkce pro získávání metadat	6
2.3	Funkce pro ukládání metadat	7
2.4	Použití funkce registration_activation_hook	8
2.5	Zdrojový kód funkce registration_activation_hook	8
2.6	Volání shortcodeu	9
2.7	Volání shortcodeu s atributy	9
2.8	Registrace shortcodeu	9
5.1	Ukázka zavedení action hooku	26
5.2	Ukázka registrace shortcodeu v pluginu	26
5.3	Ukázka konfigurace DI	27
5.4	Skript s voláním funkce pro přidání videa před broadcast	33
5.5	Pomocná JS funkce pro přidání videa před broadcast	34
5.6	JS funkce pro přidání videa před broadcast	34
1	Jednotkový test testSavePartnerStory	55
2	Pomocná funkce pro jednotkové testy	56

Úvod

Wordpress je díky své flexibilitě a jednoduchosti používání nejpoužívanější systém na správu obsahu na webu. Pomocí řady hotových pluginů lze rychle přidat novou funkcionalitu. Prostřednictvím administračního panelu může správce webu jednotlivé moduly vypínat, měnit vzhled stránky, nebo provádět různé konfigurace, a to i bez znalosti kódu. Nové pluginy lze psát v PHP.

Czech-American TV (CATVUSA) je nezisková organizace propagující Českou republiku v USA. České tradice, kuchyni, památky a kulturu obecně divákům přibližuje pomocí videí, podcastů a článků. Cílem projektu je rozšiřovat povědomí o českém dědictví. Web organizace je postaven na platformě Wordpress, což otevírá dveře rozšiřování funkcionality pomocí pluginů.

Zadavatel potřebuje mít možnost jednoduše, rychle a bez zásahu do kódu přidat nové, nebo odebrat dříve přidané partnery projektu. Jelikož webu zatím chybí takové administrační rozhraní, projekt nemůže efektivně spolupracovat se zájemci. U jednotlivých partnerů je třeba mít možnost specifikovat, na jakých místech se bude zmínka v podobě banneru zobrazovat.

Cílem této bakalářské práce je vytvořit nástroj, který problémy s partnery pro zadavatele vyřeší a umožní organizaci jednoduše spravovat záznamy o partnerech a upravovat místa, kde se bude jejich obsah zobrazovat. Přínosem práce bude umožnění získávání finančních prostředků pro projekt, což zajistí pokračování činnosti webu a možnost dalšího rozvoje.



Kapitola 1

Cíle práce

Cílem této bakalářské práce je vytvoření Wordpress pluginu pro web CATVUSA, pomocí něhož bude možné spravovat partnery webu.

Teoretická část si klade za cíl analyzovat stávající řešení, které je na webu používáno. Dále provést analýzu potřeb uživatelů a zdefinovat funkční a nefunkční požadavky. Následně provést rešerši existujících nástrojů na tvorbu Wordpress pluginu.

Cílem praktické části je podle analýzy navrhnout architekturu pluginu a pomocí rešerše zvolit vhodné nástroje a způsoby na integraci do webu. Dále podle návrhu implementovat plugin tak, aby splňoval funkční i nefunkční požadavky. Praktická část si dále klade za cíl implementovaný plugin otestovat pomocí jednotkových testů a uživatelského testování. Posledním cílem praktické části je vytvoření uživatelské příručky v anglickém jazyce.

Kapitola 2

Teorie

V této kapitole se budu věnovat vybraným tématům, která jsou pro práci důležitá. Stručně popíši, jak některé technologie fungují, a jak je vývojáři při tvorbě aplikace používají. Cílem této části není vysvětlit celou problematiku, ale nezasvěcenému čtenáři alespoň část nastínit.

2.1 Webové technologie

Základním stavebním kamenem při vývoji webových stránek, nebo webových aplikací jsou technologie HTML, CSS a JavaScript (JS). [1]

2.1.1 HTML

HTML je značkovací jazyk, pomocí kterého je definován obsah a struktura webové stránky. Pomocí jednotlivých značek je jsou popsány části stránky a je jim přiřazen význam (nadpis, záhlaví). [2] Obsah HTML dokumentu je zpracován prohlížečem a následně je vykreslen. [3]

2.1.2 CSS

CSS (kaskádové styly) je jazyk využíván k popisu vzhledu HTML, nebo XML dokumentu. CSS popisuje, jak mají být jednotlivé elementy zobrazeny na různých typech médií (obrazovka, tisk). [4] Stejně jako HTML, i CSS je parsováno na straně klienta, tedy typicky ve webovém prohlížeči.

2.1.3 JavaScript

Jedná se o programovací jazyk, který umožňuje přidat komplexnější funkcionality na webové stránky. Jde o skriptovací interpretovaný jazyk, který je běžně používán nejen na webových stránkách na straně klienta, ale i na serverové straně. [5]

2.1.4 PHP

PHP je jedním z nejpoužívanějších programovacích jazyků používaných k tvorbě webových aplikací. Jde o tzv. server-side skriptovací jazyk, což znamená, že jeho kód je prováděn na serveru. Výsledek PHP skriptů bývá HTML kód, který je následně poslán klientovi. [6]

2.2 Wordpress

Wordpress je nejpoužívanějším CMS (Content Management System) - systém na správu obsahu na webu. [7] Jde o software, pomocí kterého lze jednoduše vytvořit web a následně spravovat obsah na něm. Hlavním přínosem CMS je, že na přidání nových stránek, nebo příspěvků není potřeba zasahovat do kódu a i člověk, který neumí vůbec programovat, může snadno přidávat obsah. [8]

2.2.1 Historie Wordpressu

Wordpress byl založen v roce 2003 jako fork projektu b2/cafelog. Zakladatelé, Mike Little and Matt Mullenweg, cítili potřebu pro vytvoření jednoduché redakčního systému, který by byl dostupný široké veřejnosti jak z uživatelského, tak cenového hlediska. Začínající uživatelé mohou systém užívat bez větších zásahů, zároveň existuje možnost, jak si funkcionalitu rozšířit. [9]

Mike a Matt se ztotožňovali s hodnotami původního projektu (b2/cafelog), bylo důležité, aby Wordpress byl jednoduchý, upravitelný a použitelný. Tyto myšlenky se tak staly základními prvky, které udávaly směr projektu v dalších letech. I proto je projekt open-source a je licencován pod GPLv2, tedy každý může tento software libovolně užívat, zkoumat, upravovat a redistribuovat. [9]

V roce 2004 byla pro blogování nejpoblíbenější platforma Movable Type, používalo ji až 70% blogů. Ze začátku byla zadarmo a zdrojové kódy byly volně dostupné, ale nebylo možné je upravené redistribuovat. V květnu roku 2004 však přišla nová verze, která uživatelům přinášela kromě nových funkcí i nové podmínky užívání a nový ceník. Dříve nejoblíbenější platforma zažila odliv uživatelů, kteří ve velkém přesunuli své blogy a stránky na Wordpress. [10]

V letech 2003-2005 prošel Wordpress rapidním vývojem a získal obrovskou popularitu díky jednoduchosti používání a flexibilitě, jakou měl. Komunita postupně přispívala k vývoji kódem a přibývaly nové funkce. Přidaly se části, díky kterým jsou podporovány témata vzhledu (themes), widgety, taxonomie, vlastní druhy příspěvků a mnoho dalšího. [11] Wordpress sice vznikl jako platforma pro blogování, vyvinul se ale v plnohodnotný systém na správu obsahu na webu. Nyní je používán přibližně na třetině webů dostupných na internetu a je volbou nejen jednotlivců, kteří chtějí prezentovat sebe, ale i velkých společností, kterým stojí část businessu právě na jejich webu. [12]

2.2.2 Theme

Při instalaci se vytvoří řada složek a stáhne řada souborů, které jsou pro fungování WP kritické. V kořenovém adresáři projektu se nachází některé důležité soubory, jako `.htaccess`, `wp-config.php`, `robots.txt` a složka `wp-content`. Při příchodím požadavku se Wordpress musí rozhodnout, jakou stránku zobrazit. Existuje mnoho komplikovaných pravidel, podle kterých se najde správný soubor a pomocí kterého se vygeneruje výsledné HTML. Zjednodušeně řečeno se nejprve hledají nejvíce specifické souboru a nakonec se při neúspěšném hledání přistoupí k souboru `index.php` v kořeni projektového adresáře. [13]

Je snaha mít oddělený obsah od prezentace, aby bylo možné jednoduše změnit celkový vzhled stránek. Proto jsou používány tzv. *themes*. Theme udává vzhled stránek, jejich rozložení, barvy a použité fonty, vždy je aktivní právě jeden theme. Lehce lze mezi jednotlivými nastaveními vzhledu přepínat (pokud je theme dobře vytvořený, přidané příspěvky zůstanou a změní se pouze způsob prezentace). Theme se skládá z řady komponent, včetně vzorů, CSS stylů, mediálních souborů (obrázky, videa), JavaScriptu a PHP souborů. [14]

Existují dva hlavní typy:

1. block theme - jsou postavené na blokových vzorech
2. classic theme - využívají PHP systém vzorů

Dále je tu ještě hybridní typ vzhledů - jde o klasický typ, který používá něco ze světa blokových vzhledů. [15]

2.2.2.1 Template

Hlavní částí, ze kterých se theme skládá, je *template*. Je to soubor, pomocí kterého lze definovat výsledný dokument, jež se zobrazí uživateli. Při navštívení určité stránky se Wordpress musí podle URL rozhodnout, jaký template načíst. Jakmile najde ten správný, zanalyzuje jeho blokový markup a vygeneruje výsledné HTML, které pošle zpět uživateli. [15]

2.2.2.2 Post type

Post type je obecně typ příspěvků, jež lze na Wordpressu vytvářet. Je to způsob, jak Wordpress uchovává textový a multimediální obsah. Wordpress nabízí následující typy příspěvků:

- post
- page
- attachment

- revision
- navigation menu
- custom CSS
- changeset

Wordpress také umožňuje vytvoření vlastních typů příspěvků. Při definování lze specifikovat různé možnosti, jako například existence vlastního archivu, jaký template pro zobrazování jednotlivých příspěvků se má používat, nebo jaké popisky (labels) má Wordpress používat v administračním rozhraní.¹[16] Registrace vlastního typu příspěvku probíhá pomocí funkce:

```
register_post_type(  
    string $post_type,  
    array|string $args = array()  
): WP_Post_Type|WP_Error
```

■ **Výpis kódu 2.1** Funkce pro registraci post type

2.2.2.3 Post meta

Post meta je způsob, jak přidat data k existujícímu příspěvku. Jsou to pojmenované záznamy navázané na konkrétní příspěvky. Meta fields jsou ukládána do samostatné tabulky *wp_postmeta* [17]. V textu používám termíny *meta fields* a *post meta*. Jde o totéž.

- Získávat data lze pomocí funkce

```
get_post_meta(  
    int $post_id,  
    string $key = "",  
    bool $single = false  
): mixed
```

■ **Výpis kódu 2.2** Funkce pro získávání metadat

- Ukládat data lze pomocí funkce

¹Všechny možnosti v dokumentaci: https://developer.wordpress.org/reference/functions/register_post_type/


```
add_post_meta(  
    int $post_id,  
    string $meta_key,  
    mixed $meta_value,  
    bool $unique = false  
): int|false
```

■ **Výpis kódu 2.3** Funkce pro ukládání metadat

2.2.2.4 Sidebar

Sidebar je pojmenovaná oblast, kterou poskytuje theme a do které lze vložit vlastní obsah. Vkládat lze widgety², text, obrázky, vyhledávací pole, obecně HTML kód. Vlastní sidebar lze zaregistrovat i v pluginu, nemusí ho dodat jen theme. Při registraci se mimo jiné definuje název, id, popis, a co má být vloženo před a za samotný obsah sidebaru. [18]

2.2.3 Plugin

Plugin je zásuvný modul, který rozšiřuje stávající funkcionalitu aplikace o další prvky. Samotná aplikace na pluginu není závislá a ke svému běhu ho nepotřebuje. Lze libovolně jednotlivé pluginy zapínat a vypínat. Jde o kus kódu, který je aplikací volán a tím zasahuje do běhu aplikace. Plugin v textovém editoru se například může starat o zvýraznění textu dle syntaxe v daném programovacím jazyce. Aby byla aplikace rozšiřitelná o pluginy, musí poskytnout rozhraní, pomocí kterého je možné plugin integrovat - způsob zaregistrování funkcí pluginu.

Pro Wordpress je dostupných mnoho bezplatných i placených pluginů, přičemž se dá i jednoduše napsat vlastní. Je snadné najít řešení, pomocí kterých lze budovat eshop³, zjednodušit integraci na sociální sítě⁴, nebo přidat optimalizaci výkonu⁵. Pluginy mohou přímo zasahovat do zobrazovaných stránek, nebo být pouze v administrační části, dostupné pro administrátora.

2.2.4 Hooks

Hook je způsob, jak pomocí kusu kódu ovlivnit chování jiné části kódu v předdefinovaných místech. Jde o základ toho, jak pluginy a theme interagují s jádrem Wordpressu. [19]

²<https://wordpress.org/documentation/article/manage-wordpress-widgets/>

³<https://wordpress.org/plugins/woocommerce/>

⁴<https://wordpress.org/plugins/ultimate-social-media-icons/>

⁵<https://wordpress.org/plugins/wp-optimize/>

Typicky vývojář napíše funkci, která má být volána při spuštění akce, například *init*, a následně ji zaregistruje. Když je pak jinde v kódu spuštěna akce pomocí *do_action()*, je dříve zaregistrovaný callback spuštěn.

Action je hook, který umožňuje přidávat, nebo měnit funkcionalitu. Nemá návratovou hodnotu a je registrován pomocí funkce *add_action()*. [19]

Filter je hook, který může měnit obsah, který je následně poslán uživateli. Má vstupní argument a má návratovou hodnotu. Jeho registrace probíhá zavoláním *add_filter()*. Měl by fungovat odděleně, a neměl by mít vedlejší účinky. [19]

2.2.4.1 Využití

Existují funkce na registraci callbacku pro určitou specifickou akci, například aktivace, či deaktivace konkrétního pluginu.

Activation hook Může se starat například o inicializaci potřebných tabulek v databázi.

```
register_activation_hook(
    __FILE__,
    'pluginprefix_function_to_run'
);
```

■ **Výpis kódu 2.4** Použití funkce *registration_activation_hook*

Zdrojový kód pro activation hook Při nahlédnutí na zdrojový kód takové funkce zjistíme, že využívá hook *action* se specificky pojmenovanou akcí.

```
function register_activation_hook( $file, $callback ) {
    $file = plugin_basename( $file );
    add_action( 'activate_' . $file, $callback );
}
```

■ **Výpis kódu 2.5** Zdrojový kód funkce *registration_activation_hook*

2.2.5 Shortcode

Shortcode je specifický kód pro Wordpress, který umožňuje dynamické vkládání obsahu na stránku. Shortcode na stránce je nahrazen voláním funkce, která je na shortcode navázána podle názvu shortcodu. [20] Volání shortcodu vypadá například následovně:

```
[spotify]
```

■ Výpis kódu 2.6 Volání shortcodeu

Je možné do funkce poslat i atributy. Volání shortcodeu s atributy:

```
[gallery id="123" size="medium"]
```

■ Výpis kódu 2.7 Volání shortcodeu s atributy

Wordpress nabízí již existující shortcode funkce, například: youtube, instagram, presentation, audio. Dále je možné si vlastní shortcode zaregistrovat pomocí funkce:

```
add_shortcode(  
    string $tag,  
    callable $callback  
)
```

■ Výpis kódu 2.8 Registrace shortcodeu

2.3 API

Application programming interface (API) je rozhraní, pomocí kterého mohou aplikace (libovolný software) díky definovaným protokolům komunikovat. Je to nástroj, jak umožnit automatickou výměnu dat mezi aplikacemi. [21]

2.3.1 REST API

Jako *Representational State Transfer API*, neboli REST API je API, které odpovídá principům REST architektury, mezi které patří:

- **Bezstavovost** - každý požadavek musí obsahovat veškeré nutné informace pro vytvoření odpovědi. Nesmí být závislost na session na straně serveru.
- **Uniformnost rozhraní** - je založeno na tzv. *resources*, neboli zdrojích. Každý zdroj dat musí mít výstižný název a dodržovat hierarchickou strukturu, například dotaz GET zaslaný na URI /person/7 musí získat osobu, s id = 7.

- **Klient - server architektura** - klient musí být oddělený a nezávislý na serveru. Jediné, co klient potřebuje musí být URI zdrojů.
- **Vrstvená architektura** - klient nemá být schopný rozlišit, zda komunikuje s proxy serverem, nebo přímo s hlavním serverem. Komunikace může procházet přes několik vrstev a API i aplikace s tím musí počítat.
- **Cachovatelnost** - pokud je to možné, zdroje by měly být cachovatelné. Informace, zda je povoleno zdroj cachovat, nebo ne. Odpověď server také musí obsahovat, po jaké době je nutné re-validovat cache.

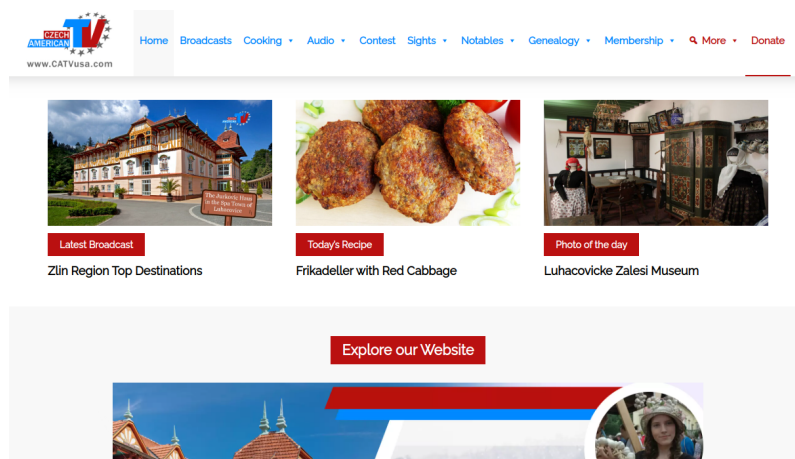
Kapitola 3

Analýza

Tato kapitola se věnuje analýze aktuálního stavu a fungování webu organizace CATVUSA. Popisuje problémy, které autor ve stávajícím řešení našel.

3.1 Představení webu

CATVUSA je nezisková organizace, která má za cíl čtenářům z celého světa představovat Českou republiku. Na svém webu 3.1 nabízí články o českých zvycích, tradicích, o české kuchyni, nebo také o českých památkách. Web nabízí také broadcasts, tedy videa, která se věnují stejným tématům, jako články. V neposlední řadě je na webu k nalezení audio obsah - rozhovory s různými hosty. Obsah na web je tvořen studenty ze Západočeské univerzity v Plzni (ZČU)¹.

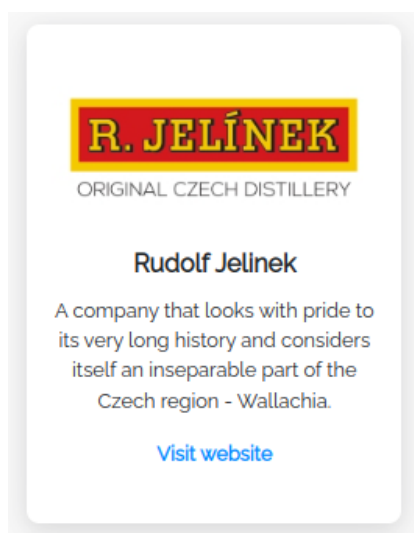


■ Obrázek 3.1 Homepage webu CATVUSA[22]

¹<https://www.zcu.cz>

3.2 Partneři

Je důležité si definovat, co se vlastně chápe pod pojmem *Partner*. Jde o subjekt z reálného světa (osoba, společnost), který navázal vztah s CATVUSA v podobě sponzoringu.



■ Obrázek 3.2 Banner partnera

3.2.1 Místa pro zobrazování

Místa pro zobrazování partnerů, ze kterých vychází následně navržené typy partnerů jsou:

Broadcasts partners na hlavní stránce

Page partners na stránkách příspěvků

Single broadcast partners na stránkách broadcastů

Page partners na stránkách všech speciálně vytvořených stránkách

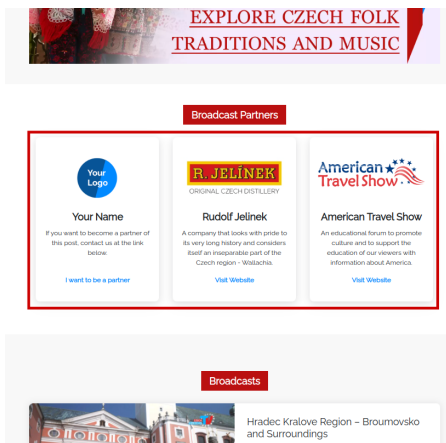
Our partners na speciální stránce, kde jsou uvedeni všichni partneři pod sebou

Slider na každé stránce těsně nad sekci *footer*, kde se automaticky posouvají obrázky partnerů

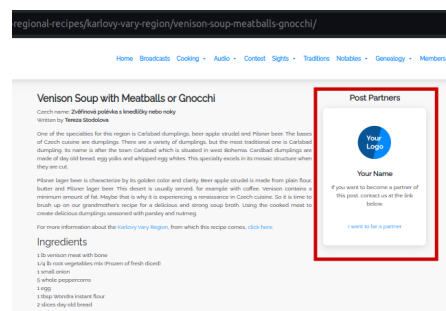
Partner stories na hlavní stránce v samostatné sekci.

V tuto chvíli nemá web žádné reálné partnery a obsah, který je prezentován jako promo partnerů, je uměle vytvořený. Následuje rychlý náhled na místa,

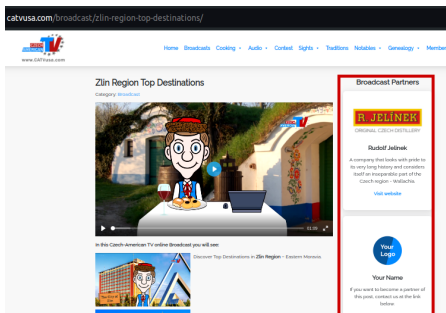
kde jsou partneři zobrazováni. Detailnější zobrazení je v samostatné sekci v přílohách.



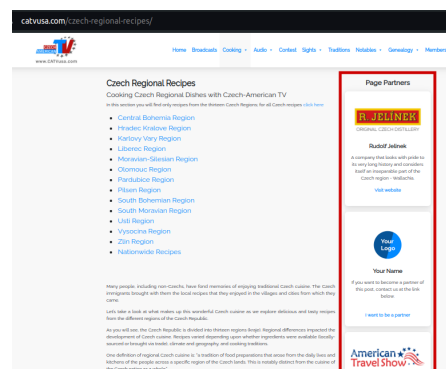
(a) Broadcast partner 1



(b) Post partner 2

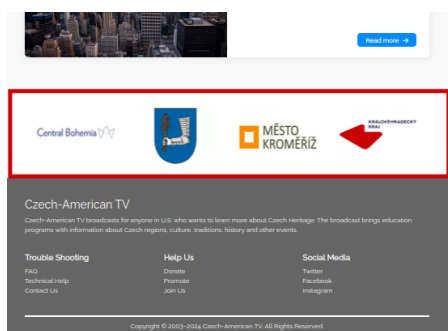


(c) Single-broadcast partner 3

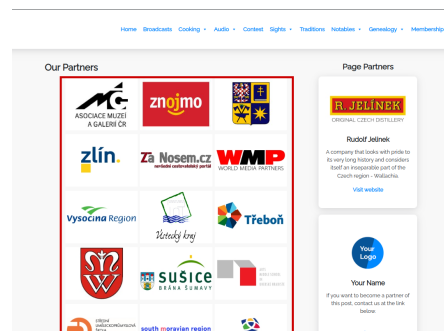


(d) Page partner 4

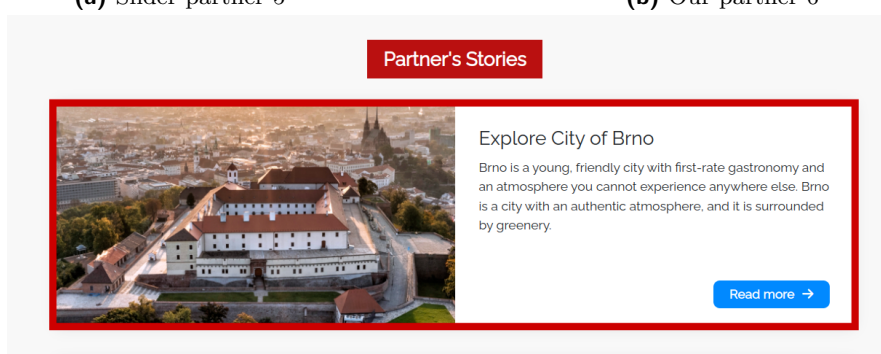
■ Obrázek 3.3 Náhledy na umístění partnerů využívající banner



(a) Slider partner 5



(b) Our partner 6



(c) Partner stories 7

■ **Obrázek 3.4** Náhledy na umístění partnerů, kteří nevyužívají banneru

3.3 Dřívější implementace

Tématu, kterému se věnuju tato bakalářská práce, se již o rok dříve věnovala bakalářská práce **Creating a plugin for managing partners on the web** [23] od Margarity Gorbatenko. Výsledný plugin ale nebyl z nasazen a téma bylo vypsané znovu. Problémem ve zmíněné práci byl princip, na kterém plugin fungoval a způsob, jakým bylo vkládání partnerského obsahu do stránky vyřešeno.

Autorka navrhla, aby její plugin fungoval následovně. Při vytvoření nového partnera s data uloží do databáze jako meta data (meta fields). Následně se do každé stránky, kde mají být partneři zobrazeni přidá HTML a stránka se uloží do databáze. Jelikož má být pořadí zobrazovaných partnerů proměnlivé, je nutné, aby každou hodinu proběhla akce nazvaná *shuffle*, při které se partneři na všech stránkách promíchají. [23] To vyžaduje:

1. najít všechny stránky, na kterých jsou partneři
2. na každé stránce najít všechny partnery
3. po jednom partnery odstranit (doufejme úspěšně)

4. opětovné vygenerování partnerů pro každou stránku
5. vložení správného HTML
6. uložení do databáze

[23]

V mém pluginu je způsob vkládání partnerského obsahu řešeno jiným způsobem, který bude popsán později. ⁵

3.4 Aktuální stav

V tuto dobu není správa partnerů na webu nijak sjednocená, na různých místech na webu jsou použity jiné nástroje.

3.4.1 Využívané pluginy

Na webu je využívána řada pluginů, přičemž některé byly vytvořeny na míru podle potřeb CATVUSA. Uvádím je pro uvedení do kontextu fungování tohoto webu (uvedené pluginy nemají na starost správu partnerů). Konkrétně jde o:

Genealogy tools Plugin pro vyhledávání předků. Dále se tímto pluginem nebudu zabývat. [24]

Radio Manager Plugin pro správu radio stanic. Dále se tímto pluginem nebudu zabývat. [25]

WPlyr - Better Video Player Plugin pro přehrávání videí pomocí přehrávače Plyr. K pluginu se později ještě vrátím při popisu 5.4.3 fungování zobrazování partnerského klipu před broadcastem. [26]

Z volně dostupných pluginů jsou využívány například:

Custom Sidebars Plugin, pomocí kterého lze vytvořit vlastní sidebar a spravovat jeho obsah. Pluginu se budu ještě věnovat později. ²

Smart Slider Plugin, pomocí kterého lze vytvořit vlastní slider a spravovat jeho obsah. Je zde vysoká možnost úpravy do vlastní podoby. Plugin ještě zmíním později. ³

²<https://wordpress.org/plugins/custom-sidebars/>

³<https://smartslicer3.com>

3.4.2 Integrace obsahu do stránky

Umístění bannerů je teď řešeno pomocí pluginu *Custom sidebar* 3.4.1. Sidebar je v templatě umístěn na vhodná místa. Následně je do sidebaru umístěn HTML kód, který reprezentuje banner partnera. 3.5

Partner stories na hlavní stránce jsou napevno vepsána v templatě 3.6 a související stránky byly k partnerům manuálně vytvořeny.

Některé broadcasty mají před sebou krátké promo video - to bylo vloženo jako součást broadcastu pomocí pluginu **WPlyr** 3.4.1, který přináší funkcionality přehrávače.

Partneři v liště nad sekci *footer* jsou spravováni pluginem **Smart slider** ⁴.



■ Obrázek 3.5 Custom sidebar[22]

3.4.3 Administrace

Přidávání, úprava, nebo odstranění jednotlivých partnerů probíhá skrze admin panel úpravou HTML, které je vloženo v sidebaru 3.5, nebo editací HTML v souborech 3.6 užívaného theme. Upravený obsah je statický. Každý zásah vyžaduje opatrnost při úpravě HTML kódu a editace je uživatelsky nepřívětivá. Administrátor, který nemá IT vzdělání, a ani není programátor, je nucen vyznat se v kódu. Pro přidání videa před broadcast je potřeba upravit samotný broadcast. Partneři zobrazovaní ve spodní části v nad sekci *footer* se dají upravit pomocí pluginu **Smart slider** 3.4.1, který má slider na starost. Ukázkou administrace v pluginu Smart slider najdete v přílohách 10

⁴<https://smartslider3.com/>

```
<div class="row">
  <div class="article-preview">
    <div class="article-image">
      <a target="blank" href="https://www.visitczechrepublic.com/en-US/">

    </div>
    <div class="article-info">
      <div class="article-info-text">
        <h4>Czech Tourism</h4>
        <p>Nowhere is too far for you to be able to set out for. Discover the historic
UNESCO monuments and also the fascinating atmosphere of ancient cities.
        </p>
      </div>
      <div class="article-info-button">
        <a target="blank" href="https://www.visitczechrepublic.com/en-US/">
          <button>Read more<i class="fas fa-arrow-right"></i></button>
        </a>
      </div>
    </div>
  </div>
</div>
```

■ Obrázek 3.6 Template[22]

3.5 Požadovaný cíl

Ideálním stavem by bylo, když by správa partnerů nebyla pro administrátora složitou operací, při které musí hledat, kde zapomněl vložit nějaký znak z HTML. Přidání, nebo úprava partnera by mělo být intuitivní a nenáročné. Celkově je tedy cílem zpříjemnit a zjednodušit management partnerů. V administracním rozhraní by měla existovat záložka, kde bude možné všechny partnery spravovat. Nemělo by být potřeba obcházet několik míst v nastavení a upravovat soubory manuálně.

3.6 Specifikace požadavků

V této části se věnuji požadavkům, které vzešly z několika schůzek se zadavatelem Johnem Honnozem.

3.6.1 Funkční požadavky

- F1 **Správa partnerů:** Administrátor může jednoduše spravovat partnery. Může nové přidávat, stávajícím upravovat detaily, nebo je mazat. Partner má jméno, zprávu, obrázek, a odkaz.
- F2 **Přiřazení typu partnera:** Administrátor může u každého partnera zvolit libovolný počet *typů partnerů* z předem definovaných možností. Podle typu

partnera a případně následně zvoleného příspěvku je partner zobrazován na jednotlivých stránkách.

F3 **Umístování:** Administrační rozhraní umožňuje:

- přidat banner partnera na konkrétní stránku
- přidat banner partnera ke konkrétnímu příspěvku
- přidat banner partnera ke konkrétnímu broadcastu
- přidat banner partnera do spodní lišty, která je na každé stránce
- přidat banner partnera do stránky *Our partners*
- přidat příběh partnera (partner stories) do samostatné sekce na hlavní stránce, společně s vytvořením samostatné stránky pro partnera
- přidat videoklip před konkrétní broadcast

F4 **Pořadí:** Bannery partnerů jsou zobrazovány v proměnlivém pořadí. Mění se s každým načtením stránky.

F5 **API:** Plugin má API, které může být využíváno mobilní aplikací a aplikací na chytrou televizi, které byly vytvořeny v dřívějších bakalářských pracích.

F6 **Zobrazování:** Návštěvníkovi stránek se podle zobrazené stránky správně ukáže partner, který má na zobrazovaný obsah vazbu (například je partnerem daného příspěvku).

3.6.2 Nefunkční požadavky

N1 **Nezávislost:** Web není závislý na pluginu, kdykoliv je možné ho deaktivovat, nebo úplně odstranit.

N2 **Intuitivní ovládání:** Pro správu není potřeba znalost HTML, ani složité zaškolení. Ovládání je prováděno v administračním panelu po přihlášení.

N3 **Izolace:** Plugin nezasahuje do jiných částí, pluginů, nebo theme

N4 **Rozšiřitelnost:** Plugin je do budoucna rozšiřitelný a má dokumentaci.

N5 **Jazyk:** Plugin je psaný v PHP 7.4

N6 **Manuál:** Pro plugin existuje manuál psaný v angličtině.

3.7 Model případů užití

Případ užití (use case), dále jen UC je posloupnost úkolů, nebo akcí, které uživatel provádí, aby dosáhl nějakého cíle, například vytvořil příspěvek. Pomocí jednotlivých UC lze definovat konkrétní funkcionality, kterými má aplikace disponovat. U každého UC je důležité, k jakému aktérovi se vztahuje. Celkově model případů užití popisuje, co se bude dít, ne jak se to uvnitř systému stane. [27]

3.7.1 Aktéři

Aktérem je objekt, který interaguje s aplikací. Nemusí nutně jít jen o uživatele, ale klidně i o jinou aplikaci. [27] V našem případě máme dva aktéry, konkrétně *administrátora* a *mobilní/televizní aplikace*.

Administrátor je přihlášený a má přístup do administračního rozhraní WordPressu. Má roli administrátora (ne jen editor...)

Mobilní/Televizní aplikace je aplikace vytvořená v předchozích bakalářských pracích. Pro komunikaci s webem CATVUSA využívá API.

3.7.2 Případy užití

Dále je výčet jednotlivých UC. Všechny UC jsou později zaneseny do diagramu v podobě obrázku. Administrátor provádí všechny úkony v administračním rozhraní, do kterého se dostane pouze přihlášený. Z toho důvodu nebudu uvádět, že se vždy musí přihlásit.

UC1: Vytvoření nového partnera: Může být provedeno pouze administrátorem

1. Administrátor klikne v levé části obrazovky na navigační panel a zvolí *Partners*
2. Zobrazí se list již existujících partnerů
3. Administrátor zvolí *Add new partner* v horní části obrazovky, dále vyplní informace o partnerovi a vše uloží.
4. Zobrazí se hláška o úspěšném uložení.

UC2: Úprava existujícího partnera: Může jít o aktivaci, deaktivaci, změnu jména, změnu obrázku, nebo kterékoliv jiné informace, která je k partnerovi ukládána.

1. Administrátor klikne v levé části obrazovky na navigační panel a zvolí *Partners*
2. Zobrazí se list již existujících partnerů, kde administrátor zvolí konkrétního partnera.
3. Dále upraví informace o partnerovi a vše uloží.
4. Zobrazí se hláška o úspěšném uložení.

UC3: Vytvoření partner story: Partner story je výrazně větší sekce pro obsah partnera. Je na hlavní stránce ve spodní části ve vlastní sekci. Na hlavní straně se zobrazí název, obrázek, delší text, a odkaz, který může vést buď na samotnou stránku vytvořenou pro partnera, nebo libovolnou vloženou URL.

1. Administrátor klikne v levé části obrazovky na navigační panel a zvolí *Partners*
2. Zobrazí se list již existujících partnerů, kde administrátor zvolí konkrétního partnera.
3. Zvolí typ partnera „Partner stories“
4. Vybere, jestli má jít pouze o odkaz, nebo se má vytvořit i nová vlastní stránka.
5. Vyplní odkaz, nebo obsah pro novou stránku a vše uloží.
6. Zobrazí se hláška o úspěšném uložení.

UC4: Vytvoření partner broadcast klipu: Pokud je u partnera zvolen typ *Single broadcast partner*, tedy partner konkrétního broadcastu, může si zvolit, zda chce mít u broadcastu i klip. Partner může mít zvoleno více broadcastů. U všech je stejný klip.

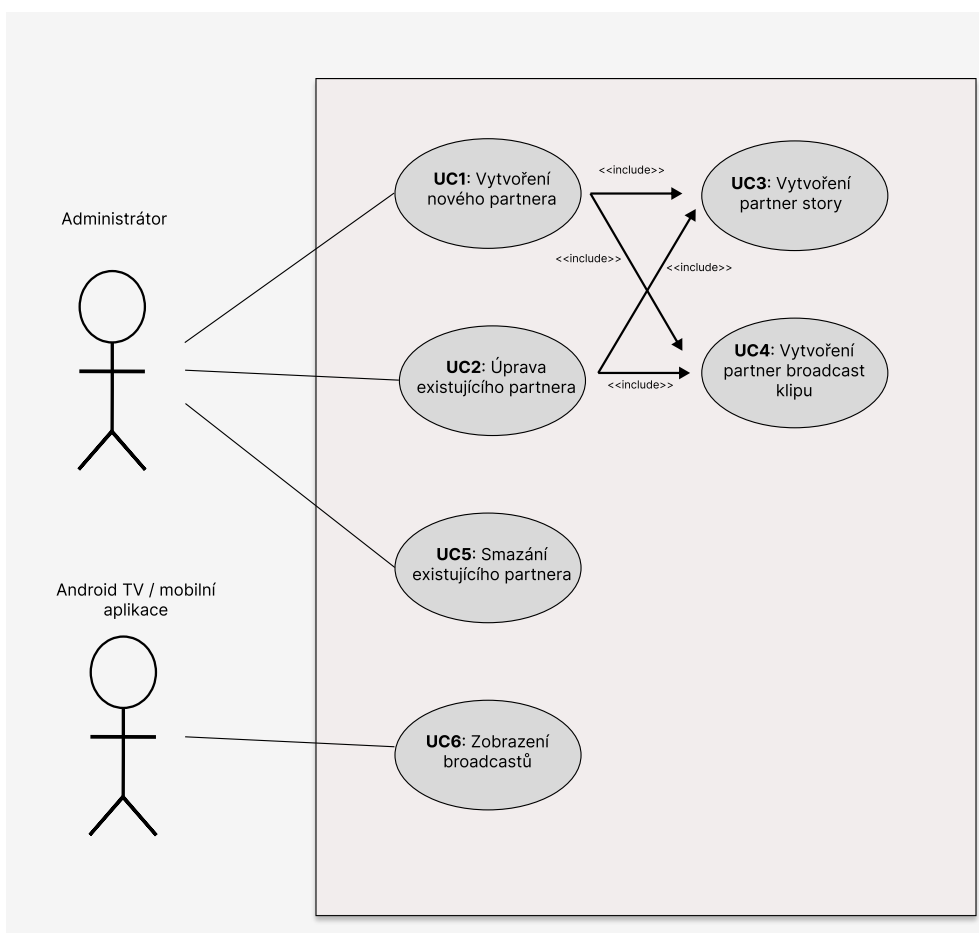
1. Administrátor klikne v levé části obrazovky na navigační panel a zvolí *Partners*
2. Zobrazí se list již existujících partnerů, kde administrátor zvolí konkrétního partnera.
3. Zvolí typ partnera „Single broadcast partner“
4. Vybere možnost, že partner má mít před broadcastem klip.
5. Nahraje nové video, nebo vybere z videí, která již byla nahrána do Wordpress media knihovny. Administrátor vše uloží.
6. Zobrazí se hláška o úspěšném uložení.

UC5: Smazání existujícího partnera: Partner je mazán stejným způsobem, jako by byl jakýkoliv jiný post type.

1. Administrátor klikne v levé části obrazovky na navigační panel a zvolí *Partners*
2. U konkrétního partnera zvolí možnost *Trash*.
3. Zvolí typ partnera „Single broadcast partner“
4. Zobrazí se hláška o úspěšném přesunutí do koše.

UC6: Zobrazení broadcastu v mobilní/televizní aplikaci: Televizní a mobilní aplikace mohou zobrazovat broadcasty, i jednotlivé příspěvky.

1. Mobilní/Televizní aplikace pošle požadavek na získání partnerů pro konkrétní broadcast.
2. REST API posílá v odpovědi požadovaný obsah partnery konkrétního broadcastu ve formátu JSON.



■ Obrázek 3.7 Use case diagram

Kapitola 4

Návrh

Tato kapitola se věnuje popisu mého návrhu řešení výše popsaných problémů. Popisuje, co návrh řeší a rozebírá principy fungování výsledného díla. V textu používám termíny aplikace a plugin. V obou případech se jedná o plugin, který si tato bakalářská práce stanovila za cíl vytvořit. Jelikož se jedná o Wordpress plugin, který bude nasazen do již existujícího a fungujícího webu, je nutné vymezit, do jakých částí bude plugin zasahovat a co všechno bude umět.

4.1 Navrhovaná funkcionality

Plugin poskytne administrační rozhraní pro vytváření a editaci partnerů. Umožní k jednotlivým partnerům přidávat vazby na příspěvky, stránky, broadcasty a vybírat, zda mají být partneři zobrazeni na stránce *Our partners* nebo ve spodní liště. Dále umožní přidávat *Partner stories* a přidávat klipy před broadcasty. Dále bude vytvořeno API, které bude moci být využito v mobilní a televizní aplikaci, nebo libovolnou, v budoucnu využitou aplikací, nebo pluginem. API bude poskytovat přístup k datům souvisejícím s partnery.

Plugin **nebude** umět pracovat s radiostanicemi, které jsou na webu dostupné.

4.2 Principy

Při návrhu řešení jsem se řídil několika principy, které jsem považoval pro plugin za důležité.

4.2.0.1 Neinvazivnost

Plugin nesmí po instalaci a aktivaci nic sám změnit, nebo ovlivnit. Nesmí se například stát, že plugin sám změní kus šablony, který využívá theme. Plugin musí pouze nabídnout určitou funkcionality, která může a nemusí být na webu

využita. Když se administrátor webu CATVUSA rozhodne funkcionalitu využít, mělo by to být jednoduché. Je předpoklad, že funkcionalita bude využita celá, ale když by z nějakého důvodu bylo potřeba kus deaktivovat, měla byt ta možnost existovat.

4.2.0.2 Modularita

Na předchozí princip navazuje princip modularity, který říká, že kód by měl být rozdělen do modulů, které jednotlivé funkcionality přidávají. Funkcionalitou je myšleno například zobrazování slideru na spodní straně stránek, nebo zobrazování klipů před broadcastem. Moduly by mělo jít aktivovat, či deaktivovat jednoduše, tím je myšleno v kódu, například zakomentováním jednoho řádku.

4.2.0.3 Přehlednost kódu

Návrh architektury pluginu by měl být čistý, intuitivní a blízký návrhu webových aplikací. Při případné budoucí úpravě by programátor měl přirozeně vědět, kam do kódu sáhnout.

4.3 Navrhovaná architektura

Návrh architektury aplikace je obecně důležitým aspektem pro dobrou udržovatelnost a rozšiřitelnost do budoucna. Po zvážení všech aspektů, které jsou uvedeny v kapitole *Analýza 3* a v předchozí sekci *Principy 4.2* navrhují následující architekturu.

Plugin bude složen ze tří hlavních částí: *Moduly*, *App* a *Api*.

Moduly - budou odpovědné za obsluhu činností jako je registrace callbacků pomocí dříve zmiňovaného nástroje - *hook 2.2.4*, registrace nových shortcodů, nebo registrace vlastních meta fields. Moduly se budou starat o to, aby byla dostupná funkcionalita druhé části pluginu - *App*. Všechny moduly budou spravovány jednou třídou, která bude mít na starost jejich instalaci.

App - bude jádrem funkcionality celého pluginu. Půjde o třídy, které budou mít na starost získávání dat z databáze, manipulaci s daty a následné sestavení HTML, které je vloženo na správné místo na stránce. Tato část bude rozdělena do tří vrstev:

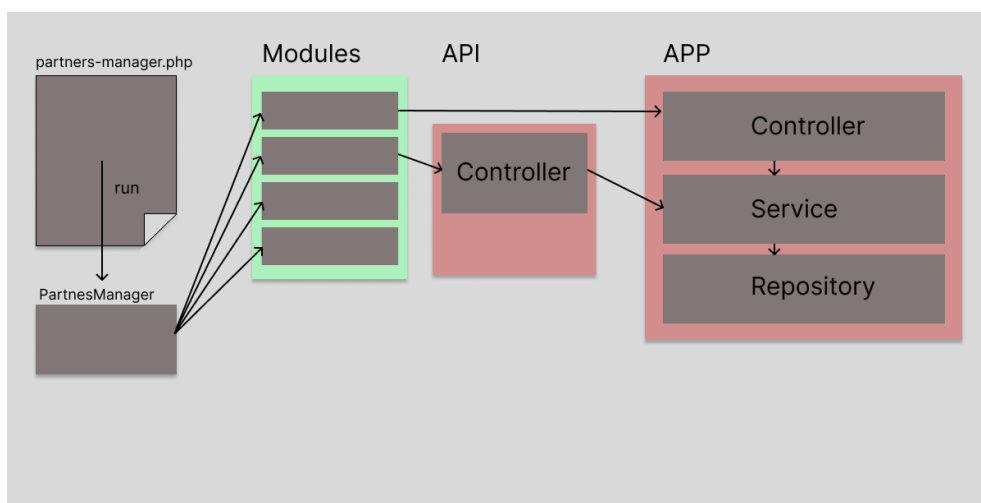
1. Controller
2. Service
3. Repository

Dále bude mít pomocné třídy:

1. Entity
2. View

Api - bude se jednat o REST API, nabízející pouze endpointy s GET metodami. Plugin nemá poskytovat možnost upravovat, vytvářet, nebo mazat partnery, ale pouze je podle typů získávat. Api bude využívat funkcionality části *App*.

Celý plugin by tak šel zakreslit do následujícího schématu.



■ **Obrázek 4.1** Schéma návrhu architektury pluginu

Implementace

V kapitole Implementace se věnuji detailnímu popisu výsledného pluginu. Je zde popsáno jaké technologie jsem se rozhodl při implementaci použít a proč tomu tak bylo. Popisuji architekturu celé aplikace, jaká část je za co odpovědná a jak spolu komponenty interagují. Dále vysvětluji, jak probíhá administrace partnerů a jak vše funguje v pozadí.

5.1 Architektura

Jak bylo navrženo v kapitole *Navrhovaná architektura* 4.3, plugin má část *Modules*, ve které jsou třídy zprostředkovávající určité napojení na Wordpress, například *SidebardModule* zaregistruje sidebary a shortcody společně s metodami z jiných tříd, které budou shortcody volány. Plugin obsahuje následující moduly:

- ApiModule
- CustomFieldModule
- PartnerBroadcastContentModule
- PartnerStoryModule
- PostTypeModule
- SidebarModule
- SliderModule

Například *Sidebar module* využívá action hook *widgets_init*:

```
add_action(  
    'widgets_init',  
    [ $this, 'registerCustomShortcode'  
]);
```

■ Výpis kódu 5.1 Ukázka zavedení action hooku

Následně je ve správný moment (když Wordpress provede akci *widgets_init*) zavolána funkce, jež stará o zaregistrování shortcodeu s funkcí z controlleru, která vrací HTML s partnery pro konkrétní příspěvek.

```
public function registerCustomShortcode() : void {  
    add_shortcode(  
        PARTNERS_MANAGER_SHORTCODE_POST_PARTNERS,  
        [$this->partnerController, 'getPostPartners']  
    );  
}
```

■ Výpis kódu 5.2 Ukázka registrace shortcodeu v pluginu

Architektura části *App* se drží navrhovaného rozdělení do vrstev Controller, Service a Repository. Pro nezávislost na implementaci jednotlivých částí je využito interfaců a dependency injection (DI) skrze konstruktor.

5.1.1 Dependency injection

Pro DI je využito PHP-DI kontejneru ¹, který registruje implementace jednotlivých rozhraní a následně je instancuje, když jsou potřeba v konstruktoru jiné třídy. PHP-DI je snadno použitelný kontejner využívaný frameworky, jako je Symfony ², nebo Slim ³. DI kontejner se stará o vytváření a správu instancí tříd. Bývá k tomu využíváno anotací, atributů, nebo definicí v PHP kódu. Když je potřeba vytvořit instanci třídy, kontejner zjistí, zda už má instanci dané třídy vytvořenou. Pokud ano, vrací existující instanci, pokud ne, zjistí kontejner, jaké má třída závislosti a následně rekurzivně řeší jednotlivé závislosti. Díky využití interfaců a DI, je možné odděleně testovat jednotlivé třídy a také jednoduše vyměnit implementaci konkrétního rozhraní.

Tento plugin využívá konfiguraci pomocí PHP. Přesněji řečeno ke každému interfacu definuje implementaci, která se má následně používat při instancio-

¹<https://php-di.org/>

²<https://symfony.com/>

³<https://www.slimframework.com/>

vání tříd. Konfigurace se nachází v souboru *config.php* a je použita při vytváření kontejneru.

```
return [  
    IPartnerRepository::class =>  
        DI\autowire(PartnerRepository::class),  
]
```

■ **Výpis kódu 5.3** Ukázka konfigurace DI

5.1.2 Controller

Controller v třívrstvé webové aplikaci je běžně odpovědný za obsluhu požadavků související s konkrétním typem entity, například *Person*, nebo *Room*. Jednotlivé metody controlleru bývají navázány na konkrétní endpoint a mají na starost vygenerování odpovědi. Odkud data pro odpověď controller získá, je jeho starostí. Rozhoduje také, co vše se v odpovědi pošle.

V tomto případě je controller navázán na shortcode, tedy shortcode volá metody controlleru. Po zavolání metody controlleru je návratová hodnota z metody vložena na místo shortcodu do stránky, která je vrácena uživateli. Třídy *Controller* jsou tak odpovědné za vložení dat do HTML. Vybírají, jaký *View* se použije a případně do jakého *WrapperView* se data zabalí. Využívají tříd *Service*.

5.1.3 Service

Service je v tzv. *business* vrstvě a bývá odpovědná za implementaci podnikových rozhodnutí.

Zde jsou tři třídy service: *PartnerService*, *PartnerStoryService* a *PartnerBroadcastContentService*. První má na starost pouze předání dat z nižší vrstvy v budoucnu zde může být přidána určitá logika. *PartnerStoryService* se stará o vytváření, či mazání stránky partnera.

5.1.4 Repository

Je v *data* vrstvě a je odpovědná za získávání a ukládání dat. Jednotlivé implementace třídy repository mohou pracovat s různými zdroji dat - jedna (příklad) může získávat data z databáze, jiná *PersonRepository* může získávat data z filesystemu z jednoho souboru, kde jsou všechna data serializována.

Repositories v tomto pluginu využívají funkcí Wordpressu, jako třeba *get_posts(\$args)*, nebo *get_post_meta(\$post_id)*, pro získávání příspěvků a metadat. V tomto pluginu existují:

- **PartnerRepository** - má na starost pouze získávání partnerů podle typu a id příspěvku, na který se vážou (post, page, broadcast), protože o ukládání dat se stará plugin ACF 5.3.1 (detailněji vysvětleno později).
- **PartnerStoryRepository** - stará se o získávání, ukládání a mazání partner stories.
- **PartnerBroadcastContentRepository** - stará se o získávání videí, která jsou vkládána před broadcasty.

5.2 Partner

Ke každému partnerovi jsou ukládány základní informace, které jsou potřeba pro vytvoření banneru. Jde o *název*, *priority*, *obrázek*, *zprávu* a *odkaz*. Priorita umožňuje administrátorovi určit, v jakém pořadí budou partneři zobrazováni. Tam, kde jsou partneři naschvál zobrazováni v náhodném pořadí je respektována priorita. Prvně jsou zobrazeni partneři s největší prioritou, kteří jsou navzájem promícháni, dále jsou zobrazeni partneři nižší priority, kteří jsou také navzájem promícháni, a tak to pokračuje až po nejnižší prioritu.

Dále jsou ke každému partnerovi ukládány jeho typy. Podle typů jsou ukládány přídatné informace a také je tím určeno, kde bude partner zobrazen.

Administrace partnerů probíhá následujícím způsobem. Administrátor vytváří jednotlivé partnery, u kterých zadá potřebné informace, zvolí jejich typ a doplní informace specifické k jednotlivým typům (například obsah stránky pro *Partner stories*). Vytvořené partnery může upravovat, nebo mazat. Vytváření a úprava konkrétního partnera probíhá skrze formulář 5.1. Administrátorovi se při zvolení konkrétního typu partnera zobrazí pole, která s daným typem souvisí.

Add New Partner

Enter Partner Name

Partner fields

Partner priority
Determines how important is the partner. Bigger number means more important.
0

Partner message *
Displays in banner. Use no more than 120 characters.

Partner image *
Displays in all partner image locations. Ideal aspect ratio is 13/8.
No image selected [Add Image](#)

Partner link *
Link to partner site. Is used in banner.

Partner type

- Post partner
- Page partner
- Single broadcast partner
- Broadcast partner
- Partner stories
- Slider partner
- Our partner

Publish

[Save Draft](#)

Status: **Draft** [Edit](#)

Visibility: **Public** [Edit](#)

[Publish Immediately](#) [Edit](#)

[Publish](#)

■ **Obrázek 5.1** Formulář partnera

Typy partnerů odpovídají místům, na kterých jsou partneři zobrazováni, jak bylo zmíněno v kapitole *Analýza 3.2.1*. Je možné vybrat libovolný počet z následujících typů:

Post partner - je potřeba zvolit, u kterých příspěvků bude zobrazován. Mezi příspěvky patří například recepty, památky, nebo jednotliví umělci. U typů *Post partner*, *Page partner* a *Single-broadcast partner* administrátor vybírá, u jakého příspěvku, stránky, respektive broadcastu bude partner zobrazen. Výběr probíhá skrze následující formulářový prvek. 5.2

Single broadcast

Search...

<p>About Czech Heritage and Landmarks</p> <p>About Czech Schools in the USA and UNESCO World Heritage Sites in the Czech Republic</p> <p>Central Bohemia – Destinations to Explore</p> <p>Chronicle of the Great Moravia</p> <p>Czech Christmas Traditions</p>	<p>Introduction</p> <p>Central Bohemia – Destinations to Explore</p>
--	--

■ **Obrázek 5.2** Relationship formulář

Page partner - je potřeba vybrat, na kterých stránkách bude zobrazován. Stránky, na kterých lze partnery zobrazovat je dán template, který stránka využívá. Aby mohl být partner zobrazen na stránce, je potřeba, aby stránka využívala template *singular.php*, nebo *custom-page-class.php*.

Single-broadcast partner - je nutné vybrat broadcasty, u kterých má být partner zobrazen. Dále je možnost přidat video, které bude před zvolenými broadcasty přehráno. Výběr videa probíhá skrze následující formulářový prvek.

Single broadcast partner type

Without content before broadcast

With video before broadcast

Pre broadcast clip

No file selected

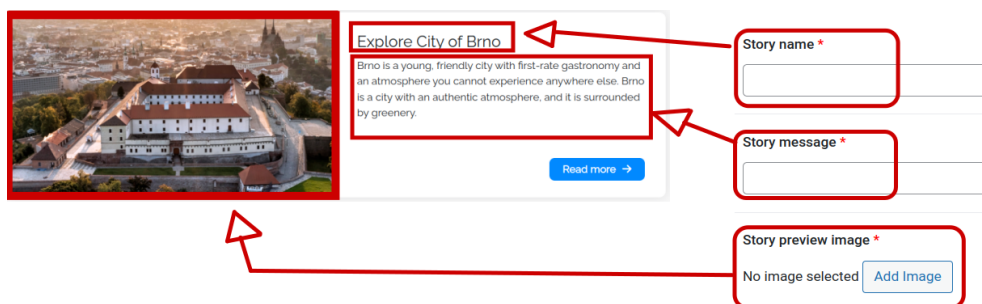
■ **Obrázek 5.3** Formulář pro nahrání klipu

Broadcast partner - není potřeba přidávat žádné další informace, partner je zobrazen na hlavní stránce v sekci *Broadcast partners*.

Our partner - není potřeba přidávat žádné další informace, partner je zobrazen na speciální stránce *Our partners*.

Slider partner - není potřeba přidávat žádné další informace, partner je zobrazen na každé stránce ve spodní liště, hned nad sekci *footer*.

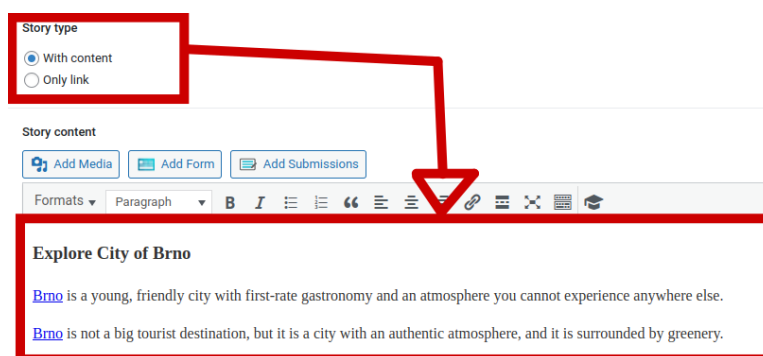
Partner stories - je nutné vyplnit *nadpis*, *náhledovou zprávu*, *náhledový obrázek* a *typ*. Je na výběr ze dvou typů, přičemž náhledová karta vypadá pro oba typy stejně. Obrázek 5.4 ukazuje, jak jsou mapovány první tři zmíněná políčka na kartu příběhu. Odkaz v prvku *Read more* na náhledové kartě je vyplněn podle typu partner story.



■ **Obrázek 5.4** Partner story fields

Typy pro *partner story* jsou konkrétně:


- **With content** - po vyplnění obsahu stránky pomocí WYSIWYG ⁴ editoru je vytvořena nová stránka. Odkaz na nově vzniklou stránku je při zobrazování vkládán do náhledové karty příběhu 7.



■ **Obrázek 5.5** Partner story s obsahem pro novou stránku

⁴<https://www.advancedcustomfields.com/resources/wysiwyg-editor/>

- **Only link** - náhledová karta obsahuje uložený link.



The image shows a form interface for creating a 'Partner story'. It features two main sections: 'Story type' and 'Story link *'. In the 'Story type' section, there are two radio buttons: 'With content' (unselected) and 'Only link' (selected). A red box highlights the 'Only link' option, and a red arrow points from it to the 'Story link' field. The 'Story link' field contains the URL 'https://www.visitczechia.com/en-us' and is also highlighted with a red box.

■ **Obrázek 5.6** Partner story s odkazem

5.3 Ukládání dat

Jelikož téměř všechno je ve Wordpressu příspěvek (post), je vhodné i Partnera ukládat jako příspěvek. Vytvořil jsem proto nový post type 2.2.2.2, *Partner*. K partnerům je potřeba ukládat výše zmíněná data 5.2. Rozhodl jsem se je ukládat jako post meta 2.2.2.3.

Jedna možnost byla vytvořit formulář a logiku zpracování formuláře vlastnoručně, druhá byla využít existující nástroj, který by ukládání dat obstaral. Zjistil jsem, že na webu CATVUSA je plugin, jež by takovou funkcionalitu přinášel, již používán. Jde o plugin **Advanced Custom Fields (ACF)**.⁵

5.3.1 ACF

Plugin nabízí možnost vytvořit skupinu meta fields, které jsou přiřazeny konkrétním typům příspěvků (post type) 2.2.2.2 a při vytváření, nebo úpravě takového příspěvku se automaticky nabídnou políčka pro úpravu nastavených meta fields. ACF se stará o vytvoření formuláře, o jeho zpracování, validaci a následné uložení dat. Každý meta field má název, podle kterého je následně možné data z databáze získat pomocí Wordpress funkce `get_post_meta(int $post_id)`.⁶

Definice meta fields v ACF pluginu probíhá skrze ACF UI. Uživatel vytvoří skupinu meta polí 11 a u každého jednotlivě zvolí jeho typ, název, a různá omezení (třeba, jestli je pole povinné, nebo kdy se má zobrazovat) 12. Skupinu pojmenuje a přiřadí konkrétnímu typu příspěvku (post type), například *Partnerovi*. V posledním kroku je skupina meta fields vyexportována 13 do PHP a vložena do kódu pluginu na vhodné místo. V mém pluginu se registraci ACF polí stará *CustomFieldModule*.

Výhodou využití pluginu ACF 5.3.1 je konzistentní vzhled formuláře s ostatními typy příspěvků a hotová, již otestovaná implementaci práce s formuláři.

⁵<https://www.advancedcustomfields.com/>

⁶https://developer.wordpress.org/reference/functions/get_post_meta/

Potenciální nevýhodou je omezená možnost ovlivnit funkčnost formuláře. ACF nabízí možnost ovlivnit například položky, které se nabízí při výběru polí typu *relationship*, tedy polí, kde se vybírá konkrétní jiný příspěvek, stránka, nebo broadcast. Pomocí filtru `acf/fields/relationship/query/key=<field_id>` lze filtrovat argumenty AJAX dotazu, pomocí kterého jsou získávány položky pro výběr ve formuláři.

5.4 Zobrazování partnerů

5.4.1 Banner

Banner partnera, který je zobrazovaný pro typy `page partner`, `post partner`, a `single-broadcast partner` je jednoduchá „karta“ 3.2, která obsahuje obrázek, název partnera, krátký text a odkaz na web partnera. Tento vzhled jsem se rozhodl z webu CATVUSA převzít a neměnit ho, protože byl vytvořen autorem, který navrhoval celý theme.

5.4.2 Story

Stejně jako *banner*, i *story preview* kartu 7, která je k vidění na hlavní stránce webu v samostatné sekci, jsem ze stejného důvodu převzal již hotovou a vzhled jsem neměnil.

5.4.3 Klip

Zobrazení klipu před broadcastem je závislé na konkrétní implementaci přehrávače, který se stará o zobrazení broadcastu. Nyní je používán plugin WPLYR 3.4.1. Plugin si připraví videa do fronty - objekt `window.videoSourceMap` drží fronty videí pro jednotlivé přehrávače, které se na stránce objevují. Můj plugin těchto front využije a svá videa vloží před videa, která ve frontách již jsou. Konkrétně `PartnerBroadcastContentController`, který má na starost klipy před broadcasty, vrací JS volání funkce, která přidává videa na začátek front.

Do stránky je pak vložen například následující skript:

```
<script>
  attemptAddingClip(
    'https://www.testwebsite.catvusa.com/wp-content/uploads/
    ↪ s/2024/05/Promo_WEB-Czech.mp4'
  );
</script>
```

■ **Výpis kódu 5.4** Skript s voláním funkce pro přidání videa před broadcast

Jelikož *PartnerBroadcastContentModule* se postaral o přidání souboru se skripty, je JS funkce `attemptAddingClip` dostupná a pokud je objekt `window.videoSourceMap` již definovaný (pluginem WPlyr), tak přidá video na začátek fronty. Pokud by se stalo, že fronty videí ještě nejsou definovány (skript s voláním funkce *attemptAddingClip* byl v HTML dokumentu dříve, než skript s definováním objektu `window.videoSourceMap`), bude zaregistrován *event listener*. Níže ukázka funkcí, které se starají o přidání videa na začátek fronty.

```
function partnerManagerAttemptAddingClip(clipPath){

    if(window.videoSourceMap === undefined){

        document.addEventListener("load", ()=>{
            partnerManagerAddClipFront(clipPath)
        },1);
    }else{
        partnerManagerAddClipFront(clipPath)
    }
}
```

■ **Výpis kódu 5.5** Funkce `partnerManagerAttemptAddingClip`

```
function partnerManagerAddClipFront(clipPath){

    let type = 'video';
    let argsMap = {
        source: clipPath,
        type: type,
    }

    for (let key in window.videoSourceMap) {
        if (window.videoSourceMap.hasOwnProperty(key)) {
            window.videoSourceMap[key].unshift(argsMap);
        }
    }
}
```

■ **Výpis kódu 5.6** Funkce `partnerManagerAddClipFront`

5.5 Integrace do stránky

O integraci do stránky by se dala rozdělit na dvě části. První je nabídnutí funkcionality, druhou je využití funkcionality na správných místech.

Funcionalitu nabízí controllery 5.1.2, které obstarají nutná data a poskytnou výsledné HTML. Controller je například schopen získat všechny partnery typu *Our partner* a data zabalené do odpovídajícího HTML vrátit.

Umístění a správné místo na stránku je řešeno s ohledem na možnost úpravy. Potřeboval jsem, aby administrátor měl definovaná místa, kde chce partnery zobrazovat, a zároveň, aby měl čím daná místa naplnit. Důležité bylo, aby existoval jednoduchý způsob, jak na všechny místa, kde se zobrazují například *Post partners*, šlo jednoduše vložit jiné řešení, než poskytuje tento plugin.

Rozhodl jsem se pro výše zmíněné důvody zaregistrovat sidebary, které budou v theme vloženy na správná místa - tím se definují lokace pro zobrazování, a následně zaregistrovat shortcody, pomocí kterých budou volány funkce controllerů. O zmíněnou registraci se stará *SidebarModule*.

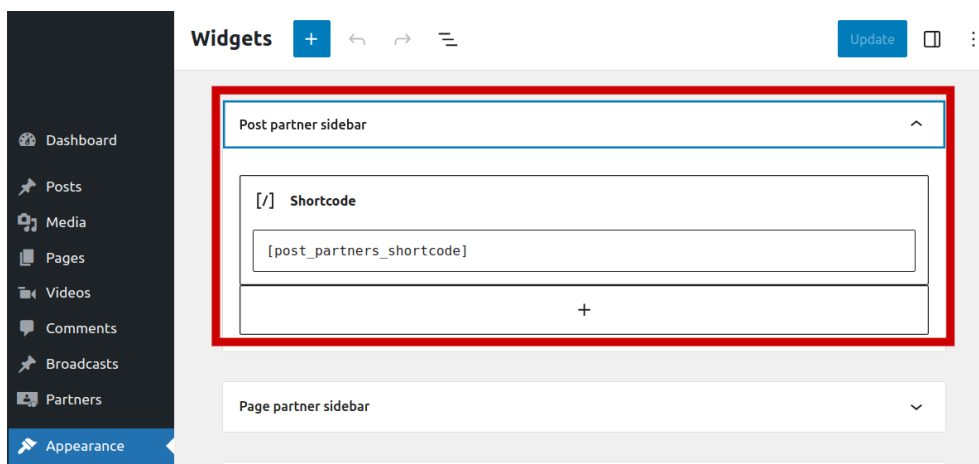
V theme bylo nutné udělat různé změny. Všechny jsou zaneseny v manuálu. Ukázka, jak vypadá volání sidebaru v dokument *singular.php* je znázorněna na následujícím obrázku:

```
if( defined('PARTNERS_MANAGER_SIDEBAR_PAGE_PARTNERS')) {  
    dynamic_sidebar(PARTNERS_MANAGER_SIDEBAR_PAGE_PARTNERS);  
}
```

■ Obrázek 5.7 Sidebar uvnitř templatu

Větší ukázkou změn provedených v dokumentu *singular.php* najdete v přílohách, viz 9

Po instalaci a aktivaci pluginu je nutné vložit správné shortcody do sidebaru. V manuálu je popsáno, jaký text (název shortcodu) je přesně potřeba vložit na jaké místo. Na následujícím obrázku je vidět, jak vypadá vložení shortcodu do sidebaru.



■ Obrázek 5.8 Vložení shortcodu do sidebaru

Spolupráci sidebaru, shortcodu a funkce controlleru, by šlo znázornit obrázkem 8

5.6 API implementace

API mělo za cíl nabídnout možnosti pro získávání partnerů podle typů stejně, jako jsou zobrazováni na webu. Api si nekladlo za cíl nabídnout možnosti pro vytváření, úpravu, nebo mazání partnerů. Nabízí proto 7 endpointů, na kterých lze pomocí metody *GET* získat partnery. API má endpointy:

[/post/id/partners](#)

[/page/id/partners](#)

[/single-broadcast/id/partners](#)

[/broadcast/partners](#)

[/our/partners](#)

[/slider/partners](#)

[/stories](#)

5.6.1 Registrace endpointů

Registrace endpointů probíhá v *ApiModule* třídě pomocí Wordpress funkce *register_rest_route*. Je nutné specifikovat namespace, aby bylo možné oddělit endpointy jednotlivých pluginů. Když by například tento plugin zaregistroval endpoint */partner* a jiný plugin by taktéž zaregistroval stejný endpoint, nebylo by jasné, jaký mám mít přednost a došlo by ke kolizi. Z toho důvodu je

využíváno tzv. *namespace*. Obecně jde o textové pojmenování skupiny souvisejících prvků, v tomto případě jde o způsob sdružení endpointů jednoho pluginu a především oddělení od jiných pluginů.

Třída `ApiModule` registruje endpointy způsobem, jak je ukázáno na obrázku:

```
$version = '1';
$namespace = 'partners-manager/v' . $version;

register_rest_route( $namespace , $endpoint->getRoute(), [
    'methods' => 'GET',
    'callback' => $endpoint->getCallback(),
    'permission_callback' => '__return_true',
]);
```

■ **Obrázek 5.9** Registraci API endpointů

Wordpress sám nabízí REST API, pomocí kterého je možné získávat příspěvky, komentáře, nastavení, nahrané media soubory atd. [28]. Wordpress REST API je dostupné v namespace `/wp/v2` a REST API je celkově dostupné na `/wp-json`. Když chce někdo chce získat všechny příspěvky na webu CATVUSA, pošle GET požadavek na `https://catvusa.com/wp-json/wp/v2/posts`.

API tohoto pluginu je umístěno do namespace `partners-manager/v1`. Aby aplikace získala všechny partnery typu *Slider* na testovacím webu, poslala by GET požadavek na `https://testwebsite.catvusa.com/wp-json/partners-manager/v1/slider/partners`.

5.7 Jednotkové testování

Část pluginu, konkrétně třídy `PartnerService`, `PartnerBroadcastContentService` a `PartnerStoryService`, je pokryta jednotkovými testy, u kterých bylo využito DI pro mockování závislostí. Ukázka jednotkového testu je k nalezení v příloze 1.

5.8 Uživatelské testování

Proběhlo uživatelské testování, které mělo za cíl odhalit potencionální nedostatky pluginu. Testování probíhalo se zadavatelem Johnem Honnoem a s 3 dalšími testery. Pouze zadavatel znal web a věděl, co tam jak funguje a kde se mají partneři zobrazovat. Pro ostatní testery bylo tímto faktem testování zhoršeno. Nejprve například nevěděli, co vůbec je příspěvek na webu, nebo co jsou

broadcasty. Po krátké ukázce, co vše se na stránkách CATVUSA nachází následovalo testování. Tester dostal manuál a měl pouze s manuálem vyzkoušet přidat nového partnera a vyzkoušet nastavit mu různé typy. Následně se měl pokusit najít, co vlastně přidal a kde je zásah na stránkách vidět.

Po otestování jsem se s testery bavil a ptal se, co jim nevyhovovalo, a co by změnili. Předložil jsem každému z nich několik kritérií, které měli ohodnotit na škále 1-10, přičemž 1 je nejhorší hodnocení a 10 nejlepší. Následující tabulka zachycuje jejich odpovědi.

Kritéria použitelnosti	Hodnocení			
	Tester 1	Tester 2	Tester 3	Tester 4
Přehlednost umístění jednotlivých ovládacích prvků	10	8	8	9
Přehlednost průchodu scénářem	9	10	9	8
Jednoduchost použití	10	10	9	9
Rozsah funkcionality	8	10	10	8
Nečekané prvky	8	8	9	9

■ **Tabulka 5.1** Uživatelské testování

Uživatelské testování mi pomohlo odhalit některé věci, které jsem sám přehlédl, nebo jsem je považoval za přehledné.



Kapitola 6

Diskuze

Plugin je zcela funkční a může být nasazen do provozu, jsou zde ale případná rozšíření, která by mohla přijít vhod.

Do budoucna by bylo určitě vhodné vytvořit komplexnější testy, nejen jednotkové, ale například i integrační a také testy pokrýt výrazně větší část pluginu. Dále se plugin dá rozšiřovat o možnost přidávat videa nejen před broadcast, ale i za něj. Zároveň by bylo vhodné přidat možnost upravit pořadí zobrazovaných klipů před broadcastem. Jelikož se aktuálně dá u partnera nastavit priorita, nebude problém tuto funkcionalitu doimplementovat.

Se zadavatelem je domluveno pokračování na implementaci některých ze zmíněných funkcí i po skončení bakalářské práce.



Kapitola 7

Závěr

Hlavním cílem této práce bylo navrhnout a implementovat plugin na správu partnerů na webu CATVUSA společně s API, které umožní partnery zobrazovat i v mobilní / Android TV aplikaci. Dalším cílem bylo vytvořit k pluginu manuál v angličtině a veškerou funkcionalitu řádně otestovat.

Výsledkem práce je plugin, nabízející administrativní rozhraní, kde lze jednoduše nové partnery vytvářet, či stávající upravovat. Plugin je navržen tak, aby byl neinvazivní a jeho funkcionalita šla kdykoliv, bez zásahu do kódu, omezit. Dosaženo toho je pomocí způsobu integrace do webu, konkrétně využití sidebar a shortcode funkcionalit Wordpressu. Když se administrátor rozhodne, že partnery již nechce na některém místě vůbec zobrazovat, může sekci jednoduše vypnout, či nahradit úplně jiným řešením. Api poskytuje 7 endpointů, pomocí kterých lze získat partnery různých typů.

Proběhlo uživatelské testování aplikace zadavatelem a několika dalšími testery. Po otestování jim byl předložen formulář pro ohodnocení použitelnosti aplikace. Testování Api probíhalo manuálně pomocí nástroje Postman. Část aplikace je pokryta jednotkovými testy.

Všechny cíle, které si práce kladla, byly splněny.

Bibliografie

1. FREECODECAMP. *Learn Web Development Basics – HTML, CSS, and JavaScript Explained for Beginners* [online]. 2021-08-10. [cit. 2024-04-21]. Dostupné z: <https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners/>.
2. W3SCHOOLS. *HTML Introduction* [online]. 2024. [cit. 2024-04-21]. Dostupné z: https://www.w3schools.com/html/html_intro.asp.
3. DEV. *How web browsers work - parsing the HTML (part 3, with illustrations)* [online]. 2022-04-25. [cit. 2024-04-21]. Dostupné z: <https://dev.to/arikaturika/how-web-browsers-work-parsing-the-html-part-3-with-illustrations-45fi>.
4. MDN WEB DOCS. *CSS: Cascading Style Sheets* [online]. 2024. [cit. 2024-04-21]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/CSS>.
5. MDN WEB DOCS. *JavaScript* [online]. Aktual. 2024-03-05. [cit. 2024-05-07]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
6. THE PHP GROUP. *What is PHP?* [online]. 2024. [cit. 2024-04-21]. Dostupné z: <https://www.php.net/manual/en/intro-what-is.php>.
7. WPBEGINNER. *2024's CMS Market Share Report – Latest Trends and Usage Stats* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://www.wpbeginner.com/research/cms-market-share-report-latest-trends-and-usage-stats/>.
8. KINSTA INC. *What Is a Content Management System (CMS)?* [online]. Aktual. 2024-04-19. [cit. 2024-04-20]. Dostupné z: <https://kinsta.com/knowledgebase/content-management-system/>.
9. WORDPRESS. *About* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://wordpress.org/about/>.

10. WORDPRESS. *Freedom Zero* [online]. Aktual. 2015-12-1. [cit. 2024-04-20]. Dostupné z: <https://wordpress.org/book/2015/11/freedom-zero/>.
11. WPMARMITE. *WordPress History: The evolution of the world's #1 CMS* [online]. 2024. [cit. 2024-03-25]. Dostupné z: <https://wpmarmite.com/en/wordpress/history/>.
12. WEBKAT. *Co je třeba vědět o tom, co umí WordPress* [online]. Aktual. 2022-10-22. [cit. 2024-03-30]. Dostupné z: <https://www.webkat.cz/co-je-treba-vedet-o-tom-co-umi-wordpress/>.
13. HOSTINGER. *WordPress Architecture: Understanding the Website Structure, Common Files, and Directories* [online]. Aktual. 2024-3-27. [cit. 2024-04-20]. Dostupné z: <https://www.hostinger.com/tutorials/wordpress-architecture>.
14. WORDPRESS. *Theme Development* [online]. 2024. [cit. 2024-03-19]. Dostupné z: https://codex.wordpress.org/Theme_Development.
15. WORDPRESS. *What Is a Theme?* [online]. Aktual. 2023-12-14. [cit. 2024-03-30]. Dostupné z: <https://developer.wordpress.org/themes/getting-started/what-is-a-theme/>.
16. WORDPRESS. *What is Post Type?* [online]. Aktual. 2023-01-13. [cit. 2024-04-21]. Dostupné z: <https://wordpress.org/documentation/article/what-is-post-type/>.
17. WORDPRESS DOWNLOAD MANAGER. *What Is Post Meta In WordPress?* Aktual. 2023-03-31. Dostupné také z: <https://www.wpdownloadmanager.com/post-meta-in-wordpress/>.
18. WORDPRESS. *Sidebars* [online]. 2024. [cit. 2024-04-18]. Dostupné z: <https://codex.wordpress.org/Sidebars>.
19. WORDPRESS. *Hooks* [online]. Aktual. 2024-01-29. [cit. 2024-03-19]. Dostupné z: <https://developer.wordpress.org/plugins/hooks/>.
20. WORDPRESS. *Shortcode API* [online]. 2024. [cit. 2024-03-19]. Dostupné z: <https://wordpress.com/support/wordpress-editor/blocks/shortcode-block/>.
21. AMAZON. *What is an API (Application Programming Interface)?* [online]. 2024. [cit. 2024-04-22]. Dostupné z: <https://aws.amazon.com/what-is/api/>.
22. CZECH-AMERICAN TV. *About Czech-American TV* [online]. 2024. [cit. 2024-04-20]. Dostupné z: <https://catvusa.com/about-czech-american-tv/>.
23. GORBATENKO, Margarita. *Creating a plugin for managing partners on the web*. Praha, 2023. bakalářská práce. České vysoké učení technické v Praze. Dostupné z <http://hdl.handle.net/10467/108596>.

24. ŠANTORA, Jakub. *Wordpress plugin pro vyhledávání předků*. Plzeň, 2019. bakalářská práce. Západočeská univerzita v Plzni. Dostupné z <http://hdl.handle.net/11025/38293>.
25. VRABEC, Karel. *Vývoj nové správy rádiových stanic pro Czech-American TV*. Praha, 2021. bakalářská práce. České vysoké učení technické v Praze. Dostupné z <http://hdl.handle.net/10467/95565>.
26. VÁVERKA, Jakub. *Wordpress plugin pro snadné přehrávání videa na webu*. Plzeň, 2020. diplomová práce. Západočeská univerzita v Plzni. Dostupné z <http://hdl.handle.net/11025/41763>.
27. ITNETWORK. *Lekce 2 - UML - Use Case Diagram* [online]. Aktual. 2022-08-04. [cit. 2024-04-22]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>.
28. WORDPRESS. *Reference* [online]. Aktual. 2024-01-16. [cit. 2024-05-11]. Dostupné z: <https://developer.wordpress.org/rest-api/reference/>.




Obsah příloženého média

	readme.txt	stručný popis obsahu média
	src	
	partners-manager	zdrojové kódy implementace
	tv_updated	upravený theme
	text	text práce
	melkesim_catvusa_bp.pdf	text práce ve formátu PDF
	CATVUSA_partners_manager_manual.pdf	manuál ve formátu PDF
	changelog.txt	textová příloha k manuálu - changelog pro theme

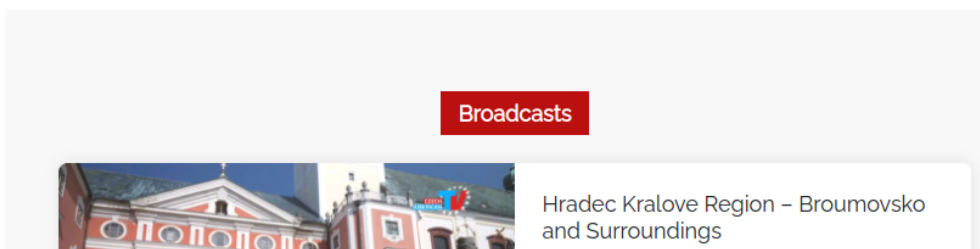
Přílohy



Broadcast Partners

 <p>Your Name</p> <p>If you want to become a partner of this post, contact us at the link below.</p> <p>I want to be a partner</p>	 <p>Rudolf Jelinek</p> <p>A company that looks with pride to its very long history and considers itself an inseparable part of the Czech region - Wallachia.</p> <p>Visit Website</p>	 <p>American Travel Show</p> <p>An educational forum to promote culture and to support the education of our viewers with information about America.</p> <p>Visit Website</p>
---	--	--

Broadcasts



■ Obrázek 1 Broadcast partner

regional-recipes/karlovy-vary-region/venison-soup-meatballs-gnocchi/

Home Broadcasts Cooking • Audio • Contest Sights • Traditions Notables • Genealogy • Membership

Venison Soup with Meatballs or Gnocchi

Czech name: Zvěřinová polévka s knedličky nebo noky
Written by Tereza Stodolova

One of the specialties for this region is Carlsbad dumplings, beer apple strudel and Pilsner beer. The bases of Czech cuisine are dumplings. There are a variety of dumplings, but the most traditional one is Carlsbad dumpling. Its name is after the town Carlsbad which is situated in west Bohemia. Carlsbad dumplings are made of day old bread, egg yolks and whipped egg whites. This speciality excels in its mosaic structure when they are cut.

Pilsner lager beer is characterize by its golden color and clarity. Beer apple strudel is made from plain flour, butter and Pilsner lager beer. This desert is usually served, for example with coffee. Venison contains a minimum amount of fat. Maybe that is why it is experiencing a renaissance in Czech cuisine. So it is time to brush up on our grandmother's recipe for a delicious and strong soup broth. Using the cooked meat to create delicious dumplings seasoned with parsley and nutmeg.


For more information about the [Karlovy Vary Region](#), from which this recipe comes, [click here](#).

Ingredients

- 1 lb venison meat with bone
- 1/4 lb root vegetables mix (Frozen or fresh diced)
- 1 small onion
- 5 whole peppercorns
- 1 egg
- 1 tbsp Wondra instant flour
- 2 slices day old bread

narodnu

Post Partners




Your Name

if you want to become a partner of this post, contact us at the link below.

[I want to be a partner](#)

■ Obrázek 2 Post partner


catvusa.com/broadcast/zlin-region-top-destinations/

 www.CATVusa.com


Home Broadcasts Cooking • Audio • Contest Sights • Traditions Notables • Genealogy • Member

Zlin Region Top Destinations

Category: Broadcast



In this Czech-American TV online Broadcast you will see:



Discover Top Destinations in Zlin Region - Eastern Moravia.

The Buchlovina Chateau

Broadcast Partners


R. JELÍNEK

ORIGINAL CZECH DISTILLERY

Rudolf Jelínek

A company that looks with pride to its very long history and considers itself an inseparable part of the Czech region - Wallachia.

[Visit website](#)




Your Name

if you want to become a partner of this post, contact us at the link below.

■ Obrázek 3 Single-broadcast partner

catvusa.com/czech-regional-recipes/

 www.CATVusa.com

Home Broadcasts **Cooking** Audio Contest Sights Traditions Notables Genealogy Membership

Czech Regional Recipes

Cooking Czech Regional Dishes with Czech-American TV

In this section you will find only recipes from the thirteen Czech Regions; for all Czech recipes [click here](#)

- [Central Bohemia Region](#)
- [Hradec Kralove Region](#)
- [Karlovy Vary Region](#)
- [Liberec Region](#)
- [Moravian-Silesian Region](#)
- [Olomouc Region](#)
- [Pardubice Region](#)
- [Pilsen Region](#)
- [South Bohemian Region](#)
- [South Moravian Region](#)
- [Usti Region](#)
- [Vysocina Region](#)
- [Zlin Region](#)
- [Nationwide Recipes](#)


Many people, including non-Czechs, have fond memories of enjoying traditional Czech cuisine. The Czech immigrants brought with them the local recipes that they enjoyed in the villages and cities from which they came.

Let's take a look at what makes up this wonderful Czech cuisine as we explore delicious and tasty recipes from the different regions of the Czech Republic.

As you will see, the Czech Republic is divided into thirteen regions (kraje). Regional differences impacted the development of Czech cuisine. Recipes varied depending upon whether ingredients were available (locally-sourced or brought via trade), climate and geography, and cooking traditions.

One definition of regional Czech cuisine is: "a tradition of food preparations that arose from the daily lives and kitchens of the people across a specific region of the Czech lands. This is notably distinct from the cuisine of the Czech nation as a whole."


Page Partners

 ORIGINAL CZECH DISTILLERY

Rudolf Jelinek

A company that looks with pride to its very long history and considers itself an inseparable part of the Czech region - Wallachia.


[Visit website](#)



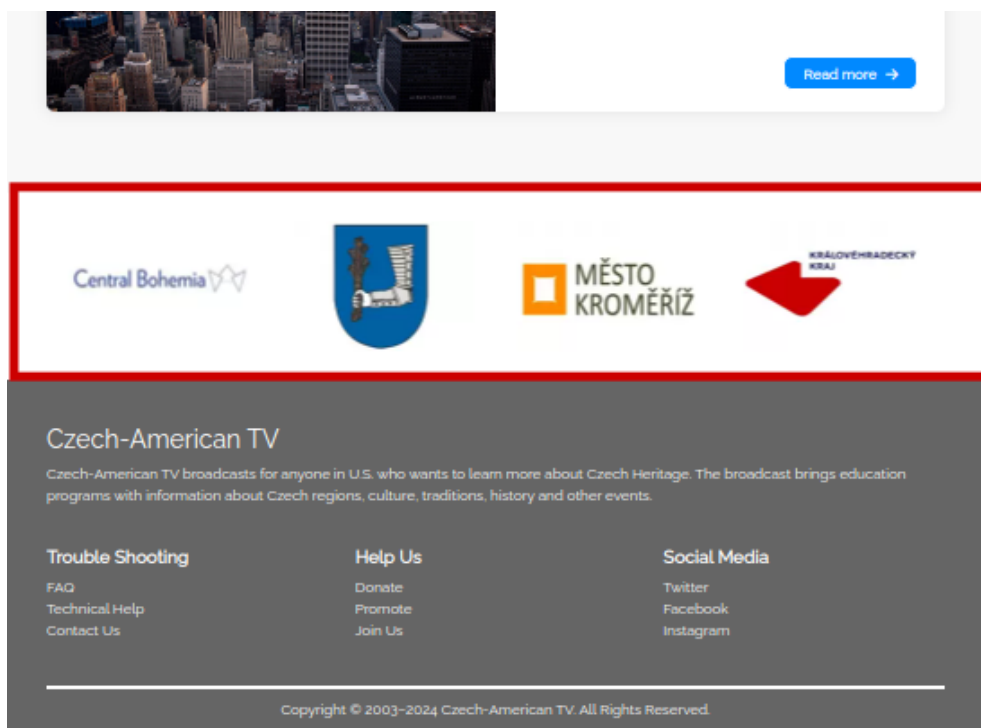
Your Name

If you want to become a partner of this post, contact us at the link below:

[I want to be a partner](#)



■ Obrázek 4 Page partner



The image shows a website slider with a cityscape background and a "Read more" button. Below the slider is a red-bordered box containing four partner logos: Central Bohemia, a blue shield with a white cross, Město Kroměříž, and Královéhradecký kraj. Below this box is a dark grey footer with the text "Czech-American TV" and a description of the service. The footer also contains three columns of links: "Trouble Shooting" (FAQ, Technical Help, Contact Us), "Help Us" (Donate, Promote, Join Us), and "Social Media" (Twitter, Facebook, Instagram). At the bottom, there is a copyright notice: "Copyright © 2003-2024 Czech-American TV. All Rights Reserved."

Read more →

Central Bohemia

Město Kroměříž

Královéhradecký kraj

Czech-American TV

Czech-American TV broadcasts for anyone in U.S. who wants to learn more about Czech Heritage. The broadcast brings education programs with information about Czech regions, culture, traditions, history and other events.

Trouble Shooting
FAQ
Technical Help
Contact Us

Help Us
Donate
Promote
Join Us


Social Media
Twitter
Facebook
Instagram

Copyright © 2003-2024 Czech-American TV. All Rights Reserved.

■ Obrázek 5 Slider partner

Home Broadcasts Cooking Audio Contest Sights Traditions Notables Genealogy Membership

Our Partners



Page Partners

R. JELÍNEK
ORIGINAL CZECH DISTILLERY

Rudolf Jelínek

A company that looks with pride to its very long history and considers itself an inseparable part of the Czech region - Wallachia.

[Visit website](#)

Your Logo


Your Name

If you want to become a partner of this post, contact us at the link below.

[I want to be a partner](#)

■ Obrázek 6 Our partner

Partner's Stories

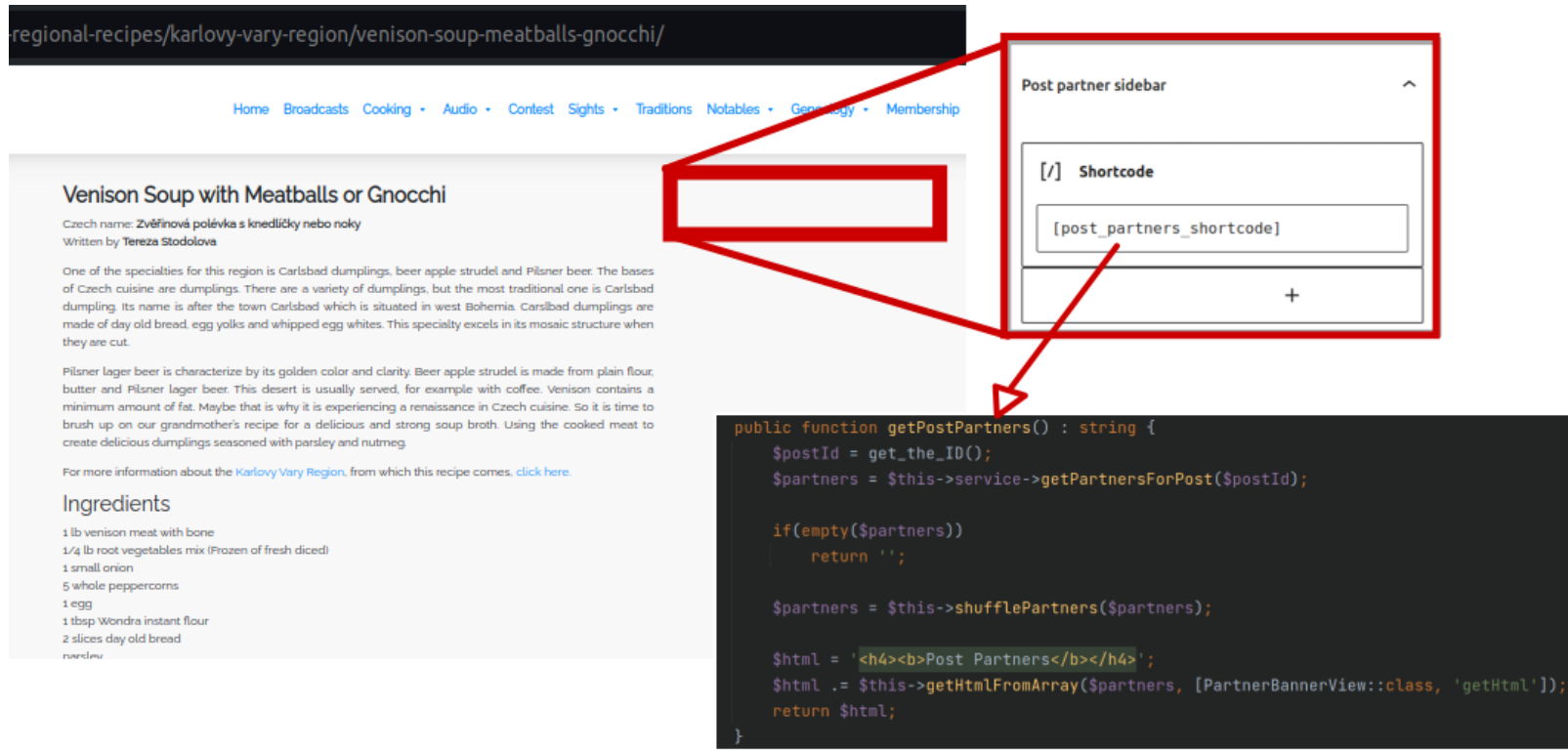


Explore City of Brno

Brno is a young, friendly city with first-rate gastronomy and an atmosphere you cannot experience anywhere else. Brno is a city with an authentic atmosphere, and it is surrounded by greenery.

[Read more →](#)

■ Obrázek 7 Partner stories



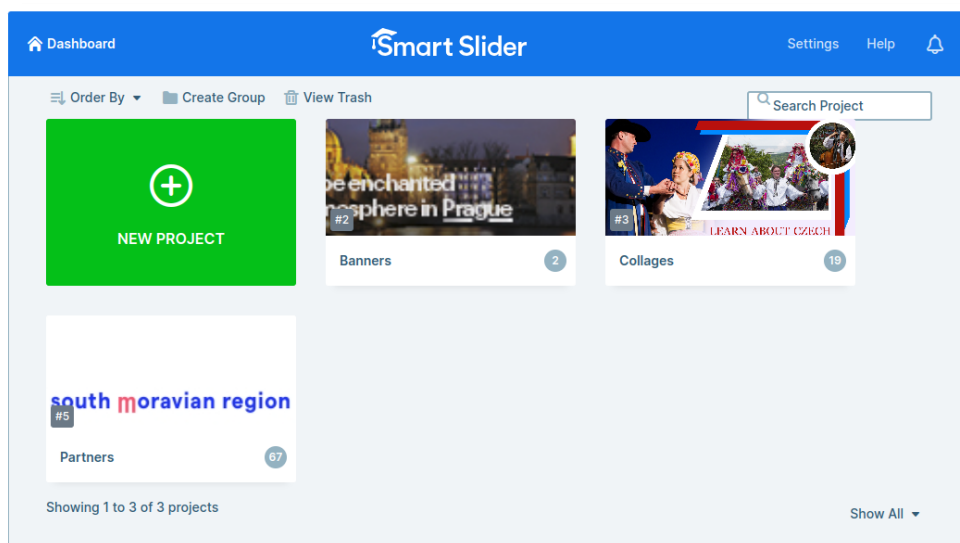
■ Obrázek 8 Integrace do stránky

```
<div class="broadcast-partners-container">
  <?php
    if($post->post_type == 'broadcast') {
      echo "<h4><b>Broadcast Partners</b></h4>";
    } else if(is_page()) {
      echo "<h4><b>Page Partners</b></h4>";
    } else{
      echo "<h4><b>Post Partners</b></h4>";
    }
    dynamic_sidebar('partners-vertical');
  ?>
</div>
```

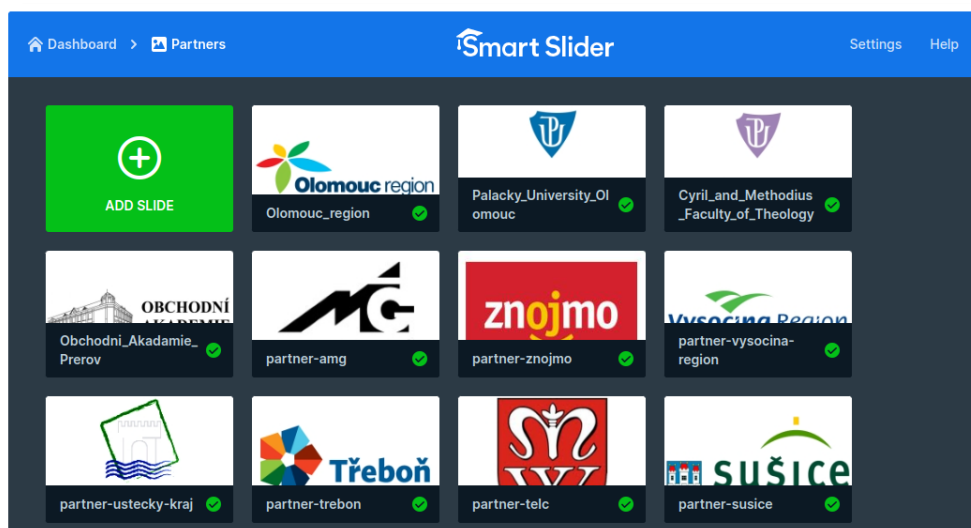


```
<div class="broadcast-partners-container">
  <?php
    if($post->post_type == 'broadcast') {
      if( defined('PARTNERS_MANAGER_SIDEBAR_SINGLE_BROADCAST_PARTNERS')) {
        dynamic_sidebar(PARTNERS_MANAGER_SIDEBAR_SINGLE_BROADCAST_PARTNERS);
      }
      if( defined('PARTNERS_MANAGER_SIDEBAR_BROADCAST_CONTENT')) {
        dynamic_sidebar(PARTNERS_MANAGER_SIDEBAR_BROADCAST_CONTENT);
      }
    } elseif(is_page()) {
      if( defined('PARTNERS_MANAGER_SIDEBAR_PAGE_PARTNERS')) {
        dynamic_sidebar(PARTNERS_MANAGER_SIDEBAR_PAGE_PARTNERS);
      }
    } elseif( defined('PARTNERS_MANAGER_SIDEBAR_POST_PARTNERS')){
      dynamic_sidebar(PARTNERS_MANAGER_SIDEBAR_POST_PARTNERS);
    }
  ?>
</div>
```

■ **Obrázek 9** Ukázka změn provedených v souboru singular.php

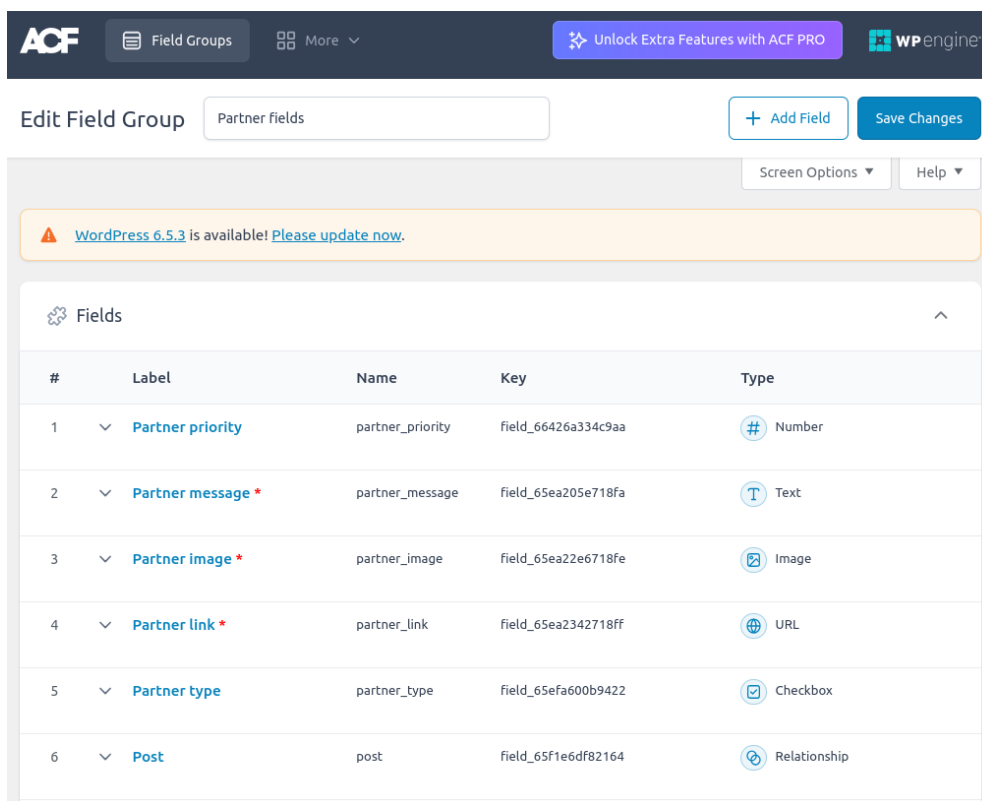


(a) Přehled projektů



(b) Přehled jednotlivých slidů

■ Obrázek 10 Administrace v pluginu Smart slider



The screenshot shows the 'Edit Field Group' interface for the 'Partner fields' group. At the top, there is a navigation bar with the ACF logo, 'Field Groups', and 'More' options. A purple banner promotes 'Unlock Extra Features with ACF PRO', and the WPengine logo is on the right. Below the navigation, the title 'Edit Field Group' is followed by a search box containing 'Partner fields', a '+ Add Field' button, and a 'Save Changes' button. On the right side, there are 'Screen Options' and 'Help' dropdown menus. A yellow warning banner at the top of the main content area states 'WordPress 6.5.3 is available! Please update now.' Below this is a section titled 'Fields' with a list of six fields. Each field row includes a number, a label with a dropdown arrow, the field name, the key, and the field type with its icon.

#	Label	Name	Key	Type
1	Partner priority	partner_priority	field_66426a334c9aa	Number
2	Partner message *	partner_message	field_65ea205e718fa	Text
3	Partner image *	partner_image	field_65ea22e6718fe	Image
4	Partner link *	partner_link	field_65ea2342718ff	URL
5	Partner type	partner_type	field_65efa600b9422	Checkbox
6	Post	post	field_65f1e6df82164	Relationship

■ **Obrázek 11** Skupina meta fields v nastavení pluginu ACF

7 ^ Post post field_65f1e6df82164 Relationship

Edit Duplicate Move Delete

General Validation Presentation Conditional Logic Advanced

Field Type

Relationship Browse Fields

Field Label

Post

This is the name which will appear on the EDIT page

Field Name

post

Single word, no spaces. Underscores and dashes allowed

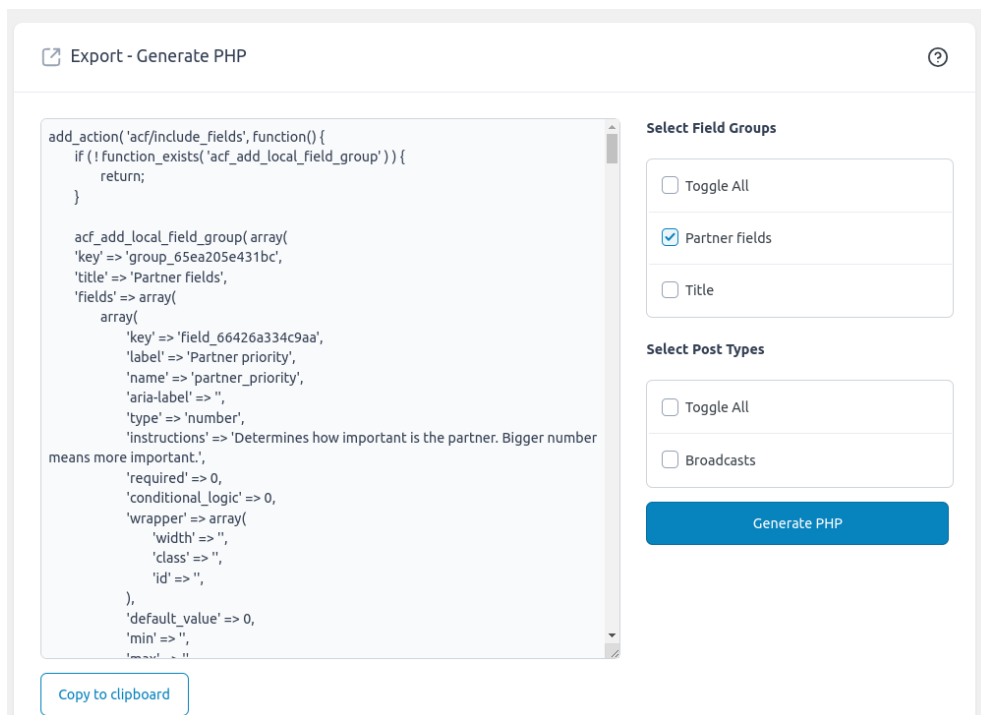
Filter by Post Type

Post

Filter by Post Status

Published Scheduled

■ **Obrázek 12** Nastavení post meta field v pluginu ACF



■ **Obrázek 13** Export skupiny meta fields v pluginu ACF

```

public function testSavePartnerStory(){
    $partner = $this->getPartnerData();
    $story = $this->getPartnerStoryWithContent();

    $this->preparePartnerStoryVariations($story, true, $partner);

    $this->partnerStoryRepositoryMock->expects($this->once())
        ->method('savePage')
        ->with($story, $partner->id);

    $this->expectNotCallingDeletePage();

    // Execute
    $this->partnerStoryService->savePartnerStory($partner->id);
}

```

■ **Výpis kódu 1** Jednotkový test testSavePartnerStory

```
private function expectNotCallingDeletePage (){
    $this->partnerStoryService =
        → $this->getMockBuilder(PartnerStoryService::class)
            ->setConstructorArgs([
                $this->partnerStoryRepositoryMock,
                $this->partnerRepositoryMock
            ])
            ->onlyMethods(['deletePartnerStoryPage'])
            ->getMock();

    $this->partnerStoryService->expects($this->never())
        ->method('deletePartnerStoryPage');
}
```

■ **Výpis kódu 2** Pomocná funkce pro jednotkové testy

Seznam zkratek

CMS	Content Management System
API	Application programming interface
JS	JavaScript
UC	Use case
CATVUSA	Czech-American TV
DI	Dependency Injection