|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Web kapely s využitím CMS Wordpress** | | |
| Daniel Havranek | | |
| [místo pro vložení obrázku] | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2018/2019 | |

#### Poděkování

* *poděkování (například vedoucímu práce).*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2016

*podpis autora práce*

**ANOTACE**

- česky

- slouží jako pomoc čtenáři rychle se zorientovat v dané práci.

- *„Redukovaný text, který charakterizuje obsah dokumentu bez rozlišování autorství abstraktu, bez doplňkových informací, bez vlastní interpretace a hodnocení dokumentu (tj. nikoliv "v práci velmi dobře hodnotím podle mne zajímavý systém...", ale "práce hodnotí systém..."). Základními vlastnostmi anotace jsou výstižnost, přehlednost, jasnost, stručnost, přesnost, objektivnost a čtivost. Anotace je formulována v přirozeném jazyce – obvykle ve větách. Anotace může používat textových formulací z referovaného dokumentu, ale jako celek je formulován nově.“*

- délka cca 100 – 250 slov

OBSAH

[Úvod 5](#_Toc370246085)

[1 Teoretická a metodická východiska 6](#_Toc370246086)

1.1 Hudební webové stránky 6

1.2 Systémy pro správu obsahu 6

[2 Využité technologie 7](#_Toc370246087)

2.1 CMS WordPress 7

2.1.1 Šablony 7

2.1.2 Pluginy 7

2.2 MySQL 8

2.3 Webové technologie 8

2.4 PHP 8

2.5 Visual Studio Code 8

[3 Způsoby řešení a použité postupy 8](#_Toc370246088)

3.1 Instalace WordPressu 9

3.2 Vytváření a design 9

3.3 Eshop 11

3.4 Vlastní šablona příspěvku 11

3.5 Zabezpečení 12

3.6 Výcejazyčnost 13

3.7 Přesun stránek na server 14

[4 Výsledky řešení, výstupy, uživatelský manuál 9](#_Toc370246089)

4.1 Správa webových stránek 15

4.1.1 Příspěvky 15

4.1.2 Koncerty 16

4.1.3 Galerie 17

4.1.4 Eshop 18

4.2 Veřejná část 18

[Závěr 10](#_Toc370246090)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 11](#_Toc370246091)

Úvod

Jako hudebník vím, že nedílnou součástí každé kapely jsou webové stránky, a tak jsem se rozhodl takové stránky vytvořit. K tomuto úkolu jsem využil systém WordPress, což je systém pro správu obsahu. Web patří smyšlené kapele Mugshots a jako všechny podobné hudební weby i tento měl obsahovat dílčí stránky se zprávami, nadcházejícími koncerty, galerií atd.

V této dokumentaci budu rozebírat vytváření různých částí stránek a postupy, které jsem použil při jejich zprovozňování.

# Teoretická a metodická východiska

**1.1 Hudební webové stránky**

Většina webových stránek kapel je svou stavbou podobná. Takové stránky obsahují nejnovější zprávy o kapele, nadcházející koncerty, galerie, informace o hudbě (ukázky skladeb) a případně eshop. Já se inspiroval moderními weby slavných kapel. Ty většinou neumožňují vytváření uživatelských účtů a komentování příspěvků přímo na webu. K těmto účelům se stránky propojují se sociálními sítěmi.

**1.2 Systémy pro správu obsahu**

Systém pro správu obsahu (CMS - *content management system*) je [software](https://cs.wikipedia.org/wiki/Software) zajišťující [správu dokumentů](https://cs.wikipedia.org/wiki/Spr%C3%A1va_dokument%C5%AF), nejčastěji [webového](https://cs.wikipedia.org/wiki/World_Wide_Web) obsahu. V dnešní době se jako CMS většinou chápou [webové aplikace](https://cs.wikipedia.org/wiki/Webov%C3%A1_aplikace). Pro CMS se někdy používají i oborově podobné termíny redakční či publikační systém.

Mezi základní funkce CMS patří:

* tvorba, modifikace a publikace dokumentů (článků), obvykle prostřednictvím webového rozhraní, často s využitím jednoduchého online [WYSIWYG](https://cs.wikipedia.org/wiki/WYSIWYG) editoru nebo jednoduchého systému formátování textu
* řízení přístupu k dokumentům, většinou se správou uživatelů a přístupových práv
* správa diskusí či komentářů, ať už k publikovaným dokumentům nebo obecných
* správa souborů, obrázků či galerií

# Využité technologie

**2.1 CMS WordPress**

WordPress je svobodný open-source redakční publikační systém napsaný v PHP a MySQL. Jeho funkce obsahují architekturu pluginů a systém šablon. Je nejvíce používaný pro vytváření blogů, ale podporuje i ostatní typy webového obsahu. Tím, že je používán na více než 60 milionech webových stránek, je WordPress nejpopulárnější webový redakční systém.

Při vytváření webu jsem používal verzi 4.9. Tu jsem později aktualizoval na 5.0, která, jako většina aktualizací, opravovala bezpečnostní chyby. Také ale přidala novou možnost vytváření stránek s editorem Gutenberg.

**2.1.1 Šablony**

Uživatelé wordpressu si mohou vybrat z velkého množství šablon. Tyto šablony umožňují uživatelům změnit vzhled a funkčnost webových stránek beze změny kódu nebo obsahu stránky.

**2.1.2 Pluginy**

Architektura pluginů dovoluje uživatelům rozšířit možnosti a funkce webových stránek. Každý plugin nabízí různé funkce, které pomáhají uživatelům přizpůsobit své stránky podle potřeby.



**2.2 MySQL**

MySQL je [systém řízení báze dat](https://cs.wikipedia.org/wiki/Syst%C3%A9m_%C5%99%C3%ADzen%C3%AD_b%C3%A1ze_dat) uplatňující [relační databázový model](https://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD_model). Je k dispozici jak pod bezplatnou licencí [GPL](https://cs.wikipedia.org/wiki/GNU_General_Public_License), tak pod komerční placenou licencí. Je to [multiplatformní](https://cs.wikipedia.org/wiki/Multiplatformn%C3%AD_software) databáze. Komunikace s ní probíhá pomocí jazyka [SQL](https://cs.wikipedia.org/wiki/SQL). Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s některými rozšířeními.

**2.3 Webové technologie**

K vytváření vlastní šablony příspěvku jsem využil značkovací jazyk pro tvorbu [webových stránek](https://cs.wikipedia.org/wiki/Webov%C3%A1_str%C3%A1nka) HTML5 a vzhled stránky jsem upravoval pomocí kaskádových stylů CSS 3.0, což je jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v jazycích [HTML](https://cs.wikipedia.org/wiki/HyperText_Markup_Language), [XHTML](https://cs.wikipedia.org/wiki/Extensible_HyperText_Markup_Language) nebo [XML](https://cs.wikipedia.org/wiki/Extensible_Markup_Language).

**2.4 PHP**

PHP je [skriptovací](https://cs.wikipedia.org/wiki/Skriptovac%C3%AD_jazyk) [programovací jazyk](https://cs.wikipedia.org/wiki/Programovac%C3%AD_jazyk), který je určený především pro programování dynamických [internetových stránek](https://cs.wikipedia.org/wiki/Webov%C3%A1_str%C3%A1nka) a webových aplikací například ve formátu [HTML](https://cs.wikipedia.org/wiki/Hypertext_Markup_Language), [XHTML](https://cs.wikipedia.org/wiki/Extensible_Hypertext_Markup_Language) či [WML](https://cs.wikipedia.org/wiki/Wireless_Markup_Language). Při použití PHP pro dynamické stránky jsou skripty prováděny na straně [serveru](https://cs.wikipedia.org/wiki/Server) – k uživateli je přenášen až výsledek jejich činnosti. Při tvorbě webu jsem používal PHP 7.2.

**2.5 Visual Studio Code**

Visual Studio Code je [multiplatformní](https://cs.wikipedia.org/wiki/Multiplatformn%C3%AD_software) editor textových souborů a zdrojového kódu vytvořený společností Microsoft. Tento program obsahuje podporu pro debugging, vestavěné ovládání Gitu, zvýraznění syntaxe, inteligentní dokončení kódu, snippety a další a další možnosti pro pohodlnější programování.

# Způsoby řešení a použité postupy

**3.1 Instalace WordPressu**

WordPress vyžaduje k instalaci databázi a server se schopností spouštět PHP skripty. Já provedl instalaci a celé vytváření stránek nejprve lokálně s využitím vývojového prostředí XAMPP.

Po zadání adresy stránek v prohlížeči (v mém případě *localhost/wordpress/*) se objeví varování, které informuje o tom, že doposud nebyl vytvořen konfigurační soubor wp-config.php a vybízí k jeho vytvoření. Se souhlasem se pak spustí samotná instalace. Její součástí je konfigurace databáze a následně základní konfigurace samotného webu, ve které jde jen o zadání názvu webu a vytvoření administrátorského účtu.

**3.2 Vytváření a design**

Při vytváření webu si nejprve musíte zvolit šablonu. WordPress nabízí nějaké vlastní, ale můžete si vybrat z mnoha jiných, které lze stáhnout. Šablonu si pak v sekci pro přizpůsobení vzhledu můžete upravit podle vlastní potřeby. Zde se vytváří menu, záhlaví, zápatí, nastavují se styly atd. Jednotlivé stránky webu pak můžete vytvářet v sekci *stránky*, kde máte také přehled již vytvořených stránek.

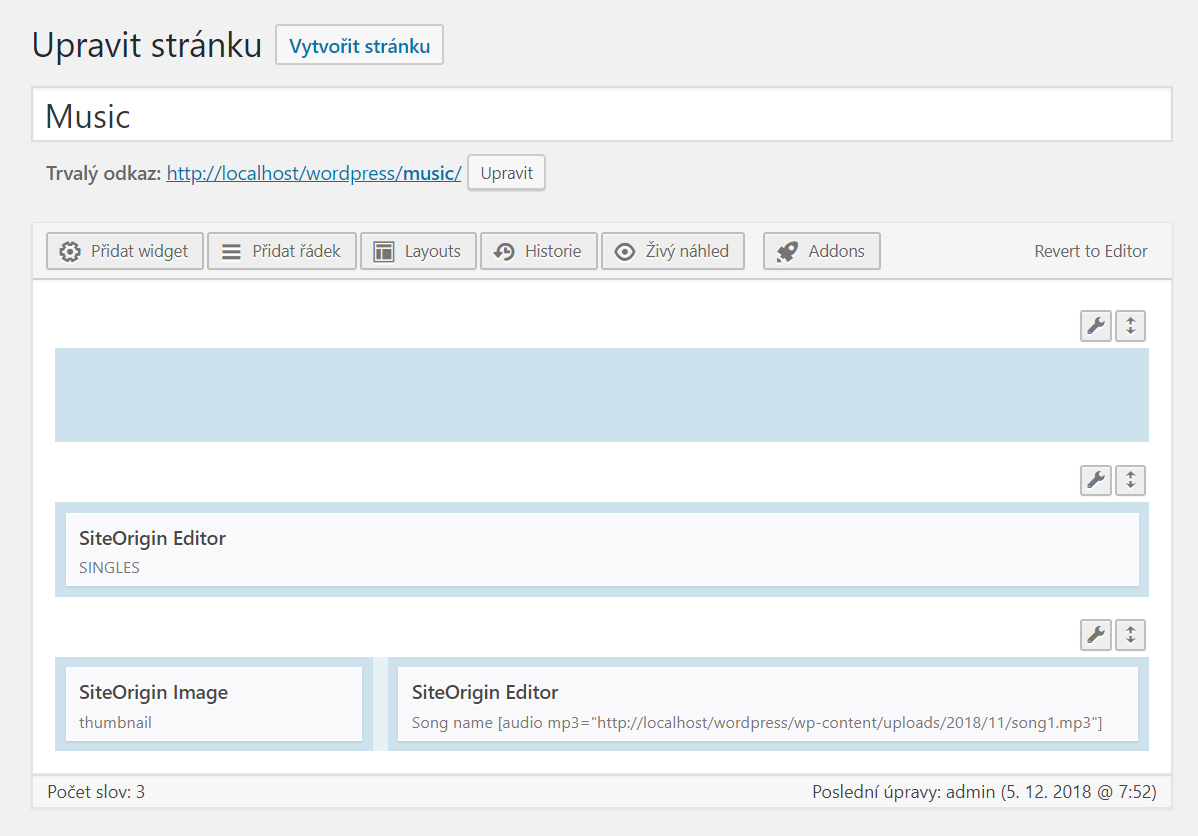
Na svůj web jsem chtěl použít jednoduchý a moderní design, a tak jsem si vybral šablonu OceanWP, která je dobře hodnocená a pro tento design vhodná. Vytváření stránek obyčejným způsobem mi ale přišlo zdlouhavé a často takto nešlo docílit vzhledu, který bych chtěl, a proto jsem se tvorbu stránek rozhodl zlepšit a urychlit pomocí pluginu.

Pluginů pro vytváření stránek je několik. Já se rozhodl použít plugin Page Builder by SiteOrigin, který je zdarma a umožňuje vytvářet responzivní stránky, kompatibilní se všemi šablonami. Obsahuje jednoduché *„drag and drop“* rozhraní, ve kterém se pro zobrazení obsahu používají widgety.

Základem pro vytváření stránek s tímto pluginem je panel nástrojů, ve kterém se nachází tyto funkce:

* Přidat widget - přidá obsah s využitím wordpress widgetu, který vyberete
* Přidat řádek - vytváří layout pomocí vkládání řádků a sloupců
* Layouts - klonuje stránky nebo vkládá předem vytvořené layouty
* Historie - ukládá postup pro případnou možnost vrácení
* Živý náhled - umožňuje upravovat obsah s náhledem v reálném čase

Vytvořené řádky a sloupce pak lze libovolně přesouvat a v každém můžete přizpůsobit styly nebo je doplnit vlastním CSS kódem.



Nová verze WordPressu obsahuje vlastní editor pro jednodušší vytváření stránek s názvem Gutenberg. Při vytváření jsem ale používal verzi 4.9, a tak jsem neměl možnost tento editor vyzkoušet.

**3.3 Eshop**

Pro vytvoření internetového obchodu na stránkách jsem zvolil plugin WooCommerce…

…

…

**3.4 Vlastní šablona příspěvku**

Protože jsem k vytváření webu použil plugin Page builder, při přidávání nových příspěvků by se otevřel tento editor a uživatel by musel každý příspěvek vytvořit jako stránku. Toto je zdlouhavé a ne každý uživatel tomuto vytváření stránek rozumí.

Jednou z možností jak tento proces zjednodušit je použití funkce *prebuilt layout*, kterou editor Page Builder nabízí. Po zvolení této možnosti se otevře okno s nabídkou již vytvořených stránek a příspěvků. Pokud už je tedy nějaký příspěvek s využitím tohoto editoru vytvořen, uživatel jej může vybrat a celá stránka se zkopíruje do nového příspěvku. Pak stačí jen v editoru změnit nadpis, text a obrázek, ale pokaždé se musí v meta boxu šablony OceanWP provést nastavení vzhledu. Toto tedy také není úplně vhodné, neboť uživatel může jednoduše měnit vzhled celé stránky a přidávání nového příspěvku zabere víc času.

V nastavení lze vypnout defaultní zobrazování editoru Page Builder při přidávání nových příspěvků a stránek. Takto se uživateli zobrazí pouze textové pole pro nadpis a pro text a další meta boxy, což už je srozumitelné pro každého. Když ale obsah příspěvku zadá takto, stránka se zobrazí podle použité šablony. Úprava této šablony je však dost omezená a vzhled této stránky by se nehodil ke vzhledu celého webu.

Aby tedy mohl uživatel jednoduše přidávat nové příspěvky, musel jsem v adresáři použité šablony vytvořit nový PHP soubor s vlastním rozvržením stránky příspěvku. Pro zobrazení meta boxu s možností výběru šablony při přidávání nového příspěvku se musí v poznámce kódu zadat název a typ šablony (*Template Name* a *Template Post Type*). Pro získání záhlaví a zápatí jako na ostatních stránkách webu lze použít funkci **get\_header();** a **get\_footer();**.

Dále stačilo pomocí HTML a CSS vytvořit a přizpůsobit stránku a standardní smyčkou vypsat obsah daného příspěvku. Ta se v kódu zapíše pomocí funkce **while ( have\_posts() ) : the\_post();** a **endwhile;** a lze v ní pak použít funkce jako např. **the\_title();** a **the\_content();** pro získání obsahu příspěvku.

<?php

/\*

\* Template Name: My post style

\* Template Post Type: post

\*/

get\_header(); ?>

<header></header>

<div id="content" class="site-content clr">

    <?php while ( have\_posts() ) : the\_post(); ?>

    <div class="main-post-div">

        <div id="content-wrap" class="container clr">

            <div class="oceanwp-row clr">

                <!-- Obrazek prispevku -->

                <?php $url = wp\_get\_attachment\_url( get\_post\_thumbnail\_id($post->ID)); ?>

                <img src="<?php echo $url ?>" width="500px" id="post-img"/>

                <!-- Nadpis prispevku -->

                <h1><?php the\_title(); ?></h1>

                <!-- Datum zverejneni-->

                <p><?php echo get\_the\_date('F j, Y'); ?></p>

                <!-- Obsah prispevku -->

                <p><?php the\_content(); ?></p>

            </div>

        </div>

    </div>

    <?php endwhile; ?>

</div>

<?php get\_footer(); ?>

**3.5 Zabezpečení**

CMS systém Wordpress je také známý díky svému nedostatečnému zabezpečení. Protože jej využívá velké množství webových stránek, stal se častým cílem útoků hackerů.

Stránky využívající tento systém jsou často napadány skrze nezabezpečený hosting a proto se v prvé řadě doporučuje tento systém využívat na bezpečném hostingu.

Základem pro bezpečný web je zálohování a pravidelné aktualizace. Starší verze pluginů, šablon nebo samotného Wordpressu mohou obsahovat bezpečnostní chyby, a proto je důležité používat jejich nejnovější verze. Aby se šablona a pluginy automaticky aktualizovaly, stačí přidat do souboru **functions.php** následující dva řádky kódu:

add\_filter( 'auto\_update\_plugin', '\_\_return\_true' );

add\_filter( 'auto\_update\_theme', '\_\_return\_true' );

Dále existuje mnoho různých pluginů, které pomáhají stránky zabezpečit. Já se rozhodl použít iThemes Security, který poskytuje bezpečnostní prvky jako například:

* pravidelná záloha databáze
* ochrana proti „brute force“ útoku - sníží počet pokusů při přihlášení
* změna URL adresy pro přihlášení do administrace
* emailové notifikace - pokud dojde ke změně souborů nebo se někdo vícekrát pokusí o přihlášení se špatnými údaji
* změna předpony tabulek databáze
* blokace specifických IP adres

**3.6 Vícejazyčnost**

K vytvoření vícejazyčných stránek existuje mnoho pluginů. Jeden z nejlepších je WPML, který má mnoho možností a je kompatibilní s většinou dalších pluginů, jako např. WooCommerce, u kterého překlad také potřebuji. Plugin WPML je však placený, a protože jsem chtěl tyto stránky vytvořit bez jakýchkoli investic, musel jsem najít alternativu.

Bezplatný a dobře fungující plugin je Polylang. Po aktivaci lze v nastavení konfigurovat. Jako první se musí nastavit základní jazyk stránek a poté přidat další jazyky, mezi kterými bude moct uživatel přepínat. Dále je zde možnost přeložit název webu, jeho popis a formát data a času.

S tímto pluginem lze jednoduše přidávat obsah v různých jazycích. Při přidávání nových příspěvků, nebo u upravování starých, se objeví meta box pro překlad do jiných jazyků. Také umožňuje ke každé stránce přidat další stránku v jiném jazyce. Tato možnost však nefunguje u stránek eshopu vytvořenými pomocí WooCommerce.

Pro překlad těchto stránek je potřeba stáhnout rozšíření **WooCommerce Polylang Integration. Ke správnému fungování jsem ale ještě musel upravit soubor uložený v adresáři** *wp-content/plugins/woocommerce/includes/wc-page-functions.php*. Zde jsem přepsal funkci **wc\_get\_page\_id($page);**, aby nejprve zjistila, jestli je Polylang nainstalovaný a existuje funkce **pll\_get\_post();**. Pokud ano, WooCommerce musí vyhledat přeloženou stránku. Pokud tato stránka neexistuje, zobrazí standardní stránku.

Aby mohl uživatel webu mezi jazyky přepínat, je nutné někde umístit widget, který byl přidán s pluginem. K tomuto widgetu je jednoduché nastavení pro zobrazení.

**3.7 Přesun stránek na server**

Nejprve jsem vyzkoušel vestavěnou funkci wordpressu pro import/export. Při přesunu tímhle způsobem se po importu objeví všechna data (stránky, příspěvky, menu atd.), přizpůsobit se však stránky musí znovu, což mi přijde zdlouhavé a nevyhovující.

Našel jsem proto dva nejčastější způsoby pro přesun stránek na server:

* manuální metoda
* pomocí pluginu

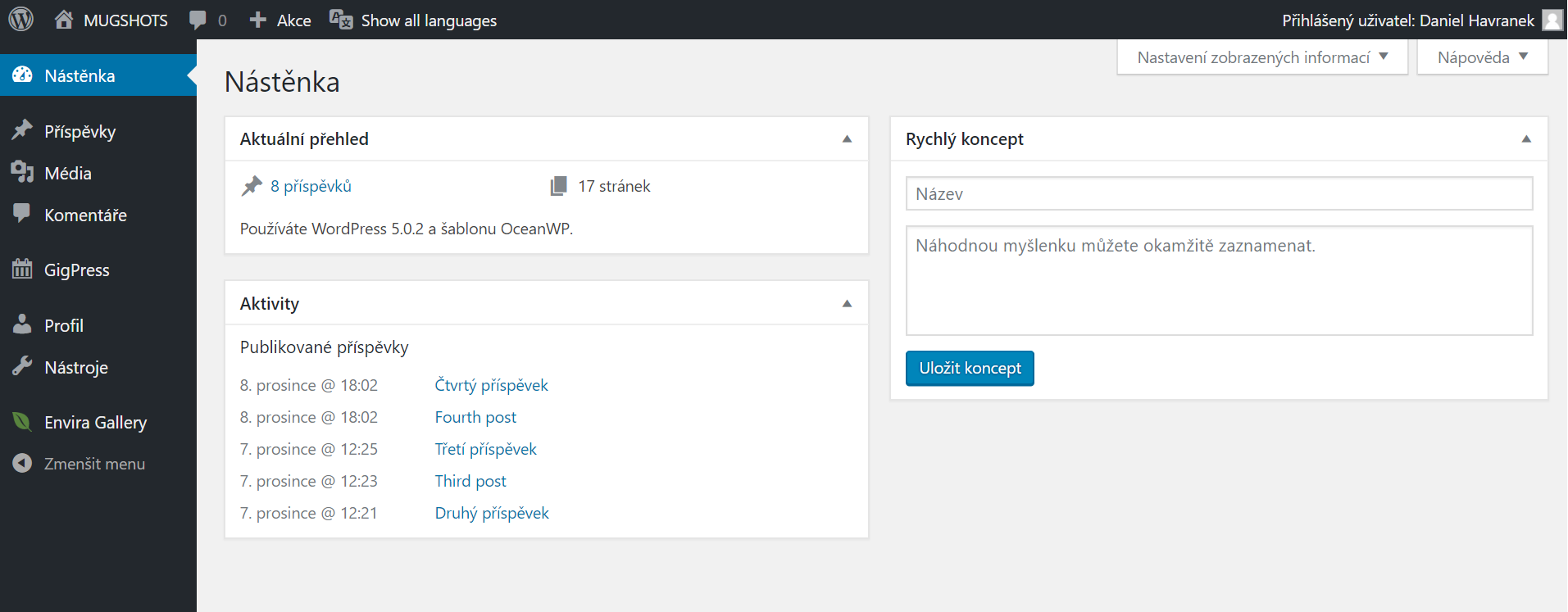
Protože jsem se snažil používat co nejméně pluginů, pro nahrání vytvořených webových stránek na server jsem použil manuální metodu.

Jako první jsem exportoval lokální wordpress databázi z phpMyAdmin do formátu SQL. V aplikaci FileZilla FTP client jsem se připojil na server a zkopíroval na něj celou složku wordpress uloženou v mém počítači. Na stránkách webhostingu jsem vytvořil datábazi a přidal do ní uživatele. Dále jsem importoval SQL soubor s wordpress datábází, který jsem získal v prvním kroku, do databáze vytvořené na webhostingu. V tabulce wp\_options jsem nahradil všechny odkazy na staré místa v localhostu novými. V souboru wp-config.php, který je uložen v adresáři wordpressu, jsem změnil jméno databáze, uživatele a heslo na hodnoty které jsem nastavil na stránkách hostingu. Po všech těchto krocích byly stránky na serveru aktivní a po přihlášení jsem v nastavení pouze uložil změny.

# Výsledky řešení, výstupy, uživatelský manuál

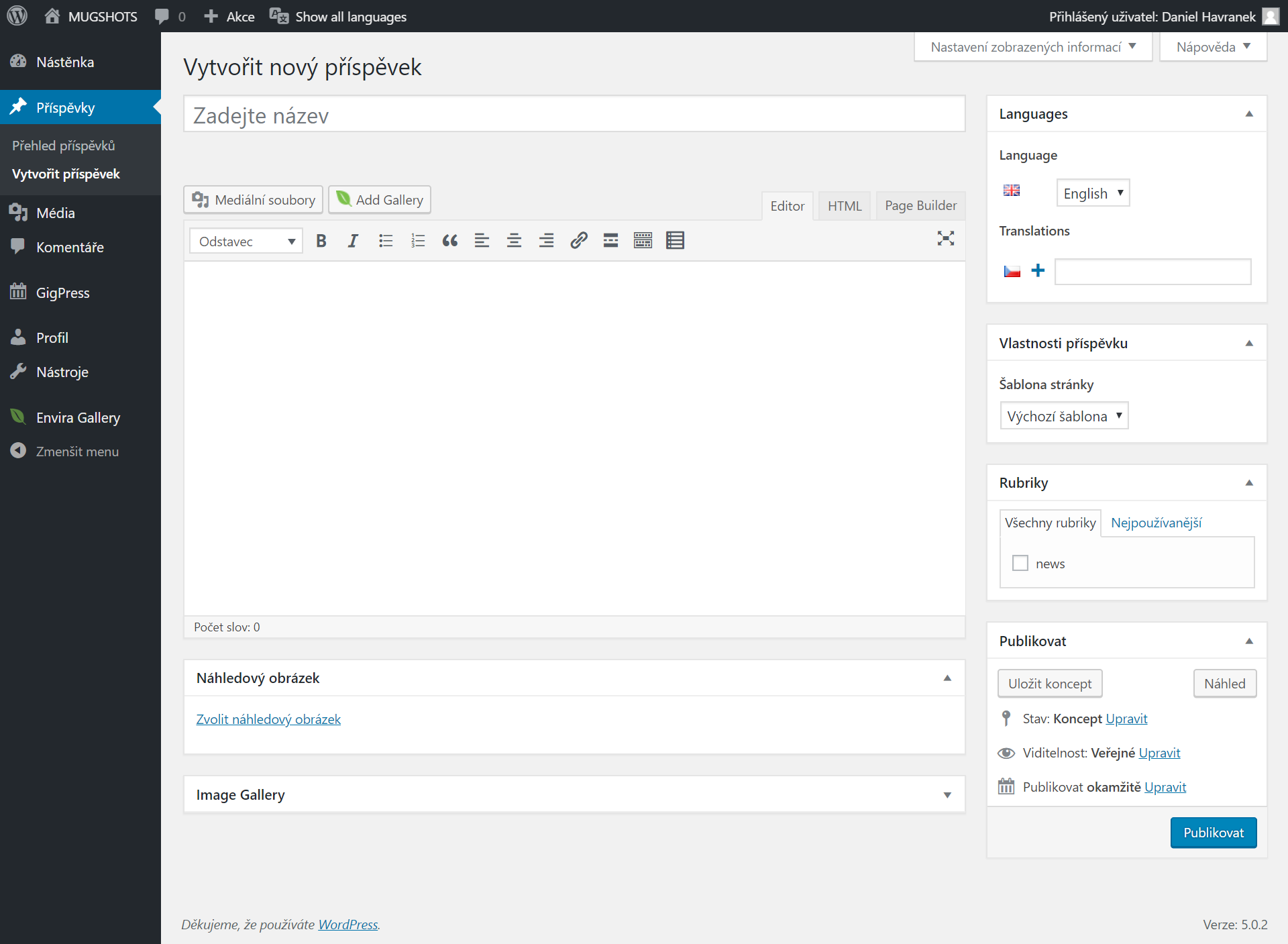
**4.1 Správa webových stránek**

Pro vstup do administrační části webu stačí za URL adresu zadat */cms*. Takto se objeví přihlašovací stránka WordPress. Po přihlášení se uživateli zobrazí nástěnka.



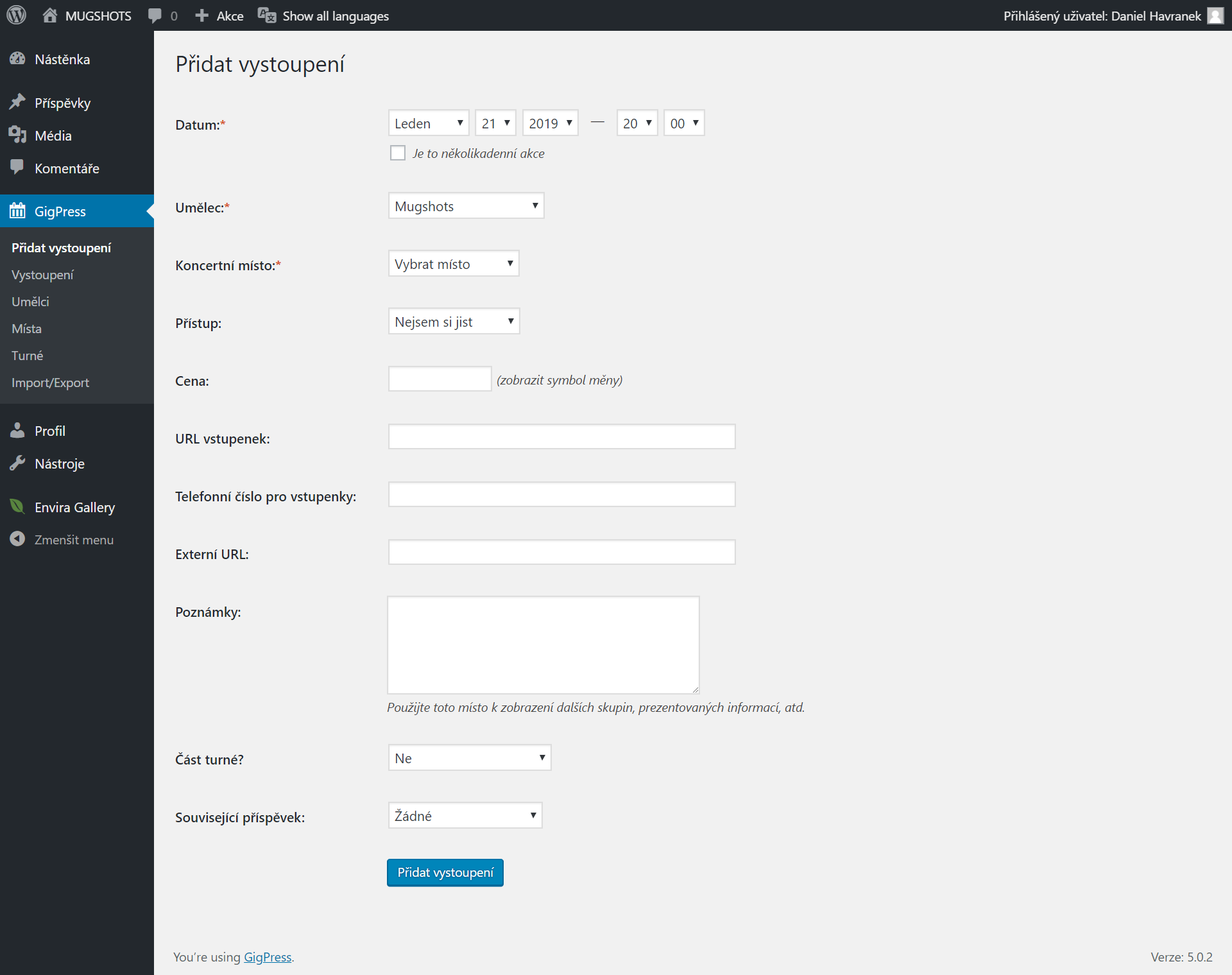
**4.1.1 Příspěvky**

Při přidávání příspěvků stačí zadat nadpis a obsah v jednoduchém textovém editoru. Dále se musí vybrat požadovaná šablona, označit rubrika a případně přidat obrázek. Při psaní příspěvku je možnost vytvořit jeho překlad pomocí meta boxu s jazyky.



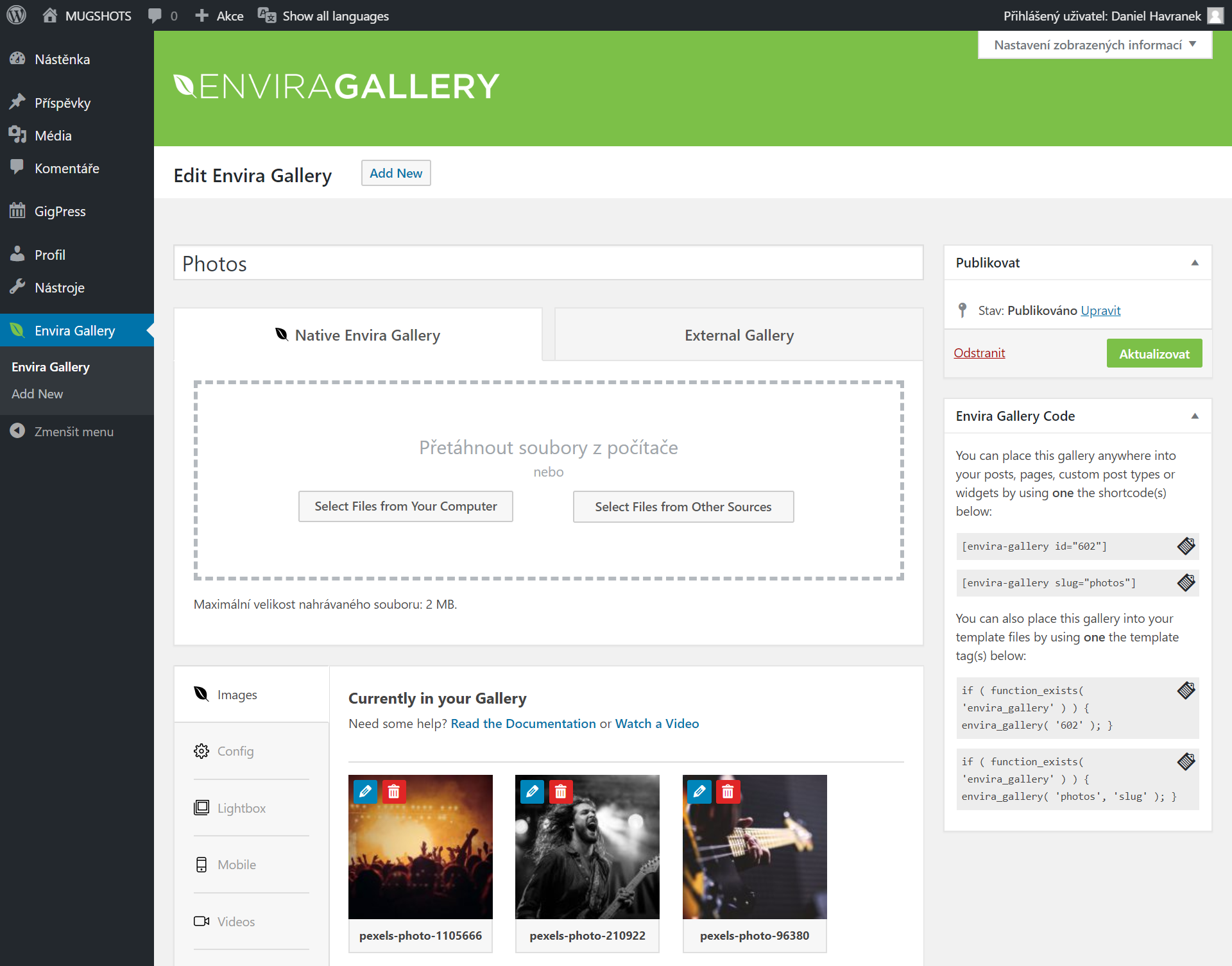
**4.1.2 Koncerty**

Pro správu plánovaných vystoupení je zde záložka GigPress, kde lze zobrazit všechny vystoupení nebo místa a upravovat je. Pro přidání nového koncertu stačí vyplnit jednoduchý formulář.



**4.1.3 Galerie**

V záložce Envira Gallery je možnost spravovat galerii, tzn. přidávat, upravovat nebo odstraňovat fotky.

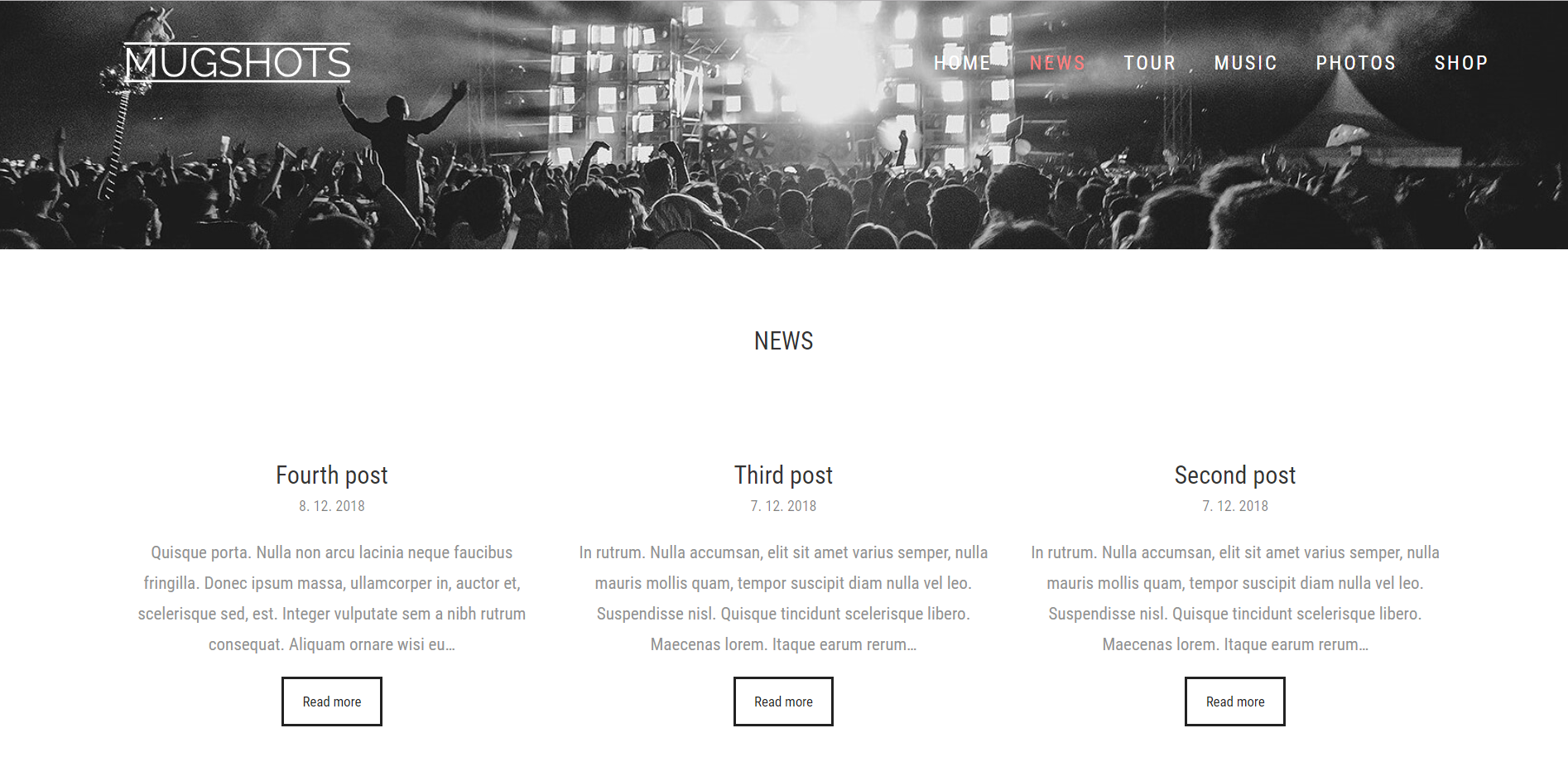
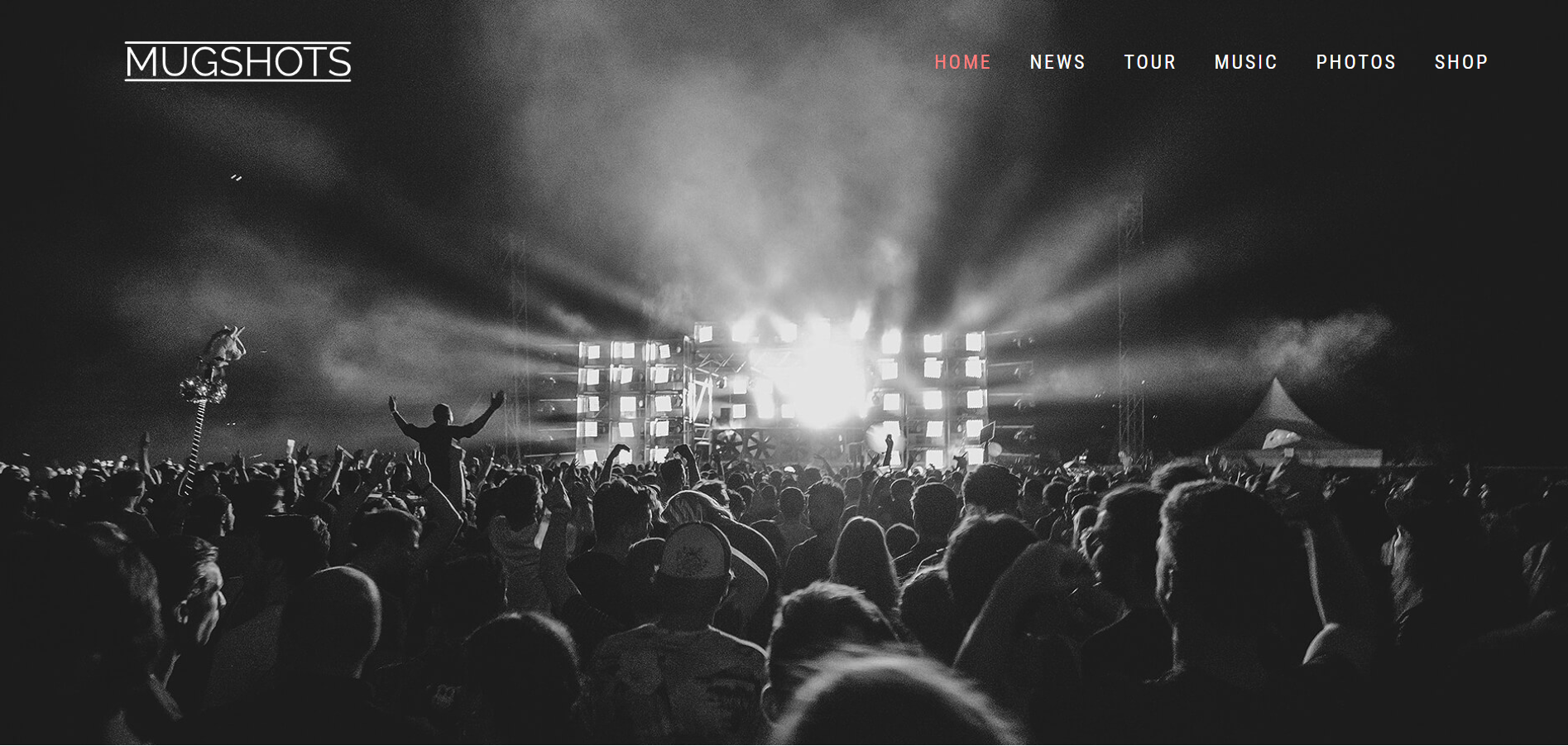


**4.1.3 Eshop**

…

**4.2 Veřejná část**

Veřejnou část tvoří stránky home, news, tour, music, photos a shop. Zde může návštěvník najít nejnovější zprávy, nadcházející koncerty, ukázky hudby, galerii a obchod. V zápatí je možnost přepínání českého a anglického jazyka.



# Závěr

Cílem tohoto projektu bylo vytvořit webové stránky kapely s možností administrace. K tomuto úkolu jsem využil systém pro správu obsahu WordPress, ve kterém jsem …

…

… Vzhledem k tomu, že je WordPress nejrozšířenější CMS, bývá často napadán a můžou tak nastat potíže se zabezpečením. Další nedostatek je pomalá rychlost načítání při používání tohoto systému.

…

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] Systémy pro správu obsahu (Wikipedie) [online]. 2018 [cit. 2018-8-1].

< <https://cs.wikipedia.org/wiki/Syst%C3%A9m_pro_spr%C3%A1vu_obsahu> >

[2] WordPress (Wikipedie) [online]. 2018 [cit. 2018-11-13].

< <https://en.wikipedia.org/wiki/WordPress> >

[3] MySQL (Wikipedie) [online]. 2018 [cit. 2018-9-11].

< <https://cs.wikipedia.org/wiki/MySQL> >

[4] PHP (Wikipedie) [online]. 2018 [cit. 2018-9-11].

< https://cs.wikipedia.org/wiki/PHP >

[5] WordPress manuál (WPBeginner) [online]. 2018 [cit. 2018-9-11].

< <https://www.wpbeginner.com/> >