

# Moderní UI webové aplikace pomocí Scroll Reveal efektů

Bc. Jan Benedík

---

Diplomová práce  
2015



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2014/2015

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan Benedík**  
Osobní číslo: **A13416**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **prezenční**

Téma práce: **Moderní UI webové aplikace pomocí Scroll Reveal efektů**  
Téma anglicky: **Modern UI Web Applications Using Scroll Reveal Effects**

Zásady pro vypracování:

1. Nastudujte problematiku Scroll Reveal efektů a popište základní principy.
2. Vyhledejte a srovnejte dostupné knihovny pro tvorbu Scroll Reveal efektů.
3. Pomocí vybrané knihovny implementujte pro open-source platformu Wordpress plugin, umožňující použití Scroll Reveal efektů na uživatelské úrovni.
4. Vytvořte zkušební web s demonstrací jednotlivých efektů, obsahující také dokumentaci pluginu.
5. Diskutujte možnosti či omezení vytvořeného řešení z hlediska nasazení v mobilních prohlížečích.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. ODELL, Den. **JavaScript: průvodce programováním ajaxových aplikací**. Vyd. 1. Brno: Computer Press, 2010, 368 s. ISBN 978-80-251-2733-9.
2. THAU, Dave. **Velký průvodce JavaScriptem: tvorba interaktivních webových stránek v praxi**. 1. vyd. Praha: Grada, 2009, 516 s. ISBN 978-80-247-2211-5.
3. KUDLÁČEK, Lubomír. **WordPress: podrobný průvodce tvorbou a správou webů**. Vyd. 1. Brno: Computer Press, 2010, 261 s. ISBN 978-80-251-2734-6.
4. CASTRO, Elizabeth a Bruce HYSLOP. **HTML5 a CSS3: názorný průvodce tvorbou WWW stránek**. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 978-80-251-3733-8.
5. **ScrollReveal.js: Easily reveal elements as they enter the viewport**. [online]. 2014 [cit. 2015-02-02]. Dostupné z: <http://scrollrevealjs.org/>
6. **Infinite Scroll** [online]. 2008 [cit. 2015-02-02]. Dostupné z: <http://www.infinite-scroll.com/>

Vedoucí diplomové práce:

**Ing. Radek Vala**

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

**6. února 2015**

Termín odevzdání diplomové práce:

**15. května 2015**

Ve Zlíně dne 6. února 2015



doc. Mgr. Milan Adámek, Ph.D.

*děkan*



L.S.



doc. Mgr. Roman Jašek, Ph.D.

*ředitel ústavu*


### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně 15.5.2015

.....  
  
.....  
podpis diplomanta

## **ABSTRAKT**

Diplomová práce se zabývá velmi aktuálním tématem z oblasti webdesignu, jímž je tvorba moderního interaktivního uživatelského prostředí (UI - User Interface) pomocí deklarativních animací - tzv. Scroll Reveal efektů. Teoretická práce má za úkol zmapovat vhodné, zejména open-source JavaScriptové frameworky, jež jsou vhodné pro aplikaci požadovaných efektů. Výstupem praktické je demo ukázka možností jednotlivých frameworků a dále integrace do open-source CMS systému WordPress.

Klíčová slova: Javascript, CSS3, HTML5, webdesign, scroll reveal, WordPress

## **ABSTRACT**

This thesis deals with a very current topic in the field of webdesign, which is the creation of a modern interactive user interface (UI - User Interface) using declarative animation - so called Scroll Reveal effects. Theoretical work is to find out a suitable, especially open-source JavaScript frameworks, which are suitable for applying the desired effects. The output of the practical part is a demonstration of options for each framework and also integration into the open-source CMS WordPress.

Keywords: Javascript, CSS3, HTML5, webdesign, scroll reveal, WordPress

Rád bych poděkoval svému vedoucímu Ing. Radku Valovi za cenné rady a připomínky, a také za čas, který mi věnoval při tvorbě diplomové práce.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 SCROLL REVEAL EFEKTY</b> .....	<b>11</b>
1.1 PARALAX .....	11
1.2 POUŽÍVANÉ ANIMACE.....	12
1.2.1 Pokročilé animace .....	13
1.3 OBECNÉ ŘEŠENÍ ANIMACÍ .....	13
1.3.1 HTML5 .....	14
1.3.2 CSS3.....	15
1.3.3 JavaScript .....	19
<b>2 SROVNÁNÍ DOSTUPNÝCH KNIHOVEN</b> .....	<b>21</b>
2.1 SCROLLREVEAL.JS .....	21
2.2 WOW.JS .....	22
2.3 INFINITE SCROLL.....	22
2.4 SKROLLR.JS .....	23
<b>3 REDAKČNÍ SYSTÉM WORDPRESS</b> .....	<b>25</b>
3.1 ADMINISTRACE OBSAHU.....	25
3.1.1 Příspěvky .....	25
3.1.2 Média.....	26
3.1.3 Stránky .....	26
3.1.4 Komentáře .....	26
3.2 ADMINISTRACE VZHLEDU.....	27
3.2.1 Šablony.....	27
3.2.2 Přizpůsobit.....	27
3.2.3 Widgety .....	27
3.2.4 Menu .....	27
3.2.5 Záhloví a pozadí .....	28
3.2.6 Editor.....	28
3.3 ADMINISTRACE PLUGINŮ.....	28
3.3.1 Přehled pluginů .....	28
3.3.2 Instalace pluginů .....	28
3.3.3 Editor.....	28
3.4 ADMINISTRACE UŽIVATELŮ.....	29
3.5 NÁSTROJE A NASTAVENÍ .....	29
<b>4 TVORBA PLUGINU PRO CMS WORDPRESS</b> .....	<b>30</b>
4.1 WORDPRESS API .....	30
4.2 INFORMAČNÍ HLAVIČKA A PUBLIKACE PLUGINU .....	31
4.2.1 Licence .....	32

4.3	AKTIVACE PLUGINU .....	32
4.4	PŘIDÁVÁNÍ AKCÍ .....	32
4.5	ZNAČKY ŠABLON.....	33
4.6	UKLÁDÁNÍ DAT DO DATABÁZE .....	34
4.7	REGISTRACE A POUŽITÍ SKRIPTŮ.....	34
<b>II</b>	<b>PRAKTICKÁ ČÁST .....</b>	<b>35</b>
<b>5</b>	<b>SCROLLREVEAL.JS WORDPRESS PLUGIN.....</b>	<b>36</b>
5.1	POŽADAVKY NA PLUGIN .....	36
5.2	PŘÍPADY UŽITÍ.....	36
5.3	SOUBOROVÁ STRUKTURA .....	37
5.4	PHP SIMLE HTML DOM PARSER .....	38
5.5	VYUŽITÍ WORDPRESS API .....	38
5.5.1	Používání databáze.....	38
5.5.2	Přidání pluginu do administračního menu .....	39
5.5.3	Výpis použitých efektů .....	41
5.5.4	Přidávání a editace efektů .....	42
5.5.5	Konfigurace pluginu.....	44
5.6	POUŽITÉ AKCE A SKRIPTY .....	46
<b>6</b>	<b>TVORBA ZKUŠEBNÍHO WEBU.....</b>	<b>47</b>
6.1	INSTALACE A NASTAVENÍ WORDPRESSU.....	47
6.1.1	Nastavení struktury webu.....	48
6.2	TVORBA ŠABLONY.....	48
6.2.1	Hierarchie šablony .....	49
6.2.2	Bootstrap .....	50
6.2.3	Rozložení stránky.....	51
<b>7</b>	<b>POUŽÍVÁNÍ EFEKTŮ POMOCÍ PLUGINU .....</b>	<b>52</b>
7.1	NASTAVENÍ NOVÉHO EFEKTU .....	52
7.2	NASTAVENÍ KONFIGURACE.....	53
<b>8</b>	<b>PUBLIKOVÁNÍ PLUGINU .....</b>	<b>55</b>
8.1	INFORMAČNÍ HLAVIČKA .....	55
8.2	SOUBOR README.TXT .....	55
<b>9</b>	<b>VYUŽITÍ NA MOBILNÍCH ZAŘÍZENÍCH .....</b>	<b>57</b>
9.1	MÁ VÝZNAM EFEKTY NA MOBILNÍCH ZAŘÍZENÍCH VŮBEC VYUŽÍVAT? .....	57
9.2	JE MOŽNÉ ZPOŽDĚNÍ ÚPLNĚ ZAKÁZAT?.....	57
9.3	OVLIVŇUJE POČET SPOUŠTĚNÝCH EFEKTŮ PLYNULOST POSOUVÁNÍ? .....	58
9.4	MOHOU EFEKTY NARUŠIT LAYOUT STRÁNKY? .....	58
	<b>ZÁVĚR .....</b>	<b>59</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>60</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>63</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>64</b>
	<b>SEZNAM TABULEK.....</b>	<b>65</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>66</b>

## ÚVOD

S nástupem nových verzí již zaběhnutých webových technologií a jazyků jakými jsou dnes HTML5 a CSS3, jež dovolují webovým vývojářům vytvářet vzhled stránek dle libosti, je na front-end stránek kladen stále větší důraz. Jelikož zmíněné kaskádové styly přinesly možnost animací, začaly vznikat různé knihovny jak tomto jazyce, tak v jazyce Javascript, které mají práci s animacemi zjednodušovat. Tím se začala zvyšovat i interaktivita webového obsahu, která je dnes více a více žádaná. Usnadnění pak právě spočívá v deklarativních animacích, mezi něž patří i tzv. scroll reveal efekty. Podstatou scroll reveal efektů je právě závislost na posouvání stránek díky čemuž můžeme návštěvníkům poskytnout poutavý vizuální zážitek prostým prohlížením obsahu. Pro samotné vývojáře je důležitý způsob implementace efektů jednotlivým blokům stránky, a v neposlední řadě rychlost či plynulost animací a stejně tak rychlost načítání knihoven. Budeme-li vycházet z dostupných a v podstatě zaužívaných principů animací, které byly deklarovány už v rámci knihovny jQuery před několika lety, lze předpokládat, že škály efektů nabízené jednotlivými knihovnami mohou být vcelku podobné. Co se týče knihovny jQuery, bývá občas označována jako pomalá, což mohlo mít vliv na to, že jsou právě vzniklé knihovny nezávislé na této knihovně.

Scroll reveal efekty lze implementovat i v rámci redakčních systémů, kterých je dnes velké množství a jsou oblíbené a žádané. Statistikám dnes vévodí redakční systém WordPress. Zde se nabízí možnost implementace pomocí pluginu nabízející i běžným uživatelům implementaci do šablon webu. Podmínkou je samozřejmě přívětivé ovládání či nastavení takového pluginu, aniž by muselo být zasahováno přímo do kódu. Takovéto předpoklady by mohl splnit plugin, který by poskytoval uživateli formulář se selekcí určitého objektu na stránce a možnostmi nastavení atributů animace.

## **I. TEORETICKÁ ČÁST**

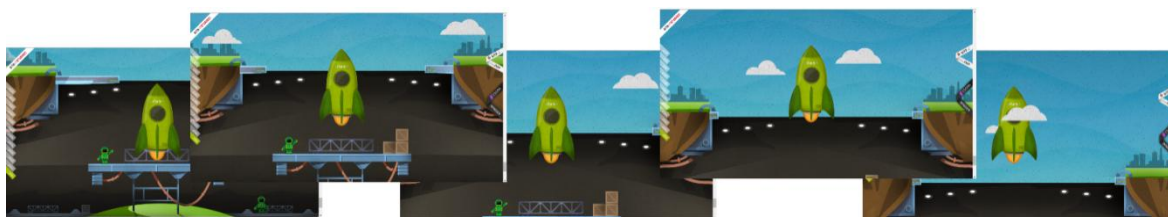
## 1 SCROLL REVEAL EFEKTY

Jak napovídá anglický název těchto efektů, jsou závislé na tzv. scrollování či posouvání stránky. Efekty nebo spíše animace jsou tedy v tomto případě spouštěny v momentě, kdy se posune se stránkou a v okně prohlížeče se objeví zatím skrytý objekt, se kterým je animace provázána. Přístupů, jak provázat tyto animace je hned několik a nejčastěji bývá použita kombinace kaskádových stylů a obsluhy událostí pomocí jazyka Javascript. Rozhodujícím aspektem volání události, je dosažení odsazení od počátku stránky, na kterém většinou skrytý objekt vstupuje společně s animací do prostředí okna. Scroll reveal efekt tak skutečně udělá stránku zajímavější. Před samotným popisem používaných animací u scroll revealingu, bude přiblížen podobný typ efektu, který je se scroll revealing efekty úzce spjatý.

### 1.1 Paralax

Řeč je o paralax efektu. Nesmíme si ho však plést s efektem, který je možné nalézt v optické fyzice. Námí popisovaný efekt dnes patří k běžně užívaným vylepšením vizuální stránky webu. Zájem o tuto metodu zpracování webu má za poslední roky strmý nárůst, o čemž svědčí spousta takto vzniklých webů a dostupných šablon, nebo také Javascriptových knihoven, které nám umožňují si takovýto web vytvořit. Na webu se pak můžeme setkat i s typy webů, které jsou kombinací paralaxu a scroll reveal efektů.

Samotný efekt spočívá v použití dvou a více vrstev, které se při posouvání stránky pohybují nerovnoměrnou rychlostí jak vůči sobě tak vůči oknu prohlížeče samotnému, což lze vidět na demonstračním obrázku 1-1, který by nám měl přiblížit chování stránky. Znárodná raketa je při posouvání stránky stále na středu a mění se pouze ubíhající pozadí vrstva s mraky, které se přisouvají ze strany. Vrátime-li se k paralaxu, ten jako samotný měl uplatnění už v prvních počítačových hrách, kde byl použit při horizontálním posunu. Tyto aspekty pak vypovídají, že paralax bývá použit pro „vyprávění“ nějakého příběhu, který má uchvátit návštěvníka stránky.



Obrázek 1-1: Demonstrace použití paralaxu na webu intacto10years.com.

Nicméně se paralax setkává s několika negativy. Jedním z nich je použití mnoho obrázkových vrstev jako pozadí, což hodně ovlivňuje jak rychlost načítání obsahu, tak i plynulost prohlížení zvláště u slabších zařízení. Dalším negativem je skutečnost, že takovéto grafické návrhy jsou realizovány jako sigle page aplikace, ve který se většinou nachází pouze málo prvků zvyšující SEO. S tímto problémem se můžeme ale také setkat i u webů se scroll reveal efekty.[1]

## 1.2 Používané animace

Jak už bylo zmíněno, používané animace u scroll reveal efektů vychází z již dřívějších a dnes už běžně používaných animací. V následujícím výčtu atributů animací společně s krátkým popisem nalezneme základní možnosti animací nacházejících se v javascriptových knihovnách, které si popíšeme později. Jelikož se jedná o cizojazyčné knihovny, budou názvy parametrů i animací uvedeny v závorce v anglickém jazyce. Nicméně v našem případě můžeme animaci elementů stránky chápat jako předem definovaný pohyb právě podle následujících parametrů:

- **Vstup (Enter)** – definuje z které strany má objekt vstupovat na scénu, na výběr máme klasicky ze všech čtyř stran vertikální a horizontální osy.
- **Posun (Move)** – říká objektu, po jak dlouhé dráze se má pohybovat. Většinou udávána v pixelech.
- **Doba trvání (Over)** – určuje jak dlouhý časový úsek v sekundách má animace trvat.
- **Zpoždění (Wait)** – je atribut, který může spuštění animace oddálit, čímž lze dosáhnout například posloupnosti spouštění efektu na stejné horizontální úrovni stránky.

Následující parametry otočení mají jako základní jednotku úhlové stupně. Rozdíl jejich průběhu je pouze v ose otáčení objektu. Změnou polarity stupňů pak můžeme ovlivňovat směr otáčení.

- **Horizontální otočení (Flip)** – v tomto případě se objekt otáčí okolo osy X a jelikož je ovlivňována hodna i v ose Z, která pro nás bývá prostorová, nemusí být animace na první pohled zřetelná.
- **Vertikální otočení (Spin)** – definuje otáčení okolo osy Y a v podstatě stále přetrvává problém s osou Z, jako v horizontálním otočení.

- **Zatočení (Roll)** – je ze všech zatočení nejvíce zřetelné, neboť kvůli otáčení kolem osy Z se během jeho průběhu mění parametry osy X i Y, které pro nás představují plošný obraz.
- **Změna měřítka (Scale)** – zajišťuje animaci rozpínání a smršťování, jednotkou jsou procenta, která říkají, o kolik se má objekt roztáhnout či smrštit vůči původní velikosti. Aby bylo jasné, jestli se má objekt roztáhnout nebo smrštit je nutné definovat směr této transformace.
- **Změna průhlednosti (Opacity)** – jednoduchým nastavením hodnoty v rozsahu 0 až 1 v podstatě nastavujeme průhlednost objektu.

### 1.2.1 Pokročilé animace

Pokročilé animace vycházejí z výše uvedených základních animací a jedná se prakticky o jejich kombinace. Každopádně si přiblížíme ty nejčastější.

- **Skákání (Bounce)** – principiálně můžeme tuto animaci chápat jako prudkou, opakovanou změnu ve většině případu parametrů v ose Y. Obecně je u této animace měněn jak pohyb, tak změna měřítka ve stejné ose. Dalším parametrem pak bývá i směr, že kterého objekt „přiskočí“ na scénu a stejně jako u většiny animací může být ovlivněn časem.
- **Vpadnutí (Fade)** – jedním s hlavních parametrů této animace je směr vstupu, který společně s použitím buď změny měřítka, nebo průhlednosti vytvoří plynulé objevení objektu na stránce.
- **Rotace (Rotation)** – z výše uvedených otáčení zde bývá nejčastěji použito otočení kolem osy Z, které bývá opět kombinováno s nějakým vstupem, změnou průhlednosti a nějakým časovým průběhem.

Z těchto animací je patrné, že lze kombinací různých atributů vytvořit nesčetné množství efektů. Proto není divu, že jsou pohyblivé prvky fron-tendu stránek více a více populární.

## 1.3 Obecné řešení animací

Řešením zde myslíme použití webových technologií, jak už bylo několikrát vzpomínáno. Animace však nevychází pouze z knihoven, ale ze všech front-endových jazyků. Tedy je zde kromě Javascriptu nutně zahrnuto i HTML5 a CSS3. Proto je důležité si tyto technologie přiblížit.

### 1.3.1 HTML5

Tento jazyk lze bez pochyby považovat za stavební kámen webových aplikací. Strukturu webu pak tvoříme sémanticky správným zápisem tzv. HTML tagů, které byly v HTML ve verzi 5 rozšířeny. Historie vývoje v této verzi je vcelku pestrá a kdy se vývoj jazyka se v roce 2006 rozšířil z W3C konsorcia na skupinu WHATWG a své definitivní ustanovení získal 28. října 2014. [2]

Podstatným zlepšením oproti předešlým verzím je zavedení nových sémantických tagů, se využívají ke zlepšení především významové přehlednosti kódu. Z části vychází z praktik používaných v XML, které je z velké části HTML podobné, a kde jednotlivé tagy bývají pojmenovány uživatelem podle příslušného významu. Proto se můžeme setkat novými tagy jako například *footer*, jež by měl vymezovat oblast patičky webu nebo jednotlivé sekce v obsahu. Sekce je také novým sémantickým tagem *section*, jejíž využití je především ve vymezení obsahu stránky, který by mohl být například jedním záznamem v databázi a může obsahovat právě svoji hlavičku, obsah a patičku. [3]

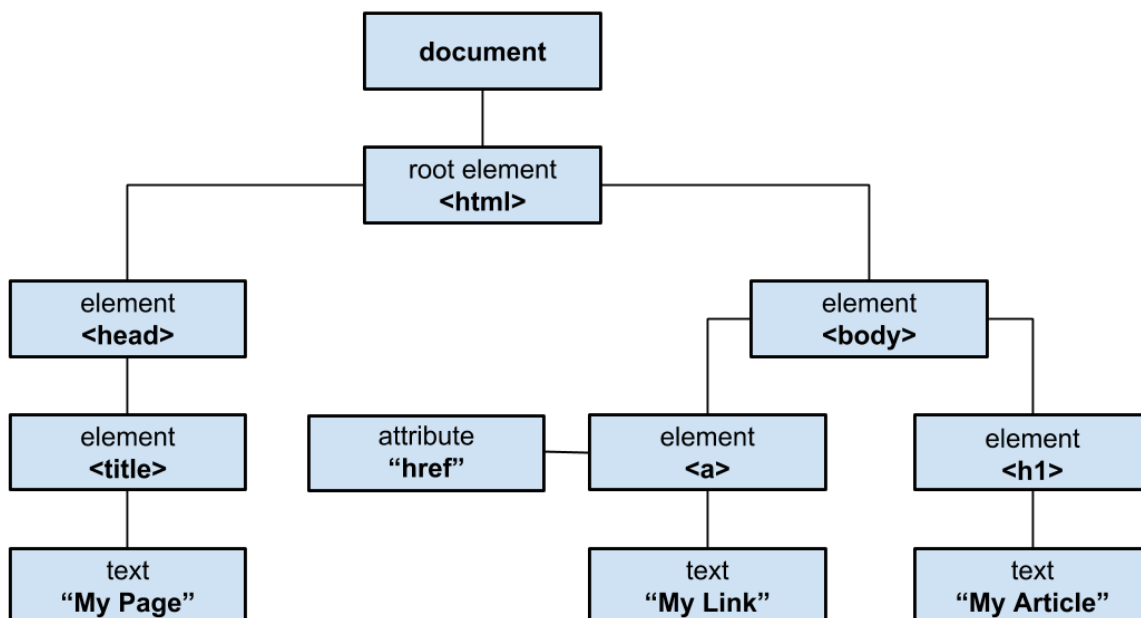
Dalšími novými tagy jsou i *canvas* a *svg*, které svým použitím posouvají webové aplikace téměř na úroveň těch desktopových. Canvas se běžně využívá pro kreslení pomocí Javascriptu, kdežto SVG používá definici objektů pomocí dalších tagů, které vychází z XML.

Poslední užitečnou novinkou je i rozšíření atributů u formulářových tagů, kde mezi ty nejdůležitější určitě patří *placeholder*, u kterého můžeme jeho hodnotou nastavit textové kolonce zašedlý našeptavač typu „Zde zadejte váš e-mail“ apod. Dalším je bezhodnotový atribut *required*, jehož použitím vymezujeme rozdíl mezi důležitými a nedůležitými údaji.

S novou verzí se také objevila nová HTML API, která rozšiřují použití webu za pomoci Javascriptu. Prvním z nich je *Geolocation*, které lze využít k určení přibližné pozice v zeměpisných jednotkách a jejich následné použití v mapách od společnosti Google. Dalším je *Local Storage*, které lze využít jako dočasné uložení pro webové aplikace, například pro uložení nějakého osobního nastavení. Posledním zajímavým API je Drag and Drop, které umožňuje uchopení elementů na stránce a jejich přetažení na novou pozici.

Zmíněné tagy představují elementy či bloky na webové stránce, které jsou pak animovány buď přímo kaskádovými styly, jejichž popis je níže nebo selekcí v rámci uvedených javascriptových knihoven. Uspořádání těchto tagů a jejich zanořováním vzniká strom, který bývá označován DOM, čehož se právě využívá při selekci elementů. Platí zde jednodu-

ché pravidlo, že všechno je uzel. I samotný dokument, který se po načtení do prohlížeče stává kromě jiného objektem, je uzel a všechny podružené HTML elementy jsou potomky tohoto uzlu, stejně tak tributy HTML elementu jsou uzly a text elementu je taktéž uzlem. Pro lepší pochopení stromu DOM je uveden následující obrázek 1-2. [4]



Obrázek 1-2: Stromová struktura DOM.

### 1.3.2 CSS3

Kaskádové styly byly právě průlomovou technologií, co se týče animací na webu bez nutnosti flashe či pohyblivých obrázků. Animace přišli právě se zveřejněním třetí verze kaskádových stylů, jež byla uvedena 14. Dubna 2000. Nicméně se zde vyskytovaly problémy kompatibility s verzemi prohlížečů. Dnešní známé prohlížeče však již tuto verzi plně podporují. Novinkami v třetí verzi byla i spousta dalších vlastností, kterými bylo možné web zkrášlovat. A to předně možnost zabolených rohů, stínů elementu a gradientu barev. [5]



Obrázek 1-3: Loga HTML5 a CSS3.

Průlomovými vlastnostmi přinášejícími možnost animací a efektů je *transition*, *transform* a *animate*, kterými jsou právě definovány změny proporcí objektů, jež jsou vůdčími znaky animaci. Tyto vlastnosti mají jak spoustu možných nastavení, tak i rozdílné metody zápisu. V následujícím výčtu jsou uvedena jak tato nastavení, tak i krátké příklady použití.

- **Přechod (Transition)**, již je možné nastavit u elementů, které mají definované pseudotřídy *:hover* či *:focus*. Této vlastnosti lze pak nastavit několik parametrů, kde první dva jsou nutnou podmínkou pro provedení změny.
  - Která vlastnost se má při provádění měnit. Běžně to bývá změna velikosti nebo změna barvy písma a pozadí.
  - Následně čas trvání přechodu uváděný v sekundách.
  - Transition-timing-function, kterou lze specifikovat rychlost průběhu například na začátku, či na konci přechodu.
  - Transition-delay, jež opět v sekundách určuje čas zpoždění přechodu. [6]

Těchto parametrů lze najednou nastavit hned několik díky zkrácenému zápisu, jehož se běžně používá. Pro lepší představu je uveden následující příklad. Element daný tagem `div`, nastaven se zeleným pozadím a šířkou 100 pixelů, bude mít nastavenou první změnu závislou právě na tomto pozadí, která bude trvat 2 sekundy a bude mít lineární průběh. Další změnou pak bude šířka, která bude trvat 3 sekundy a bude mít na začátku pomalejší průběh:

```
div {  
  
    background: green;  
  
    height: 100px;  
  
    width: 100px;  
  
    transition: background 2s linear, width 3s ease-in;  
  
}
```

Aby se změny provedly, je nutné nějakou vyvolat pomocí pseudotřídy v následujícím zápisu:

```
div:hover {  
  
    background: red;  
  
    width: 150px;}
```

- **Přeměna (Transform)** na rozdíl od změn nemá pouze jednoduché nastavení parametrů, ale využívá nečetné množství metod, které dovolují transformaci ve dvou či třech rozměrech a parametry přeměny se nastavují až těmto metodám. Je nutné říci, že samotná transformace nijak animována není, proto se používá ve spojení s *transition*. Pro transformace ve dvou rozměrech slouží čtyři základní metody:
  - **translate** posune element o nastavenou šířku a výšku,
  - **rotate** pak otáčí element v rámci obou os o nastavený úhel,
  - **scale** mění poměr šířky a výšky původního elementu,
  - **skew** jako poslední nastavuje skosení elementu.

Některé tyto metody mají ještě své modifikace, které dovolují element ovlivňovat pouze ve vybrané ose. Modifikace metody *rotate* pak umožňuje transformaci v prostoru, tedy ve třech rozměrech. Tyto metody v podstatě představují animace, které byly uvedeny výše. Níže uvedené rozšíření kódu je příkladem, jak lze použít transformaci v rámci nějaké změny. Samotné transformace jsou totiž samy o sobě pouze statické a bez začlenění právě do nějaké změny není ani přeměna dynamická. Již definovanému bloku je pak nastaven během změny ještě rotace. [7]

```
div {  
  
    background: green;  
  
    height: 100px;  
  
    width: 100px;  
  
    transition: background 2s linear, width 3s ease-in,  
    transform 2s;  
  
}
```

Do pseudotřídy pak stačí doplnit požadovanou přeměnu.

```
div:hover {  
  
    background: red;  
  
    width: 150px;  
  
    transform: rotate(180deg);  
  
}
```

- **Animace (Animation)** umožňují provádět už plnohodnotné animace elementů stránky bez použití Javascriptu. Nicméně je jejich zápis už složitější, co se obsáhlosti kódu týče. Podoba animace je zde už zakódována v tzv. keyframu, který je rozdělen na sadu příkazů definujících v různých časových úsecích vlastnosti daného elementu, kdy bývají tyto úseky prezentovány procenty ušlého času celé animace. Tento keyframe má pak své jméno a v podstatě připomíná funkci, která je pak volána pod vlastností *animation* přímo ve stylovém zápisu daného elementu. Vlastnost *animation* má pak další parametry, které upravují průběh animace. [8]
  - **Animation-name** není nic než název keyframu, kterým se má animace řídit.
  - **Animation-duration** definuje čas trvání animace a s předchozím parametrem je důležitý pro provedení animace.
  - **Animation-delay** jak název napovídá umožňuje posunout spuštění animace.
  - **Animation-iteration-count** rozšiřuje animace o počet opakování a je možné mu kromě čísla nastavit například i nekonečné opakování.
  - **Animation-direction** umožňuje například provádění animace v opačném směru.
  - **Animatin-timing-function** stejně jako u změny nastavuje časový průběh animace na začátku či na konci jejího průběhu.

Jako u každé vlastnosti kaskádových stylů, tak i u animace lze všechny parametry zapisovat ve zkráceném zápisu, který je použit v následujícím příkladu definice animací. Opět je dán div, který bude během své animace měnit pozici a barvu ve čtyřech časových úsecích.

```
@keyframe klesani {  
    0%    {background-color: red; top: 0px}  
    25%   {background-color: yellow; top: 50px}  
    50%   {background-color: green; top: 100px}  
    100%  {background-color: blue; top: 150px}  
}
```

Pak stačí takto nadefinovaná keyframe zavolat ve vlastnosti *animation* u daného divu, kde animaci nastavíme ještě pomalý začátek a postup animace oběma směry s prodlevou a nekonečným opakováním.

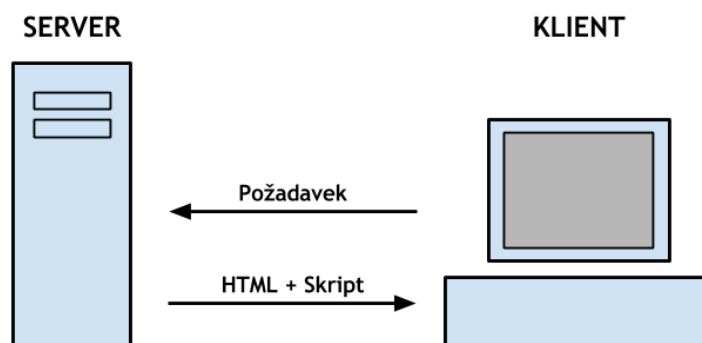
```

div {
    height: 100px;
    width: 100px;
    animation: klesani 4s ease-in 2s infinite alternate;
}

```

### 1.3.3 JavaScript

Tento programovací jazyk patří mezi klientské skripty. Klientský skript znamená, že program je prováděn na straně klienta, tedy v prohlížeči, na rozdíl od PHP a podobných jiných serverových jazyků, jež jsou prováděny na straně serveru. Od roku 1995 slouží vývojářům webu jako nepostradatelný nástroj při tvorbě interaktivních webových stránek. Některé z vysoce interaktivních aplikací jsou někdy označovány jako RIA aplikace. RIA tak nahrazují běžné desktopové aplikace a jejich provoz je zcela v rámci prohlížeče. [9][10][11]



Obrázek 1-4: Princip klientského skriptu.

V rámci efektů či animací bývá JavaScript použit k obsluze událostí. Touto událostí, jak už bylo zmíněno, může být právě posunutí stránky, což je podstatou právě scroll reveal efektů. JavaScript však disponuje celou řadou obsluhy událostí od kliknutí na odkaz, stlačení klávesy až po změnu velikosti okna. V následující tabulce jsou pro představu uvedeny nejčastěji používané odchyťované události myši. [12][13]

Název události	Popis události
onClick	Kliknutí na element stránky
onMouseOver	Myš se nachází nad elementem stránky
onMouseOut	Myš opustí element stránky

onMouseMove	Myš je v pohybu nad elementem stránky, pokud se zastaví, nevykonává se obsluha
onMouseDown	Moment, kdy je tlačítko myši stlačeno a drženo nad elementem stránky
onMouseUp	Moment, kdy je tlačítko upuštěno (ihned po tomto momentu se odchyťává událost onClick)

Tabulka 1-1: Nejčastěji využívané události v JavaScriptu.

Odchyťování událostí je v JavaScriptu hned několik a jsou uvedeny v následujících příkladech. Prvním je přidání atributu s hodnotou obslužné funkce či přímého skriptu do HTML tagu elementu.

```
<div onclick="alert('klikli jste na div')">
```

Další možností je použití v rámci skriptu a to hned dvojím způsobem. Ten první využití metody *onclick* na přímé instanci elementu. Způsob zápisu lze vidět na následujícím příkladu.

```
element.onclick = function() {  
    alert('klikli jste na div');  
};
```

Další možností je využít tzv. Event Listeneru.

```
element.addEventListener("click", function() {  
    alert('klikli jste na div');  
});
```

## 2 SROVNÁNÍ DOSTUPNÝCH KNIHOVEN

Využívání hotových knihoven je v dnešní době součástí práce téměř každého vývojáře. K programování knihoven je často využíván objektově orientovaný JavaScript, díky kterému lze celou funkcionalitu knihovny zahrnout do jediné třídy. Takovéto knihovny mají ve většině případů vývoj urychlit. Stejně tak vzniklo několik knihoven usnadňující použití scroll reveal efektů. Každá knihovna pak poskytuje jiné možnosti a různé úrovně implementace, jejichž porovnání by mělo pomoci vybrat tu správnou pro použití na webu.

### 2.1 ScrollReveal.js



Obrázek 2-1: Grafika spojená se ScrollReveal.js.

Knihovna je open-source projektem kalifornského designéra a vývojáře Juliana Lloyda, který ji poskytuje pod licencí MIT a zdrojový kód společně s přehlednou a jednoduchou dokumentací je k dispozici na GitHubu. [14]

Implementace této javascriptové knihovny se provádí inicializací do nové proměnné objektu *window*, který je součástí standartu BOM. Tento standart lze jednoduše zaměnit s DOM. Nicméně skutečnost je taková, že DOM reprezentuje objekt *document*, který je vnitřním proměnnou objektu *window*, jež reprezentuje prohlížeč. Na následujícím příkladu je pak uvedený zápis inicializace knihovny ScrollReveal.js [15]

```
window.sr = new scrollReveal();
```

To umožňuje používání *data* atributu v kterémkoliv začátečním tagu elementu na stránce, který chceme animovat. Bez zadání parametrů je použito defaultní nastavení. Parametry se pak zadávají až na výjimky v páru klíč/hodnota a jsou obvykle odděleny čárkou. Možné použití demonstruje následující příklad

```
<div data-sr="move 50px, scale down 20%, over 2s, reset"> Baz  
</div>
```

Množství parametrů a hlavně uživatelské nastavení hodnot umožňuje rozsáhlé použití efektů. Další možností jak přizpůsobit použití knihovny vlastním potřebám je vlastní konfigurace, kterou lze předat jako parametr inicializaci. Jednou z přínosných vlastností konfigurace je možnost zakázání efektů na mobilních zařízeních.

## 2.2 WOW.js

Tato knihovna si je s předchozí ScrollReveal.js projektem v mnohém podobná. Také je projektem jedné osoby a to francouzského vývojáře Matthieu Aussaguela, který ji opět pod MIT licenci poskytuje volně dostupnou na GitHubu.

Použití je zde však trochu rozdílné. Podstatou je použití cizí knihovny kaskádových stylů Animate.css, z níž WOW.js spouští předem nastavené animace. Další odlišností je způsob provázání HTML s knihovnou. Elementům jsou přidávány místo atributu další třídy. Jednou z tříd je *wow*, kterou nastavujeme element jako „animovatelný“ knihovnou WOW.js s defaultním nastavením. Další rozšiřující třídou je určitý název animace z knihovny Animate.css, jak je tomu v následujícím příkladu kódu. Použití Animate.css však není podmínkou a v konfiguraci lze změnit používanou knihovnu animací na kteroukoli jinou.

```
<section class="wow slideInLeft" data-wow-duration="2s" data-wow-delay="5s"></section>
```

V příkladu výše lze vidět další *data* atributy, kterými můžeme napřímo měnit některé parametry animace jako je délka, zpoždění, opakování nebo odsazení animace. Další možností je napsat tato nastavení v konfiguraci volání knihovny, kde lze nastavit opět spouštění na mobilních zařízeních nebo asynchronní provádění animací. [16]

## 2.3 Infinite Scroll



Obrázek 2-2: Logo Infinite Scroll.

Infinite Scroll se liší od předchozích dvou knihoven ve své funkčnosti. Knihovnu, jejíž vývoj je pozastaven, vyvíjel Paul Irish a má sloužit k zobrazování či načítání zatím skrytého obsahu při rolování stránky. Tohoto lze využít u obsahu, který má kvůli své četnosti

nastaveno stránkování, zvláště pak u příspěvků na blogu apod. Proto se tato knihovna dočkala také pluginu pro redakční systém WordPress.

Kromě natavení, zda se má nebo nemá zobrazení dalšího obsahu animovat, nedisponuje knihovna nastavením průběhu této animace. [17]

## 2.4 Skrollr.js

Javascriptová knihovna německého vývojáře Alexandra Prinzhorna je oproti předešlým knihovnám o něco náročnější na použití, protože vyžaduje pokročilejší znalosti HTML/CSS. Metoda, jakou je tato knihovna implementována do kódu stránky, je podstatně odlišná a v mnoha případech se knihovna využívá spíše pro tvorbu paralax efektů, které byly popsány výše. Nicméně nalezneme i případy užití pro scroll reveal efekty. Možnosti použití této knihovny pak rozšiřují i dostupné pluginy přímo od vývojáře, nebo pluginy třetích stran. U knihovny existuje i podpora pro použití v rámci javascriptového frameworku Require.js, kde ji lze inicializovat jako nový modul. [18][19]

Inicializace se provádí v podstatě stejně jako u ScrollReveal.js, ale s malým rozdílem, že se instance ukládá do obyčejné proměnné, jako je tomu v následujícím příkladu. Inicializaci pak lze předat jako parametr i vlastní konfiguraci knihovny. Kromě inicializační metody jsou k dispozici i další metody, kterými lze použití ještě více optimalizovat.

```
var s = skrollr.init();
```

Jak už bylo zmíněno na začátku této podkapitoly, pro samotné použití v rámci je už potřeba znát lépe používání kaskádových stylů, protože funkce této knihovny je založena na provádění tzv. keyframů (je tu jistá podobnost s keyframy u CSS animací), kterých se má docílit na přesně definovaných pozicích a Skrollr provádí interpolaci mezi těmito pozicemi. Tyto pozice jsou zapisovány jako *data* atributy přímo do tagů. Pro lepší pochopení je níže uveden příklad.

```
<div
  data-0="background-
color:rgb(0,0,255);transform:rotate(0deg);"
  data-500="background-
color:rgb(255,0,0);transform:rotate(360deg);"
>WOOOT</div>
```

Pozice jsou zapisovány jako odsazení od začátku dokumentu v pixelech. Tento způsob je označován jako absolutní mód. Knihovna však disponuje i relativním módem, který se řídí odsazením od začátku okna. V předchozím příkladu je jako hodnota *data* atributu použit běžný zápis vlastnosti kaskádových stylů, což navazuje na jejich zmíněnou potřebu znalosti.

### 3 REDAKČNÍ SYSTÉM WORDPRESS

Tento redakční systém je vyvíjen jako open-source projekt pod GNU GPL licencí a v statistikách se řadí k nejužívanějšímu redakčnímu systému. Původně především blogovací systém je dnes používán pro rozsáhlé webové aplikace. Začátky vývoje WordPressu sahají až do roku 2001, v dnešní době však kromě rozsáhlé komunity vývojářů pracuje na WordPressu od roku 2005 především společnost Automattic, která jej začala vyvíjet na jádru b2/cafeblog, původním projektu zakladatele společnosti Matta Mullenwega, z jeho jména je taky odvozen název společnosti. Společnost Automattic mimo jiné vyvíjí i pluginy používané ve WordPressu jako například bbPress pro tvorbu fór. [20] [21]



Obrázek 3-1: Logo redakčního systému WordPress.

Díky své jednoduchosti instalace i užívání je WordPress ideálním nástrojem i pro méně zdatné uživatele, kteří mají zájem vytvářet své osobní weby, blogy či prezentovat malé projekty. Jednoduché použití má zajistit především přehledné menu v administrační části. Proto budou v následujících podkapitolách popsány jednotlivé položky správy jak obsahu, tak samotného webu.

#### 3.1 Administrace obsahu

Kromě menu disponuje WordPress na úvodní stránce administrace také nástěnkou, na které je možné různé její části uspořádat dle vlastní potřeby a mít tak hned po ruce aktuální přehled webu, aktivity či novinky o WordPressu společně se základními informacemi pro práci s WordPressem. Pro správu obsahu slouží v hlavním administračním menu následující čtyři položky popsané níže. [22]

##### 3.1.1 Příspěvky

Tato část menu slouží ke správě příspěvků a její součástí jsou i další podstránky, které rozčleňují správu na následující části.

- **Přehled příspěvků** – nastavitelný seznam všech příspěvků na webu, kde lze příspěvky různě filtrovat.
- **Vytvořit příspěvek** – obsahuje standardně vizuální editor TinyMCE, který patří do kategorie WYSIWYG. Kromě editoru se zde nacházejí různá nastavení příspěvku.
- **Rubriky** – umožňuje správu kategorií, do kterých lze články zařazovat.
- **Štítky** – správa štítků nebo také tagů, které lze přidávat k příspěvkům.

### 3.1.2 Média

Jako každý správný redakční systém i WordPress umožňuje publikovat různá média v příspěvcích. Tato položka menu má dvě podstránky, které umožňují práci s médii.

- **Knihovna médií** – je opět správou různých typů médií a to fotografií, video a audio nahrávek, které je, jak už bylo zmíněno dříve, možné používat v příspěvcích.
- **Nahrát soubor** – disponuje šikovným drag and drop oknem nebo obyčejným dialogovým oknem umožňujícím nahrávání médií do redakčního systému.

### 3.1.3 Stránky

Stránky v rámci WordPresu lze chápat i jako jednotlivé položky menu ve front-endu. Jednotlivým stránkám lze nastavovat jiné nadřazené stránky, čímž lze vytvořit strukturu několika úrovněového menu. Samotná položka v administračním menu je rozdělená na:

- **Přehled stránek** – zde se nachází opět seznam používaných stránek s možností jejich filtrace např. dle data,
- **Přidat stránku** – slouží logicky k přidávání nových stránek. Tato část menu disponuje podobně jako formulář pro přidání příspěvku textovým editorem TinyMCE a také nastavení nové stránky obsahující nastavení nadřazené stránky a případné pořadí v seznamu stránek, které ovlivňuje i rozložení položek menu ve front-endu.

### 3.1.4 Komentáře

Kromě samostatné položky v administračním menu nalezneme odkaz položku komentářů i v nastavení redakčního systému. V nastavení lze měnit globální nastavení komentářů jako například zda se mohou na webu vůbec používat nebo jednotlivé úrovně přístupu. Tak je možné nastavit používání komentářů jen registrovaným uživatelům. Pod položkou v hlavním administračním menu se nachází seznam všech komentářů, které byly napsány k jednotlivým příspěvkům na webu. Tento seznam lze opět různě filtrovat a kategorizovat.

## 3.2 Administrace vzhledu

WordPress nabízí svým uživatelům také možnost editace a přizpůsobení vzhledu front-endu. Pod touto položkou administračního menu lze nalézt několik dalších položek, které budou popsány podrobněji. [22]

### 3.2.1 Šablony

Základní podobu front-endu lze ovlivnit především výběrem šablony. Pokud vývojáři na vzhledu nezáleží, může použít některou ze standartních šablon WordPressu, které jsou k dispozici společně s instalací systému. Jinou možností je použití volných či placených profesionálních šablon. Jelikož se většinou šablony skládají z několika PHP souborů, tak mohou disponovat i jazykovou podporou některých částí obsahu. Takovýchto šablon je na Internetu nepřeberné množství, nicméně je zde možnost použít šablonu vlastní. Šablony lze v této části administrace instalovat, spravovat a aktualizovat.

### 3.2.2 Přizpůsobit

Umožňuje uživatelům více přizpůsobit základní obsahové části front-endu. Mezi základní možnosti patří nastavení názvu a popisu webu, použitých barev, obrázku na pozadí a v záhlaví. V této editační části můžeme na stránku přidávat tzv. widgety do postranního panelu, který bývá součástí základního layoutu webu.

### 3.2.3 Widgety

Widgety jsou v podstatě malými komponentami, které lze přidávat na stránku, a mají svou funkci přínosnou pro rozšíření webu. V administraci lze tyto widgety spravovat stejně jako v nástroji Přizpůsobit s rozdílem toho, že nevidíme okamžitě výsledek úpravy. Mezi widgety lze najít různé rychlé přehledy například rubrik, nejpoužívanějších tagů nebo nejnovějších komentářů. Samotné menu webu je také widgetem a lze tak měnit jeho pozici v panelu.

### 3.2.4 Menu

Ve WordPressu může být i několik menu a v této části je lze jak vytvářet tak i spravovat. Základním nastavením pro každé menu je přiřazení stránek webu, vlastních odkazů nebo rubrik. Dále je možné nastavit menu jako základní nebo aby se každá nově vytvořená stránka přiřadila do menu. Lze zde najít i nastavení umístění v šabloně.

### 3.2.5 Záhloví a pozadí

Tyto dvě položky přesměrovávají uživatele na nástroj Přizpůsobit, kde lze nastavit v záhlaví nebo na pozadí celé stránky obrázek z Galerie médií nebo nahráním z počítače.

### 3.2.6 Editor

V editoru je možné upravovat šablony přímo v kódu, čímž je použitelný spíše pro pokročilé uživatele. Tato část disponuje jednoduchým textovým polem, ve kterém lze upravovat kód souboru ze seznamu uvedeného hned vedle u vybrané šablony. Přívětivou vlastností editoru je selekce vnitřních metod WordPressu do seznamu pod editorem a umožňuje tak dohledat na internetu dokumentaci vybrané metody.

## 3.3 Administrace pluginů

Základním významem pluginů je rozšíření funkcí a použitelnosti jádra WordPressu. Pluginy lze přednostně získávat z přednostně z oficiálního webu WordPressu nebo na stránkách vývojářů těchto pluginů. Tato část administrace pak umožňuje pluginy instalovat a spravovat. [22]

### 3.3.1 Přehled pluginů

Obsahuje již klasicky seznam instalovaných pluginů. Instalované pluginy nejsou automaticky aktivní a tato stránka slouží jak k jejich správě, tak i k aktivaci. Pokud je tedy například nutné kvůli nevydané aktualizaci na chvíli vyřadit plugin z provozu, stačí jej deaktivovat a po aktualizaci znovu použít bez opětovné instalace.

### 3.3.2 Instalace pluginů

V této části administrace je možné pluginy vyhledávat, procházet a instalovat bez nutného stahování balíčků. Instalace balíčků však není vyloučena a odkazem spouštěné dialogové okno dovoluje nahrávání pluginů z počítače.

### 3.3.3 Editor

Editor pluginů je v podstatě úplně stejný jako editor šablon. To znamená, že po vybrání příslušného pluginu lze upravovat plugin přímo v kódu.

### 3.4 Administrace uživatelů

Při použití jádra WordPressu například pro e-shop nebo nějaké fórum musí vývojář webu počítat s více uživateli. Tyto uživatele je nutné nějak kontrolovat a spravovat. Proto má WordPress v administraci i tuto část, která je rozdělená na:

- **Přehled uživatelů** - v této části se nachází opět seznam uživatelů, které lze filtrovat dle úrovně přístupu nebo je možné uživatele také vyhledávat,
- **Vytvořit uživatele** – pomocí formuláře je možné přidávat uživatele vyplněním registračních údajů a přiřazením úrovně přístupu,
- **Profil** – ze seznamu lze v této části upravovat profil vybraného uživatele nebo přímo přihlášený uživatel může upravovat vlastní nastavení.

### 3.5 Nástroje a nastavení

WordPress umožňuje pomocí nástrojů exportovat nebo importovat buď obsahové části WordPressu a nebo různých obsahových částí z jiných systémů a webů jako je například Blogger nebo Tumblr. Tyto nástroje lze také rozšiřovat pomocí pluginů.

Nastavení obsahuje několik částí, pomocí kterých lze nastavit obecné věci především pro obsahovou část redakčního systému. Tato položka v administračním menu je opět rozdělena na několik částí, které jsou popsány v následujícím výčtu.

- **Obecné** – zde se nachází základní nastavení názvu a popisu webu, formáty data a času nebo povolení registrace nových uživatelů.
- **Publikování** – zde se nachází například formátování příspěvků, nastavení výchozí rubriky pro příspěvky nebo nastavení publikování přes email.
- **Zobrazování** – umožňuje nastavování článků na úvodní stránce, maximální počty příspěvků nebo dostupnost pro vyhledávače.
- **Komentáře** – nejobsáhlejší z nastavení umožňuje optimalizaci komentářů pomocí různých bezpečnostních pravidel. Lze zde i nastavit automaticky generované avatary pro nově registrované uživatele.
- **Média** – obsahují krátké nastavení velikosti náhledu obrázku a limity pro nahrávané obrázky, nebo lze nastavit přehledné ukládání medií dle data.
- **Trvalé odkazy** – nachází se zde seznam struktur použitelných pro trvalé odkazy stránek nebo je možné vytvořit strukturu vlastní.

## 4 TVORBA PLUGINU PRO CMS WORDPRESS

Díky dokumentaci veškerého kódu se do vývoje samotného WordPressu může zapojit kterýkoliv programátor. Rozsáhlá dokumentace dovoluje také programátorům upravovat jeho části podle vlastních představ. Za tímto účelem vznikají právě pluginy. Při vývoji pluginů mohou programátoři využívat rozsáhlé API WordPressu, které se nachází právě ve zmiňované dokumentaci. Application Programming Interface, ve zkratce API, je výraz používaný v souvislosti se sbírkou tříd, funkcí či protokolů nějaké knihovny, které mohou programátoři využívat. Stejně tak API určuje, jakým způsobem jsou volány některé části knihovny volány ve zdrojovém kódu programu. [23]

### 4.1 WordPress API

Toto API se skládá z několika dalších API, která mají svá označení a funkci. Pro různé účely je možné využít níže uvedené části, které mohou být spolu provázány a v rámci jednoho pluginu jich lze použít několik nezávisle na sobě. [24]

- **Dashboard Widget API** – nachází se zde metody pro přidávání a obsluhu nových částí na Nástěnku v administrační části.
- **Database API** – pod tímto značením se nachází několik dalších API, která dohromady tvoří podporu pro práci s databází.
- **HTTP API** – obsahuje pomocné funkce pro tzv. http požadavky.
- **File Header API** – sestává se z funkcí, které mohou načítat informace hlaviček pro vybraný plugin nebo šablonu.
- **Filesystem API** – zastřešuje několik tříd, které jsou určeny pro čtení a zápis souborů v lokálním souborovém systému.
- **Metadata API** – tato část slouží pro načítání a manipulaci metadat několika možných objektů v rámci WordPressu.
- **Options API** – jednoduchý a standardizovaný způsob jak ukládat, aktualizovat a mazat různá nastavení v databázi.
- **Plugin API** – pojednává o použití provázání nových akcí nebo filtrů z pluginu do celého WordPressu.
- **Quicktags API** – umožňuje přidávat do textového editoru novou rychlou volbu pro vybraný tag z HTML.

- **Rewrite API** – použitím této API je možné přepisovat pravidla použití některých metod WordPressu.
- **Settings API** – spolupracuje s Options API. Využívá se pro přidání nových sekcí nastavení a obslužných formulářů, které jsou po zpracování uloženy do vybraného nastavení v databázi.
- **Shortcode API** – pojednává o použití zkrácených zápisů v textových editorech příspěvků nebo stránek, které umožňují přidávat bloky kódu do obsahu stránky.
- **Theme Modification API** – sestává se ze všech funkcí a provázání používajících nějaké modifikační hodnoty šablony, které mohou autoři použít ve svých šablonách k uložení nastavení změn.
- **Theme Customization API** – umožňuje vývojářům přidávat vlastní ovládací prvky do nástroje Přizpůsobit v administraci vzhledu.
- **Transients API** – je dost podobné Options API. Slouží k dočasnému uložení dat do databáze pod určitým jménem a nastavením času expirace, po kterém jsou smazána.
- **Widgets API** – obsahuje metody pro přidávání nových widgetů použitelných při optimalizaci šablon.

## 4.2 Informační hlavička a publikace pluginu

V začátcích vývoje pluginu je vždy prvním krokem vytvoření správné struktury souborů. Na internetu jsou k dispozici i předpřipravené šablony souborové struktury. V hlavním adresáři musí být pak umístěn alespoň jeden PHP soubor, ve kterém se musí být umístěna informační hlavička pluginu. Tato hlavička pomáhá WordPressu rozeznat, že se jedná o adresář s pluginem a měla by obsahovat nutné základní údaje o pluginu. To znamená, že informace uvedené v hlavičce se zobrazí ve přehledu pluginu v administrační části. [25]

Hotový plugin je možné publikovat na Official WordPress Plugin Repository, kde je možné dohledat cizí pluginy. Tato oficiální stránka pluginů má však svůj systém jakým vyplnit údaje a stránku o pluginu. Tento systém spočívá v tom, že se do kořenového adresáře pluginu musí umístit textový soubor s názvem *readme.txt*, který musí mít předepsanou strukturu. V následujícím příkladu je uvedeno, že lze specifickým zápisem určit název stránky, nadpis a běžný text.

```
== Název záložky ==
```

```
= Nadpis =
```

```
Běžný text
```

Při publikaci i samotném vývoji pluginu existují tzv. nejlepší praktiky jako například zvolit správný název pluginu, používat běžné názvy záložek na stránce pluginu. Tyto praktiky lze najít na oficiálních stránkách WordPressu a je dobré se jimi řídit. Pokud se zdá vývojáři postup komplikovaný, může využít dostupného generátoru této dokumentace a po vyplnění formuláře získá text, který doplní do souboru. [26]

#### 4.2.1 Licence

Další otázkou při vývoji pluginu je použití licence, pod kterou je možné jej používat. Nejčastější používanou licencí je GNU General Public License, verze 2., pod kterou je vydán i samotný WordPress, anebo použít licenci kompatibilní s GPL2. Znění této licence se pak musí přidat do informační hlavičky k údajům o pluginu. [25][26]

### 4.3 Aktivace pluginu

Dalším krokem při vývoji pluginu pro WordPress je řešení co se má stát po aktivaci pluginu. Pro tento účel nabízí metody, které slouží k provázání s touto akcí. V dokumentaci lze najít i další provázání při deaktivaci a odinstalování pluginu. Tyto tzv. hooky jsou následující:

- **register\_activation\_hook** – slouží k registraci funkce prováděné po aktivaci.
- **register\_deactivation\_hook** – provázání s funkcí prováděnou po deaktivaci.
- **register\_uninstall\_hook** – volá funkci po odinstalování pluginu.

Při používání těchto provázání si musí vývojáři dávat pozor na to, aby se nesnažili ovlivňovat části kódu, které nejsou inicializovány nebo v případě odinstalování nemusí být k dispozici některé soubory. Nejčastěji se pak aktivační provázání používá k instalaci vlastní databáze, přidávání nových nastavení používaných v pluginu nebo k aktualizaci pluginu. [27]

### 4.4 Přidávání akcí

Přidávání akcí je při vývoji pluginů věc, bez které se programátor neobejde. Jakoukoliv vlastní funkci v pluginu lze díky přidávání akcí přiřadit různým událostem ve WordPressu.

Těchto událostí či spíše akcí je nesčetné množství a jsou v dokumentaci rozděleny do několika kategorií. Mezi asi nejužívanější patří akce z administrace a šablon. Někoho by mohlo svádět přidávat akci při aktivaci pluginu. Zde musí mít vývojář na paměti, že existují dvojí způsoby jak akci přidat neboť by za jistých okolností mohlo dojít k chybě. [28]

- **add\_action** – nemůže použít vně funkce vykonávané při aktivaci pluginu, protože tato metoda má dva parametry, kdy jedním je už vzpomínaná akce, která existuje při běhu WordPressu a vlastní plugin by nemusel k této akci mít zrovna přístup. Proto se tímto způsobem přidávají akce mimo nějaké funkce. Příkladem může být přidání vlastního textu do patičky HTML jak je tomu v následujícím příkladu.

```
function print_my_echo() {  
    echo 'This is what I want to say!';  
}  
add_action( 'wp_footer' , 'print_my_echo' );
```

Takovýmto způsobem lze přidávat například do administračního menu novou položku, která reprezentuje vyvíjený plugin.

- **do\_action** – je možné použít vně aktivace pluginu. Tato metoda má jediný povinný parametr, kterým je název vlastní funkce, čímž není závislá na žádné akci WordPressu. Rozdíl oproti předchozí metodě lze pochopit z následujícího příkladu, který demonstruje, jak je volána vlastní akce.

```
function print_my_echo() {  
    do_action( 'print_my_echo' );  
}  
register_activation_hook( __FILE__, 'print_my_echo' )
```

## 4.5 Značky šablon

Jednou z možností, kterých lze využít při vývoji pluginu jsou vlastní části obsahu, které lze volat vně jakékoliv šablony. Několik takovýchto značek má i samotný WordPress. Pokud mají být vlastní značky dostupné vývojářům šablon, musí být umístěny v hlavním PHP souboru pluginu. Každou značku představuje unikátní funkce, kterou lze pak volat v souborech šablon, jako je tomu na následujícím příkladu, kde je využita funkce z předchozích příkladů. [29]

```
<div><?php print_my_echo(); ?></div>
```

## 4.6 Ukládání dat do databáze

Jak už bylo několikrát zmíněno v této kapitole, mohou vývojáři využít k ukládání vlastních dat databázi, v níž je WordPress nainstalován. Pro ukládání dat do databáze je možné využít několik metod podle vhodnosti použití. [23]

- **WordPress Option Mechanism** - pomocí této metody lze do databáze ukládat, aktualizovat nebo z databáze načítat pod vlastním názvem různé volby z pluginu. Výhodou této metody je, že mohou být pod volbou ukládány jak řetězce, tak pole nebo PHP objekty, které jsou před vložením do databáze serializovány na řetězec a při načítání složeny zpět.
- **Post Meta** - je vhodné pro ukládání různých údajů týkajících se příspěvků a stránek.
- **Custom Taxonomy** – pro klasifikaci příspěvků a jiných objektů jako komentáře nebo uživatelé lze využít vlastní taxonomie, podle které se tyto objekty ukládají do databáze.
- **Vlastní tabulka** – nabízí možnost vytvoření vlastní tabulky, do které lze ukládat vlastní záznamy dle libosti pomocí SQL dotazů a stejně tak je načítat.

## 4.7 Registrace a použití skriptů

S vývojem pluginů je spojena i potřeba JavaScriptových souborů například při využití některé knihovny. Prvním krokem je registrace skriptů, při jejichž zápisu lze rozlišit, kdy a kde se mají použít. Skripty lze tedy načítat na straně šablony nebo pluginu. Dalšími možnostmi je nastavení, zda má skript využívat například knihovnu jQuery, s jakou prioritou se má načítat a zda se má načíst v hlavičce nebo v patičce. Posledním krokem je tuto registraci zabalit do nějaké funkce, kterou pak přidáme akci WordPressu, která obstarává načítání skriptů. Pro lepší představu je uveden příklad níže. [30]

```
function my_scripts() {  
    wp_register_script( 'custom-script', plugins_url( '/js/custom-script.js', __FILE__ ), array( 'jquery' ) );  
}  
add_action( 'wp_enqueue_scripts', my_scripts );
```

## **II. PRAKTICKÁ ČÁST**

## 5 SCROLLREVEAL.JS WORDPRESS PLUGIN

Hlavním bodem praktické části bylo vybrat jednu z porovnávaných knihoven a implementovat ji formou pluginu pro redakční systém WordPress. Při výběru jsem zohledňoval především jednoduchost použití knihovny a možnosti modifikace efektů. Vybrána byla knihovna ScrollReveal.js, která disponuje možností nastavit efektům různé parametry dle potřeby. Dalším důvodem byla i četnost referencí a návodů na použití této knihovny, která ve mně nabyla dojmu, že je o tuto knihovnu zájem. Nikde jsem se však nesetkal s žádným pluginem, který by umožňoval tuto knihovnu implementovat přímo na určité elementy stránky nebo do WordPressu vůbec. Myslím, že implementaci této knihovny jako pluginu mnozí vývojáři aplikací ve WordPressu přivítají.

### 5.1 Požadavky na plugin

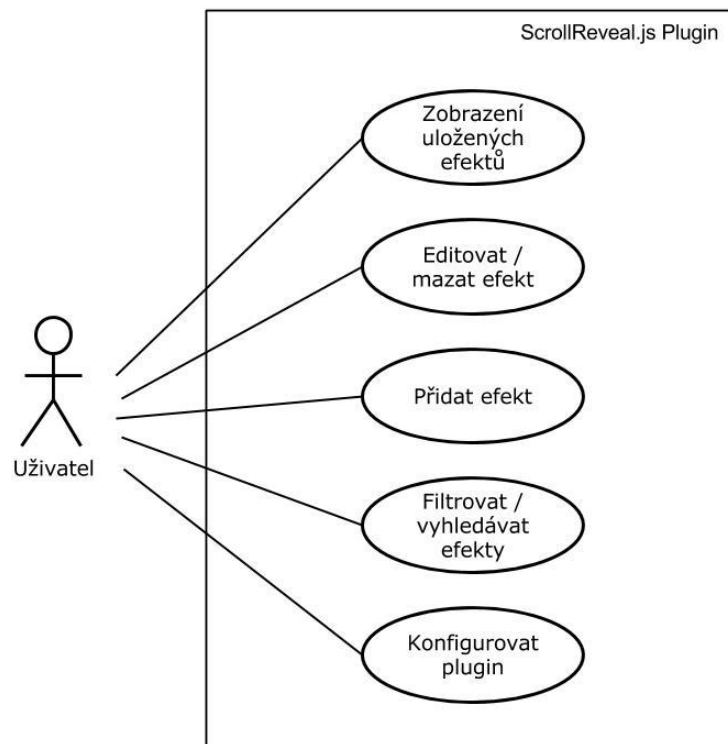
Před samotným vývojem pluginu bylo potřeba určit si jednotlivé vlastnosti, kterými měl plugin disponovat a co by měl uživatelům poskytovat a proto jednotlivé požadavky uvedu v následujícím výčtu:

- Dostupný jako položka v administračním menu
- Přidávat, editovat a mazat použité efekty vybrané ze seznamu.
- Filtrovat seznam použitých efektů a vyhledávání efektů.
- Provádět konfiguraci samotné knihovny.
- Implementovat knihovnu do WordPressu.
- Aplikovat uložené efekty na front-end webu.

Bez těchto bodů by nebyl plugin téměř využitelný. Nicméně se podařilo realizovat všechny body a vytvořit tak uživatelsky přívětivý plugin, po jehož uživatelích nejsou vyžadovány základní znalosti programovacích jazyků kromě znalosti HTML.

### 5.2 Případy užití

V rámci pluginu je v případě užití pouze jediná role a tou je samotný uživatel WordPressu. Předpokladem je, že musí být uživatel přihlášen do administrace redakčního systému WordPress aby měl přístup k pluginu. Pro plugin existuje pouze pět případů užití (Obrázek 5-1) a je patrné, že všechny vychází z některých požadavků na funkcionalitu pluginu.



Obrázek 5-1: Případy užití.

### 5.3 Souborová struktura

Jako každý jiný projekt, tak i plugin by měl mít jasně danou strukturu, aby další vývojáři měli možnost se v projektu zorientovat. Jednou z možností bylo použít tzv. boilerplate.

```
scroll-reveal-js
├── includes/
│   ├── class-wp-list-table.php
│   ├── config.php
│   ├── edit.php
│   ├── simple_html_dom.php
│   └── table-list.php
├── js
│   └── scrollReveal.min.js
├── template
│   ├── footer.php
│   └── sr-init.php
├── icon.png
└── index.php
```

Boilerplate je předem daná prázdná šablona souborového systému přímo pro plugin. Na internetu lze dohledat i generátor těchto šablon. Výše uvedené uspořádání adresářů použité v našem pluginu částečně vychází z boilerplate a ze zkušeností s uspořádáním souborů a složek u jiných webových projektů. V následujícím výčtu se nachází popis jednotlivých složek.

- scroll-reveal-js – obsahuje hlavní PHP soubor s informační hlavičkou a screenshoty.
  - includes – zde jsou umístěny všechny funkční soubory pro jednotlivé části pluginu.
  - js – obsahuje knihovnu ScrollReveal.js
  - template – zde jsou soubory, které nějakým způsobem ovlivňují obsah šablony front-endu.

## 5.4 PHP Simple HTML DOM Parser

Tato jediná PHP knihovna třetí strany slouží k parsování HTML a byla použita kvůli zjednodušení výběru ID selektorů ze stránek námi vytvořené webové aplikace. Samotný WordPress nic takového neumí, protože jak šablony, tak samotný obsah nemusí být dostupný během procesu generování webu. Výhodou toho parseru je právě to, že stahuje stejný stream dat jako například prohlížeč a má tak celou aplikaci k dispozici a je schopen rozebrat celý HTML DOM, z kterého se pak vyselektují ID elementů. [31]

## 5.5 Využití WordPress API

V teoretické části byly uvedeny jednotlivé části WordPress API (dále pouze API), které lze pro vývoj pluginu využít. Samozřejmostí pro vývoj pluginu je i dobrá znalost objektového programování v jazyce PHP a používání MySQL dotazů na databázi. Na každou část pluginu, která je reprezentována podstránkou v administračním menu, jež má svou funkci, bylo použito různých částí API.

### 5.5.1 Používání databáze

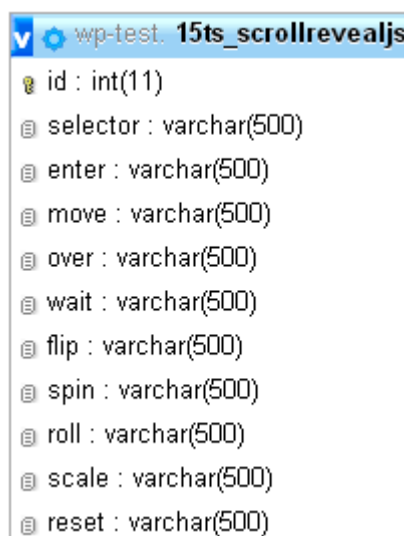
Jednou z funkcí pluginu je i konfigurace knihovny ScrollReveal.js. Aby bylo možné zvolenou konfiguraci uložit, je využito přidání volby z Options API, které ukládá do tabulky `[prefix_db]_options` námi přidané volby. Pro plugin jsou ukládány hned dvě volby. Jednou je uložení volby zda má být použita konfigurace, která ukládá pouze výsledek jediného

checkboxu. Další je ukládání konfigurace/nastavení knihovny z více polí formuláře, kde se ukládají jak volby z checkboxů, tak i z textových polí nebo rolovacího seznamu.

option_id	option_name	option_value	autoload
616	use_sr_config	a:1:{s:3:"use";s:1:"1";}	yes
617	sr_config	a:4:{s:7:"opacity";s:1:"1";s:8:"viewport";s:0:"";s:...	yes

Obrázek 5-2: Záznam voleb v databázi WordPressu.

Stejně tak lze vidět (Obrázek 5-2), že různá nastavení jsou uložena jako jediný string, který jak už bylo vzpomenuo v teoretické části, byl serializován z několika proměnných.



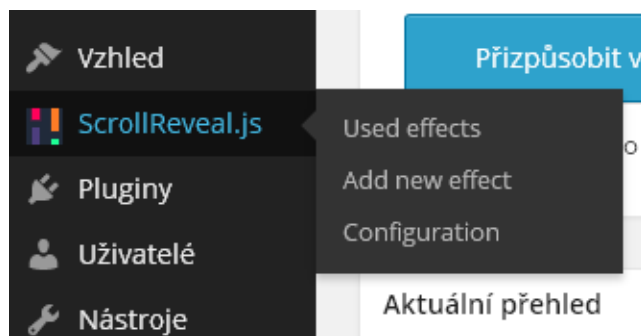
Obrázek 5-3: Nastavení tabulky pro ukládání efektů.

Důležitou funkcí pluginu je správa efektů. Plugin potřebuje ukládat tolik nastavení, kolik je použitých efektů. Pro tento účel už však nelze využít ukládání do tabulky voleb, protože je potřeba více záznamů a vytvářet pro každý další efekt novou volbu by bylo téměř nepoužitelné. Proto se při instalaci pomocí aktivačního provázání z Plugin API vytváří tabulka, jejíž nastavení lze vidět na obrázku 5-3. Tuto tabulku pak využívá jak výpis použitých efektů, tak i formulářová stránka pro přidání efektu, které jsou popsány níže.

### 5.5.2 Přidání pluginu do administračního menu

Aby bylo možné k pluginu přistupovat z administrace je potřeba přidat jej jako položku do některé z částí menu. Zde se nabízí několik možností. Jednou z nich je přidání pluginu jako podpoložku části Nastavení nebo Nástroje. Pro plugin však byla zvolena ideální možnost přidání a to jako další položky administračního menu. Jelikož má plugin funkčně tři části,

je možné přidat do této položky tyto části jako rozšiřující položky (Obrázek 5-4). Jako jazyk položek menu pluginu byla zvolena angličtina. Důvodem je předpoklad, že budou plugin využívat především zahraniční uživatelé. Stejně tak je v angličtině text všech stránek pluginu.



Obrázek 5-4: ScrollReveal.js Plugin jako položka administračního menu.

Obsluha přidání nového menu je prováděna v hlavním souboru pluginu *index.php*. Pro přidávání slouží metoda *add\_menu\_page*, u které je klíčové zadání tzv. slugu, který lze chápat jako jakýsi alias, jež je pak použit u dalších metod. Dalšími nastaveními jsou uvedeny názvy pluginu, stránek, použitá ikonka v menu a důležitá obslužná funkce pro zobrazení dané stránky.

```
function scroll_reveal_js_menu() {  
    $page_title = 'ScrollReveal.js Plugin';  
    $menu_title = 'ScrollReveal.js';  
    $capability = 'activate_plugins';  
    $menu_slug = 'scroll_reveal_js';  
    $function = 'scroll_reveal_js_page';  
    $icon_url = plugins_url( 'scroll-reveal-js/icon.png' );  
    $position = 64;  
  
    add_menu_page( $page_title, $menu_title, $capability,  
        $menu_slug, $function, $icon_url, $position );  
}
```

K takto vytvořenému menu, jehož zápis je na příkladu výše, jsou následně přidány podstránky pomocí metody *add\_submenu\_page*, která má podobné nastavení jako metoda pro hlavní menu. Prvním parametrem metody je rodičovský slug, který rozhoduje kam se má položka přiřadit. Pokud by měla být stránka přiřazena do Nástrojů, dosadil by se za tento rodičovský slug název funkčního souboru *tools.php* nebo jiný uvedený v dokumentaci

WordPressu. Pokud se má stránka přiřadit našemu menu zápisem uvedeným níže, použije se slug rodičovského menu, který byl definován ve výše uvedené funkci. Pro podstránku je použita stejná obslužná funkce jako pro hlavní menu, která je pak přidána do akcí.

```
add_submenu_page(  
    'scroll_reveal_js',  
    'Used effects',  
    'Used effects',  
    'activate_plugins',  
    'scroll_reveal_js',  
    'scroll_reveal_js_page'  
);
```

### 5.5.3 Výpis použitých efektů

Pro tuto stránku bylo použito dědění třídy *WP\_List\_Table*, kterou interně využívá WordPress k výpisům všech tabulek v databázi. Přetěžování funkcí této třídy je v našem pluginu využito právě k výpisu všech uložených efektů z databáze v tabulce *[prefix db]\_scrollrevealjs*, která byla vytvořena při instalaci. Aby bylo možné tuto třídu použít, doporučuje dokumentace WordPressu zdrojový soubor třídy zkopírovat do složky pluginu, jmenovitě do složky *include*, neboť je tato třída neustále upravována a mohlo by tak dojít k chybám pluginu po některých aktualizacích Wordpressu. Pro výpis je pak definována vlastní třída s děděním třídy *WP\_List\_Table*, která se nachází v souboru *table-list.php*. Pro výpis efektů jsou však přetěžovány jen některé funkce a jejich použití pro funkcionalitu stránky jsou v následujícím výčtu.

- **column\_default** – definuje základní zobrazení sloupce v tabulce.
- **column\_selector** – definuje výpis a chování sloupce s názvem identifikátoru elementu. Jako jediný lze seřazovat a má tzv. rychlé volby pro editaci a mazání záznamu.
- **column\_cb** – vykresluje checkboxy pro hromadné úpravy.
- **column\_param** – spojuje všechny použité parametry z více sloupců tabulky a vykresluje je jako jeden řetězec.
- **get\_columns** – definuje, které sloupce mají být použity.
- **get\_bulk\_action** a **process\_bulk\_action** – definuje, které hromadné akce mají být použity a co se má při nich provést.

- **get\_sortable\_columns** – výběr všech tříditelných sloupců.
- **prepare\_item** – nejdůležitější metoda, která se stará o selekci záznamů z databáze a převod do podoby použitelné pro ostatní metody.
- **extra\_tablenav** – obsahuje rozšíření pro práci s výpisem, v našem případě filtrování efektů podle toho na jaké stránce byly použity.

Naše takto upravená třída je instancována následujícím způsobem:

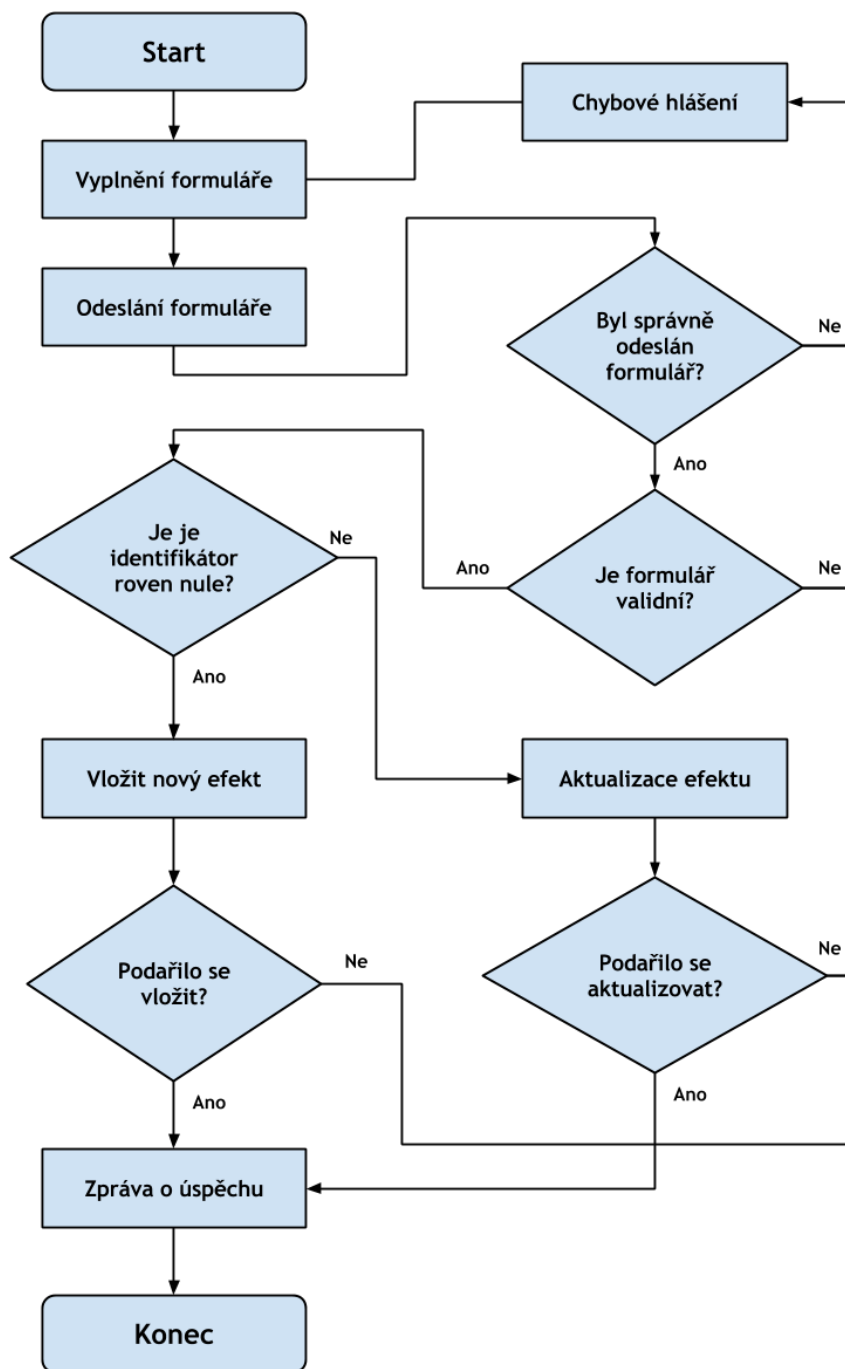
```
$table = new Scroll_Reveal_List_table();
```

Abychom mohli vůbec něco vykreslit, musí být zavolána metoda *prepare\_item*, která jak už bylo zmíněno výše, načte a připraví záznamy z tabulky. Pak už stačí zavolat jenom metodu pro vykreslení uvedené níže, která je uzavřena ve formuláři s metodou GET. Společně s vykreslením je volána ještě metoda pro vyhledávací pole, která umožňuje vyhledávat podle selektoru, jež znázorňuje element spojený s efektem.

```
<form id="effects-table" method="GET">
  <input type="hidden" name="page" value="<?php echo
  $_REQUEST['page'] ?>"/>
  <?php
      $table->search_box('Search Selector',
  'search_selector');
      $table->display()
  ?>
</form>
```

#### 5.5.4 Přidávání a editace efektů

U přidávání efektů je použito především Plugin API. Tato stránka má hlavní funkcionalitu ve zpracování údajů z formuláře. Protože tato stránka slouží pro dvojí účel je tomu přizpůsobeno i zpracování formuláře s ošetřením některých podmínek (Obrázek 5-5). První podmínkou je dotaz na tzv. nonce. Nonce je globální proměnnou WordPressu a lze jej chápat jako bezpečnostní pojistku, zda nedošlo při odeslání formuláře k neočekávanému nebo dvojímu odeslání požadavku, který by mohl způsobit nesprávné změny v databázi. Pokud je provedeno odeslání správně a jednou, pokračuje se k validaci dat odeslaných formulářem. Validaci obsluhuje samostatná funkce, která přijme jako parametr objekt položky a prochází jednotlivé části, při čemž ukládá případné chybové zprávy. Pokud je vše v pořádku dojde buď k zápisu, nebo aktualizaci databáze.



Obrázek 5-5: Validační proces přidávání a editování efektu.

Vstupy formuláře jsou také v samostatné funkci, která se připojuje do WordPressu pomocí další metody z Plugin API a to `add_metabox`. V parametrech funkce se kromě nastavení použití předává opět slug našeho pluginu a název obslužné funkce. Tady lze dobře pochopit k čemu slug slouží. V dalším kroku se musí nějak vykreslit definovaný formulář. K tomu slouží metoda `do_meta_boxes` z následujícího příkladu, kde je opět jedním z parametrů funkce slug pluginu a metoda tak ví, který meta box použít v rámci našeho pluginu.

```
<form id="form" method="POST">
    ...
    <?php do_meta_boxes('scroll_reveal_js', 'normal', $item); ?>
    <input type="submit" value="Save" id="submit" class="button-
primary" name="submit">
    ...
</form>
```

Na této stránce se také mimo jiné využívá výše vzpomínaného HTML DOM parseru. Na základě vybraného indexu je parsována stránka a je tak v meta boxu k dispozici rolovací seznam ID všech elementů na této stránce.

```
$url = get_page_link($page_id);
$dom = file_get_html($url);
$divs = $dom->find('[id]');
$sorted = [];
foreach ($divs as $div) {
    array_push($sorted, $div->id);
}
sort( $sorted, SORT_STRING | SORT_FLAG_CASE );
```

Na tomto příkladu lze vidět, jak je získána adresa zvolené stránky (části) WordPressu a rozebrána. Ze získaného stromu jsou pomocí metody *find* s parametrem podobným selektoru používaného v jQuery vyfiltrovány pouze elementy s atributem *id*. Z těchto elementů musí být však vybrány pouze *id* prvky, aby mohli být následně seřazeny podle názvu.

### 5.5.5 Konfigurace pluginu

Něž dojde k několikrát vzpomínanému uložení konfigurace do voleb, musí být pro proces serializace a ukládání definován formulář, který se má zpracovávat. V případě voleb se používají tzv. pole a sekce nastavení, které jsou definovány v Settings API. Využití ukládání nastavení je pro konfiguraci knihovny ideální.

Prvním krokem je registrace nastavení pomocí metody *register\_settings*, která říká WordPressu, aby ukládal do tabulky *[prefix\_db]\_options* novou volbu. Tato metoda má jako první parametr název skupiny, do které má nastavení patřit a je pro použití nastavení rozhodující.

```
register_setting( 'sr-option-group', 'use_sr_config' );
```

Na předchozím příkladu je registrace nastavení, zda má být použita konfigurace knihovny. Pro toto nastavení musí být pak definována ještě sekce, vně které bude generováno pole, v našem případě checkbox. Přidání sekce v následujícím příkladu je pak použito pro sjednocování více polí formuláře.

```
add_settings_section('sr_main', 'ScrollReveal.js Settings',
    'sr_section_text', 'sr-option-group');
```

Prvním parametrem je název sekce, který bude použit právě pro přiřazení a posledním parametrem je definováno, které skupině nastavení má patřit. Dalším krokem je přidání pole.

```
add_settings_field(
    'sr_text_string',
    'Use configuration',
    'sr_setting_string',
    'sr-option-group',
    'sr_main'
);
```

Posledním parametrem u přidání pole je právě název sekce, do kterého se má přidat. Kromě dalších nastavení jako je název apod., nás zajímá třetí parametr definující obslužnou třídu pro vykreslení checkboxu.

```
function sr_setting_string() {
    $options = get_option('use_sr_config'); ?>
    <input type="checkbox" name="use_sr_config[use]" value="1"<?php checked( isset( $options['use'] ) ); ?> />
    <?php
}
```

Aby mohlo být nastavení použitelné, musíme si funkcí *get\_option* načíst obsah volby z databáze, na základě kterého se do vstupu generuje jeho hodnota. Zvláštností je zde atribut *name*, kterému neobvyklým zápisem nastavíme, do které volby (*use\_sr\_config*) se má vstup ukládat a pod jakým klíčem (*use*). Pokud je takto vše definováno nezbyvá než nastavení vykreslit způsobem v následujícím příkladu. Pro zpracování formuláře je zde použit interní soubor *options.php*, který obstarává zmíněnou serializaci. Z parametru v metodách *settings\_fields* a *do\_setting\_sections* lze poznat, že se mají vykreslit sekce a pole, které jsme definovali výše.

```
<form method="post" action="options.php">

    <?php

        settings_fields( 'sr-option-group' );

        do_settings_sections( 'sr-option-group' );

        submit_button();

    ?>

</form>
```

## 5.6 Použité akce a skripty

Některé zmíněné obslužné funkce na všech stránkách pluginu musí být provázány s různými vnitřními akcemi WordPressu. V následující tabulce je uvedeno, pro jakou akci se má která obslužná funkce provést společně s jejím krátkým popisem. Mezi těmito akcemi lze také najít tzv. připnutí používané knihovny pro scroll reveal efekty.

Název akce	Název funkce	Popis funkce
admin_init	regis- ter_scroll_reveal_js_settings	Použití nastavení v konfiguraci knihovny po načtení administrace.
admin_menu	scroll_revela_js_menu	Při generování administračního menu se přidá položka pluginu.
plugins_loaded	scroll_reveal_js_update_db_ check	Kontrola zda nebyl updatován plugin po načtení pluginů.
wp_enqueue_scripts	load_sr_js	Načtení skriptu knihovny ScrollReveal.js
wp_footer	sr_footer	Připojení souboru do patičky šablony generujícího efekty do šablony.
	sr_init	Připojení souboru obstarávající inicializaci knihovny ScrollReveal.js a výpis její konfigurace.

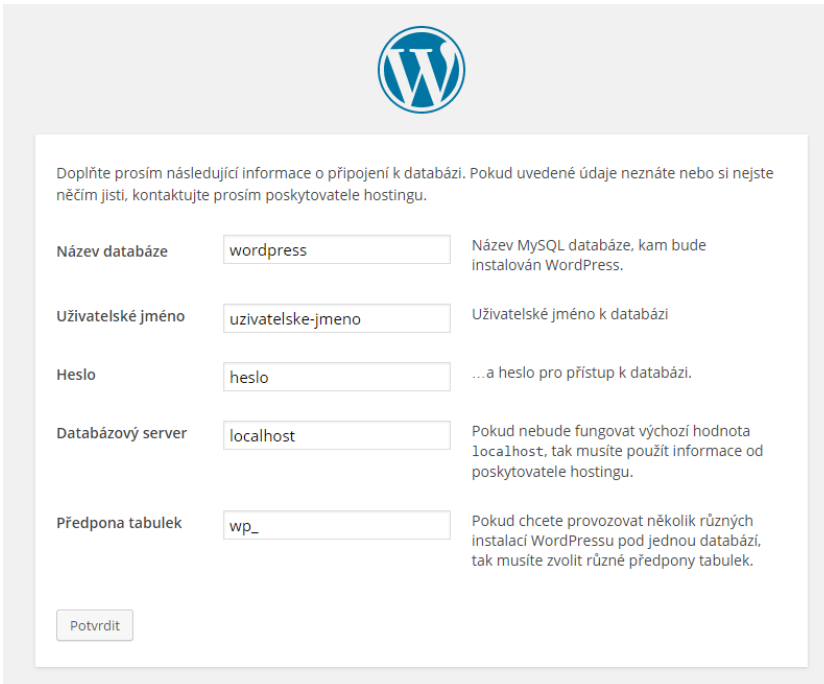
Tabulka 5-1: Seznam použitých akcí a funkcí.

## 6 TVORBA ZKUŠEBNÍHO WEBU

Během vývoje pluginu byla ze začátku pro testování funkčnosti využita dočasná instalace WordPressu na localhostu. Pro aplikování efektů na front-endu pak byla využita základní šablona Twentyfiteen, která je standardně součástí instalace WordPressu. Šablona však nevyhovovala dostatečně potřebám pro konečnou demonstraci efektů, což bylo jednou z částí zadání práce, a proto byla vytvořena vlastní šablona, jejíž vývoj společně s implementací v rámci WordPressu je popsána níže.

### 6.1 Instalace a nastavení WordPressu

Instalace WordPressu je v dnešní době vcelku jednoduchý krok. Celý produkční balíček lze stáhnout v podobě komprimovaného archívu společně s českou lokalizací a jazykovou podporou na [www.cs.wordpress.org](http://www.cs.wordpress.org). O překlad se z velké části stará komunita na webu [separatista.net](http://separatista.net), kde lze také sledovat novinky okolo překladu. Po stažení byl balíček rozbalen do adresáře na localhostu. Zadáním cesty ke složce se zdrojovými do prohlížeče se dostaneme k průvodci instalace, kde je potřeba vyplnit důležité informace (Obrázek 6-1) jako název databáze a serveru, do níž má být WordPress nainstalován, společně s přihlašovacími údaji. Posledním údajem je předpona (v práci byl několikrát použit prefix) databáze, který je z bezpečnostního hlediska doporučeno změnit, čím lze znepříjemnit práci potenciálním útočníkům, kteří by chtěli nějak narušit funkci této webové aplikace.



The image shows the WordPress installation database configuration screen. At the top center is the WordPress logo. Below it is a heading: "Doplňte prosím následující informace o připojení k databázi. Pokud uvedené údaje neznáte nebo si nejste něčím jisti, kontaktujte prosím poskytovatele hostingu." Below this are five rows of input fields with labels and descriptions:

Název databáze	<input type="text" value="wordpress"/>	Název MySQL databáze, kam bude instalován WordPress.
Uživatelské jméno	<input type="text" value="uzivatelske-jmeno"/>	Uživatelské jméno k databázi
Heslo	<input type="text" value="heslo"/>	... a heslo pro přístup k databázi.
Databázový server	<input type="text" value="localhost"/>	Pokud nebude fungovat výchozí hodnota localhost, tak musíte použít informace od poskytovatele hostingu.
Předpona tabulek	<input type="text" value="wp_"/>	Pokud chcete provozovat několik různých instalací WordPressu pod jednou databází, tak musíte zvolit různé předpony tabulek.

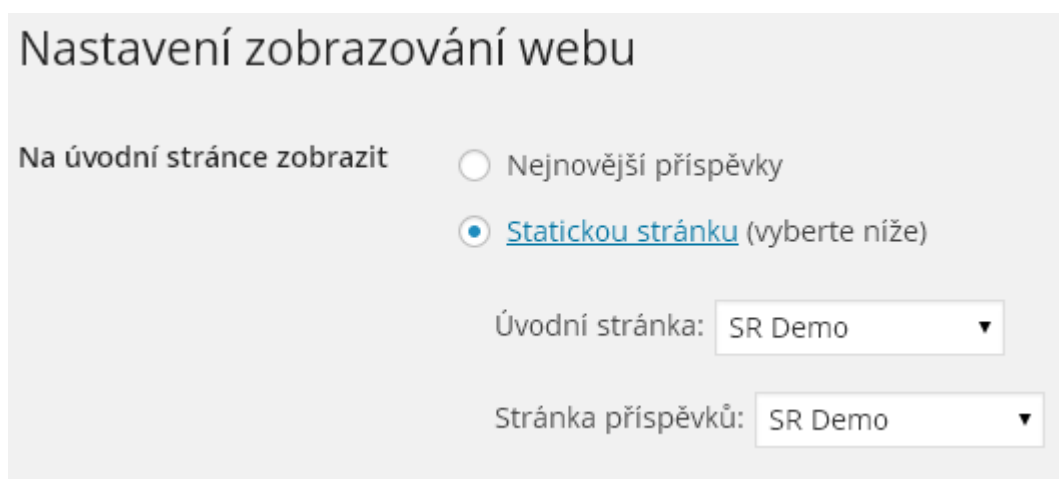
At the bottom left is a button labeled "Potvrdit".

Obrázek 6-1: Instalační formulář WordPressu.

Na dalších stránkách instalace se pak vyplní pouze další údaje jako název webu a přihlašovací údaje pro hlavního správce webu. Do administrace se po dokončení instalace dostaneme přes podstránku *wp-admin*.

### 6.1.1 Nastavení struktury webu

Pro náš zkušební web byl zvolen přístup známý jako „single page application“, který lze chápat jako jedinou statickou stránku, v níž však existuje možnost navigace pomocí kotev v obsahu. Statickou stránku je možné ve WordPressu nastavit přes Nastavení -> Zobrazování, kde lze pomocí volby (Obrázek 6-2) vybrat stránku, která se má takto zobrazit na úvodu.



Obrázek 6-2: Nastavení statické stránky.

Pokud v rámci celého webu vytvoříme pouze jednu stránku, která je následně vybrána, a zakážeme možnost registrace dalších uživatelů, splňuje web zvolený přístup.

## 6.2 Tvorba šablony

Námi vyvíjená šablona byla vytvořena na základě grafického návrhu Bc. Alžběty Suré, studentky Digitálního designu na Fakultě multimediálních komunikací UTB ve Zlíně. Návrh šablony byl během konzultací se studentkou optimalizován tak, aby v jeho rámci bylo možné použít scroll reveal efekty a zároveň splňoval předpoklady moderní podoby webu.

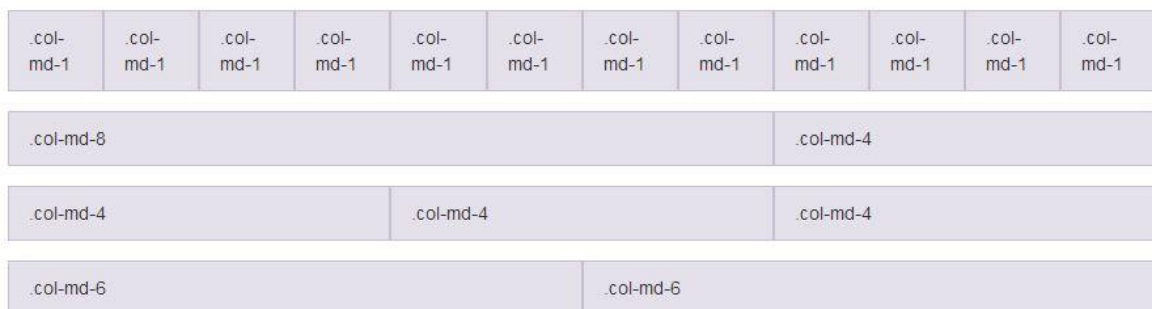
Vývoj šablony podobně jako plugin a vlastně celý WordPress má svůj kodex, který navádí vývojáře k dodržování jednotné úpravy šablon například pro publikaci šablon. Dodržením určitých postupů lze tak vytvářet náhledy šablon a zobrazení informací o šabloně.



- `functions.php` – zde lze umístit jakýkoliv funkční kód z WordPress API, který se provádí při načtení šablony. V našem případě je v tomto souboru registrace a připojení skriptů Bootstrapu.
- `header.php` – do tohoto souboru jsou přiřazeny výpisy meta tagů, titulek stránky, RSS feed a skripty volané v hlavičce z jádra WordPressu nebo pluginů.
- `page.php` – zde se nachází tělo naší stránky. Funkční částí je výpis příspěvků.
- `screenshot.png` – náhledový obrázek šablony.
- `style.css` – na začátku tohoto skriptu se nachází informační hlavička podobná hlavičce v pluginu, ve které je opět uvedeno jméno autora, licence apod. načítané při náhledu šablony v administraci.

## 6.2.2 Bootstrap

V předchozí kapitole padlo o tomto front-endovém frameworku několikrát slovo. Framework byl použit především kvůli skvělé vlastnosti podpory responsivního designu, který je v dnešní době mobilních zařízení neodmyslitelnou součástí webových aplikací. Responsivní design by především zvolen kvůli možnosti otestovat chování scroll reveal efektů na mobilních zařízeních. Kromě možnosti responsivního návrhu disponuje Bootstrap stylistickými a funkčními komponentami pro formulářové prvky, prvky navigace, kolizní odstavec, záložky apod. Stěžejní částí je však tzv. Grid systém.



Obrázek 6-4: Instalační formulář WordPressu.

Tento systém 12 sloupců (Obrázek 6-4) umožňuje sestavovat vývojářům nepřeborné množství layoutů stránky. Každému divu jdou však přiřadit rozdílné třídy pro různé typy zařízení, které se rozpoznávají pomocí tzv. *media queries*. Bootstrap rozlišuje 4 druhy zařízení – chytré mobily, tablety, stolní počítače (téměř nepoužívaný poměr rozlišení 4:3) a širokoúhlé displeje. Čtyřmi různými třídami tak můžeme jednomu divu/sloupci nastavit, aby se pro mobil zobrazoval na celou šířku (`col-md` prefix), pro tablet na 50% šířky (`col-sm` prefix),

na desktopu (col-md prefix) a širokoúhlých zařízeních (col-lg prefix) na 1/3 šířky, jak je tomu na příkladu zápisu níže. Sloupce však musí být uzavřeny ještě v řádku, čímž je tento přístup hodně podobný používání HTML tabulek. Pokud chceme vymezit nějakou maximální šířku celého layoutu u příliš velkých rozlišení, uzavřeme vytvořené řádky ještě do kontejneru. [33]

```
<div class="container">
  <div class="row">
    <div class="col-lg-4 col-md-4 col-sm-6 col-xs-12">
      Nějaký obsah
    </div>
  </div>
</div>
```

### 6.2.3 Rozložení stránky

Celá stránka je rozložena do pěti oddílů (Příloha I.), jejichž názvy i obsah jsou v anglickém jazyce, opět kvůli předpokladu, že se plugin dostane především k zahraničním uživatelům. Tyto oddíly jsou na stránce umístěny chronologicky jako ve výčtu níže a každý zabírá celou šířku okna.

- 1) navbar – zde se nachází menu s odkazy na kotvy v dokumentu. Celá navigace je fixována na horní okraj okna, tudíž je k dispozici při posouvání celou stránkou.
- 2) homepage – je první kotva na stránce a je definována šedým pozadím. Slouží jako úvodní část s krátkým představením.
- 3) about – druhá kotva, která má modré pozadí. Zde se nachází informační obsah.
- 4) use – je poslední kotva na stránce. Na červeném pozadí pak probíhá demonstrace jednotlivých efektů s krátkým popisem. Oproti zbytku šablony se zde pro výpis obsahu využívá příspěvek z WordPressu, který lze editovat přímo v administraci.
- 5) footer – patička stránky je poslední částí webu se šedým pozadím obsahující motto a autory šablony.

## 7 POUŽÍVÁNÍ EFEKTŮ POMOCÍ PLUGINU

### 7.1 Nastavení nového efektu

Možnosti použití pluginu byly vzpomenuty již v dřívějších kapitolách. Detaily animace, které jdou efektům nastavovat, jsou z části popsány v kapitole 1.2. Ve formuláři nastavení efektu (Obrázek 7-1) je však ještě další parametr, které ovlivňují chování animace.

ScrollReveal.js Plugin

Choose page SR Demo

[back to list](#)

Effect details

Selector	<input type="text" value="footer"/>	Enter	<input type="text" value="left"/>
Move	<input type="text" value="Length of move"/>	Over	<input type="text" value="Time interval in seconds"/>
Wait	<input type="text" value="Time of delay in seconds"/>	Flip	<input type="text" value="Horizontal rotation in degrees"/>
Spin	<input type="text" value="Vertical rotation in degrees"/>	Roll	<input type="text" value="Rolling in degrees"/>
Scale	<input type="text" value="Ratio in percent"/>	Reset	<input type="checkbox"/>

[Save](#)

Obrázek 7-1: Formulář pro přidání a editaci efektu.

Formulář mimo jiné nabízí výběr stránky (volba *Choose page*) z, které se mají vybrat identifikátory elementů. Výběr stránky reaguje na změnu volby novým odesláním formuláře, takže není třeba volbu nijak potvrzovat. Po načtení formuláře stačí z volby *Selector* vybrat potřebný identifikátor a provést potřebná nastavení. Další možností, jak získat nové identifikátory pro elementy, je použití HTML rozhraní editoru stránek nebo příspěvků, kde je možné elementům stránek přidávat atribut *id*. Případně, při dobré znalosti HTML i samotný editor šablon. Našeptávače textových vstupů informují uživatele, v jakých jednotkách má být zadaný údaj, čímž uživateli odpadáva potřeba jednotku zadávat ve správném tvaru a stačí pouze hodnota parametru. Jednotky, pokud nějaké u parametru jsou, se nastavují až při aplikaci na vybraný element. Dalším novým parametrem je volba *Reset*. Její význam je jednoduchý, pokud je checkbox zatrhnut, přidává se elementu stejnojmenný parametr, který nastavuje, aby se prováděla animace elementu pokaždé, kdy vstoupí do okna prohlížeče. Plugin tak skutečně dovoluje používání knihovny bez znalosti syntaxe zápisu.

## 7.2 Nastavení konfigurace

Používání konfigurace knihovny umožňuje uživateli ovlivňovat chování knihovny. Konfiguraci nalezne uživatel na stránce Configuration, kde se nachází formulář (Obrázek 7-2) s několika možnostmi. První z nich udává, zda má být konfigurace použita.

SR.js Settings

Choose your configuration of ScrollReveal.js Plugin

Use configuration

Configuration Settings

Set configuration of ScrollReveal.js Plugin

Opacity

Mobile

Reset

Viewport

Delay

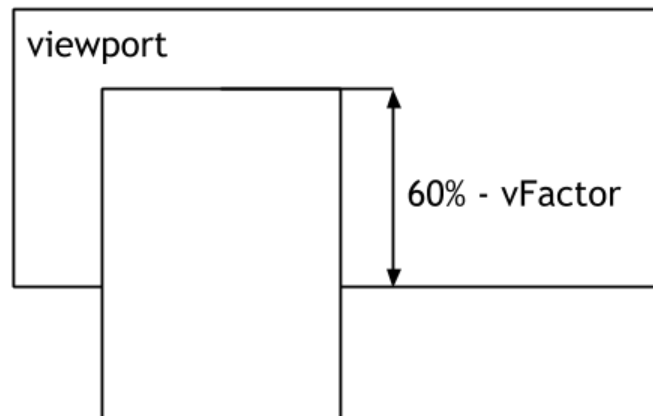
vFactor

Obrázek 7-2: Formulář konfigurace knihovny ScrollReveal.js.

Význam dalších možností konfigurace ze sekce *Configuration Settings* zatím nebyl nikde popsán. Jak uvedené volby ovlivňují chování knihovny je uvedeno v následujícím výčtu.

- Opacity – nastavuje pro všechny efekty počáteční hodnotu průhlednosti elementu před provedením animace. Hodnota průhlednosti se pohybuje v rozmezí 0-1 a defaultně je nastavena na nulu. Tímto zajišťujeme animaci efekt vpadnutí. Pokud však nastavíme průhlednost na 1, efekt vpadnutí není viditelný.
- Mobile – říká knihovně, zda se mají efekty používat i na mobilních zařízeních. Standardně je používání na mobilních zařízeních zakázáno.
- Reset – s touto volbou se uživatel může setkat i v přidávání jednotlivých efektů. Zde je její význam rozšířen na všechny efekty, a pokud je volba zatrhnuta, resetují se všechny efekty na stránce.
- Viewport – zadáním identifikátoru elementu lze omezit používání z celého okna pouze na část zastoupenou některým elementem stránky.

- Delay – zde může uživatel vybrat, zda se má zpoždění efektu provádět pouze jednou (defaultně) nebo pokaždé, je-li některému efektů nastaven reset.
- vFactor – umožňuje uživateli nastavit pro elementy, při kolika procentech (standardně je nastaveno 60%) své plochy zasahující do okna prohlížeče (Obrázek 7-3) má být animace elementu spuštěna. Hodnota v poli je pak zadávána jako poměr, tedy desetinné číslo v rozsahu 0 až 1.



Obrázek 7-3: Použití vFactor parametru konfigurace.

## 8 PUBLIKOVÁNÍ PLUGINU

Publikování pluginu je možné buď na stránkách WordPress.org nebo na vlastních stránkách. Pro publikování jsou důležité dva aspekty, které jsou uvedeny v následujících podkapitolách.

### 8.1 Informační hlavička

Bylo již několikrát uvedeno, že se informační hlavička musí nacházet v hlavním souboru pluginu. Bez této hlavičky není WordPress schopen plugin rozpoznat a nemohl by tak být náš plugin aktivován v administraci. Pro vyvíjený plugin má informační hlavička následující tvar.

```
/*
Plugin Name: ScrollReveal.js Plugin
Plugin URI: http://sr.cekuj.net
Description: Use and manage scroll reveal effects using
ScrollReveal.js right in WordPress
Version: 1.0
Author: Jan Benedík
Author URI: https://cz.linkedin.com/pub/jan-benedik/a0/84b/636
Tags: ScrollReveal.js, scroll, reveal, effects
License: GNU General Public License v2 or later
*/
```

### 8.2 Soubor readme.txt

Tento soubor, jak už bylo řečeno, slouží WordPress Plugin Directory k parsování informací o pluginu, které jsou následně produkovány uživatelům, pokud na stránkách WordPress.org na plugin narazí. Některé informace jsou použity i v náhledu Instalace pluginů v administraci. Pokud jsou dodrženy všechny pravidla pro sestavení tohoto textového souboru, vypadá jeho struktura následovně.

```
=== ScrollReveal.js Effects ===
// základní údaje, název licence a krátký popis
== Description ==
// rozsáhlý popis pluginu
== Installation ==
```

```
// stručný popis instalace společně s dokumentací
== Screenshots ==
// náhledové obrázky
== Changelog ==
// informace o změnách v jednotlivých verzích
```

Vytvořený soubor byl otestován validačním formulářem a byl vyhodnocen bez chyb. Tímto plugin splňuje první krok pro úspěšnou publikaci ve WordPress Plugin Directory. Dalším krokem je poskytnout soubor *readme.txt* ve formuláři žádosti o zveřejnění ve zmiňované WordPress Plugin Directory, jejíž správci rozhodnou o zveřejnění. [26]

## 9 VYUŽITÍ NA MOBILNÍCH ZAŘÍZENÍCH

Scroll reveal efekty vytvářené pomocí zvolené knihovny ScrollReveal.js jsou využitelné i na mobilních zařízeních. Nicméně se lze setkat i s uživateli, kteří měli zájem vypnout používání efektů. Většina z nich zasypala svými dotazy přímo na GitHubu vývojáře knihovny Juliana Lloyda, který od verze 2 umožnil vypnutí efektů, což je zahrnuto i v konfiguraci pluginu. Používání scroll reveal efektů na mobilních zařízeních sebou nese několik otázek.



Obrázek 9-1: Konverzace J. Lloyda s uživatelem na GitHub.com.

### 9.1 Má význam efekty na mobilních zařízeních vůbec využívat?

Určitě ano, pokud se dodrží jisté logické praktiky, je možné vytvořit webovou aplikaci zajímavou i při rolování na mobilu. A pokud se dodrží následující dvě nepsaná pravidla, není problémem efekty využívat:

- **Používat krátký čas průběhu animace** – ideálně do 1 sekundy, posouvání stránky na mobilu je hodně rychlé.
- **Pokud možno vyhnout se zpožděním** – podobné jako předchozí, při použití dlouhého času si nemusí uživatel animace vůbec všimnout.

### 9.2 Je možné zpoždění úplně zakázat?

Zpoždění lze zakázat, nicméně zatím není zahrnuto v konfiguraci knihovny. Autor knihovny však zveřejnil v diskuzi na GitHubu možnost odkomentování následujícího kódu, který tuto potřebu obsluhuje. Tato část kódu se v knihovně nachází na řádce 375.

```
if (self.isMobile() && self.config.mobile) elem.config.wait = 0
```

### 9.3 Ovlivňuje počet spouštěných efektů plynulost posouvání?

Plynulost posouvání má více faktorů, které ji mohou ovlivňovat. Může to být hardwarová výbava mobilního zařízení, výkonost používaného mobilního prohlížeče. Pokud je však *viewport* zahlcen nepřiměřeným počtem, který není vždy možný vzhledem k omezené zobrazovací ploše, mohou se objevit trhavé pohyby apod.

### 9.4 Mohou efekty narušit layout stránky?

Stejně jako u běžného kódování webu je využíván přístup „mobile first“, měl by se jím uživatel či vývojář řídit i při používání efektů. To znamená myslet dopředu. Pokud se například při pohybu elementu neopatrně použije vzdálenost, jejíž začátek leží mimo layout stránky, může se na stránce objevit například nepříjemné přetečení (Obrázek 9-2) v ose X.



Obrázek 9-2: Demonstrace horizontálního přetečení u efektu.

Tomuto se naštěstí lze vyhnout použitím správných meta tagů v hlavičce webu. Pokud tedy doplníme do hlavičky následující kód, vyhneme se tomuto problému.

```
<meta name='viewport' content='content="width=device-width, maximum-scale=1.0, minimum-scale=1.0, initial-scale=1.0' />  
  
<meta name="apple-mobile-web-app-capable" content="yes" />
```

## ZÁVĚR

Cílem diplomové práce bylo popsat scroll reveal efekty a demonstrovat jejich použití v rámci redakčního systému WordPress, kam měly být implementovány pomocí pluginu. A také vytvořit demonstrační web obsahující tyto efekty.

Teoretická část se zabývá popisem scroll reveal efektů na moderních webech, kde by měly tyto efekty zvyšovat atraktivitu obsahu. Další částí je popis obecných animací, které tyto efekty používají. Na tuto část pak navazuje teoretický popis realizace animací pomocí základních webových technologií jako je HTML5, CSS3 a JavaScript. Dále byly popsány vlastnosti a rozdíly mezi dostupnými JavaScript knihovnami scroll reveal efektů. Předposlední částí je pak přiblížení funkční výbavy CMS WordPress. V závěru teoretické části je popsán přístup k jádru WordPressu a jeho využití při tvorbě pluginů.

Praktická část diplomové práce nejprve popisuje postup při vývoji pluginu pro CMS WordPress a také jak byly při vývoji pluginu přímo využívány jednotlivé části WordPress API. Dalším krokem byla tvorba zkušebního webu, který disponuje také responsivním designem. Byla tak vytvořena, dle návrhu studentky Fakulty multimediální komunikací, šablona pro CMS WordPress využívající populární front-endový framework Twitter Bootstrap. Při tvorbě šablony bylo nutné pochopit hierarchii, kterou se řídí WordPress při výpisu obsahu, což je v této části také popsáno. Dále byl popsán princip publikování pluginu na oficiálních stránkách WordPress.org. V poslední části je uvedena diskuze použití scroll reveal efektů na mobilních zařízeních, kde je mimo jiné také popsáno jak se vyhnout nepříjemnostem spojených s použitím například na chytrých telefonech.

Používání scroll reveal efektů v budoucnosti určitě má své místo a vytvořený plugin může uživatelům WordPressu přiblížit jejich využívání. A to především díky možnosti jejich správy a rychlé editace bez znalosti jakékoliv syntaxe zápisu a nutnosti zasahovat do zdrojových kódů šablony. Často bývají scroll reveal spojeny taky s parallax efekty, které jsou také závislé na rolování stránky, ale jejich podstatou je využívání vrstev stránky. Nicméně scroll reveal disponuje plynulejšími průběhy a vizuálně je jejich použití mnohem příjemnější.

**SEZNAM POUŽITÉ LITERATURY**

- [1] What Is Parallax Web Design? - Definitions, Tips & Considerations. *Unleashed Technologies* [online]. 2013 [cit. 2015-04-02]. Dostupné z: <https://www.unleashed-technologies.com/blog/2013/08/15/what-parallax-web-design-%E2%80%93-definitions-tips-considerations>
- [2] HTML5 Introduction. *W3C School* [online]. 2015 [cit. 2015-04-20]. Dostupné z: [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)
- [3] Webdesignérův průvodce po HTML5 – nová sémantika. *Zdroják.cz* [online]. 2015 [cit. 2015-05-02]. Dostupné z: <http://www.zdrojak.cz/clanky/webdesigneruv-pruvodce-po-html5-nova-semantika/>
- [4] The HTML DOM Document Object. *W3C School* [online]. 2015 [cit. 2015-05-02]. Dostupné z: [http://www.w3schools.com/jsref/dom\\_obj\\_document.asp](http://www.w3schools.com/jsref/dom_obj_document.asp)
- [5] CSS3 INTRODUCTION PUBLICATION HISTORY. *W3C* [online]. [cit. 2015-04-20]. Dostupné z: <http://www.w3.org/standards/history/css3-roadmap>
- [6] CSS3 Transitions. *W3C School* [online]. 2015 [cit. 2015-04-20]. Dostupné z: [http://www.w3schools.com/css/css3\\_transitions.asp](http://www.w3schools.com/css/css3_transitions.asp)
- [7] CSS3 2D Transforms. *W3C School* [online]. 2015 [cit. 2015-04-21]. Dostupné z: [http://www.w3schools.com/css/css3\\_2dtransforms.asp](http://www.w3schools.com/css/css3_2dtransforms.asp)
- [8] CSS3 Animations. *W3C School* [online]. 2015 [cit. 2015-04-21]. Dostupné z: [http://www.w3schools.com/css/css3\\_animations.asp](http://www.w3schools.com/css/css3_animations.asp)
- [9] A Short History of JavaScript. *W3C* [online]. 2012 [cit. 2015-04-23]. Dostupné z: [https://www.w3.org/community/webed/wiki/A\\_Short\\_History\\_of\\_JavaScript](https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript)
- [10] Úvod do JavaScriptu. *Jak psát web* [online]. 2014 [cit. 2015-04-23]. Dostupné z: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>
- [11] ODELL, Den. *JavaScript: průvodce programováním ajaxových aplikací*. Vyd. 1. Brno: Computer Press, 2010, 368 s. ISBN 978-80-251-2733-9.
- [12] JavaScript Events. *W3C School* [online]. 2015 [cit. 2015-04-23]. Dostupné z: [http://www.w3schools.com/js/js\\_events.asp](http://www.w3schools.com/js/js_events.asp)
- [13] THAU, Dave. *Velký průvodce JavaScriptem: tvorba interaktivních webových stránek v praxi*. 1. vyd. Praha: Grada, 2009, 516 s. ISBN 978-80-247-2211-5.

- [14] *ScrollReveal.js: Easily reveal elements as they enter the viewport*. [online]. 2014 [cit. 2015-02-02]. Dostupné z: <http://scrollrevealjs.org/>
- [15] JavaScript Window: The Browser Object Model. W3C School [online]. 2015 [cit. 2015-05-02]. Dostupné z: [http://www.w3schools.com/js/js\\_window.asp](http://www.w3schools.com/js/js_window.asp)
- [16] *WOW.js: Reveal Animations When Scrolling* [online]. [cit. 2015-04-23]. Dostupné z: <http://mynameismatthieu.com/WOW/index.html>
- [17] Infinite Scroll [online]. 2008 [cit. 2015-02-02]. Dostupné z: <http://www.infinite-scroll.com/>
- [18] The Guide to Scrolling Animation Libraries. *Ihatetomatoes.net: Front End Development Blog* [online]. 10.7.2104 [cit. 2015-04-24]. Dostupné z: <https://ihatetomatoes.net/guide-scrolling-animation-libraries/>
- [19] Prinzhorn/skrollr. *GitHub* [online]. 10.7.2104 [cit. 2015-04-24]. Dostupné z: <https://github.com/Prinzhorn/skrollr>
- [20] O WordPressu. *WordPress.cz* [online]. 2015. [cit. 2015-04-27]. Dostupné z: <http://www.iwp.cz/o-wordpressu/>
- [21] *WordPress portál: vše o redakčním systému zdarma* [online]. 2015. [cit. 2015-04-27]. Dostupné z: <http://www.cwordpress.cz/>
- [22] MCNULTY, Scott. *WordPress: efektivní publikování na webu*. Vyd. 1. Brno: Zoner Press, 2009, 256 s. Encyklopedie webdesignera. ISBN 978-80-7413-042-7.
- [23] Writin a Plugin. 2015. *WordPress.org* [online]. [cit. 2015-04-28]. Dostupné z: [https://codex.wordpress.org/Writing\\_a\\_Plugin](https://codex.wordpress.org/Writing_a_Plugin)
- [24] WordPress APIs. 2015. *WordPress.org* [online]. [cit. 2015-04-28]. Dostupné z: [https://codex.wordpress.org/WordPress\\_APIs](https://codex.wordpress.org/WordPress_APIs)
- [25] File Header. 2015. *WordPress.org* [online]. [cit. 2015-04-30]. Dostupné z: [https://codex.wordpress.org/File\\_Header](https://codex.wordpress.org/File_Header)
- [26] Plugin Submission and Promotion. 2015. *WordPress.org* [online]. [cit. 2015-05-13]. Dostupné z: [https://codex.wordpress.org/Plugin\\_Submission\\_and\\_Promotion](https://codex.wordpress.org/Plugin_Submission_and_Promotion)
- [27] WordPress Plugin Activation, Deactivation and Uninstall Hooks. 2015. *QNimate* [online]. [cit. 2015-05-03]. Dostupné z: <http://qnimate.com/wordpress-plugin-activation-deactivation-and-uninstall-hooks/>

- [28] Plugin API. 2015. *WordPress.org* [online]. [cit. 2015-05-03]. Dostupné z: [https://codex.wordpress.org/Plugin\\_API](https://codex.wordpress.org/Plugin_API)
- [29] Template Tags. 2015. *WordPress.org* [online]. [cit. 2015-05-03]. Dostupné z: [https://codex.wordpress.org/Template\\_Tags](https://codex.wordpress.org/Template_Tags)
- [30] Template How to Properly Add JavaScripts and Styles in WordPress. 2015. *WPBeginner* [online]. [cit. 2015-05-03]. Dostupné z: <http://www.wpbeginner.com/wp-tutorials/how-to-properly-add-javascripts-and-styles-in-wordpress/>
- [31] *PHP Simple HTML DOM Parser* [online]. 2015. [cit. 2015-05-09]. Dostupné z: <http://simplehtmldom.sourceforge.net/>
- [32] Template Hierarchy. 2015. *WordPress.org* [online]. [cit. 2015-05-13]. Dostupné z: [https://codex.wordpress.org/Template\\_Hierarchy](https://codex.wordpress.org/Template_Hierarchy)
- [33] *Bootstrap: The world's most popular mobile-first and responsive front-end framework* [online]. 2015. [cit. 2015-05-13]. Dostupné z: <http://getbootstrap.com/>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

API	Application Program Interface
BOM	Browser Object Model – neoficiální standart v JavaScriptu
CSS	Cascading Style Sheets
DOM	Document Object Model
GNU	GNU's Not Unix, Linux OS
GPL	General Public License
HTML	HyperText Markup Language
PHP	Hypertext Preprocessor
RIA	Rich Internet Application
SQL	Structured Query Language
SVG	Scalable Vector Graphics
W3C	World Wide Web Consortium
WHATWG	Web Hypertext Application Technology Working Group
WYSIWYG	What you see is what you get – typ webových textových editorů
XML	Extensible Markup Language

**SEZNAM OBRÁZKŮ**

Obrázek 1-1: Demontrace použití paralaxu na webu intacto10years.com.....	11
Obrázek 1-2: Stromová struktura DOM. ....	15
Obrázek 1-3: Loga HTML5 a CSS3. ....	15
Obrázek 1-4: Princip klientského skriptu. ....	19
Obrázek 2-1: Grafika spojovaná se ScrollReveal.js. ....	21
Obrázek 2-2: Logo Infinite Scroll.....	22
Obrázek 3-1: Logo redakčního systému WordPress.....	25
Obrázek 5-1: Případy užití. ....	37
Obrázek 5-2: Záznam voleb v databázi WordPressu.....	39
Obrázek 5-3: Nastavení tabulky pro ukládání efektů. ....	39
Obrázek 5-4: ScrollReveal.js Plugin jako položka administračního menu. ....	40
Obrázek 5-5: Validací proces přidávání a editování efektu.....	43
Obrázek 6-1: Instalační formulář WordPressu. ....	47
Obrázek 6-2: Nastavení statické stránky. ....	48
Obrázek 6-3: Provázání souborů pro jedinou statickou stránku. ....	49
Obrázek 6-4: Instalační formulář WordPressu. ....	50
Obrázek 7-1: Formulář pro přidání a editaci efektu. ....	52
Obrázek 7-2: Formulář konfigurace knihovny ScrollReveal.js. ....	53
Obrázek 7-3: Použití vFactor parametru konfigurace.....	54
Obrázek 9-1: Konverzace J. Lloyda s uživatelem na GitHub.com.....	57
Obrázek 9-2: Demontrace horizontálního přetečení u efektu. ....	58

**SEZNAM TABULEK**

Tabulka 1-1: Nejčastěji využívané události v JavaScriptu. ....	20
Tabulka 5-1: Seznam použitých akcí a funkcí. ....	46

## SEZNAM PŘÍLOH

Příloha I      Náhled celé demonstrační stránky

Přálaha II     Obsah vloženého CD

# PŘÍLOHA P I: NÁHLED CELÉ DEMONSTRAČNÍ STRÁNKY

H Benčík | homepage | about | [use](#)

# Honza Bendík

promising front-end developer

presents **ScrollReveal.js**  
Plugin for WordPress

## ABOUT

### ScrollReveal Plugin for WordPress

## IS YOUR PORTFOLIO OR BLOG BORING?

Or should you use Julius Lloyd's ScrollReveal.js on your blog? ScrollReveal.js WordPress Plugin is the best choice.

ScrollReveal.js WordPress Plugin is result of Maceo's Thesis from Tomas Bata University developed by student Jan Bendík. Plugin allows:

- using scroll-reveal effects without knowledge of any syntax
- manage used effect in your project
- configure behavior of ScrollReveal.js

For more details see Julius Lloyd's ScrollReveal.js or plugin docs at WordPress.org

## What can ScrollReveal do?

- Move text through the website
- Rotate your pictures
- What kind your coffee is done
- Scaling

and everything you could see when browsing **this site!**

plugin & good graphics  
= a's good choice

benčík.jan@gmail.com  
alzbeta.sung@semmi.cz

## PŘÍLOHA P II: OBSAH VLOŽENÉHO CD

Na zadní straně desek je přiloženo v papírovém obalu CD-ROM, který obsahuje diplomovou práci ve formátu PDF v souboru „fulltext.pdf“. Na CD se nachází i následující složky:

- **scrollreveal\_demo** – obsahuje kompletní zdrojové kódy ukázkového webu.  
V podadresáři *wp-content/plugins/scroll-reveal-js* se nachází všechny zdrojové kódy pluginu ScrollReveal.js Effets. V podadresáři *wp-content/themes/sr-demo* se nachází šablona ukázkového webu.
- **scrollreveal\_database** – obsahuje zálohu MySQL databáze v souboru *sr\_database.sql*