

# **Interaktivní prezentační modul**

## **Interactive slider**

## Zadání bakalářské práce

Student: **Roman Kotyra**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Interaktivní prezentační modul  
Interactive slider**

Zásady pro vypracování:

Cílem práce je realizace administrativního rozhraní pro existující webovou komponentu umožňující interaktivní vizualizaci obsahu, tzv. slider.

1. Zmapujte oblast existujících webových (HTML 5) komponent pro interaktivní prezentaci obsahu a vyberte jedno konkrétní řešení pro další realizaci.
2. Detailně popište vybrané řešení, jeho možnosti a způsob fungování.
3. Navrhněte optimální způsob pro administraci a konfiguraci podkladů pro prezentaci obsahu.
4. Realizujte samostatně fungující front-endovou aplikaci (HTML5, JS, CSS) umožňující vytváření konfigurací a scénářů pro slider.
5. Zhodnoťte vyvinutou aplikaci, a to především z pohledu její integrace do jiných řešení

Seznam doporučené odborné literatury:

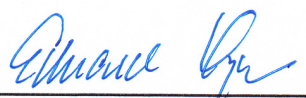
- [1] P. Lubbers, HTML5 - Programujeme moderní webové aplikace, 2011, CPRESS, ISBN: 9788025135396
- [2] Brian P. Hogan, HTML5 a CSS3 - Výukový kurz webového vývoje, 2011, CPRESS, ISBN: 9788025135761
- [3] [www.html5.cz](http://www.html5.cz)

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Michal Radecký, Ph.D.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015

  
doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. Května 2015

Roman Kohyňa

Mé poděkování patří panu Ing. Michalovi Radeckému, Ph.D. za odborné vedení mé bakalářské práce, ochotnou spolupráci, řadu podnětných nápadů a věnovaný čas.

## **Abstrakt**

Tato bakalářská práce se zabývá problematikou tvorby interaktivních prezentací (slider) použitelných na webových stránkách. V této práci jsou popsány dostupné JS komponenty, umožňující integraci prezentací do stránek. Cílem této práce je navrhnout a vytvořit HTML5 administrační aplikaci, která poskytne uživatelsky přívětivé prostředí, postavenou právě na jedné z dostupných komponent. Výsledkem je aplikace, kterou je možné integrovat do systémů třetích stran, umožňující tvorbu prezentací.

**Klíčová slova:** prezentační modul, internet, html5, JavaScript, jQuery

## **Abstract**

This bachelor thesis deals with the creation of interactive presentations (sliders) applicable on the website. This work describes the available JS components designed to integrate into website. The aim of this work is to design and build HTML5 administration application that provides a user friendly environment built on just one of the available components. The result is an application that can be integrated into third-party systems, allowing the creation of presentations.

**Keywords:** slider, internet, html5, JavaScript, jQuery

## **Seznam použitých zkratk a symbolů**

CSS	– Cascading Style Sheets
HTML	– HyperText Markup Language
JS	– JavaScript
JSON	– JavaScript Object Notation
JPG	– Joint Photographic Experts Group
PNG	– Portable Network Graphics
CMS	– Content Management System

---

## Obsah

<b>1</b>	<b>Úvod</b>	<b>4</b>
<b>2</b>	<b>Slider</b>	<b>5</b>
<b>3</b>	<b>Prezentační moduly pro webové stránky</b>	<b>8</b>
3.1	Dostupné komponenty (moduly) a jejich porovnání . . . . .	8
3.2	Výběr a detailní popis komponenty . . . . .	10
<b>4</b>	<b>Návrh a implementace administračního rozhraní</b>	<b>13</b>
4.1	Analýza . . . . .	13
4.2	Použité knihovny . . . . .	15
4.3	Implementace . . . . .	17
4.4	Problémy při implementaci . . . . .	28
<b>5</b>	<b>Použití administračního rozhraní v praxi</b>	<b>29</b>
5.1	Možnosti dalšího vývoje . . . . .	30
<b>6</b>	<b>Závěr</b>	<b>31</b>
<b>7</b>	<b>Reference</b>	<b>32</b>
	<b>Přílohy</b>	<b>32</b>
<b>A</b>	<b>Příloha ve formě CD média</b>	<b>33</b>
<b>B</b>	<b>Konfigurační parametry</b>	<b>34</b>

---

## Seznam obrázků

1	Microsoft.com - Titulní stránka se sliderem . . . . .	5
2	Kasa.cz - Titulní stránka se sliderem . . . . .	6
3	Ukázka slideru vytvořeného pomocí LayerSlider pluginu . . . . .	11
4	jQuery MiniColors . . . . .	17
5	Konfigurační záložka . . . . .	19
6	Náhled snímku . . . . .	21
7	Správa snímku . . . . .	22
8	Výběr typu obsahu vrstvy . . . . .	23
9	Výběr vstupního a výstupního efektu vrstvy . . . . .	25
10	Časové osy jednotlivých vrstev v rámci jednoho snímku . . . . .	26
11	Ukázka exportu . . . . .	27



## Seznam výpisů zdrojového kódu

1	Integrace LayerSlider do stránky . . . . .	10
2	Ukázka kódu slideru na obrázku 3 . . . . .	12
3	jQuery event handler . . . . .	18
4	Nahrání obrázků přetáhnutím (drop) . . . . .	24
5	Funkce pro importování stylů z šablony . . . . .	25

# 1 Úvod

V dnešní době moderních informačních technologií patří internet a webové stránky mezi velmi oblíbené témata. Žijeme v době, kdy se můžeme téměř kdykoliv a kdekoliv připojit na internet hlavně díky rychlým bezdrátovým sítím a mobilním zařízením. Tato oblast je neustále rozvíjena a v běžném životě se s ní pravidelně dostáváme do kontaktu. Webové stránky a aplikace patří mezi nejpoužívanější internetové služby umožňující sdílet informace mezi uživateli.

S rozvojem internetu se rozvíjí i možnosti podnikání na internetu. Internet se tak stal oblíbeným marketingovým nástrojem, který dává možnost i malým firmám a živnostníkům se veřejně reprezentovat a nabízet vlastní služby nebo produkty. Velmi oblíbeným se stalo obchodování přes internet, kde vzniklo silné konkurenční prostředí. Proto se hledají neustále nové možnosti jak zaujmout klienta a odlišit se od konkurence.

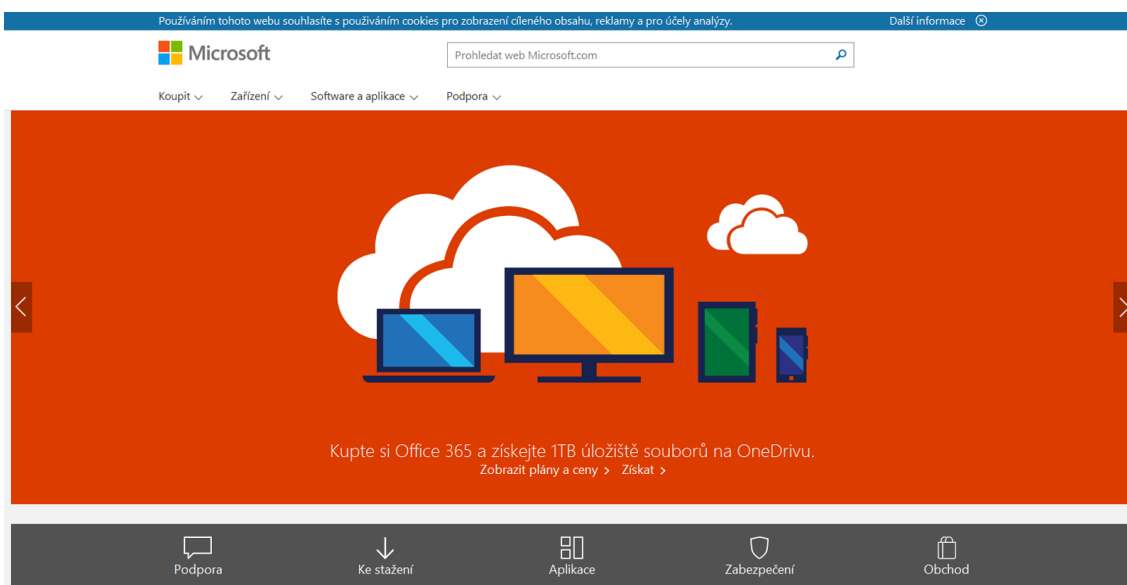
Tato bakalářská práce řeší problematiku tvorby interaktivních prezentací, pomocí kterých můžeme upoutat pozornost návštěvníka internetových stránek. Ve světě se pro tyto prezentace používá spíše pojem slider a budu jej používat i v této práci, protože neexistuje žádný ekvivalent v českém jazyce. Tento pojem je znám jak mezi vývojáři webových stránek, tak i jejich klienty, kteří tuto funkcionalitu na svých stránkách vyžadují. Slovo slider lze také přeložit jako posuvník obsahu nebo konkrétněji jako posuvník obrázků.

Jedním z důvodů, proč jsem si zvolil tuto práci, je zájem o problematiku tvorby moderních webových stránek, vývoje internetových obchodů a s tím i související nové trendy a způsoby prezentace. Cílem této práce je zmapovat oblast interaktivních prezentací a poté navrhnout a implementovat administrační rozhraní pro tvorbu zmiňovaných prezentací. Základním předpokladem pro implementaci aplikace je využití aktuálních technologií jako je HTML5, JS, CCS3.

## 2 Slider

Pojem slider nebo také slideshow je v oblasti moderních webových stránek často používaný a známý termín. Jednoduchý slider lze popsat jako posuvník (prezentaci) série obrázků. Smyslem slideru je obvykle upoutat návštěvníka stránek a podsunout mu zajímavý obsah, na který klikne. Jedná se o způsob jak návštěvníka přesměrovat na požadovanou stránku nebo obsah a zabránit tak okamžitému opuštění stránky. Slider umožňuje na relativně malém prostoru zobrazit zajímavý obsah, především graficky propracované obrázky. Slider je většinou umístěn na titulní stránce webu ještě před hlavním obsahem, ale není to podmínkou. Slider lze umístit prakticky v kterékoliv části webové stránky.

Jednoduché slidery obvykle umí prezentovat pouze obrázky, takže je nutná tvorba obrázků v grafickém programu. Tento typ slideru obvykle neumožňuje zobrazovat textové vrstvy ve formátu HTML, takže jsou pro internetové vyhledávače téměř nečitelné. Navíc tento způsob realizace poskytuje jen minimální míru interakce.

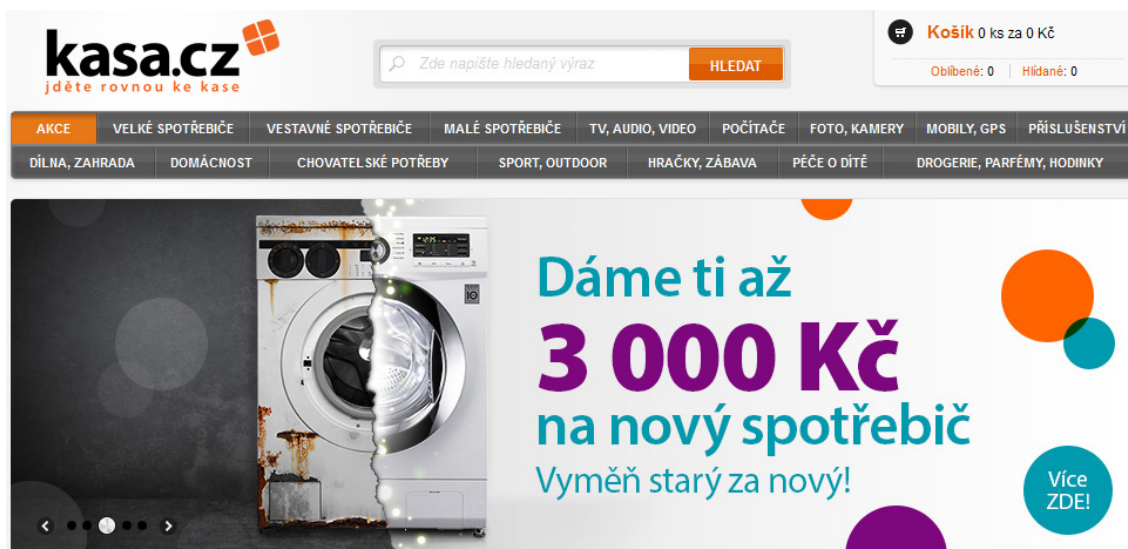


Obrázek 1: Microsoft.com - Titulní stránka se sliderem

S příchodem nových technologií, jako je HTML5, CSS3 je možné za pomoci javascriptu vytvořit animované interaktivní prezentace, které umožňují zobrazovat různorodý obsah - text, obrázky, videa a to i v několika vrstvách na jednom snímku. Pokud si vezmeme slider jako jednu prezentaci, lze jej rozdělit na jednotlivé snímky. Každý snímek má své pozadí, obvykle obrázek, přechod nebo jen barvu. Ve snímcích je možné skládat jednot-

livé vrstvy. Do vrstvy může být vložen text, obrázek, libovolný HTML obsah a tedy i video. Tyto vrstvy lze různě animovat a časovat. Místo obyčejného statického obrázku lze v podstatě vytvořit animace. Slider poskytuje možnost promítat jednotlivé snímky automaticky bez nutnosti uživatelského vstupu. Snímky si může uživatel přepínat pomocí tlačítek „předchozí/další“ nebo lze zvolit snímek kliknutím na miniaturu (podle nastavení a typu slideru). Na obrázku 2 lze vidět jak slider vypadá a jak se používá v praxi. Zde vidíme, že je snaha upoutat návštěvníka na výhodnou nabídku. V ideálním případě návštěvník klikne na jeden z odkazů a budou mu zobrazeny další informace. Tímto způsobem se sníží míra okamžitého opuštění stránky a o to se zvýší šance na vytvoření konverze. Konverze nastává, když se návštěvník promění v zákazníka.

Nevýhodou sliderů je jejich zvýšená datová náročnost, musí se navíc stáhnout jednotlivá data ve snímcích (hlavně obrázky) a také se musí stáhnout javascriptové knihovny. Proto by se měl slider navrhovat tak, aby nebyl pro stránky přítěží. Je třeba také myslet na mobilní zařízení, které mají obvykle slabší výkon než běžný počítač a také horší připojení k internetu. Jednou z vlastností slideru by měla být schopnost přizpůsobit se responzivnímu zobrazení a postupné načítání obrázků (lazy loading)<sup>1</sup>



Obrázek 2: Kasa.cz - Titulní stránka se sliderem

Slider lze na webových stránkách realizovat několika způsoby. Dříve se hojně používala technologie Flash, ale po příchodu HTML5 se od technologie Flash postupně ustupuje. Ačkoliv lze díky kombinaci HTML a JavaScriptu vytvořit slider bez knihoven

<sup>1</sup>lazy loading = technika, při které se načítá obsah, až když jej uživatel chce zobrazit

třetích stran, na trhu existuje mnoho hotových jQuery knihoven. Tyto knihovny poskytují velké množství funkcí a možností. Vývoj takové knihovny se nevyplatí a ani nemá smysl vymýšlet nebo vyvíjet vymyšlené. Nevýhodou těchto knihoven je neexistující administrační rozhraní pro jednoduché generování sliderů. Slidery lze tedy vytvořit jen napsáním HTML kódu podle přiložené dokumentace dané knihovny a z toho vyplývá, že pro méně zkušené správce webových stránek jde o nepřekonatelný problém a i pro zkušené programátory se jedná o časově náročnou činnost.

Cílem této práce je tedy vybrat vhodnou knihovnu a pro tuto knihovnu navrhnout a implementovat jednoduché a intuitivní administrační rozhraní, které bude vhodné pro integraci do dalších systémů, například do redakčních systémů CMS, administrace internetových obchodů apod. Toto administrační rozhraní má sloužit jako generátor zmíněných sliderů. Administrační rozhraní bude navrženo na míru možnostem dané knihovny, proto je nutné věnovat pozornost při výběru použitého pluginu. Proto v následující kapitole blíže popisují dostupné knihovny, jejich vlastnosti a funkce. Pro nejvhodnější knihovnu je navržena administrační aplikace, která je hlavním záměrem této bakalářské práce.

## 3 Prezentační moduly pro webové stránky

Cílem této práce není vytvořit samotný prezentační modul, nýbrž efektivně využít již existující plugin a vytvořit administrační rozhraní postavené právě na tomto pluginu. Slidery jsou v oblasti internetových stránek velmi oblíbené. Existuje poměrně široký výběr pluginů, nicméně těch kvalitních a ověřených je jen několik. Tady je třeba počítat, že kvalitní plugin bude zpoplatněn. V této práci se zaměřuji na pluginy postavené na knihovně jQuery. Knihovnu jQuery popisuji v další části práce.

### 3.1 Dostupné komponenty (moduly) a jejich porovnání

Tato kapitola se zabývá nejznámějšími pluginy pro generování sliderů. Tyto pluginy otestuji a zhodnotím z hlediska použití pro tuto práci. Nejvhodnější plugin použiji pro implementace administračního rozhraní.

#### 3.1.1 Camera slideshow

**Webové stránky:** <http://www.pixedelic.com/plugins/camera/>

**Testovaná verze:** 1.3.3

**Licence:** MIT - Zdarma ke stažení, volně šiřitelné

**Velikost skriptů:** 259kB

Camera slideshow je volně dostupný plugin, který umožňuje tvořit jednoduchý posuvník s obrázky, nicméně nepodporuje vrstvy. Z tohoto důvodu je pro tuto práci nevhodný, ale pro jednodušší projekty je použitelný. K dispozici je 33 připravených stylů ovládacích prvků.

#### 3.1.2 Revolution Slider

**Webové stránky:** <http://www.themepunch.com/>

**Testovaná verze:** 4.6.0

**Licence:** Envato Market - Pro každý projekt (webovou stránku) je nutné zaplatit licenci[7]

**Velikost skriptů:** 279kB

Slider Revolution patří mezi nejznámější jQuery komponenty umožňující realizaci interaktivních prezentací.[1] Tento plugin umožňuje zobrazení téměř jakéhokoliv druhu obsahu včetně textu, obrázků, videí. Na obsah dokáže aplikovat přizpůsobitelné efekty a animace. Podporuje také Parallax efekt, při kterém jednotlivé prvky reagují na pohyb myši změnou pozice. Splňuje požadavky pro použití v této práci. Nevýhodou je poměrně velká velikost skriptů, z testovacích pluginů je největší.

### 3.1.3 Masterslider

**Webové stránky:** <http://www.masterslider.com>

**Testovaná verze:** 2.1.2

**Licence:** Envato Market - Pro každý projekt (webovou stránku) je nutné zaplatit licenci[7]

**Velikost skriptů:** 257kB

Masterslider, jQuery plugin, poskytuje podobné funkce jako Revolution Slider, má navíc lepší podporu videí, včetně HD videa na pozadí. Nevýhodou je malý výběr efektů pro snímky, kterých je v popsané verzi pouze 6.

### 3.1.4 LayerSlider

**Webové stránky:** <http://www.kreaturamedia.com/>

**Testovaná verze:** 5.2.0

**Licence:** Envato Market - Pro každý projekt (webovou stránku) je nutné zaplatit licenci[7]

**Velikost skriptů:** 126kB

---

LayerSlider je jQuery plugin<sup>2</sup>, který umožňuje zobrazit na webových stránkách víceúčelové prezentace (slideshow). [2] LayerSlider má implementováno přes 200 2D a 3D efektů při přechodu mezi snímky, výbornou podporu pro mobilní zařízení, včetně podpory responzivního zobrazení, tzn. že velikost prezentace se přizpůsobí rozlišení displeje zařízení. Mimo jiné podporuje také ovládání pomocí gest (mobilní zařízení), hlavně pro posouvání mezi snímky. Pro optimalizaci výkonu a datových toků, používá technologii lazy-loading, takže všechny obrázky se stahují, až když jsou potřebné k zobrazení. V každém snímku může být vložen jakýkoliv text, HTML obsah, obrázky, YouTube nebo Vimeo videa nebo HTML5 média. LayerSlider má 13 připravených skinů ovládacích prvků. [2]

## 3.2 Výběr a detailní popis komponenty

Pro tuto práci jsem se rozhodl použít plugin LayerSlider. Hlavním důvodem byla výborná dokumentace pluginu, z testovaných skriptů má nejlepší poměr funkce/velikost. LayerSlider má ve srovnání s pluginem Slider Revolution poloviční velikost. Nabízí velké množství efektů (přes 200 2D a 100 3D efektů) s možností vytvářet vlastní efekty. Další detaily tohoto pluginu popisují v další části práce.

### 3.2.1 Implementace do stránek

Vzhledem k tomu, že se jedná o jQuery plugin, samozřejmostí je integrace jQuery knihovny do webové stránky, kde bude prezentace zobrazena. Plugin LayerSlider musí být nahrán na serveru stránek a musí být integrován v HTML.

---

```
<script type="text/javascript">
  // jQuery("document").ready(); function
  jQuery("document").ready(function() {

    // Calling LayerSlider on your selected element after the document loaded
    jQuery(selector).layerSlider ();

  });
</script>
```

---

#### Výpis 1: Integrace LayerSlider do stránky

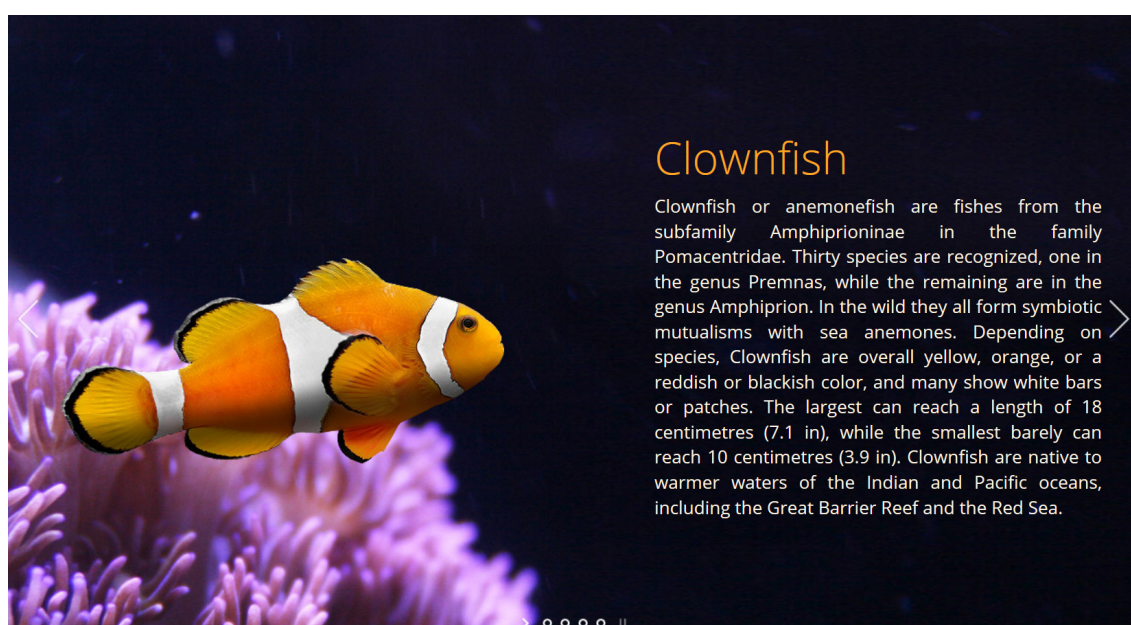
---

<sup>2</sup>plugin = zásuvný modul (komponenta)



### 3.2.2 Snímek

Základním prvkem prezentace vytvořené pomocí LayerSlider je snímek. Všechny snímky v jedné prezentaci mají stejné základní vlastnosti, těmito vlastnostmi jsou rozměry snímku a pozadí snímku. Prezentace tohoto charakteru se zakládají na obrázkovém pozadí. Každý snímek má možnost nastavit obrázkové pozadí ve formátu JPG nebo PNG. Rozměr obrázku by měl souhlasit s globálním nastavením prezentace. Mezi další konfigurační parametry patří doba trvání snímku a seznam efektů, které se použijí při změně snímku. Počet snímků v jedné prezentaci je neomezený, ale pro použitelnost na mobilních zařízeních nemá smysl použít více než 10 snímků v jedné prezentaci.



Obrázek 3: Ukázka slideru vytvořeného pomocí LayerSlider pluginu

### 3.2.3 Vrstva

Vrstvu lze považovat za obsah snímku. Každý snímek může mít teoreticky libovolný počet vrstev a obsahem vrstvy je text, HTML kód, obrázek nebo video. U vrstvy je důležitý údaj její výsledná pozice (umístění) ve snímku. Doba trvání snímku je důležitá pro nastavení časování jednotlivých vrstev. Kromě časování vrstvy se nastavuje vstupní a výstupní efekt.

**Fáze vrstvy:**

- vrstva není zobrazena,
- probíhá vstupní efekt,
- vrstva je zobrazena,
- probíhá výstupní efekt,
- vrstva není zobrazena.

**Pro vrstvu se nastavují tyto časy:**

- zpoždění (delayin) - určuje zpoždění zahájení vstupního efektu,
- trvání vstupního efektu (durationin),
- doba zobrazení vrstvy (showuntil) - pokud není nastaveno, výstupní efekt se nespustí,
- trvání výstupního efektu (durationout).

---

```

<div id=" layerslider " style="width:1280px;height:720px;max-width:1280px;background:grey;">
  <div class="ls-slide" data-ls="slidedelay:5000;transition2d:104;">
    
    
    <h1 class="ls-l" style="top:150px;left:730px;font-size:50px;color:#ffa724;"
    data-ls="offsetxin:0;durationin:3000;delayin:1500;rotatexin:90;">
    Clownfish</h1>
    <div class="ls-l" style="top:220px;left:730px;width:500px;font-size:20px;color:#fff5e5;"
    data-ls="offsetxin:150;durationin:2000;rotateyin:40;offsetxout:150;rotateyout:40;">
      Clownfish or anemonefish are fishes from the subfamily Amphiprioninae in the family
      Pomacentridae ..... </div>
  </div>
</div>

```

---

Výpis 2: Ukázka kódu slideru na obrázku 3

## 4 Návrh a implementace administračního rozhraní

Hlavním cílem této práce je navrhnout a zrealizovat aplikaci, jejímž smyslem je usnadnění tvorby sliderů. Ve výpisu zdrojového kódu č.2 z předchozí kapitoly je vidět, jak je složitý a nepřehledný kód specifikující slider. Napsat takovou specifikaci slideru bez přiložené dokumentace je v podstatě nemožné a i s dokumentací se nejedná o jednoduchý úkol. Tento problém má řešit výsledek této práce, tedy aplikace pro jednoduchou tvorbu sliderů. Tato kapitola se zabývá zmíněnou aplikací, od návrhu až po realizaci.

### 4.1 Analýza

V této části práce se zaměřím na základní funkce, požadavky na aplikaci, problematiku návrhu uživatelsky přívětivého prostředí. Aplikace je vyvíjena se zacílením na široké spektrum uživatelů, jak specialistům na vývoj webových stránek, tak i běžným uživatelům, kteří mají jen základní znalosti a chtějí spravovat své webové stránky. Předpokládaný výsledek je samostatně fungující front-endová aplikace, která využívá technologie HTML5, CSS3 a JS. Jedním z požadavků je implementovat aplikaci tak, aby byla možná její integrace do systému třetích stran, hlavně do administračních rozhraní redakčních systémů (CMS) a internetových obchodů, proto je aplikace navrhována bez nutnosti použití serverových scriptů nebo databáze.

#### 4.1.1 Funkce aplikace a možnosti

##### Globální nastavení

Před samotnou tvorbou obsahu slideru je nutné nastavit jeho základní parametry, podstatné údaje jsou hlavně šířka a výška. Veškeré základní parametry se nastavují na konfigurační stránce.

##### Správa snímků

Základem aplikace je správa snímků, protože snímek je základní prvek prezentace. Seznam všech snímků bude stále viditelný pro rychlou správu a přepínání mezi snímky.

**Mezi základní funkce ve správě snímků patří:**

- vytvoření nového snímku,

- editace snímku,
- odstranění snímku,
- změna pořadí snímků.

Po zvolení snímku se zobrazí konfigurace snímku, náhled a možnost správy vrstev v daném snímku.

#### **Ve správě vrstev jsou zahrnuty funkce:**

- vytvoření nové vrstvy,
- editace vrstvy,
- odstranění vrstvy,
- změna pořadí vrstvy.

#### **Export / Import**

Aplikace bude generovat HTML kód, který se poté vkládá na požadované místo na webu. Vzhledem k velkému počtu parametrů, které se uchovávají při tvorbě slideru je nutné, aby aplikace poskytla rozhraní pro uložení rozpracované práce a následné znovunačtení. Toto je nutné i pro případné editace již vytvořeného slideru.

#### **Export lze rozdělit na části:**

- export HTML části slideru obsahující veškerý obsah, který návštěvník stránek uvidí,
- export inicializačního JavaScriptu, který volá na daný obsah jQuery funkci daného pluginu,
- export JSON kódu, který lze použít pro následný import slideru zpět do administrativního rozhraní.

Vzhledem ke složitosti nastavení efektů vrstvy, umožním v aplikaci dva režimy. Standardní režim bude určen pro běžné uživatele a nastavení bude zjednodušené tak, že pro konfiguraci vrstev bude možné využít předpřipravené šablony. Šablony je možné definovat pro efekt vrstvy a pro styl vrstvy. Expertní režim umožní nastavit veškeré parametry ručně.

## Layout

Administraci bych realizovat formou přepínacích záložek. Tento způsob dovoluje dynamicky přidávat další konfigurační stránky a zanechává v aplikaci určitou přehlednost. Budu rozlišovat dva typy záložek:

- konfigurace slideru, tzv. inicializační parametry a
- správa jednotlivých slidů (snímků)

Cílem této práce není navržení serverové části, tato část je probrána v poslední kapitole. Aplikace musí být implementována jako samostatně fungující. Veškerá aplikační logika bude realizována jako kombinace HTML a JavaScriptu.

## 4.2 Použité knihovny

V této části práce se zaměřím na knihovny, které jsem využil při implementaci administračního rozhraní. Popisuji zde jejich základní vlastnosti a možnosti. Výhodou těchto knihoven je ulehčení implementace, přehlednější kód a moderní vzhled.

### 4.2.1 jQuery

jQuery je JavaScriptová knihovna, podporující CCS3 a většinu dostupných internetových prohlížečů, včetně mobilních. jQuery knihovna umožňuje vývojářům psát JavaScriptové funkce jednodušeji a efektivněji. Knihovna je dostupná pod licencí MIT, je volně šiřitelná, takže je vhodná i pro komerční použití. jQuery patří mezi nejpoužívanější[3] JavaScriptovou knihovnu, výhodou je její výborná rozšiřitelnost a proto se bez její přítomnosti neobejde velké množství JS pluginů, které jsou založené právě na jQuery. Projekt jQuery disponuje mimo jiné výbornou dokumentací dostupnou veřejně na internetu a vzdělávacím centrem.

**Mezi hlavní funkce a vlastnosti této knihovny patří:**

- výběr, přístup k DOM elementům HTML stránky,
- funkce pro změnu DOM elementů,
- práce s událostmi,
- efekty, animace,

- AJAX.

### 4.2.2 jQuery UI

jQuery UI je javascriptový balíček nástrojů, který je zaměřen na uživatelské rozhraní (UI = user interface). Jedná se o součást projektu „The jQuery Project“. Tento balíček umožňuje implementovat pokročilé interakce mezi uživatelem a rozhraním webové aplikace:

- draggable - přesouvat objekty na stránce,
- droppable - implementace drag and drop,
- resizable - měnit velikosti DOM elementů,
- sortable - měnit pořadí objektů na stránce.

### 4.2.3 Bootstrap

Bootstrap je jeden z nejznámějších HTML, CSS a JavaScript balíčků nástrojů pro tvorbu moderních webových stránek.[4] Bootstrap je vyvíjen a distribuován firmou Twitter, Inc pod licencí MIT. Je závislý na jQuery knihovně.

S tímto balíčkem nástrojů je možné rychle a efektivně vytvářet webové stránky bez nutnosti návrhu, stylování nebo programování základních elementárních prvků jako jsou tlačítka, formuláře, apod.

Při vývoji aplikace se lze zaměřit na její aplikační logiku a nezabývat se do hloubky s návrhem základních stylů, elementů funkcí. Výborná je také podpora a optimalizace pro nejpoužívanější internetové prohlížeče.

### 4.2.4 jQuery MiniColors

Plugin MiniColors je velmi užitečný nástroj pokud potřebujeme v aplikaci umožnit uživatelům jednoduchý výběr barvy. Tento plugin po kliknutí na vybrané vstupní pole zobrazí panel s paletou barev. Po výběru barvy z palety se vloží do vstupního pole kód barvy. Součástí pluginu jsou i CSS styly podporující Bootstrap, ukázkou pluginu vidíme na obrázku 4.

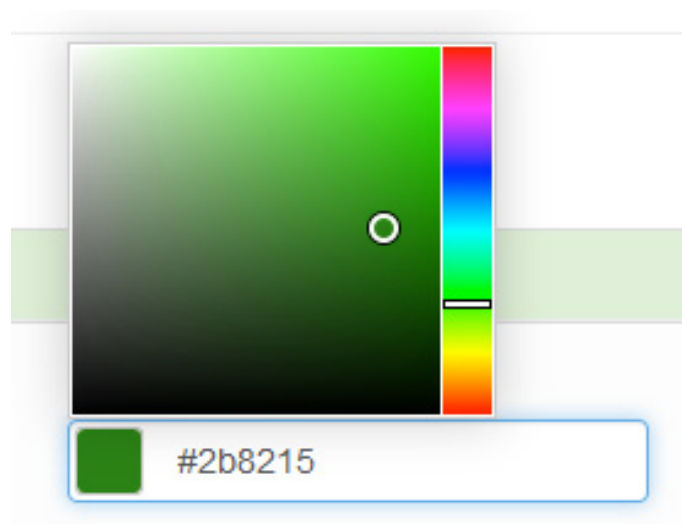
### 4.2.5 FileSaver.js

Vzhledem k tomu, že aktuální specifikace HTML5 neposkytuje API pro ukládání dat vytvořených pomocí JavaScriptu a vývoj specifikace File API Writer byl pozastaven[5], tak se vývojář Eli Grey rozhodl napsat JavaScriptovou knihovnu FileSaver umožňující vygenerovat jakýkoliv typ souboru ve většině aktuálních prohlížečů bez potřeby komunikace se serverem.[6] V této knihovně je implementována funkce „SaveAs“, která má dva vstupní parametry, prvním parametrem je BLOB<sup>3</sup> a druhým parametrem je název souboru. Po zavolání této funkce se otevře dialog vyzývající k uložení souboru.

### 4.3 Implementace

Aplikace je tvořena jedním HTML souborem (index.html), který obsahuje základní strukturu stránky, formulář pro základní nastavení slideru a šablony pro vytvoření snímku a vrstvy. Veškerá dynamika na stránce je realizována pomocí JavaScriptu. Hlavní JavaScriptový kód (admin.js) lze rozdělit na dvě pomyslné části. První část je objekt „AdminLayerSlider“, který obsahuje veškeré funkce potřebné pro běh a ovládání aplikace. V druhé části se nacházejí tzv. „event handlers“ – funkce které se spustí na základě definované události (například po kliknutí na tlačítko).

<sup>3</sup>BLOB = Binary Large Object, rozhraní nebo objekt reprezentující binární data



Obrázek 4: jQuery MiniColors

Aplikace má možnost přepínání mezi základním režimem a expertním režimem. Tyto režimy ovlivňují především konfiguraci vrstev, z důvodu složitosti nastavení efektů, tento bod je popsán v další části práce.

Vzhledem k tomu, že aplikace nemá serverovou část a nevyužívá žádnou externí databázi (např. MySQL), všechny data jsou uloženy ve vstupních polích (input) v rámci HTML. V případě potřeby data uchovat, je nutné použít funkci pro export dat.

```
// add layer
jQuery('#slides-content').on('click', '.add-layer', function(e) {
    e.preventDefault();
    AdminLayerSlider.addLayer(this);
});
```

Výpis 3: jQuery event handler

**Layout stránky je rozdělen na dvě části:**

- část pro správu snímků ve formě levého postranního panelu,
- hlavní obsahovou část.

Vzhled aplikace lze vidět na obrázku 5. Stránka obsahuje skrytý (pomocí CSS) HTML oddíl<sup>4</sup> s ID „templates“, který obsahuje šablony pro vytvoření nových snímků a nových vrstev. Při této události se používá funkce „clone“ z jQuery knihovny, která vytvoří nový DOM element a ten se pomocí funkce „appendTo“ vloží na správné místo.

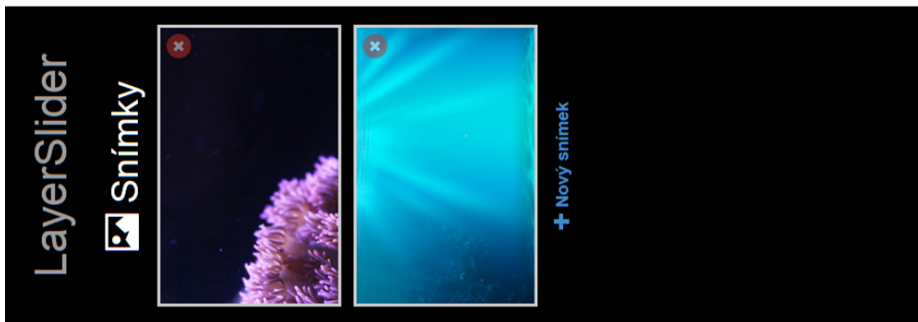
Obrázky lze vkládat do aplikace buď jako odkaz na internetový zdroj ve formě URL nebo lze nahrát pomocí specifikace FileAPI z HTML5, která umožňuje komunikovat s lokálními soubory v počítači. Po výběru obrázku v PC se obrázek převede pomocí kódování Base64 do textového řetězce.

### 4.3.1 Správa snímků

Jednotlivé snímky jsou zobrazeny jako malé náhledy (pro lepší orientaci), kdy pozadím je buď výchozí barva, nastavená v konfiguraci, nebo nahraný obrázek, který slouží jako pozadí snímku. Vytvořené snímky lze libovolně seřazovat, tato funkce byla implementována pomocí knihovny jQuery UI a její funkcionality „draggable“. Při vytvoření nového

<sup>4</sup>oddíl = html tag <div>





Konfigurace
Export
Import
Vyčistit

## Konfigurace

### Layout

Šířka

Výška

Responsivní  ANO  NE

### SlideShow

SlideShow AutoStart  ANO  NE

Pauza při hoveru  ANO  NE

### Vzhled

Skin

Pozadí (background)

Obrázek 5: Konfigurační záložka

snímku se automaticky stane aktivním a zobrazí se jeho konfigurační stránka, která je pomyslně rozdělená na části:

- konfigurace snímku,
- náhled snímku,
- seznam a správa vrstev.

## Konfigurace

Každý snímek má možnost nastavit dobu trvání snímku, efekt prolínání a obrázkové pozadí. Časování snímku je jednoduché, zadává se čas v ms do běžného vstupního pole. Obrázek lze nahrát buď přímo z počítače (zařízení) nebo lze vložit formou URL<sup>5</sup> adresy. Při načítání obrázku z PC se používá instance třídy FileReader z HTML5, která umožňuje asynchronně číst data v lokálním zařízení. Z objektu třídy FileReader se využívá metoda ReadAsDataURL. Po dokončení nahrání obrázku je možné získat řetězec, který reprezentuje obrázek zakódovaný pomocí base64, ke kterému se přistupuje pomocí atributu result objektu FileReader. Po vložení obrázku ať už pomocí URL nebo FileReader objektu se výsledná adresa uloží do skrytého pole formuláře, který slouží k uchování dat. K tomuto poli poté přistupují další funkce aplikace.

## Náhled

Náhled usnadňuje práci při tvorbě prezentace, v náhledu je zobrazeno nastavené pozadí snímku a všechny vytvořené vrstvy s jejich výslednou pozicí. Náhled je generován vždy při změně některých ze základních parametrů nebo při změně jakékoliv vrstvy v daném snímku.

### Základní parametry pro generování náhledu jsou:

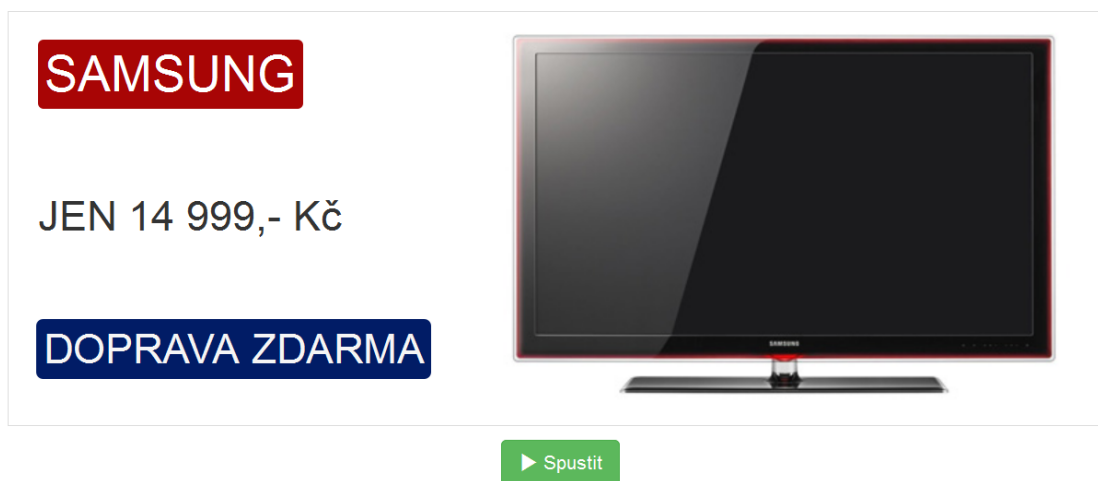
- šířka prezentace (slideru),
- výška prezentace (slideru),
- pozadí prezentace.

---

<sup>5</sup>URL = Unique Resource Locator, čili jednoznačné umístění zdroje na internetu

Generování náhledu probíhá tak, že se v cyklu prochází všechny snímky a každý cyklus obsahuje vnořený cyklus, který prochází parametry efektů a styly vrstvy. Tento způsob umožňuje poměrně jednoduše pracovat s daty bez nutnosti ukládání dat do proměnných apod. Celá aplikace se chová jako dynamicky generovaný formulář.

 Náhled



Obrázek 6: Náhled snímku

V náhledu je umožněno nastavit pozici vrstvy. Při kliknutí na prvek se aktivuje daná vrstva a zobrazí se její nastavení, pomocí kurzoru myši lze vrstvu přesunout na libovolné místo v náhledu. Tato funkce byla realizována pomocí metody `draggable` z balíčku `jQuery UI`. Využil jsem události `drag` a `stop`, které volají funkci v aplikaci, která má za úkol zjistit aktuální pozici vrstvy a tyto hodnoty vložit do vstupních polí udávající pozici. Ačkoliv by tato funkce šla realizovat pomocí `HTML5`, které poskytuje funkcionality `drag` a `drop`, tak jsem se rozhodl využít funkci z balíčku `jQuery UI`, kde je připravené funkční, ověřené API a používám z tohoto balíku i další funkce.

V této části je také možnost spustit ukázkou celého slideru. Při spuštění reálné ukázkou se skryje původní náhled a vloží se nový oddíl, do kterého se vygeneruje kompletní kód slideru podle dodané dokumentace pluginu `LayerSlider`. Jakmile je kód připraven, tak se zavolá funkce `layerSlider`. Výsledkem je zobrazení kompletního slideru, včetně všech ovládacích tlačítek, tak jako by byl integrován ve finální stránce.

The image shows a software interface for video editing. At the top, there is a 'LayerSlider' panel with a 'Snímky' (Frames) section containing two thumbnails: one of coral and one of a blue underwater scene. Below this is the 'Konfigurace snímku' (Frame Configuration) panel, which includes a 'Doba trvání (ms)' (Duration) field set to 4000, a 'Pozadí' (Background) preview window showing coral, an 'Efekt snímku' (Frame Effect) dropdown menu, and a 'Vybrat efekty' (Select effects) button. To the right of the configuration panel is the 'Náhled' (Preview) window, which displays a large video frame with two clownfish swimming in front of a dark background with purple coral. A 'Spustit' (Start) button is located at the bottom right of the preview window.

Konfigurace

Export

Import

Vyčistit

Konfigurace snímku

Doba trvání (ms)

4000

Pozadí

Efekt snímku

Vybrat efekty ▾

Náhled

Spustit

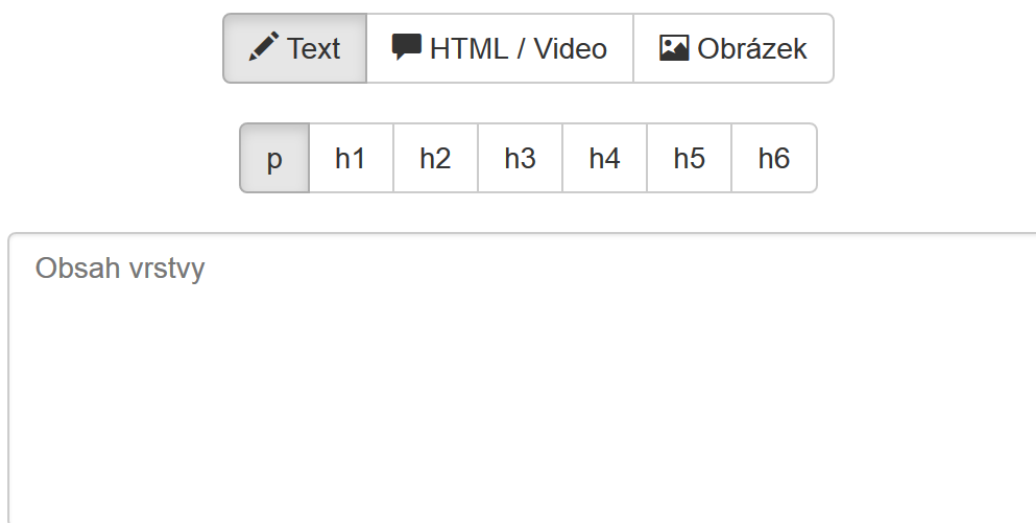
Nový snímek

Obrázek 7: Správa snímku

## Správa vrstev

Správu vrstev lze rozdělit na tři pomyslné části. První část je správa obsahu vrstvy, kde definujeme, co se v dané vrstvě bude zobrazovat. Další částí je konfigurace vstupního efektu, výstupního efektu a časování vrstvy. Poslední část umožňuje nastavit vzhled vrstvy, to se týká hlavně textové vrstvy, kde můžeme zvolit CSS styly. Volba obsahu je implementována pomocí přepínacích tlačítek, na výběr jsou tři možnosti:

- **Text** - v tomto případě se navíc zobrazí další přepínací tlačítka, které umožňují výběr HTML tagu, který bude pro tento text použit:  
h1, h2, h3, h4, h5, h6, p
- **HTML** - umožňující vkládat do vrstvy jakýkoliv HTML obsah, např. HTML5 média (tag „<video>“), iframe (YouTube video, Vimeo video, mapu, apod.)
- **Obrázek** - zobrazí se drag&drop prostor pro vložení obrázku, popřípadě kliknutím lze vybrat obrázek z počítače, drag&drop je realizováno pomocí HTML5, podobně jako u pozadí snímku, za použití objektu FileReader



Obrázek 8: Výběr typu obsahu vrstvy

---

```
jQuery('#slides-content').on('drop', '.upload-area', function(e) {
    var slide = jQuery(this).parent();
    e.preventDefault();
    jQuery(this).toggleClass('hover', false);
    var dataTransfer = e.originalEvent.dataTransfer;
    if (dataTransfer.files[0].type == "image/png" || dataTransfer.files[0].type == "image/jpeg") {
        var fileReader = new FileReader();
        fileReader.readAsDataURL(dataTransfer.files[0]);
        fileReader.onload = function(ev) {
            jQuery('input[name="img"]', slide).val(ev.target.result);
            AdminLayerSlider.generatePreview(jQuery('.slide-box.active').index());
        };
    }
    else {
        alert('Vstupni soubor neni ve spravnem formtu');
    }
});
```

---

#### Výpis 4: Nahrání obrázků přetáhnutím (drop)

Vrstvy mají implementované základní funkce jako je přidávání, mazání a přesouvání vrstev. Přesouvání vrstev je realizováno pomocí funkce `sortable` z balíčku `jQuery UI`. Pořadí vrstvy poté ovlivňuje `z-index`, tzn. čím vyšší `z-index`, tím více je vrstva v popředí.

### Přechod vrstvy

U vrstvy nás zajímá, jaký bude její vstupní efekt a výstupní efekt. Jednotlivé parametry jsou vypsány v příloze B (konfigurační parametry). Vzhledem k velkému množství parametrů, jsem se rozhodl umožnit aplikaci přepínat mezi základním a expertním režimem. Právě v této části aplikace mají tyto režimy největší vliv. V expertním režimu se ve správě přechodů zobrazují veškeré parametry a je možné všechny změnit dle libosti.

V základním režimu jsou tyto parametry skryté a jsou nahrazeny dvěma výběrovými poli<sup>6</sup>, kdy jeden reprezentuje vstupní efekt a druhý výstupní efekt. Lze vidět na obrázku ?? Možnosti v těchto výběrových polích se načítají z externího souboru „`transitions.js`“, kde jsou jednotlivé možnosti nadefinované. Toto umožňuje mít předefinované efekty v externím souboru bez vazby na hlavní kód aplikace, je to způsob jak ulehčit nastavení často opakovaných efektů nebo připravit šablony pro nezkušené uživatele.

---

<sup>6</sup>výběrové pole = HTML tag `SELECT`

---

```

jQuery('#slides-content').on('click', '.style-templates', function() {
    var layer = jQuery(this).closest('.sublayer');
    var valueSelected = this.value;
    if (valueSelected !== "none") {
        jQuery.each(AdminLayerSlider.styles[valueSelected].style, function(key, value) {
            jQuery('input[name="'+key+"']", layer).val(value);
        });
        AdminLayerSlider.generatePreview( jQuery('.slide-box.active').index() );
    }
});

```

---

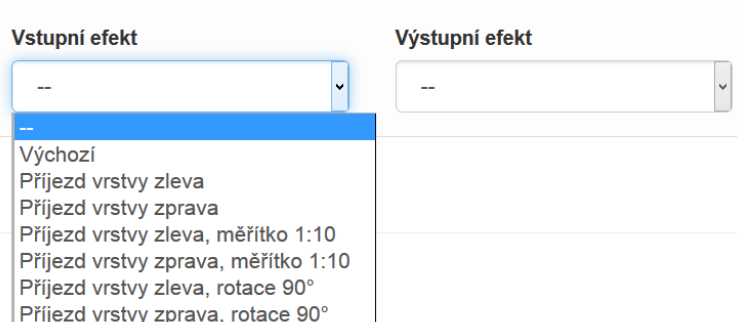
Výpis 5: Funkce pro importování stylů z šablony

## Časování vrstvy

Každý snímek má nastavenou dobu zobrazení a s tím i úzce souvisí časování vrstev. U každé vrstvy je možné nastavit dobu trvání jednotlivých fází vrstvy. K dispozici jsou čtyři vstupní pole, do kterých se zadává čas v ms.

U každé vrstvy se zobrazuje časová osa, která je reprezentována tabulkou HTML. Tato tabulka má 5 sloupců, kdy každý sloupec reprezentuje jednu z 5 fází: vrstva není zobrazena, probíhá vstupní efekt vrstvy, vrstva je zobrazena, probíhá výstupní efekt vrstvy a vrstva je skrytá (po výstupním efektu). Každý z těchto sloupců je pomocí CCS stylů obarven odlišnou barvou a při najetí myši (hover) se zobrazí titulek s popisem fáze vrstvy. Tento titulek je zobrazen pomocí funkce „tooltip“<sup>7</sup> z frameworku Bootstrap.

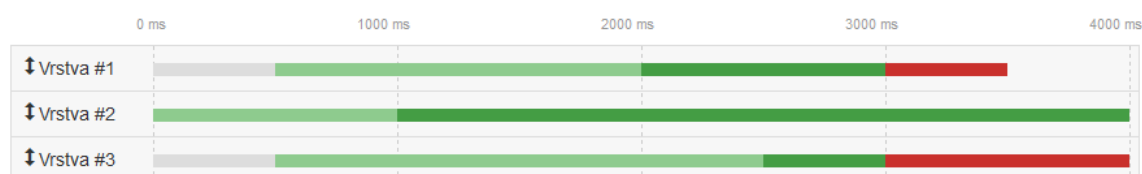
<sup>7</sup>tooltip = vyskakovací okénko s nápovědou, obvykle popisující označený element



Obrázek 9: Výběr vstupního a výstupního efektu vrstvy

Při změně jakéhokoliv ze vstupních parametrů se změní zároveň i zobrazení časové osy. Pro lepší představu, jak se budou vrstvy zobrazovat v závislosti na čase v rámci jednoho snímku, je implementováno zobrazení časových os všech vrstev najednou. Tuto funkci lze vidět na obrázku 10.

V případě, že se spustí náhled slideru a zároveň budou zobrazené časové osy, tak se bude na časové ose zobrazovat aktuální pozice pomocí jezdce, který využívá API pluginu LayerSlider, který jezdce odstartuje vždy, když je na řadě zvolený snímek.



Obrázek 10: Časové osy jednotlivých vrstev v rámci jednoho snímku

## Styl vrstvy

Podobně jako je možné u vrstvy nastavit její přechod, tak je možné nastavit i její styl. Styl vrstvy chápeme jako CSS styly. Styl vrstvy má smysl především u textových vrstev, kde můžeme nastavit základní parametry jako je barva písma, barva pozadí, zaoblené rohy, velikost písma a další.

Parametry se zadávají opět do vstupních polí, jen u barev je rozdíl v tom, že se využívá plugin jQuery MiniColors, který umožňuje jednoduchý výběr barvy. Zde je opět rozdíl mezi základním a expertním režimem. V expertním režimu jsou zobrazena všechna vstupní pole, kdežto u základního režimu se zobrazuje pole pro výběr barvy písma, pole pro výběr barvy pozadí a výběrové pole s předpřipravenými šablonami, které se načítají s externího souboru „style.js“ stejným způsobem jako šablony přechodu vrstvy. Při změně některého ze stylu, se znova vygeneruje náhled snímku a tím je zajištěno, že vidíme vždy aktuální vzhled vrstvy.

### 4.3.2 Export a import

Export je klíčová část aplikace, která umožňuje získat výstup z aplikace pro použití na webových stránkách. Na rozdíl od generování náhledu se generování exportních dat vykoná až po kliknutí na tlačítko export a to hlavně z důvodu, aby se omezilo vytěžování



procesoru při každé změně ve slideru. Pro export i import jsem využil modální dialogové okno z balíku Bootstrap. V tomto okně se zobrazí export HTML, JS a dat ve formátu JSON pro další editaci.

## JS kód

```
<script type="text/javascript">
  $(document).ready(function(){
    $('#layerslider').layerSlider({
      width : 800,
      height : 300,
      responsive : true,
      autoStart : true,
      pauseOnHover : true,
      skin : 'borderlessdark',
      skinsPath : 'skins/',
      navPrevNext : true,
      navStartStop : true
    });
  });
</script>
```

## JSON kód

```
{"sliders":[{"slidedelay":"4000","background":"data:image/jpeg;base64,/9j/4QbRRXhp
```

Obrázek 11: Ukázka exportu

V této části aplikace jsem implementoval funkci „Uložit jako zip“, která umožní všechny data uložit do souboru .zip bez nutnosti použití serverové části aplikace. Pro tuto funkci jsem využil javascript knihovnu JSZip umožňující číst, editovat a vytvářet .zip soubory. Nejprve se vytvoří nový objekt JSZip, který reprezentuje .zip soubor, vezme se export HTML a k němu se přidá HTML hlavička. Tento kód se jako celek poté vloží do archivu jako „index.html“. Do archivu se dále přidají všechny potřebné JS soubory pro spuštění slideru: LayerSlider plugin a jQuery knihovna. Součástí exportu je i export v JSON formátu pro znovu načtení celého slideru pro případ potřeby další editace. Objekt JSZip se převede do formátu BLOB a pomocí funkce „saveAs()“ z knihovny FileSaver.js se vytvoří .zip, který je možné stáhnout do počítače.

Import dat lze provést buď vložením řetězce ve formátu JSON nebo vložením souboru, který byl vygenerován exportem. Nejprve se spustí funkce, která odstraní veškeré data v aplikaci, aby nedošlo ke kolizi. Při importu dat se volají stejné funkce, které se využívají při ovládání uživatelského rozhraní. Při vložení JSON kódu se vytvoří nový objekt. Do tohoto objektu se uloží výsledek funkce `JSON.parse`, tzn. že se rozparsuje JSON kód a jako objekt se uloží do proměnné. V cyklu se vytvářejí jednotlivé snímky a vrstvy, do stylů vrstev a nastavení přechodu vrstev se načtou hodnoty a po skončení cyklu se zavolá funkce pro vygenerování náhledu.

#### 4.4 Problémy při implementaci

Při implementaci jsem začal rozvrhnutím základního layoutu, a naprogramováním základních funkcí jako je přidávání a odebírání snímků a jednotlivých vrstev. Poté jsem vymyslel a implementoval správu obsahu v jednotlivých vrstvách, abych mohl začít pracovat na náhledu slideru. V této fázi jsem řešil správu pozic vrstev, zaznamenávání polohy do vstupních polí a pracoval jsem na stylech vrstvy. Správnému generování náhledu jsem věnoval dlouhou dobu. Bylo nutné zjistit, za jakých událostí se musí generovat náhled slideru a s tím souviselo i náročné testování.

Vzhledem k tomu, že jsem používal na některé vstupní pole (inputy) jQuery plugin `minicolors`, vznikal problém v případě, že se vrstva nebo celý snímek přesunul na jinou pozici. Po změně pozice input přestal fungovat, tzn. že se nezobrazilo výběrové okno s barvou. Naprogramoval jsem funkci, která je volána při změně pozice vrstvy nebo snímku a prochází všechny vstupní pole označené třídou „color“ a znovu inicializuje plugin `minicolors` na tyto elementy.

Poměrně náročná část byla implementace a zajištění správné funkce časové osy, protože během testování jsem zjistil, že když jsem pro realizaci využil tabulku, tak se v prohlížečích Mozilla Firefox a Google Chrome chovala odlišně a nezobrazovala správné hodnoty. Bylo nutné napsat cyklus, který vzal dobu zobrazení snímku a od tohoto čísla odečítal jednotlivé fáze vrstvy a převáděl čas v ms na procenta (včetně nulových hodnot), které udávaly šířku jednotlivých sloupců v tabulce.

## 5 Použití administračního rozhraní v praxi

V této kapitole probírám možnosti využití aplikace v oblasti tvorby webových stránek. Výstup z aplikace je možné použít na libovolné webové stránce, která bude mít integrovanou komponentu LayerSlider včetně jQuery knihovny. Aplikace je vhodná pro uživatele, kteří tvoří a spravují statické stránky (tzn. bez dynamicky generovaného obsahu serverem). Statické stránky jsou spíše na ústupu, protože správa HTML obsahu je pro uživatele složitá a to jak po stránce znalostní, tak i časové.

Webové stránky jsou častěji generovány dynamicky internetovou aplikací, která je napsána v programovacím jazyce (obvykle PHP, ASP.NET, Python a další). Tyto webové aplikace mají obvykle implementovanou část pro jednoduchou správu obsahu. V této oblasti má význam zauvažovat nad použitím tohoto administračního prostředí, implementovat jej do aplikace a rozšířit tak funkce dané aplikace. Toto by mohlo být zajímavé pro firmy, které vyvíjejí vlastní systém pro správu obsahu a chtěly by snadno využít potenciál interaktivních prezentací.

Smyslem práce bylo navrhnout vhodné a univerzální řešení pro praktické využití včetně možnosti integrace administračního rozhraní do systémů třetích stran. Jak jsem již zmínil, vhodné je nasazení do redakčních systémů pro správu obsahu (CMS), internetových obchodů a dalších aplikací.

Oblast open-source systémů (aplikace jako WordPress, Joomla, PrestaShop, Drupal) je jediná oblast webových aplikací, kde výstup této bakalářské práce nebude mít uplatnění. Tyto open-source systémy mají velkou komunitu uživatelů a vývojářů, takže pro tyto systémy již existují podobné aplikace pro generování slideru. Tyto aplikace jsou ale vždy vytvořené přesně na míru pro daný systém a nelze je využít pro integraci do jiného systému.

Aplikace je implementována jako samostatně fungující webová stránka. Aplikace není závislá na programovacím jazyce, ve kterém je vyvíjena výsledná aplikace. V případě, že by se tato práce integrovala do systému, bylo by vhodné toto administrační rozhraní přizpůsobit cílovému systému. Jedna z hlavních funkcí administračního rozhraní je práce s obrázky, proto by bylo vhodné aplikaci napojit na souborový nebo obrázkový manažer používaný v celé aplikaci, který by do administrace vracel jen odkazy k požadovaným obrázkům. Také by bylo vhodné celý výstup z aplikace ukládat do databáze systému (obvykle MySQL nebo MSSQL).

Pro fungování výstupu z administračního rozhraní je nutné, aby byla ve webové apli-

kaci integrována jQuery knihovna a jQuery plugin LayerSlider. Poté stačí, aby systém vložil výstup z administračního rozhraní (HTML kód slideru) na požadované stránce a JavaScriptovou část, která inicializuje objekt LayerSlider. V ideálním případě by mohl být systém upraven tak, že umožní uchovat v administraci několik sliderů s možností umístění v různých částech webových stránek. Poté je možné využití sliderů pro reklamní banery, individuální slidery pro jednotlivé stránky a podobně. Aplikace je navržena tak, že pokud se vezme export nastavení slideru, který je v JSON formátu, stačí při následné editaci zavolat funkci pro import nastavení a data předat jako parametr. Při každém spuštění administrace by stačilo získat data z databáze a zavolat importní funkci, která zavede rozpracovaný slider.

Integrace do jakéhokoliv systému by měla být poměrně jednoduchá. Tato možnost může být zajímavá hlavně pro vývojářské studia zabývající se tvorbou webových stránek, internetových obchodů a dalších prezentací. Většina těchto studií má vyvinuté vlastní systémy, ať už postavené na nějakém open source systému nebo napsané kompletně dle vlastní potřeby.

## 5.1 Možnosti dalšího vývoje

V této kapitole bych chtěl shrnout nápady pro další vývoj aplikace. Mezi hlavní změny bych zařadil možnost generování náhledu podle cílové stránky. V této chvíli se generuje náhled podle zadaných rozměrů, tím se předpokládá standardní šířka webové stránky, kde bude slider umístěn. Při responsivním zobrazení se slider automaticky přizpůsobí. Některé weby používají tzv. „full-width“ slidery, tzn. slidery které nemají definovanou maximální šířku, ale zobrazují se od levého okraje do pravého okraje prohlížeče. Na tento typ slideru není aplikace úplně připravena, ačkoliv by nebyl problém takový slider v aplikaci vytvořit, lepší by bylo aplikaci rozšířit.

Další prostor k rozšíření aplikace je v oblasti efektů přechodu snímků, tzn. prolínání mezi snímky. Aktuálně jsou efekty řešeny tak, že jsou na výběr buď jen náhodně 2D efekty, náhodně 3D efekty anebo všechny efekty náhodně. Tady by bylo vhodné umožnit zobrazit jednotlivé efekty, vč. náhledu s možností individuálního výběru pro každý snímek. Změnu bych udělal také u stylů vrstev, tady bych si dokázal představit možnost psaní vlastního CSS stylu, především v expertním režimu.

## 6 Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat administrační rozhraní pro tvorbu interaktivních prezentací vhodné pro použití na moderních a responzivních internetových stránkách. V této bakalářské práci se podařilo úspěšně vyvinout samostatně fungující aplikaci, která umožňuje generovat slidery a to takovým způsobem, že to zvládne i běžný uživatel. Administrační rozhraní je vyvinuto na základě jQuery pluginu LayerSlider a výstupem aplikace je kód, který poté zpracovává tento plugin.

V rámci této práce jsem si rozšířil zároveň teoretické znalosti v jazyce JavaScript i praktické dovednosti v oblasti tvorby moderních webových aplikací za použití volně dostupných frameworků jako je Bootstrap.

Tento projekt má velký potenciál pro praktické využití a nasazení do systémů třetích stran a to hlavně v systémech pro správu obsahu a internetových obchodech. Aplikaci se podařilo implementovat dle počátečních požadavků a předpokladů a stále je zde prostor další vývoj pro využití plného potenciálu LayerSlider pluginu a možnost využití v praxi.

Roman Kotyra

## 7 Reference

- [1] *Slider Revolution Responsive jQuery Plugin* [online]. 2015 [cit. 2015-03-04].  
Dostupné z WWW:  
<<http://www.themepunch.com/portfolio/slider-revolution-jquery-plugin/>>
- [2] *LayerSlider Responsive jQuery Slider Plugin* [online]. 2015 [cit. 2015-03-02].  
Dostupné z WWW:  
<<http://kreaturamedia.com/layerslider-responsive-jquery-slider-plugin/>>
- [3] *JavaScript Technologies Market Share and Web Usage Statistics* [online]. 2015 [cit. 2015-03-08]  
Dostupné z WWW:  
<<https://www.similartech.com/categories/javascript>>
- [4] *Bootstrap 3* [online]. 2015 [cit. 2015-03-12]  
Dostupné z WWW:  
<<http://www.w3schools.com/bootstrap/>>
- [5] *File API: Writer* [online]. 2015 [cit. 2015-04-11]  
Dostupné z WWW:  
<<http://www.w3.org/TR/file-writer-api/the-filesaver-interface>>
- [6] *Saving generated files on the client-side - Eli Grey* [online]. 2015 [cit. 2015-04-11]  
Dostupné z WWW:  
<<http://updates.html5rocks.com/2011/08/Saving-generated-files-on-the-client-side>>
- [7] *Envato Market Licences* [online]. 2015 [cit. 2015-05-16]  
Dostupné z WWW:  
<<http://themeforest.net/licenses>>

## A Příloha ve formě CD média

Součástí práce je elektronická příloha na CD. Informace o umístění dat uložených na disku CD jsou uvedeny v souboru README.TXT, který je nahrán v hlavním adresáři CD.

### Příloha obsahuje:

- kompletní zdrojové kódy aplikace,
- text bakalářské práce (tento dokument) v elektronické podobě,
- plugin LayerSlider,
- spustitelné ukázky sliderů,
- videozáznam s ukázkou funkce aplikace.

## B Konfigurační parametry

### Parametry vstupního efektu

#### offsetxin

- **Hodnoty:** „left“, „right“ nebo číselná hodnota
- **Výchozí:** 80
- **Popis:** Určuje počáteční pozici vrstvy v horizontální ose na začátku efektu. Hodnota „left“ určuje, že vrstva „přijede“ zleva, počáteční pozice je mimo snímek, tedy není viditelná. Číselná hodnota je relativní vychýlení počáteční pozice.

#### offsetyin

- **Hodnoty:** „top“, „bottom“ nebo číselní hodnota
- **Výchozí:** 0
- **Popis:** Určuje počáteční pozici vrstvy ve vertikální ose na začátku efektu. Hodnota „top“ určuje, že vrstva „přijede“ z vrchu, počáteční pozice je mimo snímek, tedy není viditelná. Číselná hodnota je relativní vychýlení počáteční pozice. Jsou povolené kladné i záporné hodnoty.

#### delayin

- **Hodnoty:** číselní hodnota
- **Výchozí:** 0
- **Popis:** Hodnota, která určuje, kdy se zahájí efekt. Pomocí této hodnoty lze časovat vrstvy. Čas se počítá od zobrazení snímku a udává se v milisekundách. Hodnota 500 ms nám říká, že efekt vrstvy se zahájí po 500 ms od zobrazení daného snímku.

#### durationin

- **Hodnoty:** číselní hodnota
- **Výchozí:** 1000
- **Popis:** Doba trvání efektu v milisekundách.



### easingin

- **Hodnoty:** Linear, swing, easeInQuad, easeOutQuad, easeInOutQuad, easeInCubic, easeOutCubic, easeInOutCubic, easeInQuart, easeOutQuart, easeInOutQuart, easeInQuint, easeOutQuint, easeInOutQuint, easeInSine, easeOutSine, easeInOutSine, easeInExpo, easeOutExpo, easeInOutExpo, easeInCirc, easeOutCirc, easeInOutCirc, easeInElastic, easeOutElastic, easeInOutElastic, easeInBack, easeOutBack, easeInOutBack, easeInBounce, easeOutBounce, easeInOutBounce
- **Výchozí:** easeInOutQuint
- **Popis:** Určuje průběh efektu v závislosti na čase.

### fadein

- **Hodnoty:** „true“ nebo „false“
- **Výchozí:** „true“
- **Popis:** Zapne nebo vypne „roztmívací“ (fade) efekt, tzn. na začátku efektu je vrstva neviditelná, a na konci efektu je 100% viditelná.

### rotatein

- **Hodnoty:** úhel (číselná hodnota)
- **Výchozí:** 0
- **Popis:** Rotace vrstvy během efektu proti směru hodinových ručiček. Hodnota určuje úhel. Záporné hodnoty jsou povolené (rotace ve směru hodinových ručiček).

### rotatexin

- **Hodnoty:** úhel (číselná hodnota)
- **Výchozí:** 0
- **Popis:** Rotace vrstvy podle osy X (horizontálně) ze zadaného úhlu na 0. Negativní hodnota je povolena pro rotaci v opačném směru.

### rotateyin

- **Hodnoty:** úhel (číselná hodnota)
- **Výchozí:** 0
- **Popis:** Rotace vrstvy podle osy Y (vertikálně) ze zadaného úhlu na 0. Negativní hodnota je povolena pro rotaci v opačném směru.

### scalexin

- **Hodnoty:** číselná hodnota
- **Výchozí:** 1
- **Popis:** Hodnota určuje počáteční šířku vrstvy. (při zahájení efektu) Počáteční šířka je násobek zadané hodnoty a konečné šířky vrstvy. Lze považovat za pomyslné měřítko / poměr z původní šířky.

### scaleyin

- **Hodnoty:** číselná hodnota
- **Výchozí:** 1
- **Popis:** Hodnota určuje počáteční výšku vrstvy. (při zahájení efektu) Počáteční výška je násobek zadané hodnoty a konečné výšky vrstvy.

### skewxin

- **Hodnoty:** úhel (číselná hodnota)
- **Výchozí:** 0
- **Popis:** Hodnota určuje pokřivení vrstvy podle osy X

### skewyin

- **Hodnoty:** úhel (číselná hodnota)
- **Výchozí:** 0
- **Popis:** Hodnota určuje pokřivení vrstvy podle osy Y

## Parametry výstupního efektu

Výstupní efekt a parametry se projeví pouze tehdy, pokud máme nastaven čas „showuntil“, což je hodnota, která určuje za jak dlouho po dokončení vstupního efektu, se zahájí výstupní efekt.

### offsetxout

- Hodnoty: „left“, „right“ nebo číselná hodnota
- Výchozí: 80
- Popis: Určuje konečnou pozici vrstvy v horizontální ose na konci efektu. Hodnota „left“ určuje, že se vrstva „schová“ za levý okraj snímku, konečná pozice je mimo snímek, tedy není viditelná. Číselná hodnota je relativní vychýlení konečné pozice.

### offsetyout

- Hodnoty: „top“, „bottom“ nebo číselní hodnota
- Výchozí: 0
- Popis: Určuje konečnou pozici vrstvy ve vertikální ose na konci efektu. Hodnota „top“ určuje, že se vrstva „schová“ za horní okraj snímku, konečná pozice je mimo snímek, tedy není viditelná. Číselná hodnota je relativní vychýlení konečné pozice.

### showuntil

- Hodnoty: číselná hodnota
- Výchozí: 0
- Popis: Určuje, po jaké době se spustí výstupní efekt, pokud tato hodnota není nastavena nebo se rovná 0, výstupní efekt se nekoná. Daná vrstva se bude zobrazovat až do konce trvání snímku a ztratí se až při přechodu na další snímek.

### durationout

- Hodnoty: číselná hodnota

- Výchozí: 1000
- Popis: Doba trvání efektu v milisekundách.

### **easingout**

- Hodnoty: Linear, swing, easeInQuad, easeOutQuad, easeInOutQuad, easeInCubic, easeOutCubic, easeInOutCubic, easeInQuart, easeOutQuart, easeInOutQuart, easeInQuint, easeOutQuint, easeInOutQuint, easeInSine, easeOutSine, easeInOutSine, easeInExpo, easeOutExpo, easeInOutExpo, easeInCirc, easeOutCirc, easeInOutCirc, easeInElastic, easeOutElastic, easeInOutElastic, easeInBack, easeOutBack, easeInOutBack, easeInBounce, easeOutBounce, easeInOutBounce
- Výchozí: easeInOutQuint
- Popis: Určuje průběh efektu v závislosti na čase. Viz. easingin

### **fadeout**

- Hodnoty: „true“ nebo „false“
- Výchozí: „true“
- Popis: Zapne nebo vypne „zatmívací“ (fade) efekt, tzn. na začátku efektu je vrstva 100% viditelná, a na konci efektu je neviditelná.

### **rotateout**

- Hodnoty: úhel (číselná hodnota)
- Výchozí: 0
- Popis: Rotace vrstvy během efektu ve směru hodinových ručiček. Hodnota určuje konečný úhel vůči původní pozici. Záporné hodnoty jsou povolené (rotace ve směru proti pohybu hodinových ručiček).

**rotatexout**

- Hodnoty: úhel (číselná hodnota)
- Výchozí: 0
- Popis: Rotace vrstvy podle osy X (horizontálně) z 0 na požadovaný úhel. Negativní hodnota je povolena pro rotaci v opačném směru.

**rotateyout**

- Hodnoty: úhel (číselná hodnota)
- Výchozí: 0
- Popis: Rotace vrstvy podle osy Y (vertikálně) z 0 na požadovaný úhel. Negativní hodnota je povolena pro rotaci v opačném směru.

**scalexout**

- Hodnoty: číselná hodnota
- Výchozí: 1
- Popis: Hodnota určuje konečnou šířku vrstvy. (při ukončení efektu) Konečná šířka je násobek zadané hodnoty a počáteční šířky vrstvy. Lze považovat za pomyslné měřítko / poměr z původní šířky.

**scaleyout**

- Hodnoty: číselná hodnota
- Výchozí: 1
- Popis: Hodnota určuje konečnou výšku vrstvy. (při ukončení efektu) Konečná výška je násobek zadané hodnoty a počáteční šířky vrstvy. Lze považovat za pomyslné měřítko / poměr z původní výšky.

**skewxout**

- Hodnoty: úhel (číselná hodnota)
- Výchozí: 0
- Popis: Hodnota určuje zkosení vrstvy podle osy X

**skewyout**

- Hodnoty: úhel (číselná hodnota)
- Výchozí: 0
- Popis: Hodnota určuje zkosení vrstvy podle osy Y