

Uživatelsky přívětivé automatické generování HTML stránek pro umístování sdělení přes firemní televizní okruh.

The User-Friendly Automatic Generation of HTML Pages for
News Placement for the TV Loop

František Partyš

Bakalářská práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

*** nescannované zadání str. 1 ***

*** nescannované zadání str. 2 ***

ABSTRAKT

Cílem této bakalářské práce bylo vytvoření jednoduchého a uživatelsky přívětivého HTML editoru pro vytváření webových stránek, určený pro prezentaci v televizním okruhu v podnikových budovách. Teoretická část rozebírá použité technologie s ohledem na stanovené podmínky realizace. V praktické části je popsán samotný proces tvorby aplikace a je zde uveden návod, jak tuto aplikaci používat.

Klíčová slova: webové stránky, reklama, webové technologie, HTML, Javascript, WebSQL

ABSTRACT

The aim of this thesis was to create a simple and user-friendly HTML editor for creating web pages intended for presentation on the television circuit in corporate buildings. The theoretical part discusses the used technology with respect to the conditions of implementation. The practical part describes the process of creating applications and there is a guide, listed in this part, on how to use this application.

Keywords: web pages, advertisement, HTML, Javascript, WebSQL, www technologies

Chtěl bych poděkovat v první řadě vedoucí mé práce Ing. Zuzaně Oplatkové Komínkové Ph.D. za pomoc při řešení problémů, spojených s vypracováním této práce. Dále bych chtěl poděkovat své rodině a všem blízkým, kteří jsou pro mě celý život oporou.

Motto:

„Don't only practice your art, but force your way into its secrets, for it and knowledge can raise men to the divine“

~Ludwig van Beethoven

Prohlašuji, že

- beru na vědomí, že odevzdáním 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 bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk 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 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 –bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování 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 bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem 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 bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....

diplomanta

.....

podpis

OBSAH

ÚVOD.....	10
TEORETICKÁ ČÁST.....	10
ZÁKLADNÍ POŽADAVKY ŘEŠENÍ.....	11
POUŽITÉ TECHNOLOGIE.....	12
1.1HTML.....	12
1.1.1Historie jazyka HTML.....	12
1.1.2HTML verze 5.....	13
1.2CSS.....	13
1.3DATABÁZE V PROHLÍZEČI.....	14
1.3.1WebSQL databáze.....	14
1.3.2IndexedDB Database API.....	15
1.4JAVASCRIPT.....	16
1.4.1Knihovna jQuery	16
1.5POUŽITÉ PLUGINY KNIHOVNY JQUERY.....	16
1.5.1jQuery Validation Plugin.....	17
1.5.2jQuery Cycle plugin.....	18
PROGRAMOVÉ VYBAVENÍ.....	20
1.6EDITOR PSPAD.....	20
1.7ADOBE PHOTOSHOP.....	20
1.8GOOGLE CHROME.....	21
PRAKTICKÁ ČÁST.....	22
REALIZACE APLIKACE.....	23
1.9ANALÝZA APLIKACE.....	23
1.9.1Funkcionalita aplikace.....	23
1.9.2Editor.....	23
1.9.3Viewer.....	24
1.10POPIS ČÁSTÍ APLIKACE.....	25
1.10.1Databáze.....	25
1.10.2Tabulka „properties“.....	26
1.10.2.1Sloupec projectid.....	26
1.10.2.2Sloupec projectname.....	26
1.10.2.3Sloupec width.....	26
1.10.2.4Sloupec height.....	26
1.10.2.5Sloupce isNorth, isMain, isEast, isWest.....	26
1.10.2.6Sloupce northHeight, westWidth, eastWidth.....	26
1.10.2.7Sloupce northTime, mainTime, westTime, estTime.....	27
1.10.2.8Sloupce northEffect, mainEffect, westEffect, eastEffect.....	27
1.10.3Tabulky northImages, mainImages, westImages, eastImages. .28	
1.10.3.1Sloupec imgid.....	28
1.10.3.2Sloupec projectname.....	28
1.10.3.3Sloupec img.....	28
1.10.4Layout	29
1.10.4.1Horní okno.....	30

1.10.4.2Levé okno.....	30
1.10.4.3Pravé okno.....	30
1.10.4.4Hlavní okno.....	30
1.11VYTVOŘENÍ NOVÉHO PROJEKTU.....	30
1.11.1Název projektu.....	30
1.11.2Rozlišení projektu.....	31
1.11.3Layout.....	31
1.11.4Nastavení rozměrů oken.....	31
1.11.5Nastavení času změny.....	31
1.11.6Nastavení efektu změny.....	32
1.11.7Přiřazení obrázků.....	32
1.12ZOBRAZENÍ PROJEKTU.....	34
1.12.1Zkopírování databáze.....	34
ZÁVĚR.....	35
ZÁVĚR V ANGLIČTINĚ.....	36
SEZNAM POUŽITÉ LITERATURY.....	37
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	38
SEZNAM OBRÁZKŮ.....	39
SEZNAM TABULEK.....	40
SEZNAM PŘÍLOH.....	41

ÚVOD

Cílem této bakalářské práce je zjednodušit vytváření statických webových stránek pro reklamní televizní systém v podnicích bez použití serverové logiky a to kvůli pracnosti nasazení takové aplikace a finančním zatížení. V takovém případě je potřeba koupit server a osadit jej příslušným programovým vybavením, případně platit podporu těchto zařízení což to je velmi finančně nákladné. Vypracováním této práce jsem si ověřil, že do jisté míry lze napsat aplikaci, která bude pracovat pouze na straně klienta, ale spoustu funkcionalit méně či více důležitých, nelze tímto způsobem realizovat. Lze dosáhnout funkčního řešení a vyhovět těmto podmínkám, ale některé výhody při použití serverové logiky nebudou a nemůžou být součástí takové aplikace. Alespoň ne v současné době.

Uživatel bude schopen pomocí této aplikace vytvořit jednoduchý layout, který naplní obsahem a zobrazí jej na počítači obsluhující televizor. Některé kroky ovšem musí realizovat mimo tuto aplikaci, jako je například přenos dat ze zdrojového počítače na počítač obsluhující televizor a to z důvodů, že tento jazyk tyto funkce nepodporuje a to především z bezpečnostního hlediska.

TEORETICKÁ ČÁST

ZÁKLADNÍ POŽADAVKY ŘEŠENÍ

Při návrhu aplikace jsem vycházel z požadavků, které editor musí splňovat. Tato skutečnost zásadním způsobem omezila množinu technologií, které jsem mohl při návrhu aplikace použít. Výčet omezení je následující:

- **Aplikace musí běžet na jakékoliv platformě**
- **K jejímu používání se nesmí instalovat jiný řídicí software, např. databáze**

POUŽITÉ TECHNOLOGIE

1.1 HTML

Jazyk HTML neboli Hypertext Markup Language je značkovací jazyk, vyvinutý pro jednoduchou tvorbu dokumentů a s tím spojenou jejich efektivní distribucí. V současné době je nejvíce používán pro vytváření a sdílení dokumentů v systému WWW přes protokol HTTP.

1.1.1 Historie jazyka HTML

První myšlenka na vytvoření nového značkovacího jazyka pro lepší sdílení dokumentů vznikla v roce 1989 ve Švýcarsku. Otcem této myšlenky byl vědec Sir Timothy John Berners Lee. Ten měl za úkol vyřešit problém s vytvářením a distribucí dokumentů ve velké korporaci. První verze jazyka uměla členit text do logických částí, umožňovala několik druhů zvýrazňování textu a přidávat odkazy s obrázky.

První oficiální definice byla vydána o dva roky později v roce 1991. Jazyk zaznamenal obrovský úspěch a rychle se šířil. Bylo tedy nutné vytvořit konsorcium, které bude dohlížet na jeho vývoj a jednotnost. V roce 1994 Sir Timothy John Berners Lee založil World Wide Web Consortium. Od té doby se jazyk vyvíjel a vzniklo několik standardů, s kterými přibývaly i další možnosti při tvorbě dokumentů. Výčet verzí jazyka HTML.

- **Verze 0.9** – vydána v roce 1991. Nepodporuje grafický režim
- **Verze 2.0** – k předchozí verzi přidává podporu grafiky a interaktivních formulářů.
- **Verze 3.0** – nebyla nikdy vydána pro svou složitost.
- **Verze 3.2** – přidává k jazyku tabulky, zarovnávání textu a stylové elementy pro ovlivňování vzhledu.
- **Verze 4.0** – byly přidány nové prvky pro tvorbu tabulek, formulářů a byl přidán standard pro podporu rámců(frames).
- **Verze 4.01** – opravuje chyby předchozí verze.
- **Verze 5.0** - poslední verze, která je stále ve vývoji. Zásadně ovlivní možnost použití jazyka. Vydání standardu se odhaduje na rok 2014.

1.1.2 HTML verze 5

Poslední verze jazyka HTML přináší velké změny. Nabídne nové technologie integrované přímo do tohoto jazyka. Nově bude možné použít databáze. Přibude možnost psát offline aplikace, přehrávání videa bez instalování pluginů apod. Zkrátí zápis tagů, nabídne vykreslování vektorové grafiky a vylepší možnosti formulářů např. kontrolu vstupních údajů. Jedno z velkých vylepšení bude možnost pracovat se soubory. Toto rozšíření posune možnosti aplikací psané v jazyce HTML na vyšší úroveň. Bude možné programovat mnohem složitější aplikace bez použití dalších programovacích jazyků jako je například Javascript.

Nová verze HTML 5 přináší i nové objekty, jejichž značky jsou následující:

- `<article>` - článek
- `<audio>` - zvukový obsah
- `<canvas>` - vektorová grafika
- `<footer>` - definuje patičku stránky
- `<header>` - definuje hlavičku stránky
- `<nav>` - definuje odkazy navigaci
- `<section>` - definuje sekci
- `<video>` - blok pro zobrazení videosouborů

Dalším velkým přínosem je začlenění drag-and-drop technologie do specifikace HTML 5. Zjednodušenou implementací metody práce s objekty typu "táhni a pusť" značně zjednoduší a rozšíří jeho používání.

1.2 CSS

Kaskádové styly CSS (Cascading Style Sheets) je jazyk pro popis způsobu zobrazení stránek napsaných v jazyce HTML. Byl navržen W3C organizací. Jeho hlavním cílem bylo oddělit obsah a strukturu obsahu od definice jeho vzhledu. Syntaxe jazyka se skládá z několika málo pravidel. Každé pravidlo obsahuje selektor, který reprezentuje daný HTML element a za ním následuje definice jeho vzhledu a vlastností.

Kaskádové styly je možné do webové stránky začlenit třemi různými způsoby:

Inline zápis - styly jsou vkládány přímo jednotlivým elementům přes atribut style.

Příklad:

```
<a href="index.html" style="color: red; font-size: 10pt;">
```

Vnořený zápis – vložení CSS stylu jako jeden ucelený blok textu do hlavičky dokumentu tj. mezi tagy `<head></head>`.

Příklad:

```
<style type="text/css"> a {color: red; font-size: 10pt;} </style>
```

Externí soubor – vytvořený soubor se zápisem kaskádových stylů připojíme k webové stránce pomocí značky `<link>`, který se umísťuje do hlavičky dokumentu tj. mezi tagy `<head></head>`.

Příklad:

```
<link rel="stylesheet" href="externy_subor_css.css" type="text/css"/>
```

1.3 Databáze v prohlížeči

Je to databáze obsažená přímo v prohlížeči. Není tedy nutné pro použití databáze a jejich výhod instalovat složitě databázový server. Další výhodou je zjednodušení použití databáze. Pokud potřebujeme jednoduché úložiště dat s výhodami relačních databází a projekt, ve kterém je použijeme, není příliš složitý, je to ideální varianta. V současnosti existují dvě možnosti a to WebSQL Database a IndexedDB.

1.3.1 WebSQL databáze

Jde o rozhraní k databázi SQLite. Pracuje se s ní tedy stejně. Nejedná se o klasický model klient-server aplikace, ale tato databáze je připojena k aplikaci jako knihovna a pomocí tohoto rozhraní ji můžeme používat. Každá databáze je uložena v samostatném souboru, kde se data ukládají za použití jednoduchého primárního klíče do stejně velkých bloků a používá hashovací techniky pro rychlý přístup k datům při vyhledávání podle klíče. Takovéto řešení je nezávislé na platformě, ale má i řadu nevýhod. Jednou z hlavních nevýhod je nemožnost konfigurace databáze. Další nevýhodou SQLite databáze je absence některých metod přístupu k datům jako je spojování tabulek pomocí příkazu OUTER

RIGHT JOIN. Tato vlastnost SQLite databáze není vzhledem k určení použití databáze až tak omezující. Největší omezení tohoto řešení je podpora prohlížečů. Některé prohlížeče ji nepodporují vůbec, jiné pouze z části. Hlavním zastáncem této technologie je prohlížeč Google Chrome. Pro moji práci jsem si vybral právě WebSQL databázi. Při vytvoření nové databáze musíme určit:

- Jméno databáze
- Verzi databáze
- Textový popis databáze
- Velikost Databáze

Příklad vytvoření databáze:

```
var db = openDatabase('mydb', '1.0', 'my first database', 2
* 1024 * 1024);
```

Prohlížeč	Podpora	Od verze	Současná verze
Internet Explorer	ne		10.0
Mozilla Firefox	ne		21.0
Google Chrome	ano	26.0	27.0

Tab. 1. Podpora WebSQL databáze v prohlížečích

1.3.2 IndexedDB Database API

Chtěl bych zde zmínit alternativu k mému použitému řešení a tím je IndexedDB Database API.

IndexedDB není klasickou relační databází, ale v podstatě implementací datové struktury typu strom. Toto řešení je v principu objektové úložiště, které lze procházet a snadno v něm vyhledávat a nabízí možnosti podobně relačním databázím. Další rozdíl je ten, že přístup k datům je přímý bez prostředníka ve formě SQL příkazů. Je zde ovšem stejný problém jako u předchozí možnosti. Podpora prohlížečů není dostatečná. Hlavním zastáncem této technologie je prohlížeč Mozilla Firefox.

Toto řešení jsem nezvolil právě z důvodu odlišnosti uchovávání dat oproti relačním databázím a způsobu dolování dat.

Prohlížeč	Podpora	Od verze	Současná verze
Internet Explorer	ano	10.0	10.0
Mozilla Firefox	ano	20.0	21.0
Google Chrome	ano	26.0	27.0

Tab. 2. Podpora IndexedDB databáze v prohlížečích

1.4 Javascript

Javascript je pseudo-objektově orientovaný skriptovací jazyk využívající modelu DOM (Document Object Model). Dnešní moderní webové prezentace se bez něj neobejdou. Jeho autor je Brendan Eich a byl vyvinut pro dnes již zaniklý webový prohlížeč Netscape. Od ostatních webových programovacích jazyků se liší zejména tím, že běží na straně klienta, nikoliv na straně serveru. Nejsilnější stránkou Javascriptu je jeho komunita. Na internetu je nepřeborné množství hotových řešení, které stačí implementovat do webové prezentace. Vznikly i rozsáhle knihovny jako je například jQuery.

Javascript slouží zejména k řízení událostí na webové stránce. Vytváří událost po stisku tlačítka, mění strukturu stránky při akci uživatele apod. Jsou dvě varianty jeho použití. Buď přímo v HTML kódu, nebo jeho načtením z externího souboru.

1.4.1 Knihovna jQuery

Knihovna jQuery rozšiřuje použití samotného jazyka Javascript. Je to otevřený a svobodný software, který je distribuován pod licencí MIT a GNU GPL. Vytváří most mezi jazykem HTML s jeho objektovým modelem a Javascriptem. Autorem této knihovny je John Resig. Podobně jako CSS, odděluje knihovna jQuery řízení události od HTML. Knihovna jQuery je zpravidla jeden soubor, který se připojuje k projektu. Můžeme soubor připojit lokálně nebo z webové stránky projektu.

Hlavním cílem knihovny jQuery je usnadnit procházení a manipulaci s dokumenty HTML, zpracování událostí prohlížeče, animace a samotné programování skriptů v Javascriptu a zaručuje fungování těchto skriptů ve všech moderních prohlížečích.

Usnadňuje vývoj v Javascriptu a umožňuje celé komunitě jednoduše sdílet funkcionality napsané v Javascriptu, tím, že zaručuje jejich kompatibilitu ve všech prohlížečích. Existuje nespočet knihoven a pluginů využívajících výhod této knihovny.

1.5 Použité pluginy knihovny jQuery

V mé práci jsem použil dva pluginy této knihovny, které jsou volně dostupné pod licencí GNU GPL.

1.5.1 jQuery Validation Plugin

Tento plugin slouží ke kontrole formulářů před jeho zpracováním a odesláním na straně klienta. Není tedy nutné formulář odeslat na server, kde se zpracují vstupní data zadané uživatelem, která se následně vyhodnotí a při případném výskytu chyb vrátí uživateli zprávu o tom, že zadaná data nejsou v pořádku a vyzve jej k opětovnému zadání. Tento proces je zbytečně dlouhý, složitý a v konečném důsledku nežádoucí, protože proběhne spousta zbytečných operací.

Validation plugin umožňuje kontrolu všech možných druhů vstupních dat a jejich specifických formátů, jako je formát emailu, kde vyžaduje přítomnost znaku zavináče a domény prvního řádu, správný formát zadávaného čísla. Například pole pro zadávání poštovního směrovacího čísla lze omezit pouze na číslice a minimální a zároveň maximální počet číslic musí být roven počtu šest.

Zadávání těchto podmínek se realizuje formou rozšiřujících atributů name příslušného HTML elementu. Rozšiřující atribut name může například obsahovat následující hodnoty:

- **email** – vstupní pole bude akceptovat pouze řetězec ve formát `jmeno@domena.cz`
- **number** – vstupní pole bude akceptovat pouze číslice
- **date** – vstupní pole bude pouze akceptovat datum v určeném formátu
- **url** – vstupní pole bude akceptovat pouze řetězec ve tvaru `http://domena.cz`
- **minlength** – určuje minimální délku řetězce
- **maxlength** – určuje maximální délku řetězce
- **rangelength** – určuje rozsah délky řetězce

Dále pomocí atributu `required` můžeme určit, zda vyplnění příslušného pole je povinné a bez jeho vyplnění nebude možné formulář odeslat.

Pokud uživatel nesplní nějakou podmínku, plugin mu nabídne nápovědu, která mu sdělí, které pole je povinné a které má špatně vyplněno.

Použití pluginu díky knihovně jQuery je velmi snadný. Pomocí kódu a identifikátoru projdeme HTML kód, až narazíme na formulář, který má daný identifikátor a tomu přiřadí událost, která spustí po stisku tlačítka kód paginu.

Příklad použití pluginu:

```
<html>
<form id = "frm">
<input type = "text" name = "email">
<button>Odešli</button>

</html>

<script type="text/javascript">
$(document).ready(function(){
$("#frm").validate();
});
</script>
```

Plugin je dostupný na adrese [www. http://jqueryvalidation.org/](http://jqueryvalidation.org/) a jeho distribuce je dostupná pod stejnou licencí jako knihovna jQuery to je GNU GPL.

The screenshot shows a web form with the following fields and validation messages:

- Jméno**: Required field, marked with an asterisk (*). Validation message: "Tento údaj je povinný."
- Email**: Required field, marked with an asterisk (*). Validation message: "Prosím, zadejte platný e-mail."
- URL**: No validation message shown.
- Komentář**: Required field, marked with an asterisk (*). Validation message: "Tento údaj je povinný."

A "Odeslat" button is located at the bottom of the form.

Obr. 1. Použití jQuery Validation pluginu

1.5.2 jQuery Cycle plugin

Cycle plugin knihovny jquery slouží k automatické změně obrázků na webové stránce. Do objektu v HTML kódu, jemuž přiřadíme unikátní identifikátor, vložíme pomocí nepárové značky `` obrázky a plugin zařídí jejich obměnu podle parametrů, které si zvolíme. Základní nastavení nám umožňuje určit časový úsek mezi jednotlivými obrázky, grafický efekt změny, směr změny, rychlost, se kterou se jednotlivé obrázky mění.

Některé vstupní parametry:

- **fx** – definuje grafický efekt změny

- **delay** – definuje časový úsek mezi obrázky
- **speed** – určuje rychlost změny
- **pause** - po najetí myši zůstane zobrazený aktuální obrázek

Můžeme použít následující grafické efekty pro změnu obrázku:

- **shuffle**
- **zoom**
- **turn down**
- **curtainX**
- **scroll right**

Protože se jako v předchozím případě také jedná o plugin knihovny jQuery, jeho použití je stejné.

Příklad použití Cycle pluginu:

```
<div id = "cycle">  
<img src = "obr1.jpg">  
<img src = "obr2.jpg">  
<img src = "obr2.jpg">  
</div>
```

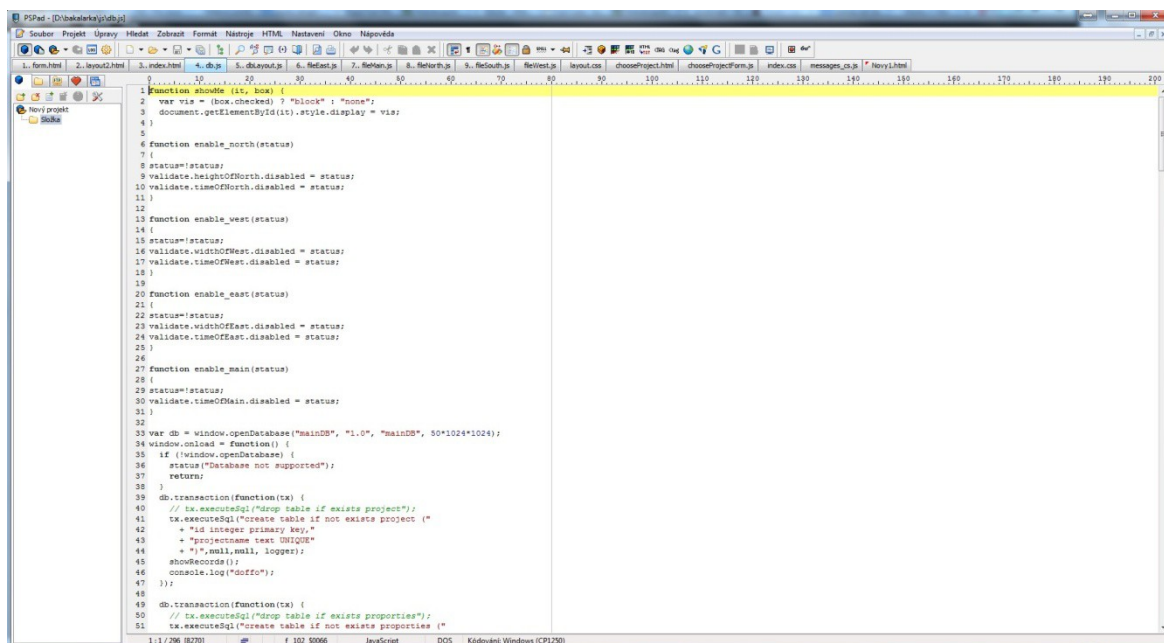
```
$('#cycle').cycle({  
  fx: 'shuffle',  
  delay: -4000  
});
```

PROGRAMOVÉ VYBAVENÍ

K vývoji každé aplikace je potřeba programové vybavení, to je software, jenž nám do značné míry usnadní vývoj ať už zvýrazňováním syntaxe, upozorněním na špatnou syntaxi nebo jednoduchým vkládáním bloků kódu. Při práci s obrázky je programové vybavení tomu určené nutností.

1.6 Editor PSPad

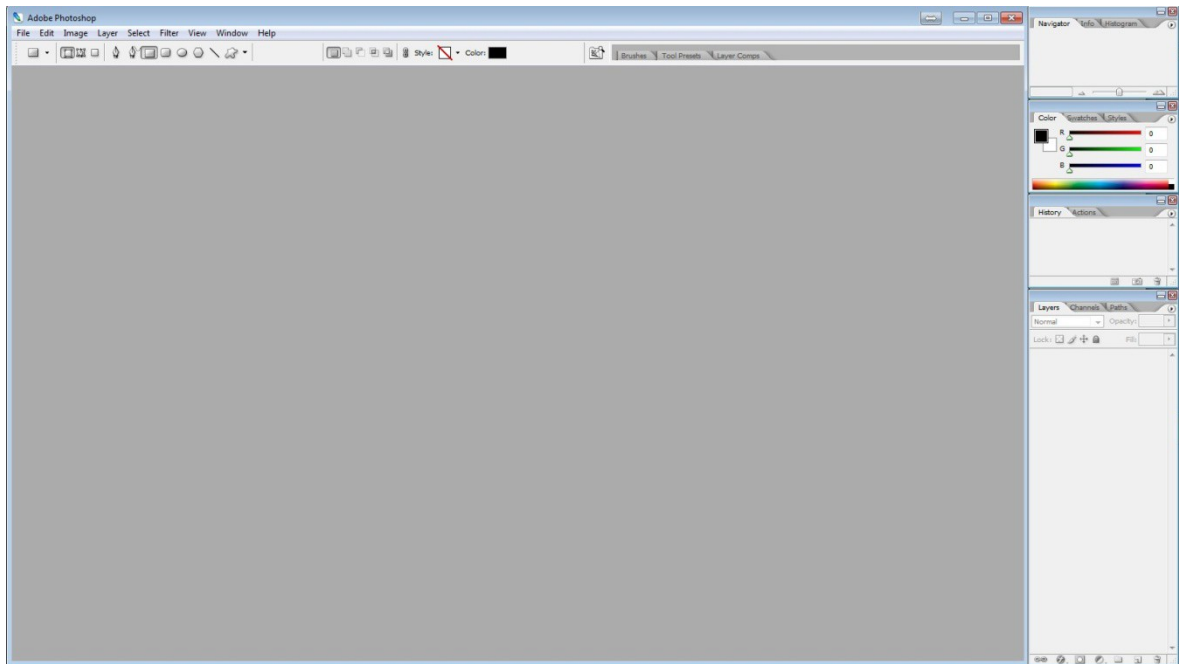
PSPad je jedním s nejrozšířenějších textových editorů. Má nespočet funkcí, umí zvýrazňovat syntaxi pro mnoho programovacích jazyků a přitom je stále jednoduchý, jeho ovládání je intuitivní a rychlost odezvy je výborná.



Obr. 2. Hlavní okno programu

1.7 Adobe Photoshop

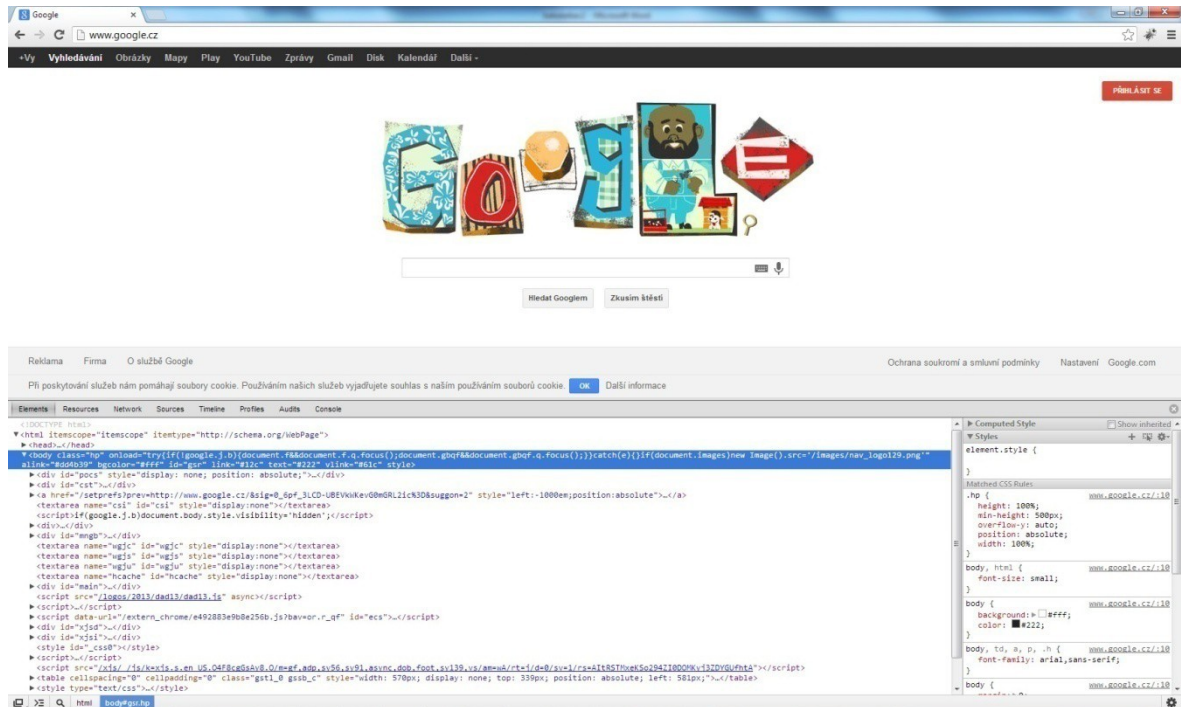
Velmi oblíbený a komplexní nástroj pro práci s fotografiemi a webovou grafikou. Byl použit pouze pro vytvoření demonstračních obrázků k vyzkoušení funkčnosti aplikace.



Obr. 3. Hlavní okno programu

1.8 Google Chrome

Tento prohlížeč webových stránek jsem zvolil kvůli jeho podpoře WebSQL databáze. Vyznačuje se velkou rychlostí a to zejména při práci s Javascriptem a jeho knihovny. Přinesl také nový pohled na grafické prostředí, které je velmi jednoduché a intuitivní. Obsahuje také velmi účinný nástroj pro vývojáře, který je podle mého názoru mnohem dál, než u konkurence.



Obr. 4. Hlavní okno prohlížeče se zobrazeným nástrojem pro vývojáře

PRAKTICKÁ ČÁST

REALIZACE APLIKACE

Cílem této bakalářské práce bylo, navrhnout a vyzkoušet, zdali je možné vytvořit aplikaci pro reklamní systém Fakulty Aplikované Informatiky bez použití serverové logiky.

Musel jsem tedy vycházet z toho, že jediné co mám k dispozici je prohlížeč webových stránek a skriptovací jazyky, které pracují na straně klienta a dále technologii HTML a její kaskádové styly. Nemohl jsem použít žádný ze silných programovacích jazyků jako je PHP, ASP.NET nebo JSP pro zpracování informací. Dále jsem nemohl použít klasickou databázi na ukládání informací, ale musel jsem za ni najít nějakou alternativu, protože i když je aplikace malého rozsahu, bez datového úložiště jako je databáze by se neobešla.

1.9 Analýza aplikace

Při analýze systému, jsem dospěl k závěru, že aplikaci bude potřeba rozdělit do dvou samostatných částí. Hlavní část, která se stará o vytvoření nového projektu jsem pojmenoval *Editor* a vedlejší část, která se stará o zobrazení projektu jsem pojmenoval *Viewer*. Obě dvě části pracují nad stejnými databázemi.

1.9.1 Funkcionalita aplikace

Úkolem aplikace je vytvořit HTML dokument, jenž se bude pomocí FTP protokolu distribuovat na jednotlivé počítače, které obsluhují reklamní televize. HTML dokumentu určíte layout a jednotlivým částem layoutu přiřadíte seznam obrázků, které se budou v čase měnit mezi sebou v nekonečné smyčce. Každý televizor má svůj počítač, přes který se na obrazovce zobrazují HTML stránky. Televizory jsou na sobě datově nezávislé a na každý z nich se musí obsah nahrát zvlášť. Tento postup má jak své klady tak i své zápory.

1.9.2 Editor

Editor je hlavní částí aplikace. Jeho úkolem je vytvořit nový projekt, nastavit mu parametry, přiřadit obrázky a tyto data pak uložit do databáze. Editor je velmi jednoduchý, grafická část je realizována pouze pomocí HTML a CSS. Nejedná se o prezentaci, nebylo nutné vytvářet moderní grafické prvky, které by vnášely do aplikace větší pravděpodobnost chyby při přenosu z jednoho počítače na jiný.

The screenshot displays the main editor window, divided into two sections:

- Základní nastavení projektu (Basic project settings):**
 - Jméno projektu:** A text input field.
 - Rozlišení projektu:** Two input fields for 'Výška (v pixelech):' and 'Šířka (v pixelech):'.
 - Pole:** Four checkboxes for 'Horní okno', 'Hlavní okno', 'Levé okno', and 'Pravé okno'.
 - Nastavení rozměrů polí:** Three input fields for 'Šířka Horního okna (v pixelech):', 'Šířka pravého okna (v pixelech):', and 'Šířka levého okna (v pixelech):'.
 - Nastavení času změny:** Four input fields for 'Horní okno (ve vteřinách):', 'Hlavní okno (ve vteřinách):', 'Levé okno (ve vteřinách):', and 'Pravé okno (ve vteřinách):'.
 - Nastavení efektu změny:** Four dropdown menus for 'Zoom' settings for each window type.
 - A button at the bottom: 'Ulož nastavení hlavní stránky'.
- Přiřazení obrázků (Image assignment):**
 - Vyberte obrázky pro horní pole.** A dashed box containing a 'Vybrat soubor' button and 'Soubor nevybrán' text, followed by an 'Ulož' button.
 - Vyberte obrázky pro hlavní pole.** A dashed box containing a 'Vybrat soubor' button and 'Soubor nevybrán' text, followed by an 'Ulož' button.

Obr. 5. Hlavní okno editoru

1.9.3 Viewer

Viewer je jak již název napovídá zobrazovací část aplikace. Má za úkol načíst data uložená v databázi, ty pak přiřadit jednotlivým elementům v HTML kódu a modifikovat Javascriptové knihovny tak, aby se aplikace chovala přesně tak, jak si uživatel navolil. V prvním kroku vybereme požadovaný projekt, který chceme zobrazit na panelu televizoru.

The screenshot shows the viewer window with the following elements:

- Viewer - zobrazovací část aplikace.**
- Zvolte projekt:** A dropdown menu showing 'Projekt' and a 'Zobraz projekt' button.
- Instruction:** 'Po vybrání projektu klikněte na tlačítko "Zobraz projekt". Projekt se po této operaci načte do stávajícího okna.'

Obr. 6. Hlavní okno vieweru

1.10 Popis částí aplikace

1.10.1 Databáze

Hlavní myšlenkou při návrhu databáze bylo vytvořit jednoduchou a účinnou databázi s co nejmenším počtem tabulek při zachování funkčnosti aplikace a současně zachovat možnost jejího snadného rozšíření. Veškeré části kódu, které pracují s databází jsou uloženy v jednom souboru *db.js*. Databáze obsahuje pouze pět tabulek, které si popíšeme níže.

Příklad vytvoření databáze v prohlížeči:

```
// vytvoření databáze v prohlížeči

var db = window.openDatabase("mainDB", "1.0", "mainDB", 50*1024*1024);
//velikost je 50MB

window.onload = function() {

  if (!window.openDatabase) {

    status("Databáze není podporována");

    return;

  }

  db.transaction(function(tx) {

    tx.executeSql("create table if not exists project ("

      + "id integer primary key,"

      + "projectname text UNIQUE"

      + ")", null, null, logger);

    showRecords();

    console.log("Operace proběhla v pořádku");

  });
```

1.10.2 Tabulka „properties“

V této tabulce jsou uloženy všechny údaje, potřebné pro správné zobrazení projektu. Obsahuje 19 sloupců.

projectid	projectname	width	height	isNorth	isMain	isWest	isEast	northHeight	westWidth	eastWidth	northTime	mainTime	westTime	eastTime	northEffect	mainEffect	westEffect	eastEffect
1	test2	1920	1080	true	true	false	true	250	920		5	8	3	6	zoom	fade	curtainX	turnDown
2	test3	1920	1080	true	true	false	true	250	920		5	8	3	6	zoom	fade	curtainX	turnDown
3	test4	1920	1080	true	true	false	true	250	920		5	8	3	6	zoom	fade	curtainX	turnDown
4				false	false	false	false								zoom	zoom	zoom	zoom
5	Jeseter	1920	1080	false	false	false	false								zoom	zoom	zoom	zoom
6	Projekt	1920	1080	true	true	false	true	250	920		10	20	30		turnDown	fade	curtainX	zoom

Obr. 7. Databáze properties

1.10.2.1 Sloupec projectid

Sloupec projectid je primární klíč tabulky, jehož hodnota se přiřazuje automaticky při vytvoření nového záznamu a je vždy unikátní. Slouží k přesné identifikaci projektu.

1.10.2.2 Sloupec projectname

V tomto sloupci se ukládá jméno projektu, které definuje uživatel. Používá se při výběru projektu a při vyhledávání v tabulkách s obrázky. Sloupec je typu *text*.

1.10.2.3 Sloupec width

Zde se ukládá hodnota šířky projektu, která definuje, jaké rozlišení bude mít výsledný projekt. Sloupec je typu *integer*.

1.10.2.4 Sloupec height

Obdobně jako u předchozího sloupce s tím rozdílem, že tento sloupec definuje výšku projektu. Sloupec je typu *integer*.

1.10.2.5 Sloupce isNorth, isMain, isEast, isWest

V těchto sloupcích se ukládá hodnota *true* nebo *false*. Ta nám určuje, jestli se dané pole vytvoří nebo ne. Sloupce jsou typu *integer*.

1.10.2.6 Sloupce northHeight, westWidth, eastWidth

Tyto sloupce udržují hodnotu důležitou pro určení velikosti pro jednotlivá pole. Velikost pro hlavní pole není potřeba definovat, ta se dopočítává z těchto zbylých třech údajů. Sloupce jsou typu *integer*.

1.10.2.7 Sloupce *northTime*, *mainTime*, *westTime*, *estTime*

Zde se ukládá časový údaj potřebný k definici času změny obrázků. Sloupce jsou typu *integer*.

1.10.2.8 Sloupce *northEffect*, *mainEffect*, *westEffect*, *eastEffect*

Uložená hodnota v těchto sloupcích nám určí, jaký efekt bude použit při změně obrázku v konkrétním poli. Sloupce jsou typu *text*.

Příkaz pro vytvoření tabulky:

```
db.transaction(function(tx) {
  tx.executeSql("create table if not exists properties ("
    + "projectid integer primary key autoincrement,"
    + "projectname text UNIQUE,"
    + "width integer,"
    + "height integer,"
    + "isNorth text,"
    + "isMain text,"
    + "isWest text,"
    + "isEast text,"
    + "northHeight integer,"
    + "westWidth integer,"
    + "eastWidth integer,"
    + "northTime integer,"
    + "mainTime integer,"
    + "westTime integer,"
    + "eastTime integer,"
    + "northEffect integer,"
```

```
+ "mainEffect integer,"
+ "westEffect integer,"
+ "eastEffect integer"
+ ")",null,null, logger);
showRecords();
console.log("Tabulka properties vytvořena");
});
```

1.10.3 Tabulky northImages, mainImages, westImages, eastImages

Tyto tabulky obsahují obrázky k jednotlivým polím. Jsou tvořeny třemi sloupci. Důvod proč jsem nepoužil hromadně jednu tabulku pro všechny obrázky a nerozlišoval jsem je cizím klíčem je ten, že obrázky jsou uloženy v datovém formátu *Base64*, který zvýší velikost obrázku o 33% a WebSQL databáze není určena pro příliš velký objem dat, tak by se mohlo snadno stát, že po překročení určitého množství dat, by databáze mohla vykazovat jistou míru nestability.

1.10.3.1 Sloupec *imgid*

Je primárním klíčem této tabulky. Obdobně jako v tabulce *properties* sloupec *projectid*.

1.10.3.2 Sloupec *projectname*

Obsahuje název projektu, ke kterému daný obrázek patří. Sloupec je typu *text*.

1.10.3.3 Sloupec *img*

V tomto sloupci jsou uloženy obrázky převedené do formátu *Base64*. Sloupec je typu *blob*.

Příkaz pro vytvoření tabulky:

```
db.transaction(function(tx) {  
  
    // tx.executeSql("drop table if exists north");  
  
    tx.executeSql("create table if not exists northImages ("  
        + "imgid integer primary key autoincrement,"  
        + "projectname integer,"  
        + "img blob"  
        + ")",null,null, logger);  
  
    showRecords();  
  
    console.log("Tabulka vytvořena");  
  
});
```

1.10.4 Layout

Aby bylo možné korektně zpracovat vstupní data, bylo nutné definovat „okna“, která budou k dispozici. Upravil jsem k těmto potřebám model „světových stran“. K dispozici je *horní, hlavní, levé a pravé* okno. Schéma rozložení je naznačeno na obrázku.



Obr. 8. Rozložení oken v layoutu

1.10.4.1 Horní okno

Horní okno má šířku jako samotný projekt. Určujeme u něj pouze jeho výšku při vytváření projektu. Ostatní okna se řadí pod něj.

1.10.4.2 Levé okno

Levé okno, jak již název naznačuje, je na levé straně dokumentu. Výška se nezadáva, bude to vždy rozdíl mezi výškou dokumentu a výškou horního pole. Zadává se pouze šířka okna.

1.10.4.3 Pravé okno

Pravé okno je zarovnáno na pravý okraj dokumentu. Výška se obdobně jako u levého okna nenastavuje, nastavuje se pouze šířka okna.

1.10.4.4 Hlavní okno

Hlavní okno je umístěno mezi oběma postranními okny. U tohoto okna se nenastavuje žádný parametr, všechny se dopočítávají z parametrů oken ostatních.

1.11 Vytvoření nového projektu

Vytvoření nového projektu je rozděleno do dvou kroků. Prvním krokem je nastavení parametrů projektu jako je rozlišení projektu, definicí, která okna chceme zobrazit a podobně. V druhém kroku přiřadíme těmto oknům obrázky.

1.11.1 Název projektu

Název projektu je klíčové slovo pro fungování celé aplikace. Je tedy důležité, aby řetězec, který jej definuje, byl unikátní a pro uživatele dobře zařaditelný. Nastavuje se příslušným vstupním polem.



Obr. 9. Vstupní pole pro jméno projektu

Vyplnění tohoto pole má následující pravidla:

- **Pole je povinné**
- **Minimální délka názvu projektu jsou 2 znaky**

1.11.2 Rozlišení projektu

Dvě vstupní pole určují rozlišení projektu. Výšku a šířku výsledného HTML dokumentu.



Obr. 10. Vstupní pole pro nastavení rozlišení projektu

Vyplnění tohoto pole má následující pravidla:

- **Pole je povinné**
- **Může obsahovat pouze číslice**
- **Minimální počet číslic je tři**

1.11.3 Layout

V této části formuláře si určíme, která okna chceme použít pro náš výsledný dokument. Pomocí prvku *checkbox* a jeho zaškrtnutím určíme, které okno bude vykreslováno ve výsledném projektu.



Obr. 11. Nastavení zobrazování oken

1.11.4 Nastavení rozměrů oken

Zde definujeme rozměry vybraných oken. Šířku horního okna a šířku levého a pravého okna.



Obr. 12. Nastavení rozměrů jednotlivých oken

Vyplnění tohoto pole má následující pravidla:

- **Pro zpřístupnění je nutné mít vybráno příslušné okno**
- **Pole je povinné**
- **Minimální délka jsou 3 znaky**
- **Může obsahovat pouze číslice**

1.11.5 Nastavení času změny

V této části definujeme čas mezi změnou jednotlivých obrázků.



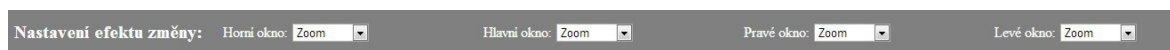
Obr. 13. Nastavení času

Vyplnění tohoto pole má následující pravidla:

- Pro zpřístupnění je nutné mít vybráno příslušné okno
- Pole je povinné
- Minimální délka jsou 3 znaky
- Může obsahovat pouze číslice

1.11.6 Nastavení efektu změny

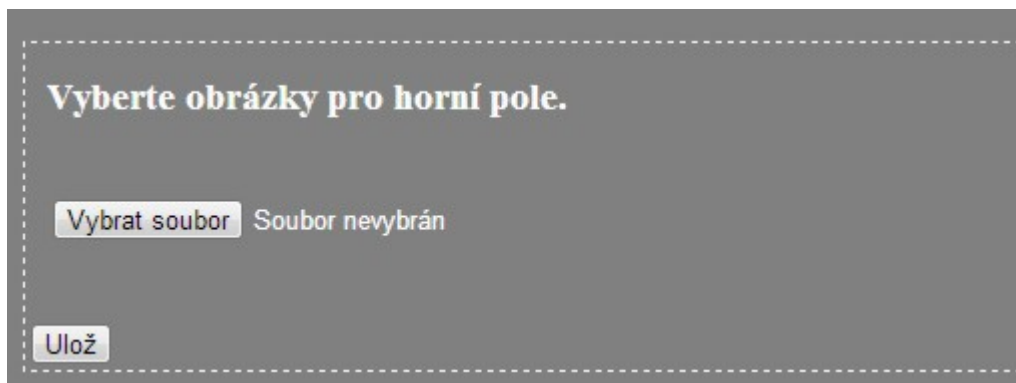
Určuje, jakým grafickým efektem se obrázky budou mezi sebou měnit. Nemá žádné omezení, explicitně je nastaven efekt „zoom“.



Obr. 14. Nastavení efektu

1.11.7 Přiřazení obrázků

V této části formuláře nastavíme jednotlivým oknům obrázky, které chceme, aby byly zobrazovány. Po vybrání obrázku se nám zobrazí jeho miniatura. Po ukončení výběru je nutné stisknout tlačítko uložit a to pro každé okno zvlášť. Práce s obrázky je realizována pomocí rozhraní File API. Načítané obrázky se převádí do Base64 a ukládají do pole. Po ukončení a uložení se tyto data uloží do databáze.



Obr. 15. Vybírání obrázků

Zdrojový kód načítání obrázků:

```
var arrayOfImagesEast = new Array;
var iteratorEast = 0;

function handleFileSelectEast(evt) {
    var filesEast = evt.target.files;
```

```
for (var i = 0, f; f = filesEast[i]; i++) {
  if (!f.type.match('image.*')) {
    continue;
  }

  var reader = new FileReader();

  reader.onload = (function(theFile) {
    return function(e) {
      arrayOfImagesEast[iteratorEast] = e.target.result;

      var span = document.createElement('span');

      span.setAttribute("id", "eastNumber" + iteratorEast)

      span.innerHTML = ['<button name = "deleteItemEast"
onClick = "removeItemEast(', iteratorEast, ')>Smazat obrazek</button>'].join("");

      document.getElementById('listEast').insertBefore(span, null);

      iteratorEast++;

    };
  })(f);

  reader.readAsDataURL(f);
};

function removeItemEast(index) {
  arrayOfImagesEast.splice(index, 1);

  iteratorEast--;

  var child = document.getElementById("eastNumber" + index);

  var parent = document.getElementById("listEast");
```

```
parent.removeChild(child);
```

1.12 Zobrazení projektu

Pokud máme hotový projekt, musíme vzít databázi a tu ručně nakopírovat na cílový počítač. FTP klient nelze realizovat jenom pomocí HTML a Javascriptu, nemohlo to být tudíž součástí řešení.

1.12.1 Zkopírování databáze

Databáze prohlížeče Google Chrome je umístěna v adresáři *AppData\Local\Google\Chrome\User Data\Default\databases*. Zde nalezneme knihovnu *Databases.db* a knihovnu *Databases.db-journal*. Tyto knihovny obsahují informace o databázích a tabulkách, které bylo vytvořeny. Dále zde nalezneme složku *File__0* která obsahuje data k těmto tabulkám. Všechny tyto soubory složky je nutné nakopírovat na cílový počítač, kde spustíme soubor *chooseProject.html* přes který vybereme projekt a výběr potvrdíme.

Název	Přípona
↑ [-]	
📁 [file__0]	
📄 Databases	db-journal
📄 Databases	db

Obr. 16. Soubory databáze

ZÁVĚR

Věřím, že tato bakalářská práce bude pro čtenáře přínosem a vnese jim nový pohled na možnosti řešení webových aplikací bez použití serverové logiky, která je značně omezující.

Hlavním cílem této práce, bylo vytvořit jednoduchý HTML editor, který bude generovat statickou webovou stránku s předem daným rozložením a obsahem, což bylo splněno.

Momentálně probíhá testování aplikace a připravují se změny, protože se ukázalo, že vyvinout systém bez použití plnohodnotných databází a programovacích jazyků jako je PHP, ASP.NET nebo JSP není při současných možnostech jazyků pracujících na straně klienta zcela optimální. Možnosti se ale mění každým dnem. Uvidíme, co nám přinese budoucnost.

ZÁVĚR V ANGLIČTINĚ

I believe that this thesis will be beneficial to the readers and will bring them a new perspective on possible solutions of web applications without using server logic, which is very restrictive.

The main objective of this work was to create a simple HTML editor that will generate a static website with a predetermined layout and content, and this was fulfilled.

Currently, testing of the application is being performed and changes are to come, because it turned out that developing a system, without the use of full-scale databases and programming languages such as PHP, ASP.NET or JSP, is not completely optimal with current options of languages working on the client side. But options change everyday. We shall see what the future will bring us.

SEZNAM POUŽITÉ LITERATURY

- [0] BOUMPHREY, Frank. *XHTML: průvodce vývojáře*. 1. vyd. Brno: Mobil Media, 2002, 479 s. ISBN 80-865-9314-2.
- [2] LUBBERS, Peter, Brian ALBERS a Frank SALIM. *HTML5: programujeme moderní webové aplikace*. Vyd. 1. Brno: Computer Press, 2011, 304 s. ISBN 978-80-251-3539-6.
- [3] W3C.org: HTML & CSS [online]. Dostupný z WWW: <<http://www.w3.org/standards/webdesign/htmlcss#whatcss>>.
- [4] STEPHENS, Ryan K, Ronald R PLEW a Arie JONES. *Naučte se SQL za 28 dní*. Vyd. 1. Překlad Lukáš Krejčí. Brno: Computer Press, 2010, 728 s. ISBN 978-80-251-2700-1.
- [5] ZAKAS, Nicholas C. *JavaScript pro webové vývojáře: programujeme profesionálně*. Vyd. 1. Překlad Lukáš Krejčí. Brno: Computer Press, 2009, 832 s. ISBN 978-80-251-2509-0.
- [6] *JQuery: kuchařka programátora*. Vyd. 1. Brno: Computer Press, 2010, 436 s. ISBN 978-80-251-3152-7.
- [7] Wikipedia – Internetová encyklopedie [online]. Dostupný z WWW:<www.wikipedia.org>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Application Programming Interface
ASP	Active Server Pages
CSS	Cascading Style Sheets
DOM	Document Object Model
FTP	Význam první zkratky.
GPL	General Public License
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
MIT	Massachusetts Institute of Technology
PHP	Hypertext Preprocessor
SQL	Structured Query Language
W3C	Význam druhé zkratky.
WWW	World Wide Web

SEZNAM OBRÁZKŮ

OBR. 1. POUŽITÍ JQUERY VALIDATION PLUGINU.....	18
OBR. 2. HLAVNÍ OKNO PROGRAMU.....	20
OBR. 3. HLAVNÍ OKNO PROGRAMU.....	21
OBR. 4. HLAVNÍ OKNO PROHLÍŽEČE SE ZOBRAZENÝM NÁSTROJEM PRO VÝVOJÁŘE.....	22
OBR. 5. HLAVNÍ OKNO EDITORU.....	24
OBR. 6. HLAVNÍ OKNO VIEWERU.....	24
OBR. 7. DATABÁZE PROPERTIES.....	26
OBR. 8. ROZLOŽENÍ OKEN V LAYOUTU.....	29
OBR. 9. VSTUPNÍ POLE PRO JMÉNO PROJEKTU.....	30
OBR. 10. VSTUPNÍ POLE PRO NASTAVENÍ ROZLIŠENÍ PROJEKTU.....	31
OBR. 11. NASTAVENÍ ZOBRAZOVÁNÍ OKEN.....	31
OBR. 12. NASTAVENÍ ROZMĚRŮ JEDNOTLIVÝCH OKEN.....	31
OBR. 13. NASTAVENÍ ČASU.....	31
OBR. 14. NASTAVENÍ EFEKTU.....	32
OBR. 15, VYBÍRÁNÍ OBRÁZKŮ.....	32
OBR. 16. SOUBORY DATABÁZE.....	34

SEZNAM TABULEK

TAB. 1. PODPORA WEBSQL DATABÁZE V PROHLÍŽEČÍCH.....	15
TAB. 2. PODPORA INDEXEDDB DATABÁZE V PROHLÍŽEČÍCH.....	16

SEZNAM PŘÍLOH

P I: CD-ROM

PŘÍLOHA P I: CD-ROM

Součástí práce je přiložené CD, které obsahuje všechny zdrojové kódy a bakalářskou práci ve formátu .pdf a .doc.

