

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA PEDAGOGICKÁ  
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**JAVASCRIPTOVÉ A PHP FRAMEWORKY A KNIHOVNY JAKO  
POMOCNÍK PŘI TVORBĚ WEBŮ S ROZŠÍŘENÝMI FUNKCEMI**  
BAKALÁŘSKÁ PRÁCE

**Petr Neužil**

*Informatika se zaměřením na vzdělávání*

Vedoucí práce: PhDr. Tomáš Jakeš, Ph.D.

**Plzeň, 2016**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury a zdrojů informací.

V Plzni, 1. června 2016

.....  
vlastnoruční podpis

Děkuji vedoucímu práce PhDr. Tomáši Jakešovi, Ph.D. za cenné rady a připomínky při tvorbě této bakalářské práce.

## OBSAH

Úvod .....	4
1 ZÁKLADNÍ POJMY .....	6
1.1 KNIHOVNA .....	6
1.2 FRAMEWORK .....	6
1.3 DALŠÍ POJMY TÝKAJÍCÍ SE TÉMATU .....	6
2 MOŽNOSTI USNADNĚNÍ TVORBY WEBOVÉ APLIKACE .....	8
2.1 DŮVODY PRO UŽITÍ FRAMEWORKŮ .....	8
2.2 JAKÁ KRITÉRIA ZOHLEDNIT PŘI VÝBĚRU FRAMEWORKU .....	8
2.3 DŮVODY PRO UŽITÍ KNIHOVEN .....	9
2.4 JAKÁ KRITÉRIA ZOHLEDNIT PŘI VÝBĚRU KNIHOVEN .....	9
3 PŘEHLED FRAMEWORKŮ .....	11
3.1 PHP FRAMEWORKY .....	11
3.1.1 Uživatelská komunita .....	11
3.1.2 Kvalita dokumentace .....	12
3.1.3 Licence .....	12
3.1.4 Domovská stránka .....	12
3.1.5 Databázová migrace .....	13
3.1.6 Caching .....	13
3.1.7 Rozšiřitelnost frameworku .....	13
3.1.8 Vícejazyčné weby .....	13
3.1.9 Komponenta pro odesílání emailů .....	14
3.1.10 Práce s formuláři .....	14
3.1.11 Objektový přístup .....	14
3.1.12 Přehled PHP frameworků .....	15
3.1.13 Závěry .....	16
3.2 LARAVEL .....	17
3.2.1 Tvorba vícejazyčných webů - ukázka .....	18
3.2.2 Formuláře - ukázka .....	20
3.2.3 AJAX request - ukázka .....	21
3.3 JAVASCRIPTOVÉ FRAMEWORKY .....	22
3.3.1 Dokumentace a uživatelská komunita .....	22
3.3.2 Kompatibilita s prohlížeči .....	22
3.3.3 Přehled Javascriptových frameworků .....	23
3.3.4 Závěry .....	23
3.4 ANGULARJS .....	24
3.4.1 Nové atributy .....	25
3.4.2 Řadiče, Objekty, filtry a služby .....	25
3.4.3 Dvoucestná synchronizace dat - ukázka .....	25
3.4.4 Výrazy - ukázka .....	26
3.4.5 Tabulka - ukázka .....	26
3.4.6 Formuláře - ukázka .....	27
4 PŘEHLED KNIHOVEN .....	28
4.1 GRAFIKA – OBRÁZKY A GALERIE .....	28
4.1.1 PHP Thumb .....	28
4.1.2 PHP Text to Image .....	28
4.1.3 Zebra Image .....	29

4.1.4	Imagine .....	29
4.1.5	Snappy .....	29
4.1.6	Widelmage.....	29
4.1.7	UberGallery.....	30
4.1.8	Coppermine .....	30
4.1.9	Ilustrace řešení problémů manipulace s obrázky .....	30
4.1.10	Ukázka tvorby galerie pomocí UberGallery .....	31
4.2	GRAFIKA - GRAFY A MAPY .....	32
4.2.1	FusionCharts .....	32
4.2.2	Ember Charts .....	32
4.2.3	Chart.js.....	33
4.2.4	Aristochart .....	33
4.2.5	BonsaiJS .....	33
4.2.6	Morris.js.....	34
4.2.7	Amcharts.....	34
4.2.8	Amcharts - ukázkový graf .....	34
4.2.9	Amcharts – ukázková mapa.....	36
4.2.10	Morris.JS – ukázka grafu.....	38
4.3	FORMULÁŘE A TABULKY .....	39
4.3.1	Fancy Form .....	39
4.3.2	ApPHP DataGrid.....	39
4.3.3	PHP Form Validation Script.....	40
4.3.4	EditableGrid .....	40
4.3.5	dhtmlxGrid.....	40
4.3.6	FineUploader .....	41
4.3.7	DataTables .....	41
4.3.8	ApPHP DataGrid – ukázka zobrazení dat.....	41
4.3.9	Fancy Form – ukázka jednoduchého formuláře .....	42
4.4	DATABÁZE .....	44
4.4.1	ADODB .....	44
4.4.2	PHP Advanced Database Class.....	44
4.4.3	sqlCache.....	45
4.4.4	CatchMyFame.....	45
4.4.5	Danpu .....	45
4.4.6	Danpu – záloha databáze .....	45
4.5	MANIPULACE S DATY .....	46
4.5.1	jQuery .....	46
4.5.2	PHP Collection .....	46
4.5.3	Services_JSON.....	47
4.5.4	TCPDF.....	47
4.5.5	PHPDocumentor .....	47
4.5.6	jQuery – jednoduchá ukázka .....	47
4.6	CAPTCHA.....	48
4.6.1	ReCAPTCHA.....	48
4.6.2	SimpleCaptcha .....	48
4.6.3	Securimage .....	49
4.6.4	BotDetect PHP CAPTCHA Generator .....	49
4.6.5	s3Capcha.....	49

---

4.6.6	Captchas.net .....	49
4.6.7	Visual and Audio PHP CAPTCHA Generation Class .....	50
4.6.8	freeCap PHP CAPTCHA.....	50
4.6.9	Cool PHP CAPTCHA .....	50
4.6.10	QuickCaptcha 1.0.....	51
4.6.11	Simple Javascript CAPTCHA .....	51
4.6.12	hoomanTest: A Mootools Bot Filtering CAPTCHA Class .....	51
4.6.13	ReCAPTCHA - Ukázkové řešení .....	52
4.7	OSTATNÍ.....	52
4.7.1	BigPicture.....	53
4.7.2	HTML Purifier.....	53
4.7.3	XML-RPC PHP Library.....	53
4.7.4	PHPMailer .....	54
4.7.5	SwiftMailer .....	54
4.7.6	Geocoder .....	54
4.7.7	Opauth.....	54
	ZÁVĚR.....	55
	RESUMÉ .....	56
	SEZNAM LITERATURY .....	57
	SEZNAM OBRÁZKŮ A TABULEK .....	63

## Úvod

Současným trendem v oblasti vývoje webových aplikací je tendence vývojářů, co nejvíce si jejich tvorbu usnadnit. K usnadnění tvorby pomáhá velké množství různých frameworků a knihoven. Vývoj webových aplikací se díky tomu stává mnohem rychlejším a levnějším. Děje se tak z toho důvodu, že vývojáři při tvorbě webů využívají předpřipravených základů, se kterými dále pracují. Programování webových aplikací tzv. „od nuly“ je dnes spíše vzácností a téměř každá firma zabývající se webovými technologiemi požaduje na svých zaměstnancích navíc ke znalosti programovacích jazyků také znalost některého z frameworků. [1] [2]

Frameworky i knihovny jsou dnes velmi rozšířené. Téměř na každý problém nebo požadovanou funkci můžeme najít konkrétní knihovnu a tím si práci značně usnadnit. i když je většina těchto „pomocníků“ (pod tímto pojmem si představte framework, knihovnu, modul, plugin či komponentu) dostupná zdarma, některé z nich mají určitá omezení nebo jsou dostupné pouze za poplatek. [2]

Téma této práce jsem si vybral hlavně proto, že již několik let programuji aplikace v PHP a za tu dobu jsem se setkal s řadou knihoven, které mi usnadnily práci a opravdu výrazně zkrátily čas potřebný k naprogramování mnoha funkcionalit. Díky tomu je mi toto téma velice blízké.

Cílem práce je vytvořit ucelený přehled vybraných „pomocníků“ a usnadnit tak čtenáři orientaci v dostupných knihovnách a frameworkcích. Informovat o důležitých vlastnostech a funkcích jednotlivých nástrojů, a určit kritéria, podle kterých je možno jednotlivé knihovny či frameworky porovnávat. Nakonec na několika konkrétních příkladech konkrétně předvést vybrané přednosti těchto nástrojů. Ve vhodných případech doplnit přehled o obrázkové ilustrace.

Práce bude rozčleněna do čtyř kapitol. V první kapitole půjde o stručné představení a definici základních pojmů souvisejících s tématem. Druhá kapitola obecně shrne důvody proč frameworky a knihovny používat a co je třeba zohlednit při výběru vhodného nástroje. Ve třetí kapitole budou kritéria pro výběr podrobněji popsána a představena v souvislosti s uceleným přehledem PHP<sup>1</sup> a Javascriptových frameworků. Dále zde budou blíže

---

<sup>1</sup> PHP - Hypertext Preprocessor

představeny dva vybrané frameworky, které mě z obou přehledů zaujaly nejvíce. Čtvrtá a poslední kapitola se bude věnovat knihovnám, které budou rozčleněny do kategorií podle způsobu využití. U každé knihovny bude podle vhodnosti uveden buď krátký popis, nebo obrázek, který danou knihovnu vystihuje.

Pro správné pochopení této práce je třeba mít základní povědomí o tvorbě webových stránek a schopnost porozumět zdrojovým kódům v Javascriptu a PHP.



## 1 ZÁKLADNÍ POJMY

Zde jsou vymezeny a definovány základní pojmy související s touto prací. Kromě toho jde o jednoduché a stručné představení témat, kterými se tato práce podrobněji zabývá v dalších kapitolách.

### 1.1 KNIHOVNA

Knihovna je soubor funkcí či metod řešící nejčastěji jeden konkrétní problém. Je vytvořena buď pro jeden konkrétní programovací jazyk nebo více jazyků najednou (například jQuery). Jedná se o kód vytvořený za účelem zjednodušení a zrychlení procesu programování. i když to není podmínkou, většinou se při tvorbě knihoven používá objektový přístup. Protože se zpravidla jedná o kód, který by měl používat i někdo jiný, než jeho autor, měl by být tento kód řádně okomentovaný. Tím se zvýší intuitivnost a rychlost práce s danou knihovnou. Pokud je knihovna dobře zdokumentovaná, dovolí nám vyřešit problém, pro který je určená, v mnohem kratším časovém období. Tato práce se zabývá PHP a Javascriptovými knihovnami. [3]

### 1.2 FRAMEWORK

Framework je stejně jako knihovna kód definující funkce a metody. Jedná se však zpravidla o robustnější nástroj řešící více problémů najednou. Součástí frameworku je tedy většinou více knihoven. Jedná se o prostředí vytvořené v konkrétním programovacím jazyce. Zjednodušeně jde o nástroj pro vývojáře, který usnadňuje tvorbu aplikací konkrétního typu. Jako příklad lze uvést webové aplikace, kterými se zabývá i tato práce. Naučit se dobře pracovat s frameworkem vyžaduje daleko více času, než seznámení se s jednou knihovnou. Programátor pak ale získává mocný nástroj, který mu pomůže snadněji a rychleji řešit konkrétní funkcionality aplikace, které požaduje zákazník. Framework se skládá z tzv. „hot spots“ a „frozen spots“. „Frozen spots“ představují neměnnou strukturu frameworku. Zato „hot spots“ jsou části, které vývojář programuje za účelem dosažení konkrétního požadovaného výsledku. Většina frameworků je objektově zaměřená. Framework se dá s trochou nadsázky označit jako soubor knihoven. [4]

### 1.3 DALŠÍ POJMY TÝKAJÍCÍ SE TÉMATU

Vývoj webové aplikace lze usnadnit také použitím redakčního systému. Pokud vyvíjíme web v některém z redakčních systémů, které jsou velice rozšířené, můžeme použít pro řešení

konkrétního problému některý z dostupných pluginů, případně si naprogramovat vlastní modul/komponentu. Plugin v tomto případě znamená rozšíření redakčního systému. Jako příklad lze použít galerii. Některé redakční systémy ji v kostře kódu neobsahují, pravděpodobně z důvodu optimalizace výkonu. Ne každý totiž ve své webové aplikaci galerii potřebuje. Ve většině případů se dá stáhnout zásuvný modul, chcete-li plugin či komponenta, která galerii implementuje a programátorovi pak většinou stačí lehká úprava zdrojového kódu k dosažení požadovaného výsledku. V nejlepším případě jen provede nastavení v redakčním systému. [5]

Plugin i modul jsou ve spojení s webovými aplikacemi synonyma. U každého nástroje se rozšíření nazývá jinak. Někdy se pro stejný význam používá anglické slovo „extension“. Toto slovo se do češtiny překládá přímo jako „rozšíření“. [6]

## 2 MOŽNOSTI USNADNĚNÍ TVORBY WEBOVÉ APLIKACE

V této kapitole jsou rozebrány důvody, proč užívat frameworky a knihovny. Dále je obecně uvedeno podle čeho tyto nástroje vybírat a na co si při výběru dát pozor. Zmíněná kritéria v této kapitole slouží jako podklad pro další kapitoly, kde jsou podrobněji rozebrána a konkrétně aplikována do kontextu vybraných frameworků a knihoven.

### 2.1 DŮVODY PRO UŽITÍ FRAMEWORKŮ

Hlavní výhodou frameworku je fakt, že pokud se v něm programátor naučí pracovat, velmi se tím zvýší efektivita práce. Kód se použitím frameworku stane zpravidla přehlednějším a zlepší se jeho dlouhodobá udržitelnost. Díky tomu jde o výborný nástroj pro týmovou práci. Vývojář nemusí programovat vše od začátku a případná rozšíření aplikace jsou často mnohem snadnější. Výstižně výhody využití frameworků vystihuje tento text: „Komponenty pro opakující se úlohy jako databázové dotazy, zpracování šablon, tvorba formulářů, ověřování údajů nebo posílání e-mailů nemusí být pro každý nový projekt znovu napsány a mohou být použity opakovaně. Dlouhá doba zaučení se při prvním projektu sice neukazuje jako efektivní, ale s každým dalším projektem poznáváte Zend Framework lépe a lépe a můžete se od té chvíle soustředit na podstatné věci, jako jsou požadavky vašeho klienta nebo implementace přání vašeho týmu. Jestliže se raději rozhodnete pro Open Source framework, přesouváte vývoj frameworku směrem ven a máte tak čas na vlastní aplikaci. Komunita se stará o další vývoj frameworku a vy můžete pomocí žádoucích změn a hlášení o chybách ovlivnit jeho evoluci.“ [1] Jako nevýhoda se uvádí, že je kód většinou pomalejší, než řešení přímo na míru. Především kvůli tomu, že framework obsahuje nadbytečný kód, který pro konkrétní řešení není potřeba. Neplatí to však ve všech případech. Framework je vyvíjen odděleně od PHP či Javascriptu a není jejich oficiální součástí, může se tak snadno stát, že skupina či jednotlivec vyvíjející framework s vývojem skončí a pak nastává velký problém z hlediska neopravených bezpečnostních chyb. [7] [8]

### 2.2 JAKÁ KRITÉRIA ZOHLEDNIT PŘI VÝBĚRU FRAMEWORKU

Při výběru je dobré zjistit si, jak kvalitní dokumentaci framework obsahuje, stejně jako reference o jeho zabezpečení. Dále je třeba hledět na velikost uživatelské komunity, recenze a v nejlepším případě otestovat několik frameworků pro konkrétní projekt z důvodu optimalizace a nalezení nejlepšího a nejrychlejšího řešení. To však může být

časové velmi náročné. Učení se pracovat s novým frameworkem není nic snadného. Každá firma tedy zvolí zpravidla jeden framework, který používá primárně ve všech svých projektech. Dalším důležitým parametrem při výběru je cena. Některé z frameworků fungují v základní verzi zdarma, ale pokud chceme pokročilé funkce, jako je například velice důležitá optimalizace a komprese kódu, často musíme zaplatit. Dokoupení komponent nebo pluginů pak zpřístupní další funkcionality. Je dobré si kromě jiného ujasnit, které funkce od frameworku vyžadujeme a toto při výběru zohlednit. Záleží také, zda jsme ochotni doprogramovat si chybějící funkcionality nebo hledat pro každou z nich zásuvný modul. Určitě je třeba zkontrolovat, jestli vybraný framework vůbec poběží na verzi PHP nainstalované na našem serveru. U Javascriptových frameworků si musíme také ujasnit, na kterých prohlížečích má naše aplikace běžet a zjistit si, jak je to s optimalizací a podporou těchto prohlížečů. [9]

### 2.3 DŮVODY PRO UŽITÍ KNIHOVEN

Stejně jako u frameworků je hlavním důvodem pro využití některé z hotových knihoven usnadnění řešení konkrétního problému. Vše souvisí s efektivitou práce a díky předpřipraveným řešením v rámci knihovny probíhá programování výrazně rychleji. Peter Gasston v souvislosti s Javascriptovými knihovnami uvádí, že knihovny jsou kolekce předpřipravených skriptů, které mají za cíl snížení nákladů na vývoj aplikací. Ke snížení nákladů dochází díky faktu, že ať už chceme programovat cokoliv, existuje velká šance, že někdo již napsal knihovnu řešící náš problém. Zmiňuje také, že v poslední době došlo k velkému nárůstu počtu dostupných knihoven. [2] Na rozdíl od frameworků není většinou k seznámení se s knihovnou potřeba velké množství času. Dalším důvodem k použití knihoven je lepší čitelnost kódu a jeho snadnější udržitelnost oproti programování aplikace bez použití pomocného nástroje. Pokud se člověk k projektu vrátí později, dokáže se pak velmi snadno a rychle zorientovat ve starším zdrojovém kódu. [7]

### 2.4 JAKÁ KRITÉRIA ZOHLEDNIT PŘI VÝBĚRU KNIHOVEN

Jedním z hlavních kritérií při výběru knihovny by určitě měla být kvalita a zpracování dokumentace. Díky kvalitně zpracované a přehledné dokumentaci může být práce s knihovnou velice usnadněna. S dokumentací také souvisí intuitivnost knihovny, neboli jak dlouho trvá programátorovi, než se s konkrétní knihovnou dokáže plně sžít a začne ji skutečně efektivně využívat. Neméně důležitým faktorem je kromě jiného uživatelská

přívětivost výstupu knihovny směrem k návštěvníkům webu. Důležitá je také podpora v případě řešení programátorských či bezpečnostních problémů.

### 3 PŘEHLED FRAMEWORKŮ

Porovnávat frameworky mezi sebou je velice složité. Neexistuje framework, který je ve všem lepší, než ty ostatní. Jeden může být dobrý na formuláře, další na administraci a jiný zase vyniká tím, že má nejmenší počet objevených chyb. V tomto přehledu není porovnávána vhodnost frameworku právě pro konkrétní funkci. Pouze je uvedeno, zda vybrané funkce framework podporuje. Pro zobrazení přehledu PHP i Javascriptových frameworků slouží tabulky s vybranými informacemi o jednotlivých nástrojích. V prvních tabulkách jde o PHP frameworky a v dalších už o frameworky Javascriptové. K oběma přehledům jsou dostupné doplňující komentáře.

#### 3.1 PHP FRAMEWORKY

V této podkapitole jsou nejprve představena a rozebrána konkrétní hodnotící a porovnávací kritéria, která jsou použita pro vytvoření tabulkového přehledu PHP frameworků. Tabulkový přehled je následně okomentován.

##### 3.1.1 UŽIVATELSKÁ KOMUNITA

Prvním porovnávacím parametrem je velikost uživatelské komunity. Jedná se o, podle mého názoru, velice důležitý parametr. Čím větší je komunita, tím větší šance, že konkrétní problém přede mnou již někdo řešil. Mohu tak najít i více různých hotových řešení, ze kterých si lze vybrat to, které mi vyhovuje nejvíce. Pro určení velikosti komunity jsou zvolena tři čísla. První z nich je počet odpovědí na známém a prestižním programátorském komunitním fóru StackOverflow. StackOverflow navštěvují často i autoři frameworků a používají ho ke komunikaci s komunitou a společnému řešení problému. Pokud je otázka správně položena, většinou se tazateli dostane odpovědi od špičkových odborníků svého oboru. Již několikrát jsem zde viděl příspěvky od programátorů Facebooku, Microsoftu a jiných předních firem z oblasti vývoje softwaru. Toto číslo je asi nejrelevantnější ze všech tří, co se týká určení velikosti komunity. Samozřejmě je třeba toto číslo brát s nadhledem, protože komunita okolo konkrétního nástroje se může shromažďovat i na jiných místech. Například přímo na oficiálním diskusním fóru frameworku. Jedná se tedy o sloupec s názvem *StackOverflow*. Druhým číslem je počet odkazů na nejznámějším a celosvětově nejpoužívanějším vyhledávači Google.com. Vychází se zde z předpokladu, že čím více výsledků Google vrátí, tím je framework mezi programátory populárnější. Sloupec se jmenuje *Google*. Posledním číslem je počet videí na Youtube.com. Zde je předpoklad,

že většina výsledků, které Youtube vrátí, budou instruktážní videa či videa řešící problémy. To je opět ukazatel popularity a celkově velikosti uživatelské komunity. Sloupec má název *Youtube*. Všechna tři čísla je samozřejmě třeba brát s určitým nadhledem.

### 3.1.2 KVALITA DOKUMENTACE

Dalším srovnávacím parametrem je kvalita dokumentace. V tomto případě se při hodnocení jedná čistě o můj subjektivní názor. Dobře čitelná, přehledná a kvalitní dokumentace je základem úspěchu každého frameworku. Usnadňuje orientaci ve frameworku a ovlivňuje, jak rychle se v něm vývojář naučí pracovat. Možná hodnocení jsou 0-3 hvězdičky. Žádnou hvězdičku však nedostal ani jeden framework, protože všechny hodnocené měly alespoň nějakou dokumentaci. Jedna hvězdička „\*” znamená špatně čitelnou, nepřehlednou dokumentaci. Dvě hvězdičky „\*\*” dostaly dokumentace, které byly přehledné, ale občas chyběla možnost vyhledávání nebo byly příliš stručné a tím ztěžovali pochopení. Tři hvězdičky „\*\*\*” dostaly dokumentace, ve kterých šlo snadno najít vše potřebné a nebyl výrazný problém s pochopením konkrétních informací o jednotlivých částech frameworků. Sloupec *Dokumentace*. [9]

### 3.1.3 LICENCE

Posledním srovnávacím parametrem je licence, pod kterou je framework šířen. Jedná se zpravidla o nějaké podmínky, při jejichž dodržování lze framework používat. Ve zkratce zde shrnu základní rozdíly mezi jednotlivými licencemi. Největší svobodu nám dává licence BSD nebo New BSD, kde stačí pro volné pouze uvést jméno autora. MIT licence znamená, že nelze framework upravit a pak prodávat, jinak lze dělat téměř cokoliv a licence se na rozdíl od BSD/New BSD vztahuje i na dokumentaci. GNU GPL licence se vyznačuje hlavně tím, že i po úpravě díla musí být tato licence zachována. S licencí CC neboli Creative Commons je to poněkud složitější. Pro účely této práce postačí vědět, že jQWidgets, kterých se to týká, lze používat pod touto licencí bezplatně v případě, že nejde o komerční použití, zachováme jméno autora, nebudeme mazat žádný copyright text z hlaviček javascriptových souborů a upozorníme uživatele aplikace na to, že jQWidgets nejsou dostupné zdarma pro komerční využití. Sloupec *Licence*. [10] [11] [12] [13]

### 3.1.4 DOMOVSKÁ STRÁNKA

Tento parametr je uveden z důvodu snadného a rychlého nalezení frameworku v případě, že se ho někdo rozhodne použít. Sloupec *Homepage*.

### 3.1.5 DATABÁZOVÁ MIGRACE

Databázová migrace slouží k zásahům do databáze přes PHP skripty. Většinou se používají dvě metody. První metoda se jmenuje „*up*“ nebo má jiný podobně jasný název. Do této metody definujeme námi požadované úpravy databáze. Druhá metoda se jmenuje například „*down*“. Do této metody zapíšeme příkazy, které změny databáze vrátí zpět. Riziko problémů souvisejících s nesprávnou manipulací s daty v databázi se výrazně zmenší. Sloupec *DB Migrace*. [14]

### 3.1.6 CACHING

Caching se u webových aplikací používá pro zrychlené načítání obsahu. Využívá se při něm faktu, že některé URL adresy jsou vyžadovány od návštěvníků častěji. Nejlepším příkladem je hlavní stránka. Ať by byl skript, který vykoná dotaz a vrátí výsledek seberychejší, nikdy se v rychlosti nevyrovná načtení obyčejné čisté HTML stránky. Díky cachování dojde k vykonání PHP skriptu a ke generování obsahu pouze jednou za čas například pomocí CRON jobu. Uživateli se pak vrací pouze předem vygenerované HTML. Velice důležité je správné nastavení cachování, protože při příliš dlouhých intervalech aktualizace HTML souboru uživatel nemusí dostat aktuální data. Příkladem mohou být například diskusní fóra, kde by lidé neviděli ihned nové příspěvky při špatném nastavení cachování. Špatně je samozřejmě také nastavení příliš krátkých intervalů. Může pak dojít ke zbytečné zátěži serveru či výpadcích výstupu, pokud někdo na stránku přistoupí přesně v čas generování a zápisu souboru. Sloupec *Caching*. [15]

### 3.1.7 ROZŠÍŘITELNOST FRAMEWORKU

Tento sloupec informuje o tom, zda je možné konkrétní framework rozšířit bez zásahu do jádra kódu. Jedná se tedy o rozšíření pomocí vlastního či staženého pluginu. Sloupec *Rozšiřitelnost*.

### 3.1.8 VÍCEJAZYČNÉ WEBY

V tomto sloupci je uvedeno, zda framework podporuje tvorbu vícejazyčných webů. Navíc se rozlišuje, zda je nutné pro tuto funkcionalitu stahovat zvláštní modul. Sloupec *Více jazyků*.



### **3.1.9 KOMPONENTA PRO ODESÍLÁNÍ EMAILŮ**

Téměř každá webová aplikace vyžaduje nějaký způsob kontaktu s návštěvníky či zákazníky. U vybraných frameworků je zkoumáno, zda obsahují přímo integrovaný nástroj pro práci s emaily nebo musíme pro tuto funkcionalitu použít externí modul. Sloupec *Emailing*.

### **3.1.10 PRÁCE S FORMULÁŘI**

V závislosti na typu aplikace se může hodit nástroj pro práci s formuláři, který usnadní tvorbu formulářů. Takový nástroj například usnadní validaci vstupů, přizpůsobení vzhledu nebo poskytuje základní bezpečnostní ochranu. Je zkoumáno, zda framework nástroj pro práci s formuláři obsahuje nebo je nutné tuto funkcionalitu doprogramovat či použít připravený plugin. Sloupec *Formuláře*.

### **3.1.11 OBJEKTIVÝ PŘÍSTUP**

V přehledu je uvedeno, zda framework jde o framework objektově orientovaný. Sloupec *Objekty*.

## 3.1.12 PŘEHLED PHP FRAMEWORKŮ

<u>Framework</u>	<u>Verze PHP</u>	<u>Licence</u>	<u>DB migrace</u>	<u>Caching</u>	<u>Dokumentace</u>
CakePHP 3	5.4	MIT	Ano	Ano	**
Laravel	5.5.9	MIT	Ano	Ano	***
Nette Framework	5.3.0	New BSD/GNU GPL	jako plugin	Ano	***
Symfony	5	MIT	jako plugin	Ano	**
Symfony2	5.3.3	MIT	jako plugin	Ano	**
Yii	5.1.0	New BSD	Ano	Ano	**
Yii 2	5.4.0	New BSD	Ano	Ano	**
Zend framework	5.3	New BSD	Ano	Ano	***
Zend framework 2	5.3.3	New BSD	Ano	Ano	***
FuelPHP	5.3	MIT	Ano	Ano	**
PHPixie	5.3	BSD	jako modul	Ano	*
Phalcon	5.3	BSD	Ano	Ano	***

Tabulka 1 - PHP frameworky část 1

<u>Framework</u>	<u>StackOverflow</u>	<u>Google</u>	<u>Youtube</u>	<u>Homepage</u>
CakePHP 3	26 400	598 000	15 100	cakephp.org
Laravel	39 855	484 000	78 700	laravel.com
Nette Framework	247	429 000	1 020	nette.org
Symfony	44 600	574 000	27 500	symfony.com
Symfony2	40 657	668 000	12 200	symfony.com
Yii	43 420	745 000	60 100	www.yiiframework.com
Yii 2	6 100	649 000	57 300	www.yiiframework.com
Zend framework	20 878	636 000	15 800	framework.zend.com
Zend framework 2	7 869	965 000	17 800	framework.zend.com
FuelPHP	461	281 000	704	fuelphp.com
PHPixie	2	29 600	116	phpixie.com
Phalcon	1 263	480 000	6 290	phalconphp.com/en

Tabulka 2 - PHP frameworky část 2

<u>Framework</u>	<u>Rozšiřitelnost</u>	<u>Více jazyků</u>	<u>Emailing</u>	<u>Formuláře</u>	<u>Objekty</u>
CakePHP 3	Ano	Ano	Ano	Ano	Ano
Laravel	Ano	Ano	Ano	Ano	Ano
Nette Framework	Ano	Ano	Ano	Ano	Ano
Symfony	Ano	Ano	Ano	Ano	Ano
Symfony2	Ano	Ano	Ano	Ano	Ano
Yii	Ano	Ano	jako modul	Ano	Ano
Yii 2	Ano	jako modul	jako modul	Ano	Ano
Zend framework	Ano	Ano	Ano	Ano	Ano
Zend framework 2	Ano	Ano	Ano	Ano	Ano
FuelPHP	Ano	Ano	Ano	Ano	Ano
PHPixie	Ano	jako modul	Ano	Ano	Ano
Phalcon	Nezjištěno	Ano	Ano	Ano	Ano

Tabulka 3 - PHP frameworky část 3

### 3.1.13 ZÁVĚRY

Z tabulek lze vyčíst, že pokud chceme používat většinu frameworků, měli bychom používat PHP ve verzi 5 a výš. Všechny sledované frameworky jsou dostupné pod jednou z licencí zdarma. Všechny také podporují databázovou migraci. Některé však pouze po instalaci přídatného modulu. U *Symfony* a *Symfony2* se většinou k tomuto účelu používá modul s názvem *Doctrine*. Stejný nástroj se dá použít i u *Nette Framework*. Zajímavostí je, že tento framework vytvořil český autor David Grudl. *PHPixie* využívá pro databázovou migraci vlastní modul s názvem *PHPixie-Migrate*. Caching podporují všechny sledované frameworky. Zajímavá jsou čísla, která pomáhají určit velikost komunity a popularitu, kdy český *Nette Framework* je sice v tuzemsku velmi populární a používaný celou řadou firem, na StackOverflow se však řeší velice málo. Celosvětově tedy takový úspěch mít nejspíš nebude, i když počet odkazů na Google je srovnatelný s ostatními frameworky. Ze srovnání vyšly vítězně, co se týče popularity tři frameworky – *Symfony*, *Laravel* a *Yii*, když obsadily přední místa v počtu vyřešených problémů na StackOverflow, počtu odkazů na Google i počtu videí na Youtube. Nejpřehledněji a nejintuitivněji působí dokumentace frameworků *Laravel*, *Nette Framework*, *Zend Framework* a *Phalcon*. Naopak dokumentace nástroje *PHPixie* není příliš povedená z důvodu nepřehlednosti a absence vyhledávání. Všechny frameworky podporují rozšíření bez zásahů do hlavní struktury. U frameworku *Phalcon* toto však nebylo zjištěno. Na podporu vícejazyčných webů se můžeme spolehnout u všech nástrojů, i když do *Yii 2* a *PHPixie* je nutné pro tuto funkcionalitu stáhnout zásuvný modul. Pro framework *Yii 2* je k dispozici zásuvný modul s názvem *yii2-multi-language*. Pro framework *PHPixie* se nachází na serveru Github.com modul s totožným názvem *PHPixie* od uživatele „Aleksandern“. Odesílání emailů podporují všechny frameworky. U obou verzí nástroje *Yii* je však nutné použít externí rozšíření. Pro obě verze *Yii* frameworku doporučuji použít rozšíření *Swiftmailer*. Všechny frameworky tohoto přehledu obsahují alespoň základní nástroj pro práci s formuláři a používají objektový přístup.

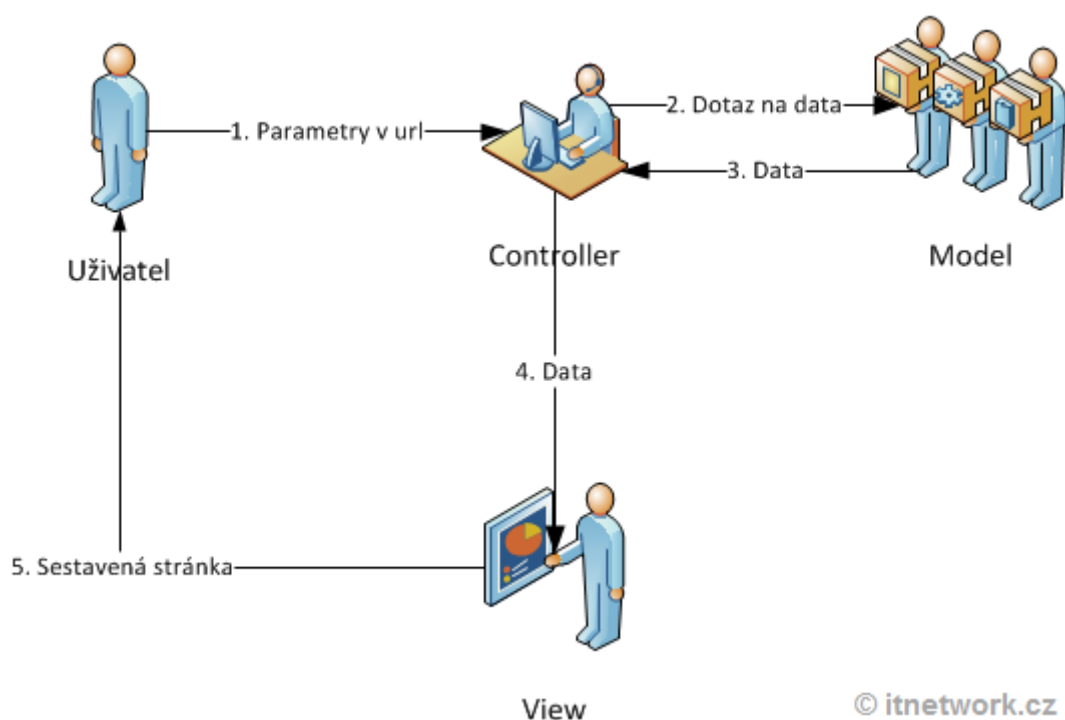
Tyto zdroje slouží jako podklad pro přehled PHP frameworků (velká většina jsou dokumentace jednotlivých nástrojů): [16] [1] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26]

Zajímavostí v tomto přehledu je PHP framework *Phalcon*, který je naprogramovaný v C. Díky tomu dosahuje skvělého výkonu bez ohledu na robustnost, i když je OpenSource, tak je čtení a úprava zdrojového kódu však mnohem složitější, zvláště pro lidi, kteří C zrovna

neovládají. Nyní podrobněji představím PHP framework Laravel, který mě zaujal ze všech nejvíce a to především svojí jednoduchostí a velkou popularitou. [22]

### 3.2 LARAVEL

Tento PHP framework přesáhl nedávno 3 000 000 stažení ze serveru Packagist.org. O jeho velké popularitě tedy nemůže být pochyb. Jeho autorem je Taylor Otwell. Jedná se o relativně nový framework. První verze spatřila světlo světa teprve v roce 2012. Framework je open-source. Zdrojový kód je tedy volně přístupný a na jeho vývoji se tak může podílet kdokoli. Je šířen zdarma pod licencí MIT. Jde o tzv. MVC<sup>2</sup> framework. To znamená, že je navržený podle této softwarové struktury. Na obrázku je zjednodušeně zobrazeno, o co se jedná. Uživatel vyvolá požadavek, který převezme Controller neboli řadič. Řadič slouží jako prostředník. Odešle dotaz na data. Komponenta Model vrátí data Controlleru, který je přepoše komponentě View. Ta data zformátuje a převede do formy, ve které je předkládá uživateli. Schéma MVC je vidět na obrázku dole. [27] [28] [29]



Obrázek 1 – Model View Controller, zdroj: itnetwork.cz

Velké množství frameworků je vytvářeno podle tohoto vzoru. Laravel tedy v tomto ohledu není žádnou výjimkou. Podrobněji co je MVC: „M jako model, V jako view (pohled) a C jako controller (řadič) – co to ale všechno znamená? MVC lze nejlépe popsat jako něco,

<sup>2</sup> MVC – Model-View-Controller

co se často označuje jako *návrhový vzor*, tedy opětovně použitelné řešení opakujícího se problému v určitém kontextu. Díky návrhovým vzorům je možné problémy spojené s návrhem rozsáhlých aplikací řešit konzistentním způsobem. Nejlepších řešení je možné dosáhnout použitím architektury, která je ověřená a vyzkoušená. V tomto případě spočívá problém v tom, jak separovat uživatelské rozhraní programu (pohled), utility a business třídy, které provádí komplexní logiku (model), a vnitřní zpracování a rozhodování (řadič) tak, aby se jednalo o tři samostatné, oddělené komponenty. Řešením je (jak už jste asi uhodli) použití MVC.“ [30]

Ke svému správnému chodu vyžaduje framework PHP verze minimálně 5.5.9, zapnutou OpenSSL PHP extension, PDO PHP Extension, Mbstring Extension a Tokenizer Extension.

Hlavními přednostmi frameworku Laravel jsou:

- Jednoduchost a snadná pochopitelnost pro začátečníky
- Zabudované základní bezpečnostní prvky
- Podpora odesílání emailů
- Snadná implementace AJAX požadavků
- Jednoduchá instalace
- Podpora více druhů databází
- Podpora vícejazyčných webů

Nyní představím framework z praktického hlediska. U jednotlivých příkladů je postupně ukázáno, jak vytvářet pohledy, jak je vyvolat a také jak přidat další funkce do frameworku. [29] [31]

### **3.2.1 TVORBA VÍCEJAZYČNÝCH WEBŮ - UKÁZKA**

Na tomto příkladu je vidět, jak jednoduchá může být tvorba vícejazyčného webu pomocí tohoto frameworku. Ve složce „/resources/lang“ je třeba nejdříve vytvořit pro každý jazyk vlastní složku. V tomto příkladu si ukážeme, jak vytvořit úvodní přivítání v angličtině a češtině. Vytvoříme tedy složky s názvy „cz“ a „en“. Struktura našeho adresáře bude podobná obrázku dole.

↑Jméno	Příp	Velikost	Datum	Atribut
↑.		<DIR>	17.06.2016 03:02	----
↑[cz]		<DIR>	17.06.2016 03:02	----
↑[en]		<DIR>	17.06.2016 03:02	----

Obrázek 2 – Laravel – ukázka struktury složky

V každé ze složek vytvoříme soubor s názvem „**lang.php**“. Tento soubor ve složce „/resources/lang/cz“ bude mít následující obsah:

```
<?php
return [
    'vitejte' => 'Vítejte na tomto webu'
];
?>
```

Soubor „**lang.php**“ ve složce „/resources/lang/en“ bude vypadat takto:

```
<?php
return [
    'vitejte' => 'Welcome on this website'
];
?>
```

Poté je třeba přidat tyto řádky kódu do libovolného existujícího Controlleru nebo nově vytvořeného. Nový Controller s názvem „Lokalizace“ lze vytvořit pomocí příkazu „php artisan make:controller Lokalizace --plain“.

```
public function index(Request $request,$locale){
    app()->setLocale($locale);
    echo trans('lang.vitejte');
}
```

Dále musíme vytvořit cestu. Do souboru „**routes.php**“ ve složce „/app/http“ přidáme tento kód:

```
Route::get('ukazkalokalizace/{locale}','Lokalizace@index');
```

Nyní je vše připravené pro závěrečnou ukázkou. Po načtení URL „/lokalizace/cz“ dostaneme tento výsledek.

Vítejte na tomto webu

Obrázek 3 – Laraval – ukázka výstupu v češtině

Naopak při načtení URL „/lokalizace/en“ dostaneme následující výsledek

Welcome on this website

Obrázek 4 – Laravel – ukázka výstupu v angličtině

### 3.2.2 FORMULÁŘE - UKÁZKA

Další příklad ukazuje, jak jednoduché je vytvořit obyčejný formulář. Pro maximální usnadnění tvorby formulářů je třeba do frameworku přidat HTML package pomocí příkazu v příkazovém řádku: „composer require illuminate/html“. Do souboru „**app.php**“ ve složce „/config“ je následně nutné přidat do pole „providers“ tuto hodnotu:

```
Illuminate\HTML\HTMLServiceProvider::class
```

Do pole „aliases“ ještě musíme přidat tyto dvě hodnoty:

```
'Form' => Illuminate\HTML\FormFacade::class,  
'Html' => Illuminate\HTML\HtmlFacade::class
```

Nyní je již vše připraveno pro tvorbu samotného formuláře. Ve složce „/resources/views/“ vytvoříme nový pohled. Soubor s názvem „**form.php**“ bude mít následující obsah:

```
<?php  
    echo Form::open(array('url' => 'libovolne/url'));  
    echo Form::text('username', 'Uživatelské jméno');  
    echo '<br/>';  
  
    echo Form::password('password');  
    echo '<br/>';  
  
    echo Form::checkbox('checkbox', 'true');  
    echo '<br/>';  
  
    echo Form::radio('name', 'hodnotajedna');  
    echo '<br/>';  
  
    echo Form::radio('name', 'hodnotadva');  
    echo '<br/>';  
  
    echo Form::file('image');  
    echo '<br/>';  
  
    echo Form::select('size', array('L' => 'Large', 'S' => 'Small'));  
    echo '<br/>';  
        echo Form::submit('Odeslat');  
    echo Form::close();  
?>
```

Pro zobrazení výsledku stačí již jen pohled přidat do cest (/app/http/routes.php). Přidání pohledu provedeme jednoduše:

```
Route::get('/form', function() {  
    return view('form');  
});
```

Po načtení URL „/form“ dostaneme následující výsledek:

Obrázek 5 – Laravel – ukázka základních formulářových prvků

### 3.2.3 AJAX REQUEST - UKÁZKA

Na následujícím příkladu bude vidět, jak vytvořit v tomto frameworku AJAX požadavek.

Nejprve zdrojový kód souboru s pohledem. Soubor ve složce „/resources/views“ s libovolným názvem, například „ajax.php“ má tento obsah:

```
<script src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>

<script>
    function getMessage () {
        $.ajax({
            type: 'POST',
            url: '/ajaxpost',
            data: '_token = <?php echo csrf_token() ?>',
            success: function (data) {
                $("#zprava").html (data.zprava);
            }
        });
    }
</script>
</head>

<body>
    <div id = 'zprava'>Původní zpráva.</div>
    <?php
        echo Form::button('Vyvolat AJAX request', ['onClick'=>'getMessage()']);
    ?>
</body>
```

Již nebudu rozebírat jak vytvořit Controller, protože to je součástí předchozího příkladu.

Controller s libovolným názvem bude mít tento obsah:

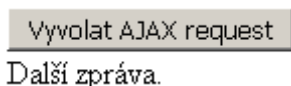
```
class AjaxController extends Controller {
    public function index() {
        $msg = "Další zpráva.";
        return response()->json(array('zprava'=> $msg), 200);
    }
}
```

Dále stačí zpřístupnit pohled. Například takto:

```
Route::get('ajax', function() {
    return view('ajax');
});
Route::post('/ajaxpost', 'AjaxController@index');
```



A poté si již po zadání požadované URL můžeme prohlédnout výsledek [32]



Obrázek 6 – výsledek Ajax requestu

### 3.3 JAVASCRIPTOVÉ FRAMEWORKY

V této podkapitole jsou nejprve zveřejněna hodnotící a porovnávací kritéria shodná s přehledem PHP frameworků. Následně jsou rozebrána kritéria dodatečná. Všechny zmíněné parametry jsou pak použity pro vytvoření uceleného přehledu Javascriptových frameworků. Tento přehled obsahuje navíc kromě základních tabulek i výpis předností vybraných frameworků. Tabulky jsou následně okomentovány.

Většina z vybraných Javascriptových frameworků je velice specifická a slouží pro jiný účel. Zobrazit všechny možné funkce pomocí tabulky by bylo příliš nepřehledné. Z toho důvodu jsou zde pouze zmíněny přednosti vybraných Javascriptových frameworků jako doplnění základního tabulkového přehledu. [33] [9]

#### 3.3.1 DOKUMENTACE A UŽIVATELSKÁ KOMUNITA

Také u Javascriptových jsem zjišťoval popularitu a velikost komunity. Pro připomenutí se jedná o sloupce *StackOverflow*, *Google* a *Youtube*, které jsou blíže popsány v předchozím přehledu PHP frameworků. a stejně jako u PHP frameworků je i zde hodnocena kvalita dokumentace. Jedná se o sloupec *Dokumentace*, který je opět podrobněji popsán v předchozím přehledu. Shodné s předchozím přehledem jsou i sloupce *Licence a Homepage*.

#### 3.3.2 KOMPATIBILITA S PROHLÍŽEČI

U Javascriptových frameworků je porovnávána kromě ostatních parametrů také kompatibilita s jednotlivými webovými prohlížeči. Sloupec *IE* označuje Internet Explorer. Další sloupce už se jmenují stejně jako konkrétní webové prohlížeče. Uvedená hodnota znamená vždy nejnížší verzi kompatibilního prohlížeče.

### 3.3.3 PŘEHLED JAVASCRIPTOVÝCH FRAMEWORKŮ

<u>Framework</u>	<u>IE</u>	<u>Firefox</u>	<u>Safari</u>	<u>Opera</u>	<u>Chrome</u>	<u>Licence</u>
AngularJS	8	4	5	11	30	MIT
Backbone.js	6	nezjištěno	nezjištěno	nezjištěno	nezjištěno	MIT
Ember.js	6	3	4	10.6	14	MIT
Google Web Toolkit	8	1	5	9	1	Apache License v2.0
jQWidgets	7	2	3	9	1	CC / Komerční
React.js	11	nezjištěno	7	nezjištěno	nezjištěno	New BSD
Webix	8	3	4	9	1	GPL / Komerční

Tabulka 4 - Javascriptové frameworky část 1

<u>Framework</u>	<u>StackOverflow</u>	<u>Google</u>	<u>Youtube</u>	<u>Dokumentace</u>	<u>Homepage</u>
AngularJS	178 669	24 200 000	301 000	***	angularjs.org
Backbone.js	19 773	11 300 000	36 700	***	backbonejs.org
Ember.js	19 033	492 000	34 800	**	emberjs.com
Google Web Toolkit	19 465	3 910 000	35 000	***	www.gwtproject.org
jQWidgets	176	111 000	131	*	www.jqwidgets.com
React.js	16 671	24 100 000	92 900	*	facebook.github.io/react
Webix	109	956 000	1 080	***	webix.com

Tabulka 5 - Javascriptové frameworky část 2

### 3.3.4 ZÁVĚRY

Z hlediska kompatibility s prohlížeči vychází podle zjištěných dat nejlépe frameworky *Ember.js* a *Webix*. Celkově na tom však není špatně žádný framework, kromě *React.js*, který plně podporuje Internet Explorer až od verze 11. Prosazuje filosofii podpory pouze nejnovějších prohlížečů a ostatní mají smůlu, což mi nepřijde jako nejmoudřejší krok. Přesto se jistě jedná o jeden z nejpoužívanějších frameworků, vzhledem k tomu, že patří pod Facebook. Od použití tohoto frameworku odradí nepřilíš povedená dokumentace, která obsahuje pouze nejzákladnější popis jednotlivých témat a podle mého názoru málo příkladů použití. Špatně je na tom s dokumentací i sada nástrojů *jQWidgets*. Tato dokumentace působí velice nepřehledným dojmem a navíc postrádá i důležitou funkci vyhledávání. Dokumentace ostatních frameworků jsou na velmi dobré úrovni. Vyzdvihnout musím především dokumentace frameworků *AngularJS* a *Webix*, které působí velice přehledně a přívětivě. Na webu nástroje *Webix* dokonce najdeme funkční interaktivní demo konzoli, ve které lze bez složitého nastavování vyzkoušet jednotlivé funkce. Všechny zkoumané frameworky jsou dostupné pod jednou z licencí zdarma. Za některé pokročilejší funkce *Webix* a *jQWidgets* je však nutné zaplatit. U obou frameworků získáme zakoupením komerční licence také podporu přímo od autorů těchto „pomocníků“. Z předností vybraných frameworků chci zmínit snadnou implementaci *jQuery* u frameworku

Backbone.js. Dále skvěle zpracované testování a Ember CLI<sup>3</sup> u frameworku *Ember.js*. Velký dojem na mě udělal i framework *Webix* a to z několika důvodů. Podporuje systém „Drag-and drop“ a zahrnuje nadstandardní množství předpřipravených řešení. Obsahuje hotový formulář pro upload souborů, kalendář, různé druhy grafů, zobrazení struktury webu pomocí stromu a v neposlední řadě také nástroj pro práci s emaily, díky kterému lze snadno vytvořit online emailového klienta. Tento framework navíc podporuje pokročilou práci s daty, jejich filtrování a zobrazení pomocí přehledných tabulek. Z hlediska relativního hodnocení popularity vychází jednoznačně nejlépe *AngularJS*. Spousta zodpovězených otázek na StackOverflow a velké množství výukových videí na Youtube z něj činí v tomto srovnání zdaleka nejoblíbenější frameworků. Nyní blíže představím právě zmíněný framework. [34] [35] [36] [37] [38] [39] [40] [24]

### 3.4 ANGULARJS

AngularJS je Javascriptový framework, který zavádí do HTML nové atributy, které zrychlují a usnadňují kódování. Nejde však jen o redukci času stráveného kódováním, ale také o nové funkce, které s sebou tyto atributy přináší. AngularJS je open-source framework vytvářený Googlem a programátorskou komunitou. První verze byla vydána již v roce 2009. V současné době je k dispozici i nová verze „AngularJS 2“. Velkou výhodou tohoto frameworku je tzv. dvoucestná synchronizace dat. Uživatel například vyplní určitou položku a to se hned projeví na vzhledu či chování aplikace. Tato vlastnost není u Javascriptových frameworků žádnou samozřejmostí. U ostatních frameworků se toto většinou musí řešit mnohem složitěji.

Hlavními přednostmi tohoto frameworku jsou:

- Práce s CSS<sup>4</sup>
- Práce s daty
- Práce s formuláři
- Animace
- Filtry a řazení dat
- Testování
- Šablony

---

<sup>3</sup> CLI – Command Line Interface

<sup>4</sup> CSS – Cascading Style Sheets

### 3.4.1 NOVÉ ATRIBUTY

Nové atributy jsou tvůrci frameworku označovány jako direktivy. V dalším textu však bude pro tento význam použito známější slovo atribut. Prvním atributem je atribut „ng-app“. Slouží pouze k inicializaci AngularJS aplikace. Dalším atributem je atribut „ng-data“. Zde se inicializují data aplikace. Atribut s názvem „ng-model“ provádí napojení HTML řadiče na data aplikace (například textového pole na proměnnou). Další atribut „ng-repeat“ slouží k opakování HTML elementu, ke kterému je přiřazen. Framework nám navíc umožňuje vytvářet a implementovat atributy vlastní.

### 3.4.2 ŘADIČE, OBJEKTY, FILTRY A SLUŽBY

V AngularJS existuje dále atribut „ng-controller“, který definuje řadič patřící ke konkrétní aplikaci. Jedná se o Javascriptový objekt a není podmínkou při tvorbě skriptu v tomto frameworku. Pokud však chceme vytvářet něco složitějšího, tak se bez řadiče těžko obejdeme. Pro propojení mezi řadičem a pohledem se používá argument „scope“. Jedná se o Javascriptový objekt, který je dostupný jak pro řadič, tak pro pohled. Pohled je v tomto případě HTML, tedy to, co vidí uživatel. Tento framework podporuje také filtrování. Výpis proměnné „jmeno“ velkými písmeny lze snadno učinit následovně: „{{ jmeno | uppercase }}“. Filtry mohou být, jak lze vidět, přidány do výrazu pomocí znaku „trubky“. Jeden z implementovaných filtrů dokáže data dokonce i řadit. K dispozici je možnost vytvářet filtry vlastní. Dalším pomocníkem nástroje AngularJS jsou tzv. služby neboli funkce či objekty určené pro danou aplikaci. Tento framework disponuje množstvím zabudovaných služeb. Pokud nám však nestačí, je samozřejmě možnost vytvořit si vlastní. [41] [42] [43] [44]

### 3.4.3 DVOUCESTNÁ SYNCHRONIZACE DAT - UKÁZKA

Na následujícím příkladu lze vidět, jak snadno lze pracovat se vstupem přijatým od uživatele. Tato ukázka obsahuje zdrojový kód pouze jednoho souboru. Jeho jednoduchost a pochopitelnost na první pohled odhaluje, jak snadná je práce s tímto frameworkem. Soubor „**index.html**“:

```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
</script>
<body>
<div ng-app="">
<p>Jméno : <input type="text" ng-model="name"></p>
<h1>Ahoj {{name}}</h1>
</div>
</body>
```

```
</html>
```

Cokoliv nyní napíšeme do textového pole, tak se okamžitě zobrazí v nadpisu začínajícím „Ahoj“.

#### 3.4.4 VÝRAZY - UKÁZKA

Tento framework zvládá dokonce i vyhodnocování výrazů. Pokud tedy máme zdrojový kód, ve kterém místo proměnné uvedeme výraz, AngularJS tento výraz vyhodnotí a vrátí uživateli jeho výsledek. Soubor „**index.html**“:

```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
</script>
<body>
<div ng-app>
<p>Výraz: {{ 4 * 4 }}</p>
</div>
</body>
</html>
```

Po načtení tohoto souboru v libovolném prohlížeči dostaneme jako výstup text „Výraz: 16“, tedy výsledek výrazu „4\*4“.

#### 3.4.5 TABULKA - UKÁZKA

Velkou předností tohoto frameworku je velmi jednoduchá a rychlá práce s daty. Vše lze snadno ukázat na jednom souboru. Scope bude obsahovat data ve formátu JSON<sup>5</sup>, která chceme zobrazit v tabulce. Zdrojový kód souboru „**index.html**“ bude vypadat takto:

```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
</script>
<body>
<div ng-app="aplikace" ng-controller="kontrolor">
  <table>
    <tr ng-repeat="x in zaznamy"><td>{{ x.jmeno }}</td><td>{{ x.mesto
      }}</td></tr>
  </table>
</div>
<script>
var app = angular.module('aplikace', []);

app.controller('kontrolor', function($scope) {
  $scope.zaznamy = [
    {"jmeno": "Pepa z depa", "mesto": "Psojedy"},
    {"jmeno": "Máňa obecná", "mesto": "Rožumberok"},
    {"jmeno": "Anna", "mesto": "Pelhřimov"} ];
});
</script>
</body>
```

<sup>5</sup> JSON - JavaScript Object Notation

```
</html>
```

Tento jednoduchý kód zobrazí po načtení souboru prohlížečem tabulku, která bude obsahovat dvojice jméno a město vytvořené ze zdrojových JSON dat. Klíčový je zde atribut „ng-repeat“, který opakuje kód pro všechny řádky tabulky.

### 3.4.6 FORMULÁŘE - UKÁZKA

Nyní se pokusím na příkladu ilustrovat, jak velké usnadnění práce s formuláři tento framework přináší. Mějme soubor s názvem „**index.html**“ s tímto obsahem:

```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js">
</script>
<body ng-app="">
<form>
  Vyberte jeden ze sportů:
  <input type="radio" ng-model="promenna" value="fotbal">Fotbal
  <input type="radio" ng-model="promenna" value="hokej">Hokej
  <input type="radio" ng-model="promenna" value="formule">Formule 1
</form>

<div ng-switch="promenna">
  <div ng-switch-when="fotbal">
    <p>Fotbal je nejlepší sport na světě.</p>
  </div>
  <div ng-switch-when="hokej">
    <p>Hokej je druhý nejlepší sport na světě.</p>
  </div>
  <div ng-switch-when="formule">
    <p>Formule 1 je třetí nejlepší sport na světě.</p>
  </div>
</div>
</body>
</html>
```

Atribut „ng-switch-when“ zařídí, že se v jednu chvíli zobrazí pouze text, který náleží k právě aktivnímu „Radio Buttonu“. Ostatní zůstane skryto. Je vidět, že pomocí AngularJS lze tuto funkcionalitu vyřešit velice rychle a efektivně. Stejně je to s většinou dalších podobných úkonů, které souvisí s formuláři. [45]

## 4 PŘEHLED KNIHOVEN

Knihovny jsem se rozhodl rozdělit do kategorií podle způsobu jejich užití. Každá z knihoven je blíže představena textem či výstižným obrázkem. Díky tomu výčet usnadňuje výběr knihovny při řešení konkrétního problému. U jednotlivých knihoven je uvedeno o jaký typ knihovny z hlediska použitého programovacího jazyka se jedná. U knihoven jsem hodnotil čistě subjektivně dokumentaci a jednoduchost užití. V obou případech se jedná se o slovní hodnocení. Jednoduchost užití znamená, jak dlouho podle mého názoru trvá doba od seznámení se programátora s knihovnou k výslednému napsání potřebného kódu. Tam, kde jsem dokázal ihned psát kód, jsem hodnotil jednoduchost užití jako triviální. Čím déle mi trvalo pochopení konkrétní knihovny a programování, tím více jsem hodnocení posouval až ke klasifikaci jednoduchosti užití jako „složitější“. Knihovny, které jsem nedokázal zprovoznit, do výběru zařazeny vůbec nebyly. U každé kategorie jsou uvedeny ukázkové zdrojové kódy vybraných knihoven pro demonstraci jejich užití. Při výběru ukázkových zdrojových kódů jsem se většinou snažil zvolit ta nejrychlejší a nejjednodušší řešení. Ukázky jsou převzaté z oficiálních stránek nebo dokumentací. Následně jsou upraveny pro potřeby této práce.

Tyto zdroje slouží jako hlavní podklad pro přehled knihoven: [46] [47] [48] [49] [50] [51]

### 4.1 GRAFIKA – OBRÁZKY A GALERIE

Tento výčet obsahuje knihovny, které se týkají grafiky. Většinou jde o knihovny určené přímo k práci s obrázky či ke tvorbě galerie.

#### 4.1.1 PHP THUMB

PHP knihovna - <http://phpthumb.sourceforge.net/>

Knihovna sloužící k vytváření tzv. thumbnailů neboli obrázkových náhledů.

Dokumentace: chybí, pouze náhledy zdrojových kódů

Jednoduchost užití: složitější

#### 4.1.2 PHP TEXT TO IMAGE

PHP knihovna - <https://www.daftlogic.com/projects-text-to-image.htm>

Hlavní přínos této knihovny spočívá ve ztížení kopírování citlivých informací z webových stránek, například emailů. Knihovna k dosažení cíle používá metodu přetvoření obyčejného textu v obrázek.

Dokumentace:       stručná, ale jasná

Jednoduchost užití: jednoduchá [52]

#### **4.1.3 ZEBRA IMAGE**

PHP knihovna - <http://stefangabos.ro/php-libraries/zebra-image/>

Jedná se o knihovnu pracující s obrázky. Dokáže je otáčet, překlápět, ořezávat, zmenšovat či zvětšovat. Kromě toho lze používat i filtry. Ke své správné funkci vyžaduje v PHP konfiguraci zapnuté GD knihovny.

Dokumentace:       podrobná a přehledná

Jednoduchost užití: průměrná [53]

#### **4.1.4 IMAGINE**

PHP knihovna - <http://imagine.readthedocs.io/en/latest/>

Další knihovna, která slouží k úpravě obrázků. Kromě manipulace s nimi dokáže i vykreslovat tvary jako čtyřúhelník, elipsa a další. Samozřejmostí je aplikace filtrů.

Dokumentace:       podrobná a velice přehledná

Jednoduchost užití: triviální [54]

#### **4.1.5 SNAPPY**

PHP knihovna - <https://github.com/Knplabs/snappy>

Knihovna, která dokáže vytvořit ze zadané URL náhled stránky v obrázkovém formátu či formátu PDF. Ke správnému chodu vyžaduje minimálně PHP 5.

Dokumentace:       stručná, ale postačující

Jednoduchost užití: triviální [55]

#### **4.1.6 WIDEIMAGE**

PHP knihovna - <http://wideimage.sourceforge.net/>



Tato knihovna slouží k manipulaci s obrázky. Jedná se o změnu velikosti, překlápění, ořezávání, aplikaci filtrů. Knihovna vyžaduje ke správné funkci minimálně PHP verze 5.2 a zapnutou knihovnu GD2.

Dokumentace:           velice přehledná s příklady

Jednoduchost užití:   triviální [56]

#### 4.1.7 UBERGALLERY

PHP knihovna/jQuery plugin - <http://www.ubergallery.net/>

Tato knihovna slouží k rychlému a snadnému vytvoření obrázkové galerie. Ke své správné funkci vyžaduje minimálně PHP verze 5.2 a zapnutou knihovnu GD.

Dokumentace:           velice stručná

Jednoduchost užití:   jednoduchá [57]

#### 4.1.8 COPPERMINE

PHP knihovna - <http://coppermine-gallery.net/>

Knihovna, pomocí které lze vytvářet složitější galerie s pokročilejšími funkcemi. Pro svoji správnou funkci vyžaduje PHP verze minimálně 4.2.0 se zapnutou GD knihovnou i MySQL databázi. Obsahuje nespočet funkcí. Mezi nejzajímavější patří vytváření náhledů, vyhledávání, funkce náhodného obrázku, slideshow, podpora tagů a popisků, informací EXIF, otáčení obrázku, třídění do alb, mnohojazyčná lokalizace včetně češtiny, podpora komentářů a upload více obrázků najednou.

Dokumentace:           přehledná

Jednoduchost užití:   jednoduchá, ale instalace zabere delší dobu [58]

#### 4.1.9 ILUSTRACE ŘEŠENÍ PROBLÉMŮ MANIPULACE S OBRÁZKY

Osobně bych k manipulaci s obrázky využil knihovny **Imagine**. Na následujícím příkladu převzatém z dokumentace knihovny lze vidět, jak jednoduše se dá změnit velikost obrázku.

```
<?php
use Imagine\Image\Box;
use Imagine\Image\Point;

$image->resize(new Box(15, 25))
?>
```

Další příklad ukazuje jednoduché přidání tzv. watermarku do pravého dolního rohu obrázku použitím stejné knihovny. Watermark česky vodoznak se používá pro označení obrázku. Většinou jde o nějaké logo nebo text obsahující URL původní stránky odkud obrázek pochází. Přidává se k obrázku zejména za účelem ochrany proti kopírování.

```
<?php
$watermark = $imagine->open('/my/watermark.png');
$image      = $imagine->open('/path/to/image.jpg');
$size      = $image->getSize();
$wSize     = $watermark->getSize();

$bottomRight = new Imagine\Image\Point($size->getWidth() - $wSize->getWidth(),
                                         $size->getHeight() - $wSize->getHeight());

$image->paste($watermark, $bottomRight);
?>
```

Pro snadné vytváření náhledů obrázků doporučuji použít knihovnu **WideImage**. Na příkladu je vidět jak jednoduché je vytvoření náhledu pomocí této knihovny. Jedná se o kód dostupný z oficiálních stránek knihovny.

```
<?php
include 'WideImage/WideImage.php';
WideImage::load('big.png')->resize(50, 30)->saveToFile('small.jpg');
?>
```

#### 4.1.10 UKÁZKA TVORBY GALERIE POMOCÍ UBERGALLERY

K rychlému a snadnému vytvoření galerie z existujících obrázků se skvěle hodí knihovna **UberGallery**. Následující kód ukazuje jednoduchost řešení galerie pomocí této knihovny. Pro správnou funkci je nutné nastavit konstantu „THEMEPATH“ a adresář s obrázky (při volání metody „readImageDirectory“)

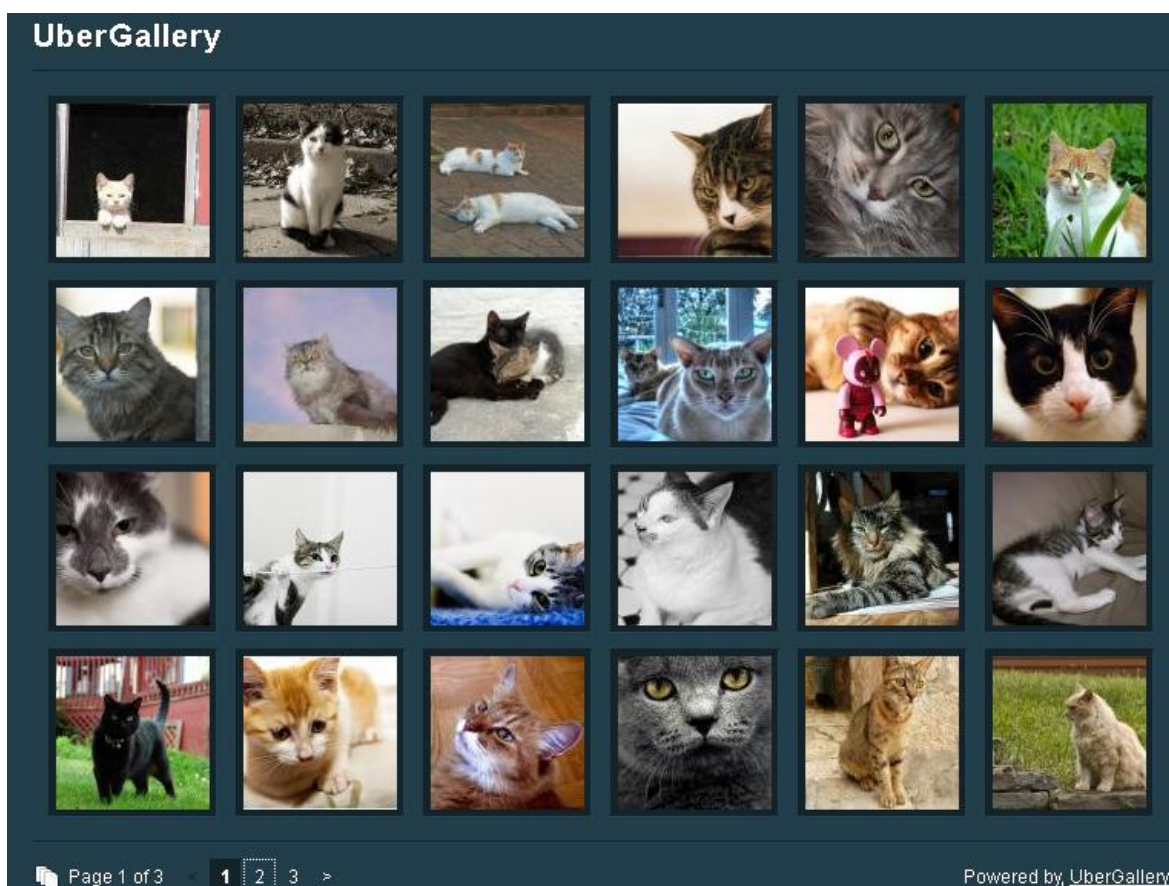
```
<?php
include('resources/UberGallery.php');
$gallery = new UberGallery();
$galleryArray = $gallery->readImageDirectory('gallery-images');

if (!defined('THEMEPATH')) {
    define('THEMEPATH', $gallery->getThemePath());
}

$themeIndex = $gallery->getThemePath(true) . '/index.php';

if (file_exists($themeIndex)) {
    include($themeIndex);
} else {
    die('ERROR: Failed to initialize theme');
}
?>
```

Na obrázku dole je pak vidět výsledek. Jedná se o převzaté řešení z oficiálních stránek.



Obrázek 7 – Galerie UberGallery, zdroj: ubergallery.net

## 4.2 GRAFIKA - GRAFY A MAPY

Jedná se o knihovny, které usnadňují tvorbu grafů a map.

### 4.2.1 FUSIONCHARTS

Javascriptová knihovna - <http://www.fusioncharts.com>

Jak již název napovídá, tato knihovna je určena pro práci s grafy. Dostupná je zdarma pro nekomerční použití. V případě vývoje komerční aplikace je třeba zakoupit licenci, která není vůbec levná. Nejlevnější přijde na \$199. Knihovna dokáže kromě grafů vytvářet i mapy. Podporuje filtry a řazení dat za běhu.

Dokumentace: dostatečně zpracovaná

Jednoduchost užití: složitější [59]

### 4.2.2 EMBER CHARTS

Javascriptová knihovna - <http://opensource.addepar.com/ember-charts>

Další knihovna, která slouží k vytváření grafů. Knihovna je sestavena pomocí frameworků Ember.JS a d3.js. Podporuje pouze základní typy diagramů. Velkou výhodou u této knihovny je možnost prohlédnout si výsledný vzhled jednotlivých grafů na domovské stránce. Vše pomocí funkčního dema zobrazujícího grafy v reálném čase podle nastavovaných hodnot.

Dokumentace:           stručná a přehledná

Jednoduchost užití:   jednoduchá [60]

### **4.2.3 CHART.JS**

Javascriptová knihovna - <http://www.chartjs.org>

Tato Javascriptová knihovna využívá pro tvorbu grafů HTML 5. Podporuje mimo jiné i animované grafy.

Dokumentace:           podrobná a přehledná

Jednoduchost užití:   jednoduchá, intuitivní [61]

### **4.2.4 ARISTOCHART**

Javascriptová knihovna - <http://adriancooney.ie/aristochart>

Další knihovna využívající HTML 5 k vykreslování grafů. Autor ji charakterizuje jako velice přizpůsobitelnou knihovnu. K dispozici je na výběr množství zobrazovacích šablon, které stále roste díky přispívající komunitě. Zobrazit však umí pouze základní druhy grafů.

Dokumentace:           dostatečně podrobná, ale méně přehledná

Jednoduchost užití:   složitější [62]

### **4.2.5 BONSAIJS**

Javascriptová knihovna - <http://bonsaijs.org/>

Knihovna vytvářející opravdu působivé grafy. K vykreslení používá vlastní SVG<sup>6</sup> renderer. Na domovské stránce je k nalezení několik funkčních příkladů. Jedná se o velice komplexní knihovnu obsahující nadstandardní množství funkcí týkajících práce s grafikou. Knihovna je šířena pod licencí MIT.

Dokumentace:           skvěle zpracovaná, přehledná

---

<sup>6</sup> SVG - Scalable Vector Graphics

Jednoduchost užití: průměrná [63]

#### 4.2.6 MORRIS.JS

Javascriptová knihovna - <http://morrisjs.github.io/morris.js/>

Knihovna dokáže vytvářet jednoduché interaktivní grafy. K vykreslování používá kombinace knihoven jQuery a Raphaël. Je šířena pod licencí BSD. Jedná se však o menší knihovnu, z toho důvodu zvládá pouze základní typy grafů.

Dokumentace: vzorově zpracovaná a přehledná

Jednoduchost užití: triviální [64]

#### 4.2.7 AMCHARTS

Javascriptová knihovna - <https://www.amcharts.com/>

Tato knihovna obsahuje opravdu velké množství typů grafů. Samozřejmostí je podpora interaktivních grafů a navíc dokáže vytvářet i interaktivní mapy. Jedná se o jednu z nejkompaktnějších knihoven tohoto přehledu. Tato knihovna existuje také jako plugin Wordpressu. Knihovnu je možno používat zdarma, ale pod podmínkou zobrazení linku na původní stránku v levém horním rohu všech grafů. Tento odkaz lze odstranit po zakoupení komerční licence. Na stránkách knihovny lze najít v sekci „Demos“ ukázky všech podporovaných grafů.

Dokumentace: skvěle zpracovaná s možností vyhledávání

Jednoduchost užití: jednoduchá práce s objekty [65]

#### 4.2.8 AMCHARTS - UKÁZKOVÝ GRAF

U tohoto tématu mě hodně zaujaly dvě knihovny. První z nich je Amcharts. Na jednoduchém příkladu je vidět, jak působivého výsledku lze dosáhnout použitím této knihovny. Pro účely ukázky nám na zobrazení grafu bude stačit jediný soubor „**index.html**“ s tímto zdrojovým kódem:

```
<html>
<head>
<script src="https://www.amcharts.com/lib/3/amcharts.js"></script>
<script src="https://www.amcharts.com/lib/3/pie.js"></script>
<script src="https://www.amcharts.com/lib/3/themes/light.js"></script>
<style>
#chartdiv {
width      : 100%;
height     : 435px;
font-size  : 11px;
}
```

```

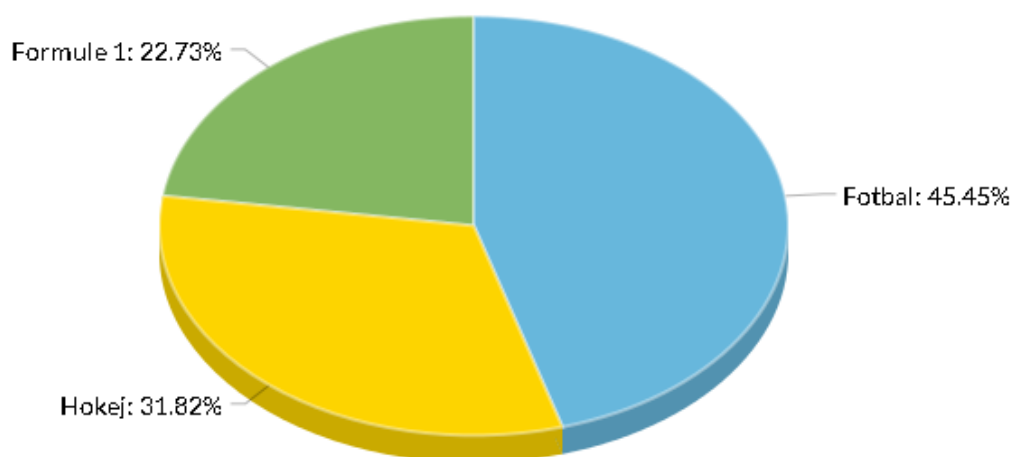
</style>
</head>
<body>
<div id="chartdiv"></div>
<script type="text/javascript">
var chart = AmCharts.makeChart( "chartdiv", {
  "type": "pie",
  "theme": "light",
  "dataProvider": [ {
    "country": "Fotbal",
    "value": 100
  }, {
    "country": "Hokej",
    "value": 70
  }, {
    "country": "Formule 1",
    "value": 50
  } ],
  "valueField": "value",
  "titleField": "country",
  "outlineAlpha": 0.4,
  "depth3D": 15,
  "balloonText": "[[title]]<br><span style='font-size:14px'><b>[[value]]</b></span> ([[percents]]%)</span>",
  "angle": 30,
  "export": {
    "enabled": true
  }
} );
jQuery( '.chart-input' ).off().on( 'input change', function() {
  var property = jQuery( this ).data( 'property' );
  var target = chart;
  var value = Number( this.value );
  chart.startDuration = 0;

  if ( property == 'innerRadius' ) {
    value += "%";
  }

  target[ property ] = value;
  chart.validateNow();
} );
</script>
</body>
</html>

```

Pokud otevřeme tento soubor v libovolném prohlížeči podporujícím Javascript, dostaneme výsledek podobný obrázku dole.



Obrázek 8 – AmCharts – ukázkový graf

#### 4.2.9 AMCHARTS – UKÁZKOVÁ MAPA

Na tomto příkladu lze vidět, jak zobrazit pomocí této knihovny na předpřipravené mapě světa animaci letu z Paříže do Toronta. Opět stačí jediný soubor „`index.html`“ s tímto obsahem:

```
<html>
<head>
<script src="https://www.amcharts.com/lib/3/amcharts.js"></script>
<script src="https://www.amcharts.com/lib/3/pie.js"></script>
<script src="https://www.amcharts.com/lib/3/themes/light.js"></script>
<style>
#chartdiv {
  width      : 100%;
  height     : 500px;
  font-size  : 11px;
}
</style>
</head>
<body>
<div id="chartdiv"></div>
<script type="text/javascript">
var targetSVG = "M9,0C4.029,0,0,4.029,0,9s4.029,9,9,9s9-4.029,9-9S13.971,0,9,0z
M9,15.93 c-3.83,0-6.93-3.1-6.93-
6.93S5.17,2.07,9,2.07s6.93,3.1,6.93,6.93S12.83,15.93,9,15.93
M12.5,9c0,1.933-1.567,3.5-3.5,3.5S5.5,10.933,5.5,9S7.067,5.5,9,5.5
S12.5,7.067,12.5,9z";
var planeSVG = "m2,106h28124,30h721-44,-133h35180,132h98c21,0 21,34 0,341-98,0 -
80,134h-35143,-133h-711-24,30h-28115,-47";

window.map = AmCharts.makeChart("chartdiv", {
  type: "map",

  dataProvider: {
    map: "worldLow",
    zoomLevel: 3.5,
    zoomLongitude: -55,
    zoomLatitude: 42,
```

```
    lines: [{
      id: "line1",
      arc: -0.85,
      alpha: 0.3,
      latitudes: [48.8567, 43.8163],
      longitudes: [2.3510, -79.4287]
    }, {
      id: "line2",
      alpha: 0,
      color: "#000000",
      latitudes: [48.8567, 43.8163],
      longitudes: [2.3510, -79.4287]
    }],
    images: [{
      svgPath: targetSVG,
      title: "Paříž",
      latitude: 48.8567,
      longitude: 2.3510
    }, {
      svgPath: targetSVG,
      title: "Toronto",
      latitude: 43.8163,
      longitude: -79.4287
    }, {
      svgPath: planeSVG,
      positionOnLine: 0,
      color: "#000000",
      alpha: 0.1,
      animateAlongLine: true,
      lineId: "line2",
      flipDirection: true,
      loop: true,
      scale: 0.03,
      positionScale: 1.3
    }, {
      svgPath: planeSVG,
      positionOnLine: 0,
      color: "#585869",
      animateAlongLine: true,
      lineId: "line1",
      flipDirection: true,
      loop: true,
      scale: 0.03,
      positionScale: 1.8
    }
  ]
},

areasSettings: {
  unlistedAreasColor: "#8dd9ef"
},

imagesSettings: {
  color: "#585869",
  rollOverColor: "#585869",
  selectedColor: "#585869",
  pauseDuration: 0.2,
  animationDuration: 2.5,
  adjustAnimationSpeed: true
},

linesSettings: {
  color: "#585869",
  alpha: 0.4
},

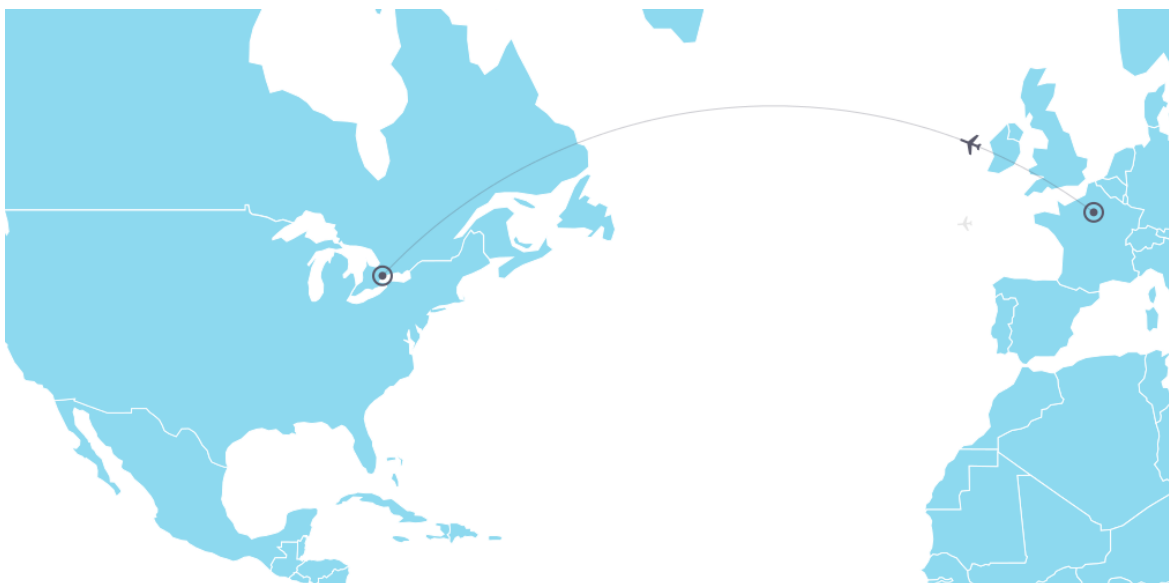
export: {
  enabled: true
}
```



```
});
</script>

</body>
</html>
```

Po otevření souboru v prohlížeči podporujícím Javascript dostaneme výsledek podobný obrázku dole. Samozřejmě i s animací letícího letadla. Na jednotlivé body cesty lze najet kurzorem myši a zjistit o jakou destinaci jde.



Obrázek 9 – Amcharts – ukázková mapa

#### 4.2.10 MORRIS.JS – UKÁZKA GRAFU

Na posledním příkladu v této kategorii chci ukázat jednoduchost tvorby grafu pomocí knihovny Morris.JS. Stačit bude opět jediný soubor „**index.html**“ s tímto obsahem:

```
<!DOCTYPE html>
<html>
<head>
<script src="http://cdnjs.cloudflare.com/ajax/libs/raphael/2.1.0/raphael-
min.js"></script>
<script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
<script src="http://cdn.oesmith.co.uk/morris-0.4.1.min.js"></script>

<meta charset=utf-8 />

</head>
<body>
<div id="graf"></div>
<script>
Morris.Bar({
  element: 'graf',
  data: [
    { y: '2006', a: 100, b: 90 },
    { y: '2007', a: 75, b: 65 },
    { y: '2008', a: 50, b: 40 },
    { y: '2009', a: 75, b: 65 },
    { y: '2010', a: 50, b: 40 },
```

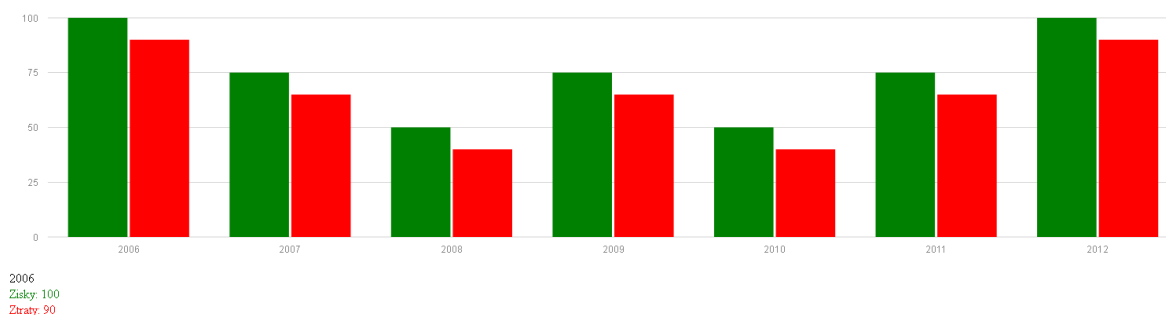
```

    { y: '2011', a: 75, b: 65 },
    { y: '2012', a: 100, b: 90 }
  ],
  xkey: 'y',
  ykeys: ['a', 'b'],
  labels: ['Zisky', 'Ztraty'],
  barColors: ['green', 'red']
});

</script>
</body>
</html>

```

Po načtení souboru v libovolném prohlížeči se zapnutým Javascriptem dostaneme výsledek podobný Obrázek 10. Jedná se o interaktivní graf. Po najetí kurzorem na sloupec, dojde k výpisu konkrétních hodnot pod grafem vlevo. Práce s touto knihovnou je skutečně velice jednoduchá.



Obrázek 10 – Morris.js – ukázkový graf

## 4.3 FORMULÁŘE A TABULKY

Přehled knihoven, které nějak souvisí s formuláři či tabulkami.

### 4.3.1 FANCY FORM

Javascriptová knihovna - <http://fancyjs.com/>

Velice zdařilá knihovna založená na knihovně jQuery. Podporuje tvorbu dobře vypadajících pokročilých formulářů. Nechybí podpora validace dat, šablon a knihovny ReCAPTCHA.

Dokumentace: skvěle zpracovaná a přehledná s příklady

Jednoduchost užití: triviální [66]

### 4.3.2 APPHP DATAGRID

PHP/Javascriptová knihovna - <http://www.apphp.com/php-datagrid>

Knihovna sloužící k tvorbě tabulek. Hlavní doménou této knihovny je však práce s daty, kdy dokáže data v tabulce řadit, filtrovat i stránkovat. Potěší také podpora formátovaného výstupu pro tisk. Zdarma je dostupná ořezaná verze. Jinak se jedná o placenou knihovnu.

Dokumentace: stručná, ale výstižná

Jednoduchost užití: triviální [67]

### 4.3.3 PHP FORM VALIDATION SCRIPT

PHP knihovna - <http://www.html-form-guide.com/php-form/php-form-validation.html>

Jedná se o jednoduchou knihovnu sloužící k validaci formulářových polí.

Dokumentace: stručná a jasná

Jednoduchost užití: triviální [68]

### 4.3.4 EDITABLEGRID

Javascriptová knihovna – <http://www.editablegrid.net/en/>

Jde o open-source knihovnu, pomocí které lze pracovat s daty a zobrazovat je v grafické podobě. Podporuje stránkování, označování, mazání a editaci dat. Dokonce dokáže z dat i vytvořit Flash grafy. Na domovské stránce je k dispozici funkční demo. Výsledné tabulky působí opravdu velice hezky.

Dokumentace: příliš stručná

Jednoduchost užití: složitější [69]

### 4.3.5 DHTMLXGRID

Javascriptová knihovna - [http://docs.dhtmlx.com/grid\\_\\_index.html](http://docs.dhtmlx.com/grid__index.html)

Jedná se o jednu z komponent balíku dhtmlx. Celý balík obsahuje velké množství komponent, které dokáží vytvořit téměř vše, na co si lze vzpomenout. Například formuláře, menu, grafy, seznamy, kalendáře a další. Pro nekomerční použití je dostupná zdarma, jinak je třeba zaplatit. Nutno však podotknout, že celý balík knihoven je velice profesionálně zpracovaný a konkrétně dhtmlxGrid vytváří skvěle vypadající výstupy. Data v tabulce dokáže řadit, stránkovat a umožňuje je uživateli dokonce i přesouvat pomocí jednoduché metody drag-and-drop neboli česky „přetahováním pomocí myši“.

Dokumentace: profesionálně zpracovaná, přehledná

Jednoduchost užití: jednoduchá [70]

#### 4.3.6 FINEUPLOADER

Javascriptová knihovna - <http://fineuploader.com>

Pravděpodobně nejlepší Javascriptová knihovna sloužící jako formulář pro upload souborů. Jedná se o open-source knihovnu. Umožňuje nastavit spoustu parametrů, z nichž nejdůležitější jsou povolené koncovky souborů, maximální velikost souboru a maximální počet souborů. Pro správný chod nejsou potřeba žádné další knihovny ani Flash. Lze uploadovat více souborů najednou a knihovna dokonce umožňuje navazovat ztracené spojení. Po úspěšném uploadu souborů zobrazuje jejich náhledy. Soubory je možné uploadovat i pomocí metody drag-and-drop. Upload souborů lze přerušit a poté navázat. Uploadované soubory se dají také přejmenovat. Na domovské stránce je několik demo příkladů k vyzkoušení. Vzhled formuláře lze jednoduše změnit pomocí dostupných šablon.

Dokumentace: výborně zpracovaná

Jednoduchost užití: jednoduchá [71]

#### 4.3.7 DATATABLES

Javascriptová knihovna - <https://datatables.net>

Jedná se o knihovnu pro práci s daty využívající jQuery. Data dokáže zobrazit pomocí tabulek. Podporuje řazení dat, filtrování, vyhledávání v celé tabulce nebo pouze v jednotlivých sloupcích. Samozřejmostí je podpora stránkování, mazání, editace a přidávání nových záznamů pomocí AJAX požadavků. Knihovna je dostupná zdarma pod licencí MIT. Vzhled tabulek lze snadno přizpůsobovat.

Dokumentace: skvěle zpracovaná a přehledná

Jednoduchost užití: jednoduchá [72]

#### 4.3.8 APHP DATAGRID – UKÁZKA ZOBRAZENÍ DAT

Tato knihovna mě zaujala především svojí jednoduchostí. Na tomto příkladu je vidět, jak jednoduché může být zobrazení dat v přehledné a dobře vypadající tabulce pomocí knihovny ApPHP DataGrid. Na to, že se jedná skutečně jen o pár řádků kódu v PHP, je výsledek opravdu působivý:

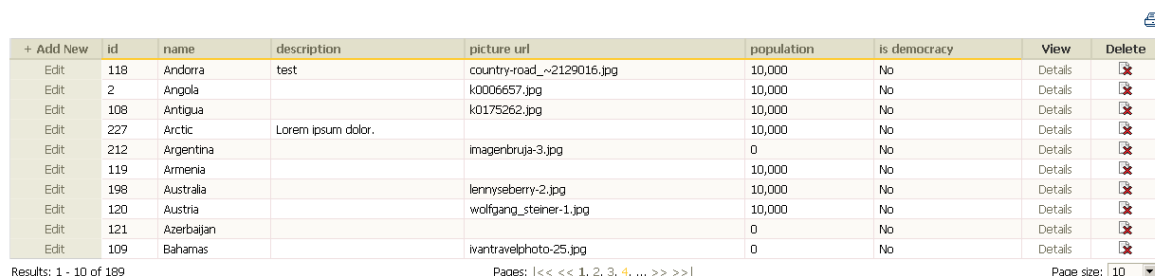
```
$dgrid = new DataGrid();
```

```

$datagrid->DataSource('PDO', 'mysql', $DB_HOST, $DB_NAME, $DB_USER, $DB_PASS,
    'SELECT * FROM demo_countries', array('name'=>'ASC'));
$datagrid->SetTableEdit('demo_countries', 'id');
$datagrid->SetAutoColumnsInViewMode(true);
$datagrid->SetAutoColumnsInEditMode(true);
$datagrid->Bind();

```

Na následujícím obrázku můžeme vidět výsledek.



+ Add New	id	name	description	picture url	population	is democracy	View	Delete
Edit	118	Andorra	test	country-road_~2129016.jpg	10,000	No	Details	
Edit	2	Angola		k0006657.jpg	10,000	No	Details	
Edit	108	Antigua		k0175262.jpg	10,000	No	Details	
Edit	227	Arctic	Lorem ipsum dolor.		10,000	No	Details	
Edit	212	Argentina		imagenbruja-3.jpg	0	No	Details	
Edit	119	Armenia			10,000	No	Details	
Edit	198	Australia		lennyseberry-2.jpg	10,000	No	Details	
Edit	120	Austria		wolfgang_steiner-1.jpg	10,000	No	Details	
Edit	121	Azerbajjan			0	No	Details	
Edit	109	Bahamas		ivantravelphoto-25.jpg	0	No	Details	

Results: 1 - 10 of 189      Pages: | << < 1, 2, 3, 4, ... >> >> |      Page size: 10

Obrázek 11 – ApPHP DataGrid – ukázka tabulky

### 4.3.9 FANCY FORM – UKÁZKA JEDNODUCHÉHO FORMULÁŘE

V tomto příkladu převzatém z oficiálních stránek ukážu, jak lze vytvořit formulář pomocí knihovny Fancy Form. Zdrojový kód je samozřejmě lehce upravený. Po stažení knihovny nám stačí jediný soubor s názvem „index.html“ s následujícím zdrojovým kódem:

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <script src="jquery/jquery-1.11.0.min.js"></script>
  <link rel="stylesheet" href="fancyform/fancyform-min.css">
  <script src="fancyform/fancyform-min.js"></script>
</head>
<body>
  <div id="form"></div>
  <script>
    window.onload = function(){
      FancyForm.vtype({
        type: 'email',
        re: /^(("[??:[^\." ])(?:([^\." ])?(?:[\w\-\!#$%&'**+\./=?\^_`{|}~]))*\1@(\w[\-\w]*\.){1,5}([A-Za-z]){2,6}$)/,
        blankText: 'vyžadováno',
        text: 'Nesprávný email'
      });

      FancyForm.vtype({
        type: 'age',
        fn: function(value){
          return value>20 && value<60;
        },
        text: 'Věk musí být mezi 20 a 60 lety'
      });

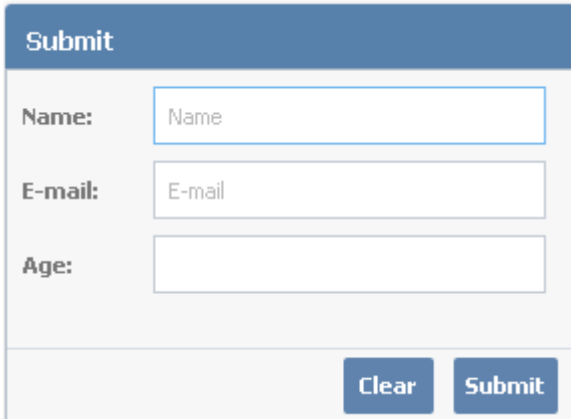
      $('#form').FancyForm({
        title: 'Submit',
        width: 285,
        height: 210,
        inputWidth: 180,
        labelWidth: 55,

```

```
        url: 'submit.php',
        params: {
            param1: 1,
            param2: 'string'
        },
        method: 'POST',

        defaults: {
            type: 'string'
        },
        items: [{
            label: 'Name',
            emptyText: 'Name',
            valid: {
                blank: false,
                blankText: 'Vyžadováno'
            },
            name: 'name'
        }, {
            label: 'E-mail',
            emptyText: 'E-mail',
            valid: {
                type: 'email',
                blank: false
            },
            name: 'email'
        }, {
            type: 'number',
            label: 'Age',
            name: 'age',
            valid: {
                type: 'age'
            }
        }
    ],
    buttons: [{
        text: 'Clear',
        handler: function(){
            this.clear();
        }
    }, {
        text: 'Submit',
        handler: function(){
            this.submit({
                params: {
                    param3: 'Some Values'
                },
                success: function(result, status, xhr){
                    console.log('success');
                    console.log(arguments);
                },
                error: function(xhr, status, error){
                    console.log('error');
                    console.log(arguments);
                }
            });
        }
    }
    ]
    });
};
</script>
</body>
</html>
```

Podmínkou samozřejmě je, že soubor se nachází ve stejné složce jako knihovna. Po načtení stránky v prohlížeči se zapnutým Javascriptem dostaneme formulář podobný obrázku dole. Formulář obsahuje i kontrolu zadání správného formátu emailu a věku.



The image shows a web form with a blue header bar containing the word "Submit". Below the header, there are three input fields. The first is labeled "Name:" and contains the placeholder text "Name". The second is labeled "E-mail:" and contains the placeholder text "E-mail". The third is labeled "Age:" and is empty. At the bottom right of the form, there are two buttons: "Clear" and "Submit".

Obrázek 12 – Fancy Form – jednoduchý formulář

## 4.4 DATABÁZE

Zde se nacházejí knihovny související s databázemi.

### 4.4.1 ADOdb

PHP knihovna - <http://adodb.org>

Knihovna usnadňující práci s databází. Podporuje opravdu velké množství databází včetně nejpoužívanějších – MySQL, PostgreSQL a Oracle. Na oficiálních stránkách autoři zmiňují podporu všech verzí PHP včetně nejnovějšího PHP 7.

Dokumentace: přehledná, roztříděná podle podporovaných databází

Jednoduchost užití: jednoduchá [73]

### 4.4.2 PHP ADVANCED DATABASE CLASS

PHP knihovna - <http://slaout.linux62.org/php/index.html>

Další knihovna sloužící k usnadnění práce s databází. Podporuje však pouze databázi MySQL. Je dostupná zdarma pod licencí LGPL a jedná se o jediný soubor.

Dokumentace: dobře zpracovaná

Jednoduchost užití: triviální [74]

### 4.4.3 SQLCACHE

PHP knihovna - <http://www.phpclasses.org/package/2646-PHP-Cache-database-query-results-in-files-.html>

Jedná se o volně dostupnou knihovnu, která pomůže zlepšit výkon databáze. Provádí to tak, že ukládá výsledky SQL<sup>7</sup> dotazů do souborů a poté rovnou vrací jako odpověď na požadavky přímo vygenerované soubory. Jedná se o tzv. SQL cachování, které má tato knihovna přímo v názvu.

Dokumentace: chybí

Jednoduchost užití: složitější

### 4.4.4 CATCHMYFAME

PHP knihovna - <http://www.catchmyfame.com/2007/07/28/finally-the-simple-pagination-class/>

Tato knihovna slouží ke stránkování výstupu databáze. Umožňuje samozřejmě nastavit počet záznamů na stránku.

Dokumentace: jednoduchá, ale dostačující

Jednoduchost užití: jednoduchá [75]

### 4.4.5 DANPU

PHP knihovna - <https://github.com/gocom/danpu>

Knihovna sloužící ke zjednodušení zálohování databáze. Podporuje pouze databázi MySQL. Dokáže data jak zálohovat, tak obnovit. Knihovna je šířena zdarma pod licencí LGPL.

Dokumentace: stručná, ale výstižná

Jednoduchost užití: triviální [76]

### 4.4.6 DANPU – ZÁLOHA DATABÁZE

Následující krátký příklad ukazuje, jak jednoduchá může být záloha databáze s využitím knihovny Danpu. Stačí k tomu tento kód, kde pozměníme potřebné hodnoty, mezi které patří například přihlašovací jméno, heslo, název databáze:

```
<?php
use Rah\Danpu\Dump;
```

---

<sup>7</sup> SQL - Structured Query Language



```

use Rah\Danpu\Export;

try {
    $dump = new Dump;
    $dump
        ->file('ZALOHA.sql')
        ->dsn('mysql:dbname=DATABASE;host=HOST')
        ->user('UZIVATELSKEJMENO')
        ->pass('HESLO')
        ->tmp('/ADRESA/DOCASNEHO/ADRESARE');

    new Export($dump);
} catch (\Exception $e) {
    echo 'Export se nezdařil: ' . $e->getMessage();
}
?>

```

## 4.5 MANIPULACE S DATY

Jedná se o knihovny, které se týkají práce s daty.

### 4.5.1 JQUERY

Javascriptová knihovna – <http://jquery.com>

„Knihovna jQuery klade důraz na inerakci mezi značkovacím jazykem HTML a programovacím jazykem JavaScript. Tato knihovna je rychlá a přesná, přičemž zjednodušuje tvorbu a správu událostí, animací a spousty dalších komponent, které s sebou přináší nejmodernější technologie. Poprvé ji představil John Resig v roce 2006. Knihovna jQuery se šíří pod licencemi GPL a MIT, což znamená, že ji můžeme volně používat ve svých projektech. Rovněž se můžeme díky těmto licencím zapojit do vývoje knihovny jQuery.“ [77] Existuje několik verzí této knihovny a jedná se o jednu z nejznámějších a nejpoužívanějších Javascriptových knihoven. Mezi největší výhody patří podpora AJAXu. To je technologie, díky které je možné aktualizovat obsah jednotlivých částí webových stránek bez nutnosti znovunačtení. [78] Tuto knihovnu je možné dále rozšířit pomocí velkého množství dostupných pluginů. Vyhledávat tyto pluginy lze na stránkách:

- <http://plugins.jquery.com/>
- <https://www.npmjs.com/browse/keyword/jquery-plugin>

Dokumentace:           podrobná, ale nepříliš přehledná

Jednoduchost užití:   složitější [79]

### 4.5.2 PHP COLLECTION

PHP knihovna - <http://jmsyst.com/libs/php-collection>

Tato knihovna přidá do PHP nový datový typ kolekce. Na oficiálních stránkách se uvádí tyto tři typy kolekcí: "Sequences", "Maps" a "Sets". Poslední typ "Sets" však ještě není plně implementován.

Dokumentace: stručná a přehledná

Jednoduchost užití: triviální [80]

### 4.5.3 SERVICES\_JSON

PHP knihovna - [http://pear.php.net/package/Services\\_JSON/download](http://pear.php.net/package/Services_JSON/download)

Tato knihovna slouží k implementaci formátu JSON i pro PHP starších verzí (normálně je JSON podporován až od PHP verze 5.2.0)

Dokumentace: nepříliš přehledná

Jednoduchost užití: jednoduchá [81]

### 4.5.4 TCPDF

PHP knihovna - <http://www.tcpdf.org>

Tato knihovna slouží ke tvorbě PDF souborů pomocí PHP. Obsahuje velké množství funkcí, díky kterým lze vytvářet i složitější dokumenty. Jedná se o open-source knihovnu.

Dokumentace: dostatečně podrobná, méně přehledná

Jednoduchost užití: jednoduchá [82]

### 4.5.5 PHPDOCUMENTOR

PHP knihovna - <https://phpdoc.org/>

Nástroj sloužící ke snadné tvorbě dokumentace. Podporuje řadu různých výstupů jako je HTML, PDF a dalších.

Dokumentace: podrobná, méně přehledná

Jednoduchost užití: složitější [83]

### 4.5.6 JQUERY – JEDNODUCHÁ UKÁZKA

Následující příklad ukazuje, jak vypadá jQuery v praxi. Jedná se o implementaci jednoduchého AJAX požadavku. Pro provedení potřebujeme dva soubory. Zdrojový kód prvního souboru s názvem například „**index.html**“ vypadá takto:

```
<head>
```

```

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js">
</script>
<script>
$(document).ready(function() {
    $("button").click(function() {
        $("#obsah").load("ajax_ukazka.txt");
    });
});
</script>
</head>
<body>
<div id="obsah"><h2>Původní obsah</h2></div>
<button>Načti externí obsah</button>
</body>

```

Druhý soubor s názvem „**ajax\_ukazka.txt**“ má tento obsah:

```
<h2>Nadpis</h2>Další text
```

Po zobrazení souboru „**index.html**“ v prohlížeči a kliknutí na tlačítko „Načti externí obsah“ se nadpis s textem „Původní obsah“ přepíše na text ze souboru „**ajax\_ukazka.txt**“. Vše ostatní zůstane zachováno (například tlačítko nezmizí). [84]

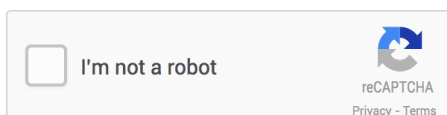
## 4.6 CAPTCHA

“Completely automated public Turing test to tell computers and humans apart“ zkráceně CAPTCHA. Používá se k prevenci proti spamu například v diskusních fórech. Slouží k ověření, zda se dotaz pokouší odeslat skutečně člověk a ne nějaký BOT. Existuje celá řada knihoven, které pomohou tento problém vyřešit pomocí několika řádek kódu. [85]

### 4.6.1 RECAPTCHA

PHP knihovna - <https://www.google.com/recaptcha/>

Tato knihovna je zvláštní tím, že řeší ověření nevšedním způsobem. Jeden ze způsobů ukazuje obrázek dole.



Obrázek 13 – ReCAPTCHA, zdroj: google.com

Dokumentace:           přehledná a dostačující

Jednoduchost užití:   triviální [86]

### 4.6.2 SIMPLECAPTCHA

Javascriptová knihovna - <http://simplecaptcha.sourceforge.net/>



Obrázek 14 – SimpleCaptcha

Dokumentace: stručná, ale jasná

Jednoduchost užití: snadné [87]

#### 4.6.3 SECURIMAGE

PHP knihovna - <https://www.phpcaptcha.org/>



Obrázek 15 – Secureimage, zdroj: phpcaptcha.org

Dokumentace: stručná a jasná

Jednoduchost užití: triviální [88]

#### 4.6.4 BOTDETECT PHP CAPTCHA GENERATOR

PHP knihovna - <https://captcha.com/php-captcha.html>



Obrázek 16 – BotDetect PHP CAPTCHA, zdroj: captcha.com

Dokumentace: podrobná

Jednoduchost užití: jednoduchá [89]

#### 4.6.5 S3CAPCHA

jQuery plugin - <http://serie3.info/s3capcha/>



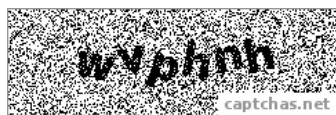
Obrázek 17 – s3Capcha, zdroj: serie3.info

Dokumentace: stručná a jasná

Jednoduchost užití: složitější [90]

#### 4.6.6 CAPTCHAS.NET

PHP knihovna - <http://captchas.net/>



Obrázek 18 – Captchas.net, zdroj: captchas.net

Dokumentace:           dostačující

Jednoduchost užití:   jednoduchá [91]

#### 4.6.7 VISUAL AND AUDIO PHP CAPTCHA GENERATION CLASS

PHP knihovna - <http://www.ejeliot.com/pages/php-captcha>



Obrázek 19 – Visual and Audio PHP CAPTCHA Generation class, zdroj: ejeliot.com

Dokumentace:           méně přehledná, ale dostatečně podrobná

Jednoduchost užití:   triviální [92]

#### 4.6.8 FREECAP PHP CAPTCHA

PHP knihovna - [http://www.puremango.co.uk/2005/04/php\\_captcha\\_script\\_113](http://www.puremango.co.uk/2005/04/php_captcha_script_113)



Obrázek 20 – freeCap PHP CAPTCHA, zdroj: puremango.co.uk

Dokumentace:           chybí

Jednoduchost užití:   jednoduchá, ale vyžaduje studium zdrojového kódu díky absenci dokumentace

#### 4.6.9 COOL PHP CAPTCHA

PHP knihovna - <https://github.com/josecl/cool-php-captcha>



Obrázek 21 – Cool PHP CAPTCHA, zdroj: github.com

Dokumentace:           jednoduchá a přehledná

Jednoduchost užití:   triviální [93]

#### 4.6.10 QUICKCAPTCHA 1.0

PHP knihovna - <http://www.web1marketing.com/resources/tools/quickcaptcha/>



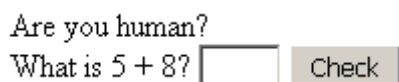
Obrázek 22 – QuickCaptcha 1.0, zdroj: [www.web1marketing.com](http://www.web1marketing.com)

Dokumentace: chybí

Jednoduchost užití: složitá, nutnost studia zdrojových kódů

#### 4.6.11 SIMPLE JAVASCRIPT CAPTCHA

Javascriptová knihovna - <https://sourcefirst.wordpress.com/2007/05/02/simple-javascript-captcha/>



Obrázek 23 – Simple Javascript CAPTCHA

Dokumentace: zcela chybí

Jednoduchost užití: velice jednoduchá

#### 4.6.12 HOOMANTEST: A MOOTOOLS BOT FILTERING CAPTCHA CLASS

Javascriptová knihovna - [http://fraggd.org/dev/hOOmanTest\\_captcha\\_for\\_mootools.php](http://fraggd.org/dev/hOOmanTest_captcha_for_mootools.php)

Tato knihovna je zajímavá tím, že nejde o klasické opisování, ale většinou o splnění nějakého pro člověka triviálního úkolu. Na obrázku dole je vidět jeden z úkolů. Identifikovat krychle a přetáhnout je na místo s textem „DROP HERE“. Jednoduché pro člověka, ale už ne pro BOTa.



Obrázek 24 – hoomanTest: A Mootools Bot Filtering Captcha Class

Dokumentace: dostačující

Jednoduchost užití: mírně složitější [94]

### 4.6.13 RECAPTCHA - UKÁZKOVÉ ŘEŠENÍ

Osobně doporučuji použít pro generování CAPTCHA kódu vybrat knihovnu **ReCAPTCHA**, protože se jedná o prověřenou, spolehlivou a plně funkční knihovnu od velké a známé firmy Google.

Následuje ukázka řešení tohoto problému pomocí knihovny **ReCAPTCHA**:

Soubor „*Index.php*“:

```
<?php
    require_once('recaptchalib.php');
    $publickey = "your_public_key"; // Tento klíč získáme po registraci
        aplikace
?>
<form method="post" action="verify.php">
    <?php
        require_once('recaptchalib.php');
        $publickey = "KLÍČ ZÍSKANÝ PŘI PŘIHLÁŠENÍ";
        echo recaptcha_get_html($publickey);
    ?>
    <input type="submit" />
</form>
```

Soubor „*verify.php*“:

```
<?php
    require_once('recaptchalib.php');
    $privatekey = "your_private_key"; // Tento klíč získáme po registraci
        aplikace
    $resp = recaptcha_check_answer ($privatekey,
                                    $_SERVER["REMOTE_ADDR"],
                                    $_POST["recaptcha_challenge_field"],
                                    $_POST["recaptcha_response_field"]);

    if (!$resp->is_valid) {
        die ("Ověřovací test proběhl NEúspěšně" .
            "(chyba reCAPTCHA: " . $resp->error . ")");
    } else {
        echo "Ověřovací test proběhl úspěšně";
    }
?>
```

Řešení je převzato z dokumentace knihovny a následně upraveno tak, aby obsahovalo pouze nezbytné části kódu. První soubor provádí zobrazení CAPTCHA, druhý soubor ověřuje, zda test na ověření proběhl úspěšně. Proměnná „*resp*“ obsahuje hodnotu „*True*“ v případě, že ověřovací test proběhl úspěšně. Důležité je upravit hodnotu proměnných „*publickey*“ a „*privatekey*“. Tyto hodnoty získáme po registraci aplikace zde: <https://www.google.com/recaptcha/admin>

## 4.7 OSTATNÍ

Do této kategorie spadají všechny knihovny, které se nedaly zařadit do žádné z předchozích.

#### 4.7.1 BIGPICTURE

Javascriptová knihovna - <http://www.bigpicture.bi>

Díky této knihovně je možné neomezeně přibližovat či oddalovat text. Umožňuje i editaci textu a může se tak snadno změnit v poznámkový blok neomezené velikosti. Napsaný text je možné samozřejmě také ukládat. Na oficiálních stránkách je k nalezení opravdu působivé funkční demo. Knihovna je šířena pod licencí MIT.

Dokumentace: chybí, ale na [github.com](https://github.com) je stručný tutoriál

Jednoduchost užití: díky dostupnému tutoriálu snadná

#### 4.7.2 HTML PURIFIER

PHP knihovna - <http://htmlpurifier.org/>

Hlavní výhodou použití této knihovny je odstranění XSS<sup>8</sup> zranitelností v původním HTML kódu. XSS je zkráceně a zjednodušeně útok proti uživateli pomocí zranitelnosti ve webové aplikaci. Tato zranitelnost se vyskytuje na spoustě stránek a často ji vývojáři ignorují. Vystavují tak ale méně zkušené uživatele zbytečnému riziku. Základní rozdělení XSS je perzistentní a non-perzistentní. Perzistentní XSS znamená, že se díky této mezeře podaří útočníkovi umístit škodlivý kód na webovou stránku natrvalo. Non-perzistentní představuje o něco menší problém. XSS však nelze podceňovat. Při správně provedeném útoku ve spojení s neznalostí oběti může vést tato zranitelnost až k zisku přihlašovacích údajů. [95]

Dokumentace: jednoduchá, ale dostačující

Jednoduchost užití: triviální [96]

#### 4.7.3 XML-RPC PHP LIBRARY

PHP knihovna - <http://gggeek.github.io/phpxmlrpc>

Knihovna sloužící k usnadnění odesílání požadavků pomocí XML-RPC protokolu. Hlavním účelem této knihovny je usnadnění trackbacku. Jedná se o techniku navyšování návštěvnosti vlastního webu. [97] [98]

Dokumentace: podrobná, ale složitější na procházení

---

<sup>8</sup> XSS – Cross Site Scripting



Jednoduchost užití: složitější [99]

#### **4.7.4 PHPMAILER**

PHP knihovna - <https://github.com/Synchro/PHPMailer>

Jedná se o knihovnu usnadňující práci s emaily. Dokáže emaily jak přijímat, tak i odesílat. Pro příjem podporuje protokoly IMAP i POP3.

Dokumentace: stručná a přehledná

Jednoduchost užití: triviální [100]

#### **4.7.5 SWIFTMAILER**

PHP knihovna - <http://swiftmailer.org/>

Knihovna, která slouží k odesílání emailů. Vyžaduje minimálně verze PHP 5 a ke komunikaci se serverem SMTP nepotřebuje a ani nepoužívá funkce „mail“. Autoři uvádějí, že díky tomu při odesílání nespotřebuje tolik systémových prostředků.

Dokumentace: méně přehledná, ale dostačující

Jednoduchost užití: jednoduchá [101]

#### **4.7.6 GEOCODER**

PHP miniframework - <http://geocoder-php.org/Geocoder/>

Jedná se o menší množství knihoven, které umožňují například určit polohu a ve spolupráci s Google maps vrátit nejen zeměpisnou výšku a šířku, ale i adresu lokace na základě získaných parametrů.

Dokumentace: dostatečně podrobná, ale méně přehledná

Jednoduchost užití: jednodušší [102]

#### **4.7.7 OPAUTH**

PHP knihovna - <http://oauth.org/>

Usnadnění OAuth autentizace, která je vyžadována u většiny webových API.

Dokumentace: dostatečně podrobná a přehledná

Jednoduchost užití: složitější [103]

## ZÁVĚR

Porovnávat různorodé frameworky je velice složité. Nakonec si myslím, že se mi podařilo vybrat vhodný styl a strukturu, které přehledně a výstižně zpracovává vybrané téma frameworků a knihoven jako pomocníků při tvorbě webů s pokročilými funkcemi. Výsledkem práce je ucelený přehled frameworků a knihoven. Frameworky jsou rozdělené na Javascriptové a PHP. Jejich přehled je uskutečněn pomocí tabulek, které jsou následně řádně okomentované. Podařilo se mi najít alespoň některá společná kritéria pro všechny frameworky. Práce obsahuje pečlivě zvolené, podle mého názoru, nejužitečnější a nejzajímavější vlastnosti a parametry, které lze u vybraných frameworků porovnat či ohodnotit. V práci jsou zjednodušeně, ale výstižně představeny dva frameworky. Na příkladech je vidět, jakým způsobem probíhá práce s frameworky a její obtížnost.

Knihovny jsou rozděleny do kategorií podle způsobu užití, což mi přišlo nejrozumnější. Existuje opravdu nepřehledné množství knihoven, proto jsem se do každé kategorie snažil vložit dostatečný počet knihoven na výběr. K usnadnění výběru jistě přispěje informace o tom, zda má knihovna kvalitní dokumentaci a jak snadné je její pochopení a užití v praxi. Některé typy knihoven charakterizoval mnohem výstižněji obrázek, proto je občas zvoleno toto řešení. U jednotlivých knihoven jsem se pak snažil ukázat na nejdůležitější vlastnosti či omezení a tím tak dopomoci ke snadnějšímu výběru. Na konci každé kategorie je vybrána knihovna či knihovny, které mi vyhovovaly ze všech nejvíce. Následuje minimálně jedna ukázka řešení konkrétního problému. Většinou jde o to ilustrovat, jak vypadá práce s vybranou knihovnou.

Jak u frameworků, tak u knihoven jsem se snažil o vypsání těch nejdůležitějších informací. Jednoznačně mi šlo o co největší výstižnost a informační hodnotu všech textů této práce.

Myslím si, že se mi podařilo úspěšně splnit zadání ve všech bodech.

## **RESUMÉ**

This bachelor work is about PHP and Javascript frameworks and libraries. There are explained the most important terms related to this topic. Then there is specified how these tools can actually help during the process of creation of web application. Frameworks are divided into two categories. Javascript and PHP frameworks. There are overviews of chosen frameworks for each of the category. These overviews are made using tables. There are several attributes that are compared between each of the framework. Two frameworks are introduced more in detail. Then you can find here overview of libraries divided into several categories. There are also several examples of usage few tools mostly to show how easy can be developing of web applications when using these tools. Readers can easily find out from the source code how hard is to work with specific library or framework. The main goal of this work is to introduce useful tools that help to develop web applications.

**SEZNAM LITERATURY**

1. BÖHMER, M. *Zend Framework: programujeme webové aplikace v PHP*. Brno: Computer Press, 2010. ISBN 978-80-251-2965-4.
2. GASSTON, P. *Moderní web*. Brno: Computer Press, 2015. ISBN 978-80-251-4345-2.
3. Knihovna (programování). [online]. [cit. 2016-06-03]. Dostupné z: [https://cs.wikipedia.org/wiki/Knihovna\\_\(programov%C3%A1n%C3%AD\)](https://cs.wikipedia.org/wiki/Knihovna_(programov%C3%A1n%C3%AD))
4. Framework. [online]. [cit. 2016-06-03]. Dostupné z: <https://cs.wikipedia.org/wiki/Framework>
5. Systém pro správu obsahu. [online]. [cit. 2016-06-03]. Dostupné z: [https://cs.wikipedia.org/wiki/Syst%C3%A9m\\_pro\\_spr%C3%A1vu\\_obsahu](https://cs.wikipedia.org/wiki/Syst%C3%A9m_pro_spr%C3%A1vu_obsahu)
6. Plugin. [online]. [cit. 2016-06-03]. Dostupné z: <https://cs.wikipedia.org/wiki/Plugin>
7. ČÁPKA, D. Úvod do knihoven a frameworků v PHP. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.itnetwork.cz/php/knihovny/php-tutorial-uvod-do-knihoven-a-frameworku/>
8. WALSH, D. 6 Reasons To Use JavaScript Libraries & Frameworks. [online]. [cit. 2016-06-03]. Dostupné z: <https://davidwalsh.name/6-reasons-to-use-javascript-libraries-frameworks>
9. 10 criteria for choosing the correct framework. [online]. [cit. 2016-06-03]. Dostupné z: <http://symfony.com/ten-criteria>
10. BSD Licence. [online]. [cit. 2016-27-06]. Dostupné z: [https://cs.wikipedia.org/wiki/BSD\\_licence](https://cs.wikipedia.org/wiki/BSD_licence)
11. GNU General Public License. [online]. [cit. 2016-06-03]. Dostupné z: [https://cs.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](https://cs.wikipedia.org/wiki/GNU_General_Public_License)
12. Licence MIT. [online]. [cit. 2016-06-03]. Dostupné z: [https://cs.wikipedia.org/wiki/Licence\\_MIT](https://cs.wikipedia.org/wiki/Licence_MIT)
13. Creative Commons. [online]. [cit. 2016-06-03]. Dostupné z: [https://cs.wikipedia.org/wiki/Creative\\_Commons](https://cs.wikipedia.org/wiki/Creative_Commons)
14. What are migrations? [online]. [cit. 2016-06-03]. Dostupné z: <http://south.readthedocs.io/en/latest/whataremigrations.html>
15. NOTTINGHAM, M. Caching Tutorial for Web Authors and Webmasters. [online]. [cit. 2016-06-03]. Dostupné z: [https://www.mnot.net/cache\\_docs/](https://www.mnot.net/cache_docs/)
16. BORYSOV, P. PHP Frameworks: Hall of Fame. [online]. [cit. 2016-06-03]. Dostupné z: <http://ddi-dev.com/blog/programming/php-frameworks-hall-fame/>
17. CakePHP 3.2 Red Velvet Cookbook. [online]. [cit. 2016-06-03]. Dostupné z: <http://book.cakephp.org/3.0/en/index.html>
18. Dokumentace | Nette Framework. [online]. [cit. 2016-06-03]. Dostupné z: <https://doc.nette.org/cs/2.4/>

19. Laravel API. [online]. [cit. 2016-06-03]. Dostupné z: <https://laravel.com/api/5.1>
20. Learn Symfony. [online]. [cit. 2016-06-03]. Dostupné z: <http://symfony.com/doc/current/index.html>
21. PHPixie Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <https://phpixie.com/docs.html>
22. Phalcon 2.0.10 documentation. [online]. [cit. 2016-06-03]. Dostupné z: <https://docs.phalconphp.com/en/latest/index.html>
23. Programmer's Reference Guide of Zend Framework 2. [online]. [cit. 2016-06-03]. Dostupné z: <http://framework.zend.com/manual/current/en/index.html>
24. Comparison of web frameworks. [online]. [cit. 2016-06-27]. Dostupné z: [https://en.wikipedia.org/wiki/Comparison\\_of\\_web\\_frameworks](https://en.wikipedia.org/wiki/Comparison_of_web_frameworks)
25. FuelPHP Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://fuelphp.com/docs/>
26. Yii PHP Framework Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.yiiframework.com/doc/>
27. Laravel. [online]. [cit. 2016-06-03]. Dostupné z: <https://cs.wikipedia.org/wiki/Laravel>
28. ČÁPKA, D. MVC architektura. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor/>
29. OTWELL, T. Laravel - The PHP Framework For Web Artisans. [online]. [cit. 2016-06-03]. Dostupné z: <https://laravel.com/docs/5.2>
30. LECKY-THOMPSON, E. a S. D NOWICKI. *PHP 6: programujeme profesionálně*. Brno: Computer Press, 2010. ISBN 978-80-251-3127-5.
31. Strengths and weakness of Laravel 5 PHP frame work. [online]. [cit. 2016-06-03]. Dostupné z: <http://agriyaservices.blogspot.cz/2015/07/strengths-and-weakness-of-laravel-5-php.html>
32. Laravel Tutorial. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.tutorialspoint.com/laravel/>
33. Comparison of JavaScript frameworks. [online]. [cit. 2016-06-03]. Dostupné z: [https://en.wikipedia.org/wiki/Comparison\\_of\\_JavaScript\\_frameworks](https://en.wikipedia.org/wiki/Comparison_of_JavaScript_frameworks)
34. AngularJS API Docs. [online]. [cit. 2016-06-03]. Dostupné z: <https://docs.angularjs.org/api>
35. Backbone.js documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://devdocs.io/backbone/>
36. Ember.js - Guides and Tutorials. [online]. [cit. 2016-06-03]. Dostupné z: <https://guides.emberjs.com/v2.6.0/>
37. GWT Project Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.gwtproject.org/doc/latest/DevGuide.html>

38. jQuery UI Widgets - Help Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.jqwidgets.com/jquery-widgets-documentation/>
39. React - Getting Started. [online]. [cit. 2016-06-03]. Dostupné z: <https://facebook.github.io/react/docs/getting-started.html>
40. Webix Docs. [online]. [cit. 2016-06-03]. Dostupné z: <http://docs.webix.com/>
41. AngularJS. [online]. [cit. 2016-06-03]. Dostupné z: <https://cs.wikipedia.org/wiki/AngularJS>
42. AngularJS - Superheroic JavaScript MVW Framework. [online]. [cit. 2016-06-03]. Dostupné z: <https://angularjs.org/>
43. MROZEK, J. Začínáme s AngularJS. [online]. [cit. 2016-06-03]. Dostupné z: <https://www.zdrojak.cz/clanky/zaciname-s-angularjs/>
44. HANÁK, D. Úvod do AngularJS. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.itnetwork.cz/javascript/angularjs/javascript-tutorial-uvod-do-angularjs/>
45. AngularJS Tutorial. [online]. [cit. 2016-06-28]. Dostupné z: <http://www.w3schools.com/angular/>
46. DRUMELIS, V. 15 Useful JavaScript Libraries for January 2015. [online]. [cit. 2016-06-03]. Dostupné z: <https://codegeekz.com/15-useful-javascript-libraries-january-2015/>
47. DRUMELIS, V. 35 JavaScript Chart and Graph Libraries for Developers. [online]. [cit. 2016-06-03]. Dostupné z: <https://codegeekz.com/javascript-chart-and-graph-libraries/>
48. PERMESSUR, A. 15 JavaScript Data Grids to Enhance your HTML Tables. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.hotscripts.com/blog/15-javascript-data-grids-enhance-html-tables/>
49. PERMESSUR, A. 50+ Extremely Useful PHP Classes & Libraries. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.hotscripts.com/blog/50-php-classes-libraries/>
50. SANDRA, J. 12 Extremely Useful PHP Classes & Libraries. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.webhostingreviewlist.com/php/12-extremely-useful-php-classes-libraries/>
51. LOU, M. 21 Free CAPTCHA Sources. [online]. [cit. 2016-06-03]. Dostupné z: <http://tympanus.net/codrops/2009/09/22/21-free-captcha-sources/>
52. PHP Text to Image. [online]. [cit. 2016-06-03]. Dostupné z: <https://www.daftlogic.com/projects-text-to-image.htm>
53. GABOS, S. Zebra\_Image, a lightweight image manipulation library written in PHP - Class Zebra\_Image. [online]. [cit. 2016-06-03]. Dostupné z: [http://stefangabos.ro/wp-content/docs/Zebra\\_Image/Zebra\\_Image/Zebra\\_Image.html](http://stefangabos.ro/wp-content/docs/Zebra_Image/Zebra_Image/Zebra_Image.html)
54. Imagine Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://imagine.readthedocs.io/en/latest/>

55. GitHub - Snappy. [online]. [cit. 2016-06-03]. Dostupné z: <https://github.com/KnpLabs/snappy>
56. WidelImage Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://wideimage.sourceforge.net/documentation/>
57. ÜberGallery introduction. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.ubergallery.net/>
58. Coppermine Photo Gallery - Documentation and Manual. [online]. [cit. 2016-06-03]. Dostupné z: <http://documentation.coppermine-gallery.net/en/>
59. FusionCharts Developer Center. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.fusioncharts.com/dev/>
60. Ember Charts. [online]. [cit. 2016-06-03]. Dostupné z: <http://opensource.addepar.com/ember-charts/#/overview>
61. Chart.js API documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.chartjs.org/docs/>
62. Aristochart Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://adriancooney.ie/aristochart/doc/index.html>
63. BonsaiJS documentation - Introduction. [online]. [cit. 2016-06-03]. Dostupné z: <http://docs.bonsaijs.org/>
64. Morris.js. [online]. [cit. 2016-06-03]. Dostupné z: <http://morrisjs.github.io/morris.js/>
65. Amcharts Docs. [online]. [cit. 2016-06-03]. Dostupné z: <https://docs.amcharts.com/3/>
66. Fancy Form - JavaScript form library on jQuery - Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://fancyjs.com/docs.html>
67. PHP DataGrid Control - Getting started. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.apphp.com/php-datagrid/index.php?page=getting-started>
68. PHP Form Validation Script. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.html-form-guide.com/php-form/php-form-validation.html>
69. GitHub - EditableGrid. [online]. [cit. 2016-06-03]. Dostupné z: <https://github.com/webismymind/editablegrid>
70. dhtmlxGrid - Documentation. [online]. [cit. 2016-06-03]. Dostupné z: [http://docs.dhtmlx.com/grid\\_\\_index.html](http://docs.dhtmlx.com/grid__index.html)
71. Fine Uploader Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://docs.fineuploader.com/>
72. DataTables - Manual. [online]. [cit. 2016-06-03]. Dostupné z: <https://datatables.net/manual/index>
73. ADOdb - Database Abstraction Layer for PHP. [online]. [cit. 2016-06-03]. Dostupné z: <http://adodb.org/dokuwiki/doku.php>

74. PHP Advanced Database Class Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://slaout.linux62.org/php/db.doc.html>
75. Finally, the simple pagination class. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.catchmyfame.com/2007/07/28/finally-the-simple-pagination-class/>
76. GitHub - Danpu. [online]. [cit. 2016-06-03]. Dostupné z: <https://github.com/gocom/danpu>
77. MARGORÍN, M. *JQuery bez předchozích znalostí*. Brno: Computer Press, 2011. ISBN 978-80-251-3379-8.
78. AJAX. [online]. [cit. 2016-06-03]. Dostupné z: <https://cs.wikipedia.org/wiki/AJAX>
79. jQuery API. [online]. [cit. 2016-06-03]. Dostupné z: <http://api.jquery.com/>
80. PHP-Collection Documentation (master). [online]. [cit. 2016-06-03]. Dostupné z: <http://jmsyst.com/libs/php-collection>
81. Services\_JSON - documentation. [online]. [cit. 2016-06-03]. Dostupné z: [http://pear.php.net/package/Services\\_JSON/docs/latest/Services\\_JSON/\\_Services\\_JSON-1.0.3---JSON.php.html](http://pear.php.net/package/Services_JSON/docs/latest/Services_JSON/_Services_JSON-1.0.3---JSON.php.html)
82. TCPDF Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.tcpdf.org/docs.php>
83. phpDocumentor Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <https://phpdoc.org/docs/latest/index.html>
84. jQuery Tutorial. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.w3schools.com/jquery/>
85. CAPTCHA. [online]. [cit. 2016-06-03]. Dostupné z: <https://cs.wikipedia.org/wiki/CAPTCHA>
86. Getting Started reCAPTCHA Google Developers. [online]. [cit. 2016-06-03]. Dostupné z: <https://developers.google.com/recaptcha/docs/start>
87. SimpleCaptcha - Customizing Your Image CAPTCHAs. [online]. [cit. 2016-06-03]. Dostupné z: [http://simplecaptcha.sourceforge.net/custom\\_images.html](http://simplecaptcha.sourceforge.net/custom_images.html)
88. Securimage PHP Captcha | Docs. [online]. [cit. 2016-06-03]. Dostupné z: <https://www.phpcaptcha.org/documentation/>
89. BotDetect CAPTCHA Documentation Lists. [online]. [cit. 2016-06-03]. Dostupné z: <https://captcha.com/documentation.html>
90. s3Capcha jQuery captcha plugin. [online]. [cit. 2016-06-03]. Dostupné z: <http://serie3.info/s3capcha/>
91. Captchas.net - Sample PHP Implementation. [online]. [cit. 2016-06-03]. Dostupné z: <http://captchas.net/sample/php/>
92. Visual and Audio PHP CAPTCHA Generation Class. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.ejeliot.com/pages/php-captcha>



93. GitHub - Cool PHP Captcha. [online]. [cit. 2016-06-03]. Dostupné z: <https://github.com/josecl/cool-php-captcha>
94. hOOmanTest: a mootools bot filtering "captcha" class. [online]. [cit. 2016-06-03]. Dostupné z: [http://fraggged.org/dev/hOOmanTest\\_captcha\\_for\\_mootools.php](http://fraggged.org/dev/hOOmanTest_captcha_for_mootools.php)
95. KÜMMEL, R. Pokročilé techniky XSS. [online]. [cit. 2016-06-03]. Dostupné z: <http://www.soom.cz/clanky/485--Pokrocile-techniky-XSS>
96. HTML Purifier - Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://htmlpurifier.org/docs>
97. What is XML-RPC? [online]. [cit. 2016-06-03]. Dostupné z: <http://xmlrpc.scripting.com/>
98. STARAK, Y. What Is A Trackback And How Can It Increase Your Blog Traffic? [online]. [cit. 2016-06-03]. Dostupné z: <https://www.entrepreneurs-journey.com/newsletters/what-is-a-trackback-and-how-can-it-increase-your-blog-traffic/>
99. XML-RPC for PHP Documentation. [online]. [cit. 2016-06-03]. Dostupné z: <http://gggeek.github.io/phpxmlrpc/doc-1.1/>
100. GitHub - PHPMailer. [online]. [cit. 2016-06-03]. Dostupné z: <https://github.com/Synchro/PHPMailer>
101. Introduction - Documentation - Swift Mailer. [online]. [cit. 2016-06-03]. Dostupné z: [swiftmailer.org/docs/introduction.html](http://swiftmailer.org/docs/introduction.html)
102. Geocoder - The almost missing Geocoder PHP library! [online]. [cit. 2016-06-03]. Dostupné z: <http://geocoder-php.org/Geocoder/>
103. GitHub - Opauth. [online]. [cit. 2016-06-03]. Dostupné z: <https://github.com/opauth/opauth/wiki>

**SEZNAM OBRÁZKŮ A TABULEK**

Obrázek 1 – Model View Controller, zdroj: itnetwork.cz .....	17
Obrázek 2 – Laravel – ukázka struktury složky .....	19
Obrázek 3 – Laraval – ukázka výstupu v češtině .....	19
Obrázek 4 – Laravel – ukázka výstupu v angličtině.....	19
Obrázek 5 – Laravel – ukázka základních formulářových prvků.....	21
Obrázek 6 – výsledek Ajax requestu .....	22
Obrázek 7 – Galerie UberGallery, zdroj: ubergallery.net .....	32
Obrázek 8 – AmCharts – ukázkový graf .....	36
Obrázek 9 – Amcharts – ukázková mapa .....	38
Obrázek 10 – Morris.js – ukázkový graf .....	39
Obrázek 11 – ApPHP DataGrid – ukázka tabulky .....	42
Obrázek 12 – Fancy Form – jednoduchý formulář.....	44
Obrázek 13 – ReCAPTCHA, zdroj: google.com .....	48
Obrázek 14 – SimpleCaptcha .....	49
Obrázek 15 – Secureimage, zdroj: phpcaptcha.org .....	49
Obrázek 16 – BotDetect PHP CAPTCHA, zdroj: captcha.com .....	49
Obrázek 17 – s3Capcha, zdroj: serie3.info.....	49
Obrázek 18 – Captchas.net, zdroj: captchas.net .....	50
Obrázek 19 – Visual and Audio PHP CAPTCHA Generation class, zdroj: ejeliot.com.....	50
Obrázek 20 – freeCap PHP CAPTCHA, zdroj: puremango.co.uk .....	50
Obrázek 21 – Cool PHP CAPTCHA, zdroj: github.com .....	50
Obrázek 22 – QuickCaptcha 1.0, zdroj: www.web1marketing.com .....	51
Obrázek 23 – Simple Javascript CAPTCHA.....	51
Obrázek 24 – hoomanTest: A Mootools Bot Filtering Captcha Class .....	51
Tabulka 1 - PHP frameworky část 1 .....	15
Tabulka 2 - PHP frameworky část 2.....	15
Tabulka 3 - PHP frameworky část 3.....	15
Tabulka 4 - Javascriptové frameworky část 1 .....	23
Tabulka 5 - Javascriptové frameworky část 2 .....	23

---