

**Ročníková práce**

Vývoj vlastního front-end frameworku pro tvorbu webdesignu

Jan, Prokop, C3

Konzultant: Lukáš Zelenka

Školní rok: 2015 / 2016

**Čestné prohlášení**

Prohlašuji, že jsem předloženou ročníkovou práci vypracoval zcela samostatně a uvádím v ní veškeré prameny, které jsem použil. Všechny převzaté informace jsem řádně ocitoval.

V Rakovníku, dne

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (podpis studenta).

# Anotace

Tématem mé ročníkové práce je tvorba CSS Framework, které má za úkol webdesignerům usnadnit jejich práci s navrhnutím struktury HTML dokumentu. Zde budu rozebírat jeho vytvoření, jeho části a jeho využití v praxi.

Tato práce má sloužit jako návod a vysvětlení, co je to Framework, kde a jak ho použít a kdo Framework využije.

V tomto návodu budete postupně seznámeni s veškerými dostupnými funkcemi tohoto frameworku a navíc je vše doplněno o příklady použití tak, aby bylo viditelné, kterými funkcemi Framework disponuje.

# Klíčová slova

Framework, Grid systém, CSS, HTML, Web, Front-end

Obsah

[1 Anotace 3](#_Toc442302659)

[2 Klíčová slova 3](#_Toc442302660)

[3 Úvod 5](#_Toc442302661)

[4 Metodika 5](#_Toc442302662)

[5 Co je to Framework 5](#_Toc442302663)

[6 Grid systém 6](#_Toc442302664)

[6.1 Pravidla grid sytému 6](#_Toc442302665)

[6.2 Příklad grid systému 7](#_Toc442302666)

[6.3 Odsazení sloupců v containeru 8](#_Toc442302667)

[6.4 Zarovnání sloupců v containeru 9](#_Toc442302668)

[7 Komponenty 9](#_Toc442302669)

[7.1 Formuláře 9](#_Toc442302670)

[7.1.1 Formulář v řádce 10](#_Toc442302671)

[7.1.2 Vodorovný formulář 10](#_Toc442302672)

[7.1.3 Základní textové pole 12](#_Toc442302673)

[7.1.4 Textarea 12](#_Toc442302674)

[7.1.5 Zaškrtávací políčka a volba možností 12](#_Toc442302675)

[7.1.6 Zaškrtávací políčka a volba možností v řádce 13](#_Toc442302676)

[7.2 Tabulky 13](#_Toc442302677)

[7.2.1 Ohraničená tabulka 14](#_Toc442302678)

[7.2.2 Pomocné třídy u tabulek 14](#_Toc442302679)

[8 Možnosti rozšíření a úprav frameworku 14](#_Toc442302680)

[8.1 Služba Github 15](#_Toc442302681)

[8.2 Git repozitář 15](#_Toc442302682)

[9 Licenční politika 15](#_Toc442302683)

[9.1 Co je to svobodný software 15](#_Toc442302684)

[9.2 MIT licence 15](#_Toc442302685)

[10 Závěr 16](#_Toc442302686)

[11 Seznam použité literatury 17](#_Toc442302687)

# Úvod

V mé ročníkové práci se budu zabývat tvorbou vlastního CSS Framework a následně seznámím uživatele této ročníkové práce se všemi možnostmi a funkcemi tohoto frameworku.

Tato práce má sloužit jako šablona a usnadnění práce budoucím kodérům a programátorům v jejich práci a také jako následný manuál a vysvětlení různých funkcí tohoto frameworku, aby měl uživatel používání tohoto frameworku co nejjednodušší a nemusel hledat, co daná třída dělá a jaké jsou její možnosti.

# Metodika

Nejdříve jsem začal s prototypováním celého projektu na papír, abych věděl, co všechno bude tento Framework obsahovat a následně jsem navrhnutý prototyp převedl do elektronické podoby a začal s tvorbou vlastního kódu CSS pomocí programu PSPad, kde jsem začal sepisovat jednotlivé funkce frameworku. Začal jsem s grid systémem. V grid systému jsem nejdříve vytvořil fluidní container a poté klasický container následně jsem zadal přesnou šířku každému sloupci a nastavil další funkce, jako je odsazení a také posunutí do stran.

Po vytvoření celého grid systému jsem začal s tvorbou formulářů. Jednotlivým položkám formuláře byly nastaveny výchozí hodnoty, které si dále uživatel může upravit. Formuláře jsem koncipoval tak, aby je bylo možné aplikovat kdekoliv na stránce a jejich strukturu ovlivňovat za pomocí grid systému.

Následně jsem nastavil výchozí hodnoty klasickým tabulkám a poté přidal další třídu, aby uživatel mohl používat i ohraničenou tabulku a využít tak všech funkcí, kterými Framework disponuje.

Po dokončení tabulek jsem přidal zakulacení u obrázků a tlačítek.

Pro kód jsem použil své znalosti z předmětu web design a také jsem se inspiroval v již některých vytvořených frameworcích, které jsem několikrát prošel, pročetl manuál a následně z nich čerpal inspiraci pro můj vlastní front-end Framework.

Tento postup jsem si vybral, protože pro mě bylo nejjednodušší navrhnout si celý projekt na papír a následně pro něj vytvořit kód, příslušný manuál a samotnou ročníkovou práci.

Data a informace pro vytvoření celé práce jsem hledal na různých webových stránkách o frameworcích a také u svého učitele Lukáše Zelenky.

# Co je to Framework

Framework neboli aplikační rámec je softwarová struktura a šablona sloužící k usnadnění práce uživateli a zjednodušení práce při různých projektech.

Cílem frameworku je co nejvíce usnadnit uživateli jeho práci, a vytvořit mu co nejjednodušší šablonu pro psaní jeho kódu.

Někdy se můžeme setkat s názory, že použití frameworku je špatné a že uživatel často musí daný Framework důkladně nastudovat. Tak to ale není. Při jednorázovém použití tomu tak může být, ale když daný Framework používáte opakovaně, je to opravdu veliké usnadnění.

Velikou výhodou je, že dostanete již předformátovanou funkční šablonu, kterou můžete dále upravovat.

Další z výhod je, že Framework můžeme použít prakticky na každý projekt a vždy si ho podle vlastní libosti upravit.

Naopak nevýhodou je, že Framework musíte nastudovat a přečíst si jeho manuál, nebo vždy vyhledat jeho funkce.

**Příklady frameworku:**

**Jquery:** „*je javascriptová knihovna s širokou podporou prohlížečů, která klade důraz na interakci mezi JavaScriptem a HTML. Byla vydána Johnem Resigem v lednu 2006 na newyorském BarCampu*.“[1]

**Bootstrap:** *„je volně stažitelná sada nástrojů pro tvorbu webu a webových aplikací. Obsahuje návrhářské šablony založené na HTML a CSS, sloužící pro úpravu typografie, formulářů, tlačítek, navigace a dalších komponent rozhraní, stejně jako další volitelná rozšíření JavaScriptu. Je to nejúspěšnější projekt na GitHub s 65000+ hvězdičkami.“[2]*

**Foundation Framework:** *„ je responsivní front-end Framework pro tvorbu webu a webových apliakcí. Obsahuje návrhářské šablony založené na HTML a CSS sloužící pro úpravu typografie, formulářů, tlačítek, navigace a dalších komponent rozhraní. Bylo vydáno v září 2011 firmou ZURB.[3]*

**Yahoo Pure CSS:** je responsivní front-end Framework pro tvorbu webu a webových apliakcí.

**Nette Framework: „***je open source framework pro tvorbu webových aplikací v PHP 5. Zaměřuje se na eliminaci bezpečnostních rizik, podporuje AJAX, DRY, KISS, MVC a znovupoužitelnost kódu. Využívá událostmi řízené programování a z velké části je založen na použití komponent. Původním autorem Nette Frameworku je David Grudl, o jeho další rozvoj se stará organizace Nette Foundation. Nette Framework je svobodný software, nabízený pod licencemi GNU GPL a licencí Nette, která je obdobou původní (staré, čtyřbodové) BSD licence“[4]*

A spousta dalších.

# Grid systém

Grid systém se využívá při tvorbě rozložení webových stránek pomocí sloupců a řádků. Pomáhá nám určit strukturu a rozložení webové stránky.

Řádky a sloupce mohou být umístěny do container-fluid (využívá plnou šíři stránky), nebo do container(využívá pevně danou šířku ze stránky).

Obsah dokumentu by měl být vždy umístěn správně do sloupců, aby se zachovala struktura webu a obsah dokumentu nebyl rozházen volně po stránce.

Grid systém tvoří dvanáct sloupců a je možné je do sebe různě vnořovat.

Dvanáct sloupců je tu z důvodu, že číslo dvanáct je dobře dělitelné dvěma, třemi a čtyřmi, proto lze vytvořit sloupce na polovinu na třetinu a na čtvrtinu stránky.

Nejužší sloupec je .col-1 naopak .col.12 je přes celou stránku.

## Pravidla grid sytému

*„Obsah by měl být umístěn uvnitř sloupce, a sloupce mohou být přímí potomci řádků.“*

*„Řádky musí být umístěny buď do .container-fluid (plná šířka stránky), nebo do .container(pevně daná šířka stránky), pro správné zarovnání a odsazení.“*

*„Předdefinované třídy, jako je .row a například .col-4 jsou k dispozici pro rychlé vytváření řádků a sloupců.“*

*„Sloupce jsou vytvořeny zadáním přesného počtu z 12 dostupných sloupců. Například, tři stejné sloupce by používaly třídu .col-4.“[5]*

## Příklad grid systému

Použitím třídy .col-1 rozdělíme šířku stránky na dvanáct stejných sloupců, přičemž musíme dvanáctkrát vytvořit element s touto třídou.

**HTML:**

<div class=“row“>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

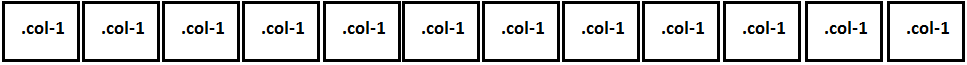
<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

<div class=“col-1“>.col-1</div>

Pokud chceme stránku rozdělit pouze na dva sloupce různé šířky, použijeme k tomu jen dvě třídy a to : .col-9 + .col-3.

**HTML:**

<div class="container">

<div class="col-9">

<h1>.col-9</h1>

</div>

<div class="col-3">

<h1>.col-3</h1>

</div>

</div>



Kdybychom chtěli stránku rozdělit na poloviny, využijeme k tomu 2 sloupce stejné šíře a to .col-6 a .col-6.

**HTML:**

<div class=“row“>

<div class=“col-6“>.col-6</div>

<div class=“col-6>.col-6</div>

</div>



## Odsazení sloupců v containeru

Pro odsazení sloupce z leva aplikujeme navíc speciální třídu .col-offset-x. X v tomto případě interpretuje počet sloupců, o které dotyčný element chceme odsadit.

**Příklad**: .col-offset-2 .col-6

Vytvoří sloupec o šíři 6, který bude z levé strany odsazen o šířku 2 sloupců

**HTML:**

<div class="container">

<div class="row">

<div class="col-3">.col-3</div>

<div class="col-3 col-offset-3">.col-3 col-offset-3</div>

</div>

</div>

****

Pro vycentrování jednoho sloupce můžeme využít speciální třídy .col-offset-x a to tak, že hodnota X bude polovina z (12 – (šíře sloupce, který chceme zastředit))/2. Tento vzorec je platný pouze za předpokladu, že šíře centrovaného sloupce je sudé číslo.

Pro liché šíře sloupců neexistuje jednoduché řešení pro vycentrování.

**Příklad:** Chceme-li vycentrovat sloupec, který má třídu .col-8. Podle vzorce spočítáme odsazení(offset):offset = (12 – 8) / 2 = 2. Proto v případě centrování .col-8 musíme přidat třídu .col-offset-2.

**HTML:**

<div class="container">

<div class="row">

<div class="col-6 col-offset-3">.col-6 .col-offset-3</div>

</div>

</div>

## Zarovnání sloupců v containeru

Sloupce v containeru můžeme zarovnat dvěma způsoby, jedním z nich je, že sloupce posuneme do prava a k tomu použijeme .col-\*-pull-right. A druhý způsob je, že zarovnáme sloupce do leva a k tomu použijeme .col-\*-pull-left.

**HTML:**

<div class="container">

<div class="row">

<div class="col-9 pull-left">.col-9 pull-left</div>

<div class="col-3 pull-right">.col-3 pull-right</div>

</div>

****</div>

# Komponenty

Komponenty představují základ pro znovu použití kódu, výrazně nám usnadňují práci a dovolují využívat práce komunity.

Komponenta je vykreslitelný objekt, například tabulky formuláře, ankety a tak dále.

Na stránce jich může být libovolný počet.

## Formuláře

Formuláře a jejich základní funkce v mém frameworku jsou nastaveny na celou šířku sloupce a samy se upravují podle šířky sloupce monitoru.

Ve formulářích nalezneme základní textová pole, která mohou mít různé typy, např.: text, heslo, textarea a tak dále.

**HTML:**

<form action="submit.php" method="post">

<input required="required" pattern=".{1,} .{1,}" type="text" name="name" placeholder="Jméno Příjmení"><br>

<input required="required" type="email" name="email" placeholder="vas@email.cz"><br>

<input required="required" pattern=".{4,32}" type="text" name="subject" placeholder="Předmět"><br>

<textarea name="zprava" placeholder="Vaše zpráva ..."></textarea><br>

<input type="submit" class="odeslat pull-right" value="Odeslat">

</form>

### Formulář v řádce

Přidáním třídy .form-inline do našeho formuláře, zajistíme, že položky formuláře budou v jedné rovině.

Klasický formulář má na svých položkách nastavenou plnou šířku 100%, ale inline formuláře mají nastaveno auto, aby položky ve formuláři mohly být vedle sebe.

**HTML:**

<form class="form-inline">

<div>

<label for="Jméno">Jméno</label>

<input type="text" class="form-control" id="Jmeno2" placeholder="Vaše Jméno">

</div>

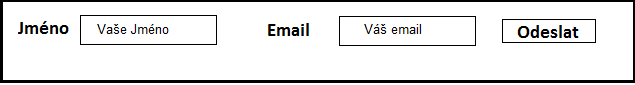
<div>

<label for="Email">Email</label>

<input type="email" class="form-control" id="exampleInputEmail2" placeholder="Váš email">

</div>

<button type="submit" class="btn btn-default">Odeslat</button>

****</form>

### Vodorovný formulář

Použitím předdefinovaného grid systému a použitím předdefinované třídy .form-horizontal, docílíme toho, že formulář bude vodorovný. Použitím třídy .form-group zajistíme, že každá položka je na nové řádce a nemusíme používat třídu .row.

**HTML:**

<form class="form-horizontal">

<div class="form-group">

<label for="inputEmail3" class="col-sm-2 control-label">Email</label>

<div class="col-10">

<input type="email" class="form-control" id="Zadatemail" placeholder="Email">

</div>

</div>

<div class="form-group">

<label for="zadej heslo">Password</label>

<div class="col-sm-10">

<input type="password" class="form-control" id="zadatheslo" placeholder="Heslo">

</div>

</div>

<div class="form-group">

<div>

<div class="checkbox">

<label>

<input type="checkbox"> Pamatovat si mě

</label>

</div>

</div>

</div>

<div class="form-group">

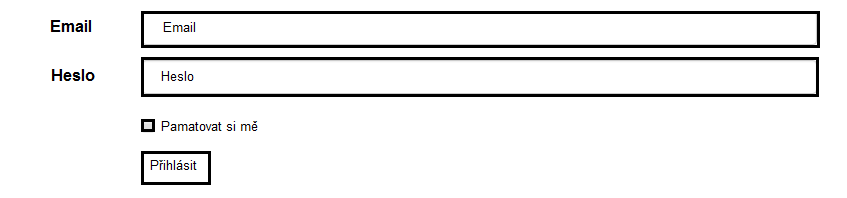
<div>

<button type="submit" class="btn btn-default">Přihlásit</button>

</div>

</div>

</form>



### Základní textové pole

<input type="text" class="form-control" placeholder="Zadejte text">

### Textarea

Položka formuláře, která umožňuje psát více textu na více řádků. Počet řádků se dá manuálně upravit podle potřeby uživatele.

<textarea class="form-control" rows="5"></textarea>

### Zaškrtávací políčka a volba možností

Zaškrtávací políčka slouží pro výběr z jedné nebo více možností. Pokud je u některých zaškrtávacích políček možnost zakázána je nutné přidat třídu .disabled.

**HTML:**

<div class="checkbox">

<label>

<input type="checkbox" value="">

První možnost

</label>

</div>

<div class="checkbox disabled">

<label>

<input type="checkbox" value="" disabled>

Druhá možnost je zablokována

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="volbamoznosti" id="moznost1" value="moznost1" checked>

První možnost

</label>

</div>

<div class="radio">

<label>

<input type="radio" name="volbamoznosti" id="moznost2" value="moznost2">

Druhá možnost

</label>

</div>

<div class="radio disabled">

<label>

<input type="radio" name="volbamoznosti" id="moznost3" value="moznost3" disabled>

Třetí možnost je zablokována

</label>

</div>



### Zaškrtávací políčka a volba možností v řádce

Použitím třídy .checkbox-inline nebo .radio-inline zajistíme, aby zaškrtávací políčka a volba možností byla vedle sebe v řádce.

<label class="checkbox-inline">

<input type="checkbox" id="pole1" value="moznost1"> 1

</label>

<label class="checkbox-inline">

<input type="checkbox" id="pole2" value="moznost2"> 2

</label>

<label class="checkbox-inline">

<input type="checkbox" id="pole3" value="moznost3"> 3

</label>

<label class="radio-inline">

<input type="radio" name="volbavradce" id="volbavradce" value="moznost1"> 1

</label>

<label class="radio-inline">

<input type="radio" name="volbavradce" id="volbavradce2" value="moznost2"> 2

</label>

<label class="radio-inline">

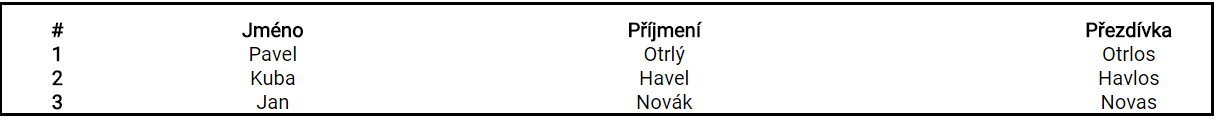
<input type="radio" name="volbavradce" id="volbavradce3" value="moznost3"> 3

</label>

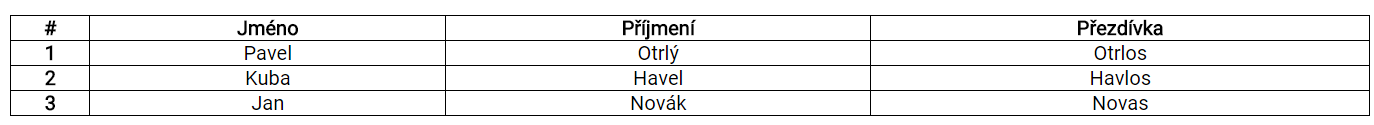
## Tabulky

Pro vytvoření tabulky v mém frameworku použijeme klasický tag <table>.

**Příklad klasické tabulky:**



### Ohraničená tabulka

Přidáním třídy .table-bordered docílíme toho, že se na naší tabulky přidá ohraničení ze všech stran.

### Pomocné třídy u tabulek

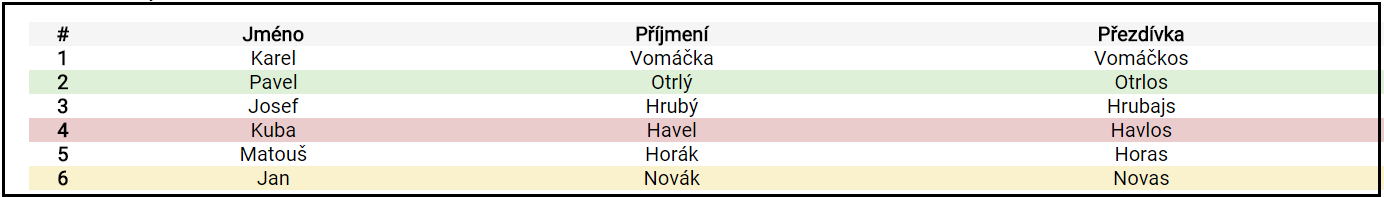
**.active** – použije barvu na konkrétní řádek nebo na konkrétní buňku.

**.succes** – označí úspěšně splněnou buňku nebo řádek.

**.info** – označí nějakou informaci v buňce nebo řádku.

**.warning** – označí buňku nebo řádek s varováním, že je potřeba této buňce nebo řádku věnovat pozornost.

**.danger –** označí buňku nebo řádek, kde by mohlo vzniknout potencionální nebezpečí.



**HTML:**

*<!-- pro řádky -->*

<tr class="active">...</tr>

<tr class="success">...</tr>

<tr class="warning">...</tr>

<tr class="danger">...</tr>

<tr class="info">...</tr>

*<!-- Pro buňky (`td` nebo `th`) -->*

<tr>

<td class="active">...</td>

<td class="success">...</td>

<td class="warning">...</td>

<td class="danger">...</td>

<td class="info">...</td>

</tr>

# Možnosti rozšíření a úprav frameworku

V HTML dokumentu je možnost připojit další stylopis a při dodržení CSS specifity v něm můžeme přetížit selektory frameworku a podle vlastní libosti můžeme dále celý dokument stylovat a upravovat.

Jelikož Framework neslouží ke stylování stránek, ale pouze jako pomocník k usnadnění pozicování elementů na stránce pomocí tříd, tak je doporučeno vytvořit vlastní stylopis a v něm následně upravovat svou stránku.

Celý projekt je možné editovat a dále rozšiřovat na službě Github. Projekt lze najít v repozitáři Proky/css-fw.

## Služba Github

*„GitHub je webová služba podporující vývoj softwaru při používání verzovacího nástroje Git. GitHub nabízí bezplatný hosting pro open-source projekty. Po zaplacení měsíčního poplatku je možné ukládat soukromé repositáře. Projekt byl spuštěn v roce 2008 zakladateli Tom Preston-Werner, Chris Wanstrath a PJ Hyett.*

*GitHub hostuje přes 11 milionů repositářů. Pro uživatele poskytuje funkce sociálních sítí - notifikace o změnách, diskuze nad kódem, návrhy změn či zasílání vlastních řešení (pull-requesty).*

*Projekt byl napsán pomocí Ruby on Rails a jazyka Erlang. GitHub poskytuje další služby.*

*Gist, jenž je součástí GitHubu, umožňuje verzování a rychlé sdílení kratších kódů.“[6]*

## Git repozitář

*„Git je distribuovaný systém správy verzí. Mezi jeho hlavní výhody patří snadné vytváření větví (branch) a jejich slučování (merge). Díky tomu, že má vývojář celý repozitář u sebe může commitovat, mergovat nebo vytvářet nové větve i když je offline. Git má oproti jiným SCM nástrojům výhodu i v tzv staging area neboli indexu. Po přidání souborů si může vývojář znovu projít nové nebo změněné soubory než provede commit. Git byl původně vytvořen Linusem Torvaldsem pro vývoj jádra Linux, dnes je spravován Junion Hamanem. Git je svobodný softare šířený pod GPL licencí verze 2.“[7]*

# Licenční politika

Prroky/css-fw je k dispozici jako volně šiřitelný software a může ho používat kdokoliv. Prroky/css-fw používá The MIT licenci.

Licence MIT je doporučena pro většinu projektů, jelikož je snadné ji pochopit a neklade téměř žádná omezení na to, co můžete s frameworkem dělat. Můžete jej používat třeba i v komerčních projektech.

## Co je to svobodný software

*„Svobodný software (anglicky free software) je počítačový program, který zaručuje uživatelům svobodu jej spouštět, kopírovat, distribuovat, studovat, měnit a zlepšovat. Přizpůsobovat ho svým potřebám. Opravovat chyby, přidávat do něj své vlastní nápady a poté tato vylepšení zveřejňovat ostatním.*

*Máte tedy volnou ruku, nemusíte nikoho žádat o povolení, nebo za něj platit.*

*Aby bylo technicky možné program měnit, je třeba mít přístup k jeho zdrojovému kódu. Proto svobodný software musí být vždy Open source. Svobodný software je podmnožinou škatulky Open source.“ [8]*

## MIT licence

*„The MIT License (MIT)*

*Copyright (c) 2016 Prroky*

*Permission is hereby granted, free of charge, to any person obtaining a copy*

*of this software and associated documentation files (the "Software"), to deal*

*in the Software without restriction, including without limitation the rights*

*to use, copy, modify, merge, publish, distribute, sublicense, and/or sell*

*copies of the Software, and to permit persons to whom the Software is*

*furnished to do so, subject to the following conditions:*

*The above copyright notice and this permission notice shall be included in all*

*copies or substantial portions of the Software.*

*THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR*

*IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,*

*FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE*

*AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER*

*LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,*

*OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE*

*SOFTWARE.[9]*

# Závěr

Při této práci jsem dospěl k tomu, že Framework je opravdu velmi užitečná věc a velké usnadnění veškeré práce nejen pro programátory ale také pro nováčky, a že při použití frameworku se uživatel může soustředit pouze na svou práci.

Tato práce mi přinesla mnoho zkušeností v oblasti webdesignu a také v oblasti programování.

Můj vytvořený Framework rozhodně najde uplatnění v menších například školních projektech, ale je možno ho využít i v nějakých menších projektech. Můj Framework bude volně k dispozici a bude volně šiřitelný.

Tento projekt je vložený na stránce <https://github.com/Prroky/css-fw>. Na této stránce si ho každý může stáhnout a použít pro své vlastní účely.

# Seznam použité literatury

[1] jQuery. Wikipedia. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://cs.wikipedia.org/wiki/JQuery>

[2] Bootstrap. Wikipedia. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>

[3] Foundation framework. Wikipedia. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://en.wikipedia.org/wiki/Foundation_(framework)>

[4] Nette Framework. Wikipedia. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://cs.wikipedia.org/wiki/Nette_Framework>

[5]CSS Bootstrap. Bootstrap. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://getbootstrap.com/css/#grid-intro>

[6] GitHub. Wikipedia. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://cs.wikipedia.org/wiki/GitHub>

[7] Git. Wikipedia. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <http://geo.fsv.cvut.cz/gwiki/Git>

[8] Licenční politika. Nette Framework. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://nette.org/cs/license>

[9] The MIT License (MIT). Open Source Initiative. [online]. 27.1.2016 [cit. 2016-01-27]. Dostupné z: <https://opensource.org/licenses/MIT>