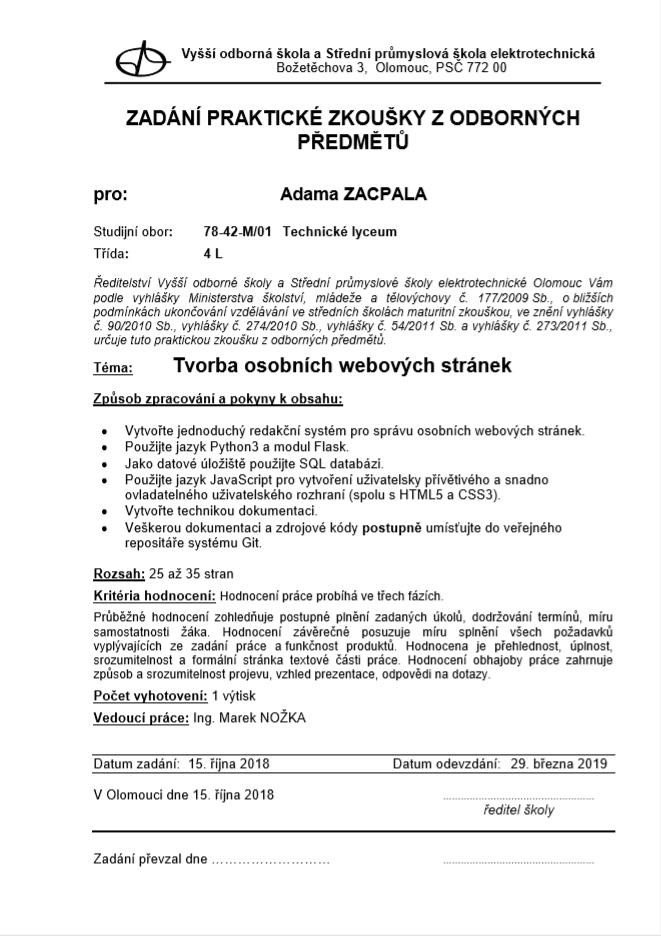
**Vyšší odborná škola a Střední průmyslová škola elektrotechnická Olomouc,**

**Božetěchova 3**

MATURITNÍ PRÁCE

Tvorba osobních webových stránek

2019 Adam ZACPAL

Prohlašuji, že jsem seminární práci vypracoval samostatně a všechny prameny jsem uvedl v seznamu použité literatury.

……………………………

jméno a příjmení studenta

Chtěl bych vyslovit poděkování panu Ing. Marku Nožkovi za odborné konzultace a poskytnuté informace.

……………………………

jméno a příjmení studenta

Prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé práce nebo její části se souhlasem školy.

……………………………

jméno a příjmení studenta

# Obsah

Obsah 4

Abstrakt 6

Úvod 7

1. Obecné informace 8

2. World wide web (www) 9

2.1 Vznik WWW 9

2.2 Normy 10

2.3 HTTP A HTTPS 10

3. HTML 11

3.1 Historie HTML 11

3.2 Verze HTML 11

3.3 Charakteristika HTML 13

3.4 Struktura dokumentu 13

4. CSS 15

4.1 Historie CSS 15

4.2 CSS 1 15

4.2.1 Jednotky a barvy 15

4.2.2 URL 15

4.2.3 Písmo 16

4.3 CSS 2 16

4.4 CSS 3 16

4.5 CSS syntax 17

4.5.1 Selektory 18

4.5.2 Blok deklarací 18

4.5.3 Propojení HTML s CSS 18

5. Python a Flask 19

5.1 Verze 19

5.2 Vlastnosti 19

5.3 Python syntaxe 20

5.3.1 Proměnné 20

5.3.2 Komentáře (poznámky) 20

20

5.3.3 Funkce 21

5.3.4 Moduly 21

5.4 Flask 21

6. JavaScript 22

6.1 Využití 22

6.2 Charakteristika 22

7. Systémy řízení báze dat 24

7.1 Schopnosti 24

7.2 MySQL 24

7.2.1 Vlastnosti MySQL 24

7.2.2 Příkazy 25

8. Git 26

8.1 GitHub 26

9. Finální produkt 27

9.1 Postup tvoření práce 27

9.1.1 Kontakt 27

9.1.2 Registrace a přihlášení 28

9.1.3 Redakční systém 28

9.2 Obsah stránek 29

9.2.1 Úvodní (domovská) stránka 29

9.2.2 O mně 30

9.2.3 Články 31

9.2.4 Vzorový článek 31

9.2.5 Kontakt 32

9.2.6 Redakce 33

9.3 Struktura složky 33

Závěr 34

Seznam použité literatury 35

Seznam tabulek a obrázků 36

# Abstrakt

Seminární práce je složena z teoretické a praktické části. V teoretické části se rozebírám použité programovací jazyky, jejich historii, vývoj a použití. Dále uvádím informace o World Wide Web, systémech řízení báze dat a Gitu. Na konci představuji webové stránky, které jsem sám vytvořil. Popisuji zde postup tvoření těchto stránek a jejich obsah.

V praktické části se zabývám samotnou tvorbou stránek. Použil jsem Python3 a jeho modul Flask, HTML, CSS, MySQL a JavaScript. Dále jsem celou práci postupně dával na GitHub.

Klíčová slova: Osobní webové stránky, webové stránky, blog, Python, Flask, HTML, CSS, JavaScript, MySQL, GitHub, informační technologie, databáze

# Úvod

Jako svou maturitní práci jsem si zvolil tvorbu osobních webových stránek. Hlavním cílem mé maturitní práce je vytvořit jednoduchý redakční systém pro správu osobních webových stránek, použít jazyk Python3 a modul Flask, jako datové úložiště použít SQL databázi, použít jazyky JavaScript, HTML5 a CSS3 pro vytvoření uživatelsky přívětivého a snadno ovladatelného rozhraní, vytvořit technickou dokumentaci a veškerou dokumentaci a zdrojové kódy postupně umisťovat do veřejného repositáře Git.

# Obecné informace

Webové stránky jsou počítačové dokumenty, které je možné zobrazit na počítači, notebooku či jiném zařízení pomocí webového prohlížeče. Webové stránky jsou poskytovány v rámci World Wide Webu. Informace jsou zobrazovány ve formě hypertextu, který je vytvořen použitím jazyku HTML. Stránky mohou obsahovat text, obrázky, videa, zvuky, gify, odkazy a tak dále. Pomocí odkazů můžete přecházet mezi různými webovými stránkami.

Webové stránky poskytují webové servery prostřednictvím počítačové sítě nebo Internetu, kde jsou přenášeny pomocí protokolu http nebo mohou být uloženy v podobě souborů na pevném disku. Webové stránky mohou být statické což znamená, že obsahují stále stejný obsah uložený v souborech nebo dynamické, které mění svůj obsah v čase, kdy je uživatel používá.

Aby byly webové stránky opravdu úspěšné, musí se dodržet několika kritérií. Záleží na kvalitě grafického a technického zpracování, informací obsažených na webu a hlavně funkčnosti všech komponentů.

# World wide web (www)

World Wide Web je systém pro ukládání, prohlížení a odkazování dokumentů(webových stránek) v Internetu. Dokumenty si jsou uloženy na webových serverech, jsou propojeny pomocí hypertextových odkazů ve formě URL.

Obrázek První web server (počítač NeXT)

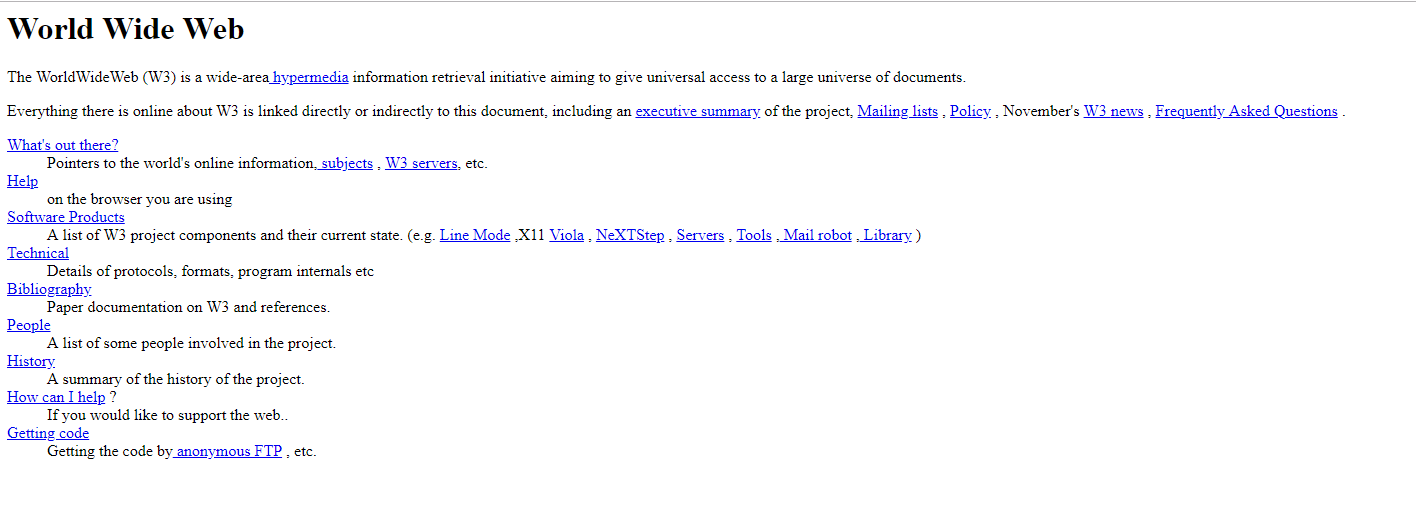
Zdroj: <https://en.wikipedia.org/wiki/CERN_httpd>

## Vznik WWW

Web založil Tim Berners-Lee při svém působení v CERNu. Navrhl jazyk HTML a protokol HTTP, napsal první webový prohlížeč WorldWideWeb a v roce 1990 spustil první webový server na světě info.cern.ch. V říjnu roku 1994 založil World Wide Web Consortium (W3C), které dohlíží na další vývoj Webu.

Vize Tima Berners-Lee o globální propojené síti informací se stala možnou v druhé polovině osmdesátých let. Začala se rozšiřovat po Evropě a začal fungovat Domain Name System, systém názvů domén, na kterém je založen systém URL. V roce 1988 proběhlo první přímé IP spojení mezi Evropou a Severní Amerikou a Berners-Lee začal otevřenou diskuzi v CERNu o možnosti vzniku webového systému.

S pomocí kolegy a podobného nadšence do hypertextu Roberta Cailliau vydal návrh 12. listopadu 1990 s názvem "WorldWideWeb“ jako síť hypertextových dokumentů, která by mohla být prohlížena "prohlížeči" v podobě architektury typu klient-server. V tu doby byly HTML a HTTP ve vývoji pouze dva měsíce a první webový server byl poprvé otestován až o měsíc později.

Tento návrh předpokládal, že síť pouze pro čtení by mohla být vyvinuta během tří měsíců a za šest měsíců by se mohlo dosáhnout bodu, kdy bude možnost tvořit nové odkazy a nové materiály i čtenář. Zatímco cíl sítě pouze pro čtení byl dosažen včas, tak přístupné autorství webového obsahu trvalo zavést mnohem déle.

Obrázek První web http://info.cern.ch/

Zdroj: <http://info.cern.ch/>

## Normy

Operace World Wide Webu, Internetu a serverů pro výměnu informací definuje mnoho formálních norem a technických specifikací. Velká část těchto dokumentů je vytvořeno World Wide Web Consortium (W3C), jiné jsou od Internet Engineering Task Force a dalších organizací.

* Doporučení pro značkovací jazyk, hlavně HTML a XHTML od W3C. Definují strukturu a interpretaci hypertextových dokumentů.
* Doporučení pro styly, především CSS od W3C.
* Normy pro ECMAScript (obvykle ve formě JavaScriptu) od Ecma International.
* Doporučení pro Document Object Model od W3C.

## HTTP A HTTPS

HTTP protokol základem pro fungování World Wide Webu. Přidáním šifrovací vrstvy vznikl HTTPS protokol, který je vhodný pro přenos důvěrných informací v počítačové síti, jako jsou hesla či bankovní údaje, které mají být přeneseny přes veřejný internet.  Je využíván především pro komunikaci webového prohlížeče s webovým serverem. Všeobecně se doporučuje podporovat HTTPS místo nezabezpečeného HTTP pro všechny webové stránky.

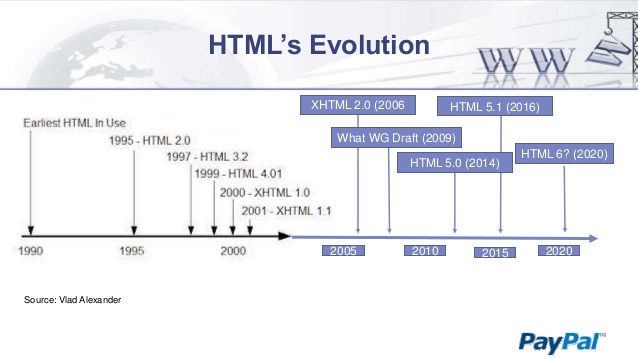
# HTML

HTML, tedy Hypertext Markup Language, v češtině hypertextový značkovací jazyk, je v informatice název značkovacího jazyka, pomocí kterého se tvoří webové stránky.

## Historie HTML

V roce 1989 pracoval Tim Berners-Lee a Robert Cailliau na propojeném informačním systému pro CERN. Pro tvorbu dokumentů se v té době používaly jazyky TeX, PostScript a SGML, které ovšem byly poměrně složité. V roce 1990 byl navržen Hypertext Markup Language, který se vyvinul ze SGML (Standart Generalized Markup Language). V roce 1993 vznikl první prohlížeč s grafickým rozhraním Mosaic. Objevil se v počítačích IMB PC a Macintosh a zaznamenal obrovský úspěch. Kvůli rychlému rozmachu, bylo pro HTML nutné definovat standardy.

## Verze HTML

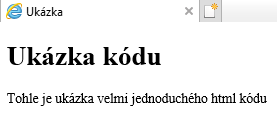
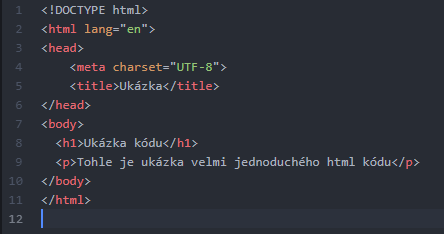
Vývoj HTML probíhá dvěma způsoby – podle specifikací nebo podle toho co umějí prohlížeče. Ideální by teoreticky bylo, kdyby se mohlo HTML vyvíjet podle specifikací, ale ve skutečnosti se HTML rozvíjí podle toho, co umí prohlížeče.

Obrázek Časová osa vývoje HTML

Zdroj: <https://www.slideshare.net/daniel_b_austin/perspectives-on-the-evolution-of-html>

* HTML 1
  + Bylo v něm pouze několik nejdůležitějších značek, které se většinově používají dodnes.
* HTML 2.0
  + Původní HTML už nestačil potřebám komunikace, a tak byl rozšířen samotnými výrobci prohlížečů. Přidává navíc také formuláře a podporuje grafiku.
* HTML 3.0
  + Příliš ambiciózní na svou dobu. Propadla v zapomnění, protože neměla podporu prohlížečů.
* HTML 3.2
  + Osekaná verze 3.0. Přidává tabulky, zarovnání textu a stylové prvky. Tato verze vůbec nepočítá s CSS.
* HTML 4.0
  + Snažila se výrazně zjednodušovat a brzy byla nahrazena verzí 4.01.
* HTML 4.01
  + Přinesla několik nových tagů (div, span, pokročilejší tabulky) a jasně stanovila oddělení smyslu a vzhledu. Plně počítala s kaskádovými styly (CSS) a formátování prvků v HTML za zastaralé. Deklarovala, že se HTML už nebude vyvíjet a nahradí jej XHTML.
* XHTML 1.0
  + Byla pouhým přenesením značek HTML 4.01 do standardů XML. To znamená uzavírat tagy, psát tagy malými písmeny, povinné hodnoty atributů a podobně.
* XHTML 1.1
  + Definitivně vycházela z předchozí verze, došlo k zakázání a zavržení určitých atributů. Nebyl moc používaný.
* XHTML 2.0
  + Rušily se některé zaběhlé tagy a vytvářely se nové. Tato verze nebyla v praxi implementována žádnými hlavními prohlížeči.
* HTML 5
  + Používá se dodnes, opravuje chyby svých předchůdců, odstraňuje mnoho nepoužívaných, zastaralých prvků a přidává nové sémantické prvky. Podporuje mnoho moderních technologií a vytváří nový systém pro vývoj jazyka.

## Charakteristika HTML

HTML používá značky (tagy), které mají určité vlastnosti. Do značek se píšou části textu a tím se určuje jeho význam. Značky se zapisují mezi úhlové závorky. Většinou jsou párové a mohou se různě vkládat do sebe. Počáteční značka je shodná s koncovou s tím rozdílem, že před koncovou se píše lomítko.

Obrázek Ukázka toho, jak vypadá kód z předchozího obrázku v prohlížeči

Značky často obsahují informaci, která značce přidá určitou vlastnost. Můžou to být třídy, odkazy, titulky a tak dále. HTML kód se většinou píše v programech, které se nazývají editory. Můžete psát prakticky v jakémkoliv programu pracujícím s textem. Ve skutečnosti se však používají editory, které dokáží barevně rozlišit části HTML kódu a doplňovat části značek.

Obrázek Ukázka jednoduchého HTML kódu

## Struktura dokumentu

Základní vymezující značky oblastí HTML souborů mají předepsanou strukturu.

* !doctype
  + Píše se na úplný začátek souboru a deklaruje typ dokumentu. „<!doctype html>“ říká prohlížeči, že otevřel HTML dokument.
* html
  + Značka „html“ začíná a končí a reprezentuje celý HTML dokument. Veškerý obsah tudíž musí být uvnitř. Značku si většina prohlížečů domyslí, takže není povinná, ale čistě pro pořádek je dobré ji používat. Jedná se o kořenový dokument a nemá žádné atributy.
* head
  + Je to hlavička dokumentu, která se nezobrazuje. Obsahuje nepovinně metadata vztahující se k celému dokumentu. Definuje například popis, jméno autora, titulek, kaskádové styly nebo kódování.
* body
  + Body, neboli tělo dokumentu obsahuje veškerý zobrazovaný obsah stránky.

# CSS

Kaskádové styly (Cascading Style Sheets) je jazyk pro stylování elementů zobrazených na stránkách napsaných v jazyku HTML. Hlavním účelem CSS je oddělit vzhled HTML dokumentu od jeho obsahu a struktury.

## Historie CSS

CSS navrhl v roce 1994 Hakon Wium Lie. Lie v té době pracoval v CERNu s Timem Berners-Lee a nebyl jediný, který navrhl podobný stylovací jazyk. World Wide Web Consortium (W3C) doporučilo právě CSS, a tak bylo vypuštěno v roce 1996 CSS1. Styly existovali už od začátku SGML v osmdesátých letech. Tyto styly (DSSS, FOSI a pod.) již nebyly pro web vhodné. CSS na rozdíl od nich nabídl, aby byl styl dokumentu ovlivněn více šablonami stylů ve formě kaskád.

Jak HTML rostlo, muselo zahrnout širší paletu stylistických schopností a splnit požadavky webových vývojářů. Tento vývoj dal projektantům větší kontrolu nad vzhledem stránek za cenu složitějšího HTML kódu. Snahou komunity bylo oddělit informace o obsahu od formátovacích pokynů. A tak vznikly CSS. První velmi rozšířený prohlížeč, který trochu podporoval CSS, byl Internet Explorer verze 3 z roku asi 1996.

## CSS 1

Byla to první CSS specifikace, která dostala podporu W3C. Hakon Wium Lie a Bert Bos jsou uznáváni jako originální vývojáři. Hypertextové dokumenty dovolily CSS 1 přizpůsobení těchto vlastností: vlastnosti písma, barvy textu a pozadí, vlastnosti textu, vlastnosti blokových elementů, způsoby zobrazení prvků a řízení pozice.

### Jednotky a barvy

Mezi jeho jednotky patří: pt, px, pc, cm, mm, in, em a ex. Barvy se daly zapsat buď číselně „#rrggbb“, kde r, g, b jsou barevné složky 00 až ff (16,8 milionů barev) nebo „#rgb“, kde mohou hodnoty r, g, b nabývat hodnot pouze 0 až f (4096 barev) nebo „rgb“, kde jsou hodnoty zadávány v desítkové soustavě 0 až 255 nebo nejjednodušší formou, pomocí názvů pojmenovaných barev např. black, red, green, blue, a mnoho dalších.

### URL

V CSS se ke specifikaci URL používá „url()“, mezi závorky se vloží adresa zdroje. Ta může být absolutní např. url(http://www.server.cz/img/logo.gif), relativní vůči aktuálnímu adresáři např. url(img/logo.gif) nebo relativní vůči serveru např. url(/img/logo.gif). Před závorky, čárky, uvozovky a mezery musíme dát zpětné lomítko.

### Písmo

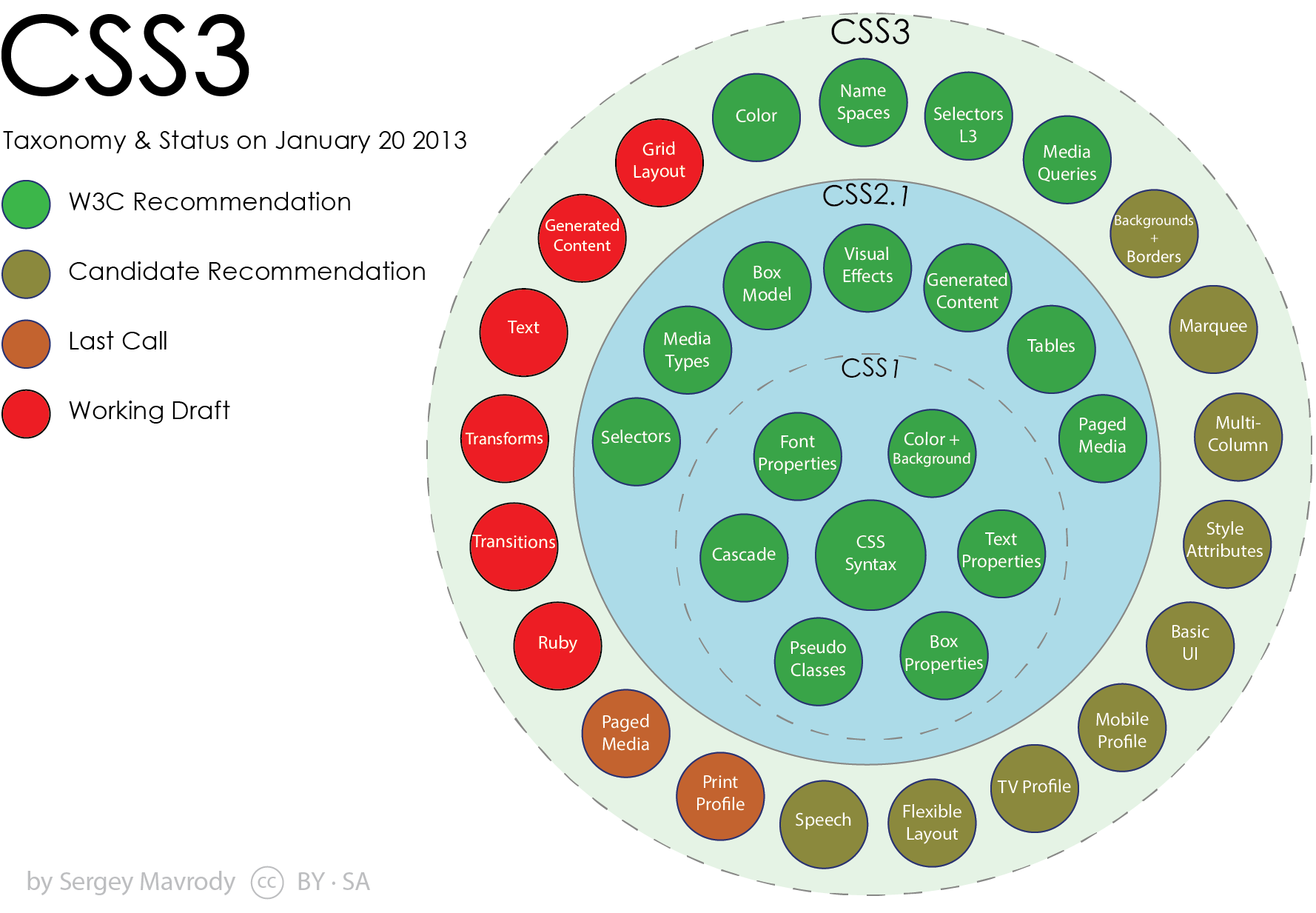
Druhy písma zadáváme jako seznam, ve kterém jsou názvy písem odděleny čárkou. Prohlížeče při načítání stránky tento seznam postupně prochází a pokud počítač uživatele nepodporuje první z nich, pokračuje v seznamu dál. Na posledním místě by měl vždy být uveden obecný název písma (serif, sans-serif, cursive, fantasy nebo monospace). Písma také mají své atributy, které dokáží změnit velikost, tučnost, řez, dekoraci a tak dále.

## CSS 2

Specifikace CSS 2 byla vyvinuta W3C a publikována jako doporučení v květnu roku 1998. Konečná verze se ustanovila kolem roku 2000. Rozšířila možnosti stylování dalšími parametry (vnější ohraničení, kurzor nad elementem, styl citací, nastavitelný obsah elementů, minimální a maximální šířka nebo výška elementu atd.)

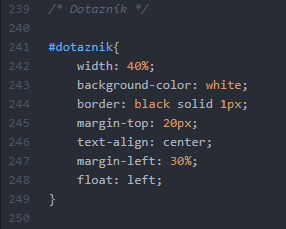
## CSS 3

CSS 3 je aktuální verze CSS. Na rozdíl od CSS 2, který je jedna velká specifikace obsahující rozdílné vlastnosti je CSS 3 rozdělen do několika rozdělených modulů. Každý modul přináší nové schopnosti nebo rozšiřuje schopnosti z CSS 2. Kvůli modularizaci mají jednotlivé moduly rozdílné stavy a stabilitu. Moduly mají různé statusy. Buď jsou doporučené (Recommendation), kandidující na doporučení (Candidate Recommendation), pracovní návrhy (Working Draft) nebo poslední výzva (Last Call). Přináší následující možnosti: animace, flexibilní blokové elementy, dodatečné možnosti stylování pozadí blokových elementů, mřížky a mnoho dalších.

 Zdroj: <https://commons.wikimedia.org/wiki/File:CSS3-Status-by-Sergey-Mavrody.png>

Obrázek Moduly CSS 3

## CSS syntax

CSS se zapisují zcela jiným způsobem než HTML. Kaskádové styly se tvoří z několika pravidel. Každé pravidlo se skládá s jednoho nebo více selektorů a bloku deklarací.

Obrázek Ukázka CSS pravidla

### Selektory

Selektory v CSS určují, která část značkovacího stylu se vztahují k odpovídajícím značkám. Selektory můžeme používat u následujících elementů: všechny elementy specifického typu (např. nadpis druhé úrovně h2), id (jedinečný identifikátor v dokumentu), třídy (identifikátor, který může v dokumentu označit více prvků). Třídy a ID rozlišují malá a velká písmena, musí začínat písmenem a můžou obsahovat alfanumerické znaky, podtržítka a pomlčky.

### Blok deklarací

Blok deklarací obsahuje seznam deklarací ve složených závorkách. Každá deklarace se skládá z identifikátoru vlastnosti, dvojtečky, hodnoty a je zakončená středníkem k oddělení od dalších deklarací. Vlastnosti a hodnoty jsou specifikovány ve standardu CSS. Příklad deklarace: „text-align: center;“.

### Propojení HTML s CSS

Propojit HTML a CSS lze několika způsoby. V praxi se nejčastěji používá odkaz na externí soubor, který se vkládá do hlavičky.

Ukázka: <link rel="stylesheet" href="/cesta/k/souboru.css" type=“text/css“>

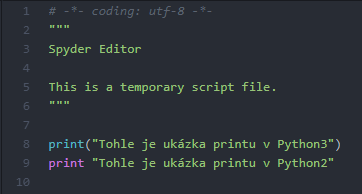
# Python a Flask

Python navrhl v roce 1991 Guido van Rossum. Je to vysokoúrovňový skriptovací programovací jazyk, který nabízí dynamickou kontrolu datových typů a podporuje různá programovací paradigmata (objektově orientované, imperativní, funkcionální, procedurální). Je navržen v jazyce C a probíhá v něm další vývoj. Python je vyvíjen jako open source projekt (otevřený software) a nabízí instalační balíky zdarma pro většinu platforem.

## Verze

V současné době se používají dvě verze Pythonu, které nejsou kompatibilní Python 2 a Python 3. Python 3 vydaný v roce 2008 odstraňuje nedostatky a chybné návrhy Pythonu 2, které nešlo odstranit při zachování kompatibility. Po vydání Pythonu 3 byl Python 2 stále vyvíjen a přidávaly se do něj vlastnosti z verze 3. Přechod na novou verzi probíhá dodnes, kvůli pomalému přenosu knihoven třetích stran a programů. Ve většině distribucí GNU/Linux systémů, kde je Python součástí základní instalace jsou běžně nainstalované obě verze Pythonu. Podpora Pythonu 2 má být ze strany vývojářů ukončena v roce 2020.

Obrázek Ukázka rozdílu v printu Python3 a Python2



## Vlastnosti

Python je hybridní jazyk. Znamená to, že dovoluje používat objektově orientované, imperativní, funkcionální a procedurální paradigma při psaní programů, podle toho, co se pro daný problém hodí nejlépe.

Mezi jeho kladné vlastnosti patří také jednoduchost jeho učení. Je dokonce považován za jeden z nejlepších programovacích jazyků pro začátečníky. Inspiroval se jazykem ABC, který byl přímo vytvořen pro výuku. Python je velmi efektivní v rychlosti psaní programů, která je velmi dobrá jak u jednodušších, tak i u rozsáhlých aplikací. S tím souvisí dobrá dostupnost a snadná použitelnost široké škály knihoven, které umožňují řešení úloh ze spousty oblastí.

Python však nemá jen kladné vlastnosti. Mezi jeho neduhy patří, že obsahuje lokální proměnné a nedokumentovatelné funkce, což je poněkud nebezpečné, protože mohou být zneužity ke spuštění příkazu v operačním systému.

## Python syntaxe

Syntaxe v Pythonu je jedna z nejpřirozenějších vůbec, to znamená že je velmi jednoduchá. Python používá komentáře, jednoduché příkazy, operátory, klíčová slova, podmínky, víceřádkové výrazy, funkce a moduly.

### Proměnné

Proměnná je něco, co se chápe jako pojmenovaný odkaz na objekt. Jméno proměnné je tak spojeno s jinak bezejmenným objektem. Například: „a = 1 + 2“. Při vytisknutí proměnné „a“ se pak zobrazí výsledek příkladu, který obsahuje. „print(a)“ tedy zobrazí číslici „3“. Tohle už se dá považovat za velmi jednoduchý příkaz.

### Komentáře (poznámky)

Komentáře jsou v Pythony velice jednoduché. Jediné co stačí udělat, je dát na libovolné místo do kódu mřížku a za ni napsat libovolný text. Před mřížkou může být napsán funkční program, vše co je na řádku za ní je však pouze poznámka, a i kdyby byl za mřížkou správně napsaný kód tak se neprovede. Další možností jsou víceřádkové komentáře. Ty používají tři uvozovky na začátku a na konci poznámky. Text poznámky se píše mezi ně a může mít libovolnou délku i počet řádků.

### 

Obrázek Ukázka komentářů (poznámek)

### Funkce

Funkce se chová jako běžný objekt dokud není zavolána. Definuje se příkazem „def“, za který se napíše její jméno, do závorky se napíše seznam argumentů a na konec dvojtečka. Poté následuje tělo funkce, které musí být odsazené a obsahuje příkazy, které funkce provádí. Příkazem „return“ se z funkce vrací hodnota. Tělo funkce může mít více příkazů, podmínek, cyklů a podobně.

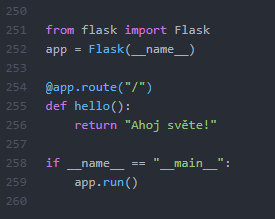
### Moduly

Většina složitějších kódů je děleno do tzv. modulů. Je to něco, z čeho importujeme různé funkce. Z modulu „math“ můžeme importovat například funkci „sqrt“. Import vypadá takto: „from math import sqrt“. Můžeme importovat také celý modul najednou („import math“). K jednotlivé funkci se potom můžeme dostat tak, že napíšeme název knihovny, tečku, za ni jméno funkce a do závorky její parametry („math.sqrt()“). Dají se také vytvářet vlastní moduly. Jediné co musíte udělat, je otevřít pythonovský soubor, napsat do něj funkce a potom soubor naimportovat do jiného.

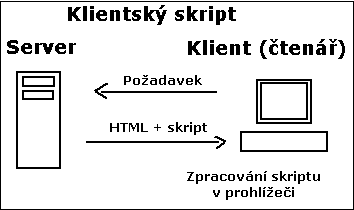
## Flask

Flask je webový micro Framework napsaný v Pythonu. Nemá žádnou vrstvu databáze, ověřování formulářů ani jiné součásti, které by poskytovaly funkce existujících knihoven třetích stran. Flask však podporuje rozšíření, která mohou přidávat funkce, jako kdyby byly implementované v samotném Flasku. Existují rozšíření pro ověřování formulářů, objektově-relační mapovače, zpracování uploadu, autentizační technologie a mnoho dalších. Rozšíření jsou aktualizována mnohem častěji než samotný program Flask. Webové aplikace, které používají Flask jsou například Pinterest a LinkedIn.

Obrázek Ukázka jednoduché webové aplikace



# JavaScript

JavaScript je programovací jazyk využívaný v internetových stránkách. Odpovídá specifikaci ECMAScript. Jeho velkou výhodou je, že se zapisuje přímo do HTML kódu. Je to objektově orientovaný, multiplatformní, událostmi řízený, skriptovací jazyk. JavaScript se vykonává na straně uživatele, což znamená, že je to klientský skript.

Zdroj: <https://www.jakpsatweb.cz/javascript/javascript-uvod.html>

Obrázek Schéma klientského skriptu

Patří do skupiny jazyků C/C++/Java, ačkoliv je JavaScript úplně jiný jazyk. ECMAScript je standardizovaná verze JavaScriptu a jsou z ní odvozovány i další implementace. Mezi výhody JavaScriptu patří pohodlnost a efektivnost bez zbytečných prodlev a načítání stránky.

## Využití

Pomocí JavaScriptu se dá měnit obsah webové stránky u uživatele. Můžeme tvořit dynamické menu, roletky a další kontejnery umožňující ušetřit místo na stránce, když nejsou používány a otevřou se po najetí myší. JavaScript je dále dobrý k formátování textu. Díky němu můžeme vkládat do napsaných zpráv například smajlíky. Používá se také na validaci webových formulářů. V JavaScriptu se díky příchodu HTML5 a podpory SVG dělají i hry jako například AngryBirds. Aplikace JavaScriptu jsou multiplatformní. To znamená, že fungují na jakémkoliv zařízení, které obsahuje webový prohlížeč (počítač, tablet, notebook, mobil, hodinky).

## Charakteristika

JavaScript se nemusí kompilovat (interpretovaný jazyk), využívá objektů prohlížeče a zabudovaných objektů (objektový jazyk), je závislý na prohlížeči, záleží mu na velikosti písem (case senzitivní jazyk) a je syntakticky podobný jazykům C a Java.

Funguje pouze v prohlížeči a uživatel může JavaScript zakázat. Vyskytují se zde časté chyby, kvůli existenci několika verzí jazyka i prohlížečů. Neumí přistupovat k souborům, k žádným systémových objektům a neumí uložit žádná data kromě cookies.

JavaScript je překládaný za běhu a je vykonáván dle vlastního zdrojového kódu. Jazyk je dynamicky typovaný. Obsahuje pouze jeden typ pro text (string), numerický typ (number), pravdivostní hodnotu (boolean) a pro cokoliv jiného využívá „Object“. JavaScript používá funkcionální paradigma. Tohle mu umožňuje do běžné proměnné uložit funkci. Funguje zde prototypová dědičnost, kdy je objekt předlohou pro jiný objekt.

# Systémy řízení báze dat

Systém řízení báze dat (SŘBD) nebo anglicky Database management systém (DBMS) je software, který zajišťuje práci s databází. Tvoří rozhraní mezi aplikačními programy a uloženými daty. SŘBD se často zaměňuje za databázový systém, který je však SŘBD dohromady s bází dat.

Následující seznam obsahuje některé SŘBD:

* CSQL
* MySQL
* DB2
* Firebird
* Microsoft Acces
* Oracle
* PostgreSQL
* SQLite

## Schopnosti

Programový systém, který je označen jako SŘBD musí efektivně pracovat s velkým množstvím dat a musí být schopen řídit (vkládat, modifikovat, mazat) a definovat strukturu těchto dat. Současné databázové systémy mají několik dalších vlastností. Podporují definici datových modelů, spravují klíče, transakce, jsou robustní a zotavují se po chybách bez ztráty dat a tak dále.

## MySQL

MySQL je systém řízení báze dat (Database management systém), který uplatňuje relační databázový model (založený na tabulkách), vytvořený firmou MySQL AB ze Švédska, vlastněný firmou Oracle Corporation. Mezi jeho hlavní autory patří Michael Widenius a David Axmark. MySQL je multiplatformní databáze a komunikace s ní se děje díky jazyku SQL.

### Vlastnosti MySQL

Je velmi oblíbená především pro svou snadnou implementovatelnost, výkon a díky tomu, že se jedná o volně šiřitelný software. MySQL je optimalizováno především na rychlost. To zapříčinilo, že má jen jednoduché způsoby zálohování, a do nedávna nepodporovalo ani pohledy, triggery a uložené procedury. Tyto vlastnosti jsou však doplňovány v posledních letech.

### Příkazy

Příkazy se dělí na čtyři základní skupiny:

* Manipulace s daty (SELECT, INSERT, UPDATE, DELETE, …)
* Definice dat (CREATE, ALTER, DROP, …)
* Řízení přístupových práv (GRANT, REVOKE)
* Řízení transakcí (START TRANSACTION, COMMIT, ROLLBACK)

# Git

Jedním z úkolů mé maturitní práce je postupně nahrávat dokumentaci a zdrojové kódy do veřejného repositáře Git. Git je distribuovaný systém správy verzí, který vytvořil Linus Torvalds a Junio Hamano pro vývoj jádra Linuxu. V současnosti je Git používaný v řadě projektů a jedná se o „free software“. Zaznamenává změny souboru, nebo sady souborů v čase a uživatel tak může kdykoliv obnovit jakoukoliv jeho verzi.

## GitHub

GitHub je webová hostingová služba pro správu verzí pomocí Gitu. Většinou se používá pro počítačové kódy. Nabízí všechny funkce Gitu pro distribuované řízení verzí a správu zdrojových kódů, ale také přidává vlastní. Poskytuje kontrolu přístupu a několik spolupracujících funkcí. GitHub nabízí plány pro podnikové, týmové, profesionální a bezplatné účty, které se běžně používají pro hostování open-source projektů. V červnu 2018 měl 28 milionů uživatelů, 57 milionů repozitářů (včetně 28 milionů veřejných), což jej dělá největším hostem zdrojových kódů na světě.

# Finální produkt

V této kapitole bych chtěl předvést finální podobu mé maturitní práce a její postup. Na mém osobním webu jsem použil převážně Python 3 a modul Flask, HTML, CSS a MySQL. Velkou část práce jsem dělal v editoru Atom.

## Postup tvoření práce

Postup mé práce byl hodně ovlivněn tím, že jsem po zadání uměl jen základy HTML, CSS a Pythonu, který jsem však nikdy dříve nepoužil při tvorbě webu a nedokázal jsem si dost dobře představit k čemu je.

Z toho důvodu jsem začal vzhledem, kterému jsem věnoval dost času a nyní to považuji za špatné rozhodnutí. Až potom co jsem měl navržený vzhled jsem začal pracovat s Pythonem a jeho modulem Flask. Na ploše jsem měl vytvořenou složku, do které jsem vkládal všechny soubory. Po vytvoření první webové aplikace, která obsahovala pouze cesty k jednotlivým stránkám, jsem v konzoli udělal virtuální prostředí pro její spuštění. Dále následovalo už jen přidávání a vylepšování dalších funkcí webu.

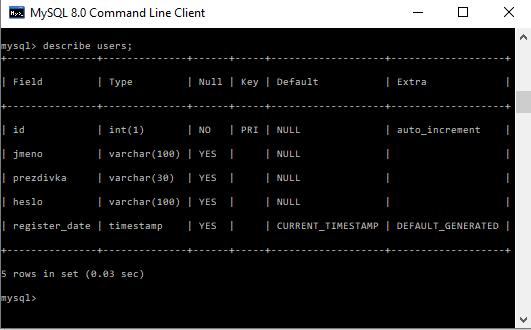
### Kontakt

Začal jsem kontaktním formulářem pro zaslání dotazu na můj e-mail. Narazil jsem na problémy, na které jsem nedokázal najít řešení. Za pomoci mého vedoucího práce se zjistilo, že nemám závažné chyby v kódu, ale v importu modulu pro Mail a v nastavení Gmailu. Díky tomuto zjištění se mi povedlo opravit kontaktní formulář během chvilky.



Obrázek Ukázka funkce pro kontaktní formulář

### Registrace a přihlášení

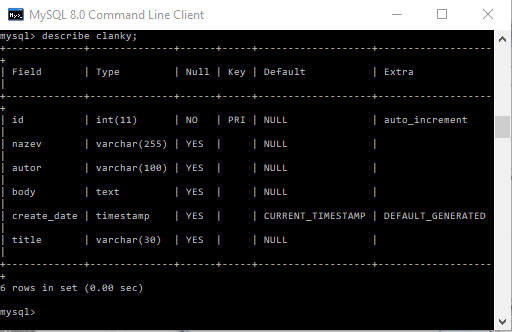
Mým úkolem bylo udělat také redakční systém pro mé webové stránky, tím pádem bylo přihlášení a registrace nezbytná věc. Pro tento úkol jsem si již musel vytvořit databázi a rozhodl jsem se pro MySQL. Stáhl jsem ji z internetu a nainstaloval podle návodu, který jsem si našel. Dále jsem MySQL používal přes příkazový řádek. Vytvořil jsem si databázi „myflaskapp“ a v ní tabulku „users“, která obsahovala id, jméno, přezdívku, heslo a datum registrace uživatele.

Obrázek Tabulka "users", vytvořená pro informace o uživateli

Poté jsem k databázi připojil mou webovou aplikaci v Pythonu a vytvořil funkce pro samotnou registraci a přihlášení. Dlouho jsem bojoval s rozšířením Flasku „flask-mysqldb“. Nepodařilo se mi přijít na důvod, proč mi nešlo toto rozšíření nainstalovat ani po dvoudenním několikahodinovém snažení a pátrání na internetu. Narazil jsem však na jiné rozšíření Flasku pro MySQL „flaskext.mysql“, které se mi podařilo nainstalovat a po změnění několika příkazů jsem došel do zdárného konce.

### Redakční systém

Když jsem již měl registraci a přihlášení mohl jsem se pustit do tvorby redakčního systémů pro přidávání jednotlivých článků na můj web. Nejdříve jsem si v již existující databázi „myflaskapp“ vytvořil tabulku pro články „clanky“ obsahující id, název, autora, tělo článku, datum vytvoření.

Poté jsem vytvořil funkci pro přidávání nových článků, což nebylo až tak složité, s vědomostmi, které jsem získal při tvorbě registrace a s databází, kterou jsem měl také vytvořenou. Později jsem přidal funkce pro upravení a smazání článku.

Obrázek Tabulka "clanky" obsahující informace o článku

## Obsah stránek

Můj web se pro uživatele skládá ze čtyř základních stránek: „Domů“, „O mně“, „Články“ a „Kontakt“. Každá z těchto stránek má hlavičku a patičku. V hlavičce je mé jméno a odkaz právě na tyto 4 stránky. V patičce je copyright a odkazy pro registraci a přihlášení. Administrátor nebo lépe řečeno autor má pak zpřístupněné stránky pro přidávání, úpravu a mazání článků na které se běžný uživatel nedostane. Jeho hlavička je stejná jako u běžného uživatele, avšak patička se liší. Místo odkazů pro registraci a přihlášení jsou zde odkazy pro odhlášení a redakční stránku.

### Úvodní (domovská) stránka

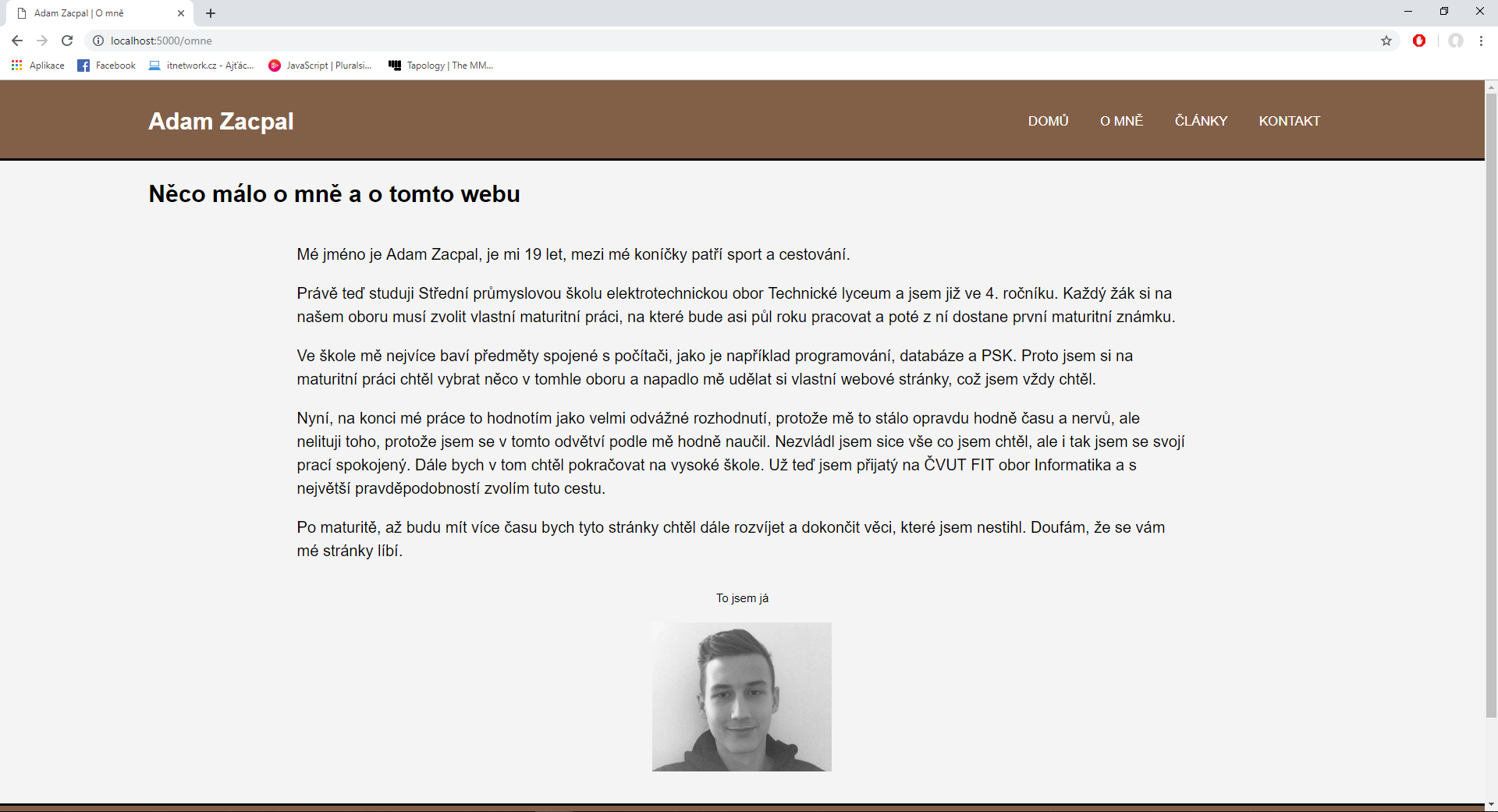
Domovská stránka je rozdělena na tři sekce. V první sekci se nachází úvodní okno obsahující fotky a text, ve kterém vítám uživatele a krátce popisuji, co na mém webu může najít. V prostřední sekci je připravený formulář pro odběr mých článků prostřednictvím emailu, který bych chtěl v budoucnu zprovoznit. Pod touto středovou sekcí se nachází poslední sekce s přednostně nabízenými články, které momentálně odkazují na jeden vzorový článek.

### O mně

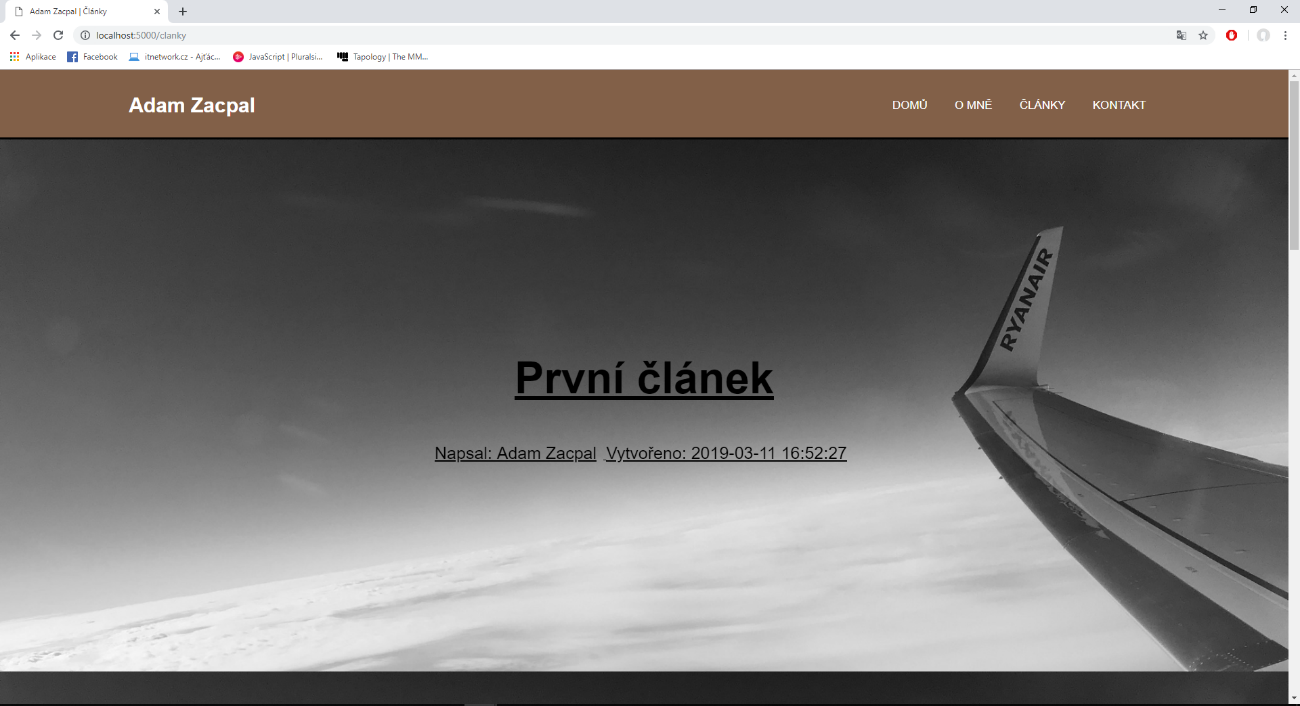
Obrázek Domovská stránka

Tato stránka logicky obsahuje informace o mně, ale i o samotném webu. Píšu v ní mimo jiné o tom, proč samotná stránka vznikla a jak s ní jsem spokojený. Úplně dole je má černobílá fotka, která se dá kliknutím otevřít ve větším okně a v barvě.

Obrázek Stránka "O mně"

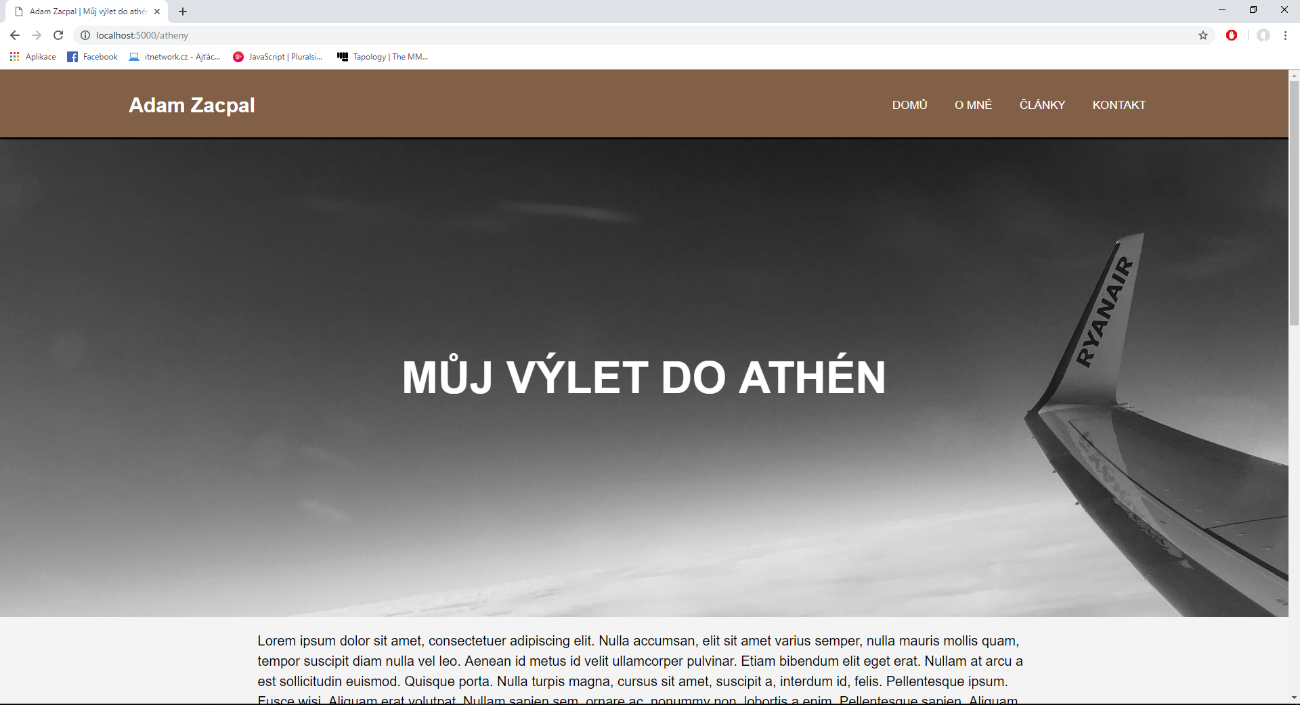


### Články

Stránka „Články“ obsahuje pouze úvodní fotku a název článku, na který se můžete prokliknout a jméno autora s datem a časem vytvoření. Tyto informace jsou v odstínu šedé a při přejetí myší kdekoliv po fotce zčernají.

### Vzorový článek

Obrázek Stránka "Články"

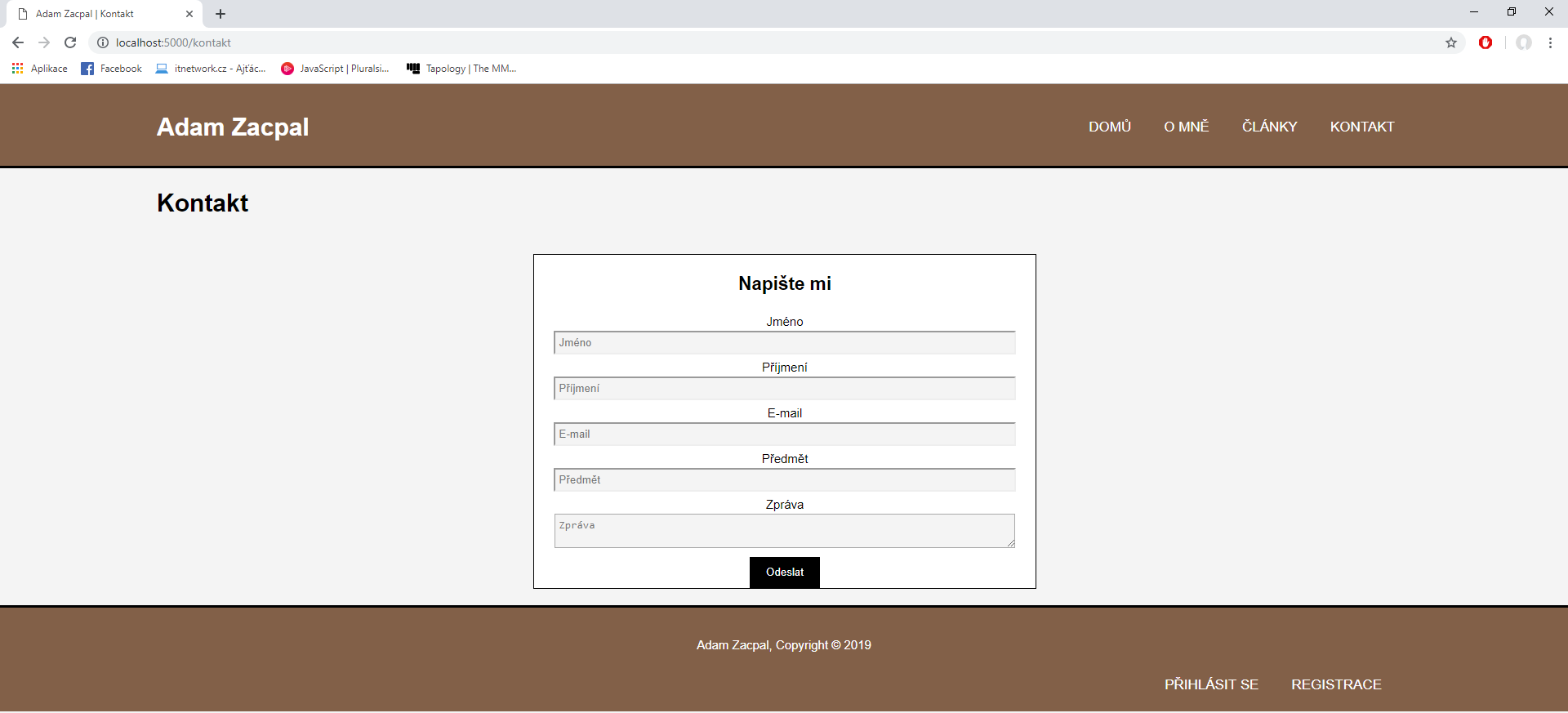
Vzorový článek se skládá ze tří částí. Nahoře je opět úvodní fotka s nadpisem tentokrát v bílé barvě, pod ním se nachází tělo samotného článku a úplně dole je galerie.

Obrázek Vzorový článek (horní část)

Galerie je dělaná pomocí JavaScriptu stejným způsobem jako fotka na stránce „O mně“. Fotografie jsou nejprve vidět v odstínech šedé, po přejetí myší se lehce zvětší a dostanou barvu a po kliknutí se otevřou ve velkém okně také v barvách.

### Kontakt

Obrázek Vzorový článek (galerie)

Na stránce „Kontakt“ se nachází pouze kontaktní formulář, který slouží k odeslání dotazu nebo připomínky ze strany uživatele na můj email.

Obrázek Kontaktní stránka

Tento formulář jsem použil i pro registraci, přihlášení, psaní a úpravu článků, proto o nich již nebudu psát a ukazovat je v dalších kapitolách, vypadají úplně stejně, jen mají jiné políčka pro vyplnění.

### Redakce

Na tuto stránku se nedostanete pokud nejste přihlášeni. Proto si můžete všimnout, že se v horní části ukazuje pod jakou přezdívkou jste přihlášeni a dole v patičce už nejsou odkazy k přihlášení a registraci, ale k odhlášení a právě ke vstupu na redakční stránku.

Pod ukazatelem uživatelského jména je tlačítko pro přidání článku. Když, na něj kliknete objeví se formulář jako je ten pro kontakt. Pod tímto tlačítkem je tabulka, která obsahuje informace o již publikovaných článcích a tlačítka pro úpravu a smazání. Tlačítko pro úpravu zobrazí stejný formulář jako tlačítko pro přidání článku s tím rozdílem, že políčka jsou již před vyplněna údaji daného článku.

## Struktura složky

Obrázek Redakce k úpravě, vytváření a mazání článků

Tento web jsem tvořil pomocí modulu Flask a k tomu se váže také struktura samotného projektu. Složka projektu obsahuje několik souborů a složek, které jsou umístěny tak, aby vše správně fungovalo.

Hlavní složku jsem pojmenoval „Maturitni-prace“. V ní se nachází soubory s konfigurací (config.py), samotná Python aplikace (startapp.py) a několik dalších složek. Složky env, static a templates. Env je virtuální prostředí, kde jsou všechny nainstalované balíčky a moduly. Složka static obsahuje všechny CSS a JavaScript soubory a další složku s obrázky. Ve složce templates jsou všechny jednotlivé html stránky.

# Závěr

Úkolem mé maturitní práce bylo vytvořit jednoduchý redakční systém pro správu osobních webových stránek, použít jazyk Python3 a modul Flask, jako datové úložiště použít SQL databázi, použít jazyky JavaScript, HTML5 a CSS3 pro vytvoření uživatelsky přívětivého a snadno ovladatelného rozhraní, vytvořit technickou dokumentaci a veškerou dokumentaci a zdrojové kódy postupně umisťovat do veřejného repositáře Git.

Tuto maturitní práci jsem si zvolil, protože jsem se chtěl již delší dobu v daném odvětví zdokonalovat, ale vždy jsem si našel něco „lepšího“ na práci. Vybráním takové práce mi tedy nezbylo nic jiného než se do toho pustit.

Vzhledem k tomu, že jsem při zadání měl jen základy HTML, CSS a Pythonu, který jsem však nikdy nepoužil pro tvorbu webu hodnotím výsledný produkt kladně. Nevěděl jsem do čeho jdu a učil jsem se za pochodu. Dle mého názoru jsem nezvolil nejlepší postup a strávil jsem příliš mnoho času vytvářením vzhledu. Spoustu věcí jsem několikrát předělával, aby byly přesně podle mých představ.

V budoucnu bych chtěl mé stránky vylepšovat a dostat se do bodu, kdy s nimi budu spokojený do takové úrovně, abych je nahrál na web.

Na závěr bych chtěl říct, že jsem se mnoho naučil a jsem spokojený s tím, jak to nakonec dopadlo.

# Seznam použité literatury

1. Informace o HTML a CSS: <https://www.jakpsatweb.cz/>
2. Různé informace: <https://www.wikipedia.org/>
3. MySQL: <https://www.itnetwork.cz/>, <https://dev.mysql.com/doc/>
4. Flask: <http://flask.pocoo.org/>
5. Python: <https://www.python.org/>

# Seznam tabulek a obrázků

Obrázek 1 První web server (počítač NeXT) 8

Obrázek 2 První web http://info.cern.ch/ 9

Obrázek 3 Časová osa vývoje HTML 11

Obrázek 4 Ukázka toho, jak vypadá kód z předchozího obrázku v prohlížeči 13

Obrázek 5 Ukázka jednoduchého HTML kódu 13

Obrázek 6 Moduly CSS 3 17

Obrázek 7Ukázka CSS pravidla 17

Obrázek 8Ukázka rozdílu v printu Python3 a Python2 19

Obrázek 9 Ukázka komentářů (poznámek) 20

Obrázek 10 Ukázka jednoduché webové aplikace 21

Obrázek 11 Ukázka funkce pro kontaktní formulář 25

Obrázek 12 Tabulka "users", vytvořená pro informace o uživateli 26

Obrázek 13 Tabulka "clanky" obsahující informace o článku 27

Obrázek 14 Domovská stránka 28

Obrázek 15 Stránka "O mně" 28

Obrázek 16 Stránka "Články" 29

Obrázek 17 Vzorový článek (horní část) 29

Obrázek 18 Vzorový článek (galerie) 30

Obrázek 19 Kontaktní stránka 30

Obrázek 20 Redakce k úpravě, vytváření a mazání článků 31