**Základní škola E. Nápravníka v Býšti**



**Jaroslav Němec**

**Osobní webové stránky**

**Závěrečná práce**

**Třída: IX.**

**Vedoucí práce: Mgr. Eva Němcová**

**2020/2021**

**Čestné prohlášení**

**Prohlašuji**, že jsem tuto závěrečnou práci vypracoval zcela samostatně a uvádím v ní veškeré prameny, které jsem použil.

*Jaroslav Němec*

**Poděkování**

Chtěl bych poděkovat Mgr. Evě Němcové za odborné vedení mé závěrečné práce, za její čas, cenné poznámky a připomínky, a také za možnost častých osobních konzultací.

**Obsah**

[Úvod 1](#_Toc64228070)

[1 Nástroje 2](#_Toc64228071)

[1.1 IDE (pracovní prostředí) 2](#_Toc64228072)

[1.2 Webový prohlížeč 4](#_Toc64228073)

[2 Technologie a služby 6](#_Toc64228074)

[2.1 Webová stránka 6](#_Toc64228075)

[2.2 Statická a dynamická stránka 6](#_Toc64228076)

[2.3 Webový server 7](#_Toc64228077)

[2.4 Hosting 7](#_Toc64228078)

[2.5 Domény 8](#_Toc64228079)

[2.6 FTP 8](#_Toc64228080)

[2.7 Gitové úložiště 9](#_Toc64228081)

[3 Jak jsem postupoval při tvorbě 11](#_Toc64228082)

[Závěr 14](#_Toc64228083)

[Zdroje 15](#_Toc64228084)

# Úvod

Důvod, proč jsem si jako téma závěrečné práce vybral právě „osobní webové stránky" je, že jsem už delší dobu chtěl zkusit vytvořit web. Poslední dobou se k tomuto chtění přidala další motivace, a to udělat si web pro prezentaci mých projektů. Potom, když se mě začali ptát, co budu dělat jako závěrečnou práci, byla možnost osobního webu jasná volba. Byl to též poslední krůček k tomu rozjet první webový projekt a zároveň získat mnoho užitečných zkušeností. V neposlední řadě se mi tato příležitost hodila pro získání základního obecného přehledu o technologiích a postupech, které se dnes používají, používaly a nebo se teprve používat začnou. Cíl mého webu je seznámit lidi s tím, co umím, co mě baví, co všechno dělám, na čem pracuji nebo, které projekty jsem již dokončil.

# Nástroje

V této části bych rád uvedl nástroje, se kterými jsem se seznámil, a které jsem použil pro svou tvorbu.

## IDE (pracovní prostředí)

Vývojová prostředí neboli IDE (integrované vývojové prostředí) jsou programy, jež slouží ke zjednodušení psaní kódu v jakémkoli programovacím jazyce. Tato IDE pomáhají kupříkladu s přehledností, strukturou kódu, nebo také s dodržováním takzvané „syntaxe" neboli pravidel, jak mají vypadat jednotlivé příkazy. Tato pravidla neboli syntaxi má každý jazyk vlastní. Bývá také pravidlem, že IDE od stejného výrobce mají stejné klávesové zkratky a design, kvůli lepší orientaci a větší efektivitě při přechodu na jiný programovací jazyk.

Mezi hlavní tvůrce IDE patří například JetBrains, NetBeans, Eclipse a Microsoft. Pro naše účely, tedy psaní webu, můžeme použít kupříkladu PhpStorm, WebStorm, PyCharm, IntelliJ IDEA od JetBrains, či VisualStudio od Microsoftu nebo NetBeans IDE od stejnojmenné společnosti.

Pro vytvoření jednoduchého webu však bohatě postačí obyčejný poznámkový blok, který máme každý ve svém počítači. Ovšem pokud potřebujeme vyvíjet větší, výkonnější a rozsáhlejší weby (dmsoftware.cz, YouTube.com nebo portál banky) budeme potřebovat nějaké lepší IDE. Ta se dělí na textové a vizuální.

Textové IDE je v podstatě vylepšený poznámkový blok (textový editor), který vám mimo výše uvedených výhod umožní celý program udělat spustitelným neboli zkompilovat (přeložit do strojového kódu - jazyka procesoru), a to bez dalších externích programů.

Vizuální nebo grafické IDE poskytuje oproti textovému jednu velkou výhodu a to, že místo psaní jen přesouváte tzv. bloky neboli dlaždice a dáváte je za sebe ve správném pořadí. Tyto bloky, jak již název napovídá, představují bloky (odstavce) kódu. Právě proto je tento typ IDE méně používaný, poněvadž jste omezeni tím, co už před vámi autor IDE napsal či vymyslel, a tudíž nemůžete funkci těchto bloků nijak ovlivnit.

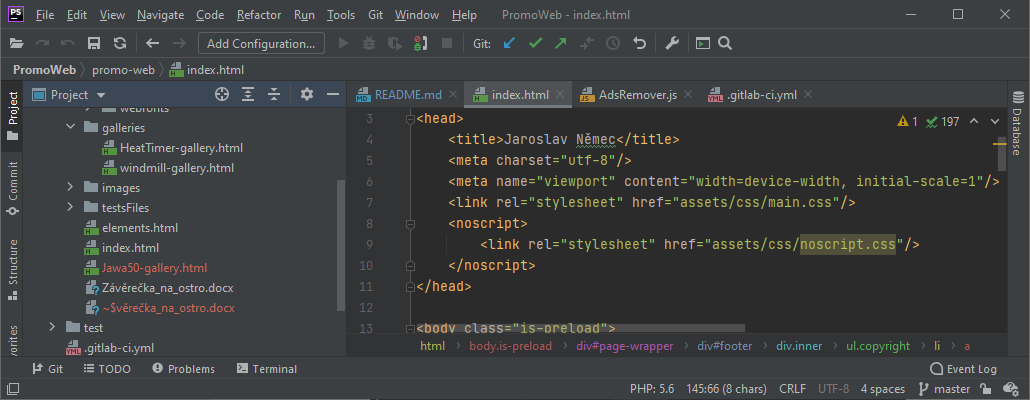
Vývojová prostředí disponují většinou také řadou nástrojů kupříkladu kompilátorem, debuggerem, průzkumníkem souborů, refaktorem, nápovědou. Mohou také obsahovat Gitového klienta nebo vlastní okno příkazového řádku.

Kompilátor (překladač) je nástroj, který používá programátor pro překlad svého kódu z vyššího programovacího jazyka (Java, Basic nebo Php) do nižšího (značkovacího) jazyka, kterému rozumí asembler, jenž celý kód ještě jednou přeloží tentokrát už do binární podoby, které rozumí procesor.

Debugger je další nástroj, který by neměl chybět v žádné pořádném IDE. Funguje pouze v rámci vývojového prostředí a dokáže zastavit daný program na Vámi určeném místě. Místo, kde chcete, aby se program zastavil, označíte takzvaným breakpointem (značkou, stopkou) a díky němu pozná debugger, kde má program pozastavit. Když se program zastaví, můžete si zobrazit názvy a hodnoty veškerých proměnných, či objektů v programu. Díky tomu si nemusíte vytvářet výpis ke každé hodnotě, kterou vyrobíte (nainicializujete). V podstatě jste na každém kroku programu schopni zjistit, jaký vliv na funkci má jakýkoli řádek v kódu.

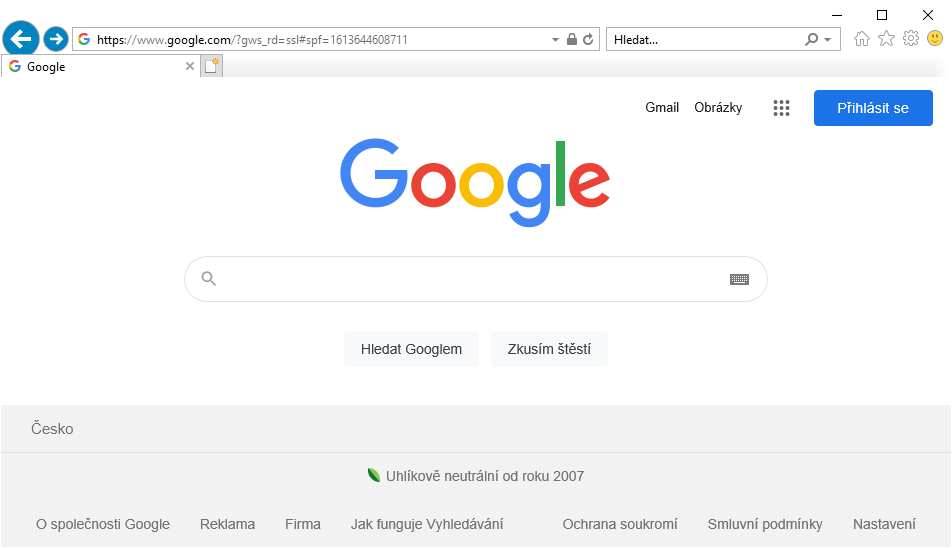
„Refaktor" se nazývá funkce, která třeba při přejmenovávání různých objektů nebo proměnných, dokáže najít všechna použití v kódu a všechny je naráz přejmenovat, abychom nemuseli chodit a každé použití proměnné či metody přepisovat ručně.

Nápověda je velice užitečná funkce, která při psaní kódu vypíše všechny možné varianty toho, co bychom mohli použít za příkaz. Je to dobré také na to, že si nemusíte pamatovat názvy všech objektů a proměnných, protože nám je to všechny vypíše. Rovněž se tato funkce hodí pro zrychlení psaní kódu, poněvadž místo psaní dlouhého názvu stačí jen vybrat z navrhované nabídky, potvrdit a název se sám doplní.

Gitový klient umožnuje připojit se a ukládat jednotlivé verze a jejich změny do cloudového úložiště Git. Pro připojení se do Gitu přes IDE, musíte nejdříve získat repositář, v němž budete pracovat. Takovýto repositář si buď založíte nebo si vyberete nějaký stávající, pak jen získáte URL adresu a vložíte ji do IDE. V případě, že je již v repositáři něco nahraného, klient tato data stáhne automaticky k vám. Dále klient obsahuje funkce jako je **commit**, **push** a kupříkladu **merge**. Funkce **commit** uloží změny ve verzi do lokálního Gitu (na Vašem počítači) a lze k tomuto takzvanému „commitu" přidat komentář, kupříkladu o tom, co je nového, jaké byly provedeny změny nebo co bylo opraveno či vylepšeno. Funkce **push** vezme commit z lokálního Gitu a nahraje jej do cloudu, odkud k němu mohou přistupovat i ostatní spolupracovníci na projektu. A nakonec pomocí funkce **merge** si takto v cloudu uloženou verzi může kolega stáhnout („mergnout") k sobě do IDE a získá tak stejný projekt (kód) jako já a může na něm pracovat. Poté, co by kolega udělal změnu nebo novou verzi a nahrál ji do Gitu, já bych dal jenom merge a klient sám porovná změny v Gitu oproti verzi projektu na mém počítači a postahuje jen změny, které následně zapojí do kódu. Tato funkce je velice výhodná převážně u větších projektů, protože se kvůli malé změně nemusí stahovat znovu celý projekt.

Vývojové prostředí - PhpStorm 2020.3.2

## Webový prohlížeč

Webový prohlížeč je program v počítači, který nám umožňuje zobrazovat si text, obrázky, videa a další součásti webových stránek. Webový prohlížeč funguje následujícím způsobem. Uživatel zadá požadavek do lišty internetového vyhledávače, prohlížeč vyšle požadavek pomocí internetové sítě na webový server, kde se požadavek zpracuje. Při zpracování se zjistí, zda je požadavek určen tomuto serveru, tudíž zda je server schopen poskytnout odpověď (data), nebo je-li potřeba ho přeposlat na další server, kde se zpracuje úplně stejně. Když konečně požadavek dorazí do „místa určení", začne se posílat odpověď v podobě požadovaných dat (text, obrázky, videa nebo celá webová stránka). Jakmile se všechna data odpovědi dostanou do počítače, ze kterého byl požadavek odeslán, webový prohlížeč je vykreslí pomocí takzvaného „vykreslovacího jádra". Právě na jádru a na kvalitě jeho zpracování záleží to, jak se bude výsledná stránky zobrazovat. Stránka totiž není něco jako „obrázek s rozšířenými funkcemi", ale je to pouze soubor dat, a to, jak se vykreslí, už je na prohlížeči. Na některém prohlížeči se Vám může zobrazovat jenom text bez jakékoli velikosti, fontů a barev. Na dalším třeba nádherně graficky vyladěný text s obrázky a pozadím, na němž se prolínají různé tvary a vzory. Nejznámější prohlížeče jsou Google Chrome, Mozilla Firefox, Opera, Safari, Edge a na ústupu Internet Explorer.

Webový prohlížeč - Internet Explorer 11

# Technologie a služby

V této kapitole bych vám chtěl představit ty nejpodstatnější technologie a služby, které je potřeba znát pro tvorbu webu.

## Webová stránka

Internet je systém propojení počítačových sítí po celém světě. Tyto počítače mezi sebou komunikují pomocí sady protokolů TCP/IP. Tato sada určuje syntaxi neboli pravidla sdílení dat mezi počítači. Zkratka WWW neboli World Wide Web je systém založený na protokolu HTTP a slouží na prohlížení, ukládání či odkazování na soubory na internetu. HTTP či Hypertext Transfer Protocol je internetový protokol, který je úzce spřažen s webovými stránkami a systémem W3. Slouží pro přenos souborů webové stránky mezi WWW serverem a prohlížečem. Struktura webové stránky se dnes tvoří převážně pomocí jazyka HTML, CSS. Tyto jazyky jsou základní, co se grafické části týče. Pokud bychom však chtěli, aby naše stránka měla nějaké funkce, tedy ne jenom pěkně vybarvený a nastylovaný text s obrázky a odstavci, ale aby kupříkladu odeslala email po stisknutí tlačítka „odeslat komentář", či změnila obrázek v galerii po stisknutí tlačítka, použili bychom JavaScript.

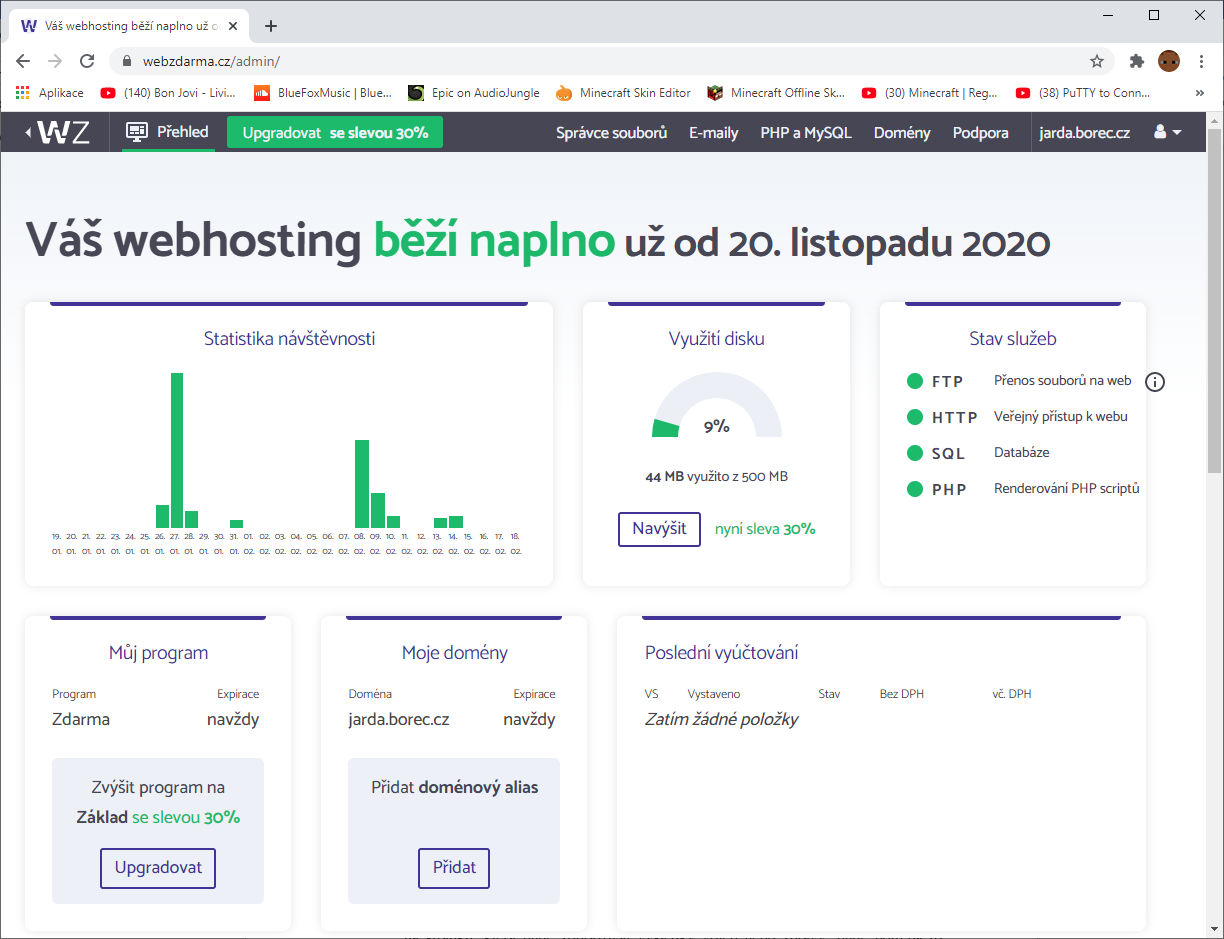
## Statická a dynamická stránka

Než začneme tvořit vlastní web, musíme se zamyslet na tím, zda bude naše stránka statická nebo dynamická. Pokud bychom chtěli stránku statickou, znamenalo by to, že by byl její obsah pevně stanoven ve zdrojovém kódu a nebylo by možno ji upravit bez zásahu do kódu. Tento typ stránek se používá převážně, když chcete prezentační web čili něco, co se například skládá z jedné hlavní stránky a k tomu třeba nějaké galerie fotografií vašich projektů, či pouze takzvanou webovou vizitku, což je jedna stránka, na které je většinou vaše fotka či logo a pár informací o vás. Pokud chceme dynamickou stránku, tak mluvíme o stránce, jejíž obsah je proměnlivý, tudíž není přesně uřčen v kódu a zobrazuje informace převážně z databází, či jiných webů. Dynamický web je například idnes.cz, který pokaždé načte poslední zprávy uložené v databázi tudíž jeho obsah není pevně stanoven v kódu. Je to kupříkladu také dmsoftware.cz, jenž zobrazuje známky, či zprávy z databáze, takže nikdo nemusí pravidelně zapisovat přímo do „zdrojáku" nové známky, ale web si je nahraje a zobrazí sám. Další dynamický web je in-pocasi.cz, který zobrazuje aktuální předpovědi počasí a opět zde nemusí nikdo sedět a dopisovat informace ručně, vše se děje automaticky.

## Webový server

Webový server je počítačový program, který běží na serverovém počítači. Tento program komunikuje pomocí protokolu http nebo HTTPS. Rozdíl mezi těmito dvěma protokoly je ten, že HTTPS je šifrovaný. Tato komunikace funguje velmi jednoduše, když prohlížeč vyšle požadavek, server na něj zareaguje, buď akcí nebo odpovědí. Nejčastější odpovědí bývají zdroje pro webové stránky, čili HTML dokumenty, CSS styly nebo obrázky. Nejčastější akcí je pak uložení souboru při ukládání do cloudu. Na každý požadavek server společně s odpovědí posílá i takzvaný stavový kód. Nejznámější je 200, když je vše v pořádku a 404, když nelze najít požadovanou stránku či soubor. Při komunikaci se používají takzvané metody, které pomáhají serveru rozhodnout, jak se má daný požadavek obsloužit. Jedna z nejpoužívanějších je metoda GET, která se používá pro získání konkrétního zdroje ze serveru, například loga, obrázků, videí, HTML dokumentů nebo stylů. Komunikace po připojení na server zadané domény probíhá tak, že server nejdříve vrátí hlavní stránku HTML na požádání prohlížeče. Ta se většinou nazývá index.html. Pokud při zpracovávání této stránky narazí prohlížeč na zdroje, například obrázky nebo videa, zažádá o ně na server právě pomocí metody GET. Po dotažení všech zdrojů ze serveru se stránka vykreslí pomocí grafického jádra. Nejznámější webový server je Apache HTTP Server.

## Hosting

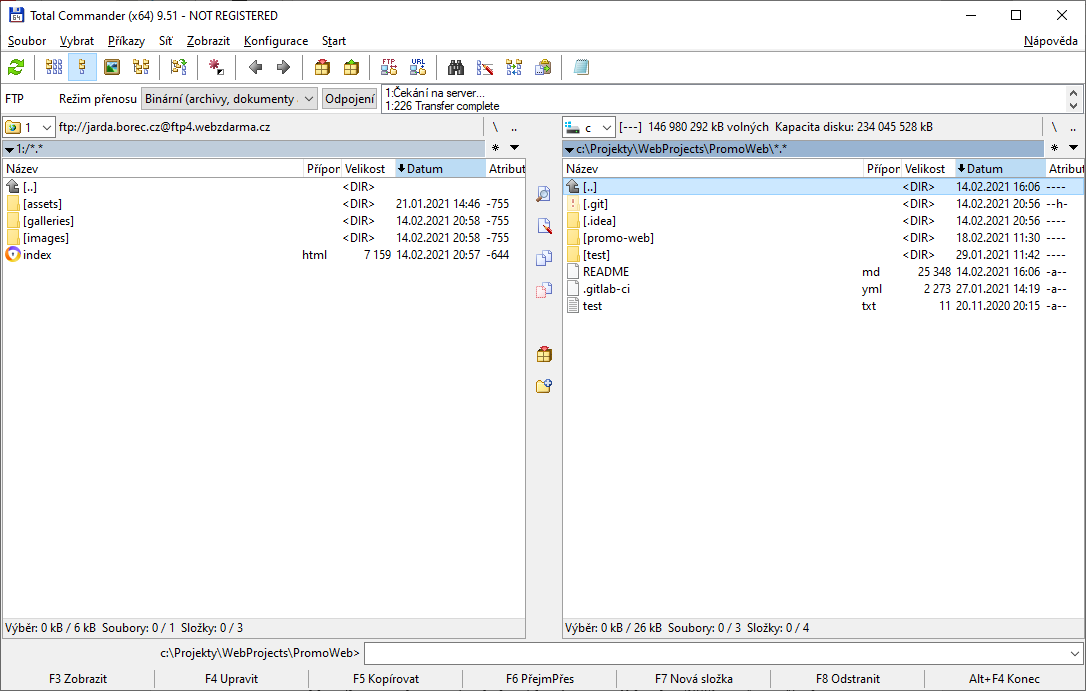
Pokud už máme jasnou představu o tom, jak bude naše stránka vypadat, zbývá nám už jen sehnat hosting a zaregistrovat si doménu. Webhosting je jenom pronajmutí kousku veřejného serveru, na němž naše stránky poběží. Důležité je také zaregistrovat si doménu, kterou propojíte s hostingem, aby se vaše stránka neukazovala jen pod IP adresou, ale také pod klasickou URL a lidé ji pak lépe našli. Podstatné je také to, abychom si řekli, zda nám postačí hosting zdarma neboli freehosting, a nebo budeme-li potřebovat něco robustnějšímu. Freehosting obnáší nižší výkon, méně služeb a často také reklamu, ale zase vám někteří poskytovatelé nabídnou zdarma vlastní doménu 3. řádu. Tato varianta se hodí pro nenáročné stránky, na které nebude chodit moc lidí, například právě prezentační web. Oproti tomu třeba na stránku, která bude zobrazovat výsledky voleb nebo zprávy, bude potřeba už nějaký podstatnější hosting, jenž zvládne několik tisíc přístupů za vteřinu. Také pak bude obsahovat větší úložiště a větší rychlost komunikace. K němu už budeme také potřeba pronajmout nějakou doménu. Tu si však nemůžeme vybrat jakoukoli, protože nemůže mít více lidí stejnou doménu, ale za to si můžeme zvolit, jak bude vypadat. Nejznámější české hostingy jsou třeba Webzdarma.cz nebo Endora.cz.

Hosting webzdarma.cz – hlavní přehled mého webu

## Domény

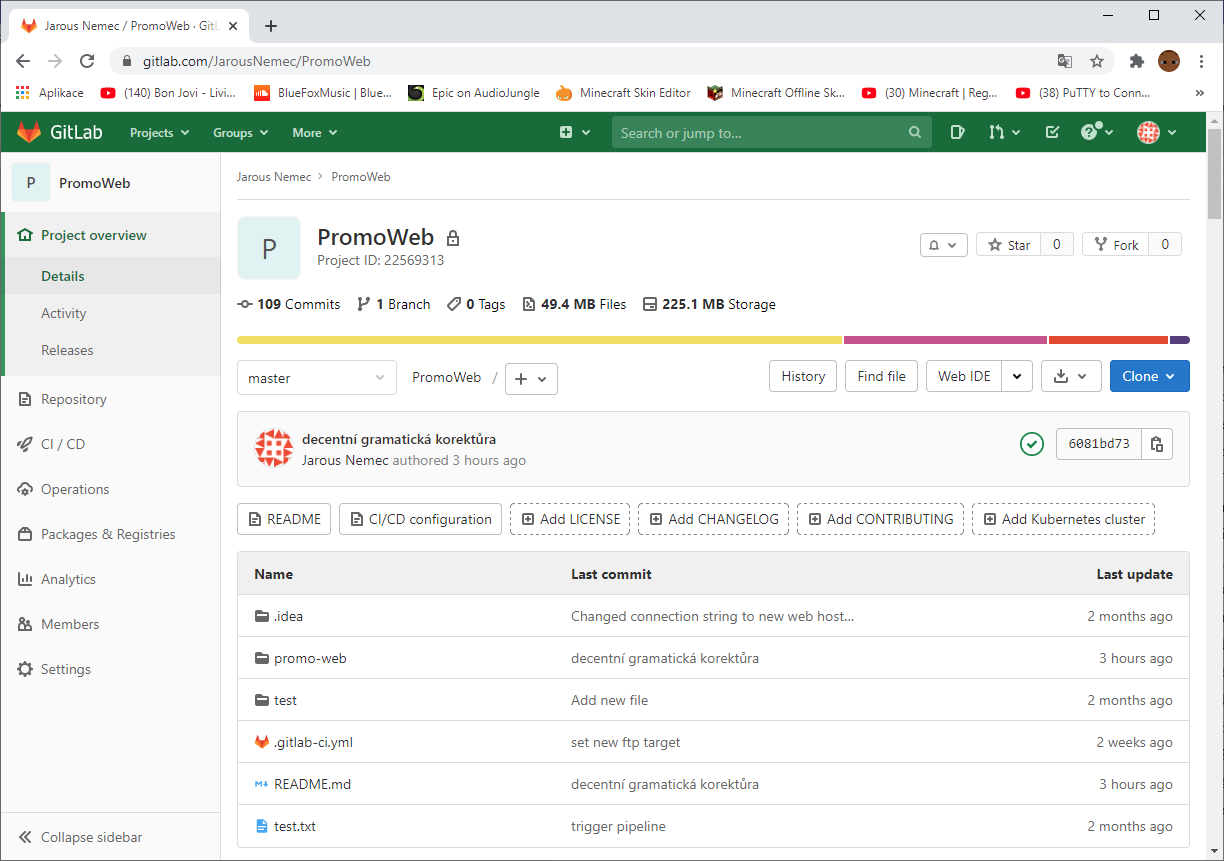
IP adresa je jednoznačný identifikátor počítače v prostředí internetu. Dříve se používaly pouze IP adresy, které byly složité na zapamatování, proto byl vyvinut DNS systém, jenž IP adresy překládá do lépe zapamatovatelné, tedy textové formy. Například pamatovat si idnes.cz je mnohem jednodušší, než si pamatovat 185.17.117.32. Webový prohlížeč se tedy po zadání domény/adresy optá na DNS server, který mu vrátí přeloženou IP adresu, na níž už potom prohlížeč rovnou volá. Domény mají 3 základní řády. Vezmeme-li si kupříkladu doménu auto.bazos.cz, tak auto je 3. řád, bazos je 2. řád a cz je 1. řád. Řády slouží pro jednodušší orientaci DNS serveru, aby rychleji věděl, jakou IP adresu má vrátit.

## FTP

FTP neboli File Transfer Protocol se využívá k přenosu dat mezi počítači, které nejsou v jedné síti. Já jsem kupříkladu použil FTP protokol, když jsem potřeboval nahrát svůj web na server. Pro připojení k FTP serveru je potřeba znát jeho IP adresu, heslo, a v případě, že se připojujete na hosting, kde používá jeden FTP server více domén, je potřeba znát i uživatelské jméno. Uživatelské jméno většinou bývá stejné jako název domény. Pro ruční přenášení se používají klienti jako je Total Commander, či Filezilla. Pokud však používáte například GitLab, který umožňuje po nahrání souboru spustit linuxový subsystém, použijete klienta lftp. Tento klient nemá grafické rozhraní, tudíž je s ním možno pracovat jen v příkazové řádce. Z tohoto důvodu se perfektně hodí pro tento typ automatizace, kdy vytvoříte jen úvodní script a ten se pak spouští pokaždé, když upravíte kód webu. V něm uvedete přihlašovací údaje k FTP serveru pro klienta lftp a přidáte příkazy, které potřebujete (tedy uvedete, co se s jakými soubory má stát).

Total Commander – připojení FTP server mého webu (okénko vlevo)

## Gitové úložiště

Git se v informatice nazývá systém nebo nástroj pro zprávu zdrojů, který využívají programátoři k ukládání, zálohování a verzování své práce. Vývoj tohoto systému začal již v roce 2005. Navrhl ho vývojář Linus Torvalds, kvůli vývoji jeho hlavního projektu, a tím bylo jádro Linuxu. Od roku 2002 totiž používali pro verzování produkt BitKeeper, který začal být v roce 2005 zpoplatněný a tudíž se už nehodil pro tvorbu free softwaru jako je Linux. V dnešní době existuje mnoho Gitových projektů jako je například GitHub, GitLab, Bitbucket nebo SourceTree. Všechny tyto produkty pracují na stejném základu, ale mohou se lišit kvalitou uživatelského rozhraní nebo také například možnostmi sdílení názorů a poznatků mezi vývojáři. Právě již zmíněný GitLab jsem použil pro tvorbu mého prezentačního webu. Gitlab totiž umožňuje pomocí jednoduchého scriptu, nahrávat předurčená data na webový server pomocí FTP protokolu. Spuštění tohoto scriptu se dá nastavit v jeho konfiguraci, ve které se určuje při změně jakého souboru či složky se má tento script spustit. Dále se tam dá nastavit, jaké soubory se mají nahrávat, protože většinou nepotřebujeme nahrávat všechny soubory projektu, ale stačí nám jen některé (klíčové pro stránku).

Gitlab (Gitový klient) – hlavní přehled projektu Prezentační web

# Jak jsem postupoval při tvorbě

Práci na mém webu jsem začal tím, že jsem si obecně navrhl to, co bych si představoval, aby můj web obsahoval. Potom jsem začal na internetu shánět nějakou volně dostupnou šablonu, protože psát celý web od základů je sice skvělá a hodně poučná věc, ale v podstatě je to neefektivní. Poněvadž přetváření celé šablony podle svého, také zabere mnoho času a pokud bych měl ještě řešit problémy typu, že mi někde nefunguje tlačítko nebo se někde špatně načítá obrázek, tak by v podstatě nic nového nevzniklo a byla by z toho jen nevzhledná kopa kódu, do které když byste udělali změnu, rozsypala by se. Právě proto je tedy dobré používat šablony, abyste nemuseli řešit špatně fungující web, ale mohli se věnovat obsahu. Dobré rozhodnutí to bylo také z toho důvodu, že se dělat web teprve učím a do složitých principů, jež se dnes používají, zatím moc nevidím. I tak jsem při formování šablony pár věcí rozbil a musel jsem si „naštudovat", jak je opravit. Po vybrání odpovídající šablony z webu HTML5 UP, jsem jako první věc začal skládat komponenty na úvodní stránku. V podstatě jsem se držel původního předváděcího návrhu, ale někde bylo potřeba kupříkladu přidat odstavce. Když jsem celý web nadesignoval, bylo nutno jej nastylovat (graficky vyladit), ale to však šlo bez větších problémů díky dobře vybrané šabloně. Poté jsem začal jednotlivé komponenty plnit předpřipraveným obsahem (obrázky, text, kontakty, logo).

Jakmile jsem dokončil úvodní stránku, začal jsem tvořit galerie svých dosavadních projektů. Na tyto galerie jsem vložil odkaz do popisku úvodních obrázků projektů. Opět jsem na to využil šablonu, ale všechny komponenty jsem z ní vyházel a nechal jen horní menu, do něhož jsem umístil tlačítko odkazující zpět na hlavní stránku. Využil jsem ji také proto, že obsahuje nastavení zobrazení stránky, takže kdybych ji vynechal, galerie by vůči hlavní stránce vypadaly jinak. Potom jsem na internetu našel kód kolotoče na obrázky, který jsem upravil a použil jako prohlížeč obrázků. A tak vypadala celá galerie. Takovou galerii jsem udělal ke každému projektu.

Tím byl web v podstatě hotový a já jsem mohl začít hledat nějaký webový hosting, samozřejmě zadarmo, protože prezentační web nemá nijak vysoké nároky na výkon, mohl jsem tedy zvolit tuto variantu. První hosting, který jsem vyzkoušel, byla Endora.cz, kde jsem zaregistroval doménu jarda.cekuj.net. Tento hosting byl asi nejlepší, který jsem do té doby viděl. Dokonce měl jenom malou nerušivou reklamu, takže se vše zdálo v pořádku až do té chvíle, kdy jsem se rozhodl propojit Gitové úložiště Gitlab s tímto hostingem. Zde jsem totiž ztroskotal na neznalosti poskytovaných služeb hostingu. Mnou vybraný hosting Endora.cz totiž nepodporuje FTP připojení ze zahraničí, a protože GitLab script běží na serveru ve Francii a naše Endora je v Česku, nemohlo to tedy fungovat. Tuto chybu jsem sice hledal dva večery, ale nakonec jsem na to přišel a změnil hosting na webzdarma.cz, kde jsem si zaregistroval doménu jarda.borec.cz, kde už vše fungovalo napoprvé a poté co jsem „pushnul" do Gitlabu novou verzi stránky, tak se sama nahrála na web server. Oproti Endoře zde však byla větší a rušivá reklama, která ani nebyla na samotný hosting jako předtím, ale byla jako ostatní reklamy cílená na to, na co se nejvíce díváte. Navíc na mobilu zde byla ještě vyskakovací reklama při připojení na stránku. To mě, ale v žádném případě nezastavilo, naopak jsem to vzal jako další výzvu, se kterou se při práci na tomto projektu budu muset vypořádat. Začal jsem tedy pomocí nástrojů pro vývojáře, které se zobrazí v internetovém prohlížeči při stisknutí klávesy F12, tedy konkrétně pomocí průzkumníka elementů, vyhledávat dané komponenty, které obsahovaly reklamu. Jakmile jsem je všechny vyhledal, pustil jsem se do hledání toho, jakým způsobem je odstraním. Vzhledem k tomu, že jsem je nemohl ovlivnit žádným nastavením nebo stylem, poněvadž je server přidal až po načtení mé stránky, zbyla jediná možnost vytvořit kus scriptu, který po načtení celé stránky umaže dané komponenty s reklamou. Pro toto určení se ukázal jako nejlepší volba starý známý Javascript. Následně jsem zjistil, jak najít a následně smazat elementy pomocí Javascriptu. Opět se to neobešlo bez komplikací. Zjistil jsem totiž, že vedle mého scriptu běží ještě další script, který kontroluje to, aby se pořád zobrazovaly reklamy. Tento script se nedal dost dobře nijak vypnout, protože se pokaždé nastartoval znovu. Tak jsem nakonec svůj kód poupravil tak, aby běžel pořád dokola a pokaždé, co by se protivník pokusil vložit novou reklamu, okamžitě ji smazal. A tím vznikl můj první „antivirus", protože funguje na stejném principu (zná nějaké příznaky a když objeví něco s podobnými znaky, smaže to a nebo nahlásí). Nakonec jsem po sobě už jen smazal soubory, do kterých jsem psal testovací kódy, protože nebyly pro chod stránek potřeba. Tím byl web hotov a s ním i má závěrečná práce.



Výsledný web

# Závěr

Cílem této práce bylo vytvořit si vlastní webové stránky, na nichž bych prezentoval své projekty. Tento cíl se nakonec povedlo splnit na výbornou. Hlavní kritéria byla, aby web obsahoval nějaký úvodní text o tom, co dělám a co mě baví. Dále, aby zde byly vypsány všechny mé projekty, ke kterým postupně dodělávám galerie. Bylo též potřeba zařídit to, aby bylo možno celý web jednoduše rozšířit z důvodu, abych mohl jednoduše přidávat další projekty, tedy aby bylo všeobecně jednoduché udělat změnu. Pro splnění těchto požadavků jsem se musel seznámit se základy jazyků HTML, CSS a JavaScript. Na zajištění jednoduchosti, rozšiřitelnosti a čistoty kódu jsem použil zdarma dostupnou šablonu z webu HTML5 UP, kterou jsem následně upravil k obrazu svému. Ve finále z toho vznikl vcelku slušný web, u jehož tvorby jsem se naučil základy výše zmiňovaných jazyků, našel spoustu vychytávek, naučil se práci s IDE a při řešení různých nepřesností a problémů jsem zjistil důležité věci a principy ohledně fungování webových stránek a internetu obecně. Největší problém asi bylo propojit GitLab s hostingem pomocí protokolu FTP. Naopak nejzajímavější problém byl onen odstraňovač reklam, u kterého jsem se více ponořil do základů JavaScriptu a struktury webů. Do budoucna by bylo potřeba vylepšit zobrazení galerií na mobilních telefonech a možná přidat nějaký popis k jednotlivým projektům.

Na závěr tedy můžeme říci, že vytvořit si v dnešní době vlastní prezentační web není za využití šablon nic nezvladatelného ani pro běžného smrtelníka. A díky právě šablonám vám postačí pouze základní znalosti HTML a kaskádových stylů (CSS) takže jde celá věc s trochou logického myšlení zvládnout relativně rychle. Nejtěžší asi bude nahrát stránky na webový server, ale většina hostingů má dnes podrobné návody, takže ani tato operace nebude pro běžného člověka nic těžkého.

# Zdroje

Martin Domes, Tvorba WWW stránek pro úplné začátečníky*.* Místo vydání: CPress, 12.03.2012. ISBN 978-80-251-2160-3.

Martin Domes, Internet pro úplné začátečníky*.* Místo vydání: CPress, 08.09.2011. ISBN 978-80-251-3650-8.

Jiří Hlavenka, Vytváříme WWW stránky*.* Místo vydání: COMPUTER PRESS, 03.04.2002. ISBN 80-7226-494-X.

cs.wikipedia.org: Git [online]. Poslední změna 4. 6. 2020. [Cit. 4. 6. 2020]. Dostupné z: https://cs.wikipedia.org/wiki/Git