

Vážení studenti.

Tento průvodce je určen pro všechny formy distančního a kombinovaného studia předmětu Tvorba webových aplikací.

Obsahuje instrukce k obsahu studia spolu s příslušnými odkazy na výukové zdroje i rozšiřující informace pro zájemce. *S jeho pomocí získáte základní přehled o součástech a technologiích, které se používají pro tvorbu webových aplikací.*

Problematika webových aplikací a jejich vývoje může být jak stručná, tak velmi obsáhlá. Záleží na Vašich vstupních znalostech, ale také potřebách a očekáváních, a v neposlední řadě ochotě naučit se toho o webových aplikacích co nejvíce.

Tento základní průvodce z Vás neudělá na první dobrou webové vývojáře, a existuje navíc celá řada dalších souvisejících oblastí o kterých se zde nebudeme přímo zmiňovat. Cílem totiž je, abyste poznali princip, poznali základní technologie a v ideálním případě si sami našli v široké paletě činností vývoje webových aplikací tu, ve které budete vynikat a která Vás bude především bavit. Tak hodně štěstí.

***Poznámka:** Pokud Vám ještě nic neříkají pojmy či zkratky jako Server, Klient, HTTP nebo URL, prosím přečtěte si nejprve 1. kapitulu staré studijní opory nebo se prostě mrkněte aspoň na tento odkaz z webu Jak Psát Web:*
<https://www.jakpsatweb.cz/jak-facha-www.html>

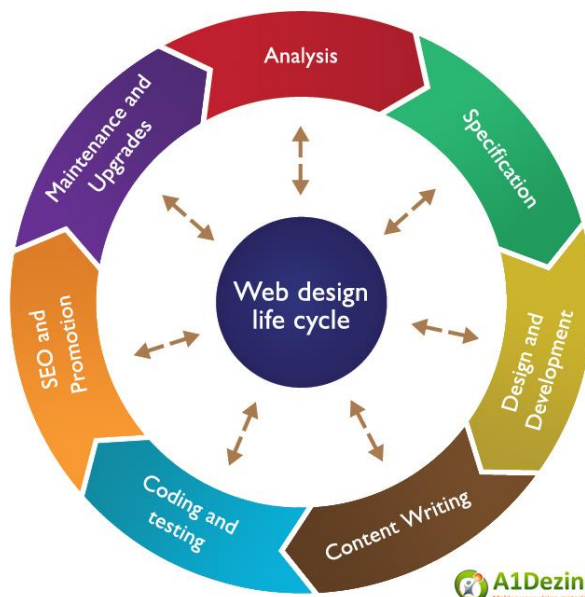
Životní cyklus (vývoje) webové aplikace

Webová aplikace prochází stejně jako jiný SW fázemi, které by měly zajišťovat vysokou kvalitu výsledného produktu díky dodržování určitých rámců a standardů. V praxi jsou tyto úzce propojené fáze označovány jako:

- Průzkum a analýza zadání
- Plánování
- Návrh
- Vývoj
- Testování a uvedení do provozu
- Údržba, Monitoring a Upgrade

Existuje spousta otázek, na které lze díky předchozímu seznamu (zejména analýzou zadání) najít docela jednoduché odpovědi. Vezmeme to ale popořadě.

Dejme tomu, že máme zadaný projekt webové aplikace, který musíte kvůli zápočtu z tohoto předmětu splnit 😊. Otázka zní: Budu muset i něco programovat či skutečně sám/sama vyvíjet?



Pokud se to chcete naučit a chcete si to vyzkoušet – tak samozřejmě ANO. Pokud ale chcete být efektivní a chcete maximálně využít to, co nám dnes svět kolem webových aplikací nabízí, můžete na to jít i jinak. Začněme proto vzorovým (jen pro ukázkou) zadáním konkrétního projektu, abyste si udělali určitou představu.

Projekt 1

Zadal jsem Vám, že byste k zápočtu měli např. vytvořit základní web obsahující tyto části (stránky):

- Úvodní stránka obsahující základní informace o Vaší osobě nebo o objektu, který web prezentuje.
- Stránka obsahující fotogalerii (seznam zmenšených obrázků fotografií, u kterých se po kliknutí zobrazí fotografie v původní velikosti).
- Stránka obsahující kontaktní informace, včetně formuláře, kterým Vás budou moci návštěvníci webu kontaktovat na Váš email.
- Stránka obsahující např. interaktivní anketu, v níž můžou návštěvníci hlasovat ve Vámi zveřejňovaných průzkumech.

Projekt je zadán, nyní se podívejme blíže na možnosti jeho řešení.

Způsoby tvorby (nebo zřízení) webové stránky nebo aplikace

Tyto způsoby bychom si mohli uvést následovně: uvedu Vám je v pořadí tak, jak je považuji od nejjednodušší po nejnáročnější (pozor ale, neznamená to automaticky vzestupnou či sestupnou úroveň kvality výsledku; tzn. že každý způsob nám v podstatě umožní vytvořit velmi kvalitní aplikaci, jen cesta je rozdílná).



1. Online systémy pro tvorbu a správu webu a webových aplikací

Jsou to systémy správy obsahu a webových komunikačních objektů, které umožňují vytvářet a spravovat zájemcům weby a webové aplikace tak snadno, že lze téměř říci: *ten kdo umí ovládat email, je schopen si zde vytvořit a zprovoznit webovou aplikaci.*

Není potřeba mít téměř žádné znalosti základních či rozšiřujících technologií tvorby webových aplikací, vše je zde připraveno i pro úplné začátečníky. Weby se ve většině těchto systémů tvoří pomocí jednoduchých průvodců a interaktivních rozhraní. Jedná se o službu, která za Vás vypracuje nezbytný základ cílového projektu, a to okamžitě na základě Vašich povelů a instrukcí. Pod vlastním uživatelským jménem a heslem získáte přístup do správy obsahu a nastavení své aplikace, a uvidíte, že tvořit vlastní web je zde hračka, kterou zvládnou třeba i žáci ZŠ.

U nás patří mezi velmi oblíbené:

- Webnode - <https://www.webnode.cz/>
- Wix - <https://cs.wix.com>



Jediné, co k tomu budete potřebovat je **webový prohlížeč** a samozřejmě připojení k internetu. No a co vlastně všechno tyto systémy dokážou? To už si můžete sami vyzkoušet v rámci prvního hodnoceného úkolu. Určitě to hravě zvládnete; podobně hravě jako zvládnete pracovat s emailem.

A jaké jsou limity tohoto způsobu?

Každý systém má určité hranice. A požadavky na webové aplikace se nám díky možnostem dnešních technologií neustále rozrůstají. Takže ani online systémy nemohou v daném čase vyhovět úplně všem nárokům. Mnoho požadavků vycházející na funkci webových aplikací jsou vysoce individualizované z hlediska potřeb zákazníků, firem, organizací, ale i jednotlivců. Přichází tak na řadu další možné způsoby tvorby webových aplikací.

2. Využití přenositelných CMS

Stejně jako existují online systémy správy obsahu, kterýchž obsluhu potřebujete jen webový prohlížeč a jejichž provoz je zabezpečen na serverech poskytovatelů, **existují také publikační systémy (CMS), které si lze stáhnout** a používat téměř kdekoli se nám zamane. No úplně kdekoli asi ne, ale určitě na všech serverech, kde je odpovídající prostředí nutné pro jejich provoz.



A provoz takového systému znamená min. následující:

- Webový server – instalovaný lokálně pro testování nebo vzdáleně pro provoz v síti.
- Databázový systém ...
- Interpret serverového skriptovacího jazyka, nejčastěji PHP;

Úvodní informace k této problematice si můžete přečíst např. na oblíbené Wikipedii.¹ Dozvíte se o základních funkcích, výhodách či přehledu dnes nepoužívanějších systémů.

Oproti předchozímu způsobu rostou nároky na technické znalosti a způsobilosti uživatele. Nejde ani tak o obsluhu těchto systémů z pohledu administrátora nebo uživatele, který plní obsah v administračním rozhraní, **ale především o znalosti související s provozem webového a databázového serveru.** Tzv. lokální testovací prostředí a vše s ním spojené si představíme později. Pokud byste měli zájem začít ihned – tak pro začátek doporučuji použít **CMS Wordpress**. Návod na instalaci a zřízení pro začátečníky je dostupný třeba zde².



¹ https://cs.wikipedia.org/wiki/Syst%C3%A9m_pro_spr%C3%A1vu_obsahu

² <https://digitalninomadstvi.cz/jak-vytvorit-web-ve-wordpressu-navod/>

3. Vývoj webových aplikací v IDE³ kombinující práci s kódem s prací v grafickém studiu

Znáte termín nebo zkratku **WYSIWYG**? Neboli „co vidíš, to dostaneš“? Určitě ano. Typickým příkladem je textový procesor, např. MS WORD. Existují ale také nástroje pro tvorbu webu a webových aplikací. Jejich hlavní devízou je, že díky grafické nadstavbě **umožňují vytvářet web nejenom přímým zápisem jeho zdrojového kódu, ale také pomocí grafických nástrojů s automatickým generováním zdrojového kódu aplikace na pozadí.**

Mezi typické představitele patří **Adobe Dreamweaver.**



My se dále těmto nástrojům věnovat nebudeme. Ne že by to nestálo za to. Nicméně jejich použití je samo o sobě podmíněno nejenom znalostí jejich obsluhy, ale také znalostí a zejména zkušeností se základními technologiemi tvorby webu. Proto jsou tyto nástroje vhodné, až pro mírně pokročilé vývojáře.

*Zároveň se zde ani nezmiňuji o programech typu **MS Publisher**, které fungují na podobném principu a jejichž výstupem může být dokument webové stránky. Nicméně účel těchto programů je v základu jiný.*

4. Tvorba webu a webových aplikací zápisem zdrojového kódu

A jsme na konci výčtu zde prezentovaných způsobů. **A zároveň jsme vlastně na úplném začátku.** *Web nebo webová aplikace ehm ... pořád používám oboje, jedno nebo druhé. Je načase si říci, jaký je v tom rozdíl. Pomůže Vám pro začátek tento obrázek?*



Možná ano, možná ne. Pokud byste chtěli číst o mnoha rozdílech – stačí si jen trochu ,zagooglit'. V této chvíli bude pro nás dostačující vědět, že:

Hlavní znaky webové aplikace

- Dynamická a Vysoce interaktivní

³ INTEGRATED DEVELOPEMENT ENVIRONMENT – Vývojové prostředí

- Obsahující většinou široké spektrum nástrojů pro výměnu informací mezi uživateli, systémy, atd..
- Umožňující každému uživateli vidět jemu přizpůsobené informace v každém čase.

Hlavní znaky webu (webových stránek)

- Statická, ale v zásadě i interaktivní
- Každý uživatel vidí v daném čase stejný obsah

Webové aplikace jsou mladšího data objevení 😊, webové stránky známe totiž už počátku služby WWW; www stránky známe všichni – firemní prezentace, osobní vizitky, ale také např. jednoduché školní weby. Webové aplikace známe také. Mohly vzniknout díky rychlému tempu vývoje technologií. Jsou to např. Sociální sítě, Firemní CRM, Webové informační systémy, Internetové obchody atd.

Pokud už v tom máte jasno, můžeme se vrátit na onen zmíněný začátek.



Zdrojový kód

Předchozí obrázek na Vás zřejmě neudělal valný dojem, ale nebude to až tak hrozné. Ať už byste využili k tvorbě webu jakoukoli z možností, vždy by základem všeho byl onen zdrojový kód. **Web nebo webová aplikace je v zásadě sada fyzických souborů, obsahujících příslušný zdrojový kód.** A webové aplikace tedy tvoříme tak, že tento zdrojový kód do těchto souborů zapisujeme. Ne nadarmo se jeden ze známých českých portálů o tvorbě webových stránek jmenuje **Jak Psát Web**.⁴ *Pravidla, podle kterých se zapisují zdrojové kódy označujeme jako **standardy** a je jich opravdu velké množství.*

Nikoho ale nechci hned děsit. Něco je potřeba napsat, něco už bylo napsáno, a tak to jen využijeme, něco si jen pro svoje potřeby upravíme z hotových šablon atd. Ale především je potřeba rozumět principu fungování a jak se to vlastně všechno dá složit dohromady.

Pokud čtete tohoto průvodce, patrně jste se mnou už absolvovali **úvodní přednášku**. V ní jsem se snažil principy fungování webu a úlohu jednotlivých technologií co nejvíce přiblížit. Poklady k tomu obsahuje PowerPoint prezentace nazvaná **Úvodní seminář**. Pojďme si tedy připomenout, co bychom už měli znát a začněme hned obrázkem.

⁴ <https://www.jakpsatweb.cz/>

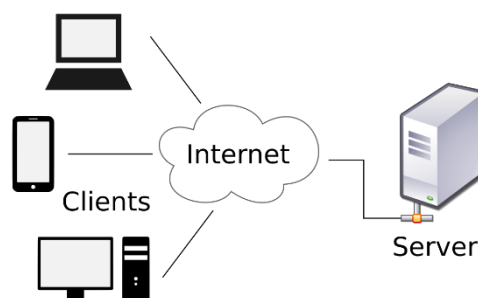


Klient - Server

Základem všeho je architektura služby WWW v komunikaci mezi klienty a servery. Klient je reprezentován webovým prohlížečem, server představuje speciální software, který požadavky klientů vyřizuje.

Server může běžet na vzdáleném počítači někde v síti, ale může být také instalován na Vaší pracovní stanici jako tzv. lokální webový server.

No a pokud chceme vytvářet weby nebo webové aplikace, mít možnost si je testovat a pohodlně spouštět u sebe doma, tak se v podstatě bez lokálního webového serveru neobejdeme.



Statické weby vs Dynamické weby (aplikace)

V této chvíli bychom měli být schopni vyslovit základní rozdíly:

Statické weby

- Skládají se většinou pouze z fyzických souborů HTML (*.html), CSS (*.css) a případně JavaScriptu (*.js).
- Neobsahují žádnou procesní logiku pro server a jejich načítání je velmi rychlé.
- Většinou se jedná o jednoduché informační weby, u nichž není požadavek časté změny obsahu.
- Na jejich tvorbu si vystačíme se znalostí jazyků (standardů) HTML, CSS a případně Javascriptu.
- Díky obrovskému množství hotových šablon, které jsou na webu buď zdarma nebo za poplatek k dispozici, je tvorba statického webu dostupnější než kdy dříve.
- K jejich tvorbě nám postačí vývojové prostředí (IDE) a webový prohlížeč.

- K jejich provozu v síti je nutný webový server.

Dynamické weby

- Kromě fyzických souborů statického webu ([bod 1 předchozího seznamu](#)) obsahují také fyzické soubory zvolených serverových programovacích jazyků, např. PHP (*.php).
- Obsahují procesní logiku pro server, která tvoří dynamickou část.
- Jejich tvorba je podmíněna znalostí zvoleného serverového programovacího jazyka.
- Jejich načítání je také velmi rychlé, ale závisí na množství a složitosti dynamických procesů.
- Umožňují zpracovávat data odeslaná klientem na server (např. data odeslaná registračním formulářem atp.).
- Často pracují s připojením do databázového systému.
- **K jejich tvorbě a testování je nutné běhové prostředí, skládající se z lokálního webového serveru, interpreta zvoleného jazyka a případně databázového systému.**
- Dynamické weby nebo aplikace také většinou neobsahují kompletní předem dané fyzické soubory HTML, ale generují je dynamicky do připravených HTML šablon, a vkládají do těchto šablon data odpovídající požadavku klienta – např. výpis produktů e-shopu v dané kategorii, nebo aktuální články z rubriky zpravodajského serveru atd.

Internetový (webový) prohlížeč



Webový prohlížeč je program, který se o to, co se děje na serveru příliš nezajímá. Prohlížeč čeká na odpověď serveru a ta musí obsahovat vždy jen ono HTML, CSS a případně JS. Něco málo k prohlížečům opět na Wikipedii⁵. Schválně – **jaký je Váš oblíbený?** Dnes vede určitě Chrome. Kdo vyvíjí webové aplikace by měl ale mít povědomí i o těch ostatních, protože klienti mohou použít jakýkoliv. Nicméně nebojte, stačí když pro začátek budete úspěšně používat aspoň ten Váš.

Projekt 2

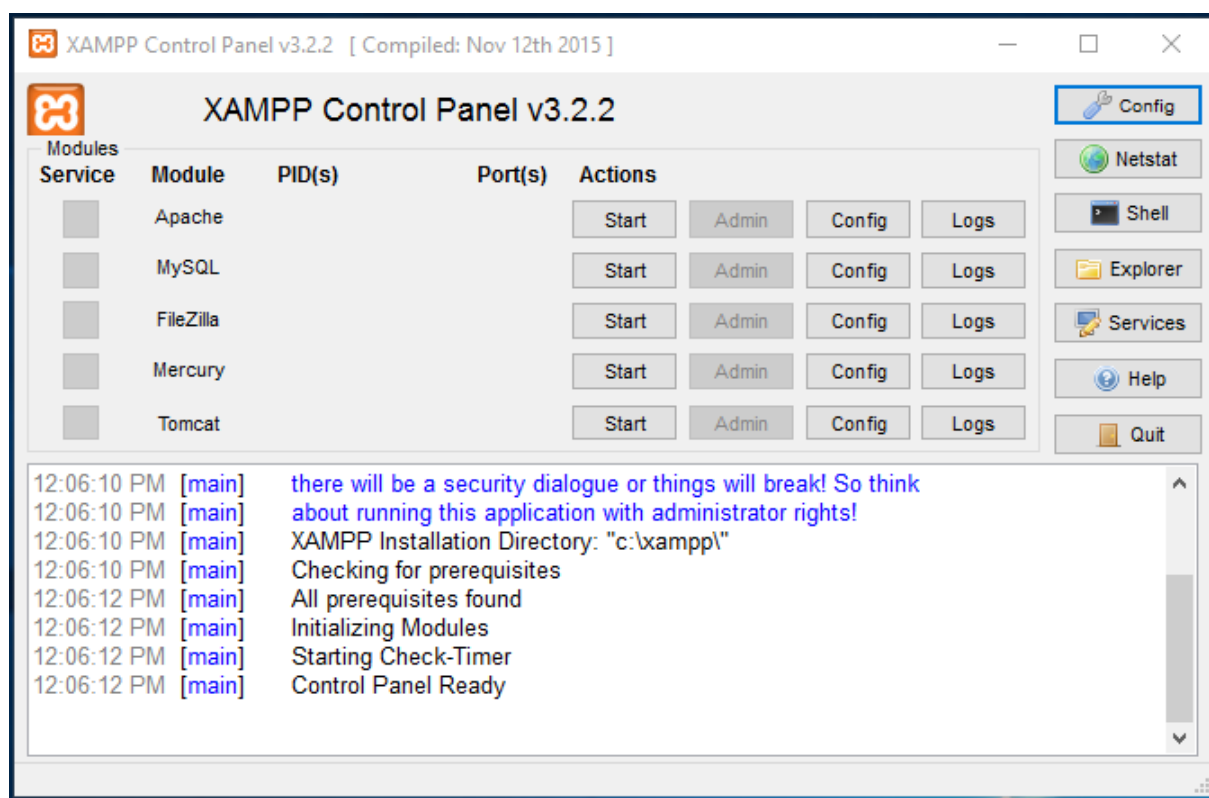
Náš druhý projekt vychází z následujícího obsahového zadání.

- Úvodní stránka obsahující základní informace o Vaší osobě nebo o objektu, který web prezentuje.
- Stránka obsahující fotogalerii.
- Stránka obsahující kontaktní informace, včetně formuláře, kterým Vás budou moci uživatelé kontaktovat na Váš email.

⁵ https://cs.wikipedia.org/wiki/Webov%C3%BD_prohl%C3%AD%C5%BEE%C4%8D

Práce nebo chcete-li studium projektu spočívá především v poznání víceméně hotové konstrukce webové aplikace a zejména ve zprovoznění vlastního prostředí pro běh aplikace. **Vaším úkolem je tak spíše studium a testování.** Hned na začátek si uděláme přípravu toho, co bude potřeba:

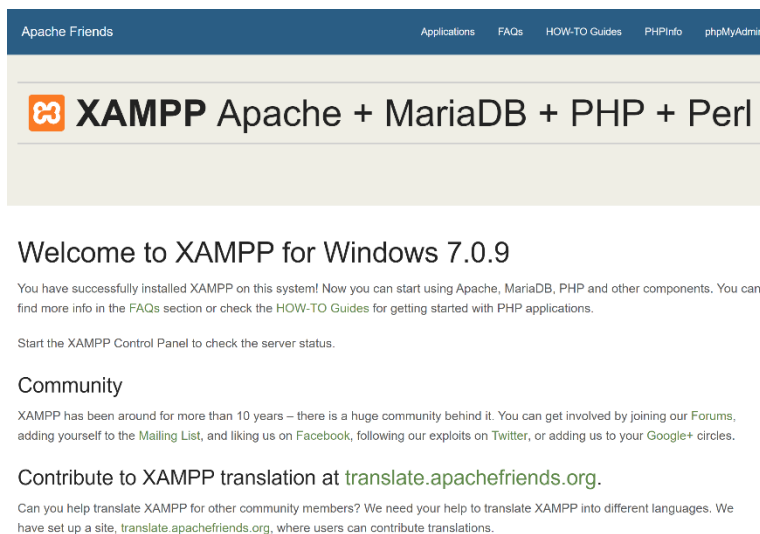
1. Asi jste už poznali, že budeme vytvářet (resp. zkusit a studovat) dynamický web. A s tím souvisí, abychom měli k dispozici **lokální běhové prostředí** obsahující **webový server** a **interpreta zvoleného serverového jazyka** (PHP). Obě tyto součásti se dají velmi pohodlně instalovat pomocí nástroje XAMPP.
2. XAMPP si stáhnete pro svůj OS na adrese: <https://www.apachefriends.org/index.html> Jedná se o přímý instalátor, díky němuž získáte **webový server Apache**, **Interpreta jazyka PHP** a některé další komponenty, především pak **databázový systém**.
3. Při instalaci Xamppu projdete vícero kroky, jejichž obsah můžete potvrdit. Instalace na Windows možná bude hlásit varování spojené s nastavením nástroje řízení uživatelských účtů, ale toho se nebojte a nainstaluje Xampp do složky **c:/xampp**.
4. Jakmile budete hotovi, tak ve složce c:/xampp najdete program **xampp-control.exe** a spusťte jej:



5. Pokud u řádku **Apache** kliknete na tlačítko **Start** a text Apache bude svítit **zeleně**, tak výborně, na Vaší stanici běží lokální webový server. Mohou nastat i problémy 😞. Např. že Apache nepůjde spustit, protože výchozí port naslouchání webového serveru (80) je blokován jiným programem. Např. Skypem. Skype vypněte a zkuste Apache nainstalovat

znovu. Pokud to stále nepůjde (**čtete log v control-panel**), musíte zřejmě vyhledat konkrétní program, který Vám port blokuje. Pomůže Vám také příloha 1 původní studijní opory.

6. Otestujeme si funkčnost lokálního serveru v prohlížeči. Do adresního řádku zadejte: <http://localhost> – výsledek by měl být zhruba jako na obrázku níže:



7. **Výborně!** Máme lokální webový server. Abychom se mohli pustit do vývoje či pouze studia webové aplikace, budeme potřebovat ještě vývojové prostředí:
 - a. Stáhněte a nainstalujte si jednoduchý editor zdrojových kódů **PSPad**. Je v češtině a práce s ním je hračka.⁶ Je to základní editor vhodný pro jednoduché projekty a neměl by chybět ve vybavení každého studenta IT.
 - b. Stáhněte a nainstalujte si IDE **Apache Netbeans**. Jedná se o free IDE pro PHP a HTML a je to perfektní nástroj i pro tvorbu profi PHP aplikací⁷. Vyberte spustitelnou verzi .exe a můžete instalovat.



Pokud máte vše připraveno, běží webový server, máte stažen Apache Netbeans a hlavně máte čas a ochotu se do toho pustit, přišla na to ta správná chvíle. Celý projekt jsem pro Vás už připravil a snažil se jej vhodně okomentovat. Na Vás je si stáhnout jeho zdrojové kódy, umístit na správné místo do adresáře serveru Apache pro webové dokumenty, otevřít ideálně v Netbeans a webovém prohlížeči a začít studovat jeho části a případně zkusit doplňovat jednoduché úlohy. Tak jdeme na to:

1. Stáhněte si složku projektu (projekt.zip).
2. Rozbalte tuto složku.

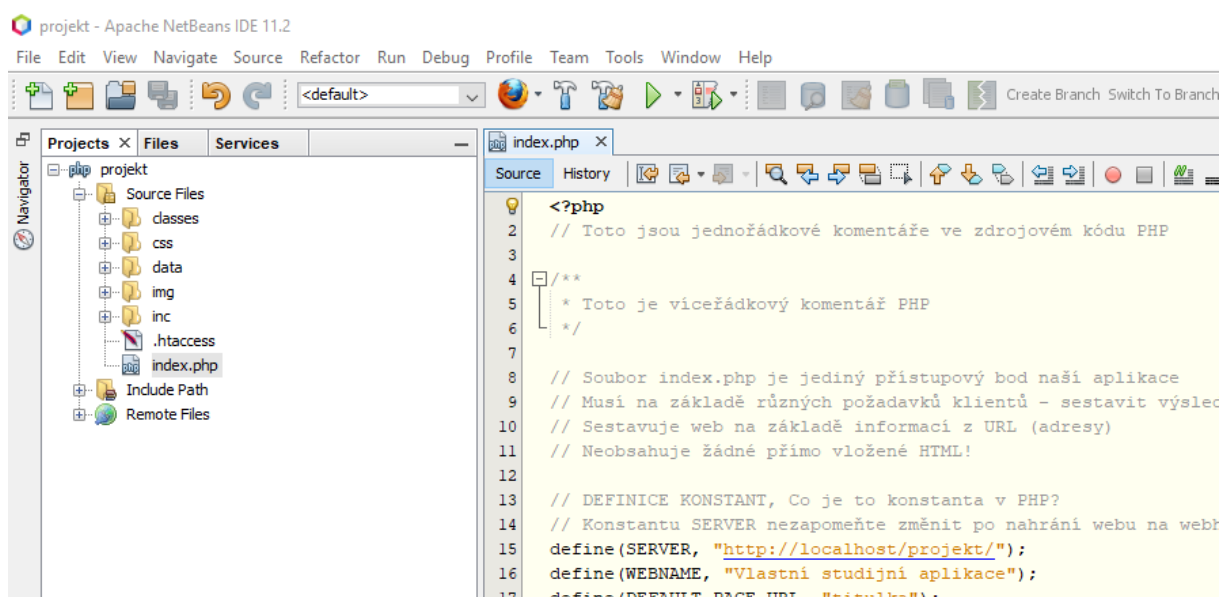
⁶ <http://www.pspad.com/cz/>

⁷ <https://netbeans.apache.org/download/nb121/nb121.html>

3. Celou rozbalenou složku zkopírujte do adresáře: `c:/xampp/htdocs`. Cesta ke složce tak vypadá následovně: `c:/xampp/htdocs/projekt`
 - a. Vzhledem k tomu, že se jedná o dynamickou aplikaci, je nezbytně nutné ji spouštět jen z tohoto umístění, protože webový server má adresář **htdocs** defaultně nastaven pro webové dokumenty.
4. Nyní můžete otevřít internetový prohlížeč a zadat adresu: <http://localhost/projekt>

Otevřeme si projekt v **Apache Netbeans**:

- V záhlaví programu vyberete: File -> Open Project
- Vyhledáte složku projektu v `c:/xampp/htdocs/projekt` a Łknete na Open Project



Projekt se skládá z několika složek a ve výchozí úrovni dvou souborů:

- **.htaccess** – jedná se o soubor, kterým může každá aplikace částečně ovlivnit nastavení (konfiguraci) webového serveru pro své prostředí. Nebudeme se jím nyní zabývat. Obecně je to soubor nepovinný, ale můžete se s ním setkat.
- **index.php** nebo **index.html** – je to výchozí soubor aplikace nebo webu. Když do prohlížeče zadám <http://localhost/projekt> nebo <http://localhost/projekt/index.php> tak se vykoná totéž. Načte se soubor index. Pokud projekt takový výchozí soubor nemá, nelze aplikaci v prohlížeči na adrese <http://localhost/projekt> načíst.

Kapitoly studia

Dále zde uvedu kapitoly studia, se kterými byste se měli teoreticky seznámit a z nichž se bude následně skládat Váš kolokviální test.

1. Základní pojmy vztahující se k Webovým aplikacím

Jedná se o pojmy, které jsou uvedeny v úvodní prezentaci tohoto předmětu. Více se k této problematice můžete dovědět také v první kapitole studijní opory.

2. Jazyk HTML a Kaskádové styly CSS

Značkovací jazyky HTML a CSS jsou základní stavební kameny tvorby webu. Pokud o nich nemáte žádné povědomí, tak jako dostačující studijní zdroj Vám mohu doporučit 2 a 3 kapitolu studijní opory.

3. Responsivita aneb Mobile First

Dávno je pryč doba, kdy jsme téměř všichni prohlíželi web na 17palcových (většinou LCD) monitorech. Dnes převládají mobilní zařízení a web na tuto skutečnost musel promptně zareagovat. Responsivita je proto velmi důležitou součástí tvorby webových aplikací. Obecně se o této problematice dovíte více např. na:

- <https://www.rascasone.com/cs/blog/co-je-responzivni-web>

O tom, že dnes existují výborné nástroje, které nám s tímto problémem mohou výrazně pomoci, svědčí např. FrontEnd framework Bootstrap⁸.

Více o něm např. zde:

- <https://www.itnetwork.cz/html-css/bootstrap/kurz/uvod-do-css-frameworku-bootstrap>

4. JavaScript, JS knihovna jQuery

JavaScript je skriptovací jazyk, pomocí něhož jsme schopni oživit webové stránky a aplikace přímo ve webovém prohlížeči. Proto je také označován jako klientský skriptovací jazyk. Stejně jako v případě HTML nebo CSS, nám pro jeho použití stačí v rámci tvorby libovolný editor zdrojových kódů a v rámci testování a prohlížení webových prohlížeč. JavaScript má dnes vlastně takový monopol na skriptování ve webovém prohlížeči. (*existuje i možnost jej používat straně serveru, ale tím se zde zabývat nebudeme*).

Vaším úkolem je poznat jeho možnosti a zlehka nahlédnout na jeho syntaxi. Pro tento účel Vám doporučím šikovný seriál na itnetwork.cz:

- <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>

⁸ <https://getbootstrap.com/>

- <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-zaciname-s-javascriptem-prvni-scripty>

Knihovna jQuery

Na základě oblíbeného JavaScriptu existují dnes knihovny, které jeho použití v rámci webu zjednodušují a navíc rozšiřují o snáze použitelnou funkcionalitu. Nejpoužívanější je knihovna jQuery. Její obliba svědčí také v tom, že ji používají takové giganty, jako je Google nebo Microsoft.

Zde bych Vám doporučil stejně jako v případě JavaScriptu seriál na itnetwork.cz.

- <https://www.itnetwork.cz/javascript/jquery-zaklady/javascript-tutorial-funkcionalni-programovani-a-jquery-webova-kalkulacka>

5. Webový server Apache a Programování s jazykem PHP

O webovém serveru už v této chvíli zřejmě něco málo víte. Možná už se Vám i podařilo si jej lokálně zprovoznit na svém počítači. Pokud si vytváříme statické dokumenty z HTML, CSS nebo JS, webový server zatím nepotřebujeme. Situace se změní v momentě, kde chceme své weby prezentovat na internetu. V tento moment musíme využít nabídku poskytovatelů webových serverů (může se jednat o službu placenou nebo i zdarma – např. <https://www.trialhosting.cz/>).

Potřeba webového serveru se rovněž stane akutní v ten moment, kdy budeme chtít své statické dokumenty rozšířit o dynamické prvky realizované pomocí některého ze serverových skriptovacích jazyků – např. PHP. Takové skripty je potřeba nějak efektivně testovat a proto je potřeba si ideálně zřídit lokální webový server. Máme k tomu, jak už víte šikovné aplikace - např. balík XAMPP.

Obecně k problematice webových serverů lze začít např. na Wikipedii:

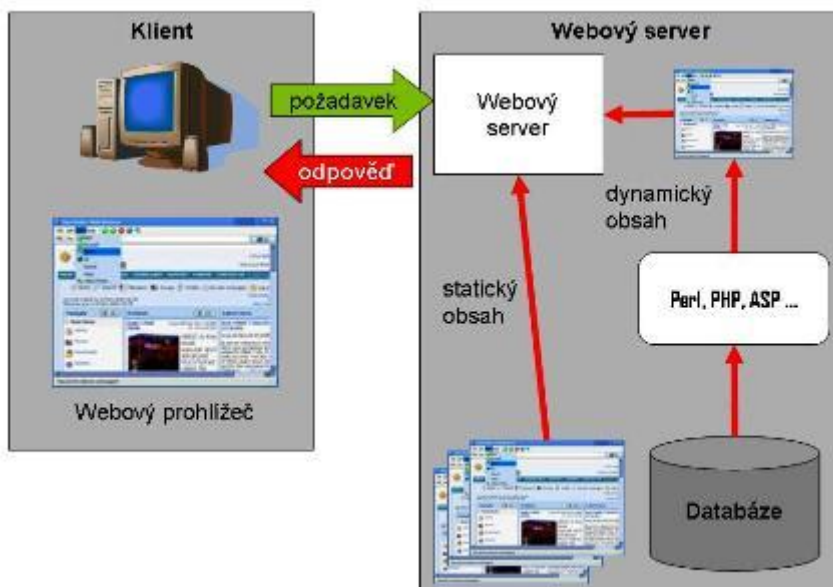
- https://cs.wikipedia.org/wiki/Webov%C3%BD_server

Programování s jazykem PHP

Statické technologie HTML a CSS nám značují obsah a v prvním případě mu dávají kontextový význam, v druhém případě jej graficky formátují. JS nám zase umožňuje sledovat události a v prostředí webového prohlížeče na ně vázat určité akce. Ale tím to v základu končí. Jeli potřeba, aby na straně serveru probíhaly akce, které na základě požadavků uživatele např.:

- Vypíšíou z databáze produkty kategorie eshopu,
- Nebo zpracují data zasláné uživatelem skrze formulář,
- Nebo provedou výpočet pojištění na základě zadaných parametrů, atd.,

je nutné do aplikace doplnit logiku, která tyto úkony zvládne. A pro tyto účely používáme právě serverové skriptovací jazyky. Poměrně jednoduše to znázorňuje následující obrázek.

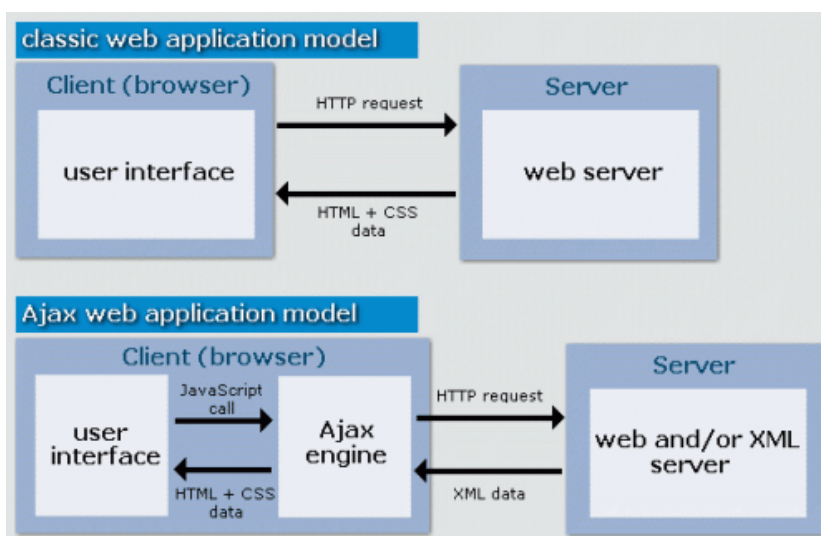


Není po Vás požadováno abyste zvládali celou problematiku tohoto jazyka. Zaměřte se na pochopení jeho základů a obecně účely jeho použití. Základní informace lze čerpat např. zde:

- <http://www.pehapko.cz/zakladni-kurz/4-zaklady-syntaxe>
- <https://www.itnetwork.cz/php/oop/php-tutorial-prvni-objektova-aplikace>

6. AJAX

Klasický požadavek uživatele na konkrétní část webu probíhá tak, že uživatel klikne na odkaz, vznikne tak http požadavek a následně mu server zašle odpověď nebo jej přeměruje na jinou část stránek či aplikace. Takový klasický požadavek dobře poznáme; obnovení aktuální stránky můžeme provést také např. stiskem klávesy F5.



V moderních aplikacích se ale dnes už běžně setkáváme s inovativnější formou načítání obsahu. Např. v internetových obchodech, sociálních sítích či informačních portálech se nový obsah načítá tak nějak nenápadně automaticky, aniž bychom to sami jako uživatelé skoro zaznamenali. Tento způsob načítání obsahu nebo komunikace klienta a serveru tzv. na pozadí má na svědomí zkratka **AJAX**.

Základní informace lze získat zde:

- <https://cs.wikipedia.org/wiki/AJAX>
- <https://www.voxcafe.cz/mindblog/clanky/programovani/jquery---uvod-do-ajax.html>

7. Optimalizace webových stránek SEO

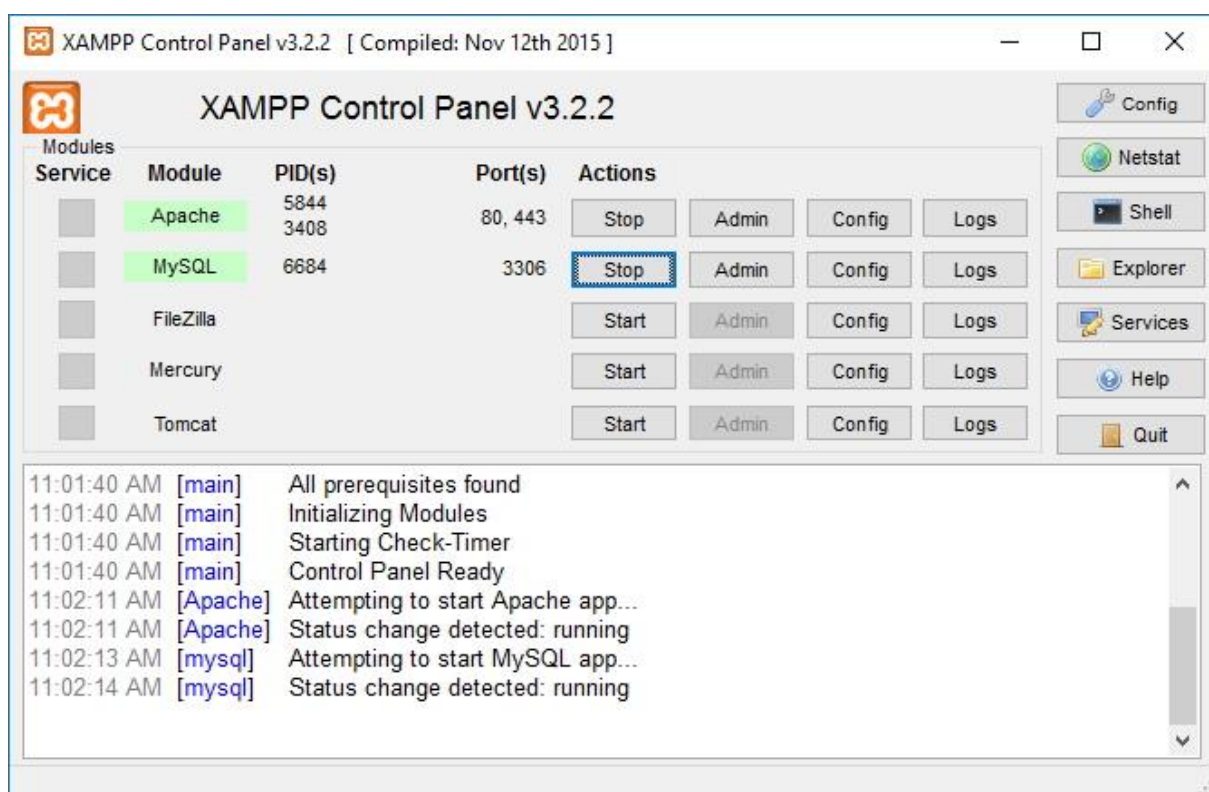
8. Publikace www na internetu (Webhosting, FTP)

Více k problematice SEO a Publikace www najdete v posledních dvou kapitolách studijní opory.

Příloha

Dále Vám přikládám aktualizovaný postup přípravy testovacího a vývojového prostředí pro lokální systém. Při použití jazyka PHP pro tvorbu dynamických částí www stránek, (PHP je serverovým jazykem) je nutné provést přípravu prostředí:

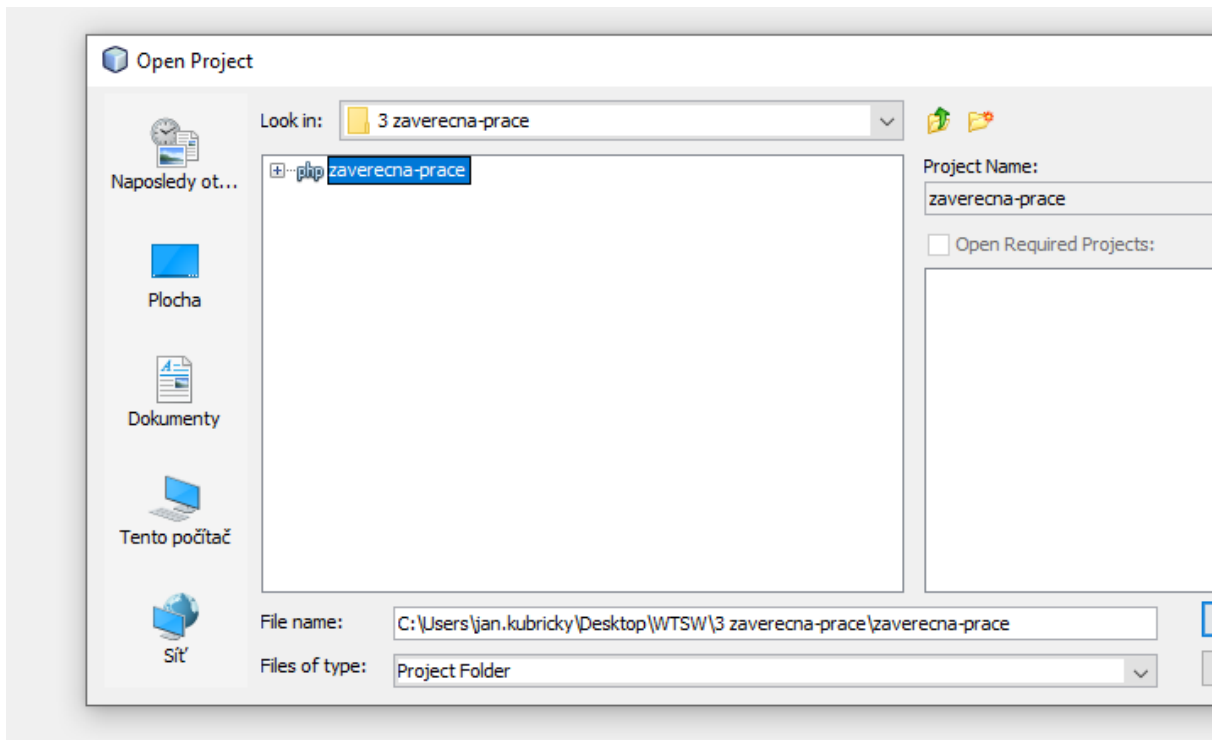
1. **Potřebujete lokální webový server na počítači.** Pro tento účel využijeme webový server **Apache**. Nejsnazší cesta jeho instalace je pomocí aplikace **XAMPP**. <https://www.apachefriends.org/index.html>
2. Instalace může být na Windows doprovázena některými problémy. (např. blokovaný port 80 protokolu http). Návod, jak si s nimi poradit najdete zde: <https://pureinfotech.com/install-xampp-windows-10/>
3. Pokud bude instalace úspěšná, server Apache se spouští pomocí nástroje **xampp-control.exe**, který najdete ve složce xampp. *Nejčastěji C:/xampp/xampp-control.exe*



- 4.
5. Rozbalenou složku *projekt* umístíte do adresáře **xampp/htdocs**⁹ a tím je vše připraveno ke spuštění. V prohlížeči zadáte adresu: <http://localhost/projekt>
6. Teď už jenom potřebujete program resp. vývojové prostředí, ve kterém budete moci prohlížet, upravovat a nebo i dále vyvíjet vaši práci. Zde doporučuji IDE **Netbeans**. <https://netbeans.org/>

⁹ Lokální webový server otestujeme v prohlížeči url ve tvaru: <http://localhost>

7. Projekt závěrečné práce už obsahuje i init projektu v Netbeans, takže stačí Netbeans spustit a zvolit volbu Otevřít projekt. Nabídka *File > Open Project* (a vyhledat projekt v `xampp/htdocs/projekt`)



8. Vlastní publikování www stránek – další informace prosím ve studijní opoře.

