



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

Semestrální práce KIW/WEB

Konferenční web

Dominik Zappe (A20B0279P)

E-mail: zapped99@students.zcu.cz

Datum vytvoření: 6. 11. 2021

Datum předvedení: 18. 11. 2021

Použité technologie

Při vytváření práce bylo využito několik jazyků. Samozřejmostí je jazyk **PHP**, ten je užit hlavně k ovládání prvků spojených s databází a k zpracovávání uživatelem odeslaných formulářů. Tento jazyk je jakousi kostrou celé aplikace.

Po zmínění databáze je nutné uvést, že databáze je celá v jazyce **MYSQL**. S tím i souvisí ošetření proti **SQL INJECTION** neboli předpřipravené dotazy (samozřejmostí je také **XSS** ošetření). Užité dotazy v práci jsou např. **SELECT**, **UPDATE** nebo **DELETE**.

Následuje jazyk **HTML** a sním spojený **TWIG**, tyto dvě technologie jsou užity pro správné zobrazení stránek na straně uživatele. Twig si navíc např. hlídá, zda je uživatel přihlášen a jestli má dostatečná práva pro zobrazení daných stránek. Dále např. počítá průměrné hodnocení daného článku.

Se zobrazením uživateli jsou také spojené **SCRIPTY**, v této práci byl užit jak čistý **JAVASCRIPT**, tak **JQUERY**, tak i trochu **AJAX**. JavaScript je užit hojně, zajišťuje funkčnosti jako zobrazení WYSIWIG textového editoru, když uživatel chce změnit svůj článek, nebo např. zajišťuje možnost smazat uživatele nebo článek. JavaScript úzce souvisí s JQuery a jejich užití v kódu je zaměnitelné. AJAX je využit pouze na jednom místě a tím je profilová fotka uživatele na hlavním menu. Nejdříve se načte defaultní fotka, která je přiřazená všem uživatelům a má-li přihlášený uživatel nastavenou nějakou svou profilovou fotku, tak asynchronní JavaScript pošle tuto informaci na server a požádá o překreslení fotky.

Výše už byl se scripty zmíněn i **WYSIWIG** editor **CKEditor**. Tento textový editor je užit při psaní abstraktu článku.

Další využití technologie jsou **BOOTSTRAP** a **FONTAWESOME**. Tyto dvě technologie byly využity na webu hojně. Bootstrap hlavně pro responzivní design a Font Awesome hlavně pro ikonky. Spolu s Bootstrapem souvisí i **CSS**, které moc využito nebylo, hlavně z důvodu, že Bootstrap jaksi zastřešuje, nebo nahrazuje, CSS.

Předposlední využitou technologií je **COMPOSER**, který všechny předchozí technologie propojuje, jelikož se stará o stažení jejich správných verzí, které spolu fungují.

Poslední využitou technologií je verzovací systém **GIT**, tato práce je celá na GitHubu, kde jsou vidět jednotlivé verze a co jak bylo přidáváno a upravováno.

Adresářová struktura

Hlavní tři adresáře jsou **controllers**, **models** a **views**. Proč zrovna tyto tři jsou nejdůležitější, bude detailněji popsáno v další kapitole o architektuře.

Adresář **controllers** má v sobě PHP třídy ovladačů, jeden je abstraktní – *AController.abstract.php* a zbylé dědí od tohoto abstraktního. Obecně ovladače zpracovávají formuláře od uživatele a zajišťují získání dat z modelů pro správné vykreslení stránky pomocí pohledů.

Adresář **models** obsahuje třídu pro správu *Session*, třídu pro správu uživatelských přihlášení, registraci a odhlášení, tedy práce se session a databází, a také třídu pro obecnou práci s databází.

Adresář **views** shromažďuje všechny soubory s příponou *.twig* neboli šablony. Šablony se pouze starají o správné vykreslení stránek na straně uživatele na základě dat, která mají od svého příslušného ovladače.

Další adresář je **uploads**. Tento adresář obsahuje dva další podadresáře, a to **article** a **avatar**. Jak už název napovídá, jedná se o hlavní adresář, kam se dostávají nahrané soubory od uživatelů. Do podadresáře **article** se nahrávají soubory s příponou *.pdf* a jedná se o celé články uživatelů. Do podadresáře **avatar** se nahrávají obrázkové soubory (*.png .jpg .jpeg* a *.gif*), které představují profilové obrázky uživatelů.

V neposlední řadě ještě existuje adresář **composer**. V něm se nachází *composer.json*, který je spojen s výše popsanou technologií composer.

Architektura aplikace

Aplikace využívá architekturu **MVC** (*Models Views Controllers*). Tato architektura prakticky znamená, že aplikace má být rozdělena do tří vrstev. Vrstva ovladačů, pohledů a modelů. Ovladače obecně zajišťují zpracování požadavku od uživatele a získání dat od příslušného modelu a dále poslání těchto dat do pohledu, který se zobrazuje uživateli. Modely komunikují s ovladači a databází a přeposílají mezi nimi data. Pohledy už jen zajišťují správnost vzhledu pro uživatele a správnost zpracování příslušných dat. Pohled už může buď přímo komunikovat s uživatelem (jednostranně), nebo může své zpracování poslat ovladači, tak je možno kompletně odstínit uživatele od celé aplikace a dát mu přístup pouze přes jeden jediný ovladač – např. *rozcestník*.

Tuto architekturu zajišťuje dříve popsaná adresářová struktura.

Seznam defaultních uživatelů

Login	Heslo
SuperAdmin	admin
Admin	admin
Recenzent1	recenzent
Autor1	autor

Závěr

Při vytváření této práce jsem byl schopen si vyzkoušet všechny požadované technologie a částečně si jich i většinu osvojit. Práce je psaná objektově orientovaně, tudíž je dále jednoduše rozšířitelná. Nikdy předtím jsem s PHP moc nepracoval, a tak jsem rád, že jsem měl možnost si to zkusit.