



**FAKULTA  
APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ  
UNIVERZITY  
V PLZNI**

# Semestrální práce KIV/WEB

Webová stránka environmentální konference

Michaela Janková

[jankovam@students.kiv.zcu.cz](mailto:jankovam@students.kiv.zcu.cz)

18. 12. 2019

## Použité technologie

### HTML

Hypertext Markup Language je standardní značkovací jazyk pro dokumenty zobrazované ve webových prohlížečích. Často je doplňován kaskádovými styly (CSS) a skriptovacími jazyky (např. Javascript). HTML slouží k vytvoření strukturovaného dokumentu pomocí značek (tzv. tagů), kterými lze text rozdělit do odstavců, vytvořit nadpis, seznamy, odkaz, vložit obrázek a další. Webový prohlížeč tyto značky použije k formátování zobrazovaného textu.

### CSS

Cascading Style Sheets, v překladu kaskádové styly, se používají k popisu způsobu zobrazení HTML, XHTML, nebo XML dokumentů. Hlavní důvod používání tkví v oddělení vzhledu dokumentu od jeho struktury a obsahu.

### PHP

PHP (Hypertext Preprocessor, původně Personal Home Page) je skriptovací programovací jazyk určený primárně k programování dynamických internetových stránek a webových aplikací, ale lze ho využít i k tvorbě konzolových či desktopových aplikací.

### MySQL

My Structured Query Language je strukturovaný dotazovací jazyk, který je používám pro práci s daty v relačních databázích. MySQL je multiplatformní databáze, ve které komunikace probíhá pomocí dialektu jazyka SQL s některými rozšířeními.

### Bootstrap

Bootstrap je jednoduchá, volně stažitelná sada nástrojů pro tvorbu responzivního webu a webových aplikací. Obsahuje návrhářské šablony sloužící pro úpravu typografie, formulářů, tlačítek a dalších komponent rozhraní. Cílem používání Bootstrapu je zjednodušení vývoje webových stránek.

Text je formátován pomocí jazyka HTML, na vzhledu samotné stránky se podílí CSS a framework Bootstrap. Bootstrap také zajišťuje responzivitu webových stránek. Celý backend je napsaný v PHP, databázi jsem zvolila MySQL.

## Adresářová struktura aplikace

Celá aplikace se nachází ve složce *Semestrálka WEB*.

Jejími podadresáři jsou:

- Views – obsahuje veškeré soubory, které jsou třeba pro zobrazení aplikace. Obsahuje soubor s kaskádovým stylem pro webové stránky, veškeré PHP třídy s šablonami jednotlivých stránek
- Models – obsahuje veškeré třídy pro práci s databází pomocí PDO
- Controllers – tato složka obsahuje třídy, které pracují jako prostředníci mezi třídami z Views a Models. Přijímají požadavky z Views, posílají požadavky do Models, výsledky těchto požadavků zpracují a pošlou do Views výsledek.
- Database – v této složce se nachází dump databáze ve formátu SQL, pro její instalaci na jiném serveru.
- uploads – v této složce se v dalších podsložkách (podle ID uživatelů) ukládají nahrané soubory k jednotlivým článkům

Dokumenty v této složce jsou soubor index.php a settings.php, ve kterém se nachází základní nastavení aplikace, jako jsou informace k připojení k databázi a posledním je tento dokument.

## Architektura aplikace

Vstupním bodem aplikace je soubor index.php, který podle proměnné z \$\_GET vybere příslušný kontroler požadované stránky. Pokud není žádná vybraná, nebo pokud požadovaná stránka neexistuje, je zobrazena výchozí stránka (v tomto případě hlavní stránka webu). Kontroler načte šablonu, načte příslušná data z databáze, obojí zpracuje a zobrazí výsledek uživateli. Výsledek se liší podle toho, jestli je uživatel přihlášen, nebo podle role přihlášeného uživatele. Požadovanou stránku uživatel vybírá pomocí menu, případně odkazem na jinou stránku.

## Views

### Hlavní stránka (HomePage.php)

- Na hlavní stránce se nachází základní informace o konferenci a menu, které odkazuje na ostatní stránky aplikace

### Články (Articles.php)

- Pro nepřihlášeného uživatele se zobrazí články, které jsou již publikované
- Pro přihlášeného uživatele se zobrazí všechny články, u článků, jejichž je autorem, navíc vidí stav (v řízení/přiját/nepřiját), své články také může upravovat a mazat
  - o V případě úpravy článku je uživatel přesměrován na stránku UpdateArticle.php, kam se načte z databáze název a obsah článku a uživatel může provést a uložit změny

### Recenze (Reviews.php)

- Pro nepřihlášeného uživatele, autora, nebo správce vypíše pouze větu, že tato stránka je přístupná pouze recenzentům
- pro recenzenta se z databáze načtou recenzentovy recenze a následně zobrazí
  - o recenzent může recenze upravovat a mazat, pokud recenzovaný článek ještě nebyl zveřejněn
    - v případě úpravy je stejně jako u článků přesměrován na novou stránku UpdateReview.php, kde se tato recenze zobrazí, uživatel ji může upravit a následně uložit

### Administrace (Administration.php)

- je přístupná pouze pro správce
- správci se zobrazí administrace uživatelů, pod nadpisem je odkaz na administraci článků
- pomocí UserModel a LoginController se načte z databáze tabulka uživatelů. V ní se dá upravovat role uživatele, jestli je uživatel zabanován, případně lze uživatele smazat.
- Při přesměrování na administraci článků (AdministrationArticles.php)
  - o Se objeví tabulka článků a odkaz zpět na administraci uživatelů
  - o U článku lze změnit stav, a zda je příspěvek veřejný, lze se podívat na recenze k danému článku
    - Vypíší se všechny již existující a přiřazené recenze a zároveň lze přiřadit další recenzi někomu dalšímu

### Přihlášení (Login.php)

- Přihlášení a odhlášení uživatele

### Registrace (Registration.php)

- Registrace nového uživatele pomocí jména a hesla
  - o Jméno musí být unikátní

## Controllers

### LoginController.php

- V metodě work() načítá šablonu Login.php a obsluhuje a zpracovává požadavky jako ostatní kontrolery
  - o Má i další metody, které využívají ostatní kontrolery, pokud chtějí data z databáze z tabulky user, nebo se session
- Obsluhuje práci se session pomocí instance SessionControlleru
- Odesílá požadavky do UserModel.php

### ArticleController.php

- V metodě work() načítá šablonu Articles.php a obsluhuje a zpracovává požadavky jako ostatní kontrolery
  - o Stejně jako LoginController má i další metody v případě nutnosti získání dat z tabulky article
- Odesílá požadavky do ArticleModel.php

### ReviewController.php

- V metodě work() načítá šablonu Reviews.php a obsluhuje a zpracovává požadavky jako ostatní kontrolery
  - o Také má další metody, které ostatní kontrolery využívají pokud potřebují data z databáze, tabulky review
- Odesílá požadavky do ReviewModel.php

### SessionController.php

- Čte, zapisuje a odstraňuje data v \$\_SESSION
- Zjišťuje, jestli je vůbec nastavena

### AdministrationArticlesController.php, HomePageController.php, NewArticleController.php, RegistrationController.php, UpdateArticleController.php, UpdateReviewController.php

- Kontrolery pro příslušnou třídu (stejný název bez Controller)
- V případě potřeby (pokud potřebují přístup k datům z některé z tabulek) mají proměnné typu LoginController, ArticleController nebo ReviewController

## Models

### DatabaseModel.php

- Zajišťuje připojení k databázi
- Spouští funkci prepare pro předpřipravené dotazy
- Má ještě dvě jiné funkce, executeQueryBool(), která vrací pouze hodnotu true a false podle toho, zda dotaz proběhl v pořádku a executeQuery(), která je volána metodou executeQueryBool(), nebo přímo z kódu ostatních modelů. Ta zajišťuje spuštění dotazu pomocí metody query() a následně vrací buď výsledek, nebo vypíše chybové hlášení a vrátí NULL.

### ArticleModel.php

- Obsluhuje požadavky z ArticleControlleru a připravené či předpřipravené dotazy posílá do DatabaseModelu, který je spustí a vrátí výsledek

### UserModel.php

- Obsluhuje požadavky LoginControlleru a připravené či předpřipravené dotazy posílá do DatabaseModelu, který je spustí a vrátí výsledek

### ReviewModel.php

- Obsluhuje požadavky z ReviewControlleru a připravené či předpřipravené dotazy posílá do DatabaseModelu, který je spustí a vrátí výsledek