



KIV/WEB

---

# Semestrální práce, Webová konference

---

*Autor*

JAN RYCHLÍK, A16B0125, RYCHLIKJ@STUDENTS.ZCU.CZ

25. ledna 2018

# Obsah

<b>1</b>	<b>Zadání</b>	<b>2</b>
1.1	Popis zadání . . . . .	2
1.2	Popis činnosti webových stránek . . . . .	2
1.3	Nutné požadavky semestrální práce . . . . .	3
<b>2</b>	<b>Popis použitých technologií</b>	<b>4</b>
2.1	HTML . . . . .	4
2.2	CSS . . . . .	4
2.3	PHP . . . . .	4
2.4	MySQL . . . . .	4
2.5	Twig . . . . .	5
2.6	JavaScript . . . . .	5
2.7	Bootstrap . . . . .	5
<b>3</b>	<b>Adresářová struktura</b>	<b>6</b>
3.1	controllers . . . . .	6
3.2	css . . . . .	6
3.3	Models . . . . .	6
3.4	Rescouse . . . . .	6
3.5	Sablon . . . . .	6
3.6	views . . . . .	7
<b>4</b>	<b>Architektura aplikace</b>	<b>8</b>
4.1	Controllers . . . . .	8
4.2	Models . . . . .	8
4.3	view . . . . .	8
<b>5</b>	<b>Závěr</b>	<b>9</b>

# Kapitola 1

## Zadání

### 1.1 Popis zadání

Standartním zadáním semestrální práce z WEBu bylo vytvořit webové stránky na námi zvolené téma.

### 1.2 Popis činnosti webových stránek

- Uživatelé systému budou autoři příspěvků (vkládají abstrakty a PDF dokumenty), recenzenti příspěvků (hodnotí příspěvky) a administrátoři (spravují uživatele, přiřazují příspěvky recenzentům a rozhodují o publikování příspěvků). Každý uživatel se bude do systému přihlašovat prostřednictvím uživatelského jména a hesla.
- Nepřihlášený uživatel vidí pouze publikované příspěvky.
- Nový uživatel se bude moci zaregistrovat, čímž získá status autora.
- Přihlášený autor vidí svoje příspěvky a stav, ve kterém se nacházejí (v recenzním řízení / přijat +hodnocení / odmítnut +hodnocení). Příspěvky může přidávat, editovat a volitelně i mazat.
- Přihlášený recenzent vidí příspěvky, které mu byly přiděleny k recenzi, a může je hodnotit (alespoň 3 kritéria). Pokud příspěvek nebyl dosud schválen, tak své hodnocení může změnit.

- Administrátor spravuje uživatele (určuje jejich role a může uživatele zablokovat či smazat), přiřazuje neschválené příspěvky recenzentům k ohodnocení (každý příspěvek bude recenzován minimálně třemi recenzenty) a na základě recenzí rozhoduje o přijetí nebo odmítnutí příspěvku. Přijaté příspěvky jsou automaticky publikovány ve veřejné části webu.
- Databáze musí obsahovat alespoň 3 tabulky dostatečně naplněné daty pro předvedení funkčnosti aplikace.

### 1.3 Nutné požadavky semestrální práce

- Práce musí být osobně předvedena cvičícímu a po schválení odevzdána na CourseWare či Portál.
- K práci musí být dodána dokumentace (viz dále) a skripty pro instalaci databáze (např. získané exportem databáze).
- Web musí dodržovat MVC architekturu.
- Pro práci s databází musí být využito PDO nebo jeho ekvivalent a používány předpřipravené dotazy (prepared statements).
- Web musí obsahovat responzivní design.
- Web musí obsahovat ošetření proti základním typům útoku (XSS, SQL injection).
- Web musí fungovat i s "ošklivými" URL adresami.
- Aplikaci není možné realizovat s využitím PHP frameworků (zakázáno např. Nette, Symfony atd.).
- Front-end je vhodné realizovat s využitím frameworku Bootstrap (get-bootstrap.com).

# Kapitola 2

## Popis použitých technologií

Základem semestrální práce je struktura MVC. Použité technologiie webu jsou: HTML, CSS, PHP, Twig, JavaScript a MySQL. Bootstrap byl použit pro responzivní vzhled stránky a rozmístění prvků na stránce.

### 2.1 HTML

**HTML5** - vzhled stránky, složka view.

### 2.2 CSS

**CSS3** - práce s HTML, pouze ve složce css.

### 2.3 PHP

**PHP** - ve složce models užito pro komunikaci s databází, ve složce controllers na testování vstupů a dokumentace v kódu.

### 2.4 MySQL

**MySQL**- vytvoření databáze.

## 2.5 Twig

**Twig** - složka sablon, sablon.html (obsahuje odkazy na používané technologie).

## 2.6 JavaScript

**JavaScript**- zobrazení minimapy, v záložce Kontakty.

## 2.7 Bootstrap

**Bootstrap** - spravuje vzhled stránky, nejvíce souvisí se složkou view

# Kapitola 3

## Adresářová struktura

### 3.1 controllers

**controllers** - obsahuje controllery, které získávají data z modelů.

### 3.2 css

**css** -Css styly aplikace.

### 3.3 Models

**models** - všechny kódy k práci s databází, včetně logických operací.

### 3.4 Rescouse

**Rescouse** - složka, kam se ukládají pdf soubory nahrané na stránce.

### 3.5 Sablon

**Sablon** - ve složce je kód, který určuje vzhled stránky, včetně odkatů na použité prvky z webu.

## 3.6 views

**views** - složka, kde jsou uloženy jednotlivé kódy k zobrazení jednotlivých částí semestrální práce.



# Kapitola 4

## Architektura aplikace

### 4.1 Controllers

Obsahuje tři třídy **homeController**, **newsController** a **useControler**  
**homeController** - zobrazuje titulní stránku s příspěvky.  
**newsController** - zobrazuje příspěvky podle přihlášeného uživatele.  
**userController** - kontroluje přihlášení uživatele. K administrátorovy se chová stejně jako k uživateli s tím že, v potřebných funkcích má administrátor přidanou výjimku.

### 4.2 Models

Obsahuje dvě třídy **db** a **phpWrapper**. Provádí se zde veškeré logické operace, včetně načítání z databází.

### 4.3 view

Obsahuje několik tříd, určených pro vzhled stránky.

# Kapitola 5

## Závěr

Semestrální práce splňuje všechny povinné části zadání a některé nepovinné (např. práce je uložena na **Github**). Uživatel se může registrovat nebo přihlásit, poté může přidávat, hodnotit nebo mazat některé příspěvky. Administrátor může evidovat uživatele, blokovat je nebo mazat. U příspěvků může rozhodnout, zda ho smazat, určit kdo ho bude hodnotit nebo určit, zda se zobrazí na fóru.