



**FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI**

---

**KIV/WEB  
SEMESTRÁLNÍ PRÁCE  
STANDARNÍ ZADÁNÍ 1 - KONFERENČNÍ SYSTÉM**

---

Michaela Benešová - A19B0008P

benesomi@students.zcu.cz

20.12.2020

## Obsah

<b>1</b>	<b>Použité technologie</b>	<b>3</b>
1.1	PHP	3
1.2	HTML	3
1.3	CSS	3
1.4	Bootstrap a FontAwesome	3
1.5	JavaScript	3
1.6	MySQL Workbench a MySQL	3
<b>2</b>	<b>Popis adresářové struktury</b>	<b>4</b>
2.1	App	4
2.1.1	Controllers	4
2.1.2	Models	4
2.1.3	Views	4
2.1.4	WebStyle	4
2.2	Pictures	4
2.3	FilesFromUsers	4
2.4	Index, Settings, ApplicationStart	4
2.5	Adresář HTML + .htaccess	4
<b>3</b>	<b>Architektura aplikace</b>	<b>5</b>

# 1 Použité technologie

V semestrální práci jsem využila PHP, HTML, CSS, JavaScript, MySQL Workbench, Bootstrap a FontAwesome.

## 1.1 PHP

Tato technologie se prolíná všemi vrstvami mého projektu. Slouží například k vypsání příslušné šablony stránky.

## 1.2 HTML

Stěžejní prvek pro vzhled webu. Většinou používán v částech s PHP. Vytváří se pomocí něj tabulky, formuláře a další náležitosti stránek. Jeho využití se dá opět najít ve všech vrstvách.

## 1.3 CSS

CSS slouží především k úpravě jednotlivých textů a celkového vzhledu stránek. Využívá se především ve své vlastní třídě označené .css. Ale zároveň se dá najít definovaný u nějakých objektů.

## 1.4 Bootstrap a FontAwesome

Bootstrap nabízí nespočet vzhledových možností a FontAwesome nespočet ikonek. Oboje jsem využívala na různých místech. Bootstrap například při dělání navigačního menu a FontAwesome při dělání článků, kde mi název článku zdobí ikonka stromčku.

## 1.5 JavaScript

Využíván zřídka, ale přesto velmi důležitý pro celý projekt. Využívá se v templatech jednotlivých stránek. Například ho využívám při otevírání článků, abych dokázala identifikovat jaký článek se má otevřít.

## 1.6 MySQL Workbench a MySQL

Slouží ke spojení webu s databází. Zároveň v MySQL Workbench byla sestavena databáze pro můj web. Databázové metody mají svoji vlastní třídu ve vrstvě "Models" a k volání metod dochází v "Controllers".

## 2 Popis adresářové struktury

### 2.1 App

Adresář s rozdělením na základní MVC strukturu, zároveň se v něm nachází složka týkající se vzhledu, vypsání chybové stránky (2.5) a ApplicationStart (2.4).

#### 2.1.1 Controllers

Zde se nachází kontrolery pro jednotlivé stránky.

#### 2.1.2 Models

Zde najdeme třídu s příslušnými databázovými metodami a třídu pracující se session. Tyto třídy jsou navzájem propojeny, kdy při nějakých metodách v databázové části dochází k volání metod ze session.

#### 2.1.3 Views

Adresář s šablonami k jednotlivým stránkám.

#### 2.1.4 WebStyle

Zde se nachází .css soubor se styly pro jednotlivé části webu.

### 2.2 Pictures

Obrázky, které se využívají na stránkách.

### 2.3 FilesFromUsers

Soubory, které autoři nahráli jako součást svých článků.

### 2.4 Index, Settings, ApplicationStart

První zmíněný Index načte nastavení webu (Settings) a třídu spouštějící aplikaci (ApplicationStart) a spustí aplikaci přes metodu appStart().

### 2.5 Adresář HTML + .htaccess

Slouží k vypsání chybové stránky.

### 3 Architektura aplikace

Jak jsem již zmínila, můj web je postaven na základní MVC struktuře. Tato struktura se skládá ze tří částí - **model, view a controller**.

Každá vrstva má svoji vlastní složku a navzájem spolu komunikují.

Model využívá komunikaci se session a databází.

View se stará o vzhled stránek.

Controller volá příslušné metody z modelu a získává z nich potřebné data.

Jednoduše by se komunikace mezi nimi dala popsat takto:

V controlleru dojde k zavolání metody show, která připojí šablonu, čímž ji i vykona. Přesněji se připojí template pro danou stránku (začneme se tedy nacházet ve view). Zde dojde k vypsání všech potřebných věcí. Následně se vracíme zpět do kontroleru a dojde k získání šablony, která se následně uloží do proměnné obsah a vypíše. Pokud by se nacházely na stránce nějaké formuláře či tlačítka, k jejich zpracování dochází vždy v kontroleru.