

Semestrální práce KIV/WEB

Standardní zadání 1 - Konferenční systém

Pavel Třeštík
trestikp@students.zcu.cz
A17B0380P

18. prosince 2019

Obsah

1	Popis použitých technologií	1
1.1	HTML	1
1.2	Twig	1
1.3	PHP	1
1.4	JavaScript	1
1.5	jQuery	1
1.6	AJAX	1
1.7	Bootstrap	2
1.8	CSS	2
2	Popis adresářové struktury	2
2.1	app	3
2.1.1	controller	3
2.1.2	core	3
2.1.3	inc	3
2.1.4	model	4
2.1.5	uploads	4
2.1.6	vendor	4
2.1.7	view	4
2.2	public	4
2.3	DB	4
2.4	DOC	4
3	Popis architektury aplikace	5
3.1	View	5
3.2	Controller	6
3.3	Model	6
4	Závěr	6

1 Popis použitých technologií

1.1 HTML

Je značkovací jazyk. Používá se k zobrazování webu. Je použit v šablonách a obsahu generovaného do těchto šablon.

1.2 Twig

Je šablonovací systém pro PHP. Umožňuje jednoduše posílat data z PHP do připravené šablony.

1.3 PHP

Je skriptovací programovací jazyk sloužící především k programování dynamického webového obsahu. Slouží jako prostředník mezi HTML a databází. V této práci je většinou používán k získání dat z databáze a následného vygenerování HTML prezentující tato data.

1.4 JavaScript

Jedná se o programovací jazyk, který lze snadno využít k práci s webovými stránkami. V této práci JavaScript není moc využit ve své originální formě, ale je častěji použita jeho knihovna jQuery.

1.5 jQuery

Jak je zmíněno výše, jedná se o JavaScript knihovnu, která značně ulehčuje komunikaci mezi JavaScriptem a HTML. V této práci je velmi běžně využita místo čistého JavaScriptu a AJAX requestů.

1.6 AJAX

Anglicky "Asynchronous JavaScript And XML" dovoluje stránkám asynchroní komunikaci se serverem. Je využit pro všechny formuláře této práce a také pro akce některých tlačítek. Díky tomu že je komunikace se serverem asynchroní neobnovuje stránku.

1.7 Bootstrap

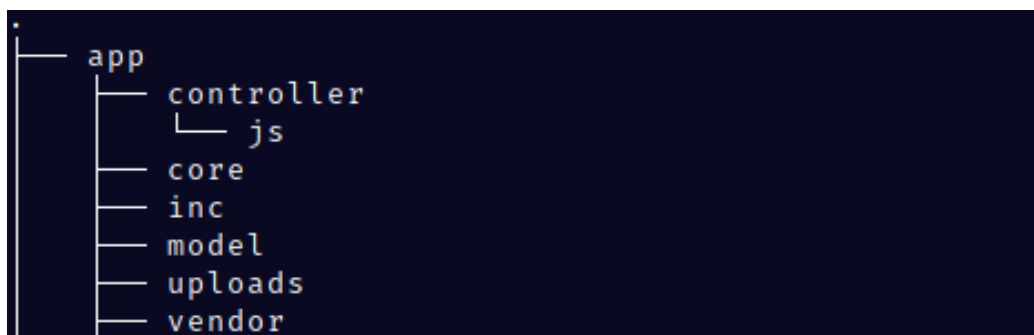
Bootstrap je sada nástrojů použitelná k usnadnění webového vývoje. Obsahuje předpřipravené možnosti - šablony pro HTML a CSS a navíc poskytuje i rozšíření JavaScriptu. V této práci je použit pouze pro design stránky, takže je použit pouze pro CSS.

1.8 CSS

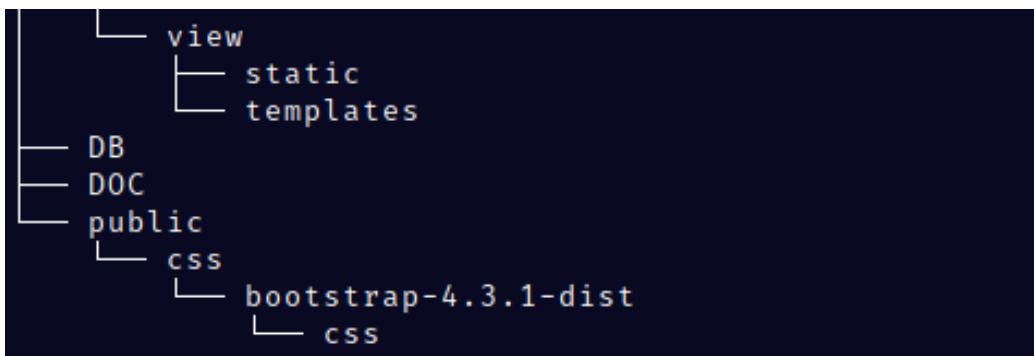
Jsou kaskádové styly měnící vzhled HTML elementů. Vzhledem k tomu, že v této práci je využit Bootstrap jsou téměř nepoužity. Práce obsahuje pouze jeden .css soubor a mimo něj je css použito u některých elementů tzv. inline - přímo v tagu elementu.

2 Popis adresářové struktury

Adresářová struktura je znázorněna následujícími obrázky.



Obrázek 1: První část adresářové struktury



Obrázek 2: Druhá část adresářové struktury

Mezi Obrázekem 1 a Obrázekem 2 jsou podadresáře adresáře **vendor**. Ovšem tento adresář a i jeho podadresáře jsou adresáře šablonového systému twig, proto jsou vynechány.

2.1 app

Obsahuje všechny podstatné soubory a je srdce celé aplikace.

2.1.1 controller

Obsahuje PHP soubory, které manipulují data. Tento adresář má také podadresář **js**, který obsahuje JavaScript soubory, které doplňují funkčnost PHP souborů.

2.1.2 core

Obsahuje hlavní soubory aplikace. Tyto soubory se starají o vytváření a volání příslušných controllerů.

2.1.3 inc

Adresář myšlený pro knihovny a podobné soubory, které poskytují funkce či informace. V této práci obsahuje pouze jeden soubor, který uchovává informace o příslušné databázi.

2.1.4 model

Obsahuje soubor, který se stará data aplikace.

2.1.5 uploads

Adresář pro uživatelské uploady. Web umožňuje uživateli k příspěvku připnout pdf soubor, který se ukládá do tohoto adresáře.

2.1.6 vendor

Adresář šablonového systému twig.

2.1.7 view

Obsahuje 2 podadresáře.

static Obsahuje HTML, které se nemění.

templates Obsahuje HTML šablony, jejichž obsah se mění během běhu aplikace.

2.2 public

Obsahuje přístupné soubory. Nachází se zde pouze index.php a podadresář **css**, který obsahuje css soubor a Bootstrap.

2.3 DB

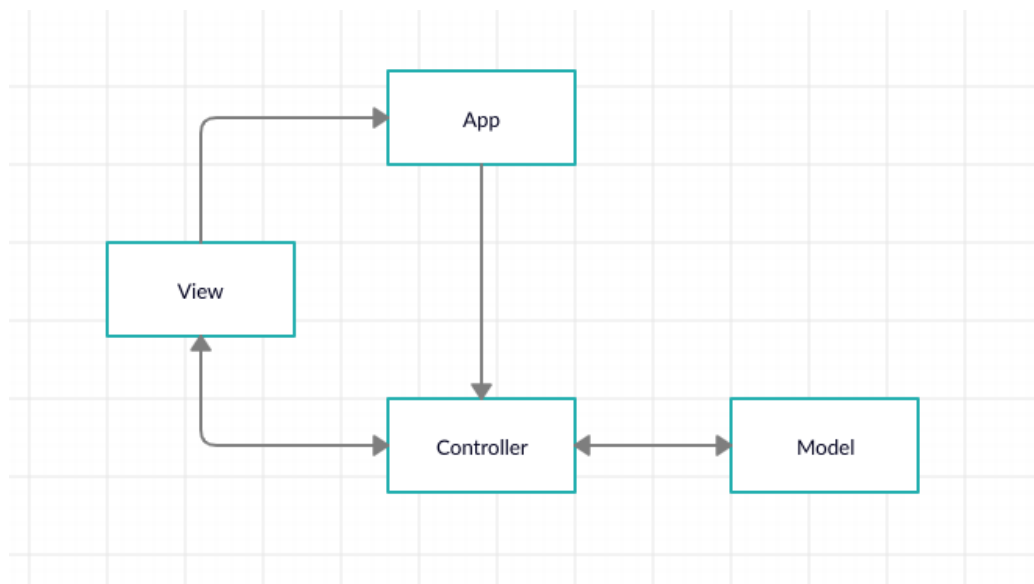
Obsahuje sql soubor, který vytvoří tabulky naplněné daty. Vytváří pouze tabulky a naplňuje je daty, databáze musí být vytvořena předem.

2.4 DOC

Obsahuje dokumentaci.

3 Popis architektury aplikace

Hlavní třídy jsou znázorněny na následujícím obrázku. Práce se snaží dodržovat architekturu MVC, ale tato struktura nejspíš není kompletně dodržena.



Obrázek 3: UML diagram hlavních tříd

Jak je znázorněno na Obrázku 3 uživatel vidí **View**, pokud se má načíst jiná stránka třída **App** rozparsuje URL a vytvoří příslušný controller. Z tohoto důvodu je třída v UML diagramu i přesto že by se dala považovat za část controlleru, jelikož provádí akci vyvolanou uživatelským vstupem.

3.1 View

Je to co uživatel vidí. U většiny stránek této práce se jedná o buď statický obsah z adresáře `app/view/static` nebo dynamický obsah vygenerovaný některým z controllerů. K zobrazení dynamického obsahu některé controllery využívají šablony z adresáře `app/view/templates`.

3.2 Controller

Každá stránka má PHP controller. Některé controllery mají navíc JavaScript soubory, které zajišťují lepší činnost a funkci. Ačkoliv se jedná o více souborů jsou myšleny tak, že tvoří jeden controller pro jednu stránku/činnost. Některé prvky stránky mají vlastní controller i přesto, že jsou to pouze malé části jedné či více stránek. Příklad takového prvku je přihlašovací formulář.

Controllery dále zajišťují komunikaci s **Modelem**, který získává data z databáze. Controllery buď získávají data a následně je zobrazují uživateli a nebo **Modelu** data předávají.

3.3 Model

Zajišťuje manipulaci s daty. Získává, aktualizuje a ukládá data, které mu předá controller. Používá PHP a pro zajištění bezpečnosti používá PDO (PHP Data Objects).

4 Závěr

Vytvořené stránky splňují požadavky zadání. Ovšem bezesporu se na nich najde spoustu věcí, které by se daly vylepšit. Ať už se jedná o design stránek či přidání další funkčnosti.

Práce bude k dispozici na GitHubu na následující adrese: https://github.com/trestikp/web_semestral. Prozatím je tento repozitář privátní a bude zveřejněn až po předvedení práce.