

ZÁPADOČESKÁ UNIVERZITA

FAKULTA APLIKOVANÝCH VĚD

KATEDRA INFORMATIKY A VÝPOČETNÍ TECHNIKY

KIV/WEB - Semestrální práce

Standardní zadání 1 - Knižní konference

Jan Kohout

Obsah

1	Použité technologie	2
1.1	HTML (Hypertext Markup Language)	2
1.2	CSS (Cascading Style Sheets)	2
1.3	Bootstrap	2
1.4	JavaScript	2
1.5	jQuery	2
1.6	AJAX (Asynchronous JavaScript and XML)	2
1.7	Twig	3
1.8	PHP	3
1.9	Composer	3
2	Adresářová struktura	4
2.1	Kořenový adresář	4
2.2	Adresář controller	4
2.3	Adresář css	4
2.4	Adresář db	4
2.5	Adresář dependencies	4
2.6	Adresář doc	4
2.7	Adresář js	4
2.8	Adresář model	4
2.9	Adresář uploads	5
2.10	Adresář view	5
3	Architektura aplikace	6
3.1	Model	6
3.2	View	6
3.3	Application	6
3.4	Controller	6
4	Závěr	8

1 Použité technologie

1.1 HTML (Hypertext Markup Language)

Značkovací jazyk pro tvorbu webových stránek. Slouží k nadefinování základního rozvržení a vzhledu webové stránky. Pro tvorbu semestrální práce byla použita verze HTML 5.

1.2 CSS (Cascading Style Sheets)

Kaskádové styly popisují vzhled jednotlivých HTML elementů. Odděluje charakteristiku stylu od struktury HTML.

1.3 Bootstrap

Front-end framework pro úpravu typografie, formulářů, tlačítek, navigace a dalších HTML elementů.

1.4 JavaScript

Objektově orientovaný programovací jazyk využíván převážně pro tvorbu webových aplikací. Obvykle jsou jím ovládány různé prvky webové stránky. To znamená, že se program napsaný v jazyce JavaScript spouští až po stažení webové stránky z internetu (tzv. na straně klienta). Interpretem je tedy v tomto případě přímo webový prohlížeč. V této práci není JavaScript využíván ve své standardní podobě, ale více je využívána jeho knihovna jQuery.

1.5 jQuery

JavaScript knihovna obsahující funkce pro výběr a změnu DOM (Document Object Model) elementů, manipulaci s CSS, možnou obsluhu událostí HTML elementů. Odděluje chování stránky od HTML struktury.

1.6 AJAX (Asynchronous JavaScript and XML)

Umožňuje asynchronní komunikaci se serverem. Není proto vždy nutné obnovovat stránku. V práci je využit výhradně ke zpracování akcí vyvolaných z formulářů.

1.7 Twig

Šablonovací systém pro PHP umožňující oddělení aplikační vrstvy od vrstvy prezenční.

1.8 PHP

Skriptovací jazyk sloužící výhradně k vytváření dynamických webových stránek. Jeho kód je vykonáván pouze na serveru a vytváří HTML, které následně zasílá klientské aplikaci.

1.9 Composer

Umožňuje automatickou instalaci a aktualizaci závislostí pro PHP.

2 Adresářová struktura

2.1 Kořenový adresář

Obsahuje pouze soubory: index.php, .htaccess a application.php.

2.2 Adresář controller

Obsahuje všechny PHP soubory, které se starají o funkčnost jednotlivých webových stránek aplikace (controllery).

2.3 Adresář css

Obsahuje soubor s kaskádovými styly.

2.4 Adresář db

Uchovává soubory:

- `db_info.php` - Obsahuje informace o databázi. Tyto informace jsou nutné pro umožnění programového přístupu do databáze.
- `semestralka.sql` - Sql skript vytváří a plní daty databází pro webovou aplikaci.

2.5 Adresář dependencies

Obsahuje všechny závislosti PHP aplikace (Twig, Bootstrap, jQuery, PDO).

2.6 Adresář doc

Obsahuje pouze pdf dokumentaci.

2.7 Adresář js

Obsahuje JavaScript soubory, které obsluhují akce jednotlivých formulářů aplikace.

2.8 Adresář model

Obsahuje soubor, který pracuje s databází aplikace.

2.9 Adresář uploads

Slouží pouze pro uložení souborů, které byly nahrány na server uživateli.

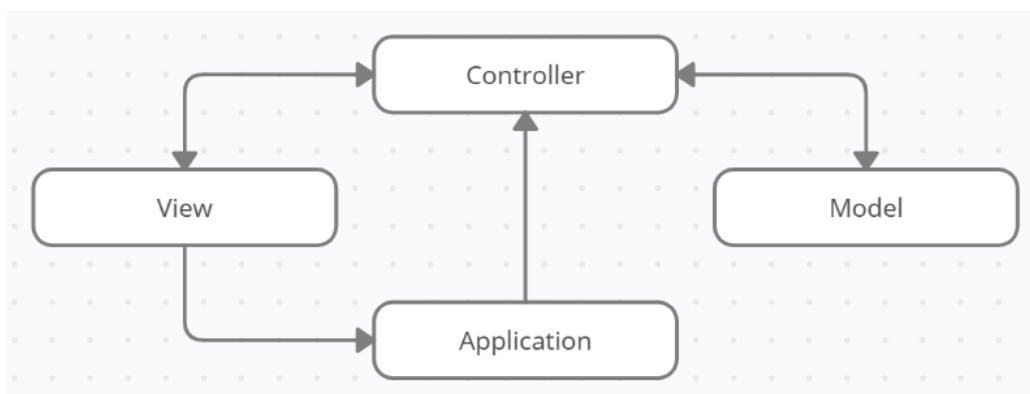
2.10 Adresář view

Má pouze dva podadresáře:

- `html` - statické HTML stránky
- `templates` - `.twig` šablony

3 Architektura aplikace

Jelikož byla jednou z podmínek provedení semestrální práce použít architekturu MVC, snažil jsem se tak učinit.



Obrázek 1: Provedení MVC architektury

3.1 Model

Zajišťuje manipulaci s daty. Pomocí PDO (PHP Data Objects) a prepared statements čte a zapisuje do databáze.

3.2 View

Výstup na obrazovku uživatele. Jedná se buď to o statické HTML stránky nebo o šablony, které používají controllery pro zobrazování dat.

3.3 Application

Zpracovává požadavky od uživatele. Po zpracování nové URL vytvoří příslušný Controller a zavolá v něm metodu, která je třeba pro to, aby poskytla výstup požadovaný uživatelem.

3.4 Controller

Každá stránka aplikace má svůj vlastní PHP Controller. Pokud daná strana obsahuje nějaký formulář, či na ni lze vykonat nějakou akci, která vyžaduje odeslání žádosti na server, pak má takováto strana i vlastní JavaScript soubor, ve kterém se vyvolané akce zpracovávají.

Controllery pak také zajišťují zobrazení správných informací uživatelům nebo předávají uživateli zadaná data pomocí komunikace s Modelem.

4 Závěr

Semestrální práce by měla splňovat podmínky zadání. Avšak může obsahovat mírné nedostatky, které mohou pramenit z nedostatku zkušeností z vytváření webových aplikací.

Práce je zveřejněna na mém GitHub repozitáři, na adrese: <https://github.com/kohoutja/kiv-web-semestralka>.