

Komunikace s klienty

KIV/WEB – Semestrální práce

student:	<i>Milan Růžička</i>
studijní číslo:	<i>A13B0412P</i>
email:	<i>ruzicka6@students.zcu.cz</i>
datum:	<i>14. 12. 2014</i>
GitHub:	<i>ruzickamilan/WEB</i>

Obsah

ZADÁNÍ	3
POUŽITÉ TECHNOLOGIE	3
ADRESÁŘOVÁ STRUKTURA	4
ERA DIAGRAM	5
POPIS ARCHITEKTURY APLIKACE	6
POZNÁMKY K PRÁCI S APLIKACÍ	7
ZÁVĚR	8

Zadání

Webové stránky pro finančního poradce s možností komunikace s klienty

Samotné stránky budou sloužit jako portfolio firmy zabývající se finančním poradenstvím. Databáze bude sloužit pro komunikaci mezi finančními poradci a klienty pomocí webového rozhraní. Každý klient se může zaregistrovat. Každý i neregistrovaný klient může položit otázku v diskuzi, registrovaní můžou odpovídat a navíc psát i soukromé zprávy poradcům. Poradci (admin) mohou posílat hromadné zprávy registrovaným klientům (user), zároveň mají práva pro správu diskuze (mazání příspěvků, banování nevhodně se vyjadřujících klientů ap.).

Použité technologie

Krom standartní podoby jazyků, které se používají při tvorbě webových stránek (HTML 5, CSS 3, PHP, SQL, JavaScript), jsem využil navíc různé šikovné knihovny těchto jazyků. Ty nejznámější z nich jsem vyjmenoval níže.

Bootstrap

Bootstrap je velká knihovna kaskádových stylů díky kterým jsem svou stránku nastyloval během chvíle. Výhoda je, že je webová stránka rovnou optimalizovaná i pro mobilní zařízení s malým displejem. Další co bych vyzvedl jsou ikonky, které knihovna obsahuje. Není nic jednoduššího než vybrat na stránkách bootstrapu potřebnou ikonu a pomocí tagu span jí vložit na stránky, ikona se velikostně i barevně přizpůsobí okolí a ve finále vypadá stránka dá se říct profesionálně.

Twig

Twig slouží jako šablonovací systém. Díky němu jsem oddělil PHP od HTML. Na začátku jsem měl problémy se vyrovnat s tím, že nemohu psát do HTML souborů php příkazy, nicméně nakonec jsem se naučil jak dělat pomocí twigu v HTML kódu podmínky, proměnné i cykly a nakonec to hodnotím jako užitečnou věc.

JQuery

JQuery je obrovská JavaScriptová knihovna, kde lze využívat mnoho funkcí, které poskytuje. Osobně jsem jí použil pro plynulý posun po stránce k dané komponentě, pro validaci formulářů a nebo pro horní panel, kde se mění fotky.

Adresářová struktura

- **application**
 - **core**
 - **controllers**
 - obsah
 - **models**
 - **views**
 - pages
 - sablona.php
 - **packages**
 - bootstrap
 - simptip
 - twig
 - validationengine
 - mistral.eot
 - mistral.woff
- **config**
 - config.inc.php
 - functions.inc.php
- **css**
 - default.css
 - ie.css
- **images**
- db1 (tabulky + data).sql
- dokumentace.pdf
- index.php
- README.md

soubory aplikace

- jádro aplikace
 - **veškeré kontrolery (PHP)**
 - kontrolery pro statický obsah
 - **veškeré modely (DB)**
 - **veškeré pohledy (HTML)**
 - veškeré stránky HTML
 - hlavní šablona webu
- **balíčky**
 - bootstrap (css a jquery)
 - simptip (našeptávač obsahu)
 - twig (šablonovací systém)
 - validation engine (kontrola formulářů)
 - písmo Mistral pro IE
 - písmo Mistral pro ostatní prohlížeče

konfigurační část

- hlavní konfigurační soubor
- přídavné funkce aplikace

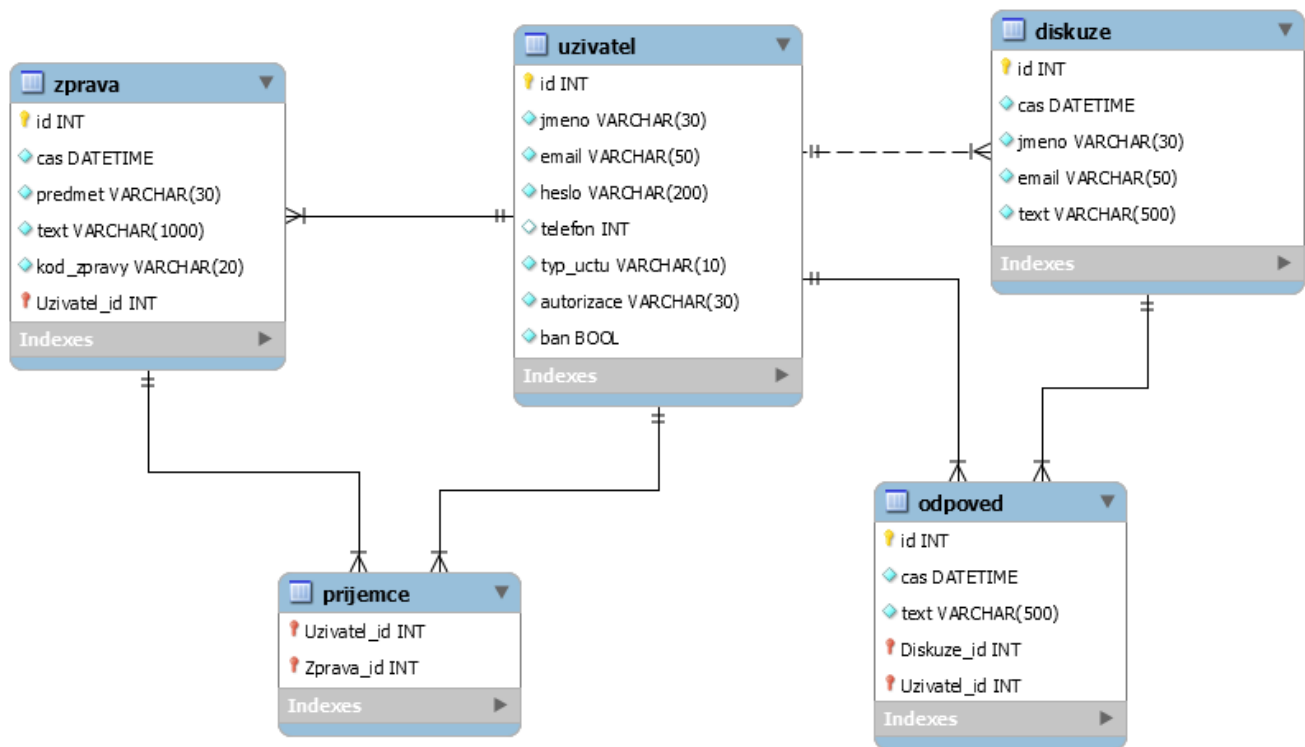
kaskádové styly

- hlavní vlastní styl
- styl pro Internet Explorer

složka s grafikou

soubor s databázovými tabulkami a daty
tato dokumentace
hlavní soubor, uvede stránky do chodu
soubor s adresářovou strukturou

ERA diagram



Popis architektury aplikace

Aplikaci jsem dle zadání vyvíjel na základě MVC (model-view-controller) architektury. Na následujících řádcích popíšu nejdůležitější soubory, které celou funkčnost aplikace umožňují.

index.php

index.php je základní soubor, který si nejprve zavolá nastavení webu ze souboru config.inc.php, poté pomocí třídy App.class zpracuje URL adresu (pomocí instance třídy Smerovac.class), kterou jako cestu ke kontroleru předá třídě Kontroler.class a ta ve finále zavolá správný pohled, který vidí uživatel. V tomto souboru se také samozřejmě musí vytvořit instance šablonovacího systému, který plní daty šablonu (sablonu.php).

App.class.php

Tato třída spolu s předešlým souborem index.php uvede aplikaci "k životu". V mém případě, když zpětně nahlížím do zdrojových souborů, bych mohl po přepsání pár řádků v souboru index.php třídu zcela vypustit. Její funkci u mě nahradí hlavní kontroler a směrovač.

config.inc.php

V tomto souboru jsou umístěné konfigurační záležitosti celé aplikace. Nastartují se Sessions, nastaví se cesty k souborům, nastaví se parametry pro přihlášení do databáze, nadefinují se databázové tabulky, načte se šablonovací systém, ad.

Kontroler.class.php

Tato třída je mateřská třída ke všem kontrolerům. Jsou zde nadefinované základní proměnné v šabloně, které potřebujeme u každého jiného kontroleru (tittle, status přihlášení, info proužek, ap.). Stejně tak má nadefinovanou funkci zpracuj, která se využívá (není přepsaná funkcí z jiného kontroleru, který jí dědí) hlavně u stránek, kde je pouze statický text a neprovádí se zde žádné přídatné PHP skripty.

Smerovac.class.php

V této třídě se dějí ty zázraky kdy z obyčejné adresy zjistím kde je umístěn kontroler k dané stránce a její pohled. Když je adresa správná uživatel je přesměrován, když ne vyskočí stránka, která informuje o neexistující stránce.

Db.class.php

Třída určená čistě pro databázi. Funkce jsou z většiny dynamické a pracují už v přímém spojení s databází a s jejími daty.

Session.class.php

Tato třída je určená pouze pro práci se Sessions. Zde jsou funkce, které mi umožňují zapamatovat si přihlášeného uživatele a následně s jeho údaji pracovat.

Kontrolery

Veškeré kontrolery jsou umístěné ve složce controllers. V podsložce obsah jsou kontrolery pro statický obsah. Každý kontroler dědí od hlavního kontroleru (Kontroler.class.php) funkci zpracuj a zpracovává PHP obsah pro danou stránku na webu.

Modely

Veškeré modely jsou umístěné ve složce models. Každý model dědí funkce pro práci s databází od hlavní třídy (Db.class.php). V každém modelu jsou jen funkce, které mají něco společného s databází a nějak s ní hospodaří (pomocí třídy Db.class.php).

Pohledy

Veškeré pohledy jsou umístěné ve složce views. V této složce se nachází i hlavní šablona webu (sablon.php). V podsložce pages jsou pak samotné pohledy, v podsložce pages/obsah jsou opět pohledy pro statický obsah. Při vkládání statického obsahu mám tak dobrý přehled o tom, kde mám v adresáři jaký soubor (neotevírám systémové soubory).

Poznámky k práci s aplikací

Aplikaci zprovozníte tím, že veškerá data nahrajete na webový server, do databáze nahrajete zkušební data s tím, že v souboru config.inc.php upravíte přihlašovací údaje do databáze, název databáze a případně podsložku ve které máte data na serveru umístěné (já je měl např. v podsložce localhost/**WEB**).

Když se vše rozběhne, můžete se zkusit zaregistrovat s tím, že autorizační email na Vaší emailovou adresu nedorazí, jelikož je funkce pro odeslání zatím zakomentovaná z důvodu nefunkčnosti na mém lokálním webovém serveru. Účet tedy autorizujete pomocí odkazu “?page=registrace&autorizace=kod_autorizace”, kde kod_autorizace vyčtete z tabulky uzivatel v databázi. Funkce mail je zakomentovaná i v případě zasílání nového hesla, nebo nového autorizačního odkazu. Na práva admina může povyšovat ostatní uživatele pouze admin, tak pro vyzkoušení této funkce se v databázi musíte povýšit manuálně.

Ostatní vlastnosti se shodují se zadáním. Přidal jsem akorát funkci pro upravování dotazů v diskuzi (pouze autor) a admin může posílat hromadné soukromé zprávy i všem adminům.

Závěr

V době vytváření dokumentace lze konstatovat, že aplikace funguje jak má a měla by být dostatečně odlazená. Nicméně nedostatky zde určitě objevíme. Jedná jak o nedostatky grafické, kdy např. levé menu zcela nezapadá do celkového grafického kontextu, tak i o nedostatky programové, kdy mohly být různé situace lépe řešené. Narážím například na příspěvky v diskuzi, kde se vždy zobrazí všechny příspěvky. Mohl jsem zde umístit funkci, která by příspěvky např. po pěti stránkovala a uživatel mezi stránkami procházel pomocí číselné osy. Dále jsem také uvažoval o menším Ajax skriptu, kdy by po příchodu nové soukromé zprávy vyskočila na uživatele nějaká notifikace. Chtěl jsem také zprovoznit soubor .htaccess pro “hezke” URL, ale můj lokální server je špatně nastaven a nedokázal jsem v přípustné době modul zprovoznit. Čas jsem raději investoval do důležitějších věcí a dovedl aplikaci do podoby v jaké je.