

Dokumentace k semestrální práci “Zpěvník”

Tento dokument obsahuje dokumentaci k webovým stránkám “Zpěvník”, které jsou semestrálním projektem Adama Škardy do předmětu “Základy webových aplikací”.

Dokumentace obsahuje popis úlohy, tak jak je zadán na stránkách “Courseware” předmětu a akceptační podmínky, tak jak jsou zadány v tabulce na zmíněné stránce. Dále se zde nachází uživatelská příručka, ve které je vysvětleno ovládání stránky a okomentované snímky obrazovky. Nakonec se zde nachází popis implementace a popis uložení dat, ve kterém je vysvětleno fungování stránky.

Obsah

1 Popis úlohy.....	2
1.1 Povinné položky hodnocení.....	3
1.2 Nepovinné položky hodnocení.....	4
2 Uživatelská příručka.....	5
2.1 Úvodní stránka – index.....	5
2.2 Registrace – registrování nového uživatele.....	5
2.3 Uživatelé – seznam registrovaných uživatelů.....	6
2.4 Zpěvník – seznam písní.....	6
2.5 Zpěvník – přidání písně.....	7
2.5.1 Jak psát text s akordy.....	7
3 Popis implementace.....	8
3.1 Tokeny.....	8
3.2 Přihlášení.....	8
3.3 Registrace.....	8
3.4 Pagination.....	9
3.5 LiveSearch.....	9
4 Popis uložení dat.....	10

1 Popis úlohy

Řešením mé semestrální úlohy na téma „Zpěvník“ je stránka, která ukazuje text a akordy písní přidanych registrovanými uživateli stránky.

Zásadní akceptační podmínky jmenovitě:

- Samostatná práce (nevím jak prokázat)
- Serverová strana v PHP
- Dokumentace kódu
- Fungování celé práce

byly splněny. Dále se v následujících dvou podkapitolách nachází tabulky se zadáním a řešením jednotlivých úloh. Řešení není detailně uvedeno, ale je řečeno, kde se řešení použilo a hlavní metoda řešení. Celé řešení použité je dokumentováno v kapitole 3 Popis implementace.

1.1 Povinné položky hodnocení

Úloha	Řešení
Js kontrola formuláře onsubmit	Ano - V souboru „add.php“ ve formuláři „song_submit“ je
Fungování na Firefox a Chrome	Ano - Chrome - Verze 87.0.4280.141 (Oficiální sestavení) (64bitový) Firefox – Verze 75.0 (64bitový) (Firefox pro Fedoru)
Stránka i bez Js	Ano - Nebude fungovat LiveSearch na stránce „Zpěvník“ a js validace přidanych písní
Validní html bez chyb	Ano – validní všechny stránky 8.1.2021
CSS – formátování bez použití html elementů	Ano – žádné formátovací html tagy se v projektu nenacházejí
CSS – formátování mimo tagy	Ano – formátování v odděleném souboru mainStyle.css
Odolnot proti XSS	Ano – při výpisu dat vložených uživatelem se používá htmlentities nebo htmlspecialchars s ENT_QUOTES
Neztrácí se uživatelem vyplněné hodnoty	Ano – funkce fill_form v connect.php vypisuje data z Post při neúspěšném odeslání formuláře
Přihlašování uživatelů	Ano – UI v site_nav.php a logika v login.php
Hesla nejsou v plaintextu	Ano – jsou uložena pomocí password_hash funkce – je uložen hash
Zpracování chyb	Ano – kontrola při připojení k databázi a query, při selhání die()
Stránkované seznamy položek	Ano – zpěvník a users „pagination“
Popis zadání	Ano – tato kapitola
Dokumentace ve zdrojovém kódu	Ano – v css /**/, js //, php //, html<!-->

1.2 Nepovinné položky hodnocení

Úloha	Řešení
Js validace na všech elementech	Ano – Js funkce validate() kontroluje všechny tři položky formuláře
Validace pomocí HTML5	Ano – např. v přihlašovacím formuláři je required i pattern
Html validní bez warningů	Ano – validní 8.1.2021
Přístupné formuláře – použití label	Ano – na všech elementech všech formulářů (kromě [type=submit])
HTML 5 – tagy formulářové prvky	Ano – Header, Footer, Article, OptGroup
Styl pomocí <link>	Ano – linknutý css mainStyle.css
Skinovatelnost	Ne
Styl pro tisk	Ano – v mainStyle.css pomocí @media print{}
Zajímavost v Css	Ne
Odolnost proti dvojímu odeslání	Ano – pomocí tokenů uložených v \$_SESSION[„tokens“] a hidden inputu ve formuláři
Správné vyhodnocení a zobrazení chyb	Ano – např. validate_reg() v reg.php – kontrola při registraci
Jasně rozlišení povinných a nepovinných údajů	Ano – pomocí required a placeholder – všechny text inputy jsou nutné
Hesla jsou osolená	Ano – rovnou dvakrát pomocí password_hash a vlastní \$salt
Stránkování s filtry	Ano – users.php – pořadí uživatelů a filtrování těch co přidali píseň

2 Uživatelská příručka

Tato kapitola se bude věnovat používání stránky uživatelem a vysvětlením uživatelského rozhraní.

2.1 Úvodní stránka – index

Úvodní stránka projektu obsahuje krátký článek o stránce a hlavní navigaci projektu. V navigaci také najdeme přihlašovací formulář a odkaz na registrační formulář.



2.2 Registrace – registrování nového uživatele

Aby se uživatel přihlásil musí se nejprve zaregistrovat. Pro registraci stačí vyplnit jméno a heslo ve specifikovaném formátu.

The screenshot shows the 'Registrace' page. It has a title 'Registrace' and a sub-header 'Údaje nutné pro registraci'. Below this are input fields for 'Jméno:' and 'Heslo:', and a 'Zaregistrovat' button. To the right of the 'Heslo:' field is a red arrow pointing to the 'Seznam požadavků pro heslo' (Password requirements list). Below the input fields is a list of requirements for the password, each preceded by a red 'x' icon: '6-20 Znaků', 'Velké písmeno', 'Malé písmeno', and 'Číslo'. At the bottom of the page is a red arrow pointing to the 'Odkaz zpět na index' (Link back to index) and a link 'Zpět na hlavní stránku'.

2.3 Uživatelé – seznam registrovaných uživatelů

V projektu se také nachází seznam všech registrovaných uživatelů a počet písní které přidali. Lze v ní listovat, řadit ji podle jména, či počtu přidanych písní a vyfiltrovat uživatele, kteří nepřidali žádnou píseň.

2.4 Zpěvník – seznam písní

Nejdůležitější stránka celého projektu se nachází pod odkazem „Zpěvník“. Je zde abecedně seřazený seznam všech uložených písní (100 na stránku). Píseň lze také vyhledat pomocí pole „Vyhledávání“. Nachází se tu taky odkaz na stránku pro přidání písně.

[illegible]

2.5 Zpěvník – přidání písně

Přihlášený uživatel může do databáze přidat píseň. Musí ovšem vyplnit všechna pole formuláře.

The screenshot shows a web form titled "Přidat píseň". At the top, there is a navigation bar with links: "Zpěvník", "Uživatelé", "Úvod", and "O autorovi". To the right of the navigation bar are input fields for "Jméno:" and "Heslo:", followed by buttons "Přihlásit" and "Registrace". The main form area has a light orange background. It contains three input fields: "Jméno písničky:" (with a red arrow pointing to it from the label "Jméno písničky"), "Interpret písničky:" (with a red arrow pointing to it from the label "Od koho je písnička"), and "Text písničky:" (with a red arrow pointing to it from the label "Text písničky"). The "Text písničky:" field is a larger text area. Below the text area is a button labeled "Přidat písničku".

2.5.1 Jak psát text s akordy

Uživatel by měl také myslet na to, jak píseň přidává. To znamená, že by si měl dávat pozor na správné rozložení textu vzhledem k akordům, správnou, jednotvárnou notaci akordů a množství slov na řádek.

The image compares two ways to write song lyrics with chords. The top example is labeled "NE" (No) in large red letters. It shows a form with "Jméno písničky:" filled with "Cukrářská Bossanova" and "Interpret písničky:" filled with "Jaromír nohavica". The "Text písničky:" field contains the following text:
Cmaj7 C#dim dMI7 g7
1. Můj přítel snídá sedm kremrolí Cmaj7C#dima když je spořádá, dá si repete,Dmi7
G7cukrlátko,
The bottom example is labeled "ANO" (Yes) in large red letters. It shows a form with "Jméno písničky:" filled with "Nutné" and "Interpret písničky:" filled with "Nutné". The "Text písničky:" field contains the following text:
Cmaj7 C#dim Dmi7 G7
1. Můj přítel snídá sedm kremrolí
Cmaj7 C#dim
a když je spořádá, dá si repete,
Dmi7 G7
cukrlátko,
The "ANO" example shows a more structured and readable layout of the lyrics and chords.

3 Popis implementace

V této kapitole je popis implementace hlavních funkcí projektu.

3.1 Tokeny

V tomto projektu se nachází právě jeden druh tokenů, a to tokeny k zamezení dvojitého odeslání formuláře. Obecně tento systém funguje tak, že se v některých důležitých formulářích nachází položka [input=hidden], která si pamatuje náhodně vygenerovaný token (heš pseudo náhodného čísla). Tuto hodnotu si také pamatuje proměnná `$_SESSION[„tokens“][]`. Když se hodnoty shodují, není problém může se vykonat logika formuláře normálně. Pokud se tokeny neshodují byl proveden re-submit, v tomto případě ignorujeme tento resubmit a případně pošleme uživatele na jinou stránku pomocí `header()`. Poslední možnost nastane, při první návštěvě stránky, jedná se případ, kdy ještě token neexistuje, v tomto případě token vygenerujeme.

3.2 Přihlášení

Klientská strana přihlášení je v souboru `site_nav.php`, je to formulář, který posílá informace přes Post na sám sebe, protože na každé stránce, kde je používána `site_nav` je includovaný soubor `login.php`, kde probíhá veškerá logika přihlašování. Tato logika zkontroluje výše zmíněný token a jestli uživatel už náhodou není přihlášený. Pokud je vše v pořádku zeptá se `login.php` databáze `users` pomocí Mysql query. Zde začne hledat uživatelské jméno, pokud nebude jméno nalezeno, přihlášení se nezdařilo, pokud je jméno nalezeno zkontrolujeme heslo pomocí funkce `password_verify()`. Pokud heslo odpovídá bude uživatel přihlášen pomocí proměnné `$_SESSION[„username“]` (funkce vyžadující přihlášení kontrolují `isset()` této proměnné).

3.3 Registrace

Veškerá registrační logika se schovává v souboru `reg.php`, kde je také formulář k registraci. Stěžejní funkcí tohoto souboru je funkce `validate_reg()`, která na vstupu přijímá tři argumenty - `$name` – uživatelské jméno k registraci, `$password` – heslo registrovaného uživatele, `$conn` – spojení mezi php a mysql databází s tabulkou uživatelů. Funkce zkontroluje uživatelské jméno a heslo (např. délku hesla, unikátnost jména) a vrátí chybové zprávy jako array stringů. Pokud je array prázdná, je jméno i heslo validní a může se pokračovat v registraci, kde je dalším a posledním krokem vložení informací o registrovaném uživateli do databáze. Při tomto kroku se také hešuje heslo pomocí funkce `password_hash()` tzn., že se ukládá tento heš, ne samotné heslo. Nakonec je po úspěšném INSERT query user odkázán pomocí funkce `header()` na stránky `thankyou.php`

3.4 Pagination

Stránkování funguje za pomoci `$_GET`, kam se ukládá údaj, o tom, na které straně (page) klient je. Podle toho, na které straně client je a kolik údajů chceme na stranu zobrazit si vyžádáme od databáze řádky (za pomoci `LIMIT` offset, počet řádků) s požadovanými údaji. Tyto údaje je možné filtrovat pomocí `WHERE` a `ORDER`, které si user nastaví pomocí formuláře a jsou ukládané v `SESSION`, aby přežili reload stránky.

3.5 LiveSearch

Live search se nachází na stránce `songs.php` využívá technologii `AJAX` k zobrazování výsledků hledání v průběhu psaní. Při každé události `keyup()` v poli `search` se pomocí javascript funkce `show_result(str: string)` odešle dotaz `xmlhttprequest` s hledaným výrazem na soubor `livesearch.php`, který se zeptá na výraz databáze písní pomocí `query` (příkaz `LIKE`) a vrátí odpověď od databáze zabalenou do tagu „`<a>`“. Pokud dostane zápornou odpověď od databáze (počet řádků odpovědi je 0), pak `livesearch.php` vrátí „`<p>Žádné výsledky</p>`“. Tyto odpovědi jsou vzápětí vepsány do `<div>` pod `searchbarem`.

4 Popis uložště dat

Data pro webové stránky jsou uloženy ve třech nezávislých tabulkách MySql databáze “skardada” na wa.toad. V tabulce “semestrálka_stránky” se nachází obsah stránek (úvod a o autoru -index.php). V tabulce “users” se nacházejí registrovaní uživatelé, tato tabulka se tedy používá pro registraci a přihlašování. Nakonec v tabulce “songs” se nachází seznam písní, které jsou ve zpěvníku. Veškerá komunikace s databází je zřízená pomocí MySql queries. Spojení je navázáno v souboru connect.php.