



Semestrální práce z předmětu
Webové aplikace

E-shop s hrami

6. ledna 2025

Autor:

Martin Reich
A22B0123P
reichm@students.zcu.cz

Cvičící:

Ing. Michal Nykl Ph.D.
nyklm@ntis.zcu.cz

Obsah

1	Zadání	2
1.1	Nutné požadavky na všechny samostatné práce	2
1.2	Dokumentace	2
2	Popis použitých technologií	3
2.1	JavaScript	3
2.2	PHP	5
2.3	CSS a Bootstrap	5
2.4	MVC architektura	5
3	Adresářová struktura aplikace	5
4	Popis architektury aplikace	6
5	Seznam defaultních uživatelů	7
6	Závěr	7

1 Zadání

1.1 Nutné požadavky na všechny samostatné práce

1. Technologie - povinně HTML5, CSS, PHP a SQL (MySQL nebo jiná databáze), volitelně šablony, JavaScript, AJAX, Bootstrap apod.
2. Aplikace musí dodržovat MVC architekturu a využívat OOP (min. controllery a model).
3. Web má jeden vstupní soubor (obvykle index.php), který na základě parametrů URL adresy provede požadovanou akci (tj. zavolá příslušný controller) a vypíše výstup uživateli.
4. Pro práci s databází musí být využito PDO nebo jeho ekvivalent.
5. Web musí být chráněn proti útokům typu XSS a SQL Injection.
6. V databázi musí být hesla hashována.
7. Web musí využívat upload souborů.
8. Web musí mít responzivní design (alespoň pro PC a mobil).
9. Web musí mít alespoň 3 uživatelské role (po přihlášení v systému provádí příslušné činnosti, např. autor, recenzent, admin).
10. K aplikaci musí být dodána dokumentace (viz dále) a skripty pro instalaci databáze (např. získané exportem databáze).
11. Práce musí být osobně předvedena cvičícímu a po schválení odevzdána na CourseWare či Portál.
12. Aplikaci není možné realizovat s využitím ucelených PHP frameworků (zakázáno např. Nette, Symfony atd.). Použití jejich komponent je možné pouze po schválení vyučujícím.
13. Pro front-end je vhodné využít framework Bootstrap (getbootstrap.com) nebo jeho ekvivalent.

1.2 Dokumentace

1. Vaše jméno, email, datum vytvoření, název předmětu a název aplikace/tématu.
2. URL vytvořených stránek (pokud jsou zveřejněny na serveru students.kiv.zcu.cz či jinde)
3. Popis použitých technologií - uveďte hlavně, ve které části jste kterou technologii použili.

4. Popis adresářové struktury aplikace - co je ve kterých adresářích a souborech.
5. Popis architektury aplikace - co mají na starosti které třídy (popř. lze využít i UML diagramy).
6. Seznam defaultních uživatelů včetně loginů a hesel.
7. U alternativního zadání uveďte celé, cvičicím schválené zadání práce.
8. Dokumentaci netiskněte, ale odevzdejte ji ve formátu PDF spolu s aplikací.

2 Popis použitých technologií

2.1 JavaScript

JavaScript je vysokoúrovňový, dynamický a interpretovaný programovací jazyk, který se používá především pro vývoj webových aplikací a interaktivních webových stránek. Původně byl vyvinut pro přidání dynamických funkcí do webových stránek, ale dnes je využíván ve velkém množství různých oblastí, od front-endu až po back-end. JavaScript je jazyk, který je schopen provádět různé úkoly na straně uživatele (client-side), jako jsou například:

- Manipulace s obsahem stránky (DOM - Document Object Model)
- Ověřování formulářů
- Tvorba interaktivních prvků (např. tlačítka, animace, interaktivní mapy)
- Komunikace se serverem bez nutnosti obnovovat stránku (pomocí AJAX, Fetch API)

JavaScript je základní součástí většiny moderních webových aplikací. Používá se pro vytvoření interaktivních prvků, animací a dynamických změn na stránkách. Příklady využití:

- **Webové aplikace:** JavaScript umožňuje vytvářet interaktivní webové aplikace, které reagují na akce uživatele v reálném čase.
- **Webové rozhraní:** JavaScript může manipulovat s HTML a CSS pro změnu vzhledu a chování webové stránky bez nutnosti jejího znovu načítání.

JavaScript jsem použil pro vytvoření interaktivní webové aplikace, která umožňuje uživatelům zadávat data do formuláře, provádět validace a dynamicky měnit obsah stránky bez jejího opětovného načítání. Tento přístup mi umožnil zvýšit uživatelskou přívětivost a efektivitu aplikace.

```

    document.addEventListener('DOMContentLoaded', () => {
'use strict';

const errors = {
  username: true,
  email: true,
  password: true,
  passwordConfirm: true,
  surname: true,
  address: true,
  login: true,
  tel: true,
};

const setError = (input, message) => {
  const errorSpan = input.parentElement.querySelector('.error');
  errorSpan.textContent = message;
  errorSpan.style.display = 'block';
  input.parentElement.classList.add('hasError');
};

const clearError = (input) => {
  const errorSpan = input.parentElement.querySelector('.error');
  errorSpan.textContent = '';
  errorSpan.style.display = 'none';
  input.parentElement.classList.remove('hasError');
};

const validateInput = (input, formType) => {
  const value = input.value.trim();

  if (input.classList.contains('name') && formType === 'signup') {
    errors.username = value.length === 0;
    errors.username ? setError(input, 'Prosím zadejte svoje jméno') : clearError(input)
  }

  if (input.classList.contains('surname') && formType === 'signup') {
    errors.surname = value.length === 0;
    errors.surname ? setError(input, 'Prosím zadejte svoje příjmení') : clearError(input)
  }
  ...
}
}

```

2.2 PHP

PHP je skriptovací jazyk, který se vykonává na serveru, což znamená, že kód napsaný v PHP je zpracován na serveru před tím, než je odeslán uživateli jako HTML stránka. PHP je široce používaný k vývoji webových aplikací a podporuje různé databáze, zejména MySQL. PHP je jednoduchý na učení a používání a je často integrován s HTML, CSS a JavaScriptem.

V mém případě jsem PHP implementoval v controllerech jako například pro přidávání produktů do systému, který umožňuje uživatelům přidávat nové produkty do databáze. Tento controller zpracovává formulář, ověřuje údaje, přidává produkt do databáze a zobrazuje zpětnou vazbu uživateli.

2.3 CSS a Bootstrap

CSS (Cascading Style Sheets) je jazyk, který slouží k definování vzhledu a formátování webových stránek. Pomocí CSS můžeme určovat, jak budou jednotlivé HTML prvky zobrazeny, jako například barvy, písma, mezery, okraje, pozadí a rozložení. CSS je nezbytnou součástí moderního webového vývoje, protože umožňuje oddělit obsah (HTML) od vzhledu (CSS), což přispívá k lepší údržbě a přehlednosti kódu. Bootstrap je open-source framework pro vývoj webových stránek a aplikací, který obsahuje sadu nástrojů pro rychlou tvorbu responzivních a mobilně přívětivých designů. Využívá HTML, CSS a JavaScript a poskytuje předdefinované styly a komponenty, které usnadňují vývoj.

V mé semestrální bylo CSS a Bootstrap využito k rychlejšímu vytvoření vzhledu stránky a jeho responzivity.

2.4 MVC architektura

MVC (Model-View-Controller) je designový vzor (nebo architektura) používaný při vývoji aplikací, který dělí aplikaci na tři základní komponenty: Model, View (pohled), a Controller (ovladač). Tento vzor pomáhá udržet kód čistý, dobře strukturovaný a snadno rozšiřitelný, což usnadňuje údržbu aplikace.

V mé semestrální práci byla MVC struktura použita pro zlepšení struktury kódu, zjednodušení údržby a ke snazší rozšiřitelnosti.

3 Adresářová struktura aplikace

Aplikace je rozdělena do několika hlavních složek podle architektury MVC (Model-View-Controller), která zajišťuje lepší organizaci a údržbu kódu. Níže je popsána struktura aplikace:

- **app**: Hlavní složka aplikace, která obsahuje všechny komponenty potřebné pro správný chod aplikace.

- **controller**: Složka, která obsahuje všechny kontrolery. Kontrolery jsou zodpovědné za obsluhu uživatelských akcí, komunikaci s modelem a aktualizaci pohledu. Každý kontroler je obvykle zaměřen na určitou část aplikace (např. přihlašování, správa produktů).
 - * **controllery**: Konkrétní soubory obsahující třídy kontrolerů, například `users_controller.php`, `login_controller.php`.
- **model**: Složka obsahující modely, které představují data aplikace a manipulují s nimi. Modely se starají o logiku aplikace a komunikaci s databází.
 - * **modely**: Součástí této složky jsou soubory, které definují třídy pro práci s daty, například `product.php`, `user.php`.
- **styles**: Složka pro uložení stylů aplikace. Tato složka obsahuje soubory CSS nebo preprocesorové soubory jako SASS, které definují vzhled aplikace.
 - * **styli**: Soubory, které obsahují stylování aplikace, např. `cart.css`, `forms.css`.
- **views**: Složka pro zobrazení (pohledy). Pohledy jsou soubory, které generují HTML, které se zobrazuje uživateli. Tyto soubory zobrazují data, která poskytl kontroler, a zajišťují interakci s uživatelem.
 - * **vzhledy**: Konkrétní soubory obsahující HTML a šablony pro generování pohledů, například `login.php`, `users.php`.
- **db**: Tato složka obsahuje vše, co souvisí s databází, jako jsou skripty pro migrace, schémata databáze, inicializace databáze a podobně.
- **docs**: Složka, která obsahuje dokumentaci k aplikaci. Může zahrnovat technické detaily, jak aplikaci používat nebo rozšiřovat, a popis architektury aplikace.

4 Popis architektury aplikace

`account_controller.php`

- Zajišťuje úpravu uživatelských údajů.

`login_controller.php`

- Zajišťuje přihlašování uživatelů.

`database.php`

- Zajišťuje práci s databází.
- Ukládá a odebírá věci z databáze.

`user_manager.php`

- Zajišťuje přihlašování a odhlašování uživatele.

users.php

- Zajišťuje vykreslování stránky se správou uživatelů.

settings.inc.php

- Nastavení všech důležitých proměnných jako jsou stránky, cesty k controllerům a dále.

5 Seznam defaultních uživatelů

Superadmin

1. Login: superadmin
2. Heslo: superadmin

Admin

1. Login: admin
2. Heslo: admin123

Skladník

1. Login: seller
2. Heslo: seller123

Zákazník

1. Login: customer
2. Heslo: customer123

6 Závěr

Závěr této semestrální práce se zaměřil na vývoj webové aplikace pro e-shop s hrami, který splňuje všechny zadané požadavky a využívá moderní webové technologie. Během vývoje byly implementovány klíčové funkce jako ověřování uživatelských vstupů, dynamická manipulace s obsahem pomocí JavaScriptu, správa produktů v administrátorském rozhraní a zabezpečení aplikace proti útokům typu SQL Injection a XSS.

Použité technologie, jako jsou PHP, MySQL, CSS a Bootstrap, umožnily vytvoření responsivního a uživatelsky přívětivého rozhraní, které je plně funkční na různých zařízeních. Důraz byl kladen na správnou architekturu aplikace, která byla postavena na vzoru MVC, čímž bylo zajištěno jasné oddělení jednotlivých částí aplikace (modelu, pohledů a kontrolerů) pro lepší údržbu a rozšiřitelnost.

Dokumentace a popis použitých technologií, struktury aplikace i implementace jednotlivých částí byly podrobně zaznamenány a odevzdány spolu s aplikací. Výsledný e-shop s hrami je plně funkční a umožňuje správu uživatelů, produktů a objednávek s různými úrovněmi přístupových práv.

Celkově lze říci, že semestrální práce úspěšně splnila všechny technické požadavky a poskytuje funkční a bezpečnou webovou aplikaci, která může být základem pro další rozšíření a vylepšení v budoucnosti.