NotScam.com – výuková webová aplikace ve formě parodického podvodného internetového obchodu s nereálním zbožím.

0. Zadání úlohy

1. Produktová dokumentace

1.1 Účel aplikace

1.2 Uživatelské role

1.3 Přehled funkcí

1.4 Popis uživatelského rozhraní

2. Programátorská dokumentace

2.1 Použité technologie

2.2 Architektura aplikace

2.3 Struktura složek

2.4 Tok dat

2.5 Bezpečnost (XSS, SQLi, hesla, session)

2.6 Validace (klient / server)

2.7 PRG pattern

0. Zadání úlohy Zadáním semestrální práce bylo vytvořit dynamickou webovou aplikaci v technologii PHP bez použití externích frameworků a knihoven třetích stran.

Aplikace měla splňovat následující požadavky:

- registrační a přihlašovací formulář s validací na straně klienta i serveru
- bezpečné ukládání hesel
- implementace uživatelských rolí (nepřihlášený uživatel, přihlášený uživatel, administrátor)
- možnost administrátorské správy obsahu a uživatelských rolí
- ochrana proti bezpečnostním zranitelnostem (XSS, SQL Injection)
- validní HTML a oddělení stylů a skriptů do samostatných souborů
- automaticky generovaná dokumentace zdrojového kódu

Na základě tohoto zadání byla navržena a implementována aplikace NotScam.com.

1. Produktová dokumentace

1.1 Účel aplikace

Cíl:Vytvořit výukovou webovou aplikaci ve formě parodického podvodného internetového obchodu, ve kterém si uživatelé mohou prohlížet a “kupovat” podvodné produkty, jako například instrukce „Jak zbohatnout za 24 hodin”, kámen “Přitáhni si štěstí” nebo kniha „Jak přesvědčit vesmír, aby ti pomohl“ a podobné. Projekt kombinuje humoristický nápad s realizací klasického internetového obchodu a slouží k procvičení principů tvorby dynamických webových stránek, práce s formuláři, validací a správou uživatelských rolí.

1.2 Uživatelské role: 1. Nepřihlášený uživatel • Může prohlížet katalog produktů a detailní karty zboží • Může přejít na registraci nebo přihlášení 2. Přihlášený uživatel • Může prohlížet katalog produtků a detailní karty zboží • Může přidávat produkty do virtuálního košíku 3. Admin • Přidává a upravuje produkty v katalogu • Může produkty mazat

1.3 Přehled funkcí

Aplikace NotScam.com poskytuje následující funkce:

- Zobrazení katalogu produktů s podporou stránkování a řazení
- Zobrazení detailu produktu včetně popisu, ceny a obrázku
- Registrace nového uživatele s validační kontrolou zadaných údajů
- Přihlášení a odhlášení uživatele
- Správa uživatelského profilu (úprava jména, e-mailu a hesla)
- Možnost smazání uživatelského účtu
- Přidávání produktů do virtuálního košíku
- Správa produktů administrátorem (vytváření, úprava, mazání)
- Správa uživatelských rolí administrátorem

1.4 Popis uživatelského rozhraní

Uživatelské rozhraní aplikace je navrženo přehledně a intuitivně tak, aby se v něm uživatel snadno orientoval bez nutnosti dalšího vysvětlení.

Hlavní navigace je umístěna v horní části stránky a její obsah se mění v závislosti na roli přihlášeného uživatele. Nepřihlášeným uživatelům jsou nabízeny odkazy na přihlášení a registraci, zatímco

přihlášený uživatel má přístup ke svému profilu a košíku. Administrátor má k dispozici odkaz na admin panel.

Katalog produktů je zobrazen ve formě přehledného seznamu s možností stránkování. Každý produkt obsahuje název, cenu a obrázek, přičemž kliknutím na produkt se zobrazí jeho detailní stránka.

Formuláře v aplikaci jsou vybaveny validačními hláškami, které se zobrazují po odeslání formuláře a informují uživatele o chybně vyplněných polích a důvodech neplatnosti zadaných údajů.

Rozhraní je přizpůsobeno i pro tisk pomocí samostatných tiskových stylů.

2. Programátorská dokumentace

2.1 Použité technologie

Aplikace je vytvořena jako klasická serverová webová aplikace bez použití externích frameworků a knihoven třetích stran, v souladu s požadavky předmětu Základy webových aplikací.

Serverová strana:

PHP 8.1.2 Použito jako hlavní serverový programovací jazyk pro zpracování požadavků, práci s formuláři, validaci vstupních dat, správu uživatelských relací a komunikaci s databází.

MySQL 8.0.44 Relační databáze sloužící k ukládání uživatelských účtů a produktů.

PDO (PHP Data Objects) Použito pro bezpečnou komunikaci s databází pomocí připravených dotazů (prepared statements), čímž je zajištěna ochrana proti SQL Injection útokům.

Klientská strana:

HTML5 Slouží k definici struktury jednotlivých stránek aplikace. Veškeré stránky jsou navrženy tak, aby byly validní dle standardů HTML5.

CSS3 Použito pro stylování uživatelského rozhraní. Styly jsou odděleny do samostatných souborů, včetně speciálního stylopisu pro tisk pomocí pravidel @media print.

JavaScript (ES6) Použit pro klientskou validaci formulářů, dynamické ovládání uživatelského rozhraní a asynchronní komunikaci se serverem pomocí technologie AJAX (fetch API).

2.2 Architektura aplikace

Aplikace je navržena jako klasická serverová webová aplikace s využitím architektonického vzoru MVC (Model–View–Controller), přizpůsobeného jednoduchému PHP projektu bez použití frameworků.

Logika aplikace je rozdělena do několika vrstev:

– Controller (řídicí logika) – Model (práce s databází) – View (vykreslení HTML) – Pomocné a servisní skripty

Controller vrstva zpracovává HTTP požadavky (GET/POST), načítá vstupní data z formulářů, provádí základní kontrolu přístupových práv a volá příslušné funkce modelu.

Model vrstva obsahuje funkce pro práci s databází (pomocí PDO a připravených SQL dotazů), například načítání, ukládání, úpravu a mazání dat. Model neobsahuje žádnou prezentační logiku.

View vrstva je tvořena PHP soubory s HTML šablonami, které vykreslují data získaná z controlleru. Při výpisu uživatelských dat je používáno zabezpečení proti XSS útokům pomocí funkce htmlspecialchars.

Součástí architektury jsou také pomocné skripty, například pro správu session, validaci vstupních dat, práci s formuláři a generování navigace.

Aplikace používá vzor Post-Redirect-Get (PRG), aby se zabránilo opakovanému odeslání formuláře při obnovení stránky.

2.3 Struktura složek

Zdrojový kód aplikace je rozdělen do přehledné struktury složek, která odpovídá použití architektury MVC a usnadňuje orientaci v projektu i jeho další rozšiřování.

Hlavní struktura projektu:

/admin Obsahuje administrační rozhraní aplikace. Zahrnuje správu produktů (vytváření, úprava, mazání) a správu uživatelských rolí. Jednotlivé akce jsou realizovány pomocí samostatných endpointů pro práci s produkty a rolemi.

/includes Obsahuje aplikační logiku a pomocné moduly. Tato složka je dále členěna podle funkčnosti:

- /products
  Modely, controllery a pomocné funkce pro práci s produkty.
- /register
  Logika registrace uživatelů včetně serverové validace a AJAX kontroly existence e-mailu.
- /login
  Logika přihlášení uživatele.
- /roles
  Modely a controllery pro správu uživatelských rolí.
- /validation
  Serverová validační logika použitá napříč aplikací.
- /UI
  Pomocné funkce pro vykreslování formulářů, navigace, chybových hlášek a stránkování.
- session_manager.php
  Centrální správa session a regenerace session ID.

- dbh.php
  Inicializace databázového připojení pomocí PDO.

/assets Obsahuje statické soubory aplikace:

- /css – styly aplikace včetně tiskových stylů
- /js – JavaScriptové soubory (UI logika, validace, AJAX)
- /img – grafické prvky aplikace

/uploads Slouží k ukládání nahraných souborů. V aplikaci jsou zde ukládány obrázky produktů. Velké binární soubory nejsou ukládány do databáze.

/docs Obsahuje automaticky generovanou dokumentaci vytvořenou nástrojem PHPDocumentor na základě dokumentačních komentářů ve zdrojovém kódu.

2.4 Tok dat

Tok dat v aplikaci odpovídá architektuře MVC.

Po odeslání HTTP požadavku uživatelem je požadavek zpracován controllerem, který provede validaci vstupních dat a zavolá příslušný model. Model komunikuje s databází, provede potřebné operace a vrátí data zpět controlleru.

Controller následně rozhodne o dalším kroku – buď provede přesměrování (PRG pattern), nebo předá data do view, které vygeneruje výsledný HTML výstup pro uživatele.

2.5 Bezpečnost (XSS, SQLi, hesla, session) Aplikace byla navržena s důrazem na základní bezpečnostní principy webových aplikací. Níže jsou popsána hlavní opatření použitá k ochraně dat a uživatelů.

Ochrana proti XSS (Cross-Site Scripting): Veškerá data zadaná uživatelem jsou ukládána do databáze v původní podobě, bez úprav nebo escapování. K ochraně proti XSS útokům dochází až při výpisu dat do HTML.

Při každém výpisu uživatelských dat je použita funkce:

```
htmlspecialchars($value, ENT_QUOTES, 'UTF-8')
```

Tím je zajištěno, že případné HTML nebo JavaScript vložené uživatelem nebude interpretováno prohlížečem jako kód, ale zobrazí se pouze jako text.

Ochrana proti SQL Injection:

Veškerá komunikace s databází probíhá pomocí prepared statements (připravených dotazů) přes rozhraní PDO.

Všechny SQL dotazy používají parametrizované vstupy, například:

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE email = :email");
$stmt->execute([':email' => $email]);
```

Díky tomu nejsou uživatelské vstupy nikdy přímo vkládány do SQL dotazů a aplikace není zranitelná vůči SQL Injection útokům.

Ukládání a ověřování hesel:

Hesla uživatelů nejsou nikdy ukládána v plaintextu.

Při registraci je heslo: zahashováno pomocí funkce

```
password_hash()
```

s algoritmem BCRYPT s nastavenou hodnotou cost

Při přihlášení je heslo ověřováno pomocí:

```
password_verify($password, $hash)
```

Tento přístup zajišťuje bezpečné uložení hesel i v případě kompromitace databáze.

Správa session a ochrana přihlášení: Aplikace využívá PHP sessions pro správu přihlášení uživatelů.

Bezpečnost session je zajištěna následujícími opatřeními:

- povolen pouze režim cookies (session.use_only_cookies)
- zapnutý strict mode (session.use_strict_mode)
- cookies jsou nastaveny jako httponly
- ID session je pravidelně regenerováno
- ID session je regenerováno také po úspěšném přihlášení

Tím je sníženo riziko útoků typu session fixation nebo session hijacking.

Role a přístupová práva:

Aplikace pracuje s uživatelskými rolemi:

- nepřihlášený uživatel
- přihlášený uživatel
- administrátor

Citlivé operace (správa produktů, změna rolí) jsou dostupné pouze administrátorům a jsou kontrolovány na straně serveru, nikoliv pouze pomocí uživatelského rozhraní.

2.6 Validace (klient / server) Aplikace využívá dvouvrstvou validaci vstupních dat – na straně klienta (JavaScript, HTML) i na straně serveru (PHP).

Klientská validace je realizována pomocí JavaScriptu a základních HTML atributů (required, type, step).

Jejím cílem je:

- okamžitě upozornit uživatele na chybně vyplněná pole,
- zabránit zbytečnému odesílání formuláře na server,
- zlepšit uživatelskou přívětivost aplikace.

Validace probíhá při odeslání formuláře, přičemž:

- chybná pole jsou vizuálně zvýrazněna,
- u každého pole je zobrazena konkrétní validační hláška,
- formulář není odeslán, pokud obsahuje chyby.

Validační logika je centralizována v souboru: assets/js/validation/validators.js

Serverová validace je implementována v jazyce PHP a je povinná pro všechny formuláře. Každé vstupní pole je kontrolováno bez ohledu na to, zda proběhla klientská validace.

Serverová validace zahrnuje:

- kontrolu prázdných hodnot,
- kontrolu délky vstupních dat,
- kontrolu formátu,
- kontrolu obchodních pravidel (např. unikátnost e-mailu),
- kontrolu nahrávaných souborů (typ, velikost).

Validační logika je centralizována v souboru: includes/validation/validators.php

2.7 PRG pattern (Post-Redirect-Get) Aplikace využívá návrhový vzor PRG (Post-Redirect-Get) pro zpracování formulářů. Cílem tohoto vzoru je zabránit opakovanému odeslání dat při obnovení stránky a zajistit správný tok HTTP požadavků.

PRG pattern rozděluje práci s formulářem do tří kroků:

1. POST: Uživatel odešle formulář (např. registrace, přihlášení).
2. Redirect: Server zpracuje data a následně provede přesměrování pomocí HTTP hlavičky Location.
3. GET: Prohlížeč načte cílovou stránku pomocí GET požadavku.

Díky tomuto postupu se zabrání tomu, aby se při obnovení stránky znovu odeslala původní POST data.