

Střední průmyslová škola Třebíč

Maturitní práce

Webová aplikace pro truhlářství

Profilová část maturitní zkoušky

Studijní obor: Informační technologie

Třída: ITA4

Školní rok: 2023/2024 Vojtěch Mastný

Zadání práce

ABSTRAKT

Tvorba maturitní práce je jedním z velmi klíčových momentů při studiu. Kvalita zpracování její formální části je pak jedním z nejdůležitějších kritérií při jejím hodnocení. Cíl této práce je popsat jednotlivé kroky během tohoto procesu, doporučit postupy a vytvořit šablonu, která usnadní celý proces.

KLÍČOVÁ SLOVA

maturitní práce, šablona

ABSTRACT

The creation of a graduation thesis is one of the most crucial moments during studies. The quality of the processing of its formal part is then one of the most important criteria in its evaluation. The aim of this work is to describe the individual steps during this process, recommend procedures and create a template that will facilitate the entire process.

KEYWORDS

graduation thesis, template

PODĚKOVÁNÍ

Děkuji Mgr. Petru Novotnému za cenné připomínky a rady, které mi poskytl při vypracování maturitní práce.

V Třebíči dne 25. ledna 2024 podpis autora

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl v ní všechny prameny, literaturu a ostatní zdroje, které jsem použil.

V Třebíči dne 25. ledna 2024

podpis autora

Obsah

[Úvod 6](#_Toc155282813)

[1 Teoretická část 7](#_Toc155282814)

[1.1 Použité technologie 7](#_Toc155282815)

[1.1.1 HTML 7](#_Toc155282816)

[1.1.2 CSS 8](#_Toc155282817)

[1.1.3 PHP 9](#_Toc155282818)

[1.2 PhpMyAdmin 10](#_Toc155282819)

[1.3 MySQL 11](#_Toc155282820)

[1 dsdsa část 12](#_Toc155282821)

[1.1 Styly 12](#_Toc155282822)

[1.1.1 Řádkování 12](#_Toc155282823)

[1.1.2 Zvýrazňování textu 12](#_Toc155282824)

[1.1.3 Členění textu 12](#_Toc155282825)

[1.1.4 Číslování stran 12](#_Toc155282826)

[1.2 Vytvoření obsahu 13](#_Toc155282827)

[1.3 Psaní úvodu 13](#_Toc155282828)

[1.4 Struktura odstavců 14](#_Toc155282829)

[1.5 Obrázky, tabulky a rovnice 14](#_Toc155282830)

[1.6 Řazení a struktura kapitol 16](#_Toc155282831)

[1.7 Závěr 16](#_Toc155282832)

[1.8 Seznam použitých zdrojů 16](#_Toc155282833)

[1.9 Seznam použitých symbolů a zkratek 16](#_Toc155282834)

[1.10 Seznamy použitých obrázků a tabulek 17](#_Toc155282835)

[1.11 Seznam příloh 17](#_Toc155282836)

[Závěr 18](#_Toc155282837)

[Seznam použitých zdrojů 19](#_Toc155282838)

[Seznam použitých symbolů a zkratek 20](#_Toc155282839)

[Seznam obrázků 21](#_Toc155282840)

[Seznam tabulek 22](#_Toc155282841)

[Seznam příloh 23](#_Toc155282842)

Úvod

Tato maturitní práce se zabývá vývojem webové aplikace na míru pro truhlářství, která byla zadána jako zakázkový projekt. Volba tohoto tématu vychází z mého hlubokého zájmu o technologie a jejich praktické využití, a to i přes počáteční nedostatek zkušeností s vývojem webových stránek. Webová aplikace, kterou vytvořím, bude navržena tak, aby vyhovovala specifickým potřebám zákazníka a jejím cílem je zefektivnit komunikaci s klienty a prezentaci truhlářských výrobků.

V následujících kapitolách budu systematicky popisovat jednotlivé fáze vývoje, technologie použité při tvorbě aplikace a věnovat se analýze potřeb truhlářské firmy a  způsobu, jakým byla aplikace přizpůsobena těmto požadavkům.

Webová aplikace vytváří komplexní prostředí prostřednictvím různých jazyků a technologií. Základní stavební kameny této aplikace budou spočívat v použití jazyku HTML, který slouží k definici struktury celého webu. Důraz bude kladen na vytvoření kvalitního uživatelského rozhraní a vizuálního designu, a to díky jazyku CSS, který umožňuje elegantní stylování obsahu.

Přihlašovací proces, klíčový pro interakci uživatelů s aplikací, bude řešen s využitím jazyka PHP. Tímto způsobem se zajistí bezpečné a spolehlivé zpracování přihlašovacích údajů uživatelů. Tato webová aplikace bude založena na solidním technologickém základu a umožňuje uživatelům efektivní a intuitivní interakci s obsahem díky kombinaci těchto jazyků a moderních technologií.

# Teoretická část

V této části se zaměřím na popis jednotlivých technických a teoretických prostředků využívaných v mé práci.

## Použité technologie

### HTML

HyperText Markup Language (HTML), představuje jazyk označování, který slouží   
k vytváření dokumentů obsahujících hypertextové odkazy a pokročilé formátování. Funkce HTML jsou rozmanité a umožňují manipulaci s obsahem webových stránek. [1]

Základními stavebními kameny HTML jsou tagy. Tyto tagy rozdělujeme na párové  
a nepárové. Párové tagy se využívají např. pro nadpisy, odkazy, odstavce apod.

*<h1>Lorem ipsum dolor</h1>*

<h1> až <h6>: Tagy pro nadpisy, kde `<h1>` je nejvyšší úrovně a `<h6>` je nejnižší.

Zato nepárové používáme např. pro vložení obrázků, nebo zalomení řádku.

*<img src="pictures/spst.png" width="450" height="320" alt="SPŠT logo">*

*<img>*:vkládá obrázek do stránky.

*`src="pictures/spst.png"`*:Cesta k obrázku.

*`width="450"`*:Šířka obrázku (450 pixelů).

*`height="320"`*: Výška obrázku (320 pixelů).

*`alt="SPŠT logo"*`: Alternativní text pro případ, že obrázek není načten (SPŠT logo).

Struktura základní HTML stránky zahrnuje několik klíčových částí:

<!DOCTYPE html>: Tato deklarace definuje verzi HTML, kterou dokument používá.

<html>: Tento kořenový element obaluje celý obsah HTML dokumentu.

<head>: Sekce `<head>` obsahuje metadata o dokumentu, jako například kódování znaků, náhledové informace a odkazy na externí styly.

<meta charset="UTF-8">: Určuje použité kódování (UTF-8) pro zpracování znaků.

<meta name="viewport" content="width=device-width, initial-scale=1.0">: Nastavuje zobrazení na zařízeních s různými šířkami obrazovky.

<title>: Určuje název stránky, který se zobrazuje v záhlaví nebo na kartě prohlížeče.

<body>: Tato část obsahuje samotný obsah dokumentu, jako text, obrázky, odkazy, atd.

<p>: Tag pro odstavce textu.

Vytváření formulářů je další klíčovou funkcí HTML, umožňující uživatelům např. odeslat e-maily, reagovat na dotazníky nebo provádět objednávky z online katalogu. HTML rovněž umožňuje vytvářet tabulky, které poskytují větší kontrolu   
nad formátováním dokumentů a uspořádáním obsahu. Definování barev pozadí   
pro dokument, tabulku, řádek nebo buňku přispívá k celkové vizuální atraktivitě.

Odkazy, další klíčový prvek HTML, umožňují navigaci mezi sekcemi dokumentu, odkazy na jiné stránky a integrovaní audio a video souborů. [1]

HTML pouze slouží jako základní struktura webové stránky, pro vizuální styl se dále využívá jazyk CSS, viz. Kapitola 1.1.2

### CSS

Cascading Style Sheets (CSS) představuje jazyk, který stanovuje základní vizuální podobu webových stránek a uspořádání jejich prvků. Jeho použití spočívá ve spolupráci s HTML, XHTML nebo XML, kde detailně popisuje, jak by měly být prezentovány jednotlivé složky. Komponenty, jež nejsou specifikovány pomocí CSS, zůstávají v původní podobě.

Struktura webového prostoru, respektive dokumentu, vytvořená pomocí HTML, zůstává oddělena od jeho vizuální podoby. Tímto způsobem vzniká možnost pro jednodušší a sofistikovanější úpravy, což přispívá k rychlejšímu zobrazení obsahu stránek. [2]

CSS můžeme propojit s HTML souborem 3 způsoby.

Zápisem do hlavičky HTML souboru.

<style>

p { color: green; }

</style>

Zápisem přímo do konkrétního prvku

<p style="color: green;"> Lorem ipsum dolor </p>

‘style="color: green;"‘: Změnění barvy písma na zelené.

Propojení s externím CSS souborem.

<head>

<link rel="stylesheet" type="text/css" href="style.css" />

</head>

### PHP

Hypertext Preprocessor (PHP) je serverový skriptovací jazyk, který byl vytvořen   
s ohledem na specifické potřeby webových stránek. To znamená, že veškeré operace prováděné pomocí PHP nejsou realizovány na straně klienta, jako je tomu např.   
u JavaScriptu, ale jsou interpretovány na straně serveru a generují výstup ve formě HTML (nebo jiného formátu), který je pak prezentován uživateli.

Významným rysem PHP je jeho Open Source povaha, což znamená, že jde   
o technologii, která je volně dostupná a šířitelná. PHP není vázané na konkrétní platformu a nemá pevnou vazbu s žádným určitým serverem. To znamená, že je schopno fungovat na různých prostředích bez ohledu na konkrétní specifikace. Tato vlastnost poskytuje flexibilitu a umožňuje nasazení PHP na různých serverech   
a prostředích podle potřeb daného projektu. [3]

PHP má 3 rozdílné typy proměnných podle rozsahu:

Globální: je deklarována mimo tělo funkcí nebo bloků kódu a je přístupná ze všech částí programu.

$globalniPromenna = 10;

function pristupKGlobalniPromenne() {

global $globalniPromenna;

echo $globalniPromenna;

}

Lokální: je deklarována uvnitř těla funkce nebo bloku kódu a je přístupná pouze   
z tohoto konkrétního kontextu.

function lokalniPromenna() {

$lokalniPromenna = 5;

echo $lokalniPromenna;

}

Statická: uchovává svou hodnotu i po opuštění kontextu, kde byla vytvořena. Při opětovném volání funkce si pamatuje svůj stav.

function statickaPromenna() {

static $pocitadlo = 0;

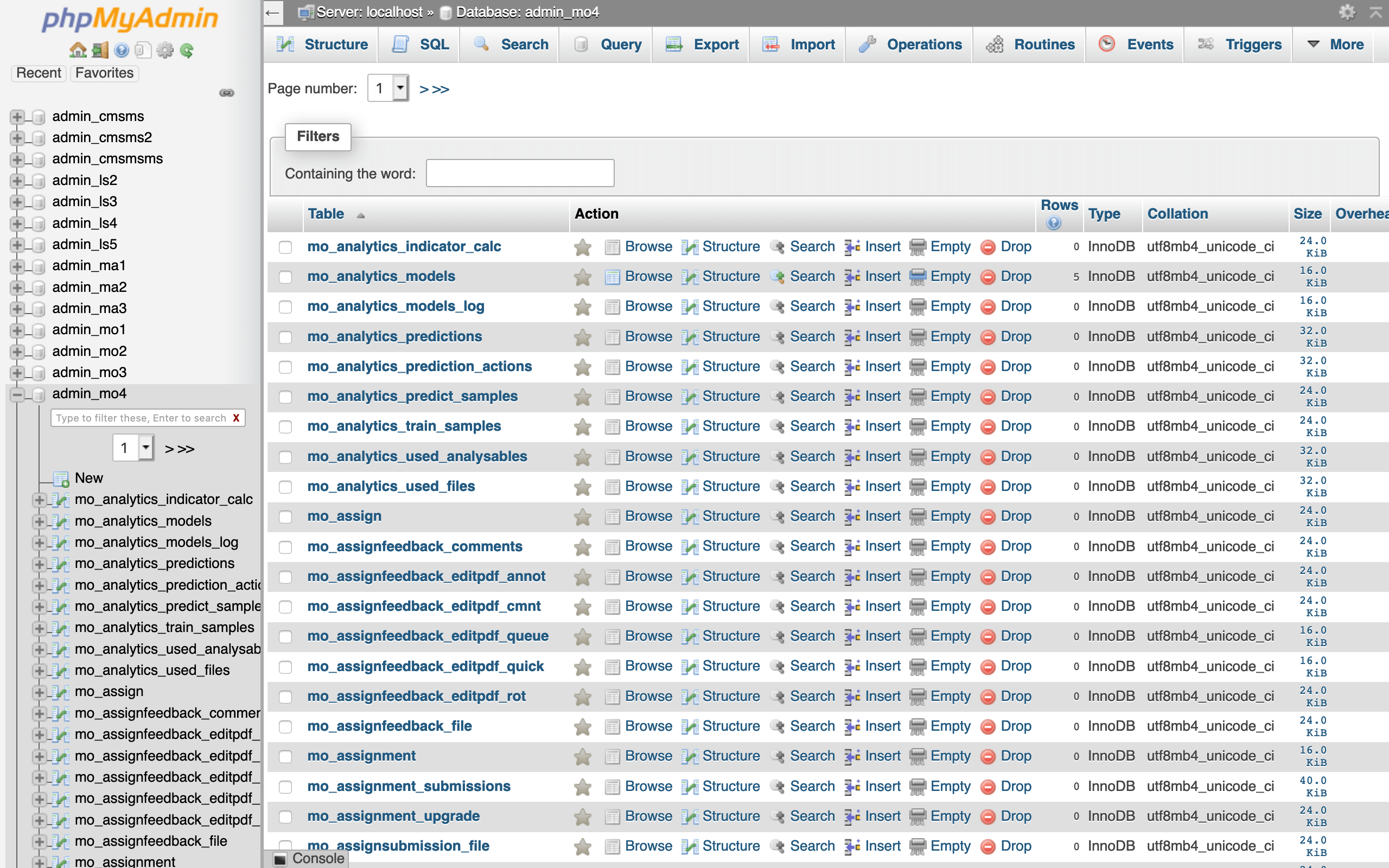
$pocitadlo++;

echo $pocitadlo;

}

### PhpMyAdmin

phpMyAdmin je bezplatný software napsaný v PHP, určený k správě MySQL pomocí grafické uživatelské rozhraní (GUI). phpMyAdmin podporuje širokou škálu operací   
v MySQL a MariaDB. Operace, které se běžně využívají, jsou klíčovými kroky v rámci správy databází, tabulek, sloupců, vztahů, indexů, uživatelů a oprávnění v prostředí MySQL a MariaDB. Tyto rutinní úkony nabízejí uživatelům možnost efektivně ovládat a optimalizovat strukturu svých databázových systémů. Uživatelské rozhraní usnadňuje provádění těchto častých operací bez nutnosti hlubší znalosti SQL syntaxe. Díky těmto činnostem mohou správci a vývojáři plynule manipulovat s daty, zajišťovat konzistentnost a optimalizovat výkon svých databázových prostředí. [4]



Obr. 1 phpMyAdmin prostředí [6]

### MySQL

The Structured Query Language (SQL) představuje relační databázový nástroj, umožňující vytváření, odstraňování, načítání a manipulaci s databázemi. MySQL funguje jako Relational Database Management System (RDBMS) a zaujímá klíčové místo v provozu moderních webových stránek a internetových služeb. Díky své efektivitě je MySQL preferovaným řešením pro plynulé ukládání a získávání rozsáhlých souborů dat. Jedním z příkladů informací, které by mohly být uloženy   
v MySQL databázi, jsou údaje jako uživatelské jméno registrovaného uživatele stránky, bezpečně šifrované heslo pro zvýšení bezpečnosti, datum registrace uživatele atd. [5]

## Hrozby

Je zde mnoho hrozeb, které mohou negativně ovlivnit webové stránky. Proto v této kapitole některé představím a jak se proti nim bránit.

### SQL Injection

SQL Injection (SQLI), je útočný vektor, který využívá škodlivý SQL kód pro manipulaci s backendovou databází za účelem přístupu k informacím, které nebyly určeny k zobrazení. Tyto informace mohou zahrnovat různé položky, včetně citlivých firemních údajů, seznamů uživatelů nebo soukromých informací o zákaznících.

Dopad, který SQLi může mít, je široce rozsáhlý. Úspěšný útok může vést   
k neautorizovanému zobrazení seznamů uživatelů, smazání celých tabulek  
a v některých případech k získání administrátorských práv k databázi. [7]

Příklad SQLI:

*uName = getRequestString("username");*

*uPass = getRequestString("userpassword");*

*sql = 'SELECT \* FROM Users WHERE Name ="' + uName + '" AND Pass ="' + uPass + '"'* [8]

*uName = getRequestString("username")*: Získává uživatelské jméno z vstupního požadavku.

*uPass = getRequestString("userpassword")*: Získává heslo uživatele z vstupního požadavku.

*sql = 'SELECT \* FROM Users WHERE Name ="' + uName + '" AND Pass ="' + uPass + '"'*: Sestavuje SQL dotaz pro vyhledání záznamu v databázi, kde uživatelské jméno a heslo odpovídají získaným hodnotám.

SQL injekce obvykle spadá do tří kategorií:

In-band SQLi

Útočník používá stejný komunikační kanál k spuštění svých útoků a k shromáždění výsledků. Jednoduchost a efektivita In-band SQLi ho činí jedním z nejčastějších typů SQLi útoku. Existují dvě podvarianty této metody:

Chybově založená SQLi: útočník provádí akce, které způsobují, že databáze vytváří chybové zprávy. Útočník může potenciálně využít data poskytnutá těmito chybovými zprávami k získání informací o struktuře databáze.

Unijní SQLi: tato technika využívá UNION SQL operátor, který spojuje více vybraných příkazů generovaných databází do jedné HTTP odpovědi. Tato odpověď může obsahovat data, která mohou být využita útočníkem.

Inferential (Blind) SQLi

Útočník odesílá datové zátěže na server a sleduje odpověď a chování serveru, aby se dozvěděl více o jeho struktuře. Tato metoda se nazývá blind SQLi, protože data nejsou přenášena z databáze webové stránky k útočníkovi, a tak útočník nemůže vidět informace o útoku in-band.

Blind SQL injekce závisí na odpovědi a chovatelských vzorech serveru, takže jsou obvykle pomalejší k provedení, ale mohou být stejně škodlivé. Blind SQL injekce lze klasifikovat následovně:

Booleovské: útočník odesílá SQL dotaz do databáze, který vyzývá aplikaci k vrácení výsledku. Výsledek se bude lišit v závislosti na tom, zda je dotaz pravdivý nebo nepravdivý. Na základě výsledku se informace v HTTP odpovědi změní nebo zůstane nezměněna. Útočník pak může zjistit, zda dotaz vygeneroval pravdivý nebo nepravdivý výsledek.

Časově založené: útočník odesílá SQL dotaz do databáze, který způsobí, že databáze počká před tím, než může reagovat. Útočník může vidět z času, který databáze potřebuje k odpovědi, zda je dotaz pravdivý nebo nepravdivý. Na základě výsledku se HTTP odpověď vygeneruje okamžitě nebo po čekací době. Útočník tak může zjistit, zda zpráva, kterou použil, vrátila pravdivý nebo nepravdivý výsledek, aniž by se spoléhal na data z databáze.

Out-of-band SQLi

Útočník může tuto formu útoku provést pouze tehdy, jsou-li na databázovém serveru používaném webovou aplikací povoleny určité funkce. Tato forma útoku se používá především jako alternativa k in-band a inferential SQLi technikám.

Out-of-band SQLi se provádí, když útočník nemůže použít stejný kanál k spuštění útoku a shromáždění informací, nebo když je server příliš pomalý nebo nestabilní pro tyto akce. Tyto techniky spoléhají na schopnost serveru vytvářet DNS nebo HTTP požadavky k přenosu dat útočníkovi. [7]

Jediným spolehlivým způsobem, jak zabránit útokům SQLi, je ověření vstupů   
a používání parametrizovaných dotazů včetně připravených příkazů. Aplikační kód by nikdy neměl používat vstupy přímo. Vývojář musí zabezpečit veškeré vstupy,  
a to nejen vstupy z webových formulářů, jako jsou přihlašovací formuláře. Je třeba odstranit potenciálně nebezpečné prvky kódu, např. jednoduché uvozovky. Dobrým opatřením je také vypnout viditelnost chyb databáze na produkčních webových stránkách, neboť chyby databáze mohou být využity při SQLi k získání informací  
o databázi. [9]

### Cross-Site Scripting

Cross-Site Scripting (XSS) útoky představují formu injekce, při které jsou do jinak nezávadných a důvěryhodných webových stránek vloženy škodlivé skripty. XSS útoky se vyskytují, když útočník využívá webovou aplikaci k odeslání škodlivého kódu, obvykle ve formě skriptu na straně prohlížeče, jinému koncovému uživateli. Chyby, které umožňují úspěch těchto útoků, jsou poměrně rozšířené a vyskytují se všude tam, kde webová aplikace používá vstup od uživatele ve výstupu, který generuje, aniž by jej ověřovala nebo zakódovávala.

Útočník může pomocí XSS odeslat škodlivý skript nevědomému uživateli. Prohlížeč koncového uživatele nemá způsob, jak by měl vědět, že by skript neměl být důvěryhodný, a provede skript. Protože si myslí, že skript pochází od důvěryhodného zdroje, může škodlivý skript získat přístup k jakýmkoli souborům cookie, relačním tokenům nebo jiným citlivým informacím, které jsou uchovávány prohlížečem   
a používány s daným místem. Tyto skripty dokonce mohou přepsat obsah HTML stránky. [10]

XSS můžeme dělit do 2 kategorií:

Reflektované útoky typu XSS

Reflektované útoky jsou ty, při kterých je vložený skript odražem z webového serveru, například v chybové zprávě, výsledku vyhledávání nebo jakékoliv jiné odpovědi, která zahrnuje část nebo celý vstup odeslaný serveru jako součást požadavku. Reflektované útoky jsou doručeny obětím prostřednictvím jiné trasy, například v e-mailové zprávě nebo na nějakém jiném webovém místě. Když je uživatel podveden, aby klikl na škodlivý odkaz, odeslal speciálně vytvořený formulář nebo dokonce jen procházel na škodlivé místo, vložený kód putuje na zranitelné webové místo, které odrazí útok zpět do prohlížeče uživatele. Prohlížeč pak provede kód, protože pochází z "důvěryhodného" serveru.

Uložené útoky typu XSS

Uložené útoky jsou ty, kde je vložený skript trvale uložen na cílových serverech, například v databázi, v diskuzním fóru, v záznamech návštěvníků, v poli pro komentáře apod. Oběť poté získá škodlivý skript ze serveru, když vyžádá uložené informace. [10]

Ochrana před XSS je v některých případech snadná, avšak může se stát podstatně náročnější v závislosti na komplexitě aplikace a způsobech, jakými zpracovává data ovládaná uživatelem.

V širším kontextu bývá účinná prevence XSS zranitelností obvykle spojena   
s kombinací následujících opatření:

Filtrování vstupů při příjmu. Na místě, kde jsou přijímány uživatelské vstupy, filtrovat co nejstriktněji na základě toho, co se očekává jako platný vstup.

Kódování dat při výstupu. Na místě, kde jsou uživatelem ovládaná data výstupem v HTTP odpovědích, kódovat výstup tak, aby nemohl být interpretován jako aktivní obsah. V závislosti na kontextu výstupu to může vyžadovat kombinaci kódování HTML, URL, JavaScriptu a CSS.

Používání vhodných hlaviček odpovědí. Pro zabránění XSS v HTTP odpovědích, které nejsou určeny obsahovat žádné HTML nebo JavaScript, lze použít hlavičky Content-Type a X-Content-Type-Options, aby prohlížeče interpretovaly odpovědi tak, jak je zamýšleno.

Politika zabezpečení obsahu. Jako poslední linie obrany lze využít Politiku zabezpečení obsahu (CSP) ke snížení závažnosti případných XSS zranitelností, které by se stále mohly vyskytovat. [11]

### Brute Force útok

Brute force útok je často využívaná metoda k prolomení zabezpečení; podle některých odhadů byly brute force útoky příčinou pěti procent potvrzených bezpečnostních průlomů. Tato taktika zahrnuje „uhodnutí“ uživatelských jmen a hesel za účelem získání neoprávněného přístupu k systému. I když je brute force jednoduchou metodou útoku, má vysokou míru úspěšnosti.

Někteří útočníci využívají aplikace a skripty jako nástroje pro brute force. Tyto nástroje zkoušejí mnoho kombinací hesel, aby obešly autentizační procesy. V jiných případech útočníci hledají správné relační ID k získání přístupu k webovým aplikacím. Účel útočníků může zahrnovat krádež informací, infikování webových stránek malwarem nebo narušení poskytovaných služeb.

I když někteří útočníci stále provádějí brute force útoky ručně, v dnešní době jsou téměř všechny útoky automatizovány pomocí botů. Útočníci mají seznamy běžně používaných přihlašovacích údajů nebo skutečných uživatelských údajů získaných z bezpečnostních průlomů nebo temného webu. Boti systematicky útočí na webové stránky, vyzkoušejí tyto seznamy přihlašovacích údajů a informují útočníka, když získají přístup. [12]

Brute force útoky rozdělujeme do několika typů:

Jednoduchý útok hrubou silou: využívá systematický přístup k odhadu, který není závislý na vnější logice.

Hybridní útok: hrubou silou: kombinuje vnější logiku k určení pravděpodobně úspěšných variant hesel a následně pokračuje jednoduchým přístupem k vyzkoušení mnoha možných kombinací.

Slovníkový útok: snaží se uhádnout uživatelská jména nebo hesla pomocí slovníku možných řetězců.

Útoky s využitím rainbow tabule: pracují s předpočítanými tabulkami pro zpětné získání kryptografických hash funkcí.

Útok hrubou silou obrácený: používá běžná hesla proti mnoha možným uživatelským jménům, zatímco plnění přihlašovacích údajů využívá dříve známé kombinace hesel a uživatelských jmen proti různým webovým stránkám. Tato taktika využívá skutečnosti, že mnoho uživatelů používá stejné přihlašovací údaje na různých systémech. [12]

## Styly

Normální velikost 12pt

Nadpis 1 velikost 20pt, tučně

Nadpis 2 velikost 16pt, tučně

Nadpis 3 velikost 14pt, tučně

### Řádkování

Pro vizuální rozdělení nadpisu kapitol a textu se doporučuje před nadpisy kapitol vložit 12 bodů, za nadpisy kapitol 6 bodů, mezi jednotlivé odstavce v textu se vkládá 6 bodů. Mezi odstavce se tedy nevkládá prázdný řádek prostřednictvím ENTERu, ale prostřednictvím odsazení odstavce. Řádkování v základním textu (styl normální) velikost 1,5 bodů.

### Zvýrazňování textu

Důležité myšlenky zvýrazňovat pomocí **tučného** písma nebo *kurzívou* nepoužívat podtržení. Používat jednotnou barvu písma, doporučuje se černá.

### Členění textu

Zarovnání textu do bloku (vpravo i vlevo). Nepoužívat odsazení textu. Číslování kapitol od čísla 1. Za nadpisy kapitol se neuvádí dvojtečka, ani tečka, ani zdroj. Hlavní kapitoly začínají na nové stránce. Ostatní podkapitoly se oddělují od konce předcházející kapitoly mezerou o velikosti 12 bodů.

### Číslování stran

Strany číslovat dole uprostřed. Strany se počítají od titulního listu, uvádějí se však až od vlastního textu (počínaje úvodem) – tedy první uváděné číslo může být např. 6.

## Vytvoření obsahu

Obsah se tvoří automaticky dle užitých stylů. Styly *Nadpis 1*, *Nadpis 2* a *Nadpis 3* slouží k víceúrovňovému vrstvení kapitol. Styl *NadpisBezObs* se v obsahu nezobrazí (viz strana 4, 5 – nadpisy nejsou v obsahu). Z hlediska přehlednosti není doporučeno využívat více než tři úrovně nadpisů. Pro projevení změn je nutné obsah ručně aktualizovat.



Obr. 2.1 Obsah

Text obsahu se píše od jednotné svislice (se zřetelem k nejdelšímu číselnému označení).

## Psaní úvodu

Úvod může být osobitějšího rázu. První odstavec by měl obsahovat motivaci či důvod, který autora přiměl k volbě daného tématu s ohledem na přínos práce. V dalších odstavcích může být popsán obsah jednotlivých kapitol.

## Struktura odstavců

V případě, že věta vychází se spojkou na konci řádku, je spojka vždy přesunuta pomocí Shift + Enter na následující řádek. Hodnoty s jednotkami musí mít mezi číslem a označením jednotky jednu mezeru. To lze uskutečnit pomocí Shift + Ctrl + Mezerník (tzv. pevná mezera).

Text práce by měl být napsán v neutrální formě, tj. ve 3. osobě v trpném rodě. V textu by se neměli objevovat slangové výrazy, citově zabarvená slova ani podmiňovací způsob (s výjimkou je-li, uvažujeme-li apod.). Autor by se měl vyhnout přílišnému opakování slov či užívání nadbytečných výrazů.

Forma textu:

* v neurčité formě (bylo zjištěno, navrhuje se…);
* v 1.  osobě jednotného čísla (zjistil jsem, navrhuji …) – zvláště vhodné v kapitolách, které jsou vlastní prací autora.

## Obrázky, tabulky a rovnice

Popisek obrázku a tabulky se vkládá kliknutím pravého tlačítka myši na objekt a výběrem možnosti *Vložit titulek*. Následně je vybrán typ objektu (*Tab.* nebo *Obr.*) a jeho poloha (obrázky pod objekt, tabulky nad objekt). Styl popisku je Times New Roman 11 kurzíva uprostřed. Seznamy obrázků a tabulek na konci dokumentu jsou automaticky vygenerovány. Obrázky a tabulky mají vždy i slovní popis a rovnice jsou bez slovního popisu. Za titulek obrázku nepatří tečka. [3] Číslování obrázků, tabulek a rovnic je provedeno dle **hlavní kapitoly**, v níž se vyskytují a jejich pořadí v této kapitole. Nástroj *Vložit Titulek* čísluje popisky obrázků a tabulek automaticky. V případě že dojde k editaci, odstranění nebo přesunu již existujícího popisku je nutné dokument aktualizovat. Pro samotné vkládání rovnic je užit nástroj *Rovnice* v záložce *Vložení* (rovnice jsou vždy psány kurzívou).

|  |  |
| --- | --- |
|  |  |

Vzorec pro měření entropie. Rovnice jsou bez slovního popisu.



Obr. 2.2 Příklad umístění legendy obrázku

Titulek ke grafu a obrázku se píše pod objekt.

Tab. 2.1 Legenda k tabulce

|  |  |  |  |
| --- | --- | --- | --- |
| Jméno | Příjmení | Body | Známka |
| Petr | Novák | 25 | 4 |
| Karel | Kolář | 36 | 3 |
| Martin | Pokorný | 89 | 1 |

Titulek tabulky se píše nad tabulku.

Vložený objekt musí být vždy okomentován. Je nutné napsat před a za objekt alespoň jeden odstavec textu.

## Řazení a struktura kapitol

Z hlediska přehlednosti každá nová kapitola (Nadpis 1) začíná na novém listu. První kapitola bývá zaměřena na rešeršní část, tedy definice pojmů, vymezení studované oblasti apod. Druhá a následující kapitoly jsou zaměřeny na řešení samotného problému.

## Závěr

Závěr obsahuje stručné shrnutí získaných poznatků, uvedení dalších možných postupů či řešení, hodnocení dostupné odborné literatury, ze které bylo čerpáno. Závěr by měl obsahovat kritické porovnání záměru práce a dosažených výsledků, srovnání dosažených výsledků s dosud známými poznatky a popis odlišností od doposud známých skutečností. Závěr může naznačit praktické uplatnění výsledků práce.

## Seznam použitých zdrojů

Účelem této části je zaznamenání publikací, jež byly využity v maturitní práci.

Citace se řídí normou ČSN ISO 690 Bibliografické citace. [1] V případě doslovné citace je potřebné citovaný text graficky odlišit od ostatního textu, ohraničit apostrofem, případně ještě kurzívou. Odkaz na citovaný zdroj, jehož plná citace bude uvedena v soupisu literatury na konci práce, se uvede přímo za doslovně citovaný text.

Citování zdrojů se provádí v hranatých závorkách, používá se metoda číselných odkazů. Tento seznam je vždy seřazen dle výskytu citací v textu práce. Necitování použité literatury je považováno za plagiátorství.

## Seznam použitých symbolů a zkratek

Seznam symbolů a zkratech je vytvořen pomocí tabulky. Zkratky a názvy veličin jsou psány stylem *Normální* a jednotky jsou vytvořeny pomocí nástroje *rovnice*. Pro přidání řádku je potřeba kliknout pravým tlačítkem myši do posledního řádku tabulky a z nabídky *vložit* vybrat možnost *vložit pod*. Pro odstranění pak obdobně vybrat možnost *odstranit celý řádek*.

## Seznamy použitých obrázků a tabulek

Tyto seznamy se vytvářejí automaticky dle použitých stylů. Seznamy je potřeba aktualizovat kliknutím pravého tlačítka myši na první položku seznamu a následným výběrem možnosti aktualizovat pole (nebo kliknutím na první položku seznamu a stiskem F9).

## Seznam příloh

Seznam příloh je nutné vyplnit ručně.

Závěr

Vytvořená šablona maturitních prací obsahuje formální požadavky maturitních prací na SPŠT Třebíč. Jedná se zejména o upravené styly v dokumentu, podrobný popis jednotlivých částí maturitní práce a jejího obsahu, snadno editovatelné záhlaví a zápatí s automatickým číslováním stránek a propojení stylů se seznamy a obsahem.

Seznam použitých zdrojů

1. PÍSEK, Slavoj. HTML: začínáme programovat. Online. 4., aktualiz. vyd. Průvodce (Grada). Praha: Grada, 2014. ISBN ﻿ISBN978-80-247-5059-0. Dostupné z: [https://books.google.cz/books?id=tLamAgAAQBAJ&printsec=frontcover&key=AIzaSyDlPfI89JdFhWBVsMVsavVo6aNh057xlTc#v=onepage&q&f=false](https://books.google.cz/books?id=tLamAgAAQBAJ&printsec=frontcover&key=AIzaSyDlPfI89JdFhWBVsMVsavVo6aNh057xlTc%23v=onepage&q&f=false). [cit. 2023-12-29].
2. KOĎOUSKOVÁ, Barbora. KASKÁDOVÉ STYLY PRO ZAČÁTEČNÍKY ANEB JAK ZAČÍT S CSS. Online. Rascasone. 2021. Dostupné z: <https://www.rascasone.com/cs/blog/kaskadove-styly-css-jak-zacit>. [cit. 2023-12-29].
3. Úvod do PHP. Online. Web Tvorba. 2004. Dostupné z: <http://www.webtvorba.cz/php/uvod-do-php.html>. [cit. 2023-12-30].
4. *PhpMyAdmin*. Online. 2003. Dostupné z: [https://www.phpmyadmin.net](https://www.phpmyadmin.net/). [cit. 2023-12-30].
5. YAQUB, Mohhamad. *Data Management: Databases - MySQL Introduction for Researchers*. Online. University of Oxford. 2020. Dostupné z: <https://skills.it.ox.ac.uk/sites/default/files/skills/documents/media/dm006w_mysql_introduction_online_oct20.pdf>. [cit. 2023-12-30].
6. *PhpMyAdmin (1)*. Online. In: Installatron. 2023. Dostupné z: <https://installatron.com/phpmyadmin?locale=sv>. [cit. 2024-01-06].
7. *SQL (Structured query language) Injection*. Online. Imperva. 2023. Dostupné z: <https://www.imperva.com/learn/application-security/sql-injection-sqli/>. [cit. 2024-01-07].
8. *SQL Injection*. Online. W3Schools. 2024. Dostupné z: <https://www.w3schools.com/sql/sql_injection.asp>. [cit. 2024-01-10].
9. *What is SQL Injection (SQLi) and How to Prevent It*. Online. Acunetix. 2019. Dostupné z: <https://www.acunetix.com/websitesecurity/sql-injection/>. [cit. 2024-01-07].
10. *Cross Site Scripting (XSS)*. Online. OWASP. 20244. Dostupné z: <https://owasp.org/www-community/attacks/xss/>. [cit. 2024-01-09].
11. *Cross-site scripting*. Online. PortSwigger. 2024. Dostupné z: <https://portswigger.net/web-security/cross-site-scripting>. [cit. 2024-01-09].
12. *Brute Force Attack*. Online. Imperva. 2023. Dostupné z: [https://www.imperva.com/learn/application-security/brute-force-attack](https://www.imperva.com/learn/application-security/brute-force-attack/)/. [cit. 2024-01-10].

Seznam použitých symbolů a zkratek

HTML HyperText Markup Language

CSS Cascading Style Sheets

PHP Hypertext Preprocessor

SQL The Structured Query Language

RDBMS Relational Database Management Systém

GUI Grafické uživatelské rozhraní

Seznam obrázků

[Obr. 1 phpMyAdmin prostředí 11](#_Toc147493921)

Seznam tabulek

[Tab. 2.1 Legenda k tabulce 12](#_Toc147493615)

Seznam příloh

Prázdná šablona maturitní práce