|  |  |
| --- | --- |
| Obrázok, na ktorom je stôl, znak, kreslenie  Automaticky generovaný popis | SOŠ StrojNÍCKA Partizánska cesta 76  957 01 Bánovce nad Bebravou |
|  |  |

Náučná Kyberbezpečnostná Webstránka

Ročníkový projekt

|  |  |
| --- | --- |
| Bánovce nad Bebravou | Patrik Pirťan |
| 2024 | Ročník štúdia: štvrtý |

|  |  |
| --- | --- |
| Obrázok, na ktorom je stôl, znak, kreslenie  Automaticky generovaný popis | SOŠ StrojNÍCKA Partizánska cesta 76  957 01 Bánovce nad Bebravou |
|  |  |

Náučná kyberbezpečnostná Webstránka

Ročníkový projekt

|  |  |
| --- | --- |
| Riešiteľ: | Patrik Pirťan |
| Študijný odbor: | 3918 M Technické lýceum |
| Katalógové čislo: |  |
| Ročník štúdia: | štvrtý |
| Konzultant: | Ing. Pavel Hazucha |
|  |  |
| Bánovce nad Bebravou | |
| 2024 | |

**Čestné vyhlásenie**

Vyhlasujem, že som ročníkovú prácu vypracoval /a/ samostatne a použil/la som len odbornú literatúru, ktorú uvádzam v zozname použitej literatúry. Moja dokumentácia neobsahuje chránené údaje podniku a/alebo zákazníka a neporušuje autorské práva.

|  |  |
| --- | --- |
| V Bánovciach nad Bebravou, dňa \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Vlastnoručný podpis |

**Poďakovanie**

Napíš vlastný text. Môžeš poďakovať školiteľovi, konzultantovi, sponzorovi atď.

Obsah

[0 Úvod 6](#_Toc153434265)

[1 Problematika a prehľad literatúry 7](#_Toc153434266)

[1.1 Figma: Dizajn a prototypovanie 7](#_Toc153434267)

[1.2 HTML: Štruktúra webových stránok 7](#_Toc153434268)

[1.3 CSS: Estetika a rozloženie webových stránok 7](#_Toc153434269)

[1.4 JavaScript: Interaktivita a dynamika webových stránok 8](#_Toc153434270)

[1.5 Visual Studio Code: Efektívne Vývojové prostredie 9](#_Toc153434271)

[1.5.1 Extensions: Efektívnejšie práca 9](#_Toc153434272)

[1.6 Git 9](#_Toc153434273)

[1.7 GitHub 10](#_Toc153434274)

[2 Ciele práce 11](#_Toc153434275)

[3 Materiál a Metodika 12](#_Toc153434276)

[4 Závery práce a zhrnutie 13](#_Toc153434277)

[5 Zoznam použitej literatúry 14](#_Toc153434278)

[6 Prílohy 16](#_Toc153434279)

# Úvod

V súčasnom, digitálnom veku, kľúčovú úlohu zohráva kyberbezpečnosť. Rozvojom digitálnej technológie a internetu je nevyhnutné, aby sme sa po internete pohybovali bezpečne ale aj, aby sme rozumeli potenciálnym technikám a hrozbám, ktoré ohrozujú našu digitálnu bezpečnosť. V práci sa budeme zaoberať dvoma kľúčovými aspektmi kyberbezpečnosti – prevenciou a pochopením techník používaných hackermi. Náš každodenný život je prepojený s internetom napríklad ako komunikáciu, prácu, nákupy alebo zábavu. Zvyšuje sa neustále riziko kybernetických hrozieb napríklad: útoky tretích strán na osobne údaje, heslá a citlivé informácie. Táto práca má za cieľ poskytnúť čitateľovi ucelené informácie aby sa s maximálnou bezpečnosťou mohol pohybovať po internete, ochraniť svoje digitálne stopy a vyhnúť sa potenciálnym nebezpečenstvám.

Okrem prevencie sa budeme venovať aj rôznym technikám používaných v oblasti hackovania. Je dôležite poznať tieto techniky aby sme zlepšili prevenciu a lepšie pochopili ako bezpečne používať technologie. Pár technik, o ktorých budeme hovoriť sú napríklad: „Cross-site-scripting“ (XSS), „Packet Sniffing,“ „Privilige Escalation“ a ďalšie. Cieľom práce je vybaviť čitateľov nástrojmi a znalosťami, ktoré umožnia bezpečnejšie využívanie internetu a lepšie chápanie kyberbezpečnosti.

# Problematika a prehľad literatúry

Táto kapitola slúži na to aby sme sa oboznámili s programami alebo technológiami, s ktorými sme pri práci na projekte pracovali alebo ich inak používali k zlepšeniu alebo uľahčeniu práce. Súčasťou tejto kapitoly bude vysvetlenie dôležitých pojmov alebo skratiek s ktorými sme sa počas vyhotovenia práce stretli.

## Figma - Dizajn a prototypovanie

Figma je webová aplikácia na návrh dizajnu alebo softvéru. Je to webová aplikácia a cloudová platforma, ktorú môžeme používať v tíme znamená to, že môžeme pracovať v reálnom čase viacerí na projekte. Figme používame na tvorbu grafických návrhov, užívateľských rozhraní (UI) a na prototypovanie.

Je to populárny nástroj, ktorý každodenne používajú dizajnéri na celom svete. Ponúka veľkú flexibilitu a zefektívňuje prácu medzi členmi tímu, ktorý pracujú na rôznych digitálnych projektoch.

## HTML - Štruktúra webových stránok

Hypert Text Markup Language alebo v skratke HTML je hypertextový značkovací jazyk vďaka, ktorému sme schopný vytvárať webové stránky. Je zložený z tagov inak povedané značiek. Tieto značky určujú, ako sa zobrazujú jednotlivé elementy na stránke. Tagy alebo teda značky sa píšu v lomených zátvorkách <> a rozlišujeme ich na párové a nepárové. Rozdiel je ten, že párové tagy majú úvodnú a aj koncovú značku. Vyzerá to takto - **<tag>obsah</tag>**. Naopak nepárové sa nemusia nijak ukončiť. To znamená že majú len ten začiatok <tag>.

HTML slúži na zobrazovanie, to znamená, že stránka nie je interaktívna. Nedokážeme pomocou HTML vytvoriť napríklad hru alebo nejakú prezentáciu. Pre to aby stránka bola interaktívna “dynamická” musíme pridať kód z iného jazyka. Napríklad ako sme použili aj v práci JavaScript.

## CSS - Estetika a rozloženie webových stránok

CSS alebo Cascading Style Sheets po slovenský kaskádové štýly používame na to aby sme naším tagom, ktoré sme použili dali nejaký grafický výzor. CSS majú osobitný dokument, ktorý potom prepájame s HTML v hlavičke. CSS ma tak ako HTML jednoduchú konštrukciu. Pre niektorých by mohla byť nevýhoda to, že práve keď robíme túto konštrukciu tak musíme používať anglické výrazy pre určenie vlastnosti elementu ktorý chceme zmeniť. Štruktúra vyzerá nejak takto:

**h1{**

**color: red;**

**font-size: 25px;**

**}**

Na príklade vyššie by sme vykonali zmeny v h1 tagu (tento tag sa používa na nadpisy) a to také že by sme zmenili farbu písma na červenú a veľkosť písma na 25 pixelov.

V CSS môžeme aj trochu animovať. Môžeme na to použiť 2 spôsoby a to cez transitions alebo animations. Pomocou transitions vytvárame jednoduché efekty napríklad “:hover” tento použijeme vtedy keď chceme napríklad aby sa nám zmenila farba pozadia keď prejdeme myškou cez obrázok. Pomocou animations môžeme vytvárať zložitejšie animácie na toto používame “keyframes”. Tieto animácie sa môžu priblížiť animáciám, ktoré sú vytvorené v programoch ako je napríklad Adobe Flash alebo After Effect

## JavaScript - Interaktivita a dynamika webových stránok

JavaScript je skriptovací programovací jazyk. Kompletizuje nám svätú trojicu, ktorá sa používa pri vytváraní webových stránok. Má veľa využití ale používa sa hlavne pri webových stránkach aby sme ich dokázali urobiť interaktívne. Môžeme vytvoriť napríklad dotazník alebo nejaký test na webovej stránke. Môžeme vďaka nemu vytvoriť takzvanú webovú aplikáciu. Pomocou javascriptu dokážeme aj vytvárať animácie alebo zlepšovať už vytvorené animácie v CSS. Javascript bol prioritne určený na zhotovovanie klientskych častí aplikácií ale s príchodom nových Node.js a podobných technológií dokáže figurovať aj na strane servera.

Javascript môžeme písať v vlastnom súbore s koncovkou “.js” ale aj v súbore kde je HTML.V prípade, že píšeme javascript v súbore s html musíme vždy písať do tagov určených pre javascript. Tieto sa musia vždy nachádzať na úplnom konci HTML súboru. Napríklad v tomto kóde môžeme vidieť syntax javascriptu a tag, ktorý sa používa na vyhradenie miesta pre javascript.

**<script>**

**function myFunction() {**

**document.getElementById("demo").innerHTML = "Ahoj!";**

**}**

**</script>**

Je dôležité povedať, že JavaScript a Java sú úplne rozdielne programovacie jazyky.

## Visual Studio Code - Efektívne Vývojové prostredie

Softvér, ktorý slúži na editovanie zdrojového kódu. Prostredie tvorí editor je to hlavná časť celého programu. Ovládanie softvéru je ľahké ma rôzne klávesové skratky, ktoré nám uľahčujú prácu. Prácu si môžeme uľahčiť aj rôznymi rozšíreniami (extensions). Môžu pridávať ďalšiu funkcie. Ďalšiu výhodou je, že podporuje systém git. VS code alebo Visual Studio Code má zabudovanú podporu niektorých jazykov ako napríklad JavaScript.

### Extensions -Efektívnejšia práca

Existuje veľké množstvo rozšírení, ktoré vylepšujú funkčnosť alebo prispôsobujú editor tak aby sa nám lepšie pracovalo. Rozšírenia sú rôzne napríklad:

* **vzhľadové** – menia ako editor vypadá, menia font alebo pridávajú ikony pre súbory.
* **kód** – tieto nám pomáhajú písať kód napríklad namiesto toho aby sme písali celí názov funkcie tak nám ju hneď po pár písmenkách ponúkne alebo sa nám automaticky generujú aj koncove tagy.
* **Formátovanie** – takéto rozšírenia nám zase môžu uľahčovať prehľadnosť kódu. Môžu meniť farbu napríklad tagov alebo nejakých blokov.
* **Ostatné** – je mnoho rôznych rozšírení, ktoré robia veľmi rôzne veci. Napríklad sú také ktoré zvýrazňujú komentáre alebo nám púšťajú server.

V práci sme používali najviac LiveServer rozšírenie. Toto rozšírenie nám slúžilo na to aby sme vydelí zmeny na stránke hneď po uložený súboru to znamená, že nemusíme znovu načítať stránku stačí nám len uložiť zmeny.

## Git – Správa verzií

Git je distribuovaný systém pre správu verzií. Používame ho na sledovanie zmien v kóde a správu verzií projektov. Slúži nám na to aby sme efektívne mohli spolupracovať na projektoch. Vďaka gitu môžeme udržiavať prehľad o zmenách v kóde. Podporuje vetvenie (branching), zlučovanie (merging) a sledovanie histórie zmien. Pri gite musíme poznať tieto základne pojmy.:

* **Repozitár (Repository)** – Je to miesto, kde sa ukladá história projektu. Ukladajú sa všetky zmeny, vetvy, značky a súbory. Môže byť lokálny alebo vzdialený. V prípade vzdialeného môže pracovať na projekte viacero ľudí.
* **Komity (Commits)** – Základná jednotka zmien v git repozitári. Reprezentuje sadu zmien v súboroch a má jedinečný identifikátor. Pri každom commite sa vytvára správa o zmene a taktiež odkaz na predchádzajúci commit. Vytvárame tak stromovú štruktúru zmien v repozitári.
* **Vetvy (Branches)** – Vďaka vetvám môžeme pracovať na rôznych verziách kódu nezávisle od tej hlavnej vetvy. Môžeme experimentovať s kódom bez toho aby sme akokoľvek zasahovali do hlavného kódu teda hlavnej vetvy.
* **Zlučovanie (Merging)** – Môžeme kombinovať zmeny v jednej vetve do inej vetvy. Toto robíme vtedy keď máme vetvy dostatočne otestované a pripravené na to aby boli jedna z druhou zlúčene.

Git je najpoužívanejší systém pre správu verzií.

## GitHub

GitHub a Git sú dve odlišné veci. GitHub je poskytovateľ internetové hostingu, je určený na vývoj softvéru a správu verzií. Ponúka distribuovanú správu zdrojového kódu prostredníctvom systému git. Ma aj iné funkcie ako sú napr.: sledovanie nahlásených chýb, požiadavky na nove funkcie a správu úloh

GitHub je najobľúbenejší hlavne pre projekty, ktoré sú open-source. To sú projekty na ktorých môžu pracovať všetci, ktorý majú na to znalosti. GitHub je taktiež taká sociálna sieť pre programátorov. Môžeme tam ukladať naše projekty, ktoré môžeme následne ukazovať aj v prípade keď sa uchádzame o prácu. Ale hlavne je to sieť kde sa nemusíme báť ukázať náš kód pretože si tam programátori navzájom dokážu ľahko pomáhať bez toho aby priamo zasahovali do nejakého kódu.

# Ciele práce

Náš hlvný cieľ je vytvoriť webovú stránku, ktorá sa bude zameriavať na kyberbezpečnosť a, ktorá výbaví čitaľov znalostami, na ktoré sa budú môcť spolahnúť pri práci s internetom alebo inými technologiami. Snahou je aj urobiť curicculum, ktoré bude prehľadné. S tím súvisi aj dizajn webovej stránky. Dizajn by mal mať schopnosť nadchnúť a udržať čitateľa na webovej stránke. Implementáciou obsahu o kyberbezpečnosti chceme zdieľať aj rôzne interaktívne úlohy tretích strán, ktoré maju schopnosť nie len prehĺbiť čitateľové znalosti ale aj ich otestovať

# Materiál a Metodika

V tejto kapitole rozoberieme ako sme pracovali na tvorbe webovej stránky. Ako sme ju dizajnovali, ako sme ju naprogramovali a čo sme použili pre dosiahnutie našich cieľov.

## Inštalácia a konfigurácia Gitu a Githubu

Aby sme vykročili správnym krokom musíme stiahnuť Git z oficiálnej webovej stránky. Následne ho nainštalujeme. Po inštalácií sa nám otvorí špecialne CLI (command line interface) dedikované práve len pre Git. Toto okno môžeme zavreť. Nájdeme si alebo vytvoríme adresár, v ktorom budeme pracovať. Klikneme pravým a otvorime Git Bash. Pre to aby sme spustili projekt v danom adresári musíme použiť príkaz **git init**. Tento príkaz slúži na založenie nového projektu v adresári. Pri tomto príkazi sa vytvorí adresár .git, do ktorého Git ukladá všetky informácie o repozitári. Teraz potrebujeme nastaviť email a meno. Je to potrebné v prípade, že pracujeme v tíme, aby ostatny vedelí kto commitol (nahral) zmeny. Meno a email nastavíme pomocou dvoch príkazov.

**git config --global user.name ‘Naše meno’**

**git config --global user.email ‘Náš email’**

Po nastavení emailu a mena sa musíme zaregistrovať/prihlasiť na GitHub. Po tom ako sa dostaneme do GitHubu klikneme v pravo hore na plusko a následne na Nový repositor. Na tejto stránke nastavíme repositor. Pri zadávaní mena je dobré zadať meno adresáta, ktorý sme vytvorili pre projekt. V prípade že chceme môžme pridať nejaký komentár alebo zmeniť repositor na súkromný repositor. Ak chceme vytvoríme README file. Po tom, ako všetko nastavíme, ak chceme klikneme na vytvoriť repozitár. Aby sme mohli posielať súbory do repozitoru potrebujeme personálny vstupný token. Tento token vytvoríme v nastaveniach profilu. Do nastavení sa dostaneme tak, že klikneme v pravo hore na obrazok profilu následne na nastavenia. Po tom, ako sa dostaneme do nastavení výhladáme na lavej strane nastavenia developera. A znova na lavej strane nájdeme personálny vstupný token ten rozklikneme a klikneme na tokeny (klasické). Na tejto podstránke klikneme na tlačítko generovať nový token a klikneme na klasický. Token môžeme pomenovať a taktiež mu nastaviť dobu trvania. Následne musíme nastaviť čo môžeme robiť v prípade, že tento token používame. Keďže tento token budeme vlastniť len mi tak mu dáme všetký práva. Klikneme na generovať token čo nám už vygeneruje token, ktorý použijeme. Je dôležite tento token niekde uložiť pretože ho budeme na tejto podstránke vidieť len raz. Teraz sa pomocou Git bashu a tokenu, ktorý sme si vytvorili možeme dostať cez Git bash do repozitára, ktorý sme vytvorili. Na toto použijeme prikaz.

**git remote add origin main https://TOKEN@github.com/GitHub meno/meno repozitáru)**

Po tomto všetkom máme Git a Github vytvorený a nastavení. Teraz pri každej zmene budeme nahrávať zmeny na GitHub. Na nahravanie budeme potrebovať ďalšie príkazy.

* git status -
* git add (názov súboru) -
* git commit -
* git push -
* git pull -

# Závery práce a zhrnutie

Sem napíš vlastný text.

Napíš svoj názor na daný problém a jeho riešenie. Vyzdvihni prínos tvojej práce na daný problém a poukáž na spôsob realizácie. Záver by mal načrtnúť ďalšiu perspektívu práce v danej problematike so získanými poznatkami. Popíš tu tiež cieľ práce, metodiku a urob súhrn najdôležitejších zistení, výsledkov svojej práce. Čitateľ by mal po prečítaní tejto časti vedieť, o čom práca je a čo si zistil.

# Zoznam použitej literatúry

**Pre učebnice a monografie:**Priezvisko a meno autora, názov (kurzívou), podnázov, vydanie, vydavateľstvo (miesto a vydavateľ), rok vydania, počet strán monografie, štandardné číslo ISBN – viď [1]  
**Pre normy:**označenie a číslo normy, názov normy, lokalizácia (miesto, vydavateľ, rok vydania)– viď [2]  
**Pre elektronické zdroje**– viď [3]

[1]STECOVÁ, Eva a ŠVICKÝ, Emil. 1999: Asanácia. 1. vyd. Bratislava: Proxima Press, 1999. s. 72-82. ISBN 80-85454-26-2

[3]ISO 690-2: 1997, Information and documentation – Bibliographicreferences Part 2: Electronicdocuments or partsthereof.

[4]MINISTERSTVO FINANCIÍ SR: Metodický pokyn na použitie odborných výrazov pre oblasť informatizácie spoločnosti. Bratislava: MF SR, [cit. 2009-02-05].  
Dostupné na internete: http://www.informatizacia.sk/ext\_dok-metodicky\_pokyn\_glosar\_pojmov/3482c.

Moje zdroje

figma

<https://en.wikipedia.org/wiki/Figma>

html

https://www.webglobe.sk/blog/html-zaklady

<https://www.fine.sk/pojmy-a-skratky/html/>

css

<https://www.fine.sk/pojmy-a-skratky/css/>

<https://www.ayris.sk/ako-vytvorit-animaciu-v-css3-prve-kroky/>

javascript

https://www.ayris.sk/ako-vytvorit-animaciu-v-css3-prve-kroky/

<https://www.fine.sk/pojmy-a-skratky/javascript/>

VS code

<https://learn.microsoft.com/sk-sk/power-apps/maker/portals/vs-code-extension>

<https://uzivatel.cz/vs-code-rozsireni/>

git

<https://coderama.com/slovnik/git>

<https://skillmea.sk/blog/14-git-prikazov-ktore-musi-poznat-kazdy-vyvojar>

github

# Prílohy

Sem vlož veľké tabuľky, obrázky, diagramy, ilustrácie, grafy, schémy a podobne. Pre nadpisy príloh používaj: Príloha1 a Príloha2 a dodaj krátky popis, toho, čo sa nachádza v prílohe, napr.

Keď budeš mať prácu hotovú. Aktualizuj Obsah. Obsah sa generuje automaticky, vieš ho vytvoriť v priebehu pár sekúnd.:  
1. Prejdi na stranu 5 (Obsah).

2. Kurzor myšky presuň napr. na časť 5. Tam stlač pravé tlačidlo na myške.

3. Vyber: Aktualizovať pole.

4. Vyber. Aktualizovať celý obsah.

5. Máš hotovo.