

Izdelava računalniške igre – maturitetna naloga pri predmetu Informatika

Matic Kraševvec 4.D

Mentor: Klemen Bajec

Šmarje – Sap, šolsko leto 2021/2022

Ključne besede

- Prijava - login
- Obrazec - form
- Gumb - button
- Registracija - signup
- Odjava - logout
- Baza - database
- Tabela - table
- Seja - session
- Strežnik - server
- Točke - points
- Spremenljivka - variable
- Poizvedba - query
- Pripravljena izjava - prepared statement
- Ireverzibilna funkcija - irreversible function

Izvleček

Pri teoretičnem delu te projektne naloge raziskujem kaj je prijavni sistem, zakaj se uporablja, kdaj so se začeli uporabljati in njihov razvoj skozi zgodovino, ter kako deluje. Poleg tega raziskujem tudi ranljivosti takega sistema, ter s kakšnimi varnostnimi sistemi se zaščitimo pred kibernetскими napadi. V praktičnem delu pa opisujem kako sem v jeziku PHP in SQL naredil spletno računalniško igro žanra »kliker«, ki napredek igralca sproti shranjuje v bazo, kar mu omogoča igranje na kateri koli napravi z dostopom do interneta.

In the theoretical part of this project, we discover what a login system is, what it's used for, when did we start using them and how it works. We also take a look at the weaknesses of such a system, and what safety precautions we use to defend against a cyber-attack. In the practical part of this assignment, I describe the process of making a "clicker" type online game using PHP and SQL, that stores the player's progress in the database, and allows him to play on any device with access to the Internet.

Kazalo

Izdelava računalniške igre – maturitetna naloga pri predmetu Informatika	1
Ključne besede	2
Izveček.....	2
1 – Uvod: Kaj je prijavni sistem?	4
2 – Teoretične osnove prijavnega sistema	4
3 – Zgodovina prijavnih sistemov	4
4 – Varnost pri postopku registracije	5
5 – Varnost prijavnih sistemov	5
6 – Praktični del projektne naloge: Prijavni sistem	8
7 – Praktični del projektne naloge: Spletna igra tipa »kliker«.....	11
8 – Zaključek	12
Viri.....	13

1 – Uvod: Kaj je prijavni sistem?

Prijavni sistem je sistem večih funkcij na spletni strani. To so registracija, prijava, odjava in seja. Kombinacija teh funkcij omogoča uporabniku stvaritev in upravljanje svojega osebnega računa, ter shranjevanje in prikazovanje prilagojene vsebine na spletnem mestu.

2 – Teoretične osnove prijavnega sistema

2.1 – Baza

Na spletnem strežniku imamo bazo, ki vsebuje tabele polne podatkov registriranih uporabnikov. Do teh dostopamo s pomočjo »mysql« funkcij. Te podatke lahko nato prikažemo uporabniku, uporabimo pri procesih na spletni strani ali validiramo ujemanje z uporabnikovimi vnosi.

2.2 – Registracija

Registracija je proces pri katerem uporabnik v obrazec navede svoje osebne podatke in geslo, s katerimi se bo lahko kasneje zopet prijavil v svoj račun. Program te podatke pregleda in potrdi, da ni nobenih napak (že obstoječa imena itd.). Če ne najde nobenih napak, geslo šifrira in vnesene podatke zapiše v tabelo uporabnikov v bazi na strežniku.

2.3 – Prijava

V postopku prijave program pregleda ali se vneseni podatki ujemajo z tistimi v bazi. Če se vsi podatki ujemajo, nas program vpiše in prične sejo. Ko ima uporabnik zagnano sejo, mu spletna stran lahko prikazuje personalizirano vsebino.

2.4 – Odjava

Ko se želimo iz svojega računa odjaviti, program prekine sejo in nas pošlje na začetno stran.

3 – Zgodovina prijavnih sistemov

Prijavni sistemi so se začeli uporabljati že v poznih 60ih letih, z uvedbo novih operacijskih sistemov (Windows NT in Linux), pa je njihova prisotnost eksponentno narasla z uporabo na domačih računalnikih. Celoten koncept se čez čas ni spremenil dosti, je pa zato doživel veliko posodobitev na področju varnosti in zasebnosti. (Povzeto po:

https://en.wikipedia.org/wiki/Login#History_and_etymology)

4 – Varnost pri postopku registracije

Ustvarjanje računa je prvi korak do uporabe spletne strani, ki ima implementiran prijavi sistem, zato je ključno, da razvijalec/skrbnik spletnega mesta poskrbi, da ta proces deluje karseda gladko in še pomembneje, varno. Zlorabo registracijskega sistema preprečimo z različnimi metodami, s katerimi validiramo zahtevo po registraciji novega računa in tako preprečimo krajo identitete ali podatkov, ter zaščitimo storitev pred zlonamerno programsko opremo.

4.1 – Potrditveni e-mail ali SMS

Dandanes ima večina spletnih prijavnih sistemov vgrajeno metodo validiranja registracije s pomočjo potrditvenega e-poštnega sporočila. Ta po poskusu registracije v bazo poslan na e-poštni naslov, ki ga uporabnik, ki se želi registrirati vpiše v registracijski obrazec. Sporočilo vsebuje povezavo do potrditvene strani. S klikom na potrditveno povezavo uporabnik dokaže lastništvo na vpisanem e-naslovu, ter tako zaključi registracijo. V primeru, da tak sistem ni vpeljan na spletni strani, se uporabniki nanjo lahko registrirajo brez potrjevanja identitete, kar odpira vrata raznim nepridipravom, ki lahko to izrabijo za krajo identitete in podobne kibernetске napade.

4.2 – Dokazovanje človečnosti in DDOS

Veliko nevarnost spletnim stranem predstavljajo programske skripte, ki so napisane z namenom destabiliziranja in motenjem poteka delovanja sistemov na spletni strani. To dosežejo s pomočjo kibernetškega napada vrste DOS/DDOS. DOS (denial of service) je programska koda, ki strežnik preplavi z več-tisoč hkratnimi prošnjami za storitve, ki jih najdemo na spletni strani. Pri varianti DDOS (distributed denial of service) za to uporabi več računalnikov imenovanih »zombiji«, ki od hekerja dobivajo ukaze za izvedbo DOS napada. Tako ima napadalec na voljo več procesorske moči, zaradi česar je ta metoda toliko bolj uničujoča. Ker strežnik vsem tem prošnjam ne uspe ugoditi, sistem zmrzne, ter tako onemogoči normalno delovanje storitev na spletni strani. V takem stanju ostane, dokler heker ne ustavi pošiljanja novih prošenj.

Napade tipa DOS lahko preprečimo z implementacijo sistemov za preverjanje človečnosti. To so izzivi, katere človek enostavno reši, programska skripta pa ne. Poznamo več vrst teh izzivov, od prepisovanja niza znakov, do označevanja slik, ki vsebujejo določene elemente iz vsakdanjega življenja (semaforji, prevozna sredstva, itd.). Z uporabo teh izzivov tako eliminiramo vsak poskus upravljanja spletne strani, ki ni človeškega izvora, ter posledično preprečimo morebitne DOS napade. (Povzeto po: https://en.wikipedia.org/wiki/Denial-of-service_attack)

5 – Varnost prijavnih sistemov

Prijavi sistem omogoča vsakemu uporabniku razpolagati z prilagojeno vsebino na njegovem zasebnem računu. Ker želimo, da do svojega lastnega računa lahko dostopa le oseba, ki je

račun registrirala in ne neprivdipravi, ki pri vdiranju v tuje račune povzročajo škodo, skoraj vsi prijavni sistemi uporabljajo vsaj eno ali več plasti zaščite, ki uporabniku omogočajo spletno varnost in varujejo njegove podatke. Na spletnem mestu, ki teh varnostnih sistemov ne uporablja, so računi uporabnikov izpostavljeni vdorom tujih oseb, ki lahko uporabniku ukradejo podatke, denar ali identiteto.

Za varovanje podatkov legitimnih uporabnikov spletne strani se uporablja več načinov oziroma plasti zaščite, ki spletni račun zavarujejo pred različnimi pristopi vdorov v račun.

5.1 – Geslo

Geslo je najstarejša in najbolj osnovna oblika varovanja spletnega računa. Gre za niz znakov (črke, števila), ki ga uporabnik določi pri postopku registracije, ko ustvarja svoj račun. Ko se želi vanj prijaviti, mora poleg svojega uporabniškega imena ali naslova elektronske pošte vpisati tudi svoje geslo. Program nato preveri ali se vpisano geslo ujema s tistim, ki je v bazi zapisano poleg imena računa v katerega se uporabnik želi prijaviti. Če se geslo ujema, ga vpiše, v nasprotnem primeru, pa mu program poskus prijave zavrne.

Gesla se v času, odkar smo jih začeli uporabljati pri računalniški avtentikaciji, niso dosti spremenila. Je pa, z razvojem boljših in hitrejših računalnikov, potreba po daljših in bolj zapletenih geslih, ki jih je težje uganiti, narasla. Spletne strani zato uporabnika pri postopku registracije pozovejo k uporabi čim bolj kompleksnega gesla, ki neprivdipravom oteži in upočasni vdiranje z uporabo metode imenovane »brute force«, pri kateri program po vrsti vpisuje vsa možna gesla, dokler ne vpiše pravega in tako pridobi dostop do računa, v katerega vdira.

V primeru vdora v bazo podatkov, pa lahko hekerji geslo pridobijo tudi neposredno iz tabele v kateri je zapisano, zato je ključno, da je geslo šifrirano s pomočjo ireverzibilnih funkcij »hash«. Te funkcije nam omogočajo, da uporabnikovo geslo zašifriramo in ga v novi obliki zapišemo v tabelo. To hekerjem prepreči množično krajo gesel, saj šifriranih gesel ne morejo hitro dešifrirati, zahvaljujoč enosmernosti hash funkcij. Žal se nekateri bolj organizirani vdiralci tej časovni zamudnosti lahko izognejo tako, da šifrirane vrednosti gesel izračunajo vnaprej, ter si jih poleg originalnih gesel zapišejo v razpredelnice, imenovane »mavrične tabele«. To jim omogoča, da šifrirana gesla, ki jih pridobijo iz baze primerjajo z šifriranimi vrednostmi v mavričnih tabelah in tako pridobijo uporabnikovo originalno geslo, s katerim pridobijo dostop do njegovega računa. Na srečo obstaja tudi metoda, ki skoraj popolnoma negirajo uporabnost mavričnih tabel, imenovana soljenje gesel (password salting). Pri tej metodi se pri postopku registracije uporabnikovemu originalnemu geslu doda določen niz znakov, imenovan sol. To »soljeno« geslo je nato zašifrirano s hash funkcijo, ter je popolnoma drugačno od nesoljeno šifriranega gesla, kar onemogoči uporabo mavričnih tabel. (Povzeto po: https://en.wikipedia.org/wiki/Hash_function)

5.2 – Biometrični podatki

Kot nadomestek gesla, nam nekatere storitve omogočajo prijavo s pomočjo biometričnih atributov fizičnega uporabnika. Tak način avtentikacije je varnejši, saj ima vsak lastnik računa edinstvene fizične lastnosti, kot so prstni odtis, obrazne poteze, glasovni vzorci, skeniranje šarenice in podobno. Pri izbiri biometričnega identifikatorja je pomembno, da je uporabnikov atribut res unikatni, saj v drugem primeru lahko do našega računa dostopa vsakdo, ki poseduje enake karakteristike (dvojčki imajo enak obraz).

Kljub temu, da je ta sistem varnejši od uporabe gesel, žal še ni tako pogosto uporabljen. To je zaradi dejstva, da za preverjanje pristnosti vsak uporabnik potrebuje biometrični čitalec, ki lahko preverja fizične attribute posameznika (bralec prstnih odtisov, skener obraza...). Na srečo se ti vedno pogosteje pojavljajo na novejših mobilnih napravah, kot so mobilni telefoni, tablični in prenosni računalniki, kar omogoča biometrično avtentikacijo več uporabnikom.

5.3 – Dvojna avtentikacija

Za dodaten nivo varnosti poskrbi uporaba dvojne avtentikacije (»two factor authentication« ali 2FA). Gre za dodatno preverjanje pristnosti zahteve dostopa do računa, saj od uporabnika poleg uporabniškega imena in pravega gesla, zahteva tudi potrditev preko vnaprej določenih zunanjih virov, do katerih ima dostop le upravičeni uporabnik, ki je ustvaril račun in te vire določil.

Viri potrditvenih kod za uporabo dvojne avtentikacije:

- **podana preko fizičnega vira**
Najstarejši način dvojne avtentikacije je izveden preko fizične zunanje naprave, ki prikazuje unikatno kodo, ki se zamenja po določenem časovnem intervalu. To kodo mora uporabnik vpisati v prijavní obrazec preden se ta zamenja. Dodatno varnost imajo avtentikatorji bančnih kartic, ki pred prikazom varnostne kode zahtevajo vstavev kartice in pripadajočo kodo PIN. Ta vir je sicer varen, a ker mora vsak uporabnik posedovati fizični avtentikator, metoda hitro postane predraga in nepriročna.
- **generirana v aplikaciji za avtentikacijo**
Modernejša, digitalizirana alternativa so aplikacije za avtentikacijo kot so »Google Authenticator«, »Lastpass« in »Authy«. Te aplikacije uporabniku, podobno kot njihovim fizičnim predhodnikom, podajo novo geslo po določenem časovnem intervalu. Ker pa je ta način zasnovan na programski osnovi, ne potrebuje specializirane naprave, ampak je lahko dostopen na vseh napravah, kamor to aplikacijo lahko prenesemo. Posledično je ta metoda cenejša in v današnjem času veliko bolj razširjena.

- **preko SMS sporočila ali elektronske pošte**

Avtentikacijsko kodo nam lahko generira tudi ponudnik spletne storitve, v katero se želimo prijaviti. To kodo nam nato pošlje preko metode prenosa, katero smo izbrali in avtorizirali za avtentikacijo. Navadno se uporablja SMS in E-poštna sporočila. Kljub svoji razširjenosti, pa ta način žal ni tako varen kot prejšnja, kjer so kode generirane pri uporabniku, saj jih lahko vlomilci med pošiljanjem prestrežejo in uporabijo.

- **avtentikacija preko potisnega obvestila**

Spletna mesta (kot je Googlov »Gmail«) nam omogočajo avtentikacijo na napravi tudi preko gumba, ki ga dobimo znotraj potisnega obvestila na vnaprej izbrani napravi. Tako lahko samo z enim pritiskom gumba dokažemo legitimnost svoje prijave. V primeru, da se v račun poskuša prijaviti neznana oseba, lahko izberemo gumb, ki zahteva po prijavi zavrne, ter tako preprečimo morebitni vdor v račun.

5.4 – Pripravljeni stavki

Največjo nevarnost prijavnemu sistemu predstavlja način vdiranja imenovan »SQL vrivanje« (SQL injection). To je najbolj razširjena metoda vdiranja v baze na strežnikih, preko spletnih strani, ki pred vrivanjem niso zavarovane. Pri tem načinu vdora heker v obrazec namesto zahtevanega niza, vpiše SQL kodo, s katero lahko nato znotraj obrazca dostopa in ureja podatkovno bazo spletne strani. Tako lahko pridobi uporabniška imena in pripadajoča gesla, ter posledično dostop do kateregakoli računa na spletni strani.

Na srečo se lahko spletno storitev pred vrivanjem zaščiti z uporabo tako imenovanih »pripravljenih stavkov« (prepared statements). To so nepopolni deli SQL kode znotraj spletne strani, ki so napisani z uporabo začasnih znakov (placeholders), ki jih v stavku označimo z vprašaji ('?'). Pri poskusu prijave se vprašaji z uporabo funkcije »bind_param()« zamenjajo z vsebino obrazcev, ki jo napiše uporabnik, ki se želi prijaviti. Tako napisan program lahko skozi obrazec sprejme le prijavne podatke in nato varno izvede navodila SQL stavka. (Povzeto po: <https://www.hackedu.com/blog/how-to-prevent-sql-injection-vulnerabilities-how-prepared-statements-work#:~:text=A%20prepared%20statement%20is%20a,safely%2C%20preventing%20SQL%20Injection%20vulnerabilities>)

6 – Praktični del projektne naloge: Prijavni sistem

Za praktični del projektne naloge sem naredil spletno stran z delujočim prijavnim sistemom, ki se poveže v bazo na šolskem strežniku. (Projekt je v času pisanja te naloge dostopen na <https://vaje.gimvic.org/inf4/KrasevecM/index.php>).

Celoten projekt je trenutno omejen na 9 PHP datotek, ki so bile napisane v beležnici Notepad++, verziji 8.3.3.

A. Povezava do baze (dbh.inc.php)

V tej datoteki določimo parametre (uporabniško ime in geslo za dostop do strežnika) in jih z vgrajeno funkcijo »mysqli_connect()« zapišemo v spremenljivko \$conn. To spremenljivko kasneje kličemo vsakič, ko potrebujemo dostop do baze.

B. Glava in navigacija (header.php)

Ta dokument vsebuje glavo in gumbе za navigacijo (domov, registracija, prijava/odjava). Z vsakim dokumentom je povezana s funkcijo »include_once«. To je zato, da nam pri dodajanju ali odstranjevanju odstrani ni treba spreminjati vsakega dokumenta posebej. V primeru, da je uporabnik prijavljen in seja odprta, se v navigacijski vrsti izpiše tudi uporabnikovo ime in ID število.



Slika 1: Navigacijska vrstica

C. Domača stran (index.php)

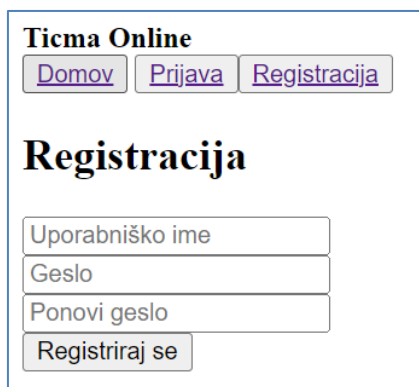
To je stran, ki jo uporabnik vidi, ko odpre spletno mesto. Vsebuje glavo, ki jo pridobi iz header.php, ter, če je uporabnik prijavljen, vsebino spletne igre. To je gumb za dodajanje točk, na katerem piše »Klikni me«; vrstica, v katerem so prikazane točke trenutno prijavljenega uporabnika, ter ime uporabnika z največ točkami.

D. Strani za registracijo (signup.php, signup.inc.php)

Stran za registracijo vsebuje poleg pripete glave še obrazec, kamor uporabnik vpiše svoje podatke (uporabniško ime, geslo, ponovno geslo za varnost), s katerimi se želi registrirati.

Te vrednosti nato pobere datoteka signup.inc.php. To je skripta, ki poganja program v postopku registracije. Povezana je z datotekama dbh.inc.php (skripta za povezavo do baze) in functions.inc.php (skripta s funkcijami). Program najprej pregleda vse vnesene podatke in išče napake. Če napak ni, uporabnika z definirano funkcijo »createUser« registrira v bazo. Pred zapisom v bazo se uporabnikovo geslo s pomočjo funkcije »password_hash()« ireverzibilno zašifrira, kar nam zagotavlja dodatno varnost podatkov.

Če se uporabnik registrira narobe (zasedeno ime, uporabnik pusti polja prazna, vneseni gesli se ne ujemata...), mu program pod obrazcem javi napako. V primeru uspešne registracije, je o tem prav tako obveščen.



Ticma Online

[Domov](#) [Prijava](#) [Registracija](#)

Registracija

Uporabniško ime

Geslo

Ponovi geslo

Registriraj se

Slika 2: Obrazec za registracijo

E. Strani za prijavo (login.php, login.inc.php)

Na strani za prijavo najdemo podoben obrazec, kot pri registraciji, le da tu ponovni vpis gesla ni potreben. Tukaj uporabnik vpiše podatke, s katerimi se je registriral. Te podatke prevzame program login.inc.php. Ta najprej podatke preveri za morebitne napake (neobstoječe ime, napačni podatki, uporabnik pusti polja prazna...). Če napak ni, uporabnika z definirano funkcijo »loginUser« prijavi v spletno stran in prične sejo.



Ticma Online

[Domov](#) [Prijava](#) [Registracija](#)

Prijava

Uporabniško ime

Geslo

Prijavi se

Slika 3: Obrazec za prijavo

F. Stran z funkcijami (functions.php)

To je stran, kjer so zbrane skoraj vse definirane funkcije v tem sistemu za lažji pregled. Vse strani, ki kličejo funkcijo iz tukaj, se morajo do nje povezati z metodo »require_once''. Te funkcije izvajajo procese na spletni strani in v bazi, preverjajo vsebino in postopke za napake in uspešne storitve, ter vračajo ustrezne rezultate.

```

<?php
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);

function emptyInputSignup($username, $pwd, $pwdRepeat) {
function pwdMatch($pwd, $pwdRepeat) {
function uidExists($conn, $username) {
function createUser($conn, $username, $pwd) {
function emptyInputLogin($username, $pwd) {
function loginUser($conn, $username, $pwd) {
function pointsquery($conn) {
function increase($conn, $points) {
function top($conn) {
?>

```

Slika 4: Dokument za funkcije

G. Odjava (logout.inc.php)

Te datoteke uporabnik ne vidi, saj vsebuje le nekaj vrstic kode, ki zaključijo trenutno sejo in s tem izpišejo uporabnika.

```

<?php
session_start();
session_unset();
session_destroy();
header("location: index.php");
exit();

```

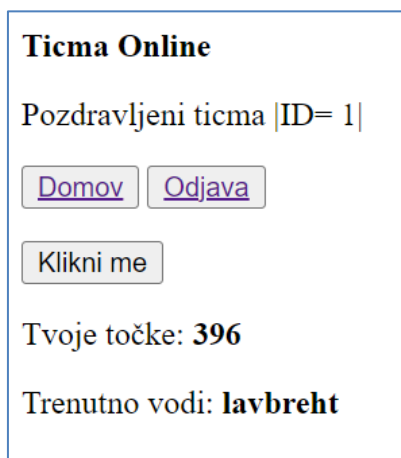
Slika 5: Koda za odjavo uporabnika

7 – Praktični del projektne naloge: Spletna igra tipa »kliker«

Ko sem postavil spletno stran z delujočim prijavnim sistemom, sem na začetni strani sprogramiral preprosto igro, katero lahko igra kdorkoli, ki je registriran v moji bazi. Dejstvo, da je igra spletna, uporabniku omogoča igranje na katerikoli napravi z dostopom do interneta, saj se njegov napredek shranjuje znotraj tabele uporabnikov.

Igra v mojem projektu je tipa »kliker«. Gre za preprost koncept, kjer je glavni cilj igralca to, da poseduje karseda veliko točk. Točke pridobiva s pritiskanjem na gumb, na katerem piše »Klikni me«; in sicer eno točko za vsak klik. Igra tako preverja igralčevo hitrost klikanja ter njegovo vztrajnost. Za »nagrado« in kot »dokaz o superiornosti«, je ime igralca z največ točkami prikazano vsem igralcem, ki igro igrajo.

Igro na spletni strani sestavljajo le trije igralcu vidni elementi, to so: gumb za dodajanje točk, vrstica za prikaz stanja točk trenutno prijavljenega uporabnika in vrstica za izpis uporabniškega imena igralca z največ točkami.



Slika 6: Igra kliker

Za delovanje, igra potrebuje le tri definirane funkcije:

1. **Pointsquery:**

Ta funkcija s povezavo do baze in pomočjo funkcije »mysqli_query()« in SQL »SELECT« metode vrne uporabnikovo trenutno število točk, vsakič ko ta pritisne na gumb. Te točke so nato s PHP metodo »echo''« prikazane na spletni strani igre.

2. **Increase:**

Funkcija »increase« od funkcije »pointsquery« prejme uporabnikovo trenutno stanje točk, ga nato poveča za ena z uporabo pripravljenega stavka in SQL »UPDATE« metodo.

3. **Top:**

Funkcija »top« je neodvisna od drugih dveh funkcij, deluje pa tako, da ob vsaki osvežitvi strani s PHP metodo »echo''« prikaže uporabniško ime igralca, ki ima trenutno najvišje stanje točk. To izvede z uporabo SQL metode »SELECT MAX« in funkcije »mysqli_query()«.

8 – Zaključek

Prijavni sistem je zapleteno sodelovanje več programov, ki delujejo istočasno. Sistem je nepogrešljiv, saj uporabniku omogoča dostop do ekskluzivnih vsebin, skrbniku baze pa pregled nad uporabniki in primerno upravljanje storitev. Pri zasnovi teh je potrebno biti pozoren na implementacijo varnostnih plasti in pregrad, ki podatke uporabnikov ščitijo pred zlonamernimi dejanji. Če je vse narejeno pravilno in korektno, dobimo okolje in orodja za kreiranje česarkoli si zamislimo, od forumov, do preprostih videoiger, kot je moj kliker.

- **WIKIPEDIA. 2021. Login[online].**[Datum zadnjega popraviljanja 26.3.2021] [datum ogleda 1.6.2021]. Dostopno na spletnem naslovu: <https://en.wikipedia.org/wiki/Login>
- **TUTORIAL REPUBLIC. 2021. PHP MySQL Login System[online].**[datum ogleda 24.5.2021]. Dostopno na spletnem naslovu: <https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php>
- **W3SCHOOLS. 2021. PHP Tutorial[online].**[datum ogleda 23.5.2021]. Dostopno na spletnem naslovu: <https://www.w3schools.com/php/DEFAULT.asp>
- **DZONE. 2021. Create a Login System Using HTML, PHP, and MySQL [online].**[Datum zadnjega popraviljanja 20.5.2020] [datum ogleda 25.5.2021]. Dostopno na spletnem naslovu: <https://dzone.com/articles/ceate-a-login-system-using-html-php-and-mysql>
- **AUTHY. 2022. What Is Two-Factor Authentication (2FA)? [online]**[datum ogleda 9.4.2022]. Dostopno na spletnem naslovu: <https://authy.com/what-is-2fa/>
- **HACKEDU. 2022. How to prevent SQL Injection vulnerabilities: How Prepared Statements Work [online]** [datum ogleda 6.4.2022]. Dostopno na spletnem naslovu: <https://www.hackedu.com/blog/how-to-prevent-sql-injection-vulnerabilities-how-prepared-statements-work#:~:text=A%20prepared%20statement%20is%20a,safely%2C%20preventing%20SQL%20Injection%20vulnerabilities.>
- **GEEKFLARE. 2022. 7 Best Two-Factor (2FA) Authentication Apps to Protect Your Email and Social Media [online]**[datum ogleda 9.4.2022]. Dostopno na spletnem naslovu: <https://geekflare.com/two-factor-authentication-apps/>
- **W3SCHOOLS. 2022. PHP MySQL Prepared Statements [online]**[datum ogleda 26.5.2021]. Dostopno na spletnem naslovu: https://www.w3schools.com/php/php_mysql_prepared_statements.asp#:~:text=A%20prepared%20statement%20is%20a,and%20sent%20to%20the%20database.
- **WIKIPEDIA. 2022. SQL injection [online]**[datum ogleda 10.4.2022]. Dostopno na spletnem naslovu: https://en.wikipedia.org/wiki/SQL_injection
- **PHP MANUAL. 2022. mysqli_query [online]**[datum ogleda 10.4.2022]. Dostopno na spletnem naslovu: <https://www.php.net/manual/en/mysqli.query.php>
- **W3SCHOOLS. 2022. SQL Tutorial [online]**[datum ogleda 26.5.2021]. Dostopno na spletnem naslovu: <https://www.w3schools.com/sql/>
- **WIKIPEDIA. 2022. Denial of service [online]**[datum ogleda 12.4.2022]. Dostopno na spletnem naslovu: https://en.wikipedia.org/wiki/Denial-of-service_attack