

Sigurnost web aplikacija

V. FAZA PROJEKTA

PROGRAMIRANJE WEB APLIKACIJA

Cilj posljednje faze projekta je upoznavanje s ranjivostima web aplikacija te dodavanje zaštite pristupa administracijskom dijelu stranice.

U sklopu pete faze projekta potrebno je dodati tablicu *korisnik* u bazu podataka. Potom je potrebno zaštititi pristup dijelu stranice za administraciju administrator.php iz 3. faze uvođenjem provjere korisničkog imena i lozinke. Ukoliko su uneseni ispravno korisničko ime i lozinka te korisnik ima administratorska prava treba korisniku prikazati administracijsku stranicu. Ukoliko je uneseno ispravo korisničko ime i lozinka te korisnik nema administratorska prava treba korisniku prikazati poruku s njegovim imenom i upozorenjem da nema pravo za pristup administratorskoj stranici. Ukoliko nije uneseno ispravno korisničko ime i/ili lozinka potrebno je korisniku izbaciti poruku da se mora prvo registrirati i prikazati link na formu za registraciju. Također je potrebno kreirati stranicu naziva registracija.php s formom za registraciju korisnika, te pripadnu skriptu za unos korisnika u bazu podataka. Nakon što to napravite potrebno je testirati ranjivost web aplikacije na SQL injection, te ispraviti tu ranjivost korištenjem PHP prepared statementa.

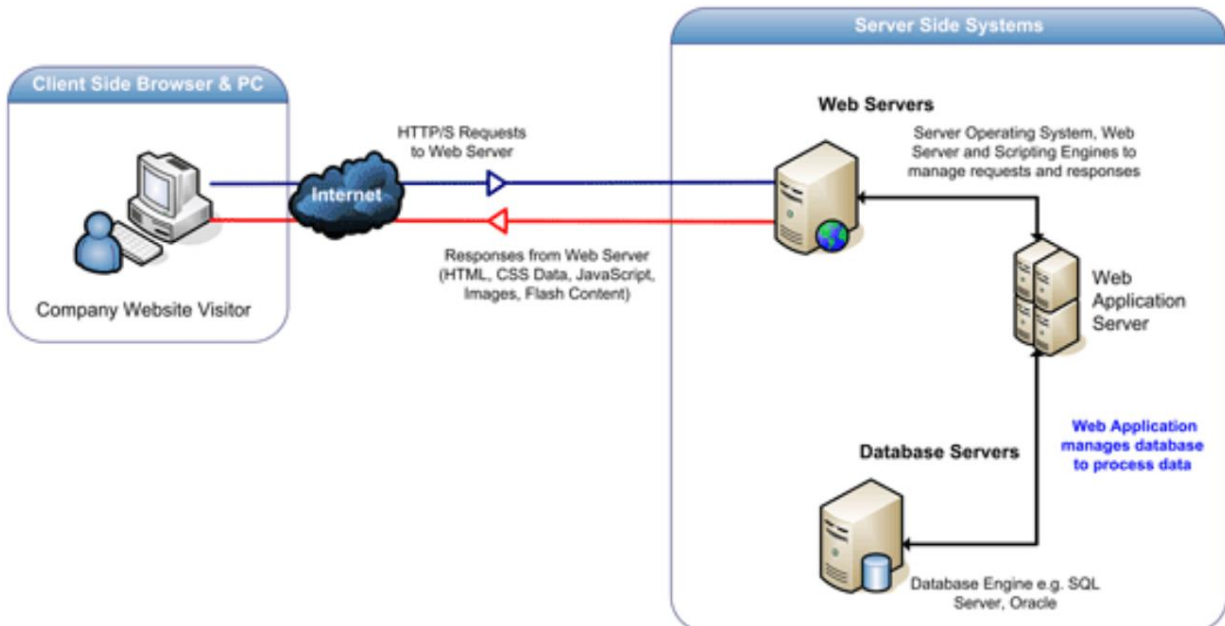
Ranjivost web aplikacija

Open Web Application Security Project (OWASP) je open source projekt sigurnosti web aplikacija. OWASP zajednica sadrži korporacije, edukacijske organizacije i pojedince iz cijelog svijeta. Svake godine OWASP navodi 10 najvećih ranjivosti web aplikacija. Zadnja lista objavljena je 2010. godine i od tada se nije puno promijenila.

10 najvećih ranjivosti web aplikacija su:

1. Injection (Ubacivanje)
2. Cross-site scripting (XSS)
3. Broken authentication and session management (puknuta autentifikacija i rad sa sjednicama)
4. Insecure Direct object references (nesigurne reference direktnih objekata)
5. Cross Site Request Forgery (CSRF)
6. Security misconfiguration (loša konfiguracija sigurnosti)
7. Failure to restrict URL access (neuspjelo ograničavanje URL pristupa)

8. Insecure cryptographic storage (nesigurna kriptografska pohrana)
9. Insufficient Transport Layer Protection (nadovoljna zaštita transportnog sloja)
10. Unvalidated Redirects and Forwards (nevalidirana preusmjerenja i prosljeđivanja)



Slika 1- Prikaz osnovnog modela rada web aplikacije

Gornja slika prikazuje osnovni model rada web aplikacija. Na klijentskoj strani osnovna komponenta je preglednik. Preglednik i poslužitelj komuniciraju pomoću transportnog protokola. Kada korisnik klikne na URL web aplikacije, preglednik šalje HTTP zahtjev poslužitelju. Poslužiteljska skripta procesira upit i po potrebi šalje upit prema bazi podataka. Zatim se formira HTTP odgovor s HTML stranicom koja se šalje pregledniku. Postoje napadi koji iskorištavaju ranjivosti preglednika i napadi koji iskorištavaju ranjivosti na poslužitelju. Ovdje ćemo se fokusirati na SQL injection.

Prijava u web aplikaciju

Potrebno je u bazi podataka napraviti tablicu naziva *korisnik* koja mora sadržavati attribute id, ime, prezime, korisničko ime, lozinka, razina. Također, potrebno je kreirati stranicu za login naziva koja se sastoji od jednostavne forme s 2 tekstualna polja za unos korisničkog imena i lozinke, te gumba za slanje forme na obradu. Stranica za prijavu može biti samostalna ili se nalaziti unutar stranice administracija.php. Ukoliko se nalazi unutar stranice administracija.php, mora biti vidljiva samo ako korisnik nije registriran. Ako je registriran onda je vidljiva forma za administraciju.

Formu za logiranje potrebno je spajanjem na bazu provjeriti korisničke pristupne podatke pomoću tablice korisnik, te potražiti da li u bazi postoji korisnik s korisničkim imenom i lozinkom koje je uneseno u formi.

Primjer upita prema bazi:

```
SELECT * FROM users WHERE username='$user' AND password='$pass'
```

Informaciju o tome da li je upit nešto vratio možemo saznati pomoću funkcije `mysqli_num_rows()`.

Korištenje funkcije `mysqli_num_rows`:

```
if (mysqli_num_rows($result)>0) echo ('Uspjesan login');
```

Ako korisnik postoji, potrebno je provjeriti status atributa level u bazi podataka, te ukoliko se utvrdi da dotični korisnik posjeduje administratorska prava prikazuje mu se administratorski dio stranice. Ukoliko ih nema potrebno je ispisati poruku “Korisničko ime, nemate dovoljna prava za pristup ovoj stranici.“. Ukoliko korisnika nema u bazi potrebno je ispisati poruku s linkom na formu za registraciju.

Registracija korisnika

Potrebno je napraviti stranicu za registraciju naziva registracija.php koja se sastoji od forme za registraciju. Korisnik mora unijeti korisničko ime, svoje pravo ime, prezime, te dva puta lozinku. Klikom na gumb se poziva skripta koja navedeno unosi u bazu podataka ukoliko su unesene dvije lozinke identične. S obzirom da je loša praksa zbog sigurnosti pohranjivati lozinke u bazi podataka u običnom tekstu, potrebno je nad lozinkom prije upisa primijeniti jedan od algoritama izračunavanja sažetaka poruke (hash). Za potrebe ovog projekta koristit ćemo hash algoritam koristeći funkciju password_hash(). Svim korisnicima registracijom treba postaviti atribut level na vrijednost 0. Admin korisnicima treba promijeniti vrijednost tog atributa direktno u bazi podataka (na proizvoljnu vrijednost).

password_hash()

Sigurnosni aspekt konkretno u ovome slučaju je zapravo autentikacija korisnika, provjera identiteta, potvrđivanje svojeg identiteta. U aplikaciji se korisnik identificira korisničkim imenom i lozinkom. Kako bi se sačuvala privatnost i sigurnost podataka, to jest lozinke, ona se ne zapisuje u obliku kako je i unesena. Kada korisnik unese svoju lozinku ona se zapravo sprema u kombinaciji slova, brojeva i znakova. U projektu je korištena PHP funkcija password_hash().

```
$lozinka = $_POST['pass'];  
$hashed_password = password_hash($lozinka, CRYPT_BLOWFISH);
```

\$_POST je super globalna varijabla koja je polje i unutar nje se nalaze sve vrijednosti elemenata forme koje su poslale metodom POST. Vrijednosti se šalju na način da ih korisnik ne može vidjeti u tome trenutku, što također doprinosi sigurnosti. U navedenom primjeru radi se o lozinki (engl. password). PASSWORD_BCRYPT govori koji algoritam se koristi. U ovome slučaju koristiti će se CRYPT_BLOWFISH što znači da će lozinka biti spremljena koristeći "\$2y\$" znakove.

Ako nakon prijave pogledate zapis lozinke u tablici vidjet će te rezultat password_hash() funkcije. Kako bi ste prilikom prijave mogli uspotrediti unešenu lozinku sa onom u tablici trebate koristiti još jednu funkciju, a to je password_verify(). Funkcija kao parametre prima unešenu lozinku i hash iz tablice, te vraća true ili false

```

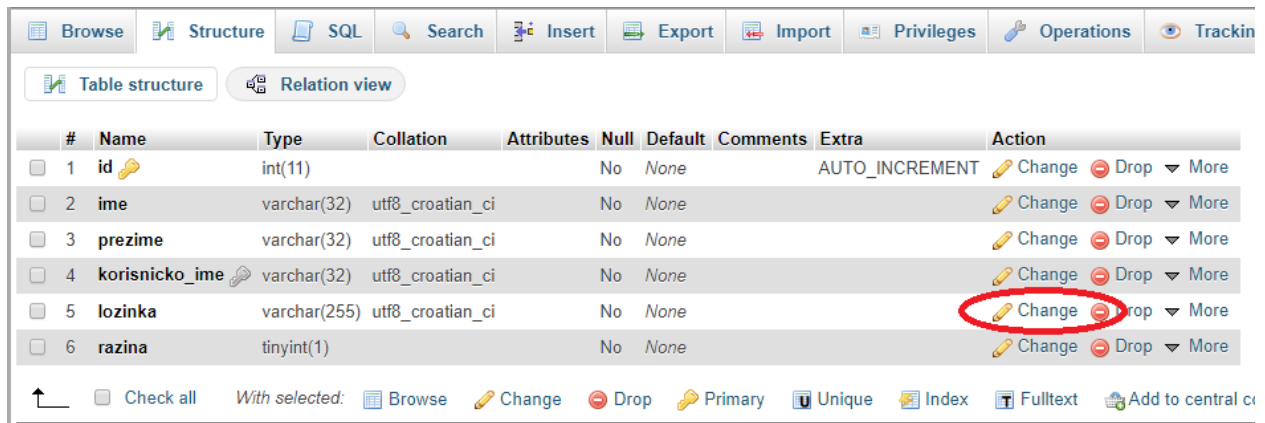
if (password_verify($_POST['lozinka'], $lozinkaKorisnika){
    $uspjesnaPrijava = true;

}else {

}

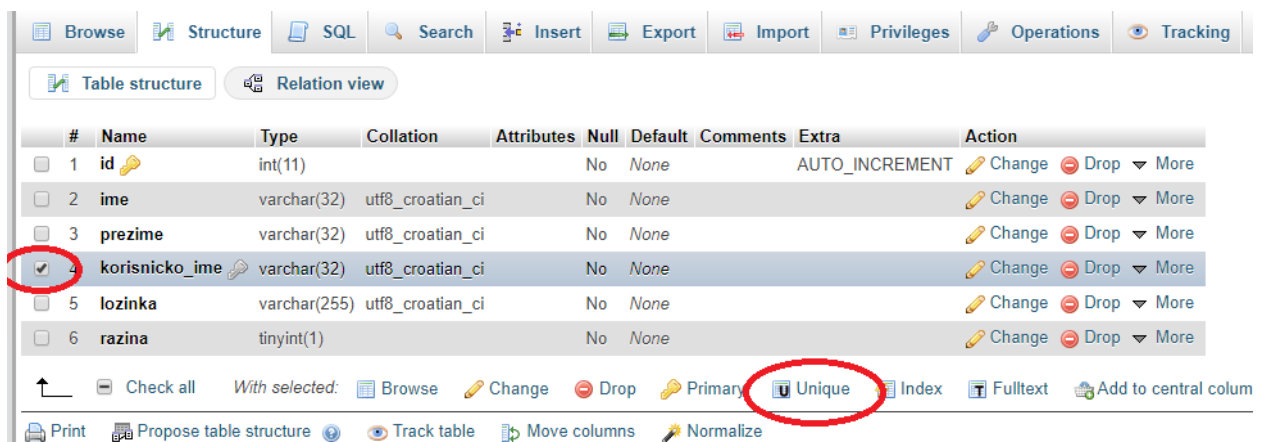
```

Kako bi mogli ispravno zapisivati vrijednosti hash funkcije potrebno je osigurati da polje u tablici koja se nalazi u vašoj bazi ima mogućnost primanja 255 znakova. Ukoliko te niste napravili prilikom kreiranja tablice, možete napraviti naknadno



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	ime	varchar(32)	utf8_croatian_ci		No	None			Change Drop More
3	prezime	varchar(32)	utf8_croatian_ci		No	None			Change Drop More
4	korisnicko_ime	varchar(32)	utf8_croatian_ci		No	None			Change Drop More
5	lozinka	varchar(255)	utf8_croatian_ci		No	None			Change Drop More
6	razina	tinyint(1)			No	None			Change Drop More

Također, dobra je praksa ograničiti ponavljanje naziva korisničkih imena, odnosno dozvoliti samo jedinstvena korisnička imena. Ukoliko to niste napravili prilikom kreiranja tablice, možete napraviti naknadno.



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	ime	varchar(32)	utf8_croatian_ci		No	None			Change Drop More
3	prezime	varchar(32)	utf8_croatian_ci		No	None			Change Drop More
4	korisnicko_ime	varchar(32)	utf8_croatian_ci		No	None			Change Drop More
5	lozinka	varchar(255)	utf8_croatian_ci		No	None			Change Drop More
6	razina	tinyint(1)			No	None			Change Drop More

Sql injecton

SQL injection napadi iskorištavaju ranjivost sučelja između poslužitelja i baze podataka

Primjer PHP koda za logiranje koji nije zaštićen od SQL injectiona:

```
<?php
$query = "SELECT * FROM users WHERE username='" .
$_POST['username'] . "' AND password='" . $_POST['password'] . "'";
$result = mysql_query($query);
$rows = mysql_num_rows($result);
If ($rows > 0) {
// do something
} else {
// do something else
}
?>
```

Ukoliko u formu na stranici za korisničko ime unesemo “korisnik“, a za password “123456“ upit će izgledati:

```
SELECT * FROM users WHERE username='korisnik' AND password='123456'
```

Primijetimo da skripta na serveru direktno ubacuje korisnikov unos u upit prema bazi podataka. Ukoliko zlonamjerni korisnik unese za username “korisnik“, a za password “OR '1'='1“, upit će izgledati:

```
SELECT * FROM users WHERE username='korisnik' AND password='' OR '1'='1'
```

Treći dio uvjeta je true, prema tome rezultat upita će biti cijela tablica users. Dva su problema s ovakvom skriptom. SQL query je string koji se slaže od korisničkog unosa nakon samog unosa pa ono što korisnik unese se može interpretirati kao SQL kod i ne postoji provjera korisničkog unosa za nedozvoljene znakove.

U PHP-u jedan od načina da riješimo taj problem je korištenje mysql prepared statementa:

```
$sql="INSERT INTO users (Username, Password) values (?, ?)";
/* Inicijalizira statement objekt nad konekcijom */
$stmt=mysqli_stmt_init($dbc);
/* Povezuje parametre statement objekt s upitom */
if (mysqli_stmt_prepare($stmt, $sql)){
    /* Povezuje parametre i njihove tipove s statement objektom */
    mysqli_stmt_bind_param($stmt,'ss',$username,$password);
    /* Izvršava pripremljeni upit */
    mysqli_stmt_execute($stmt);
}
```

Prilikom pridruživanja parametara upitu pomoću funkcije `mysqli_stmt_bind_param()`, moramo kao drugi argument funkcije navesti popis tipova parametara upita. Tip može biti: s – string, i – integer, d – decimal ili b – blob. Nakon što postavimo vrijednosti parametara pozovemo funkciju `execute` koja izvršava upit. U slučaju navedenom u primjeru parametri upita su dva stringa.

Kada nam trebaju rezultati upita (npr. kao kod `SELECT`), moramo nakon izvršavanja upita rezultate pridružiti nekoj varijabli:

```
$sql="SELECT username, password FROM users WHERE Username=? AND
Password=?";
/* Inicijalizira statement objekt nad konekcijom */
$stmt=mysqli_stmt_init($dbc);
/* Povezuje parametre statement objekt s upitom */
if (mysqli_stmt_prepare($stmt, $sql)){
    /* Povezuje parametre i njihove tipove s statement objektom */
    mysqli_stmt_bind_param($stmt,'ss',$username,$password);
    /* Izvršava pripremljeni upit i pohranjuje rezultate */
    mysqli_stmt_execute($stmt);
    mysqli_stmt_store_result($stmt);
}
/* Povezuje attribute iz rezultata s varijablama */
mysqli_stmt_bind_result($stmt, $a, $b);
/* Dohvaća redak iz rezultata, i posprema vrijednosti atributa u varijable
navedene funkcijom mysqli_stmt_bind_result() */
mysqli_stmt_fetch($stmt);
```


Nakon poziva funkcije `mysqli_stmt_fetch()` u varijablama `$a` i `$b` će biti pohranjene vrijednosti `username` i `password` atributa prvog retka rezultata. Ukoliko želimo u njih pohraniti iste attribute sljedećeg retka rezultata moramo tu funkciju ponovo pozvati.

Umjesto funkcije `mysqli_num_rows()` kada koristimo prepared statemente, podatak o broju vraćenih redaka dobivamo pomoću funkcije `mysqli_stmt_num_rows()` :

```
if (mysqli_stmt_num_rows($stmt)>0) echo ('Uspjesan login');
```

Vaš zadatak je koristeći prepared statemente modificirati vašu web aplikaciju tako da ju zaštitite od SQL injectiona.

PHP SESSION

Rukovanje sesijama je ključni koncept u PHP-u koji omogućuje da se informacije o korisniku zadrže na svim stranicama web lokacije ili aplikacije. Sesija je mehanizam za zadržavanje informacija na različitim web-stranicama radi identifikiranja korisnika prilikom kretanja web lokacijom ili aplikacijom.

HTTP protokol je protokol bez statusa, što znači da poslužitelj ne može zapamtiti određenog korisnika između više zahtjeva. Na primjer, kada pristupite web stranici, poslužitelj je samo odgovoran za pružanje sadržaja tražene stranice. Dakle, kada pristupate drugim stranicama iste web stranice, web poslužitelj tumači svaki zahtjev zasebno, kao da nisu međusobno povezani. Poslužitelj ne može znati da svaki zahtjev potječe od istog korisnika.

Sesija vam omogućuje dijeljenje informacija na različitim stranicama jedne web lokacije ili aplikacije. To poslužitelju omogućuje da zna da svi zahtjevi potječu od istog korisnika, što omogućuje web-lokaciji da prikaže informacije i postavke specifične za korisnika.

Kad god želite nositi se s varijablama sesije, morate biti sigurni da je sesija već počela. Postoji nekoliko načina na koje možete započeti sesiju u PHP-u.

Metoda koju ćete najčešće vidjeti, gdje se sesija pokreće je funkcijom `session_start()`.

```
<?php
// start a session
session_start();

// manipulate session variables
?>
```

Važno je da se funkcija `session_start` mora pozvati na početku skripte, prije nego se bilo koji izlaz pošalje u preglednik.

Sljedeći primjer skripte pokazuje kako inicijalizirati varijable sesije.

```
<?php
// start a session
session_start();

// initialize session variables
$_SESSION['logged_in_user_id'] = '1';
$_SESSION['logged_in_user_name'] = 'Tutsplus';

// access session variables
echo $_SESSION['logged_in_user_id'];
echo $_SESSION['logged_in_user_name'];
?>
```

Kao što možete vidjeti, započeli smo sesiju na početku skripte pomoću funkcije `session_start`. Nakon toga pokrenuli smo nekoliko varijabli sesije. Konačno, pristupili smo tim varijablama pomoću superglobalnog `$_SESSION`.

Kao što smo raspravljali, informacije o sesiji dijele se na zahtjeve, pa se tako varijablama sesije koje su inicijalizirane na jednoj stranici može pristupiti i sa drugih stranica, sve do isteka sesije. Općenito, sesija istječe kada je preglednik zatvoren.

Primjer registracija.php 1. dio

```
$ime = $_POST['ime'];
$prezime = $_POST['prezime'];
$username = $_POST['username'];
$lozinka = $_POST['pass'];
$hashed_password = password_hash($lozinka, CRYPT_BLOWFISH);
$razina = 0;
$registriranKorisnik = '';

//Provjera postoji li u bazi već korisnik s tim korisničkim imenom
$sql = "SELECT korisnicko_ime FROM korisnik WHERE korisnicko_ime = ?";
$stmt = mysqli_stmt_init($dbc);
if (mysqli_stmt_prepare($stmt, $sql)) {
    mysqli_stmt_bind_param($stmt, 's', $username);
    mysqli_stmt_execute($stmt);
    mysqli_stmt_store_result($stmt);
}
if(mysqli_stmt_num_rows($stmt) > 0){
    $msg='Korisničko ime već postoji!';
}else{

    // Ako ne postoji korisnik s tim korisničkim imenom - Registracija korisnika
    u bazi pazeći na SQL injection
    $sql = "INSERT INTO korisnik (ime, prezime,korisnicko_ime, lozinka,
razina)VALUES (?, ?, ?, ?, ?)";
    $stmt = mysqli_stmt_init($dbc);
    if (mysqli_stmt_prepare($stmt, $sql)) {
        mysqli_stmt_bind_param($stmt, 'ssssd', $ime, $prezime, $username,
$hashed_password, $razina);
        mysqli_stmt_execute($stmt);
        $registriranKorisnik = true;
    }
}
mysqli_close($dbc);
```

```
<?php
    //Registracija je prošla uspješno
    if($registriranKorisnik == true) {
        echo '<p>Korisnik je uspješno registriran!</p>';
    } else {
        //registracija nije protekla uspješno ili je korisnik prvi put došao na
stranicu
        ?>

        <section role="main">
            <form enctype="multipart/form-data" action="" method="POST">
                <div class="form-item">
                    <span id="porukaIme" class="bojaPoruke"></span>
                    <label for="title">Ime: </label>
                    <div class="form-field">
                        <input type="text" name="ime" id="ime" class="form-field-
textual">

                    </div>
                </div>
                <div class="form-item">
                    <span id="porukaPrezime" class="bojaPoruke"></span>
                    <label for="about">Prezime: </label>
                    <div class="form-field">
                        <input type="text" name="prezime" id="prezime" class="form-
field-textual">

                    </div>
                </div>
                <div class="form-item">
                    <span id="porukaUsername" class="bojaPoruke"></span>

                    <label for="content">Korisničko ime:</label>
                <!-- Ispis poruke nakon provjere korisničkog imena u bazi -->
                <?php echo '<br><span class="bojaPoruke">' . $msg . '</span>'; ?>
                <div class="form-field">
                    <input type="text" name="username" id="username" class="form-
field-textual">

                </div>
            </div>
            <div class="form-item">
                <span id="porukaPass" class="bojaPoruke"></span>
                <label for="pphoto">Lozinka: </label>
                <div class="form-field">
```

```

        <input type="password" name="pass" id="pass" class="form-
field-textual">
    </div>
</div>
<div class="form-item">
    <span id="porukaPassRep" class="bojaPoruke"></span>
    <label for="pphoto">Ponovite lozinku: </label>
    <div class="form-field">
        <input type="password" name="passRep" id="passRep"
class="form-field-textual">
    </div>
</div>

    <div class="form-item">
        <button type="submit" value="Prijava"
id="slanje">Prijava</button>
    </div>

</form>

</section>
<script type="text/javascript">

    document.getElementById("slanje").onclick = function(event) {

        var slanjeForme = true;

        // Ime korisnika mora biti uneseno
        var poljeIme = document.getElementById("ime");
        var ime = document.getElementById("ime").value;
        if (ime.length == 0) {
            slanjeForme = false;
            poljeIme.style.border="1px dashed red";
            document.getElementById("porukaIme").innerHTML="<br>Unesite
ime!<br>";
        } else {
            poljeIme.style.border="1px solid green";
            document.getElementById("porukaIme").innerHTML="";
        }

        // Prezime korisnika mora biti uneseno
        var poljePrezime = document.getElementById("prezime");
        var prezime = document.getElementById("prezime").value;
        if (prezime.length == 0) {
            slanjeForme = false;

```

```

        poljePrezime.style.border="1px dashed red";
document.getElementById("porukaPrezime").innerHTML="<br>Unesite Prezime!<br>";
    } else {
        poljePrezime.style.border="1px solid green";
        document.getElementById("porukaPrezime").innerHTML="";
    }

    // Korisničko ime mora biti uneseno
    var poljeUsername = document.getElementById("username");
    var username = document.getElementById("username").value;
    if (username.length == 0) {
        slanjeForme = false;
        poljeUsername.style.border="1px dashed red";

document.getElementById("porukaUsername").innerHTML="<br>Unesite korisničko
ime!<br>";
    } else {
        poljeUsername.style.border="1px solid green";
        document.getElementById("porukaUsername").innerHTML="";
    }

    // Provjera podudaranja lozinki
    var poljePass = document.getElementById("pass");
    var pass = document.getElementById("pass").value;
    var poljePassRep = document.getElementById("passRep");
    var passRep = document.getElementById("passRep").value;
    if (pass.length == 0 || passRep.length == 0 || pass != passRep) {
        slanjeForme = false;
        poljePass.style.border="1px dashed red";
        poljePassRep.style.border="1px dashed red";
        document.getElementById("porukaPass").innerHTML="<br>Lozinke
nisu iste!<br>";

document.getElementById("porukaPassRep").innerHTML="<br>Lozinke nisu iste!<br>";
    } else {
        poljePass.style.border="1px solid green";
        poljePassRep.style.border="1px solid green";
        document.getElementById("porukaPass").innerHTML="";
        document.getElementById("porukaPassRep").innerHTML="";
    }

    if (slanjeForme != true) {
        event.preventDefault();
    }

```

```
};

</script>
<?php
}

?>
```

Primjer administracija.php 1. dio

```
<?php
session_start();
include 'connect.php';

// Putanja do direktorija sa slikama
define('UPLPATH', 'img/');

// Provjera da li je korisnik došao s login forme
if (isset($_POST['prijava'])) {

    // Provjera da li korisnik postoji u bazi uz zaštitu od SQL injectiona
    $prijavaImeKorisnika = $_POST['username'];
    $prijavaLozinkaKorisnika = $_POST['lozinka'];

    $sql = "SELECT korisnicko_ime, lozinka, razina FROM korisnik
            WHERE korisnicko_ime = ?";
    $stmt = mysqli_stmt_init($dbc);
    if (mysqli_stmt_prepare($stmt, $sql)) {
        mysqli_stmt_bind_param($stmt, 's', $prijavaImeKorisnika);
        mysqli_stmt_execute($stmt);
        mysqli_stmt_store_result($stmt);
    }
    mysqli_stmt_bind_result($stmt, $imeKorisnika, $lozinkaKorisnika,
    $levelKorisnika);
    mysqli_stmt_fetch($stmt);

    //Provjera lozinke
    if (password_verify($_POST['lozinka'], $lozinkaKorisnika) &&
    mysqli_stmt_num_rows($stmt) > 0) {
        $uspjesnaPrijava = true;
```

```

        // Provjera da li je admin
        if($levelKorisnika == 1) {
            $admin = true;
        }
        else {
            $admin = false;
        }
        //postavljanje session varijabli
        $_SESSION['$username'] = $imeKorisnika;
        $_SESSION['$level'] = $levelKorisnika;
    } else {
        $uspjesnaPrijava = false;
    }
}

// Brisanje i promijena arhiviranosti

?>

```

Administracija.php 2. dio

```

<?php
    // Pokaži stranicu ukoliko je korisnik uspješno prijavljen i
    administrator je
    if (($uspjesnaPrijava == true && $admin == true) ||
(isset($_SESSION['$username'])) && $_SESSION['$level'] == 1) {

        $query = "SELECT * FROM vijesti";
        $result = mysqli_query($dbc, $query);
        while($row = mysqli_fetch_array($result)) {

            //forma za administraciju
        }
        // Pokaži poruku da je korisnik uspješno prijavljen, ali nije
        administrator
    } else if ($uspjesnaPrijava == true && $admin == false) {

        echo '<p>Bok ' . $imeKorisnika . '! Uspješno ste prijavljeni, ali
        niste administrator.</p>';

        } else if (isset($_SESSION['$username']) && $_SESSION['$level'] == 0) {

```



```
        echo '<p>Bok ' . $_SESSION['$username'] . '!! Uspješno ste
prijavljeni, ali niste administrator.</p>';
```

```
    } else if ($uspjesnaPrijava == false) {
        ?>
```

```
        <!-- Forma za prijavu -->
```

```
        <script type="text/javascript">
```

```
            //javascript validacija forme
```

```
        </script>
```

```
<?php
```

```
}
```

```
?>
```

```
?>
```