



## SERVERSKE TEHNOLOGIJE

### Vežba 8: Primer drugog kolokvijuma

U ovoj vežbi ćemo obraditi kraći PHP projekat koji zahteva log in, sign up, i web formu sa par jednostavnih komponenti. Na kolokvijumu će biti nešto veoma slično, s tim što ćete sami da razvijete bazu, stavite sve neophodne veze, i uneti desetak podataka, ili "iz glave" ili preko ChatGPT-a. Mogu da naglasim da će biti lakše od ovog primera u smislu veza između entiteta. Na primer, imaćete radnike, šefove i projekte, ne kao ovde vlasnike, veterinare, ljubimce i preglede. Ovo je za vežbu sada 😊

Što se dizajna tiče, neophodno je da koristite paletu boja sa colors.co, da imate CSS fajl sa više stilova i da sve izgleda uredno i pregledno. Za PHP možete koristiti ili PDO ili mysqli, imate slobodu pri odabiru tog pristupa. Baza mora da bude sačuvana kao .sql fajl i priložena u .zip folderu sa projektom. Lozinka mora da bude zaštićena heširanjem, to je obavezni zahtev.

Tema ovog projekta je veterinarska ambulanta, gde će korisnici moći da se registruju, uloguju i zakažu preglede za svoje ljubimce. Jedan korisnik može biti vlasnik više ljubimaca, i treba imati više doktora na raspolaganju kod kojih mogu zakazati preglede. Takođe, izuzetno je bitno da u željenom terminu izabrani veterinar već nema neki zakazan pregled. Ukoliko je to slučaj, prvi sledeći slobodni termin treba da bude 1h kasnije. Ne dozvolite zakazivanje u 12:01, ukoliko je u 12:00 željeni veterinar zauzet, svakako. Predlog za tabele u bazi je dat u nastavku:

```
CREATE TABLE Vlasnik (  
    username VARCHAR(100) PRIMARY KEY,  
    lozinka VARCHAR(255) NOT NULL -- heširana lozinka  
    ime VARCHAR(100) NOT NULL,  
    prezime VARCHAR(100) NOT NULL,  
    email VARCHAR(255) UNIQUE NOT NULL  
);  
  
CREATE TABLE Ljubimac (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    ime VARCHAR(100) NOT NULL,  
    rasa VARCHAR(100) DEFAULT NULL,  
    datum_rodjenja DATE DEFAULT NULL,  
    vlasnik_id VARCHAR(100) NOT NULL,  
    FOREIGN KEY (vlasnik_id) REFERENCES Vlasnik(username) ON DELETE CASCADE  
);  
  
CREATE TABLE Veterinar (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    ime VARCHAR(100) NOT NULL,  
    prezime VARCHAR(100) NOT NULL,  
    specijalizacija VARCHAR(255) DEFAULT NULL -- Npr. ortopedija ili hirurgija  
);
```

```

CREATE TABLE Pregled (
    id INT AUTO_INCREMENT PRIMARY KEY,
    ljubimac_id INT NOT NULL,
    veterinar_id INT NOT NULL,
    opis VARCHAR(255) NOT NULL,
    datum DATETIME NOT NULL,
    FOREIGN KEY (ljubimac_id) REFERENCES Ljubimac(id) ON DELETE CASCADE,
    FOREIGN KEY (veterinar_id) REFERENCES Veterinar(id) ON DELETE CASCADE,
    UNIQUE (veterinar_id, datum) -- Jedan veterinar može imati samo jedan
                                pregled u jednom terminu
);

```

Sledi primer unosa za bazu, za samostalnu izradu vežbe svakako treba da imate bar 5 korisnika, bar 5 veterinara, bar 7 ljubimaca i bar 10 zakazanih pregleda.

```

INSERT INTO Vlasnik (username, lozinka, ime, prezime, email) VALUES
('mmarkovic', SHA2('sifra123', 256), 'Milan', 'Marković', 'mmarkovic@mail.com'),
('ssimic', SHA2('lozinka456', 256), 'Sofija', 'Simić', 'ssimic@mail.com'),
('jpetrovic', SHA2('pass789', 256), 'Jovana', 'Petrović', 'jpetrovic@mail.com');

```

```

INSERT INTO Ljubimac (ime, rasa, datum_rodjenja, vlasnik_id) VALUES
('Bobi', 'Zlatni retriever', '2021-04-15', 'mmarkovic'),
('Maza', 'Persijska mačka', '2019-07-20', 'ssimic'),
('Luna', 'Pudlica', '2020-12-05', 'mmarkovic'),
('Cezar', 'Labrador', '2022-01-10', 'jpetrovic');

```

```

INSERT INTO Veterinar (ime, prezime, specijalizacija) VALUES
('Ivana', 'Stojković', 'Ortopedija'),
('Nikola', 'Jovanović', 'Hirurgija'),
('Milica', 'Milićević', 'Opsta veterina');

```

```

INSERT INTO Pregled (ljubimac_id, veterinar_id, opis, datum) VALUES
(1, 1, 'Rentgentski snimak', '2025-01-10 09:00:00'), -- Bobi kod Ivane Stojković
(2, 2, 'Saniranje rane', '2025-01-10 10:00:00'), -- Maza kod Nikole Jovanovića
(3, 3, 'Vakcijancija', '2025-01-11 11:00:00'), -- Luna kod Milice Milićević
(4, 1, 'Stavljanje kragne', '2025-01-11 12:00:00'); -- Cezar kod Ivane Stojković

```

Na početnoj stranici treba da imate polja za username i password, dugmeta za Log in, Sign up i Reset, ukoliko želite da poništite unose u poljima. Sign up vodi na novu stranicu gde se nalaze sva neophodna polja za registraciju novog korisnika (username, pass, ime, prezime, mejl).

Dodatno, pri registraciji korisnika možete prijaviti i nekog ljubimca, tako da na ovoj stranici treba da imate i polja za upis novog ljubimca. Nakon uspešne registracije, kada se svi SQL upiti izvrše, možete se vratiti na početnu stranicu i ulogovati se.

Pri uspešnom logovanju, u headeru se ispod banera ispisuje poruka „Dobrodošli, Nikola“, ukoliko je ime ulogovanog korisnika Nikola. Vodite računa da ovde prikažete ime, a ne username tog korisnika. Ispod toga je potrebno dodati komponente za prikaz, zakazivanje i otkazivanje pregleda, kao i dugme za log out, gde će se na klik sve sesije i konekcije „ugasiti“ i korisnik će biti vraćen na početnu stranicu kako bi se ulogovao ili registrovao sa nekim drugim nalogom.

Od slika su vam jedino potrebni baner, logo i favicon, a to ćete dobiti na kolokvijumu. Za potrebe ove vežbe možete ih generisati preko AI ili pronaći nešto na internetu. U footeru stoje vaši podaci i jedan kontakt email. U nastavku ove vežbe je dat i jedan primer rasporeda komponenti, tako da možete to koristiti kao osnovnu ideju i upravljati se prema tome u samostalnoj izradi.

- **Prikaz zakazanih pregleda** treba realizovati preko HTML tabele. Kolone su id pregleda, naziv ljubimca, ime veterinara, datum i vreme, kao i opis situacije. Podrazumeva se da ćete prikazati samo zakazane preglede za ljubimce ulogovanog korisnika, nema potrebe prikazivati i ljubimce drugih korisnika. Koristite pravilan složeni SQL query za ovo.
- **Zakazivanje pregleda** treba realizovati preko komponenti forme. Korisnik bira iz prve padajuće liste svog ljubimca, zatim iz druge padajuće liste željenog veterinara, pa onda i datum i vreme kada želi pregled da bude održan. Ukoliko željeni veterinar već ima pregled u tom terminu, obavestiti korisnika porukom o tome i zakazati sat vremena kasnije. Ovde nećete imati mnogo unosa pa su šanse minimalne da je i sat vremena kasnije zakazano. Naravno, pri kreiranju i popunjavanju baze vodite računa da vam nisu svi unosi „zgurani“ u jednom danu. Na kolokvijumu će ova funkcionalnost biti jednostavnija, neće biti toliko zahtevnih parametara za insert query. Ali možete očekivati insert i delete svakako.
- **Otkazivanje pregleda** se izvršava tako što unosite ID pregleda u tekstualno polje i klikom na OK dugme treba obrisati taj pregled iz baze i osvežiti tabelu. Naravno, otkazivanje je moguće ukoliko je taj pregled u budućnosti. Ne dozvoliti otkazivanje pregleda koji su se već dogodili. Napominjem da će i ova funkcionalnost biti jednostavnija na samom kolokvijumu. U nastavku možete videti neke primere koda:

#### Konekcija sa bazom (db.php)

```
<?php
$host = 'localhost';
$db = 'veterinarska_ambulanta';
$user = 'root';
$password = '';

try {
    $pdo = new PDO("mysql:host=$host;dbname=$db;charset=utf8", $user, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    die("Greška sa bazom podataka: " . $e->getMessage());
}
?>
```

#### session.php

```
<?php
session_start();
function isLoggedIn() {
    return isset($_SESSION['user']);
}
?>
```

#### functions.php

```
<?php
function hashPassword($password) {
    return password_hash($password, PASSWORD_BCRYPT);
}

function verifyPassword($password, $hash) {
    return password_verify($password, $hash);
}
?>
```

## Početna stranica:

```
<?php
include 'includes/db.php';
include 'includes/session.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = $_POST['username'];
    $password = $_POST['password'];

    $stmt = $pdo->prepare("SELECT * FROM Vlasnik WHERE username = :username");
    $stmt->execute(['username' => $username]);
    $user = $stmt->fetch();

    if ($user && verifyPassword($password, $user['lozinka'])) {
        $_SESSION['user'] = $user;
        header('Location: dashboard.php');
        exit;
    } else {
        $error = "Pogrešan username ili lozinka.";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Veterinarska Ambulanta</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <header>
        
    </header>
    <main>
        <form method="POST">
            <h2>Log In</h2>
            <input type="text" name="username" placeholder="Korisničko ime"
                required>
            <input type="password" name="password" placeholder="Lozinka" required>
            <button type="submit">Uloguj se</button>
            <a href="signup.php">Registracija</a>
            <button type="reset">Odustani</button>
        </form>
        <?php if (isset($error)) echo "<p>$error</p>"; ?>
    </main>
</body>
</html>
```

## Registracija novog korisnika:

```
<?php
include 'includes/db.php';
include 'includes/functions.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Podaci o vlasniku
    $username = $_POST['username'];
    $password = hashPassword($_POST['password']);
    $ime = $_POST['ime'];
    $prezime = $_POST['prezime'];
    $email = $_POST['email'];

    // Podaci o ljubimcu
    $ljubimacIme = $_POST['ljubimac_ime'];
    $ljubimacRasa = $_POST['ljubimac_rasa'];
    $ljubimacDatumRodjenja = $_POST['ljubimac_datum_rodjenja'];

    try {
        // Pokretanje transakcije
        $pdo->beginTransaction();

        // Unos podataka o vlasniku
        $stmtVlasnik = $pdo->prepare("INSERT INTO Vlasnik
                                     (username, lozinka, ime, prezime, email) VALUES
                                     (:username, :password, :ime, :prezime, :email)");
        $stmtVlasnik->execute([
            'username' => $username,
            'password' => $password,
            'ime' => $ime,
            'prezime' => $prezime,
            'email' => $email
        ]);

        // Unos podataka o ljubimcu
        $stmtLjubimac = $pdo->prepare("INSERT INTO Ljubimac
                                     (ime, rasa, datum_rodjenja, vlasnik_id) VALUES
                                     (:ime, :rasa, :datum_rodjenja, :vlasnik_id)");
        $stmtLjubimac->execute([
            'ime' => $ljubimacIme,
            'rasa' => $ljubimacRasa,
            'datum_rodjenja' => $ljubimacDatumRodjenja,
            'vlasnik_id' => $username
        ]);

        // Potvrda transakcije
        $pdo->commit();

        // Preusmeravanje na logovanje
        header('Location: index.php');
        exit;
    } catch (Exception $e) {
        // Otkazivanje transakcije u slučaju greške
    }
}
```

```

        $pdo->rollBack();
        die("Došlo je do greške: " . $e->getMessage());
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Registracija</title>
    <link rel="stylesheet" href="css/styles.css">
</head>
<body>
    <header>
        <h1>Registracija novog korisnika</h1>
    </header>
    <main>
        <form method="POST">
            <h2>Podaci o vlasniku</h2>
            <input type="text" name="username" placeholder="Korisničko ime"
                required>
            <input type="password" name="password" placeholder="Lozinka" required>
            <input type="text" name="ime" placeholder="Ime" required>
            <input type="text" name="prezime" placeholder="Prezime" required>
            <input type="email" name="email" placeholder="Email" required>

            <h2>Podaci o ljubimcu</h2>
            <input type="text" name="ljubimac_ime" placeholder="Ime ljubimca"
                required>
            <input type="text" name="ljubimac_rasa" placeholder="Rasa ljubimca">
            <input type="date" name="ljubimac_datum_rodjenja"
                placeholder="Datum rođenja">

            <button type="submit">Registruj se</button>
        </form>
    </main>
</body>
</html>

```

### Kod za dashboard.php (kada se korisnik uspešno uloguje)

```

<?php
session_start();
include 'includes/db.php';

if (!isset($_SESSION['user'])) {
    header('Location: index.php');
    exit;
}

$username = $_SESSION['user'];
$stmt = $pdo->prepare("SELECT ime FROM Vlasnik WHERE username = :username");
$stmt->execute(['username' => $username]);

```

```

$vlasnik = $stmt->fetch();
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Veterinarska Ambulanta</h1>
    <h2>Dobrodošli, <?php echo htmlspecialchars($vlasnik['ime']); ?>!</h2>
    <a href="logout.php" class="logout-btn">Odjava</a>
  </header>

  <h3>Zakazani Pregledi Vaših Ljubimaca</h3>
  <table>
  <thead>
    <tr>
      <th>ID</th>
      <th>Ime Ljubimca</th>
      <th>Veterinar</th>
      <th>Datum i Vreme</th>
      <th>Opis</th>
    </tr>
  </thead>
  <tbody>
    <?php
    $stmt = $pdo->prepare("
      SELECT p.id, l.ime AS ljubimac_ime, CONCAT(v.ime, ' ', v.prezime) AS
      veterinar, p.datum, p.opis
      FROM Pregled p
      JOIN Ljubimac l ON p.ljubimac_id = l.id
      JOIN Veterinar v ON p.veterinar_id = v.id
      WHERE l.vlasnik_id = :username
      ORDER BY p.datum ASC
    ");
    $stmt->execute(['username' => $username]);
    while ($pregled = $stmt->fetch()) {
      echo "<tr>
        <td>{$pregled['id']}</td>
        <td>" . htmlspecialchars($pregled['ljubimac_ime']) . "</td>
        <td>" . htmlspecialchars($pregled['veterinar']) . "</td>
        <td>{$pregled['datum']}</td>
        <td>" . htmlspecialchars($pregled['opis']) . "</td>
      </tr>";
    }
  </tbody>
</table>

```

```

<h3>Zakazivanje Pregleda</h3>
<form method="POST" action="schedule.php">
  <label for="ljubimac">Izaberite Ljubimca:</label>
  <select name="ljubimac_id" id="ljubimac" required>
    <?php
      $stmt = $pdo->prepare("SELECT id, ime FROM Ljubimac WHERE
                            vlasnik_id = :username");
      $stmt->execute(['username' => $username]);
      while ($ljubimac = $stmt->fetch()) {
        echo "<option value='{ $ljubimac['id'] }'>"
          . htmlspecialchars($ljubimac['ime']) . "</option>";
      }
    ?>
  </select>

  <label for="veterinar">Izaberite Veterinara:</label>
  <select name="veterinar_id" id="veterinar" required>
    <?php
      $stmt = $pdo->query("SELECT id, CONCAT(ime, ' ', prezime) AS ime
                          FROM Veterinar");
      while ($veterinar = $stmt->fetch()) {
        echo "<option value='{ $veterinar['id'] }'>"
          . htmlspecialchars($veterinar['ime']) . "</option>";
      }
    ?>
  </select>

  <label for="datum">Datum i Vreme:</label>
  <input type="datetime-local" name="datum" id="datum" required>

  <label for="opis">Opis Pregleda:</label>
  <textarea name="opis" id="opis" required></textarea>

  <button type="submit">Zakazati Pregled</button>
</form>

<h3>Otkazivanje Pregleda</h3>
<form method="POST" action="cancel.php">
  <label for="pregled_id">ID Pregleda za Otkazivanje:</label>
  <input type="number" name="pregled_id" id="pregled_id" required>

  <button type="submit">Otkazati Pregled</button>
</form>

<footer>
  <p>Kontakt: support@ambulanta.com | Vaši podaci: Ime i prezime</p>
</footer>
</body>
</html>

```

## schedule.php

```
<?php
session_start();
include 'includes/db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $ljubimac_id = $_POST['ljubimac_id'];
    $veterinar_id = $_POST['veterinar_id'];
    $datum = $_POST['datum'];
    $opis = $_POST['opis'];

    // Provera zauzetosti termina
    $stmt = $pdo->prepare("SELECT datum FROM Pregled WHERE veterinar_id =
                                :veterinar_id AND datum = :datum");
    $stmt->execute(['veterinar_id' => $veterinar_id, 'datum' => $datum]);
    if ($stmt->fetch()) {
        $datum = date('Y-m-d H:i:s', strtotime($datum . ' +1 hour'));
        echo "<p>Izabrani termin je zauzet. Sledeći slobodan termin: $datum</p>";
    }

    $stmt = $pdo->prepare("
        INSERT INTO Pregled (ljubimac_id, veterinar_id, opis, datum)
        VALUES (:ljubimac_id, :veterinar_id, :opis, :datum)
    ");
    $stmt->execute([
        'ljubimac_id' => $ljubimac_id,
        'veterinar_id' => $veterinar_id,
        'opis' => $opis,
        'datum' => $datum
    ]);
    header('Location: dashboard.php');
    exit;
}
?>
```

## logout.php

```
<?php
// Uključivanje session.php kako bi se osigurao početak sesije
include 'includes/session.php';

// Pokrećemo sesiju
session_start();

// Uništavamo sve podatke sesije
session_unset();

// Uništavamo sesiju
session_destroy();

// Preusmeravamo korisnika na početnu stranicu
header('Location: index.php');
exit;
?>
```

## cancel.php

```
<?php
session_start();
include 'includes/db.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $pregled_id = $_POST['pregled_id'];

    // Provera da li je pregled u budućnosti
    $stmt = $pdo->prepare("SELECT datum FROM Pregled WHERE id = :pregled_id");
    $stmt->execute(['pregled_id' => $pregled_id]);
    $pregled = $stmt->fetch();

    if ($pregled && strtotime($pregled['datum']) > time()) {
        $stmt = $pdo->prepare("DELETE FROM Pregled WHERE id = :pregled_id");
        $stmt->execute(['pregled_id' => $pregled_id]);
        echo "<p>Pregled je uspešno otkazan.</p>";
    } else {
        echo "<p>Nije moguće otkazati prošle preglede.</p>";
    }

    header('Location: dashboard.php');
    exit;
}
?>
```

## Predlog za raspored grafičkih komponenti:

