



## SERVERSKE TEHNOLOGIJE

### Vežba 7

## Interaktivni kviz i igra memorije u PHP-u

(radi se dve nedelje)

Ova vežba sadrži dva dela na osnovu kojih ćete biti ocenjeni na odbrani lab vežbi, neposredno pred ispit. Na kolokvijumu neće biti zadataka ovog tipa. Primeri koda su dati u nastavku, možete ih iskoristiti kao ideju i inspiraciju za svoja rešenja. Takođe, što se dizajna i tematike tiče, imate slobodu da budete kreativni, pri tome vodeći računa da svi sadržaji budu originalni i primereni za potrebe ovog predmeta.

### Interaktivni kviz u PHP-u

Kviz aplikacija u PHP-u omogućuje korisnicima da se registruju i prijave kako bi učestvovali u igri znanja. Nakon uspešne prijave, korisnik započinje kviz koji nasumično bira 10 pitanja iz baze podataka. Svako pitanje ima četiri ponuđena odgovora, od kojih je jedan tačan. Korisnik odgovara na pitanja jedno po jedno, prelazeći na sledeće klikom na dugme. Tematiku pitanja za kviz birate sami (sport, kultura, gradivo, itd), samo vodite računa da imate 30 do 50 pitanja u bazi i da su odgovori relevantni.

Na kraju kviza, aplikacija izračunava rezultat na osnovu tačnih odgovora i čuva ga u bazi podataka zajedno sa korisničkim imenom, datumom igranja i rezultatom. Rezultati svih igrača prikazuju se na "scoreboard" stranici, pružajući pregled najboljih rezultata. Aplikacija kombinuje PHP tehnologiju za backend, HTML za strukturu, CSS za stilizaciju, i MySQL za skladištenje podataka.

### Korak 1: Predlog za bazu podataka

```
CREATE DATABASE student_quiz;
```

```
USE student_quiz;
```

```
-- Tabela za korisnike (dodavanje imena, prezimena i email-a)
```

```
CREATE TABLE users (  
    username VARCHAR(50) PRIMARY KEY,  
    password VARCHAR(255) NOT NULL,  
    first_name VARCHAR(100) NOT NULL,  
    last_name VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL  
);
```

```
-- Tabela za pitanja
CREATE TABLE questions (
    id INT AUTO_INCREMENT PRIMARY KEY,
    question_text TEXT NOT NULL,
    option1 VARCHAR(255) NOT NULL,
    option2 VARCHAR(255) NOT NULL,
    option3 VARCHAR(255) NOT NULL,
    option4 VARCHAR(255) NOT NULL,
    correct_option INT NOT NULL CHECK (correct_option BETWEEN 1 AND 4)
);
```

```
-- Tabela za igre (povezana sa korisnicima)
CREATE TABLE games (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    play_date DATETIME DEFAULT CURRENT_TIMESTAMP,
    score INT NOT NULL,
    FOREIGN KEY (username) REFERENCES users(username)
);
```

## Korak 2: Predlog za registraciju korisnika (sign up)

```
<?php
session_start();

$conn = new mysqli('localhost', 'root', '', 'student_quiz');

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

    $checkUser = $conn->prepare("SELECT username FROM users WHERE username = ?");
    $checkUser->bind_param("s", $username);
    $checkUser->execute();
    $checkUser->store_result();
```

```

if ($checkUser->num_rows > 0) {
    echo "Korisničko ime već postoji.";
} else {
    $hashed = hash('sha256', $password);
    $stmt = $conn->prepare("INSERT INTO users
        (username, password, first_name, last_name, email) VALUES (?, ?, ?, ?, ?)");
    $stmt->bind_param("sssss", $username, $hashed, $first_name, $last_name, $email);
    if ($stmt->execute()) {
        echo "Registracija uspešna! <a href='login.php'>Prijavite se</a>";
    } else {
        echo "Greška pri registraciji.";
    }
}
$conn->close();
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign Up</title>
</head>
<body>
    <form action="signup.php" method="post">
        <label for="username">Korisničko ime:</label>
        <input type="text" name="username" required><br>

        <label for="password">Lozinka:</label>
        <input type="password" name="password" required><br>

        <label for="first_name">Ime:</label>
        <input type="text" name="first_name" required><br>

        <label for="last_name">Prezime:</label>
        <input type="text" name="last_name" required><br>
    </form>

```

```

        <label for="email">Email:</label>
        <input type="email" name="email" required><br>

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

```

### Korak 3: Predlog za logovanje korisnika (log in)

```

<?php
session_start();
$conn = new mysqli('localhost', 'root', '', 'student_quiz');

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = $_POST['username'];
    $password = hash('sha256', $_POST['password']);

    $stmt = $conn->prepare("SELECT username FROM users WHERE username = ? AND password = ?");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        $_SESSION['username'] = $username;
        header('Location: start_quiz.php');
    } else {
        echo "Pogrešno korisničko ime ili lozinka.";
    }
    $conn->close();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
</head>

```

```

<body>
  <form action="login.php" method="post">
    <label for="username">Korisničko ime:</label>
    <input type="text" name="username" required><br>

    <label for="password">Lozinka:</label>
    <input type="password" name="password" required><br>

    <button type="submit">Prijavite se</button>
  </form>
  <p>Nemate nalog? <a href="signup.php">Registrujte se</a></p>
</body>
</html>

```

#### **Korak 4: Predlog za početak kviza (nova igra)**

```

<?php
session_start();
if (!isset($_SESSION['username'])) {
    header('Location: login.php');
    exit();
}

$conn = new mysqli('localhost', 'root', '', 'student_quiz');

// Uzimanje nasumičnih pitanja
$questions = [];
$result = $conn->query("SELECT * FROM questions ORDER BY RAND() LIMIT 10");

while ($row = $result->fetch_assoc()) {
    $questions[] = $row;
}

$_SESSION['quiz'] = $questions;
$_SESSION['score'] = 0;
$_SESSION['current_question'] = 0;

header('Location: question.php');
?>

```

Na početnoj stranici implementirana je opcija za registraciju i prijavu korisnika. Registracija uključuje unos podataka koji se validiraju i čuvaju u bazi, dok prijava omogućava autentifikaciju korisnika pomoću sesija. Nakon uspješne prijave, korisniku se prikazuje kviz sa nasumično izabranih 10 pitanja iz baze podataka, što se postiže SQL upitom sa ORDER BY RAND().

### Korak 5: Predlog za prikazivanje pitanja

```
<?php
session_start();

if (!isset($_SESSION['quiz']) || $_SESSION['current_question'] >= count($_SESSION['quiz']))
{
    header('Location: result.php');
    exit();
}

$currentQuestion = $_SESSION['quiz'][$_SESSION['current_question']];
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Kviz</title>
</head>
<body>
    <h1>Pitanje <?= $_SESSION['current_question'] + 1 ?></h1>
    <p><?= $currentQuestion['question_text'] ?></p>
    <form action="submit_answer.php" method="post">
        <?php for ($i = 1; $i <= 4; $i++): ?>
            <label>
                <input type="radio" name="answer" value="<?= $i ?>" required>
                <?= $currentQuestion["option$i"] ?>
            </label><br>
        <?php endfor; ?>
        <button type="submit">Dalje</button>
    </form>
</body>
</html>
```

### Korak 6: Predlog za obradu odgovora i prelazak na sledeće pitanje

```
<?php
session_start();

if (!isset($_SESSION['quiz']) || $_SESSION['current_question'] >= count($_SESSION['quiz']))
{
    header('Location: result.php');
    exit();
}

$currentQuestion = $_SESSION['quiz'][$_SESSION['current_question']];
$correctAnswer = $currentQuestion['correct_option'];

$userAnswer = $_POST['answer'];

if ($userAnswer == $correctAnswer) {
    $_SESSION['score']++;
}

$_SESSION['current_question']++;
header('Location: question.php');
?>
```

### Korak 7: Predlog za prikaz rezultata

```
<?php
session_start();

$conn = new mysqli('localhost', 'root', '', 'student_quiz');

$username = $_SESSION['username'];
$score = $_SESSION['score'];

// Upisivanje rezultata u bazu
$stmt = $conn->prepare("INSERT INTO games (username, score) VALUES (?, ?)");
$stmt->bind_param("si", $username, $score);
$stmt->execute();

// Prikazivanje rezultata
echo "<h1>Vaš rezultat: $score</h1>";
```

```
// Prikazivanje scoreboard-a
$result = $conn->query("SELECT username, score, play_date FROM games
                        ORDER BY play_date DESC");

echo
"<table border='1'><tr><th>Korisnik </th>
<th>Rezultat</th><th>Datum</th></tr>";
while ($row = $result->fetch_assoc()) {
    echo
        "<tr><td>{$row['username']}</td><td>{$row['score']}</td>
        <td>{$row['play_date']}</td></tr>";
}
echo "</table>";
?>
```

### Detaljnije objašnjenje za korake 5-7

#### **Prikaz pitanja jedno po jedno:**

Kada korisnik započne kviz, sistem postavlja prvi indeks pitanja na 0 (početno pitanje) i prikazuje ga na ekranu. Ovo se postiže korišćenjem podataka o pitanjima, koji su prethodno nasumično odabrani i sačuvani u sesiji (`$_SESSION['questions']`). Svako pitanje ima svoj tekst, četiri ponuđene opcije i informaciju o tačnom odgovoru. Opcije odgovora se prikazuju kao radio dugmeta, tako da korisnik može odabrati samo jednu.

#### **Odabir tačnog odgovora:**

Korisnik bira jednu od ponuđenih opcija klikom na odgovarajuće radio dugme, a zatim klikne na dugme za prelazak na sledeće pitanje. Obrazac sa izborom korisnika šalje se PHP skripti putem POST metode, gde se korisnikov odgovor obrađuje.

#### **Praćenje trenutnog statusa kviza:**

PHP skripta koristi sesije za praćenje trenutnog stanja kviza. Na primer:

- Trenutni indeks pitanja (`$_SESSION['current_question']`) označava koje pitanje se trenutno prikazuje.
- Niz odgovora korisnika (`$_SESSION['user_answers']`) ažurira se nakon svakog odgovora. Na ovaj način, sistem pamti šta je korisnik odgovorio na svako pitanje.

#### **Obrada odgovora korisnika:**

Kada korisnik pređe na sledeće pitanje, skripta proverava da li je njegov odgovor tačan. Ovo se postiže poređenjem odabranog odgovora sa tačnim odgovorom, koji je takođe sačuvan u sesiji (preuzet iz baze podataka). Ako je odgovor tačan, rezultat korisnika se povećava.

#### **Prelazak na sledeće pitanje:**

Nakon obrade trenutnog odgovora, indeks trenutnog pitanja se povećava za 1, kao što je dato u kodu: `$_SESSION['current_question']++` i novo pitanje se prikazuje korisniku. Skripta takođe proverava da li je korisnik stigao do poslednjeg pitanja.

## Igra memorije u PHP-u

Da bismo napravili interaktivnu igru memorije u PHP-u sa gridom 4x4, gde korisnici otkrivaju slike, potrebno je integrisati PHP sa HTML-om, JavaScript-om i CSS-om za front-end. Svaka tačna kombinacija slika donosi pozitivne poene (npr. +10 poena), dok netačna donosi negativne poene (npr -2 poena). Igra koristi slike sa servera, a rezultati se mogu sačuvati u JSON ili tekstualnom fajlu. I ovde možete odabrati tematiku za slike po želji, mogu biti turističke destinacije, sportovi, automobili, hrana, i sl.

### Korak 1: Postavljanje strukture projekta

```
/memory_game  
  
  /images      // Folder sa slikama koje će biti korišćene u igri  
  /results    // Folder za JSON ili tekstualne fajlove  
  index.php   // Glavna PHP stranica koja pokreće igru  
  game.js     // JavaScript fajl koji kontroliše logiku igre  
  game.css    // Stilovi za igru
```

### Korak 2: HTML + PHP (index.php)

```
<?php  
session_start();  
$dir = 'images/';  
$images = array_diff(scandir($dir), array('.', '..')); // Učitavanje slika iz foldera  
  
// Limitiramo broj slika na 8 parova (16 slika za 4x4 grid)  
$selectedImages = array_slice($images, 0, 8);  
$allImages = array_merge($selectedImages, $selectedImages); // Dupliramo slike za parove  
shuffle($allImages); // Nasumično raspoređivanje slika  
  
// Kreiramo grid za igru  
$_SESSION['grid'] = $allImages;  
$_SESSION['score'] = 0;  
$_SESSION['flipped'] = [];  
?>  
  
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Memorijska Igra</title>  
  <link rel="stylesheet" href="game.css">  
</head>
```

```

<body>
  <h1>Igra Memorije</h1>
  <p>Trenutni skor: <span id="score"><?=$_SESSION['score'] ?></span></p>
  <div class="grid-container">
    <?php foreach ($_SESSION['grid'] as $index => $image): ?>
      <div class="card" data-index="<?=$index ?>" data-image="<?=$image ?>">
        <div class="card-inner">
          <div class="card-front"></div>
          <div class="card-back"></div>
        </div>
      </div>
    <?php endforeach; ?>
  </div>
  <script src="game.js"></script>
</body>
</html>

```

### Korak 3: Predlog za stil (game.css)

```

body {
  font-family: Arial, sans-serif;
  text-align: center;
}

.grid-container {
  display: grid;
  grid-template-columns: repeat(4, 150px);
  grid-template-rows: repeat(4, 150px);
  gap: 10px;
  justify-content: center;
  margin-top: 50px;
}

.card {
  width: 150px;
  height: 150px;
  perspective: 1000px;
}

```

```

.card-inner {
  position: relative;
  width: 100%;
  height: 100%;
  transform-style: preserve-3d;
  transition: transform 0.6s;
}

.card.flipped .card-inner {
  transform: rotateY(180deg);
}

.card-front,
.card-back {
  position: absolute;
  width: 100%;
  height: 100%;
  backface-visibility: hidden;
}

.card-front {
  background-color: #f3f3f3;
  display: flex;
  justify-content: center;
  align-items: center;
  font-size: 36px;
}

.card-back img {
  width: 100%;
  height: 100%;
  object-fit: cover;
}

```

Ovaj stil koristi CSS Grid za raspored elemenata u formi mreže, sa 4 kolone i 4 reda dimenzija 150px x 150px. **.grid-container** definiše raspored elemenata, koristeći **grid-template-columns** i **grid-template-rows** za postavljanje veličine svake ćelije, a **gap** stvara razmak između njih.

Svaka **.card** ima dimenzije 150px x 150px i koristi 3D perspektivu, omogućujući flip efekat. **.card-inner** se rotira na 180 stepeni kad se doda klasa **flipped**, što daje okretanje kartice. Unutrašnji elementi **.card-front** i **.card-back** su postavljeni apsolutno i prekrivaju celu karticu, s time što **.card-back** prikazuje sliku koja se prilagođava veličini kartice zahvaljujući **object-fit: cover**.

#### Korak 4: JavaScript logika igre (game.js)

JavaScript će kontrolisati igru: otkrivanje kartica, praćenje pogotka i promašaja, i ažuriranje rezultata.

```
let flippedCards = [];  
let matchedCards = 0;  
let score = 0;  
  
document.querySelectorAll('.card').forEach(card => {  
  card.addEventListener('click', () => {  
    if (flippedCards.length < 2 &&  
        !card.classList.contains('flipped') && !flippedCards.includes(card)) {  
      card.classList.add('flipped');  
      flippedCards.push(card);  
  
      if (flippedCards.length === 2) {  
        setTimeout(checkMatch, 1000);  
      }  
    }  
  });  
});  
  
function checkMatch() {  
  const [card1, card2] = flippedCards;  
  
  const image1 = card1.getAttribute('data-image');  
  const image2 = card2.getAttribute('data-image');  
  
  if (image1 === image2) {  
    score += 10; // Dodavanje pozitivnih poena za tačan pogodak  
    matchedCards += 2;  
  } else {  
    score -= 5; // Oduzimanje poena za netačan pogodak  
    card1.classList.remove('flipped');  
    card2.classList.remove('flipped');  
  }  
  
  document.getElementById('score').textContent = score;
```

```

flippedCards = [];
if (matchedCards === 16) {
    setTimeout(() => {
        alert(`Kraj igre! Vaš rezultat je ${score} poena.`);
        saveScore();
    }, 500);
}
}

function saveScore() {
    const xhr = new XMLHttpRequest();
    xhr.open('POST', 'save_score.php', true);
    xhr.setRequestHeader('Content-Type', 'application/json');
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
            console.log('Rezultat sačuvan');
        }
    };
    xhr.send(JSON.stringify({ score: score }));
}

```

Objašnjenje varijabli:

1. **flippedCards:** Ova promenljiva je niz koji čuva trenutno okrenute kartice. Služi da se prate kartice koje su u procesu otkrivanja, omogućujući proveru kada su tačno dve kartice okrenute.
2. **matchedCards:** Ova promenljiva broji broj uparenih kartica. Svaka dva pogođena para kartica povećavaju ovaj broj, što pomaže u detekciji kada su svi parovi otkriveni i igra može da se završi.
3. **score:** Ova promenljiva prati ukupni broj poena igrača. Poeni se dodaju za tačan pogodak (kada se kartice poklope), a oduzimaju za netačan pogodak (kada kartice nisu iste).

Kako kod funkcioniše:

- Kada igrač klikne na karticu, ona se okreće, a kartica se dodaje u niz **flippedCards**. Ako su već dve kartice okrenute, funkcija **checkMatch()** se poziva sa vremenskom odgodom od 1 sekunde.
- Funkcija **checkMatch()**: Proverava da li su dve okrenute kartice iste. Ako jesu, poeni za tačan pogodak se povećavaju za 10 i brojač uparenih kartica (**matchedCards**) se povećava za 2. Ako kartice nisu iste, oduzimaju se poeni i kartice se vraćaju u početnu poziciju (ponovo se okreću). Rezultat se ažurira u realnom vremenu na ekranu.
- Kraj igre: Kada su sve kartice uparene (**matchedCards === 16**), igra se završava i igrač dobija obaveštenje o svom rezultatu. Zatim, funkcija **saveScore()** se poziva, i ona koristi AJAX da pošalje rezultat na server, omogućujući igraču da sačuva svoj rezultat.

## Korak 5: Čuvanje rezultata u JSON fajlu (save\_score.php)

```
<?php
$data = json_decode(file_get_contents('results/scores.json'), true);
if (!$data) {
    $data = [];
}

$score = json_decode(file_get_contents('php://input'), true);
$data[] = ['score' => $score['score'], 'date' => date('Y-m-d H:i:s')];

file_put_contents('results/scores.json', json_encode($data, JSON_PRETTY_PRINT));

echo "Rezultat sačuvan!";
?>
```

Ova PHP skripta prima rezultat igre preko POST zahteva, dodaje ga u niz postojećih rezultata i čuva podatke u JSON fajlu scores.json. Ako fajl već sadrži podatke, skripta učitava te podatke, dodaje novi rezultat sa trenutnim datumom i vremenom, i zatim upisuje ažurirani niz u fajl. Na kraju, korisniku se šalje poruka da je rezultat uspešno sačuvan.

## Korisni resursi

Za interaktivni kviz:

[https://www.youtube.com/watch?v=32e\\_Ty-huw0&ab\\_channel=TimothyUnkert](https://www.youtube.com/watch?v=32e_Ty-huw0&ab_channel=TimothyUnkert)

[https://youtube.com/playlist?list=PLDIWc9AfQBfZlkdVaOOXi1tizJeNJipEx&si=rNxwCQCjhBhLs\\_NC](https://youtube.com/playlist?list=PLDIWc9AfQBfZlkdVaOOXi1tizJeNJipEx&si=rNxwCQCjhBhLs_NC)

[https://www.youtube.com/watch?v=EGHH2\\_VcCW0&ab\\_channel=ZarxBiz](https://www.youtube.com/watch?v=EGHH2_VcCW0&ab_channel=ZarxBiz)

<https://medium.com/@swisswebmiss/create-your-very-own-interactive-quiz-ae020a18f766>

<https://www.sourcecodester.com/php/16969/online-quiz-system-using-php-source-code.html>

Za igru memorije:

<https://www.geeksforgeeks.org/build-a-memory-card-game-using-html-css-and-javascript/>

[https://www.youtube.com/watch?v=xWdkt6KSirw&ab\\_channel=JavaScriptAcademy](https://www.youtube.com/watch?v=xWdkt6KSirw&ab_channel=JavaScriptAcademy)

[https://www.youtube.com/watch?v=DABkhfsBAWw&ab\\_channel=CodingNepal](https://www.youtube.com/watch?v=DABkhfsBAWw&ab_channel=CodingNepal)

<https://webdevplayground.com/2009/09/a-basic-memory-game-with-jquery-and-php/>

<https://github.com/felix-hauger/memory>