LV4 - PHP

Web programiranje

2024./2025.

1 Cilj vježbe

Cilj vježbe je izrada jednostavne web-aplikacije u kojoj studenti koriste PHP i MySQL za povezivanje s bazom podataka, unos i validaciju podataka putem web-formi, korisničku autentifikaciju, rad s CSV datotekama i ocjenjivanje slika.

Aplikacija treba omogućiti trajno spremanje podataka (za razliku od LV3, gdje su se podaci čuvali na klijentskoj strani) te implementaciju osnovnih sigurnosnih mjera (validacija, hashiranje lozinki, zaštita od **SQL injection**).

Integracija rješenja u postojeću web stranicu

U ovoj laboratorijskoj vježbi potrebno je integrirati rješenja svih zadataka (1. [a ili b] i 2. zadatak) u već postojeću web stranicu iz prethodnih laboratorijskih vježbi. To se može ostvariti na jedan od sljedećih načina:

- dodavanjem novog sadržaja unutar postojeće početne stranice `index.html`, ispod postojećih elemenata (npr. ispod tablice iz prvog zadatka) - za 1. zadatak, odnosno na galeriju.html (za 2. zadatak, ali ispod kreirane galerije iz LV1 i LV2) ili
- dodavanjem novih HTML stranica, uz obvezno povezivanje te stranice s postojećom strukturom web stranice (npr. putem navigacije).

Važno! Nije dopušteno kreirati potpuno odvojenu web stranicu za rješenja ovih zadataka. Svi funkcionalni elementi (prikaz i filtriranje podataka, interaktivne komponente i sl.) moraju biti integrirani kao dio jedinstvene web aplikacije, čime se osigurava kontinuitet u razvoju.

Na kraju, potrebno je ažurirati projekt na odabranom hosting servisu (npr. *Railway*) kako bi sve funkcionalnosti bile dostupne nakon osvježavanja stranice. Također, ažuriranu verziju projekta potrebno je postaviti na *GitHub* (u repozitorij `LV4`), a poveznica na repozitorij je dostupna na *Merlinu*. Također, izvesti sql tablicu i postaviti ju u repozitorij na GitHub-u (kao što je objašnjeno 4.4).

2 Zadaci laboratorijske vježbe

Zadatak 1. (a) Filmovi

Cilj zadatka je izrada interaktivne web-aplikacije za upravljanje virtualnom videotekom, u kojoj se svi podaci o filmovima, korisnicima i posudbama trajno pohranjuju i obrađuju na serverskoj strani korištenjem PHP-a i MySQL baze podataka.

Funkcionalnosti:

- **Sigurnu autentifikaciju korisnika:** Registraciju i prijavu korisnika s različitim ulogama (korisnik i administrator), uz sigurno pohranjivanje lozinki korištenjem hashiranja.
- **Unos i validaciju filmova:** Dinamičan unos, uređivanje i brisanje filmova putem web-formi, uz serversku validaciju podataka radi točnosti i sigurnosti (npr. provjera raspona trajanja filma, ispravan format godine). PHP Form Validation – W3Schools
- **Filtriranje i pretraživanje filmova:** Filtriranje prema žanru, godini i zemlji te sortiranje rezultata. Svi upiti izvršavaju se SQL naredbama na serverskoj strani, a rezultati se prikazuju u tabličnom prikazu. MySQL Filtering
- **Odabir i spremanje željenih filmova:** Korisnik može odabrati filmove koje želi gledati i dodati ih u osobnu videoteku ("osobna košarica"). Odabrani filmovi trajno se pohranjuju u bazu, povezani s korisničkim računom (npr. tablica `zeljeni_filmovi`).
- **Pregled i upravljanje osobnom videotekom:** Pregled, dodavanje i uklanjanje filmova iz osobnog popisa, uz trajno spremanje svih promjena na serveru.
- **Upozorenje na nisku prosječnu ocjenu:** Ako korisnik doda film čija je prosječna ocjena ispod 5.0, aplikacija pomoću PHP-a prikazuje upozorenje na stranici (npr. u crvenom okviru) s porukom poput „Ovaj film ima nisku ocjenu – jeste li sigurni da ga želite dodati?“. Opcionalno, aplikacija može poslati korisniku email obavijest koristeći funkciju `mail()` iz PHP-a. (PHP `mail()` function)
- (Opcionalno) Ocjenjivanje filmova s prikazom prosječne ocjene (npr. IMDb stil), pri čemu se ocjene pohranjuju u bazu.
- (Opcionalno) Uvoz i brisanje većih skupova filmskih podataka iz CSV ili Excel datoteka putem administratorskog sučelja.
- (Opcionalno) Zaštita od CSRF napada korištenjem sigurnosnih tokena te zaštita od SQL injectiona uporabom pripremljenih SQL naredbi (prepared statements). CSRF je kratica za Cross-Site Request Forgery – napad kojim zlonamjerna web stranica pokušava prevariti korisnika da nenamjerno izvrši neku radnju na drugoj web aplikaciji na kojoj je prijavljen.

Aplikacija koristi PHP sesije za upravljanje korisničkim stanjem, MySQL transakcije za očuvanje integriteta podataka, a korisničko sučelje je responzivno i prilagođeno različitim uređajima korištenjem vlastitog CSS-a.

Zadatak 1. (b) Vrijeme

Cilj zadatka je izrada web-aplikacije koja korisnicima omogućuje trajno spremanje i upravljanje planom izleta na temelju vremenskih podataka. Za razliku od LV3, gdje su se podaci privremeno čuvali na klijentskoj strani (npr. u LocalStorageu), ovdje se svi podaci pohranjuju i obrađuju na serveru pomoću PHP-a i MySQL-a.

Funkcionalnosti:

- **Sigurna autentifikacija korisnika:** Pristup funkcionalnostima omogućen je samo prijavljenim korisnicima, a lozinke se pohranjuju korištenjem hashiranja.
- **Unos i validacija meteoroloških zapisa:** Unos temperature, oborina, lokacije i datuma putem forme, uz serversku validaciju podataka (npr. provjera raspona temperature, ispravan format datuma). PHP Form Validation – W3Schools
- **Filtriranje i pretraživanje dana:** Filtriranje prema sezoni, lokaciji, tipu vremena, temperaturi i oborinama putem SQL upita. Rezultati se prikazuju tablično.
- **Odabir i spremanje datuma za izlet:** Odabrani datumi pohranjuju se u osobni plan povezan s korisničkim računom (npr. tablica `planirani_izleti`).
- **Pregled i upravljanje planom izleta:** Pregled, dodavanje i uklanjanje planiranih datuma, uz trajno spremanje promjena.
- **Upozorenja na ekstremne uvjete:** Ako korisnik pokuša planirati izlet za dan s ekstremnim vremenskim uvjetima (npr. temperatura > 35 °C), aplikacija prikazuje upozorenje ili šalje email obavijest. PHP `mail()` function
- (Opcionalno) Uvoz meteoroloških podataka iz CSV-a s automatskom konverzijom datuma pomoću PHP `DateTime` klase.

Ovaj sustav omogućuje dugoročno planiranje izleta temeljem stvarnih vremenskih podataka. Za razliku od prethodne vježbe, podaci se pohranjuju trajno, dostupni su svim korisnicima nakon prijave i podržavaju višekorisničko korištenje.

Zadatak 2 – Ocjenjivanje fotografija

Cilj zadatka je izraditi sustav za ocjenjivanje slika koji korisnicima omogućuje pregledavanje, ocjenjivanje i upravljanje slikama uz trajno spremanje podataka u bazu. Poseban naglasak stavlja se na integraciju korisničkog sučelja s ranije izrađenim CSS-om i trajnu pohranu ocjena u MySQL bazu.

Funkcionalnosti

- **Autentifikacija korisnika:** Korisnici se prijavljuju pomoću korisničkog imena i lozinke koja je hashirana u bazi. Ocjenjivanje i unos slika dostupni su samo prijavljenim korisnicima.
- **Učitavanje slika:**
  - Slike se mogu automatski povući iz unaprijed definiranog foldera na serveru (npr. `/slike/`), a moguće je i dohvaćanje slika preko vanjskog API-a.
  - Aplikacija prikazuje slike u rasporedu koristeći prethodno izrađen responzivni CSS (LV1).
- **Validacija formata i veličine:** Ako korisnik dodaje nove slike, potrebno je provjeriti da su u JPEG/PNG formatu te da ne prelaze 5MB po slici.
- **Obvezno! Ocjenjivanje slika:**
  - Korisnici mogu ocijeniti svaku sliku ocjenom od 1 do 5 zvjezdica.
  - Svaka ocjena se trajno sprema u bazu podataka zajedno s ID-om korisnika i ID-om slike (npr. tablica `ocjene` s poljima `id_korisnik`, `id_slika`, `ocjena`, `vrijeme_ocjene`).
  - Ako korisnik ponovno ocijeni istu sliku, ocjena se ažurira.
- **Prikaz prosječne ocjene:** Ispod svake slike prikazuje se prosječna ocjena izračunata na temelju svih korisničkih ocjena (npr. stilom sličnim IMDb-u).
- **(Opcionalno) Izvoz i uvoz ocjena:**
  - Korisnici mogu izvesti svoje ocjene u CSV format (npr. `moje_ocjene.csv`).
  - Administrator može uvesti ocjene iz CSV datoteke radi inicijalne postave baze ili testiranja.
- **(Opcionalno) Komentari uz slike:** Uz ocjenu, korisnik može unijeti i kratak komentar koji se prikazuje ispod slike.

Tehničke napomene

- Za prikaz zvjezdica može se koristiti CSS (npr. `::before` i `::after` selektori) ili JavaScript biblioteka, ali sustav mora funkcionirati i s osnovnim HTML/CSS-om.
- Podaci o slikama mogu se pohraniti u tablici `slike` s atributima poput `id`, `naziv_datoteke`, `opis`, `putanja`, `izvor` (lokalno/API).

- Pristup podacima mora koristiti pripremljene SQL upite radi zaštite od SQL injekcija.

Zajednički zahtjevi za sve zadatke

- Korištenje PHP-a za serversku obradu i rad s bazom podataka
- Rad sa sesijama, autentifikacijom i korisničkim stanjima
- Učitavanje i prikaz dinamičkog sadržaja
- Validacija korisničkih unosa i upravljanje datotekama
- Trajna pohrana korisničkih podataka i analiza rezultata (prosječna ocjena)
- Responzivni dizajn uz korištenje vlastitog CSS-a

3 Kako raditi s PHP-om, MySQL-om, JavaScriptom, HTML-om i CSS-om koristeći XAMPP

1. Preuzmite i instalirajte XAMPP sa službene web stranice.
2. Pokrenite XAMPP i pokrenite usluge Apache i MySQL.
3. Napravite novu mapu za svoj projekt u mapi htdocs koja se nalazi u instalacijskom direktoriju XAMPP-a.

Da biste zajedno koristili PHP, MySQL i HTML, možete slijediti korake u nastavku:

1. Napravite HTML datoteku u mapi svog projekta i dodajte svoj HTML i CSS kod.
2. Napravite JavaScript datoteku u mapi svog projekta i dodajte svoj JavaScript kod.
3. Napravite PHP datoteku u mapi svog projekta i dodajte svoj PHP kod.
4. U svojoj PHP datoteci povežite se s MySQL bazom podataka pomoću funkcije `mysqli_connect()`.
5. Napišite SQL upite za interakciju s vašom MySQL bazom podataka pomoću funkcije `mysqli_query()`.
6. Koristite PHP za generiranje dinamičkog sadržaja na vašoj web stranici postavljanjem upita vašoj MySQL bazi podataka i ispisivanjem rezultata.

4 Tips & Tricks

4.1 Prijedlog strukture datoteka

Web aplikacija za upravljanje `košaricom` strukturirana je u zasebne funkcionalne module. Aplikacija koristi PHP za povezivanje s MySQL bazom podataka, serversku obradu forme, validaciju unosa, sesije za praćenje stanja košarice te osnovne sigurnosne mehanizme.

U nastavku je generalni primjer strkture košarice.

- `functions.php` – Sadrži sve pomoćne funkcije potrebne za rad sustava:
  - povezivanje s MySQL bazom podataka (npr. preko PDO s prepared statements)
  - generiranje zajedničkih elemenata stranica (npr. zaglavlje, podnožje)
  - funkcije za dohvat i manipulaciju podacima iz baze (proizvodi, košarica)
- `index.php` – Glavna ulazna točka aplikacije. Uključuje osnovnu strukturu stranice, upravljanje sesijama i usmjeravanje korisnika na podstranice (npr. `home.php`, `products.php`).
- `home.php` – Početna stranica koja prikazuje istaknutu sliku i četiri najnovije dodana proizvoda iz baze.
- `products.php` – Prikaz svih dostupnih proizvoda s osnovnom paginacijom i filtriranjem po kategorijama. Dohvat podataka vrši se preko SQL upita s validacijom ulaznih parametara.
- `product.php` – Prikaz detalja pojedinog proizvoda temeljen na `GET` parametru. Sadrži obrazac za odabir količine i dodavanje proizvoda u košaricu. Podaci se validiraju na serverskoj strani prije spremanja u sesiju.
- `cart.php` – Prikaz sadržaja korisničke košarice. Sadrži:
  - tablični prikaz proizvoda (naziv, količina, cijena)
  - mogućnost ažuriranja količina
  - izračun ukupne cijene i međuzbroja
  - validaciju svih unosa pri izmjeni košarice
- `placeorder.php` – Završni korak kupovine. Prikazuje sažetak narudžbe i potvrdu korisniku. (Opcionalno: slanje potvrde emailom pomoću PHP `mail()` funkcije.)
- `admin.php` – Osnovna autentifikacija administratora putem forme (s hashiranjem lozinki). Pristup administracijskom sučelju dostupan je samo prijavljenim administratorima.
- `dashboard.php` – Administracijsko sučelje za upravljanje proizvodima: (opcionalno, ovisno o vrsti stranice)
  - dodavanje, uređivanje i brisanje proizvoda

  - prikaz narudžbi i statistike
  - unos podataka putem validiranih web formi

- `style.css` – Stilovi stranica, uključujući layout košarice, proizvoda, gumba i formulara. Potrebno prilagoditi prethodno razvijen CSS iz LV1.
- `imgs/` – Mapa sa slikama proizvoda, istaknutim bannerima i korisničkim ikonama. Slike se povezuju s bazom putem relativnih putanja.

Napomena: Sustav koristi PHP sesije za upravljanje stanjem korisnika i sadržajem košarice. Prilikom dodavanja i obrađivanja podataka koristi se validacija i zaštita od SQL injekcija korištenjem pripremljenih SQL upita (prepared statements).

Prilagodba strukture datoteka prema temama iz predloška

Prijedlog strukture datoteka za sustav košarice može se koristiti kao osnova za sve tri teme zadataka (filmovi, vrijeme, slike), uz manje prilagodbe imena datoteka i funkcionalnosti:

Za temu Filmovi (Videoteka)

- `products.php` postaje `films.php` – prikaz svih dostupnih filmova.
- `product.php` postaje `film.php` – prikaz detalja pojedinog filma.
- Umjesto „košarice“, koristi se pojam „moja videoteka“ (npr. tablica `zeljeni_filmovi`).
- `dashboard.php` se koristi kao administracijsko sučelje za dodavanje, uređivanje i brisanje filmova.

Za temu Vrijeme (Planiranje izleta)

- `products.php` → `days.php` – prikaz vremenskih zapisa (dani, lokacije, uvjeti).
- `product.php` → `day.php` – detalji vremenskog zapisa.
- `cart.php` → `plan.php` – prikaz korisničkog plana izleta.
- `dashboard.php` – omogućuje administratoru uvoz vremenskih podataka iz CSV datoteka.
- Ne koristi se količina, ali korisnik može dodavati dane u svoj plan.

Za temu Slike (Ocjenjivanje fotografija)

- `products.php` → `gallery.php` – prikaz svih dostupnih slika.
- `product.php` → `photo.php` – detaljni prikaz slike s mogućnošću ocjenjivanja.
- `cart.php` → `myratings.php` – prikaz korisničkih ocjena (može biti i opcionalan).
- `dashboard.php` – sučelje za administratora za upload slika, pregled i brisanje ocjena.

Zajednička preporuka za sve teme

Preporučuje se izraditi mapu `includes/` s pomoćnim datotekama koje se učitavaju prema potrebi:

- `db.php` – konekcija na bazu podataka.
- `auth.php` – provjera prijave korisnika i kontrola pristupa.
- `csrf.php` – implementacija zaštite od CSRF napada (ukoliko se koristi naprednija zaštita forma).

4.2 Pokrenuti web poslužitelj i stvoriti datoteke i direktorije koje ćete koristiti za sustav košarice.

- - Otvorite *XAMPP Control Panel*
- Pored Apache modula kliknite *Start*
- Pored MySQL modula kliknite *Start*
- Dođite do XAMPPs instalacijskog direktorija
- Otvorite direktorij `htdocs`
- Stvorite potrebne direktorije i datoteke

Sljedećim kodom povezuje se s MySQL bazom podataka, izvršava upit za odabir svih redaka iz tablice i ispisuje rezultate kao HTML tablicu. Možete izmijeniti HTML kôd kako biste formatirali tablicu kako želite.

```
<?php
// Connect to the database
$conn = mysqli_connect("localhost", "username", "password", "
   database_name");

// Check connection
if (!$conn) {
      die("Connection␣failed:␣" . mysqli_connect_error());
}

// Execute a query to select all rows from a table
$sql = "SELECT␣*␣FROM␣table_name";
$result = mysqli_query($conn, $sql);

// Create an HTML table to display the results
echo "<table>";
echo "<tr><th>ID</th><th>Name</th><th>Email</th></tr>";
while($row = mysqli_fetch_assoc($result)) {
      echo "<tr><td>" . $row["id"] . "</td><td>" . $row["name"] . "
         </td><td>" . $row["email"] . "</td></tr>";
}
echo "</table>";
```

```
// Close the database connection
mysqli_close($conn);
?>
```

Izrada baze

```
CREATE DATABASE ime_baze
```

Izrada tablice

```
CREATE TABLE IF NOT EXISTS `products` (
`id` int(10) NOT NULL AUTO_INCREMENT,
`name` varchar(250) NOT NULL,
`code` varchar(100) NOT NULL,
`price` double(9,2) NOT NULL,
`image` varchar(250) NOT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `code` (`code`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Kreiranje pozivanaj na bazu

(**u dokumentu **db.php**)

```
// Enter your Host, username, password, database below.
$con = mysqli_connect("localhost","root","","ime_baze");
if (mysqli_connect_errno()){
      echo "Failed to connect to MySQL: " . mysqli_connect_error();
      die();
}
```

4.3 Stvaranje `index.php`

Dodavanje vrijednosti odabranog proizvoda u niz kako bismo ih mogli prikazati na stranici cart.php.

```
<?php
session_start();
include('db.php'); //povezivanje s bazom
$status="";
if (isset($_POST['code']) && $_POST['code']!=""){
      $code = $_POST['code'];
      $result = mysqli_query(
      $con,
      "SELECT * FROM `products` WHERE `code`='$code'"
      );
      $row = mysqli_fetch_assoc($result);
      $name = $row['name'];
      $code = $row['code'];
      $price = $row['price'];
      $image = $row['image'];
```

```
      $cartArray = array(
      $code=>array(
      'name'=>$name,
      'code'=>$code,
      'price'=>$price,
      'quantity'=>1,
      'image'=>$image)
      );

      if(empty($_SESSION["shopping_cart"])) {
            $_SESSION["shopping_cart"] = $cartArray;
            $status = "<div class='box'>Product is added to your
               cart!</div>";
      }else{
            $array_keys = array_keys($_SESSION["shopping_cart"]);
            if(in_array($code,$array_keys)) {
                  $status = "<div class='box' style='color:red;'>
                        Product is already added to your cart!</div>
   ";
            } else {
                  $_SESSION["shopping_cart"] = array_merge(
                  $_SESSION["shopping_cart"],
                  $cartArray
                  );
                  $status = "<div class='box'>Product is added to
                     your cart!</div>";
            }

      }
}
?>
```

Dodavanje ikone košarice

Dodati sljedeću skriptu u istu datoteku u odjeljku tijela za prikaz ikone košarice.

```
<?php
if(!empty($_SESSION["shopping_cart"])) {
      $cart_count = count(array_keys($_SESSION["shopping_cart"]));
      ?><div class="cart_div"><a href="cart.php"><img src="cart-icon
         .png" /> Cart<span><?php echo $cart_count; ?></span></a></
         div><?php
}
?>
```

4.4 Pripremiti svoju bazu podataka za dijeljenje

Izvoz baze podataka:

1. Trebate izvesti svoju bazu podataka u SQL formatu. To možete učiniti koristeći phpMyAdmin koji dolazi s XAMPP-om:

2. Otvoriti phpMyAdmin (‘http://localhost/phpmyadmin‘).
3. Odabrati bazu podataka koju žele izvesti.
4. Ići na tab "Export".
5. Odabrati opciju "Quick" ili "Custom" ovisno o potrebi za specifičnim postavkama.
6. Format datoteke treba biti SQL.
7. Kliknuti na "Go" za preuzimanje .sql datoteke.

Dijeljenje SQL datoteke:

Izvezeni .sql file postaviti u svoj GitHub repozitorij zadatka, kao i sve druge dokumente potrebne za testiranje Vašeg zadatka.