Univerzitet Singidunum

Tehnički fakultet

Online aukcija

- Projektni rad -

Predmet: **Veb programiranje**

Profesor:

doc. dr Nebojša Bačanin Džakula

Asistent:

mast. Dušan Marković

Student:

Bojan Sovtić 2016/203529

Beograd, 2019. godine

1. Uvod

U ovoj dokumentaciji je prikazana funkcionalnost veb aplikacije “Online aukcija”, kao i tehnologije koje su korišćene prilikom njene izrade. Aplikacija je zamišljena kao interaktivni sajt za kupoprodaju predmeta različitih kategorija. Bazira se na sistemu aukcija, tj. korisnik koji ponudi najviše, kupio je predmet.

Sajtu može da pristupi svako i da pogleda dostupne aukcije, ali samo registrovani korisnici mogu da postavljaju aukcije i da daju svoje ponude. Korisnicima je takođe ostavljena mogućnost da postave (prilikom kreiranja ili naknadno) otkupnu cenu, tj. cenu za koju neki drugi korisnik može odmah kupiti određeni predmet, bez čekanja da se sama licitacija završi. Korisnik sam bira dokle traje aukcija i početnu cenu. Prilikom registracije korisnik je dužan da dostavi račun banke i odabere način plaćanja, dok je prilikom postavljanja aukcija obavezno i slanje slike predmeta.

U aplikaciji su postavljena određena ograničenja radi sprečavanja manipulisanja oko ponuda, kao što je ograničenje da ponuda mora da bude barem za 100$ veća od prethodne. Najvažnije ograničenje je da korisnik ne može da daje ponudu na svoje aukcije i tako im podiže trenutnu vrednost.

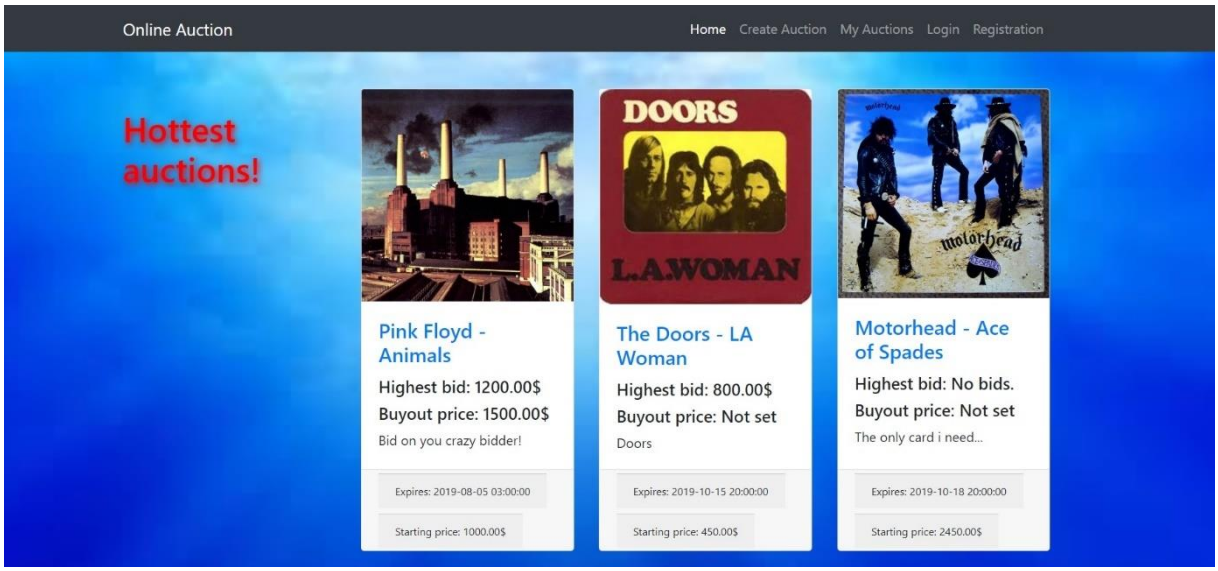
Sam projekat je nastao kao logični nastavak projekta kreiranog za Oracle kurs (Oracle Database Design/Database Programming with SQL) rađenog na Oracle APEX platformi (videti dodatnu dokumentaciju). U projektu su korišćene navedene tehnologije i softver:

- Oracle SQL Developer Data Modeler
- Navicat Premium 12
- NetBeans IDE 8.2
- XAMPP platforma sa Apache i MariaDB serverom
- HTML, CSS, JavaScript, jQuery, Ajax, PHP

2. Opis aplikacije

1. Uvod

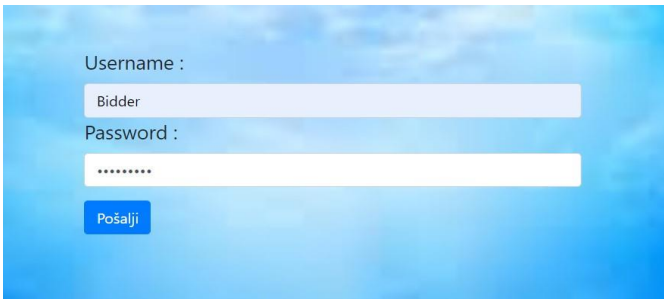
Prilikom inicijalne posete sajtu korisniku su prikazane trenutno dostupne aukcije sortirane prema vremenu isticanja (one koje ističu najskorije su prikazane prve). Korisnik može samo da pregleda aukcije, za sve ostalo je potrebna registracija.



Slika 1 Prikaz dela početne stranice

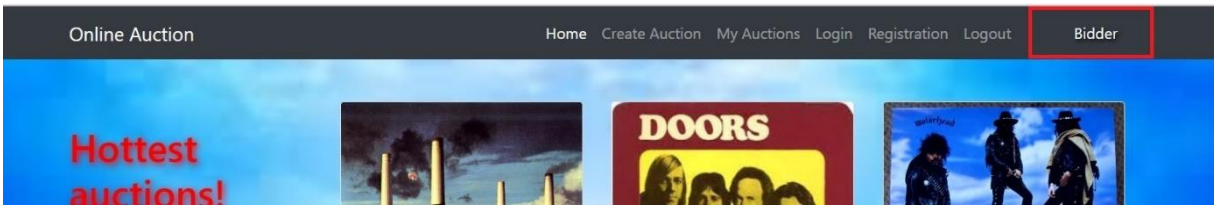
2. Registracija i logovanje

Prilikom registracije korisnik je u obavezi da dostavi sve prikazane podatke osim adrese (kućne) i broja telefona. Takođe, korisnik mora izabrati jedinstveno korisničko ime i šifru (ona koja nisu već u upotrebi). Nakon toga korisnik može da pristupi login formi datoj na slici:



Slika 2 Login forma

Važno je napomenuti da se sve korisničke šifre čuvaju u šifrovanom formatu. Podatak da li je korisnik logovan se čuva unutar korisničke sesije koja služi za proveru autentikacije korisnika i pristup određenim stranicama sajta. Podatak o korisničkom imenu se upisuje i u kolačić (eng. cookie), dok se korisničko ime (nakon uspešnog logovanja) prikazuje na vrhu stranice.

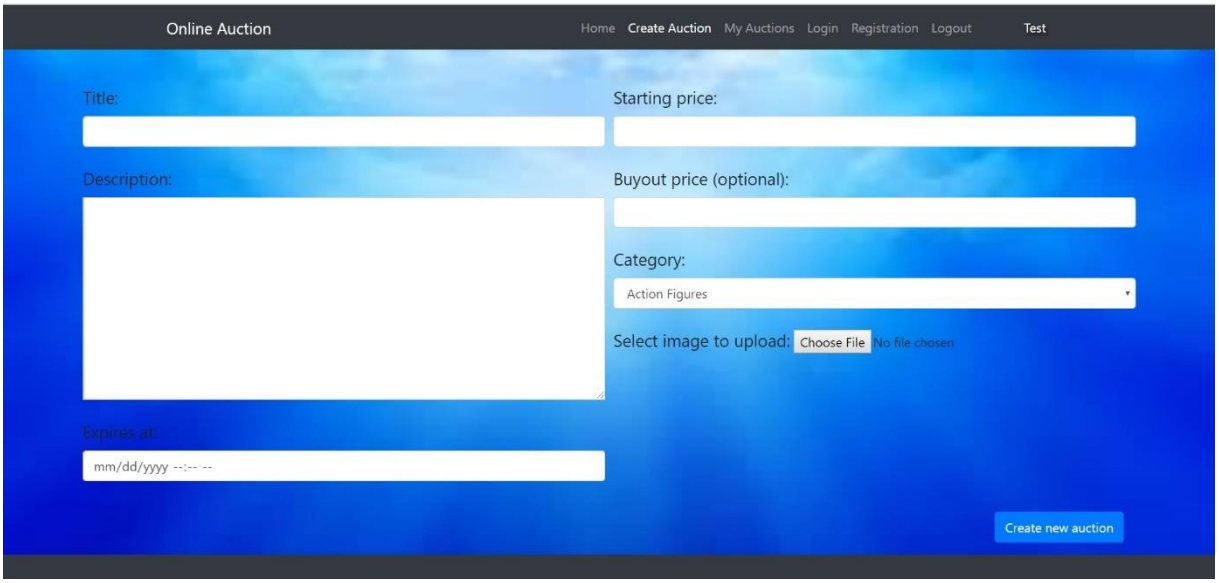


Slika 3 Prikaz dela početne strane nakon uspešnog logovanja

Nakon navedenih koraka, korisnik može da daje ponude, kao i da pristupi stranicama 'Create Auction' i 'My Auctions'.

3. Kreiranje aukcije

Na narednoj slici je prikazana forma za kreiranje aukcije:



Slika 4 Forma za kreiranje aukcije

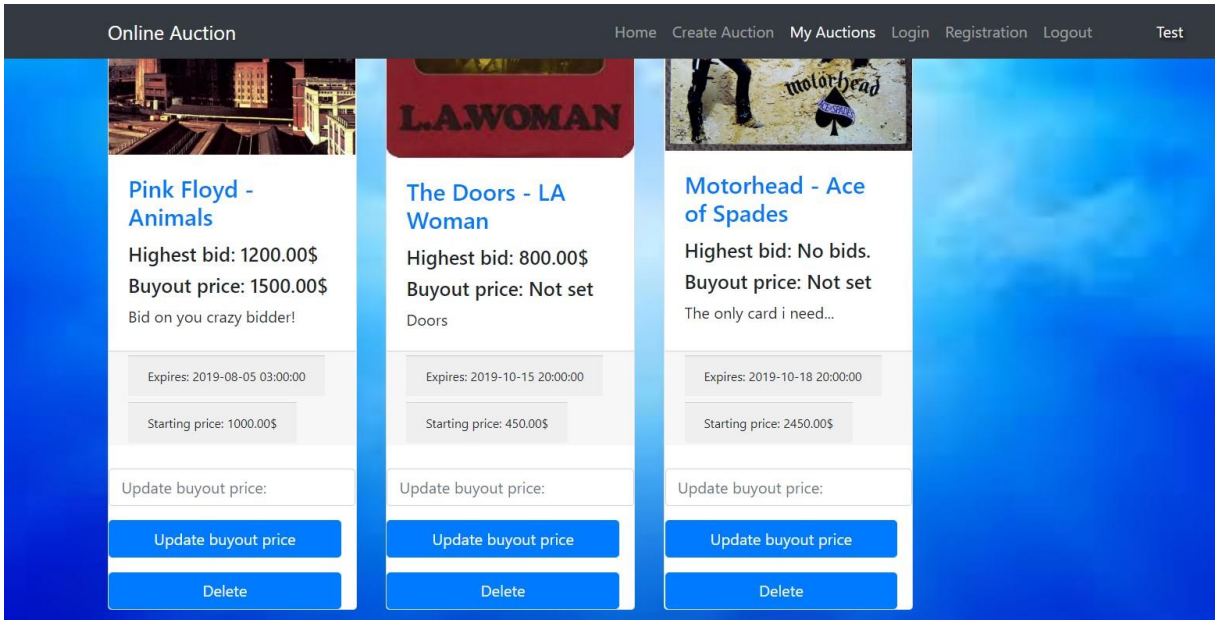
Korisnik je u obavezi da unese naslov, kratak opis, vreme isticanja aukcije i početnu cenu. Takođe, mora izabrati jednu od ponuđenih kategorija i dostaviti sliku za navedenu aukciju. Slika se dostavlja u okviru odgovarajućeg fajla. Prilikom ovog postupka treba voditi pažnju na nekoliko ograničenja:

- Početna cena ne sme biti manja od 100$

- Vreme završetka aukcije mora biti postavljeno na najmanje 30 dana od trenutka kreiranja
- Slika mora biti u nekom od podržanih formata (jpg, png, jpeg, gif)
- Veličina slike ne sme biti veća od 5MB

4. Pregled svojih aukcija i modifikovanje

Na sledećoj slici je prikazan deo aukcija postavljen od strane korisnika sa korisničkim imenom “Test”:

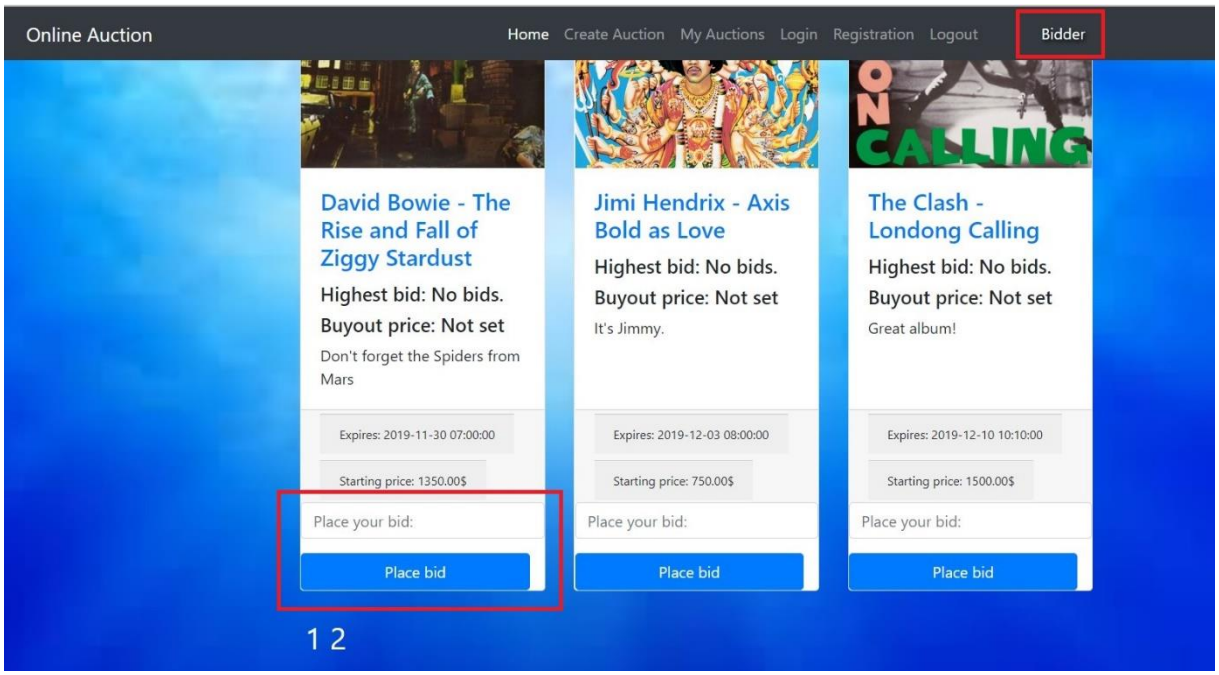


Slika 5 Prikaz aukcija korisnika

Korisnik ima uvid u najveće ponude za svaku od svojih aukcija, kao i o osnovnim informacijama koje je uneo. Ima mogućnost i da obriše aukciju kao i da postavi, tj. promeni otkupnu cenu. Korisnik ne može da promeni početnu cenu, niti vreme završetka aukcije.

5. Postavljanje ponuda

Nakon logovanja korisnika, na početnoj stranici sa aukcijama se pojavljuje opcija za unošenje ponuda, prikazano na slici:



Slika 6 Postavljanje ponuda

Korisnik može da postavi ponudu pod uslovom da je barem za 100$ veća od prethodne. Ako korisnik postavi ponudu koja se poklapa sa otkupnom cenom, smatra se da je aukcija kupljena, korisnik dobija obaveštenje, dok se sama aukcija briše iz baze podataka. Korisnik ne može postaviti ponudu na svoje aukcije.

6. Prekidanje sesije

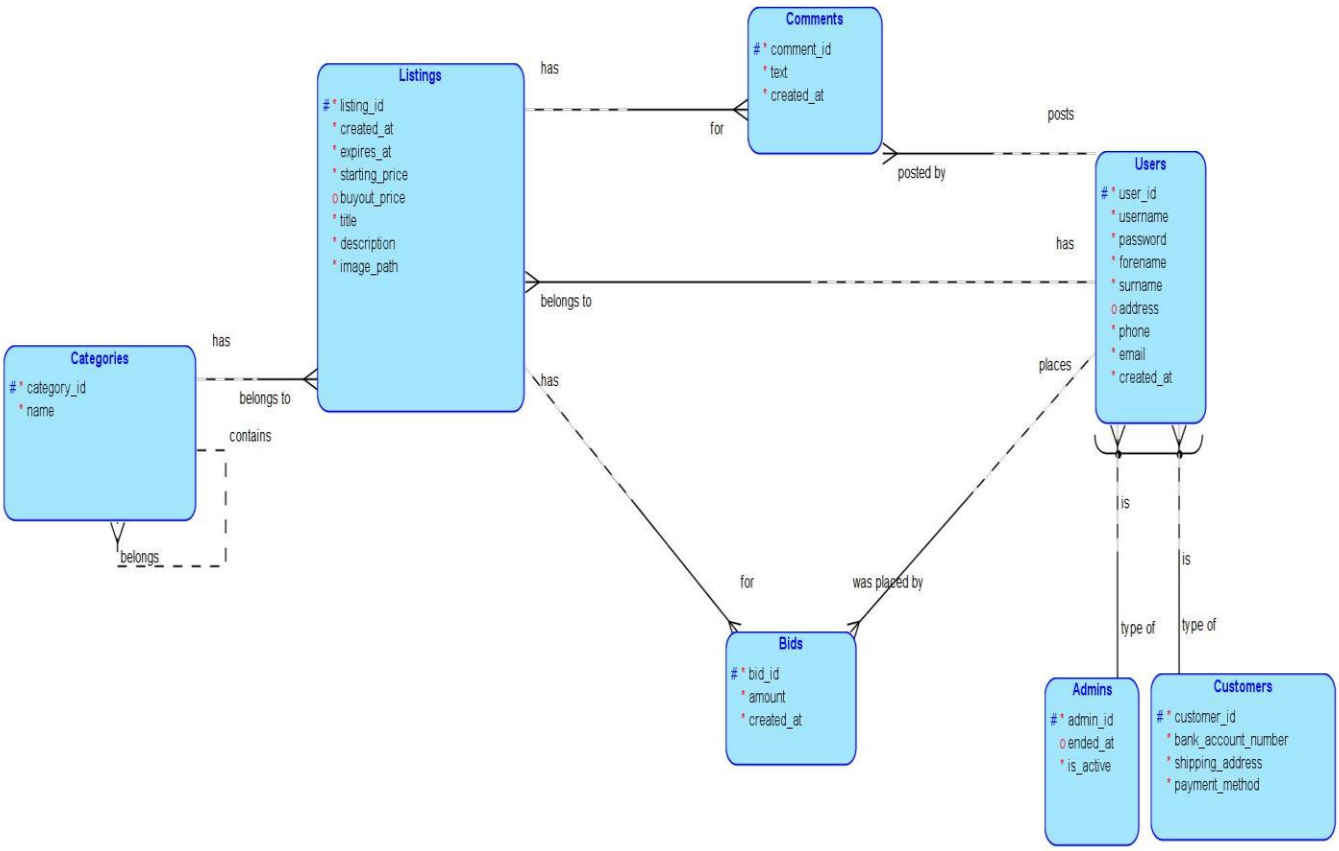
Korisnik prekida sesiju prostim klikom na ‘Logout’ dugme koje se nalazi u navigacionom meniju.

3. Prikaz tehnologije

U ovom delu će biti prikazana sama implementacija korišćenih tehnologija.

1. Baza podataka

Na sledećoj slici je prikazan ERD dijagram baze podataka, urađen pomoću Oracle SQL Developer Data Modeler-a:



Slika 7 ERD dijagram

Ovde će biti napomenute samo neke specifičnosti u vezi sa dijagramom, tj. vezama između entiteta. Kod entiteta ‘Categories’ postoji rekurzivna veza koja omogućava postavljanje potkategorija, tj. prikaz kategorije sa svim njenim potkategorijama kao na sledećoj slici:

Categories
Collectibles
__Antiques
__Art
____Paintings
____Ceramics
____Photogeraphy
__Coins & Paper

Slika 8 Prikaz svih potkategorija počevši od kategorije 'Collectibles'

Upit je dat na sledećoj slici:

```
SELECT LPAD(categories.name, LENGTH(categories.name) +
    (LEVEL * 2) - 2, '_') AS "Categories"
FROM categories
START WITH categories.name = 'Collectibles'
CONNECT BY PRIOR categories.category_id = categories.parent_id;
```

Slika 9 Oracle hijerarhijski upit

Važno je napomenuti da je ‘CONNECT BY PRIOR’ ekstenzija u vlasništvu Oracle korporacije, tako da navedeni upit neće raditi na svim tipovima relacionih baza.

Na ERD dijagramu je prikazana i arc veza između tabele ‘Users’ i tabela ‘Admins’ i ‘Customers’. Arc veza predstavjla isključivu ILI (XOR) vezu, što znači da korisnik može biti ili ‘Admin’ ili ‘Customer’. Ovo je realizovano postavljanjem oba strana ključa u tabeli ‘Users’ kao opciona i ubacivanjem sledeće provere (trigger-a):

```
BEGIN
  IF NEW.admin_id IS NULL && NEW.customer_id IS NULL THEN
      SIGNAL SQLSTATE '50004' SET MESSAGE_TEXT = "User must be admin or customer.";
  END IF;
END
```

Slika 10 Trigger koji omogućava implementaciju arc veze

Na sledećoj slici su prikazani trigger-i koji se odnose na tabelu ‘Listings’:

```
BEGIN
  IF NEW.starting_price < 100 THEN
      SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = "Starting price must be equal or above 100.";
  END IF;

  IF NEW.expires_at < date_add(NOW(), INTERVAL 30 DAY) THEN
      SIGNAL SQLSTATE '50003' SET MESSAGE_TEXT = "Expire date must be 30 or more days later.";
    END IF;

END
```

Slika 11 Trigger-i tabele 'Listings'

Navedeni triggeri se aktiviraju pre kreiranja same aukcije i omogućavaju implementaciju proceduralnih pravila. Prvi osigurava da početna cena ne sme biti manja od 100$, dok drugi obezbeđuje da kraj aukcije mora bizi postavljen barem 30 dana od trenutka postavljanja (posmatra se trenutno vreme servera uz pomoć funkcije **NOW()**).

2. PHP

2. 1. Klasa za konekciju sa bazom podataka

Za konekciju sa bazom podataka koristi se posebna omotačka (wrapper) klasa **db** (db.php). Na sledećim slikama su date neke od funkcija klasa.

```
public function __construct($dbhost = 'localhost', $dbuser = 'root', $dbpass = '', $dbname = '', $charset = 'utf8') {
        $this->connection = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
        if ($this->connection->connect_error) {
                die('Failed to connect to MySQL - ' . $this->connection->connect_error);
        }
        $this->connection->set_charset($charset);
}
```

Slika 12 Konstruktor klase 'db'

Konstruktor sadrži podrazumevane vrednosti za sve potrebne parametre, kao što su naziv host-a, korisničko ime korisnika baze, šifra, ime baze (prazan string u konkretnom primeru) i skup karaktera. Ako ne dođe do uspešne konekcije, prikazuje se poruka o grešci i skripta se prekida.

```
public function query($query) {
        if ($this->query = $this->connection->prepare($query)) {
    if (func_num_args() > 1) {
        $x = func_get_args();
        $args = array_slice($x, 1);
                    $types = '';
        $args_ref = array();
        foreach ($args as $k => &$arg) {
                            if (is_array($args[$k])) {
                                    foreach ($args[$k] as $j => &$a) {
                                            $types .= $this->_gettype($args[$k][$j]);
                                            $args_ref[] = &$a;
                                    }
                            } else {
                    $types .= $this->_gettype($args[$k]);
                $args_ref[] = &$arg;
                            }
        }
                    array_unshift($args_ref, $types);
        call_user_func_array(array($this->query, 'bind_param'), $args_ref);
    }
    $this->query->execute();
        if ($this->query->errno) {
                    die('Unable to process MySQL query (check your params) - ' . $this->query->error);
        }
                $this->query_count++;
} else {
    die('Unable to prepare statement (check your syntax) - ' . $this->connection->error);
}
        return $this;
}
```

Slika 13 Funkcija za pripremu upita

Na početku funkcije, u prvoj if proveri, poziva se funkcija **prepare** koja priprema SQL upit i vraća proceduru za dalju obradu događaja (statement handle). Ako je došlo do greške pirkazuje se poruka o grešci (vezana za konekciju) i prekida se skripta. Proverava se broj argumenata i svaki od njih se smesta u poseban niz (**$args**). Prolaskom kroz niz, utvrđuje se tip argumenta funkcijom _gettype (vraća ‘s’ za string, ‘d’ za float, ‘i’ za int i ‘b’ za BLOB (Binary large object) vrednosti). Kroz prolazak se vrši provera da li argument predstavlja niz, ako je to tačno, onda se za svaki element niza utvrđuje tip. Tipovi vrednosti se ubacuju u poseban niz (**$types**), dok se sami elementi ubacuju u niz **$args_ref**. Primetite da se tokom foreach petlje vrednost elementa prenosi po referenci, ne po vrednosti.

Nakon ovoga se funkcijom **array_unshift** vrednosti iz niza $types ubacuju na početak niza $args_ref (potrebno kod povezivanja parametara, prvi argument moraju biti tipovi parametara). Na kraju se korišćenjem funkcije **call_user_func_array** povezuju parametri sa upitom koji je prosleđen kao argument. Funkcija **execute** izvršava upit i povećava broj izvršenih upita. Ukoliko dođe do greške, ona se prijavljuje preko poruke o grešci vezanoj za sam upit i skripta se prekida.

2. 2. Rad sa fajlovima

Na sledeće dve slike su dati primeri rada sa fajlovima prilikom kreiranja aukcije:

```php
$target_dir = "auctions/$category/";
$target_file = $target_dir . basename($_FILES['picture']['name']);
$uploadOk = 1;
$imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));

$check = getimagesize($_FILES['picture']['tmp_name']);
if ($check !== false) {
    echo "File is an image - " . $check['mime'] . ".";
    $uploadOk = 1;
} else {
    echo "File is not an image.";
    $uploadOk = 0;
}

if (file_exists($target_file)) {
    echo "Sorry, file already existss.";
    $uploadOk = 0;
}

if ($_FILES['picture']['size'] > 5000000) {
    echo "Sorry, your files is too large.";
    $uploadOk = 0;
}

if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
        && $imageFileType != "gif") {
    echo "Sorry, only JPG, JPEG, PNG and GIF files are allowed.";
    $uploadOk;
}
```

Slika 14 Provera ispravnosti fajla

Nakon slanja podataka iz forme za kreiranje aukcije, na osnovu odabrane kategorije priprema se direktorijum za smeštanje fajlova i uzima ekstenzija fajla radi provere. Naredna if grananja redom rade:

- Proveru da li je fajl slika
- Proveru da li fajl već postoji
- Proveru da li je fajl veći od 5000000 bajtova
- Proveru da li je fajl odgovarajućeg formata (jpg, png, jpeg, gif)

Ukoliko neka provera vrati *false*, promenljiva **$uploadOk** se postavlja na 0 (false).

```php
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
} else {
    include 'db.php';

    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = '';
    $dbname = 'auctions';

    $db = new db($dbhost, $dbuser, $dbpass, $dbname);

    $userId = $_SESSION['id'];
    $categoryId = $db->query('SELECT category_id FROM categories WHERE name = ?', $category)->fetchArray();
    $categoryId = $categoryId['category_id'];

    $db->query('INSERT INTO listings(expires_at, buyout_price, starting_price, title,
        description, image_path, user_id, category_id)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?)', $expires, $buyout, $starting, $title, $description, $target_file, $userId,
            $categoryId);

    echo "    " . $target_file;
    if (move_uploaded_file($_FILES['picture']['tmp_name'], $target_file)) {
        echo "File has been uploaded";
    }

    $db->close();
}
```

Slika 15 Unos aukcije u bazu podataka i upload fajla

Ukoliko je sve u redu sa fajlom, promenljiva **$uploadOk** je true. U tom slučaju, otvara se konekcija prema bazi, uzima id kategorije za koju se ubacuje aukcija i nakon toga ubacuje sama aukcija u bazu podataka. Sam fajl se pomera u odgovarakući direktorijum i konekcija se zatvara.

2. 3. Sesije i kolačići

```php
if (password_verify($password, $result['password'])) {
    session_regenerate_id();
    $_SESSION['logged'] = TRUE;
    $_SESSION['name'] = $username;
    $_SESSION['id'] = $result['user_id'];

    $cookie_name = "user";
    $cookie_value = $username;
    setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
    header('Location: index.php');
    echo 'Welcome ' . $_SESSION['name'] / '!';
```

Slika 16 Modifikovanje sesije i kolačića

Na slici je prikazan deo koda vezan za logovanje korisnika. Ukoliko je šifra (**password_verify** proverava da li se poklapaju hash vrednosti enkriptovanih šifri) odgovarajuća, kreira se novi id sesije, ime se postavlja na korisničko ime, a polje ‘logged’ na **true**, što se koristi za kasniju proveru autentifikacije korisnika.

Ponovo se postavlja kolačić sa vrednošću korisničkog imena i vremenom trajanja od jednog dana.

3. Javascript (jQuery) i AJAX

Na narednoj slici je prikazano korišćenje jQuery i AJAX-a:

```
<script>

    $(document).ready(function () {
        $(".bid").click(function () {
            var button = this;
            var id = $(button).siblings('[name="bid"]').attr("id");
            var result = "#" + id;

            var bid = $(result).val();

            var user_id = "<?php echo $user_id ?>";

            $.ajax({
                url: 'place_bid.php',
                method: 'POST',
                data: {
                    bid: bid,
                    id: id,
                    user_id: user_id
                },
                success: function (response) {
                    alert(response);
                }
            });
            location.reload();
        });
    });
```

Slika 17 Korišćenje AJAX-a za dinamičko slanje ponuda

U kodu je prikazana funkcija koja nakon klika na dugme klase “bid” uzima odgovarajuće vrednosti i prosleđuje ih stranici “place_bid.php” koja obavlja unos ponude u bazu podataka. Pošto se id svake ponude kreira dinamički, njegova vrednost se koristi korišćenjem **siblings** funkcije, tj. funkcije proverava elemente koji su “u srodstvu” (unutar istog elementa) u odnosu na dugme koje je pritisnuto. Id korisnika se dobija iz vrednosti prethodno inicijalizovane promenljive **$user_id** unutar php koda. Pre ove funkcije se proverava da li su svi html elementi učitani, a nakon nje se vrši osvežavanje stranice kako bi se korisniku prikazale izvršene promene.

4. Zaključak

Predstavljeni projekat predstavlja osnovu za kreiranje aplikacije koja bi se bavila online aukcijom. Moguća proširenja su mnogobrojna, kao već predviđeno ubacivanje komentara, povećana sigurnost prilikom upisa/brisanja iz baze podataka, pravljenje rezervnih kopija podataka, dodatne mogućnosti koje bi olakšale i poboljšale iskustvo korišćenja (pre svega unapređivanjem front-end dela). Takođe, predviđena je i uloga administratora koji bi filtrirao sadržaj na sajtu (izbacivanjem neprikladnih komentara, slika, aukcija), kao i pratio ponašanje korisnika (da li neko koristi više naloga kako bi manipulisao ponudama i sl.).

Najbitnije je uvođenje sigurnog sistema novčanih transakcija i povećane bezbednosti korisničkih podataka, što je naročito bitno nakon uvođenja GDPR regulative, ali ipak prevazilazi zahteve ovog projekta.