

PROJEKTNI ZADATAK

WEB SISTEMI I TEHNOLOGIJE

Izrada web sajta za prodaju

Igrica

Profesor:

Dr, Ljubomir Lazic

Student:

Luka Stoisavljevic

Milan Risojevic

Table of Contents

1.	Uvod	3
2.	Tok izrade projekta	3
3.	Dizajn	4
4.	Struktura stranice	5
5.	UML Dijagrami	5
6.	Opis arhitekture aplikacije	8
7.	Izrada projekta	10

1. Uvod

Ideja projekta

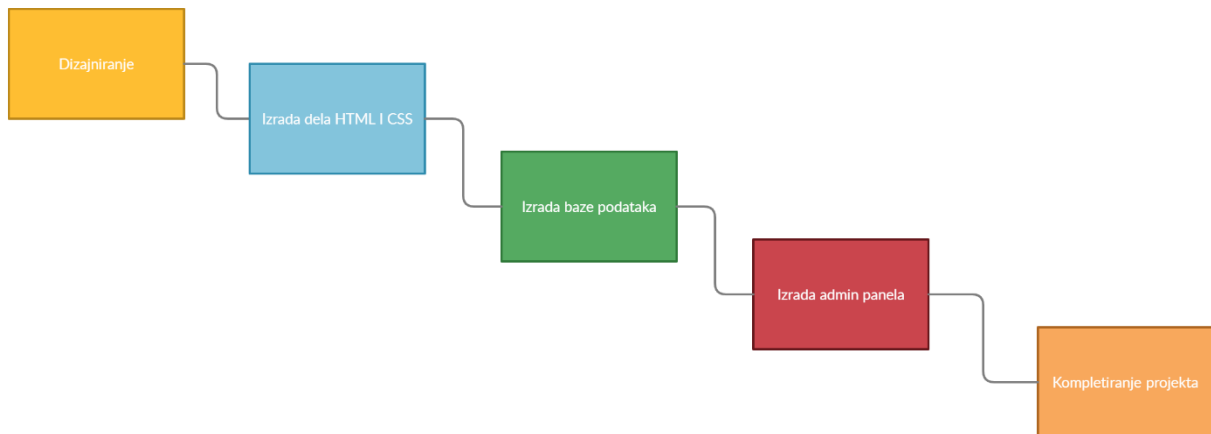
Glavna ideja projekta je da napravimo sajt koji omogućava pregledno i lako kupovinu kompjuterske i gejmerske opreme. Korisnik preko sajta unosi informacije koje kasnije pristizu administratoru. Administrator ima pristup licnom panelu pomocu koga moze da menja cene, stanje na lageru i na kom moze da ima jasan pregled porudzbina.

Od korisnika se jedino trazi kolicina nekog proizvoda koji on zeli da naruci. Kada korisnik potvrdi kupovinu njegov proizvod se nalazi u korpi i kada stisne dugme potvrdi otvara se prozor za prijavu ili registraciju korisnika. Kraj ovog procesa je kada klijent dobije informaciju da je kupovina potvrđena.

Cilj projekta

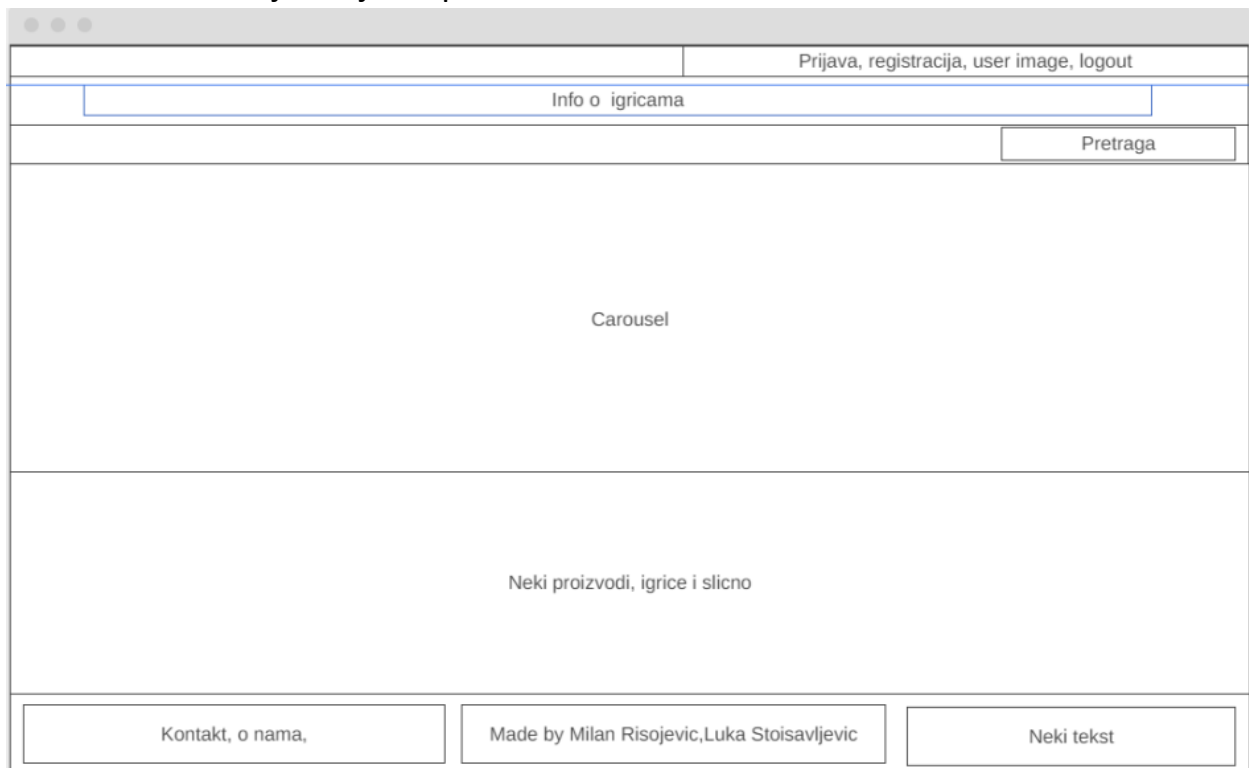
Pomocu ovog sistema kupovine proverava cene, stanja na lageru isključuje bespotrebno dolazjenje u radnju ili kontaktiranje prodavca.

2. Tok izrade projekta



3. Dizajn

Koncept dizajna ove stranice je jako slican kao I prava prodavnica. U prvi plan se isticu cena kao I sam proizvod, a za dizajniranje smo koristili alat wireframe koji Inace služi za dizajniranje responsive internet stranice.



4. Struktura stranice

Na vrhu stranice ce biti fiksni navigacioni div koji ce sadrzati polje za login korisnika, polje za prijavu I registrovanje korisnika, kontakt prodavnice

Prva stranice ce sadrzati same proizvode u ponudi koji ce biti jasno uocljivi I opcije sa njihovu kupovinu.

Druga stranica ce sadrzati najcesca postavljena pitanja.

Treca stranice ce sadrzati funkcionalnu gogl mapu sa lokacijama lokala.

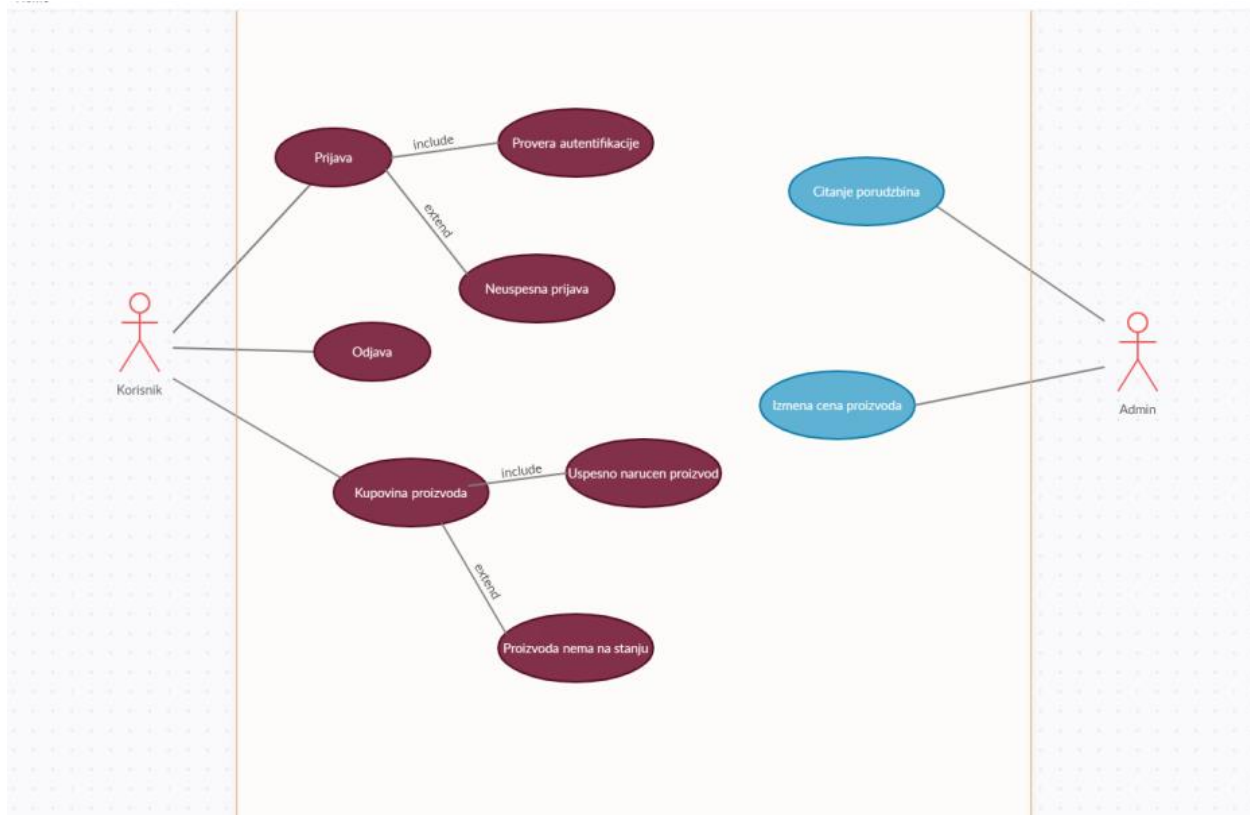
5. UML Dijagrami

Unified Modelling Language je standardni grafički jezik za modelovanje objektno-orjentisanog softvera.

Unified Modeling Language (UML) je jezik za specifikaciju, vizuelizaciju, konstrukciju i dokumentovanje elemenata softverskog sistema, kao i poslovno modelovanje.

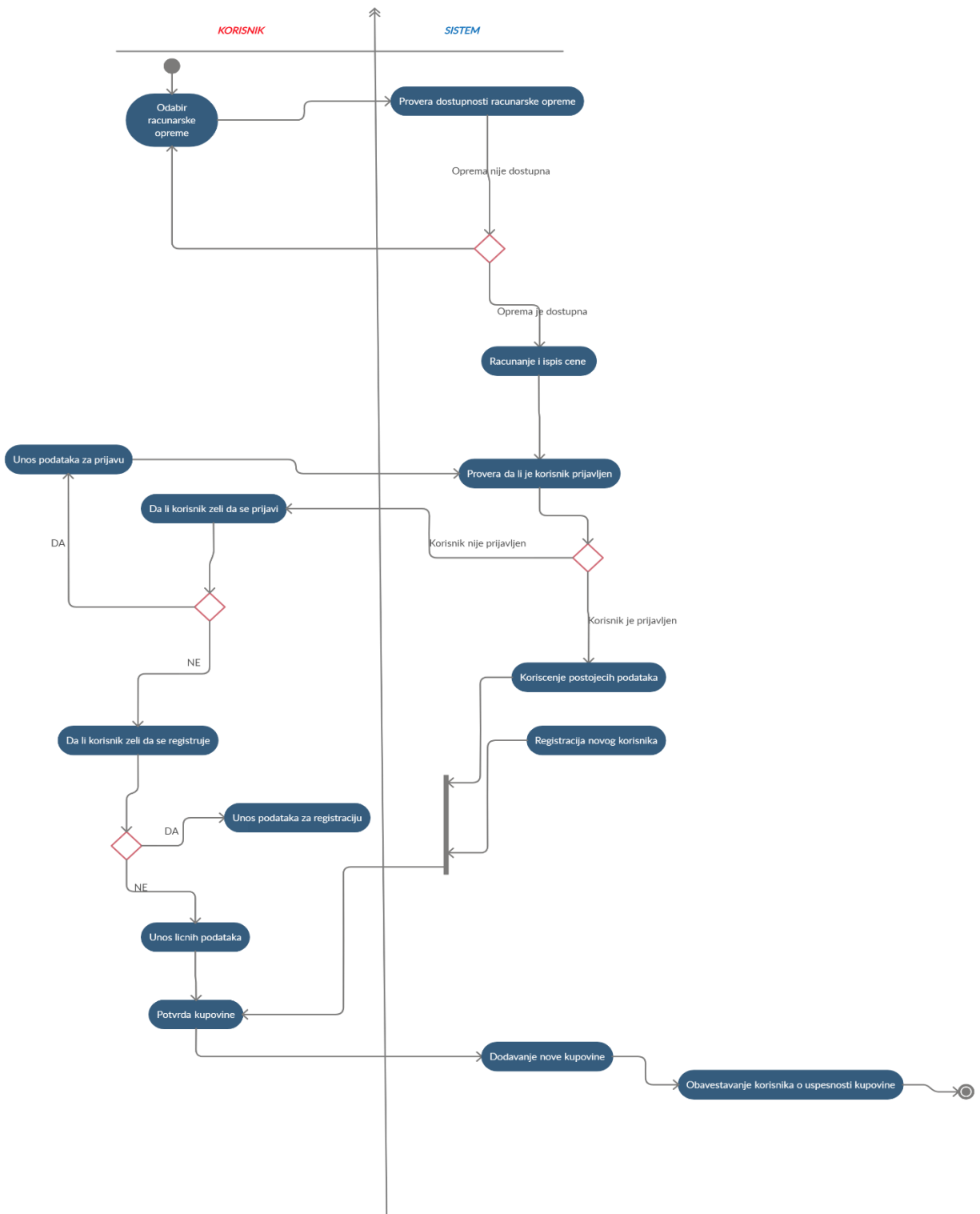
Dijagram slucajeva upotrebe

Dijagram slucaja upotrebe(Use Case diagram)prikazuje odnos izmedju ucesnika-aktera(Actor)I slucajeva koriscenja(Use Case) sistema.Opisuje sta sistem radi sa gledista spoljnjeg posmatraca,ali ne opisuje I kako sistem radi.



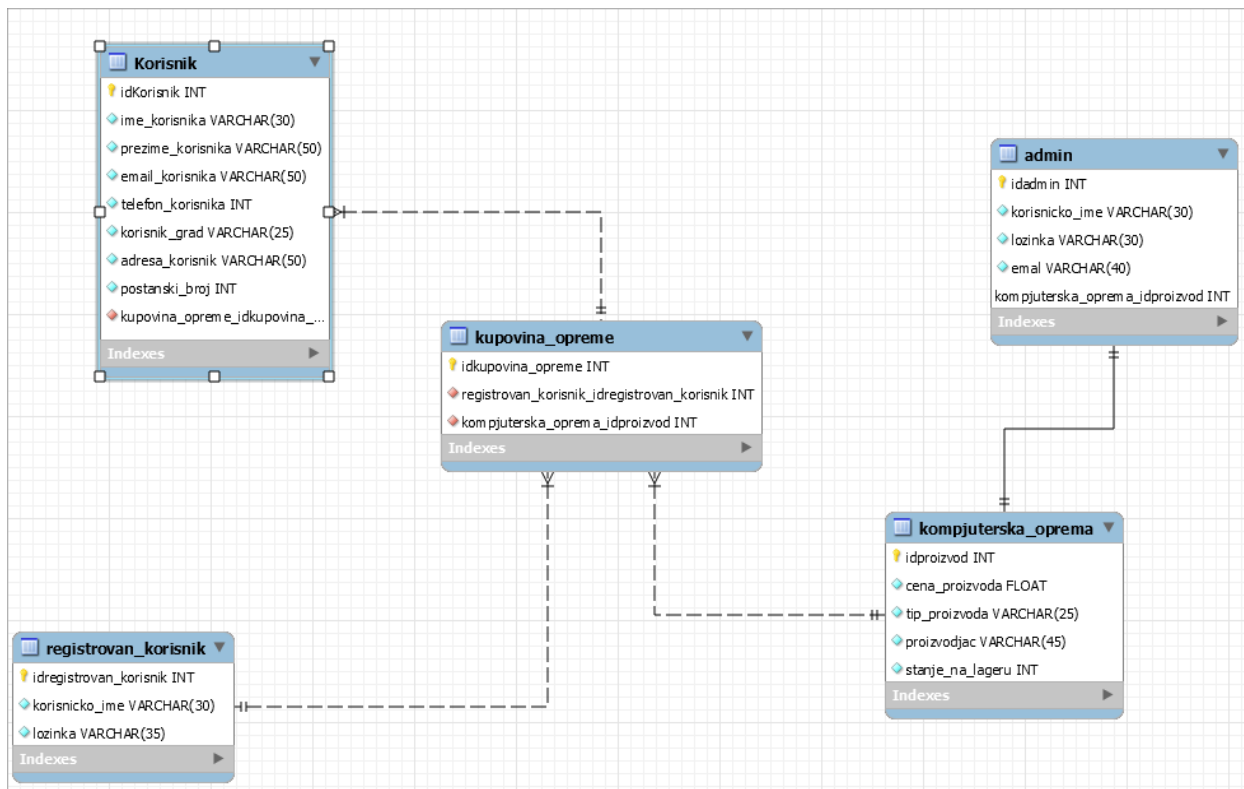
Dijagram aktivnosti

Na slici smo pojednostavili algoritam za procese korisnika i pozadinske procese.



Dijagram klasa

Tokom izrade dijagrama trudili smo se da sto vise uprostimo i uopstimo celu strukturu ovog sistema. Stavili smo dve vrste korisnika: 1. Registrovani korisnici 2. Neregistrovani korisnici, koji nemaju iste privilegije. Sam registrovan korisnik ima privilegiju vise, jer moze da kupuje proizvod na sajtu.



6. Opis arhitekture aplikacije

Prvo je potrebna realizovati izgled pomocu HTML i CSS. Polja za registraciju i za prijavi cemo takodje uraditi preko HTML i CSS. Podaci koji su potrebni za registraciju koji ce komunicirati sa serverom preko php i kada svi podaci budu ispunjeni i potvrđeni dobijamo iskacuci prozor "Registracija uspesna". Maksimalan broj korisnika koji ce moci istovremeno da koriste aplikaciju je priblizno 500.

Admin panel

Cene, popuste kao I stanja na lageru uredjuje admin preko admin panela napravljenog pomocu laravela. Svrha panela je da administrator ima jasan pregled I modifikaciju svih unetih podataka, istom se pristupa pomocu korisnickog imena I lozinke admina iz baze podataka.

Mogucnosti

Korisnik ce moci da:

- Napravi nalog
- Prijavi na nalog
- Odjavi sa naloga
- Naruci proizvod
- Vidi istoriju kupovine
- Otkaze kupovinu

Admin ce moci da :

- Menja cenu proizvoda
- Menja stanje na lageru
- Dodaje popuste na proizvode
- Vidi spisak svih rezervacija

7. Izrada projekta

```
connection.php X
config > connection.php
1  <?php
2
3  require_once "config.php";
4
5  log_page_access();
6
7  try {
8      $conn = new PDO("mysql:host=".SERVER.";dbname=".DATABASE.";charset=utf8", USERNAME, PASSWORD);
9      $conn->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_OBJ);
10     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11 }
12 catch(PDOException $ex){
13     echo $ex->getMessage();
14 }
15
16 function executeQuery($query){
17     global $conn;
18     return $conn->query($query)->fetchAll();
19 }
20
21
22 function log_page_access(){
23     $open = fopen(LOG_FILE, "a");
24     if($open){
25         fwrite($open, "{$_SERVER['PHP_SELF']}\t{$_SERVER['REMOTE_ADDR']}\n");
26         fclose($open);
27     }
28 }
```

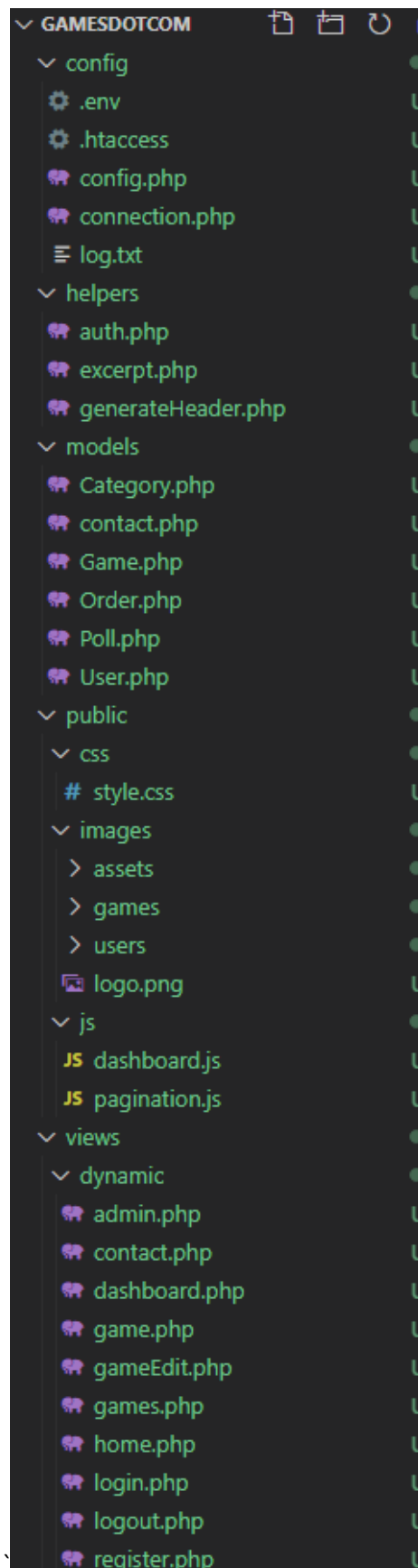
Projekat je radjen u objektno orijentisanom php-u, potrebno je ostvariti konekciju sa bazom podataka. Sva logika vezana za konekciju je predstavljena u config folderu, connection.php fajlu.

Pre koriscenja funkcija u connection.php-u koristi se config fajl u kome se nalaze konstante koje su bitne za povezivanje naseg projekta sa bazom podataka I koristi se funkcija koja iscitava .env paramtere I stavlja ih u konstante.

Da bi se ostvarila konekcija sa bazom koristi se PDO objekat koji je pogodan za objektno orijentisano programiranje. Klasa PDO se instancira sa parametrima iz config fajla iscitanih iz gore navedene funkcije.

Tipovi podataka koje PDO objekat vraca su objekti.

Struktura projekta je slikana da bi se prikazao nacin rada projekta koji podseca na mini framework. Sva logika je smestan u modele, sva struktura stranice i same stranice su smestene u poseban pogled koji se zove views. U public folderu se nalaze fajlovi kojima se moze javno pristupiti kao sus like usera, igrice, sav CSS i JS kod. Sav kod koji nije uokviren ni u jednu klasu se nalazi u folderu helpers.



```

15 function registerUser()
16 {
17     global $conn;
18
19     $user_name = $_POST['username'];
20     $name = $_POST['first_name'];
21     $email = $_POST['email'];
22     $password = $_POST['password'];
23     $passwordConf = $_POST['passwordConf'];
24
25
26
27     if (empty($user_name) || empty($password) || empty($email) || empty($passwordConf) || empty($name)) {
28         header("Location: ../index.php?error=emptyfields&" . $user_name . "&email=" . $email);
29         exit();
30     } else if (!preg_match("/^([a-zA-Z]{0-9})(3,16)$$/", $user_name) && (!filter_var($email, FILTER_VALIDATE_EMAIL))) {
31         header("Location: ../index.php?page=register&error=incorrectfields&" . $user_name . "&email=" . $email);
32         exit();
33     } else if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
34         header("Location: ../index.php?page=register&error=emptyfields&" . $user_name);
35         exit();
36     } else if (!preg_match("/^(?=[A-Za-z])(?=[A-Za-z\d])([A-Za-z\d]{0,15})$/", $password)) {
37         header("Location: ../index.php?page=register&error=incorrectpassword&" . $user_name . "&email=" . $email);
38         exit();
39     } else if (!preg_match("/^([a-zA-Z0-9])(3,16)$$/", $user_name)) {
40         header("Location: ../index.php?page=register&error=emptyusername&&email=" . $email);
41         exit();
42     } else if ($password != $passwordConf) {
43         header("Location: ../index.php?page=register&error=passwordsunmatch&" . $user_name . "&email=" . $email);
44         exit();
45     } else {
46         $query = "SELECT username FROM users WHERE username = :username OR email = :email";
47         $stat = $conn->prepare($query);
48         if (!$conn->prepare($query)) {
49             header("Location: ../index.php?page=register&error=sqlfailed");
50             exit();
51         } else {
52             $stat->bindParam(":username", $user_name);
53             $stat->bindParam(":email", $email);
54             $stat->execute();
55             $result = $stat->fetch(PDO::FETCH_ASSOC);
56             if ($result > 0) {
57                 header("Location: ../index.php?page=register&error=userexists");
58                 exit();
59             } else {
60
61                 $query = "INSERT INTO users (name, email, password, role_id, image, username) VALUES (:uname, :email, :pw, :roleId, :userImage, :username)";
62                 $stat = $conn->prepare($query);
63

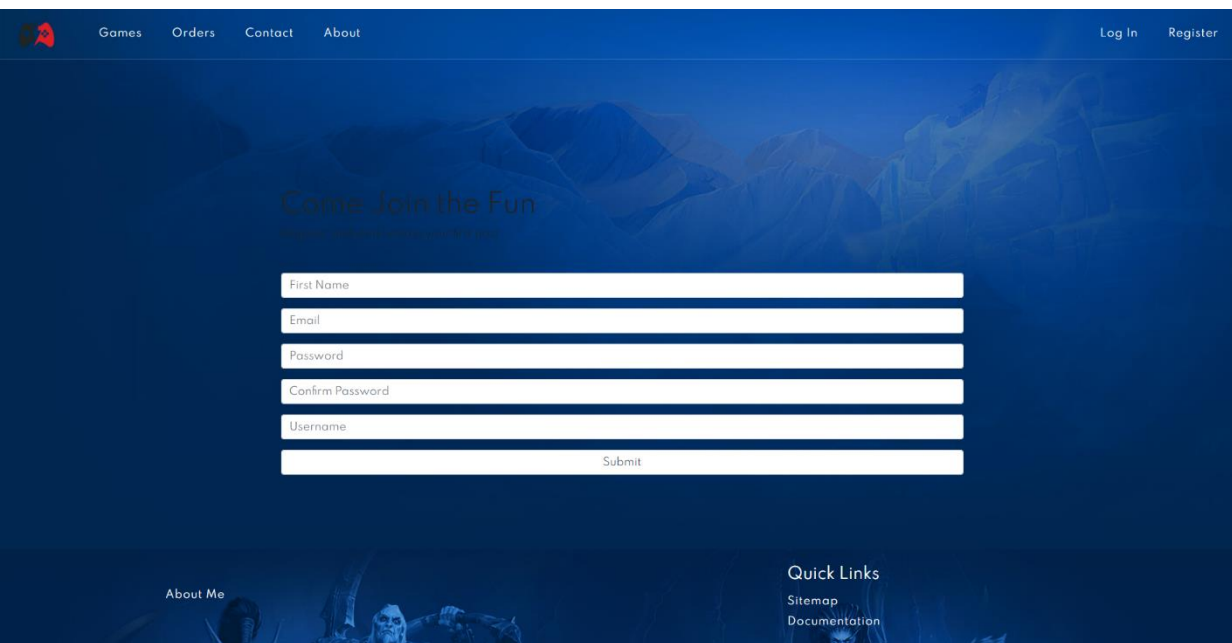
```

```

64         $stat = $conn->prepare($query);
65         $userImage = "public/images/assets/notset.jpg";
66         if (!$conn->prepare($query)) {
67             header("Location: ../index.php?page=register&sqlFail");
68             exit();
69         } else {
70             $roleId = 0;
71             $hashPw = password_hash($password, PASSWORD_DEFAULT);
72             $stat->bindParam(":uname", $name);
73             $stat->bindParam(":email", $email);
74             $stat->bindParam(":pw", $hashPw);
75             $stat->bindParam(":roleId", $roleId);
76             $stat->bindParam(":userImage", $userImage);
77             $stat->bindParam(":username", $user_name);
78             $stat->execute();
79
80             header("Location: ../index.php?page=login&register=success");
81             exit();
82         }
83     }
84 }
85 }
86 }
87

```

Na ovim slikama imam prikazanu metodu za prosledjivanje korisnikovih podataka tokom registracije ka bazi podataka.



The screenshot shows a registration form on a website. The header has a navigation bar with links: Games, Orders, Contact, About, Log In, and Register. The main heading is "Come Join the Fun" with a subheading "Register, and you'll be able to play!". The form consists of five input fields: First Name, Email, Password, Confirm Password, and Username. A Submit button is at the bottom of the form. The footer has two sections: "About Me" and "Quick Links" with links to Sitemap and Documentation.

Games Orders Contact About Log In Register

Come Join the Fun

Register, and you'll be able to play!

First Name

Email

Password

Confirm Password

Username

Submit

About Me

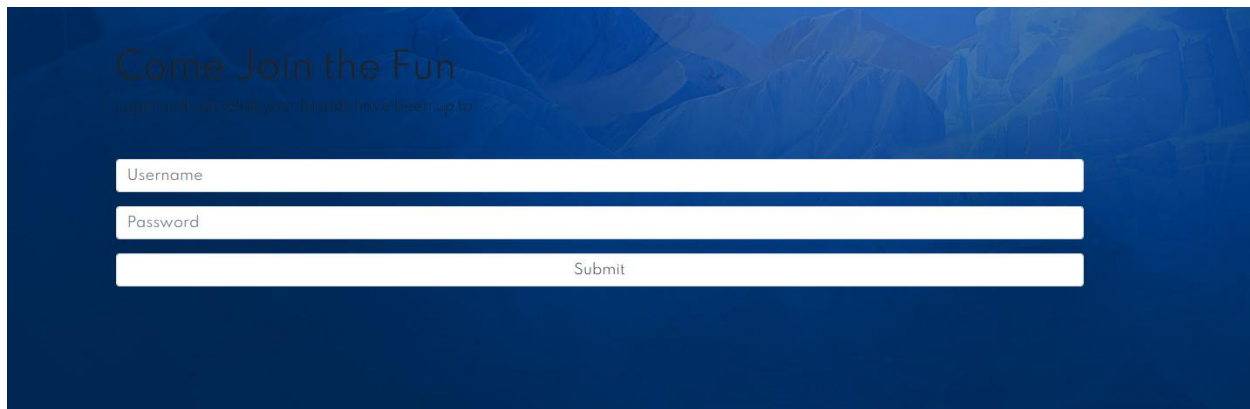
Quick Links

Sitemap

Documentation

Forma za registraciju korisnika se nalazi na posebnoj stranici kojoj se može pristupiti klikom na polje Register u nav bar-u. Izgled stranice možete videti na sledećoj slici.

Sva polja su obavezna za unos!

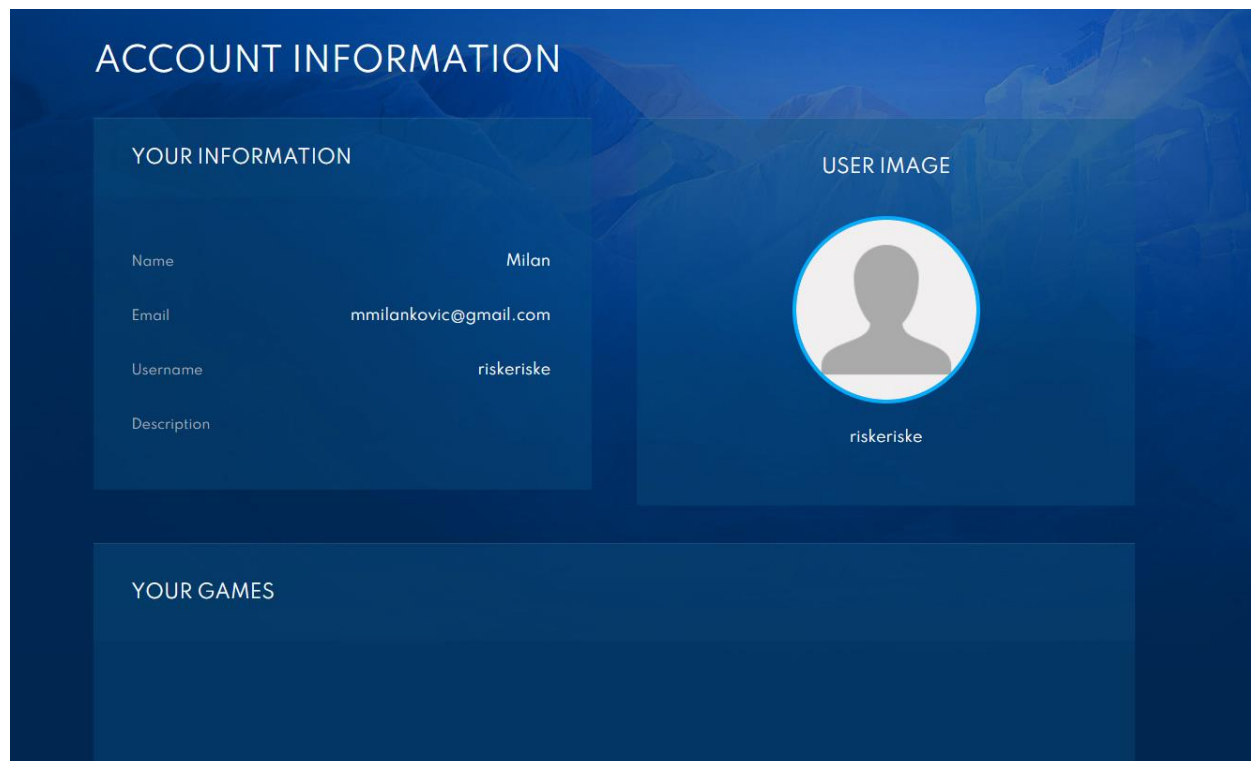


Prijava korisnika

Korisnik se loguje putem forme kojoj pristupamo na stranici login koja se nalazi u nav baru.

```
models > User.php
110 function loginUser()
111 {
112     global $conn;
113
114     $user_name = $_POST['username'];
115     $password = $_POST['password'];
116
117     if (empty($password) || empty($user_name)) {
118         header("Location: ../index.php?page=login&error=emptyfields");
119         exit();
120     } else {
121         $query = 'SELECT * FROM users WHERE username=:username;';
122         $stmt = $conn->prepare($query);
123         if (!$conn->prepare($query)) {
124             header("Location: ../index.php?page=login&statementNotReady");
125             exit();
126         } else {
127             $stmt->bindParam(":username", $user_name);
128             $stmt->execute();
129             $result = $stmt->fetch(PDO::FETCH_ASSOC);
130             $pwdCheck = password_verify($password, $result['password']);
131             if ($pwdCheck == false) {
132                 header("Location: ../index.php?page=login&error=wrongPwd");
133                 exit();
134             } else if ($pwdCheck == true) {
135                 session_start();
136                 $_SESSION['user_id'] = $result['id'];
137                 $_SESSION['name'] = $result['name'];
138                 $_SESSION['user_email'] = $result['email'];
139                 $_SESSION['username'] = $result['username'];
140                 $_SESSION['role_id'] = $result['role_id'];
141                 $_SESSION['user_image'] = $result['image'];
142                 $_SESSION['user_description'] = $result['description'];
143                 header("Location: ../index.php?page=dashboard&login=success");
144                 exit();
145             } else {
146                 header("Location: ../index.php?page=login&error=godKnowsWhatThisIs");
147                 exit();
148             }
149             header("Location: ../index.php?page=login&error=nouser");
150             exit();
151         }
152     }
153 }
```

Login.php



Dashboard-stranica kojoj pristupamo kada se uspesno loginujemo I na kojoj imamo opcije da editujemo nas nalog I same podatke.

Logout.php

```
views > dynamic > 🐞 logout.php
1  <?php
2  if (!isset($_GET['page'])) {
3      header("Location: ../index.php?error=invalidAccessMethodLogout");
4      exit();
5  }
6  session_start();
7  session_unset();
8  session_destroy();
9
10 header("Location: index.php?success=loggedOut");
11
```


Admin panel

Edit Your Account


ALL USERS

Id	Name	Email	Role Id	Username	Actions
1	Kurjak23	kurjakurjakovic22@gmail.com	0	Kurjak	Edit Delete
2	Milan	mmilankovic@gmail.com	1	riskeriske	Edit Delete

Id	Name	Studio	Genre	Price	Actions
2	DOTA 2	Valve	MOBA	0	Edit Delete
2	Counter Strike: Global Offensive	Valve	FPS		Edit Delete

Informacije na panel unam pokazuju da li je logovan user obican korisnik ili admin. Admin ce imati podatke o ostalim korisnicima. Admin ce moci da menja ili brise ostale korisnike ili igrice.

Featured




Description:
Dota 2 is a multiplayer online battle arena (MOBA) video game developed and published by Valve. The game is a sequel to Defense of the Ancients (DotA), which was a community-created mod for Blizzard E...

Name: DOTA 2
Genre: MOBA

Game Studio(s): Valve
Platforms: Steam, and your sooul

Specifications:
OS: Windows 7 or newer Processor: Dual core from Intel or AMD at 2.8 GHz Memory: 4 GB RAM Graphics: nVidia GeForce 8600/9600GT, ATI/AMD Radeon HD2600/3600 DirectX: Version 9.0c Network: Broadband Internet connection Storage: 15 GB available space Sound Card: DirectX Compatible



Description:
Counter-Strike: Global Offensive (CS:GO) is a multiplayer first-person shooter video game developed by Valve and Hidden Path Entertainment. It is the fourth game in the Counter-Strike series and was r...

Name: Counter Strike: Global Offensive
Genre: FPS

Game Studio(s): Valve
Platforms: Steam

Specifications:
RAM - 2 GB Graphics Card - Video card should be at least 256 MB. The AMD Radeon HD 6450 is recommended. Minimum Hard Drive Space - 8 GB DirectX - Version 9 Processor - Intel Core 2 Duo Processor E6600 (2.40 GHz, 1066 MHz FSB, 4M Cache) Operating system - Windows XP 32 bit. The

Naslovna strana sa prikazom igrice mogucih za iznajmljivanje

Send me an e-mail about this project!

Send Mail

Stranica koja služi za kontakt.