

# Dokumentacija - Šah, Računarski praktikum 2

Dora Raštegorac, Vinko Lovrenčić i Luka Jandrijević

July 2021

## 1 Main page

U dokumentu *main2.html* nalazi se view glavne stranice web aplikacije.

Omogućili smo korisniku da u tražilicu upiše korisničko ime osobe protiv koje bi htio igrati i pošalje mu zahtjev. Ako ta druga osoba prihvati oboje su preusmjereni na stranicu sa igrom šah. (Vinkov dio)

Kako bi izbjegli stalno refreshanje stranice pri korisnikovom pretraživanju, listu osoba s tim ili sličnim korisničkim imenom dobivamo uz pomoć ajax-a kojeg pozivamo na dokument *main3.php*

Linije 28-31: naznačujemo da će se ta funkcija pozvati, tj. izvršiti samo kad korisnik pritisne gumb Search

Linije 34-58: funkcija u kojoj se nalazi ajax. Ako je uspješan u listu ću upisati element liste koji vraća stranica *main3.php*

Linije 62-72: imamo ponavljanje koda sa stranice za login - dio koji upućuje korisnika na pomoć pri učenju igranja šaha

Linija 78: link na stranicu za Logout koji smo u css-u oblikovali da izgleda kao gumb

### 1.1 Sparivanje 2 igrača

Jednom kada smo u tražilicu utipkali ime protivnika, možemo izabrati gumb **Send Request** pomoću kojeg šaljemo notifikaciju protivniku da netko s njim/njom želi igrati te automatski korisnika šaljemo na ploču da igra s bijelim figurama.

Pritiskom na gumb **Send Request** poziva se skripta *checkData.php* u kojoj generiramo **gameId** te u varijablu **\_SESSION** stavljamo podatke o protivniku, s kojom bojom se igra i **gameId**. Ako je korisnik stisnuo na nekog drugog osim sebe, tada u bazu podataka spremamo podatke o tome tko je protivnik, **gameId**, **color** i šaljemo ga da igra s bijelim figurama.

Također, na stranici možemo vidjeti i gumb **Check request** pomoću kojeg možemo provjeriti imamo li zahtjev za igru. Ako imamo tada klikom na taj gumb poziva se skripta **requestSent.php** te prelazimo na ploču gdje igramo s crnim figurama. U slučaju da nemamo zahtjev za igru tada je gumb označen crvenom bojom, u protivnom, gumb je označen zelenom bojom. U toj skripti se provjerava ako korisnik ima **gameId** postavljen na 0. Ako nema tada ga ne šaljemo nigdje.

Boju gumba mijenjamo pomoću **ajax-a** i skripte **checkData.php**. U skripti **checkData.php** provjeravamo je li **gameId** jednak 0. Ako je različit od 0 tada šaljemo poruku da se **gameId** promijenio te **ajax** mijenja boju gumba. U protivnom je boja gumba crvena.

## 1.2 style for main page

Linije 9-27: navodimo svojstva scrollbar-a kojeg ima/će imati lista korisnika

Linije 36-50: pozicioniramo naslov na stranici i navodimo njegova osnovna svojstva. Posebno, linije 46-49 zaobljuju bridove naslova - tu metodu koristimo i u oblikovanju ostalih div-ova

Linije 52-80: Definiramo izgled h1 , h2 i h3 tag-ova

Linije 82-90: pozicioniranje i definiranje diva u koji će stizati za poruku/obavijest - biti će hidden dok nema poruke

Linije 92-113: Oblikujemo kućicu za unos teksta kod pretraživanja

Linije 111-121: pozicioniramo kućicu za upis imena

Linije 124-132: pozicioniramo i oblikujemo gumb Search

Linije 134-149: oblikujemo i pozicioniramo div koji sadrži kućicu za pretraživanje imena

Linije 151-164: oblikujemo i pozicioniramo div koji sadrži listu korisničkih imena koja su pretražena

Linije 166-179: Oblikujemo listu i elemente liste

Linije 181-197: oblikuju div u kojem se nalaze linkovi

Linije 199-206: pozicioniranje linkova unutar div-a

Linije 208-232: oblikujemo link za Log out da izgleda kao gumb  
(212-215: oblikujemo rubove, 217-224: zadajemo veličine i pozicioniramo, 226-229: zaobljujemo rubove)

Od linije 234 uz pomoć @media naša stranica postaje responzivna. Uz svaki @media odredimo za koje veličine ekrana vrijedi i unutar njega unosimo izgled div-ova kojih želimo da se vide, za pojedine za koje ne želimo da se vide samo stavimo display:none;

### 1.3 main3.php

Sličan dokument ovom smo prošli na predavanjima. Jedina razlika je što u ovom prvo pristupamo bazi i iz nje izvlačimo potrebne podatke u polje te onda, ako polje nije prazno, vraćamo element liste koji sadrži ime osobe i gumb s kojim korisnik tu osobu zove na igru

## 2 style za login i register

Linije 15-29: oblikuju i pozicioniraju div u kojem se nalazi forma za login/register

Linije 31-46: oblikuju i pozicioniraju div u kojem se nalazi tekst i slika

Linije 48-56: oblikovanje h3 tag-a

Linije 58-66: pozicioniranje i oblikovanje slike šah

Linije 68-76: oblikovanje h2 tag-a

Linije 78-84: pozicioniranje forme za login/register

Linije 86-103: oblikovanje kućica u koje korisnik unosi podatke

Linije 103-110: oblikovanje forme

Linije 113-121: pozicioniranje gumba za login

Linije 123-142: pozicioniranje gumba za register

Linije 152-159: oblikovanje linkova za pomoć pri učenju

Linije 165-179: oblikovanje linkova

Od linije 187 uz pomoć @media naša stranica postaje responzivna. Uz svaki @media odredimo za koje veličine ekrana vrijedi i unutar njega unosimo izgled div-ova kojih želimo da se vide, za pojedine za koje ne želimo da se vide samo stavimo display:none;

## 3 view, login/register i logout

### 3.1 chess.php

Početna stranica aplikacije je `chess.php` u kojem korisnik mora unijeti username i password kako bi nastavio dalje. `chess.php` prvo crta log-in formu te prati koji gumb je kliknut te ovisno o kliknutom gumbu prikazuje određenu stranicu. Ukoliko korisnik prvi put pristupa aplikaciji, tada se mora registrirati pritiskom na gumb register. Klikom na taj gumb korisnika se prosljeđuje na stranicu za registriranje u kojem je potrebno upisati željeni username i password te svoju email adresu. Ako je korisnik krivo utipkao email adresu o tome se javlja greška. Ako username već postoji tada se opet javlja greška i ispisuje poruka.

### 3.2 login

U `login.php` imamo dvije funkcije: `procesiraj_login()` i `procesiraj_novi()`. Prva funkcija provjerava da li se u varijabli `POST` nalazi username i je li taj username valjan. Zatim iz baze podataka dohvaća taj username te provjerava je li se hashirane šifre podudaraju. Ako je sve dobro tada otvara novu sesiju, u tu sesiju sprema username korisnika te ga prosljeđuje na iduću stranicu. Funkcija `procesiraj_novi()` se poziva pri registraciji. Ona služi da kada se korisnik ide registrirati tada u bazu podataka sprema tog korisnika sa unesenim username-om i passwordom.

### 3.3 register.php, processRegistration.php

U `register.php` se nalazi funkcija koja služi za crtanje forme pri registraciji korisnika. Nju pozivamo u `processRegistration.php` kada korisnik još nije kliknuo na gumb register. Jednom kada je kliknuo gumb register, poziva se funkcija `procesiraj_novi()` iz `login.php`.

## 4 Baza podataka

### 4.1 users

U bazi podataka imamo nekoliko tablica. Prva tablica je `users` u kojima spremaamo podatke poput: `username` koji je ujedno i primarni ključ jer nije dozvoljeno da postoje dva korisnika s istim username-om, `opponent` u kojem spremaamo username protivnika protiv kojeg korisnik igra (na početku je prazan), `color` koja označava s kojom bojom korisnik igra (black ili white), `gameId` koji označava u kojoj partiji šaha korisnik igra (ako je `gameId` jednak 0 to znači da korisnik ne igra trenutno partiju). Na kraju, još moramo čuvati i hashiranu lozinku korisnika koju provjeravamo tijekom logiranja na stranicu.

## 5 Model

### 5.1 User

U modelu imamo nekoliko klasa. Najbitnija od njih svih je klasa `User`. U toj klasi imamo definirane funkcije kao što su `getAllUsers()` pomoću koje dohvaćamo sve korisnike aplikacije. Pomoću funkcije `getUserByUsername(username)` dohvaćamo traženog korisnika iz baze podataka. Funkcijom `availablePlayers` pronalazimo korisnike koji trenutno ne igraju partiju, odnosno njima možemo poslati zahtjev za igru. Znamo da su određeni korisnici slobodni za partiju ako im je `gameId = 0`, inače ako je korisnik u partiji tada mu je `gameId` drugačiji broj. Na kraju, imamo funkciju `deleteGame(gameId)` koja služi da obriše `gameId`, odnosno da taj `gameId` postavi ponovno na 0 i tako označi da su ta dva igrača opet slobodni za igru.

### 5.2 Game

U klasi `Game` imamo funkciju `findLastMove` koja pronalazi zadnji potez igre `gameId`. Pomoću funkcije `findColor` možemo pronaći s kojom bojom korisnik igra. Funkcija `insert` ubacuje potez u bazu podataka gdje je `gameId=id`