

Univerzitet u Beogradu
Fakultet organizacionih nauka
Katedra za elektronsko poslovanje

JBOSS APPLICATION SERVER

Seminarski rad iz Elektronskog poslovanja

Nastavnik: Dr. Dušan Barać
Saradnik: Miloš Ninković

Student: Tatjana Parežanović 331/2018
Jovana Perišić 70/2018

Beograd, 2021

SADRŽAJ

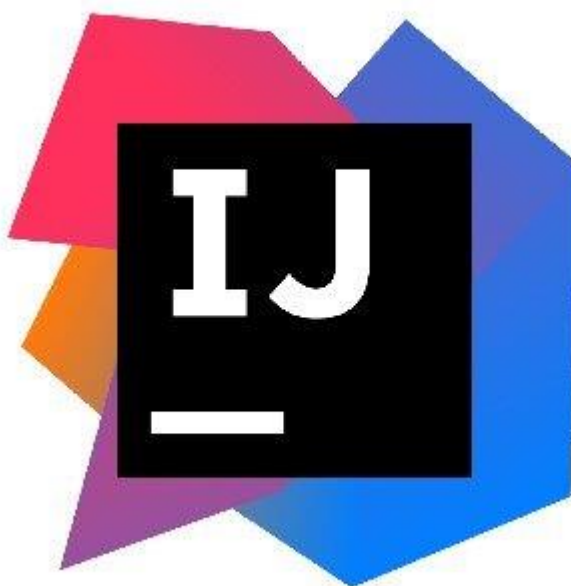
1 UVOD.....	3
2 PREGLED LITERATURE	4
3 PREGLED POSTOJEĆIH REŠENJA.....	5
4 RAZVOJ APLIKACIJE	5
5 ZAKLJUČAK.....	17
6 KORIŠĆENA I REFERENTNA LITERATURA.....	18

1 UVOD

U okviru ovog projektnog rada biće obrađena izrada aplikacije koja simulira rad servera, koji se koristi pri prebrojavanju i provere glasanja. Aplikacija je izrađena korišćenjem Jboss/WildFly aplikativnog servera.

2 PREGLED LITERATURE

Okruženje u kom će aplikacija biti izgrađena naziva se IntelliJ IDEA i predstavlja prevashodno Java orijentisano okruženje. IntelliJ je lansiran od strane JetBrains-a 2001. godine i najvažniji benefiti koje pruža svojim korisnicima su lako korišćenje, fleksibilnost i lep dizajn. Godine 2010. IntelliJ IDEA proglašen je za najbolje Java programsko okruženje.



Slika 1: IntelliJ IDEA logo

WildFly vrši povezivanje objekata napisanih u Javi sa bazom podataka. Takođe pruža konstantnu podršku i okruženje prilikom pisanja koda za rad ove baze. WildFly je zapravo naslednik Jboss-a, koji je preimenovan i rebrendiran 2012. godine i od tada nosi naziv WildFly. Jboss takođe poseduje Enterprise verziju, koja je slična verziji koja je prebačenu u WildFly, ali se ipak razlikuju u nekim osnovnim konceptima i stvari napravljene u jednom okruženju često se ne mogu jednostavno prebaciti u drugo.



Slika 2: WildFly

3 PREGLED POSTOJEĆIH REŠENJA

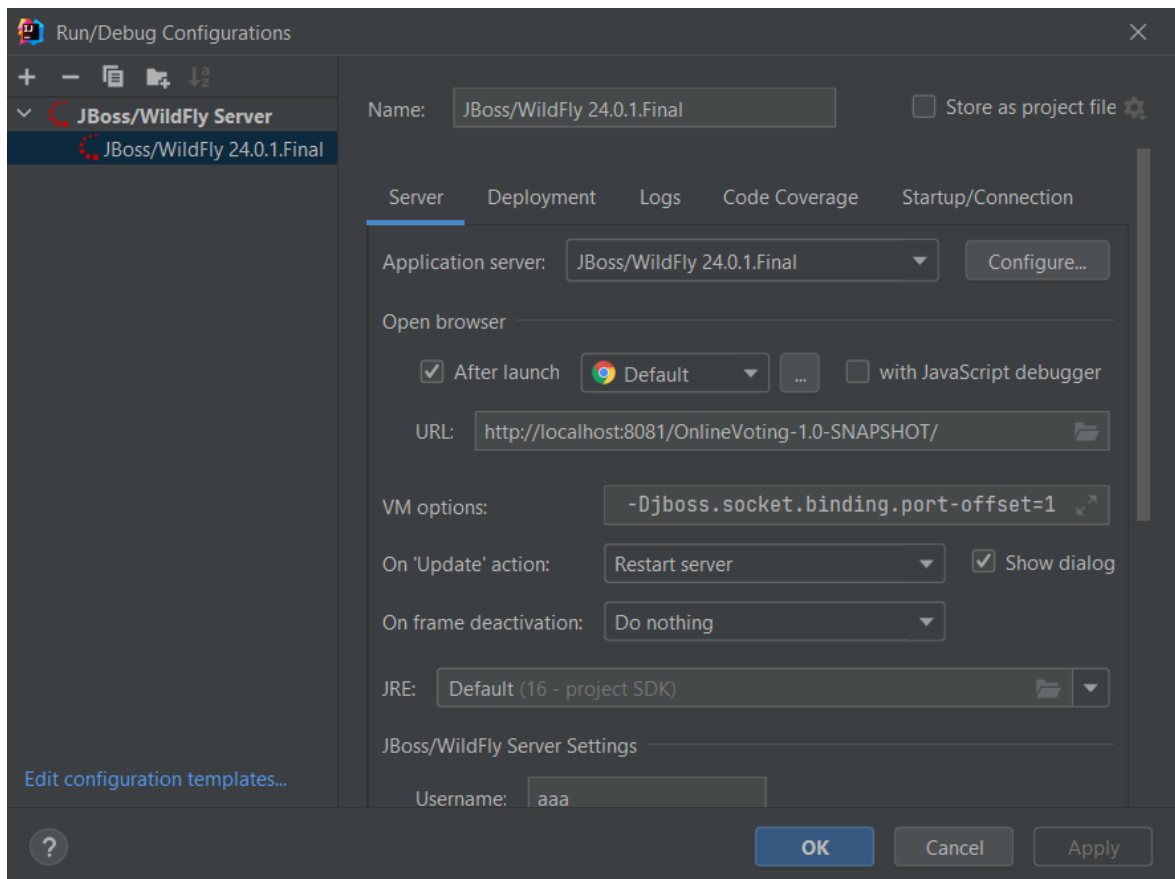
Ovakav tip serverskih aplikacija se teško može naći u upotrebi, međutim ovakva aplikacija bi u budućnosti mogla da se koristi čak i za izborno glasanje, čime bi se značajno olakšao ceo proces glasanja, mada bi naravno postojao problem sa autentifikacijom korisnika i sigurnošću glasanja.

Jednostavan oblik ovakve aplikacije, koji je opisan u ovom radu, bi morao da prođe kroz brojna poboljšanja da bi stvarno mogao da se koristi u takve svrhe.

4 RAZVOJ APLIKACIJE

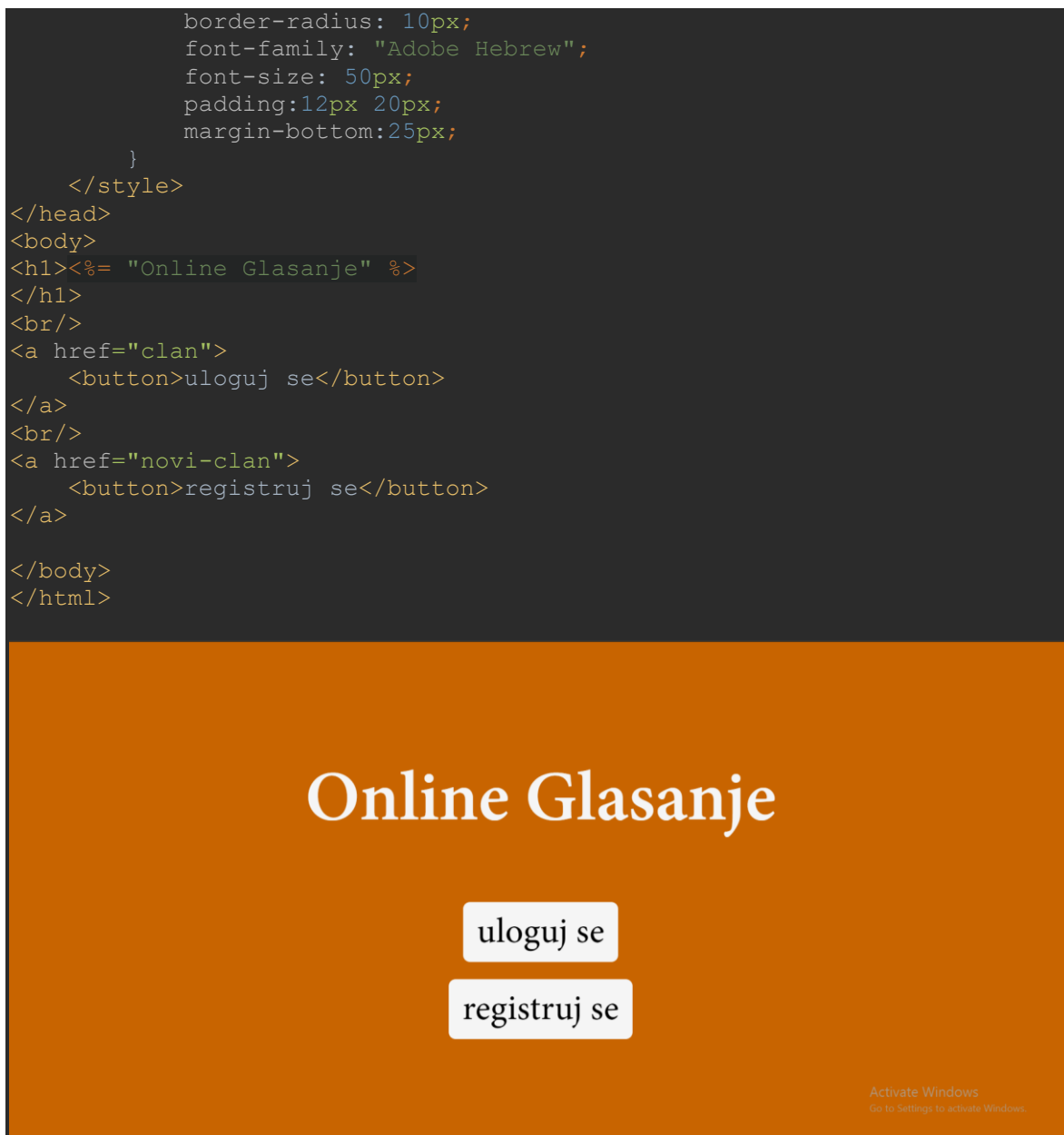
Prvi korak u izradi ove aplikacije jeste preuzimanje IntelliJ IDEA i WildFly-a sa njihovih zvaničnih sajtova. Ovi softveri su besplatni i dostupni za široku upotrebu korisnika, međutim samo na mesec dana, dok se trajna verzija naplaćuje u slučaju IntelliJ-a, dok je WildFly open-source platforma.

Pre početka rada neophodno je povezati WildFly server sa projektom.



Početna strana je odrađena u jsp fajlu korišćenjem HTML5-a. Izgled elemenata na početnoj strani izrađen je u CSS-u koji možemo videti iznad samih elemenata u okviru dela style.

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <title>Online glasanje</title>
  <style>
    body{
      text-align: center;
      padding: 5px;
      background-color: rgb(200,100,0);
    }
    h1{
      margin-top: 10%;
      color: whitesmoke;
      font-family: "Adobe Hebrew";
      font-size: 100px;
    }
    button{
      background-color: whitesmoke;
      border: black 5px;
```



Slika 3: Početna strana

Sa ove stranice, pri prvom ulasku na sajt, korisnik bi trebalo da klikne na dugme registruj se, odakle ga sajt vodi na stranicu za registrovanje (Slika 4). Na ovoj stranici korisnik treba da unese svoje ime, prezime, šifru za nalog koji kreira I svoj JMBG. U okviru ove forme izvršena je validacija ulaska, polje za šifru je postavljeno kao type password, tako da se neće očitavati ono što korisnik unese, a u polje za JMBG moguće je uneti tačno 13 karaktera, koji moraju biti cifre.

First name:

Last name:

Password:

JMBG:

Slika 4: Stranica registracije

```
package com.example.onlinevoting;

import java.io.*;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

@WebServlet(name = "noviclan", value = "/novi-clan")
public class noviclan extends HttpServlet {
    public static List<Glasac> list= new ArrayList<Glasac>();

    public void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException {
        response.setContentType("text/html");

        // Hello
        PrintWriter out = response.getWriter();
        out.println("<html><style>\n" +
            "    body{\n" +
            "        text-align: center;\n" +
            "        background-color: rgb(200,100,0);\n" +
            "    }\n" +
            "    label{\n" +
            "        color: whitesmoke;\n" +
            "        font-family: \"Adobe Hebrew\";\n" +
            "        font-size: 50px;\n" +
            "        margin-top: 2px;\n" +
            "    }\n" +
            "    .textbox{\n" +
            "        width: 40%;\n" +
            "        height: 50px;\n" +
            "    }\n" +
            "    #submit{\n" +
            "        background-color: whitesmoke;\n" +
            "        border: black 5px;\n" +
            "        border-radius: 10px;\n" +
            "    }
        ");
    }
}
```



```

        "            font-family: \"Adobe Hebrew\";\n" +
        "            font-size: 50px;\n" +
        "            padding:12px 20px;\n" +
        "            margin-bottom:25px;\n" +
        "        }\n" +
        "    </style><body>");
    out.println("<form method=\"post\" action=\"noviclanobrada\" >\n"
+
        "    <label for=\"fname\">First name:</label>\n" +
        "    <input class=\"textbox\" type=\"text\" id=\"ime\" "
name=\"ime\"><br><br>\n" +
        "    <label for=\"lname\">Last name:</label>\n" +
        "    <input class=\"textbox\" type=\"text\" id=\"prezime\" "
name=\"prezime\"><br><br>\n" +
        "    <label for=\"password\">Password:</label>\n" +
        "    <input class=\"textbox\" type=\"password\" "
id=\"password\" name=\"password\"><br><br>\n" +
        "    <label for=\"JMBG\">JMBG:</label>\n" +
        "    <input class=\"textbox\" type=\"text\" id=\"jmbg\" "
name=\"jmbg\"><br><br>\n" +
        "    <input id=\"submit\" type=\"submit\" value=\"Submit\" "
>\n" +
        "    </form>");
    out.println("</body></html>");

    }

    public void destroy() {
    }
}

```

Ovim Java kodom je definisana stranica za registraciju, čime se iskucava HTML kod. Ovaj kod se malo razlikuje od koda za prethodnu (početnu) stranu, zbog toga što je u osnovi napisan u Javi, tako da su stvari napisane u HTML-u i CSS-u prikazane korišćenjem funkcije doGet(), koja predstavlja Override metodu u Javi i koja služi za slanje podataka na servlet. Iz tih razloga možemo u delu koda vezanom za HTML i CSS videti da su svi redovi napisano pod navodnicima i korišćenjem \n karaktera za novi red.

Dugme Submit nas vodi na sledeću stranu na kojoj se vrši validacija podataka odnosno proverava se da li je uneti JMBG ima 13 cifara i da korisnik nije prethodno registrovan s istim JMBG-om. U slučaju da nema nikakvih problema uneti podaci se obrađuju funkcijom tipa doPost i kreirani glasač se dodaje u listu koja predstavlja sve registrovane glasače.

```
package com.example.onlinevoting;
```

```

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/noviclanobrada")
public class noviclanobrada extends HttpServlet {

    protected void doPost(HttpServletRequest request,
                           HttpServletResponse response) throws
ServletException, IOException {

        // read form fields
        String password = request.getParameter("password");
        String ime = request.getParameter("ime");
        String prezime = request.getParameter("prezime");
        String jmbg = request.getParameter("jmbg");
        boolean check1=false;
        try {
            Integer.parseInt(jmbg);
        }
        catch (NumberFormatException exp){
            check1=true;
        }
        if (jmbg.length()==13){check1=true;}

        boolean check=false;

        PrintWriter writer = response.getWriter();
        String htmlResponse = "<html><style>body{\n" +
            "    text-align: center;\n" +
            "    padding: 5px;\n" +
            "    background-color: rgb(200,100,0);\n" +
            "    }\n" +
            "    h2{\n" +
            "        margin-top: 15%;\n" +
            "        color: whitesmoke;\n" +
            "        font-family: \"Adobe Hebrew\";\n" +
            "        font-size: 100px;\n" +
            "    }\n" +
            "    button{\n" +
            "        background-color: whitesmoke;\n" +
            "        border: black 5px;\n" +
            "        border-radius: 10px;\n" +
            "        font-family: \"Adobe Hebrew\";\n" +
            "        font-size: 50px;\n" +
            "        padding:12px 20px;\n" +
            "        margin-bottom:25px;\n" +
            "    }\n" +
            "    </style><body>";
        if(check1){
            for(Glasac it:noviclan.list) {
                if (jmbg.equals(it.getJmbg())) {
                    check=true;
                }
            }
        }
    }
}

```

```

        if(!check){
            Glasac g=new Glasac(ime,prezime,password,jmbg);
            noviclan.list.add(g);
            htmlResponse += "<h2>uspesno ste se registrovali<br/>";

        }else{
            htmlResponse += "<h2>već je iskoriscen ovaj JMBG<br/>";
        }else {
            htmlResponse += "<h2>pogresan JMBG<br/>";
        }

        htmlResponse += "<a href=\"index.jsp\"><button>nazad na
pocetnu</button></a>";
        htmlResponse += "</body></html>";

        // return response
        writer.println(htmlResponse);

    }
}

```



Slika 5: Stranica uspešne registracije

Nakon registracije potrebno je ulogovati se sa ranije unetim podacima na stranici clan. Kada se unesu podaci funkcija u klasi clanobrada vrši njihovu proveru odnosno da li se nalaze u prethodno kreiranoj listi. U klasi clanobrada se takođe nalazi mogućnost glasanja gde korisnik mora izabrati prvog ili drugog kandidata. Nakon izvršenog glasanja poziva se stranica brojglasova gde se mogu videti rezultate svih prethodno izvršenih glasanja.

JMBG:

Password:

Slika 6: Stranica za logovanje postojećeg korisnika

```
package com.example.onlinevoting;

import java.io.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

@WebServlet(name = "clan", value = "/clan")
public class clan extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException {
        response.setContentType("text/html");

        // Hello
        PrintWriter out = response.getWriter();
        out.println("<html><style>\n" +
            "                body{\n" +
            "                    text-align: center;\n" +
            "                    background-color:
rgb(200,100,0);\n" +
            "                }\n" +
            "                label{\n" +
            "                    color: whitesmoke;\n" +
            "                    font-family: \"Adobe
Hebrew\";\n" +
            "                    font-size: 50px;\n" +
            "                    margin-top: 2px;\n" +
            "                }\n" +
            "                .textbox{\n" +
            "                    width: 40%;\n" +
            "                    height: 50px;\n" +
            "                }\n" +
            "                button{\n" +
            "                    background-color:
whitesmoke;\n" +
            "                    border: black 5px;\n" +
            "                    border-radius: 10px;\n" +
            "                    font-family: \"Adobe
Hebrew\";\n" +
            "                    font-size: 50px;\n" +
            "                    padding:12px 20px;\n" +
            "                    margin-bottom:25px;\n" +
            "                }\n" +
```

```

        "
        out.println("<form method=\"post\" action=\"clanobrada\" >\n"+
        "    <label for=\"JMBG\">JMBG:</label>\n" +
        "    <input class=\"textbox\" type=\"text\" id=\"jmbg\"
name=\"jmbg\"><br><br>\n" +
        "    <label for=\"password\">Password:</label>\n" +
        "    <input class=\"textbox\" type=\"password\"
id=\"password\" name=\"password\"><br><br>\n"+
        "    <button>uloguj se</button>\n" +
        "    </form>");
        out.println("</body></html>");
    }

    public void destroy() {
    }
}

```

```

package com.example.onlinevoting;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/clanobrada")
public class clanobrada extends HttpServlet {
    public static int g1=0;
    public static int g2=0;
    protected void doPost(HttpServletRequest request,
        HttpServletResponse response) throws
ServletException, IOException {
        boolean x=true;
        // read form fields
        String jmbg = request.getParameter("jmbg");
        String password = request.getParameter("password");
        PrintWriter writer = response.getWriter();
        String htmlResponse = "<html><style>body{\n" +
            "    text-align: center;\n" +
            "    padding: 5px;\n" +
            "    background-color: rgb(200,100,0);\n" +
            "    }\n" +
            "    h2{\n" +
            "        margin-top: 10%;\n" +
            "        color: whitesmoke;\n" +
            "        font-family: \"Adobe Hebrew\";\n" +
            "        font-size: 100px;\n" +
            "    }\n" +
            "    #submit{\n" +
            "        background-color: whitesmoke;\n" +
            "        border: black 5px;\n" +
            "        border-radius: 10px;\n" +
            "        font-family: \"Adobe Hebrew\";\n" +
            "        font-size: 50px;\n" +
            "        padding:12px 20px;\n" +

```

```

        "            margin-bottom:25px;\n" +
        "        }\n" +
        "        .radio{\n" +
        "            width: 50px;\n" +
        "            height: 50px;\n" +
        "        }\n" +
        "    </style>";
    for(Glasac it:noviclan.list) {
        if
(jmbg.equals(it.getJmbg()) && password.equals(it.getPassword())) {
            x=false;
            if(!it.getGlas()){
                htmlResponse += "<h2>" + it.getIme() + "
"+it.getPrezime() + "<br/>";
                htmlResponse += "<form method=\"post\"
action=\"brojglasova\">\n" +
                    "    <input class=\"radio\" type=\"radio\"
name=\"glas\" value=\"1\" checked>\n" +
                    "    <label for=\"html\">kandidat
1</label><br>\n" +
                    "    <input class=\"radio\" type=\"radio\"
name=\"glas\" value=\"2\">\n" +
                    "    <label for=\"css\">kandidat
2</label><br>\n" +
                    "    <input id=\"submit\" type=\"submit\"
value=\"Submit\" >\n" +
                    "</form> ";
                it.setGlas();
                break;}
            else{
                htmlResponse += "<h2>"+ "VEC STE GLASALI" + "<br/>";
                htmlResponse += "<div>prvi kandidat:
"+clanobrada.g1 + "\ndrugi kandidat: " + clanobrada.g2 + "\n</div><br/>";
                htmlResponse += "<a href=\"index.jsp\"><button>nazad
na pocetnu</button></a>";
            }
        }
    }
    if(x){
        htmlResponse += "<h2>pokusajte opet<br/>";
        htmlResponse += "<a href=\"index.jsp\"><button>nazad na
pocetnu</button></a>";
    }
    htmlResponse += "</html>";
    writer.println(htmlResponse);
}
}

```

```

package com.example.onlinevoting;

import java.io.*;
import javax.servlet.http.*;
import javax.servlet.annotation.*;

@WebServlet(name = "brojglasova", value = "/brojglasova")
public class brojglasova extends HttpServlet {

```

```

    public void doPost(HttpServletRequest request, HttpServletResponse
response) throws IOException {
        response.setContentType("text/html");
        String gl = request.getParameter("glas");
        System.out.println(gl);
        if(gl.equals("1")) {clanobrada.g1++;}else {clanobrada.g2++;}

        PrintWriter out = response.getWriter();
        out.println("<html><style>body{\n" +
            "        text-align: center;\n" +
            "        padding: 5px;\n" +
            "        background-color: rgb(200,100,0);\n" +
            "    }\n" +
            "    h1{\n" +
            "        margin-top: 10%;\n" +
            "        color: whitesmoke;\n" +
            "        font-family: \"Adobe Hebrew\";\n" +
            "        font-size: 100px;\n" +
            "    }\n" +
            "    button{\n" +
            "        background-color: whitesmoke;\n" +
            "        border: black 5px;\n" +
            "        border-radius: 10px;\n" +
            "        font-family: \"Adobe Hebrew\";\n" +
            "        font-size: 50px;\n" +
            "        padding:12px 20px;\n" +
            "        margin-bottom:25px;\n" +
            "    }\n" +
            "    </style><body>");
        out.println("<h1>broj glasova:\n");
        out.println("<div>prvi kandidat: "+clanobrada.g1+"\ndrugi
kandidat: "+clanobrada.g2+"\n</div>");
        out.println("<a href=\"index.jsp\"><button>nazad na
pocetnu</button></a>");
        out.println("</body></html>");
    }

    public void destroy() {
    }
}

```

Prethodno pomenuta lista koja simulira bazu podataka je formirana od objekata tipa klase Glasac koja sadži attribute tipa string: ime, prezime, password i JMBG; i atribut tipa boolean koji označava da li je glasač već glasao. Nakon obavljenog glasanja, atribut glas se postavlja na vrednost true.

```

package com.example.onlinevoting;

public class Glasac {
    String ime;
    String prezime;

```

```

String password;
boolean glas=false;
String jmbg;
public Glasac(String ime, String prezime, String password, String
jmbg){
    this.ime=ime;
    this.prezime=prezime;
    this.password=password;
    this.jmbg=jmbg;
}

public String getIme() {
    return ime;
}
public String getPrezime() {
    return prezime;
}

public String getPassword() {
    return password;
}
public boolean getGlas() {
    return glas;
}
public void setGlas() {
    this.glas = true;
}
public String getJmbg() {
    return jmbg;
}
}

```

Tatjana Parezanovic

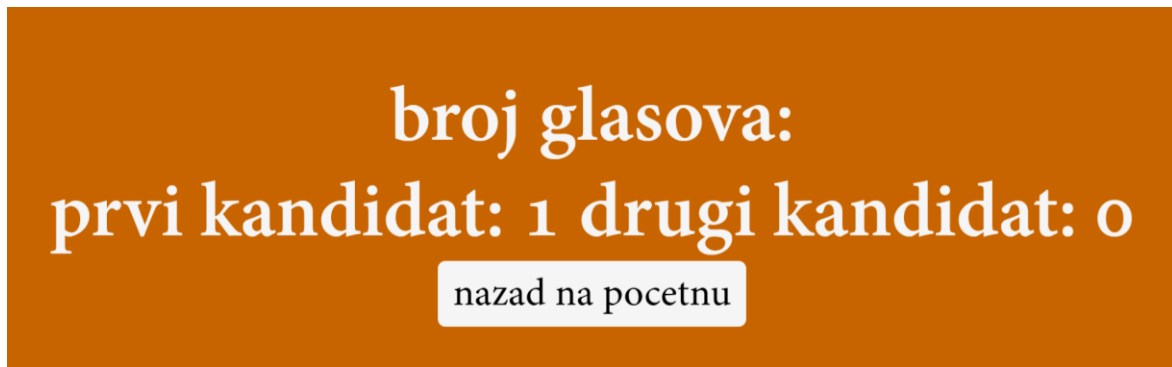
☒ kandidat 1

☐ kandidat 2

Submit

Slika 7: Stranica glasanja

Nakon izvršenog glasanja ili ponovnog pristupa istog korisnika moguće je proveriti trenutne rezultate glasanja.



Slika 8: Prebrojani glasovi

Pristup svakoj stranici je definisan korišćenjem name i value atributa.

```
@WebServlet(name = "brojglasova", value = "/brojglasova")
```

5 ZAKLJUČAK

Serverska aplikacija obrađena u ovom projektnom radu ima veliki potencijal za dalje usavršavanje i korišćenje za brojne svrhe. Trenutno je kompatibilna za popunjavanje nekih jednostavnijih anketa, ali uz dalji napredak očekujemo širu upotrebu. Rad u okviru IntelliJ-a uz WildFly ima veliki broj prednosti, koje se mahom ogledaju u fleksibilnosti i lakoći kodiranja, tako da nije potrebno veliko predznanje da bi se napravio ovakav tip aplikacije. Uz to postoje brojne biblioteke sa dostupnom dokumentacijom, koje mogu dodatno olakšati pisanje koda početnicima.

Jedan od nedostataka na koje smo naišli pri izradi i pokretanju, jeste samo aktiviranje aplikacije, pošto se svaki put mora restartovati server da bi se učitale sve napravljene izmene, što zahteva gotovo ceo minut za proveru svake izmene.

6 KORIŠĆENA I REFERENTNA LITERATURA

- [1] Java Basics: What is WildFly? <https://www.jrebel.com/blog/wildfly>
- [2] IntelliJ IDEA <https://www.jetbrains.com/idea/>
- [3] WildFly <https://www.wildfly.org/>