Anonimne in varne elektronske ankete s sistemom Helios

(Anonymous and secure e-surveys with Helios)

Projekt pri predmetu KTK2

Avtor: Andrej Slapnik 63060240, IŠRM

Mentor: Aleksandar Jurišić

Kazalo

Uvod 3
Že implementirani sistemi za e-volive oz. ankete....4
Naivni pristop....4
Identifikacija in avtentikacija....5
Pregled možnih načinov identifikacije in avtentikacije....5
Osnovna HTTP avtentikacija....5
HTTP avtentiakcija z izvlečkom....5
HTTPS avtentikacija....5
Avtentikacija preko prijavnega okna....5
Anonimizacija....5
Uporaba odprtokodnega sistema za elektronske volitve Helios....6
Predstavitev....6
Namestitev....6
Uporaba....9
Zaključek....12
Viri....13
Priloge....14
Nginx nastavitve....14
Apache nastavitve....15

1. Uvod

Ocenjevanje profesorjev trenutno poteka po naslednjem sistemu: Študenti dobijo ocenjevalni list formata A4 za vsak predmet v letniku posebej, na tem listu ocenijo 7 lastnosti profesorja in asistenta (1-4), označijo kako pogosto so predavanja ali vaje obiskovali, podajo komentar na izvajanje in dopišejo pričakovano oceno na koncu pa eden izmed študentov zbere vse ocenjevalne liste in jih odnese v referat. Seveda se študenti na liste ne podpišejo ter tako omogočijo svojo anonimnost.

Pri tem načinu ocenjevanja je torej omogočena določena mera anonimnosti, saj se le ta veča premosorazmerno s številom študentov, ki so sodelovali pri ocenjevanju. Izvedba ocenjevanja je razmeroma lahka.

Slabih strani trenutnega načina ocenjevanja pa je kar nekaj. Veliko dela s pripravo ocenjevalnih listov, velika poraba papirja/svinčnikov, neustrezno pobarvana polja za oceno, skupinsko ocenjevanje med študenti, možnost večkratnega ocenjevanja enega študenta (fotokopiranje ocenjevalnega lista) ter velika poraba časa za vpisovanje ocen in samo analizo ocenjevalnih listov.

Vidimo, da je potrebno ocenjevanje čimprej digitalizirati in ga izvesti s sodobnimi tehnologijami, opis tega sistema pa je tudi cilj te seminarske naloge. Zadnja leta se izredno hitro razvijajo spletne tehnologije, zato bom svojo rešitev predstavil kot spletni sistem za ocenjevanje. V tej seminarski nalogi bom poskusil predlagati ustrezen elektronski sistem za izvajanje ocenjevanja profesorjev. Sistem mora biti transparenten, ustrezno mora biti poskrbljeno za varnost (avtentikacija in šifrirana povezava), omogočati pa mora popolnoma anonimno ocenjevanje.

2. Že implementirani sistemi za e-volive oz. ankete

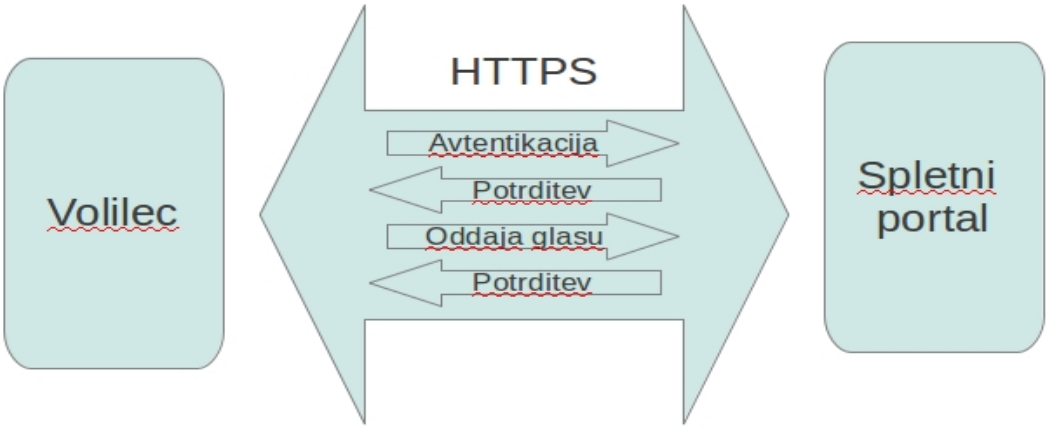
Na splošno ločimo dve vrsti elektronskih volitev. Prvi tip so takoimenovani aparati za glasovanje, ki se ponavadi nahajajo na neki lokaciji, nadzorovani s strani upravitelja (npr. predstavnikov vlade). Volilec mora priti na mesto, kjer se aparat nahaja, tam pa potem odda svoj glas. Drugi tip je takoimenovani oddaljeni način, ki omogoča, da volilec odda glas neodvisno od svoje lokacije, npr. prek interneta. Ta sistem je veliko bolj priročen in bolj enostaven za volilca, a zahteva veliko večjo skrb pri zagotavljanju varnosti glasovanja in pristnosti rezultatov.

Medijsko najbolj odmevne so bile predsedniške volitve v ZDA leta 2000, ko so v Kaliforniji našteli kar nekaj napak med elektronskim glasovanjem. Prav tako so imeli leta 2004 težave s štetjem glasov na Floridi. Elektronske volitve so testirali tudi na Finskem, vendar so jih najprej izvedli le v nekaj okrožjih. Ker je prihajalo do uporabniških napak (npr. volilec je prišel na avtomatu do oddaje glasu, vednar glasu ni oddal in odšel iz volišča) so prekinili proces uvajanja elektrosnkih volitev v celotni državi. Elektronske volitve so izvedli tudi v Estoniji, žal pa je elektronsko glasovalo le nekaj čez 1 procent volilnih upravičencev.

Kot vidimo, se večina nepredvidenih in težko rešljivih problemov pojavi, ko več volilcev na enem mestu oddaja svoje glasove. Ta problem bomo poskusili zaobiti z oddaljenim glasovanjem.

3. Naivni pristop

Naivni sistem izvedbe ankete bom opisal v naslednjih podpoglavjih. naivni potek glasovanja prikazuje spodnja skica.



Volilec se bo s pomočjo SSL (http://httpd.apache.org/docs/2.0/ssl/ssl_intro.html) certikata prijavil na ustrezen spletni portal. Ta bo preveril ali je volilec ustrezen in ali še ni oddal svojega glasu. Vkolikor bosta pogoja zadoščena, ga bo portal spustil do strani za oddajo glasu oz. oddaje ocene. Po uspešni oddaji si bo sistem označil, da je volilec že sodeloval na volitvah, ter njegov glas shranil v množico že oddanih glasov. Glavni problem, s katerim bi se soočili pri tej implementaciji je volilčeva možnost preverjanja, da je bil njegov glas res upoštevan.

Identifikacija in avtentikacija

Za pravilno in varno delovanje moramo dobro poskrbeti tudi za identifikacijo obiskovalca in njegovo avtentikacijo. Uporabnika mora sistem pravilno prepoznati, uporabniku pa se mora seveda preprečiti lažno prijavo v sistem.

Pregled možnih načinov identifikacije in avtentikacije

Osnovna HTTP avtentikacija z geslom

Je najosnovnejši tip avtentikacije. Strežnik vpraša uporabnika o uporabniškem imenu in geslu. Obe informaciji se preneseta na strežnik pretvorjeni v bazo 64, kar pomeni, da bi te podatke tretja oseba brez večjega problema prebrala.

HTTP avtentikacija z izvlečkom gesla

Ta tip deluje podobno kot osnovna avtentikacija, opisana v predhodnem odstavku. Strežnik vpraša uporabnika po uporabniškem imenu in geslu, nato pa ju primerja z zapisi na svoji strani.

Glavna razlika je v tem, da so gesla shranjena v neberljivi obliki. Ponavadi je to narejeno s pomočjo zgoščevalnih funkcij. Tako tudi administratorju strežnika preprečimo branje gesel. Ker ima sposobnost prepoznavanja pravilnosti gesla tako le strežnik, lahko trdimo, da je sama varnost večja kot pri osnovni avtentiakciji.

HTTPS avtentikacija

HTTPS dosežemo, ko protokol HTTP združimo s protokolom Secure Socket Layer (SSL). Tako so vsi podatki, ki se prenašajo po medmrežju oviti v varni šifrirani okvir SSL-a. To omogoča strežniku, da za vsako stran, ki jo uporabnik zahteva, preveri, ali jo le ta lahko vidi.

Avtentikacija preko prijavnega okna

Prijavno okno/prijavna stran sta pogosto uporabljena na straneh, kjer potreba po varnosti ni kritična. Uporabnik se uspešno avtenticira tako, da v predvidena in zahtevana polja na spletni strani ali spletnem obrazcu, pravilno vnese zahtevane podatke. Ponavadi sta to uporabniško ime in geslo, vedno bolj pogosta pa je tudi uporaba tehnike branja znakov s slik, s čimer preprečimo napade robotov (captcha).

4. Anonimizacija

Potem ko je uporabnik avtenticiran in imamo zagotovoljeno varno/šifrirano povezavo pride na vrsto glavni del, to je izvedba glasovanja. Poskrbeti moramo, da vsak uporabnik po oddaji dvojega glasu, ocene ali mnenja ostane anonimen. Potreba po anonimnosti je seveda nujna, saj tako preprečimo kakršnokoli kaznovanje/nagrajevanje, do katerega bi privedlo poznavanje opredeljenosti.

Anonimizacijo moramo izvesti tako dobro, da tudi administrator strežnika ne no mogel ugotoviti kako je kdo glasoval. Ponavadi pr anonimizaciji pridemo do problema večkratnega glasovanja. Problem pri večkratnem glasovanju se pojavi, ker si ne smemo beležiti, kdo je oddal kateri glas. V nadaljevanju

bom opisal rešitev tudi za ta problem.

Večkratno glasovanje lahko preprečimo z ločenim pomnenjem že oddane glasovnice in samih glasov. Tako bi uporabniku le enkrat dovolili glasovanje, po končanem glasovanju pa njegovo identifikacijsko številko shranili v množico "že sodelujočih". Ob ponovni prijavi bi preverili ali njegova številka v tej množici že obstaja, in če, bi mu glasovanje onemogočili.

5. Uporaba odprtokodnega sistema za elektronske volitve Helios

Predstavitev

Odprtokodni sistem Helios (www.heliosvoting.org) je leta 2008 razvil Ben Adida. Vsebuje kar nekaj šifrirnih prijemov, ki omogočajo, da se sistem približa potrebni stopnji varnosti. Uporablja ElGamalov sistem, odšifriranje glasovnic je lahko porazdeljeno med več zaupnikov, Chatum-Pedersenov dokaz o enakosti diskretnih logaritmov ter Schnorrov dokaz o poznavanju diskretnega logaritma nekega elementa.

Sistem omogoča avtentikacijo preko že preverjenih avtentikacijskih sistemov kot so Google, Facebook in Yahoo. Z nekaj prilagoditvami lahko vpeljemo tudi avtentikacijo s pomočjo certifikatov. Avtentikacijo se lahko preverja tudi po oddanem glasu – na email naslov dobimo posebno geslo za potrditev glasu – ter takoše dodatno zaščitimo svoj glas.

Glasovi se pred zapisom v bazo ustrezno zašifrirajo, vseeno pa glasovalec dobi za vsak svoj glas poseben "izvleček", s katerim lahko kasneje na "oglasni deski" preveri ali je bil njegov glas res upoštevan. Glasove lahko da prešteti le administrator, ob potrebi pa se glasovnice razkrijejo le s strani zaupnikov volitev.

Helios podpira tako "referendumski" tip volitev (možna le glasova za ali proti), kot tudi navadne volitve (več kandidatov). S pomočjo teh funkcionalnosti lahko rešimo naš problem ocenjevanja profesorjev.

Namestitev

Helios za pravilno delovanje potrebuje doklaj specifično programsko opremo. Za postavitev na namenskem strežniku je potrenih kar nekaj izkušenj s področja administracije strežnikov, kot tudi poznavanja Python jezika in frameworka Django. Nekaj težav sem imel le pri poznavanju Djanga, ostale posebnosti pa sem z nekaj truda prepoznal sam.

Operacijski sistem namenskega strežnika naj bil bil Linux. Distribucija naj ne bi bila pomembna, vendar Ben Adida predlaga Ubuntu verzije 10.10. To verzijo sem uporabil tudi sam. Uporabljen sistem za upravljanje podatkovne baze je Postgresql. Priporočena verzija je 8.4. sam sem uporabil 9.1 in lahko potrdim, da deluje brez problema. T.i. Modele za podatkovno bazo uvozimo s pomočjo knjižnice python-psycopg2, framework mora OBVEZNO biti Django 1.2. Helios ne deluje niti na starejši 1.1 verziji, niti na novejši 1.3. Za delovanje dnevnih nalog (cronjob) na strežniku je uporabljem rabbitmq-server v kombinaciji s Celery knjižnico. Vse skupaj namenstimo s pomočjo naslednjih ukazov:

```
$ apt-get install postgresql
$ apt-get install python-psycopg2
```

```
$ apt-get install python-django
$ apt-get install rabbitmq-server
$ apt-get install python-setuptools
$ easy_install celery
$ easy_install django-celery
```

Helios za avtentikacijo v osnovi podpira sisteme, ki jih jih ponujajo Google, Tweeter, Yahoo in Facebook. Če imamo račun pri katerem od naštetih ponudnikov storitev, lahko Helios uporabimo brez registracije ali dodatne avtentikacije. Da podpremo te fukcionalnosti pri Heliosu, moramo namestiti še sledeča dva paketa:

```
$apt-get install python-django-openid-auth
$ easy install south
```

Za pravilno in varno delovanje podatkovne baze moramo v SUPB-ju ustvariti tudi posebnega uporabnika, ki bo zadolžen le za Heliosovo podatkovno bazo, ter podatkovno bazo samo:

```
$ su - postgres
$ createuser --superuser helios
$ createdb -O helios heliosdb
```

Helios za komunikacijo z uporabniki, za del avtentikacije in poročila uporablja email sporočila. Strežnik mora torej podpirati tudi pošiljanje elektronske pošte. Največkrat uporabljanai sistem na Linuxu je prav zagotovo Postfix. S sponjim ukazom le tega namestimo in se sprehodimo čez kar enkaj nastavitev, ki pa niso del te seminarske naloge (lahko pa me brez problema povprašate o specifičnih nastavitvah):

```
$ apt-get install postfix
```

Najpomembnejši del namestitve je prav gotovo namestitev sistema Helios samega. Avtor ponuja vedno svežo različico apliakcije preko porazdeljenega sistema za verzioniranje GIT. Ob predpostavki, da je GIT nameščen na strežniku, si Helios prenesemo z naslednjim ukazom:

```
$ git clone git://github.com/benadida/helios-server.git
$ cd helios-server
```

Helios za delovanje potrebuje tudi nekaj podmodulov, ki jih namestimo s spodnjimi ukazi:

```
$ git submodule init
$ git submodule update
```

Glavne nastavitve Heliosa se nahajajo v datoteki settings.py. Na začetku v apliakciji te datoteke ni, obstaja pa podobna datoteka s predlaganimi nastavitvami. Skopiramo to datoteko in spremenimo vse potrebne nastavitve glede na naše želje in potrebe:

```
$ cp settings.py.sample  settings.py
```

Nastavitve v datoteki settings.py:

```
DATABASE_NAME  = 'heliosdb'
DATABASE_USER  = 'helios'
DATABASE_PASSWORD  = 'heliospass'
DATABASE_HOST  = '127.0.0.1'
DATABASE_PORT  = '5432'
DEFAULT_FROM_EMAIL  = 'Andrej Slapnik <andrejev.mail@gmail.com>'


URL_HOST = "URL vaše strani, kjer bo dostopen sistem Helios"
SECURE_URL_HOST = "https://URL vaše strani, kjer bo dostopen sistem Helios:443"
SOCIALBUTTONS_URL_HOST = "URL vaše strani, kjer bo dostopen sistem Helios:80"
SITE_TITLE = 'Helios Server'
FOOTER_LINKS = [{'url':'URL vaše strani, kjer bo dostopen sistem Helios',
'text':'Opis'}]
WELCOME_MESSAGE = "welcome"
```

Nastavitve, kot ste jih nastavili v sistemu Postfix:

```
EMAIL_HOST  = 'localhost'
EMAIL_PORT  = 25
EMAIL_HOST_USER  = ''
EMAIL_HOST_PASSWORD  = ''
EMAIL_USE_TLS  = False
```

Urediti moramo še datoteko reset.sh, ki nam postavi podatkovno bazo in resetira vse nastavitve:

```
#!/bin/bash
dropdb heliosdb -U helios -W
createdb heliosdb -U helios -W
python manage.py syncdb;
python manage.py migrate;
echo "from auth.models import User;
User.update_or_create(user_type='password',user_id='benadida');”
```

Zaženemo prej omenjeno datoteko in zaženemo Celery proces:

```
sh ./reset.sh
$ python manage.py celeryd
```

Helios mora za delovanje prek spletna teči na spletnem strežniku. Ker potrebujemo tudi varno SSL povezavo ter ustrezno hitrost, bomo uporabili dva spletna strežnika. Nginx bo deloval kot proxy, strežnik, ki bo sprejemal zunanje povezave, Apache strežnik, pa bo sprejemal povezave od Nginxa in jih preusmerjal na sistem Helios. Na našem namenskem strežniku moramo tako namestiti tudi oba spletna strežnika:

```
$ apt-get install libapache2-mod-wsgi apache2 nginx
```

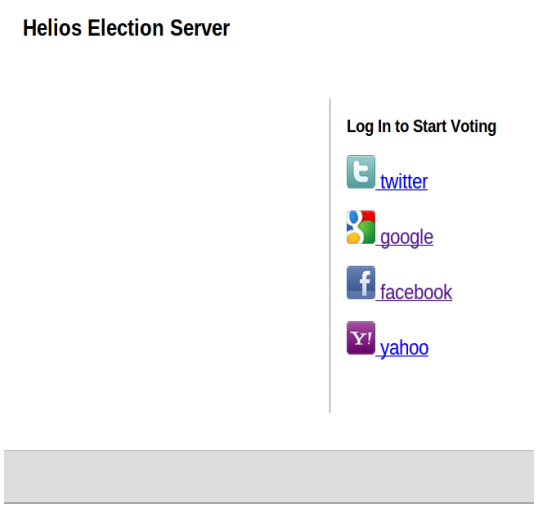
Za delovanje SSL povezave potrebujemo tudi SSL certifikat, ki pa za naše potrebe ne potrebuje biti podpisan s strani zaupnic. Certifikat zgeneriramo s pomočjo naslednjega ukaza, ter ga prestavimo na varno mesto. Mesto si zapomnimo, saj se moramo ob prihajajoči SSL povezavi sklicevati nanj.

```
$ sudo make-ssl-cert /usr/share/ssl-cert/ssleay.cnf
/etc/apache2/ssl/apache.p
```

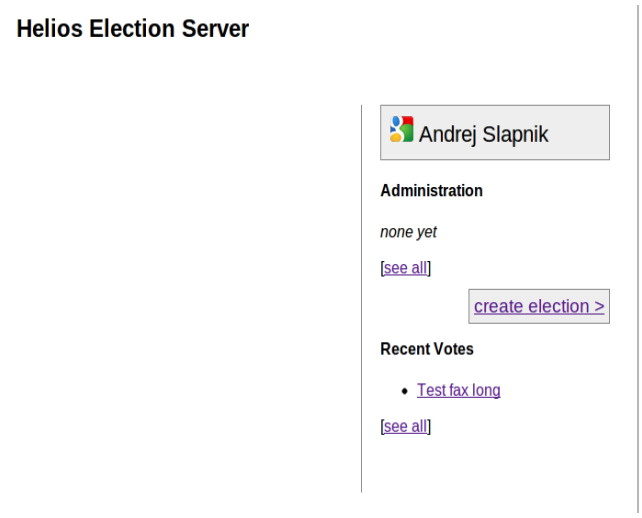
Prihajajoče povezave na naš strežnik preko porta 80 (HTTP) in 443 (HTTPS) sprejema spletni strežnik Nginx. Ta glede na tip povezave preusmeri le-to na ustrezno mesto. Če povezava prihaja preko porta 80 (neSSL), strežnik povezavo preusmeri direktno na Helios aplikacijo. Če povezava zahteva preverjanje certifikata, Nginx povezavo preusmeri na lokalni sosednji strežnik Apache, ki preveri ustreznost certifikata in nato povezavo usmeri na apliakcijo Helios. Nastavitve obeh strežnikov najdete v prilogi na koncu seminarske naloge.

Uporaba

Ko pridemo na prvo stran aplikacije se nam najprej ponudijo vse možnosti prijave. Na voljo so nam prijava s pomočjo Tweeter, Google, Yahoo ali Facebook računa. Za naše potrebe se bomo prijavili z Google računom. Če smo v nastavitvah označeni kot administratorji dobimo po prijavi možnost dodajanja volitev ali glasovanja, če smo v katere volitve povabljeni ali so odprte za vse.



Prijavno okno

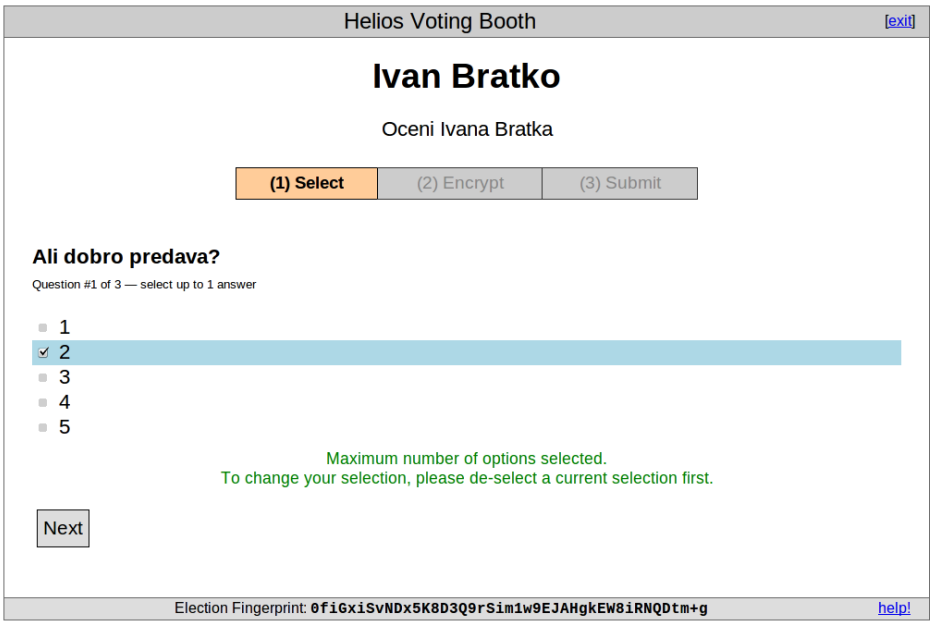


Prva stran

Če hočemo dodati volitve ali anketo, preprosto vpišemo vprašanje, dodamo možne odgovore in določimo število možnih odgovorov. Nato določimo kdo vse lahko voli ter izračunamo vse potrebne kriptograme in izvlečke. Na koncu še pošljemo emaile vsem volilcem. Ko se volitve zaključijo, sistem

izračuna rezultate in jih preko emaila pošlje tudi vsem volilcem. Vsak volilec si lahko nato ogleda rezultate in preveri, da je bil njegov glas res upoštevan.

Kot volilec najprej dobimo obvestilo preko emaila, da lahko sodelujemo v določenih volitvah. Nato gremo na ustrezno spletno stran, se prijavimo npr. S pomočjo Google avtentikacije in sodelujemo v volitvah. Ko odgvorimo na vsa vprašanja, nam sistem pošlje še email z dodatnim geslom za potrditev glasovanja. Ko vnesemo še to, lahko dokončno potrdimo svoje glasovanje. Po končanem glasovanju in razkritju rezultatov pravtako dobimo email obvestilo s povezavo za ogled le teh.



Glasovanje

Andrej voting andrej@moebius.si
to me

Dear Andrej Slapnik,

Sodeluj v anketi

Election URL: http://www.moebius.si/helios/e/aleksandarjurisic/vote
Election Fingerprint: IG31yprOa+caB3oNh82B9clunXtcZTY0zuhxolLlnko

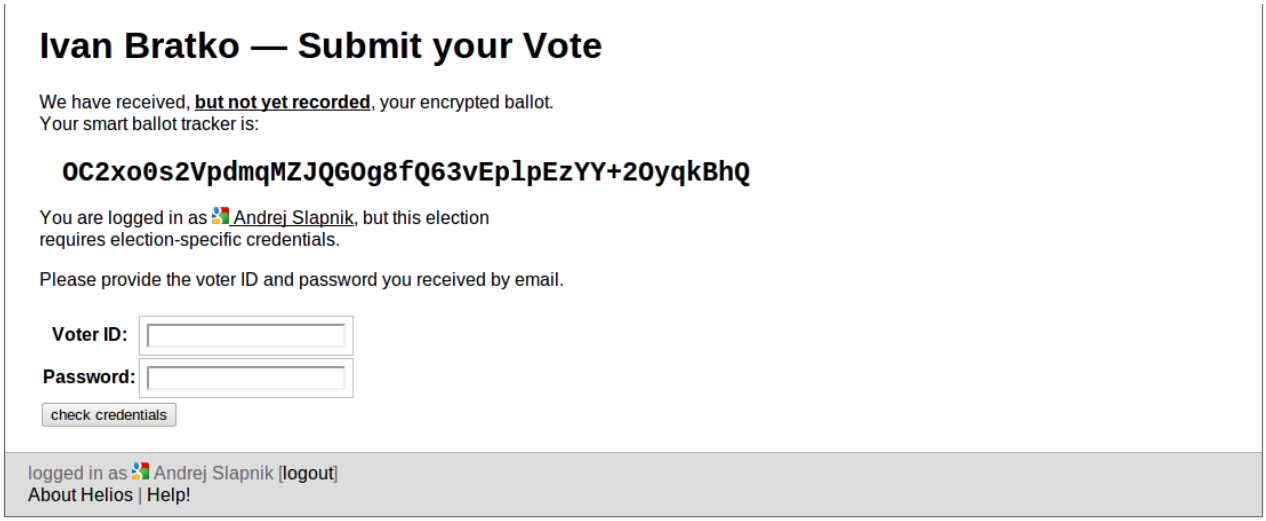
Your voter ID: andy
Your password: U2pY3EQybm

In order to protect your privacy, this election is configured
to never display your voter login ID, name, or email address to the public.
Instead, the ballot tracking center will only display your alias.

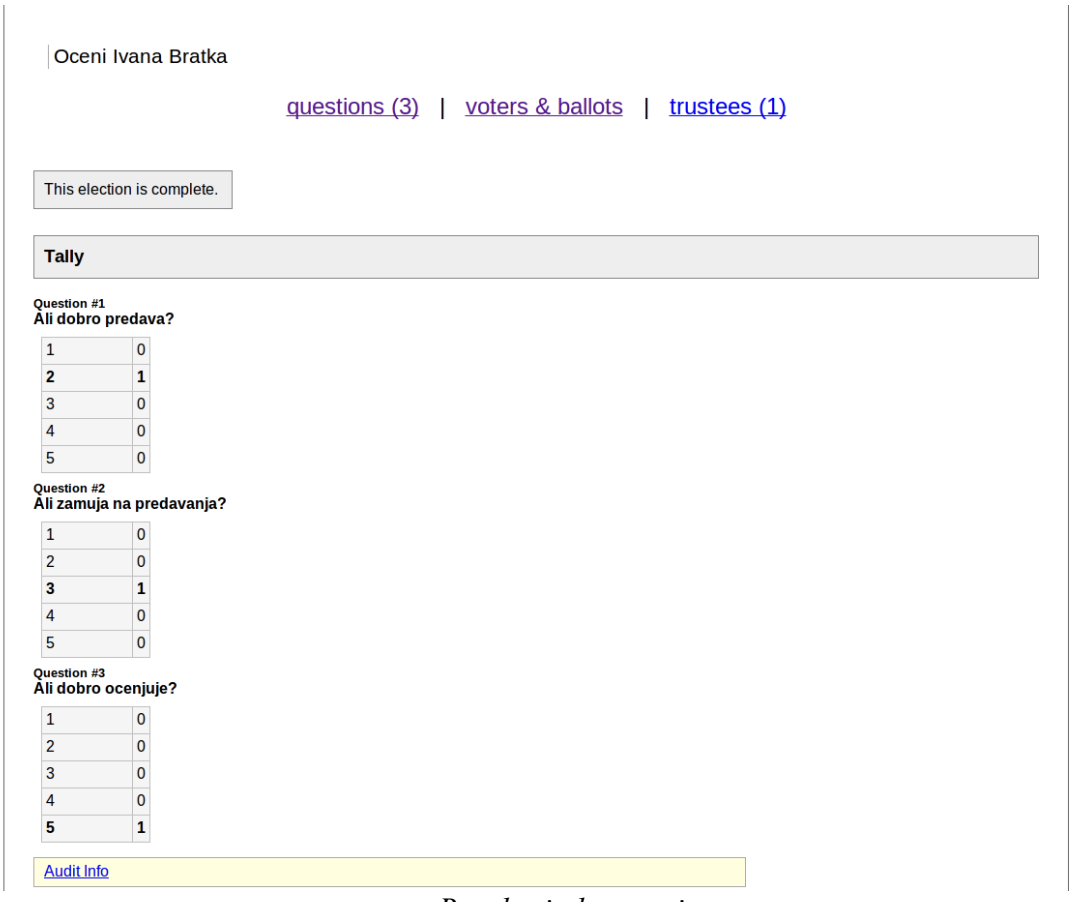
Your voter alias is V1.

IMPORTANTLY, when you are prompted to log in to vote,
please use your *voter ID*, not your alias.

Vsebina emaila, ki ga dobimo za dodatno preverjanje



Dodatno preverjanje ob koncu glasovanja



Rezultati glasovanja

6. Zaključek

Motivacija za izdelavo te seminarske naloge je bilo dosedaj dokaj neučinkovito zastavljeno ocenjevanje profesorjev. Ko smo se poglobili v iskanje elektronske rešitve, spoznavanje že poznanih sistemov volitev in anket, smo spoznali, da elektronska izvedba le teh ni tako trivialna. Za rešitev našega problema smo uporabili in prilagodili odprtokodno rešitev Helios. Postavili smo ga na namenski strežnik, poskrbeli za ustrezno programsko podporo, certifikate, varno povezavo in zaščitili strežnik pred vdori. Za prikaz delovanja ocenjevanja profesorjev smo postavili nekaj "anket", le te izvedli in preverili rezultate. Preverili smo tudi delovanje vseh potrebnih lastnosti (anonimizacija, avtentikacija, anonimno shranjevanja v podatkovno bazo, …) ter tako potrdili pravilno in varno delovanje. Z nekaj programskimi in dizajnerskimi prilagoditvami bi lahko naslednje ocenjevanje ali anketo izvedli s sistemom Helios tudi na naši fakulteti.

7. Viri

[1] Andrej Tolič, *diplomska naloga Kriptografija e-volitev, 2011*

[2] Chou-Chen Yang, Ching-Ying Lin, Hung-Wen Yang: *Improved secured e-voting over a network, E-Government and Security of Information No. 2*

[3] *Helios install documentation, http://documentation.heliosvoting.org/install, december 2011*

[4] *How to use django with apache and mod wsgi, https://docs.djangoproject.com/en/1.2/howto/deployment/modwsgi/, december 2011*

[5] Security Criteria for Electronic Voting, *http://www.csl.sri.com/users/neumann/ncs93.html, december 2011*

[6] *Modwsgi django integration, http://code.google.com/p/modwsgi/wiki/ IntegrationWithDjango, december 2011*

[7] *Libapache2-mod-wsgi info, http://packages.debian.org/unstable/python/libapache2-mod-wsgi, december 2011*

8. Priloge

Nginx nastavitve

```
upstream django {
      server www.moebius.si:9000;
}

server {
      listen 443;
      server_name www.moebius.si moebius.si;
      root /var/www/helios/helios-server/helios;
      access_log /var/log/nginx/access.log;
      error_log /var/log/nginx/error.log;
      ssl on;
      ssl_certificate /etc/nginx/ssl/certs/helios.pem;
      ssl_certificate_key /etc/nginx/ssl/certs/helios.key;
      ssl_prefer_server_ciphers on;
      try_files $uri @django;

      location @django{
            proxy_pass http://django;
            proxy_redirect off;
            proxy_set_header Host $host;
            proxy_set_header   X-Real-IP        $remote_addr;
            proxy_set_header   X-Forwarded-For  $proxy_add_x_forwarded_for;
            proxy_set_header   X-Forwarded-Protocol https;
      }
}

server {
      set $base /var/www;
      listen       80;
      server_name  www.moebius.si moebius.si;
      root /var/www/helios/helios-server/helios;
      access_log  /var/log/nginx/access.log;
```

```
        error_log   /var/log/nginx/error.log;
        # This will compress the content to increase page loading speed
        gzip on;
        gzip_types text/plain application/xml text/css text/java-script;
        try_files $uri @django;
        location @django {
                proxy_pass http://django;
                proxy_redirect off;
                proxy_set_header Host $host;
                proxy_set_header   X-Real-IP        $remote_addr;
                proxy_set_header   X-Forwarded-For  $proxy_add_x_forwarded_for;
        }
}
```

Apache nastavitve

```
<VirtualHost *:9000>
        ServerName www.moebius.si
        ServerAlias *.moebius.si
        ServerAdmin webmaster@domain.com
        ErrorLog /var/log/apache2/domain.com.log
        SetEnvIf X-Forwarded-Protocol "^https$" HTTPS=on
        WSGIDaemonProcess domain display-name=%{GROUP} maximum-requests=10000
        WSGIProcessGroup domain
        WSGIScriptAlias / /var/www/helios/helios-server/django.wsgi

        <Directory /var/www/helios/helios-server>
                Order deny,allow
                Allow from all
        </Directory>
</VirtualHost>
```