

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Čater

**ZAJEM »MODROSTI MNOŽIC« S SPLETNIM
ANKETIRANJEM**

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Ljubljana, 2011

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Čater

**ZAJEM »MODROSTI MNOŽIC« S SPLETNIM
ANKETIRANJEM**

DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: prof. dr. Marko Robnik Šikonja

Ljubljana, 2011



Št. naloge: 00044/2010

Datum: 05.10.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **PRIMOŽ ČATER**

Naslov: **ZAJEM »MODROSTI MNOŽIC« S SPLETNIM ANKETIRANJEM
CAPTURING »WISDOM OF THE CROWDS« WITH WEB SURVEYS**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Za pridobivanja informacij o nekaterih vprašanjih se vse bolj uporablja princip "modrosti množic", ki predvideva, da je pri odgovorih na mnoga vprašanja povprečje množice odgovorov natančnejše kot posamezniki in celo boljše od strokovnjakov. Želimo preveriti veljavnost tega principa. V ta namen potrebujemo zanimiva in relevantna vprašanja in ustrezen spletni portal, ki bo omogočal zajem odgovorov in bo motiviral udeležence. Zasnujte in implementirajte spletni portal, ki bo administratorjem omogočal postavljati vprašanja z različnih področij in bo omogočal zanesljiv zajem odgovorov. Definirajte dostopno shemo z uporabniki in več tipi administratorjev ter realizirajte portal na osnovi podatkovne baze. Prikaz rezultatov naj vsebuje statistično analizo odgovorov, kakor tudi uspešnosti posameznih uporabnikov in administratorjev.

Mentor:


doc. dr. Marko Robnik Šikonja

Dekan:


prof. dr. Nikolaj Zimic



IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Primož Čater,
z vpisno številko 63000019,

sem avtor diplomskega dela z naslovom:

Zajem »modrosti množic« s spletnim anketiranjem

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
prof. dr. Marka Robnik Šikonje
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.)
ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 12.3.2011

Podpis avtorja:

Zahvala

Zahvaljujem se prof. dr. Marku Robnik Šikonji za vse koristne nasvete ter strokovno vodeno pomoč pri nastajanju tega diplomskega dela.

KAZALO

1 UVOD	3
2 MODROST MNOŽIC	4
2.1 POGOJI ZA MODROST MNOŽIC	5
2.2 PRIMERI MODROSTI MNOŽIC	7
2.2.1 Lepo je biti milijonar	7
2.2.2 Napovejte število bonbonov v skledi	9
2.3 MODROST MNOŽIC V INFORMATIKI	13
2.3.1 Google PageRank	13
3 SPLETNI PORTAL	15
3.1 UPORABA PORTALA	16
3.2 BAZA PODATKOV	18
3.3 REALIZACIJA ANKET IN GLASOVANJA	20
3.3.1 Vrste anket.....	21
3.4 GD KNJIŽNJICA.....	23
3.5 REGISTRACIJA UPORABNIKA.....	25
3.5.1 JQuery	26
3.6 FORUM.....	29
4 TEHNOLOGIJE	30
4.1 AJAX.....	30
4.2 CAPTCHA	32
4.3 CSS.....	34
5 ZAKLJUČEK	35
6 KAZALO SLIK	37
7 SEZNAM VIROV IN LITERATURE.....	38

SEZNAM KRATIC

AJAX – Asynchronous Javascript and XML (asinhroni Javascript in XML)

CAPTCHA - Completely Automated Public Turing test to tell Computers and Humans Apart
(test, ki ugotovi, ali je določen uporabnik človek ali program)

CSS – Cascading Style Sheets (predloge, ki določajo izgled spletnih strani)

DPT– Diversity Prediction Theorem (teorem različnih napovedi)

HTML – HyperText Markup Language (jezik za označevanje besedila)

HTTP – HyperText Transfer Protocol (protokol za prenos informacij na spletu)

JS – JavaScript (skriptni jezik za pomoč pri ustvarjanju interaktivnih spletnih vsebin)

PHP – HyperText Preprocessor (programski jezik za razvoj dinamičnih spletnih vsebin)

PR – PageRank algoritem

POVZETEK

Cilj diplomskega dela je implementacija spletnega portala za testiranje teorije modrosti množic, ki pravi, da je kolektivna odločitev več ljudi boljša od odločitev posameznikov. Modrost množic je zelo zanimiv koncept, ki pod določenimi pogoji dobro deluje tudi v praksi, in je lahko odličen način pridobivanja verodostojnih informacij. Spletni portal uporabnikom omogoča enostavno in preprosto pridobivanje podatkov in informacij v zvezi z njihovo problematiko. Pridobivanje informacij je realizirano v obliki spletnih anket, ki jih uporabnik zastavlja, lahko pa glasuje tudi v anketah, ki so jih postavili drugi uporabniki. Realizirali smo tudi ocenjevanje kakovosti uporabnikov, ki je povezano s točnostjo glasovanja pri vprašanih drugih uporabnikov. Ocenjevanje določa tudi ankete, ki se bodo pojavile na prvi strani portala in bodo s tem deležne največje pozornosti.

Implementirali smo grafično predstavitev končnih rezultatov v obliki diagrama oz. krožnega diagrama in krožnega prikaza. Ankete so sortirane po kategorijah s podobno vsebino, vse zaključene ankete in njihovi rezultati pa so na voljo v arhivu anket, ki skrbi tudi zato, da se ankete ne ponavljajo. Večina spletnega portala je implementirana s spletnim skriptnim jezikom PHP in podporo MySQL. Za privlačnejši videz in kakovost storitev skrbijo spletne tehnologije kot so JQuery, Javascript, Ajax in CSS.

Spletni portal smo obogatili še z forumom, kjer si registrirani uporabniki delijo in izmenjujejo mnenja, oddajajo kritike, pohvale in nasvete.

Ključne besede: PHP, MySQL, spletne aplikacije, spletni portal, ankete, pridobivanje informacij, modrost množic

ABSTRACT

We implemented a web portal for testing the wisdom of the crowds theory. This theory states that collective wisdom is better than knowledge of a single person. The wisdom of crowds is an interesting concept, which in practice works well under certain circumstances and can be an excellent way of gathering information. Web portal offers its users an easy and simple way of gathering data and information in a form of web surveys. Users can create their own surveys and can vote in surveys of other users. We implemented user ratings, which are calculated from their correct answers at surveys. User ratings determine which surveys will be shown on first page of portal and thus receive the most attention.

We also implemented a graphic presentation of final scores in shape of diagram and pie chart . Surveys are sorted in categories by similarity of content. All closed surveys and their final results are available in an archive, which also prevent repetition of the questions.

Most of the web portal was implemented using PHP script language and with MySQL database. For attractive design and quality of services we applied web technologies JQuery, Javascript, Ajax and CSS.

We enhanced our web portal with forum, where users can share and exchange ideas and opinions, express criticism, praises and advices.

Keywords: PHP, MySQL, web applications, web portal, survey, information gathering, wisdom of the crowds

1 UVOD

V današnjem svetu so informacije ključnega pomena. Kljub njihovem preobilju je včasih pot do pomembne informacije vse prej kot lahka. Ker za marsikatero odločitev osebne izkušnje in zdrava pamet nista dovolj, se obračamo k drugim virom, kjer upamo, da bomo našli verodostojen odgovor.

Modrost množic je koncept, ki pravi, da, ob izpolnitvi določenih pogojev, skupina ljudi pride do boljše skupne odločitve, kot če bi za mnenje povprašali posameznike ali celo strokovnjake. Ljudje, ki sestavljajo množico, so v večini primerov čisto navadni ljudje, ki imajo omejeno količino znanja, rezultati skupine pa so vseeno dobri.

Modrost množic je v povezavi z internetom še močnejša, saj spletni medij zelo pospeši pridobivanje informacij. S spletno implementacijo modrosti množic lahko preskočimo marsikatero organizacijsko oviro. Tako si olajšamo izbor posameznikov, ki tvorijo množico, poenostavimo vnose podatkov, izračunavanja in analize rezultatov ter prikaz rezultatov.

Da bi boljše razumeli realizacijo spletnega portala, si moramo za začetek pogledati, kaj internet kot medijski fenomen sploh je. Vse skupaj se je začelo konec 60-ih let, ko je internet začel delovati kot ARPANET za potrebe ameriške vojske. Sčasoma se je omrežje razvijalo in je postalo zanimivo tudi za komercialne namene. Leta 1983 je ARPANET spremenil jedro svojih omrežnih protokolov iz NPC (Network Control Program) v TCP/IP (protokol za nadzor prenosa) protokol, ki ga poznamo in uporabljamo danes. Prvotno so bile spletne strani statične in so bile zasnovane na podlagi HTML-ja. Zadovoljevale so ciljno skupino uporabnikov, ki so ga uporabljali za prebiranje člankov in hitro ter pregledno brskanje po internetu s pomočjo povezav (*angl. linkov*).

Z razvojem interneta so rasle težnje uporabnikov k večji interaktivnosti med uporabnikom in spletnim mestom, kar je vzpodbudilo razvoj velikega števila spletnih orodij in jezikov, s katerimi so se nam odprla vrata v svet dinamičnih spletnih strani, brez katerih si svetovnega omrežja danes ne moramo več predstavljati. Internet je tako postal največji medij, ki omogoča ljudem bliskovit dostop do informacij, podatkov in interakcij med ljudmi, zato je njegova uporaba vsesplošna. Naš namen je združiti internet in modrosti množic ter ustvariti spletni portal, kjer uporabniki z minimalno količino vložene energije, časa in denarja pridejo do verodostojnih informacij, ki jim pomagajo pri njihovih odločitvah.

V drugem poglavju smo opisali teoretični koncept modrosti množic. Podali smo principe delovanja in pogoje za uspešno delovanje množice. Vse skupaj smo podkrepili še z praktičnimi primeri in metodami za njihovo reševanje. Tretje poglavje opisuje implementacijo in primere uporabe spletnega portala. Poglobljeno smo opisali vrste anket, potek glasovanja, ter delovanje registracije uporabnikov. Predstavili smo tudi forum, ki deluje kot skupnost uporabniških idej in mnenj, ter nekaj algoritmov, ki sestavljajo spletni portal. Pri četrtem poglavju smo navedli in opisali principe delovanja tehnologij Ajax in CAPTCHA, ter nekaj osnovnih besed o CSS. Tema petega poglavja so možne dodatne funkcije, ki bi še izboljšale funkcionalnost portala in primeri, v katerih modrost množic ne pride do zelenih rezultatov.

2 MODROST MNOŽIC

Študije so pokazale, da se, pod določenimi pogoji, koncept modrosti množic odlično odnese v praksi, saj so odločitve množic boljše od odločitev strokovnjakov. Modrost množic v širšem smislu se je v zadnjem desetletju močno razvila v računalniški industriji. Skorajda si ne moremo več predstavljati nekaterih spletnih aplikacij brez kolektivnega odločanja. Googlov iskalni algoritem PageRank in Amazonovo ocenjevanje izdelkov sta samo dva izmed primerov spletnih aplikacij, katerih delovanje temelji na ocenjevanju množic.

Lep primer spletnega portala, ki temelji izključno na odločitvah množice je tudi Digg.com, kjer uporabniki sami odločajo o usodi povezav. Uporabniki objavljajo povezave do spletnih člankov, novic, slik in videoposnetkov, ki so razporejeni po kategorijah, hkrati pa imajo možnost o njih tudi glasovati. Če jim je povezava všeč, jo izkopljejo (*angl. digg*), če pa se jim zdi neverodostojna oz. pristranska, jo zakopljejo. Tako se na naslovnici pojavijo največkrat izkopani članki.

The screenshot shows the Digg.com homepage. At the top, there's a navigation bar with 'My News', 'Top News', and 'Upcoming' tabs. Below this, there are filters for 'Most Recent', 'Top in 24 Hours', '7 Days', and '30 Days', along with a 'Submit a link' input and a 'Digg it' button. A search bar is also present. The main content area is titled 'Top News' and features a list of articles with their respective Digg counts and titles. The sidebar on the left lists various categories like Business, Entertainment, Gaming, etc. The right sidebar includes 'Hot Stories on Digg' and 'Top News in All Topics'.

Top News

- 65** **Real Life Superwomen Solve Crimes and Save Lives**
newsweek.com — Take Ann Timson, the 71-year-old "Supergran" who singlehandedly defused a jewelry-store heist, armed with only her oversize purse. Here's... 5 min ago
via Newsweek 0 Comments Save
- 109** **ABC, CBS, CNN, FOX, NBC All Want Your Chemtrail Photos | Conscious Life News**
consciouslifenews.com — Instructions for everyone on this planet, who is aware of "chemtrails", to flood the U.S. media regarding the heinous crime being committ... 35 min ago
via mysticdave 14 Comments Save Bury
- 2766** **What's Slowing Down Your PC?**
howlifeworks.com — How Life Works shows you the two most common issues and offers easy solutions.
Sponsored by HowLifeWorks Bury
- 92** **We Spent Half a Million Dollars to Fly Planes Over the Super Bowl**
good.is — The Navy spent almost \$500,000 in taxpayer money to fly jets over this year's Super Bowl. It only took them three seconds. 1 hr 25 min ago

Hot Stories on Digg

- 43** **Social? Really? Folks - It's About the Data**
readwriteweb.com
- 54** **New Study says 'Niceness' Is Partly Genetic**
Chosen by digg staff. Learn more.

Top News in All Topics

Slika 1: Spletni portal Digg.

2.1 POGOJI ZA MODROST MNOŽIC

Modrost množic ni popoln sistem, na katerega bi se lahko naslanjali ob vsaki dilemi. Da bi množica zagotavljala uspešne rezultate, so potrebni naslednji pogoji [6,7]:

Enostavnost – pomembna je enostavnost naloge. Čeprav je množica sposobna rešiti še tako kompleksen problem, je zelo pomembno, da se ta razbije na manjše, preprostejše komponente, kar omogoča jasno razumevanje problema.

Uporabniški vmesnik – kompleksni vmesniki so dobri za zapletene storitve, vendar to ne drži za modrost množic. Bolj kompleksen kot je uporabniški vmesnik, manj udeležencev lahko pričakujemo. Vmesniki naj bodo izbirni mehanizmi v obliki anket ali vseh/ni mi vseh odločitve, lahko pa zahtevajo tudi minimalen vnos rezultatov. Hitrost in jasnost sta tu ključnega pomena.

Udeleženosť – je eden od parametrov, ki definirajo modrost množic, in pravi, da več kot je udeležencev, pametnejša je množica.

Decentralizacija in neodvisnosť – ljudje, ki skupino sestavljajo, morajo delovati med sabo kar se da neodvisno. Zanašati se morajo na svoje lastno znanje in sposobnosti, saj obstaja nevarnosť, da se pri človeških množicah izmenjujejo mnenja oz. ideje, ki jih posameznik v primeru decentralizacije ne bi zagovarjal. Napačna mnenja se širijo, ker jih je izrekel nekdo, ki lahko vpliva na glasujočo osebo. Dokler se takšni primeri dogajajo naključno brez centralizacije, jih skupinska odločitev množice brez težav izloči. Če pa se na enak način moti večina, ki glede konkretnega vprašanja ni več skupek neodvisnih posameznikov, ampak homogena celota, skupno mnenje takšnih napak ne odpravi več.

Sebičnost – čeprav je v večini primerov v skupinah sebičnost neproduktivna, so množice najbolj pametne takrat, ko so sestavljene iz posameznikov, ki mislijo predvsem na svoje potrebe in ne na potrebo celotne skupine. Posamezniki morajo biti »sebične narave«. Če začnejo postavljati cilje skupine pred osebnimi cilji, je to pri tem konceptu negativno.

Raznolikost – vsak posameznik mora imeti na zastavljeno vprašanje svoj lasten pogled, vire, informacije in znanja, ki zagotavljajo raznolikost odgovorov. Modrost množic deluje prav zato, ker se skrajna mnenja posameznikov izničijo. Če je preveč udeležencev usmerjeno v skrajnosť, bodo rezultati napačni.

Združevanje – za interpretacijo vseh odgovorov množice je potreben mehanizem, ki skrbi, da se vsi rezultati združijo v končno skupinsko odločitev. Potrebujemo združevalca in algoritem, po katerem se združevanje odvija. V primeru »Napovej število bonbonov v skledi« je združevalec oseba, ki zbira podatke udeležencev, algoritem pa je mediana odgovorov. Pri Googlovem iskanju zadetkov je združevalec Google, njihov algoritem, ki se imenuje PageRank, pa rangira spletne strani.

Modrost množic so torej individualne odločitve posameznikov v neki skupini, brez medsebojnega posvetovanja. V primeru, da se množica drži tega pogoja, obstaja velika možnost, da je kolektivna odločitev natančna in pravilna.

Zanimivo je, da so končni rezultati množice večinoma boljši od rezultata najboljšega posameznika. V večini primerov bosta eden ali dva posameznika pametnejša od skupine, kar pa ne pomeni, da bosta ta posameznika vedno dosegala boljše rezultate od skupine. Povedano drugače, če naredimo določen poizkus desetkrat, je velika verjetnost, da bosta ena ali dve osebi dosegali boljše rezultate od skupine, vendar bo to veljalo samo za določen poizkus. Pri naslednjem poizkusu bosta pametnejši od skupine drugi dve osebi. Tako se pri več poizkusih skupina vedno izpostavi kot najpametnejša. Kot dokaz, da teorija modrosti množic deluje, si oglejmo še nekaj praktičnih primerov.

2.2 PRIMERI MODROSTI MNOŽIC

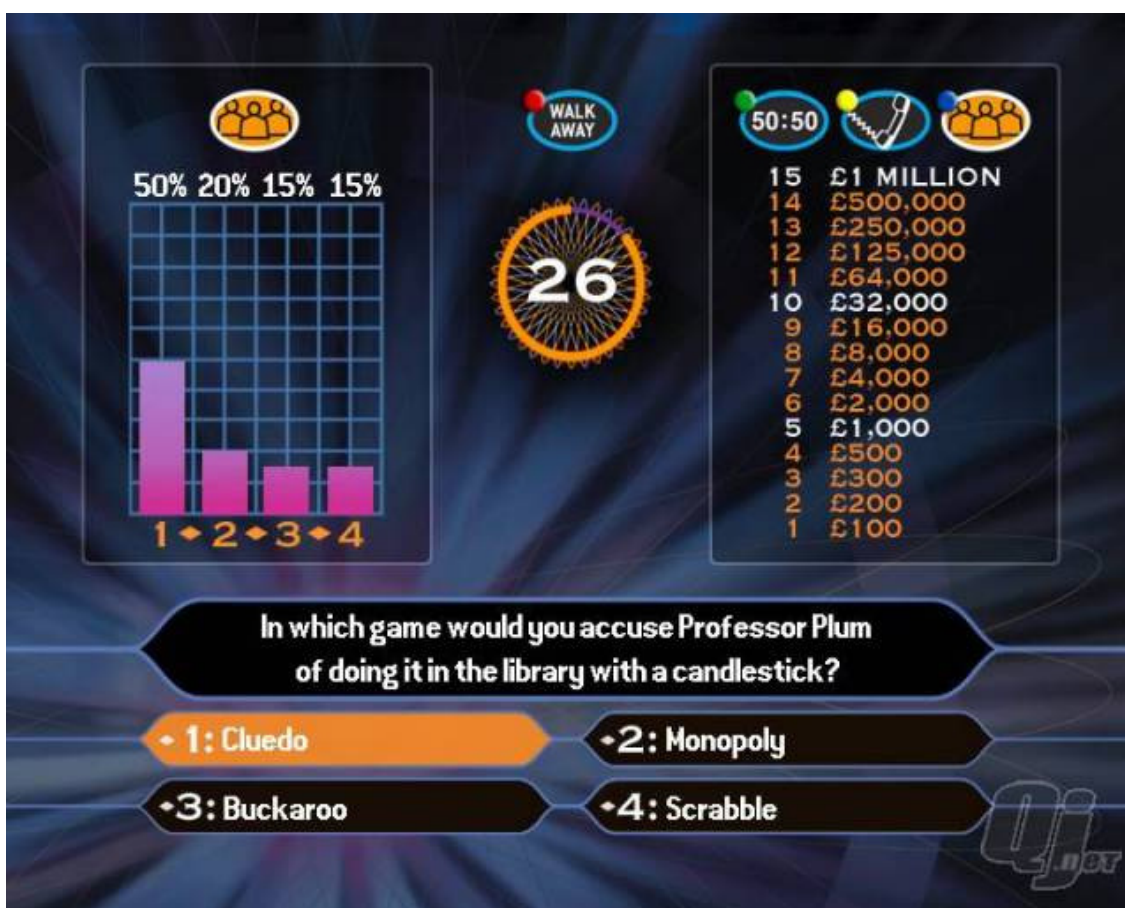
2.2.1 Lepo je biti milijonar

Modrost kolektivnih odločitev je lepo ponazorjena v popularni televizijski oddaji »Lepo je biti milijonar« (angl. *Who want's to be a millionaire?*). Naloga tekmovalca je, da pravilno odgovori na čim več zastavljenih vprašanj, med katerimi dobi za vsako vprašanje štiri že vnaprej pripravljene odgovore (eden pravilen, trije pa so napačni) in si, v skladu s številom pravih odgovorov, prisluži finančno nagrado. Da mu tekmovanje olajšajo, ima tekmovalec na voljo tri dodatne pomoči:

Polovička – izmed štirih podanih odgovorov se izločita dva.

Klic v sili – tekmovalec lahko pokliče osebo, za katero misli, da lahko pravilno odgovori na njegovo vprašanje. Oseba na drugi strani zveze lahko uporablja vse izvore informacij, ki jih ima na voljo.

Glas ljudstva – tekmovalec za pravilni odgovor povpraša prisotno občinstvo, ki glasuje za enega izmed štirih odgovorov po lastni presoji.



Slika 2: Primer glasovanja ljudstva.

Čeprav gre za vprašanja, katerih točen odgovor pozna zelo malo ljudi, pa glas ljudstva opravi svojo nalogo pravilno v skoraj neverjetnih 91% primerov. Za primerjavo, če se tekmovalec odloči za pomoč »klic v sili«, kjer dobi mnenje strokovnjaka oz. osebe, ki lahko razpolaga z neomejeno količino virov in informacij, je rezultat solidnih 65% [5].

Kako lahko razložimo takšno uspešnost kolektivnega razmišljanja? Kako lahko razložimo pojav, pri katerem velika večina ljudi dejansko ne ve točnega odgovora, skupina pa je vseeno uspešna v čez 90% primerov? Ena izmed možnih razlag je teorija *Condorcet Jury Theorem (CJT)*, ki jo je razvil Marquis de Condorcet v svojem delu »*Essay on the Application of Analysis to the Probability of Majority Decisions*« davnega leta 1785 [5]. CJT pravi, da če ima povprečen udeleženec skupine več kot 50% možnosti, da pozna pravilen odgovor, in če so odgovori podani na izbiro, verjetnost pravilnega odgovora raste proti 100%, ko se število članov skupine povečuje.

Zadevo si oglejmo na konkretnem primeru:

Množici ljudi zastavimo vprašanje »Kdo od navedenih ni bil član britanske zasedbe The Monkeys, ki je delovala v 60ih letih?«

Odgovori:

- a) Peter Tork
- b) Davy Jones
- c) Roger Noll
- d) Michael Nesmith

Pravilen odgovor se glasi c) Roger Noll. Ker gre za relativno težko vprašanje, velika večina posameznikov pravilnega odgovora ne pozna. Anketirana je množica 100 ljudi, katerih poznavanje skupine *The Monkeys* ali Roger Nolla je naslednje:

- 7 ljudi pozna vse 3 člane skupine The Monkeys,
- 10 ljudi pozna 2 člana skupine The Monkeys,
- 15 ljudi pozna 1 člana skupine The Monkeys,
- 68 ljudi ne pozna članov skupine.

Povedano drugače, manj kot 10% anketirancev pozna točen odgovor, medtem ko več kot 2/3 glasovalcev ne ve, kateri odgovor je pravilen. Predpostavimo, da posamezniki, ki odgovora ne poznajo, odgovarjajo naključno. CJT tukaj ne velja, ker pravilni odgovor pozna malo glasovalcev.

Poglejmo analizo odgovorov posameznikov:

- 7 ljudi bo očitno glasovalo za Rogerja Nolla,
- 5 od 10ih, ki poznajo 2 člana The Monkeys bo glasovalo za Rogerja Nolla,
- 5 od 15ih, ki poznajo 1 člana The Monkeys bo glasovalo za Rogerja Nolla,
- 17 od 68, ki glasujejo naključno, bo glasovalo za Rogerja Nolla.

Rezultati ankete »Kdo ni bil član britanske zasedbe The Monkeys?«

- a) Peter Tork z 22% glasov
- b) Davy Jones z 22% glasov
- c) Roger Noll z 34% glasov
- d) Michael Nesmith z 22% glasov

Kot je moč razbrati iz rezultata ankete je izbira skupine odgovor c) Roger Noll z 34% glasov. Teorija modrosti množic deluje, ker se naključni napačni odgovori med sabo izničijo in pravilen odgovor pride na površje. Tudi v primeru, da bi še več ljudi glasovalo na pamet, rezultata ankete ne bi spremenili, zmanjšala bi se le dominantnost pravega odgovora.

Zakaj kolektivno znanje ni 100% uspešno in je »le« 91%?

Pri modelu modrosti množic sta pomembni predvsem dve spremenljivki:

- procent ljudi, ki pozna dejanski odgovor,
- stopnja naključnosti pri odgovorih.

Delež ljudi s pravilnim odgovorom bo vedno relativno majhen in na končen rezultat nima bistvenega vpliva. Količina naključnih odgovorov je vedno velika. Varianca v naključnosti odgovorov posameznikov je vzrok, da kolektivni odgovori niso pravilni v 100% primerov.

2.2.2 Napovejte število bonbonov v skledi

Analizirali smo problem, kjer množica izbira med posameznimi podanimi odgovori in pokazalo se je, da kolektivne odločitve dosegajo odlične rezultate. Kako pa se obnesejo množice v primeru, kadar odgovori niso podani? Poglejmo si, kako uspešna je skupina pri ocenjevanju rezultata.

»Napovejte število bonbonov v skledi« je naslednji tip ankete, ki ga bomo analizirali in je prikazan na sliki 3. Pri tej anketi morajo posamezniki uganiti število bonbonov v prozorni skledi. Ker je bonbonov veliko, je naloga vse prej kot enostavna. Kot se izkaže, množica tudi tokrat postreže z neverjetno natančnim skupnim odgovorom.

Razlaga uspeha leži v tako imenovani *Diversity Prediction Theorem* (teorem različnih napovedi) (DPT) [5], ki ima nekaj bistvenih lastnosti:

- natančnost množice bo v napovedovanju boljša kot njen povprečni član. Z drugimi besedami: množica napoveduje natančneje od posameznikov v njej,
- v večini primerov je množica boljša celo od njenega najboljšega člana,
- sposobnost skupnega napovedovanja je odvisna od natančnosti in raznolikosti odgovorov.

Kolektivno napako zmanjšamo s povečanjem natančnosti na enoto ali pa s povečanjem raznolikosti na enoto. Po DPT lahko izračunamo kolektivno napako po sledeči formuli:

Kolektivna napaka = povprečno odstopanje posameznikov – povprečna napaka glede na povprečni odgovor

Kot povprečno odstopanje posameznikov razumemo povprečje vseh absolutnih odstopanj posameznih napovedovalcev od pravilnega rezultata. Povprečna napaka glede na povprečni rezultat pomeni, koliko se je napovedovalec zmotil glede na povprečni rezultat skupine.

Primer:

Množica na anketo odgovori s **povprečjem 1.151** bonbonov v skledi, medtem ko je **pravi rezultat 1.116**. Množica se zmoti za 3.1 % in od 73 udeležencev v množici, sta 2 bila boljša od kolektivnega odgovora celotne množice.

Na razpolago imamo naslednje podatke:

- natančno število bonbonov = 1.116
- povprečni odgovor na št. bonbonov = 1.151
- povprečno odstopanje posameznikov (kvadrirano) = 490.949
- povprečna napaka od povprečnega rezultata (kvadrirano) = 489.692

Student	Rezultat	Razlika od pravega rezultata	Kvadrirana razlika od pravega rez	Razlika od povprečnega rezultata	Kvadrirana razlika od povprečnega rezultata
1	250	-866	749,956	-901	811,801
2	315	-801	641,601	-836	698,896
3	399	-717	514,089	-752	565,504
4	400	-716	512,656	-751	564,001
5	420	-696	484,416	-731	534,361
6	437	-679	461,041	-714	509,796
7	479	-637	405,769	-672	451,584
8	500	-616	379,456	-651	423,801
9	540	-576	331,776	-611	373,321
10	585	-531	281,961	-566	320,356
11	600	-516	266,256	-551	303,601
12	600	-516	266,256	-551	303,601
13	604	-512	262,144	-547	299,209
14	616	-500	250,000	-535	286,225
15	624	-492	242,064	-527	277,729
16	632	-484	234,256	-519	269,361
17	645	-471	221,841	-506	256,036
18	650	-466	217,156	-501	251,001
19	651	-465	216,225	-500	250,000
20	699	-417	173,889	-452	204,304
21	721	-395	156,025	-430	184,900
22	723	-393	154,449	-428	183,184
23	734	-382	145,924	-417	173,889
24	750	-366	133,956	-401	160,801
25	750	-366	133,956	-401	160,801
26	750	-366	133,956	-401	160,801
27	750	-366	133,956	-401	160,801
28	768	-348	121,104	-383	146,689
29	780	-336	112,896	-371	137,641
30	800	-316	99,856	-351	123,201
31	800	-316	99,856	-351	123,201
32	820	-296	87,616	-331	109,561
33	850	-266	70,756	-301	90,601
34	874	-242	58,564	-277	76,729
35	876	-240	57,600	-275	75,625
36	900	-216	46,656	-251	63,001
37	900	-216	46,656	-251	63,001
38	900	-216	46,656	-251	63,001
39	1,000	-116	13,456	-151	22,801
40	1,000	-116	13,456	-151	22,801
41	1,008	-108	11,664	-143	20,449
42	1,120	4	16	-31	961
43	1,120	4	16	-31	961
44	1,152	36	1,296	1	1
45	1,234	118	13,924	83	6,889
46	1,234	118	13,924	83	6,889
47	1,250	134	17,956	99	9,801
48	1,250	134	17,956	99	9,801
49	1,260	144	20,736	109	11,881
50	1,288	172	29,584	137	18,769
51	1,300	184	33,856	149	22,201
52	1,400	284	80,656	249	62,001
53	1,500	384	147,456	349	121,801
54	1,500	384	147,456	349	121,801
55	1,500	384	147,456	349	121,801
56	1,523	407	165,649	372	138,384
57	1,564	448	200,704	413	170,569
58	1,575	459	210,681	424	179,776
59	1,580	464	215,296	429	184,041
60	1,583	467	218,089	432	186,624
61	1,588	472	222,784	437	190,969
62	1,700	584	341,056	549	301,401
63	1,732	616	379,456	581	337,561
64	1,872	756	571,536	721	519,841
65	1,896	780	608,400	745	555,025
66	1,899	783	613,089	748	559,504
67	1,963	847	717,409	812	659,344
68	2,000	884	781,456	849	720,801
69	2,250	1,134	1,285,956	1,099	1,207,801
70	3,000	1,884	3,549,456	1,849	3,418,801
71	3,000	1,884	3,549,456	1,849	3,418,801
72	3,024	1,908	3,640,464	1,873	3,508,129
73	4,100	2,984	8,904,256	2,949	8,696,601
Povprečje	1,151		490,949		489,692

Slika 3: Glasovanje množice o številu bonbonov v skledi[5].

Slika 3 prikazuje ugibanje pravilnega rezultata skupine, ki je sestavljena iz 73 posameznikov. Kot vidimo, se rezultati razlikujejo od posameznika, ki je napovedal najmanj (250 bonbonov), do posameznika, ki je napovedal največ (4.100 bonbonov), za neverjetnih 3.850 bonbonov. Da pridemo do skupnega rezultata uporabimo:

Kolektivna napaka = povprečno odstopanje posameznikov – povprečna napaka od povprečnega rezultata

Povprečno odstopanje posameznikov in povprečno napako od povprečnega rezultata kvadriramo zato, da se znebimo negativnega predznaka in dobimo:

Povprečno odstopanje posameznikov = 490.949
 Povprečna napaka od povprečnega rezultata = 489.692

Kolektivna napaka = povprečno odstopanje posameznikov - povprečna napaka od povprečnega rezultata

Ko odštejemo povprečni vrednosti je vrednost kolektivne napake 1258. Rezultat korenimo in dobimo končno vrednost kolektivne napake, ki znaša 35,4.

Dobljeni rezultat ≈ 35 odgovarja formuli:

ABS (povprečno ugibanje množice - natančno število bonbonov)

ABS(1151-1116) = 35

2.3 MODROST MNOŽIC V INFORMATIKI

V računalniški industriji se je modrost množic začela uporabljati z razvojem interneta. Na vsakem koraku srečujemo spletne aplikacije in algoritme, ki delujejo na principu kolektivnega znanja oz. odločanja v takšni ali drugačni obliki. Eden najbolj znanih je Googlov PageRank algoritem, ki skrbi za rangiranje strani na spletu. Natančno delovanje je poslovna skrivnost, to pa ne spremeni dejstva, da so sestavni del številnih algoritmov človeške množice, ki na podlagi svojih odločitev spreminjajo dinamiko rezultatov.

2.3.1 Google PageRank

Začetki razvoja algoritma segajo v leto 1998, ko je mladi raziskovalec Jon Kleinberg delal na projektu o spletnih iskalnikih imenovanem Hits [4]. Njegov algoritem je uporabljal povezave spletnih strani za izboljšanje rezultatov pri iskanju. Algoritem je bil inovativen, saj so spletni iskalniki do takrat uporabljali za iskanje le tekstovno vsebino. Svoj projekt je predstavil januarja istega leta, medtem ko sta študenta S. Brin in L. Page v Kaliforniji delala na podobnem projektu imenovanem PageRank. Avgusta sta objavila svoj projekt *The PageRank citation ranking: Bringing order to the web* [4].

PageRank je algoritem, ki skrbi za rangiranje posameznih spletnih strani na podlagi števila povezav, ki jih stran vsebuje. Je eden od številnih metod, ki jih Google uporablja za določanje pomembnosti strani in ji dodeli rang: število od 1 do 10. Algoritem interpretira povezavo iz strani A do strani B kot glas strani A za stran B. Teorija PageRanka pravi:

Če ima stran A povezavo na stran B, potem stran A nakazuje, da je stran B pomembna.

Če ima stran A povezavo na stran B, in ima stran B povezavo na stran C, strani A in B nakazujeta da je stran C pomembna.

Strani A1, A2 in A3 imajo povezavo na stran A, strani B1 in B2 pa imata povezavo na stran B. Seštevek PR A1, A2 in A3 je večji od seštevka B1 in B2, iz tega sledi, da je PR strani A večji od strani B, kar pomeni da je stran A pomembnejša od strani B.

V principu gre za sistem nakazovanja, kjer vsaka spletna stran nakazuje na druge strani s spletno povezavo. Enačba za izračun PR se glasi [1]:

$$PR(A) = (1-d) + d(PR(T1) / C(T1) + + PR(Tn)/C(Tn)),$$

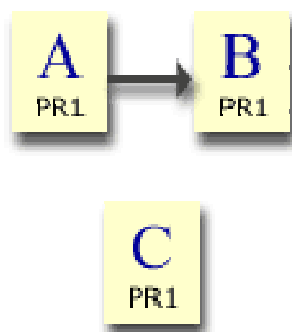
PR(A) – stran, za katero računamo PageRank

PR(Tn) – PageRank strani Tn, ki vsebujejo povezave na stran A

C(Tn) – število odhodnih povezav na strani Tn

d – faktor dušenja (*angl. damping*), ki ga lahko nastavimo na vrednost med 0 do 1

Za ponazoritev delovanja, predpostavimo primer, kjer imamo tri strani. Stran A, B in C. Stran A ima povezavo na stran B (Slika 4). Faktor dušenja ima vrednost 0.85 (standardna vrednost, ki jo uporablja tudi Google). $N = 3$.



Slika 4: Povezava strani A na stran B[1].

Za začetno vrednost določimo $PR = 1$, tako so pred prvo iteracijo naslednje PR vrednosti:

$$PR(A) = 1$$

$$PR(B) = 1$$

$$PR(C) = 1$$

Da ohranimo pravilnost algoritma, predpostavimo, da imata B in C povezave na vse preostale spletne strani in tako izpolnimo enačbe:

$$PR(A) = 0.15 + 0.85 (PR(B)/2 + PR(C)/2)$$

$$PR(B) = 0.15 + 0.85 (PR(A) + PR(C)/2)$$

$$PR(C) = 0.15 + 0.85 (PR(B)/2)$$

Po prvi iteraciji so rezultati naslednji:

$$PR(A) = 1$$

$$PR(B) = 1.425$$

$$PR(C) = 0.575$$

Za demonstracijo delovanja, ponazorimo 7 iteracij:

Iteracija	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	1.425	0.575
2	1	1.244375	0.755625
3	1	1.3211406	0.6788593
4	1	1.2885151	0.7114847
5	1	1.3023809	0.6976188
6	1	1.2964879	0.7035118
7	1	1.2989925	0.7010072

Vsota vseh vrednosti PR je enaka N , ne glede na to, kakšne približke smo izbrali za začetne vrednosti. Stran B še zmeram dominira in bi imela največjo vrednost tudi po 100 iteracijah, ker se vrednosti bistveno več ne spremenijo.

3 SPLETNI PORTAL

Potrebno je bilo sprogramirati rešitev, ki bo omogočala uporabnikom spletnega portala zastavljanje anket in vpogled v njihove rezultate. Ankete uporabnik dodaja dinamično, poleg tega pa jim določa tudi parametre, kot sta tip ankete in maksimalno število glasov, preden se glasovanje zaključi.

Implementirane so še druge storitve, kot so dodajanje novih kategorij za vsebinsko kategorizacijo anket, pa tudi njihova poglobljena analiza. Pri analizi anket smo rezultate ponazorili tudi z 3D grafom, ki je realiziran s pomočjo GD (*Graphic Library*) knjižnice, ki je odprtokodna knjižnica za kreacijo dinamičnih slik. Uporabnost je zagotovljena z dinamičnim generiranjem kakovosti uporabnika, ki ga nagraduje za pravilno glasovanje, saj je s tem prisiljen pošteno glasovati tudi pri anketah drugih uporabnikov. Portal smo začinili z možnostjo izmenjavanja privatnih sporočil med uporabniki ter forumom, kjer je mesto za izmenjavanje mnenj uporabnikov, idej in predlogov za izboljšave. Slika 5 prikazuje prvo stran portala, kjer lahko glasujemo neposredno na ankete iz različnih kategorij. Na prvi strani so prikazane ankete uporabnika z največjo kakovostjo.

The screenshot shows the homepage of a web portal. At the top, there is a navigation bar with links: Domov, Ankete, Arhiv anket, Forum, Test, and a user profile link [Izpiši Primož]. Below the navigation bar is a large 3D pie chart. To the right of the chart is a user menu for 'Primož' with options: Dodaj Anketo, Analiziraj Ankete, Sporočila, Dodaj Kategorijo, Statistika uporabnikov, and Izpiši se.

The main content area features several polls:

- Pozdravljeni!**: A welcome message with a small bar chart and a 'Registracija' button.
- Zadnje teme na forumu**: A list of forum topics with their respective counts: Testirma (1), What happens now? (2), WWW: Federer - Robredo (12), Tori Amos je zakon! (7), and Testirammm (5).
- Top 5 uporabnikov**: A list of top users with their profile pictures and star ratings: Steven (5 stars), Admin (5 stars), Primož (5 stars), Primoz (5 stars), and Daniel (5 stars).
- Sport: Kdo bo osvojil ligo prvakov?**: A poll with options: Barcelona (22%), Real Madrid (17%), Milan (17%), Inter (11%), and Manchester United (33%). Total votes: 18.
- Kultura: Kdo ni bil član Monkeys?**: A poll with options: Peter Turk (0%), Davy Jones (50%), Roger Noll (50%), and Michael Nesmith (0%). Total votes: 2.
- Zdravje: Zaupate našim zdravnikom?**: A poll with options: Absolutno, Zaupam, Delno, Ne preveč, and Niiti najmanj. A 'Glasuj' button is visible.
- Gospodarstvo: Koliko znaša povprečna slovenska plača?**: A poll with options: 350€ - 500€ (0%), 500€ - 600€ (0%), 600€ - 750€ (50%), 750€ - 1000€ (50%), and 1000€+ (0%). Total votes: 2.

Slika 5: Prikaz našega spletnega portala.

3.1 UPORABA PORTALA

Spletni portal je dostopen registriranim in neregistriranim uporabnikom. Neregistrirani uporabniki imajo možnosti glasovanja pri obstoječih anketah, medtem ko je zastavljanje lastnih anket privilegij registriranih uporabnikov. Obiskovalci imajo možnost registracije, ki poleg veljavnih vnosov, zahteva aktivne elektronske naslove, kamor se pošlje povezava za aktiviranje uporabniškega računa, ki jo je potrebno potrditi. Registracija ima implementiran tudi sistem C.A.P.T.C.H.A. (*Completely Automated Public Turing test to tell Computers and Humans Apart*), ki skrbi, da se registrira oseba in ne računalniški bot.

Opišimo nekaj komponent in možnosti v sistemu:

Prijava v sistem - obiskovalec se prijavi v sistem z uporabniškim imenom in geslom, ki si ga je izbral ob registraciji. S prijavo se transformiramo iz navadnega obiskovalca v uporabnika.

Ogled portala – portal prijavljenemu uporabniku nudi vmesnik za dodajanje anket, analiziranje svojih anket ter pošiljanje privatnih sporočil ostalim uporabnikom. Na začetni strani portala je prikazana ena anketa za vsako obstoječo kategorijo. Katera anketa bo prikazana na prvi strani, je odvisno od uspešnosti uporabnika, ki se dinamično spreminja glede na uspešnost pri glasovanju anket drugih uporabnikov, kar je dodatna motivacija za glasovanje. Prva stran prikazuje 5 dodanih tem na forumu in lestvico uspešnosti uporabnikov.

Dodajanje anket – uporabnik lahko doda tri tipe anket:

- anketa z enim specifičnim odgovorom,
- anketa z več možnimi odgovori,
- anketa, kjer glasovalec sam vpiše odgovor.

Izpolnimo lahko tudi parameter maksimalnega števila glasov za določeno anketo. Če tega ne naredimo, je maksimalno število glasov 50.

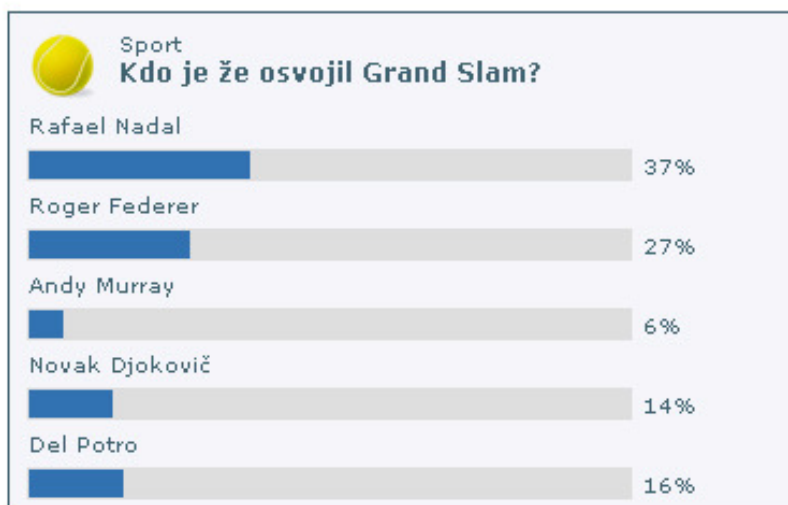
Sporočila – pošiljanje privatnih sporočil drugim uporabnikom ali administratorju strani. Administrator strani lahko pošilja sporočila vsem registriranim uporabnikom.

Arhiv anket – tu se nahajajo zaključene ankete vseh uporabnikov in njihovi rezultati. Arhiv je dostopen vsem obiskovalcem portala, ankete pa lahko sortiramo kronološko po letnici nastanka.

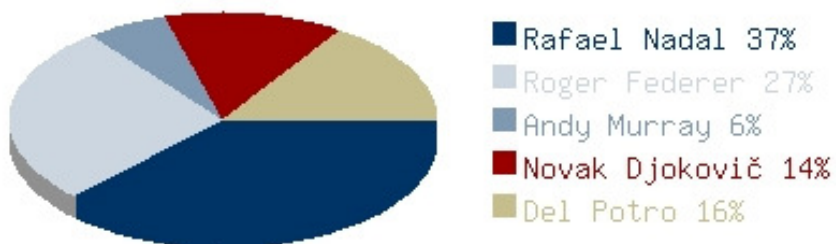
Forum – vsakemu uporabniku se ob registraciji ustvari račun za uporabo foruma, kjer lahko odpira teme in diskutira o aktualnih temah in novicah z drugimi uporabniki.

Analiza anket – vmesnik zajema pregled vseh anket uporabnika, rezultate, status in grafični krožni prikaz porazdeljenosti podatkov, kot je prikazano na sliki 6.

Analize anket



Grafični prikaz:



Slika 6: Končno stanje ankete in grafični prikaz.

Administratorske pravice omogočajo še:

Dodajanje kategorije – dodajanje nove kategorije, kamor se bodo uvrščale ankete z isto tematiko.


























Statistika uporabnikov – statistika vseh uporabnikov: število zastavljenih anket, uspešnost, število objavljenih tem na forumu,...

3.2 BAZA PODATKOV

Pri implementaciji portala smo uporabili različna orodja in tehnike. Kot osnova služi baza podatkov MySQL, ki je vodilna odprtokodna relacijska podatkovna baza. MySQL vsebuje vse funkcije, ki so potrebne za implementacijo malih in srednjih spletnih aplikacij, zato je ta projekt primeren izbor. V bazi hranimo naslednje tabele:

Anketa – tu so shranjeni vsi podatki o posamezni anketi. Sestavljena je iz identifikacijske številke, ki je hkrati tudi ključ tabele, kategorije ankete, identifikacije uporabnika ankete, vprašanje, tip ankete, vsi posamezni odgovori in njihovo število, število glasov, pravilen odgovor, maksimalno število glasov, stanje ankete (zaključena ali odprta) in leto vnosa ankete.

Uporabniki – v tabeli uporabniki so shranjeni podatki o posameznemu uporabniku na spletnem portalu. Samo strukturo tabele nadzorno prikazuje slika 7. Evidenca uporabnikov je sestavljena iz uporabnikove identifikacijske številke (ključ), imena, gesla, ranka (administrator ali navadni uporabnik), elektronske pošte, IP naslova, evidence aktivnosti pri anketah, sodelovanja na forumu ter uspešnosti.

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	uporabnikID	int(2)			No	None	auto_increment	     
<input type="checkbox"/>	ime	varchar(65)	latin1_swedish_ci		No	None		     
<input type="checkbox"/>	geslo	varchar(65)	latin1_swedish_ci		No	None		     
<input type="checkbox"/>	rank	tinyint(1)			No	None		     
<input type="checkbox"/>	aktiviran	tinyint(1)			No	None		     
<input type="checkbox"/>	email	varchar(65)	latin1_swedish_ci		No	None		     
<input type="checkbox"/>	odgovor	int(4)			No	None		     
<input type="checkbox"/>	rating	int(1)			No	None		     
<input type="checkbox"/>	ip	int(10)			No	None		     
<input type="checkbox"/>	pravilnih	int(5)			No	None		     
<input type="checkbox"/>	zgresenih	int(5)			No	None		     
<input type="checkbox"/>	forum_teme	int(4)			No	None		     
<input type="checkbox"/>	forum_posts	int(5)			No	None		     

Slika 7: Struktura tabele uporabniki.

IP – poleg ključa (ipID) je evidentiran tudi IP uporabnika. Zraven sta še polji anketaID (da vemo za katero anketo gre) in pa odgovorID (da vemo na kateri odgovor se nanaša), ki skrbita za to, da lahko uporabnik glasuje za posamezno anketo le enkrat.

Kategorija – vsebuje kategorije.

Sporočila – tu so shranjena sporočila, ki si jih med seboj pošiljajo registrirani uporabniki.

Forum_tema – vsebuje podatke o temi, ki jo uporabniki napišejo na forumu.

Forum_odgovor – vsebuje odgovore na določeno temo posameznega uporabnika.

Pri izvajanju poizvedbe zapisa v bazo moramo biti pozorni na možne napake pri procesu. Največkrat pride do napak v bazi zaradi zapisovanju napačnih znakov, ki niso v skladu z zahtevami MySQL-a in postanejo ranljive na vrsto napada t.i. »*MySQL injection*«. Pri tem napadu lahko napadalec zlorabi neprimerno poizvedbo tako, da z njeno pomočjo pride v naš sistem. Zato je treba podatke za vnos najprej preveriti, če so primerne oblike za vpis v podatkovno bazo. To smo realizirali z gnezdenjem funkcij `mysql_escape_string(strip_tags(htmlspecialchars ()))`. Na ta način zagotovimo, da v bazi ne pride do zapisa nezaželenih nizov ali znakov, ki so lahko vzrok različnim napakam pri delovanju sistema. Primer posodabljanja baze podatkov na varen način je prikazan na sliki 8.

`mysql_escape_string()` – funkcija pripravi podani niz za MySQL poizvedbo. Nadomesti znake, ki imajo v MySQL poseben pomen z pravilnim zaporedjem znakov, primernim za vpis v bazo.

`strip_tags()` – funkcija iz niza odstrani XML, PHP in HTML oznake.

`htmlspecialchars()` – pretvori nekatere vnaprej definirane znake v HTML entitete. Kot primer se s pomočjo funkcije znak »><« (več kot) spremeni v niz »><«.

```
if(!empty($topic) && !empty($text) && !empty($datum))
{
    $topic = mysql_escape_string(strip_tags(htmlspecialchars($topic)));
    $text = mysql_escape_string(strip_tags(htmlspecialchars($text)));
    $datum = mysql_escape_string(strip_tags(htmlspecialchars($datum)));
    $avtor = mysql_escape_string(strip_tags(htmlspecialchars($avtor)));

    $dodajTopic = mysql_query("INSERT INTO forum_thread (topic,text,datum,avtor) VALUES
    ('$topic', '$text', '$datum', '$avtor')");
    $topicIDSqL = mysql_query("SELECT id FROM forum_thread WHERE id = (SELECT MAX(id) FROM forum_thread)");
    $topicIDRez = mysql_fetch_assoc($topicIDSqL);
    $topicID = $topicIDRez['id'];

    $dodajReply = mysql_query("INSERT INTO forum_reply (thread_id,avtor,text,datum) VALUES (
    '$topicID', '$avtor', '$text', '$datum')");

    if($dodajTopic)
    {
        $posodobiUporabnika = mysql_query("UPDATE uporabniki SET forum_teme = forum_teme+1 ,
        forum_posts = forum_posts +1 WHERE ime = '$avtor' ");
        print "Tema je bila uspešno dodana!<br>";
        print "<a href=index.php>Nazaj</a>";
    }
}
```

Slika 8: Primer posodabljanja baze pri dodajanju nove teme.

3.3 REALIZACIJA ANKET IN GLASOVANJA

Pri implementaciji anket v našem portalu smo morali biti pozorni na kar nekaj dejavnikov, ki jih množica uporabnikov potrebuje za uspešno delovanje. Ankete so realizirane s prijaznim vmesnikom in so zastavljene enostavno, tako da lahko posameznik hitro glasuje tudi na večje število anket. Pri anketah, za katere uporabnik še ni glasoval, so prikazani samo mogoči odgovori, brez trenutnih rezultatov. Skrivanje rezultatov je pomembno, saj le-ti lahko sugestivno vplivajo na uporabnikovo izbiro, ki namesto lastnega mnenja začne podlegati izbiri celotne skupine, kar je eno glavnih problemov koncepta modrosti množic. Uporabnik lahko glasuje za posamezno anketo samo enkrat, saj se ob oddanem glasu posodobi zapis v IP tabeli, ki mu preprečuje ponovno glasovanje. S tem se izognemo nevšečnosti zaradi večkratnega glasovanja, ki so lahko posledica napak ali pa zlorab.

 **Zaupate našim zdravnikom?**

Absolutno

Zaupam

Delno

Ne preveč

Niti najmanj

 **Kam radi zahajate?**

Predstave

Gledališča

Koncerte

Kinematografi

Humanitarne prireditve

Slika 9: Prikaz tipa anket.

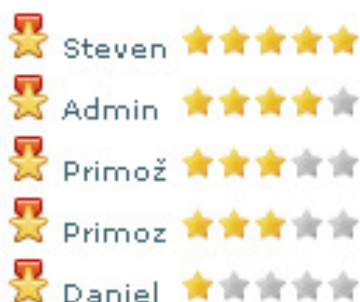
Izvedba glasovanja poteka hitro in brez zapletov. Hitrost in enostavnost sta pomembni lastnosti pri procesu glasovanja, saj lahko močno vplivata na število oddanih glasov. Kadar je vmesnik neprimeren, nerazločen ali časovno potraten, obstaja velika verjetnost, da uporabnik zaradi tega ne bo glasoval. Zato strmimo k čim enostavnejšemu vmesniku, ki omogoča uporabniku hitro izvedbo glasovanja, in si s tem zagotovimo čimvišjo udeleženo uporabnikov pri glasovanju.

3.3.1 Vrste anket

Ankete z posameznim odgovorom so primerne, kadar iščemo en dominanten odgovor, medtem ko ankete z več odgovori uporabimo, kadar je na razpolago več alternativ. Ankete pri katerih uporabniki vpisujejo rezultate, so namenjene ocenjevanju vrednosti oz. individualnim napovedim.

Pri veljavnem glasovanju se v bazo zapiše ID glasovalca, ID ankete, za katero je glasoval, ter številka odgovora. Ko število odgovorov doseže svoj maksimum za določeno anketo, se anketa zaključi in se odstrani iz seznama aktivnih anket, tako da zanjo obiskovalci ne morejo več glasovati. Ob zaključku se posodobi kakovost registriranim glasovalcem, glede na to ali so glasovali v skladu s pravilnim odgovorom, ki ga je po končani anketi vnesel njen avtor. V pozitivnem primeru se uporabnikova kakovost zviša, sicer pa zmanjša. Kakovost uporabnika se šteje le pri anketah s posameznim odgovorom, ker pri tipu anket z več odgovori in ročnim vnosom ni smiseln. Motivacija za pridobitev višje kakovosti je v pozicioniranju posameznih anket. Višja kot je kakovost, več možnosti ima uporabnik, da se njegove ankete pojavijo na prvi strani in so tako deležne večje pozornosti. S kakovostjo smo rešili enega bistvenih problemov modrosti množic, to je motivacija. Najboljših 5 uporabnikov je prikazano na začetni strani, njihova kakovost pa je simbolizirana z zvezdicami na sliki 10.

Top 5 uporabnikov



Slika 10: Prikaz kakovosti uporabnikov z zvezdicami.

Algoritem za posodabljanje kakovosti uporabnikov je prikazan na sliki 11. Po končani anketi njen ustvarjalec ročno vnese številko pravilnega odgovora, ki je v algoritmu shranjena v spremenljivki `maxIndex`. Algoritem nato preveri vse odgovore, ki so bili oddani na anketo in na ustrezen način posodobi bazo uporabnika. Če je uporabnik glasoval pravilno, se poveča polje uporabnika pravih odgovorov, sicer pa polje napačnih odgovorov. V vsakem primeru se poveča tudi polje s skupnim številom odgovorov. V primeru, da je uporabnik glasoval vsaj enkrat, na koncu posodobimo še razmerje kakovosti uporabnika, ki ga izračunamo s posodobljenimi podatki s formulo:

Kakovost uporabnika = (število pravih odgovorov / skupno število odgovorov) * 10

```

$maxIndex = $rezultat['pravilenOdgovor'];

//posodobimo za rating uporabnikov
$posodobiUporabnikaSql = mysql_query("SELECT * FROM ip WHERE anketaID='$anketaID'") or die(mysql_error());

while($vrstica = mysql_fetch_assoc($posodobiUporabnikaSql))
{
    if($vrstica['odgovorID'] == $maxIndex) //pravilni odgovori
    {
        $ip = $vrstica['ip'];
        mysql_query("UPDATE uporabniki SET pravih = pravih+1, odgovor = odgovor+1 WHERE ip='$ip'");
    }
    else //napačni odgovori
    {
        $ip = $vrstica['ip'];
        mysql_query("UPDATE uporabniki SET zgresenih = zgresenih+1, odgovor = odgovor+1 WHERE ip='$ip'");
    }
}

//izračunamo in dodamo rating
$ratingSql = mysql_query("SELECT * FROM uporabniki");

while($vrstica = mysql_fetch_assoc($ratingSql))
{
    $id = $vrstica['uporabnikID'];
    $pravih = $vrstica['pravih'];
    $skupaj = $vrstica['odgovor'];

    if($skupaj != 0)
    {
        $rating = ($pravih / $skupaj) * 10;
        mysql_query("UPDATE uporabniki SET rating = '$rating' WHERE uporabnikID = '$id'");
    }
}

```

Slika 11: Zaključevanje ankete in posodabljanje kakovosti uporabnika.

3.4 GD KNJIŽNJICA

GD je odprtokodna knjižnica za dinamično ustvarjanje slik. Napisana je v Cju, lahko pa jo uporabljamo tudi v drugih jezikih kot so PHP, Pearl in ostalih. Pri PHPju moramo biti pozorni, da imamo vključeno podporo knjižnici. Omogoča nam kreacijo slik formatov png, gif in jpeg, na voljo pa so tudi ostali formati. Uporabljamo jo za spletne aplikacije, saj sliko ustvari tako rekoč brez časovnega zamika, zato pride prav pri risanju grafov in ostalih manjših slik. Za slikovni prikaz rezultatov smo naredili funkcijo, ki na podlagi identifikatorja ankete zgradi grafični prikaz rezultatov v obliki krožnega prikaza.

```
function naredi3DPie()
{
    $anketaID = $_GET['anketaID'];
    $anketaSql = mysql_query("SELECT * FROM anketa WHERE anketaID=' $anketaID'") or die(mysql_error());
    $anketaRez = mysql_fetch_assoc($anketaSql);

    $skupajGlasov = $anketaRez['skupajGlasov'];
    @$racun1 = ((100*$anketaRez['glasov1']/$skupajGlasov))*(3.6);
    @$racun2 = ((100*$anketaRez['glasov2']/$skupajGlasov))*(3.6)+$racun1;
    @$racun3 = ((100*$anketaRez['glasov3']/$skupajGlasov))*(3.6)+$racun2;
    @$racun4 = ((100*$anketaRez['glasov4']/$skupajGlasov))*(3.6)+$racun3;
    @$racun5 = ((100*$anketaRez['glasov5']/$skupajGlasov))*(3.6)+$racun4;
    $procent1 = ((100*$anketaRez['glasov1']/$skupajGlasov));
    $procent2 = ((100*$anketaRez['glasov2']/$skupajGlasov));
    $procent3 = ((100*$anketaRez['glasov3']/$skupajGlasov));
    $procent4 = ((100*$anketaRez['glasov4']/$skupajGlasov));
    $procent5 = ((100*$anketaRez['glasov5']/$skupajGlasov));

    $image = imagecreatetruecolor(400,200);

    $darkBlue = imagecolorallocate($image,0x00,0x33,0x66);
    $blue = imagecolorallocate($image,0x7f,0x99,0xb2);
    $darkgray = imagecolorallocate($image, 0x90, 0x90, 0x90);
    $darknavy = imagecolorallocate($image, 0x00, 0x00, 0x50);
    $gray = imagecolorallocate($image, 0xc0, 0xc0, 0xc0);
    $white = imagecolorallocate($image, 255, 255, 255);
    $yellow = imagecolorallocate($image,0xc6,0xbe,0x8c);
    $lightBlue = imagecolorallocate($image,0xcc,0xd6,0xed);
    $darkRed = imagecolorallocate($image,0x99,0x00,0x00);
    imagefill($image, 0, 0, $white);

    $tableOfColors = array($darkBlue,$lightBlue,$blue,$darkRed,$yellow); $height=220; $width = 220; $label_place=45;
    $label_output = array($anketaRez['odgovor1']." ".round($procent1)."%",$anketaRez['odgovor2']." ".round($procent2)."%",
    $anketaRez['odgovor3']." ".round($procent3)."%",$anketaRez['odgovor4']." ".round($procent4)."%",$anketaRez['odgovor5']." ".round($procent5)."%");

    for($i = 0; $i < 5; $i++)
    {
        imagefilledrectangle($image,$width+6,$label_place,$width+16,$label_place+10,$tableOfColors[$i]);
        imagerectangle($image,$width+6,$label_place,$width+16,$label_place+10,$tableOfColors[$i]);
        imagestring($image,'4',$width+20,$label_place,$label_output[$i],$tableOfColors[$i]);
        $label_place = $label_place+20;
    }

    for ($i = 100; $i > 90; $i--)
    {
        imagefilledarc($image, 100, $i, 200, 100, 0, $racun1, $darkBlue, IMG_ARC_PIE);
        imagefilledarc($image, 100, $i, 200, 100, $racun1, $racun2, $darkgray, IMG_ARC_PIE);
        imagefilledarc($image, 100, $i, 200, 100, $racun2, $racun3, $darknavy, IMG_ARC_PIE);
        imagefilledarc($image, 100, $i, 200, 100, $racun3, $racun4, $gray, IMG_ARC_PIE);
        imagefilledarc($image, 100, $i, 200, 100, $racun4, $racun5, $white, IMG_ARC_PIE);
    }

    imagefilledarc($image, 100, 90, 200, 100, 0, $racun1, $darkBlue, IMG_ARC_PIE);
    imagefilledarc($image, 100, 90, 200, 100, $racun1, $racun2, $lightBlue, IMG_ARC_PIE);
    imagefilledarc($image, 100, 90, 200, 100, $racun2, $racun3, $blue, IMG_ARC_PIE);
    imagefilledarc($image, 100, 90, 200, 100, $racun3, $racun4, $darkRed, IMG_ARC_PIE);
    imagefilledarc($image, 100, 90, 200, 100, $racun4, $racun5, $yellow, IMG_ARC_PIE);
}
```

Slika 12: Prikaz funkcije, ki zgradi grafično predstavitev ankete.

Funkcija za tvorbo krožnega prikaza je prikazana na sliki 12 in je implementirana tako, da pretvorimo vse potrebne rezultate v obliko, ki bo zadovoljevala krožno predstavitev.

Algoritem poteka tako, da najprej poišče za katero anketo gre. To storimo s standardno PHP funkcijo `$_GET[]`, kjer pridobimo vrednost parametra `anketaID` preko naslovne vrstice.

Na podlagi identifikatorja ankete naredimo MySQL poizvedbo, da pridemo do rezultatov ankete. Rezultate je nato treba razporediti od 0 do 360 stopinj, ki nam nato služijo kot vhodni parametri pri funkciji za risanje. Ko rezultate med 0 in 100 pomnožimo z 3.6 dobimo deleže, ki ustrezajo krožnemu prikazu. Za risanje skrbi funkcija `imagefilledarc()`, ki na podano sliko izriše delež, katerega dimenzijo, pozicijo in barvo podamo kot parameter funkcije. Za privlačnejši videz smo grafični prikaz dodelali v 3D obliko, ki je narejena po istem postopku z različno barvo in koordinatami.

3.5 REGISTRACIJA UPORABNIKA

Registracija uporabnikov omogoča obiskovalcem možnost ustvarjanja lastnega računa na portalu (slika 13). Z uspešno registracijo obiskovalec pridobi vse pravice registriranega uporabnika in možnost aktivnega sodelovanja na portalu. Pri implementaciji registracije smo bili pozorni na:

- ime uporabnika ne sme biti ponovno uporabljeno,
- vneseni gesli se morata ujemati,
- naslov elektronske pošte mora biti veljaven, saj dobite avtomatično pošto, ki vsebuje povezavo za potrditev registracije.

Pri registraciji smo uporabili tehnologiji Ajax (Jquery) in CAPTCHA. Jquery nam olajša vnašanje podatkov v polja, tako da sproti preverja prostost oz. zasedenost podatkov.

CAPTCHA je sistem proti registraciji avtomatskih agentov in skrbi, da je registrant zares oseba.



Registracija

Ime:	<input type="text" value="Matic"/>	Veljavno!
Geslo:	<input type="password" value="•••••"/>	
Potrdi geslo:	<input type="password" value="•••••"/>	
Email:	<input type="text" value="matic@primer.si"/>	Veljavno!
Vpišite črne črke:	<input type="text" value="f8810j"/>  <input type="text" value="f80"/>	
<input type="button" value="Registracija"/>		

Slika 13: Registracija uporabnikov.

3.5.1 JQuery

Jquery je knjižnica za skriptni jezik Javascript in je implementirana tako, da nudi uporabnikom hitrejši razvoj spletnih aplikacij in skript. Primeren je za razvoj širokega spektra skript od vizualnih učinkov do večjih aplikacij z kompleksnejšimi vmesniki. Največkrat se uporablja za:

- Ajax aplikacije,
- dodajanje animiranih učinkov spletnim elementom,
- ustvarjanje večnivojskih animiranih menijev,
- predstavitev galerije slik z vizualnimi učinki,
- implementacijo strani, pri kateri lahko s povleci in spusti (*drag and drop*) načinom preuredimo razporeditev elementov na strani,
- podporo pri izpolnjevanju spletnih obrazcev (*forms*).

Jquery je orodje, ki skrbi da naše strani oživijo. Temelji na preprosti uporabi, kar potrjuje njegov slogan »*Write less, do more*«. Omogoča nam reševanje težav v zvezi s spletnimi brskalniki, kjer poenoti implementacije pri različnih brskalnikih. V primeru, da JQuery podpira določen brskalnik, iste JQuery funkcije delujejo povsod na enak način.

Za uporabo ga moramo vključiti v svojo spletno stran. Na uradni spletni strani <http://jquery.com/> dobimo dve različici, minimizirano ali v izvorni obliki. Minimizirana različica vsebuje stisnjeno Javascript izvorno kodo z namenom, da zavzame čim manj prostora. Izvorna oblika je namenjena tistim, ki bi radi izvedeli, kako JQuery dejansko deluje.

Na našem portalu potek registracije spremlja JQuery skripta, ki skrbi za obveščanje uporabnika o razpoložljivosti podatkov, ki so pomembni pri registraciji (slika 14).

Skripta preverja razpoložljivost pri vnosu imena in elektronskega naslova uporabnika in ga sproti obvešča o statusu vnešenih podatkov. Ta način omogoča, da se lahko uporabnik pri registraciji izogne večkratnim poizkusom izpolnjevanja in potrjevanja nepravilnih oz. zasedenih podatkov.

```

<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
    $(document).ready(function(){
        $('#feedName').load('check.php').show();

        $('#ime_vnos').keyup(function(){
            $.post('check.php', {username: form.uporabniskoIme.value},
                function(result){
                    $('#feedName').html(result).show();
                });
        });
    });
</script>
<script type="text/javascript">
    $(document).ready(function(){
        $('#feedEmail').load('check.php').show();

        $('#email_vnos').keyup(function(){
            $.post('check.php', {email: form.email.value},
                function(result){
                    $('#feedEmail').html(result).show();
                });
        });
    });
</script>

```

Slika 14: Skripta za preverjanje zasedenosti imena in elektronske pošte.

Za dejansko delovanje skripta ne zadostuje. Potrebujemo še funkcijo, ki nam pove, v katerih primerih je vnesena vrednost pravilna in kdaj ne. Uporabniško ime je veljavno, kadar vsebuje več kot dva znaka in kadar v bazi podatkov še ni registriranega uporabnika z istim imenom. Za primer delovanja si oglejmo sliko 15, ki prikazuje primer kode za preverjanje veljavnosti elektronskega naslova.

Naslov elektronske pošte je primeren kadar:

- vnos ni prazen (prazno polje),
- vnos vsebuje znaka afna (@) in pika (.),
- vnos ni enak že obstoječem elektronskem naslovu v bazi podatkov.

```

else if(isset($_POST['email']))
{
    @$email = mysql_real_escape_string($_POST['email']);
    $checkEmail = mysql_query("SELECT email FROM uporabniki WHERE email = '$email' ");
    $preveriEmail = mysql_num_rows($checkEmail);

    if($email == NULL)
    {
        print "Vnesi email";
    }
    else if($preveriEmail == 0 )
    {
        if((strpos($email,'@') == false) || (strpos($email, '.') == false))
            print "<font color = red>Neveljavno</font>";
        else
            print "<font color = green>Veljavno!</font>";
    }
    else if($preveriEmail == 1)
    {
        print "<font color = red>Zasedeno!</font>";
    }
}

```

Slika 15: Validacija elektronske pošte.

Na sliki 15 je prikazano preverjanje treh pogojev pri validaciji elektronske pošte. V prvem primeru, ko uporabnik še ni vnesel nobenega znaka, ga na vnos elektronske pošte opozorimo. V primeru, če je uporabnik izpolnil polje z obstoječim elektronskim naslovom, ga opozorimo, da ta elektronska pošta v naši bazi že obstaja. Sicer je elektronski naslov veljaven, če vsebuje znaka afna (@) in pika (.), ki sta sestavni del vsakega elektronskega naslova.

3.6 FORUM

Implementacija spletnega foruma je nastala kot podpora uporabnikom za izmenjavanje uporabniških idej in mnenj. Forum je dostopen samo registriranim uporabnikom. Kljub temu, da neposredno ne vpliva na sam proces anketiranja, pa z odpiranjem novih tem ustvarjamo skupnost, ki je povezana z anketami, njihovimi rezultati ali pa z uporabo samega portala. Pri implementaciji je bil poudarek predvsem na preglednosti in enostavnosti delovanja.

Domov Ankete Arhiv > Forum Test [Izpiši Ben]

Forum

Pozdravljen Ben!
[Nova tema](#)

#	Tema	Avtor	Pogledi	Odgovori	Datum	Zadnji
	Podobne spletne aplikacije	Ben	0	0	21.2.2011,	Ben
	Najpogostejša vprašanja	Max	4	3	21.2.2011,	Ben
	Kako si zvišati rating?	Daniel	0	0	21.2.2011,	Daniel
	Analiziranje posameznih anket	Daniel	4	3	21.2.2011,	Daniel
	Najenostavnejši način pridobivanja informacij	Steven	2	1	21.2.2011,	Ben
	Testna tema	Primož	4	3	21.2.2011,	Daniel
	Zaupate teoriji modrost množic?	Primož	1	0	21.2.2011,	Primož
	Vam je všeč portal?	Primož	2	1	21.2.2011,	Primož

Strani: [1] 2 3

Slika 16: Spletni forum.

Registracija na forum ni potrebna, saj je uporabnik že identificiran. Pri uporabi se v bazi evidentirajo zapisi naslova teme, avtorja, število pogledov teme, število odgovorov na temo in datum. Teme so sortirane kronološko od najnovejše do najstarejše. Na ta način ohranjamo njihovo aktualnost. Kot estetski dodatek smo k dodajanju odgovorov dodali možnost nasmehov (*angl. Smilies*), ki so implementirani s posebno funkcijo ter Javascriptom.

4 TEHNOLOGIJE

4.1 AJAX

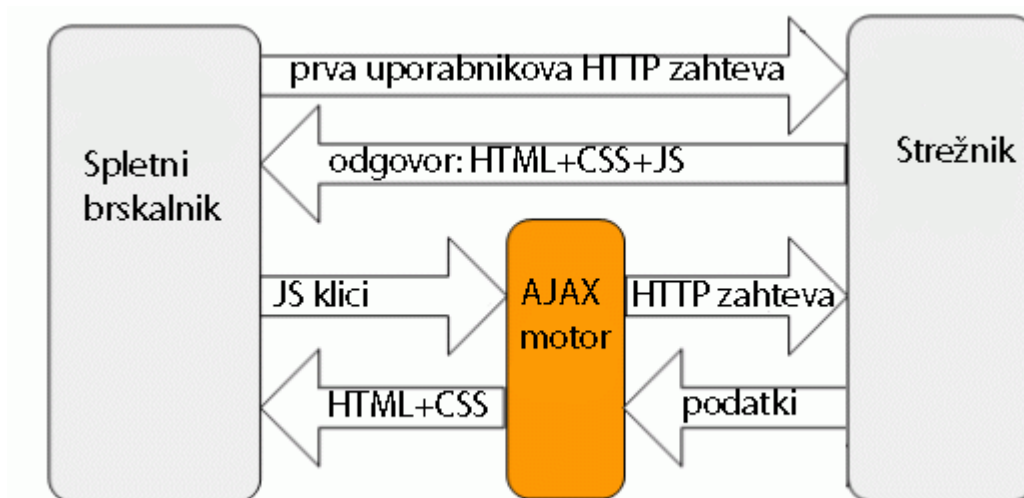
Pri običajnih vnosih podatkov preko spletnega brskalnika je treba vnos najprej potrditi in nato počakati na ponovno nalaganje strani. Podobno se zgodi tudi, če hočemo dobiti nove podatke iz strežnika. Takšen način je precej zamuden in neučinkovit, kadar se med klientom in strežnikom prenaša majhna količina podatkov. Ajax je tehnologija, ki nam s pomočjo Javascript klicev omogoča, da uporabniku ni vedno potrebno čakati na vnovično nalaganje strani. Ajax je odmik od načela zahteva/odgovor, ki smo ga poznali do sedaj. Spletne aplikacije lahko komunicirajo s strežnikom asinhrono in s tem dosegajo bistveno večjo hitrost in enostavnost.

Ajax dodaja komunikaciji nivo, ki mu pravimo Ajaxov pogon (*Ajax engine*) in se obnaša kot vmesnik med uporabnikom na klientu in na strežniku. Sliki 17 in 18 prikazujeta razliko med komunikacijo brez in z Ajaxom:



Slika 17: Potek navadne komunikacije med klientom in strežnikom[3].

Pri tradicionalnih spletnih aplikacijah klient komunicira z strežnikom neposredno preko HTTP zahtev. Ko uporabnik sproži zahtevo po strani, strežnik pošlje celotno HTML in CSS kodo. Ko uporabnik vnese podatke in ponovno sproži zahtevo, strežnik sprocesa podatke in zgradi celotno stran ter ponovno pošlje klientu celoten paket (slika 17).



Slika 18: Potek komunikacije z Ajax pogonom[3].

Pri aplikacijah z Ajaxom se stran v celoti naloži samo enkrat pri prvi zahtevi. Poleg standardnih HTML in CSS podatkov se prenesejo še Javascript datoteke, ki jih imenujemo Ajax pogon. Nadaljnje zahteve do strežnika bodo poslane v oblik Javascript klicev do Ajaxovega pogona. Ajax pogon zahteva podatke od spletnega strežnika na asinhron način. Tako se zahteva in prenese majhna količina podatkov, ki jih uporabnik potrebuje. Ajax pogon nato prikaže informacije brez ponovnega nalaganja celotne strani, kot je prikazano na sliki 18. Tak način komunikacije omogoča uporabniku prijeten način pridobivanja informacij, saj se prenese samo potrebna količina podatkov in ne celotna stran, zato daje občutek takojšnjega odziva, kar zelo približa spletne aplikacije tistim, ki jih uporabljamo lokalno.

Ajax postaja nepogrešljiv del sodobnih spletnih aplikacij in tudi z njegovo pomočjo smo implementirali neposredno preverjanje zasedenosti uporabniškega imena in elektronske pošte pri registraciji. Ajax lahko uporabljamo še za vrsto drugih nalog:

- preprost sistem glasovanja,
- validacija forme z uporabniškimi podatki,
- hitro prikazovanje vsebin na straneh z veliko uporabniki,
- avtomatsko ponujanje zadetkov pri iskanju posameznih stvari.

4.2 CAPTCHA

CAPTCHA je sistem, ki poskuša ugotoviti, ali je določen uporabnik oseba (človek) ali stroj. Temelji na načelu poziv – odgovor, kar pomeni, da se uporabniku prikaže slika, na kateri je zaporedje znakov, in ga pozove, naj znake iz slike prepíše v ustrezno vnosno polje.

Računalnik test CAPTCHA ustvari in oceni, ali ga je uporabnik pravilno rešil, ne more pa ga rešiti sam. To omogoča, da lahko na podlagi testa razločimo med človeškimi in nečloveškimi (programi, boti) uporabniki.

Tehnologijo CAPTCHA je mogoče uporabiti v različne namene. Lahko jo uporabljamo pri registraciji uporabnika, pri prijavi ali za potrjevanje sporočil. Tehnologijo uporabljajo mnoga podjetja, kot so Google, Yahoo in še mnoga druga. Googlova implementacija je prikazana na sliki 19.

Kako zanesljiv je CAPTCHA sistem je odvisno od tega, kako dobro so zasnovane slike. Dobro zasnovane je strojno tako rekoč nemogoče rešiti, za šibke slike so na voljo programi za razbijanje in samodejno reševanje testov.



Sign in to Gmail with your
Google Account

Username: fajalar

Password: ●●●●●●●●

Enter the correct password above
and then type the characters you see
in the picture below.



&

Enter the letters as they are shown
in the image above.
Letters are not case-sensitive

Remember me on this computer.

[I cannot access my account](#)

Slika 19: Primer CAPTCHA sistema pri Googlu.

Eden najbolj inovativnih načinov za uporabo sistema CAPTCHA je zaščita elektronskih naslovov pred nezaželenimi sporočili. Takšno rešitev ponuja spletna stran reCAPTCHA MailHide (<http://www.google.com/recaptcha/mailhide/>), kjer uporabnik vnese svoj elektronski naslov, spletna stran pa mu izpiše povezavo ali HTML programsko kodo. Uporabnik jo objavi na svoji spletni strani, povezava oz. HTML programska koda pa zahteva, da je treba pred izpisom elektronskega naslova rešiti test CAPTCHA in tako zaščititi naš naslov pred avtomatiziranimi zbiralci poštnih naslovov.

Predvidimo lahko, da se bo potreba po tej tehnologiji še povečala. Microsoft je s projektom Asirra predlagal nov pristop, ki naj bi bil odporen proti vsem trenutno znanim napadom. Uporabljajo pristop s slikovnim pozivom, ki za razliko od klasičnih tekstovnih pozivov CAPTCHA, od uporabnikov zahteva prepoznavanje objekta na sliki (npr. ali gre za mačko ali psa). Pri slikovnih vrstah pozivov se srečujemo z določenimi težavami. Slikovni pozivi namreč zasedajo več pomnilniškega prostora, to pa lahko predstavlja težavo pri velikih množicah. Če imamo premajhno množico, pa obstaja nevarnost, da napadalec oblikuje zbirko rešitev. Trenutno se slikovni testi CAPTCHA v praksi ne uporabljajo, temveč skrbniki spletnih storitev še zmeraj uporabljajo izboljšane besedilne CAPTCHA [2].

Implementacija CAPTCHE na našem portalu je realizirana tako, da se nam ob registraciji avtomatsko zgenerira 6 znakovni niz črk oz. števil. Niz se pretvori v sliko, kot je prikazana na sliki 20. Varnostni sistem od uporabnika zahteva, da iz slike v ustrezno polje prepíše le črne znake, rdeče pa izpusti. V primeru, da je niz črnih znakov nepravilen, registracija ne bo uspešna.



Slika 20: Primer implementacije CAPTCHA.

4.3 CSS

CSS so predloge, ki določajo izgled same spletne strani. Slog lahko določimo za vsak HTML element in s tem določimo, kako so elementi oblikovani. Slogovne predloge so ponavadi shranjene v posebnem dokumentu, kar je tudi ena glavnih prednosti CSSja, saj tako loči oblikovno od dejanske vsebine spletne strani in tako prispeva k berljivosti in razumevanju kode. Elementom HTML spreminjamo pisavo, obliko, velikost ter ostale funkcije, ki so povezane z izgledom strani, ne da bi spreminjali HTML kodo. CSS podpira tudi grupiranje elementov v razrede, ki še poenostavijo oblikovanje spletne strani.

Vzorec oblikovnih predlog za tabeli anketa in forumTable je prikazan na sliki 21.

```
table.anketa
{
    font: 10px verdana, arial, helvetica;
    width: 300px;
    padding: 3px 4px 2px 4px;
    border: 1px solid #416271;
    background-color:#F4F5FA ;
}

table.forumTable
{
    font: 10px verdana, arial, helvetica ;
    width: 1050px;
    padding: 3px 4px 2px 4px;
    border: 1px solid #416271;
    background-color:#FBFBFB ;
    border: 0.5px #DDE4EB solid;
    border-collapse: collapse;
    border-spacing: 1px;
    margin:0px;
}
```

Slika 21: Vzorec CSS kode.

5 ZAKLJUČEK

V diplomski nalogi smo pokazali spletno implementacijo portala za uporabo modrosti množic. Predstavili smo tehnike in orodja, s katerimi je portal realiziran, ter prikazali delovanje.

Modrost množic se je izkazala kot zanimiva teorija, ki nam v pravih pogoji lahko postreže z zelo točnimi informacijami. Obstajajo tudi primeri, kjer gre za očitno zgrešeno oceno množice. Surowiecki, avtor knjige »*Wisdom of Crowds*«, trdi, da prednosti kolektivnega odločanja odpovedo, kadar so potlačene možnosti posameznikovega sklepanja na osnovi lastnih informacij [7]. To se lahko zgodi, kadar je:

- skupina preveč homogena,
- odločanje preveč centralizirano,
- informacije preveč razdrobljene,
- delovanje preveč posnemovalno.

Veliko negativnih lastnosti, ki naredijo mnenje skupine neučinkovito, je pri implementaciji na internetu manj verjetnih. Pri spletnem glasovanju je verjetnost homogene skupine manjša, kot bi to bilo sicer na družbenih dogodkih. Tudi posnemanja skoraj ni, saj glasovalec vidi rezultate ostalih šele po svojem glasovanju.

Večina portala je realiziranega z skriptnim jezikom PHP, ki še danes ostaja eden vodilnih jezikov za implementacije spletnih aplikacij. Kljub temu vidimo, da dandanes za implementacijo spletne aplikacije PHP in HTML nista dovolj. Na voljo imamo veliko orodij, ki spletne strani poživijo, jim dodajo lepši videz in izboljšajo funkcionalnost. Medtem, ko je za estetski del CSS skoraj nepogrešljiv, za funkcionalni del največkrat uporabljamo Javascript in njegovo knjižnico Jquery ter Ajax. S pomočjo Ajaxa so spletne aplikacije dobile novo dinamiko, saj ob pravilni implementaciji lahko uporabniku zelo olajša ter pohitri uporabo aplikacije.

Med razvojem se je porodilo kar nekaj idej, ki bi spletno aplikacijo lahko izboljšale.

- Glasovanju za ankete bi lahko dodali še atribut spol glasovalca. S tem bi lahko analizirali rezultate ankete tudi po spolu, kar bi pokazalo raznolikost razmišljanja med žensko in moško populacijo.
- Rezultate ankete bi lahko spremenili v Pdf, Exel ali druge oblike datotek, ki bi jih lahko uporabnik po želji prenesel na lokalni računalnik.
- Iskanje anket po določenih besedah: ker bi se ob velikem številu uporabnikov posamezne ankete začele ponavljati, bi bilo na strani dobro implementirati tudi iskalnik anket.
- Možnost dodajanja in glasovanja na forumu. Zadevo bi lahko še bolj poenostavili, če bi implementirali možnost dodajanja in glasovanja na ankete v sami temi foruma. Tako bi lahko vzporedno potekale anketa in debata nanjo.

Kljub omenjenim pomanjkljivostim, se je naš portal izkazal za uporabnega. Omenjene izboljšave bi bile potencialno dobra rešitev, če bi portal obiskovala večja množica uporabnikov, za manjšo skupnost pa zadostuje tudi opisana implementacija.

6 KAZALO SLIK

Slika 1: Spletni portal Digg	4
Slika 2: Primer glasovanja ljudstva	7
Slika 3: Glasovanje množice o številu bonbonov v skledi [5]	11
Slika 4: Povezava iz strani A na stran B [1]	14
Slika 5: Prikaz našega spletnega portala	15
Slika 6: Končno stanje ankete in grafični prikaz	17
Slika 7: Struktura tabele uporabniki	18
Slika 8: Primer posodabljanja baze pri dodajanju nove teme	19
Slika 9: Prikaz tipa anket	20
Slika 10: Prikaz kakovosti uporabnikov z zvezdicami	21
Slika 11: Zaključevanje ankete in posodabljanje kakovosti uporabnika	22
Slika 12: Prikaz funkcije, ki zgradi grafično predstavitev ankete	23
Slika 13: Registracija uporabnikov	25
Slika 14: Skripta za preverjanje zasedenosti imena in elektronske pošte	27
Slika 15: Validacija elektronske pošte	28
Slika 16: Spletni forum	29
Slika 17: Potek navadne komunikacije med klientom in strežnikom [3]	30
Slika 18: Potek komunikacije z Ajax motorjem [3]	31
Slika 19: Primer CAPTCHA sistema pri Googlu	32
Slika 20: Primer implementacije CAPTCHA	33
Slika 21: Vzorec CSS kode	34

7 SEZNAM VIROV IN LITERATURE

[1] Phil Craven: »Google PageRank Explained and how to make the most of it.«

Dostopno na: <http://www.webworkshop.net/pagerank.html>

[2] Marko Holbl, Preverjamo s CAPTCHA.

Revija Monitor, marec 2008, dostopno na:

<http://www.monitor.si/clanek/preverjamo-s-captcha/>

[3] Interakt: »Ajax: Asynchronously moving forward.«

November 2005, dostopno na:

http://www.interaktonline.com/Support/Articles/Details/AJAX%3A+Asynchronously+Moving+Forward-Introduction.html?id_art=36&id_asc=306

[4] Amy N. Langville and Carl D. Meyer: »Deep inside PageRank.«

Oktober 2004, dostopno na:

http://meyer.math.ncsu.edu/Meyer/PS_Files/DeeperInsidePR.pdf

[5] Michael J. Mauboussin: »Explaining the wisdom of crowds.«

Legg Mason Capital Management, marec 2007, dostopno na:

http://www.lmcm.com/pdf/Explaining_the_Wisdom_of_Crowds.pdf

[6] Derek Powazek: »The wisdom of community.«

A List Apart, maj 2009, dostopno na:

<http://www.alistapart.com/articles/the-wisdom-of-community/>

[7] James Surowiecki, »*The wisdom of the crowds, why the many are smarter than the few and how collective wisdom shapes business, economies, societies and nations.*«

New York: Doubleday 2004