

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N**

Luka Banožić

**RAZVOJ PERSONALIZIRANOG WEB
PORTALA**

DIPLOMSKI RAD

Varaždin, 2015.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Luka Banožić

Matični broj: 42563/13-R

Studij: Informacijsko i programsko inženjerstvo

RAZVOJ PERSONALIZIRANOG WEB
PORTALA

DIPLOMSKI RAD

Mentor:

Dr. sc. Tihomir Orehovački

Varaždin, rujan 2015.

Sadržaj

1. Uvod	1
2. Povijest Weba	2
3. Web portal	4
3.1. Povijest web portala	4
3.2. Funkcije web portala	5
3.3. Vrste web portala.....	6
3.3.1. Opći web portali	7
3.3.2. Vertikalni web portali.....	8
3.3.3. Horizontalni web portali.....	10
3.3.4. Web portali zajednica	11
3.3.5. Korporativni web portali	12
3.3.6. Web portali e-tržišta	13
3.3.7. Mobilni web portali	15
3.3.8. Informativni web portali.....	16
3.3.9. Specijalizirani web portali.....	17
3.4. Zadovoljstvo korisnika web portalima	18
3.4.1. Zadovoljstvo sadržajem.....	18
3.4.2. Zadovoljstvo dizajnom	19
3.4.3. Zadovoljstvo personalizacijom.....	20
3.4.4. Zadovoljstvo podrškom zajednicama	20
4. Personalizacija web portala	22
4.1. Pojava i uspon personaliziranih web portala	22
4.2. Elementi personalizacije web portala	22
4.2.1. Strukturalni elementi	23
4.2.1.1. Izgled i osjećaj.....	23
4.2.1.2. Zaglavlje i podnožje	23
4.2.1.3. Navigacija.....	24
4.2.2. Funkcionalni elementi	24
4.2.2.1. Stranice i sadržaj	24
4.2.2.2. Rezultati pretraživanja.....	25
4.2.2.3. Katalozi	26
4.3. Eksplicitna i implicitna personalizacija web portala	26
5. Vrste personaliziranih web portala	29
5.1. Distribuirani web portal.....	29

5.2. Hibridna web aplikacija (<i>mashup</i>).....	30
5.3. Usporedba.....	31
6. Web tehnologije za razvoj personaliziranog web portala.....	32
6.1. HTML.....	32
6.2. CSS.....	33
6.3. JavaScript	35
6.4. Ajax	36
6.5. PHP.....	37
6.6. MySQL	39
6.7. RSS.....	40
7. Razvoj personaliziranog web portala: praktični dio	42
7.1. Logotip	42
7.2. Zaglavlje.....	43
7.3. Glavni dio	45
7.4. Dodavanje izvora pomoću izbornika.....	61
7.5. Postavke.....	63
7.6. Registracija i prijava.....	66
8. Zaključak	68
9. Literatura	69

1. Uvod

Razvoj Weba je kroz godine ljudima donio nove načine pristupa i dijeljenja informacija. Porastom web stranica su se pojavili i web portali kao posebne web stranice koje prikupljaju veliki broj informacija i prikazuju ih na jednom mjestu kako bi im korisnici jednostavnije pristupili.

U današnje vrijeme korisnici imaju veliki izbor portala i tema sadržaja. No, preveliki izbor nekad i ne mora značiti pozitivno iskustvo za korisnike. Iako se neki portali ističu kvalitetnim sadržajem, a neki privlačno oblikovanim sučeljem, nekim korisnicima takvi portali neće odgovarati. U takvim slučajevima bi trebali postojati web portali preko kojih korisnici mogu sebi prilagoditi cijelo iskustvo kako bi dobili osjećaj da je taj web portal namijenjen samo njima. Upravo to pružaju personalizirani web portali.

Na sve većem broju web portala i uz sve veći raspon dostupnih informacija je teško pronaći ono što je korisnicima osobno važno i zanimljivo. Personalizirani web portali daju mogućnost da se istakne ono što je važno, te da se cijeli izgled i osjećaj portala prilagodi korisnicima onako kako im odgovara. Iako postoje korisnici koji na portalu neće imati potrebu mijenjati elemente, sama prisutnost tih mogućnosti ne bi trebala negativno utjecati na zadovoljstvo korisnika korištenjem portala. Primjerice, nije suvišno imati mogućnost premještanja odjeljka s vijestima iz politike na dno portala. Uz to, korisnicima nekad jednostavno ne odgovara zadana tema boja na portalu ili je veličina fonta prevelika, pa za to postoji potreba osobnog prilagođavanja.

Ishod razvoja personaliziranog web portala bi mogao pružiti veće zadovoljstvo korištenjem i potencijalno zadržati korisnike na duže vrijeme. Kako bi važnost i potreba samog razvoja personaliziranog web portala bila predočena, u ovom radu je opisan utjecaj Weba i razvoj istog kroz godine, a također je opisana pojava i utjecaj web portala, uz navedene funkcije i vrste web portala. Navedeno je koji su točno čimbenici koji utječu na zadovoljstvo korisnika web portalima. Opisano je kako i što se kod web portala može personalizirati. Kao mogući pristupi razvoju personaliziranog web portala su navedeni distribuirani web portali i *mashup*-ovi, tj. hibridne web aplikacije, te su opisane web tehnologije u kojima je izrađen praktični dio rada koji je opisan na kraju.

2. Povijest Weba

World Wide Web, odnosno popularnije nazivan kraćim nazivom Web, je danas toliko važan i popularan da je većini ljudi bez njega teško zamisliti život. Do početka razvoja Weba je došlo kada je krajem 1989. godine britanski računalni znanstvenik Tim Berners-Lee predložio umreženi hipertekstualni sustav za CERN (Cailliau, 1995). Cailliau (1995) navodi i da je sljedeće godine sustav predložen upravi CERN-a i dobio naziv *World-Wide Web*, a sam sustav, koji je uključivao HTML jezik, HTTP protokol i prototip web preglednika, je Berners-Lee implementirao na svojem NeXT računalu koje je tada bilo prvi web poslužitelj. Već u sljedećih nekoliko godina je interes za Webom, a i Internetom porastao, a s time i broj web poslužitelja. Početkom 1993. godine su Marc Andreessen i Eric Bina razvili Mosaic, prvi web preglednik s grafičkim sučeljem koji je popularizirao Web (Yang, 2012). U narednih nekoliko godina je web promet znatno porastao.

Sama evolucija Weba je velika, pa se može podijeliti u više faza: Web 1.0, Web 2.0, Web 3.0 i Web 4.0 (Murugesan, 2010). U prvoj fazi, Webu 1.0, je najvažnije objavljivanje informacija kojima korisnici mogu pristupiti koristeći web preglednik preko Interneta. Murugesan (2010) navodi da su se pojavile platforme i alati za web razvoj, rastao je broj web stranica, Web se je prvi puta počeo koristiti u komercijalne svrhe, te su se pojavili novi web poslovni modeli.

Dok je Web 1.0 usmjeren na informacije, u drugoj fazi, Webu 2.0, je najvažnije povezivanje ljudi. Izraz Web 2.0 je 2005. godine popularizirao Tim O'Reiley (Constantinides, 2010). Faza Weba 2.0 je predstavila dinamičnost i interaktivnost, a dopušta i potiče sve korisnike na izradu, dijeljenje i distribuiranje informacija, te obuhvaća web tehnologije i servise kao što su blogovi, društvene mreže, wiki stranice, komunikacijski alati i alati koji su usmjereni na dijeljenje sadržaja i suradnju preko Weba (Murugesan, 2010). U ovoj fazi se je pojavio *mashup*, hibridna web aplikacija koja kombinira informacije i servise iz više izvora u svrhu kreiranja novih informacija i servisa (Fichter, 2009).

U 2006. godini, u članku *The New York Times*-a je John Markoff objavio članak u kojem sljedeću fazu evolucije Weba naziva Web 3.0 (još nazivan i semantički Web), a odnosi se na treću generaciju web tehnologija i servisa koji su usmjereni na razumijevanje informacija na Webu od strane računala u svrhu pružanja produktivnijih i intuitivnijih

korisničkih iskustava (Murugesan, 2010). Web 3.0 daje naglasak na integriranje podataka, znanja i aplikacija na Webu kako bi isti mogli surađivati u svrhu stvaranja smislenog Weba.

Sljedeću fazu u evoluciji Weba Murugesan (2010) naziva Web 4.0, ili „Web X.0“. Murugesan (2010) također navodi da je osnovni cilj Weba 4.0 postizanje više razine inteligencije na Webu preko softverskih agenata i sustava koji se međusobno razumiju, komuniciraju i surađuju kako bi postigli ciljeve korisnika. Web 4.0 je još nazivan i „inteligentni Web“, ili „pametni Web“, zbog usmjerenosti na interakciju i razumijevanje od strane računala i ljudi.

Obilježavanje svake faze evolucije Weba brojevima kao kod verzija softvera ne znači da je Web 1.0 zamijenjen s Web 2.0, ili da je Web 2.0 zamijenjen s Web 3.0. Umjesto toga, svaka faza obilježava svoje ciljeve, probleme, tehnologije i mogućnosti. Web 1.0, Web 2.0, Web 3.0 i Web 4.0 postoje istodobno, jer je jedan temelj za drugoga. Prvi temelj, Web 1.0 je zato izuzetno bitan. Upravo je u toj fazi došlo do pojave i rasta web portala.

3. Web portal

3.1. Povijest web portala

Web portal je teško precizno definirati jer je sam izraz bio korišten i koristi se u različitim kontekstima i u različitim izvorima. Mogao bi se definirati kao posebna Internet ili intranet stranica oblikovana da djeluje kao ulaz (eng. *gateway*) koji pruža pristup drugim stranicama. Web portal prikuplja informacije iz više izvora i pruža te informacije različitim korisnicima. Također se može definirati i kao stranica za pretraživanje i pristup drugim stranicama, a djeluje kao mjesto za centralizirani pristup relevantnom sadržaju.

Zhou (2003) navodi da povijest sadašnjih web portala seže do Booleove tehnologije pretraživanja razvijene u 1994. i 1995. godini. Uslijed eksponencijalnog rasta web stranica, korisnici nisu mogli pretraživati web dokumente na način da ih pretražuju preko direktorija. Pojavile su se tražilice (eng. *search engine*) koje su nudile sadržaj dokumenata s pripadajućim poveznicama na dokumente u Web okruženju. Zhou (2003) još navodi i da su prvi korisnici Weba uglavnom bili znanstvenici i nastavnici koji su s pozitivnim reakcijama dočekali dolazak tražilica.

Sve više ljudi je počelo koristiti Web, ne samo za istraživanje, nego i za zabavu i prikupljanje svakodnevnih informacija. Ubrzo se je došlo do saznanja da je korisnicima potrebno pružiti učinkovitiji način traženja važnih informacija. Web stranice koje su se usmjerile na pretraživanje informacija su dodale mogućnosti kategorizacije, filtrirajući popularne stranice i dokumente u unaprijed konfigurirane grupe. Kategorizacija je bila korak dalje od obične web stranice, a smatra se ranom fazom portala (Zhou, 2003). Web pruža širok raspon informacija za širok raspon korisnika, a kategorizacija pruža sužen raspon informacija za širok raspon korisnika. Koncept kategorizacije se može pronaći i u televizijskim programima gdje su gledateljima pruženi specijalizirani sadržaji kao npr. sport, vijesti, politika, vrijeme, itd.

Zhou (2003) još navodi i da je tehnologija koja je od tražilica dovela do web portala bila *push* tehnologija. Ideja *push* tehnologije je da se korisnicima informacija s web stranice isporuči direktno na radnu površinu, umjesto da korisnici povlače informaciju s Weba (Shtykh i Jin, 2003). *Push* je opći izraz za softver za isporuku informacije na strani klijenta ili servis koji može biti prilagođen za obavješćavanje korisnika o novim informacijama i dostavljanje

istih na radnu površinu (Guenther, 1999). Guenther (1999) navodi i da je za korištenje *push* tehnologije potrebno na računalo instalirati softver poznat kao *push* klijent koji nudi sučelje za *push* servis.

Neki od uvođenja *push* tehnologije su bili PointCast, Astound, Marimba, BackWeb i InterMind (Collins, 1997). Zhou (2003) navodi da je naziv PointCast tijekom 1995. i 1996. godine predstavljao novi poslovni model na Webu, a opisan je kao vijesti preko Interneta iz različitih izvora prilagođene prema onome što korisnik treba. Naglasak je bio na prilagođenim informacijama izravno na radnu površinu korisnika (Bidgoli, 2004a). PointCast je nudio mogućnost prilagođavanja informacija koje će se ažurirati, bez potrebe za ponovnim traženjem i zahtijevanjem informacija na Internetu (Himmelstein i Siklos, 1999). Himmelstein i Siklos (1999) navode i da je krajem 1996. PointCast brojao 1.5 milijuna korisnika. Nažalost, PointCast nije doživio veliki uspjeh. Sama *push* tehnologija je u 1997. bila kritizirana zbog prevelikih zahtjeva za računalne i mrežne resurse, a loš menadžment je doveo do prodaje PointCast-a (Himmelstein i Siklos, 1999).

Koncept automatske isporuke prilagođenog sadržaja se je preobrazio u tehnologiju web portala (Zhou, 2003). Web portali nude mogućnost pretraživanja iz skupa kategoriziranog sadržaja, a s vremenom su počeli nuditi i mogućnosti kao pristup web stranicama od posebnog interesa. Rani primjeri takvih portala su stranice kao:

- my.fool.com - za informacije iz područja financija,
- travelocity.com - za putne informacije i
- my.weather.com - za vrijeme i lokalne vijesti.

Od kraja 90-ih nadalje je *my* trend nastavio dobivati na popularnosti (Browne, 2008). Poznatije kompanije kao Yahoo nude *my* portal na svojoj web stranici.

3.2. Funkcije web portala

Funkcije web portala predstavljaju komponente koje nude korisnicima pristup informacijama iz različitih izvora. Jalal i Al-Debei (2012) navode da su glavne funkcije web portala sljedeće:

1. **Upravljanje i prilagođavanje sadržaja** - omogućuju korisnicima da dostupne podatke prilagode sebi po svojim zahtjevima i postavkama, a to uključuje

prilagođavanje, personalizaciju, uređivanje profila, upravljanje sadržajem i prezentaciju.

2. **Integracija** - pomoću koje se podaci iz različitih izvora i različitih formata preuzimaju i sinkroniziraju kako bi se prikazali u web portalu.
3. **Sigurnost** - koja pruža siguran pristup različitim izvorima, a da pritom opisuje različite razine pristupa za svakog korisnika, odnosno grupu korisnika i to za svaku komponentu portala i softversku funkciju uključenu u portal.
4. **Pretraživost** - daje mogućnost dohvata zahtijevane informacije izravno preko korištenja tražilica, umjesto traženja informacija na način da se informacija pregledava po različitim kategorijama.
5. **Kolaboracija** - koja nudi kolaborativne alate kako bi se rad optimizirao i kako bi se postigao zadani cilj.
6. **Skalabilnost** - koja opisuje sposobnost sustava da se nosi s povećanim radnim opterećenjem i da se pritom nastavi izvoditi.
7. **Dostupnost** - opisuje sposobnost sustava da bude dostupan za pristup bilo gdje, bilo kada.

Svaka navedena funkcija pomaže korisnicima u postizanju zadanog cilja. Upravljanje i prilagođavanje sadržaja i pretraživanje pomažu u dohvat i prilagođavanju sadržaja na portalu, čime se dobivaju željene informacije u željenom formatu koje mogu biti i temelj za odlučivanje. Integracijom se mogu uskladiti važni podaci iz drugih aplikacija i time postići brži dohvat informacija koje su tada sve na jednom mjestu. Kolaborativnim alatima se korisnici mogu približiti kolegama i time ostvariti komunikaciju na jednostavniji način, a i postići zajedničke ciljeve i riješiti zajedničke probleme. Dostupnost daje mogućnost pristupa web portalu bilo gdje i u bilo koje vrijeme, što služi kao poticaj da se obavi bilo koji posao integriran u portal ili dohvate informacije s portala. Uz sve to je važna sigurnost, koja pruža siguran pristup informacijama, što pridonosi korištenju web portala kao alata koji pruža potrebne informacije.

3.3. Vrste web portala

Iako ne postoji konačna kategorizacija web portala, različiti izvori nude svoje popise u kojima su neke vrste portala zajedničke, nekad i pod različitim nazivom. Neki izvori navode veći broj vrsta web portala, a neki manje, smatrajući da nekoliko vrsta spada pod jednu.

Kategorizacije web portala su većinom temeljene na sadržaju portala i ciljanim korisnicima. Tatnall (2005) navodi 9 vrsta web portala: opći, vertikalni, horizontalni, web portali zajednica, korporativni web portali, web portali e-tržišta, mobilni, informativni i specijalizirani web portali.

3.3.1. Opći web portali

Web portali mogu nuditi poveznice na web stranice koje mogu biti ili usko povezane ili dosta različite. U slučaju općih web portala se radi o pružanju poveznica na sve vrste web stranica, koje god korisnici odaberu. Tatnall (2005) navodi da je većina općih web portala razvijena iz jednostavnih tražilica (kao npr. Yahoo), pružatelja Internet usluga (kao npr. AOL) i servisa e-pošte (kao npr. Hotmail).

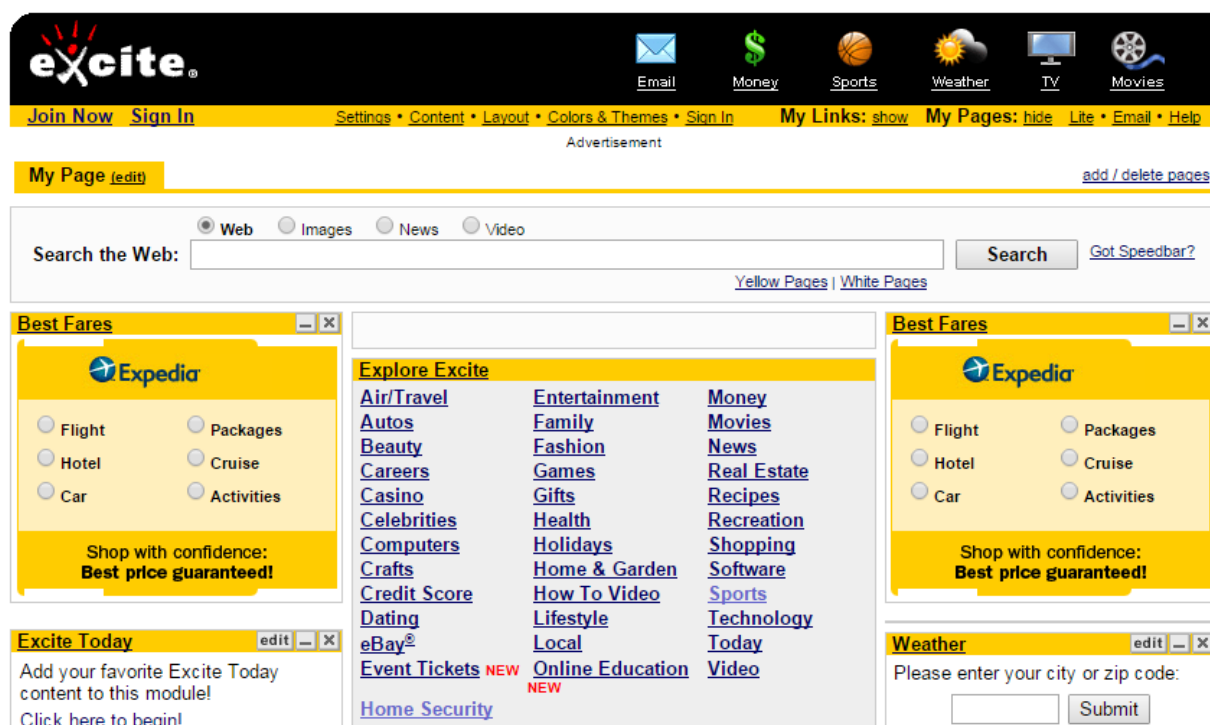
Važan cilj općih web portala je postati stranica kojoj se korisnici vraćaju nakon svake potrebe za pristupom nečeg na Webu. Također je poželjno za opći web portal da pruža većinu servisa, informacija i poveznica koje korisnici žele. Opći web portali često sljedeće servise: besplatna e-pošta, poveznice na tražilice i kategorije informacija, servisi članstva, vijesti, članke i naslove iz poslovnog svijeta, personalizirani prostor s odabranim komponentama portala, poveznice na virtualne trgovačke centre, itd. Način zarade kod općih web portala je većinom reklamiranje, a uspjeh portala ovisi o količini prometa posjetitelja, što uključuje privlačenje novih posjetitelja, zadržavanje posjetitelja na portalu čim duže moguće i uvjeravanje posjetitelja da ostanu na portalu.

Glavni koncept općih web portala je približiti korisnicima najvažnije web resurse na brz i jednostavan način (Hock, 2005). Pri tome opći web portali često imaju sljedeće karakteristike: raznovrsnost korisnih alata, pozicioniranje kao početna stranica i personalizacija. Također se nude alati koji omogućuju jednostavan pristup informacijama koje su često potrebne, pa opći web portali često uključuju te alate na svojim početnim stranicama. Veliki broj općih web portala je zamišljen tako da ih korisnici odaberu kao njihovu početnu stranicu web preglednika, a ta početna stranica se može personalizirati što daje korisnicima izbor da odaberu koje točno kategorije vijesti prikazivati, za koji grad prikazivati vrijeme, koje točno dionice, itd.

Za koji će se opći web portal korisnici odlučiti često ovisi o sadržaju koji portal nudi i prezentaciji samog sadržaja. Većina popularnih općih web portala nudi veliki broj mogućnosti koje svaki portal može prikazati i ponuditi na svoj jedinstven način. Jedan od općih web

portala koji je bio popularan 90-tih, a još uvijek je aktivno korišten je Excite (excite.com). Na slici 1 se vidi snimka zaslona na kojoj se nalazi početna stranica portala Excite, a vidljive su i tipične karakteristike općih web portala:

- početna stranica koja cilja na poticanje korištenja portala kao početne stranice web preglednika,
- ključne riječi *my* koje ukazuju na to da se portal može personalizirati,
- raznovrsne kategorije sadržaja s poveznicama,
- nekoliko integriranih alata kao npr. tražilica i rezervacija putovanja, itd.



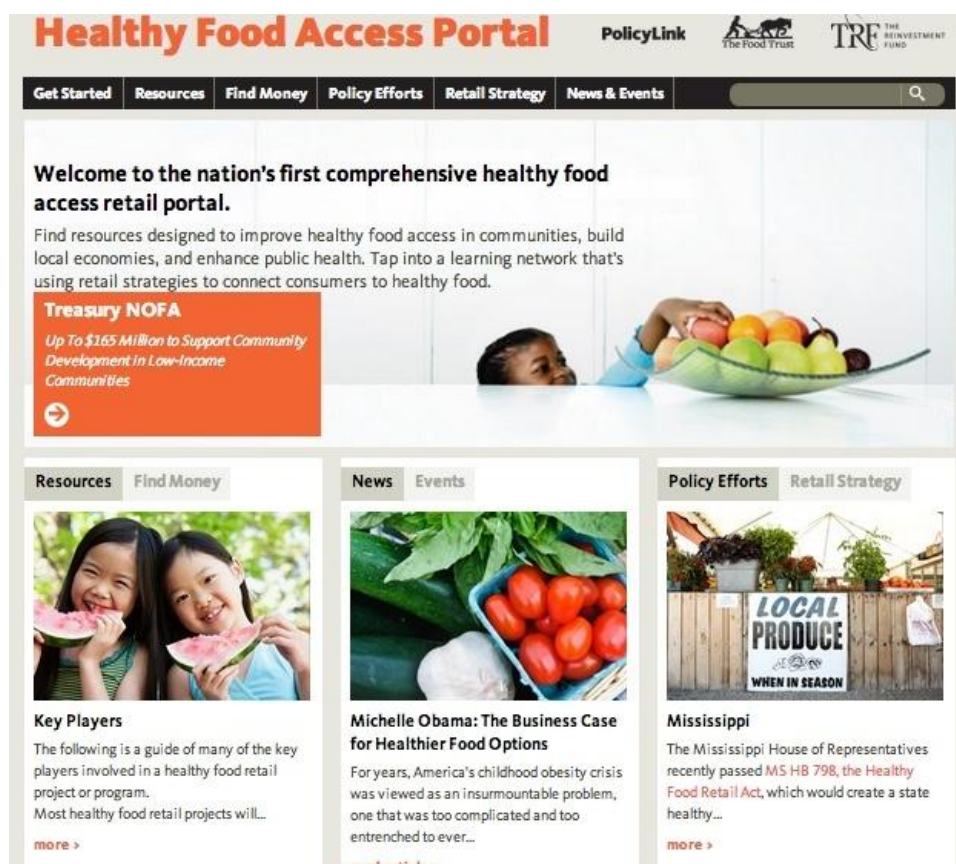
Slika 1: Početna stranica web portala Excite

3.3.2. Vertikalni web portali

Vertikalni web portali (u nekim izvorima poznatiji kao vortali) su obično temeljeni na nekoj specifičnoj industriji, a glavni cilj im je prikupljati informacije koje su važne nekim posebnim grupama usko povezanih industrija u svrhu omogućavanja razmjene informacija na specifičnom tržištu (Mansourvar i Yasin, 2010). Vertikalni web portali se često specijaliziraju u određenim robama i materijalima kao što su kemikalije, metali, naftni i drvni proizvodi (Tatnall, 2005). Neki se specijaliziraju u uslugama kao što su hrana i transport, a neki i u nekim zanimljivim područjima kao kampiranje, planinarenje, sportska oprema, itd.

Vertikalni web portali ciljaju na uzak raspon usluga i proizvoda, a pošto se većina industrija specijalizira u nekom području, te industrije se smatraju vertikalnima. Bilo koja vertikalna industrija spaja ljude koji dijele interes za informacije o toj specifičnoj industriji, a te informacije mogu brže i jednostavnije pronaći preko vertikalnih web portala. Za jedno područje obično postoji više vertikalnih portala. Korisnici se nekad mogu pronaći u situaciji da u rezultatima pretraživanja za nekom vertikalnom industrijom pronađu i stotine vertikalnih portala.

Vertikalne web portale je jednostavno prepoznati. Iako su im komponente slične kao i kod ostalih vrsta portala, kategorije informacija su usmjerene na samo jednu industriju, odnosno teme o toj industriji. Takvi portali će obično sadržavati vijesti, popise događaja, aukcije, grupe za diskusije, statistike i istraživanja, poveznice i alate vezane uz samo jednu industriju. Healthy Food Access Portal (healthyfoodaccess.org), prikazan na slici 2, je primjer vertikalnog web portala jer su članci, alati, poveznice, novosti i događaji na portalu vezani samo uz zdravu prehranu.

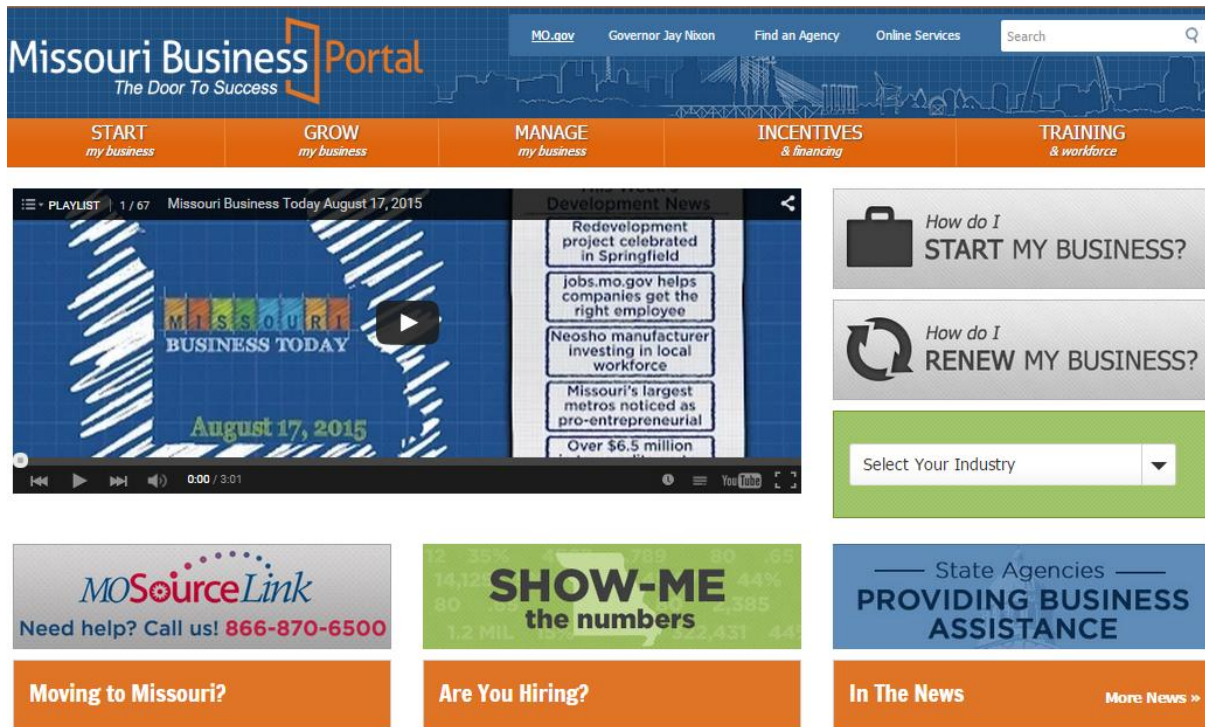


Slika 2: Healthy Food Access Portal (Izvor: <http://yogarose.net/wp-content/uploads/2013/02/HealthyFoodAccessPorta.jpg>, 2013)

3.3.3. Horizontalni web portali

Za razliku od vertikalnih web portala koji sadrže teme vezane uz jednu industriju, horizontalni web portali su temeljeni na širokom rasponu industrija. Horizontalni web portali obično ciljaju na zadovoljavanje različitih potreba korisnika jer uključuju različite industrije (Tatnall, 2005). Također se mogu usmjeriti i na objedinjavanje više povezanih industrija, a mogu i integrirati jedan ili više vertikalnih web portala. Zbog većeg raspona dostupnog sadržaja, horizontalni web portali korisnicima obično daju veće mogućnosti personalizacije i prilagodbe nego vertikalni.

U nekim slučajevima su horizontalni web portali razvijeni na način da korisnicima ponude različite teme iz različitih industrija na temelju područja u kojem žive. Primjeri toga mogu biti portali neke grupe industrija u nekoj regiji države, ili portali vezani uz veliki raspon industrija na području nekog grada. Postoje i različiti *business-to-business* horizontalni web portali koji se upotrebljavaju u poslovne svrhe (Al-Qirim, 2004). Na slici 3 se vidi snimka zaslona početne stranice portala Missouri Business Portal (business.mo.gov), tipičnog primjera horizontalnog web portala, koji pruža informacije i alate vezane uz različite industrije na području savezne države Missouri.



Slika 3: Početna stranica web portala Missouri Business Portal

3.3.4. Web portali zajednica

Web portale zajednica osnivaju zajednice, ili su temeljeni na posebnim grupama koje imaju isti interes. Tatnall (2005) navodi da takvi portali pokušavaju poticati koncept virtualne zajednice u kojoj svi dijele zajednički interes ili lokaciju, te ovisno o orijentaciji nude odgovarajuće usluge. Preko web portala zajednice članovi zajednice mogu doprinijeti informacijama ili mogu pristupiti informacijama drugih članova zajednice, ili pristupiti informacijama iz ostalih izvora koji su od interesa članovima zajednice, odnosno korisnicima portala. Pošto su izvori informacija često različiti, ovakvim portalima se obično dodaje funkcija integracije više različitih izvora.

Web portalima zajednica je najčešće glavni cilj pružiti informacije članovima zajednice, potaknuti članove da doprinose svojim informacijama, a da pritom omoguće komunikaciju i interakciju članova. Komunikacija se na ovakvim portalima obično odvija preko foruma, personaliziranih stranica članova, e-pošte, čavrljanja (eng. *chat*) i ostalih oblika komunikacije. Neki od web portala zajednica su:

- linux.com - sadrži informacije o događajima, vijesti, uputstva, forume, informacije o knjigama, vodiče i ostale informacije vezane uz Linux zajednicu,
- sharepointpals.com - portal namijenjen zajednici programera, administratora i korisnika SharePoint platforme,
- dosomethingnearyou.com.au - omogućuje uključanje u zajednicu volontera i sudjelovanje na dobrotvornim događajima u Australiji,
- tennis.my - portal teniske zajednice u Maleziji, čija je početna stranica prikazana na slici 4, a komunikacija se na portalu većinom odvija preko foruma i čavrljanja,
- blogeri.hr - portal namijenjen hrvatskoj blog zajednici...



Slika 4: Početna stranica web portala tennis.my

3.3.5. Korporativni web portali

Korporativni web portali (ili web portali poduzeća) se često primjenjuju kada postoji potreba za jednostavnijim upravljanjem informacijama unutar poduzeća. Ovakvi portali su najčešće oblikovani za *business-to-employee* procese i nude zaposlenicima sredstva za pristup i dijeljenje podataka i informacija unutar poduzeća (Tatnall, 2005). Tatnall (2005) još navodi i da korporativni web portali mogu uključivati sljedeće komponente: kategorizaciju informacija koja je dostupna na intranetu, tražilicu koja pokriva cijeli intranet, vijesti vezane uz poduzeće, pristup e-pošti, pristup zajedničkim aplikacijama, upravljanje dokumentima, poveznice na interne stranice i popularne web stranice i mogućnost personaliziranja stranica.

Obično je glavni cilj korporativnih web portala pružiti poduzeću cjelovito upravljanje razmjenom informacija i znanja, te integrirati servise i aplikacije poduzeća (Benbya i sur., 2004). Benbya i sur. (2004) još navode i da se pri tome zaposlenicima poduzeća pruža pristup potrebnim informacijama na jednostavan i brz način. U nekim slučajevima je od velike važnosti pružiti pristup poslovnim podacima i procesima bilo gdje i bilo kada. Kod integracije je potrebno uključiti sve baze podataka, ERP sustave, sustave za upravljanje dokumentima i ostale informacijske sustave. Također je važno unaprijediti sustav sigurnosti, s naglaskom na autentikaciju i autorizaciju korisnika i pružanje pouzdanih sigurnosnih mehanizama (White, 2011).

Korporativni web portali mogu biti razvijeni u obliku portala za poslovnu inteligenciju koji služe kao potpora procesima odlučivanja ili u obliku portala koji nude potporu za specifične poslovne procese kao npr. upravljanje ljudskim resursima, upravljanje lancem opskrbe, potpora prodaji, itd. Također mogu biti razvijeni kao složenije poslovno rješenje koje uključuje više internih portala od kojih svaki predstavlja neki odjel u poduzeću.

Na slici 5 se može vidjeti primjer korporativnog web portala koji je sadržan od komponenti koje većina ovakvih portala ima. Neke komponente u ovom primjeru uključuju obrazac za prijavu u sustav za zaposlenike, kategorije informacija koje su dostupne u obliku izbornika, poveznice na interne i eksterne stranice, vijesti vezane uz poduzeće i mogućnost personalizacije.



Slika 5: Primjer korporativnog web portala (Izvor:

docs.oracle.com/cd/E13292_01/pt849pbr0/eng/psbooks/tptr/img/image10.gif, 2015)

3.3.6. Web portali e-tržišta

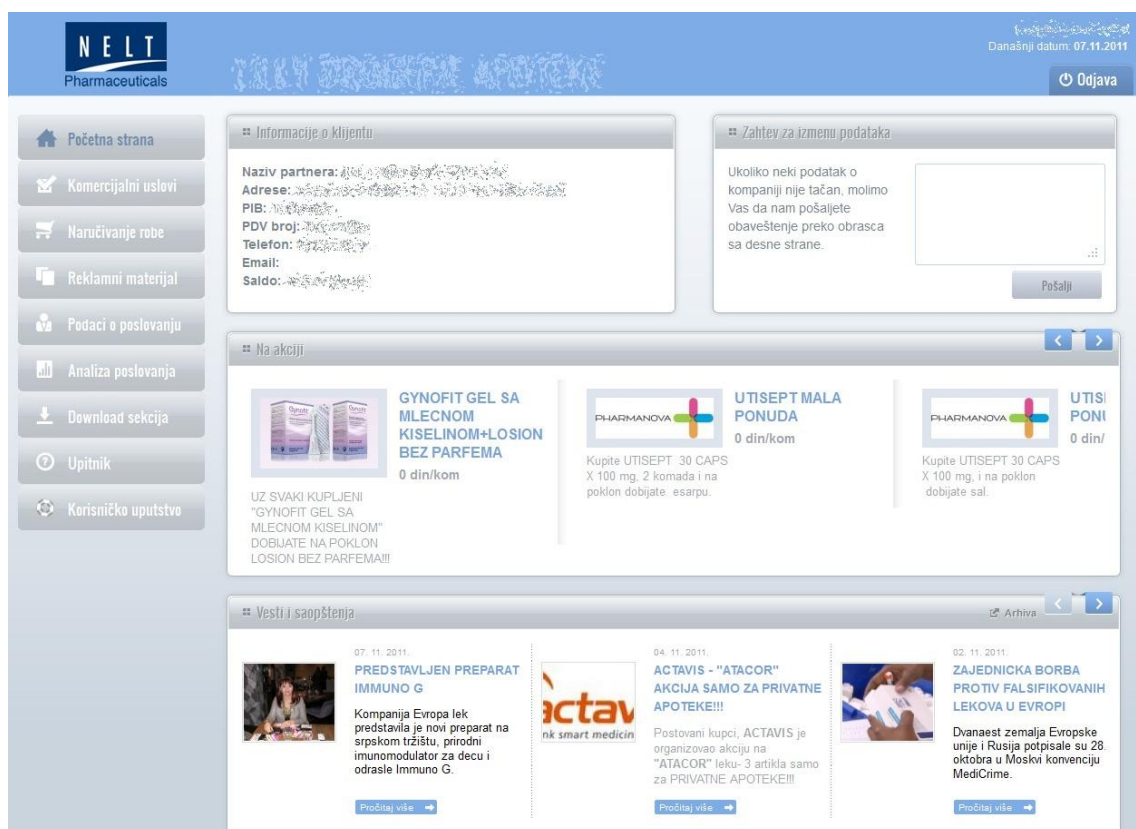
Web portali e-tržišta djeluju kao proširena verzija korporativnih web portala, a često nude pristup ektranet uslugama poduzeća i korisni su za *business-to-business* procese kao

npr. nabava robe (Tatnall, 2005). Ekstranet uslugama poduzeće na svojem intranetu daje djelomični pristup vanjskim autoriziranim korisnicima i omogućuje poduzećima da razmjenjuju informacije na siguran način (Bidgoli, 2004b). Obično je glavni cilj web portala e-tržišta ostvariti efikasnije poslovanje između poduzeća.

Neke od mogućnosti ovakvih portala su sustavi transakcija, e-katalozi, e-aukcije, e-pregovori, novosti koje su od interesa za poduzeće i poslovne partnere i ostale informacije koje su važne za poslovanje. Osim toga, web portali e-tržišta obično omogućuju tražilicu i personalizaciju. Web portali e-tržišta mogu biti razvijeni kao:

- portali koji služe kao posrednici između poduzeća i dobavljača u svrhu ostvarivanja efikasnije narudžbe i dostave robe,
- proširenja na korporativni portal sa sustavima naplaćivanja robe ili usluga,
- portali čiji je glavni cilj ostvariti efikasniji odnos s klijentima, itd.

Na slici 6 se može vidjeti primjer web portala e-tržišta.



Slika 6: Primjer web portala e-tržišta (Izvor:

http://www.tehnicomsolutions.com/genera/app/webroot/upload/Portfolio/Images/2011_11/nelt2.jpg, 2011)

3.3.7. Mobilni web portali

Zbog rastućih trendova u mobilnom računalstvu, mobilni web portali su se sve više počeli koristiti u mobitelima i ostalim mobilnim uređajima kao što su npr. tablet računalo i dlanovnik (eng. *Personal Digital Assistant - PDA*) (Tatnall, 2005). Mobilni web portali su najčešće po glavnim funkcijama slični općim web portalima, ali uvelike se razlikuju izgledom jer su optimizirani za mobilne uređaje. Kod optimizacije se posebno mora obratiti pozornost na različitost u web preglednicima, samim mobilnim uređajima i veličinama zaslona (Balachandar, 2012). Zbog povećane popularnosti mobilnih uređaja, počeo se razvijati sve veći broj mobilnih verzija postojećih portala. Glavni cilj većine mobilnih web portala je pružiti jednostavan i brz pristup informacijama kao što su vijesti, vrijeme, poveznice na ostale stranice i servisi e-pošte i čavrljanja.

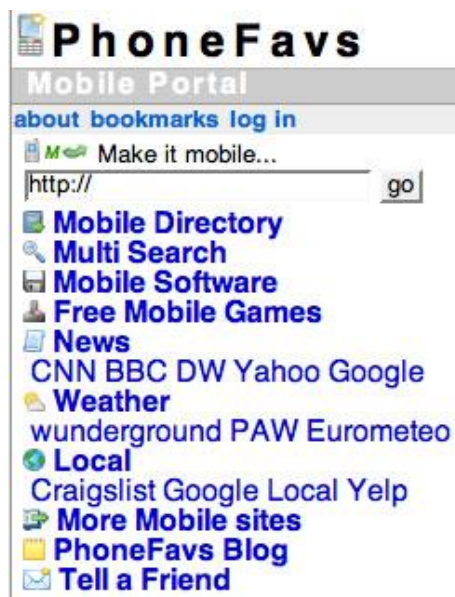
Mobilni web portali mogu biti namijenjeni i u poslovne svrhe, pri čemu se stavlja naglasak na komponente koje je jednostavno koristiti i koje su učinkovite, a sam portal bi trebao privući i zadržati korisnike, te im pružiti pojednostavljenu integraciju aplikacija i pružatelja sadržaja kako bi na kraju mogli dobiti prednost na tržištu. Pri tome bi se također klijentima trebalo omogućiti prilagođavanje sadržaja i pristup sadržaju na siguran način. Zbog povećane različitosti u mobilnim uređajima i platformama se također pažnja daje optimizaciji mobilnih web portala.

Tipični mobilni web portali su oblikovani na način da izgledaju kao smanjena i minimalizirana verzija portala. Obično imaju veće ikone ili vertikalne izbornike za jednostavnu i brzu navigaciju. Također mogu imati i kartice za brzu promjenu sadržaja. Velika većina mobilnih web portala ima komponente organizirane vertikalno.

Neki od primjera mobilnih web portala su:

- m.aol.com - mobilna verzija popularnog AOL portala,
- yeswap.com - portal isključivo namijenjen za poveznice na ostale mobilne web portale i mobilne verzije web stranica i
- twilightwap.com - mobilni web portal s bogatim sadržajem, tražilicom, servisom čavrljanja, poveznicama na korisne alate, igre i sve ostalo optimizirano za mobilne uređaje.

Na slici 7 je snimka zaslona mobilnog web portala PhoneFavs, koji cilja na jednostavno korištenje, bogati sadržaj, veliki broj poveznica i nekoliko korisnih alata optimiziranih za mobilne uređaje.



Slika 7: Mobilni web portal PhoneFavs (Izvor: <http://www.berryreview.com/wp-content/uploads/2008/04/phonefavs.jpg>, 2008)

3.3.8. Informativni web portali

Iako se za informativne web portale može smatrati da spadaju pod neku drugu vrstu web portala, može ih se promatrati i kao zasebnu vrstu čiji je glavni cilj pružiti neku specifičnu vrstu informacije (Tatnall, 2005). Najčešće su kod ovakvih portala to vijesti ili općenite informacije i korisni savjeti o nekoj temi kao npr. zdravlje. U nekim slučajevima pružanje informacija korisnicima može biti i jedina funkcija na portalu. Neki informativni web portali nude mogućnost stvaranja sadržaja, a neki za pristup određenom sadržaju zahtijevaju novčanu naknadu.

Najpopularniji oblik informativnih web portala su zasigurno portali za vijesti, ali osim takvih postoje i portali koji su razvijeni u obliku web udžbenika, kao npr. Alpha History (alphahistory.com), besplatni web udžbenik s velikim brojem bogatog sadržaja, poveznica i izvora vezanih uz povijest, a namijenjen je i studentima i profesorima. Još jedan primjer informativnog web portala je ESPN (espn.com), čiji je glavni cilj pružiti informacije o sportu. Primjer hrvatskog informativnog web portala je Legalis (legalis.hr), pravni web portal čiji je glavni cilj pružiti pravne informacije. Početna stranica portala Legalis je prikazana na slici 8.

Neki od najpopularnijih hrvatskih portala za vijesti uključuju index.hr, jutarnji.hr, net.hr, monitor.hr, 24sata.hr, dnevnik.hr, tportal.hr i ostale.



Slika 8: Početna stranica web portala Legalis

3.3.9. Specijalizirani web portali

Specijaliziranim web portalima je glavni cilj zadovoljiti specifične segmente tržišta (Housley, 2005). Tatnall (2005) navodi da se u velikom broju slučajeva ovakvi portali mogu kategorizirati i kao informativni web portali. Već spomenuti ESPN je ciljan korisnicima koje zanima sport. Prema tome je ESPN i specijalizirani web portal jer je namijenjen samo specifičnom segmentu tržišta. Tema specijaliziranih web portala može biti i neki specifični proizvod unutar pojedine grupe proizvoda, kao npr. portal o cross motorima. Osim toga, česti su specijalizirani web portali koji su namijenjeni specifičnim korisnicima prema dobi, npr. portali za tinejdžere kao što je teenink.com ili portali za djecu kao što je stokoluri.org, čija je početna stranica prikazana na slici 9. Kako se kod ovakvih portala uvijek radi o nekom specifičnom segmentu tržišta, u nekim slučajevima mogu imati i e-trgovinu sa specifičnim proizvodima ili uslugama.



Slika 9: Početna stranica web portala stokoluri.org

Kod specijaliziranih web portala se prilagođavanje sadržaja može pokazati izuzetno korisnim. Većina korisnika na Webu nakon kratkog vremena postaje nestrpljiva, pogotovo kada duže vrijeme traže potrebnu informaciju ili kada su pred njima velike količine nepotrebnog sadržaja. Tada bi se trebala ponuditi personalizacija sadržaja tako da se sadržaj može prilagoditi i da se mogu dodati informacije koje su važne, prema osobnom ukusu promijeniti izgled portala, ukloniti nepotrebne komponente, itd.

3.4. Zadovoljstvo korisnika web portalima

Web portali uglavnom ciljaju na to da služe kao početna točka za korisnike sa specifičnim interesima. Prema tome bi zadovoljstvo korisnika trebalo predstavljati najvažniji prioritet za sve koji su dio razvoja i funkcioniranja web portala. Čimbenici koji utječu na zadovoljstvo korisnika web portalima uključuju sadržaj web portala, dizajn web portala, personalizaciju web portala i podršku zajednicama web portala (Sampson i Manouselis, 2005).

3.4.1. Zadovoljstvo sadržajem

Sadržaj je izuzetno važan dio svakog web portala, a zadovoljstvo korisnika sadržajem se može promatrati na nekoliko područja (Sampson i Manouselis, 2005):

1. **Zadovoljstvo organizacijom sadržaja** - odnosi se na kategorizaciju informacija kako bi pretraživanje i preuzimanje informacija bilo efikasnije. Kvalitetni web portali imaju servise za organizaciju sadržaja kao što su direktoriji, koji omogućavaju pregledavanje i pretraživanje. Osim toga postoje i dobro organizirani skupovi kategoriziranog sadržaja. Iako je usko povezana s arhitekturom informacija na web portalu, klasifikacija sadržaja u prikladne kategorije interesa se više odnosi na logičku organizaciju, nego na fizičku.
2. **Zadovoljstvo vjerodostojnošću sadržaja** - odnosi se na povjerenje i pouzdanost informacija i pružatelja sadržaja, a može imati nekoliko aspekta. Neki od njih su točnost i jasnoća sadržaja, te pouzdanost, priznanje i ugled autora ili pružatelja sadržaja.
3. **Zadovoljstvo korisnošću sadržaja** - usredotočuje se na pažnju na sadržaj, upotrebu prikladnog jezika i korisnost informacije prema potrebama korisnika. Usko je povezana s vjerodostojnošću sadržaja.
4. **Zadovoljstvo integracijom sadržaja** - odnosi se na servise sadržaja koji su povezani s integracijom vanjskih izvora informacija i s pružanjem poveznica na vanjske izvore.

3.4.2. Zadovoljstvo dizajnom

Čimbenik koji utječe na zadovoljstvo korisnika je zasigurno dizajn web portala, jer je usko povezan s osnovnim principom općenitog web dizajna. Dizajn web portala se može promatrati na sljedećim važnim područjima (Sampson i Manouselis, 2005):

1. **Zadovoljstvo arhitekturom informacija** - važno područje koje se veže uz pitanja o organizaciji informacija na portalima kao što su struktura, grupiranje i označavanje informacija. Usko je povezana s prethodno spomenutom organizacijom sadržaja, no u ovom kontekstu se promatra s perspektive dizajna sustava, što može uključivati funkcionalnosti, hijerarhije, baze podataka, strukturu stranice, itd.
2. **Zadovoljstvo iskoristivošću** - također važno područje zadovoljstva korisnika dizajnom web portala, a odnosi se na interakciju i kretanje korisnika po portalu. Obično se razmatra nekoliko ključnih principa web iskoristivosti kao

što su jednostavnost, konzistentnost, relevantnost, fleksibilnost, sprječavanje grešaka, oporavak od grešaka, itd.

3. **Zadovoljstvo grafičkim dizajnom** - područje vrijedno posebnog razmatranja jer grafički dizajn pridonosi tome da web portal izgleda privlačno, te potiče korisnike da čim duže koriste web portal. U nekim slučajevima je nakon vremena potrebno promijeniti grafički dizajn web portala, no bez velikog utjecaja na funkcioniranje portala.
4. **Zadovoljstvo performansama** - područje koje je vezano uz ispravno funkcioniranje web portala, a bavi se pitanjima tehničkih performansi kao što su dostupnost, stabilnost sustava, kompatibilnost različitih web preglednika, neaktivne poveznice, itd.

3.4.3. Zadovoljstvo personalizacijom

Da bi se na web portalima korisnicima udovoljilo njihovim potrebama i postavkama, posebnu pažnju treba dati personalizaciji. Detaljnije se mogu promatrati sljedeća područja zadovoljstva korisnika personalizacijom (Sampson i Manouselis, 2005):

1. **Zadovoljstvo personalizacijom navigacije** - odnosi se na podešavanje navigacijskih mehanizama i funkcija prema različitim potrebama. Korisnici obično žele izdvojiti koje su komponente navigacije korisne za njih, a koje beznačajne.
2. **Zadovoljstvo personalizacijom sadržaja** - vezano je uz obavješćavanje korisnika o novom sadržaju koji im je od interesa i pružanju informacija koje su prilagođene samo za njihove potrebe i koje odgovaraju postavkama.
3. **Zadovoljstvo personalizacijom sučelja** - područje koje se veže uz pitanja prilagođavanja sučelja različitim potrebama i postavkama.

3.4.4. Zadovoljstvo podrškom zajednicama

Podrška zajednicama je postala jedan od osnovnih dijelova web portala. Zadovoljstvo korisnika podrškom zajednicama se može promatrati na dva područja (Sampson i Manouselis, 2005):

1. **Zadovoljstvo podrškom komunikaciji** - odnosi se na alate i servise koje koriste članovi virtualne zajednice kako bi ostvarili komunikaciju. Neki od takvih su forumi, servisi čavljanja, interesne grupe, oglasne ploče, itd.
2. **Zadovoljstvo podrškom kolaboraciji** - odnosi se na alate i servise koji su potrebni kako bi se omogućila efikasna i učinkovita kolaboracija između korisnika. Neki od takvih alata i servisa su videokonferencije, dijeljenje dokumenata, e-pošta, organizacija zadataka, kalendar, itd.

4. Personalizacija web portala

4.1. Pojava i uspon personaliziranih web portala

Kao što je spomenuto, personalizacija je važan čimbenik koji utječe na zadovoljstvo korisnika web portalima, a s vremenom je postala jedan od najvažnijih trendova na Webu (Brecht, 2013). Glavni cilj personalizacije je isporučiti informaciju koja je važna pojedincu ili grupi pojedinaca u određenom obliku, rasporedu i vremenskim intervalima (Kim, 2002).

S pojavom i ubrzanim razvojem Weba u 1990-tim godinama se je počela razvijati i personalizacija. Web je ljudima s različitim ciljevima i karakteristikama putem portala počeo nuditi pristup sve većoj količini informacija. No, s time je došlo i do pitanja isporuke prave informacije, u pravom obliku i u pravo vrijeme. Važnost personalizacije su prepoznali prvi popularni web portali, a među njima je bio Yahoo, koji je dodao *my* opciju na svoj portal, nudeći mogućnost prilagođavanja opcija servisa koji su se nudili na portalu. Ubrzo je i većina tada popularnih web portala počela nuditi svoje varijante personalizacije.

Problem prevelike količine informacija i servisa koje su portali nudili i prikazivali na svojim početnim stranicama je riješen personalizacijom. Korisnici su sebi mogli filtrirati ono što ne žele vidjeti, postaviti nekoliko opcija koje su im odgovarale i odrediti kategorije informacija koje su ih zanimale. Jedna od prvih mogućnosti personalizacije je uključivala i određivanje broja prikazivanja vijesti iz dostupnih izvora. Jacso (1999) navodi da su tadašnji portali svi bili potpuno slični *my* verziji Yahoo-a. Nekoliko je web portala nudilo i dodatne opcije prilagođavanja, koje su uključivale razmještaj komponenata i mijenjanje boja na portalu po želji. U samo nekoliko koraka se je moglo odrediti hoće li neka komponenta biti na desnoj ili lijevoj strani, te odabrati predefinirane predloške boja ili kombinirati boje prema željama. Uskoro su i ostali web portali počeli prepoznavati personalizaciju kao jedan od najutjecajnijih čimbenika zadovoljstva korisnika web portalima.

4.2. Elementi personalizacije web portala

Da bi se personalizacija uspješno primijenila na web portal, prvo je potrebno identificirati elemente portala koji se mogu personalizirati. Svaki portal se sastoji od elemenata koji moraju zajedno funkcionirati. Gyara i Sachdev (2007) navode da se ti elementi mogu kategorizirati u dvije vrste: strukturalne i funkcionalne elemente.

4.2.1. Strukturalni elementi

Strukturalni elementi web portala se odnose na elemente kojima se uspostavlja stil, raspored, skup akcija korisnika, izgled i osjećaj portala. Gyara i Sachdev (2007) još navode i da se strukturalni elementi mogu razlikovati od portala do portala, no sljedećih nekoliko je najčešće korišteno.

4.2.1.1. Izgled i osjećaj

Izgled i osjećaj za portal predstavljaju općeniti nastup koji će uvelike djelovati na korisnika i utjecati na korisničko iskustvo. Izgled se odnosi na općeniti dizajn portala ili dijela portala koji može uključivati font, raspored, margine, pozadinu, boje, te ostale vizualne elemente, a osjećaj se odnosi na sveukupni osjećaj korištenja portala na kojeg utječe izgled.

Iako je važno da portal zadrži konzistentni izgled i osjećaj, personalizacija ovih elemenata može privući i zadržati veliki raspon korisnika, te im pružiti pozitivno iskustvo. Time se općeniti izgled i osjećaj portala može prilagoditi različitim korisnicima s različitim karakteristikama i potrebama pri čemu postoji više pristupa. Korisnicima portala se može izravno dati mogućnost promjena vizualnih elemenata ili se isti mogu predefinirano personalizirati za određenu grupu korisnika, što se često koristi kad su na portalu definirane različite uloge korisnika. Primjerice, cjelokupni izgled i osjećaj nekog sportskog portala se može prilagoditi različitim ulogama koje mogu uključivati trenere, preprodavače sportske opreme, aktivne sportaše, ili ostale korisnike koji dijele interes za sportske vijesti ili rezultate.

4.2.1.2. Zaglavlje i podnožje

Ovi elementi portala imaju najrazličitija svojstva. Prilikom posjete web portala, korisnici će vrlo vjerojatno prvi pogled usmjeriti na zaglavlje koje se obično sastoji od logotipa, izbornika, poveznica, raznih opcija, kategorija informacija, tražilice, itd. Dno stranice, na kojem se nalazi podnožje, je mjesto na koje korisnici portala neće previše trošiti vrijeme. Iako je podnožje manje važniji dio portala, korisnici to malo vremena na dno stranice izdvajaju tražeći tipku za povratak na vrh stranice, kontakt podatke, općenite informacije o portalu, opcije koje nisu mogli pronaći na ostatku stranice, pravne informacije, itd.

Obično je slučaj da je korisnicima portala većina elemenata iz zaglavlja ili podnožja nepotrebna i jednostavno nije namijenjena njihovim potrebama, pa je u tim slučajevima dobar

pristup personalizirati zaglavlje i podnožje na temelju karakteristika korisnika ili grupe korisnika.

4.2.1.3. Navigacija

Navigacijski elementi portala bi trebali biti personalizirani kako bi se omogućio pristup najvažnijim dijelovima portala i to uz minimalni broj klikova. Neki od elemenata koji se mogu personalizirati je popis brzih poveznica koji bi omogućio određivanje često posjećenih stranica stvarajući time brz i jednostavan pristup istima. U nekim slučajevima bi se popis brzih poveznica mogao izraditi i dinamički, na temelju prethodnih aktivnosti korisnika na portalu.

Popis nedavno pregledanih stavki također može pomoći pri navigaciji. Na portalima za vijesti to mogu biti nedavno pregledane vijesti, a na e-trgovinama su to nedavno pregledani artikli. Elementi portala koji se mogu personalizirati bi mogla biti i predefinirana područja portala koja služe za konzistentni prikaz ključnih informacija, a obično uključuju promotivne elemente kao što su ponude i reklame, prodajne elemente kao što su povezani artikli kod e-trgovina i obavijesti korisnicima kao što su informacije o statusu narudžbe i broju artikala u košarici. Takvi elementi web portala bi trebali biti personalizirani i relevantni korisnicima.

4.2.2. Funkcionalni elementi

Funkcionalni elementi su elementi na web portalu koji predstavljaju specifični sadržaj ili funkcionalnost portala. Kao i strukturalni, mogu se razlikovati od portala do portala, no sljedećih nekoliko je najčešće korišteno (Gyara i Sachdev, 2007).

4.2.2.1. Stranice i sadržaj

Svaki web portal se sastoji od skupa stranica od kojih svaka sadrži dijelove informacija i funkcionalnosti ili prikazuju neki sadržaj. Svaki od elemenata stranice, uključujući i samu stranicu se može personalizirati na više načina.

Korisnicima se može dati mogućnost prikazivanja, odnosno skrivanja nekog elementa. To može biti od velike koristi ako je prikazan neki element koji korisnicima nije od važnosti, pa se jednostavno može sakriti, te naknadno otkriti, kao npr. skrivanje nekog elementa portala za vijesti koji sadrži neku vijest za koju je korisnik odlučio da nije dovoljno važna za pročitati. Isto može biti korisno i kod korporativnog web portala, gdje bi se mogla sakriti neka

funkcionalnost koja se ne koristi, a nepotrebno oduzima pažnju ili zauzima previše mjesta na stranici portala. Također se može ponuditi mogućnost da se neki elementi web portala minimaliziraju, najčešće na način da se njihov sadržaj ili funkcionalnost sakriju, a ostane prikazan samo naslov.

Premještanje elemenata po prostoru portala je još jedan način personalizacije stranice i sadržaja. U nekim slučajevima je premještanje elemenata na drugačije lokacije unutar prostora portala najbolji način da korisnici postave važnost tih elemenata onako kako im odgovara. Primjerice, ako neki portal za vijesti prikazuje vijesti po kategorijama organiziranim u zasebnim prozorima, onda će pozitivnom korisničkom iskustvu zasigurno pridonijeti mogućnost da korisnike koje zanima politika premjeste prozor s vijestima o politici na vrh, ili na bilo koje željeno mjesto. Sukladno tome, korisnici će htjeti manje važne elemente premjestiti na manje istaknutu lokaciju na portalu.

Prikaz relevantnih elemenata na temelju odabira isto može biti dio personalizacije web portala. To uključuje odluku o tome koje relevantne elemente korisnicima prikazati iz dobivenog skupa izbora ili aktivnosti korisnika. Kao i kod izgleda i osjećaja portala, ovi elementi se također mogu personalizirati na temelju predefiniраниh uloga na portalu. Npr. nekoj grupi korisnika će se odabirom kontakt obrasca na portalu prikazati neki skup informacija i opcija, dok će se drugoj grupi korisnika prikazati skup informacija i opcija koji odgovaraju samo njima.

4.2.2.2. Rezultati pretraživanja

Kako se portal povećava i postaje složeniji, tako i uspješnost rezultata pretraživanja dolazi do sve većeg izražaja. U ovim slučajevima tipični navigacijski elementi kao traka izbornika postaju sve manje iskoristivi, a pretraživost kao funkcija web portala postaje jedan od najvažnijih prioriteta. Zadovoljstvo korisnika iskoristivošću takvih portala je izravno vezano uz uspješnost pretraživosti.

Kod velikih i složenih web portala će personalizacija rezultata pretraživanja povećati korisnicima vjerojatnost pronalaska željenog rezultata, pružiti im pozitivno iskustvo i potaknuti ih da se zadrže na portalu. Kod personalizacije rezultata pretraživanja se može djelovati na nekoliko područja: prikaz rezultata, rangiranje, tj. redoslijed rezultata i grupiranje rezultata.

Primjer utjecaja personalizacije rezultata pretraživanja može biti neki portal koji sadrži velike kolekcije slika i nudi tražilicu za iste. Ako na tom portalu korisnik u tražilicu upiše riječ *mustang*, rezultati bi mogli vratiti slike životinje i automobila. No, ako portal posjeduje podatke o karakteristikama korisnika, kao npr. neke podatke prema kojima je korisnik izrazio interes za automobile, onda se u personaliziranim rezultatima prikazuju slike automobila Mustang. Još jedan primjer može biti ako je korisnik na nekom portalu s informatičkom opremom koji ima e-trgovinu prilikom registracije unio da više preferira uređaje marke Logitech, prema čemu bi u personaliziranim rezultatima pretraživanja artikala u e-trgovini, npr. tipkovnice, veći rang dobile tipkovnice marke Logitech.

4.2.2.3. Katalozi

Kataloge obično sadrže samo portali koji su usmjereni na prodaju velikog broja proizvoda ili usluga i imaju e-trgovinu. Za uspjeh takvih portala je nužno ponuditi personalizirano iskustvo kupovine. Najveći utjecaj na zadovoljstvo korisnika će obično imati sposobnost portala da korisnike brzo i jednostavno dovedu do proizvoda ili usluge koje traže. Kod toga može pomoći personalizacija kataloga.

Personalizacija kataloga proizvoda i usluga može uvelike doprinijeti tome da se korisnike dovede do onoga što će vjerojatno kupiti. Kako bi bila što učinkovitija, personalizacija kataloga mora odgovarati karakteristikama korisnika da bi se na temelju toga mogao identificirati proizvod ili usluga kao potencijalni predmet kupnje. Personalizirani katalog bi kod kupovine na portalima s e-trgovinom trebao biti oblikovan za dvije kategorije kupaca. Prva se odnosi na kupce koji većinom znaju što žele i ne zahtijevaju smjernice ni pomoć, a druga uključuje kupce koji nisu sigurni što žele i zahtijevaju pomoć pri traženju i kupnji. Svakoj kategoriji kupaca se kod personalizacije kataloga treba posebno pristupiti. Iako su katalozi na portalima većinom vezani uz ponudu proizvoda i usluga preko e-trgovina, mogu se pojaviti i u drugačijim oblicima kao npr. knjižnični katalozi.

4.3. Eksplicitna i implicitna personalizacija web portala

Pojmovi prilagođavanje (eng. *customization*) i personalizacija (eng. *personalization*) su u velikom broju izvora i u više konteksta definirani na različite načine i na različitim razinama međusobne povezanosti. Bowen i Filippini-Fantoni (2004) su definirali prilagođavanje kao sposobnost korisnika da izmijeni sučelje prema pojedinim zahtjevima, a

da se personalizacija javlja kada sustav više potajno podešava sučelje korisniku. Slično tome, Dickerson (2012) navodi da se personalizacija javlja kada se sustav koji se koristi podešava korisniku i ponašanju korisnika, a prilagođavanje kada korisnik mijenja stvari kao što su veličina fonta, boje, raspored, itd. S druge strane, Dholakia i sur. (2000) koriste pojmove personalizacija i prilagođavanje kao sinonime, te navode da odražavaju stupanj po kojem je informacija podešena kako bi zadovoljila potrebe pojedinog posjetitelja.

U pogledu personaliziranih web portala se personalizacija može smatrati višim konceptom koji uključuje prilagođavanje. Eksplicitno personalizirani sustav se odnosi na personalizirani sustav s visokom razinom prilagođavanja, dok se implicitno personalizirani sustav odnosi na personalizirani sustav s niskom razinom prilagođavanja (Beam, 2011). Pri tome se podrazumijeva da prilagođavanje određuje koliko je točno korisnik uključen u proces personalizacije, odnosno stupanj po kojem korisnik eksplicitno sudjeluje u mijenjanju sustava.

Iako se nekad na nekim portalima pokazuje da nema potrebe za eksplicitnom personalizacijom, u nekim slučajevima je za korisnike presudno imati mogućnost prilagodbe nekih elemenata portala, odnosno da im je omogućena eksplicitna personalizacija. Nekad se jednostavno radi o prilagođavanju početne stranice, npr. kod financijskih portala može biti slučaj potrebe za premještanjem elementa s vijestima na istaknutiju lokaciju na portalu, ili skrivanje nekih elemenata kao npr. stanje burze. Eksplicitna personalizacija može uključivati i pružanje informacija od strane korisnika kod registracije. Primjer toga može biti registracija na nekom portalu o glazbi, gdje je korisnik pružio osobne podatke vezane uz glazbeni ukus, od kojih neki mogu biti preferirani žanrovi glazbe, najomiljeniji izvođači i slično, a na temelju tih podataka portal personalizira odgovarajuće elemente. Još jedan primjer eksplicitne personalizacije je Google News portal koji nudi mogućnost prilagođavanja prikazivanja pojedinih tema vijesti, dodavanje vlastitih tema, prilagođavanje izvora vijesti, itd.

Već je spomenuto da se podaci o korisnicima mogu prikupljati dinamički, prema prethodnim aktivnostima, pa se i neki elementi portala mogu personalizirati na temelju tih podataka. To se odnosi na implicitnu personalizaciju, a primjer toga može biti Amazon, koji u pozadini prikuplja podatke o korisnicima bez njihove eksplicitne interakcije. Iz prethodnih aktivnosti korisnika se mogu prikupiti podaci kao što su kupljeni artikli, pregledani artikli, posjećene stranice, količina vremena provedenog na specifičnim stranicama, podaci izvedeni iz pretraživanja, itd. Implicitna personalizacija na portalu mora reagirati na aktivnost korisnika. Primjerice, ako je korisnik na nekom informativnom portalu često pregledavao

neke informacije koje pripadaju istoj kategoriji, portal reagira ponudom skupa informacija koje bi mogle zanimati korisnika, a koje još nije pregledao. Osim podataka izvedenih prema prethodnim aktivnostima, za implicitnu personalizaciju se mogu izvoditi i podaci kao što su IP adresa, jezik preglednika, lokacija, uređaj kojim se pristupa portalu, itd. Podaci se također mogu prikupiti i korištenjem vanjskih izvora podataka kao npr. Facebook računa.

Koja će personalizacija biti korištena u kojoj mjeri ovisi od portala do portala. U međusobnoj usporedbi je, što se tiče točnosti podataka, važno napomenuti da je eksplicitna personalizacija većinom točnija jer je podatke pružio sam korisnik, dok kod izvođenja podataka za implicitnu personalizaciju može doći do pogreške ako su pravila prema kojima se personalizira portal krivo postavljena. Dok eksplicitna personalizacija od korisnika zahtijeva pružanje podataka, za implicitnu personalizaciju korisnici ne moraju potrošiti toliko vremena. Osim eksplicitne i implicitne, neki izvori navode i hibridnu personalizaciju. Raju i Babu (2015) navode da hibridna personalizacija kombinira eksplicitnu i implicitnu personalizaciju kako bi se iz svake postigao najbolji utjecaj.

5. Vrste personaliziranih web portala

Kod razvoja personaliziranog web portala je potrebno razmotriti i odrediti u kojem će obliku portal biti razvijen. Neki od oblika u kojem personalizirani web portal može biti razvijen uključuju distribuirani web portal i hibridnu web aplikaciju (*mashup*).

5.1. Distribuirani web portal

Miletsky (2009) navodi da distribuirani web portal prikuplja informacije od web stranica i drugih portala temeljeno na centralnoj temi, a kao primjer spominje SimplyHired (simplyhired.com) koji daje posjetiteljima mogućnost da preko tražilice na temelju ključnih riječi pretražuju poslove koji su objavljeni od strane poduzeća i organizacija na Webu, te uz to spominje da pretraživanje na SimplyHired stranici vraća poveznice na objave poslova na pojedinim stranicama poduzeća i ostalim stranicama koje su vezane uz pretraživanje poslova.

Distribuirani web portali služe kao središnje mjesto na kojem se prikupljaju informacije dostupne na drugim izvorima, te služe njihovom prikazu na jednom mjestu. Važno je napomenuti da distribuirani web portali iz prikupljenih informacija ne stvaraju nove funkcije. Za web portale koji imaju obilježje distribuiranog web portala se obično podrazumijeva da ne pružaju mogućnosti stvaranja sadržaja. Na način sličan već navedenoj definiciji bi se distribuirani web portal mogao definirati kao poseban oblik web portala koji služi za centralizirani prikaz informacija prikupljenih od drugih web portala. Prema tome se od ostalih web portala razlikuje u tome da je usmjeren samo na prikupljanje i prikazivanje informacija.

Postoji nekoliko mjesta na kojima bi distribuirani web portali mogli pronaći prikladnu primjenu. Razlog zbog kojeg bi neki web portal mogao biti razvijen u obliku distribuiranog je potreba korisnika za posjetom velikog broj portala čiji sadržaj pripada istoj temi, ili je cilj specifičnoj grupi korisnika. Primjer toga može biti ako bi korisnik u svrhu informiranja o aktualnim stanjima na burzi imao potrebu posjetiti 10 različitih financijskih portala. Slično tome, često se događa da korisnici posjećuju barem 5 različitih portala za vijesti kako bi bili informirani o različitim temama. Kod takvih situacija može biti prepoznat potencijal razvoja distribuiranog web portala. Kod razvoja personaliziranog web portala u obliku distribuiranog je potrebno razmotriti mogućnosti koje bi bile ponuđene korisnicima, a neke od njih su odabir prikazivanja pojedinih izvora informacija, količina prikazivanja informacija od pojedinih

izvora, filtriranje prikazivanja informacija po različitim kategorijama, prilagođavanje sučelja, dodavanje vlastitih izvora informacija i sl.

5.2. Hibridna web aplikacija (*mashup*)

Postoji veliki broj poimanja i definicija izraza *mashup*. Neki izvori ga ograničavaju u smislu područja kojeg obuhvaća, dok ga drugi opisuju u općenitijem smislu. Koschmider i sur. (2009) navode da je ideja *mashup*-a stvarati novi sadržaj ponovno koristeći i kombinirajući postojeći sadržaj iz raznovrsnih izvora. Adida (2007) je izraz web *mashup* definirao kao kombinaciju podataka iz različitih izvora temeljenih na Webu i da *mashup* tipično učitava podatke iz unaprijed određenih izvora, stvara smisao iz različitih skupova podataka da bi ih međusobno povezao, te ponudio korisnu kombinaciju. Većina izvora veže *mashup* isključivo uz Web, pa se može smatrati hibridnom web aplikacijom. No, Daniel i Matera (2014) navode da je *mashup* složena aplikacija razvijena počevši od ponovno iskoristivih podataka, aplikacijske logike i/ili korisničkih sučelja tipično, ali ne obavezno, s izvorima na Webu. Međutim, uz to potvrđuju da većina *mashup*-a za njihovu implementaciju koristi web tehnologije.

Hibridna web aplikacija, tj. *mashup* kombinira informacije i servise iz više web izvora kako bi se stvorile nove informacije i servisi prikazani korisnicima na web stranici. Integracija tih izvora je obično jednostavna i brza, te implementirana koristeći aplikacijsko programsko sučelje (eng. *Application programming interface* - *API*) koje izvor, odnosno pružatelj API-a, daje na korištenje i povezuje s podacima preko web servisa. U evoluciji Weba su se *mashup*-ovi pojavili u Web 2.0 fazi, pa koriste i Web 2.0 tehnologije. Obično su usmjereni dinamičnom i interaktivnom sadržaju, te onome izrađenom od strane korisnika, što je i jedna od glavnih značajki Web 2.0 faze i tehnologija.

Broj *mashup*-ova se povećava, a tome pridonosi i sve veći broj javnih API-a. Jedan od najpopularnijih i najkorištenijih API-a kod *mashup*-ova je Google Maps API. Na stranici ProgrammableWeb (programmableweb.com) se može pronaći veliki broj raznih API-a i *mashup*-ova, gdje je uz svaki navedeno koji je API korišten. Primjer *mashup*-a može biti CityStrides (citystrides.com), koji bilježi rute trčanja po gradovima, omogućuje dijeljenje pretrčanih ruta, natjecanje s ostalim korisnicima, dohvaćanje podataka o vremenu, itd. Jedni od API-a koje koristi CityStrides su Google Maps, OpenStreetMap, Facebook, RunKeeper Health Graph i Forecast. WeatherOdds.info je primjer *mashup*-a koji vizualno putem

HighCharts API-a prikazuje korisnicima kakvo će biti vrijeme temeljeno na povijesnim podacima preuzetih pomoću Forecast API-a i lokaciji određenoj pomoću Google Geocoding API-a. Još jedan primjer *mashup*-a je TripFELT.com, koji kombinira Facebook, Google Maps, Youtube, Foursquare i još nekoliko API-a u svrhu pružanja intuitivnog iskustva pregledavanja turističkih odredišta, aktivnosti, hrana i pića, te pripadajućih ocjena koje su dodijelili korisnici.

Česta svrha *mashup*-ova je pomoću postojećih podataka stvoriti neki alat ili aplikaciju s jedinstvenom primjenom i mogućnostima koja podacima daju novu razinu korisnosti, a to pokazuje i širok raspon *mashup*-ova na stranici ProgrammableWeb. No, *mashup*-ovi mogu biti i poseban oblik web portala. Personalizirani web portal također može biti razvijen u obliku *mashup*-a. Kod takvog pristupa je potrebno razmotriti i odrediti koje nove vrijednosti će kombinirani podaci za korisnike pružati, koje API-e koristiti i koje mogućnosti će korisnicima biti pružene. Primjerice, personalizirani filmski web portal razvijen u obliku *mashup*-a bi mogao kombinirati nekoliko API-a vezanih uz filmove da ponudi korisnicima intuitivno pregledavanje i pretraživanje filmova i sličnog sadržaja uz pripadajuće poveznice. Pritom bi mogle bile ponuđene mogućnosti personalizacije kao odabiranje omiljenih žanrova, prilagođavanje sučelja portala, obavještavanje o nadolazećim prikazivanjima filmovima u kinima koja su blizu korisniku, personalizirano pretraživanje filmova, itd.

5.3. Usporedba

Iz definicija distribuiranog web portala i *mashup*-a se može zaključiti da između njih postoji velika sličnost, a odnosi se na prikupljanje informacija iz više izvora, te prikazivanje istih na jednom mjestu. Distribuirani web portali obično ne nude stvaranje sadržaja, već samo prikupljaju sadržaj s drugih portala i prikazuju ga, dok su *mashup*-ovi često usmjereni sadržaju stvorenim od strane korisnika.

Ključna razlika između web portala razvijenog u obliku distribuiranog i onog u obliku *mashup*-a je ta da *mashup* iz dostupnih informacija pruža novu i dodatnu vrijednost korisnicima. Razliku je prepoznao i Nations (2015), navodeći da i distribuirani web portal i web *mashup* povlače informacije s ostalih stranica, ali web *mashup* se razlikuje od distribuiranog web portala u tome što stvara nešto novo iz informacije. U praktičnom dijelu ovog rada je personalizirani web portal razvijen u obliku distribuiranog web portala.

6. Web tehnologije za razvoj personaliziranog web portala

U nastavku su navedene i opisane web tehnologije u kojima je izrađen praktični dio rada, odnosno u kojima je razvijen personalizirani web portal.

6.1. HTML

HTML (eng. *Hypertext Markup Language*) je standardni jezik za opisivanje web stranica. Komponente web stranice su HTML elementi i opisani su HTML oznakama, prema kojima web preglednik prikazuje web stranicu. HTML jezik je razvio Tim Berners-Lee, zajedno s HTTP protokolom i prototipom web preglednika u sklopu *World Wide Web* sustava. Tim Berners-Lee je HTML smatrao primjenom SGML-a (eng. *Standard Generalized Markup Language*), te su prvih 18 elemenata jezika većinom bili temeljeni na SGMLguid-u, SGML dokumentacijskom formatu kojeg je koristio CERN (Bacic i Chmielewski, 2013). Popularnost HTML-a je porasla pojavom prvih web preglednika, a IETF (eng. *The Internet Engineering Task Force*) je formirao HTML radnu grupu koja je 1995. izdala HTML 2.0, prvu službenu HTML specifikaciju (Berners-Lee i Connolly, 1995). W3C (eng. *World Wide Web Consortium*), međunarodna organizacija za standardizaciju web tehnologija koju je osnovao Tim Berners-Lee, u siječnju 1997. godine potvrđuje HTML 3.2 specifikaciju kao standardnu verziju HTML-a (Raggett, 1997). Početkom 1998. prva verzija XML (eng. *Extensible Markup Language*) jezika postaje W3C standardom (Paoli i sur., 1998.). W3C je 1999. izdao 4.01 verziju HTML specifikacije, koja je 2000. godine postala ISO/IEC međunarodni standard (Cs.Tcd.Ie, 2000). Iste godine W3C izdaje prvu verziju XHTML (eng. *Extensible Hypertext Markup Language*) jezika, koji je predstavljao primjenu XML-a za opisivanje HTML-a (W3C, 2002). W3C je 2014. godine izdao HTML5, verziju s velikim brojem novih značajki (W3C, 2014). Neke od njih uključuju nove semantičke, grafičke i multimedijske elemente, te nove API-e za geo-lokaciju, povlačenje i ispuštanje (eng. *drag and drop*), lokalnu pohranu podataka, rad s web aplikacijama van mreže, upravljanje s datotekama, itd.

Svaki HTML dokument, odnosno web stranica se sastoji od HTML elemenata opisanih otvarajućim i, većinom, zatvarajućim oznakama između kojih se nalazi sam sadržaj HTML elementa. Otvarajuće počinju znakom manje od („<“), zatim je naveden naziv HTML elementa (npr. *body*) i završavaju znakom veće od („>“). Zatvarajuće oznake se razlikuju od otvarajućih jedino u tome što se nakon znaka manje piše kosa crta (npr. *</body>*). HTML

elementi koji nemaju sadržaj, odnosno prazni elementi, nemaju ni zatvarajuću oznaku, a neki od takvih su sljedeći: `<br>` - prijelom retka, `<img>` - slika, `<input>` - dio HTML formi, `<meta>` - metapodaci, itd. HTML dokument se sastoji od početnog `<html>` elementa unutar kojeg su sadržani svi ostali, `<head>` elementa koji sadrži metapodatke i sadrži opće podatke o dokumentu, te `<body>` elementa u kojem se nalazi sam sadržaj dokumenta kojeg prikazuje web preglednik. Dodatne informacije o HTML elementima pružaju atributi. HTML atributi se pišu u otvarajućim oznakama, a označavaju se u parovima ime-vrijednost, na sljedeći način: *ime*="vrijednost". Kod sljedećeg `<img>` elementa: `` atribut *src* označava poveznicu na sliku, *width* određuje širinu slike u pikselima, a *height* visinu slike u pikselima. HTML element koji definira poveznicu je `<a>` element, a sama poveznica se navodi unutar *href* atributa.

HTML je temelj svake web stranice, pa služi i kao temelj za web portal. Svi elementi web portala koji prikazuju sadržaj i definiraju strukturu web stranice se odnose na HTML elemente. Mogućnosti prilagođavanja sučelja kod personaliziranih web portala su vezani uz manipulaciju HTML elemenata, npr. mijenjanjem lokacije slike na web stranici se mijenja i sam položaj `<img>` elementa unutar HTML dokumenta. Već spomenuta registracija korisnika se omogućuje HTML formama. Također, svaki resurs koji je uključen u web stranicu, lokalni ili vanjski, mora biti na nekom mjestu naveden u HTML elementu.

6.2. CSS

CSS (eng. *Cascading Style Sheets*) je jezik stilskih uputa za određivanje oblikovanja web stranice. CSS odvaja prezentaciju od sadržaja dokumenta. Håkon Wium Lie je 1994. prvi predložio CSS koncept dok je radio u CERN-u (Lie, 1994). Rad na CSS-u je nastavljen i 1996. godine kada W3C organizacija izdaje prvu standardiziranu verziju CSS-a, koja uvodi podršku za font, boju nekih elemenata uključujući tekst, poravnanje teksta, slika i tablica, margine, prostor između sadržaja i obruba elementa (eng. *padding*), pozicioniranje elemenata i ostalo (Lie i Bos, 1996). Početkom 1997. godine je W3C osnovao CSS Working Group, radnu grupu formiranu kao rezultat razdvajanja grupe HTML Editorial Review Board (Sfetcu, 2014). Stefcu (2014) navodi da CSS Working Group grupa nastavlja s razvojem CSS-a, te krajem 1997. razvija CSS verziju 2 koja je 1998. postala W3C standardom. CSS2 kao nadogradnja na CSS1 uvodi podršku za *z-index*, relativno, apsolutno i fiksno pozicioniranje elemenata, slušne stilske upute, nova svojstva za font, dvosmjerni tekst, mijenjanje kursora,

itd (Lie i sur., 1998). Nakon više godina razvijanja i predlaganja, 2011. je CSS 2.1 verzija postala W3C standardom, a razvijana je zbog ispravaka nekoliko grešaka u CSS2 (Lie i sur., 2011). CSS3 je zadnja verzija jezika CSS, a svoju specifikaciju dijeli na module od kojih se svaki posebno standardizira (Mozilla Developer Network, 2015). CSS3 je doveo veliki broj novih značajki, a neke od njih su zaobljeni rubovi, sjene, animacije, nova svojstava za raspored, gradijenti, transformacije elemenata i web fontovi (Mozilla Developer Network, 2015).

CSS definira kako će web preglednik prikazati HTML elemente, te se može vezati uz HTML dokument na nekoliko načina:

- navođenjem vrijednosti u *style* atribut nekog HTML elementa,
- dodavanjem `<style>` elementa unutar `<head>` oznaka i
- uključivanjem CSS datoteke putem elementa `<link>`.

Navedeni redoslijed načina uključivanja predstavlja i prioritet određivanja stilskih uputa nekog elementa, odnosno najveći prioritet imaju stilske upute navedene u *style* atributu, zatim one navedene unutar `<style>` oznaka, a najmanji imaju one iz CSS datoteke. Uz to je potrebno napomenuti da najveći prioritet imaju zadane postavke web preglednika. CSS stilske upute se navode počevši sa selektorom (eng. *selector*) koji predstavlja HTML element, a zatim se unutar vitičastih zagrada u paru navode CSS svojstva (eng. *property*) i CSS vrijednosti (eng. *value*) odvojeni točkom i zarezom na sljedeći način: *element { svojstvo:vrijednost; svojstvo:vrijednost; }*, npr. *p { color:blue; font-size:16px; font-style:italic; }*. CSS selektorima se odabire kojim elementima se pridružuju stilske upute, a neki selektori uključuju ID, klasu, naziv elementa, naziv atributa, stanje elementa (npr. *hover*) i međusobnu vezu elementa (npr. *div p* selektor primjenjuje stilske upute za *p* element koji je potomak od *div* elementa). Klasa elementa nije jedinstvena i može se primijeniti na više elemenata, te na jednom elementu može biti primijenjeno više klasa, dok je ID elementa jedinstven, odnosno samo jedan element može imati jedan ID i jedan HTML dokument može imati samo jedan element s tim ID-em.

Već je spomenuto zadovoljstvo korisnika grafičkim dizajnom web portala kao čimbenik koji utječe na sveukupno zadovoljstvo korisnika web portalima, pa je sama važnost CSS-a neosporiva. Web portal je potrebno oblikovati kako bi korisnicima izgledao privlačno, što će pridonositi tome da ostanu na portalu, pa je kod razvoja web portala važno posebnu

pažnju posvetiti stilskim uputama. Uz to je s CSS-om potrebno smisleno oblikovati raspored elemenata stranice kako bi se povećala iskoristivost. CSS je također od velike važnosti kod personaliziranog web portala, jer je na nekim mjestima potrebno prikladno mijenjati CSS, ovisno o samim postupcima prilagođavanja sučelja od strane korisnika.

6.3. JavaScript

JavaScript je skriptni jezik koji se koristi kod interaktivnih web stranica za izvođenje dinamičkih zadataka i kod nekih okruženja koja nisu vezana uz web stranice. Kod web stranica se izvodi na strani klijenta, odnosno u web pregledniku, te kod izvođenja dinamičkih zadataka nema potrebe za ponovnim učitavanjem web stranice. JavaScript je 1995. godine razvio Brendan Eich dok je radio u Netscape-u (Champeon, 2001). Netscape je 1996. predao JavaScript ECMA (eng. *European Computer Manufacturers Association*) organizaciji za standardizaciju informacijskih i komunikacijskih tehnologija, u svrhu postizanja standardizirane specifikacije (Valentine i Reid, 2013). Valentine i Reid (2013) navode da je ECMA 1997. izdala službeni standard ECMAScript, te se razvoj nastavio s ECMAScriptom, odnosno JavaScriptom kao osnovom za implementacije. Nakon prvog izdanja ECMAScripta je W3C organizacija krajem 1998. godine izdala prvi DOM (eng. *Document Object Model*) standard (W3C, 1998). DOM model omogućuje dinamičko mijenjanje dijelova HTML dokumenta, što je izuzetno važno za JavaScript u obavljanju dinamičkih zadataka na web stranici. Šesto izdanje ECMAScripta je izdano je u lipnju 2015. godine i predstavlja zadnji ECMAScript standard (Krill, 2015). Kroz godine razvoja je JavaScript na Webu postao sveprisutan i jedan od najpopularnijih programskih jezika korišten za implementaciju velikog broja alata, web aplikacija, okvira, biblioteka, a i popularnih platformi koje nisu vezane uz web stranice kao npr. node.js, platforma na strani poslužitelja koja se koristi za razvoj mrežnih aplikacija. Najpopularnija JavaScript biblioteka je jQuery, usmjeren jednostavnijem korištenju JavaScripta. Koliko je popularan jQuery pokazuje činjenica da od 10 milijuna najposjećenijih web stranica njih 65.2% koristi jQuery (W3Techs, 2015a).

HTML definira sadržaj i strukturu web stranice, CSS definira prezentaciju, a JavaScript se koristi za kontroliranje ponašanja web stranice. JavaScript može funkcionirati kao proceduralni i kao objektno orijentirani jezik, a sintaksa mu je slična jezicima C++ i Java. JavaScript kôd se u HTML dokumentu piše kao sadržaj elementa `<script>` ili se dodaje kao datoteka preko `src` atributa elementa `<script>`. Oznake `<script>` se mogu dodati unutar

`<head>` i `<body>` oznaka. JavaScript kôd se može izvršiti i kao odgovor na HTML događaj, npr. na klik miša. Varijable se u JavaScriptu deklariraju ključnom riječi *var*, a funkcije ključnom riječi *function*. Kod JavaScripta je tip varijable pridružen dinamički prilikom dodjele vrijednosti. Jedan od tipova je objekt, koji ima svoje atribute (svojstva) i metode. Atributi objekata se navode unutar vitičastih zagrada u parovima *ime:vrijednost* odvojeni zarezom, a metode se navode kao vrijednosti nekog atributa u kojeg se spremaju, te se definiraju kao funkcije. Manipuliranje HTML elementima preko objektnog modela omogućuje HTML DOM, a BOM (eng. *Browser Object Model*) preko objekata omogućuje interakciju s web preglednikom, kao npr. dohvaćanje podatka o jeziku web preglednika preko atributa *language* objekta *navigator*. Neke od osnovnih primjena JavaScripta uključuju validaciju HTML formi, dinamičku promjenu sadržaja, predlaganje rezultata kod pretraživanja, animacije i efekte, mijenjanje rasporeda i izgleda dijelova web stranice, interakciju s web preglednikom, izračune, povratne informacije korisnicima, a s pojavom HTML5 verzije i novih API-a je s JavaScriptom moguće izvoditi i složenije zadatke.

JavaScript kontrolira ponašanje web stranice na strani klijenta. Dinamički zadaci na strani klijenta su od velike važnosti kod personaliziranih web portala, jer se korisnicima daju mogućnosti kao što je prilagođavanje elemenata portala i dinamičko mijenjanje sadržaja. Primjerice, mijenjanje lokacije prozora s nekim sadržajem na web stranici omogućuje upravo JavaScript.

6.4. Ajax

Ajax (eng. *Asynchronous Javascript and XML*) je skup tehnologija koje zajedno funkcioniraju u svrhu ažuriranja web stranice bez potrebe za ponovnim učitavanjem. Ajax omogućava asinkronu komunikaciju s web poslužiteljem, odnosno asinkronu razmjenu podataka s poslužiteljem preko objekta *XMLHttpRequest*. Izraz Ajax je prvi definirao Jesse James Garrett, 18. veljače 2005. u članku *Ajax: A New Approach to Web Applications*. Garret (2005) je definirao nekoliko tehnologija koje uključuje Ajax:

- XHTML i CSS - za prezentaciju temeljenu na standardima,
- DOM - za dinamički prikaz i interakciju,
- XML i XSLT - za razmjenu podataka i manipulaciju podacima,
- *XMLHttpRequest* - za asinkrono dohvaćanje podataka i
- JavaScript koji povezuje sve navedeno.

Na populariziranje Ajax-a je veliki utjecaj imao Google, s aplikacijama kao što su Maps i Gmail, a pogotovo sa Suggest-om, odnosno predlaganjem rezultata kod pretraživanja. Iako je XML sadržan u nazivu, Ajax za razmjenu podataka ne mora koristiti XML (Ullman, 2007).

Bez Ajax-a korisnici zahtijevaju stranicu preko HTTP zahtjeva web preglednika, čekaju, te zatim primaju stranicu, dok s Ajax-om korisnici zahtijevaju stranicu koja sadrži Ajax aplikaciju (Ajax *engine*) i nakon što je primljena aplikacija, daljnji HTTP zahtjevi su obavljeni od strane Ajax *engine*-a u pozadini (van Wijngaarden, 2009). Time se omogućuje da korisnici obave sve zadatke na klijentskoj strani bez potrebe za ponovnim osvježavanjem web stranice. Glavni dio Ajax-a je objekt *XMLHttpRequest* koji se koristi za razmjenu podataka s poslužiteljem. Da bi zahtjev bio poslan poslužitelju, u JavaScript kôdu se prvo mora kreirati *XMLHttpRequest* objekt, a zatim se koriste njegove *open()* i *send()* metode. Pomoću parametara *open()* metode se određuje sljedeće: hoće li biti korišten GET ili POST zahtjev, poveznica na zahtijevanu datoteku na poslužitelju i da li je zahtjev asinkroni ili sinkroni. Korist Ajax-a dolazi do izražaja kod izvršavanja programske logike na strani poslužitelja, odnosno kada zahtijevana datoteka (npr. PHP skripta) sadrži programsku logiku. Odgovor od poslužitelja se preuzima preko *responseText* (ako format odgovora nije XML) i *responseXML* (ako je format odgovora XML) atributa objekta *XMLHttpRequest*. Neke JavaScript biblioteke značajno pojednostavljuju korištenje Ajax-a, npr. jQuery omogućava korištenje Ajax-a preko samo jedne metode.

Za pružanje boljeg korisničkog iskustva kod web portala je potrebno dijelove stranice ažurirati bez ponovnog učitavanja, što omogućuje Ajax. Prema tome je i kod personaliziranih web portala Ajax izuzetno važan jer se korisnicima pruža još više opcija i mogućnosti nego kod ostalih web portala. Kad bi se korisnicima kod personaliziranih web portala za svako dohvaćanje podataka ili mijenjanje dijelova stranice ponovno učitavala stranica, takav način zasigurno ne bi pozitivno utjecao na zadovoljstvo korisnika korištenjem web portala.

6.5. PHP

PHP (eng. *PHP: Hypertext Preprocessor*) je široko korišten opće namjenski skriptni jezik otvorenog kôda koji je posebno prikladan za web razvoj i može biti ugrađen u HTML (PHP.net, 2015). PHP kôd na strani poslužitelja obrađuje interpreter, a nakon izvršavanja kôda se generirani HTML na strani klijenta u web pregledniku prikazuje korisnicima kao rezultat. PHP.net (2015) navodi da je PHP 1994. godine razvio Rasmus Lerdorf, a u 6.

mjesecu 1995. ga je javno objavio kao otvoreni kôd pod nazivom *Personal Home Page Tools* (*PHP Tools*). Kasnije te godine su PHP-u dodane neke funkcionalnosti, a naziv se promijenio u FI (eng. *Forms Interpreter*) (PHP.net, 2015). PHP/FI je u 4. mjesecu 1996. predstavljen kao druga implementacija koja je počela dobivati na popularnosti, a u 11. mjesecu 1997. je izdana službena PHP/FI verzija 2.0 (PHP.net, 2015). Popularnost se je povećavala, isto kao i broj programera koji su počeli raditi na PHP-u (Hughes i Zmievski, 2002). PHP.net (2015) navodi da je PHP/FI 2.0 preimenovan u *PHP: Hypertext Preprocessor*, te da je u 6. mjesecu 1998. izdan PHP 3.0. PHP 4.0 je službeno izdan u 5. mjesecu 2000., a bio je temeljen na novom interpreteru pod nazivom *Zend Engine* (PHP.net, 2015). PHP 5 je izdan u 7. mjesecu 2004. s novim *Zend Engine*-om 2.0 (PHP.net, 2015) i poboljšanom podrškom za objektno orijentirano programiranje, MySQL baze podataka i XML (Trachtenberg, 2004). Razvoj PHP-a je nastavljen, te su se izdavala poboljšanja verzije 5, s 5.6.11 verzijom kao zadnjom, dok 6. verzija nije izdana, a PHP 7.0.0 Beta 2 je izdana 24. srpnja 2015 (PHP.net, 2015). Kroz godine razvoja od osnivanja je popularnost PHP-a rasla, te se može smatrati najkorištenijim web programskim jezikom na strani poslužitelja. Od 10 milijuna najposjećenijih web stranica koje koriste programski jezik na strani poslužitelja, 81.2% njih koristi PHP (W3Techs, 2015b).

PHP skripte se izvršavaju na poslužitelju, a mogu biti ugrađene unutar HTML-a. Također se mogu uključiti i kao datoteke. PHP se može izvoditi na gotovo svakoj platformi i na gotovo svakom web poslužitelju. PHP kôd se navodi nakon `<?php` oznake, a prekida se oznakom `?>`. Varijable se deklariraju i koriste sa znakom „\$“ na početku, a tip im se pridružuje dinamički ovisno o dodijeljenoj vrijednosti. Primjer tipa varijable su asocijativna polja, odnosno posebna polja kod kojih je svakom ključu s nazivom dodijeljena vrijednost. PHP podržava superglobalne varijable (eng. *superglobals*), posebne ugrađene varijable koje su dostupne na svim mjestima u kôdu, a neke od njih su:

- `$_GET` - služi za dohvat podataka iz URL parametara,
- `$_POST` - koristi se za prikupljanje podataka iz HTML formi poslanih preko metode POST,
- `$_SESSION` - sadrži varijable sesije i
- `$_COOKIE` - koristi se za dohvat kolačića, odnosno *cookie*-a.

Važan dio PHP-a su funkcije, a deklariraju se ključnom riječi *function*. PHP nudi veliki broj ugrađenih funkcija, a neke od njih omogućuju rad s datotekama, bazama podataka, kriptografijom, kompresijom i arhiviranjem, datumom i vremenom, e-poštom, XML-om, regularnim izrazima, filtriranjem, obradom grešaka, itd.

Da bi personalizirani web portal bio dinamičan i funkcionalan potrebno je koristiti web tehnologije na strani poslužitelja. PHP je kod razvoja personaliziranog web portala izuzetno koristan jer je pomoću ugrađenih funkcija na jednostavan način moguće obaviti veliki raspon zadataka na strani poslužitelja. Za primjer se može uzeti registracija novih korisnika web portala, gdje unos novih korisnika u bazu podataka izvodi PHP skripta.

6.6. MySQL

MySQL je sustav za upravljanje relacijskim bazama podataka (eng. *Relational database management system - RDBMS*) koji je otvorenog koda i izvodi se na gotovo svim platformama. SQL (eng. *Structured Query Language*) je standardizirani jezik za pristup bazama podataka. MySQL je najpopularniji RDBMS otvorenog koda i drugi najpopularniji RDBMS općenito (DB-Engines.com, 2015). Glavni osnivač MySQL-a (i poduzeća MySQL AB) Michael Widenius je 1995. godine izdao prvu verziju MySQL-a (Buytaert, 2010). Razvoj MySQL-a kao proizvoda poduzeća MySQL AB se nastavio, a popularnost rasla. MySQL je 2000. izdan kao otvoreni kod, a 2003. broji 4 milijuna aktivnih instalacija i preko 30 000 preuzimanja dnevno (Buytaert, 2010). Buytaert (2010) još navodi i da je poduzeće MySQL AB 2008. godine kupljeno od strane Sun Microsystems poduzeća. Od 2010. godine s razvojem MySQL-a službeno nastavlja Oracle korporacija, nakon što je kupila Sun Microsystems poduzeće (Oracle.com, 2015). Velika primjena MySQL-a je na Webu, a koriste ga neke od najpopularnijih web stranica kao što su Facebook, YouTube, Yahoo i Twitter (MySQL.com, 2015).

MySQL je često korišten s PHP-om na jednostavan način, a s puno mogućnosti. Upravljanje MySQL bazom podataka je olakšano alatima kao što je phpMyAdmin. PHP nudi široku podršku za rad s MySQL-om preko dva pristupa: MySQLi (eng. *MySQL Improved*) i PDO (eng. *PHP Data Objects*). MySQLi je ograničen na MySQL baze podataka, ali nudi proceduralni i objektno orijentirani pristup, dok PDO nije ograničen samo na MySQL, ali se koristi samo preko objektno orijentiranog pristupa (Marjanovic, 2012).

Personalizirani web portal obično ima veći broj opcija i mogućnosti nego ostali web portali, što može značiti i da je potrebno spremati veći broj podataka i dohvaćati ih na više različitih mjesta. Primjerice, sve postavke korisnika na portalu moraju biti spremljene na nekom mjestu, kako bi ih korisnici opet mogli koristiti, što naglašava važnost MySQL-a kod personaliziranih web portala.

6.7. RSS

RSS (eng. *Really Simple Syndication*), poznat i kao *RDF Site Summary* i *Rich Site Summary*, je format za objavljivanje i pregledavanje sadržaja s web stranica koje ga često osvježavaju. RSS omogućuje pregledavanje sadržaja koji je automatski ažuriran od strane pružatelja prilikom pojave novog sadržaja. S druge strane, pružatelj preko RSS-a na jednostavan način može dijeliti sadržaj koji je često osvježavan. Korisnici se mogu pretplatiti na određeni RSS kanal (eng. *channel* ili *feed*), te tako prikupljati i pregledavati sadržaj iz više izvora na jednom mjestu. Holzner (2006) navodi da je 1997. godine Dave Winer predstavio svoj XML format za sindikaciju u Scripting News weblog-u. Holzner (2006) također navodi da je Dan Libby iz Netscape-a 1999. godine osmislio RSS verziju 0.90 za korištenje na My Netscape portalu i da je kasnije te godine verzija 0.90 promijenjena, te predstavljena verzija 0.91 koja je usvojila dijelove Winerovog formata. U to vrijeme je RSS bio kratica za *Rich Site Summary*. Netscape ubrzo prestaje s razvojem RSS-a, pa se formira RSS-DEV grupa (Zakas i sur., 2011). Holzner (2006) navodi da Dave Winer izdaje promijenjenu verziju RSS-a 0.91, te se odvaja od RSS-DEV grupe. Krajem 2000. godine RSS-DEV grupa, odvojeno od verzije 0.91, izdaje RSS 1.0, novu verziju s velikim odstupanjem od prijašnjih verzija RSS-a koja je bila temeljena na RDF-u (eng. *Resource Description Framework*) i nazvana je *RDF Site Summary* (Zakas i sur., 2011). Ubrzo je Dave Winer izdao RSS 0.92, te nastavio s razvojem i u 2002. godini izdaje RSS 2.0 verziju s nazivom *Really Simple Syndication* (Holzner, 2006). Kao alternativa RSS-u, razvijen je novi format za sindikaciju pod nazivom Atom. Atom 1.0 format je krajem 2005. godine (pod nazivom *Atom Syndication Format*) standardiziran od strane IETF organizacije (Nottingham i Sayre, 2005).

RSS dokument je pisan u XML formatu i sastoji se od posebnih elemenata koji opisuju dokument. Zbog XML formata, svi RSS elementi moraju imati otvarajuću i zatvarajuću oznaku. Svaki RSS dokument počinje deklaracijom XML-a pri čemu se navodi verzija XML-a i način kodiranja znakova (npr. UTF-8). Nakon toga se navodi `<rss>` element u čijem se

atributu obavezno navodi verzija RSS-a. U `<rss>` elementu se navodi samo jedan `<channel>` element koji opisuje RSS kanal i sam sadržaj, te se sastoji od obaveznih i opcionalnih elemenata (Winer, 2003). Winer (2003) navodi da su obavezni elementi sljedeći:

- `<title>` koji definira naziv kanala,
- `<link>` koji definira poveznicu na odgovarajuću web stranicu kanala i
- `<description>` koji opisuje kanal.

Neki od opcionalnih elemenata koje `<channel>` element može sadržavati su:

- `<copyright>` koji sadrži napomenu vezanu uz autorska prava,
- `<lastBuildDate>` koji definira vrijeme zadnje promjene sadržaja kanala i
- `<pubDate>` koji definira datum objavljivanja sadržaja kanala.

RSS `<channel>` element može sadržavati proizvoljan broj `<item>` elemenata koji predstavljaju stavke sadržaja kanala, odnosno različite sadržaje objavljene u kanalu, a najčešće su u obliku članka ili objave (Winer, 2003). Svi elementi koje sadrži `<item>` element su opcionalni, uz uvjet da je naveden ili `<title>` ili `<description>` element (Winer, 2003), no obično svaki `<item>` element sadrži `<title>`, `<link>` i `<description>` elemente, a nekad i neke od opcionalnih elemenata kao npr. `<pubDate>` koji definira datum objavljivanja stavke i `<guid>` koji definira jedinstveni identifikator za `<item>`.

RSS kod web portala može biti od velike koristi. Pretplaćivanjem na željene RSS kanale korisnici na jednom mjestu mogu pregledavati najnoviji sadržaj bez potrebe za posjećivanjem svake web stranice. Zadovoljstvo sadržajem je važan čimbenik koji utječe na zadovoljstvo korisnika web portalima, na što personalizirani web portali mogu pozitivno utjecati s personalizacijom sadržaja preko RSS-a.

7. Razvoj personaliziranog web portala: praktični dio

U nastavku je opisan razvoj personaliziranog web portala. Točnije, razvijen je personalizirani web portal za vijesti koji prikuplja sadržaj iz više portala i prikazuje ih na jednom mjestu, a nazvan je Infoport. Na portalu Infoport je moguće odabrati iz kojih izvora sadržaja, tj. portala će se prikazivati vijesti, s time da su izvori organizirani po kategorijama. Izvore je po želji moguće i obrisati, minimizirati, uvećati preko cijelog ekrana, a moguće im je i mijenjati lokaciju i veličinu, te veličinu teksta. Za korisnike koji ne žele trošiti vrijeme na raspoređivanje izvora, ni mijenjanje veličine istih, ponuđeno je nekoliko unaprijed definiranih opcija za raspored, kao npr. organizacija izvora u dva stupca. Ako se nekim korisnicima ne sviđa zadana tema boja, istu je moguće i promijeniti u postavkama.

7.1. Logotip

Kako bi sam portal imao vizualni identitet, potrebno je osmisлити logotip koji će na konzistentan način biti prikazan na portalu. Logotip portala Infoport je prikazan na slici 10. Pošto sam naziv portala podsjeća na informaciju i portal (ili neku vrstu prolaza ili vrata), u logotip je dodana slika vrata i slovo „i“. Osim spomenute slike i slova, logotip sadrži cijelu riječ naziva samog portala. Za osnovnu boju portala je odabrana ljubičasta boja, pa je shodno tome i slovo „i“ u logotipu te boje. Kako riječ „Infoport“ u logotipu ne bi ostavila monoton dojam, napisana je fontom koji je oblikovan da izgleda kao da je rezolucija slova mala.



Slika 10: Logotip portala Infoport

Iz logotipa je izdvojeno slovo „i“ kao slika koja će se prikazati u kartici web preglednika, koja se obično naziva *favicon* i uključuje se u stranicu navođenjem sljedećeg elementa unutar `<head>` oznaka u HTML dokumentu: `<link rel="icon" href="slike/favicon.ico">`. Kod navedenog elementa se podrazumijeva da postoji slika *favicon.ico* unutar direktorija *slike*.

7.2. Zaglavlje

Kako bi opcije dodavanja izvora i postavke korisniku bile na vidljivom mjestu, potrebno ih je dodati u zaglavlje stranice, prikazano na slici 11.



Slika 11: Zaglavlje s otvorenim opcijama

U sljedećem isječku se nalazi HTML struktura prikazanog zaglavlja.

```
<header>
  <div id="dodaj_izvor_spremnik">
    <div id="dodaj_izvor">
      
    </div>
  </div>
  <div id="logo_spremnik">
    <a href="index.php"></a>
  </div>
  <div id="profil_spremnik">
    <div id="profil">
      
      <span id="profil_naziv">Luka</span>
      
    </div>
  </div>
  <div id="profil_opcije_spremnik">
    <ul>
      <li><a id="opcije_postavke" href="">Postavke</a></li>
      <li><a href="odjavi_korisnika.php">Odjavi se</a></li>
    </ul>
  </div>
</header>
```

Trenutno je unutar `<span>` elementa s *id*-em `profil_naziv` ime korisnika napisano izravno, no kasnije će se na to mjesto dodati ispis prijavljenog korisnika iz sesije. Element `<div id="profil_opcije_spremnik">` sadrži opcije koje se pojavljuju dinamički, klikom na tipku koja se sastoji od slike profila, imena korisnika i slike trokuta. Kako bi zaglavlje izgledalo kao na slici 11, navedene HTML elemente je potrebno oblikovati pomoću CSS-a na način prikazan u sljedećem isječku.

```

header {
    background-color: #303030;
}

header div#dodaj_izvor_spremnik {
    width: 30%;
    float: left;
}

header div#dodaj_izvor_spremnik div#dodaj_izvor {
    transition: all 0.2s ease;
}

header div#dodaj_izvor_spremnik div#dodaj_izvor:hover {
    cursor: pointer;
    opacity: 0.6;
}

header div#logo_spremnik {
    width: 40%;
    text-align: center;
    float: left;
}

header div#profil_spremnik {
    width: 30%;
    float: right;
    color: #c8c8c8;
}

header div#profil_opcije_spremnik {
    display: none;
    background-color: #fff;
    position: absolute;
    right: 10px;
    top: 50px;
    border: 1px #aaa solid;
    border-bottom-left-radius: 4px;
    border-bottom-right-radius: 4px;
    box-shadow: 0px 2px 15px 0px rgba(0, 0, 0, 0.25);
}

```

Kako bi `<div>` elementi koje sadrži `<header>` element bili raspoređeni jedan do drugog, a ne jedan ispod drugog, svakome je potrebno pridružiti odgovarajući CSS vrijednost za svojstvo *float* i odrediti širinu što je u ovom slučaju izraženo u postotku preko CSS svojstva *width*. Za prelazak mišem preko tipke za otvaranje postavki je dodan efekt prijelaza iz jedno stanje u drugo svojstvom *transition*. Drugo stanje se opisuje CSS svojstvima unutar *:hover* selektora, a u ovom slučaju ta svojstva uključuju promjenu prozirnosti i kursora miša. Da bi logotip na stranici uvijek bio u središtu, `<div>` elementu koji ga sadrži je potrebno pridružiti svojstvo *text-align: center*.

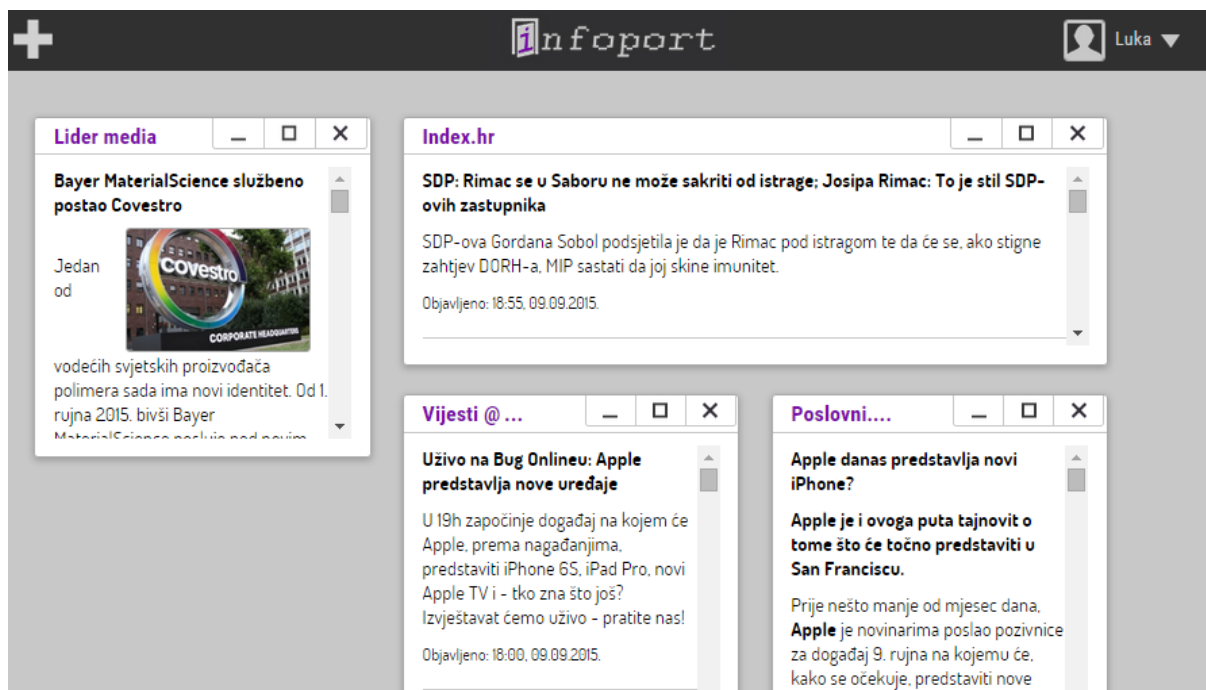
Opcije će isprva biti sakrivene, pa je `<div>` elementu koji sadrži opcije dodano svojstvo *display: none*. Za privlačniji izgled opcija je dodano svojstvo *border-radius* kojem se postiže zarubljenost elementa, te *box-shadow* za postizanje sjene. Klikom na tipku za otkrivanje opcija je iste potrebno prikazati, što se može postići JavaScriptom. U ovom radu je JavaScript korišten pomoću biblioteke jQuery, a u svakoj datoteci je kôd napisan unutar `$(document).ready()` bloka, što znači da će se taj kôd biti izvršen tek kada je učitani cijeli HTML dokument. Kako bi se koristio jQuery, prvo je potrebno dodati poveznicu na jQuery izvornu datoteku unutar `<head>` oznaka HTML dokumenta, a poveznica se navodi preko *src* atributa oznake `<script>`. U sljedećem isječku je prikazano otkrivanje `<div>` elementa koji sadrži opcije.

```
$( "#profil" ).click(function (event) {  
    event.stopPropagation();  
    $( "#profil_opcije_spremnik" ).fadeToggle(200);  
});  
  
$(document).click(function () {  
    $( "#profil_opcije_spremnik" ).fadeOut(200);  
});
```

Znakom „\$“ se pristupa jQuery-u, a zatim se u zagradama na isti način kao i kod CSS-a navode selektori HTML elemenata uz koje se veže događaj i funkcija koja se izvršava nakon pojave tog događaja. U ovom slučaju, klikom miša na `<div>` element s *id-em profil* se otkriva, odnosno sakriva `<div>` element s *id-em profil_opcije_spremnik*. Otkrivanje i skrivanje se odvija pomoću metode *fadeToggle()* i to s efektom promjene stanja u trajanju od 200 milisekundi kako bi sama promjena bila ugodnija oku. Također je dodano skrivanje `<div>` elementa s opcijama nakon svakog klika mišem bilo gdje na HTML dokument, tj. web stranicu. Pošto taj klik miša bilo gdje na dokument uključuje i tipku za otkrivanje opcija, u funkciji koja se veže uz taj događaj je potrebno pozvati *stopPropagation()* metodu proslijeđenog parametra *event* kako bi se spriječilo obavješćavanje roditelja u DOM stablu o pojavi događaja, odnosno kako bi se u ovom slučaju spriječilo obavješćavanje *document* selektora o događaju *click*, a samim time i skrivanje opcija.

7.3. Glavni dio

Glavni dio portala Infoport, koji se može vidjeti na slici 12, se odnosi i na glavni prostor portala u kojem su smješteni izvori sadržaja, odnosno ostali portali koje korisnici mogu prilagođavati osobnim potrebama.



Slika 12: Glavni dio

Osnovna HTML struktura glavnog dijela portala je prikazana u sljedećem isječku.

```
<div id="glavno">
  <div class="gridster">
    <ul>
      <li id="47" data-col="1" data-row="1" data-size="3" data-
size="4">
      <li id="50" data-col="4" data-row="1" data-size="6" data-
size="3">
      <li id="48" data-col="4" data-row="4" data-size="3" data-
size="5">
      <li id="51" data-col="7" data-row="4" data-size="3" data-
size="5">
    </ul>
  </div>
</div>
```

Element `<ul>` unutar `<div>` elementa s klasom *gridster* sadrži sve izvore koji se na stranicu učitavaju dinamički. Svaki izvor je dodan kao `<li>` element s *id*-em i posebnim atributima s pripadajućim vrijednostima koje koristi jQuery dodatak Gridster. Tako učitani `<li>` element sadrži svoju HTML strukturu i stil koji su opisani kasnije. Pomoću dodatka Gridster se lokacija i veličina elemenata na stranici može mijenjati na dinamičan i interaktivan način. Gridster je dostupan na stranici gridster.net koja sadrži osnovne primjere i upute korištenja, a i dokumentaciju za attribute i metode objekta *gridster*.

Kako bi se koristio dodatak Gridster, prvo je potrebno preuzeti Gridster JavaScript i CSS datoteku, te uključiti ih u HTML dokument unutar `<head>` oznaka. Kod korištenja dodatka Gridster je izuzetno korisno spremanje pozicija i veličine elemenata u JSON formatu koji se može jednostavno spremiti u bazu podataka i po potrebi dohvatiti. HTML attribute prikazane u isječku iznad Gridster koristi kako bi organizirao raspored i veličinu `<li>` elemenata, odnosno izvora. Ti atributi su: *data-col* za određivanje stupca, *data-row* za određivanje retka, *data-size-x* za širinu i *data-size-y* za visinu elementa na stranici.

Spremljene pozicije i veličine koje Gridster koristi za raspoređivanje elemenata moraju biti učitane. U ovom slučaju su pozicije spremljene u bazu podataka u JSON formatu, ali su spremljene kao obični tekst, pa je isti potrebno parsirati kako bi ga mogao koristiti *gridster* objekt. Sljedeći isječak prikazuje dohvaćanje pozicija na strani klijenta.

```
var serialization = "";
$.ajax({
  async: false,
  url: "baza_dohvati_pozicije.php",
  success: function (pozicije) {
    serialization = JSON.parse(pozicije);
  }
});
serialization = Gridster.sort_by_row_and_col_asc(serialization);
```

Koristi se jQuery *ajax()* metoda kako bi se PHP skripta *baza_dohvati_pozicije.php* koja dohvaća pozicije iz baze podataka izvršila bez potrebe za ponovnim pokretanjem stranice. Atribut *async* je postavljen na *false* kako bi se spriječilo daljnje izvršavanje JavaScript kôda u kojem je na nekim mjestima neophodna vrijednost učitanih pozicija, pa bi došlo do greške. U *success* atributu je navedena funkcija koja će se izvršiti nakon odgovora na Ajax zahtjev, odnosno nakon izvršavanja PHP skripte na strani poslužitelja. Odgovor od izvršene PHP skripte se dohvaća preko parametra *pozicije*. U ovom slučaju se odgovor, tj. dohvaćene pozicije u JSON formatu parsiraju i spremaju u varijablu za daljnje korištenje. Nakon dohvaćanja pozicija se iste sortiraju pomoću posebne Gridster metode *sort_by_row_and_col_asc()* koja za parametar prima polje pozicija i sortira ga prema retcima i stupcima.

U bazi podataka su tablice izrađene pomoću alata phpMyAdmin. Za potrebe portala Infoport su izrađene dvije tablice: *korisnici* i *izvori*. Tablica *korisnici* sadržava sve podatke vezane uz korisnike, a to uključuje i pozicije izvora na portalu, a svaki zapis u tablici *izvori*

prikazuje koji točno izvor pripada kojem korisniku. Primjer jednog zapisa pozicija u tablicu *korisnici* je dan u sljedećem isječku.

```
[{"id": "50", "col": 4, "row": 1, "size_x": 6, "size_y": 3},
{"id": "48", "col": 4, "row": 4, "size_x": 3, "size_y": 5},
{"id": "51", "col": 7, "row": 4, "size_x": 3, "size_y": 5},
{"id": "47", "col": 1, "row": 1, "size_x": 3, "size_y": 4}]
```

Takav zapis kao obični tekst se kasnije dohvaća i parsira kao JSON format na već prikazan način. Atribut *id* odgovara i atributu *id* u tablici *izvori*, a ostale attribute koristi objekt *gridster* kako bi rasporedio elemente i dodijelio im odgovarajuće veličine.

U sljedećem isječku se vidi sadržaj PHP skripte *baza_dohvati_pozicije.php*.

```
<?php

$veza = mysqli_connect("localhost", "", "", "");

if (!$veza) {
    die('Greška kod spajanja: ' . mysqli_error($veza));
}

mysqli_select_db($veza, "");

$sql = "SELECT pozicije FROM korisnici WHERE id = 1";

$resultat = mysqli_query($veza, $sql);

while ($zapis = mysqli_fetch_array($resultat)) {
    echo $zapis["pozicije"];
}

mysqli_close($veza);
```

U ovom radu je za svaki pristup MySQL bazi podataka korišten PHP-ov ugrađeni MySQLi proceduralni pristup prikazan u isječku iznad. Prvo je potrebno otvoriti vezu prema bazi podataka što se obavlja pozivom funkcije *mysqli_connect()* čiji je prvi parametar poslužitelj, drugi korisnik baze podataka, treći lozinka za korisnika, a četvrti parametar naziv baze podataka. U isječku iznad su zadnja tri parametra ostavljena prazna. Ako je veza prema bazi neuspješna, PHP skripta prekida s izvođenjem i ispisuje se greška. Zatim je potrebno postaviti bazu podataka kao zadanu za sve daljnje upite prema bazi podataka preko funkcije *mysqli_select_db()* koja prima dva parametra: vezu i naziv baze podataka.

Slijedi izvršavanje upita prema bazi podataka koji u ovom slučaju dohvaća pozicije iz tablice *korisnici* za korisnika s *id*-em 1. Trenutno se *id* korisnika iz baze podataka dohvaća na temelju upisane vrijednosti, no kasnije će nakon omogućene registracije i prijave tu vrijednost

poprimati *id* prijavljenog korisnika iz sesije. Sam upit se izvršava pomoću funkcije *mysqli_query()* koja za prvi parametar prima vezu prema bazi podataka, a za drugi tekst upita. Svaki zapis rezultata upita je potrebno dohvatiti u *while* petlji u kojoj se svaki zapis sprema pomoću funkcije *mysqli_fetch_array()* koja za parametar prima rezultat. Pojedinom atributu tablice se pristupa tako da se navodi naziv indeksa polja koji odgovara nazivu atributa iz tablice. U ovom slučaju se taj zapis ispisuje pomoću naredbe *echo* kako bi zapis bio dohvaćen na strani klijenta u obliku Ajax odgovora. Na kraju se veza prema bazi podataka zatvara pozivom funkcije *mysqli_close()* kojoj se kao parametar proslijeđuje veza.

U sljedećem isječku je prikazan JavaScript kôd s inicijalizacijom *gridster* objekta.

```
var gridster = $(".gridster ul").gridster({
  widget_base_dimensions: [100, 50],
  widget_margins: [13, 13],
  serialize_params: function ($w, wgd) {
    return {
      id: wgd.el[0].id,
      col: wgd.col,
      row: wgd.row,
      size_x: wgd.size_x,
      size_y: wgd.size_y
    };
  },
  resize: {
    enabled: true,
    min_size: [3, 2],
    stop: function () {
      var $poz = JSON.stringify(gridster.serialize());
      $.ajax({
        type: "GET",
        dataType: "json",
        url: "baza_azuriraj_pozicije.php",
        data: {data: $poz}
      });
    }
  },
  draggable: {
    handle: ".izvor_naziv",
    stop: function () {
      var $poz = JSON.stringify(gridster.serialize());
      $.ajax({
        type: "GET",
        dataType: "json",
        url: "baza_azuriraj_pozicije.php",
        data: {data: $poz}
      });
    }
  }
}).data("gridster");
```

Prvo je potrebno navesti selektor kojim se označava gdje će točno *gridster* dodati elemente. U ovom slučaju to je `<ul>` element koji je potomak `<div>` elementa s klasom *gridster*. Atributom *widget_base_dimensions* se postavlja osnovna veličina elemenata u pikselima po kojoj će biti skalirana ishodna veličina nakon učitavanja, s time da prvi indeks označava širinu, a drugi visinu. Atributom *widget_margins* se postavljaju horizontalne i vertikalne margine između svakog učitano elementa. Unutar *serialize_params* atributa se može definirati funkcija koja vraća podatke za svaki učitani element nakon poziva metode *serialize()*. Spomenuta metoda je izuzetno korisna jer postojeće pozicije i veličine elemenata sprema u JavaScript polje objekata koje je moguće pretvoriti u običan tekst i spremiti u bazu podataka. A to se i u ovom slučaju obavlja i to definiranjem funkcije unutar atributa *resize* i *draggable*.

Unutar atributa *resize* se određuju postavke vezane uz interaktivno mijenjane veličine elementa, a u ovom slučaju je definirana minimalna širina i veličina koju može poprimiti element, te funkcija koja će se izvršiti nakon što korisnik prestane s mijenjanjem veličine. Navedena funkcija sprema trenutnu poziciju i veličinu elemenata u JSON formatu (točnije, u JavaScript polje objekata), koji se pretvara u tekst i prenosi preko GET parametra *data* putem Ajax zahtjeva prema PHP skripti *baza_azuriraj_pozicije.php* koja te podatke zatim sprema u bazu podataka. Isto se obavlja i unutar atributa *draggable* kojim se određuju postavke vezane uz mijenjanje lokacije elemenata. U *draggable* atributu je preko atributa *handle* navedeno nad kojim točno elementom će se obavljati mijenjanje lokacije, odnosno gdje će točno korisnici imati mogućnost klika mišem kako bi zatim mijenjali lokaciju elementa. Kako bi se koristile metode objekta *gridster*, još je na kraj potrebno pozvati metodu *data()* s proslijeđenim nazivom objekta *gridster*.

U sljedećem isječku se nalazi dio sadržaja PHP skripte koja ažurira podatke o poziciji i veličini elemenata.

```
$pozicije = $_GET["data"];  
$sql = "UPDATE korisnici SET pozicije = '" . $pozicije . "' WHERE id = 1";
```

Podaci o poziciji i veličini elemenata preneseni preko GET parametra *data* se čitaju iz superglobalne varijable *\$_GET*, pa se zatim koriste kod SQL upita prema bazi podataka, odnosno kod ažuriranja atributa *pozicije* u tablici *korisnici* s vrijednostima koje su proslijeđene i to za korisnika s *id*-em 1. Isto kao i kod prethodno prikazane PHP skripte,

trenutno je *id* korisnika upisan izravno, no kasnije će isti poprimiti vrijednost iz sesije prijavljenog korisnika.

Učitane i sortirane pozicije iz polja *serialization* je potrebno pridružiti HTML strukturi i sadržaju svakog izvora. Kako bi se to postiglo, prvo je potrebno u petlji proći kroz svaki element polja u koje su spremljene pozicije, a to je polje *serialization*. U sljedećem isječku je prikazan JavaScript kôd za prolazak kroz polje *serialization* u petlji.

```
var zahtjevi = [];  
var polje = [];  
$.each(serialization, function () {  
    var $id = parseInt(this.id);  
    var $size_x = parseInt(this.size_x);  
    var $size_y = parseInt(this.size_y);  
    var $col = parseInt(this.col);  
    var $row = parseInt(this.row);  
    var zahtjev = $.get("baza_dohvati_izvore.php?r=" + $id, function  
(odgovor) {  
        var izvor = {html: odgovor, id: $id, size_x: $size_x, size_y:  
$size_y, col: $col, row: $row};  
        polje.push(izvor);  
    });  
    zahtjevi.push(zahtjev);  
});
```

Svaki element, odnosno svaki izvor koji će biti učitani kao ** element, je spremljen u polje *serialization* kao objekt s atributima koji sadrže podatke o *id*-u, veličini i poziciji. Polje *serialization* je potrebno po redu pročitati u petlji što je obavljeno pomoću jQuery metode *each()*. U spomenutoj metodi je polje proslijeđeno kao parametar, a nakon toga se navodi funkcija koja će se izvršiti nakon svakog prolaska u petlji. U funkciji se sprema svaki atribut polja *serialization* koji je parsiran kao broj, te se u polje *zahtjevi* sprema Ajax zahtjev prema PHP skripti koja u odgovoru vraća HTML strukturu i sadržaj izvora prema proslijeđenom *id*-u preko GET parametra. U ovom slučaju se Ajax zahtjev šalje pomoću jQuery metode *get()*.

Svaki Ajax zahtjev se sprema u polje zbog toga što se izvodi asinkrono, odnosno ne nužno po redu, a u ovom slučaju se želi postići da svi izvori budu učitani po redu, onako kako su ih korisnici postavili. Iz istog razloga se svaki odgovor od Ajax zahtjeva sprema u polje, umjesto da se izravno učitava na stranicu, a dodavanje izvora se mora izvršiti po redu i to kada su dohvaćeni svi izvori, kao što je prikazano u sljedećem isječku.

```

$("#ajax_ucitavanje").show();
$.when.apply($, zahtjevi).done(function () {
    $.each(serialization, function () {
        var html;
        for (var i = 0; i < polje.length; i++) {
            if (polje[i].id === parseInt(this.id)) {
                html = polje[i].html;
            }
        }
        gridster.add_widget(html, parseInt(this.size_x),
        parseInt(this.size_y), parseInt(this.col), parseInt(this.row));
    });
    $("#ajax_ucitavanje").hide();
});

```

Na početku se prikazuje `<div>` element s *id-em* `ajax_ucitavanje` koji sadrži GIF animaciju učitavanja kako bi korisnici dobili dojam da se u pozadini nešto izvršava. Čekanje na Ajax zahtjeve koji se spremaju u polje *zahtjevi* se postiže pomoću metode `when()` i to na način da se ista pozove s poljem *zahtjevi* kao parametrom preko `apply()` metode kojoj se polje *zahtjevi* proslijeđuje kao parametar. Uz `when()` metodu se kôd koji se izvršava navodi unutar `done()` metode, a izvršava se još jedna petlja pomoću `each()` metode u kojoj se svaki objekt polja *serialization* uspoređuje sa svakim elementom polja *polje*, koje sadrži učitani odgovor od PHP skripte koja dohvaća HTML strukturu i sadržaj izvora na temelju proslijeđenog *id-a* izvora. Ako je *id* iz polja *polje* jednako *id-u* polja *serialization*, u tom slučaju je element potrebno dodati na stranicu. To se izvršava preko objekta *gridster* i metode `add_widget()` kojoj se proslijeđuje sadržaj koji će biti prikazan, širina, visina, stupac i redak elementa. Nakon učitavanja svih izvora se `<div>` element s GIF animacijom sakriva.

PHP skripta `baza_dohvati_izvore.php` na temelju GET parametra *id* dohvaća odgovarajući izvor i sadržaj koji se na dosad prikazan način učitavaju na stranicu. Sadržaj koji se učitava je dohvaćen preko RSS poveznice i to pomoću PHP biblioteke SimplePie, dostupne na stranici simplepie.org. RSS poveznicu koja je potrebna za učitavanje RSS sadržaja preko SimplePie biblioteke je potrebno dohvatiti iz baze, što je prikazano u sljedećem isječku.

```

$id = $_GET["r"];
$sql = "SELECT poveznica FROM izvori WHERE id = " . $id;

```

Poveznica koja je potrebna za učitavanje RSS sadržaja se dohvaća iz tablice *izvori* na temelju proslijeđenog *id-a* izvora, a samo učitavanje RSS sadržaja pomoću biblioteke SimplePie je prikazano u sljedećem isječku.

```

date_default_timezone_set("Europe/Zagreb");
setlocale(LC_TIME, "hr");

$feed = new SimplePie();
$feed->set_feed_url($zapis["poveznica"]);
$feed->init();
$feed->handle_content_type();

```

Kako bi odgovaralo vrijeme objavljivanja unutar RSS sadržaja, prvo je potrebno postaviti odgovarajuću vremensku zonu i lokalizaciju. Zatim se instancira objekt klase *SimplePie* preko čije se metode *set_feed_url()* postavlja poveznica i to preko proslijeđenog parametra koji je u ovom slučaju zapis iz baze podataka koji sadrži poveznicu. S metodom *init()* se pokreće *SimplePie*, a metoda *handle_content_type()* je potrebna kako bi se koristio ispravan način kodiranja znakova. HTML struktura izvora je dana u sljedećem isječku.

```

<li id="<?php echo $id ?>">
  <div class="izvor">
    <div class="izvor_naziv">
      <a class="izvor_naziv_link" href="<?php echo $feed-
>get_permalink(); ?>">
        <h1 title="<?php echo $feed->get_description(); ?>"><?php
echo $feed->get_title(); ?></h1>
      </a>
      <div class="izvor_opcije">
        <div class="izvor_opcije_stavka minimizirati">
          
        <div class="izvor_opcije_stavka mijenjati">
          
        <div class="izvor_opcije_stavka obrisati">
          
      </div>
    </div>
    <div class="izvor_sadrzaj">
      <?php foreach ($feed->get_items() as $item): ?>
        <div class="izvor_sadrzaj_stavka">
          <div class="izvor_sadrzaj_glavno_spremnik">
            <a href="<?php echo $item->get_permalink(); ?>">
              <h2><?php echo $item->get_title(); ?></h2>
            </a>
            <p><?php echo $item->get_description(); ?></p>
            <p><small>Objavljeno: <?php echo $item-
>get_local_date('%H:%M, %d.%m.%Y.');" ?></small></p>
          </div>
        </div>
      <?php endforeach; ?>
    </div>
  </div>
</li>

```

Element `<li>`, koji je glavni element izvora, poprima onu vrijednost atributa `id` koja je prosljeđena PHP skripti `baza_dohvati_izvore.php`. Poveznica na portal koji pruža RSS sadržaj je ispisana pomoću SimplePie metode `get_permalink()`, a obuhvaća i naslov RSS kanala koji se dohvaća metodom `get_title()`. U atribut `title` elementa `<h1>` je pomoću metode `get_description()` dodan opis RSS kanala. Svakom izvoru su dodane tipke za minimiziranje, maksimiziranje i brisanje. Svaka stavka RSS sadržaja će biti ispisana u *foreach* petlji u kojoj se prolazi kroz sve stavke dohvaćene preko metode `get_items()`. Stavka je zapisana u varijablu `$item` preko koje se može pristupiti metodama za dohvaćanje naslova, opisa i vremena objave stavke. Vrijeme se ispisuje metodom `get_local_date()` kojoj se kao parametar prosljeđuje željeni format ispisa vremena.

Za izvore su definirane CSS stilske upute od kojih se neke mogu vidjeti u sljedećem isječku.

```
div#glavno .izvor {
    background-color: #fff;
    border: 1px #aaa solid;
    border-radius: 4px;
    box-shadow: 0px 2px 7px 0px rgba(0, 0, 0, 0.25);
}

div#glavno .izvor .izvor_naziv {
    cursor: move;
}

div#glavno .izvor .izvor_naziv h1 {
    overflow: hidden;
    text-overflow: ellipsis;
}

div#glavno .izvor .izvor_naziv .izvor_opcije {
    margin-left: auto;
}

div#glavno .izvor .izvor_sadrzaj {
    overflow-y: auto;
    height: calc(100% - 55px);
}

div#glavno .izvor .izvor_sadrzaj .izvor_sadrzaj_stavka h2, div#glavno
.izvor .izvor_sadrzaj .izvor_sadrzaj_stavka p {
    font-family: 'Dosis', sans-serif;
}

div#glavno .izvor .izvor_sadrzaj .izvor_sadrzaj_stavka img {
    max-width: 150px;
    max-height: 100px;
}
```

Da bi svaki izvor bio vizualno istaknut, dodana je bijela pozadina, sivi obrub, te zaobljenost i sjena. Već spomenuti atribut *handle* kod inicijalizacije objekta *gridster* određuje nad kojim točno elementom će se obavljati mijenjanje lokacije, pa je tom elementu i promijenjen odgovarajući kursor miša koji će korisnicima dati do znanja da pomoću tog elementa izvorima može mijenjati lokacija. Naslovu RSS kanala koji je sadržan unutar `<h1>` elementa su pridružena svojstva *overflow: hidden* i *text-overflow: ellipsis* kako bi se spriječilo širenje teksta u slučaju kada je naslov duži od samog izvora, te kako bi se onda na kraj naslova koji je skraćen dodale tri točke koje bi korisnicima davale do znanja da nije prikazan cijeli naslov. Opcije minimiziranja, maksimiziranja i brisanja izvora bi uvijek trebale biti na desnoj strani pa im se pridružuje vrijednost *auto* lijeve margine, odnosno svojstva *margin-left*.

Kako bi se postiglo da se korisnici mogu kretati vertikalno sadržajem unutar izvora, umjesto da se isti proširi kroz cijelu stranicu, potrebno je svojstvu *overflow-y* pridružiti vrijednost *auto*. Iz istog razloga se visina sadržaja izvora računa CSS funkcijom *calc()* u koju se navodi da će izvor poprimiti cijelu vrijednost elementa roditelja umanjenu za 55 piksela kako visina ne bi bila veća od visine elementa roditelja. Naslov i odlomak svake stavke učitano RSS sadržaja ima font *Dosis* koji je preuzet kao Google Font sa stranice [google.com/fonts](https://www.google.com/fonts). Iako nije navedeno u isječku iznad, osim prethodno navedenih, svi ostali elementi na portalu Infoport imaju font *Roboto Condensed*, također preuzet kao Google Font. U slučaju da uz RSS sadržaj bude učitana i slika, ista se ograničava na 150 piksela širine i 100 piksela visine kako ne bi zauzela cijeli prostor izvora.

Opcijama za minimiziranje, maksimiziranje i brisanje je potrebno u JavaScriptu, odnosno jQueryu dodati funkcionalnosti. Kako bi korisnicima bilo lakše čitanje sadržaja na portalu Infoport, opcija minimiziranja je dodana kako bi u svakom trenutku korisnici mogli sakriti sadržaj koji ih ne zanima, ili su odlučili da će ga kasnije pročitati. Opcija maksimiziranja je dodana kako bi se sadržaj po izboru mogao proširiti preko cijelog ekrana što omogućuje pregledan način čitanja. Također se u svakom trenutku korisnicima pruža mogućnost brisanja izvora, ukoliko odluče da im nije potreban. U sljedećem isječku je prikazan JavaScript kôd s osnovnim funkcionalnostima minimiziranja izvora.

```

var $stara_velicina;
$("body").on("click", ".izvor .izvor_naziv .izvor_opcije .minimizirati",
function () {

    if ($(this).children(":first").attr("src")
        === "slike/tipka_izvor_minimizirati.png") {
        $stara_velicina = $(this).parents().eq(3).attr("data-sizey");
        var $id = $(this).parents().eq(3).attr("id");
        $.get("baza_azuriraj_staru_velicinu.php?r="
            + $id + "&sv=" + $stara_velicina);
        gridster.resize_widget($(this).parents().eq(3), "", parseInt("1"));
        $(this).children(":first")
            .attr("src", "slike/tipka_izvor_prosiriti.png");
    }

    else {
        var $html = $(this).parents().eq(3);
        var $id = $(this).parents().eq(3).attr("id");
        $.get("baza_dohvati_staru_velicinu.php?r=" + $id, function (odgovor) {
            $stara_velicina = odgovor;
            gridster.resize_widget($html, "", parseInt($stara_velicina));
        });
        gridster.resize_widget($html, "", parseInt($stara_velicina));
        $(this).children(":first")
            .attr("src", "slike/tipka_izvor_minimizirati.png");
    }

    spremi_pozicije();
    azuriraj_min_sliku();

});

```

U prvi selektor se upisuje element roditelj, u pozivu *on()* metode se kao prvi parametar navodi događaj, a kao drugi se navodi selektor nad kojim se događaj pojavljuje. Zatim se navodi funkcija koja će se izvršiti nakon pojave događaja nad elementom. Prvo je potrebno preko DOM stabla provjeriti atribut *src* prvog djeteta od elementa s opcijom minimiziranja, a to je slika. Ako je ta slika *tipka_izvor_minimizirati.png*, onda sadržaj izvora treba minimizirati. Kako bi ostala zapamćena, stara veličina izvora se dohvaća i zapisuje pomoću atributa *data-sizey* kojeg sadrži element ** koji sadrži cijelu HTML strukturu i sadržaj izvora. Na isti se način dohvaća *id*, koji se uz staru veličinu prenosi PHP skripti kao GET parametar. PHP skripta *baza_azuriraj_staru_velicinu.php* u bazi podataka ažurira staru veličinu minimiziranog izvora na temelju prenesenog *id*-a. Sadržaj izvora se zatim minimizira preko objekta *gridster* i metode *resize_widget()* koja omogućuje promjenu veličine elementa i koja kao prvi parametar prima element koji mijenja veličinu, kao drugi parametar prima širinu, a kao treći željenu visinu elementa koja će u ovom slučaju biti minimalna, odnosno 1.

Sliku opcije za minimiziranje je zatim potrebno promijeniti u sliku za proširivanje kako bi se korisnicima dalo do znanja da je minimizirani izvor moguće i proširiti.

U slučaju da opcija slike minimiziranja nakon klika mišem nije *tipka_izvor_minimizirati.png*, već da je promijenjena na prethodno naveden način, to znači da je izvor minimiziran i da ga je potrebno proširiti. Iz baze podataka se na temelju *id*-a izvora dohvaća stara veličina, pa se zatim poziva metoda *resize_widget()* koja mijenja visinu izvora na temelju dohvaćene vrijednosti stare veličine. Kako se tijekom korištenja portala ne bi svaki put čekalo na dohvat stare veličine iz baze kada je to nepotrebno, odnosno kada varijabla *\$stara_velicina* već ima dodijeljenu vrijednost, još jednom se poziva metoda *resize_widget()* i to izvan bloka Ajax zahtjeva.

Zbog toga što se u ovom slučaju mijenja veličina izvora, to je potrebno i ažurirati u bazi podataka, što se i izvršava pozivom funkcije *spremi_pozicije()* čiji je sadržaj jednak onoj funkciji definiranoj kod inicijalizacije objekta *gridster* kod mijenjanja veličine i pozicije unutar atributa *resize* i *draggable*. Također se mijenja i slika opcije za minimiziranje, što je također potrebno ažurirati u bazi podataka, pa se zato poziva funkcija *azuriraj_min_sliku()* čiji je sadržaj prikazan u sljedećem isječku.

```
function azuriraj_min_sliku() {  
  
    $.each($(".izvor .izvor_naziv .izvor_opcije .minimizirati"),  
    function () {  
        var $id = $(this).parents().eq(3).attr("id");  
        var $min_slika = $(this).children(":first").attr("src");  
        $.get("baza_azuriraj_min_sliku.php?r=" +  
            $id + "&s=" + $min_slika);  
    });  
}
```

Pomoću metode *each()* se iz svakog izvora na portalu dohvaća *id* i *src* atribut slike koji se prenose PHP skripti *baza_azuriraj_min_sliku.php* preko GET parametra, a skripta u bazi podataka ažurira naziv slike za izvor na temelju prenesenog *id*-a.

Osnovne funkcionalnosti vezane uz maksimiziranje izvora su prikazane u JavaScript kôdu u sljedećem isječku.

```

$("body").on("click",
".gridster ul li div.izvor .izvor_naziv .izvor_opcije .mijenjati",
function () {

    if ($(this).children(":first").attr("src")
        === "slike/tipka_izvor_vratiti.png") {
        $(this).parents().eq(3).toggleClass("max");
        $(this).children(":first").attr("src",
            "slike/tipka_izvor_maksimizirati.png");
    }

    else {
        $(this).parents().eq(3).toggleClass("max");
        $(this).children(":first").attr("src",
            "slike/tipka_izvor_vratiti.png");
    }

});

```

Nakon klika mišem na `<div>` element s opcijom maksimiziranja se provjerava slika prvog elementa djeteta, te ako je ona *tipka_izvor_vratiti.png*, to znači da je izvor već maksimiziran, te da je istog potrebno vratiti u prethodno stanje. To se postiže uklanjanjem klase *max* pozivom metode *toggleClass()* s proslijeđenim nazivom klase u parametru. Slično se izvršava i u slučaju kada je element u maksimiziranom stanju, odnosno kada slika opcije u trenutku klika mišem nije *tipka_izvor_vratiti.png*. Sadržaj klase *max* je prikazan u sljedećem isječku CSS stilskih uputa.

```

.max {
    width: 100%;
    height: 100%;
    position: fixed;
    top: 0;
    right: 0;
    left: 0;
    z-index: 100;
}

```

Element s klasom *max* će poprimiti cijelu širinu i visinu ekrana, te će pozicija takvog elementa biti fiksna. Također će element biti prikazan iznad svih elementa koji imaju svojstvo *z-index* manje od 100. Navedene stilske upute omogućuju izvoru da bude u maksimiziranom stanju i daju korisniku nesmetano iskustvo u čitanju vijesti.

Osnovne funkcionalnosti vezane uz brisanje izvora su prikazane u sljedećem isječku JavaScript kôda.

```

$("body").on("click",
".gridster ul li div.izvor .izvor_naziv .izvor_opcije .obrisati",
function () {

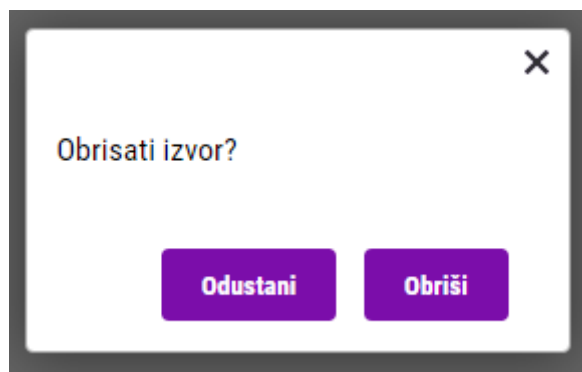
    var $izvor = $(this).parents().eq(3);
    $("#upitnik_obrisi_izvor").fadeIn(200);
    $(".obrisi_tipka").click(function () {
        gridster.remove_widget($izvor);
        $.ajax({
            url: "baza_obrisi_izvor.php?d=" + $izvor.attr("id")
        });
        spremi_pozicije();
        $("#upitnik_obrisi_izvor").fadeOut(200);
    });

    $(".upitnik_odustani, .odustani_tipka").click(function () {
        $("#upitnik_obrisi_izvor").fadeOut(200);
    });

});

```

Klikom na tipku za brisanje izvora se izvor ne briše odmah, već se otvara upitnik (prikazan na slici 13) u kojem se namjera za brisanjem izvora treba potvrditi.



Slika 13: Upitnik za brisanje

Nakon klika na tipku za potvrdu brisanja se preko *gridster* objekta poziva metoda *remove_widget()* koja kao parametar prima element kojeg briše sa stranice. Da bi izvor bio u potpunosti obrisani, potrebno ga je obrisati i u bazi podataka što se obavlja izvršavanjem PHP skripte *baza_obrisi_izvor.php* na temelju proslijeđenog *id*-a izvora. Pošto se u slučaju brisanja mijenja pozicija ostalih izvora, novonastalo stanje pozicija treba ažurirati pozivom funkcije *spremi_pozicije()*. Nakon brisanja se upitnik sakriva, što se obavlja i nakon klika na tipku odustani ili klikom na tipku u obliku slova „x“ u gornjem desnom kutu. Osnovna HTML struktura upitnika je prikazana u sljedećem isječku.

```

<div id="upitnik_obrisi_izvor" class="prozor_upitnik_pozadina">
  <div class="prozor_upitnik">
    
    <span class="prozor_upitnik_naslov">Obrisati izvor?</span>
    <div class="tipka_spremnik">
      <button class="odustani_tipka">Odustani</button>
      <button class="obrisi_tipka">Obriši</button>
    </div>
  </div>
</div>

```

Za danu HTML strukturu su pridružene CSS stilske upute od kojih su neke prikazane u sljedećem isječku.

```

button {
  border-radius: 4px;
  color: #fff;
  background: #7b0daa;
  padding: 10px 20px 10px 20px;
  outline: none;
}

.prozor_upitnik_pozadina {
  position: fixed;
  width: 100%;
  height: 100%;
  background: rgba(0, 0, 0, 0.5);
  margin: 0;
  padding: 0;
  top: 0;
  display: none;
  z-index: 101;
}

.prozor_upitnik {
  position: fixed;
  top: 50%;
  left: 50%;
  background-color: rgba(255, 255, 255, 1);
  padding: 10px 15px 15px 15px;
  border-radius: 4px;
  box-shadow: 0px 2px 45px 0px rgba(0, 0, 0, 0.5);
  z-index: 102;
}

.prozor_upitnik img.upitnik_odustani {
  position: absolute;
  right: 0;
  top: 0;
}

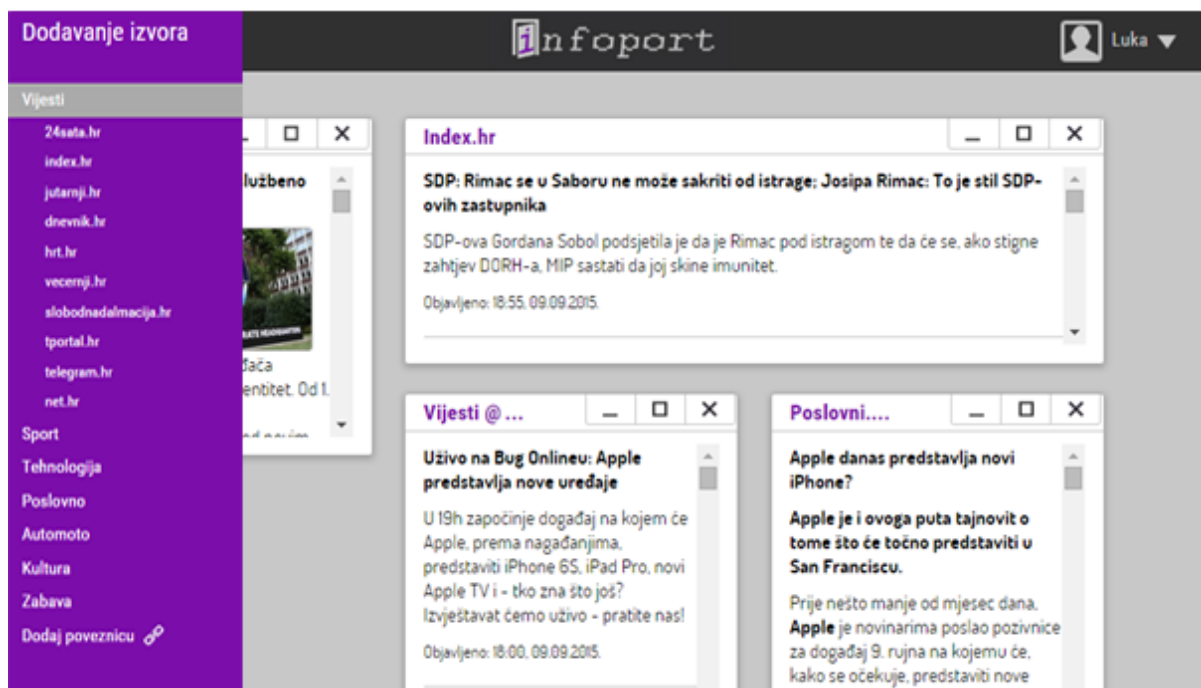
```

Pozadini upinika je dana cijela širina i visina ekrana, te fiksna pozicija i vrijednost *z-index* svojstva veća od one koju ima maksimizirani izvor tako da se upitnik pojavi ispred izvora, ako je izvor u maksimiziranom stanju. Sam upitnik se fiksno pozicionira na sredinu

ekrana, te poprima vrijednost *z-index* svojstva veću od pozadine upitnika kako bi se nalazio ispred pozadine. Da bi tipka za odustajanje, odnosno tipka u obliku slova „x“ bila smještena u gornjem desnom kutu, potrebno ju je apsolutno pozicionirati s nultim odstupanjem od vrha i desne strane.

7.4. Dodavanje izvora pomoću izbornika

U svakom trenutku bi korisnici trebali imati mogućnost dodati izvore, odnosno portale čije vijesti žele čitati. Tu mogućnost je potrebno ponuditi na nenametljiv način, ali da ta mogućnost uvijek bude na raspolaganju, pa je tipka za dodavanje izvora smještena u zaglavlje u gornjem lijevom kutu. Klikom na tu tipku se sa strane otvara izbornik, prikazan na slici 14, pomoću kojeg korisnici po kategorijama mogu pregledati dostupne portale i po želji ih dodati u glavni prostor.



Slika 14: Izbornik za dodavanje izvora

Kako bi izbornik na početku bio sakriven, a klikom na tipku u zaglavlju otkriven, prvo mu je potrebno dodati CSS svojstvo *left: -300px*, a zatim ga dinamički prikazati pomoću JavaScript kôda koji se može vidjeti u sljedećem isječku.

```
$("#dodaj_izvor").click(function () {  
    $("#dodavanje_izvora_izbornik").animate({left: "0"}, 200);  
});
```

Metodom *animate()* se izborniku pridružuje svojstvo *left: 0* i to u trajanju od 200 milisekundi s čime se postiže efekt pojave izbornika s lijeve strane. Dodavanje izvora se odvija klikom miša na *<h3>* element koji predstavlja izvor, odnosno portal i čiji se *id* proslijeđuje PHP skripti *baza_dodaj_izvor.php* kako je prikazano u sljedećem isječku.

```
$("#izvori ul li ul li h3").click(function () {
    var $izvor = $(this).attr("id");
    $("#ajax_ucitavanje").show();
    $.ajax({
        url: "baza_dodaj_izvor.php?i=" + $izvor,
        type: "GET",
        success: function (odgovor) {
            var $html = odgovor;
            gridster.add_widget($html,
                parseInt("5"), parseInt("5"), parseInt("5"), parseInt("1"));
            spremi_pozicije();
            $("#ajax_ucitavanje").hide();
        }
    });
});
```

Odgovor od Ajax zahtjeva prema PHP skripti se prima u obliku HTML sadržaja koji se na portal dodaje metodom *add_widget()*, a PHP skripta prema proslijeđenom *id*-u dohvaća poveznicu, što se može vidjeti i u sljedećem isječku u kojem je prikazan dio sadržaja PHP skripte *baza_dodaj_izvor.php*.

```
$izvor = $_GET["i"];

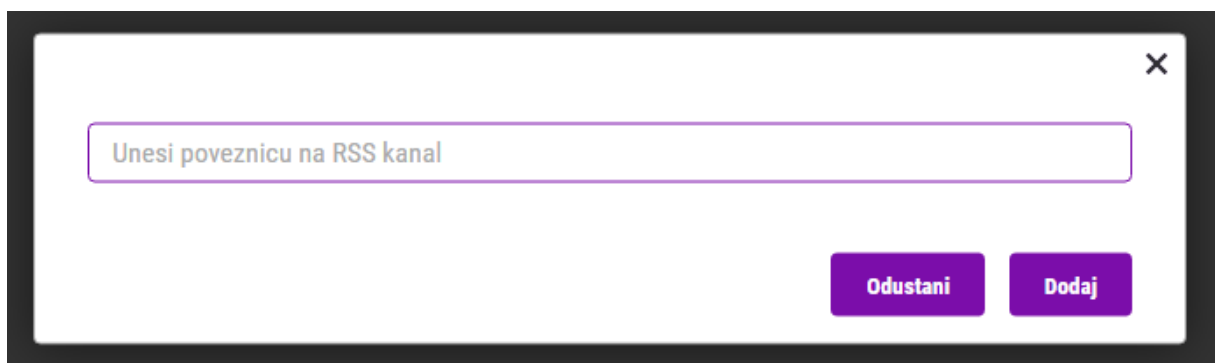
switch ($izvor) {
    case 'v1':
        $poveznica = 'http://www.24sata.hr/feeds/najnovije.xml';
        break;
    case 'v2':
        $poveznica = 'http://www.index.hr/najnovije/rss.ashx';
        break;
    case 'v3':
        $poveznica = 'http://www.jutarnji.hr/rss';
        break;
    case 'v4':
        $poveznica = 'http://rss.dnevnik.hr/index.rss';
        break;
}
```

U prikazanom isječku se vidi dio *switch* bloka unutar kojeg se provjerava proslijeđeni *id*, te se dodijeljuje odgovarajuća poveznica. Na temelju poveznice se dohvaća RSS sadržaj pomoću SimplePie biblioteke. Osim toga se i sam izvor uz *id* korisnika sprema u bazu podataka kako je i prikazano u sljedećem isječku iste PHP skripte.

```
$sql = "INSERT INTO izvori(poveznica, korisnik_id)
VALUES ('" . $poveznica . "', 1)";
mysqli_query($veza, $sql);
```

Isto kao i kod prethodnih PHP skripti, trenutno je *id* korisnika upisan kao vrijednost 1, no nakon omogućene prijave i registracije će tu vrijednost poprimiti *id* prijavljenog korisnika iz sesije.

Korisnicima također treba biti omogućeno da dodaju RSS sadržaj iz vlastite poveznice, što je omogućeno iz izbornika. Klikom na opciju za dodavanje poveznice iz izbornika se otvara forma, prikazana na slici 15, u koju se unosi poveznica, a sadržaj iz poveznice se dodaje pozivanjem prethodno spomenute PHP skripte.

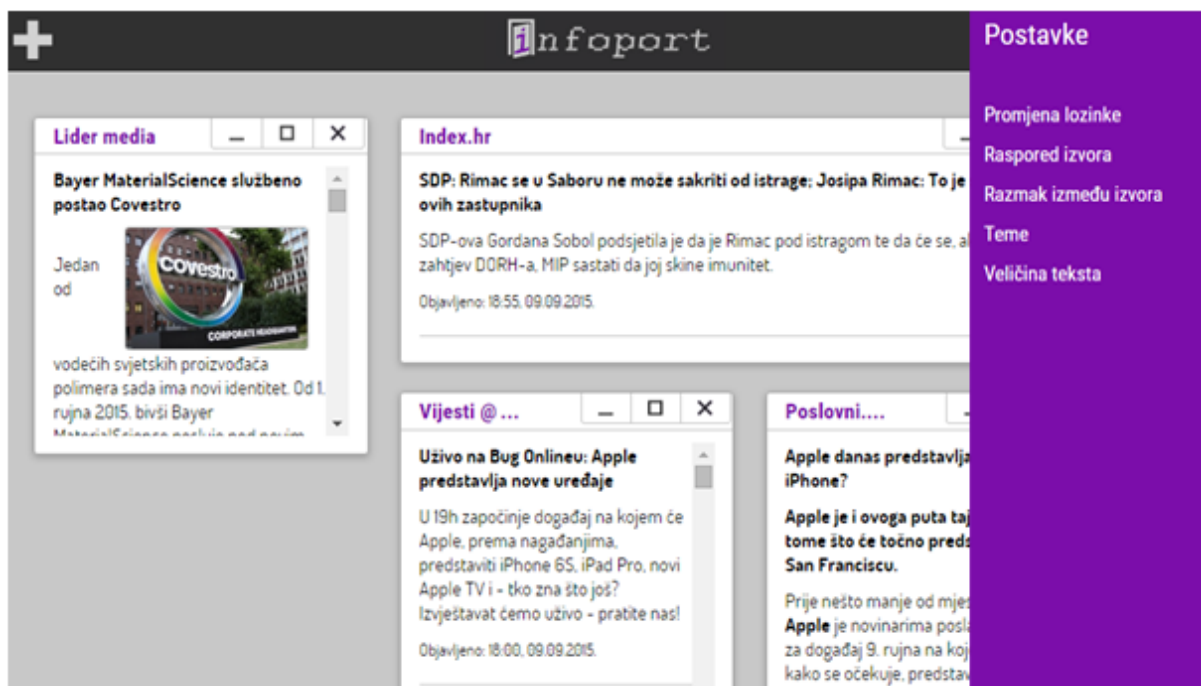


Slika 15: Forma za dodavanje poveznice na RSS sadržaj

U PHP skripti *baza_dodaj_izvor.php* se također pomoću SimplePie biblioteke provjerava ispravnost poveznice, te se u slučaju greške ispisuje odgovarajuća poruka.

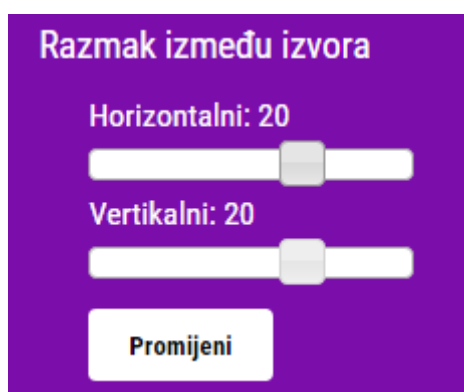
7.5. Postavke

Ukoliko korisnicima ne odgovaraju zadane postavke boja, razmaka između izvora, rasporeda izvora ili veličina teksta sadržaja izvora, iste je moguće promijeniti preko izbornika postavki koji se otvara klikom na opcije korisnika. Izbornik koji sadrži postavke je prikazan na slici 16.



Slika 16: Postavke

Promjena lozinke je korisnicima omogućena preko jednostavne forme u kojoj je potrebno ispuniti polja trenutne, nove i ponovljene lozinke. Za raspored izvora su ponuđene sljedeće postavke: miješano, jedan stupac, dva stupca i tri stupca. Ako se odabere miješani raspored, pozicije i veličine izvora će biti učitanе na temelju slučajne vrijednosti. Razmak između izvora, koji je prikazan na slici 17, je izrađen pomoću jQuery UI komponente *slider*, koja je dostupna na sljedećoj stranici: jqueryui.com/slider. Da bi bio korišten jQuery UI, potrebno je unutar `<head>` oznaka uključiti potrebnu CSS i JavaScript datoteku.



Slika 17: Postavke razmaka između izvora

Komponentu *slider* je potrebno inicijalizirati, a primjer toga je dan u sljedećem isječku.

```
$h_marg_slider = $("#h_margina").slider({
    min: 1,
    max: 30,
    value: parseInt($h_margina_vrijednost),
    slide: function (event, ui) {
        $("#h_margina_vrijednost").text(ui.value);
    }
});
```

Kao selektor se navodi element uz kojeg se veže *slider* komponenta. U atributima se navodi minimalna i maksimalna vrijednost koju može poprimiti *slider*, a također može biti i postavljena početna vrijednost preko atributa *value*. Početna vrijednost je u ovom slučaju varijabla s dohvaćenom vrijednosti prethodno spremljene vrijednosti iz baze podataka. Atributom *slide* se definira što će se točno izvršiti nakon svakog *slide* događaja, odnosno svaki puta kada korisnik mišem povuče komponentu *slider*. U ovom slučaju se izvršava ispis vrijednosti *slider* komponente u `<div>` element s *id*-em *h_margina_vrijednost*. Na isti način se mijenjanju i postavke veličine teksta.

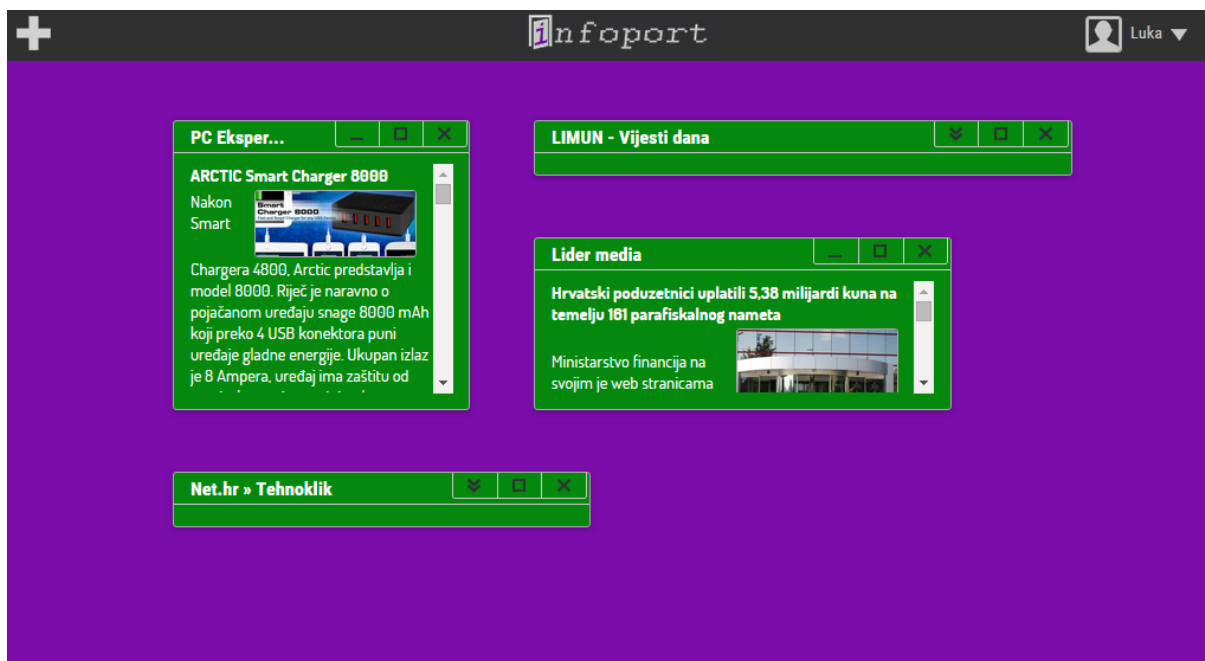
Osim navedenog, korisnici još mogu mijenjati i temu boja portala. Svaka promjena tema se ažurira u bazi podataka, a CSS stilske upute za svaku pojedinu temu se učitavaju dinamički i to preko PHP skripte koja se unutar `<head>` oznaka uključuje kao svaka CSS datoteka. Unutar te PHP skripte se dodaje sljedeći redak: `header("Content-type: text/css; charset: UTF-8")`. Kako bi CSS svojstva bila dinamički učitana, prvo se iz baze podataka dohvaća tema, pa se na temelju dohvaćene teme u varijable pridružuju odgovarajuća CSS svojstva kako je prikazano u sljedećem isječku kôda iz dijela PHP skripte *stil.php*.

```
if ($tema == 0) {
    $pozadina_body = "#c8c8c8";
    $pozadina_izvor = "#fff";
    $pozadina_opcije = "#fff";
    $tekst_izvor_naslov = "#7b0daa";
    $tekst_izvor_stavka_naslov = "#000";
    $tekst_izvor_glavno = "#000";
```

Varijable s dodijeljenim vrijednostima se koriste u definiranju CSS svojstva. Primjer jedne dodjele vrijednosti CSS svojstvu iz varijable je prikazan u sljedećem isječku.

```
div#glavno.izvor {
    background-color: <?php echo $pozadina_izvor ?>;
}
```

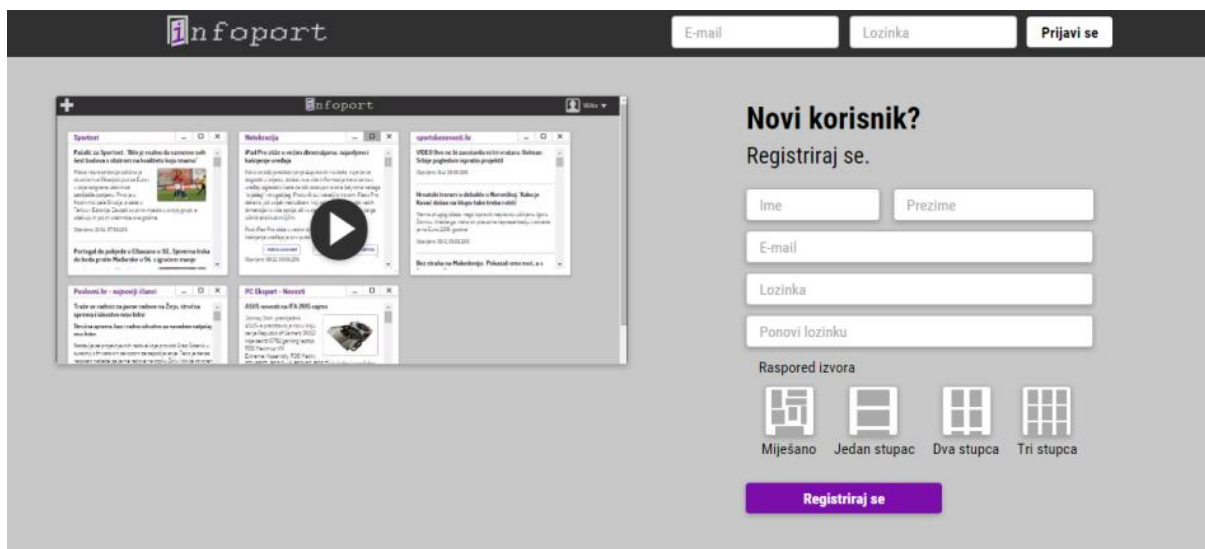
Primjer glavnog prostora portala s promijenjenom temom je prikazan na slici 18.



Slika 18: Glavni dio s promijenjenom temom

7.6. Registracija i prijava

Da bi sve postavke i svi izvori, zajedno sa s veličinama i pozicijama izvora bili trajno zapamćeni, odnosno spremljeni u bazu podataka, korisnici se trebaju registrirati i prijaviti na portal Infoport. Stranica s registracijom je prikazana na slici 19.



Za prijavu je potrebno unijeti e-poštu i lozinku, a za registraciju novih korisnika je potrebno unijeti ime, prezime, e-poštu, lozinku, te odabrati raspored izvora. Na registracijskoj stranici se također može pregledati kratki video isječak koji prikazuje korištenje portala

Infoport kako bi korisnici dobili dojam o tome što točno mogu očekivati. Nakon registracije se novi korisnik upisuje u bazu podataka, te mu je iz svake kategorije vijesti slučajnim odabirom učitani jedan izvor, odnosno jedan portal i to prema rasporedu kojeg je korisnik odabrao kod registracije.

Nakon registracije (ili prijave) se korisnik dodaje u sesiju što se može vidjeti u sljedećem isječku PHP kôda.

```
if (!isset($_SESSION)) {  
    session_start();  
    $_SESSION["korisnik_id"] = $korisnik_id;  
    $_SESSION["korisnik_ime"] = $ime;  
}
```

Superglobalnoj varijabli `$_SESSION` se pridružuju ime korisnika koje će biti ispisano u zaglavlju glavnog dijela portala i *id* korisnika koji se koristi kod svakog upita prema bazi podataka. Primjer jednog takvog upita je dan u sljedećem isječku a odnosi se na ažuriranje teme iz postavki.

```
$tema = $_GET["t"];  
$sql = "UPDATE korisnici SET tema = " . $tema .  
      " WHERE id = " . $_SESSION["korisnik_id"];  
mysqli_query($veza, $sql);
```

8. Zaključak

Personalizirani web portal može korisnicima pružiti drugačije iskustvo nego obični web portal. Iako nekim korisnicima drugačije iskustvo ne znači nužno i bolje, personalizirani web portal bi trebao biti razvijen na način da se takvim korisnicima mogućnosti personalizacije ne nude na nametljiv i neophodan način, već da se takav portal može koristiti i bez korištenja tih mogućnosti. Iz tog razloga se na početku korištenja portala Infoport korisnicima prikazuju već učitani izvori, odnosno vijesti iz portala koje korisnici mogu nesmetano čitati, bez korištenja mogućnosti personalizacije, ali su te mogućnosti uvijek dostupne.

Ukoliko se odluče, korisnici na portalu Infoport mogu dodavati dostupne izvore organizirane prema kategorijama kao što su sport, tehnologija i kultura. Prema tome se Infoport može pretvoriti i u vlastitu verziju portala u kojoj će biti dodani samo izvori iz određene kategorije. Primjerice, ako korisnike zanima samo sport, imaju mogućnost prilagoditi Infoport na način da dodaju izvore samo iz kategorije sport. Infoport se može dodatno prilagoditi dodavanjem vlastite poveznice na RSS sadržaj. Kod čitanja vijesti je svaki izvor po potrebi moguće minimizirati, maksimizirati, a i obrisati. Uz to je također moguće i izvorima mijenjati veličinu i poziciju, što može biti izuzetno korisno ukoliko se vijesti žele rasporediti po važnosti. Primjerice, ako postoje korisnici koji čitaju vijesti većinom iz samo dva portala, a ostale samo na brzinu pregledaju, na Infoport portalu se ta dva portala mogu povećati i premjestiti na vrh stranice, dok se ostalima može veličina smanjiti, te ih se može premjestiti na dno stranice. Infoport portal također nudi promjenu boje i to odabirom unaprijed definiranih tema. Za korisnike koji imaju potrebu promijeniti veličinu teksta sadržaja, razmak između izvora ili sam raspored izvora na stranici, za navedeno postoje postavke ponuđene preko izbornika.

Infoport korisnicima na jednom mjestu omogućuje čitanje vijesti iz više portala, a personalizacijom samog portala se korisnicima dopušta stvaranje osobnog i jedinstvenog iskustva.

9. Literatura

- [1] Adida B (2007) *The Browser as a Secure Platform for Loosely Coupled, Private-Data Mashups*. U: Zbornik radova prve radionice W2SP 2007: *Web 2.0 Security & Privacy 2007*, konferencije 2007 IEEE Symposium on Security and Privacy, 24. svibnja 2007. Oakland, CA.
- [2] Al-Qirim NAY (2004) *Electronic Commerce in Small to Medium-sized Enterprises: Frameworks, Issues, and Implications*. Hershey, PA: Idea Group Publishing.
- [3] Bacic M, Chmielewski J (2013) Usefulness of New HTML5 and CSS3 Features for Internet Applications on Mobile Devices. U: Kiełtyka L (ur.) *Information Technologies in Organizations - Management and Applications of Multimedia*. Torun: Wydawnictwa Towarzystwa Naukowego Organizacji i Kierownictwa - Dom Organizatora, pp. 97-110.
- [4] Balachandar K (2012) *Best Practices for Design, Development and Testing Mobile Portals*. Preuzeto 22. lipnja 2015. s https://www.ibm.com/developerworks/community/blogs/96960515-2ea1-4391-8170-b0515d08e4da/entry/best_practices_for_design_development_and_testing_mobile_portals5?lang=en.
- [5] Beam MA (2011) *Personalized News: How Filters Shape Online News Reading Behavior*. Doktorska disertacija, The Ohio State University.
- [6] Benbya H, Passiante G, Belbaly NA (2004) *Corporate portal: a tool for knowledge management synchronization*. International Journal of Information Management 24: 201-220.
- [7] Berners-Lee T, Connelly D (1995). *RFC 1866 - Hypertext Markup Language - 2.0*. Internet Engineering Task Force. Preuzeto 26. srpnja 2015. s <http://tools.ietf.org/html/rfc1866>.
- [8] Bidgoli H (2004a) *The Internet Encyclopedia, Volume 3*. Hoboken, New Jersey: John Wiley & Sons.
- [9] Bidgoli H (2004b) *The Internet Encyclopedia, Volume 2*. Hoboken, New Jersey: John Wiley & Sons.
- [10] Bowen JP, Filippini-Fantoni S (2004) *Personalization and the web from a museum perspective*. Museums and the Web 2004. Preuzeto 13. srpnja 2015. s http://www.researchgate.net/publication/239066637_Personalization_and_the_web_from_a_museum_perspective
- [11] Brecht RM (2013) *The Rise of Personalization in 2014*. Preuzeto 28. lipnja 2015. s <http://www.dmn3.com/dmn3-blog/the-most-important-2014-digital-marketing-trend>.
- [12] Browne D (2008) *On the Internet, It's All About 'My'*. Preuzeto 13. lipnja 2015. s http://www.nytimes.com/2008/04/20/fashion/20website.html?pagewanted=print&_r=0.
- [13] Buytaert D (2010) *The history of MySQL AB*. Preuzeto 5. kolovoza 2015. s <http://buytaert.net/the-history-of-mysql-ab>.
- [14] Cailliau R (1995) *A Short History of the Web*. Text of a speech delivered at the launching of the European branch of the W3 Consortium. Preuzeto 6. lipnja 2015. s http://www.netvalley.com/archives/mirrors/robert_cailliau_speech.htm.
- [15] Champeon S (2001) *JavaScript: How Did We Get Here?*. Preuzeto 30. srpnja 2015. s http://archive.oreilly.com/pub/a/javascript/2001/04/06/js_history.html.

- [16] Collins MJ (1997) *A Push In The Right Direction*. Preuzeto 13. lipnja 2015. s http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol2/mjc4/.
- [17] Constantinides E (2010) *Survival in the era of the empowered customer: Turning the Web 2.0 menace into a strategic opportunity*. Preuzeto 6. lipnja 2015. s http://www.researchgate.net/profile/Efthymios_Constantinides/publication/228394303_Survival_in_the_era_of_the_empowered_customer_Turning_the_Web_2.0_menace_into_a_strategic_opportunity/links/0fcfd50bfaf633dfea000000.pdf
- [18] Cs.Tcd.Ie (2000) *ISO/IEC 15445:2000(E)*. Preuzeto 26. srpnja 2015. s <https://www.cs.tcd.ie/misc/15445/15445.HTML>.
- [19] Daniel F, Matera M (2014) *Mashups: Concepts, Models and Architectures*. Springer.
- [20] DB-Engines.com (2015) *DB-Engines Ranking - popularity ranking of relational DBMS*. Preuzeto 5. kolovoza 2015. s <http://db-engines.com/en/ranking/relational+dbms>.
- [21] Dholakia RR, Zhao M, Dholakia N, Fortin DR (2000) *Interactivity and Revisits to Websites: A Theoretical Framework*. Preuzeto 19. srpnja 2015. s <http://ritim.cba.uri.edu/wp2001/wpdone3/Interactivity.pdf>
- [22] Dickerson J (2012) *What is the difference between Customization and Personalization?* Preuzeto 16. srpnja 2015. s <http://www.josephdickerson.com/blog/2012/06/15/what-is-the-difference-between-customization-and-personalization/>
- [23] Fichter D (2009) What Is a Mashup?. U: Engard NC (ur.) *Library Mashups: Exploring New Ways to Deliver Library Data*. Medford, NJ: Information Today, pp. 3-17.
- [24] Garret JJ (2005) *Ajax: A New Approach to Web Applications*. Preuzeto 31. srpnja 2015. s <https://web.archive.org/web/20080702075113/http://www.adaptivepath.com/ideas/essays/archives/000385.php>.
- [25] Guenther K (1999) *The Web Gets Pushy*. CIO 12(7): 66.
- [26] Gyara F, Sachdev T (2007) *Personalization of Portals*. Preuzeto 1. srpnja 2015. s <http://www.infosys.com/consulting/information-management/white-papers/Documents/personalizing-portals.pdf>
- [27] Himelstein L, Siklos R (1999) *PointCast: The Rise and Fall of an Internet Star*. Preuzeto 13. lipnja 2015. s http://www.businessweek.com/1999/99_17/b3626167.htm.
- [28] Hock R (2005) *Yahoo! to the Max: An Extreme Searcher Guide*. Medford, NJ: Information Today.
- [29] Holzner S (2006) *Secrets of RSS*. San Francisco: Peachpit Press.
- [30] Housley S (2005) *Developing Content Sites As Marketing Tools*. Preuzeto 24. lipnja 2015. s <http://archive.smallsitenews.com/2005/0524.html>.
- [31] Hughes S, Zmievski A (2002) *PHP Developer's Cookbook*. Carmel: Sams Publishing.
- [32] Jacso P (1999) *Rise of the Personalized Web Portals*. Information Today 16(7): 25-26. Preuzeto 28. lipnja 2015. s <http://www.jacso.info/PDFs/jacso-riseofportals.pdf>.
- [33] Jalal D, Al-Debei MM (2012) *Portals and Task Innovation: A Theoretical Framework Founded on Business Intelligence Thinking*. The Eleventh Annual International Conference

on Business Intelligence and Knowledge Economy 2012: 1-20. Preuzeto 13. lipnja 2015. s <http://arxiv.org/abs/1208.0892v1>.

[34] Kim W (2002) *Personalization: Definition, Status, and Challenges ahead*. Journal of Object Technology 1(1): 29-40. Preuzeto 28. lipnja 2015. s http://www.jot.fm/issues/issue_2002_05/column3/.

[35] Koschmider A, Torres V, Pelechano V (2009) Elucidating the mashup hype: Definition, challenges, methodical guide and tools for mashups. U: Zbornik radova druge radionice *Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*, 18. međunarodne World Wide Web konferencije, 20. travnja 2009. Madrid.

[36] Krill P (2015) *It's official: ECMAScript 6 is approved*. Preuzeto 30. srpnja 2015. s <http://www.infoworld.com/article/2937716/javascript/its-official-ecmascript-6-is-approved.html>.

[37] Lie HW (1994) *Cascading HTML style sheets -- a proposal*. Preuzeto 29. srpnja 2015. s <http://www.w3.org/People/howcome/p/cascade.html>.

[38] Lie HW, Bos B (1996) *Cascading Style Sheets, level 1*. W3C. Preuzeto 29. srpnja 2015. s <http://www.w3.org/TR/2008/REC-CSS1-20080411/>.

[39] Lie HW, Bos B, Lilley C, Jacobs I (1998) *Cascading Style Sheets, Level 2*. W3C. Preuzeto 29. srpnja 2015. s <http://www.w3.org/TR/2008/REC-CSS2-20080411/>.

[40] Lie HW, Bos B, Hickson I, Çelik T, Bos B (2011) *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. W3C. Preuzeto 29. srpnja 2015. s <http://www.w3.org/TR/CSS2/>.

[41] Mansourvar M, Yasi NM (2010) *Web portal As A Knowledge Management System In The Universities*. World Academy of Science, Engineering and Technology 70: 968-974.

[42] Marjanovic D (2012) *PDO vs. MySQLi: Which Should You Use?*. Preuzeto 5. kolovoza 2015. s <http://code.tutsplus.com/tutorials/pdo-vs-mysqli-which-should-you-use--net-24059>.

[43] Miletsky (2009) *Principles of Internet Marketing: New Tools and Methods for Web Developers*. Boston: Cengage Learning.

[44] Mozilla Developer Network (2015) *CSS3*. Preuzeto 29. srpnja 2015. s <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>.

[45] Murugesan S (2010) *Web X.0: A Road Map*. U: Murugesan S (ur.) *Handbook of Research on Web 2.0, 3.0, and X.0: Technologies, Business, and Social Applications*. Hershey, PA: IGI Global, pp. 1-11.

[46] MySQL.com (2015) *About MySQL*. Preuzeto 5. kolovoza 2015. s <https://www.mysql.com/about/>.

[47] Nations D (2015) *What is the Difference Between a Web Mashup and a Distributed Portal?* Preuzeto 24. srpnja s http://webtrends.about.com/od/webmashups/a/web_mashup.htm.

[48] Nottingham M, Sayre R (2005) *RFC 4287 - The Atom Syndication Format*. IETF. Preuzeto 7. kolovoza 2015. s <https://tools.ietf.org/html/rfc4287>.

[49] O'Reiley T (2005) *What Is Web 2.0*. Preuzeto 6. lipnja 2015. s <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>.

- [50] Oracle.com (2015) *Oracle and Sun Microsystems*. Preuzeto 5. kolovoza 2015. s <http://www.oracle.com/us/sun/index.htm>.
- [51] Paoli J, Bray T, Sperberg-McQueen CM (1998) *Extensible Markup Language (XML) 1.0*. W3C. Preuzeto 26. srpnja 2015. s <http://www.w3.org/TR/1998/REC-xml-19980210>.
- [52] PHP.net (2015) *What is PHP?*. Preuzeto 2. kolovoza 2015. s <http://php.net/manual/en/intro-what-is.php>.
- [53] Raggett D (1997) *HTML 3.2 Reference Specification*. W3C. Preuzeto 26. srpnja 2015. s <http://www.w3.org/TR/REC-html32>.
- [54] Raju Y, Babu DS (2015) *A NOVEL APPROACHES IN WEB MINING TECHNIQUES IN CASE OF WEB PERSONALIZATION*. International Journal of Research in Computer Applications and Robotics 3(2): 6-12. Preuzeto 21. srpnja s http://ijrcar.com/Volume_3_Issue_2/v3i203.pdf
- [55] Sampson D, Manouselis N (2005) A Flexible Evaluation Framework for Web Portals Based on Multi-Criteria Analysis. U: Tatnall A (ur.) *Web Portals: The New Gateways to Internet Information and Services*. Hershey, PA: Idea Group Publishing, pp. 185-211.
- [56] Sfetcu N (2014) *Web Design & Development*. Nicolae Sfetcu.
- [57] Shtykh R, Jin Q (2003) Push Technology and New Media Community. U: Bianchi-Berthouze N (ur.) Zbornik radova treće međunarodne radionice *Databases in Networked Information Systems*, 22.-24. rujna 2003. Aizu, Japan. Springer.
- [58] Tatnall A (2005) Portals, Portals Everywhere. U: Tatnall A (ur.) *Web Portals: The New Gateways to Internet Information and Services*. Hershey, PA: Idea Group Publishing, pp. 1-14.
- [59] Trachtenberg A (2004) *Why PHP 5 Rocks!*. Preuzeto 2. kolovoza 2015. s <http://www.onlamp.com/pub/a/php/2004/07/15/UpgradePHP5.html>.
- [60] Ullman C (2007) *What is Ajax?*. Preuzeto 31. srpnja 2015. s <http://www.wrox.com/WileyCDA/Section/id-303217.html>.
- [61] Valentine T, Reid J (2013) *JavaScript Programmer's Reference*. New York: Apress.
- [62] van Wijngaarden T (2009) *Ajax for Content Management*. Magistarski rad, VU University Amsterdam. Preuzeto 1. kolovoza 2015. s <http://www.cs.vu.nl/~eliens/im/@archive/student/mCS-MM/thesis-teunis.pdf>.
- [63] W3C (1998) *Document Object Model (DOM) Level 1 Specification*. Preuzeto 30. srpnja 2015. s <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/>.
- [64] W3C (2002) *XHTML 1.0: The Extensible HyperText Markup Language (Second Edition)*. Preuzeto 26. srpnja 2015. s <http://www.w3.org/TR/xhtml1/>.
- [65] W3C (2014) *HTML5*. Preuzeto 27. srpnja 2015. s <http://www.w3.org/TR/html5/>.
- [66] W3Techs (2015a) *Usage Statistics and Market Share of JavaScript Libraries for Websites*. Preuzeto 30. srpnja 2015. s http://w3techs.com/technologies/overview/javascript_library/all.
- [67] W3Techs (2015b) *Usage Statistics and Market Share of Server-side Programming Languages for Websites*. Preuzeto 3. kolovoza 2015. s http://w3techs.com/technologies/overview/programming_language/all.

- [68] White CJ (2011) *Safeguarding the Corporate Portal: A Review of Portal Security*. DataBase Associates. Preuzeto 20. lipnja 2015. s http://www.providersedge.com/docs/km_articles/Safeguarding_the_Corporate_Portal.pdf.
- [69] Winer D (2003) *RSS 2.0 Specification*. Berkman Center for Internet & Society at Harvard Law School. Preuzeto 5. kolovoza 2015. s <http://cyber.law.harvard.edu/rss/rss.html>.
- [70] Yang H (2012) *Mosaic - The Browser Popularized the Web*. Preuzeto 6. lipnja 2015. s <http://www.herongyang.com/Computer-History/WWW-Mosaic-The-Browser-Popularized-the-Web.html>.
- [71] Zakas NC, McPeak J, Fawcett J (2011) *Professional Ajax*. Hoboken, New Jersey: John Wiley & Sons.
- [72] Zhou J (2003) *A History of Web Portals and Their Development in Libraries*. Information Technology and Libraries 22(3): 119-128.