

Razvoj naprednih korisničkih sučelja u web 2.0 okruženju

Čačić, Marko

Master's thesis / Diplomski rad

2012

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:087480>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-10-08**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

MARKO ČAČIĆ

**RAZVOJ NAPREDNIH KORISNIČKIH
SUČELJA U WEB 2.0 OKRUŽENJU**

DIPLOMSKI RAD

Zagreb, 2012



Sveučilište u Zagrebu
Grafički fakultet

MARKO ČAČIĆ

**RAZVOJ NAPREDNIH KORISNIČKIH
SUČELJA U WEB 2.0 OKRUŽENJU**

DIPLOMSKI RAD

Mentor:

izv. prof. dr. sc. Nikola Mrvac

Student:

Marko Čačić

Zagreb, 2012

Zahvaljujem se mentoru dr. sc. Nikoli Mrvcu i neposrednom voditelju Mili Matijeviću, dipl. ing. na stručnim savjetima prilikom izrade ovog diplomskog rada.

SAŽETAK:

Moderne *Web* stranice i aplikacije zahtijevaju korisnička sučelja koja svojom naprednom tehničkom izvedbom dosljedno prate preporuke grafičkog oblikovanja u *Web 2.0* okruženju. Širenjem Interneta i razvojem *Web* tehnologija, omogućena je postupna transformacija tekstualno orijentiranih *Web* stranica u privlačne multimedijske proizvode ^[1]. Obzirom na visoku razinu potrebne interaktivnosti s korisnicima, prepoznata je potreba za grafičkim stručnjacima, specijaliziranim u polju oblikovanja korisničkih sučelja u skladu s grafičkim standardima *Web 2.0* okruženja ^[2]. Stoga, ovaj diplomski rad obrađuje metode razvoja naprednih korisničkih sučelja pomoću modernih programskih tehnologija, istražuje utjecaj pojedinih karakteristika *Web 2.0* okruženja na njihovo grafičko i funkcionalno oblikovanje te daje pregled tehnologija koje se tek razvijaju, a još će više naglasiti važnost grafičke komponente *Web-a*, što je od posebnog značaja za grafičke stručnjake specijalizirane u području multimedije. Praktičan dio diplomskog rada prikazati će postupak izrade *Web* orijentiranog korisničkog sučelja u skladu s grafičkim standardima *Web 2.0* okruženja, koje zahvaljujući naprednoj tehničkoj izvedbi karakterizira visoka iskoristivost izvornog koda te mogućnost prilagodbe različitim rezolucijama zaslona.

Ključne riječi: *Web* stranice, aplikacije, korisnička sučelja, *Web 2.0* okruženje, multimedija, grafički standardi, tehnologije, tehnička izvedba

SADRŽAJ

1. UVOD.....	1
2. KORISNIČKA SUČELJA U WEB 2.0 OKRUŽENJU	3
2.1. Povijest razvoja Web orijentiranih korisničkih sučelja	3
2.2. Specifičnosti Web orijentiranih korisničkih sučelja	5
2.3. Web 2.0 okruženje	12
2.4. Karakteristike Web 2.0 okruženja	16
2.4.1. Pravila grafičkog i funkcionalnog oblikovanja u Web 2.0 okruženju.....	17
2.4.2. Utjecaj karakteristika Web 2.0 okruženja na grafičko i funkcionalno oblikovanje korisničkih sučelja	22
3. RAZVOJ NAPREDNIH KORISNIČKIH SUČELJA U WEB 2.0 OKRUŽENJU	31
3.1. Razvojne tehnologije.....	31
3.1.1. HTML opisni jezik.....	31
3.1.2. CSS opisni jezik	33
3.1.3 JavaScript programski jezik	35
3.1.4. PHP programski jezik.....	36
3.1.5. Tehnologije budućnosti	37
3.2. Praktičan razvoj naprednog korisničkog sučelja u Web 2.0 okruženju	38
4. ZAKLJUČAK	60
5. LITERATURA.....	62
POPIS PRILOGA	65
POPIS OZNAKA I KRATICA.....	66
POPIS SLIKA.....	67
POPIS TABLICA	69
POPIS IZVORNIH KODOVA.....	70

1. UVOD

U vremenima kada Internet nije bio tako široko dostupan kao danas, područje izrade *Web* stranica gotovo se u cijelosti smatralo poslom informatičara, no razvojem tehnologije prepoznata je potreba za uvođenjem grafičkih stručnjaka, specijaliziranih u polju oblikovanja atraktivnih sučelja bez kojih se danas ne može zamisliti niti jedna ozbiljnija *Web* stranica ili aplikacija [3]. Moderan *Web* postupno se transformirao iz prvobitno statičnog oblika u interaktivan multimedijски sustav koji omogućuje brzu razmjenu informacija među pojedinim korisnicima, a ne više isključivo između jednog autora i zainteresirane publike [2]. Novi način organizacije i uporabe sadržaja na *Web-u*, simbolično nazvan *Web 2.0* trendom, nametnuo je niz pravila grafičkog i funkcionalnog oblikovanja korisničkih sučelja kako bi se ostvarila potrebna razina interaktivnosti i kvalitetan korisnički doživljaj (eng. *user experience*) [4].

U početku, sučelja navedenih karakteristika izrađivali su gotovo isključivo informatički stručnjaci zbog potrebe za programiranjem složenih programskih komponenti, no razvoj specijaliziranih paketa unaprijed definiranih programskih funkcija omogućio je izgradnju naprednih korisničkih sučelja na brz i jednostavan način [5, 6], pri čemu se grafički stručnjak u najvećoj mjeri može usmjeriti prema izradi kvalitetne grafike.

Unazad nekoliko godina, od pojave prvih pametnih telefona (eng. *smart phones*) s kvalitetnim zaslonom s mogućnošću prikaza velikog broja boja, prikaz grafike na *Web-u* počeo se znatno prilagođavati prikazu na mobilnim uređajima. Danas, pri oblikovanju *Web* sadržaja moramo svakako voditi računa o mogućnostima prilagodbe sučelja različitim rezolucijama zaslona zbog velikog broja različitih formata mobilnih uređaja. Također, kako bi se postigla puna dosljednost i profesionalan izgled te funkcionalnost, prilagođeno sučelje mora poštovati iste općeprihvaćene standarde *Web 2.0* okruženja.

Razvoj *Web* tehnologija u bliskoj budućnosti, a pogotovo dolazak *HTML5* i *CSS3* opisnih jezika kao novih, opće prihvaćenih i implementiranih standarda,

omogućit će daljnji razvoj *Web*-a prema grafički iznimno bogatoj, multimedijски usmjerenoj platformi visoke razine interaktivnosti [7]. Stoga, segment razvoja korisničkih sučelja u *Web 2.0* okruženju potrebno je prepoznati i prihvatiti kao novo, atraktivno i iznimno profitabilno područje unutar grafičke struke [2,8] te ga početi tretirati ravnopravno s tradicionalnim poslovima grafičkih stručnjaka.

Stoga, ovaj diplomski rad obrađuje metode razvoja naprednih korisničkih sučelja pomoću modernih programskih tehnologija, istražuje utjecaj pojedinih karakteristika *Web 2.0* okruženja na njihovo grafičko i funkcionalno oblikovanje te daje pregled tehnologija koje se tek razvijaju.

Osim teoretske obrade navedene problematike, diplomski rad će istražiti na koji način pojedine karakteristike *Web 2.0* okruženja utječu na grafičko i funkcionalno oblikovanje korisničkih sučelja. Prvi dio istraživanja će determinirati osnovne karakteristike *Web 2.0* okruženja proučavanjem *Web 2.0* aplikacija i relevantne literature te na temelju dobivenih podataka izdvojiti one karakteristike koje bi se mogle primijeniti kao reference pri razvoju korisničkih sučelja. Drugi dio istraživanja će na temelju izdvojenih karakteristika analizirati sučelja nekoliko *Web 2.0* aplikacija te odrediti u kojoj mjeri su određene karakteristike utjecale na njihovo grafičko i funkcionalno oblikovanje. Dobiveni rezultati istraživanja primijenit će se pri izradi naprednog korisničkog sučelja u praktičnom dijelu diplomskog rada.

2. KORISNIČKA SUČELJA U WEB 2.0 OKRUŽENJU

2.1. Povijest razvoja Web orijentiranih korisničkih sučelja

Web je izvorno zamišljen kao *Internet* servis za objavljivanje znanstvenih sadržaja koji će se moći lako pretraživati i pregledavati od strane zainteresiranih korisnika, odnosno drugih znanstvenika ^[9]. U skladu s namjenom i tada dostupnom tehnologijom, bio je izveden u cijelosti tekstualno, bez suvišnih grafičkih elemenata koji bi odvlačili pozornost i nepotrebno opterećivali prijenos podataka ^[9, 10]. Jedini aktivni elementi korisničkih sučelja prvih *Web* stranica bile su tekstualne poveznice s drugim stranicama (eng. *links*) i jednostavni formulari za pretragu sadržaja prema upisanim ključnim riječima (eng. *search forms*), izvedeni pomoću prve radne verzije *HTML* opisnog jezika (eng. *HyperText Markup Language*) i *Perl* programskog jezika ^[10].

Nedugo zatim prepoznate su komercijalne mogućnosti *Web-a* kao medija za publikaciju reklamnih sadržaja velikom broju potencijalnih kupaca, pri čemu se u središtu zbivanja našao običan korisnik, a ne više znanstvenik ^[9]. Tada je započeo brz razvoj *Web-a* prema globalnoj komunikacijskoj i prezentacijskoj platformi. Sve veća dostupnost moderne tehnologije i razvoj brzina pristupa *Internetu*, omogućile su transformaciju prvobitno tekstualno orijentiranih *Web* stranica u multimedijske proizvode koji uspješno objedinjuju sliku, tekst, pa čak i zvuk. U to vrijeme dolazi do naglog razvoja *Web* orijentiranih tehnologija poput *CSS* opisnog jezika (eng. *Cascading Style Sheets*) te novih, poboljšanih verzija *HTML* jezika ^[10]. No, pravu revoluciju u oblikovanju *Web* sučelja započela je tvrtka *Netscape Communications* razvojem *JavaScript* programskog jezika koji je i danas jedan od najpopularnijih *Web* tehnologija ^[6]. Brojne *Web* stranice koristile su razne interaktivne navigacijske trake, stilizirane satove i slične atraktivne sitnice izvedene pomoću *JavaScript* jezika, što je posebno pridonosilo njihovoj vizualnoj upečatljivosti i prepoznatljivosti.

Tadašnji model upravljanja sadržajem na *Web-u* uglavnom se bazirao na protoku informacija između jednog autora stranice i niza zainteresiranih korisnika [11]. Međutim, uskoro je postalo jasno kako takav model nema budućnost, već da je potrebno izbrisati stroge uloge i korisnicima omogućiti stvaranje sadržaja [11]. Opće prihvaćanje novog modela, simbolično nazvanog *Web 2.0*, rezultiralo je razvojem velikog broja uspješnih *on-line* projekata poput *YouTube* i *Vimeo* servisa za dijeljenje multimedijских zapisa, zatim jedne od najbrže rastućih enciklopedija na Svijetu – *Wikipedije*, brojnih servisa za dijeljenje (eng. *sharing*) fotografija (*Flickr*, *Panoramio*), dokumenata (*Box*) i raznih datoteka (*DropBox*, *RapidShare*) te svakako neizostavnih društvenih *on-line* okupljališta, poput svjetski popularnih *Facebook* i *MySpace* mreža. Nekada obične, statične *HTML* stranice počele su se transformirati u složene *Web* aplikacije. Osim potpuno novog načina upravljanja sadržajem, tehnološki zahtjevi novih *Web 2.0* orijentiranih projekata ubrzali su stvaranje cijelog niza naprednih multimedijских tehnologija [11, 12].

Jednostavnost, upotrebljivost (eng. *usability*) i vizualna atraktivnost nametnuli su se kao standardi u oblikovanju modernih korisničkih sučelja *Web* aplikacija. Stručnjaci i informatički entuzijasti, okupljeni oko ideje slobodnog softvera, razvili su niz alata poput *jQuery JavaScript* radnog okvira (eng. *framework*) s namjerom da razvoj atraktivnih korisničkih sučelja u velikoj mjeri olakšaju te ujedno njihovu kvalitetu podignu na višu razinu [13]. U segmentu planiranja, vizualnog oblikovanja i realizacije sučelja modernih *Web* projekata, prepoznata je potreba za stručnim grafičkim kadrom obrazovanim u području primjene multimedije. Daljnjim razvojem *Web* tehnologija, a posebice dolaskom *HTML5* i *CSS3* jezika koji obiluju mogućnostima animacije i stiliziranja [7], u potpunosti će se prepoznati značaj grafičke struke u području razvoja naprednih korisničkih sučelja u *Web 2.0* okruženju.

2.2. Specifičnosti Web orijentiranih korisničkih sučelja

U svakodnevnom životu susrećemo se s nizom različitih korisničkih sučelja koji nam omogućavaju interakciju s uređajima. Moderni televizori, telefoni, igraće konzole, automobili i niz drugih elektroničkih uređaja isporučuju se s ugrađenim softverom za kontrolu i upravljanje od strane korisnika, posredstvom pažljivo projektiranog korisničkog sučelja.

Popularizacija informatike i razvoj računalne tehnologije potakli su stvaranje velikog broja različitih aplikacija za stolna (eng. *desktop*) računala. Prvobitni računalni programi, aktualni u periodu između 1950. i 1960. godine, koristili su tzv. *batch* sučelja čija je posebnost u tome da ne ostvaruju interakciju s korisnikom već se svi potrebni parametri definiraju prethodno procesiranjem [14]. Tijekom procesiranja sučelje više ne nudi mogućnost unosa novih parametara, sve dok ne zaprimi rezultat i ponudi ponovno procesiranje. U periodu između 1960. i 1990. godine razvijaju se interaktivna sučelja, prvobitno u obliku komandne linije, a zatim kao tekstualna sučelja koja se razvojem miša kao nove ulazne jedinice računala, transformiraju u grafička korisnička sučelja [14, 15]. 1990. godine, u samim počecima razvoja *Web*-a, započinje i postupan razvoj *Web* orijentiranih korisničkih sučelja kao posrednika između korisnika i *Web* stranica, odnosno aplikacija u *Web* okruženju [10].

Korisnička sučelja aplikacija u klasičnom grafičkom okruženju operativnih sustava stolnih računala i *laptopa* uvelike se razlikuju od *Web* orijentiranih sučelja, ponajprije zahvaljujući tehničkim karakteristikama pojedine platforme. Tradicionalne, *desktop* aplikacije izvršavaju se na računalu korisnika i sve potrebne datoteke prethodno su postavljene u memoriju računala postupkom instalacije, čime je osiguran brz i neometan rad jer nema potrebe za učitavanjem novih datoteka. *Web* aplikacije prenose podatke korisniku po pravilima *HTTP* protokola aplikativne razine Interneta, koji se bazira na ideji *request/response* načina prijenosa podataka između korisnika i poslužitelja (eng. *server*), odnosno računala na mreži koji sadrži tražene podatke [16, 17].

Navedeni način prijenosa podataka na *Web-u* uspostavljen je zbog potrebne optimizacije veze između korisnika i poslužitelja, kako se resursi poslužitelja ne bi nepotrebno trošili na održavanje stalne povezanosti (eng. *connection*) dok korisnik pregledava trenutno generiranu stranicu. Izravna posljedica takvog načina prijenosa podataka je stalno osvježavanje (eng. *refresh*) prikaza, što nije slučaj kod *desktop* aplikacija koje zbog toga odaju dojam fluidnosti i veće atraktivnosti sučelja.

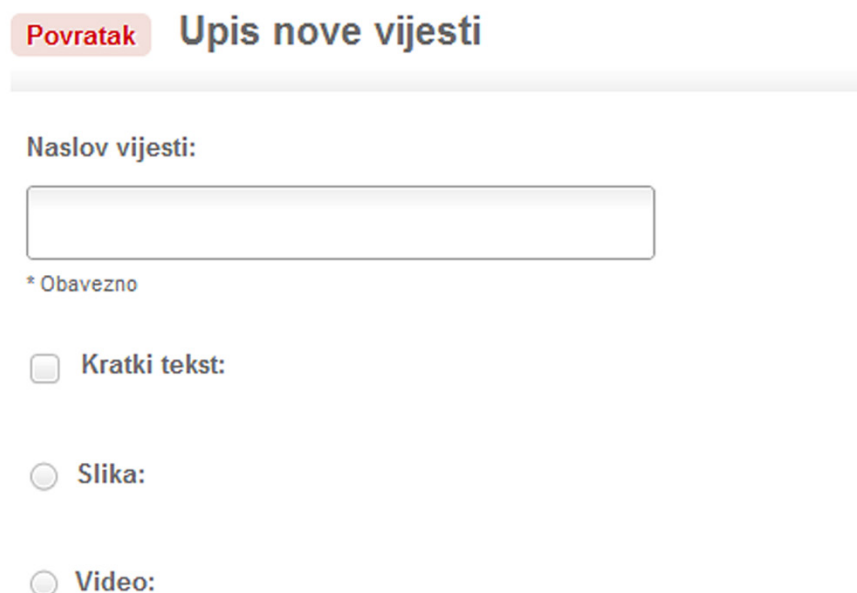
Problem osvježavanja najviše dolazi do izražaja kod *Web* aplikacija koje zahtijevaju visoku razinu interakcije s korisnikom, a ne koriste nikakve metode oblikovanja korisničkog sučelja pomoću *JavaScript* programskog jezika [13, 18] koji se izvršava na strani korisnika, odnosno interpretira izravno u korisnikovom *Web* pregledniku nakon što se stranica podataka prenijela s poslužitelja.

Ukoliko *Web* aplikacija sadrži interaktivne elemente koji ostvaruju unaprijed predvidljiv rezultat interakcije, moguće ih je oblikovati tako da se svi potrebni elementi i mogući rezultati prenesu u jednoj *server request/response* sesiji, a konačno ponašanje elementa definira pomoću *JavaScript* jezika i prikladno stilizira *CSS* (eng. *Cascading Style Sheets*) definicijama.

U suprotnom, svi elementi sučelja koji su interaktivnog karaktera, pri ostvarenoj interakciji s korisnikom morali bi pozivati određene funkcije poslužitelja posredstvom *URL* (eng. *Uniform Resource Locator*) adrese i prijenosom određene vrijednosti varijable koja bi na *server* strani generirala novu *HTML* (eng. *Hypertext Markup Language*) stranicu s rezultatom interakcije. U slučaju velikih formulara za unos podataka čiji raspored elemenata ovisi o odabirima korisnika, navedeni problem uvelike bi otežao rad s aplikacijom, a upravljanje unosom postalo bi vrlo komplicirano zbog prirode *HTTP* protokola [16]. Programeri takve *Web* aplikacije morali bi izraditi kompliciran sustav međuspremnikâ u koje bi pohranjivali podatke pri svakoj interakciji korisnika s određenim elementom. U suprotnom sav uneseni sadržaj do točke interakcije bio bi izgubljen zbog stalne potrebe za osvježavanjem trenutne *HTML* stranice. Stoga, današnja moderna sučelja koriste napredne tehnike

kontrole interakcije korisnika i elemenata pomoću *JavaScript* radnih okvira (eng. *framework*), pri čemu valja spomenuti *jQuery JavaScript framework* koji je trenutno jedan od najpopularnijih i najkvalitetnijih [5, 6, 13].

Kako bi praktično ilustrirali kvalitetno rješenje navedenog problema, promotrimo primjer unosa sadržaja kroz dinamični *HTML* formular ostvaren pomoću unaprijed definiranih funkcija *jQuery* radnog okvira. Primjerice, ukoliko u *CMS* (eng. *Content Management System*) *Web* aplikaciji imamo mogućnost unosa vijesti i pripadajuće multimedije, pri čemu se može unijeti samo jedna vrsta multimedije – slika ili video, korisničko sučelje možemo lako i brzo realizirati korištenjem metode oblikovanja sučelja pomoću *JavaScript* jezika i pripadajućih *CSS* definicija.



Povratak Upis nove vijesti

Naslov vijesti:

* Obavezno

Kratki tekst:

Slika:

Video:

Slika 2.1. Prikaz segmenta sučelja stiliziranog *HTML* formulara za unos vijesti u *CMS* sustav

Prikazani dio formulara za unos sadržaja sastoji se od *HTML input* elemenata tipa *text* („Naslov vijesti“), *checkbox* („Kratki tekst“) i *radio* („Slika“, „Video“). Polje „Naslov vijesti“ nije interaktivnog karaktera nego je statično definirano i ne mijenja oblik pri interakciji s korisnikom, već samo prihvaća tekstualni unos. Klikom na polje „Kratki tekst“, bez osvježavanja cijele prikazane

stranice prikazuje se dodatno polje za unos kratkog teksta vijesti. Kako ne dolazi do osvježavanja stranice, odnosno generiranja novog upita prema poslužitelju, već se promjene događaju isključivo na strani korisnika, prethodno uneseni podaci ostaju upisani u elemente sve do konačne predaje uređenih podataka poslužitelju pritiskom *submit* gumba pri dnu formulara.

Povratak Upis nove vijesti

Naslov vijesti:

* Obavezno

Kratki tekst:

Dopuštena duljina 125 znakova uključujući tri točke "..."

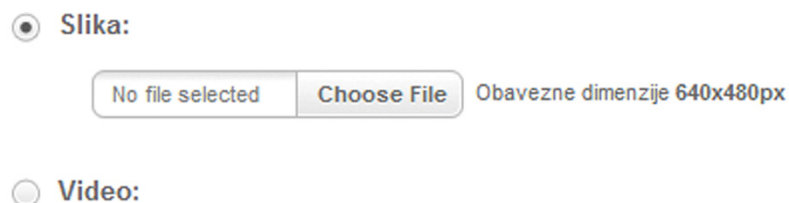
Slika 2.2. Prikaz segmenta sučelja stiliziranog *HTML* formulara nakon odabira opcije „Kratki tekst“

Ukoliko u prikazano *textarea* polje upišemo sažetak, odnosno kratki tekst vijesti i isključimo ga pritiskom na *checkbox* „Kratki tekst“, tekst će ostati upisan u polje, no neće se prenijeti na poslužitelj prilikom konačne predaje formulara. Navedena funkcionalnost izvedena je tako da je *checkbox* „Kratki tekst“ prozvan roditeljem (eng. *parent*) *textarea* polju pa pri obradi prenesenih podataka, ukoliko roditelj nije označen (vrijednost postavljena na istina - *true*), vrijednost polja djeteta se zanemaruje. Imenovanje pojedinih polja i dodjela pripadajućih statičnih vrijednosti izvodi se pomoću *name* i *value HTML* atributa.

Kako prikazani formular omogućuje unos samo jedne vrste multimedije, odnosno sliku ili video, način unosa i prikaza pojedinih polja modificiran je na

način da odabir jedne vrste multimedije skriva polja za unos druge vrste i obratno. Polja su i u ovome slučaju izvedena kao roditelji i djeca da bi se omogućila kontrola prenesenih podataka i odbacio višak, a vizualna izmjena prikazanih polja realizirana je pomoću *jQuery* funkcija *show()* i *hide()* te CSS definicija potrebnih za početno skrivanje interaktivnih elemenata pri prvom ispisu formulara, dok još nije odabrana niti jedna od ponuđenih opcija sučelja.

Primjerice, ukoliko odaberemo *checkbox* „Slika“, prikazat će se dodatno polje za prijenos (eng. *upload*) slikovne datoteke, dok su elementi koje prikazuje opcija „Video“ i dalje skriveni.



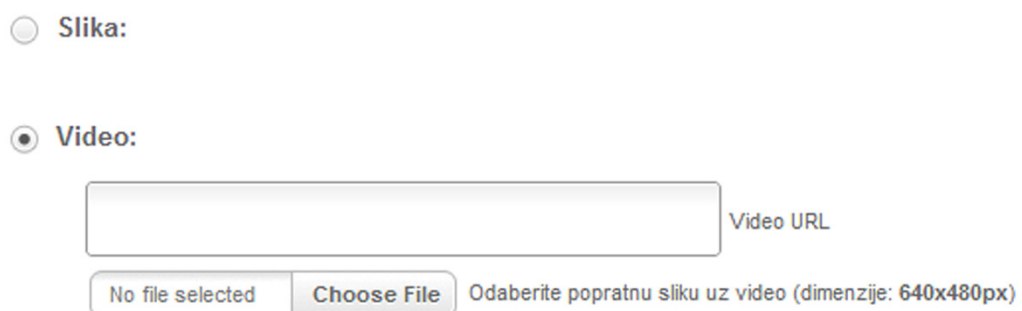
Slika:

No file selected Choose File Obavezne dimenzije 640x480px

Video:

Slika 2.3. Prikaz segmenta sučelja stiliziranog *HTML* formulara nakon odabira opcije „Slika“

Ukoliko odaberemo *checkbox* „Video“, sučelje će sakriti polje za odabir slikovne datoteke i prikazati pripadajuća polja za definiranje video parametara, u ovome slučaju *URL streaming video player-a* i popratnu sliku.



Slika:

Video:

Video URL

No file selected Choose File Odaberite popratnu sliku uz video (dimenzije: 640x480px)

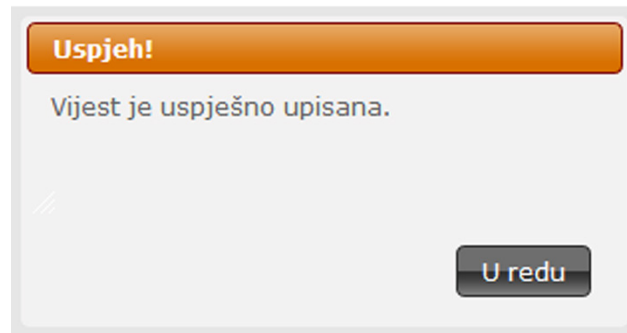
Slika 2.4. Prikaz segmenta sučelja stiliziranog *HTML* formulara nakon odabira opcije „Video“

Ispunjena polja prenose se na poslužitelj klikom na *submit* gumb koji se obično nalazi smješten pri dnu formulara, nakon čega je poželjno ispisati poruku

o uspješno ili neuspješno izvršenoj operaciji kako bi korisnik znao treba li ponoviti postupak unosa.



Slika 2.5. HTML *submit* gumb naziva „Spremi“ i *anchor* gumb za poništavanje trenutne akcije



Slika 2.6. *jQuery UI* dijaloški okvir (eng. *dialog*) s porukom o uspješno obavljenoj akciji

Uz prikazane metode dinamičkog oblikovanja sučelja, pri izradi *Web* orijentirane aplikacije koja zahtijeva određen stupanj interakcije s korisnikom poželjno je ugraditi sustav kontrole unosa podataka, prethodno završnoj predaji podataka *serveru* [11, 13]. Naime, ukoliko korisnik na valjani način ne ispuni sva potrebna polja, odnosno ne ostvari minimalne nužne preduvjete za uspješno izvršenje određene akcije *Web* aplikacije, a poruka o grešci prikaže se tek nakon predaje podataka *serveru*, svi uneseni podaci će biti izgubljeni, što je poseban problem kod ispunjavanja većih formulara. Kvalitetno rješenje navedenog problema je implementacija tzv. *JavaScript* validacijskih sustava koji na strani klijenta u realnom vremenu kontroliraju unose i odmah ispisuju poruku o grešci te onemogućuju predaju podataka *serveru* sve dok sva polja nisu pravilno ispunjena [11, 13].

Sučelja *Web* aplikacija zbog prirode *HTTP* protokola moraju biti kvalitetno optimizirana kako pri radu s aplikacijom ne bi dolazilo do usporavanja veze s korisnikom [16]. Stoga, grafika sučelja obično se pretvara u

optimizirani .jpeg ili .gif format, odnosno .png format ukoliko je potrebno zadržati visoku kvalitetu uz određenu razinu transparentnosti [19]. Također, pri oblikovanju sučelja poželjno je koristiti ujednačene grafičke elemente i pri osvježavanju izmjenjivati tek nužne dijelove izvornog koda (eng. *source code*) kako bi se u što većoj mjeri koristio međuspremnik (eng. *cache*) Web preglednika korisnika i tako više puta iskoristila većina prethodno prenesenog koda [3,4,7,19].

Navedene metode odnose se za prirodne (eng. *native*) Web aplikacije izrađene pomoću *native Web* tehnologija, što vrijedi za *HTML* dokumente oblikovane pomoću *HTML*, *CSS* i *JavaScript* jezika. Web aplikacije izrađene pomoću *Adobe Flash* alata ili bilo kojeg drugog rješenja koje zahtijeva upotrebu specijaliziranog *player-a*, obično se korisniku isporučuju kao nekoliko međusobno povezanih većih datoteka pa se prijenos veće količine podataka izvršava odjednom u više navrata. Također, takve aplikacije izrađene su po pravilima vlastitog razvojnog okruženja pa nije nužno poštivanje klasičnih pravila oblikovanja u Web okruženju.

Unazad nekoliko godina, Web aplikacije sve više prakticiraju asinkrono učitavanje sadržaja sa servera i tako uspješno zaobilaze problem stalne potrebe osvježavanja [13, 18]. Naime, pomoću *AJAX* (eng. *Asynchronous JavaScript and XML*) tehnologije, ostvaruje se interakcija sa serverom u pozadini aplikacije, što omogućuje slanje zahtjeva i primanje odgovora, odnosno rezultata izvršavanja određene operacije, izravno na trenutno aktivnoj *HTML* stranici, bez potrebe za osvježavanjem sadržaja [18]. Metoda je postala popularna zbog svoje atraktivnosti jer iznimno pridonosi kvalitetnom korisničkom doživljaju [4, 18], što posebno dolazi do izražaja u velikim multimedijским Web aplikacijama visokog stupnja potrebne interakcije, poput *Google Maps* ili *Google Gmail* aplikacija. Nove verzije *jQuery* radnog okvira dolaze s ugrađenom *AJAX* podrškom [13], što je širokom krugu korisnika

omogućilo implementaciju navedene funkcionalnosti pomoću tek nekoliko jednostavnih linija koda.

Primjerice, ukoliko na *Web* stranici imamo kontakt formular, možemo ga izraditi tako da korisniku ponudimo unos podataka u polja formulara kojemu dodijelimo attribute *id="kontakt-forma"* i *name="kontakt-forma"*. Svako polje također je označeno *id* i *name* atributom kako bi mogli jednostavno implementirati kontrolu unosa te unesene vrijednosti kasnije procesirati pomoću *AJAX* metode. Pri unosu podataka, polja reagiraju na pogrešne unose i kratkim porukama obavještavaju korisnika. Nakon što korisnik ispuni sva predviđena polja, omogućava se predaja podataka klikom *submit* gumba. Proslijeđene unose prihvaća *jQuery \$.ajax()* metoda koja ih prilagođava daljnjoj obradi te *POST* ili *GET* metodom usmjerava *server side* datoteci. Na strani servera, odredišna datoteka prihvaća vrijednosti i izvršava slanje poruke pomoću vlastitih funkcija. Nakon izvršenih svih operacija, ispisuje se poruka o uspjehu ili neuspjehu, bez potrebe za osvježavanjem trenutne *HTML* stranice s kontakt formularom.

Na prikazane načine moderna *Web* orijentirana korisnička sučelja savladavaju ograničenja *Web-a* kao medija i pripadajućeg *HTTP* protokola te po upotrebljivosti (eng. *usability*), brzini izvršavanja, vizualnom izgledu i općem korisničkom iskustvu (eng. *user experience*, *UX*) konkuriraju sučeljima tradicionalnih *desktop* aplikacija [4, 13, 16, 18].

2.3. Web 2.0 okruženje

Web kao medij namijenjen brzom razmjeni informacija, u svojim počecima funkcionirao je kao klasična oglasna ploča. Takav statični model upravljanja sadržajem bazirao se na jednosmjernom protoku informacija između jednog autora *Web* stranice i niza zainteresiranih korisnika. Međutim, uskoro je postalo jasno kako *Web* u takvome obliku nema budućnost, već da je potrebno izbrisati stroge uloge te korisnicima omogućiti međusobnu razmjenu informacija kroz

interakciju s *Web* stranicom [1, 11, 12]. Opće prihvaćanje novog modela, simbolično nazvanog *Web 2.0*, rezultiralo je razvojem velikog broja uspješnih *on-line* projekata poput *YouTube* i *Vimeo* servisa za dijeljenje multimedijских zapisa, zatim jedne od najbrže rastućih enciklopedija na Svijetu – *Wikipedije*, brojnih servisa za dijeljenje (eng. *sharing*) fotografija (*Flickr*, *Panoramio*), dokumenata (*Box*) i raznih datoteka (*DropBox*, *RapidShare*) te svakako neizostavnih društvenih *on-line* okupljališta, poput svjetski popularnih *Facebook* i *MySpace* mreža [20]. Nekada obične, statične *HTML* stranice počele su se transformirati u složene *Web* aplikacije visokog stupnja interaktivnosti. Osim potpuno novog načina upravljanja sadržajem, tehnološki zahtjevi novih *Web 2.0* orijentiranih projekata ubrzali su stvaranje cijelog niza naprednih multimedijских tehnologija [7, 20].

Web 2.0 model bazira se na međuljudskoj komunikaciji kao temelju stvaranja korisnog sadržaja. Kako se koncept razmjene informacija u potpunosti razlikuje od prethodnog *Web 1.0* modela, stvorila se potreba za razvojem novih metoda objave sadržaja [1, 11, 12, 20, 21]. U početku, *Web* stranice su se izrađivale tako što je autor posredstvom *HTML editor-a* izravno u kod stranice unosio sadržaj i poveznice te ih zatim fizički prenosio na željeni *server*. *Web 2.0* model briše stroge uloge te korisniku omogućuje interakciju sa sučeljem u svrhu stvaranja sadržaja [20, 21]. Postojeće metode izrade *Web* stranica morale su se unaprijediti na način da konačan proizvod nije statičnog već interaktivnog karaktera. Tada se uz razne forume pojavljuju i prvi sustavi koji omogućuju pisanje vlastitih *on-line* dnevnika, tzv. *blogovi*, što je kovanica od riječi *web* i *log*. Sadržaj *blog-a* obično je organiziran tako da se autorovi tekstovi prikazuju u vremenskom slijedu, a ispod svakog teksta omogućena je rasprava s drugim korisnicima kroz objavu komentara. Time je u svojem osnovnom obliku ostvarena mogućnost brze i jednostavne razmjene mišljenja među korisnicima. Danas svatko može izraditi vlastiti *blog* već za nekoliko minuta korištenjem specijaliziranih servisa poput *Google Blogger* ili *on-line WordPress* platforme, što je svakako velik doprinos globalnom informacijskom društvu. Sadržaji

blogova mogu se lako dijeliti s drugim *Web* stranicama i aplikacija posredstvom *RSS* (eng. *Really Simple Syndication*) tehnologije [21], što je rezultiralo stvaranjem agregatnih aplikacija koje prikupljaju specifične tematske sadržaje i na smislen način prezentiraju ih vlastitim korisnicima. Posredstvom iste tehnologije, korisnici čak mogu pratiti željene sadržaje više različitih *blogova* na raznim mobilnim uređajima bez potrebe korištenja *Web* preglednika, već kroz jednostavan *RSS* čitač.

RSS tehnologija pronašla je veliku primjenu od strane novinskih i informativnih portala. Naime, za razliku od novina u tiskanom formatu koje izlaze jednom dnevno, portali s vijestima su stalno dostupni i uobičajeno svakih nekoliko sati objavljuju novu informaciju. S vremenom, primjetilo se da informacije kasno stižu do publike, što je osobit problem ukoliko se radi o važnim obavijestima. Postalo je očito kako je potrebno pod hitno pronaći i uvesti valjan način distribucije informacija osobama ili drugim web sjedištima kojima su te vijesti od osobite važnosti. Obzirom da takav mehanizam distribucije mora biti prilagodljiv različitim načinima čitanja (putem *Web* ili *desktop* sučelja te mobilnih uređaja), rješenje je pronađeno upravo u *RSS* tehnologiji koja se bazira na lako proširivom *XML* (eng. *eXtensible Markup Language*) jeziku [21]. Jednostavnost i konačna smislenost prenesenih podataka osnovne su značajke *RSS* tehnologije koja je zahvaljujući svojim dobrim karakteristikama postala općeprihvaćeni standard za brzu razmjenu vijesti u *Web* okruženju.

Uz brzi protok informacija te razmjenu mišljenja, važni čimbenici *Web 2.0* okruženja su društveno umrežavanje (eng. *social networking*), međusobna suradnja (eng. *colaboration*) i dijeljenje sadržaja (eng. *content sharing*) [12, 20, 21]. Milijuni korisnika svakodnevno koriste neku od općih (eng. *general*) ili tematskih društvenih mreža. *Facebook*, *Twitter*, *Google Plus*, *Hi5* i *LinkedIn* trenutno su najpopularnija virtualna okupljališta, što je prepoznato i od strane brojnih investitora pa se često može čuti kako vrijednost navedenih projekata dostiže čak milijarde *USD*. Zbog velikog broja stalnih korisnika, takvo okruženje povoljno je za oglašavanje pa mnoge tvrtke objavljuju tzv. *fan* stranice svojih

proizvoda ili dogovaraju plaćene oglase [20, 22]. Osim virtualnog druženja, tematski projekti poput *LinkedIn* mreže omogućuju povezivanje poslodavaca i kvalitetnog kadra putem osobnih profila, a tzv. *micro blogging* servisi poput *Twitter-a* čak omogućuju da saznamo trenutno aktualne događaje već nekoliko minuta nakon što se dogode, što je odličan korak prema stvaranju *real time* tražilice u budućnosti.

Pojam međusobne suradnje korisnika (eng. *user colaboration*) osobito dolazi do izražaja u raznim *project management* aplikacijama poput popularnog *Basecamp* rješenja te *e-learning* sustavima namijenjenim učenicima i studentima [1, 11]. Takve aplikacije omogućuju razmjenu znanja i zadavanje individualnih zadataka, olakšavaju organiziranje timova te u bilo kojem trenutku pružaju uvid u proizašle rezultate. Međusobna suradnja korisnika ogleda se i u označavanju objavljenog sadržaja na *Web-u* prikladnim *tagovima* koji se zatim organiziraju u smislene skupine i predočavaju na prikladan način, što omogućuje dinamičko razvrstavanje sadržaja prema vrsti i temi [11, 12, 21]. Takav način organizacije sadržaja putem međusobne interakcije korisnika naziva se *folksonomy* [11], što je kovanica od riječi *folks* (hrv. društvo) i *taxonomy* (hrv. metoda uređivanja).

Dijeljenje sadržaja među korisnicima temelj je poslovanja velikog broja *Web 2.0* servisa [11, 20, 21]. Glazbeni i video sadržaji sve do unazad nekoliko godina razmjenjivali su se gotovo isključivo posredstvom optičkih medija i analognih kazeta, dok je danas dovoljno imati tek prosječan priključak na Internet pa da posredstvom nekog od popularnih servisa pogledamo dugometražni film, glazbeni spot ili čak listu pjesama organiziranih u kategorije od strane korisnika. Unatoč pravnim problemima, uz određene kompromise s organizacijama za borbu protiv piratstva, servisi za dijeljenje sadržaja napreduju iz dana u dan, a broj pogledanih i preslušanih naslova mjeri se u milijunima. Osim velike dostupnosti različitog sadržaja, servisi poput *YouTube-a* omogućili

su raznim anonimnim izvođačima besplatnu promociju, što je rezultiralo nizom uspješnih priča [20, 21].

Ukoliko sagledamo sve navedeno, možemo zaključiti kako je temelj *Web 2.0* modela stvaranje okruženja koje svakodnevno producira koristan sadržaj, pruža pravovremeno informiranje, potiče suradnju među korisnicima te raznim tehnikama označavanja, grupiranja i organizacije omogućava postupnu transformaciju *Web-a* prema globalnoj bazi korisnog znanja.

2.4. Karakteristike Web 2.0 okruženja

Unazad desetak godina, *Web* se transformirao iz običnog statičnog medija u interaktivnu multimedijску platformu. Aplikacije su nekada bile isključivo vezane uz tradicionalna okruženja *desktop* računala, dok je *Web* služio tek kao statično sredstvo informiranja [12, 21]. U međuvremenu razvio se cijeli niz naprednih informatičkih i računalnih tehnologija koje su omogućile kvalitetno izvršavanje aplikacija u *Web* okruženju. Implementacija novih tehnologija rezultirala je stvaranjem velikog broja korisnih *Web* aplikacija poput *Google Maps*, *Google Gmail* ili *Scribd* servisa gdje milijuni korisnika međusobno dijele pisane sadržaje poput raznih uputa, besplatnih časopisa i korisnih dokumenata. Unazad nekoliko godina, od pojave prvih moćnih pametnih telefona (eng. *smart phones*), mnoge *Web* aplikacije prilagodile su svoje usluge korištenju pomoću mobilnih uređaja. Primjerice, podaci o lokacijama upisani u *Google Maps* sustav mogu se koristiti za *GPS* navigaciju, što je odličan primjer integracije *Web* i mobilne aplikacije u svrhu ostvarenja korisne funkcionalnosti, čemu teži *Web 2.0* model. Mnogi korisni sadržaji na *Web-u* kategoriziraju se i organiziraju posredstvom raznih servisa za prikupljanje sadržaja od strane korisnika, kao što su *Reddit*, *Digg* ili *Newsvine*. Milijuni korisnika dnevno pregleda takve sadržaje te ih pritom ocjenjuju, komentiraju i označavaju, što vodi prema stvaranju globalnih baza korisnog sadržaja, organiziranog prema informativnosti i kvaliteti. Usluge modernih *Web* aplikacija obično su dostupne programerskoj zajednici putem specijaliziranih *API-a* (eng. *Application*

Interfaces) [11, 21] kako bi se mogle lako implementirati u nove projekte, što dovoljno govori o otvorenosti *Web 2.0* sustava na korist opće populacije.

U skladu s navedenim te prema relevantnoj literaturi, možemo zaključiti kako su temeljne karakteristike *Web 2.0* okruženja otvorenost, sloboda te kolektivna inteligencija kao posljedica aktivnog sudjelovanja korisnika u oblikovanju informacija [1, 11, 12, 21]. Visok stupanj potrebne interakcije s korisnicima te potpuno drugačiji sustav kontrole i objave sadržaja stvorio je potrebu razvoja prikladnih korisničkih sučelja. U skladu s potrebom, definiran je niz pravila grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju kako bi se korisnika potaknulo na interakciju i tijekom cijelog procesa zadržao kvalitetan korisnički doživljaj.

Stoga, u nastavku rada determinirat ćemo osnovne karakteristike *Web 2.0* okruženja koje bi se mogle primijeniti kao reference pri razvoju korisničkih sučelja te proučavanjem nekoliko popularnih aplikacija i relevantne literature definirati osnovna pravila grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju.

2.4.1. Pravila grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju

Web 2.0 okruženje potiče korisnike na stvaranje i dijeljenje sadržaja, što zahtijeva visoku razinu potrebne interaktivnosti sa sučeljem aplikacije [11, 21]. U skladu s potrebom, definiran je niz pravila grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju kako bi se korisnika potaknulo na interakciju i tijekom cijelog procesa zadržao kvalitetan korisnički doživljaj.

Prema relevantnoj literaturi iz područja oblikovanja (eng. *design*) u *Web 2.0* okruženju, definirane su opće karakteristike sučelja modernih *Web* aplikacija. Proučavanjem navedenih karakteristika te analizom sučelja nekoliko

popularnih aplikacija, definirat ćemo opća pravila grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju.

Sučelja modernih *Web 2.0* aplikacija karakteriziraju slijedeće osnovne grafičke i funkcionalne odlike:

a) Jednostavnost

Sučelja aplikacija u *Web 2.0* okruženju prožeta su jednostavnošću i čistoćom, bez nepotrebnih detalja koji korisnika mogu omesti u postizanju željene interakcije. Pri oblikovanju sučelja preporučuje se korištenje tek nužnih elemenata potrebnih za ostvarenje željene funkcionalnosti, uz stalno usmjeravanje korisnika pomoću prikladno stiliziranih poruka. Osim pri izradi sučelja različitih aplikacija, jednostavnost je prihvaćena kao standard u oblikovanju modernih *Web* stranica [2, 11, 19] pa zbog dosljednog poštivanja pravila često možemo čuti kako su to *Web 2.0* stranice, iako su statičnog karaktera.

b) Jasno naglašene površine od posebnog interesa

Površinama od posebnog interesa smatramo dijelove sučelja koji sadrže elemente potrebne za ostvarenje željene interakcije s korisnikom. Primjerice, ukoliko korisnika želimo navesti na ispunjavanje određenog formulara, tada ćemo sve ostale elemente podrediti konačnom cilju na način da površinu od posebnog interesa smjestimo na upečatljivo mjesto, vidljivo odmah pri učitavanju *HTML* stranice. Osim rasporedom, naglašavanje određene površine obično se postiže i odabirom valjane boje pozadine, postizanjem kontrasta, balansiranjem veličina pojedinih dijelova sadržaja te ugradnjom posebnih elemenata koji korisnika jednoznačno upućuju na akciju [23, 24, 25].

c) Pregledan razmještaj elemenata

Elementi sučelja moraju biti pregledno razmješteni i pažljivo organizirani kako se ne bi narušio tok procesa interakcije s korisnikom [24]. Ukoliko u prvih nekoliko sekundi nakon učitavanja trenutne stranice nije jasno vidljiva svrha i cilj pojedinih elemenata, velika je vjerojatnost da korisnik neće ostvariti interakciju na željeni način [2, 24, 25], što svakako želimo izbjeći. Pažljivo planiranje rasporeda elemenata u svrhu postizanja preglednosti i jednostavnosti posebice dolazi do izražaja kod *Web* stranica koje se bave *on-line* prodajom.

d) Smisljena i jednoznačna navigacija

Loša navigacija frustrira korisnika i u potpunosti onemogućuje kvalitetnu interakciju s aplikacijom te postizanje željenih ciljeva [23, 24]. Kvalitetno riješen sustav navigacije kroz sadržaj u svakom trenutku omogućuje korisniku orijentaciju u ovisnosti o trenutnom položaju te poduzimanje željenih akcija. Stoga, navigacija mora biti riješena jednoznačno, smisljeno i jednostavno, a trenutni položaj korisnika u sustavu mora biti jasno vidljiv i naglašen u svakom trenutku [23].

e) Jasno vođenje korisnika kroz procese pomoću sustava provjere unosa i obavještavanja u realnom vremenu

U svrhu pojednostavljenja, a time i olakšanja procesa interakcije za krajnjeg korisnika, *Web 2.0* aplikacije imaju ugrađen složen sustav provjere unosa i obavještavanja u realnom vremenu, kako bi se održao visok stupanj korisničkog doživljaja i spriječili eventualni problemi pri kasnijoj obradi podataka.

f) Atraktivno stilizirani elementi sučelja

Sučelja modernih *Web 2.0* aplikacija karakteriziraju stilizirani elementi prožeti jednostavnošću [23, 24]. Dojam atraktivnosti postiže se zaobljenim rubovima, blagim linijama, tonskim prijelazima (eng. *gradient*) i različitim tehnikama koje stvaraju dojam blage trodimenzionalnosti površine, no uz zadržanu jednostavnost i s dozom dobrog ukusa.

g) Upečatljive, izražajne, pastelne boje

Snažne, svijetle boje privlače poglede i doprinose stvaranju profesionalnog općeg dojma. Stoga, *Web* stranice i aplikacije izrađene po pravilima *Web 2.0* okruženja, obično koriste dvije do najviše tri primamljive pastelne boje kako bi privukle znatiželjne poglede i kod promatrača stvorile osjećaj ugone [2, 23, 25]. Boje se koriste i za skretanje pažnje na određenu površinu, razdvajanje sadržaja te naglašavanje važnih elemenata.

h) Jasnoća i oštrina

Pojam oštine odnosi se na kvalitetu pojedinih grafičkih elemenata ukomponiranih u konačni dizajn elemenata *Web* stranice ili aplikacije. Pri izradi grafike u grafičkom uređivaču (eng. *editor*) postoji mogućnost pojave zamagljenih (eng. *blur*) područja, pogotovo na mjestima zaobljenih rubova. Stoga, pri izradi grafike potrebno je koristiti opcije koje rubove čine jasnim i oštrim jer će se u protivnom zbog neuglednog izgleda pojedinih elemenata narušiti opći grafički dojam. Navedeni problem rješava *CSS3* jezik koji uvodi mogućnost generiranja zaobljenih rubova izravno kroz kod stranice [7, 23], bez potrebe za vanjskim grafičkim datotekama, no zbog trenutno slabe podrške od strane *Web* preglednika, *HTML5* i *CSS3* tehnike još nije preporučljivo koristiti.

i) Dosljednost

Dosljednost se odnosi na poštivanje odabranih metoda i grafičkih stilova kroz cijeli dokument. Parametre poput veličine pojedinih elemenata, oblika i veličine fonta, boja, margina te razmaka između riječi i slova potrebno je prethodno definirati i poštivati ih tijekom cijelog procesa oblikovanja *Web* stranice [2, 23, 24, 25], odnosno sučelja aplikacije. U protivnom, konačan rezultat odaje dojam neprofesionalnosti, što svakako želimo izbjeći.

j) Prilagodljivost različitim dimenzijama, odnosno rezolucijama zaslona

Unazad nekoliko godina, od pojave prvih pametnih telefona (eng. *smart phones*) s dovoljno širokim i kvalitetnim zaslonom, *Web 2.0* aplikacije počele su se prilagođavati standardima novog medija optimizacijom sučelja kako bi kvalitetno mogle funkcionirati na zaslonima različitih dimenzija. Navedeni problem danas se obično rješava tako da proizvođač aplikacije izradi više verzija sučelja te zatim pomoću dinamičkog prepoznavanja vrste uređaja prikaže valjanu verziju [24, 26]. Jednostavnija sučelja mogu se izraditi i na način da se dinamički prilagođavaju različitim dimenzijama, što se postiže izradom tekućih (eng. *fluid*) predložaka [26].

k) Kompatibilnost osnovnih funkcija s vodećim Web preglednicima

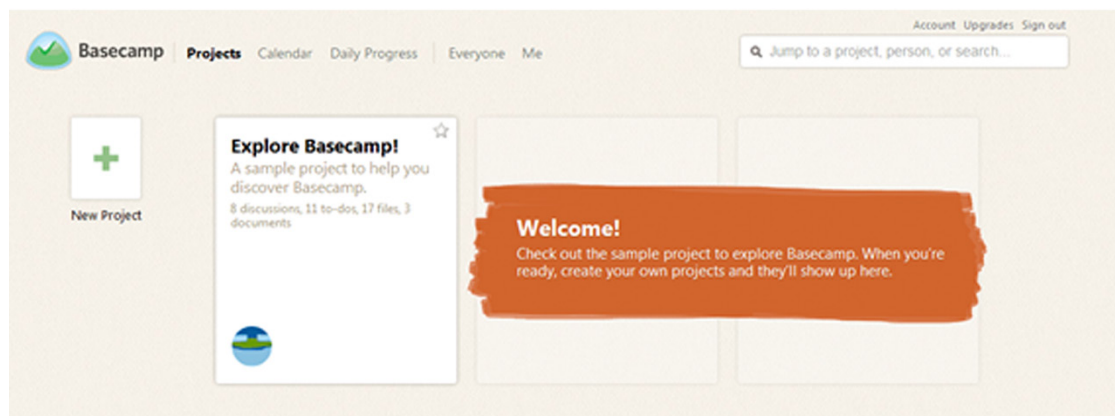
Kompatibilnost s vodećim *Web* preglednicima (eng. *cross browser compatibility*) od iznimne je važnosti za *Web 2.0* aplikacije zbog širokog kruga korisnika kojima su namijenjene. Naime, određene ugrađene funkcionalnosti aplikacije možda nisu podjednako podržane od strane svih vodećih preglednika pa se može dogoditi da rade polovično ili uopće ne rade [7, 19, 23, 24]. Također, određeni elementi sučelja mogu se drugačije prikazivati u različitim preglednicima pa je pri oblikovanju sučelja potrebno postići određeni kompromis kako bi se konačan proizvod u svim preglednicima prikazao na približno podjednak način sa zadržanim svim potrebnim funkcionalnostima.

2.4.2. Utjecaj karakteristika Web 2.0 okruženja na grafičko i funkcionalno oblikovanje korisničkih sučelja

Proučavanjem relevantne literature te promatranjem *Web 2.0* aplikacija determinirali smo temeljne karakteristike korisničkih sučelja u *Web 2.0* okruženju. Slijedeći korak je istražiti na koji način se navedene karakteristike manifestiraju u konkretnim primjerima sučelja odabranih *Web 2.0* aplikacija. Kao predmet proučavanja odabrat ćemo dvije popularne *Web 2.0* aplikacije te po prethodno određenim karakteristikama, koje su ujedno i preporuke oblikovanja u *Web 2.0* okruženju, utvrditi na koji način i u kojoj mjeri je pojedina preporuka ispoštovana pri grafičkom i funkcionalnom oblikovanju korisničkog sučelja.

Kao predmet proučavanja uzet ćemo *Basecamp project management* aplikaciju i *Stumbleupon* sustav za preporuku web sadržaja po unaprijed definiranim interesima.

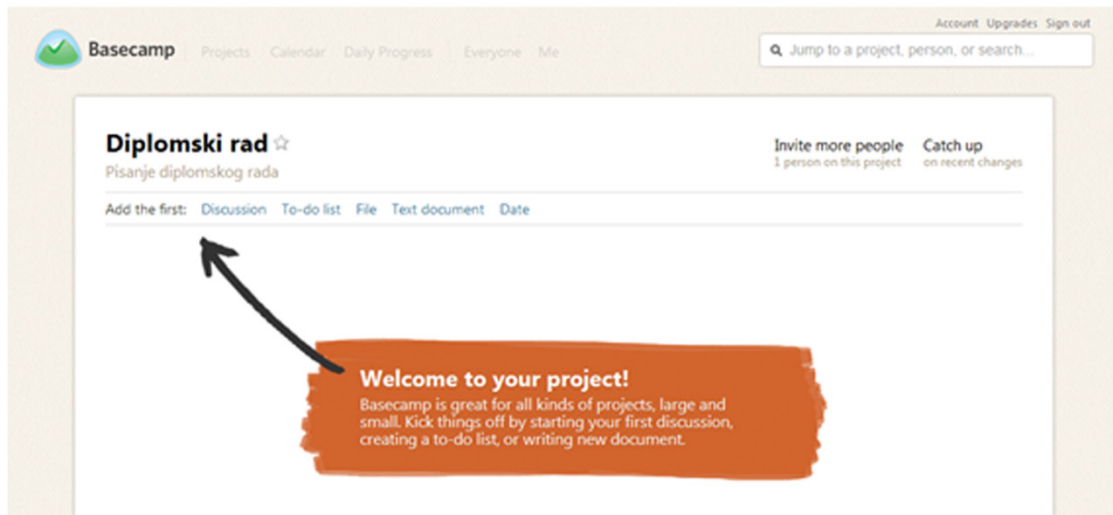
a) *Basecamp*



Slika 2.7. Prikaz početne stranice sučelja *Basecamp project management* aplikacije [27]

Basecamp je *on-line* aplikacija namijenjena vođenju projekata [27], što je po prirodi svakako kompliciran proces. Međutim, aplikacija je organizirana krajnje jednostavno, a sučelje je pregledno i kvalitetno osmišljeno. Površine od

posebnog interesa i razne poruke koje nas navode kroz sustav jasno su naglašene izražajnom pastelnom narančastom bojom, dok sučeljem dominira izrazito ugodna, blaga, svijetlo smeđa boja. Raspored elemenata je jasan i pregledan, a navigacija kroz sadržaj uvelike je olakšana jasno naznačenim obavijestima i prikladnim strjelicama koje nas navode na logičan slijed radnji.



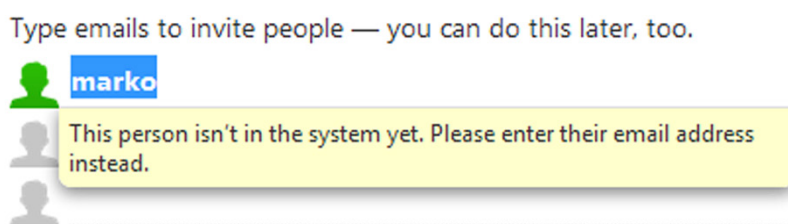
Slika 2.8. *Basecamp project management* aplikacija – prikaz stranice projekta [27]

Osnovna navigacija riješena je vrlo pregledno i jednostavno, u obliku običnih *HTML linkova* koji se podcrtavaju prilikom prelaska kursora. Sadržaji većine stranica učitavaju se asinkrono pomoću *AJAX* tehnologije, što je uklonilo potrebu stalnog osvježavanja stranica prilikom izmjene sadržaja, čime se održao visok stupanj korisničkog doživljaja tijekom cijelog vremena korištenja aplikacije.



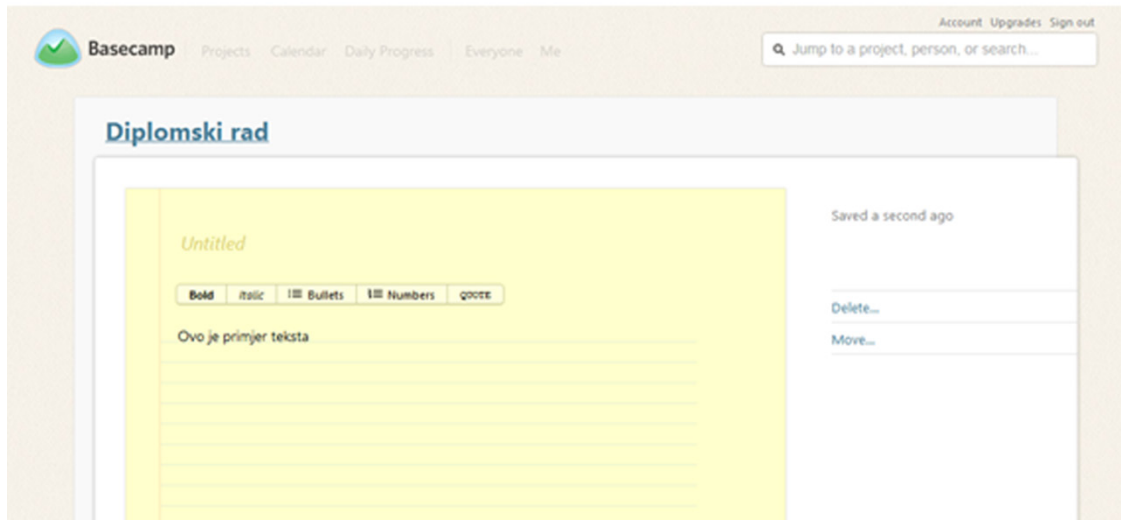
Slika 2.9. Sučelje *Basecamp project management* aplikacije – prikaz polja za unos sadržaja i obavijesti o pogrešnom unosu [27]

Polja za unos sadržaja riješena su jednostavno, no funkcionalno, uz detalje poput stiliziranog *submit* gumba i unaprijed definiranog teksta u *input* poljima koji nas navodi kroz unos potrebnih podataka. Provjera unosa riješena je tako da u realnom vremenu obavještava korisnika o greškama, ispisom prikladnih ikona. Na prikazanoj slici može se vidjeti ikona s uskličnikom koja nam signalizira kako smo u polje predviđeno za unos *e-mail* adrese upisali samo riječ bez specifičnih karakteristika poput @ znaka i naziva domene. Ukoliko grešku ne popravimo, a podatke pokušamo predati *serveru* klikom na *submit* gumb, sustav će označiti neispravan unos i ispisati dodatnu poruku, što uklanja mogućnost naknadnih problema.



Slika 2.10. *Basecamp project management* aplikacija – prikaz dodatne poruke o nelogičnom unosu [27]

Pisanje dokumenata vezanih uz pojedini projekt vrši se kroz poseban *editor*, stiliziran u skladu s vizualnim identitetom cijele aplikacije, čime je u cjelini održana dosljednost pri oblikovanju sučelja.



Slika 2.11. *Basecamp project management* aplikacija – prikaz sučelja za pisanje projektne dokumentacije [27]

Osim inačice za klasične zaslone velikih dimenzija, razvijena je i posebna *Basecamp Web* aplikacija za mobilne uređaje. Važno je naglasiti kako to nije prirodna (eng. *native*) aplikacija za pojedino mobilno okruženje već *Web* aplikacija prilagođenog sučelja koja se izvršava izravno u *Web* okruženju pa stoga mobilni uređaj pri korištenju mora biti povezan na Internet.



Slika 2.12. Prikaz *Basecamp project management* aplikacije na različitim mobilnim uređajima [27]

Obzirom na sve prethodno navedene značajke, možemo zaključiti kako je *Basecamp* prava *Web 2.0* aplikacija jer dosljedno prati preporuke oblikovanja u *Web 2.0* okruženju te posjeduje sve karakteristike moderne *Web 2.0* aplikacije.

b) *StumbleUpon*

StumbleUpon je zanimljiva i inovativna aplikacija čija je osnovna ideja prikaz različitog sadržaja obzirom na unaprijed definirane korisnikove interese [28]. Servis je doživio veliki uspjeh zbog svoje zabavnosti, no i korisnosti jer često prikazuje zanimljive članke i fotografije koje bi sami teško pronašli posredstvom klasičnih pretraživača. Sučelje aplikacije je jednostavno i kvalitetno riješeno kako bi se korisnički doživljaj održao na visokom stupnju kroz cijelo vrijeme korištenja servisa.

Tok rada s aplikacijom podijeljen je u nekoliko jednostavnih cjelina, a to su registracija, prijava na sustav, odabir interesa i pretraživanje, odnosno izmjena relevantnih *Web* stranica na vrlo jednostavan način, pomoću tzv. *Stumble!* gumba. Zanimljivo je kako pri registraciji sustav omogućava povezivanje s *Facebook* profilom u svrhu automatske registracije, bez potrebe za ispunjavanjem formulara, što krajnjem korisniku dodatno olakšava cijeli postupak.

Sign in to StumbleUpon

Your Email or Username

Your Password

Remember me

Sign In



Sign in with Facebook

Don't have an account yet?

Join Now

Slika 2.13. *StumbleUpon* aplikacija – prikaz polja za unos sadržaja i mogućnosti registracije pomoću osobnog *Facebook* računa (eng. *account*) ^[28]

Formulari za unos sadržaja riješeni su jednostavno i funkcionalno, s prikladnim detaljima poput stiliziranog *submit* gumba i *onfocus* efekta koji se pojavljuje na trenutno označenom polju. Odabrani font je jasan i izražajan te se bojom i oblikom dobro uklapa u elemente sučelja. Pri neispravnom unosu, sustav obavještava korisnika prikladnom porukom ili promjenom boje problematičnih *input* polja.

Your Email or Username

Invalid username/email or password

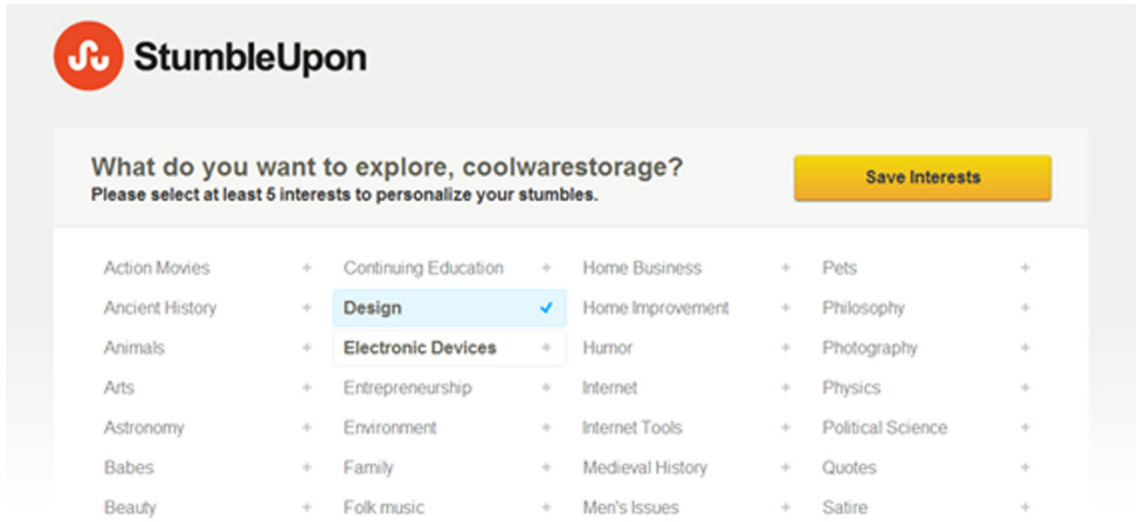


Your Password

Your Password

Slika 2.14. *StumbleUpon* aplikacija – obavještavanje korisnika o neuspješnom unosu ^[28]

Odabir interesa riješen je također jednostavno, pomoću *HTML* liste čiji se elementi označavaju jednostavnim klikom, a konačan odabir potvrđuje pomoću stiliziranog „*Save Interests*“ gumba.



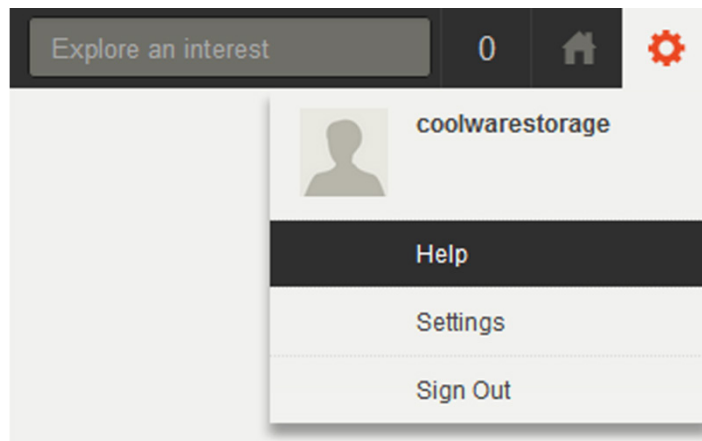
Slika 2.15. *StumbleUpon* aplikacija – prikaz sučelja za odabir korisnikovih interesa [28]

Sučelje glavnog dijela aplikacije izvedeno u obliku tanke trake i nalazi se pri vrhu prikaza. Sastoji se od tzv. *Stumble!* gumba za izmjenu prikazanog sadržaja sukladno korisnikovim interesima, funkcija za ocjenjivanje u svrhu optimizacije daljnjeg prikazivanja te funkcija za dijeljenje sadržaja s drugim korisnicima putem društvenih mreža poput *Facebook*, *Twitter* i *LinkedIn* servisa.



Slika 2.16. *StumbleUpon* aplikacija – prikaz sučelja za izmjenu, ocjenjivanje i dijeljenje (eng. *sharing*) prikazanog sadržaja [28]

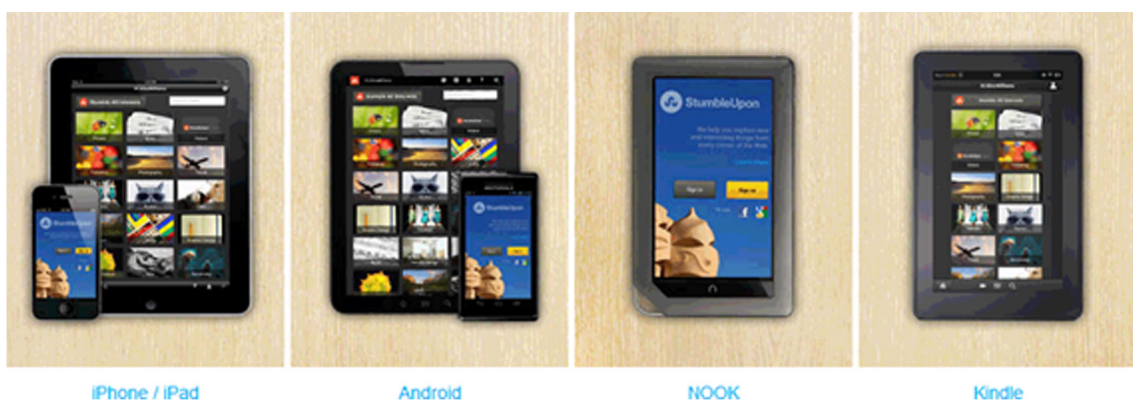
Osim navedenog, u krajnjem desnom kutu smještene su osnovne funkcije upravljanja profilom te vodič kroz aplikaciju, tzv. *Help*. Odjava iz sustava vrši se jednostavnim odabirom opcije *Sign Out*.



Slika 2.17. *StumbleUpon* aplikacija – prikaz osnovnog izbornika s opcijama „Pomoć“ (eng. *Help*), „Postavke“ (eng. *Settings*) i „Odjava“ (eng. *Sign Out*) [28]

StumbleUpon servis postoji i u inačici za mobilne uređaje, no za razliku od *Basecamp* aplikacije, *StumbleUpon* dolazi kao prirodna (eng. *native*) aplikacija za *iOS*, *Android*, *Amazon Kindle* i *Barnes & Noble NOOK* uređaje, što dodatno podiže kvalitetu same usluge. No, zbog prirode servisa, nužna je stalna veza s Internetom kako bi se predložene stranice mogle neometano izmjenjivati.

Obzirom na navedene značajke, *StumbleUpon* servis, jednako kao i *Basecamp* dosljedno prati preporuke oblikovanja u *Web 2.0* okruženju te predstavlja dobar primjer kvalitetno zamišljene i realizirane aplikacije.



Slika 2.18. Prikaz *StumbleUpon* aplikacije na različitim mobilnim uređajima [28]

U slijedećem poglavlju, praktično ćemo realizirati vlastito korisničko sučelje prateći viđene dobre prakse grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju te implementirajući moderne informatičke tehnologije.

3. RAZVOJ NAPREDNIH KORISNIČKIH SUČELJA U WEB 2.0 OKRUŽENJU

3.1. Razvojne tehnologije

Korisnička sučelja aplikacija u *Web 2.0* okruženju karakterizira niz naprednih mogućnosti koje se ostvaruju implementacijom modernih informatičkih tehnologija. Grafičko oblikovanje sučelja obično započinje planiranjem elemenata i funkcionalnosti te izradom skice prema kojoj se zatim oblikuju pojedini segmenti pomoću grafičkog uređivača (eng. *editor*), kao što je primjerice *Adobe Photoshop*. Slijedeći korak u izradi sučelja je pretvorba grafičkog rješenja u statičan *HTML* dokument, na način da se pojedinim *HTML* elementima pridružuje grafika izrađenog predloška, tzv. *slicing/coding* procesom. Nakon uspješne izrade *HTML* dokumenta, sučelje se na valjan način povezuje s funkcijama koje se izvršavaju na strani *servera*. U konkretnom primjeru, izradit ćemo napredan kontakt formular koji sa *server side* funkcijom za slanje *e-mail* poruke komunicira posredstvom *jQuery AJAX* metode. Tako ćemo se u najvećoj mjeri usmjeriti prema tehnologijama koje se izvršavaju na strani korisnika, dok ćemo *server side* programiranje pomoću *PHP* jezika spomenuti tek zbog potrebe realizacije potrebne funkcije.

Prethodno praktičnoj realizaciji sučelja kontakt formulara, pružit će se uvid u osnovne karakteristike potrebnih razvojnih tehnologija kako bi kvalitetno sagledali i shvatili sve potrebne korake izrade. Također, osvrnut ćemo se na tehnologije koje su već prisutne, no tek će u budućnosti dostići svoj puni potencijal.

3.1.1. HTML opisni jezik

HTML (eng. *HyperText Markup Language*) je opisni jezik koji služi za oblikovanje *Web* dokumenata. U osnovi, sastoji se od niza oznaka (eng. *tags*) kojima se opisuje struktura pojedine *Web* stranice, odnosno dokumenta [7, 30].

Izrađeni *HTML* dokument interpretira se u *Web* pregledniku korisnika (eng. *Web browser*) prema određenim standardima, ovisno o verziji *HTML* jezika koji se koristio pri izradi. Kvaliteno izrađen *HTML* dokument jednako će se interpretirati u svim modernim *Web* preglednicima (npr. *Microsoft Internet Explorer 8*, *Mozilla Firefox*, *Google Chrome*), odnosno vizualni doživljaj *Web* stranice biti će u skladu s izvornom idejom bez obzira koji preglednik koristimo za pregledavanje sadržaja [7, 23, 24, 30].

Slijedi prikaz izvornog koda jednostavnog *HTML* dokumenta:

```
<html>
  <head>
    <title>Jednostavna stranica</title>
  </head>
  <body>
    <p>Primjer teksta <a href="marko.html">Primjer linka</a> </p>
  </body>
</html>
```

Izvorni kod 3.1. Struktura jednostavnog *HTML* dokumenta

Kompletan sadržaj *HTML* dokumenta, odnosno pojedine *Web* stranice nalazi se unutar osnovnog *<html>* taga koji *interpreteru Web* preglednika govori kako se unutar njega nalazi *HTML* kod stranice. Sadržaj unutar početnog (*<html>*) i završnog (*</html>*) *HTML* taga podijeljen je na dvije osnovne cjeline, zaglavlje i tijelo stranice. Zaglavlje stranice definirano je *<head>* tagom, a tijelo *<body>* tagom. U zaglavlje, odnosno unutar početnog i završnog *<head>* taga smještaju se informacije o stranici u obliku *<title>* i *<meta>* tagova te poveznice prema vanjskim *CSS* i *JavaScript* datotekama.

Osnovni sadržaj *Web* stranice, odnosno sadržaj koji će se korisniku prikazati na zaslonu pri dovršetku interpretacije na strani *Web* preglednika, smješta se unutar početnog i završnog *<body>* taga. Prikazani primjer jednostavnog *HTML* dokumenta u tijelu stranice sadrži tag *<p>* koji označava paragraf teksta te poveznicu, odnosno *link* koji se označava *<a>* tagom. Poveznica povezuje trenutni *HTML* dokument s drugim dokumentom, u prikazanom primjeru s *marko.html* stranicom.

3.1.2. CSS opisni jezik

CSS (eng. *Cascading Style Sheets*) je opisni jezik koji se koristi za grafičko stiliziranje *HTML* dokumenata [7, 30]. Osnovna ideja CSS jezika je razdvajanje *HTML* strukture *Web* stranice i pripadajućeg sadržaja od vizualnih, stilskih elemenata. Povezivanjem različitih CSS datoteka čije su definicije na valjan način pridružene određenim *HTML tagovima* unutar dokumenta, izgled *Web* stranice (margine, raspored elemenata, boje, stilovi teksta) može se u potpunosti promijeniti. Stoga, zbog iznimne prilagodljivosti i jednostavnosti, CSS je danas opće prihvaćeni standard pri izradi grafičkih sučelja *Web* stranica [30].

Slijede primjeri stilizacije pomoću CSS jezika.

CSS stilovi - u prikazanom primjeru pretpostavlja se kako su stilovi upisani u pripadajuću *CSS style sheet* datoteku:

Primjer 1:

```
p {  
    font-size: 12pt;  
    font-style: italic;  
    color: red;  
}
```

Izvorni kod 3.2. Primjer stiliziranja izravnim vezanjem na *HTML tag*

Primjer 2:

```
.paragrafTeksta {  
    font-size: 18pt;  
    font-weight: bold;  
    color: #000000;  
}
```

Izvorni kod 3.3. Primjer stiliziranja posredstvom *CSS klase*

Primjer 3:

```
#plaviParagraf {  
    font-size: 21pt;  
    font-weight: bold;  
    color: #000000;  
}
```

Izvorni kod 3.4. Primjer stiliziranja posredstvom *CSS identifikatora*

Povezivanje prikazanih definicija s pripadajućim elementima u *HTML* datoteci:

Primjer 1:

Naknadno povezivanje nije potrebno jer se izvršava automatski zbog izravnog referenciranja na `<p>` tag pripadajućeg *HTML* dokumenta.

Primjer 2:

```
| <p class="paragrafTeksta">Primjer teksta</p>
```

Izvorni kod 3.5. Primjer povezivanja *HTML* taga i pripadajuće CSS klase

Primjer 3:

```
| <p id="plaviParagraf">Primjer teksta</p>
```

Izvorni kod 3.6. Primjer povezivanja *HTML* taga i pripadajućeg CSS identifikatora

U prvom primjeru povezivanje željenih definicija s pripadajućim *HTML* tagom provedeno je automatski, referenciranjem stila na `<p>` tag unutar *HTML* dokumenta. Drugi i treći primjer prikazuju povezivanje pomoću CSS klase i CSS identifikatora. Razlika između ta dva načina je u dozvoljenom broju njihovog ponavljanja unutar *HTML* dokumenta. Naime, identifikatori (eng. *id*) se smiju iskoristiti samo jednom unutar pojedinog *HTML* dokumenta, dok se klase (eng. *class*) mogu upotrijebiti neograničeno mnogo puta.

CSS kod koji se nalazi u vanjskoj CSS datoteci povezuje se s *HTML* dokumentom slijedećim *HTML* kodom koji se dodaje unutar zaglavlja *Web* stranice, odnosno između početnog i završnog `<head>` taga:

```
| <link rel="stylesheet" type="text/css" href="stilovi.css"
  media="screen" />
```

Izvorni kod 3.7. Primjer definiranja vanjske CSS datoteke pomoću *HTML* link taga

U prikazanoj liniji koda, *href* atribut definira putanju i naziv CSS datoteke, a *media* određuje vrstu medija za koji su stilovi prilagođeni (može biti i npr. *print*).

Osim na prikazani način, CSS stilovi se mogu dodati i izravno u željeni *HTML* tag, tzv. *inline* stiliziranjem:

```
<p style="font-size: 18pt; font-style: italic; color: red;">Primjer  
teksta</p>
```

Izvorni kod 3.8. Primjer *inline* stiliziranja odabranog *HTML* taga

3.1.3 JavaScript programski jezik

JavaScript je skriptni programski jezik koji se izvršava na strani klijenta, odnosno u *Web* pregledniku korisnika [6, 13]. Razvila ga je tvrtka *Netscape* čiji je *Web* preglednik *Netscape Navigator* devedesetih godina prošlog stoljeća bio dominantan na tržištu, s udjelom od čak 90%. *JavaScript* je sintaksom vrlo sličan jeziku *Java*, no *Java* je za razliku od *JavaScript*-a objektno orijentirani jezik.

U početku *JavaScript* se uglavnom koristio za dekorativne sitnice na *Web* stranicama, poput različitih satova, stiliziranih kursora i navigacijskih traka. No, napretkom tehnologije stvorio se niz zanimljivih mogućnosti poput izravne komunikacije *JavaScript* programa s poslužiteljskim tehnologijama, što je omogućilo asinkrono pokretanje funkcija na poslužitelju (eng. *server*) i dinamičko prihvaćanje dobivenih rezultata. Navedena tehnologija naziva se *AJAX*, odnosno eng. *Asynchronous JavaScript and XML* [5, 13, 18].

Jednako tako, važno je spomenuti niz *JavaScript* radnih okvira (eng. *framework*) koji su se pojavili, a uvelike olakšavaju izvršavanje složenih funkcija koje bi samostalno bilo teško realizirati. Naime, radni okvir, odnosno *framework*, sadrži niz unaprijed definiranih mogućnosti koje se jednostavno pozivaju i slažu u željeni proizvod [5, 13], što je osobito korisno pri izgradnji složenih sustava jer skraćuje vrijeme, trud i troškove izrade.

U praktičnom primjeru realizacije sučelja kontakt formulara, koristit ćemo *jQuery* radni okvir u svrhu provjere unosa te asinkrone komunikacije sa *server side* funkcijom za slanje *e-mail* poruke.

3.1.4. PHP programski jezik

PHP (eng. *PHP: Hypertext Preprocessor*) je moderan programski jezik prvenstveno namijenjen izvršavanju u *Web* okruženju, na strani poslužitelja (eng. *server side*) ^[31]. Razvio ga je Rasmus Lerdorf, s osnovnom zamisli da pomoću njega na svojoj osobnoj stranici broji posjete. U početku, to je bio skup skripti napisanih u *Perl* jeziku, no poslije kada je došlo do potrebe za više funkcija, napisao je potpuno novu verziju u programskom jeziku C. U to vrijeme, *PHP* je koristilo oko 50 000 *Web* stranica, što je tek oko 1% ukupnog broja *Web* stranica na Internetu u to vrijeme, dok je danas *PHP* najpopularniji i najčešće korišteni programski jezik na *Web-u* ^[31]. Tvorac jezika odlučio je *PHP* ponuditi svijetu kao slobodan softver, odnosno kao tehnologiju otvorenog koda, što je uvelike pomoglo njegovu razvoju i postizanju trenutne razine popularnosti i raširenosti na *Web-u*.

PHP kod se može pisati u obliku skripte te po objektno orijentiranim principima, što ga čini posebno moćnim jezikom. Također, važno je spomenuti kako je trenutno besplatno dostupan niz kvalitetnih *PHP* radnih okvira (eng. *frameworks*) koji omogućuju brzu izradu složenih *Web* aplikacija po *Web 2.0* principima.

Slijedi primjer *PHP* koda koji unutar otvorenog i zatvorenog `<p>` taga ispisuje određeni tekst ovisno da li varijabla `$broj` sadrži paran ili neparan broj:

```
<p>
<?php
    $broj=2;
    if($broj%2==0) {
        echo 'Broj '$broj.' je paran';
    }

```

```

        else {
            echo 'Broj ' . $broj . ' nije paran';
        }
    ?>
</p>

```

Izvorni kod 3.9. Primjer *PHP* koda koji unutar *HTML* paragrafa ispisuje određeni tekst ovisno o parnosti vrijednosti spremljene u varijablu \$broj

PHP kod ograđen je početnim `<?php` i završnim `?>` *PHP* tagom. Varijable se u *PHP* jeziku označavaju s dolar znakom pa se stoga naša varijabla naziva `broj` piše kao `$broj`. Ispitivanje parnosti vrši se pomoću *if* provjere koja dijeli broj spremljen u varijablu `$broj` s brojem dva te uspoređuje dobiveni ostatak dijeljenja s nulom. Ako ostatka nema, broj je djeljiv s brojem dva, odnosno paran je, a ako nije, program prelazi na *else* uvijet. Ukoliko je broj paran, naredba *echo* ispisuje poruku da je broj paran, a ako nije, ispisuje se poruka unutar *else* uvjeta.

U konkretno prikazanom primjeru, ako pregledavamo kod generirane *Web* stranice, vidimo slijedeće:

```
<p>Broj 2 je paran</p>
```

Izvorni kod 3.10. Prikaz konačnog ispisa u *Web* pregledniku, nakon izvršavanja **Izvornog koda 3.9.**

3.1.5. Tehnologije budućnosti

Tehnologijama budućnosti obično se prozivaju nove verzije *HTML* i *CSS* opisnih jezika, odnosno *HTML5* i *CSS3* [7, 23]. Navedeni jezici implementiraju brojne napredne mogućnosti koje smo do sada mogli samo imitirati pomoću grafičkih elemenata i prikladnog *JavaScript* koda. Primjerice, zaobljeni rubovi, refleksije i razne sjene obično se izrađuju u grafičkom editoru, a zatim u svojstvu pozadine određenog *HTML* elementa prenose u strukturu *Web* stranice. Novi *CSS* jezik dolazi s implementiranom podrškom za sve navedeno te

omogućuje postizanje jednakih efekata izravnom modifikacijom CSS definicija, što je posebno korisno i praktično jer više nemamo potrebu korištenja nekoliko različitih programa i metoda [7]. Također, sve vizualne izmjene na tako realiziranoj *Web* stranici brzo se izvršavaju jednostavnom izmjenom CSS parametara. Novi tagovi *HTML5* jezika također imaju velik potencijal, a zanimljivo je spomenuti i kako *YouTube* servis već nudi mogućnost korištenja *HTML5 video player-a*, umjesto dosadašnje *Flash* verzije.

Službena specifikacija *HTML5* i *CSS3* jezika još nije dovršena od strane *W3C* (eng. *World Wide Web Consortium*) organizacije pa trenutno dostupne metode navedenih tehnologija podržavaju tek noviji *Web* preglednici među kojima su vodeći *Mozilla Firefox* i *Google Chrome* [7]. Korisno je spomenuti kako su trenutne mogućnosti radnih verzija *HTML5* i *CSS3* jezika realizirane na način da se izvršavaju u preglednicima koji ih podržavaju, dok se u ostalima zanemaruju bez eventualnih kolizija s postojećim elementima. Navedeni princip je odlično vidljiv u primjeni zaobljenih rubova produciranih pomoću *CSS3* jezika. Naime, ukoliko tako stiliziranu stranicu pregledavamo pomoću preglednika koji podržava navedenu opciju, vidjet ćemo element sa zaobljenim rubom, dok će ostali preglednici prikazati isti element bez zaobljenih rubova, no sa zadržanom temeljnom funkcionalnošću. Stoga, zbog rastuće *HTML5* i *CSS3* podrške u modernim preglednicima te rasprostranjenosti *Mozilla Firefox* i *Google Chrome* rješenja, uporaba *CSS3* stilova postaje sve češća praksa.

3.2. Praktičan razvoj naprednog korisničkog sučelja u *Web 2.0* okruženju

Praktičan dio ovog diplomskog rada demonstrirat će izradu funkcionalnog sučelja za slanje poruka, tzv. kontakt formulara (eng. *contact form*), prema pravilima i standardima *Web 2.0* okruženja. Kontakt formulari se često koriste na statičnim *Web* stranicama u svrhu ostvarenja komunikacije s posjetiteljima, poput primjerice rezervacije termina, upita u vezi proizvoda ili određene vrste

anketiranja o uslugama. Osim klasičnih, statičnih *Web* stranica, kontakt formulari se u određenom obliku pojavljuju i u sklopu velikih *Web* aplikacija. Primjerice, *Claroline* aplikacija za e-učenje (eng. *e-learning*) ima ugrađen sustav komunikacije među korisnicima koji poruke dostavlja u vlastitom *on-line* okruženju, a kopije prosljeđuje izravno na registrirane *e-mail* adrese.

Sučelje za slanje poruka koje ćemo izraditi u svrhu demonstracije, dosljedno će pratiti preporuke grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju. Posebne efekte, viđene pri analizi *Basecamp* i *Stumbleupon* servisa, ostvarit ćemo korištenjem trenutno podržanih *CSS3* tehnika te određenim funkcijama *jQuery* radnog okvira, dok će specifični grafički elementi biti izrađeni pomoću *Adobe Photoshop* grafičkog uređivača (eng. *editor*). Struktura formulara bazirat će se na *XHTML 1.0 Strict* jeziku kako bi se zadržala kompatibilnost s većinom trenutno dostupnih *Web* preglednika.

Izrađeni kontakt formular zaista će biti funkcionalan, odnosno slati će poruke na unaprijed definiranu *e-mail* adresu. Navedenu funkcionalnost možemo ostvariti jedino korištenjem *server side* tehnologija pa ćemo stoga izraditi jednostavnu *PHP* skriptu koja će prihvatiti upisane podatke i *mail()* funkcijom poslati ih na valjanu destinaciju. Kako bi se tijekom cijelog vremena korištenja sučelja održao visok stupanj korisničkog doživljaja (eng. *user experience*) podaci će se *server side* datoteci prosljeđivati *AJAX* metodom pomoću valjane funkcije *jQuery* radnog okvira.

Prikaz konačno izrađenog sučelja prilagođavat će se karakteristikama korisnikovog uređaja pomoću *Modernizr JavaScript* biblioteke (eng. *library*) koja nudi mogućnost detekcije tzv. *touch* značajki (eng. *features*) *Web* preglednika, što je osnovna karakteristika mobilnih uređaja.




Prvi korak pri izradi sučelja je planiranje i grafičko oblikovanje pojedinih elemenata te definiranje svrhe, cilja i konačnih funkcionalnosti.

Slijedi prikaz planiranih karakteristika sučelja:

Tablica 3.1. Prikaz planiranih grafičkih i funkcionalnih karakteristika sučelja kontakt formulara

Naziv sučelja / aplikacije	Kontakt formular
Svrha	Slanje <i>e-mail</i> poruka na unaprijed određenu adresu, posredstvom <i>Web</i> sučelja
Opće karakteristike	<ul style="list-style-type: none"> • Jednostavnost • Vizualna atraktivnost • Visok stupanj korisničkog doživljaja (eng. <i>UX</i>) • Dosljednost u korištenju grafičkih elemenata i posebnih funkcionalnosti • Kompatibilnost s modernim <i>Web</i> preglednicima • Prilagođavanje prikaza karakteristikama zaslona mobilnih uređaja
Tehničke karakteristike	<ul style="list-style-type: none"> • Oblikovanje bez uporabe tablica (eng. <i>tableless design</i>) • Izvorni kod (eng. <i>source code</i>) baziran na <i>XHTML 1.1 Strict</i> i <i>CSS 2.1</i> jeziku • Uporaba posebnih stilskih <i>CSS3</i> definicija (zaobljeni rubovi, sjene) • Uporaba <i>jQuery v 1.7.2</i> radnog okvira (eng. <i>framework</i>) i <i>Modernizr</i> biblioteke (eng. <i>library</i>) • Provjera korisnikovih unosa (eng. <i>input validation</i>) pomoću vlastitih <i>JavaScript</i> funkcija i funkcija <i>jQuery</i> radnog okvira • Prosljeđivanje upisanih vrijednosti formulara <i>server side</i> datoteci posredstvom <i>jQuery</i>

	<p><i>AJAX</i> funkcije, <i>POST</i> metodom</p> <ul style="list-style-type: none"> • Detekcija <i>touch</i> značajki (eng. <i>features</i>) korisnikovog <i>Web</i> preglednika u svrhu prilagodbe širine pojedinih elemenata sučelja
<p>Raspored elemenata sučelja</p>	<ul style="list-style-type: none"> • Zaglavlje – sadrži naziv sučelja / aplikacije • Sadržaj: <ul style="list-style-type: none"> • Formular – sadrži slijedeće elemente: <ul style="list-style-type: none"> • Paragraf: <ul style="list-style-type: none"> • Naziv polja (<i>label</i>) • Polje za unos (<i>input</i> ili <i>textarea</i>) • Obavijest (<i>span</i>) • Paragraf: <ul style="list-style-type: none"> • Gumb za konačnu predaju podataka (<i>input type="submit"</i>)
<p>Nazivi polja za unos, kratak opis i vrsta pripadajućeg polja</p>	<ul style="list-style-type: none"> • Ime – <i>input type="text"</i> <ul style="list-style-type: none"> • Korisnikovo ime, ime i prezime ili nadimak • E-mail – <i>input type="text"</i> <ul style="list-style-type: none"> • Korisnikova e-mail adresa • Telefon – <i>input type="text"</i> <ul style="list-style-type: none"> • Korisnikov broj telefona • Adresa – <i>input type="text"</i> <ul style="list-style-type: none"> • Korisnikova fizička adresa • Web site – <i>input type="text"</i> <ul style="list-style-type: none"> • Adresa korisnikovog <i>Web site-a</i> • Poruka – <i>textarea</i> <ul style="list-style-type: none"> • Korisnikova poruka

	<ul style="list-style-type: none"> Pošalji – <i>input type="submit"</i> <ul style="list-style-type: none"> Gumb (eng. <i>button</i>) za prosljeđivanje unesenih podataka <i>server side</i> datoteci
Obavezna polja	<ul style="list-style-type: none"> Ime E-mail Poruka
Izgled obaveznog polja i pripadajuće poruke u početnom stanju	
Izgled obaveznog polja i pripadajuće poruke pri neuspješno provedenoj provjeri unosa	
Izgled obaveznog polja i pripadajuće poruke pri uspješno provedenoj provjeri unosa	
Parametri provjere unosa	<ul style="list-style-type: none"> Prazna polja Neispravna <i>e-mail</i> adresa
Parametri ovisni o vrsti uređaja	<ul style="list-style-type: none"> Širina (eng. <i>width</i>) osnovnog <i>div</i> kontejnera klase sadržaj Širina (eng. <i>width</i>) <i>submit</i> gumba (eng. <i>submit button</i>) kontakt formulara

Slijedi idejno grafičko rješenje sučelja izrađeno pomoću *Adobe Photoshop* grafičkog uređivača (eng. *editor*):

Kontakt formular

Ime:

* Obavezno!

E-mail:

Neispravan format email adrese!

Telefon:

Adresa:

Web site:

Poruka:

* Obavezno!

Pošalji

Slika 3.1. Prikaz grafičkog idejnog rješenja korisničkog sučelja kontakt formulara, izrađeno pomoću *Adobe Photoshop* grafičkog uređivača (eng. *editor*)

Prikazano statično grafičko rješenje potrebno je pretvoriti u funkcionalan *HTML* dokument. Stoga, slijedeći korak je izdvajanje pozadina i analiza pojedinih segmenata sučelja u svrhu definiranja osnovne *HTML* strukture i pripadajućih *CSS* definicija.

Izradu funkcionalnog formulara započinjemo zadavanjem osnovne strukture *HTML* dokumenta, bez pripadajućih stilova.

Kontakt formular

Ime: * Obavezno!

E-mail: * Obavezno!

Telefon:

Adresa:

Web site:

Poruka: // * Obavezno!

Slika 3.2. Prikaz osnovne strukture sučelja kontakt formulara, bez apliciranih *CSS* stilskih definicija

Slijedi *HTML* kod prikazane *HTML* stranice:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="hr" lang="hr">
  <head>
    <title>Kontakt formular</title>
  </head>
  <body>
    <div class="sadrzaj">
      <div class="zaglavlje">
        <h1>Kontakt formular</h1>
      </div>
```

```

        <form id="kontakt-formular" name="kontakt-formular"
action="">
        <p>
            <label for="ime"><strong>Ime:</strong></label>
            <input type="text" id="ime" name="ime" />
            <span class="obavijest"><span>*</span>
Obavezno!</span>
        </p>
        <p>
            <label for="email"><strong>E-mail:</strong></label>
            <input type="text" id="email" name="email" />
            <span class="obavijest"><span class="crveno">*
Obavezno!</span></span>
        </p>
        <p>
            <label><strong>Telefon:</strong></label>
            <input type="text" id="telefon" name="telefon" />
        </p>
        <p>
            <label><strong>Adresa:</strong></label>
            <input type="text" id="adresa" name="adresa" />
        </p>
        <p>
            <label><strong>Web site:</strong></label>
            <input type="text" id="website" name="website" />
        </p>
        <p>
            <label
for="poruka"><strong>Poruka:</strong></label>
            <textarea name="poruka" id="poruka"
class="normalno obavezno"></textarea>
            <span class="obavijest"><span
class="crveno">*</span> Obavezno!</span>
        </p>
        <p>
            <input type="submit" id="posalji" name="posalji" />
        </p>
    </form>
</div>
</body>
</html>

```

Izvorni kod 3.11. Prikaz *HTML* koda osnovne strukture sučelja kontakt formulara

Iz priloženog izvornog koda možemo vidjeti kako je cijeli sadržaj stranice smješten u `<div class="sadrzaj"></div>` kontejner kako bi mogli jednostavno ugraditi sustav automatske prilagodbe sadržaja veličini zaslona korisnikovog uređaja. Kontakt formular smo formirali pomoću prethodno definiranih `<p>`,

`<label>`, `<input>` i `` elementa te određena polja označili prikladnim `id=""` i `name=""` atributima u svrhu kasnije implementacije potrebnih funkcionalnosti.

Slijedeći korak je određivanje CSS stilova u skladu s grafičkim idejnim rješenjem. Postupak oblikovanja sučelja započet ćemo oblikovanjem zaglavlja (eng. *header*) stranice. Prema predlošku, zaglavlje treba imati oblik prikazan slijedećom slikom:



Slika 3.3. Prikaz grafičkog idejnog rješenja zaglavlja stranice

Struktura zaglavlja definirana je slijedećim HTML kodom:

```
<div class="zaglavlje">
  <h1>Kontakt formular</h1>
</div>
```

Izvorni kod 3.12. Prikaz *HTML* koda zaglavlja stranice

U svrhu postizanja željenog grafičkog oblika, klasi zaglavlje (`class="zaglavlje"`) potrebno je dodijeliti slijedeći CSS kod:

```
.zaglavlje {
  padding: 15px;
  line-height: 45px;
  text-shadow: 1px 1px 1px white;
  background: #66b723;
  border-top-left-radius: 10px;
  border-top-right-radius: 10px;
  height: 45px;
}

h1 {
  text-shadow: 1px 1px 1px #2A2A2A;
  color: #FFF;
}
```

Izvorni kod 3.13. Prikaz CSS definicija zaglavlja stranice

Prikazani kod upisujemo u *style.css* datoteku koju definiramo u `<head></head>` segmentu stranice na slijedeći način:

```
<link rel="stylesheet" media="screen" href="style.css" />
```

Izvorni kod 3.14. Definiranje vanjske *style.css* datoteke pomoću *HTML link taga*

Slijedeći korak je definiranje CSS stilova za pojedine elemente formulara:

```
form {
    background: url('img/top-content-bg.png') repeat-x top;
    padding: 15px 15px 0 15px;
}

p {
    padding-bottom: 15px;
    color: #777;
}

label {
    padding-bottom: 10px;
    display: block;
    color: #614b30;
}

input[type="text"], textarea {
    width: 100%;
    padding: 10px;
    -moz-border-radius: 3px;
    border-radius: 3px;
}

input[type="submit"] {
    width: 100%;
    height: 40px;
    color: #FFF;
    font-weight: bold;
    border: none;
    cursor: hand;
    background: url('img/button-bg.png') repeat-x top;
    -moz-border-radius: 5px;
    border-radius: 5px;
}

input[type="submit"]:hover {
    background: #483621;
}
```

```

textarea {
    height: 100px;
}

.obavijest {
    display: block;
    padding: 5px 0 5px 0;
    font-size: 8pt;
}

.crveno {
    color: #C61B1B;
}

.normalno {
    border: solid 1px #999;
    background: url('img/input-bg.png') repeat-x top;
    color: #555;
}

.greska {
    border: 1px solid #C00;
    background: #FCF0EF url('img/error-bg.png') repeat-x top;
    color: #C00;
}

.uspjeh {
    border: 1px solid #9DB832;
    background: #F3F8E0 url('img/uspjeh-bg.png') repeat-x top;
    color: #9DB832;
}

.poruka {
    width: auto;
    height: 45px;
    line-height: 45px;
    color: #417814;
    margin-top: 15px;
    padding-left: 15px;
    -moz-border-radius: 5px;
    border-radius: 5px;
}

```

Izvorni kod 3.15. Definiranje CSS stilova pojedinih elemenata kontakt formulara

Osim prikazanog koda, u CSS datoteku potrebno je dodati i definicije koje formatiraju stranicu, definiraju željeni font te poništavaju zadane (eng. *default*) postavke *HTML* elemenata.

```

* {
  margin: 0px;
  padding: 0px;
  font-family: Arial, Helvetica, sans-serif;
}

body {
  font-family: Arial, Helvetica, sans-serif;
  font-size: 10pt;
}

.sadržaj {
  margin: 0 auto;
  padding: 15px 15px 0 15px;
}

```

Izvorni kod 3.16. Definiranje CSS stilova za poništavanje zadanih (eng. *default*) postavki *HTML* elemenata te formatiranje osnovnog *div* kontejnera klase *sadržaj*

Klasa *normalno* (**.normalno**) definira početno stanje *input* polja formulara, pa ju je potrebno povezati s pripadajućim poljima na slijedeći način:

```
<input type="vrsta" name="ime" id="id" class="normalno" />
```

Izvorni kod 3.17. Povezivanje *input* polja formulara sa željenom klasom naziva „normalno“

Nakon apliciranja prikazanih CSS definicija, stranica s kontakt formularom poprima slijedeći izgled:

Kontakt formular

Ime:

* Obavezno!

E-mail:

* Obavezno!

Telefon:

Adresa:

Web site:

Poruka:

* Obavezno!

Pošalji

Slika 3.4. Prikaz *HTML* dokumenta s pripadajućim kontakt formularom nakon apliciranja željenih *CSS* definicija

Slijedeći korak je izrada sustava provjere upisanih vrijednosti, što ćemo izvesti izradom vlastitih *JavaScript* funkcija, uz pomoć *jQuery* radnog okvira (eng. *framework*).

Kako bi mogli koristiti mogućnosti *jQuery* radnog okvira, u `<head></head>` segmentu stranice moramo definirati putanju (eng. *path*) do *jquery.js* datoteke na slijedeći način:

```
<script src="js/jquery-1.7.2.min.js" type="text/javascript"></script>
```

Izvorni kod 3.18. Definiranje putanje *jquery.js* datoteke, u *head* segmentu *HTML* stranice

Navedenu datoteku preuzeli smo sa službenog *Web site-a jQuery* projekta (<http://code.jquery.com/jquery-1.7.2.js>) te u svrhu dobre organizacije potrebnih datoteka smjestili ju u *js* direktorij.

Slijedeći korak je izrada *kontakt-formular.js* datoteke u koju ćemo smjestiti cjelokupan *JavaScript* kod potreban za provjeru unosa i kasniju predaju podataka *server side* datoteci. Putanju do navedene datoteke također je potrebno upisati u `<head></head>` segment stranice na prikazani način:

```
<script src="js/kontakt-formular.js" type="text/javascript" charset="utf-8"></script>
```

Izvorni kod 3.19. Definiranje putanje *kontakt-formular.js* datoteke, u *head* segmentu *HTML* stranice

Osim klasičnih parametara, u prikazani kod uključen je i atribut `charset="utf-8"` kako bi se posebni znakovi upisani u datoteku prikazali na valjani način. Navedeno se posebno odnosi na dijakritičke znakove – č, ć, š, đ, ž. U svrhu valjanog prikaza dijakritičkih znakova u svim segmentima stranice, *charset* je potrebno definirati i kao *meta* (hrv. opisni) *tag* na slijedeći način:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

Izvorni kod 3.20. Definiranje *UTF-8 charseta* posredstvom opisnog (eng. *meta*) *taga*

Meta tagovi se definiraju u `<head></head>` segmentu *HTML* stranice.

Slijedi prikaz *JavaScript* koda potrebnog za provjeru unosa:

```
$(document).ready(function() {

    function validateEmail(email) {
        var a = document.getElementById(email).value;
        var filter = /^[a-zA-Z0-9]+[a-zA-Z0-9_.-]+[a-zA-Z0-9_-]+@[a-zA-Z0-9]+[a-zA-Z0-9.-]+[a-zA-Z0-9]+.[a-z]{2,4}$/;
        if(filter.test(a)) {
            return true;
        }
        else {
            return false;
        }
    }

    var obavijest = $('#obavijest');

    obavijest.html("<span class='crveno'>*</span> Obavezno!");

    $('#posalji').click(function() {

        var hasError = false;

        obavijest.html("<span class='crveno'>*</span> Obavezno!");

        var ime = $("input#ime");
        if ((ime.val() == "") || (ime.val() == "Ime")) {
            ime.removeClass('uspjeh').addClass('greska').focus();
            isFocus=1;
            hasError = true;
            ime.next().html("<span class='crveno'>Polje je obavezno!</span>");
            return false;
        }
        else {
            ime.addClass('uspjeh').focus();
        }

        var email = $("input#email");
        if ((email.val() == "") || (email.val() == "Email")) {
            email.removeClass('uspjeh').addClass('greska').focus();
            isFocus=1;
            hasError = true;
            email.next().html("<span class='crveno'>Polje je obavezno!</span>");
            return false;
        }
        else {
```

```

        email.addClass('uspjeh').focus();
    }

    if(!validateEmail('email')) {
        email.removeClass('uspjeh').addClass('greska').focus();
        isFocus=1;
        email.next().html("<span class='crveno'>Neispravan format
email adrese!</span>").fadeIn(1500);
        hasError = true;
        return false;
    }

    var poruka = $("#textarea#poruka");
    if ((poruka.val() == "") || (poruka.val() == "Poruka")) {
        poruka.removeClass('uspjeh').addClass('greska').focus();
        isFocus=1;
        hasError = true;
        poruka.next().html("<span class='crveno'>Polje je
obavezno!</span>");
        return false;
    } else {
        poruka.addClass('uspjeh').focus();
    }

    return false;
});
});

```

Izvorni kod 3.21. Prikaz *JavaScript* koda potrebnog za provjeru unosa (eng. *input validation*)

Apliciranjem prikazanog *JavaScript* koda, kontakt formular je primio željene funkcije provjere unosa.

U početnom stanju, sva obavezna polja su slijedećeg izgleda:

* Obavezno!

Slika 3.5. Prikaz *input type="text"* polja formulara u početnom stanju

Ukoliko jedno ili više obaveznih polja pri predaji podataka ne sadrži potreban unos, odnosno nije ispunjeno, sustav aplicira klasu crveno (*.crveno*) problematičnom polju i ispisuje prikladnu obavijest.

Polje je obavezno!

Slika 3.6. Prikaz `input type="text"` polja formulara nakon detekcije pogreške

Polje *e-mail* podliježe dodatnoj provjeri valjanosti upisane *e-mail* adrese. Naime, sustav ne dopušta predaju podataka sve dok *e-mail* adresa ne sadrži sve potrebne karakteristične parametre, kao što su naziv primaoca, *monkey* znak (@), točka (.), naziv domene i vršna domena, odnosno upisana adresa mora biti oblika primaoc@naziv-domene.vršna-domena (npr. marko.cacic@gmail.com).

Neispravan format email adrese!

Slika 3.7. Prikaz `input type="text"` *e-mail* polja formulara nakon detekcije pogreške

Nakon što se prepoznate greške isprave, pri ponovnoj predaji podataka, označenim poljima se uklanja klasa crveno (`.crveno`) i aplicira klasa zeleno (`.zeleno`) pa stoga poprimaju slijedeći izgled:

* Obavezno!

Slika 3.8. Prikaz `input type="text"` polja formulara nakon ispravke pogreške

Slijedeći korak je izrada sustava za predaju unesenih podataka *server side* datoteci i konačno slanje na odabranu *e-mail* adresu.

Navedenu funkcionalnost ćemo ostvariti korištenjem *jQuery AJAX* funkcije, na slijedeći način:

```
if(!hasError) {  
    var dataString = 'ime='+ $('#kontakt-formular #ime').val() +  
'&email=' + $('#kontakt-formular #email').val() + '&telefon=' +  
$('#kontakt-formular #telefon').val() + '&adresa=' + $('#kontakt-  
formular #adresa').val() + '&website=' + $('#kontakt-formular
```

```

#website').val() + '&poruka=' + $('#kontakt-formular #poruka').val() +
'&website=';

$.ajaxSetup ({
    cache: false
});

var url = 'php/mail.php'
$.ajax({
    type: 'POST',
    url: url,
    data: dataString,
    success: function() {
        $('#div.poruka-uspjeh').fadeIn();
        $('#kontakt-formular').slideUp("slow", function() {
            $(this).before('<div class="poruka
uspjeh"><strong>Hvala!</strong> Poruka je uspješno poslana!</div>');
        });
    },
});
}

```

Izvorni kod 3.22. Prikaz *JavaScript* koda za predaju unesenih podataka server side datoteci

Prikazani kod funkcionira tako što provjerava postoji li greška u unosu posredstvom varijable *hasError* te ukoliko ne postoji, prilagođava prikupljene podatke i *POST* metodom ih prosljeđuje *server side* datoteci naziva *mail.php*, koju ćemo pri izradi smijesiti u *php* direktorij. Ukoliko se komunikacija s *mail.php* datotekom uspješno provede, formular nestaje popraćen animacijom sporog povlačenja (*.slideUp("slow")*) te se ispisuje poruka o uspjehu.

Server side datoteka izrađena je pomoću *PHP* programskog jezika te sadrži slijedeći kod:

```

<?php
$ime = (isset($_POST['ime'])) ? $_POST['ime'] : '';
$email = (isset($_POST['email'])) ? $_POST['email'] : '';
$telefon = (isset($_POST['telefon'])) ? $_POST['telefon'] : '';
$adresa = (isset($_POST['adresa'])) ? $_POST['adresa'] : '';
$website = (isset($_POST['website'])) ? $_POST['website'] : '';
$poruka = (isset($_POST['poruka'])) ? $_POST['poruka'] : '';
$prima = 'marko.cacic@gmail.com';
$tema = 'Kontakt formular';

$poruka_slanje = "Detalji poruke:\r\n\n";
$poruka_slanje.= "Ime: ".$ime."\r\n";

```

```

$poruka_slanje.= "E-mail: ".$email."\r\n";
$poruka_slanje.= "Telefon: ".$telefon."\r\n";
$poruka_slanje.= "Adresa: ".$adresa."\r\n";
$poruka_slanje.= "Website: ".$website."\r\n\r\n";
$poruka_slanje.= "Poruka: \n".$poruka."\r\n\r\n";
$poruka_slanje.= "Lijep pozdrav,\r\n";
$poruka_slanje.= $ime;
$zaglavlje = 'MIME-Version: 1.0' . "\r\n";
$zaglavlje.= 'Content-type: text/plain; charset=UTF-8' . "\r\n";

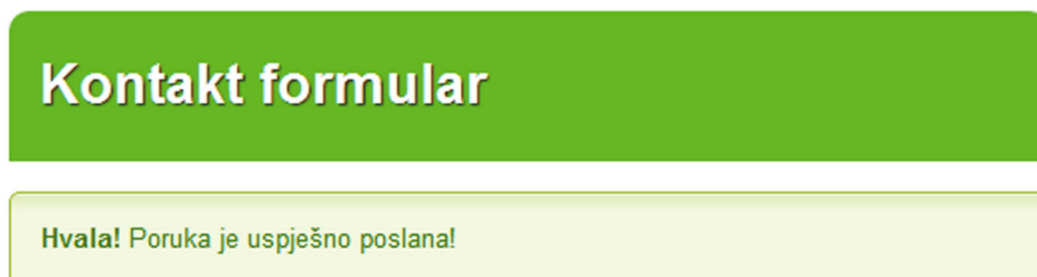
mail($prima, $tema, $poruka_slanje, $zaglavlje);
?>

```

Izvorni kod 3.23. Prikaz *PHP* koda za prihvata unosa formulara i slanje na konačnu *e-mail* adresu

Prikazani kod prikuplja unose prosljeđene *AJAX POST* metodom i sprema ih u varijable prikladnog naziva. Konačan izgled poruke generira se pridruživanjem prikladnog teksta i vrijednosti pojedinih varijabli u jedinstvenu varijablu naziva *\$poruka_slanje*. Konačna poruka šalje se *mail()* funkcijom na adresu definiranu *\$prima* varijablom, u prikazanom slučaju na *marko.cacic@gmail.com*.

Pri uspješnom slanju poruke posredstvom izrađenog formulara, sustav prikazuje poruku o uspjehu, uz zadržan prikaz zaglavlja stranice, dok su polja formulara uklonjena prikladnom animacijom.



Slika 3.9. Prikaz obavijesti o uspješno poslanoj *e-mail* poruci

Slijedi konačan izgled pristigle *e-mail* poruke:



Slika 3.10. Izgled pristigle e-mail poruke

Kako bi konačno izrađeno sučelje u potpunosti odgovaralo standardima modernog *Web 2.0* okruženja, kvaliteta prikaza i korisničkog doživljaja mora biti podjednaka na što je moguće više različitih uređaja. Stoga, koristeći *Modernizr JavaScript* biblioteku, potrebno je implementirati mogućnost provjere tzv. *touch* značajki korisnikovog *Web* preglednika, što je osnovna karakteristika modernih mobilnih uređaja.

Stoga, kako bi mogli koristiti mogućnosti *Modernizr biblioteke*, u `<head></head>` segmentu stranice moramo definirati putanju (eng. *path*) do *modernizr.js* datoteke na slijedeći način:

```
<script src="js/modernizr-2.5.3.js" type="text/javascript"></script>
```

Izvorni kod 3.24. Definiranje putanje *modernizr.js* datoteke, u *head* segmentu *HTML* stranice

Navedenu datoteku preuzeli smo sa službenog *Web site-a Modernizr* projekta (<http://modernizr.com/downloads/modernizr-2.5.3.js>) te ju smjestili u *js* direktorij.

Slijedi ugradnja funkcije koja će prepoznati *touch* značajke i ovisno o vrsti korisnikovog uređaja, odnosno *Web* preglednika, aplicirati *width* (hrv. širina) parametar na željene elemente sučelja. Kako bi praktičnije mogli upravljati širinom *div* kontejnera klase sadržaj, uklonit ćemo trenutnu definiciju širine definiranu u *CSS* datoteci.

Slijedi prikaz koda kojim upravljamo širinom željenih elemenata ovisno o *touch* značajkama korisnikovog *Web* preglednika, odnosno uređaja:

```
<script type="text/javascript">
$(document).ready(function() {
  if (Modernizr.touch) {
    $('.sadrzaj').width('auto');
  }
  else {
    $('.sadrzaj').width('800px');
    $('input[name=posalji]').width('250px');
  }
});
</script>
```

Izvorni kod 3.25. Prikaz *JavaScript* koda za upravljanje širinom *div* kontejnera klase *sadrzaj* i *submit* gumba formulara

Prikazani kod potrebno je smjesiti u `<head></head>` segment stranice.

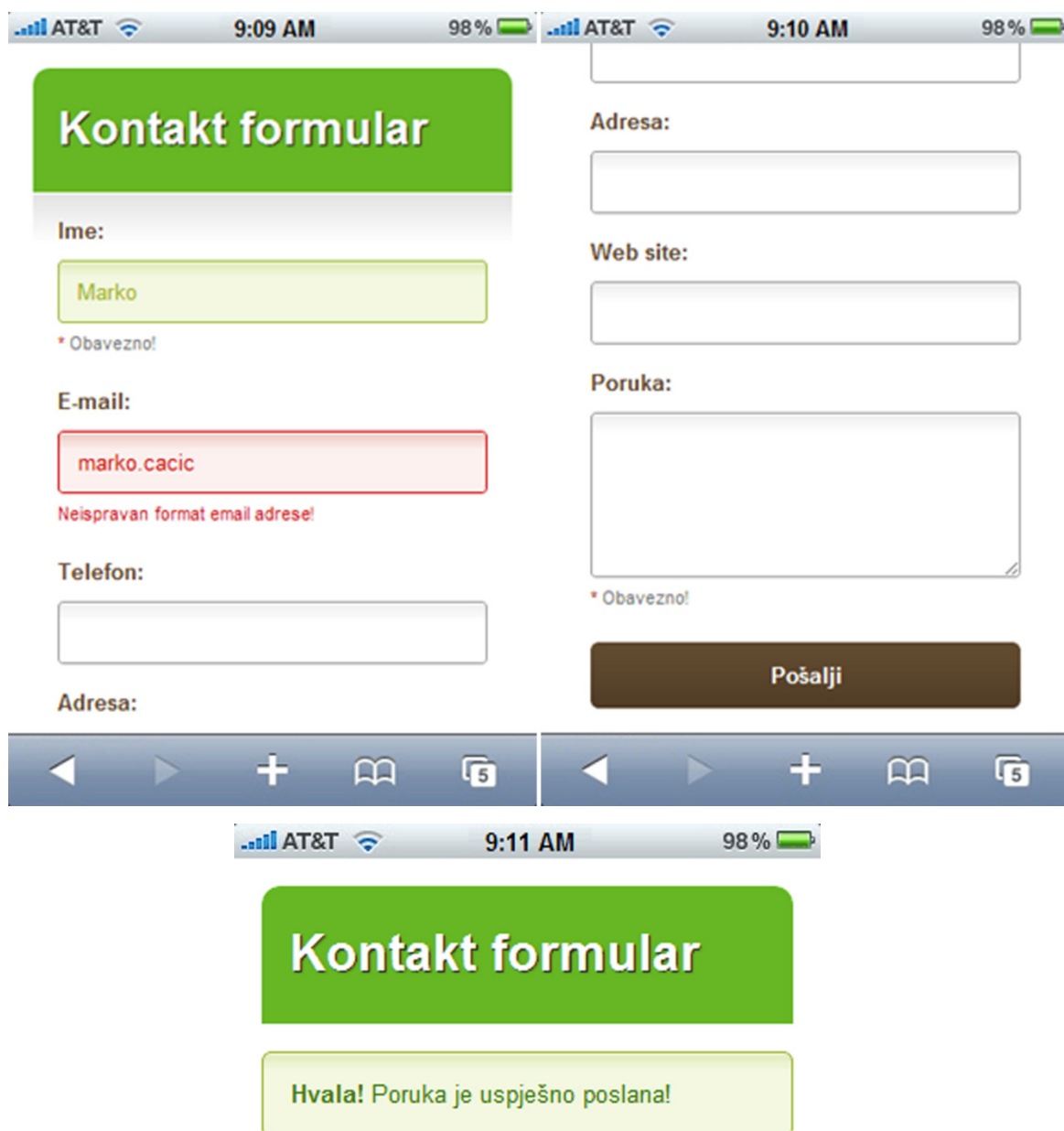
Osim navedenog, u svrhu bolje kontrole prikaza, u je potrebno smjestiti i slijedeći *meta tag* specifičan za mobilne uređaje:

```
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1">
```

Izvorni kod 3.26. Definiranje *viewport* opisnog *taga* potrebnog za optimizaciju prikaza sučelja na mobilnim uređajima

Također, pri izradi sustava provjere unosa, ugrađena je tzv. fokus (eng. *focus*) funkcionalnost koja prilikom detekcije greške, prikaz na mobilnom uređaju fokusira na problematično polje, što je posebno korisno kod većih formulara.

Slijedi prikaz konačnog rješenja sučelja na *iPhone* mobilnom uređaju:



Slika 3.11. Prikaz konačnog rješenja na *iPhone* mobilnom uređaju

4. ZAKLJUČAK

Širenjem Interneta i razvojem *Web* tehnologija, započela je postupna transformacija prvobitno tekstualno orijentiranih *Web* stranica u privlačne multimedijske proizvode. Unazad desetak godina, *Web* se transformirao iz običnog statičnog medija u interaktivnu multimedijску platformu. Temeljna karakteristika modernog *Web* okruženja, simbolično nazvanog *Web 2.0*, poticanje je korisnika na stvaranje i dijeljenje sadržaja, što zahtijeva visoku razinu potrebne interaktivnosti sa sučeljem aplikacije. U skladu s potrebom, definiran je niz pravila grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju kako bi se korisnika potaknulo na interakciju i tijekom cijelog procesa zadržao kvalitetan korisnički doživljaj (eng. *user experience* - *UX*). Prema relevantnoj literaturi iz područja oblikovanja (eng. *design*) u *Web 2.0* okruženju, definirane su opće karakteristike sučelja modernih *Web* aplikacija. Proučavanjem navedenih karakteristika te analizom sučelja *Basecamp* i *Stumbleupon* servisa, definirali smo opća pravila grafičkog i funkcionalnog oblikovanja u *Web 2.0* okruženju. To su:

- Jednostavnost
- Jasno naglašene površine od posebnog interesa
- Pregledan razmještaj elemenata
- Smisljena i jednoznačna navigacija
- Jasno vođenje korisnika kroz procese pomoću sustava provjere unosa i obavještanja u realnom vremenu
- Atraktivno stilizirani elementi sučelja
- Upečatljive, izražajne, pastelne boje
- Jasnoća i oštrina
- Dosljednost
- Prilagodljivost različitim dimenzijama, odnosno rezolucijama zaslona
- Kompatibilnost osnovnih funkcija s vodećim *Web* preglednicima

Implementacija navedenih odlika zahtijeva složene razvojne postupke i naprednu tehničku izvedbu, pomoću modernih programskih tehnologija. Ovaj

diplomski rad posebno se osvrnuo na slijedeće tehnologije koje smo i praktično koristili pri izradi sučelja kontakt formulara:

- *Client side* tehnologije:
 - *HTML* opisni jezik
 - *CSS* opisni jezik
 - *JavaScript* programski jezik (*jQuery* radni okvir, *Modernizr* biblioteka)
- *Server side* tehnologije:
 - *PHP*

Izradom sučelja u praktičnom dijelu diplomskog rada prikazali smo cjelokupan proces planiranja, grafičkog oblikovanja i kodiranja, odnosno programiranja funkcionalnog sučelja za slanje poruka, tzv. kontakt formulara (eng. *contact form*), prema prethodno definiranim pravilima i standardima *Web 2.0* okruženja.

Izrađeno sučelje koristi napredne tehnike provjere unosa (eng. *input validation*), inovativne metode grafičkog oblikovanja pomoću trenutno podržanih mogućnosti modernog *CSS3* jezika te posjeduje sposobnost prilagodbe prikaza karakteristikama korisnikovog uređaja. Navedene funkcionalnosti osiguravaju kvalitetan korisnički doživljaj tijekom cijelog vremena korištenja sučelja, odnosno aplikacije te omogućavaju podjednako kvalitetan prikaz na većini *desktop* i mobilnih uređaja.

Razvoj *Web* tehnologija u bliskoj budućnosti, a pogotovo završetak službene specifikacije *HTML5* i *CSS3* opisnih jezika kao novih, opće prihvaćenih i implementiranih standarda, omogućit će daljnji razvoj *Web-a* prema grafički iznimno bogatoj, multimedijски usmjerenoj platformi visoke razine interaktivnosti. Stoga, segment razvoja naprednih korisničkih sučelja u *Web 2.0* okruženju potrebno je prepoznati i prihvatiti kao novo, atraktivno i iznimno profitabilno područje unutar grafičke struke te ga početi tretirati ravnopravno s tradicionalnim poslovima grafičkih stručnjaka.

5. LITERATURA

- [1] Lynne Schrum, Web 2.0: New Tools, New Schools, International Society for Technology in Education, 2007, ISBN: 1564842347
- [2] Patrick Mcneil, The Web Designer's Idea Book Volume 2: The Latest Themes, Trends and Styles in Website Design, How, 1st Edition, 2012, ISBN: nedostupno, Amazon Digital Services ASIN: B005453A5I
- [3] Dan Cederholm, Web Standards Solutions: The Markup and Style Handbook, Apress, 2nd edition, 2009, ISBN: 1430219203
- [4] Cennydd Bowles, Undercover User Experience Design (Voices That Matter), New Riders Press, 1st Edition, 2010, ISBN: nedostupno, Amazon Digital Services ASIN: B0045JKJ7G
- [5] Leslie M. Orchard, Ara Pehlivanian, Scott Koon, Harley Jones, Professional JavaScript Frameworks: Prototype, YUI, Ext JS, Dojo and MooTools, John Wiley & Sons, 2009, ISBN: 0470384596
- [6] Nicholas C. Zakas, Professional JavaScript for Web Developers, Wrox, 3rd edition, 2012, ISBN: 1118026691
- [7] Terry Morris, Terry Felke-Morris, Basics of Web Design: HTML5 and CSS3, Addison Wesley, 1st edition, 2011, ISBN: 0137003382
- [8] Smashing Magazine, Successful Freelancing for Web Designers: The Best of Smashing Magazine, Wiley, 1st edition, 2011, ISBN: 1119992737
- [9] James Gillies, Robert Cailliau, How the Web was Born: The Story of the World Wide Web, Oxford University Press, USA, 2000, ISBN: 0192862073
- [10] Erik Wilde, Wilde's WWW: Technical Foundations of the World Wide Web, Springer, 1st edition, 1998, ISBN: 3540642854
- [11] Gary B. Shelly, Web 2.0: Concepts and Applications (Shelly Cashman), Course Technology, 1st edition, 2010, ISBN: 1439048029
- [12] Tom Funk, Web 2.0 and Beyond: Understanding the New Online Business Models, Trends, and Technologies, Praeger, 2008, ISBN: 0313351872
- [13] David McFarland, JavaScript & jQuery: The Missing Manual, Pogue Press, 2nd edition, ISBN: 1449399029

- [14] Eric G. Swedin, Computers: The Life Story of a Technology, Greenwood Publishing Group, 2005, ISBN: nedostupan, Amazon Digital Services ASIN: B003UHUMMG
- [15] Books, LLC, Graphical User Interface: History of the Graphical User Interface, General Books LLC, 2010, ISBN: 1156773113
- [16] K. Scott Allen, What Every Web Developer Should Know About HTTP, OdeToCode LLC, 2012, ISBN: nedostupan, Amazon Digital Services ASIN: B0076Z6VMI
- [17] Subrata Goswami, Internet Protocols: Advances, Technologies and Applications, Springer, 2003, ISBN: 140207476X
- [18] Anthony T. Holdener III, Ajax: The Definitive Guide, O'Reilly Media, 1st Edition, 2008, ISBN: 0596528388
- [19] Jennifer Niederst Robbins, Learning Web Design: A Beginner's Guide to (X)HTML, Style Sheets, and Web Graphics, 2007, O'Reilly Media, 2007, ISBN: 0596527527
- [20] Terry Burrows, Blogs, Wikis, MySpace, and More: Everything You Want to Know About Using Web 2.0 but Are Afraid to Ask, Chicago Review Press, 2008, ISBN: 155652756X
- [21] James Governor, Duane Nickull, Dion Hinchcliffe, Web 2.0 Architectures, O'Reilly Media, 2009, ISBN: 0596514433
- [22] Shama Kabani, The Zen of Social Media Marketing: An Easier Way to Build Credibility, Generate Buzz, and Increase Revenue, BenBella Books, 2nd Edition, 2012, ISBN: 1936661632
- [23] Dan Cederholm, Bulletproof Web Design: Improving Flexibility and Protecting Against Worst-case Scenarios with HTML5 and CSS3, New Riders, 2011, ISBN: 0132883384
- [24] Giles Colborne, Simple and Usable Web, Mobile, and Interaction Design, New Riders, 2010, ISBN: 0321714156
- [25] Julius Wiedemann, Web Design: Studios 2, Taschen, 2007, ISBN: 3822830100
- [26] Janine Warner, David LaFontaine, Mobile Web Design For Dummies, For Dummies, 1st Edition, 2010, ISBN: 0470560967
- [27] <http://basecamp.com/one-page-project> - Basecamp.com, Basecamp's exclusive single-page projects keep everything organized, 15.04.2012.
- [28] <http://www.stumbleupon.com/help/basics/stumbleupon-new-user-guide/> - StumbleUpon.com, StumbleUpon New User Guide, 19.04.2012.

[29] Janine Warner, David LaFontaine, Mobile Web Design For Dummies, For Dummies, 1st Edition, 2010, ISBN: 0470560967

[30] Andy Harris, HTML, XHTML and CSS All-In-One For Dummies, For Dummies, 2nd Edition, 2010, ISBN: 0470537558

[31] Vikram Vaswani, PHP: A Beginner's Guide, McGraw-Hill Prof Med/Tech, 2008, ISBN: 0071549013

POPIS PRILOGA

Oznaka	Objašnjenje
PRILOG 1 (CD)	Diplomski rad u digitalnom obliku
PRILOG 2 (CD)	Datoteke izrađene u praktičnom dijelu diplomskog rada

POPIS OZNAKA I KRATICA

Oznaka / Kratica	Objašnjenje / Puni naziv
HTML	HyperText Markup Language
XHTML	eXtensible HyperText Markup Language
CSS	Cascading Style Sheets
URL	Uniform Resource Locator
HTTP	HyperText Transfer Protocol
JPEG	Joint Photographic Experts Group
GIF	Graphic Interchange Format
AJAX	Asynchronous JavaScript and XML
UX	User eXperience
RSS	RDF Site Summary, Really Simple Syndication
XML	eXtensible Markup Language
USD	United States Dollar
API	Application Programming Interface
CD	Compact Disk

POPIS SLIKA

Oznaka	Naziv	Str.
Slika 2.1.	Prikaz segmenta sučelja stiliziranog <i>HTML</i> formulara za unos vijesti u <i>CMS</i> sustav	7
Slika 2.2.	Prikaz segmenta sučelja stiliziranog <i>HTML</i> formulara nakon odabira opcije „Kratki tekst“	8
Slika 2.3.	Prikaz segmenta sučelja stiliziranog <i>HTML</i> formulara nakon odabira opcije „Slika“	9
Slika 2.4.	Prikaz segmenta sučelja stiliziranog <i>HTML</i> formulara nakon odabira opcije „Video“	9
Slika 2.5.	<i>HTML submit</i> gumb naziva „Spremi“ i <i>anchor</i> gumb za poništavanje trenutne akcije	10
Slika 2.6.	<i>jQuery UI</i> dijaloški okvir (eng. <i>dialog</i>) s porukom o uspješno obavljenoj akciji	10
Slika 2.7.	Prikaz početne stranice sučelja <i>Basecamp project management</i> aplikacije	22
Slika 2.8.	<i>Basecamp project management</i> aplikacija – prikaz stranice projekta	23
Slika 2.9.	Sučelje <i>Basecamp project management</i> aplikacije – prikaz polja za unos sadržaja i obavijesti o pogrešnom unosu	24
Slika 2.10.	<i>Basecamp project management</i> aplikacija – prikaz dodatne poruke o nelogičnom unosu	24
Slika 2.11.	<i>Basecamp project management</i> aplikacija – prikaz sučelja za pisanje projektne dokumentacije	25
Slika 2.12.	Prikaz <i>Basecamp project management</i> aplikacije na različitim mobilnim uređajima	25
Slika 2.13.	<i>Stumbleupon</i> aplikacija – prikaz polja za unos sadržaja i mogućnosti registracije pomoću osobnog <i>Facebook</i> računa (eng. <i>account</i>)	27
Slika 2.14.	<i>Stumbleupon</i> aplikacija – obavještavanje korisnika o neuspješnom unosu	27
Slika 2.15.	<i>Stumbleupon</i> aplikacija – prikaz sučelja za odabir korisnikovih interesa	28
Slika 2.16.	<i>Stumbleupon</i> aplikacija – prikaz sučelja za izmjenu, ocjenjivanje i dijeljenje (eng. <i>sharing</i>) prikazanog sadržaja	28

Slika 2.17.	<i>Stumbleupon</i> aplikacija – prikaz osnovnog izbornika s opcijama „Pomoć“ (eng. <i>Help</i>), „Postavke“ (eng. <i>Settings</i>) i „Odjava“ (eng. <i>Sign Out</i>)	29
Slika 2.18.	Prikaz <i>Stumbleupon</i> aplikacije na različitim mobilnim uređajima	29
Slika 3.1.	Prikaz grafičkog idejnog rješenja korisničkog sučelja kontakt formulara, izrađeno pomoću <i>Adobe Photoshop</i> grafičkog uređivača (eng. <i>editor</i>)	43
Slika 3.2.	Prikaz osnovne strukture sučelja kontakt formulara, bez apliciranih CSS stilskih definicija	44
Slika 3.3.	Prikaz grafičkog idejnog rješenja zaglavlja stranice	46
Slika 3.4.	Prikaz <i>HTML</i> dokumenta s pripadajućim kontakt formularom nakon apliciranja željenih CSS definicija	50
Slika 3.5.	Prikaz <i>input type="text"</i> polja formulara u početnom stanju	53
Slika 3.6.	Prikaz <i>input type="text"</i> polja formulara nakon detekcije pogreške	54
Slika 3.7.	Prikaz <i>input type="text"</i> <i>e-mail</i> polja formulara nakon detekcije pogreške	54
Slika 3.8.	Prikaz <i>input type="text"</i> polja formulara nakon ispravke pogreške	54
Slika 3.9.	Prikaz obavijesti o uspješno poslanoj <i>e-mail</i> poruci	56
Slika 3.10.	Izgled pristigle <i>e-mail</i> poruke	57
Slika 3.11.	Prikaz konačnog rješenja na <i>iPhone</i> mobilnom uređaju	59

POPIS TABLICA

Oznaka	Naziv	Str.
Tablica 3.1.	Prikaz planiranih grafičkih i funkcionalnih karakteristika sučelja kontakt formulara	40

POPIS IZVORNIH KODOVA

Oznaka	Naziv	Str.
Izvorni kod 3.1.	Struktura jednostavnog <i>HTML</i> dokumenta	32
Izvorni kod 3.2.	Primjer stiliziranja izravnim vezanjem na <i>HTML</i> tag	33
Izvorni kod 3.3.	Primjer stiliziranja posredstvom <i>CSS</i> klase	33
Izvorni kod 3.4.	Primjer stiliziranja posredstvom <i>CSS</i> identifikatora	33
Izvorni kod 3.5.	Primjer povezivanja <i>HTML taga</i> i pripadajuće <i>CSS</i> klase	34
Izvorni kod 3.6.	Primjer povezivanja <i>HTML taga</i> i pripadajućeg <i>CSS</i> identifikatora	34
Izvorni kod 3.7.	Primjer definiranja vanjske <i>CSS</i> datoteke pomoću <i>HTML link taga</i>	34
Izvorni kod 3.8.	Primjer <i>inline</i> stiliziranja odabranog <i>HTML taga</i>	35
Izvorni kod 3.9.	Primjer <i>PHP</i> koda koji unutar <i>HTML</i> paragrafa ispisuje određeni tekst ovisno o parnosti vrijednosti spremljene u varijablu \$broj	37
Izvorni kod 3.10.	Prikaz konačnog ispisa u <i>Web</i> pregledniku, nakon izvršavanja Izvornog koda 3.9.	37
Izvorni kod 3.11.	Prikaz <i>HTML</i> koda osnovne strukture sučelja kontakt formulara	45
Izvorni kod 3.12.	Prikaz <i>HTML</i> koda zaglavlja stranice	46
Izvorni kod 3.13.	Prikaz <i>CSS</i> definicija zaglavlja stranice	46
Izvorni kod 3.14.	Definiranje vanjske <i>style.css</i> datoteke pomoću <i>HTML link taga</i>	47
Izvorni kod 3.15.	Definiranje <i>CSS</i> stilova pojedinih elemenata kontakt formulara	48
Izvorni kod 3.16.	Definiranje <i>CSS</i> stilova za poništavanje zadanih (eng. <i>default</i>) postavki <i>HTML</i> elemenata te formatiranje osnovnog <i>div</i> kontejnera klase sadržaj	49
Izvorni kod 3.17.	Povezivanje <i>input</i> polja formulara sa željenom klasom naziva „normalno“	49
Izvorni kod 3.18.	Definiranje putanje <i>jquery.js</i> datoteke, u <i>head</i> segmentu	51

	<i>HTML</i> stranice	
Izvorni kod 3.19.	Definiranje putanje kontakt-formular.js datoteke, u <i>head</i> segmentu <i>HTML</i> stranice	51
Izvorni kod 3.20.	Definiranje <i>UTF-8 charseta</i> posredstvom opisnog (eng. <i>meta</i>) <i>taga</i>	51
Izvorni kod 3.21.	Prikaz <i>JavaScript</i> koda potrebnog za provjeru unosa (eng. <i>input validation</i>)	53
Izvorni kod 3.22.	Prikaz <i>JavaScript</i> koda za predaju unesenih podataka <i>server side</i> datoteci	55
Izvorni kod 3.23.	Prikaz <i>PHP</i> koda za prihvat unosa formulara i slanje na konačnu <i>e-mail</i> adresu	56
Izvorni kod 3.24.	Definiranje putanje <i>modernizr.js</i> datoteke, u <i>head</i> segmentu <i>HTML</i> stranice	57
Izvorni kod 3.25.	Prikaz <i>JavaScript</i> koda za upravljanje širinom <i>div</i> kontejnera klase <i>sadrzaj</i> i <i>submit</i> gumba formulara	58
Izvorni kod 3.26.	Definiranje <i>viewport</i> opisnog <i>taga</i> potrebnog za optimizaciju prikaza sučelja na mobilnim uređajima	58