

Istraživanje mogućnosti CSS okvira

Vuzdar, Ivan

Undergraduate thesis / Završni rad

2016

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:216:446714>

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

Download date / Datum preuzimanja: **2025-02-28**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU
GRAFIČKI FAKULTET ZAGREB

ZAVRŠNI RAD

Ivan Vuzdar



Sveučilište u Zagrebu
Grafički fakultet

Smjer: Tehničko - tehnološki

ZAVRŠNI RAD

ISTRAŽIVANJE MOGUĆNOSTI
CSS OKVIRA

Mentor:
doc. dr. sc. Tajana Koren Ivančević

Student:
Ivan Vuzdar

Zagreb, 2016

SAŽETAK

Cilj ovog završnog rada je obraditi, analizirati i prikazati mogućnosti CSS radnih okvira (*eng. CSS framework*) koji se koriste u izradi i optimizaciji web stranica.

CSS okviri su pripremljeni kako bi ubrzali proces izrade internetskog sadržaja, omogućili korištenje nekih novih tehnologija te pripremili lakši pristup kodu koji se koristi za izradu sadržaja.

Pomoću CSS okvira moguće je upravljati responzivnošću web stranice, odnosno prilagoditi njen sadržaj, oblik i funkciju za mnogobrojne medije koji se danas koriste. U ovom radu postavljene su razlike između čistog CSS koda te već pripremljenih radnih okvira, koje su njihove prednosti i mane te njihove posebnosti.

Također analizirani su neki od najpoznatijih radnih okvira, na koji način funkcioniraju te zašto su postali toliko popularni unazad 10 godina uz razvitak responzivnog dizajna. Na kraju završnog rada prikazana je funkcionalnost okvira uz primjere na web stranici.

KLJUČNE RIJEČI: HTML, web stranica, CSS okviri, CSS framework, internetski sadržaj, responzivnost stranice, responzivni dizajn.

ABSTRACT

The purpose of this thesis is to process, analyze and present the functions of CSS Frameworks which are used to build and optimize web pages. CSS Frameworks are used to speed up process of making internet pages, open room for new technologies and deliver easier handling of the code itself. This leads to writing less code and spending fewer time making pages.

With the help of CSS frameworks it is possible to manipulate the dynamics of page. What this means is that pages are becoming responsive, rather, their content is making adjustments according to media from which the page is accessed. This thesis sets differences between clean CSS and pre-made frameworks, also it surveys their benefits and flaws and distinguishes their uniqueness.

Moreover, some of the most used frameworks are introduced, their functions are analyzed and it is presented why are they so popular in the past 10 years. In the end, the functionality of frameworks is shown with examples on web page.

KEY WORDS: HTML, web site, CSS, CSS framework, internet content, responsive design

SADRŽAJ

1. UVOD	1
2. TEORIJSKI DIO	2
2.1. CSS	2
2.2. UPORABA CSS-A	2
2.2.1. Interno povezivanje HTML-a i CSS-a	3
2.2.2. Vanjsko povezivanje HTML-a i CSS-a	3
2.2.3. Umetanje pravila unutar HTML elemenata	4
2.2.4. CSS sintaksa.....	4
2.3. MOGUĆNOSTI CSS-A	5
2.4. RESPONZIVNI DIZAJN	7
2.5. CSS OKVIRI	7
2.6. BOOTSTRAP	9
2.6.1. Uporaba Bootstrapa.....	10
2.7. FOUNDATION	16
2.8. SEMANTIC USER INTERFACE.....	18
3. EKSPERIMENTALNI DIO	22
3.1. BOOTSTRAP	22
3.2. FOUNDATION	30
4. ZAKLJUČAK	36
5. LITERATURA	37

1. UVOD

Internet je globalna računalna mreža nastala još u 1960-im godinama. Određene organizacije održavaju dijelove mreže te na njoj korisnici mogu objavljivati sadržaj. Količina informacija koja je dostupna na internetu raste eksponencijalno iz dana u dan, a s time i broj oblikovanog sadržaja. Sve te informacije sadržane su, oblikovane i prikazane na webu (eng. *World Wide Web*). To je jedan od najpopularnijih internetskih servisa, a nastao je početkom 1990-ih godina. Web se najčešće koristi HTTP protokolom, odnosno protokolom za prijenos hipertekstualnih dokumenata (eng. *Hypertext Transfer Protocol*).

Korisnici uz pomoć internetskog preglednika pristupaju sadržajima na webu. Na primjer, putem weba moguće je pristupiti različitim internetskim stranicama, internet bankarstvu, moguće je čitati novine, gledati televizijske programe, prodavati i kupovati stvari itd. Sve ove vrste sadržaja prikazane su u dokumentima pod nazivom web stranica. Ona sadrži kod, odnosno jezik, koji se prikazuje u internetskom pregledniku, a zove se HTML (eng. *Hypertext Markup Language*). U početku HTML je bio jezik namijenjen objavljivanju znanstvenih radova, a kod je upravljao tekstem prema njegovoj funkciji. Naslovi, podnaslovi, odlomci, razmaci između odlomaka, natuknice i slično bile su samo neke od mogućnosti oblikovanja teksta.

Kako je web postajao sve popularniji javila se potreba za uljepšavanjem stranice, a upravo je HTML poslužio u ostvarenju toga cilja. S vremenom, korisnicima se javila velika želja imati stilski jezik koji bi sadržaj prikazao na način veoma sličan novinskom rasporedu. 1994. Hakon Wium Lie predložio je stilski jezik CSS (eng. *Cascading Style Sheets*). Pomoću CSS jezika moguće je napraviti raspored informacija i sadržaja na web stranici te ih oblikovati prema želji.

Zbog razvitka tehnologije korisnici su uz primarni pristup webu, osobnim računalima, počeli koristiti svoje mobilne telefone i tablete. Ovakvi mediji zahtijevaju dodatnu prilagođenost različitim veličinama zaslona. Nadalje, izrada web stranice te mogućnost prikaza njenog sadržaja na raznim uređajima zahtjeva mnogo više vremena i mnogo više resursa. Iz tog razloga počele su se razvijati nove tehnologije kako bi proces izrade bio efikasniji, brži i kvalitetniji.

2. TEORIJSKI DIO

2.1. CSS

CSS je stilski jezik pomoću kojeg uređujemo sadržaj unutar HTML-a. 1994. godine osmislio ga je Hakon Wium Lie [1], profesor na MIT-u, dok je radio u Švicarskom institutu za nuklearna istraživanja (CERN).

Hakonova firma davale je standarde stilskom jeziku, koji određuju način rukovanja web preglednika jezikom. Tvrtke koje su razvijale web preglednike su trebale odlučiti da li će se u novoj verziji preglednika pronaći pravila i propisi iz nove inačice jezika. World Wide Web Consortium (W3C) [2] je organizacija koja se bavi standardizacijom tehnologija korištenih na webu. W3C također sudjeluje na razvoju drugih standarda kao što su HTML, XMP, SVG i sličnih tehnologija.

CSS je bio moćan jezik u to doba jer je omogućavao promjenu fontova, razmaka između riječi, slova i redaka teksta. Promjena margine i rubova su bili nešto što nije bilo dostupno sa samim HTML-om.

Nova inačica jezika nazvana "CSS2" pojavljuje se 1998. godine. U toj novoj inačici pojavljuju se mogućnosti apsolutnog, relativnog i fiksnog pozicioniranja HTML, sjena, prostor oko sadržaja (eng. padding) i slično. Nakon izlaska CSS2 verzije pa do danas koristi se tako zvana treća inačica, "CSS3", no uz nju postoje još deseci raznih dodataka koji na primjer smanjuju volumen koda kojeg je potrebno napisati za funkcioniranje određenih stilova ili pak unose nove "pseudo" klase koje pružaju nove mogućnosti.

2.2. Uporaba CSS-a

Kada se spominje CSS zapravo se misli pravila i propise kojima oblikujemo web stranicu. Način na koji CSS i HTML dokument komuniciraju može biti određen na 3 različita načina: vanjskim povezivanjem, internim povezivanjem ili ubacivanjem CSS izravno u HTML elemente. HTML elementi su ključni faktori za funkciju stranice jer oni nose njen sadržaj. HTML elementi imaju oznake (eng. *tag*). Većina elemenata imaju početnu i završnu

oznaku, a između njih je neka informacija, odnosno sadržaj. Također elementi sa sobom nose attribute koji ih opisuju, a dijele se na klase i identifikacije.

Pravila stilskog jezika imaju određene selektore, svojstva i vrijednosti koje ih opisuju. Svojstva i njihove vrijednosti stavljena su unutar vitičastih zagrada, dok se selektor nalazi ispred zagrade.

2.2.1. Interno povezivanje HTML-a i CSS-a

Interno povezivanje HTML-a i CSS-a se vrši tako da se upisuje element *style* između početne i završne oznake *head* elementa te unutar *style* elementa upisujemo pravila, što prikazuje slika 1.

```
1 <html>
2 <head>
3 <style>
4 <!-- Interno povezivanje-->
5 <!-- CSS pravila-->
6 </style>
7 </head>
8 <body>
9
10
```

Slika 1. *Interno povezivanje*

Ovakav način se rijetko koristi, a povoljan je samo ako je potrebno imati CSS svojstva sadržana u jednom HTML dokumentu.

2.2.2. Vanjsko povezivanje HTML-a i CSS-a

Vanjsko povezivanje se najčešće koristi iz razloga što se koristi jedna CSS datoteka za sve HTML dokumente vezane za web stranicu. Kasnije održavanje i upravljanje web stranicom je na ovaj način dodatno olakšano.

Način na koji se povezuju HTML i CSS jest da se unutar *head* elementa dodaje poveznica sa destinacijom gdje se nalazi CSS datoteka sa svim pravilima. Slika 2 prikazuje vanjsko povezivanje HTML dokumenta sa CSS-om.

```
1 <html>
2 <head>
3 <!-- Vanjsko povezivanje -->
4 <link rel="stylesheet" href="CSS_datoteka.css" />
5 </head>
6 <body>
```

Slika 2. Vanjsko povezivanje

2.2.3. Umetanje pravila unutar HTML elemenata

U elemente se dodaje atribut *style* sa pravilima CSS-a. Pravila su definirana između navodnih znakova, a mogu se koristiti bilo koja od postojećih pravila. Slika 3. prikazuje umetanje CSS-a unutar *p* elementa.

```
1 <html>
2 <head>
3 </head>
4 <body>
5 <p style="color:red;">
6 Ovaj paragraf biti će
7 prikazan u crvenoj boji
8 </p>
9 </body>
```

Slika 3. umetanje CSS-a

2.2.4. CSS sintaksa

CSS sintaksa sastoji se od 3 dijela: selektori, svojstva i vrijednosti. Selektor je HTML element kojem se dodjeljuje vrijednost nekog svojstva. Svojstvo je već od prije definirana oznaka koja ukazuje na što se djeluje pomoću koda. Svako svojstvo ima neku vrijednost kojom opisujemo njen stil. Slika 4 prikazuje CSS sintaksu.

```
1
2 selektor {
3     svojstvo: vrijednost;
4 }
```

Slika 4. CSS sintaksa

Svojstva i vrijednosti su uvijek odvojeni dvotočkom, te se nalaze unutar vitičastih zagrada. Nakon zadnje vrijednosti, svojstvo se završava sa točkom-zarez koji odvaja jedno od drugog svojstva. Način na koji se dodaje novo svojstvo jest da se u novi redak, još uvijek unutar zagrada, napiše novo svojstvo sa nekim novim vrijednostima. Na taj način selektor je potpun te nosi informaciju u HTML dokument gdje oblikuju zadane elemente.

Selektore je moguće grupirati radi jednostavnijeg korištenja pravila. Na ovaj se način istovremeno definiraju jednaka svojstva za različite selektore. Selektori se upisuju jedan iza drugoga odvojeni zarezom, te se nakon upisa stavlja vitičasta zagrada te se dalje tretirao kao običan selektor.

Način na koji CSS oblikuje određene dijelove HTML dokumenta jest pomoću klasa i identifikacija. HTML elementi određeni su klasom ili identifikacijom, a u CSS datoteci se kao selektor uzima klasa ili identifikacija te joj se upisuju svojstva. Slika 5 prikazuje korištenje ID atributa.

```
1 <div id="izbornik">
2   <!-- Opcije izbornika -->
3 </div>
4
5 #izbornik {
6   text-align: center;
7   color: blue;
8 }
```

Slika 5. ID i Class

2.3. Mogućnosti CSS-a

Pomoću CSS-a moguće je odrediti font, njegovu veličinu i boju, vrstu fonta, dodati mu sjenu, debljinu i slično. Taj font je moguće odrediti pomoću klase ili identifikacije te ga je moguće pozvati i iskoristiti u toku izrade stranice. Na ovom principu funkcionira cjelokupan CSS jezik. Korištenjem CSS-a ubrzava se i olakšava uređivanje web stranica, ali se i doprinosi boljoj kvaliteti što u konačnici dostavlja pregledniju stranicu, veću snalažljivost korisnika i ugodniji doživljaj.

Mogućnosti CSS-a na neki način su ograničene sa njegovim svojstvima. Današnja inačica "CSS3" omogućuju se promjene svojstva pozadine, uređivanje teksta, fontova, okvira, margina, praznog prostora oko sadržaja, općenito dimenzija, pozicioniranja elemenata, klasifikacije raznih svojstva itd.

Mogućnosti "CSS3" inačice proširene su pseudo klasama koje dozvoljavaju korištenje efekata u nekim selektorima. Najčešći primjer je poveznica koju možemo prikazati kao već posjećenu, ne posjećenu, poveznicu sa pokazivačem, aktivnu poveznicu itd; tako da se svaka stilski razlikuje od prethodne. Slika 6 prikazuje svaki od navedenih primjera.

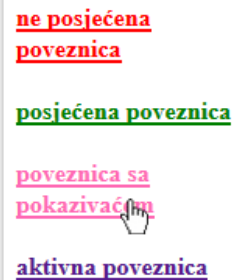
```
<!DOCTYPE html>
<html>
<head>
<style>
/* ne posjećena poveznica */
#a1:link {
    color: red;
}

/* posjećena poveznica */
#a2:visited {
    color: green;
}

/* poveznica sa pokazivačem */
#a3:hover {
    color: hotpink;
}

/* aktivna poveznica */
#a4:active {
    color: blue;
}
</style>
</head>
<body>

<p><b>
<a href="default.asp" target="_blank" id="a1">ne posjećena poveznica</a>
</br>
</br>
<a href="default.asp" target="_blank" id="a2">posjećena poveznica</a>
</br></br>
<a href="default.asp" target="_blank" id="a3">poveznica sa pokazivačem</a>
</br></br>
<a href="default.asp" target="_blank" id="a4">aktivna poveznica</a></b>
</p>
```



**ne posjećena
poveznica**

posjećena poveznica

**poveznica sa
pokazivačem**

aktivna poveznica

Slika 6. poveznice

2.4. Responzivni dizajn

Iako je responzivni dizajn jedna od mogućnosti CSS-a, zbog svoje važnosti i sve prisutnosti zahtjeva posebnu pozornost. Sredinom 2000-ih godina korisnici su počeli pristupati web stranicama sa raznih medija. Zbog toga se javlja potreba prilagodbe web stranice uređajima sa različitim rezolucijama i veličinama ekrana, odnosno re-dizajna kako bi sadržaj bio pregledan na osobnim računalima te u isto vrijeme i na npr. mobilnim uređajima. U CSS se uvode "medijski upiti" (eng. *media queries*). Oni su još postojali u "CSS2" inačici pod nazivom "tip medija" (eng. *media types*) no nisu bili previše korišteni. Tipovi medija korišteni su za drugačiji prikaz sadržaja prilikom printanja, korištenja uređaja za Brailleovo pismo i pisače, prezentacije itd.

Medijski upiti koriste se umetanjem "@media" pravila u CSS kod.[3] Pomoću ovih upita definira se širina i visina početnog prikaza, dimenzije uređaja ili medija, orijentaciju uređaja, rezoluciju i slično.

Danas se pomoću medijskih upita dizajniraju web stranice prilagođene za svaki postojeći mobilni uređaj, tablet, laptop te bilo kakav drugi uređaj kojime se pristupa na web stranicu. Budući da se broj medija sve više povećava postaje gotovo nemoguće ispuniti sve zahtjeve korisnika, odnosno održavati web stranicu te je isto tako i učiniti prilagođenim za svaki medij. Iz tog razloga web programeri se okreću alternativnim rješenjima ovakvih i sličnih problema. Počinje razvijanje CSS radnih okvira (eng. *CSS Framework*) kako bi se olakšala izrada web stranica, te se smanjio utrošak vremena na pisanje jedno te istog koda iznova za svaki projekt.

2.5. CSS okviri

CSS okviri ili radni okviri prvi put se pojavljuju oko 2005. godine u obliku CSS knjižnica (eng. *CSS library*), koje su sadržavale pravila sa već zadanim selektorima i njihovim svojstvima. Potrebno je samo poznavati nazive selektora koji se zapišu u HTML dokumentu te se na taj način "pozivaju" klase ili identifikatori, koji kasnije čine stil.

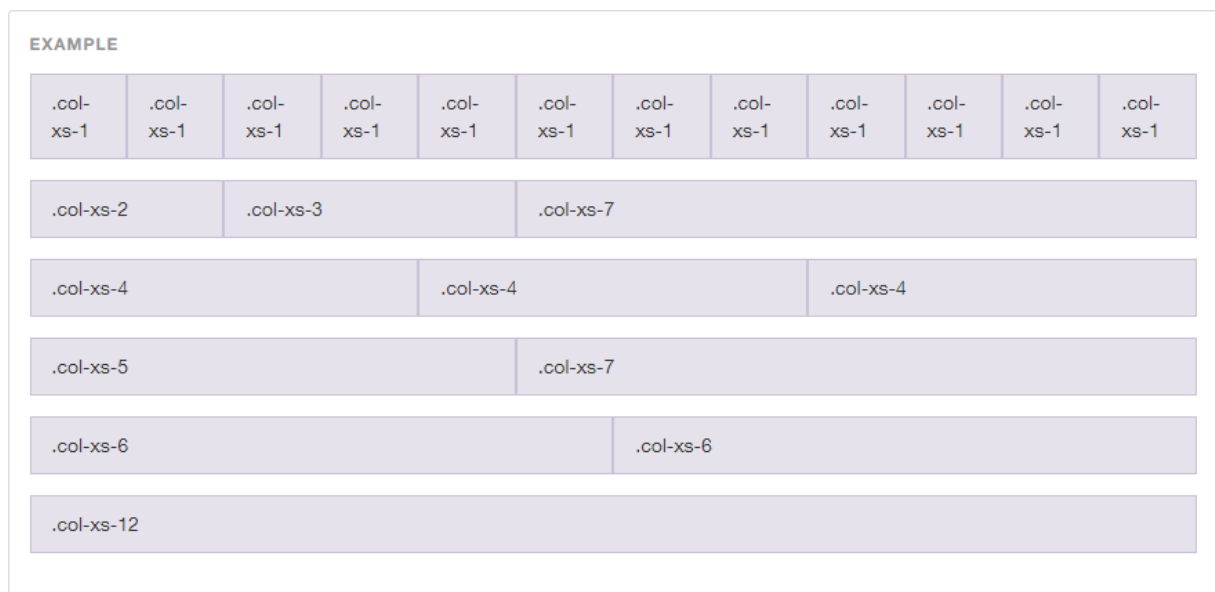
YUI (Yahoo User Interface Library) i *Blueprint* su pioniri među radnim okvirima koji su se tada pojavili. Glavni cilj radnog okvira je olakšati razvojni proces web stranice.

Postoje razne etape u razvojnog procesu koje se često ponavljaju i mogu biti zamorne za programera. Postupci kao što su resetiranje postavki preglednika, izrada rasporeda web

stranice, postavljanje tipografskih pravila i slično može oduzeti jako puno vremena, pogotovo kako se neki dijelovi često ponavljaju iz projekta u projekt. CSS knjižnice sadržavale su resurse kao što su CSS resetiranja, fontove, CSS rešetke, efekte animacija, gumbi, poveznice, gotove izbornike, izrađene forme i tako dalje.

S vremenom sve više programera je počelo koristiti ovakve knjižnice dok se nisu razvili potpuni radni okviri kao što su *Bootstrap*, *Foundation*, *Semantic*, *Pure CSS*, *W3.CSS*. CSS radni okviri temelje se na "sistemu rešetki" (eng. *grid system*). CSS tretira elemente kao da su u zamišljenim okvirima, odnosno kutijama (eng. *box element*), u koje ubacujemo sadržaj. Radni okviri se razlikuju prema broju stupaca i redaka koji ih simuliraju.

Karakteristike radnog okvira mogu se svesti na to koliko stupaca ima u jednom retku te koja je veličina tih stupaca i redaka. Sustav rešetki u *Bootstrapu* prikazan je na slici 7.



Slika 7. Sustav rešetki - stupci i redovi

Gornji primjer prikazuje sustav rešetki u *Bootstrapu*. Sustav je definiran preko klase *.col* sa veličinom *xs*, što predstavlja *extra small*, te *1* što označuje broj stupaca u jednom retku. Broj stupaca u retku može varirati između 1 i 12, a kod nekih drugih radnih okvira broj stupaca u retku može varirati između 1 i 30. Svaki *box element* ima svoju širinu i visinu, prazninu oko sadržaja (eng. *padding*), granice (eng. *border*) i margine. Margina predstavlja prostor oko ruba, a koristi se za definiranje razmaka između dva *box elementa*. *Padding* se koristi da bi odvojili

sadržaj od ruba, odnosno udaljili ga iz estetičkih i funkcionalnih razloga. Rubovi mogu biti različitih debljina i različitih boja, a najčešće ili nevidljivi ili vrlo malih debljina.

2.6. Bootstrap



Slika 8. *Bootstrap Logo*

Bootstrap je radni okvir razvijen 2011. godine od strane 2 zaposlenika *Twitter* platforme.[4] Mark Otto i Jacob Thornton odlučili su razviti alat koji će im pomoći automatizirati procese tijekom izrade web stranica, uvelike ubrzati pisanje koda te ostaviti prostora za inovativna dizajnerska rješenja. Slika 8 predstavlja vizualni identitet radnog okvira *Bootstrap*. Prva inačica *Bootstrapa* bila je prilagođena samo za Internet Explorer 7 i 8, te se sastojala od samo CSS i HTML komponenata. Kasnije, početkom 2012. godine u već poprilično veliku knjižnicu *Bootstrapa* mjesto nalazi i *JavaScript* te cijeli rad počinje težiti ka responzivnom dizajnu. U inačici *Bootstrap 2.0.0.* sve komponente bile su prilagođene širokom spektru mobilnih uređaja, laptopa i ekrana velikih dimenzija. U 2013. godini američki servis za zapošljavanje, *Indeed*, dao je podatke o oglasima za posao vezanih za *Bootstrap* koji govore o popularnosti samog radnog okvira. Slika 9 pokazuje statistiku korištenja radnog okvira *Bootstrap*.



Slika 9. *Indeed.com* statistike

Jedna od glavnih prednosti *Bootstrapa* jest što je *open source*, odnosno besplatan je i dostupan svima. Korisnici mogu dodavati svoje dijelove koda te uzimati već postojeće i mijenjati ih prema svojim potrebama.

2.6.1. Uporaba Bootstrapa

Bootstrap uključuje responzivan, mobilno fluidan sistem rešetki koji se prilagođava u čak 12 stupaca ovisno o uređaju ili veličini početnog prikaza. Također, sastoji se od predodređenih klasa, CSS ekstenzija kao što su *Less* i *Sass* te novih i zanimljivih tipografskih rješenja.

Na službenim stranicama *Bootstrapa* moguće je pronaći dvije forme radnog okvira. Prva forma je sastavljeni i "minimaliziran" kod CSS-a, JavaScripta i fontova. Ne postoje izvorni dokumenti, niti su priloženi bilo kakvi dodatni dokumenti. Ovakva inačica je spremna za korištenje odmah, te se povezuje sa HTML dokumentom preko poveznice u *style* dijelu koda. Razlog ovako jednostavne implementacije jest MaxCDN - mreža za dostavu sadržaja (eng. *Content delivery network*) preko koje se dostavlja zadnja inačica sastavljenog i

minimaliziranog CSS-a. Uz CSS, potrebno je staviti i poveznice za jQuery i JavaScript. Slika 10 prikazuje implementaciju "minimaliziranog" *Bootstrapa*.

```
1 <!-- Posljednja inačica sastavljenog CSS-a -->
2 <link rel="stylesheet"
3 href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
4
5 <!-- jQuery library -->
6 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">
7 </script>
8
9 <!-- JavaScript -->
10 <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
11 </script>
12
```

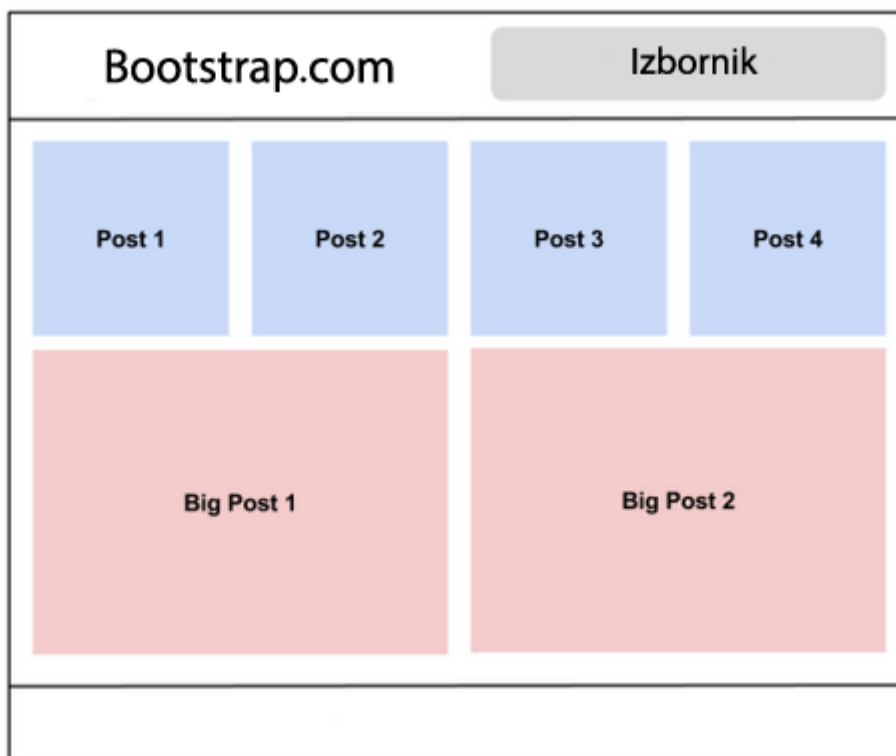
Slika 10. Implementacija *Bootstrapa*

Ovakva implementacija ima dodatnih prednosti. Budući da je primjena *Bootstrapa* u izradi web stranica jako velika, postoji mogućnost da su korisnici već prije skinuli *Bootstrap* sa MaxCDN-a posjećujući neke druge stranice što dovodi do puno bržeg učitavanja stranice.

Druga forma *Bootstrapa* jest potpun izvorni kod, izvorni kod za *Less*, *JavaScript*, fontove i ikone glifova (eng. *glyphicon*). Za ovakvo postavljanje najčešće se koriste se programi za upravljanje paketima (eng. *Package manager*) kao što su *Bower*, *npm* ili *Composer*. Uz packet managere, potrebno je instalirati *Grunt*, program koji pokreće skripte u JavaScriptu, automatizira procese, sastavlja izvorne kodove. *Grunt* će upravo sastaviti izvorni kod pisan u jednom programskog jeziku te ga transformirati u kod za CSS i JS. Budući da je *Grunt* ovisan o *Node.js* to je još jedan od uvjeta prije postavljanja *Bootstrapa*. *Node.js* jest okruženje za razvoj serverskih aplikacija.

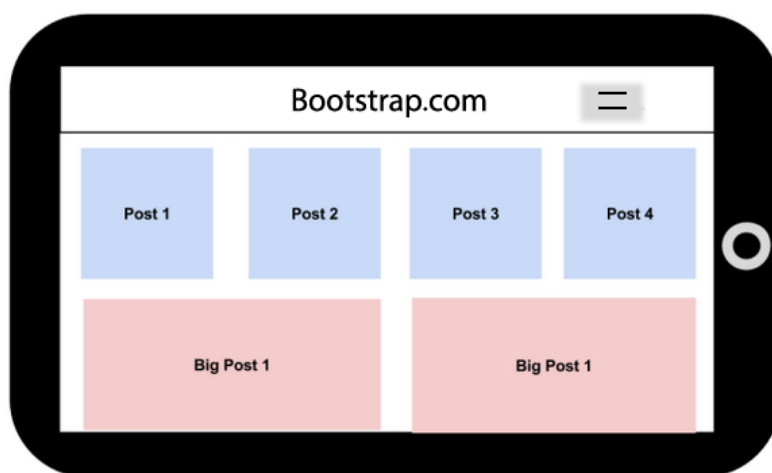
Nakon što su ispunjeni svi uvjeti, moguće je koristiti radni okvir zajedno sa svim njegovim opcijama, mogućnostima te pristupom izvornom kodu. Podložan je promjenama te ga je vrlo lako prilagoditi vlastitim željama.

Sustav rešetki u *Bootstrapu* ima 4 klase, *xs* za mobilne uređaje, *sm* za tablete, *md* za osobna računala te *lg* za veće dimenzije ekrana. Svaka klasa prikazuje elemente u većim ili manjim dimenzijama, ovisno o kojem se uređaju radi. Ove 4 klase određuju na koji način će reagirati HTML elementi kada dođe do promjene veličine prozora koji je otvoren u web pregledniku.



Slika 11. *Primjer responzivnosti - osobna računala*

Slika 11 prikazuje raspored elemenata za osobna računala. Stranica ima *header*, *content* i *footer*. *Header* predstavlja glavni izbornik, logo ili naziv stranice te često neku karakteristiku prema kojoj se web stranica prepoznaje (slika, boja, video ili nešto slično). Poanta responzivnog dizajna jest da se sadržaj prilagodi uređaju, no da se pritom ne narušava vizualni identitet stranice.



Slika 12. *Primjer responzivnosti - tableti*

Nakon što se veličina ekrana smanji, sadržaj stranice se prilagođava rezoluciji. *Box elementi* su se smanjili za određen postotak, kao i izbornici. Slika 12 je primjer responzivnog dizajna za tablete.

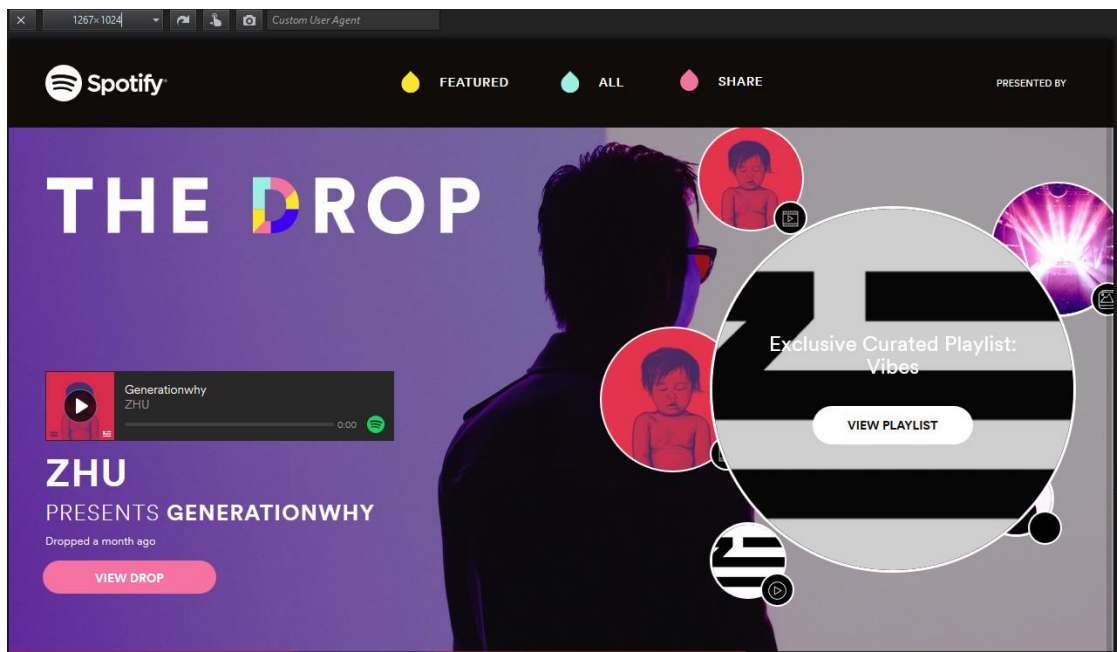


Slika 13. *Primjer responzivnosti - mobilni uređaji*

Kod mobilnih uređaja, *box elementi* su razmješteni i smanjeni, kao što je prikazano na slici 13, kako bi ostali na istom početnom prikazu, te prenijeli željenu informaciju. Pritom nisu izgubili identitet niti smisao.

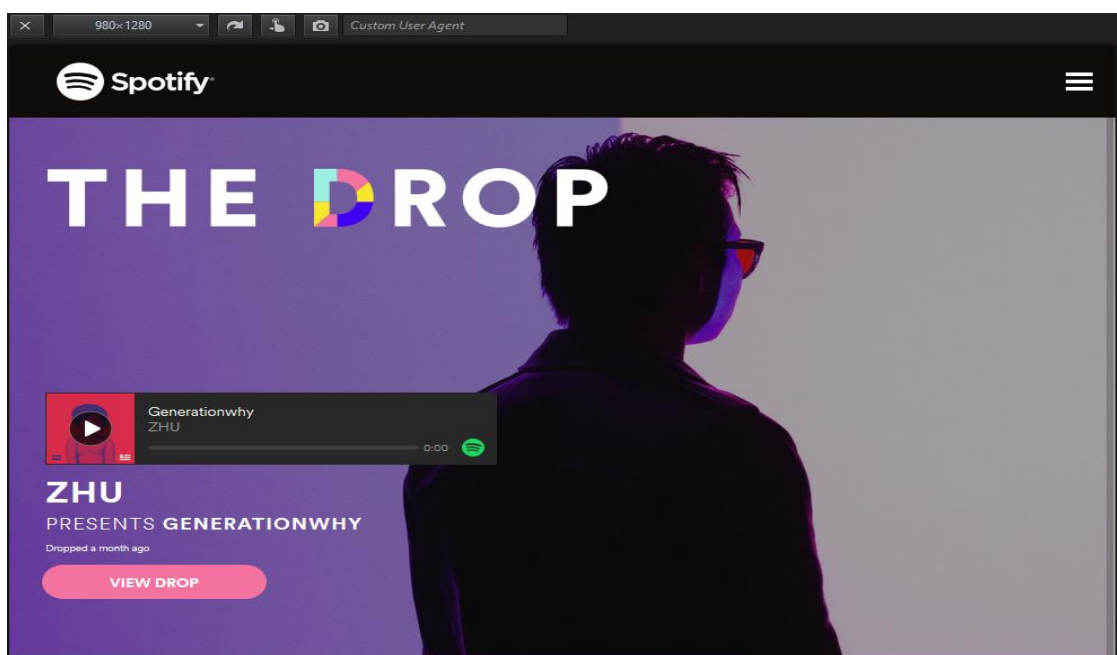
Jedan od modernih primjera web stranica rađenih u *Bootstrapu* jest *Spotify - The Drop*.

Slika 14. prikazuje veličina ekrana od 1267 x 1024 px - Potpun sadržaj, sa headerom i footerom, web player, animacije i poveznice sa slikama.



Slika 14. *Spotify responzivnost*

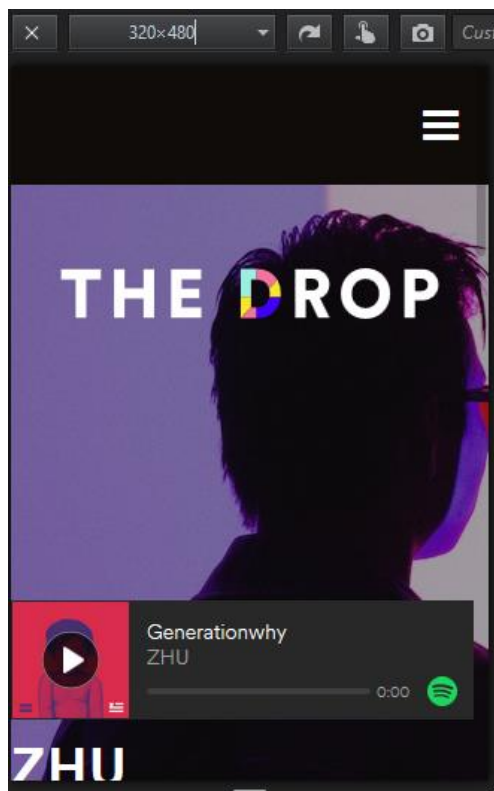
Kada se veličina ekrana promjeni na karakterističnu rezoluciju za tablete, 900 x 1280px, moguće je uočiti prve promjene. Zbog tehničkih specifikacija tableta i sličnih uređaja, animacije su uklonjene, a poveznice u *headeru* zbijene su u jedan padajući izbornik.



Slika 15. *Spotify responzivnost 2*

Slika 15 prikazuje početnu stranicu *Spotify*-ja prilagođenu za tablete. U konačnici, kada je veličina ekrana smanjena na 320 x 480 px, što je veličina ekrana mobilnog uređaja, vidi se prilagodba za takve ekrane. Sadržaj je napravio poravnavanje po sredini ekrana, *Spotify* logo više nije vidljiv, a aplikacija za reprodukciju glazbe dolazi u prvi plan.

Slika 16 prikazuje prilagođenu stranicu za mobilne telefone sa veličinom ekrana od 320 x 480 px.



Slika 16. *Spotify* responzivnost 3

2.7. Foundation



Slika 17. *Foundation* Logo

Foundation je vjerojatno drugi najpoznatiji *Front-end* radni okvir.[5] Foundation je nastao kao proizvod mnogobrojnih web stranica, aplikacija i e-mail servisa Bryana Zmijewskog i tvrtke ZURB. [6] Slika 17 prikazuje logo radnog okvira *Foundation*.

Izvorni kod *Foundationa* postaje dostupan na internetu 2011. godine te se počinje primjenjivati na velikom broju web stranica.

Foundation je velika kolekcija HTML-a, CSS-a i JS-a koja sadrži dizajnerske uzorke preko kojih se izgrađuju web stranice. Smanjuje se količina koda koju je potrebno pisati. Velik naglasak je na responzivnom dizajnu uz korištenje medijskih upita te *mobile-first approach*, odnosno prvo se dizajnira za mobilne uređaje. Isto tako, postoji velik naglasak na korištenju *Sass-a*, ekstenzije CSS-a koja omogućuje korištenje varijabli, pseudo klasa, posebnih nasljeđivanja i slično.

Temelj *Foundationa* jest rešetka od 12 stupaca, a postoji i mogućnost ugnježđivanja stupaca i redaka. Razlikuju se 3 klase, ovisno o uređajima za koje se dizajnira web stranica. Budući da je ovaj radni okvir *mobile-first*, prvo će se pisati kod za manje uređaje, a veći uređaji će naslijediti male stilove. Stoga je potrebno napraviti izmjene za veće uređaje. *Small* opisuje mobilne uređaje, *medium* stilovi nasljeđuju pravila od *small* osim ako nisu definirani drugačijim rasporedom koristeći klase za "srednje" uređaje. Isto pravilo vrijedi i za treću klasu koja se naziva *large*.

U novoj inačici *Foundation 6.1.* dodani su tako zvani "žljebovi" (eng. *gutter*) koji određuju prazan prostor između dva stupca u retku isto kao i prostor između ruba rešetke i ruba same stranice. Žljebovi postaju veći kod većih rezolucija što znači da su responzivni.

Budući da su gotovo svi radni okviri do 6.1. verzije *Foundationa* koristili opciju plutanja (eng. *float*) kako bi odredili poredak *box elementa*, *Foundation* uvodi nove rešetke nazvane *Flex Grid*. Princip rada je vrlo sličan *float-u*, no ovdje postoji mogućnost horizontalnog i vertikalnog poravnavanja, automatske veličine itd. Struktura je jednaka sustavu rešetki sa 12 stupaca, redci koriste klasu *.row*, a stupci koriste klasu *.column*. Ako klasa nema veličinu ona će ovaj put ispuniti prostor sa stupcima/redcima kao da je upisana maksimalna vrijednost. Ako je slučaj da 2 ili više klase stupaca nemaju određenu veličinu, prostor će se ispuniti dijeleći se na broj stupaca koji nemaju određenu veličinu.

Automatsko gomilanje je još jedna novost u radnim okvirima. Moguće je odabrati koji redovi će gomilati stupce jedan iza drugoga, a koji redovi će isključiti opciju gomilanja. Opcija je dodana kao svojstvo veličine klase npr. *small-unstuck*, *large-unstuck* itd.

Stupce je moguće poravnati po vertikalnoj ili horizontalnoj osi retka u kojem se nalaze. Horizontalno poravnavanje funkcioniра na isti način kao i poravnavanje teksta, moguće je poravnati stupce na lijevu ili desnu stranu, ostaviti ih na sredini, ostaviti jednak prostor između njih ili ih razdvojiti jedne od drugih.

Jedna od najvećih promjena jest korištenje svojstva *order*. Jedno od najmoćnijih svojstava *Foundationa*, a služi za definiranje poretka selektora. Određuje se koji će se element pojaviti prvi. *Order:1* označava da će se element sa tim svojstvom pokazati prvi, što je dosta zanimljivo s obzirom na koji način se koriste stupci i njihov poredak.

Uvedene su nove klase vidljivosti koje imaju sposobnosti prikazivanja elemenata pod različitim uvjetima te isto tako i sakrivanja elemenata kada dođe do promjene veličine ekrana. Ove klase moguće je koristiti za određivanje orijentacije uređaja što može pomoći tokom testiranja responzivnosti. Slika 18 je primjer klase vidljivosti.

```
1 <p class="show-for-landscape">Pejzažna orijentacija uređaja.</p>  
2 <p class="show-for-portrait">Portretna orijentacija uređaja.</p>
```

Slika 18. Klase vidljivosti

Brojne organizacije iz svijeta marketinga i prodaje koriste *Foundation* kao izbor radnog okvira. Među brendovima koji su učinili re-dizajn, ili jednostavno izradile svoje web stranice u *Foundationu*, su *Mini Cooper*, *Subaru*, *Lotus*, *Nissan*, *Coca Cola*, *Pixar*, *HTC*, *HP*

Store, Timberland, Vans, Adobe Open Source i mnogi drugi. Dnevni upis riječi "*Foundation*" na poznatoj tražilici *Google* broji preko 27,000 pretraživanja što govori o samoj popularnosti radnog okvira.

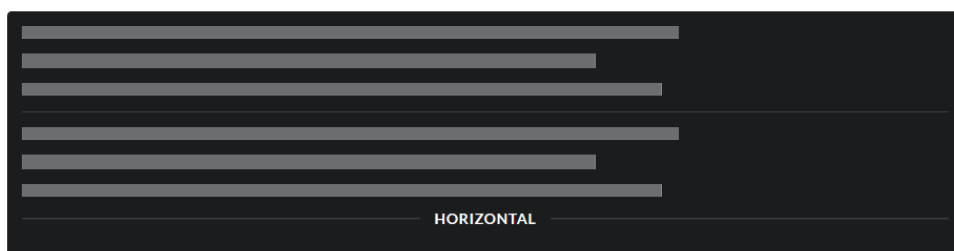
2.8. Semantic User Interface



Slika 19. *Semantic UI* Logo

Semantic User Interface (UI - korisničko sučelje) je jedan od popularnijih radnih okvira u *Front-end* svijetu. Slika 19 prikazuje logo *Semantic* radnog okvira. Karakteristike *Semantic UI* su što programerima dostavlja jednostavan i čitljiv stilski jezik kojeg opisuju cjelovite i smislene riječi. Klase, elementi, vrijednosti i svojstva su intuitivno nazvana tako da ih običan korisnik može prepoznati.[7] *Semantic* koristi 5 opisnih kategorija koje definiraju komponente korisničkog sučelja. *Elements, Collections, View i Module*[8] olakšavaju rad sa radnim okvirom te ga čine jedinstvenim. Instalacija radnog okvira može se napraviti pomoću CDN-a ili povezivanjem preko *Node.js* i ostalih upravljačima datoteka (eng. *packet manager*)

Skupinu *Elements* čine već poznati dijelovi za izgradnju web stranice kao što su gumbi, *containeri, header i footer*, forme za upis podataka, ikone, liste, no i neki novi elementi kao što je "pregrada" (eng. *divider*) pomoću koje odjeljujemo *box elemente*. Moguće je napraviti vertikalnu ili horizontalnu pregradu, te ona može sadržavati neki opisni tekst. Slika 20 predstavlja klasu pregrada u 2 forme, horizontalnoj i vertikalnoj.



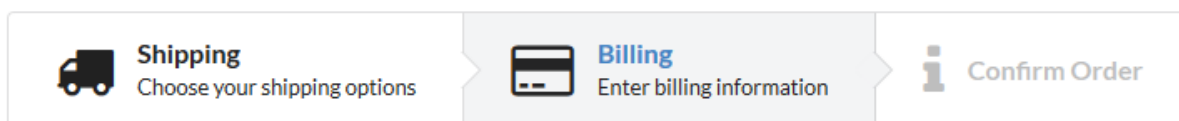
Slika 20. Klasa divider

Novost kod elemenata je korištenje zastava koje su inače uvedene preko SVG-a, no ovdje se koristi kolekcija slika koja je spremljena pod jednu slikovnu datoteku. Razlog tome je što klasa zastava sadrži slike svih država. Kada bi se učitala svaka pojedinačno koristilo bi se previše podatkovnog prometa te bi brzina učitavanja stranice bila vrlo mala.

U *Semanticu* se također koriste klase spojnice (eng. *rail*) koje prikazuju sadržaj izvan početnog prikaza web stranice. To je najčešće sadržaj koji dodatno opisuje prikazane informacije. Spojnice najčešće izgledaju kao blog postovi, a mogu se prikazati animacijama. Prelazak mišem preko naslova spojnice, započinje animacija te se novi stupac pojavljuje iz jedne strane web stranice te postaje dostupan za interakciju.

Reveal klasa (eng. *otkriti*) omogućuje izradu *box elementa* koji se otkrivaju, pomiču ili postaju ne vidljivi uz interakciju miša.

Step klasa služi za grafički prikaz neke radnje koja se odrađuje u više koraka. Slika 21 prikazuje *Step* klasu. Najčešći primjer je tijekom kupnje proizvoda putem web dućana, prikazati način poštarine, naplate te dostave.



Slika 21. Step klasa

Skupinu *Collections* čine upisne forme, sustav rešetki, izbornici, tablice i slično. Uvedena je klasa *breadcrumb* (prijevod "krušne mrvice") koja daje mogućnost prikaza hijerarhije među sadržajem. Slika 22 prikazuje klasu *breadcrumbs*.

[Home](#) > [Registration](#) > [Personal Information](#)

Slika 22. Klasa *breadcrumb*

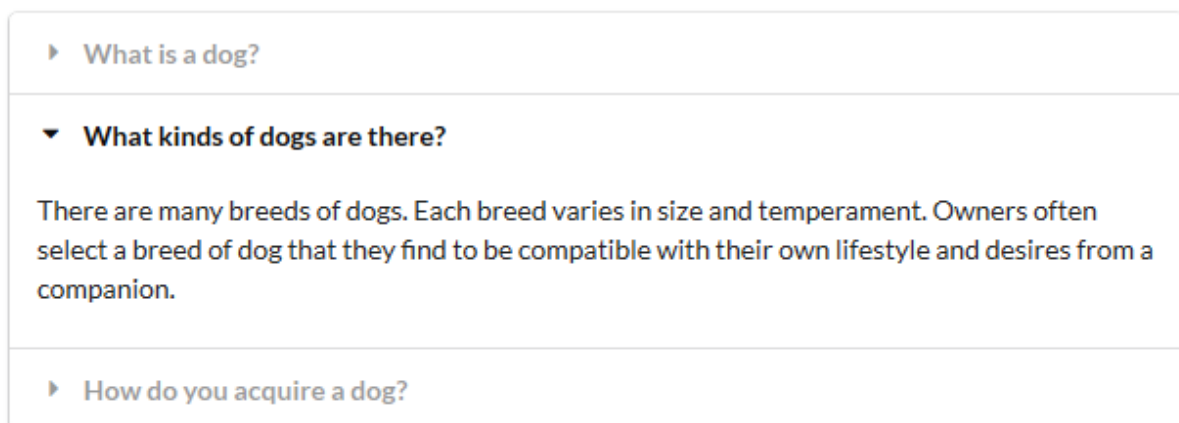
Sustav rešetki u *Semanticu* određen je sa 16 stupaca. Rešetke su napravljene fluidno, što znači da se tokom bilo kakve promjene veličine ekrana automatski prilagođavaju, a pritom zauzimaju maksimalno dostupan prostor.

Skupina *Views* (eng. pogledi) imaju klase koje se koriste za definiranje reklamnih oglasa, kartica profila, upisa i prikaza komentara, izrade proizvoda te statistike na kontrolnim pločama. Klasa statistika prikazana je na slici 23.



Slika 23. Klasa statistika

Skupina modula (eng. *modules*) su komponente sa interaktivnim JS funkcionalnostima. Klasa harmonika (eng. *Accordion*) daje mogućnost prikaza sadržaja pomoću prekidača. Prikaz je animiran, fluidan i u potpunosti responzivan. Postoji opcija prilagodbe tamnijim pozadinama. Primjer na slici 24.



Slika 24. Klasa *Accordion*

Klasa *Dimmer* skriva elemente, a cilj klase je da skrene pozornost na važne dijelove sadržaja.

Modal je još jedna nova pseudo klasa u *Semanticu*, a služi kako bi se spriječila interakcija korisnika sa sadržajem te mu prenijela neku poruku ili opciju. Način na koji se događa akcija jest da pozadina dobije transparentnu "zavjesu" te se pojavljuje poruka u središtu ekrana. Radnja koja se odvija u tom trenutku vrlo je slična *pop-up* prozorima.

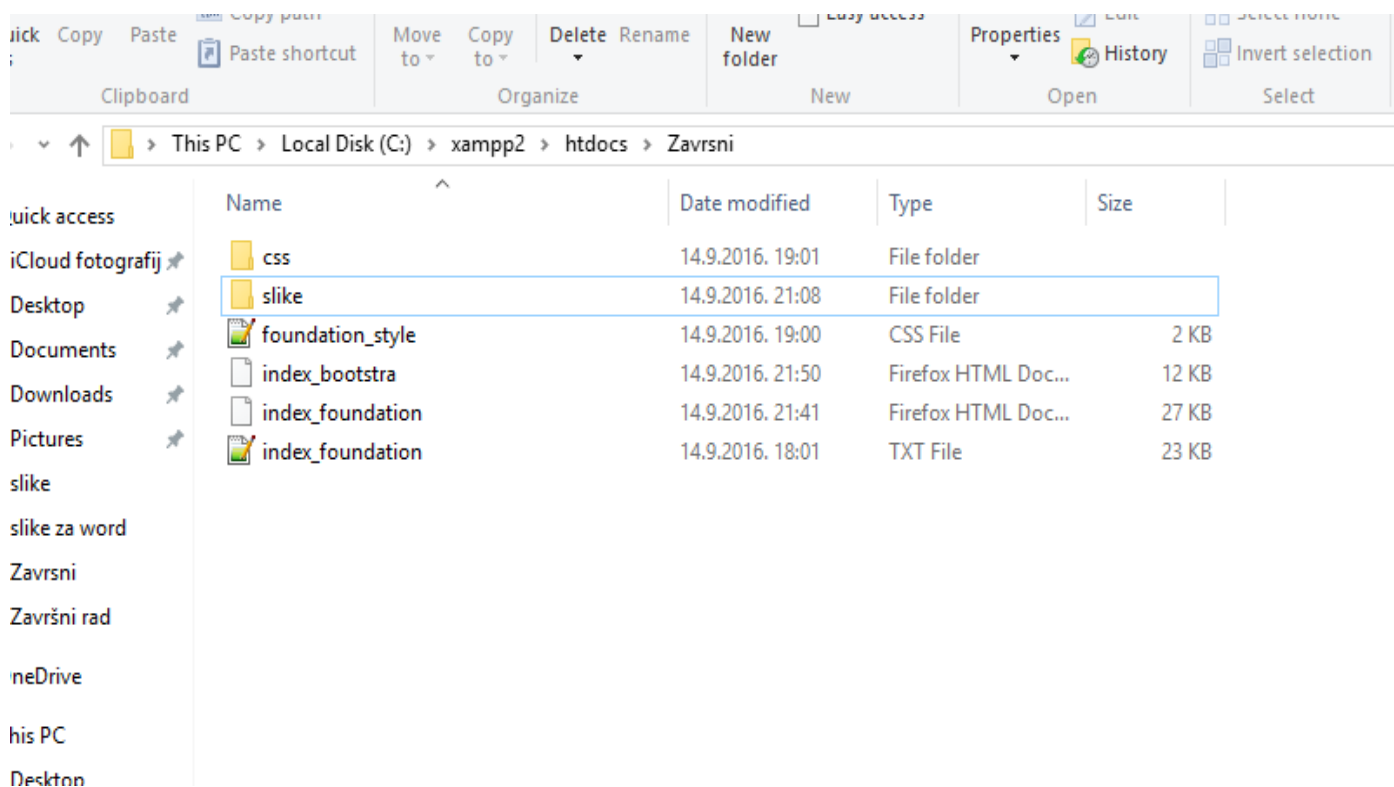
Shape opcija prikazuje tro-dimenzionalne objekte u dvije dimenzije, ali ostavlja mogućnost pogleda svih dijelova tog objekta. To je postignuto zanimljivim animacijama kojima korisnik može upravljati. Ova opcija se koristi za prikazivanje profila, dodatnih informacija ili manjih slika.

Semantic je postao popularan zbog svojeg naglaska na tranzicijama, animacijama i prikazima koji su jedinstveni. Ove opcije su dosta fluidne i događaju se vrlo brzo što nije uvijek slučaj kod radnih okvira.

3. EKSPERIMENTALNI DIO

3.1. Bootstrap

U eksperimentalnom dijelu završnog rada izrađena je i analizirana web stranica sastavljena od komponenti 2 različita radna okvira. Web stranica se sastoji od 2 HTML dokumenta te 2 knjižnice od svakog radnog okvira. Resursi radnih okvira pozvani su u HTML dokumentu svake stranice pozivanjem poveznica preko mreže za dostavu sadržaja MaxCDN. Mape slik, stilova te dokumenti radnih okvira nalaze se u istoj mapi kao i HTML dokumenti kako bi poveznica između njih bila valjana. Za potrebe izrade web stranice korišteni su programi *Photoshop*, *Notepad++*, *xampp* i *Waterfox* preglednik.



Slika 25. Root mapa web stranice

Slika 25 prikazuje mapu u kojoj se nalaze osnovni HTML dokumenti, slike i prikladan CSS datoteke.

Glavni kod napisan je u dokumentu "index_bootstrap.html" i "index_foundation.html". U tim dokumentima nalaze se ključne poveznice za radne okvire, informacije o vrsti znakova i vrsta prikaza. Slika 26 prikazuje vezu HTML dokumenta sa stilovima, temama i skriptama radnog okvira.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- vrsta prikaza, prilagodba preglednika i vrsta znakova -->
    <title>Istraživanje mogućnosti CSS okvira</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
    integrity="sha384-BVYiiSIFeK1dGmJRAKycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

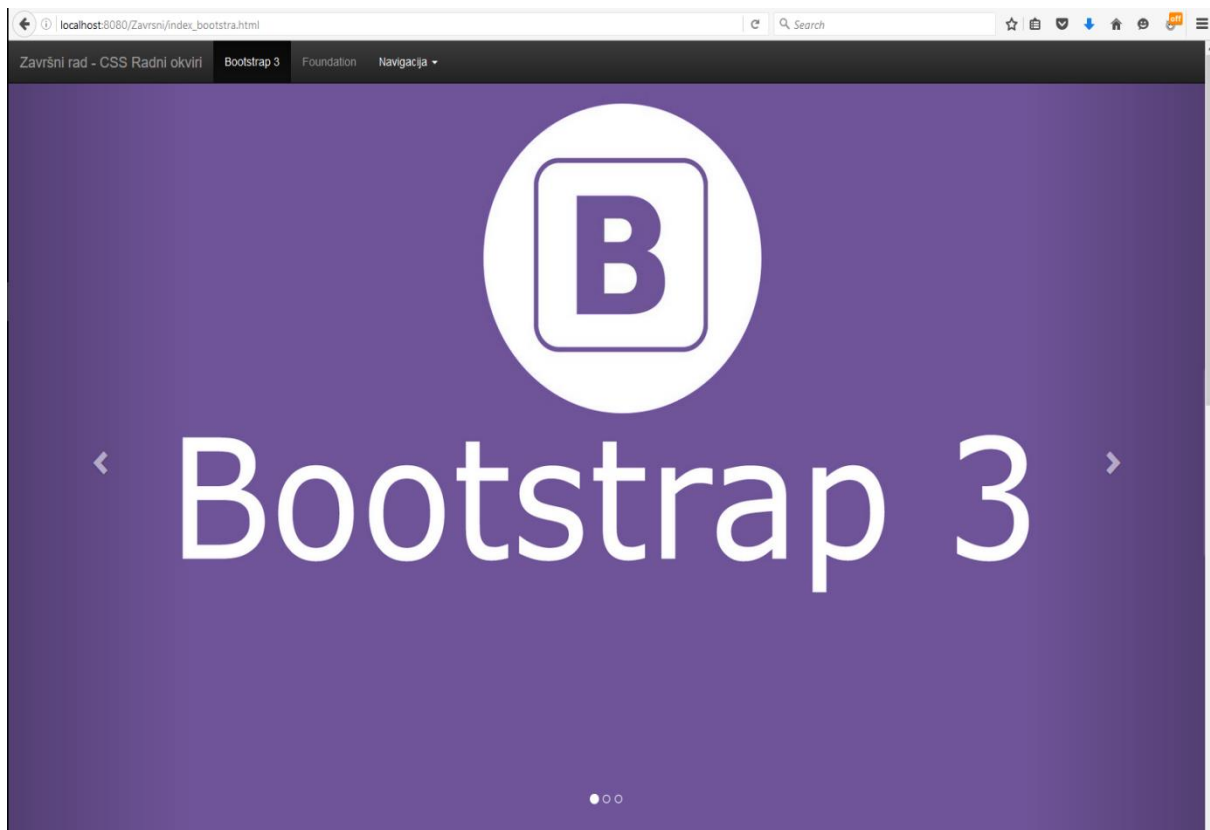
    <!-- Tema -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css"
    integrity="sha384-rHyoNliRsVxV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmgJQIXwl/Sp" crossorigin="anonymous">

    <!-- JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"
    integrity="sha384-Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA712mCWNIpG9mGCD8WGNIcPD7Txa" crossorigin="anonymous"></script>
  </head>
  <body { padding-top: 51px; }
</body>
```

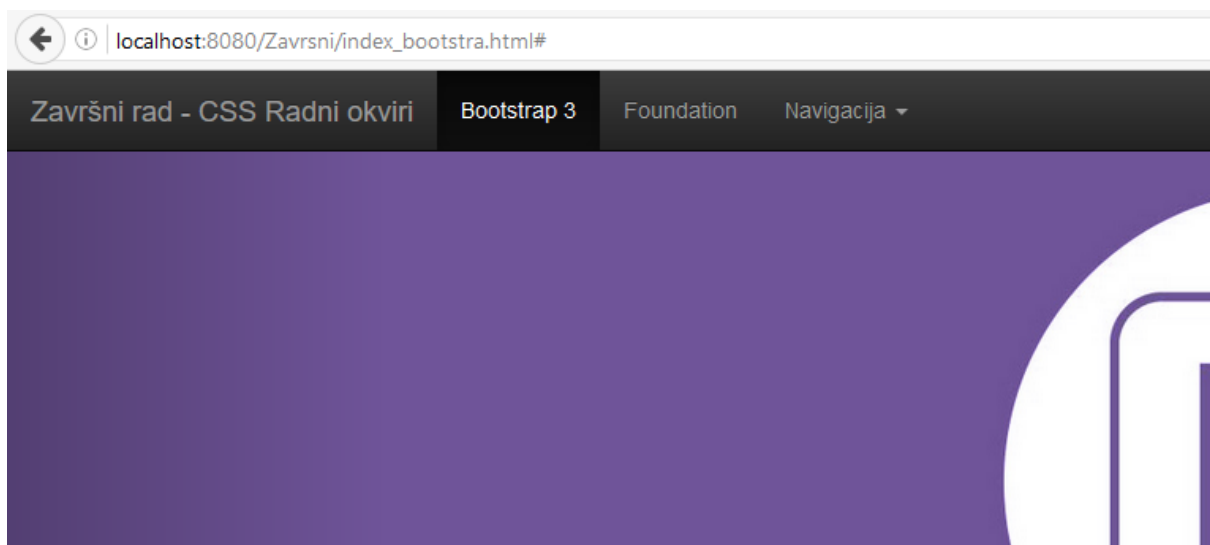
Slika 26. Poveznica za radni okvir

U radnom okviru *Bootstrap* napravljena je početna stranica koja se sastoji od *headera*, sadržaja i *footera*. U *headeru* je smješten glavni izbornik gdje se nalaze poveznice prema ostalim stranicama. U sadržaju stranice napravljen je vrtuljak (eng. *carousel*) koji prikazuje fotografije sa tekstom. Fotografije su ujedno i poveznice koje vode korisnika na druge dijelove stranice. Slika 27 primjer je početne stranice sa vrtuljkom.

U *headeru* se nalaze linkovi prema radnim okvirima pomoću kojih je stranica izrađena. *Header* je nosač vizualnog identiteta stranice, a na njemu se osim glavnih poveznica nalazi i izbornik sa navigacijom. Slika 28 prikazuje *header* stranice.



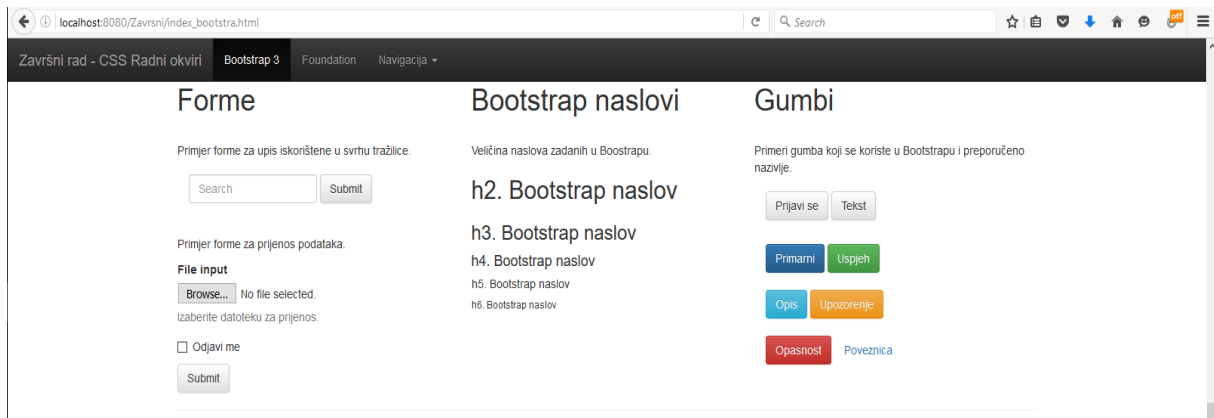
Slika 27. Početna stranica



Slika 28. Header početne stranice.

Carousel je već od prije napravljen element sa svojom JS skriptom i HTML/CSS kodom. *Bootstrap* nam omogućuje jednostavnu i brzu implementaciju ovog elementa upisivanje kodu HTML dokument. Izrađen je u jednom retku, a njegov stupac zauzima maksimalnu širinu web stranice. Ovaj vrtuljak je najmoderniji primjer elementa koji se stavlja ispod samog izbornika na web stranici. Glavni razlog tome je uspostava komunikacije između korisnika i web stranice. Na vrtuljku su stavljene slike sa zastavama (eng. *banner*) radnih okvira obrađenih u završnom radu.

Postoji mogućnost uređivanja koda pa se tako može dodati veći broj slika, drugačiji tekst, a može se i modificirati brzina okretanja slika te efekti okreta. Nakon vrtuljka predstavljeni su elementi koji se najčešće koriste u Bootstrapu, odnosno općenito u izradi web stranica. Redom to su forme za unos i slanje podataka, tipografija (tekst, naslovi, paragrafi itd.) te gumbi, što je prikazano na slici 29.



Slika 29. *Forme, gumbi i tipografija*

Ovi elementi pozicionirani su u jednom retku kroz 3 stupca. Svaka grupa elementa zauzima jedan stupac, a to je i vidljivo iz priložene slike. Svaki stupac zauzima širinu od 4 *box element* te nema nikakvu horizontalnu podjelu.

Kako bi forme funkcionirale potrebno ih je definirati unutar HTML dokumenta, sa opisnim klasama i identifikatorima. Pravila se uzimaju od prije povezanih stilskih datoteka koje sadržavaju sve mogućnosti i kombinacije stilova. Primjeri su prikazani na slici 30.

Forme

Primjer forme za upis iskorištene u svrhu tražilice.

Primjer forme za prijenos podataka.

File input

 No file selected.
Izaberite datoteku za prijenos.

Odjavi me

```
<h1>Forme</h1></br>
<p>Primjer forme za upis iskorištene u svrhu tražilice.</p>
<form class="navbar-form navbar-left" role="search">
  <div class="form-group">
    <input type="text" class="form-control" placeholder="Search">
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Slika 30. Forme za unos i HTML kod

Sljedeći elementi koji se koriste su naslovi. Podjela naslova temelji se na njihovoj veličini, odnosno veličini fonta. Koristi se tag *h* popraćen sa brojem koji označuje veličinu naslova. *Bootstrap* naslovi prikazani su na slici 31 zajedno sa kodom u HTML dokumentu.

Bootstrap naslovi

Veličina naslova zadanih u Bootstrapu.

h2. Bootstrap naslov

h3. Bootstrap naslov

h4. Bootstrap naslov

h5. Bootstrap naslov

h6. Bootstrap naslov

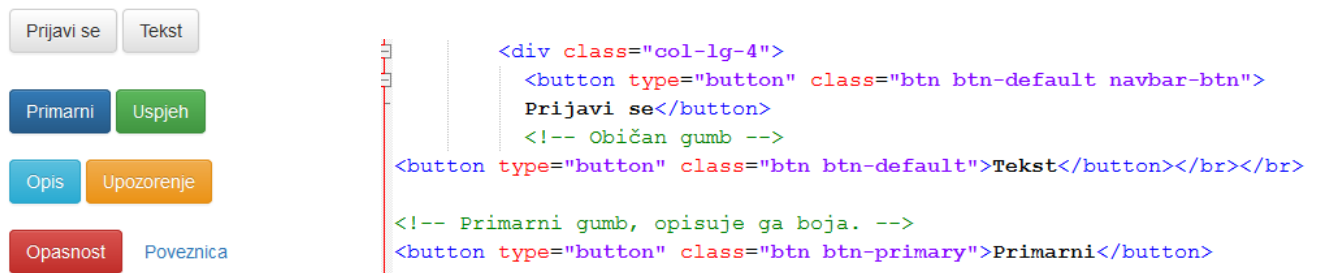
```
<h1> Bootstrap naslovi</h1>
</br>
<p>Veličina naslova zadanih u Bootstrapu.</p>
<h2>h2. Bootstrap naslov</h2>
<h3>h3. Bootstrap naslov</h3>
<h4>h4. Bootstrap naslov</h4>
<h5>h5. Bootstrap naslov</h5>
<h6>h6. Bootstrap naslov</h6>
```

Slika 31. *Bootstrap* naslovi

Vjerojatno najviše korišteni elementi na web stranicama su gumbi. Uloga gumba može biti potvrda unosa podataka, prijenos informacija na server, poveznica i slično. U *Bootstrapu* postoji 8 vrsti gumba koji su od prije zadani, ali postoji mogućnost dodatnog uređivanja gumba. Primjer gumba prikazuje slika 32.

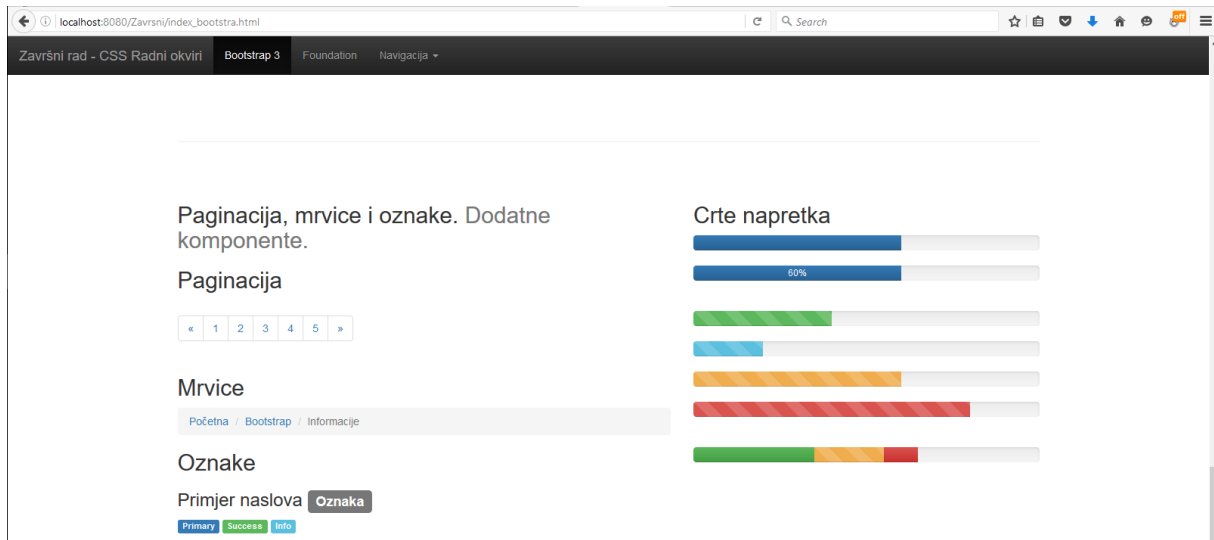
Gumbi

Primjeri gumba koji se koriste u *Bootstrapu* i preporučeno nazivlje.



Slika 32. *Bootstrap* gumbi

Paginacija, *breadcrumbs*, ili mrvice, i oznake nova su skupina elemenata u *Bootstrapu*. Pozicioniranje ovih elemenata riješeno je kroz 2 stupca. U prvom stupcu nalaze se 3 retka, a u drugom stupcu postoji samo jedan redak. Širina oba stupca je 6 *box elementa*. *Paginacija* služi za brzu navigaciju kroz sadržaj web stranice. Moguće ju je iskoristiti u web dućanima kod listanja proizvoda, kod prikaza velikog broja podataka i kod prikaza slika. *Breadcrumbs* daje informaciju korisniku na kojem djelu web stranice se nalazi te pruža opciju vraćanja na prethodni sadržaj. Oznake se koriste kod definiranja određenih skupina informacija. Gore navedeni elementi prikazani su na slici 33.



Slika 33. *Paginacija, breadcrumbs, oznake i crte*

Crte napretka iskoristive su kod preuzimanja i slanja datoteka, pokretanja aplikacija ili učitavanja video sadržaja. Uz pomoć JavaScripta dodane su 2D i 3D animacije ovih elemenata.

Elementi na web stranici su prilagođeni svim rezolucijama, od pametnih telefona i tableta, do ekrana visoke rezolucije. Postavljanjem veličine ekrana na tablet, elementi mijenjaju poziciju, ali ne i redoslijed. Elementi koji dolaze na kraju, nalazit će se ispod prijašnjih elemenata. Slika 34. prikazuje promjenu veličine ekrana uređaja s kojeg se pristupa web stranici.



Slika 34. Prilagođeni sadržaj novom uređaju

Zaključno, prema dostupnim resursima i informacijama na službenoj stranici Bootstrapa, vidljivo je da je *Bootstrap* vrlo razvijen radni okvir. Sa svojim mogućnostima usmjeren ka izradi administracijskog sučelja, web stranica za internetska poslovanja (e-commerce, e-bank.) i internetskih blogova. Naravno, budući da je izvoran kod dostupan svima i da je uređivanje elemenata, svojstava i atributa uvelike olakšano, moguće je stranicu prilagoditi i izraditi prema vrlo kompleksnim zahtjevima.

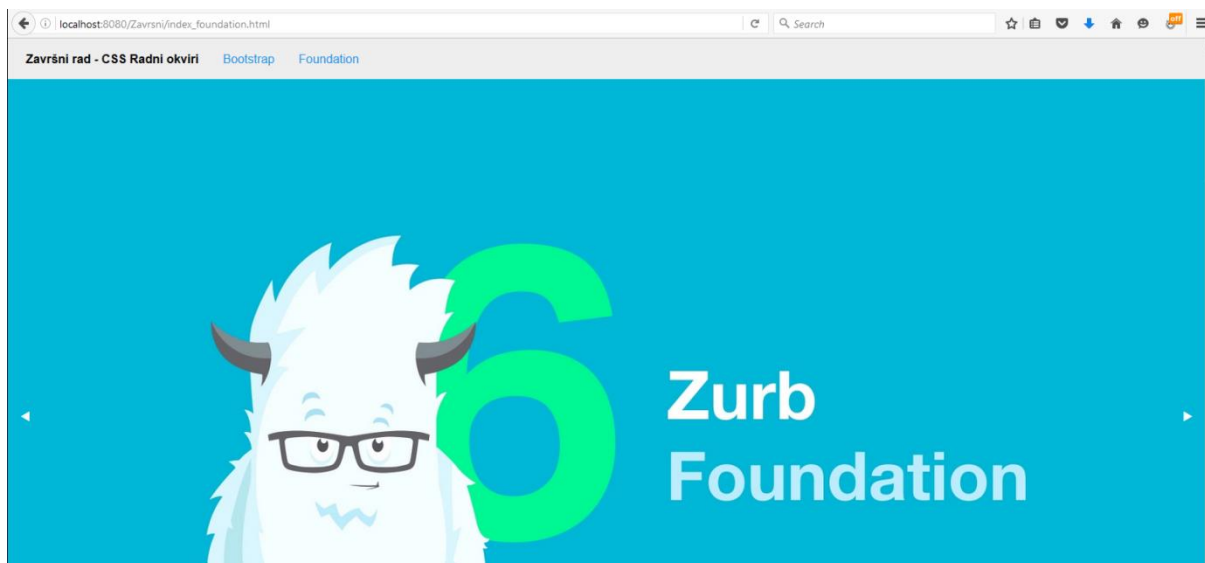
3.2. Foundation

Izrada stranica i uređivanje elemenata u *Foundationu* je nešto drugačija nego u kod drugih radnih okvira. Na službenoj stranici dostupni su kodovi svih elemenata, a postoji i grafičko sučelje u kojem je moguće ispisivati kod te pratiti promjene. Uz određene alate promjene se mogu pratiti u realnom vremenu te je izrada stranice mnogo lakša. Slika 35 prikazuje sučelje za pisanje i uređivanje koda.



Slika 35. Grafičko sučelje *foundationa* – uređivanje koda

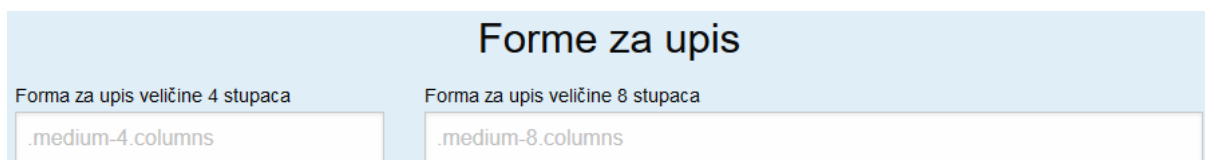
Sljedeća stranica napravljena je u radnom okviru *Foundation 6*. Praćen je sličan princip kao kod izrade sa *Bootstrapom*. Na samom vrhu stranice, u sklopu *headera*, nalazi se navigacijski izbornik sa poveznicama. Ispod *headera* stavljen je vrtuljak koji se u *Foundationu* naziva *orbit*. *Orbit* također koristi JavaScript kod pomoću kojeg je definiran okret svake slike. Koristeći *Foundation* postoji oko 10 različitih efekata pomoću kojih se slike mogu prikazati na vrtuljku što je velika prednost nad *Bootstrapom*. U *Foundationu* također postoji i koncept građevnih blokova (eng. *Building blocks*), preko kojeg postoji čak više vrsta *Orbita* koje se mogu implementirati kopiranjem i malom prilagodbom koda.



Slika 36. Header stranice sa caroseulom napravljen u Foundation radnom okviru

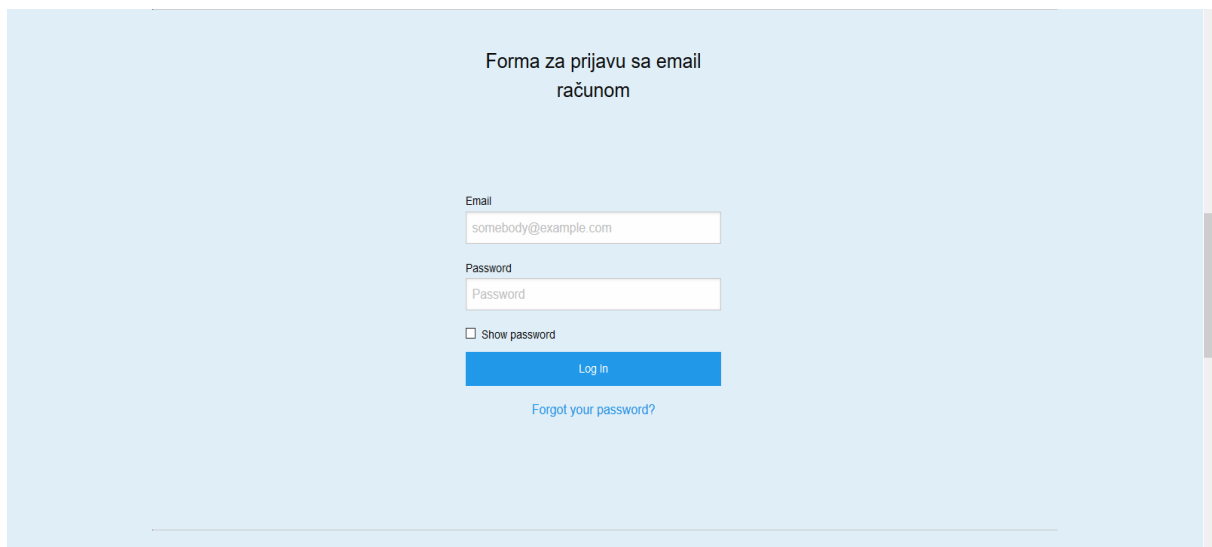
Navigacijski izbornik podijeljen je u 3 stupca od kojih svaki zauzima 2 *box elementa*, a postavljeni su da plutaju ulijevo. Slika 36 prikazuje *Header* sa vrtuljkom i navigacijskim izbornikom napravljenim u *Foundation* radnom okviru. Postoji način da se isti izbornik napravi i bez opcije plutanja (eng. *Float*), a to je da se selektoru dodaje posebno svojstvo poretka (eng. *order*). Tako je moguće unutar `<div>` elementa odrediti redoslijed prikaza pomoću selektora usprkos njihovom poretku u HTML dokumentu koji će na kraju biti pregažen.

Forme za unos definirane su brojem stupaca u retku. Primjer forme za upis prikazuje slika 37. Svaka forma zauzima jedan redak, a njena veličina je određena brojem *box elemenata*. Pa tako na web stranici postoje 2 forme za unos podataka od kojih je jedna veličine 4 stupaca, a druga 8 stupaca, što u konačnici odgovara broju stupaca u sustavu rešetki *Foundation* radnog okvira.



Slika 37. Forme za unos podataka

Foundation ima velik naglasak na svojim građevnim blokovima pa se tako među njima može naći velik broj formi za prijavu. Takve forme se koriste na poznatim društvenim mrežama, email servisima ili nekim drugim web aplikacijama. Slika 38 prikazuje formu za prijavu.



Slika 38. *Forma za email prijavu.*

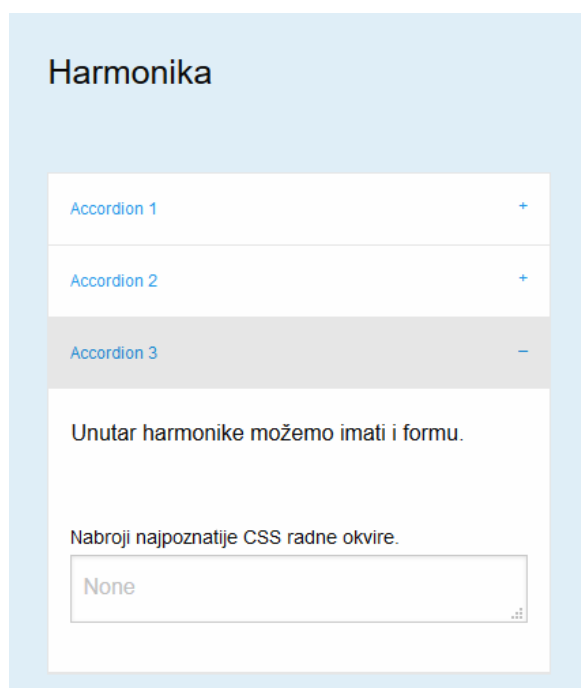
Foundation daje veliku pažnju sadržaju i na koji način se on prikazuje. Iz tog razloga u dostupnim kodovima moguće je pronaći jako velik broj okvira za prikaz sadržaja. Postoje okviri sa raznim tranzicijama, efektima, bojama i dodatnim značenjima. Okviri za prikaz sadržaja nalaze se na slici 39.



Slika 39. *Okviri za prikaz sadržaja.*

Klizne trake uključujemo unutar okvira za prikaz sadržaja te možemo odrediti posebna svojstva traci. Na primjer, mijenjanjem vrijednosti klizne trake također se mijenja veličina fonta u okviru. Klizne trake upotrebljive su i kao navigacija kroz tekst unutar nekog okvira. Postoje trake sa broječanim prikazom njihovih vrijednosti, a takve se koriste kao trake za podešavanje jačine zvuka u glazbenim sviračima.

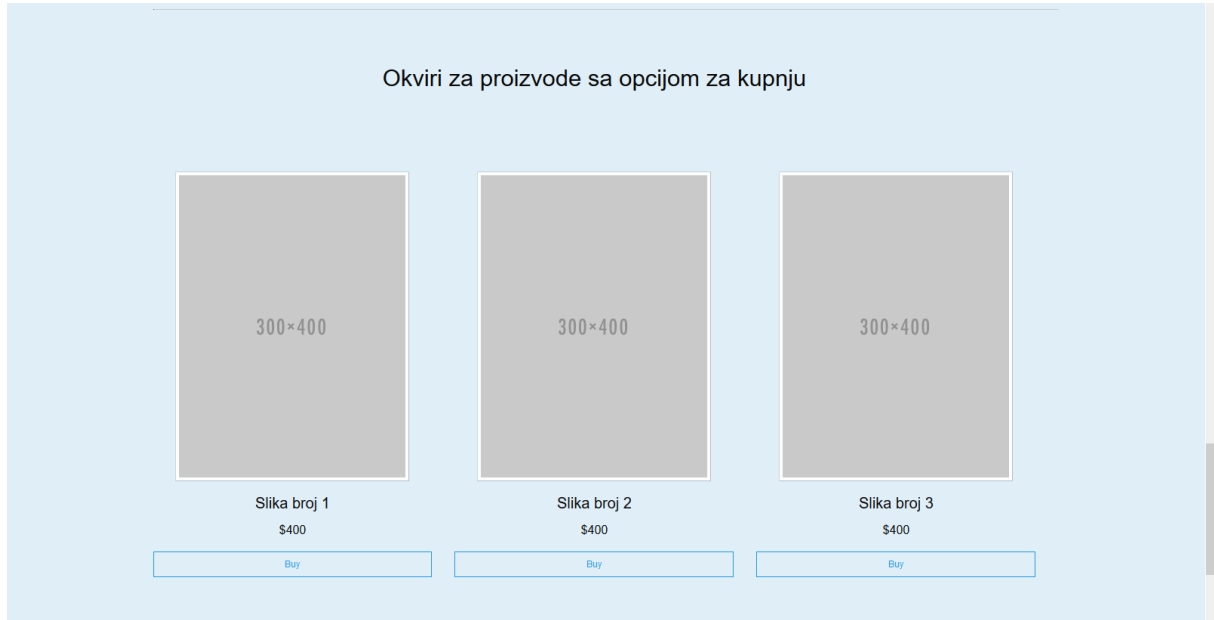
Harmonika sadržava tekst ili može imati dodatne informacije, forme ili gumbе. Najčešće se koristi kada je dostupan veći broj formi ili opcija. Harmonika animira otvaranje i zatvaranje dodatnog sadržaja. Primjer harmonike nalazi se na slici 40.



Slika 40. *Harmonika*

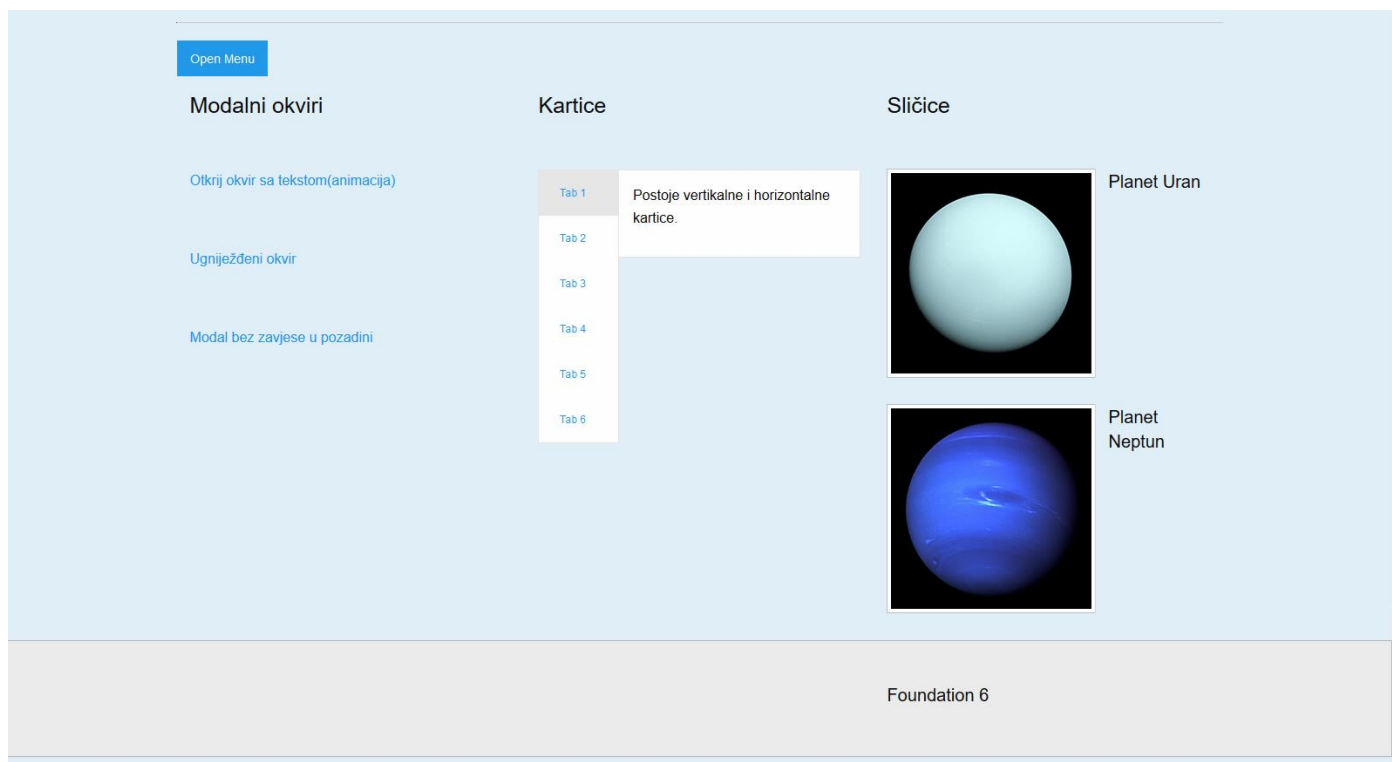
Korištenjem gumbi moguće je otvarati i zatvarati dijelove harmonike. Pritiskom na gumb počinje animacija kojom se zatvara aktivni dio harmonike, te se otvara odabrani dio.

Još neki od *Building blocks-a* su predlošci za proizvode u web-shopu. Okviri sa prostorom za sliku nekog proizvoda, cijenom i gumbom za potvrdu kupnje neki su od jednostavnijih primjera predložaka u Foundationu. Slika 41 prikazuje predložak online dućana sa slikama za proizvode.



Slika 41. Predložak proizvoda za web shop

Modalni okviri isto prenose i prikazuju sadržaj, ali ga isto tako obogaćuju sa tranzicijama, zavjesama i efektima. Okviri za prikaz teksta ili slika mogu imati lagani odsjaj, sjene u bojama te animacije tokom promjene veličine prikaza. Kartice (eng. *Tabs*) se dijele na vertikalne i horizontalne. U primjeru na slici 42 prikazane su vertikalne kartice. Koriste se kao navigacija kroz grupu određenih podataka, a funkcioniraju na principu linka. Odabirom željene kartice, zatvara se aktivna kartica te se otkriva sadržaj u njoj. Sličice (eng. *Thumbnails*) su okviri za prikaz slika. Manjih su dimenzija, imaju definirane rubove, a prelaskom mišem preko sličice pojavljuje se odsjaj oko ruba.



Slika 42. Modalni okvir, kartice i sličice.

Obrati li se pažnja na okvire, sustav rešetki, navigacijske izbornike i općenito dostupne predloške, razlike između *Bootstrapa* i *Foundationa* nisu velike. Foundation je svoje resurse usmjerio prema marketinškim agencijama, *online* dućanima, prema poznatim i velikim brandovima. Mnogo poznatih sportskih agencija, putničkih agencija, tvrtka iz automobilske industrije te svijeta zabave napravilo je svoju web stranicu pomoću *Foundationa*. Dostupne su neke funkcije jedinstvene za *Foundation* radni okvir. Osim izrade web stranica, moguća je izrada dinamičnih web aplikacija te responzivnih mailova. Repozitorij *Foundationa* sadrži mnogo opcija za prilagodbu aplikacija i stranica korisniku. Potrebe krajnjeg korisnika moguće je zadovoljiti sa skoro svih aspekata uz korištenje ovog radnog okvira. Osim građevnih blokova, u *Foundationu* su dostupni gotovi HTML predlošci, spremni za prilagodbu sadržaja.

Širok raspon responzivnih tablica, unaprijed izrađenih SVG ikona te tzv. matrica samo su neki od naprednih koncepata korištenja koji nisu dostupni kod niti jednog radnog okvira. Uzevši u obzir sve mogućnosti može se zaključiti da je *Foundation* izrazito napredan radni okvir te da uvelike premašuje opcije svojih suparnika.

4. ZAKLJUČAK

CSS radni okviri u velikoj mjeri pomažu web programerima. Oni omogućuju korištenje dodatnih opcija, ubrzavaju proces izrade, smanjuju volumen koda koji je napisan, ostavljaju prostor za nove ideje, a možda najvažnija prednost od svega je ta što se programer može posvetiti dizajnu i razradi novih ideja. Još jedna velika prednost CSS okvira je ta što ih ima mnogo. Svaki programer može istražiti pojedine okvire te se odlučiti za korištenje jednog od njih ovisno prema svojim željama, ali i zahtjevima klijenta.

Mogućnost promjene izvornog koda, dodavanja svojih ideja u već postojeće radne okvire dodaje novu dimenziju za web dizajn. Tehnologija se razvija, broj medija preko kojih posjećujemo internetski sadržaj raste velikom brzinom, te je uloga radnih okvira sve veća. Programerima koji rade na razvoju olakšan je posao zbog toga što je izvorni kod većine radnih okvira dostupan u javnosti, te sami korisnici pridonose razvoju istih.

Za kraj, može se reći da su CSS radni okviri donijeli revoluciju u web dizajnu te da će nastaviti ispunjavati zahtjevima tehnologije, tržišta i klijenata. Mogućnosti CSS okvira će se razvijati dalje te će uvijek biti novih ideja i dizajnerskih rješenja.

5. LITERATURA

1. <https://www.w3.org/Style/LieBos2e/history/> 05.08.2016.
2. <https://www.w3.org/Style/LieBos2e/enter/Overview.en.html> 05.08.2016.
3. <https://www.w3.org/TR/2012/REC-css3-mediaqueries-20120619/#media0> 05.08.2016.
4. <http://getbootstrap.com/> 22.08.2016.
5. <http://foundation.zurb.com/> 22.08.2016.
6. <http://zurb.com/about/profile/bryan-zmijewski> 23.08.2016.
7. <http://semantic-ui.com/introduction/getting-started.html> 22.08.2016.
8. <https://github.com/Semantic-Org/Semantic-UI> 30.08.2016.