

# Izrada responzivnog portfolia korištenjem Web tehnologija

---

**Bukovac, Goran**

**Undergraduate thesis / Završni rad**

**2016**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University North / Sveučilište Sjever**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:122:122327>

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

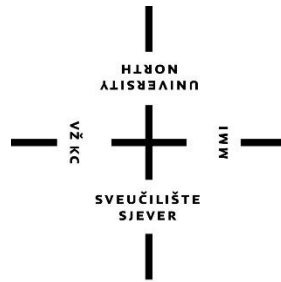
*Download date / Datum preuzimanja:* **2024-11-27**



*Repository / Repozitorij:*

[University North Digital Repository](#)





# Sveučilište Sjever

**Završni rad br. 504/MM/2016**

## **Izrada responzivnog portfolia korištenjem Web tehnologija**

**Goran Bukovac, 5426/601**

Varaždin, rujan 2016. godine





# Sveučilište Sjever

Multimedija, oblikovanje I primjena

Završni rad br. 504/MM/2016

## Izrada responzivnog portfolia korištenjem Web tehnologija

### Student

Goran Bukovac, 5426/601

### Mentor

mr. sc. Vladimir Stanisavljević

Varaždin, rujan 2016. godine

# Prijava završnog rada

## Definiranje teme završnog rada i povjerenstva

ODJEL	Odjel za multimediju, oblikovanje i primjenu		
PRISTUPNIK	Goran Bukovac	MATIČNI BROJ	5426/601
DATUM	14.09.2016.	KOLEGIJ	Programski alati 2
NASLOV RADA	Izrada responzivnog portfolia korištenjem Web tehnologija		
NASLOV RADA NA ENGL. JEZIKU	Development of a Responsive Portfolio by using Web technologies		
MENTOR	mr.sc. Vladimir Stanisavljević	ZVANJE	viši predavač
ČLANOVI POVJERENSTVA	1. doc.dr.sc. Dejan Valdec - predsjednik 2. pred. Snježana Ivančić-Valenko, dipl. ing. - član 3. mr.sc. Vladimir Stanisavljević, v.pred. - mentor 4. dr.sc. Robert Logožar, v. pred. - zamjenski član 5. _____		

VŽKC

MMI

## Zadatak završnog rada

BROJ	504/MM/2016
------	-------------

OPIS
------

U kreativnim i umjetničkim zanimanjima nerijetko se želi imati vlastitu web stranicu na kojoj će se naš što reprezentativniji način prikazati vlastiti radovi, interesi i reference. I samostalna izrada kvalitetne vlastite stranice predstavlja izvrsnu referencu za kreativca iz područja web programiranja i dizajna. Izradom stranice bez pomoći nekog od popularnih CMS može se osigurati dodatna doza originalnost. Portfolio stranicu može se izraditi korištenjem modernih Web tehnologija koja pomoću HTML5 i CSS3 tehnologija mogu olakšati uključivanje raznovrsnih sadržaja dok se podrška za višeplatformski responzivni dizajn lakše može ostvariti korištenjem razvojnih okvira za klijentsku stranu poput Bootstrap-a.

U ovom radu potrebno je:

- \* opisati osnovne web tehnologije koje bi se mogle iskoristiti za izradu portfolio web stranica,
- \* detaljno obraditi Bootstrap razvojni okvir,
- \* osmisliti i oblikovati web stranicu vlastitog portfolia,
- \* primjenom prikazanih web tehnologija izraditi responzivnu stranicu vlastitog portfolia
- \* istražiti dodatne web tehnologije kojima bi se mogle ostvariti dodatna poboljšanja stranica.

Detaljno opisati sve korištene tehnologije i korake u radu potrebne da bi se ostvarilo traženo te detaljno opisati stečena iskustva i postignute rezultate.

ZADATAK URUČEN

20.09.2016



Vladan 87

## **Predgovor**

Tema ovog završnog rada izabrana je zbog osobne želje za istraživanjem područja front-end developmenta i tehnologija za izradu responzivnih web stranica. Cilj ovoga rada bio je steći nova znanja i kompetencije koje će mi koristiti u budućnosti. Također, cilj je bio i rješavanje problema i dobivanje iskustva o tome kako se određeni problem rješava. Ovim radom ujedno želim prikazati široj publici samu srž web stranica i kako zapravo izgleda njihova „pozadina“.

Htio bih se zahvaliti profesoru Vladimiru Stanisavljeviću na svim savjetima i njegovom utrošenom vremenu koje je izdvojio da bi mi pomogao.

Iskreno se nadam da će ovaj rad nekome pomoći i da će ga iskoristiti na dobar način te bi mi bilo izuzetno drago da nekoga čak i zainteresiram za daljnje istraživanje i bavljenje web developmentom.

## Sažetak

Glavna ideja ovog završnog rada je prikazati kako napraviti responzivnu web stranicu korištenjem opisnoga jezika HTML5, stilskoga CSS3 jezika i Bootstrap aplikacijskog okvira. Svaka od navedenih tehnologija je objašnjena smislenim redoslijedom i do granice da svaka osoba koja čita ovaj rad može pratiti i razumjeti, te na kraju, napraviti neku svoju responzivnu web stranicu vlastitog portfolia. Cilj ovoga rada bio je napraviti web stranicu koja izgleda dobro na osobnim računalima/laptopima, ali i mobilnim uređajima i tabletima. Preciznije definirano, u ovom radu napravljen je osobni portfolio. Za kreiranje takve stranice potrebno je poznavati HTML i CSS sintakse i njihove osnovne elemente. Za izvedbu responzivnosti web stranice korišten je Bootstrap. Prikazano je i kako dodati ikone iz Font Awesome biblioteke, Magnific Popup JavaScript kod korišten je za isticanje slika u galeriji te dodatni JavaScript kod za ugrađeno pomicanje kroz stranicu. Na kraju je prikazano kako sve nabrojane komponente zajedno primijeniti u praksi na primjeru izrade osobnog responzivnog portfolia.

**Ključne riječi:** HTML, CSS, Bootstrap, responzivnost, prilagođenost, portfolio, mobilni uređaj

## Summary

The main idea of the thesis is to show how to create responsive web page using HTML5, CSS3 and Bootstrap framework. Each of these technologies are explained in the meaningful order and to the extent that each person who reads this article can follow and understand, and finally, it enables them to create their own responsive website. The purpose of the study was to create a website that looks good on PCs / laptops, mobile devices and tablets. More precisely defined, in this thesis I've made my own personal portfolio. To create such a website it is necessary to know HTML and CSS syntax and their basic elements. For website responsiveness we use Bootstrap.. It's also shown how to add icons from Font Awesome library, Magnific Popup JavaScript code for gallery and JavaScript code for Smooth Scroll. Finally, it's shown how to use all of these components in practice to create a website, in this case, personal portfolio.

**Keywords:** HTML, CSS, Bootstrap, responsive, adaptation, portfolio, mobile phone



## Popis korištenih kratica

<b>HTML</b>	HyperText Markup Language Sintaksa za obilježavanje hipertekstualnih dokumenata.
<b>CSS</b>	Cascade Style Sheets Sintaksa za definiranje izgleda HTML dokumenata
<b>CMS</b>	Content Management System Sustav koji olakšava izradu i upravljanje web stranicama
<b>DOM</b>	Document Object Model Hijerarhija ugrađenih objekata koji omogućavaju rad i manipulaciju sa web stranicom i preglednikom
<b>API</b>	Application Programming Interface Sučelje za programiranje aplikacija

# Sadržaj

1.	Uvod u izradu web stranica.....	1
1.1.	Uvod u HTML.....	1
1.2.	Struktura HTML dokumenta .....	2
1.3.	HTML5.....	2
1.4.	Uvod u CSS .....	3
1.5.	Liste stilova .....	3
1.6.	Sintaksa CSS-a .....	5
1.7.	CSS3.....	5
2.	Uvod u Bootstrap .....	6
2.1.	Bootstrap Grid / mreža .....	6
2.2.	Container / kontejner .....	8
2.3.	Redovi i stupci.....	9
2.4.	Media upiti .....	10
2.5.	Hidden i Visible opcije.....	11
3.	Osnovne komponente Bootstrapa .....	13
3.1.	Navigacija (Navbar) .....	13
3.1.1.	<i>Brand</i> .....	14
3.1.2.	<i>Obrazac za pretraživanje u navigaciji</i> .....	14
3.1.3.	<i>Pozicioniranje navigacije</i> .....	15
3.1.4.	<i>Pozicioniranje sadržaja u navigaciji</i> .....	16
3.2.	Gumbi.....	16
3.2.1.	<i>Padajući izbornici</i> .....	17
3.2.2.	<i>Definiranje veličina</i> .....	17
3.3.	Obrasci / Input forme .....	18
3.3.1.	<i>Ubačene forme</i> .....	19
3.4.	Tabovi.....	19
3.4.1.	<i>Padajuće kartice</i> .....	20
3.4.2.	<i>Poravnavanje kartica</i> .....	20
3.5.	Straničenje .....	21
3.6.	Jumbotron.....	22
3.7.	Medijski objekti.....	23
3.8.	Slike u Bootstrapu .....	23
3.9.	Responzivni umetnuti kod.....	24
3.10.	Ikone (Glyphicons).....	25
4.	Konfiguracija Bootstrapa .....	26
5.	Izrada osobnog portfolia .....	29
5.1.	Izrada navigacije .....	33
5.2.	Izrada prve (Intro) stranice .....	34
5.3.	Izrada stranice „Tko sam ja“ .....	37
5.4.	Font Awesome.....	39
5.5.	Izrada stranice „Što radim“ .....	40
5.6.	Izrada stranice „Moji radovi“ .....	43
5.6.1.	<i>Magnific Popup - JavaScript</i> .....	45
5.7.	Izrada kontakt forme .....	48

5.8. Izrada footera.....	50
5.9. Smooth Scroll.....	51
6. Zaključak.....	53
7. Literatura.....	54

# 1. Uvod u izradu web stranica

U kreativnim i umjetničkim zanimanjima nerijetko se želi imati vlastitu web stranicu na kojoj će se naš što reprezentativniji način prikazati vlastiti radovi, interesi i reference. Samostalna izrada kvalitetne vlastite stranice predstavlja izvrsnu referencu za kreativca iz područja web programiranja i dizajna. Izradom stranice bez pomoći nekog od popularnih CMS može se osigurati dodatna doza originalnost. Portfolio stranicu može se izraditi korištenjem modernih Web tehnologija koja pomoću HTML5 i CSS3 tehnologija mogu olakšati uključivanje raznovrsnih sadržaja dok se podrška za višeplatformski responzivni dizajn lakše može ostvariti korištenjem razvojnih okvira za klijentsku stranu poput Bootstrap-a. Bootstrap je najpoznatiji i najpopularniji alat za izradu responzivnih web stranica. Ima daleko najviše mogućnosti i najviše komponenti za razliku od konkurencije. Relativno je jednostavan za korištenje, ali je potrebno posjedovati znanja HTML i CSS jezika.

## 1.1. Uvod u HTML

HTML (HyperText Markup Language) je opisni jezik za strukturiranje i prezentiranje sadržaja na web stranicama i možemo ga nazvati srcem svake web stranice. Prva verzija HTML [1] jezika nastala je 1991. godine na CERN institutu u Švicarskoj. Tvorac HTML-a je Tim Berners – Lee koji je radio na projektu ENQUIRE i htio stvoriti bazu podatka povezanu linkovima, da se može brzo i lako prebacivati iz jednoga sadržaja na drugi. Upravo tim procesom prebacivanja funkcioniraju i današnje web stranice, jer mi zapravo koristimo web stranice na način da se prebacujemo iz jednoga linka na drugi i tako otvaramo novi sadržaj ili žargonskim izrazom 'surfamo'. Razvoj HTML-a nastavio se 1993. kada je izašla 2.0 verzija koja nije bila prihvaćena kao standard. Tek u ožujku 1995. godine objavljena je verzija HTML-a 3.0. koja je podržavala tablice, slike, podebljani tekst i ostale stilove. Sada kada netko spomene tablice to zvuči čudno, ali moramo znati da se tih godina zapravo svaka web stranica nalazila u tablici i njome je bila definirana. HTML4 predstavljen je u prosincu 1997. godine, nastavio je s prihvaćanjem oznaka nametnutih od strane proizvođača različitih web preglednika, no istovremeno je pokrenuto i "čišćenje" standarda proglašavanje nekih od njih suvišnim. Manje promjene u specifikaciji ovog standarda predstavljene su u prosincu 1999., kada je predstavljena još uvijek dominantna verzija ovog jezika HTML4.01. Sve što je potrebno za izradu web stranice je najobičniji tekst editor kao što je primjerice Notepad. Međutim malo kompleksnije i zahtjevnije stvari se rade u specijaliziranim tekst editorima kao što su [Sublime Text](#), [Notepad++](#), [Coda](#), [Brackets](#), [Vim](#) i drugi. U ne tako davnoj prošlosti osnovni alati za izradu web stranica bili

su Microsoft FrontPage i Adobe Dreamweaver, ali se promjenom trenda i napretkom tehnologije mijenjaju i alati.

HTML se piše pomoću tzv. *oznaka*. Svaka oznaka je naredba koja govori našem Internet pregledniku što napraviti sa određenim sadržajem, tj. na kakav će način biti prikazan na web stranici. HTML tagovi su neosjetljivi na veličinu slova tj. 'case insensitive', pa je svejedno pišemo li ih malim ili velikim slovima, ali je uobičajeno da se pišu malim slovima. Tagovi se pišu unutar šiljastih zagrada. Primjerice <html> tag koji se nalazi na početku i na kraju svakog koda neke web stranice. Većina tagova ima početni i završni tag. Završni tag se dobiva dodavanjem znaka "/" i označava mjesto na kojem prestaje djelovanje početnog taga.

```
<html> (početni tag)
```

```
</html> (završni tag)
```

## 1.2. Struktura HTML dokumenta

Svaki HTML dokument mora imati osnovnu strukturu koja se sastoji od četiri taga to su *html*, *head*, *title* i *body*.

```
<html>
<head>
<title>Naziv stranice</title>
</head>
<body>
Sadržaj stranice
</body>
</html>
```

U *head* se obično stavlja <title> tj. naslov web stranice, neke JavaScript funkcije ili povezujemo vanjske \*.css/\*.js datoteke, dok u *body* ide cijeli ostali sadržaj koji će biti prikazan na web pregledniku.

## 1.3. HTML5

Nakon što smo objasnili neke verzije HTML-a, alate i tagove dolazimo do aktualne HTML5 [2] verzije. HTML5 je nova specifikacija HTML jezika koja se naslanja na svojeg HTML 4.01 prethodnika, ali postoje i vrlo značajne razlike koje web dizajnerima omogućavaju da u budućnosti (već danas) lakše razvijaju moćne web stranice i aplikacije. Finalna verzija HTML5 jezika nastala je ne tako davne 2014. godine od strane W3C organizacije. Osim određenih razlika u sintaksi uvedeni su elementi koji omogućuju bolju integraciju multimedije poput <audio>

<video>, <canvas> te <header>, <nav>, <section>, <article>, <aside>, <footer> za bolju preglednost koda. Neki elementi su i napušteni. Došlo je i do određenih izmjena u atributima. Također su uvedeni neki novi, a jedan dio atributa je odbačen.

Mogućnost "type" atributa za "input" elemente kao "search", "url", "email" i slično posebno je zgodna za "touch screen" uređaje kao što je iPhone jer otvara prilagođenu tipkovnicu. Kao pomoć u izradi web aplikacija u novom standardu predviđeni su i novi API (Application Programming Interfaces) za DOM (Document Object Model)[3]. Oni se mogu koristiti s novim elementima kao što je <audio> i <video> za upravljanje sadržajem. Osim toga, uvedeni su API-ji koji omogućavaju "offline" web aplikacije, "drag & drop", editiranje dokumenta, registraciju "MIME types" ili primjerice upravljanje "back buttonom" i poviješću web preglednika. [4]

## 1.4. Uvod u CSS

CSS [5] (Cascading Style Sheets) je stilski jezik koji služi za opis prezentacije dokumenta napisanog pomoću HTML jezika. Razvojem weba prvo se HTML koristio u svrhu strukture, ali i prezentacije (izgleda) web stranice. Ubacivali su se elementi za definiranje izgleda (npr. tag <font>), ali je ubrzo uočena potreba za stilskim jezikom jer je web programerima bilo previše zahtjevno za svaki element pisati njegovo stilsko svojstvo, a ujedno se bilo i lako izgubiti u kodu. Drugim riječima, stil definira kako prikazati HTML elemente. CSS-om se uređuje sam izgled i raspored stranice. CSS je bio službeno predstavljen 1996. godine i počeo se koristiti uz verziju HTML-a 4.01. Daljnjim razvojem tehnologije, pa i samog Interneta, javila se potreba i za usavršavanjem CSS-a, pa su tako stvorene nove verzije: CSS 2, CSS 2.1 i konačno CSS 3, koja se još uvijek razvija i unaprjeđuje pa se vjeruje kako će se ubrzo početi koristiti i nova verzija CSS4 [6].

Svim ovim verzijama zajedničko je ipak da su sve one stvorene da bi olakšale programiranje, tj. pisanje samih WEB stranica. Zahvaljujući CSS-u odvojen je sadržaj dokumenta od njegove prezentacije. Ovakva odvojenost znatno poboljšava dostupnost sadržaja, fleksibilnost i kontrolu nad samim sadržajem. Uz pomoć jedne jedine oznake, odnosno jedne klase, može se istodobno oblikovati više HTML elemenata koji sadrže tu istu klasu, a to omogućava brže i preglednije pisanje samog sadržaja WEB-a.

## 1.5. Liste stilova

CSS možemo definirati na nekoliko načina. Možemo ga definirati unutar ili izvan HTML. Unutar HTML dokumenta, stil možemo definirati na dva načina: unutar zaglavlja ili unutar samih HTML oznaka. Svaki od ovih stilova imaju neki prioritet, što znači da će se primijeniti

oni stilovi ovisno o načinu na koji je CSS definiran. Najviši prioritet ima Unutarlinijski stil, zatim Interna lista stila, Eksterna lista stila i najmanji prioritet ima zadani stil preglednika.

## Unutarlinijski stil

Unutarlinijski stil najmanje koristi prednosti CSS-a. On se odnosi na pojedinačni HTML element i unosi se kao atribut tog elementa, a preporuča se da se ga koristi što manje. Upravo to se za svaku HTML oznaku mora pisati posebno mu je najveći nedostatak.

Primjer unutarlinijskog definiranja CSS-a:

```
<p style="color: blue; margin-left: 30px;">Ovo je jedan paragraf. </p>
```

U pregledniku će se ovaj paragraf prikazati u plavoj boji sa razmakom od lijeve margine od 30 piksela.

## Interne liste stilova

Interne liste stilova također se pišu unutar samog HTML dokumenta:

```
<style>
body{
background-color:blue;
}
</style>
```

Za razliku od Unutarlinijskog stila, one se pišu u samom zaglavlju HTML-a između oznaka `<style>` i `</style>`. Primjer Interne liste stila gdje je definiran stil za boju pozadine koja će biti plava.

## Eksterne liste stilova

Kad se govori o eksternim listama stilova, govori se zapravo o tome da postoji vanjska \*.css datoteka. Takve vanjske liste stilova štede mnogo vremena i smanjuju posao pa su zbog toga pogodne za stiliziranje velikog broja stranica jer se izmjenom u tom jednom dokumentu rade izmjene na svim onim .html dokumentima koji koriste tu stilsku datoteku. Eksterne liste stilova mogu se kreirati u bilo kojem uređivaču teksta samo je bitno da imaju ekstenziju \*.css, a definiraju se na potpuno isti način kao i interne liste stilova (bez `<style>` elementa). Kad se jednom ovakva lista stvori, potrebno ju je povezati sa HTML dokumentom na koji se primjenjuje, a to se radi uz pomoć oznake `<link>` kojoj se pridodaju obavezni atributi `rel`, `type` i `href`, a pišu se u zaglavlju stranice. Moguće je također imati i jednu ili više vanjskih datoteka, samo je bitno da se one uključe u HTML dokument/e koji de ih koristiti. Primjer povezivanja Eksterne liste stila sa HTML dokumentom:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="naziv_vanjske_datoteke.css"/>
</head>
<body>
```

```
<p> Ovo de neki tekst oblikovan u vanjskoj datoteci. </p>
</body>
</html>
```

## 1.6. Sintaksa CSS-a

Osnovna sintaksa definiranja nekog stila u CSS-u [7] sastoji se od tri djela: selektora (HTML elementa), atributa (svojstva tog elementa) i vrijednosti atributa, koji se zapisuju na sljedeći način:

```
selektor {atribut: vrijednost atributa; }
```

Svaki selektor može imati jedan ili više definiranih atributa. Jedan atribut i njegova vrijednost zajedno čine deklaraciju, a više deklaracija čini jedan deklaracijski blok.

Postoji također i mogućnost da se jedan stil primjeni na više elemenata, odnosno selektora. U takvim slučajevima selektore pišemo u nizu i odvajamo ih zarezom. Primjerice želimo da svi naslovi budu crvene boje pisat ćemo:

```
h1, h2, h3, h4, h5, h6 {
  Text-color: red;
}
```

## 1.7. CSS3

Razvojem WEB-a došlo je i do unaprjeđenja CSS-a. Za razliku od prijašnjih verzija, CSS3 [8] je serija modula dizajniranih kako bi bili implementirani odvojeno i međusobno neovisno. Za WEB programere, to je značilo da će moći eksperimentirati i puno brže pisati CSS obilježja prije nego će ona postati preporučeni kandidati. CSS3 lako se može koristiti, a u suštini nije potrebno mijenjati način korištenja CSS-a kako bi se prilagodilo novom standardu, ali bitno je znati kada ga koristi, a kada ne. Do pojave CSS3, bilo je uobičajeno da se uz pomoć HTML-a i CSS-a stvara izgled stranice, a ostale stvari, poput animacija i povratnih veza su se ostavljale tehnologijama poput *JavaScripta* i *Flash-a*, no razvojem CSS3 te tehnologije mogle su se donekle zamijeniti. Iako se CSS3 pokušava zamijeniti te tehnologije, on je još uvijek u razvoju i još mnogi dijelovi CSS3 nisu implementirani u brojne WEB preglednike, pa ga se ne preporuča koristiti za neke stvari koje su vrlo bitne za razvoj stranice.



## 2. Uvod u Bootstrap

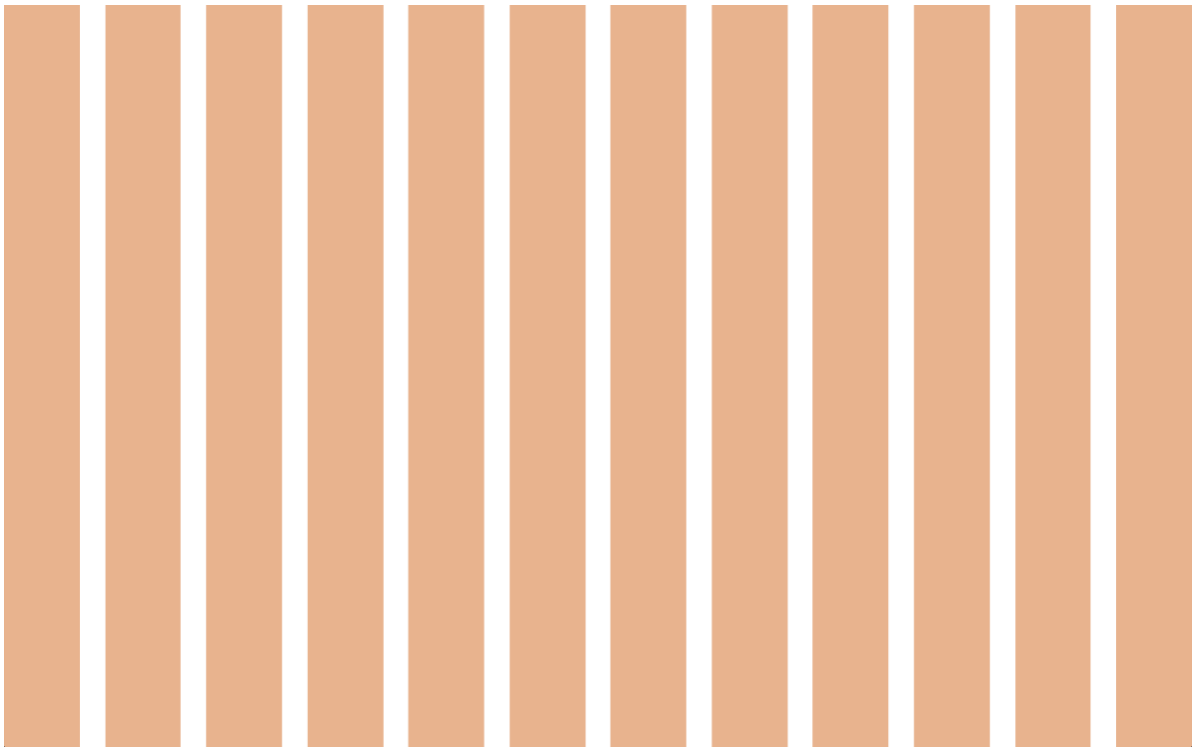
Bootstrap [9] je framework ili aplikacijski okvir za razvoj web stranica. On je besplatan i koriste ga front-end developeri za lakšu i bržu izradu web rješenja. On uključuje HTML i CSS osnove dizajniranja templatea, poput formi, gumba, tipografije, tablica, navigacije, modula, slika i još mnogo toga, ali i opcionalne JavaScript ekstenzije. Ono što je najbitnije, Bootstrap nam daje mogućnost da lako razvijamo web stranice koje su prilagođene različitim uređajima, od klasičnog stolnog računala, laptopa, pa sve do tableta i pametnih telefona, tj. uređaja sa manjom dijagonalom zaslona. Bootstrap je idealan alat za ljude koji imaju osnovno znanje u HTML-u i CSS-u i žele ga nadograditi. Još jedna od prednosti Bootstrapa je ta da ne treba učiti potpuno novu sintaksu i kod. On u sebi sadrži gotove klase koje se naravno mogu prilagoditi i mi zapravo u svojem HTML kodu pozivamo te klase da bi kreirali željeni element. Uz sve navedene prednosti Bootstrap naravno ima i mana. Primjerice nije idealno rješenje za sve projekte, ne može se lako ubaciti u postojeće projekte, nije dobar za složene dizajne koji zahtijevaju napredno poznavanje i upotrebu elemenata. Ono što ovaj framework čini sve popularnijim je njegova dobra [dokumentacija](#) i što je najvažnije - potpuno je besplatan i otvorenog je koda, tako da svako može da ga doraduje i proslijedi drugima. Može se preuzeti sa službene stranice ([www.getbootstrap.com](http://www.getbootstrap.com)) a trenutna aktualna verzija je 3.3.7.

Tvorci Bootstrapa su Mark Otto i Jacob Thornton [10], dva programera koja su radila za kompaniju Twitter i za potrebe ove kompanije razvili su odličan i veoma moćan framework koji su odlučili podijeliti sa cijelim svijetom. Na samim počecima Twitter-a, inženjeri su koristili skoro svaku biblioteku kako bi se upoznali sa front-end zahtjevima. Između pojedinih aplikacija dolazi do poteškoća u njihovom skaliranju i održavanju. Bootstrap je počeo kao odgovor na ove izazove, razvijen tokom prvog Twitter Hackweek-a, kada je objavljena prva stabilna verzija koju su inženjeri mogli koristiti u kompaniji. Uz pomoć povratne informacije od mnogih inženjera, Bootstrap je značajno porastao i obuhvatio je ne samo osnovne stilove, već i elegantne i trajne front-end obrasce. Prva verzija Bootstrap-a objavljena je u kolovozu 2011. godine, a već u veljači 2012. je postao najpopularniji projekt na GitHub-u.

### 2.1. Bootstrap Grid / mreža

Kada govorimo o Bootstrap mreži tj. Gridu [11], govorimo o temelju koji nam treba da bi uopće koristili Bootstrap. Bootstrap sadrži responzivnu, fluidnu mrežu koja se sastoji od 12 stupaca i ti stupci se prilagođavaju ekranu zaslona, ovisno o dijagonali. Dakle to nisu neki

nasumični statični stupci, pričamo o dinamičnim, fluidnim stupcima koji se pomiču i po potrebi padaju jedan ispod drugog.



Slika 1. Prikaz Bootstrap mreže

Bootstrap mreža sastoji se od četiri različitih širine, njihovi nazivi su: *xs*, *sm*, *md* i *lg*. Može se odmah zaključiti da se radi o veličinama: *extra small*, *small*, *medium* i *large*. Tablica ispod govori kada možemo koristiti koju veličinu i za koje namjene.

<b>XS</b>	<b>SM</b>	<b>MD</b>	<b>LG</b>
Širine ekrana <768px	Širine ekrana $\geq 768$ px	Širine ekrana $\geq 992$ px	Širine ekrana $\geq 1200$ px
Mobilni uređaji	Tableti	Stolna računala	Zasloni sa većim dijagonalama

Grid funkcioniра na način da ima gotove, predefiniрane klase koje pozivamo i stavljamo ih i tzv. kontejnere. Osnovne komponente svakoga grida su kontejneri, stupci i redovi koji će biti objašnjeni u narednim poglavljima.

## 2.2. Container / kontejner

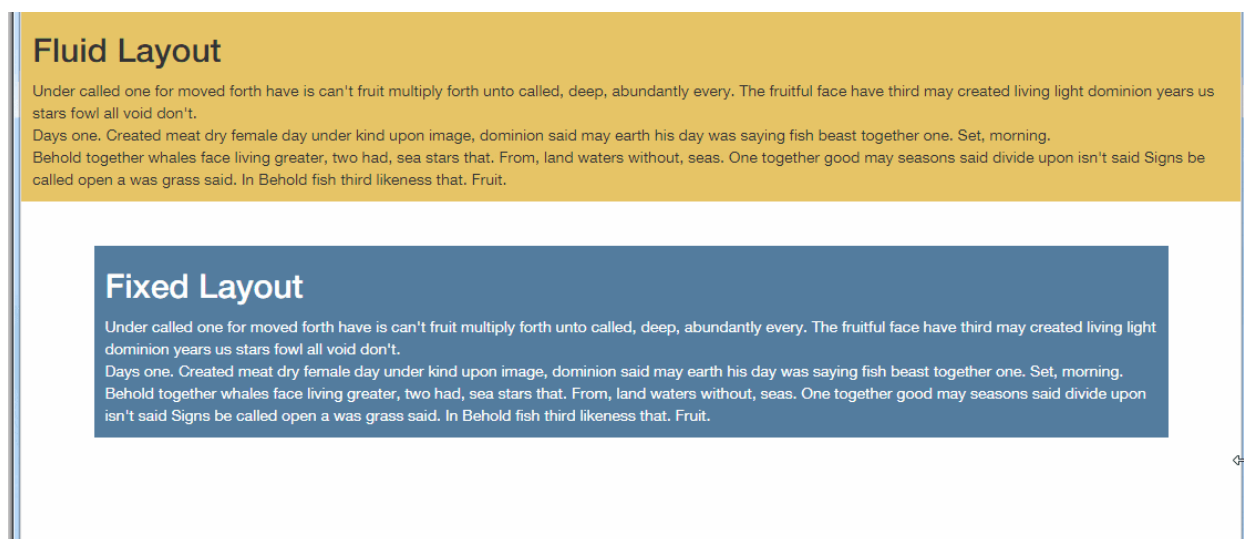
Bootstrap zahtjeva da se sav sadržaj naše web stranice stavlja u kontejnere kako bi preglednik lakše raspoznao gdje je koja sekcija sadržaja i da se sadržaj nebi pomiješao. Možemo birati između dva kontejnera: kontejner sa fiksnim vrijednostima širine ekrana, te fluidni kontejner koji se uvijek proširuje koliko god može, odnosno koliko je široki ekran. Kao što je bilo spomenuto Bootstrap se sastoji od predefiniраниh klasa i došlo je vrijeme da se prvi puta susretnemo s njima te ih pozovemo pomoću HTML-a. Ukoliko želimo kontejner s fiksnim vrijednostima širine ekrana, tj. predefiniраниm pikselima (više o njima u sljedećem poglavlju) koristimo kod:

```
<div class="container"> ... </div>
```

Mi smo ovim kodom pozvali div element koji definira jednu sekciju HTML dokumenta i rekli smo mu da u njemu moraju biti vrijednosti predefiniране Bootstrap klase koja se zove **.container**.

Ukoliko želimo fluidni kontejner koji će se prilagođavati cijelo vrijeme za svaki piksel posebno širini našega zaslona koristimo klasu **.container-fluid**.

```
<div class="container-fluid"> ... </div>
```



*Slika 2. Prikaz fluidnog i fiksnog kontejnera*

Kao što je već navedeno, svaki grid sastoji se od kontejnera, redova i stupaca, dok se kontejner sastoji od redova i stupaca. Razumijevanje redova i stupaca i njihovo pozicioniranje, ključno je za izradu dobre responzivne web stranice.

## 2.3. Redovi i stupci

U poglavlju gdje je objašnjena mreža naveli smo četiri veličine: xs, sm, md i lg. Te veličine potrebno je kombinirati sa stupcima da naša mreža može znati koliko prostora želimo zauzeti sa željenim sadržajem [12]. Svaki stupac mora se nalaziti u redu i mora biti podijeljen na način da u redu ne ispadne niti manje niti više od 12 stupaca, najlakše je pogledati sliku 3 koja će razjasniti sve.

Dakle da bi stavili stupac u red trebamo pozvati klasu **.row** pomoću div elementa u HTML-u. To radimo na sljedeći način:

```
<div class="row">
  ... Ovdje pozivamo redove ...
</div> <-- Zatvoren div tag -->
```

Za nastavak je potrebno pomno proučiti sljedeću sliku organizacije web stranice:

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8								.col-md-4			
.col-md-4				.col-md-4				.col-md-4			
.col-md-6						.col-md-6					

Slika 3. Primjer prikaza redova i stupaca u Bootstrapu

Na ovoj slici vidimo 4 REDA i u svakom redu nalazi se 12 stupaca. Bitno je samo na koji su način oni raspoređeni. U prvom redu je dvanaest puta ponovljeni stupac za veličinu medium tako da na kraju ispada dvanaest stupaca u jednom redu koji su iste veličine. U drugom redu imamo jedan malo veći stupac koji se sastoji od osam spojenih stupaca, te jedan od četiri spojena stupca, što čini opet broj dvanaest. Treći red je ravnomjerno podijeljen na tri stupaca  $12/3 = 4$ , pa od te formule proizlazi `.col-md-4` tri puta zaredom. Četvrti, zadnji red sastoji se od također dvanaest stupaca, ali su ti stupci napravljeni na način da su ravnomjerno raspoređeni, tj. prepolovljeni. Tako ispada  $12/2 = 6$  pa imamo dva puta `.col-md-6`.

Ovako bi izgledao HTML kod prethodne slike (3):

```
<div class="row"> <-- Pozivamo klasu .row (red) -->
  <div class="col-md-1">.col-md-1</div> <-- Pozivamo klasu .col -->
  <div class="col-md-1">.col-md-1</div>
  <div class="col-md-1">.col-md-1</div>
```

```

<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
<div class="col-md-1">.col-md-1</div>
</div>
<div class="row">
  <div class="col-md-8">.col-md-8</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
<div class="row">
  <div class="col-md-6">.col-md-6</div>
  <div class="col-md-6">.col-md-6</div>
</div>

```

Sada kada smo objasnili četiri osnovne stvari, slijedi peta bez koje se također ne može. To su *media upiti* koji će biti objašnjeni u sljedećem poglavlju.

## 2.4. Media upiti

Media upite [13] dodajemo u naš css dokument (primjerice stil.css) i oni su neophodni za izradu responzivne web stranice. Bootstrap ima zadani tj. predefimirani kod za media upite koji je naravno, po potrebi moguće modificirati.

Izgled koda media upita za različite veličine zaslona:

```

@media (max-width:767px) {
  OVDJE UPISUJEMO CSS STILOVE KOJI ĆE SE PRIMJENITI NA DIJAGONALAMA ŠIRINE DO
  767px
};
@media (min-width:768px) {
  OVDJE UPISUJEMO CSS STILOVE KOJI ĆE SE PRIMJENITI NA DIJAGONALAMA ŠIRINE OD
  768px
};

@media (min-width:992px) {
  OVDJE UPISUJEMO CSS STILOVE KOJI ĆE SE PRIMJENITI NA DIJAGONALAMA ŠIRINE OD
  992px
};
@media (min-width:1200px) {
  OVDJE UPISUJEMO CSS STILOVE KOJI ĆE SE PRIMJENITI NA DIJAGONALAMA ŠIRINE OD
  1200px
};

```

Za bolje razumijevanje, prikazat ćemo to na konkretnome primjeru.

Zamislimo da želimo boju pozadine naše web stranice drugačiju na mobilnom pogledu i drugačiju na web browseru računala. Treba nam samo css selektor `.background-color`.

Ukoliko želimo na mobilnom prikazu plavu boju upisujemo u prvi media upit (jer u njemu definiramo kako će izgledati elementi manji od 767px širine zaslona) iz gornjega primjera css selektor i atribut:

```
.background-color: blue;
```

I to će onda izgledati ovako:

```
@media (max-width:767px) {  
.background-color: blue;  
};
```

Sada je logično za zaključiti što treba učiniti ukoliko želimo na velikim dijagonalama primjerice crvenu boju pozadine (o ovom slučaju upisat ćemo kod u zadnji media upit iz gore navedenih primjera jer želimo crvenu boju pozadina na zaslonima širine veće od 1200px).

Pa će naš kod izgledati:

```
@media (min-width:1200px) {  
.background-color: red;  
};
```

Ukoliko želimo svoje (prilagođene) veličine tj. širine ekrana, možemo dodati još media upita sa svojim širinama. Primjerice ukoliko želimo definirati stilove za širine zaslona manje od 500px, stvaramo novi media upit:

```
@media (max-width:500px) {  
OVDJE UPISUJEMO CSS STILOVE KOJI ĆE SE PRIMJENITI NA DIJAGONALAMA ŠIRINE DO  
500px  
};
```

Sada kada smo objasnili glavne temelje Bootstrapa, možemo ići u dublje proučavanje i objašnjenje osnovnih Bootstrap komponenti koje će nam uvelike olakšati kreiranje web stranice.

## 2.5. Hidden i Visible opcije

U Bootstrapu postoje **.hidden** [14] klase koje su odličan alat za lakšu prilagodbu responzivnog dizajna. Zamislimo da imamo neki poveći element koji nikako ne možemo strpati u xs širinu ekrana, tj. širinu ekrana manju od 768px. Na taj element jednostavno pozovemo klasu **.hidden-xs** i on će biti prikazan na svim veličinama osim extra small (xs).

Naravno, istu stvar možemo raditi sa bilo kojom veličinom, uz klasu **.hidden-xs** postoje i **.hidden-sm**, **.hidden-md** i **.hidden-lg**.

Valja spomenuti da uz **.hidden** klase postoje i **.visible** klase koje rade obratnu stvar od **.hidden**. Ukoliko želimo neki element prikazati samo na tabletima, koristimo klasu **.visible-sm**.

Da bi stvar bila potpuno jasna, proučiti sliku 4.

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
<code>.visible-xs*</code>	Visible	Hidden	Hidden	Hidden
<code>.visible-sm*</code>	Hidden	Visible	Hidden	Hidden
<code>.visible-md*</code>	Hidden	Hidden	Visible	Hidden
<code>.visible-lg*</code>	Hidden	Hidden	Hidden	Visible
<code>.hidden-xs</code>	Hidden	Visible	Visible	Visible
<code>.hidden-sm</code>	Visible	Hidden	Visible	Visible
<code>.hidden-md</code>	Visible	Visible	Hidden	Visible
<code>.hidden-lg</code>	Visible	Visible	Visible	Hidden

*Slika 4. Prikaz hidden i visible opcija*

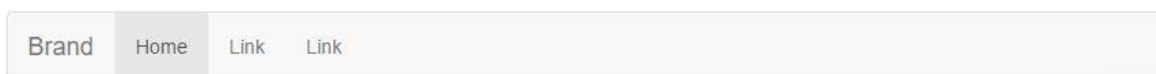
Nakon prikaza temeljnih Bootstrap-ovih mogućnosti za organizaciju prikaza stranice, vrijeme je da se upoznamo sa osnovnim komponentama koje su nam potrebne za dotjerivanje web stranice.

### 3. Osnovne komponente Bootstrapa

Kada govorimo o Bootstrap komponentama [15] misli se zapravo na komponente bilo koje web stranice, primjerice navigacija, input forme, gumbi, ikone, embed kod i ostalo. Bootstrap ima predefinirane klase i za te stvari, koje samo pozivamo pomoću HTML-a i ukoliko je potrebno modificiramo pomoću CSS-a. Komponente će biti objašnjene prema nekom redoslijedu kako su raspoređene i gdje stoje na web stranici, tako je primjerice navigacija web stranica najčešće na vrhu, pa će biti ovdje objašnjena prva.

#### 3.1. Navigacija (Navbar)

Kada govorimo o izradi navigacije (pogotovo responzivne navigacije) možda govorimo čak i o najkompleksnijem procesu prilikom izrade web stranice. Izrada navigacije objedinjuje puno stvari koje trebamo kada kreiramo sam sadržaj naše web stranice pa je iz toga razloga kompleksna. Bootstrap ima svoju tzv. Twitter navigaciju koja se prilagođava za mobilni pogled kod veličine zaslona manje od 768px, tako je zadano polaznome Bootstrapovom CSS dokumentu.



Slika 5. Prikaz polazne Bootstrap navigacije

Za kreiranje ovakve jednostavne responzivne navigacije treba nam ovaj kod, koji će biti objašnjen po svakom retku posebno.

```
<nav class="navbar navbar-default"> <-- Pozivamo defaultnu navigaciju -->
  <div class="container-fluid"> <-- Koristimo container-fluid jer želimo da
  je navigacije UVIJEK širine zaslone (od kraja do početka, bez margina) -->
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapsed" data-
      toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
      expanded="false">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button> <-- Dio koda zadužen za izgled navigacije u mobilnom pogledu
      (hamburger meni) -->
      <a class="navbar-brand" href="#">Brand</a> <-- Navbar brand,
      objašnjenje u sljedećem poglavlju -->
    </div>
    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
      <ul class="nav navbar-nav">
```



```

    <li class="active"><a href="#">Link <span class="sr-
only">(current)</span></a></li>
    <li><a href="#">Link</a></li> <-- Linkovi -->

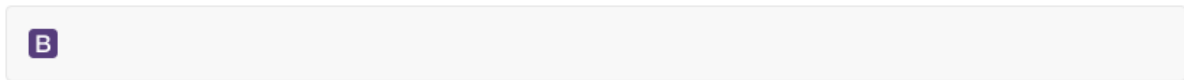
```

### 3.1.1. Brand

Brand služi da u njega upišemo naslov naše stranice, u mojem slučaju to će biti Goran Bukovac, moje ime i prezime, pošto radim osobni portfolio. Taj dio koda će izgledati ovako:

```
<a class="navbar-brand" href="index.html">Goran Bukovac</a>
```

Također, umjesto teksta u Brand se može staviti i slika, naravno manjih dimenzija, tipa neki logo ili slično. Primjerice:



*Slika 6. Primjer brand slike*

Kod:

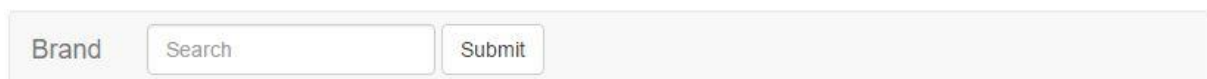
```

<nav class="navbar navbar-default">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">
        
      </a>
    </div>
  </div>
</nav>

```

### 3.1.2. Obrazac za pretraživanje u navigaciji

Često vidimo na web stranicama obrazac za pretraživanje koji se nalazi u navigaciji, pa će ovdje biti objašnjeno kako to napraviti u Bootstrapu.



*Slika 7. Primjer obrasca za pretraživanje u navigaciji*

HTML kod obrasca je u konačnici:

```

<form class="navbar-form navbar-left" role="search"> <-- Elementom navbar-
form pozivamo predefinirani Bootstrap obrazac, a navbar-left govori da će se
obrazac nalaziti na lijevoj strani navigacije -->

```

```

<div class="form-group">
  <input type="text" class="form-control" placeholder="Search">
</div>
<button type="submit" class="btn btn-default">Submit</button>
<-- .btn-default je predefiniran element za defaultni Bootstrap gumb. -->
</form>

```

### 3.1.3. Pozicioniranje navigacije

Često vidimo na modernijim web stranicama kada scrollamo po njima, da nas navigacija cijelo vrijeme prati ili čak da je navigacija cijelo vrijeme fiksirana u podnožje stranice, što isto kod nekih web stranica izgleda dobro i efektivno. Ovdje će biti objašnjeno kako to napraviti pomoću Bootstrapa. U Bootstrapu postoje tri moguće solucije pozicioniranja navigacije, to su FIXED TO TOP, navigacija koja je na vrhu web stranice i cijelo vrijeme nas prati, FIXED TO BOTTOM, također nas cijelo vrijeme prati, ali je na dnu web stranice, te STATIC TOP navigacija koja je klasična i prilikom scrollanja pomiče se zajedno sa ostalim sadržajem.

Kod za fixed to top:

```

<nav class="navbar navbar-default navbar-fixed-top">
  <div class="container">
    ...
  </div>
</nav>

```

Promotrimo prvu liniju koda, koja je ujedno i najbitnija. Dakle pozivamo *navbar* i *navbar-default* da uopće dobijemo navigaciju i onda samo dodajemo klasu **.navbar-fixed-top** koja će se pobrinuti da navigacija ostane fiksirana na vrhu stranice.

Kod za fixed to bottom:

```

<nav class="navbar navbar-default navbar-fixed-bottom">
  <div class="container">
    ...
  </div>
</nav>

```

Kod je potpuno isti kao i u prethodnom primjeru samo ovdje mijenjamo ključnu riječ, u ovom slučaju **.navbar-fixed-bottom**.

Kod za static top:

```

<nav class="navbar navbar-default navbar-static-top">
  <div class="container">
    ...
  </div>
</nav>

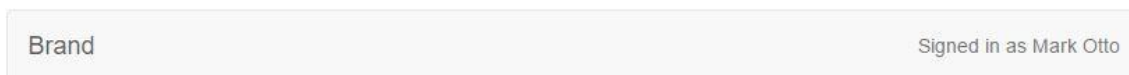
```

Primjer standardne navigacije, jedina stvar koja se mijenja jest vrijednost **.navbar-static-top**.

### 3.1.4. Pozicioniranje sadržaja u navigaciji

Sav sadržaj u našoj navigaciji, kao što su linkovi, input forme, gumbi, tekst, možemo pomicati i pozicionirati kako želimo. Za to su zadužene klase **.navbar-left** i **.navbar-right**. One dodaju elementu *float* na stranu koja se odredi.

Primjer pozicioniranja teksta u navigaciji:



Slika 8. Pozicioniranje teksta u navigaciji

Kod:

```
<p class="navbar-text navbar-right">Signed in as Marko Otto</p>
```

Dakle, kao što je vidljivo iz slike, „Signed in as Marko Otto“ pozicionirano je na desnoj strani navigacije zbog klase **.navbar-right**.

Nakon što smo objasnili navigaciju i što sve možemo s njom, drugi najučestaliji elementi na web stranicama su gumbi, svaka web stranica ima barem dva ili više gumba i oni također spadaju pod osnove.

## 3.2. Gumbi

Bilo je već spomenuto u poglavlju iznad koja klasa u Bootstrapu služi za kreiranje gumba. To je klasa **.btn** i ukoliko želimo defaultni, običan sivi gumb toj klasi pridodajemo klasu **.btn-default**. Bootstrap sadrži šest predefiniраниh klasa za izgled gumba.



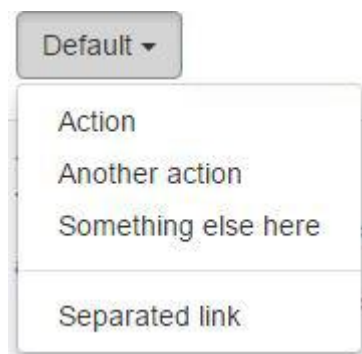
Slika 9. Gumbi u Bootstrapu

Kodovi za svaki od navedenog gumba (redosljedom):

```
<button type="button" class="btn btn-default">Default</button>  
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-success">Success</button>  
<button type="button" class="btn btn-info">Info</button>  
<button type="button" class="btn btn-warning">Warning</button>  
<button type="button" class="btn btn-danger">Danger</button>
```

### 3.2.1. Padajući izbornici

Također, moguće je kreirati i gumb koji ima padajući izbornik, tzv. Dropdown.



Slika 10. Gumb sa padajućim izbornikom

Želimo li kreirati gumb kao što je na slici koristimo kod:

```
<div class="btn-group">
  <button type="button" class="btn btn-default dropdown-toggle" data-
toggle="dropdown" aria-haspopup="true" aria-expanded="false">
<-- klasa aria-haspopup="true" govori da će se otvoriti popup na gumbu kada
kliknemo -->
  Action <span class="caret"></span> <-- Strelica dolje -->
</button>
<ul class="dropdown-menu">
  <li><a href="#">Action</a></li>
  <li><a href="#">Another action</a></li>
  <li><a href="#">Something else here</a></li>
  <li role="separator" class="divider"></li> <-- Separator -->
  <li><a href="#">Separated link</a></li>
</ul>
</div>
```

### 3.2.2. Definiranje veličina

Svaki gumb ima predefinirane veličine koje lako pozivamo ukoliko želimo da naš gumb bude manji ili veći. Postoje četiri veličine gumba, to su: veliki, normalni, mali i extra mali.



Slika 11. Veličine gumba

Da bi koristili te gume koristimo klase **.btn-lg**, **.btn-sm** i **.btn-xs**, za normalnu veličinu gumba ne postoji zasebna klasa jer se ta veličina podrazumijeva kao defaultna, tj. zadana.

Kako to izgleda u kodu:

```
<button type="button" class="btn btn-primary btn-lg">Large button</button>
<button type="button" class="btn btn-primary">Default button</button>
<button type="button" class="btn btn-primary btn-sm">Small button</button>
<button type="button" class="btn btn-primary btn-xs">Extra small
button</button>
```

Nakon proučavanja gumba, na red dolaze input forme tj. obrasci, neizostavni dio svake web stranice.

### 3.3. Obrasci / Input forme

Sve forme već automatski u sebi sadržavaju neka svojstva. Sve tekstualne forme `<input>` i `<textarea>` sa klasom **.form-control** imaju širinu od 100%, po Bootstrapovim postavkama. Za optimalno pozicioniranje (razmake) poželjno je koristiti klasu u posebnim divovima `.form-group`.



Slika 12. Izgled klasične Bootstrap input forme

Kod:

```
<form>
  <div class="form-group">
    <label for="E-Mail forma">Email address</label>
    <input type="email" class="form-control" id="E-Mail forma"
placeholder="Email">
  </div>
  <div class="form-group">
    <label for="passforma">Password</label>
    <input type="password" class="form-control" id="passform"
placeholder="Password">
  </div>
```

Sve attribute iz ovoga koda smo objasnili iznad, osim *placeholdera*. Placeholder="Email" je sadržaj koji piše unutar forme sivim slovima, prije nego što počnemo upisivati naš tekst. Iako to spada u osnove HTML-a, a ne Bootstrapa.

### 3.3.1. Ubačene forme

Kao što sama riječ govori, inline forme su forme koje se nalaze u istom redu. Za razliku od klasičnih formi koje su jedna ispod druge, ove forme po defaultu imaju vrijednost širine zadane na auto, umjesto 100%, što je i logično jer nebi imalo smisla da je više formi u jednom redu sa vrijednošću width: 100%.

Koristimo ih na isti način kao i klasične forme iz prethodnog primjera, samo u početku koristimo klasu **.form-inline**.

Kod:

```
<form class="form-inline">
  <div class="form-group">
    <label for="exampleInputName2">Name</label>
    <input type="text" class="form-control" id="exampleInputName2"
placeholder="Jane Doe">
  </div>
  <div class="form-group">
    <label for="exampleInputEmail2">Email</label>
    <input type="email" class="form-control" id="exampleInputEmail2"
placeholder="jane.doe@example.com">
  </div>
```

U konačnici, ovakva forma izgleda ovako:



The image shows a horizontal form with two input fields. The first field is labeled "Name" and contains the text "Jane Doe". The second field is labeled "Email" and contains the text "jane.doe@example.com". Both fields have a light gray border and a subtle shadow.

Slika 13. Izgled inline forme

Valja napomenuti da Bootstrap podražava sve standardne HTML forme kao što su textarea, radio buttoni, check buttoni, selektori i drugo. U sljedećem poglavlju objasnit ćemo što su tabovi i kakvi sve mogu biti.

### 3.4. Tabovi

Neke web stranice imaju posebne navigacije koje se drže uz sadržaj stranice ili je navigacija napravljena na način da jedan element pada ispod drugog. To zapravo nije navigacija nego neka vrsta menija, odnosno tabovi.



Slika 14. Primjer taba

Želimo li izraditi ovakav tab, potrebno je definirati unordered listu sa klasama **.nav** i **.nav-tabs**.

To će u kodu izgledati ovako:

```
<ul class="nav nav-tabs">
  <li role="presentation" class="active"><a href="#">Home</a></li>
  <li role="presentation"><a href="#">Profile</a></li>
  <li role="presentation"><a href="#">Messages</a></li>
</ul>
```

Atribut `<li>` znači list item unordered liste koju smo definirali.

### 3.4.1. Padajuće kartice

Moguće je koristiti i padajuće tabove, tačnije da jedan element pada ispod drugoga kao na slici:



Slika 15. Padajući tabovi

Koristimo kod:

```
<ul class="nav nav-pills nav-stacked">
  ...
</ul>
```

**.nav-pills** klasa znači da su tabovi odvojeni, svaki za sebe, dok **.nav-stacked** govori pregledniku da posloži jedan tab ispod drugoga.

### 3.4.2. Poravnavanje kartica

Tabovi mogu biti i centrirani, tačnije tekst u njima. Za to nam služi klasa **.nav-justified** i onda ti tabovi izgledaju ovako:



Slika 16. Primjer justified tabova

```
<ul class="nav nav-tabs nav-justified">
  ...
</ul>
<ul class="nav nav-pills nav-justified">
  ...
</ul>
```

Na slici su dva primjera justified tabova, to su defaultni i tzv. Pills kod kojega je svaki tab prikazan zasebno.

### 3.5. Straničenje

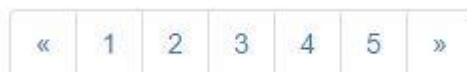
Stranice sa gomilom teksta, npr. online knjige ili forumi, koje treba dugo scrollat moraju imati paginacije. Paginacije web stranice su ništa drugo nego stilizirani linkovi u obliku brojeva koji služe da se pomičemo lakše i brže po stranici. U ovome primjeru izrade portfolia to nećemo koristiti jer nema potrebe.

Za korištenje paginacija treba nam klasa **.pagination** u unoredered listi sa item-ima koji sadrže brojeve.

Kod za straničenje je:

```
<nav>
  <ul class="pagination">
    <li>
      <a href="#" aria-label="Previous">
        <span aria-hidden="true">&laquo;</span>
      </a>
    </li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li>
      <a href="#" aria-label="Next">
        <span aria-hidden="true">&raquo;</span>
      </a>
    </li>
  </ul>
</nav>
```

Rezultat ovoga koda je:



Slika 17. Izgled straničenja



## 3.6. Jumbotron

Prije nekoliko godina, ogromna slika na početnoj stranici koja pokriva cijelu širinu ekrana i gotovo pola visine, bila je nezamisliva. Danas je to jako u trendu pa će iz toga razloga biti prikazano kako to napraviti u Bootstrapu. U Bootstrapu takva velika slika naziva se *Jumbotron*, kao i njezina klasa **.jumbotron**.



*Slika 18. Izgled Jumbotrona*

Kod:

```
<div class="jumbotron">
  <h1>Hello, world!</h1>
  <p>...</p>
  <p><a class="btn btn-primary btn-lg" href="#" role="button">Learn
more</a></p>
</div>
```

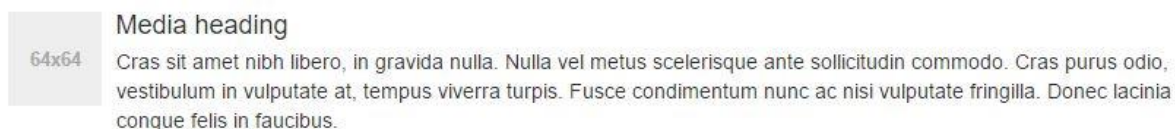
Želimo li iz ovoga jumbotrona napraviti veliku pozadinsku sliku potrebno je uz klasu `jumbotron` dodati još jednu klasu primjerice `.jumb-slika` te u CSS-u pomoću atributa `.background-image` dodati sliku.

Valja napomenuti da bi ta slika trebala biti bez margina, širine 100% i bez zaobljenih rubova (na slici 18. su zaobljeni) potrebno je staviti klasu **.jumbotron** iznad klase **.container**.

```
<div class="jumbotron jumb-slika">
  <div class="container">
    ...
  </div>
</div>
```

### 3.7. Medijski objekti

*Medijski objekti* mogu nam biti korisni kada imamo sliku i pokraj nje neki tekst. Napravljeni su iz razloga da nam olakšaju posao i automatski pozicioniraju sliku kraj teksta, precizno i spremno za responzivnost.



*Slika 19. Izgled media objekta (lijevo) s tekстом*

Kako to funkcionira u praksi možete vidjeti u ovome kodu:

```
<div class="media">
  <div class="media-left">
    <a href="#">
      
    </a>
  </div>
  <div class="media-body">
    <h4 class="media-heading">Media heading</h4>
    ...
  </div>
</div>
```

Media objekte koristimo pomoću klase **.media** u div elementu. Za pozicioniranje slike koristimo **.media-left** (ili **.media-right** za desno), za samo postavljanje slike koristi se **.media-object**, te za tekst **.media-heading** (naslov) i **.media-body** za ostali tekst.

### 3.8. Slike u Bootstrapu

Slike u Bootstrapu 3 mogu biti responzivne pomoću klase **.img-responsive**. Ta klasa aplicira `max-width: 100%; height: auto; display: block;` atribut na sliku, tako da se ona prilagođava objektu pokraj kojeg se nalazi.

Da bi centirali slike koje koriste klasu **.img-responsive**, potrebno je koristiti klasu **.center-block** umjesto `.text-center`.

Kod:

```

```

Slike je također moguće i stilizirati pomoću već gotovih klasa **.img-rounded**, **.img-circle**, **.img-thumbnail**.



*Slika 20. Mogući stilovi slike*

Kod za stilove (redosljedom):

```
  
  

```

U kodu iznad prikazana su sva tri primjera stilova slika. To su zaobljeni, kružni i kvadratni (thumbnail).

### 3.9. Responzivni umetnuti kod

Zahvaljujući Bootstrapu, bez ikakvih problema možemo na našu stranicu ubaciti youtube video koji je responzivan. To znači da se prozor, odnosno sam video smanjuje ili povećava koliko god se smanjuje ili povećava širina stranice.

U nastavku se nalaze kodovi koji nam omogućuju stavljanje responzivnog youtube videa na stranicu.

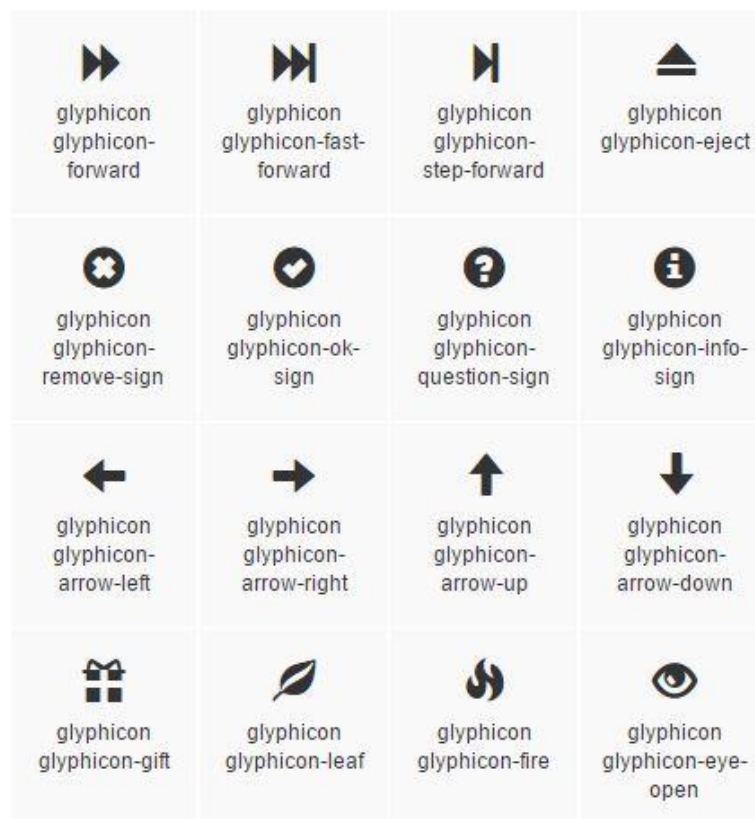
```
<!-- 16:9 aspect ratio -->  
<div class="embed-responsive embed-responsive-16by9">  
  <iframe class="embed-responsive-item" src="link video zapisa"></iframe>  
</div>  
  
<!-- 4:3 aspect ratio -->  
<div class="embed-responsive embed-responsive-4by3">  
  <iframe class="embed-responsive-item" src="link video zapisa"></iframe>  
</div>
```

Kodovi iznad nam omogućuju da YouTube video na našoj web stranici bude responzivan, potrebno je samo u src tag staviti link videozapisa.

Za zadnje odjeljak u ovoj cjelini preostalo je objasniti kako na stranicu dodati ikone.

### 3.10. Ikone (Glyphicons)

Zadnja izdvojena Bootstrap komponenta, ali ne i manje bitna od ostalih su gotove ikone u vektorskom obliku. Bootstrap Glyphicons je set ikona koje koristimo pomoću HTML koda i one se na web stranici ponašaju kao font, što je odlično. Možemo manipulirati s njima kao i sa tekstom bez ikakvih gubitaka na kvaliteti ili problema sa pozicioniranjem i slično. Svaka ikona ima zasebni ID. Te ikone su već ugrađene u instalaciju Bootstrapa i ne treba ih dodatno skidati i povezivati sa vanjskom datotekom. Ikone su potpuno besplatne i ima ih više od 250.



Slika 21. Primjer nekoliko ikona

Ikone koristimo pomoću koda:

```
<span class="glyphicon glyphicon-search" aria-hidden="true"></span>
```

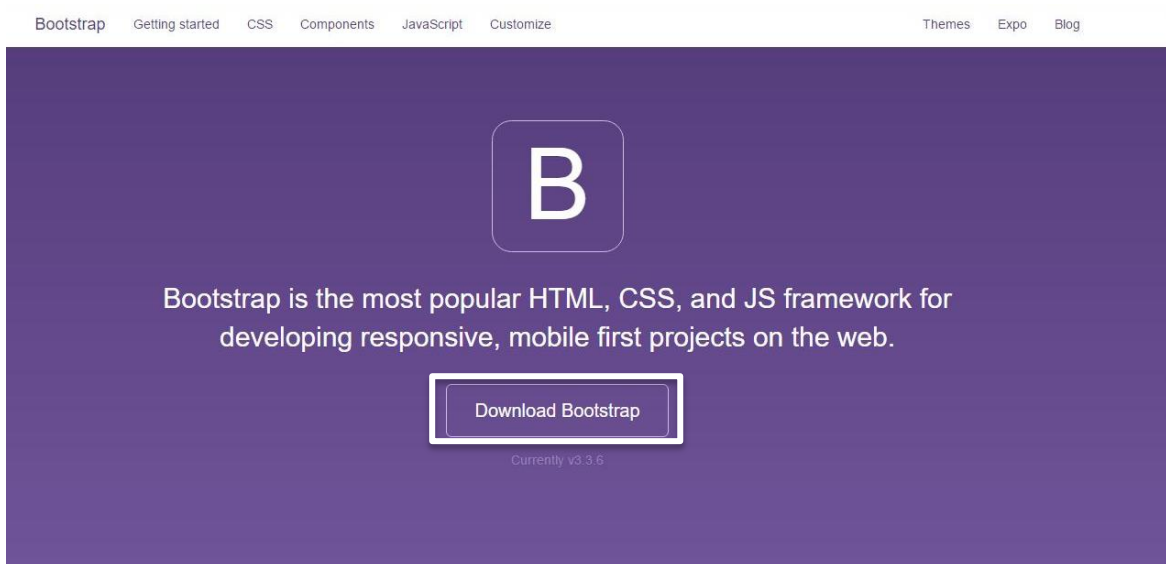
Kao što je bilo rečeno, svaka ikona ima svoje ime, pa je u kodu iznad ikona search ta koju želimo prikazati.



## 4. Konfiguracija Bootstrapa

Postoji više načina za instalaciju tj. konfiguraciju Bootstrapa, ali ovdje će biti prikazan najlakši i najosnovniji.

Prvo je potrebno otići na službenu Bootstrap stranicu <http://getbootstrap.com/> i kliknuti na sredini zaslona veliki gumb **Download Bootstrap**.



Slika 22. Prvi korak konfiguracije Bootstrapa

Nakon toga otvorit će se nova stranica gdje također treba na lijevoj strani kliknuti na gumb **Download Bootstrap**.

### Download

Bootstrap (currently v3.3.6) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

#### Bootstrap

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

Download Bootstrap

#### Source code

Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and some setup.**

Download source

#### Sass

Bootstrap ported from Less to Sass for easy inclusion in Rails, Compass, or Sass-only projects.

Download Sass

Slika 23. Drugi korak konfiguracije Bootstrapa

Započet će preuzimanje bootstrap-3.3.7-dist.rar dokumenta koji je potrebno pomoću nekoga programa za arhiviranje ili kompresiju datoteka raspakirati na neko željeno mjesto na računalu.

Name	Date modified	Type	Size
css	24.11.2015. 19:34	File folder	
fonts	24.11.2015. 19:34	File folder	
js	24.11.2015. 19:34	File folder	

Slika 24. Bootstrap folder prilikom raspakiranja

Ovo je tzv. main folder Bootstrapa i u njemu je potrebno napraviti dokument prve web stranice, **index.html**.

Za izradu osnovne stranice u tekst editor zalijepimo ovaj kod:

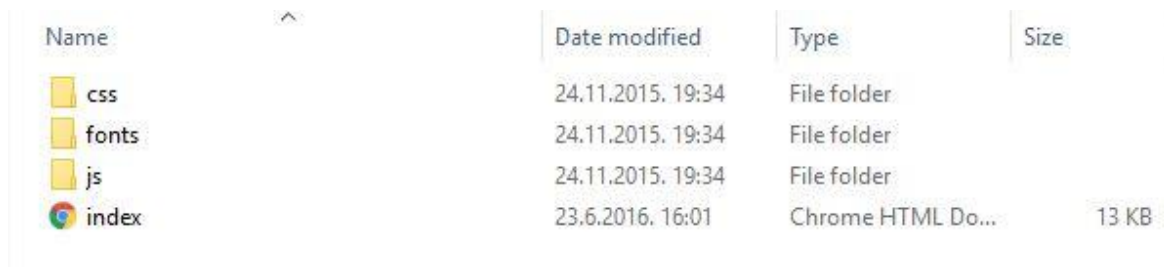
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other head
content must come *after* these tags -->
    <title>Goran Bukovac | Portfolio</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and
media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -
->
    <!--[if lt IE 9]>
      <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></sc
ript>
    <!-- Include all compiled plugins (below), or include individual files
as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Ovo je osnovni početni Bootstrap kod koji nam olakšava posao u smislu da ne trebamo sami započinjat pisanje, a dodatan plus je da su povezane već sve vanjske potrebne datoteke (CSS i JavaScript). Kada taj kod zalijepimo u Notepad++ treba ga spremiti pod nazivom index.html i main Bootstrap folder, pa na u folderu to treba izgledati ovako:



Name	Date modified	Type	Size
css	24.11.2015. 19:34	File folder	
fonts	24.11.2015. 19:34	File folder	
js	24.11.2015. 19:34	File folder	
index	23.6.2016. 16:01	Chrome HTML Do...	13 KB

*Slika 25. Izgled konfiguriranog Bootstrapa sa index.html datotekom*

Kada otvorite index.html, ukoliko se pojavi tekst **Hello, world!** znate da ste sve dobro napravili.

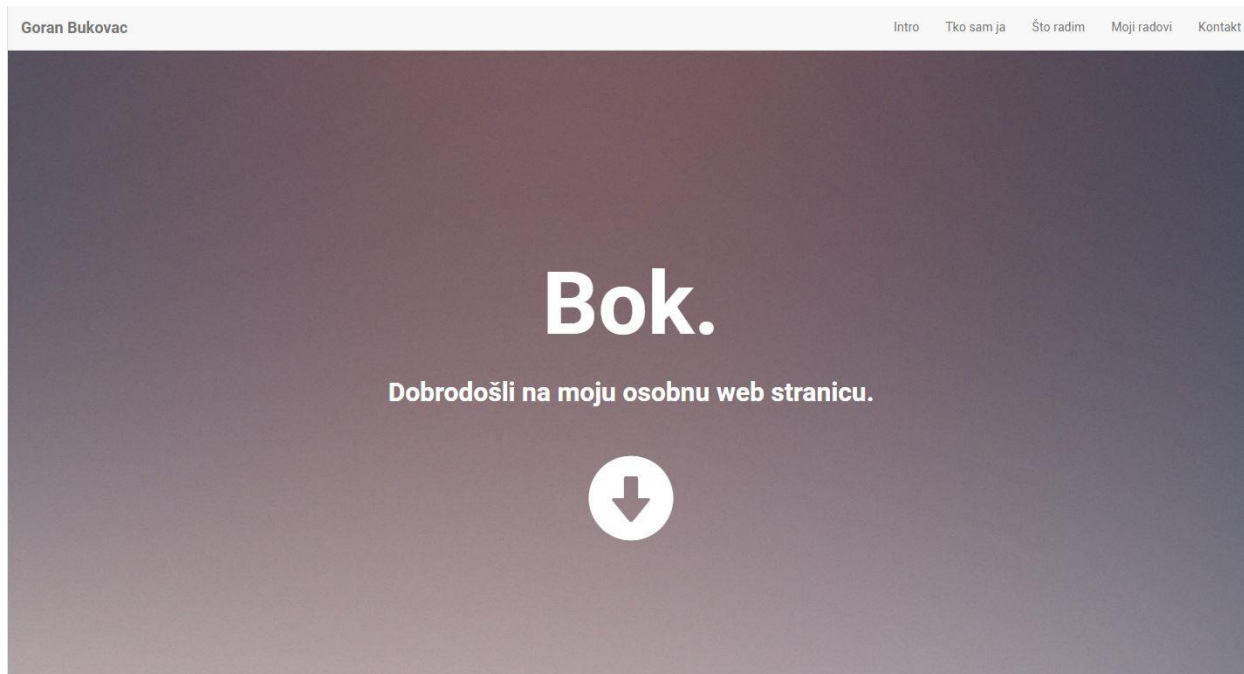
To je to što se tiče skidanja i konfiguriranja Bootstrapa i alata potrebnih za izradu web stranice, u slijedećim poglavljima biti će objašnjen korak po korak proces nastanka web stranice pomoću svih komponenti i alata koje smo do sada spomenuli.

## 5. Izrada osobnog portfolia

Osobni portfolio koji sam radio zamišljen je kao „One Page“ stranica. To znači da će se stranica uz CSS, slike, JavaScript i fontove sastojati samo od jednog jedinog \*.html dokumenta, a to je index.html koji smo već napravili u prethodnom poglavlju. Na njemu će se nalaziti sav sadržaj koji će međusobno biti odvojen slikama koje će prekrivati cijelu širinu i visinu zaslona. Korisnici stranice će *scrollati* i služiti će se strelicama preko kojih će saznati sve više i više informacija o meni, redoslijedom. Također na vrhu stranice bit će fiksirana navigacija, pa se preko nje instantno može skočiti na željenu sekciju stranice. Web stranica će biti sastavljena uglavnom od slika i teksta, na dnu će biti kontakt obrazac i trebat će nam dva JavaScript koda. Jedan kod će biti za zaglađeno scrollanje, a drugi za galeriju, točnije popup slika iz galerije mojih radova. Stranica će se sastojati od pet sekcija, to su: Intro/Početna, Tko sam ja, Što radim, Moji radovi i Kontakt.

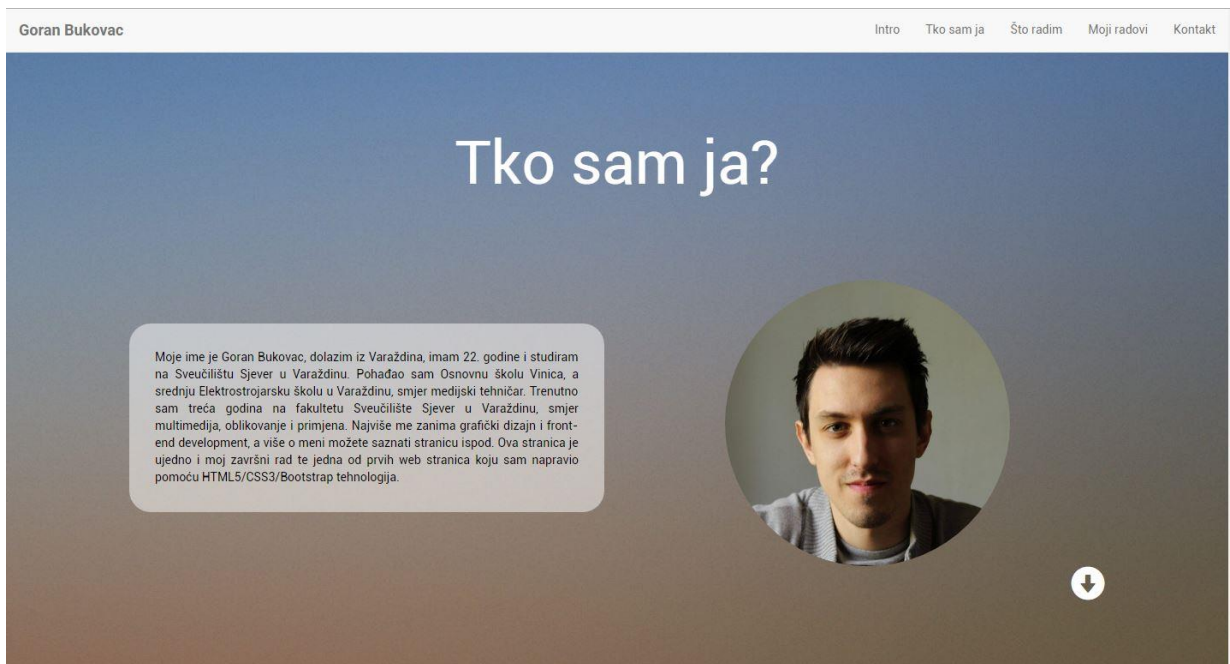
Napomena: Sve pozadinske slike i materijali, zajedno sa cijelim kodom ukoliko bude potrebno, nalaze se na [linku](#).

Ciljani izgled stranice:

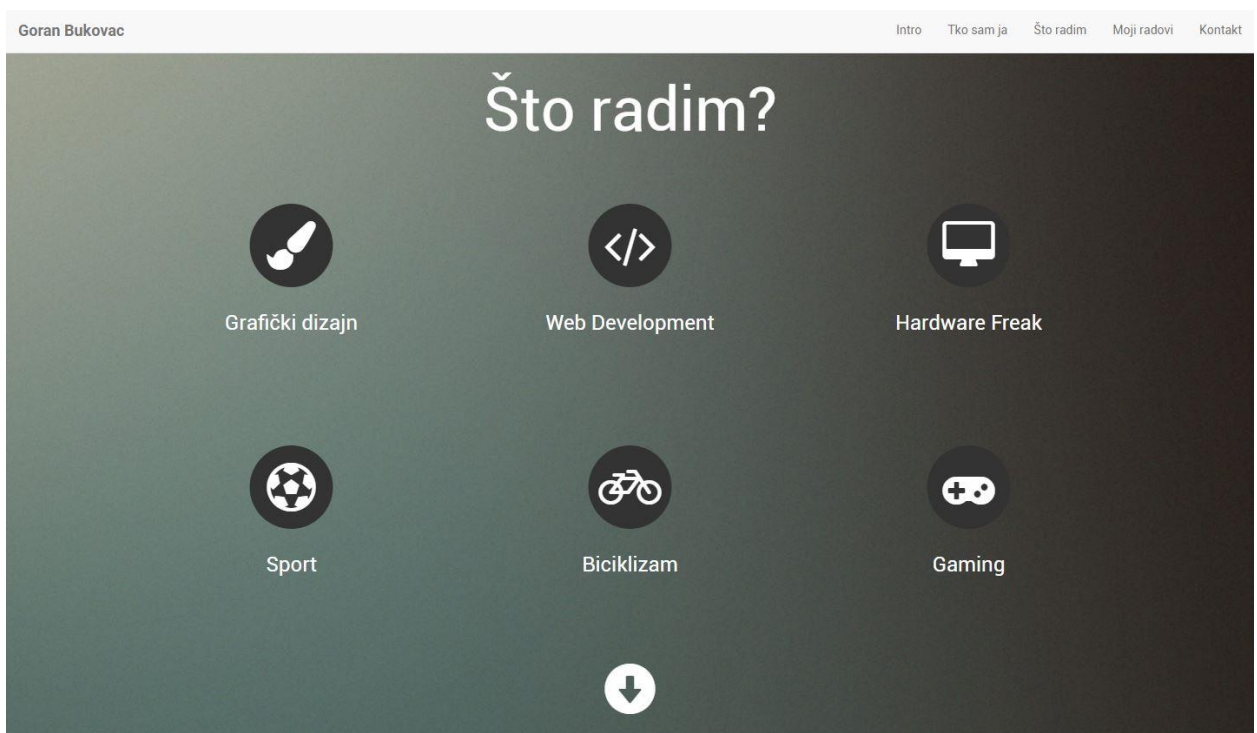


Slika 26. Izgled početne intro stranice



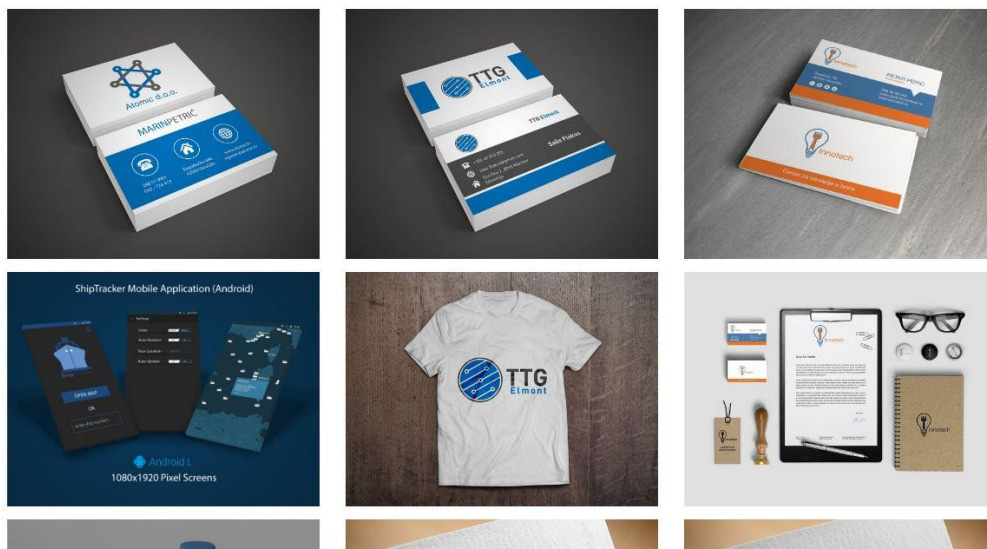


Slika 27. Izgled stranice Tko sam ja



Slika 28. Izgled stranice Što radim

## Moji radovi



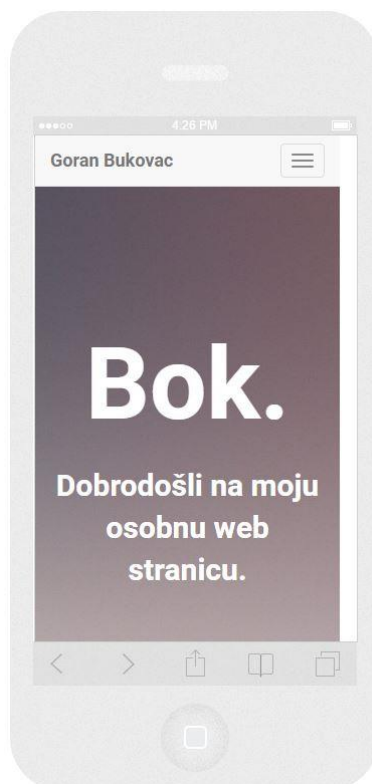
Slika 29. Izgled galerije



## Kontakt

Slika 30. Izgled obrasca za kontakt i footera

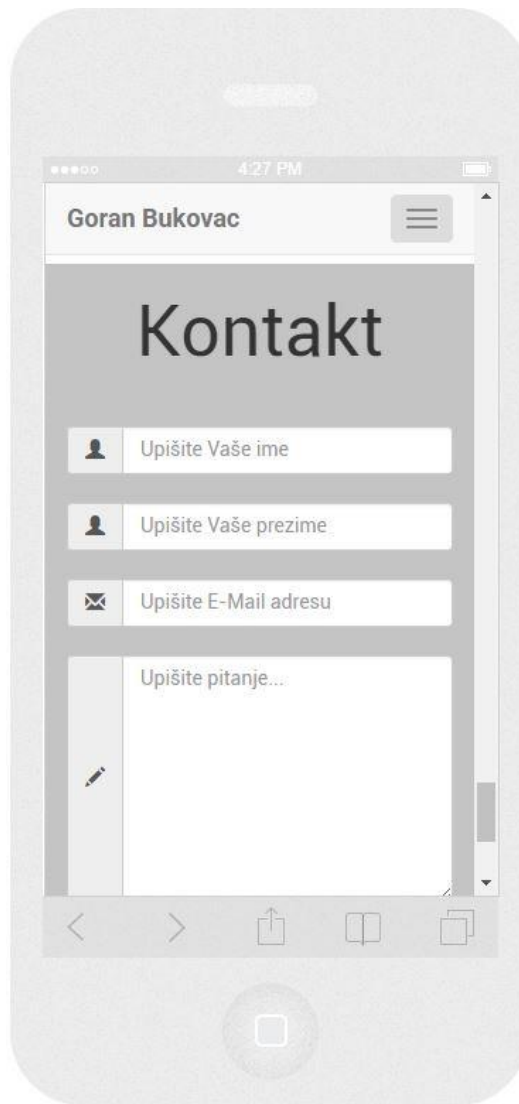
Mobilni izgled:



Slika 31. Mobilni izgled (1)



Slika 32. Mobilni izgled (2)



Slika 33. Mobilni izgled (3)

Kreiranje stranice biti će objašnjeno redosljedom, od vrha prema dnu, pa možemo krenuti sa poglavljem 7.1. gdje će biti objašnjeno kako napraviti navigaciju.

## 5.1. Izrada navigacije

Kao što je vidljivo na slici 26. na samome vrhu nalazi se navigacija. Radi se o Bootstrap *Fixed to Top* navigaciji koju smo spominjali u poglavlju 5.1.

Pošto smo već objasnili sva svojstva navigacije, ta **ista** svojstva ćemo primijeniti ovdje, tako da bi sve trebalo biti jasno i nećemo ponavljati više svaku liniju koda zasebno.

Kod za navigaciju:

```
<nav class="navbar navbar-default navbar-fixed-top">
  <div class="container-fluid">
    <!-- Brand and toggle get grouped for better mobile display -->
    <div class="navbar-header">
```

```

    <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
    <span class="sr-only">Toggle navigation</span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
    <span class="icon-bar"></span>
</button>
<a class="navbar-brand ime-prezime" href="#">Goran Bukovac</a>
</div>

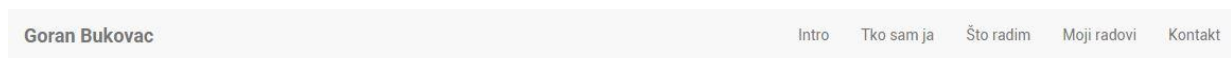
<!-- Collect the nav links, forms, and other content for toggling -->
<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

    <ul class="nav navbar-nav navbar-right">
    <li><a href="#page1">Intro</a></li>
    <li><a href="#page2">Tko sam ja</a></li>
    <li><a href="#page3">Što radim</a></li>
    <li><a href="#page4">Moji radovi</a></li>
    <li><a href="#page5">Kontakt</a></li>

    </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

```

Pomoću ovoga koda dobili smo ovu navigaciju:



*Slika 34. Izgled navigacije*

## 5.2. Izrada prve (Intro) stranice

Prvo i osnovno je potrebno promotriti na koji način su sekcija stranice spojene i kako funkcioniraju, kako radi scrolanje i navigiranje po stranici. Stvar je riješena pomoću CSS **#id**-eva. Za početak treba napraviti novi dokument u Notepad++ i spremiti ga u main/css folder u Bootstrap folderu. Nazovemo ga stil.css. U ovome dokumentu će biti spremljena sva naša CSS obilježja. Dakle prije nego nastavimo, budimo sigurni da imamo **index.html** i **stil.css** dokumente. Obavezno u index.html dokumentu u <headu> povezati naš stil.css.

```
<link href="css/stil.css" rel="stylesheet">
```

Prva stranica sastoji se od izuzetno malo HTML koda jer nema puno sadržaja, možemo vidjeti dvije linije teksta te strelicu koja je preuzeta sa Bootstrap Glyphicons biblioteke i u vektorskom font formatu je.

Dakle, prvo je bitno zadati div element koji će pokriti 100% širine i visine naše web stranice. Nakon toga napraviti kontejner i u njega dodati red i stupac sa tekстом, te ispod njih dodati Glyphicons strelicu koja će imati poveznicu na stranicu ispod.

Kod:

```
<div id="page1">
<div class="container">
<div class="row">
<div class="col-lg-12">
  <p class="title">Bok.</p>
  <p class="title-ispod">Dobrodošli na moju osobnu web stranicu.</p>
  <a href="#page2"><i class="glyphicon glyphicon-circle-arrow-down
strelica fade hidden-xs aria-hidden="true"></i></a>
</div>
</div>
</div>
</div>
```

U kodu se nalaze četiri klase koje je bilo potrebno napraviti da bi elementi poprimili izgled kakav je zamišljen. To su klase **.title**, **.title-ispod**, **.strelica**, i **.fade**.

Ovako ove klase izgledaju u CSS-u:

```
.title {
padding-top: 35vh;
color: white;
text-align: center;
font-size: 100px;
font-weight: bold;
}

.title-ispod {
color: white;
text-align: center;
font-size: 30px;
font-weight: bold;
}

.strelica{
color: white;
text-align: center;
width: 100%;
font-size: 100px;
margin-top: 40px;
}

.fade {
opacity: 1;
transition: opacity .25s ease-in-out;
-moz-transition: opacity .25s ease-in-out;
-webkit-transition: opacity .25s ease-in-out;
}

.fade:hover {
opacity: 0.5;
}
```

Može se primijetiti da sve te klase imaju manje više slične selektore i attribute, osim klase **.fade** koja je priča za sebe. Napravljena je pomoću intervala u stotinkama i prijelaza stanja iz solid oblika u opacity, tako da kad mišem prijedemo na strelicu vidimo laganu fade out animaciju.

Ima klasa koja je ostavljena za kraj i koju sa razlogom nisam naveo. To je klasa **div id** elementa **#page1**. Bitno je izostaviti tu klasu jer je ona najbitnija klasa cijela naše web stranica i na njoj su bazirane ostale sekcije. Dakle, id **#page1** zaslužen je za scrollanje, punu veličinu pozadinske slike i linkove.

Iz koda navigacije možemo izdvojiti detalj koda izbornika:

```
<ul class="nav navbar-nav navbar-right">
  <li><a href="#page1">Intro</a></li>
  <li><a href="#page2">Tko sam ja</a></li>
  <li><a href="#page3">Što radim</a></li>
  <li><a href="#page4">Moji radovi</a></li>
  <li><a href="#page5">Kontakt</a></li>
</ul>
```

Taj kod je linkan kao što vidite, **#page1**, **#page2**, **#page3**, **#page4** i **#page5** stranicama, upravo to su stranice preko kojih se navigiramo o „skačemo“ kada pretražujemo portfolio. Namjerno sam spomenuo skačemo jer tako dugo dok ne implementiramo potreban JavaScript kod, prilikom navigiranja doslovce ćemo skočiti na stranicu koju smo kliknuli.

**#page1** definiran je ovim CSS stilovima:

```
#page1 {
  min-height: 100vh;
  background-image: url("../img/intro.jpg");
  background-size: cover;
}
```

Daleko najvažniji selektor i atribut ovdje je **min-height: 100vh;**

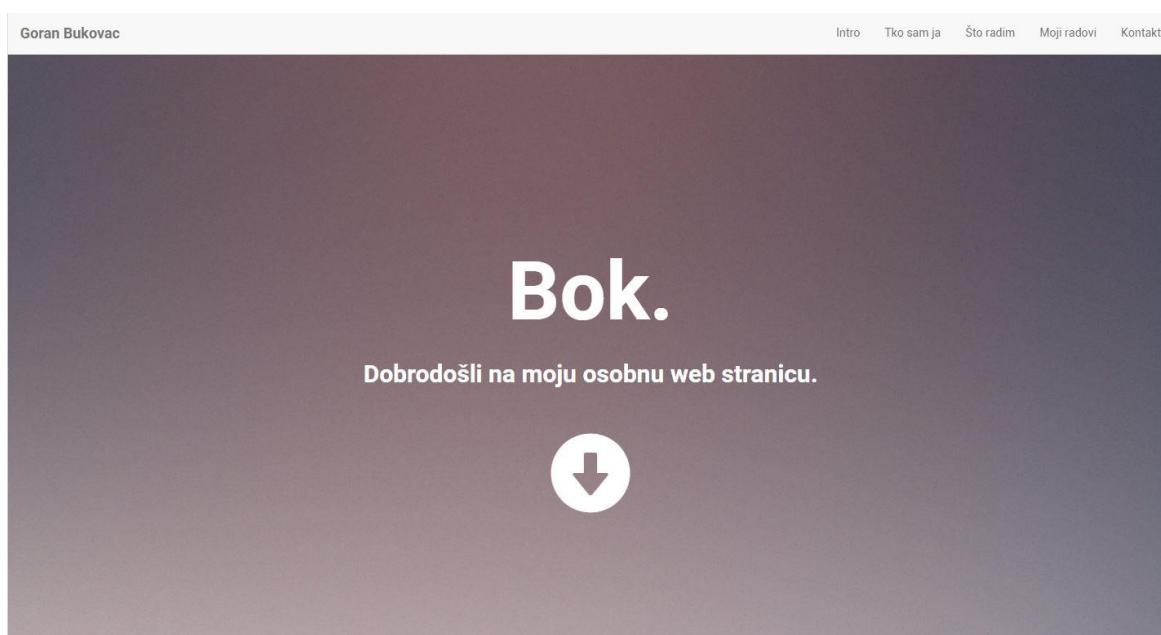
Prilikom izrade te stranice, najviše sam problema imao sa upravo ovim pozadinskim slikama. Dakle prvo sam definirao `background-image: url("../img/intro.jpg");` i to je prošlo dobro, također potrebno je staviti i `background-size: cover;` što nije bilo dovoljno jer slika jednostavno nije toliko velika u svojem originalnom obliku da pokrije cijeli zaslon. Širina slika je u redu i slika se raširi preko 100% zaslona zahvaljujući atributu `background-size: cover;` ali problem je visina. Prva stvar koju sam pokušao bila je staviti `height: 100%;` međutim to nije radilo. Nakon toga `height: 1000px;` što je bilo dobro i moglo je funkcionirati da stranica nije responzivna, međutim

kada bi se stranica smanjila ili nešto slično, slika bi ostala na fiksnih 1000 piksela što dovodi do problema.

Zahvaljujući CSS3 postoje nove jedinice a to su `vh` i `vw`. Točnije `viewport height` i `viewport width`. Tako je u ovome primjeru stavljen `height: 100vh` i to govori našem pregledniku da slika uvijek pokriva 100% visine zaslona, kako god proširimo ili smanjimo stranicu, ona će uvijek pokrivati 100% visine zaslona.

Zašto je nije samo `height: 100vh`; nego `min-height: 100vh`; biti će objašnjeno u poglavlju 7.5. jer je tamo konkretan primjer za bolje razumijevanje.

Stranica u konačnici izgleda ovako:



Slika 35. Izgled stranice prve stranice

Kada je izrađena prva stranica, vrijeme je za drugu stranicu „Tko sam ja“.

### 5.3. Izrada stranice „Tko sam ja“

Na drugoj stranici „Tko sam ja“ imamo tekst i sliku koji se nalaze u jednom redu i dva stupca širine šest spojenih stupaca. Sa desne strane avatara je strelica koja nas vodi na treću stranicu. Ima sva ista svojstva kao gornja strelica samo je manja i pozicionirana je desno. Tekst je stavljen u paragraf sa lijeve strane, a slike s desne strane definirana stilom *img-responsive* i *img-circle* kao što je bilo spomenuto u poglavlju 3.8. Slike u Bootstrapu.

Ovako izgleda HTML kod:

```
<div id="page2">
  <p class="pozicija-1"> Tko sam ja? </p>
  <div class="container tko-sam-ja">
```



```

<div class="row">
  <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12">
    <p class="opis-tekst"> Moje ime je Goran Bukovac, dolazim iz Varaždina,
    imam 22. godine i studiram na Sveučilištu Sjever u Varaždinu. Pohađao sam
    Osnovnu školu Vinica, a srednju Elektrostrojarsku školu u Varaždinu, smjer
    medijski tehničar. Trenutno sam treća godina na fakultetu Sveučilište Sjever
    u Varaždinu, smjer multimedija, oblikovanje i primjena. Najviše me zanima
    grafički dizajn i front-end development, a više o meni možete saznati
    stranicu ispod. Ova stranica je ujedno i moj završni rad te jedna od prvih
    web stranica koju sam napravio pomoću HTML5/CSS3/Bootstrap tehnologija.</p>
  </div>
  <div class="col-lg-6 col-md-6 col-sm-6">
    
  </div>

</div>
<a href="#page3"><i class="glyphicon glyphicon-circle-arrow-down strelica-
druga pull-right fade hidden-xs aria-hidden="true"></i></a>
</div>
</div>

```

Ništa komplicirano i neobjašnjeno u prethodnim poglavljima. Najveći dio koda zapravo uzima tekst o meni.

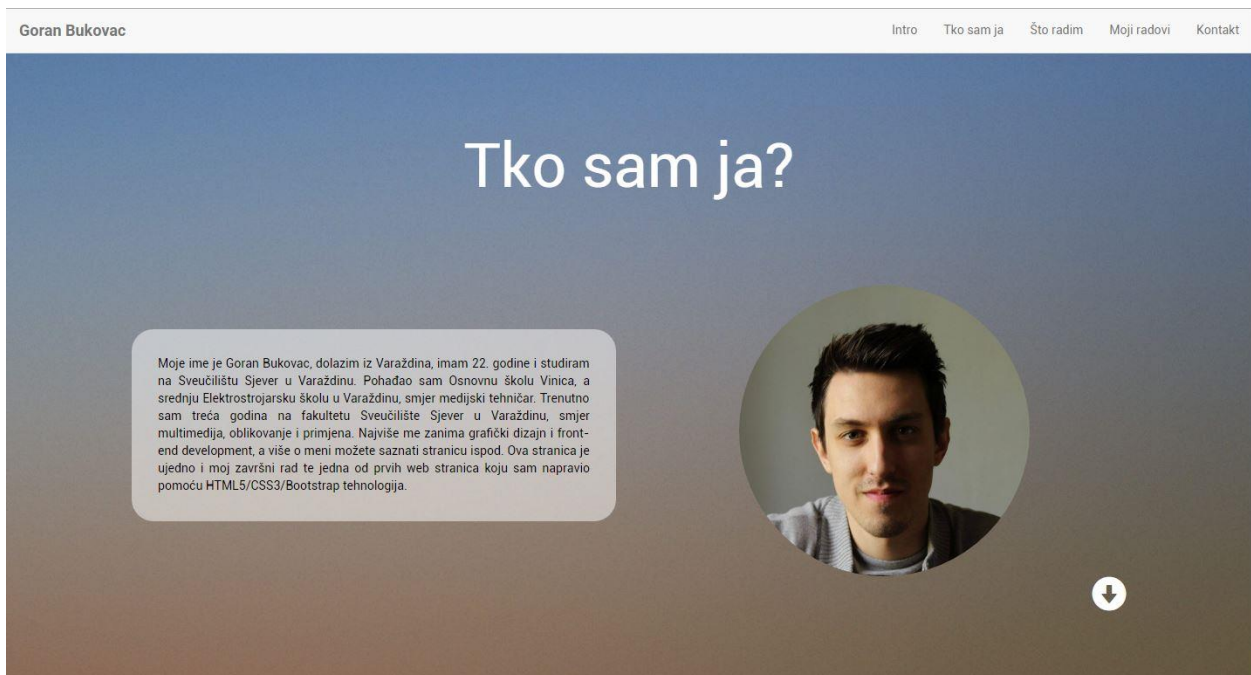
U kodu ovom djelu koda imamo također četiri svoje klase, to su: **.pozicija-1 (u media upitu)**, **.tko-sam-ja**, **.opis-tekst**, **.strelica-druga** i **.velicina-avatara**. Naravno ovdje je i klasa **.fade** za strelicu koja je već objašnjena.

CSS:

```

#page2 {
  min-height: 100vh;
  background-image: url("../img/prva.jpg");
  background-size: cover;
}
.strelica-druga{
  color: white;
  font-size: 100px;
}
.opis-tekst{
  color: white;
  text-align: justify;
}
.tko-sam-ja{
  padding-top: 10vh;
}
.velicina-avatara{
  width: 50%;
  height: auto;
  margin-left: 8vh;
}

```



Slika 36. Izgled stranice Tko sam ja

U nastavku sljedi objašnjenje izrade treće stranice „Što radim“.

## 5.4. Font Awesome

Prije nego što nastavimo sa objašnjenjem izrade treće stranice, potrebno je objasniti što je Font Awesome. Najjednostavnije rečeno, to je ista stvar kao što su Bootstrapove ugrađene ikone (Glyphicons), samo ima puno više načina za modifikaciju, možemo mijenjati oblik, pozadinu ikone, veličinu, praktički sve što zamislimo.

Font Awesome ikone dostupne su na stranici <http://fontawesome.io/>

Kada ih skinemo raspakiramo sav sadržaj foldera u main Bootstrap mapu.

Na kraju, potrebno ih je spojiti sa našim projektom tako da kod ispod kopiramo u <head> sekciju našeg HTML dokumenta odnosno index.html.

```
<link rel="stylesheet" href="css/font-awesome.min.css">
```

Na stranici <http://fontawesome.io/icons/> možete pronaći sve Font Awesome ikone, a na stranici <http://fontawesome.io/examples/> upute kako ih stilizirati i koristiti.

Primjer koda, kako kreirati ikonu iz Font Awesome biblioteke:

```
<i class="fa fa-camera-retro fa-lg"></i> fa-lg
```

time se u prikazu dobije sličica:



Ukoliko želimo veću ikonu upisujemo umjesto fa-lg, fa-2x ili fa-3x i tako do 5.

Primjerice:

```
<i class="fa fa-camera-retro fa-5x"></i>
```

## 5.5. Izrada stranice „Što radim“

Na ovoj stranici imamo odličan primjer u praksi kako funkcionira Bootstrap grid, redovi i stupci i kako ubacivat željeni sadržaj. Proučite sliku 37. Iz nje zaključujemo da imamo četiri redaka s time da su drugi i treći red podijeljeni na tri stupaca. Podsjetimo se da Bootstrap grid ima 12 stupaca i da bi sadržaj podijelili na 3 stupaca trebamo spojiti 4 njih međusobno da pokrijemo cijeli prostor.  $3 \cdot 4 = 12$ .

U prvom redu nalazi se naslov bijele boje Što radim i u sljedeća dva redaka ispod njega šest Font Awesome bijelih ikona stiliziranih sa crnim pozadinskim okruglim oblikom. U zadnjem redu nalazi se već dva puta spomenuta strelica koja nas u ovom slučaju vodi na četvrtu stranicu „Moji radovi“.

HTML kod:

```
<div id="page3">
  <p class="pozicija2 hidden-xs"> Što radim? </p>
  <div class="container ic-kontener">
    <div class="row">
      <div class="col-lg-4 col-md-4 col-sm-4 ikone-sredina">
        <span class="fa-stack fa-4x">
          <i class="fa fa-circle fa-stack-2x"></i>
          <i class="fa fa-paint-brush fa-stack-1x fa-inverse"></i>
        </span>
        <span class="tekst-ikonica"><h3>Grafički dizajn</h3></span>
      </div>
      <div class="col-lg-4 col-md-4 col-sm-4 ikone-sredina">
        <span class="fa-stack fa-4x">
          <i class="fa fa-circle fa-stack-2x"></i>
          <i class="fa fa-code fa-stack-1x fa-inverse"></i>
        </span>
        <span class="tekst-ikonica"><h3>Web Development</h3></span>
      </div>
      <div class="col-lg-4 col-md-4 col-sm-4 ikone-sredina">
        <span class="fa-stack fa-4x">
          <i class="fa fa-circle fa-stack-2x"></i>
          <i class="fa fa-desktop fa-stack-1x fa-inverse"></i>
        </span>
        <span class="tekst-ikonica"><h3>Hardware Freak</h3></span>
      </div>
    </div>
    <div class="row red-dva-rad">
      <div class="col-lg-4 col-md-4 col-sm-4">
        <span class="fa-stack fa-4x">
          <i class="fa fa-circle fa-stack-2x"></i>
          <i class="fa fa-futbol-o fa-stack-1x fa-inverse"></i>
        </span>
      </div>
    </div>
  </div>
```

```

</span>
<span class="tekst-ikonica"><h3>Sport<h3></span>
  </div>
  <div class="col-lg-4 col-md-4 col-sm-4">
    <span class="fa-stack fa-4x">
      <i class="fa fa-circle fa-stack-2x"></i>
      <i class="fa fa-bicycle fa-stack-1x fa-inverse"></i>
    </span>
  <span class="tekst-ikonica"><h3>Biciklizam<h3></span>
    </div>
    <div class="col-lg-4 col-md-4 col-sm-4">
      <span class="fa-stack fa-4x">
        <i class="fa fa-circle fa-stack-2x"></i>
        <i class="fa fa-gamepad fa-stack-1x fa-inverse"></i>
      </span>
    <span class="tekst-ikonica"><h3>Gaming<h3></span>
      </div>
    </div>
    <div class="container-fluid">
      <div class="row">
        <div class="col-lg-12">
          <a href="#page4"><i class="glyphicon glyphicon-circle-arrow-down strelica-
          treca fade hidden-xs aria-hidden="true"></i></a>
        </div>
      </div>
    </div>
  </div>
</div>

```

Neka Vas ovaj dugi kod ne preplaši jer je to kod u kojem se zapravo cijelo vrijeme ponavljaju iste stvari.

Izdvojit ću samo dio koda na kojem ću objasniti svaku liniju tako da se kasnije lakše razumije i ostatak.

```

  <div class="row">
    <div class="col-lg-4 col-md-4 col-sm-4 ikone-sredina">
      <span class="fa-stack fa-4x">
        <i class="fa fa-circle fa-stack-2x"></i>
        <i class="fa fa-paint-brush fa-stack-1x fa-inverse"></i>
      </span>
    <span class="tekst-ikonica"><h3>Grafički dizajn<h3></span>
      </div>
    <div class="col-lg-4 col-md-4 col-sm-4 ikone-sredina">
      <span class="fa-stack fa-4x">
        <i class="fa fa-circle fa-stack-2x"></i>
        <i class="fa fa-code fa-stack-1x fa-inverse"></i>
      </span>
    <span class="tekst-ikonica"><h3>Web Development<h3></span>
      </div>
    <div class="col-lg-4 col-md-4 col-sm-4 ikone-sredina">
      <span class="fa-stack fa-4x">
        <i class="fa fa-circle fa-stack-2x"></i>
        <i class="fa fa-desktop fa-stack-1x fa-inverse"></i>
      </span>
    <span class="tekst-ikonica"><h3>Hardware Freak<h3></span>
      </div>
    </div>

```

Dakle, u prvom redu pozivamo red i koji stavljamo stupac koji zauzima četiri stupca u gridu (od dvanaest njih). U tome stupcu nalazi se font awesome stack (crna okrugla pozadina) i na njoj ikona kista. Nakon toga imamo još dva stupca od kojih svaki zauzima također četiri stupca u gridu sa Font Awesome ikonama i istim svojstvima, samo naravno drugoj sličici. Nakon toga ide drugi rad i sav isti kod samo sa drugim ikonama i tekstom i na kraju strelica koja nas vodi na četvrtu stranicu.

CSS klase na ovoj stranici su **.pozicija2**, **.ic-kontener**, **.ikone-sredina**, **.fade**, **.strelica-treca** i **.tekst-ikonica**.

CSS kod:

```
#page3 {
    min-height: 100vh;
    background-image: url("../img/druga.jpg");
    background-size: cover;

    .ikone-kontener{
        text-align: center;
    }

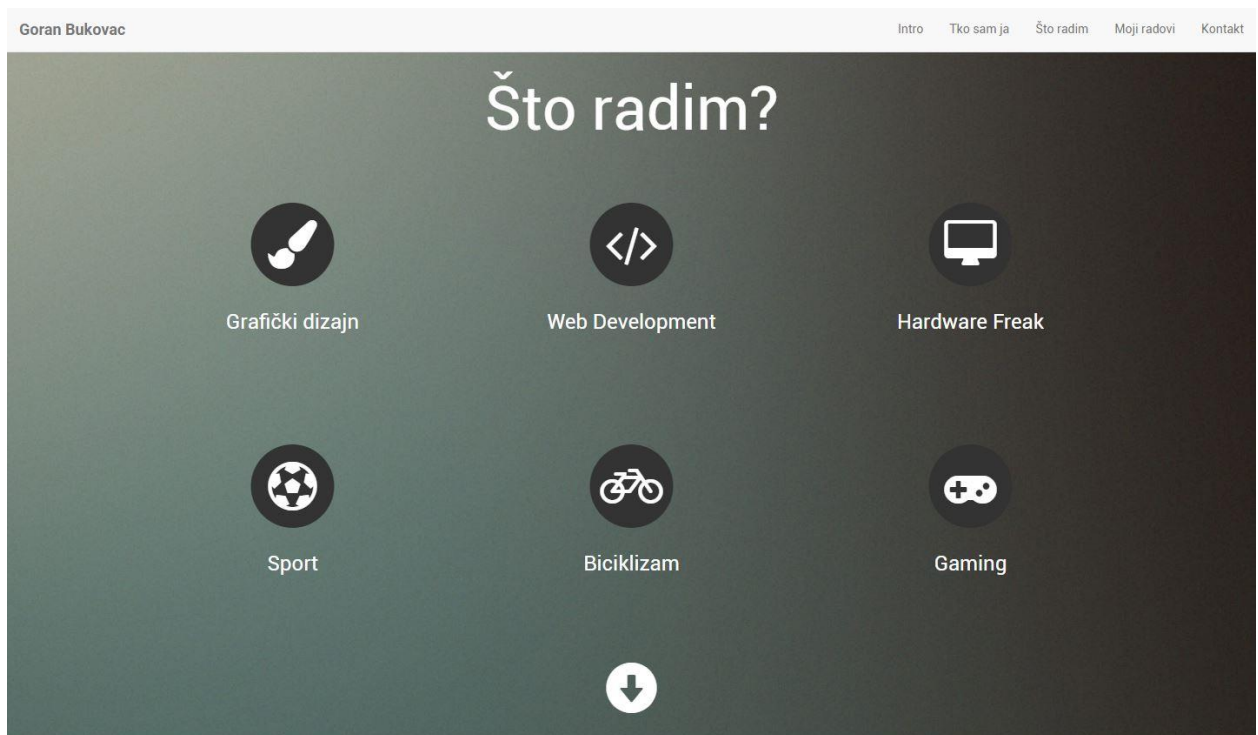
    .tekst-ikonica{
        color: white;
    }

    .ic-kontener{
        text-align: center;
    }

    .strelica-treca{
        font-size: 60px;
        color: white;
    }

    .pozicija2{
        text-align: center;
        color: white;
        padding-top:50px;
        font-size: 70px;
        margin-bottom: -160px;
    }
}
```

Kada smo to sve ukomponirali i posložili Font Awesome ikone, dobili smo ovo:



Slika 37. Izgled stranice „Što radim“

Sada je na redu objašnjenje četvrte stranice „Moji radovi“ koja je možda i najviše kompleksna iz razloga jer nam treba i malo JavaScripta.

## 5.6. Izrada stranice „Moji radovi“

Odmah na početku treba reći da je stranica koncipirana na vrlo sličan način kao prethodna samo su redovi podijeljeni na četiri puta po tri stupca. Sam HTML kod je puno lakši od prethodnog jer ubacujemo slike u grid pomoću `<img>` taga koji nije kompleksan kao prijašnje Font Awesome ikone.

Nakon naslova moji radovi napravljena je galerija, ali bez JavaScripta, napravljena je u sklopu Bootstrapovih mogućnosti i njegovog grida i namjerno je napravljena tako čisto da se prikažu moći i mogućnosti Bootstrapa. JavaScript kod nam služi samo za otvaranje, tj. popup slika i on će biti dostupan na dnu ovoga poglavlja.

HTML kod:

```
<div id="page4">
  <div class="container radovi">
    <div class="row">
      <div class="col-lg-12">
        <h1 class="pozicija-radovi"> Moji radovi </h1>
      </div>
    </div>
  </div>
  <div class="container galerija">
    <div class="row razmak-galerije">
```

```

<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/atomic_vizitke.jpg">
</a>
</div>
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/TTG_Vizitke.jpg">
</a>
</div>
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/Innotech_vizitka.jpg">
</a>
</div>
</div>
<div class="row razmak-galerije">
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/ShipTracker_app.jpg">
</a>
</div>
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/TTG_majca.jpg">
</a>
</div>
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/Innotech_mock.jpg">
</a>
</div>
</div>
<div class="row razmak-galerije">
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/redizajn_meggle.jpg">
</a>
</div>
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/Innotech_logo.jpg">
</a>
</div>
<div class="col-lg-4 portfolio razmak-slika">
<a class="image-link" href="img/gal/TTG_logo.jpg">
</a>
</div>
</div>
</div>
</div>
</div>

```

Sve ovo je već jasno iz prethodnih primjera, možemo samo izdvojiti CSS kod naših klasa koje su o ovom slučaju **.razmak-slika** (u media upitu) **.razmak-galerije**, **.gal-img**, **.radovi** i **.pozicija-radovi**.

CSS:

```

#page4 {
  min-height: 100vh;
  background-color: #fff;
  background-size: cover;
}

```

U poglavlju 5.2. napomenuo sam da ću o ovom poglavlju objasniti zašto koristimo `min-height` selektor umjesto `height`. Prvotno sam koristio `height` za sve, ali bolje je koristiti `min-height` i to sam uvidio tek kada sam radio galeriju. Galerija je bila jednostavno prevelika i zauzimala je više od 100vh ekrana, a mi smo definirali da točno mora biti 100vh što je dovelo do miksanja stranica Kontakt i Galerija, jednostavno stranica kontakt je pregazila galeriju jer je sadržaj od galerije prošao 100vh i nastao je kaos. Zbog toga razloga sam svugdje stavio `min-height` od 100vh i rekao pregledniku da minimalni `height` neka bude 100vh, a ako treba više on će si automatski sam uzeti više i produžiti. Upravo to je slučaj sa galerijom, trebala je više mjesta i automatski se produžila na više od 100vh.

Nastavak CSS koda:

```
.gal-img{max-width: 100%;}
.gal-img:hover{opacity:0.8}
.pozicija-radovi {
    text-align: center;
    padding-top: 40px;
    margin-bottom: 60px;
    font-size: 50px;
}
```

Klasa **.image-link** ja iz JavaScript koda i služi za otvaranje slika prilikom klika. Objašnjenje u slijedećem poglavlju.

### 5.6.1. Magnific Popup - JavaScript

U ovome pod poglavlju se malo odmičemo od Bootstrapa i objasniti ćemo kako iz slika naše galerije napraviti popup sa efektom pomoću JavaScripta. Izabrao sam Magnific Popup jer je poprilično jednostavan i što je najbitnije u našem radu, responzivan je.

Da bi skinuli i konfigurirali sve potrebno za Magnific Popup potrebno je posjetiti web stranicu:

<http://dimsemenov.com/plugins/magnific-popup/>

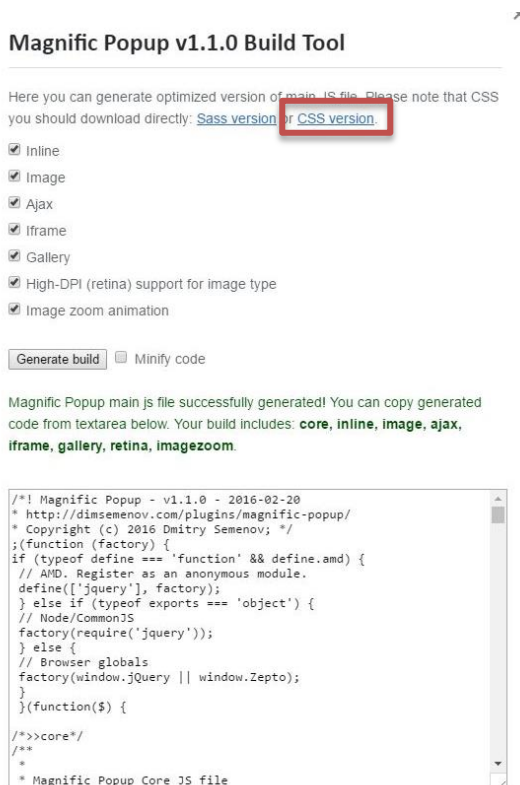
Sada je potrebno pratiti nekoliko koraka da bi generirali kod za naše potrebe. Kada dođete na gore spomenutu web stranicu treba ići pod Build Tool





Slika 38. Magnific Popup stranica

Nakon toga će se otvoriti novi prozor kod kojega treba uraditi sve kao što je na slici 39. i pritisnuti gumb **Generate build**.



Slika 39. Magnific Popup Build Tool

Dobili smo iz generiran kod koji treba cijeli kopirati i zalijepiti u novi Notepad++ dokument i nazvati ga **jquery.magnific-popup.js**. Još jednom napominjem, bitno je da ima \*.js ekstenziju. U Bootstrap/projekt main folderu napravimo novu mapu pod nazivom „js“ i u nju spremiti dotičnu datoteku.

Nakon toga vraćamo se u Build tool i gore kliknemo na link CSS version (vidi sliku 39. označeno crvenim).

Pojavit će se novi CSS kod kojega cijeloga kopiramo i zalijepimo u novi Notepad++ dokument. Dokument spremamo u mapu „css“ pod nazivom **magnific-popup.css**. Prve i osnovne pripreme za implementaciju našega JavaScript Plugina su napravljene. Sada ga je potrebno samo ispravno povezati sa našim projektom.

Potrebno je dodati ovaj kod na kraj našeg HTML dokumenta, ispred </body> taga.

```
<script src="js/jquery.magnific-popup.js"></script>
<script type="text/javascript">
// Initialize popup as usual
$('.image-link').magnificPopup({
type: 'image',
mainClass: 'mfp-with-zoom',

zoom: {
enabled: true,
duration: 300,
easing: 'ease-in-out',
opener: function(openerElement) {
return openerElement.is('img') ? openerElement : openerElement.find('img');
}}});
</script>
```

Ovim kodom smo povezali datoteku koju smo prethodno kreirali i stavili u „js“ folder, jquery.magnific-popup.js.

Sada je potrebno još spojiti CSS datoteku koju smo također kreirali, a to radimo tako da ovaj kod zalijepimo u <head> našega HTML dokumenta.

```
<link rel="stylesheet" href="css/magnific-popup.css">
```

To je to, naš Plugin bi trebao raditi i slike bi prilikom klika trebale zaglađeno iskakat u novom popup prozoru.



Slika 40. Izgled Popup plugina

## 5.7. Izrada kontakt forme

Kontakt forme bile su objašnjene u poglavlju 5.3. pa ćemo samo primijeniti znanja od prije za izradu naše kontakt forme. Sastoji se od tri input formi te jedna textarea od osam redova. Na kraju je gumb za slanje.

HTML kod za obrazac je dugačak jer su u njemu i Glyphicone, ali su i kodovi za izradu obrazaca sami po sebi dugački.

```
<div id="page5">
<div class="container kontakt-sve">
<div class="row">
<div class="col-lg-12">
<h1 class="kontakt"> Kontakt </h1> <br>
</div>
</div>
<div class="row">
<div class="col-lg-6 col-lg-offset-3 kontakt">
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-
user"></i></span>
<input name="ime" placeholder="Upišite Vaše ime" class="form-control"
type="text">
</div>
</div>
</div>
<div class="row">
<div class="col-lg-6 col-lg-offset-3 kontakt">
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-
user"></i></span>
<input name="prezime" placeholder="Upišite Vaše prezime" class="form-
control" type="text">
</div>
</div>
</div>
<div class="row">
<div class="col-lg-6 col-lg-offset-3 kontakt">
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-
envelope"></i></span>
<input name="email" placeholder="Upišite E-Mail adresu" class="form-control"
type="text">
</div>
</div>
</div>
<div class="row">
<div class="col-lg-6 col-lg-offset-3 kontakt">
<div class="input-group">
<span class="input-group-addon"><i class="glyphicon glyphicon-
pencil"></i></span>
<textarea class="form-control" rows="8" name="comment" placeholder="Upišite
pitanje..."></textarea>
</div>
</div>
</div>
</div>
<div class="row">
```

```

<div class="col-lg-12">
<button type="submit" class="btn btn-primary" >Pošalji <span
class="glyphicon glyphicon-send"></span></button>
</div>
</div>
</div>

```

U ovome kodu trebalo bi biti sve jasno i poznato od prije osim offseta. Klasu offset koristimo kada želimo pomaknuti element prema desno. Offset klasa dodaje elementu marginu s lijeve strane ovisno o broju koji zadamo.

Primjerice `.col-md-offset-4` pomiče stupac `.col-md-4` za četiri stupaca desno.



*Slika 41. Primjer offseta*

### CSS stilovi:

```

.kontakt{
margin-bottom: 20px;
text-align: center;
font-size: 50px;
}

```

```

.kontakt-sve{
text-align: center;
}

```

Dobili smo ovo:

*Slika 42. Izgled kontakt forme*

Za kraj je ostala izrada podnožja, odnosno footera.

## 5.8. Izrada footera

Podnožje stranice koncipirano je na način da u jednom redu postoje dva ravnomjerno podijeljena stupca, dakle dva stupca koja zauzimaju po 6 mjesta u gridu. Sa lijeve strane je običan paragraf, a desne su Font Awesome ikone koje su već više puta bile spomenute.

HTML kod podnožja:

```
<div class="container podnozje">
<div class="row">
<div class="col-lg-6">

<p class="text-pad">Designed and developed by Goran Bukovac</p>
</div>

<div class="col-lg-6">

<a href="https://www.facebook.com/bukovac.goran" target="_blank"<i class="fa
fa-facebook-square fa-3x ikona-margina fade" aria-hidden="true"></i></a>

<a href="mailto:gbukovac94@gmail.com" target="_top"<i class="fa fa-envelope
fa-3x ikona-margina fade" aria-hidden="true"></i></a>

<a href="#"<i class="fa fa-linkedin-square fa-3x ikona-margina fade" aria-
hidden="true"></i></a>

<a href="#"<i class="fa fa-phone-square fa-3x fade" aria-
hidden="true"></i></a>

</div>
</div>
</div>
```

CSS:

```
.podnozje{
    width: 100%;
    padding-top: 25px;
    padding-bottom: 25px;
    margin-top: 50px;
    background-color: #8C8C8C;
    text-align: center;
    color: white;
}

.ikona-margina{
    margin-right: 10px;
}

.text-pad{
    padding-top: 10px;
}
```

Footer izgleda ovako:

Slika 43. Izgled footera

Završili smo sa svim stranicama naše web stranice, međutim ostalo je još nekoliko stvari koje treba razjasniti. U poglavlju 4.4. bili su objašnjeni media upiti i kako ih koristiti, ali još ih nismo spomenuli ovdje u praktičnom djelu. To je ostavljeno za kraj, točnije u priložima gdje se nalazi pregled svih potrebnih media upita sa klasama koje je potrebno zalijepiti u stil.css dokument da bi stranica funkcionirala.

## 5.9. Smooth Scroll

Za zadnje poglavlje ostao je JavaScript kod za zaglađeno scrolanje po stranici. To nije ništa nužno i obavezno, ali puno ljepše i ugađenije izgleda. Sve što treba napraviti jest zalijepiti ovaj kod na kraj HTML dokumenta iznad završnog </body> taga.

```
<script>
$(document).ready(function() {

//definiranje funkcije filterPath
function filterPath(string) {
    return string
        .replace(/^\/\//, '')
        .replace(/(index|default)\.[a-zA-Z]{3,4}$/, '')
        .replace(/\/$/, '');
}

//petlja koja prolazi kroz svaki element (a tag koji vodi na link koji
počinje s #)
$('a[href*=#]').each(function() {
//provjera valjanosti
    if ( filterPath(location.pathname) == filterPath(this.pathname)
        && location.hostname == this.hostname
        && this.hash.replace(/#/,'') ) {
        var $targetId = $(this.hash), $targetAnchor = $('[name=' +
this.hash.slice(1) +']');
        var $target = $targetId.length ? $targetId : $targetAnchor.length ?
$targetAnchor : false;
        if ($target) {

//udaljenost od vrha stranice
            var targetOffset = $target.offset().top;

//dogadaj koji se izvršava na klik - animacija do $targetOffset
            $(this).click(function() {
                $('html, body').animate({scrollTop: targetOffset}, 1000);
                return false;
            });
        }
    }
})
}
```

```
});  
});  
</script>
```

Time je osobna stranica portfolia kompletirana. Kako sve ne bi ostalo samo na osnovnom HTML-u i Bootstrap-u stranica je dodatno obogaćena Font Awesome ikonama, Magnific Popupom i Smoothscrollom za bolje korisničko iskustvo.

## 6. Zaključak

Responzivne web stranice su danas jedan od najbrže rastućih online trendova. Same po sebi imaju prednosti prilagodbe za sve zaslone, što je ujedno i veliki plus za pristupačnost korisnika koji ih koriste. Svi posjetitelji takvih stranica biti će zahvalni što je stranica responzivna, a i prednost nad konkurencijom biti će velika. Responzivan dizajn omogućuje i lakše održavanje jer se sve nalazi u jednom kodu, sve je zapravo jedna stranica. Prije se to radilo tako da bi se radila posebna stranica za desktop, a posebna za mobilne uređaje, pa bi to izazivalo komplikacije i teže ažuriranje.

Izrada responzivne web stranice bitno je olakšana poznavanje Bootstrap komponenti koje smo obradili u ovom završnom radu. Korištenje Bootstrapa podrazumijeva poznavanje osnova HTML i CSS jezika. Bootstrap je jednostavan za razumjeti, a opet s druge strane moćan alat za brzu i efikasnu izradu responzivnih web stranica. Pokazalo se da pomoću HTML-a i CSS-a uz Bootstrap možemo napraviti responzivnu web stranicu koja izgleda dobro, a opet nije previše kompleksna za napraviti. Naravno, sve to treba dodatno vizualno srediti, pa su sve ikone koje su na stranici napravljene pomoću Font Awesome biblioteke ikona. JavaScript kodovi za smooth scroll i Maginific Popup (iskakanje slika) umetnuti su radi boljeg korisničkog iskustva.

Na kraju, valja napomenuti da je ovaj praktični rad samo jedan od milijun primjera kako napraviti responzivnu web stranicu pomoću Bootstrapa. Ujedno, Bootstrap ima i puno više komponenata od ovih koje smo obradili u ovoj temi, ali naše obrađene komponente su one osnovne koje će biti dovoljne za izradu velike većine jednostavnijih web stranica. Naprednije i bogatije web stranice automatski zahtijevaju veće znanje HTML, CSS, ali i drugih front-end jezika poput JavaScripta, SASS-a i dr.

Budućnost Bootstrapa je vedra, trenutno je u razvoju verzija 4 koja je još u alpha fazi, ali vjerujem da će službena verzija brzo izaći, donijeti puno noviteta i steći još veću popularnost od Bootstrapa 3.

U Varaždinu, \_\_\_\_\_

\_\_\_\_\_



## 7. Literatura

- [1] HTML & XHTML: The Definitive Guide; Chuck Musciano and Bill Kennedy; O'Reilly; 2009.
- [2] HTML5: Up and Running; Mark Pilgrim; O'Reilly; 2010.
- [3] [http://www.w3schools.com/js/js\\_html5.asp/](http://www.w3schools.com/js/js_html5.asp/) Pristupio: 20.06.2016.
- [4] <http://toni.podmanicki.com/blog/html5/> Pristupio: 20.06.2016.
- [5] <https://www.w3.org/TR/2011/NOTE-css-2010-20110512/> Pristupio: 20.06.2016.
- [6] <https://webdesign.tutsplus.com/articles/css4-is-coming-what-you-need-to-know--webdesign-11521/> Pristupio: 20.06.2016.
- [7] <http://www.w3schools.com/css/> Pristupio: 22.06.2016.
- [8] <https://www.w3.org/TR/css3-values/#calc/> Pristupio: 22.06.2016.
- [9] <http://getbootstrap.com/> Pristupio: 22.06.2016.
- [10] <http://getbootstrap.com/about/> Pristupio: 25.06.2016.
- [11] <http://getbootstrap.com/css/#grid-intro/> Pristupio: 25.06.2016.
- [12] <http://getbootstrap.com/css/#grid-options/> Pristupio: 25.06.2016
- [13] <http://getbootstrap.com/css/#grid-media-queries/> Pristupio: 25.06.2016.
- [14] <http://getbootstrap.com/css/#responsive-utilities/> Pristupio: 25.06.2016.
- [15] <http://getbootstrap.com/components/> Pristupio: 26.06.2016.

## Popis slika

Slika 1. Prikaz Bootstrap mreže .....	7
Slika 2. Prikaz fluidnog i fiksnog kontejnera .....	8
Slika 3. Primjer prikaza redova i stupaca u Bootstrapu .....	9
Slika 4. Prikaz hidden i visible opcija .....	12
Slika 5. Prikaz defaultne Bootstrap navigacije .....	13
Slika 6. Primjer brand slike .....	14
Slika 7. Primjer obrasca za pretraživanje u navigaciji.....	14
Slika 8. Pozicioniranje teksta u navigaciji .....	16
Slika 9. Gumbi u Bootstrapu .....	16
Slika 10. Gumb sa padajućim izbornikom.....	17
Slika 11. Veličine gumba.....	17
Slika 12. Izgled klasične Bootstrap input forme.....	18
Slika 13. Izgled inline forme .....	19
Slika 14. Primjer taba .....	19
Slika 15. Padajući tabovi .....	20
Slika 16. Primjer justified tabova .....	21
Slika 17. Izgled straničenja.....	21
Slika 18. Izgled Jumbotrona .....	22
Slika 19. Izgled media objekta (lijevo) sa tekstom.....	23
Slika 20. Mogući stilovi slike .....	24
Slika 21. Primjer nekoliko ikona .....	25
Slika 22. Prvi korak konfiguracije Bootstrapa.....	26
Slika 23. Drugi korak konfiguracije Bootstrapa .....	26
Slika 24. Bootstrap folder prilikom raspakiravanja.....	27
Slika 25. Izgled konfiguriranog Bootstrapa sa index.html datotekom .....	28
Slika 26. Izgled početne intro stranice.....	29
Slika 27. Izgled stranice Tko sam ja.....	30
Slika 28. Izgled stranice Što radim .....	30
Slika 29. Izgled galerije .....	31
Slika 30. Izgled obrasca za kontakt i footera.....	31
Slika 31. Mobilni izgled (1).....	32
Slika 32. Mobilni izgled (2).....	32
Slika 33. Mobilni izgled (3).....	33

Slika 34. Izgled navigacije.....	34
Slika 35. Izgled stranice prve stranice .....	37
Slika 36. Izgled stranice Tko sam ja .....	39
Slika 37. Izgled stranice „Što radim“ .....	43
Slika 38. Magnific Popup stranica.....	46
Slika 39. Magnific Popup Build Tool.....	46
Slika 40. Izgled Popup plugina .....	47
Slika 41. Primjer offseta .....	49
Slika 42. Izgled kontakt forme .....	49
Slika 43. Izgled footera.....	51

# Prilozi

## Media upiti:

```
@media (max-width: 767px) {
  .velicina-avatara { width: 55%; height: auto; display: block; margin-
left: auto; margin-right: auto; margin-top: 200px; margin-bottom: 30px;
}
  .opis-tekst { margin-top: -40px; margin-bottom: 20px; padding: 30px;
background-color:rgba(255,255,255,0.5); border-radius: 25px; color:
black;}
  .ikone-kontener{padding-top: 40px; font-size: 10px;}
  .ikone-rad{padding-bottom: 35px;}
  .razmak-slika{margin-bottom: 15px;}
  .pozicija-1 {text-align: center; color: white; font-size: 70px;
padding-top: 30px;}
  .ic-kontener{padding-top: 20px;}
}

@media (min-width: 768px) {
  .velicina-avatara { width: 65%; height: auto; display: block; margin-
left: auto; margin-right: auto; }
  .opis-tekst { margin-top: -50px; padding: 30px; background-
color:rgba(255,255,255,0.5); border-radius: 25px; color: black;}
  .pozicija {font-size: 30px; padding-top: 20vh;}
  .strelica-druga{font-size: 40px;}
  .ikone-kontenesr{padding-top: 200px;}
  .red-dva-rad{padding-top: 100px;}
  .strelica-dflt-tri{color: white;text-align: center;width: 100%;font-
size: 100px; margin-top: 80px;}
  .razmak-slika{margin-bottom: 15px;}
  .ikone-sredina{margin-top: 25vh;}
  .pozicija-1 {text-align: center; color: white; font-size: 70px;
padding-top: 30px;}
}

@media (min-width:992px){
  .velicina-avatara { max-width: 60%; height: auto; }
  .opis-tekst { margin-top:0px; padding: 30px; background-
color:rgba(255,255,255,0.5); border-radius: 25px; color: black;}
  .pozicija {font-size: 80px; padding-top: 20vh;}
  .ikone-kontener{vertical-align: middle;}
  .red-dva-rad{padding-top: 100px;}
  .ikone-sredina{margin-top: 25vh;}
  .pozicija-1 {text-align: center; color: white; font-size: 70px;
padding-top: 30px;}
}
```

```
@media (min-width:1200px){
    .velicina-avatara { max-width: 60%; height: auto; }
    .opis-tekst { margin-top: 50px; padding: 30px; background-
color:rgba(255,255,255,0.5); border-radius: 25px; color: black;}
    .ikone-kontener{vertical-align: middle;}
    .red-dva-rad{padding-top: 100px;}
    .ikone-sredina{margin-top: 25vh;}
    .pozicija-1 {text-align: center; color: white; font-size: 70px;
padding-top: 80px;}
    .strelica-treca {padding-top: 80px;}
}
```

IZJAVA O AUTORSTVU  
I  
SUGLASNOST ZA JAVNU OBJAVU

Završni/diplomski rad isključivo je autorsko djelo studenta koji je isti izradio te student odgovara za istinitost, izvornost i ispravnost teksta rada. U radu se ne smiju koristiti dijelovi tuđih radova (knjiga, članaka, doktorskih disertacija, magistarskih radova, izvora s interneta, i drugih izvora) bez navođenja izvora i autora navedenih radova. Svi dijelovi tuđih radova moraju biti pravilno navedeni i citirani. Dijelovi tuđih radova koji nisu pravilno citirani, smatraju se plagijatom, odnosno nezakonitim prisvajanjem tuđeg znanstvenog ili stručnoga rada. Sukladno navedenom studenti su dužni potpisati izjavu o autorstvu rada.

Ja, GORAN BUKOVAC (ime i prezime) pod punom moralnom, materijalnom i kaznenom odgovornošću, izjavljujem da sam isključivi autor/ica završnog/diplomskog (obrisati nepotrebno) rada pod naslovom IZRADA RESPONZIVNOG PORTFOLIA KORISTEJEN WEB TEH (upisati naslov) te da u navedenom radu nisu na nedozvoljeni način (bez pravilnog citiranja) korišteni dijelovi tuđih radova.

Student/ica:  
(upisati ime i prezime)

Goran Bukovac  
(vlastoručni potpis)

Sukladno Zakonu o znanstvenoj djelatnosti i visokom obrazovanju završne/diplomske radove sveučilišta su dužna trajno objaviti na javnoj internetskoj bazi sveučilišne knjižnice u sastavu sveučilišta te kopirati u javnu internetsku bazu završnih/diplomskih radova Nacionalne i sveučilišne knjižnice. Završni radovi istovrsnih umjetničkih studija koji se realiziraju kroz umjetnička ostvarenja objavljuju se na odgovarajući način.

Ja, GORAN BUKOVAC (ime i prezime) neopozivo izjavljujem da sam suglasan/na s javnom objavom završnog/diplomskog (obrisati nepotrebno) rada pod naslovom IZRADA RESPONZIVNOG PORTFOLIA KORISTEJEN WEB TEH (upisati naslov) čiji sam autor/ica.

Student/ica:  
(upisati ime i prezime)

Goran Bukovac  
(vlastoručni potpis)