

Korištenje CSS Grid rasporednog modula u razvoju korisničkih sučelja prilagodljivih različitim širinama zaslona

Milić, Nikola

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Humanities and Social Sciences / Sveučilište Josipa Jurja Strossmayera u Osijeku, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:142:338324>

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

Download date / Datum preuzimanja: **2024-12-28**



FILOZOFSKI FAKULTET
SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

Repository / Repozitorij:

[FFOS-repository - Repository of the Faculty of Humanities and Social Sciences Osijek](#)



Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet

Odsjek za informacijske znanosti

Preddiplomski studij informatologije

Nikola Milić

**Korištenje CSS Grid rasporednog modula u razvoju
korisničkih sučelja prilagodljivih različitim širinama zaslona**

Završni rad

Mentor: izv.prof.dr.sc. Boris Badurina

Komentor: dr.sc. Tomislav Jakopc, poslijedoktorand

Osijek, rujan 2018.

Sveučilište J.J. Strossmayera u Osijeku

Filozofski fakultet

Odsjek za informacijske znanosti

Preddiplomski studij informatologije

Nikola Milić

**Korištenje CSS Grid rasporednog modula u razvoju
korisničkih sučelja prilagodljivih različitim širinama
zaslona**

Završni rad

Društvene znanosti, informacijske i komunikacijske znanosti, informacijski sustavi
i informatologija

Mentor: izv.prof.dr.sc. Boris Badurina

Komentor: dr.sc. Tomislav Jakopec, poslijedoktorand

Osijek, rujan 2018.

Sadržaj

Sažetak	1
Uvod	2
CSS Grid.....	3
Podrška u preglednicima.....	4
CSS Grid svojstva.....	5
Razmak između stupaca i redova	5
Pozicioniranje elemenata unutar <i>grida</i>	6
Definiranje područja unutar grida	10
Razvijanje korisničkog sučelja prilagodljivog različitim širinama zaslona.....	12
Zaključak	17
Literatura	19
Prilozi	20

Sažetak

Tema rada je CSS Grid rasporedni modul čija je namjena organizacija sadržaja mrežnih stranica ili aplikacija. Riječ je o poprilično novom CSS modulu koji trenutno nije podržan u svim preglednicima, no razina podrške poprilično je visoka. CSS Grid modul sadrži veći broj različitih svojstava, a u radu su obrađena ona najosnovnija i najbitnija. Organizacija sadržaja primjenom ovog modula prilično je jednostavna, a od posebne je pomoći kada je u pitanju vertikalno i horizontalno pozicioniranje različitih elemenata stranice, odnosno aplikacije. U radu su obrađena učestala sučelja s kojima se korisnici susreću, riječ je o navigacijskoj traci i galeriji fotografija. Primjeri u radu poprilično su jednostavni kako bi se na što jednostavniji i jasniji način mogla prikazati različita svojstva ovog modula. U trenutku pisanja ovog rada, najveći problem s kojim se programeri susreću u razvoju mrežnih stranica ili aplikacija je činjenica da modul nema potpunu pregledničku podršku, te s toga moraju pribjegavati pisanju takozvanog rezervnog (*fallback*) koda. U slučaju da preglednik ne može pravilno prikazati sadržaje stranice razvijene primjenom CSS Grid modula, učitava se rezervni kod na osnovu kojega se sadržaj prikazuje pravilno. U radu je prikazana jednostavnost kojom programeri mogu organizirati sadržaje prilagodljive različitim širinama zaslona primjenom ovog modula.

Ključne riječi: *CSS Grid, CSS svojstvo, širina zaslona, razvijanje mrežnih stranica, pozicioniranje sadržaja*

Uvod

Mrežne stranice od svojih su se početaka razvile iz relativno jednostavnih tekstualnih dokumenata, najčešće popraćenih slikama ili videozapisima, u složene i interaktivne **web aplikacije** različitih primjena, sastavljene od velikog broja različitih odjeljaka, modula i elemenata. Posao programera mrežnih stranica samim time postao je složeniji, no jednim dijelom i jednostavniji. Pojavili su se brojni alati, tehnologije i CSS moduli koji značajno olakšavaju razvijanje jedne mrežne stranice, programerima su otvorene brojne mogućnosti u razvoju stranica, a samim time i veći stupanj slobode i kreativnosti. Ono što komplicira razvijanje mrežnih stranica prvenstveno su različite širine zaslona i različiti mrežni preglednici, ali i veća očekivanja od strane samih korisnika.

Osim što su moderne mrežne stranice često poprilično složene, one također moraju biti prilagodljive različitim širinama zaslona. Tako jedna mrežna stranica mora biti istovjetno (koliko god je to moguće) prikazana na zaslonu pametnog telefona dimenzija 320 x 568 (piksela) i zaslonu osobnog računala dimenzija 1920 x 1080. Najpopularnija veličina zaslona u trenutku pisanja ovog rada je 360 x 640¹, riječ je dakle o zaslonu pametnog telefona.

Osim što mora biti istovjetno prikazana na različitim širinama zaslona, mrežna stranica mora biti istovjetno prikazana u različitim mrežnim preglednicima koji još uvijek nisu u potpunosti standardizirani i usuglašeni po pitanju različitih mrežnih standarda. Najčešći problem predstavljaju zastarjeli preglednici koji su unatoč lako dostupnim, boljim i suvremenijim preglednicima još uvijek poprilično popularni. Prema podacima W3 Counter, Internet Explorer 11 je 5. najpopularniji preglednik, a koristi ga 2.44% korisnika.² Ovaj naizgled malen postotak čini se puno veći ako se uzme u obzir to da na stolnim (**desktop**) računalima Internet Explorer 11 ima udio od čak 12% (srpanj, 2018.).³ Ne-prilagođavanje mrežne stranice Internet Exploreru 11 ujedno bi značilo zanemariti čak 12% korisnika stolnih računala. Ovisno o ciljanoj publici stranice, to može biti prihvatljivo ili potpuno ne prihvatljivo.

¹ StatCounter GlobalStats. URL: <http://gs.statcounter.com/screen-resolution-stats> (2018-08-29)

² W3Counter. URL: <https://www.w3counter.com/globalstats.php> (2018-08-29)

³ Net MarketShare. URL: <https://netmarketshare.com/browser-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%22Flaptop%22%5D%7D%7D%5D%7D%7D%7D%2C%22dateLabel%22%3A%22Trend%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22browser%22%2C%22sort%22%3A%7B%22share%22%3A%1%7D%2C%22id%22%3A%22browsersDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222017-08%22%2C%22dateEnd%22%3A%222018-07%22%2C%22segments%22%3A%22-1000%22%7D> (2018-08-29)

Rad se bavi tematikom korištenja CSS Grid rasporednog modula, prvenstveno u kreiranju korisničkih sučelja. Korisničko sučelje može se definirati kao vizualni dio aplikacije koji ima ulogu posrednika u komunikaciji između korisnika i računalnog softvera⁴ – u ovom slučaju mrežne stranice. Kada su u pitanju mrežne stranice korisničko sučelje čine jedan ili više manjih modula koji sastavljaju stranicu, a odgovorni su za njenu funkcionalnost. Najčešći je primjer navigacijska traka koju korisnici koriste kako bi se kretali kroz različite dijelove mrežnih stranica, što je naravno osnovan i ključan element korištenja stranice. Shodno tome, kvalitetan dizajn korisničkog sučelja korisniku olakšava korištenje stranice. CSS Grid rasporedni modul znatno olakšava stvaranje stranica, od manjih modula pa do većih dijelova stranice.

Kroz veći dio rada, na CSS Grid skraćeno se referira kao *grid*. To je vrijednost koja se dodaje CSS svojstvu *display* kako bi određeni element bio prikazan kao CSS Grid. Za primjere u radu korišten je Sublime uređivač teksta i Mozilla Firefox preglednik. Firefox je posebno pogodan za rad s *gridom* jer nudi alate namijenjene baš za tu ulogu. Podrazumijeva se i osnovna razina poznavanja CSS-a za uspješno razumijevanje sadržaja rada. Svi primjeri u radu dostupni su na Githubu (vidi Prilog 2).

CSS Grid

CSS Grid predstavlja novi model organiziranja sadržaja mrežnih stranica koji omogućava visok stupanj kontrole nad blokovima (elementima koji sačinjavaju stranicu) u pogledu njihovog pozicioniranja, veličinama te samim sadržajem unutar tih blokova. Sadržaj stranice organizira se u 2 dimenzije, vertikalno i horizontalno⁵, odnosno u stupce i redove.

Redoslijed kojim su HTML elementi složeni u dokumentu ujedno označava i sam redoslijed kojim će oni biti prikazani na samoj stranici, taj raspored moguće je promijeniti korištenjem različitih tehnika u samom CSS-u, no te tehnike najčešće nisu jednostavne, ni praktične. CSS Grid omogućava točno određeno ili pak automatsko pozicioniranje sadržaja bez obzira na strukturu HTML dokumenta⁶. Element koji je definiran na početku samog dokumenta pri prikazu stranice može biti pozicioniran na samom kraju te stranice. Horizontalna navigacijska traka se s lakoćom može pretvoriti u vertikalnu navigacijsku traku (*sidebar*) upisivanjem samo nekoliko linija koda. Korištenje *grida* u kombinaciji sa *medijskim upitima (media queries)*

⁴ Business Dictionary. URL: <http://www.businessdictionary.com/definition/user-interface.html> (2018-08-29)

⁵ Atkins, Tab Jr.; Etemad, Erika J.; Atanassov, Rosen. CSS Grid Layout Module Level 1., 2017. URL: <https://www.w3.org/TR/css-grid-1/> (2018-08-29)

⁶ Isto.

omogućava visok stupanj kontrole nad prilagođavanjem sadržaja stranice različitim širinama zaslona⁷.

Glavne karakteristike **grida** su mogućnost određivanja fiksnih ili prilagodljivih veličina stupaca i redova. Za potrebe **grida** stvorena je i nova mjerna jedinica **fr (djelić)**⁸. U slučaju da želimo definirati **grid** od tri stupca, jednostavno možemo odrediti svojstvo **grid-template-areas: 1fr 1fr 1fr**, gdje svaki stupac zauzima jedan djelić retka.

Podrška u preglednicima

Kada su u pitanju novi CSS moduli, kao što je **grid**, vrlo je važno u obzir uzeti pregledničku podršku tih modula⁹. Kada su u pitanju nove tehnologije na području CSS-a, čest je slučaj da nemaju podršku u nekim preglednicima, najčešće u već spomenutom Internet Exploreru 11. Podršku kroz različite preglednike najjednostavnije je saznati pregledom podataka na stranici Can I Use¹⁰. Riječ je o stranici koja pruža uvid u pregledničku podršku za različite **frontend** tehnologije. Podaci se ažuriraju redovno, najčešće u roku od nekoliko dana¹¹.

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			63						
			66		10.3				
			67		11.2				4
11	17	61	68	11.1	11.4	all	67	11.8	7.2
	18	62	69	12	12				
		63	70	TP					
			71						

Slika 1. CSS Grid – preglednička podrška prema podacima stranice Can I Use (srpanj, 2018.)

Can I Use podaci pokazuju kako je podrška za **grid** poprilično visoka pa je tako u potpunosti podržan od strane najpopularnijih mrežnih preglednika: Google Chrome, Firefox, Safari i Opera. Internet Explorer 11 i 10 djelomično podržava Grid uz uvjet da se koristi prefiksa.

⁷ Atkins, Tab Jr.; Etemad, Erika J.; Atanassov, Rosen. CSS Grid Layout Module Level 1., 2017. URL: <https://www.w3.org/TR/css-grid-1/> (2018-08-29)

⁸ Basic concepts of grid layout URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout (2018-08-30)

⁹ Andrew, Rachel. Using CSS Grid: Supporting Browsers Without Grid, 2017. URL: <https://www.smashingmagazine.com/2017/11/css-grid-supporting-browsers-without-grid/> (2018-08-29)

¹⁰ Isto.

¹¹ Can I Use. URL: https://caniuse.com/#info_about (2018-08-30)

*-ms*¹². Opera Mini jedini je popularniji mrežni preglednik koji trenutno ne podržava *grid*.

Od ostalih preglednika koji ne podržavaju *grid*, najčešće je riječ o zastarjelim preglednicima, starijim verzijama Google Chromea ili Firefoxa, te naravno, Internet Explorera¹³. S obzirom na to da je stvaranje sučelja mrežne stranice koristeći *grid* bitno drukčije od nekih starijih metoda stvaranja sučelja, postavlja se pitanje vrijedi li uopće uvažavati takve preglednike. U slučaju da je stranica u potpunosti organizirana koristeći *grid*, kako bi bila funkcionalna na jednom od stariji preglednika, njeno sučelje moralo bi biti organizirano nekom drugom metodom, čime se stvara dvostruki posao. Ovo možda ne predstavlja veliki problem kada su u pitanju manje stranice, kao primjerice blogovi i različite statične stranice, no može biti velik problem u slučaju velikih stranica jer se tako troše dodatni resursi.

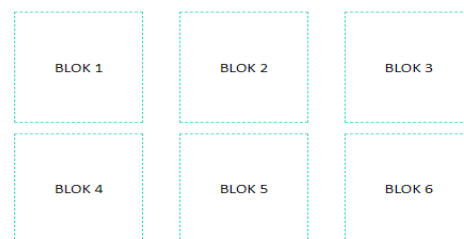
CSS Grid svojstva

Razmak između stupaca i redova

Svojstva *grid-column-gap* i *grid-row-gap* određuju dimenzije linija koje razdvajaju ćelije unutar *grida*. Razdvajanje se događa između stupaca i redaka, ali ne i na njihovim vanjskim rubovima¹⁴. Ova svojstva mogu se skraćeno zajednički definirati kroz svojstvo *grid-gap*. Za potrebe primjera blokovima je određena puna širina stupca te visina 200px.

```
grid-template-columns: 1fr 1fr 1fr;
```

```
grid-gap: 20px 50px;
```



Slika 2. primjer primjene *grid-gap* svojstva

U slučaju deklariranja samo jedne veličine, primjerice *grid-gap: 20px*, razmak dimenzija 20px biti će određen i za stupce i za redove.

¹² Can I Use. URL: https://caniuse.com/#info_about (2018-08-30)

¹³ Isto.

¹⁴ House, Chris. A Complete Guide to Grid, 2018. URL: <https://css-tricks.com/snippets/css/complete-guide-grid/> (2018-08-30)

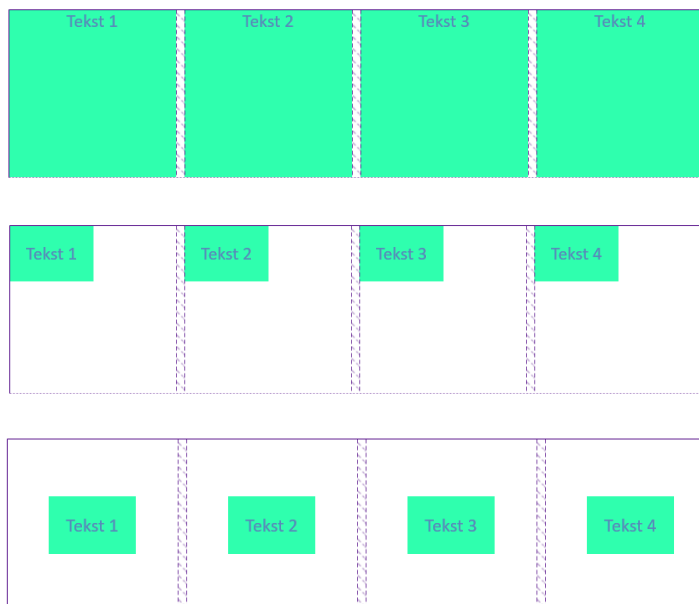
Pozicioniranje elemenata unutar *grida*

Nekoliko je svojstava *grida* odgovorno za određivanje pozicije elemenata unutar ćelija. Riječ je o svojstvima *justify-items* i *align-items* koji se kombiniraju kroz *skraćeno (shorthand)* svojstvo *place-items*, te *justify-content* i *align-content* koji se također kombiniraju u svojstvu *place-content*¹⁵.

Svojstvo *justify-items* odgovorno je za horizontalno poravnanje elementa unutar ćelije. Vrijednosti ovog svojstva su:

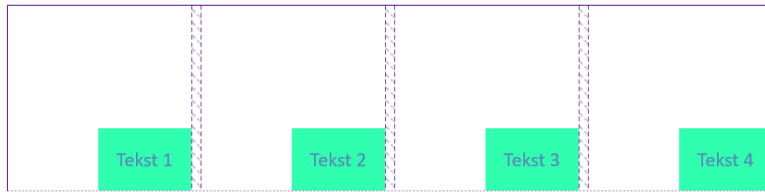
- *start* – pozicionira elemente na početak ćelije
- *end* – pozicionira elemente na kraj ćelije
- *center* – pozicionira elemente u sredinu ćelije
- *stretch* – elementi zauzimaju punu širinu ćelije

Svojstvo *align-items* odgovorno je za vertikalno poravnanje elementa unutar ćelije. Vrijednosti ovog svojstva su jednake su vrijednostima prethodnoga svojstva. S obzirom na to da su vrijednosti oba svojstva jednake, u slijedećem primjeru oba svojstva biti će prikazana istovremeno koristeći svojstvo *place-items*:



Slika 3. gore (place-items: stretch), sredina (place-items: start), dolje (place-items: center)

¹⁵ House, Chris. A Complete Guide to Grid, 2018. URL: <https://css-tricks.com/snippets/css/complete-guide-grid/> (2018-08-30)



Slika 4. *place-items: end*

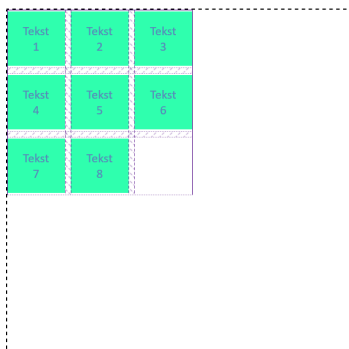
Svojstva *justify-content* i *align-content* razlikuju se od svojstava *justify-items* i *align-items* po tome što umjesto položaja elemenata unutar *grida* oni određuju položaj samih stupaca i redova *grida*.

justify-content horizontalno poravnava stupce i redove. Vrijednosti koje može imati su navedene:

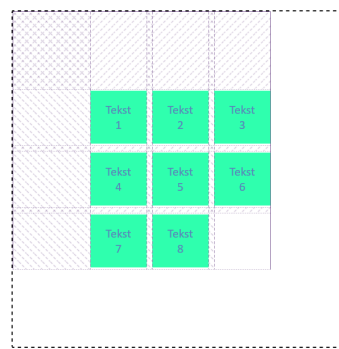
- *start* – poravnanje na početak *grida*
- *end* – poravnanje na kraj *grida*
- *center* – poravnanje u sredinu
- *stretch* – poravnanje punom širinom
- *space-around* – ravnomjieran raspored stupaca sa marginama promjera polovice prostora između stupaca uz desni i lijevi rub *grida*
- *space-between* – ravnomjieran raspored bez margina uz desni i lijevi rub
- *space-evenly* – ravnomjieran raspored među stupcima

align content može sadržavati jednake vrijednosti, samo što se u tom slučaju poravnanje vrši vertikalno.

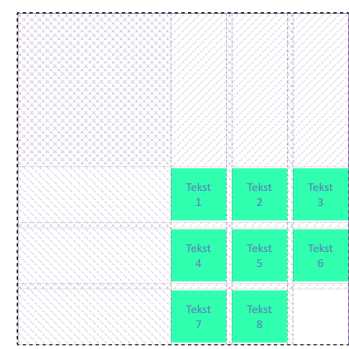
Za potrebe primjera također je upotrijebljeno *skraćeno* svojstvo, u ovom slučaju to je *place-content*:



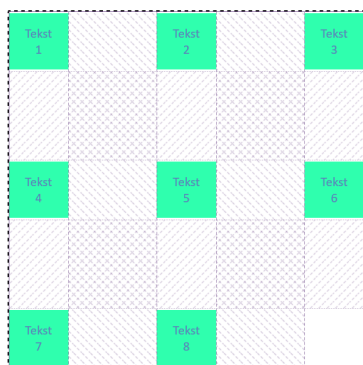
Slika 5. *place-content: start*



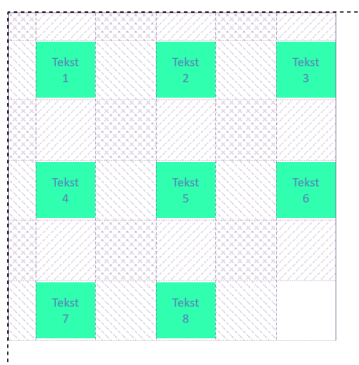
Slika 6. *place-content: center*



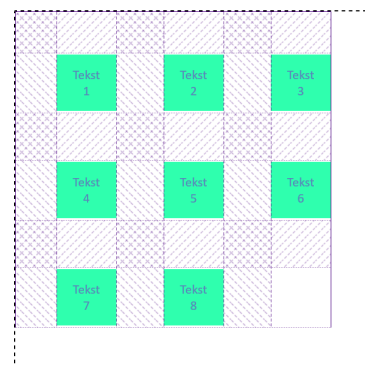
Slika 7. *place-content: end*



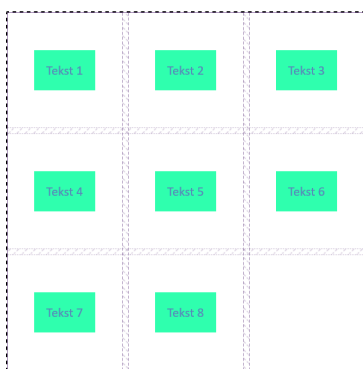
Slika 8 place-content: space-between



Slika 9 place-content: space-around



Slika 10 place-content: space-evenly

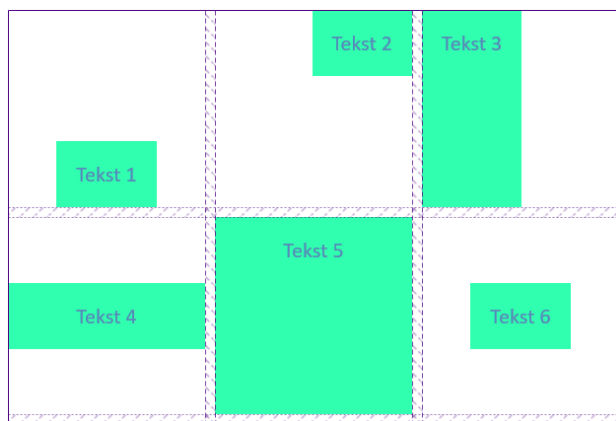


Slika 11. place-content: stretch

Osim na razini samoga **grida**, elemente je moguće pozicionirati i samostalno. Samostalno pozicioniranje elemenata vrši se uporabom **align-self** i **justify-self** svojstava. U ovom slučaju funkcija je ista kao i kad je riječ o prethodno navedenim svojstvima koja vrše pozicioniranje na razini **grida**. Vrijednosti koje mogu imati također su iste kao i kod prethodno navedenih svojstava.

Mogućnosti pozicioniranja sadržaja primjenom različitih svojstava na razini elemenata zaista su

velike. U ovom je primjeru primijenjena kombinacija različitih vrijednosti **align-self** i **justify-self** svojstava.



Slika 12. Pozicioniranje sadržaja primjenom različitih align-self i justify-self vrijednosti

Sadržaji se u **gridu** mogu pozicionirati i na osnovu stupaca i redaka **grida**. Za određeni blok moguće je odrediti da se proteže od početka prvog stupca **grida** sve do kraja trećeg stupca. Može se protezati od početka drugog reda sve do kraja

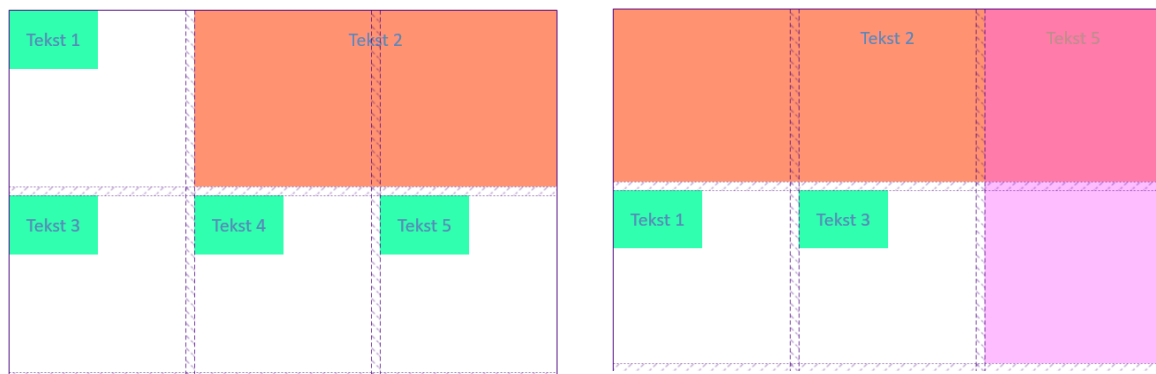
četrtoćg reda. U isto vrijeme se može preklapati sa ostalim elementima koji zauzimaju prostor u tim stupcima i redovima, ili se pak mogu fleksibilno prelomiti kako bi se preklapanje izbjećglo. Za ovakav oblik kontrole nad sadržajem potrebno je primijeniti sljedeća svojstva:

- *grid-column-start*
- *grid-column-end*
- *grid-row-start*
- *grid-row-end*

Navedena se svojstva odnose na toćne linije *grida*. *grid-column-start* i *grid-row-start* odnose se na početne, a *grid-column-end* i *grid-row-end* na završne linije. Vrijednosti mogu biti raspon, broj linije, ime linije ili pak automatski¹⁶. Ova svojstva se također mogu kombinirati kroz *shorthand* svojstva, u ovom su slučaju *grid-column* i *grid-row*.

U prvom je primjeru za element *#tekst2* određeno da se proteže kroz dva stupca (*grid-column: span 2*). Ostali se elementi automatski poravnavaju. Blok iz trećeg stupca premješta se u prvi stupac trećeg reda. U drugom primjeru *#tekst2* proteže se širinom tri stupca te zauzima visinu jednog retka. *#tekst5* proteže se od početka do kraja trećeg stupca, te od početka prvog do početka trećeg reda. Kako bi se postiglo ovakvo preklapanje potrebno je za element odrediti toćne linije.

```
#tekst2 { grid-column: 1 / 4; grid-row: 1 }  
#tekst5 { grid-row: 1 / 3; grid-column: 3 / 4 }
```



¹⁶ House, Chris. A Complete Guide to Grid, 2018. URL: <https://css-tricks.com/snippets/css/complete-guide-grid/> (2018-08-29)

Definiranje područja unutar grida

Vjerojatno najzanimljivije svojstvo *grida* je *grid-template-areas*, koje omogućava točno određen raspored elemenata stranice bez obzira na redoslijed kojim su navedeni u HTML dokumentu. Područje unutar *grida* može se definirati kao logičan prostor u kojemu se raspoređuju jedan ili više elemenata. Prostor čine jedna ili više ćelija, a omeđen je sa četiri *linije*. Ime prostora unutar *grida* dodjeljuje se eksplicitno deklariranjem *grid-template-areas* u kombinaciji sa *grid-area* koji se određuje na razini podređenog elementa unutar *grida*¹⁷. Osim kroz imena, na ime područja može se referirati pozivanjem na linije unutar *grida*.

`<div>` elementu dodijeljena je klasa *.grid*. U njega ulaze svi sadržaji stranice. Svojstvo *grid-template-columns* određuje raspored u 3 stupca, svaki stupac čini 1/3 širine `<div>` elementa. *grid-template-areas* definira raspored *grida*, područjima su dodijeljena proizvoljna imena.

```
.grid{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  grid-template-areas:
    "zaglavlje zaglavlje zaglavlje"
    "navigacija navigacija navigacija"
    "sadrzaj sadrzaj sadrzaj"
    "podnozje podnozje podnozje";}
```

Svaki redak teksta s navodnim znacima označava jedan redak u *gridu*. Svaka riječ unutar navodnika predstavlja jedno područje, te ujedno i jedan stupac. Pošto definirani *grid* čine tri stupca, svaki redak sadrži po tri riječi. Područje *zaglavlje* zauzima 3 stupca prvog retka, *navigacija* 3 stupca drugog retka, te *sadrzaj* 3 stupca trećeg retka, dakle sva 3 područja zauzimaju punu širinu stranice. Nužno je na razini elemenata koji ulaze u *grid* definirati navedena područja.

Svakom od elemenata određuje se svojstvo *grid-area*:

```
.zaglavlje{  grid-area: zaglavlje;}
.navigacija{ grid-area: navigacija;}
```

¹⁷ Atkins, Tab Jr.; Etemad, Erika J.; Atanassov, Rosen. CSS Grid Layout Module Level 1., 2017. URL: <https://www.w3.org/TR/css-grid-1/> (2018-08-29) <https://www.w3.org/TR/css-grid-1/#grid-area-concept> (2018-08-29)

```
.sadrzaj{ grid-area: sadrzaj;}
.podnozje {grid-area: podnozje;}
```

Konačan rezultat izgleda ovako:



Slika 15. Primjer rasporeda korištenjem grid-template-areas svojstva

Raspored elemenata stranice može se vrlo lako promijeniti mijenjanjem rasporeda unutar **grid-template-areas**. U slučaju pretvaranja horizontalne navigacijske trake u bočnu potrebno je odrediti slijedeći raspored:

```
"navigacija zaglavlje zaglavlje"
```

```
"navigacija sadrzaj sadrzaj"
```

```
"navigacija sadrzaj sadrzaj"
```

```
"podnozje podnozje podnozje"
```

U ovom će slučaju navigacija zauzimati prvi stupac prva 3 retka, dok će ostale stupce zauzimati preostali elementi stranice. Također je bitno promijeniti raspored samih poveznica unutar navigacijske trake pošto promjene područja koji ona zauzima unutar **grida** neće imati nikakav utjecaj na elemente unutar navigacijske trake. Raspored tih elemenata potrebno je promijeniti iz horizontalnog smjera u vertikalni. Poveznice unutar navigacijske trake nalaze se unutar `` (neredani popis) elementa. Najjednostavnije je rješenje određivanje **display: flexbox**

svojstva za `` element. Određivanjem *flex-direction: column* podređeni elementi unutar popisa posložiti će se u vertikalnom smjeru:

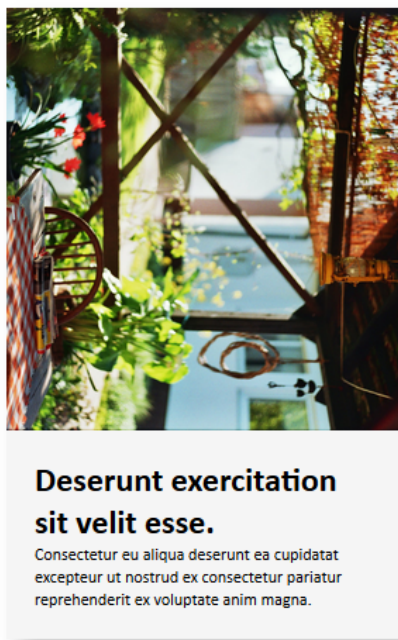
```
.navigacija{
  grid-area: navigacija;
  ul{
    display: flex;
    flex-direction: column;
  }
}
```



Slika 16. Promjena rasporeda stranice primjenom grid-template-areas

Razvijanje korisničkog sučelja prilagodljivog različitim širinama zaslona

Za potrebe primjera stvarne upotrebe *grida* odabrana je galerija fotografija. Sačinjena je blokova koji sadrže fotografiju te određene informacije (naslov i opisni tekst) o toj fotografiji.



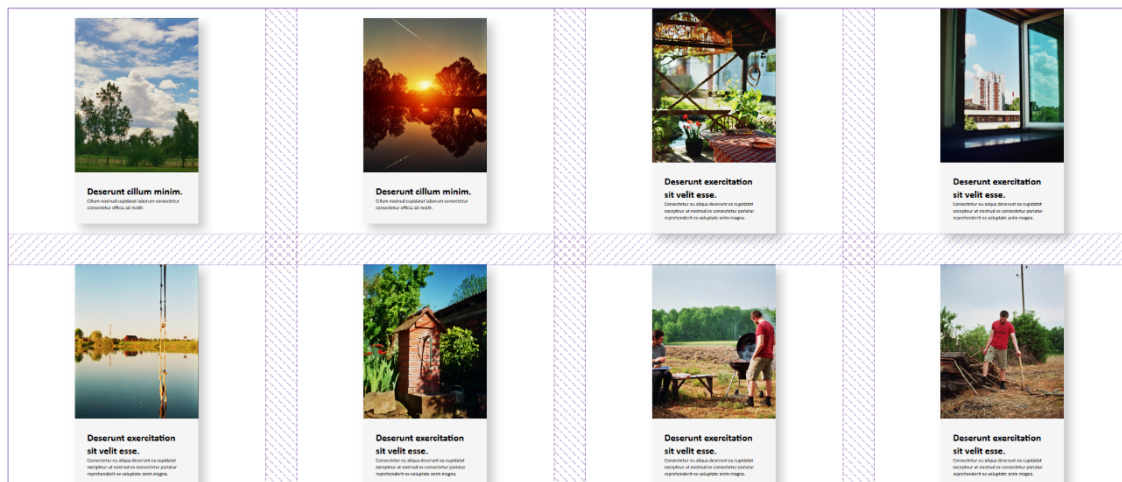
```
<div class="blok">
  <div class="blok__fotografija">
    
  </div>
  <div class="blok__info">
    <h1>Naslov</h1>
    <p>Opis fotografije</p>
  </div>
</div>
```

Slika 17. Primjer bloka

Galerija fotografija naizgled može biti jednostavna, posebice jer često ima jednostavnu strukturu. Usprkos toj relativno jednostavnoj strukturi, problem može nastati pri prilagođavanju različitim širinama zaslona. Na zaslonu širine 320px nemoguće je prikazati tri bloka fotografija širine 200px. U slučaju kada su blokovi različitih širina i visina, njihovo pozicioniranje primjenom konvencionalnih CSS modula i njihovih svojstava može biti poprilično složeno i zahtjevno, posebice po pitanju prilagođavanja različitim širinama zaslona. Izuzetak tomu je **flexbox**, relativno novi CSS modul sličan **gridu**, s kojim dijeli i određena svojstva (**align-items**)¹⁸.

U slučaju ovakve galerije čest problem može biti visina tekstualnih informacija koja utječe na cjelokupnu visinu bloka. Različiti blokovi imaju različite veličine tekstova te su samim time različitih visina. Korištenjem **grida** njihovo je pozicioniranje vrlo jednostavno.

¹⁸ Relationship of grid layout to other layout methods, 2018. URL:https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Relationship_of_Grid_Layout



Blokovi su smješteni u **grid** za koji su definirana slijedeća svojstva:

```
.galerija{
    display: grid;
    grid-template-columns: 200px 200px 200px 200px;
    place-content: center;
    place-items: center;
    grid-gap: 50px;
}
```

Broj stupaca iznosi 4. Svojstvom **place-content** određeno je pozicioniranje stupaca i redova u središte **grida**. Svojstvom **place-items** određeno je srednje vertikalno i horizontalno poravnanje blokova unutar **grida**. Razmak među stupcima i redovima je 50px, no stvarni razmak između elemenata veći je od toga. Razlog tomu je širina bloka 200px. U zaslona manjih širina, kao što je 1024x768x broj stupaca trebao bi se smanjiti sa 4 stupca na 3. U slučaju kad se broj stupaca ne bi smanjio, sveukupna *širina* blokova biti će veća od širine **grida**, a samim time dolazi do „probijanja“ okvira **grida**.

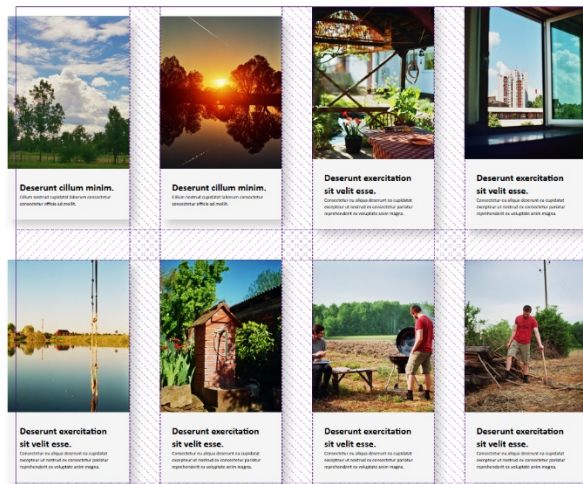
Konvencionalna metoda za određivanje broja stupaca ovisno o širini zaslona je korištenje **medijskih upita (media query)**:

```
@media only screen and (min-width: 512px) {
    .galerija { grid-template-columns: 1fr; }
}

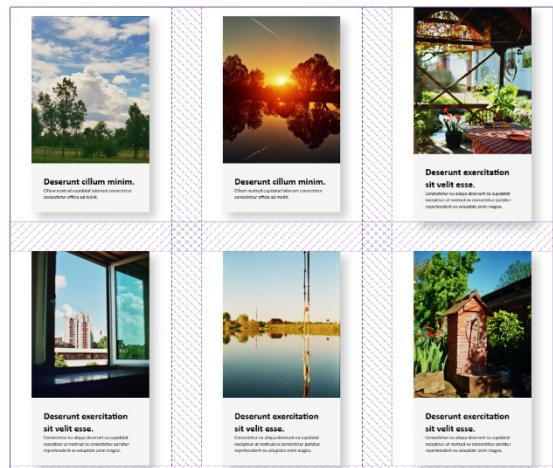
@media only screen and (min-width: 1024px) {
```

```
.galerija { grid-template-columns: 1fr 1fr 1fr; }
}
```

U ovom će slučaju broj stupaca na zaslonima manjima od 1024px širine iznositi 3. S obzirom na to da širina blokova iznosi 200px za najmanje ekrane bilo bi poželjno odrediti 1 stupac. Primjenom *medijskih upita* može se odrediti da broj stupaca na zaslonima širine manje od 512px broj stupaca iznosi 1.



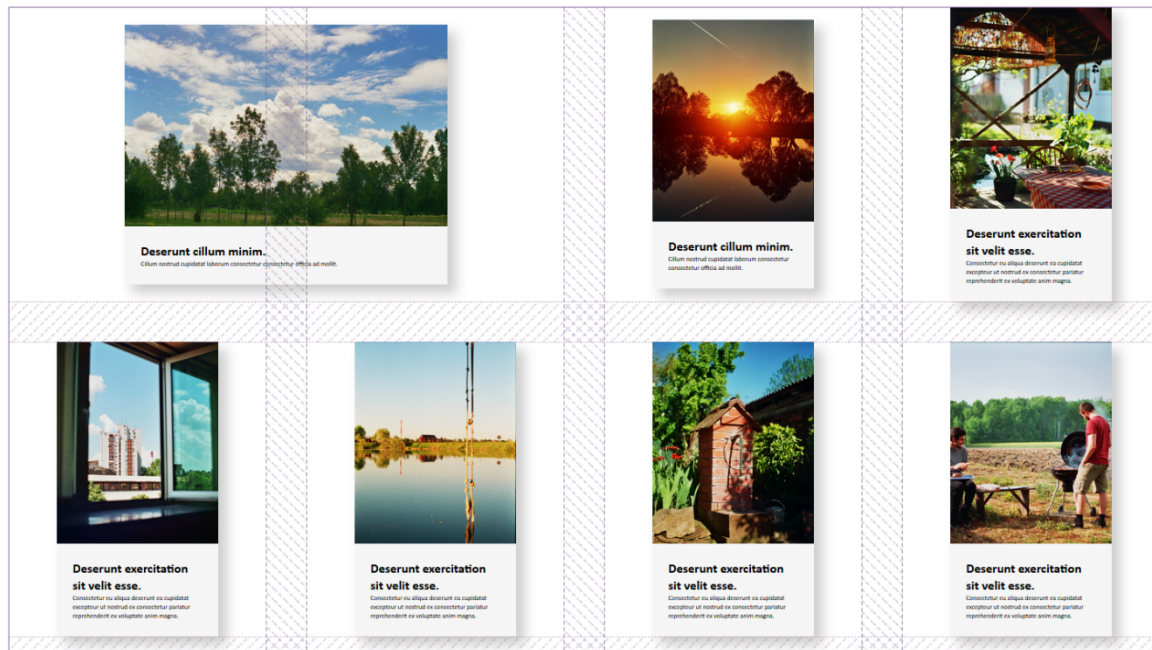
Slika 19. Nepravilan prikaz na zaslonu dimenzija 1024x768



Slika 20. Pravilan prikaz na zaslonu dimenzija 1024x768

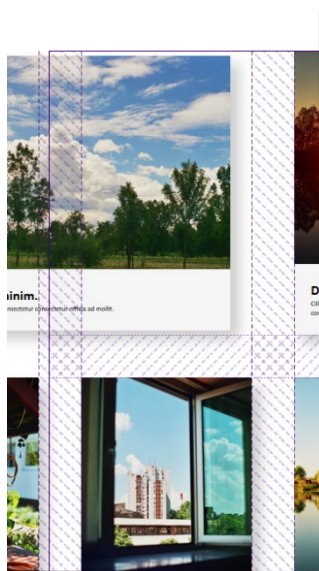
Za fotografije je karakteristično da mogu biti orijentirane horizontalno ili vertikalno, iako je danas poprilično popularan i kvadratni format fotografije (primjerice na društvenim mrežama poput Instagrama). Neke fotografije u ovoj galeriji su orijentirane horizontalno, neke vertikalno. Kako bi se omogućio pravilan prikaz horizontalne fotografije, blok koji sadrži tu fotografiju trebao bi biti širine najmanje 2 stupca. Takvim se blokovima može dodati klasa *.horizontalna-fotografija* te odrediti svojstvo *grid-column: span 2*. U ovom slučaju nužno je prilagoditi i širinu bloka.

```
.horizontalna-fotografija {
  grid-column: span 2;
  width: 400px;
}
```

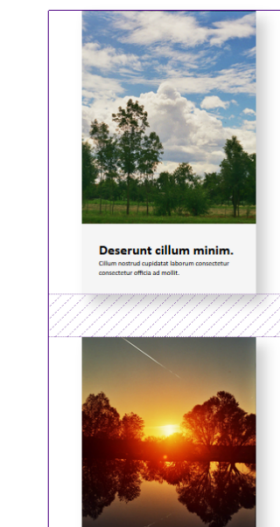


Slika 21. Horizontalna fotografija u galeriji

Svojstvo **grid-column** bitno je prilagoditi određenim širinama zaslona. Blok ne može zauzimati širinu 2 stupca ukoliko je na zaslonu određen prikaz samo 1 stupca. U tom slučaju dolazi do prijelaza blokova van okvira **grida**.



Slika 23. Nepravilan prikaz na zaslonu pametnog telefona



Slika 22. Pravilan prikaz na zaslonu pametnog telefona

Zaključak

CSS Grid rasporedni modul programerima mrežnih stranica pruža visok stupanj kontrole nad pozicioniranjem sadržaja stranice. Tu visoku razinu kontrole omogućuje nekoliko svojstava, ponajprije su to svojstva za pozicioniranje sadržaja na razini **grida**: *justify-items* i *align-items* te *justify-content* i *align-content*. Primjena ovih svojstava na razini elementa omogućuje točno određeno pozicioniranje elemenata, a kada se kombinira sa mogućnošću ciljanja određenih linija **grida** programeru se omogućuje zaista visok stupanj preciznosti i kontrole. Prostor koji određeni element zauzima u **gridu** može se promijeniti izmjenom samo jedne linije koda. Ovakav zadatak korištenjem je konvencionalnih metoda i modula znatno kompleksniji. Uklanjanje te kompleksnosti ujedno označava čišći i pregledniji CSS kod. Čist i pregledan kod od velike je važnosti posebice kada su u pitanju timski projekti, bilo da je riječ o manjem ili velikom timu. Vrlo je bitno da svi sudionici u izradi stranice mogu lako razumjeti napisani CSS kod. CSS, sam po sebi, nije težak za razumjeti, niti je jako kompleksan. Najčešći problem predstavlja nepreglednost zbog velike količine primijenjenih CSS svojstava. Korištenje modernih CSS modula poput **grida** i **flexboxa** u kombinaciji sa CSS predprocesorima poput SASS-a može znatno poboljšati preglednost koda.

Nadalje, vrlo zanimljivo svojstvo **grida** je mogućnost imenovanja, ciljanja i raspoređivanja područja unutar **grida**. To svojstvo je *grid-template-areas* koje se određuje na razini **grid** elementa/okvira, te *grid-area* koje se dodjeljuje podređenim elementima unutar navedenog elementa/okvira. Sam način na koji se ta područja raspoređuju poprilično je neobičan i zanimljiv, znatno je drukčiji od svih ostalih CSS svojstava. Može se usporediti sa stvaranjem ASCII grafika, zbog čega na prvi pogled može djelovati pomalo „primitivno“, no sama jednostavnost ovakvog rasporeda elemenata stranice iznenađujuće je korisna. Ipak, ovo svojstvo najbolje bi bilo primijeniti za neka jednostavnija područja ili module stranice pošto pisanje kompleksnih struktura može postati komplicirano.

Poprilično visoka razina implementacija **grida** u mrežnim preglednicima u relativno kratkom vremenu svjedoči revolucionarnosti koju ovaj modul donosi u područje organiziranje sadržaja stranice. Ipak, korištenje **grida** ne znači da se ostali CSS moduli iste ili slične namijene trebaju zanemariti. Upravo suprotno, istovremena upotreba nekoliko različitih modula, primjerice, **grida**, **flexboxa** različitih *relativnih* i *apsolutnih* pozicija, *margin* i *padding* programeru pruža sve potrebne alate za iznimno intuitivno i jednostavno stvaranje mrežne stranice ili aplikacije. Uporaba **grida** u kombinaciji s *medijskim upitima* ključna je za uspješno stvaranje kompleksnih

ili manje kompleksnih mrežnih stranica i aplikacija prilagodljivih različitim širinama zaslona. Prilagodavanje sadržaja stranice različitim širinama zaslona nije jednostavno, no na ovaj način ono postaje znatno jednostavnije nego što je to prije bilo moguće.

Literatura

1. Andrew, Rachel. Using CSS Grid: Supporting Browsers Without Grid, 2017. URL: <https://www.smashingmagazine.com/2017/11/css-grid-supporting-browsers-without-grid/> (2018-08-29)
2. Atkins, Tab Jr.; Etemad, Erika J.; Atanassov, Rosen. CSS Grid Layout Module Level 1., 2017. URL: <https://www.w3.org/TR/css-grid-1/> (2018-08-29)
3. Basic concepts of grid layout URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout (2018-08-30)
4. Business Dictionary. URL: <http://www.businessdictionary.com/definition/user-interface.html> (2018-08-29)
5. Can I Use. URL: https://caniuse.com/#info_about (2018-08-30)
6. House, Chris. A Complete Guide to Grid, 2018. URL: <https://css-tricks.com/snippets/css/complete-guide-grid/> (2018-08-30)
7. Net MarektShare. URL: <https://netmarketshare.com/browser-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%2Fflaptop%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Trend%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22browser%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22browsersDesktop%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222017-08%22%2C%22dateEnd%22%3A%222018-07%22%2C%22segments%22%3A%22-1000%22%7D> (2018-08-29)
8. StatCounter GlobalStats. URL: <http://gs.statcounter.com/screen-resolution-stats> (2018-08-29)
9. Relationship of grid layout to other layout methods, 2018. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Relationship_of_Grid_Layout
10. W3Counter. URL: <https://www.w3counter.com/globalstats.php> (2018-08-29)

Prilozi

1. *Slika 1.* - Can I Use. URL: https://caniuse.com/#info_about (2018-08-30)
2. Github repozitorij. URL: <https://github.com/nmilic96/zavrsni>