

ИЗРАДА СИМУЛАЦИЈЕ РЕДИЗАЈНА WORDPRESS ВЕБ САЈТА

Тања Крунић¹, Петар Васиљевић²

Резиме: У оквиру овог рада је приказана примена *Inspect element* алата у циљу симулације редизајна веб сајта Високе техничке школе струковних студија у Новом Саду креираног у *WordPress*-у. На основу симулације је добијен скуп *CSS* правила које треба изменити на сајту како би се извршио његов редизајн. Овакав приступ представља најбржи и најједноставнији пут ка модернијем изгледу веб сајта, који се може извршити без да сајт или неки његови делови буду недоступни одређено време. У раду су приказани кључни аспекти редизајна, од инспекције *HTML* и *CSS* кода у својству проналажења класа које треба изменити, до *CSS* правила која потом треба применити у стилу сајта да би се жељени редизајн извршио.

Кључне речи: *Inspect element* алат, *responsive design mode*, редизајн веб сајта, *WordPress*

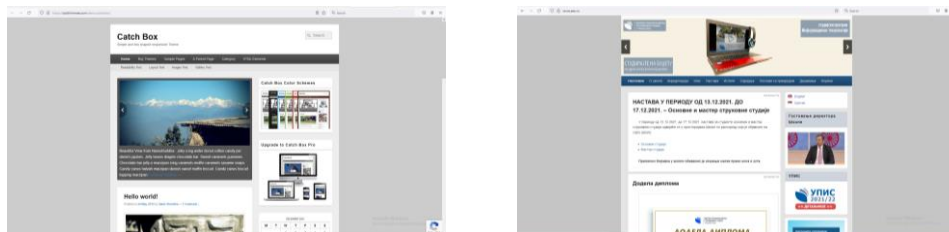
CREATING A SIMULATION OF REDESIGN OF A WORDPRESS WEBSITE

Abstract: Herein we show the use of the *Inspect element* tool in order to get a simulation of redesign of The Higher Education Technical School of Professional Studies in Novi Sad website created in *WordPress*. Based on the simulation, a set of *CSS* rules is obtained that need to be changed on the website in order to redesign it. This approach is the fastest and easiest way to obtain a more modern look of the website, which can be done without the website or some of its parts being inaccessible for some time. The paper presents the key aspects of redesign, from inspection of *HTML* and *CSS* code to find classes to be modified, to *CSS* rules which should then be applied in the website style in order to perform the desired redesign.

Key words: *Inspect element* tool, *responsive design mode*, web site redesign, *WordPress*

1. УВОД

Развој тренутно активног веб сајта Високе техничке школе струковних студија у Новом Саду (ВТШ НС) започет је у јуну 2013. године. За основу је изабран *WordPress*, верзија 3.1. У септембру 2013. године инсталирана је *WordPress* тема *Catch Box*, верзија 2.7. У том периоду то је била једина бесплатна а респонсивна (тема која се прилагођава различитим уређајима, [1]) *WordPress* тема. Тренутно је доступна верзија 4.8.9 ове теме [2]. Више детаља о верзијама ове теме се може наћи у [3]. У суштини, задржан је основни изглед верзије, Сл. 1 (лево). Примена ове верзије захтева инсталацију *WordPress*-а најмање верзије 5.1, [4]. На Сл.1 (десно) је приказан веб сајт ВТШ НС који користи ову тему. Као што можемо видети, веб сајт (као и сама тема) је одређене фиксне ширине, што је тренутно једна од главних замерки изгледа веб сајта.



Слика 1- *WordPress* тема *Catch Box* (лево), веб сајт ВТШ НС (десно)

¹ Др, Висока техничка школа струковних студија у Новом Саду, Школска 1, Нови Сад, krunic@vtsns.edu.rs

² Спец. инж., Висока техничка школа струковних студија у Новом Саду, Школска 1, Нови Сад, vasiljevic@vtsns.edu.rs

Почетком 2021. године донета је одлука да се изврши унапређење веб сајта преласком на новију верзију *WordPress*-а, уз задржавање постојеће теме, структуре и свих садржаја на веб сајту. Након ажурирања *WordPress*-а на најновију верзију, евентуално би се размотрила опција да се постојећа тема замени неком другом темом која заузима читаву ширину екрана. Међутим, познато је да промена теме већ од раније активног веб сајта са собом носи још један проблем, а то је могућност да нестану навигациони менији, вицети, сајдбар, и сл., па се мора уложити додатни напор и време док се сви елементи сајта поново ставе у функцију и провери њихов рад, [5].

Направљен је план активности који подразумева постепено унапређење верзије *WordPress*-а. Након почетне верзије 3.1, инсталиране су редом верзије 3.2.1, 3.3.3, 3.4.2 итд. Цео процес инсталације једне новије верзије *WordPress*-а подразумева:

1. Прављење резервне копије комплетног веб сајта (свих фајлова и базе података).
2. Постављање свих фајлова новије верзије *WordPress*-а путем FTP-а.
3. Проверу функционалности комплетног веб сајта (*WordPress* теме и свих *plugin*-ова) након инсталације новије верзије *WordPress*-а.

Овакав процес инсталације једне новије верзије *WordPress*-а траје у просеку око 4 сата, где највећи део времена одлази на прављење резервне копије комплетног веб сајта. Тренутно инсталирана верзија је 4.5.24. У међувремену, повећавањем верзије *WordPress*-а, поједини застарели *plugin*-ови су престали да функционишу. Код неких је проблем решен ажурирањем самих *plugin*-ова, док код других то није било могуће, те се морало приступити тражењу и инсталацији нових *plugin*-ова сличне функционалности. Тренутно је на веб сајту активан укупно 21 *plugin* и сви су ажурирани на најновије верзије. У наредном периоду наставиће се са унапређењем верзије *WordPress*-а, све док се не стигне до последње актуелне верзије 5.8.2, [6].

С обзиром на чињеницу да креирање резервних копија и тражење новијих верзија *plugin*-ова захтева доста времена и истраживачког рада, као и горе поменуто чињеницу да промена теме захтева поновну активацију или додавање неких делова веб сајта, тражила се нека алтернативна могућност да се дође до жељеног редизајна веб сајта. У ту сврху је урађено ово истраживање, чија је основна идеја да се нађе скуп правила у *CSS*-у постојеће теме које треба изменити да би се добио веб сајт пуне ширине. На овај начин би се могао редизајнирати сајт у много краћем року, тј. пре него што се изврши ажурирање до најновије верзије *WordPress*-а. При томе, треба водити рачуна о респонсивности, односно, да се приликом прегледа веб сајта на мањим уређајима, као што су мобилни телефони, сви елементи сајта приказују коректно.

У другом поглављу овог рада је описана методологија истраживања, тј. начин на који се може доћи до неопходног скупа *CSS* правила које треба изменити да би се веб сајт редизајнирао тако да заузима целу ширину екрана на свим уређајима. У овом поглављу је описан и алат којим се оваква симулација може изврши. У трећем поглављу је описано добијање скупа *CSS* правила коришћењем претходно описаног алата за симулацију измене стила.

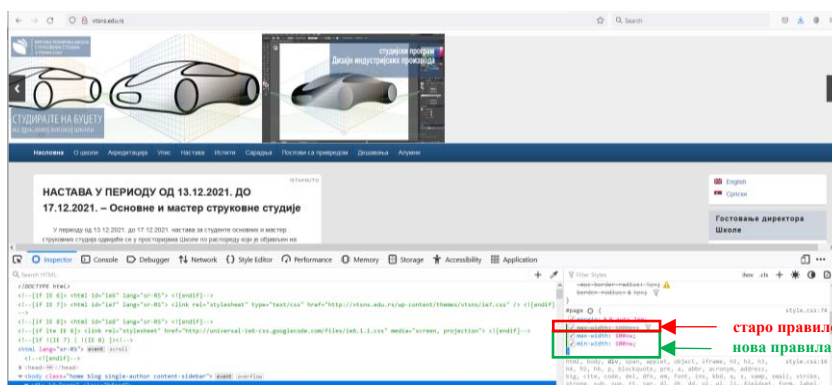
2. МЕТОДОЛОГИЈА ИСТРАЖИВАЊА – КОРИШЋЕНИ АЛАТИ

Као што је већ напоменуто, једно од једноставнијих решења за промену стила веб сајта ВТШ НС, а које се може урадити не чекајући ажурирање верзија *WordPress*-а, јесте израда симулације редизајна сајта у сврху добијања одговарајућег скупа *CSS* правила које треба применити директно у *WordPress* платформи, да би се сајт редизајнирао без промене теме. У сврху израде симулације редизајна веб сајта

коришћен је алат *Inspect element*. Овај алат нам омогућава да десним кликом на било који елемент веб странице, добијемо увид у његов *HTML* код, као и у одговарајуће *CSS* класе и правила којима је стилизован тај елемент, [7]. Потом можемо на одговарајућем месту у коду вршити измене да бисмо видели како оне изгледају у веб читачу. Наравно, ове измене су виртуелне, тј. у питању је симулација. Уколико нисмо задовољни добијеним изменама, можемо их мењати, и на крају добити скуп правила која можемо прекопирати из овог алата у одговарајући текст едитор, како бисмо их сачували и после извршене симулације извршили у *WordPress* платформи измену *CSS* кода по овом списку правила. У оквиру овог алата постоји могућност прегледа измена које симулирамо на различитим ширинама екрана, тако да у сваком тренутку можемо видети како ће наше измене изгледати нпр. на екрану мобилног телефона, [8]. Након добијеног скупа правила које доводе до жељеног редизјана сајта, неопходно је извршити измену стила на самом веб сајту. У *WordPress*-у, изглед веб сајта, као и његове функционалности су дефинисане у теми која је одабрана, [9]. Тако да, уколико желимо нешто да мењамо, морамо изменити тему. При томе, препоручује се да се не врше директне измене теме, већ најпре креира тзв. *child* тема и на њој врше одговарајуће измене, [10]. У оквиру овог рада је приказана само израда симулације редизјана, а више детаља о томе како се мења *CSS* код у *WordPress*-у може се наћи у [11]- [13].

3. СИМУЛАЦИЈА РЕДИЗАЈНА ВЕБ САЈТА ВИСОКЕ ТЕХНИЧКЕ ШКОЛЕ СТРУКОВНИХ СТУДИЈА У НОВОМ САДУ

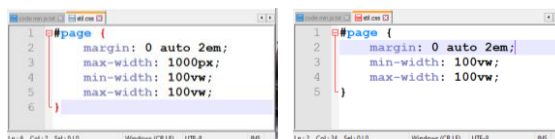
За почетак, неопходно је пронаћи главни дивајдер у телу веб сајта, и подесити му минималну и максималну ширину на вредност од *100vw*. Овим смо ширину која је у темплејту сајта била фиксна, и задата у апсолутним јединицама, претворили у релативну. Наиме, *1vw* заправо представља 1% ширине екрана уређаја (енгл. *Viewport Width*), [14]. Ово значи да главни дивајдер веб сајта заузима 100% ширине екрана. Као што можемо видети на Сл. 2, главни дивајдер има *id="page"* и био је иницијално стилизован тако да му је максимална могућа ширина *1000px*. Овом дивајдеру постојеће правило *max-width:1000px* замењујемо правилима *max-width:100vw; min-width: 100vw*. При томе, *CSS* својство *min-width* представља најмању могућу ширину елемента који се стилизује, док *max-width* представља његову највећу могућу ширину, [15].



Слика 2 – Подешавање минималне и максималне ширине главног дивајдера сајта

На крају, измене чувамо у одговарајућем текст едитору, Сл. 3 (лево). Наравно, при томе бришемо стара правила која нам више нису потребна, Сл. 3 (десно). Као резултат промена које смо урадили у коду, веб сајт се сада протеже читавом ширином екрана, али слајдер и даље заузима само један део те ширине, Сл. 2. Зато је следећа ствар коју

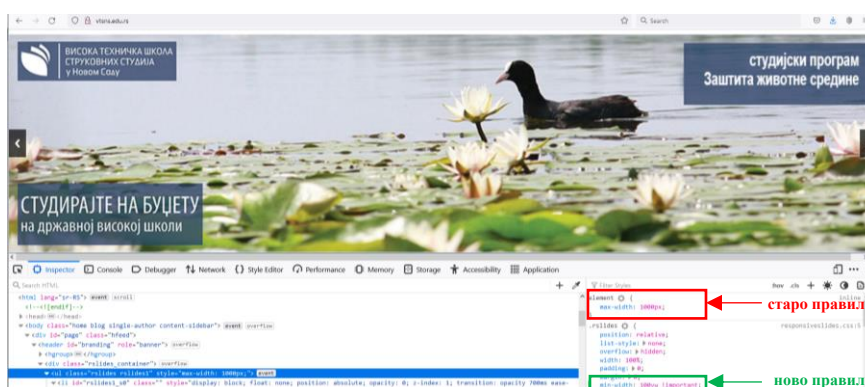
треба да изменимо ширина слајдера. Увидом у *HTML* код странице, Сл. 4, учачамо да се слајдер заправо састоји од неуређене листе чији су елементи слике, које се смењују на екрану помоћу одговарајуће скрипте. Овој листи је придружена класа *rslides*. Као што видимо, овој листи (тј. слајдеру) је подешена максимална ширина на *1000px*. Ово ћемо изменити додавањем наредбе *min-width:100vw !important*; тј. подешавањем минималне ширине на *100vw*. Након ове измене, слајдер је коначно заузео целу ширину екрана.



```
1 #page {
2   margin: 0 auto 2em;
3   max-width: 1000px;
4   min-width: 100vw;
5   max-width: 100vw;
6 }
```

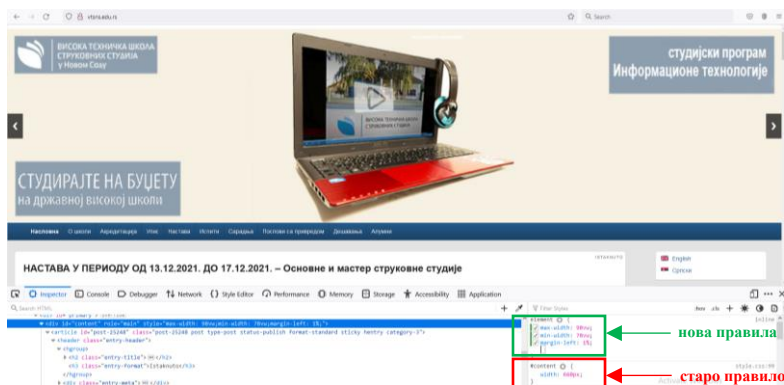
Слика 3 – Чување одговарајућих CSS правила у текст едитору

Међутим, као што видимо на Сл. 2, након наших измена, дошло је до раздвајања садржаја левог дела странице сајта која се састоји од чланака са актуелним постовима и сајдбара који се налази са десне стране. Између њих се појавио значајан размак. Овај проблем решавамо тако што ћемо помоћу *Inspect element* алата пронаћи дивајдер који садржи те чланке. Ако погледамо Сл. 5, можемо закључити да се ради о дивајдеру који има *id="content"*, те да му је првобитна ширина постављена на *660px*. Уместо ове вредности поставићемо максималну ширину на *98vw*, а минималну на *78vw*. Ово значи да ће се приликом прегледа веб сајта на мобилном телефону, дивајдер са постовима раширити на готово читаву ширину екрана, с тим што ћемо из естетских разлога оставити по 1% размака са леве и десне стране овог дивајдера од ивице екрана. Ово решавамо постављањем маргине на 1%. У овом случају, сајдбар ће такође заузети исту ширину екрана (98%), и бити распоређен испод дивајдера са постовима. С друге стране, приликом прегледа веб сајта на уређајима са ширим екранима, исти дивајдер заузима нешто мање од 78% ширине екрана, остављајући места за сајдбар који треба да заузме око 18% ширине екрана. Преостали део је предвиђен за маргине између дивајдера са чланцима, сајдбара и ивица екрана.



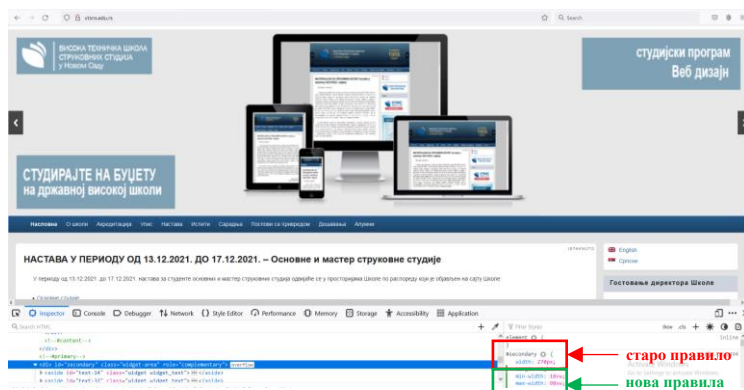
Слика 4 –Имена класе слајдера

Након подешавања минималне и максималне ширине дивајдера са чланцима, неопходно је урадити исто и са сајдбаром. Сл. 6 нам указује на чињеницу да је сајдбар заправо дивајдер чији је *id="secondary"*, те да му је иницијално одређена фиксна ширина од *270px*. Измену вршимо тако што ћемо додати правила: *min-width:18vw*, *max-width:98vw*; *margin-right:1*. Овим подешавамо да се ширина сајдбара при различитим ширинама екрана понаша на горе описан начин.



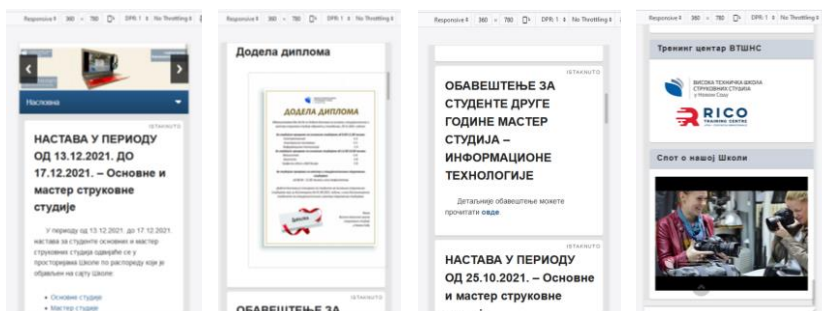
Слика 5 – Подешавање минималне и максималне ширине дивајдера са чланцима

На Сл. 6 видимо да, након до сада урађених измена у коду, веб сараница изгледа сасвим коректно на уређајима са широким екранима. Приказ исте странице сајта на Сл. 7 добијен је коришћењем *responsive design* мода *Inspect element* алата. При томе смо одабрали димензије $360 \times 780 \text{px}$, што представља једну од препоручених димензија за тестирање изгледа веб сајта на мобилним телефонима у [16].



Слика 6 – Подешавање минималне и максималне ширине сајдбара

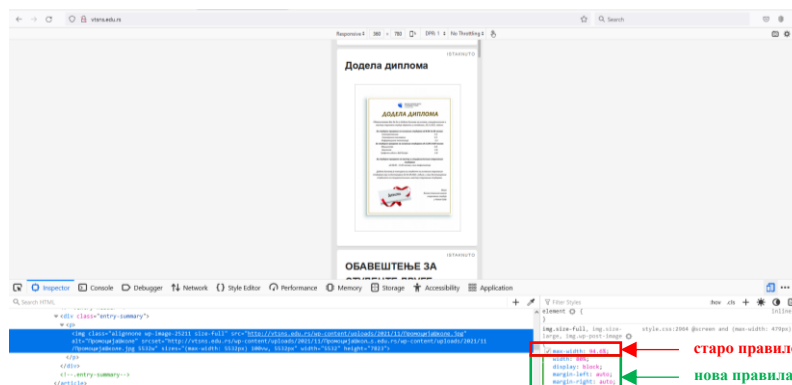
Као што можемо видети, распоред елемената на страници је сасвим коректан, али постоје неки детаљи који треба да се додатно подесе, како би сајт изгледао лепше на мобилним телефонима. Пре свега, ово се односи на слике унутар дивајдера са постовима (друга слика с лева) које су благо диспозициониране, као и на лево поравнање текста у насловима постова (прва и трећа слика с лева).



Слика 7 - Приказ добијених промена у *responsive design* моду *Inspect element* алата за екране димензија $360 \times 780 \text{px}$

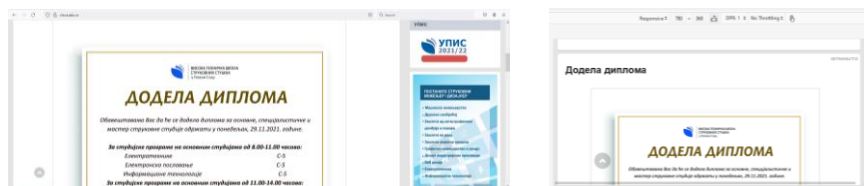
Кренимо најпре од поправљања стила за слике унутар дивајдера са постовима. Применом *Inspect element*-а проналазимо дотичну слику и запажамо да јој је придружена класа *size.full*. У овој класи смо затекли својство *max-width:94.6%*, Сл. 8. С обзиром да је увек ефектније правити измене у класама него директно на елементу, како би промена важила за све елементе исте класе, а не само за један који тренутно

мењамо, изменили смо стил управо ове класе додавањем CSS правила `width:80%; display:block; margin-left:auto; margin-right:auto`. Уколико елемент (у нашем случају слика) има својство `display:block` и одређену ширину (у нашем случају, то је 80% ширине дивајдера у коме се налази слика), ово се користи за центрирање слике, [17], [18]. Као што видимо, након поменутих измена у стилу, слика је успешно центрирана и лепо изгледа на мобилним телефонима.



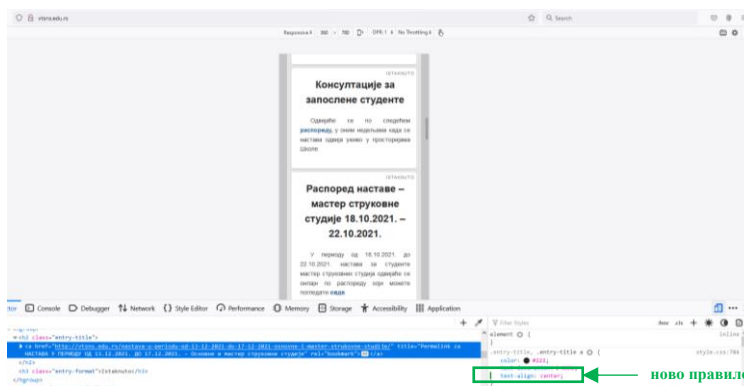
Слика 8 – Измена стила класе за слике у циљу њиховог центрирања

На Сл. 9 (лево) видимо да се слика такође коректно приказује на десктоп рачунарима. Исто проверавамо и за тзв. *Landscape* оријентацију екрана мобилног телефона, и констатујемо да је приказ коректан, Сл. 9 (десно).

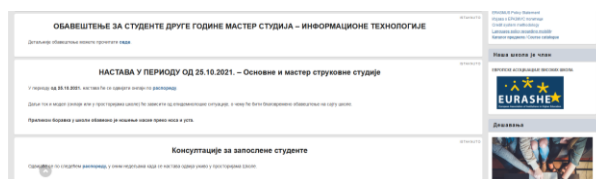


Слика 9 – Изглед слике на рачунару (лево) и на мобилном телефону у тзв. *Landscape* оријентацији (десно)

Ако погледамо Слика 7 можемо уочити да су наслови текста у чланцима лево поравнати, што не изгледа лепо на мобилним телефонима уколико је наслов чланка дужи. Стога проналазимо да су у питању заправо наслови другог реда са класом `entry-title`. Наслов центрирамо додајући правило `text-align:center`, Сл.10 Као што видимо, наслов је сада центриран. Ако погледамо исти део сајта у моду за широке екране, можемо закључити да ово такође изгледа сасвим коректно, Сл. 11.

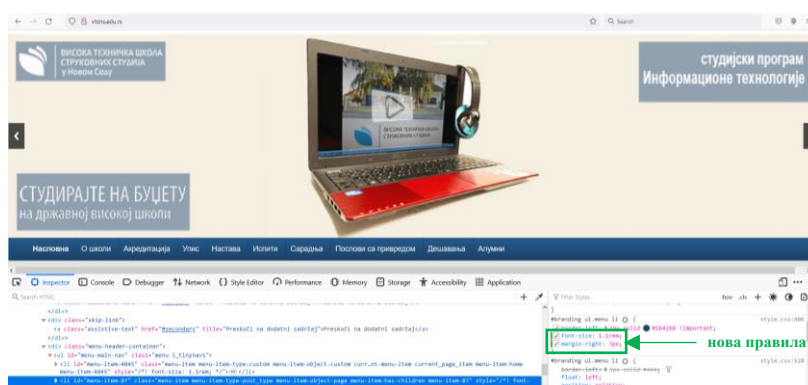


Слика 10 - Центрирање наслова чланка у моду за мобилне телефоне



Слика 11 – Изглед дивајдера са чланцима након центрирања наслова у моду за широке екране

На крају, ако погледамо Слика 6, видећемо да су линкови у навигационом менију приказани доста ситним фонтом и да је размак између њих мали. Разлог за ову појаву лежи у томе што је сајт до сада био фиксне ширине, па је простор у хоризонталном навигационом менију био ограничен. Управо због тога се претходних година прибегавало смањивању фонтова линкова, као и размака између њих, како би неопходни линкови могли да стану у један ред. С обзиром да сада веб сајт заузима читаву ширину екрана, фонтов, као и размак између линкова, слободно може да се повећа, што је и учињено додавањем правила `font-size:1.1rem; margin-right:7px;` у навигационим елементима означеним са `#branding ul.menu li`, Сл. 12. Према старим подешавањима, величина фонтова је била фиксна, тј. `14px` (по принципу наслеђивања од `body` тага), док је сада искоришћена релативна јединица `rem`. При томе, `1 rem` представља величину фонтова који одговара `html` тагу (у већини читача то је `16px`), [19], па `1.1 rem` у ствари представља фонтов који је увећан за `10%` у односу на ову величину.



Слика 12 – Повећање фонтова линкова и подешавање размака између њих

4. ЗАКЉУЧАК

Информационе технологије се развијају изузетно великом брзином. Да бисмо држали корак са временом, неопходно је пратити нове технологије у иградњи веб сајтова. Такође је важно редовно ажурирање верзија технологија које се користе, да бисмо могли користити њихове нове особине и могућности. Међутим, понекад трка са технологијама захтева да уложимо много труда и издвојимо пуно времена, којег у овим убрзаним временима нема напретек. Управо из тог разлога је понекад сасвим прихватљиво тражити тзв. међурешења. У оквиру овог рада је приказана израда симулације редизајна сајта ВТШ НС, који се потом лако може имплементирати у `WordPress` окружењу. Тиме је омогућено да сајт добије модернији изглед пре него што се заврши дугорочни процес ажурирања верзија. Такође, измене стила сајта добијене у овом истраживању се могу применити без да делови сајта буду привремено недоступни, што би се десило променом `WordPress` теме.

5. ЛИТЕРАТУРА

- [1] Fiedman V., (2018), *responsive design – what it is and how to use it*, *Smashing magazine*, <https://www.smashingmagazine.com/2011/01/guidelines-for-responsive-web-design/>
- [2] *Catch box, Simple and box shaped responsive Theme*, (2021), <https://catchthemes.com/demo/catchbox/>
- [3] *Catch Box Theme Change Log*, (2021), <https://catchthemes.com/changelogs/catch-box-theme/>
- [4] *Catch themes*, (2021), <https://catchthemes.com/themes/catch-box/>
- [5] Duo M., (2021), *How to change a Wordpress theme (Without breaking you website)*, *Kinsta*, <https://kinsta.com/blog/change-wordpress-theme/>
- [6] *Get wordpress*, (2021), <https://wordpress.org/download/>
- [7] *Inspect element: How to change any webpage on any browser - tutorial*, *Devmountain*, (2021), <https://blog.devmountain.com/how-to-use-inspect-element-jump-into-what-makes-a-web-page-tick/>
- [8] *Responsive design mode*, (2021), *MDN web docs Mozilla*, https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_Mode
- [9] *What are Wordpress themes (And where to find them)*, (2021), *DreamHost*, <https://www.dreamhost.com/wordpress/how-to-find-wp-themes/>
- [10] Mc Collin R., (2021), *How to create a child theme in WordPress (extended guide)*, <https://kinsta.com/blog/wordpress-child-theme/>
- [11] Duo M., (2021), *How to edit CSS in WordPress (Edit, Add and Customize How your Site Looks)*, *Kinsta*, <https://kinsta.com/blog/wordpress-css/>
- [12] *How to Modify Your WordPress CSS Styles*, (2020), *Visual Composer*, <https://visualcomposer.com/blog/wordpress-css-styles/>
- [13] *How to Easily Add Custom CSS to your WordPress Site*, (2021), <https://www.wpbeginner.com/plugins/how-to-easily-add-custom-css-to-your-wordpress-site/>
- [14] Laxmi A., Perna M. A., (2020), *CSS Viewport Units: A Quick Start*, *Sitepoint*, <https://www.sitepoint.com/css-viewport-units-quick-start/>
- [15] Michael, N., (2021), *CSS Properties – Max-width, Min-width, Max-height, and Min-height Explained with Examples*, <https://www.freecodecamp.org/news/css-properties-examples/>
- [16] Nimritee, (2021) *How to use CSS breakpoints for responsive design*, *Lambda test*, <https://www.lambdatest.com/blog/how-to-use-css-breakpoints-for-responsive-design/>
- [17] Ranjit P., (2018), *CSS – margin:auto; - How it works*, *Hongkiat*, <https://www.hongkiat.com/blog/css-margin-auto/>
- [18] *CSS layout – Horizontal @ vertical align*, (2021), *w3schools*, https://www.w3schools.com/css/css_align.asp
- [19] Sandu A., (2019), *Rem in CSS: Understanding and using rem units*, *Sitepoint*, <https://www.sitepoint.com/understanding-and-using-rem-units-in-css/>