

УНИВЕРЗИТЕТ „СВ. КИРИЛ И МЕТОДИЈ“ – СКОПЈЕ

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И
КОМПЈУТЕРСКО ИНЖЕНЕРСТВО – СКОПЈЕ

НАСОКА: СОФТВЕРСКО ИНЖИНЕРСТВО И
ИНФОРМАЦИСКИ СИСТЕМИ



СЕМИНАРСКА РАБОТА

ПРЕДМЕТ:

ОСНОВИ НА ВЕБ ДИЗАЈН

ТЕМА:

Дизајнирање на веб страна за онлајн продавница со користење SCSS

ментор:

м-р Бобан Јоксимоски

кандидат:

Вангел Трајковски 181261

Јуни 2019

Соджина

Апстракт	4
1. Вовед	5
2. Објаснување на веб страната	5
2.1 Составни делови на веб страната	5
2.2 Употребени функционалности и нивно објаснување	6
3. Изработка на веб страната	9
3.1 Поставување на структурата во HTML	9
3.2 Позиционирање на елементите во веб страната	12
3.3 Дизајн на елементите од веб страната	14
4. Заклучок	21
5. Користена литература	22

Листа на Слики

Слика 1 Темна тема	5
Слика 2 Светла тема.....	5
Слика 3 Променливи за светла тема	6
Слика 4 Променливи за темната тема.....	6
Слика 5 Структура на страничното мени од главната страна.....	7
Слика 6 Дизајн на страничното мени од главната страна.....	7
Слика 7 Mixin за поставување на преграда и радиус до таа преграда	7
Слика 8 Функција за пресметување ширина или висина.....	8
Слика 9 Grid layout на главната страна.....	9
Слика 10 Главно мени.....	9
Слика 11 Линкови до продуктите.....	10
Слика 12 Страната Shop со светла тема.....	10
Слика 13 Страната Shop со темна тема	10
Слика 14 Структура на елементите во Contact страната	11
Слика 15 Структура на елементите од страната за најава	11
Слика 16 Страната на продуктот со светла тема	12
Слика 17 Страна на продуктот со темна тема	12
Слика 18 Поставување на елементот во мрежата.....	12
Слика 19 Код за поставување на grid layout.....	12
Слика 20 Комплетен grid layout на веб страната	13
Слика 21 Grid layout на главното мени	14
Слика 22 Особини на копче за пребарување	14
Слика 23 Особините на главното мени.....	14
Слика 24 Особини на полето за пребарување	15
Слика 25 Особини на линковите на главното мени	15
Слика 26 Код за менување на тема	16
Слика 27 Позиционирање на елементите во мрежата	16
Слика 28 Особини на линковите до продуктите.....	16
Слика 29 Главното мени со намалена големина на екран со светла тема	17
Слика 30 Главното мени со намалена големина на екран со темна тема.....	17
Слика 31 Продукт во нормална состојба	17
Слика 32 Продукт кога врз него се наоѓа стрелката од маусот	17
Слика 33 Поставување слики на линковите	18
Слика 34 Структура на панелот за слики	18
Слика 35 Скрипта за промена на слики во панелот.....	19
Слика 36 Особини на панелот за слики.....	19
Слика 37 Особините на сите копчиња на веб страната	20

Апстракт

Веб страната беше направена за дополнително истражување во полето за дизајн со дополнителни алатки за кои бев заинтересиран и инспириран од карактеристиките на веб дизајнот и сакав подлабоко да истражувам во таа област, проектот е изработен во SCSS(Sassy Cascading Style Sheets), со кој се изработува дизајн на веб страните, употребени се повеќето функционалности на SCSS за дизајнирање и решавање на проблемите со кој се соочуваат дизајнерите, изработена беше како една општа веб страна, со наједноставно главно мени и чист дизајн.

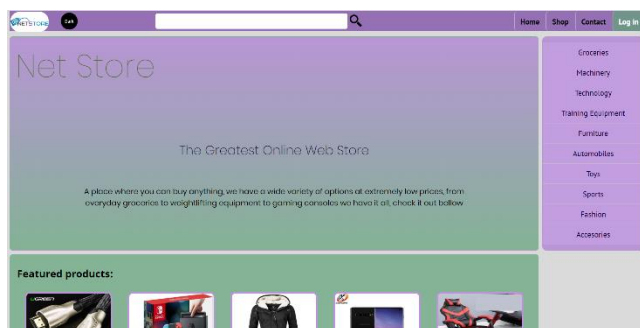
1. Вовед

Веб дизајнот се состои од две најосновни компоненти HTML(Hyper text markup language) кој служи за поставување на основните елементи и структурата на веб страната и CSS (Cascading Style Sheets) со кој се означуваат елементите од HTML и се менува нивниот изглед. SCSS (Sassy Cascading Style Sheets) претставува варијанта на CSS со дополнителни функционалности кои овозможува повеќе видови на прилагодувања. Мојата цел беше да изработам веб страна со користење на дополнителните функционалности кои ги овозможува SCSS кој се користи за изработка на дизајнот на веб страната, целта беше да се изработи веб страна во која ќе може корисникот да менува тема на веб страната, и да се направи дизајн кој ќе биде модерен и практичен за користење со употреба на варијанта на CSS наречена SCSS. За да се стигне до главната цел потребно беше искористување на флексибилноста што ја нуди SCSS.

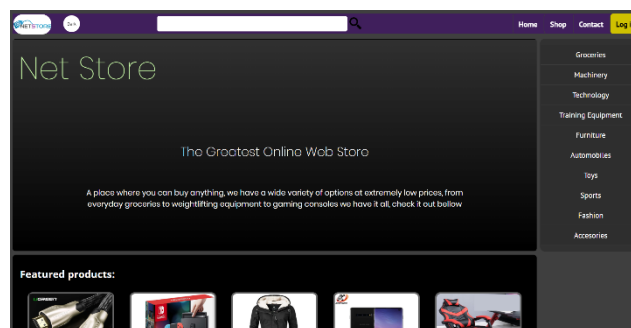
2. Објаснување на веб страната

2.1 Составни делови на веб страната

Веб страната се состои од 5 страници со различен дизајн, една од главните особини на веб страната е тоа што е направена со “grid layout” – што значи дека веб страната е направена во вид на табела односно во вид на мрежа 6x3 како што е прикажано на слика 3. Страната исто така се состои од 2 „теми“, што значи дека веб страната е дизајнирана на два различни начини од кои единствена разлика е бојата. Во првиот дизајн се искористени светли бои (слика 1), а во вториот се искористени потемни бои и подоминантна е црната (слика 2). Имаме 3 главни чекори за правење на дизајнот на веб страната. А тоа се: (1) поставување на структурата во HTML, (2) сместување на елементите на посакуваните позиции и (3) додавање дополнителен дизајн на елементите за секој елемент посебно. Постојат 6 главни видови на дизајн на веб страната, другите се повторување на овие 6.



Слика 2 Светла тема



Слика 1 Темна тема

2.2 Употребени функционалности и нивно објаснување

Во SCSS постојат повеќе алатки кои многу го олеснуваат дизајнирањето на веб страни, тоа се:

1. Променливи

- Променливи со кои може да се зачува некоја вредност од документот како што е висина или боја на одреден елемент и може да се користи повеќе пати, и ако сакаме да ја смениме таа боја со друга боја, само ја менуваме променливата и се менува во целата веб страна, но во случај на обичен CSS тоа не е возможно и треба да се менува во рамките на цел документ. Променливите се пишуваат со знакот \$ пред нив, како што е покажано на слика 3 и слика 4, тоа се истите променливи но се користат во 2 различни случаи.

```
$bg-color: #9670b4; // #71538a
$bg-color1: #b996d6;
$bg-color2: rgba(98, 163, 123, 0.72);
$bg-color3: rgba(200, 200, 200, 0.67);
$Text-color: #000000;
$text-color1: #4f6d2e;
$text-color2: #140041;
$nav-Height: 60px;
$link-hover-color: #d1d696;
$dark: #000000;
$light: #FFFFFF;
$Price-text-color: #e42c2a;
$bg-color-checkbox: #3b3b3b;
$submit-button-color: #3e3b3c;
$sign-up-color: rgba(98, 163, 123, 0.72);
```

Слика 3 Променливи за светла тема

```
$bg-color: #401f5c; // #71538a
$bg-color1: rgba(43, 43, 43, 0.72);
$bg-color2: #000000;
$bg-color3: #3f3f3f; // 7b7b7b
$Text-color: #FFFFFF;
$text-color1: #ccfeab;
$text-color2: #FFFFFF;
$nav-Height: 60px;
$link-hover-color: #df2130; // 8c8086
$dark: #FFFFFF;
$light: #000000;
$Price-text-color: #e42c2a;
$bg-color-checkbox: #6f6f6f;
$submit-button-color: #a2000a;
$sign-up-color: #cfc100;
```

Слика 4 Променливи за темната тема

2. Вгнездување

- Вгнездувањето ни овозможува да го пишуваме кодот според неговата структура напишана во HTML, ова е корисно затоа што избегнуваме користење на покомплицирани селектори и станува полесен начинот на дизајнирање. Според слика 5 и слика 6 можеме да видиме пример за вгнездување, во слика 5 елементите се поставени во некаква структура, а тоа го користиме во SCSS, и според таа структура ги поставивме елементите како што се поставени во HTML. Можеме да видиме на слика 5 дека елементот „a“ е поставен во елемент „li“ кој се наоѓа во елемент „ul“, кој се наоѓа во елементот „aside“, според слика 6 може да се види како се поставени елементите.

```

<aside class="sidemenu">
  <ul>
    <li>
      <a href="Shop.html">Groceries</a>
    </li>
    <li>
      <a href="Shop.html">Machinery</a>
    </li>
    <li>
      <a href="Shop.html">Technology</a>
    </li>
    <li>
      <a href="Shop.html">Training Equipment</a>
    </li>
    <li>
      <a href="Shop.html">Furniture</a>
    </li>
    <li>
      <a href="Shop.html">Automobiles</a>
    </li>
    <li>
      <a href="Shop.html">Toys</a>
    </li>
    <li>
      <a href="Shop.html">Sports</a>
    </li>
    <li>
      <a href="Shop.html">Fashion</a>
    </li>
    <li>
      <a href="Shop.html">Accessories</a>
    </li>
  </ul>
</aside>

```

Слика 5 Структура на страничното мени од главната страна

```

.sidemenu {
  display: inline-block;
  float: right;
  text-align: center;
  @include border-radius(15px);
  background: $bg-color1;
  ul {
    display: inline-block;
    list-style-type: none;
    width: 100%;
    padding: 0;
    li {
      display: block;
      width: 100%;
      a {
        display: block;
        width: 100%;
        border-bottom: 1px solid $bg-color3;
        height: 100%;
        box-sizing: border-box;
        padding: 15px;
        text-decoration: none;
        color: $text-color2;
        background: $bg-color1 * 1.05;
        font-family: "ET Sans", arial, sans-serif;
        font-size: 6mm;
        grid-area: elem;
        &:hover {
          background: $bg-color1 * 0.8;
          color: $link-hover-color;
        }
      }
    }
  }
}

```

Слика 6 Дизајн на страничното мени од главната страна

3. Mixins

- Претставуваат делчиња со кои се пишува одреден дизајн, и тој дизајн може да се искористи на повеќе слични елементи без да го повторуваме истиот код, исто така може да направиме пофлексибилен дизајн со користење на променливи, како што е должина и висина на елемент, одредена боја како позадина, и многу други случаи. Се креираат со зборчето
 @mixin lme-Na-Mixin(\$promenliva)
 а се користат со
 @include lme-Na-Mixin(\$promenliva),

```

@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
  -moz-border-radius: $radius;
  border-radius: $radius;
  border: transparent 2px solid;
}

```

Слика 7 Mixin за поставување на презграда и радиус до таа презграда

како што е прикажано на слика 7, за едно исто својство се користат повеќе различни начини на промена, во зависност на кој веб прелистувач се користи, па за тоа би требало да го пишуваме истиот код повеќе пати, но со mixin, го пишуваме кодот еднаш, а потоа само го искористуваме каде што ни е потребен.

4. Наследување

- Наследувањето работи слично како mixins, но различно е во тоа што не се пишува посебен код, се искористува веќе напишан код за еден елемент

и се користи дизајнот за друг елемент, и повторно со промена на дизајнот на првиот елемент се менува и се што го наследува дизајнот од тој елемент. Се користат со зборчето „@extend”.

5. Оператори

- Операторите се користат за пресметки на одредени вредности со голема прецизност, и можеме со математички равенки попрецизно да позиционираме елемент, исто така може да се смени боја од потемна во посветла и обратно.

6. Функции

- Функциите се исто така слични со mixins, но за нив карактеристично е тоа што се додаваат променливи и се вршат пресметки, а на крајот ја враќаат одредената вредност која можеме да ја користиме и исто така е невозможно да се менува дизајн на елементите како што може во mixins. Во SCSS имаме и готови функции како што се Darken(\$boja) и

```
}  
@function Divide($n) {  
  $r: (100% / $n);  
  $r: calc(#{$r} - 2px);  
  @return $r;  
}
```

Слика 8 Функција за пресметување ширина или висина

Lighten(\$boja), кои служат за затемнување или осветлување на одредената боја зададена како параметар и многу други. За разлика од mixins функциите се користат со нивното име, и се користат како променливи на одредени особини на елементите како што се ширина, висина итн. Според слика 8 прикажано е делење на елементи и

користење на оператори, во функцијата се пресметува делењето на просторот на еднакви делови и се одзема мала вредност од елементите за тие да се правилно поставени, тоа се прави со користење на друга функција дадена од SCSS наречена „calc”.

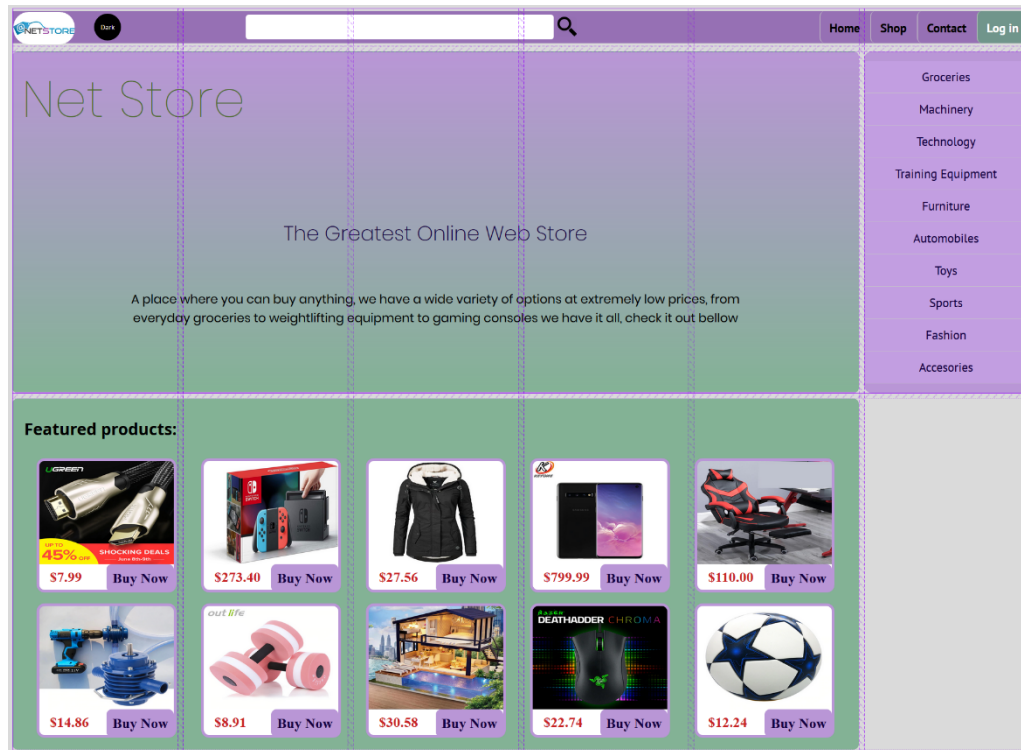
7. Циклуси

- Циклусите служат за повторување на дизајн со мали промени, така што кога ќе ставиме ширина на еден елемент, на следниот елемент можеме да ја намалиме ширината за 1 помала или поголема од предходниот итн.

3. Изработка на веб страната

3.1 Поставување на структурата во HTML

Структурата во главната страна е поделена на 4 делови, главно мени, односно header со структура претставена на слика 10, кое ги има на сите



Слика 9 Grid layout на главната страна

страници од веб страната, главен наслов сместен во елемент section, странично мени сместено во елементот aside, и делот за продуктите е сместен во уште еден section, во header имаме дополнително 4 елемента, слика, копче за менување на тема, поле за пребарување, и неопределена листа од линкови дизајнирани како копчиња прикажано на слика 11, потоа во елементот section каде што се сместени продуктите за секој продукт посебно има еден елемент поставено од типот „а“ како на слика 11.



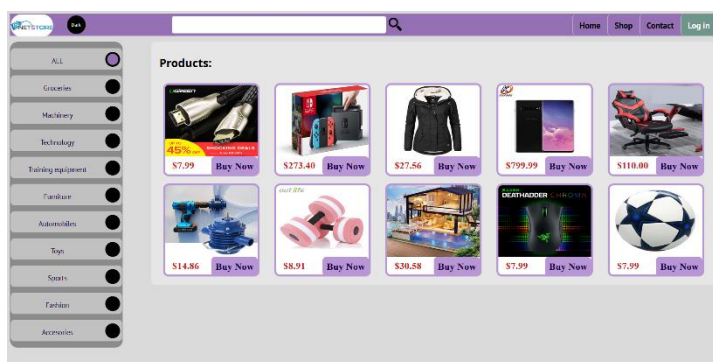
```

<a href="Products/Product4.html">
  <article>
    <h3>New Samsung Galaxy S10 G9730 Mobile Phone 6.1" 8GB RAM 128GB ROM Snapdragon 855 IP68 Waterproof Dustproof Android 9.0 Dual SIM</h3>
  </article>
  <h4>$799.99</h4><h5>Buy Now</h5>
</a>
<a href="Products/Product5.html">
  <article>
    <h3>Gaming Chair Electrified Internet Cafe Pink Armchair High Back Computer Office Furniture Executive Desk Chairs Recliner</h3>
  </article>
  <h4>$110.00</h4><h5>Buy Now</h5>
</a>

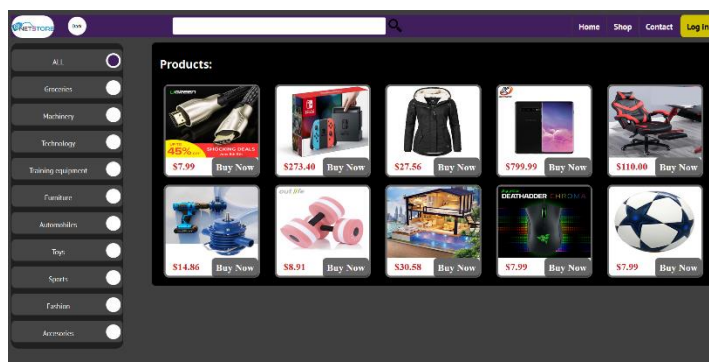
```

Слика 11 Линкови до продуктите

Потоа следува страницата каде се поставени продуктите со подобар преглед, овде на ист начин е поставено мени од страна во елемент aside, и потоа повеќе продукти исто како и во главната страна како што можеме да видиме на слика 12 со светла тема или на слика 13 со темна тема.



Слика 12 Страницата Shop со светла тема



Слика 13 Страницата Shop со темна тема

Страната за контакт се состои од еден главен елемент од типот „div“ и ги содржи сите други елементи во него, се состои од главен наслов, форма во која има полиња за внесување на името, емаилот, видот на проблем и објаснување на проблемот, од страна на корисникот, потоа следуваат дополнителните информации за контакт прикажано на слика 14.

Следна страница е страницата за најава, слично како и кај контакт страницата, се состои од еден главен елемент од типот „div“ претставено со слика 15 во кој има форма со полиња за впишување на корисничко име или е-маил и лозинка, потоа има и копче за најава кое треба да ги испраќа податоците за проверка. И уште едно копче што служи за регистрирање на нов корисник и уште еден елемент кој што служи за насловот на страницата.

```

<div class="main-contact">
  <div class="contact">CONTACT</div>
  <div class="short-text">Tell us your problem</div>
  <div class="Container">
    <div>
      <form>
        <label>
          <input class="name" type="text" placeholder="Name:">
        </label>
        <label>
          <input class="email" type="email" placeholder="E-mail:">
        </label>
        <label>
          <select id="select-problem">
            <option value="none">--Type of problem--</option>
            <option value="account">Account problems</option>
            <option value="seller">Problems with seller</option>
            <option value="product">Product</option>
          </select>
        </label>
        <label>
          <textarea class="description" placeholder="Describe your problem... "></textarea>
        </label>
        <button type="submit">Submit</button>
      </form>
    </div>
    <div class="contact-info">
      <div><a href="https://goo.gl/maps/Ty96Vct5k4T8PUw4A">4th of November, Bitola 7000</a></div>
      <div><span>071 298 360</span></div>
      <div><span>Vangel.trajkovski@finki.ukim.mk</span></div>
      <div>
    </div>
  </div>
</div>

```

Слика 14 Структура на елементите во Contact страницата

```

<div class="login-text">
  <span>Please login or sign up here</span>
</div>
<div class="login-form">
  <form>
    <fieldset>
      <legend>E-mail or Username</legend>
      <label>
        <input type="text">
      </label>
    </fieldset>
    <fieldset>
      <legend>Password</legend>
      <label>
        <input type="password">
      </label>
    </fieldset>
    <button type="submit">Log in</button><button>Sign up</button>
  </form>
</div>

```

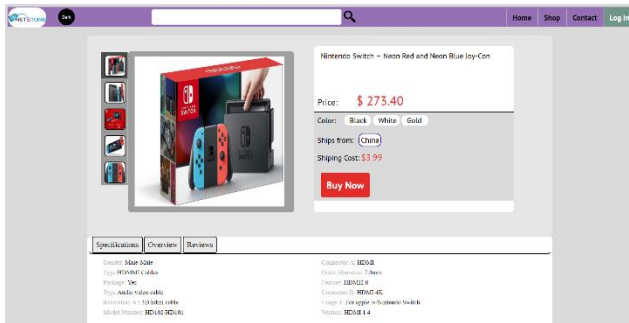
Слика 15 Структура на елементите од страницата за најава

прикажано на слика 16 и слика 17.

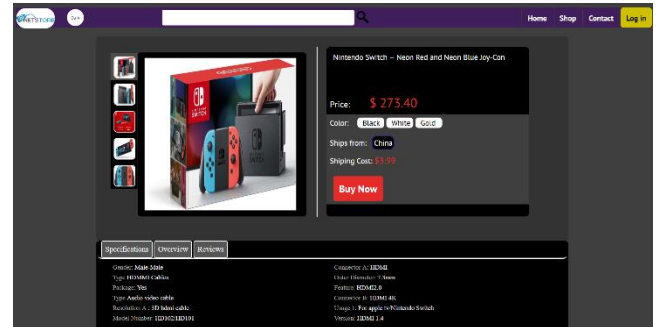
Последната страна што има свој дизајн е страницата за продуктот, таа се состои од три главни компоненти, тоа се панел на кој се поставени сликите од продуктот, елемент каде што се поставени основните информации на продуктот како што се цената, бојата и копче за купување, и последниот елемент е за повеќе информации за продуктот. Панелот за слики се состои од неопределена листа од слики, вертикално поставени од лево

3.2 Позиционирање на елементите во веб страната

Позиционирањето е направено во вид на табела, на начин наречен grid layout, со што се поставува мрежа на целата страна и секој елемент се поставува



Слика 16 Страната на продуктот со светла тема



Слика 17 Страна на продуктот со темна тема

```
header{
  grid-row-start: 1;
  grid-row-end: 2;
  grid-column-start: 1;
  grid-column-end: 7;
  margin: auto 0;
}
```

Слика 18 Поставување на елементот во мрежата

во мрежата. Мрежата на главната страна е 6x3 како што е покажано на слика 9. Тоа е направено со употребување на својството прикажано на слика 18. Со менување на својството display: grid; се поставува мрежата на страната, потоа со менување на својствата што следуваат се наведуваат колку колони и редици ќе има во

мрежата, за тоа служат својствата grid-template-rows и grid-template-columns, во зависност од колкав број на променливи се поставени толку ќе биде голема мрежата, искористена беше и функција што ја нуди SCSS наречена repeat(), оваа функција го повторува вториот аргумент, за онолку пати колку што бројот првиот аргумент. Последното својство grid-column-gap служи за оставање место помеѓу колоните.

Елементите се поставуваат во мрежата така што се специфицира во кој ред започнува и кој ред завршува елементот и која колона започнува и завршува, тоа се прави со својствата наведени на

```
@mixin grid-layout{
  display: grid;
  grid-template-rows: 70px auto;
  grid-template-columns: repeat(5, 1fr) auto;
  grid-row-gap: 10px;
  grid-column-gap: 10px;
}
```

Слика 19 Код за поставување на grid layout

```

header{
  grid-row-start: 1;
  grid-row-end: 2;
  grid-column-start: 1;
  grid-column-end: 7;
  margin: auto 0;
}

.Main{
  height: auto;
  grid-row-start: 2;
  grid-column-start: 1;
  grid-column-end: 6;
}

.Featured{
  grid-column-start: 1;
  grid-column-end: 6;
}

.sidemenu{
  margin:auto;
}

.products{
  grid-column-start: 2;
  grid-column-end: 7;
}

.main-contact{
  grid-column-start: 3;
  grid-column-end: 4;
}

.selectmenu{
  grid-row-start: 2;
  grid-row-end: 2;
  grid-column-start: 1;
  grid-column-end: 1;
}

.login-form{
  grid-row-start: 3;
  grid-column-start: 3;
  grid-column-end: 4;
}

.login-text{
  grid-column-start: 3;
  grid-column-end: 4;
}

.main{
  grid-column-start: 3;
  grid-column-end: 3;
}

.additional-info{
  grid-row-start: 3;
  grid-row-end: 4;
  grid-column-start: 3;
  grid-column-end: 3;
}
}

```

Слика 20 Комплетен grid layout на веб страната

слика 19, со својството grid-row-start се специфицира од кој ред почнува, а со grid-row-end се специфицира до кој ред се шири тој елемент, со својствата grid-column-start и grid-column-end се специфицираат од која колона започнуваат и до која колона се шират. На слика 13 е дадено позиционирањето на главното мени кое се наоѓа на првата редица, и се шири од почеток до крај.

Еден grid layout функционира на целата веб страна, така што главното мени е заедничко за сите и секогаш ќе биде во иста позиција, потоа имавме елементи во страната кои се карактеристични по страна и може да се постават независно од другите елементи што ги нема на дадената страна, на слика 14, се приметувa дека за елементот .sidemenu нема ставено почетна колона или редица и крајна колона или редица, тоа е затоа што елементот без тие особини беше поставен па правото место, исто така се забележува дека немора да се специфицираат сите својства, може само да се специфицираат само некои од тие својства но немора, се додека не се позиционира елементот. Во мрежата се позиционираат само елементите што се наоѓаат директно внатре во елементот со display: grid; својството односно тие се негови директни деца.

Grid layout беше искористено и кај главното мени, во димензии 2x16 како што е покажано на слика 21, односно има 2 колони од кои првата се наоѓа се што припаѓа на менито, како што се логото, копчето за менување на темата, полето за пребарување и неподредената листа од линкови, а втората редица поставена е за случаи кога екранот е намален да се преместат линковите во втората колона, за подобра прегледност, тоа е направено со користење на JavaScript, кој претставува програмски јазик наменет за програмирање на повеќето функционалности на веб страни, искористен е кодот од слика 20.



Слика 21 Grid layout на главното мени

3.3 Дизајн на елементите од веб страната

Веб страната има 8 главни делови на дизајн, првиот составен дел на веб страната е главното мени, тоа го има на сите страници, дизајнирано беше така што покрај grid layout има поставено позадинска боја, висина на менито, посебен фонт за линковите, и големина на буквите, прикажано на слика 22, бојата се менува во зависност од променливите на слика 3 и слика 4, потоа следат елементите што

```
nav{
  background: $bg-color;
  height: $nav-Height ;
  font-family: 'PT Sans',Arial,sans-serif;
  font-weight: 700;
  font-size: 6mm;
```

Слика 23 Особини на копче за пребарување

припаѓаат на главното мени, прв елемент е логото тоа е направено така што е специфицирана неговата висина и дополнително додадено заоблување на неговите рабови. Следен елемент е копчето за пребарување тоа е направено така што е ставена позадина за да го добие тој изглед, поставена висина и центрирано со помош на маргина, тоа претставува простор помеѓу соседните елементи, поставен е со апсолутна позиција што значи точно е специфицирана позицијата во кој дел од страната се наоѓа, но тоа својство беше искористено само за да не си пречи со соседните елементи, исто така и тука имаме заоблување на рабовите, и специфицирано колкав простор зафаќа позадината. Ова е прикажано на слика 23, дополнително на настан кога

```
.search-button{
  display: inline-block;
  height: $nav-Height - 10;
  position: absolute;
  margin-top: 5px;
  margin-left: 5px;
  width: 40px;
  top: -3px;
  @include border-radius(15px);
  background: url("../Images/search.png") no-repeat;
  background-size: 100% ;
  &:hover{
    background: url("../Images/hover.png") no-repeat;
    background-size: 100% ;
  }
```

Слика 22 Особините на главното мени

стрелката од глумчето се однесе над елементот се менува изгледот, тоа се прави со селектор наречен „hover“. Веднаш до копчето се наоѓаат, полето за пребарување за кое е специфицирано само позицијата релативно и центрирано со поставување на маргината да биде автоматска, дополнително имаше дадено ширина, и поместување во лево за подобро позиционирање на елементот, ова е прикажано на слика 24.

Последен елемент од главното мени е неподредената листа од линкови, поставено е во десно, со тоа што е специфицирано својството `position: absolute;` и поставено со својство `right: 0;` што значи дека елементот ќе се наоѓа најдесно што може. Поради тоа што елементот се наоѓа во мрежата односно `grid layout` кајниот раб до кој може да се шири е работ на мрежата. За листата исто така е специфицирано да не бидат означени елементите од листата. Во листата се наоѓаат елементи од типот „li“ што означува `list item`, односно елемент од листата, обично овие елементи се поставени вертикално, за тоа да се смени искористено е својство `inline-block`, со што елементите нема да зафаќаат цел ред и ако е возможно ќе се шират во еден ред. Исто така специфицирани се висината, ширината, која е поставена со равенка и го дели просторот на 4 делови, исто така рабовите се заоблени на секој елемент поединечно, и е додадено раб на левата страна со различна боја за подобар изглед. Веднаш во елементите од листата се наоѓаат линковите, за нив е специфицирано ширина и висина колку што се широки елементите од листата, големина на внатрешниот дел на линкот за да изгледа како копче, и боја на текстот и позадина кога се донесе стрелката над елементот, прикажано на слика 25 со својството `hover`, исто така имаме за последниот линк посебифичен дизајн направено со селектор за последно дете односно `:last-child`, со ова се менува само крајниот елемент и тој ќе има поразличен дизајн. Од главното мени останува уште копчето за менување на тема.

Една од задачите беше да се направи веб страна така што ќе може да се менува тема на страната. Тоа е постигнато со користење на JavaScript, тоа е направено така што со притискање на даденото копче се додава „класа“ на телото на веб страната наречено `Body`, со додавање на оваа „класа“, се менува дизајнот на веб страната, кога повторно се притисне копчето се трга дополнително додадената класа од `Body` елементот и ќе биде вратен стариот дизајн. Потоа имаме случај кога ја исклучиме или ја ресетираме веб страната дизајнот што го гледаме моментално се брише и секогаш се враќа на почетниот

```
.formsearch{
  display: inline-block;
  position: relative;
  margin: auto;
  width: 100%;
  left: 80px;
}
.search{
  @include border-radius(5px);
  display: inline-block;
  height: $nav-Height - 20;
  width: 80%;
  font-size: 6mm;
}
```

Слика 24 Особини на полето за пребарување

```
ul{
  position: absolute;
  right: 0;
  list-style-type: none;
  margin: 0;
  padding: 0;
  float: right;
}
li{
  display: inline-block;
  height: $nav-Height;
  width: 100% / 4;
  word-spacing: 0;
  margin-left: -4px;
  @include border-radius(10px);
  border: 0.1px;
  border-left: $bg-color3 solid 2px;
}
a{
  @include border-radius(10%);
  display: inline-block;
  box-sizing: border-box;
  padding: 15px;
  width: 100%;
  text-decoration: none;
  height: $nav-Height;
  color: $Text-color;
  &:hover{
    background: $bg-color * 0.5;
    color: $link-hover-color;
  }
}
&:last-child{
  background: $sign-up-color;
}
a{
  color: $light;
  &:hover{
    background: darken($sign-up-color, 10);
    color: $link-hover-color;
  }
}
}
```

Слика 25 Особини на линковите на главното мени

дизајн, тоа е решено така што се запаметува дали е додадена таа дополнителна класа или не, и е искористено тоа, така што се проверува дали била додадена таа дополнителна класа што го менува дизајнот и ако веќе била додадена но избришана се додава повторно, ова е прикажано на слика 26.

```
document.getElementById('theme-button').addEventListener('click', function () {
  let darkThemeEnabled = document.body.classList.toggle('dark-theme');
  localStorage.setItem('dark-theme-enabled', darkThemeEnabled);
});
if (JSON.parse(localStorage.getItem('dark-theme-enabled'))) {
  document.body.classList.add('dark-theme');
}
```

Слика 26 Код за менување на тема

На слика 27 е поставен grid layout од главното мени.

Исто така имавме додадено дополнителна функционалност на главното мени, така што кога прозорецот се намалува, главното мени се собира позади едно копче наречено хамбургер мени, по притискање на ова копче се појавува менито во вертикална положба прикажано на слика 29 и 30.

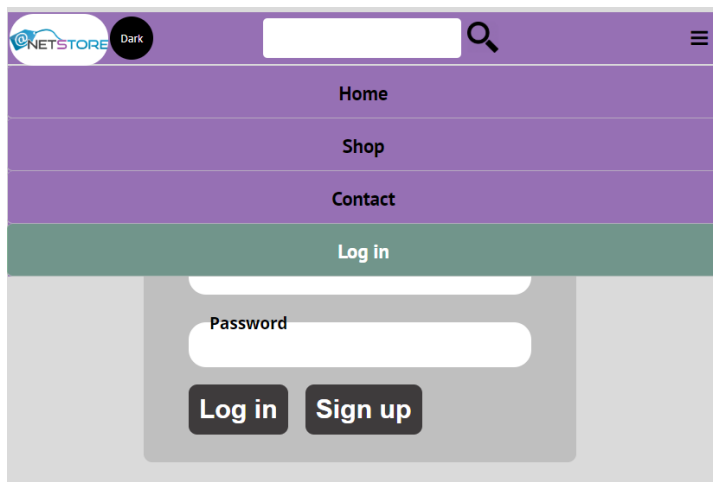
Главната страна се уште од состои од главно мени, Потоа има и странично мени, за одбирање на категорија на производи за кои сме заинтересирани, дизајнирано е слично како и неподредената листа од линкови во главното мени со таа разлика што линковите се поставени вертикално. Во средината се наоѓа голем елемент на кој го пишува името на компанијата и беше направено само со позиционирање на текстот и поставување на фон, боја итн. Исто така на него има и некои минимални информации, и веднаш под него е делот со најпопуларните продукти, изгледот на линковите до продуктите е претставен со слика 28. Ова е вториот составен дел од дизајнот на веб страната кои се

```
@mixin grid-menu{
  nav{
    display: grid;
    grid-template-columns: repeat(16,1fr);
    grid-template-rows: 60px auto;
    grid-row-gap: 20px;
    .logo{
      grid-column-start: 1;
      grid-column-end: 1;
    }
    .formsearch{
      grid-column-start: 4;
      grid-column-end: 10;
    }
    ul{
      grid-column-start: 13;
      grid-column-end: 17;
    }
  }
  #theme-button{
    grid-column-start: 2;
    grid-column-end: 3;
  }
}
```

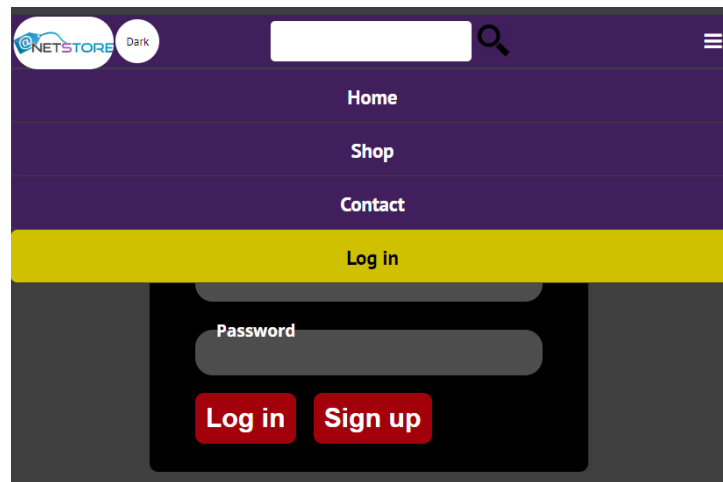
Слика 27 Позиционирање на елементите во мрежата

```
a{
  display: inline-block;
  width: 250px;
  justify-content: space-around;
  align-items: center;
  color: #c82224;
  height: $nav-height*4;
  @include border-radius(15px);
  border: $bg-color solid 5px;
```

Слика 28 Особини на линковите до продуктите

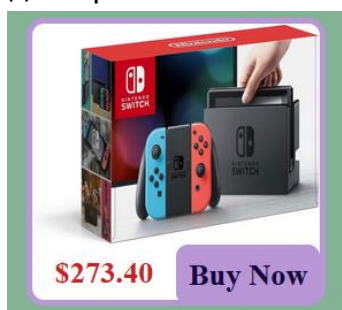


Слика 29 Главното мени со намалена големина на екран со светла тема

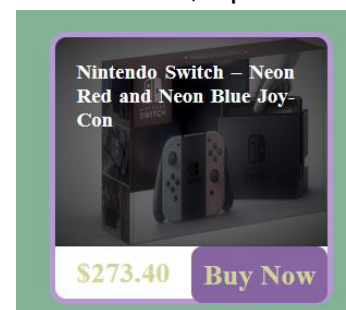


Слика 30 Главното мени со намалена големина на екран со темна тема

продаваат на веб страната тие исто така содржат grid layout за позиционирање под еднакво 5 продукти во една редица, во најпрвата редица се наоѓа насловот на тој сегмент од веб страната, потоа за секој продукт е дадена информација за неговата цена, како изгледа, дизајнот на линковите е направен така што направени се да се прикажуваат како квадрати и да се шират во една линија со својството inline-block, и дополнително елементите се центрирани во линкот, дополнително има направено и дизајн за текстовите и другите информации што се наоѓаат во линкот, на нив само им се става боја, ширина, должина, и позиционирање, за крај кога со стрелката од глумчето се помине над одреден продукт се затемнува сликата како на и преминува од слика 31 во слика 32 и се прикажува името на продуктот, но пред тоа треба да се напомени како се ставени сликите за продуктот. Тие се ставени на начин што имаме циклус кој ги брои сите линкови и на секој линк става различна слика како што е прикажано на слика 33, прво имаме една променлива што ќе ги брои елементите, и таа започнува од вториот елемент тоа е затоа што, првиот елемент во полето за линковите е насловот на полето, потоа доделиваме променлива наречена линк, тука се почнува од првиот продукт и се оди се додека не се стигне до последниот, и се става сликата со минимални измени во линкот, исто така имаме и дополнителна слика за моментот кога ја однесеме стрелката врз елементот.



Слика 31 Продукт во нормална состојба



Слика 32 Продукт кога врз него се наоѓа стрелката од маусот

Дополнителни информации за продуктот може да се добијат со притискање на продуктот, така што се отвара страната на притиснатиот продукт, прикажана на слика 16 и слика 17. Горниот дел од елементите се поставени во grid layout или мрежа за поубаво позиционирање, а дизајнот е направен така што за делот со слики е искористена неподредена листа, и како елементи

во листата се ставени по 2 слики, една да се прикажува како мала а другата се прикажува на поголемиот дел. Структурата е претставена на слика 34, се прикажува така што повторно е искористен JavaScript, но овој пат се става класа на еден елемент од низата на слики со тоа што кога стрелката од глумчето се донесе над некоја слика од низата, тогаш се прикажува втората слика која што во нормална положба е скриена од корисникот и се прикажува се додека не се селектира друга слика. Пишувањето на таков код е доста едноставно, на слика 35 е претставен кодот што ни го овозможува тоа, така што прво декларираме промелнива „i“ која ни служи за броење на сите елементи што се наоѓаат во листата, односно сликите, потоа на првата слика ја поставуваме класата „keep“ што служи за да се појави

```
@for $i from 2 through 11 {
  $link: "../Images/Product" + ($i - 1);
  a:nth-child(#{ $i }) {
    @include backgroundSet($link + "/Image.jpg");
    &:hover {
      @include backgroundSet($link + "/ImageH.jpg");
    }
  }
}
```

Слика 33 Поставување слики на линковите

```
<div class="main">
  <div class="image-panel">
    <ul id="image-list">
      <li id="pic1">
        
        
      </li>
      <li id="pic2">
        
        
      </li>
      <li id="pic3">
        
        
      </li>
      <li id="pic4">
        
        
      </li>
      <li id="pic5">
        
        
      </li>
    </ul>
    <script src="../scripts/Image_Display.js"></script>
  </div>
</div>
```

Слика 34 Структура на панелот за слики

големата слика на панелот и се запаметува на кој елемент е поставена оваа класа, откако селектираме друга слика, прво се гледа која слика е, со тоа што се бројат елементите, и се поништува класата од предходната запаметена слика, потоа се запаметува следната слика и се додава класата „keep“, и ова цело време се повторува.

```

let i;
let Imagelist = document.getElementById( elementId: "image-list");
let t = document.getElementById( elementId: "pic1");
t.classList.toggle( token: "keep");
for(i = 1; i < Imagelist.childElementCount + 1; i++){
    let p = "pic" + i;
    document.getElementById( p.toString()).addEventListener( type: "mouseenter", listener: function () {
        t.classList.toggle( token: "keep");
        t = document.getElementById(p.toString());
        t.classList.toggle( token: "keep");
    });
}

```

Слика 35 Скрипта за промена на слики во панелот

Дизајнот на панелот за слики е доста едноставен, претставен е со фиксна висина и ширина, центриран е со маргините, се состои само од една неподредена листа,

```

.image-panel{
    display: inline-block;
    width: 500px;
    height: 500px;
    position: relative;
    margin-left: auto;
    ul{
        display: inline-block;
        list-style-type: none;
        padding: 0;
        height: 480px;
        width: 496px;
        background: darken($bg-color3, 30);
        @include border-radius(10px);
    }
    li{
        width: $size;
        height: $size;
        padding: 10px;
        background: darken($light, 40);
        position: relative;
        left: -$size - 20px - 3px;
        img:first-child{
            width: $size;
            height: $size;
            @include border-radius(10px);
            border: 2px solid $dark;
        }
        .keep{
            .bigpic{
                display: inline-block;
                position: absolute;
                left: $size + 20px + 12px + 9px;
                height: 450px;
                width: 450px;
                margin-top: 14px;
            }
            @for $i from 1 through 10{
                :nth-child(#{ $i }){
                    .bigpic{
                        top: -(( $i ) * ($size+20px) - 80px);
                    }
                }
            }
            background: #FFFFFF * 0.3;
        }
    }
}

```

Слика 36 Особини на панелот за слики

внатре се наоѓаат елементите на листата во кои има слики, елементите дизајнирани се така што се прикажуваат ако мали слики и со простор до рабовите на елементот, и се сместени лево од неподредената листа за да се искористи празниот простор за прикажување на сликите на поголема површина, потоа сите први деца на листата ќе бидат малечките сликички што служат за прикажување на вторите поголеми слики со користење на скриптата од слика 36, и рабовите на сликите се исто така заоблени. Потоа има посебен дизајн за големата слика во елементот што ја содржи класата „keep“, веднаш до панелот се наоѓаат општите информации за продуктот покрај цената и насловот се избира и дополнителни информации од корисникот во врска со нарачката, потоа под тие две се наоѓа делот за повеќе информации, изработен е со едно горно мени како што е неподредената листа од главното мени. Исто така искористен беше циклус за позиционирање на големите слики во празниот простор. Следен составен дел од дизајнот е страната за контакт која содржи еден главен елемент кој е поставен во средина со помош на grid layout, и е со фиксна ширина, а висина има колку што ги собира елементите, потоа тој елемент содржи уште два други од кои едниот е форма која се пополнува од страна на корисникот и се испраќа на сервер, полињата се широки колку што е должината на надворешниот елемент, и е додадено заоблување

```

@mixin button-style{
  @include appearance;
  font-size: 8mm;
  border: 2px transparent solid;
  background: $submit-button-color;
  color: #FFFFFF;
  padding: 10px;
  border-radius: 10px;
  margin-left: 10px;
  &:hover{
    border-color: $dark;
    background: darken($submit-button-color, 10);
    color: darken(#FFFFFF, 30);
    cursor: pointer;
  }
}

```

Слика 37 Особините на сите копчиња на веб страната

има и две копчиња што се со сличен дизајн на копчето за контакт, нивниот дизајн е претставен на слика 37.

на рабовите, исто така е специфицирана големина на фонт. А дизајнот на копчето е слично како и линковите од главното мени, веднаш до тој елемент се наоѓа друг елемент што содржи слика и линк и служи за давање дополнителни опции за контакт. Последниот составен дел од дизајнот е формата за логирање, формата содржи два „fieldset“ елементи кои ги обиколуваат полињата и се со иста позадина како и полето, со фиксна должина, оставен простор помеѓу рабовите на главниот елемент и внатрешните елементи,

4. Заклучок

По изработувањето на веб страната, можам да заклучам дека, SCSS е подобар од обичен CSS, со тоа што овозможува поголема флексибилност со тоа што се намалува кодот за пишување, лесно преискористување на веќе напишани делови, страната се состои од повеќе делови. Мојата цел беше да направам копче што би ја менувало темата на целата страна, тоа е многу поефикасно да се направи со SCSS одколку со обичен CSS, бидејќи SCSS содржи многу повеќе функционалности и ни овозможува да имаме променливи кои ги менуваме во зависност од тоа која тема се користи во моментот. Дизајнирањето е процес на позиционирање елементи, тоа е главното нешто што треба да се направи, потоа се ставаат позадините, големините, и другите особини неопходни за поубав изглед. Изгледот е многу важен за корисникот на веб страната. Генерално CSS е многу флексибилен и дозволува со малку повеќе креативност да направиме да изгледа како што сакаме, а SCSS е негова надградба.

5. Користена литература

[1] Sass Basics - <https://sass-lang.com/guide>

[2] Justin Avery - Difference between .sass and .scss - <https://responsivedesign.is/articles/difference-between-sass-and-scss/> - 1 March 2013

[3] The Complete Guide to SCSS/SASS - <https://www.freecodecamp.org/news/the-complete-guide-to-scss-sass-30053c266b23/> - 21 January 2019