Висока школа струковних студија за информационе и комуникационе технологије

Web programiranje – PHP2

Документација за сајт
http://awesomesocks.byethost10.com/
https://github.com/alokin95/Socks_PHP_OOP

Никола Митровић 147-14

Интернет технологије

Фебруар 2018. год.
Београд

1. Увод

1.1 Коришћени програмски језици

HTML, CSS, PHP, JAVASCRIPT, JQUERY, AJAX. Користио сам и MySQL базу података. Цео код је писан у Visual Studio Code.

1.2 Опис функционалности

Admin

email: admin@gmail.com

password: masnocebe

Korisnik

email: carapan@gmail.com

password: masnocebe

Динамички мени

Налази се у header-у сваке стране, линкови се позивају из базе.

Логовање и заштита сесијама

Неауторизовани корисник може да прегледа производе (/products), њихове слике, детаље о њима да види њихове оцене, коментаре које су ауторизовани корисници поставили. (/details) Ауторизовани корисник може да убацује производе у корпу, да брише из корпе,(/account) да гласа за сваки производ посебно, да оставља коментаре на сваком производу посебно. Админ може све што и ауторизовани корисник плус што може да управља садржајем сајта.(/panel)

Анкета

Сваки улоговани корисник може да гласа на производима (/details). Сваки глас се уписује у посебне табеле у бази. У једној се табели чувају идентификатор корисника и идентификатор производа како корисник не може поново да гласа на истом производу, а у другој укупан број гласова и вредност гласова како би се приказала просечна оцена производа. Резултати анкете се шаљу ајаксом, и ајакс одмах враћа нову оцену производа на страницу.

Регистрација и логовање

Постоје странице за регистрацију и логовање (/login, /register).Када неко покуша да се региструје, регуларним изразима се проверава тачност унетих података, и на клијенту и на серверу (Js, PHP). Подаци ново регистрованих корисника се чувају у посебну табелу у бази.

Сваки регистровани корисник може да се улогује са подацима које је унео при регистрацији. Логовањем добија привилегије које су наведене горе. Шифра се проферава функцијом password_hash. Ако је улоговани корисник добио улогу админа отвара му се посебан линк за админ панел који га води на страницу где може да едитује, додаје, брише податке у табелама које не утичу на нормалан рад сајта.

Ајакс

Ајакс се користи за филтрирање производа по полу кликом на radio button(/products). Користи се такође за брисање производа из корисникове корпе(/account), за уписивање и исписивање гласања за сваки производ (/details).

Админ панел

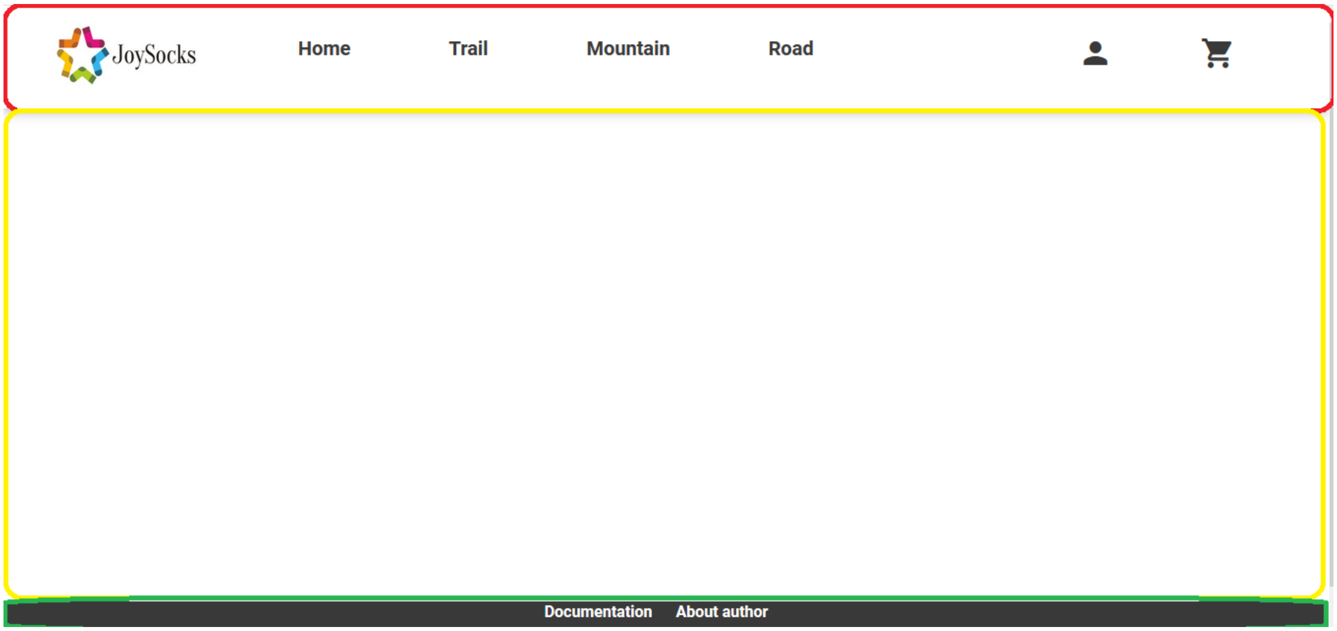
Админ добија на коришћење странице која може да манипулише свим табелама из базе. Табеле су урађене као carousel, администратор бира табелу кликом на дугме које је у врху, одмах испод навигације.

1.3 Template

Header

Main

Footer



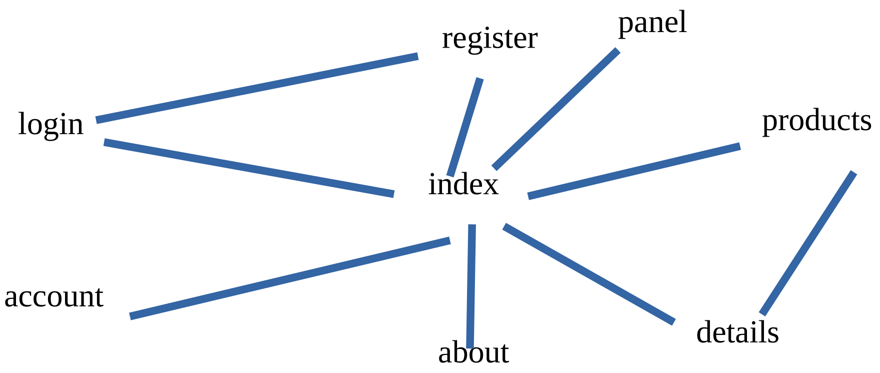
```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <link rel="icon" href="assets/img/other/favicon.svg" type="image/gif" sizes="16x16">
  <meta name="description" content="Buy socks. Great socks for sale for men, women and kids.">
  <meta name="keywords" content="Socks, shop, ecommerce, buy, sell, comment, products, trail, mountain, blister">
  <meta name="author" content="Nikola Mitrovic">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://fonts.googleapis.com/css?family=Roboto:700" rel="stylesheet">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <link rel='stylesheet' type='text/css' href='/assets/css/main.css'>
  <title>Socks Awesome</title>
</head>
```

```
<body>
  <header class='flex-row'>
    <div class='logo'>
     <a href="index"><img src='assets/img/other/logo.jpg' width="180px" height="100px"></a>
    </div>
    <nav>
      <ul>
        <?php

            $get_nav = "SELECT * FROM nav";

            $nav = $conn->execute($get_nav);


             while ($n = mysqli_fetch_object($nav)){
                echo "<li><a href='".$n->link."'>".$n->title."</a></li>";
             }
          ?>
      </ul>
    </nav>
    <div class='cart flex-row'>
      <?php
      if (isset($_SESSION['role']) && $_SESSION['role']==1){ ?>
      <a href='panel'>
        <i class="material-icons" style="font-size:36px">person</i>
        <p>ADMIN PANEL</p>
      </a>

      <?php }else { ?>
        <a href='login'>
        <i class="material-icons" style="font-size:36px">person</i>
      </a>
      <?php } ?>
      <a href='account'>
        <i class="material-icons" style="font-size:36px">shopping_cart</i>
      </a>

      <?php if (isset($_SESSION['email'])):?>

      <a href='logout.php'><i class="material-icons" style="font-size:36px">close</i></a>


      <?php endif; ?>
    </div>
  </header>
  <main>

<footer>
<a href="/documentation.pdf" download>Documentation</a>
  <a href="about">About author</a>
</footer>
</main>

</body>
<script src="/assets/js/script.js"></script>

</html>
```

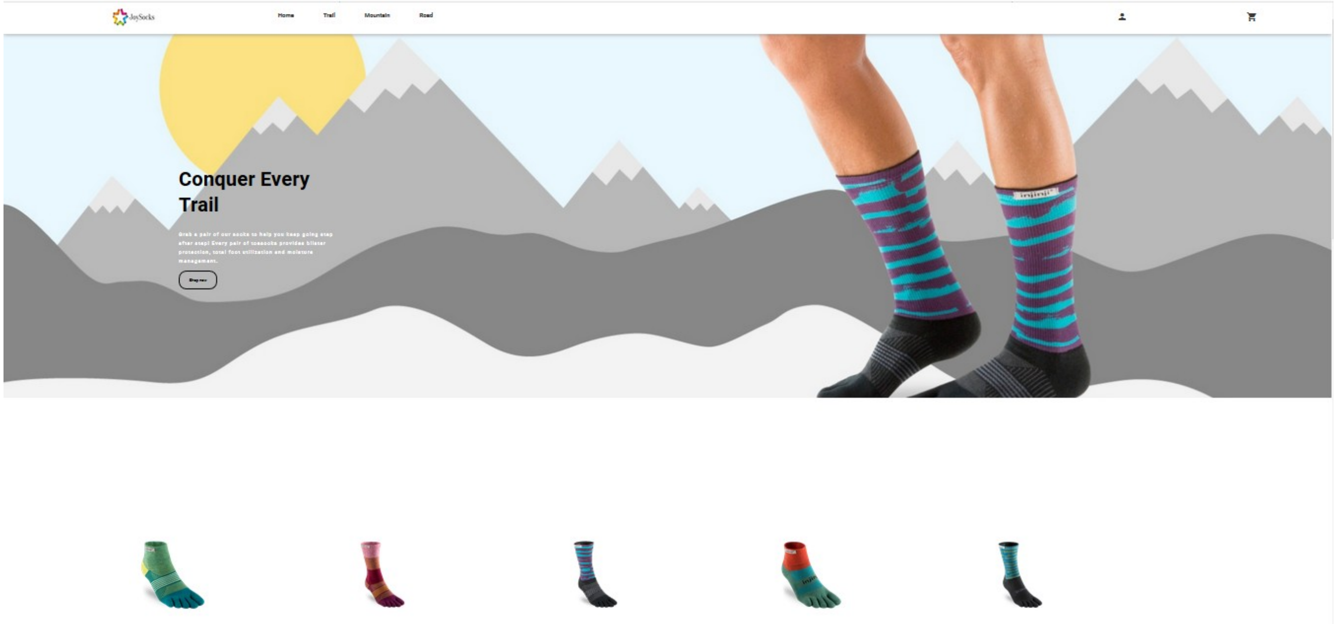
1.4 Није коришћен туђи CSS код

2. Организација

2.1 Организациона шема



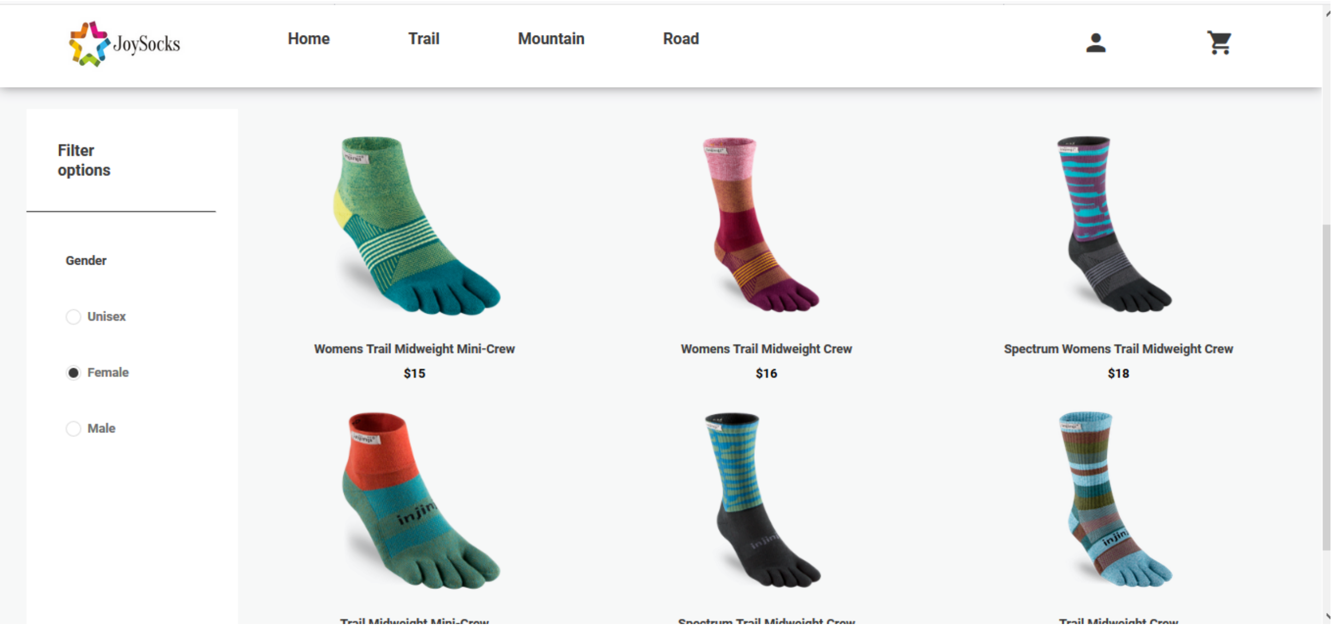
2.2 Странице



Grab a pair of our socks to help you keep going step after step! Every pair of toesocks provides blister protection, total foot utilization and moisture management.

views/home.template.php

На овој страници се, осим динамичког менија који је приказан на свакој страни, налазе првих пет производа из базе.



Filter options

Gender

○ Unisex

● Female

○ Male

Womens Trail Midweight Mini-Crew
$15

Womens Trail Midweight Crew
$16

Spectrum Womens Trail Midweight Crew
$18

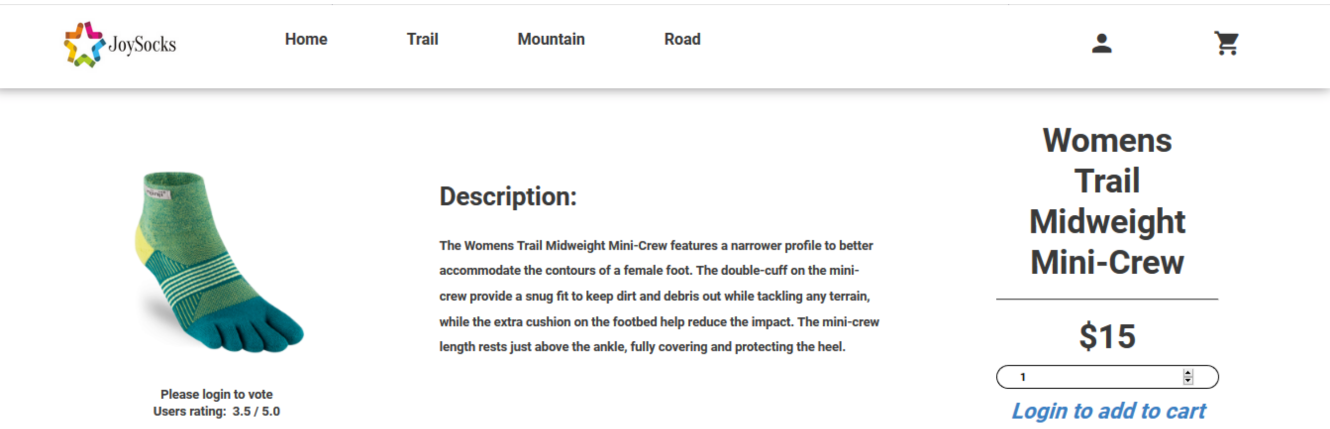
Trail Midweight Mini-Crew

Spectrum Trail Midweight Crew

Trail Midweight Crew

views/product.template.php

На овој страни се излиставају производи по категоријама и полу из базе. Са леве стране се налазе дугмићи преко којих се ајаксом филтрира пол (Мушки, женски, унисекс).



Description:

The Womens Trail Midweight Mini-Crew features a narrower profile to better accommodate the contours of a female foot. The double-cuff on the mini-crew provide a snug fit to keep dirt and debris out while tackling any terrain, while the extra cushion on the footbed help reduce the impact. The mini-crew length rests just above the ankle, fully covering and protecting the heel.

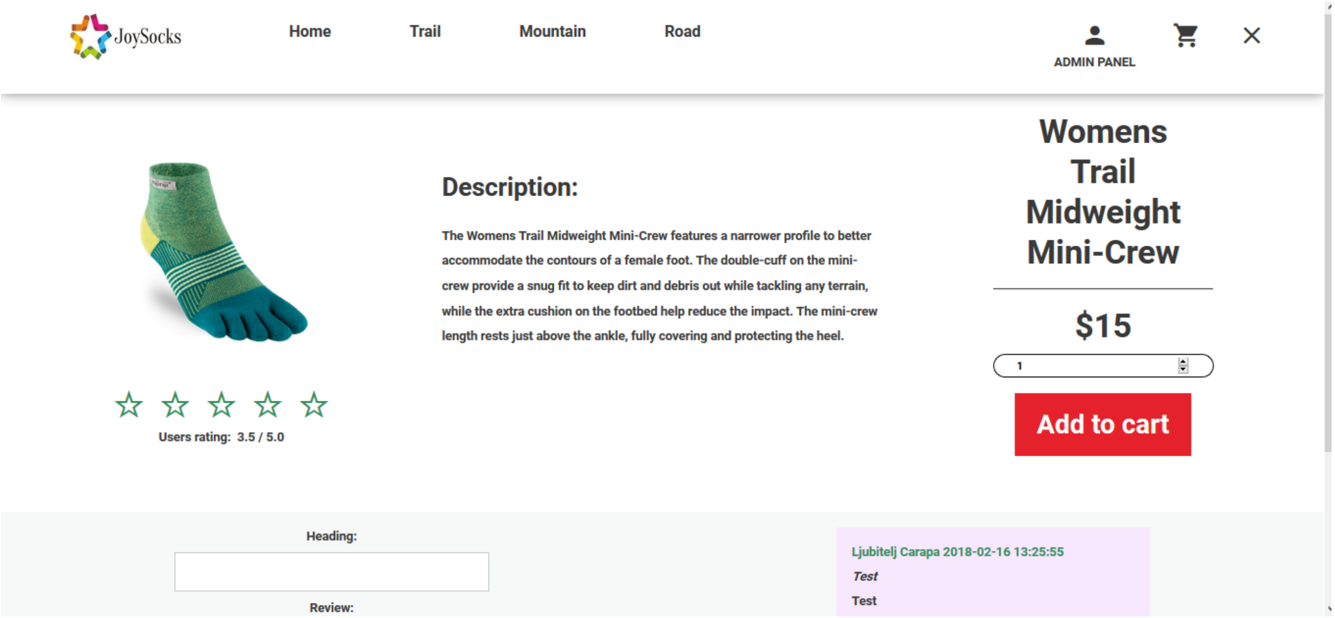
Please login to vote
Users rating: 3.5 / 5.0

Womens Trail Midweight Mini-Crew

$15

1

Login to add to cart



Description:

The Womens Trail Midweight Mini-Crew features a narrower profile to better accommodate the contours of a female foot. The double-cuff on the mini-crew provide a snug fit to keep dirt and debris out while tackling any terrain, while the extra cushion on the footbed help reduce the impact. The mini-crew length rests just above the ankle, fully covering and protecting the heel.

Womens Trail Midweight Mini-Crew

$15

1

☆ ☆ ☆ ☆ ☆

Users rating: 3.5 / 5.0

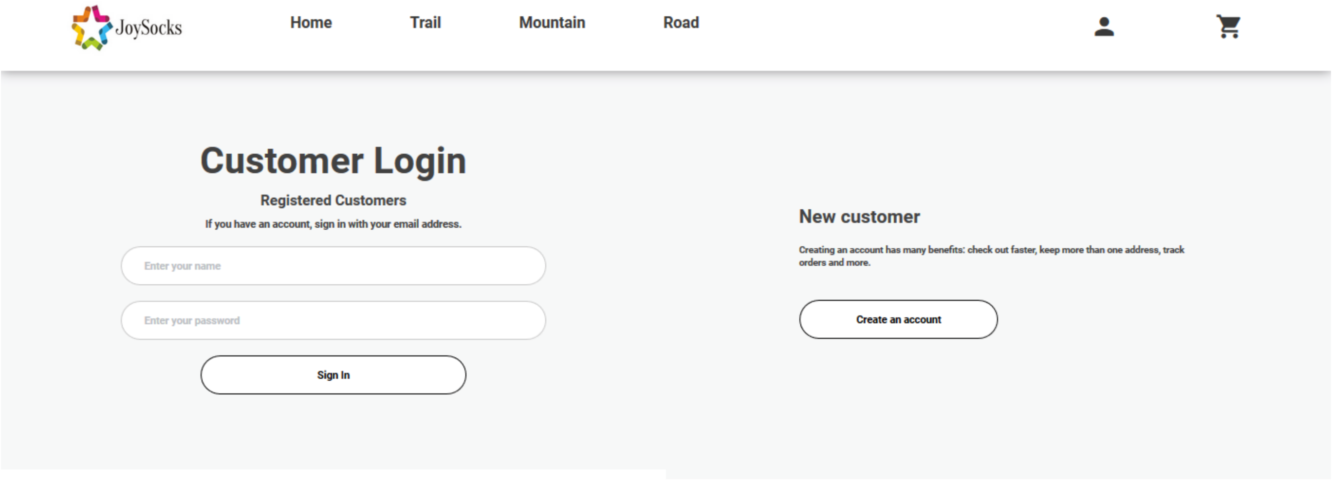
Ljubitelj Carapa 2018-02-16 13:25:55

Test

Test

views/details.template.php

На овој страни се из базе приказују подаци производа на који је кликнут. Ако је корисник улогован даје му се могућност да гласа (ајакс), дода производ у своја колица и да остави коментар на тај одређени производ.



Customer Login

Registered Customers

If you have an account, sign in with your email address.

Enter your name

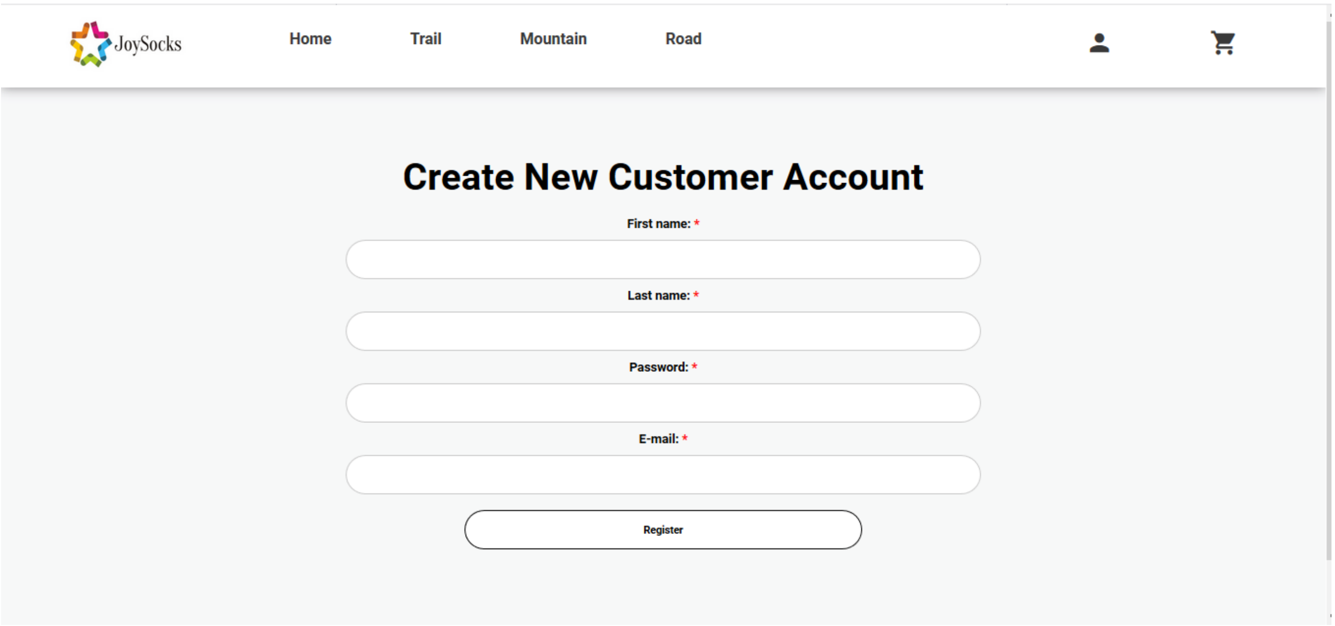
Enter your password

New customer

Creating an account has many benefits: check out faster, keep more than one address, track orders and more.

views/login.template.php

Са леве стране је форма за логовање, са десне је линк ка страни за регистрацију.



Create New Customer Account

First name: *

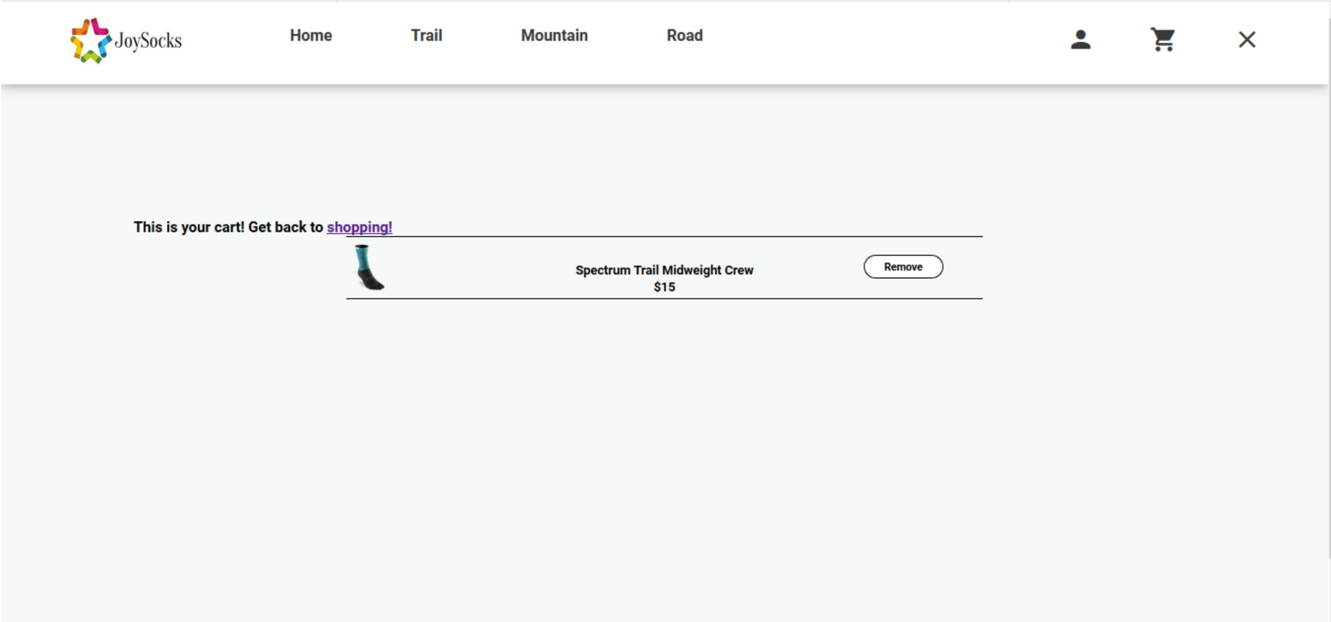
Last name: *

Password: *

E-mail: *

views/register.template.php

Страна за регистрацију корисника. Све се проверава регуларним изразима, клијентски и серверски.

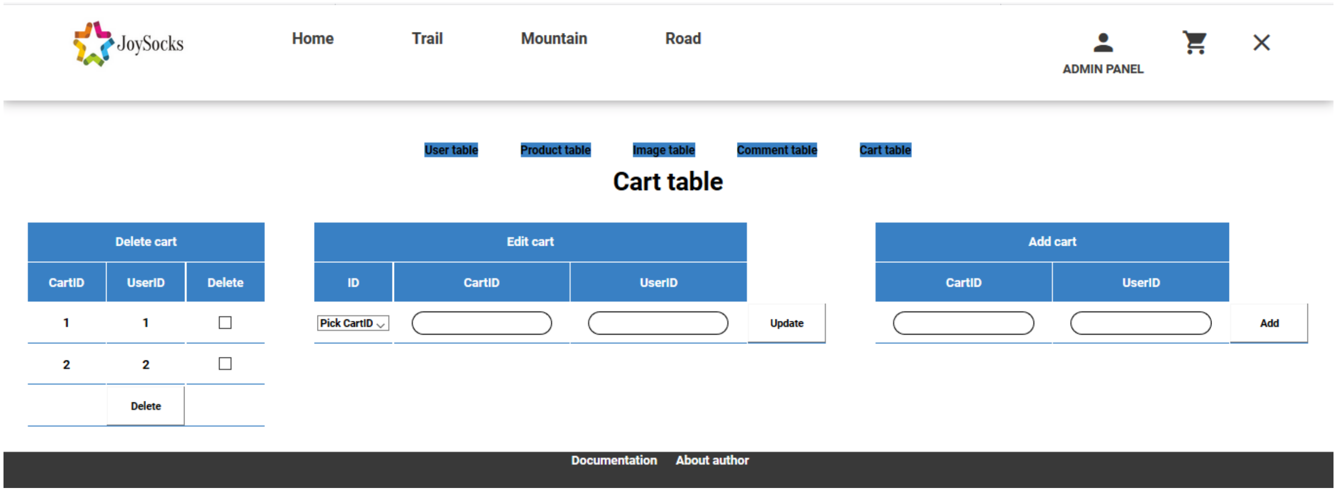


This is your cart! Get back to shopping!

Spectrum Trail Midweight Crew
$15

views/account.template.php

Ово је страница коју улоговани корисник види када у корпу убаци један производ. Производ може да обрише. То се извршава уз помоћ ајакса.



Cart table

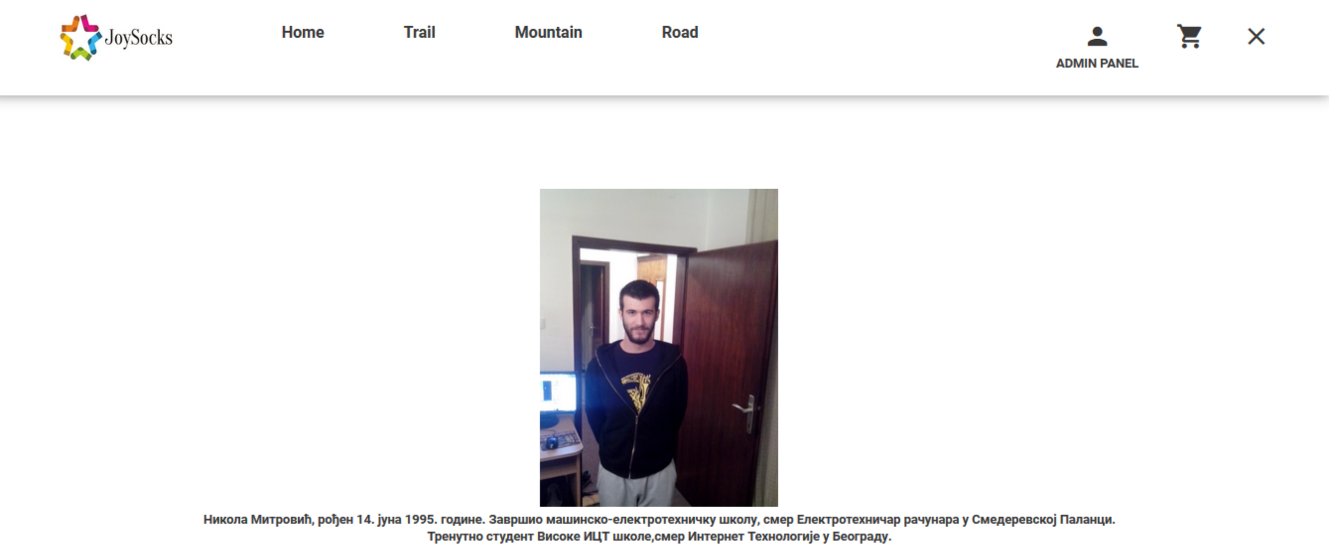
Delete cart		
CartID	UserID	Delete
1	1	□
2	2	□
	Delete	

Edit cart			
ID	CartID	UserID	
Pick CartID			Update

Add cart		
CartID	UserID	
		Add

views/admin/admin.template.php

Страница коју администратор види. Кликом на плаве линкове са црним словима му се приказује захтевана табела. Могуће је брисање, додавање и мењање података из табеле.



Никола Митровић, рођен 14. јуна 1995. године. Завршио машинско-електротехничку школу, смер Електротехничар рачунара у Смедеревској Паланци.
Тренутно студент Високе ИЦТ школе,смер Интернет Технологије у Београду.

views/about.template.php

3. Кодови

Важно: сајт је рађен као нека врста Custom MVC-a. Постоји фолдер controllers који садржи фајлове. Ти фајлови позивају одговарајуће фајлове из фолдера models(у фолдеру models се налазе класе које служе за комуникацију са одговарајућом табелом из базе, затим се у контролерима прави инстанца те позване класе, позива нека функција из модела и резултати се делегирају у неки од фајлова у фолдеру views. Ту је и фолдер admin у којем се налазе контролери за админ панел, као и фолдер ајакс који се бави ајакс захтевима. Окачио сам на првој страни линк ка github репозиторијуму где се налази цео код тако да се лакше може испратити.

core/Route.php

```
<?php


class Route {


    private $routes = [
        'GET' => [],
        'POST' => []
    ];


    public function get($uri, $controller) {


        $this->routes['GET'][$uri] = $controller;


    }
    public function post($uri, $controller)  {
```

```
    $this->routes['POST'][$uri] = $controller;

  }
  public function require($uri, $request){

    if (array_key_exists($uri, $this->routes[$request])) {

      return $this->routes[$request][$uri];

    }
    else {

     return $this->routes['GET'][''];

    }
  }
}
```

core/Connection.php

```
<?php

 class Connection {

  private $conn;

  public function __construct() {

   $this->conn = mysqli_connect("localhost", "root", "", "socks1");

  }
```

```
    public function getConnection() {
      return $this->conn;
    }
    public function execute($query) {
      $result = mysqli_query($this->conn, $query);
      return $result;
    }
  }
```

core/bootstrap.php

```
<?php
  // $config = require_once 'config.php';
  require_once 'Connection.php';
  $conn = new Connection();
  require_once 'Route.php';
  function view($template, $data = '1') {
    return 'views/'.$template.'.template.php';

  }
```

/index.php

```
<?php
  session_start();

  require_once 'core/bootstrap.php';

  require_once 'views/partials/header.php';

  $router = new Route;

  //GET ROUTES
  $router->get('','controllers/home.php');
  $router->get('index','controllers/home.php');
  $router->get('products','controllers/products.php');
  $router->get('details','controllers/details.php');
  $router->get('logout', 'controllers/home.php');
  $router->get('account','controllers/cart_show.php');
  $router->get('register','views/register.template.php');
  $router->get('login','views/login.template.php');
  $router->get('about', 'views/about.template.php');

  //POST ROUTES
  $router->post('products', 'controllers/comment_add.php');
  $router->post('account', 'controllers/cart_insert.php');
  $router->post('register','controllers/register.php');
  $router->post('index','controllers/login.php');

  //ADMIN ROUTES
  $router->get('panel', 'admin/admin_index.php');
  $router->post('cart', 'admin/admin_cart.php');
  $router->post('user', 'admin/admin_user.php');
  $router->post('product', 'admin/admin_product.php');
  $router->post('image', 'admin/admin_image.php');
  $router->post('comment', 'admin/admin_comment.php');



  $uri = trim(parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH),'/');
  $method = $_SERVER['REQUEST_METHOD'];

  require $router->require($uri, $method);

  require_once 'views/partials/footer.php';
```

controllers/cart_insert.php

```
<?php

  require_once 'classes/Cart.php';

  if (!isset($_SESSION['cart'])){
    echo "You must login first";
  }
  else {
  $cart_session = $_SESSION['cart'];
  $cart = new Cart;

  $cart->insert($cart_session, $_POST['productid'], $_POST['quantity']);


  $result = $cart->show($cart_session);

  require_once view('account', $result);
}
```

controllers/cart_show.php

```
<?php

  if (!isset($_SESSION['cart'])){

    Header("Location:login");

  }
  else {

  require_once 'classes/Cart.php';

  $cart = new Cart;

  $result = $cart->show();

  require_once view('account', $result);
}
```

controllers/comment_add.php

```
<?php

  require_once 'classes/Comment.php';



    if (isset($_POST['review'])){
      $userid = $_SESSION['userid'];
      $heading = $_POST['heading'];
      $content = $_POST['content'];
      $productid = $_POST['productid'];

      $comm = new Comment;

      $comm->addComment($heading, $content, $userid, $productid);

  }



require_once view('home');
```

controllers/details.php

```
<?php

if(empty($_REQUEST['productid']))
   {
     $_GET['productid'] = 1;
   }

  require_once 'classes/Product.php';
  require_once 'classes/Comment.php';
  require_once 'classes/Rating.php';

  $detailsPage = new Product();
  $comm = new Comment();
  $rating = new Rating();

  $details['comment'] = $comm->showComment();

  $details['product'] = $detailsPage->getInfo();

  $productid = $_GET['productid'];
  $details['ratings'] = $rating->getRating($productid);

  $result = extract($details);

  require_once view('details', $result);
```

controllers/home.php

```
<?php

  require_once 'classes/Product.php';

  $product = new Product;

  $row = $product->top();
  require_once view('home', $row);
```

controllers/login.php

```
<?php



  require_once 'classes/User.php';

  if (isset($_POST['login'])) {

    $user = new User;

    $result = $user->login();

    if ($result) {

      Header('Location:/index');

    }
    Header('Location:/login');

  }
```

controllers/products.php

```
<?php

  if(empty($_REQUEST['category']))
   {
     $_GET['category'] = 1;
   }
  require_once 'classes/Product.php';

  $product = new Product();

  $result = $product->getProducts();

  require view('products', $result);
```

controllers/register.php

```
<?php

  require_once 'classes/User.php';

  if (isset($_POST['register'])){



    $user = new User;

    $user->register();

    //require_once 'views/register.template.php';

    require_once view('register');

  }
```

classes/Cart.php

```
<?php

  class Cart extends Connection {

    private $session;

    public function __construct() {

      parent::__construct();
      $this->session = $_SESSION['cart'] ?? null;

    }

    public function getAll(){

      $sql = "SELECT * FROM cart";

      return $this->execute($sql);

    }

    public function singleCart($id){

      $sql = "SELECT * FROM cart WHERE cartid=".$id;

      return $this->execute($sql);
    }

    public function delete(){
      $delete = implode(', ' ,$_POST['for-delete-cart']);

      $sql = "DELETE FROM cart WHERE cartid in ($delete)";

      $this->execute($sql);

    }

    public function insertCart(){

      $cartid = $_POST['cartid'];
      $userid = $_POST['userid'];

      $sql = "INSERT INTO cart VALUES('$cartid', '$userid')";

      $this->execute($sql);
    }

    public function show() {

      // $this->session = $session;

      // $sql = "SELECT * FROM user INNER JOIN cart ON user.userid = cart.userid INNER JOIN
cart_items ON cart.cartid = cart_items.cartid INNER JOIN product ON cart_items.productid =
product.productid INNER JOIN image ON image.imageid = product.imageid WHERE user.email ='".
$this->session."'";

      $sql = "SELECT * FROM `cart_items` INNER JOIN product ON
cart_items.productid=product.productid INNER JOIN image ON image.imageid=product.imageid
WHERE cart_items.cartid ='".$this->session."'";
```

```
  $result = $this->execute($sql);

  return $result;

}

public function insert($session, $productid, $quantity){


  // $this->session = $session;

  $insert_cart = "INSERT INTO cart_items VALUES ('','$session', '$productid', '$quantity')";

  $inserted = $this->execute($insert_cart);

}

public function update(){

  $cartid = $_POST['cartid'];
  $updated_cartid = $_POST['update-cartid'];
  $userid = $_POST['update-userid'];

  $sql = "UPDATE cart SET cartid='$updated_cartid', userid='$userid' WHERE cartid='$cartid'";

  return $this->execute($sql);

}

public function removeFromCart($id) {
      $sql = "DELETE FROM cart_items WHERE cartid='$this->session' AND id='$id'";

      $this->execute($sql);

      return $this->show();

    }
  }
```

classes/Comment.php

```
<?php

  class Comment extends Connection {

    private $productid;

    public function __construct(){

      parent::__construct();

      $this->productid = $_GET['productid'] ?? 1;

    }

    public function addComment($heading, $content, $userid, $productid){

      $date = date('Y-m-d H:i:s');
```

```
        $insert_comment = "INSERT INTO comment VALUES(", '$heading', '$content', '$date', '$userid', '$productid')";

        $this->execute($insert_comment);

    }

    public function showComment(){

        $sql = "SELECT * FROM comment INNER JOIN user ON comment.userid = user.userid WHERE productid =".$this->productid;
        $result = $this->execute($sql);

        return $result;

    }

    public function getAll(){

        $sql = "SELECT * FROM comment";

        return $this->execute($sql);

    }

    public function delete() {

        $delete = implode(', ', $_POST['for-delete-comment']);

        $sql = "DELETE FROM comment WHERE commentid in ($delete)";
```

```
      $this->execute($sql);
    }

    public function getSingle($id) {

      $sql = "SELECT * FROM comment WHERE commentid=".$id;

      return $this->execute($sql);
    }

    public function update() {

      $id = $_POST['commentid'];
      $head = $_POST['update-heading'];
      $content = $_POST['update-content'];
      $date = $_POST['update-date'];
      $user = $_POST['update-c-userid'];
      $product = $_POST['update-c-productid'];

      $sql = "UPDATE comment SET heading='$head', content='$content', date_created='$date', userid='$user', productid='$product' WHERE commentid='$id'";

      $this->execute($sql);
    }

    public function insert(){

      $head = $_POST['heading'];
      $content = $_POST['content'];
      $user = $_POST['comment-userid'];
      $product = $_POST['comment-productid'];

      $date = date('Y-m-d H:i:s');

      $sql = "INSERT INTO comment VALUES('', '$head', '$content', '$date', '$user', '$product')";

      $this->execute($sql);

    }

  }
```

classes/Image.php

```
<?php

  class Image extends Connection {

    public function __construct() {

      parent::__construct();

    }

    public function single($id) {

      $imageid = $_POST['imageid'];

      $sql = "SELECT * FROM image WHERE imageid ='$imageid'";
      return $this->execute($sql);

    }

    public function getAll() {

      $sql = "SELECT * FROM image";

      return $this->execute($sql);

    }

    public function insert() {

      $alt = $_POST['alt'];

      $image=$_FILES['src']['name'];
                          $tmp=$_FILES['src']['tmp_name'];
                          $file=$_FILES['src']['type'];

      $path = "assets/img/products/".$image;

        if (move_uploaded_file($tmp, $path)){
          $sql = "INSERT INTO image VALUES (",'$path', '$alt')";

        $this->execute($sql);
        }
    }

        public function update() {
          $id = $_POST['imageid'];
          $src = $_POST['update-src'];
          $alt = $_POST['update-alt'];

          $sql = "UPDATE image SET src='$src', alt='$alt' WHERE imageid='$id'";

          $this->execute($sql);

        }




    public function delete() {

      $delete = implode(', ', $_POST['for-delete-image']);

      $sql = "DELETE FROM image WHERE imageid IN  ($delete)";

      $this->execute($sql);

    }
  }
```

classes/Product.php

```
<?php

  class Product extends Connection{
```

```
private $categoryid;
private $productid;

public function __construct() {

  parent::__construct();
  $this->categoryid = $_GET['category'] ?? 1;
  $this->id = $_GET['productid'] ?? 1;

}

public function single($id) {

  $sql = "SELECT * FROM product WHERE productid=".$id;

  return $this->execute($sql);
}

public function getAll(){

  $sql = "SELECT * FROM product";

  return $this->execute($sql);
}

public function top(){

  $sql = "SELECT * FROM product JOIN image ON product.imageid=image.imageid LIMIT 5";

  return $this->execute($sql);
```

```
    }

    public function getProducts() {

      $sql = "SELECT * FROM product p JOIN category c ON p.categoryid = c.categoryid JOIN image i ON p.imageid = i.imageid WHERE p.categoryid = ".$this->categoryid;
      $result = $this->execute($sql);

      return $result;

    }

    public function getProductByGender($gender){


      $sql = "SELECT * FROM product p JOIN category c ON p.categoryid = c.categoryid JOIN image i ON p.imageid = i.imageid WHERE p.genderid ='$gender'";

      $result = $this->execute($sql);

      return $result;
    }

    public function getInfo() {

      $sql = "SELECT productid AS id, p.name as productname, description as productdescription, price as productprice, p.categoryid as catid, genderid as genderid, p.imageid as imgid, src, alt FROM product p JOIN category c ON p.categoryid = c.categoryid JOIN image i ON p.imageid = i.imageid WHERE productid =".$this->id;
```

```
  $result = $this->execute($sql);

  $r = mysqli_fetch_object($result);

  return $r;

}

public function insert(){

  $name = $_POST['product'];
  $desc = $_POST['desc'];
  $price = $_POST['price'];
  $cat = $_POST['cat'];
  $gender = $_POST['gender'];
  $image = $_POST['image'];

  $sql = "INSERT INTO product VALUES (",'$name', '$desc', '$price', '$cat', '$gender', '$image')";

  $inserted = $this->execute($sql);

  $last_id = mysqli_insert_id($this->getConnection());

  if ($inserted) {

    $add_rating = "INSERT INTO rating VALUES (",",",'$last_id')";

    $this->execute($add_rating);
```

```
      }


    }

    public function delete() {

      $delete = implode(', ', $_POST['for-delete-product']);

      $sql = "DELETE FROM product WHERE productid IN ($delete)";

      $this->execute($sql);

    }

    public function update() {

      $id = $_POST['productid'];
      $name = $_POST['update-name'];
      $desc = $_POST['update-desc'];
      $price = $_POST['update-price'];
      $cat = $_POST['update-cat'];
      $gender = $_POST['update-gender'];
      $image = $_POST['update-image'];

      $sql = "UPDATE product SET name='$name', description='$desc', price='$price', categoryid='$cat', genderid='$gender', imageid='$image' WHERE productid='$id'";

      $this->execute($sql);
    }

  }
```

classes/Rating.php

```
<?php



  class Rating extends Connection {

    public function __construct(){

      parent::__construct();

    }

    public function getRating($productid){

      $sql = "SELECT ROUND(sum/number_of_votes, 1) as rating FROM rating WHERE productid='$productid'";

      $result = $this->execute($sql);

      return mysqli_fetch_object($result);
    }

     public function checkVoted($userid, $productid) {

      $sql = "SELECT * FROM rating_log WHERE userid='$userid' AND productid='$productid'";
```

```
   return $this->execute($sql);

   }

  public function vote($userid, $productid, $value) {

   $check_voted = $this->checkVoted($userid, $productid);

   if (mysqli_num_rows($check_voted)>0){

    $output = "You already voted";

   }
   else {

    $vote = "UPDATE rating SET number_of_votes = number_of_votes+1, sum = sum+'$value'
WHERE productid = '$productid'";

    if ($this->execute($vote)) {

     $vote_log = "INSERT INTO rating_log VALUES ('','$productid', '$userid')";

     $this->execute($vote_log);

     $output = $this->getRating($productid);
    }

   }

   return $output;
```

```
    }
  }
```

classes/User.php

```
<?php


  class User extends Connection{


    private $first;
    private $last;
    private $pass;
    private $email;
    private $session;


    public function __construct() {


      parent::__construct();


    }


    public function getAll(){


      $sql = "SELECT * FROM user";


      return $this->execute($sql);


    }


    public function update(){
      $id = $_POST['userid'];
      $first = $_POST['update-first'];
      $last = $_POST['update-last'];
      $email = $_POST['update-email'];
      $pass = password_hash($_POST['update-password'], PASSWORD_DEFAULT);
      $role = $_POST['update-roleid'];

      $sql = "UPDATE user SET first_name='$first', last_name='$last', email='$email',
password='$pass',roleid='$role' WHERE userid='$id'";

      $this->execute($sql);

    }

    public function insert() {

      $first = $_POST['firstname'];
      $last = $_POST['lastname'];
      $email = $_POST['email'];
      $pass = password_hash($_POST['password'], PASSWORD_DEFAULT);
      $role = $_POST['roleid'];

      $sql = "INSERT INTO user VALUES ('','$first','$last','$pass','$email','$role')";

      $inserted = $this->execute($sql);

      $last_id = mysqli_insert_id($this->getConnection());

      if ($inserted) {
```

```
    $cart_sql = "INSERT INTO cart VALUES (",'$last_id')";
    $this->execute($cart_sql);

  }

}

public function getUser($id) {

  $sql = "SELECT * FROM user WHERE userid=".$id;

  return $this->execute($sql);

}

public function delete(){

  $delete = implode(', ', $_POST['for-delete-user']);

  $sql = "DELETE FROM user WHERE userid in ($delete)";

  $this->execute($sql);

}

 public function login(){

  $this->email = $_POST['email'];;
  $this->pass = $_POST['password'];;
```

```
      $sql = "SELECT * FROM user INNER JOIN cart ON user.userid=cart.userid WHERE email = '$this->email'";

      $result = $this->execute($sql);

      if (mysqli_num_rows($result) == 1){

        $user = mysqli_fetch_object($result);

        $hashed_pass = $user->password;

        if (password_verify($this->pass ,$hashed_pass)){

        $_SESSION['email'] = $user->email;
        $_SESSION['role'] = $user->roleid;
        $_SESSION['cart'] = $user->cartid;
        $_SESSION['userid'] = $user->userid;

        return $result;

      }
      else {
        Header("location:login");
      }

     }

    }
```

```
public function register() {

 $this->first = $_POST['firstName'];
 $this->last = $_POST['lastName'];
 $this->pass = password_hash($_POST['password'], PASSWORD_DEFAULT);
 $this->email = $_POST['email'];

 $regMail = "/^\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,3}$/";
 $regName="/^[A-Za-z]{4,15}$/";

 $errors = [];

 if(!preg_match($regName ,$this->first ))
 {
   $errors[]="First name is in bad format";
 }

 if(!preg_match($regName ,$this->last ))
 {
   $errors[]="Last name is in bad format";
 }

 if(!preg_match($regMail ,$this->email ))
 {
   $errors[]="Email is in bad format";
 }

 if (empty($this->pass)){
   $errors[] ="Password is empty";
 }
```

```
if (count($errors)>0) {
  // Header:("location:register");
  foreach ($errors as $e){
    echo $e."<br/>";
  }


}
else {


$check_existing = "SELECT * FROM user WHERE email ='$this->email'";


$checked = $this->execute($check_existing);


if (mysqli_num_rows($checked)>0){
  echo "Email already in use";
}
else {


  $register = "INSERT INTO user VALUES (",'$this->first','$this->last','$this->pass','$this->email','2')";


  $registered = $this->execute($register);


  $last_id = mysqli_insert_id($this->getConnection());


  if ($registered) {


  $cart_sql = "INSERT INTO cart VALUES (",'$last_id')";
  $this->execute($cart_sql);
```

```
      }

      }

    }
    }

    // public function showCart($session) {

    //   $this->session = $session;

    //   $sql = "SELECT * FROM user INNER JOIN cart ON user.userid = cart.userid INNER JOIN cart_items ON cart.cartid = cart_items.cartid INNER JOIN product ON cart_items.productid = product.productid INNER JOIN image ON image.imageid = product.imageid WHERE user.email ='". $this->session."'";

    //   $result = $this->execute($sql);

    //   return $result;

    // }
  }
```

admin/admin_cart.php

```
<?php

  require_once 'classes/Cart.php';
  $cart = new Cart;

  if (isset($_POST['cart-delete'])){
    $cart->delete();

  }

  if (isset($_POST['cart-insert'])){

    $cart->insertCart();

  }

  if (isset($_POST['cart-update'])) {

    $cart->update();

  }



  Header("Location:/panel");
```

admin/admin_comment.php

```
<?php

  require_once 'classes/Comment.php';

  $comment = new Comment;

  if (isset($_POST['comment-delete'])){

    $comment->delete();
  }

  if (isset($_POST['comment-update'])) {

    $comment->update();

  }

  if (isset($_POST['comment-insert'])){


    $comment->insert();
  }

  Header("Location:panel");
```

admin/admin_image.php

```
<?php

  require_once 'classes/Image.php';
  $image = new Image;

  if (isset($_POST['image-delete'])){

    $image->delete();

  }

  if (isset($_POST['image-update'])) {
    $image->update();

  }

  if (isset($_POST['image-insert'])){

    $image->insert();

  }

  Header("Location:panel");
```

admin/admin_index.php

```
<?php

  if ($_SESSION['role']!=1){
    header("Location:index");
  }

  require 'classes/Cart.php';
  require 'classes/User.php';
  require 'classes/Product.php';
  require 'classes/Image.php';
  require 'classes/Comment.php';

  //CART
  $cart = new Cart;

  $result = $cart->getAll();
```

```
$data['cart'] = [];

while ($row = mysqli_fetch_object($result)) {
  $data['cart'][] = $row;
}

//USER
$user = new User;

$result = $user->getAll();

$data['user'] = [];

while ($row = mysqli_fetch_object($result)){
  $data['user'][] = $row;
}


//PRODUCT
$product = new Product;

$result = $product->getAll();

$data['product'] = [];

while ($row = mysqli_fetch_object($result)){
  $data['product'][] = $row;
}
```

```
//IMAGE
$image = new Image;

$result = $image->getAll();

$data['image'] = [];

while ($row = mysqli_fetch_object($result)) {
  $data['image'][] = $row;
}



//COMMENT

$comment = new Comment;

$result = $comment->getAll();

$data['comment'] = [];

while ($row = mysqli_fetch_object($result)) {
  $data['comments'][] = $row;
}

require view('admin/admin', extract($data));

//header("Refresh:0");
```

admin/admin_product.php

```
<?php


  require_once 'classes/Product.php';
  $product = new Product;


  if (isset($_POST['product-delete'])){


    $product->delete();


  }


  if (isset($_POST['product-update'])) {


    $product->update();


  }


  if (isset($_POST['product-insert'])){



    $product->insert();
  }


  Header("Location:panel");
```

admin/admin_user.php

```
<?php


  require_once "classes/User.php";
  $user = new User;
  if (isset($_POST['user-delete'])){
    $user->delete();
  }
  if (isset($_POST['user-update'])){
    $user->update();
  }

  if (isset($_POST['user-insert'])){
    $user->insert();
  }

  Header("Location:panel");
```

ajax/cart_update.php

```
<?php

  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/Cart.php');

  $cart = new Cart;
  $id = $_POST['cartid'];
  $result = $cart->singleCart($id);

  $output = [];

  while ($row = mysqli_fetch_object($result)){
    $output[] = $row;
  }

  echo json_encode($output);
```

ajax/comment_update.php

```
<?php

  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/Comment.php');

  $comment = new Comment;
  $id = $_POST['value'];
  $result = $comment->getSingle($id);

  $output = [];

  while ($row = mysqli_fetch_object($result)){
    $output[] = $row;
  }

  echo json_encode($output);
```

ajax/gender.php

```
<?php


  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/Product.php');
  // require __DIR__ . "/classes/Product.php";
  // require __DIR__ . "/core/bootstrap.php";

  $product = new Product;

  $gender = $_POST['gender'];

  $result = $product->getProductByGender($gender);;

  $output=[];
  while ($r = mysqli_fetch_object($result)){
   $output[]=$r;
  }
  echo json_encode($output);
```

ajax/image_update.php

```
<?php


  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/Image.php');

  $image = new Image;
  $id = $_POST['imageid'];
  $result = $image->single($id);
  $output = [];


  while ($row = mysqli_fetch_object($result)){
    $output[] = $row;
  }


  echo json_encode($output);
```

ajax/item_remove.php

```
<?php


  session_start();


  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/Cart.php');


  $productid = $_POST['productid'];


  $cart = new Cart;


  $result = $cart->removeFromCart($productid);


  $output = [];
  while ($row = mysqli_fetch_object($result)) {
    $output [] = $row;
  }
  echo json_encode($output);
```

ajax/product_update.php

```
<?php


  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/Product.php');

  $product = new Product;
  $id = $_POST['productid'];
  $result = $product->single($id);

  $output = [];

  while ($row = mysqli_fetch_object($result)){
    $output[] = $row;
  }

  echo json_encode($output);
```

ajax/rating.php

```
<?php
  session_start();

  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/Rating.php');

  $rate = new Rating;

  $userid = $_SESSION['userid'];

  $productid = $_POST['productid'];
  $vote_value = $_POST['value'];

  $result = $rate->vote($userid, $productid, $vote_value);

  echo json_encode($result);
```

ajax/user_update.php

```
<?php

  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/core/bootstrap.php');
  require_once(realpath($_SERVER["DOCUMENT_ROOT"]) .'/classes/User.php');

  $user = new User;
  $id = $_POST['userid'];
  $result = $user->getUser($id);

  $output = [];

  while ($row = mysqli_fetch_object($result)){
    $output[] = $row;
  }

  echo json_encode($output);
```

views

views/partials/header.php

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <link rel="icon" href="assets/img/other/favicon.svg" type="image/gif" sizes="16x16">
  <meta name="description" content="Buy socks. Great socks for sale for men, women and kids.">
  <meta name="keywords" content="Socks, shop, ecommerce, buy, sell, comment, products, trail, mountain, blister">
  <meta name="author" content="Nikola Mitrovic">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://fonts.googleapis.com/css?family=Roboto:700" rel="stylesheet">
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
  <link rel='stylesheet' type='text/css' href='/assets/css/main.css'>
  <title>Socks Awesome</title>
</head>

<body>
  <header class='flex-row'>
    <div class='logo'>
     <a href="index"><img src='assets/img/other/logo.jpg' width="180px" height="100px"></a>
    </div>
    <nav>
      <ul>
        <?php

            $get_nav = "SELECT * FROM nav";

            $nav = $conn->execute($get_nav);



             while ($n = mysqli_fetch_object($nav)){
                echo "<li><a href='".$n->link."'>".$n->title."</a></li>";
             }
          ?>
      </ul>
    </nav>
    <div class='cart flex-row'>
      <?php
      if (isset($_SESSION['role']) && $_SESSION['role']==1){ ?>
```

```
   <a href='panel'>
    <i class="material-icons" style="font-size:36px">person</i>
    <p>ADMIN PANEL</p>
   </a>

   <?php }else { ?>
    <a href='login'>
    <i class="material-icons" style="font-size:36px">person</i>
   </a>
   <?php } ?>
   <a href='account'>
    <i class="material-icons" style="font-size:36px">shopping_cart</i>
   </a>

   <?php if (isset($_SESSION['email'])):?>

   <a href='logout.php'><i class="material-icons" style="font-size:36px">close</i></a>



   <?php endif; ?>
  </div>
 </header>
 <main>
```

views/about.template.php

```
<div class='author'>
 <img src='assets/img/other/ja.jpg' width="300px" height="300px">
 <p>
```

Никола Митровић, рођен 14. јуна 1995. године. Завршио машинско-електротехничку школу, смер Електротехничар рачунара у Смедеревској Паланци. Тренутно студент Високе ИЦТ школе,смер Интернет Технологије у Београду.

```
</p>

</div>
```

views/account.template.php

```
<section id='items'>

  <h3> This is your cart! Get back to <a href="products?category=1">shopping!</a></h3>

<?php


  // if (!isset($_SESSION['email'])){

  //   Header("Location:index");

  // }

  while ($row = mysqli_fetch_object($result)) :?>

    <div class='item flex-row'>

     <div class='item-left'>

       <img src='<?=$row->src?>' alt='<?=$row->alt?>'>

     </div>

     <div class='item-mid'><?=$row->name?><br/>

       $<?=$row->price?>

     </div>

     <div class='item-right'>

     <button class='delete' value='<?=$row->id?>'>Remove</button>

     </div>

   </div>




  <?php endwhile; ?>

</section>
```

views/details.template.php

```
<div class='flex-row product-details'>
  <div class='product-image'>


    <?= "<img src='/".$product->src."' alt='".$product->alt."'>";?>
      <div class="rating">
        <?php if (!isset($_SESSION['email'])){
          echo "Please login to vote";
        }else {
        ?>
        <button class='vote' id='<?=$product->id?>' value='5'>☆</button>
        <button class='vote' id='<?=$product->id?>' value='4'>☆</button>
        <button class='vote' id='<?=$product->id?>' value='3'>☆</button>
        <button class='vote' id='<?=$product->id?>' value='2'>☆</button>
        <button class='vote' id='<?=$product->id?>' value='1'>☆</button>
        <?php }  ?>
      </div>
      <p>Users rating:&nbsp;&nbsp;<span id='user-rating'><?=$ratings->rating?></span>&nbsp;/&nbsp;5.0</p><br/><span id='denied'></span>
  </div>



  <div class='product-description'>
    <h1>Description:</h1>
    <?= "<p>".$product->productdescription."</p>";?>
  </div>


  <div class='product-right'>
```

```
    <?= "<h1>" .$product->productname."</h1>";?>


      <hr>
      <form method='POST' action='/account' id='add'>
        <?= "<h1>$".$product->productprice."</h1>";?>
          <input type="number" min='1' max='5' value='1' name='quantity'>
          <br/>
          <input type='hidden' name='price' id='price' value='<?=$product->productprice?>' readonly>
          <input type='hidden' name='productid' id='productid' value='<?=$product->id?>' readonly>
          <?php if (!isset($_SESSION['userid'])){ ?>
          <a href='login'>Login to add to cart</a>
          <?php ;} else {?>
          <button name='add'>Add to cart</button>
          <?php } ?>
      </form>
  </div>
</div>


<div class="flex-row more">
  <?php if(!isset($_SESSION['userid'])){?>


  <a href="/login">Login to leave a review</a>


  <?php } else {?>
  <form action='/products' method="POST" class='review'>
    <label for='heading'>Heading:</label>
    <br/>
    <input type='text' id='heading' name='heading' required/>
    <br/>
    <label for='content'>Review:</label>
    <br/>
    <textarea rows="4" cols="50" name='content' id='content' required></textarea>
    <br/>
    <input type='hidden' name='productid' value='<?=$product->id?>' readonly>
    <button type='submit' name='review'>Send</button>
  </form>
  <?php } ?>
  <div class='comments'>


    <?php while ($comm =  mysqli_fetch_object($comment)):?>
    <div class='comment'>
      <span>
        <?=$comm->first_name ." ". $comm->last_name ." ". $comm->date_created?>
      </span>
      <p>
        <?=$comm->heading?>
      </p>
      <p>
        <?=$comm->content?>
      </p>
    </div>
    <?php endwhile; ?>


  </div>
</div>
```

views/home.template.php

```
<section class='cover'>
    <div class='hero'>
```

```
        <h1>Conquer Every<br/> Trail</h1>
        <p>
            Grab a pair of our socks to help you keep going step<br/> after step! Every pair of toesocks provides blister<br/> protection, total foot utilization and moisture<br/> management.
        </p>
        <a href='products?category=1'><button class='shop-now-btn'>Shop now</button></a>
      </div>
    </section>
    <section class='arrivals flex-row'>
      <?php while ($r = mysqli_fetch_object($row)) :?>
      <div class='arrival'>
      <a href="details?productid=<?=$r->productid?>"><img src='<?=$r->src?>' alt='<?=$r->alt?>'></a>
</div>


      <?php endwhile;?>
    </section>
    <section class='trail'>
      <div class='trail-bg'>
      </div>
      <div class='trail-socks' id='trailSocks'>
      </div>
      <div class='trail-text' id='trailText'>
          <h1 class='trail-heading'>To the trail</h1>
          <p>
            Tackle the trails blister-free while keeping dirt and debris out so you can keep going mile after mile.
          </p>
          <a href='products?category=1'><button class='shop-now-btn'>Shop now</button></a>
    </div>
    </section>
```

```
  <section class='mountain'>
   <div class='mountain-bg'>
   </div>
   <div class='mountain-socks' id='mountainSocks'>
   </div>
   <div class="mnt-text" id='mountainText'>
     <h1>To the mountain</h1>
     <p>
      Support for every adventure, big or small—not to<br/> mention protection from unpredictable elements and terrain.
     </p>
     <a href='products?category=2'><button class='shop-now-btn'>Shop now</button></a>
    </div>
  </section>
  <section class='road'>
   <div class='road-bg'>
   </div>
   <div class='road-socks'  id='roadSocks'>
   </div>
   <div class='trail-text' id='roadText'>
    <h1 class='road-heading'>To the road</h1>
    <p>
     Take your run to the next level with our patented five-toe advantage and experience blister-free comfort.
    </p>
    <a href='products?category=3'><button class='shop-now-btn'>Shop now</button></a>
   </div>
  </section>
  <section class='shop-now flex-row'>
   <div class="shop-now-text">
```

```
        <h2>Compare product lines</h2>
        <h1>Find Your Sock</h1>
        <a href='products?category=1'><button class='btn-white'>Shop now</button></a>
      </div>
    </section>
    <section class='benefits'>
      <div class='left'>
        <h1>Discover the benefits</h1>
        <div class='benefits-buttons flex-row'>
          <div class='benefit-btn util-btn dark' id='util'>
          </div>
          <div class='benefit-btn blister-btn' id='blister'>
          </div>
          <div class='benefit-btn manage-btn' id='manage'>
          </div>
        </div>
      </div>
      <div class='benefit-bg util fade'>
        <div class='description'>
          <h1>Total Foot Utilization</h1>
          <p>
            Injinji toesocks keep your toes separated, supporting proper alignment and toe splay. This distributes your weight evenly on your foot, keeping it fully engaged in any athletic activity.
          </p>
        </div>
      </div>
     <div class='benefit-bg blister fade'>
        <div class='description'>
          <h1>Blister Prevention</h1>
          <p>
```

```
           With each toe protected, skin-on-skin friction is completely eliminated, making those painful blisters and hot spots a thing of the past.
          </p>
        </div>
     </div>
     <div class='benefit-bg manage fade'>
        <div class='description'>
            <h1>Moisture Management</h1>
            <p>
              Keeping your toes individually wrapped in sweat-wicking material means your toes and feet are drier and therefore more comfortable than in a traditional sock.
            </p>
          </div>
    </div>
    </section>
```

views/login.template.php

```
<?php


if (isset($_SESSION['cart'])){


  Header("location:/account");


}


?>
<div class='login'>
  <h1> Customer Login</h1>
  <h3>Registered Customers</h3>
  <h5>If you have an account, sign in with your email address.</h5>
```

```
    <form method='POST' action="/index">
      <input type='text' name='email' id='email' placeholder='Enter your name'/><br/>
      <input type='password' name='password' id='password' placeholder="Enter your password"/><br/>
      <button type="submit" id="register" class='register-btn' name='login'>Sign In</button>
    </form>
</div>
<div class='new-customer'>
  <h2>New customer</h2>
  <p>Creating an account has many benefits: check out faster, keep more than one address, track orders and more.</p>

  <a href="/register"><button class='register-btn login-btn'>Create an account</button></a>
</div>
```

views/product.template.php

```
<?php

// $categoryid = $_GET['category'] ?? 1;

// $get_products = "SELECT * FROM product p JOIN category c ON p.categoryid = c.categoryid JOIN image i ON p.imageid = i.imageid WHERE p.categoryid = ".$categoryid;

//$products = mysqli_query($conn, $get_products);
?>
<div class='shop-cover'>

</div>
<section class='products-content flex-row'>
  <div class='filter'>
    <h5>Filter <br/> options</h5>
    <hr>
    <p>Gender</p>
  <form action="#">
  <p>
    <input type="radio" id="unisex" value='1' name="radio-filter">
    <label for="unisex">Unisex</label>
  </p>
  <p>
    <input type="radio" value='2' id="female" name="radio-filter"  checked>
    <label for="female" >Female</label>
  </p>
  <p>
    <input type="radio" value='3' id="male" name="radio-filter">
    <label for="male">Male</label>
  </p>
</form>
  </div>
  <div class='products'>
    <?php  while ($row = mysqli_fetch_object($result)) : ?>

      <div class='product'>
        <a href='details?productid=<?=$row->productid?>'>
          <img src='/<?=$row->src?>' alt='<?=$row->alt?>'>
          <div class='product-name'>
            <?=$row->name ?>
          </div>
        </a>
        <p>$<?=$row->price?></p>
      </div>
      <?php endwhile; ?>
  </div>
</section>
```

views/register.template.php

```
<section class="register">
  <h1>Create New Customer Account</h1>
  <form id='insert' method='POST' action="/register">
    <p>First name: </p>
    <input type='text' name='firstName' id='firstName' onBlur="first();"/><br/>
    <p>Last name: </p>
    <input type='text' name='lastName' id='lastName' onBlur="last();"/><br/>
    <p>Password: </p>
    <input type='password' name='password' id='password' required/><br/>
    <p>E-mail: </p>
    <input type='text' name='email' id='email' onBlur="mail();"/><br/>
    <button type="submit" id="register" class='register-btn' name='register'>Register</button>
  </form>
</section>
```

views/admin/admin.template.php

```
<div style="text-align:center">
    <span onclick="showAdminTable(1)" class='show-table'>User table</span>
    <span onclick="showAdminTable(3)" class='show-table'>Product table</span>
    <span onclick="showAdminTable(4)" class='show-table'>Image table</span>
    <span onclick="showAdminTable(5)" class='show-table'>Comment table</span>
    <span onclick="showAdminTable(2)" class='show-table'>Cart table</span>
  </div>
```

```
<?php


    require_once view('admin/user');
    require_once view('admin/cart');
    require_once view('admin/product');
    require_once view('admin/image');
    require_once view('admin/comment');
```

views/admin/cart.template.php

views/admin/comment.template.php

views/admin/image.template.php

views/admin/cart.template.php

Углавном је исто осим што се мењају ID и Name атрибути

```
<!-- DELETE -->
<div class='admin-table'>
  <h1>Cart table</h1>
  <div class='flex-row'>
  <form action="cart" method="POST">
    <table>
      <tr>
        <th colspan='3'>Delete cart</th>
      </tr>
      <tr>
        <th>CartID</th>
        <th>UserID</th>
        <th>Delete</th>
      </tr>
      <?php foreach ($cart as $row) :?>
      <tr>
        <td>
          <?=$row->cartid?>
        </td>
        <td>
          <?=$row->userid?>
        </td>
        <td>
          <input type='checkbox' name='for-delete-cart[]' value='<?=$row->cartid?>'>
        </td>
      </tr>
      <?php endforeach;?>
      <tr>
        <td colspan='3'>
          <button type='submit' name='cart-delete'>Delete</button>
        </td>
      </tr>
    </table>
  </form>

  <!-- EDIT -->
  <form action="cart" method="POST">

    <table>
      <tr>
        <th colspan='3'>Edit cart</th>
      </tr>
      <tr>
        <th>ID</th>
```

```
   <th>CartID</th>
   <th>UserID</th>
  </tr>
  <tr>
   <td>
    <select id='update-cart' name='cartid'>
     <option value='0'>Pick CartID</option>
     <?php foreach ($cart as $row):?>
     <option value='<?=$row->cartid?>'>
      <?=$row->cartid?>
     </option>
     <?php endforeach;?>
    </select>
   </td>
   <td>
    <input type='text' name='update-cartid' id='update-cartid' required readonly>
   </td>
   <td>
    <input type='text' name='update-userid' id='update-userid' required>
   </td>
   <td>
    <button type='submit' name='cart-update'>Update</button>
   </td>
  </tr>
 </table>

</form>

<!-- ADD -->
<form action="cart" method="POST">
```

```
    <table>
      <tr>
        <th colspan='2'>Add cart</th>
      </tr>
      <tr>
        <th>CartID</th>
        <th>UserID</th>
      </tr>
      <tr>
        <td>
          <input type='text' name='cartid' required>
        </td>
        <td>
          <input type='text' name='userid' required>
        </td>
        <td>
          <button type='submit' name='cart-insert'>Add</button>
        </td>
      </tr>
    </table>
  </form>
</div>
</div>
```

assets/js/script.js

```
// showAdminTable(1);


// AJAX FOR DELETING ITEMS FROM CART
$(document).ready(function () {
```

```
$('#items').on('click', '.delete', function () {
  var productid = this.value;
  if (confirm("Do you want to remove this item from your cart?")) {
    $.ajax({
      type: "POST",
      url: 'ajax/item_remove.php',
      data: {
        productid: productid
      },
      success: function (response) {
        //
        var result = JSON.parse(response);
        var output = "";
        if (result.length > 0) {
          $.each(result, function (key, value) {
            output += `<div class='item flex-row'>
            <div class='item-left'>
              <img src='${value['src']}' alt='${value['alt']}'>
            </div>
            <div class='item-mid'>${value['name']}<br/>
              $${value['price']}
            </div>
            <div class='item-right'>
            <button class='delete' value='${value['id']}'>Remove</button>
            </div>
          </div>`;
            $("#items").html(output);
          })
        } else {
            output = "You have no items in your cart";
            $("#items").html(output);
          }

        }
      })
    }
  })
})
//AJAX CART END

//AJAX FOR COMMENT EDIT
$("#update-comment").on('change', function(){
  var id = this.id;
  var value = this.value;
  $.ajax({
    url: 'ajax/comment_update.php',
    type: 'POST',
    data: {
      id:id,
      value: value
    },
    success: function(response){
      var data = JSON.parse(response);
      $("#update-heading").val(data[0]['heading']);
      $("#update-content").val(data[0]['content']);
      $("#update-date").val(data[0]['date_created']);
      $("#update-c-userid").val(data[0]['userid']);
      $("#update-c-productid").val(data[0]['productid']);
    }
  })
})
//END

//AJAX FOR RATING
$(".vote").on('click', function(){
  var value = this.value;
  var productid = this.id;
  $.ajax({
    url: "ajax/rating.php",
    type: "POST",
    data : {
      value: value,
      productid: productid
    },
    success: function(response){
      var data = JSON.parse(response);
      if (jQuery.type(data)=='string'){
        $("#denied").html(data);
      }
      else {
        $("#user-rating").html(data['rating']);
      }

    }
  })
})
//END

//AJAX FOR GENDER FILTER
```

```
$('[name=radio-filter]').on('click', function () {
  var gender = $(this).val();
  $.ajax({
    type: "POST",
    url: "ajax/gender.php",
    data: {
      gender: gender
    },
    success: function (response) {

      var products = JSON.parse(response);
      $(".products").empty();
      var text = "";
      if (products.length < 1) {
        text = `<h1>No products found for selected gender</h1>`;
        $(".products").append(text);
      }
      $.each(products, function (key, value) {

        text = `<div class='product'><a href='details?productid=${products[key]['productid']}'><img src='/${products[key]['src']}' alt='${products[key]['alt']}'><div class='product-name'>${products[key]['name']}</div></a><p>$${products[key]['price']}</p></div>`;
        $(".products").append(text);
      })
    }
  })
})
// END

//EDIT CART AJAX//
```

```
$("#update-cart").on("change", function () {
  var cartid = this.value;
  $.ajax({
    url: "ajax/cart_update.php",
    data: {
      cartid: cartid
    },
    type: "POST",
    success: function (response) {

      var result = JSON.parse(response);

      $("#update-userid").val(result[0]['userid']);
      $("#update-cartid").val(result[0]['cartid']);
    }
  })
})

//END

//EDIT USER AJAX
$("#update-user").on('change', function () {
  var userid = this.value;
  $.ajax({
    url: 'ajax/user_update.php',
    type: "POST",
    data: {
      userid: userid
    },
    success: function (response) {
      var result = JSON.parse(response);
      $("#update-first").val(result[0]['first_name']);
      $("#update-last").val(result[0]['last_name']);
      $("#update-email").val(result[0]['email']);
      $("#update-password").val(result[0]['password']);
      $("#update-roleid").val(result[0]['roleid']);
    }
  })

})

//END

//EDIT PRODUCT AJAX
$("#update-product").on('change', function () {
  var productid = this.value;
  $.ajax({
    url: "ajax/product_update.php",
    method: "POST",
    data: {
      productid: productid
    },
    success: function (response) {
      var result = JSON.parse(response);
      $("#update-name").val(result[0]['name']);
      $("#update-desc").val(result[0]['description']);
      $("#update-price").val(result[0]['price']);
      $("#update-cat").val(result[0]['categoryid']);
      $("#update-gender").val(result[0]['genderid']);
      $("#update-image").val(result[0]['imageid']);
    }
  })
})

//END

//EDIT IMAGE AJAX
$("#update-images").on('change', function () {
  var imageid = this.value;
  $.ajax({
    url: "ajax/image_update.php",
    type: "POST",
    data: {
      imageid: imageid
    },
    success: function (response) {
      var result = JSON.parse(response);
      $("#update-src").val(result[0]['src']);
      $("#update-alt").val(result[0]['alt']);
    }
  })
})
//EDIT

//CHANGE BENETIFS ON CLICK
showBenefits(1);

function showBenefits(n) {

  let benefits = document.getElementsByClassName('benefit-bg');
  for (let i = 0; i < benefits.length; i++) {

    benefits[i].style.display = 'none';

  }
  benefits[n - 1].style.display = 'block';

}
//END

//ADD LISTENERS
addListeners();

function addListeners() {
  document.getElementById('util').addEventListener('click', function () {

    showBenefits(1);

    addDark(this, 'dark');

    removeManageAndBlister();

  });
  document.getElementById('blister').addEventListener('click', function () {

    showBenefits(2);

    addDark(this, 'dark1');
    removeUtilAndManage();

  });
  document.getElementById('manage').addEventListener('click', function () {

    showBenefits(3);

    addDark(this, 'dark2');

    removeUtilAndBlister();
  });

  window.addEventListener('scroll', function () {
    scroll();
  })
}
//ADD DARKENED BACKGROUNDS TO BENEFITS BUTTONS
function addDark(element, className) {

  let lista = element.classList;

  if (!lista.contains(className)) {

    element.classList.add(className);

  }
}
//END

//REMOVING DARK BACKGROUND FROM NON-CLICKED BUTTONS
```

```
function removeManageAndBlister() {

  let btnBlister = document.getElementById('blister');
  let btnManage = document.getElementById('manage');

  btnBlister.classList.remove('dark1');
  btnManage.classList.remove('dark2');
}

function removeUtilAndBlister() {

  let btnUtil = document.getElementById('util');
  let btnBlister = document.getElementById('blister');

  btnUtil.classList.remove('dark');
  btnBlister.classList.remove('dark1');
}

function removeUtilAndManage() {

  let btnUtil = document.getElementById('util');
  let btnManage = document.getElementById('manage');

  btnUtil.classList.remove('dark');
  btnManage.classList.remove('dark2');
}



//SLIDE IN / OUT WHEN SCROLLING
function scrollFirst(posY) {
  }

  if (posY > 850) {

    document.getElementById("trailText").classList.add('slideToRight');
    document.getElementById("trailSocks").classList.add('slideToLeft');

  }

}

function scrollSecond(posY) {

  if (posY > 1300) {

    document.getElementById("mountainText").classList.add('slideToLeft');
    document.getElementById("mountainSocks").classList.add('slideToRight');

  }
}

function scrollThird(posY) {

  if (posY > 1800) {

    document.getElementById("roadText").classList.add('slideToRight');
    document.getElementById("roadSocks").classList.add('slideToLeft');

  }
}
```

```
function scroll() {

  let posY = window.scrollY || window.scrollTop || document.getElementsByTagName("html")[0].scrollTop;

  scrollFirst(posY);
  scrollSecond(posY);
  scrollThird(posY);

}



function showAdminTable(n) {

  let tables = document.getElementsByClassName('admin-table');

  for (let i = 0; i < tables.length; i++) {

    tables[i].style.display = 'none';

  }
  tables[n - 1].style.display = 'block';

}


function first() {
  var regName = /^[A-Za-z]{4,15}$/;
  var first = document.getElementById('firstName');
  if(!regName.test(first)) {
    first.style.borderColor='red';
  }
  else {
    first.style.borderColor='green';
  }
}

function last() {
  var regName = /^[A-Za-z]{4,15}$/;
  var last = document.getElementById('lastName');
  if(!regName.test(last)) {
    last.style.borderColor='red';
  }
  else {
    last.style.borderColor='green';
  }
}

function mail(){
  var mail = document.getElementById('email');
  var regMail = /^\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,3}$/;
  if(!regMail.test(mail)) {
    mail.style.borderColor='red';
  }
  else {
    mail.style.borderColor='green';
  }
}
```

main.css

```
* {
  font-family: 'Roboto', sans-serif;
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

.flex-row {
  display: flex;
  justify-content: space-between;
}

header {
  box-shadow: 0 0 15px 0 #817e81;
  position: fixed;
  background-color: white;
  z-index: 2;
  width: 100%;
}

.logo {
  flex: 1;
  text-align: center;
}

nav {
  flex: 3;
}
```

```
nav ul {
  list-style: none;
}

nav ul li {
  display: inline-block;
  padding: 30px 50px;
}

nav a {
  text-decoration: none;
  color: #333;
  font-weight: bold;
  font-size: 20px;
}

.cart {
  flex: 1;
  text-align: center;
  padding: 30px 50px;
  justify-content: space-around;
}

.cart a {
  text-decoration: none;
  color: #333;
}

main {
  position: relative;
  top: 105px;
  min-height: 400px;
  background-color: #FFFFFF;
}

.author {
  min-height: 300px;
  padding: 8% 15%;
  color: #333;
  text-align: center;
}

.author img {
  max-width: 100%;
  height: auto;
}



footer {
  width: 100%;
  height: 50px;
  background-color: #333;
  color: white;
  text-align: center;
}

footer a {
  text-decoration: none;
  color: white;
  margin-right: 20px;
```

```
}

.cover {
  min-height: 691px;
  background-image: url('../img/other/cover.jpg');
  background-repeat: no-repeat;
  background-size: cover;
  background-position: 70% 50%;
  padding: 10%;
}

.hero {
  width: 80%;
  margin-left: 4%;
}

.hero h1 {
  font-size: 68px;
  font-weight: bold;
}

.hero p {
  margin-top: 40px;
  color: white;
  font-weight: bold;
  letter-spacing: 2px;
  font-size: 18px;
  line-height: 30px
}
```

```
.shop-now-btn {
  padding: 15px;
  position: absolute;
  border-radius: 25px;
  width: 130px;
  margin-top: 20px;
  border: 2px solid black;
  font-weight: bold;
  background-color: transparent;
  cursor: pointer;
  line-height: 25px;
}

.hero button:hover {
  background-color: black;
  color: white;
}

.arrivals {
  min-height: 500px;
  padding: 10%;
  /*background-color: rgb(241, 241, 241);*/
  background-color: white;
}

.arrivals a {
  text-decoration: none;
  color: #333;
}
```

```
}

.arrivals a:hover {
  opacity: 0.8;
}

.arrival {
  flex-basis: 20%;
}

.arrivals p {
  position: relative;
  left: -100px;
  width: 200px;
  transition: left 1s;
}

.trail {
  height: 500px;
  background-color: rgb(241, 241, 241);
}

.trail-bg {
  position: absolute;
  width: 100%;
  height: 500px;
  background-image: url('../img/sections/trail-bg.png');
  background-position: right;
  background-repeat: no-repeat;
}

.trail-socks {
  height: 500px;
  width: 50%;
  float: right;
  position: relative;
  background-image: url('../img/sections/trail-fg.png');
  background-repeat: no-repeat;
  background-position: right;
  opacity: 0;
  visibility: hidden;
}

.trail-text {
  max-width: 33%;
  margin-left: 13%;
  padding-top: 130px;
  opacity: 0;
  visibility: hidden;
}

.trail-text h1,
.mnt-text h1 {
  font-size: 48px;
  display: inline;
  color: #333;
}

.trail-heading::after {
  background-color: #fe5a7d;
  content: ";
  display: block;
  height: 2px;
  margin-top: 17px;
  width: 170px;
}

.trail-text p,
.mnt-text p {
  margin-top: 15px;
  letter-spacing: 2px;
  font-size: 15px;
  line-height: 30px;
  color: #333;
}

.trail-text button:hover {
  background-color: #fe5a7d;
  color: white;
  border-color: #fe5a7d;
  cursor: pointer;
}

.mountain {
  height: 500px;
  background-color: white;
}

.mountain-bg {
  width: 100%;
  height: 500px;
  position: absolute;
  background-image: url('../img/sections/mnt-bg.png');
  background-position: left;
  background-repeat: no-repeat;
}

.mountain-socks {
  width: 50%;
  height: 500px;
  position: relative;
  float: left;
  background-image: url('../img/sections/mnt-fg.png');
  background-repeat: no-repeat;
  background-position: left;
  opacity: 0;
  visibility: hidden;
}

.mnt-text {
  float: right;
  max-width: 33%;
  padding-top: 130px;
  margin-right: 13%;
  opacity: 0;
  visibility: hidden;
}

.mnt-text h1::after {
  background-color: #41cbb0;
  content: ";
  display: block;
  height: 2px;
  margin-top: 17px;
  width: 170px;
}

.mnt-text button:hover {
  background-color: #41cbb0;
  border-color: #41cbb0;
  color: white;
  cursor: pointer;
}

.road {
  height: 500px;
  background-color: rgb(241, 241, 241);
}

.road-bg {
  height: 500px;
  width: 100%;
  position: absolute;
  background-image: url('../img/sections/road-bg.png');
  background-repeat: no-repeat;
  background-position: right;
}

.road-socks {
  background-image: url('../img/sections/road-fg.png');
  position: relative;
  width: 50%;
  height: 500px;
  float: right;
  background-repeat: no-repeat;
  background-position: right;
  opacity: 0;
  visibility: hidden;
}

.road-heading::after {
  background-color: #FD8E00;
  content: '';
  display: block;
  height: 2px;
  margin-top: 17px;
  width: 170px;
}

.road-socks button:hover {
  background-color: #FD8E00;
  border-color: #FD8E00;
  color: white;
  cursor: pointer;
}

.shop-now {
  width: 100%;
  min-height: 400px;
  background-image: url('../img/sections/shop-now.jpg');
  background-position: center;
  background-repeat: no-repeat;
  color: white;
}

.shop-now-text {
  flex-basis: 33%;
  text-align: center;
  margin: auto;
}

.shop-now-text h1 {
  margin-top: 10px;
  font-size: 3em;
}

.btn-white {
  color: white;
  padding: 15px;
  border-radius: 25%;
  width: 50%;
  margin-top: 20px;
  border: 2px solid white;
  font-weight: bold;
  background-color: transparent;
  cursor: pointer;
}

.btn-white:hover {
  border-color: white;
  color: black;
  background-color: white;
```

```
}

.benefits {
  min-height: 700px;
  width: 100%;
  position: relative;
}

.left {
  position: absolute;
  color: #333;
  font-size: 25px;
  margin-top: 10%;
  margin-left: 12%;
  z-index: 1;
}

.benefits-buttons {
  min-height: 110px;
  width: 100%;
  position: absolute;
}

.benefit-btn {
  flex-basis: 30%;
  background-repeat: no-repeat;
  background-position: center;
  background-size: contain;
  cursor: pointer;
}
```

```
.util-btn {
  background-image: url('../img/benefits/util.png');
}

.blister-btn {
  background-image: url('../img/benefits/blister.png');
}

.manage-btn {
  background-image: url('../img/benefits/manage.png');
}

.dark {
  background-image: url('../img/benefits/util-bg.png');
}

.dark1 {
  background-image: url('../img/benefits/blister-bg.png');
}

.dark2 {
  background-image: url('../img/benefits/manage-bg.png');
}

.benefit-bg {
  background-position: right;
  background-size: contain;
  width: 100%;
  min-height: 700px;
  background-repeat: no-repeat;
  display: none;
  /*opacity: 0.5;
  /*transition: opacity 2s;
 /* animation-name: fade;
  animation-duration: 0.5s;*/
}

.description {
  margin-left: 12%;
  margin-top: 21%;
  width: 30%;
  line-height: 2;
}

.description h1 {
  position: relative;
  top: 20px;
  color: #333;
}

.description p {
  font-weight: bold;
  margin-top: 40px;
  font-size: 16px;
  color: #747779;
}

.util {
  background-image: url('../img/benefits/1.png');
  background-color: #DEEDF4;
  position: absolute;
}

.blister {
  background-image: url('../img/benefits/2.png');
  background-color: #D8C9C5;
  position: absolute;
}

.manage {
  background-image: url('../img/benefits/3.png');
  background-color: #EAECEA;
  position: absolute;
}

.shop-cover {
  min-height: 400px;
  background-image: url('../img/sections/shop-cover.jpg');
  background-repeat: no-repeat;
  background-size: cover;
  background-position: 70% 50%;
  padding: 10%;
}

.products-content {
  min-height: 800px;
  width: 100%;
  padding: 2%;
  background-color: #F6F7F8;
```

```
}

.filter {
  margin-top: 50px;
  flex: 1;
  background-color: white;
}

.filter p {
  margin: 50px;
  color: #333;
}

.filter h5 {
  margin: 40px;
  color: #333;
  font-size: 20px;
}

.filter hr {
  width: 90%;
}

[type="radio"]:checked,
[type="radio"]:not(:checked) {
  position: absolute;
  left: -9999px;
}

[type="radio"]:checked+label,
[type="radio"]:not(:checked)+label {
  position: relative;
  padding-left: 28px;
  cursor: pointer;
  line-height: 20px;
  display: inline-block;
  color: #666;
}

[type="radio"]:checked+label:before,
[type="radio"]:not(:checked)+label:before {
  content: ";
  position: absolute;
  left: 0;
  top: 0;
  width: 18px;
  height: 18px;
  border: 1px solid #ddd;
  border-radius: 100%;
  background: #fff;
}

[type="radio"]:checked+label:after,
[type="radio"]:not(:checked)+label:after {
  content: ";
  width: 12px;
  height: 12px;
  background: #333;
  position: absolute;
  top: 4px;
  left: 4px;
  border-radius: 100%;
  -webkit-transition: all 0.2s ease;
  transition: all 0.2s ease;
}

[type="radio"]:not(:checked)+label:after {
  opacity: 0;
  -webkit-transform: scale(0);
  transform: scale(0);
}

[type="radio"]:checked+label:after {
  opacity: 1;
  -webkit-transform: scale(1);
  transform: scale(1);
}

.products {
  text-align: center;
  flex: 5;
  margin-top: 50px;
}

.products h1 {
  margin-top: 200px;
}

.products img {
  max-width: 100%;
  height: auto;
}

.product {
  width: 33%;
  min-height: 350px;
  display: inline-block;
}

.product img:hover {
  opacity: 0.8;
}

.product a {
  text-decoration: none;
  color: #333;
}

.product a:hover {
  opacity: 0.6;
}

.product-name {
  width: 100%;
  position: relative;
  bottom: 10px;
}

.product-details {
  color: #333;
  min-height: 550px;
}

.product-image {
  /* position: relative;
  top: 50px; */
  flex: 1;
  text-align: center;
  height: 415px;
}

.product-image img {
  margin-top: 70px;
  max-width: 100%;
  height: auto;
}

.product-right {
  margin-top: 40px;
  /* position: relative;
  top: 10px; */
  flex: 1;
  min-height: 415px;
  text-align: center;
  font-size: 21px;
}

.product-right input{
  border: 1px solid black;
  height: 30px;
  width: 50%;
  border-radius: 25px;
  color: black;
  padding: 0 29px;
  margin: 10px auto;
}

.product-right h1 {
  width: 230px;
  margin: 0 auto;
}

.rating {
  direction: rtl;
  text-align: center;
}
.rating > button {
  display: inline-block;
  position: relative;
  width: 1.1em;
}
.rating > button:hover:before,
.rating > button:hover ~ button:before {
   content: "\2605";
   position: absolute;
}

.rating button {
  background-color: transparent;
  border: none;
  font-size: 50px;
  color: #2E8B57;
  cursor: pointer;
}

.product-right a{
  font-size: 28px;
  height: 50px;
  flex: 1;
  margin: 100px;
  text-decoration: none;
  font-style: italic;
  color: #337AC0;
}

.product-right button {
  border: none;
  color: white;
  text-decoration: none;
  width: 40%;
  margin-top: 10px;
  height: 80px;
  font-size: 34px;
  background-color: #e31e26;
  cursor: pointer;
}

.product-right button:hover {
  opacity: 0.8;
}
```

```
.more {
  background-color: #F6F7F8;
  color: #333;
  justify-content: space-around;
}

.product-description {
  flex: 1;
  text-align: left;
  /* margin: 0 auto; */
  padding-top: 100px;
  line-height: 2em;
  color: #333;
}

.product-description h1 {
  margin: 20px 0px 30px 0;
}

.more input {
  border: 1px solid #c7c7c7;
  height: 50px;
  width: 50%;
  color: #798189;
  padding: 0 20px;
  margin: 10px auto;
}

.more textarea {
  border: 1px solid #c7c7c7;
  width: 50%;
  color: #798189;
  padding: 0 20px;
  margin: 10px auto;
}

.review {
  flex: 1;
  text-align: center;
  padding: 20px;
}

.review button {
  border: none;
  color: white;
  text-decoration: none;
  width: 20%;
  margin-top: 10px;
  height: 40px;
  font-size: 25px;
  background-color: #e31e26;
  cursor: pointer;
}

.comments {
  flex: 1;
  padding: 20px;
}
```

```
.more a {
  font-size: 36px;
  height: 50px;
  flex: 1;
  margin: 100px;
  text-decoration: none;
  font-style: italic;
  color: #337AC0;
}

.comment {
  width: 400px;
  min-height: 200px;
  margin: auto;
  margin-bottom: 10px;
  background-color: #F6E6FF;
  padding: 20px;
}

.comment span {
  color: seagreen;
}

.comment p:first-of-type {
  font-style: italic;
}

.comment p {
  margin-top: 10px
}
```

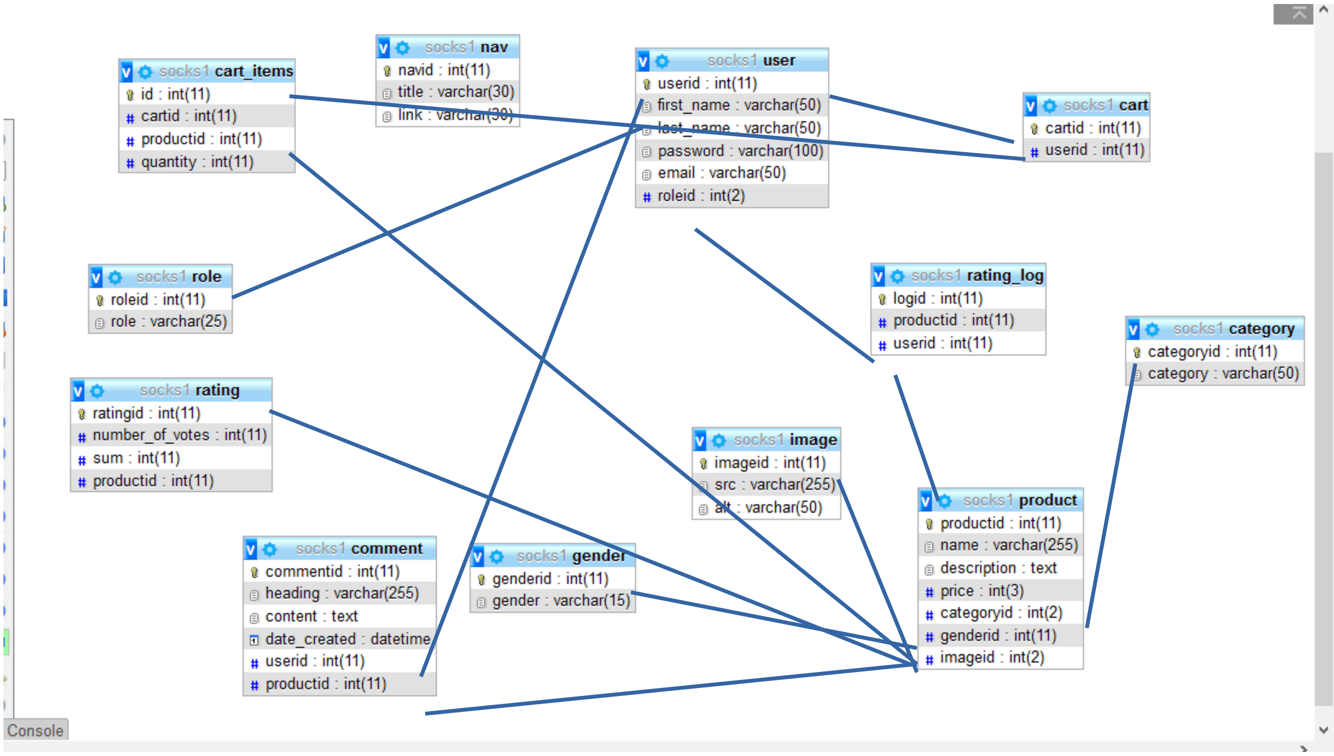
```
#items {
  padding: 10%;
  min-height: 700px;
  width: 100%;
  background-color: #F6F7F8;
}

.item:first-of-type {
  border-top: 1px solid black;
}

.item {
  width: 60%;
  height: 80px;
  margin: 0 auto;
  border-bottom: 1px solid black;
}

.cart-buttons {
  min-height: 700px;
  width: 50%;
  margin: 0 auto;
  background-color: #F6F7F8;
  text-align: center;
}

.cart-buttons button {
  border: none;
  background-color: royalblue;
  cursor: pointer;
  width: 150px;
  height: 35px;
}

.item-left {
  flex: 1;
  height: 100%;
}

.item-mid {
  height: 100%;
  flex: 2;
  margin-top: 40%;
  text-align: center;
}

.item-right {
  flex: 1;
  height: 100%;
  text-align: center;
  margin-top: 30%;
}

.item-right button {
  border: 1px solid black;
  height: 30px;
  width: 50%;
  border-radius: 25px;
  background-color: white;
  margin: auto;
  cursor: pointer;
}

.item img {
  max-width: 100%;
  height: 100%;
}

.product-right hr {
  width: 50%;
  margin: 20px auto;
}

.register {
  min-height: 700px;
  width: 100%;
  padding-top: 5%;
  background-color: #F6F7F8;
  text-align: center;
}

.register p::after {
  content: "*";
  color: red;
}

.register h1 {
  font-size: 48px;
  margin-bottom: 20px;
```

```
}

#insert {
  width: 60%;
  margin: 0 auto;
}

.register input,
.login input {
  border: 1px solid #c7c7c7;
  height: 50px;
  width: 80%;
  border-radius: 25px;
  color: #798189;
  padding: 0 29px;
  margin: 10px auto;
}

.login {
  color: #3b3b3b;
  width: 50%;
  min-height: 500px;
  float: left;
  text-align: center;
  padding: 5%;
  background-color: #F6F7F8;
}

.new-customer {
  width: 50%;
  min-height: 500px;
  padding: 10%;
  float: right;
  background-color: #F6F7F8;
  color: #3b3b3b;
}

.new-customer p {
  margin-top: 20px;
  margin-bottom: 20px;
  font-size: 12px;
}

.register-btn:hover {
  color: white;
  background-color: black;
}

.register-btn {
  border: 1px solid black;
  height: 50px;
  width: 50%;
  border-radius: 25px;
  background-color: white;
  padding: 0 29px;
  margin: 10px auto;
}

.login-btn {
  margin-top: 20px;
```

```
}

.login h1 {
  font-size: 48px;
  margin: 0px;
}

.login h3 {
  margin: 10px;
}

.login h5 {
  margin: 10px;
}

.register-btn {
  font-weight: bold;
  cursor: pointer;
  color: black;
}

.admin-table {
  display: none;
  height: auto;
  margin: 0 auto;
  text-align: center;
}

.admin-table table {
  padding: 30px;
```

```
}

.admin-table button {
  height: 50px;
  width: 100px;
  background-color: transparent;
  cursor: pointer;
}

.admin-table h1 {
  width: 400px;
  margin: 0 auto;
  text-align: center;
  padding-top: 80px;
}

.admin-table input {
  border: 1px solid black;
  height: 30px;
  width: 80%;
  border-radius: 25px;
  color: black;
  padding: 0 29px;
  margin: 10px auto;
}

.show-table {
  border: none;
  position: relative;
  top: 70px;
  margin: 0px 25px;
  font-size: 15px;
  background-color: #337AC0;
  cursor: pointer;
}

table {
  text-align: center;
}

td {
  border-bottom: 1px solid #337AC0;
  min-width: 100px;
}

th {
  background-color: #337AC0;
  color: white;
}

tr {
  height: 50px;
}

tr:hover {
  background-color: #f5f5f5;
}

table img {
  max-width: 10%;
  height: 10%;
}

/*ANIMATIONS*/

/*BENEFITS FADE*/

.fade {
  animation-name: fade;
  animation-duration: 0.5s;
  opacity: 1;
}

@keyframes fade {
  from {
    opacity: 0;
  }
  to {
    opacity: 1;
  }
}

/*BENEFITS END*/

/*SLIDE RIGHT*/

.slideToRight {
  animation-name: slideToRight;
  animation-duration: 1s;
  opacity: 1;
  visibility: visible;
}

@keyframes slideToRight {
  0% {
    opacity: 0;
    -webkit-transform: translateX(-70%);
  }
  100% {
    opacity: 1;
    -webkit-transform: translateX(0%);
  }
}

/*SLIDE RIGHT END*/

/*SLIDE TO LEFT*/

.slideToLeft {
  animation-name: slideToLeft;
  animation-duration: 1s;
  opacity: 1;
  visibility: visible;
}

@keyframes slideToLeft {
  0% {
    opacity: 0;
    -webkit-transform: translateX(70%);
  }
  100% {
    opacity: 1;
    -webkit-transform: translateX(0%);
  }
}

/*SLIDE LEFT END*/
```

4. Дизајн базе



socks1 cart_items
- id : int(11)
- cartid : int(11)
- productid : int(11)
- quantity : int(11)

socks1 nav
- navid : int(11)
- title : varchar(30)
- link : varchar(30)

socks1 user
- userid : int(11)
- first_name : varchar(50)
- last_name : varchar(50)
- password : varchar(100)
- email : varchar(50)
- roleid : int(2)

socks1 cart
- cartid : int(11)
- userid : int(11)

socks1 role
- roleid : int(11)
- role : varchar(25)

socks1 rating_log
- logid : int(11)
- productid : int(11)
- userid : int(11)

socks1 category
- categoryid : int(11)
- category : varchar(50)

socks1 rating
- ratingid : int(11)
- number_of_votes : int(11)
- sum : int(11)
- productid : int(11)

socks1 image
- imageid : int(11)
- src : varchar(255)
- alt : varchar(50)

socks1 product
- productid : int(11)
- name : varchar(255)
- description : text
- price : int(3)
- categoryid : int(2)
- genderid : int(11)
- imageid : int(2)

socks1 comment
- commentid : int(11)
- heading : varchar(255)
- content : text
- date_created : datetime
- userid : int(11)
- productid : int(11)

socks1 gender
- genderid : int(11)
- gender : varchar(15)