

Visoka škola strukovnih studija za
informacione i komunikacione
tehnologije

COMP-SHOPPER

<https://nemanjajezdic.github.io/Compshop/>

Smer: Internet tehnologije
Modul: Web programiranje
Predmet: Web programiranje 2

Nemanja Jezdic 30/19

Sadržaj

1. Uvod

1.1 Korisćene tehnologije

1.2 Opis funkcionalnosti

1.3 Template

2. Organizacija

2.1 Organizacija sajta

2.2 Opis strane sajta

3. Kodovi

3.1 HTML

3.2 CSS

3.3 JavaScript

1 . Uvod

1.1 Korisćene tehnologije

- HTML 5
-
- CSS 3
- Bootstrap
- JavaScript
- jQuery
- Xampp
- Visual Studio code

1.2 Opis funkcionalnosti

- Dinaičko ispisivanje proizvoda.
- Dinaičko ispisivanje kategorija.
- Dinamičko ispisivanje proizvoda u slajderu.
- Dinamičko ispisivanje gresaka u validaciji.
- Dinamičko ispisivanje u cart-u.
- Celokupna funkcionalnost carta(Dodavanje u cart,Brisanje iz cart-a,renderovanje proizvoda-povecanje I smanjene broja proizvoda,kao I racunjane cene na svaku promenu).
- Dohvatanje JSON fajlova AJAX-om.
- Koriscenje Local Storage za cuvanje podataka.
- Filtriranje po kategorijama.
- Filtriranje po ceni.
- Sortiranje po imenima i ceni po opadajucem I rastucem poretku.
- Validacija forme i ispis gresaka radjeno regexima.

1.3 Template

Korisćen je gotov css template <https://www.free-css.com/free-css-templates/page203/e-shopper>.

2 . Organizacija

2.1 Organizacija sajta

Index.html<->shop.html<->contact-us.html<->checkout.html<->author.html(Svaka stranica vodi na svaku stranicu).

2.2 Mapa sajta

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
    http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">

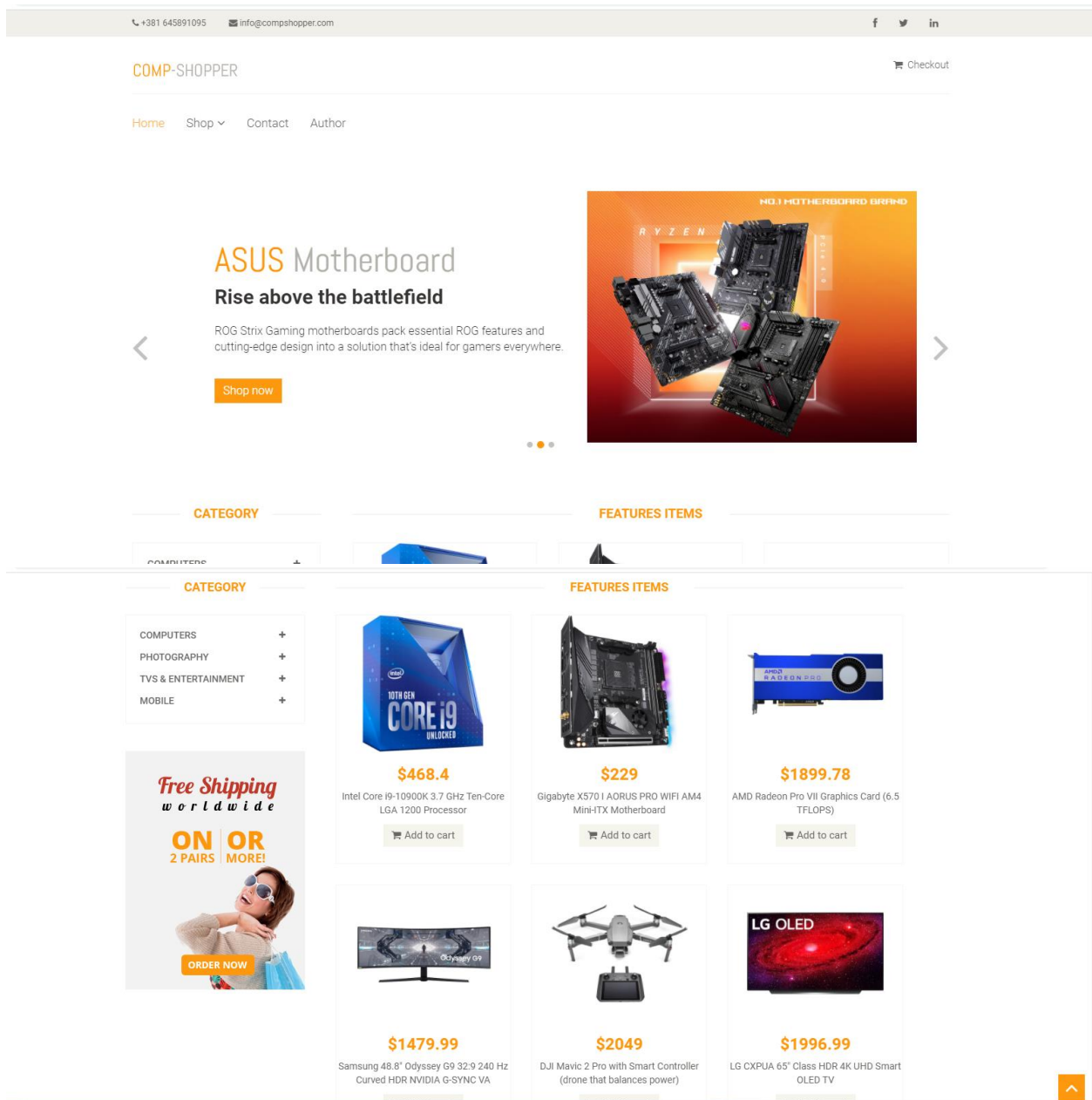
  <url>
    <loc>https://nemanjajezdic.github.io/Treemoon/</loc>
    <lastmod>2020-12-11</lastmod>
    <changefreq>daily</changefreq>
    <priority>1.00</priority>
  </url>
  <url>
    <loc>https://nemanjajezdic.github.io/Treemoon/index.html</loc>
    <lastmod>2020-12-11</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.80</priority>
  </url>

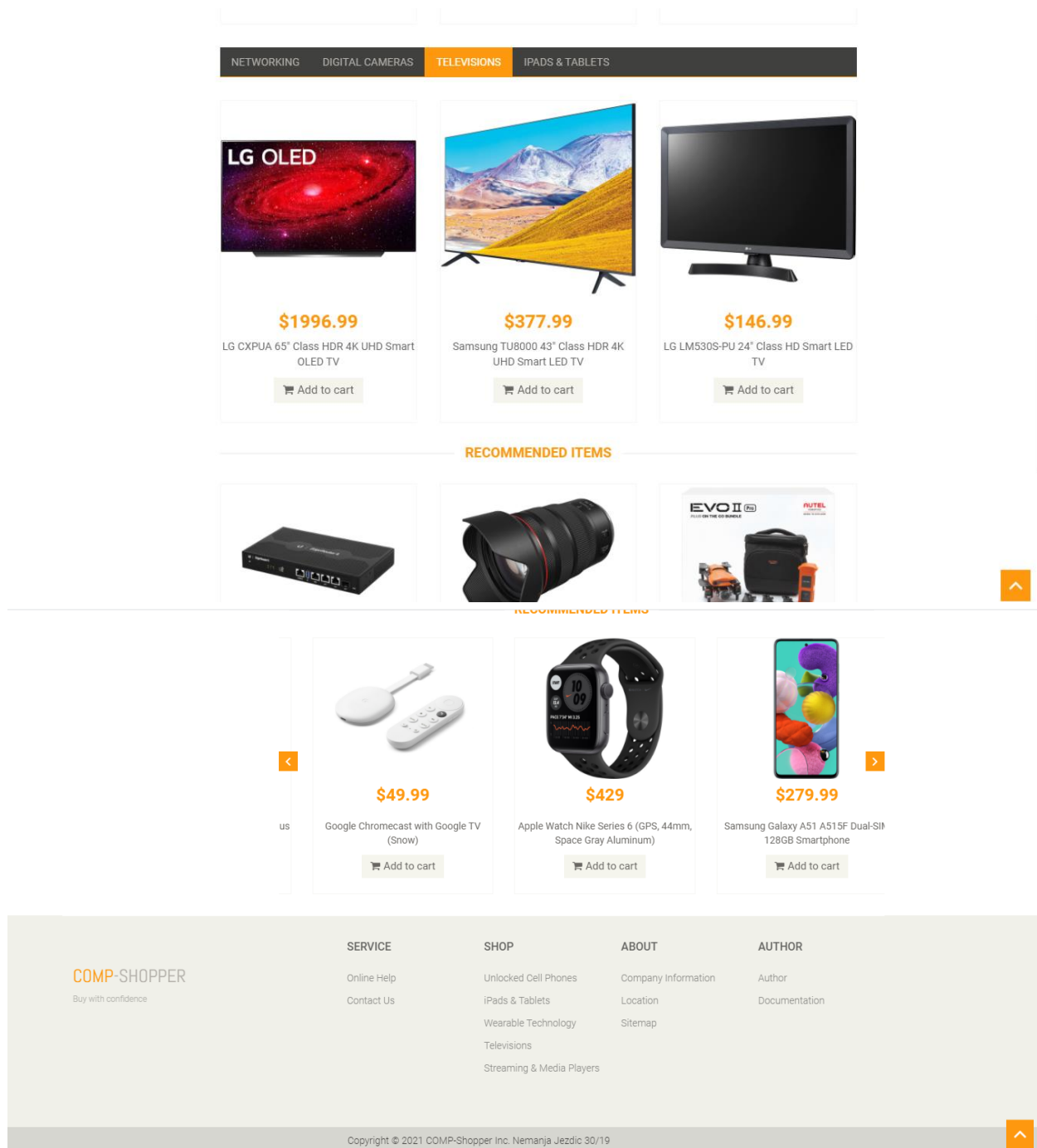
</urlset>
```

2.3 Opis strane sajta

Index.html

Pocetna stranica koja vodi ka drugim stranicama. Na glavnoj strani su ucitani najprodavaniji proizvodi, svaki proizvod pripada odredjenoj kategoriji. Ako se izabere odredjena kategorija, vrsi se redirekcija na stranicu shop.html i vrsi se filtriranje po toj kategoriji. Ako korisnik izabere opciju Add to cart na zeljeni proizvod, isti ce biti ubacen u cart, biranjem svakog sledeceg proizvoda puni se cart, ako se izabere isti proizvod povecava se kolicina tog proizvoda u cart-u.





shop.html

Stranica za kupovinu proizvoda. Ispisani proizvodi u zavisnosti od nacina filtriranja I sortiranja. Odabirom odredjene kategorije vrši se filtriranje proizvoda po toj kategoriji. Odabirom odredjenog opsega cena vrši se filtriranje proizvoda čije cene spadaju u taj opseg. Odabirom opcija ASC ili DESC vrši se sortiranje proizvoda u rastućem ili opadajućem poretka, respektivno po ceni ili po imenu proizvoda. Odabir na koji način se vrši sortiranje je preko radio button-a ASC I DESC, a po čemu na osnovu opadajuće

liste. Odabirom taster Add to cart vrši se ubacivanje zadatog proizvoda u cart (Isti process opisan u index.html).

+381 645891095info@compshopper.com

f t in

COMP-SHOPPER

Checkout

HomeShop▼ContactAuthor

CATEGORY

COMPUTERS +

PHOTOGRAPHY +

TVS & ENTERTAINMENT +

MOBILE +

PRICE RANGE


\$0\$10000

SORT BY

Select..

☒ ASC☐ DESC


SHOP



\$39.95

Crucial 8GB Laptop DDR4 2666 MHz SODIMM Memory Module (1 x 8GB)


Add to cart



\$86.99

Crucial 16GB Ballistix DDR4 3600 MHz UDIMM Gaming Desktop Memory Kit


Add to cart



\$468.4

Intel Core i9-10900K 3.7 GHz Ten-Core LGA 1200 Processor


Add to cart



\$339.99

AMD Ryzen 7 3700X 3.6 GHz Eight-Core AM4 Processor


Add to cart



\$229

Gigabyte X570 I AORUS PRO WIFI AM4 Mini-ITX Motherboard

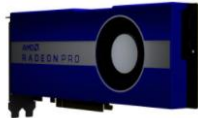
Add to cart



\$169.99

MSI Z490-A PRO LGA 1200 ATX Motherboard


Add to cart



\$899.32

AMD Radeon Pro W5700 Graphics Card


Add to cart



\$1899.78

AMD Radeon Pro VII Graphics Card (6.5 TFLOPS)


Add to cart



\$899.32

AMD Radeon Pro W5700 Graphics Card


Add to cart



\$1899.78

AMD Radeon Pro VII Graphics Card (6.5 TFLOPS)

Add to cart



COMP-SHOPPER

Buy with confidence

SERVICE

Online Help

Contact Us

SHOP

Unlocked Cell Phones

iPads & Tablets

Wearable Technology

Televisions

Streaming & Media Players

ABOUT

Company Information

Location

Sitemap

AUTHOR

Author

Documentation

Copyright © 2021 COMP-Shopper Inc. Nemanja Jezdic 30/19

CATEGORY

COMPUTERS +

PHOTOGRAPHY +

TVS & ENTERTAINMENT +

MOBILE +


PRICE RANGE

\$ 0 \$ 10000

SORT BY

Price


ASC DESC



\$230.26

Samsung 2TB T5 Portable Solid-State Drive (Black)


Add to cart



\$139.99

Samsung 1TB 970 EVO NVMe M.2 Internal SSD

Add to cart



\$28.99

SanDisk 256 GB Ultra USB 3.0 Flash Drive

Add to cart

contact-us.html

Stranica za kontaktiranje shop-a sa pratecim podacima za lokaciju I socijalne mrze.Da bi se potvrdila poruka,potrebno je popuniti formu koja je validirana regex-ima,u suportonom poruka nece biti poslata. Polje Name je obavezno,sadrzi ime korisnika I mora da pocne Velikim slovom I ostatak mora da bude malim slovima.

Email je obavezno,mora biti popunjen u skladu pravilima za email.Subject obavezno polje.Message je obavezno I ne sme sadrzati specijalne znake sem tacke.

+381 645891095
info@compshopper.com
f
t
in

COMP-SHOPPER

Checkout

Home
Shop
Contact
Author

GET IN TOUCH

Name
Email
Subject
Your Message Here

Submit

CONTACT INFO

COMP-Shopper Inc.
935 W. Webster Ave New Streets Chicago, IL
60614, NY
Newyork USA
Mobile: +2346 17 38 93
Fax: 1-714-252-0026
Email: info@COMP-Shopper.com

SOCIAL NETWORKING

f
t
in

Submit

SERVICE

SHOP

ABOUT

AUTHOR

Submit

COMP-SHOPPER

Buy with confidence

SERVICE

SHOP

ABOUT

AUTHOR



Online Help
Contact Us
Unlocked Cell Phones
iPads & Tablets
Wearable Technology
Televisions
Streaming & Media Players
Company Information
Location
Sitemap
Author
Documentation

Copyright © 2021 COMP-Shopper Inc. Nemanja Jezdic 30/19

checkout.html

Stranica za dalju kupovinu. Na ovoj stranici se vrši obrada svih proizvoda koje je korisnik izabrao da zeli da kupi i slanje podataka za kupovinu. Korisnik može da doda ili da izbaci proizvod iz cart-a, može da poveća ili smanji njegovu količinu (odabirom na 0 proizvoda ili pritiskom na taster x, proizvod se briše iz cart-a). Cena određeno proizvoda se skalira u zavisnosti količine izabranog proizvoda, a ukupna cena se skalira kao zbir svih proizvoda u kartu. Ako je cart prazan ispisan je opis da treba ubaciti neki proizvod u cart. Da bi se izvršila kupovina cart ne sme biti prazan inace je taster submit disabled. Da bi se izvršila kupovina polja za Billing i Shipping moraju biti popunjena. Polja First Name Last Name i Email imaju iste uslove kao i na contact-us.html. Polje zip mora imati tačno 5 broja. Address ne sme imati specijalne karaktere. Address 2 nije obavezno polje. Telefon mora početi sa +381 i da ima tačno 9 karaktera nakon toga.

CART REVIEW

Item	Price	Quantity	Total	
 Intel Core i9-10900K 3.7 GHz Ten-Core LGA 1200 Processor Web ID: 1089772	\$468.40	+ 2 -	\$936.80	✕
 Sony Alpha 1 Mirrorless Digital Camera (Body Only) Web ID: 1089772	\$6498.00	+ 1 -	\$6498.00	✕
			Cart Sub Total	\$7434.80
			Shipping Cost	Free
			Total	\$7434.80

CHECKOUT

Billing Information

First Name*
Last Name*
Email*
Address 1 *
Address 2
Zip / Postal Code *
Phone *
Country ▾

Shipping information

First Name*
Last Name*
Email*
Address 1 *
Address 2
Zip / Postal Code *
Phone *
Country ▾

Shipping Order

Notes about your order, Special Notes for Delivery

Submit



<input type="text"/>	<input type="text"/>	<div></div> <div>Submit</div>
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	
<input type="text"/>	<input type="text"/>	

<div>Billing Information</div> <div><input type="text"/></div> <div>First Name*</div> <div>Is Required and first letter must be uppercase and the rest lowercase</div> <div><input type="text"/></div> <div>Last Name*</div> <div>Is Required and first letter must be uppercase and the rest lowercase</div> <div><input type="text"/></div> <div>Email*</div> <div>Is Required or in invalid format</div> <div><input type="text"/></div> <div>Address 1 *</div> <div>Is Required, no special characters except .(dot)</div> <div><input type="text"/></div> <div>Address 2</div> <div>Zip / Postal Code *</div> <div>Only numbers, Exactly five</div> <div><input type="text"/></div> <div>Phone *</div> <div>Only numbers. Starting with +381, Exactly nine</div> <div><input type="text"/></div> <div>Country</div>	<div>Shipping information</div> <div><input type="text"/></div> <div>First Name*</div> <div>Is Required and first letter must be uppercase and the rest lowercase</div> <div><input type="text"/></div> <div>Last Name*</div> <div>Is Required and first letter must be uppercase and the rest lowercase</div> <div><input type="text"/></div> <div>Email*</div> <div>Is Required or in invalid format</div> <div><input type="text"/></div> <div>Address 1 *</div> <div>Is Required, no special characters except .(dot)</div> <div><input type="text"/></div> <div>Address 2</div> <div>Zip / Postal Code *</div> <div>Only numbers, Exactly five</div> <div><input type="text"/></div> <div>Phone *</div> <div>Only numbers. Starting with +381, Exactly nine</div> <div><input type="text"/></div> <div>Country</div>	<div>Shipping Order</div> <div>Notes about your order, Special Notes for Delivery</div> <div>Submit</div>
--	---	---

CART REVIEW

Item	Price	Quantity	Total
Cart empty. Please add products to cart...			

CHECKOUT

author.html

Opis I slika autora sajta.

AUTHOR



I would describe myself as positive hard working person that's ready for new challenges. So far I'm skilled in HTML,CSS,Bootstrap,C,Java,C#,JavaScript,Jquery,AJAX and I'm also trying to improve by learning new technologies.

I would describe myself as positive hard working person that's ready for new challenges. So far I'm skilled in HTML,CSS,Bootstrap,C,Java,C#,JavaScript,Jquery,AJAX and I'm also trying to improve by learning new technologies.

3 . Kodovi

3.1 HTML

3.2 CSS

(CSS korisncen sa teampleata ili bootstrapov)

3.3 JavaScript

checkout.js

```
compshop > js > main > JS checkoutjs > ...
1  $(document).ready(function () {
2
3      const firstNameAndLastNameRegex = /^[A-Z][a-z]+$/;
4      const emailRegex = /\S+@\S+\.\S+$/;
5      const addressRegex = /^[A-Z\d\s\.\,]+$/;
6      const zipRegex = /^[0-9]{5}$/;
7      const phoneRegex = /\+[0-9]{10,12}$/;
8
9      const checkoutFormValidationRules = {
10         firstname: {
11             regex: firstNameAndLastNameRegex,
12             message: 'Is Required and first letter must be uppercase and the rest lowercase'
13         },
14         lastname: {
15             regex: firstNameAndLastNameRegex,
16             message: 'Is Required and first letter must be uppercase and the rest lowercase'
17         },
18         email: {
19             regex: emailRegex,
20             message: 'Is Required or in invalid format'
21         },
22         address: {
23             regex: addressRegex,
24             message: 'Is Required, no special characters except .(dot)'
25         },
26         zip: {
27             regex: zipRegex,
28             message: 'Only numbers, Exactly five'
29         },
30         phone: {
31             regex: phoneRegex,
32             message: 'Only numbers. Starting with +381, Exactly nine'
33         },
34     };
35
36     $('#submit-checkout').on('click', function (e) {
37         e.preventDefault();
38
39         const billingInfoFormData = $('#billing-info-form').serializeArray().reduce((obj, f) => { obj[f.name] = f.value; return obj; }, {});
40         const shippingInfoFormData = $('#shipping-info-form').serializeArray().reduce((obj, f) => { obj[f.name] = f.value; return obj; }, {});
41
42         const billingInfoFormErrors = validateCheckoutForm(billingInfoFormData);
43         const shippingInfoFormErrors = validateCheckoutForm(shippingInfoFormData);
44
45         // show errors
46         renderErrors($('#billing-info-form'), billingInfoFormErrors);
47         renderErrors($('#shipping-info-form'), shippingInfoFormErrors);
48
49         if ($.isEmptyObject(billingInfoFormErrors) && $.isEmptyObject(shippingInfoFormErrors)) {
```

```

48
49     if ($.isEmptyObject(billingInfoFormErrors) && $.isEmptyObject(shippingInfoFormErrors)) {
50         $('#myModal').modal('show');
51         $('.billing-info-form').get(0).reset();
52         $('.shipping-info-form').get(0).reset();
53
54         let cart = getCart();
55         cart.products = [];
56         saveCart(cart);
57     }
58 });
59
60 function validateCheckoutForm(data) {
61     const errors = {};
62
63     $.each(data, (fieldName, fieldValue) => {
64         if (checkoutFormValidationRules[fieldName]) {
65             if (!checkoutFormValidationRules[fieldName].regex.test(fieldValue)) {
66                 errors[fieldName] = checkoutFormValidationRules[fieldName].message;
67             }
68         }
69     });
70
71     return errors;
72 }
73
74 function renderErrors(form, errors) {
75
76     // clear existing errors
77     $.each($(form).find('input, select'), function (i, element) {
78         $(element).parent().removeClass('has-error');
79         $(element).parent().find('.help-block').html('');
80     });
81
82     // render errors
83     if (!$.isEmptyObject(errors)) {
84         $.each(errors, function (fieldName, errorMessage) {
85             $(form).find(`input[name=${fieldName}]`).parent().addClass('has-error');
86             $(form).find(`input[name=${fieldName}]`).parent().find('.help-block').html(errorMessage);
87         });
88     }
89 };
90
91 $('body').on('click', '.cart_quantity_up', function (e) {
92     e.preventDefault();
93     const productId = $(this).attr('data-product-id');
94     let cart = getCart();
95     cart.products = cart.products.map((p) => {
96         if (p.id == productId) {

```

```

106         if (p.id == productId) {
107             p.quantity += 1;
108         }
109         return p;
110     });
111     saveCart(cart);
112     renderCart();
113 });
114
115 $('body').on('click', '.cart_quantity_down', function (e) {
116     e.preventDefault();
117     const productId = $(this).attr('data-product-id');
118     let cart = getCart();
119     cart.products = cart.products.map((p) => {
120         if (p.id == productId) {
121             p.quantity -= 1;
122         }
123         return p;
124     }).filter((p) => p.quantity > 0);
125     saveCart(cart);
126     renderCart();
127 });
128
129 $('body').on('click', '.cart_quantity_delete', function (e) {
130     e.preventDefault();
131     const productId = $(this).attr('data-product-id');
132     let cart = getCart();
133     cart.products = cart.products.filter((p) => p.id != productId);
134     saveCart(cart);
135     renderCart();
136 });
137
138 function renderCart() {
139     getProducts().done((products) => {
140         const cart = getCart();
141
142         if (!cart || !cart.products || cart.products.length <= 0) {
143             $('#cart_items .cart-table > tbody').html(renderCartEmpty());
144             $('.submit-checkout').addClass('disabled');
145             return;
146         }
147
148         $('.submit-checkout').removeClass('disabled');
149
150         const productsInCart = products
151             .filter((p) => cart
152                 .products
153                 .find(pr => p.id == pr.id) !== undefined)
154             .map((p) => {

```

JS checkout.js X

compshop > js > main > JS checkout.js > ...

```
143     .find(pr => p.id == pr.id) !== undefined)
144     .map(p => {
145         const foundProduct = cart.products.find((pr) => p.id == pr.id);
146         if (foundProduct) {
147             p.quantity = foundProduct.quantity;
148         }
149         return p;
150     });
151
152     const cartInfo = {
153         subtotal: productsInCart.reduce((subtotal, p,) => subtotal += p.price * p.quantity, 0),
154         total: productsInCart.reduce((subtotal, p,) => subtotal += p.price * p.quantity, 0),
155     };
156
157     $('#cart_items .cart-table > tbody').html(productsInCart.map(renderCartItem).join(''));
158     $('#cart_items .cart-table > tbody').append(renderCartInfo(cartInfo));
159 });
160
161 };
162
163 function renderCartItem(product) {
164     return `
165     <tr>
166         <td class="">
167             <a href=""></a>
168         </td>
169         <td class="cart_description">
170             <p>${product.name}</p>
171             <p>Web ID: ${product.webId}</p>
172         </td>
173         <td class="cart_price">
174             <p>${Number.parseFloat(product.price).toFixed(2)}</p>
175         </td>
176         <td class="cart_quantity">
177             <div class="cart_quantity_button">
178                 <a class="cart_quantity_up" href="" data-product-id="${product.id}"> + </a>
179                 <input id="cart_quantity_input_${product.id}" class="cart_quantity_input" type="text" name="quantity" value="${product.quantity}" autocomplete="off" size="2">
180                 <a class="cart_quantity_down" href="" data-product-id="${product.id}"> - </a>
181             </div>
182         </td>
183         <td class="cart_total">
184             <p class="cart_total_price">${Number.parseFloat(product.quantity * product.price).toFixed(2)}</p>
185         </td>
186         <td class="cart_delete">
187             <a class="cart_quantity_delete" href="" data-product-id="${product.id}"> <i class="fa fa-times"></i></a>
188         </td>
189     </tr>`;
190
191 }
```



```

190
191
192 function renderCartEmpty() {
193     return `
194     <tr>
195         <td colspan="6" class="text-center">
196             Cart empty. Please add products to cart...
197         </td>
198     </tr>`;
199
200 function renderCartInfo(data) {
201     return `
202     <tr>
203         <td colspan="4">&nbsp;</td>
204         <td colspan="2">
205             <table class="table table-condensed total-result">
206                 <tr>
207                     <td>Cart Sub Total</td>
208                     <td>${Number.parseFloat(data.subtotal).toFixed(2)}</td>
209                 </tr>
210                 <tr class="shipping-cost">
211                     <td>Shipping Cost</td>
212                     <td>Free</td>
213                 </tr>
214                 <tr>
215                     <td>Total</td>
216                     <td><span>${Number.parseFloat(data.total).toFixed(2)}</span></td>
217                 </tr>
218             </table>
219         </td>
220     </tr>
221     `;
222
223     renderCart();
224 });
225
226

```

contact.js

```
compshop > js > main > JS contactjs > ...
1 $(document).ready(function () {
2     const contactFormValidationRules = {
3         name: {
4             regex: /^[A-Z][a-z]+$/,
5             message: 'Is Required and first letter must be uppercase and the rest lowercase'
6         },
7         email: {
8             regex: /\S+@\S+\.\S+/,
9             message: 'Is Required or in invalid format'
10        },
11        subject: {
12            regex: /^[A-z\s]+$/,
13            message: 'Is Required'
14        },
15        message: {
16            regex: /^[A-z\s0-9\.\-]{10,300}$/,
17            message: 'Is Required, no special characters except .(dot)'
18        },
19    };
20
21    $('#main-contact-form').on('submit', function (e) {
22        e.preventDefault();
23        const errors = validateContactForm($(this).serializeArray().reduce((obj, f) => { obj[f.name] = f.value; return obj; }, {}));
24        renderErrors($(this), errors);
25
26        if ($.isEmptyObject(errors)) {
27            $.toast({ text: 'Successfully submitted message, we will respond as soon as possible.', position: 'right-bottom' });
28            $(this).get(0).reset();
29        }
30    });
31
32    function validateContactForm(data) {
33        const errors = {};
34        $.each(data, (fieldName, fieldValue) => {
35            if (contactFormValidationRules[fieldName]) {
36                if (!contactFormValidationRules[fieldName].regex.test(fieldValue)) {
37                    errors[fieldName] = contactFormValidationRules[fieldName].message;
38                }
39            }
40        });
41
42        return errors;
43    };
44
45    function renderErrors(form, errors) {
46        // clear existing errors
47        $.each($(form).find('input, select, textarea'), function (i, element) {
48            $(element).parent().removeClass('has-error');
49            $(element).parent().find('.help-block').html('');
50        });
51
52        // render errors
53        if (!$.isEmptyObject(errors)) {
54            $.each(errors, function (fieldName, errorMessage) {
55                $(form).find('input[name=' + fieldName + '], textarea[name=' + fieldName + ']').parent().addClass('has-error');
56                $(form).find('input[name=' + fieldName + '], textarea[name=' + fieldName + ']').parent().find('.help-block').html(errorMessage);
57            });
58        }
59    };
60 });
```

global.js

```
compshop > js > main > JS global.js > getProducts
1  function getProducts() {
2      return $.ajax({
3          dataType: 'json',
4          url: '/data/products.json'
5      });
6  }
7
8  function getCategories() {
9      return $.ajax({
10         dataType: 'json',
11         url: '/data/categories.json',
12     });
13 }
14
15 function getCart() {
16     let defaultValues = { products: [] };
17
18     try {
19         let cart = JSON.parse(localStorage.getItem('cart'));
20         return cart ? cart : defaultValues;
21     } catch (e) {
22         return defaultValues;
23     }
24 };
25
26 function saveCart(cart) {
27     return localStorage.setItem('cart', JSON.stringify(cart));
28 };
29
30 function addProductToCart(productId) {
31     let cart = getCart();
32
33     if (cart.products) {
34         let existingProduct = cart.products.find((p) => p.id == productId);
35         if (existingProduct) {
36             existingProduct.quantity++;
37         } else {
38             cart.products.push({ id: productId, quantity: 1 });
39         }
40     } else {
41         cart.products = [{ id: productId, quantity: 1 }];
42     }
43
44     saveCart(cart);
45 };
46
47 function removeProductFromCart(productId) {
48     let cartJson = localStorage.getItem('cart');
```

```

46
47 function removeProductFromCart(productId) {
48   let cartJson = localStorage.getItem('cart');
49
50   if (cartJson) {
51     let cart = JSON.parse(cartJson);
52
53     if (cart.products) {
54       cart.products = cart.products.filter((p) => p != productId);
55       localStorage.setItem('cart', JSON.stringify(cart));
56     }
57   }
58 };
59
60 $(document).ready(function () {
61
62   // add to cart
63   $('body').on('click', '.add-to-cart', function (e) {
64     e.preventDefault();
65     let productId = $(this).attr('data-product-id');
66
67     addProductToCart(productId);
68
69     let beforeHtml = $(this).html();
70     $(this).html('<div class="loader"></div>');
71     $(this).addClass('loading-add-to-cart');
72
73     setTimeout(() => {
74       $(this).html(beforeHtml);
75       $(this).removeClass('loading-add-to-cart');
76       $.toast({ text: 'Successfully added product to cart. <a href="/checkout.html">Go to cart</a>', position: 'bottom-center' });
77     }, 1500);
78   });
79   /*price range*/
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105 function renderCategories() {
106   getCategories()
107     .done(categories => {
108       // render on the side
109       $('#categories').html(categories.map(renderCategoryItem).join(''));
110
111       // render inside dropdown
112       let categoriesForMenu = categories.reduce((accumulator, currentValue) => currentValue.subcategories.concat(accumulator), []).map(renderCategoryMenuItem);
113       $('#categories-menu').html(categoriesForMenu.join(''));
114       $('#categories-footer').html(categoriesForMenu.slice(0, 5).join(''));
115     })
116     .fail(error => console.log('Error loading categories'));
117 };
118
119 function renderCategoryItem(category) {
120   let subcategories = '';
121   let hasSubcategories = category.subcategories && category.subcategories.length > 0;
122
123   if (hasSubcategories) {
124     subcategories = `
125       <div id="accordion-${category.id}" class="panel-collapse collapse">
126         <div class="panel-body">
127           <ul>
128             ${category.subcategories.map((c) => `<li id="${c.id}"> <a href="/shop.html?categoryId=${c.id}">${c.name}</a></li>`).join('')}
129           </ul>
130         </div>
131       </div>
132     `;
133   }
134
135   return `
136     <div class="panel panel-default">
137       <div id="category-${category.id}" class="panel-heading">
138         <h4 class="panel-title">
139           <a data-toggle="collapse" data-parent="#accordion" href="#accordion-${category.id}">
140             ${hasSubcategories ? `<span class="badge pull-right"><i class="fa fa-plus"></i></span>` : ''}
141             ${category.name}
142           </a>
143         </h4>
144       </div>
145       ${subcategories}
146     </div>`;
147 };
148
149 function renderCategoryMenuItem(category) { return `<li><a href="/shop.html?categoryId=${category.id}">${category.name}</a></li>`; }
150 renderCategories();
151 });

```

Index.js

```

1 $(document).ready(function () {
2
3     function renderProducts() {
4         getProducts()
5             .done(products => {
6                 // featured items
7                 $('#featured-items').html(
8                     products.filter((p) => p.featured).map(renderProductItem).join('')
9                 );
10
11                 // recommended items
12                 $('#recommended-item-carousel .carousel-inner').html(
13                     `
14                     <div class="item active">${products.filter((p) => p.recommended).splice(0, 3).map(renderProductCarouselItem).join('')}</div>
15                     <div class="item">${products.filter((p) => p.recommended).splice(3, 6).map(renderProductCarouselItem).join('')}</div>
16                     `
17                 );
18             })
19         .fail(error => console.log("Error loading products"));
20     };
21
22     function renderProductItem(product) {
23         return `
24         <div class="col-sm-4">
25             <div class="product-image-wrapper" style="height:365px;">
26                 <div class="single-products">
27                     <div class="productinfo text-center">
28                         
29                         <h2>${product.price}</h2>
30                         <p>${product.name}</p>
31                         <a href="#" class="btn btn-default add-to-cart" data-product-id="${product.id}" ><i class="fa fa-shopping-cart"></i>Add to cart</a>
32                     </div>
33                     <div class="product-overlay">
34                         <div class="overlay-content">
35                             <h2>${product.price}</h2>
36                             <p>${product.name}</p>
37                             <a href="#" class="btn btn-default add-to-cart" data-product-id="${product.id}" ><i class="fa fa-shopping-cart"></i>Add to cart</a>
38                         </div>
39                     </div>
40                 </div>
41             </div>
42         </div>`;
43     };
44

```

```

45     function renderProductCarouselItem(product) {
46         return `
47         <div class="col-sm-4">
48             <div class="product-image-wrapper" style="height:365px;">
49                 <div class="single-products">
50                     <div class="productinfo text-center">
51                         
52                         <h2>${product.price}</h2>
53                         <p>${product.name}</p>
54                         <a href="#" class="btn btn-default add-to-cart" data-product-id="${product.id}" ><i class="fa fa-shopping-cart"></i>Add to cart</a>
55                     </div>
56                     <div class="product-overlay">
57                         <div class="overlay-content">
58                             <h2>${product.price}</h2>
59                             <p>${product.name}</p>
60                             <a href="#" class="btn btn-default add-to-cart" data-product-id="${product.id}" ><i class="fa fa-shopping-cart"></i>Add to cart</a>
61                         </div>
62                     </div>
63                 </div>
64             </div>
65         </div>`;
66     };
67
68     const renderFiltersBySubcategory = function () {
69         return {
70             currentCategory: null,
71             categoriesWithProducts: [],
72             init: function () {
73                 getCategories().done((categories) => {
74                     getProducts().done((products) => {
75                         // randomly select subcategories
76                         categories.forEach((c) => {
77                             let subcategory = c.subcategories[Math.floor(Math.random() * (c.subcategories.length - 0) + 0)];
78                             let productsForSubcategory = products.filter((p) => subcategory.id === p.categoryId).slice(0, 3);
79                             this.categoriesWithProducts.push(Object.assign({}, subcategory, { products: productsForSubcategory }));
80                         });
81                         this.currentCategory = this.categoriesWithProducts[0];
82                     });
83                     $('#category-tabs').html(
84                         this.renderCategoryTabNav(this.categoriesWithProducts) + this.renderCategoryTabs(this.categoriesWithProducts)
85                     );
86                 });
87             },
88             renderCategoryTabNav: function (categories) {
89                 return `
90                 <div class="col-sm-12">
91                     <ul class="nav nav-tabs">
92                         ${categories.map((c) => this.renderCategoryTabNavItem(c)).join('')}
93                     </ul>
94                 </div>`;
95             },
96             renderCategoryTabNavItem: function (category) { return `<li class="${this.currentCategory.id === category.id ? 'active' : ''}"><a href="#category_tab_${category.id}" data-toggle="tab">${category.name}</a></li>`; },
97             renderCategoryTabs: function (categories) { return `<div class="tab-content">${categories.map((c) => this.renderCategoryTab(c)).join('')}</div>`; },
98             renderCategoryTab: function (category) { return `<div class="tab-pane fade in ${this.currentCategory.id === category.id ? 'active' : ''}" id="category_tab_${category.id}"> ${category.products.map((p) => this.renderProductItem(p)).join('')}</div>`; },
99             fade in ${this.currentCategory.id === category.id ? 'active' : ''}" id="category_tab_${category.id}"> ${category.products.map(this.renderProductItem).join('')}</div>`; },
100         };
101     };

```

```

92     renderCategoryTabItem: function (product) {
93         return `
94         <div class="col-sm-4">
95             <div class="product-image-wrapper">
96                 <div class="single-products">
97                     <div class="productinfo text-center">
98                         
99                         <h2>${product.price}</h2>
100                        <p>${product.name}</p>
101                        <a href="#" class="btn btn-default add-to-cart" data-product-id="${product.id}"><i class="fa fa-shopping-cart"></i>Add to cart</a>
102                    </div>
103                </div>
104            </div>
105        </div>`;
106    },
107 }
108 });
109
110 renderProducts();
111 renderFiltersBySubcategory().init();
112 });

```

shop.js

```

compshop > js > main > JS shop.js > ...
1  $(document).ready(function () {
2      let urlParams = new URLSearchParams(window.location.search);
3      let categoryId = urlParams.get('categoryId');
4
5      $('.price-range #sl2').slider();
6
7      $(".price-range #sl2").on("slideStop", function (event) {
8          loadProducts();
9      });
10
11     getProducts().done((products) => {
12         let filteredProducts = products;
13
14         if (categoryId) {
15             filteredProducts = products.filter((p) => p.categoryId == categoryId);
16         }
17
18         $('#category-container').html(
19             filteredProducts.map(renderProductItem).join('')
20         );
21     });
22
23     function loadProducts() {
24         getProducts().done((products) => {
25             let filteredAndSortedProducts = products;
26
27             if (categoryId) {
28                 filteredAndSortedProducts = products.filter((p) => p.categoryId == categoryId);
29             }
30
31             let range = $(".price-range #sl2").val() == '' ? [0, 3000] : $(".price-range #sl2").val().split(',');
32             let sortField = $('#sort-by-container #sort-by-select').val();
33             filteredAndSortedProducts = filteredAndSortedProducts
34                 .filter(p => p.price > range[0] && p.price < range[1])
35                 .sort(function (a, b) {
36                     if ($('#sort-by-container input:checked').val() == 'asc') {
37                         if (a[sortField] < b[sortField]) {
38                             return -1;
39                         }
40                         if (a[sortField] > b[sortField]) {
41                             return 1;
42                         }
43                     } else {
44                         if (a[sortField] > b[sortField]) {
45                             return -1;
46                         }
47                         if (a[sortField] < b[sortField]) {
48                             return 1;
49                         }

```

```

49         }
50     });
51 });
52
53 $('#category-container').html(
54     filteredAndSortedProducts.map(renderProductItem).join('')
55 );
56 });
57
58
59 $('#sort-by-container #sort-by-select, #sort-by-container .checkbox-inline input[type=radio]').on('change', function (e) {
60     if ($('#sort-by-container #sort-by-select').val() != '') {
61         loadProducts();
62     }
63 });
64
65 function renderProductItem(product) {
66     return
67     <div class="col-sm-4">
68         <div class="product-image-wrapper" style="height:365px;">
69             <div class="single-products">
70                 <div class="productinfo text-center">
71                     
72                     <h2>${product.price}</h2>
73                     <p>${product.name}</p>
74                     <a href="#" class="btn btn-default add-to-cart" data-product-id="${product.id}" ><i class="fa fa-shopping-cart"></i>Add to cart</a>
75                 </div>
76                 <div class="product-overlay">
77                     <div class="overlay-content">
78                         <h2>${product.price}</h2>
79                         <p>${product.name}</p>
80                         <a href="#" class="btn btn-default add-to-cart" data-product-id="${product.id}" ><i class="fa fa-shopping-cart"></i>Add to cart</a>
81                     </div>
82                 </div>
83             </div>
84         </div>
85     </div>`;
86 });
87

```