

НАПРЕДЕН ВЕБ ДИЗАЈН – ПРОЕКТ GLAM HOUSE (ONLINE SHOP)

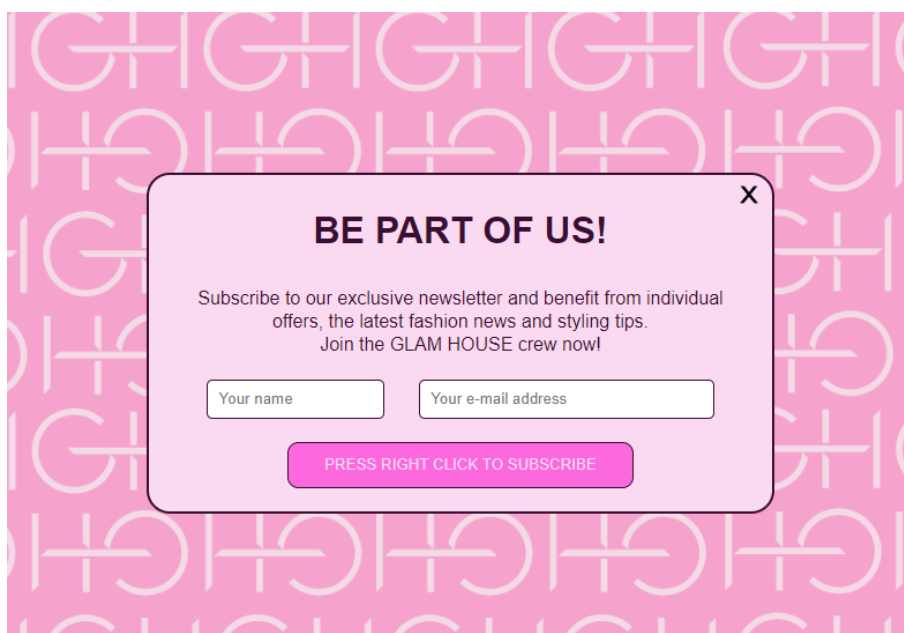
Како проект по предметот Напреден Веб Дизајн изработив веб сајт Glam House, кој е онлајн продавница за облека и различни секојдневни производи наменети за женската популација. За реализација на овој веб сајт користев HTML, CSS и Javascript, а го користев framework-от Vue.js, преку **SFC (Single File Components)** и навигација преку **Vue Router**. Во овој документ е опишан овој веб сајт како функционира и со кој код/функционалност е реализиран истиот. **(ЗАБЕЛЕШКА:** Овој веб сајт е направен со цел да се прикаже користењето на framework-от Vue.js од повеќе различни аспекти и акцентот е на тоа, за разлика од дизајнот и user experience-от. Во реален случај веб сајтот би бил многу повеќе оптимизиран со запазување на детали од секој аспект)

(ЗАБЕЛЕШКА 2: Некои од текстуалните описи на vue-функционалностите подолу, се подвлечени со иста боја со соодветниот код за подобро разбирање на нивната поврзаност)

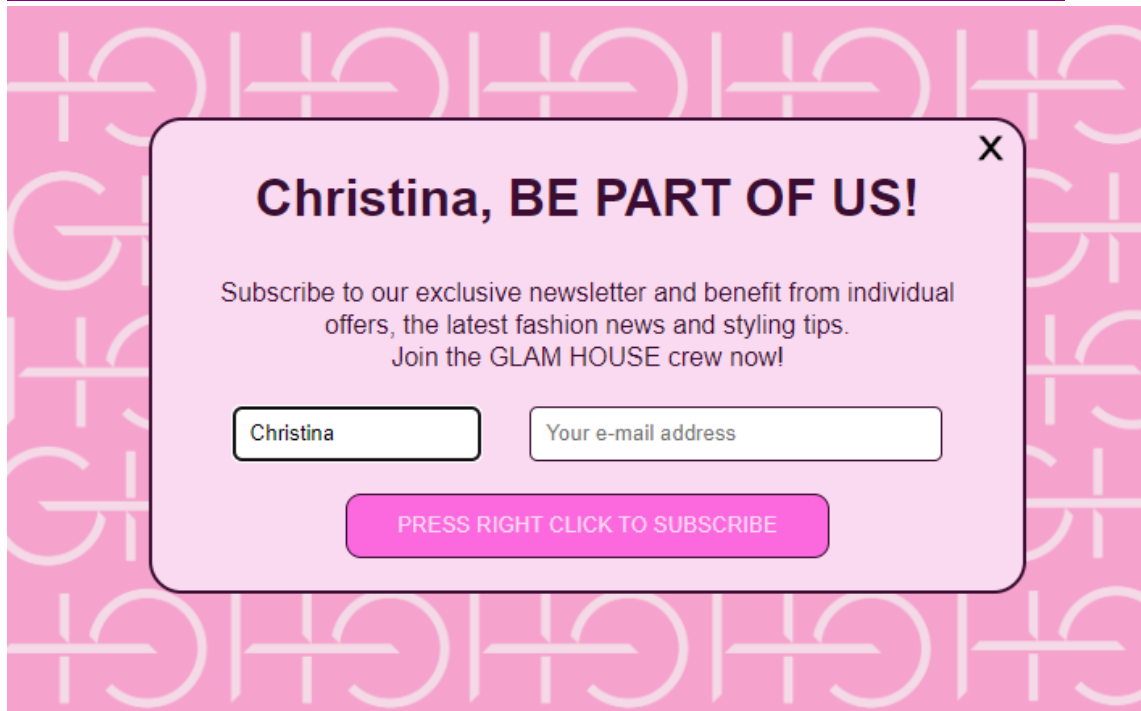
Искористени други vue функционалности во проектот: **v-bind, v-if, v-show, v-for, events, v-on, methods, event modifiers, forms, v-model, computed properties, props, \$emit, slots, refs, lifecycle hooks, local storage, animations, composition API.**

ОПИС НА ВЕБ САЈТ:

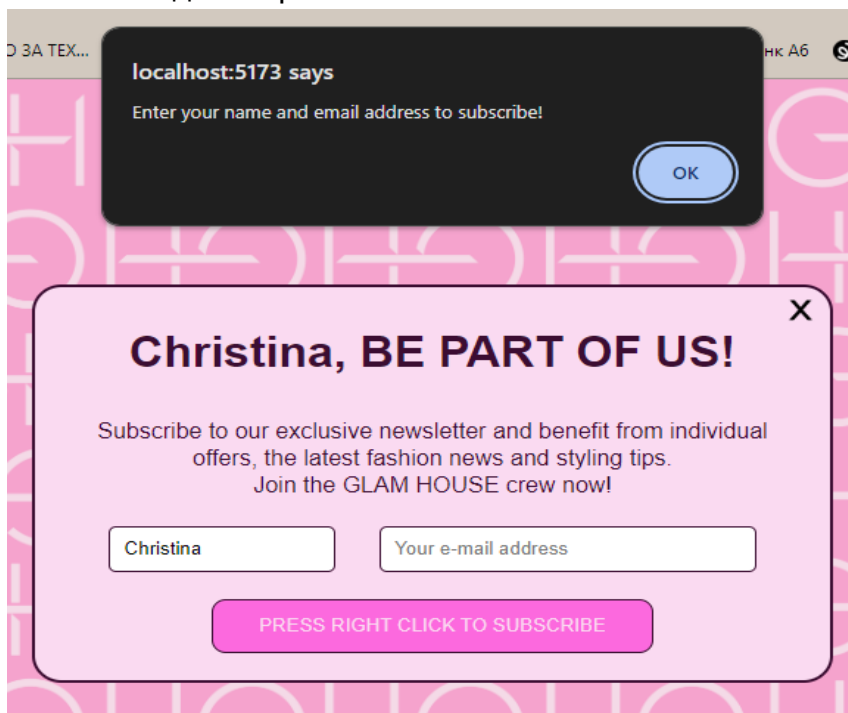
Кога корисникот ќе го отвори веб сајтот Glam House, прво му се прикажува компонента (SubscriptionFormComponent.vue) што е форма за претплата.



Овде има опција да внесе име и меил за да се претплати или да ја исклучи формата од X-от горе десно. Доколку внесе име, во истиот момент додека го внесува името ќе се прикаже горе: “неговото име, BE PART OF US!”.



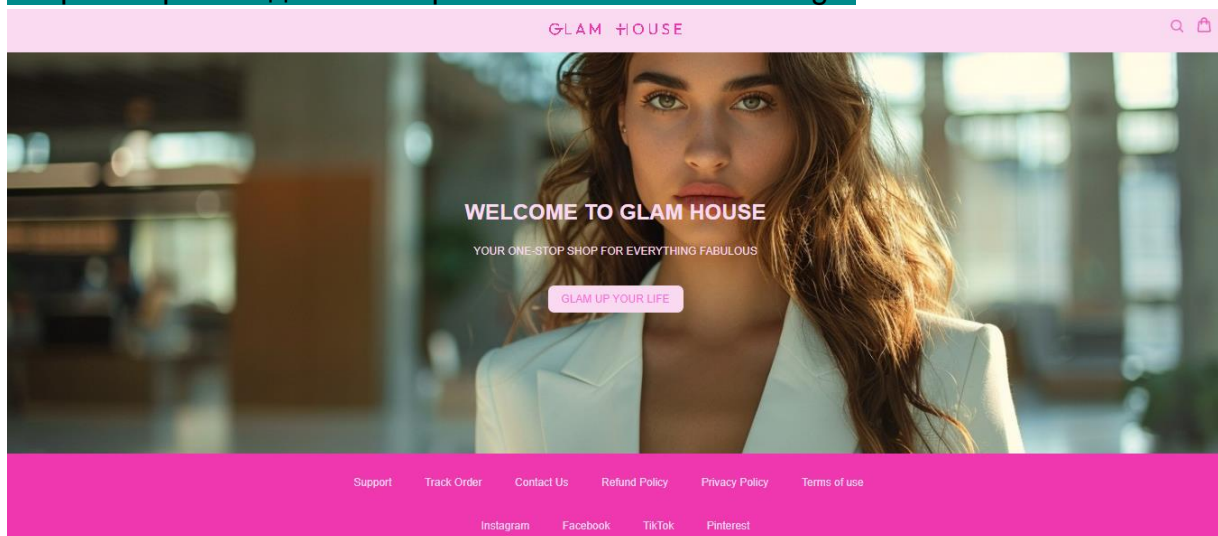
Доколку само едно поле од двете е пополнето (било кое), а корисникот притисне (десен клик како што е наведено) на копчето: “PRESS RIGHT CLICK TO SUBSCRIBE” ќе го извести корисникот alert дека мора да ги внесе двете полиња за да се претплати.



Кога корисникот ќе ги пополни двете полиња, ќе може со десен клик на копчето да се “претплати” и ќе му се појави анимација на копчето со која тоа се придвижува нагоре (доколку можете да приметите на сликата подолу дека копчето е поместено погоре).



Потоа ќе му се прикаже главната страна на веб сајтот (HomePage.vue) што е друга компонента и таа содржи: мени со лого и 2 копчиња (што се дел од компонентата SearchAndCartComponent.vue) што пренасочуваат едното до страна за пребарување на производи (SearchPage.vue) а другото до купувачка кошничка (ShoppingCartPage.vue), централна слика и опис на продавницата како и копче кое води до страна каде што се наоѓаат сите производи (GlamUpYourLifePage.vue), (HomeCenterComponent.vue), и футер (FooterComponent.vue) што содржи линкови до социјални мрежи од мојата реална продавница, и празни линкови за корисничка поддршка и слично. Корисникот нема да може да се врати на формата за претплата се додека не се “ресетира” соодветната променлива во Local Storage.



(сликата од HomePage е одзумирана на 75% за да се види целосно без скролање)

Кодот на SubscriptionFormComponent.vue:

```
<template>
  <div id="form">
    
    <h1>{{ formattedUserData }} BE PART OF US!</h1>
    <p>Subscribe to our exclusive newsletter
      and benefit from individual offers,
      the latest fashion news and styling tips.
    <br>Join the GLAM HOUSE crew now!</p>
    <form>
      <div id="nameAndEmail">
        <input id="name" placeholder="Your name" type="text" v-
model="userName" required>
        <input id="email" placeholder="Your e-mail address"
type="email" v-model="userEmail" required>
      </div>
      <input ref="b" id="button" type="button" value="PRESS RIGHT CLICK
TO SUBSCRIBE"
        :class="{bounce: isBouncing}"
        @click.right.prevent="bounceAndExit"> // ovde se koristi ref
    </form>
  </div>
</template>

<script>
export default {
  data() {
    return {
      userName: '',
      userEmail: '',
      isBouncing: false
    };
  },
  computed: {
    formattedUserData() {
      return this.userName ? `${this.userName}, ` : '';
    }
  },
  methods: {
    bounceAndExit() {
      this.isBouncing = false;
      this.$nextTick(() => {
        // so ovoj kod se osigurame deka DOM se apdejtira pred da ja
        dodademe bounce klasata
        const button = this.$refs.b; // ovde se koristi ref
        button.offsetHeight; //se presmetuva visinata na elementot
        this.isBouncing = true;
        setTimeout(() => {
```

```

        this.isBouncing = false;
        if (this.userEmail === '' || this.userName === '') {
            alert('Enter your name and email address to
subscribe!');
        } else {
            this.$emit('catchExit', true);
        }
    }, 2000);
    });
},
exit() {
    this.$emit('catchExit', true);
}
}
};
</script>

<style scoped>
.bounce {
    animation: bounce 1s;
}
@keyframes bounce {
    0%, 100% {
        transform: translateY(0);
    }
    20% {
        transform: translateY(-0.8rem);
    }
    50% {
        transform: translateY(-0.4rem);
    }
    80% {
        transform: translateY(-0.2rem);
    }
}
}. . . останат CSS

```

Кодот на HomePage.vue:

```

<template>
    <div id="subscriptionPage" v-show="showSubPage">
        <div id="pozadina">
            <SubscriptionFormComponent
@catchExit="changeSeenValueAndHideFormPage"/>
        </div>
    </div>
    <div id="homePage" v-show="showHomePage">
        <div id="rozovoMeni">
            
            <SearchAndCartComponent id="sc"/>

```

```

    </div>
    <div id="slikaSoDevojkaIForma">
      
      <HomeCenterComponent id="hcc"/>
    </div>
    <FooterComponent/>
  </div>
</template>

<script>
export default {
  data() {
    return {
      userHasSeenSubscriptionPage: false,
      showSubPage: true,
      showHomePage: false
    };
  },
  methods: {
    changeSeenValueAndHideFormPage() {
      this.userHasSeenSubscriptionPage = true;
      this.showSubPage = false;
      this.showHomePage = true;
      localStorage.setItem('userHasSeenSubscriptionPage', 'true');
    }
  },
  created() {
    if (localStorage.getItem('userHasSeenSubscriptionPage') === 'true') {
      this.showSubPage = false;
      this.showHomePage = true;
    } else {
      this.showSubPage = true;
      this.showHomePage = false;
    }
  }
};
</script>
. . . CSS styles

```

Кодот на HomeCenterComponent.vue:

```

<template>
  <div id="form">
    <h1>WELCOME TO GLAM HOUSE</h1>
    <p>YOUR ONE-STOP SHOP FOR EVERYTHING FABULOUS</p>
    <RouterLink id="kopce" to="/products">GLAM UP YOUR LIFE</RouterLink>
  </div>
</template> . . . CSS styles

```

Кодот на SearchAndCartComponent.vue:

```
<template>
  <div>
    <RouterLink to="/search">
      
    </RouterLink>
    <RouterLink to="/myshoppingcart">
      
    </RouterLink>
  </div>
</template> . . . CSS styles
```

Кодот на FooterComponent.vue: **(Composition API)**

```
<template>
  <div id="f">
    <nav id="fl">
      <a v-for="link in footerLinks" :key="link.name" :href="link.url"
rel="noopener noreferrer">{{ link.name }}</a>
    </nav>
    <nav id="sl">
      <a v-for="sm in socials" :key="sm.name" :href="sm.url"
target="_blank" rel="noopener noreferrer">{{ sm.name }}</a>
    </nav>
  </div>
</template>
<script> // koristeno Composition API
import { ref } from 'vue';

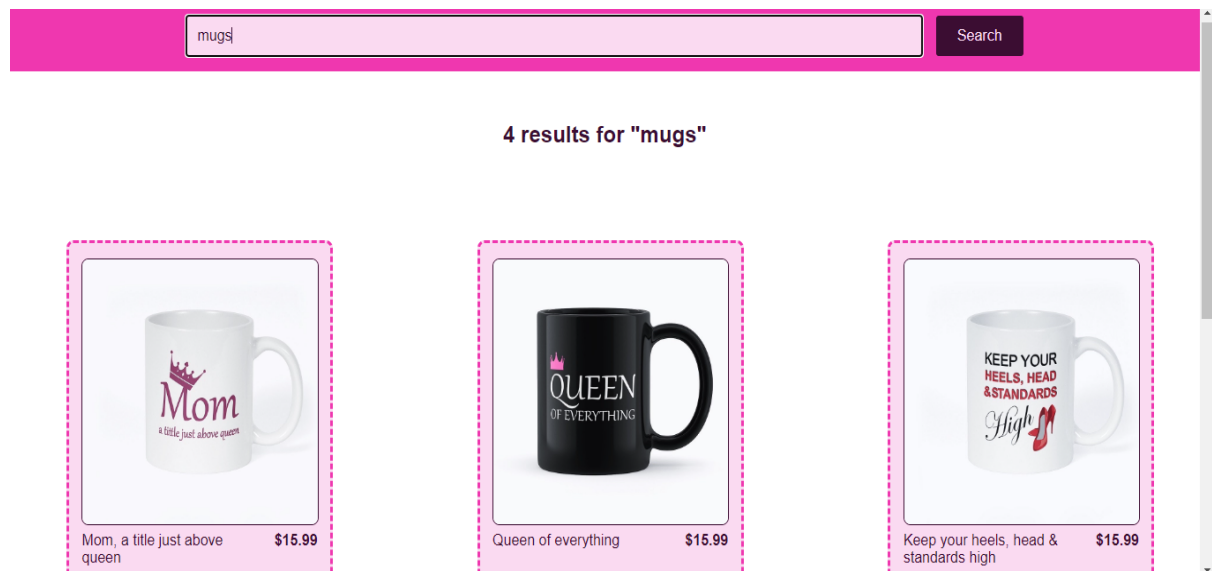
export default{
  setup () {
    const footerLinks = ref ([
      { name: 'Support', url: '#'},
      { name: 'Track Order', url: '#'},
      { name: 'Contact Us', url: '#'},
      { name: 'Refund Policy', url: '#'},
      { name: 'Privacy Policy', url: '#'},
      { name: 'Terms of use', url: '#'}
    ]);
    const socials = ref ([
      { name: 'Instagram', url:
'https://www.instagram.com/glamhouseofficial'},
      { name: 'Facebook', url:
'https://www.facebook.com/profile.php?id=100090908719703'},
      { name: 'TikTok', url:
'https://www.tiktok.com/@glamhousestoreofficial'},
      { name: 'Pinterest', url:
'https://www.pinterest.com/glamhouseofficial'}
```

```

    });
    return {
      footerLinks, socials
    }
  }
};
</script>
. . . CSS styles

```

На сликава подолу е прикажана страната SearchPage.vue на која со внес на податоци и клик на копчето Search или **клик на тастатурата на Enter** може да се побара соодветен производ по било која категорија или име на истиот. Доколку производот постои се прикажуваат сите такви производи со исто/слично име или категорија **со повторно искористување на SlotComponent.vue каде податоците ги праќаме преку props.** А доколку производот не постои соодветно се прикажува порака доколку внесениот производ што се бара не постои.



Кодот на SearchPage.vue:

```
<template>
  <div id="searchPage">
    <div id="searchBar">
      <input
        type="text"
        v-model="text"
        placeholder="Search everything"
        @keyup.enter="search"
      />
      <button @click="search">Search</button>
    </div>
    <div id="productsWindow" v-show="showResults">
      <h2 v-if="numberOfFoundProducts > 1">
        {{ numberOfFoundProducts }} results for "{{ searchText }}"
      </h2>
      <h2 v-else-if="numberOfFoundProducts === 1">
        {{ numberOfFoundProducts }} result for "{{ searchText }}"
      </h2>
      <h2 v-else-if="numberOfFoundProducts === 0 && text &&
showResults">
        No results for "{{ searchText }}"
      </h2>
      <div id="prikazNaProizvodi" v-show="numberOfFoundProducts > 0">
        <SlotComponent
          v-for="product in filteredProducts"
          :key="product.url"
          :url="product.url"
          :productName="product.name"
          :type="product.type"
          :price="product.price"
        />
      </div>
    </div>
  </div>
</template>

<script>
export default {
  data() {
    return {
      text: '',
      showResults: false,
      searchText: '',
      productsDatabase: [
        { section1: 'Clothing', section2: 'T-shirts', name: 'LOVE',
type: 'Classic t-shirt', price: '$21.99', url: '/products/clothing/t-shirts/t-
shirt.png' },
```

```

        { section1: 'Clothing', section2: 'Athletic', name: 'GHL',
type: 'Leggings', price: '$49.99',
url: '/products/clothing/athletic/leggings.png' },
        { section1: 'Clothing', section2: 'Athletic', name: 'GHB',
type: 'Sports Bra', price: '$39.99', url: '/products/clothing/athletic/sports
bra.png' },
        { section1: 'Accessories', section2: 'Bags', name: 'Heart
Art', type: 'Bag', price: '$32.99', url: '/products/accessories/bags/bag.png'
},
        { section1: 'Accessories', section2: 'Phone Cases', name:
'Behind every successful woman is herself', type: 'Phone Case', price:
'$17.99', url: '/products/accessories/phone cases/phone case.png' },
        { section1: 'Homeware', section2: 'Beach Towels', name:
'LOVE', type: 'Beach Towel', price: '$34.99', url: '/products/homeware/beach
towels/beach towel.jpg' },
        { section1: 'Homeware', section2: 'Blankets', name: 'LOVE',
type: 'Blanket', price: '$49.99', url:
'/products/homeware/blankets/blanket.jpg' },
        { section1: 'Homeware', section2: 'Pillows', name: 'LOVE',
type: 'Pillow', price: '$24.99', url: '/products/homeware/pillows/pillow.png'
},
        { section1: 'Office', section2: 'Notebooks', name: 'L O V E',
type: 'Notebook', price: '$18.99', url:
'/products/office/notebooks/notebook.jpg' },
        { section1: 'Office', section2: 'Mugs', name: 'Mom, a title
just above queen', type: 'Mug', price: '$15.99', url:
'/products/office/mugs/mug1.jpg' },
        { section1: 'Office', section2: 'Mugs', name: 'Queen of
everything', type: 'Mug', price: '$15.99', url:
'/products/office/mugs/mug2.jpg' },
        { section1: 'Office', section2: 'Mugs', name: 'Keep your
heels, head & standards high', type: 'Mug', price: '$15.99', url:
'/products/office/mugs/mug3.jpg' },
        { section1: 'Office', section2: 'Mugs', name: 'Queens are born
in March', type: 'Mug', price: '$15.99', url: '/products/office/mugs/mug4.png'
}
    ]
};
},
computed: {
    filteredProducts() {
        const searchTextLower = this.searchText.toLowerCase();
        return this.productsDatabase.filter(product =>
            product.type.toLowerCase().includes(searchTextLower) ||
            product.section2.toLowerCase().includes(searchTextLower) ||
            product.section1.toLowerCase().includes(searchTextLower) ||
            product.name.toLowerCase().includes(searchTextLower)
        );
    }
}

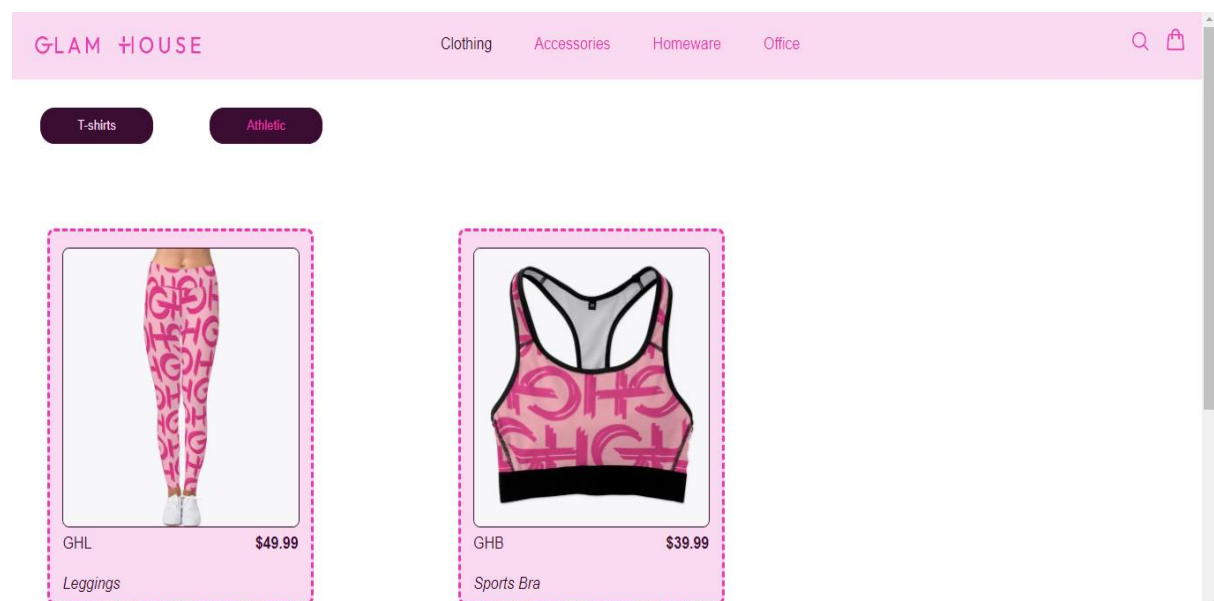
```

```

    },
    numberOfFoundProducts() {
        return this.filteredProducts.length;
    }
},
methods: {
    search() {
        if(this.text === ''){
            this.showResults = false;
        }else{
            this.searchText = this.text;
            this.showResults = true;
        }
    }
}
}
</script> . . . CSS styles

```

На следнава слика е прикажана GlamUpYourLifePage.vue компонентата. Со клик на една од главните категории на менито (MenuComponent.vue) на корисникот му се прикажуваат соодветни копчиња, каде одбраните производи се поделени на под-категории, и тој треба да избере копче по желба за да му се прикажат посакуваните производи, а пред да избере копче воопшто прикажана е првата опција по default. Производите се прикажануваат преку слот компоненти чии податоци до SlotComponent.vue се испратени преку props, а со клик на сликата на производот што корисникот сака да го види детално, се пренасочува до компонентата ProductPage.vue што е страна со приказ на податоците од избраниот производ, а податоците до неа се праќаат преку routerLink. Најдоле си стои футерот.



Кодот на MenuComponent.vue:

```
<template>
<div id="meni">
  <RouterLink to="/">
    
  </RouterLink>
  <nav id="linkovi">
    <a id="section1" v-for="section in sections1" :key="section.name"
      :class="{ 'purpleButton': clickedBtn === section.name}"
      @click="sePrakjaKliknatoKopce(section.name)">{{ section.name }}</a>
  </nav>
  <SearchAndCartComponent id="sc"/>
</div>
</template>

<script>
export default {
  data() {
    return {
      sections1: [
        { name: 'Clothing'},
        { name: 'Accessories'},
        { name: 'Homeware'},
        { name: 'Office'}
      ],
      clickedBtn: 'Clothing'
    };
  },
  methods: {
    sePrakjaKliknatoKopce(sectionName){
      this.clickedBtn = sectionName;
      this.$emit('primanjeNaIzbor', sectionName)
    }
  }
};
</script> . . . CSS styles
```

Кодот на GlamUpYourLifePage.vue:

```
<template>
  <MenuComponent @primanjeNaIzbor="izbor"/>
  <div id="productsWindow">
    <div id="kopcinja">
      <button v-for="(k, index) in kopcinja"
        :key="index"
        v-show="k.section1 === selected1DONE"
        :class="{ 'clickedButton': currSelectedBtn ===
k.section2ImeNaKopce}"
        @click="handleChosenProducts(k.section2ImeNaKopce)">
```

```

        {{ k.section2ImeNaKopce }}
      </button>
    </div>
    <div id="prikazNaProizvodi">
      <SlotComponent v-for="(product,index) in productsDatabase"
:key="index"
      :url="product.url"
      :productName="product.name"
      :type="product.type"
      :price="product.price"
      v-show="product.section2 === kategorijaZaPrikaz" />
    </div>
  </div>
  <FooterComponent/>
</template>
<script>
  export default {
    data () {
      return {
        productsDatabase: [
          {
            section1: 'Clothing',
            section2: 'T-shirts',
            name: 'LOVE',
            type: 'Classic t-shirt',
            price: '$21.99',
            url: '/products/clothing/t-shirts/t-shirt.png'},
          {
            section1: 'Clothing',
            section2: 'Athletic',
            name: 'GHL',
            type: 'Leggings',
            price: '$49.99',
            url: '/products/clothing/athletic/leggings.png'},
          {
            section1: 'Clothing',
            section2: 'Athletic',
            name: 'GHB',
            type: 'Sports Bra',
            price: '$39.99',
            url: '/products/clothing/athletic/sports bra.png'},
          {
            section1: 'Accessories',
            section2: 'Bags',
            name: 'Heart Art',
            type: 'Bag',
            price: '$32.99',
            url: '/products/accessories/bags/bag.png'},

```

```

    {
      section1: 'Accessories',
      section2: 'Phone Cases',
      name: 'Behind every successful woman is herself',
      type: 'Phone Case',
      price: '$17.99',
      url: '/products/accessories/phone cases/phone
case.png'},
    {
      section1: 'Homeware',
      section2: 'Beach Towels',
      name: 'LOVE',
      type: 'Beach Towel',
      price: '$34.99',
      url: '/products/homeware/beach towels/beach
towel.jpg'},
    {
      section1: 'Homeware',
      section2: 'Blankets',
      name: 'LOVE',
      type: 'Blanket',
      price: '$49.99',
      url: '/products/homeware/blankets/blanket.jpg'},
    {
      section1: 'Homeware',
      section2: 'Pillows',
      name: 'LOVE',
      type: 'Pillow',
      price: '$24.99',
      url: '/products/homeware/pillows/pillow.png'},
    {
      section1: 'Office',
      section2: 'Notebooks',
      name: 'L O V E',
      type: 'Notebook',
      price: '$18.99',
      url: '/products/office/notebooks/notebook.jpg'},
    {
      section1: 'Office',
      section2: 'Mugs',
      name: 'Mom, a title just above queen',
      type: 'Mug',
      price: '$15.99',
      url: '/products/office/mugs/mug1.jpg'},
    {
      section1: 'Office',
      section2: 'Mugs',
      name: 'Queen of everything',

```

```

        type: 'Mug',
        price: '$15.99',
        url: '/products/office/mugs/mug2.jpg'},
    {
        section1: 'Office',
        section2: 'Mugs',
        name: 'Keep your heels, head & standards high',
        type: 'Mug',
        price: '$15.99',
        url: '/products/office/mugs/mug3.jpg'},
    {
        section1: 'Office',
        section2: 'Mugs',
        name: 'Queens are born in March',
        type: 'Mug', price: '$15.99',
        url: '/products/office/mugs/mug4.png'
    }
],
kopcinja: [
    {section1: 'Clothing', section2ImeNaKopce: 'T-shirts'},
    {section1: 'Clothing', section2ImeNaKopce: 'Athletic'},
    {section1: 'Accessories', section2ImeNaKopce: 'Bags'},
    {section1: 'Accessories', section2ImeNaKopce: 'Phone
Cases'}},
    {section1: 'Homeware', section2ImeNaKopce: 'Beach
Towels'}},
    {section1: 'Homeware', section2ImeNaKopce: 'Blankets'},
    {section1: 'Homeware', section2ImeNaKopce: 'Pillows'},
    {section1: 'Office', section2ImeNaKopce: 'Mugs'},
    {section1: 'Office', section2ImeNaKopce: 'Notebooks'}
],
selected1DONE: 'Clothing',
kategorijaZaPrikaz: 'T-shirts',
currSelectedBtn: 'T-shirts'
}
},
methods: {
    izbor (izbranoKopce){
        this.selected1DONE = izbranoKopce;
    },
    handleChosenProducts (section2Name){
        this.kategorijaZaPrikaz = section2Name;
        this.currSelectedBtn = section2Name;
    }
}
}
}
</script>. . . CSS styles

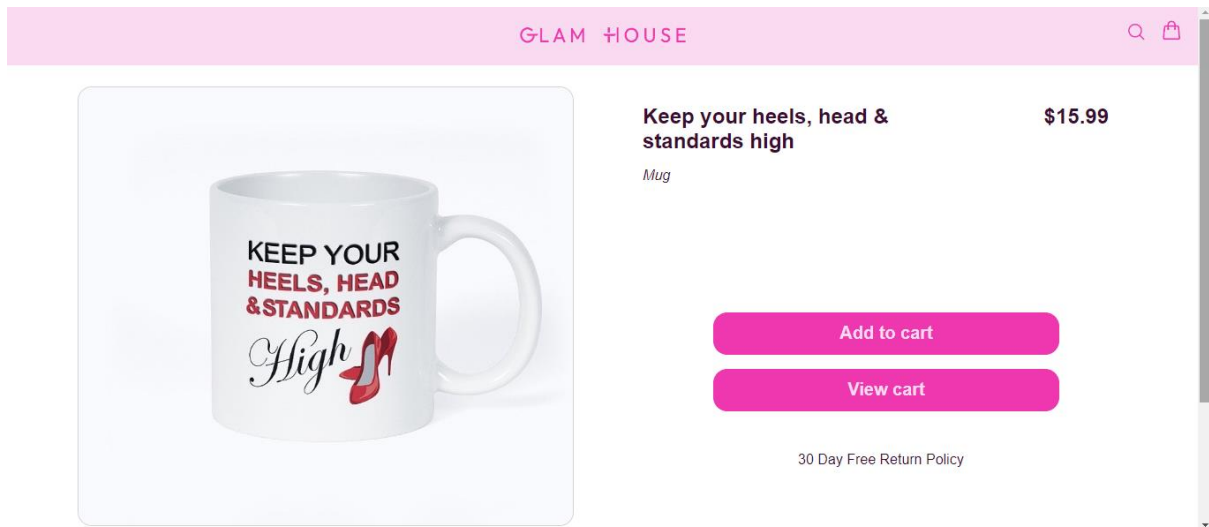
```

Кодот на SlotComponent.vue:

```
<template>
  <div id="GUYL">
    <slot>
      <router-link :to="{name: 'ProductPage', params: {productName:
productName},
      query: {
        price: price,
        url: url,
        type: type
      }}">
        
      </router-link>
      <div>
        <p class="ime">{{ productName }}</p>
        <p class="цена"><strong>{{ price }}</strong></p>
        <p class="тип"><i>{{ type }}</i></p>
      </div>
    </slot>
  </div>
</template>

<script>
export default {
  props: {
    url: String,
    productName: String,
    price: String,
    type: String
  }
}
</script> . . . CSS styles
```

Следно на сликата подолу е ProductPage.vue компонентатата со прикажани податоци за производот што корисникот го избра на претходната страна. Со клик на логото во средина на менито корисникот се враќа на HomePage.vue, а со клик на копчето Add to cart се додава производ во купувачка кошничка, односно низата со производи од купувачката кошничка се зачувува во local storage и ја земаме од таму кога додаваме нов производ и потоа ја враќаме апдејтирана. До страната каде што се наоѓа нашата купувачка кошничка (shoppingCartPage.vue) може да пристапиме преку иконата со торбичка во горниот десен кош или со клик на копчето View cart кое го пренасочува корисникот до таму преку router-link. Најдоле на страната си стои футерот.



Кодот на ProductPage.vue:

```
<template>
  <div id="wholePage">
    <div id="rozovoMeni">
      <RouterLink to="/">
        
      </RouterLink>
      <SearchAndCartComponent id="sc"/>
    </div>
    <div id="productDetails">
      <div class="photoContainer">
        
      </div>
      <div id="dataContainer">
        <h2 class="ime">{{ productName }}</h2>
        <h2 class="cena"><strong>{{ price }}</strong></h2>
        <p class="tip"><i>{{ type }}</i></p>
        <button id="atc" @click="addToCart"><h2>Add to
cart</h2></button>
        <router-link to="/myshoppingcart">
          <button id="vc"><h2>View cart</h2></button>
        </router-link>
        <p class="return">30 Day Free Return Policy</p>
      </div>
    </div>
    <FooterComponent/>
  </div>
</template>
<script>
export default {
  data() {
    return {
      productName: '',
```

```

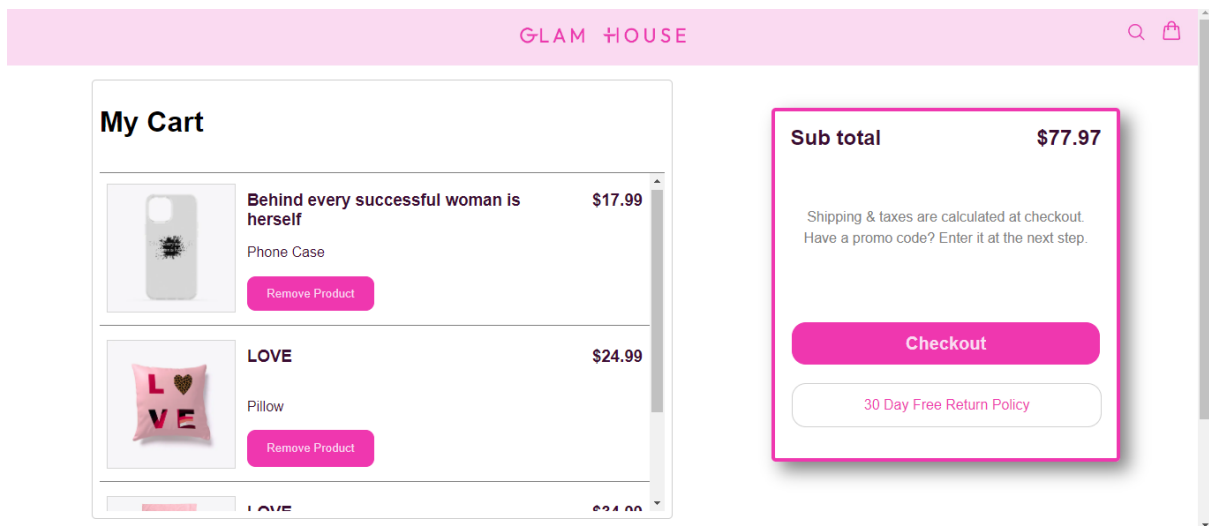
        type: '',
        price: '',
        url: ''
    },
    created() {
        this.productName = this.$route.params.productName || '';
        this.type = this.$route.query.type || '';
        this.price = this.$route.query.price || '';
        this.url = this.$route.query.url || '';
    },
    methods: {
        addToCart() {
            const product = {
                productName: this.productName,
                type: this.type,
                price: this.price,
                url: this.url
            };
            let productsInCart =
JSON.parse(localStorage.getItem('productsInCart')) || [];
            // se zima postoeckata niza ako ja ima vo LS ili se kreira nova
prazna

            productsInCart.push(product);
            localStorage.setItem('productsInCart',
JSON.stringify(productsInCart));
            //zacuvuvanje na apdejtirana niza vo LS

            alert('Product added to cart!');
        }
    }
}
</script>. . . CSS styles

```

Последната страна што ќе му се прикаже на корисникот пред да продолжи со плаќање преку банка на копчето “Checkout” (empty link), е компонентата `ShoppingCartPage.vue` (прикажана на сликата подолу) со додадени 3 производи – кои се земаат од local storage преку повлекување на низата со производи за купување, за време на `created()` lifecycle hook-от на компонентата – ако немаше производи во купувачката кошничка ќе се креираше нова празна низа. Со клик на копчето “Remove Product” на соодветниот производ, истиот ќе се избрише од купувачката кошничка. Кај Sub total секогаш се прикажува вкупната сума што треба да ја плати корисникот за сите производи.



Кодот на ShoppingCartPage.vue:

```
<template>
  <div id="wholePage">
    <div id="rozovoMeni">
      <RouterLink to="/">
        
      </RouterLink>
      <SearchAndCartComponent id="sc"/>
    </div>
    <div id="middleDiv">
      <div id="myCartDiv">
        <h1>My Cart</h1>
        <div id="divForComponents">
          <div v-for="(product, index) in productsInCart"
:key="index" id="komponenta">
            
            <div id="imeITip">
              <h3 id="ime">{{ product.productName }}</h3>
              <p id="tip">{{ product.type }}</p>
              <button id="remove"
@click="removeProduct(index)">Remove Product</button>
            </div>
            <h3 id="cena">{{ product.price }}</h3>
          </div>
        </div>
      </div>
      <div id="checkoutDiv">
        <div class="sttp">
          <h2 class="subTotal">Sub total</h2>
          <h2 class="totalPrice">{{ totalPrice }}</h2>
        </div>
        <p class="desc">Shipping & taxes are calculated at checkout.
Have a promo code? Enter it at the next step.</p>
      </div>
    </div>
  </div>
</template>
```

```

        <button class="checkout"><h2>Checkout</h2></button>
        <p class="return">30 Day Free Return Policy</p>
    </div>
</div>
<FooterComponent/>
</div>
</template>

<script>
export default {
  data() {
    return {
      productsInCart: [],
      productName: '',
      type: '',
      price: '',
      url: '',
      vkupno: '0',
      product: {
        productName: '',
        type: '',
        price: '',
        url: ''
      }
    };
  },
  created() {
    const savedProducts = localStorage.getItem('productsInCart'); //se
    zema nizata od LS
    if (savedProducts) {
      this.productsInCart = JSON.parse(savedProducts);
      // se vrakjaat name type url i price da ne se veke string kako vo
      JSON-ot i nizata si e so objekti normalna
    } else {
      this.productsInCart = [];
    }
  },
  computed: {
    totalPrice() {
      this.vkupno = this.productsInCart.reduce((total, product) => {
        const productPrice = parseFloat(product.price.replace('$',
        ''));
        //go trga $ od cenata i go pretvara stringot vo broj
        return total + productPrice;
      }, 0);
      return this.vkupno.toFixed(2); // za da e so 2 decimali sigurno
    },
  },

```

```
    },  
    methods: {  
      removeProduct(index) {  
        this.productsInCart.splice(index, 1); // se brise proizvodot na  
konkretniot indeks  
        localStorage.setItem('productsInCart',  
JSON.stringify(this.productsInCart)); // apdejt na local storage  
      }  
    }  
  }  
}  
</script>. . . CSS styles
```

Изработи:

Кристина Неделкова 171187