



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У
НОВОМ САДУ



Ања Пушкаш, PR 68/2019

**Веб продавница са проширењем начином плаћања
PayPal и приказом непреузетих пакета на мапи
користећи OpenStreetMap**

ПРОЈЕКАТ

- Примењено софтверско инжењерство (ОАС) -

Нови Сад, август 2023

САДРЖАЈ

1. ОПИС РЕШАВАНОГ ПРОБЛЕМА
2. ОПИС КОРИШЋЕНИХ ТЕХНОЛОГИЈА И АЛАТА
3. ОПИС РЕШЕЊА ПРОБЛЕМА
4. ПРЕДЛОЗИ ЗА ДАЉА УСАВРШАВАЊА
5. ЛИТЕРАТУРА

ОПИС РЕШАВАНОГ ПРОБЛЕМА

У оквиру овог пројекта реализована је веб апликација која има задатак да омогући онлине куповину и продају производа, у склопу које постоји могућност PayPal плаћања као и приказ непреузетих пакета на мапи помоћу OpenStreetMap.

Апликација се састоји од три врсте корисника - администратори, продавци и купци. Корисницима који нису пријављени, апликација пружа само основне функционалности као што су пријава и регистрација.

Приликом регистрације, корисник може одабрати пријаву на систем путем свог Google налога у којој аутоматски добија улогу купца или класичну регистрацију уносећи своје валидне податке у којој између осталог може да одабере улогу купца или продавца [1]. Обезбеђена је додатна заштита, тако што корисник мора да унесе два пута лозинку приликом регистрације.

Администратори се уносе директно у базу и не могу да се региструју на претходно описане начине. Регистровани корисници се затим могу пријавити на апликацију уношењем свог *username-a* и лозинке.

Администратори имају увид у све регистроване продавце, као и могућност да захтев за верификацију истих прихвате или одбију. Када то учине, поред имена продавца постоји информација да ли је верификован или није. Такође, имају увид у све направљене поруџбине као и њихов статус.

Продавци, након што су верификовани од стране администратора, на свој *e-mail* добијају обавештење о својој одобреној верификацији. Након тога, могу да се пријаве на систем. Они имају могућност додавања производа, као и њихово ажурирање и брисање. Имају увид у тек пристигле поруџбине са њиховим детаљима, које могу да прихвате или одбију. На мапи могу да виде непреузете пакете(на тачној адреси купца) и у случају прихватања, почиње одбројавање до доставе. Такође имају увид у историју већ достављених поруџбина.

Купци имају могућност да виде списак свих доступних производа од свих регистрованих продаваца који их продају. Приликом поручивања, имају могућност да одаберу начин плаћања готовином приликом поузећа, кредитном картицом или плаћање путем *Paypal-a* [2]. Након успешне куповине, могу отказати своју поруџбину у року од сат времена. Имају увид у историју свих поручених поруџбина са њиховим детаљима, осим поруџбина које су отказане. Након што продавац одобри поруџбину, почиње одбројавање до доставе коју купац такође може да види.

Свим корисницима који су регистровани доступна је могућност прегледа и измена профила, међутим тип корисника се не може мењати. У зависности од типа корисника, сваки корисник има свој *dashboard* који је у складу са њиховим типом и могућностима које он пружа.

ОПИС КОРИШЋЕНИХ АЛАТА И ТЕХНОЛОГИЈА

Visual Studio 2022 – бесплатно интегрисано развојно окружење компаније Microsoft које пружа богат скуп алата и функционалности. Пружа разноврсну подршку за многе језике укључујући C#, C++, Python као и за различите платформе попут Windows, Adnroid, IOS, Linux. Омогућава програмерима да лакше креирају, тестирају, дебагују и оптимизују свој код. У овој апликацији је коришћен за развој backend-a [3].

Visual Studio Code - бесплатни интерфејс, отвореног кода који служи за равој различитих врста апликација. Укључује развојне функције, као што су означавање синтаксе, подударање заграда и има подршку за мноштво језика као што су TypeScript, JavaScript, JSON, HTML, CSS, C#, C++ и још многе друге. У овој апликацији је коришћен за развој frontend-a [4].

Node.js - је вишенаменско JavaScript радно окружење отвореног кода за извршавање JavaScript-а на серверској страни. Node.js омогућава да се JavaScript користи за скрипте на серверској страни које омогућавају да се садржај динамичних веб страница генерише на серверу пре него што се пошаље до веб прегледача корисника. Због тога је Node.js постао један од основних елемената парадигме "ЈаваСкрипт свуда" јер омогућава развој веб апликација у једном програмском језику, без потребе да се за скрипте на серверској страни користи различит програмски језик [5].

Automapper – је конвенцијски базиран мапер који на основу назива поља у класама може да намапира један објекат на други. Како би знао да претвори један тип објекта у други, називи поља морају бити исти као и типови поља [6].

SQL Server Managment Studio 2019 - релациона база података која се користи у различите сврхе. Он може да се користи у различите сврхе као што су пословна интелигенција или за складиштење података. Ипак, SQL Server се најчешће користи као крајња компонента за смештање података, а традиционалне Web апликације често користе SQL Server за смештање апликативних података. Подаци се складиште у виду табела које су међусобно повезане. Циљ је да се кориснику олакша управљање базом. SQL је стандардни језик који се користи за комуникацију са базама података [7].

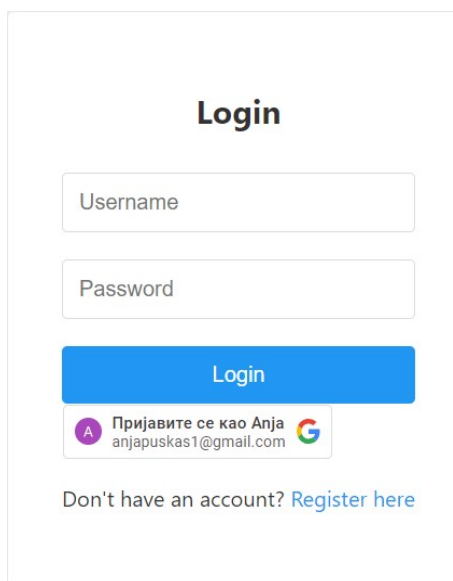
React - JavaScript библиотека за развој корисничког интерфејса у веб апликацијама. Она се користи за креирање интерактивних веб апликација и компоненти које реагују на промене података и омогућавају ефикасно управљање приказом садржаја. Користи виртуелни DOM за брзе и ефикасне промене на интерфејсу, што резултира бољом перформансом. Чест је избор за модерно програмирање на вебу. React је коришћен за израду frontend-a [8].

ASP.NET Core 6.0 Web API - бесплатна платформа отвореног типа за креирање многих врста апликација, као што су: веб, клауд, мобилне, десктоп, конзолне апликације. Одликује се високом перформансом, универзалношћу, флексибилношћу и подршком за различите врсте апликација и сервиса. Коришћен је за backend ове апликације [9] .

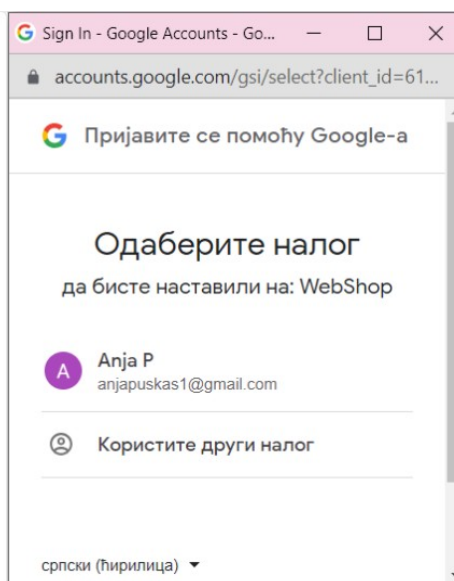
ОПИС РЕШЕЊА ПРОБЛЕМА

Непријављени корисник

Када се апликација покрене, почетна страница апликације јесте пријава на систем. Корисник се пријављује на систем тако што попуни поља *username* и *password*. Непријављени корисник има могућност пријаве на систем путем линка за регистрацију(слика 1.1) или путем свог *Google* налога(слика 1.2). Уколико се улогује путем *Google* налога, аутоматски се пријављује на систем као купац.



Слика 1.1 - регистрација



Слика 1.2 - Google налог

Уколико корисник одабере класичан начин регистрације, мора валидно попунити поља. Као што је приказано на слици 1.3, корисник мора да унесе све податке који се од њега траже. *Username* и *email* морају бити јединствени, јер ће у супротном добити повратну инфомацију да корисничко име или *email* већ постоји у систему. Лозинку мора да унесе два пута, због додатне провере и смањеног процента прављења грешке. Она се криптује помоћу *Bcrypt.Net*, и приликом логовања се пореди са криптованом вредношћу сачуваној у бази, да би обезбедила кориснику успешно логовање. [10] Слика корисник није у обавези да постави одмах приликом регистрације, али има и ту могућност. Додатно обавезно поље које корисник мора да попуни приликом регистрације је да одабере своју улогу купца или продавца, коју накнадно не може да промени.

На слици 1.4 приказана је имплементација кода за методу логовања. Додатна провера приликом логовања која постоји, јесте да продавац мора бити верификован од стране администратора да би уопште могао да се улогује на систем са својим креденцијалима. Уколико је корисник унео валидне податке, креира се токен са својим *claim*-овима који су потребни за ауторизацију и аутентификацију корисника, и трају 20минута.

Registration Form

Username *

First Name *

Last Name *

Email *

Password *

Repeat Password *

Address

Birthday

User Role

Seller

Buyer

Picture: Choose File No file chosen

REGISTER

Already have an account? [Login here](#)

Слика 1.3 - Форма за регистрацију корисника

```
public async Task<LoginResultDTO> login(LoginAttemptDTO login)
{
    var users = await _repository._userRepository.GetAll();
    User? user = users.FirstOrDefault(u => u.Username == login.Username);
    if (user == null)
    {
        throw new CredentialsException("User does not exist");
    }
    if (user.UserRole == UserRole.SELLER && user.UserStatus == UserStatus.ON_HOLD)
    {
        throw new CredentialsException("User is not verified. Please wait from verification from Admin.");
    }
    if (user.UserRole == UserRole.SELLER && user.UserStatus == UserStatus.REJECTED)
    {
        throw new CredentialsException("User is rejected and can not log in.");
    }
    if (!BCrypt.Net.BCrypt.Verify(login.Password, user.Password))
    {
        throw new CredentialsException("Incorrect login credentials!");
    }
    List<Claim> claims = new List<Claim>
    {
        new Claim("id", user.Id.ToString()),
        new Claim(ClaimTypes.NameIdentifier, user.Id.ToString()),
        new Claim(ClaimTypes.Name, user.Username),
        new Claim(ClaimTypes.Role, user.UserRole.ToString())
    };
    SymmetricSecurityKey secretKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(_configuration.GetSection("Jwt")["secret"]));
    var signinCredentials = new SigningCredentials(secretKey, SecurityAlgorithms.HmacSha256);
    var tokenOptions = new JwtSecurityToken(
        _configuration["Jwt:Issuer"],
        _configuration["Jwt:Audience"],
        claims: claims,
        expires: DateTime.Now.AddMinutes(20),
        signingCredentials: signinCredentials
    );
    string token = new JwtSecurityTokenHandler().WriteToken(tokenOptions);
    LoginResultDTO loginResult = _mapper.Map<LoginResultDTO>(user);
    loginResult.Token = token;
    loginResult.UserStatus = Enum.GetName(typeof(UserStatus), user.UserStatus);
    return loginResult;
}
```


Слика 1.4 - Имплементација кода за Login

Пријављени корисник

Сваки пријављени корисник након успешне регистрације и логовања, се преусмерава на страницу апликације која има свој *dashboard* сходно томе која је улога додељена том конкретном кориснику. Једина заједничка ствар свим корисницима јесте опција у којој могу да прегледају и модификују свој профил. Име, презиме, адреса и слика су неки од понуђених поља која су склона модификацијама, и наравно може да се измени само једно поље које корисник одабере. Корисник не може да промени свој *username* и *email*, као ни улогу и статус. (Слика 2.1)

Наравно, сваки корисник такође има навигациони бар са опцијом *Shop* која води на почетну страницу апликације, и дугме за *Logout*.

Profile



First Name

seller

Last Name

sell

Address


Address

Status

VERIFIED

Birthday

23/12/1998



Change Picture:

Choose File

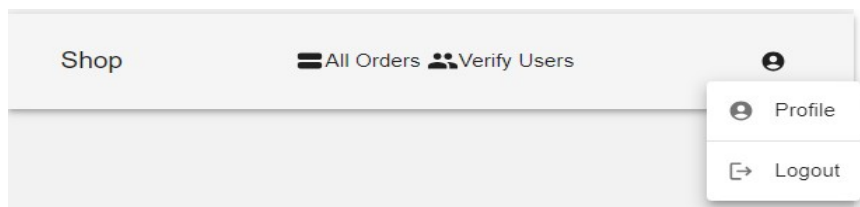
No file chosen

UPDATE

Слика 2.1 - преглед и измена профила

Администратор

Када се корисник са улогом администратора пријави на систем, поред претходно описаних опција заједничких за све кориснике, он додатно у навигационом бару има опције за преглед свих поруџбина као и увид у све регистроване продавце чији захтев за верификацију има могућност да прихвати или одбије (слика 3.1).



Слика 3.1 - навигациони бар администратора

Приликом регистрације продавца, администратор је тај који ће да прихвати или одбије његову верификацију. На слици 3.2 можемо видети списак свих регистрованих продаваца, који поред основних информација садрже и информацију о статусу тог продавца. Статус може бити *verified*, у случају прихватања, *rejected* у случају одбијања и *on_hold* у случају чекања на верификацију.

Тек након што администратор кликне дугме за одобравање верификације, продавцу на *email* стиже обавештење о успешној верификацији и он може даље да се улогује на систем и користи апликацију.

У супротном, приликом логовања ће му бити назначено да његов профил није верификован и да не може да приступи апликацији.

Shop

All Orders

Verify Users

Name	Username	Email	Address	Date of Birth	User Status	Actions
Nikola Petrovic	seller	s@gmail.com	Address1	1998.12.23 21:00:00	VERIFIED	<div>VERIFY</div> <div>REJECT</div>
Petrovic Petar	seller2	s2@gmail.com	Adress2	1999.12.31 22:00:00	VERIFIED	<div>VERIFY</div> <div>REJECT</div>
s s	seller3	seller3@gmail.com	Adress	1995.10.09 23:00:00	REJECTED	<div>VERIFY</div> <div>REJECT</div>
Anja Puskas	anja	anjapuskas@hotmail.com	Adress3	2000.05.23 22:00:00	ON_HOLD	<div>VERIFY</div> <div>REJECT</div>

Слика 3.2 - листа свих продаваца и њихова верификација

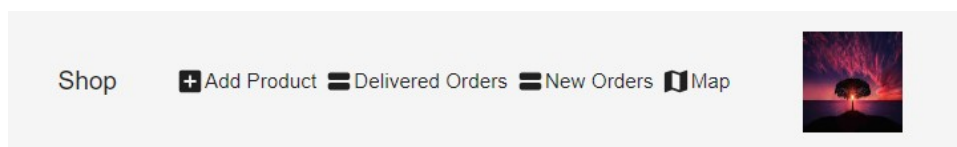
Администратори такође имају увид у историју свих поруџбина. Поруџбине имају информације о адреси купца, цени, датуму креирања поруџбине као и датуму када се очекује достава исте. Статус поруџбине означава да ли је достављена, тек поручена и није још увек достављена или је отказана. (Слика 3.3)

Address	Created Date	Delivery Date	Status	Price
Lukijana Musickog 43	2023.08.23 17:59:59	2023.08.24 07:00:29	DELIVERED	15
Kisacka 6	2023.08.24 22:55:09	2023.08.25 21:55:45	DELIVERED	15
6	2023.08.26 22:05:34	2023.08.28 12:00:03	ORDERED	15
adresa1	2023.08.27 23:51:23	2023.08.28 07:04:23	ORDERED	30
Adresa2	2023.08.27 23:55:03		CANCELLED	25

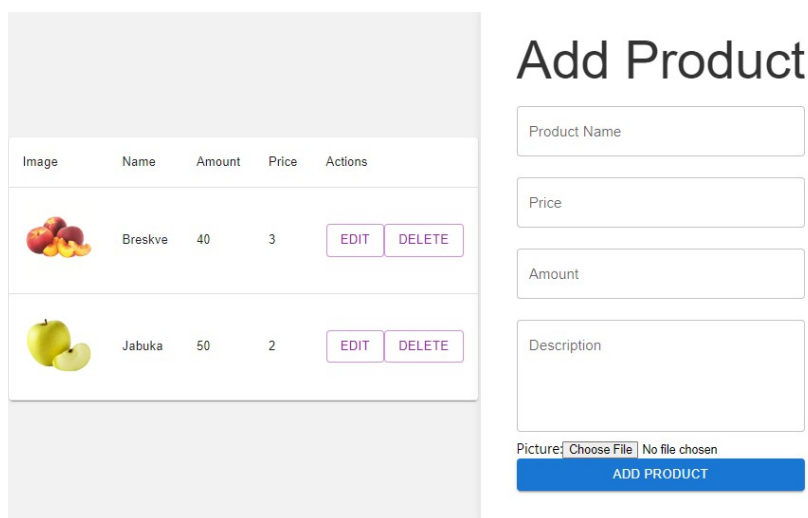
Слика 3.3 - табела са прегледом свих поруџбина

Продавац

Након што администратор одобри верификацију продавца, он може несметано да се улогује и користи апликацију. Продавац има могућност да дода нове производе које жели да прода кликом на *Add Product*, да их модификује као и да их обрише са стања (Слика 4.2).



Слика 4.1- навигациони бар продавца



Слика 4.2 - преглед и додавање артикала

Продавац када додаје нови производ мора попунити сва поља која се од њега захтевају као што су име производа, цена, количина, опис производа као и његову слику. Када жели да модификује производ, може да кликне на дугме *Edit*, а када жели да га обрише то чини кликом на дугме *Delete*.

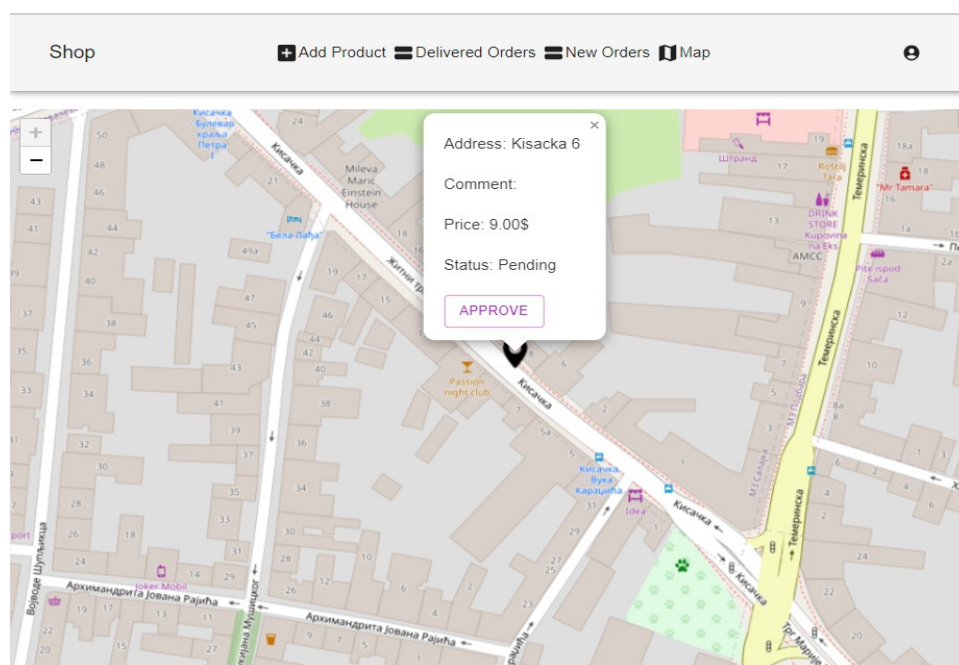
Кликом на *New Orders*, продавац има увид у тек пристигле поруџбине и њихове детаље. Да би одбројавање до доставе почело, продавац мора да одобри поруџбину(слика 4.3).

User Id	Address	Created Date	Price	Approved	Actions	Delivery Time
10006	adresa2	2023.08.28 20:03:07	3	Yes	DETAILS	11:57:23
10007	Adresa5	2023.08.28 20:05:09	6	No	APPROVE DETAILS	0:0:0
10006	Kisacka 6	2023.08.28 20:10:27	9	No	APPROVE DETAILS	0:0:0

Слика 4.3 - преглед нових поруџбина

Продавац има могућност да одобри поруџбину на страници где има увид у све пристигле поруџбине, као и на мапи где може тачно да види адресу где треба доставити пакет(слика 4.4). Уколико је купац отказао своје поруџбину, продавац неће моћи да је види.

На слици 4.5 можемо да видимо да се кликом на *Delivered Orders* приказују све испоручене поруџбине тог продавца са свим потребним детаљима, као и да се кликом на *Details* отвара нови прозор у коме се могу видети додатни детаљи поруџбине(слика 4.6).




Слика 4.4 - мапа поруџбина

Delivered Orders

User Id	Address	Created Date	Delivery Date	Price	Actions
10000	adresa1	2023.08.27 23:51:23	2023.08.28 07:04:23	3	DETAILS

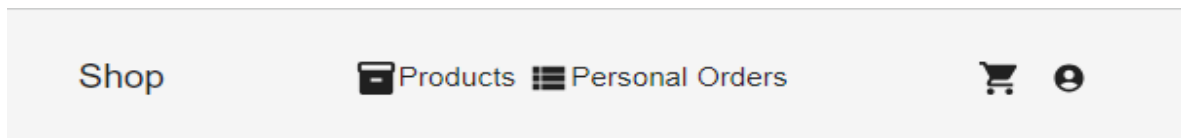
Слика 4.5 - листа свих достављених поруџбина

Image	Name	Amount	Price
	Breskve	1	3

Слика 4.6 - детаљи поруџбине

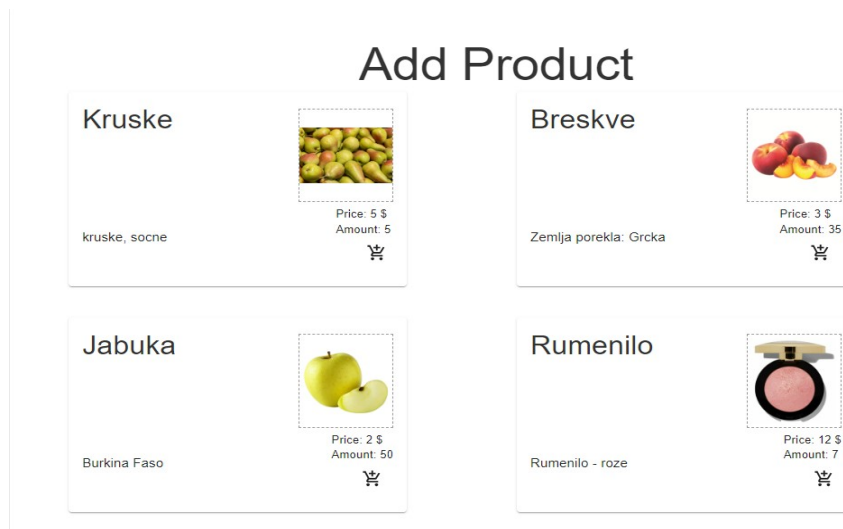
Купац

Купац има себи својстен навигациони бар који можемо видети на слици 5.1.



Слика 5.1 - навигациони бар купца

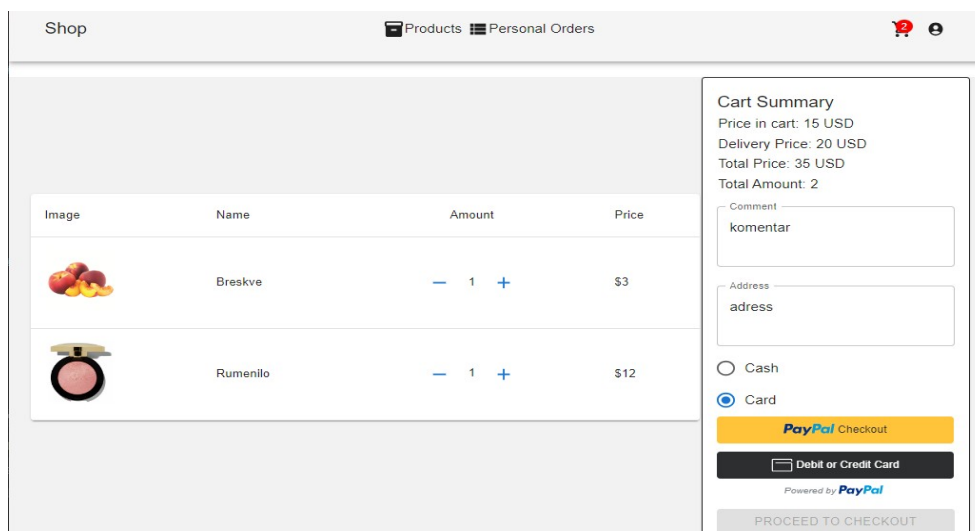
Кликом на дугме *Products* купац има увид у све доступне производе у продавници. Сваки производ има своју слику, име, опис као и цену и информацију колико тачно тих артикала има на стању. Купац може додати производе у корпу кликом на корпицу испод тог производа, али не може додати артикал којем је количина на стању 0, као већу количину од доступне(слика 5.2).



Слика 5.2 - сви доступни производи у продавници

На слици 5.3 приказан је изглед корпе купца након додавања производа у корпу, где купац може накнадно да дода или уклони жељени артикал који је већ у корпи помоћу пречица „+“ и „-“. Такође, има информацију о цени производа, а ако је додао артикле од различитих продаваца, сходно томе ће му се рачунати и цена доставе (сваки продавац има своју цену доставе). Поља за адресу доставе као и додатни коментар се морају попунити, да би поруџбина била успешна, у супротном купац ће добити информацију о грешци. Могућности за плаћање има три. Прва јесте плаћање готовином приликом доставе пакета на адресу, друга јесте кредитном картицом, а трећа је плаћање путем *Paypal-a*.

Уколико купац одабере плаћање путем *Paypal-a*, отвориће му се додатни прозор за унос неопходних података, и ако је плаћање успешно, купац ће добити обавештење о успешно порученој куповини. Имплементација кода приказана је на слици 5.4.



Слика 5.3 - изглед корпе купца

```

191 <FormControlLabel value='Card' control={radio} label='Card' />
192 </RadioGroup>
193 {paymentMethod === 'Card' && products.length !== 0 && address.length !== 0 && (
194   <PayPalScriptProvider
195     options={{
196       currency: 'USD',
197       clientId: process.env.REACT_APP_PAYPAL_CLIENT_ID
198     }}
199   >
200     <PayPalButtons
201       style={{ label: 'checkout' }}
202       createOrder={async (data, actions) => {
203         return actions.order
204           .create({
205             purchase_units: [
206               {
207                 amount: {
208                   value: price.toFixed(2)
209                     .toString(),
210                   currency_code: 'USD'
211                 }
212               }
213             ]
214           })
215           .then((result) => {
216             return result
217           })
218           .catch((error) => {
219             return Promise.reject('')
220           })
221       })
222       onApprove={async (data, actions) => {
223         return actions.order
224           ?.capture()
225           .then(() => {
226             toast.success('Payment successful', {
227               position: "top-center",
228               autoClose: 2500,
229               closeOnClick: true,
230               pauseOnHover: false,
231             });
232             orderFromCart();
233           })
234           .catch(() => {
235             //
236           })
237       })
238     </PayPalScriptProvider>

```

Слика 5.4 - имплементација кода за плаћање путем PayPal-a

Након што је купац завршио са куповином, он има увид у све своје поруџбине кликом на дугме *Personal Orders*. Купац има могућност да откаже своју поруџбину првих сат времена од поручивања, након истека тог времена, опција отказивања поруџбине ће му бити онемогућена. Тек када продавац потврди поруџбину, купцу почиње одбројавање времена до доставе. Адреса, датум креирања поруџбине, цена исте, као и статус поруџбине(отказана, достављена или поручена) само су неке од детаља које купац може да види на овој страници(слика 5.5).

Address	Created Date	Order Status	Price	Action	Delivery Date
Adresa2	2023.08.27 23:55:03	CANCELLED	25	<button>CANCEL</button>	
adresa2	2023.08.28 20:03:07	ORDERED	50	<button>CANCEL</button>	11:12:34
Kisacka 6	2023.08.28 20:10:27	ORDERED	19	<button>CANCEL</button>	0:0:0

Слика 5.5 - детаљан преглед поруџбина

Додатно, на слици 5.6 можемо видети имплементацију кода за поручивање производа. *Default-no* поруџбина има статус *false*, а тек када је продавац одобри постаје *true*. Количина производа коју купац додаје у корпу мора бити на стању, у супротном купац добија грешку о недовољно количине на стању за тај исти производ. Цена доставе сваког продавца износи *10USD*, те уколико купац поручи производе од два различита продавца, његова укупна цена доставе износиће *20USD*.

```
public async Task<bool> addOrder(CreateOrderDTO createOrderDTO)
{
    Order order = _mapper.Map<Order>(createOrderDTO);
    order.Buyer = await _userService.getUser(order.UserId);
    order.Created = DateTime.Now;
    order.OrderProducts = new List<OrderProduct>();
    order.Approved = false;
    List<long> differentSellerIds = new List<long>();

    foreach (ProductDTO product in createOrderDTO.Products)
    {
        OrderProduct orderProduct = new OrderProduct();
        orderProduct.ProductId = product.Id;
        orderProduct.OrderId = order.Id;
        orderProduct.Order = order;
        orderProduct.Product = await _productService.getProduct(product.Id);
        orderProduct.Price = product.Price;
        orderProduct.Amount = product.Amount;

        if (orderProduct.Product.Amount < product.Amount)
        {
            throw new Exception("Nedovoljno kolicine za proizvod " + product.Name);
        }
        orderProduct.Product.Amount = orderProduct.Product.Amount - product.Amount;
        if (!differentSellerIds.Contains(orderProduct.Product.SellerId))
        {
            differentSellerIds.Add(orderProduct.Product.SellerId);
        }

        order.OrderProducts.Add(orderProduct);
    }

    order.Price = order.Price + differentSellerIds.Count * 10.0;
    await _repository._orderRepository.Insert(order);
    await _repository.SaveChanges();

    return true;
}
```

Слика 5.6 - имплементација кода за креирање поруџбине

ПРЕДЛОЗИ ЗА ДАЉА УСАВРШАВАЊА

Већина функционалности ове апликације је успешно реализована. Међутим, у циљу побољшања саме апликације требало би је проширити додатним функционалностима.

- Да би се корисницима пружила једноставнија куповина, требало би увести категоризацију производа која би помогла купцима да лакше и брже нађу жељени артикал.
- Корисницима би се такође могла увести опција персонализације, тј. да се омогући корисницима да направе своје листе жеља или сачувају омиљене производе.
- Такође, у циљу побољшања задовољства купаца требало би обезбедити подршку путем чета уживо или путем *email-a*, за сва потенцијална питања које купци имају.
- Опција често постављена питања, такође би се могла имплементирати у циљу смањења потребе за контактирањем подршке.
- *Live tracking* или праћење поруџбине је функционалност која би била потребна купцу да би тачно у одређеном моменту знао где му се налази поруџбина у циљу лакшег преузимања пакета.
- Рецензије и коментари на производе као и на продавце била би корисна информација свим будућим купцима који желе да поруче одређене производе или производе одређеног продавца.
- Филтрирања, сортирања, као и брзе претраге су још неке од опција које би помогле како купцима, продавцима тако и администраторима за лакше сналажење на самој апликацији.
- Још једна мана апликације јесте та што ће поруџбина бити прихваћена иако је само један од продавца из исте потврдио поруџбину. Оно што би требало додати јесте да поруџбина буде прихваћена тек оног момента када сви продавци чији је производ у датој поруџбини потврде исту.
- Такође, у циљу проширења апликације на тржишту, могла би се имплементирати опција дељења производа путем линка као и друштвених мрежа.
- Продавцима би се могла имплементирати опција за снижење свог атрикла, а купцима опција скупљања бодова и додатних погодности за јефтинију куповину.
- Увођењем могућности кориснику да одабере језик на ком ће бити цела апликација, такође би била погодна за све кориснике апликације. Тренутно је апликација на енглеском језику.

ЛИТЕРАТУРА

- [1] <https://console.cloud.google.com/welcome>
- [2] <https://developer.paypal.com/tools/sandbox/>
- [3] <https://docs.microsoft.com/en-us/visualstudio/windows/?view=vs-2022>
- [4] <https://code.visualstudio.com/docs>
- [5] <https://sr.wikipedia.org/wiki/Node.js>
- [6] <https://docs.automapper.org/en/latest/Getting-started.html>
- [7] <https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>
- [8] <https://react.dev/learn>
- [9] <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0>
- [10] <https://learn.microsoft.com/en-us/answers/questions/830417/verify-passwords-with-bcrypt-net>