



PROJEKTOVANJE I RAZVOJ VEB APLIKACIJE ZA PODRŠKU POSLOVANJA
POLJOPRIVREDNOG GAZDINSTVA

DESIGN AND DEVELOPMENT OF A WEB APPLICATION TO SUPPORT
AGRICULTURAL HOUSEHOLD

Milica Pavlović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – INFORMACIONE TEHNOLOGIJE

Kratka sadržaj – Istraživačkim delom rada dat je uvid u probleme sa kojima su se susretala poljoprivredna gazdinstva za vreme pandemije COVID-19. U nastavku rada prikazano je moguće rešenje problema koje je imao veliki broj poljoprivrednih gazdinstava, a to je veb aplikacija. Detaljno su opisane korišćene tehnologije, kao i postupak implementacije svih delova aplikacije. Na kraju je prikazan primer korišćenja aplikativnog rešenja.

Ključne reči: Veb aplikacija, Poljoprivredno gazdinstvo

Abstract – The research part of the master thesis gave an insight into the problems faced by agricultural households during the COVID-19 pandemic. In this thesis a possible solution to the problem faced by large number of agricultural households is presented, which is a web application. The technologies used are described in detail, as well as the implementation procedure of all parts of the application. At the end, an example of using the application solution is shown.

Keywords: Web application, Agricultural household

1. UVOD

Izbijanje bolesti i globalne pandemije su najveća pretnja za održivost ljudskog postojanja. Čovečanstvo je svedočilo mnogim pandemijama tokom ljudske istorije, koje su ubile milione ljudi i oštetile globalnu ekonomiju i politiku [1]. Čovečanstvo se susretalo sa brojnim pandemijama, poput španske gripe, svinjske gripe, ebola i slično [2]. **Pandemija** je epidemija zarazne bolesti koja se širi kroz populaciju ljudi jedne veće oblasti, kontinenta ili celog sveta. Prema Svetskoj zdravstvenoj organizaciji, o pandemiji se može govoriti kada su ispunjena tri uslova:

- u nekoj populaciji se pojavilo do tada nepoznato oboljenje,
- uzročnici inficiraju ljude i izazivaju teške posledice,
- uzročnici bolesti se šire brzo i zadržavaju se među ljudima [3].

Bolest mora biti zarazna kako bi pokrenula pandemiju. Tokom istorije su zabeleženi slučajevi pandemija kuge, kolere, tifusa, gripa, side, boginja, tuberkuloze, lepre, malarije, žute groznice itd. Jedna od najsmrtonosnijih je bila pandemija španske groznice.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bila dr Teodora Lolić, docent.

1.1. Motivacija za istraživanje

U Kini, tačnije u Vuhanu, 31. decembra 2019. godine, prijavljeni su prvi slučajevi infekcije novim korona virusom, koji uzrokuje bolest koja je sad poznata kao COVID-19, različita od SARS-CoV i MERS-Cov [2].

Iz prethodnih pandemija koje je svet iskusio, pokazalo se da karantin i panika imaju uticaj na ljudske aktivnosti i ekonomski rast, ali ovaj efekat se javlja i u poljoprivrednim delatnostima. Kada dođe do izbijanja zarazne bolesti, dolazi i do povećanja gladi i neuhranjenosti.

Situacija se pogoršava kako bolest napreduje, čineći ograničenja kretanja sve strožijima, uzrokujući nedostatak radne snage za žetvu ili poteškoće poljoprivrednicima da svoje proizvode plasiraju na tržište.

Poljoprivreda je jedan od najvažnijih grana u ljudskom razvoju. U okviru ovog rada je pre svega istraženo kako su na poljoprivredu uticali događaji povezani sa bolešću COVID-19, a zatim je predstavljen predlog rešenja najčešćeg problema sa kojim su se poljoprivrednici susretali [2].

Shvatajući ozbiljnost situacije svaka zemlja je preduzela potrebne korake u borbi protiv pandemije, uglavnom nefarmaceutskim merama koje uključuju socijalno distanciranje i samoizolaciju. Pored toga, ograničenje u putovanju i trgovini se obavlja u većini zemalja kako bi se ograničilo širenje virusa. Sve ove borbe protiv pandemije imale su veliki uticaj na glavne ekonomske sektore kao što je poljoprivreda. Poljoprivreda je najvažniji privredni sektor koji podržava sigurnost hrane i ljudski razvoj.

2. METODOLOGIJA ISTRAŽIVANJA

Pandemija koja je zadesila svet 2020. godine imala je ogroman uticaj na gotovo svaku granu privrede. Najnegativniji uticaj je definitivno bio na turizam i ugostiteljstvo. Cilj ovog istraživanja jeste detaljniji uvid u uticaj pandemije na poslovanje poljoprivrednih gazdinstava, odnosno na koje se sve načine i u kojim delovima poslovanja osetio uticaj ove krize nezapamćenih razmera.

U ovom poglavlju, pre svega je predstavljen upitnik, sa svim pitanjima koja su važna za ovo istraživanje.

Upitnik je plasiran najviše putem društvenih mreža, jer gotovo svako poljoprivredno gazdinstvo ima svoju Facebook ili Instagram stranicu. Takođe, na internetu je dostupan registar poljoprivrednih gazdinstava koja izlažu svoje proizvode na jednoj Beogradskoj pijaci. Pomenuti registar je između ostalog nudio i e-mail adrese gazdinstava pomoću kojih kupci mogu da stupe u kontakt sa njima, pa su upitnici poslani i putem mejla. Odziv gazdinstava je bio odličan, ali s obzirom da je za

istraživanje sasvim dovoljno oko 50 odgovora, zaustavljen je dalji prijem, kako bi se odmah pristupilo analizi prikupljenih odgovora za potrebe ovog rada. U nastavku su predstavljeni rezultati istraživanja i na sve odgovore je dat kratak komentar.

1.2. Zaključak istraživanja i predlog rešenja

Epidemija je u velikom broju gazdinstava dovela do osetnog smanjenja zarade i potražnje u poređenju sa prethodnom godinom, dok su određena gazdinstva baš u vreme epidemije doživela procvat. Smanjenjem zarade, smanjivao se obim rada, kao i potreba za radnicima. Nažalost, pojedina gazdinstva morala su da otpuštaju radnike, a dešavalo se i da nisu bili u mogućnosti da ih adekvatno isplate. Još jedan problem bila je i nemogućnost dolaska radnika na posao, u nekim slučajevima zbog ograničenja kretanja, a nekad i zbog bolovanja zaposlenih od virusa COVID-19. Samim tim, određeni radovi su se odvijali sporo ili su u potpunosti stajali, zbog čega su vlasnici gazdinstava trpeli gubitke upravo zbog nemogućnosti proizvodnje na uobičajen način.

Jedan od najvećih problema bio je plasman robe u uslovima pandemije, otežavajuća okolnost bilo je ograničenje kretanja, zatvaranje tržnica i opšti rizik od zaraze prodavaca i kupaca.

U ovakvim vremenima, kada je ugroženo ono najvrednije kod ljudi, a to je zdravlje, najvažnije je preduzeti sve potrebne mere i stvarati nove ideje kako bi se smanjio rizik od širenja bolesti.

Kao što se može zaključiti iz prethodnog istraživanja, najveći broj poljoprivrednih gazdinstava se okrenuo porudžbinama putem telefona. Taj način je sigurno bezbedniji od ličnog dolaska u gazdinstvo ili na pijacu, ali je veoma nepraktičan. Ukoliko veći broj ljudi zove u isto vreme, linija može dugo biti zauzeta, što bi vremenom dovelo do dugog čekanja na poručivanje, a samim tim i do nezadovoljstva kupaca. Takođe, za primanje velikog broja porudžbina putem poziva u toku dana, gazdinstva bi morala da angažuju radnika. Još jedan nedostatak je nemogućnost da se kupcima da uvid u proizvode, kao što je moguće putem kataloga. U ovom slučaju, kompletna ponuda proizvoda bi se nudila usmeno.

Druga vrsta prodaje kojoj su se okrenuli poljoprivrednici je putem oglašavanja na društvenim mrežama Facebook i Instagram. Prodaja preko društvenih mreža je sasvim legalan i legitiman kanal distribucije proizvoda, ukoliko su ispoštovani svi zakonski propisi. Prednost ovog načina prodaje je pre svega uvid kupaca u kompletnu ponudu prodavca putem objava odnosno fotografija na samoj stranici. Komunikacija, odnosno ugovaranje kupovine se vrši preko direktnih poruka između kupca i prodavca. Naravno i u ovom slučaju može doći do preopterećenosti stranice porukama, dugog čekanja na odgovor, te ponovo nezadovoljstva kupaca.

Alternativa može biti rešenje koje podrazumeva uvezivanje stranice na društvenoj mreži sa sopstvenom internet prodavnicom, na kojoj će se zapravo odvijati postupak e-trgovine, a koja po sadržaju i strukturi ispunjava sve uslove koje propisi nalažu vezano za elektronsku trgovinu [4]. Upravo to i jeste ideja ovog rada, iz više razloga. Uvođenje *web* prodavnice u poslovanje svakog poljoprivrednog gazdinstva bi bilo odlično i praktično rešenje, kako za kupce, tako i za poljoprivrednike.

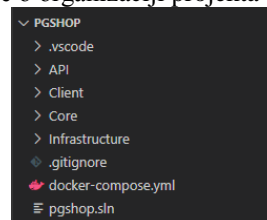
Ovaj način kupovine je odlično rešenje za vreme pandemijskih uslova. Dakle i u ovom slučaju, kupovina se obavlja potpuno bezbedno po stanovništvo, iz udobnosti njihovog doma, bez kontakta sa ljudima, odlaska na pijacu i slično. *Web* prodavnica je generalno jednostavnija za korišćenje, ponuda je razvrstana po kategorijama, preglednija je nego na društvenim mrežama, kupci se mnogo lakše snalaze. S obzirom da je većina potrošača upoznata sa *online* kupovinom, oni sami pune potrošačku korpu željenim količinama proizvoda, nije im potrebna asistencija prodavca, što samim tim smanjuje njegove obaveze prilikom kupovine. Ono što je zajedničko svim drugim načinima prodaje (telefon, Facebook, Instagram) je što se plaćanje proizvoda najčešće vrši pouzećem, direktno kuriru. Kako je u vreme pandemije poželjno izbeći sve nepotrebne kontakte sa ljudima, ideja rada je da *web* prodavnica poljoprivrednog gazdinstva ima opciju plaćanja unapred, odnosno plaćanja karticom. Ovakvom kupovinom, smanjio bi se rizik od zaraze kupca ili kurira putem kontakta prilikom preuzimanja i plaćanja porudžbine jer taj proces često uključuje i veoma prljave novčanice koje dokazano jesu prenosioci virusa i bakterija.

3. SOFTVERSKO REŠENJE

U ovom poglavlju predstavljeno je jedno moguće rešenje problema sa kojima su se poljoprivredna gazdinstva susretala tokom pandemije. U pitanju je *web* aplikacija za podršku poslovanja poljoprivrednog gazdinstva.

3.1. Backend

Na samom početku, predstavljena je struktura kompletnog projekta. U glavnom folderu ovog projekta su kreirana četiri podfoldera, koja predstavljaju četiri segmenta aplikacije. U pitanju su API, Core, Infrastructure i Client (Client služi za smeštanje *frontend* dela aplikacije). API projekat sadrži kontrolere, DTO klase, *appsettings.json* fajlove i tako dalje. Infrastructure projekat sadrži podatke poput *StoreContext*-a, repozitorijuma, servisa, migracije za *identity* i podatke, kao i konfiguracije tabela u bazi podataka. Core projekat sadrži sve entitete aplikacije, interfejse servisa i repozitorijuma, kao i specifikacije. U glavnom folderu se nalazi još nekoliko pomoćnih fajlova. Kao i obično, sastavni deo projekta je *.gitignore* fajl koji ukazuje koje datoteke i fascikle ne treba ignorisati u datom izvornom kodu. Veoma je važan *docker-compose.yml* fajl koji se koristi za konfiguraciju servisa aplikacije. Fajl *pgshop.sln* reprezentuje Visual Studio *solution* fajl koji čuva informacije o organizaciji projekta (Slika 1).



Slika 1. Struktura projekta

Kako bi se izbegle dodatne modifikacije koda i kako bi bio prilagođen promenama u *hosting model*-u koje sa sobom donosi .NET 6, u ovom projektu se koristi verzija .NET 5. Najveći deo primera koda na internetu koristi klasičan odnosno tradicionalni *hosting model*.

Prvo je prikazana implementacija API-ja aplikacije. U radu je obrađen primer jednog kontrolera, konkretno *ProductController*. U ovom slučaju, svaki kontroler kao baznu klasu nasleđuje apstraktnu klasu *ControllerBase*.

U folderu Entites, kreirane su sve potrebne *Entity* klase. S obzirom da će aplikacija biti prodavnica sa velikim brojem proizvoda, kao primer entiteta prikazana je klasa *Product*, koja je definitivno i najvažnija u ovoj aplikaciji. Ova klasa se sastoji od svih potrebnih atributa za proizvod. Objektno-relaciono mapiranje koje se koristi je bazirano na konvenciji.

U sledećem koraku, prikazan je postupak konfiguracije *database context*-a. U ASP .NET se upit ne postavlja direktno bazi, već se tipično postavlja uz pomoć objektno-relacionog mapiranja, odnosno apstrakcije baze van koda. Ovaj način donosi veliki broj pogodnosti, a najvažnija je da omogućava zamenu baze podataka koja se koristi. Prilikom razvoja je korišćena SQLite baza podataka, a u kasnijem razvoju je korišćena PostgreSQL baza.

S obzirom da je korišćeno objektno-relaciono mapiranje, proces prelaska na PostgreSQL bazu je bio veoma jednostavan. Objektno-relaciono mapiranje koje je korišćeno u projektu je *Entity framework*, koji je kreiran od strane Microsoft-a i koji odlično radi sa .NET Core. U folderu Infrastructure kreiran je folder Data i u njemu klasa *StoreContext*. *StoreContext* nasleđuje klasu *DbContext*. *DbContext* je klasa iz *Entity framework*-a.

Svaka ASP.NET Core aplikacija mora da sadrži *Startup* klasu. Kao što se po samom nazivu može zaključiti, ova klasa se izvršava pri pokretanju aplikacije.

Nakon što se izvrši komanda za migraciju, trebala bi da bude vidljiva tabela za entitet na putanji *Core/Infrastructure/Data/Migrations*. Na kraju, migracija treba da kreira tabelu za svaki entitet koji se koristi u aplikaciji.

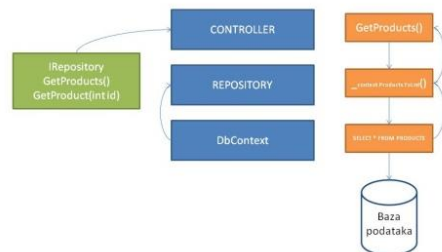
Kada je potrebno instalirati *database* server, mnogo je lakše koristiti Docker u tu svrhu, jer je instalacija veoma jednostavna. Docker omogućava pokretanje željenog servera unutar kontejnera i to je poput virtualizacije, ali bez instaliranja kompletnog virtuelnog operativnog sistema, potrebni su samo kontejneri za određene aplikacije i oni su odvojeni od operativnog sistema. Na Docker *hub*-u dostupan je *image* za PostgreSQL kao i uputstvo kako se instalira i pokreće. Mnogo je lakše dodati podatke za pokretanje PostgreSQL u postojeći *docker-compose.yml* fajl koji se trenutno koristi.

Nakon što je kreirana baza podataka i dodato nekoliko podataka u nju, ti podaci se mogu dobavljati uz pomoć kontrolera. U svim kontrolerima se koristi *The Task Asynchronous Programming Model* (TAP). Na srpskom, ovaj model se naziva model asinhronog programiranja zadataka.

Kako kompletna logika dobavljanja podataka ne bi bila smeštena u kontroleru, koristi se *repository pattern* koji omogućava da se logika koja se koristi za postavljanje upita bazi podataka uz pomoć *Entity framework*-a smesti u repozitorijum. Kontroleri pozivaju repozitorijum kako bi dobavili podatke. Potrebno je *inject*-ovati repozitorijum u kontroler. S obzirom da se *inject*-uje repozitorijum u kontroler, kontroler će znati šta je dostupno za korišćenje u repozitorijumu.

Repozitorijum ima pristup *StoreContext*-u i poziva *_context.Products.ToList()* metodu, a *DbContext* to

prevodi u upit bazi podataka (*Select * from Products*) (Slika 2).



Slika 2. Dijagram *repository pattern*

U repozitorijumu se koristi *IReadOnlyList* koji predstavlja *read-only* kolekciju elemenata kojima se pristupa uz pomoć *index*-a [5].

U ovom projektu koristi se i *specification pattern*. *Specification pattern* je *pattern* koji omogućava da se neko parče logike enkapsulira u posebnu jedinicu - specifikaciju i koristi se u različitim delovima koda.

U Core projektu u *Specification* folderu kreirana je klasa *BaseSpecification*. Ova klasa nasleđuje *ISpecification* interfejs i u njoj su implementirane sve metode iz ovog interfejsa. U nastavku je kreirana *SpecificationEvaluator* klasa koja zna šta je uključeno u specifikaciju (*queries* i *expressions*), a zatim se kreira *IQueryable* koji se može proslediti i napraviti listu od liste *expressions*-a koja je napravljena u *SpecificationEvaluator*-u [6].

U nastavku rada obrađuje se pitanje identiteta, dakle cilj je da se setuje ASP.NET Identity. ASP.NET Identity je veoma opširan, a koristi se samo mali deo njega u ovoj aplikaciji. Izdavaće *token*-e korisnicima i oni će moći da ih koriste za autentifikaciju u različitim metodama i klasama unutar API projekta. Između ostalog, koriste se i *JWT token*-i i to za tipične metode koje će omogućavati da se klijent autentifikuje u API-ju. To su zapravo *string*-ovi koji sadrže određene informacije o klijentu i takođe su potpisani od strane *API server*-a [7]. U ovom projektu koristi se *Payment Card Industry Data Security Standard* (PCI DSS). Sve industrije platnih kartica poput Visa, MasterCard i slično prate iste standarde. Konkretno se koristi *Stripe*, koji je PCI infrastruktura usklađenosti. *Stripe* vodi računa o svim ovim elementima kako bi se bezbedno pristupalo kreditnim karticama korisnika. *Stripe* obezbeđuje API-je koje *web* programeri mogu da koriste za integraciju obrade plaćanja u svoje *web* stranice i mobilne aplikacije [8].

3.2. Frontend

Frontend deo aplikacije razvijen je uz pomoć *Angular framework*-a. Uz pomoć njega je omogućeno korišćenje prethodno napravljenog API-ja. *Angular* razvojno okruženje se koristi za kreiranje klijentskog dela *web* aplikacije (*client-side framework*) korišćenjem *HTML*-a, *CSS*-a i *TypeScript*-a. Za stilizovanje *frontend* dela aplikacije korišćeni su *Bootstrap* i *Font Awesome*. Ova aplikacija se sastoji od mnoštva deljenih i pojedinačnih komponenti, a kreiranje jedne velike komponente je prikazano na primeru *Basket* komponente.

Fajl *basket-routing.module.ts* se koristi za definisanje kretanja korisnika kroz aplikaciju, odnosno iz jednog dela aplikacije u drugi, uz pomoć ruta. U ovoj aplikaciji se jedino model ne nalazi unutar komponente. Na putanji *shared/models* kreiran je *basket.ts* fajl u kojem se nalaze

interfejsi *Basket*, *BasketItem* i *BasketTotals*, kao i implementacija za *Basket*.

Izgled jedne komponente se definiše unutar pripadajućeg HTML fajla. U ovoj aplikaciji, to je fajl *basket.component.html*. Izgled HTML fajla komponente se definiše unutar fajla *basket.component.css*. CSS fajl se obično sastoji od selektora koji *target*-iraju HTML kako bi se primenio određeni stil na njegovu sadržinu.

Dekorator `@Component` označava da je klasa Angular komponenta i pruža *metadata* za konfiguraciju koji određuje kako komponenta treba da bude obrađena, instancirana i korišćena u toku izvršavanja. Angular komponente predstavljaju *subset* direktiva, koje su uvek povezane sa *template*-om [9].

BasketService importuje Angular-ov *injectable* simbol i vrši anotaciju klase sa `@Injectable()` dekoratorom. Servis može da povuče podatke odakle god je potrebno: *web* servis, lokalno skladište ili *mock data source* [10].

4. ZAKLJUČAK

U ovom radu sprovedeno je istraživanje uticaja COVID-19 na poslovanje poljoprivrednih gazdinstava. U istraživačkom delu rada predstavljen je merni instrument sa svim važnim pitanjima koji su odlični pokazatelji uticaja pandemije na određene sfere poslovanja poljoprivrednih gazdinstava. Pitanja su se odnosila na zaposlene, proizvodnju, kanale prodaje, preduzete mere za očuvanje i oporavak gazdinstava, kao i na finansije za vreme pandemije. Izvršeno je poređenje određenih segmenata poslovanja pre, u toku i nakon pandemije.

Istraživanjem je utvrđeno da je najveći problem bila prodaja poljoprivrednih proizvoda, odnosno njihovo poručivanje i plaćanje. Proces poručivanja je bio potpuno nepraktičan za kupce i prodavce. Najveći broj gazdinstava primalo je porudžbine telefonski ili putem društvenih mreža. Plaćanje se vršilo pouzecom, kuriru. Predloženo je da moguće rešenje problema bude *web* aplikacija za podršku poslovanja poljoprivrednog gazdinstva. Prilikom izrade uspešne *web* prodavnice najbitnije je obezbediti dobro korisničko iskustvo. Ova aplikacija pre svega olakšava proces poručivanja, početna stranica je jednostavna i pregledna, svi proizvodi su razvrstani u kategorije, jasno su istaknute cene za komad odnosno kilogram i jednostavno se dodaju u korpu. Svaki korisnik se pre svega registruje na portalu.

Nakon što izabere željene proizvode, korisnik nastavlja proces kupovine tako što unosi podatke za dostavu, bira željenu kurirsku službu i ukoliko želi plaćanje može da izvrši unapred, karticom. Ovaj način plaćanja je preporučen od strane zdravstva, njime se izbegava bespotreban kontakt sa papirnim novcem, a samim tim se smanjuje i rizik od zaraze kurira i kupca virusom COVID-19. Takođe, korisnik ima uvid u istoriju svojih porudžbina i u njihove statuse.

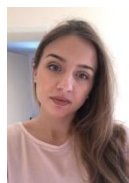
U radu je detaljno opisan proces razvoja pomenute aplikacije. Pre svega su nabrojane i opisane sve korišćene tehnologije. Aplikacija je razvijena u Visual Studio Code razvojnom okruženju. Korišćene tehnologije su ASP.NET Core, Angular *framework*, PostgreSQL baza podataka i Docker. Za realizaciju plaćanja karticom korišćena je JavaScript biblioteka Stripe. U okviru rada detaljno je prikazan postupak implementacije *backend* i *frontend* dela aplikacije, kao i konfiguracija baze uz pomoć Docker-a.

U ovom radu je opisan deo aplikacije koji koristi kupac, prikazan je njegov razvoj i upotreba. Postoje velike mogućnosti za dalji razvoj i unapređenje aplikacije. Primarno je razviti deo aplikacije za prodavca. Trenutno se proizvodi dodaju kroz bazu podataka, a potrebno je obezbediti prodavcu formu za dodavanje novog proizvoda, kao i mogućnost promene fotografije, naziva i cene proizvoda. Takođe, potrebno je obezbediti prodavcu posebnu stranicu na kojoj će imati uvid u sve pristigle porudžbine. Ideja je i da na toj stranici prodavac ima mogućnost promene statusa porudžbine, u zavisnosti da li je porudžbina u pripremi, poslata i slično. Mogućnost promene statusa porudžbine je od velike koristi za prodavca i kupca. Statistika pokazuje da najveći broj ljudi *online* kupovinu obavlja putem mobilnih uređaja. Bilo bi veoma korisno razviti i prilagoditi ovu *web* prodavnicu za mobilne uređaje.

5. LITERATURA

- [1] Poudel, Padam Bahadur, et al. "COVID-19 and its global impact on food and agriculture." *Journal of Biology and Today's World* 9.5 (2020): 221-225.
- [2] Siche, Raúl. "What is the impact of COVID-19 disease on agriculture?." *Scientia Agropecuaria* 11.1 (2020): 3-6.
- [3] Pandemija, Wikipedija, <https://sr.wikipedia.org/sr-el/Пандемија> (pristupljeno u septembru 2022.)
- [4] Dragan Milić, Prodaja putem društvenih mreža kao deo e-trgovine, <https://www.domen.rs/prodaja-putem-društvenih-mreža-kao-deo-e-trgovine>
- [5] How to work with read-only collections in C#, <https://www.infoworld.com/article/3610473/how-to-work-with-read-only-collections-in-csharp.html> (pristupljeno u septembru 2022.)
- [6] .NET Core - Using the Specification pattern alongside a generic Repository, <https://medium.com/@rudyzio92/net-core-using-the-specification-pattern-alongside-a-generic-repository-318cd4eea4aa> (pristupljeno u septembru 2022.)
- [7] Introduction to JSON Web Tokens, <https://jwt.io/introduction> (pristupljeno u septembru 2022.)
- [8] Stripe, Inc., https://en.wikipedia.org/wiki/Stripe,_Inc. (pristupljeno u septembru 2022.)
- [9] Component, <https://angular.io/api/core/Component> (pristupljeno u septembru 2022.)
- [10] Tutorial: Tour of Heroes, <https://angular.io/tutorial/toh-pt4> (pristupljeno u septembru 2022.)

Kratka biografija:



Milica Pavlović rođena je 21.10.1996. godine u Novom Sadu. Osnovnu školu "Mihajlo Pupin" je završila u Veterniku 2011. Godine, nakon čega upisuje srednju ekonomsku školu "Svetozar Miletić" u Novom Sadu. Posle završene srednje škole, 2015. godine započinje studije na Fakultetu tehničkih nauka, smer Inženjerstvo informacionih sistema. Diplomirala je 2019. Godine, nakon čega je upisala master studije na istom fakultetu, smer Inženjerstvo informacionih sistema.