

Web trgovina zasnovana na .NET tehnologiji

Barukčić, Danijela

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Bjelovar University of Applied Sciences / Veleučilište u Bjelovaru**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:144:780498>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-05**



Repository / Repozitorij:

[Digital Repository of Bjelovar University of Applied Sciences](#)



VELEUČILIŠTE U BJELOVARU
PREDDIPLOMSKI STRUČNI STUDIJ RAČUNARSTVO

**WEB TRGOVINA ZASNOVANA NA .NET
TEHNOLOGIJI**

Završni rad br. 06/RAČ/2022

Danijela Barukčić

Bjelovar, listopad 2022.



Veleučilište u Bjelovaru
Trg E. Kvaternika 4, Bjelovar

1. DEFINIRANJE TEME ZAVRŠNOG RADA I POVJERENSTVA

Student: **Danijela Barukčić**

JMBAG: **0314020220**

Naslov rada (tema): **Web trgovina zasnovana na .NET tehnologiji**

Područje: **Tehničke znanosti**

Polje: **Računarstvo**

Grana: **Programsko inženjerstvo**

Mentor: **Krunoslav Husak, dipl. ing. rač.**

zvanje: **predavač**

Članovi Povjerenstva za ocjenjivanje i obranu završnog rada:

1. **Ante Javor, struč. spec. ing. comp., predsjednik**
2. **Krunoslav Husak, dipl. ing. rač., mentor**
3. **Ivan Sekovanić, mag.ing.inf.et comm.techn., član**

2. ZADATAK ZAVRŠNOG RADA BROJ: 06/RAČ/2022

U sklopu završnog rada potrebno je:

1. Analizirati aktualne tehnologije za izradu web aplikacija.
2. Analizirati dostupna rješenja unutar .NET tehnologije za razvoj web aplikacija.
3. Predložiti i opisati relacijski model baze podataka koji će omogućiti pohranu svih podataka web trgovine.
4. Izraditi i opisati web trgovinu koristeći odabranu .NET tehnologiju.
5. Predložiti i opisati načine naplate proizvoda krajnjim korisnicima.

Datum: 19.07.2022. godine

Mentor: **Krunoslav Husak, dipl. ing. rač.**



Sadržaj

1. Uvod	1
2. Aktualna rješenja za izradu web aplikacije	3
2.1 Razvojni okviri na strani poslužitelja i klijenta	3
2.1.1 Ruby on Rails	4
2.1.2 Django	5
2.1.3 Laravel	7
2.2 Razvojna okruženja na strani poslužitelja	8
2.2.1 Node.js i Express	8
2.2.2 ASP.NET	9
2.2.3 Go	10
2.3 Razvojni okviri na strani klijenta	10
2.3.1 Angular	11
2.3.2 React	13
2.3.3 Vue.js	14
3. Korištenje tehnologije i razvojni alati	16
3.1 ASP.NET MVC	16
3.2 ASP.NET Identity	17
3.3 Entity Framework	17
3.4 SQL Server proširenje	18
3.5 Visual Studio Code	19
4. Implementacija programskog rješenja	20
4.1 Baza podataka	20
4.2 Korisnička strana aplikacije	21
4.2.1 Registracija i prijava korisnika	22
4.2.2 Korisnički profil	22
4.2.3 Pregled proizvoda	23
4.2.4 Detalji proizvoda	24
4.2.5 Izvršavanje kupovine	24
4.2.6 Povijest narudžbi	25
4.3 Administratorska strana aplikacije	25
4.3.1 Upravljanje kategorijama	25
4.3.2 Upravljanje proizvodima	26
4.3.3 Pregled narudžbi	27
5. Predložak naplate proizvoda	28
5.1 CorvusPay	28
5.2 WSPay	29
5.3 Stripe	29
6. ZAKLJUČAK	31
7. LITERATURA	32
8. OZNAKE I KRATICE	34
9. SAŽETAK	35

10. ABSTRACT.....	36
--------------------------	-----------

1. Uvod

Internet trgovina postala je sastavni dio svakog većeg trgovačkog lanca i sve je više i manjih tvrtki koje također omogućavaju kupovinu putem interneta. Ona omogućuje obavljanje kupovine bez fizičkog odlaska u npr. često pretrpane trgovačke centre. Internet trgovina odličan je odabir za proboj na nova tržišta, pokazuje ozbiljnost poslovanja, privlači dodatne kupce te je već na početku moguće predvidjeti isplativost poslovanja s obzirom na zadovoljstvo kupaca. Dostupne su 24 sata dnevno, stoga je vrlo praktično kupovinu obavljati putem interneta iz udobnosti vlastitog doma. Pokazalo se to kao dobro rješenje posebno za vrijeme blagdanskog vremena kada je velika vjerojatnost za čekanjem u dugim redovima. Također, vanjski čimbenici poput poskupljenja naftnih derivata i poteškoća u dolasku do fizičkih trgovina, uvelike su doveli do rasta prodaje putem interneta.

Zbog korištenja mrežnog sustava naplate, internet trgovina postala je sigurna kupovina što je dodatan razlog zbog kojeg ju potrošači odabiru. Mrežno plaćanje povećava promet samim time što korisniku pruža sigurnost, no osim sigurnosti razlog korištenja mrežnog plaćanja je smanjena upotreba fizičkog novca.

Razvojem trgovine, i internet trgovina postaje sve kompleksnija i zahtjevnija. Kupci traže sigurnost i što više informacija o pojedinim artiklima kako bi se odlučili za kupovinu odabranog proizvoda. Dizajn i preporuke ostalih kupaca presudni su kod odluke, stoga je jako bitno biti u korak s konkurencijom i maksimalno zadovoljiti potrebe kupaca. Kako bi to bilo moguće potrebno je konstantno pratiti trendove u svijetu te njima prilagođavati i internet trgovinu. Sve više kupaca osim računala za kupovinu koriste i pametne telefone ili tablete. Prema tome, svaki trgovac trebao bi biti svjestan da je nužno pored web stranice imati i stabilnu inačicu namijenjenu pametnim telefonima i tabletima. Amazon se pokazao kao najbolje rješenje internet trgovine, a bilježi prihod od 469,822 milijardi dolara u 2021. god, što je povećanje od 21,7% u odnosu na 2020. godinu. [1] Amazon je najpopularnija internet trgovina na svijetu te svake godine bilježi sve veći rast svojih prihoda.

Povijest internet kupovine datira iz 70-ih godina. CompuServe, osnovan 1969. godine, bio je prvi komercijalni internetski servis. Naplaćivao je korištenje brojnih igara, izmjenjivanje poruka i korištenje foruma. Amazon je prva potpuna internet trgovina, a osnovan je 1995. godine čime je službeno počeo prodavati knjige putem interneta.

Kupovina internetom uzela je dodatan zamah početkom pandemije Covid-19. Pandemija je znatno promijenila pogled na internet kupovinu i globalno pojam interneta. Zatvaranje i prisiljavanje na boravak u kući ljude je natjeralo da pronađu nove načine obavljanja

svakodnevnih kupovina. Najveći rast zabilježen je u zemljama koje su najviše pogođene pandemijom kao što su Španjolska, Italija i Belgija. Prema statistici kompanije Apptus [2], čak 44% stanovništva Španjolske kupovinu je obavljalo putem interneta. Također, među zemljama s većim postotkom rasta internet kupovine nalazi se i Belgija s 41% od ukupnog stanovništva.

Zadatak ovog završnog rada jest izraditi web aplikaciju visokih performansi i prihvatljivog dizajna za prodaju proizvoda putem interneta uz pomoć aktualnih tehnologija. Programsko rješenje imat će mogućnost korištenja stranice s administratorske strane te sa strane korisnika. Administratorski dio programskog rješenja imat će više ovlasti i mogućnosti u odnosu na običnog korisnika stranice. Osoba prijavljena kao administrator imat će mogućnost unosa, uređivanja i brisanja proizvoda kao i kategorija proizvoda, pregled statističkih podataka poput pregleda novih korisnika. Korisnička strana aplikacije mora omogućavati pregled proizvoda, detalje pojedinog proizvoda, dodavanje u košaricu, registraciju i prijavu na stranicu te pregled vlastitog profila. Općeniti pregled stranice i proizvoda te detalja proizvoda bit će omogućen svim korisnicima bez prethodne prijave.

Rad je organiziran po poglavljima na sljedeći način. Pregled aktualnih rješenja za izradu web stranice i opisi tehnologija prikazani su u poglavlju broj dva. U poglavlju broj tri obrazložen je odabir korištenih tehnologija u ovom radu. Slijedi detaljan opis programskog rješenja u poglavlju četiri, dok je u petom poglavlju opisan predložak naplatnog sustava. Poglavlje šest rezervirano je za zaključak i sadržaj cijelog rada.

2. Aktualna rješenja za izradu web aplikacije

U posljednjih nekoliko desetljeća web tehnologija doživjela je drastičnu transformaciju. Od tada nekoliko postojećih tehnologija, došlo je do razdoblja kada postoji širok spektar novih. Razvoj web stranica se u početku svodio na poznavanje tri glavne tehnologije poput HTML-a (engl. *Hyper Text Markup Language*), CSS-a (engl. *Cascading Style Sheets*) i JavaScripta. Danas je poželjno biti upoznat sa što više različitih tehnologija i programskih jezika. Tehnologije se razvijaju brže no ikada i sukladno tome potražnja za programerima je sve veća i veća.

Kako dolazi do razvoja tehnologija, tako se pojavljuju i novi programski jezici koji gotovo „preko noći“ mogu postati popularni među programerima. Postoje mnogi programski jezici i programski okviri za izgradnju web aplikacija, a neki od njih su: C#, ASP.NET, PHP, Go, JavaScript, Angular, React i još mnogo drugih. Prilikom samog odabira tehnologije koja će se koristiti za razvoj aplikacije, bitno je dobro istražiti koja bi najbolje odgovarala potrebama aplikacije koja se razvija.

Prema statističkim podacima iz 2021. godine, JavaScript je najtraženiji programski jezik među programerima s postotkom od 63%, slijedi ga HTML s 56% te Python s 48%. TypeScript zauzima 7 mjesto s 30%, a C# ga slijedi s 27% potražnje. [3] Budući da broj programskih jezika i pripadnih tehnologija svakim danom sve više raste, dobro je poznavati osnove različitih jezika radi lakšeg prebacivanja s jednom programskog jezika na drugi.

U ovom poglavlju opisani su razvojni okviri (engl. *framework*) na strani poslužitelja i klijenta (engl. *full-stack*), te tehnologije za razvoj na strani klijenta (engl. *front-end*) i poslužitelja (engl. *back-end*). Opisane su neke od najpopularnijih tehnologija, njihova zastupljenost te pojedini programski jezici.

2.1 Razvojni okviri na strani poslužitelja i klijenta

Kada je riječ o velikim aplikacijama, programiranje može postati dugotrajan zadatak koji zahtjeva puno odgovornosti, stečenog znanja i truda. Kako bi se izbjegao dugotrajan proces stvaranja aplikacije, alat koji programeri najčešće odabiru je programski okvir. Okvir je kostur unaprijed napisanog koda koji pruža jednostavniji i učinkovitiji pristup razvoju web aplikacija. Uključuju unaprijed definirane klase i funkcije koje se mogu koristiti za unos, obradu i prikaz podataka. Omogućuje zaštitu web stranice od neovlaštenih pristupa poput ubacivanja SQL-a i sprječava neovlašteno mijenjanje podataka.

Programiranje na strani poslužitelja i klijenta je jedan od najtraženijih poslova programera u razvoju softvera. To su programeri koji mogu raditi oba dijela web razvoja. Upoznati su sa svim dijelovima izgradnje poput dizajna, razvoja korisničkog sučelja, arhitekture i upravljanja bazom podataka, pozadinske logike i ostalih dijelova. Programeri s ovim vještinama visoko su cijenjeni i traženi tehnološki profesionalci. Programski jezici za izgradnju korisničkog sučelja razlikuju se od onih za izgradnju pozadinske logike. Alati, programi i sustavi za upravljanje bazom podataka za pozadinski razvoj, razlikuju se u odnosu na one koji se koriste na strani klijenta.

Node.js, Django, Laravel, Spring, Angular, Ruby on Rails, Express i Flask, samo su neki od trenutno postojećih okvira i u sebi sadrže već unaprijed pisan kod koji se može iskoristiti pri gradnji aplikacije. Pisani su različitim programskim jezicima što rezultira različitim performansama i značajkama koje su ključne kod odabira tehnologije. Jezici koji se najčešće koriste za razvoj uključuju C#, Go, Javu, JavaScript, Python, PHP i Ruby. Na primjer, primarni razlog zašto bi programeri odabrali Django je brzina razvoja, dok je npm (engl. *Node Package Manager*) razlog odabira programskog okvira Node.js.

2.1.1 *Ruby on Rails*

Ruby on Rails programski okvir poznat kao Rails ili RoR, izgrađen je na temelju programskog jezika Ruby, stvorenog 90-ih godina. David Heinemeier Hansson razvio je i objavio Ruby on Rails kao projekt otvorenog koda u srpnju 2004. godine dostupan putem web stranice rubyonrails.org. Ideja je bila programerima dati okvir za brzi razvoj robusnih web stranica s visokom učinkovitošću.

RoR je okvir koji radi po principu MVC-a (engl. *Model - View - Controller*) gdje je model zaslužan za odnos između baze podataka i objekata te sadrži kod koji radi s podacima. Pogled (engl. *View*) je HTML datoteka s dijelovima Ruby koda koja se koristi za izgradnju korisničkog dijela aplikacije, odnosno kako će korisnik pristupiti aplikaciji i način na koji se podaci prikazuju. Kontroler (engl. *Controller*) povezuje prikaze i modele, prima zahtjeve od preglednika odnosno korisnika te komunicira s modelom kako bi mogao obraditi zahtjev te daje upute pregledniku na koji način je potrebno prikazati rezultat zahtjeva korisniku.

Jednostavna upotreba omogućuje brzi razvoj aplikacija i odličan je izbor za timove s manjim brojem članova. Koristi se u raznim vrstama industrije za izradu web aplikacija poput marketinških stranica, internet trgovina, CMS-ova (engl. *Content Management System*) i prilagođenih web aplikacija. Rails podržava MySQL, PostgreSQL, SQLite, SQL

Server i DB2 sustave za upravljanje bazom podataka. Ima mnogo programskih biblioteka sa svim potrebnim alatima koje će pomoći pri stvaranju visokokvalitetnog proizvoda. AJAX biblioteka, biblioteka za pristup bazi podataka. Na slici 2.1. prikazan je prepoznatljiv logo Railsa.



Slika 2.1: Logo programskog okvira Ruby on Rails

Brzina izvođenja pripisuje se jedinstvenim značajkama i pojednostavljenim zadacima što rezultira bržim izvođenjem aplikacije. Dobro se nosi sa složenom logikom te daje preciznije rezultate što govori da je pouzdan i točan programski jezik. Ima integriranu značajku za testiranje kojom postiže dosljednost i jednostavno održavanje koda.

Vodi se politikom konvencije umjesto konfiguracije (engl. *Convention over Configuration*) i principom ne ponavljaj se (engl. *Don't Repeat Yourself*). Konvencija umjesto konfiguracije govori kako bi trebali pisati kod, a da bi svima na jednak način bio razumljiv, na primjer imenovanje varijabli, funkcija i klasa. Princip DRY uključuje recikliranje koda, njegovu ponovnu uporabu bez dupliciranja. Poštivanjem navedenog načela smanjuje se broj grešaka i time kod postaje čišći i lakši za održavanje.

Iza Railsa je dugih 16 godina, postao je pouzdan i zreo okvir za razvoj aplikacija. AngelList, GitHub, Airbnb i Crunchbase samo su neke od tvrtki koje su se oslonile na Ruby on Rails. S najnovijim značajkama koje su uvedene ažuriranjem Ruby 3 i Railsa 7, okvir je postao savršeno i ujedno jednostavno rješenje za poslovne projekte. Svojom zrelošću i zajednicom garantira da će biti sve bolji i bolji.

2.1.2 Django

Django je jedan od najpopularnijih okvira i napisan je u Pythonu. U razvoju je od 2003. godine. Službeno je objavljen u srpnju 2005. godine i od tada je jedan od traženijih okvira. Prepoznatljiv logo Django okvira prikazan je na slici 2.2. Uspješan je to okvir otvorenog

koda i broji tisuće korisnika. Razvio se u svestran okvir uz pomoć kojeg je moguće razviti gotovo bilo koju vrstu web aplikacije.



Slika 2.2: Logo Django programskog okvira

Slijedi MVT (engl. *Model – View - Template*) arhitekturu, primjenjuje DRY principe te pruža više alata koji programerima štede vrijeme, kao na primjer objektno - relacijski mapper (engl. *Object – Relation Mapper*) ili Django admin. Django podržava brojne baze podataka uključujući MariaDB/MySQL, PostgreSQL, SQLite i Oracle. Dostupno je mnoštvo biblioteka napisanih u Pythonu koje naglašavaju učinkovitost, mogućnost ponovne uporabe komponenti i manje koda.

Neke od poznatijih tvrtki diljem svijeta koriste Django okvir za svoje stranice, a to su Pinterest, NASA, Instagram, The Washington Times (popularan izvor najnovijih vijesti na internetu), Mozilla, Disqus (najpoželjnije mjesto za blogove i komentare na globalnoj razini) i Bitbucket (uslužno hosting spremište izvornog koda u vlasništvu Atlassiana).

Podržava ga velika zajednica korisnika i zagovornika, postoje razni kanali i forumi za razmjenu mišljenja, besplatne radionice i sastanci kojima privlače nove i zadržavaju postojeće korisnike. Django odgovara gotovo svakoj vrsti projekta, radilo se o mobilnoj ili web stranici. Dobro se nosi s jednostavnim i visoko opterećenim aplikacijama jer radi s većinom velikih baza podataka i pruža fleksibilnost odabira prikladne baze podataka s obzirom na kompleksnost projekta. Budući da je Django temeljen na Pythonu, nasljeđuje sve njegove prednosti kao što su jednostavnost, produktivnost te podrška vanjskih biblioteka. Korištenjem pravila sintakse Pythona, programeri mogu održavati kod čistim i čitljivim.

Među svim prednostima koje donosi ovaj okvir, postoje i mane zbog kojih se inženjeri odlučuju graditi aplikaciju u drugim okvirima. Veliki nedostatak predstavlja konvencija. Za razliku od drugih okvira, u Django sve treba biti eksplicitno definirano što ponekad dovodi do usporenog razvoja. Postoji određeni skup datoteka i unaprijed definiranih varijabli koje programeri moraju naučiti prije nego se upuste u izgradnju. Programeri ne mogu koristiti

vlastite strukture datoteka već su primorani slijediti implementirana pravila Djanga. Krivulja učenja dosta je strma jer dolazi s puno značajki i različitom sintaksom koja nekima nije lako i odmah razumljiva, te je ovo prepreka svima koji se prebacuju s drugih jezika.

2.1.3 *Laravel*

Laravel je besplatan PHP okvir otvorenog koda koji je razvio Taylor Otwell. Njegov logo prikazan je na slici 2.3. Dok je PHP jedan od najstarijih i najpopularnijih jezika za web razvoj, Laravel je njegov najpoznatiji okvir. Prva stabilna verzija objavljena je u lipnju 2011. godine, dok je zadnja stabilna verzija 9 izdana u veljači 2022. godine. Pruža alate i resurse potrebne za izgradnju modernih PHP aplikacija.



Slika 2.3: Laravel logo

Slijedi MVC strukturu, ima ugrađenu provjeru autentičnosti i autorizacije, podršku za pohranu u oblaku ili lokalno. Ima vlastito sučelje naredbenog retka imena Artisan. Njegova uobičajena uporaba uključuje upravljanje migracijama, bibliotekama te stvaranje osnovnog koda. Laravel ima najbolju ORM (engl. *Object-relation mapping*) implementaciju u odnosu na ostale okvire i omogućuje interakciju s objektima baze podataka koristeći ekspresivnu sintaksu. Podržava MySQL, SQL Server, PostgreSQL i SQLite baze podataka.

Laravel nudi performanse idealne za poduzeća u razvoju (engl. *Start-up*) i one već razvijene tvrtke. Dominantan je izbor među iskusnim i profesionalnim programerima što im omogućuje izradu visoko produktivne, bogate raznim značajkama i dinamičke web aplikacije. Neke najpoznatije web aplikacije izgrađene okvirom Laravel jesu MyRank aplikacija koja omogućuje studentima pristup ispitima ili Internet trgovina za prodaju markiranih satova Invicta Watch Group. Startups.co je aplikacija namijenjena tzv. *start-up* poduzetnicima koji tek ulaze na tržište i nudi im obrazovanje i pomoć u bržem razvoju i predstavljanju usluga i proizvoda koje nude.

Kroz dane primjere, vidljivo je da se radi o univerzalnom okviru, bilo da se radi o marketinškoj stranici, skromnoj ili bogatoj aplikaciji te web portalu. Prednost sigurnog programiranja ne dovodi u upit budući razvoj i potražnju za ovim okvirom. Dobro poznat po novim inovativnim idejama zasigurno je da će u budućnosti visoko kotirati na tržištu.

2.2 Razvojna okruženja na strani poslužitelja

Izgradnja web aplikacije uključuje izradu i poslužiteljske strane jer bez nje je nemoguće imati funkcionalnu aplikaciju. Strana poslužitelja je sve ono što korisnik ne vidi, a nužno je za osnovni rad web aplikacije. Na strani poslužitelja nalaze se sve logičke operacije nad pohranjenim podacima, funkcije za rad podacima i prikaz elemenata kako bi se olakšalo korištenje aplikacije. Kroz korisničko sučelje korisnik šalje zahtjev koji se obrađuje u pozadini kako bi obrađena informacija povratno stigla do korisnika. Rad na strani poslužitelja uključuje veliki opseg znanja različitih programskih jezika i tehnologija. U sljedećem poglavlju opisani su pojedini programski jezici i tehnologije namijenjeni za izgradnju poslužiteljske strane aplikacije.

2.2.1 *Node.js i Express*

Express kao razvojno okruženje Node.jsa razvijen je za programiranje poslužiteljske strane web aplikacije. Express je vrlo brzo postao popularno razvojno okruženje za razvoj zahvaljujući velikoj popularnosti Node.jsa. Kompatibilan je s okruženjima poput Kraken, Loopback i Sails. Omogućuje brzi razvoj aplikacija i pogodan je za one primjene kada aplikacije moraju obraditi puno zahtjeva korisnika. Posjeduje razne pakete za rješavanje mnogih programskih problema, dobro je dokumentiran i relativno jednostavan za učenje. Nedostatak standardizacije Expressa stvara probleme velikim timovima jer ne postoji određena struktura projekta koje se treba pridržavati. Zbog toga, razvoj je često otežan jer svaki programer na različit način razvija funkcionalnost aplikacije.

Radi na JavaScript pogonskom sustavu (engl. *engine*) koji omogućava izvršavanje JavaScript koda izvan web preglednika. Koriste ga velike tvrtke poput PayPala, Ubera i Netflix. Skalabilan je, lagan i brz kada je u pitanju rad s podacima. Node u ovom slučaju pruža produktivnost budući da se JavaScript može koristiti na poslužiteljskoj i klijentskoj strani.

2.2.2 ASP.NET

ASP.NET (engl. *Active Server Pages Network Enabled Technologies*) okvir je otvorenog koda izgrađen od strane Microsoft korporacije. Prvi put je objavljen u siječnju 2002. godine s verzijom 1.0 .NET okvira i nasljednik je ASP tehnologije. Prva je to tehnologija Microsofta za izgradnju dinamičkih web stranica, aplikacija i usluga na strani poslužitelja. Za razliku od ASP-a, ASP.NET programska rješenja mogu se izvršavati na Windows, Linux ili macOS operativnim sustavima.

ASP.NET izgrađen je na CLR-u (engl. *Common Language Runtime*) što omogućuje pisanje ASP.NET koda uz pomoć bilo kojeg podržanog .NET jezika. Programski jezik C# je vrlo čest odabir kod razvoja ASP.NET aplikacija.

Ovim okvirom mogu se izgraditi različite vrste web aplikacija. ASP.NET ima tri okvira za izradu aplikacija: ASP.NET Web API, ASP.NET MVC i ASP.NET Web (Razor) Pages. Svaki od njih donosi svoje značajke koji podržava drugačiji stil razvoja. Odabir ovisi o znanju i vještinama, vrsti aplikacije koja će se razvijati te o samoj namijeni aplikacije. Velika brzina, odlična dokumentacija te velika podrška su među najznačajnijim prednostima korištenja ASP.NET okvira.

Daje potpunu kontrolu nad razvojem te se može koristiti na bilo kojem projektu, velikom ili malom. Ažuriranja poput poboljšanja sigurnosti aplikacije i Hot reloada te veća opremljenost za razvoj internet stvari (engl. *Internet of Things*), poboljšanja su koja okvir čine čestim odabirom. Hot reload omogućuje pregled novih promjena bez ponovnog učitavanja aplikacije. Programskim kodom 2.1. i 2.2. prikazano je pojednostavljeno korištenje imenskog prostora (engl. *namespace*) u novim verzijama .NET okvira. Dovoljno je na početku dokumenta deklarirati *namespace* bez ugnježdivanja preostalog sadržaja.

Programski kod 2.1: Stara sintaksa uporabe imenskog prostora

```
namespace MyNamespace {  
    class MyClass {...}  
}
```

Programski kod 2.2: Nova sintaksa uporabe imenskog prostora

```
namespace MyNamespace ;  
  
    class MyClass {...}
```

2.2.3 Go

Go, poznat još kao Golang, statički je tipiziran programski jezik, dizajniran od strane Googlea. Javno je objavljen 2009. godine. Brzo je postao popularan i čest izbor mnogih programera. Njegovoj popularnosti pridonijela je jednostavnost sintakse, čitljivost te visoka učinkovitost s ugrađenom podrškom za konkurentno izvođenje koda. Prepoznatljiv logo Go programskog jezika prikazan je na slici 2.4.

Dizajneri Robert Griesemer, Rob Pike i Ken Thompson imali su ideju stvoriti programski jezik na razini sustava što ih je dovelo do stvaranja novog jezika otvorenog koda. Višenamjenski je to programski jezik namijenjen za programiranje na strani poslužitelja, web razvoj, razvoj igara, te za programiranje u oblaku. Google, Netflix, Twitch, Ethereum, Dropbox, Docker, PayPal, Uber, Microsoft samo su neke tvrtke koje koriste Go prilikom razvoja svojih aplikacija.

U usporedbi s drugim jezicima njegova sintaksa je jednostavnija što omogućava lakše učenje. Ima ugrađene mape, razne značajke, popise te primitivne tipove kao byte, int, string koji su jednostavni za korištenje. Za razliku od programskih jezika C i C++ jednostavnije je upravljati memorijom.

Jedan je od jednostavnijih programskih jezika, stoga ga je lako naučiti uz osnovno znanje drugih programskih jezika. Sve je više popularan među programerima zbog svoje aktivne zajednice te kvalitetne i čitljive dokumentacije.



Slika 2.4: Logo programskog jezika Go

2.3 Razvojni okviri na strani klijenta

Programiranje na strani klijenta obuhvaća izradu dijela web aplikacije koja se izvršava na strani krajnjeg korisnika. Dok strana poslužitelja uglavnom upravlja cijelom logikom web aplikacije, programer na strani klijenta ima zadatak svu logiku prikazati na korisniku jasan način. Dizajnirati i izgraditi korisničko sučelje (engl. *User Interface*) i omogućiti

jednostavno rukovanje aplikacijom. Ovakva vrsta programiranja zahtjeva poznavanje osnovnih alata za izgradnju korisničkog dijela aplikacije, a to su HTML, CSS i JavaScript.

HTML temeljni je prezentacijski jezik. Namjena mu je oblikovati sadržaj za prikaz na web stranici. Omogućuje uključivanje multimedijskog sadržaja, što je eliminiralo potrebu korištenja Adobe Flash Playera i sličnih dodataka. Ova funkcija omogućena je u HTML5, koji je objavljen 2014. godine.

JavaScript je dobro poznat programski jezik poznat po revoluciji na webu zbog svojih dinamičkih mogućnosti. Dinamički zapisuje sadržaj na kojem su moguće promjene, omogućuje interaktivnost poput pomicanja elemenata, dvostrukih klikova i slično.

CSS je stilski jezik koji omogućuje uređivanje dokumenata napisanih pomoću HTML-a. Pomoću CSS-a mogu se kontrolirati primjerice fontovi i boje na web stranici.

Osim navedenih temeljnih jezika za razvoj klijentske strane, programeri mogu koristiti tehnike poput AJAX (engl. *Asynchronous JavaScript And XML*) što omogućuje ažuriranje podataka bez potrebe za osvježivanjem cijele stranice. Kako bi se to ostvarilo, programeri koriste razne biblioteke kao što su Angular, React ili Vue.js.

Osim same izgradnje web aplikacije, programeri moraju osigurati kod bez grešaka češćim testiranjem i uklanjanjem istih. TDD (engl. *Test Driven Development*) jedan je od načina na koji programeri mogu pisati kod. To je tehnika agilnog razvoja softvera, usredotočenog na pisanje testova nad izgrađenim kodom. Izgrađeni testovi mogu vrlo rano ukazati na postojanje grešaka koje inače ne bi bile otkrivene na vrijeme.

2.3.1 *Angular*

Angular je okvir i platforma za izgradnju dijela web aplikacije na strani klijenta. Razvijen je od strane Google inženjera Miska Heverya i Adama Abronsa. Prva verzija ovog okvira izašla je 2009. godine pod imenom AngularJS, dok se ostale verzije vode pod imenom Angular. Sufiks „JS“ označava kako se radi o JavaScript okviru za izgradnju web aplikacija, njegovo izbacivanje označava napredniju verziju i sve nove verzije nose jednostavan naziv Angular.

Napisan je programskim jezikom TypeScript razvijenim od strane Microsofta. TypeScript se prevodi u JavaScript kako bi ga web preglednici mogli interpretirati. Angular je moćan programski okvir na strani klijenta koji se koristi za razvoj velikih aplikacija, stoga ga koriste neke od većih tvrtki poput BMW-a, Xboxa, te HBO-a i Nikea.

Prednosti korištenja Angular okvira su MVC arhitektura, pojednostavljeno upravljanje komponentama, korištenje mehanizma modula za grupiranje direktiva, komponentata i servisa. Angular stranice su SPA (engl. *Single Page Applications*) što omogućuje učitavanje svih stranica bez potrebe za poslužiteljem, dajući dinamičan i interaktivniji doživljaj prilikom korištenja od strane krajnjeg korisnika. Ima ugrađenu podršku za servise i ubrizgavanje ovisnosti (engl. *Dependency Injection*) što dodatno odvaja logiku od komponenti rezultirajući čistim kodom. Primjer njegova korištenja i razdvajanja prikazan je programskim kodom 2.3 i 2.4.

Programski kod 2.3: Dependency injection, izgled servisa

```
export class ProductsService {  
  
    constructor(private httpClient: HttpClient) { }  
  
    getProducts(): Observable<Product[]> {  
        return this.httpClient.get<Product[]>(API);  
    }  
}
```

Programski kod 2.4: Dependency injection, izgled komponente

```
export class ProductDisplayComponent implements OnInit {  
    constructor(private productService: ProductsService) { }  
  
    ngOnInit(): void {  
        this.productService.getProducts().subscribe((items) =>  
        {  
            this.products = items;  
        })  
    }  
}
```

Iako moćan, nije najpopularniji zbog svoje strme krivulje učenja koja odbija programere. Složen je i opširan stoga nije pogodan za male aplikacije poput drugih okvira za razvoj aplikacija na strani klijenta. Ima ograničene SEO (engl. *Search Engine Optimization*) mogućnosti, no moguće je unaprijed renderirati Angular uz pomoć Angular Universal. Jedan od najvećih problema je to što su izgrađene aplikacije relativno velike pa zato i sporije.

Njegova upotrebljivost, fleksibilnost i mogućnost održavanja ga čini jedinstvenim okvirom. Pred Angularom je svijetla budućnost, prošao je dug put, i s vremenom postaje sve

bolji. Trenutno pokreće visokokvalitetne aplikacije na tržištu, stoga je odličan odabir za izradu velikih i kompleksnih aplikacija.

2.3.2 *React*

React je JavaScript biblioteka namijenjena za razvoj korisničkog sučelja podržana od strane Facebooka. Prvi put je objavljena u svibnju 2013. godine i trenutno je jedna od najčešće korištenih biblioteka za razvoj aplikacija na strani klijenta. Njegova uporaba fokusirana je na izradu web i mobilnih aplikacija.

Nudi različite pogodnosti poput jednostavne izrade dinamičkih aplikacija. Zahtjeva manje programiranja i nudi širok spektar funkcionalnosti što kod čini jednostavnijim. Nudi *Virtual DOM* čime se uspoređuju prethodna stanja komponenti te ažurira samo stavke koje su izmijenjene što rezultira bržim radom aplikacije. React prati jednosmjerni protok podataka što rezultira lakšim otklanjanjem grešaka i ugniježđenim komponentama. Komponente Reacta građevni su blokovi aplikacije, imaju svoju logiku te se mogu ponovno koristiti na drugim mjestima aplikacije. Implementiraju *render* metodu koja uzima ulazne podatke i vraća ono što treba prikazivati. Tim se ulaznim podacima pristupa putem *this.props* kao što je vidljivo iz primjera 2.5.

Programski kod 2.5: Dohvaćanje podataka pomoću render metode

```
class HelloMessage extends React.Component {
  render() {
    return <div>Hello {this.props.name}</div>;
  }
}

root.render(<HelloMessage name="Danijela" />);
```

Visok tempo razvoja Reacta može biti njegova prednost i nedostatak. Brzim razvojem biblioteke dovodi do redovitog ažuriranja aplikacija. Njegova relativno loša dokumentacija također je uzrokovana stalnim ažuriranjem aplikacije bez popratnih izmjena. Pisanjem vlastitih uputa moguće je izbjeći ovaj problem. Pomoću Reacta moguće je graditi korisničko sučelje, stoga je potrebna upotreba drugih tehnologija kako bi se postigla kompletna aplikacija.

U odnosu na Angular, jednostavniji je za učiti jer uglavnom kombinira osnovne koncepte HTML-a i JavaScripta s nekim dodacima. Njegova jednostavna struktura i

fleksibilnost ga čini idealnim za sve početnike. Trenutna popularnost i velika zajednica pokazuju kako će potražnja za Reactom i dalje rasti.

2.3.3 *Vue.js*

Vue.js je JavaScript okvir namijenjen za izgradnju korisničkog sučelja i SPA aplikacija čiji je logo prikazan na slici 2.5. Dizajnirao ga je Evan You te je objavljen 2014. godine kao okvir otvorenog koda. Ideja Evana Youa je izgraditi okvir kombiniranjem najboljih značajki iz Angular i React okvira.



Slika 2.5: *Vue.js* logo

Korištenje arhitekture MVVM (engl. *Model – View - ViewModel*) omogućuje odvajanje logike i modela od sučelja korisnika ili pogleda. Ima temeljnu biblioteku fokusiranu na sloj prikaza dok se sve dodatne funkcije moraju prikupiti korištenjem drugih. *Vue Router* i *Vuex* biblioteke okvira služe za usmjeravanje i upravljanje stanjem. Usmjeravanje je mehanizam koji vodi HTTP zahtjeve do koda koji na kraju obrađuje iste.

Osim što je progresivan također se naziva *incrementally acceptable* što znači da su aplikacije koje ga koriste dizajnirane iz temelja. Lako je započeti izradu, no aplikacija se može lako zakomplicirati sukladno daljnjem razvoju aplikacije. Veličina njegovog paketa je 21KB što ga čini „lakšim“ i bržim u odnosu na druge okvire. Pruža jednu od najsnažnijih biblioteka za jedinično ili *end-to-end* testiranje.

Ima veliku zajednicu te dobro napisanu dokumentaciju što uvelike olakšava pronalazak odgovora i rješavanje problema. Proći će neko vrijeme prije nego što Vue dostigne svoj vrhunac korištenja prvenstveno jer je vrlo mlad okvir.

Dobro poznavanje CSS, HTML i JavaScript tehnologija odlična je podloga za početak učenja Vue.js okvira. Njegova lagana krivulja učenja govori o fleksibilnosti i jednostavnosti.

Ubraja se u tri najbolja JavaScript okvira. Podržavaju ga velike tvrtke poput Alibaba i Xiaomi i drugih. Anketa *Jet Brains Dev Ecosystem* iz 2021. otkriva da je uporaba Vuea

visoka te da će i dalje rasti. Prema [4] udio korisnika, u 2021. godini, porastao je na 43% dok je React doživio strmoglav pad sa 64% na 49%.

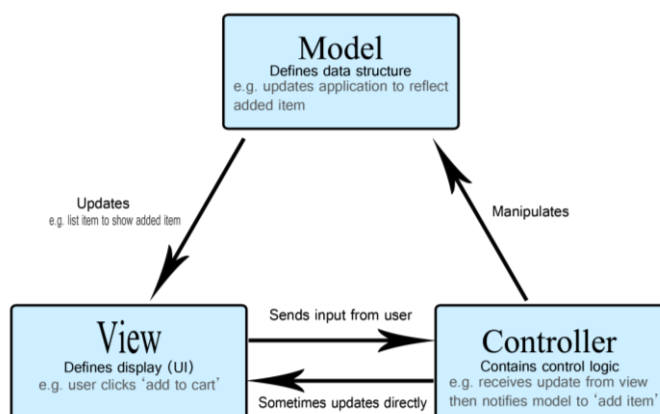
3. Korištenje tehnologije i razvojni alati

U ovom poglavlju detaljnije su opisane odabrane tehnologije za izradu internet trgovine. Tehnologije korištene za razvoj izabrane su prema sljedećim kriterijima: tehnologija koja najbolje odgovara složenosti aplikacije te u kojem smjeru će se ona u budućnosti razvijati, prema prethodnom znanju i vještinama, potrebnoj razini sigurnosti aplikacije te faktorima poput trendova i najviše korištenih jezika i okvira za izradu web aplikacija.

Glavni cilj ovog rada jest izraditi jednostavno i jeftino rješenje internet trgovine sa svim osnovnim funkcijama koje su opisane u poglavlju četiri. Okvir odabran za razvoj aplikacije jest ASP.NET MVC, a podsustav korišten za autentifikaciju i autorizaciju korisnika jest ASP.NET Identity. Pomoću *Entity Frameworka* izgrađena je baza podataka, a sustav za upravljanjem bazom podataka je SQL Server 2019 kojem se je pristupalo pomoću proširenja SQL Server Database.

3.1 ASP.NET MVC

MVC je standardni obrazac dizajna koji dijeli aplikaciju na tri djela: model (engl. *Model*), pogled (engl. *View*) i kontroler (engl. *Controller*). Korištenjem MVC-a omogućeno je razdvajanje obrade podataka od korisničkog sučelja i primanja zahtjeva korisnika. Uz pomoću ove tehnologije omogućen je razvoj vrlo jednostavnih do vrlo složenih aplikacija. Prema MVC strukturi (slika 3.1) aplikacija komunicira s korisnikom preko pogleda koji šalje zahtjev prema kontroleru. Kontroler u sebi sadrži logiku koja je zadužena za ažuriranje modela ili pogleda. Rezultat kontrolera šalje se u pogled koji definira kako bi podaci aplikacije trebali biti prikazani.



Slika 3.1: MVC struktura

3.2 ASP.NET Identity

ASP.NET Identity podsustav je okvira ASP.NET koji programerima aplikacije omogućuje autentifikaciju i autorizaciju korisnika. Microsoftova je to biblioteka kojom je moguće upravljati korisnicima. Sadrži značajke poput pohrane korisnika prilikom njegove registracije, prijavljivanje u aplikaciju, upravljanje zahtjevima i provjeru valjanosti zaporke. Sve vrste ASP.NET aplikacija, uključujući MVC, web-obrasce i web stranice mogu ju ugraditi i koristiti njene funkcionalnosti.

Dvofaktorska autentifikacija dodaje dodatan sloj zaštite kod provjere korisnika. Umjesto klasične provjere upisanog korisničkog imena i lozinke, omogućena je dodatna provjera jednokratnim pinom. Ukoliko korisnik određeni broj puta unese pogrešnu lozinku, njegov račun bit će zaključan na određeno vrijeme. Ove značajke moguće je onemogućiti ovisno o vrsti aplikacije koja se razvija. Potvrdu mail adrese koriste gotovo sve web aplikacije, čime se relativno sprječava pohrana lažnih računa. Mogućnost potvrde mail računa još je jedna prednost *Identity* sustava. Pored nabrojanih, tu su još značajke poput ponovnog postavljanja zaboravljene lozinke te brisanje korisničkog računa.

Dizajniran je da zamijeni klasične sustave sigurnosti te pruži pouzdane i provjerene metode provjere autentičnosti.

3.3 Entity Framework

Prema [5], Entity Framework je ORM okvir otvorenog koda podržanog od strane Microsoft tvrtke. Omogućuje programerima rad s podacima koristeći klase te uklanja potrebu pisanja koda koji je potreban za pristup podacima. ORM je akronim koji označava objektno/relacijsko preslikavanje i koristi se za upravljanje podacima korištenjem objekata modela. Na ovaj način se upravlja podatkovnim slojem korištenjem modela odnosno objekata izgrađenim u odabranom programskom jeziku.

Budući da se podaci dohvaćaju pomoću objekata, koristi se manje koda za upravljanje podacima, a samim time manje je koda za održavanje. Prati vrijednosti svojstva entiteta koje su promijenjene te ažurira promjene ukoliko je obavljena neka od operacija poput brisanja, ažuriranja ili umetanja. Slijedi određene konvencije, omogućuje predmemoriranje (engl. *caching*) što znači da pohranjuje rezultate često korištenih upita.

Korištenje *Entity Frameworka* smanjuje veličinu koda i povećava čitljivost prilikom njegove uporabe. Koristeći ovaj okvir, postavljaju se upiti prema bazi uz pomoć LINQ-a

(engl. *Language - Integrated Query*), a potom se dohvaćen odgovor automatski pretvara u instance objekata kako bi se mogli koristiti unutar aplikacije. LINQ je jezično proširenje za dohvaćanje podataka iz izvora poput XML dokumenta (engl. *Extensible Markup Language*) baze podataka i zbirki.

Entity Framework nudi tri oblika uporabe okvira:

- Prvo baza (engl. *Database-first*)
- Prvo kod (engl. *Code-first*)
- Prvo model (engl. *Model-first*)

U ovom radu, baza podataka realizirana je uporabom oblika prvo kod. Ovaj oblik izrade baze podataka namijenjen je za bazu koja ne postoji ili ne sadrži podatke. Prvotno je potrebno definirati klasu i njegova svojstva što je vidljivo programskim kodom 3.1. Na temelju svojstva odnosno stvorenog modela, okvir izrađuje bazu podataka i potrebne objekte u bazi.

Programski kod 3.1: Primjer modela Category

```
using System.ComponentModel.DataAnnotations;

public class Category {
    public int Id { get; set; }
    public string? Name { get; set; }
    public List<Product>? Product { get; set; }
}
```

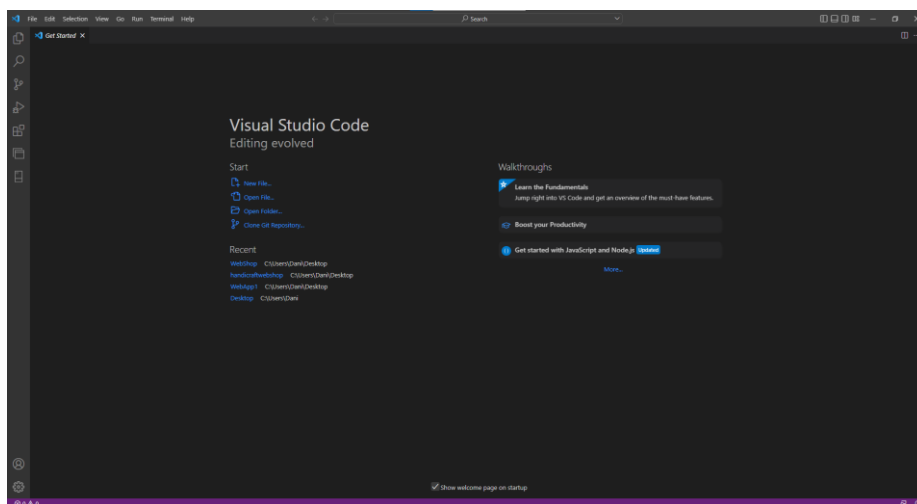
3.4 SQL Server proširenje

SQL Server proširenje je za Visual Studio Code razvojno okruženje. Koristi se za rad s bazama podataka SQL Server na Windows, macOS i Linux operativnim sustavima. Ovo proširenje pruža mogućnost postavljanja upita za Azure SQL i SQL Server. Razvio se iz metodologije poznate kao ERM (engl. *Entity-Relationship Modeling*). ERM definira shemu entiteta i njihove odnose. Entitet definira shemu objekta, no ne i njegovo ponašanje, prema tome entitet nije isto što i objekt. Programeru pojednostavljuje proces mapiranja te smanjuje vrijeme same izrade.

3.5 Visual Studio Code

Visual Studio Code je pojednostavljeni uređivač koda s podrškom za razvojne operacije poput otklanjanja pogrešaka, pokretanja zadataka i kontrole verzija [6]. Besplatan je uređivač za privatnu ili komercijalnu upotrebu te podržava macOS, Linux i Windows operativne sustave. Omogućuje izradu aplikacija brojnim programskim jezicima poput Java, JavaScript, Go, Node.js, Python, C++, C, C#. Njegovo sučelje prikazano je na slici 3.2.

VS Code može dati savjete o najboljem parametru funkcije, sintaksi ili varijabli. Otkriva pogreške prilikom programiranja i nudi prijedlog kako ih ispraviti. Doživio je veliki porast popularnosti među programerima. Ima ugrađeni terminal, kao i podršku za Git, a to su dvije velike značajke koje programere privlače. Git je besplatni distribuirani sustav za kontrolu verzija otvorenog koda koji je dizajniran za brzu i učinkovito praćenje koda raznih projekata, od malih do vrlo velikih projekata [7].



Slika 3.2: Sučelje Visual Studio Codea

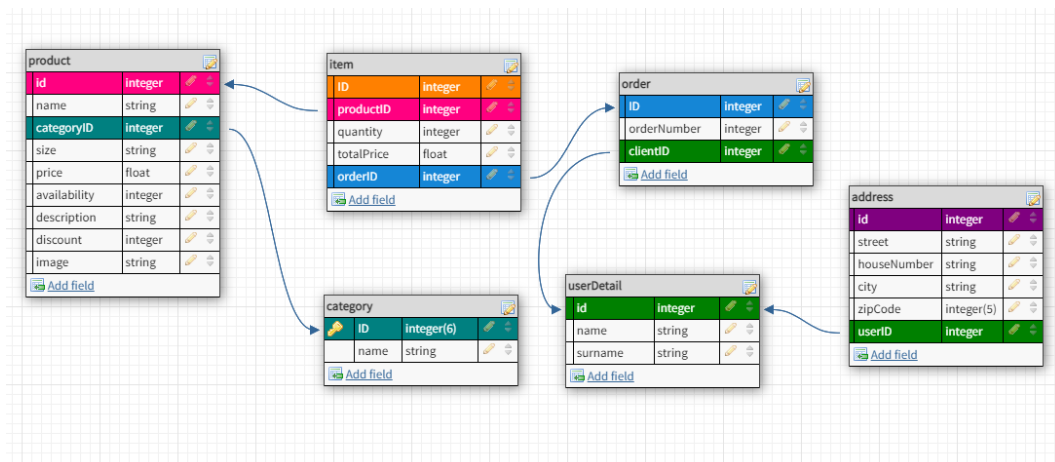
4. Implementacija programskog rješenja

U ovom poglavlju opisan je proces izrade baze podataka sukladno zahtjevima aplikacije ovog rada. Nakon toga slijedi detaljan opis svih funkcionalnosti koje se mogu izvoditi sa strane korisnika aplikacije te s administratorske strane.

4.1 Baza podataka

U okviru ovog rada korišten je SQL Server ekstenzija za upravljanje bazama podataka opisan u poglavlju 3.4. Za izradu odgovarajuće baze korišten je *DB Designer*, mrežni alat za dizajniranje i modeliranje sheme baze podataka bez pisanja SQL-a.

Na slici 4.1 prikazan je dijagram baze podataka izrađen prema zahtjevima internet trgovine ovog rada. Svaka tablica sa slike predstavlja i tablicu u bazi podataka. Pristupom prvo kod izrađeni su modeli s odgovarajućim svojstvima, a zatim su uz pomoć migracije izgrađene odgovarajuće tablice unutar baze podataka.



Slika 4.1: Dijagram baze podataka

Migracija je C# datoteka koju je moguće prepoznati po .cs ekstenziji. Generirana je od strane *Entity Frameworka*, a u njoj su definirana sva pravila pojedinog modela za izgradnju tablica. Pravilima se mogu odrediti kojeg tipa podatka će biti pojedino svojstvo, definirati primarni i strani ključ i sl. Shema baze podataka definira strukturu i međusobne odnose relacijskih tablica. Zbog promjena koje mogu nastati rastom i razvojem aplikacije, nemoguće je isto tako izbjeći promjene baze podataka. Migracija u tom slučaju omogućuje održavanje

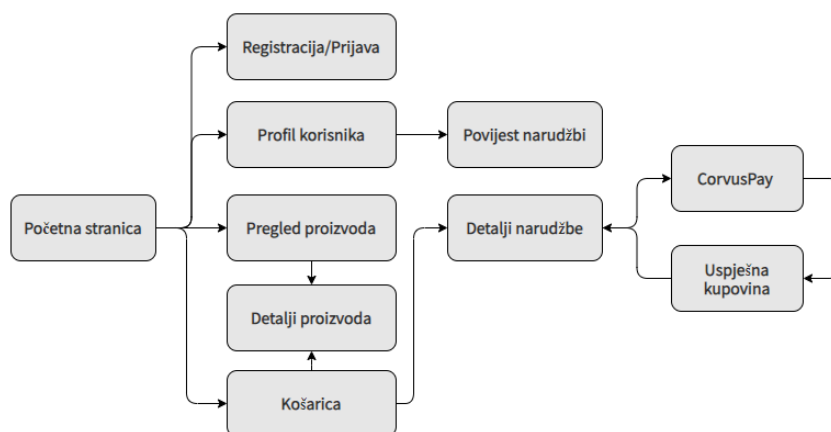
bazu podataka ažurnom. Znači da se primjerice dodavanjem novog svojstva, bez migracije, stanje u bazi neće promijeniti.

Kreiranjem migracije uspoređuje se trenutno stanje modela s prethodnom migracijom ukoliko ona postoji te se generira datoteka koja sadrži naslijeđenu klasu sa *Up* i *Down* metodama. Migracije je moguće stvarati, brisati i primijeniti na bazu podataka.

4.2 Korisnička strana aplikacije

Aplikacija na strani korisnika namijenjena je krajnjem korisniku, odnosno kupcima. To se odnosi na sve funkcionalnosti koje korisnik može obavljati unutar web aplikacije. Korisnički dio aplikacije osmišljen je na način da korisnicima omogućuje sljedeće:

- registracija i prijava na web stranicu
- pregled ponude proizvoda kao i detalje pojedinog
- pregled profila korisnika
- pregled povijesti narudžbi
- dodavanje proizvoda u košaricu
- pregled detalja narudžbe.



Slika 4.2: Struktura korisničke strane aplikacije

Aktivnosti poput pregled profila korisnika, povijest narudžbi nisu omogućeni korisnicima koji pristupaju aplikaciji bez prethodne prijave. Na slici 4.2 prikazana je struktura aplikacije sa aspekta korisničke strane. Početna stranica je polazišna točka cijele aplikacije s koje korisnik započinje svoju kupovinu. To je zajednički *layout* koji koriste i ostali pogledi aplikacije, a sastoji se od navigacijske trake za snalaženje po aplikaciji,

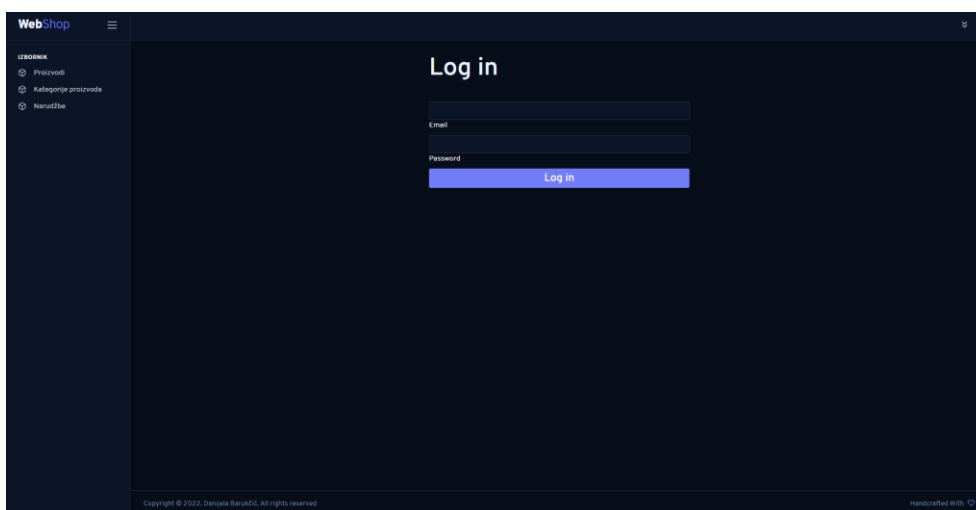
zaglavlja i podnožja. U sljedećim poglavljima detaljnije su opisane sve funkcionalnosti internet trgovine u okviru ove stranice te načini pristupa istoj.

4.2.1 *Registracija i prijava korisnika*

Registracija korisnika u aplikaciju nije nužno potrebna ukoliko korisnik želi pristupiti aplikaciji. Logika za registraciju i prijavu izrađena je uz pomoć sustava *ASP.NET Identity* sustava. *Identity* sadrži sve potrebne tablice i metode za provjeru korisničkih podataka.

Prilikom izrade novog računa dodjeljuje se uloga korisniku kojom upravlja identitet u bazi podataka. Svakom novom korisniku dodijelit će se uloga *user*, a ona se definira u tablici *Roles*. Na taj način moguće je zabraniti pristup pojedinim dijelovima aplikacije. Nakon pravilnog unosa podataka, potrebno je potvrditi valjanost mail računa. Račun je potrebno potvrditi kako bi se mogla odvijati daljnja prijava u stranicu.

Na slici 4.3 prikazan je dizajn stranice za prijavu kroz koju korisnici vrše prijavu. Za uspješnu prijavu potrebno je upisati točne podatke koji su dani prilikom registracije.

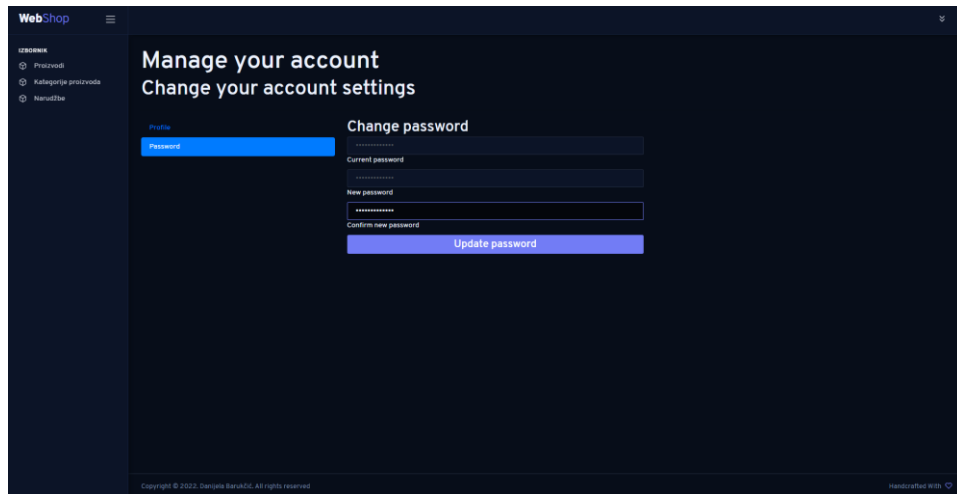


Slika 4.3: Dizajn stranice putem koje se korisnici prijavljuju

4.2.2 *Korisnički profil*

Prijavljeni korisnici mogu pregledavati i ažurirati svoje osobne podatke. Korisnik nakon uspješne registracije i prijave ima mogućnost izmjene osobnih podataka. Na profilu korisnika dostupni su osnovni podaci: ime, prezime, kontakt broj, korisničko ime (engl. *username*). Izmjena osobnih podataka izvršava se odabirom polja koje korisnik želi promijeniti te se izmijenjeni podaci spremaju odabirom tipke „Spremi“ što će rezultirati ažuriranjem profila. Osim ažuriranja osobnih podataka, korisnik može ažurirati i lozinku.

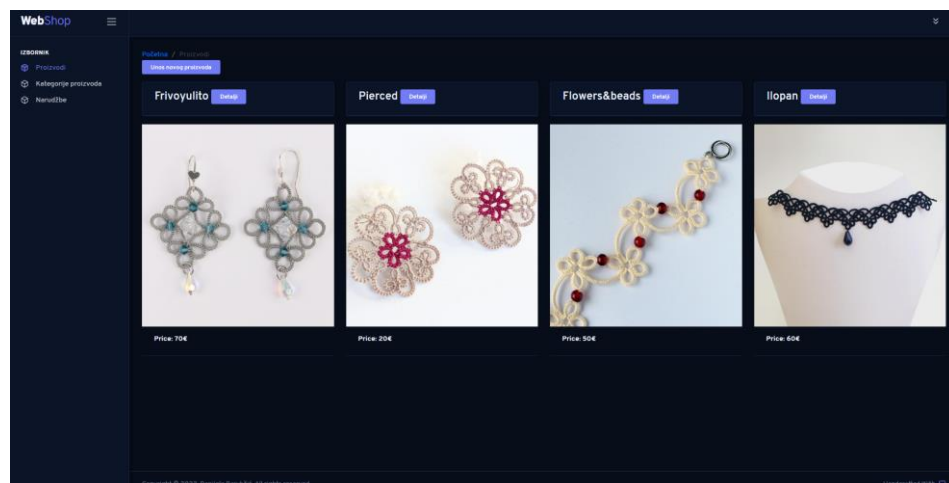
Korisnik kroz stranicu prikazanu na slici 4.4 upisuje trenutnu lozinku, a zatim novu koja će se uspješno pohraniti ukoliko ju korisnik potvrdi još jednim upisom.



Slika 4.4: Stranica namijenjena promjeni lozinke

4.2.3 Pregled proizvoda

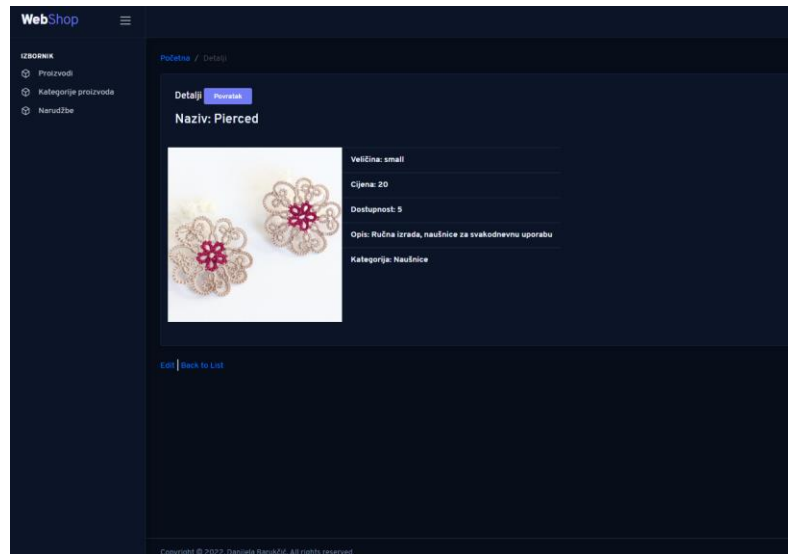
Glavna stranica internet trgovine je pregled proizvoda. Na njoj se nalaze svi oni koji su dostupni za kupovinu te njihovi osnovni podaci poput naziva, cijene i slike proizvoda. Korisnik može pregledavati proizvode i detalje istih kojim upravlja administrator stranice. Na slici 4.5 prikazana je ponuda proizvoda kao i mogućnost detaljnog prikaza. Prijavljeni i neprijavljeni korisnici mogu pristupiti ovoj stranici.



Slika 4.5: Pregled ponude proizvoda

4.2.4 *Detalji proizvoda*

Detaljan opis proizvoda omogućuje selekciju prilikom kupovine, privlači kupce jer zbog opširnijeg opisa pridodaje većem povjerenju. Korisnik detalju proizvoda može pristupiti odabirom tipke detalji koja će ga odvesti na prikaz detalja odabranog proizvoda. Samo na ovaj način moguće je pristupiti detaljima proizvoda. Na slici 4.6. prikazana je stranica detaljnog opisa proizvoda koja se otvara prilikom klika na odabrani proizvod.



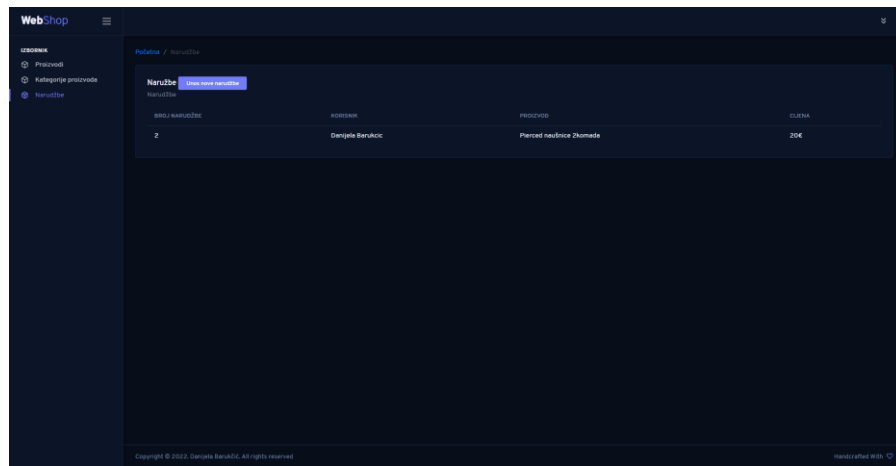
Slika 4.6: *Detalji proizvoda*

4.2.5 *Izvršavanje kupovine*

Kupovinu mogu obavljati samo registrirani korisnici. Na slici 4.8. prikazano je izvršavanje narudžbe registriranim korisnicima, stoga su pojedini podaci automatski popunjeni njegovim podacima. Potrebno je provjeriti je li korisnik prijavljen, ukoliko je prosljeđuje se objekt korisnika. Ukoliko bi narudžbu obavljao neregistrirani korisnik taj obrazac bio bi potpuno prazan. Njegove podatke također je potrebno spremi u bazu kako bi administrator imao uvid u podatke potrebne za slanje narudžbe. Razlika je u tome što će neregistrirani korisnik u tablici dobiti status gosta u svrhu jednostavnijeg upravljanja narudžbama.

4.2.6 Povijest narudžbi

Kako bi projekt u okviru ovog rada bio potpun, omogućen je pregled povijesti narudžbi i naručenih proizvoda. Takvoj stranici mogu pristupiti samo oni korisnici koji su prijavljeni te oni koji su prethodno obavili kupovinu. Prilikom pregleda prošlih narudžbi korisnik ima uvid u osnovne podatke o istoj, što je prikazano na slici 4.7.



ID	Korisnik	Proizvod	Cijena
2	Danijela Beraković	Plećica neudruga Zlomada	20€

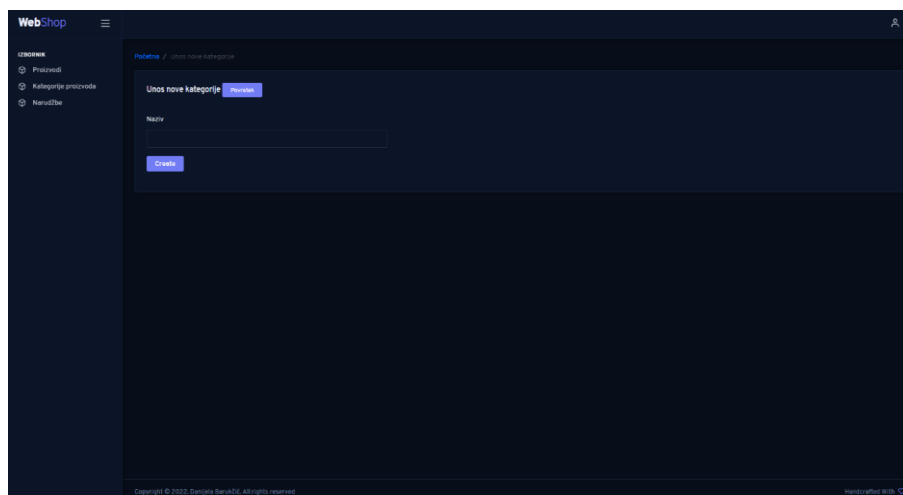
Slika 4.7: Povijest narudžbi.

4.3 Administratorska strana aplikacije

Dodavanje novog proizvoda, brisanje starih ili izmjene postojećih nije moguće obavljati kroz profil običnog korisnika, stoga je definirana uloga *admin* koja ima sve ovlasti za uređivanje ponude web trgovine. Da bi administrator mogao pristupiti stranici mora se prijaviti pomoću zadane lozinke i korisničkog imena. U odnosu na običnog korisnika, administrator ima dodatne mogućnosti poput upravljanja proizvodima, pregled registriranih korisnika te pregled narudžbi.

4.3.1 Upravljanje kategorijama

Kategorija je podatak koji pripada proizvodu te je pomoću nje lako odabrati proizvode za prikaz prema odabranoj kategoriji. Na slici 4.8 prikazan je pogled za dodavanje novih kategorije te brisanje ili izmjenu postojećih.



Slika 4.8: Unos kategorije proizvoda

4.3.2 Upravljanje proizvodima

Iz prikaza proizvoda dodana je mogućnost za brisanje ili uređivanje istoga. Ukoliko se odabere brisanja proizvoda, on više neće postojati u bazi podataka te neće biti vidljiv korisnicima. Ukoliko je potrebno izmijeniti pojedine podatke proizvoda, odabirom gumba za uređivanje, otvorit će se nova stranica gdje je moguće ažurirati podatke proizvoda. Na slici 4.9 se nalazi stranica za unos novog proizvoda.

The screenshot shows the 'Unos novog proizvoda' (Add new product) form in a dark-themed admin interface. The form is titled 'Unos novog proizvoda' and has a 'Povratak na proizvode' (Return to products) button. It contains several input fields: 'Ime' (Name) with the value 'Pierced', 'Veličina' (Size) with the value 'small', 'Cijena' (Price) with the value '20', 'Dostupnost' (Availability) with the value '5', 'Opis' (Description) with the value 'Ručno izrada, neudnice za svakodnevnu uporabu', 'Popust' (Discount) with the value '0', 'Slika' (Image) with a file selection button 'Odeberi datoteku' and the filename 'pierced.jpg', and 'Kategorija' (Category) with the value '3'. There is a 'Unesi' (Save) button at the bottom.

Slika 4.9: Unos novog proizvoda

Unos novog proizvoda omogućen svima koji imaju ulogu administratora. U programskom kodu 4.1. prikazano je korištenje *Authorize* atributa koji onemogućuje pristup korisnicima. Postavljen je prije implementacije metode *Create* u *ProductsControlleru*.

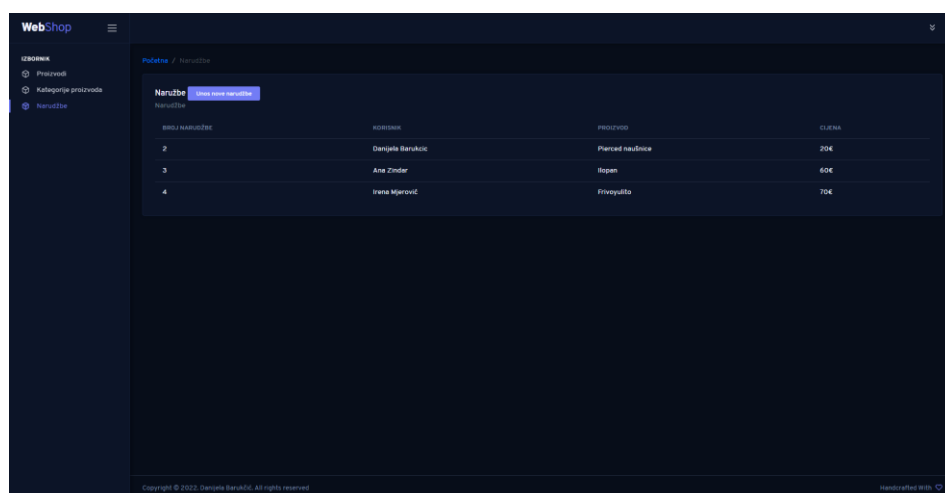
Programski kod 4.1: Korištenje atributa Authorize

```
[Authorize(Roles = ("Admin"))]  
public IActionResult Create()  
{  
    //ostali dio koda  
    return View();  
}
```

Slike proizvoda ne spremaju se u bazu podataka jer zahtijevaju puno više prostora za pohranu baze podataka. Također, spremanje u bazu narušilo bi performanse aplikacije te bi ubrzo došlo do popunjavanja iste. S obzirom da se ova ideja pokazala lošom, slike se pohranjuju lokalno u zasebnom direktoriju *images*.

4.3.3 Pregled narudžbi

Od važnosti je imati pregled na sve obavljene narudžbe kroz internet stranicu. One daju važne podatke o kupcu i naručenim proizvodima te služe kao dokumentacija i dokaz kupovine. Na slici 4.10 prikazan je popis svih obavljenih narudžbi kao i pojedini detalji iste.



ID	KODIRANJE	PROIZVOD	CIJENA
2	Daniela Barukic	Pierced naučnica	20€
3	Ana Zinder	Šapan	60€
4	Irena Mjericic	Privepalita	70€

Slika 4.10: Pregled narudžbi svih korisnika

5. Predložak naplate proizvoda

Gotovo sve današnje internet trgovine omogućuju korisnicima sigurno plaćanje proizvoda koje naručuju nekim od postojećih metoda plaćanja. Da bi trgovina privukla sve veći broj korisnika, poduzetnici se odlučuju na implementaciju *online* plaćanja. *Payment Gateway* je sustav sigurne autorizacije i naplate kartica. Kupcima omogućuje obavljanje sigurne, brze i jednostavan kupovine. Nužan je ukoliko tvrtka želi ponuditi *online* plaćanja i plaćanje karticama. Tehnologija provodi financijske podatke do potrebnih entiteta za autorizaciju plaćanja i premješta novac od kupca do trgovca. [8]

Neke od poznatih *Payment Gateway* opcija za hrvatske poduzetnike su *CorvusPay*, *WS Pay*, *PayPal*, *Pay Way (T-com)* te *Monri*. Sustavi nude *Form* i *API* način integracije kartičnog plaćanja odnosno spajanja na sustav naplate. Korištenjem *Form* integracije korisnik se privremeno preusmjerava na zaštićeni server gdje ispunjava podatke potrebne za autorizaciju kreditnih kartica. Nakon uspješne autorizacije, bit će vraćen na internet trgovinu. Za ovakvu vrstu naplate nije potrebno imati SSL (engl. *Secure Socket Layer*) certifikat dok je za *API* integraciju svakako potreban. Uporabom *API* pristupa, kupac izvršava kupovinu na serveru internet trgovine. Uneseni podaci korisnika šalju se na adresu *Payment Gatewaya* te slijedi preusmjeravanje korisnika na početnu stranicu ukoliko je plaćanje uspješno provedeno.

U sljedećim poglavljima detaljnije su opisani *Payment Gateway* sustavi *CorvusPay*, *WS Pay* te sustav *Stripe* poznat na svjetskoj razini.

5.1 CorvusPay

CorvusPay je prva licencirana institucija koja trgovcima i kupcima omogućuje virtmansko plaćanje. Na slici 5.1 prikazan je njegov prepoznatljiv logo. Suraduju s deset različitih banaka te njihove usluge mogu ugovoriti tvrtke registrirane u Hrvatskoj, Bosni i Hercegovini, Srbiji i Europskoj Uniji. Za mjesečni promet do 44.000,00 HRK mjesečna naknada iznosi 220,00 HRK + PDV.

Povećanjem mjesečnog prometa obračunava se nova naknada dodavanjem 0,5% na vrijednost mjesečnog prometa. [9] Nakon potpisivanja ugovora s bankom slijedi implementacija, testiranje sustava te aktivacija. Osjetljivi podaci korisnika nisu dostupni trgovcu jer se oni unose u *CorvusPay* formu, a ona je osigurana visokim stupnjem SSL enkripcije. Svi podaci koji se skladište, dodatno su zaštićeni kriptografijom.



Slika 5.1: Logo CorvusPaya Payment Gateway sustava

5.2 WSPay

Prema [10], WSPay, odnosno *Web Studio Payment Gateway* je hrvatski platni sustav koji omogućuje visok stupanj zaštite podataka te brzu naplatu. Logo po kojem je poznat prikazan je na slici 5.2. Za njegovo korištenje trgovci moraju ugovoriti prihvaćanje kartica s jednim ili više partner kartičarima, potrebno je priložiti određenu dokumentaciju te potpisati ugovor u svrhu ostvarivanja licence. Cijena osnovne licence ovog sustava iznosi 2.380,00 HRK + PDV godišnje. Omogućuje *WSPayAuto* i *WSPayForm* sustav spajanja koji se odabire pri ispunjavanju zahtjeva. Također, pri ispunjavanju zahtjeva poslodavac odabire između dva sustava za autorizaciju i naplatu kartica, pred autorizaciju i naplatu ili način direktne naplate.



Slika 5.2: Logo WSPay Payment Gateway sustava

5.3 Stripe

Stripe je *Payments Gateway* sustav poznat na svjetskoj razini. Koriste ga velike i male tvrtke koje svoje kupce imaju po cijelom svijetu. Jedini trošak jest provizijska naknada za uslugu naplate koja obično iznosi 1% - 3% što ovisi u kakvoj kartici je riječ te iz koje zemlje potječe naplata. Prihvatljive cijene i dodatne usluge samo su neke od prednosti korištenja

Stripea. Njegova jednostavna integracija API sučelja, programerima omogućuje proširenje usluga trgovine.



Slika 5.3: Logo Stripe Payment Gateway sustava

Programerski pristup, Stripe je učinio popularnim sustavom. Pruža vrlo jednostavnu uporabu te za njegovo korištenje kupci ne trebaju imati račun. Omogućuje pregled transakcija te što se s njima događa putem nadzorne ploče u oblaku. Ova snažna opcija privlači tvrtke kada je u pitanju obrada plaćanje te s njegovim daljnjim poboljšanjima mogao bi postati dominantni sustav naplate. [11]

6. ZAKLJUČAK

U okviru ovog rada predstavljene su osnovne mogućnosti rada internet trgovine. Implementirani su zahtjevi korisnika poput pretraživanja proizvoda, pregled detalja, registracija i prijava korisnika na stranicu te sami pristup vlastitom profilu uz mogućnost ažuriranja osnovnih podataka. Također, osim zahtjeva korisnika, ostvareni su i zahtjevi administratora stranice. Administrator ima mogućnost dodavanja novih te brisanje ili ažuriranje postojećih proizvoda i kategorija proizvoda. Statistički podaci vrlo su bitni za poslovanje, stoga je omogućen pregled prijave novih korisnika te pregled obavljenih narudžbi.

Programsko rješenje u okviru ovog rada ostvareno je, većim dijelom, uz pomoć *ASP.NET MVC*, *Entity Framework* i *Identity Framework* tehnologija. Rad projekta podijeljen je na administratorski i korisnički modul. Prijava korisnika ili administratora implementirana je uporabom *Identity Framework* tehnologije koja omogućuje kompletnu autorizaciju i autentifikaciju korisnika. Izrada baze podataka ostvarena je korištenjem *Entity Frameworka* koja omogućuje izradu tablica i njihovo međusobno povezivanje, pisanjem programskog koda.

Buduća nadogradnja ovog programskog rješenja bila bi implementacija potpunog izvršavanja narudžbe te odabir i aktivacija *Payment Gateway* sustava. Lista želja (engl. *wish list*) i mogućnost recenziranja proizvoda su neke od funkcionalnosti koju bi u budućnosti ovo programsko rješenje trebalo implementirati. Internet trgovina, budućnost je trgovanja te svako novo poboljšanje ostavlja pozitivan dojam na kupce.

7. LITERATURA

[1] Macrotrends. Amazon revenue 2010 – 2022 | AMZN: pročišćeni tekst [Online]. 2022.

Dostupno na:

<https://www.macrotrends.net/stocks/charts/AMZN/amazon/revenue>

[2] Firealt Studio. Top programming languages that will rule in 2022: pročišćen tekst

[Online]. 2022. Dostupno na:

<https://fireart.studio/blog/top-programming-languages-that-will-rule-in-2021/>

[3] Apptus. 10 Ways Covid-19 Affects eCommerce Growth in Europe: pročišćeni tekst

[Online]. 2022. Dostupno na:

<https://www.apptus.com/whitepapers/how-covid-19-affected-ecommerce-in-europe/>.

[4] Maja Nowak. Vue vs React in 2022 - Comparison of Two Most Popular JS

Frameworks: pročišćeni tekst [Online]. 08.04.2022. Dostupno na:

<https://www.monterail.com/blog/vue-vs-react>.

[5] GeeksforGeeks. What is Entity Framework in .NET Framework?: pročišćeni tekst

[Online]. 09 Aug, 2019. Dostupno na:

<https://www.geeksforgeeks.org/what-is-entity-framework-in-net-framework/>

[6] Visual Studio Code. Visual Studio Code FAQ: pročišćeni tekst [Online]. 09.01.2022.

Dostupno na: <https://code.visualstudio.com/docs/supporting/FAQ>

[7] Atlassian. What is Git: pročišćeni tekst [Online]. 2022. Dostupno na:

<https://www.atlassian.com/git/tutorials/what-is-git>

[8] Virtualna tvornica. 5 najboljih Payment Gateway Sistema: pročišćeni tekst [Online].

08.04.2015. Dostupno na:

<https://www.virtualna-tvornica.com/payment-gateway/>

[9] CorvusPay. FAQ: pročišćeni tekst [Online]. 2021. Dostupno na:

<https://www.corvuspay.com/novifaq/>

[10] Web Studio. WSPay Standard: pročišćeni tekst [Online]. 2021. Dostupno na:

<https://www.wspay.info/product/15/wspay-standard>

[11] Stripe. A complete payments platform for ecommerce: pročišćeni tekst [Online].
2022.

Dostupno na:

<https://stripe.com/en-hr/use-cases/ecommerce>

8. OZNAKE I KRATICE

CLR (engl. Common Language Runtime)

CMS (engl. Content management system)

CoC (engl. Convention over Configuration)

CSS (engl. Cascading style sheetS)

DI (engl. Dependency Injection)

ERM (engl. Entity - Relationship Modeling)

HTML (engl. HyperText Markup Language)

JIT (engl. Just-In-Time)

LINQ (engl. Language - Integrated Query)

MVC (engl. Model -View - Controller)

MVT (engl. Model – View - Template)

ORM (engl. Objects Relation Mapper)

PHP (engl. Hypertext Preprocessor)

RoR (engl. Ruby on Rails)

SEO(engl. Search Engine Optimization)

SPA (engl. Single Page Applications)

SSL (engl. Secure Socket Layer)

SQL (engl. Structured Query Language)

TDD (engl. Test Driven Development)

UI (engl. User interface)

Virtual DOM (engl. Virtual Document Object Model)

9. SAŽETAK

Naslov: Web trgovina zasnovana na .NET tehnologiji

Prema zadanim ishodima, u okviru ovog rada izgrađena je internet trgovina zasnovana na .NET tehnologiji. Internet trgovina, korisnicima i administratorima omogućuje osnovne funkcije za pravilno poslovanje putem interneta. Korisniku je omogućena registracija i prijava na stranicu, pregled proizvoda i detalja pojedinog, pretraživanje po kategoriji, dodavanje proizvoda u košaricu te pregled profila korisnika. Administrator kao uređivač stranice ima dodatne mogućnosti poput dodavanja proizvoda i kategorija, uređivanja ili brisanja postojećih, pregled povijesti narudžbi kao i njihovih detalja.

U svrhu poboljšanja programskog rješenja, u budućnosti je moguće implementirati platni sustav, listu želja gdje korisnik može spremati željene proizvode te registriranim korisnicima omogućiti recenziranje proizvoda.

Ključne riječi: ASP.NET MVC, internet trgovina, Entity Framework, Identity Framework

10. ABSTRACT

Title: Online store based on .NET technology


According to the expected outcomes, a .NET-based online store was created within the scope of this paper. The online store allows its users and website administrators to access basic functions needed to properly do business online. Users are able to register and log in to the site, view products and their details, search by categories, add products to their shopping carts and view their user profiles. The administrator as a website editor has access to the additional functions, such as adding new products and categories, editing or deleting existing ones, and reviewing order histories and their details.

In order to improve the software solution, in the future, it is possible to implement a payment system and a wish list that users could add their wanted products to, and to allow registered users to review products.

Key words: ASP.NET MVC, online store, Entity Framework, Identity Framework

IZJAVA O AUTORSTVU ZAVRŠNOG RADA

Pod punom odgovornošću izjavljujem da sam ovaj rad izradio/la samostalno, poštujući načela akademske čestitosti, pravila struke te pravila i norme standardnog hrvatskog jezika. Rad je moje autorsko djelo i svi su preuzeti citati i parafraze u njemu primjereno označeni.

Mjesto i datum	Ime i prezime studenta/ice	Potpis studenta/ice
U Bjelovaru, <u>24.10.2022.</u>	DANIJELA BARUKČIĆ	

Prema Odluci Veleučilišta u Bjelovaru, a u skladu sa Zakonom o znanstvenoj djelatnosti i visokom obrazovanju, elektroničke inačice završnih radova studenata Veleučilišta u Bjelovaru bit će pohranjene i javno dostupne u internetskoj bazi Nacionalne i sveučilišne knjižnice u Zagrebu. Ukoliko ste suglasni da tekst Vašeg završnog rada u cijelosti bude javno objavljen, molimo Vas da to potvrdite potpisom.

Suglasnost za objavljivanje elektroničke inačice završnog rada u javno dostupnom nacionalnom repozitoriju

DANIJELE BARUKČIĆ

ime i prezime studenta/ice

Dajem suglasnost da se radi promicanja otvorenog i slobodnog pristupa znanju i informacijama cjeloviti tekst mojeg završnog rada pohrani u repozitorij Nacionalne i sveučilišne knjižnice u Zagrebu i time učini javno dostupnim.

Svojim potpisom potvrđujem istovjetnost tiskane i elektroničke inačice završnog rada.

U Bjelovaru, 24.10.2022.

Daniela Jonković
potpis studenta/ice