

Izrada mrežne trgovine u programskom okviru Symfony

Lončarić, Boris

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:938033>

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

Download date / Datum preuzimanja: **2024-07-09**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Boris Lončarić

**IZRADA MREŽNE TRGOVINE U PROGRAMSKOM
OKVIRU *SYMPHONY***

ZAVRŠNI RAD

Čakovec, rujan 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

Boris Lončarić

**IZRADA MREŽNE TRGOVINE U PROGRAMSKOM
OKVIRU *SYMFONY***

**CREATION OF AN ONLINE STORE USING SYMFONY
FRAMEWORK**

ZAVRŠNI RAD

Mentorica:

dr. sc. Sanja Brekalo

Čakovec, rujan 2023.

MEĐIMURSKO VELEUČILIŠTE U
ČAKOVCU ODBOR ZA ZAVRŠNI RAD

Čakovec, 9. siječnja 2023.

država: Republika Hrvatska
Predmet: PHP programiranje-izborni

ZAVRŠNI ZADATAK br. 2022-RAČ-
R-32

Pristupnik. Boris Lončarić (0313023776)
studij: Redoviti preddiplomski stručni studij Računarstvo
Smjer: Inženjerstvo računalnih sustava i mreža


Zadatak: Izrada mrežne trgovine u programskom okviru *Symfony*
Opis zadatka:

Aplikacija se izrađuje u PHP-u korištenjem programskog okvira *Symfony*. Koristi se baza podataka za zapisivanje podataka koji se skupljaju od strane korisnika i administratora. Aplikacija se izrađuje na primjeru mrežnog dućana. Za aplikaciju se predviđa da korisnici koji su prodavači unose svu robu koja se prodaje u sustav te tako mrežno prikazuju robu koju je moguće naručiti online preko dostave. Pozadinski dio aplikacije omogućuje izdavanje računa i vođenje skladišta dućana. Administratori nakon naručivanja robe izdaju račune za *online* prodaju automatski ili sami izrađuju račune za prodaju u trgovini. Klijenti u aplikaciji mogu pregledavati vlastite narudžbe i ispisivati račune u PDF formatu.

Zadatak uručen pristupniku: 9. siječnja 2023.,
Rok za predaju rada: 20. rujna 2023.

Mentor:

Predsjednik
povjerenstva za završni
ispit:



dr. sc. San Brekalo, prof. v. š.

ZAHVALA

Želim zahvaliti dragoj mentorici dr. sc. Sanji Brekalo na usmjeravanju, vođenju i pomoći prilikom pisanja završnog rada. Isto tako, htio bih zahvaliti svim profesorima na Međimurskom veleučilištu u Čakovcu na prenesenom znanju, dobrim savjetima i ugodnom okruženju na predavanjima.

SAŽETAK

Ljudi danas sve više koriste mrežne trgovine za kupovinu zbog njihove praktičnosti, širokog izbora proizvoda i pristupačnosti. Stalno unapređenje tehnologije i korisničkog iskustva pridonijelo je njihovoj sve većoj popularnosti u modernom svijetu. Kupcima su trgovine nadohvat ruke i njihova kupovina više ne ovisi o fizičkim lokacijama trgovina.

Glavna ideja rada je napraviti jednostavnu mrežnu trgovinu za prodaju informatičke opreme, mobitela, laptopa, računala i ostalih proizvoda korištenjem programskog jezika PHP, programskog okvira *Symfony* te ostalih alata i tehnologija za izradu mrežnih trgovina. U trgovini koja je predstavljena u radu koristi se objektnoorijentirani pristup programiranju. Kako bi kod aplikacije bio bolje organiziran, koristi se arhitekturni obrazac MVC (eng. *Model View Controller*). MVC se koristi u razvoju aplikacija zbog kvalitetnije organizacije i boljeg razdvajanja koda, tako da se kod razdvaja u tri komponente: model, pogled i kontroler. Aplikacija koristi server XAMPP i bazu podataka *MySQL* za pohranu proizvoda, kategorija, podkategorija, narudžbi i košarica korisnika. Izgled trgovine je jednostavan kao i njeno korištenje, tako da se trgovinom mogu koristiti korisnici koji nemaju predznanja o radu na računalu. Za uređivanje izgleda stranica koriste se HTML, CSS i *Twig*, a za responzivni dizajn koristi se *Bootstrap*. Mrežna trgovina sastoji se od dva dijela, to su dio za administratore koji služi za upravljanje mrežnom trgovinom i dio za korisnike.

Glavne funkcije mrežne trgovine su mogućnosti prijave i registracije korisnika, mogućnost obnove lozinke, dodavanje proizvoda u košaricu i mogućnosti naručivanja. Korisnik račune za svoje narudžbe dobiva na adresu elektroničke pošte u PDF formatu nakon postavljanja narudžbe. Prijavljeni korisnici mogu naručivati proizvode, pregledavati svoje narudžbe i preuzimati račune u PDF formatu. Administrator ima pristup vlastitom sučelju za upravljanje sadržajem gdje može: dodavati, uređivati i brisati proizvode, kategorije, podkategorije, preuzeti račun za bilo koju narudžbu i staviti popust na proizvode. U radu je objašnjen dizajn i izrada baze podataka, korištenje mrežne trgovine te korištenje tehnologija, alata i paketa kod izrade trgovine.

Ključne riječi: *Symfony, PHP, mrežna trgovina, MySQL, MVC.*

SADRŽAJ

1. UVOD	1
2. KORIŠTENE TEHNOLOGIJE, ALATI I PAKETI	2
2.1. <i>Symfony</i>	3
2.2. PHP	5
2.3. HTML i CSS	6
2.4. <i>Twig</i>	7
2.5. XAMPP	8
2.6. <i>Visual Studio Code</i>	9
2.7. Baze podataka	10
2.8. <i>Doctrine ORM</i>	11
3. MREŽNA TRGOVINA	12
3.1. Dizajn baze podataka	13
3.1.1. Spajanje baze podataka s aplikacijom	15
3.1.2. Klase i objektno-relacijsko mapiranje	16
3.2. Kontroleri	18
3.3. Administrativni dio trgovine	20
3.3.1. Administrativno sučelje	22
3.4. Korisnički dio trgovine	27
4. ZAKLJUČAK	31
5. POPIS LITERATURE	32
6. PRILOZI	33

1. UVOD

Trgovine danas koriste internet za prodaju i reklamiranje više nego ikad prije. Isto tako sve više ljudi koristi internet za kupovinu iz razloga što trgovine na internetu nude razne mogućnosti, kao na primjer mogućnost dostave, kupovinu proizvoda bez odlaska u trgovinu, korištenje akcijskih kodova za popust, posebne ponude i ostale nagrade za registrirane korisnike.

Kako bi trgovine mogle napredovati u svom poslovanju i ponuditi svoje proizvode većem broju ljudi, uz otvaranje trgovine na nekoj fizičkoj lokaciji, otvaraju i trgovinu na internetu. Upotreba mrežne trgovine znatno povećava prodaju i ponekad smanjuje gužve u manjim trgovinama.

Ideja ovog rada je izraditi mrežnu trgovinu za prodaju informatičke opreme, mobitela, tableta i ostalih proizvoda pomoću programskog jezika PHP, programskog okvira *Symfony* koji omogućuje izradu modernih aplikacija te drugih alata i tehnologija za izradu mrežnih trgovina.

U izradi dizajna, uz HTML, CSS i *Bootstrap*, koristi se *Twig*. Moderan i siguran predložak za PHP koji omogućuje lakše upravljanje sadržajem i ubrzava izradu stranica za prikaz. U radu su također objašnjeni dizajn i izrada baze podataka te korištenje mrežne trgovine.

2. KORIŠTENE TEHNOLOGIJE, ALATI I PAKETI

Kako bi bilo moguće izraditi mrežnu trgovinu, potrebno je koristiti razne alate, tehnologije i pakete. *Symfony* nudi veliki izbor alata i paketa koji olakšavaju izradu mrežnih trgovina, omogućavaju rad s bazom podataka i upravljanje podacima. U nastavku rada će biti opisani korišteni alati, programski jezici, tehnologije i paketi, njihove mogućnosti i funkcionalnosti te izgled njihovih sučelja. Korištene tehnologije, programski jezici, alati i paketi su: *XAMPP*, *PHPMyAdmin*, *Visual Studio Code*, PHP, HTML, CSS, *Bootstrap*, *Symfony*, *EasyAdmin*, *Doctrine*, *Twig*.

2.1. *Symfony*

Symfony je jedan od najpopularnijih i najkorištenijih PHP programskih okvira za izradu mrežnih aplikacija i internet-stranica. Proizvela ga je tvrtka pod imenom *SensioLabs*, 2005. godine. *Symfony* je programski okvir otvorenog koda i poznat je po svojoj fleksibilnosti, skalabilnosti, robusnosti. *Symfony* koristi arhitekturu temeljenu na komponentama gdje se komponente mogu koristiti zasebno ili kombinirati za stvaranje mrežne aplikacije. Ovakva modularnost omogućuje programerima da odaberu najbolje komponente za svoje projekte. [1]

Slijedi arhitektonski obrazac Model-Pogled-Kontroler (eng. *Model-View-Controller*) koji pomaže u odvajanju koda aplikacije u tri različita sloja: model koristimo za rukovanje podacima, pogled za prezentaciju podataka korisnicima i kontroler za logiku aplikacije. Ovo odvajanje čini kod aplikacije organiziranijim i lakšim za održavanje. Ostale bitne značajke programskog okvira *Symfony*:

- *Doctrine ORM*: *Symfony* koristi *Doctrine* kao svoj zadani alat za objektno-relacijsko mapiranje (ORM). *Doctrine* programerima omogućuje rad s bazama podataka na objektnoorijentiran način, time olakšava upravljanje interakcijama i odnosima (vezama) podataka. [1]
- *Twig*: *Symfony* koristi *Twig* za generiranje prikaza. *Twig* je poznat po svojoj jednostavnosti i sigurnosti jer provodi stroge standarde kodiranja kako bi spriječio uobičajene sigurnosne propuste, kao što je napad skriptama (eng. *Cross-Site Scripting*) (XSS).
- *Symfony-konzola*: alat naredbenog retka pod nazivom *Symfony Console*. [1] Konzola pojednostavljuje razne zadatke kod razvoja aplikacija, uključujući pokretanje migracija, generiranje koda i upravljanje konfiguracijom aplikacije.
- Sigurnost: *Symfony* stavlja naglasak na sigurnost. Omogućuje ugrađene sigurnosne značajke, uključujući autentifikaciju, autorizaciju i zaštitu od uobičajenih sigurnosnih prijetnji kao što su krivotvorenje zahtjeva (CSRF) i ubacivanje *SQL* skripti. [1]

- **Fleksibilnost:** omogućuje programerima da prilagode gotovo svaki aspekt programskog okvira kako bi odgovarao potrebama njihovog projekta. Fleksibilnost je jedan od glavnih razloga zašto se *Symfony* često bira za velike, složene mrežne aplikacije.
- **Paketi (eng. *Boundles*):** *Symfony*-aplikacije često se izrađuju pomoću paketa, paketi sadrže specifične značajke ili funkcionalnosti i mogu se lako integrirati u *Symfony*-projekte, pružaju niz mogućnosti kao što su kreiranje paginacije ili izradu PDF datoteka iz HTML predloška.

Programski okvir imao je nekoliko izdanja, a *Symfony 2* i *Symfony 3* su značajne prekretnice. *Symfony 4* predstavio je pojednostavljeniji i fleksibilniji okvir, dok su se *Symfony 5* i naredne verzije usredotočile na poboljšanja performansi i jednostavnost korištenja. Slika 1. prikazuje logo programskog okvira *Symfony*.



Slika 1. Programski okvir *Symfony*

Izvor: <https://symfony.com/logo> (13.8.2023.)

2.2. PHP

PHP (engl. *Hypertext Preprocessor*) je skriptni jezik koji se izvršava na strani poslužitelja [2], otvorenog je koda. Poznat je po jednostavnosti korištenja pa se često koristi za izradu dinamičnih internetskih stranica, a omogućuje razvijanje internet-stranica ili mrežnih aplikacija koristeći tehnike proceduralnog i objektnoorijentiranog programiranja.

Sintaksa programskog jezika slična je drugim programskim jezicima kao što su C++ i Java. [2] PHP skripte mogu biti jednostavne naredbe ili složeni programi. PHP skripte obrađuju se prije učitavanja stranice, na taj način korisnici ne mogu vidjeti sadržaj skripte ni kod unutar elemenata stranice već samo HTML elemente koji su rezultat izvođenja skripte [2]. Unutar PHP datoteke, PHP kod moguće je kombinirati s HTML, CSS i *Javascript*-kodom. U datoteci se PHP kod obilježava posebnom oznakom što ga čini prepoznatljivim PHP procesoru prilikom izvođenja skripte. Slika 2. prikazuje primjer PHP skripte.

```
<?php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\Routing\Annotation\Route;
use Symfony\Component\Security\Http\Authentication\AuthenticationUtils;

class SecurityController extends AbstractController
{
    #[Route(path: '/login', name: 'app_login')]
    public function login(AuthenticationUtils $authenticationUtils): Response
    {
        if ($this->getUser()) {
            return $this->redirectToRoute('app_home');
        }

        // dohvati gresku kod prijave ako postoji
        $error = $authenticationUtils->getLastAuthenticationError();
        // zadnji username koji je korisnik unio
        $lastUsername = $authenticationUtils->getLastUsername();

        return $this->render('security/login.html.twig', ['last_username' => $lastUsername, 'error' => $error]);
    }
}
```

Slika 2. Primjer PHP skripte

Izvor: autor

2.3. HTML i CSS

HTML (skraćeno od eng. *Hypertext Markup Language*) je standardni jezik za označavanje koji se koristi za stvaranje i strukturiranje sadržaja na internetu. [3] HTML se sastoji od skupa elemenata i oznaka koji se koriste za oblikovanje i strukturu sadržaja. HTML elementi organizirani su hijerarhijski, s korijenskim elementom (`html`) koji sadržava dva glavna odjeljka: zaglavlje (eng. *head*) koje sadrži metapodatke i skripte te tijelo (eng. *body*) u kojem se nalazi sadržaj koji je vidljiv na stranici. [3]

HTML kod upotpunjuje se CSS kodom (eng. *Cascading Style Sheets*) za uređivanje i stiliziranje izgleda. [3] CSS je stilski jezik koji se koristi za opisivanje prezentacije i vizualnog izgleda HTML dokumenata. Koristi se za kontrolu načina na koji se prikazuje sadržaj HTML dokumenta na internetu, uključujući izgled, boje, fontove, boje fontova, razmake i još mnogo toga. Jedna od najbitnijih uloga primjene CSS-a je responzivni dizajn koji omogućuje prilagodbu sadržaja veličini ekrana na kojem se prikazuje. HTML i CSS zajedno čine temeljne građevne blokove svake stranice na internetu, omogućujući stvaranje vizualno privlačnog i interaktivnog sadržaja.

2.4. Twig

Kod razvoja mrežnih aplikacija, bitno je stvarati dinamičan i vizualno privlačan sadržaj. Međutim, održavanje čistog i organiziranog koda za postizanje ovog cilja može biti poprilično izazovno. *Symfony*, *Laravel* i ostali programski okviri koriste alate i mehanizme za kreiranje predložaka. *Symfony* kao zadani alat za kreiranje predložaka koristi *Twig*.

Twig je moderni i popularni mehanizam za stvaranje predložaka. Prilagođen je dizajnerima i programerima. [4] Drži se standardnih PHP načela i najbolje prakse u pisanju koda uz korištenje jednostavne sintakse. *Twig* sadrži funkcionalnosti korisne za izradu predložaka tako da smanjuje količinu koda na minimum. [4] Pomaže razvojnim programerima u kreiranju HTML, XML ili drugih vrsta dokumenata jer olakšava upravljanje dinamičnim sadržajem. Na uobičajenim postavkama čitav kod napisan u *Twig* predlošku zaštićen je od XSS napada (eng. *Cross-Site Scripting*). [4] Kod kompleksnijih aplikacija kod napisan u *Twig*-predlošku znatno smanjuje opterećenje aplikacije u usporedbi s običnim PHP kodom. Slika 3. prikazuje primjer koda iz *Twig*-predloška.

```
<p>Proizvodi</p>
<ol>
  {% for item in items %}
    <li>{{item.product}} x{{item.kolicina}} {% if item.getProduct.getPopust != 0 %}, popust: {{item.getProduct.getPopust}}% {% endif %}
    </li>
  {% endfor %}
</ol>

[[ form(form, {attr: {novalidate: 'novalidate'}}) ]]
<h5>Cijena s PDV-om: {{cijena}} &euro;</h5>
<p>Cijena dostave: 4.00 EUR</p>
<h5>Ukupno: {{cijena + 4}} &euro;</h5>

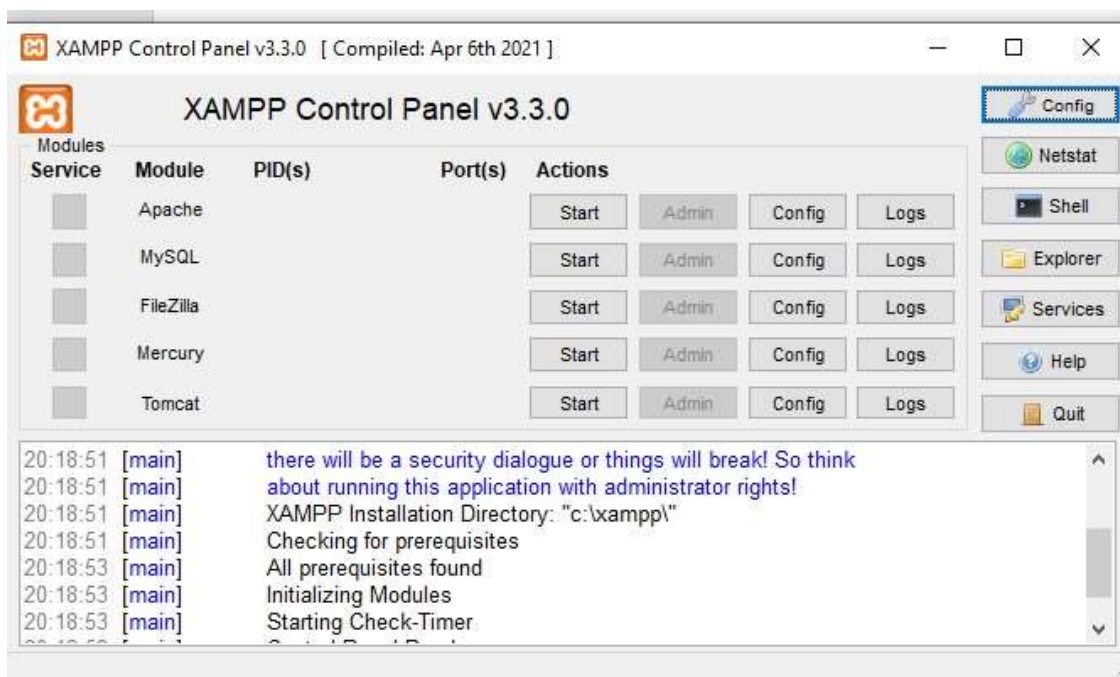
</div>
{% endblock %}
```

Slika 3. Primjer *Twig*-koda

Izvor: autor

2.5. XAMPP

XAMPP (*Cross-Platform* (X), *Apache* (A), *Maria DB* (M), *PHP* (P), *Perl* (P)) je alat otvorenog koda koje se koristi za postavljanje lokalnog poslužitelja na osobnom računalu. *XAMPP* razvila je tvrtka *Apache Friends*, a radi na različitim operativnim sustavima, uključujući: *Windows*, *macOS*, *Linux*. Integrira programske jezike *PHP* i *Perl*. Server *Apache* poznat je po svojim performansama, sigurnosti i fleksibilnosti što ga čini najkorištenijim poslužiteljem za prikazivanje mrežnog sadržaja. *XAMPP* koristi se za razvoj i testiranje mrežnih aplikacija ili stranica prije nego što se postave na udaljeni poslužitelj. Instalacija *XAMPP*-a jednostavna je kao i korištenje upravljačke ploče. Slika 4. prikazuje sučelje *XAMPP* kontrolne ploče.

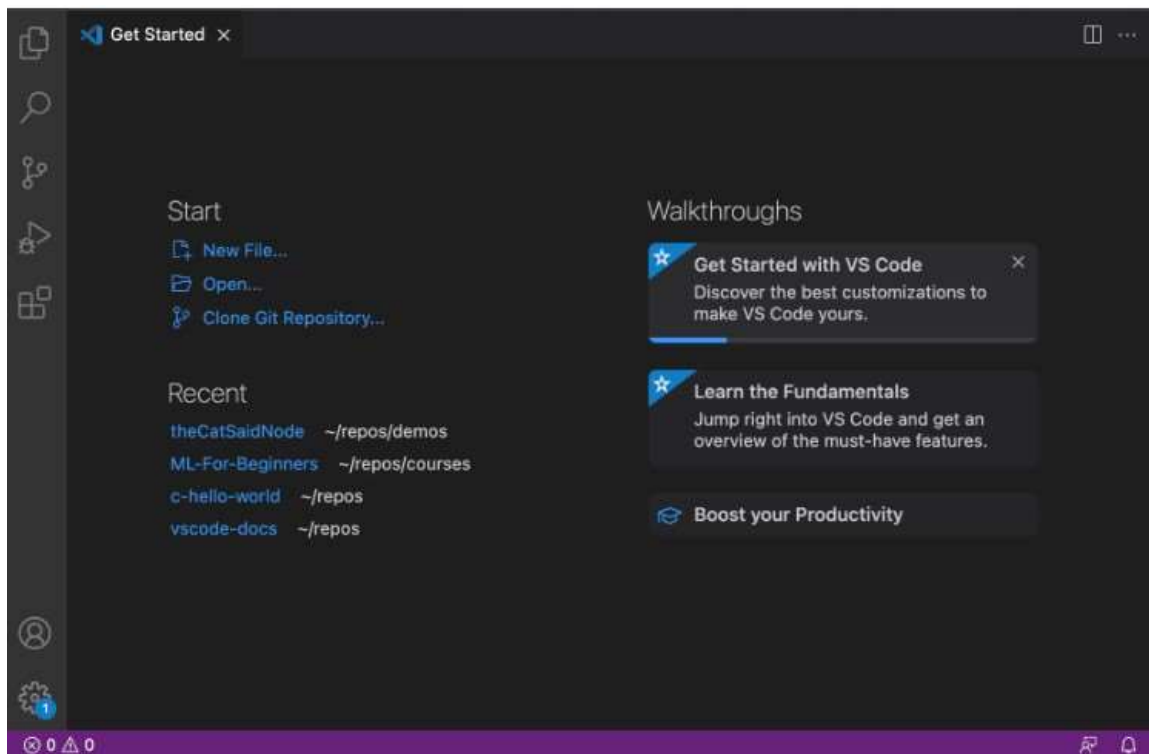


Slika 4. Sučelje *XAMPP* kontrolne ploče

Izvor: autor

2.6. *Visual Studio Code*

Visual Studio Code je besplatan program za uređivanje teksta, otvorenog koda. [5] Razvila ga je tvrtka *Microsoft*, a dostupan je za sve operacijske sustave. Alat nudi mogućnost proširenja za razne programske jezike. Posebno je popularan među razvojnim programerima zbog mnogo raznih dodataka i proširenja koje je moguće instalirati. Funkcionalnosti uređivača moguće je prilagoditi instaliranjem proširenja. [5] Proširenja pokrivaju širok raspon funkcionalnosti od kojih su neke: dodatna jezična podrška, teme za promjenu izgleda sučelja, korištenje razvojnih alata. VSC je izvrstan za razvoj PHP aplikacija, nudi značajke kao što su isticanje sintakse, uparivanje zagrada i dovršavanje koda. Slika 5. prikazuje sučelje VSC alata za uređivanje teksta.



Slika 5. Sučelje uređivača *Visual Studio Code*

Izvor: autor

2.7. Baze podataka

Baze podataka su bitan dio modernih informacijskih sustava. Služe kao strukturirana spremišta za pohranjivanje, organiziranje i dohvaćanje podataka. U današnjem svijetu koji se temelji na podacima, baze podataka igraju ključnu ulogu. Svaka stranica na internetu ili mrežna aplikacija koristi neku vrstu baze podataka za pohranjivanje podataka. Podaci se u bazu podataka pohranjuju u obliku neovisnom o aplikacijama koje ih koriste. Sustav za upravljanje bazom podataka (SUBP, eng. *DMBS – Database Management System*) je programska podrška koja izvodi sve operacije nad bazom podataka. Primjeri operacija koje se mogu izvoditi nad bazom podataka su kreiranje strukture podataka, brisanje, mijenjanje i dohvaćanje podataka, administracija baze i dr. Sustav za upravljanje bazom podataka brine o fizičkom smještaju podataka, obnovi podataka nakon rušenja baze podataka i administraciji sustava.

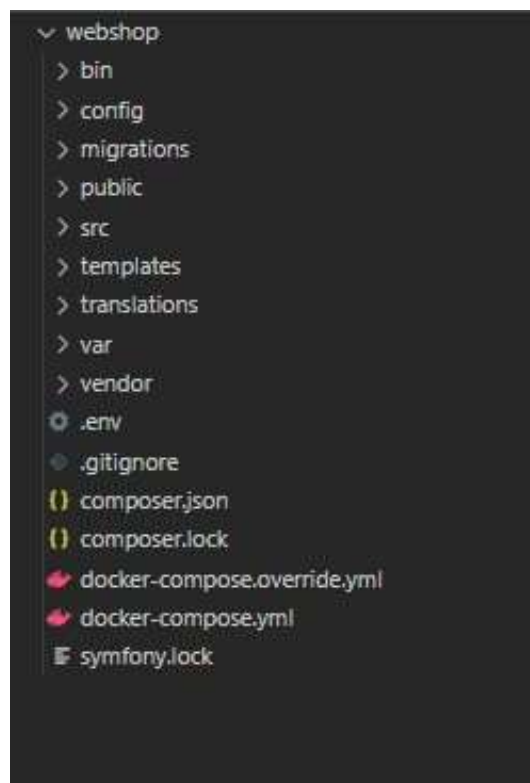
U ovom radu baza podataka koristi se za pohranjivanje podataka o proizvodima, kategorijama, podkategorijama, košaricama korisnika i narudžbama.

2.8. Doctrine ORM

Symfony pruža sve alate koji su potrebni za rad s bazama podataka kod izrade mrežnih aplikacija ili stranica. *Doctrine* je set biblioteka za rad s bazama podataka. [6] Omogućuje rad s relacijskim bazama podataka kao što su *MySQL* i *PostgreSQL*. Programerima omogućuje rad sa zapisima baze podataka kao da su obični PHP objekti. U *Doctrineu* entiteti su PHP klase koje predstavljaju tablice baze podataka. [6] Svaki entitet odgovara tablici u bazi podataka sa svojstvima klase koja se preslikavaju na stupce tablice. [6] *Doctrine* podržava različite vrste odnosa između entiteta kao što su: jedan na jedan, jedan na više i više na više. Biblioteka koristi odgođeno učitavanje, što znači da se povezani entiteti ne učitavaju iz baze prije nego što im se stvarno pristupi. To poboljšava performanse aplikacije smanjenjem nepotrebnih upita. *Doctrine* se zbog svoje praktičnosti i jednostavnosti korištenja često koristi s popularnim PHP programskim okvirima kao *Symfony* i *Laravel*.

3. MREŽNA TRGOVINA

Nove *Symfony*-aplikacije kreiraju se pokretanjem naredbe iz naredbenog retka u VSC uređivaču. Upisivanjem i izvršavanjem naredbe "*symfony new*" iz željenog direktorija kreira se nova aplikacija u tom direktoriju. Slika 6. prikazuje sadržaj direktorija *webshop* nakon kreiranja nove *Symfony*-aplikacije.



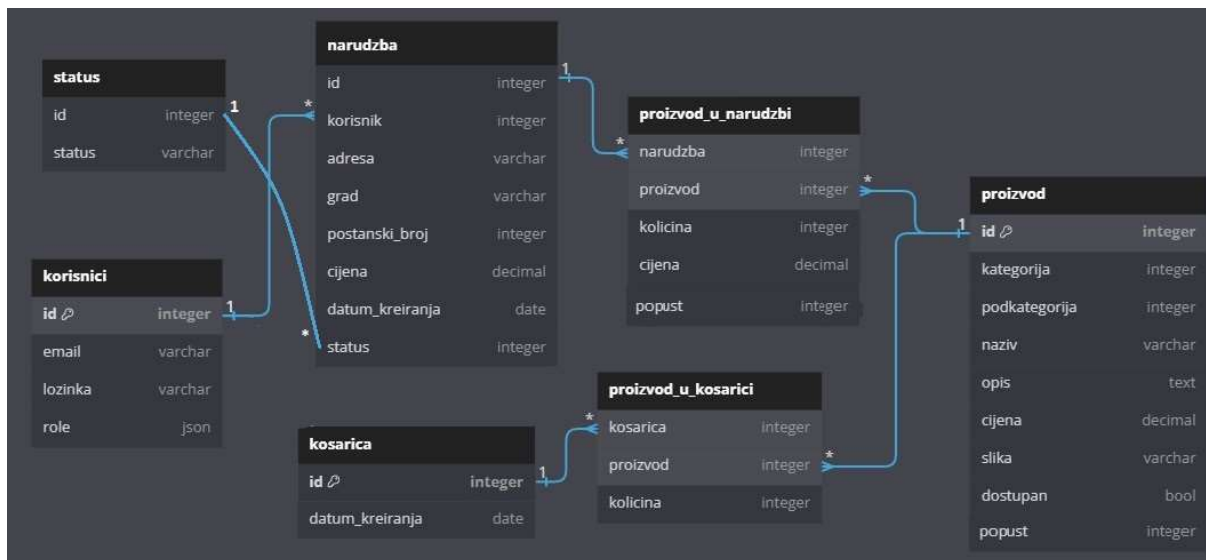
Slika 6. Sadržaj direktorija aplikacije

Izvor: autor

3.1. Dizajn baze podataka

Prije početka izrade aplikacije potrebno je dizajnirati i izraditi bazu podataka, odrediti sve tipove podataka i relacije (veze između tablica).

Slika 7. prikazuje shemu baze podataka koja se koristi u izradi mrežne trgovine.



Slika 7. Shema baze podataka

Izvor: autor

Svaki korisnik može imati jednu košaricu, a svaka košarica sadrži jedan ili više različitih proizvoda. Korisnik može imati više narudžbi, a svaka narudžba sadrži jedan ili više različitih proizvoda. Košarica i narudžba ne mogu imati više istih (dvostrukih) proizvoda već se ponovnim dodavanjem proizvoda u košaricu mijenja samo količina.

Tablica za korisnike sastoji se od polja "id", on je numerički cijeli broj i ujedno primarni ključ tablice. Primarni ključevi čine glavnu razliku između tablica u bazi podataka. Sljedeća polja od kojih se tablica "korisnik" sastoji su "email", "lozinka" i "role" (uloga korisnika, može biti korisnik ili administrator). "Email" i "lozinka" su polja tipa *VARCHAR* (varijabilni znak), a uloga (eng. *Role*) je polje tipa *JSON* (znakovni niz). Tablica "košarica" sadrži polja "id" i "datum_kreiranja". "Id" je primarni ključ tipa *INTEGER*, "datum_kreiranja" označava datum kreiranja košarice, košarica se kreira kad korisnik prvi put stavi proizvod u košaricu. Tablica "proizvod_u_košarici" povezuje tablice "košarica" i "proizvod". Polja koja sadrži su "košarica", "proizvod" i "količina", tipa *INTEGER*. "Košarica" i "proizvod" su vanjski ključevi dok

"količina" označava broj pojedinog proizvoda u košarici. "Popust" je polje tipa *INTEGER*, u to polje se sprema popust svakog proizvoda u cjelobrojnom numeričkom obliku.

Tablica "proizvod" sadrži polja: "id", "kategorija", "podkategorija", "naziv", "opis", "cijena", "slika", "dostupan" i "popust". "Id" je podatak tipa *INTEGER* i ujedno primarni ključ tablice. "Kategorija" i "podkategorija" također su podaci tipa *INTEGER* i predstavljaju vanjske ključeve, povezuju proizvode s pripadajućim kategorijama i podkategorijama. Tablica "proizvod_u_košarici" sadrži polja: "narudžba", "proizvod", "količina", "cijena", "popust". "Narudžba" i "proizvod" su vanjski ključevi tablice tipa *INTEGER*. Polje "narudžba" povezuje svaki proizvod u narudžbi (naručeni proizvod) s narudžbom kojoj pripada. Polje "proizvod" povezuje svaki proizvod u narudžbi s proizvodom u skladištu trgovine. Polje "popust" je tipa *INTEGER* i služi za spremanje podataka o popustu proizvoda. Tablica "narudžba" sadrži polja: "id", "korisnik", "adresa", "grad", "poštanski_broj", "cijena" i "datum_kreiranja". Polje "id" je tipa *INTEGER*. Predstavlja primarni ključ tablice. "Korisnik" je polje koje služi kao vanjski ključ, tj. veza između korisnika i narudžbe. Svaki korisnik može imati više postavljenih narudžbi, ali svaka narudžba može pripadati samo jednom korisniku. "Adresa" i "grad" su podaci tipa *VARCHAR* i služe za pohranjivanje grada i adrese na koju je korisnik naručio proizvode. "Poštanski broj" je polje tipa *INTEGER*. "Cijena" je polje tipa *DECIMAL*, u to se polje pohranjuje ukupna cijena svih naručenih proizvoda. Tablica "status" povezuje se s tablicom "narudžba" preko vanjskog ključa "status". Tablica "status" sadrži polje "id" koje je primarni ključ i polje "status", svaka narudžba mora imati status koji može biti: zaključana, poslana, zaprimljena i poništena.

3.1.1. Spajanje baze podataka s aplikacijom

Prije početka izrade aplikacije potrebno je povezati novoizrađenu aplikaciju s bazom podataka. Konfiguracija za spajanje baze s aplikacijom nalazi se u datoteci za upravljanje varijablama okruženja koju *Symfony* kreira kod stvaranja nove aplikacije (datoteka `.env`). U toj datoteci potrebno je navesti vrstu baze podataka, korisnika, naziv ili adresu baze i lozinku ako postoji. Slika 8. prikazuje konfiguraciju za spajanje s bazom podataka *MySQL*.

```
4 # IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml
5 #
6 # DATABASE_URL="sqlite://%kernel.project_dir%/var/data.db"
7 # DATABASE_URL="mysql://app:!ChangeMe!@127.0.0.1:3306/app?serverVersion=8.0.32&charset=utf8mb4"
8 # DATABASE_URL="mysql://root:@127.0.0.1:3306/webshop?serverVersion=10.11.2-MariaDB&charset=utf8mb4"
9 #DATABASE_URL="postgresql://app:!ChangeMe!@127.0.0.1:5432/app?serverVersion=15&charset=utf8"
0 ###< doctrine/doctrine-bundle ###
```

Slika 8. Konfiguracija za spajanje baze podataka

Izvor: autor

Nakon povezivanja baze podataka s aplikacijom, mogu se izvršavati osnovni zahtjevi prema bazi, kao što su: dohvaćanje, dodavanje, brisanje i uređivanje podataka. Navedene akcije koriste se za upravljanje sadržajem u bazi podataka i zajedno se nazivaju *CRUD* (eng. *Create, Read, Update, Delete*).

3.1.2. Klase i objektno-relacijsko mapiranje

Korištenjem *Doctrine*ovog objektno-relacijskog mapiranja moguće je povezati apstraktne klase u aplikaciji s tablicama u bazi podataka. Klase koje su potrebne za izradu mrežne trgovine su: korisnik, admin, kategorija, podkategorija, košarica, proizvod u košarici, narudžba i proizvod u narudžbi. Navedene klase kreiraju se preko *Symfony*-konzole upisivanjem i izvršavanjem naredbe: "symfony console make:entity *naziv*". Slika 9. prikazuje dio koda u klasi proizvod.

```
#[ORM\Entity(repositoryClass: ProductRepository::class)]
class Product
{
    #[ORM\Id]
    #[ORM\GeneratedValue]
    #[ORM\Column]
    private ?int $id = null;

    #[ORM\Column(length: 255)]
    private ?string $naziv = null;

    #[ORM\Column(type: Types::TEXT)]
    private ?string $opis = null;

    #[ORM\Column(type: Types::DECIMAL, precision: 10, scale: 2)]
    private ?string $cijena = null;

    #[ORM\Column(length: 255)]
    private ?string $slika = null;

    #[ORM\ManyToOne(inversedBy: 'products')]
    private ?Category $kategorija = null;

    #[ORM\ManyToOne(inversedBy: 'products')]
    private ?Subcategory $podkategorija = null;

    #[ORM\OneToMany(mappedBy: 'product', targetEntity: OrderItem::class)]
    private Collection $orderItems;

    #[ORM\OneToMany(mappedBy: 'product', targetEntity: CartItem::class)]
    private Collection $cartItems;

    #[ORM\Column]
    private ?bool $dostupan = null;

    #[ORM\Column(nullable: true)]
    private ?int $popust = null;
}
```

Slika 9. Dio koda u klasi proizvod

Izvor: autor

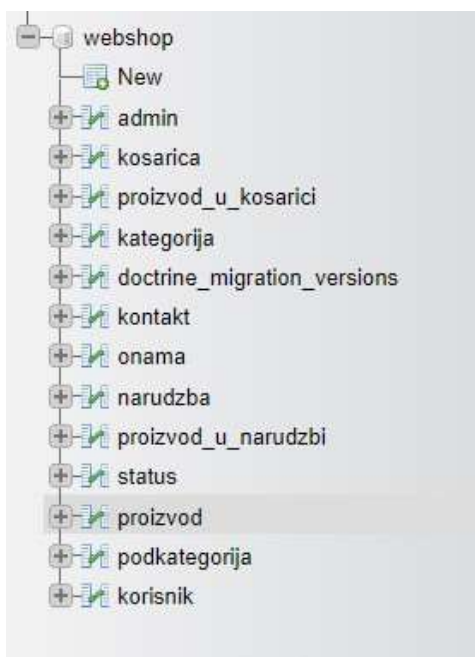
Upisivanjem i izvršavanjem naredbe: "symfony console make:migration", stvara se novi direktorij pod nazivom *Migrations*. U direktoriju *Migrations* također je stvorena datoteka koja sadrži SQL izjavu za kreiranje potrebnih entiteta u bazi podataka. Svi entiteti (klase) koji su stvoreni pomoću konzole postat će tablice u bazi podataka. Migracija prema bazi pokreće se upisivanjem i izvršavanjem naredbe: "symfony console doctrine migrations:migrate". Nakon uspješne migracije stvorena je tablica "proizvod" s poljima iz kreirane klase. Slika 10.



Slika 10. Tablica "proizvod"

Izvor: autor

Ostale potrebne klase (entiteti) koje se koriste u izradi mrežne trgovine kreirane su na isti način pomoću konzole, a nakon uspješne migracije dobivene su tablice u bazi podataka prikazane na slici 11.



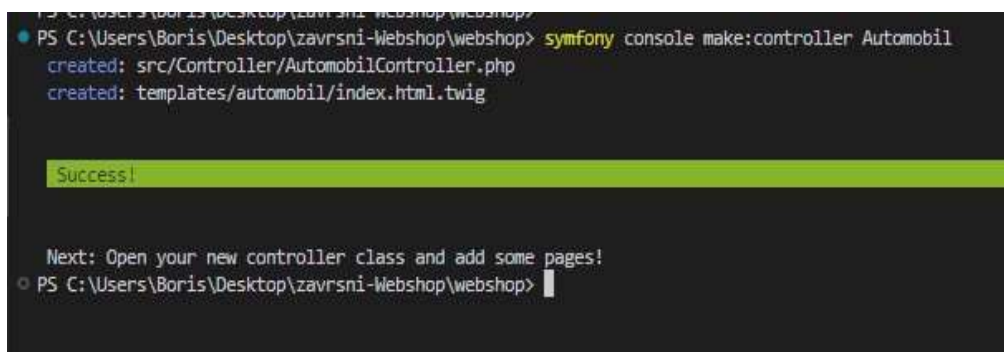
Slika 11. Tablice u bazi podataka

Izvor: autor

3.2. Kontroleri

Kontroleri služe kao posrednici između korisničkog sučelja (eng. *View*) i podataka aplikacije (eng. *Model*) i osiguravaju da aplikacija prikladno reagira na korisničke radnje. Oni rukuju korisničkim unosom, obrađuju ga i određuju kako se podaci trebaju ažurirati i koji pogled je potrebno prikazati kao odgovor na taj unos.

U programskom okviru *Symfony* kreiranje kontrolera radi se pomoću konzole. Upisivanjem naredbe: "symfony console make:controller naziv", kreira se kontroler u direktoriju *Controller*. Kako bi se osigurala ispravna funkcionalnost aplikacije, potrebno je dodati kontrolere za: navigacijsku traku, košaricu, narudžbe, proizvode, registraciju, prijavu i korisnike. Kreiranjem novog kontrolera *Symfony* automatski kreira pogled (*Twig*-datoteku) u direktoriju *Templates*. Kreiranje kontrolera pomoću konzole prikazano je na slici 12.



```
PS C:\Users\Boris\Desktop\završni-Webshop\webshop> symfony console make:controller Automobil
created: src/Controller/AutomobilController.php
created: templates/automobil/index.html.twig

Success!

Next: Open your new controller class and add some pages!
PS C:\Users\Boris\Desktop\završni-Webshop\webshop>
```

Slika 12. Primjer kreiranja kontrolera

Izvor: autor

Na ovaj način kreirani su svi gore navedeni kontroleri. Sadržaj navigacijske trake mijenja se ovisno o prijavljenom korisniku. Ako je korisnik prijavljen kao administrator, kontroler navigacijske trake dopušta mu ulazak u administrativno sučelje. Isto tako, kontroler navigacije dohvaća košaricu korisnika iz sesije. Košarica ima vlastiti kontroler koji upravlja njenim sadržajem, omogućuje dodavanje ili brisanje proizvoda u košarici te povećavanje količine proizvoda. Kontroler za narudžbe omogućuje korisnicima naručivanje proizvoda, kreira obrazac za naručivanje i validira unos adrese, grada i poštanskoga broja. Nakon uspješne narudžbe šalje korisnicima *e-mail* s potvrdom narudžbe. Kontroler za prijavu upravlja prijavom korisnika, dohvaća korisnika iz baze podataka te provjerava lozinku i ulogu (eng. *Role*). Kontroler za registraciju upravlja registracijom novih korisnika, generira formu za unos podataka, validira unesene podatke i dodaje korisnika u bazu podataka ako je unos ispravan.

Dio koda za dohvaćanje korisnika i košarice u navigacijskoj traci, slika 13.

```
//dohvatimo korisnika
if($this->getUser()) $user=$this->getUser();
    else $user=null;
$categories=$categoryRepository->findAll();
$session = $request->getSession();
$cart=$session->get('kosarica'); //dohvatimo kosaricu
if (!$cart) {
    $cart= new Cart();
    $cart->setCreatedAt(new \DateTimeImmutable('now'));
    $em->persist($cart);
    $em->flush();
    $session->set('kosarica',$cart->getId());
}
$carts=$carts->find($session->get('kosarica'));
$brojProizvoda=0;
if($cart){
    if(count($cart->getCartItems()) > 0){
        $brojProizvoda= $brojProizvoda + count($cart->getCartItems());
    }
} //vratimo response
return $this->render('navigation.html.twig', [
    'brojProizvoda' => $brojProizvoda,
    'categories' => $categories,
    'user' => $user,
]);
}
```

Slika 13. Dio koda za dohvaćanje korisnika i košarice

Izvor: autor

3.3. Administrativni dio trgovine

Kod mrežnih trgovina i dinamičnih mrežnih stranica administracija je veoma bitna. Ovaj dio upravlja cjelokupnim sadržajem koji se prikazuje i sadržajem u bazi podataka. Administratori imaju mogućnost upravljati kategorijama, podkategorijama, proizvodima, njihovim cijenama i narudžbama. Aplikacija kod prijave provjerava ulogu (eng. *Role*) korisnika. Korisnici koji su u bazi definirani kao administratori, tj. imaju vrijednost polja „ROLE_ADMIN“ mogu pristupiti administrativnom dijelu aplikacije.

Administrativni dio sadrži logiku za upravljanje kategorijama, podkategorijama, kontakt-podacima, narudžbama, proizvodima, korisnicima, sadržajem stranice „O nama“.

Za izradu administrativnog dijela koristi se besplatan paket pod nazivom *EasyAdmin* koji omogućuje generiranje administrativnog sučelja za upravljanje sadržajem. Nakon instaliranja paketa potrebno ga je konfigurirati prema potrebama aplikacije.

Paket *EasyAdmin* sadrži funkcije koje se implementiraju u kontrolere za lakšu konfiguraciju administrativnog sučelja. Neke od funkcija su *configureFields*, *configureCrud* i *configureFilters*. Funkcija *configureFields* koristi se za konfiguraciju sadržaja koji se prikazuje i koji je moguće uređivati, definira polja obrasca koja administrator može vidjeti i podatke kojima može upravljati. Funkcija *configureCrud* omogućuje konfiguraciju *CRUD* akcija na svakoj od stranica. Funkcija *configure Filters* omogućuje dodavanje i konfiguriranje filtera na stranici za lakše snalaženje u velikoj količini podataka. Slika 14. prikazuje izgled *ConfigureFields* funkcije u *CRUD* kontroleru za proizvode.

```
public function configureFields(string $pageName): iterable
{
    $podkategorije = $this->em->getRepository(Subcategory::class);
    yield BooleanField::new('dostupan')->renderAsSwitch(true);
    yield IntegerField::new('popust')->setLabel('Popust (cijeli brojevi od 1 do 100)');
    yield TextField::new('naziv')->setLabel('Naziv proizvoda');
    yield TextareaField::new('opis')->setLabel('Opis proizvoda')->renderAsHtml();
    yield MoneyField::new('cijena')->setCurrency('EUR')->setLabel('Cijena proizvoda')
        ->setCustomOption('storedAsCents', false);
    yield ImageField::new('slika')->setBasePath('uploads/images/')->setUploadDir('public/uploads/images/')
        ->setFormTypeOption('allow_delete', false)
        ->setRequired($pageName !== 'edit');
    yield AssociationField::new('kategorija')->setLabel('kategorija')
        ->setFormTypeOption('choice_label', 'naziv');
    yield AssociationField::new('podkategorija')->setFormTypeOption('choice_label', 'naziv')
        ->setFormTypeOption('required', false)->setLabel('Podkategorija');
```

Slika 14. ConfigureField funkcija, Izvor: autor

Paket *EasyAdmin* omogućuje rad s raznim vrstama podataka. Za tekstualne podatke kao što je naziv proizvoda koristi se polje *TextField* koje omogućuje unos i validaciju teksta. Za rad sa slikama koristi se polje *ImageField*, potrebno je samo odrediti direktorij u koji se spremaju slike. Za unos cijene proizvoda koristi se polje *MoneyField* kojem je potrebno odrediti valutu. Postoji još mnogo različitih vrsta polja koja je moguće koristiti za konfiguraciju forme, kao što su: *TextareaField* za unos tekstova dužih od 255 znakova i *IntegerField* za rad s cijelim brojevima.

3.3.1. Administrativno sučelje

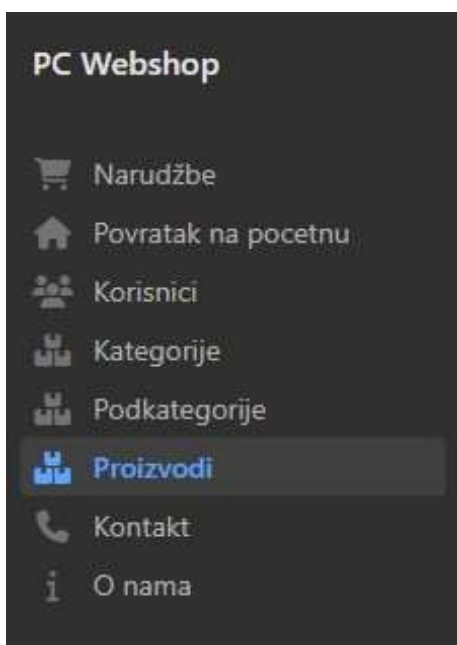
Administratori nakon uspješne prijave mogu pristupiti administrativnom sučelju klikom na gumb „Admin“ na navigacijskoj traci. Navigacijska traka za administratore razlikuje se od trake za korisnike. Izgled trake za administratore prikazan je na slici 15.



Slika 15. Izgled navigacijske trake

Izvor: autor

Klikom na gumb „Admin“ otvara se administrativno sučelje gdje je moguće upravljati sadržajem mrežne stranice. Administratorima navigaciju po sučelju olakšava izbornik koji sadrži: narudžbe, korisnike, kategorije, podkategorije, proizvode, kontakt, stranicu „O nama“. Izbornik je prikazan na slici 16.



Slika 16. Izbornik za administratore

Izvor: autor

Klikom na „Narudžbe“ u izborniku, otvara se popis svih narudžbi koje su korisnici napravili. Administrator može mijenjati status svake narudžbe i ispisati račune za narudžbe. Narudžbe koje se prikazuju sortirane su po datumu tako da se prvo prikazuju one najnovije. Slika 17. prikazuje izgled stranice za narudžbe.

Broj narudžbe	Korisnik	Status	Cijena	Adresa dostave	Datum	Račun(PDF)	Promjeni status
101	da@ne.hr	Zaprimljena	622.10 €	Koprivnicka ulica 10a, 48000, koprivnica	09/09/2023	Preuzmi	Status ▾
100	da@ne.hr	Zaprimljena	560.00 €	zagrebacka 34a, 10000, zagreb	09/09/2023	Preuzmi	Status ▾
99	da@ne.hr	Zaprimljena	62.10 €	zagrebacka 10a, 10000, zagreb	09/09/2023	Preuzmi	Poslana Zaprimljena Poništena Zaključana
98	da@ne.hr	Zaprimljena	1831.32 €	Zagrebacka ulica 34, 10000, Zagreb	09/09/2023	Preuzmi	
97	da@ne.hr	Zaprimljena	433.30 €	varazdinska ulica 25, 12345, varazdin	09/09/2023	Preuzmi	

Slika 17. Izgled stranice za narudžbe

Izvor: autor

Boja polja „Status“ u tablici mijenja se ovisno o statusu narudžbe, kod tek zaprimljenih narudžbi polje „Status“ je zelene boje, kod poništenih crvene, kod poslanih žute, a kod zaključenih plave boje. Administratori mogu promijeniti status narudžbe klikom na padajući izbornik „Status“ i odabirom željenog statusa. Preuzimanje računa za narudžbu u PDF formatu omogućeno je klikom na gumb „Preuzmi“.

Administratori mogu pregledavati registrirane korisnike klikom na gumb „Korisnici“ u izborniku (slika 16.), nakon čega im se otvara popis svih korisnika. Korisnike je moguće samo izbrisati iz baze podataka. Slika 18. prikazuje popis svih korisnika.

Svi korisnici		
<input type="checkbox"/>	ID ↑	Email ⇅
<input type="checkbox"/>	4	da@ne.hr
<input type="checkbox"/>	6	admin@boris.hr

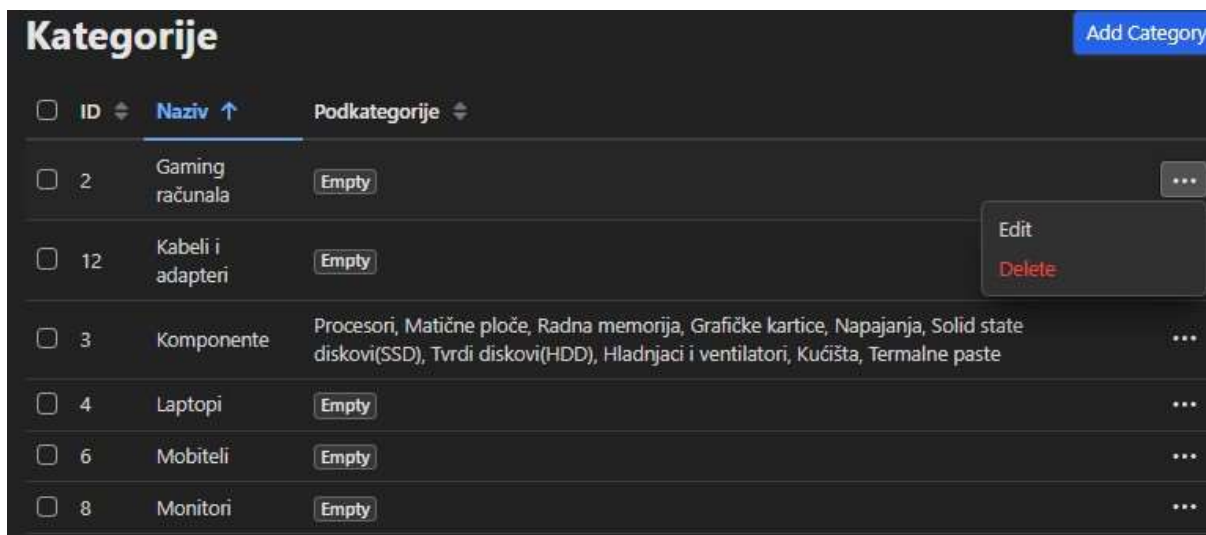
2 results

< Previous 1 Next >

Slika 18. Popis korisnika

Izvor: autor

Administratorima je dodavanje novih kategorija ili uređivanje već postojećih omogućeno klikom na gumb „Kategorije“ u izborniku (slika 16.). Klikom na gumb „Kategorije“ otvara se stranica s popisom svih kategorija koje su pohranjene u bazi podataka. Slika 19. prikazuje stranicu za upravljanje kategorijama.



Slika 19. Stranica za upravljanje kategorijama

Izvor: autor

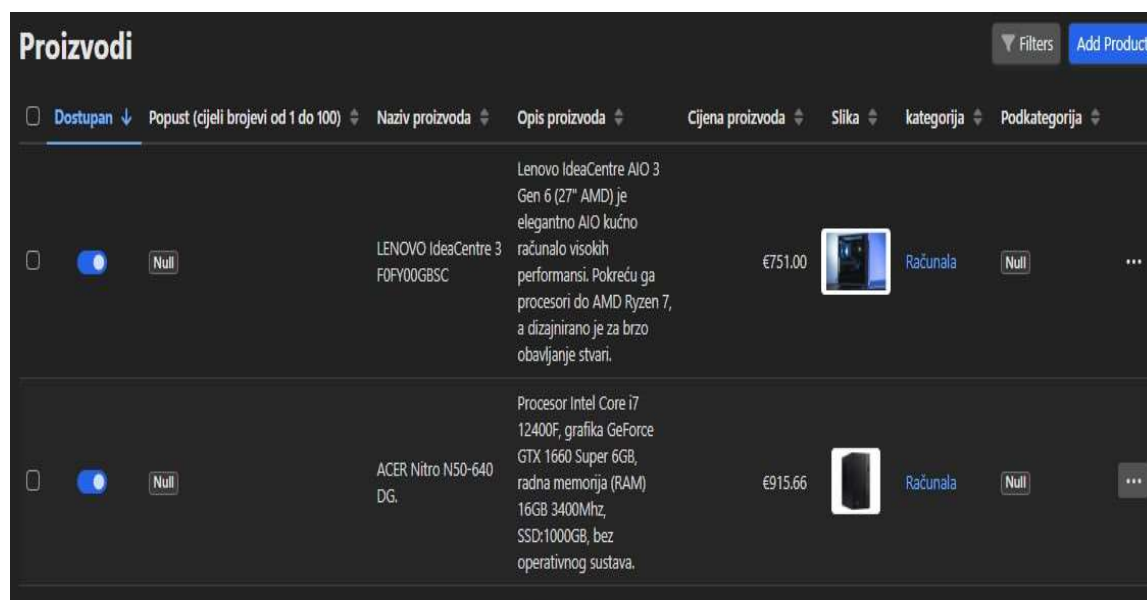
Stranica za upravljanje podkategorijama otvara se klikom na gumb „Podkategorije“ u izborniku (slika 16.). Administratori mogu dodavati nove podkategorije i povezivati ih s pripadajućom kategorijom u bazi. Slika 20. prikazuje izgled stranice za upravljanje podkategorijama.



Slika 20. Stranica za upravljanje podkategorijama

Izvor: autor

Jedna od glavnih zadaća administratora mrežne trgovine je upravljanje proizvodima u skladištu. Administratori mogu upravljati proizvodima klikom na gumb „Proizvodi“ u izborniku (slika 16.). Klikom na taj gumb otvara se stranica s popisom svih proizvoda koji se trenutno nalaze u bazi podataka.

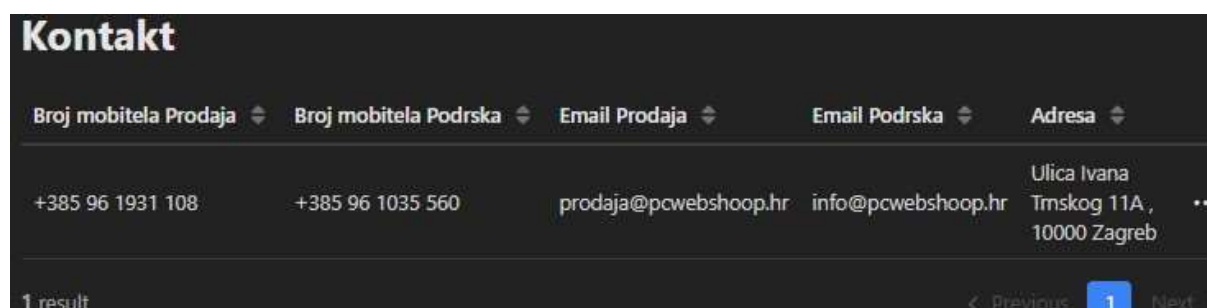


Slika 21. Stranica za upravljanje proizvodima

Izvor: autor

Administratori na ovoj stranici mogu uređivati proizvode i dodavati nove proizvode u bazu. Proizvodi se neće prikazivati na stranicama mrežne trgovine ako administrator deselektira polje „Dostupan“.

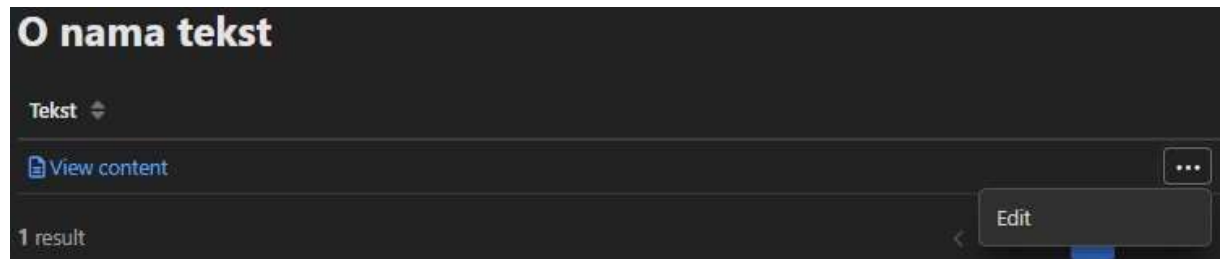
Kad administrator klikne na gumb „Kontakt“ u izborniku (slika 16.), može uređivati podatke za kontakt koji se prikazuju na stranici za kontakt mrežne trgovine.



Slika 22. Stranica za upravljanje podacima za kontakt

Izvor: autor

Klikom na gumb „O nama“ u izborniku (slika 16.) omogućeno je uređivanje teksta koji se prikazuje na stranici „O nama“. Slika 23.



Slika 23. Stranica za upravljanje tekстом na stranici „O nama“

Izvor: autor

3.4. Korisnički dio trgovine

Kada posjete mrežnu trgovinu, korisnici prvo mogu vidjeti početnu stranicu. Na početnoj stranici prikazuju se proizvodi koji su na popustu i ostali proizvodi dostupni u skladištu trgovine. Proizvodi na popustu prikazuju se na vrhu stranice, odvojeno od ostalih proizvoda, slika 24.



Slika 24. Lista proizvoda na popustu

Izvor: autor

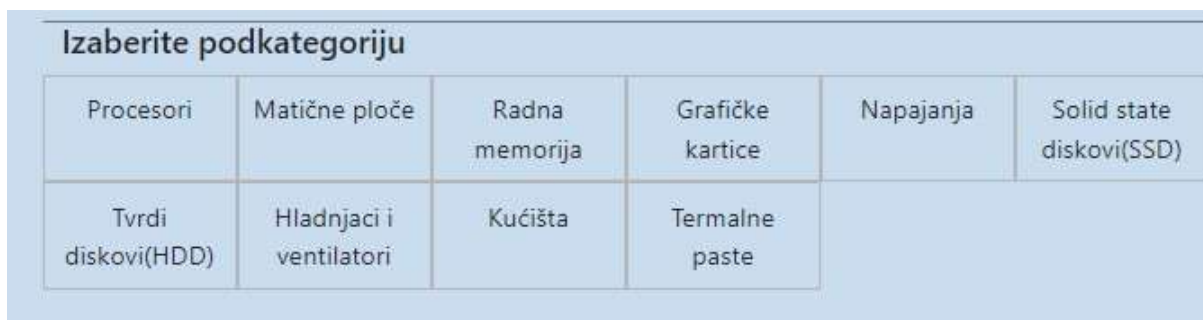
Navigiranje kategorijama i podkategorijama proizvoda korisnicima olakšava padajući izbornik koji se nalazi u navigacijskoj traci, slika 25.



Slika 25. Padajući zbornik

Izvor: autor

Klikom na neku od kategorija u padajućem izborniku otvara se stranica gdje se prikazuju proizvodi samo iz odabrane kategorije. Ako kategorija ima podkategorije, one se prikazuju na vrhu stranice, slika 26.



Slika 26. Popis podkategorija

Izvor: autor

Klikom na neku od podkategorija, otvara se stranica gdje su prikazani proizvodi iz te podkategorije. Korisnici mogu dodavati proizvode u košaricu klikom na gumb „Dodaj u košaricu“. Prije dodavanja u košaricu, moguće je promijeniti količinu proizvoda, a nakon što je proizvod dodan u košaricu, aplikacija ispisuje obavijest da je proizvod dodan u košaricu.

Proizvodi u košarici prikazani su u tablici, svim proizvodima u košarici korisnik može promijeniti količinu ili ih izbrisati iz košarice. Izgled košarice prikazan je na slici 27.



Slika 27. Izgled košarice, Izvor: autor

Klikom na gumb „Nastavi s narudžbom“ omogućeno je naručivanje proizvoda. Stranica za naručivanje sadrži obrazac za unos adrese, grada i poštanskog broja. Svi podaci koje korisnik unosi provjeravaju se prije postavljanja narudžbe. Slika 28. prikazuje izgled obrasca za naručivanje.

Slika 28. Obrazac za naručivanje

Izvor: autor

Nakon ispravnog unosa postavlja se narudžba sa statusom „zaprimljena“, aplikacija šalje *e-mail* s potvrdom narudžbe na korisnikovu e-adresu i ispisuje obavijest da je narudžba zaprimljena. Korisnik može svoje narudžbe pregledati klikom na „Narudžbe“ u navigacijskoj traci. Stranica za pregled narudžbi omogućuje korisnicima preuzimanje potvrde narudžbe ili računa ako narudžba ima status „poslana“.

Broj narudžbe	Cijena	Status	Adresa za dostavu	Datum naručivanja	Pregled/Račun(PDF)
101	622.10 €	Zaprimljena	Koprivnicka ulica 10a, 48000, koprivnica	09/09/2023	Pregled
100	560.00 €	Zaprimljena	zagrebacka 34a, 10000, zagreb	09/09/2023	Pregled

Slika 29. Stranica za pregled narudžbi

Izvor: autor

4. ZAKLJUČAK

U ovom završnom radu napravljen je prototip jednostavne mrežne trgovine koja omogućuje brzo i jednostavno naručivanje informatičke opreme, mobitela, računala, laptopa i drugih proizvoda. Korištenjem programskog okvira *Symfony*, alata i paketa, napravljena je aplikacija jednostavnog i responzivnog dizajna koja odgovara svim profilima korisnika, za korištenje aplikacije nije potrebno nikakvo predznanje rada na računalu. *Symfony* se pokazao kao jako dobar programski okvir za izradu mrežnih trgovina zbog jednostavnosti korištenja i mnogobrojnih alata i paketa koji znatno olakšavaju izradu mrežnih trgovina. Naravno, uvijek postoji prostor za napredak pa bi se tako i ova aplikacija mogla poboljšati raznim funkcionalnostima, kao što su ostavljanje recenzije na proizvodima i mogućnost korištenja kodova za popuste. Izrada mrežne trgovine nije jednostavna, iziskuje mnogo truda i vremena. Na početku izrade javili su se brojni problemi oko potvrde e-adrese kod registracije korisnika, slanja računa korisnicima nakon naručivanja proizvoda, no u konačnici se može zaključiti da je ova aplikacija zadovoljila svoju svrhu.

5. POPIS LITERATURE

Internet:

[1] Symfony documentation, dostupno na: <https://symfony.com/doc/current/index.html> (13.8.2023.)

[2] 431BUvod u PHP programiranje (mev.hr), dostupno na: https://www.mev.hr/wp-content/uploads/2019/01/Uvod_u_PHP_programiranje.pdf (13.8.2023.)

[3] W3Schools Online Web Tutorials, dostupno na <https://www.w3schools.com/> (13.8.2023.)

[4] Twig - The flexible, fast, and secure template engine for PHP, dostupno na: <https://twig.symfony.com/doc/> (13.8.2023.)

[5] Visual Studio Code - Code Editing. Redefined, dostupno na: <https://code.visualstudio.com/> (13.8.2023.)

[6] Getting started with Doctrine, dostupno na: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.16/tutorials/getting-started.html> (17.8.2023.)

6. PRILOZI

Slika 1. Programski okvir <i>Symfony</i>	4
Slika 2. Primjer PHP skripte.....	5
Slika 3. Primjer <i>Twig</i> -koda.....	7
Slika 4. Sučelje <i>XAMPP</i> kontrolne ploče.....	8
Slika 5. Sučelje uređivača <i>Visual Studio Code</i>	9
Slika 6. Sadržaj direktorija aplikacije	12
Slika 7. Shema baze podataka	13
Slika 8. Konfiguracija za spajanje baze podataka	15
Slika 9. Dio koda u klasi proizvod	16
Slika 10. Tablica "proizvod"	17
Slika 11. Tablice u bazi podataka	17
Slika 12. Primjer kreiranja kontrolera	18
Slika 13. Dio koda za dohvaćanje korisnika i košarice.....	19
Slika 14. Funkcija <i>configureField</i>	20
Slika 15. Izgled navigacijske trake.....	22
Slika 16. Izbornik za administratore	22
Slika 17. Izgled stranice za narudžbe	23
Slika 18. Popis korisnika.....	23
Slika 19. Stranica za upravljanje kategorijama	24
Slika 20. Stranica za upravljanje podkategorijama	24
Slika 21. Stranica za upravljanje proizvodima.....	25
Slika 22. Stranica za upravljanje podacima za kontakt.....	25
Slika 23. Stranica za upravljanje teksta na stranici „O nama“	26
Slika 24. Lista proizvoda na popustu	27
Slika 25. Padajući izbornik.....	28
Slika 26. Popis podkategorija.....	29
Slika 27. Izgled košarice	29
Slika 28. Obrazac za naručivanje	30
Slika 29. Stranica za pregled narudžbi	30