Универзитет у Београду

Електротехнички факултет



**Реализација интернет апликације за преглед асортимана продавница коришћењем веб индексера**

дипломски рад основних академских студија

|  |  |
| --- | --- |
| Ментор: | Кандидат: |
| доц. др Дражен Драшковић | Матија Пешић 0428/2017 |

Београд, Септембар 2021.

Садржај

[1. Увод 4](#_Toc82228231)

[2. Значај и развој веб система 5](#_Toc82228232)

[3. Захтеви за реализацијом система 6](#_Toc82228233)

[3. 1. Нефункционални захтеви система 6](#_Toc82228234)

[3. 1. 1. Ауторизација и аутентификација корисника 6](#_Toc82228235)

[3. 2. Нерегистровани корисници 10](#_Toc82228236)

[3. 2. 1. Регистрација корисника 10](#_Toc82228237)

[3. 2. 2. Пријављивање на систем 10](#_Toc82228238)

[3. 2. 3. Преглед садржаја апликације 10](#_Toc82228239)

[3. 3. Регистровани корисници 10](#_Toc82228240)

[3. 3. 1. Купац 10](#_Toc82228241)

[3. 3. 1. 1. Преглед садржаја корпе 10](#_Toc82228242)

[3. 3. 1. 2. Куповина садржаја корпе 11](#_Toc82228243)

[3. 3. 1. 3. Измена личних информација 11](#_Toc82228244)

[3. 3. 2. Администратор 11](#_Toc82228245)

[3. 3. 2. 1. Преглед активности „crawler“-а 11](#_Toc82228246)

[3. 3. 2. 2. Уклањање свих артикала из одређене продавнице 11](#_Toc82228247)

[4. Преглед коришћених технологија 12](#_Toc82228248)

[4. 1. Технологије коришћене на серверској страни 12](#_Toc82228249)

[4. 1. 1. Програмски језик С# 12](#_Toc82228250)

[4. 1. 2. ASP.NET 5 12](#_Toc82228251)

[4. 1. 3. Entity Framework Core 5.0 13](#_Toc82228252)

[4. 1. 4. *Language Integrated Query (LINQ)* 13](#_Toc82228253)

[4. 1. 5. SQLite 14](#_Toc82228254)

[4. 1. 6. AutoMapper 14](#_Toc82228255)

[4. 1. 7. Html Agility Pack 15](#_Toc82228256)

[4. 1. 8. Puppeteer 15](#_Toc82228257)

[4. 2. Технологије коришћене на клијентској страни 15](#_Toc82228258)

[4. 2. 1. Hypertext Markup Language (HTML) 15](#_Toc82228259)

[4. 2. 2. Cascading Style Sheets (CSS) 16](#_Toc82228260)

[4. 2. 3. JavaScript 16](#_Toc82228261)

[4. 2. 4. TypeScript 16](#_Toc82228262)

[4. 2. 5. Angular 12 16](#_Toc82228263)

[4. 2. 6. ngx – bootstrap 17](#_Toc82228264)

[4. 2. 7. JavaScript Object Notation (ЈЅОN) 17](#_Toc82228265)

[5. Опис рада система 18](#_Toc82228266)

[5. 1. Нерегистровани корисници 18](#_Toc82228267)

[5. 1. 1. Регистрација корисника 18](#_Toc82228268)

[5. 1. 2. Пријављивање на систем 18](#_Toc82228269)

[5. 1. 3. Преглед садржаја апликације 18](#_Toc82228270)

[5. 2. Регистровани корисници 18](#_Toc82228271)

[5. 2. 1. Купац 18](#_Toc82228272)

[5. 2. 1. 1. Преглед историје поруџбина 19](#_Toc82228273)

[5. 2. 1. 2. Преглед садржаја корпе 19](#_Toc82228274)

[5. 2. 1. 3. Куповина садржаја корпе 19](#_Toc82228275)

[5. 2. 1. 4. Измена личних информација 19](#_Toc82228276)

[5. 2. 2. Администратор 19](#_Toc82228277)

[5. 2. 2. 1. Преглед активности „crawler“-а 19](#_Toc82228278)

[5. 2. 2. 2. Уклањање свих артикала из одређене продавнице 19](#_Toc82228279)

[6. Реализација система 21](#_Toc82228280)

[7. Закључак 22](#_Toc82228281)

[8. Литература 23](#_Toc82228282)

1. Увод

Доста људи воли да слободно време проводи у шопингу и обилазећи продавнице по тржним центрима. Пандемија SARS-CoV-2 вируса је унела многе промене у животе људи, па тако и куповину одеће. Обилазак тржних центара више није био могућ, па је људима једина опција била разгледање и куповина проивода преко интернета. Проблем таквог обиласка продавница је лоша ефикасност, тј. свака продавница се мора посебно разгледати и појединачно се морају подешавати критеријуми претраге.

Циљ овог рада је креирање апликације која представља обједињени веб сајт за неколико продавница, где би били приказани артикли који се довлаче са званичних веб сајтова продавница. За имплементацију апликације изабране технологије су ASP.NET 5 за серверску и Angular за клијентску страну. За ефикаснији рад са базом података коришћен је Entity Framework. На серверској страни је апликација организована као веб сервис, док је на клијентској страни орагнизована као једностранична апликација.

Странице над којима би радили „crawler“-и би биле странице продавница гардеробе, које ја најчешће посећујем.

|  |  |
| --- | --- |
| **Назив продавнице** | **Почетни линк од ког би се вршило кроловање** |
| Pull&Bear | [https://www.pullandbear.com/rs/](https://www.pullandbear.com/rs/mu%C5%A1karci-c1030204564.html) |
| Zara | <https://www.zara.com/rs/> |
| Terranova | <https://www.terranovastyle.com/rs_sr/> |

*Табела 1 Приказ продавница и њихових линкова*

У другом поглављу описан је значај и развој веб система у данашњем друштву, као и основни механизми прикупљања података са интернета. У тећем поглављу биће наведени захтеви за реализацијом система. Четврто поглавље садржи детаљније описане технологије и библиотеке коришћене за развој интернет апликације. Детаљан преглед корисничког интерфејса и корисничко упутство за сваку од основних функционалности апликације налази се у петом поглављу. У шестом поглављу су приказани најинтересантији делови имплементације и техничке карактеристике решења. Седмо, и последње поглавље је заључак у оквиру кога је описан значај имплементираног решења и могућа даља унапређења.

1. Значај и развој веб система
2. Захтеви за реализацијом система

У следећем поглављу су описане функционалности које поседује реализовани систем. У обзир су узети сви сценарији коришћења апликације.

Систему могу да приступе нерегистровани корисници и регистртовани корисници. Типови регистрованих корисника у систему су Купац и Администратор. Узимајући у обзир доступне функционалности система, није било потребно додатно раслојавање типова корисника. Сваки новорегистровани корисник једино може постати купац, док је администратор унапред регистрован, од почетка постојања система.

Цео систем је изграђен тако да довлачи податке са постојећих веб сајтова и да их на одређени начин приказује корисницима.

* 1. Нефункционални захтеви система

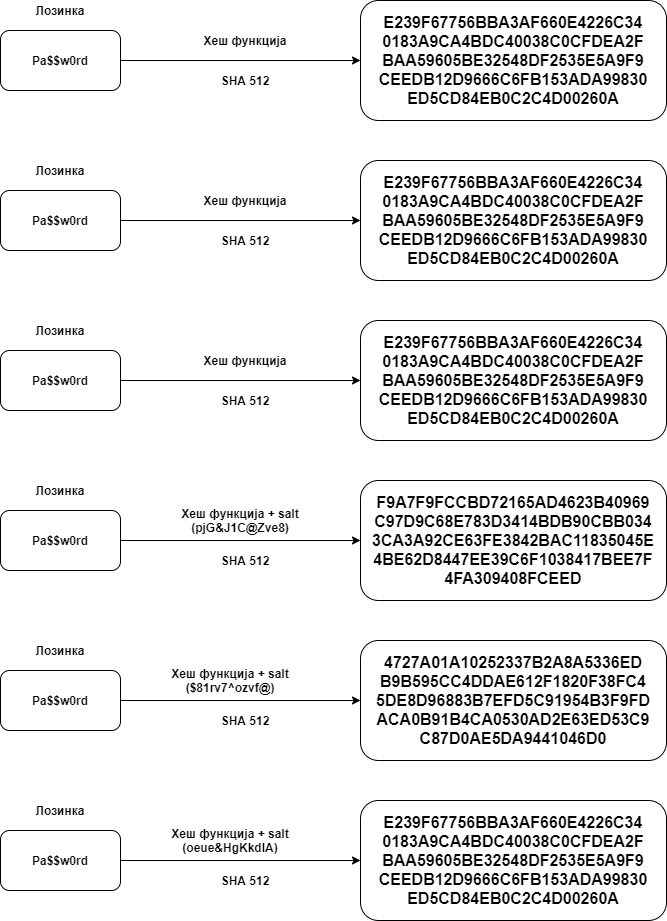
Основне функционалности сваког веб система су ауторизација и аутентификација корисника, тј. кориснику не сме бити могућ приступ страницама и ресурсима за које нема привилегије.

При покушају да се приступи страници, ка којој корисник нема право приступа, биће преусмерен на почетну страну апликације уз поруку „You shall not pass!“, која је видљива у доњем десном углу екрана, у црвеном правоугаонику.

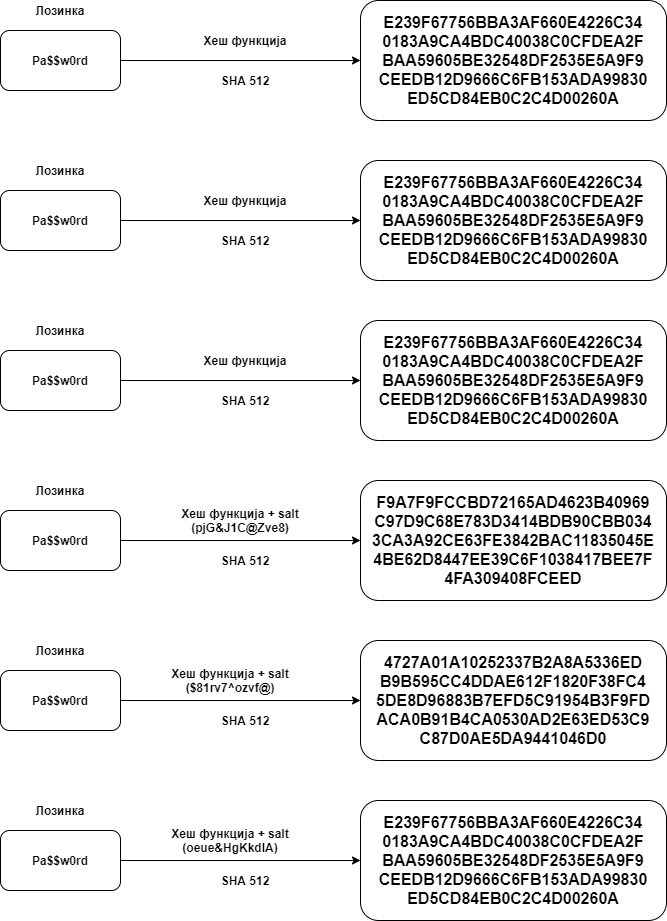
* + 1. Ауторизација и аутентификација корисника

Једна од најбитнијих ствари при креирању апликација, које у свом раду користе и чувају неке поверљиве информације, јесте одлука о томе како ће се те информације чувати у бази података. Неприхватљиво је чувати лозинке и остале поверљиве информације њиховом основном облику. Најчешће се примењују различите методе заштите података, као што су енкрипција и хеширање.

У овом раду је за чување лозинки примењено само хеширање, коришћењем salt-a. Хеширање представља метод који улазни податак, у овом случају лозинку, пропушта кроз хеш функцију и тако добија јединствену хеш вредност. Кључна карактеристика хеш функција је што су иреверзибилне, тј. из хеширане вредности се не може добити оригинална вредност. При регистрацији корисника се израчунава хеш вредност и она се чува у бази података. При логовању корисника се од унете лозинке поново израчунава хеш вредност и она се упоређује са оном сачуваном у бази података. Ако и само ако су вредности исте, закључује се да је корисник унео исправну лозинку и биће му омогућен приступ систему. У супротном му се исписује одговарајућа порука.



Слика # Приказ примене хеш функције над истим лозинкама

Мана хеширања је што једна хеш функција за исту улазну вредност увек даје исту излазну вредност, што може бити проблем ако више корисника има исту шифру. Овај проблем се решава увођењем насумичне вредности, зване salt, која утиче на излаз хеш функције. Ако се користи иста salt вредност за исти улазни податак, увек ће се добити иста излазна вредност хеш функције. Због тога се у бази података чува и salt вредност која је добијена при регистрацији корисника и која је јединствена само за њега.

Слика # Приказ примене хеш функције уз коришћење salt вредности над истим лозинкама

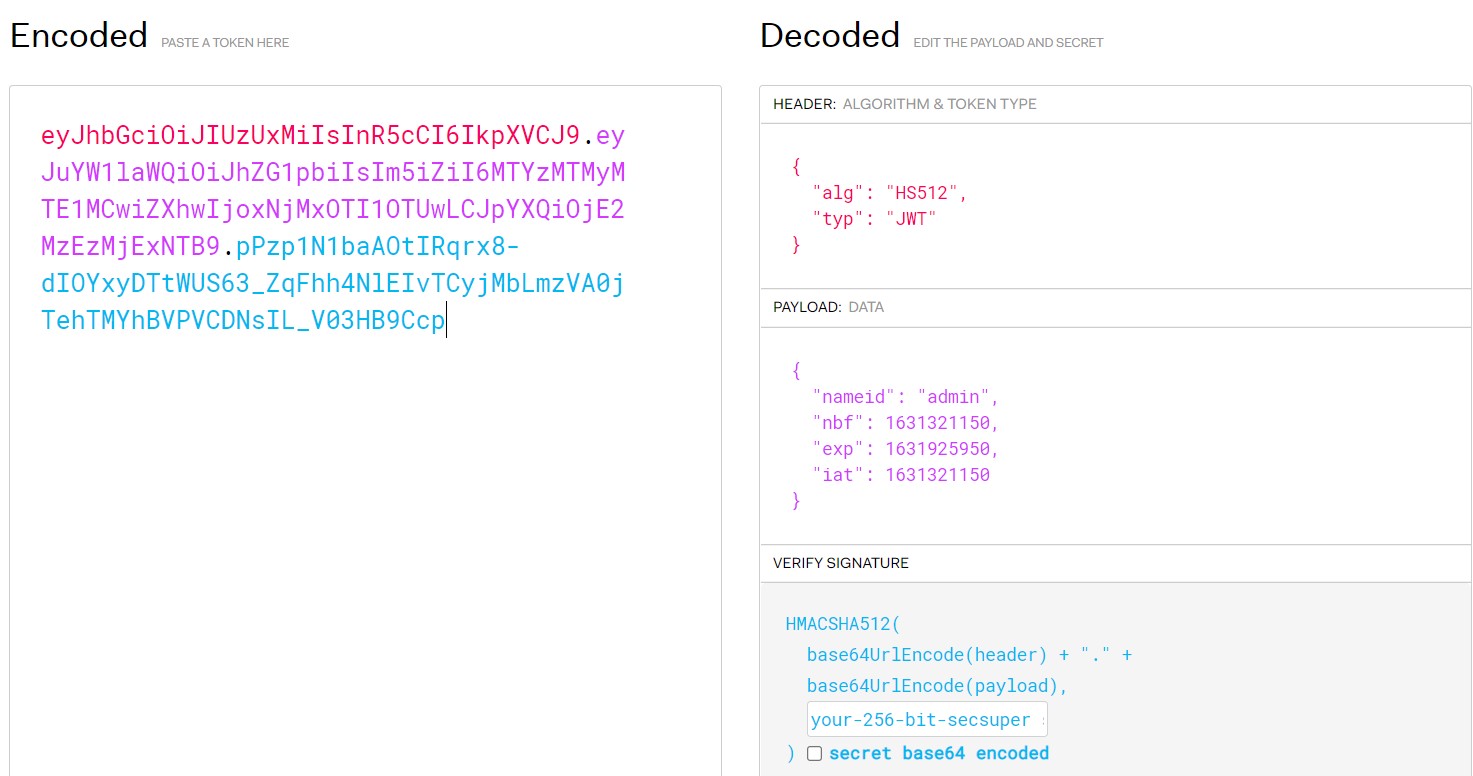
Слика # Приказ примене хеш функције уз коришћење salt вредности над истим лозинкама

Захваљујући коришћењу salt вредности не може се одредити да ли више корисника има исту шифу, нити да ли исту шифру користи на више сајтова.

Пошто хеширање може одузети доста времена, није практично да се при сваком захтеву израчунава хеш вредност лозинке корисника. Због тога се за аутентификацију корисника, при изради овог рада, користи *JWT (JSON Web Token)* механизам аутентификације. *JWT* представља сигурно преношење информација између клијента и сервера путем *ЈЅОN* објеката.

Пример једног токена дат је на слици Слика #. На левој половини слике се види енкриптована вредност *JWT*-а. На десној половини јасно виде оригиналне вредности три дела *JWT*-а:

* **Заглавље (Header) –** сатоји се из два обавезна поља: Ознаке типа токена (*JWT*) и Алгоритма који је коришћен за потписивање (SHA512).
* **Payload** – Садржи информације о кориснику апликације, као и о временима од важности за токен. Времена од важности представљају време када је издат токен (iat – Issued At), почетно време важења токена (nbf – Not Before) и време истека рока трајања токена (exp – Expiration Time)
* **Потпис (Signature) –** над заглављем, *рayload*-ом и тајном вредношћу (*Secret*), која је позната само серверској страни, примењује се алгоритам назначен у заглављу.



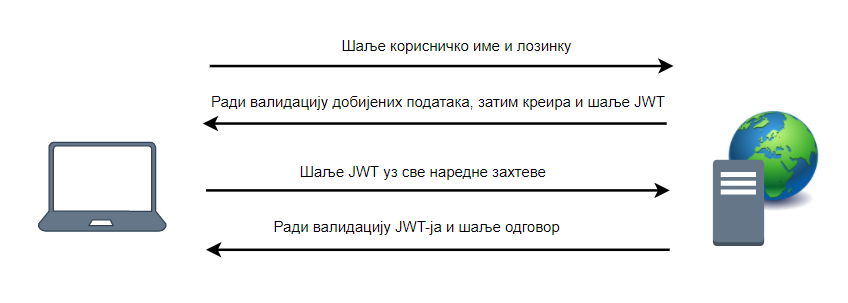
Слика # Приказ формираног JWT-a

Клијент при сваком захтеву у хедеру захтев прослеђује токен, а на серверској страни се на основу *ѕecret* вредности врши валидација токена. Тајна вредност је заслужна за обезбеђивање велике сигурности да иако неки корисник покуша да подметне токен неће проћи валидацију са серверске стране, јер му *ѕecret* вредност није доступна.

Процес аутентификације помоћу токена се састоји из четири корака, који су приказани на слици Слика #:

1. При логовању на систем, корисник серверу шаље своје корисничко име и лозинку. Сервер на основу корисничког имена дохвата salt и хеширану вредност из базе података и покушава да установи исправност унете лозинке.
2. Уколико дође до подударања хеш вредности сачуване у бази података и новоизрачунате на основу лозинке и salt-а, сервер формриа нови токен и шаље га кориснику.
3. Корисник токен чува локално и шаље га приликом свих наредних захтева ка серверу, који захтевају аутентикацију.
4. Сервер сада само врши валидацију токена помоћу тајне вредности и шаље одговор назад ка клијенту.

Слика # Шематски приказ комуникације клијента и сервера приликом коришћења JWT-a за аутентификацију



* 1. Нерегистровани корисници

Опис свих функционалности које су доступне нерегистрованим корисницима система се налази у овој секцији. Нерегистровани корисници имају могућност регистрације као купац и пријављивања на систем, прегледа садржаја апликације.

* + 1. Регистрација корисника

При регистрацији корисници се могу регистровати само као купци и тада попуњавају форму где уносе име, презиме, мејл, јединсвено корисничко име, лозинку, коју морају два пута унети, из безбедносних разлога, датум рођења и пол.

* + 1. Пријављивање на систем

При пријављивању на систем, корисник уноси корисничко име или мејл и лозинку. У случају успешног пријављивања корисник се преусмерава на одговарајућу страницу. У случају неуспешног пријављивања, кориснику се исписује порука о грешци.

* + 1. Преглед садржаја апликације

Преглед садржаја представља приказивање свих производа, који се налазе у бази, кориснику. Коришћењем различитих филтера корисник може да промени број производа који му се приказују.

* 1. Регистровани корисници

Корисницима који су регистровани, након пријављивања нуди се број различитих функционалности. Функционалности које ће им бити на располагању зависе од типа корисника, тј. да ли су купац или администратор.

* + 1. Купац

Купац, од додатних функционалности има могућност убацивање производа у корпу и избацивање садржаја из корпе, преглед садржаја корпе као и куповина садржаја корпе. За купца се трајно памти садржај корпе (тј. док је он сам не испразни). Купац може и да врши промену својих информација.

* + - 1. Преглед садржаја корпе

При прегледу садржаја корпе, кориснинку се приказује табеларни преглед садржаја корпе, као и опције да промени количину једног производа у корпи или да га избаци из корпе.

* + - 1. Куповина садржаја корпе

Када корисник пожели да купи садржај корпе, потребно је да попуни форму са подацима потребним за испоруку. Куповина садржаја је заправо само симулација куповине, тј. производи се не наручују из стварних продавница, јер то није циљ дипломског рада.

* + - 1. Измена личних информација

Корисник има могућност да промене свих својих информација, осим корисничког имена.

* + 1. Администратор

У систему постоји јединствен администратор. Он може да врши све што и купци, а може ид да изврши преглед активних „crawler“-а, изврши измену њихове активности или обрише све артикле неке продавнице.

* + - 1. Преглед активности „crawler“-а

Када врши преглед активности „crawler“-а, администратору се приказује табеларни преглед, а као назив „crawler“-а приказује назив продавнице над којом „crawler“ ради. Поред сваког назива се налази и поље за потврду (checkbox), које ако је означено, „crawler“ ради, тј. дохвата податке са одговарајућих страница, а ако није означено, „crawler“-а је блокиран, тј. не дохвата податке. Промена статуса „crawler“-а се врши одмах при промени означености поља.

* + - 1. Уклањање свих артикала из одређене продавнице

При прегледу свих продавница, поред поља за потврду (checkbox), налази се дугме чијим кликом админ из базе уклања све производе који су добијени преко сајта одговарајуће продавнице.

1. Преглед коришћених технологија

У овом поглављу биће описани сви програмски језици, технологије и библиотеке које су коришћене при изради система. Опис је подељен у две категорије – технологије коришћене на серверској и клијентској страни, ради прегледности.

* 1. Технологије коришћене на серверској страни

За развој серверског дела апликације је коришћен *.NET 5* који кроз *Nuget Package* *Manager* обезбеђује могућност лаке инсталације великог броја библиотека. .*NET 5* је најновија платформа из .*NET* породице и наследник *.NET Core 3.0* платформе.

* + 1. Програмски језик С#

Један је од млађих програмских језика и настао је 2000. године као саставни део Мајкрософтовог развојног окружења .*NET Framework* *1.0*. Језик је опште примене и намењен је изради апликација за .*NET Framework* платформу и један је од водећих језика у интернет програмирању [1].

Развој аликација у развојној платформи .*NET 5* захтева језик *С#*, па је из овог разлога и коришћен у имплементацији серверског дела апликације.

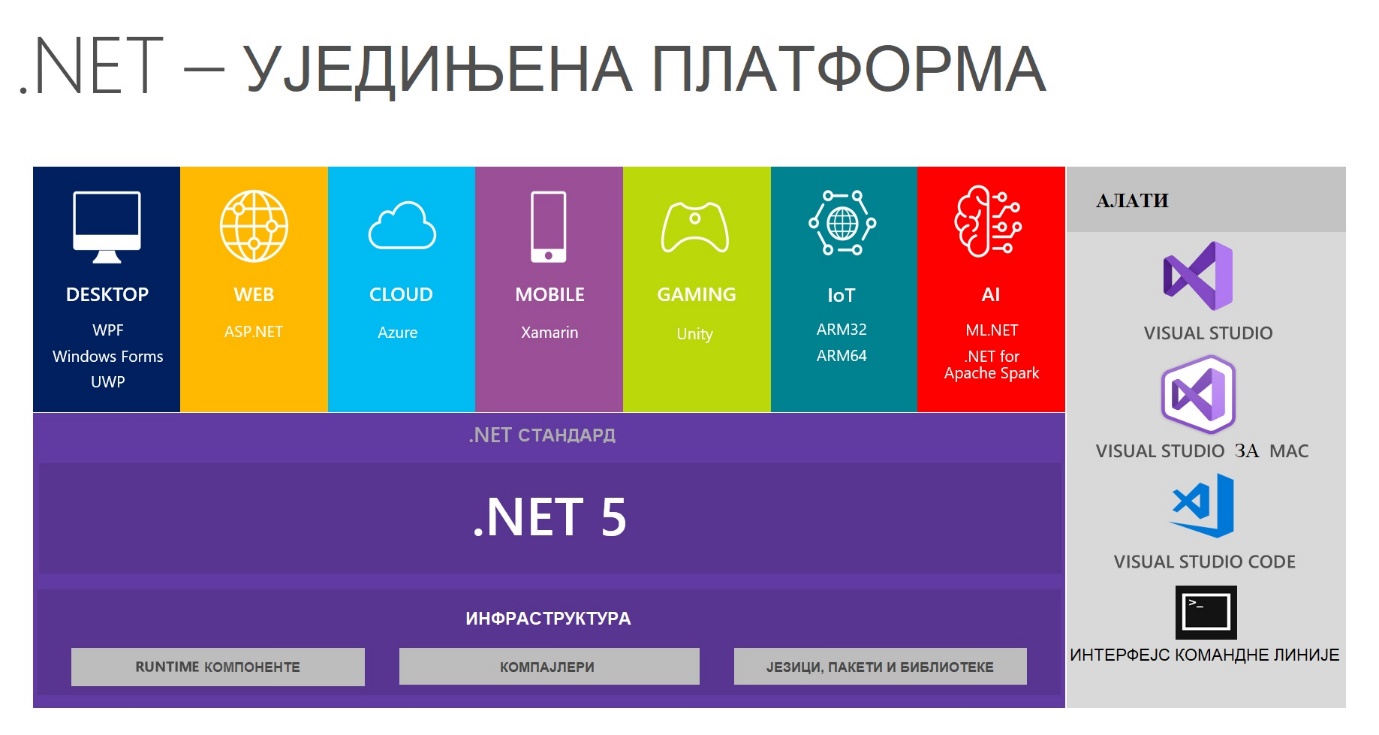
* + 1. ASP.NET 5

ASP.NET 5 (у даљем делу текста .NET 5) је јавно дсотупна и бесплатна платформа развијена од стране Мајкрософта. Објављена је у Новембру 2020. године, наследник је .NET Core 3.0 платформе, а уједно представља и обједињавање .NET Framework и .NET Core платформи у јединствену платформу. Назван је .NET 5 уместо .NET Core 4.0 из два разлога[2]:

* Прескочена је верзија броја 4.x како би се избегла конфузија са .NET Framework 4.x верзијама.
* Избачен је „Core“ из имена како би се нагласило да је ово сада главна имплементација која ће се развијати. .NET 5.0 подржава више типова апликација и платформи од .NET Framework или .NET Core.

Неке од функционалности овог радног оквира су:

* Могућност развијања апликација за Windows, Linux, macOS, iOS, Android, tvOS, watchOS и многе друге.
* Софтвер јавно доступног кода
* Модуларне компоненте
* Континуално превођење
* Комплетно заснован на *NuGet* пакетима
* Обједињава *MVC* и *Web API*
* Омогућава извршавање различитих верзија на истом рачунару

 На слици (Слика 1) је приказана .NET 5 платформа, типови апликација које се могу развијати на њој, алати у којима се могу програмирати те апликације и сажети костур инфраструктуре[3].

*Слика 1* *Приказ .NET 5 платформе*

* + 1. Entity Framework Core 5.0

*Entity Framework Core 5.0* [4] je технологија за приступ подацима, која је ујендо и мултиплатформска, проширива, једноставнија верзија популарног радног пакета *Entity Framework*. Служи за објектно-релационо мапирање (ORM) и представља спрегу између објектно орјентисане и релационе архитекруре. Омогућава програмерима да раде са базама података коришћењем .NET објеката. Такође, елиминише потребу за писањем гломазног кода који је потребан за притуп бази података.

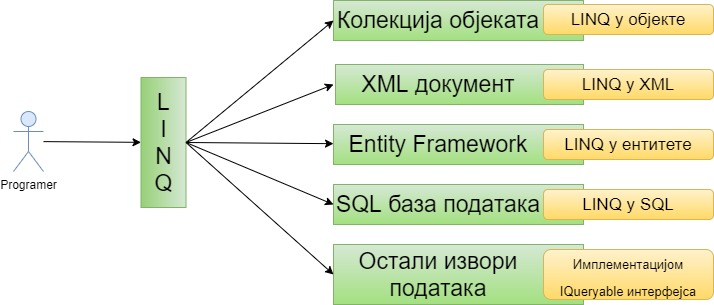
*Entity Framework Core* може приступити разним база података помоћу уграђених библиотека, као што су SQL базе података, SQLite, MySQL, PostgreSQL, Oracle и многе друге. Подржава коришћење *LINQ* упита, праћење промена, ажурирање и миграције шема.

* + 1. *Language Integrated Query (**LINQ)*

Писање упита као стринга, тј. низа карактера, је превазиђен начин програмирања, поготову јер често доводи до грешака које испољавају тек при покретању кода. Додатни проблем је што је потребно научити различите синтаксе упита за сваки тип извора података: SQL базе података, XML документа, бројни веб сервиси итд.

Наведени проблеми се могу лако превазићи коришћењем *LINQ* скупа технологија. LINQ је униформна синтакса за писање упита, која се користи у С# за преузимање података из различитих извора и формата. Интегрисана је у С# чиме се елиминише несклад између програмских језика и база података, као и обезбеђује јединствени интерфејс за упите за различите врсте извора података[5].

LINQ упити враћају резултате извршавања у виду објеката, што омогућава коришћење објектно орјентисаног приступа на скупу резултата и уклања бригу о конверзији различитих формата резултата у објекте[6].



Слика 2 Шематски приказ коришћења LINQ

* + 1. SQLite

SQLite је систем за управљање релационим базама податакa садржан у релативно малој С програмској библиотеци. Изворни код за SQLite је у јавном власништву што значи да се може користити у све сврхе, комерцијалне или приватне.

SQLite је ACID[[1]](#footnote-1)-компатибилан и имплементира већину SQL стандарда, углавном пратећи PostgreSQL синтаксу. Популаран је као уграђени софтвер базе података за локално складиштење у апликативном софтверу као што су веб прегледачи[7].

* + 1. AutoMapper

AutoMapper је библиотека која се користи за пресликавање једног објекта у други. Функционише тако што трансформише улазни објекат једног типа у излазни објекат неког другог типа. Оно што чини AutoMapper занимљивим је што обезбеђује конвенцију која олакшава мапирање објекта А у објекат Б. Скоро да није потребна додатна подешавања два типа, докле год објекат Б типа прати установљену конвенцију AutoMapper-а[8].

AutoMapper решава проблем писања монотоног и досадног кода за мапирање два типа. Мапирање се врло лако може подесити да буде и двосмерно, што значи и да се тип Б може трансформисати у тип А

* + 1. Html Agility Pack

*Html Agility Pack* [9]је библиотека за парсиранје и преузимање садржаја са статичких HTML веб страница. Заснована је на технологијама веб индексирања. Коришћењем функција које нуди, могуће је из HTML кода веб странице преузети податке који се налазе унутар градивних елемената.

Пре коришћења библиотеке над неком страницом, потребно је детаљно проучити њену структуру, тј. анализирати њен код и издвојити карактеристике које једнозначно одређују елементе од интереса.

* + 1. Puppeteer

*Puppeteer* је још једна библиотека за парсирање и преузимање садржаја са веб страница. Њена предност је што се може користити и за парсирање динамичких JavaScript страница. *Puppeteer* је креиран у Јануару 2018. године од стране Гугла и веома брзо је постао један од два водећа развојна оквира. Други водећи развојни оквир је Selenium, који није коришћен при развоју ове апликације.

Поред парсирања једностраних апликација, *Puppeteer* се може користити и за аутоматизовано попуњавање форми, унос са тастатуре или тестирање корисничког интерфејса, као и осталих функционалности. У реализацији ове апликације коришћен је само за парсирање веб страница.

Мана *Puppeteer* -а је што потребно много више времена за парсирање од *Html Agility Pack*-а, али омогућава дохватање динамички учитаних делова странице.

* 1. Технологије коришћене на клијентској страни

Клијентска страна апликације представља део апликације са којом корисник има директан додир и она уједно представља и сам изглед апликације у претраживачу. Технологије на клијентској страни подразумевају основне језике попут HTML-a, CSS-a JavaScript-a које веб претраживач интерпретира. Поред ових неизоставних делова клијентске апликације *Angular* је компатибилан и са многим библиотекама које олакшавају посао програмера, а доприносе изгледу и динамици апликације.

* + 1. Hypertext Markup Language (HTML)

*HTML* je описни језик специјално намењен опису веб страница. Помоћу њега се једноставно могу одвојити елементи као што су наслови, параграфи, линкови, слике и слично. Веб претраживачи од сервера прихватају *HTML* документа и рендерују их како би садржај веб странице био приказан корисницима.

*HTML* има јасно дефинисан скуп елемената (тагова), који представљају његову синтаксу. Елементи се састоје из назива елемента окруженог са „<“ и „>“ и притом сваки елемент мора имати и одговарајући затварајући елемент окружен са „</“ и „>“. Најчешће коришћени елементи су <head>, <body>, <img>, <a>, <div>, <p>, <input> …

* + 1. Cascading Style Sheets (CSS)

*CSS* је језик форматирања помоћу ког се дефинише изглед елемената веб-странице. Може се придружити *HTML*-у као засебан фајл, писањем у одређеном делу *HTML* фајла или у оквиру сваког *HTML* елемента у оквору syle атрибута. Најпрегледнији начин, који је и примењен у реализацији апликације је писањем *CSS* кода у засебним фајловима.

* + 1. JavaScript

*JavaSript* [10] је један од најзаступљенијих језика коришћених за интернет програмирање. Платформски је неутралан, а уједно и објектно базиран језик. Додатна предност је што није строго типизиран језик, већ се тип променљиве динамички додељује. Иако се најчешће користи за писање скрипти за веб странице, многа жења без преглача га користе, нпр: Node.js, Apache CouchDB и Adobe Acrobat.

За развој ове апликације коришћен је *TypeScript* који се у позадини преводи у *JavaSript*, како би га претраживач могао интерпретирати .

* + 1. TypeScript

*TypeScript* је надскуп *JavaSript* језика и има један компајлер, изворног кода, који развија један произвођач: Мајкрософт. Циљ *TypeScript*-аје да помогне у раном откривању грешки проурокованих неподударањем типова променљивих и тиме обезбеди развој *JavaSript* апликација још ефикаснијим. Развијен је и као клијентски и као серверски језик.

* + 1. Angular 12

*Angular* је радни оквир за развој савремених апликација за веб, мобилне и десктоп платформе. *Angular* притом обезбеђује чист и структуриран приступ писању кода, као и добру проширивост и висок степен реупотребе. Заснован је на *HTML*-у и *СЅЅ*-у. *Angular* има добру компатибилност са другим технологијама (веза са серверском страном ка Node.JS или MVC .NET) што је разлог зашто је одабран за развој ове апликације.

* + 1. ngx – bootstrap

*ngx – bootstrap* библиотека је верзија Bootstrap-а подржана од стране Angular-а. Angular не подржава стандардни Bootstrap јер он у позадини користи JQuery библиотеку. Све основне компоненте Bootstrap-а, као и неке додатне компоненте су подржане у *ngx – bootstrap* библиотеци.

* + 1. JavaScript Object Notation (ЈЅОN)

*JSON* је формат представљања података у текстуалном облику, дизајниран за размену података између удаљених система, којa се постиже механизмом серијализације података, односно превођењем података у формат погодан за пренос путем мреже. *JSON* се темељи на синтакси *JavaScript* објеката, а будући да већина веб апликација користи JavaScript са клијентске стране, *JSON* је постао стандард за паковање података на вебу, потискујући *XML* формат.

1. Опис рада система
   1. Нерегистровани корисници

Опис свих функционалности које су доступне нерегистрованим корисницима система се налази у овој секцији. Нерегистровани корисници имају могућност регистрације као купац и пријављивања на систем, прегледа садржаја апликације.

* + 1. Регистрација корисника

При регистрацији корисници се могу регистровати само као купци и тада попуњавају форму где уносе име, презиме, мејл, јединсвено корисничко име, лозинку, коју морају два пута унети, из безбедносних разлога, датум рођења и пол.

* + 1. Пријављивање на систем

При пријављивању на систем, корисник уноси корисничко име или мејл и лозинку. У случају успешног пријављивања корисник се преусмерава на одговарајућу страницу. У случају неуспешног пријављивања, кориснику се исписује порука о грешци.

* + 1. Преглед садржаја апликације

Преглед садржаја представља приказивање свих производа, који се налазе у бази, кориснику. Коришћењем различитих филтера корисник може да промени број производа који му се приказују.

* 1. Регистровани корисници

Корисницима који су регистровани, након пријављивања нуди се број различитих функционалности. Функционалности које ће им бити на располагању зависе од типа корисника, тј. да ли су купац или администратор.

* + 1. Купац

Купац, од додатних функционалности има могућност убацивање производа у корпу и избацивање садржаја из корпе, преглед садржаја корпе као и куповина садржаја корпе. За купца се трајно памти садржај корпе (тј. док је он сам не испразни). Купац може и да врши промену својих информација.

* + - 1. Преглед историје поруџбина

Када корисник пожели да изврши преглед поруџбине приказује му се табеларни преглед поруџбина, сортираних по датуму куповине, од скоријег датума ка ранијим. Корисник може да промени колону по којој жели да сортира поруџбине (цени, количини артикала или датуму поруџбине).

* + - 1. Преглед садржаја корпе

При прегледу садржаја корпе, кориснинку се приказује табеларни преглед садржаја корпе, као и опције да промени количину једног производа у корпи или да га избаци из корпе.

* + - 1. Куповина садржаја корпе

Када корисник пожели да купи садржај корпе, потребно је да попуни форму са подацима потребним за испоруку. Куповина садржаја је заправо само симулација куповине, тј. производи се не наручују из стварних продавница, јер то није циљ дипломског рада.

* + - 1. Измена личних информација

Корисник има могућност да промене свих својих информација, осим корисничког имена.

* + 1. Администратор

У систему постоји јединствен администратор. Он може да врши преглед података о купцима, може да изврши преглед активних „crawler“-а и изврши измену активних.

* + - 1. Преглед активности „crawler“-а

Када врши преглед активности „crawler“-а, администратору се приказује табеларни преглед, а као назив „crawler“-а приказује назив продавнице над којом „crawler“ ради. Поред сваког назива се налази и поље за потврду (checkbox), које ако је означено, „crawler“ ради, тј. дохвата податке са одговарајућих страница, а ако није означено, „crawler“-а је блокиран, тј. не дохвата податке. Промена статуса „crawler“-а се врши одмах при промени означености поља.

* + - 1. Уклањање свих артикала из одређене продавнице

При прегледу

1. Реализација система
2. Закључак
3. Литература
4. C Sharp, доступан на веб адреси: <https://sr.wikipedia.org/wiki/C_Sharp>, посећен дана 10. 9. 2021. године
5. What's new in .NET 5, доступан на веб адреси: <https://docs.microsoft.com/en-us/dotnet/core/dotnet-five>, посећен дана 10. 9. 2021.
6. Introducing .NET 5, доступан на веб адреси:  
   <https://devblogs.microsoft.com/dotnet/introducing-net-5/>, посећен дана 10. 9. 2021. године
7. Entity Framework Core, доступан на веб адреси: <https://docs.microsoft.com/en-gb/ef/core/>, посећен дана 10. 9. 2021. године.
8. Language Integrated Query (LINQ): <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>, посећен дана 10. 9. 2021. године
9. What is LINQ?: <https://www.tutorialsteacher.com/linq/what-is-linq>, посећен дана 10. 9. 2021. године.
10. About SQLite, доступан на веб адреси: <https://www.sqlite.org/about.html>, посећен дана 10. 9. 2021. године.
11. What is AutoMapper?, доступан на веб адреси: <https://docs.automapper.org/en/stable/Getting-started.html>, посећен дана 10. 9. 2021. године.
12. Html Agility Pack: <https://html-agility-pack.net/from-web>, посећен дана 10. 9. 2021. године.
13. JavaScript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, посећен дана 10. 9. 2021. године.
14. Jennifer Marsh – „Web Scraping with C#“ – доступан на веб адреси: <https://www.scrapingbee.com/blog/web-scraping-csharp/>, посећен дана 10. 9. 2021. године.
15. Puppeteer <https://pptr.dev/>, посећен дана 9. 9. 2021. године.
16. Sam Crowther – What is Puppeteer? Why Developers and Fraudsters Love it - доступан на веб адреси: <https://www.kasada.io/what-is-puppeteer-developers-fraudsters-love-it/>, посећен дана 9. 9. 2021. године.
17. Страница која омогућава декодовање, верификацију и генерисање JWT-а – дсотупно на веб адреси: <https://jwt.io/>, посећен дана 10. 9. 2021.

1. ACID (atomicity, consistency, isolation, durability) – (атомичност, исправност, изолација, трајност) скуп својства трансакције базе података [↑](#footnote-ref-1)