ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ  
ПРИРОДНО – МАТЕМАТИЧКИ ФАКУЛТЕТ  
УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ

Завршни рад

Сајт за преглед цена ИТ производа

**Ментор**

**др Владимир Цвjетковић**

**Студент**

**Андрија Лазић 53/2019**

Крагујевац, 2024.

Садржај:

[1 Увод 3](#_Toc160141227)

[2 Коришћене технологије 4](#_Toc160141228)

[2.1 MongoDB 4](#_Toc160141229)

[2.2 React 4](#_Toc160141230)

[2.3 Node.js 5](#_Toc160141231)

[2.4 Express.js 5](#_Toc160141232)

[2.5 TypeScript 5](#_Toc160141233)

[2.6 Redis 5](#_Toc160141234)

[2.7 Python 6](#_Toc160141235)

[2.8 Scrapy 6](#_Toc160141236)

[2.9 Git 6](#_Toc160141237)

[2.10 Postman 6](#_Toc160141238)

[3 Структура пројекта 7](#_Toc160141239)

[3.1 Архитектура система 7](#_Toc160141240)

[3.2 Листа функционалности 8](#_Toc160141241)

[3.3 Дијаграми случајева коришћења 8](#_Toc160141242)

[3.4 МОВ дијаграм 10](#_Toc160141243)

[Литература 12](#_Toc160141244)

[Кратка биографија кандидата 13](#_Toc160141245)

# Увод

Савремени човек, који је посвећен свом послу и напретку, има мало слободног времена. Ово доводи до тога да човек не може или не жели да издвоји мало времена како би се што боље информисао пре куповине производа.

Продавци ово користе како би својим купцима продали застареле и неповољне производе. Најбољи пример ове појаве код нас је “Акцијска распродаја” за Црни петак, где продавци месец дана раније подигну цене производа само да би их за Црни петак вратили и тврдили нереалне попусте. Ова манипулација цена остаје незапажена од стране купаца који се надају повољним ценама и великим уштедама.

Апликација “EzDeals” је настала као решење за горе наведене проблеме.

# Коришћене технологије

Пројекат се састоји из два дела: WEB апликација и SCRAPER сервис.

У основи WEB апликације се налази “MERN stack” који се састоји од следећих технологија:

* MongoDB
* Express.js
* React.js
* Node.js

Уз “MERN stack” је коришћена и REDIS база података за кеширање података у радној меморији. SCRAPER сервис је развијен у Python окружењу на основу Scrapy framework-a.

## MongoDB

MongoDB је NoSQL база података која складишти податке у BSON (Binary JSON)

формату. Свака база се састоји од једне или више колекција а свака колекција у себи садржи документе у BSON формату.

Предности NoSQL база су:

* Флексибилност- NoSQL базе података лако могу обрадити било који формат података, као што су структурисани, полу-структурисани и неструктурисани подаци, у једном складишту података. Ово омогућава брз и поједностављен развој апликације.
* Скалабилности- у случају да нам је потребан већи меморијски простор можемо цео систем скалирати у ширину(изнајмљивање нових сервера) што је јефтиније и лакше од скалирања у висину(надоградња тренутног сервера).
* Високе перформансе- због честих измена података очекују се много боље перформансе у односу на традиционалне базе података. Ово је један од главних разлога што је одабрана NoSQL база података
* Лака имплементација- из базе можемо читати и уписивати JSON објекте што олакшава приступ подацима.

## React

React је библиотека за развој интерактивних и динамичних веб апликација. У тренутку писања овог документа је и најпопуларнија frontend технологија.

Кориснички интерфејс се разбија на мање компоненте које су представљене у JSX формату. Више компонената се комбинују у појединачне стране и једна компонента се може искористити у креирању више различитих страна. Могућност поновне употребе компонената је једна од главних особина React-а која знатно убрзава развој апликације. Модуларна структура такође омогућава лакше одржавање кода што додатно утиче на време потребно за израду апликације.

Virtual DOM представља копију DOM-а и омогућава React-у да утврди промене настале од стране корисника и позадинских процеса и на основу датих промена одлучује које компоненте ће се поново рендеровати.

## Node.js

Node.js је open-source JavaScript окружење које нам омогућава да покренемо JavaScript код изван интернет претраживача. Ово је постигнуто помоћу Гугловог V8 engine-a. Node.js се може користити на различитим платформама: Windows, Linux, Unix, Mac OS X…

Кључне карактеристике Node.js-а су:

* Неблокирајући и асинхрон, што га чини погодним за управљање конкурентним конекцијама и извршавање задатака као што су унос/износ података из фајлова, мрежни захтеви и операције са базама података без блокирања извршавања читавог програма.
* NPM (Node Package Manager), екосистем отворених библиотека и модула доступних за JavaScript. Лака инсталација и надоградња пакета што додатно омогућава лак транспорт пројекта са једног система на други.
* Заједница и екосистем: Node.js има активну заједницу која доприноси развоју библиотека и тиме олакшава процес израда веб апликација.

## Express.js

Express је framework имплементиран унутар Node.js-а који се користи за развој серверске стране веб апликације коришћењем JavaScript језика. У овом пројекту је конкретно коришћен за дефинисање HTTP рута и middleware-a.

## TypeScript

Бесплатан и open-source језик који представља надоградњу JavaScript-а. Главна разлика између TypeScript-а и JavaScript-а је коришћење статичких типова насупрот динамичких што омогућава рано детектовање грешака и лакши развој апликације.

TypeScript подржава наслеђивање што омогућава креирање хијерархије класа.

TypeScript код се компајлира у JavaScript код који може бити извршен на било којем JavaScript окружењу. Ова особина нам је омогућила да TypeScript користимо при развоју React и Node.js апликација.

## Redis

Open-source, in-memory база података која се може користити за кеширање, стримовање, сервис за дистрибуцију порука…

У овом пројекту Redis је коришћен као систем за кеширање резултата одређених HTTP захтева што позитивно утиче на перфромансе. Једина мана овог система је што Redis није подржан на windows системима па је потребно користити виртуелну машину са Linux оперативним системом.

## Python

Python је одабран као основа scraping сервиса због великог броја доступних библиотека, framework-ова и алата за обраду података.

## Scrapy

Scrapy је open-source framework који се користи за преузимање различитих врста података са веб-а. Највећа предност овог framework-а је лако одржавање и надоградња што је у овом пројекту битно због честих измена и додавања нових производа. Пошто је framework написан у Python-у то му омогућава да ради на различитим окружењима као што су : Linux, Windows, Mac…

Scrapy се састоји од појединачних радника(Spider) који су задужени за рад на појединачним продавницама. Ово доводи до бољих перформанси и лакшег одржавања.

## Git

Git је бесплатан и open-source систем за контролу верзије кода. Главна функционалност Git-a је праћење измена у коду и другим фајловима током времена. Ово вам омогућава да видимо развој пројекта током времена и да се вратимо на претходне верзије ако је то потребно услед техничких проблема.

## Postman

Postman је познати алат који се користи за тестирање веб API-a које може бити ручно или аутоматско.

# Структура пројекта

Пре него што започнемо рад на самом пројекту добра пракса је да се уради спецификација дизајна софтвера у којој ће бити описане следеће ставке:

* Листа функционалности
* Архитектура система
* Дијаграми случајева коришћења
* Модел објекти везе
* Изглед корисничког интерфејса

## Архитектура система

Апликација „EzDeals“ ће се састојати из три самосталне целине: клијентски део, серверски део и микросервис за „web scraping“, као и базе података којој ће приступати и серверски део и микросервис. Све три целине ће бити на истом серверу ради практичности, али постоји могућност да се налазе на дистрибуираном систему уз минималне исправке конфигурационих фајлова. Ова карактеристика нам омогућава да лако проширимо цео система ако апликација буде захтевала више рачунарских ресурса. Комуникација између њих ће се одвијати путем интернета користећи HTTP протоколе.

A diagram of a diagram

Description automatically generated

## Листа функционалности

У оквиру система ће постојати једна врста корисника а то је **нерегистровани** корисник. Они ће моћи да врше претрагу производа и имаће увид у тренутне цене код различитих продаваца као и историју цена.

Апликација мора да испуни следећу листу функционалности:

1. Обавезне функционалности
   1. Претрага производа
   2. Визуелизација цена производа
2. Пожељне функционалности
   1. Поређење производа
   2. Систем за обавештавање корисника о попустима
3. Опционе функционалности
   1. Логин и регистрација корисника
   2. Остављање коментара о производу

## Дијаграми случајева коришћења

A diagram of a diagram

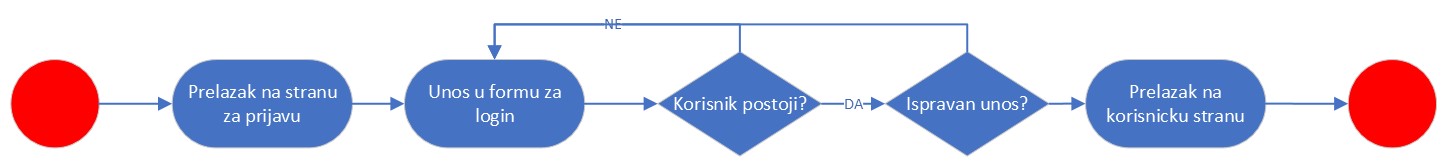
Description automatically generated

Дијаграм активности претраге производа

A diagram of a diagram

Description automatically generated

Дијаграм активности писања коментара



Дијаграм активности пријаве корисника

A diagram of a flowchart

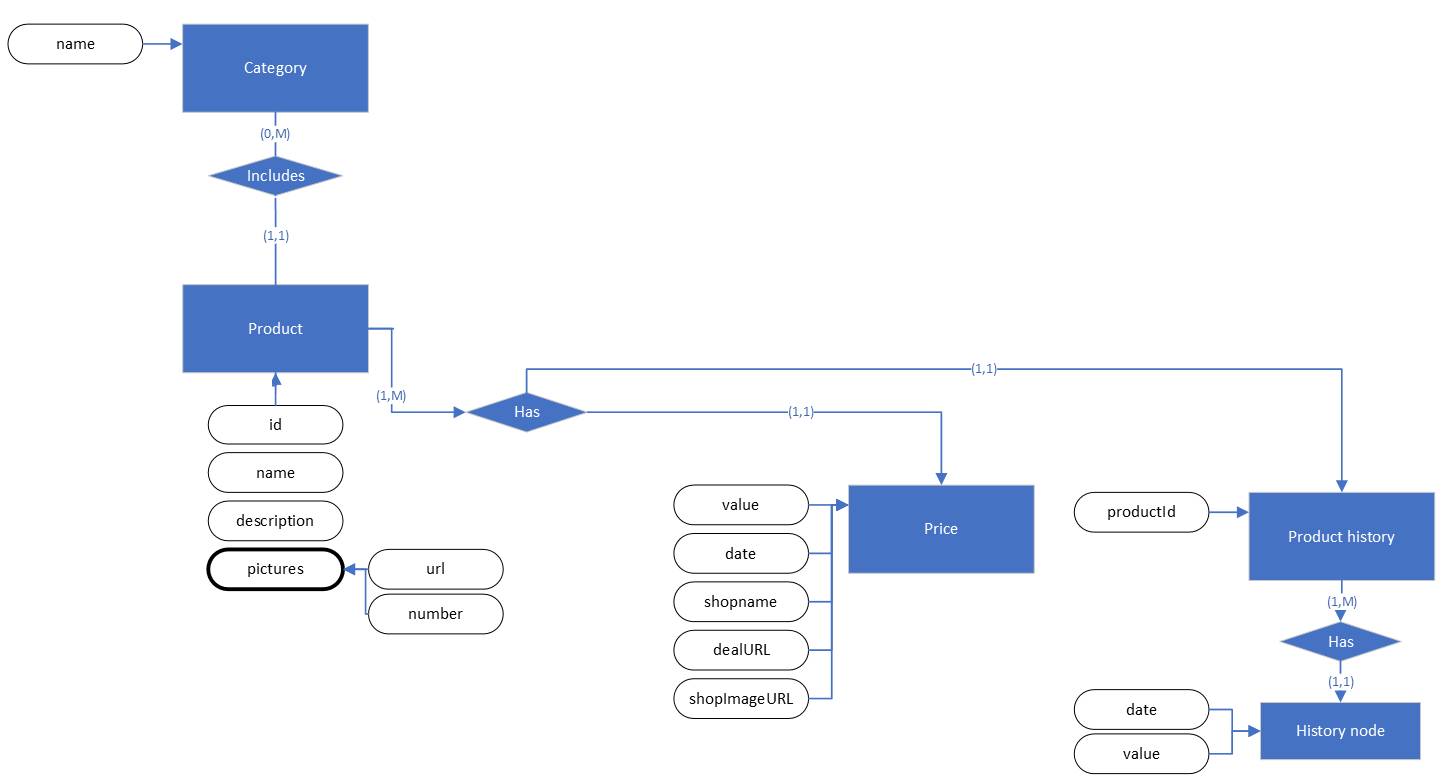
Description automatically generated

Дијаграм активности пријаве корисника

## МОВ дијаграм

МОВ(Модел објекти и везе) дијаграм представља визуелну репрезентацију базе података. Кључне компоненте ER дијаграма су:

* Ентитети: Објекти односно концепти у систему. Представљени помоћу правоугаоника
* Атрибути: Описују објекте
* Везе: Између објеката постоје везе које могу имати различите кардиналности.



# Структура кода и организација

## WEB Scraper

WEB Scraper је развијен на Scrapy [[1]](#ref1) framework-у унутар Python [[2]](#ref2) развојном окружењу. Главна компонента Scrapy-a су Spider-и који се користе за scraping. Постоје два начина да се један сајт scrape-ује. Један је директно преузимање потребних података са API-a датог сајта, што је и најбржа метода scrape-овања. Друга метода подразумева пролазак кроз појединачне HTML стране и преузимање потребних података помоћу xpath-a и css селектора. Из овог разлога за сваки веб сајт је потребно направити посебан Spider зато што сваки сајт има различиту структуру.

### Пример креирања spider-a

#### Креирање класе

Процес креирања Spider-a започињемо креирањем класе која треба да наследи класу Spider из библиотеке Scrapy. У класу дефинишемо име, почетни url и стране које желимо да scrape-ујемо.

|  |
| --- |
| **Изворни код 4.1: Дефинисање spider класе** |
| 1. class GigatronscraperSpider(scrapy.Spider):  2. name = "gigatronScraper"  3. allowed\_domains = ["gigatron.rs"]  4. start\_urls = ["https://gigatron.rs"]  5. dt\_string=""  6.  7. pagesToScrape=[  8. PageInfo("https://search.gigatron.rs/v1/catalog/get/racunari-i-komponente/komponente/procesori?poredak=opadajuci&strana=",1,"RacunarskeKomponente"),  9. PageInfo("https://search.gigatron.rs/v1/catalog/get/racunari-i-komponente/komponente/maticne-ploce?poredak=opadajuci&strana=",1,"RacunarskeKomponente"),  10. PageInfo("https://search.gigatron.rs/v1/catalog/get/racunari-i-komponente/monitori?poredak=opadajuci&strana=",1,"Monitori"),  11. PageInfo("https://search.gigatron.rs/v1/catalog/get/tv-audio-video/slusalice?poredak=opadajuci&strana=",1,"Slusalice"),  12. PageInfo("https://search.gigatron.rs/v1/catalog/get/prenosni-racunari/laptop-racunari?strana=",1,"Laptopovi"),  13. PageInfo("https://search.gigatron.rs/v1/catalog/get/mobilni-telefoni-i-oprema/mobilni-telefoni?strana=",1,"MobilniTelefoni"),  14. PageInfo("https://search.gigatron.rs/v1/catalog/get/oprema-za-racunare/smestanje-podataka/eksterni-hdd-i-ssd?strana=",1,"EksterniDiskovi")] |

#### Функција за иницијализацију

Ова функција је наслеђена апстрактна функција која се извршава приликом иницијализације spider-a. Прво проверавамо да ли имамо приступ датој страни и памтимо време када је scrape-овање започето. На крају итерирамо кроз претходно дефинисане стране и шаљемо захтеве са њиховим url-овима и креирамо Listener којем ће се резултати захтева проследити када се захтев заврши.

|  |
| --- |
| **Изворни код 4.2: Функција за иницијализацију** |
| 17. def parse(self, initialResponse):  18. if(initialResponse.status!=200):  19. print("Could not access the: "+self.start\_urls[0])  20. return  21. self.now = datetime.now()  22. self.dt\_string = self.now.strftime("%d/%m/%Y %H:%M")  23.  24. for page in self.pagesToScrape:  25. yield scrapy.Request(  26. url=page.getCurrentURL(),  27. callback=self.parsePage) |

#### Пример scrape-овања позивањем API захтева

Када се захтев изврши резултат се прослеђује функцији parsePage. У колико је захтев успешан узимамо податке из response-a и прелазимо на следећу страну ако постоји.

|  |
| --- |
| **Изворни код 4.3: Пример scrape-овања позивањем API захтева** |
| 29. def parsePage(self,response):  30. if(response.status!=200):  31. print("Could not access the: "+response.url)  32. return  33.  34. currentPage=None  35. for page in self.pagesToScrape:  37. if page.URL in response.url:  38. currentPage=page  39. break  40.  41. data=response.json()  42.  43. if(currentPage.maxIndex==-1):  44. currentPage.maxIndex=data["totalPages"]  45.  46. for hit in data["hits"]["hits"]:  47. rowJSON=hit["\_source"]["search\_result\_data"]  48. price=rowJSON["price"].split(".")[0]  49. product=Product(rowJSON["name"],rowJSON["image"],self.dt\_string,currentPage.category)  50. product.addPrice(Price(price,"Gigatron",self.start\_urls[0]+rowJSON["url"],"https://gigatron.rs/images/gigatron.png"))  51. yield product  52.  53. if currentPage.index>currentPage.maxIndex:  54. return #self.products  55.  56. yield scrapy.Request(  57. url=currentPage.getCurrentURL(),  58. callback=self.parsePage) |

#### Пример scrape-овања помоћу xpath-a

|  |
| --- |
| **Изворни код 4.4: Пример scrape-овања помоћу xpath-a** |
| 28. def parsePage(self,response):  29.  30. if(response.status!=200):  31. print("\n")  32. print("Could not access the: "+response.url)  33. print("\n")  34. return  35.  36. currentPage=None  37. for page in self.pagesToScrape:  38.  39. if page.URL in response.url:  40. currentPage=page  41. break  42.  43. productsElements=response.xpath('.//div[@class="shop-product-card relative"]')  44.  45. if len(productsElements)==0:  46. return  47.  48. for productEl in productsElements:  49. pictureURL=(productEl.xpath(".//div[@class='product-image-wrapper']/a/img/@src")).get()  50. if pictureURL is None:  51. pictureURL=""  52.  53. name=(productEl.xpath('.//h2[@class="product-name"]/a/text()').get()).strip()  54. product=Product(  55. name,  56. pictureURL,  57. self.dt\_string,  58. currentPage.category  59. )  60.  61. price=(productEl.xpath('.//div[@class="price-holder"]/div[@class="text-bold"]/span/following-sibling::text()[1]')).get()  62.  63. #if product is on action  64. if price is None:  65. price=(productEl.xpath('.//div[@class="price-holder"]/span[@class="text-bold"]/span/following-sibling::text()[1]')).get()  66.  67. price=price.strip().split(",")[0]  68. price=price.split(".")[0]+price.split(".")[1]  69.  70. product.addPrice(Price(  71. price,  72. "Gstore",  73. (productEl.xpath('.//h2[@class="product-name"]/a/@href').get()).strip(),  74. "https://www.gstore.rs/images/gstore-final-logo-with%20background%20w%20or%20b-21.png"))  75. yield product  76.  77. yield scrapy.Request(url=currentPage.getCurrentURL(),callback=self.parsePage) |

# Литература

|  |  |
| --- | --- |
| [1] | [На мрежи]. Available: https://scrapy.org/. [Последњи приступ 3 3 2024]. |
| [2] | [На мрежи]. Available: https://www.mongodb.com. [Последњи приступ 3 3 2024]. |
| [3] | [На мрежи]. Available: https://redis.io/. |
| [4] | [На мрежи]. Available: https://www.python.org/. |
| [5] | [На мрежи]. Available: https://www.typescriptlang.org/. |
| [6] | [На мрежи]. Available: https://expressjs.com/. [Последњи приступ 3 3 2024]. |

Кратка биографија кандидата

Андрија Лазић је рођен **10.** **априла** **2000**. године у **Крагујевцу**. Основну школу „**Сретен Младеновић**“ у **Десимировцу** **завршио** је **2016**. године, а затим исте године уписује смер **Електротехничар рачунара** у **Првој техничкој школи у Крагујевцу**. Након завршене средње школе уписује основне академске студије на Природно-математичком факултету Универзитета у Крагујевцу, на Институту за математику и информатику, смер информатика.

Универзитет у Крагујевцу

Природно-математички факултет

Институт за математику и информатику

Завршни рад под називом

Назив теме

одбрањен је \_\_\_\_\_\_\_\_\_\_\_\_.

МЕНТОР:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

др Име Презиме, звање,

Институт за математику и информатику

Природно-математички факултет Крагујевац

ЧЛАНОВИ КОМИСИЈЕ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

др Име Презиме, звање,

Институт за математику и информатику

Природно-математички факултет Крагујевац

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

др Име Презиме, звање,

Институт за математику и информатику

Природно-математички факултет Крагујевац

Завршни рад је оцењен оценом \_\_\_\_\_\_\_\_\_\_\_\_.