

**Departamenti i Shkencave Kompjuterike**

**SFIDAT PËRGJATË WEB CRAWLING TË PATUNDSHMËRIVE ONLINE NË KOSOVË (RAST STUDIMI: GJIRAFA, MERR JEP DHE OFERTA SUKSESI)”**

Niveli Bachelor

Pleurat Pula

Dhjetor / 2022

Prishtinë



**Departamenti i Shkencave Kompjuterike**

Punim Diplome

2021-2022

Pleurat Pula

**SFIDAT PËRGJATË WEB CRAWLING TË PATUNDSHMËRIVE ONLINE NË KOSOVË** **(RAST STUDIMI: GJIRAFA, MERR JEP DHE OFERTA SUKSESI)**

Mentor: PhD, Arben Damoni

Prishtinë, Dhjetor 2022

Ky punim është përpiluar dhe dorëzuar në përmbushjen e kërkesave të pjesshme për Shkallën Bachelor

**DEKLARATË**

Unë, Pleurat Pula deklaroj se ky punim përfaqëson punën time origjinale dhe se   
të gjitha fjalitë, frazat dhe idetë nga puna e autorëve të tjerë të cilat janë përdorur në këtë punim janë cituar me referenca të qarta dhe të drejtpërdrejta të lidhura me autorin, punimin dhe faqen e caktuar.

Jam i vetëdijshëm që në qoftë se nuk kam vepruar kështu, punimi mund të konsiderohet plagjiarizëm dhe do të përdoret si arsye për dështim në këtë temë si dhe në provimin e diplomës në tërësi.

Emri Mbiemri: \_\_\_\_Pleurat Pula\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nënshkrimi: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Data: \_\_\_\_01.11.2022\_\_\_\_

## **ABSTRAKT**

Sot motorët e kërkimit janë të pajisur me “*Web Crawlers*”. Një “Web Crawler” (zvarritës) ka të bëjë me gjetjen ose zbulimin e URL-ve apo linqeve në ueb. Ndërsa një “Web Scraper” (nxjerrjen) ka të bëjë me nxjerrjen e të dhënave nga një ose më shumë ueb faqe (Kenny, n.d.). Disa ueb scraping skripta funksionojnë në mënyrë autonome dhe nuk nuk kanë nevojë për instruksione shtesë pasi të fillojnë punën dhe disa të tjerë duhet të krijohen specifikisht dhe modifikohen në atë mënyrë që t’i përshtaten kompleksicitetit të ueb faqeve të ndryshme.

Një ndër “*web scraping*” frameworks që do të përdorim përgjatë këtij punimi është “*Scrapy*” i cili është shkruar në gjuhën programuese Python dhe ndryshe njihet edhe si web scraping i levelit të lartë për nga shpejtësia e nxjerrjes së të dhënave të strukturuara nga ueb faqet (Developers, 2022).

Gjatë këtij punimi, do të dokumentohet dhe shpjegohet në detaje ndërtimi i web scrap-it nga fillimi gjerë në fund, rëndësia e tij si web scrape, krahasimin me teknologjitë tjera që kryejnë funksionin e njëjtë, të metat dhe përparësitë, komponentet dhe qëllimi i përdorimit të tyre, etj. Të gjitha këto do të shtjellohen shkurtimisht në mënyrë që lexuesi të ketë një njohuri bazike rreth Scrapy dhe që ta ketë më të lehtë orientimin në hulumtim.

Në këtë punim, si pjesë e hulumtimit janë marrë tre nga ueb faqet më të njohura dhe me më së shumti të dhëna në Kosovë, të cilat posedojnë të dhëna për patundshmëri. Përgjatë këtij punimi do të bëjmë ndërtimin e skriptave të ndryshme duke përdorur framework të gatshëm, mes tjerash edhe duke përdorur IDE për scraping ku pastaj do të krahasohen rezultatet.

## **MIRËNJOHJE DHE FALENDERIME**

Përfundimi i këtij punimi nuk do të realizohej me sukses pa inkurajimin, mbështetjen dhe ndihmën e disa personave ndaj të cilëve dua të shpreh mirënjohjen time të thellë.

Kam pasur nderin dhe kënaqësinë që të udhëhiqem nga Prof. Dr. Arben Damoni, të cilit i jam shumë mirënjohës për udhëzimet, këshillat dhe motivimin në mënyrë që ky punim të ketë një paraqitje sa më të mirë.

Një falenderim i përzemërt u takon të gjithë profesorëve të nderuar të kolegjit Riinvest për dijen, mbështetjen dhe bashkëpunimin e ofruar përgjatë këtyre tre viteve të studimeve.

Për fund, shpreh mirënjohjen e thellë ndaj prindërve e mi, të cilët qysh në vogëli më edukuan me kulturën e vlerësimit për veten, dëshirën për punën, vullnetin dhe këmbënguljen, pa të cilat nuk do t’ia kisha dalë mbanë.

**TABELA E PËRMBAJTJES**

[ABSTRAKT IV](#_Toc124371078)

[MIRËNJOHJE DHE FALENDERIME V](#_Toc124371079)

[FJALORI DHE SHKURTESAT E TERMAVE X](#_Toc124371080)

[1. HYRJE 1](#_Toc124371081)

[2. SHQYRTIMI I LITERATURËS 2](#_Toc124371082)

[2.1. Çfarë është Web Crawling? 2](#_Toc124371083)

[2.2. Cfarë është Web Scraping? 2](#_Toc124371084)

[2.3. Scrapy vs Beautiful Soup vs Selenium 3](#_Toc124371085)

[2.3.1. Libraria Beautiful Soup 4](#_Toc124371086)

[2.3.2. Libraria Selenium 4](#_Toc124371087)

[2.3.3. Shablloni Scrapy 5](#_Toc124371088)

[3. DEFINIMI I PROBLEMIT 7](#_Toc124371089)

[3.1. Komponentet kryesore të Scrapy 7](#_Toc124371090)

[3.1.1. Spiders 7](#_Toc124371091)

[3.1.2. Item Pipeline 8](#_Toc124371092)

[3.1.3. Middleware 8](#_Toc124371093)

[3.1.4. Engine 8](#_Toc124371094)

[3.1.5. Scheduler 8](#_Toc124371095)

[3.2. Arkitektura dhe ndërveprimi i komponenteve të Scrapy-t 9](#_Toc124371096)

[3.3. Komandat e Scrapy Framework 10](#_Toc124371097)

[3.4. Selektorët si mekanizma dhe përdorimi i tyre 12](#_Toc124371098)

[3.4.1. XPath Selektorët 12](#_Toc124371099)

[3.4.2. CSS Selektorët 12](#_Toc124371100)

[3.4.3. RegEx Selektorët 13](#_Toc124371101)

[3.5. Sfidat përgjatë Web Scraping dhe tejkalimi i tyre 13](#_Toc124371102)

[3.5.1. Pengesat Ligjore 13](#_Toc124371103)

[3.5.2. Pengesat teknike 14](#_Toc124371104)

[3.5.2.1. Menaxhimi i qasjes së botëve përmes dosjes Robots.txt 15](#_Toc124371105)

[3.5.2.2. Rate Limitting Requests dhe bllokuesit e IP-ve 16](#_Toc124371106)

[3.5.2.3. Bypass Cloudflare 18](#_Toc124371107)

[3.5.2.4. Ndryshimet në strukturën e ueb faqeve 18](#_Toc124371108)

[3.5.2.5. Bllokuesit CAPTCHA 19](#_Toc124371109)

[3.5.2.6. Përmbajtja dinamike 19](#_Toc124371110)

[3.5.2.7. Honey Pots 20](#_Toc124371111)

[3.5.2.8. Login Requirements 20](#_Toc124371112)

[3.5.2.9. Slow Loading Speed 21](#_Toc124371113)

[4. METODOLOGJIA DHE VEGLAT 22](#_Toc124371114)

[5. ANALIZA E KËRKESAVE, IMPLEMENTIMI DHE REZULTATET ME OBSERVIME 24](#_Toc124371115)

[5.1. Analiza e strukturës së faqeve 24](#_Toc124371116)

[5.2. Kontrolloimi i ueb faqeve për robots.txt 24](#_Toc124371117)

[5.3. Kontrollimi i ueb faqeve për Cloudflare? 27](#_Toc124371118)

[5.4. Implementimi 29](#_Toc124371119)

[5.4.1. Ambienti punues dhe instalimi i shabllonit dhe librarive 29](#_Toc124371120)

[6. KONKLUZIONET 46](#_Toc124371121)

[7. REFERENCAT DHE BIBLIOGRAFIA 48](#_Toc124371122)

**TABELA E FIGURAVE**

[Figura 1 - Avantazhet dhe disavantazhet e teknologjive (Andrade, 2020) 3](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451532)

[Figura 2 –Beautiful Soup dhe modulet e varura (Andrade, 2020) 4](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451533)

[Figura 3 - Libraria Selenium (Andrade, 2020) 5](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451534)

[Figura 4 – Procesi i kërkesave asinkrone (Andrade, 2020) 6](#_Toc122451535)

[Figura 5 - Rrjedhja e proceseve dhe të dhënave (Developers, 2022) 9](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451536)

[Figura 6 - Komandat e shabllonit Scrapy 10](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451537)

[Figura 7 - Bench komanda - Përformanca e maqinës për kërkesa në minut 11](#_Toc122451538)

[Figura 8 – Sfidat e Web Scraping (Dilmegani, December 15, 2020) 15](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451539)

[Figura 9 - Norma e kufizimit të kërkesave (Rafik, 2021) 16](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451540)

[Figura 10 – Konsiderimi i kërkesave të shumta si DDOS sulm (Rafik, 2021) 17](#_Toc122451541)

[Figura 11 - Skanimi i ueb faqeve për kontroll të mbrojtjes nga Cloudflare 18](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451542)

[Figura 12 - Një shembull i linkut të përdorur si honeypot (Rafik, 2021) 20](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451543)

[Figura 13 - Robots.txt për www.gjirafa.com 25](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451544)

[Figura 14 - Robots.txt për www.merrjep.com 25](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451545)

[Figura 15 - Robots.txt për www.ofertasuksesi.com 26](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451546)

[Figura 16 - Rregullat që ndjekin botët përgjatë crawling 26](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451547)

[Figura 17 - Skanimi i ueb faqes www.gjirafa.com për kontroll të Cloudflare 27](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451548)

[Figura 18 - Skanimi i ueb faqes www.merrjep.com për kontroll të Cloudflare 28](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451549)

[Figura 19 - Skanimi i ueb faqes www.ofertasuksesi.com për kontroll të Cloudflare 28](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451550)

[Figura 20 - Krijimi i një projekti dhe mjedisi virtual në PyCharm 29](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451551)

[Figura 21 - Menuja për instalimin e librarive dhe shablloneve të ndryshëm. Rasti “Scrapy” 30](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451552)

[Figura 22 - Startimi i projektit përmes komandave të shabllonit Scrapy 30](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451553)

[Figura 23 - Parametrat e sjelljes së botëve 31](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451554)

[Figura 24 - Kërkesat konkurrente për uew faqen www.merrjep.com 31](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451555)

[Figura 25 - Stimulimi i vonesave për kërkesat konkurrente 32](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451556)

[Figura 26 - Kërkesat konkurrente për ueb faqen www.gjirafa.com 32](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451557)

[Figura 27 - Kërkesat konkurrente për domain dhe IP të ueb faqes www.gjirafa.com 32](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451558)

[Figura 28 - Kërkesat konkurrente për ueb faqen www.ofertasuksesi.com 32](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451559)

[Figura 29 - Krijimi i skriptave për nxjerrjen e linqeve nga ueb faqet 33](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451560)

[Figura 30 - Skripta për nxjerrjen e linqeve nga ueb faqja www.gjirafa.com 34](#_Toc122451561)

[Figura 31 - Kodi për faqosje tek skripta për nxjerrjen e linqeve në MerrJep 35](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451562)

[Figura 32 - Përdorimi i parametrit “Meta” për bartjen e itema-ve nga një metodë në tjetrën 36](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451563)

[Figura 33 - Skripta për nxjerrjen e të dhënave për www.gjirafa.com 1/3 38](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451564)

[Figura 34 - Skripta për nxjerrjen e të dhënave për www.gjirafa.com 2/3 40](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451565)

[Figura 35 - Skripta për nxjerrjen e të dhënave për www.gjirafa.com 3/3 41](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451566)

[Figura 36 - Mostër e nxerrjes së të dhënave për patundshmëri nga faqja www.gjirafa.com 42](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451567)

[Figura 37 - Metoda për nxjerrjen e të dhënave nga ueb faqja www.ofertasuksesi.com 42](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451568)

[Figura 38 - Ruajtja e të dhënave në formatin JSON për projektin www.ofertasuksesi.com 43](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451569)

[Figura 39 – Ruajtja e linqeve të procesuara nga projekti www.ofertasuksesi.com 43](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451570)

[Figura 40 - UTF-8 për enkodim të karaktereve Unicode në projektin www.merrjep.com 44](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451571)

[Figura 41 - Libraria scrapy-user-agents me 2200 user-agents 44](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451572)

[Figura 42 - Mostër e nxerrjes së të dhënave për patundshmëri nga faqja www.merrjep.com 45](file:///C:\Users\Pleurat\Desktop\Tema%20Diplomes%20Riinvest\Pleurat%20Pula%20-%20Sfidat%20përgjatë%20Web%20Crawling%20të%20patundshmërive%20online%20në%20Kosovë%20(rast%20studimi%20Gjirafa,%20Merr%20Jep%20dhe%20Oferta%20Suksesi%20-%20Final%20Version%2019.12.2022.docx#_Toc122451573)

## **FJALORI DHE SHKURTESAT E TERMAVE**

IDE Integrated Development Environment

BOT Robot

CSV Comma Separated Value

JSON JavaScript Object Notation

DOM Document Object Model

HTML Hypertext Markup Language

XML Extensible Markup Language

FIFO First In First Out

VENV Virtual Environment

URL Uniform Resource Locator

MB Mega Byte

REP Robot Exclusion Protocol

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans

DDOS Distributed Denial of Service

CDN Content Delivery Network

SSL Secure Socket Layer

# **HYRJE**

Siç e dimë, në ditët e sotme, të dhënat luajnë një rol tejet të rëndësishëm në ekonominë botërore dhe janë një ndër faktorët më të rëndësishëm të suksesit të bizneseve, për shumë arsye. Disa nga këto arsye janë: të dhënat e përpunuara dhe të strukturuara mirë, mund të ndihmojnë bizneset si për të targetuar më lehtë klientët e tyre, po ashtu edhe për të analizuar më lehtë dhe më saktë kërkesat dhe preferencat e tyre ndaj produkteve dhe shërbimeve, për të bërë kërkime më të lehta në platformat e mëdha si “Google” apo “Bing”, targetimin dhe mbulimin më efikas të tregut, depërtimin më të lehtë në treg, marketingun digjital etj.

Në anën tjetër, është e njohur se motorët më të mëdhenj të kërkimit, ndër të tjerë janë: “Google” dhe “Bing”. Këta dy motorë kërkimi funksionojnë duke indeksuar të gjitha faqet në arkivat e tyre, në mënyrë që të kthejnë rezultate sa më relevante përmes kërkimeve (ang query-ve[[1]](#footnote-1)). Web Crawler-at janë ata që ua bëjnë të mundur motorëve të kërkimit që të përballojnë këtë process (Dilmegani, December 15, 2020).

Web Scrapy është një teknologji që bën automatizimin e procesit të nxjerrjes së informacionit nga ueb faqet, shpejtë dhe saktë. Të dhënat e nxjerra përmes kësaj teknologjie ofrohen në një format të strukturuar, gjë që e bënë më të lehtë analizmin dhe përdorimin e tyre (Kenny, n.d.).

Ekzistojnë shumë teknologji, librari dhe “frameworks” [[2]](#footnote-2) (shabllone) të ndryshëm që përdoren për kryerjen e procesit të scrap-imit. Mirëpo, përgjatë hulumtimit tonë do të përdoret një shabllon tejet i njohur si “Scrapy”, i cili është ndërtuar në gjuhën programuese Python dhe kodi do të zhvillohet në PyCharm Integrated Development Environment.

Fokusi kryesor i këtij hulumtimi, janë sfidat që hasen përgjatë krijimit të web scrap-ëve si dhe metodat dhe teknikat e ndryshme për tejkalimin e tyre, për arritjen e rezultatit final.

Në mënyrë që hulumtimi të jetë sa më relevant, si rast hulumtimi janë përzgjedhur tre ueb faqet më të mëdha ne Kosovë: [www.ofertasuksesi.com](http://www.ofertasuksesi.com); [www.gjirafa.com](http://www.gjirafa.com); [www.merrjep.com](http://www.merrjep.com), të cilat posedojnë të dhëna për patundshmëri, fushë kjo tejet e zhvilluar në ekonomi.

# **SHQYRTIMI I LITERATURËS**

Termet si “Web Scraping” dhe “Web Crawling”, shpeshherë përdoren në mënyrë të ndërsjellë për shkak të ngjashmërive mes tyre. Mirëpo, në anën tjetër, ekzistojnë dallime mes tyre të cilat i veçojnë ato mes vete. Që të dyja i referohen nxjerrjes së të dhënave nga ueb faqet, por dallojnë për nga mënyra e qasjes në to. “Web Scraping” fokusohet në një grup specifik të të dhënave, ndërkaq “Web Crawling” përshkon të gjitha faqet një nga një brenda një ueb faqeje (Perez, 2020).

# **Çfarë është Web Crawling?**

“Web Crawling” si term i referohet procesit të përdorimit të bot-ëve[[3]](#footnote-3) (spiders[[4]](#footnote-4)) për të lexuar dhe ruajtur të gjithë përmbajtjen e një faqeje për qëllime të arkivimit apo indeksimit të saj. Pra, motorët e kërkimit: si “Bing” apo “Google” përdorin “Web Crawling” për të nxjerrë të gjithë informacionin nga ueb faqja dhe për të bërë indeksimin e tij në motorin e kërkimit. Përmes këtij procesi, “Google” na tregon se cilat faqe kanë informacionin e kërkuar (Perez, 2020).

Crawler-ët veprojnë nëpër web faqe si eksploruesit në një tokë të re. Web crawling si term është tejet i përgjithshëm dhe përdoret si ombrellë e cila përfshinë të gjitha teknologjitë që kanë të bëjnë me shfletimin, nxjerrjen, përpunimin, filtrimin dhe strukturimin e të dhënave. Një web crawler, njihet si një bot-motor i kërkimit apo merimangë e faqes (website spider), së cilës mund t’i referohemi shkurt si “crawler” dhe nuk është asgjë më shumë se sa një bot i internetit që kërkon në mënyre sistematike në www (world wide web) dhe bën indeksimin e faqeve për motorin e kërkimit. (Shirey, n.d.)

# **Cfarë është Web Scraping?**

“Web Scraping” si term i referohet nxjerrjes së të dhënave nga ueb faqja dhe ruajtja e tyre në një format të dosjes (file) si p.sh. në një Excel faqe, CSV (Comma Seperated Value) apo JSON (JavaScript Object Notation) file. Një aspekt kyç i Web Scraping është nxjerrja e një grupi të të dhënave specifike për analiza të mëtejshme. Shembull, një kompani nxjerr të dhëna për një produkt specifik nga faqe të ndryshme, për të analizuar pozicionimin e produktit të saj karshi konkurrentëve në treg (Perez, 2020).

Në këtë periudhë të zhvillimit të hovshëm të teknologjisë, ekzistojnë shumë gjuhë programuese dhe teknologji (frameworks) të ndryshme që përdorën për ndërtimin e skriptave të ndryshme (web scraping) të cilat shërbejnë për qëllime të caktuara si: nxjerrjen/kopjimin, përpunimin dhe manipulimin e të dhënave nëpër web faqe të ndryshme.

Në këtë hulumtim, si gjuhë programuese për zhvillimin e skriptave (web scraping) do të përdorim gjuhën Python e cila është më e lehtë për t’u përdorur dhe e pasur me librari dhe shabllone të ndryshme. Libraritë dhe shabllonet më të pasura në Python që përdoren nga zhvilluesit e web scraping janë: Scrapy, Beautiful Soup dhe Selenium. Sidoqoftë, secila librari i ka benefitet dhe të metat e saj, asgjë nuk është perfekte në këtë botë (Palakollu, 2019).

# **Scrapy vs Beautiful Soup vs Selenium**

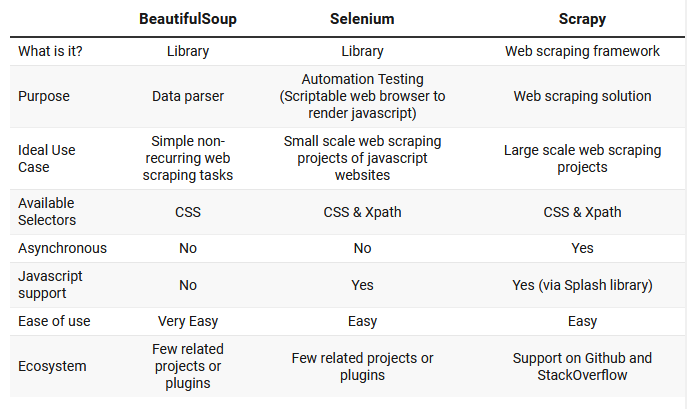
Që të tri teknologjitë të cilat do të trajtohen në vijim përdoren për qëllimin e njëjtë, mirëpo dallojnë mes vete për nga mënyra se si funksionojnë dhe i qasen problemit. Të gjitha këto teknologji kanë avantazhet dhe disavantazhet e tyre. Këto janë të përmbledhura si në figurën 1 më poshtë.

Figura 1 - Avantazhet dhe disavantazhet e teknologjive (Andrade, 2020)

# **Libraria Beautiful Soup**

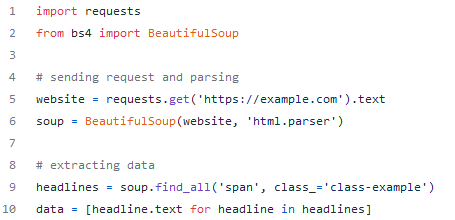
“Beautiful Soup” është opsioni më i thjeshtë për web scraping. Kjo librari, ndihmon zhvilluesin që të nxjerrë të dhënat në mënyrë të shpejtë nga një faqe e caktuar. “Beautiful Soup” bën nxjerrjen e të dhënave nga Hypertext Markup Language (HTML) dhe Extensible Markup Language (XML) fajllat, por problemi qëndron se libraria nuk mund ta realizojë të gjithë punën si e vetme. Kjo librari kërkon module specifike për të kryer punën me sukses. Për të dërguar kërkesa në server i nevojitet ndihma e një librarie tjetër si “Requests” ose “urlib2”. Ndërsa për të nxjerrë të dhënat i nevojitet një parser i jashtëm si: XML, lxml’s HTML apo HTML5lib parser (Andrade, 2020).

Figura 2 –Beautiful Soup dhe modulet e varura (Andrade, 2020)

# **Libraria Selenium**

“Selenium” si librari, edhe pse përdoret për të bërë scraping në ueb faqe, nuk është i dizajnuar për këtë qëllim. Në fakt, seleniumi është një ueb driver [[5]](#footnote-5) i dizajnuar për të renderuar faqet për testime të automatizuara të ueb aplikacioneve. Shumica e ueb faqeve përdorin “JavaScript” për të krijuar përmbajtje dinamike të ueb faqeve. Për këtë arsye “Selenium” është më i mirë në ueb scraping sepse punon shumë lehtë me DOM-in[[6]](#footnote-6) (Document Object Model) e “JavaScript”. Nuk është i lehtë për t’u mësuar, mirëpo është i lehtë për t’u përdorur sepse kodi i përshtatet sjelljeve njerëzore si: klikimi i butonit, selektimi i dropdown menuve, madhësia e faqes, shfletimi i faqeve, mbushja e formave, etj. Ua mundëson një qasje më të lehtë zhvilluesve që të shkruajnë teste në gjuhë të ndryshme programuese si: C#, Java, Python, Ruby etj. Një ndër disavantazhet e “Selenium”  
është shpejtësia (Andrade, 2020). Seleniumi ka një ekosistem shumë të mirë dhe të lehtë për mësim dhe zhvillim, mirëpo problemi qëndron se nuk mund t’i përdorim me lehtësi prox-it[[7]](#footnote-7). (Palakollu, 2019).

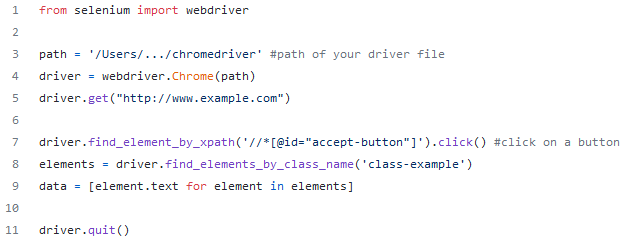


Figura 3 - Libraria Selenium (Andrade, 2020)

# **Shablloni Scrapy**

Scrapy është një “open source framework”[[8]](#footnote-8) që shërben për nxjerrjen e të dhënave nga ueb faqet. Një ndër avantazhet më të mëdha të përdorimit të Scrapy është shpejtësia dhe libraria që e mbështet mënyrën asinkrone[[9]](#footnote-9) të kërkesave e cila mundëson që të bëj “scrape”[[10]](#footnote-10) 960 faqe për minut (Rafik, 2021). Kjo nënkupton që scrapy spider nuk ka nevojë për të pritur çdo kërkesë një nga një, mirëpo dërgon shumë “HTTP” kërkesa në mënyrë paralele dhe i parson[[11]](#footnote-11) të dhënat që kthehen nga serveri. Performanca (shpejtësia e zvarritjes) e Scrapy-it ndaj faqeve lokale në maqinën tonë, është realizuar përmes komandës *bench,* si në figurën 7. Kjo metodë e rrit shpejtësinë dhe efikasitetin.

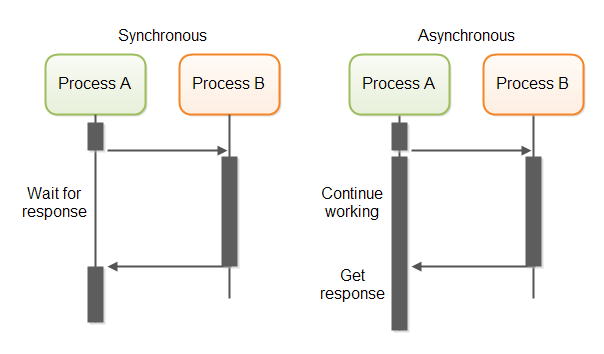
Një ndër të metat e scrapy është se nuk e përkrah javascript-in në mënyrë sikurse “Splash” dhe është pak më e vështirë për t’u mësuar dhe konfiguruar sesa “Beautiful Soup” (Andrade, 2020). Në këtë hulumtim, skriptat për qasjen në faqet e patundshmërive do të zhvillohen përmes “Scrapy Framework”.

Figura 4 – Procesi i kërkesave asinkrone (Andrade, 2020)

# **DEFINIMI I PROBLEMIT**

Duke konsideruar që Scrapy si teknologji për Web Scraping, kundrejt teknologjive tjera të ndryshme që përdoren për qëllimin e njëjtë, është më e përshtatshme në shumë aspekte, kemi vendosur që ta përdorim për ndërtimin e skriptave përgjatë hulumtimit. Gjithashtu vlen të theksohet se Scrapy na mundëson një qasje më të lehtë dhe më të shpejtë të problemit, pa pasur nevojë për ndihmën e moduleve tjera shtesë.

# **Komponentet kryesore të Scrapy**

“Scrapy” është shkruar në një framework event-driven në Python i njohur si “*Twisted*” dhe si teknologji përbëhet nga pesë komponentet kryesore: *Spiders*, *Item Pipelines*, *Middlewares*, *Engine*, *Scheduler* (Developers, 2022)*.* Secila nga këto teknologji kryen një funksion të veçantë për të cilat do të flitet në vijim.

# **Spiders**

“Spider”,është një komponentë e cila na mundëson që të definojmë se çfarë duam që të nxjerrim nga një faqe. Scrapy na ofron pesë “spiders” të ndryshëm të cilët janë të definuar si klasa të Python (Rafik, 2021). Këta “spiders” janë:

1. scrapy.Spider – për scraping të një grupi të caktuar të dhënash,
2. CrawlSpider – për scraping të të gjitha faqeve,
3. XMLFeedSpider – për scraping të XML fajllave,
4. CSVFeedSpider – për scraping të CSV fajllave dhe
5. SitemapSpider – për scraping të sitemaps

“Spiders” ndryshe njihen edhe si klasa të shkruara nga përdoruesit e Scrapy-t, që shërbejnë për të parsuar përgjigje (responses) dhe për të nxjerrë të dhënat nga to apo për të përcjellë kërkesa (requests) të tjera (Developers, 2022).

# **Item Pipeline**

“Item Pipeline”*,* është një komponentë e cila ka të bëjë me nxjerrjen e të dhënave. “Pipeline” na shërben gjithashtu për procesimin e të dhënave si: pastrimin e të dhënave, largimin e kopjeve dhe ruajtjen e të dhënave në ndonjë databazë të jashtme (Rafik, 2021).

Fokusi i hulumtimit tonë nuk do të jetë tek procesimi i të dhënave dhe ruajtja e tyre në ndonjë

databazë të jashtme. Për këtë arsye nuk do të përdorim “Item Pipeline”, por do të procesojmë disa

të dhëna si mostër studimi duke i ruajtur ato në dosje (files) të formatit JSON apo CSV.

# **Middleware**

“Middleware”*,* është një komponentë apo një shtresë ndërmjetësuese, që ka të bëjë gjithçka me kërkesat që i dërgojmë në një ueb faqe dhe përgjigjet që i pranojmë nga ueb faqja (Rafik, 2021). Pra, mund të kemi dy middleware të ndryshëm si:

a) Downloader Middleware dhe

b) Spider Middleware.

“Downloader Middleware” është një middleware në mes të komponenteve “Engine” dhe “Spider” që shërben për procesimin e kërkesave dhe përgjigjeve ndërmjet këtyre të dyjave, ndërsa në anën tjetër kemi “Spider Middleware” mes komponenteve “Engine” dhe “Downloader” që bëjnë procesimin e kërkesave dhe përgjigjeve në mes të po këtyre komponentave (Developers, 2022).

# **Engine**

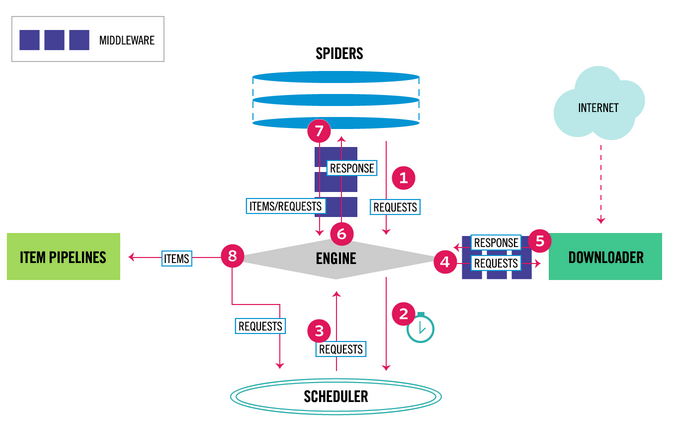
Komponenta Engine është përgjegjëse për kontrollimin dhe koordinimin e rrjedhjes së të dhënave në mes të gjitha komponenteve të sistemit. Nëse ndodhin vendime të caktuara, atëherë manifestohen evente të ndryshme të cilat i menaxhon Engine (Developers, 2022).

# **Scheduler**

Një ndër pesë komponentet e Scrapy është edhe Scheduler-i, i cili luan një rol tejet të rëndësishëm në tërë ciklin e procesit përgjatë Web Scraping. Scheduleri pranon të gjitha kërkesat nga “Engine”, duke i radhitur ato një pas një në pritje në mënyrë që ta shërbejë engine-in kurdo që lirohet, për të pranuar kërkesat e reja (Developers, 2022). Teknikisht, njihet si një strukturë e thjeshtë e të dhënave “Que”[[12]](#footnote-12) e cila përdor metodologjinë “First In First Out“ (FIFO) (Rafik, 2021).

# **Arkitektura dhe ndërveprimi i komponenteve të Scrapy-t**

Që të kemi një pasqyrë sa më të qartë se si funksionon ”Scrapy” dhe se si ndërveprojnë komponentet e saj përgjatë procesit, do të mirret si shembull faqja “OfertaSuksesi” e cila përmban postime për patundshmëri nga njerëz të rëndomtë. Le të supozojmë se veçse kemi ndërtuar një “spider” dhe ta ekzektuojmë që të nxjerrim të gjitha këto të dhëna nga ueb faqja.

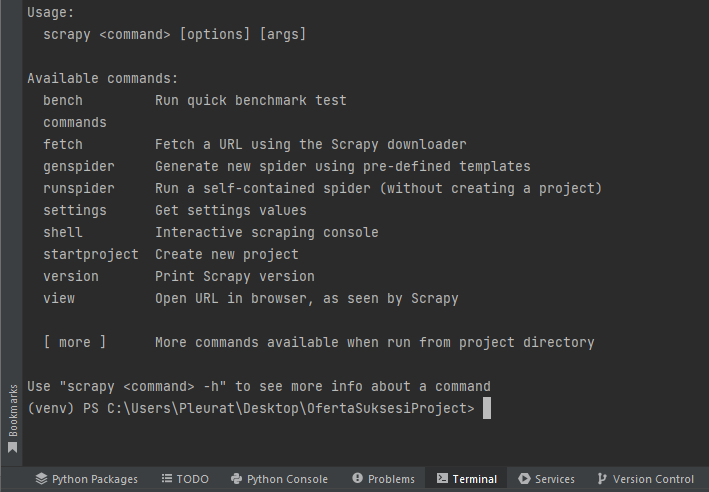
Në momentin e ekzekutimit të skriptës, spider do të dërgoj një kërkesë në faqe dhe i gjithë cikli i proceseve dhe rrjedhës së të dhënave, do të kontrollohet nga Engine si më poshtë. Referoju figurës dhe hapave të detajuar (Rafik, 2021).

**Figura 5 - Rrjedhja e proceseve dhe të dhënave (Developers, 2022)**

Rrjedhja e të dhënave do të vijojë si:

1. Engine pranon kërkesën e parë për crawling nga “Spider”.
2. Këtë kërkesë Engine e dërgon tek Scheduleri, për ta vendosur në radhë të pritjes dhe kërkon për një kërkesë tjetër e cila është në radhë nga Scheduleri, për të bërë crawling.
3. Scheduleri e shërben Enginin me kërkesën e radhës.
4. Engine e dërgon kërkesën tek Downloaderi përmes Downloader Middleware-it (process\_request()).
5. Kur faqja ta përfundojë shkarkimin, Downloaderi gjeneron një përgjigje të kësaj faqeje dhe e dërgon atë tek Engine, përmes Downloader Middleware-it (process\_response()).
6. Pas pranimit të përgjigjes nga Downloader, Engine e dergon atë tek Spider për procesim duke kaluar përmes Spider Middleware (process\_spider\_input()).
7. Spider bën procesimin e kërkesës dhe kthen scraping items si dhe kërkesa të reja tek Engine, përmes Spider Middleware (process\_spider\_output()).
8. Engine dërgon iteme-at e procesuar tek Item Pipelines, pastaj dërgon kërkesat e reja tek Scheduleri, për të pritur në radhë. Po ashtu, kërkon për kërkesën e radhës nga Scheduleri.

# **Komandat e Scrapy Framework**

Në platformën PyCharm do të instalojmë frameworks e Scrapy-it në Virtual Environment (Venv), për secilin projekt ndaras, në mënyrë që të mos interferojnë me njëra tjetrën. Pas instalimit, nëse shtypim komandën “scrapy”, në terminal do të na shfaqen të gjitha komandat e mundshme të cilat i ofron Scrapy (shih foton më poshtë).

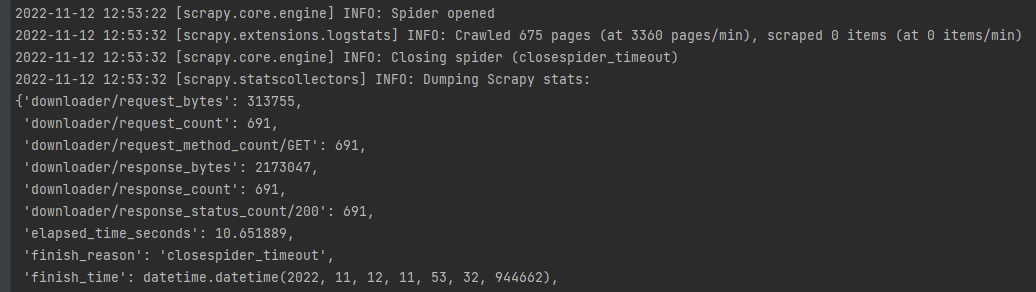
**Figura 6 - Komandat e shabllonit Scrapy**

Scrapy përdoret duke vendosur fjalën e rezervuar (sintaksën) “Scrapy” dhe duke ia bashkangjitur pas saj njërën nga komandat të cilat do të shtjellohet në vijim. Gjithashtu, pas komandës mund të bashkangjitet ndonjë alternativë tjetër dhe argument .

Komandat e Scrapy dhe përshkrimi i funksionit të tyre (Rafik, 2021):

* Bench - përdoret për të testuar se sa shpejt mund të performojë makina jonë, në dërgimin e kërkesave dhe shfrytëzimin e resurseve (Shiko figurën 7).
* Fetch – përdoret për nxjerrjen e HTML, nga ndonjë faqe që dëshirojmë.
* Genspider – bën gjenerimin e një spider bot-i, duke shfrytëzuar shabllonet e definuara.
* Runspider – e përdorim kur dëshirojmë të bëjmë testimin e ndonjë faqeje pa pasur nevojë të krijojmë projekt të ri.
* Settings – përdoret për vendosjen e vlerave përcaktuese sipas kërkesave të projektit që na nevojiten. Mirëpo, në fillim të çdo projekti na ofrohen disa vlera bazike si shabllon.
* Startproject – përdoret për krijimin e një projekti të ri.
* Version – tregon versionin e Scrapy Framework, i cili tek ne është v2.7.0.
* View – përdoret për hapjen e një URL në shfletues, ashtu siç e sheh Scrapy.

Nëse dëshirojmë të dimë më shumë detaje rreth një komande specifike, duhet të shtypim në terminal “scrapy <komandën> -h”.



**Figura 7 - Bench komanda - Përformanca e maqinës për kërkesa në minut**

# **Selektorët si mekanizma dhe përdorimi i tyre**

Pasi të dërgojmë kërkesë nëpër faqe përmes Scrapy-it, do të na duhet që të nxjerrim të dhëna specifike nga përgjigja e kthyer prapa. Për nxjerrjen e të dhënave përdorim mekanizmat e quajtur Selektorë (Selectors). Selektorët janë të ndërtuar sipas lxml librarisë, e cila është librari e Python dhe shërben për të na mundësuar menaxhimin sa më të lehtë të XML dhe HTML dosjeve, si dhe për përdorim të ueb scraping (Developers, 2022).

Kemi disa lloje të selektorëve, por më të përdorurat për web scraping janë:

1. XPath (XML Path Language) selektorët;
2. CSS (Cascade Style Sheets) selektorët dhe
3. RegEx (Regular Expressions) selektorët

# **XPath Selektorët**

Xpath selektorët në teori janë më të ngadalshëm, por nuk dallojnë shumë mes vete në të dhjetat e milisekondit të CSS selektorët. Në anën tjetër, Xpath ofron fleksibilitet më të madh, sidomos për strukturat e të dhënave komplekse. Xpath gjithashtu mund të bredhë në gjithë hierarkinë e HTML DOM përderisa CSS është më i limituar (Sulzenko, 2021).

# **CSS Selektorët**

Pothuajse çdo faqe përmban CSS, e cila përdoret për stilizimin e faqeve (p.sh. tekstit, fonteve etc.). CSS selektorët tregojnë gjithashtu se si elementet janë të selektuar. CSS selektorët mund të bredhin nga rrënja e dokumentit tek cilido element, mirëpo nuk mund të bredhin dy drejtime sikurse Xpath selektorët.

Ata njihen si selektorët më të shpejtë të metodave të nxjerrjes së të dhënave të faqeve. Janë lehtë për t’u mirëmbajtur dhe menaxhuar dhe mbështesin pothuajse çdo librari të Python për ueb scraping (Sulzenko, 2021).

# **RegEx Selektorët**

RegEx është metoda e fundit për nxjerrjen e të dhënave, e cila ndryshe njihet edhe si “shprehjet e rregullta”. Nuk është aq praktike sa dy metodat tjera për HTML, për shkak se kërkon nëpër stringje[[13]](#footnote-13), por njihet si metodë tejet e mirë për raste specifike.

Përdoret zakonisht në rastet kur HTML dosjet apo elementet kanë përmbajtje shumë të ngarkuar. RegEx-at punojnë duke u bazuar në shabllonet (patterns) të cilat i krijojmë specifikisht për të gjetur përmbajtjen e duhur nga përgjigja (Sulzenko, 2021). Rasti i studimit do të punohet vetëm me këtë metodë në të tri projektet.

# **Sfidat përgjatë Web Scraping dhe tejkalimi i tyre**

Zhvillimi i teknologjisë dhe digjitalizimi i bizneseve përpos që kanë një trend tejet të madh të rritjes, janë duke ndikuar gjithashtu edhe në krijimin e një numri të madh të të dhënave. Për momentin në botë, për çdo person brenda një sekondi janë duke u krijuar 1.7 MB (Mega Byte) të dhëna, të cilat shfrytëzohen nga bizneset. Kjo rritje eksponenciale e vëllimit të të dhënave po ndikon gjithashtu edhe në rritjen e tregut për Web Scraping të ueb faqeve, e cila u vlerësua në vitin 2020 me rreth 1.7 miliardë dollarë dhe parashikohet që deri në vitin 2027 të arrijë rreth 24 miliardë dollarë. Sidoqoftë, web scraping përballet me sfida të ndryshme, për arsye se qeveritë dhe kirjuesit e të dhënave vendosin pengesa të reja ligjore dhe teknike për të siguruar privatësinë e informacionit të tyre (DIlmegani, 2021).

Ky hulumtim, do të zhytet më thellësisht në disa nga sfidat me të cilat përballemi më shpesh përgjatë ueb crawling dhe gjetjen e praktikave më të mira për t’u përballur me to.

# **Pengesat Ligjore**

Përpos pengesave teknike ekzistojnë edhe ato ligjore. Barrierat ligjore hyjnë si masë mbrojtëse ndaj shqetësimeve etike dhe të privatësisë në lidhje me nxjerrjen e të dhënave (Rivera, 2022).

Web Scraping nuk është veprim i paligjshëm, nëse të dhënat e nxjerra nuk përdoren për qëllime joetike. Në shumë raste ligjore, kur bizneset kanë përdorur ueb crawlers për të nxjerrë të dhënat publike të konkurrencës, gjyqtarët nuk kanë gjetur ndonjë arsye legjitime që t’i kundërvihen zvarritësve (scrapers) si në rastin “eBay vs Bidder’s Edge”.

Sidoqoftë, nëse përdorimi i të dhënave të nxjerra shkakton direkt apo indirekt shkelje të së drejtës së autorit, atëherë ueb scraping konsiderohet si i paligjshëm, ngjashëm si në rastin “Facebook vs Power Ventures”.

Edhe pse procesi është i ligjshëm, administratorët e ueb faqeve shmangin zvarritësit për disa arsye:

1. Shumicën e rasteve nuk i respektojnë rregullat e vendosura nga faqja;
2. Ngarkojnë domenin duke nxjerrë të dhëna në masë të madhe;
3. Mbledhin informacione që kanë të bëjnë me privatësinë e përdoruesit.

Për këto aryse, askush nuk dëshiron që bizneset ose individë të tjerë të përfitojnë një avantazh konkurrues (Rivera, 2022).

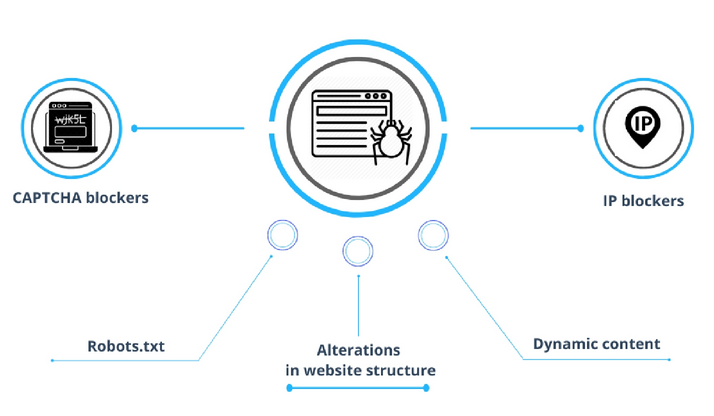
Si përfundim, ueb scraping dhe përdorimi i softuerëve për ueb scraping nuk janë të paligjshëm, por janë rregulluar në dhjetë vitet e fundit përmes rregulloreve të privatësisë të cilat përkufizojnë shkallën e zvarritjes për një faqe. Këto njihen si “Rregulloret e Përgjithshme të Mbrojtjes së të Dhënave” (GDPR) (DIlmegani, 2021).

# **Pengesat teknike**

Përveç çështjeve ligjore, ekzistojnë shumë sfida të ndryshme teknike me të cilat ballafaqohen ueb scrapers për shkak të barrierave të vendosura nga pronarët e të dhënave, me qëllimin për të kufizuar qasjet jonjerëzore (botëve) në informacionin e tyre.

Disa nga llojet e këtyre pengesave janë:

* Robot.txt;
* Rate Limiting dhe bllokuesit e IP-ve;
* Bypass Cloudflare;
* Ndryshimet në strukturën ueb faqeve;
* Bllokuesit CAPTCHA;
* Kontenti Dinamik;
* Honeypots;
* Login Requirements;
* Slow or unstable load speed.



**Figura 8 – Sfidat e Web Scraping (Dilmegani, December 15, 2020)**

# **Menaxhimi i qasjes së botëve përmes dosjes Robots.txt**

Web faqet janë të lira që të vendosin nëse duan apo jo, të lejojnë ueb scrapers botët në ueb faqe për qëllime të nxjerrjes së të dhënave. Disa faqe nuk lejojnë ueb scraping sepse duan që të mbrojnë të dhënat e tyre nga konkurrentët apo të mbrojnë shfrytëzimin e resurseve të faqes, sepse botët mund të ndikojnë në performancën e faqes.

Robots.txt është një dosje (file) që e krijojnë krijuesit e faqes, për të udhëzuar bot-ët e kërkimit se si të bëjnë crawling në faqet e tyre. Kjo dosje (file) është pjesë e Robots Exclusion Protocol (REP) i cili është një grup i standardeve të faqeve që bën rregullimin se si duhet të sillen botët, si të qasen, si të bëjnë indeksimin dhe shërbimin e kontentit tek përdoruesit.

Në praktikë, dosja robots.txt tregon se çfarë user-agents munden apo jo të bëjnë crawl pjesë të ueb faqeve. Instruksionet e crawling janë të specifikuar sipas emrit “disallowing” ose “allowing”[[14]](#footnote-14) (Connell, 2021). Për më shumë, në kapitullin 5 shpjegohet mënyra e trajtimit të kësaj pengese, gjatë kryerjes së hulumtimit.

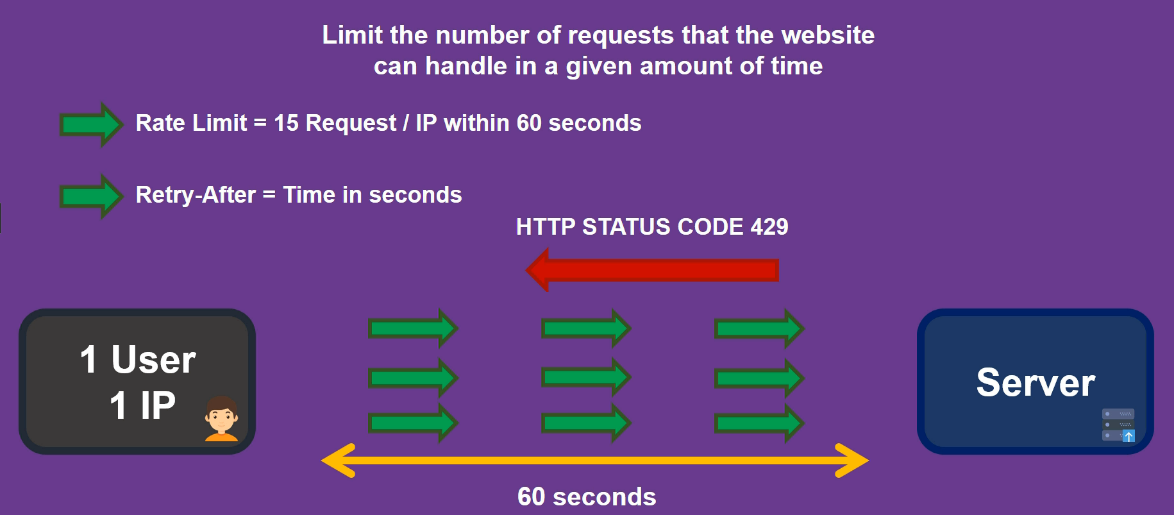
# **Rate Limitting Requests dhe bllokuesit e IP-ve**

Rate Limitting Requests ndryshe njihet si “Norma Kufizuese e Kërkesave” e cila është tejet e njohur për të luftuar scrapers. Kjo teknikë funksionon duke u bazuar në ueb faqen e cila imponon një numër të kufizuar veprimesh që një përdorues mund të kryej nga një IP adresë e vetme. Kufijtë mund të ndryshojnë varësisht nga ueb faqja dhe bazohen në:

1. Numrin e operacioneve të kryera brenda një periudhe të caktuar kohore ose,
2. Sasinë e të dhënave që e përdorim. (Rivera, 2022)

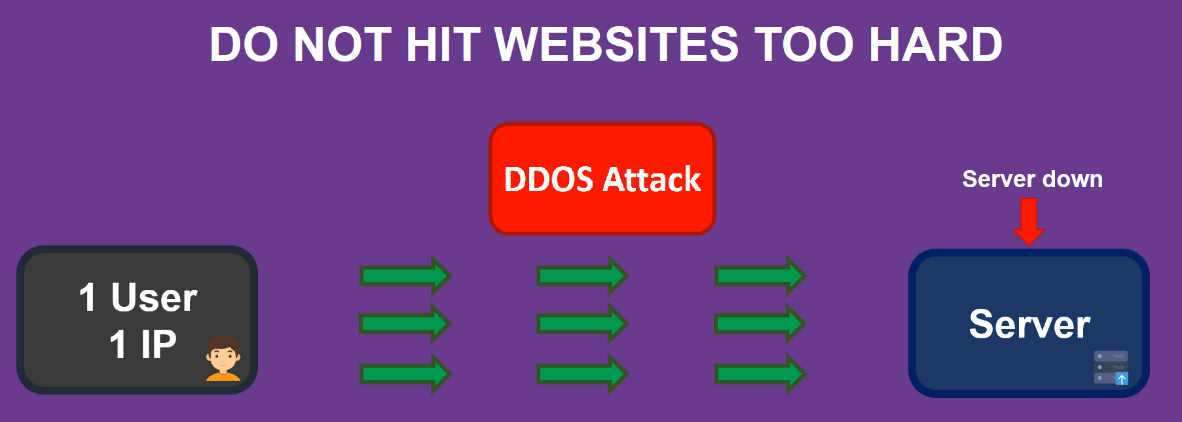
Për të demonstruar se si funksionon kjo teknikë, do të marrim shembull një ueb faqe e cila ka vendosur normën kufizuese e cila pranon 15 kërkesa nga një IP e vetme brenda një periudhe prej 60 sekondash. Kjo nënkupton që nëse një përdorues nga një IP adresë dërgon më shumë se 15 kërkesa në server do të bëhet “ban” (përjashtohet) për t’u qasur në ueb faqe për një kohë të caktuar. Përjashtimet mund të jenë ndalesa të përkohshme apo ndalesa të përhershme.

Kur përjashtohet përkohësisht një përdorues, statusi si përgjigje që kthehet është “429 Too Many Requests“ që nënkupton numër i madh kërkesash dhe injektohet një vlerë e re në përgjigjen e njohur si “Retry-After: 300” që do të thotë se mund të provojmë të qasemi në faqe pas një periudhe ndëshkimi prej 300 sekonda (Rafik, 2021).



**Figura 9 - Norma e kufizimit të kërkesave (Rafik, 2021)**

Një rregull që duhet t’i përmbahemi gjithmonë, është që të mos lejojmë botin tonë që të dërgojë volum të madh dhe të pakontrolluar në ueb server për shkak se jo të gjithë servers mund të përballojnë një volum të tillë të kërkesave. Nëse ndodh një gjë e tillë, atëherë ueb server-i mund të mos funksionojë si pasojë e volumit të madh të trafikut dhe konsumimit të madh të resurseve. Teknikisht në këso rastesh, ueb scraping mund të shihet si “Distributed Denial of Service Attack” (DDOS Attack) (Rafik, 2021). Kjo mund të dërgojë deri tek bllokimi i përhershëm i IP adresës.



**Figura 10 – Konsiderimi i kërkesave të shumta si DDOS sulm (Rafik, 2021)**

Nëse bëhemi përjashtohemi në mënyrë permanente, atëherë mund të përdorim disa metoda për të tejkaluar këtë:

1. Restartimi i Router-it – mundëson rigjenerimin e një IP adrese të re;
2. Shërbime të Prox-it – mundëson ndërrimin e IP adresave;
3. Crawlera Proxy – një prej proksive më eficient por që kushton shumë.

Për të tejkaluar normën e kufizimit të kërkesave dhe përjashtimet e përhershme të IP-ve, duhet të përdorim një proxy shërbim që na mundëson ndërrimin e IP adresave në mënyrë automatike kohë pas kohe. Scrapy na mundëson që një shërbim të tillë ta përfshijmë në spider (Singhal, 2020).

Ekzistojnë lloje të ndryshme të IP proxy shërbimeve, duke filluar nga ato të thjeshta që mund të ofrohen falas deri tek ato më të sofistikuarat për të cilat duhet paguar për t’i përdorur. Po ashtu, edhe kjo pengesë është hasur gjatë hulumtimit, për të cilën flitet më shumë në kapitullin 5.

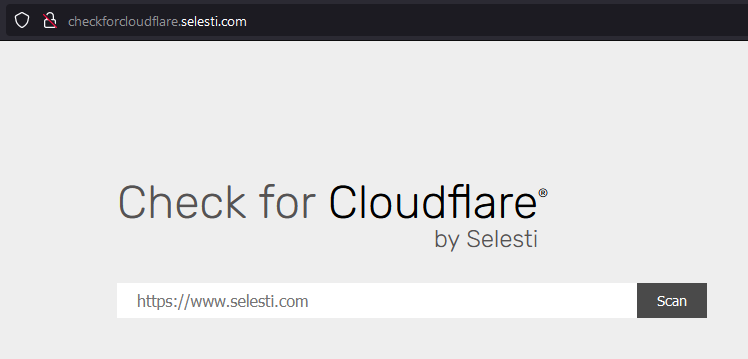
# **Bypass Cloudflare**

Cloudflare është një shërbim që shërben për mbrojtje, e cila ndryshe njihet edhe si “Content Delivery Network” (CDN) dhe përdoret nga kompanitë apo ueb faqet për të mbrojtur kontentin e tyre nga spiders dhe botët (Rafik, 2021).

Menaxhimi i Botëve Cloudflare është dizajnuar në atë mënyrë për të bllokuar nxjerrjen e përmbatjes nga scrapy dhe botët e tjerë që shërbejnë për lloje të tjera të trafikut me qëllim të keq. Menaxhimi i botëve Cloudflare mund të identifikojë botët në bazë të paternave të sjelljes së tyre (Cloudflare, 2022).

Ky shërbim funksionon duke luajtur rolin e një observuesi të kërkesave që vijnë. Nëse boti jonë dërgon shumë kërkesa për një periudhë të shkurtër kohore, atëherë to ta bëj largimin e tij dhe do të na bllokojë për t’iu qasur ueb faqes.

Për të ditur nëse faqet të cilat do t’i bëjmë scrapy gjatë hulumtimit përdorin mbrojtje CloudFlare apo jo, duhet që të shkojmë tek ueb faqja <http://checkforcloudflare.selesti.com/> duke vendosur në URL- të e faqeve për t’i skanuar. Ani pse është hasur në këtë pengesë në skriptën e “Merr Jep”, është përdorur një mënyrë alternative e qasjes dhe nuk ka pasur nevojë të intervenohet në të.

****

**Figura 11 - Skanimi i ueb faqeve për kontroll të mbrojtjes nga Cloudflare**

# **Ndryshimet në strukturën e ueb faqeve**

Crawler-ët e ueb faqeve janë të programuar për të bërë crawl një ueb faqe duke u bazuar në elementet e JavaSript dhe HTML që përmban faqja. Këto elemente shpesh mund të ndryshohen nga dizajnuesit e ueb faqeve për të optimizuar pamjen e faqes dhe për ta bërë atë më atraktive për përdoruesit. Momentin që ndryshimet ndodhin në elemente, botët që bëjnë crawling nuk do të jenë në gjendje të nxjerrin të dhëna të sakta dhe për këtë arsye duhet që të ndryshojmë kodin e skriptave, sadoherë që ka ndryshime të ueb faqes duke iu përshtatur atyre (DIlmegani, 2021).

Për të anashkaluar këtë pengesë, mund të ndërtojmë një parser që përdor mësimin e makinerisë për t’iu përshtatur ndryshimeve strukturale të ueb faqes, por kjo është tejet e vështirë për t’u arritur. Mënyra më e mirë për ndryshimet strukturale është që të mirëmbajmë dhe përditësojmë skriptën bazuar në ndryshime (Rivera, 2022). Sidoqoftë, gjatë këtij hulumtimi nuk është hasur në një pengesë të tillë.

# **Bllokuesit CAPTCHA**

CAPTCHA është një masë parandaluese e sigurisë së ueb faqeve që është krijuar për të identifikuar botët dhe për t’i parandaluar ata që t’iu qasen ueb faqeve (DIlmegani, 2021).

Qëllimi kryesor i CAPTCHA është që të ndajë njerëzit nga botët, duke shfaqur fotografi, probleme logjike, enigma etj. për të cilat njerëzit mund të gjejnë lehtë zgjidhjet e tyre, ndërsa botët do ta kenë tepër të vështirë. CAPTCHA konsiderohet si një ndër barrierat më të rënda për ueb crawler-ët.

CAPTCHA mund të jetë “hard-coded” në HTML dhe nëse hasim në këso lloj rasti testet do të shfaqen nëpër faqe të ndryshme herë pas here si një formë regjistrimi apo kontrolle për çfarëdo veprimi që bëjmë.

Për të zgjidhur këtë problem, web scrapers duhet të implementojnë një zgjidhje për të tejkaluar testet e shfaqura nga CAPTCHA, që të mund të arrijë bot-i të qaset (DIlmegani, 2021). Një mënyrë tjetër mund të jetë duke provuar ndërrimin e IP adresave dhe duke rifreskuar faqen përmes ndonjë Proxy. (Rivera, 2022)

Megjithëse përbën një nga pengesat më sfiduese, gjatë këtij hulumtimi nuk është hasur në një të tillë.

# **Përmbajtja dinamike**

Kontenti Dinamik është një pengesë tjetër që ndikon tek ueb crawler-ët por nuk do të thotë që është aplikuar si masë parandaluese për ta, sepse në prapavijë ka një qëllim tjetër.

Kontenti dinamik është element i ueb faqes i cili ndryshon sipas sjelljes së përdoruesit dhe të dhënave që ai përdor. Shembull, përdoruesit e “Netflix” ndjekin preferencat dhe të dhënat e përdoruesve në mënyrë që të bëjnë rekomandime personale sipas sjelljes së tyre.

Pra, të dhënat në ueb faqe nuk janë gjithmonë të strukturuara njëjtë dhe për këtë arsye vështirësohet qasja e ueb crawling botëve të cilët janë programuar për të bërë scrape faqet statike. Sidoqoftë, crawling botët mund të programohen për të përshkuar të gjithë faqen që të gjejnë të dhënat e targetuara dhe nxjerrjen e tyre. (DIlmegani, 2021)

Megjithatë, as kjo nuk ka përbërë pengesë gjatë periudhës së shkurtër të hulumtimit.

# **Honey Pots**

Honey Pots janë sisteme kompjuterike të krijuara për të penguar hakerët dhe për t’i bllokuar ata që t’iu qasen ueb faqeve. Një kurthë Honey Pot paraqitet si një pjesë legjitime e ueb faqes dhe përmban të dhëna të cilat një sulmues mund t’i targetojë. (DIlmegani, 2021)

Kurthat mund të jenë linqe të cilat janë të fshehura për njerëzit, por jo për scrapers. Në momentin që një scraper prek linkun bie në kurthë, ueb faqja mund të përdorë informacionin që e merr si p.sh. IP adresën dhe në bazë të saj e bllokon atë.

Për t’i ikur kurthës Honey Pot duhet që të sigurohemi që gjatë ueb scraping, linqet në CSS të cilat kanë atributet, si: “*display:none*” ose “*visibillity:hidden*” të mënjanohen. As kjo pengesë nuk është paraqitur gjatë hulumtimit, mirëpo përbën një nga pengesat më të shpeshta.



**Figura 12 - Një shembull i linkut të përdorur si honeypot (Rafik, 2021)**

# **Login Requirements**

Për të arritur deri tek nxjerrja e të dhënave nga disa ueb faqe na kërkohet që të identifikohemi. Pasi të vendosim kredencialet e hyrjes, shfletuesi automatikisht na bashkangjet vlerën e *cookie*-ve tek shumë kërkesa që i bëjmë përgjatë shfletimit dhe në këtë mënyrë ueb faqja e di që jemi personi i njëjtë që është qasur në faqe më herët.

Kjo paraqitet si një barrierë përgjatë procesit të scraping sepse mund të mos na kthejë të dhënat e duhura ose ta bëjë konfuz botin, duke kthyer përgjigje të ndryshme.

Për të mënjanuar këtë barrierë, duhet të sigurohemi që në skriptë, kur dërgon kërkesat, bashkangjiten vlerat e cooki-ve dhe dërgohen së bashku.

Për disa faqe, mund të përdorim metodën POST me kredencialet e përdoruesit dhe t’i ruajmë ato në një cookie. Nëse ndonjë gjë shkon keq përgjatë verifikimit të identitetit të POST kërkesës, mund të pranojmë një HTTP status me kodet 401, 403 etj. në vend të 200 (Singhal, 2020). Sidoqoftë, në asnjërën nga faqet e analizuara, nuk është kërkuar krijimi i një llogarie për qasje.

# **Slow Loading Speed**

Kur një ueb faqe pranon një numër të madh të kërkesave brenda një periudhe të shkurtër kohore, shpejtësia e hapjes së faqes mund të ngadalësohet ose të bëhet jo stabile. Në disa raste kërkesa mund të skadojë. Nga ana e përdoruesit kjo mund të rregullohet nëse bëjmë rifreskimin e faqes, mirëpo nga ana e scraper nuk funksionon sepse bot-i do ta ndalojë punën dhe nuk do të dijë se si të reagojë.

Për të tejkaluar këtë pengesë, duhet që në kod të vendosim ndonjë funksion që mundëson ringarkimin e faqes, të cilën e bën në mënyrë automatike nëse vëren që ueb faqja ka dështuar në kthimin e përgjigjes. (Rivera, 2022)

Edhe përgjatë këtij hulumtimi, është hasur në një pengesë të tillë, mirëpo për më shumë duhet t’i referoheni kapitullit 5.

# **METODOLOGJIA DHE VEGLAT**

Veglat të cilat mund të përdoren për zhvillimin e skriptave janë të shumta, mirëpo në këtë punim janë përdorur disa nga to, si: *Python*, *PyCharm*, *Scrapy*, *CSV* dhe *middleware* të ndryshëm sipas nevojës.

* *Python* është një gjuhë programimi e interpretuar, një gjuhë e nivelit të lartë programuese me semantikë dinamike dhe e orientuar në objekte. Sintaksa e Python është e thjeshtë dhe e lehtë për t’u mësuar, lexuar dhe zbatuar. Python mbështet module dhe paketa të ndryshme që mundësojnë programimin modular dhe ripërdorimin e kodit, si dhe debugimin[[15]](#footnote-15) (korigjimin), i cili është tejet i lehtë për t’u përdorur.
* *PyCharm* është një IDE mjedis i dedikuar për Python i cili ofron një gamë të gjerë të mjeteve thelbësore për zhvilluesit e Python. Këto mjete janë të integruara dhe ndihmojnë për të krijuar një mjedis të përshtatshëm për zhvillim produktiv në Python, Web dhe në shkencë të të dhënave. Python mbështet versionet: (Bill Tucker, Mukul Mantosh, Nafiul Islam, Paul Everitt, 2022):
* Python 2: verzioni 2.7 dhe
* Python 3: nga verzioni 3.6 deri tek versioni 3.11
* *Scrapy* është një shabllon i nivelit të lartë për ueb crawling dhe ueb scraping, i cili përdoret për zvarritjen e ueb faqeve dhe për nxjerrjen e të dhënave të strukturuara nga ueb faqet. Scrapy përdoret për qëllime të ndryshme, duke filluar nga nxjerrja e të dhënave e deri tek monitorimi dhe testimi i automatizuar. Burimi i kodit të Scrapy-t është i hapur dhe është shkruar në Python si dhe zhvilluar në Cambuslang. Aktualisht mirëmbahet nga një kompani zhvillimi dhe shërbimesh për scraping në internet, e njohur si Zyte (ish Scrapinghub) (Developers, 2022).
* *CSV* si term nënkupton “vlera të ndara me presje”, pra është një tekst file që shërben për ruajtjen e të dhënave të cilat ndahen në bazë të presjeve dhe si file mund të hapet në programe të ndryshme. File-at CSV përdoren për të lëvizur të dhënat midis programeve që nuk mund të shkëmbejnë të dhëna dhe më së shpeshti i hasim në tabela dhe baza të të dhënave (Johnson, 2022).

# **ANALIZA E KËRKESAVE, IMPLEMENTIMI DHE REZULTATET ME OBSERVIME**

Në vijim, do të ndërtojmë nga fillimi deri në fund web scraping skriptat për 3 faqet e patundshmërive duke bërë krahasimin mes tyre me fokus tek gjetja dhe tejkalimin e barrierave që na paraqiten. Gjatë zhvillimit të skriptave, do të spjegojmë çdo hap përmes shembujve të marrur nga kodi si dhe praktikat më të mira për tejkalimin e vështirësive.

# **Analiza e strukturës së faqeve**

Para se të fillojmë zhvillimin e skriptave dhe nxjerrjen e të dhënave, duhet që së pari të bëjmë analizën e ueb faqeve gjirafa, merrjep dhe ofertasuksesi, problemeve të mundshme që mund të hasim në to dhe nivelin e sigurisë së faqeve.

Analiza e ueb faqeve bëhet nga perspektiva e përdoruesit dhe është shumë e rëndësishme për shkak se jo të gjitha faqet janë të ndërtuara nga teknologji të njëjta dhe kanë strukturë të njëjtë. Disa ueb faqe janë tejet të thjeshta dhe kanë mbrojtje më të dobët, si në rastin tonë faqja OfertaSuksesi, ndërsa në anën tjetër kemi ueb faqet që përdorin teknologji më të avancuara dhe kanë mbrojtje më të fortë si: Gjirafa50 dhe MerrJep.

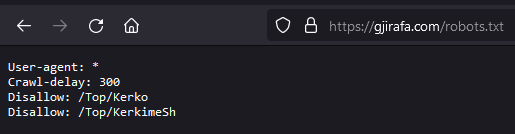
Përpos nivelit të strukturës dhe teknologjive, mund të hasim edhe probleme të ndryshme si vështirësia e qasjes në ueb faqe, siguria e mbrojtjes nga jashtë, enkriptimi i të dhënave, masat mbrojtëse të avancuara si Recaptcha etj. Pas një analize të gjatë të faqeve MerrJep, Gjirafa dhe OfertaSuksesi, kemi arritur të identifikojmë teknologjitë me të cilat janë ndërtuar dhe strukturën e kodit të tyre. Përpos anës teknike kemi bërë edhe identifikimin e problemeve dhe vështirësive të ndryshme.

Pra, analiza na ndihmon që të zbërthejmë problemet e ueb faqeve të cilat duam t’iu nënshtrohen procesit të crawl, të zgjedhim mënyrën më të mirë dhe më të lehtë për t’iu qasur atyre dhe teknologjitë që duhen përdorur për ta arritur rezultatin e duhur.

# **Kontrolloimi i ueb faqeve për robots.txt**

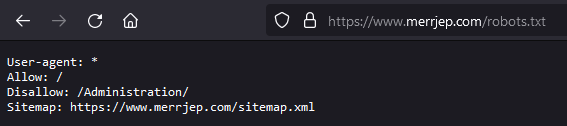
Për të parë nëse faqet posedojnë apo jo robots.txt file, do të marrim shembull faqet e patundshmërive: Gjirafa, Ofertasuksesi dhe Merrjep. Së pari duhet të shkruhet rrënja e domenit dhe pastaj të shtohet shkurtesa “/robots.txt” si p.sh. <https://gjirafa.com/robots.txt> dhe do të shfaqen instruksionet për ato faqe.

1. **Gjirafa** – mund të shohim se lejohet të përdoret çfarëdo user-agenti sipas shenjës “\*”, mirëpo kemi një instruksion “Crawl-delay: 300” që na lë të kuptojmë që gjatë crawling duhet të respektojmë shpejtësinë e paracaktuar, në mënyrë që të mos ndëshkohemi. Instruksionet “Disallow: /Top/Kerko” dhe “Disallow: /Top/Kerkimesh” janë të ndaluara për t’u përdorur në botët tonë.



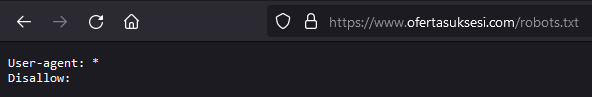
**Figura 13 - Robots.txt për www.gjirafa.com**

1. **MerrJep** - gjithashtu ngjashëm si Gjirafa lejon të përdoren të gjithë user-agjentët ndërsa na ndalon të bëjmë crawling në kategorinë “/Administration/”. Më poshtë gjendet një instruksion “Sitemap: <https://www.merrjep.com/sitemap.xml>” i cili tregon se është një XML file që përmban listën e të gjitha faqeve që na lejon t’iu qasemi me botët tanë.



**Figura 14 - Robots.txt për www.merrjep.com**

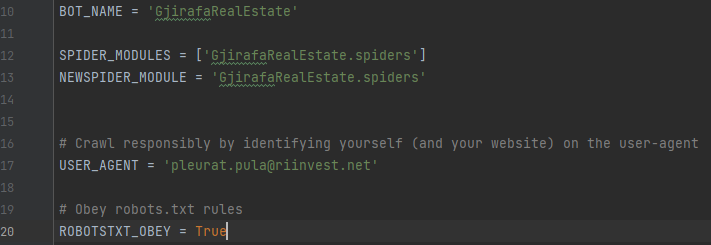
1. **OfertaSuksesi** - për dallim nga dy faqet e mësipërme, nuk ka instruksione.



**Figura 15 - Robots.txt për www.ofertasuksesi.com**

Pra, kur bëhet crawling bot-i do t’iu përshtatet rregullave që paraqiten në robots.txt file dhe nuk do të arrijë zonat e ndaluara. Ekzistojnë disa teknika të cilat na mundësojnë arritjen e qëllimit tonë, pa ndjekur rregullat e robot files:

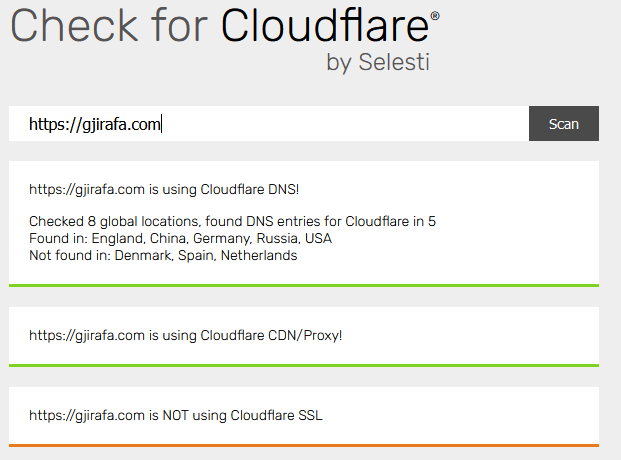
* Një ndër teknikat për të bërë crawling nëpër zonat e ndaluara në mënyrë legjitime është që të marrim leje duke kontaktuar pronarin e të dhënave dhe duke i ofruar një arsye të fortë për të bërë crawling në faqe (DIlmegani, 2021);
* Gjithashtu, ekziston edhe mundësia për të anashkaluar rregullat e vendosura nga robot files të cilat i ndjek bot-i përgjatë crawling. Kjo teknikë aplikohet duke ndryshuar në settings të projektit rregullin nga vlera “ROBOTSTXT\_OBEY = True” në “ROBOTSTXT\_OBEY = False” e cila ia mundëson botëve të injorojnë rregullat e faqes (Rafik, 2021).
* Teknika e fundit, është ofrimi i një email-i kontaktues në agjentin e përdoruesit, në mënyrë që nëse ndonjëherë kompanitë kanë problem me crawlering tonë të na kontaktojnë prapa. Kjo arrihet duke ndryshuar vlerën në settings të “USER\_AGENT = ‘pleurat.pula@rrinvest.net” (Rafik, 2021). Kjo teknikë nuk preferohet shumë nga krijuesit e botë-ve.



**Figura 16 - Rregullat që ndjekin botët përgjatë crawling**

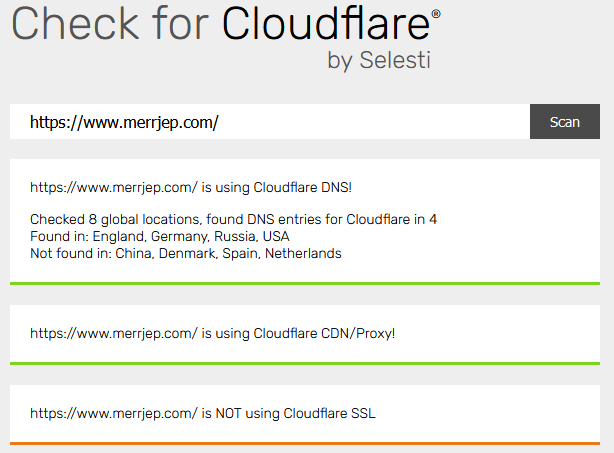
# **Kontrollimi i ueb faqeve për Cloudflare?**

1. **Gjirafa** – me skanimin e ueb faqes së Gjirafës, shohim se janë kontrolluar tetë regjionet globale. Ndër to janë gjetur të dhëna DNS për cloudflare në pesë regjione si: Angli, Francë, Kinë, Gjermani, Rusi dhe SHBA ndërsa nuk janë gjetur në Danimarkë, Spanjë dhe Holandë. Gjithashtu, ueb faqja është duke përdorur Cloudflare CDN/Proxy ndërsa nuk është duke përdorur Cloudflare Secure Socket Layer (SSL).



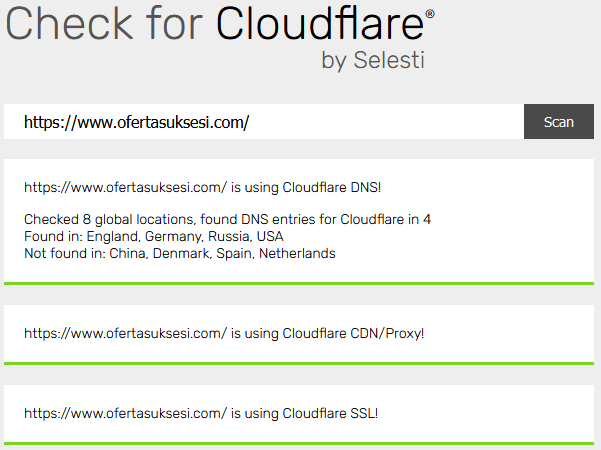
**Figura 17 - Skanimi i ueb faqes www.gjirafa.com për kontroll të Cloudflare**

1. **MerrJep –** në foton më poshtë të skanimit të ueb faqes, mund të verehet se është e ngjashme me atë të Gjirafës, por me një ndryshim të vogël tek regjionet ku nuk gjindet ajo e Kinës.



**Figura 18 - Skanimi i ueb faqes www.merrjep.com për kontroll të Cloudflare**

1. **OfertaSuksesi –** skanimi i kësaj ueb faqe është i njëjtë si me dy të tjerat më lartë, mirëpo kjo faqe ka edhe një shtresë mbrojtëse të njohur si “Cloud SSL” që e bën më të sigurtë.



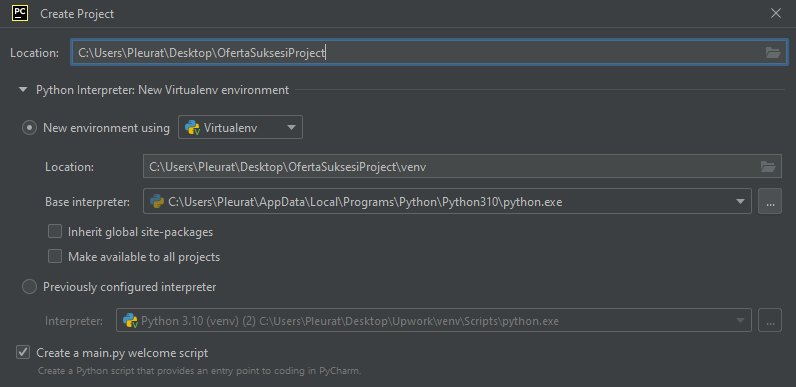
**Figura 19 - Skanimi i ueb faqes www.ofertasuksesi.com për kontroll të Cloudflare**

# **Implementimi**

# **Ambienti punues dhe instalimi i shabllonit dhe librarive**

Së pari, duhet të përgatisim vendin e punës ku do të zhvillojmë projektet. Mjeti kryesor që do të instalojmë është PyCharm IDE 2022.1.3 (Community Edition) për sistemin operativ Windows, i cili na nevojitet për zhvillimin e skriptave në vijim. Si gjuhë programuese do të instalojmë Python v3.10. Pasi t’i instalojmë këto, hapim PyCharm IDE dhe krijojmë një projekt të ri si në foto.

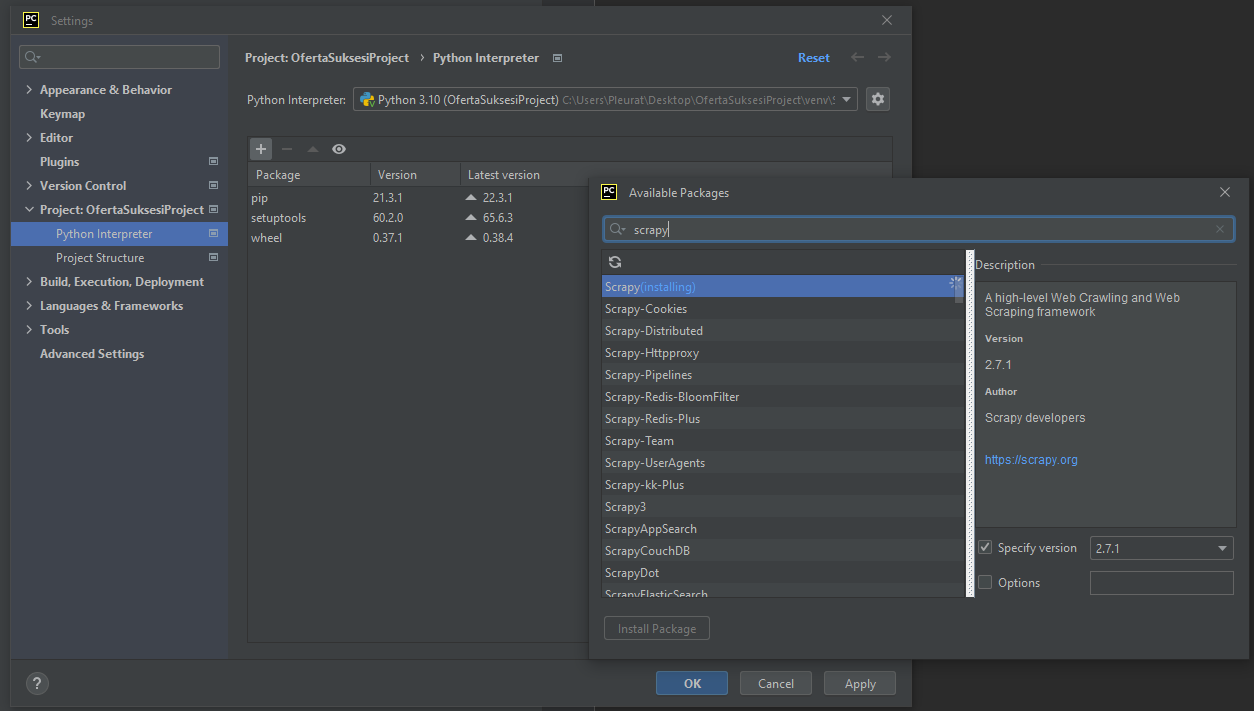
**Figura 20 - Krijimi i një projekti dhe mjedisi virtual në PyCharm**



Te rubrika *Location* përcaktojmë vendin për krijim dhe emërtim të projektit, që në rastin tonë, është “*OfertaSuksesiProject*”. Më poshtë zgjedhim mjedisin *Virtualenv* dhe vendin se ku do të krijohet mjedisi virtual dhe me cilin interpretues do të punojë.

***Virtualenv*** *–* është një shkurtesë që nënkupton “mjedis virtual” dhe konsiderohet si një njësi e izoluar përbrenda projektit tonë. Mjedisi virtual luan një rol të rëndësishëm në hulumtimin tonë sepse na ndihmon që ta izolojmë ambientin punues për secilin projekt veçmas. Nëse instalojmë librari apo shabllone të ndryshme në projektin tonë *OfertaSuksesi*, atëherë projektet e tjera si *Gjirafa* dhe *MerrJep,* nuk do të afektohen nga instalimet e librarive apo shablloneve të projektit *OfertaSuksesi.* Ky lloj mjedisi na ndihmon që të ndërtojmë tri projekte unike.

Pasi të krijojmë projektin dhe mjedisin virtual, duhet të instalojmë shabllonin “Scrapy*”* e cila bëhet përmes menusë *File-Settings,* që pason me nën-menunë *Python Intepreter*. Me hapjen e kësaj menuje na shfaqet një tabelë me libraritë aktuale të projektit dhe po në këtë menu kërkojmë dhe instalojmë shabllonin *Scrapy,* versionin 2.7.1. (Shiko figurën 22)

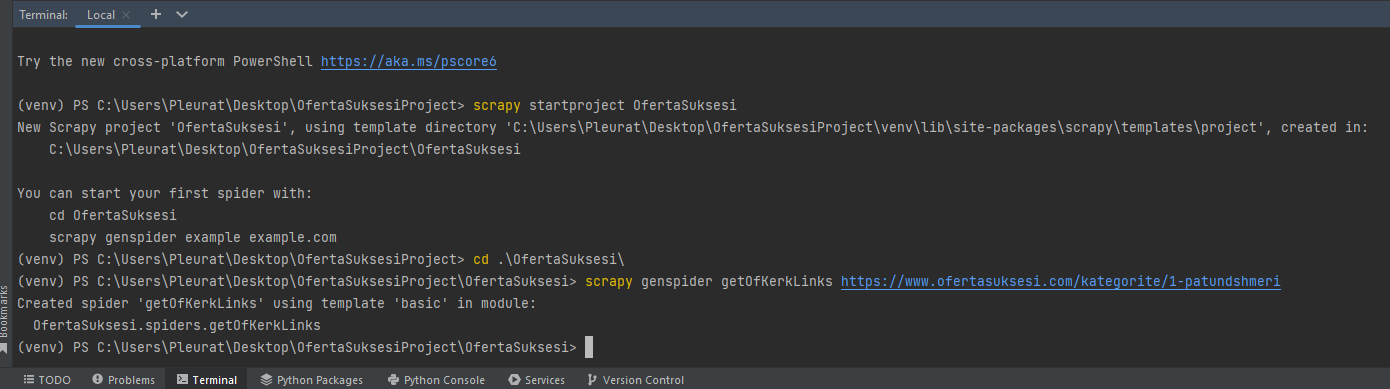


**Figura 21 - Menuja për instalimin e librarive dhe shablloneve të ndryshëm. Rasti “Scrapy”**

**Ndërtimi i projekteve**

Scrapy na ofron disa komanda, ku njëra ndër to na shërben për startimin dhe inicializimin e projektit tonë. Komanda që e kryen këtë punë është ***scrapy startproject OfertaSuksesi.*** Pas krijimit të projektit duhet që të hyjmë brenda projektit përmes komandës ***cd OfertaSuksesi*** dhe pastaj mund të shtypim komandën ***scrapy genspider getOfKerkLinks***

[***https://www.ofertasuksesi.com/kategorite/1-patundshmeri***](https://www.ofertasuksesi.com/kategorite/1-patundshmeri) për të gjeneruar një spider bot që do të na shërbejë si zvarritës në faqe.



**Figura 22 - Startimi i projektit përmes komandave të shabllonit Scrapy**

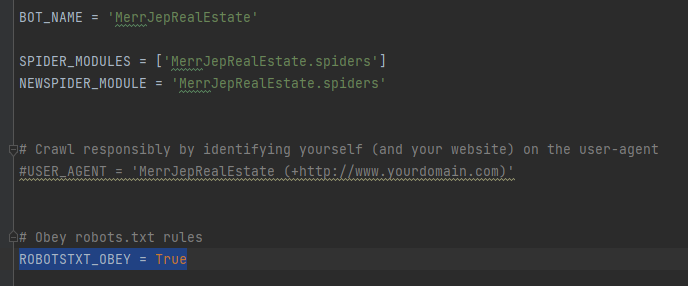
E njëjta procedurë do të përdoret edhe për projektet *Gjirafa* dhe *MerrJep*.

**Modifikimi sipas nevojës të Middlewar-ëve**

**Rregullat për robots.txt**

Në të tre projektet kemi gjeneruar nga një spider bot për zvarritjen dhe nxerrjen e linqeve të patundshmërive. Gjithashtu, me krijimin e botëve janë krijuar edhe “default” Middleware për projektet. Middleware-t kontrollojnë mënyrën e sjelljes së botëve dhe vendi se ku ruhen dhe ku mund të modifikohen botët gjendet në fajllin *Settings.py*.

Për shkak të sigurisë, mbrojtjes nga bllokimi i përkohshëm apo i përhershëm dhe për të respektuar ueb faqet, kemi vendosur që në të tri projektet botët të respektojnë rregullat e ueb faqeve.



**Figura 23 - Parametrat e sjelljes së botëve**

**Kërkesat konkurrente nga Scrapy**

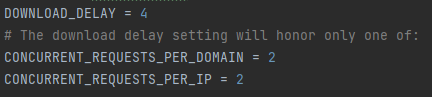
1. ***MerrJep*** – si rezultat i analizës së faqes dhe testimeve, është vërejtur që me dërgimin e shumë kërkesave në të njëjtën kohë në faqen MerrJep, jemi ndëshkuar nga mbrojtja e saj. Për këtë arsye kemi vendosur që ta ulim numrin e kërkesave në minimum. (Shiko figurën 25)



**Figura 24 - Kërkesat konkurrente për uew faqen www.merrjep.com**

Pastaj kemi vendosur të stimulojmë vonesat për kërkesa në katër sekonda, duke dërguar dy kërkesa konkurrente për domen apo dy për IP. Formula e llogaritjes bëhet në atë mënyrë që të stimulojë kërkesat sikur të ishin bërë nga një përdorues njeri, duke shumëzuar kohën e vonesës, me një numër të rëndomtë nga vargu 0.5 deri në 1.5.

**Formula: Download\_Delay = Download\_Delay \* Random\_Number[0.5-1.5]**

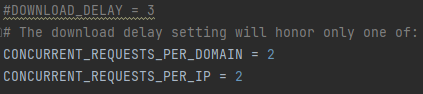


**Figura 25 - Stimulimi i vonesave për kërkesat konkurrente**

1. ***Gjirafa*** – Edhe te kjo faqe kemi hasur në pengesa dhe i kemi ulur kërkesat konkurrente si në figurën 27.

**Figura 26 - Kërkesat konkurrente për ueb faqen www.gjirafa.com**

Tek Gjirafa nuk ishte e nevojshme të stimulohet vonesa, e mjaftueshme ishte që të gjenden dy kërkesa për domen ose IP.



**Figura 27 - Kërkesat konkurrente për domain dhe IP të ueb faqes www.gjirafa.com**

1. ***OfertaSuksesi*** – në këtë faqe nuk ishte e nevojshme që të konfigurohen kërkesat konkurrente sepse nuk është hasur në ndonjë pengesë dhe për këtë arsye, në settings është lënë default vlera. Default vlera nënkupton 16 kërkesa konkurrente si në foto.



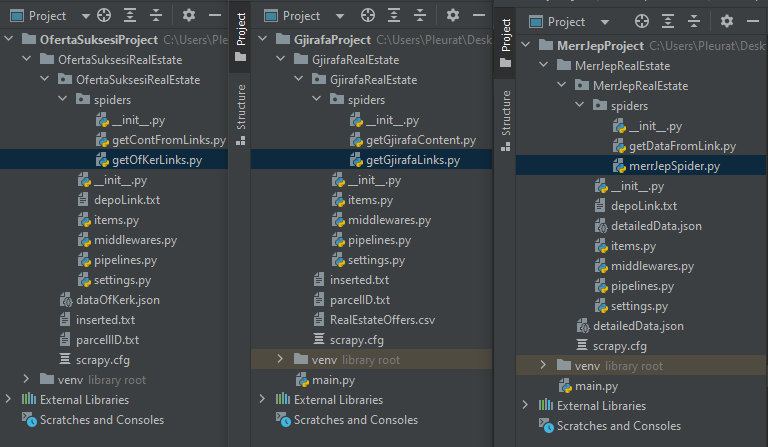
**Figura 28 - Kërkesat konkurrente për ueb faqen www.ofertasuksesi.com**

Gjithashtu, tek ky projekt nuk ka pasur nevojë as për stimulim të vonesave në kërkesa.

**Skripta për nxjerrjen dhe ruajtjen e linqeve të patundshmërive**

Projektet janë ndarë në dy faza duke krijuar nga dy spiders për secilën fazë. Faza e parë ka të bëjë me spiders të cilët kanë për qëllim që të zvarriten nëpër të gjitha ueb faqet, nga fillimi deri në fund dhe të nxjerrin linqet nga to duke i ruajtur në dosjet e krijuara posaçërisht për linqe.

Në momentin e gjenerimit të botëve, krijohen Python files për secilin projekt si: *getOfKerLinks.py*, *getGjriafaLinks.py* dhe *merrJepSpider.py,* në të cilët do të shkruajmë kodin e veprimit të botëve përgjatë scraping. Më poshtë mund të shihen të gjitha dosjet e projekteve (Shih figurën 30)

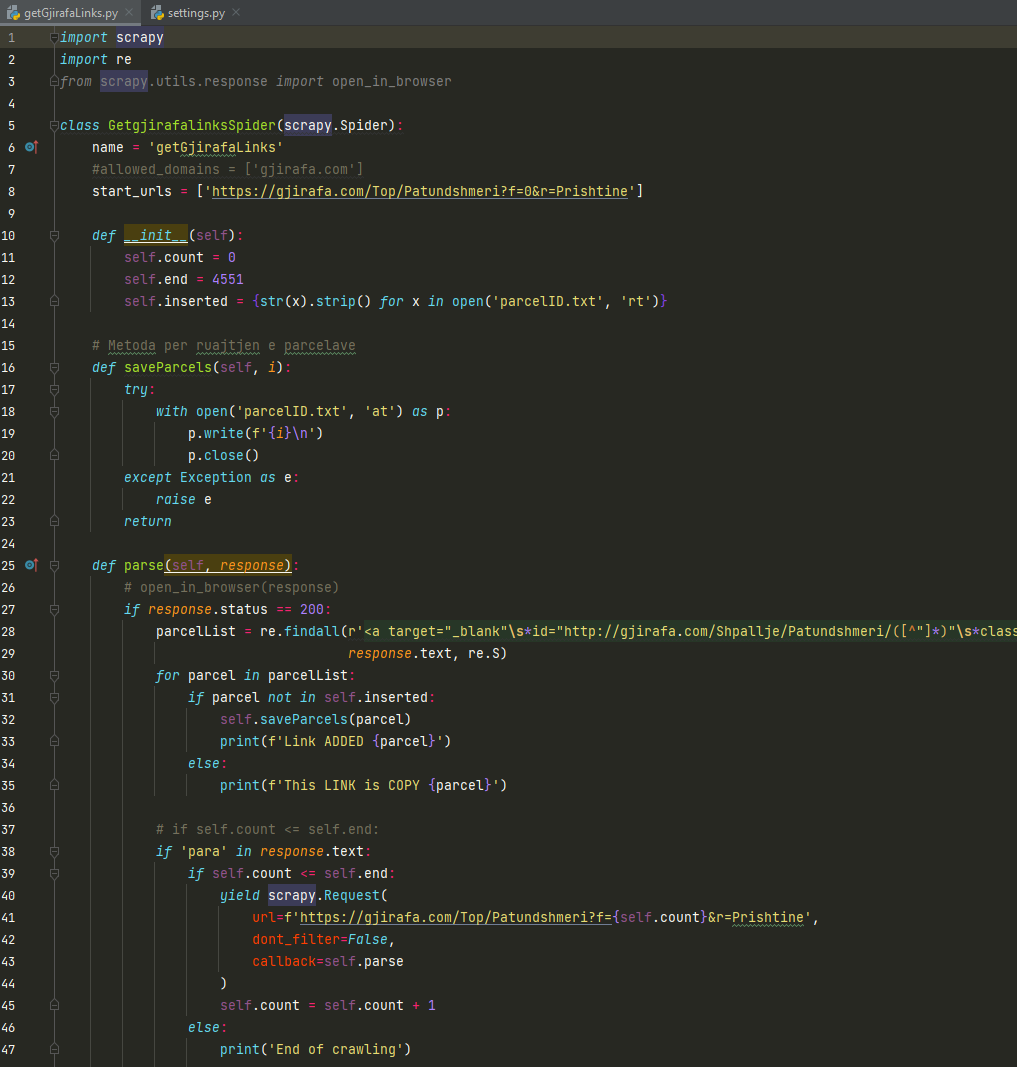


**Figura 29 - Krijimi i skriptave për nxjerrjen e linqeve nga ueb faqet**

***Gjirafa*** – në këtë skriptë do të përdoren dy librari si: “*scrapy*” dhe “*re*”. Libraria “*scrapy*” shërben për përdorimin e scrapyt, ndërsa libraria “*re*” shërben për importimin dhe përdorimin e regex-ave dhe metodave të saj.

Skripta përmban një klasë e cila trashëgon nga “*scrapy.Spider*” dhe përmban anëtarë të ndryshëm. Linkun me të cilin do të startojmë crawling, e vendosim në “*start\_urls*” dhe në konstruktor definojmë vlerat që na duhen pergjatë procesit. Është e rëndësishme të theksohet “*self.inserted*” e cila na ndihmon për të menjanuar shkrimin e të dhënave të dyfishta në dosjen “*parcelID.txt*”.

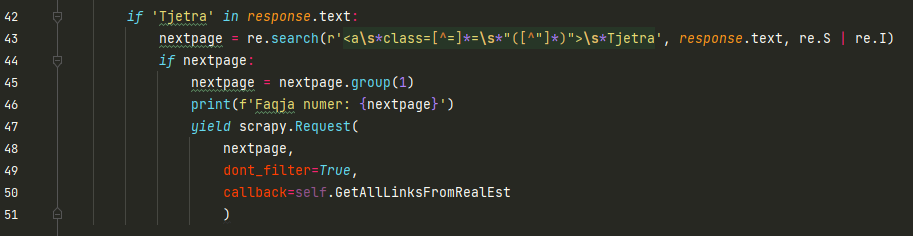
Metoda “*saveParcels*” është një metodë e cila shërben për ruajtjen e linqeve në dosjen “*parcelID.txt*”. Ndërsa metoda “*parse*” shërben për parsimin e përgjigjes së parë dhe përgjatë kësaj metode nxirren linqet e patundshmërive, me ndihmën e regex-ave të cilët pastaj ruhen në dosje.



**Figura 30 - Skripta për nxjerrjen e linqeve nga ueb faqja www.gjirafa.com**

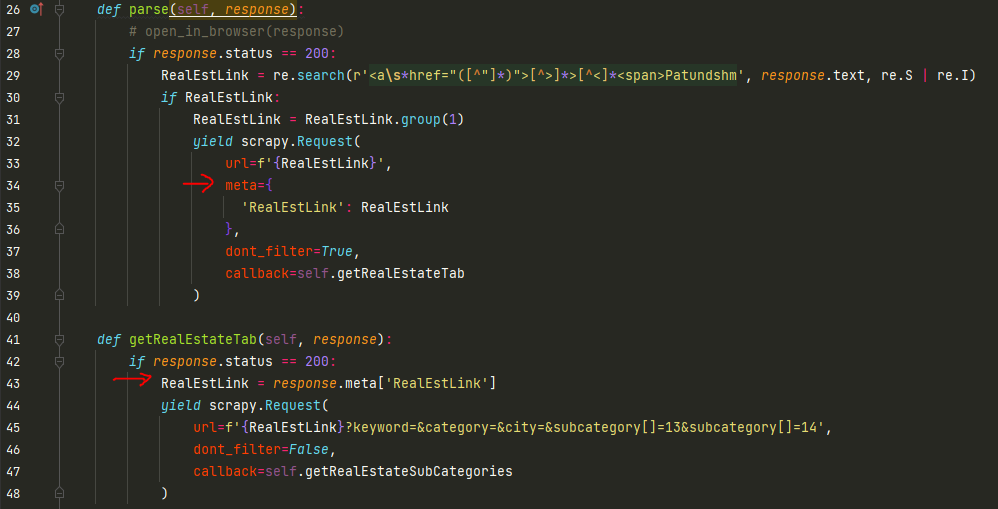
Brenda po të njëjtës metodë, gjendet një pjesë e kodit që shërben për shfletimin e faqeve të njëpasnjëshme. Gjatë analizës është vërejtur se Gjirafa posedon 4551 faqe për rajonin e Prishtinës dhe në kod kemi vendosur kushtin për iterim[[16]](#footnote-16) në bazë të numrit të faqeve.

***MerrJep*** – në këtë skriptë do të përdoren libraritë, metoda për ruajtjen e linqeve dhe metoda e parsimit, ngjashëm si në skriptën e Gjirafës. Mirëpo, tek metoda e parsimit, kodi i shfletimit do të ketë një qasje më ndryshe se ai i Gjirafës. Kushti tregon se përderisa kemi fjalën ‘*Tjetra’* tek përgjigja aktuale, atëherë përmes regex-ave do të targetojmë linkun e faqes së ardhshme dhe do të dërgojmë kërkesë të re për të. Shpesh përdoret edhe parametri “*dont\_filter*”, sipas nevojës.



**Figura 31 - Kodi për faqosje tek skripta për nxjerrjen e linqeve në MerrJep**

***OfertaSuksesi*** – edhe kjo skriptë ka një rrjedhje të ngjashme me atë të Gjirafës dhe MerrJep, mirëpo ka edhe disa elemente të reja. Siç shihet në figurën më poshtë, “meta” është një parametër me vlerë “dict” e cila shërben që vlerat e inicializuara të pasohen si kopje tek një metodë tjetër.



**Figura 32 - Përdorimi i parametrit “Meta” për bartjen e itema-ve nga një metodë në tjetrën**

**Skripta për nxjerrjen dhe ruajtjen e të dhënave të patundshmërive**

Faza e dytë, ka të bëjë me spiders të cilët kanë për qëllim që të dërgojnë kërkesa nëpër të gjitha linqet e gjeneruara më parë dhe të nxjerrin të dhënat e caktuara nga to. Të dhënat e nxjerra do të ruhen në CSV dosje.

Me gjenerim të botëve krijohen dosjet Python për secilin projekt si: *getContFromLinks.py*, *getGjriafaContent.py* dhe *getDataFromLink.py,* në të cilët do të shkruhet kodi i veprimit të botëve për nxjerrjen e të dhënave.

***Gjirafa*** – “get*GjirafaContent.py*” është një skriptë e cila shërben për nxjerrjen dhe ruajtjen e të dhënave në një CSV dosje. Në këtë skriptë janë shfrytëzuar librari të gatshme si “*scrapy*”, “*re*”, “*os.path*”, “*csv*”, “*time*”, dhe “*textwrap*”. Kjo skriptë starton ngjashëm sikurse skripta e parë, me një kërkesë bazë në menunë kryesore të faqes për kërkim.

Komanda: *scrapy crawl getGjirafaContent*

Në konstruktor kontrollohet se a ekziston CSV dosja, në të cilën do të ruhen të dhënat dhe gjithashtu kontrollohet gjendja e shkarkimit të të dhënave dhe linqeve përgjatë procesit.

Metoda “*saveParcels*” na mundëson ruajtjen e linqeve të të dhënave të cilat veçse janë procesuar duke i insertuar tek dosja “*inserted.txt*”, në mënyrë që të ruhet gjendja e ekzekutimit dhe të mënjanohen kërkesat e njëjta.

Metoda “*DecodeData*” shërben për dekodimin e të dhënave të enkriptuara.

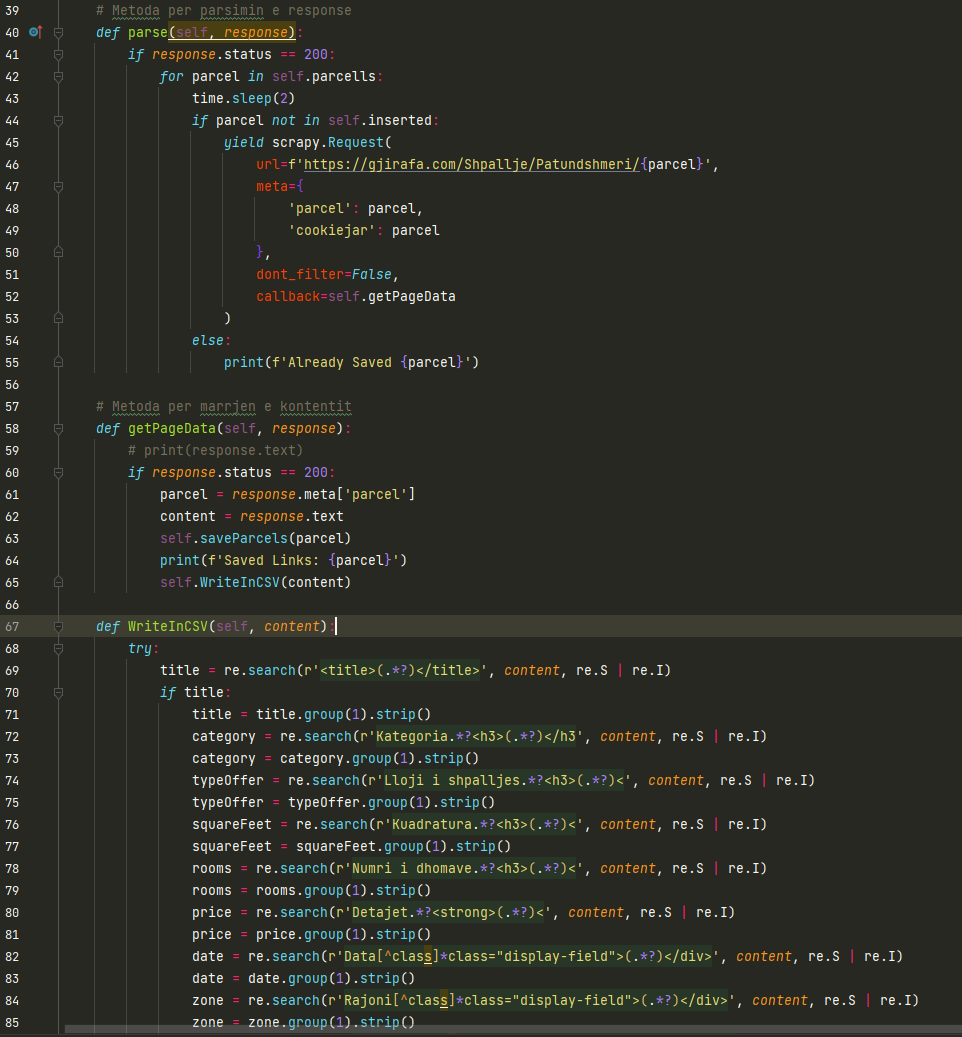
Responsi parsohet tek metoda “*parse*” dhe rishiqohet statusi i saj. Nëse statusi është në rregull atëhere fillon dërgimi i requestave për çdo parcellë duke pauzuar në një interval prej 2 sekondash. Pas kësaj metodë vjen “*getPageData*” një metodë e cila merr responsin dhe e konverton në tekst duke e ruajtur si “content”, në të njëjtën kohë pranon edhe cookies[[17]](#footnote-17) dhe parcelën nga metoda paraprake. Pastaj thirret metoda “*WriteInCsv*”. Kjo metodë na mundëson filtrimin e të dhënave specifike përmes regex-ave në përmbajtjen që është nxjerrë. Pas filtrimit të këtyre të dhënave, po ashtu mundësohet shkrimi i tyre në dosjen CSV.



**Figura 33 - Skripta për nxjerrjen e të dhënave për www.gjirafa.com 1/3**

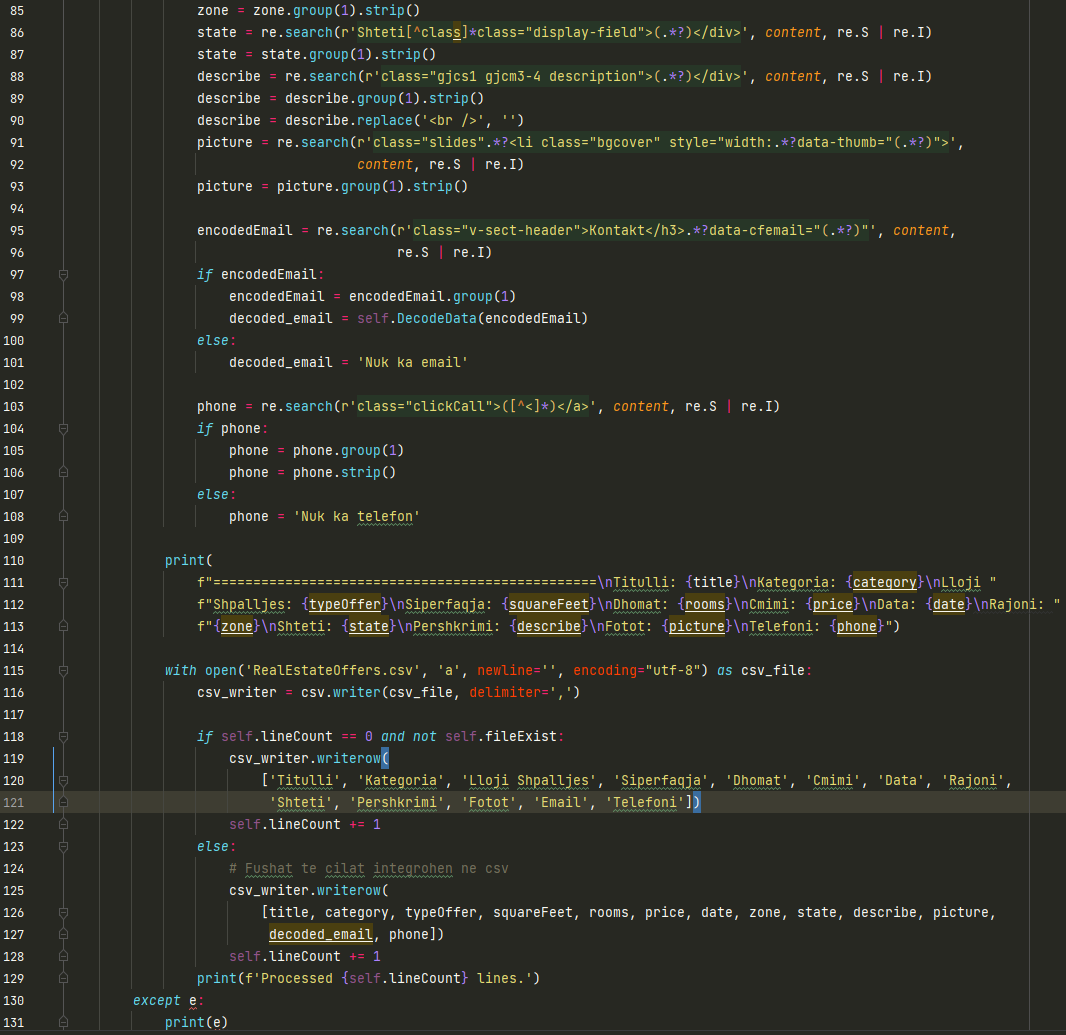
***Enkodimi*** - Një nga problemet kryesore që është hasur përgjatë këtij hulumtimi, është dekodimi të dhënave. Për t’iu qasur këtyre të dhënave dhe për t’i gjeneruar ato, së pari është dashur të bëhet hulumtimi për llojin e algoritmit të enkriptimit që është aplikuar nga Gjirafa. Pasi është gjetur algoritmi i enkriptimit, atëherë është ndërtuar një metodë e cila bën dekodimin e vlerave të enkriptuara dhe pastaj këto vlera janë filtruar më tutje. Një rast konkret i tillë, është dekodimi i të dhënave për email-a. (Shih figurën 34)

***Vëllimi i kërkesave dhe përgjigjeve*** - Gjatë nxjerrjes së të dhënave është hasur në një problem ku serveri nuk arrinte që të kthente përgjigjje në të njëjtën shpejtësi me dërgimin e kërkesave në drejtim të tij. Ky problem shkakton replikim të të dhënave dhe përgjigjes. Për zgjidhjen e këtij problemi, është ngadalësuar dërgimi i kërkesave në drejtim të serverit, duke vendosur një interval pauze për 1 ose 2 sekonda ndërmjet kërkesave.

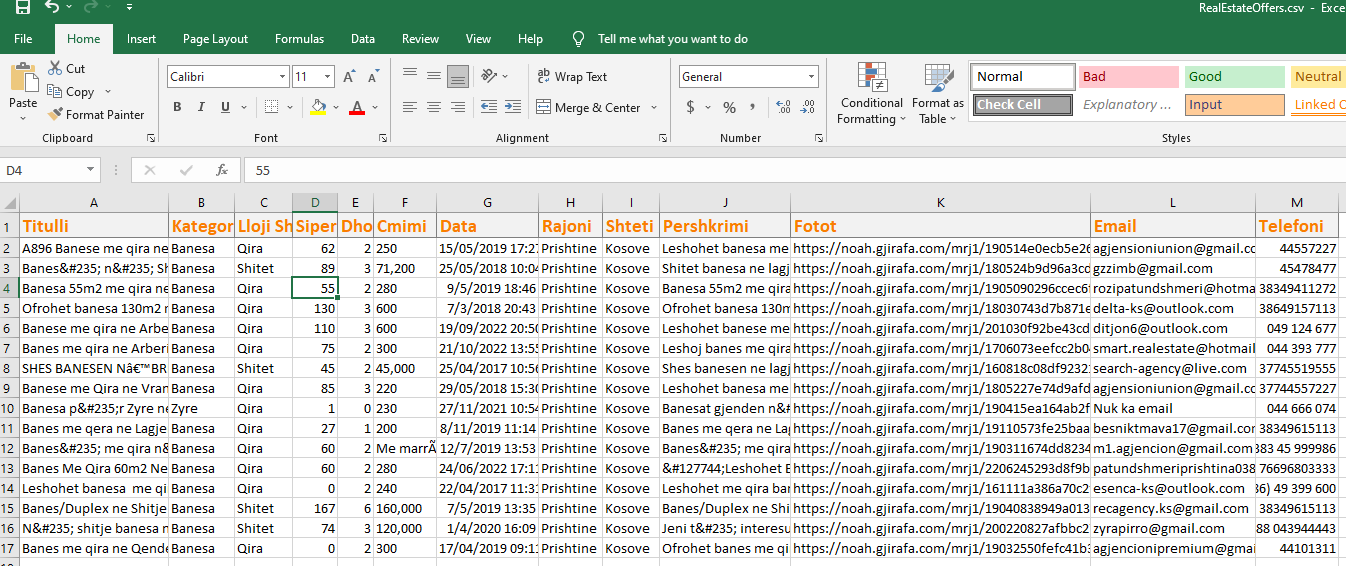


**Figura 34 - Skripta për nxjerrjen e të dhënave për www.gjirafa.com 2/3**

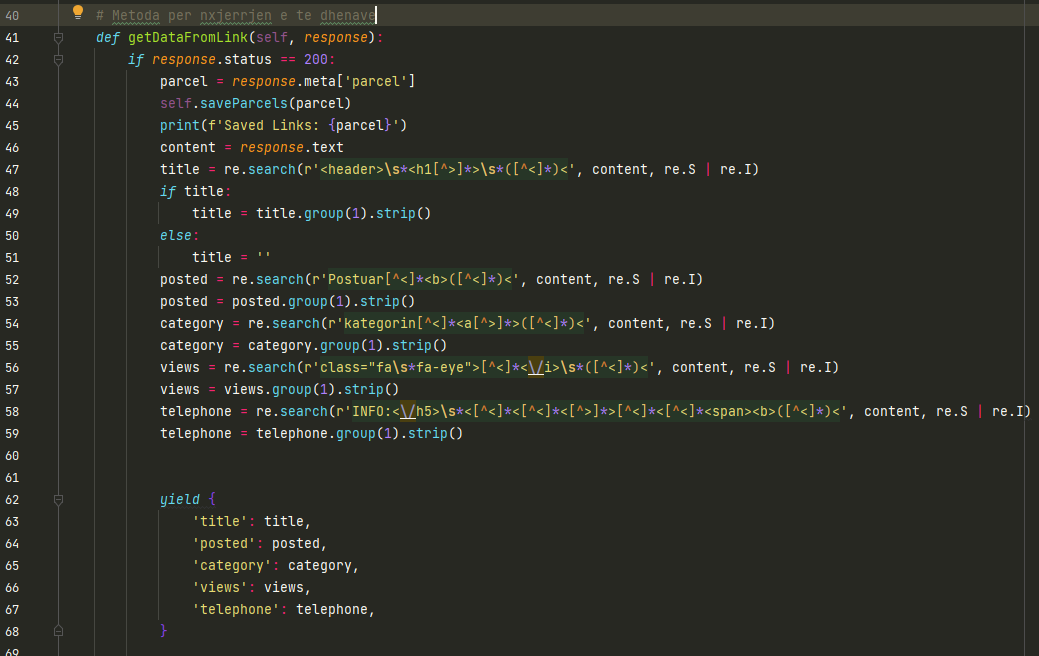
***Cookies*** - kur scrapy spider bën kërkim në ueb faqe, ai automatikisht i trajton cookies për ne duke i ruajtur dhe dërguar ato në kërkesat e radhës në të njëjtën ueb faqe. Kërkesat në scrapy janë asinkrone, që nënkupton se kemi shumë kërkesa konkurrente që e ndajnë të njëjtin cookie. Për t’iu shmangur problemit, përdorim *CookieJar* ose parametrin *meta* të cilët na ndihmojnë të përdorim cookies të ndryshëm në kërkesa të ndryshme (Shih figurën 35).



**Figura 35 - Skripta për nxjerrjen e të dhënave për www.gjirafa.com 3/3**

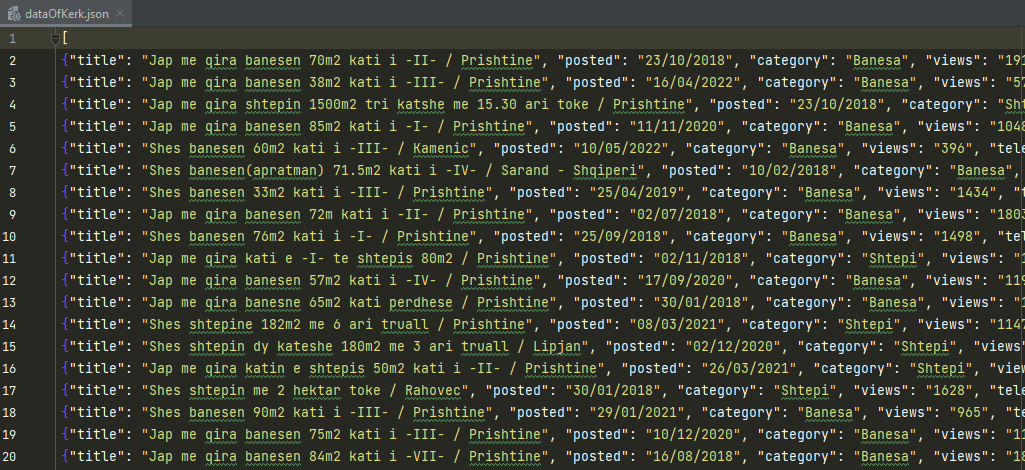


**Figura 36 - Mostër e nxerrjes së të dhënave për patundshmëri nga faqja www.gjirafa.com**

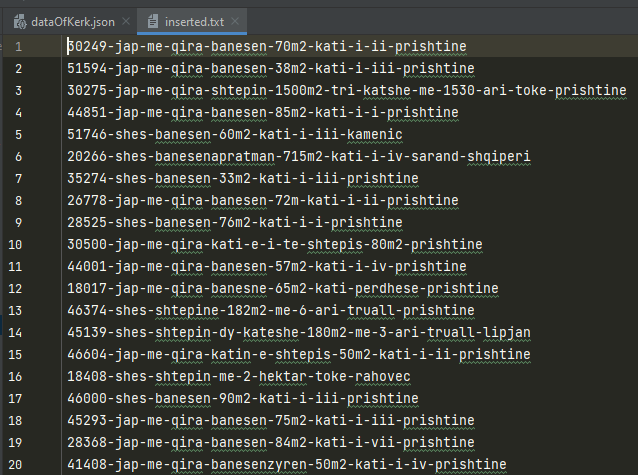
***OfertaSuksesi*** – “*getContFromLinks.py*” është skriptë për nxjerrjen e të dhënave, e ngjashme si ajo e Gjirafës, por më e thjeshtë për nga kompleksiteti, sepse faqja ka strukturë më të thjeshtë. Në këtë skriptë janë përdorur vetëm dy librari “*scrapy*” dhe “*re*”.

**Figura 37 - Metoda për nxjerrjen e të dhënave nga ueb faqja www.ofertasuksesi.com**

Në këtë projekt, të dhënat do të ruhen në një format tjetër të njohur si JSON dhe kjo realizohet duke shtypur komandën: *scrapy crawl getContFromLinks -o dataOfKerk.json* (Shih figurën 39)



**Figura 38 - Ruajtja e të dhënave në formatin JSON për projektin www.ofertasuksesi.com**

Me ruajtjen e të dhënave do të ruhen gjithashtu edhe linqet e procesuara ne dosjen “inserted.txt” .

**Figura 39 – Ruajtja e linqeve të procesuara nga projekti www.ofertasuksesi.com**

(Shih figurën 40)

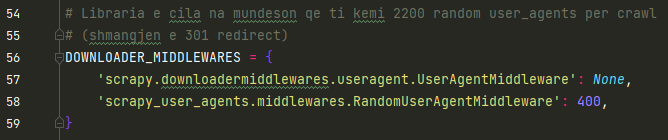
***MerrJep*** – “*getDataFromLink.py*” është një skriptë që shërben për nxjerrjen dhe ruajtjen e të dhënave. Duke u bazuar në strukturë, ka një ngjashmëri me skriptat e Gjirafës dhe OfertaSuksesit por ka edhe veçoritë e saj që e veçojnë nga të tjerat.

***UTF-8*** – gjatë nxjerrjes së të dhënave janë vërejtur se disa karakere janë të shprehura me shifrime të ndryshme. Për t’i konvertuar këto shifrime nga përgjigja është përdorur metoda e zëvendësimit si p.sh. *replace(“&#203;”, “Ë”).*

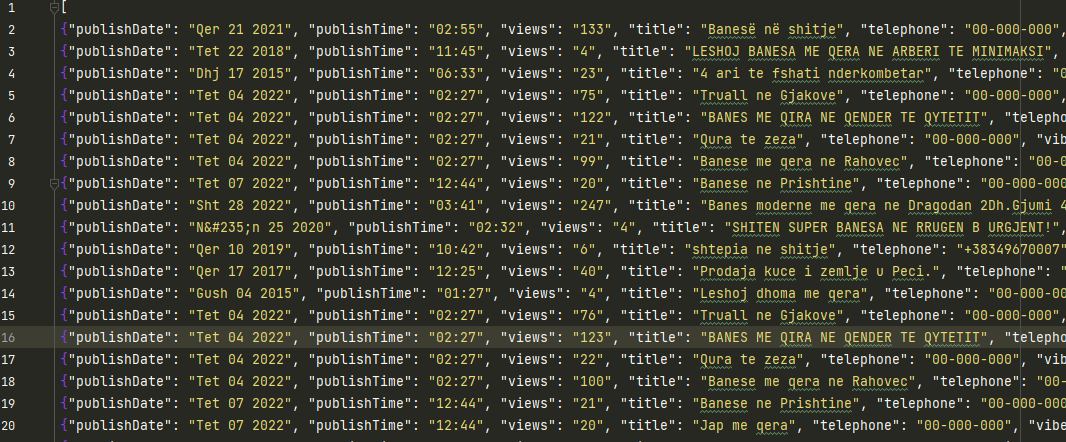


**Figura 40 - UTF-8 për enkodim të karaktereve Unicode në projektin www.merrjep.com**

***Redirect 301*** – një nga vështirësitë më të mëdha në këtë projekt ka qenë përballja me ridrejtimin e kërkesave në 301. Për ta shmangur këtë problem është instaluar një librari e njohur si “scrapy-user-agents 0.1.1” e cila shërben me User-Agent të rëndomtë. Dosja e cila gjendet në këtë librari përmban 2200 user-agents. Në momentin që dërgojmë kërkesë, do të testohen disa user-agents dhe do të përdoret ai më i përshtatshmi për realizimin e kërkesës.



**Figura 41 - Libraria scrapy-user-agents me 2200 user-agents**



**Figura 42 - Mostër e nxerrjes së të dhënave për patundshmëri nga faqja www.merrjep.com**

# **KONKLUZIONET**

Përgjatë ndërtimit të skriptave, për të bërë zvarritjen dhe nxjerrjen e të dhënave nëpër ueb faqe kemi arritur në disa konkluzione të ndryshme të cilat elaborohen si më poshtë.

Është vërejtur se nuk është aq e lehtë që të ndërtohen skriptat për të bërë zvarritjen dhe nxjerrjen e të dhënave nëpër ueb faqe për shumë faktorë të ndryshëm. Disa nga to janë kompleksiteti dhe struktura dinamike e ueb faqeve, që nënkupton se nuk janë asnjëherë të njëjta dhe kjo e bën tejet të vështirë sepse nuk mund të përdorim skriptën e njëjtë për të gjitha ueb faqet. Gjithashtu vlenë të theksohet se përmbajtja, format dhe struktura e ueb faqeve mund të ndryshojnë vazhdimisht përgjatë kohës, nga faktori njeri. Shembull janë programerët që tentojnë t’i përditësojnë ueb faqet me qëllimin për t’i shërbyer sa më mirë përdoruesit të tyre. Kjo na lë të kuptojmë se skriptën të cilën e kemi ndërtuar sot për një ueb faqe adekuate, nuk do të thotë se të nesërmen do të funksionojë në po të njëjtën ueb faqe. Gjatë punës dhe analizës brenda këtyre faqeve për shkak të periudhës së shkurtër nuk është hasur në ndonjë përditësim të përmbajtjes së tyre.

Faktor tjetër i rëndësishëm në këtë fushë është edhe faktori kohë. Pra, gjatë këtij hulumtimi kemi arritur në një përfundim se përdorimi i veglës Scrapy, e cila funksionon në mënyre asinkrone, na mundëson që volumet e mëdha të të dhënave t’i nxjerrim në mënyrë sa më të lehtë dhe të shpejtë në krahasim me teknologjitë e tjera.

Gjatë këtij hulumtimi është hasur në pengesa të ndryshme dhe për shumicën e tyre është gjetur ndonjë zgjidhje apo alternativë për t’i tejkaluar ato, si:

* Vëllimi i kërkesave dhe përgjigjeve – përdorimi i settings si: *Concurrent Requests*, *Concurrent Reuqests Per Domain, Concurrent Requests Per IP;*
* Rate Limiting dhe bllokuesit e IP-ve –modifikimi në settings *Download Delay;*
* Faqezimi – përdorimi i metodave kreative për të realizuar faqezimin;
* Dërgimi i kërkesave të njejta – largimi i kërkesave të dyfishta përmes ruajtjes së linqeve;
* Enkodimi i të dhënave (rasti Gjirafa) – dekodimi i të dhënave (rasti email-ave);
* Cookies – kemi bërë bartjen e cooki-es së bashku me kërkesat;
* UTF-8 – konvertimi i karaktereve duke i zëvendësuar ato;
* Redirect 301 – përdorimi i librarisë *scrapy user agents* për ndërrimin e tyre;

Por duhet përmendur, se nuk është etike që të abuzojmë me botët e ndërtuar, ndaj ueb faqeve apo resurseve të ueb serverëve për të arritur qëllimin tonë, sepse mund të shkaktojmë dëme të mëdha ndaj ofruesve të shërbimeve, përdoruesve të tjerë në ueb faqe, duke ua mohuar shërbimet atyre dhe ndaj vetvetes duke rrezikuar me bllokim të përhershëm. Me këtë arrijmë në konkluzion se mund të ndërtojmë botë të ndryshëm për të arritur qëllimet tona, duke respektuar dhe iu përmbajtur rregullave të vendosura nga zhvilluesit e ueb faqeve dhe pa dëmtuar palët e tjera.

# **REFERENCAT DHE BIBLIOGRAFIA**

Andrade, F., 2020. Web Scraping with Beautiful Soup, Selenium, or Scrapy?. *Towards Data Science,* Issue Dec 16, 2020.

Bill Tucker, Mukul Mantosh, Nafiul Islam, Paul Everitt, 2022. *jetbrains.* [Në linjë]   
Available at: https://www.jetbrains.com/help/pycharm/quick-start-guide.html

Cloudflare, 2022. *What is content scraping? Web scraping.* [Në linjë]   
Available at: https://www.cloudflare.com/learning/bots/what-is-content-scraping/

Connell, A., 2021. *What Is A Robots.txt File? And How Do You Create One? (Beginner’s Guide).* [Në linjë]   
Available at: https://bloggingwizard.com/create-custom-robots-txt-file

Developers, S., 2022. *Scrapy 2.7 Documentation.* [Në linjë]   
Available at: https://docs.scrapy.org

DIlmegani, C., 2021. *Beginner’s Guide for Web Scraping: Challenges & Best Practices.* [Në linjë]   
Available at: https://research.aimultiple.com/web-scraping-challenges/  
[Qasja 26 10 2022].

Dilmegani, C., December 15, 2020. *AI Multipl Web-Crawler.* [Në linjë]   
Available at: https://research.aimultiple.com/web-crawler/  
[Qasja 17 September 2022 ].

Johnson, D., 2022. *What is a CSV file? How to open and use the popular spreadsheet file.* [Në linjë]   
Available at: https://www.businessinsider.com/guides/tech/what-is-csv-file

Kenny, C., a.d. *Web crawling vs web scraping.* [Në linjë]   
Available at: https://www.zyte.com/learn/difference-between-web-scraping-and-web-crawling/

Palakollu, S. M., 2019. Scrapy Vs Selenium Vs Beautiful Soup for Web Scraping. *Medium,* Issue Jul 4, 2019.

Palakollu, S. M., 2019. Scrapy Vs Selenium Vs Beautiful Soup for Web Scraping. *Medium,* Issue Jul 4, 2019.

Perez, M., 2020. Web Scraping vs Web Crawling: What’s the Difference?. *Parse Hub.*

Rafik, A., 2021. *Modern Web Scraping with Python using Scrapy, Splash, Selenium,* s.l.: Udemy.

Rivera, I., 2022. *Main Web Scraping Challenges and Ways to Overcome Them.* [Në linjë]   
Available at: https://proxyway.com/guides/web-scraping-challenges

Shirey, T., a.d. Web Crawler 101: What Is a Web Crawler and How Do Crawlers Work?. *Web FX.*

Singhal, G., 2020. *Advanced Web Scraping Tactics.* [Në linjë]   
Available at: https://www.pluralsight.com/guides/advanced-web-scraping-tactics-python-playbook

Sulzenko, A., 2021. *Data Science Central.* [Në linjë]   
Available at: https://www.datasciencecentral.com/optimal-scraping-technique-css-selector-xpath-amp-regex/  
[Qasja 12 11 2022].

1. Query – pjesë e URL, që u cakton vlera parametrave specifikë për kërkim në “Google” apo ndonjë motor tjetër kërkues. [↑](#footnote-ref-1)
2. Framework – mjet që përmban komponente apo zgjidhje të gatshme për të përshpejtuar zhvillimin e kodeve. Në punim do t’iu referohet si “shabllone”. [↑](#footnote-ref-2)
3. Spiders – i referohet skriptave që shërbejnë për nxjerrjen e të dhënave, e që ndryshe njihen si “bot”. [↑](#footnote-ref-3)
4. Bot – shkurtesë për “robot” [↑](#footnote-ref-4)
5. Web Driver – është një ndërfaqe kontrolluese, që mundëson ekzaminimin dhe kontrollimin e agjentëve të përdorimit. [↑](#footnote-ref-5)
6. DOM – është një program ndërmjetësues për HTML dhe XML dokumentet. [↑](#footnote-ref-6)
7. Proxy – diçka që vepron si ndërmjetësues pë diçka tjetër. [↑](#footnote-ref-7)
8. Open Source – i referohet çfarëdo programi, kodi i të cilit është në dispozicion për përdorim apo përditësim nga cilido përdorues dhe kontribues. [↑](#footnote-ref-8)
9. Asinkrone – në programim nënkupton, kur një process operon i pavarur nga proceset tjera. [↑](#footnote-ref-9)
10. Scrape – i referohet veprimit të kërkimit, shfletimit apo nxjerrjes së të dhënave/linqeve. [↑](#footnote-ref-10)
11. Parsing - shndërrimi i të dhënave nga një format në tjetrin. [↑](#footnote-ref-11)
12. Que – pritje në radhë [↑](#footnote-ref-12)
13. Stringje – sekuencë karakteresh. [↑](#footnote-ref-13)
14. Lejuara ose të pa lejuara. [↑](#footnote-ref-14)
15. Debug – identifikimi dhe largimi i gabimeve nga softuerët. [↑](#footnote-ref-15)
16. Iterim – sekuencë e udhëzimeve ose kodit, që përsëritet derisa të arrihet një rezultat specifik. [↑](#footnote-ref-16)
17. Cookies – informacion i ruajtur në kompjuterin e një përdoruesi të internetit për qëllime të menaxhimit të sesioneve, personalizimit dhe gjurmimit. [↑](#footnote-ref-17)