

Міністерство освіти і науки України
Криворізький національний університет
Кафедра моделювання та програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття ступеня вищої освіти магістра
зі спеціальності 123 – Комп'ютерна інженерія

На тему: Програмне забезпечення та принципи організації парсінгу даних у соціальних мережах

Засвідчую, що в цій
кваліфікаційній роботі немає
запозичень із праць інших
авторів без відповідних
посилань.

Студент гр. ІПЗ-21м
_____ / М.А. Слюсар /

Керівник
кваліфікаційної роботи _____ / А. І. Купін /

Завідувач кафедри _____ / А. І. Купін /

Кривий Ріг

2022

Криворізький національний університет

Факультет: Інформаційних технологій

Кафедра: Моделювання та програмного забезпечення

Ступінь вищої освіти: магістр

Спеціальність: 123 – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ:

Зав. кафедри

_____ А. І. Купін

«___» _____ 2021 р.

ЗАВДАННЯ

на кваліфікаційну роботу

студенту групи КІ-21м Микиті Олексійовичу

1. Тема: дослідження комп'ютеризованих системи безпеки та розробка програмного забезпечення інтелектуальної протиугінної системи затверджено наказом по КНУ № 211с від «30» березня 2022 р.
2. Термін подання студентом закінченої роботи: «4» листопада 2022 р.
3. Вихідні дані по роботі: розроблюване програмне забезпечення повинне зробити системи протиугінних систем більш доступними для населення при умові забезпечення високого ступеня захисту та функціоналу.
4. Зміст пояснювальної записки (перелік питань, що їх треба розробити): виявити слабкі місця у системах захисту сучасних авто, проаналізувати існуючі аналоги та їх функції, розробити ситему на більш дешевій платформі Arduino без шкоди для функціоналу та з індивідуальними, досі не існуючими функціями.
5. Перелік ілюстративного матеріалу: функціональна схема, блок-схеми розроблених алгоритмів та роботи системи, макети інтерфейсу користувача.

Календарний план:

№	Найменування етапів кваліфікаційної роботи	Термін виконання етапів роботи
1	Огляд літературних джерел за темою	13.04.2022 – 19.04.2022
2	Аналіз існуючих аналогів системи	20.05.2022 – 25.05.2022
3	Аналіз методів розробки системи	26.06.2022 – 04.06.2022
4	Аналіз предметної області, формулювання вимог до системи	05.07.2022 – 09.07.2022
5	Підготовка матеріалів першого розділу роботи	10.08.2022 – 17.08.2022
6	Розробка блок-схем та діаграм розроблюваного програмного забезпечення	18.09.2022 – 25.09.2022
7	Підготовка матеріалів другого розділу роботи	26.10.2022 – 30.10.2022
8	Реалізація схемотехніки системи	31.10.2022 – 31.10.2022
9	Реалізація функціоналу інтерфейсу користувача	01.11.2022 – 21.11.2022
10	Підготовка матеріалів третього розділу роботи	21.11.2022 – 28.11.2022
11	Оформлення пояснювальної записки	28.11.2022 – 30.11.2022

Дата видачі завдання:

«30» березня 2022 р.

Студент

_____ / Н. С. Слюсар /

Керівник роботи

_____ / А. І. Купін /

(підпис)

РЕФЕРАТ

Пояснювальна записка: 49 сторінки, 16 рисунків, 20 використаних джерел, 9 додатків.

Проект складається з трьох розділів.

Перший розділ присвячений опису поняття парсеру, веб-скрейпінгу та краулінгу.

У другому розділі розглянуто варіанти сучасні інструменти для веб-скрейпінгу.

Третій розділ присвячений створенню програми парсеру студентів Криворізького Національного Університету.

ABSTRACT

Explanatory note: 49 pages, 16 figures, 20 sources used, 9 appendix.

The project consists of three sections.

The first section is devoted to the concept of parser, web scraping and crawling.

The second section discusses the options for modern tools for web scraping.

The third section is devoted to creating a parser program for students of Kryvyi Rih National University.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. ВПЛИВ СОЦІАЛЬНИХ МЕРЕЖ НА СУЧАСНЕ СУСПІЛЬСТВО	6
1.1 Загальні відомості про об'єкт дослідження	6
1.2 Історія розвитку соціальних мереж	8
1.3 Аналітика методів впливу соціальних мереж	9
1.4 Негативний вплив соціальних мереж	11
1.5 Цілі і завдання дипломної роботи	15
РОЗДІЛ 2. ОСНОВОПОЛОЖНІ ПРИНЦИПИ ТЕХНОЛОГІЇ ВЕБ-СКРЕЙПІНГА	16
2.1 Технологія веб-скрейпінгу	16
2.2 Краулінг	21
2.3 Парсинг	26
2.4 Способи зберігання інформації	30
РОЗДІЛ 3. СУЧАСНІ ІНСТРУМЕНТИ ДЛЯ СКРЕЙМІНГА	32
3.1 Обмеження на використання скрейперів	33
3.2 LinkedIn – професійна соціальна мережа	35
РОЗДІЛ 4. ПРОЦЕС РОЗРОБКИ ТА ТЕСТУВАННЯ	42
РОЗДІЛ 5. ФІНАНСОВЕ ПІДГРУНТЯ ПРОЕКТУ	50
ВИСНОВОК	51

					КНУ.РБ.123.20.01.ВС				
Змн.	Арк.	№ документа	Підпис	Дата	ЗМІСТ	Літера	Аркуш	Аркушів	
Розробив		Слюсар							
Перевірив		Купін							
Н. контроль		Кузнєцов							
Затвердив		Купін				KI-21M			

ВСТУП

Сьогодні, в епоху інформаційних технологій, знання є найважливішим фактором успіху в суспільстві. Завдяки Інтернету кожен має можливість здобувати знання.

Хоча формально Інтернет є глобальною мережею обчислювальних ресурсів, колективний доступ яких базується на використанні єдиної стандартної схеми адресації, потужної магістралі та високошвидкісних ліній зв'язку з головними комп'ютерами мережі. Сьогодні Інтернет - це не просто засіб спілкування між людьми, він також є найбільшим у світі джерелом інформації. І це джерело швидко зростає, адже щодня створюється багато веб-сайтів та порталів з важливим та цікавим контентом. Однак людина не може оволодіти такою кількістю інформації, і саме тут з'являється допомога комп'ютерів, особливо технології веб-вишкрібання.

Актуальність роботи. Веб-скрапінг - відносно новий винахід, який вплинув на життя кожного, хто в одному. Так чи інакше, стикається з необхідністю збору даних з Інтернету значно спрощується. Вишкрібання - це технологія, яка використовує сценарії для входу на веб-сайт, який видається звичайним користувачем, та збору інформації відповідно до заздалегідь визначених параметрів. Таким чином, дані з тисяч веб-сайтів можна отримати, обробити, систематизувати та зберегти у форматі простого тексту протягом декількох хвилин. Ця технологія особливо затребувана в галузі журналістики та статистики, що підтверджує її актуальність.

Мета і завдання дослідження. Основною метою даної є вивчення технології веб-вишкрібання та глибока дослідницько-аналітична робота по

					КНУ.РБ.123.20.01.ВС		
Зм.	Арк.	№ документа	Підпис	Дата			
Розробив		Слюсар			ВСТУП	Пітера	Аркуш
Перевірив		Купін					Аркушів
Н. контроль		Кузнєцов					
Затвердив		Купін				KI-21M	

спрощенню технологій надання важливої та відповідної інформації. Результатом цієї роботи є створення програми з використанням технології веб-скрапінгу.

Об’єкт дослідження – процеси оптимізації пошуково-фільтруючих операцій за допомогою сучасних комплексів методів фільтрації інформації

Предмет дослідження – моделі та засоби реалізації оптимізації специфічних пошукових операцій у соціальних мережах.

Методи досліджень. Використовуючи наукові інформаційні джерела, аналітичні методи та реалізацію прикладу програми дослідити параметри оптимізації пошукових операцій за вузьким спектром факторів.

Наукова новизна одержаних результатів. Визначена послідовність дослідження питання комплексної оптимізації пошукових процесів та тестова реалізація програми дозволить реалізувати фільтраційний процес та отримати більш оптимізований результат пошуку.

Практичне значення отриманих результатів. Результати досліджень дозволять пришвидшити процеси пошукових процесів та наладжувати їх на комплексно-вибірковий лад.

Апробація роботи. Апробація досліджень відбувалась на конференції KICM 2019 від Криворізького національного університету. Участь у «Всеукраїнському науково-практичній конференції здобувачів вищої освіти та молодих учених “Комп’ютерна інженерія і кібербезпека: досягнення та інновації” (27–29 листопада 2018 р., м. Кропивницький).

РОЗДІЛ 1. ВПЛИВ СОЦІАЛЬНИХ МЕРЕЖ НА СУЧАСНЕ СУСПІЛЬСТВО

1.1 Загальні відомості про об'єкт дослідження

Однією з найдивовижніших знахідок, виявлених в давньоримському місті Помпеї, стали численні графіті, які вчені в один голос назвали прообразом сучасних соціальних мереж.

Власники міських будинків свідомо віддавали свої стіни під розпис всім бажаючим. Римські графіті активно креслили і малювали представники всіх станів – від рабів до знатних вельмож. Будь-хто міг зізнатися кому-небудь в любові, залишити добрі побажання, потролити господаря приміщення, а поодинокі повідомлення часто переростали в «чат».

Приклад Помпеїв зайвий раз доводить, що все нове – це добре забуте старе.

Нинішні соціальні мережі спираються на інший, більш технологічний фундамент, але знаходять приблизно те ж застосування, що і дві тисячі років тому. [4]

“Актуальність теми даного дослідження обумовлена перш за все тим, що уявити сучасну комунікацію без соціальних мереж неможливо. Так, щомісяця у всесвітню павутину в Росії виходять 65,9 млн осіб, з яких соцмережі відвідують понад 90% 1.” [2] За даними “Brand Analytics, кожен день в соціальних мережах з'являється близько 30 млн нових повідомлень (350 постів в секунду!), і 35 млн осіб залишають хоча б одне публічне повідомлення на місяць.” [5]

					КНУ.РБ.123.20.01.ВС			
Зм.	Арк.	№ документа	Підпис	Дата	ВПЛИВ СОЦ МЕРЕЖ	Літера	Аркул	Аркулів
Розробив		Слюсар						
Перевірив		Купін						
Н. контроль		Кузнєцов						
Затвердив		Купін				КІ-17-1		

Мета теоритичнонь частини проекту - проаналізувати вплив соціальних мереж на життя сучасного суспільства.

Для досягнення цієї мети були поставлені наступні завдання:

1. розкрити поняття "соціальна мережа" і розглянути процес розвитку цього явища;
2. виявити позитивні сторони соціальних мереж, підкреслити їх значимість для сучасної людини;
3. привернути увагу до негативних наслідків використання соціальних мереж.

Методологічну основу роботи складають праці вітчизняних і зарубіжних вчених, пов'язані з оцінкою становлення і розвитку соціальних мереж; загальнофілософські методи і принципи аналізу діалектики і логіки, а також системно-структурний, функціональний і статистичний методи.

Соціальні мережі-це унікальне явище двадцять першого століття, яке міцно увійшло в життя людей. Середній вік користувачів варіюється від 7 до 48 років, а час, проведений у віртуальному, вигаданому світі, порахувати просто неможливо. Під соціальною мережею (від англ. socialnetworkingservice) розуміється платформа, онлайн-сервіс або веб-сайт, призначені для побудови, відображення та організації соціальних взаємин, візуалізацією яких є соціальні графи. [12] Соціальний граф (англ . Socialgraph)-це графік, вузли якого представлені соціальними об'єктами, такими як користувацькі профілі з різними атрибутами (наприклад: ім'я, день народження, рідне місто тощо), спільноти, медіа – контент тощо, а ребра-соціальними зв'язками між ними. Причиною появи цього феномена є підвищена потреба в спілкуванні властива самим різним людям, в силу чого популярність деяких соціальних мереж в даний час б'є всі мислимі рекорди.

1.2 Історія розвитку соціальних мереж

Процес розвитку і становлення соціальних мереж пов'язаний з США, припадає на кінець 20 століття, а на пострадянському просторі соціальні мережі стали з'являтися з 2006 року. Першими великими соціальними мережами у пострадянському сегменті були «Однокласники» і «Вконтакте», соціальні мережі російського виробництва.

«Однокласники» є аналогом проекту Classmates. «Вконтакте» можна назвати точною копією сайту FaceBook.

Проект "Однокласники" був запущений 4 березня 2006 Альбертом Попковим. Головним завданням соціальної мережі був пошук однокласників, однокурсників, колишніх випускників, друзів, знайомих, а так само родичів, з якими після розпаду Радянського Союзу і масової міграції населення була втрачена зв'язок. За даними на 2014 р. кількість активних акаунтів у соціальній мережі «Однокласники» перевищила 70 мільйонів. Про це йдеться в статистичному звіті, поширеному адміністрацією ресурсу. Щоденна аудиторія соціальної мережі становить близько 20 мільйонів чоловік. Приблизно чверть всіх активних користувачів заходять в «Однокласники» за допомогою мобільних пристроїв, а третина аудиторії ресурсу проживає за кордоном [7].

Найвідоміша і популярна соціальна мережа в СНД "Вконтакте" була запущена 10 жовтня 2006 року і позиціонувала себе як соціальна мережа для випускників вузів. На сьогоднішній день щодня сайт відвідується більше 228 мільйонів користувачів і є найпопулярнішим і відвідуваним сайтом на арені ресурсів СНД. У 2012 році «Вконтакте» перейшов на новий домен "VK.com". На сьогоднішній день «Вконтакте» є найбільшим відео- і аудіо-хостингом в усьому пострадянському сегменті.

Кількість мобільних користувачів Вконтакте в 2014 р. зросла в 5 разів [5].

За даними дослідження AdobeSocial виходячи з кількості зареєстрованих акаунтів, в 2014 р. перше місце в світі займає соціальна мережа Facebook з 1,4 млрд акаунтів, друге місце у YouTube з 1 млрд, третє Китайська соціальна мережа Qzone (623 млн акаунтів)[5]. Крім того, у своєму дослідженні AdobeSocial зазначає, що загальна кількість зареєстрованих користувачів у 21 соціальній мережі з рейтингу майже досягла кількості людей на Землі, а Китаю належать 3 соціальні мережі, одні з найбільших і найвпливовіших [5].

1.3 Аналітика методів впливу соціальних мереж

Чим же так притягують соціальні мережі і чи так вони безпечні?

Перш за все, тим, що надають нам безмежні можливості для спілкування, на яке 21-е століття майже не залишає нам часу. Як зазначалося вище, основною метою соціальних мереж є саме спілкування з друзями, родичами і незнайомими раніше людьми, незалежно від відстаней, що розділяють співрозмовників. Люди знаходять друзів за інтересами, обмінюються фотографіями і відео, викладають їх на загальний доступ для коментування і та інше.

Великий плюс таких мереж в тому, що кожен може збирати свою команду друзів і спілкуватися приватно в своєму вузькому колі. У більшості подібних сервісів забезпечена можливість голосових і відеодзвінків через Інтернет з будь-яким абонентом, де б той не знаходився.

Соціальні мережі дозволяють нам всебічно розвиватися: ми можемо подивитися будь-який зацікавив нас художній або науково популярний фільм, послухати музику, почитати будь-яку книгу, освоїти гру на гітарі, вивчити іноземну мову, зайнятися йогою або навчитися танцювати.

Соціальні мережі-це майданчик для розвитку власного бізнесу. Абсолютно безкоштовно тут можна прорекламувати свій книжковий магазин, Новий ресторан або службу таксі, що дозволяє зробити свій бізнес відомим не тільки

у вузькому колі знайомих, а й більшості жителів свого міста або навіть країни.

Це далеко не повний перелік позитивних фактів пов'язаних з впливом соціальних мереж на наше життя. Але є і негативні моменти, до числа яких можна віднести той факт, що вони створюють вільний майданчик для вираження себе, яка автоматично перетворює все в гру.

Соціальні мережі, асоціюються з ігровим полем, де зникає тілесність гравця, його соціальний статус, вік, стать, фінансове благополуччя, в силу чого середньостатистична людина відчуває себе гравцем, рівним іншим на цьому полі. В такому випадку можна злегка збрехати, виставляючи себе за іншого, обмежити і не вказати до кінця про себе інформацію, прикрасити реальність. Соціальні мережі поступово перероджуються в дзеркала, що відображають не реальний образ індивіда, а бажаний. Виглядаючи під час спілкування з-за маски, людина відчуває себе в абсолютній безпеці, думаючи, що він настільки вдало замаскований, що ніхто ні за яких умов не зможе його дізнатися, що дозволяє вступати в різні зв'язки, показувати себе експертом у всіх сферах науки і мистецтва, в той час, як його обізнаність в даних областях обмежена поверхневими знаннями. [11]

Для самовираження соціальні мережі зручні і тим, що вони не засуджують людські слабкості, і навіть часом пропагують їх. Найбільш сильно функція самовираження реалізується в таких сервісних послугах, як блогінг, фоторепортажі, і т.п. адже вони дають кращу можливість реалізувати себе як творча особистість, не відчуваючи ні труднощів, що виникають у творчому процесі, ні позитивних вражень від успіху. Бо творчість бачиться в його очах, як загальне визнання, популярність, але, ні в якому разі, як важка праця. Проте, арена типової соціальної мережі дозволяє середньостатистичній людині стати художником, письменником, композитором, режисером. Він без праці здатний відшукати тих людей, які оцінять його творчість. Для того необхідно лише влитися в групу деякої кількості інших користувачів,

обдарувавши похвалою творчу працю одного з вищезазначених. Після цього залишається очікувати відповідного приємного відгуку на адресу своєї роботи і себе. “Можливо, в цьому можна побачити і позитивні тенденції, які можуть проявитися в тому, люди не бояться і не хочуть бути пасивними спостерігачами, вони готові проявляти спільну творчу діяльність, хочуть показати себе: «Може я і гірше, але зате я в цьому беру участь». Напевно, дана позиція могла б заслуговувати уваги, якби за всім цим не таївся стереотип середньостатистичної людини, що полягає в тому, що для приходу до успіху не потрібно прикладати особливі зусилля. Керований цими думками, користувач соціальної мережі перетворюється на людину, одержиму марнославством, який хоче, щоб його помітили і оцінили по заслугах, і не має значення, в чому саме.” [3]

Тим часом, досить часто Користувач не бачить необхідності в зворотного зв'язку, занурюючись у власну ілюзію спілкування, що полягає в розмові ні з ким. Користувач, коли він залишає пост в якомусь блозі, запускає ідею в інформаційний простір. Це дає можливість одиницям отримати в руки інструменти впливу на маси людей, але швидше рідше, ніж частіше інформація досягає споживача. Але автору цього часом достатньо, щоб відчувати себе реалізованим, успішним.

1.4 Негативний вплив соціальних мереж

“Крім того, соціальні мережі надають потужний негативний вплив на фізичний стан людини, так як мають великий потенціал адиктивності, тобто, ризиком виникнення залежності. Для цього існує кілька істотних причин.

Перша причина полягає в тому, що проведення часу в соціальній мережі дратує центри задоволення в головному мозку. Ми відчуваємо хороші емоції, кожен раз, коли читаємо комплімент під своєю фотографією, або коли хтось залишає приємний відгук про будь-якому нашій творчості.” [2]

Бажання отримати повторно ці емоції і затягує нас знову і знову на простори соціальних мереж, змушуючи проводити там все більшу і більшу кількість вільного часу.

Друга причина полягає в особливостях того, як інформація засвоюється при роботі в багатокористувацьких веб-платформах. Людина, яка сидить, наприклад, в контакті отримує багато різноманітної і різнорідної інформації дрібними дозами за маленький проміжок часу: відповів на повідомлення, побачив коментар, тут же відкрив лінійку новин, там зустрів цікаву публікацію наукової групи, почав її читати, паралельно включивши вподобану пісню, не встиг дочитати, з огляду на те, що увагу відвернуло повідомлення від знайомого, відповів і зайшов на сторінку цієї людини, щоб подивитися, що там з'явилося нового і та ін.

Соціальні мережі цілком здатні стати чимось на зразок наркотику для мозку. Людина звикає до постійного отримання будь-якої інформації і якщо несподівано цей інформаційний конвеєр зупиняється, починає відчувати ломку, в зв'язку з нестачею. Ломка виражається в тому, що нам стає складно розслабитися, коли мозок отримує мало інформації, наприклад, коли добираємося додому з роботи або працюємо на дачі. Адже наш мозок вічно турбується з приводу того, щоб переставали відправляти йому дози даних і вимагає нових порцій, тому що звик кожен день отримувати її з соціальних мереж.

Ще один негативний момент – зниження тривалості концентрації уваги. Це побічний ефект, що з'явився завдяки філософії інтеграції інформації: коли робота з одним мережевим ресурсом починає включати в себе великий функціонал, таких як спілкування, прослуховування аудіозаписів, перегляд відеороликів, обговорення і так далі. У користувача виникає спокуса приступити до всього одночасно і здійснювати відразу кілька процесів.

Це негативно позначається на можливостях нашого мислення. Стає важче тримати довго увагу, на чому або, наприклад, на читанні довгої статті. Наш розум, слідуючи отриманої в ході довгої присутності в соц. мережі звичкою, починає перескакувати з одного предмета на інший. Тому виникають складнощі з послідовним роздумом, обмірковуванням однієї проблеми: увага постійно «спливає» в сторону від поточного заняття.

Ця проблема особливо актуальна у підростаючого, нового покоління. Дитяче мислення набагато пластичніше дорослого, і, з огляду на те легше може засвоїти негативні «штампи» і «зразки» мислення.

Працюючи в режимі безперервного потоку інформації і змінюють один одного емоційних вражень, мозок сильно втомлюється, організм відчуває стрес. До того ж, під час роботи в соціальних платформах ви дивитесь в монітор, а надлишок такої діяльності сам по собі веде до стомлення, що б ви при цьому не читали.

Крім усього, соціальні мережі породили і нові види злочинної діяльності. У глобальній мережі вже досить давно прогресує добре продуманий вид шахрайства, метою якого є заволодіння вашим обліковим записом для того щоб викрасти вашу контактну і соціальну інформацію або управляти аккаунтом для подачі інформації від вашого імені. Цей вид хакерської атаки називається фішингом і полягає в тому, щоб підсунути нічого не підозрюючому користувачеві підроблену веб-сторінку з формою для введення логіна і пароля. Після передачі даних користувач може перейти вже до справжньої сторінки соціальної мережі, ось тільки вся секретна інформація вже буде відправлена зловмиснику.

Схема підробки досить проста. На пошту або месенджер користувача приходить повідомлення з пропозицією завести дружбу в соціальній мережі, скачати цікаву гру, послухати музику, подивитися новий відеоролик. При цьому добре замаскована посилання, що вказує на ресурс, веде до точної копії

сторінки який-небудь соціальної мережі. Все це може бути закамуюфльовано настільки професійно, що користувач навіть не здогадається про обман і сам передасть шахраєві всі свої дані [9].

Ще один вид найбільш часто скоєних злочинів в мережі-дифамація, під якою розуміються образливі або образливі фрази, уїдливі або іронічні жарти особистого характеру, використання конфіденційної інформації, розголошення якої може завдати шкоди, публікація фотографій, які можуть мати негативний вплив на репутацію сфотографованої особи, оголошення особистих даних, таких як ім'я та прізвище або інша інформація, з якої буде зрозуміло, про кого йде мова, створення груп або тим, що містять наклепницькі висловлювання. Деякі вважають, що " Інтернет – територія віртуальна, а значить – нічия. Що хочу, те й пишу"» Але це тільки ілюзія всездозволеності і безкарності, яка тягне за собою реальну відповідальність.

1.5 Цілі і завдання дипломної роботи

Основна ідея та самоціль кваліфікаційної роботи магістра – дослідження та розробка системи, що спрямована допогати людям в питаннях фільтрованого пошуку та реалізації технології веб-скрейпінгу.

Без наявності розробки, люди мають вручну здійснювати багато пошукових, аналітичних та порівняльних операцій. Наявність теоритичних та практичних висновків цієї роботи - дуже спрощує операцію, що трапляється замість тривіального ручного пошуку.

РОЗДІЛ 2. ОСНОВОПОЛОЖНІ ПРИНЦИПИ ТЕХНОЛОГІЇ ВЕБ-СКРЕЙПІНГА

2.1 Технологія веб-скрейпінгу

Веб-скрейпінг - технологія для синтаксичного перетворення HTML-сторінок в більш зручні для споживання форми. Технологія веб-скрейпінга є вузько частиною більш загального поняття web mining (WM).

WM - програмне забезпечення, призначене для здобуття знань з даних, зазвичай з документів і супроводжуючих їх гіперпосилань, з відомостей про користувачів і їх активності, зафіксованих в різних журналах, в загальному, з усього, що може бути доступно в Мережі. Перші публікації на тему WM датуються серединою 90-х років, коли мирно співіснували два підходи: процесний, що представляє WM як послідовність виконуваних завдань, і дата-центричний, прив'язаний до типам даних. Надалі переважне поширення набув другий. Сьогодні під WM розуміють сукупність методів автоматичного систематичного обходу мережі інтернет з метою збору необхідних даних про компанії і людей для вироблення інформації та прийняття рішень.

Дата-центричність означає акцент WM на роботі з трьома джерелами інформації, кожному з яких відповідає вид WM:

- Дані про дії користувачів, від журналів серверів до відстеження звернень до браузеру. Web Usage Mining (WUM) або навантажувальний WM.

					КНУ.РБ.123.20.01.ВС				
Зм.	Арк.	№ документа	Підпис	Дата	ТЕХНОЛОГІЇ	Літера		Аркул	Аркулів
Розробив		Слюсар							
Перевірив		Купін							
Н. контроль		Кузнєцов				KI-21M			
Затвердив		Купін							

• Веб-графи, що описують прямі зв'язки між сторінками WWW. У веб-графі вершини - це сторінки WWW, а дуги - гіперпосилання між ними. За графу встановлюються зв'язки між сторінками, людьми і будь-якими іншими об'єктами. WCM став областю активних досліджень, і основні складності тут викликані гетерогенністю веб-даних і їх низькою структуризацією, що утрудняє виділення цільової інформації. Крім того, в WCM необхідно вирішувати ряд специфічних завдань:

1. витяг структурованих даних з веб-сторінок з використанням методів машинного навчання і нейронних мереж;
2. формування процедур уніфікації форматів представлення даних і їх інтеграції з різних джерел;
3. виділення оцінок продуктів і послуг у відгуках, що розміщуються на форумах, в блогах і чатах.

В WCM майже завжди виконується процедура перекладу даних з форми, призначеної для читання людиною, в форму, зручну для обробки комп'ютером. Така процедура називається data scraping. Через деякий час ця технологія модифікувалася і стала називатися web scraping.

Повністю автоматизована генерація можлива поки тільки на експериментальному рівні, а найвищим досягненням веб-скрейпінга є аналізатори веб-сторінок з елементами штучного інтелекту на базі систем комп'ютерного зору і машинного навчання .

Веб-скрейпінг працює з трьома типами інформації: структурованими, неструктурованими і квазіструктурованими.

Для роботи зі структурованими даними досить застосувати службові процедури спочатку обходу сторінок, потім генерації і виконання пакувальника, а потім можна переходити до аналізу вмісту сторінки.

Складніше з квазіструктурованими даними. Прикладом таких даних може бути граф відвідувань сайту. Для роботи з квазіструктурованими даними

запропоновані спеціальні мови, такі як NEXIR ELOG, призначені для програм-пакувальників. З їх допомогою описується процедура виділення даних, що завершується створенням об'єктної моделі даних (ОЕМ).

неструктурованих ДМ покликаний полегшити сприйняття користувачам великих масивів текстів. Виділяють кілька типів такого роду операцій:

- відстеження тематики - оцінка області інтересів користувача;
- згортка - створення резюме документів;
- ранжування - впорядкування документів і їх розподіл за заздалегідь визначеними категоріями;
- кластеризація - об'єднання схожих документів в групи;
- візуалізація даних - вирішення проблеми комунікації користувача з даними.

Близько дюжини компаній виробляють сьогодні інструменти для WCM у вигляді традиційних завантажуються комерційних і вільно-распространяемых програм і хмарних сервісів. Однак всі інструменти мають деяку схожу структуру.

Так, будь-який інструмент для веб-скрейпінга включає в себе наступні елементи:

- краулінга;
- парсинг;
- зберігання інформації.

За останнє десятиліття інформація стала основним ресурсом для розвитку бізнесу, а Інтернет є основним джерелом, де 5 мільярдів користувачів щомиті генерують нові дані. Витягуючи і аналізуючи ці веб дані, компанії розробляють свої бізнес-стратегії і досягають поставлених цілей. Однак зібрати і витягти такий великий обсяг даних непросто; особливо для тих, хто все ще думає, що кнопка «Експортувати в Excel» існує або ручна обробка даних - це єдине рішення.

Веб скрапінг дозволяє компаніям автоматизувати процеси збору веб даних за допомогою ботів або автоматизованих скриптів, які називаються веб-сканерами і завантажувати ці дані в форматі Excel, CSV або XML для подальшої аналітики.

Зараз я хочу показати декілька інструментів для веб-скрабінгу:

1) Scraper Api- дозволяє отримати вміст HTML з будь-якої сторінки за допомогою виклику API. З Scraper API можна з легкістю працювати з браузерами і проксі-серверами і обходити перевірочні код CAPTCHA. Єдине на що необхідно зосередитися це перетворення веб-сайтів в цінну інформацію. З цим інструментом практично неможливо бути заблокованим, так як він змінює IP-адреси при кожному запиті, автоматично повторює невдалі спроби і вирішує капчу за вас.

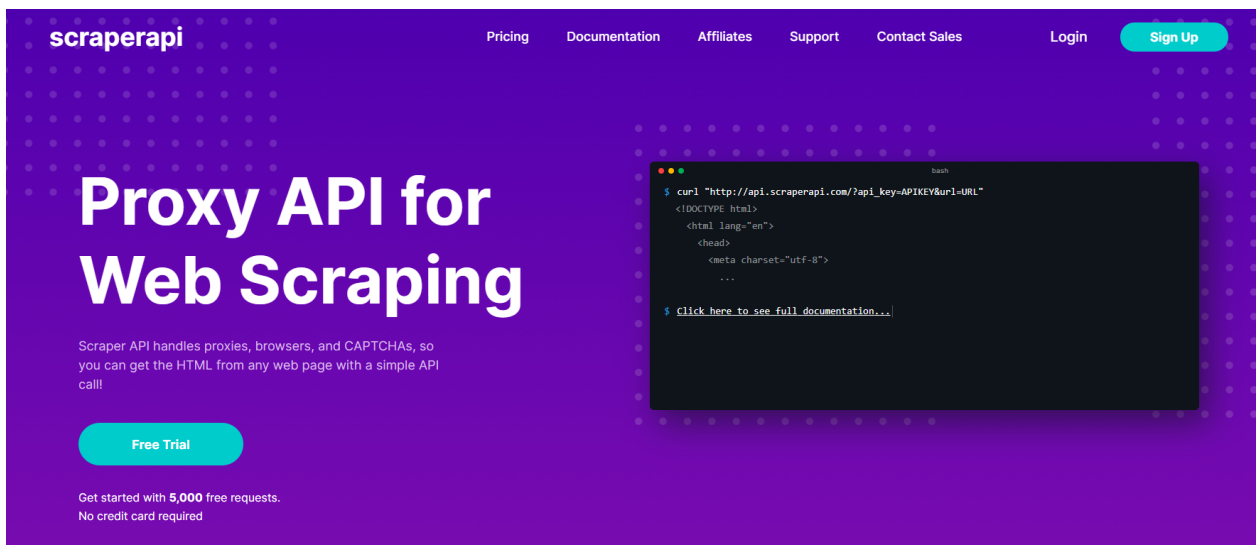


Рисунок 2.1 – Офіційна сторінка scraperapi

2) Ostorparse - це безкоштовний інструмент призначений для веб скрапінга. Він дозволяє отримувати дані з інтернету без строчки коду і перетворювати веб-сторінки в структуровані дані всього за один клік. Завдяки автоматичної ротації IP-адрес для запобігання блокування та можливості планування подальшого скрапінга цей інструмент є одним з найефективніших.

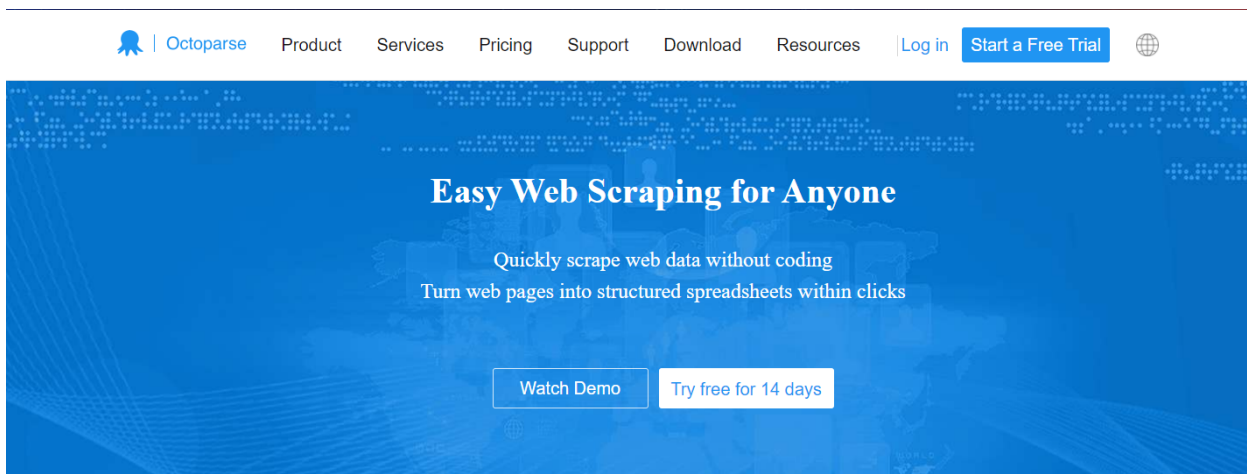


Рисунок 2.2 – Офіційна сторінка Octoparse

3) DataOx - справжній експерт в області скрапінга веб-сторінок. Інструменти Предлагаються компанією DataOx забезпечують великомасштабні збори даних і надають комплексні рішення адаптовані до потреб клієнтів. Цій компанії можуть довіряти як стартапи, що створюють продукти на основі даних, так і великі підприємства, які вважають за краще доручати збір власних даних професіоналам.

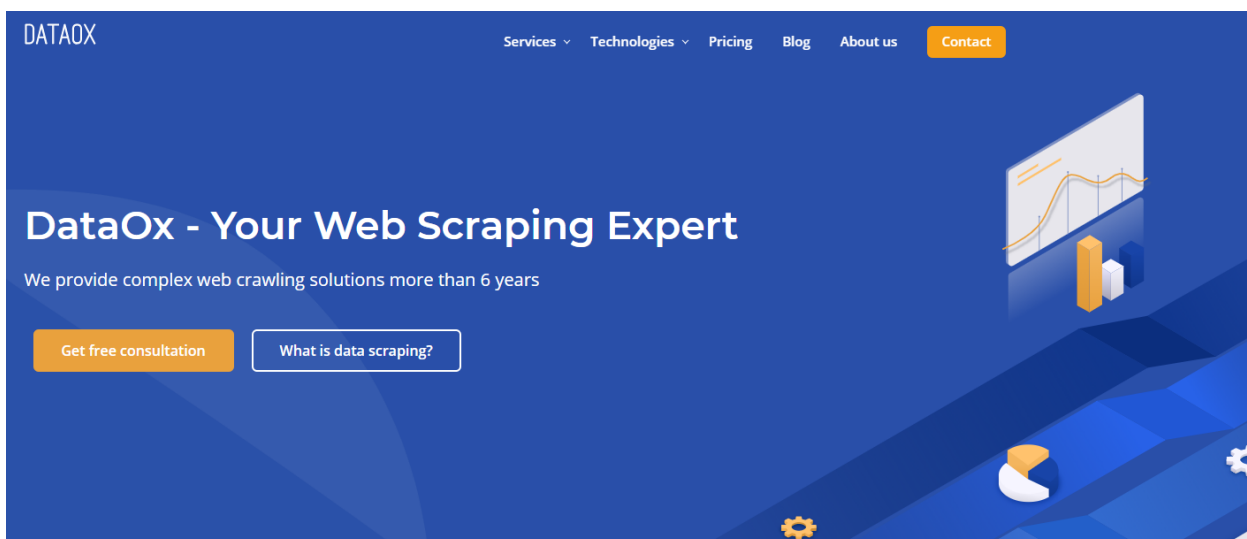


Рисунок 2.3 – Офіційна сторінка DataOx

2.2 Краулінг

2.2.1 Основні відомості про пошукових роботах

Краулінг - це процес збору пошуковим роботом нових сторінок для додавання в індекс пошукових систем. Сканування - початковий етап, дані збираються тільки для подальшої внутрішньої обробки і не відображаються в результатах пошуку.

При скануванні роботи переглядають сторінки і виконують перехід за що містяться, як і звичайні користувачі. При цьому в їх роботі є послідовність. Це дозволяє одночасно обробляти величезні масиви даних.

Пошуковий робот або краулер - це програма, яка реалізує технологію краулінга, яку розробили для парсингу веб-сторінок. Краулінга дозволяє автоматизувати такі функції, як збір особливих видів інформації, наприклад, адрес електронної пошти.

Функціонування пошукового робота будується за тими ж принципами, за якими працює браузер. Павук відвідує сайт, оцінює вміст сторінки, переносить їх у базу пошукової системи, потім по посиланнях переходить на інші ресурси, повторюючи один і той же алгоритм дій. Результат цих пошуків - перебір веб-ресурсів в суворій послідовності. Потрапляючи на веб-ресурс, павук знаходить призначений для нього файл robots.txt, і подальші пошуки повинен проводити згідно з правилами, встановленими в robots.txt.

При використанні технології краулінга необхідно ставити наступні характеристики:

- глибина - вказує, на скільки кліків від стартової сторінки ви хочете зануритися;
- кількість паралельних завантажень сторінок;
- пауза між сторінками для полегшення навантаження на сервер, нерідко власник ресурсу визначає значення цього параметра в файлі robots.txt;

краулер не повинен сильно навантажувати сервер сайту, інакше такий краулінг сприймається як кібер атака і призводить до блокування і сайту, і краулер.

2.2.2 Можливості бібліотеки Scrapy

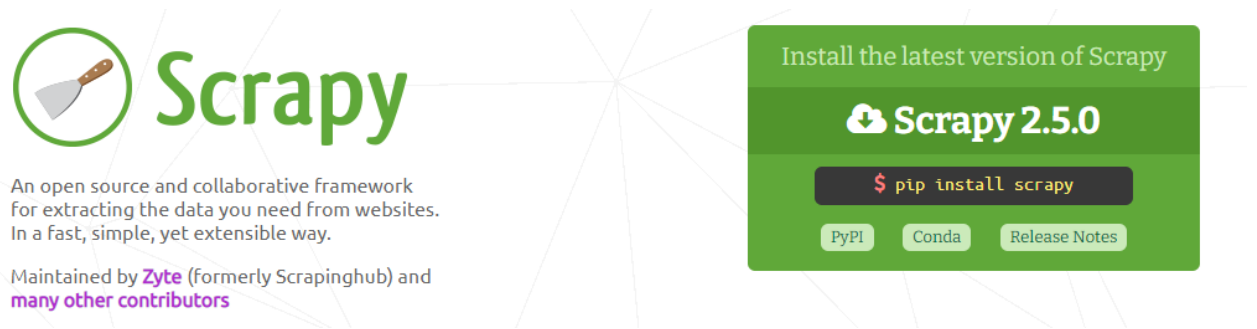


Рисунок 1.1 – Офіційна сторінка бібліотеки Scrapy

Scrapy - це прикладна середовище для сканування веб-сайтів і вилучення структурованих даних, яка може використовуватися для широкого спектра корисних додатків, таких як аналіз даних, обробка інформації або архівування даних.

Scrapy спочатку був розроблений для перегляду веб-сторінок, але також його можна використовувати для отримання даних за допомогою API. Scrapy дає ряд крутих функцій для легкого і ефективного очищення, таких як:

- Вбудована підтримка вибору і отримання даних з джерел HTML / XML з використанням розширених селекторів CSS і виразів XPath, а також допоміжні методи з використанням регулярних виразів;
- Інтерактивна оболонка консолі IPython для обробки CSS і XPath виразів;
- Вбудована підтримка генерації експорту фідів в декількох форматах (JSON, CSV , XML) і їх зберігання в декількох бекенда (FTP, S3, локальна файлова система);

- Надійна підтримка кодування і автовизначення для роботи з зовнішніми, нестандартними і непрацюючими оголошеннями кодування;
- Підтримка розширюваності, що дозволяє вам підключати свої власні функції, використовуючи сигнали і чітко визначений API (сполучна ПО, розширення та конвеєри);
- Широкий спектр вбудованих розширень і проміжного програмного забезпечення для обробки:
 - а) cookies та обробка сесії ;
 - б) функції HTTP, такі як стиснення, аутентифікація, кешування;
 - в) підміна призначеного для користувача агента;
 - г) robots.txt;
 - д) обмеження глибини сканування.

Scraping складається з двох етапів:

- Систематичний пошук і завантаження веб-сторінок;
- Витяг даних з веб-сторінок .

Створити пошукового робота з нуля можна за допомогою різних модулів і бібліотек, які надає мова програмування, проте в подальшому - у міру зростання програми - це може викликати ряд проблем. Наприклад, вам знадобиться переформатувати витягнуті дані в CSV, XML або JSON.

Також ви можете зіткнутися з сайтами, для аналізу яких необхідні спеціальні настройки і моделі доступу.

Тому краще відразу розробити робота на основі бібліотеки, яка усуває всі ці потенційні проблеми. Для цього використовуються Python і Scrapy.

Scrapy - одна з найбільш популярних і продуктивних бібліотек Python для отримання даних з веб-сторінок, яка включає в себе більшість загальних функціональних можливостей. Це означає, що вам не доведеться самотійно прописувати багато функцій. Scrapy дозволяє швидко і без зусиль створити «веб-павука».

Пакет Scrapy (як і більшість інших пакетів Python) можна знайти в PyPI (Python Package Index, також відомий як pip) - це підтримуваний співтовариством репозиторій для всіх вийшли пакетів Python.

Пошукові роботи дозволяють отримати інформацію про ряд продуктів, отримати великий обсяг текстових або кількісних даних, витягти дані з сайту без офіційного API і багато іншого.

2.2.3 Robots.txt

Файл robots.txt – файл у корні директорії сайту, туди записують функції пошукових роботів. Такі функції можуть забороняти до індексації частини сайту, тому треба дотримуватись часових інтервалів, блокувати неважливі зображення і скрипти. Після вивчення файлу, робот відвідує головну сторінку, а з неї переходить по посиланнях, просуваючись в глибину.

Під індексацією розуміється аналіз і занесення в базу пошукових систем сторінок сайту, після якого сайт видається в результатах пошукових запитів. В першу чергу, при роботі з robots.txt, роботи перевіряють наявність записів, що починаються з User-agent, такі записи укладають між собою інструкції для певного робота. Однак існують пошукові роботи, які не дослухаються до

загальних забороняє правилами в robots.txt, тому потрібно прописувати блоки команд для таких роботів окремо.

Після звернення до робота, за допомогою директиви user-agent, слідує команди, що забороняють або дозволяють звернення до певних розділів сайту. Для заборони до перегляду використовується директива Disallow, а для дозволу Allow, параметрами, яких є URL розділу або сторінки сайту.

В основному директива Disallow застосовується для заборони відвідування павуком службових розділів, конфіденційної інформації, а також для видалення дублів сторінок. Під дублями сторінок розуміється виняток звернення до однієї і тієї ж сторінки за різними адресами.

Менш затребувана директива Allow більшою мірою призначається для роботів пошукових систем, вона допомагає правильно проіндексувати сайт. Використовуючи директиву allow, ви позначаєте роботу сторінки необхідні до індексації. Allow рідше використовується в порівнянні з disallow, це пов'язано з тим, що за замовчуванням всі сторінки сайту повинні індексуватися.

Дуже важливою і корисною директивою в robots.txt, при розгляді теми веб-скрейпінга, є crawl-delay, з її допомогою власник сайту може задати мінімальний проміжок часу між закінченням завантаження однієї сторінки і початком завантаження наступної. Crawl-delay використовується, коли сервер перевантажений і не встигає обробляти запити на завантаження сторінок.

Так само існують директиви host - вказує основне дзеркало сайту, і sitemap - вказує URL карти сайту, яка відображає структуру побудови сайту і розташування контенту на ньому. Дзеркалом сайту називається його повний або частковий дублікат.

2.3 Парсинг

Робити граматичний розбір або структурувати - дії корисні і потрібні. Мовою всіх, хто працює з даними на сайтах це слово має свій відтінок.

Парс - збирати і систематизувати інформацію, розміщену на певних сайтах, за допомогою спеціальних програм, що автоматизують процес.

Якщо ви коли-небудь задавалися питанням, що таке парсер сайту, то ось він відповідь. Це програмні продукти, основною функцією яких є отримання необхідних даних, які відповідають заданим параметрам.

Чи законно використовувати парсинг

Після з'ясування що таке парсинг, може здатися, що це щось, що не відповідає нормам чинного законодавства. Насправді це не так. Законом не переслідується парсинг. Зате заборонені:

- злом сайту (тобто отримання даних особистих кабінетів користувачів і т. П.);
- DDOS-атаки (якщо на сайт в результаті парсинга даних лягає занадто високе навантаження);
- запозичення авторського контенту (фотографії з копірайтами, унікальні тексти, справжність яких засвідчена у нотаріуса і т. п. краще залишити на їх законному місці).

Парсинг законний, якщо він стосується збору інформації, що знаходиться у відкритому доступі. Тобто все, що можна і так зібрати вручну.

Парсери просто дозволяють прискорити процес і уникнути помилок через людський фактор.

Інша справа, як власник свіжозібраної бази розпорядиться подібною інформацією. Відповідальність може наступити саме за наступні дії.

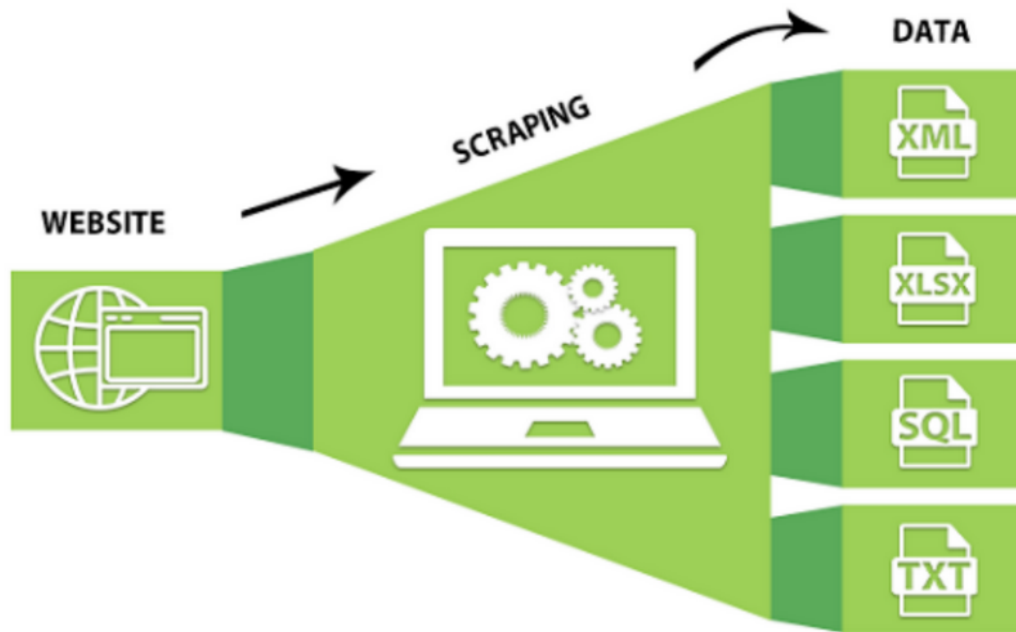


Рисунок 1.2 – Парсинг даних

Парсер - це програма, створена для автоматизації обробки поста отриманої с веб-сайту інформації. Іншими словами, парсер - це програма з аналізу та перетворення тексту з метою виділити з нього певні фрагменти або видалити зайве. На відміну від людини, парсер швидко і безпомилково відокремлює потрібну і відкидає зайву інформацію з html документа, і ефективно упаковує результат в певному форматі.

Існує кілька способів отримання даних:

- Аналіз DOM дерева html сторінки. DOM - це уявлення HTML / XML документа у вигляді дерева об'єктів, яке дозволяє скриптам змінювати вміст і структуру документа. Дані в такому підході виходять з атрибуту елемента дерева або при відсутності таких, спускаючись вниз по DOM дереву. Отримані дані можуть бути будь-якої структури, а для отримання значення елемента, достатньо знати його розташування. Однак скрипт, який використовує даний метод потрібно прив'язувати до движку, а при зміні розташування елемента втрачається доступ до нього;

- Парсинг рядків. Цей спосіб парсинга має вузьку сферу застосування, так як отримання даних відбувається шляхом парсинга окремих рядків, що, в свою чергу, можливо лише в разі чітко фіксованого формату даних;
- Використання регулярних виразів. Цей метод, в основному, використовують для вирішення невеликих завдань або для написання власних процедур;
- XML парсинг. Ще одним підходом є розгляд HTML як XML дані. Причина в тому, що HTML рідко буває дійсним, під валідність HTML, в даному випадку, розуміється відповідність XML стандартам. Бібліотеки, які реалізували такий підхід, більше часу приділяли перетворенню HTML в XML, ніж безпосередньо парсингу даних;
- Візуальний підхід. Суть підходу в тому, щоб користувач міг без використання програмного мови або API налаштувати систему для отримання потрібних даних будь-якої складності і вкладеності. Однак такий підхід перебуває на початковій стадії розвитку.

Проблеми при парсингу HTML даних:

- використання JavaScript / AJAX / асинхронних завантажень елементів сторінок дуже ускладнюють написання парсерів;
- різні движки для рендеринга HTML можуть видавати різні DOM дерева;
- великі обсяги даних вимагають писати розподілені парсери, що тягне за собою додаткові витрати на синхронізацію.

Переходимо до того, навіщо ж це може знадобитися. Тут відкривається широкий простір для дій.

Основна проблема сучасного Інтернету - надлишок інформації, яку людина не в змозі систематизувати вручну.

Парсинг використовується для:

1)Аналізу цінової політики. Щоб зрозуміти середню вартість тих чи інших товарів на ринку, зручно використовувати дані по конкурентах. Однак якщо це сотні і тисячі позицій, зібрати їх вручну оперативно неможливо.

2)Відстеження змін. Парсинг можна здійснювати на регулярній основі, наприклад, щотижня, виявляючи на що підвищилися ціни в середньому по ринку і які новинки з'явилися у конкурентів.

3)Наведення порядку на своєму сайті. Так, так теж можна. І навіть потрібно, якщо в інтернет-магазині кілька тисяч товарів. Знайти неіснуючі сторінки, дублі, неповне опис, відсутність певних характеристик або невідповідність даних по складських залишків того, що відображається на сайті. З парсером швидше.

4)Наповнення карток товарів в інтернет-магазині. Якщо сайт новий, рахунок зазвичай йде навіть не на сотні. Вручну на це піде недозволено кількість часу. Часто використовують парсинг з іноземних сайтів, переводять отримані тексти автоматизованим методом, після чого отримують практично готові опису. Іноді те долають з російськомовними сайтами, а отримані тексти змінюють за допомогою синонімайзер, але за це можна отримати санкції від пошукових систем.

5)Отримання баз потенційних клієнтів. Існує парсинг, пов'язаний зі складанням, наприклад, списку осіб, які приймають рішення, в тій чи іншій галузі і місті. Для цього може застосовуватися особистий кабінет на сайтах пошуку роботи з доступом до актуальних і архівних резюме. Етичність подальшого використання подібної бази кожна компанія визначає самостійно.

2.4 Способи зберігання інформації

2.4.1 Реляційна база даних

Реляційна база даних - це база даних, яка зберігає дані в таблицях. Більшість систем управління базами даних є системами керування базами даних. Кожна таблиця в реляційній базі даних містить рядки і стовпці. Таблиці бази даних включають первинний ключ, він надає ідентифікатор для кожного рядка в таблиці. Ключ може бути призначений колонки. Значення первинного ключа з однієї таблиці можливо присвоїти полю в рядку іншої таблиці. Значення, які були імпортовані з інших таблиць, називають зовнішніми ключами.

2.4.2 Формат JSON

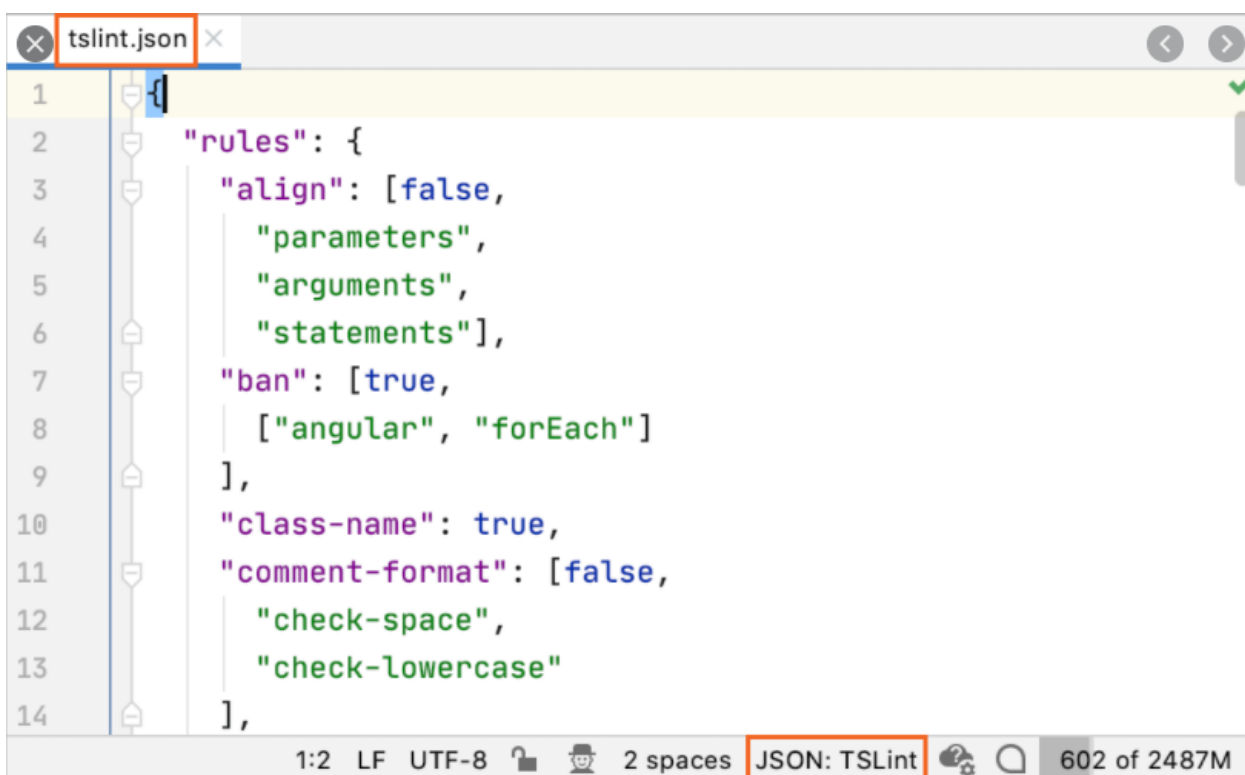


Рисунок 1.3 – Приклад використання формату JSON

Формат даних не має будь-яких обмежень, але, найчастіше, використовується JSON. JSON - текстовий формат, повністю незалежний від мови реалізації,

визнаний простим способом для зберігання та передачі структурованих даних.

Чому його використовують:

- пропозиції не важко зчитуються;
- створюються структури JSON за допомогою більшості мов програмування;
- на його основі можна зібрати базу даних, наприклад, використовуючи утиліту `jl-sql`, яка приймає на вхід потік JSON-об'єктів, розділених символом нового рядка, і поміщає в базу даних, використовуючи SQL запити. SQL - формальний непроцедурного мова програмування застосовується в реляційної базі даних, керованої відповідною системою управління базами даних;
- Найбільш часто поширене використання JSON - пересилання даних по мережі. Так, наприклад, виконавши скрипт скрейпера на сервері і отримавши результат у форматі JSON, його можна з легкістю відправити замовнику по інтернету для подальшого використання.

Файл JSON має два способи представлення як набір пар ключ-значення або як упорядкований набір значень, де спосіб реалізації уявлень залежить від мови реалізації.

Ще однією сильною стороною JSON формату є його встроєність в більшість сучасних браузерів, що робить використання формату зручним і прискорює обробку JSON даних.

РОЗДІЛ 3. СУЧАСНІ ІНСТРУМЕНТИ ДЛЯ СКРЕЙМІНГА

Розгляд теми веб-скрейпінга неможливо без вивчення стану справ на ринку готового ПО. На жаль вибір інструментів не такий широкий, як хотілося б, до того ж більшість інструментів призначений для англomовних користувачів, що робить скрутним вибір програми для роботи, однак все ж є гідні пропозиції.

Для початку слід розділити дві основні категорії продуктів. До першої категорії відносяться продукти, що дозволяють створити на своїй базі повноцінний скрейпер. Такі програми найчастіше є фреймворками або розширеннями для браузерів, рідше веб-сервісами. У таких продуктах заздалегідь визначений мову програмування і існує ряд обмежень. Плюсами таких рішень, є можливість використання різних бібліотек і найголовніше, якість скрейпінга залежить тільки від користувача. До другої категорії можна віднести повноцінні програми, що дозволяють користувачам скрейпі дані не вдаючись до програмування. Існують різні формати таких програм. Веб-скрейпер може представляти із себе як просте розширення в браузері, так і онлайн сервіс. Розглянемо провідні продукти для веб-скрейпінга.

Відмінним ПО для веб-скрейпінга, на сьогодні, є OutWit Hub від компанії OutWit Technologies. Дана програма є розширенням браузера Firefox. Така програма може служити для скрейпінга невеликих обсягів даних, скоріше для домашнього використання. Програма є найпростішою для використання серед своїх конкурентів, однак навіть її використання неможливо без знань основ веб-програмування. У даній програмі користувач грає важливу роль, так як він задає теги, в яких укладена шукана інформація, визначає критерії.

					КНУ.РБ.123.20.01.ВС			
Зм.	Арк.	№ документа	Підпис	Дата	ВСТУП	Літера	Аркус	Аркусів
Розробив		Слюсар						
Перевірив		Купін						
Н. контроль		Кузнєцов				KI-17-1		
Затвердив		Купін						

пошуку і види посилань, за якими буде проходити скрейпер. Також користувач визначає формат, в якому збережеться знайдена інформація. Для якісного скрейпінга необхідно докласти чимало зусиль в налаштуванні програми.

Конкурентом, розглянутого програмного забезпечення, є продукт під назвою OpenRefine від однойменної компанії. OpenRefine дуже популярний серед журналістів і статистиків. Про простоті використання говорити не доводиться, тому знадобиться не один день ознайомлення. Розуміючи те, що основними користувачами продукту не є програмісти з досвідом веб-розробки, компанія вирішила випустити повноцінну книгу, яка є своєрідним

гідом по програмі. Ще однією перевагою інструменту є безліч розширень, які значно збільшують можливості програми. Можливо, ці фактори стали ключовими в боротьбі за таку велику аудиторію.

Для вирішення завдань, пов'язаних з скрейпінгом даних немає універсального інструменту. Необхідно попередньо вивчити можливості пропонованих скрейперів і, зробивши вибір інструменту, доведеться витратити багато часу на його освоєння. Програми для веб-скрейпінга хоч і дозволяють піти від чистого програмування, але все-таки мало придатні для середньостатистичних користувачів.

3.1 Обмеження на використання скрейперів

Веб-скрейпінг - це витяг контенту з веб-сайтів. У свою чергу у будь-якого сайту є свій правовласник або адміністратор, який встановлює правила користування ресурсом. Тому необхідно взаємодіяти з власником сайту або контенту перш ніж запускати скрейпер. У популярних сайтів сотні тисяч відвідувачів за добу, тому неможливо кожному користувачеві, що бажає скачати інформацію, зв'язатися безпосередньо з власником домену. Однак

нехтувати правами автора можна, адже порушення авторських прав може спричиняти цивільно-правову, адміністративну та / або кримінальну відповідальність.

Власники сайтів, зацікавлені в захисті своїх прав, сповіщають користувачів в наступному форматі: знак копірайту © + ім'я правовласника (автора) + рік першого опублікування. Однак можливі й інші формати. Наприклад, © 2018 YouTube, LLC.

Так само зустрічаються сайти, на яких є для користувача угоди або заяви про дотримання конфіденційності - це, як правило, приватні веб-сайти закритого або комерційного характеру. Такі сайти публікують вельми об'ємні правила користування своїм ресурсом. Вони обумовлюють доступ до сайту, порядок його використання, тут містяться численні попередження, мета яких - захист інформаційних технологій і захист товарного знака, різноманітна захист від копіювання та взагалі захист сайту в плані авторських прав і охорони інтелектуальної власності.

Мінус даних способів оповіщення про обмеження використання контенту полягає в тому, що при автоматизованому копіюванні даних, наприклад, з використанням скрейпера, користувач буде сповіщений тільки після вчинення правопорушення, а якщо бути точніше, під час виконання парсинга сторінки. Для запобігання таких випадків в кореневій директорії сайту вебмастера поміщають файл robots.txt.

3.2 LinkedIn – професійна соціальна мережа

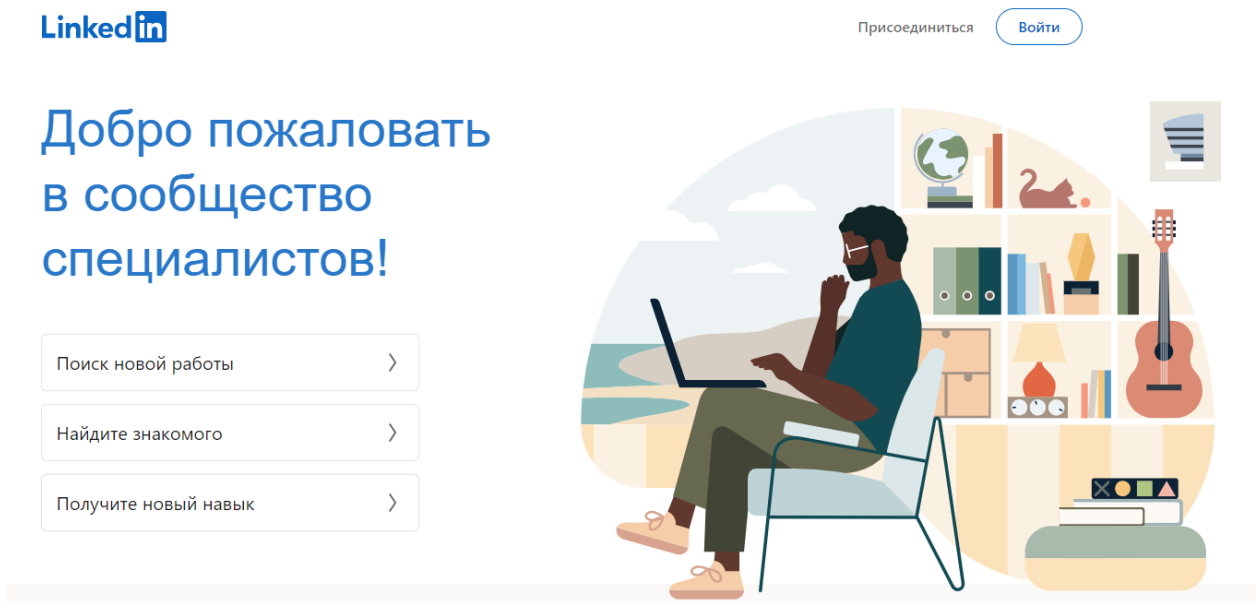


Рисунок 3.1 – Сторінка авторизації LinkedIn

LinkedIn - це найбільша в світі мережа професійних контактів, розташована в Інтернеті. За допомогою LinkedIn ви зможете знайти потрібну роботу або стажування, налагодити і зміцнити професійні взаємини, а також оволодіти навичками, необхідними для досягнення успіху у вашій кар'єрі. Доступ до LinkedIn можна отримати через настільну та мобільну версію веб-сайту, мобільний додаток LinkedIn або додаток LinkedIn Lite для Android.

Максимально заповнений профіль в LinkedIn допоможе вам знаходити нові можливості за рахунок демонстрації вашої унікальної професійної біографії за допомогою відомостей про досвід роботи, навичках і освіті.

Ви також можете використовувати LinkedIn для проведення офлайн-заходів, участі в групах, написання статей, розміщення фотографій і відеороликів і для інших цілей.

LinkedIn вважається гігантом соціальних мереж. Він менше популярний, мабуть, через професійний характер його користувачів. Проте LinkedIn продовжує надавати сильний вплив на шукачів роботи, торгових марок, рекрутерів та галузь.

Як і більшість соціальних мереж, LinkedIn заохочує своїх користувачів створювати особистий профіль - сторінку, де вони можуть продемонструвати інформацію про свій досвід роботи та навички.

На відміну від багатьох інших соціальних мереж, важливо заповнити свій профіль на LinkedIn, особливо якщо ви використовуєте його для пошуку роботи. LinkedIn оцінює "повноту профілю" за шкалою 0-100%. Чим повнішим є ваш профіль, тим більша ймовірність показу у результатах пошуку. Наприклад, демонстрація того, що ви володієте такими програмами, як Final Cut Pro та Photoshop, може допомогти потенційним роботодавцям знайти ваш профіль у розширеному пошуку за цими ключовими словами. Корисно.

Щоб зробити ваш профіль на 100% заповненим, LinkedIn рекомендує включити таку інформацію.

1. галузь та поштовий індекс
2. поточне місце роботи з описом роботи
3. дві інші попередні роботи
4. освіта
5. принаймні п'ять навичок (додатків)
6. зображення профілю
7. принаймні 50 контактів
8. резюме

Групи LinkedIn - це групи, де професіонали та експерти можуть обмінюватися вмістом, запитувати поради, шукати чи запрошувати на роботу та спілкуватися з іншими. Приклади варіюються від "Як розпочати роботу - бізнес" до "Cal | UC Berkeley Alumni Association".

З іншого боку, групи LinkedIn не слід плутати з бізнесом. Coca-Cola має групу "Поточні та колишні співробітники Coca-Cola", але діяльність компанії перелічена на сторінці "Coca-Cola Company".

На вибір є понад 1,3 мільйона груп, і ви обов'язково знайдете групи, які відповідають вашій області та інтересам. Майте на увазі, що багато груп потребують схвалення, перш ніж адміністратор може дозволити вам приєднатися. Однак приблизно третина груп не потребує реєстрації і мають позначку "відкрито". Так само, як у вас є особиста сторінка, багато компаній також воліють представляти себе на LinkedIn. Подібно до сторінок для бізнесу у Facebook, ви можете стежити за діяльністю компанії та новинами на LinkedIn.

LinkedIn Люди ▾ Mykyta Sliusar 🔍 Присоединиться Войти

Mykyta Sliusar
Студент в уч. заведении ШАГ и КНУ. Back End разработчик.
Кривой Рог, Днепропетровская область, Украина · 50 контактов

Присоединиться

Space Site
КНУ

Другие участники также смотрели

- Tatiana Leksikova**
Freelance Recruiter at Freelancer
Украина
- Alina Bashkatova**
Looking for Business Analyst, Business Data Analyst, Business Analyst/Product Owner
Украина
- Oleksii Khrypko**
Senior Software Engineer at Sigma Software
Украина
- Ruslana Pyvovarova**
HR at Technorely
Харьков
- Mykyta Shupyliuk**
React.js developer
Украина
- Ihor Marshuba**
Инженер-программист – Харитрон-аркос Лтд
Украина
- Maksym Chuhaievskyi**
Java Technical Lead at SoftServe
Украина
- Yevhen Rohozianskyi**
.Net Developer - Softjourn, Inc
Украина

Общие сведения

Менеджер по продажам Big Money University

Действия

 Twenty years ago, Google started with a goal to organize the world's information and make it universally accessible and useful. Here are 20...
Отмечено как понравившееся участником Mykyta Sliusar

Рисунок 3.2 – Моя сторінка на сайті LinkedIn

Сторінки компаній містять загальну інформацію, таку як огляд бізнесу, список працівників та посилання на друковану компанію. Багато компаній також воліють розміщувати вакансії на своїх сторінках, а деякі навіть заохочують кандидатів підключатися до LinkedIn, що забезпечує зручні можливості для цього.

Якщо ви вирішите підписатися на компанію, ви можете знайти оновлення на домашній сторінці компанії LinkedIn, а також новини про інші контакти. Наприклад, Mashable зазвичай намагається розміщувати ділові статті на LinkedIn, оскільки його зміст здається більш доречним для аудиторії мережі. Компанії також використовують LinkedIn для розміщення своїх оголошень, таких як нові придбання, нові співробітники або зміни в політиці компанії. Однак LinkedIn застерігає від розсилки повідомлень: "Компанії, які надто часто публікують незначні оновлення, підлягають суворому контролю з боку LinkedIn і ризикують видалити їх сторінку".

Пошук роботи та підбір персоналу - одні з найцінніших можливостей LinkedIn. Все більше компаній заохочують кандидатів шукати роботу через

LinkedIn через високу довіру до соціальної мережі та простоту використання.

Опыт работы



Веб-разработчик

Space Site

сент. 2018 – февр. 2020 · 1 год 6 месяцев

Киев, Украина

Работал на должности Front End разработчика, получив достаточно опыта и знаний перешел на фриланс для дальнейшего профессионального роста в этой сфере.

Образование



КНУ

Не законченное высшее образование · Разработка программного обеспечения · 4.6

2017 – 2022

Деятельность и сообщества: Опыт разработки кроссплатформенных приложения более 3 лет. Больше нравится работать с Front End разработкой

Мой профиль на Upwork: <https://www.upwork.com/freelancers/~016d8b9840ad80b30c>



ШАГ

Джуниор · Разработка программного обеспечения

2017 – 2020

Рисунок 3.3 – Це інформація про місця навчання та роботи людини

Перейдіть на вкладку "Вакансії", де ви знайдете варіанти для кандидатів. Виконайте розширений пошук доступних вакансій за ключовим словом, іменем, місцезнаходженням, назвою компанії, класом оплати праці та галуззю (пошук "програміста" в радіусі 50 миль від Манхеттена дав 488 результатів). Зберігайте робочі місця для детального перегляду пізніше і навіть зберігайте результати пошуку, щоб перевірити оновлені результати пізніше.

Як роботодавець ви можете опублікувати доступну роботу в LinkedIn за 295 доларів США протягом 30-денного періоду. (Оптові пакети доступні для вигідніших пропозицій). Після розміщення ці завдання з'являться не лише в результатах пошуку, але й у розділі «Кар'єра» на сторінці вашої компанії.

Нарешті, рекрутери можуть «знаходити таланти» в LinkedIn, але вони повинні підписатись на план підписки на

Щойно ви дослідите LinkedIn безкоштовно, ви зможете перейти на інший обліковий запис LinkedIn за допомогою більше можливостей. Доступні

різноманітні платні підписки на LinkedIn, що починаються з 15,95 доларів на місяць і призначені для компаній, шукачів роботи, рекрутерів тощо.

Однією з особливостей більш просунутих облікових записів є можливість надсилати InMail кожному. InMail - це внутрішнє повідомлення LinkedIn, надіслане комусь, з ким ви не пов'язані. Ви можете надсилати безкоштовні повідомлення своїм контактам, але не можете - вам потрібно надіслати InMail тим, до кого ви ще не підключені, але кількість обмежена. Основний обліковий запис Business Premium дозволяє відправляти три повідомлення InMail на місяць, тоді як план Business Plus - 10, а план Executive - 25 на місяць. Тож виберіть потрібну кількість таких повідомлень для вас.

Оновлені облікові записи мають доступ до більшої кількості результатів пошуку, що може бути великою перевагою для рекрутерів LinkedIn. Крім того, ви матимете доступ до додаткових інструментів для зберігання та організації профілів, і ви зможете побачити повний список людей, які переглядають ваш профіль LinkedIn. "

3.3 Розробка логіко-функціональної структури парсеру

Інтернет як найбільш стрімко розвивається засіб комунікації набуває все більш важливу роль в діяльності людини. У цій ситуації є актуальним розробка програмних систем, що дозволяють відбирати інформаційні ресурси мережі Інтернет відповідно до запиту користувача. Ефективність цього процесу можуть забезпечити, що з'явилися з розвитком інтелектуальних технологій, спеціальні програми, які називаються інтелектуальними помічниками.

Збір інформації в інтернеті - трудомістка, рутинна, що забирає багато часу робота. Парсери, здатні протягом доби перебрати більшу частину веб-ресурсів в пошуках потрібної інформації, автоматизують її.

Найбільш активно в цілях парсинга використовується всесвітня мережа пошукових систем. Роботи збирають інформацію і в приватних інтересах.

Адміністратори сайтів найчастіше використовують автоматичний аналіз для наповнення свого сайту. Якщо часто змінюєте та оновлюєте текст на сайті, це знадобиться.

Веб-скрейпінг - необхідність для організаторів спам-розсилок по електронній пошті або каналів мобільного зв'язку. Для цього їм треба запуснути бота подорожувати по соціальним мережам і збирати телефони, адреси, явки.

Ну і господарі деяких, особливо недавно організованих веб-ресурсів, люблять наповнити свій сайт чужим контентом.

Звичайно ж, парсери не читають тексти, вони всього лише порівнюють текст с запрошеним та діють за програмою. Розглянемо функціонал програми. Програма запускає пошук по випускникам "Криворізького Національного Університету". У знайдених студентах розберемо поля які необхідні в цілях програми.

Розроблене додаток можна розширити функціями, пов'язаними з синтаксичним аналізом, такими як, пошук слів кожної статті або підрахунок пропозицій всіх новинних ресурсів конкретного сайту. Також можна написати інтерфейс для програми або сформувати файл у форматі XML і JSON.

```
Thread.Sleep( millisecondsTimeout: 1000);
string name = _driver // IWebDriver
    .FindElement(By.XPath(
        "//*[@class='text-heading-xlarge inline t-24 v-align-middle break-words']")) // IWebElement
    .GetAttribute("innerText");
string title = _driver.FindElement(By.XPath("//*[@class='text-body-medium break-words']")) // IWebElement
    .GetAttribute("innerText");
string resultStr = $"{name};{title}";
var education :ReadOnlyCollection<IWebElement> = _driver.FindElements(By.XPath(
    "//*[@class='pv-entity__school-name t-16 t-black t-bold']"));

foreach (var element in education)
{
    resultStr +=

```

Рисунок 3.4 – Приклад с коду програми

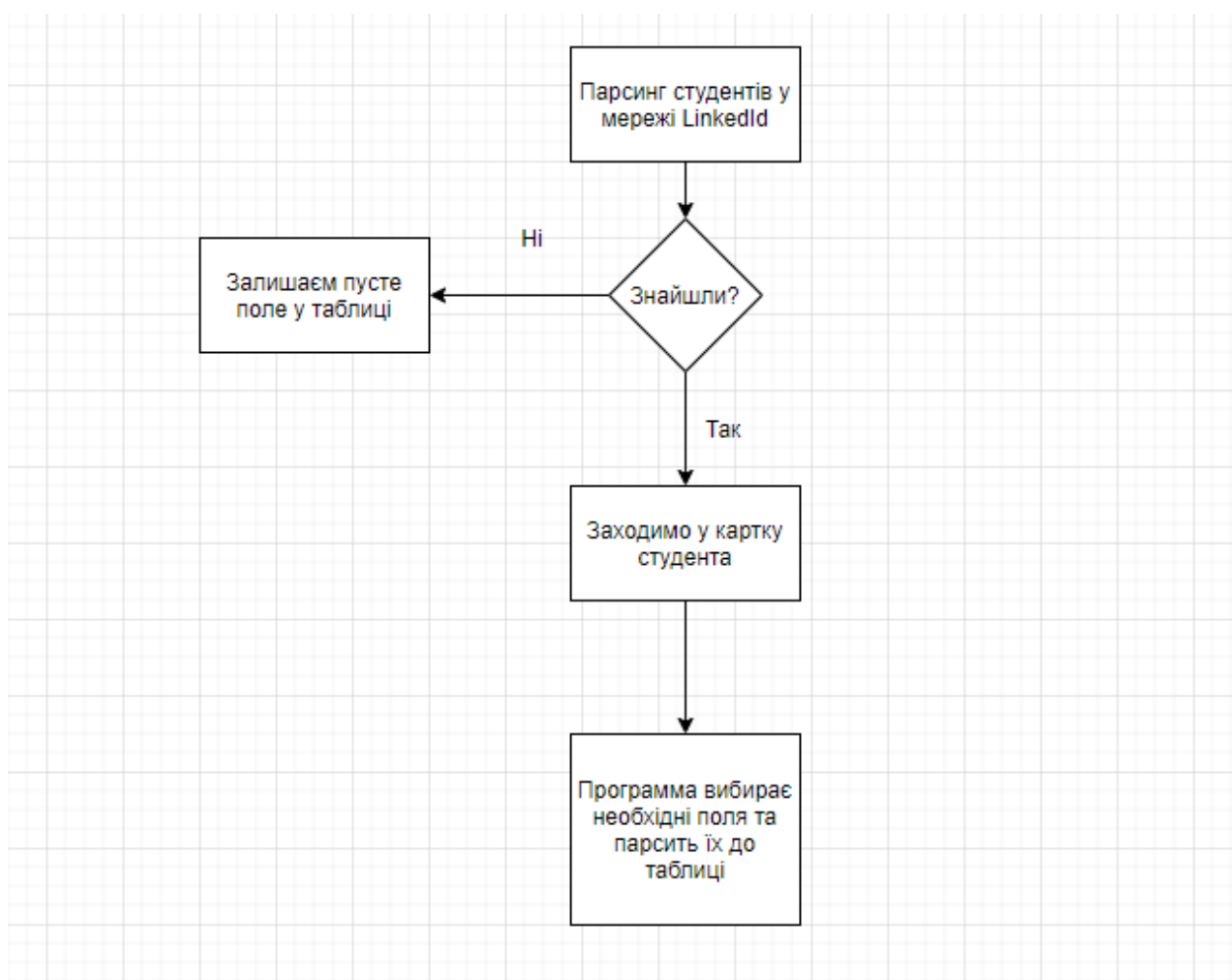


Рисунок 3.5 – Алгоритм дії програми

РОЗДІЛ 4. ПРОЦЕС РОЗРОБКИ ТА ТЕСТУВАННЯ

4.1 Розробка програми

Парсер - програма, що забезпечує парсинг, тобто синтаксичний аналіз контенту в мережі по певної математичної моделі, створеної на одній з мов програмування, зазвичай на Python, PHP або Java. Алгоритм роботи програми-парсера в кожному випадку приблизно однаковий:

Програма отримує доступ до ресурсу в мережі інтернет;

-завантаження коду сторінки, парсинг якою необхідно провести;

-читання і обробка інформації;

-надання результату в одному із зручних форматів - .html, .xml, .sql та інших.

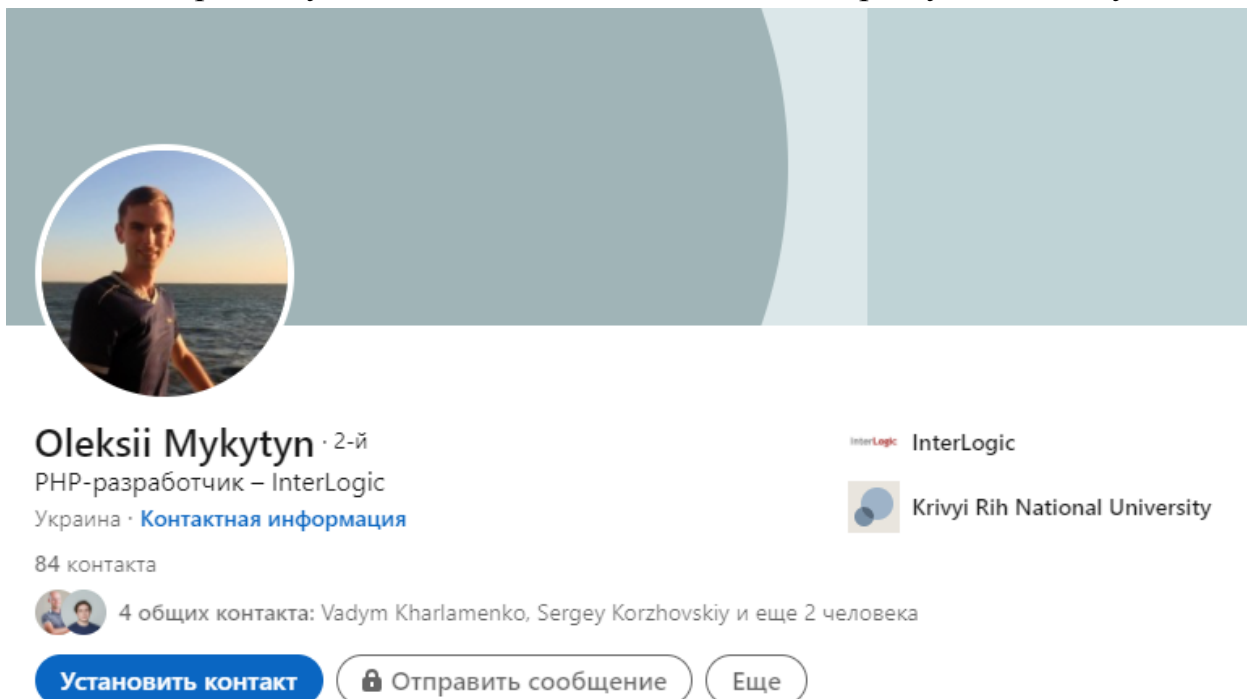
Програма порівнює заданий набір слів або значень з знайденим в мережі - з урахуванням заданих же обмежень . Алгоритм подальших дій зі знайденим контентом також прописаний заздалегідь - це може бути просто надання інформації або певні дії, наприклад, вчинення ставок при парсингу спортивних подій.

Завдання моєї програми – це зібрати данні випускників мого університету, щоб ми могли знайти фахівців у тій чи іншій галузі. При цьому ми будемо отримувати лише актуальну інформацію о галузі людини. Тому що в університеті часто не знають про актуальне місце роботи випускників.

Тому мережа LinkedIn добре підходить для таких цілей, вона дозволяє дізнатися актуальне місце роботи, освіти, хобі та компанії, в яких працював студент.

					КНУ.РБ.123.20.01.ВС		
Зм.	Арк.	№ документа	Підпис	Дата	ВСТУП		
Розробив	Слюсар						
Перевірив	Купін				KI-17-1		
Н. контроль	Кузнєцов						
Затвердив	Купін						

Для прикладу я знайшов сторінку випускника:



Oleksii Mykytyn · 2-й
PHP-розробчик – InterLogic
Україна · [Контактная информация](#)

84 контакти

4 общих контакта: Vadym Kharlamenko, Sergey Korzhovskiy и еще 2 человека

[Установить контакт](#) [Отправить сообщение](#) [Еще](#)

Рисунок 2.4 – Сторінка випускника

Одразу ми можете побачити поля Прізвища та Ім'я, місце навчання та професію.

Також нам треба зібрати інформацію з колишнім місцем роботи:

Опыт работы



PHP-разработчик

InterLogic · Полный рабочий день
март 2020 г. – настоящее время · 1 г. 4 мес.
Львов, Львовская область, Украина



Web Software Developer

Туристическая компания «Namaste»
июнь 2017 г. – настоящее время · 4 г. 1 мес.
Украина



Chief Technology Officer

2Event
апр. 2018 г. – март 2020 г. · 2 г.
Украина



Инженер-программист АСУ

Проектремонтаж
июнь 2017 г. – май 2018 г. · 1 г.
Кривой Рог, Украина



Техник научно-исследовательской части

ГБУЗ "Криворожский национальный университет"
январь 2016 г. – январь 2017 г. · 1 г. 1 мес.
Кривой Рог, Украина

Продвижение журнала "Computer Science Information Technology Automation": разработка сайта, дизайн, верстка. Работа с международными научно-метрическими базами для индексирования изданий.

Дополнительные сведения об опыте работы (3) ▾

Рисунок 2.4 – Актуальне та минулі місця роботи

В майбутньому я хочу допрацювати програму для парсингу не лише с нашого університету, а інші, щоб студенти могли знайти роботу швидше.

Писати програму я буду на C#, вивчати я починав його ще в комп'ютерній академії ШАГ.

4.2 Проектування

Для роботи в С# з парсингом підключаю такі бібліотеки:

```
using System;
using System.Collections.Generic;

#region Библиотека контроля потоков
using System.Threading;
#endregion

#region Библиотеки для работы с Selenium WebDriver (позволяет управлять браузером , чтобы загрузить JS)
using OpenQA.Selenium;
using OpenQA.Selenium.Chrome;
using OpenQA.Selenium.Interactions;
#endregion
```

Рисунок 2.4 – Використані бібліотеки

У парсингу через LinkedIn є свої нюанси, такі як:

- обмежена кількість пошуків людей та переглянути унікальних сторінок
- для парсингу потрібно робити це з авторизованої сторінки, акаунта. Тому в програмі я вказував свій логін та пароль

```
/// <summary>
/// Класс отвечающие за внутренние манипуляции внутри браузера
/// </summary>
Ссылка: 3
public class WebDriverPanel
{
    private readonly IWebDriver _driver;
    private const string loginString = "XXXXXXXXXXXX";
    private const string passwordString = "XXXXXX";
```

Рисунок 2.4 – Частина коду, де можна замінити логін та пароль

Щоб мати необмежену кількість пошуків треба придбати бізнес акаунт.

```

/// <summary>
/// Конструктор ініціалізації WebDriver
/// </summary>
ссылка: 1
public WebDriverPanel()
{
    string pathToFile = AppDomain.CurrentDomain.BaseDirectory + @"\";

    _driver = new ChromeDriver(pathToFile);

    Thread.Sleep(200);
}
/// <summary>
/// Перейти на відповідний URL
/// </summary>
/// <param name="URL"></param>
Ссылка: 2
public void GoToUrl(string URL)
{
    _driver.Navigate().GoToUrl(URL);
}

```

Рисунок 2.4 – Частина коду з функцією WebDriver

В мене сразу од разу виникли проблеми с авторизацією та капчею, але таким способом я зміг без проблем авторизуватися:

```

/// <summary>
/// Метод авторизації користувача по константам login, password
/// </summary>
ссылка: 1
public void Autorisation()
{
    _driver.FindElement(By.ClassName("main__sign-in-link")).Click();
    Thread.Sleep(200);

    var login = _driver.FindElement(By.Name("session_key"));
    var password = _driver.FindElement(By.Name("session_password"));
    login.SendKeys(loginString);
    password.SendKeys(passwordString);
    _driver.FindElement(By.XPath("//div[@class='login__form_action_container']/button")).Click();
}

```

Рисунок 2.4 – Функція авторизації з моїми даними

```

/// <summary>
/// Получить результат парсинга
/// </summary>
ссылка:1
public void GetParsedItemsSearchResult()
{
    Thread.Sleep(1000);

    var items = _driver.FindElements(
        By.XPath("//*[[@class='entity-result__title-text t-16']/a/span/span[1]]"));

    for (int i = 0; i < items.Count; i++)
    {
        Actions action = new Actions(_driver);
        action.SendKeys(Keys.Escape);
        items = _driver.FindElements(
            By.XPath(
                "//*[[@class='entity-result__title-text t-16']/a/span/span[1]]"));

        items[i].Click();
        Thread.Sleep(1000);
        // Имя пользователя
        string name = _driver
            .FindElement(By.XPath(
                "//*[[@class='text-heading-xlarge inline t-24 v-align-middle break-words']]"))
            .GetAttribute("innerText");
        // Описание для пользователя
        string title = _driver.FindElement(By.XPath("//div[@class='text-body-medium break-words']"))
            .GetAttribute("innerText");
        // Результирующая строка, которая будет сохраняться в файл
        string resultStr = $"{name};{title}";

        // Полный список все образований выбранного пользователя
        var education = _driver.FindElements(By.XPath(
            "//*[[@class='pv-entity__school-name t-16 t-black t-bold']]"));
        // Разделить список на отдельные элементы и сохранить
        foreach (var element in education)
        {
            resultStr += $"{element.GetAttribute("innerText")};";
        }

        resultParsing.Add(resultStr);
        //Вернуться назад и перезагрузить страницу
        _driver.Navigate().Back();
        _driver.Navigate().Refresh();
    }
}

```

Рисунок 2.4 – Фінальна частина самого збору інформації

Тут нарешті я дійшов до частини, де саме збираю інформацію.

Збирав я:

- Ім'я та Прізвище
- Опис
- Список місць навчання

Список я ділив для зручного пошуку по критеріям

4.3 Реалізація

```

namespace LinkedInParser
{
    Ссылка: 0
    static class Program
    {
        Ссылка: 0
        static void Main(string[] args)
        {
            WebDriverPanel webDriverPanel = new WebDriverPanel();
            webDriverPanel.GoToUrl(
                @"https://www.linkedin.com/search/results/people/?origin=FACETED_SEARCH&page=1&schoolFreetext=" +
                "\"Криворожский%20Национальный%20университет\"");
            webDriverPanel.Autorisation();
            //Количество страниц полученных поиском, может сломаться если страниц будет меньше
            const int index = 100;
            for (int i = 1; i < index; i++)
            {
                webDriverPanel.GoToUrl(
                    $"https://www.linkedin.com/search/results/people/?origin=FACETED_SEARCH&page={i}&schoolFreetext=" +
                    "\"Криворожский%20Национальный%20университет\"");

                webDriverPanel.GetParsedItemsSearchResult();
            }

            DataSaver.SaveToCSV(webDriverPanel.resultParsing);
        }
    }
}

```

Рисунок 2.4 – Тут описано ресурс, з якого ми парсимо та функція запуску програми

Якщо простими словами, то браузер робить запит серверу з проханням дати потрібну інформацію, а сервер в свою чергу, цю інформацію надає. Все це здійснюється за допомогою HTTP запитів.

Щоб подивитися на запити, які відправляє ваш браузер на конкретному сайті, то просто відкрийте цей сайт, натисніть F12 і перейдіть у вкладку Network, після цього перезавантажте сторінку.

```

C:\Users\brozz\Desktop\LinkedInParser\LinkedInParser\bin\Debug\netcoreapp3.1\LinkedInParser.exe
Starting ChromeDriver 91.0.4472.19 (1bf021f248676a0b2ab3ee0561d83a59e424c23e-refs/branch-heads/4472@{#288}) on port 4988
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
DevTools listening on ws://127.0.0.1:49888/devtools/browser/1bc85940-2e52-43d8-afb8-0942cd19a09d

```

Рисунок 2.4 – Вікно роботи програми

Пошук програми виконую автоматично у вашому браузері. Для написання цієї програми я вивчив методи парсингу на С#, різні сторони інструменти для веб-скрабінгу. Контент обробляється ботами в заданій послідовності. Це дає можливість одночасно обробляти різні масиви даних.

Плюси парсингу на мою думку:

- збирати дані швидше і в будь-якому режимі, хоч цілодобово;
- слідувати всім заданим параметрам, навіть дуже тонким;
- уникати помилок від неувважності або втоми;
- виконувати регулярну перевірку по заданому інтервалу (кожну тиждень і т. п.);

РОЗДІЛ 5. ФІНАНСОВЕ ПІДГРУНТЯ ПРОЕКТУ

Google Play — найбільший магазин додатків, в каталог якого входять сотні ігор, книг, музики і фільмів. На поточний момент він обганяє AppStore в 2,5 рази. Його потенційними користувачами є половина власників смартфонів у світі. Як це можна використовувати?

Завдяки платформі всі бажаючі від великих компаній до приватних розробників мають можливість демонструвати і монетизувати свої додатки.

Скільки можна заробити на додатку Android і чи можна це робити, маючи невелику аудиторію? Google Play-це вітрина, яка дає отримувати дохід від мобільних додатків. За статистикою 45% розробників, представлених в маркеті, заробляють більше \$1000 в місяць.

І це не рахуючи прогнозів аналітиків, які хором пророкують 6 млрд смартфонів у 2022 році та понад 200 млрд установок додатків на рік.

Чи можна спробувати удачі на цьому ринку і чи потрібно продавати свій додаток, щоб отримати частину цього пирога? Не обов'язково.

Величезний трафік маркету дозволяє монетизуватися за рахунок показу реклами, вбудованих покупок, підписок, донатів і DLC-контенту.

Наприклад, *Balloon Island* опублікували в сторі більше 10 безкоштовних головоломок, які генерують прибуток за рахунок рекламної монетизації. Вони одні з тих, хто зробив заробіток на додатках Android дуже прибутковим напрямком.

Інший приклад: *Blood Pressure* заробляє, використовуючи відразу два методи: рекламу в додатку і вбудовані покупки.

Google Play Market дає вільно поєднувати обидва типи реклами.

Але саме зараз треба враховувати кризове світове становище, саме тому не один з методів реклами або монетизації, чи зазначений вище, чи можливий в перспективі, не є правильним.

Я, як автор цієї кваліфікаційної роботи, не вважаю дорічним впроваджувати розвиток економічно-експлуатаційних систем та наживання на користувачах у часи повальної смуті, коли сама людяність та суверенітет нашої Батьківщини знаходиться під загрозою.

У такому становищі максимально можливим задля забезпечення поліпшення та підтримки роботи програми у подальшому є добровільні благодійні внески користувачів, як основний фактор підтримки роботи програми.

ВИСНОВОК

Мета дипломної роботи - дослідити основні способи веб-скрейпінга для пошуку інформації та реалізувати відповідну програму - досягнута.

При підготовці до дипломної роботи досліджувалися питання, пов'язані з вивченням і застосуванням технології веб-скрейпінга, розглянуті види пошукових робіт і способи роботи синтаксичного аналізу.

У роботі розроблено додаток, що дозволяє знайти всю потрібну інформацію з веб-ресурсу LinkedIn і записати ці дані в реляційної базі даних простої структури.

Розроблене додаток можна розширити функціями, пов'язаними з синтаксичним аналізом, такими як, пошук роботи кожного студента або підрахунок студентів того чи іншого закладів.

Також можна написати інтерфейс програми і розширити проект парсинга інших веб-ресурсів.

В майбутньому я буду розвивати програму, одразу хочу додати підтримку WPF, і зробити вбудовану таблицю. Додати інші можливості для пошуку роботодавців, чи навпаки фахівців.

Якщо мені вдасться, я хочу зробити с цього проекту сервіс, що зможе дати необхідне роботодавцям, людям, які в пошуках необхідної роботи, фрілансерам та іншим.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Ріхтер Джеффрі. CLR via C#. Програмування на платформі Microsoft.NET Framework 4.5 мовою C#. 4-е изд. / Джеффрі Ріхтер. - Пітер, 2013. - 896 с.
- 2) Стілмен Е. Вивчаємо C#. Включаючи C# .NET 4.0 і Visual Studio 2010. 2-е видання / Е. Стілмен., Дж. Грін - O'Reilly, 2012. - 689 с.
- 3) Троелсен Ендрю. Мова програмування C# 2010 і платформа.NET 4 / Ендрю Троелсен. - Вільямс, 2010. - +1392 с.
- 4) Фленов М. Біблія C#, 2-е видання / М. Фленов. - БХВ-Петербург, 2011. - 560 с.
- 5) Шілдт Герберт. C# 4.0: Повне керівництво / Герберт Шілдт. - Вільямс, 2011. - 1056 с.
- 6) Агуров П. C#. Збірник рецептів / П. Агуров. - БХВ-Петербург, 2007. - 432 с.
- 7) Павловська Т.О. C#. Програмування на мові високого рівня: Підручник для вузів / Т.А. Павловська. - Пітер, 2009. - 432 с.
- 8) Бьюлі А. Вивчаємо SQL / Алан Бьюлі. - Символ-Плюс, 2007. - 312 с.
- 9) Молінаро Ентоні. SQL. Збірник рецептів / Ентоні Молінаро. - O'Reilly, 2009. - 672 с.
- 10) Бураков П.В. Введення в системи баз даних. Навчальний посібник / П.В. Бураков., В.Ю. Петров. - СПбГУ ІТМО, 2010. 129 с.
- 11) Прикладна та комп'ютерна лінгвістика . Изд. 2-е. – М.: ЛЕНАНД, 2017. – 320 с.
- 12) Сайт Microsoft Virtual Academy // MVA: <https://mva.microsoft.com/>.
- 13) Сайт SQLite // SQLite Home Page: <http://sqlite.org/>.

					КНУ.РБ.123.20.01.ВС		
Зм.	Арк.	№ документа	Підпис	Дата	СПИСОК ДЖЕРЕЛ		
Розробив	Слюсар						
Перевірив	Купін				KI-17-1		
Н. контроль	Кузнєцов						
Затвердив	Купін						

- 14) Сайт NewtonsoftJson // Json.NET: <http://www.newtonsoft.com/json>.
- 15) Microsoft Visual Studio // Матеріал з Вікіпедії - Вільної енциклопедії: http://ru.wikipedia.org/wiki/Microsoft_Visual_Studio.
- 16) Microsoft.NET. // Матеріал з Вікіпедії - Вільної енциклопедії: <http://uk.wikipedia.org/wiki/>.
- 17) NET Framework Developer Center // Головна сторінка: <http://msdn.microsoft.com/ruru/netframework/default.aspx>.
- 18) Документація компілятора C # Roslyn // Github: <https://github.com/dotnet/roslyn>.
- 19) Мова програмування C #: https://uk.wikipedia.org/wiki/C_Sharp.
- 20) XAML // Матеріал з Вікіпедії - Вільної енциклопедії: <https://ru.wikipedia.org/wiki/XAML>

Додаток А

Код програми

```

namespace LinkedInParser
{
    static class Program
    {
        static void Main(string[] args)
        {
            WebDriverPanel webDriverPanel = new WebDriverPanel();
            webDriverPanel.GoToUrl(
                @"https://www.linkedin.com/search/results/people/?origin=FACETED_SEARCH
                &page=1&schoolFreetext=" +
                "\"Криворожский%20Национальный%20университет\"");
            webDriverPanel.Autorisation();
            //Количество страниц полученных поиском, может сломаться если
            страниц будет меньше
            const int index = 100;
            for (int i = 1; i < index; i++)
            {
                webDriverPanel.GoToUrl(
                    $"https://www.linkedin.com/search/results/people/?origin=FACETED_SEARCH&
                    page={i}&schoolFreetext=" +
                    "\"Криворожский%20Национальный%20университет\"");

                webDriverPanel.GetParsedItemsSearchResult();
            }

            DataSaver.SaveToCSV(webDriverPanel.resultParsing);

        }
    }
}

using System;
using System.Collections.Generic;
using System.IO;

namespace LinkedInParser
{
    public static class DataSaver
    {

```

```

    /// <summary>
    /// Сохранить в CSV файл полученные строки
    /// </summary>
    /// <param name="collection"></param>
    public static void SaveToCSV(List<String> collection)
    {
        using var f = new StreamWriter("test.csv");
        foreach (var t in collection)
        {
            f.WriteLine(t);
            f.Flush();
        }
    }
}

```

Index.js

```
/**
```

```
 * @format
```

```
 */
```

```

import {AppRegistry} from 'react-native';
import App from './App';
import {name as appName} from './app.json';

```

App.js

```

import React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';

import HomeScreen from './src/screens/Home/Home';
import ScannerScreen from './src/screens/Scanner/Scanner';
import ListAdditiveScreen from './src/screens/ListAdditive/ListAdditive';

```

```
import AdditiveInfoScreen from './src/screens/AdditiveInfo/AdditiveInfo';

const Stack = createNativeStackNavigator();

const App = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator
        screenOptions={{
          headerShown: false,
        }}
      >
        <Stack.Screen
          name="Home"
          component={HomeScreen}
        />
        <Stack.Screen
          name="Scanner"
          component={ScannerScreen}
        />
        <Stack.Screen
          name="ListAdditive"
          component={ListAdditiveScreen}
        />
        <Stack.Screen
          name="AdditiveInfo"
          component={AdditiveInfoScreen}
        />
      </Stack.Navigator>
    </NavigationContainer>
  )
}
```

```
);
};
```

```
export default App;
```

```
using System;
```

```
using System.Collections.Generic;
```

```
#region Библиотека контроля потоков
```

```
    using System.Threading;
```

```
#endregion
```

```
#region Библиотеки для работы с Selenium WebDriver (позволяет управлять браузером , чтобы загружать JS)
```

```
    using OpenQA.Selenium;
```

```
    using OpenQA.Selenium.Chrome;
```

```
    using OpenQA.Selenium.Interactions;
```

```
#endregion
```

```
namespace LinkedInParser
```

```
{
```

```
    /// <summary>
```

```
    /// Класс отвечающие за внутренние манипуляции внутри браузера
```

```
    /// </summary>
```

```
    public class WebDriverPanel
```

```
    {
```

```
        private readonly IWebDriver _driver;
```

```
        private const string loginString = "brozzers666@gmail.com";
```

```
        private const string passwordString = "viber719";
```

```
        public List<String> resultParsing = new List<string>();
```

```
        /// <summary>
```

```
        /// Конструктор инициализации WebDriver
```

```
        /// </summary>
```

```
        public WebDriverPanel()
```

```
        {
```

```
            string pathToFile = AppDomain.CurrentDomain.BaseDirectory + @"\";
```

```
            _driver = new ChromeDriver(pathToFile);
```

```
            Thread.Sleep(200);
```

```

    }
    /// <summary>
    /// Перейти на соответствующий URL
    /// </summary>
    /// <param name="URL"></param>
    public void GoToUrl(string URL)
    {
        _driver.Navigate().GoToUrl(URL);
    }
    /// <summary>
    /// Метод авторизации пользователя по константам login, password
    /// </summary>
    public void Autorisation()
    {
        _driver.FindElement(By.ClassName("main__sign-in-link")).Click();
        Thread.Sleep(200);

        var login = _driver.FindElement(By.Name("session_key"));
        var password = _driver.FindElement(By.Name("session_password"));
        login.SendKeys(loggingString);
        password.SendKeys(passwordString);

        _driver.FindElement(By.XPath("//div[@class='login__form_action_container']/button")).Click();
    }

    /// <summary>
    /// Получить результат парсинга
    /// </summary>
    public void GetParsedItemsSearchResult()
    {
        Thread.Sleep(1000);

        var items = _driver.FindElements(
            By.XPath("//*[@class='entity-result__title-text t-16']/a/span/span[1]"));

        for (int i = 0; i < items.Count; i++)
        {
            Actions action = new Actions(_driver);
            action.SendKeys(Keys.Escape);
            items = _driver.FindElements(

```

```

        By.XPath(
            "//*[(@class='entity-result__title-text t-16')/a]");
Theme.js

```

```

import { Platform } from 'react-native';
import { Dimensions } from "react-native";
const { width, height } = Dimensions.get("screen");

```

```

export const COLORS = {
  primary: 'white',
  darkRed: '#9E1C1C',
  strongRed: '#D03434',
  lightGray: '#C4C4C4',
  veryLightGray: '#EAEAEA',
  mostlyBlack: 'rgba(35, 35, 35, 0.9)',
  moderateRed: '#D44B4B',
  brightOrange: '#F3BB43',
  moderateGreen: '#92D23B',
  mostlyWhite: '#F3F3F3',
  grayishGreen: '#c1d2ab',
  verySoftRed: '#efb4b4',
  veryLightOrange: '#FFDD8A',
  greenBorder: '#9BB676',
  redBorder: '#E97E7E',
  orangeBorder: '#EFC55E'
}

```

```

export const SIZES = {
  // global sizes
  font: 17,

```

```

radius: 50,
padding: 25,

// font sizes
h1: 30,
h2: 28,
h3: 24,
h4: 20,
h5: 18,
body18: 18,
body17: 17,
body14: 14,

// app dimensions
width,
height
};

export const FONTS = {
  h1: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' : 'SFProDisplay-Bold',
    fontSize: SIZES.h1, fontWeight: 'bold' },
  h2: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' : 'SFProDisplay-Bold',
    fontSize: SIZES.h2, fontWeight: 'bold' },
  h2_R: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Regular', fontSize: SIZES.h2, fontWeight: '400' },
  h3: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' : 'SFProDisplay-Bold',
    fontSize: SIZES.h3, lineHeight: 34, fontWeight: 'bold' },
  h4: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Semibold', fontSize: SIZES.h4, lineHeight: 20, fontWeight: '600' },
  h5: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' : 'SFProDisplay-Bold',
    fontSize: SIZES.h5, lineHeight: 22, fontWeight: 'bold' },

```

```
body_14_SF_R: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Regular', fontSize: SIZES.body14, lineHeight: 24, fontWeight: '400'
},
```

```
body_16_SF_R: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Regular', fontSize: SIZES.body16, lineHeight: 18, fontWeight: '400'
},
```

```
body_17_SF_R: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Regular', fontSize: SIZES.body17, lineHeight: 24, fontWeight: '400'
},
```

```
body_18_SF_R: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Regular', fontSize: SIZES.body18, lineHeight: 24, fontWeight: '400'
},
```

```
body_17_SF_SB: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Semibold', fontSize: SIZES.body17, lineHeight: 24, fontWeight:
'600' },
```

```
body_18_SF_SB: { fontFamily: Platform.OS === 'ios' ? 'SF Pro Display' :
'SFProDisplay-Semibold', fontSize: SIZES.body18, lineHeight: 24, fontWeight:
'600' },
```

```
};
```

```
const appTheme = { COLORS, SIZES, FONTS };
```

```
export default appTheme;
```

```
    if (i > items.Count-1)
        break;
    if (items.Count == 0)
        break;
    items[i].Click();
    Thread.Sleep(1000);
    // Имя пользователя
    try
    {
        string name = _driver
            .FindElement(By.XPath(
                "//*[@class='text-heading-xlarge inline t-24 v-align-middle
break-words']"))
            .GetAttribute("innerText");
```

```

        // Описание для пользователя
        string title =
_driver.FindElement(By.XPath("//div[@class='text-body-medium break-words']"))
        .GetAttribute("innerText");
        // Результирующая строка, которая будет сохраняться в файл
        string resultStr = $" {name}; {title}";

        // Полный список все образований выбранного пользователя
        var education = _driver.FindElements(By.XPath(
            "/*[@class='pv-entity__school-name t-16 t-black t-bold']"));
        // Разделить список на отдельные элементы и сохранить
        foreach (var element in education)
        {
            resultStr += $" {element.GetAttribute("innerText")}";
        }
        resultParsing.Add(resultStr);
        _driver.Navigate().Back();
    }
    catch
    icons.js

```

```

const backImg = require('../assets/images/backImg.png')
const flashOff = require('../assets/images/flash.png')
const flashOn = require('../assets/images/flashOn.png')
const list = require('../assets/images/list.png')
const buttonScan = require('../assets/images/button-scan.png')
const search = require('../assets/images/search.png')
const close = require('../assets/images/close.png')
const arrowDownUp = require('../assets/images/arrow-down.png')
import React, { useEffect, useRef, useState } from 'react';
import {
    Image,
    SafeAreaView,
    ScrollView,
    StatusBar,
    Text,

```

```

    TouchableOpacity,
    View,
  } from 'react-native';

  //#region plugins
  import LottieView from 'lottie-react-native';
  import DropShadow from "react-native-drop-shadow";
  //#endregion

  //#region components
  import { COLORS, images, theme, icons } from '../constants';
  //#endregion

  import { styles } from './styles'

  const AdditiveInfoScreen = ({ navigation, route }) => {
    const data = route.params.data

    const goBack = () => {
      navigation.goBack()
    }

    return (
      <SafeAreaView style={styles.container}>
        <StatusBar backgroundColor={COLORS.primary} barStyle={'dark-content'} />
        <ScrollView showsVerticalScrollIndicator={false}>
          <View style={styles.backArrowInner}>
            <DropShadow style={styles.backArrowShadow}>
              <TouchableOpacity style={styles.backArrow} onPress={() => goBack()}>

```

```

        <Image source={icons.backImg} style={styles.backArrowImg} />
      </TouchableOpacity>
    </DropShadow>
  </View>
  <View style={styles.statusImgInner}>
    {data.danger.number_code === 0 ?
      <LottieView source={images.green} autoPlay loop
style={styles.statusImg} />
    }
import React, { useEffect, useRef, useState } from 'react';
import {
  FlatList,
  Image,
  SafeAreaView,
  StatusBar,
  Text,
  TextInput,
  TouchableOpacity,
  View,
} from 'react-native';

##region plagins
import SearchableDropdown from 'react-native-searchable-dropdown';
##endregion

##region components
import { COLORS, theme, icons } from '../constants';
##endregion

import { styles } from './styles'
```

```

const ListAdditiveScreen = ({ navigation, route }) => {
  const data = route.params.data
  const flatListRef = useRef()
  const [sectionData, setSectionData] = useState([])
  const [searchableItems, setSearchableItems] = useState([])

  const Item = ({ section }) => (
    <View>
      <Text style={[theme.FONTS.h5, styles.sectionTitle]}>{section.name}</Text>
      <View style={styles.line} />
      <View style={styles.sectionItemContainer}>
        {section.nutritionalsupplement.map((item, i) => (
          <TouchableOpacity
            key={i}
            style={styles.sectionItemInner}
            onPress={() => navigation.navigate('AdditiveInfo', { data: item })}
          >
            <Text style={[theme.FONTS.body_17_SF_R]}>{item.name}</Text>
          </TouchableOpacity>
        ))}
      </View>
    </View>
  );

  function goToElemtns(index) {
    flatListRef.current.scrollToIndex({ index: index })
  }

  useEffect(() => {
    let itemsTemp = []

```

```

Object.keys(data).map(item => {
  data[item].nutritionalsupplement.map((item, i) => {
    itemsTemp.push({ id: i, name: item.name, data: item })
  })
})
setSearchableItems(itemsTemp)
let sectionDataTemp = []
Object.keys(data).map(item => {
  if (data[item].nutritionalsupplement.length > 0) {
    sectionDataTemp.push(data[item])
  }
})
setSectionData(sectionDataTemp)
}, [])

```

```

      : data.danger.number_code === 1 ?
        <LottieView source={images.orange} autoPlay loop
style={styles.statusImg} />
      :
        <LottieView source={images.red} autoPlay loop style={styles.statusImg}
/>
    }
    <View style={styles.statusTitleInner}>
      <Text style={[theme.FONTS.h5, styles.statusTitle]}>
        {data.danger.title}
      </Text>
      <Text style={[theme.FONTS.body_16_SF_R, styles.statusSubtitle]}>для
применения</Text>
    </View>
  </View>
  <DropShadow style={styles.infoWrapperShadow}>

```

```

<View style={styles.infoWrapper}>
  <Text style={[theme.FONTS.h2_R]}>Пищевая добавка</Text>
  <Text
    style={[
      theme.FONTS.h5,
      { color: data.danger.number_code === 0 ? COLORS.moderateGreen :
data.danger.number_code === 1 ? COLORS.brightOrange :
COLORS.moderateRed }
    ]}
  >
    {`«${data.name}»`}
  </Text>
  <Text style={[theme.FONTS.body_17_SF_R,
styles.category]}>{data.category.name}</Text>
  <Text style={[theme.FONTS.body_17_SF_R,
styles.text]}>{data.text}</Text>
  <TouchableOpacity style={styles.backBtnInner} onPress={() =>
goBack()}>
    <Text style={[theme.FONTS.body_18_SF_SB,
styles.backBtn]}>Назад</Text>
  </TouchableOpacity>
</View>
</DropShadow>
</ScrollView>
</SafeAreaView>
);
};

export default AdditiveInfoScreen;
export default {
  buttonScan,
  flashOff,

```

```

    flashOn,
    list,
    backImg,
    search,
    close,
    arrowDownUp
  }
  const scanning = require('../assets/images/video1.json')
  const recogniting = require('../assets/images/video2.json')
  const red = require('../assets/images/red.json')
  const green = require('../assets/images/green.json')
  const orange = require('../assets/images/orange.json')
  const windowsScan = require('../assets/images/windows-scan.png')
  const menuBackground = require('../assets/images/menu-background.png')
  const scanProcessing = require('../assets/images/scan-processing.json')

  export default {
    scanning,
    recogniting,
    red,
    green,
    orange,
    windowsScan,
    menuBackground,
    scanProcessing,
  }
  import images from "../images";
  import theme, { COLORS, SIZES, FONTS } from '../theme'
  import tempData from "../tempData";
  import icons from "../icons";

```

```

export {
  images,
  theme,
  COLORS,
  SIZES,
  FONTS,
  icons,
  tempData
}

import { StyleSheet } from "react-native";
import { COLORS, SIZES } from "../constants";

import { StyleSheet } from "react-native";
import { COLORS, SIZES } from "../constants";

export const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingRight: SIZES.padding,
    paddingLeft: SIZES.padding,
    backgroundColor: COLORS.primary
  },
  header: {
    flexDirection: 'row',
    alignItems: 'center',
    paddingTop: 40,
    paddingBottom: 18
  },
  backImg: {

```

```
    width: 10,
    height: 18
  },
  title: {
    flex: 1,
    textAlign: 'center'
  },
  searchInner: {
    flexDirection: 'row',
    //alignItems: 'center',
    borderRadius: SIZES.radius,
    //paddingHorizontal: 20,
    marginBottom: 10,
    //backgroundColor: COLORS.mostlyWhite
  },
  searchImgInner: {
    marginVertical: 5,
    backgroundColor: COLORS.mostlyWhite,
    flexDirection: 'column',
    justifyContent: 'center',
    paddingLeft: 20,
    paddingRight: 15,
    borderTopLeftRadius: SIZES.radius,
    borderBottomLeftRadius: SIZES.radius,
    height: 49,
  },
  searchImg: {
    width: 16,
    height: 17,
  },
```

```

SearchableDropdownContainer: {
  paddingVertical: 5,
  flex: 1,
},
SearchableDropdownItemContainer: {
  maxHeight: 140,
  marginLeft: -45,
  paddingHorizontal: 15
},
SearchableDropdownItem: {
  padding: 10,
  marginTop: 2,
  borderWidth: 1,
  borderRadius: 2,
  borderColor: COLORS.veryLightGray,
  paddingHorizontal: 12,
  paddingVertical: 5,
},
searchInput: {
  backgroundColor: COLORS.mostlyWhite,
  borderTopRightRadius: SIZES.radius,
  borderBottomRightRadius: SIZES.radius
},
listCategory: {
  flexDirection: 'row',
  flexWrap: 'wrap'
},
listCategoryItemInner: {
  marginVertical: 2,
  marginRight: 4
}

```

```

    },
    listCategoryItem: {
      borderWidth: 1,
      borderRadius: 2,
      borderColor: COLORS.veryLightGray,
      paddingHorizontal: 12,
      paddingVertical: 5,
    },
    sectionTitle: {
      color: COLORS.darkRed,
      paddingTop: 20,
      paddingBottom: 9
    },
    line: {
      height: 1,
      width: '100%',
      backgroundColor: COLORS.veryLightGray
    },
    sectionItemContainer: {
      flexDirection: 'row',
      flexWrap: 'wrap'
    },
    sectionItemInner: {
      marginRight: 60,
      marginVertical: 8
    }
  }
}

return (
  <SafeAreaView style={styles.container}>

```

```

<StatusBar backgroundColor={COLORS.primary} barStyle={'dark-content'}
/>

<View style={styles.header}>
  <TouchableOpacity onPress={() => navigation.goBack()}>
    <Image source={icons.backImg} style={styles.backImg} />
  </TouchableOpacity>
  <Text style={[theme.FONTS.h5, styles.title]}>Поиск по
ингредиенту</Text>
</View>
<View style={styles.searchInner}>
  <View style={styles.searchImgInner}>
    <Image source={icons.search} style={styles.searchImg} />
  </View>
  <SearchableDropdown
    items={searchableItems}
    resetValue={false}
    textInputProps={{
      placeholder: 'Поиск ингредиентов...',
      style: [theme.FONTS.body_14_SF_R, styles.searchInput]
    }}
    containerStyle={styles.SearchableDropdownContainer}
    itemsContainerStyle={styles.SearchableDropdownItemContainer}
    itemStyle={styles.SearchableDropdownItem}
    itemTextStyle={theme.FONTS.body_14_SF_R}
    listProps={
      {
        enableScrollViewScroll: true,
        nestedScrollEnabled: true,
      }
    }
  </SearchableDropdown>
</View>

```

```

onItemSelect={(item) => {
  navigation.navigate('AdditiveInfo', { data: item.data })
}}
/>
</View>
<FlatList
  ref={flatListRef}
  showsVerticalScrollIndicator={false}
  //scrollEnabled={false}
  ListHeaderComponent={(
    <View style={styles.listCategory}>
      {Object.keys(data).map((item, i) => {
        if (data[item].nutritionalsupplement.length > 0) {
          return (
            <TouchableOpacity key={i} onPress={() => goToElemtns(i)}
style={styles.listCategoryItemInner}>
              <Text style={[theme.FONTS.body_14_SF_R,
styles.listCategoryItem]}>{data[item].short_name}</Text>
            </TouchableOpacity>
          )
        }
      })}
    </View>
  )}
  data={sectionData}
  keyExtractor={(item, index) => index}
  renderItem={({ item }) => <Item section={item} />}
/>
</SafeAreaView>
);

```

```
};

export default ListAdditiveScreen;

export const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: COLORS.primary
  },
  backButtonInner: {
    paddingTop: 20,
    paddingLeft: SIZES.padding
  },
  backButtonShadow: {
    shadowColor: '#000',
    shadowOffset: {
      width: 0,
      height: 0
    },
    shadowOpacity: 0.2,
    shadowRadius: 20
  },
  backButton: {
    width: 40,
    height: 40,
    backgroundColor: COLORS.primary,
    borderRadius: SIZES.radius,
    justifyContent: 'center',
    alignItems: 'center'
  },
});
```

```
backArrowImg: {
  width: 10,
  height: 18
},
statusImgInner: {
  position: 'relative',
  flexDirection: 'row',
  justifyContent: 'center',
  alignItems: 'center'
},
statusImg: {
  width: '90%',
},
statusTitleInner: {
  top: 80,
  position: 'absolute',
  width: '61%',
},
statusTitle: {
  color: COLORS.primary,
  textTransform: 'uppercase',
  textAlign: 'center'
},
statusSubtitle: {
  color: COLORS.primary,
  textAlign: 'center'
},
infoWrapperShadow: {
  flex: 1,
  shadowColor: '#000',
```

```
shadowOffset: {
  width: 4,
  height: 8
},
shadowOpacity: 0.1,
shadowRadius: 64
},
infoWrapper: {
  flex: 1,
  borderTopEndRadius: 80,
  backgroundColor: COLORS.primary,
  paddingTop: 30,
  paddingHorizontal: 23
},
category: {
  opacity: 0.4
},
text: {
  paddingTop: 10,
  lineHeight: 31
},
backBtnInner: {
  marginVertical: 30
},
backBtn: {
  backgroundColor: COLORS.strongRed,
  textAlign: 'center',
  color: COLORS.primary,
Home.js
```

```

import React, { useEffect, useRef, useState } from 'react';
import {
  Animated,
  SafeAreaView,
  StatusBar,
  Text,
  TouchableOpacity,
  View,
} from 'react-native';

//#region plagins
import LottieView from 'lottie-react-native';
import RNFetchBlob from 'rn-fetch-blob'
//#endregion

//#region components
import { COLORS, images, SIZES, theme, tempData } from '../constants';
import { storage } from '../utils';
//#endregion

import { styles } from './styles'

const HomeScreen = ({ navigation }) => {

  const [nextFlag, changeNestFlag] = useState(false)
  const [getDataFlag, changeGetDataFlag] = useState(false)
  const [data, setData] = useState(null);

  const translate = useRef(new Animated.Value(0)).current
  const rotate = useRef(new Animated.Value(0)).current

```

```

const next = () => {
  if (!nextFlag) {
    changeNestFlag(true)

    Animated.timing(translate, { toValue: -(SIZES.width + SIZES.padding * 2 +
32), useNativeDriver: true, duration: 300 }).start()

    Animated.timing(rotate, { toValue: 1, useNativeDriver: true, duration: 300
}).start()
  }
  else {
    navigation.navigate('Scanner', { data: data })
  }
}

```

```

const skip = () => {
  navigation.navigate('Scanner', { data: data })
}

```

```

const getData = async () => {
  let answer = await RNFetchBlob
    .config({
      fileCache: true,
    })
    .fetch('GET', 'https://scanner-api.staj.fun//storage/file/file.txt', {
    })
  // res.json().then(json => {
  //   let tempGetData = {}
  //   console.log('The file data', json[0].nutritionalsupplement)
  // })
  let json = await answer.json()

```

```

//console.log(json);
let tempGetData = {}
json.map(item => {
  tempGetData[item.short_name] = item
})
setData(tempGetData)
storage.setData('data', 'true')
return tempGetData
}

useEffect(() => {
  storage.getData('data', true)
    .then(res => {
      if (res) {
        getData().then(data => {
          navigation.navigate('Scanner', { data: data })
        })
      }
      else {
        getData()
        changeGetDataFlag(true)
      }
    })
}, [])

return (
  <SafeAreaView style={styles.container}>

    <StatusBar backgroundColor={COLORS.primary} barStyle={'dark-content'}
  />

```

```

{
    getDataFlag ?
    <>
    {!nextFlag ?
        <LottieView source={images.scanning} autoPlay loop style={styles.gif}
/>

        :
        <LottieView source={images.recognizing} autoPlay loop
style={styles.gif} ref={ref=> ref ? ref.play() : null} />
    }
    <View style={styles.inner}>
        <View>
            <Animated.View
                style={[
                    styles.textWrapper,
                    { transform: [{ translateX: translate.interpolate({ inputRange: [0, 100],
outputRange: [0, 70] }) }] },
                ]}
            >
                <View style={styles.textInner}>
                    <Text style={[theme.FONTS.h5, styles.title]}>Сканирование</Text>
                    <Text style={[theme.FONTS.body_17_SF_R,
styles.text]}>Поднесите устройство к продукту, и отсканируйте состав</Text>
                </View>
                <View style={styles.textInner}>
                    <Text style={[theme.FONTS.h5, styles.title]}>Распознавание</Text>
                    <Text style={[theme.FONTS.body_17_SF_R, styles.text]}>Выберите
нужный ингредиент, оцените степень влияния на организм</Text>
                </View>
            </Animated.View>
            <View style={styles.dotsWrapper}>

```

```

    <Animated.View
      style={[
        styles.dotsInner,
        { transform: [{ rotate: rotate.interpolate({ inputRange: [0, 1],
outputRange: ['0deg', '-180deg'] }) }] }
      ]}
    >
      <View style={styles.dotsOne} />
      <View style={styles.dotsTwo} />
    </Animated.View>
  </View>
</View>
<View>
  <TouchableOpacity onPress={() => next()}>
    <Text style={[theme.FONTS.body_18_SF_SB,
styles.next]}>Далее</Text>
  </TouchableOpacity>
  <TouchableOpacity onPress={() => skip()}>
    <Text style={[theme.FONTS.body_14_SF_R,
styles.skip]}>Пропустить</Text>
  </TouchableOpacity>
</View>
</View>
</>
: <></>
}
</SafeAreaView>
);
};
paddingVertical: 15,
borderRadius: SIZES.radius,

```

```

    }
  }

  // ignored
  }

  //Вернуться назад и перезагрузить страницу
  _driver.Navigate().Refresh();
}
}
}
}
}

```

```

import { StyleSheet } from "react-native";
import { COLORS, SIZES } from "../constants";

export const styles = StyleSheet.create({
  container: {
    backgroundColor: COLORS.primary,
    flex: 1
  },
});

import React, { useEffect, useRef, useState } from 'react';
import {
  BackHandler,
  Image,
  ImageBackground,
  Modal,
  SafeAreaView,
  ScrollView,
  StatusBar,
  Text,
  TouchableOpacity,

```

```

    TouchableWithoutFeedback,
    View,
  } from 'react-native';

```

```

//#region plugins
import { RNCamera } from 'react-native-camera';
import AnimatedLottieView from 'lottie-react-native';
import MlkitOcr from 'react-native-mlkit-ocr';
import { useIsFocused } from '@react-navigation/native';
import DropShadow from "react-native-drop-shadow";
//#endregion

```

```

//#region
import { COLORS, images, theme, icons } from '../constants';
//#endregion

```

```

import { styles } from './styles'

```

```

const ScannerScreen = ({ navigation, route }) => {
  const isFocused = useIsFocused();
  const camera = useRef();
  const [timerCancelId, setTimerCancelId] = useState(false);
  const [processingFlag, changeProcessingFlag] = useState(false);
  const [flashFlag, changeFlashFlag] = useState(false)
  const [modalSearchFlag, changeModalSearchFlag] = useState(false);
  const [modalInfoFlag, changeModalInfoFlag] = useState(false)
  const [moreInfoFlag, changeMoreInfoFlag] = useState(false)

  const data = route.params.data
  const [resRecognize, setResRecognize] = useState(null);

```

```

const [additiveStatus, setAddiviStatus] = useState(0);

const textRecognize = async (img) => {
  const res = await MLkitOcr.detectFromUri(img.uri);
  console.log('res', res);
  let str = ""
  res.map(item => {
    str += item.text
  })
  let additiveArray = []
  let statusBad = 0
  let statusWarn = 0
  console.log('str', str);
  Object.values(data).map(item => {
    Object.values(item.items).map(item2 => {
      if (str.includes(item2.name) || str.includes(item2.name.replace('-', ''))) {
        if (item2.danger === 1) {
          statusWarn++
        }
        else if (item2.danger === 2) {
          statusBad++
        }
        additiveArray.push(item2)
      }
    })
  })
  if (statusBad >= 3) {
    setAddiviStatus(2)
  }
  else if (statusWarn >= 3 || statusBad !== 0) {

```

```

    setAddiviStatus(1)
  }
  else {
    setAddiviStatus(0)
  }
  setResRecognize(additiveArray)
}

const flash = () => {
  changeFlashFlag(!flashFlag)
}

const makePhoto = () => {
  camera.current.takePictureAsync()
    .then(img => {
      changeProcessingFlag(true)
      camera.current.pausePreview()
      setTimeout(() => {
        setTimerCancelId(null)
        textRecognize(img)
        .then(res => {
          changeProcessingFlag(false)
          camera.current.resumePreview()
        })
      }, 1500)
    })
    .catch(err => {
      console.log(err);
    })
  },
  1500
))

```

```

    })
    .catch(err => {
      console.log(err);
    })
  }

const cancel = () => {
  clearTimeout(timerCancelId)
  changeProcessingFlag(false)
  camera.current.resumePreview()
}

const list = () => {
  navigation.navigate('ListAdditive', { data: data })
}

const backAction = () => {
  if (isFocused) {
    BackHandler.exitApp()
  }
  else {
    return false
  }
};

useEffect(() => {
  BackHandler.addEventListener("hardwareBackPress", backAction);
  return () =>
    BackHandler.removeEventListener("hardwareBackPress", backAction);
}, [isFocused]);

```

```

useEffect(() => {
  if (resRecognize?.length > 0 && isFocused) {
    changeModalInfoFlag(true)
  }
  else if (resRecognize?.length === 0 && isFocused) {
    changeModalSearchFlag(true)
  }
}, [resRecognize, isFocused])

const ListAdditiveItem = ({ item }) => {
  return (
    <TouchableOpacity
      style={styles.modalInfoListInner}
      onPress={() => { changeModalInfoFlag(false);
        navigation.navigate('AdditiveInfo', { data: item }) }}
    >
    <Text
      style={[
        theme.FONTS.body_17_SF_R,
        styles.modalInfoListItem,
        {
          borderColor: item.danger === 0 ? COLORS.greenBorder : item.danger
            === 1 ? COLORS.orangeBorder : COLORS.redBorder,
          backgroundColor: item.danger === 0 ? COLORS.grayishGreen :
            item.danger === 1 ? COLORS.veryLightOrange : COLORS.verySoftRed,
        }
      ]}
    >
      {item.name}
    </Text>
  )
}

```

```

    </Text>
  </TouchableOpacity>
)
}

inner: {
  flex: 1,
  paddingLeft: SIZES.padding,
  paddingRight: SIZES.padding,
  flexDirection: 'column',
  justifyContent: 'space-between',
},
gif: {
  width: '100%'
},
textWrapper: {
  paddingTop: 11,
  flexDirection: 'row',
  //transform: [{ translateX: 0 }]
},
textInner: {
  width: SIZES.width - SIZES.padding * 2,
  paddingLeft: 32,
  paddingRight: 32,
},
title: {
  textAlign: 'center',
  color: COLORS.darkRed
},
text: {

```

```
    textAlign: 'center'
  },
  dotsWrapper: {
    flexDirection: 'row',
    justifyContent: 'center',
    paddingTop: 20,
    paddingBottom: 15,
  },
  dotsInner: {
    flexDirection: 'row',
  },
  dotsOne: {
    width: 20,
    height: 8,
    borderRadius: 17,
    backgroundColor: COLORS.strongRed,
    marginRight: 5
  },
  dotsTwo: {
    width: 8,
    height: 8,
    borderRadius: 17,
    backgroundColor: COLORS.lightGray
  },
  next: {
    textAlign: 'center',
    borderRadius: SIZES.radius,
    color: COLORS.primary,
    backgroundColor: COLORS.strongRed,
    paddingTop: 15,
```

```

paddingBottom: 15
},
skip: {
  textAlign: 'center',
  marginTop: 15,
  marginBottom: 50,
  color: COLORS.lightGray,
  textDecorationLine: 'underline'
}
}))

return (
  <SafeAreaView style={styles.container}>
    <StatusBar backgroundColor={COLORS.primary} barStyle={'dark-content'}
  />
    <RNCamera
      style={styles.cameraContainer}
      ref={camera}
      flashMode={flashFlag ? 'auto' : 'off'}
    >
      <View style={styles.substrateWrapper}>
        <View style={styles.substrateTop}>
          <Text style={[theme.FONTS.body_17_SF_R, styles.text]}>
            {'Поднесите устройство к продукту в раздел “Состав”\nНажмите на
кнопку'}
          </Text>
        </View>
        <View style={styles.substrateCenterWrapper}>
          <View style={styles.substrateCenterLine} />
          <View style={styles.substrateCenter}>
            <ImageBackground

```

```

        source={images.windowsScan}
        imageStyle={styles.substrateScanImg}
        style={styles.substrateScan}
    >

    </ImageBackground>
    <View style={styles.substrateCancel}>
        <Text>as</Text>
    </View>
</View>
<View style={styles.substrateCenterLine} />
</View>
<View style={styles.menuWrapper}>
    <ImageBackground style={styles.menu} imageStyle={styles.menuImg}
source={images.menuBackground}>
        <TouchableOpacity style={styles.flashBtnInner} onPress={flash}>
            <Image style={styles.flashBtn} source={flashFlag ? icons.flashOn :
icons.flashOff} />
        </TouchableOpacity>
        <TouchableOpacity style={styles.scanBtnInner} onPress={makePhoto}>
            <Image style={styles.scanBtn} source={icons.buttonScan} />
        </TouchableOpacity>
        <TouchableOpacity style={styles.searchBtnInner} onPress={list}>
            <Image style={styles.searchBtn} source={icons.list} />
        </TouchableOpacity>
    </ImageBackground>
</View>
</View>
</RNCamera>
</ScrollView>

```

```

        {resRecognize?.length > 6 ?
            <TouchableOpacity style={styles.modalInfoArrowInner} onPress={() =>
changeMoreInfoFlag(!moreInfoFlag)}>
                <Image source={icons.arrowDownUp} style={[styles.modalInfoArrow,
{ transform: [{ rotate: moreInfoFlag ? '180deg' : '0deg' }] }]} />
            </TouchableOpacity>
            : <></>
        }
    </View>
</View>
</Modal>

<Modal
    animationType="slide"
    transparent={true}
    visible={modalSearchFlag}
    //visible={true}
    onRequestClose={() => { changeModalSearchFlag(false);
setResRecognize(null) }}
>
    <View style={styles.modalWrapper}>
        <View style={styles.modalContainer}>
            <View style={styles.modalSearchTitleInner}>
                <DropShadow style={styles.modalCloseShadow}>
                    <TouchableOpacity style={styles.modalCloseInner} onPress={() => {
changeModalSearchFlag(false); setResRecognize(null) }}>
                        <Image source={icons.close} style={styles.modalClose} />
                    </TouchableOpacity>
                </DropShadow>
            </View>
            <Text style={[theme.FONTS.body_18_SF_R, styles.modalSearchText]}>

```

```
        Не удалось отсканировать состав
    </Text>
  </View>
</View>
</Modal>
</SafeAreaView>
);
};

export default ScannerScreen;
```

Додаток Д
Файл Students.xlsx

Vladislav Kuznetsov	Backend Node.js developer (GameDev) – Aussie Adrenaline	Криворожский Национальный Университет
Dmitry Moskalenko	Developer – DataArt	Криворожский национальный университет
Alexandr Fedorenko	Web Software Developer – IT Group	Криворожский Национальный Университет
Yuriy Drobniy	JS / React Native	Криворожский национальный университет
Aleksandr Pavlenko	C++ Developer - Globallogic	Криворожский Национальный Университет
Антон Хроленко	Computer Vision & Machine Learning Engineer	Криворожский национальный университет
Oleh Milha	Frontend Developer - Vinarux	Академия «Цар»
Руслан Соковев	C#/.NET	Криворожский Национальный Университет / Kyiv Rih National University
Надя Николенко	Преподавание – Компьютерная академия США Кривой Рог	Криворожский Национальный Университет
Андрей Кононенко	Участийся - Криворожский национальный университет	Криворожский национальный университет
Alexander Shulga	Front End Developer	National Aviation University
Iliya Zabolotskyi	RNR Developer / C# Developer / Chatbot Developer	STEP IT Academy
Олег Залуженко	Web Software Developer – Polytech Software	Национальный авиационный университет
Иван Залюда	Программист - DIO Trader International	Харьковский Национальный Университет Радиозлектроники
Богдан Мигилев	Software Developer at MacPaw Inc.	
Oleksandr Drozdenko	Senior System Administrator - STEP IT Academy (@tstep.org)	Криворожский Национальный Университет
Дмитрий Хоменко	Собственник "МЕТАЛ - ПРОЕКТ"	Криворожский Национальный Университет
Дарья Онищенко	Ассистент национальной дирекции	Криворожский национальный университет
Nikolay Kilyets	QA Lead at King Ukraine	Криворожский Национальный Университет
Ворников Денис	Специалист по работе с учебными заведениями - ПАО «Арселормиттал Кривой Рог»	Донецкий национальный университет им. М. Туган-Барановского
Maia Noholan	Руководитель региона (розница), руководитель отдела продаж.	Криворожский национальный университет
Виктор Ватри	People Partner – Intellis	Криворожский национальный Университет (бывший КЭИ КНЗУ)
Dmitry Matyusha	Android Developer	Криворожский Национальный Университет
Igor Kozlin	Программист 1С	Национальный криворожский университет
Олег Колобаев	ведущий специалист отдела энергоменеджмента – ЧАО "СЕВСОК"	Национальная металлургическая академия Украины
Дмитрий Терещенко	Full Stack Engineer – ERAM Systems	National Aerospace University N.E. Zhukovsky "Kraivik Aviation Institute"
Александр Р.	ENS Officer - Sandvik	Donesk National University
Татьяна Янова	Участийся - КНУ	Криворожский Национальный Университет
Станислав Демичев	Разработчик программного обеспечения (Unity, C#, Java) – Serveyalty	
Vladislav Shcherbina	Front-end Developer - Cwork	Криворожский Национальный университет
Anna Yatsun	IOS Developer	Криворожский национальный университет
Екатерина Горбачевич	Инженер-конструктор – АХЮМА	Криворожский национальный университет
Alex Sergea	Frontend Developer	Киууі Rih National University
Andrew B.	инженер-эколог	Криворожский национальный университет
Alexander Shevchenko	Front-end web developer	Криворожский национальный университет
Владислав Задорожний	Python-разработчик	Криворожский национальный университет

Додаток Е

Файл Students.xlsx

Дарья Кононенко	QA engineer		Криворожский национальный университет
Vacheslav Kovalchuk	Диплом специалиста - Одесская национальная академия связи им. А.С. Попова и Криво		
Святослав Ключенко	System administrator - SIM-NETWORKS		Криворожский национальный университет
Marina Mutsakina	Project manager at Customertimes		Криворожский национальный университет
Victor Tsiurko	Junior Quality Assurance Tester – Zafafasoft		Криворожский национальный университет
Elena Uziyva	QC Engineer		Государственное высшее учебное заведение "Криворожский национальный университет
Забара Сергеевич	Учащийся - Криворожский Национальный Университет		Криворожский Национальный Университет
Ilya Vityuk	Выпускник уч. заведения Криворожский национальный университет		Криворожский национальный университет
maxim mishuk	Научный сотрудник – Криворожский национальный университет		Криворожский национальный университет, аспирантура кафедры минералогии и петро
Мария Станчук	User Interface Designer		Криворожский Национальный Университет
Алексей Козар	Программист микроконтроллеров		Криворожский национальный университет
Наталья Дышлоук	find new opportunities		Skvot
Stas Chizhevsky	Engineer programmer - Vistgroup[UA]		
Yulia Samoilenko	business analyst		Криворожский национальный университет
Bohdan Polishchuk	Senior Visualizer – BOGZA		Криворожский Национальный Университет
Volodymyr Kalistatov	Senior Web Designer – Dydka Agency		Криворожский Национальный Университет
Serhiy Netrebko	Territory Sales Supervisor		Криворожский Национальный Университет
Mikhailo Shereliuk	Geologist - LP Deshpilsky MPP		Криворожский Национальный Университет
Yana Shcherbak	Студент (Криворожский Национальный Университет)		Криворожский национальный университет
Наталья Стулак	Студент (Криворожский национальный университет)		Криворожский национальный университет
Оксана Zinichak	Киевский национальный торгово-экономический университет (КНТЭУ)		Криворожский Национальный Университет
Иван Мельник	Прораб – Альтеко групп		Криворожский национальный университет
Serheiy Romaniou	Учащийся - КНУ		Криворожский национальный университет
Дмитрий Мартыненко	Инженер-электронщик – ООО "Охранный холдинг"		Криворожский национальный университет
Sofia Said	Smtt-manager		Криворожский национальный университет
Виктория Шлакова	smtt manager МАМАМ.ЦА		Криворожский Национальный Университет
Евгений Дерновой	Старший инженер отдела эксплуатации магистральной пакетной сети – ПАТ "Укртелеко		Криворожский национальный университет
Александр Дубовой	Инженер-программист - ТОВ "КРИВБАСРЕМОНТ"		Национальный университет экономики и технологий
Владимир Федорченко	брокер (Укркредит)		Криворожский Национальный университет
Гуда Андрей	Руководитель направления в PublicISC PrivatBank		Киевский Национальный Экономический Университет
Виталий Щербина	Заместитель руководителя направления		Днепропетровский национальный университет
Дарья Колодьяк	Преподаватель, переводчик		НПУ имени М.П. Драгоманова
Марина Копыцкая	Дизайнер пользовательского веб-интерфейса		КНУ Криворожский национальный университет
Denis Nazard	Junior Web Designer (Фриланс)		Криворожский Национальный Университет
Yaroslav Kebal	Руководитель отдела проектирования – K-MINE		Криворожский национальный университет
Алена Ш.	Сметчик – TARYAN Group		Криворожский Национальный Университет

Файл Students.xlsx

Davydov Daniil	3D artist	
Tetiana Tselyshcheva	UI/UX Designer	Projector School
Алина Александрова	Директор – Sportmaster	Криворожский Национальный Университет
Tetiana Melnykova	Junior Seo	Криворожский Национальный Университет
Kiril K	Facebook Media Buyer	Криворожский Национальный Университет
Вероника Р.	Recruiter	Криворожский Национальный Университет
Владимир Биличенко	Lean Manufacturing Manager – Metinvest	Криворожский национальный университет (ранее Криворожский технический универс
Andriew Bondarenko	Учащийся – Криворожский Национальный Университет	
Zhukova Liana	Экономист	Криворожский национальный университет (КНУ)
Mikhail Zavenko	Ведущий инженер автоматизированных систем управления – ПАО "ЮТОК"	Криворожский Национальный Университет
Irina Yimer	.	Криворожский национальный университет
Юлия Дьяченко	Специалист - Финансовые услуги	Криворожский национальный университет
Yevhen Nikolaenko	surveyor engineer	Криворожский национальный университет
Артём Репин	777777777777	Криворожский Национальный Университет, Криворожский строительный колледж
Юлия Бабич	модельер-конструктор, АС-ster	Криворожский национальный университет
Игорь Козеев	КНУ	Криворожский национальный университет
Юрий Бородин	Engineer of automatic & electric systems	Криворожский национальный университет
Vera Yastreb	marker - ArcelorMittal	Криворожский национальный университет
Наталья Недазельская	инженер ТТВ, "Оптимизация Инжиниринг"	Криворожский национальный университет (КНУ)
Minik Vladislav	--	Криворожский Национальный Университет
Ирина Концеропилова	Бухгалтер – ООО "ТОРГОВЫЙ ДОМ "КОМТЕК"	Криворожский Национальный Университет
Станислав Вельчев	Менеджер – ПП Мелиор	Криворожский Национальный Университет
Виталий Зарицкий	Специалист Машиностроение и производственная инженерия	Криворожский национальный университет
Юлия Манаенко	медиа-менеджер - Оксфорд Медикал-Крибас	Криворожский национальный университет
Кристина Яковенко	Бухгалтер - Парк Плюс	Криворожский Национальный Университет
Евгений Робота	Заместитель начальника фабрики по дробильному переделу ДОО – ПТОК	Криворожский национальный университет
Александр Григорьев	Специалист по контекстной рекламе (Google Ads, analytics, Яндекс Директ)	Криворожский национальный университет, Кривой Рог
Артём Жиглий	Буровик - ПАО ЮТОК	Криворожский Национальный Университет
Анна Миклашченко	Специалист по обучению и развитию в ювелирной компании "Золотой Век"	Криворожский национальный университет
Дмитрий Шишов	Региональный руководитель направления, тренер по продукту, HR – IG Electronics	Криворожский национальный университет
Наталья Малиренко	IT Business Analyst - Ericcent K	Институт последипломного образования "ПРОК"
Kostiantyn N.	iOS App Developer	Криворожский Национальный Университет
Нна Голуб	Начальник отдела кадров – ООО "МЕТИНВЕСТ-КРМЗ"	Криворожский национальный университет
Maksim Slezzyuk	Key Account Sales Manager - Vega Telecom Group (official)	Криворожский национальный университет
Александр Нисурин	Android Developer at DodoTар	Криворожский Национальный Университет
Sergo Qoval	the designer of your dreams	
ООО "Аутсорсинг-КР"	Услуги по охране труда и пожарной безопасности	Харьковская Государственная Академия Дизайна и Искусств криворожский национальный университет

Файл Students.xlsx

Елена Кузубова	Маркшейдер	Криворожский Технический Университет
Александр Наминат	Assistant of the department of Geodesy at Kyiv Riih National University	Университет новейших технологий
Anastasiya A.	автор научных статей – Avitfers.net	Криворожский национальный университет
Руслан Полюпан	и.о. ведущего геолога – ООО "Статус", ГП "Соник"	Криворожский национальный университет
Serhi Komishnek	Генеральный Директор - LLC PICO GROUP	Криворожский Национальный Университет
Sergey Lysoch	Senior Technical Project Manager – ПАО «Арселормиттал Кривой Рог»	Криворожский Национальный Университет
Наталья Торбина	Менеджер по обучению и развитию персонала	Криворожский экономический институт ГВУЗ Криворожский национальный университет
Дмитрий Вдовин	Инженер отдела капитального строительства – ЧАО "ЕВРАЗ СУХА БАЛКА"	Криворожский национальный университет
Ксения (Ксения Аралкина) Недашковская	Специалист Машиностроение и производства инженерии	Криворожский национальный университет
Liliya Uevstaterko	Commercial director LLC "INPRO"	КНУ
Анастасия Олейникова	Специалист Изобразительное искусство	Криворожский национальный педагогический университет
Людмила Лашкул	Главный бухгалтер – казначейство	Криворожский национальный университет
Alex Rapak	Graphic designer	Криворожский Национальный университет
Vladyslav Vanchenko	Ведущий маркетолог – Marketing Fathers	Криворожский Национальный Университет
Stanislav Steliedsov	Supervisor of Samsung Electronics Ukraine Contact Center	Криворожский Национальный Университет
Елена И.	Ведущий специалист по развитию персонала – ЧАО "СЕВЮК"	Криворожский национальный университет
Eugeniy Kotlevskiy	Business Development Manager – Eritos Ukraine	Киевский национальный торгово-экономический университет
Владимир Мурашкин	Движение вперед без границ	Криворожский Национальный Университет
Sergei Shubin	Заместитель начальника по подготовке производства Доменного цеха №2 агродомени	Национальная Металлургическая Академия Украины
Екатерина Романенко	Старший преподаватель, доцент - КНУ	Криворожский национальный университет (КНУ)
Виленин Антон	UI/UX Designer – Lindenvalley GmbH	Криворожский национальный университет
Mašina Rostoka	Директор – Артопартнер Днепр	Криворожский национальный университет
Дмитрий Матвеевко	PhD, с.наук.с. відділу наукового інформаційно-аналітичного супроводу освіти ДНПБ Укр	Криворожский национальный университет
Иван Дрогаль	Региональный руководитель - Stekloplast	Криворожский национальный университет
Максим Машков	Старший менеджер отдела операционных улучшений	Криворожский национальный университет
Валерий Василенко	Head of Programmatic – Resolution Media Ukraine	Криворожский национальный университет
Владимир Мован	Менеджер социальных проектов - Метинвест, Кривой Рог, "СЕВЮК"	Криворожский национальный университет (КНУ),
Марина Мироненко	Администратор ресторана – Софра	Криворожский национальный университет
Оксана Тасаркова	заведующая кабинетом НМЦ ПТО в Днепропетровской обл. – Н	Чарковский национальный университет им.Б.Хмельницкого
Антон Митрофанов	Инженер-конструктор I категории – Angelomital	Киевский национальный торгово-экономический университет (КНТЭУ)
Юлия Бондарец	Главный инженер - АПС-Кривой Рог	Криворожский профессиональный лицей
Елена Прокопенко	Учредитель предприятия по пошиву меховых и кожаных изделий	Криворожский Национальный Университет(КТУ)
Vitaly Leuchenko	специалист по ремонту ЧАО "ЦОК" ООО "Метинвест Холдинг"	Криворожский Технический Университет
Елена Бурлак	trade representative - PersiCo (Sandola)	Харьковский Национальный Университет Радиозлектроники
Aleksy Mezensev	Совладелец – fitcupives Ukraine and Eastern Europe	Национальный авиационный университет
Елена Дюмина	Специалист горнодобывающая и металлургическая промышленности	Киевский национальный экономический университет им. В.Гетьмана
Людмила Пилипенко	Graphic design, illustration.	Art&craft school
	Менеджер по персоналу – Gazy Group	Запорожский Национальный университет