ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Системного аналізу

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «**СИСТЕМА АНАЛІЗУ РЕЙТИНГІВ ВАКАНСІЙ, РОЗТАШОВАНИХ НА САЙТАХ ПОШУКУ РОБОТИ**»

Виконав: студент 4 курсу, групи САД-41
спеціальності 124 Системний аналіз
(шифр і назва спеціальності)
Кожедуб В.О.
(прізвище та ініціали)
Керівник Штіммерман А.М.
(прізвище та ініціали)
Рецензент_____
(прізвище та ініціали)

Київ – 2020

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Системного аналізу
Ступінь вищої освіти - «Бакалавр»
Спеціальність – 124 Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедри
Системного аналізу
Золотухіна О.А.
“ _____” 2020 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Кожедуба Віталія Олеговича
(прізвище та ініціали)

1.Тема роботи: «СИСТЕМА АНАЛІЗУ РЕЙТИНГІВ ВАКАНСІЙ, РОЗТАШОВАНИХ НА САЙТАХ ПОШУКУ РОБОТИ»

Керівник роботи Штіммерман Аксєнія Миколаївна, старший викладач кафедри
(прізвище та ініціали, науковий ступень, вчене звання)
системного аналізу
затверджені наказом вищого навчального закладу від “ ___” 2020 року №_____

2.Строк подання студентом роботи_____

3.Вхідні дані до роботи:
Статистичні дані з сайтів працевлаштування, інформація з баз вакансії

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1 АНАЛІЗ БІЗНЕС-ПРОЦЕСІВ ОПЕРАТИВНОЇ АНАЛІТИКИ ДАНИХ
4.2 РОЗРОБКА ДОДАТКА ДЛЯ ЗБОРУ ІНФОРМАЦІЇ ПРО ВАКАНСІЇ З РІЗНИХ ДЖЕРЕЛ ПОШУКУ РОБОТИ
4.3 АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ
4.4 ВИСНОВКИ

5.Перелік графічного матеріалу

1. Індивідуальне завдання
2. Актуальність бакалаврської роботи
3. Об’єкт, предмет, мета бакалавської роботи

4. Огляд інформаційного забезпечення в галузі пошуку вакансій
5. Аналіз інфокомунікаційних рішень, що використовуються на поточний момент в галузі пошуку вакансій
6. Недоліки та проблемні питання в існуючих рішеннях пошуку вакансій.
7. Технології створення аналізу вакансій, зібраних на сайтах працевлаштування.
8. Модель бази даних системи аналізу рейтингів вакансій
9. Діаграма взаємодії класів системи і таблиць бази даних
10. Діаграма компонентів системи аналізу рейтингів вакансій
11. АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ
12. Висновки

6.Дата видачі завдання 08 травня 2020 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	08.05.2020	Виконав
2	Визначення показників ефективності аналітики сайтів пошуку роботи	12.05.2020	Виконав
3	Визначення недоліків та проблемних питаннях у існуючих системах аналітики вакансій	15.05.2020	Виконав
4	Аналіз моделей та рішень у існуючих системах аналітики вакансій	17.05.2020	Виконав
5	Пошук рішення та механізмів реалізації додатка аналізу вакансій	19.05.2020	Виконав
6	Моделювання інформаційної системи	20.05.2020	Виконав
7	Побудова Бази даних	22.05.2020	Виконав
8	Створення парсера	28.05.2020	Виконав
9	Аналіз сайтів працевлаштування	01.06.2020	Виконав
10	Висновки	05.06.2020	Виконав
11	Оформлення пояснювальної записки	10.06.2020	Виконав

Студент _____Кожедуб В.О.

Керівник практики _____ Штіммерман А.М.

ЗМІСТ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

ПОДАННЯ

ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ ЩОДО ЗАХИСТУ БАКАЛАВРСЬКОЇ РОБОТИ

Направляється студент_ Кожедуб В.О.. до захисту бакалаврської роботи
(прізвище та ініціали)
за спеціальністю 124 Системний аналіз
(шифр і назва спеціальності)
на тему: «система аналізу рейтингів вакансій, розташованих на сайтах пошуку роботи».
Бакалаврська робота і рецензія додаються.
Директор інституту _____ А.П. Бондарчук
(підпис)

Довідка про успішність

Кожедуб В.О за період навчання в Навчально-науковому інституті інформаційних
(прізвище та ініціали студента)
технологій з 2016__року до 2020___ року повністю виконав навчальний план за спеціальністю з таким розподілом оцінок за:
національною шкалою: відмінно ____%, добре ____%, задовільно ____%;
шкалою ECTS: A ____%; B ____%; C ____%; D ____%; E ____%.
Методист факультету _____ Алексіна Л.Т.
(підпис) (прізвище та ініціали)

Висновок керівника бакалаврської роботи

Студент (ка) Кожедуб Віталій Олегович
_____
_____
_____
_____
Все це дозволяє оцінити виконану бакалаврську роботу студента Кожедуба В. О. на оцінку «_____» та присвоїти йому кваліфікацію фахівець з інформаційних технологій.

Керівник роботи _____ ст.вик. Штіммерман А.М.
(підпис)
“___”_____2020 року

Висновок кафедри про бакалаврську роботу

Бакалаврську роботу розглянуто(а). Студент (ка) _ Кожедуб В. О.
допускається до захисту даної роботи в Державній екзаменаційній комісії.
Завідувач кафедри Системного аналізу__ _____ ___О.А. Золотухіна

(підпис) (прізвище та ініціали)

“___” _ ____2020 рок

РЕФЕРАТ

Текстова частина бакалаврської роботи 55 с., 6 табл., 28 рис., 2 дод., 23 джерела.

АНАЛІЗ, РЕЙТИНГ ВАКАНСІЙ, СКБД, MVC , POSTGRESQL, PHP, SQL ,

Об’єктом дослідження, є процес автоматизації збору та аналізу даних з сайтів працевлаштування. Вхідними даними, є інформація з різних джерел, а вихідними — програмне рішення.

Предметом дослідження, виступають методи проектування та технології створення додатка для аналізу вакансій, зібраних на сайтах працевлаштування.

Мета роботи – підвищення якості автоматизованого процесу аналітики сайтів пошуку роботи для визначення найбільш популярних професії, аналіз попиту компетенцій галузі інформаційних технологій.

Методи дослідження – обробка та аналіз отриманих даних з різних джерел працевлаштування.

Визначено, що системи обробки і пошуку вакансій, розташовані на великої кількості ресурсів, недоліками існуючих рішень аналізу, є розрізненість даних на сайтах працевлаштування, що потребує додаткового час обробки контенту і виявлення статистичних даних.

В бакалаврської роботі здійснена розробка додатка для збору інформації про вакансії з різних джерел пошуку роботи, що прискорить обробку зібраної інформації з сайтів працевлаштування та створення статистики за запитом користувача про популярність вакансій, заробітну плату.

Показниками ефективності, дипломного додатка, є прискорення часу на збереження та обробку даних, отриманих з різних джерел пошуку роботи, спрощення взаємодії користувача з великою кількістю сайтів та зручність використання отриманих даних.

ВСТУП

У сучасних умовах, важливо мати уявлення про попит різних спеціалістів та можливих тенденціях працевлаштування. Для цього потрібно розумітися на інструментах аналізу і вміти оцінювати зміни у потребі тих чи інших вакансій. Щоб володіти сучасною інформацією у галузі працевлаштування, необхідно вирішити питання моніторингу і обробки інформації з різних джерел пошуку вакансій.

Для здійснення інформаційного моніторингу, необхідна система, яка відповідає вимогам сучасних технології аналізу.

Об’єктом бакалаврської роботи, є процес автоматизації збору та аналізу даних з сайтів працевлаштування. Вхідними даними, є інформація з різних джерел, а вихідними — програмне рішення.

Предметом дослідження, виступають методи проектування та технології створення додатка для аналізу вакансій, зібраних на сайтах працевлаштування.

Мета роботи – підвищення якості автоматизованого процесу аналітики сайтів пошуку роботи для визначення найбільш популярних професії, аналіз попиту компетенцій галузі інформаційних технологій.

Для досягнення поставленої мети мають бути вирішені такі задачі:

1. Проаналізувати бізнес-процеси в галузі інформаційного забезпечення для оперативної аналітики даних працевлаштування;
2. Проаналізувати та порівняти показники ефективності розглянутих бізнес-процесів сайтів пошуку роботи;
3. Описати моделі бізнес-процесів сайтів пошуку роботи, в тому числі в графічній нотації;
4. Сформулювати критерії оцінки систем аналітики вакансій;
5. Визначити недоліки та проблемні питання у існуючих системах аналітики вакансій;
6. Спроектувати та розробити систему аналізу рейтингів вакансій, розташованих на сайтах пошуку роботи.

Завданням бакалаврської роботи, є розробка додатка для збору інформації про вакансії з різних джерел пошуку роботи, що прискорить обробку зібраної інформації з сайтів працевлаштування та створення статистики за запитом користувача про популярність вакансій, заробітну плату.

Показниками ефективності, дипломного додатка, є прискорення часу на збереження та обробку даних, отриманих з різних джерел пошуку роботи, спрощення взаємодії користувача з великою кількістю сайтів та зручність використання отриманих даних.

Бакалаврська робота є самостійним дослідженням, одержані результати співпадають з метою та мають практичне значення. Узагальнено та класифіковано попит вакансій і мов програмування на ринку інформаційних технологій.

1. АНАЛІЗ БІЗНЕС-ПРОЦЕСІВ ОПЕРАТИВНОЇ АНАЛІТИКИ ДАНИХ З САЙТІВ ПОШУКУ РОБОТИ

1.1 Аналіз існуючих рішень

Оперативна аналітика та управління інформацією допомагає виявити і розкрити цінність, приховану глибоко всередині великого обсягу даних. Одним з прикладів такої аналітики, є збір даних з різноманітних ресурсів пошуку вакансій, включаючи сайти пошуку роботи, які працюють з різними методами представлення інформації і аналіз зібраних даних.

Задачею переддипломної практики, було дослідити процес оперативної аналітики даних для сайтів пошуку роботи.

Переважно, аналіз вакансій проходить через API клієнта, запит-відповідь взаємодіючи з базами даних. На рисунку 1.1, представлена архітектура клієнт-серверної взаємодії з СКДБ (Система керування базами даних).

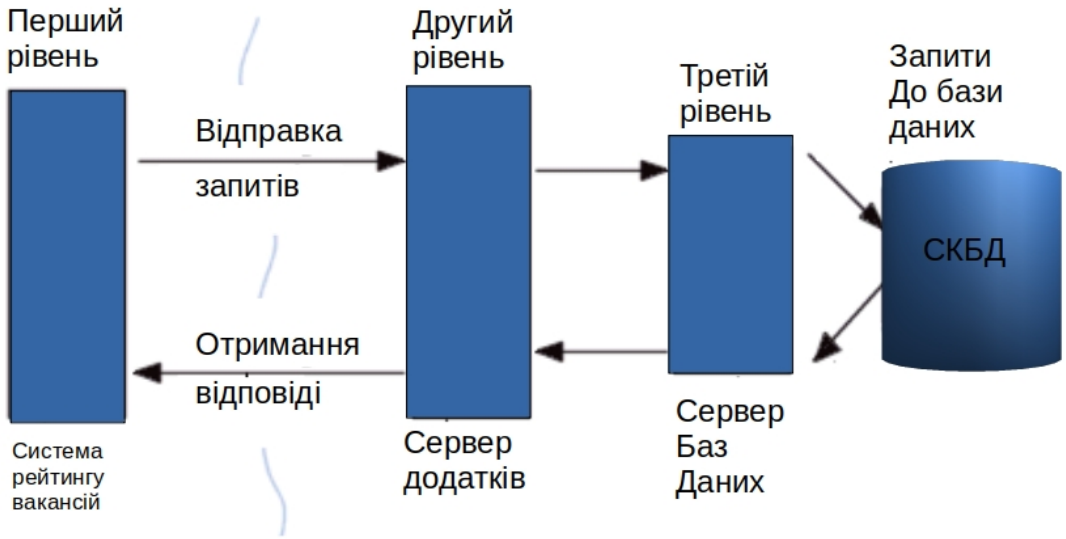


Рисунок 1.1 – Робота Client -> Server -> Database

API (Application Programming Interface), це сукупність засобів та правил, що вможливлюють взаємодію між окремими складниками програмного забезпечення або між програмним та апаратним забезпечення. [1] Більшість великих компаній створили API для своїх клієнтів або для внутрішнього використання. Коли ви вводите адресу http://www.dut.edu.ua/ у свій браузер, на віддалений сервер

DUT.EDU.UA надходить запит. Як тільки браузер отримає відповідь, він інтерпретує код і відображає сторінку. Для браузера (клієнта), сервер DUT.EDU.UA- це АР. Це означає, що кожного разу, коли відвідуєтеться сторінка в Інтернеті, відбувається взаємодієте з API деяких віддалених серверів.

API не є віддаленим сервером, це частина, яка приймає запити та надсилає відповіді. Набір запитів (на рисунку 1.2, наведено взаємодію API):

GET -> отримати всі данні які містяться у базі даних.

POST -> записати якісь данні у базу даних.

GET{id} -> знайти запис у базі по унікальному id.

PUT{id} -> замінити запис у базі по унікальному id.

DELETE{id} -> видалити запис із бази даних по унікальному id.

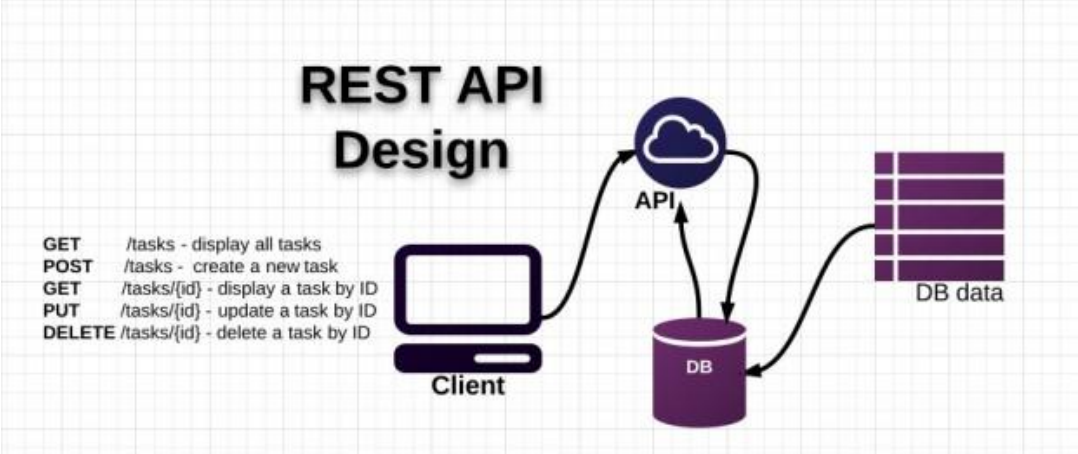


Рисунок 1.2 – Взаємодія API

У таблиці 1.1 наведено відповідність операцій в SQL та HTTP.

Таблиця 1.1 - Відповідність операцій створення-читання-оновлення-видалення в SQL та HTTP.

OPERATION	SQL	HTTP
CREATE	INSERT	POST
READ	SELECT	GET
UPDATE	UPDATE	PUT/PUTCH

DELETE	DELETE	DELETE

1.2. Аналіз рішень забезпечення інформаційно-аналітичних процесів на сайтах працевлаштування

Незважаючи на широке використання API, недоліком даної технології є показник швидкості, щодо збору ресурсів з великої кількості джерел. Що в свою чергу знижує ефективність. Тому для вилучення даних будь-якого сайту в мережі Інтернет, використовують парсенг. Веб-парсер сканує веб-сторінки, завантажує "контент", отримує з нього потрібні дані і потім зберігає їх в файлах або базі даних.

Парсингом називають процеси, розроблені ресурсом, що складаються з подальшими видами даних. Ресурси можуть бути на різних сайтах, документи в різних форматах, зображення. [2]

Парсер, насамперед це скрип написаний на мові програмування. Кожен сайт котрий є в мережі, складається з коду, який інтерпретує браузер у зовнішній вигляд, представлений користувачеві.

Розглянемо використння HTML DOM парсер для отримання даних. Для використання парсера, необхідно створити БД і інструмент додавання до неї таблиць, окремо для кожного сайту з якого потрібно отримувати данні і записувати їх в таблиці. На рисунку 2.1 наведено приклад створення таблиць.

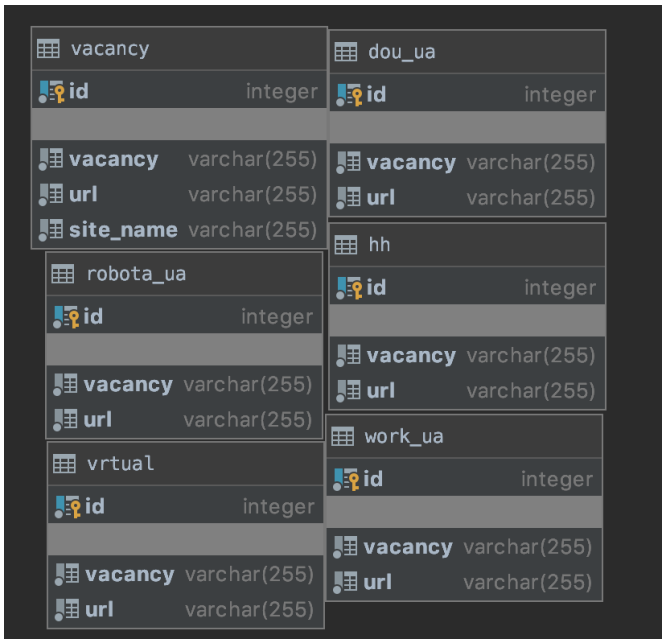


Рисунок 2.1. - Приклад таблиці в базі даних

До переваг використання парсерів відноситься:

- Економія часу на етапі розробки, зовнішнього обслуговування.

- Для парсерів може бути створено розклад для запусків, тож вам не потрібно буде їх запускати вручну.

- Парсери можуть працювати в хмарі.

- Після компіляції, завантажений парсер буде повністю автономним і може запускатись з будь-якого місця скільки завгодно.

- У сценарії парсера, можливо використовувати фрагменти коду JS.

- Створені дані перед записом у набір даних можуть бути автоматично опрацьовані з допомогою JSON-схем.

Проаналізувавши методи представлення вакансій на сучасних сайтах пошуку роботи та проаналізувавши функціонал аналітики, можливо навести приклад одного з web-сайтів, який надає послуги пошуку роботи та на його прикладі виявити недоліки, які існують на даний час.

Розглянемо аналог – аналітика вакансій на сайті HH.UA. (рисунок 2.2)

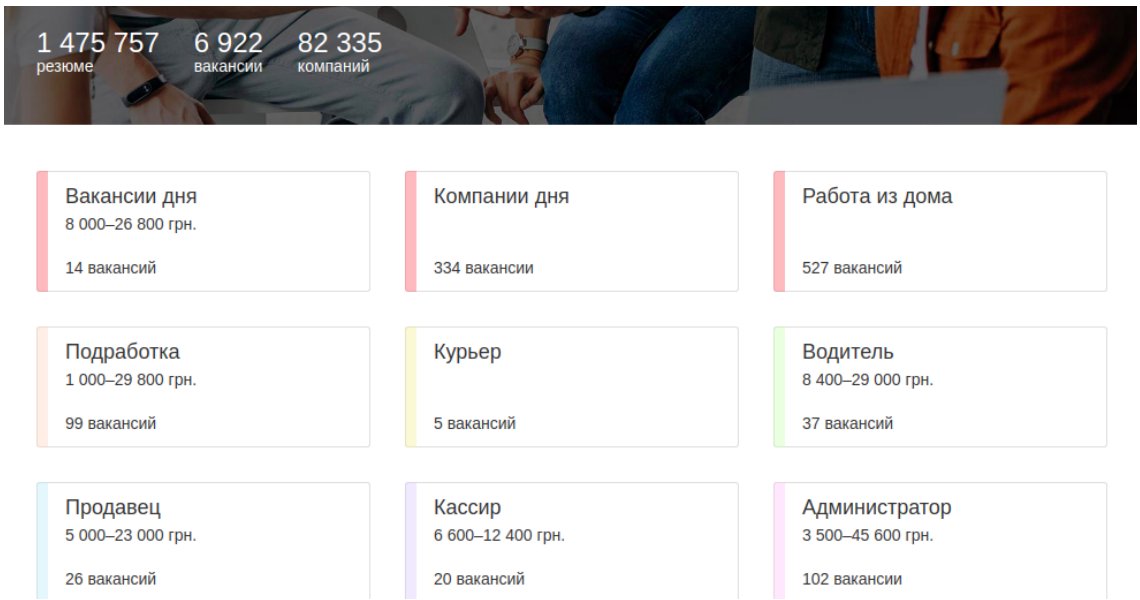


Рисунок 2.2 – Приклад сайту аналітики вакансій.

Для пошуку вакансій за різними параметрами є метод GET /vacancies:

curl -A 'irenica (https://irenica.com/)' 'https://api.hh.ua/vacancies'

Результати пошуку містять дані про вакансії у вигляді, визначеному на сайті і нема можливості будувати графіки та отримати аналітику по будь-якому запиту.

Часткові дані про вакансії знаходяться в полі *items* результатів пошуку:

declare vacanciesIds="$(echo "$page" | jq -r '.items[].id')"

Для того, щоб отримати повну інформацію про відповідні вакансії окремо, потрібно впроваджувати додаткові скрипти:

for vacancyId in $vacanciesIds; do declare url="https://api.hh.ru/vacancies/$vacancyId" declare vacancy="$(curl -A 'irenica (https://irenica.com/)' "$url")" # обрабатываем $vacancy done

Варто звернути увагу, що повертаються вакансії, опубліковані тільки за останній місяць.

Таким чином, щоб отримати інформацію для повноцінної аналітики, необхідно впроваджувати додаткові інструменті, такі як скрипти у форматі Json за заданими критеріям. І тоді є можливість проводити аналітику та попит на вакансій цього сайту.

Ще одним прикладом аналізу сайтів пошуку роботи, є результат аналітики, проведеної сайтом «ХАБР». Приклад графіку, наведено на рисунку 2.3.

Графіки показують інформацію про:

- відсоток роботи з дому;
- відсоток вакансій для студентів;
- попит у працівниках з інвалідністю;
- загальний рейтинг вакансій.

Як видно з графіків, це тільки та аналітика, яку представили на сайті і нема можливості самостійно обирати параметри для аналізу.

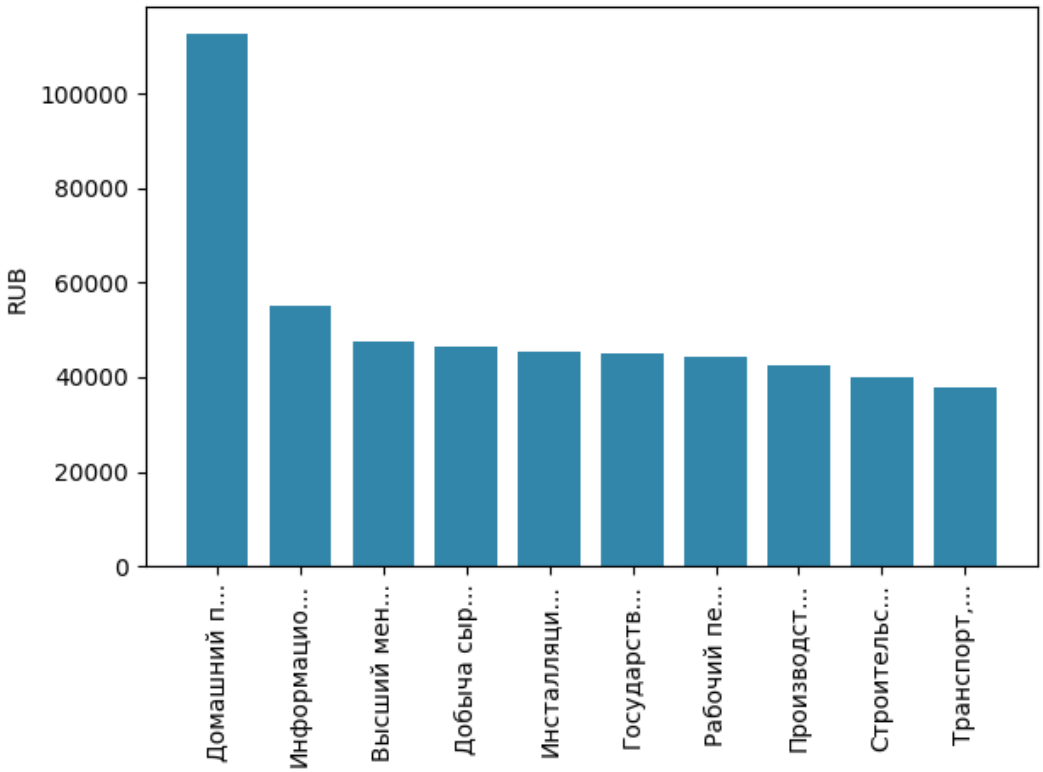


Рисунок 2.3 – Результат аналізу вакансій на сайті «ХАБР»

Таблиця 2.1 – Рейтинг вакансій сайту «Хабр»

Name	Salary, average	Salary, minimum	Salary, maximum	Number
Высший менеджмент	78789	150	2000000	2408
Добыча сырья	61699	8000	180000	2302
Консультирование	59797	1500	500000	3762
Информационные технологии, интернет, телеком	52777	26	684804	25900
Строительство, недвижимость	48587	20	949989	33229
Производство	42007	1	261000	27269
Рабочий персонал	41203	25	200000	43079

Автомобильный бизнес	38555	20	824254	9269
Инсталляция и сервис	38412	25	180000	2390
Закупки	37846	50	261000	2658

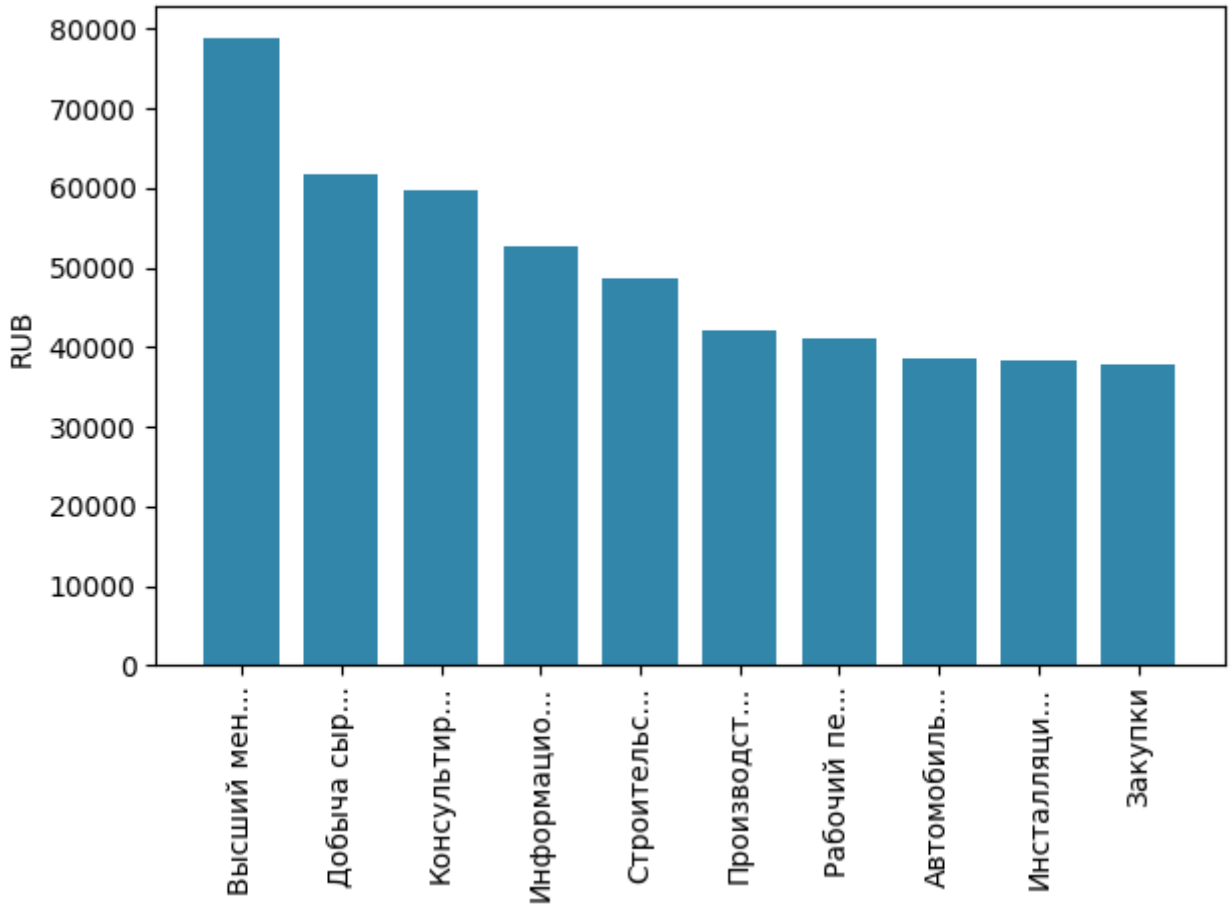


Рисунок 2.4 – Результат аналізу вакансій на сайті «ХАБР» Загальний топ.

1.3 Висновки до розділу

Якщо потрібно обробляти багато вакансій і проводити аналітику автоматично, наприклад графічно у вигляді графіка порівнювати з тисяч вакансій всіх можливих сайтів. Зокрема, якщо потрібно збирати інформацію з усіх можливих сфер заробітку, то існуючі варіанти пошуку сайтів не підійдуть, тому що відповідь повертається у дуже тісному діапазоні і немає повного доступу до

ресурсів API сайту для фільтрації та переробки для наших цілей. Тому потрібно впроваджувати сучасні технології обробки і аналітики даних.

2. РОЗРОБКА ДОДАТКА ДЛЯ ЗБОРУ ІНФОРМАЦІЇ ПРО ВАКАНСІЇ З РІЗНИХ ДЖЕРЕЛ ПОШУКУ РОБОТИ

2.1 Моделювання інформаційної системи та бази даних

Для розробки системи аналізу рейтингів вакансій, розташованих на сайтах пошуку роботи, необхідно провести етап моделювання системи та бази даних. На рисунку 2.1, представлена схема взаємодії компонентів інформаційної системи, операції та взаємодія з базами даних.

Інформаційна система, збирає дані з доступних у глобальній мережі ресурсів, аналізує, модифікує, зберігає на локальній базі даних, та виводить результат аналізу, користувачеві.

Аналіз інформації, ґрунтується на запитах вибірки даних і обробки отриманих результатів. До елементів вибірки, належать професії та основні компетенції, тобто знання мов програмування. Вибірка даних проводиться як з заздалегідь внесеними назвами, так і унікальними запитами обробки даних, внесеними користувачем.

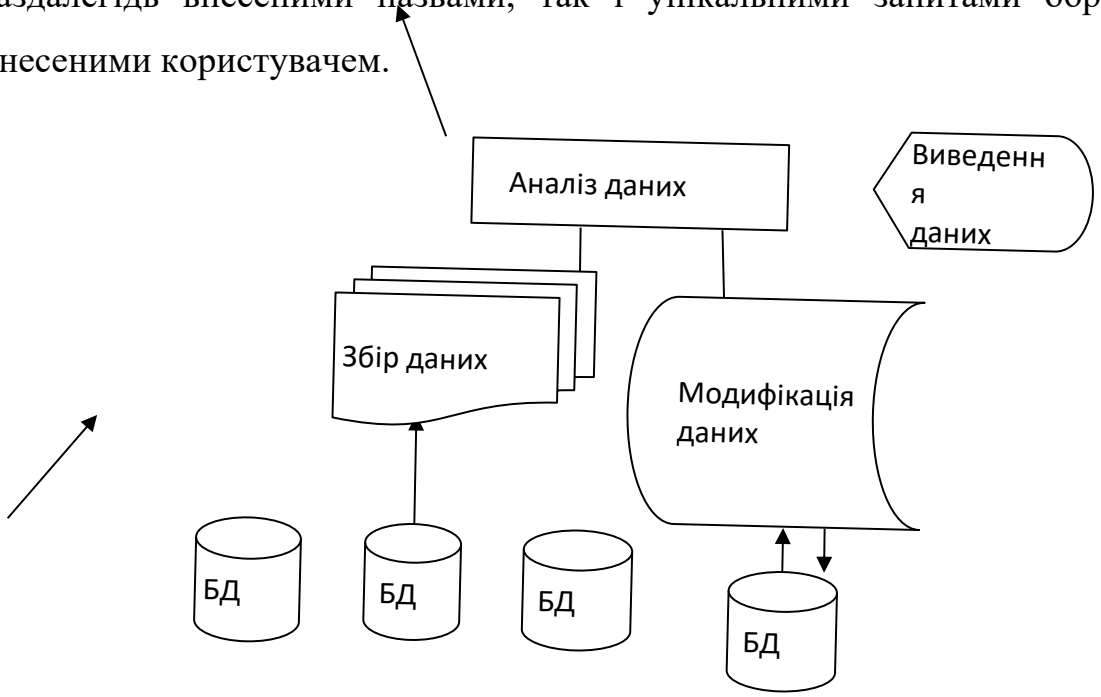


Рисунок 2.1 — Структура інформаційної системи

Для зберігання інформації, отриманої з різних сайтів працевлаштування, доцільно використовувати локальну СКБД. Для проектування бази даних, необхідно обрати СКБД, яка б відповідала певним вимогам, а саме швидкістю, надійністю, масштабованістю.

СКБД PostgreSQL, базується на мові SQL і володіє всіма перерахованими вимогами. На рисунку 2.2 показана модель бази даних, для зберігання отриманої інформації з сайтів працевлаштування.

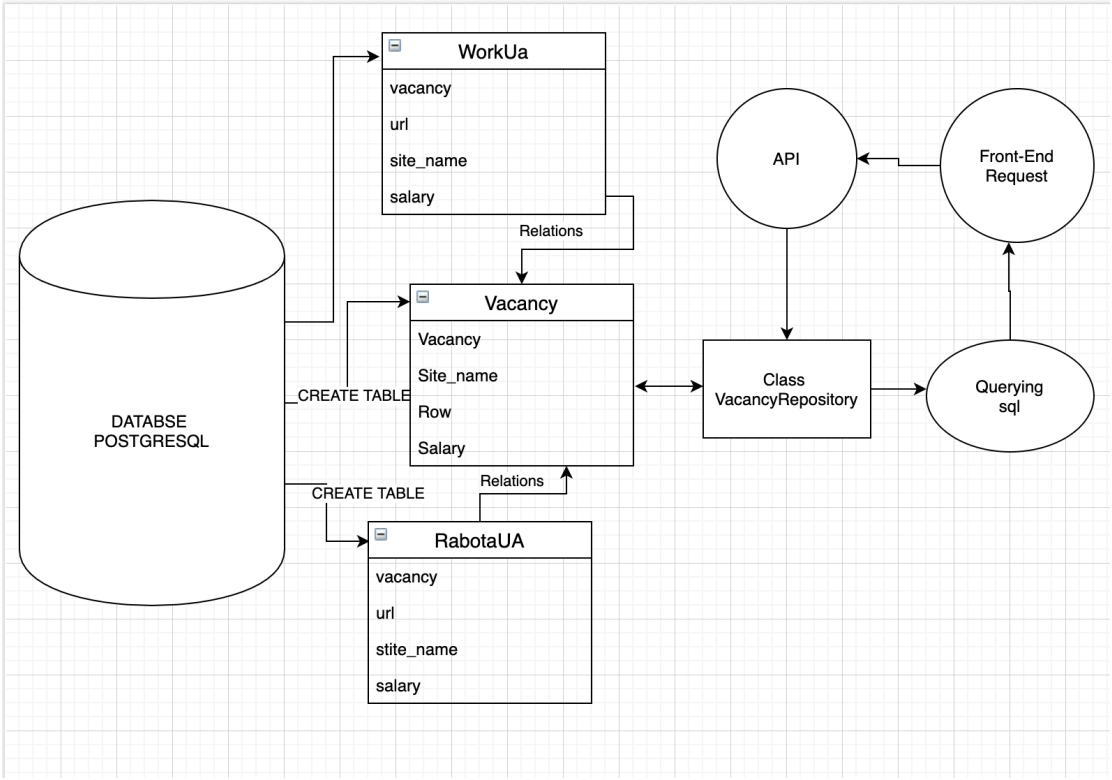


Рисунок 2.2 — Модель бази даних системи аналізу рейтингів вакансій

Основними компонентами бази даних системи аналізу рейтингів вакансій, є таблиці, у яких зберігатимуться дані про вакансії. Кожна таблиця використовується для структуризації даних одного з сайтів. Взаємозв'язки між таблицями, показані на рисунку 2.2 і визначають зв'язок між сутностями інформаційної системи. Спроектована база даних, містить стільки таблиць, скільки сайтів

працевлаштування буде надано. При розробці бази даних даних системи аналізу рейтингів вакансій, доцільно використовувати наступну інформацію:

1. Назву вакансії;
2. Адресу ресурсу;
3. Назву ресурсу;
4. Заробітну плату;
5. Завдання первинних ключів.

2.2 Побудова таблиць бази даних

Після єтапу проектування концептуальної моделі даних, йде процес сворення БД, а також побудова таблиць для запису даних в них.

Кожна таблиця бази даних системи аналізу рейтингів вакансій, має поля “vacancy” для запису вакансії с того сайту який їй належить, “id”- унікальний для знаходження записів, “url”- для запису прямого посилання на вакансію. І є 1 загальна таблиця, яка має всі ті самі поля і поле “site_name” у яке записується назва сайту, на якому знаходиться вакансія, і поле “salary” із заробітною платою.

На рисунку 2.3, представлена діаграма, відображаюча класи інформаційної системи. Кожна таблиця у схемі розглядається як клас (Class). Тобто нам на кожну таблицю потрібен свій клас (Class).

Тобто кожна таблиця створена через Class, має методи у класі , кожен метод це поле у таблиці, наприклад:

public function $id; - це поле у таблиці id;

має додаткові методи такі як:

public function $getId; - через об'єкт отримати данні с поля id;

public function $setId; - через об'єкт записати данні у поле id;

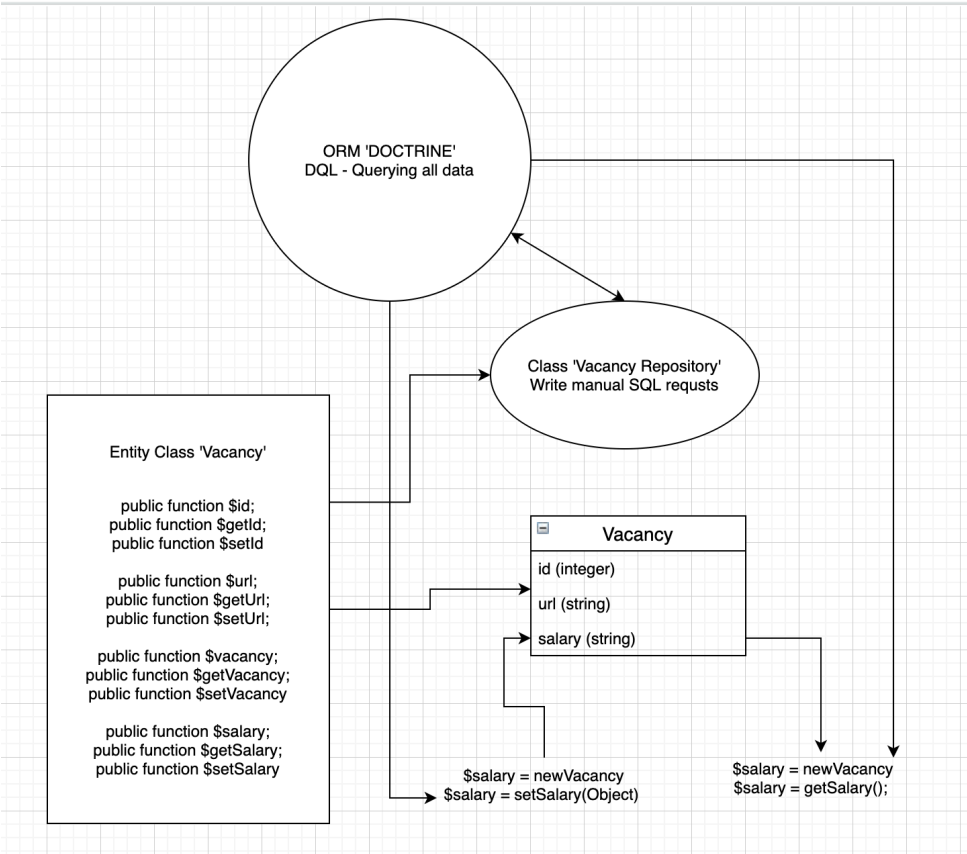


Рисунок 2.3 — Діаграма взаємодії класів системи і таблиць бази даних

На рисунку 2.4 показана діаграма взаємодії сутностів і об'єктів створеної системи.

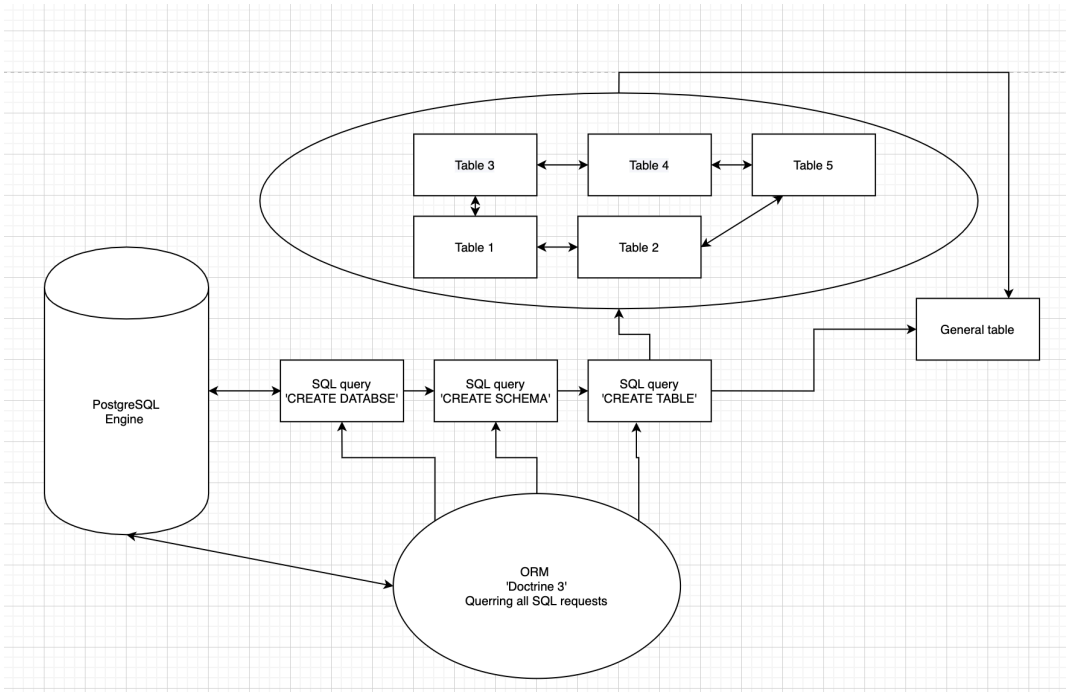


Рисунок 2.4 — Діаграма бази даних і таблиць

2.3 Створення Інтернет додатка системи аналізу рейтингів вакансій

При виборі мови програмування, слід визначити критерії, за якими мова програмування, буде відповідати поставленій задачі. У даному випадку, система, взаємодіє к користувачем, завдяки клієнт-серверній архітектурі і використовує глобальну мережу Інтернет. Швидким, надійним і простим рішенням для реалізації Інтернет-додатка, є мова PHP, останньої сьомої версії.

Для спрощення написання коду, та для використання бібліотек і засобів вбудованої безпеки, було обрано платформу для програмування: фреймворк Symfony.

Для розробки Інтернет-додатка системи аналізу рейтингів вакансій потрібно написати парсер, який буде збирати данні з різних Інтернет-джерел, такі як вакансії і заробітна плата і записувати в таблиці створеної бази даних. На кожну таблицю, реалізовано окремий парсер.

На рисунку 2.5, представлено принцип роботи парсеру.

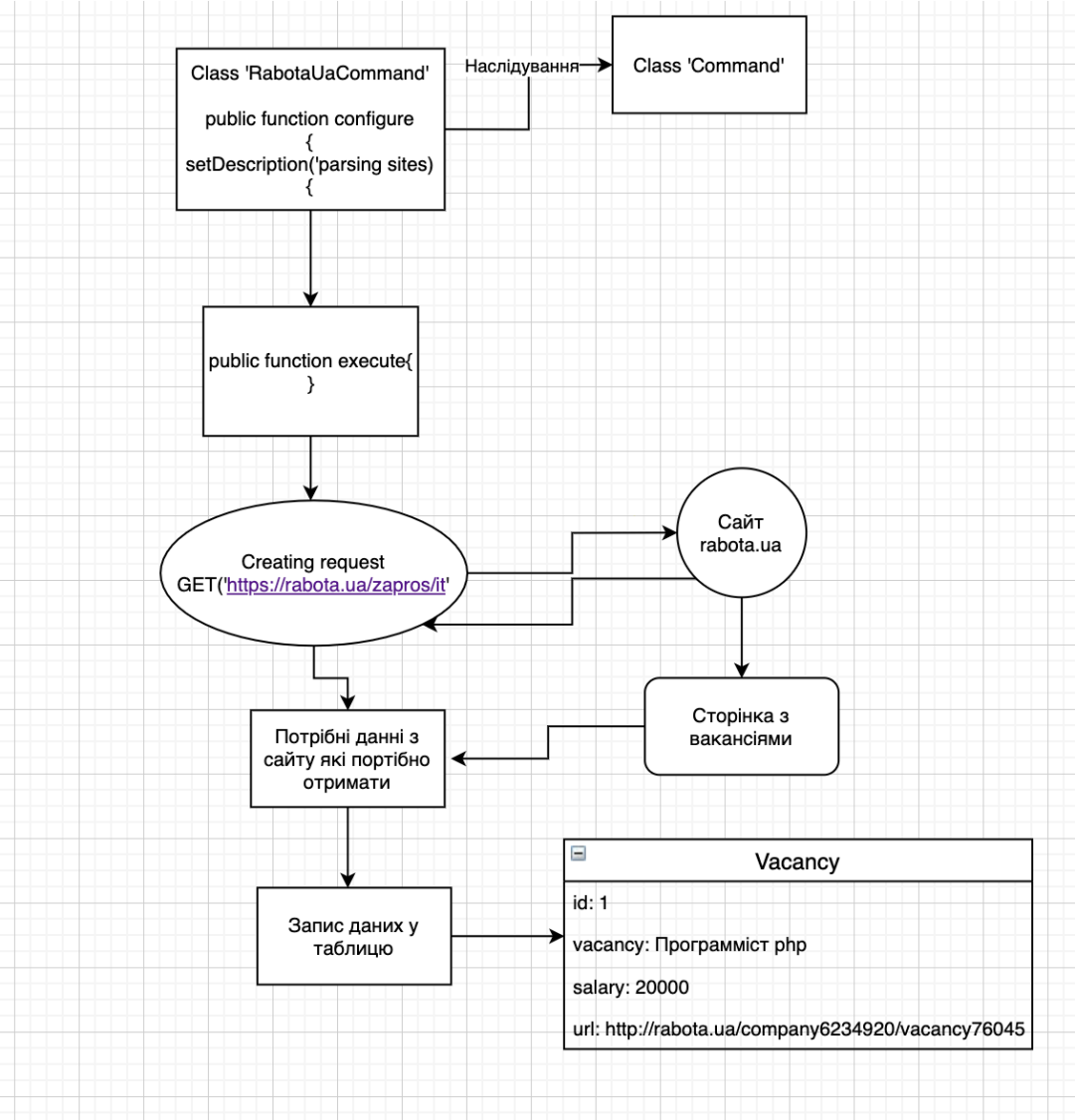


Рисунок 2.5 —Принцип роботи парсера

Для початку нам потрібно створити Bash команду яка буде запускати прасер.

Метод створення Bash команди. Перевіряємо.

На рисунку 2.6, показана команда, яка створена і є у терміналі.

```
RabotaUa                                    parsing sie RabotaUa
```

Рисунок 2.6 —Метод створення Bash команди

Наступним кроком0, є додавання функціоналу створеної команди. Для процесу парсингу сайту, будемо використовувати метод css-selctors для отримання блоків об'єктної моделі документа (DOM-Element), які потрібні для функціювання Інтернет-додатка. Для доступу до сайту, будемо використовувати запит (http-request). На сайті не одна сторінка , для цього потрібно зациклити запит (request) і посилати його до тих пір поки не повернеться помилка. Переходимо на сайт пошуку роботи, знаходимо вакансії які нас цікавлять, виділяємо поле – після чого відкриваємо браузерну консоль, знаходимо потрібній блок, тобто елемент div , і отримуємо його елемент: css-selector.

Після отримання необхідних даних, робимо перебір всіх вакансій з цього селектору за допомогою циклу: foreach. Одна вакансія витягнута за допомогою селектора це лише 1 вакансія на сайті, а нам потрібно отримати всі. За допомогою циклу “foreach”, парсер буде шукати всі схожі вакансії с цим елементом (css-selector), а не лише 1 з якої ми його дістали.

Після отримання всіх необхідних даних, потрібно їх записати у таблицю БД.

На рисунку 2.7, показано результат перевірки парсера. На сайті 217 сторінок тобто парсер обпрацював 217 сторінок і на 218 закінчити роботу.

Рисунок 2.7— Результат перевірки парсера

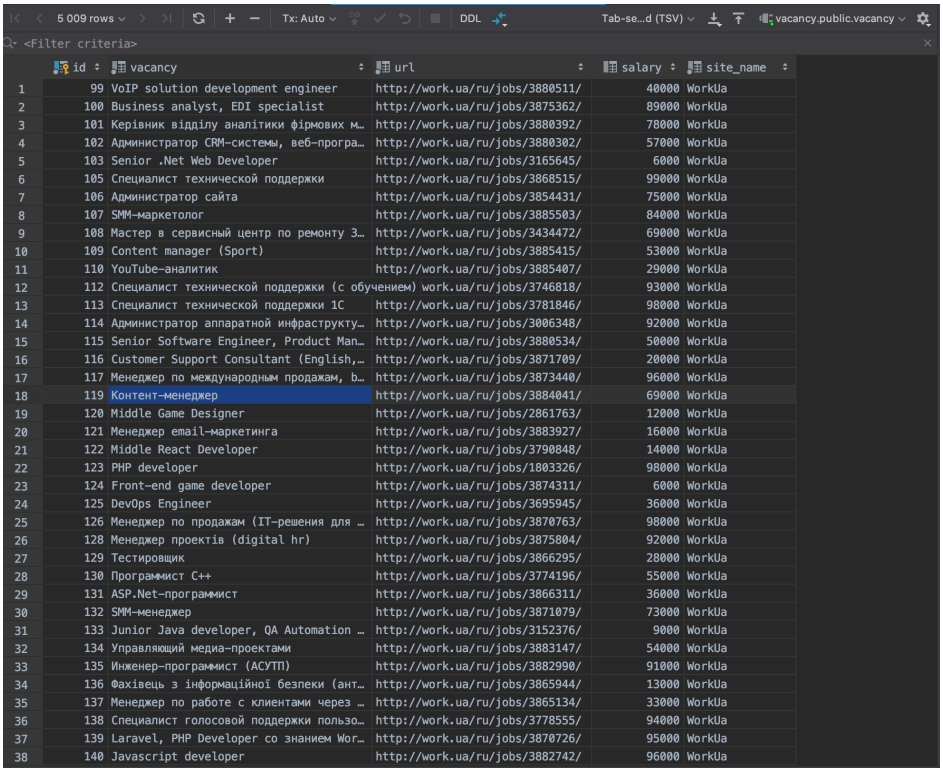
На рисунку 2.8, показано приклад консольної команди для запуску парсера.

Рисунок 2.8— Консольна команда для запуску парсера.

На рисунку 2.9, наведено результат роботи Парсера.

Рисунок 2.7— Результат роботи Парсера

У результаті роботи парсера, з 217 сторінок, було винайдено 15 вакансій . На 218тій сторінки нічого не було знайдено, тому роботу парсера було припинено. Перевіримо таблицю на наявність записів.



5 009 rows | Tx: Auto | DDL | Tab-se...d (TSV) | vacancy.public.vacancy

<Filter criteria>

	id	vacancy	url	salary	site_name
1	99	VoIP solution development engineer	http://work.ua/ru/jobs/3880511/	40000	WorkUa
2	100	Business analyst, EDI specialist	http://work.ua/ru/jobs/3875362/	89000	WorkUa
3	101	Керівник відділу аналітики фірмових м…	http://work.ua/ru/jobs/3880392/	78000	WorkUa
4	102	Администратор CRM-системы, веб-програ…	http://work.ua/ru/jobs/3880302/	57000	WorkUa
5	103	Senior .Net Web Developer	http://work.ua/ru/jobs/3165645/	6000	WorkUa
6	105	Специалист технической поддержки	http://work.ua/ru/jobs/3868515/	99000	WorkUa
7	106	Администратор сайта	http://work.ua/ru/jobs/3854431/	75000	WorkUa
8	107	SMM-маркетолог	http://work.ua/ru/jobs/3885503/	84000	WorkUa
9	108	Мастер в сервисный центр по ремонту 3…	http://work.ua/ru/jobs/3434472/	69000	WorkUa
10	109	Content manager (Sport)	http://work.ua/ru/jobs/3885415/	53000	WorkUa
11	110	YouTube-аналитик	http://work.ua/ru/jobs/3885407/	29000	WorkUa
12	112	Специалист технической поддержки (с обучением)	work.ua/ru/jobs/3746818/	93000	WorkUa
13	113	Специалист технической поддержки 1С	http://work.ua/ru/jobs/3781846/	98000	WorkUa
14	114	Администратор аппаратной инфраструкту…	http://work.ua/ru/jobs/3006348/	92000	WorkUa
15	115	Senior Software Engineer, Product Man…	http://work.ua/ru/jobs/3880534/	50000	WorkUa
16	116	Customer Support Consultant (English,…	http://work.ua/ru/jobs/3871709/	20000	WorkUa
17	117	Менеджер по международным продажам, b…	http://work.ua/ru/jobs/3873440/	96000	WorkUa
18	119	Контент-менеджер	http://work.ua/ru/jobs/3884041/	69000	WorkUa
19	120	Middle Game Designer	http://work.ua/ru/jobs/2861763/	12000	WorkUa
20	121	Менеджер email-маркетинга	http://work.ua/ru/jobs/3883927/	16000	WorkUa
21	122	Middle React Developer	http://work.ua/ru/jobs/3790848/	14000	WorkUa
22	123	PHP developer	http://work.ua/ru/jobs/1803326/	98000	WorkUa
23	124	Front-end game developer	http://work.ua/ru/jobs/3874311/	6000	WorkUa
24	125	DevOps Engineer	http://work.ua/ru/jobs/3695945/	36000	WorkUa
25	126	Менеджер по продажам (IT-решения для …	http://work.ua/ru/jobs/3870763/	98000	WorkUa
26	128	Менеджер проектів (digital hr)	http://work.ua/ru/jobs/3875804/	92000	WorkUa
27	129	Тестировщик	http://work.ua/ru/jobs/3866295/	28000	WorkUa
28	130	Программист C++	http://work.ua/ru/jobs/3774196/	55000	WorkUa
29	131	ASP.Net-программист	http://work.ua/ru/jobs/3866311/	36000	WorkUa
30	132	SMM-менеджер	http://work.ua/ru/jobs/3871079/	73000	WorkUa
31	133	Junior Java developer, QA Automation …	http://work.ua/ru/jobs/3152376/	9000	WorkUa
32	134	Управляющий медиа-проектами	http://work.ua/ru/jobs/3883147/	54000	WorkUa
33	135	Инженер-программист (АСУТП)	http://work.ua/ru/jobs/3882990/	91000	WorkUa
34	136	Фахівець з інформаційної безпеки (ант…	http://work.ua/ru/jobs/3865944/	13000	WorkUa
35	137	Менеджер по работе с клиентами через …	http://work.ua/ru/jobs/3865134/	33000	WorkUa
36	138	Специалист голосовой поддержки пользо…	http://work.ua/ru/jobs/3778555/	94000	WorkUa
37	139	Laravel, PHP Developer со знанием Wor…	http://work.ua/ru/jobs/3870726/	95000	WorkUa
38	140	Javascript developer	http://work.ua/ru/jobs/3882742/	96000	WorkUa

Рисунок 2.8 — Результат роботи парсера

На рисунку 2.8, показано роботу парсера, у результаті чого, було знайдено і відображено 5009 вакансій, після чого дані були записані у відповідні таблиці БД. Аналогічно опрацьована інформація з інших сайтів працевлаштування, доступних у відкритому доступі Інтернет мережі.

2.4 Створення графічної візуалізації отримання даних вакансій

Наступним кроком, є створення графічної візуалізації отримання даних про вакансії. Для цього, у роботі використовується об'єктно-орієнтована скриптова мова програмування JavaScript і бібліотека графічних компонентів CanvasJS.

Використання скриптової мови програмування JavaScript, обумовлено рядом переваг, наприклад JavaScript, дозволяє реалізувати ряд складних рішень Інтернет-додатків, які використовують графічне представлення інформації.

Логіка для графіків написана за допомогою шаблонізатора TWIG.

Основні особливості Twig, це швидкість компіляції в PHP код; безпечність, завдяки перевірці сумнівного коду; гучкість яка дозволяє визначати свої власні теги і фільтри.

Для взаємодії програмної частини і клієнтської сторони користувача інтерфейсу, використовується написане у бакалаврській роботі API на мові програмування PHP7. Для всіх розрахунків і пошуків, використовуються SQL запити.

Для реалізації системи аналізу рейтингів вакансій, було розроблено два варіанта пошуку:

1. Вибір назви професії з готового переліку (check-box);
2. Режим самостійного написання назви професії (input).

На рисунку 2.9, приведена діаграма компонентів, на якої представлена взаємодія програмної частини і клієнтської сторони користувача інтерфейсу. На стороні користувача, відбувається або вибір вакансії з переліку, або вводиться назва вакансії вручну. Після чого за допомогою компонент маршрутизації Symfony

Routing, передається до програмної частини Інтернет-додатка і за допомогою API, обробляються і виводяться в проаналізованому вигляді користувачу, тобто у вигляді діаграм, графіків і таблиць.

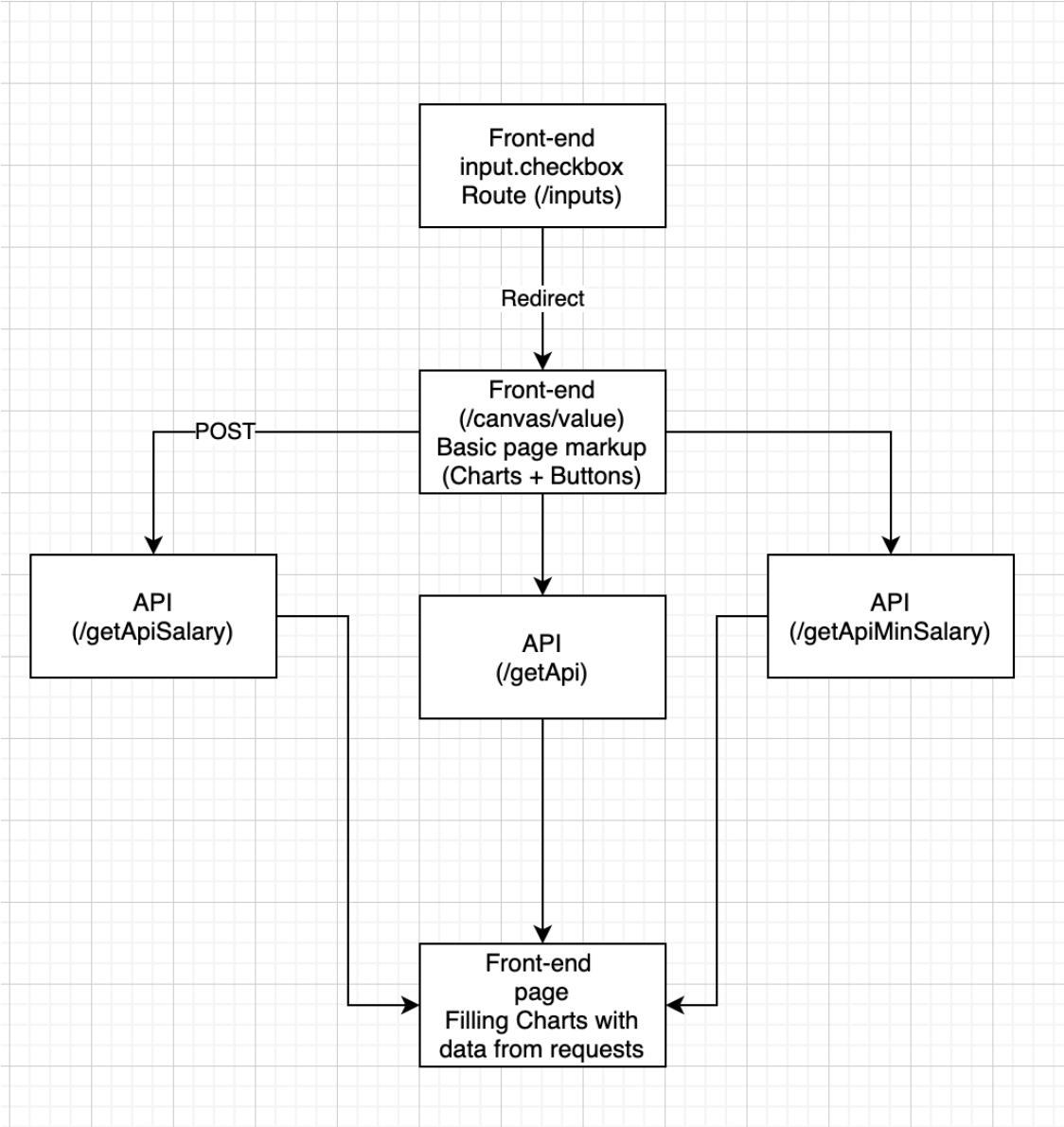


Рисунок 2.9 — Діаграма компонентів Front-end і Back-endоб’єктів

На рисунку 2.10, показано процес роботи клієнтської сторони системи аналізу рейтингів вакансій. За допомогою компонентів Input або checkboxes, днаі надсилаються на хост. Отриманні дані з параметрів запиту, обчислюються для того, щоб надсилати відповідь на наступні запити:

1. Мінімальна та максимальна заробітна плата;
2. Середня аробітна плата;

3. Кількість вакансій.

Інформація відправляється за допомогою метода POST-запиту в API. Після завершення запитів, інформація візуалізується за допомогою діаграм.

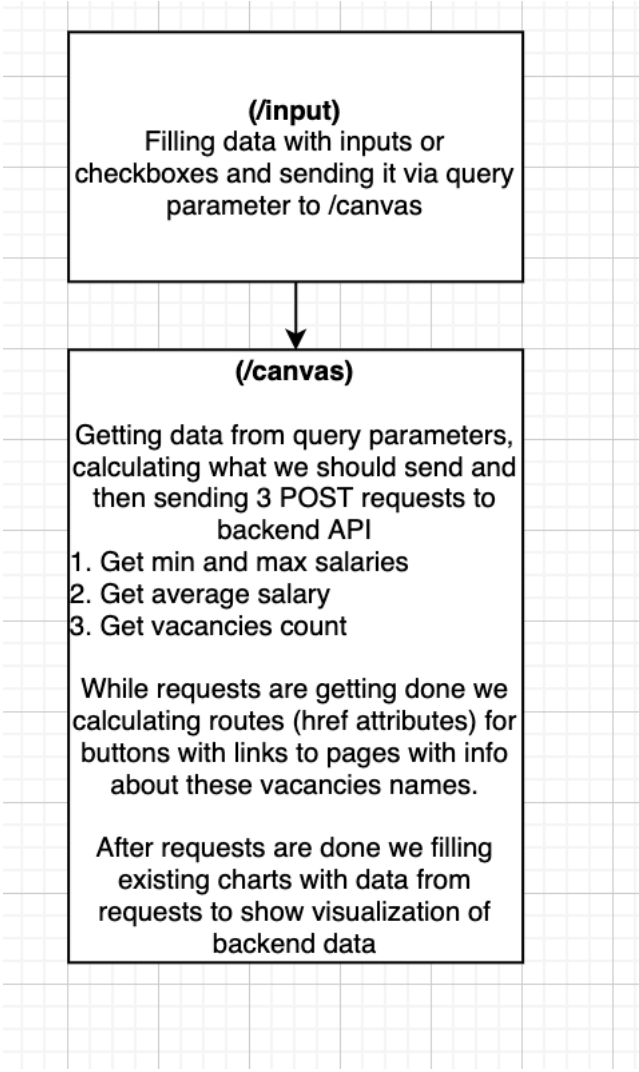


Рисунок 2.10 — Процес роботи клієнтської сторони системи аналізу рейтингів вакансій

На рисунку 2.11, показано процес роботи програмної частини. Для початку, створено один з патернів моделі MVC, а саме контролер (Controller). У даному класі . здійснюється обробка отриманих від користувача даних.

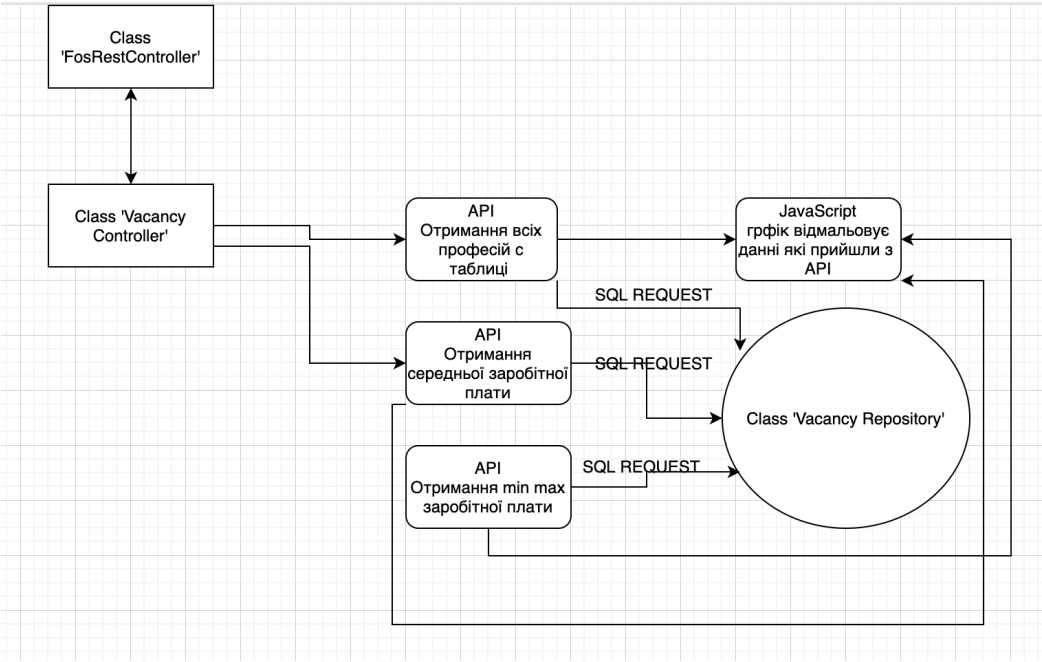


Рисунок 2.11 — Декомпозиція роботи API

Тобто API робить запит до репозиторію таблиці. У репозиторії прописан SQL запит який при необхідності виконується, наприклад для отримання всіх вакансій які ми шукаємо. Якщо шукаємо ‘android’ наш запит виглядає так:

```
SELECT count(v0_.id) AS sclr_0 FROM vacancy v0_ WHERE LOWER(v0_.vacancy) LIKE '%android%';
```

Рисунок 2.12 — Приклад запита вакансії

Аналогічно запиту з іншими даними буде змінюватися лише Value.

API працюють по майже однаковому принципу API отримує данні з FRONT-END наприклад з input чи check-box. На базі отриманих даних генерую SQL запит , у репеозиторії , повертає результат через API до FRONT-END і відображає результат.

Наприклад коли користувач шукає “android” йому повертаються данні з кількістю вакансій паралельно виконується SQL запит для підрахування середньої заробітної плати:

```
SELECT sum(v0_.salary) AS sclr_0 FROM vacancy v0_ WHERE LOWER(v0_.vacancy) LIKE '%android%';
```

Рисунок 2.13 — Приклад запита для підрахування середньої заробітної плати

```
SELECT count(v0_.id) AS sclr_0 FROM vacancy v0_ WHERE LOWER(v0_.vacancy) LIKE '%android%';
```

Рисунок 2.14 — Приклад запита для підрахування середньої заробітної плати

Аналогічно і для API який рахує MIN і MAX заробітну плату.

По запиту з FRONT-END .

```
SELECT min(v0_.salary) AS sclr_0 FROM vacancy v0_ WHERE LOWER(v0_.vacancy) LIKE '%android%';
```

```
SELECT max(v0_.salary) AS sclr_0 FROM vacancy v0_ WHERE LOWER(v0_.vacancy) LIKE '%android%';
```

Рисунок 2.15 — Приклад запита для підрахування MIN і MAX середньої заробітної плати

2.5 Висновки до розділу

Даний розділ описує процес розробки системи аналізу рейтингів вакансій. В бакалаврській роботі побудовано Інтернет-додаток, реалізуючий обробку заданих параметрів. На стороні клієнта, обирається список вакансій, за яким проводиться їх підрахунок та виводиться мінімальна, максимальна і середня заробітна плата .

3. АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ

3.1 Синтез етапів аналізу сайтів працевлаштування

В ході підготовки до проведення аналізу, було систематизовано інформацію з доступних сайтів пошуку роботи. Визначено мінімальну, максимальну і середню заробітну плату у галузі інформаційних технологій.

В бакалаврської роботі були проведені наступні етапи аналізу:

1. Визначення параметрів, за якими системи аналізу рейтингів вакансій проводить обрахування;
2. Побудова вибірки аналітики за допомогою Інтернет-додатка системи аналізу рейтингів вакансій;
3. Перевірка побудови вибіркової сукупності.

Побудована в ході бакалаврської роботи система аналізу рейтингів вакансій, показує попит на вакансії в галузі інформаційних технологій, стан мінімальної, максимальної і середньої заробітної плати, відповідних вакансій.

На рисунку 3.1, показано результат аналізу заробітної плати для програмістів працюючих з мовами програмування PHP, Java, Pyhton та аналітиків.

Розподіл заробітної плати майже однаковий, що співпадає з попитом на вишеперераховані компетентності на сучасному ринку інформаційних технологій. Так, найбільш популярними мовами програмування багато років були мови PHP та Java, розподіливши попит на замовлення між корпоративним та приватними секторами в галузі Інтернет технологій. В останні роки, зростання інтелектуальних систем, привело до зростання попиту на мову програмування Pyhton, та на попит до спеціальності аналітиків.

Найбільша заробітна плата, залишається за спеціалістами у мові програмування Java, але можливість написання програм зі штучним інтелектом на мові Pyhton, виводить спеціалістів у даній галузі на друг місце і показує, що мінімальна заробітна плата Pyhton-програмістів складає ... $, максимальна ... $, а

середня ... $. Програмісти на мові Java, отримують мінімальну заробітну плату ... $ максимальну .. $, середню ... $. Спеціалісти, програмуючі на мові Pyhton, отримують мінімальну ...$., мінімальну ... $, середню ... $.

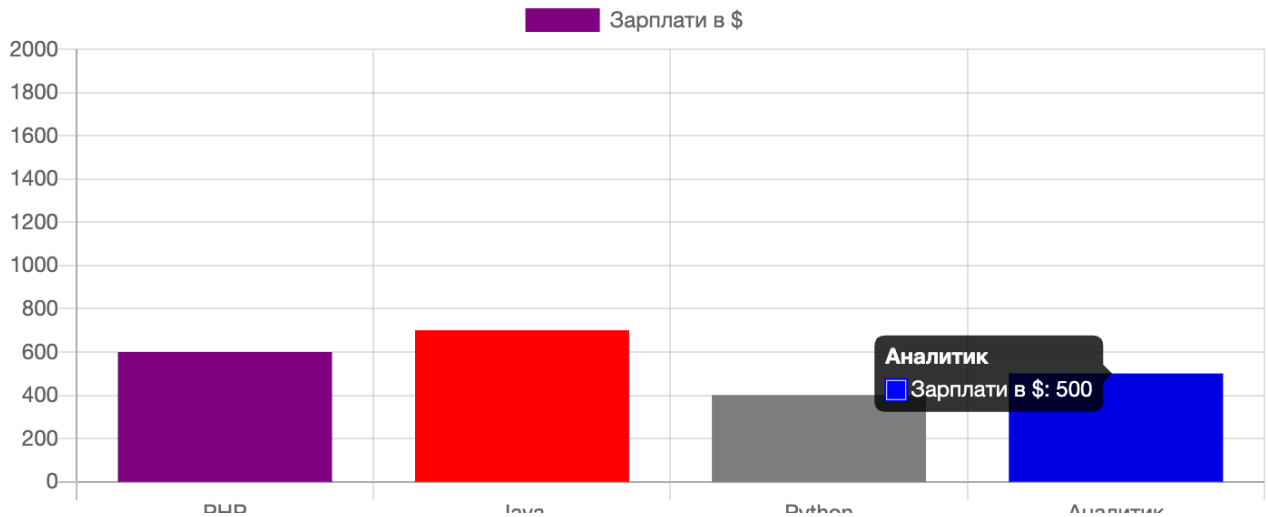


Рисунок 3.1 — Розподіл заробітної плати спеціалістів мов програмування PHP, Java, Pyhton та аналітиків.

На рисунку 3.2 показано співвідношення заробітної плати спеціалістів мов програмування PHP, Java, Pyhton та аналітиків. Діаграма відображає попит і доцільність розвитку в одному з цих напрямків, при вибіру професії, або підвищення компетенцій.

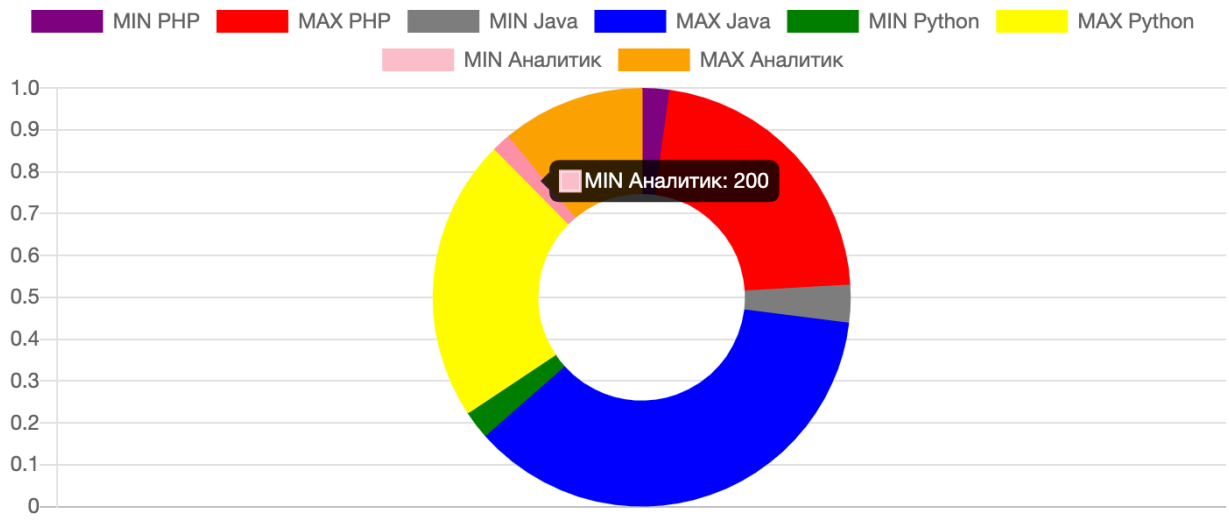


Рисунок 3.2 — Співвідношення заробітної плати спеціалістів мов програмування PHP, Java, Pyhton та аналітиків.

Найбільша кількість вакансій, розміщених на сайтах працевлаштування в галузі інформаційних технологій, передбачають роботу в компетентності мови програмування PHP. Це обумовлено великою кількістю Інтернет-додатків, написаних на цій мові у секторі малого та середнього бізнесу, а також ледінг сторінок для приватних задач. Невелика ціна, дає доступність до продукту написаному на мові PHP, для будь-кого, що збільшує попит. Простота написання коду, збільшує кількість спеціалістів, які навіть без офіційного робочого місця, можуть створювати невеликі додатки для приватних задач. Саме це віображено на рисунку 3.3.

Спеціалісти на мові Java, зазвичай працюють у корпоративному секторі, тому маючи найбільшу заробітну плату на ринку труда, попит на спеціалістів в рази менший за PHP-розробників.

Спеціалісти мови Pyhton, у нашій країні менш затребувані, тому що інтелектуальні системи, тільки починають заповнювати інформаційний ринок нашої країни. Але перехід на цифрову обробку бізнес-інформації, привів до попиту на вакансії з компетенцією “Аналітик”. Це обумовлено переходом до цифрового обчислення інформації не тільки великого, але й середнього і малого бізнесу.

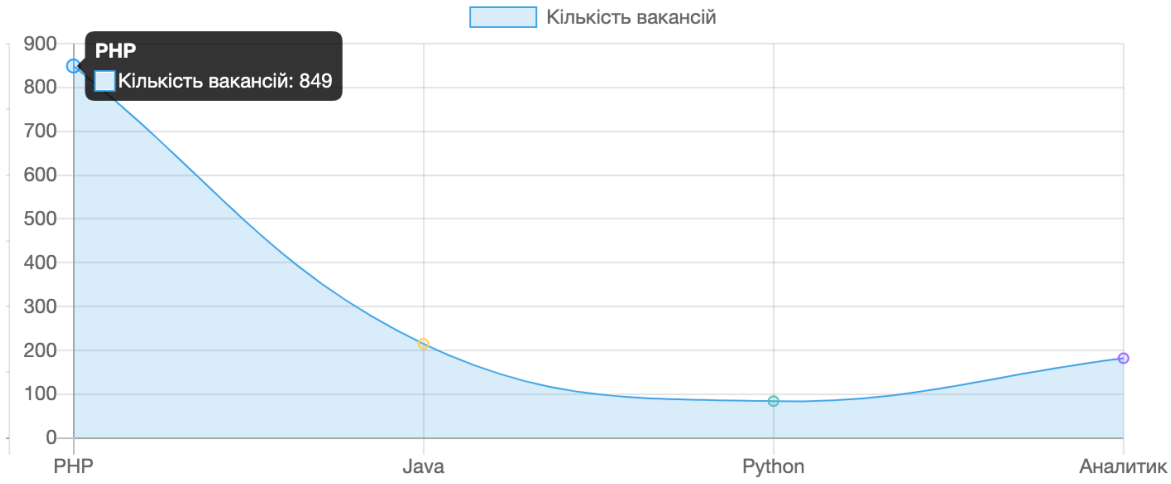


Рисунок 3.3 — Попит на спеціалістів у галузі програмування та аналітики.

Найбільша кількість вакансій, розміщених на сайтах працевлаштування, передбачають роботу саме в компетентності мов програмування PHP, Java, Pyhton, а перехід на цифрову обробку бізнес-інформації, приводить до великої кількості вакансій з компетенцією “Аналітик”.

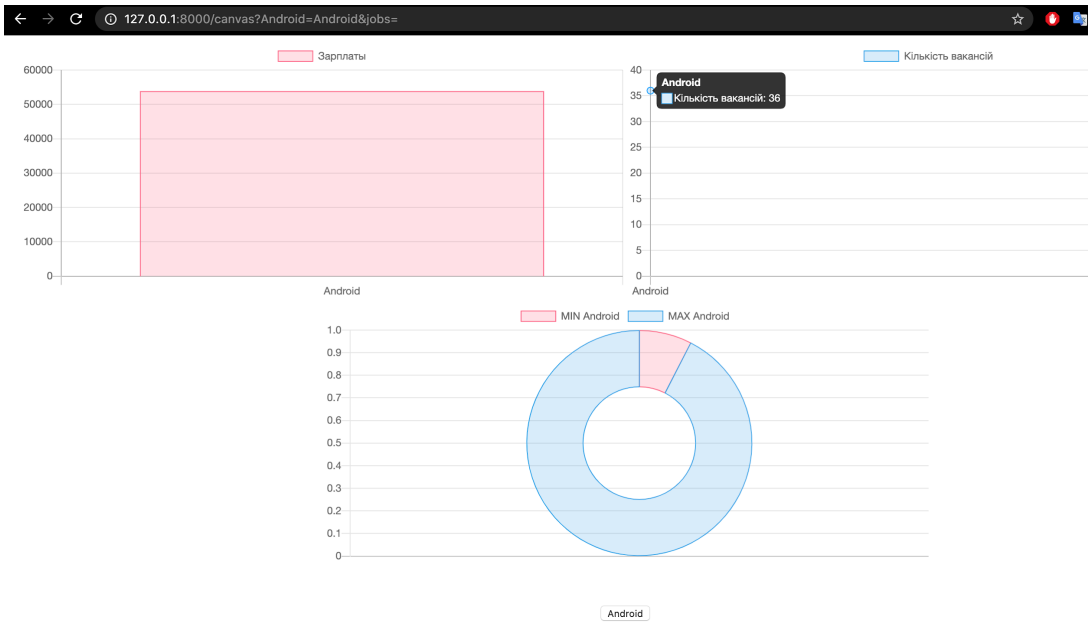


Рисунок 3.4 — Результат аналізу заробітної плати для фахівців операційної системи “Android”.

На рисунку 3.4, показано результат аналізу заробітної плати для фахівців операційної системи “Android”.

Для вибору компетентностей, у Інтернет-додатку системи аналізу рейтингів вакансій було представлено найбільш популярні мови програмування і назви вакансій, такі як PHP, Java, JavaScript, IOS, Pyhton, Android, Game Developer, C#, CEO. Але для більш детального розгляду попиту вакансій, користувач може самостійно ввести назву, по який буде проведено пошук на сайтах працевлаштування і виведено результат аналізу. Що показано на рисунку 3.5.

Рисунок 3.5 — Вибір компетентностей

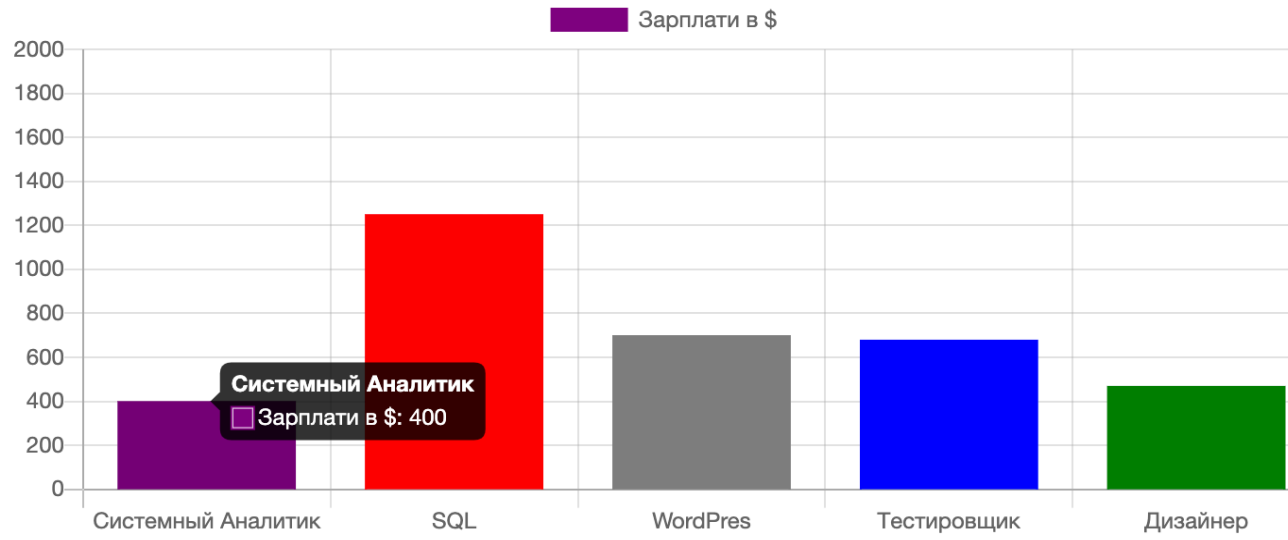


Рисунок 3.6 — Розподіл заробітної плати спеціалістів

На рисунку 3.6, показано результат аналізу заробітної плати спеціалістів у галузі створення програмних продуктів. Найбільший попит, спостерігається серед дизайнерів і спеціалістів шаблонізаторів. Перше обумовлено розповсюдженістю задач дизайну і результат аналізу, відповідає дійсності. А попит на спеціалістів WordPress, обумовлена простотою навчання даного продукту і попитом його використання у зарубіжного малого і середнього бізнесу. Багато спеціалістів у нашій країні працюють з аутсорсинговими компаніями, тому попит на задачі зарубіжних замовників також відповідає дійсності і результатам аналізу даної бакалаврської роботи. Далі рейтинг підтримують спеціалісти запитів до баз даних. Це обумовлено тим, що будь-яка інформація повина зберігатись і спеціалісти, які розуміються на мові запитів до СКБД, також мають великій попит на ринку труда. Останньою нишою, яка є в даному аналізі, це тестировщики. Попит на цю вакансію, обумовлено тим, що на будь-якому середньому або великому проекті, у сучасній компаній є дана вакансія і один або команда тестировщиків. Тому і заробітна плата на перелічені компетентності, майже однакова Але розподіл заробітної плати не відповідає кількості вакансій, як видно з графіка, представленого на рисунку 3.7. Це обумовлено кількістю компаній і відповідністю задач. Так, наприклад спеціалісти мові запитів до СКБД, є не тільки у середніх і крупних компаніях, а майже всюди, де впроваджуються інформаційні технології. Тому попит на

спеціаліста, знаючого запити sql, можуть зустрічатись навіть у вакансіях розробників інтерфейсів користувача. Ця компетенція може зустрічатись в описі вакансії, як додаткова.

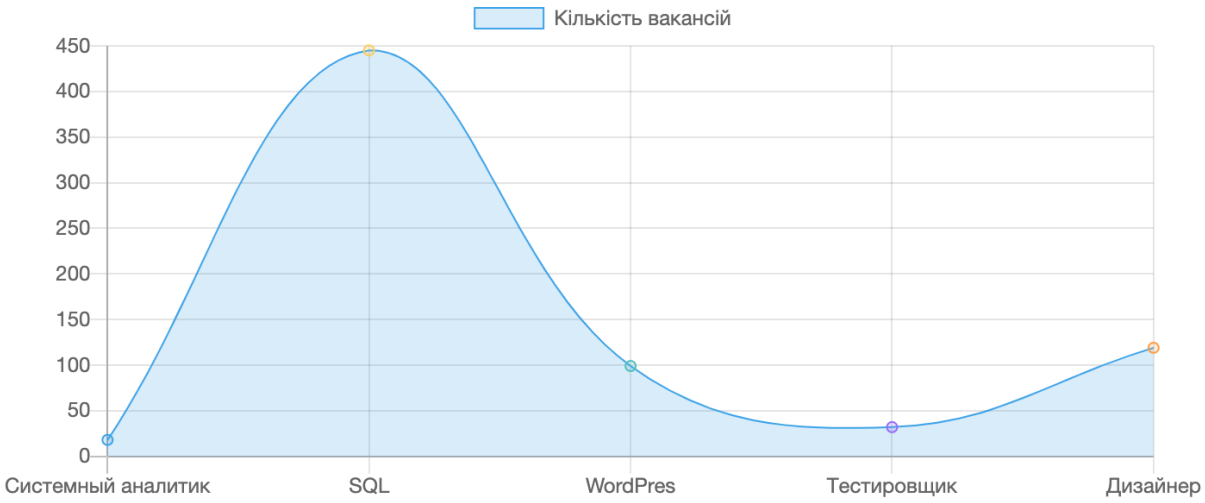


Рисунок 3.7 — Попит на спеціалістів у галузі створення програмних продуктів.

На рисунку 3.8 представлено розподіл попиту серед спеціалістів в галузі інформаційних технологій, якій показує залучення CEO спеціалістів і рівень заробітної плати. Ріст заробітної плати CEO спеціалістів обумовлено додатковим заохочення персоналу, якій долучається до процесу продажу інформаційних продуктів і часто залежить від процента продажу.

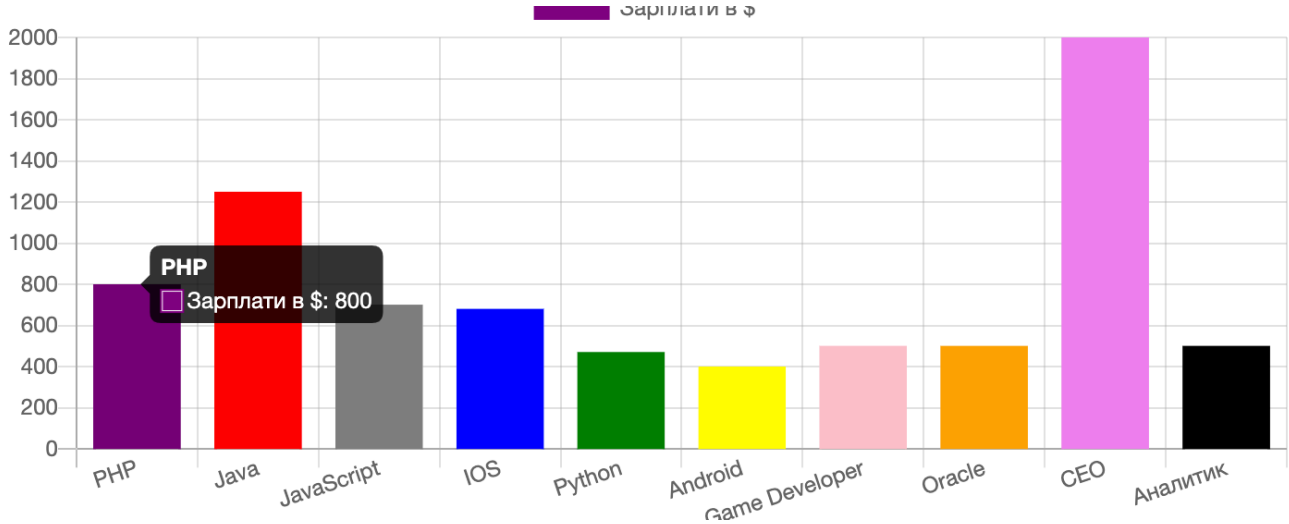


Рисунок 3.8 — Розподіл попиту серед спеціалістів в галузі інформаційних технологій

На рисунку 3.9 представлено кількість вакансій спеціалістів в галузі інформаційних технологій . Аналіз показує, рівень заробітної плати різниться з розподілом попиту вакансій. Так самою затребуваною вакансією, є спеціалісти, які знаються на мові PHP, у відмінності від попереднього графіка заробітних плат (рис. 3.7).

Рисунок 3.9 — Кількість вакансій спеціалістів в галузі інформаційних технологій

На рисунку 3.10 показано розподіл мінімальних, максимальних, та середніх заробітних плат спеціалістів в галузі інформаційних технологій, таких як JavaScript, IOS, Game Developer, Android, Oracle, PHP, Java, Game Developer, CEO. Завдяки даному аналізу видно попит на відповідні компетенції, а також як змінюється заробітна плата від мінімального до максимального покажчика.

Аналіз динаміки мінімальних, максимальних, та середніх заробітних плат, показує попит на наступні мови програмування:

1. CEO спеціалісти;
2. Java спеціалісти;
3. Розробники ігор (Game Developer);
4. Android спеціалісти;
5. PHP спеціалісти;
6. JavaScript спеціалісти;
7. Pyhton спеціалісти;
8. Oracle спеціалісти/

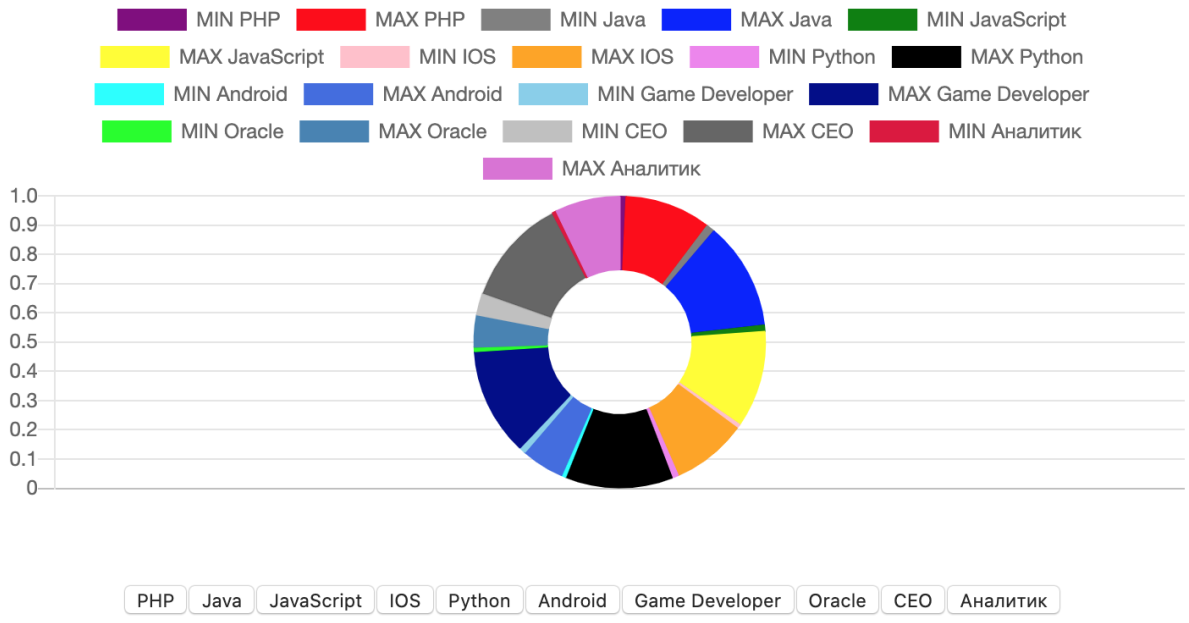


Рисунок 3.10 — Розподіл мінімальних, максимальних, та середніх заробітних плат спеціалістів в галузі інформаційних технологій

Мінімальна і максимальна заробітна плата залежить від рівня спеціаліста, але на показник рівня середньої заробітної плати має вплив кількість затребуваних на ринку місць. На рисунку 3.11 приведено зовнішній вигляд Інтернет-додатка системи аналізу рейтингів вакансій, на якому у зручному вигляді демонструються тенденції попиту на сайтах працевлаштування. Користувач, має змогу порівняти як кількість вакансій, так і заробітну плату, по будь-якій обраній компетенції, або введеній власноруч.

Аналіз бакалаврської роботи, показує, що серез затребуваних професій, в галузі інформаційних технологій, великим попитом є, як менеджери просуваючі продукти, так і аналітики, обробляючі дані, а не тільки програмісти і адміністратори. Пропозиції вакансій можливо аналізувати на всіх доступних у мережі Інтернет сайтах працевлаштування. Заробітна плата оцінюється тільти з тих джерел, де вона вказана.

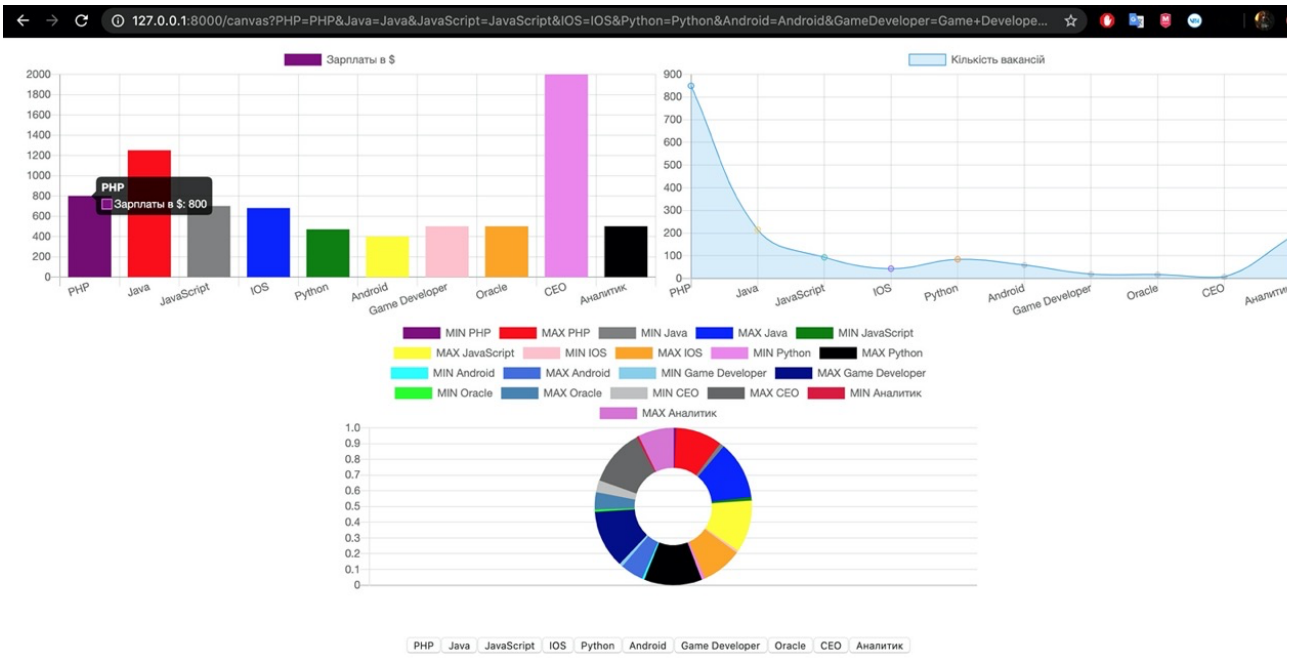


Рисунок 3.11 — Зовнішній вигляд Інтернет додатка системи аналізу рейтингів вакансій

Так, на рисунку 3.12, можна наглядно порівняти попит на спеціалістів у сфері програмування на мовах:

1. Angular;
2. C++;
3. Ruby;
4. .Net;
5. Swift;
6. DataBase;
7. 1C.

Як видно з рисунка 3.12, кількість вакансій інша, розподіл заробітної плати. Так пропозиція зі сторони роботодавців на позицію програмістів мови .Net, перевищує інші пропозіції, в той час як заробітна плата найвища у спеціалістів Angular і C++.

У побудованій в бакалаврській роботі системі , можливо наглядно бачити яка професія найбільше перевищує попит, обираючи при цьому професії і сайти для обробки.

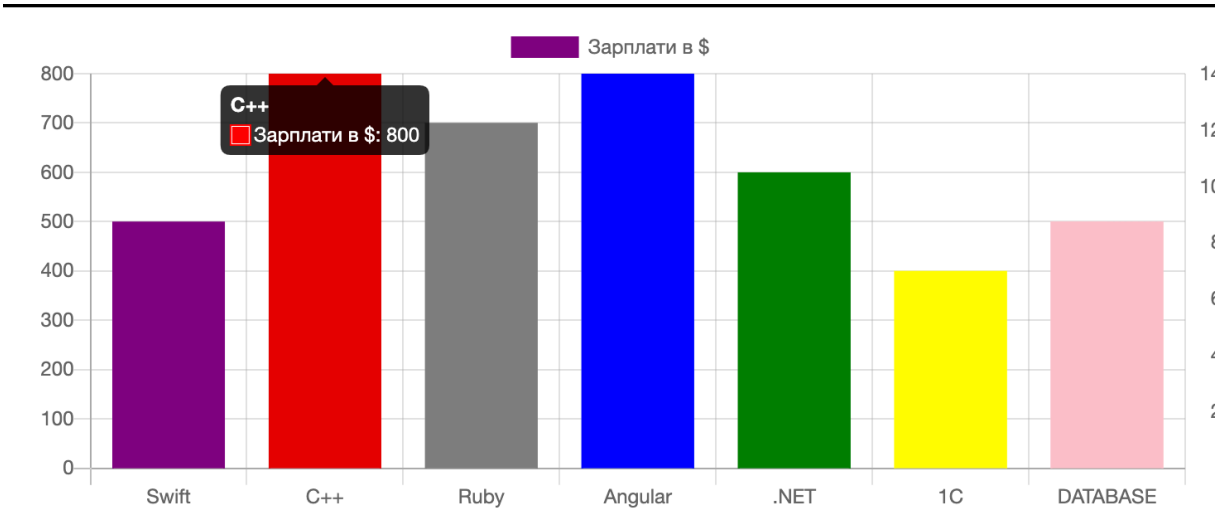


Рисунок 3.12 — Рівень заробітної плати програмістів

На рисунку 3.13, показана кількість затребуваних роботодавцями спеціалістів у галузі програмування. Як видно з графіка, розподілення наступне:

1. .Net;

2. 1C;
3. C++;
4. Angular;
5. DataBase;
6. Ruby;
7. Swift;

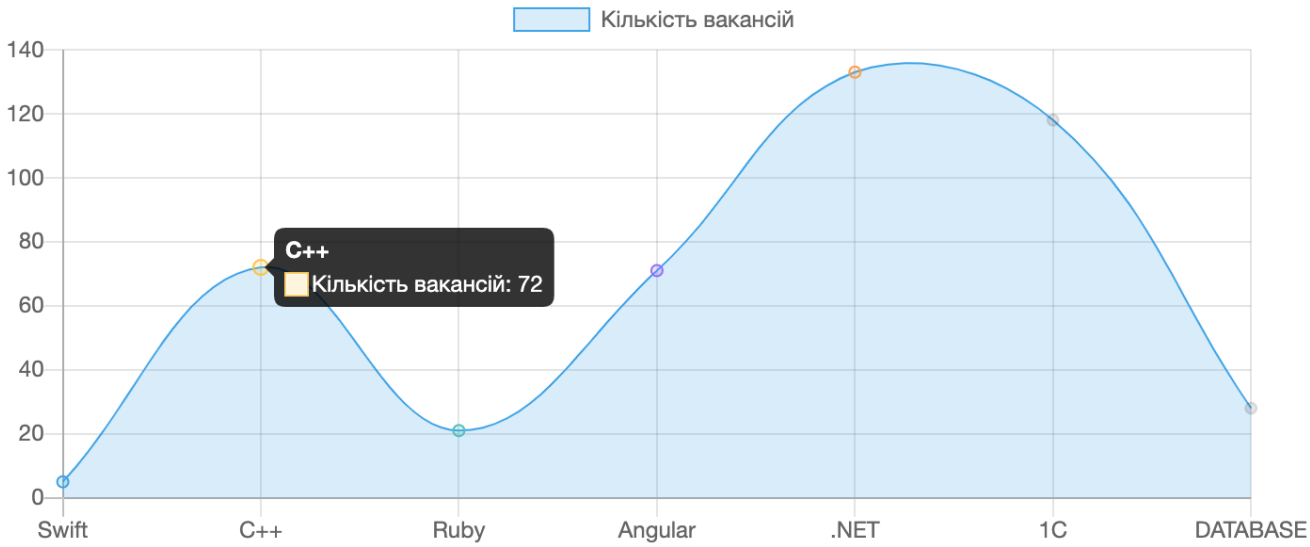


Рисунок 3.13 — Кількість затребуваних роботодавцями спеціалістів у галузі програмування

Попит на спеціалістів наведених на графіку 3.13, обумовлено розповсюдження операційної системи Windows, тому спеціалісти працюючі з Net, потрібні найчастіше. Подібним чином, потрібні спеціалісти 1С, так як бухгалтерській облік, є у будь якому бізнесі.

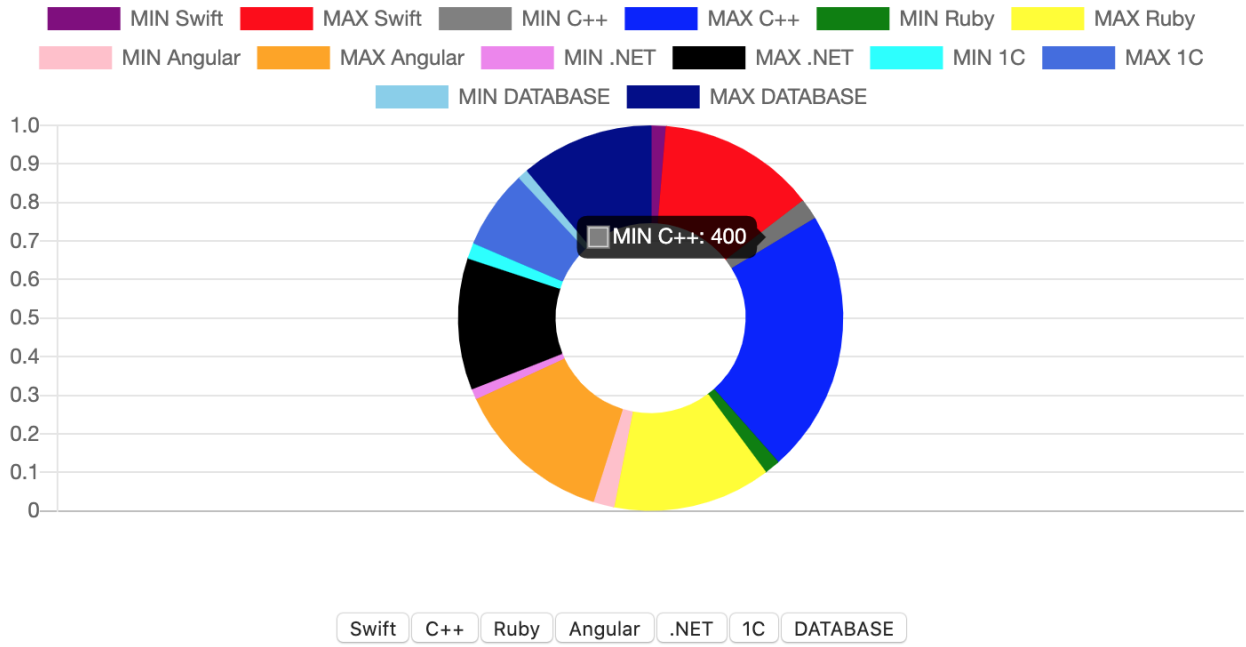


Рисунок 3.14 — Розподіл мінімальних, максимальних і середніх заробітних плат спеціалістів у галузі програмування

На рисунку 3.14 показано Розподіл мінімальних, максимальних і середніх заробітних плат спеціалістів у галузі програмування

Таблиця 3.1 Розподіл мінімальних, максимальних і середніх заробітних плат спеціалістів у галузі програмування

Мови програмування	Мін Зарплата	Макс Зарплата	Серендя Зарплата
Java	500$	5000$	1600$
PHP	300$	4000$	1500$
C#	300$	5000$	1000$
Python	400$	4000$	1200$
Ruby on Rails	420$	2,800$	1300$
1C	300$	2500$	1400$
C++	500$	3900$	1800$
.NET	300$	2500$	1200$
SWIFT	400$	3000$	1500$
R	400$	3500$	1400$
Angular	500$	5000$	1600$
SQL	200$	2500$	900$

Назва вакансій	Мін Зарплата	Макс Зарплата	Середня Зарплата
Java Developer Senior	2000$	6000$	3500$
PHP Developer Symfony Mid	1500$	3000$	2000$
Ситсемний аналітик	300$	3000$	900$
Database Developer	400$	2000$	800$
Angular Front-End Developer	300$	3000$	1200$
Swift IOS Developer	400$	3000$	1500$
Java Back-End Developer Middle	1200$	2500$	1500$
1C Developer Bitrix	200$	800$	400$
Android Developer	300$	2500$	900$
QA engineer	300$	1500$	900$

Таблиця 3.2 Розподіл мінімальних, максимальних і середніх заробітних плат спеціалістів у галузі інформаційних технологій

Мови програмування	Мін Зарплата	Макс Зарплата	Серендя Зарплата
Java	500$	5000$	1600$
PHP	300$	4000$	1500$
C#	300$	5000$	1000$
Python	400$	4000$	1200$
Ruby on Rails	420$	2,800$	1300$
1C	300$	2500$	1400$
C++	500$	3900$	1800$
.NET	300$	2500$	1200$
SWIFT	400$	3000$	1500$
R	400$	3500$	1400$
Angular	500$	5000$	1600$
SQL	200$	2500$	900$

Назва вакансій	Мін Зарплата	Макс Зарплата	Середня Зарплата
Java Developer Senior	2000$	6000$	3500$
PHP Developer Symfony Mid	1500$	3000$	2000$
Ситсемний аналітик	300$	3000$	900$
Database Developer	400$	2000$	800$
Angular Front-End Developer	300$	3000$	1200$
Swift IOS Developer	400$	3000$	1500$
Java Back-End Developer Middle	1200$	2500$	1500$
1C Developer Bitrix	200$	800$	400$
Android Developer	300$	2500$	900$
QA engineer	300$	1500$	900$

У таблицях 3.1 і 3.2, показано розподіл мінімальних, максимальних і середніх заробітних плат спеціалістів у галузі інформаційних технологій та програмування. Таблиці повторюють графік, подаючи інформацію у більш зручному форматі для звітів

Після обробки інформації з різних джерел і вивода результатів у графічному вигляді, користувач, має можливість обрати будь яку вакансію, або компетенцію, і натиснувши на неї, перейти по посиланню зі списком сайтів, звідки бралась інформація для аналізу. Перевірка працездатності посилання показана на рисунку 3.15

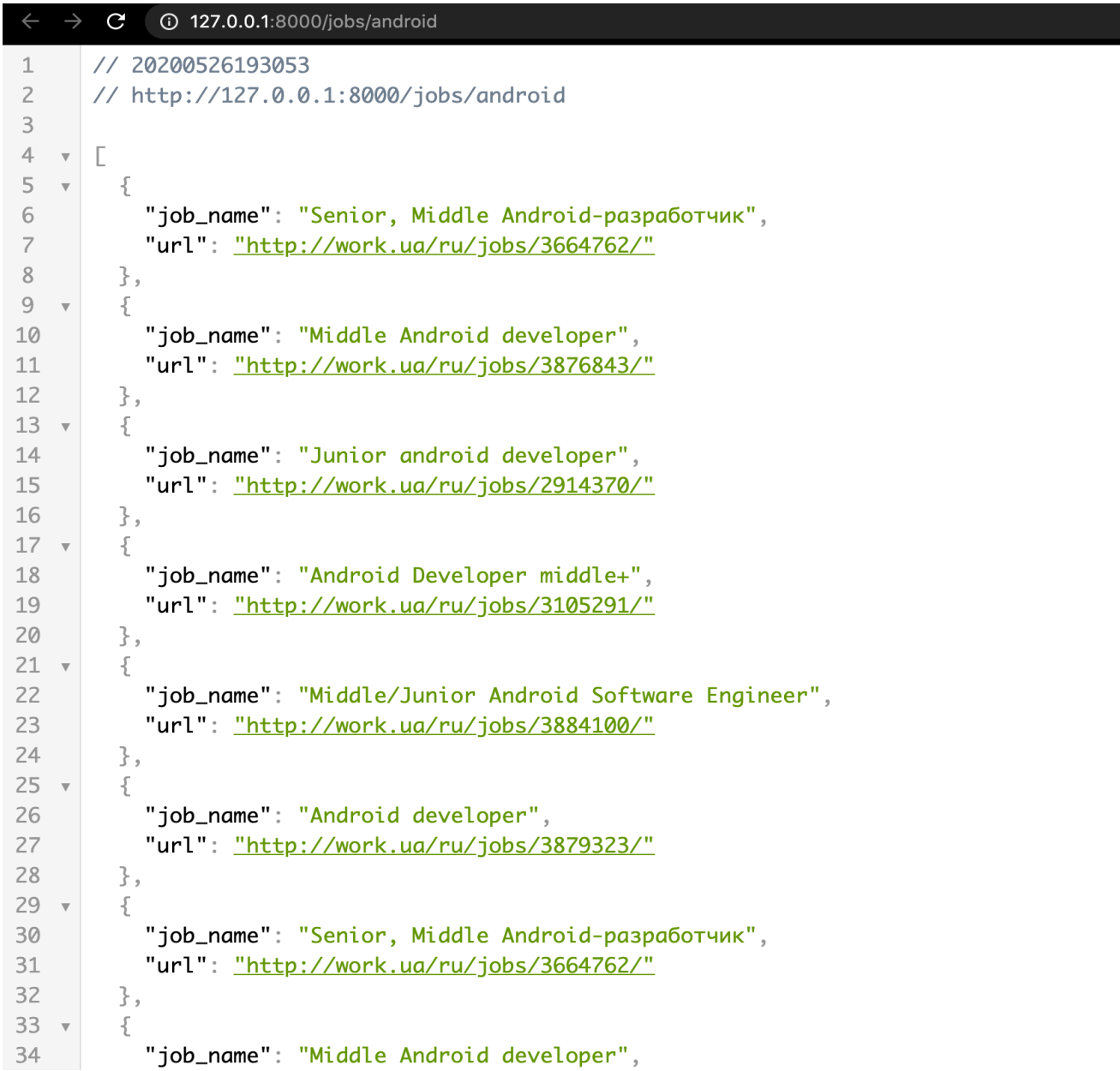


Рисунок 3.15 — Посилання на джерела інформації для проведеного в роботі аналізу

Після переходу на список ресурсів, з яких було взято інформацію про вакансії, користувач має можливість перейти безпосередньо на сайт працевлаштування, що показано на рисунку 3.16

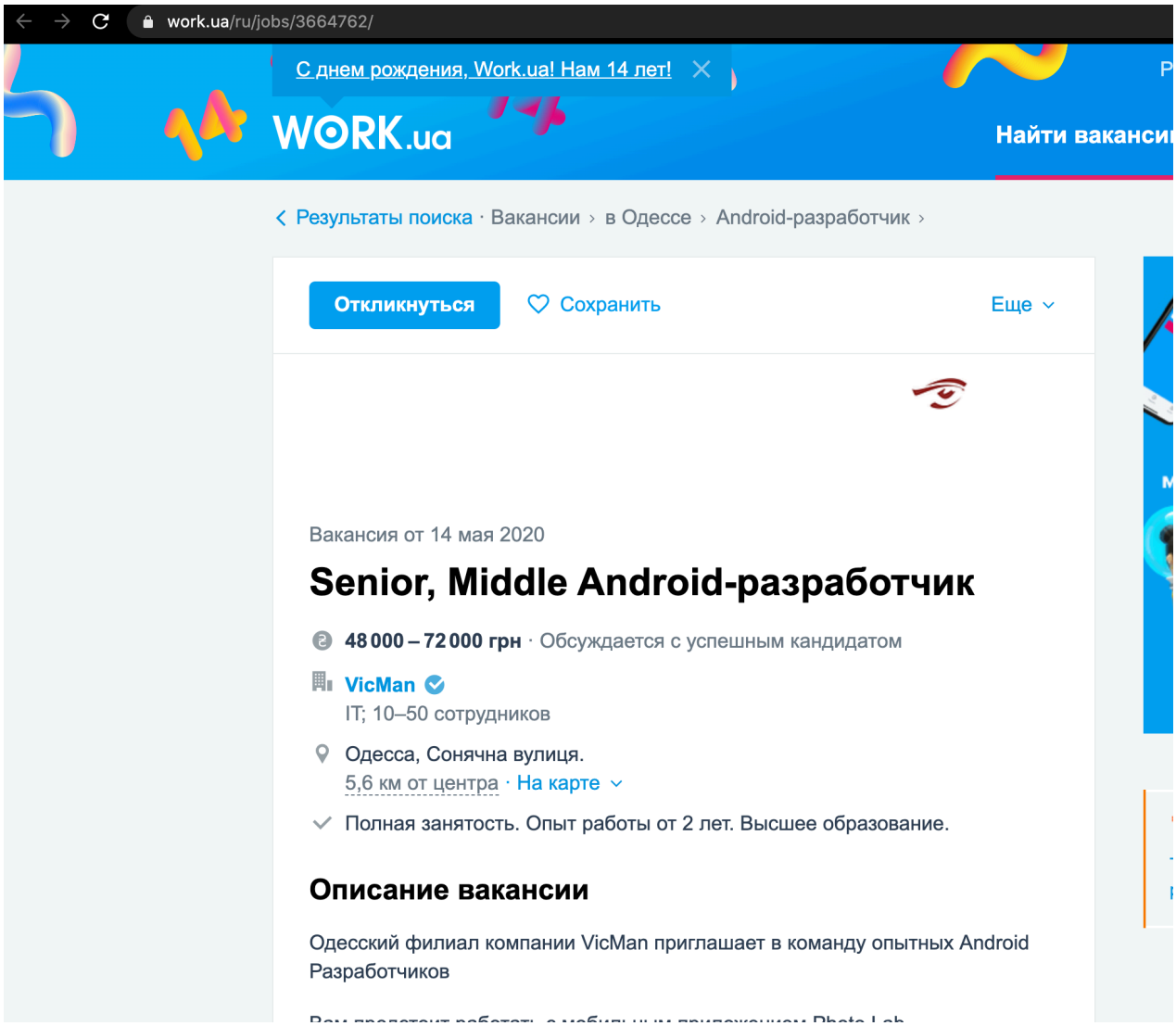


Рисунок 3.16 — Посилання на джерела інформації для проведеного в роботі аналізу

4. ТЕСТУВАННЯ

В таблиці 4.1 наведено розроблені тест-кейси для тестування дієздатності посилення. При тестуванні системи усі наведені тести було пройдено із результатом passed.

Таблиця 4.1 - Тестування дієздатності посилання

№ Тест-кейсу	ОК/NOK	Дії	Очікуваний результат
1.	ОК	Перейти за вказаним посиланням на сайт	• Виконується завантаження сайту • Після завантаження на екрані правильно відображаються елементи інтерфейсу сайту.
2.	NOK	Під час переходу по посиланню перейти по другому посиланню в іншій вкладці	• Завантаження сайту триватиме без прямого знаходження на вкладці із посиланням • Після відкриття завантаженого сайту із іншої вкладки, всі елементи інтерфейсу відображатимуться вірно
3.	ОК	Натиснути на клавішу “Reload” браузера	• Відбувається перезавантаження поточної відкритої сторінки

В таблиці 1.2 наведено розроблені тест-кейси для тестування взаємодії з інтерфейсом сайту. При тестуванні системи усі наведені тести було пройдено із результатом passed.

Таблиця 4.2 - Тестування взаємодії з інтерфейсом

№ Тест-кейсу	ОК/NOK	Дії	Очікуваний результат
1.	ОК	Натискання на елементи типу Checkbox	• В момент натискання елементи типу checkbox змінюють свій вигляд. • Повторне натискання повертає елемент в початковий стан
2.	ОК	Натискання на елемент типу Inputfield	• В момент натискання елемент змінює свій колір та з'являється каретка • В полі відображаються введені букви та цифри • Введені дані можна редагувати • Після натискання курсору поза елементом, введені дані залишаються в полі • Після повторного натискання курсору на поле з даними, дані залишаються в полі.

3.	ОК	Натискання на елементи типу Button	• При наведенні на елемент відбувається зміна кольору • При натисканні на елемент відбувається закріплена за цією дією функція.
4.	ОК	Вибір фільтру пошуку та одночасне введення даних в Inputfield	• При одночасному натисканні на фільтри типу checkbox та введені даних в поле inputfield сайт продовжує свою роботу по заданому сценарію. • При натисканні курсору на порожні простір вибраний фільтр пошуку та введені данні не скидаються. • При натисканні на кнопку “Количество вакансий” відбувається перехід на іншу сторінку
5.	ОК	Перевірка локалізації	• Усі локалі написано правильно на російській мові

В таблиці 4.3 наведено розроблені тест-кейси для тестування працездатності функціоналу кнопки “Количество вакансий” через InputField. При тестуванні системи усі наведені тести було пройдено із результатом passed.

Таблиця 1.3 - Тестування функціоналу кнопки “Количество вакансій” через InputField

№ Тест-кейсу	ОК/NOK	Дії	Очікуваний результат
1.	ОК	Пошук введеної вакансії “Програміст” Через Inputfield	• Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку • На іншій сторінці правильно відображаються графіки з середньою зарплатою по заданому запиту, кількістю вакансій, та порівняння мінімальної та максимальної зарплат
2.	ОК	Пошук декількох вакансій через Inputfield	• Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку • На іншій сторінці правильно відображаються графіки з середньою зарплатою, діаграма кількості вакансій, та порівняння мінімальної та максимальної зарплат декількох вакансій

3.	OK	Натиснути на клавішу “Back” браузера	• Відбувається перехід на попередню сторінку
4.	OK	Натиснути на клавішу “Reload” браузера	• Відбувається перезавантаження поточної відкритої сторінки
5.	NOK	Введення неправильних даних для пошуку (наприклад звичайний набір незв'язаних символів)	• Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку • На іншій сторінці не відображається результат по введеному запиту
6.	NOK	Введення декількох неправильних даних для пошуку (наприклад звичайний набір незв'язаних символів)	• Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку • На іншій сторінці не відображається результат по введеному запиту
7.	NOK	Пошук вакансій з правильними	• Натискання на кнопку “Количетсво

		та неправильними даними	вакансий” виконає перехід на іншу сторінку • На іншій сторінці відображається результат правильного запиту поряд з яким не відображається результат неправильного запиту

В таблиці 4.4 наведено розроблені тест-кейси для тестування працездатності функціоналу кнопки “Количество вакансий” через checkbox. При тестуванні системи усі наведені тести було пройдено із результатом passed.

Таблиця 1.4 - Тестування функціоналу кнопки “Количество вакансій” через checkbox

№ Тест-кейсу	ОК/NOK	Дії	Очікуваний результат
1.	ОК	Пошук вакансії через обраний checkbox (наприклад “PHP”)	Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку На іншій сторінці правильно відображаються графіки з середньою зарплатою по заданому запиту, кількістю вакансій, та порівняння мінімальної та

			максимальної зарплат
2.	ОК	Пошук декількох вакансій через обраний checkbox (наприклад “PHP” та “Java”)	Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку На іншій сторінці правильно відображаються графіки з середньою зарплатою, діаграма кількості вакансій, та порівняння мінімальної та максимальної зарплат декількох вакансій
3.	ОК	Пошук вакансій через усі обрані checkbox	Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку На іншій сторінці правильно відображаються графіки з середньою зарплатою, діаграма кількості вакансій, та порівняння мінімальної та максимальної зарплат усіх обраних вакансій Усі елементи відображаються на екранні правильно з дотриманням розмірів та без утискання елементів

4.	ОК	Натиснути на клавішу “Back” браузера	Відбувається перехід на попередню сторінку
5.	ОК	Натиснути на клавішу “Reload” браузера	Відбувається перезавантаження поточної відкритої сторінки

В таблиці 4.5 наведено розроблені тест-кейси для тестування працездатності функціоналу кнопки “Количество вакансий” через checkbox та поле Inputfield одночасно. При тестуванні системи усі наведені тести було пройдено із результатом passed.

Таблиця 4.5 - Тестування функціоналу кнопки “Количество вакансій” через checkbox та поле Inputfield одночасно

№ Тест-кейсу	ОК/NOK	Дії	Очікуваний результат
1.	NOK	Пошук вакансії через обраний checkbox та через введені данні в поле Inputfield (наприклад “Програміст” та обраний Checkbox “Java”)	Натискання на кнопку “Количетсво вакансий” виконає перехід на іншу сторінку На іншій сторінці правильно відображаються графіки з середньою зарплатою, кількістю вакансій, та порівняння мінімальної та максимальної зарплат лише по запиту заданому через inputfield так як

			у нього більший пріоритет.

В таблиці 1.6 наведено розроблені тест-кейси для тестування працездатності функціоналу отримання результату пошуку натисканням на кнопку с назвою запиту. При тестуванні системи усі наведені тести було пройдено із результатом passed.

Таблиця 4.6 - Тестування працездатності функціоналу отримання результату пошуку натисканням на кнопку с назвою запиту.

№ Тест-кейсу	ОК/NOK	Дії	Очікуваний результат
1.	ОК	Зробити пошук бажаної вакансії через Inputfield. На сторінці з результатом пошуку натиснути на кнопку из назвою запиту.	Натискання на кнопку відкривае Json файл з назвою вакансії та посиланням на сайт з вакансією Перехід за посиланням відкриває правильну працездатну сторінку із заданою вакансією
2.	ОК	Зробити пошук бажаної вакансії через checkbox. На сторінці з результатом пошуку натиснути на кнопку із назвою запиту	Натискання на кнопку відкривае Json файл з назвою вакансії та посиланням на сайт з вакансією Перехід за посиланням відкриває правильну працездатну сторінку із заданою вакансією

3.	NOK	Зробити пошук неіснуючої вакансії або запит що містить в собі лише не пов’язані між собою символи через Inputfield. На сторінці з результатом пошуку натиснути на кнопку із назвою запиту.	Натискання на кнопку відкриває пустий Json файл.
4.	ОК	Натиснути на клавішу “Back” браузера	Відбувається перехід на попередню сторінку
5.	ОК	Натиснути на клавішу “Reload” браузера	Відбувається перезавантаження поточної відкритої сторінки

ВИСНОВКИ

В бакалаврській роботі була розроблена системи аналізу рейтингів вакансій, розташованих на сайтах пошуку роботи. Інтернет-додаток, наглядно демонструє аналіз вакансій у сфері інформаційних технологій, показує зріст чи спадання попиту на вакансії, завдяки обробці інформації з різних сайтів працевлаштування. В бакалаврській роботі були вирішені наступні питання:

1. Розробка скрипта для збору інформації про вакансії з різних джерел пошуку роботи.
2. Покращення швидкості обробки зібраної інформації з джерел пошуку роботи..
3. Створення статистики за запитом користувача про популярність вакансій, заробітну плату, потрібний попередній досвід.

Основними перевагами додатку для аналізу вакансій, є максимальна зручність для користувача в обробці інформації про вакансії.

ДОДАТОК А

Back-End:

```
class WorkUaCommand extends Command
{
    protected static $defaultName = 'WorkUa';

    /**
     * @var EntityManager
     */
    private $entityManager;

    public function __construct(string $name = null, EntityManagerInterface
$entityManager)
    {
        parent::__construct($name);
```

```
        $this->entityManager = $entityManager;
    }

    protected function configure()
    {
        $this
            ->setDescription('Parsin site for a Vacancy')
            ->addArgument('arg1', InputArgument::OPTIONAL, 'Argument description')
            ->addOption('option1', null, InputOption::VALUE_NONE, 'Option
description');
    }

    protected function execute(InputInterface $input, OutputInterface $output):
int
    {


        for ($i = 1; true; $i++) {
            $client = HttpClient::create();
            $response = $client->request('GET', 'https://www.work.ua/ru/jobs-
it/?page=' . $i);
            $content = $response->getContent();


            $crawler = new Crawler();
            $crawler->addContent($content);


            /** @var \DOMElement[] $jobNodes */
            $jobNodes = $crawler->filter('#pjax-job-list > div.card-hover > h2 >
a');
//            var_dump($jobNodes);die;
//
            var_dump('https://www.work.ua/ru/jobs-it/?page=' . $i);
            var_dump($i . ' ' . count($jobNodes) . '\n');
            if (count($jobNodes) === 0) {
                break;
            }
            foreach ($jobNodes as $jobNode) {

                $job = new Vacancy();
                $job->setVacancy($jobNode->textContent);
                $job->setUrl('http://work.ua'.$jobNode->getAttribute('href'));

                $job->setSiteName('WorkUa');

                $this->entityManager->persist($job);
            }


            if (($i % 6) === 0) {
                $this->entityManager->flush();
            }
        }
        $this->entityManager->flush();

        return 0;

    }
}
class Vrtualvocations extends Command
{
```

```
    protected static $defaultName = 'vr';

    /**
     * @var EntityManager
     */
    private $entityManager;

    public function __construct(string $name = null, EntityManagerInterface
$entityManager)
    {
        parent::__construct($name);
        $this->entityManager = $entityManager;
    }

    protected function configure()
    {
        $this
            ->setDescription('Add a short description for your command')
            ->addArgument('arg1', InputArgument::OPTIONAL, 'Argument description')
            ->addOption('option1', null, InputOption::VALUE_NONE, 'Option
description');
    }

    protected function execute(InputInterface $input, OutputInterface $output):
int
    {
        for ($i = 1; true; $i++) {
            $client = HttpClient::create();
            $response = $client->request('GET',
'https://www.virtualvocations.com/jobs');
            $content = $response->getContent();


            $crawler = new Crawler();
            $crawler->addContent($content);


            /** @var \DOMElement[] $jobNodes */
            $jobNodes = $crawler->filter('ul > li > h2 > a');

            var_dump('https://www.virtualvocations.com/jobs');
            var_dump($i . ' ' . count($jobNodes) . '\n');
            if (count($jobNodes) === 0) {
                break;
            }
            foreach ($jobNodes as $jobNode) {

                $job = new Vacancy();
                $job->setVacancy($jobNode->textContent);
                $job->setUrl('https://www.virtualvocations/' . $jobNode-
>getAttribute('href'));

                $job->setSiteName('VirtualVacancy');

                $this->entityManager->persist($job);
            }

            if (($i % 6) === 0) {
                $this->entityManager->flush();
            }
        }
        $this->entityManager->flush();
```

```
        return 0;
    }
}
class RabotaUa extends Command
{
    protected static $defaultName = 'RabotaUa';

    /**
     * @var EntityManager
     */
    private $entityManager;

    public function __construct(string $name = null, EntityManagerInterface
$entityManager)
    {
        parent::__construct($name);
        $this->entityManager = $entityManager;
    }

    protected function configure()
    {
        $this
            ->setDescription('parsing sie RabotaUa')
            ->addArgument('arg1', InputArgument::OPTIONAL, '')
            ->addOption('option1', null, InputOption::VALUE_NONE, '');
    }

    protected function execute(InputInterface $input, OutputInterface $output):
int
    {

        for ($i = 1; true; $i++) {
            $client = HttpClient::create();
            $response = $client->request('GET', 'https://rabota.ua/zapros/it/pg' .
$i);
            $content = $response->getContent();

            $crawler = new Crawler();
            $crawler->addContent($content);

            /** @var \DOMElement[] $jobNodes */
            $jobNodes = $crawler->filter('#ctl00_content_vacancyList_gridList > tr
> td > article > div.card-body > div.card-main-content > div.common-info > p.card-
title > a');

            var_dump('https://rabota.ua/zapros/it/pg' . $i);
            var_dump($i . ' ' . count($jobNodes) . '\n');
            if (count($jobNodes) === 0) {
                break;
            }

            foreach ($jobNodes as $jobNode) {

                $job = new Vacancy();
                $job->setVacancy($jobNode->textContent);
                $job->setUrl('http://rabota.ua'.$jobNode->getAttribute('href'));
                $job->setSiteName('rabota_ua');
                $job->setSalary(random_int(5, 100) * 1000);

                $this->entityManager->persist($job);
```

```
            }

            if (($i % 6) === 0) {
                $this->entityManager->flush();
            }
        }
        $this->entityManager->flush();

        return 0;
    }
}
class VacancyRepository extends ServiceEntityRepository
{
    public function __construct(ManagerRegistry $registry)
    {
        parent::__construct($registry, Vacancy::class);
    }



    public function getCountByName(string $searchName)
    {
        $qb = $this->createQueryBuilder('r');

        $qb
            ->select('count(r.id)')
            ->andWhere('lower(r.vacancy) LIKE :val')
            ->setParameter('val', '%' . mb_strtolower($searchName) . '%');

        return $qb->getQuery()->getSingleScalarResult();
    }

    public function getJobsByName(string $searchName)
    {
        $qb = $this->createQueryBuilder('r');

        $qb
            ->andWhere('lower(r.vacancy) LIKE :val')
            ->setParameter('val', '%' . mb_strtolower($searchName) . '%');

        return $qb->getQuery()->getResult();
    }

    public function getAverageSalaryByName(string $searchName)
    {
        $qb = $this->createQueryBuilder('r');

        $qb
            ->select('sum(r.salary)')
            ->andWhere('lower(r.vacancy) LIKE :val')
            ->setParameter('val', '%' . mb_strtolower($searchName) . '%');

        $sum = $qb->getQuery()->getSingleScalarResult();

        $vacanciesCount = $this->getCountByName($searchName);

        if ($vacanciesCount === 0) {
            return 0;
        }

        return round($sum / $vacanciesCount);
    }
```

```
    public function getMinSalaryByName(string $searchName)
    {
        $qb = $this->createQueryBuilder('r');

        $qb
            ->select('min(r.salary)')
            ->andWhere('lower(r.vacancy) LIKE :val')
            ->setParameter('val', '%' . mb_strtolower($searchName) . '%');


        return $qb->getQuery()->getSingleScalarResult();
    }

    public function getMaxSalaryByName(string $searchName)
    {
        $qb = $this->createQueryBuilder('r');

        $qb
            ->select('max(r.salary)')
            ->andWhere('lower(r.vacancy) LIKE :val')
            ->setParameter('val', '%' . mb_strtolower($searchName) . '%');


        return $qb->getQuery()->getSingleScalarResult();
    }

}
/**
 * @ORM\Entity(repositoryClass="App\Repository\VacancyRepository")
 */
class Vacancy
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $vacancy;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $url;

    /**
     * @ORM\Column(type="integer", nullable=true)
     */
    private $salary;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $siteName;

    /**
     * @ORM\OneToOne(targetEntity="WorkUa" , mappedBy="vacancyMain")
     */
```

```
    private $workUa;

    public function getId(): ?int
    {
        return $this->id;
    }

    /**
     * @return mixed
     */
    public function getSiteName()
    {
        return $this->siteName;
    }

    /**
     * @param mixed $siteName
     */
    public function setSiteName($siteName): void
    {
        $this->siteName = $siteName;
    }

    public function getVacancy(): ?string
    {
        return $this->vacancy;
    }

    public function setVacancy(string $vacancy): self
    {
        $this->vacancy = $vacancy;

        return $this;
    }

    public function getUrl(): ?string
    {
        return $this->url;
    }

    public function setUrl(string $url): self
    {
        $this->url = $url;

        return $this;
    }

    /**
     * @return mixed
     */
    public function getSalary()
    {
        return $this->salary;
    }

    /**
     * @param mixed $salary
     */
    public function setSalary($salary): void
    {
        $this->salary = $salary;
    }
```

```
}
class VacancyController extends FOSRestController
{
    /**
     * @Rest\Post(path="/getApi", name="get_api")
     * @param Request $request
     * @return JsonResponse
     */
    public function ApiAll(Request $request)
    {
        $content = $request->getContent();
        $jobNames = json_decode($content, true)['jobs'];
        $result = ['labels' => [], 'values' => []];

        foreach ($jobNames as $jobName) {
            $repository = $this->getDoctrine()->getRepository(Vacancy::class);
            $result['labels'][] = $jobName;
            $result['values'][] = $repository->getCountByName($jobName);
        }

        return new JsonResponse($result);
    }

    /**
     * @Rest\Post(path="/getApiSalary", name="get_api_salary")
     * @param Request $request
     * @return JsonResponse
     */
    public function ApiAllSalaries(Request $request)
    {
        $content = $request->getContent();
        $jobNames = json_decode($content, true)['jobs'];
        $result = ['labels' => [], 'values' => []];

        foreach ($jobNames as $key => $jobName) {
            $repository = $this->getDoctrine()->getRepository(Vacancy::class);
            $result['labels'][] = $jobName;
            $result['values'][] = $repository->getAverageSalaryByName($jobName);
        }

        return new JsonResponse($result);
    }

    /**
     * @Rest\Post(path="/getApiMinSalary", name="get_api_min_salary")
     * @param Request $request
     * @return JsonResponse
     */
    public function minimumSalary(Request $request)
    {
        $content = $request->getContent();
        $jobNames = json_decode($content, true)['jobs'];
        $result = ['labels' => [], 'values' => []];

        foreach ($jobNames as $jobName) {
            $repository = $this->getDoctrine()->getRepository(Vacancy::class);
            $result['labels'][] = 'MIN ' . $jobName;
            $result['values'][] = $repository->getMinSalaryByName($jobName);

            $result['labels'][] = 'MAX ' . $jobName;
            $result['values'][] = $repository->getMaxSalaryByName($jobName);
```

```
        }

        return new JsonResponse($result);
    }

    /**
     * @Route("/jobs/{jobName}", name="jobs_by_job_name")
     */
    public function jobs(string $jobName)
    {

        $repository = $this->getDoctrine()->getRepository(Vacancy::class);
        $resultVacancies = $repository->getJobsByName($jobName);
        $result = [];

        /** @var Vacancy $resultVacancy */
        foreach ($resultVacancies as $resultVacancy) {
            $result[] = ['job_name' => $resultVacancy->getVacancy(), 'url' =>
$resultVacancy->getUrl()];
        }

        return new JsonResponse($result);
    }

    /**
     * @Route("/canvas", name="canvas")
     */
    public function graph()
    {
        return $this->render('canvasJs.html.twig');
    }

    /**
     * @Route("/input", name="input")
     */
    public function input()
    {
        return $this->render('input.html.twig');
    }

    /**
     * @Route("/table", name="table")
     */
    public function table()
    {
        $em = $this->getDoctrine()->getManager();
        $table = $em->getRepository(Vacancy::class)->findAll();
        return $this->render('table.html.twig', [
        'tables' => $table]);
    }
}
```

Front-End:

```
{% block head %}
    <style type="text/css">
        TABLE {
            width: 500px; /* Ширина таблицы */
            border-collapse: collapse;
        }

        TD, TH {
            padding: 1px;
            border: 1px solid grey;
        }

        TH {
            background: white;
        }
    </style>

    {% block body %}
        <table style="float: left">
            <tr>
                <th>Мови програмування</th>
                <th>Мін Зарплата</th>
                <th>Макс Зарплата</th>
                <th>Серендя Зарплата</th>
            </tr>
            <tr>
                <th>Java</th>
                <th>500$</th>
                <th>5000$</th>
                <th>1600$</th>
            </tr>
            <tr>
                <th>PHP</th>
                <th>300$</th>
                <th>4000$</th>
                <th>1500$</th>
            </tr>
            <tr>
                <th>C#</th>
                <th>300$</th>
                <th>5000$</th>
                <th>1000$</th>
            </tr>
            <tr>
                <th>Python</th>
                <th>400$</th>
                <th>4000$</th>
                <th>1200$</th>
            </tr>
            <tr>
                <th>Ruby on Rails</th>
                <th>420$</th>
                <th>2,800$</th>
                <th>1300$</th>
            </tr>
            <tr>
                <th>1C</th>
                <th>300$</th>
                <th>2500$</th>
                <th>1400$</th>
```

```
            </tr>
            <tr>
                <th>C++</th>
                <th>500$</th>
                <th>3900$</th>
                <th>1800$</th>
            </tr>
            <tr>
                <th>.NET</th>
                <th>300$</th>
                <th>2500$</th>
                <th>1200$</th>
            </tr>
            <tr>
                <th>SWIFT</th>
                <th>400$</th>
                <th>3000$</th>
                <th>1500$</th>
            </tr>
            <tr>
                <th>R</th>
                <th>400$</th>
                <th>3500$</th>
                <th>1400$</th>
            </tr>
            <tr>
                <th>Angular</th>
                <th>500$</th>
                <th>5000$</th>
                <th>1600$</th>
            </tr>
            <tr>
                <th>SQL</th>
                <th>200$</th>
                <th>2500$</th>
                <th>900$</th>
            </tr>
        </table>

        <table style="float:right">
            <tr>
                <th>Назва вакансій</th>
                <th>Мін Зарплата</th>
                <th>Макс Зарплата</th>
                <th>Середня Зарплата</th>
            </tr>
            <tr>
                <th>Java Developer Senior</th>
                <th>2000$</th>
                <th>6000$</th>
                <th>3500$</th>
            </tr>
            <tr>
                <th>PHP Developer Symfony Mid</th>
                <th>1500$</th>
                <th>3000$</th>
                <th>2000$</th>
            </tr>
            <tr>
                <th>Ситсемний аналітик</th>
                <th>300$</th>
                <th>3000$</th>
```

```
                <th>900$</th>
            </tr>
            <tr>
                <th>Database Developer</th>
                <th>400$</th>
                <th>2000$</th>
                <th>800$</th>
            </tr>
            <tr>
                <th>Angular Front-End Developer</th>
                <th>300$</th>
                <th>3000$</th>
                <th>1200$</th>
            </tr>
            <tr>
                <th>Swift IOS Developer</th>
                <th>400$</th>
                <th>3000$</th>
                <th>1500$</th>
            </tr>
            <tr>
                <th>Java Back-End Developer Middle</th>
                <th>1200$</th>
                <th>2500$</th>
                <th>1500$</th>
            </tr>
            <tr>
                <th>1C Developer Bitrix</th>
                <th>200$</th>
                <th>800$</th>
                <th>400$</th>
            </tr>
            <tr>
                <th>Android Developer</th>
                <th>300$</th>
                <th>2500$</th>
                <th>900$</th>
            </tr>
            <tr>
                <th>QA engineer</th>
                <th>300$</th>
                <th>1500$</th>
                <th>900$</th>
            </tr>
        </table>
    {% endblock %}

{% endblock %}
<html>
<head>
    <link rel="stylesheet" type="text/css"
href="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.3/Chart.css">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.3/Chart.bundle.js"></scri
pt>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.3/Chart.js"></script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
</head>
<body>
```

```
<div style="flex-direction: row; display: flex; justify-content: center;">
    <canvas id="myChart" width="700" height="300"></canvas>
    <canvas id="vacancyCount" width="700" height="300"></canvas>
</div>
<div style="flex-direction: row; display: flex; justify-content: center;">
    <canvas id="myChartMin" width="700" height="300"></canvas>
</div>
<div id="button_container" style="flex-direction: row; display: flex; justify-
content: center; padding: 50px">

</div>
</body>
<script>
    window.onload = function () {

        //запрос для получения  данных для графика кол-ва вакансий
        var requestConfig = {
            method: 'post',
            url: "{{ path('get_api') }}",
            data: {
                jobs: arr
            },
        };

        var result;
        //непосредственно сам запрос
        axios.request(requestConfig)
            .then((data) => {
                result = data.data;
            }).then(() => {

            // ищем элемент с графиком на странице
            var ctx = document.getElementById('vacancyCount');

            // отрисовка графика данными из запроса
            var myChart = new Chart(ctx, {
                type: 'line',
                data: {
                    labels: result.labels,
                    datasets: [{
                        label: 'Кількість вакансій',
                        data: result.values,
                        backgroundColor: [
                            'rgba(54, 162, 235, 0.2)',
                            'rgba(255, 206, 86, 0.2)',
                            'rgba(75, 192, 192, 0.2)',
                            'rgba(153, 102, 255, 0.2)',
                            'rgba(255, 159, 64, 0.2)'
                        ],
                        borderColor: [
                            'rgba(54, 162, 235, 1)',
                            'rgba(255, 206, 86, 1)',
                            'rgba(75, 192, 192, 1)',
                            'rgba(153, 102, 255, 1)',
                            'rgba(255, 159, 64, 1)'
                        ],
                        borderWidth: 1
                    }]
                },

                options: {
                    responsive: false,
```

```
                scales: {
                    yAxes: [{
                        ticks: {
                            beginAtZero: true
                        }
                    }]
                }
            }
        });
    })

    var requestConfig = {
        method: 'post',
        url: "{{ path('get_api_salary') }}",
        data: {
            jobs: jobs
        },
    };

    var result;
    axios.request(requestConfig)
        .then((data) => {
            result = data.data;
        }).then(() => {
        var ctx = document.getElementById('myChart');

        var myChart = new Chart(ctx, {
            type: 'bar',
            data:
                {
                    labels: result.labels,
                    datasets: [{
                        label: 'Заплати в $',
                        data: result.values,
                        backgroundColor: [
                            'purple',
                            'red',
                            'grey',
                            'blue',
                            'green',
                            'yellow',
                            'pink',
                            'orange',
                            'violet',
                            'black',
                            'aqua',
                            'royalblue',
                            'skyblue',
                            'darkblue',
                            'lime',
                            'steelblue',
                            'silver',
                            'bronze',
                            'crimson',
                            'orchid',
                            'teal',
                            'azure'
                        ],
                        borderColor: [
                            'purple',
                            'red',
                            'grey',
```

```
                            'blue',
                            'green',
                            'yellow',
                            'pink',
                            'orange',
                            'violet',
                            'black',
                            'aqua',
                            'royalblue',
                            'skyblue',
                            'darkblue',
                            'lime',
                            'steelblue',
                            'silver',
                            'bronze',
                            'crimson',
                            'orchid',
                            'teal',
                            'azure'
                        ],
                        borderWidth: 1
                    }]
                },

            options: {
                responsive: false,
                scales: {
                    yAxes: [{
                        ticks: {
                            beginAtZero: true
                        }
                    }]
                }
            }
        });
    })

    var requestConfig = {
        method: 'post',
        url: "{{ path('get_api_min_salary') }}",
        data: {
            jobs: jobs
        },
    };

    var result;
    axios.request(requestConfig)
        .then((data) => {
            result = data.data;
        }).then(() => {
        var ctx = document.getElementById('myChartMin');

        var myChart = new Chart(ctx, {
            type: 'doughnut',
            data: {
                labels: result.labels,
                datasets: [{
                    label: 'Зарплаты min max $',
                    data: result.values,
                    backgroundColor: [
                        'purple',
                        'red',
```

```
                        'grey',
                        'blue',
                        'green',
                        'yellow',
                        'pink',
                        'orange',
                        'violet',
                        'black',
                        'aqua',
                        'royalblue',
                        'skyblue',
                        'darkblue',
                        'lime',
                        'steelblue',
                        'silver',
                        'bronze',
                        'crimson',
                        'orchid',
                        'teal',
                        'azure'
                    ],
                    borderColor: [
                        'purple',
                        'red',
                        'grey',
                        'blue',
                        'green',
                        'yellow',
                        'pink',
                        'orange',
                        'violet',
                        'black',
                        'aqua',
                        'royalblue',
                        'skyblue',
                        'darkblue',
                        'lime',
                        'steelblue',
                        'silver',
                        'bronze',
                        'crimson',
                        'orchid',
                        'teal',
                        'azure'
                    ],
                    borderWidth: 1
                }]
            },

            options: {
                responsive: false,
                scales: {
                    yAxes: [{
                        ticks: {
                            beginAtZero: true
                        }
                    }]
                }
            }
        });
    })
```

```
    }


    //Получаем параметры которые пришли с /input
    const urlParams = new URLSearchParams(window.location.search);
    var arr = [];

    // перебираем их так чтобы получился массив

    //в первом случае обрабатывает если данные пришли с инпута
    if (urlParams.get('jobs')) {
        //разбиваем строку с параметрами пофразно с разделителем  - ,
        arr = urlParams.get('jobs').split(',');

        //если данные с инпута пустые берем данные чекбоксов
    } else {
        //получение данных с чекбокса
        for (var value of urlParams.keys()) {
            if (value == 'jobs') {
                continue;
            }
            arr.push(urlParams.get(value));
        }
    }

    const jobs = arr;
    var array = jobs

    //ищем контейнер на странице в котором будем отрисовывать кнопки
    var divBlock = document.getElementById('button_container');
    // проходимя по массиву с названиями вакансий который мы получили на строках
196-214
    for (var i = 0; i < array.length; i++) {

        //создаем тег ссылки
        var a = document.createElement("a");
        //создаем тег кнопки
        var button = document.createElement("button");
        //задаем ссылке URL с названием вакансии
        a.setAttribute('href', '/jobs/' + array[i]);
        //задаем чтобы открывался в нововой странице
        a.setAttribute('target', '_blank');
        //задаем текст внутри кнопки назаванием вакансии которое получили на
строках 196-214
        button.textContent = array[i];
        // оборачиваем тегом ссылки кнопку чтобы было что-то вроде <a
href="/jobs/название вакансии"><button> название вакансии</button></a>
        a.appendChild(button);
        //добавляем нашу кнопку в наш контейнер с кнопками
        divBlock.appendChild(a);
    }


</script>
```

ДОДАТОК Б

Міністерство освіти і науки України
Державний університет телекомунікацій
Навчально-науковий інститут інформаційних технологій
Кафедра системного аналізу

за темою «Система аналізу рейтингів вакансій, розташованих на сайтах пошуку роботи.»

Виконав: студент групи САД-41
Кожедуб Віталій Олегович
Науковий керівник:
ст.вик. кафедри Системного аналізу, Штіммерман А.М.

Київ-2020

2 Індивідуальне завдання

1. Провести аналітичний огляд бізнес-процесів інформаційного забезпечення в галузі пошуку вакансій.

2. Проаналізувати програмні, технічні, організаційні, інфокомунікаційні та інші рішення, що використовуються на поточний момент в галузі пошуку вакансій та описати моделі інформаційного забезпечення в галузі пошуку вакансій.

3. Визначити недоліки та проблемні питання в галузі пошуку вакансій.

4. Розробити скрипт для збору інформації про вакансії з різних джерел пошуку роботи.

5. Створити статистику за запитом користувача про популярність вакансій та заробітну плату.

3 Актуальність бакалаврської роботи

Системи обробки і пошуку вакансій, розташовані на великій кількості ресурсів, тому недоліками існуючих рішень аналізу сайтів працевлаштування, є розрізненість інформації і як наслідок, довгий час обробки.

Показниками ефективності, бакалаврської роботи, є прискорення часу на збереження та обробку даних, отриманих з різних відкритих Інтернет-джерел працевлаштування та спрощення взаємодії користувача з великою кількістю інформації. Система аналізу рейтингів вакансії, розроблена в бакалаврській роботі, надає зручність у використані аналітики з сайтів працевлаштування.

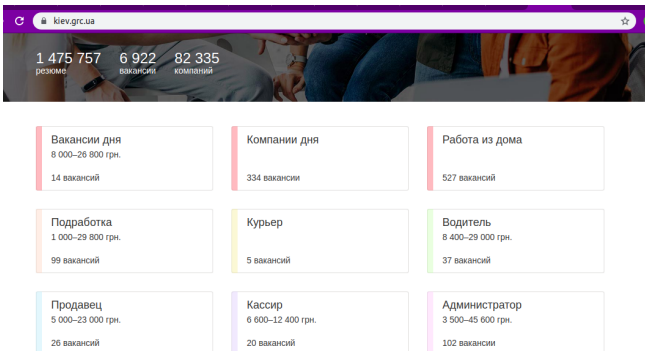
Об’єкт, предмет, мета бакалавської роботи 4

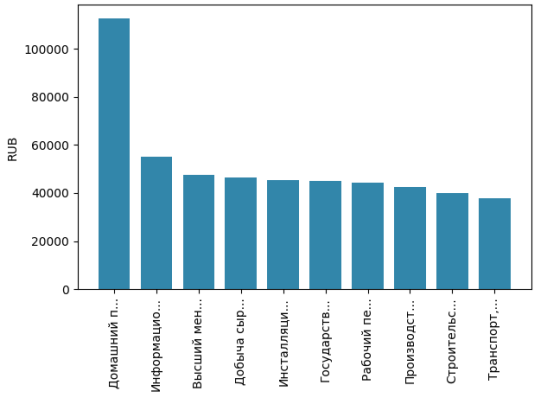
Об’єкт дослідження — процеси автоматизації збору та аналізу даних про вакансії (вхідними даними, є інформація з різних джерел, а вихідними — створення статистики за запитом користувача про популярність вакансій)

Предмет дослідження – методи проектування та технології створення аналізу вакансій, зібраних на сайтах працевлаштування.

Мета роботи – підвищення якості автоматизованого процесу аналітики з сайтів працевлаштування (статистика, які професії найпопулярніші, які приносять більші заробітні плати, які є більш збитковими.)

Огляд інформаційного забезпечення в галузі пошуку вакансій 5



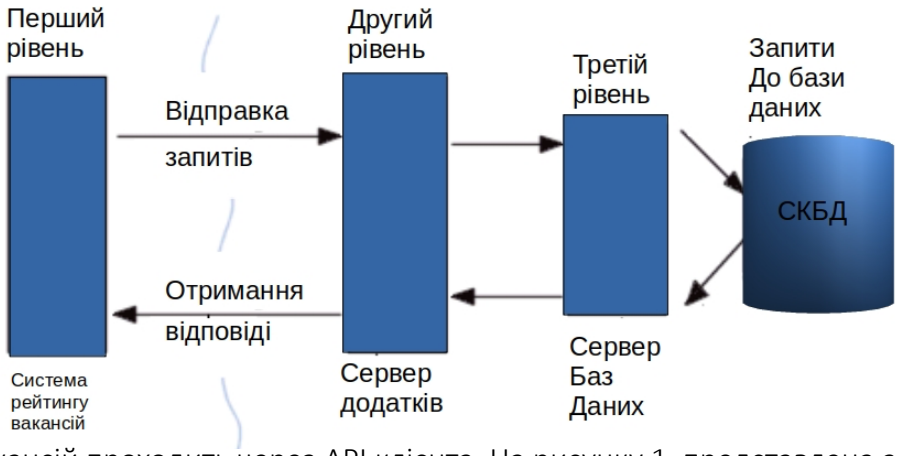


1. Результати пошуку містять дані про вакансії у вигляді, визначеному на сайті і нема можливості будувати графіки та отримати аналітику по будь-якому запиту

2. Якщо потрібно збирати інформацію з усіх можливих сфер заробітку, існуючі варіанти пошуку сайтів повертають відповідь у тісному діапазоні і немає повного доступу до ресурсів API сайту для фільтрації та переробки для обраної мети.

Аналіз інфокомунікаційних рішень, що використовуються на поточний момент в галузі пошуку вакансій

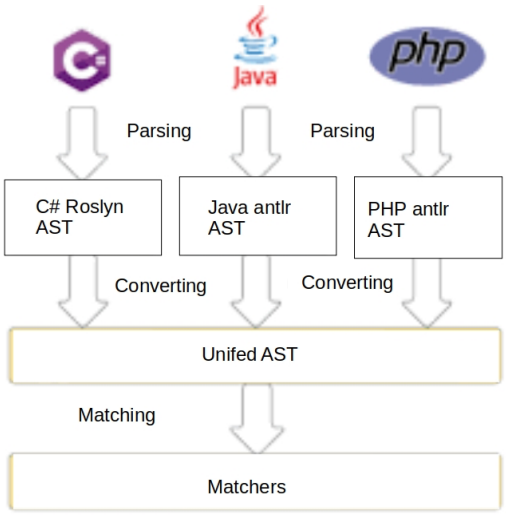
6



Аналіз вакансій проходить через API клієнта. На рисунку 1, представлена архітектура клієнт-серверної взаємодії системи аналізу рейтингів вакансій з СКБД.

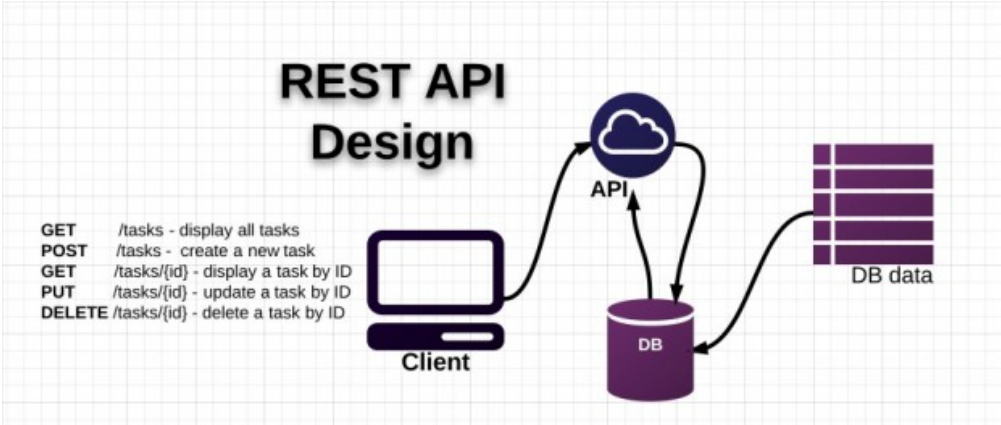
Недоліки та проблемні питання в існуючих рішеннях пошуку вакансій.

7



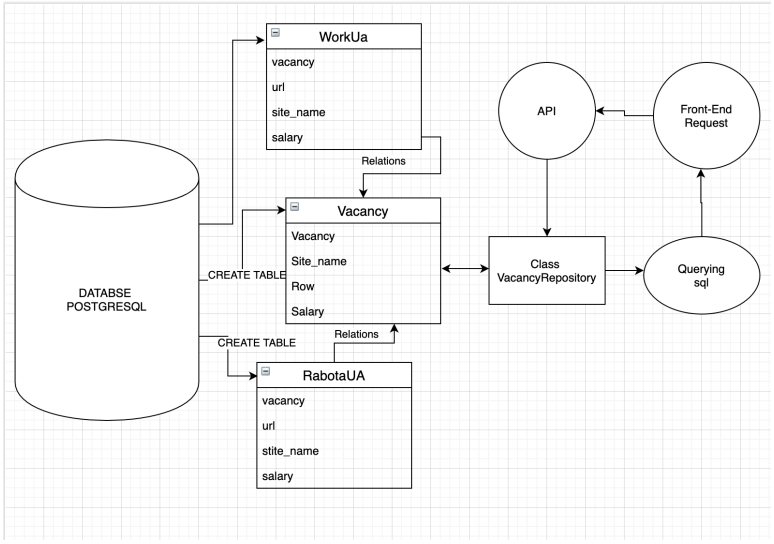
Незважаючи на використання API для збіра даних, у вирішенні задач з оперативним представленням інформації, щодо пошуку вакансій, доцільно використовувати більш швидке рішення — скрипт, якій є парсером.

Технології створення аналізу вакансій, зібраних на сайтах працевлаштування. 8



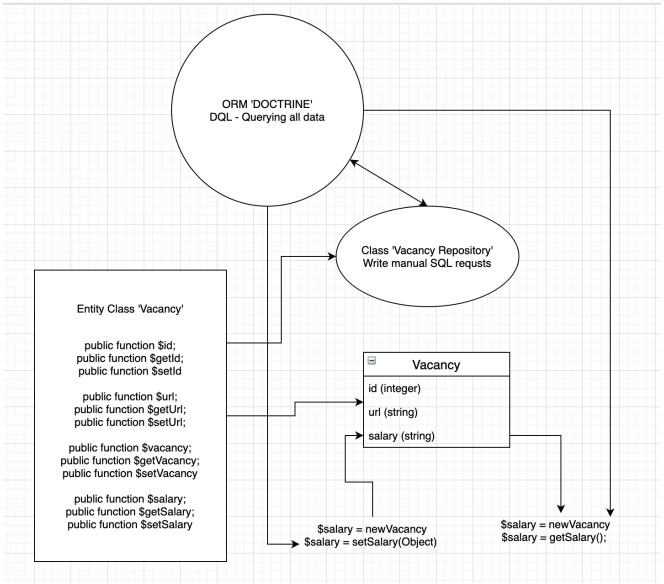
API це частина сервера, яка приймає запити та надсилає відповіді. На рисунку 2, наведено взаємодію API та набір запитів.

Модель бази даних системи аналізу рейтингів вакансій 9



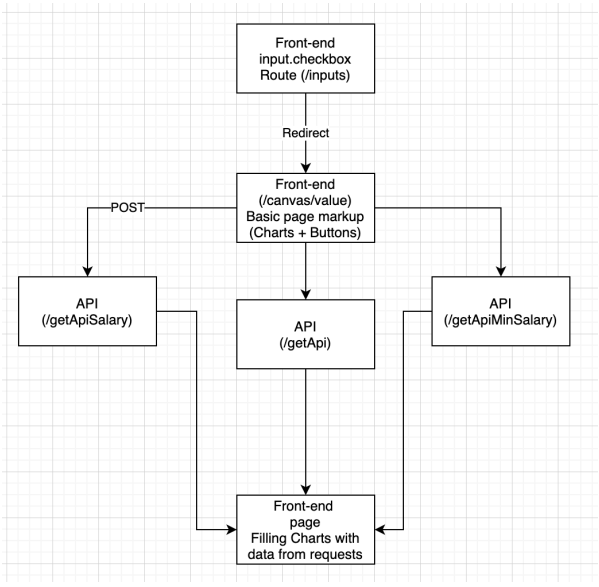
Основними компонентами бази даних системи аналізу рейтингів вакансій, є таблиці, у яких зберігатимуться дані про вакансії.

Діаграма взаємодії класів системи і таблиць бази даних 10



На рисунку представлена діаграма, відображаюча класи інформаційної системи. Кожна таблиця у схемі розглядається як клас

Діаграма компонентів системи аналізу рейтингів вакансій 11



На рисунку показано процес роботи клієнтської сторони системи аналізу рейтингів вакансій

12

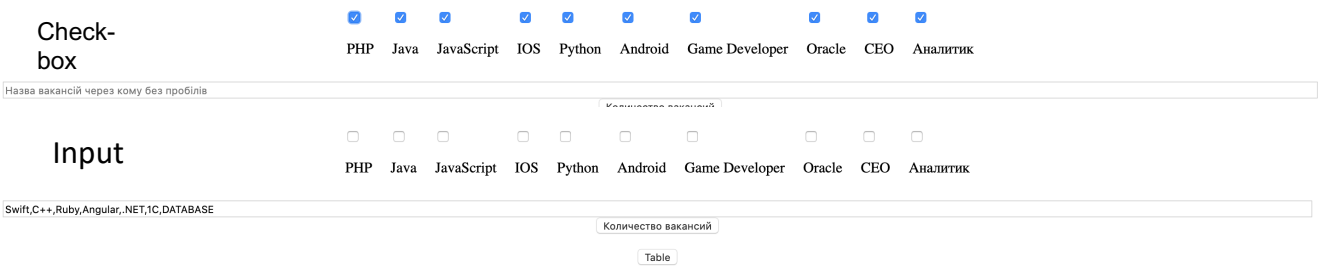
АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ

В бакалаврської роботі були проведені наступні етапи аналізу:

— Визначення параметрів, за якими система аналізу рейтингів вакансій проводить обрахування;

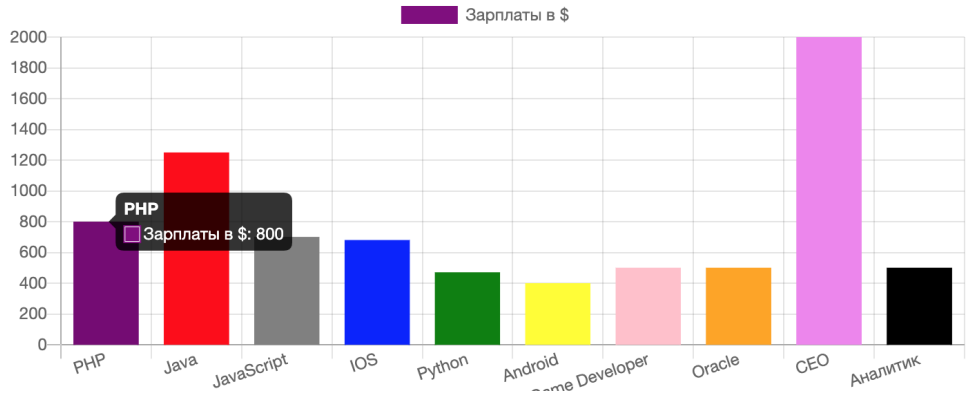
— Побудова вибірки аналітики за допомогою Інтернет-додатка системи аналізу рейтингів вакансій;

— Перевірка побудови вибіркової сукупності.



13

АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ



Графік який відображає середню заробітну плату за обраними користувачем вакансіями

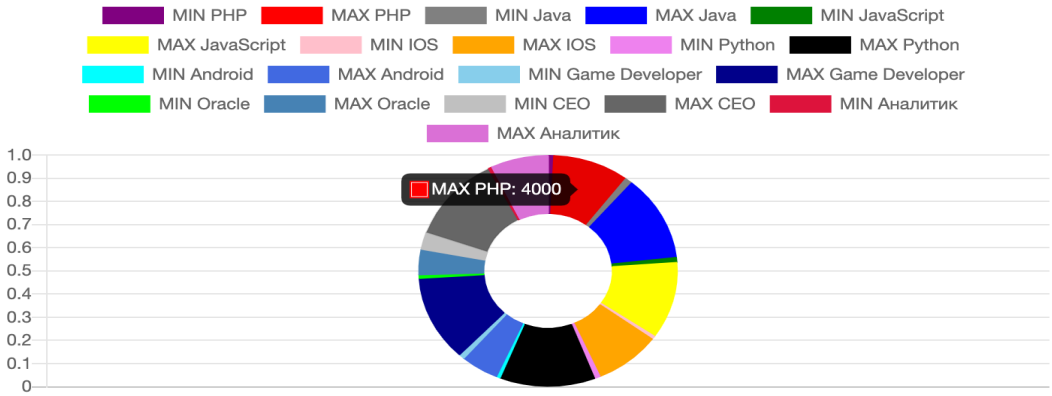
14

АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ

Графік який відображає попит на спеціалістів у галузі програмування та аналітики.

15

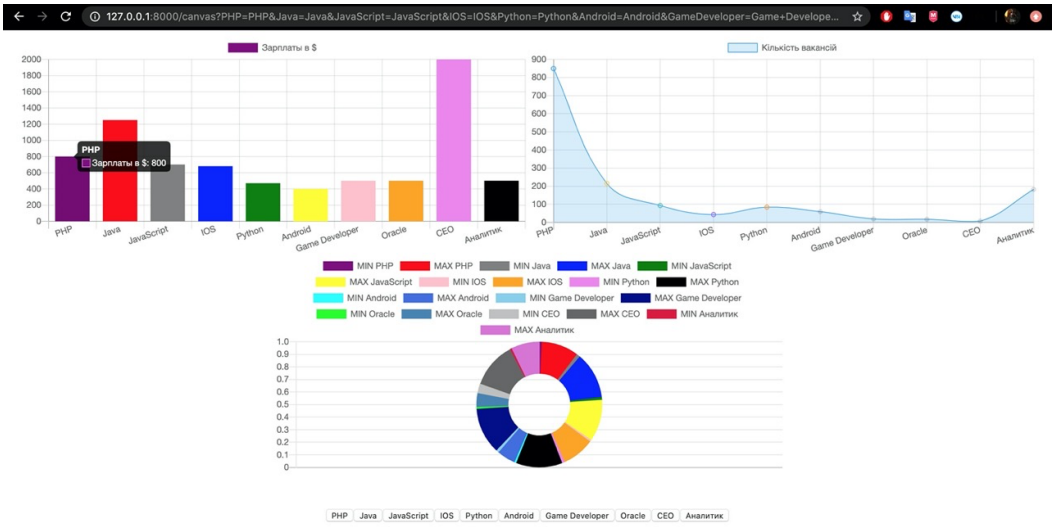
АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ



Графік з відображенням мінімальні та максимальні заробітних плат у галузі програмування та аналітики.

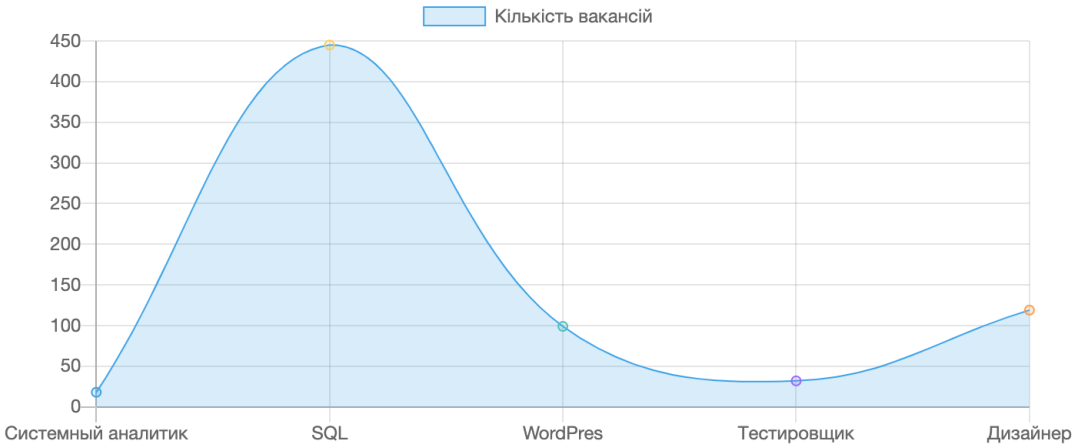
16

Сторінка Результату пошуку даних по вакансіям.



17

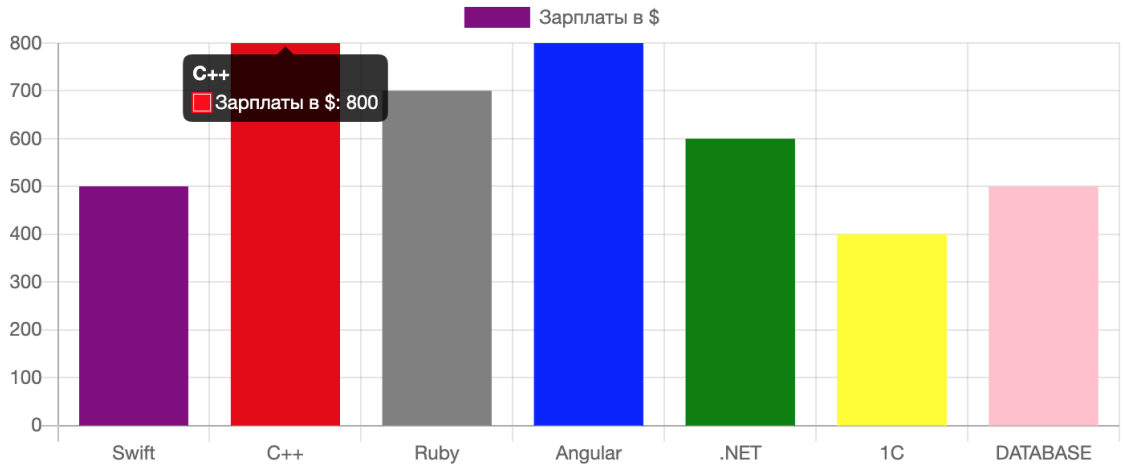
АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ



Графік з відображенням Попит на спеціалістів у галузі створення програмних продуктів.

18

АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ



Графік з відображенням рівень заробітної плати програмістів

19

АНАЛІЗ ДАНИХ З САЙТІВ ПРАЦЕВЛАШТУВАННЯ

Мови програмування	Мін Зарплата	Макс Зарплата	Серендя Зарплата
Java	500$	5000$	1600$
PHP	300$	4000$	1500$
C#	300$	5000$	1000$
Python	400$	4000$	1200$
Ruby on Rails	420$	2,800$	1300$
1C	300$	2500$	1400$
C++	500$	3900$	1800$
.NET	300$	2500$	1200$
SWIFT	400$	3000$	1500$
R	400$	3500$	1400$
Angular	500$	5000$	1600$
SQL	200$	2500$	900$

Таблиця 1 - Розподіл мінімальних, максимальних і середніх заробітних

Плат спеціалістів у галузі програмування

Посилання на джерела вакансій

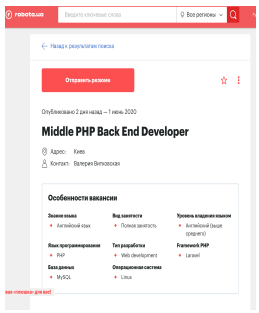
20

PHP | Java | JavaScript | IOS | Python | Android | Game Developer | Oracle | CEO | Аналитик

Для кожної вакансії яку ми шукаємо генеруються кнопки з назвами вакансій.

При натисканні на кнопку ми отримаємо всі вакансії і посилання на сайт де знаходиться ця вакансія.


```
[
  {
    "job_name": "Middle+/Senior PHP Developer",
    "url": "http://work.ua/ru/jobs/3894659/"
  },
  {
    "job_name": "Програміст PHP (Bitrix 24)",
    "url": "http://work.ua/ru/jobs/3890810/"
  },
  {
    "job_name": "Full Stack программист (PHP, Laravel) Middle",
    "url": "http://work.ua/ru/jobs/3892169/"
  },
  {
    "job_name": "Програміст PHP",
    "url": "http://work.ua/ru/jobs/3874076/"
  },
  {
    "job_name": "Разработчик Full Stack, PHP",
    "url": "http://work.ua/ru/jobs/3360629/"
  },
  {
    "job_name": "PHP developer (стажер, с обучением Bitrix Framework)",
    "url": "http://work.ua/ru/jobs/3514259/"
  },
  {
    "job_name": "Middle PHP Developer",
    "url": "http://work.ua/ru/jobs/3544916/"
  },
  {
    "job_name": "Программист Middle PHP (Yii2)",
```

21

Висновок

В бакалаврській роботі була розроблена системи аналізу рейтингів вакансій, розташованих на сайтах пошуку роботи. Інтернет-додаток, наглядно демонструє аналіз вакансій у сфері інформаційних технологій, показує зріст чи спадання попиту на вакансії, завдяки обробці інформації з різних сайтів працевлаштування. В бакалаврській роботі були вирішені наступні питання:

- Розробка скрипта для збору інформації про вакансії з різних джерел пошуку роботи.
- Покращення швидкості обробки зібраної інформації з джерел пошуку роботи.

3. Створення статистики за запитом користувача про популярність вакансій, заробітну плату, потрібний попередній досвід.

Основними перевагами додатку для аналізу вакансій, є максимальна зручність для користувача в обробці інформації про вакансії.