

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інтегровані інформаційні системи»
спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Система пошуку вакансій з використанням геолокаційної
системи»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІА-91

Іщенко Андрій Віталійович _____

Керівник:

доцент кафедри ІСТ, к.т.н., доцент

Резніков Сергій Анатолійович _____

Рецензент:

доцент кафедри ІІІ, к.т.н., доцент

Баклан Ігор Всеволодович _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Іщенко Андрію Віталійовичу

1. Тема проєкту «Система пошуку вакансій з використанням геолокаційної системи», керівник проєкту Резніков Сергій Анатолійович, доцент, затверджені наказом по університету від «31» травня 2023 р. №2101-с
2. Термін подання студентом проєкту: 12 червня 2023
3. Вихідні дані до проєкту: гнучкість та зручність використання, здатність різних частин застосунку працювати незалежно одна від одної
4. Зміст пояснювальної записки: Проаналізувати існуючі реалізації. Обрати необхідні засоби для реалізації. Сформулювати вимоги до програмного забезпечення. Реалізувати функціонал відповідно до вимог.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): алгоритм показу сторінок, алгоритм парсингу вакансій, IDEF3 діаграма Додати вакансію в БД, діаграма активності
6. Дата видачі завдання 1 березня 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд та аналіз предметної області	01.03.2023 – 08.03.2023	
2	Аналіз та порівняння існуючих рішень	09.03.2023 – 11.03.2023	
3	Ознайомлення з бібліотекою PyQt	12.03.2023 – 13.03.2023	
4	Розроблення користувацького інтерфейсу	14.03.2023 – 28.03.2023	
5	Ознайомлення з Google Maps API	29.03.2023 – 30.03.2023	
6	Інтеграція Google Maps API в застосунок	31.03.2023 – 18.04.2023	
7	Ознайомлення з бібліотекою QGIS	19.04.2023 – 21.04.2023	
8	Реалізація фільтрування вакансій	22.04.2023 – 30.04.2023	
9	Виконання графічних документів	01.05.2023 – 10.05.2023	
10	Оформлення пояснювальної записки	11.05.2023 – 20.05.2023	

Студент

Андрій ІЩЕНКО

Керівник

Сергій РЕЗНІКОВ

АНОТАЦІЯ

Іщенко А.В. Система пошуку вакансій з використанням геолокаційної системи. КПІ ім. Ігоря Сікорського, Київ, 2023.

Проект містить 60 с. тексту, 14 рисунків, 2 таблиці, посилання на 16 літературних джерел та 4 конструкторські документи.

Ключові слова: пошук вакансій, геоінформаційні системи, QGIS, Google Maps.

Об'єктом розробки є система пошуку вакансій за віддаленістю від заданої точки.

Мета розробки – полегшення процесу пошуку роботи для користувачів за рахунок надання інструменту фільтрування вакансій в межах, заданих користувачами та надання зручного користувацького інтерфейсу для виконання поставлених задач.

В результаті виконання дипломного проєкту було виконано поставлену мету. Було створено настільний застосунок, здатний надати користувачам інтуїтивно зрозумілий інтерфейс. Застосунок виконує покладені на нього задачі, такі як фільтрація вакансій за віддаленістю від заданої точки, оновлення бази даних вакансій та очищення її від застарілих оголошень.

SUMMARY

Ishchenko A.V. GIS-based job search system. Igor Sikorsky KPI, Kyiv, 2023.

The project contains 60 pages of text, 14 figures, 2 tables, references to 16 literary sources and 4 design documents.

Keywords: job search, geoinformation systems, QGIS, Google Maps.

The object of development is a job search system by distance from a given point.

The purpose of the development is to facilitate the job search process for users by providing a tool for filtering vacancies within the distance set by users and providing a convenient user interface for performing the tasks.

As a result of the completion of the project, the set goal was fulfilled. A desktop application was created, capable of providing users with an intuitive interface. The application performs tasks assigned to it, such as filtering vacancies by distance from a given point, updating the database of vacancies and cleaning it of outdated ads.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер елем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA91.110БАК.003 ПЗ	Пояснювальна записка	60		
6	A3	IA91.110БАК.003 Д1	Система пошуку вакансій з	1		
7			використанням			
8			геолокаційної системи.			
9			Алгоритм відображення			
10			сторінок			
11	A3	IA91.110БАК.003 Д2	Система пошуку вакансій з	1		
12			використанням			
13			геолокаційної системи.			
14			Алгоритм парсингу вакансій			
15	A3	IA91.110БАК.003 Д3	Система пошуку вакансій з	1		
16			використанням			
17			геолокаційної системи.			
18			IDEF3 Додати вакансію в БД			
19	A3	IA91.110БАК.003 Д4	Система пошуку вакансій з	1		
20			використанням			
21			геолокаційної системи.			
22			Діаграма активності UML			
23						
24						
25						
26						
27						
28						

IA91.110БАК.003 ТП

Зм.	Аркуш	№ докум.	Підпис	Дата				
Розроб.		Іщенко А.В.			Система пошуку вакансій з використанням геолокаційної системи	Літ.	Аркуш	Аркушів
Керівн.		Резніков С.А.				Т	1	60
					Відомість проекту			
Затв.					КПІ ім. Ігоря Сікорського Група ІА-91			

Пояснювальна записка
до дипломного проєкту
на тему: «Система пошуку вакансій з
використанням геолокаційної системи»

Київ – 2023 року

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Характеристика сфери працевлаштування	6
1.2 Опис систем пошуку вакансій.....	8
1.3 Роль геоінформаційних систем в пошуку вакансій.....	9
1.4 Огляд існуючих рішень.....	11
1.4.1 work.ua.....	11
1.4.2 robota.ua.....	13
Висновки до розділу	16
2 ОПИС ПРОЦЕСУ ПРОЄКТУВАННЯ.....	17
2.1 Вимоги до програмного забезпечення.....	17
2.1.1 Функціональні вимоги	17
2.1.2 Нефункціональні вимоги.....	18
2.2 Огляд обраних технологій та бібліотек	18
2.2.1 Геоінформаційна система QGIS	18
2.2.2 Мова програмування Python	19
2.2.3 Google Maps API.....	21
2.2.4 Бібліотека PyQt.....	22
2.2.5 База даних PostgreSQL.....	24
2.2.6 Бібліотека BeautifulSoup4.....	25
Висновки до розділу	26

					ІА91.110БАК.003 ПЗ				
Зм.	Лист	№ докум.	Підпис	Т	Літ.	Арк.	Аркушів		
Розробив	Іщенко А.В.								
Перевірив	Резніков С.А.					2	60		
Затв.					КПІ ім. Ігоря Сікорського				
					Група ІА-91				
					Система пошуку вакансій з використанням геолокаційної системи. Пояснювальна записка				

3	ОГЛЯД ЗАСТОСУНКУ	28
3.1	Клієнтська частина	28
3.2	Реалізація клієнтської частини	34
3.3	Серверна частина	47
3.4	Безпека та налаштуваність	55
	Висновки до розділу	56
	ВИСНОВОК.....	58
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
	Додаток А	61

ВСТУП

Сучасна освіта вважається важливим соціально-економічним благом. Вона покращує якість життя та розширює можливості людини для повноцінного існування. Освіта також має економічне значення, оскільки освічені працівники зазвичай отримують вищу оплату праці. Проте не кожен вид освіти гарантує високі доходи, а лише ті, на які існує попит на ринку праці. Чим більше попит на фахівців, тим вище їх заробітна плата, і якщо отримана освіта не відповідає вимогам роботодавців, фахівець може залишитися безробітним і йому доведеться навчатися знову або влаштовуватися на некваліфіковану роботу.

Проблема працевлаштування випускників вузів стає особливо гострою в останні роки. Зростає попит на робочі спеціальності, але роботодавці вимагають наявності робочого досвіду, якого у молодих фахівців зазвичай немає. Студенти останніх курсів університету найчастіше зосереджуються на мотивах матеріального благополуччя та самореалізації при виборі місця роботи. Мотиви, пов'язані зі значущістю та суспільним визнанням, виявляються менш впливовими. Найбільшим попитом серед випускників зазвичай користуються вакансії з високою оплатою та перспективністю, але більшість вакансій, які пропонує служба зайнятості населення, не відповідають цим критеріям.

Служба зайнятості населення має завдання вирішувати питання, пов'язані з працевлаштуванням і кадровим забезпеченням усіх галузей діяльності в державі. Важливо визначати спосіб підготовки та професії, які найбільше затребувані на ринку праці. Актуальність цієї теми полягає в тому, що працевлаштування молоді на сьогоднішній день є однією з найгостріших соціально-економічних проблем. Ринок праці та ринок освітніх послуг, подібно до товарного ринку, розвиваються відповідно до законів попиту та

					ІА91.110БАК.003 ПЗ	Арк.
						4
Зм.	Лист	№ докум.	Підпис	Дата		

пропозиції. Тому важливо зберігати баланс між цими ринками для процвітання економіки України та суспільства в цілому.

Система пошуку вакансій за відстанню від певної точки на карті базується на зборі та аналізі вакансій, які доступні на різних платформах та сайтах з працевлаштування. Основним функціоналом системи є виявлення вакансій, що знаходяться в певному радіусі від певної точки на карті, враховуючи географічні координати вакансій та користувачів

У зв'язку з цим, метою даного проєкту є розробка сервісу для онлайн-пошуку вакансій з можливістю задавати точку на карті та відстань від неї для фільтрації вакансій. Це дозволить користувачам знаходити робочі місця в певному радіусі від їх поточного місцезнаходження або вибраної точки на мапі.

У результаті реалізації проєкту очікується створення функціонального та ефективного сервісу для пошуку вакансій онлайн з можливістю фільтрації за географічними критеріями. Користувачі зможуть шукати вакансії, задавати точки на карті та відстані від них, а отримані результати будуть представлені у зручному форматі.

					ІА91.110БАК.003 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Характеристика сфери працевлаштування

Україна має різноманітний ринок праці з різними секторами і галузями. Основні галузі, де можна знайти роботу, включають інформаційні технології, фінансовий сектор, сільське господарство, легку промисловість, будівництво, туризм і послуги.

У секторі інформаційних технологій (ІТ) в Україні спостерігається значний розвиток. Крім того, в Україні постійно з'являються нові стартапи та ІТ-компанії, що створює додаткові можливості для працевлаштування. Україна дійсно відома своїми ІТ-компаніями та програмістами, які надають послуги на міжнародному рівні. Багато українських ІТ-фахівців працюють на проєкти замовників з-за кордону, включаючи США, Європу та інші країни. Крім того, українські ІТ-компанії активно залучаються до розроблення настільного програмного забезпечення, веб-розробки, мобільних додатків та інших інноваційних проєктів.

Існують деякі зовнішні джерела та рейтинги, такі як Global Services Location Index (GSLI), які оцінюють країни за їх потенціалом у сфері ІТ-послуг. У минулому Україна була відзначена в таких рейтингах як надійний постачальник ІТ-послуг [1].

Фінансовий сектор в Україні є важливою складовою економіки країни і пропонує різноманітні можливості працевлаштування. Він включає банківські установи, страхові компанії, інвестиційні фонди, пенсійні фонди, фінансові консалтингові компанії та інші фінансові установи.

Банківський сектор є одним з найбільших і найбільш розвинених у сфері фінансів. В Україні працюють державні, комерційні та іноземні банки, що надають широкий спектр банківських послуг, включаючи розрахунково-касове обслуговування, кредитування, вклади, іпотеку, інвестиційні послуги

					ІА91.110БАК.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

та інше. У банківському секторі можна знайти робочі місця для фахівців з фінансів, економіки, банківського права, аналітики та інших спеціалізацій.

Страховий сектор також є важливою галуззю фінансового сектору в Україні. Він включає страхові компанії, які надають різноманітні види страхових послуг, таких як автомобільне страхування, медичне страхування, майнове страхування та інше. Робочі місця в страховому секторі доступні для фахівців з актуарної науки, страхового бізнесу, обробки даних та інших спеціалізацій.

Поряд з традиційними сегментами фінансового сектору, в Україні також спостерігається розвиток фінансових технологій (FinTech). Це охоплює компанії, які використовують інноваційні технології для надання фінансових послуг, наприклад, електронні платежі, онлайн-кредитування, криптовалюти та інше. Сектор FinTech надає нові можливості для працевлаштування в сфері фінансів та інформаційних технологій.

Сільське господарство залишається важливою галуззю українського економіки. Україна має потужний аграрний сектор, який займається вирощуванням зерна, овочів, фруктів, тваринництвом та іншими галузями сільськогосподарського виробництва. Це створює попит на робочу силу в аграрному секторі. Україна має широкий потенціал у сфері сільського господарства завдяки своїм родючим ґрунтам, кліматичним умовам та розташуванню.

В Україні вирощуються різні культури, включаючи зернові (пшениця, ячмінь, кукурудза), олійні культури (соняшник, соя), овочі (картопля, морква, цибуля) та фрукти (яблука, груші, виноград). Також розвинене тваринництво, зокрема виробництво молока, м'яса, яєць.

Україна є одним з провідних світових виробників зерна і олійних культур. Значна частина сільськогосподарської продукції експортується на зовнішні ринки, зокрема в країни Європи, Азії та Близького Сходу.

					ІА91.110БАК.003 ПЗ	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

Робочі місця в сільському господарстві доступні для різних професій, включаючи фермерів, агрономів, ветеринарів, техніків, машиністів, робітників урожаю та інших. Крім того, розвиток сільського туризму надає додаткові можливості для працевлаштування в сфері гостинності та послуг.

Важливо відзначити, що сільське господарство також стикається з викликами, такими як зміна клімату, розширення міських територій та низькі ціни на продукцію. Проте уряд та сільськогосподарські організації активно працюють над заходами для підтримки та розвитку цієї галузі.

Легка промисловість, будівництво, туризм і послуги також пропонують широкий спектр можливостей для працевлаштування. Легка промисловість включає текстильну, швейну та інші галузі, де можна знайти робочі місця. Будівництво також є активною галуззю, оскільки в Україні існує постійна потреба у зведенні нових житлових комплексів, інфраструктури та комерційних об'єктів. Туризм та послуги пов'язані з розширенням туристичної індустрії та гостинності.

1.2 Опис систем пошуку вакансій

У сучасному цифровому світі існує розмаїття платформ та додатків, спрямованих на допомогу у пошуку роботи, тому важливо розуміти їхні основні функції та особливості.

Одним з найпоширеніших видів ресурсів для знаходження вакансій є веб-платформи. Вони надають користувачам доступ до великої бази даних з робочими місцями, де можна проводити пошук за різними критеріями, такими як ключові слова, категорії робіт, регіони тощо. Веб-платформи часто пропонують можливість створення особистого профілю, де користувачі можуть внести свої дані, досвід роботи та навички, щоб привернути увагу потенційних роботодавців.

					ІА91.110БАК.003 ПЗ	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

Крім веб-платформ, існують мобільні додатки, які спрощують процес пошуку вакансій, дозволяючи швидко шукати роботу зі смартфона або планшета. Ці додатки мають зазвичай інтуїтивно зрозумілий інтерфейс та функціонал, що дозволяють користувачам встановлювати сповіщення про нові вакансії або зберігати обрані пропозиції для подальшого перегляду.

Також існують спеціалізовані ресурси для пошуку роботи, які зосереджені на конкретних галузях або професіях. Ці ресурси надають додаткові функції та фільтри, що допомагають користувачам знайти вакансії, які точно відповідають їхнім потребам та спеціалізації.

У системах пошуку роботи велику роль відіграють геоінформаційні системи (ГІС). Вони дозволяють користувачам здійснювати пошук робочих місць з урахуванням географічного положення. Завдяки ГІС, користувачі можуть обирати регіони, міста або навіть конкретні вулиці, де вони бажають знайти роботу. Це особливо корисно для людей, які шукають роботу в певному місці або планують змінити місце проживання.

1.3 Роль геоінформаційних систем в пошуку вакансій

Геоінформаційні системи відіграють важливу роль у процесі пошуку вакансій, особливо з урахуванням аспектів просторової аналітики та розташування робочих місць. Вони дозволяють збирати, аналізувати та візуалізувати географічні дані, що дає змогу компаніям та пошуковим системам пропонувати більш точні й зорієнтовані вакансії для кандидатів.

У пошуку вакансій ГІС можуть бути використані наступними способами:

1. Просторовий аналіз робочих місць: ГІС надають зручну можливість аналізувати розташування компаній та їх вакансій на карті. Використовуючи ГІС, можна відображати дані про робочі місця на мапі та проводити глибокий аналіз попиту та пропозиції праці в певній області. Кандидати отримують

					ІА91.110БАК.003 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

можливість швидко зорієнтуватися у вакансіях, доступних у їх регіоні або в областях, до яких вони готові переїхати. Це дозволяє збільшити ймовірність знаходження робочого місця, що відповідає їхнім потребам.

2. Пошук робочих місць за місцезнаходженням: Завдяки ГІС кандидати можуть знайти вакансії, які знаходяться неподалік від їх поточного місцезнаходження. Застосовуючи мобільні додатки або веб-платформи, які використовують ГІС, кандидати можуть шукати робочі місця в заданому радіусі від своєї домівки або поточного місця перебування. Це дає можливість зручно оцінити доступні варіанти праці, що є особливо корисним для тих, хто активно розглядає можливості в своєму найближчому оточенні.

3. Планування маршруту до співбесіди: ГІС забезпечують можливість кандидатам скласти оптимальний маршрут до місця співбесіди. Вони надають інформацію про транспортні маршрути, пробки на дорогах та інші фактори, які можуть вплинути на час подорожі. Це допомагає ефективно використовувати час та сприяє позитивному першому враженню, яке кандидат може зробити.

4. Розташування вакансій в конкретних географічних зонах: ГІС можуть допомагати компаніям знаходити кандидатів для роботи в певних регіонах або географічних зонах. Вони дозволяють визначити потенційних кандидатів, які вже проживають або працюють в цих областях. Це спрощує процес пошуку та залучення місцевих талантів, оскільки рекрутери можуть зосередитися на конкретних географічних зонах та привернути увагу кандидатів, які вже знайомі з цими місцевостями.

5. Візуалізація розподілу вакансій: ГІС дозволяють створювати графічні зображення, які ілюструють географічний розподіл вакансій. Це допомагає кандидатам та рекрутерам отримати ширший огляд ринку праці і зрозуміти, які області мають найбільшу концентрацію робочих місць у певних галузях. Візуалізація розподілу вакансій може бути корисною не тільки для кандидатів,

					ІА91.110БАК.003 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

але й для роботодавців, допомагаючи їм зосередитися на певних географічних областях або галузях, де попит на працю є найбільшим.

Отже, ГІС є потужним інструментом у сфері пошуку вакансій, що допомагає кандидатам більш ефективно орієнтуватися у просторі та знаходити відповідні працевлаштування, враховуючи їхнє розташування та інші фактори, що впливають на вибір робочого місця.

1.4 Огляд існуючих рішень

1.4.1 work.ua

Work.ua[2] – це один з найбільш популярних рекрутингових сайтів в Україні, який надає широкий спектр можливостей для ефективного пошуку роботи. Завдяки своєму зручному інтерфейсу та розширеним функціональним можливостям, сайт допомагає знайти вакансії, що відповідають інтересам і потребам кожного користувача.

Починаючи з головної сторінки, відвідувачі можуть скористатися полем пошуку, де можна ввести ключові слова, щоб знайти вакансії відповідно до своєї професійної спрямованості або галузі. Крім того, можна вказати місто, у якому вони бажають працювати, щоб знайти вакансії, доступні саме у цьому регіоні.

Одна із найбільших переваг сайту work.ua – це можливість використання фільтрів пошуку. Користувачі можуть обрати категорію вакансії, яка відповідає їхній професійній спрямованості, встановити діапазон заробітної плати та вибрати тип зайнятості, який їм підходить (повна зайнятість, часткова зайнятість, проектна робота тощо). Це дозволяє точніше визначити вимоги та отримати результати, що найкраще відповідають потребам.

Work.ua також пропонує зручну систему категорій вакансій. Користувачі мають можливість обрати галузь, яка їх цікавить, таку як ІТ, фінанси, маркетинг, медицина, освіта тощо, і легко знайти вакансії, пов'язані з

					ІА91.110БАК.003 ПЗ	Арк.
						11
Зм.	Лист	№ докум.	Підпис	Дата		

цими галузями. Це дозволяє швидко орієнтуватися в пошуку та знайти вакансії, які відповідають наявним навичкам та інтересам.

На кожній сторінці вакансії розміщено детальну інформацію про посаду, вимоги до кандидатів, обов'язки, умови праці та контактні дані роботодавця. Також є засоби для збереження вакансій для подальшого перегляду або відправлення їх іншим користувачам.

Після створення особистого кабінету на work.ua можна завантажити своє резюме та створити свій профіль. Це допомагає зберегти особисті дані та легко подавати заявки на вакансії. Крім того, є функціонал для налаштування сповіщень про нові вакансії, які відповідають заданим критеріям пошуку, що дозволяє користувачам бути в курсі актуальних можливостей.

Крім вищезазначених переваг, work.ua також надає корисні ресурси та поради щодо пошуку роботи, підготовки резюме та інтерв'ю. На сайті можна знайти статті, які надають цінну інформацію та допомагають підготуватися до успішного пошуку роботи.

Переваги Work.ua:

– Широкий вибір робочих місць: Work.ua пропонує велику кількість вакансій у різних сферах діяльності. На сайті розміщено пропозиції роботи в різних містах України, від крупних метрополій до невеликих населених пунктів.

– Зручний пошук: Сайт має простий та зручний інтерфейс, що дозволяє швидко знайти необхідну вакансію. Він надає різні фільтри, такі як місце розташування, галузь, рівень зарплати тощо, для точного визначення своїх вимог.

– Інформативність вакансій: Кожна вакансія на Work.ua зазвичай містить достатньо інформації про вимоги до кандидатів, умови праці, заробітну плату та контактні дані роботодавця. Це дозволяє користувачам заздалегідь оцінити, чи відповідає ця вакансія їхнім потребам та вимогам.

					ІА91.110БАК.003 ПЗ	Арк.
						12
Зм.	Лист	№ докум.	Підпис	Дата		

– Функціонал "Онлайн-резюме": Сай надає інструменти для створення свого онлайн-резюме, що дозволяє роботодавцям звертатися до шукачів безпосередньо з пропозиціями. Це зручно та ефективно для пошуку роботи.

Недоліки Work.ua:

– Конкуренція: За великою кількістю вакансій на сайті Work.ua йде значна кількість претендентів. Це означає, що можна зіткнутися зі значною конкуренцією при пошуку роботи, особливо у популярних галузях.

– Обмежена географія: Хоча сайт пропонує вакансії у різних містах України, деякі регіони можуть бути менше представлені. Якщо необхідно знайти роботу у меншому місті або селі, можливо, буде складніше знайти відповідні пропозиції.

– Відсутність персоналізованих рекомендацій: Work.ua не надає персоналізованих рекомендацій вакансій на основі даних профілю. Користувачам необхідно самим шукати та відбирати відповідні пропозиції, що може зайняти багато часу та зусиль.

– Платні послуги: Деякі додаткові функції на Work.ua, наприклад, виділення резюме або розміщення преміум-вакансій, є платними. Це може становити додаткові витрати для користувачів.

1.4.2 robota.ua

Robota.ua[3] пропонує широкий вибір вакансій у різних галузях та сферах діяльності. Користувачі можуть використовувати різні критерії пошуку, такі як ключові слова, назви посад, категорії роботи, міста та заробітна плата. Це дозволяє швидко знайти вакансії, які відповідають конкретним потребам користувача. Сайт групує вакансії за різними категоріями, що полегшує пошук роботи у конкретній галузі або сфері діяльності. Категорії можуть включати ІТ, фінанси, маркетинг, HR, логістику

					ІА91.110БАК.003 ПЗ	Арк.
						13
Зм.	Лист	№ докум.	Підпис	Дата		

та інші. Користувачі можуть вибрати відповідну категорію та отримати список вакансій, пов'язаних з нею.

Сайт також дозволяє користувачам обирати конкретне місто, в якому вони шукають роботу. Це особливо корисно для тих, хто хоче знайти роботу в певному регіоні або зв'язатися з роботодавцями в конкретному місці. Одна з важливих можливостей сайту полягає в тому, що користувачі можуть встановити мінімальну та максимальну заробітну плату, яку вони очікують. Це допомагає звужити вибір вакансій до тих, що відповідають фінансовим вимогам користувача.

Robota.ua надає додаткові фільтри, що дозволяють користувачам точніше налаштувати свій пошук. Вони можуть фільтрувати вакансії за типом зайнятості (повна зайнятість, неповна зайнятість, фріланс), рівнем досвіду, мовами та необхідними навичками. Кожна вакансія на сайті має детальний опис, включаючи інформацію про компанію, її профіль та контактні дані. Це допомагає користувачам отримати більш повну картину про робоче місце та компанію, зокрема її діяльність, репутацію та можливості.

Robota.ua також пропонує зручний кабінет користувача, де можна зберігати улюблені вакансії, стежити за статусом заявок та отримувати сповіщення про нові робочі місця, які відповідають заданим критеріям. На сайті також доступна інформація про тенденції на ринку праці, корисні поради щодо пошуку роботи, підготовки резюме та співбесіди.

Переваги сайту robota.ua:

– Широкий вибір вакансій: Сайт має велику базу робочих місць різних галузей та рівнів складності. Це дає можливість користувачам знайти роботу, що відповідає їхнім потребам та професійним навичкам.

– Регіональний фокус: Robota.ua фокусується на робочих місцях в Україні. Це дозволяє користувачам зосередитися на вакансіях, доступних у їхньому регіоні, що полегшує пошук роботи в конкретній локації.

					ІА91.110БАК.003 ПЗ	Арк.
						14
Зм.	Лист	№ докум.	Підпис	Дата		

– Корисні функції фільтрації: Сайт надає можливість використовувати різні фільтри для пошуку, такі як галузь, регіон, рівень зарплати тощо. Це допомагає користувачам зіставляти свої вимоги з доступними вакансіями та знаходити найбільш відповідні пропозиції.

– Персоналізовані рекомендації: Robota.ua використовує алгоритми рекомендацій, щоб підбирати вакансії, які можуть зацікавити користувачів на основі їхнього профілю та пошукових запитів. Це допомагає зекономити час та зусилля при пошуку роботи.

Недоліки сайту robota.ua:

– Конкуренція: Завдяки популярності сайту, на деякі вакансії може бути велика кількість кандидатів, що призводить до підвищеної конкуренції між претендентами. Це може зробити пошук роботи складнішим для деяких користувачів.

– Платні послуги: Деякі функції та можливості сайту robota.ua доступні лише за плату. Наприклад, доступ до розширеного пошуку або розміщення преміум-оголошень може вимагати підписки або оплати. Це може бути недоліком для користувачів, які не бажають або не можуть витратити додаткові кошти.

– Відсутність взаємодії з роботодавцями: Сайт robota.ua пропонує платформу для розміщення вакансій, але він не забезпечує прямої взаємодії між кандидатами та роботодавцями. Це може ускладнити процес спілкування та отримання додаткової інформації про вакансії.

– Обмежена база вакансій: Незважаючи на широкий вибір вакансій, база robota.ua може бути обмеженою порівняно з іншими великими міжнародними сайтами. Це може призвести до того, що деякі робочі місця не будуть представлені на сайті, особливо якщо вони пов'язані з іноземними компаніями або міжнародними проектами.

					ІА91.110БАК.003 ПЗ	Арк.
						15
Зм.	Лист	№ докум.	Підпис	Дата		

Висновки до розділу

У даному розділі було проведено детальний аналіз предметної області, пов'язаної з системою пошуку вакансій з використанням геоінформаційних систем. Була вивчена сфера працевлаштування, включаючи основні проблеми і вимоги до пошуку вакансій.

Досліджено різні системи пошуку вакансій і встановлено, що вони надають широкий спектр можливостей для працевлаштування, але часто не враховують географічний аспект. З'ясовано, що використання геоінформаційних систем може значно полегшити і покращити процес пошуку вакансій, дозволяючи користувачам швидко знаходити роботу в конкретному регіоні або місцезнаходженні, що є важливим фактором для багатьох кандидатів.

Також були розглянуті існуючі рішення, зокрема work ua та robota ua, які успішно впроваджують геоінформаційні системи у своїх платформах для пошуку вакансій. Вони надають користувачам можливість швидко знаходити вакансії в обраному регіоні та отримувати актуальну інформацію про робочі місця.

Отже, можна зробити висновок, що використання геоінформаційних систем у системі пошуку вакансій має великий потенціал для поліпшення процесу працевлаштування. Розглянуті існуючі рішення підтверджують цю можливість і свідчать про успішне використання геоінформаційних систем у цій сфері. Далі в роботі будуть розглянуті конкретні аспекти розробки системи пошуку вакансій з використанням геоінформаційних систем з метою покращення процесу працевлаштування.

					ІА91.110БАК.003 ПЗ	Арк.
						16
Зм.	Лист	№ докум.	Підпис	Дата		

2 ОПИС ПРОЦЕСУ ПРОЄКТУВАННЯ

2.1 Вимоги до програмного забезпечення

Для успішного аналізу та вибору програмних засобів реалізації поставленої задачі необхідно сформулювати перелік з основних вимог до фінального програмного забезпечення. Правильно та повноцінно сформульовані вимоги також дозволять краще досягнути необхідний функціонал програми, зменшити можливі ризики та визначити обсяг проєкту.

2.1.1 Функціональні вимоги

– Пошук вакансій: Користувачі повинні мати можливість шукати вакансії з різних джерел (компанії, рекрутингові агенції тощо). Система повинна забезпечувати можливість фільтрації вакансій за різними критеріями, включаючи географічні (відстань від заданої точки на карті).

– Відображення результатів: Результати пошуку повинні бути відображені у зручному для користувача форматі, який містить основну інформацію про вакансії (назва, опис, компанія, заробітна плата тощо). Користувачі повинні мати можливість переглядати детальну інформацію про кожну вакансію, включаючи контактні дані роботодавця.

– Відгук: Користувачі повинні мати можливість відгукнутися на вакансії, які їх зацікавили. Програма повинна надавати можливість зв'язатися з роботодавцями за наданими ними засобами, такими як мобільний телефон, електронна пошта, соціальні мережі тощо.

					IA91.110BAK.003 ПЗ	Арк.
						17
Зм.	Лист	№ докум.	Підпис	Дата		

2.1.2 Нефункціональні вимоги

- Інтерфейс: Система повинна мати інтуїтивно зрозумілий та зручний для використання інтерфейс. Користувачам повинно бути зручно шукати вакансії, задавати точки на карті та радіус пошуку.
- Ефективність: Система повинна працювати швидко та ефективно, забезпечуючи миттєву відповідь на запити користувачів.
- Безпека: Система повинна забезпечувати безпеку особистих даних користувачів та унеможлилювати несанкціонований доступ до них.
- Масштабованість: Система повинна бути готовою до масштабування та здатною працювати з великою кількістю користувачів та вакансій

2.2 Огляд обраних технологій та бібліотек

2.2.1 Геоінформаційна система QGIS

Основної механікою створюваного сервісу є можливість точного визначення координат на карті та відстані від заданої точки. Для вирішення цієї задачі було вирішено використати пакет геоінформаційного програмного забезпечення QGIS. Це програмне забезпечення надає гнучкі можливості для аналізу геоінформаційних даних, включаючи необхідний функціонал для розрахунку відстані між точками на Землі за заданими координатами. QGIS має детальну та повну документацію, а також широку спільноту користувачів та розробників, що сприяє легкому ознайомленню з програмним забезпеченням. Крім цього, QGIS постачається з бібліотекою Python, що дозволяє використовувати його алгоритми як бібліотеку у зовнішньому програмному забезпеченні, не потребуючи інтерфейсу QGIS.

QGIS є потужною та безкоштовною географічною інформаційною системою з відкритим вихідним кодом, що дозволяє безкоштовно використовувати, змінювати і поширювати його. QGIS є кросплатформеним і доступним для Windows, macOS та Linux. Програма має широкий спектр

					IA91.110BAK.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

функцій, що дозволяють створювати, редагувати, візуалізувати та аналізувати геопросторові дані. Вона також підтримує різні формати даних, включаючи векторні та растрові шари.

QGIS має активну спільноту розробників, яка постійно розширює функціонал програми. Користувачі можуть використовувати плагіни для додавання нових можливостей та функцій. Інтерфейс користувача QGIS є інтуїтивно зрозумілим та простим у використанні, а також має багато додаткових функцій, що сприяють зручній роботі з геоданими.

QGIS дотримується геопросторових стандартів, що дозволяє безпроблемний обмін даними з іншими географічними інформаційними системами та сервісами. Крім того, QGIS має велику спільноту користувачів, яка готова надати допомогу та поділитися знаннями. Це створює можливість знайти відповіді на питання та отримати підтримку при вирішенні проблем.

Загалом, QGIS є потужним інструментом для роботи з геопросторовими даними, який поєднує в собі багато функціональності з простим використанням та підтримкою активної спільноти користувачів. Він є популярним у наукових, професійних та любительських геопросторових галузях по всьому світу.

2.2.2 Мова програмування Python

Python є високорівневою інтерпретованою мовою програмування, яка має широке застосування в різних сферах розроблення програмного забезпечення. Однією з основних переваг Python є його простота та легкість вивчення. Синтаксис мови Python простий і зрозумілий, що дозволяє новачкам швидко розпочати програмувати і розвиватися.

Python також відомий своєю читабельністю коду. За допомогою відступів і чітких конструкцій, код на Python стає легким для розуміння і

підтримки. Це робить мову особливо привабливою для командної розробки, де кілька програмістів можуть спільно працювати над проєктом.

Ще одна перевага Python – широка бібліотека і модульний підхід до розробки. Велика кількість сторонніх бібліотек робить Python потужним інструментом для розв'язання різноманітних задач. Наприклад, NumPy і pandas дозволяють працювати з числовими даними та аналізувати дані, а Django та Flask – фреймворки для веб-розробки. Це дозволяє розробникам ефективно використовувати наявні інструменти і прискорює процес розробки.

Інтерпретованість Python також є перевагою. Код Python виконується рядок за рядком, що дозволяє розробникам швидко перевіряти результати своєї роботи та виправляти помилки на льоту, не витрачаючи додатковий час на компіляцію проєкту. Крім того, Python має широку підтримку платформ, включаючи Windows, macOS і різні дистрибутиви Linux, що робить його універсальним і доступним для використання на різних операційних системах.

Python також відомий своєю великою активною спільнотою розробників. У випадку, якщо виникає питання або проблема, шанси великі, що вже існує рішення або бібліотека, яка може допомогти. Спільнота Python активно розвивається, оновлюючи мову та розробляючи нові інструменти, що підтримує і популяризує використання Python у світі програмування.

Не можна забувати і про широке застосування Python в науці, особливо в областях як машинного навчання та аналізу даних. Бібліотеки, такі як TensorFlow і PyTorch, роблять Python популярним вибором для розробки моделей машинного навчання і нейромереж[4].

Інтеграція геоінформаційної системи QGIS з Python стала вирішальною у виборі мови програмування для проєкту, оскільки як програмний інтерфейс QGIS, так і Qt, бібліотека для користувацького інтерфейсу, розроблена з використанням мови C++ і використана для створення інтерфейсу QGIS, мають мовні прив'язки до Python.

2.2.3 Google Maps API

Google Maps API – це набір програмних інструментів, які надаються Google для розробників з метою інтеграції картографічних функцій та сервісів Google Maps у власні додатки. Цей API відкриває доступ до різноманітних можливостей, таких як відображення карт, отримання даних про місцезнаходження, розрахунок маршрутів, пошук підприємств та багато іншого.

Однією з основних переваг Google Maps API є його широкий функціонал. Він надає розробникам доступ до всіх популярних функцій Google Maps, таких як відображення карт, нанесення маркерів, створення полігонів та ліній, а також розрахунок маршрутів. Це дозволяє розробникам створювати різноманітні картографічні додатки, від простих до складних.

Іншою важливою перевагою Google Maps API є його надійність та швидкість. Google має потужну серверну інфраструктуру, яка забезпечує швидкий доступ до картографічних даних і маршрутів. Крім того, Google постійно оновлює свої дані, що забезпечує точність та актуальність інформації, яка надається розробникам через їх API.

Ще одна перевага сервісу полягає в його простоті використання. Інтеграція з Google Maps API вимагає від розробника всього кількох кроків, таких як отримання API-ключа та включення необхідних бібліотек у свій проєкт. Документація Google Maps API є детальною та добре описаною, що полегшує розробку та налагодження.

Крім того, Google Maps API має широку підтримку та активне співтовариство розробників. Існує багато додаткових бібліотек та розширень, які розроблені спільнотою та доступні для використання з Google Maps API. Розробники можуть швидко знайти рішення для своїх конкретних потреб завдяки активним форумам та блогам, присвяченим Google Maps API.

					ІА91.110БАК.003 ПЗ	Арк.
						21
Зм.	Лист	№ докум.	Підпис	Дата		

Однією з цілей створення даного сервісу є зручність для користувача у використанні геоінформаційних засобів. Google Maps є чи не найбільш популярним сервісом для відображення місцезнаходження, пошуку маршрутів тощо. Тому було вирішено використовувати саме його в якості засобу відображення та обміну даними про координати.

Загалом, Google Maps API є потужним інструментом для розробки картографічних додатків. Він надає широкий функціонал, є надійним та швидким, простим у використанні та має підтримку активної спільноти розробників. Ці переваги роблять Google Maps API популярним вибором для багатьох розробників у порівнянні з альтернативними картографічними сервісами.

2.2.4 Бібліотека PyQt

Qt є потужною кросплатформовою бібліотекою розробки програмного забезпечення, яка надає набір інструментів та функціональностей для створення графічних інтерфейсів користувача (GUI), мережевих додатків та іншого програмного забезпечення. Вона розробляється компанією Qt Company та має велику спільноту розробників.

Однією з основних переваг Qt є її кросплатформовість. Бібліотека дозволяє розробляти програмне забезпечення, яке працює на різних операційних системах, включаючи Windows, macOS, Linux, Android та iOS, а також знаходиться в процесі реалізації WebAssembly[5, 6] як однієї з можливих платформ, відкриваючи шлях до веб-розробки з використанням бібліотеки та роблячи її максимально кросплатформною. Це забезпечує широку аудиторію користувачів та спрощує процес розробки, оскільки не потрібно створювати окремий код для кожної платформи.

Ще однією перевагою Qt є його повнота. Бібліотека містить в собі велику кількість готових компонентів і інструментів, які допомагають розробникам

					ІА91.110БАК.003 ПЗ	Арк.
						22
Зм.	Лист	№ докум.	Підпис	Дата		

створювати різноманітне програмне забезпечення. Наприклад, Qt надає можливості для роботи зі звуком, мультимедіа, базами даних, мережевими протоколами та багатьма іншими аспектами розробки програмного забезпечення.

Qt також має потужну систему подій та сигналів, яка дозволяє ефективно взаємодіяти з графічним інтерфейсом та обробляти користувацькі дії. Ця модель програмування робить розробку GUI простішою та більш зрозумілою для розробників.

Іншою важливою перевагою Qt є його спільнота розробників. Багато програмістів активно використовують Qt та допомагають один одному у спільній роботі. Це означає, що можна знайти велику кількість документації, прикладів коду, форумів і бібліотек, що полегшує розробку та вирішення проблем.

Qt є комерційною бібліотекою, але також доступна у вільно поширюваних версіях. Це дає розробникам можливість використовувати Qt безкоштовно, якщо їх проєкт також є вільним програмним забезпеченням або відкритим кодом.

Оскільки в проєкті використовується мова програмування Python, для створення користувацького інтерфейсу було вирішено використати бібліотеку PyQt, яка є так званими «мовними прив'язками» («language bindings») до бібліотеки Qt, виступаючи «обгорткою» над основною бібліотекою, написаною мовою C++. Як вже було сказано, QGIS використовує Qt у якості бібліотеки для користувацького інтерфейсу, а система типів двох бібліотек тісно пов'язана. Тому використання PyQt для інтерфейсу створюваного проєкту є природнім та логічним вибором. Також це є важливим для інтеграції з Google Maps, оскільки для повноцінного функціонування сервіс потребує JavaScript та браузерне середовище, для яких бібліотека Qt має спеціальні компоненти.

2.2.5 База даних PostgreSQL

У якості бази даних для сервісу використовується PostgreSQL. Це реляційна база даних, яка дає можливість організувати дані у вигляді таблиць та встановити зв'язки між ними. PostgreSQL є відкритою базою даних з активною спільнотою розробників. Її вихідний код відкритий, що дозволяє співпрацювати та вносити поліпшення. Така відкритість сприяє надійності та швидкому вирішенню проблем.

Підтримка ACID забезпечує надійність та цілісність даних. PostgreSQL гарантує атомарність, консистентність, ізолюваність та стійкість, що важливо для надійної обробки транзакцій.

PostgreSQL має широкий спектр розширень та модулів, що дозволяє розробникам додавати функціональність до бази даних. Є широка база розширень, таких як PostGIS для географічних даних або pgcrypto для шифрування.

Ця база даних має потужну систему кешування, оптимізатор запитів та індексацію, що забезпечує продуктивність для обробки великого обсягу даних.

PostgreSQL масштабована і може працювати як для невеликих, так і для великих проєктів. Вона підтримує реплікацію, кластеризацію та розподілені запити, що дозволяє розгортати базу даних на кількох серверах та обробляти великі обсяги трафіку.

База даних також має вбудовану підтримку JSON і географічних даних. Вона надає інструменти для зберігання, опрацювання та запиту даних у форматі JSON. Крім того, можна використовувати розширення PostGIS для роботи з географічними об'єктами.

PostgreSQL дозволяє розробникам створювати власні розширення та типи даних, розширюючи можливості бази даних під свої потреби.

Ця база даних підтримується на різних операційних системах, що дозволяє розгортати додатки на різних платформах без зайвих зусиль.

					ІА91.110БАК.003 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

Вона також має програмний інтерфейс з мовою Python, що робить її підходящим варіантом для проекту. База даних містить інформацію про отримані вакансії, а також потенційно міститиме дані користувачів та інформацію для організації пошуку вакансій за ключовими словами. База даних оновлюється автоматично сервером, який виконує запити до сайтів, перетворює отримані дані у форму, що підходить до збереження в базі даних та перевіряє, чи активні збережені вакансії та чи є нові.

2.2.6 Бібліотека BeautifulSoup4

Основним контентом застосунку є оголошення про пошук претендентів на відкриті вакансії для підприємств, бізнесів та приватних підприємців. На момент запуску застосунку він не матиме широкої аудиторії роботодавців, яка викладатиме свої оголошення на сервісі. Тому було вирішено використати технології скрейпінгу (scraping) та парсингу (parsing) вакансій, викладених на інших веб-сайтах. Важливо зазначити, що кожне оголошення в застосунку міститиме посилання на оригінальний пост з метою вираження поваги до ресурсу, з якого було отримано контент та для реалізації можливості відгуку на вакансію, яка зацікавила користувача.

Теоретично веб-скрейпінг – це збір даних з використанням будь-яких коштів, за винятком програм, взаємодіють із API. Зазвичай для цього пишуть автоматизовану програму, яка звертається до веб-серверу, запитує дані (як правило, у форматі HTML або в інших форматах веб-сторінок), а потім аналізує ці дані та витягує звідти корисну інформацію. На практиці веб-скрейпінг включає широкий спектр методів та технологій програмування, таких як аналіз даних, синтаксичний аналіз природних мов та інформаційна безпека [7].

BeautifulSoup – це бібліотека для парсингу HTML і XML даних в мові програмування Python. Вона надає простий і зручний спосіб витягувати

					ІА91.110БАК.003 ПЗ	Арк.
						25
Зм.	Лист	№ докум.	Підпис	Дата		

інформацію з веб-сторінок, використовуючи потужні засоби для навігації по структурі документів.

Однією з головних переваг Beautiful Soup є його простота використання. Бібліотека надає зрозумілі інтерфейси для пошуку, вилучення і модифікації елементів HTML і XML документів. Вона дозволяє швидко і легко виконувати різні операції з даними веб-сторінок, незалежно від їх складності.

Ще одна велика перевага Beautiful Soup полягає в його гнучкості. Розробники можуть використовувати різні методи пошуку і фільтрації елементів документа, включаючи пошук за тегами, класами, атрибутами та текстом. Бібліотека також має потужні інструменти для роботи з CSS-селекторами, що дозволяє точно вибирати необхідні елементи.

Іншою значною перевагою Beautiful Soup є його здатність працювати з пошкодженими або неправильно відформатованими даними. Він може обробляти HTML і XML документи, що не відповідають суворим стандартам, і намагається видобути корисну інформацію навіть з таких документів. Це робить його особливо корисним для парсингу різних джерел даних з Інтернету, де якість HTML-коду може бути неоднорідною.

Крім того, Beautiful Soup є популярним вибором серед розробників Python завдяки великій активній спільноті користувачів. Це означає, що ви зможете знайти багато документації, прикладів і підтримки в Інтернеті, якщо у вас виникають питання або проблеми.

Висновки до розділу

У даному розділі були сформульовані функціональні та нефункціональні вимоги до програмного забезпечення, що реалізує систему пошуку вакансій з використанням геоінформаційних систем.

Функціональні вимоги визначають основні можливості системи, які включають пошук вакансій з урахуванням геолокації, відображення

					IA91.110BAK.003 ПЗ	Арк.
						26
Зм.	Лист	№ докум.	Підпис	Дата		

результатів на карті, фільтрацію вакансій за різними критеріями, додавання і редагування вакансій, збереження обраних вакансій, тощо.

Нефункціональні вимоги визначають якісні аспекти системи, такі як продуктивність, надійність, безпека, зручність інтерфейсу, масштабованість тощо.

З метою реалізації системи пошуку вакансій з використанням геоінформаційних систем було обрано такі бібліотеки та інструменти:

- QGIS – для роботи з геоінформаційними даними та картографією;
- Python – мову програмування для розробки програмного забезпечення;
- Google Maps API – для отримання геолокаційних даних та відображення на карті;
- PyQt – для розробки графічного інтерфейсу користувача;
- PostgreSQL – для збереження даних про вакансії та користувачів;
- BeautifulSoup4 – для парсингу веб-сторінок з вакансіями.

Таким чином, розділ відображає сформульовані вимоги до функціональності та якості програмного забезпечення, а також обрані бібліотеки та інструменти для реалізації системи пошуку вакансій з використанням геоінформаційних систем.

					ІА91.110БАК.003 ПЗ	Арк.
						27
Зм.	Лист	№ докум.	Підпис	Дата		

3 ОГЛЯД ЗАСТОСУНКУ

Застосунок складається з двох основних частин: клієнтської, яка являє собою настільний застосунок, задача якого надати зручний користувацький інтерфейс та основний функціонал застосунку, такий як отримання даних від користувачів, аналіз геоінформаційних даних та фільтрація вакансій за заданими параметрами, та серверу, відповідального за отримання та парсинг вакансій з різних сайтів, підключення і оновлення бази даних та передачі цих даних на клієнт.

3.1 Клієнтська частина

Клієнтська частина представлена вікном з переліком вакансій та інтерактивною картою Google Maps, на якій користувачі можуть задавати необхідні параметри для пошуку, такі як координати та відстань. Загальний вигляд користувацького інтерфейсу зображено на рисунку 1.

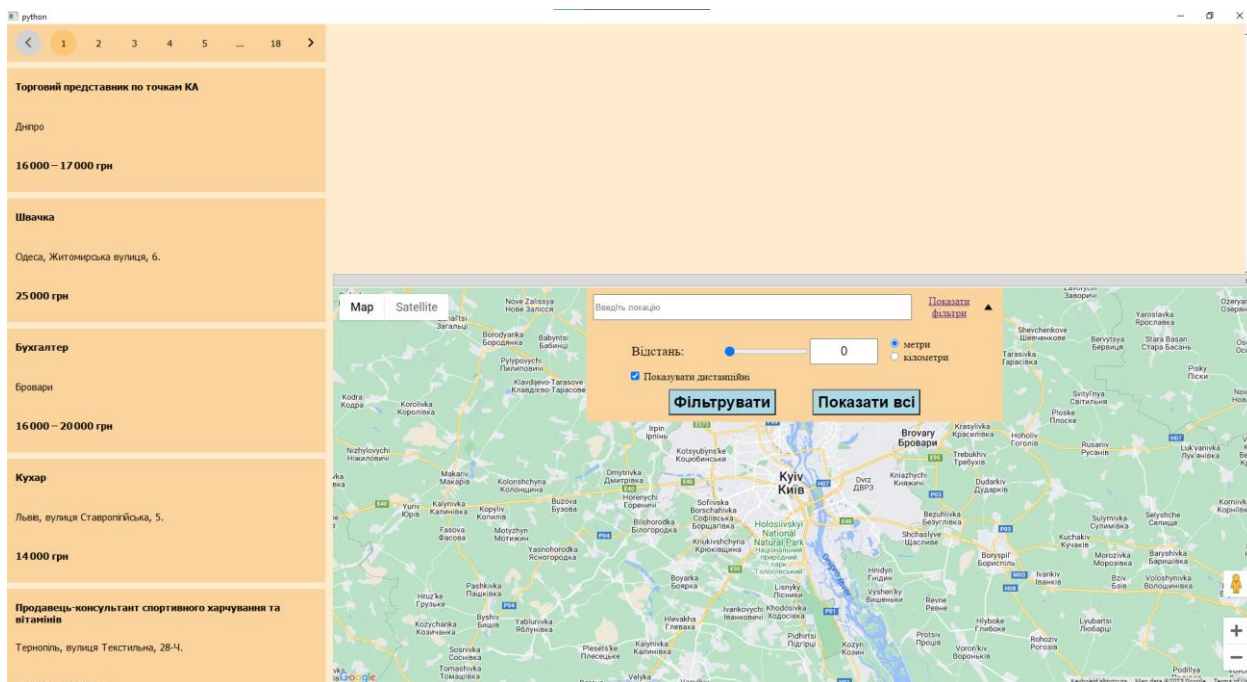


Рисунок 1 – Загальний вигляд користувацького інтерфейсу

Кожна вакансія в переліку містить загальну інформацію, таку як назва, компанія, адреса, заробітна платня тощо. У випадку, коли вакансія дистанційна, замість адреси висвічується напис «Дистанційно». Приклад відображення вакансії представлено на рисунку 2.

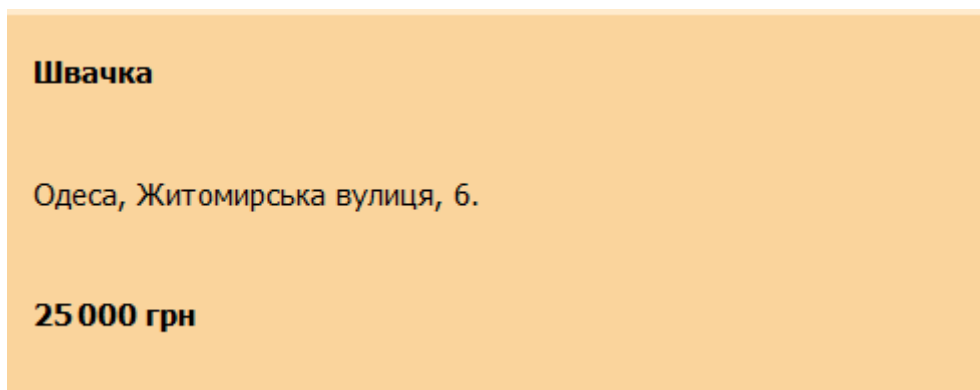


Рисунок 2 – Приклад відображення вакансії

При натисканні на обрану вакансію користувачеві відтворюється детальний опис, наданий роботодавцем (рисунок 3), який зазвичай включає в себе вимоги, робочі обов'язки, контактну інформацію та місце розташування.

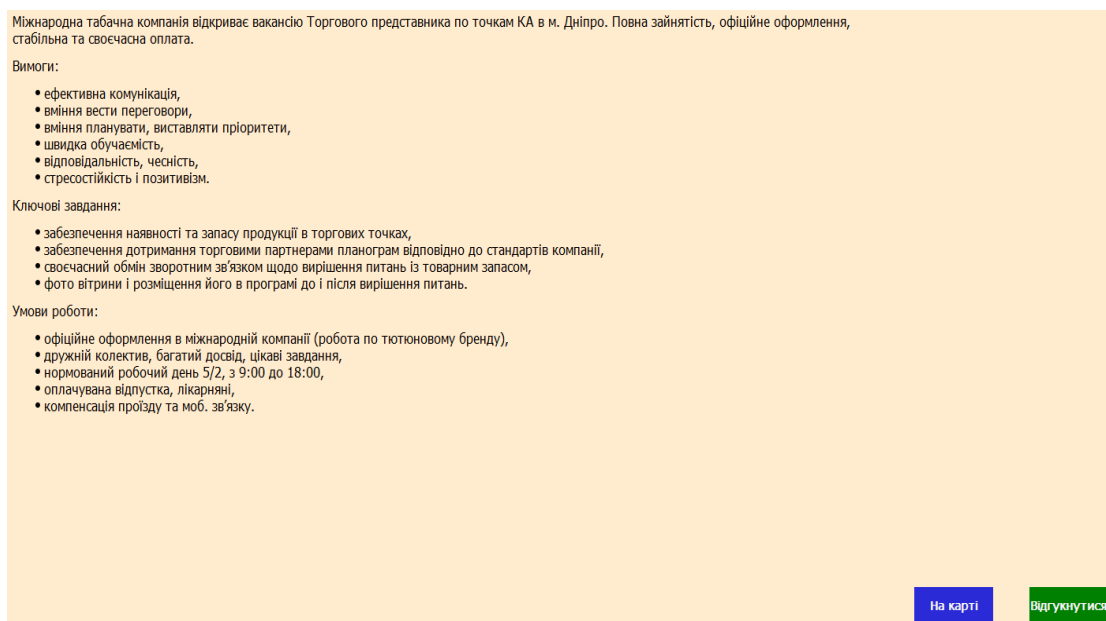


Рисунок 3 – Приклад опису вакансії

					ІА91.110БАК.003 ПЗ	Арк.
						29
Зм.	Лист	№ докум.	Підпис	Дата		

Якщо користувача зацікавила вакансія, він може натиснути кнопку «Відгукнутися». У цьому випадку його переадресує на оригінальний пост на відповідному сайті, де буде можливість відгукнутися відповідно до вимог сайту. Варто зазначити, що натискання на будь-яке посилання або кнопку «Відгукнутися» викличе діалогове вікно (рисунок 4), яке попередить користувача, що його буде перенаправлено на зовнішній сайт з використанням браузера за замовчуванням або відкрито відповідний застосунок на комп'ютері (наприклад, у випадку натискання на номер телефону). Таким чином можна запобігти випадковому натисканню на посилання, яке могло б призвести до непередбачуваних наслідків.

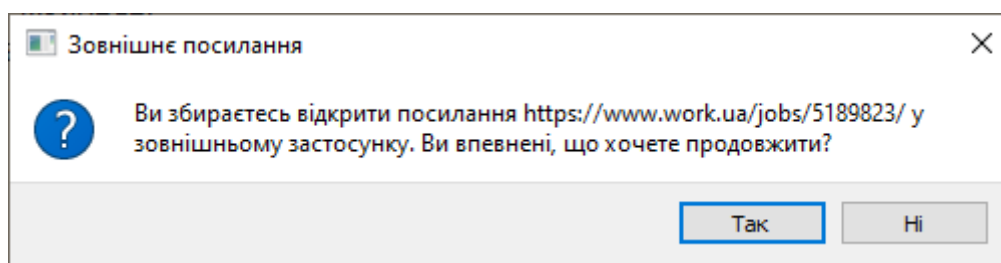
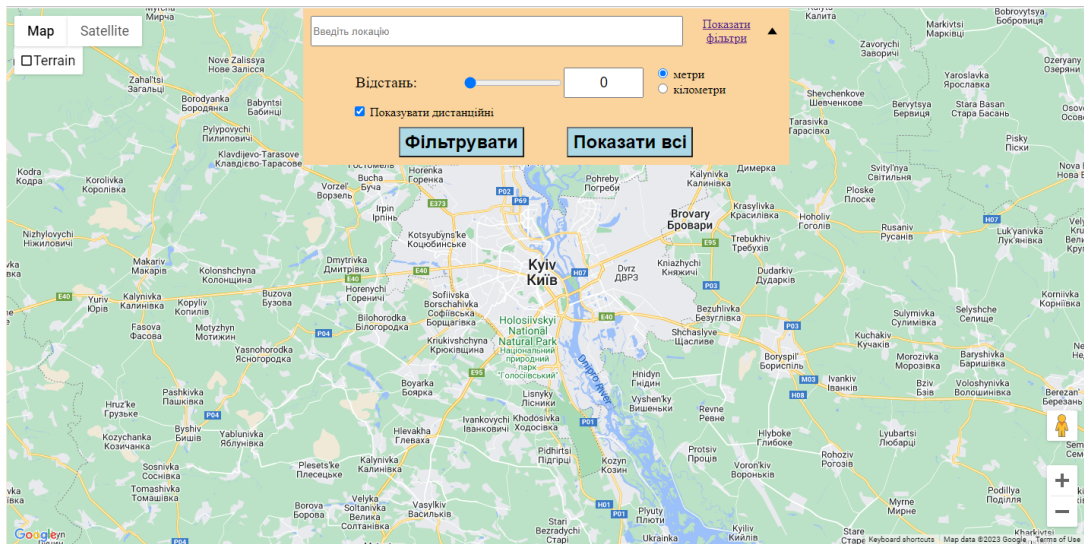


Рисунок 4 – Діалогове вікно

У нижній частині користувацького інтерфейсу знаходиться карта Google Maps, представлена на рисунку 5. Саме це вікно використовується для визначення координат та відстані, які потім використовуватимуться для фільтрування вакансій. Вікно з картою створено з використанням QDockWidget, елементу бібліотеки Qt, який надає можливість змінювати розмір вікна, переміщувати його всередині основного вікна програми та навіть повністю відокремлювати його від основного вікна. Таким чином користувачеві надається можливість підігнати карту під свої вимоги та розмістити її так, як їм буде зручно.



Рисунк 5 – Google Maps

Вікно з картою використовується як основний інструмент для фільтрації вакансій. У блоці у верхній частині вікна є текстове поле, до якого прив'язаний сервіс автозаповнення Google Maps Places Autocomplete. Він надає можливість проаналізувати введену локацію (місто, будь-яке специфічне місце, таке як парк, торговий центр тощо) та надати її координати на карті. Окрім цього, є можливість задати точку простим перетягуванням карти та натисканням у місці, яке слід вважати центром при фільтруванні.

Відстань від визначеного центру можна задавати за допомогою повзунка або текстового поля. Зміни в одному відобразяться в іншому. Це було зроблено для зручності користувача та для більшої гнучкості, оскільки це надає можливість задати відстань як точно (навіть числами з плаваючою комою) за необхідності, так і приблизно, спостерігаючи за колом на мапі, яке візуалізує задану відстань.

Відстань можна визначати як в метрах, так і в кілометрах. Для цього поруч з текстовим полем є перемикач. Максимальну відстань було обрано рівною 500 км, оскільки при центрі, розміщеному в географічному центрі України, та максимальній відстані у 500 км отримане коло діаметром 1000 км покриватиме майже всю територію країни[8].

Користувачі можуть також визначати, чи хочуть вони бачити дистанційні вакансії у відфільтрованих результатах. Для цього є перемикач «Показувати дистанційні». Якщо його увімкнено, користувачеві буде надано прелік дистанційних вакансій навіть якщо при фільтруванні за відстанню немає жодного результату (за умови наявності дистанційних вакансій в базі даних).

Коли всі необхідні параметри задано, користувач може натиснути кнопку «Фільтрувати» для передачі даних на аналіз. Результати фільтрування буде відображено у лівій частині користувацького інтерфейсу. Отримані вакансії буде відсортовано за відстанню від заданого центру в порядку зростання (перші – найближчі), при цьому дистанційні вакансії будуть вважатися такими, що мають найбільшу відстань від центру.

Для прикладу введемо в пошук «Київ» та оберемо необхідний варіант з випадаючого списку, задамо відстань 100 км, оберемо «Показувати дистанційні» та натиснемо «Фільтрувати». Вигляд карти після проведення цих операцій зображено на рисунку 6.

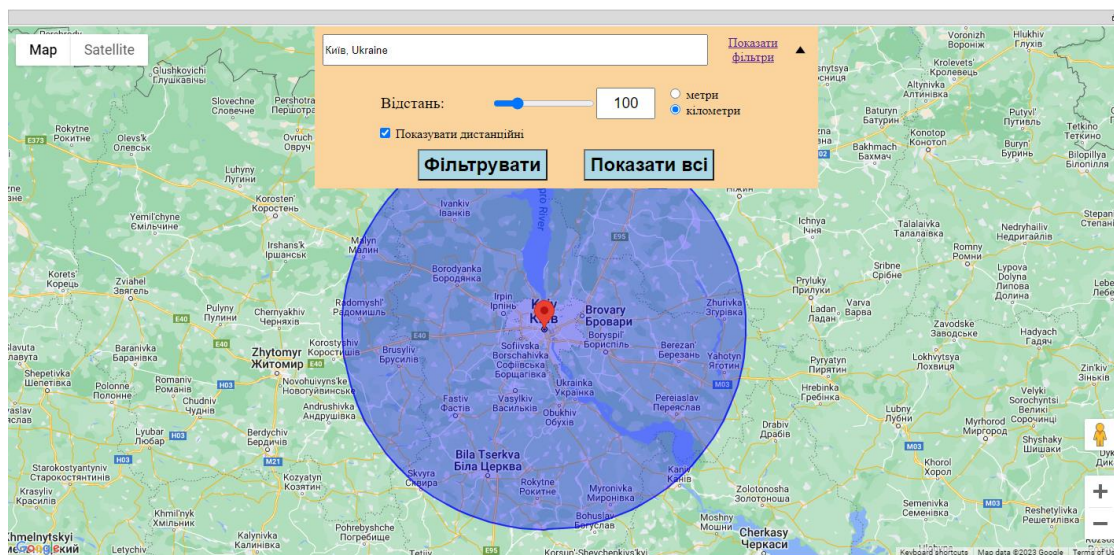


Рисунок 6 – Приклад завдання параметрів фільтрування

У результаті фільтрування отримаємо перелік вакансій, що знаходяться в заданому радіусі від центру. Картка вакансії тепер містить додаткове поле з зазначенням відстані (рисунок 7).



Рисунок 7 – Вакансія з відстанню

У даному випадку відстань 0.00 метрів, оскільки роботодавець зазначив просто «Київ» у адресі вакансії, тому Google Maps використовує координати центру Києва.

Вікно з картою також використовується для відображення місцезнаходження будь-якої обраної вакансії. Для цього користувачеві необхідно натиснути на кнопку «На карті», яку можна також побачити на рисунку 3. При натисканні на неї, маркер на карті перенесеться на відповідні координати, при цьому не скидаючи налаштування для фільтру, оскільки вони можуть використовуватися для подальших операцій.

Кнопка «Показати всі» використовується для відображення усіх доступних вакансій. Також, через особливості реалізації, вона може використовуватися для оновлення списку вакансій у разі, коли в базі даних відбулися зміни.

Кнопка «Показати фільтри» необхідна у випадку, коли користувачеві необхідно більше простору для огляду карти. При натисканні вона згортає усі контролюючі елементи, окрім поля пошуку, звільняючи місце.

3.2 Реалізація клієнтської частини

Як було сказано раніше, користувацький інтерфейс було створено за допомогою бібліотеки PyQt, яка є мовним прив'язками до бібліотеки Qt, написаної мовою C++. Основою даної бібліотеки є поняття віджету. Віджети є основними будівельними блоками графічного інтерфейсу користувача. Вони використовуються для створення різноманітних елементів у вікні програми, таких як кнопки, поля введення тексту, вікна, таблиці, меню тощо.

Віджети в Qt використовуються для відображення графічного інтерфейсу та обробки подій. Вони мають різноманітні властивості, які можна налаштовувати, такі як розмір, положення, колір, шрифт тощо. Віджети також можуть мати різні методи для виконання дій, наприклад, відповідати на натискання кнопки або зміни значення поля введення.

Qt надає велику кількість вбудованих віджетів для різних потреб. Наприклад, клас QPushButton для створення кнопки, клас QLineEdit для поля введення тексту, клас QTableView для таблиці тощо. Також можна створювати власні віджети шляхом успадкування від базових класів, таких як QWidget або QFrame, і визначення власної поведінки та зовнішнього вигляду.

Qt також має систему подій та сигналів, яка дозволяє віджетам взаємодіяти між собою та реагувати на події, такі як натискання кнопки або переміщення миші. Ця система використовується для обробки подій та передачі даних між віджетами або іншими об'єктами в Qt-додатку.

Сигнали і слоти – це спеціальні функції-члени, які визначаються в класах нащадках від QObject (базового класу для багатьох класів Qt). Сигнал виконує роль "видавця" (emitter), який видає сигнал про виникнення певної події, а слот – роль "підписника" (receiver), який реагує на цей сигнал та виконує відповідну обробку [9].

Один об'єкт може мати кілька сигналів, а інший – кілька слотів. Через механізм підписки-повідомлення можна зв'язувати сигнали об'єктів зі слотами

інших об'єктів, встановлюючи відповідні зв'язки. При виникненні події, сигнал викликається, і всі підписані на нього слоти відповідають на цей сигнал і виконують свою функціональність.

Бібліотека PyQt дещо спрощує систему слотів та сигналів тим, що немає необхідності оголошувати функцію або метод слотом явно, як це відбувається в основній бібліотеці. У якості слоту може виступати будь-який метод або функція, в тому числі анонімна. Для надсилання сигналу об'єкт типу `pyqtSignal` має метод `emit`, а для підписки – `connect`, який приймає в якості параметра слот, який необхідно викликати у момент виникнення події.

Система сигналів та слотів також діє як засіб зв'язку між об'єктами, що знаходяться у різних потоках. Потокобезпечність (`thread-safety`) є одним з важливих аспектів бібліотеки Qt. Наприклад, неможливо додати віджет, створений в одному потоці, в якості дочірнього до віджету, створеного в іншому потоці. Сигнали дають можливість безпечно передавати дані між потоками.

Клас `QGISApplication` є основним класом для застосунку. Його основна задача полягає в тому, щоб правильно ініціалізувати застосунок та бібліотеку QGIS на початку виконання та правильно деініціалізувати при завершенні. Цей клас також містить екземпляр класу `MainWindow`, який грає роль основної частини користувацького інтерфейсу та контейнеру для усіх інших віджетів.

Ще однією функцією класу є запуск процесу оновлення бази даних вакансій. Для цього використовується стандартна бібліотека Python `asuncio`, необхідна для асинхронного виконання функцій. Ця функція є опціональною, оскільки можливо запуснути оновлення окремим процесом (наприклад, у Docker-контейнері або на віддаленому хостингу тощо), і керується параметром конструктору класу `startDatabaseServer` типу `bool`.

Клас `MainWindow` наслідується від класу бібліотеки `QMainWindow`. Цей клас визначає зовнішній вигляд користувацького інтерфейсу всього застосунку. Для цього в ньому широко використовуються об'єкти класів

QVBoxLayout та QHBoxLayout. Це стандартні класи бібліотеки, необхідні для організації дочірніх об'єктів віджету стовпцем або рядком відповідно. Об'єкти всередині цих контейнерів розміщуються відносно один одного відповідно до заданих правил, таких як вирівнювання (до будь-якого з країв, по центру), проміжок між елементами тощо. Контейнери також задають правила для зміни розмірів дочірніх елементів: чи повинні вони збільшувати або зменшувати свою ширину та/або висоту, коли змінюються розміри інших елементів всередині контейнера або розміри самого контейнера.

Для відображення опису вакансій використовується віджет QTextBrowser. Цей віджет є потужним інструментом для відображення великого обсягу тексту з можливістю навігації та взаємодії з текстом.

Одна з основних особливостей QTextBrowser – це його здатність відображати HTML-код, що дозволяє створювати багатофункціональні інтерфейси з використанням різних елементів форматування, таких як заголовки, списки, таблиці, посилання та багато іншого. Віджет підтримує багато стандартних тегів HTML та CSS, що дозволяє створювати зручні, читабельні інтерфейси. Це стає в нагоді, оскільки опис вакансії зберігається у базі даних у вигляді HTML-розмітки.

Для відображення переліку вакансій було створено кілька додаткових класів. Перший з них – JobListItem. Він являє собою обгортку над класом Job, який є POD для збереження інформації про вакансію, такої як адреса, назва, заробітна плата тощо, для відображення його екземплярів у користувацькому інтерфейсі. Клас визначає сигнал selected, який надсилається, коли користувач натискає на віджет лівою кнопкою миші. Також він використовує клас QFontMetrics та його метод elidedText для скорочення тексту назви вакансії у випадку, коли він надто довгий, та заміни надмірних символів трикрапкою.

Другий клас, JobList, – це клас, який містить в собі список з екземплярів класу JobListItem. Його основна задача – розбиття вакансій, що зберігаються у базі даних, на сторінки для зручного відображення частини вакансій у

інтерфейсі. Для цього він приймає в конструкторі `itemGenerator` – функцію з двома параметрами, відступ та кількість елементів, яка повертає екземпляр класу `Job`. Ця функція є так званим генератором – функцією, яка здатна перериватися у процесі виконання, запам’ятовувати місце, на якому вона перервалася, та продовжувати виконання з цього місця при наступному виклику функції[10]. Для переривання виконання використовується ключове слово `yield`. За ключовим словом також може бути значення, яке повернеться як проміжковий результат виконання функції. Таким чином немає необхідності створювати додатковий список, до якого потім додавати елементи, та повертати його. Натомість, кожне значення повертається напямучу в момент виконання команди `yield`.

Другою функцією даного класу є зміна зовнішнього вигляду `JobListItem` залежно від того, чи обрано його.

Для розподілу даних на сторінки було створено клас `PageSelector`. Він відображає загальну кількість сторінок даних та визначає, яку сторінку має відображати `JobList`. Він містить в собі 7 кнопок для відображення номерів сторінок та 2 кнопки для переходу на наступну або попередню сторінку. Алгоритм, за яким вирішується, як відображаються доступні сторінки, зображено на кресленику ІА91.110БАК.003 Д1. Залежно від кількості доступних сторінок та поточної сторінки, визначається, чи активна кнопка переходу на наступну або попередню сторінку. При зміні відображуваної сторінки клас надсилає сигнал `pageChanged`, основним прослуховувачем якого є `JobList`. У відповідь на цей сигнал, він викликає функцію-генератор з необхідним відступом для отримання вакансій.

`JobList` також визначає сигнал `selectedItemChanged`, який надсилається, коли користувач обирає іншу вакансію зі списку. На цей сигнал реагує `QTextBrowser`, у якому відображається опис вакансії, та змінює інформацію, яку він відображає, на ту, що відповідає обраному оголошенню.

Google Maps – це, в першу чергу, веб-бібліотека. Вона потребує браузерного оточення та двигуна виконання JavaScript для функціонування. Звичайні віджети бібліотеки Qt не можуть цього забезпечити. Однак на цей випадок є плагін QtWebEngine. Цей плагін необхідно встановлювати додатково, оскільки він не входить в основну бібліотеку. Він базується на сучасному двигуні веб-рендерингу Chromium і дозволяє інтегрувати веб-сторінки із медіа-вмістом, веб-формами та багатьма іншими функціями в Qt-додатки.

QtWebEngine дозволяє відображати веб-сторінки безпосередньо в Qt-інтерфейсі, додаючи можливості браузера до додатків. Це означає, що розробники можуть відображати HTML-контент, використовуючи всю силу Qt для створення інтерактивних і користувальницьких віджетів навколо веб-сторінок.

Завдяки використанню двигуна веб-рендерингу Chromium, QtWebEngine забезпечує високу сумісність зі стандартами веб-розробки, такими як HTML5, CSS і JavaScript. Це дозволяє відображати складний медіа-вміст, включаючи відео, аудіо, анімацію та 3D-графіку, які можуть бути інтегровані в Qt-додатки.

Плагін QtWebEngine також надає можливості роботи з веб-сторінками, включаючи можливість виконувати JavaScript-код на стороні клієнта, взаємодію з формами, керування веб-контентом і обробку подій веб-сторінок. Це дає розробникам широкий спектр можливостей для взаємодії з веб-засобами в рамках своїх Qt-додатків.

Основна перевага використання QtWebEngine полягає в тому, що розробники можуть створювати веб-подібний контент, що інтегрується безпосередньо в їх додатки, замість використання зовнішнього веб-браузера. Це спрощує розробку, підтримку і розгортання додатків, оскільки всі компоненти, включаючи веб-вміст, управляються за допомогою Qt. Крім того, QtWebEngine забезпечує більш гнучкий і контрольований спосіб

					ІА91.110БАК.003 ПЗ	Арк.
						38
Зм.	Лист	№ докум.	Підпис	Дата		

відображення веб-сторінок, адаптуючись до вимог та потреб конкретного додатка.

Фактично, весь вміст вікна, в якому відображається карта, – це одна HTML-сторінка. На цю сторінку завантажуються бібліотеки Google Maps, необхідні для виконання функціоналу програми. Ці бібліотеки є обгортками над HTTP запитами, які надсилаються на сервери Google для виконання. Це робить взаємодію з API зручніше, оскільки немає необхідності вручну створювати HTTP запити з усіма необхідними параметрами. Кожен запит має містити в собі ключ API, який генерується в особистому кабінеті. Для зручності, він вводиться лише один раз при ініціалізації бібліотек і надалі використовується для виконання запитів автоматично.

Обмін даними між QtWebEngine та основним застосунком відбувається за допомогою технології WebSocket. Це протокол зв'язку, який дозволяє встановити постійне з'єднання між веб-сервером та клієнтом. Замість традиційної моделі запит-відповідь HTTP, WebSocket надає можливість передачі даних в обох напрямках між сервером і клієнтом у реальному часі.

WebSocket використовує одне з'єднання для багатьох повідомлень, що робить його більш ефективним порівняно з повторними запитами HTTP. Він базується на протоколі TCP, що забезпечує надійну передачу даних через Інтернет[11].

Процес встановлення з'єднання WebSocket складається з декількох етапів. Клієнт і сервер починають обмінюватися спеціальними заголовками HTTP, щоб встановити початкове з'єднання. Після успішного встановлення з'єднання вони переходять до бінарного режиму зв'язку, де можуть обмінюватися даними в режимі реального часу.

Одна з ключових особливостей WebSocket полягає в тому, що він підтримує події, які дозволяють серверу сповіщати клієнта про події, що відбуваються. Це реалізується шляхом використання механізму зворотного

					ІА91.110БАК.003 ПЗ	Арк.
						39
Зм.	Лист	№ докум.	Підпис	Дата		

виклику (callback mechanism). Клієнт може зареєструвати функцію-обробник подій, яка буде викликатися, коли сервер посилає нове повідомлення.

JavaScript має в стандартній бібліотеці тип WebSocket, який дозволяє встановити з'єднання з сервером, знаючи хост і порт. Він же надає і механізми для призначення функцій зворотнього зв'язку, які викликатимуться при певних подіях. Наприклад, функція onopen викликатиметься, коли з'єднання з сервером було успішно встановлено. У випадку створюваного застосунку, при встановленні з'єднання буде ініціалізовано код, необхідний для роботи карти, тобто без з'єднання вікно втрачає весь свій функціонал.

Для взаємодії з JavaScript кодом на стороні PyQt було створено допоміжний клас MapsHandler. Він запускає WebSocket сервер в новому потоці, щоб була можливість обробляти взаємодію з клієнтом асинхронно. Для створення серверу використовується бібліотека websocket-server. Це проста по своїй суті бібліотека, яка обгортає функціонал TCP-сервера зі стандартної бібліотеки Python та надає зручний програмний інтерфейс, схожий на інтерфейс WebSocket на стороні JavaScript. Ми також маємо визначити функцію для зворотнього зв'язку, яка викликатиметься при виникненні певних подій. Наприклад, при підключенні клієнта сервер перестає приймати нові підключення, оскільки в застосунку це з'єднання використовується виключно для взаємодії всередині застосунку і не приймає зовнішні з'єднання. Таким чином, у будь-який момент часу має існувати лише одне з'єднання, і це з'єднання має бути тим, що з'явилося першим при запуску застосунку.

Коли користувач натискає на кнопку «Фільтрувати», встановивши перед цим необхідні параметри, клієнт надсилає до серверу повідомлення у форматі JSON виду, зображеного на рисунку 8.

					ІА91.110БАК.003 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис	Дата		

```
{
  type: string,
  show_remote: boolean,
  lat: number,
  long: number,
  distance: number,
}
```

Рисунок 8 – Формат повідомлення для фільтрування

- type – тип повідомлення, у данному випадку це має бути «filter»,
- show_remote – значення, яке визначає, чи входять віддалені вакансії в результат пошуку,
- lat – широта центру,
- long – довгота центру,
- distance – відстань від центру, в межах якої виконується пошук.

При отриманні повідомлення від клієнту, викликається метод `on_message_received`. У цьому методі повідомлення, отримане у вигляді текстового рядку, за допомогою функції `loads` стандартного модулю Python `json` перетворюється на словник Python, у якому ключами виступають назви параметрів (`type`, `show_remote` тощо). Після цього перевіряється тип повідомлення і викликається відповідний метод, який оброблятиме запит. У випадку, коли тип повідомлення не визначено, піднімається виключення, яке повідомляє про помилку. Виключення завершує виконання програми, оскільки всі типи повідомлень визначені завчасно і якщо сталася помилка, то це ймовірно проблема в вихідному коді і програма не повинна далі виконуватися.

Для фільтрування вакансій використовується бібліотека QGIS. Для цього спершу створюється векторний шар `QgsVectorLayer`[12]. Векторні шари використовуються для зберігання геометричних об'єктів, таких як точки, лінії або полігони. Вони можуть містити також атрибутивні дані, що описують ці об'єкти, такі як назви, числові значення або дати.

Для створення векторного шару в QGIS можна використовувати різні джерела даних, наприклад, введення даних вручну, імпорт з файлів чи підключення до бази даних. Після створення шару можна виконувати різноманітні операції, такі як редагування геометрії, стилізація, аналіз просторових відношень та виконання запитів.

Один з потужних інструментів, що доступний для роботи з векторними шарами в QGIS, – це векторний аналіз. Він дозволяє виконувати різноманітні операції над векторними об'єктами, такі як об'єднання, перетин, видалення дублікатів та багато інших. Це дає можливість проводити складний аналіз географічних даних і отримувати цінну інформацію.

Ще однією важливою функцією векторних шарів у QGIS є їх візуалізація. QGIS надає широкі можливості для стилізації векторних шарів, включаючи зміну кольорів, типів ліній та заповнень, використання градієнтів та символів. Це дозволяє надати карте привабливий вигляд та зробити важливі елементи виділеними.

Крім того, векторні шари можна використовувати для виконання запитів та аналізу даних. QGIS надає можливість виконувати запити на основі атрибутів шару, що дозволяє відбирати об'єкти за певними критеріями. Також можна використовувати геопроектингові інструменти для виконання складних аналітичних завдань, таких як побудова буферних зон, обчислення відстаней та знаходження найкоротших маршрутів.

Створений векторний шар зберігатиме координати вакансій, збережені в базі даних. Для аналізу беруться всі вакансії з бази даних. Оскільки між викликами функції дані в базі можуть змінюватися, це необхідно робити щоразу, коли робиться аналіз відстані.

Відстань між точками на карті знаходиться за допомогою класу QgsDistanceArea. Для точності результатів необхідно задати еліпсоїд, тобто геодезичні параметри Землі, такі як радіус, за якими визначатиметься відстань. Є кілька варіантів еліпсоїдів, які можна обрати, але у нашому випадку

найкраще підійде WGS84. Система WGS84 (World Geodetic System 1984) є глобальною геодезичною системою координат, яка використовується для опису положення точок на поверхні Землі. Ця система була прийнята в 1984 році і є міжнародним стандартом для картографії, навігації, геодезії та супутникових систем позиціонування.

Одним з ключових аспектів системи WGS84 є її еліпсоїдальна модель Землі. Вона використовує еліпсоїд, що найкраще відповідає геометрії Землі, щоб наблизити її форму. Цей еліпсоїд використовується для створення трьох основних параметрів: величини осі, апоцентру (відстані від центру еліпсоїда до найближчої точки на поверхні) і ексцентриситета (відношення різниці між величиною осі і апоцентру до величини осі).

Система координат WGS84 використовує географічні координати для опису положення точок на Землі. Географічні координати включають широту і довготу, що вимірюються в градусах. Широта відлічується від екватора, де 0 градусів відповідає його площині, а довгота відлічується від осі, яка проходить через Гринвіч в Лондоні.

Окрім географічних координат, система WGS84 також визначає геоцентричні координати, що описують положення точок відносно центра мас Землі. Ці координати вимірюються в метрах і використовуються в аерокосмічних дослідженнях та супутникових системах позиціонування.

Додатково, система WGS84 надає модель геоїда, яка описує фізичну форму поверхні Землі. Геоїд є поверхнею, яка відображає теоретичний рівень моря на всій планеті. Враховуючи аномалії тяжіння та інші фактори, геоїд допомагає визначити точне положення точок на поверхні Землі.

QgsDistanceArea використовує зворотню формулу Вінсенті для обчислення відстані [13]. На відміну від методу гаверсинусів, який базується на прямому обчисленні відстані великого кола між двома точками на ідеально сферичній Землі, формули Вінсенті є ітераційним методом, який більш реалістично передбачає Землю як сплюснутий сфероїд. Компактність рішення

					ІА91.110БАК.003 ПЗ	Арк.
						43
Зм.	Лист	№ докум.	Підпис	Дата		

зумовлена використанням вкладених рівнянь для обчислення еліптичних членів і лише трьох тригонометричних функцій: синус, косинус і арктангенс. Вкладеність зменшує кількість операцій за участю зберігання та отримання проміжних результатів (зокрема при програмуванні мовою асемблера), зменшує довжину програми та час виконання, і мінімізує можливість недонаповнення пам'яті[14].

Отримана в результаті аналізу відстань порівнюється з заданою фільтром і вакансії, які знаходяться в окреслених межах додаються до списку, який потім сортується в порядку зростання.

Але JobList організований таким чином, щоб не зберігати всередині свого внутрішнього стану список з усіх вакансій. Він зберігає функцію-генератор, яка за вимогою надає рівно ту кількість вакансій, яку список може відобразити на одній сторінці. Тут стає в нагоді така техніка, як замикання. Вона є потужним концептом, який дозволяє передавати функції як аргументи, повертати їх як результати та зберігати їх у змінних. Сутність замикання полягає в тому, що функція зберігає оточення, включаючи змінні, доступ до яких мала на момент створення. Це дозволяє функції звертатися до цих змінних, навіть коли вона виконується в іншому контексті або пізніше в часі.

Замикання особливо корисне при роботі з анонімними функціями, такими як лямбда-функції. Вони можуть замикати змінні з оточення, де вони були створені, і використовувати їх у своєму внутрішньому коді. Це дає можливість передавати функції як аргументи в інші функції та зберігати їх у змінних для подальшого використання.

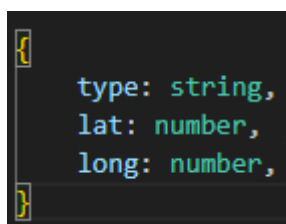
Одна з основних переваг замикання полягає в тому, що воно дозволяє створювати замикані функції, які зберігають свій стан і можуть продовжувати працювати з цим станом після завершення виконання зовнішньої функції. Це дуже корисно при роботі з асинхронним програмуванням, де функції можуть викликатися асинхронно та зберігати свій стан між викликами.

					ІА91.110БАК.003 ПЗ	Арк.
						44
Зм.	Лист	№ докум.	Підпис	Дата		

Замикання також дозволяє створювати імітацію приватних змінних у мовах програмування, що не підтримують цей концепт безпосередньо. Замикані функції можуть мати доступ до змінних, оголошених в зовнішньому контексті, але ці змінні не доступні зовнішній функції або ззовні.

Метод `get_filtered_generator` класу `MapsHandler` приймає в якості параметру список, а повертає нову функцію виду, необхідного для `JobList`, створену в оточенні, яке має доступ до цього списку. Таким чином, можна надати `JobList` посилання на отриману функцію і вона надасть необхідний інтерфейс для отримання вакансій зі списку. Даний підхід робить систему більш гнучкою, оскільки тепер джерелом вакансій може бути не лише база даних, а й будь-який список з вакансіями.

`WebSocket` надає можливість обміну в обох напрямках, як з клієнту на сервер, так і навпаки. Коли користувач натискає на кнопку «На карті» у вікні опису вакансії, сервер надсилає клієнту повідомлення виду, зображеного на рисунку 9.



```
{  
  type: string,  
  lat: number,  
  long: number,  
}
```

Рисунок 9 – Формат повідомлення для показу координат на карті

Тип у цьому випадку повинен бути «show». При отриманні такого повідомлення, код на боці JavaScript переміщує маркер на карті в задане місце.

Для налаштування зовнішнього вигляду отриманого застосунку можна використовувати класи та методи, визначені бібліотекою `Qt`, однак це не найзручніший спосіб. Альтернативно, `Qt` надає можливість використовувати `QSS` (`Qt Style Sheets`). Це мова стилів, яка використовується в бібліотеці `Qt` для опису зовнішнього вигляду графічного інтерфейсу користувача. `QSS` файли

містять набір правил стилів, які визначають вигляд різних елементів інтерфейсу, таких як кнопки, текстові поля, таблиці, вкладки тощо.

QSS базується на CSS, який використовується для стилізації елементів веб-інтерфейсів. Двигун QSS підтримує більшість функціоналу стандарту CSS2 та частково стандарт CSS3[15].

Завдяки QSS, розробники можуть відокремити вигляд інтерфейсу від його функціональності. Замість визначення стилів прямо в коді програми, їх можна описати в QSS файлах. Це робить код більш читабельним і легким для редагування.

QSS файли можна використовувати у багатьох різних контекстах, де використовується бібліотека Qt. Наприклад, вони широко використовуються в програмах на основі Qt Widgets, які використовують класи, такі як QWidget, QPushButton, QComboBox тощо. Прикладом може бути написання стилів для кнопок у QSS файлі і їх подальше використання в програмі.

Окрім того, QSS файли також використовуються в більш сучасній технології Qt Quick, яка базується на QML (Qt Meta-Object Language). У цьому контексті QSS використовується для стилізації різних QML елементів, таких як кнопки, текстові поля, фони тощо. Завдяки цьому можна забезпечити єдиний зовнішній вигляд для додатків, що використовують як Qt Widgets, так і Qt Quick.

Основними конструкціями в QSS файли є селектори та декларації стилів. Селектори визначають, до яких елементів інтерфейсу будуть застосовуватись певні стилі. Наприклад, селектор QPushButton визначає стилі для всіх кнопок. Декларації стилів визначають вигляд елементів, що підпадають під задані селектори. Наприклад, `background-color: red` визначає червоний фон для відповідних елементів.

3.3 Серверна частина

Серверна частина складається з двох елементів: менеджер бази даних та скрейпер-парсер вакансій. Як вже було сказано вище, на даний момент основний перелік вакансій складається з оголошень, отриманих зі сторонніх сайтів. Задача скрейпера-парсера отримати веб-сторінки з оголошеннями, виділити необхідну інформацію з них та перетворити на програмну модель, яку буде зручно зберігати в базі даних та відображати в застосунку. Варто зазначити, що для цього попередньо проводиться аналіз html-розмітки сайтів для визначення важливих компонентів та шляхів їх отримання. Це тягне за собою деякі обмеження, оскільки для стабільної роботи застосунку необхідно, щоб інтерфейс веб-сайту залишався незмінним. Це є ще однією з причин в далекоглядній перспективі відмовитися від цього варіанту реалізації та перейти повністю на власні оголошення, отримані від роботодавців через спеціальну форму. Алгоритм процесу скрейпінгу та парсингу представлено на кресленнику IA91.110БАК.003 Д2.

Менеджер бази даних, як видно з назви, відповідальний за встановлення та підтримку з'єднання з базою даних. Він тісно співпрацює зі скрейпером-парсером, отримуючи від нього дані та зберігаючи їх у базі даних. Він же відповідальний за отримання актуальних вакансій для відображення в користувацькому інтерфейсі. Менеджер представлений набором функцій, створених для виконання основних дій над базою даних, таких як додавання та видалення записів та отримання даних з таблиць.

База даних представлена двома сутностями: posts та locations. Діаграму відношень сутностей зображено на рисунку 10.

					IA91.110БАК.003 ПЗ	Арк.
						47
Зм.	Лист	№ докум.	Підпис	Дата		

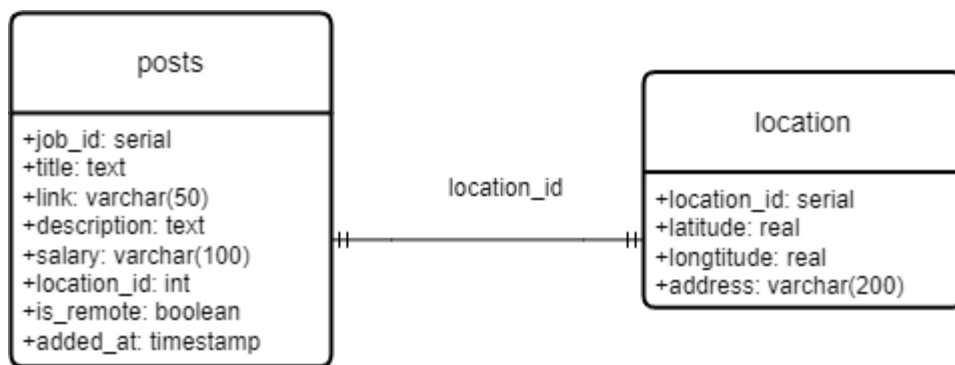


Рисунок 10 – Діаграма відношення сутностей

Таблиця 1. Атрибути сутності posts

job_id	int autoincrement	not null	primary key	Унікальний ідентифікатор
title	text	not null		Заголовок вакансії
link	varchar(50)	not null		Посилання на оригінальне оголошення
description	text	not null		HTML-розмітка, яка містить опис вакансії
salary	varchar(100)	nullable		Заробітна плата. Може бути відсутня. Також не завжди представлена числами
location_id	int	nullable	foreign key	Ідентифікатор географічної інформації

is_remote	boolean	not null (default false)		Показує, чи є вакансія віддаленою
added_at	timestamp	not null (default current_timestamp)		Час додавання вакансії до бази даних

Таблиця 2. Атрибути сутності location

location_id	int autoincrement	not null	primary key	Унікальний ідентифікатор
latitude	real	not null		Широта розміщення вакансії
longtitude	real	not null		Довгота розміщення вакансії
address	varchar(200)	not null		Текстова репрезентація адреси

Для взаємодії з базою даних використовується мова SQL (Structured Query Language). Це мова програмування, спеціально створена для керування базами даних. Вона є потужним інструментом, що дозволяє виконувати різноманітні операції з даними, такі як створення, зміна, видалення та отримання інформації з бази даних.

SQL має декларативний синтаксис, що означає, що користувач описує, яку інформацію потрібно отримати або яку дію потрібно виконати, а не як саме це зробити. Основною одиницею в SQL є запит, який може бути створений для вибірки даних, модифікації таблиць, створення бази даних та багатьох інших завдань.

Одним із ключових аспектів SQL є можливість виконувати реляційні операції над базами даних[16]. Запити можна виконувати до однієї таблиці або

комбінувати кілька таблиць для складніших операцій. SQL надає різноманітні конструкції, такі як SELECT, INSERT, UPDATE, DELETE, для виконання різних операцій над даними.

SELECT є однією з найбільш важливих конструкцій SQL, оскільки вона дозволяє отримати певні дані з таблиці або кількох таблиць. За допомогою SELECT можна вибирати конкретні стовпці, сортувати результати, застосовувати умови фільтрації, об'єднувати дані з різних таблиць за певними умовами та багато іншого.

INSERT використовується для додавання нових записів в таблицю бази даних. Можна вставити дані в один або декілька стовпців таблиці за один раз, вказавши значення для цих стовпців.

UPDATE використовується для зміни вже існуючих даних в таблиці. Можна оновлювати значення в одному або декількох стовпцях, а також застосовувати умови для обмеження оновлення лише певних рядків.

DELETE дозволяє видаляти рядки з таблиці бази даних. Можна видалити всі записи з таблиці або обмежити видалення лише певних рядків за допомогою умов.

SQL також надає можливість використовувати умовні оператори, такі як IF, CASE і IN, для більш складних логічних операцій. Вони необхідні для створення складних умов, комбінування їх за допомогою логічних операторів (AND, OR) та виконання різних дій залежно від умови.

Крім того, SQL має можливість групування даних, сортування, об'єднання таблиць за допомогою операторів GROUP BY, ORDER BY і JOIN відповідно. Ці оператори дозволяють створювати більш складні запити, які оброблюють і агрегують дані з різних джерел.

На рисунках 11 та 12 зображено SQL запити, необхідні для створення таблиць, описаних вище, для збереження даних.

```

cursor.execute("""CREATE TABLE IF NOT EXISTS locations (
    location_id serial PRIMARY KEY,
    latitude real NOT NULL,
    longitude real NOT NULL,
    place_id text NOT NULL,
    address varchar(200) NOT NULL
);""")

```

Рисунок 11 – SQL запит для створення таблиці locations

```

cursor.execute("""CREATE TABLE IF NOT EXISTS posts (
    job_id serial PRIMARY KEY,
    title text NOT NULL,
    link varchar(50) NOT NULL,
    description text NOT NULL,
    salary varchar(100),
    location_id int,
    is_remote boolean NOT NULL DEFAULT false,
    added_at timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
    CONSTRAINT fk_location FOREIGN KEY(location_id) REFERENCES locations(location_id)
);""")

```

Рисунок 12 – SQL запит для створення таблиці posts

Після отримання вакансій від парсеру менеджер бази даних перевіряє, чи вже існує дана вакансія в базі даних. Для цього виконується наступний запит:

```

query = sql.SQL("""SELECT CASE
    WHEN EXISTS (SELECT * FROM posts WHERE "link"={}) LIMIT 1) THEN 1
    ELSE 0
END""").format(sql.Literal(job.link))

```

Рисунок 13 – SQL запит для перевірки наявності оголошення в базі даних

Тут ми користуємося тим фактом, що кожне посилання на кожну вакансію унікальне. Таким чином, якщо база даних містить вакансію з заданим посиланням, менеджер не додаватиме її повторно. Наведений запит використовує таку властивість SQL, як підзапит (subquery), тобто результат виконання внутрішнього запиту потім використовується як параметр для

					IA91.110BAK.003 ПЗ	Арк.
						51
Зм.	Лист	№ докум.	Підпис	Дата		

зовнішнього. Це демонструє гнучкість мови SQL та можливість створювати достатньо комплексні запити.

Отримані дані про вакансію, однак, не містять географічних даних, таких як широта та довгота, лише текстову репрезентацію адреси. Для перетворення одного в інше знову звернемося до Google Maps API, зокрема його частиною Geocoding API, але тепер, коли ми на боці Python і не маємо доступу до бібліотек JavaScript, доведеться самотійно робити HTTP-запит.

Це можна легко зробити, використовуючи стандартний модуль Python requests. Для цього необхідно використати функцію requests.get, передавши в неї URL-адресу виду:

```
f"https://maps.googleapis.com/maps/api/geocode/json?address={address}&key={GOOGLE_API_KEY}"
```

Рисунок 14 – Формат адреси запиту для отримання координат

Тут address – текстова репрезентація адреси, GOOGLE_API_KEY – ключ, отриманий в особистому кабінеті, який до цього використовували для роботи з картою Google. Варто зауважити, що перед тим, як вставляти адресу в посилання, її необхідно відповідно закодувати: замінити символи, які мають спеціальне значення в URL, на послідовності вигляду "%xx", де "xx" представляє шістнадцятковий код символу. Наприклад, символ пробілу кодується як "%20", а символ косої риски ("/") – як "%2F". Це дозволяє ігнорувати спеціальне значення символів в URL-адресі та забезпечує правильну передачу даних через мережу.

Основні причини використання URL-кодування:

– Універсальність: URL-кодування дозволяє передавати будь-які символи через URL, включаючи ті, які мають спеціальне значення (наприклад, символи, що використовуються для розділення компонентів URL, які не можуть бути використані безпосередньо).

– Безпека: URL-кодування забезпечує безпечну передачу символів через мережу, оскільки забороняє використання спеціальних символів, які можуть бути сприйняті як команди або роздільники.

– Сумісність: URL-кодування дозволяє використовувати URL-адреси з різними типами символів, включаючи нелатинські символи, пробіли, знаки пунктуації та інші спеціальні символи.

– Достовірність: URL-кодування забезпечує коректне відображення символів у URL-адресах і уникнення помилок при їх обробці.

Серверна частина застосунку з певною періодичністю відправляє нові запити до сайтів з вакансіями для актуалізації інформації, збереженої у базі даних. Вона перевіряє, чи є нові оголошення, які ще не було збережено. При отриманні запиту від клієнтської частини, вона обирає необхідну кількість вакансій, які можна зручно і швидко відобразити в переліку. У цей же час, за допомогою додаткового запиту до веб-сайту, сервер перевіряє, чи актуальне ще посилання на оголошення. У разі, коли вакансії вже не існують з будь-яких причин, менеджер бази даних видаляє всю інформацію, пов'язану з цим оголошенням. Таким чином, застосунок зменшує кількість вакансій, які вже не актуальні і лише займають місце в базі даних. Але це не очищує її повністю і, більш того, деякі вакансії можуть ніколи не бути видалені через нестачу запитів, велику давність оголошення або значну кількість подібних оголошень. Для цього з заданою періодичністю сервер перевіряє всі вакансії в базі даних, видаляючи ті, що вже не активні.

Для виконання цих задач було створено дві основні функції: `update_database` та `clear_database` для оновлення та очищення бази даних відповідно. Обидві функції оголошено як асинхронні. Асинхронне програмування в Python з використанням бібліотеки `asyncio` дозволяє створювати ефективні та швидкодіючі програми, які можуть одночасно обробляти багато задач без блокування виконання. Основною ідеєю

асинхронності є використання нечекаючого (неблокуючого) вводу-виводу для уникнення затримок та ефективного використання ресурсів.

Центральним поняттям в `asuncio` є корутина. Корутина – це функція, яка може бути призупинена у визначених точках, дозволяючи іншим задачам виконуватись протягом цього часу. Призупинення відбувається за допомогою ключового слова `await`, яке вказує програмі на те, що потрібно зачекати, поки виконається певна операція.

Для створення асинхронного середовища у програмі використовується поняття циклу подій (`event loop`). Цикл подій – це механізм, який керує виконанням корутин та обробкою подій. У процесі роботи, коли корутина зустрічає ключове слово `await`, вона призупиняється і передає управління іншим корутинам. Цикл подій відстежує готовність корутин до продовження виконання і забезпечує їх безперервну обробку.

Одним з корисних понять в асинхронному програмуванні є "задача" (`task`). Задача – це об'єкт, який представляє виконання корутини у циклі подій. Задачі можуть бути створені з допомогою функції `asuncio.create_task()`, яка додає корутину до циклу подій та повертає об'єкт задачі.

`asuncio` також надає можливість створювати обробники подій (`event handlers`), які можуть бути призначені для обробки певних подій, таких як завершення вводу-виводу або таймери. Обробники подій дозволяють реагувати на події без блокування виконання програми.

Крім того, `asuncio` надає багато інших корисних функцій та класів, які полегшують роботу з асинхронним програмуванням. Наприклад, `asuncio.Lock` дозволяє створювати блокування для координування доступу до ресурсів, а `asuncio.Queue` надає безпечну спільну чергу для обміну даними між корутинами.

Оскільки ці дві функції викликаються з заданою періодичністю, вони можуть виконуватися одночасно на одному потоці, використовуючи на повну ресурси, надані їм операційною системою.

Як вже було сказано раніше, процес оновлення та видалення даних з бази не обов'язково розпочинати в тому самому процесі, що й основний клієнтський застосунок. Однією з особливостей мови Python є те, що файл з вихідним кодом може виступати одночасно і в якості модуля, тобто бібліотеки, і в якості виконуваного застосунку. Якщо передати файл серверу до інтерпретатора Python як основний, то він запустить оновлення бази даних в окремому процесі. Таким чином, серверна частина може працювати незалежно від клієнтської, окремим процесом або навіть на іншій машині в мережі.

3.4 Безпека та налаштування

У системі використовується кілька сервісів, такі як база даних та Google Maps API, які вимагають авторизації користувача для доступу до функціоналу. Google Maps потребує ключ API, PostgreSQL – назву бази даних, ім'я користувача та пароль. Усі ці дані не мають потрапити до стороннього користувача, оскільки можуть бути використані в протизаконних та шкідливих діях.

Одним зі шляхів забезпечення безпеки таких даних є використання змінних оточення (environmental variables). Ідея полягає в тому, щоб зберігати чутливі дані поза програмним кодом і завантажувати їх, коли це потрібно. Для цього часто використовують так звані «.env файли», тобто звичайні текстові файли з розширенням .env. У них розробники визначають змінні середовища під час розроблення програмного забезпечення і зчитують за допомогою таких бібліотек, як dotenv. Після цього доступ до них можна отримати, ніби це системні змінні оточення, тобто використовуючи модуль Python os.getenv.

При розгортанні застосунку на сервері, хостинги надають можливість визначити змінні оточення в налаштуваннях. Таким чином програма може отримувати ці дані так само, як робила це під час розроблення, не вимагаючи додаткових змін у визідному коді.

					ІА91.110БАК.003 ПЗ	Арк.
						55
Зм.	Лист	№ докум.	Підпис	Дата		

Файли та змінні оточення можна використовувати не лише для збереження чутливих даних, а й для параметрів системи, які можуть використовуватися для налаштувань. Таким чином достатньо змінити значення змінної в одному місці і всі залежні від цієї змінної системи продовжать далі функціонувати, не вимагаючи ручної зміни значення в кожному місці. Це надає системі гнучкості та робить внесення змін безболісним для розробника.

У проєкті використовується три файли оточення: один для клієнту, один для серверу і один спільний. Клієнтський визначає такі змінні, як `WEBSOCKET_HOST` та `WEBSOCKET_PORT`, необхідні для встановлення з'єднання між частиною застосунку, написаної на Python та картою, що використовує JavaScript. Серверний файл визначає змінні `DB_NAME`, `DB_USER`, `DB_PASS`, `DB_HOST` для встановлення з'єднання з базою даних, а також `DB_UPDATE_INTERVAL`, яка визначає, з якою періодичністю сервер перевірятиме наявність нових вакансій, та `DB_CLEAR_INTERVAL` для визначення періодичності, з якою сервер проходитиме через всю базу даних та видалятиме неактивні вакансії.

Висновки до розділу

У даному розділі було розглянуто детальніше функціонал та реалізацію клієнтської та серверної частин системи пошуку вакансій з використанням геоінформаційних систем.

Клієнтська частина системи була реалізована з використанням мови програмування Python та бібліотеки PyQt. Python, як мова програмування, забезпечує гнучкість та легкість в розробці програмного забезпечення. PyQt, надійна бібліотека для розробки графічного інтерфейсу користувача, дозволяє створювати інтуїтивно зрозумілі та привабливі інтерфейси. Клієнтська частина включає в себе відображення результатів пошуку вакансій на карті,

					ІА91.110БАК.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

що реалізовано за допомогою Google Maps API. Це дозволяє користувачеві візуально оцінити розташування вакансій та зручніше планувати свої подальші кроки.

Серверна частина системи була реалізована з використанням стандартних бібліотек Python та бібліотеки BeautifulSoup4. Стандартні бібліотеки Python надають широкий набір інструментів для обробки даних та мережевої взаємодії. BeautifulSoup4 є потужним інструментом для парсингу веб-сторінок, що дозволяє отримати необхідну інформацію про вакансії з різних джерел. Також для збереження та отримання даних про вакансії використовувалась бібліотека psycopg2 для взаємодії з базою даних PostgreSQL. Це забезпечує ефективне зберігання та доступ до даних про вакансії.

Результатом реалізації клієнтської та серверної частин стала функціональна система пошуку вакансій з використанням геоінформаційних систем. Клієнтська частина надає зручний та привабливий інтерфейс для користувача, де результати пошуку відображаються на карті, що допомагає в зоровому сприйнятті та аналізі розташування вакансій. Серверна частина забезпечує збирання та обробку даних про вакансії з різних джерел, їх збереження та ефективне управління.

У цьому розділі було представлено детальний опис реалізації клієнтської та серверної частин системи пошуку вакансій з використанням геоінформаційних систем. Використання Python, PyQt, Google Maps API, QGIS, BeautifulSoup4 та PostgreSQL дозволило успішно реалізувати необхідний функціонал системи. Комбінація цих інструментів забезпечує ефективну обробку даних, зручний користувацький інтерфейс та точність відображення вакансій на карті з урахуванням геолокації. Всі ці аспекти сприяють поліпшенню процесу пошуку вакансій та надають користувачам потужний інструмент для знаходження релевантних пропозицій роботи.

ВИСНОВОК

У проєкті було проведено аналіз предметної області, де розглянуто сферу працевлаштування, системи пошуку вакансій та роль геоінформаційних систем в цьому процесі. Було виявлено, що існуючі рішення, такі як work.ua та robota.ua, не надають можливості фільтрування вакансій за географічним положенням.

У процесі проєктування були визначені функціональні та нефункціональні вимоги до програмного забезпечення. Для реалізації системи були обрані відповідні інструменти, такі як QGIS, Python, Google Maps API, PyQt, PostgreSQL та BeautifulSoup4, які допомогли забезпечити функціональність та ефективність системи.

Система має клієнтську та серверну частини. Клієнтська частина реалізована з використанням Python та PyQt для створення інтерфейсу користувача, Google Maps API для роботи з картою та QGIS для фільтрації вакансій за відстанню. Серверна частина використовує стандартні бібліотеки Python, а також BeautifulSoup4 та pyscorg2 для парсингу та збереження даних про вакансії.

Проєкт успішно реалізований та надає зручний та ефективний інструмент для пошуку вакансій з використанням геоінформаційних систем. Система допомагає користувачам швидко знаходити вакансії, що знаходяться в певному радіусі від їхнього місця розташування на карті. Вона сприяє покращенню процесу працевлаштування та забезпечує зручну інтерактивну картографічну систему для ефективного пошуку вакансій.

Проєкт має потенціал для подальшого розширення та вдосконалення. Його впровадження сприятиме поліпшенню пошуку вакансій, враховуючи географічні вимоги та потреби користувачів.

					ІА91.110БАК.003 ПЗ	Арк.
						58
Зм.	Лист	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Україна піднялася на 17 пунктів у Global Services Location Index та увійшла до 50 країн із найбільш інноваційною економікою. [Електронний ресурс]. Режим доступу: <https://ny.mfa.gov.ua/news/44137-ukrajina-pidnyalasya-na-17-punktiv-u-global-services-location-index-ta-vvijshla-do-50-krajin-iz-najbilysh-innovacijnoju-jekonomikoju>
2. Сайт пошуку вакансій work.ua [Електронний ресурс]. Режим доступу: <https://www.work.ua/>
3. Сайт пошуку вакансій robota.ua [Електронний ресурс]. Режим доступу: <https://rabota.ua/ua>
4. StackOverflow Developer Survey 2022 [Електронний ресурс]. Режим доступу: <https://survey.stackoverflow.co/2022/#overview>
5. Qt documentation. Qt for WebAssembly [Електронний ресурс]. Режим доступу: <https://survey.stackoverflow.co/2022/#overviewhttps://doc.qt.io/qt-6/wasm.html>
6. WebAssembly [Електронний ресурс]. Режим доступу: <https://developer.mozilla.org/ru/docs/WebAssembly>
7. Райан Митчелл Современный скрапинг веб-сайтов с помощью Python. 2-е межд. издание . — СПб.: Питер, 2021, с. 6
8. П. О. Масляк, С. Л. Капіруліна Географія : підруч. для 8-го кл. загальноосвіт. навч. закладів. — Кам'янець-Подільський : Аксіома, 2016. — 304 с., с. 55
9. Qt documentation. Signals & Slots [Електронний ресурс]. Режим доступу: <https://doc.qt.io/qt-6/signalsandslots.html>
10. Python documentation. Generators [Електронний ресурс]. Режим доступу: <https://wiki.python.org/moin/Generators>
11. Andrew Lombardi WebSocket: Lightweight Client-Server Communications — O'Reilly Media, Inc., 2015 — 144 с., с. 5

					ІА91.110БАК.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

12. Joel Lawhead QGIS Python Programming Cookbook – Packt Publishing, 2015. – 464 с., с. 140
13. Вихідний код класу QgsDistanceArea [Електронний ресурс]. Режим доступу:
<https://github.com/qgis/QGIS/blob/9db101a49b697f79038d3112c02cd83c4d78b5b5/src/core/qgsdistancearea.cpp#L283>
14. T. Vincenty Survey Review, Volume 23, Issue 176 DIRECT AND INVERSE SOLUTIONS OF GEODESICS ON THE ELLIPSOID WITH APPLICATION OF NESTED EQUATIONS – Directorate of Overseas Surveys of the Ministry of Overseas Development Kingston Road, Tolworth, Surrey, 1975, с. 88
15. QSS Style Sheet Syntax [Електронний ресурс]. Режим доступу:
<https://doc.qt.io/qt-6/stylesheet-syntax.html>
16. В.В. Дунаев Базы данных. Язык SQL для студента: 2-е изд. доп и перераб. – СПб.: БХВ-Петербург, 2007. – 320 с.: ил.

					IA91.110БАК.003 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		60

Додаток А

QR-код з посиланням на код програми

<https://github.com/c1nnam0nbun/QGISProject>

