

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

## Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ  
SOFTWARE DEVELOPMENT TO RECRUIT SKILLED STAFF

Виконав(ла): студент(ка) 4 курсу, групи КНТ-211

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

СМЬОРДОВ Д.С.

(ПРИЗВИЩЕ та ініціали)

Керівник ШИТІКОВА О.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**  
 (повне найменування закладу вищої освіти)

Факультет КНТ  
 Кафедра програмних засобів  
 Ступінь вищої освіти бакалавр  
 Спеціальність 122 Комп'ютерні науки  
 (КОД І НАЙМЕНУВАННЯ)  
 Освітня програма (спеціалізація) Комп'ютерні науки  
 (назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
Сергій СУББОТІН  
 “ ” 2025 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

СМБОРДОВА Данила Сергійовича  
 (ПРІЗВИЩЕ, ім'я, по батькові)

- Тема проєкту (роботи) Розробка програмного забезпечення підбору кваліфікованого персоналу  
Software Development to Recruit Skilled Staff  
 керівник проєкту (роботи) к.т.н. ШИТИКОВА Олена Вікторівна,  
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)  
 затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209
- Строк подання студентом проєкту (роботи) 02 червня 2025 року
- Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання
- Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Проєктування програми. 3. Розробка програми. 4. Експлуатація та тестування програми.
- Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)  
Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | ПРІЗВИЩЕ, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-4 Основна частина | ШИТІКОВА О.В., доцент                     |                |                           |
| Нормоконтроль       | КАМІНСЬКА Ж.К., асистент                  |                |                           |

7. Дата видачі завдання «14» квітня 2025 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                             | Строк виконання етапів проєкту (роботи) | Примітка     |
|-------|----------------------------------------------------------------------|-----------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                          | 1 тиждень                               | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                           | 1 тиждень                               | Розділ 1     |
| 3     | Вибір мови програмування та інших технологій розробки.               | 2 тиждень                               | Розділ 2     |
| 4     | Розробка архітектури програми.                                       | 2 тиждень                               | Розділ 3     |
| 5     | Розробка програми.                                                   | 3-4 тижні                               | Розділ 3, 4  |
| 6     | Тестування та експериментальне дослідження програмного забезпечення. | 5 тиждень                               | Розділ 4     |
| 7     | Оформлення пояснювальної записки та документів до неї.               | 6 тиждень                               | Додатки      |
| 8     | Нормоконтроль та рецензування.                                       | 7 тиждень                               |              |
| 9     | Захист роботи.                                                       | 8 тиждень                               |              |

Студент(ка)

\_\_\_\_\_ Данило СМЬОРДОВ  
( підпис ) (Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_ Олена ШИТІКОВА  
( підпис ) (Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 124 с., 3 табл., 27 рис., 3 дод., 6 джерел.

КВАЛІФІКОВАНИЙ ПЕРСОНАЛ, ВАКАНСІЯ, ПРАЦЕДАВЕЦЬ, РЕКРУТИНГ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт роботи – процес підбору кваліфікованого персоналу. Предмет роботи – програмне забезпечення підбору кваліфікованого персоналу. Мета роботи – розробити програмне забезпечення підбору кваліфікованого персоналу для надання можливості реалізувати вибір потрібних працівників.

Матеріали, методи та технічні засоби: мова Python, фреймворк Django, що сукупно забезпечує розробку вебзастосунку, система MySQL для керування базою даних.

Результати. Створена в підсумку програма дозволяє не тільки виконувати пошук кваліфікованого персоналу, але і подальший відбір серед всіх кандидатів, які подали заявки на вакансії. Забезпечується підтримка зі сторони рекрутингових компаній, які можуть надавати анкети додаткових кандидатів.

Висновки. Програма розроблена, щоб забезпечити для компаній-працедавців підбір кваліфікованого персоналу на основі пошуку такого персоналу компаніями на основі визначення вакансій, пошуку вакансій кандидатами та організації відбору кваліфікованого персоналу через проведення етапів такого відбору з голосуванням представників компанії-роботодавця, проведенням за необхідності перевірок вмінь і знань кандидатів на вакансії для кваліфікованого персоналу на основі тестування.

Галузь використання. Компанії, які потребують відбору кваліфікованого персоналу для себе, реалізуючи цю необхідність через самостійний пошук або за допомогою звернення до рекрутингової компанії.

## ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 124 pages, 3 tables, 27 figures, 3 appendixes, 6 sources.

SKILLED STAFF, VACANCY, EMPLOYER, RECRUITMENT, SOFTWARE.

The object of the work is the process of selecting skilled staff. The subject of the work is software for selecting skilled staff. The purpose of the work is to develop software for selecting skilled staff to provide the opportunity to select the necessary employees.

Materials, methods and technical tools: Python language, Django framework, providing web application development, MySQL system for database management.

Results. The resulting program allows not only to search for skilled staff, but also to further select among all candidates who have applied for vacancies. Support is provided by recruiting companies, which can provide questionnaires for additional candidates.

Conclusions. The program is designed to provide employers with the selection of skilled staff based on the search for such personnel by companies based on the definition of vacancies, the search for vacancies by candidates and the organization of the selection of skilled staff through the stages of such selection with the voting of representatives of the employer company, conducting, if necessary, tests of the skills and knowledge of candidates for vacancies for skilled staff based on testing.

Field of use. Companies that need to select skilled staff for themselves, realizing this need through an independent search or by contacting a recruiting company.

## ЗМІСТ

|                                                                                      |     |
|--------------------------------------------------------------------------------------|-----|
|                                                                                      | С.  |
| Перелік скорочень та умовних познач.....                                             | 7   |
| Вступ.....                                                                           | 8   |
| 1 Аналіз проблеми та постановка завдань дослідження .....                            | 9   |
| 1.1 Стан розробки програмного забезпечення підбору кваліфікованого персоналу.....    | 9   |
| 1.2 Порівняння програм підбору кваліфікованого персоналу, що існують                 | 11  |
| 1.3 Висновки за розділом 1 .....                                                     | 12  |
| 2 Проєктування програми.....                                                         | 14  |
| 2.1 Обґрунтування вибору засобів розробки програми .....                             | 14  |
| 2.2 Визначення вимог користувачів до програми підбору кваліфікованого персоналу..... | 15  |
| 2.3 Проєктування структури бази даних.....                                           | 24  |
| 2.4 Висновки за розділом 2 .....                                                     | 43  |
| 3 Розробка програми .....                                                            | 44  |
| 3.1 Послідовність роботи з програмою.....                                            | 44  |
| 3.2 Реалізація програми підбору кваліфікованого персоналу .....                      | 45  |
| 3.3 Висновки за розділом 3 .....                                                     | 73  |
| 4 Експлуатація та тестування програми .....                                          | 74  |
| 4.1 Призначення програми .....                                                       | 74  |
| 4.2 Опис роботи з програмою підбору кваліфікованого персоналу .....                  | 74  |
| 4.3 Тестування програми підбору кваліфікованого персоналу .....                      | 78  |
| 4.4 Висновки за розділом 4 .....                                                     | 86  |
| Висновки .....                                                                       | 87  |
| Перелік джерел посилання .....                                                       | 88  |
| Додаток А Технічне завдання .....                                                    | 89  |
| Додаток Б Текст програми .....                                                       | 96  |
| Додаток В Слайди презентації.....                                                    | 117 |

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

БД – база даних;

ВС – вебсайт.

## ВСТУП

Від мотивації персоналу залежить ефективність роботи всієї компанії. Однак іншою важливою складовою є сама якість підібраних кадрів, адже вона також безпосередньо впливає на результати компанії.

Підібрати кваліфікований персонал не так і просто на сучасному ринку. Пропозицій може бути багато, проте багато часу витрачається на відбір тих кандидатів, які дійсно відповідають вимогам працедавця. Тому важливо розробити програмне забезпечення, яке б з одного боку підтримувало процес пошуку персоналу, з іншого боку враховувало саме кваліфікацію такого персоналу, і з третього – забезпечувало би і підтримку самого відбору, а не тільки пошук.

У підсумку метою роботи є розробити програмне забезпечення підбору кваліфікованого персоналу для надання можливості реалізувати вибір потрібних працівників.

Розроблюване програмне забезпечення повинно враховувати і надавати пов'язані з цим такі можливості:

- пошуку кваліфікованого персоналу зі створенням відповідних вакансій на основі відгуків від самих кандидатів;
- надання пропозицій щодо кандидатів для вакансій рекрутинговими компаніями, які повинні мати своє представництво в програмі;
- реалізації багатоетапного відбору кандидатів для визначення вибору кваліфікованого персоналу з підтримкою голосування на цих етапах;
- забезпечення базової перевірки кандидатів за поданими заявками на вакансії на основі автоматизованого тестування.

# 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

## 1.1 Стан розробки програмного забезпечення підбору кваліфікованого персоналу

Одним з найбільш відомих у світі програмних засобів пошуку кваліфікованого персоналу є вебзастосунок Indeed (рис. 1.1), представлений у першу чергу вебсайтом (ВС) [1].

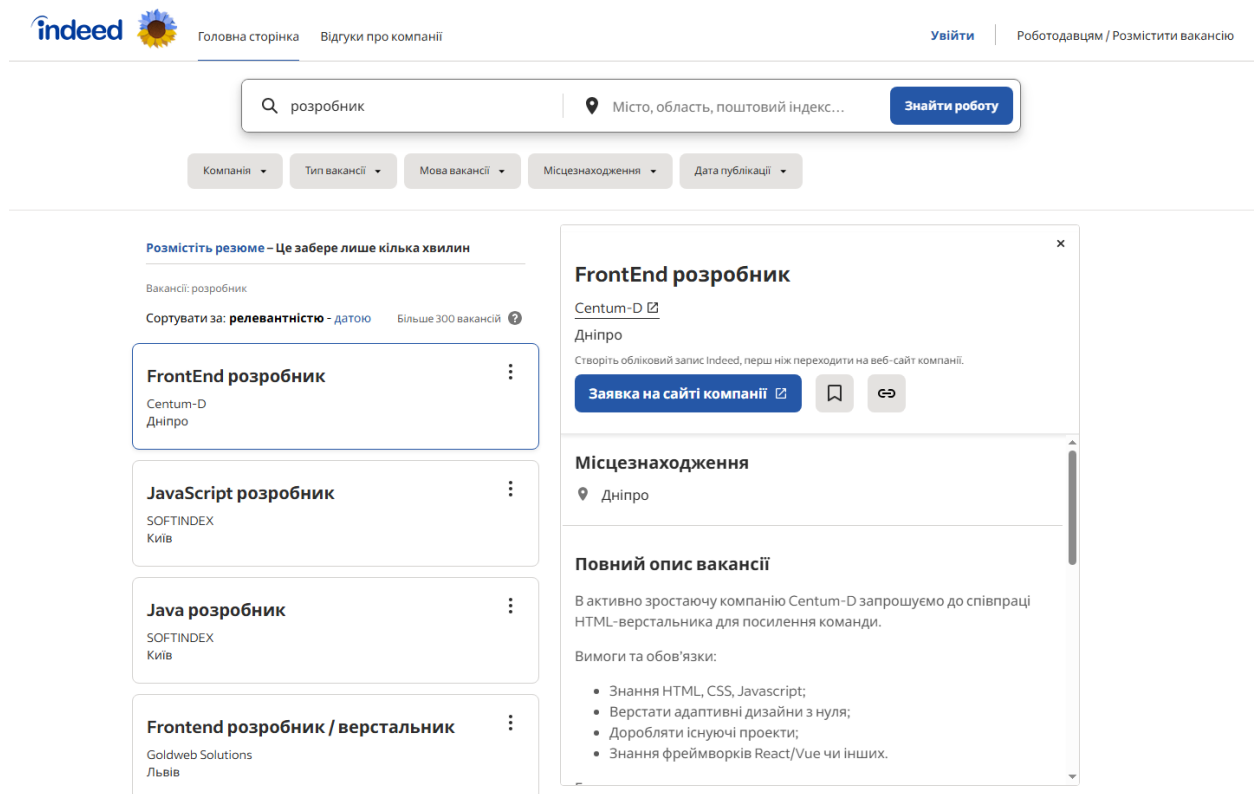


Рисунок 1.1 – Інтерфейс ВС Indeed

Indeed дозволяє шукати вакансії, зберігати потрібні, виставляти свої вакансії, підписувати на працедавців, переглядати сторінки компаній-працедавців, залишати відгуки на працедавців, переглядати вакансії за компаніями. Проте у програмі немає засобів відбору безпосередньо, тобто подальший процес після знаходження анкети, контактування не підтримується. Окрім того і подання заявок фактично відбувається через ВС самих компаній.

В Україні найбільшу популярність має ВС сервісу work.ua (рис. 1.2) [2].

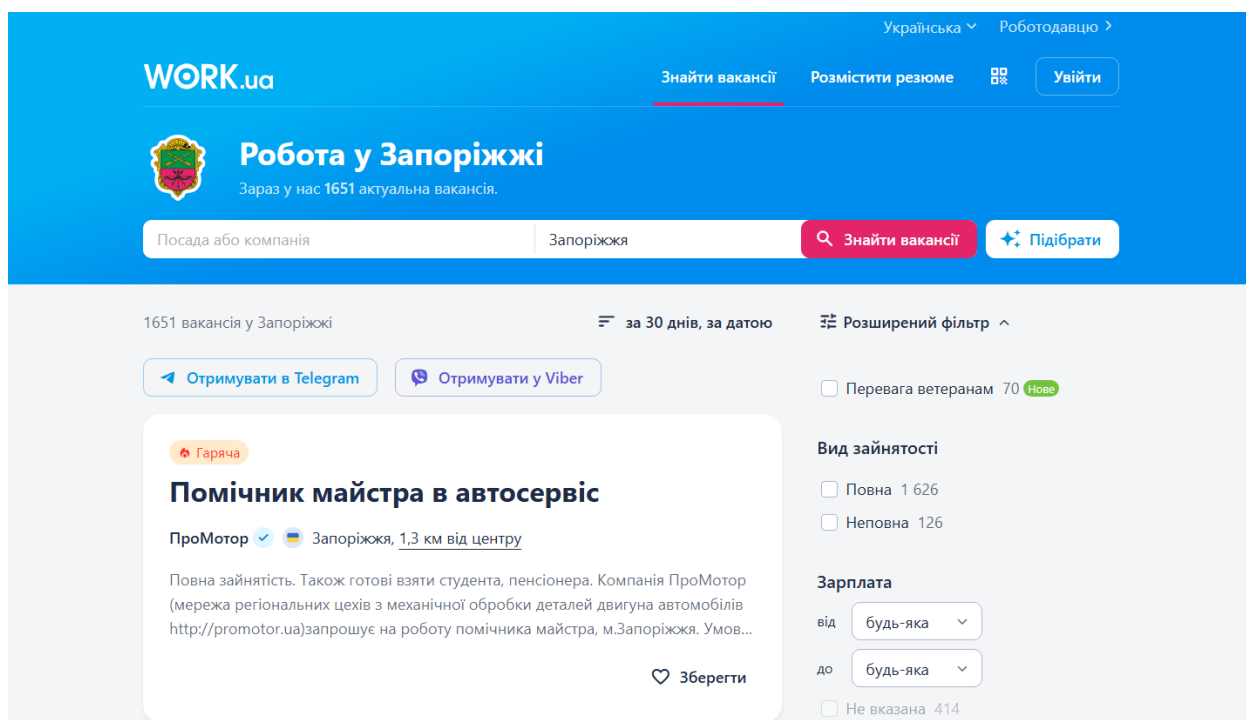


Рисунок 1.2 – Інтерфейс ВС work.ua

ВС work.ua надає можливість шукати вакансії, містить окремий інтерфейс для працедавця. За результатами пошуку вакансії можна відкрити сторінку самої вакансії, зберегти її, відгукнутися на вакансію засобами самого ВС. Можна також переглядати дані працедавця, доступні за ним вакансії. При цьому якось оцінювати, реагувати на працедавців неможливо, що дещо обмежує зв'язок у загальній взаємодії одностороннім. Кандидати, тобто звичайні користувачі, можуть розташовувати свої резюме.

ВС robota.ua (рис. 1.3) також є одним з найпопулярніших засобів пошуку роботи в Україні [3]. Він також надає можливість шукати серед вакансій потрібну, відкривати її сторінку, зберігати вакансію, відгукуватись на неї, а також негативно оцінювати вакансії. Зі сторінки працедавця можна підписатися на його вакансії, відповідно потім переглядати вакансії тих працедавців, на які підписався користувач. Також можна переглядати вакансії працедавця.

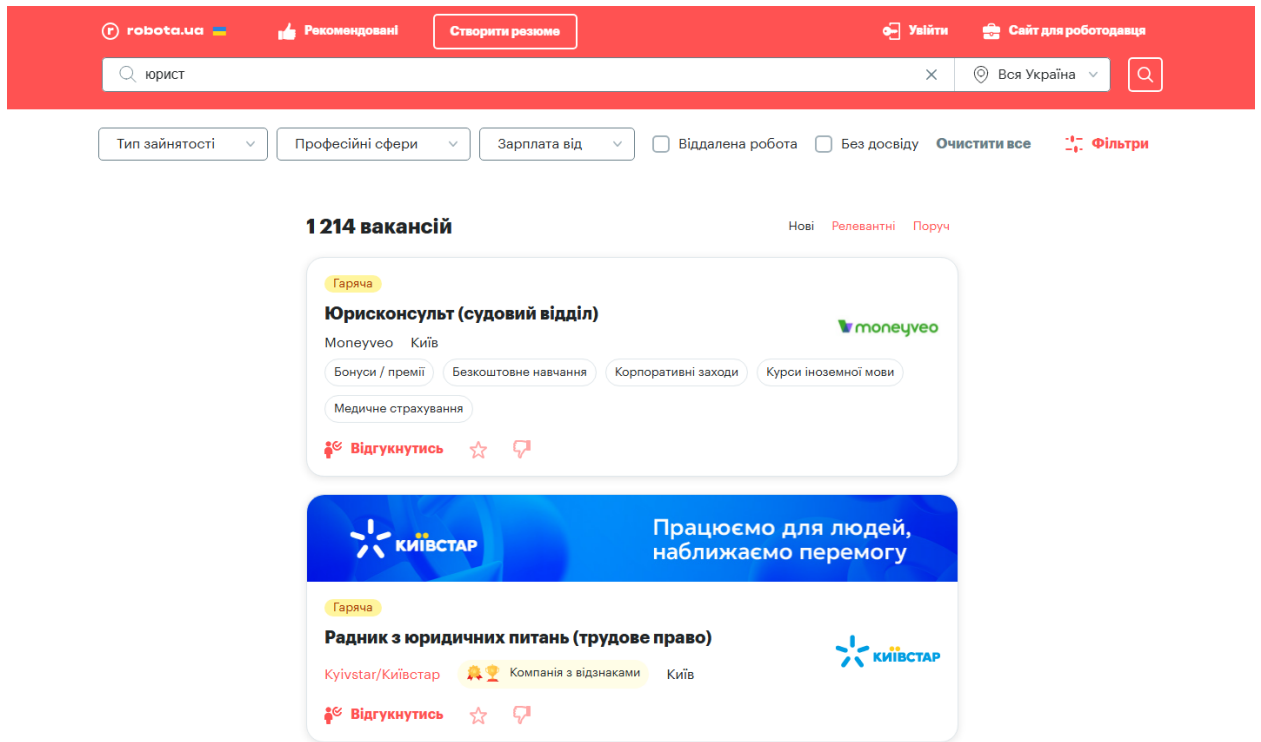


Рисунок 1.3 – Інтерфейс ВС robota.ua

## 1.2 Порівняння програм підбору кваліфікованого персоналу, що існують

Враховуючи визначені особливості програм підбору кваліфікованого персоналу, що існують, виділені у попередньому підрозділі, до результуючого порівняння виділено Indeed та work.ua (табл. 1.1).

Розробка повинна відрізнятися від них головним чином наявністю засобів підтвердження працедавцем даних анкети кандидата, реалізації багатоетапного відбору кваліфікованого персоналу на основі голосування, перевірки вмінь заявників автоматичним способом у вигляді тестування, взаємодією з рекрутинговими компаніями, коли працедавці можуть надавати їм запити на визначення кандидатів за вакансіями, а вони можуть в тому числі вносити дані своїх кандидатів на вакансії у програму, якщо вони в програмі не присутні.

Таблиця 1.1 – Порівняння програм підбору кваліфікованого персоналу

| Критерій порівняння                                                | Indeed | Work.ua | Розробка |
|--------------------------------------------------------------------|--------|---------|----------|
| Пошук вакансій                                                     | +      | +       | +        |
| Збереження вакансії                                                | +      | +       | +        |
| Перегляд сторінки<br>працедавця                                    | +      | +       | +        |
| Підписування на<br>працедавців                                     | +      | —       | +        |
| Реагування на<br>працедавців                                       | +      | —       | +        |
| Підтвердження<br>працедавцем даних<br>анкети кандидата<br>(резюме) | —      | —       | +        |
| Реалізація багатоетапного<br>відбору кваліфікованого<br>персоналу  | —      | —       | +        |
| Перевірка вмінь<br>заявників                                       | —      | —       | +        |
| Створення власної анкети<br>за професією (резюме)                  | +      | +       | +        |
| Взаємодія з<br>рекрутинговими<br>компаніями                        | —      | —       | +        |

### 1.3 Висновки за розділом 1

Необхідно забезпечити виконання завдань:

– аналіз стану проблеми розробки програм підбору кваліфікованого персоналу;

- проєктування програми підбору кваліфікованого персоналу;
- реалізація, тестування програми підбору кваліфікованого персоналу.

Перший розділ є реалізацією завдання аналізу стану проблеми розробки програм підбору кваліфікованого персоналу, що дозволило також визначити особливості власної розробки порівняно з Indeed, work.ua.

## 2 ПРОЄКТУВАННЯ ПРОГРАМИ

### 2.1 Обґрунтування вибору засобів розробки програми

Розробка вебзастосунків на основі мови Python має широку перспективу за рахунок доступності великої кількості засобів розробки для цієї мови, що важливо при подальшому розширенні, масштабуванні. Окрім того мова надає високу підтримку веброзробки, зокрема на основі фреймворків FastAPI [4] і Django [5] (табл. 2.1). Django було віддано перевагу, враховуючи його придатність для більших проєктів, швидшу розробку, вбудовані засоби, що також пришвидшує розробку і робить її більш стабільною. Також було враховано, що Django має дещо меншу продуктивність порівняно з FastAPI, але в комплексі засобів для реальних застосунків не поступається за продуктивністю фактично.

Таблиця 2.1 – Порівняння засобів розробки програми підбору кваліфікованого персоналу

| Критерій порівняння       | FastAPI                                                                             | Django                                                                           |
|---------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Основна направленість     | Мікросервіси                                                                        | Великі (можливо середні) вебзастосунки                                           |
| Обмеженість часу розробки | Потребує менше часу вивчення засобів за рахунок меншої кількості вбудованих засобів | Направлений на швидке прототипування та швидку розробку                          |
| Вбудовані засоби          | WebSocket, підтримка різних форматів даних (JSON, XML, Protobuf, GraphQL)           | DjangoAdmin, об'єктно-реляційне відображення, рушій шаблонів, фреймворк для форм |
| Продуктивність            | Висока                                                                              | Комплексно висока                                                                |

## 2.2 Визначення вимог користувачів до програми підбору кваліфікованого персоналу

Користувач до авторизації у програмі підбору кваліфікованого персоналу має бути забезпечений потоком таких основних сценаріїв:

- авторизація;
- перегляд сторінки компанії-працедавця;
- перегляд сторінки офісу-представництва компанії-працедавця;
- перегляд вакансії для кваліфікованого персоналу;
- перегляд відкритих вакансій офісу-представництва компанії-працедавця;
- пошук вакансій для кваліфікованого персоналу (рис. 2.1).



Рисунок 2.1 – Діаграма варіантів використання програми підбору кваліфікованого персоналу користувачем до авторизації

Альтернативний сценарій у такому випадку тільки один – реєстрація користувачів-кандидатів, що є альтернативою авторизації, коли користувач не має облікового запису користувача-кандидата для подальшої роботи.

Користувач-кандидат у програмі підбору кваліфікованого персоналу має бути забезпечений потоком таких основних сценаріїв:

- встановлення підтримки компанії-працедавцю;
- перегляд підтриманих компаній-працедавців користувачем-кандидатом;
- перегляд сторінки користувача-кандидата;
- додавання даних освіти користувача-кандидата;
- додавання історії роботи користувача-кандидата;
- створення анкети за професією;
- подання анкети за професією на вакансію для кваліфікованого персоналу;
- перегляд власних поданих анкет за професією на вакансії для кваліфікованого персоналу;
- внесення даних користувача-кандидата;
- оформлення підписки на компанію-працедавця;
- перегляд вакансій для кваліфікованого персоналу компаній-працедавців з підписок;
- збереження вакансії для кваліфікованого персоналу у свій перелік;
- перегляд збережених вакансій для кваліфікованого персоналу;
- перегляд сторінки перевірки за вакансією для кваліфікованого персоналу;
- збереження результатів перевірки користувача за вакансією для кваліфікованого персоналу;
- перегляд призначених перевірок за вакансіями для користувача;
- перегляд анкети за професією на вакансію для кваліфікованого персоналу.

Альтернативні сценарії користувача-кандидата:

- вилучення підтримки компанії-працедавцю для встановлення підтримки компанії-працедавцю, коли підтримка компанії-працедавцю цим користувачем-кандидатом була встановлена раніше;

- оновлення даних освіти користувача-кандидата для додавання даних освіти користувача-кандидата, коли ці дані освіти користувача-кандидата були додані, але потрібні уточнення;
- вилучення даних освіти користувача-кандидата для додавання даних освіти користувача-кандидата, коли ці дані освіти користувача-кандидата були додані, але помилково;
- оновлення історії роботи користувача-кандидата для додавання історії роботи користувача-кандидата, коли ці дані історії роботи користувача-кандидата були додані, але потрібні уточнення;
- вилучення історії роботи користувача-кандидата для додавання історії роботи користувача-кандидата, коли ці дані історії роботи користувача-кандидата були додані, але помилково;
- оновлення даних анкети за професією для створення анкети за професією, коли ці дані анкети за професією були внесені, але потрібні уточнення;
- вилучення анкети за професією для створення анкети за професією, коли ця анкета за професією більше не буде використовуватись;
- вилучення анкети за професією на вакансію для кваліфікованого персоналу для подання анкети за професією на вакансію для кваліфікованого персоналу, коли ця анкета за професією була подана на вакансію для кваліфікованого персоналу, але більше це не потрібно;
- вилучення підписки на компанію-працедавця для оформлення підписки на компанію-працедавця, коли підписка цим користувачем-кандидатом на цю компанію-працедавця була створена, але більше не потрібна;
- вилучення збереження вакансії для кваліфікованого персоналу зі свого переліку для збереження вакансії для кваліфікованого персоналу у свій перелік, коли збереження вакансії для кваліфікованого персоналу у свій перелік було виконано цим користувачем-кандидатом, але більше не потрібне (рис. 2.2).

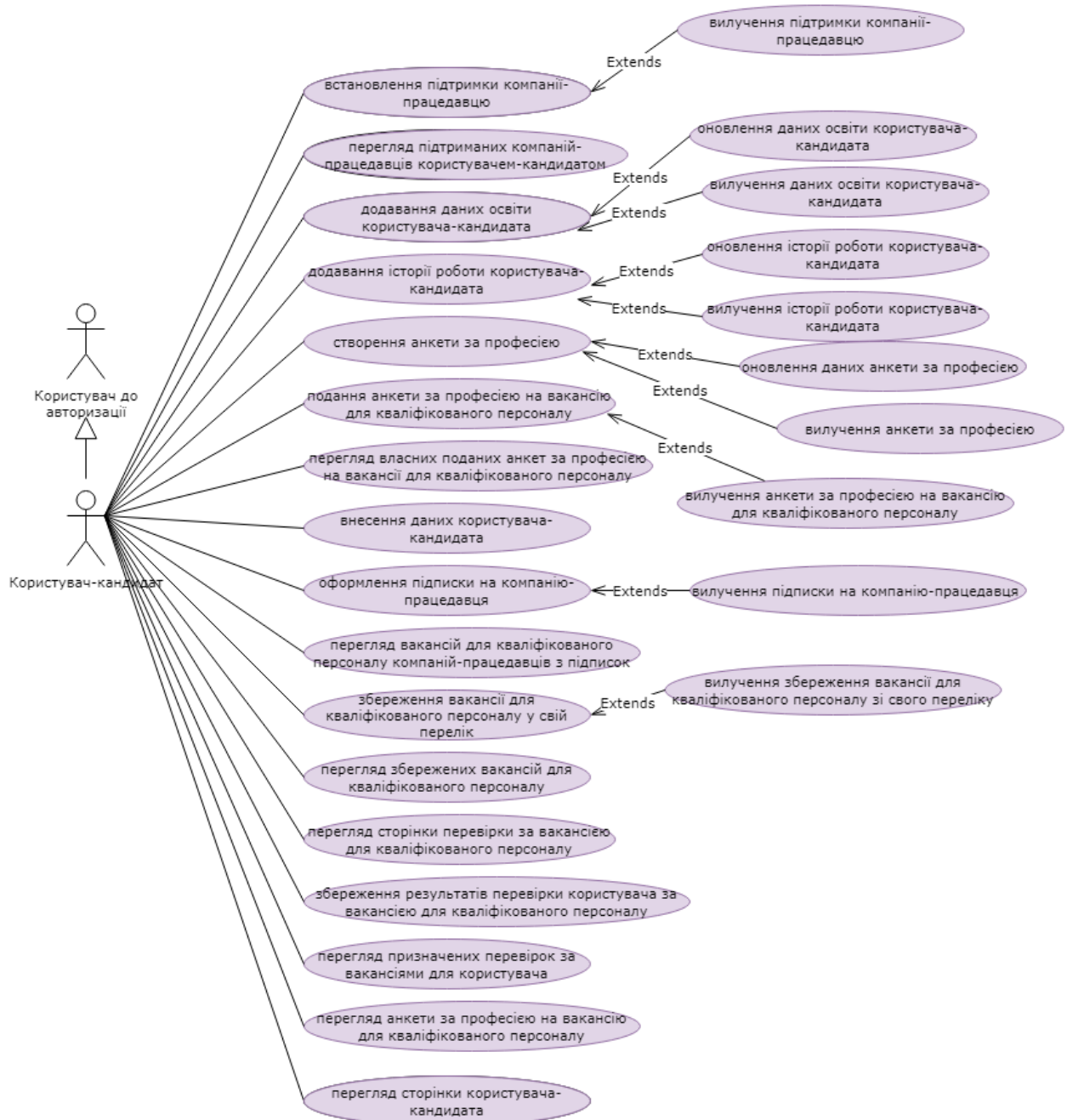


Рисунок 2.2 – Діаграма варіантів використання програми підбору кваліфікованого персоналу для користувача-кандидата

Представник компанії-працедавця у програмі підбору кваліфікованого персоналу має бути забезпечений потоком таких основних сценаріїв:

- реєстрація представників компаній-працедавців;
- перегляд сторінки рекрутингової компанії;
- визначення контакту з рекрутинговою компанією компанії-працедавця;

- перегляд сторінки користувача-кандидата;
- перегляд рекрутингових компаній компанії-працедавця;
- створення вакансії для кваліфікованого персоналу;
- голосування за анкету кандидата за вакансією для кваліфікованого персоналу;
- перегляд доступних для голосування вакансій для кваліфікованого персоналу;
- перегляд доступних для голосування анкет кандидатів за обраною вакансією;
- перегляд анкет кандидатів за вакансіями, за які проголосував користувач;
- визначення рекрутингової компанії за вакансією компанії-працедавця;
- створення етапу відбору за вакансією для кваліфікованого персоналу;
- перегляд результатів проведення відбору за вакансією для кваліфікованого персоналу;
- керування анкетною за професією на вакансію для кваліфікованого персоналу;
- перегляд користувачів-кандидатів за даними рекрутингової компанії;
- підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем;
- перегляд користувачів, які працювали в компанії-працедавці;
- внесення даних компанії-працедавця;
- внесення даних офісу-представництва компанії-працедавця;
- створення перевірки за вакансією для кваліфікованого персоналу;
- додавання запитання до перевірки за вакансією для кваліфікованого персоналу;
- додавання відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу;

- перегляд сторінки перевірки за вакансією для кваліфікованого персоналу;
- перегляд анкети за професією на вакансію для кваліфікованого персоналу.

Альтернативні сценарії для представника компанії-працедавця:

- скасування контакту з рекрутинговою компанією компанії-працедавця для визначення контакту з рекрутинговою компанією компанії-працедавця, коли такий контакт з рекрутинговою компанією був визначений цією компанією-працедавцем, але більше не використовується;
- оновлення вакансії для кваліфікованого персоналу для створення вакансії для кваліфікованого персоналу, коли ця вакансія для кваліфікованого персоналу була створена, але має бути уточнена;
- вилучення голосування за анкету кандидата за вакансією для кваліфікованого персоналу для голосування за анкету кандидата за вакансією для кваліфікованого персоналу, коли таке голосування за анкету кандидата за вакансією для кваліфікованого персоналу цим представником компанії-працедавця було виконано, але не відповідає дійсності;
- видалення рекрутингової компанії за вакансією компанії-працедавця для визначення рекрутингової компанії за вакансією компанії-працедавця, коли таке визначення рекрутингової компанії за вакансією компанії-працедавця було встановлено, але більше не відповідає дійсності;
- оновлення етапу відбору за вакансією для кваліфікованого персоналу для створення етапу відбору за вакансією для кваліфікованого персоналу, коли такий етап відбору за вакансією для кваліфікованого персоналу був створений, але повинен бути уточнений;
- вилучення етапу відбору за вакансією для кваліфікованого персоналу для створення етапу відбору за вакансією для кваліфікованого персоналу, коли такий етап відбору за вакансією для кваліфікованого персоналу був створений, але повинен бути уточнений;

– вилучення підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем для підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем, коли таке підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем було виконано, але не відповідає більше дійсності;

– оновлення параметрів перевірки за вакансією для кваліфікованого персоналу для створення перевірки за вакансією для кваліфікованого персоналу, коли така вакансія для кваліфікованого персоналу була створена, але потребує уточнення;

– вилучення перевірки за вакансією для кваліфікованого персоналу для створення перевірки за вакансією для кваліфікованого персоналу, коли така перевірка за вакансією для кваліфікованого персоналу була створена, але потребує уточнення;

– вилучення запитання до перевірки за вакансією для кваліфікованого персоналу для додавання запитання до перевірки за вакансією для кваліфікованого персоналу, коли таке запитання до перевірки за вакансією для кваліфікованого персоналу було додане, але більше не потрібне;

– оновлення запитання до перевірки за вакансією для кваліфікованого персоналу для додавання запитання до перевірки за вакансією для кваліфікованого персоналу, коли таке запитання до перевірки за вакансією для кваліфікованого персоналу було додано, але потребує уточнення;

– вилучення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу для додавання відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу, коли така відповідь на запитання до перевірки за вакансією для кваліфікованого персоналу була створена, але більше не потрібна;

– оновлення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу для додавання відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу, коли така відповідь на запитання

до перевірки за вакансією для кваліфікованого персоналу була створена, але потребує уточнення (рис. 2.3).

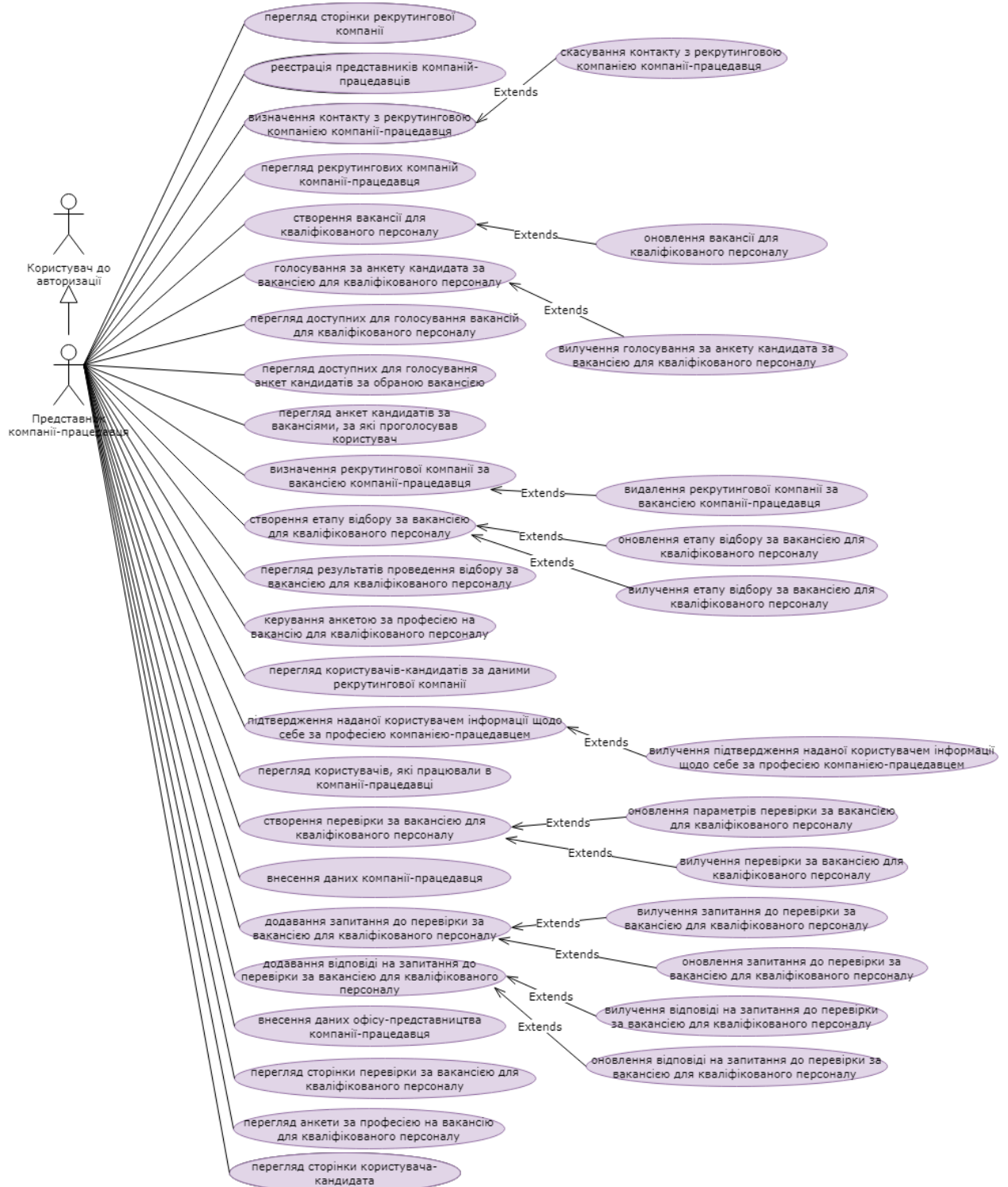


Рисунок 2.3 – Діаграма варіантів використання програми підбору кваліфікованого персоналу для представника компанії-працедавця

Представник рекрутингової компанії у програмі підбору кваліфікованого персоналу має бути забезпечений потоком таких основних сценаріїв:

- реєстрація представників рекрутингових компаній;
- перегляд сторінки рекрутингової компанії;
- скасування контакту з рекрутинговою компанією компанії-працедавця;
- перегляд сторінки користувача-кандидата;
- перегляд компаній-працедавців рекрутингової компанії;
- перегляд вакансій для кваліфікованого персоналу, визначених для обробки рекрутинговою компанією;
- визначення завершення оброблення вакансії рекрутинговою компанією;
- створення анкети за професією;
- подання анкети за професією на вакансію для кваліфікованого персоналу;
- перегляд користувачів-кандидатів за даними рекрутингової компанії;
- перегляд анкети за професією на вакансію для кваліфікованого персоналу;
- внесення даних рекрутингової компанії (рис. 2.4).

Альтернативні сценарії для представника рекрутингової компанії:

- оновлення даних анкети за професією для створення анкети за професією, коли така анкета за професією була створена, але потребує уточнення;
- вилучення анкети за професією для створення анкети за професією, коли така анкета за професією була створена, але більше не потрібна;
- вилучення анкети за професією на вакансію для кваліфікованого персоналу для подання анкети за професією на вакансію для кваліфікованого персоналу, коли така анкета за професією на цю вакансію для кваліфікованого персоналу була подана, але більше не потрібна.



Рисунок 2.4 – Діаграма варіантів використання програми підбору кваліфікованого персоналу для представника рекрутингової компанії

## 2.3 Проєктування структури бази даних

База даних (БД) важлива для побудови роботи з даними як такими, визначаючи підхід до роботи зі всією наявною в програмі інформацією. БД керована системою MySQL у цій програмі, що вирішено зробити, врахувавши її активне практичне застосування, високий рівень тих засобів, які надаються, безкоштовність.

У БД компанія-працедавець, яка потребує кваліфікованого персоналу, представлена таблицею Staffemployer з характеристиками:

- id – ідентифікатор компанії-працедавця, яка потребує кваліфікованого персоналу;
- staffemployercaption – коротка презентаційна назва компанії-працедавця, яка потребує кваліфікованого персоналу;

- `staffemployeroffname` – офіційна повна назва компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployerdescription` – текстовий опис компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployerphoto` – зображення для представлення компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployerweb` – вебсайт компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployeremail` – електронна адреса компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployerfacebook` – Facebook-сторінка компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployerinstagram` – Instagram-сторінка компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployeractiveyear` – рік початку роботи компанії-працедавця, яка потребує кваліфікованого персоналу;
- `staffemployerphone` – номер телефону компанії-працедавця, яка потребує кваліфікованого персоналу.

Підтримування компаній-працедавців, які потребують кваліфікованого персоналу, користувачами-кандидатами у кваліфікований персонал представлені таблицею `Staffemployersupport` з характеристиками:

- `id` – ідентифікатор підтримування компанії-працедавця, яка потребує кваліфікованого персоналу, користувачем-кандидатом у кваліфікований персонал;
- `staffemployer` – компанія-працедавець, яка потребує кваліфікованого персоналу і яку підтримав користувач-кандидат у кваліфікований персонал (зовнішній ключ);
- `skilledstaffaccount` – користувач-кандидат у кваліфікований персонал, який підтримав цю компанію-працедавця, яка потребує кваліфікованого персоналу (зовнішній ключ).

У БД міста представлені таблицею City з характеристиками:

- id – ідентифікатор міста;
- cityname – назва міста.

У БД офіси-представництва компаній-працедавців, які потребують кваліфікованого персоналу, представлені таблицею Staffemployeroffice з характеристиками:

- id – ідентифікатор офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу;
- staffemployer – компанія-працедавець, що потребує кваліфікованого персоналу та яку представляє цей офіс (зовнішній ключ);
- staffemployerofficephoto – зображення, що представляє офіс компанії-працедавця, що потребує кваліфікованого персоналу;
- staffemployerofficacity – місто, в якому знаходиться цей офіс-представництво компанії-працедавця, яка потребує кваліфікованого персоналу (зовнішній ключ);
- staffemployerofficeweb – вебсайт офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу;
- staffemployerofficedescription – текстовий опис офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу;
- staffemployerofficeaddress – адреса офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу;
- staffemployerofficeemail – електронна пошта офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу;
- staffemployerofficefacebook – Facebook-сторінка офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу;
- staffemployerofficeinstagram – Instagram-сторінка офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу;

– `staffemployerofficephone` – номер телефону офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу.

У БД облікові записи користувачів-представників офісів компаній-працедавців, які потребують кваліфікованого персоналу, представлені таблицею `Staffemployeraccount` з характеристиками:

– `id` – обліковий запис користувача-представника офісу компанії-працедавця, яка потребує кваліфікованого персоналу;

– `staffemployeroffice` – офіс компанії-працедавця, яка потребує кваліфікованого персоналу та яку представляє користувач (зовнішній ключ);

– `userindjango` – обліковий запис користувача, який представляє офіс компанії-працедавця, яка потребує кваліфікованого персоналу, в Django (зовнішній ключ);

– `staffphone` – номер телефону користувача, який представляє офіс компанії-працедавця, яка потребує кваліфікованого персоналу.

У БД рекрутингові компанії представлені таблицею `Recruiter` з характеристиками:

– `id` – ідентифікатор рекрутингової компанії;

– `recruitercaption` – коротка презентаційна назва рекрутингової компанії;

– `recruiteroffname` – офіційна повна назва рекрутингової компанії;

– `recruitercity` – місто, в якому знаходиться рекрутингова компанія (зовнішній ключ);

– `recruiterdescription` – текстовий опис рекрутингової компанії;

– `recruiterphoto` – зображення, що представляє рекрутингову компанію;

– `recruiterweb` – вебсайт рекрутингової компанії;

– `recruiteremail` – електронна адреса рекрутингової компанії;

– `recruiterfacebook` – Facebook-сторінка рекрутингової компанії;

– `recruiterinstagram` – Instagram-сторінка рекрутингової компанії;

– `recruiteractiveyear` – рік початку діяльності рекрутингової компанії;

– `recruiterphone` – номер телефону рекрутингової компанії.

У БД облікові записи користувачів, які представляють рекрутингові компанії, представлені таблицею `Recruiteraccount` з характеристиками:

- `id` – ідентифікатор облікового запису користувача, який представляє рекрутингову компанію;
- `recruiter` – рекрутингова компанія, яку представляє користувач (зовнішній ключ);
- `userindjango` – обліковий запис користувача, який представляє рекрутингову компанію, в Django (зовнішній ключ);
- `recruiterphone` – номер телефону користувача, який представляє рекрутингову компанію.

У БД встановлені контакти (або контракти) на обслуговування між рекрутинговими компаніями та компаніями-працедавцями, які потребують кваліфікованого персоналу, представлені таблицею `Staffemployerrecruiter` з характеристиками:

- `id` – ідентифікатор контакту на обслуговування між рекрутинговою компанією та компанією-працедавцем, яка потребує кваліфікованого персоналу;
- `staffemployer` – компанія-працедавець, яка потребує кваліфікованого персоналу та має контакт на обслуговування з цією рекрутинговою компанією (зовнішній ключ);
- `recruiter` – рекрутингова компанія, яка має контакт на обслуговування з цією компанією-працедавцем, яка потребує кваліфікованого персоналу (зовнішній ключ);
- `contractdate` – дата встановлення контакту між рекрутинговою компанією та компанією-працедавцем, яка потребує кваліфікованого персоналу.

У БД категорії професії кваліфікованого персоналу представлені таблицею `Staffprofessioncategory` з характеристиками:

- `id` – ідентифікатор категорії професії кваліфікованого персоналу;

– staffprofessioncategoryname – назва категорії професії кваліфікованого персоналу.

У БД професії кваліфікованого персоналу представлена таблицею Staffprofession з характеристиками:

- id – ідентифікатор професії кваліфікованого персоналу;
- staffprofessioncategory – категорія цієї професії кваліфікованого персоналу (зовнішній ключ);
- staffprofessionname – назва професії кваліфікованого персоналу.

У БД навички, які можуть визначатися для кваліфікованого персоналу, представлені таблицею Staffskill з характеристиками:

- id – ідентифікатор навички, яка може визначатися для кваліфікованого персоналу;
- staffskillname – назва навички, яка може визначатися для кваліфікованого персоналу.

У БД вакансії для кваліфікованого персоналу в компаніях-працедавцях представлені таблицею Skilledstaffdemand з характеристиками:

- id – ідентифікатор вакансії для кваліфікованого персоналу в компанії-працедавці;
- staffemployeroffice – компанія-працедавець, яка висунула цю вакансію для кваліфікованого персоналу (зовнішній ключ);
- skilledstaffdemandcaption – коротка назва вакансії для кваліфікованого персоналу;
- staffprofession – професія, кваліфікованого персоналу за якою потребує ця компанія-працедавець (зовнішній ключ);
- skilledstaffdemandtext – текстові вимоги до кваліфікованого працівника від компанії-працедавця;
- demandfinaltime – остання дата, до якої може бути відкрита ця вакансія для кваліфікованого персоналу;
- demandstatus – поточний стан цієї вакансії для кваліфікованого персоналу;

- `skilledstaffsalary` – заробітна плата, встановлена у вакансії для кваліфікованого персоналу;
- `skilledstaffattentiveness` – наявність вимоги до уважності кваліфікованого працівника за цією вакансією;
- `skilledstaffresponsibility` – наявність відповідальності до уважності кваліфікованого працівника за цією вакансією;
- `skilledstaffteam` – наявність вимоги до вміння працювати в команді кваліфікованого працівника за цією вакансією;
- `skilledstaffpatience` – наявність вимоги до терплячості кваліфікованого працівника за цією вакансією;
- `skilledstaffevolution` – наявність вимоги до бажання розвиватись кваліфікованого працівника за цією вакансією;
- `skilledstaffcritique` – наявність вимоги до критичності кваліфікованого працівника за цією вакансією;
- `currentskilledstaffdemandstage` – поточний етап відбору кваліфікованого персоналу за цією вакансією (зовнішній ключ);
- `skilledstaffeducationlevel` – потрібний рівень освіти кандидата.

У БД користувачі, які представляють компанії-працедавців та мають право голосувати під час відбору кваліфікованого персоналу за вакансіями, представлені таблицею `Skilledstaffdemandvoter` з характеристиками:

- `id` – ідентифікатор права користувача, який представляє компанію-працедавця, голосувати під час відбору кваліфікованого персоналу за вакансією;
- `skilledstaffdemand` – вакансія для кваліфікованого персоналу в компанії-працедавці, за якою може голосувати користувач (зовнішній ключ);
- `staffemployeraccount` – користувача, який представляє компанію-працедавця і який має право голосувати під час відбору кваліфікованого персоналу за вакансією (зовнішній ключ).

Вакансії для кваліфікованого персоналу в програмі представлені не тільки загальним оголошенням, але і вміннями, за якими цю кваліфікацію встановлено.

У БД вимоги до вмінь за вакансією для кваліфікованого персоналу представлені таблицею Skilledstaffdemandskill з характеристиками:

- id – ідентифікатор вміння за вакансією для кваліфікованого персоналу;
- skilledstaffdemand – вакансія для кваліфікованого персоналу, за якою є вимога за цим вмінням (зовнішній ключ);
- staffskill – вміння, за яким висунуто вимогу за вакансією для кваліфікованого персоналу (зовнішній ключ);
- skilllevel – рівень вмінь, який має бути забезпечений за цією вакансією для кваліфікованого персоналу;
- requirementplus – обов’язковість вимоги щодо цього вміння за цією вакансією для кваліфікованого персоналу (обов’язкова чи додатковий плюс).

У БД відповідальність рекрутингових компаній за пошук кваліфікованого персоналу представлена таблицею Skilledstaffdemandrecruiter з характеристиками:

- id – ідентифікатор встановленої відповідальності рекрутингової компанії за пошук кваліфікованого персоналу за вакансією;
- skilledstaffdemand – вакансія для кваліфікованого персоналу, за якою встановлено відповідальність рекрутингової компанії (зовнішній ключ);
- recruiter – рекрутингова компанія, за якою встановлено відповідальність за вакансією для кваліфікованого персоналу (зовнішній ключ);
- rightsmemo – нагадування щодо особливостей вимог до відповідальності рекрутингової компанії за пошук кваліфікованого персоналу за вакансією;
- finishedornot – завершено виконання пошуку кваліфікованого персоналу за вакансією рекрутинговою компанією.

У БД етапи відбору кваліфікованого персоналу за вакансією представлені таблицею `Skilledstaffdemandstage` з характеристиками:

- `id` – ідентифікатор етапу відбору кваліфікованого персоналу за вакансією;
- `skilledstaffdemand` – вакансія для кваліфікованого персоналу, за якою відбувається цей етап відбору (зовнішній ключ);
- `stagenumber` – номер етапу відбору кваліфікованого персоналу за вакансією;
- `stagesincetime` – час початку відбору кваліфікованого персоналу за вакансією;
- `stagetotime` – час завершення відбору кваліфікованого персоналу за вакансією;
- `demandstagestatus` – стан відбору кваліфікованого персоналу за вакансією.

У БД користувачі, які здійснюють пошук вакансій, (користувачі-кандидати) представлені таблицею `Skilledstaffaccount` з характеристиками:

- `id` – ідентифікатор користувача, який здійснює пошук вакансії;
- `userindjango` – обліковий запис користувача, який здійснює пошук вакансії, у Django (зовнішній ключ);
- `staffphone` – номер телефону користувача, який здійснює пошук вакансії;
- `staffphoto` – фотографія користувача, який здійснює пошук вакансії;
- `createdbyrecruiter` – позначка про те, що це не обліковий запис, а заведена сторінка рекрутинговою компанією (зовнішній ключ).

У БД анкети користувача, який здійснює пошук вакансії, за професіями представлені таблицею `Skilledstaffposition` з характеристиками:

- `id` – ідентифікатор анкети користувача, який здійснює пошук вакансії, за професією;
- `skilledstaffaccount` – обліковий запис користувача, який здійснює пошук вакансії і створив цю анкету за професією (зовнішній ключ);

- staffprofession – професія, якої стосується анкета користувача, який здійснює пошук вакансії (зовнішній ключ);
- skilledstaffpositiontext – текст, який супроводжує цю анкету користувача, який здійснює пошук вакансії за професією;
- skilledstaffsalary – мінімальна зарплата, на яку згоден користувач, який здійснює пошук вакансії за цією професією;
- skilledstaffattentiveness – уважність користувача, який здійснює пошук вакансії за професією;
- skilledstaffresponsibility – відповідальність користувача, який здійснює пошук вакансії за професією;
- skilledstaffteam – вміння працювати в команді користувача, який здійснює пошук вакансії за професією;
- skilledstaffpatience – терплячість користувача, який здійснює пошук вакансії за професією;
- skilledstaffevolution – бажання розвиватись користувача, який здійснює пошук вакансії за професією;
- skilledstaffcritique – критичність користувача, який здійснює пошук вакансії за професією.

У БД вміння за анкетами користувачів, які здійснюють пошук вакансій, представлені таблицею Skilledstaffpositionskill з характеристиками:

- id – ідентифікатор вміння за анкетною користувача, який здійснює пошук вакансії;
- skilledstaffposition – анкета користувача, який здійснює пошук вакансії за професією, якої стосується це вміння (зовнішній ключ);
- staffskill – вміння, яке за цією професією забезпечує цей користувач, який здійснює пошук вакансії за професією (зовнішній ключ);
- skilllevel – рівень вмінь, який забезпечує цей користувач, який здійснює пошук вакансії за професією.

У БД подані заявки кандидатів за вакансіями для кваліфікованого персоналу представлені таблицею Skilledstaffdemandposition з характеристиками:

- id – ідентифікатор поданої заявки кандидата за вакансією для кваліфікованого персоналу;
- skilledstaffposition – анкета користувача, який здійснює пошук вакансії і який подав цю заявку кандидата за вакансією (зовнішній ключ);
- skilledstaffdemand вакансія для кваліфікованого персоналу, за якою подана заявка (зовнішній ключ);
- skilledstafffile – файл, який додатково до стандартної анкети поданий користувачем у заявці кандидата за вакансією для кваліфікованого персоналу;
- skilledstaffpositionstatus – стан поданої заявки кандидата за вакансією для кваліфікованого персоналу;
- skilledstaffpositioncreated – час подання заявки кандидата за вакансією для кваліфікованого персоналу.

У БД підтвердження наданої користувачем інформації щодо себе за професією представниками компаній-працедавців представлені таблицею Skilledstaffpositionconfirm з характеристиками:

- id – ідентифікатор підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця;
- skilledstaffpositionforconfirm – анкета користувача, який здійснює пошук вакансії, за професією, за якою було підтверджено надану інформацію представником компанії-працедавця (зовнішній ключ);
- staffemployeraccount – користувач, що є представником компанії-працедавця і який підтвердив надану в анкеті інформацію (зовнішній ключ);
- staffconfirmdate – дата підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця;
- staffconfirmtype – тип підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця: правдива або неправдива інформація;

– `staffconfirmtext` – текст підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця.

Оскільки підтвердження стосуються певного стану анкети, а далі цей стан, тобто інформація в анкеті може змінюватись, то важливо зберегти ту інформацію, яка була підтверджена. У БД вміст анкети користувача, який здійснює пошук вакансії, за професією, що було підтверджено, представлений таблицею `Skilledstaffpositionforconfirm` з характеристиками:

– `id` – ідентифікатор збереженого вмісту анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця;

– `skilledstaffposition` – анкета користувача, який здійснює пошук вакансії і на основі якої здійснювалось підтвердження (зовнішній ключ);

– `skilledstaffattentiveness` – збережена уважність користувача, який здійснює пошук вакансії за професією;

– `skilledstaffresponsibility` – збережена відповідальність користувача, який здійснює пошук вакансії за професією;

– `skilledstaffteam` – збережене вміння працювати в команді користувача, який здійснює пошук вакансії за професією;

– `skilledstaffpatience` – збережена терплячість користувача, який здійснює пошук вакансії за професією;

– `skilledstaffevolution` – збережене бажання розвиватись користувача, який здійснює пошук вакансії за професією;

– `skilledstaffcritique` – збережена критичність користувача, який здійснює пошук вакансії за професією;

– `skilledstaffpositiontext` – збережений текст, який супроводжує цю анкету користувача, який здійснює пошук вакансії за професією.

У БД збережені вміння анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця представлені таблицею `Skilledstaffpositionforconfirmskill` з характеристиками:

– id – ідентифікатор збереженого вміння анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця;

– skilledstaffpositionforconfirm – збережений вміст анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця, якого стосується це вміння (зовнішній ключ);

– staffskill – збережене вміння анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця (зовнішній ключ);

– skilllevel – рівень вмінь, який було підтверджено.

У БД результати реалізації етапів відбору кваліфікованого персоналу за вакансією стосовно конкретної заявки представлені таблицею Skilledstaffdemandpositionstage з характеристиками:

– id – ідентифікатор етапу відбору кваліфікованого персоналу за вакансією стосовно конкретної заявки;

– skilledstaffdemandstage – етап відбору кваліфікованого персоналу за вакансією, якого стосується цей результат для заявки (зовнішній ключ);

– skilledstaffposition – анкета користувача, який здійснює пошук вакансії, за професією, результати відбору за якою збережені стосовно цього етапу (зовнішній ключ);

– skilledstaffvote – кількість голосів, відданих сумарно за цю анкету користувача, який здійснює пошук вакансії, за професією;

– positionstagestatus – стан проходження цього етапу відбору кваліфікованого персоналу за вакансією стосовно цієї конкретної заявки.

У БД голосування за заявками за вакансіями для кваліфікованого персоналу в межах проведення відбору представлені таблицею Skilledstaffdemandpositionstagevote з характеристиками:

– id – ідентифікатор голосування за заявкою за вакансією для кваліфікованого персоналу в межах проведення відбору;

– `skilledstaffdemandpositionstage` – етап відбору кваліфікованого персоналу за вакансією стосовно конкретної заявки, за яким відбулось голосування (зовнішній ключ);

– `staffemployeraccount` – користувач, який представляє компанію-працедавця та який віддав свій голос за цю заявку за вакансією для кваліфікованого персоналу в межах проведення відбору (зовнішній ключ);

– `staffvotetime` – час голосування за заявкою за вакансією для кваліфікованого персоналу в межах проведення відбору.

У БД перевірки вмінь заявників за вакансіями для кваліфікованого персоналу представлені таблицею `Skilledstaffdemandcheck` з характеристиками:

– `id` – ідентифікатор перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– `skilledstaffdemand` – вакансія для кваліфікованого персоналу, за якою має відбутись перевірка вмінь (зовнішній ключ);

– `skilledstaffcheckcaption` – назва перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– `skilledstaffcheckstarttime` – час, коли має бути розпочата перевірка вмінь заявника за вакансією для кваліфікованого персоналу (початок доступу);

– `skilledstaffcheckendtime` – час, коли має бути завершена перевірка вмінь заявника за вакансією для кваліфікованого персоналу (кінець доступу);

– `skilledstaffcheckrequired` – вимога обов'язкового проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу: якщо така вимога встановлена, то кандидат не переходить до наступного етапу, якщо не пройшов перевірку;

– `skilledstaffcheckquestionrandom` – випадковий порядок відтворення запитань перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– `skilledstaffdemandcheckminlevel` – кількість балів, яку мінімально потрібно набрати, щоб вважати перевірку вмінь заявника за вакансією для кваліфікованого персоналу пройденою;

– `skilledstaffdemandcheckmaxlevel` – максимальна кількість балів, яку потрібно набрати, щоб вважати перевірку вмінь заявника за вакансією для кваліфікованого персоналу пройденою.

У БД запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу представлені таблицею `Skilledstaffdemandcheckquestion` з характеристиками:

– `id` – ідентифікатор запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– `skilledstaffdemandcheck` – перевірка вмінь заявника за вакансією для кваліфікованого персоналу, якої стосується запитання (зовнішній ключ);

– `skilledstaffcheckquestiontext` – текст запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– `skilledstaffcheckquestionnumber` – номер запитання серед всіх запитань перевірки вмінь заявника за вакансією для кваліфікованого персоналу, якщо порядок не має бути випадковим;

– `skilledstaffcheckanswerrandom` – визначення того, чи порядок запитань під час перевірки вмінь заявника за вакансією для кваліфікованого персоналу має бути випадковим;

– `skilledstaffcheckquestionminlevel` – кількість балів, яку можна набрати за одну правильну відповідь для запитання (коли не один варіант відповіді є правильним);

– `skilledstaffcheckquestionmaxlevel` – кількість балів, яку можна набрати за всі правильні відповіді на запитання (в тому числі на одну, коли правильна тільки одна).

У БД відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу представлені таблицею `Skilledstaffdemandcheckquestionanswer` з характеристиками:

– id – номер відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– skilledstaffdemandcheckquestion – запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, до якого належить ця відповідь (зовнішній ключ);

– skilledstaffcheckanswertext – текст відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– skilledstaffcheckanswernumber – порядковий номер відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, коли порядок не є випадковим;

– skilledstaffcheckanswercorrect – визначення того, чи ця відповідь на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу є правильною.

У БД результати проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу представлені таблицею Skilledstaffdemandcheckposition з характеристиками:

– id – ідентифікатор результатів проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– skilledstaffdemandposition – подана заявка кандидата за вакансією для кваліфікованого персоналу, за якою отримано результати перевірки (зовнішній ключ);

– skilledstaffdemandcheck – перевірка вмінь заявника за вакансією для кваліфікованого персоналу, за якою була здійснена перевірка цієї заявки (зовнішній ключ);

– skilledstaffcheckstarttime – час початку проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– skilledstaffcheckendtime – час завершення проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу;

– skilledstaffchecklevel – отримана оцінка за результатами проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу.

У БД відповіді результатів проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу представлені таблицею Skilledstaffdemandcheckpositionquestionanswer з характеристиками:

- id – ідентифікатор відповіді результатів проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу;
- skilledstaffdemandcheckposition – результати проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу, до яких належить ця відповідь (зовнішній ключ);
- skilledstaffdemandcheckquestionanswer – відповідь на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, яка була обрана кандидатом (зовнішній ключ);
- skilledstaffchecklevel – отримана оцінка за цю відповідь на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, яка була обрана кандидатом.

У БД підписки користувачів, які здійснюють пошук вакансії, на компанії-працедавців, які потребують кваліфікованого персоналу, представлені таблицею Staffemployerfollow з характеристиками:

- id – ідентифікатор підписки користувача, який здійснює пошук вакансії, на компанію-працедавця, який потребує кваліфікованого персоналу;
- staffemployer – компанія-працедавець, яка потребує кваліфікованого персоналу і на яку підписався користувач (зовнішній ключ);
- skilledstaffaccount – користувач, який здійснює пошук вакансії і підписався на цю компанію-працедавця, що потребує кваліфікованого персоналу (зовнішній ключ).

У БД збереження вакансій для кваліфікованого персоналу користувачами, які здійснюють пошук вакансій, представлені таблицею Skilledstaffdemandsave з характеристиками:

- id – ідентифікатор збереження вакансії для кваліфікованого персоналу користувачем, який здійснює пошук вакансії;

– `skilledstaffdemand` – вакансія для кваліфікованого персоналу, яку зберіг користувач, що здійснює пошук вакансії (зовнішній ключ);

– `skilledstaffaccount` – користувач, який здійснює пошук вакансії та який зберіг цю вакансію для кваліфікованого персоналу (зовнішній ключ);

– `skilledstaffmark` – текстова позначка, визначена під час збереження вакансії для кваліфікованого персоналу користувачем, який здійснює пошук вакансії;

– `skilledstafftime` – час збереження вакансії для кваліфікованого персоналу користувачем, який здійснює пошук вакансії.

У БД дані про здобуту освіту користувачів, які здійснюють пошук вакансій, представлені таблицею `Skilledstaffaccounteducation` з характеристиками:

– `id` – ідентифікатор здобутої освіти користувача, який здійснює пошук вакансій;

– `skilledstaffaccount` – користувач, який здійснює пошук вакансій і здобув цю освіту (зовнішній ключ);

– `skilledstaffeducationlevel` – рівень здобутої освіти користувача, який здійснює пошук вакансій;

– `staffeducationalinstitution` – заклад, в якому було здобуто цю освіту користувачем, який здійснює пошук вакансій;

– `staffeducationstartyear` – рік початку здобуття освіти користувачем, який здійснює пошук вакансій;

– `staffeducationendyear` – рік завершення здобуття освіти користувачем, який здійснює пошук вакансій;

– `staffeducationdegree` – ступінь, який був отриманий користувачем, який здійснює пошук вакансій, за цією освітою.

У БД історія роботи користувача, який здійснює пошук вакансій, представлена таблицею `Skilledstaffaccountexperience` з характеристиками:

– `id` – ідентифікатор запису з історії роботи користувача, який здійснює пошук вакансій;

- staffemployer – компанія-працедавець, у якій працював користувач, який здійснює пошук вакансій (зовнішній ключ);
- skilledstaffaccount – користувач, який здійснює пошук вакансій і який має цей запис з історії роботи (зовнішній ключ);
- staffexperiencestartyear – рік початку роботи користувача, який здійснює пошук вакансій, у цій компанії;
- staffexperienceendyear – рік закінчення роботи користувача, який здійснює пошук вакансій, у цій компанії;
- staffprofession – професія, за якою в цій компанії-працедавці працював цей користувач, який здійснює пошук вакансій;
- staffexperiencedetail – додані текстом деталі роботи цього користувача, який здійснює пошук вакансій у цій компанії-працедавці (рис. 2.5).

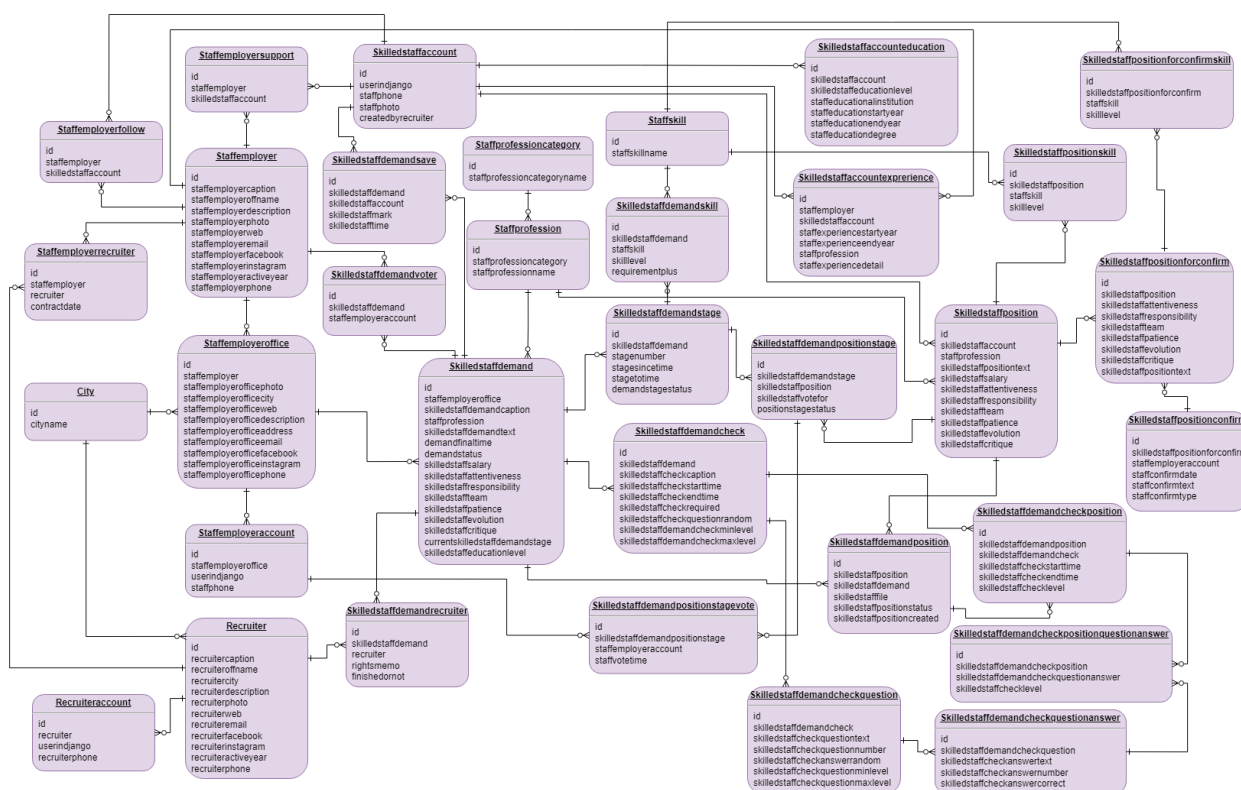


Рисунок 2.5 – Схема БД програми підбору кваліфікованого персоналу

## **2.4 Висновки за розділом 2**

Проектування програми підбору кваліфікованого персоналу було успішно завершено з визначенням функціональних вимог до програми для представника компанії-працедавця, користувача-кандидата, представника рекрутингової компанії, користувача до авторизації, з побудовою структури БД, визначенням Django і Python засобами розробки.

## 3 РОЗРОБКА ПРОГРАМИ

### 3.1 Послідовність роботи з програмою

Враховуючи розділення ролей та збереження загальних принципів роботи, було розроблено порядок роботи з програмою з впорядкуванням всіх доступних сценаріїв (рис. 3.1).

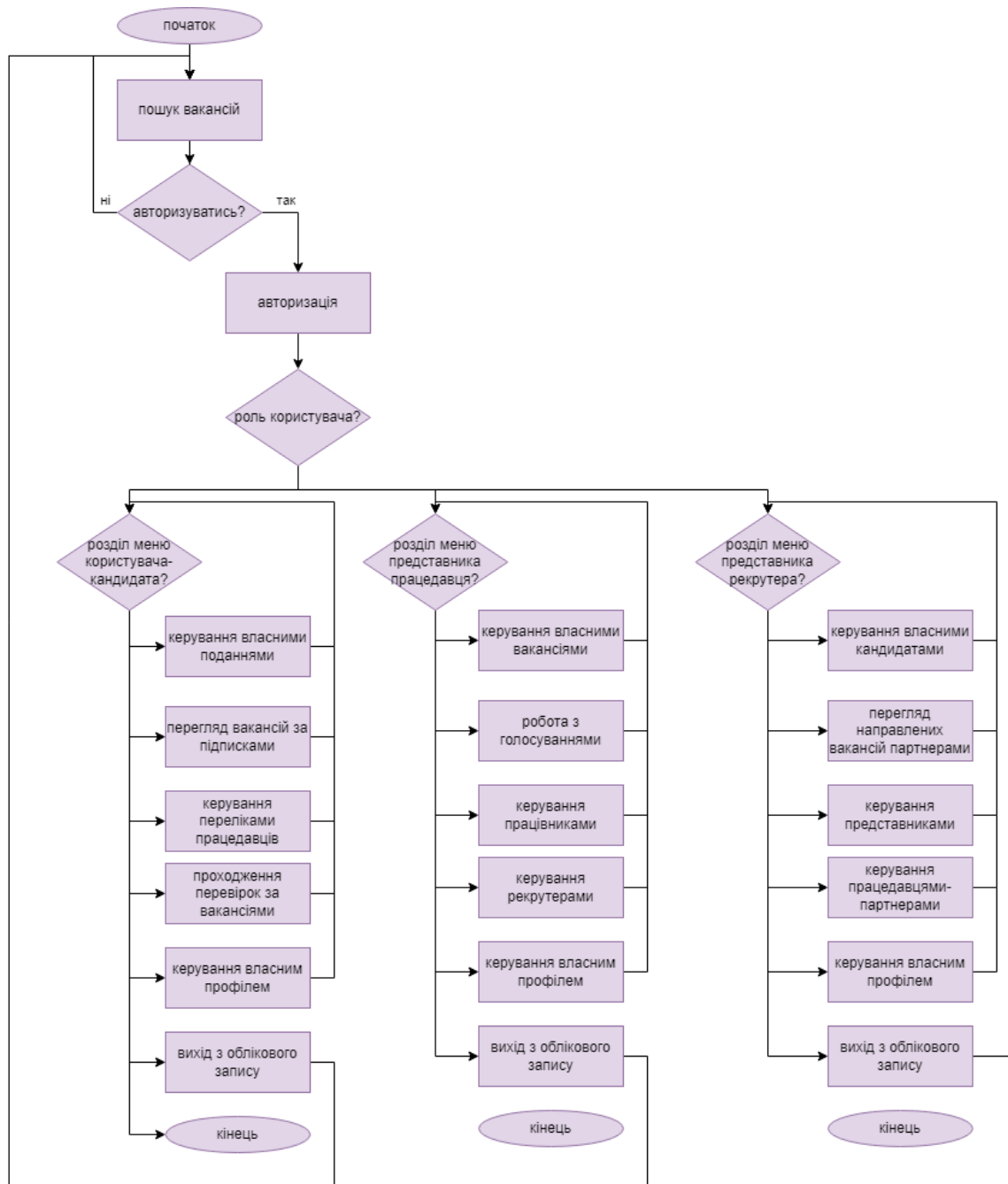


Рисунок 3.1 – Схема функціонування програми підбору кваліфікованого персоналу

### 3.2 Реалізація програми підбору кваліфікованого персоналу

Перегляд сторінки рекрутингової компанії забезпечено функцією представлення `recruiter_page_view` з параметром `recruiter_id`, який встановлює ідентифікатор рекрутингової компанії, а якщо він не заданий, то дані беруться з рекрутингової компанії, яку представляє авторизований користувач. Розробку шаблону вебсторінки під це представлення та шаблонів вебсторінок під всі наступні представлення виконано з застосуванням базового вільного шаблону [6].

Перегляд сторінки компанії-працедавця забезпечено функцією представлення `staffemployer_page_view` з параметром `staffemployer_id`, який встановлює ідентифікатор компанії-працедавця.

Перегляд компаній-працедавців рекрутингової компанії забезпечено функцією представлення `recruiter_employers_view` з параметром `recruiter_id`, що задає ідентифікатор рекрутингової компанії, який може не використовуватись для представника цієї рекрутингової компанії у випадку авторизації попередньо.

Перегляд рекрутингових компаній компанії-працедавця забезпечено функцією представлення `staffemployer_recruiters_view` з параметром `staffemployer_id` що задає ідентифікатор компанії-працедавця, який може не використовуватись для представника цієї компанії-працедавця у випадку авторизації попередньо.

Перегляд підтриманих компаній-працедавців користувачем-кандидатом забезпечено функцією представлення `staffemployer_supported_view`.

Перегляд сторінки офісу-представництва компанії-працедавця забезпечено функцією представлення `staffemployer_office_page_view` з параметром `staffemployeroffice_id`, який встановлює ідентифікатор офісу-представництва компанії-працедавця.

Перегляд відкритих вакансій офісу-представництва компанії-

працедавця забезпечено функцією представлення `staffemployer_office_demands_view` з параметром `staffemployeroffice_id`, який задає ідентифікатор офісу-представництва компанії-працедавця.

Додавання даних освіти користувача-кандидата забезпечено функцією представлення `skilledstaffaccount_education_add_view` за допомогою спеціально створеного класу форми `SkilledstaffaccounteducationForm`.

Оновлення даних освіти користувача-кандидата забезпечено функцією представлення `skilledstaffaccount_education_edit_view` з параметром `skillededucation_id`, який задає ідентифікатор даних освіти користувача-кандидата, які на основі форми `SkilledstaffaccounteducationForm` треба змінити.

Вилучення даних освіти користувача-кандидата забезпечено функцією представлення `skilledstaffaccount_education_delete_view` з параметром `skillededucation_id`, який задає ідентифікатор даних освіти користувача-кандидата.

Додавання історії роботи користувача-кандидата забезпечено функцією представлення `skilledstaffaccount_experience_add_view` за допомогою спеціально створеного класу форми `SkilledstaffaccountexpreienceForm`.

Оновлення історії роботи користувача-кандидата забезпечено функцією представлення `skilledstaffaccount_experience_edit_view` з параметром `skillexperience_id`, який задає ідентифікатор даних історії роботи користувача-кандидата, які на основі форми `SkilledstaffaccountexpreienceForm` треба змінити.

Вилучення історії роботи користувача-кандидата забезпечено функцією представлення `skilledstaffaccount_experience_delete_view` з параметром `skillexperience_id`, який задає ідентифікатор даних історії роботи користувача-кандидата.

Перегляд сторінки користувача-кандидата забезпечено функцією представлення `skilledstaff_page_view` з параметром `staff_id`, який задає ідентифікатор користувача-кандидата.

Створення етапу відбору за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_stage_add_view` за допомогою спеціально створеного класу форми `SkilledstaffdemandstageForm`.

Оновлення етапу відбору за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_stage_edit_view` з параметром `skilledstaffdemandstage_id`, який задає ідентифікатор етапу відбору за вакансією для кваліфікованого персоналу, який на основі форми `SkilledstaffdemandstageForm` треба змінити.

Перегляд доступних для голосування вакансій для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemands_voting_view`.

Перегляд доступних для голосування анкет кандидатів за обраною вакансією забезпечено функцією представлення `skilledstaffdemand_propositions_voting_view` з параметром `skilledstaffdemand_id`, який задає ідентифікатор вакансії.

Перегляд анкет кандидатів за вакансіями, за які проголосував користувач, забезпечено функцією представлення `skilledstaffdemand_propositions_votes_view`.

Голосування за анкету кандидата за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemandposition_stage_vote_view` з параметром `skilledstaffdemandpositionstage_id`, що задає ідентифікатор етапу вибору кваліфікованого персоналу за вакансією. Якщо голосування було виконано цим користувачем, то реалізується вилучення голосування за анкету кандидата за вакансією для кваліфікованого персоналу.

Встановлення підтримки компанії-працедавцю забезпечено функцією представлення `staffemployer_support_view` з параметром `staffemployer_id`, що задає ідентифікатор компанії-працедавця. Вилучення підтримки компанії-працедавцю забезпечується тією ж функцією, якщо підтримка цим користувачем була визначена цій компанії-працедавцю раніше.

Визначення контакту з рекрутинговою компанією компанії-працедавця забезпечено функцією представлення `staffemployer_recruiter_view` з параметрами `recruiter_id`, що задає ідентифікатор рекрутингової компанії, `staffemployer_id`, що задає ідентифікатор компанії-працедавця (може бути пропущений, тоді за даними авторизації визначається), `contractdate`, що задає дату встановлення контакту. Якщо такий контакт встановлено раніше, то реалізується скасування контакту з рекрутинговою компанією компанії-працедавця.

Перегляд вакансії для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_view` з параметром `skilledstaffdemand_id`, що задає ідентифікатор вакансії для кваліфікованого персоналу.

Створення вакансії для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_add_view` за допомогою класів форм `SkilledstaffdemandForm`, `SkilledstaffdemandskillForm`, `SkilledstaffdemandrecruiterForm`. Оновлення вакансії для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_edit_view`, яка використовує ці ж форми, але з параметром `skilledstaffdemand_id`, що задає ідентифікатор вакансії для кваліфікованого персоналу.

Визначення рекрутингової компанії за вакансією компанії-працедавця забезпечено функцією представлення `skilledstaffdemand_recruiter_connect_view` з параметрами `recruiter_id`, що задає ідентифікатор рекрутингової компанії, `skilledstaffdemand_id`, що задає ідентифікатор вакансії компанії-працедавця, за допомогою класу форми `SkilledstaffdemandrecruiterForm`.

Видалення рекрутингової компанії за вакансією компанії-працедавця забезпечено функцією представлення `skilledstaffdemand_recruiter_delete_view` з параметрами `recruiter_id`, що задає ідентифікатор рекрутингової компанії, `skilledstaffdemand_id`, що задає ідентифікатор вакансії компанії-працедавця.

Перегляд вакансій для кваліфікованого персоналу, визначених для

обробки рекрутинговою компанією, забезпечено функцією представлення `recruit_to_skilledstaff_demands_view`.

Створення анкети за професією забезпечено функцією представлення `skilledstaffposition_add_view` головним чином за допомогою класу форми `SkilledstaffpositionForm`.

Оновлення даних анкети за професією забезпечено функцією представлення `skilledstaffposition_edit_view` з параметром `skilledstaffposition_id`, що задає ідентифікатор анкети користувача-кандидата за професією.

Вилучення анкети за професією забезпечено функцією представлення `skilledstaffposition_delete_view` з параметром `skilledstaffposition_id`, що задає ідентифікатор анкети користувача-кандидата за професією.

Пошук вакансій для кваліфікованого персоналу забезпечено функцією представлення `search_demands_view`, у яку параметри передаються ключами запиту, включаючи `city` – місто роботи, `staffprofession_id` – ідентифікатор професії, `salarylevel` – мінімальний рівень зарплати, `skills` – розділений комами перелік ідентифікаторів вмінь тощо.

Визначення завершення оброблення вакансії рекрутинговою компанією забезпечено функцією представлення `recruit_skilledstaffdemand_finish_view` з параметром `skilledstaffdemandrecruiter`, що задає ідентифікатор відповідальності рекрутингової компанії за вакансію для кваліфікованого персоналу.

Подання анкети за професією на вакансію для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_position_set_view` з параметрами `skilledstaffposition_id`, що задає ідентифікатор анкети, `skilledstaffdemand_id`, що задає ідентифікатор вакансії, та `skilledstafffile`, що задає доданий файл.

Вилучення анкети за професією на вакансію для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_position_delete_view` з параметром

`skilledstaffdemandposition_id`, що задає ідентифікатор подання анкети за професією на вакансію для кваліфікованого персоналу.

Перегляд користувачів-кандидатів за даними рекрутингової компанії забезпечено функцією представлення `recruit_positions_view`.

Перегляд власних поданих анкет за професією на вакансії для кваліфікованого персоналу забезпечено функцією представлення `skilledstaff_own_positions_view`.

Підтвердження наданої користувачем інформації щодо себе за професією компанії-працедавцем забезпечено функцією представлення `skilledstaffposition_confirm_view` з параметром `skilledstaffposition_id`, що задає ідентифікатор подання анкети за професією на вакансію для кваліфікованого персоналу.

Вилучення підтвердження наданої користувачем інформації щодо себе за професією компанії-працедавцем забезпечено функцією представлення `skilledstaffposition_deconfirm_view` з параметром `skilledstaffpositionconfirm_id`, що задає ідентифікатор підтвердження.

Перегляд користувачів, які працювали в компанії-працедавці забезпечено функцією представлення `employer_workers_view`.

Створення перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemandcheck_add_view` за допомогою класу форми `SkilledstaffdemandcheckForm`.

Оновлення параметрів перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemandcheck_edit_view` з параметром `skilledstaffdemandcheck_id`, що задає ідентифікатор перевірки за вакансією для кваліфікованого персоналу.

Вилучення перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemandcheck_delete_view` з параметром `skilledstaffdemandcheck_id`, що задає ідентифікатор перевірки за вакансією для кваліфікованого персоналу.

Перегляд призначених перевірок за вакансіями для користувача

забезпечено функцією представлення `skilledstaffdemandchecks_to_view`.

Керування анкетою за професією на вакансію для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffposition_manage_view` з параметрами `skilledstaffposition_id`, що задає ідентифікатор анкети користувача-кандидата за професією, `posstatus`, що задає потрібний стан у результаті.

Перегляд анкети за професією на вакансію для кваліфікованого персоналу `skilledstaffposition_see_view` з параметром `skilledstaffposition_id`, що задає ідентифікатор анкети користувача-кандидата за професією.

Перегляд збережених вакансій для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_saves_view`.

Перегляд вакансій для кваліфікованого персоналу компаній-працедавців з підписок забезпечено функцією представлення `follow_skilledstaff_demands_view`.

Перегляд сторінки перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_check_get_view` з параметром `skilledstaffdemandcheck_id`, що задає ідентифікатор цієї перевірки за вакансією для кваліфікованого персоналу.

Збереження результатів перевірки користувача за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemandcheck_position_add_view` з параметрами `skilledstaffdemandposition_id`, що задає ідентифікатор поданої заявки за вакансією, та `skilledstaffdemandcheck_id`, що задає ідентифікатор перевірки.

Збереження вакансії для кваліфікованого персоналу у свій перелік забезпечено функцією представлення `skilledstaffdemand_save_view` з параметрами `skilledstaffdemand_id`, що задає ідентифікатор вакансії для кваліфікованого персоналу, `skilledstaffmark`, що задає позначку до збереження. Якщо таке збереження цієї вакансії для кваліфікованого персоналу в перелік користувача-кандидата було зроблено, то відбувається вилучення збереження

вакансії для кваліфікованого персоналу зі свого переліку.

Оформлення підписки на компанію-працедавця забезпечено функцією представлення `staffemployer_follow_view` з параметром `staffemployer_id`, що задає ідентифікатор компанії-працедавця. Якщо така підписка на цю компанію-працедавця цим користувачем-кандидатом була оформлена, то відбувається вилучення підписки на компанію-працедавця.

Внесення даних компанії-працедавця забезпечено функцією представлення `staffemployer_set_view` на основі класу форми `StaffemployerForm`.

Внесення даних офісу-представництва компанії-працедавця забезпечено функцією представлення `staffemployeroffice_set_view` на основі класу форми `StaffemployerofficeForm`.

Внесення даних рекрутингової компанії забезпечено функцією представлення `recruiter_set_view` на основі класу форми `RecruiterForm`.

Внесення даних користувача-кандидата забезпечено функцією представлення `recruiteraccount_set_view` на основі класу форми `RecruiteraccountForm`.

Додавання запитання до перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_question_add_view` за допомогою спеціально створеного класу форми `SkilledstaffdemandcheckquestionForm`.

Вилучення запитання до перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_question_delete_view` з параметром `skillcheckquestion_id`, який задає ідентифікатор запитання до перевірки за вакансією для кваліфікованого персоналу.

Оновлення запитання до перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_question_edit_view` з параметром `skillcheckquestion_id`, який задає ідентифікатор запитання до перевірки за вакансією для кваліфікованого

персоналу, яке на основі форми `SkilledstaffdemandcheckquestionForm` треба змінити.

Додавання відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_answer_add_view` за допомогою спеціально створеного класу форми `SkilledstaffdemandcheckquestionanswerForm` з параметром `skilledstaffdemandcheckque_id`, який задає ідентифікатор запитання до перевірки за вакансією для кваліфікованого персоналу, до якого додається відповідь.

Вилучення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_answer_delete_view` з параметром `skillcheckanswer_id`, який задає ідентифікатор відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу.

Оновлення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу забезпечено функцією представлення `skilledstaffdemand_answer_edit_view` з параметром `skillcheckanswer_id`, який задає ідентифікатор відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу, яку на основі форми `SkilledstaffdemandcheckquestionanswerForm` треба змінити.

Для роботи з даними у програмі необхідно було реалізувати класи моделей, оскільки вони, по-перше, визначають самі сутності предметної області, а по-друге, формують відповідну структуру БД на основі створеного коду.

Модель компанії-працедавця, яка потребує кваліфікованого персоналу, є класом `Staffemployer`.

Клас `Staffemployer` містить поле `staffemployercaption`, яким визначається коротка презентаційна назва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `CharField`.

Клас `Staffemployer` містить поле `staffemployeroffname`, яким визначається офіційна повна назва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `CharField`.

Клас `Staffemployer` містить поле `staffemployerdescription`, яким визначається текстовий опис компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `TextField`.

Клас `Staffemployer` містить поле `staffemployerphoto`, яким визначається зображення для представлення компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `ImageField`.

Клас `Staffemployer` містить поле `staffemployerweb`, яким визначається вебсайт компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `URLField`.

Клас `Staffemployer` містить поле `staffemployeremail`, яким визначається електронна адреса компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `CharField`.

Клас `Staffemployer` містить поле `staffemployerfacebook`, яким визначається Facebook-сторінка компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `URLField`.

Клас `Staffemployer` містить поле `staffemployerinstagram`, яким визначається Instagram-сторінка компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `URLField`.

Клас `Staffemployer` містить поле `staffemployeractiveyear`, яким визначається рік початку роботи компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `DateField`.

Клас `Staffemployer` містить поле `staffemployerphone`, яким визначається номер телефону компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `CharField`.

Клас `Staffemployer` містить поле `skilledstaffsupports`, яким визначаються підтримування компанії-працедавця користувачами-кандидатами у

кваліфікований персонал, за реалізацією зв'язку між моделями типу багато-до-багатьох за типом ManyToManyField до моделі Skilledstaffaccount.

Модель міста представлений є класом City.

Клас City містить поле cityname, яким визначається назва міста, за типом CharField.

Модель офісів-представництв компаній-працедавців, які потребують кваліфікованого персоналу, є класом Staffemployeroffice.

Клас Staffemployeroffice містить поле staffemployer, яким визначається компанія-працедавець, що потребує кваліфікованого персоналу та яку представляє цей офіс, за типом ForeignKey.

Клас Staffemployeroffice містить поле staffemployerofficephoto, яким визначається зображення, що представляє офіс компанії-працедавця, що потребує кваліфікованого персоналу, за типом ImageField.

Клас Staffemployeroffice містить поле staffemployerofficacity, яким визначається місто, в якому знаходиться цей офіс-представництво компанії-працедавця, яка потребує кваліфікованого персоналу, за типом ForeignKey.

Клас Staffemployeroffice містить поле staffemployerofficeweb, яким визначається вебсайт офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом URLField.

Клас Staffemployeroffice містить поле staffemployerofficedescription, яким визначається текстовий опис офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом TextField.

Клас Staffemployeroffice містить поле staffemployerofficeaddress, яким визначається адреса офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом CharField.

Клас Staffemployeroffice містить поле staffemployerofficeemail, яким визначається електронна пошта офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом CharField.

Клас `Staffemployeroffice` містить поле `staffemployerofficefacebook`, яким визначається Facebook-сторінка офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `URLField`.

Клас `Staffemployeroffice` містить поле `staffemployerofficeinstagram`, яким визначається Instagram-сторінка офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `URLField`.

Клас `Staffemployeroffice` містить поле `staffemployerofficephone`, яким визначається номер телефону офісу-представництва компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `CharField`.

Модель облікових записів користувачів-представників офісів компаній-працедавців, які потребують кваліфікованого персоналу, є класом `Staffemployeraccount`.

Клас `Staffemployeraccount` містить поле `staffemployeroffice`, яким визначається офіс компанії-працедавця, яка потребує кваліфікованого персоналу та яку представляє користувач, за типом `ForeignKey`.

Клас `Staffemployeraccount` містить поле `userindjango`, яким визначається обліковий запис користувача, який представляє офіс компанії-працедавця, яка потребує кваліфікованого персоналу, в Django, за типом `OneToOneField` до моделі `User`, базової моделі будь-якого Django-застосунку.

Клас `Staffemployeraccount` містить поле `staffphone`, яким визначається номер телефону користувача, який представляє офіс компанії-працедавця, яка потребує кваліфікованого персоналу, за типом `CharField`.

Модель рекрутингових компаній є класом `Recruiter`.

Клас `Recruiter` містить поле `recruitercaption`, яким визначається коротка презентаційна назва рекрутингової компанії, за типом `CharField`.

Клас `Recruiter` містить поле `recruiteroffname`, яким визначається офіційна повна назва рекрутингової компанії, за типом `CharField`.

Клас `Recruiter` містить поле `recruitercity`, яким визначається місто, в якому знаходиться рекрутингова компанія, за типом `ForeignKey`.

Клас Recruiter містить поле recruiterdescription, яким визначається текстовий опис рекрутингової компанії, за типом TextField.

Клас Recruiter містить поле recruiterphoto, яким визначається зображення, що представляє рекрутингову компанію, за типом ImageField.

Клас Recruiter містить поле recruiterweb, яким визначається вебсайт рекрутингової компанії, за типом URLField.

Клас Recruiter містить поле recruiteremail, яким визначається електронна адреса рекрутингової компанії, за типом CharField.

Клас Recruiter містить поле recruiterfacebook, яким визначається Facebook-сторінка рекрутингової компанії, за типом URLField.

Клас Recruiter містить поле recruiterinstagram, яким визначається Instagram-сторінка рекрутингової компанії, за типом URLField.

Клас Recruiter містить поле recruiteractiveyear, яким визначається рік початку діяльності рекрутингової компанії, за типом DateField.

Клас Recruiter містить поле recruiterphone, яким визначається номер телефону рекрутингової компанії, за типом CharField.

Модель облікових записів користувачів, які представляють рекрутингові компанії, є класом Recruiteraccount.

Клас Recruiteraccount містить поле recruiter, яким визначається рекрутингова компанія, яку представляє користувач, за типом ForeignKey.

Клас Recruiteraccount містить поле userindjango, яким визначається обліковий запис користувача, який представляє рекрутингову компанію, в Django, за типом OneToOneField.

Клас Recruiteraccount містить поле recruiterphone, яким визначається номер телефону користувача, який представляє рекрутингову компанію, за типом CharField.

Модель встановлених контактів на обслуговування між рекрутинговими компаніями та компаніями-працедавцями, які потребують кваліфікованого персоналу, є класом Staffemployerrecruiter,

Клас `Staffemployerrecruiter` містить поле `staffemployer`, яким визначається компанія-працедавець, яка потребує кваліфікованого персоналу та має контакт на обслуговування з цією рекрутинговою компанією, за типом `ForeignKey`.

Клас `Staffemployerrecruiter` містить поле `recruiter`, яким визначається рекрутингова компанія, яка має контакт на обслуговування з цією компанією-працедавцем, яка потребує кваліфікованого персоналу, за типом `ForeignKey`.

Клас `Staffemployerrecruiter` містить поле `contractdate`, яким визначається дата встановлення контакту між рекрутинговою компанією та компанією-працедавцем, яка потребує кваліфікованого персоналу, за типом `DateField`.

Модель категорії професії кваліфікованого персоналу є класом `Staffprofessioncategory`.

Клас `Staffprofessioncategory` містить поле `staffprofessioncategoryname`, яким визначається назва категорії професії кваліфікованого персоналу, за типом `CharField`.

Модель професії кваліфікованого персоналу є класом `Staffprofession`.

Клас `Staffprofession` містить поле `staffprofessioncategory`, яким визначається категорія цієї професії кваліфікованого персоналу, за типом `ForeignKey`.

Клас `Staffprofession` містить поле `staffprofessionname`, яким визначається назва професії кваліфікованого персоналу, за типом `CharField`.

Модель навичок, які можуть визначатися для кваліфікованого персоналу, є класом `Staffskill`.

Клас `Staffskill` містить поле `staffskillname`, яким визначається назва навички, яка може визначатися для кваліфікованого персоналу, за типом `CharField`.

Модель вакансії для кваліфікованого персоналу в компаніях-працедавцях є класом `Skilledstaffdemand`.

Клас `Skilledstaffdemand` містить поле `staffemployeroffice`, яким визначається компанія-працедавець, яка висунула цю вакансію для кваліфікованого персоналу, за типом `ForeignKey`.

Клас `Skilledstaffdemand` містить поле `skilledstaffdemandcaption`, яким визначається коротка назва вакансії для кваліфікованого персоналу, за типом `CharField`.

Клас `Skilledstaffdemand` містить поле `staffprofession`, яким визначається професія, кваліфікованого персоналу за якою потребує ця компанія-працедавець, за типом `ForeignKey`.

Клас `Skilledstaffdemand` містить поле `skilledstaffdemandtext`, яким визначається текстові вимоги до кваліфікованого працівника від компанії-працедавця, за типом `TextField`.

Клас `Skilledstaffdemand` містить поле `demandfinaltime`, яким визначається остання дата, до якої може бути відкрита ця вакансія для кваліфікованого персоналу, за типом `DateField`.

Клас `Skilledstaffdemand` містить поле `demandstatus`, яким визначається поточний стан цієї вакансії для кваліфікованого персоналу, за типом `PositiveSmallIntegerField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffsalary`, яким визначається заробітна плата, встановлена у вакансії для кваліфікованого персоналу, за типом `FloatField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffattentiveness`, яким визначається наявність вимоги до уважності кваліфікованого працівника за цією вакансією, за типом `BooleanField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffresponsibility`, яким визначається наявність відповідальності до уважності кваліфікованого працівника за цією вакансією, за типом `BooleanField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffteam`, яким визначається наявність вимоги до вміння працювати в команді кваліфікованого працівника за цією вакансією, за типом `BooleanField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffpatience`, яким визначається наявність вимоги до терплячості кваліфікованого працівника за цією вакансією, за типом `BooleanField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffevolution`, яким визначається наявність вимоги до бажання розвиватись кваліфікованого працівника за цією вакансією, за типом `BooleanField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffcritique`, яким визначається наявність вимоги до критичності кваліфікованого працівника за цією вакансією, за типом `BooleanField`.

Клас `Skilledstaffdemand` містить поле `currentskilledstaffdemandstage`, яким визначається поточний етап відбору кваліфікованого персоналу за цією вакансією за типом `ForeignKey`.

Клас `Skilledstaffdemand` містить поле `skilledstaffeducationlevel`, яким визначається потрібний рівень освіти кандидата, за типом `PositiveSmallIntegerField`.

Клас `Skilledstaffdemand` містить поле `skilledstaffdemandvoters`, яким визначаються користувачі, які мають право голосувати під час відбору кваліфікованого персоналу за цією вакансією компанії-працедавця, за типом `ManyToManyField`.

Модель вимог до вмінь за вакансією для кваліфікованого персоналу є класом `Skilledstaffdemandskill`.

Клас `Skilledstaffdemandskill` містить поле `skilledstaffdemand`, яким визначається вакансія для кваліфікованого персоналу, за якою є вимога за цим вмінням, за типом `ForeignKey`.

Клас `Skilledstaffdemandskill` містить поле `staffskill`, яким визначається вміння, за яким висунуто вимогу за вакансією для кваліфікованого персоналу, за типом `ForeignKey`.

Клас `Skilledstaffdemandskill` містить поле `skilllevel`, яким визначається рівень вмінь, який має бути забезпечений за цією вакансією для кваліфікованого персоналу, за типом `PositiveSmallIntegerField`.

Клас `Skilledstaffdemandskill` містить поле `requirementplus`, яким визначається обов'язковість вимоги щодо цього вміння за цією вакансією для кваліфікованого персоналу, за типом `BooleanField`.

Модель відповідальності рекрутингових компаній за пошук кваліфікованого персоналу є класом `Skilledstaffdemandrecruiter`.

Клас `Skilledstaffdemandrecruiter` містить поле `skilledstaffdemand`, яким визначається вакансія для кваліфікованого персоналу, за якою встановлено відповідальність рекрутингової компанії, за типом `ForeignKey`.

Клас `Skilledstaffdemandrecruiter` містить поле `recruiter`, яким визначається рекрутингова компанія, за якою встановлено відповідальність за вакансією для кваліфікованого персоналу, за типом `ForeignKey`.

Клас `Skilledstaffdemandrecruiter` містить поле `rightsmemo`, яким визначається нагадування щодо особливостей вимог до відповідальності рекрутингової компанії за пошук кваліфікованого персоналу за вакансією, за типом `CharField`.

Клас `Skilledstaffdemandrecruiter` містить поле `finishedornot`, яким визначається завершеність виконання пошуку кваліфікованого персоналу за вакансією рекрутинговою компанією, за типом `BooleanField`.

Модель етапів відбору кваліфікованого персоналу за вакансією є класом `Skilledstaffdemandstage`.

Клас `Skilledstaffdemandstage` містить поле `skilledstaffdemand`, яким визначається вакансія для кваліфікованого персоналу, за якою відбувається цей етап відбору, за типом `ForeignKey`.

Клас `Skilledstaffdemandstage` містить поле `stagenumber`, яким визначається номер етапу відбору кваліфікованого персоналу за вакансією, за типом `PositiveSmallIntegerField`.

Клас `Skilledstaffdemandstage` містить поле `stagesincetime`, яким визначається час початку відбору кваліфікованого персоналу за вакансією, за типом `DateTimeField`.

Клас `Skilledstaffdemandstage` містить поле `stagetotime`, яким визначається час завершення відбору кваліфікованого персоналу за вакансією, за типом `DateTimeField`.

Клас `Skilledstaffdemandstage` містить поле `demandstagestatus`, яким визначається стан відбору кваліфікованого персоналу за вакансією, за типом `PositiveSmallIntegerField`.

Модель користувачів, які здійснюють пошук вакансій, є класом `Skilledstaffaccount`.

Клас `Skilledstaffaccount` містить поле `userindjango`, яким визначається обліковий запис користувача, який здійснює пошук вакансії, у Django, за типом `OneToOneField` зі зв'язком до моделі `User`.

Клас `Skilledstaffaccount` містить поле `staffphone`, яким визначається номер телефону користувача, який здійснює пошук вакансії, за типом `CharField`.

Клас `Skilledstaffaccount` містить поле `staffphoto`, яким визначається фотографія користувача, який здійснює пошук вакансії, за типом `ImageField`.

Клас `Skilledstaffaccount` містить поле `staffemployers`, яким визначаються підписки цього користувача, який здійснює пошук вакансії, на компанії-працедавців, за типом `ManyToManyField`.

Клас `Skilledstaffaccount` містить поле `createdbyrecruiter`, яким визначається позначка про те, що це не обліковий запис, а заведена сторінка рекрутинговою компанією, за типом `ForeignKey`.

Модель анкети користувача, який здійснює пошук вакансії, за професіями є класом `Skilledstaffposition`.

Клас `Skilledstaffposition` містить поле `skilledstaffaccount`, яким визначається обліковий запис користувача, який здійснює пошук вакансії і створив цю анкету за професією, за типом `ForeignKey`.

Клас `Skilledstaffposition` містить поле `staffprofession`, яким визначається професія, якої стосується анкета користувача, який здійснює пошук вакансії, за типом `ForeignKey`.

Клас `Skilledstaffposition` містить поле `skilledstaffpositiontext`, яким визначається текст, який супроводжує цю анкету користувача, який здійснює пошук вакансії за професією, за типом `TextField`.

Клас `Skilledstaffposition` містить поле `skilledstaffsalary`, яким визначається мінімальна зарплата, на яку згоден користувач, який здійснює пошук вакансії за цією професією, за типом `FloatField`.

Клас `Skilledstaffposition` містить поле `skilledstaffattentiveness`, яким визначається уважність користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffposition` містить поле `skilledstaffresponsibility`, яким визначається відповідальність користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffposition` містить поле `skilledstaffteam`, яким визначається вміння працювати в команді користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffposition` містить поле `skilledstaffpatience`, яким визначається терплячість користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffposition` містить поле `skilledstaffevolution`, яким визначається бажання розвиватись користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffposition` містить поле `skilledstaffcritique`, яким визначається критичність користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Модель вміння за анкетами користувачів, які здійснюють пошук вакансій, є класом `Skilledstaffpositionskill`.

Клас `Skilledstaffpositionskill` містить поле `skilledstaffposition`, яким визначається анкета користувача, який здійснює пошук вакансії за професією, якої стосується це вміння, за типом `ForeignKey`.

Клас `Skilledstaffpositionskill` містить поле `staffskill`, яким визначається вміння, яке за цією професією забезпечує цей користувач, який здійснює пошук вакансії за професією, за типом `ForeignKey`.

Клас `Skilledstaffpositionskill` містить поле `skilllevel`, яким визначається рівень вмінь, який забезпечує цей користувач, який здійснює пошук вакансії за професією, за типом `PositiveSmallIntegerField`.

Модель поданих заявок кандидатів за вакансіями для кваліфікованого персоналу є класом `Skilledstaffdemandposition`.

Клас `Skilledstaffdemandposition` містить поле `skilledstaffposition`, яким визначається анкета користувача, який здійснює пошук вакансії і який подав цю заявку кандидата за вакансією, за типом `ForeignKey`.

Клас `Skilledstaffdemandposition` містить поле `skilledstaffdemand`, яким визначається вакансія для кваліфікованого персоналу, за якою подана заявка, за типом `ForeignKey`.

Клас `Skilledstaffdemandposition` містить поле `skilledstafffile`, яким визначається файл, який додатково до стандартної анкети поданий користувачем у заявці кандидата за вакансією для кваліфікованого персоналу, за типом `FileField`.

Клас `Skilledstaffdemandposition` містить поле `skilledstaffpositionstatus`, яким визначається стан поданої заявки кандидата за вакансією для кваліфікованого персоналу, за типом `PositiveSmallIntegerField`.

Клас `Skilledstaffdemandposition` містить поле `skilledstaffpositioncreated`, яким визначається час подання заявки кандидата за вакансією для кваліфікованого персоналу, за типом `DateTimeField`.

Модель підтвердження наданої користувачем інформації щодо себе за професією представниками компаній-працедавців є класом `Skilledstaffpositionconfirm`.

Клас `Skilledstaffpositionconfirm` містить поле `skilledstaffpositionforconfirm`, яким визначається анкета користувача, який

здійснює пошук вакансії, за професією, за якою було підтверджено надану інформацію представником компанії-працедавця, за типом `ForeignKey`.

Клас `Skilledstaffpositionconfirm` містить поле `staffemployeraccount`, яким визначається користувач, що є представником компанії-працедавця і який підтвердив надану в анкеті інформацію, за типом `ForeignKey`.

Клас `Skilledstaffpositionconfirm` містить поле `staffconfirmdate`, яким визначається дата підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця, за типом `DateField`.

Клас `Skilledstaffpositionconfirm` містить поле `staffconfirmtype`, яким визначається тип підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця, за типом `BooleanField`.

Клас `Skilledstaffpositionconfirm` містить поле `staffconfirmtext`, яким визначається текст підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця, за типом `CharField`.

Модель вмісту анкети користувача, який здійснює пошук вакансії, за професією, що було підтверджено, є класом `Skilledstaffpositionforconfirm`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffposition`, яким визначається анкета користувача, який здійснює пошук вакансії і на основі якої здійснювалось підтвердження, за типом `ForeignKey`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffattentiveness`, яким визначається збережена уважність користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffresponsibility`, яким визначається збережена відповідальність користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffteam`, яким визначається збережене вміння працювати в команді користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffpatience`, яким визначається збережена терплячість користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffevolution`, яким визначається збережене бажання розвиватись користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffcritique`, яким визначається збережена критичність користувача, який здійснює пошук вакансії за професією, за типом `BooleanField`.

Клас `Skilledstaffpositionforconfirm` містить поле `skilledstaffpositiontext`, яким визначається збережений текст, який супроводжує цю анкету користувача, який здійснює пошук вакансії за професією, за типом `TextField`.

Модель збережених вмінь анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця є класом `Skilledstaffpositionforconfirmskill`.

Клас `Skilledstaffpositionforconfirmskill` містить поле `skilledstaffpositionforconfirm`, яким визначається збережений вміст анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця, якого стосується це вміння, за типом `ForeignKey`.

Клас `Skilledstaffpositionforconfirmskill` містить поле `staffskill`, яким визначається збережене вміння анкети для підтвердження наданої користувачем інформації щодо себе за професією представником компанії-працедавця, за типом `ForeignKey`.

Клас `Skilledstaffpositionforconfirmskill` містить поле `skilllevel`, яким визначається рівень вмінь, який було підтверджено, за типом `PositiveSmallIntegerField`.

Модель результатів реалізації етапів відбору кваліфікованого персоналу за вакансією стосовно конкретної заявки є класом `Skilledstaffdemandpositionstage`.

Клас `Skilledstaffdemandpositionstage` містить поле `skilledstaffdemandstage`, яким визначається етап відбору кваліфікованого персоналу за вакансією, якого стосується цей результат для заявки, за типом `ForeignKey`.

Клас `Skilledstaffdemandpositionstage` містить поле `skilledstaffposition`, яким визначається анкета користувача, який здійснює пошук вакансії, за професією, результати відбору за якою збережені стосовно цього етапу, за типом `ForeignKey`.

Клас `Skilledstaffdemandpositionstage` містить поле `skilledstaffvote`, яким визначається кількість голосів, відданих сумарно за цю анкету користувача, який здійснює пошук вакансії, за професією, за типом `PositiveIntegerField`.

Клас `Skilledstaffdemandpositionstage` містить поле `positionstagestatus`, яким визначається стан проходження цього етапу відбору кваліфікованого персоналу за вакансією стосовно цієї конкретної заявки, за типом `PositiveIntegerField`.

Модель голосування за заявками за вакансіями для кваліфікованого персоналу в межах проведення відбору є класом `Skilledstaffdemandpositionstagevote`.

Клас `Skilledstaffdemandpositionstagevote` містить поле `skilledstaffdemandpositionstage`, яким визначається етап відбору кваліфікованого персоналу за вакансією стосовно конкретної заявки, за яким відбулось голосування, за типом `ForeignKey`.

Клас `Skilledstaffdemandpositionstagevote` містить поле `staffemployeraccount`, яким визначається користувач, який представляє компанію-працедавця та який віддав свій голос за цю заявку за вакансією для кваліфікованого персоналу в межах проведення відбору, за типом `ForeignKey`.

Клас `Skilledstaffdemandpositionstagevote` містить поле `staffvotetime`, яким визначається час голосування за заявкою за вакансією для

кваліфікованого персоналу в межах проведення відбору, за типом DateTimeField.

Модель перевірки вмінь заявників за вакансіями для кваліфікованого персоналу є класом Skilledstaffdemandcheck.

Клас Skilledstaffdemandcheck містить поле skilledstaffdemand, яким визначається вакансія для кваліфікованого персоналу, за якою має відбутись перевірка вмінь, за типом ForeignKey.

Клас Skilledstaffdemandcheck містить поле skilledstaffcheckcaption, яким визначається назва перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом CharField.

Клас Skilledstaffdemandcheck містить поле skilledstaffcheckstarttime, яким визначається час, коли має бути розпочата перевірка вмінь заявника за вакансією для кваліфікованого персоналу, за типом DateTimeField.

Клас Skilledstaffdemandcheck містить поле skilledstaffcheckendtime, яким визначається час, коли має бути завершена перевірка вмінь заявника за вакансією для кваліфікованого персоналу, за типом DateTimeField.

Клас Skilledstaffdemandcheck містить поле skilledstaffcheckrequired, яким визначається вимога обов'язкового проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом BooleanField.

Клас Skilledstaffdemandcheck містить поле skilledstaffcheckquestionrandom, яким визначається випадковий порядок відтворення запитань перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом BooleanField.

Клас Skilledstaffdemandcheck містить поле skilledstaffdemandcheckminlevel, яким визначається кількість балів, яку мінімально потрібно набрати, щоб вважати перевірку вмінь заявника за вакансією для кваліфікованого персоналу пройденою, за типом FloatField.

Клас Skilledstaffdemandcheck містить поле skilledstaffdemandcheckmaxlevel, яким визначається максимальна кількість

балів, яку потрібно набрати, щоб вважати перевірку вмінь заявника за вакансією для кваліфікованого персоналу пройденою, за типом FloatField.

Модель запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу є класом Skilledstaffdemandcheckquestion.

Клас Skilledstaffdemandcheckquestion містить поле skilledstaffdemandcheck, яким визначається перевірка вмінь заявника за вакансією для кваліфікованого персоналу, якої стосується запитання, за типом ForeignKey.

Клас Skilledstaffdemandcheckquestion містить поле skilledstaffcheckquestiontext, яким визначається текст запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом CharField.

Клас Skilledstaffdemandcheckquestion містить поле skilledstaffcheckquestionnumber, яким визначається номер запитання серед всіх запитань перевірки вмінь заявника за вакансією для кваліфікованого персоналу, якщо порядок не має бути випадковим, за типом PositiveIntegerField.

Клас Skilledstaffdemandcheckquestion містить поле skilledstaffcheckanswerrandom, яким визначається випадковість порядку запитань під час перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом BooleanField.

Клас Skilledstaffdemandcheckquestion містить поле skilledstaffcheckquestionminlevel, яким визначається кількість балів, яку можна набрати за одну правильну відповідь для запитання, за типом FloatField.

Клас Skilledstaffdemandcheckquestion містить поле skilledstaffcheckquestionmaxlevel, яким визначається кількість балів, яку можна набрати за всі правильні відповіді на запитання, за типом FloatField.

Модель відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу є класом Skilledstaffdemandcheckquestionanswer.

Клас `Skilledstaffdemandcheckquestionanswer` містить поле `skilledstaffdemandcheckquestion`, яким визначається запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, до якого належить ця відповідь, за типом `ForeignKey`.

Клас `Skilledstaffdemandcheckquestionanswer` містить поле `skilledstaffcheckanswertext`, яким визначається текст відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом `CharField`.

Клас `Skilledstaffdemandcheckquestionanswer` містить поле `skilledstaffcheckanswernumber`, яким визначається порядковий номер відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, коли порядок не є випадковим, за типом `PositiveIntegerField`.

Клас `Skilledstaffdemandcheckquestionanswer` містить поле `skilledstaffcheckanswercorrect`, яким визначається правильність відповіді на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом `BooleanField`.

Модель результатів проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу є класом `Skilledstaffdemandcheckposition`.

Клас `Skilledstaffdemandcheckposition` містить поле `skilledstaffdemandposition`, яким визначається подана заявка кандидата за вакансією для кваліфікованого персоналу, за якою отримано результати перевірки, за типом `ForeignKey`.

Клас `Skilledstaffdemandcheckposition` містить поле `skilledstaffdemandcheck`, яким визначається перевірка вмінь заявника за вакансією для кваліфікованого персоналу, за якою була здійснена перевірка цієї заявки, за типом `ForeignKey`.

Клас `Skilledstaffdemandcheckposition` містить поле `skilledstaffcheckstarttime`, яким визначається час початку проходження

перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом DateTimeField.

Клас Skilledstaffdemandcheckposition містить поле skilledstaffcheckendtime, яким визначається час завершення проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом DateTimeField.

Клас Skilledstaffdemandcheckposition містить поле skilledstaffchecklevel, яким визначається отримана оцінка за результатами проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу, за типом FloatField.

Модель відповіді результатів проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу є класом Skilledstaffdemandcheckpositionquestionanswer.

Клас Skilledstaffdemandcheckpositionquestionanswer містить поле skilledstaffdemandcheckposition, яким визначається результат проходження перевірки вмінь заявника за вакансією для кваліфікованого персоналу, до яких належить ця відповідь, за типом ForeignKey.

Клас Skilledstaffdemandcheckpositionquestionanswer містить поле skilledstaffdemandcheckquestionanswer, яким визначається відповідь на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, яка була обрана кандидатом, за типом ForeignKey.

Клас Skilledstaffdemandcheckpositionquestionanswer містить поле skilledstaffchecklevel, яким визначається отримана оцінка за цю відповідь на запитання перевірки вмінь заявника за вакансією для кваліфікованого персоналу, яка була обрана кандидатом, за типом FloatField.

Модель збереження вакансій для кваліфікованого персоналу користувачами, які здійснюють пошук вакансій, є класом Skilledstaffdemandsave.

Клас `Skilledstaffdemandsave` містить поле `skilledstaffdemand`, яким визначається вакансія для кваліфікованого персоналу, яку зберіг користувач, що здійснює пошук вакансії, за типом `ForeignKey`.

Клас `Skilledstaffdemandsave` містить поле `skilledstaffaccount`, яким визначається користувач, який здійснює пошук вакансії та який зберіг цю вакансію для кваліфікованого персоналу, за типом `ForeignKey`.

Клас `Skilledstaffdemandsave` містить поле `skilledstaffmark`, яким визначається текстова позначка, визначена під час збереження вакансії для кваліфікованого персоналу користувачем, який здійснює пошук вакансії, за типом `CharField`.

Клас `Skilledstaffdemandsave` містить поле `skilledstafftime`, яким визначається час збереження вакансії для кваліфікованого персоналу користувачем, який здійснює пошук вакансії, за типом `DateTimeField`.

Модель даних про здобуту освіту користувачів, які здійснюють пошук вакансій, є класом `Skilledstaffaccounteducation`.

Клас `Skilledstaffaccounteducation` містить поле `skilledstaffaccount`, яким визначається користувач, який здійснює пошук вакансій і здобув цю освіту, за типом `ForeignKey`.

Клас `Skilledstaffaccounteducation` містить поле `skilledstaffeducationlevel`, яким визначається рівень здобутої освіти користувача, який здійснює пошук вакансій, за типом `PositiveSmallIntegerField`.

Клас `Skilledstaffaccounteducation` містить поле `staffeducationalinstitution`, яким визначається заклад, в якому було здобуто цю освіту користувачем, який здійснює пошук вакансій, за типом `CharField`.

Клас `Skilledstaffaccounteducation` містить поле `staffeducationstartyear`, яким визначається рік початку здобуття освіти користувачем, який здійснює пошук вакансій, за типом `DateField`.

Клас `Skilledstaffaccounteducation` містить поле `staffeducationendyear`, яким визначається рік завершення здобуття освіти користувачем, який здійснює пошук вакансій, за типом `DateField`.

Клас `Skilledstaffaccounteducation` містить поле `staffeducationdegree`, яким визначається ступінь, який був отриманий користувачем, який здійснює пошук вакансій, за цією освітою, за типом `PositiveSmallIntegerField`.

Модель історії роботи користувача, який здійснює пошук вакансій, є класом `Skilledstaffaccountexpreience`.

Клас `Skilledstaffaccountexpreience` містить поле `staffemployer`, яким визначається компанія-працедавець, у якій працював користувач, який здійснює пошук вакансій, за типом `ForeignKey`.

Клас `Skilledstaffaccountexpreience` містить поле `skilledstaffaccount`, яким визначається користувач, який здійснює пошук вакансій і який має цей запис з історії роботи, за типом `ForeignKey`.

Клас `Skilledstaffaccountexpreience` містить поле `staffexperiencestartyear`, яким визначається рік початку роботи користувача, який здійснює пошук вакансій, у цій компанії, за типом `DateField`.

Клас `Skilledstaffaccountexpreience` містить поле `staffexperienceendyear`, яким визначається рік закінчення роботи користувача, який здійснює пошук вакансій, у цій компанії, за типом `DateField`.

Клас `Skilledstaffaccountexpreience` містить поле `staffprofession`, яким визначається професія, за якою в цій компанії-працедавці працював цей користувач, який здійснює пошук вакансій, за типом `ForeignKey`.

Клас `Skilledstaffaccountexpreience` містить поле `staffexperiencedetail`, яким визначаються додані текстом деталі роботи цього користувача, який здійснює пошук вакансій у цій компанії-працедавці, за типом `TextField`.

### **3.3 Висновки за розділом 3**

Програмне забезпечення підбору кваліфікованого персоналу реалізовано повністю. Отриману структуру проєкту, створені функції та класи описані в цьому розділі.

## **4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ**

### **4.1 Призначення програми**

Програма розроблена, щоб забезпечити для компаній-працедавців підбір кваліфікованого персоналу на основі пошуку такого персоналу компаніями на основі визначення вакансій, пошуку вакансій кандидатами та організації відбору кваліфікованого персоналу через проведення етапів такого відбору з голосуванням представників компанії-роботодавця, проведенням за необхідності перевірок вмінь і знань кандидатів на вакансії для кваліфікованого персоналу на основі тестування.

### **4.2 Опис роботи з програмою підбору кваліфікованого персоналу**

З програмою підбору кваліфікованого персоналу може працювати будь-який користувач, у тому числі не маючи облікового запису. Під час звернення без облікового запису він обмежений в можливостях, адже фактично може тільки виконувати пошук за вакансіями для кваліфікованого персоналу, а також відповідно переходити на сторінки офісу-представництва, компанії-працедавця за відповідними посиланнями. При цьому він має, наприклад, на сторінці вакансії посилання для подання власної анкети за вакансією, збереження вакансії для кваліфікованого персоналу, проте для їх використання повинен авторизуватись (рис. 4.1). Після авторизації при цьому ці посилання візуально виглядають інакше. Також такому користувачу можливо перейти до сторінки перегляду відкритих вакансій для кваліфікованого персоналу за офісом компанії-працедавця.

Користувач до авторизації зі сторінки авторизації може також перейти до реєстрації облікового запису користувача-кандидата. Зареєструвати інші ролі користувачів може або адміністратор через адміністративну панель, або користувач, який має таку роль сам.

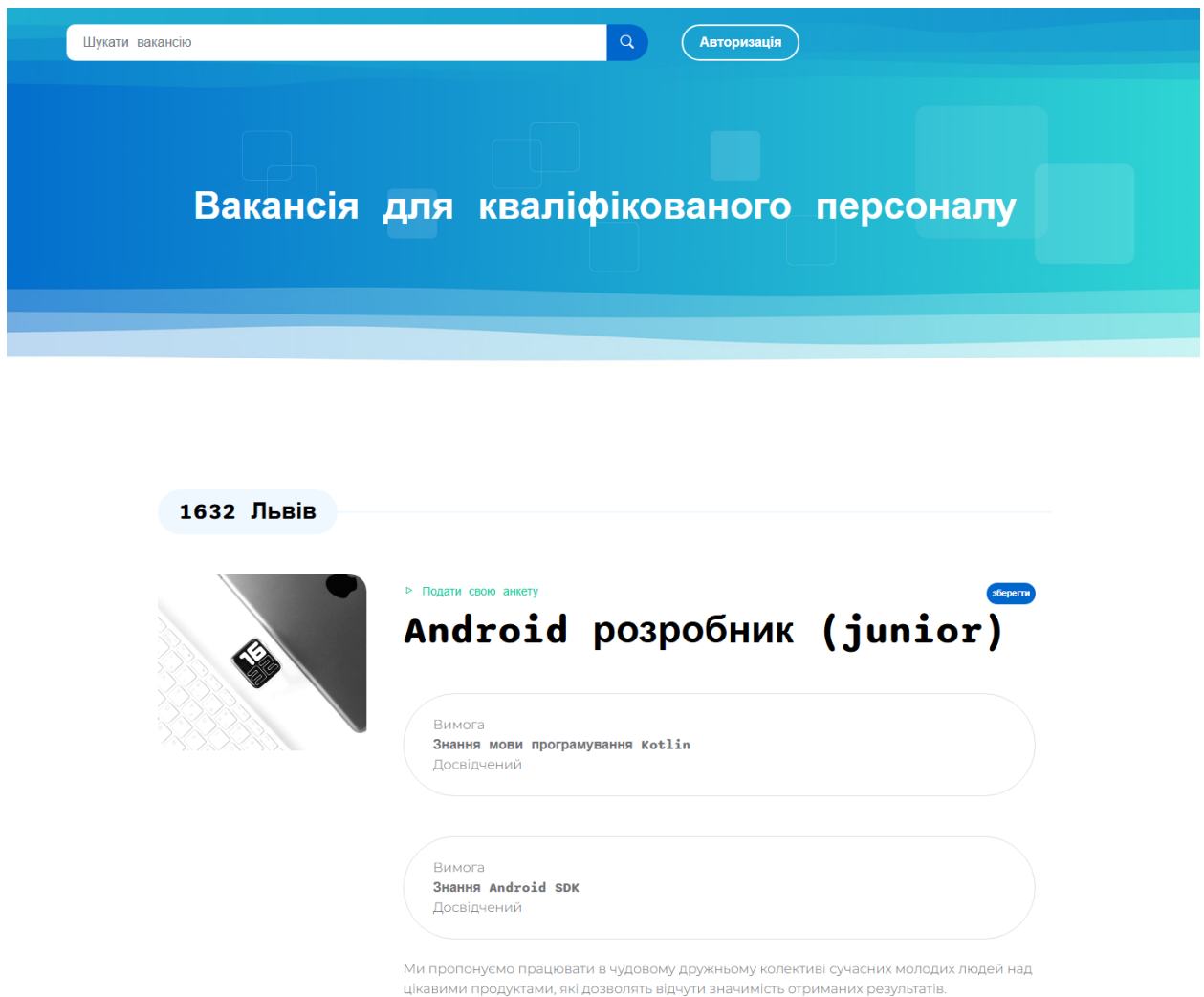


Рисунок 4.1 – Сторінка вакансії для кваліфікованого персоналу

Користувач-кандидат у програмі підбору кваліфікованого персоналу може окрім рядка пошуку використовувати також меню. При цьому рядок пошуку дозволяє ввести назву, але далі відображається окрема сторінка пошуку, на якій можна задати інші параметри пошуку вакансії для кваліфікованого персоналу (необхідну зарплатню, навички тощо).

Меню для користувача-кандидата включає засоби переходи до виконаних ним подань за вакансіями для кваліфікованого персоналу, при цьому створення самих анкет за професіями відбувається з профілю. Необхідні для проходження користувачем перевірки за вакансіями відображаються у відповідному розділі. Вакансії компанії-працедавців, на які користувач-кандидат підписався, доступні з окремого розділу меню. Розділ працедавців включає засоби відображення підтриманих працедавцем цим користувачем та

самого переліку працедавців, на які підписався користувач.

Зі сторінки відкритих вакансій офісу-представництва компанії-працедавця (рис. 4.2) користувач-кандидат може одразу підтримати саму компанію-працедавця, підписатись на неї, подати одну зі своїх анкет на вакансію, зберегти вакансію собі за допомогою відповідної іконки.

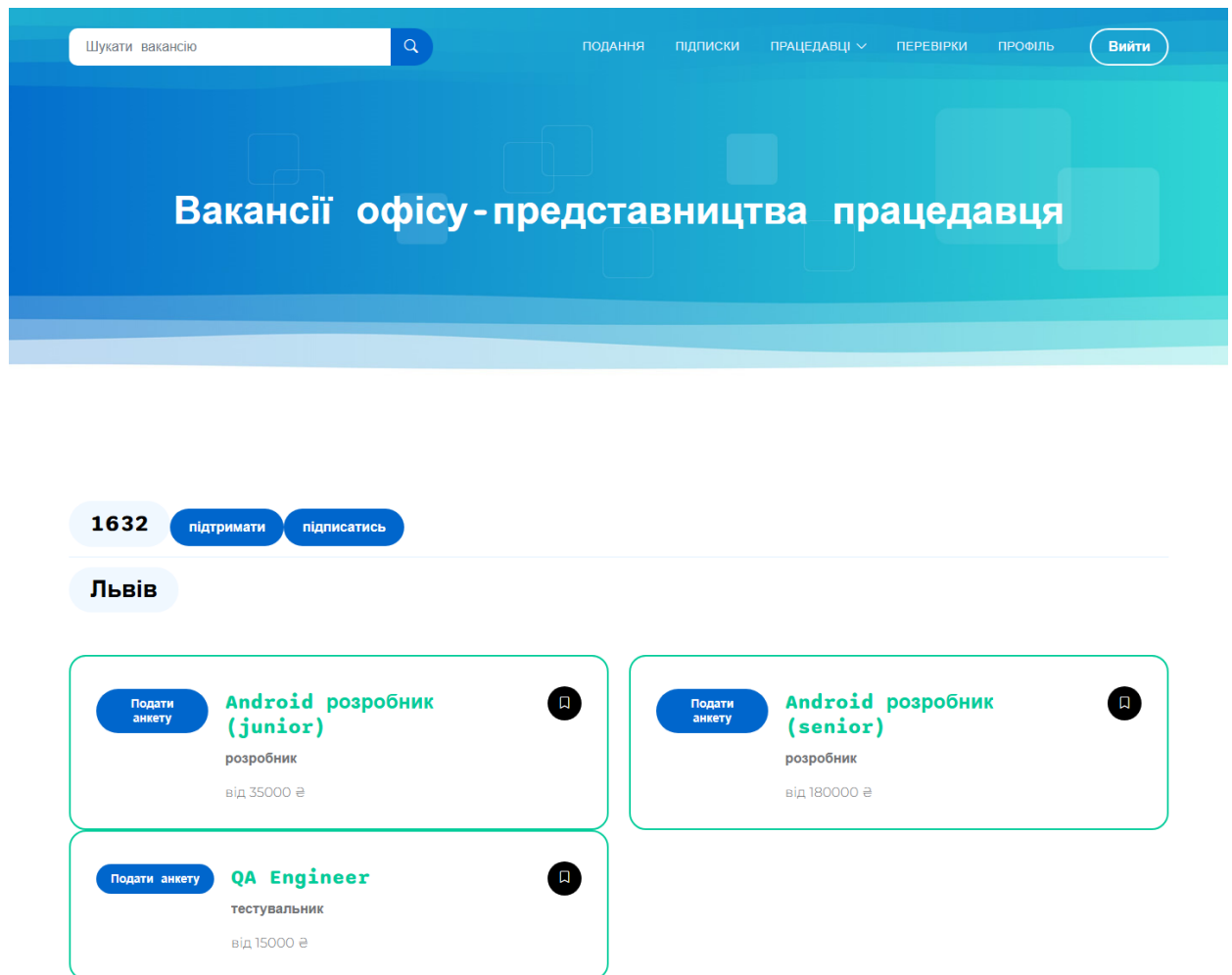


Рисунок 4.2 – Перегляд відкритих вакансій офісу-представництва компанії-працедавця

Представник компанії-працедавця з меню може переходити до засобів роботи з власними вакансіями, створення нової вакансії для кваліфікованого персоналу. З розділу голосувань доступні засоби перегляду визначених для його голосування вакансій для кваліфікованого персоналу, перегляду анкет, за які він проголосував. З переліку вакансій можна обрати потрібну, тоді

відобразяться анкети кандидатів (рис. 4.3). За анкету можна проголосувати або скасувати свій голос, можна також натиснути на посилання завершити, щоб взагалі відхилити цю анкету з відбору. Для того щоб підтвердити внесені користувачем дані в анкеті, переглянути їх повністю, потрібно відкрити сторінку цієї заявки, натиснувши на її назву.

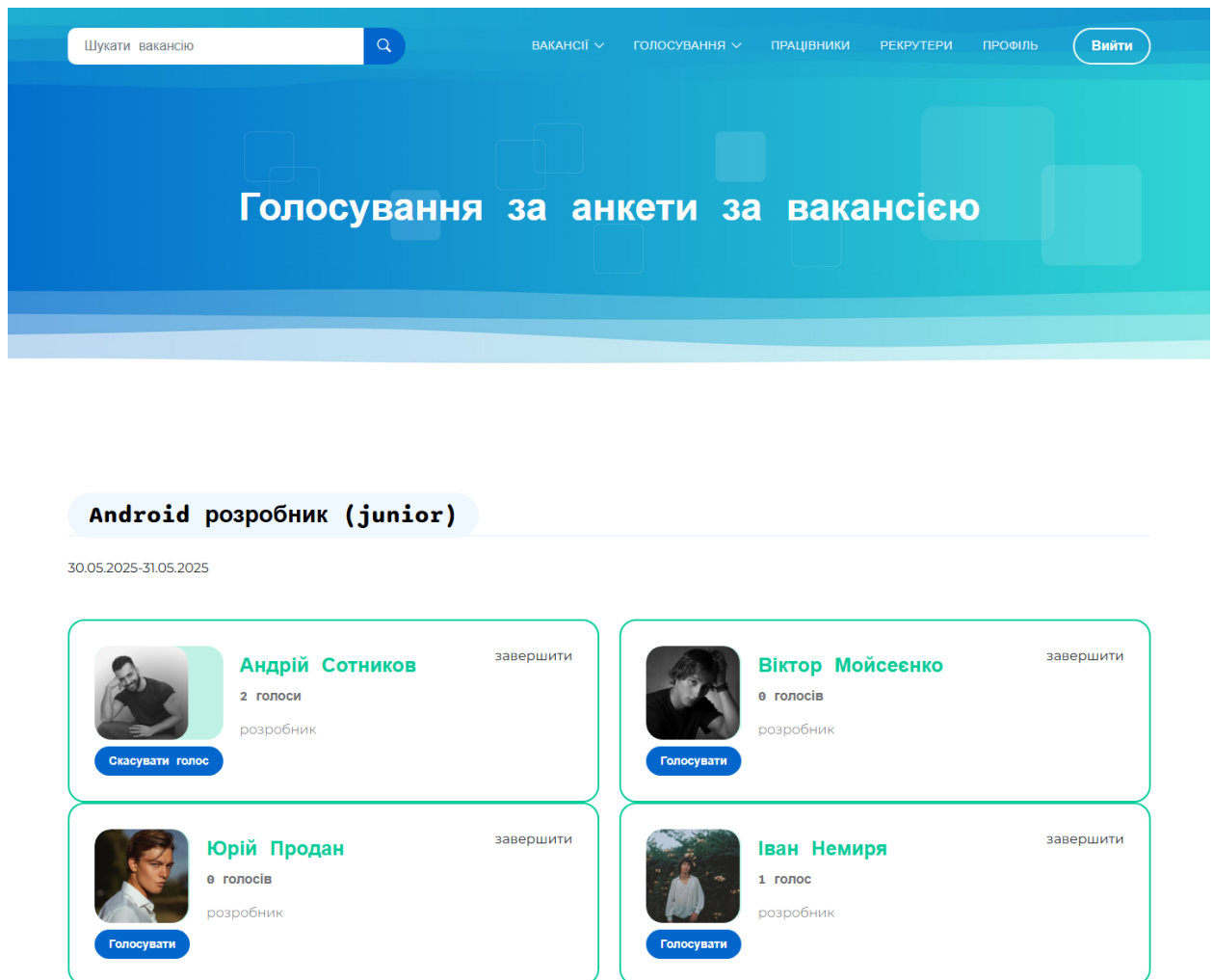


Рисунок 4.3 – Перегляд доступних для голосування анкет кандидатів за обраною вакансією

Також представник компанії-працедавця може з меню перейти до окремих розділів перегляду працівників компанії, які створили свої анкети за професіями у програмі. Це необхідно, щоб зручно підтверджувати дані таких працівників. Розділ рекрутерів дозволяє керувати переліком рекрутингових компаній компанії-працедавця.

Самі ж представники рекрутингових компаній можуть додавати нові анкети за професіями, переглядати власні додані такі анкети з першого розділу, також можуть переглядати подані рекрутинговій компанії вакансії для пошуку кваліфікованого персоналу від компаній-працедавців, з якими встановлено контакт (рис. 4.4). Також доступний розділ для керування такими компаніями-працедавцями.

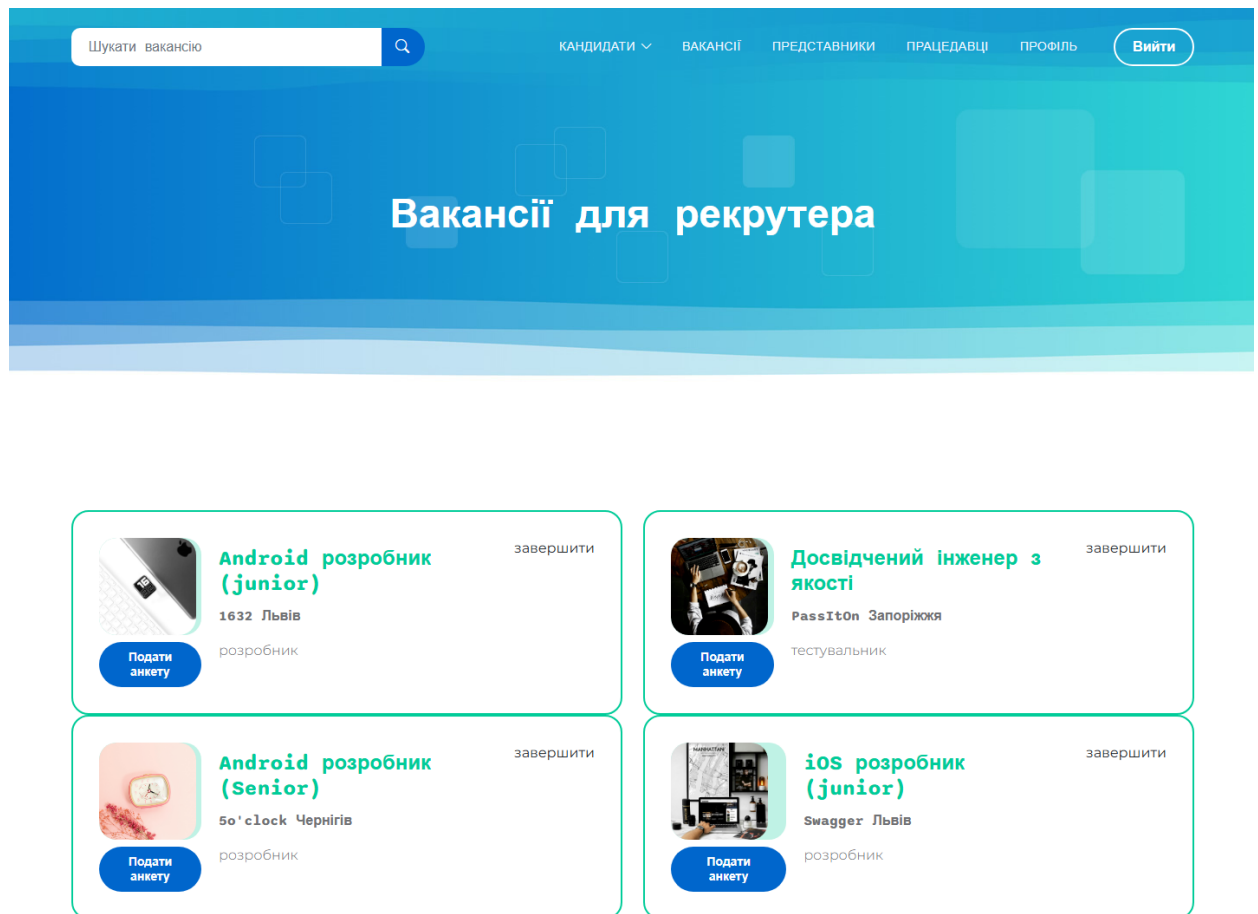


Рисунок 4.4 – Перегляд вакансій для кваліфікованого персоналу, визначених для обробки рекрутинговою компанією

### 4.3 Тестування програми підбору кваліфікованого персоналу

Тестування програми підбору кваліфікованого персоналу охоплювало весь перелік функцій програми повністю (табл. 4.1), в результаті чого всі вони були перевірені та мали успішне завершення.

Таблиця 4.1 – Результати тестування програми підбору кваліфікованого персоналу

| Функція                                                           | Користувач до авторизації | Користувач-кандидат | Представник компанії-працедавця | Представник рекрутингової компанії |
|-------------------------------------------------------------------|---------------------------|---------------------|---------------------------------|------------------------------------|
| 1                                                                 | 2                         | 3                   | 4                               | 5                                  |
| Авторизація                                                       | +                         | +                   | +                               | +                                  |
| Реєстрація користувачів-кандидатів                                |                           | +                   |                                 |                                    |
| Реєстрація представників компаній-працедавців                     |                           |                     | +                               |                                    |
| Реєстрація представників рекрутингових компаній                   |                           |                     |                                 | +                                  |
| Перегляд сторінки рекрутингової компанії                          |                           |                     | +                               | +                                  |
| Перегляд сторінки компанії-працедавця                             | +                         | +                   | +                               | +                                  |
| Перегляд сторінки користувача-кандидата                           |                           | +                   | +                               | +                                  |
| Встановлення підтримки компанії-працедавцю                        |                           | +                   |                                 |                                    |
| Вилучення підтримки компанії-працедавцю                           |                           | +                   |                                 |                                    |
| Визначення контакту з рекрутинговою компанією компанії-працедавця |                           |                     | +                               |                                    |

Продовження таблиці 4.1

| 1                                                                          | 2 | 3 | 4 | 5 |
|----------------------------------------------------------------------------|---|---|---|---|
| Скасування контакту з рекрутинговою компанією компанії-працедавця          |   |   | + | + |
| Перегляд компаній-працедавців рекрутингової компанії                       |   |   |   | + |
| Перегляд рекрутингових компаній компанії-працедавця                        |   |   | + |   |
| Перегляд підтриманих компаній-працедавців користувачем-кандидатом          |   | + |   |   |
| Перегляд сторінки офісу-представництва компанії-працедавця                 | + | + | + | + |
| Перегляд вакансії для кваліфікованого персоналу                            | + | + | + | + |
| Перегляд відкритих вакансій офісу-представництва компанії-працедавця       | + | + | + | + |
| Пошук вакансій для кваліфікованого персоналу                               | + | + | + | + |
| Створення вакансії для кваліфікованого персоналу                           |   |   | + |   |
| Оновлення вакансії для кваліфікованого персоналу                           |   |   | + |   |
| Голосування за анкету кандидата за вакансією для кваліфікованого персоналу |   |   | + |   |

Продовження таблиці 4.1

| 1                                                                                               | 2 | 3 | 4 | 5 |
|-------------------------------------------------------------------------------------------------|---|---|---|---|
| Перегляд доступних для голосування вакансій для кваліфікованого персоналу                       |   |   | + |   |
| Перегляд доступних для голосування анкет кандидатів за обраною вакансією                        |   |   | + |   |
| Вилучення голосування за анкету кандидата за вакансією для кваліфікованого персоналу            |   |   | + |   |
| Перегляд анкет кандидатів за вакансіями, за які проголосував користувач                         |   |   | + |   |
| Визначення рекрутингової компанії за вакансією компанії-працедавця                              |   |   | + |   |
| Видалення рекрутингової компанії за вакансією компанії-працедавця                               |   |   | + |   |
| Перегляд вакансій для кваліфікованого персоналу, визначених для обробки рекрутинговою компанією |   |   |   | + |
| Визначення завершення оброблення вакансії рекрутинговою компанією                               |   |   |   | + |

Продовження таблиці 4.1

| 1                                                                                  | 2 | 3 | 4 | 5 |
|------------------------------------------------------------------------------------|---|---|---|---|
| Створення етапу відбору за вакансією для кваліфікованого персоналу                 |   |   | + |   |
| Перегляд результатів проведення відбору за вакансією для кваліфікованого персоналу |   |   | + |   |
| Оновлення етапу відбору за вакансією для кваліфікованого персоналу                 |   |   | + |   |
| Вилучення етапу відбору за вакансією для кваліфікованого персоналу                 |   |   | + |   |
| Додавання даних освіти користувача-кандидата                                       |   | + |   |   |
| Оновлення даних освіти користувача-кандидата                                       |   | + |   |   |
| Вилучення даних освіти користувача-кандидата                                       |   | + |   |   |
| Додавання історії роботи користувача-кандидата                                     |   | + |   |   |
| Оновлення історії роботи користувача-кандидата                                     |   | + |   |   |
| Вилучення історії роботи користувача-кандидата                                     |   | + |   |   |
| Створення анкети за професією                                                      |   | + |   | + |
| Оновлення даних анкети за професією                                                |   | + |   | + |

Продовження таблиці 4.1

| 1                                                                                                    | 2 | 3 | 4 | 5 |
|------------------------------------------------------------------------------------------------------|---|---|---|---|
| Вилучення анкети за професією                                                                        |   | + |   | + |
| Подання анкети за професією на вакансію для кваліфікованого персоналу                                |   | + |   | + |
| Вилучення анкети за професією на вакансію для кваліфікованого персоналу                              |   | + |   | + |
| Перегляд користувачів-кандидатів за даними рекрутингової компанії                                    |   |   | + | + |
| Перегляд власних поданих анкет за професією на вакансії для кваліфікованого персоналу                |   | + |   |   |
| Підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем           |   |   | + |   |
| Вилучення підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем |   |   | + |   |
| Перегляд користувачів, які працювали в компанії-працедавці                                           |   |   | + |   |
| Створення перевірки за вакансією для кваліфікованого персоналу                                       |   |   | + |   |
| Оновлення параметрів перевірки за вакансією для кваліфікованого персоналу                            |   |   | + |   |

Продовження таблиці 4.1

| 1                                                                                        | 2 | 3 | 4 | 5 |
|------------------------------------------------------------------------------------------|---|---|---|---|
| Вилучення перевірки за вакансією для кваліфікованого персоналу                           |   |   | + |   |
| Додавання запитання до перевірки за вакансією для кваліфікованого персоналу              |   |   | + |   |
| Вилучення запитання до перевірки за вакансією для кваліфікованого персоналу              |   |   | + |   |
| Оновлення запитання до перевірки за вакансією для кваліфікованого персоналу              |   |   | + |   |
| Додавання відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу |   |   | + |   |
| Вилучення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу |   |   | + |   |
| Оновлення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу |   |   | + |   |
| Збереження вакансії для кваліфікованого персоналу у свій перелік                         |   | + |   |   |
| Вилучення збереження вакансії для кваліфікованого персоналу зі свого переліку            |   | + |   |   |

Продовження таблиці 4.1

| 1                                                                                       | 2 | 3 | 4 | 5 |
|-----------------------------------------------------------------------------------------|---|---|---|---|
| Перегляд збережених вакансій для кваліфікованого персоналу                              |   | + |   |   |
| Перегляд сторінки перевірки за вакансією для кваліфікованого персоналу                  |   | + | + |   |
| Збереження результатів перевірки користувача за вакансією для кваліфікованого персоналу |   | + |   |   |
| Перегляд призначених перевірок за вакансіями для користувача                            |   | + |   |   |
| Оформлення підписки на компанію-працедавця                                              |   | + |   |   |
| Вилучення підписки на компанію-працедавця                                               |   | + |   |   |
| Перегляд вакансій для кваліфікованого персоналу компаній-працедавців з підписок         |   | + |   |   |
| Внесення даних компанії-працедавця                                                      |   |   | + |   |
| Внесення даних офісу-представництва компанії-працедавця                                 |   |   | + |   |
| Внесення даних рекрутингової компанії                                                   |   |   |   | + |
| Внесення даних користувача-кандидата                                                    |   | + |   |   |

Кінець таблиці 4.1

| 1                                                                         | 2 | 3 | 4 | 5 |
|---------------------------------------------------------------------------|---|---|---|---|
| Керування анкетною за професією на вакансію для кваліфікованого персоналу |   |   | + |   |
| Перегляд анкети за професією на вакансію для кваліфікованого персоналу    |   | + | + | + |

#### 4.4 Висновки за розділом 4

Тестування програми підбору кваліфікованого персоналу дозволило пересвідчитись у виконанні тих функціональних вимог, які були висунуті для програми. Описано процес роботи зі сторони користувачів всіх ролей (користувачів до авторизації, користувачів-кандидатів, представників компанії-працедавця та рекрутингових компаній).

## ВИСНОВКИ

У роботі розроблено програмне забезпечення підбору кваліфікованого персоналу для надання можливості реалізувати вибір потрібних працівників, чим досягнуто поставлену мету.

Програмне забезпечення підбору кваліфікованого персоналу базується на наступних концептуальних можливостях:

- пошуку кваліфікованого персоналу зі створенням відповідних вакансій на основі відгуків від самих кандидатів;
- надання пропозицій щодо кандидатів для вакансій рекрутинговими компаніями, які повинні мати своє представництво в програмі;
- реалізації багатоетапного відбору кандидатів для визначення вибору кваліфікованого персоналу з підтримкою голосування на цих етапах представниками компанії-працедавця;
- забезпечення базової перевірки кандидатів за поданими заявками на вакансії на основі автоматизованого тестування.

Для програми підбору кваліфікованого персоналу виконано аналіз аналогів, визначено концепцію розробки, забезпечено проектування програми, вибравши для розробки Python-фреймворк Django, побудувавши структуру бази даних, визначивши весь набір необхідних сценаріїв, реалізовано програму з описом результатів, забезпечуючи перевірку програми за допомогою тестування.

Програма розроблена, щоб забезпечити для компаній-працедавців підбір кваліфікованого персоналу на основі пошуку такого персоналу компаніями на основі визначення вакансій, пошуку вакансій кандидатами та організації відбору кваліфікованого персоналу через проведення етапів такого відбору з голосуванням представників компанії-роботодавця, проведенням за необхідності перевірок вмінь і знань кандидатів на вакансії для кваліфікованого персоналу на основі тестування.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Пошук вакансій Indeed (Індід робота) [Електрон. ресурс]. – Режим доступу : <https://ua.indeed.com/?r=us>.
2. Work.ua – сайт пошуку роботи №1 в Україні [Електрон. ресурс]. – Режим доступу : <https://www.work.ua/>.
3. Robota.ua [Електрон. ресурс]. – Режим доступу : <https://robota.ua/>.
4. FastAPI [Electronic resource]. – Access mode : <https://fastapi.tiangolo.com/>.
5. Django Tutorial. Learn Django Framework. GeeksforGeeks [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/django-tutorial/>.
6. Pod Talk Free Template [Electronic resource]. – Access mode : <https://templatemo.com/tm-584-pod-talk>.

**ДОДАТОК А**  
**Технічне завдання**

## **Вступ**

Від мотивації персоналу залежить ефективність роботи всієї компанії. Однак іншою важливою складовою є сама якість підібраних кадрів, адже вона також безпосередньо впливає на результати компанії. Підібрати кваліфікований персонал не так і просто на сучасному ринку. Пропозицій може бути багато, проте багато часу витрачається на відбір тих кандидатів, які дійсно відповідають вимогам працедавця. Тому важливо розробити програмне забезпечення, яке б з одного боку підтримувало процес пошуку персоналу, з іншого боку враховувало саме кваліфікацію такого персоналу, і з третього – забезпечувало би і підтримку самого відбору, а не тільки пошук.

### **A.1 Підстави для розробки**

Наказ № 209 від 28 квітня 2025 року був виданий у Національному університеті «Запорізька політехніка», визначивши для виконання цієї дипломної кваліфікаційної роботи бакалавра тему «Розробка програмного забезпечення підбору кваліфікованого персоналу», яку було далі уточнено у вигляді завдання на дипломну кваліфікаційну роботу бакалавра.

### **A.2 Призначення розробки**

Програма розробляється для того, щоб забезпечити для компаній-працедавців підбір кваліфікованого персоналу на основі пошуку такого персоналу компаніями на основі визначення вакансій, пошуку вакансій кандидатами та організації відбору кваліфікованого персоналу через проведення етапів такого відбору з голосуванням представників компанії-роботодавця, проведенням за необхідності перевірок вмінь і знань кандидатів на вакансії для кваліфікованого персоналу на основі тестування.

### **А.3 Основні вимоги до програми**

#### **А.3.1 Вимоги до функціональних характеристик**

Програма підбору кваліфікованого персоналу повинна надавати функції:

- авторизація;
- реєстрація користувачів-кандидатів;
- реєстрація представників компаній-працедавців;
- реєстрація представників рекрутингових компаній;
- перегляд сторінки рекрутингової компанії;
- перегляд сторінки компанії-працедавця;
- перегляд сторінки користувача-кандидата;
- встановлення підтримки компанії-працедавцю;
- вилучення підтримки компанії-працедавцю;
- визначення контакту з рекрутинговою компанією компанії-працедавця;
- скасування контакту з рекрутинговою компанією компанії-працедавця;
- перегляд компаній-працедавців рекрутингової компанії;
- перегляд рекрутингових компаній компанії-працедавця;
- перегляд підтриманих компаній-працедавців користувачем-кандидатом;
- перегляд сторінки офісу-представництва компанії-працедавця;
- перегляд вакансії для кваліфікованого персоналу;
- перегляд відкритих вакансій офісу-представництва компанії-працедавця;
- пошук вакансій для кваліфікованого персоналу;
- створення вакансії для кваліфікованого персоналу;
- оновлення вакансії для кваліфікованого персоналу;
- голосування за анкету кандидата за вакансією для кваліфікованого

персоналу;

- перегляд доступних для голосування вакансій для кваліфікованого персоналу;

- перегляд доступних для голосування анкет кандидатів за обраною вакансією;

- вилучення голосування за анкету кандидата за вакансією для кваліфікованого персоналу;

- перегляд анкет кандидатів за вакансіями, за які проголосував користувач;

- визначення рекрутингової компанії за вакансією компанії-працедавця;

- видалення рекрутингової компанії за вакансією компанії-працедавця;

- перегляд вакансій для кваліфікованого персоналу, визначених для обробки рекрутинговою компанією;

- визначення завершення оброблення вакансії рекрутинговою компанією;

- створення етапу відбору за вакансією для кваліфікованого персоналу;

- перегляд результатів проведення відбору за вакансією для кваліфікованого персоналу;

- оновлення етапу відбору за вакансією для кваліфікованого персоналу;

- вилучення етапу відбору за вакансією для кваліфікованого персоналу;

- додавання даних освіти користувача-кандидата;

- оновлення даних освіти користувача-кандидата;

- вилучення даних освіти користувача-кандидата;

- додавання історії роботи користувача-кандидата;

- оновлення історії роботи користувача-кандидата;

- вилучення історії роботи користувача-кандидата;

- створення анкети за професією;

- оновлення даних анкети за професією;
- вилучення анкети за професією;
- подання анкети за професією на вакансію для кваліфікованого персоналу;
- вилучення анкети за професією на вакансію для кваліфікованого персоналу;
- перегляд користувачів-кандидатів за даними рекрутингової компанії;
- перегляд власних поданих анкет за професією на вакансії для кваліфікованого персоналу;
- підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем;
- вилучення підтвердження наданої користувачем інформації щодо себе за професією компанією-працедавцем;
- перегляд користувачів, які працювали в компанії-працедавці;
- створення перевірки за вакансією для кваліфікованого персоналу;
- оновлення параметрів перевірки за вакансією для кваліфікованого персоналу;
- вилучення перевірки за вакансією для кваліфікованого персоналу;
- додавання запитання до перевірки за вакансією для кваліфікованого персоналу;
- вилучення запитання до перевірки за вакансією для кваліфікованого персоналу;
- оновлення запитання до перевірки за вакансією для кваліфікованого персоналу;
- додавання відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу;
- вилучення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу;
- оновлення відповіді на запитання до перевірки за вакансією для кваліфікованого персоналу;

- збереження вакансії для кваліфікованого персоналу у свій перелік;
- вилучення збереження вакансії для кваліфікованого персоналу зі свого переліку;
- перегляд збережених вакансій для кваліфікованого персоналу;
- перегляд сторінки перевірки за вакансією для кваліфікованого персоналу;
- збереження результатів перевірки користувача за вакансією для кваліфікованого персоналу;
- перегляд призначених перевірок за вакансіями для користувача;
- оформлення підписки на компанію-працедавця;
- вилучення підписки на компанію-працедавця;
- перегляд вакансій для кваліфікованого персоналу компаній-працедавців з підписок;
- внесення даних компанії-працедавця;
- внесення даних офісу-представництва компанії-працедавця;
- внесення даних рекрутингової компанії;
- внесення даних користувача-кандидата;
- перегляд анкети за професією на вакансію для кваліфікованого персоналу;
- керування анкетною за професією на вакансію для кваліфікованого персоналу.

### **A.3.2 Умови експлуатації**

Програма не має особливих вимог до користувачів, до клієнтської сторони.

На сервері, враховуючи наявність сервера зокрема, має бути забезпечена роботою адміністратора. Така вимога обумовлена необхідністю початкового розгортання застосунку на сервері, а для цього потрібні спеціалізовані знання, а також необхідність керування застосунком через

адміністративну панель.

### **A.3.3 Вимоги до складу та параметрів технічних засобів**

Технічно сервер повинен забезпечуватися обладнанням, що включає мінімум процесор з 2 ядрами, 4 Гб оперативної пам'яті та 10 Гб виділеного простору на жорсткому диску.

**ДОДАТОК Б**  
**Текст програми**

```

from .models import Staffemployer
from .models import City
from .models import Staffemployeroffice
from .models import Staffemployeraccount
from .models import Recruiter
from .models import Recruiteraccount
from .models import Staffemployerrecruiter
from .models import Staffprofessioncategory
from .models import Staffprofession
from .models import Staffskill
from .models import Skilledstaffdemand
from .models import Skilledstaffdemandskill
from .models import Skilledstaffdemandrecruiter
from .models import Skilledstaffdemandstage
from .models import Skilledstaffaccount
from .models import Skilledstaffaccounteducation
from .models import Skilledstaffaccountexperience
from .models import Skilledstaffposition
from .models import Skilledstaffpositionskill
from .models import Skilledstaffdemandposition
from .models import Skilledstaffpositionconfirm
from .models import Skilledstaffpositionforconfirm
from .models import Skilledstaffpositionforconfirmskill
from .models import Skilledstaffdemandpositionstage
from .models import Skilledstaffdemandpositionstagevote
from .models import Skilledstaffdemandcheck
from .models import Skilledstaffdemandcheckquestion
from .models import Skilledstaffdemandcheckquestionanswer
from .models import Skilledstaffdemandcheckposition
from .models import
Skilledstaffdemandcheckpositionquestionanswer
from .models import Skilledstaffdemandsave
from django.core.paginator import Paginator
from django.shortcuts import render
from django.shortcuts import redirect

```

```

def recruiter_page_view(request, recruiter_id = None):
    auth_recruiter = None
    auth_rectuiter_account = None
    if request.user.is_authenticated:

```

```

        auth_rectuiter_account =
Recruiteraccount.objects.select_related("recruiter",
"recruiter__recruitercity").get(userindjango = request.user)
        if auth_rectuiter_account:
            auth_recruiter = auth_rectuiter_account.recruiter
        if recruiter_id:
            try:
                recruiter_get =
Recruiter.objects.select_related("recruitercity").get(id =
recruiter_id)
            except Recruiter.DoesNotExist:
                recruiter_get = None
        else:
            if auth_recruiter:
                recruiter_get = auth_recruiter
            if not recruiter_get:
                return render(request, "skistaff/error.html",
{"caption": "Помилка. Такої рекрутингової компанії мені не вдалось
знайти."})
        if auth_rectuiter_account:
            recruiter_accounts =
Recruiteraccount.objects.select_related("userindjango").filter(recruit
er = recruiter_get)
        return render(request, "skistaff/recruiter.html",
{"recruiter": recruiter_get, "accounts": recruiter_accounts })

```

```

def staffemployer_page_view(request, staffemployer_id = None):
    auth_staffemployer = None
    auth_staffemployer_account = None
    if request.user.is_authenticated:
        auth_staffemployer_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer",
"staffemployeroffice__staffemployerofficecity").get(userindjango =
request.user)
        if auth_staffemployer_account:
            auth_staffemployer =
auth_staffemployer_account.staffemployeroffice.staffemployer
        if staffemployer_id:
            try:
                staffemployer_get = Staffemployer.objects.get(id =
staffemployer_id)

```

```

        except Staffemployer.DoesNotExist:
            staffemployer_get = None
        else:
            if auth_staffemployer:
                staffemployer_get = auth_staffemployer
            if not auth_staffemployer:
                return render(request, "skistaff/error.html",
{"caption": "Помилка. Такої компанії-працедавця мені не вдалось
знайти."})
            staffemployer_offices =
Staffemployeroffice.objects.select_related("staffemployerofficecity").
filter(staffemployer = staffemployer_get)
            employer_support = False
            employer_follow = False
            if auth_staffemployer_account:
                staffemployer_accounts =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer",
"staffemployeroffice__staffemployerofficecity").filter(staffemployerof
fice__staffemployer = auth_staffemployer)
            else:
                if request.user.is_authenticated:
                    auth_staff_account =
Skilledstaffaccount.objects.prefetch_related("skilledstaffsupports",
"staffemployers").select_related("userindjango").get(userindjango =
request.user)
                    if auth_staff_account:
                        if staffemployer_get in
auth_staff_account.staffemployers:
                            employer_follow = True
                        if staffemployer_get in
auth_staff_account.skilledstaffsupports:
                            employer_support = True
                return render(request, "skistaff/employer.html",
{"employer": staffemployer_get, "accounts": staffemployer_accounts,
"support": employer_support, "follow": employer_follow, "offices":
staffemployer_offices })

def staffemployer_office_page_view(request,
staffemployeroffice_id = None):
    auth_staffemployeroffice = None
    auth_staffemployeroffice_account = None

```

```

        if request.user.is_authenticated:
            auth_staffemployeroffice_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer",
"staffemployeroffice__staffemployerofficecity").get(userindjango =
request.user)
            if auth_staffemployeroffice_account:
                auth_staffemployeroffice =
auth_staffemployeroffice_account.staffemployeroffice.staffemployer
                if staffemployeroffice_id:
                    try:
                        staffemployeroffice_get =
Staffemployeroffice.objects.select_related("staffemployer").get(id =
staffemployeroffice_id)
                    except Staffemployeroffice.DoesNotExist:
                        staffemployeroffice_get = None
                else:
                    if auth_staffemployeroffice:
                        staffemployeroffice_get = auth_staffemployeroffice
                    if not auth_staffemployeroffice:
                        return render(request, "skistaff/error.html",
{"caption": "Помилка. Такого офісу компанії-працедавця мені не вдалось
знайти."})
                skilledstaff_demands =
Skilledstaffdemand.objects.select_related("staffprofession").filter(st
affemployeroffice = staffemployeroffice_get)
                skilledstaff_demands =
skilledstaff_demands.filter(demandstatus = 0)
                skilledstaff_demands = skilledstaff_demands.order_by("-
demandfinaltime")
                skilledstaff_demands = skilledstaff_demands[:4]
                employer_support = False
                employer_follow = False
                if auth_staffemployeroffice_account:
                    staffemployeroffice_accounts =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer",
"staffemployeroffice__staffemployerofficecity").filter(staffemployerof
fice = auth_staffemployeroffice)
                else:
                    if request.user.is_authenticated:
                        auth_staff_account =
Skilledstaffaccount.objects.prefetch_related("skilledstaffsupports",

```

```

"staffemployers").select_related("userindjango").get(userindjango =
request.user)
        if auth_staff_account:
            if staffemployeroffice_get.staffemployer in
auth_staff_account.staffemployers:
                employer_follow = True
            if staffemployeroffice_get.staffemployer in
auth_staff_account.skilledstaffsupports:
                employer_support = True
        return render(request, "skistaff/employeroffice.html",
{"employeroffice": staffemployeroffice_get, "accounts":
staffemployeroffice_accounts, "demands": skilledstaff_demands,
"support": employer_support, "follow": employer_follow })

def recruiter_employers_view(request, recruiter_id = None):
    auth_recruiter = None
    auth_rectuiter_account = None
    if request.user.is_authenticated:
        auth_rectuiter_account =
Recruiteraccount.objects.select_related("recruiter",
"recruiter__recruitercity").get(userindjango = request.user)
        if auth_rectuiter_account:
            auth_recruiter = auth_rectuiter_account.recruiter
    if recruiter_id:
        try:
            recruiter_get =
Recruiter.objects.select_related("recruitercity").get(id =
recruiter_id)
        except Recruiter.DoesNotExist:
            recruiter_get = None
    else:
        if auth_recruiter:
            recruiter_get = auth_recruiter
    if not recruiter_get:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Такої рекрутингової компанії мені не вдалось
знайти."})
    recruiter_employers = []
    if auth_rectuiter_account.recruiter == recruiter_get:
        recruiter_employers =
Staffemployerrecruiter.objects.select_related("staffemployer",
"recruiter").filter(recruiter = recruiter_get)

```

```

        recruiter_employers = recruiter_employers.order_by("-
contractdate")
        return render(request, "skistaff/recruiteremployers.html",
{"recruiter": recruiter_get, "employers": recruiter_employers })

def staffemployer_recruiters_view(request, staffemployer_id =
None):
    auth_staffemployer = None
    auth_staffemployer_account = None
    if request.user.is_authenticated:
        auth_staffemployer_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer",
"staffemployeroffice__staffemployerofficecity").get(userindjango =
request.user)
        if auth_staffemployer_account:
            auth_staffemployer =
auth_staffemployer_account.staffemployeroffice.staffemployer
            if staffemployer_id:
                try:
                    staffemployer_get = Staffemployer.objects.get(id =
staffemployer_id)
                except Staffemployer.DoesNotExist:
                    staffemployer_get = None
            else:
                if auth_staffemployer:
                    staffemployer_get = auth_staffemployer
                if not auth_staffemployer:
                    return render(request, "skistaff/error.html",
{"caption": "Помилка. Такої компанії-працедавця мені не вдалось
знайти."})
            employer_recruiters = []
            if
auth_staffemployer_account.staffemployeroffice.staffemployer ==
staffemployer_get:
                employer_recruiters =
Staffemployerrecruiter.objects.select_related("staffemployer",
"recruiter").filter(staffemployer = staffemployer_get)
                employer_recruiters = employer_recruiters.order_by("-
contractdate")
                return render(request, "skistaff/employer.html",
{"employer": staffemployer_get, "recruiters": employer_recruiters })

```

```

def staffemployer_supported_view(request):
    if request.user.is_authenticated:
        try:
            auth_staff_account =
Skilledstaffaccount.objects.prefetch_related("skilledstaffsupports",
"staffemployers").select_related("userindjango").get(userindjango =
request.user)
        else:
            auth_staff_account = None
    else:
        auth_staff_account = None
    if not auth_staff_account:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Потрібно авторизуватись, а тоді звертатись до
цієї функції"})
    for staffemployer in
auth_staff_account.skilledstaffsupports:
        if staffemployer in auth_staff_account.staffemployers:
            staffemployer.follow = True
        else:
            staffemployer.follow = False
    return render(request, "skistaff/employeroffice.html",
{"account": auth_staff_account})

def staffemployer_office_demands_view(request,
staffemployeroffice_id = None):
    auth_staffemployeroffice = None
    auth_staffemployeroffice_account = None
    if request.user.is_authenticated:
        auth_staffemployeroffice_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer",
"staffemployeroffice__staffemployerofficecity").get(userindjango =
request.user)
        if auth_staffemployeroffice_account:
            auth_staffemployeroffice =
auth_staffemployeroffice_account.staffemployeroffice.staffemployer
            if staffemployeroffice_id:
                try:
                    staffemployeroffice_get =
Staffemployeroffice.objects.select_related("staffemployer").get(id =
staffemployeroffice_id)

```

```

        except Staffemployeroffice.DoesNotExist:
            staffemployeroffice_get = None
    else:
        if auth_staffemployeroffice:
            staffemployeroffice_get = auth_staffemployeroffice
        if not auth_staffemployeroffice:
            return render(request, "skistaff/error.html",
{"caption": "Помилка. Такого офісу компанії-працедавця мені не вдалось знайти."})
        skilledstaff_demands =
Skilledstaffdemand.objects.select_related("staffprofession").filter(st
affemployeroffice = staffemployeroffice_get)
        skilledstaff_demands =
skilledstaff_demands.filter(demandstatus = 0)
        skilledstaff_demands = skilledstaff_demands.order_by("-
demandfinaltime")
        skilledstaff_demands = Paginator(skilledstaff_demands, 10)
        skilledstaff_demands_pos = request.GET.get('pos')
        skilledstaff_demands =
skilledstaff_demands.get_page(skilledstaff_demands_pos)
        employer_support = False
        employer_follow = False
        if request.user.is_authenticated:
            auth_staff_account =
Skilledstaffaccount.objects.prefetch_related("skilledstaffsupports",
"staffemployers").select_related("userindjango").get(userindjango =
request.user)
            if auth_staff_account:
                if staffemployeroffice_get.staffemployer in
auth_staff_account.staffemployers:
                    employer_follow = True
                if staffemployeroffice_get.staffemployer in
auth_staff_account.skilledstaffsupports:
                    employer_support = True
            return render(request,
"skistaff/employerofficedemands.html", {"employeroffice":
staffemployeroffice_get, "demands": skilledstaff_demands, "support":
employer_support, "follow": employer_follow })

from .forms import SkilledstaffaccounteducationForm

def skilledstaffaccount_education_add_view(request):
    if request.user.is_authenticated:

```

```

        auth_staff_account =
        Skilledstaffaccount.objects.select_related("userindjango").get(userind
        jango = request.user)
    else:
        return render(request, "skistaff/error.html",
        {"caption": "Помилка. Для внесення даних про освіту потрібно спершу
        авторизуватись як користувач-кандидат."})
        if request.method == "POST":
            skilledstaffaccount_education_form =
            SkilledstaffaccounteducationForm(request.POST)
            if skilledstaffaccount_education_form.is_valid():
                skilledstaffaccount_education =
                skilledstaffaccount_education_form.save(commit=False)
                skilledstaffaccount_education.skilledstaffaccount =
                auth_staff_account
                skilledstaffaccount_education.save()
                return redirect(request.path_info)
            else:
                skilledstaffaccount_education_form =
                SkilledstaffaccounteducationForm()
                return render(request, "skistaff/addededucation.html",
                {"form": skilledstaffaccount_education_form })

def skilledstaffaccount_education_edit_view(request,
skillededucation_id):
    if request.user.is_authenticated:
        auth_staff_account =
        Skilledstaffaccount.objects.select_related("userindjango").get(userind
        jango = request.user)
        try:
            skilledstaffaccount_education =
            Skilledstaffaccounteducation.objects.select_related("skilledstaffaccou
            nt").get(id = skillededucation_id)
        except Skilledstaffaccounteducation.DoesNotExist:
            skilledstaffaccount_education = None
            if skilledstaffaccount_education and
            skilledstaffaccount_education.skilledstaffaccount ==
            auth_staff_account:
                skilledstaffaccount_education_form =
                SkilledstaffaccounteducationForm(request.POST or None, instance =
                skilledstaffaccount_education)

```

```

        if request.method == "POST" and
skilledstaffaccount_education_form.is_valid():
            skilledstaffaccount_education_form.save()
            return redirect('skilledstaffeducationview', id
= skillededucation_id)
        else:
            skilledstaffaccount_education_form = None
            return render(request, "skistaff/editeducation.html",
{"form": skilledstaffaccount_education_form })

from .forms import SkilledstaffaccountexperianceForm

def skilledstaffaccount_experience_add_view(request):
    if request.user.is_authenticated:
        auth_staff_account =
Skilledstaffaccount.objects.select_related("userindjango").get(userind
jango = request.user)
    else:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Для внесення даних про історію роботи потрібно
спершу авторизуватись як користувач-кандидат."})
    if request.method == "POST":
        skilledstaffaccount_experience_form =
SkilledstaffaccountexperianceForm(request.POST)
        if skilledstaffaccount_experience_form.is_valid():
            skilledstaffaccount_experience =
skilledstaffaccount_experience_form.save(commit=False)
            skilledstaffaccount_experience.skilledstaffaccount =
auth_staff_account
            skilledstaffaccount_experience.save()
            return redirect(request.path_info)
        else:
            skilledstaffaccount_experience_form =
SkilledstaffaccountexperianceForm()
            return render(request, "skistaff/addexper.html", {"form":
skilledstaffaccount_experience_form })

def skilledstaffaccount_experience_edit_view(request,
skillexperience_id):
    if request.user.is_authenticated:

```

```

        auth_staff_account =
Skilledstaffaccount.objects.select_related("userindjango").get(userind
jango = request.user)
        try:
            skilledstaffaccount_experience =
Skilledstaffaccountexperience.objects.select_related("skilledstaffacc
ount", "staffemployer").get(id = skillexperience_id)
        except Skilledstaffaccountexperience.DoesNotExist:
            skilledstaffaccount_experience = None
            if skilledstaffaccount_experience and
skilledstaffaccount_experience.skilledstaffaccount ==
auth_staff_account:
                skilledstaffaccount_experience_form =
SkilledstaffaccountexperienceForm(request.POST or None, instance =
skilledstaffaccount_experience)
                if request.method == "POST" and
skilledstaffaccount_experience_form.is_valid():
                    skilledstaffaccount_experience_form.save()
                    return redirect('skilledstaffexperview', id =
skillexperience_id)
            else:
                skilledstaffaccount_experience_form = None
                return render(request, "skistaff/editexper.html", {"form":
skilledstaffaccount_experience_form })

```

```

from .forms import SkilledstaffdemandcheckquestionForm

```

```

def skilledstaffdemand_question_add_view(request,
skilledstaffdemandcheck_id):
    try:
        skilledstaffdemand_check =
Skilledstaffdemandcheck.objects.select_related("skilledstaffdemand",
"skilledstaffdemand__staffemployeroffice").filter(id =
skilledstaffdemandcheck_id)
    except Skilledstaffdemandcheck.DoesNotExist:
        skilledstaffdemand_check = None
        if not skilledstaffdemand_check:
            return render(request, "skistaff/error.html",
{"caption": "Помилка. Перевірка, яку задано, не існує."})
        if request.user.is_authenticated:
            auth_employer_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",

```

```

"staffemployeroffice__staffemployer", "userindjango").get(userindjango
= request.user)
    else:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Для внесення запитання для перевірки потрібно
спершу авторизуватись як представник компанії-працедавця."})
        if auth_employer_account.staffemployeroffice !=
skilledstaffdemand_check.skilledstaffdemand.staffemployeroffice:
            return render(request, "skistaff/error.html",
{"caption": "Помилка. Ви не можете працювати з цією перевіркою,
недостатньо прав."})
        if request.method == "POST":
            skilledstaffcheck_question_form =
SkilledstaffdemandcheckquestionForm(request.POST)
            if skilledstaffcheck_question_form.is_valid():
                skilledstaffcheck_question_form =
skilledstaffcheck_question_form.save(commit=False)
                skilledstaffcheck_question_form.skilledstaffdemand =
skilledstaffdemand_check
                skilledstaffcheck_question_form.save()
                return redirect(request.path_info)
        else:
            skilledstaffcheck_question_form =
SkilledstaffdemandcheckquestionForm()
            return render(request, "skistaff/addque.html", {"form":
skilledstaffcheck_question_form })

def skilledstaffdemand_question_edit_view(request,
skillcheckquestion_id):
    if request.user.is_authenticated:
        auth_employer_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer", "userindjango").get(userindjango
= request.user)
    else:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Для внесення відповіді для перевірки потрібно
спершу авторизуватись як представник компанії-працедавця."})
    try:
        skilledstaffdemand_check_que =
Skilledstaffdemandcheckquestion.objects.select_related("skilledstaffde
mandcheck", "skilledstaffdemandcheck__skilledstaffdemand",

```

```

"skilledstaffdemandcheck__skilledstaffdemand__staffemployeroffice").filter(id = skillcheckquestion_id)
    except Skilledstaffdemandcheckquestion.DoesNotExist:
        skilledstaffdemand_check_que = None
        if skilledstaffdemand_check_que and
skilledstaffdemand_check_que.skilledstaffdemandcheck.skilledstaffdemand.staffemployeroffice == auth_employer_account.staffemployeroffice:
            skilledstaffdemand_question_form =
SkilledstaffdemandcheckquestionForm(request.POST or None, instance =
skilledstaffdemand_check_que)
            if request.method == "POST" and
skilledstaffdemand_question_form.is_valid():
                skilledstaffdemand_question_form.save()
                return redirect('skilledstaffcheckview', id =
skilledstaffdemand_check_que.skilledstaffdemandcheck.id)
            else:
                skilledstaffdemand_question_form = None
                return render(request, "skistaff/editque.html", {"form":
skilledstaffdemand_question_form })

```

```

from .forms import SkilledstaffdemandcheckquestionanswerForm

```

```

def skilledstaffdemand_answer_add_view(request,
skilledstaffdemandcheckque_id):
    try:
        skilledstaffdemand_check_que =
Skilledstaffdemandcheckquestion.objects.select_related("skilledstaffdemandcheck", "skilledstaffdemandcheck__skilledstaffdemand",
"skilledstaffdemandcheck__skilledstaffdemand__staffemployeroffice").filter(id = skilledstaffdemandcheckque_id)
    except Skilledstaffdemandcheckquestion.DoesNotExist:
        skilledstaffdemand_check_que = None
        if not skilledstaffdemand_check_que:
            return render(request, "skistaff/error.html",
{"caption": "Помилка. Запитання перевірки, яке задано, не існує."})
        if request.user.is_authenticated:
            auth_employer_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer", "userindjango").get(userindjango = request.user)
        else:
            return render(request, "skistaff/error.html",
{"caption": "Помилка. Для внесення відповіді на запитання для

```

```

перевірки потрібно спершу авторизуватись як представник компанії-
працедавця."})
        if auth_employer_account.staffemployeroffice !=
skilledstaffdemand_check_que.skilledstaffdemandcheck.skilledstaffdeman
d.staffemployeroffice:
            return render(request, "skistaff/error.html",
{"caption": "Помилка. Ви не можете працювати з цим запитанням
перевірки, недостатньо прав."})
        if request.method == "POST":
            skilledstaffcheck_question_answer_form =
SkilledstaffdemandcheckquestionanswerForm(request.POST)
            if skilledstaffcheck_question_answer_form.is_valid():
                skilledstaffcheck_question_answer_form =
skilledstaffcheck_question_answer_form.save(commit=False)

skilledstaffcheck_question_answer_form.skilledstaffdemandcheckquestion
= skilledstaffdemand_check_que
                skilledstaffcheck_question_answer_form.save()
                return redirect(request.path_info)
            else:
                skilledstaffcheck_question_answer_form =
SkilledstaffdemandcheckquestionanswerForm()
                return render(request, "skistaff/addanswer.html", {"form":
skilledstaffcheck_question_answer_form })

def skilledstaffdemand_answer_edit_view(request,
skillcheckanswer_id):
    if request.user.is_authenticated:
        auth_employer_account =
Staffemployeraccount.objects.select_related("staffemployeroffice",
"staffemployeroffice__staffemployer", "userindjango").get(userindjango
= request.user)
    else:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Для внесення відповіді для перевірки потрібно
спершу авторизуватись як представник компанії-працедавця."})
    try:
        skilledstaffdemand_check_answer =
Skilledstaffdemandcheckquestionanswer.objects.select_related("skilleds
taffdemandcheckquestion",
"skilledstaffdemandcheckquestion__skilledstaffdemandcheck",
"skilledstaffdemandcheckquestion__skilledstaffdemandcheck__skilledstaf

```

```

fdemand",
"skilledstaffdemandcheckquestion__skilledstaffdemandcheck__skilledstaffdemand__staffemployeroffice").filter(id = skillcheckanswer_id)
    except Skilledstaffdemandcheckquestionanswer.DoesNotExist:
        skilledstaffdemand_check_answer = None
    if skilledstaffdemand_check_answer and
skilledstaffdemand_check_answer.skilledstaffdemandcheckquestion.skilledstaffdemandcheck.skilledstaffdemand.staffemployeroffice ==
auth_employer_account.staffemployeroffice:
        skilledstaffdemand_answer_form =
SkilledstaffdemandcheckquestionanswerForm(request.POST or None,
instance = skilledstaffdemand_check_answer)
        if request.method == "POST" and
skilledstaffdemand_answer_form.is_valid():
            skilledstaffdemand_answer_form.save()
            return redirect('skilledstaffcheckview', id =
skilledstaffdemand_check_answer.skilledstaffdemandcheckquestion.skilledstaffdemandcheck.id)
        else:
            skilledstaffdemand_answer_form = None
    return render(request, "skistaff/editansw.html", {"form":
skilledstaffdemand_answer_form })

def skilledstaff_page_view(request, staff_id = None):
    auth_staff_account = None
    if staff_id:
        try:
            skilledstaff_get =
Skilledstaffaccount.objects.select_related("userindjango").get(id =
staff_id)
        except Skilledstaffaccount.DoesNotExist:
            skilledstaff_get = None
        elif request.user.is_authenticated:
            auth_staff_account =
Skilledstaffaccount.objects.prefetch_related("skilledstaffsupports",
"staffemployers").select_related("userindjango").get(userindjango =
request.user)
            skilledstaff_get = auth_staff_account
        else:
            skilledstaff_get = None
    if not skilledstaff_get:

```

```

        return render(request, "skistaff/error.html",
{"caption": "Помилка. Такого користувача-кандидата не вдалось
знайти."})
        skilledstaff_positions =
Skilledstaffposition.objects.select_related("staffprofession").filter(
skilledstaffaccount = skilledstaff_get)
        for skilledstaff_position in skilledstaff_positions:
            skilledstaff_position.skills =
Skilledstaffpositionskill.objects.select_related("staffskill").filter(
skilledstaffposition = skilledstaff_position)
            skilledstaff_position.confirms =
Skilledstaffpositionconfirm.objects.select_related("staffemployeraccou
nt").filter(skilledstaffposition = skilledstaff_position)
            skilledstaff_position.confirms =
skilledstaff_position.confirms.order_by("-staffconfirmdate")
            skilledstaff_position.asconfirmed =
Skilledstaffpositionforconfirm.objects.filter(skilledstaffposition =
skilledstaff_position)
            for skilledstaff_position_asconfirmed in
skilledstaff_position.asconfirmed:
                skilledstaff_position_asconfirmed.skills =
Skilledstaffpositionforconfirmskill.objects.filter(skilledstaffpositio
nforconfirm = skilledstaff_position_asconfirmed)
            return render(request, "skistaff/staff.html", {"staff":
skilledstaff_get, "positions": skilledstaff_positions })

```

```

def skilledstaffdemands_voting_view(request):
    if request.user.is_authenticated:
        auth_employer_account =
Staffemployeraccount.objects.prefetch_related("skilledstaffdemandvoter
s").select_related("staffemployeroffice",
"staffemployeroffice__staffemployer", "userindjango").get(userindjango
= request.user)
    else:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Для голосування потрібно спершу авторизуватись
як представник компанії-працедавця."})
        skilledstaffdemands =
Skilledstaffdemand.objects.select_related("staffprofession").filter(st
affemployeroffice = auth_employer_account.staffemployeroffice)
        skilledstaffdemands = skilledstaffdemands.order_by("-
demandfinaltime")

```

```

        skilledstaff_demands = []
        for skilledstaffdemand in skilledstaffdemands:
            if auth_employer_account in
skilledstaffdemand.skilledstaffdemandvoters:
                skilledstaff_demands.append(skilledstaffdemand)
        return render(request, "skistaff/forvoting.html",
{"demands": skilledstaff_demands })

def skilledstaffdemand_propositions_voting_view(request,
skilledstaffdemand_id):
    try:
        skilledstaffdemand_get =
Skilledstaffdemand.objects.select_related("staffprofession",
"staffemployeroffice",
"staffemployeroffice__staffemployer").prefetch_related("skilledstaffde
mandvoters").filter(id = skilledstaffdemand_id)
    except Skilledstaffdemand.DoesNotExist:
        skilledstaffdemand_get = None
    if request.user.is_authenticated:
        auth_employer_account =
Staffemployeraccount.objects.prefetch_related("skilledstaffdemandvoter
s").select_related("staffemployeroffice",
"staffemployeroffice__staffemployer", "userindjango").get(userindjango
= request.user)
    else:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Для голосування потрібно спершу авторизуватись
як представник компанії-працедавця."})
    if skilledstaffdemand_get.staffemployeroffice ==
auth_employer_account.staffemployeroffice and auth_employer_account in
skilledstaffdemand_get.skilledstaffdemandvoters:
        skilledstaffdemand_propositions =
Skilledstaffdemandposition.objects.select_related("skilledstaffpositio
n__skilledstaffaccount", "skilledstaffposition__skilledstaffaccount__
userindjango", "skilledstaffposition__staffprofession",
"skilledstaffdemand", "skilledstaffdemand__staffemployeroffice",
"skilledstaffdemand__staffprofession",
"skilledstaffdemand__staffemployeroffice__staffprofession").filter(sk
illedstaffdemand = skilledstaffdemand_get)
        for skilledstaffdemand_proposition in
skilledstaffdemand_propositions:

```

```

        skilledstaffdemand_propositions.votes =
        Skilledstaffdemandpositionstagevote.objects.select_related("skilledsta
        ffdemandpositionstage", "staffemployeraccount",
        "skilledstaffdemandpositionstage__skilledstaffdemandstage",
        "skilledstaffdemandpositionstage__skilledstaffposition").filter(skil
        leddemandpositionstage__skilledstaffposition =
        skilledstaffdemand_proposition, staffemployeraccount =
        auth_employer_account)
    else:
        skilledstaffdemand_propositions = []
        return render(request, "skistaff/demandforvoting.html",
        {"propositions": skilledstaffdemand_propositions })

```

```

def skilledstaffdemand_propositions_votes_view(request):
    if request.user.is_authenticated:
        auth_employer_account =
        Staffemployeraccount.objects.prefetch_related("skilledstaffdemandvoter
        s").select_related("staffemployeroffice",
        "staffemployeroffice__staffemployer", "userindjango").get(userindjango
        = request.user)
    else:
        return render(request, "skistaff/error.html",
        {"caption": "Помилка. Для голосування потрібно спершу авторизуватись
        як представник компанії-працедавця."})
    skilledstaffdemand_propositions_votes =
    Skilledstaffdemandpositionstagevote.objects.select_related("skilledsta
    ffdemandpositionstage", "staffemployeraccount",
    "skilledstaffdemandpositionstage__skilledstaffdemandstage",
    "skilledstaffdemandpositionstage__skilledstaffposition").filter(staffe
    mployeraccount = auth_employer_account)
    skilledstaffdemand_propositions_votes =
    skilledstaffdemand_propositions_votes.order_by("-staffvotetime")
    return render(request, "skistaff/votes.html",
    {"propositions": skilledstaffdemand_propositions_votes })

```

```

from .forms import SkilledstaffdemandstageForm

```

```

def skilledstaffdemand_stage_add_view(request,
skilledstaffdemand_id):
    try:
        skilledstaffdemand_get =
        Skilledstaffdemand.objects.select_related("staffprofession",

```

```

"staffemployeroffice",
"staffemployeroffice__staffemployer").prefetch_related("skilledstaffdemandvoters").filter(id = skilledstaffdemand_id)
    if request.user.is_authenticated:
        auth_employer_account =
Staffemployeraccount.objects.prefetch_related("skilledstaffdemandvoters").select_related("staffemployeroffice",
"staffemployeroffice__staffemployer", "userindjango").get(userindjango
= request.user)
    else:
        return render(request, "skistaff/error.html",
{"caption": "Помилка. Для внесення етапів відбору потрібно спершу
авторизуватись як представник компанії-працедавця."})
        if skilledstaffdemand_get.staffemployeroffice !=
auth_employer_account.staffemployeroffice:
            return render(request, "skistaff/error.html",
{"caption": "Помилка. Недостатньо прав для створення етапу відбору за
цією вакансією."})
        if request.method == "POST":
            skilledstaff_demand_stage_form =
SkilledstaffdemandstageForm (request.POST)
            if skilledstaff_demand_stage_form.is_valid():
                skilledstaff_demand_stage =
skilledstaff_demand_stage_form.save(commit=False)
                skilledstaff_demand_stage.skilledstaffdemand =
skilledstaffdemand_get
                skilledstaff_demand_stage.save()
                return redirect(request.path_info)
            else:
                skilledstaff_demand_stage_form =
SkilledstaffdemandstageForm()
                return render(request, "skistaff/addstage.html", {"form":
skilledstaff_demand_stage_form })

def skilledstaffdemand_stage_edit_view(request,
skilledstaffdemandstage_id):
    if request.user.is_authenticated:
        auth_employer_account =
Staffemployeraccount.objects.prefetch_related("skilledstaffdemandvoters").select_related("staffemployeroffice",
"staffemployeroffice__staffemployer", "userindjango").get(userindjango
= request.user)

```

```

        try:
            skilledstaffdemand_stage =
Skilledstaffdemandstage.objects.select_related("skilledstaffdemand",
"skilledstaffdemand__staffprofession",
"skilledstaffdemand__staffemployeroffice",
"skilledstaffdemand__staffemployeroffice__staffemployer",
"skilledstaffdemand__staffemployeroffice__staffemployerofficecity").ge
t(id = skilledstaffdemandstage_id)
        except Skilledstaffdemandstage.DoesNotExist:
            skilledstaffdemand_stage = None
            if skilledstaffdemand_stage and
skilledstaffdemand_stage.staffemployeroffice ==
auth_employer_account.staffemployeroffice:
                skilledstaffdemand_stage_form =
SkilledstaffdemandstageForm(request.POST or None, instance =
skilledstaffdemand_stage)
                if request.method == "POST" and
skilledstaffdemand_stage_form.is_valid():
                    skilledstaffdemand_stage_form.save()
                    return redirect('skilledstaffdemandview', id =
skilledstaffdemand_stage.skilledstaffdemand.id)
            else:
                skilledstaffdemand_stage_form = None
            return render(request, "skistaff/editstage.html", {"form":
skilledstaffdemand_stage_form })

```

**ДОДАТОК В**  
**Слайди презентації**

Міністерство освіти і науки України  
Національний університет «Запорізька політехніка»  
Кафедра програмних засобів

Дипломна кваліфікаційна робота бакалавра на тему:

## РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ SOFTWARE DEVELOPMENT TO RECRUIT SKILLED STAFF

Виконав студент групи КНТ-211  
Керівник: кандидат технічних наук

Данило СМЬОРДОВ  
Олена ШИТІКОВА

2025

Рисунок В.1 – Титульний слайд

## ОБ'ЄКТ, ПРЕДМЕТ, МЕТА РОБОТИ

- Об'єкт роботи – процес підбору кваліфікованого персоналу.
- Предмет роботи – програмне забезпечення підбору кваліфікованого персоналу.
- Мета роботи – розробити програмне забезпечення підбору кваліфікованого персоналу для надання можливості реалізувати вибір потрібних працівників.

2

Рисунок В.2 – Об'єкт, предмет, мета роботи

## ЗАВДАННЯ РОБОТИ

- аналіз стану проблеми розробки програм підбору кваліфікованого персоналу;
- проектування програми підбору кваліфікованого персоналу;
- реалізація, тестування програми підбору кваліфікованого персоналу.

3

Рисунок В.3 – Завдання роботи

## ПОРІВНЯННЯ ПРОГРАМ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ

| Критерій порівняння                                        | Indeed | Work.ua | Розробка |
|------------------------------------------------------------|--------|---------|----------|
| Пошук вакансій                                             | +      | +       | +        |
| Збереження вакансії                                        | +      | +       | +        |
| Перегляд сторінки працедавця                               | +      | +       | +        |
| Підписування на працедавців                                | +      | –       | +        |
| Реагування на працедавців                                  | +      | –       | +        |
| Підтвердження працедавцем даних анкети кандидата (резюме)  | –      | –       | +        |
| Реалізація багаторічного відбору кваліфікованого персоналу | –      | –       | +        |
| Перевірка вмінь заявників                                  | –      | –       | +        |
| Створення власної анкети за професією (резюме)             | +      | +       | +        |
| Взаємодія з рекрутинговими компаніями                      | –      | –       | +        |

4

Рисунок В.4 – Порівняння програм підбору кваліфікованого персоналу

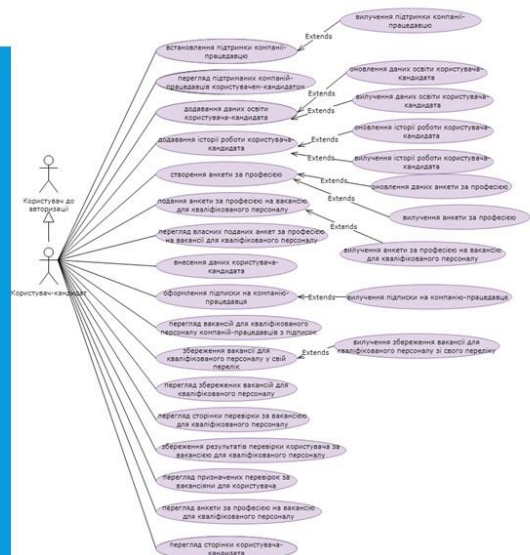
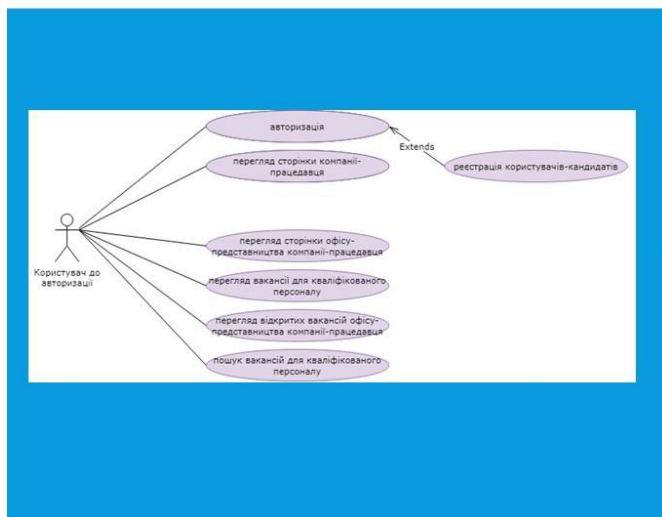
## ПОРІВНЯННЯ ЗАСОБІВ РОЗРОБКИ ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ

| Критерій порівняння       | FastAPI                                                                             | Django                                                                           |
|---------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Основна направленість     | Мікросервіси                                                                        | Великі (можливо середні) вебзастосунки                                           |
| Обмеженість часу розробки | Потребує менше часу вивчення засобів за рахунок меншої кількості вбудованих засобів | Направлений на швидке прототипування та швидку розробку                          |
| Вбудовані засоби          | WebSocket, підтримка різних форматів даних (JSON, XML, Protobuf, GraphQL)           | DjangoAdmin, об'єктно-реляційне відображення, рушій шаблонів, фреймворк для форм |
| Продуктивність            | Висока                                                                              | Комплексно висока                                                                |

5

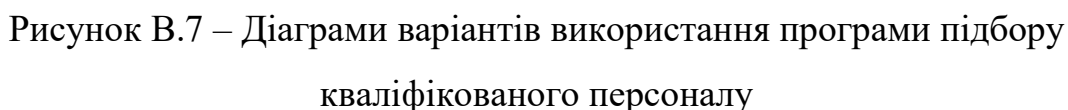
Рисунок В.5 – Порівняння засобів розробки програми підбору кваліфікованого персоналу

## ДІАГРАМИ ВАРІАНТІВ ВИКОРИСТАННЯ ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ

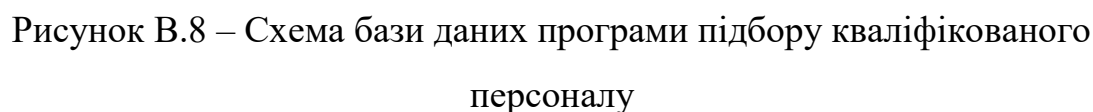


6

Рисунок В.6 – Діаграми варіантів використання програми підбору кваліфікованого персоналу



## СХЕМА БАЗИ ДАНИХ ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ



## СХЕМА ФУНКЦІОНУВАННЯ ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ

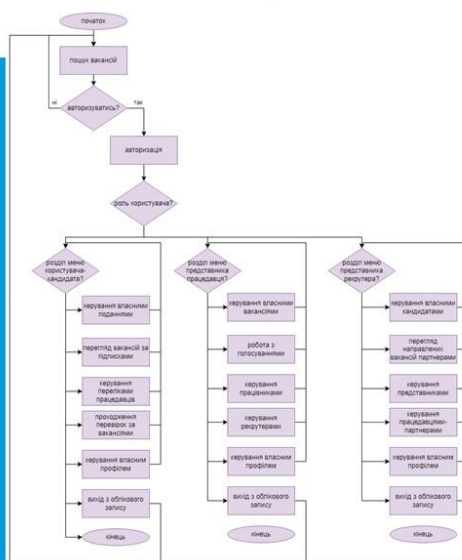


Рисунок В.9 – Схема функціонування програми підбору кваліфікованого персоналу

## ІНТЕРФЕЙС ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ

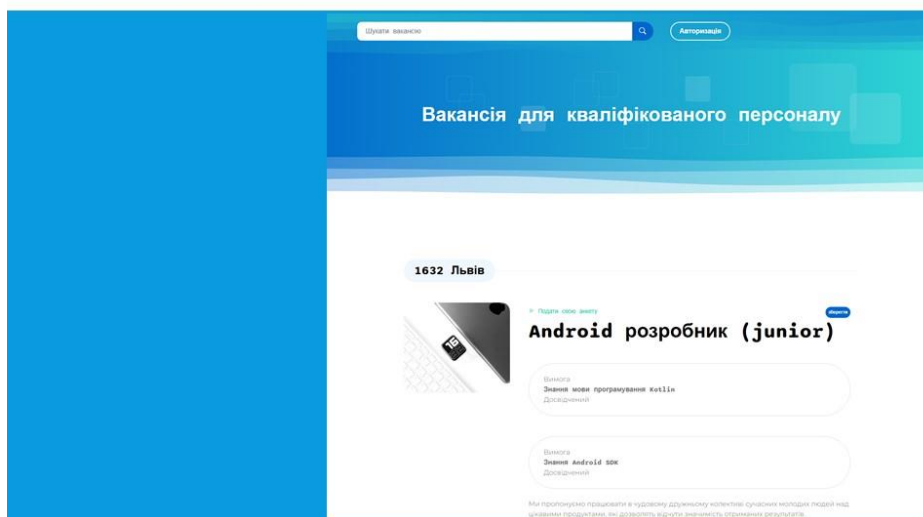
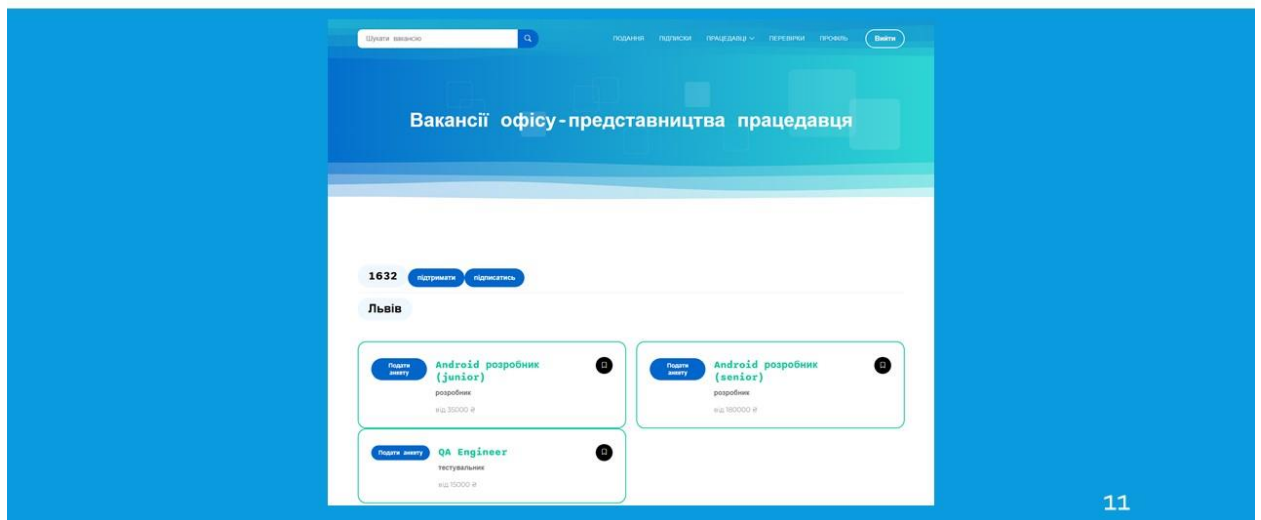


Рисунок В.10 – Інтерфейс програми підбору кваліфікованого персоналу

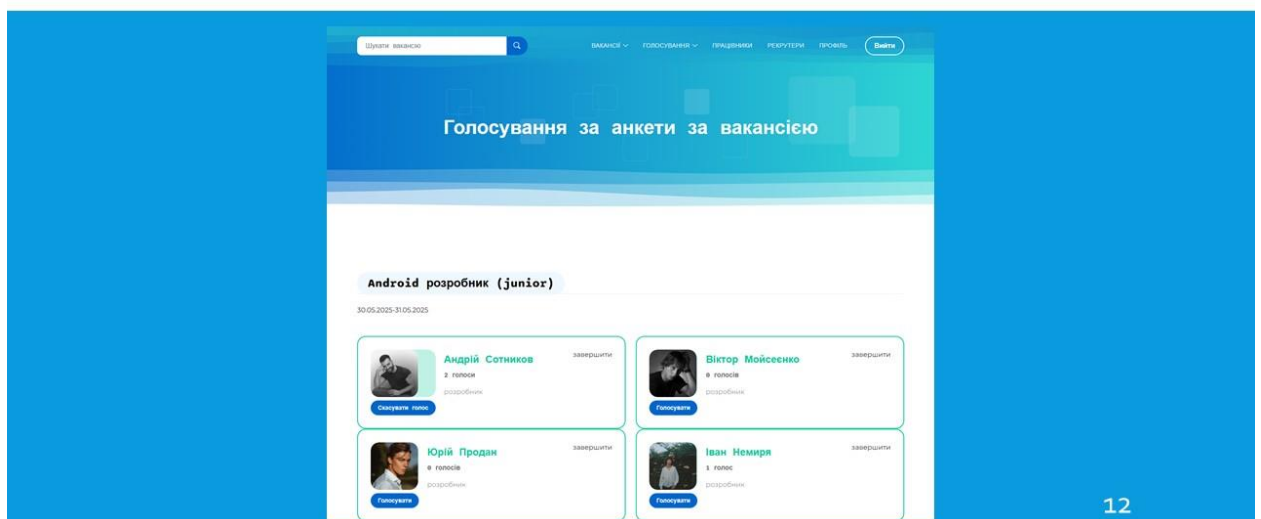
## ІНТЕРФЕЙС ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ



11

Рисунок В.11 – Інтерфейс програми підбору кваліфікованого персоналу

## ІНТЕРФЕЙС ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ



12

Рисунок В.12 – Інтерфейс програми підбору кваліфікованого персоналу

## ІНТЕРФЕЙС ПРОГРАМИ ПІДБОРУ КВАЛІФІКОВАНОГО ПЕРСОНАЛУ

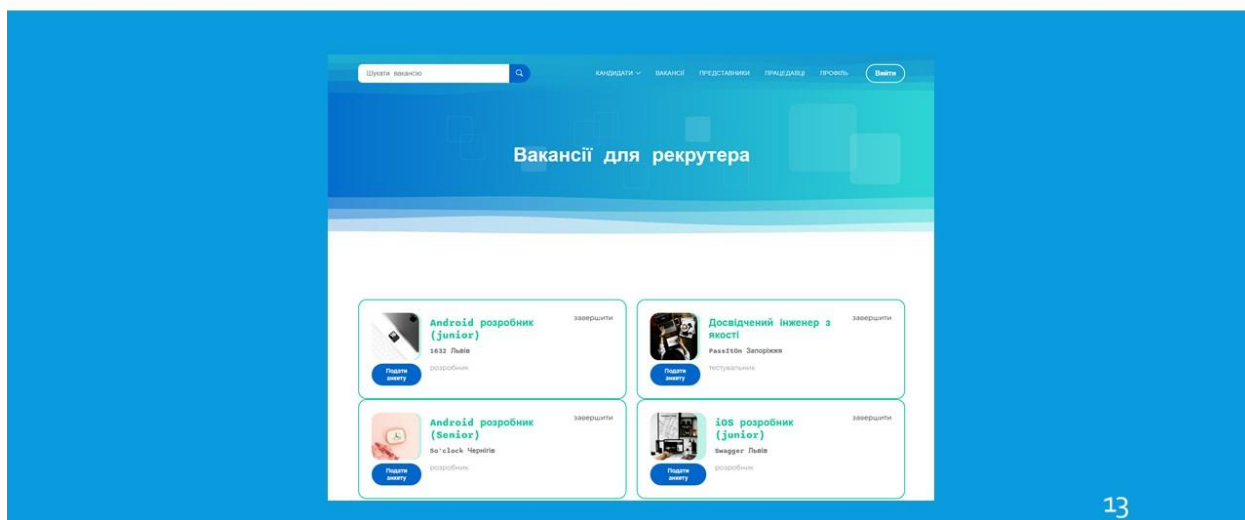


Рисунок В.13 – Інтерфейс програми підбору кваліфікованого персоналу

## ВИСНОВКИ

Програмне забезпечення підбору кваліфікованого персоналу базується на наступних концептуальних можливостях:

- пошуку кваліфікованого персоналу зі створенням відповідних вакансій на основі відгуків від самих кандидатів;
- надання пропозицій щодо кандидатів для вакансій рекрутинговими компаніями, які повинні мати своє представництво в програмі;
- реалізації багатоетапного відбору кандидатів для визначення вибору кваліфікованого персоналу з підтримкою голосування на цих етапах представниками компанії-працедавця;
- забезпечення базової перевірки кандидатів за поданими заявками на вакансії на основі автоматизованого тестування.

Програма розроблена, щоб забезпечити для компаній-працедавців підбір кваліфікованого персоналу на основі пошуку такого персоналу компаніями на основі визначення вакансій, пошуку вакансій кандидатами та організації відбору кваліфікованого персоналу через проведення етапів такого відбору з голосуванням представників компанії-роботодавця, проведенням за необхідності перевірок вмінь і знань кандидатів на вакансії для кваліфікованого персоналу на основі тестування.

14

Рисунок В.14 – Висновки