

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ

Інструментально-аналітична система рекрутингу

Галузь знань 12 «Інформаційні технології»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма «Комп'ютерні науки»
Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи КН- 42

Панасюк М.Б.

(прізвище та ініціали)

Керівник Доманецька І.М.

(прізвище та ініціали)

к.т.н., доцент

(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*
Протокол № 11 від 06.06.2022 р.
зав. кафедри _____ доц. Іларіонов О.Є.

Київ – 2022

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.

“___” _____ 2022 р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Панасюку Миколі Борисовичу


(прізвище, ім'я, по батькові)


1. Тема проекту (роботи)
«Інструментально-аналітична система рекрутингу»
затверджена протоколом засідання кафедри від « 23 » грудня 2021 р. № 4
2. Термін здачі студентом закінченого проекту (роботи) 29 травня 2022 року
3. Вихідні дані до проекту (роботи)
Рекомендація кращого кандидата з вхідного масиву кандидатів.
Вектори локальних та глобальних пріоритетів методу аналізу ієрархій.
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
Аналіз особливостей задачі прийняття рішення у контексті вибору кандидата на вакансію,
проектуювання програмного забезпечення, розробка застосунку.
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
 1. Постановка задачі: мета, об'єкт, предмет, вимоги, функціональний аналіз (3)
 2. Предметна область: огляд існуючих рішень, огляд задачі прийняття рішення (4)
 3. Проектні рішення вибору кращого кандидата (7)
 4. Програмна реалізація інструментально-аналітичної системи рекрутингу (4)

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 15 лютого 2022 року

Керівник  / Доманецька І. М. /
(підпис) (ПІБ)

Завдання прийняв до виконання  / Панасюк М. Б. /
(підпис) (ПІБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1	Опрацювання літератури	15.02 – 27.02	
2	Робота над розділом 1. Аналітичний огляд задачі прийняття рішення у контексті вибору кандидата на вакансію. Постановка задачі	27.02 – 08.03	
3	Робота над розділом 2. Проектування програмного забезпечення	08.03 – 08.04	
4	Робота над розділом 3. Розробка застосунку	08.04 – 13.05	
5	Робота над оформленням пояснювальної записки	13.05 – 25.05	
6	Робота над презентацією	25.05 – 29.05	

Студент-дипломник  / Панасюк М. Б. /
(підпис) (ПІБ)

Керівник випускної кваліфікаційної роботи  / Доманецька І. М. /
(підпис) (ПІБ)

АНОТАЦІЯ

Дипломна робота викладена на 76 сторінках, містить 3 розділи, 45 ілюстрацій, 10 таблиць, 27 джерел в переліку посилань.

Об'єктом дослідження роботи є процес підбору персоналу на підприємстві.

Предметом дослідження роботи є методи прийняття рішень для вибору найкращого кандидата із запропонованих.

Метою роботи є розробка інструментально-аналітичної системи для допомоги експертам та рекрутерам у виборі найкращого кандидата на посаду та забезпеченні справедливої конкуренції серед претендентів.

У першому розділі проведено аналіз предметної області рекрутингу, обрано математичний метод для розв'язання поставленого завдання, проаналізовано існуючі рішення на ринку. У другому розділі спроектовано функціональну структуру системи, розроблено алгоритми методу аналізу ієрархій, структуру базу даних для системи та інтерфейсні рішення застосунку. У третьому розділі наведено інформацію про використані програмні засоби у дипломному проекті, протестовано основні функції готового застосунку та описано інструкцію користування програмним застосунком.

Ключові слова: рекрутинг, метод аналізу ієрархій, web-застосунок, Flask, Python.

ANNOTATION

The thesis is presented on 76 pages, contains 3 sections, 45 illustrations, 10 tables, 27 sources in the list of references.

The object of research is the process of personnel selection at the enterprise.

The subject of the study is the methods of decision-making to select the best candidate from the proposed.

The aim of the work is to develop an instrumental-analytical system to help experts and recruiters in choosing the best candidate for the position and ensuring fair competition among applicants.

In the first section the analysis of the subject area of recruitment is carried out, the mathematical method for the decision of the set task is chosen, the existing decisions in the market are analyzed. In the second section the functional structure of the system is designed, the algorithms of the method of hierarchy analysis, the structure of the database for the system and the interface solutions of the application are developed. In the third section provides information about the software used in the thesis project, tested the main functions of the finished application and describes the instructions for using the software application.

Keywords: recruitment, method of analysis of hierarchies, web-application, Flask, Python.

ЗМІСТ

ВСТУП	9
Розділ 1 Аналітичний огляд задачі прийняття рішення у контексті вибору кандидата на вакансію. Постановка задачі.....	10
1.1 Сутність задачі рекрутингу	10
1.2 Узагальнена технологія проведення відбору персоналу	14
1.3 Аналіз існуючих математичних моделей та методів для вирішення задачі рекрутингу	15
1.3.1. MAI (Метод Аналізу Ієрархій).....	16
1.3.2. Метод аналітичних мереж.....	20
1.3.3. Прийняття рішення в умовах повної невизначеності.....	22
1.4 Аналіз існуючих рішень на ринку ІТ.....	23
1.5 Постановка задачі	26
Розділ 2 Проектування інструментально-аналітичної системи рекрутингу	30
2.1 Проектування функціональної структури ІАС рекрутингу	30
2.2 Розробка алгоритмів методу аналізу ієрархій	35
2.3 Проектування БД застосунку	37
2.3.1 Аналіз предметної області.....	37
2.3.2 Даталогічна модель бази даних.	37
2.3.3 Фізична модель бази даних	41
2.4 Розробка інтерфейсних рішень	44
Розділ 3 Розробка програмного застосунку ІАС рекрутингу	47
3.1 Опис програмного забезпечення використаного при розробці ІАС	47
3.2 Структура програмного забезпечення ІАС рекрутингу	49

3.3	Тестування коректності роботи застосунку.....	52
3.4	Опис роботи застосунку.....	56
3.4.1	Опис роботи підсистеми кандидата	56
3.4.2	Опис роботи підсистеми експерта.....	57
3.4.3	Опис роботи підсистеми рекрутера.....	60
	ВИСНОВКИ.....	62
	Список використаних джерел	63
	ДОДАТОК 1	66

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ІАС – інструментально-аналітична система.

ЗМІ – засоби масової інформації.

ЗВО – заклади вищої освіти.

СППР – система підтримки прийняття рішень.

МАІ – метод аналізу ієрархій.

МПП – матриця попарних порівнянь.

МAM – метод аналітичних мереж.

БД – база даних.

СУБД/СКБД – система управління (керування) базами даних.

ВСТУП

Питання підбору якісного персоналу завжди було актуальним та затребуваним, адже щодня сотні компаній шукають у свою команду досвідчених кваліфікованих співробітників для досягнення нових висот та цілей.

Безпосередній удар при вирішенні даного питання приймає на себе рекрутер – спеціаліст з пошуку персоналу. На його плечі покладається завдання знайти, опитати, відранжувати та обрати найкращого кандидата на ту чи іншу посаду. Найскладнішим з переліченого є опитати та відранжувати кандидатів, адже з'являється необхідність попарно ретельно порівнювати кожного кандидата з кожним аби не пропустити важливі деталі. А як же бути, якщо кількість кандидатів на посаду переважає за десятком? Об'єктивно оцінити стільки людей по кожному з численних критеріїв стає непосильною задачею, то ж на допомогу приходять інформаційні технології.

На ринку уже існують схожі програми, які частково вирішують дане питання, але вони або не націлені на вирішення саме задачі підбору персоналу, або коштують суттєвих коштів, або не відповідають запитам, які ставляться на сьогоднішній день рекрутерами.

Таким чином, розробка інструментально-аналітичної системи рекрутера є виправданою та актуальною задачею.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЗАДАЧІ ПРИЙНЯТТЯ РІШЕННЯ У КОНТЕКСТІ ВИБОРУ КАНДИДАТА НА ВАКАНСІЮ. ПОСТАНОВКА ЗАДАЧІ

1.1 Сутність задачі рекрутингу

Підбір персоналу - це процес активного пошуку, пошуку та найму кандидатів на конкретну посаду чи роботу. Під визначенням «підбір персоналу» мається на увазі весь процес найму, від початкової стадії, тобто пошуку, до інтеграції рекрута, який найбільшим чином підходить на вакантну посаду, в компанію. Процес найму складається з наступних етапів: виявлення, залучення, проведення співбесід, відбір, прийняття на роботу, навчання та адаптація нових працівників. Підбором персоналу може займатися як одна особа, так і ряд працівників, це залежить від того, яку кількість нових працівників шукають, а також від розмірів самої організації. Так, великі організації, які мають 250 та більше найманих працівників, можуть мати цілі відділи рекрутерів, а малі - лише одного рекрутера. Крім того, на практиці часто спостерігається така картина, що організації процес підбору персоналу передають на аутсорсинг, тобто на виконання спеціальним компаніям, які є спеціалістами в даній справі. Компанії здійснюють пошук кандидатів на вакантні посади за допомогою оголошень, дошки вакансій, сайтів пошуку роботи, соціальних мереж тощо. Для ефективнішого пошуку найкращих кандидатів, компанії часто використовують спеціальне програмне забезпечення для підбору персоналу, що значно полегшує весь цей процес.

Рекрутинг вважається одним із найдієвіших інструментів ефективного проведення кадрової політики в організації та розглядається у контексті як комплексний, так і індивідуальний підхід до виконання функцій з оцінювання посади, створення профілю компетенцій, визначення оптимальних каналів пошуку персоналу, інструментарію оцінювання, презентації кандидатів, супроводження у період адаптації [1, с. 185].

Рекрутинг, як професія, існує вже дуже багато років, навіть тисячоліть. Ще стародавні єгиптяни, а також класична грецька та римська культури користувалися послугами рекрутерів, для того щоб віднайти солдатів для військової служби. У двадцятому столітті під час і після Другої світової війни заходи з підбору кандидатів на певні посади здійснювалися за допомогою розміщення реклами в газетах, дошках оголошень і агітації в публічних місцях, де перебуває велика кількість людей, наприклад, на вокзалах. Заявки можна було подавати поштою. У другій половині двадцятого століття, коли відбулося масове телефонів, пошук кандидатів і проведення співбесід почав відбуватися значно швидше, а ніж подорожами чи поштою. А з приходом комп'ютерів в повсякденному житті людей, рекрутери, клієнти та кандидати почали спілкуватися та взаємодіяти за допомогою електронної пошти, прете і далі продовжували користуватися традиційною поштою, телефонними дзвінками та особистими зустрічами [2].

На даний час спостерігаються такі основні тенденції розвитку рекрутингу: зниження вагомості реклами вакансій на job-порталах та корпоративних веб-сайтах (кандидати хочуть отримувати лише правдиву інформації про їхніх потенційних роботодавців); робота за первинним пошуком претендента на вакантну посаду є значно простішою, адже профіль кандидата у соціальних мережах є більш правдиво інформативним, ніж його резюме; з кожним роком усе більше посилюється обмеженість ресурсів на пошук і підбір персоналу, що вимагає від рекрутингових агенцій знаходити найкращих клієнтів швидше, ніж конкуренти; змінюються підходи до рекрутингу, щодо випускників та студентів (для визначення очікувань та побажань майбутніх молодих спеціалістів організуються маркетингові дослідження); малоефективні джерела все швидше відходять на другий план (такі ресурси, як дошки оголошень, Facebook та ярмарки вакансій, не дають можливості знайти високоякісних спеціалістів); з'являється потреба в утриманні співробітників (висококваліфіковані фахівці знають собі ціну та не бояться шукати нову, кращу роботу); гостро стоїть

проблема недостатньої кількості висококваліфікованих рекрутерів, що призводить до підвищення конкуренції за таланти [3, с. 133].

Наведемо перелік основних обов'язків рекрутера:

- Розробка та реалізація кадрової політики та стратегій, щодо підбору персоналу;
- Пошук та залучення кандидатів, що здійснюється за допомогою баз даних, соціальних мереж тощо;
- Проведення співбесід і відбір кандидатів на вакантні посади.

У сучасних реаліях можемо виділити головні переваги застосування рекрутингу на практиці:

- Керівник організації, тобто роботодавець, не витрачає власний час на підбір кандидатів, тому що попередня співбесіда за його вимогами проводиться командою рекрутерів;
- Рекрутинг дає змогу з більшою вірогідністю підібрати найкращого кандидата, відповідно до вимог виконуваної роботи, що в результаті економить час, зусилля та кошти;

Також необхідно розглянути процес рекрутингу з декількох ракурсів. З одного боку, рекрутинг – це процес створення баз даних, з іншого боку, рекрутинг - це розроблення комплексу заходів, щодо залучення і первинного підбору персоналу. Якщо розглядати цей процес зі сторони бази даних, то варто приділити увагу повному та чіткому опису вакансії, що повинен містити назву посади, опис основних обов'язків, інформацію про винагороду й пільги, а також рекрутерську інформацію. Така інформація є дуже цінною і ключовою під час відбору кандидата, адже вона чітко дає характеристику про його біографію, освіту, професійно-кваліфікаційні навички, досвід роботи, фізичні дані, особистісні риси і здібності, інтелект, інтереси, а також мотивацію до працевлаштування та розвитку. А якщо ж розглядаємо це як розроблення комплексу заходів, щодо залучення і первинного підбору персону – то основними завданнями рекрутингу є виявлення потреби в працівникові, аналіз

ринку праці, залучення персоналу, відсіювання за допомогою співбесід, професійне оцінювання за допомогою психологічних методів (рис.1.1) [4].

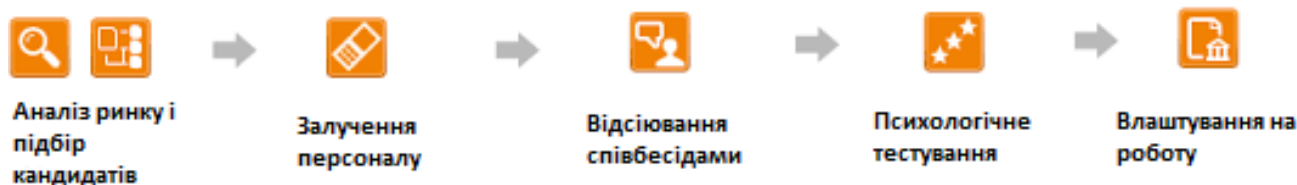


Рисунок 1.1 – Схема рекрутингу

Ще існує класифікація, відповідно до якої, рекрутинг поділяють на зовнішній та внутрішній. Внутрішній рекрутинг – це діяльність відповідного підрозділу кадрової служби з пошуку, підбору, відбору потенційних кандидатів та їх адаптації. Існують дві основні форми внутрішнього найму - підвищення та переміщення. Підвищення по службі є найбільш поширеною формою найму; це дає змогу співробітникам знаходитися в постійному розвитку своїх навичок, в той час як компанія зберігає таланти. Внутрішні переміщення теж дозволяють співробітникам покращувати свої навички, проте більш стороннім чином, що не передбачає збільшення відповідальності. Зовнішній, відповідно, передбачає підбір персоналу за межами компанії. Зовнішній кандидат - це потенційний співробітник, який наймається за межами компанії. Зовнішні кандидати складають більшість претендентів на роботу на більшість посад. Очікується, що вони принесуть певні навички, необхідні для роботи на посаді, на яку наймають, а також нові перспективи та підходи, що принесуть користь для компанії. Зовнішні кандидати шукають нову роботу з багатьох причин, зазвичай це кар'єрне зростання, вища заробітна плата, зміна життєвої ситуації, наприклад переїзд, тощо [5].

Де ж шукати джерела зовнішнього рекрутингу? Кандидатів ззовні зазвичай набирають за допомогою наступних джерел:

- Онлайн агрегатори вакансій
- Соціальний капітал (пошук через знайомих)
- ЗМІ
- Соціальні мережі

- Переманювання спеціалістів від компаній-конкурентів
- ЗВО

Варто зазначити, що рекрутинг можна поділити на такі види:

- Хедхантинг – переманювання спеціалістів від компаній-конкурентів;
- Масовий набір. Зазвичай такий підхід використовується, коли до персоналу немає особливих вимог (наприклад підбір персоналу в супермаркет);
- Класичний рекрутинг (без обмеження в джерелах пошуку);
- Е-рекрутинг – пошук за допомогою мережі Інтернет (соціальні мережі, агрегатори вакансій, тощо);

Отже, ефективне застосування рекрутингу сприяє: стабілізації ринку праці; досягненню стратегічних цілей організацій-роботодавців; підвищенню мобільності і конкурентоспроможності працівників. Аби підбір персоналу відбувався успішно, потрібно шукати кадри і на внутрішньому, і на зовнішньому ринку праці. За своєю суттю рекрутинг - це нескладна концепція - вона охоплює пошук та підбір кандидатів і, відповідно, їх найм для закриття вакантних посад. Однак на практиці ефективний рекрутинг поєднує в собі трохи мистецтва з наукою. Це вимагає впровадження певних процесів, які допоможуть досягнути необхідних результатів, з одного боку. А з іншого боку, для того, щоб знайти свого ідеального кандидата, потрібне нестандартне та креативне мислення.

1.2 Узагальнена технологія проведення відбору персоналу

Підбір персоналу певним чином творчий процес, тому часто технологія відбору кандидатів у кожного менеджера своя. Незмінним залишається лише той факт, що кандидат повинен мати кращі та сильніші якості, які важливі для компанії, за якості інших претендентів. А які саме якості вважати важливими у більшій степені – інше питання, відповідь на яке зможе надати роботодавець з рекрутером.

Абстрагуємося від питання, як необхідно проводити оцінку кандидата на посаду та наведемо послідовний перелік дій при відборі персоналу. Звичайно,

процес пошуку та підбору персоналу є унікальним для кожної організації, або навіть кожного рекрутера, проте існує 15 основних етапів процесу найму [6]:

- | | |
|---|----------------------------|
| 1. Визначення потреби в персоналі; | 9. Оцінка заявника; |
| 2. Складання плану найму; | 10. Перевірка фону; |
| 3. Здійснення опису роботи; | 11. Рішення; |
| 4. Рекламу позиції; | 12. Перевірка довідки; |
| 5. Проведення набору; | 13. Пропозиція про роботу; |
| 6. Перегляд заявок; | 14. Наймання; |
| 7. Телефонне інтерв'ю/Первинний скринінг; | 15. Онбординг. |
| 8. Проведення інтерв'ю; | |

Підбір ідеального кандидата це не завжди легко, цей процес зазвичай є дуже клопітким, тому до нього потрібно підходити ретельно. Іноді бувають такі випадки, коли неможливо логічно обґрунтувати вибір, коли відкинтий варіант в цифрах краще, але він нам не підходить. Комп'ютер не може, навіть з сучасними технологіями, досконало оцінити людські фактори, які неозброєним оком може побачити експерт. Тому перелік вимог, їх ранжування по важливості та кінцеве затвердження на посаду краще залишити на людину, що є експертом в даній справі.

1.3 Аналіз існуючих математичних моделей та методів для вирішення задачі рекрутингу

Для початку вивчимо основну концепцію задачі рекрутингу. При відборі персоналу ми маємо справу з різними людьми з необхідними нам навичками, які у кожного розвинені у різній ступені. Тобто задача зводиться до того, що постає необхідність проранжувати кандидатів за рівнем їхніх навичок. З цією задачею

нам допоможуть справитися різноманітні математичні методи, а також нейронні мережі. Нажаль нейронні мережі для навчання потребують об'ємної бази даних, якої немає у вільному доступі, адже така інформація про співробітників у компаніях є прихованою. Тож цей метод ми розглядати і використовувати не будемо. Звернемося до простіших, але не менш дієвих методів, таких як:

- МАІ (Метод Аналізу Ієрархій)
- Метод аналітичних мереж
- Прийняття рішень в умовах повної невизначеності

Звісно методів, які б допомогли вирішити дану задачу існує більше, але зупинимося та розберемо детальніше ці три.

1.3.1. МАІ (Метод Аналізу Ієрархій)

Одною із основних переваг МАІ перед іншими методами прийняття рішень є те, що він дозволяє враховувати людський фактор при підготовці прийняття рішення. В рамках методу аналізу ієрархій немає загальних правил для формування структури моделі прийняття рішення. Метод дозволяє врахувати цю обставину за допомогою узгоджених різних думок, за допомогою визначення їх пріоритетів [7].

Збір інформації для підтримки прийняття рішення відбувається з допомогою процедури парних порівнянь. МАІ надає великі можливості для виявлення протиріч в даних. Для мінімізації цих протиріч експерт переглядає та коректує вхідну інформацію. Процедура парних порівнянь і процес перегляду результатів порівнянь для мінімізації суперечностей часто є трудомісткими. Обчислення ваги критеріїв та альтернатив здійснюється за шкалою Сааті наведеною в таблиці №1.1.

Таблиця №1.1 Шкала Сааті та її тлумачення.

Рівень переваги	Визначення	Пояснення
1	Переваги немає	Альтернативи впливають на ціль однаково

Продовження таблиці №1.1

2	Перевага мінімальна	
3	Перевага середня	Одна альтернатива впливає на ціль трохи більше за іншу
4	Перевага більше, ніж середня	
5	Перевага сильна	Одна альтернатива впливає на ціль значно більше за іншу
6	Більш ніж сильна перевага	
7	Перевага дуже міцна	Домінування однієї альтернативи над іншою по впливу на ціль дуже міцна та очевидна
8	Дуже, дуже міцна перевага	
9	Перевага максимальна	Сприяння однієї з альтернатив по досягненню цілі у порівнянні з іншою альтернативою є найвищим можливим порядком.

Результатом попарних порівнянь критеріїв та альтернатив є матриця попарних порівнянь виду:

$$\begin{matrix} & A_1 & \dots & A_n \\ \begin{matrix} A_1 \\ \dots \\ A_n \end{matrix} & \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \dots & \dots & \dots \\ a_{n,1} & \dots & a_{n,n} \end{bmatrix} & & \end{matrix} \quad (1.1)$$

Наступний крок полягає у визначенні вагових коефіцієнтів з попарних порівнянь. Вектор вагових коефіцієнтів складається з нормованого середнього

геометричного W_i по кожному рядку з матриці попарних порівнянь. Розрахунок W_i відбувається за формулою:

$$W_i = \frac{\sqrt[n]{\prod_{j=1}^n a_{i,j}}}{\sum_{i=1}^n \sqrt[n]{\prod_{j=1}^n a_{i,j}}} \quad (1.2)$$

При оцінці критеріїв та альтернатив могли бути допущені помилки. Тому є необхідність в перевірці узгодженості матриць порівнянь. Для цього із значень матриці порівнянь та значень власного вектора будується матриця E розміром n х n , кожен елемент якої

$$\epsilon_{i,j} = a_{i,j} \frac{W_j}{W_i}, \quad (1.3)$$

де $i, j = 1 \dots n$,

сумуються усі елементи матриці E

$$\lambda_{max} = \sum_{j=1}^n \epsilon_{i,j}, \quad (1.4)$$

у кожному рядку, обирається максимальне значення. Якщо $\lambda_{max} = n$, то матриця узгоджена. Якщо ж $\lambda_{max} > n$, то обчислюється індекс узгодженості

$$\mu = \frac{|\lambda_{max} - n|}{n-1}. \quad (1.5)$$

Матриця попарних порівнянь узгоджена, якщо $\mu / \mu_T < 0.1$, де μ_T це табличний індекс узгодженості (таблиця №1.2).

Таблиця №1.2. Значення табличного індексу узгодженості

Розмір матриці (n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Табличний індекс μ_T	0	0	0,52	0,89	1,11	1,25	1,35	1,40	1,45	1,49	1,52	1,54	1,56	1,58	1,59

Якщо матриця попарних порівнянь не є узгодженою, то необхідно у матриці E знайти максимальне значення $\epsilon_{i,j}$ і переглянути порівняння альтернатив A_i та A_j .

Коли відбувалося порівняння альтернатив, вони порівнювалися за кожним з критерієм. В результаті маємо k матриць попарних порівнянь альтернатив з k критеріїв, а у наслідок маємо також k векторів вагових коефіцієнтів альтернатив. З цих векторів формується матриця наступного виду:

$$\begin{matrix} A_1 \\ \dots \\ A_n \end{matrix} \begin{bmatrix} W_1^{(1)} & \dots & W_1^{(k)} \\ \dots & \dots & \dots \\ W_n^{(1)} & \dots & W_n^{(k)} \end{bmatrix}, (1.6)$$

де кожному рядку в матриці відповідає одна альтернатива. Дана матриця множиться на вектор вагових коефіцієнтів критеріїв і результатом буде вектор глобальних пріоритетів альтернатив. Альтернатива з найбільшим значенням має найбільшу перевагу. Схематична робота МАІ наведена на рисунку 1.2.

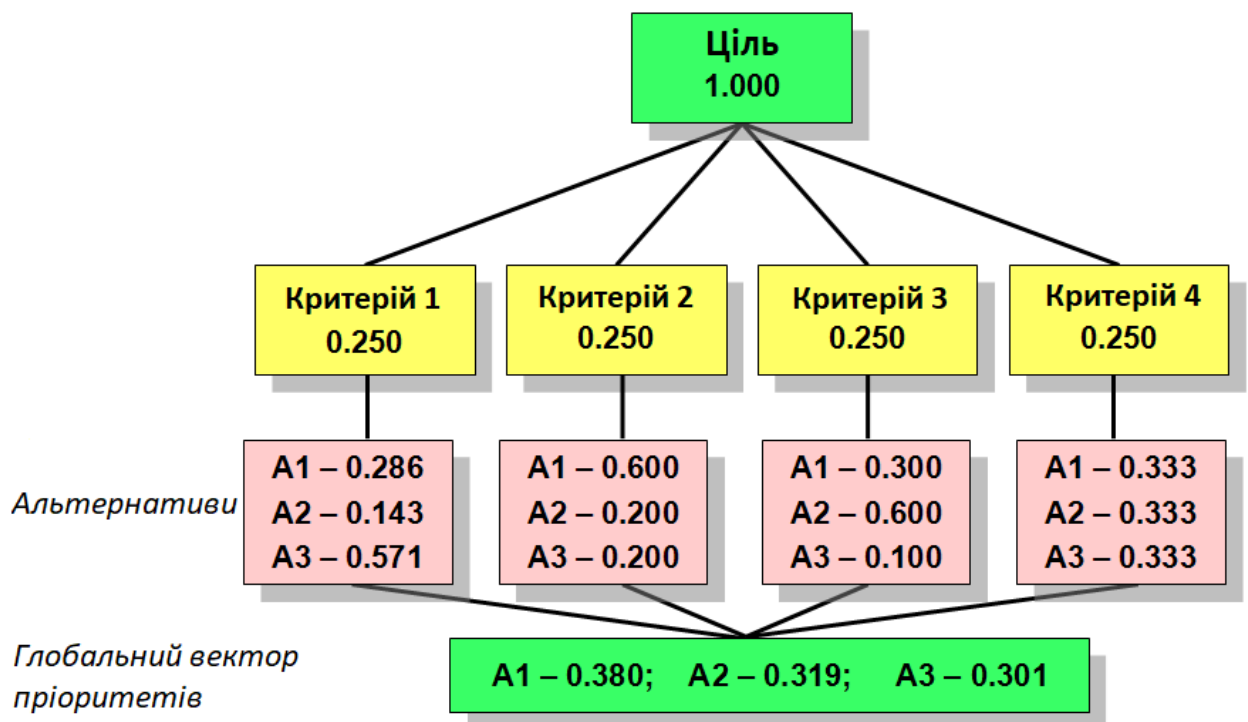


Рисунок 1.2 – Схема алгоритму роботи методу аналізу ієрархій

Схема застосування методу зовсім не залежить від сфери діяльності, в якій приймається рішення. Тому метод є універсальним, його застосування дозволяє організувати систему підтримки прийняття рішення.

Метод відбиває природний перебіг людського мислення. Він дозволяє кількісно висловити ступінь переваги за допомогою рейтингування. Це сприяє повному та адекватному виявленню переваг особи, яка приймає рішення. Крім того, оцінка міри суперечливості використаних даних дозволяє встановити ступінь довіри до отриманого результату.[8]

МАІ відноситься до методів вирішення слабоструктурованих задач прийняття рішень. Метод надає математичні методи для відображення особистих вподобань експерта в числові характеристики елементів МАІ.

Переваги МАІ:

- наочність побудованих ієрархічних моделей;
- простота інтерпретації результатів;
- можливість оцінювання альтернатив та критеріїв не тільки за кількісним, але й за якісними критеріями;
- контроль узгодженості оцінок експертом.

Недоліки МАІ:

- процедура попарної оцінки критеріїв та альтернатив трудомістка та має виконуватися вручну. [9]

1.3.2. Метод аналітичних мереж

Метод аналітичних мереж (МАН) – це узагальнення методу аналізу ієрархій. При структуруванні одної і тієї ж цілі різні люди можуть побудувати різні ієрархії. При цьому у багатьох проблемах прийняття рішення існують залежності між елементами різних рівнів ієрархії, що приводить до неможливості їх представлення у вигляді ієрархічних структур. В ієрархічних структурах важливість критеріїв впливає на пріоритети альтернатив. В структурах, відмінних від ієрархічних, важливість альтернатив впливає на пріоритети критеріїв. МАН вирішує задачі прийняття рішення шляхом виконання наступних етапів:

- 1) структурування проблеми прийняття рішення у вигляді аналітичної мережі;
- 2) використання системи парних порівнянь для підрахунку ваги компонентів структури;
- 3) ранжування альтернатив у рішенні.

Структури рішень зі зворотними зв'язками не можна лінійно впорядкувати зверху вниз, вони є мережами, що містять цикли і безлічі елементів

(компоненти), які вже не можуть називатися рівнями, а також петлі зворотного зв'язку, що показують зв'язок між елементами одного компонента (рис.4).

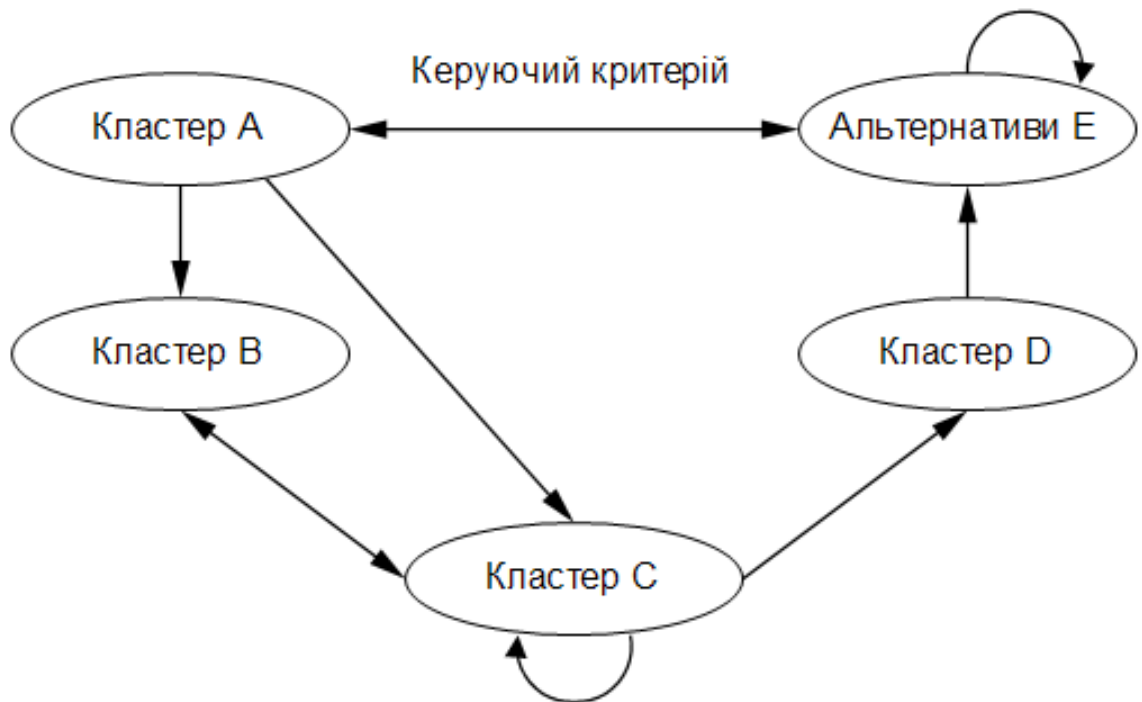


Рисунок 1.3 – Приклад мережевої структури

Визначення пріоритетів елементів у мережі, зокрема, альтернатив рішень, є складною проблемою. Оскільки наявність зворотних зв'язків приводить до виникнення циклів, і, отже, нескінченних маршрутів, то в МММ виникає необхідність застосування складнішого, ніж у МАІ, алгоритму обчислення пріоритетів. Крім того, вирішення мережевих завдань вимагає застосування обчислень з високою точністю.

Переваги МММ:

- висока точність роботи алгоритму
- метод враховує більше факторів ніж альтернативний МАІ

Недоліки МММ:

- складність алгоритмізації;
- необхідність обчислень з високою точністю. [10]

1.3.3. Прийняття рішення в умовах повної невизначеності

Прийняття рішень в умовах невизначеності полягає в тому, що ймовірності різних варіантів розвитку подій невідомі.

Методологія прийняття рішення в умовах ризику та невизначеності передбачає побудову у процесі обґрунтування ризикових рішень так званої «матриці рішень», яка має такий вигляд (табл. 1.3).

Таблиця №1.3. Вигляд «матриці рішень» у задачах прийняття рішення в умовах невизначеності.

Альтернативи	Критерії			
	B1	B2	...	Bn
A1	E11	E12	...	E1n
A2	E21	E22	...	E2n
...
An	En1	En2	...	Enn

У наведеній матриці значення $A_1; A_2; \dots; A_n$ характеризують кожен із варіантів альтернатив прийняття рішення; значення $B_1; B_2; \dots; B_n$ – вимоги до кандидата; значення $E_{11}; E_{12}; E_{1n}; E_{21}; E_{22}; E_{2n}; E_{n1}; E_{n2}; \dots; E_{nn}$ - конкретний рівень ефективності рішення, що відповідає певній альтернативі при певній вимозі. На основі зазначеної матриці розраховується найкраще альтернативних рішень за обраним критерієм.

Прийняття рішень за умов невизначеності будується на тому, що ймовірності різних варіантів ситуацій розвитку подій суб'єкту, який приймає ризикове рішення, невідомі. У цьому випадку при виборі альтернативи прийнятого рішення суб'єкт керується, з одного боку, своєю ризиковою перевагою, а з іншого — відповідним критерієм вибору з усіх альтернатив за складеною ним «матрицею рішень».

Основні критерії, що використовуються в процесі прийняття рішень в умовах невизначеності, наведені нижче:

1. Критерій Вальда (критерій "Максиміну").
2. Критерій "максимуму".
3. Критерій Гурвіца (критерій "оптимізму-песимізму" або "альфа-критерій").
4. Критерій Севіджа (критерій втрат від "мінімуму").

Переваги:

- простота інтерпретації результатів;
- можливість оцінювання альтернатив та критеріїв не тільки за кількісним, але й за якісними критеріями;
- контроль узгодженості оцінок експертом;

Недоліки:

- Можуть виникнути ситуації, коли цей метод буде справлятися гірше за наведені вище методи, оскільки він не націлений на задачу рекрутингу, хоч і може її вирішити. [11]

1.4 Аналіз існуючих рішень на ринку ІТ

1. СППР «DecisionGrid».

DecisionGrid [12] являє собою програмну оболонку для автоматизації процесу зіставлення дискретних альтернатив за багатьма критеріями. Виробник системи – компанія «Softkit Technologies Inc». За допомогою DecisionGrid є можливість аналізу і зіставлення багатьох поглядів, поданих одним або кількома користувачами. У різних сценаріях використовуються ті самі альтернативи і критерії, але можна задавати інші оцінки. Коли всі сценарії описані, можна агрегувати результати для знаходження середнього ранжування. Результати порівняння альтернатив можуть бути подані в графічному вигляді, що дає змогу краще оцінити інтервал між кожною альтернативою та її конкурентами. До складу DecisionGrid входить набір з 20 готових для використання шаблонів із

різних сфер людської діяльності (бізнес, суспільний сектор, особисте життя), які можуть бути використані для швидкої побудови таблиць порівняння.

2. Онлайн система «Business Performance Management».

Business Performance Management [13] – це безкоштовне веб-рішення, розроблене на базі методу аналізу ієрархій (МАІ), як допоміжний інструмент для процесу прийняття рішень, і інформація на ньому розповсюджується в навчальних цілях. МАІ використовується також для групового прийняття рішень. Можна зберігати повні вирішення ієрархій та використовувати їх, щоб оцінити вагу критеріїв та підкритеріїв можна оцінити до семи альтернативних рішень. Інтерфейс зрозумілий та простий, доступний тільки англійською мовою. Для того, щоб скористатись ресурсом, потрібно зареєструватись та авторизуватись.

3. Онлайн система «Decision Lens».

Онлайн система «Decision Lens» [14] призначена для підтримки процесу прийняття рішень для отримання оптимальних результатів в умовах обмеженості ресурсів. Система розроблена за підтримки Томаса Сааті. Головною особливістю системи є організація онлайн хосту для проведення віртуальних зустрічей експертів для прийняття спільних рішень протягом спеціально виділених сесій. Інтерактивний спрощений інтерфейс надає можливість проводити голосування експертів. Система підтримує створення одночасних сценаріїв; надає можливість встановлювати пріоритети на кожному рівні ієрархії шляхом попарних порівнянь; перевіряти вплив та зміни пріоритетів методом аналізу чутливості; надає можливість застосовувати бізнес-правила для розподілу бюджету, має розвинені засоби візуалізації результатів роботи. Даний комерційний продукт має вбудований персональний помічник для кожного акаунту (Client Decision Manager) та навчальну дистанційну програма для університетів.

4. Онлайн система «Expert Choice».

Онлайн система «Expert Choice» [15] також базується на класичному методі МАІ. Система надає можливість будувати багаторівневі ієрархії, проводити попарні порівняння альтернатив і критеріїв за допомогою таблиць і

графічно. Розраховує індекси узгодженості експертів і надає можливість відразу скорегувати висновки експертів і перерахувати результати. Система має можливість графічно відтворювати результати аналізу у різних перспективах, виконувати імпорту/експорту даних. Також є можливість отримати обмежену десктопну система в рамках співпраці з університетами.

5. СППР «Super Decisions».

Система «Super Decisions» [16] постачається для ОС Windows та Mac. Ця система для прийняття рішень передбачає не тільки ручне введення даних, але і завантаження із зовнішніх джерел. Але, дана система не передбачає використання експертних оцінок. Натомість, в ній наявний потужний метод аналізу мереж, який був запропонований Томасом Сааті. Система дозволяє працювати з раніше створеними і демонстраційними моделями, а також шаблонами.

Після аналізу існуючих рішень (таблиця №1.4), на ринку не виявилось застосунку, який виконував би усі потреби у підборі персоналу для автошколи. Тому розробка власної системи, що базується на МАІ є цілком виправданою та актуальною задачею. У програмі необхідно запозичити та реалізувати усі переваги проаналізованих рішень.

Таблиця №1.4. Порівняння характеристик існуючих рішень.

	Зручність	Робота з шаблонами	Безкоштовність	Графічне подання результатів	Ред. крит. бази
DecisionGrid	+	+	-	+	+
Business Performance Management	-	+	+	-	-

Продовження таблиці №1.4

Decision Lens	+	+	-	+	+
Expert Choice	+	-	-	+	+
Super Decisions	-	-	-	+	+

1.5 Постановка задачі

Об'єктом дослідження роботи є процес підбору персоналу на підприємстві.

Предметом дослідження роботи є методи прийняття рішень для вибору найкращого кандидата із запропонованих.

Метою роботи є розробка інструментально-аналітичної системи для допомоги експертам у виборі найкращого кандидата на посаду та забезпеченні справедливої конкуренції серед претендентів.

Необхідно розробити експертно-аналітичну систему для вирішення задачі підбору персоналу.

На вхід системі має подаватися інформація з бази даних про кандидатів на посаду, які пройшли співбесіду та були оцінені експертом. На виході система повинна вивести на екран інформацію про кандидата, якого вона вважає найкращим варіантом на посаду.

Застосунок розробити як web-додаток з легким для розуміння та користування інтерфейсом.

В якості методу вирішення задачі обрати метод аналізу ієрархій (МАІ) та забезпечити можливість зберігання шаблонних матриць попарних порівнянь для різних посад.

Розглянемо бізнес процес вибору кандидата на підприємстві:

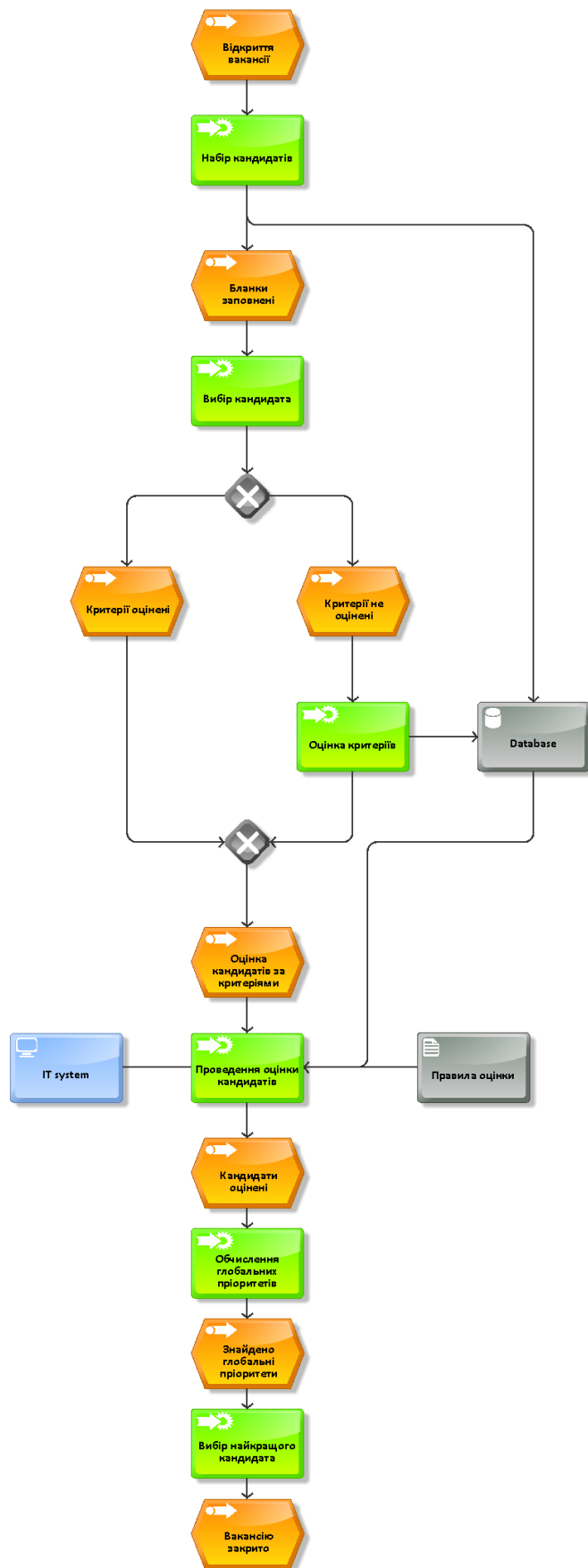


Рисунок 1.4 – Діаграма бізнес процесу «Вибір співробітника»

Узагальнена схема алгоритму МАІ:

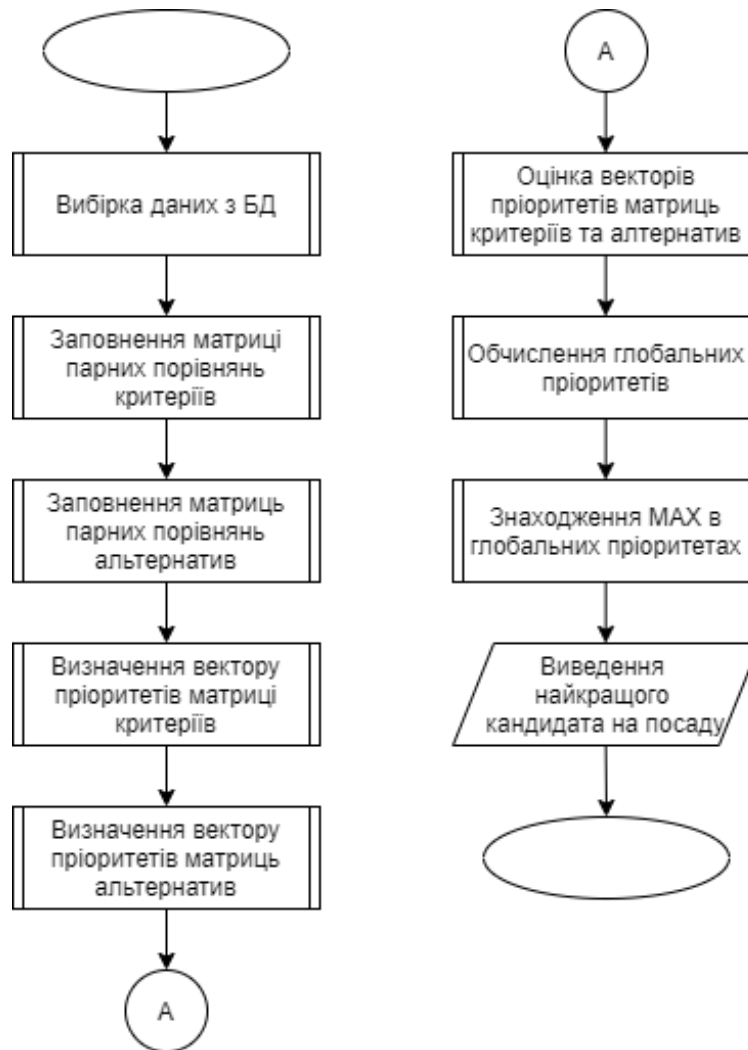


Рисунок 1.5 – Узагальнена схема алгоритму МАІ

Функціональні вимоги:

- Додавання вакансій

У програмі необхідно реалізувати можливість додавання налаштувань підбору персоналу і на інші вакансії, в тому числі і на ті, яких на підприємстві наразі немає.

- Видалення вакансій

Для підтримки незагромадженості в програмі необхідно додати функцію видалення зайвих, або ж помилково доданих, вакансій.

- Створення шаблонів оцінки для вакансії

Програма має надати можливість експерту створювати та записувати в БД матрицю попарних порівнянь критеріїв.

- Видалення шаблонів

Для підтримки незагромадженості в програмі необхідно додати функцію видалення зайвих, або ж помилково доданих, шаблонів.

- Редагування критеріальної бази

Для покращення результатів експерт має мати доступ до зміни критеріїв та матриці попарних порівнянь цих критеріїв.

- Вивантаження результатів в файл

Для формування звітів і наказів, програма повинна мати функцію формування результатів у файл.

- Графічне представлення результатів

Для прозорості відбору та наочності результатів представляти результати необхідно також у графічному вигляді.

- Генерація матриці попарних порівнянь кандидатів

Обробити і порівняти велику кількість кандидатів тяжко, тому необхідно розробити алгоритм підрахунку коефіцієнтів матриці попарних порівнянь кандидатів.

Нефункціональні вимоги:

- Українська мова застосунку

Оскільки застосунок розробляється для державної установи, використання державної мови є обов'язковою.

- Підтримка різних ОС

Експертно-аналітична система не має бути прив'язана до платформи, оскільки на підприємстві використовуються машини з різними ОС.

- Відповідність сучасним тенденціям у дизайні

Для комфортної роботи менеджерів із застосунком, дизайн повинен відповідати звичним тенденціям, які менеджер, як користувач, звик бачити щодня.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ІНСТРУМЕНТАЛЬНО-АНАЛІТИЧНОЇ СИСТЕМИ РЕКРУТИНГУ

2.1 Проектування функціональної структури ІАС рекрутингу

Розглянемо контекстну діаграму процесу роботи системи рекрутингу (рис.2.1). Основні зв'язки контекстної діаграми наведені в таблиці 2.1.

Таблиця 2.1 – Опис зв'язків контекстної діаграми.

Назва	Тип	Зовнішній об'єкт
Список усіх критеріїв	Вхідні дані	Експерт
Дані кандидата	Вхідні дані	Кандидат
Список вакансій	Вхідні дані	Експерт
Найкращий кандидат	Вихідні дані	Рекрутер



Рисунок 2.1 – Контекстна діаграма процесу роботи системи рекрутингу

З контекстної діаграми стає чітко зрозуміле основне завдання системи – відібрати найкращого кандидата з вхідного масиву анкет.

Розглянемо декомпозицію контекстної діаграми роботи системи рекрутингу (рис.2.2). Даний процес складається з наступних бізнес процесів:

1. Формування пулу критеріїв

Вхідна інформація:

- Список усіх критеріїв
- Список вакансій

Вихідна інформація:

- Вибірка критеріїв

2. Заповнення матриці попарних порівнянь критеріїв

Вхідна інформація:

- Вибірка критеріїв

Вихідна інформація:

- Матриця попарних порівнянь критеріїв

3. Заповнення анкети

Вхідна інформація:

- Дані кандидата

Вихідна інформація:

- Анкетні дані кандидата

4. Запуск алгоритму відбору персоналу

Вхідна інформація:

- Матриця попарних порівнянь критеріїв
- Анкетні дані кандидата

Вихідна інформація:

- Вектор глобальних пріоритетів

5. Формування звіту

- Звіт-рекомендація кращого кандидата на посаду

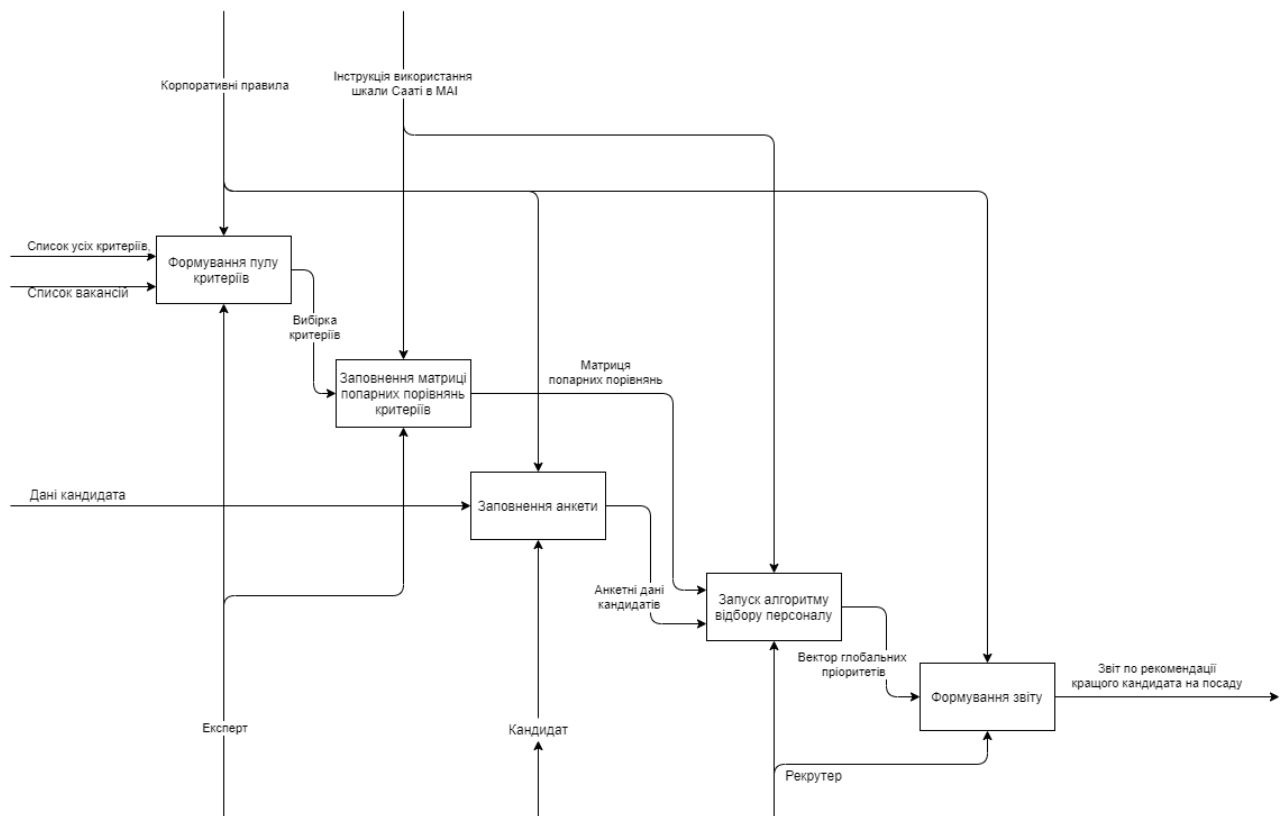


Рисунок 2.2 – IDEF0 процесу роботи системи рекрутингу

Процес роботи системи рекрутингу полягає у обробці критеріїв, формуванні матриці їх попарних порівнянь та визначенні, на скільки один критерій важливіше іншого. Який критерій вважати важливіше іншого тлумачать корпоративні правила. Цю ж процедуру необхідно провести і з кандидатами по кожному з критеріїв. Отримані матриці аналізуються за алгоритмом наведеним у попередньому розділі, який дозволяє нам отримати глобальний вектор пріоритетів. Найкращий кандидат буде відповідати найбільшому коефіцієнту у векторі глобальних пріоритетів. Вихідними даними програми є звіт-рекомендація кращого кандидата на посаду із запропонованих.

Для дослідження бізнес-процесів у системі представимо її функції у вигляді дерева (рис.2.3). Систему рекрутингу можна розділити на такі підсистем:

- Підсистема обробки анкет кандидатів;
- Підсистема експертної оцінки критеріїв;
- Підсистема формування моделі оцінювання (підсистема рекрутера).

Для кандидатів доступна лише система обробки анкет кандидатів. У ній можна обрати бажану відкрити вакансію та заповнити анкету на відбір до неї. Для кандидата також виділена можливість редагувати та видаляти свою анкету.

Підсистема експертної оцінки критеріїв розроблена для надання інструментів експертної оцінки експерту, а саме:

- Інструмент для формування пулу критеріїв під вакансію;
- Інструмент попарних порівнянь критеріїв.

У інструменті формування пулу критеріїв можна додавати як із уже внесених в базу даних критеріїв, так і додавати нові. У цьому ж інструменті передбачена можливість додавання в систему нову вакансію, яка поки відсутня в системі.

Інструмент попарних порівнянь критеріїв дозволяє заповнити матрицю попарних порівнянь критеріїв, а також редагування уже готових шаблонів.

Перший і другий інструмент в сумі дозволяють створити шаблон для відбору кандидата у підсистемі рекрутера. Експерт у цій підсистемі наділений можливістю видалення цих шаблонів, а також критеріїв і вакансій з бази даних за потреби.

Підсистема формування моделі оцінювання надає рекрутеру інструменти аналізу даних для відбору персоналу. На сторінці підсистеми виводиться основна інформація, опираючись на яку алгоритм методу аналізу ієрархій приймає рішення.

Їх можна розділити на функції роботи з вакансіями та кандидатами. Робота з вакансіями передбачає додавання нових вакансій, створення шаблонів цих вакансій і подальший запис їх в базу даних. Має бути надана можливість створення, редагування та видалення критеріальної бази шаблонів. Також необхідно надати інструменти для видалення вакансій з їх шаблонами.

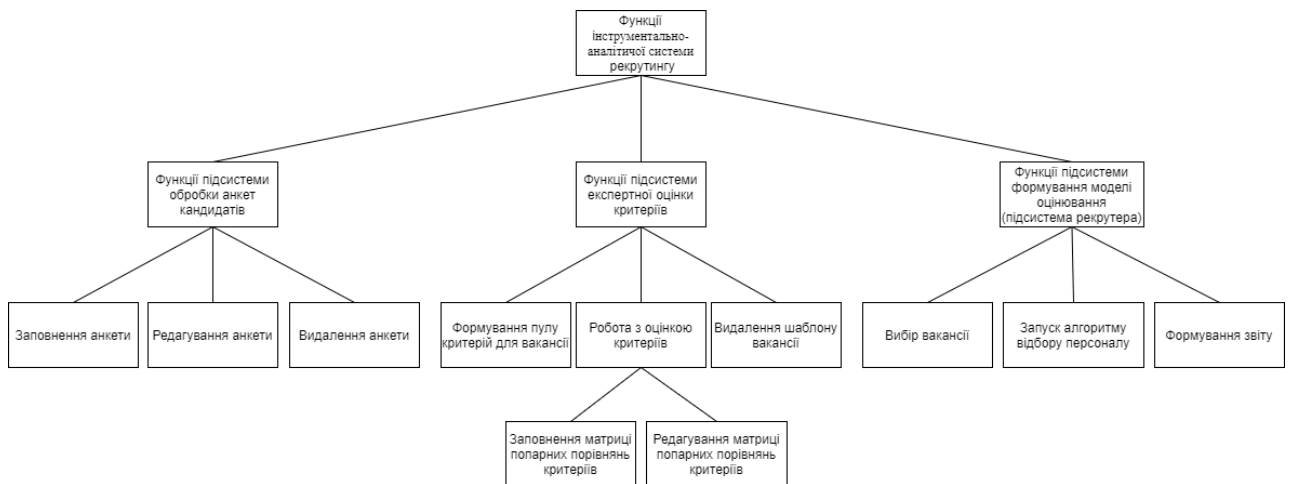


Рисунок 2.3 – Дерево функцій системи рекрутингу

Для програмної реалізації ІАС рекрутера було обрано дворівневу архітектуру Web-застосунку. При такій архітектурі БД зберігається на тій же машині, де знаходиться Web-сервер (рис 2.4).

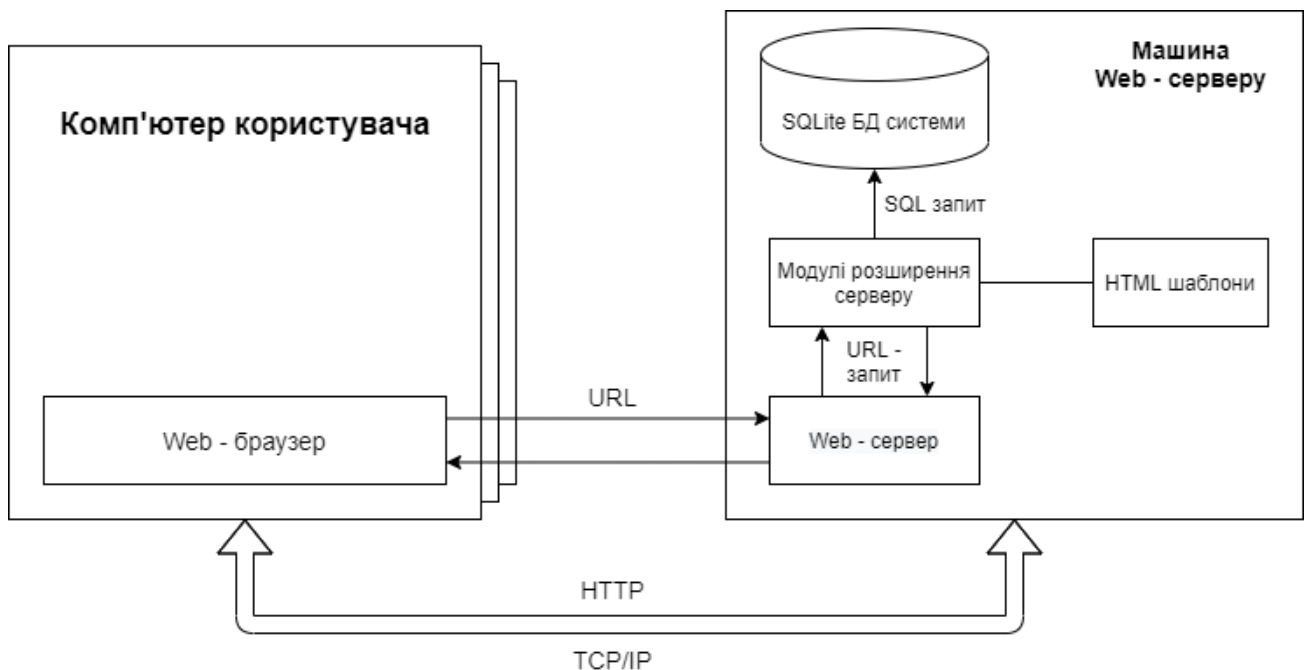


Рисунок 2.4 – Дворівнева архітектура Web-застосунку інструментально-аналітичної системи рекрутингу.

Переваги дворівневої архітектури Web-застосунку:

- Простота реалізації за допомогою фреймворків;
- Позбавлення від необхідності розробляти desktop-застосунок;
- Зручне адміністрування і конфігурування за рахунок того, що зміни не торкають клієнтський рівень.

Недоліки дворівневої архітектури Web-застосунку:

- Підвищене навантаження на сервер за рахунок того, що обробка URL-запитів, формування HTML-сторінок та виконання SQL-запитів відбувається на одному сервері;
- Низький рівень безпеки даних. [17]

2.2 Розробка алгоритмів методу аналізу ієрархій

Підсумувавши раніше наведені дані наведемо узагальнену блок-схему алгоритму Методу Аналізу Ієрархій (рис.2.5). Математична складова методу наведена у частині №3 розділу №1 цієї роботи.



Рисунок 2.5 – Узагальнена схема алгоритму МАІ

Окремо необхідно розглянути алгоритм експертних оцінок, на основі яких будуються матриці попарних порівнянь як критеріїв, так і альтернатив. Для початку критерії з анкет необхідно звести до шкали Сааті за наступною формулою:

$$y_i = (x_i - x_{min}) * \frac{9-1}{x_{max}-x_{min}} + 1; \quad (2.1)$$

де $i=0,N$, N -кількість кандидатів;

x_{min} , x_{max} – мінімальне та максимальне значення у векторі кількісних характеристик за критерієм;

x_i – значення i -го кандидата за k -м критерієм;

y_i – значення i -го кандидата за k -м критерієм у шкалі Сааті;

Цю процедуру необхідно провести за кожним з критеріїв.

Наступний крок – формування матриць попарних порівнянь:

$$a_{ij}^k = \frac{y_i^k}{y_j^k}; \quad (2.2)$$

де a_{ij}^k – значення у матриці попарних порівнянь при порівнянні i -го та j -го кандидата за k -им критерієм, $k=0,M$, M -кількість критеріїв.

У результаті отримаємо k матриць попарних порівнянь з експертними оцінками.

Підсумувавши раніше згадані дані, наведемо узагальнену блок-схему алгоритму створення експертних оцінок (рис.2.6).



Рисунок 2.6 – Узагальнений алгоритм розрахунку експертних оцінок

2.3 Проектування БД застосунку

2.3.1 Аналіз предметної області

Провівши аналіз даних, які будуть циркулювати у системі, були виділені наступні сутності:

- Критерій
- Кандидат
- Вакансія
- Експерт

Критерії у зв'язку з вакансіями утворюють ще одну сутність, а саме критерії, які притаманні якійсь вакансії. Назвемо цю сутність «Критерії – Вакансії».

Таблиця 2.2 – Зв'язки між сутностями предметної області

№ п/п	Сутності, що утворюють зв'язок	Тип зв'язку	Пояснення
1.	Критерій – Вакансія	M:N	Одна вакансія вимагає багатьох критеріїв; один критерій може фігурувати в декількох вакансіях.
2.	Критерій_Вакансія - Експерт	M:N	Один експерт порівнює між собою багато критеріїв; критерії можуть порівнювати декілька експертів.
3.	Критерій_Вакансія - Кандидат	M:N	Один критерій притаманний багатьом кандидатам; один кандидат відповідає багатьом критеріям

2.3.2 Даталогічна модель бази даних.

Даталогічна модель (ДЛМ) – модель логічного рівня, яка являє собою відображення логічних зв'язків між елементами даних незалежно від їх вмісту і середовища збереження. ДЛМ є подальшою формалізацією інфологічної моделі і являє собою по суті модель даних. Ця модель будується в термінах інформаційних одиниць, допустимих в тій СУБД, в середовищі якої буде

створюватися БД. Етап створення даної моделі називається даталогічним або логічним проектуванням. [18]

За правилами перетворення ERD моделі, у даталогічній моделі зв'язки типу М:N замінюються зв'язувальними таблицями. Також, так як оцінка критеріїв і кандидатів будуть подаватися у вигляді матриці попарних порівнянь, то у реляційній схемі з'явиться ще дві таблиці, які мають подвійні зв'язки з батьківськими таблицями, “opinion_cand”, “opinion_crit”. Таблиця, яка відповідає сутності «Експерт» уже буде зв'язуватися з даною таблицею “opinion_crit” (рис. 2.7).

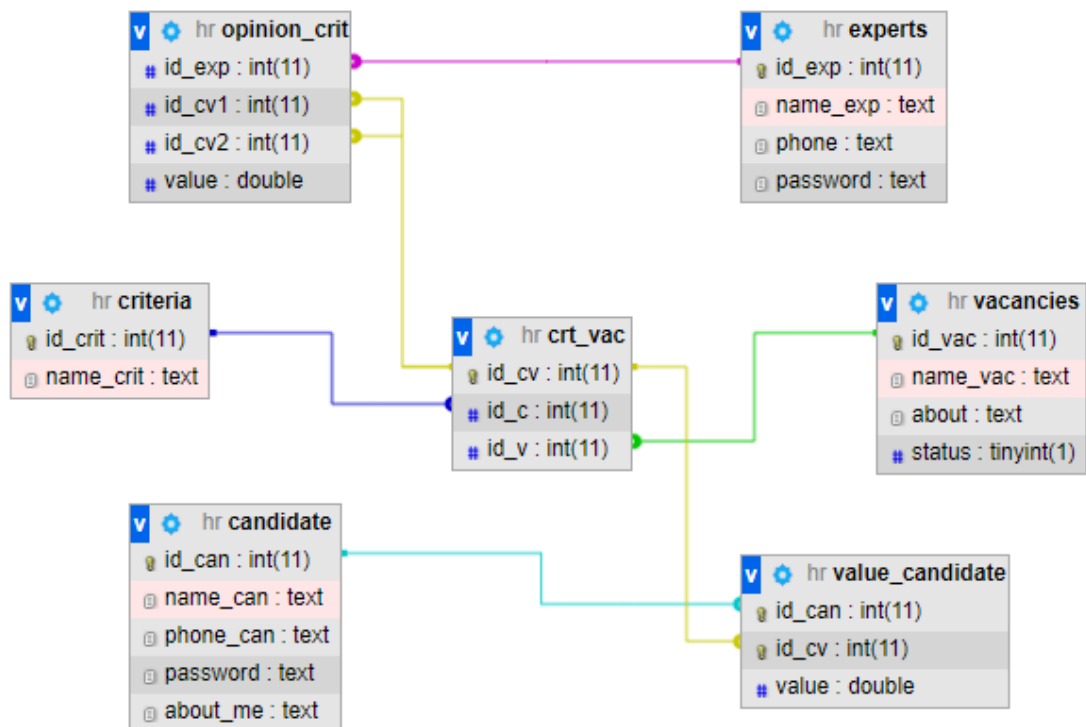


Рисунок 2.7 – Схема даталогічної моделі бази даних ІАС рекрутера

Оскільки усі атрибути атомарні, відношення знаходяться у 1НФ, проаналізувавши функціональні відношення, усі відношення знаходяться у 2НФ, тому що вони знаходять у 1НФ і усі відношення не мають часткових залежностей. Також прослідковується відсутність транзитивної залежності у відношеннях. Отже відношення знаходиться у 3НФ.

Наступним кроком необхідно прийняти рішення щодо ідентифікаторів таблиць і атрибутів, а також їхніх типів даних та інших характеристик. Рішення запишемо в таблицю (табл. 2.3).

Введемо скорочення в таблиці:

- Первинний ключ – ПК
- Зовнішній ключ – ЗК
- Складений первинний ключ – СПК

Таблиця №2.3 - Склад та характеристика атрибутів ДЛМ

Назва атрибуту	Ключ: ПК, ЗК, ЗК+СПК	Not NULL	Тип даних
Таблиця CRITERIA – «Критерій»			
id_crit	ПК	Так	Ціле число
name_crit		Так	Текст
Таблиця CANDIDATE – «Кандидат»			
id_can	ПК	Так	Ціле число
name_can		Так	Текст
phone_can		Так	Текст
password		Так	Текст
about		Ні	Текст
Таблиця EXPERTS – «Експерт»			
id_exp	ПК	Так	Ціле число
name_exp		Так	Текст
phone		Так	Текст
password		Так	Текст

Продовження таблиці №2.3

Таблиця VACANCIES – «Ваканція»			
id_vac	ПК	Так	Ціле число
name_vac		Так	Текст
about		Ні	Текст
status		Так	Бінарне значення
Таблиця CRT_VAC – «Критерій - Вакансія»			
id_cv	ПК	Так	Ціле число
id_c	ЗК	Так	Ціле число
id_v	ЗК	Так	Ціле число
Таблиця VALUE_CANDIDATE – «Оцінка кандидата за критерієм»			
id_cv	ЗК+СПК	Так	Ціле число
d_can	ЗК+СПК	Так	Ціле число
value		Так	Число з плаваючою комою
Таблиця OPINION_CRIT – «Попарні порівняння критеріїв»			
id_exp	ЗК+СПК	Так	Ціле число
id_cv1	ЗК+СПК	Так	Текст
id_cv2	ЗК+СПК	Так	Текст
value		Так	Число з плаваючою комою

2.3.3 Фізична модель бази даних

Провівши аналіз потреб нашої бази даних, для реалізації фізичної моделі було обрано СУБД SQLite, оскільки вона не вимагає додаткових налаштувань та чудово працює на будь-якій платформі.

Структура фізичної бази даних та вміст таблиць цієї БД наведені на рисунках 2.8 – 2.15.

Ім'я	Тип
Таблиці (8)	
candidate	
id_can	INTEGER
name_can	TEXT
phone_can	TEXT
password	TEXT
about_me	TEXT
criteria	
id_crit	INTEGER
name_crit	TEXT
crt_vac	
id_cv	INTEGER
id_c	INTEGER
id_v	INTEGER
experts	
id_exp	INTEGER
name_exp	TEXT
phone	TEXT
password	TEXT
opinion_crit	
id_exp	INTEGER
id_cv1	INTEGER
id_cv2	INTEGER
value	REAL
sqlite_sequence	
vacancies	
id_vac	INTEGER
name_vac	TEXT
about	TEXT
status	NUMERIC
value_candidate	
id_can	INTEGER
id_cv	INTEGER
value	REAL

Рисунок 2.8 – Структура бази даних ІАС рекрутера

Таблиця: candidate

	id_can ▼ ¹	name_can	phone_can	password	about_me
	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	1	Микола Панасюк	+380689992535	2535	NULL
2	2	Василь Баранчук	+380960000000	0000	NULL
3	3	Артур Трофімук	+380631111111	1111	NULL
4	4	Оксана Зарічна	+380961992027	2027	NULL

Рисунок 2.9 – Вміст таблиці «Кандидат»

Таблиця: criteria

	id_crit	name_crit
	Фільтр	Фільтр
1	1	Стаж водіння
2	2	Тривалість володіння водійським посвідченням
3	3	К-ть відзнак

Рисунок 2.10 – Вміст таблиці «Критерій»

Таблиця: crt_vac

	id_cv	id_c	id_v
	Філ...	Філ...	Філ...
1	1	1	1
2	2	2	1
3	3	1	2
4	4	2	2
5	5	3	1

Рисунок 2.11 – Вміст таблиці «Критерій - Вакансія»

Таблиця: experts

	id_exp	name_exp	phone	password
	Фільтр	Фільтр	Фільтр	Фільтр
1	1	Панасюк Б.І.	+380868363325	3325
2	2	Доманецька І. М.	+380403873425	3425

Рисунок 2.12 – Вміст таблиці «Експерт»

Таблиця: opinion_crit

	id_exp	id_cv1	id_cv2	value
	Фільтр	Фільтр	Фільтр	Філ...
1	1	1	2	4.0
2	1	3	4	0.33
3	1	1	5	6.0
4	1	2	5	0.25

Рисунок 2.13 – Вміст таблиці «Попарні порівняння критеріїв»

Таблиця: vacancies

	id_vac ▼ ¹	name_vac	about	status
	Фільтр	Фільтр	Фільтр	Фільтр
1	1	Інструктор	NULL	0
2	2	Викладач	NULL	0
3	3	Бухгалтер	Бухгалтерськ...	0

Рисунок 2.14 – Вміст таблиці «Вакансії»

	id_can	d_cv ▼	value
	Фільтр	Філь...	Філь...
1	1	1	3.0
2	4	1	10.0
3	2	1	2.0
4	1	2	2.0
5	4	2	15.0
6	2	2	1.0
7	1	3	3.0
8	4	3	0.0
9	2	3	0.0

Рисунок 2.15 – Вміст таблиці «Оцінка кандидата за критерієм»

2.4 Розробка інтерфейсних рішень

Опираючись на дерево функцій (рис.2.3), були розроблені ескізи головних екранів для усіх типів користувачів, а саме для кандидата, експерта, рекрутера.

В першу чергу спроектоване вікно входу, яке, при введенні номеру телефону, паролю та виборі типу користувача, відправляє у відповідну систему (рис. 2.16). Якщо залогінився звичайний користувач – перенаправлення відбудеться у підсистему обробки анкет, якщо експерт – у підсистему експертної оцінки критеріїв, якщо рекрутер – підсистему формування моделі оцінювання.

Рисунок 2.16 – Екран входу в систему

Розроблене вікно для подачі кандидатом анкети на вакансію (рис. 2.17). З цього вікна кандидат може обрати бажану вакансію, подати заявку, відредагувати заявку та видалити заявку, якщо така уже створена.

The screenshot shows a web browser window titled "HR-система" with the URL "http://hr.com/register". The page has a navigation bar with "Відкриті вакансії" and "Заповнення анкети" links, and a "Вихід" button. The main content area contains a form for registration. On the left, there is a "Вакансія:" dropdown menu with "Select" and a "Опис вакансії" text area. On the right, there are input fields for "ПІБ:", "Телефон", "Коротко про себе:", and a series of "Критерій 1" through "Критерій n" fields. At the bottom right, there are two buttons: "Подати анкету" (green) and "Видалити" (red).

Рисунок 2.17 – Головний екран підсистеми обробки анкет кандидатів.

Заповнення анкетної форми.

Розроблено вікно для заповнення МПП і створення шаблону моделі оцінки кандидатів на вакансію (рис. 2.18). З цього ж вікна можна перейти для додавання, редагування та видалення вакансій та критеріїв до них.

The screenshot shows a web browser window titled "HR-система" with the URL "http://hr.com/expert". The page has a navigation bar with a "Вихід" button. The main content area contains a form for expert evaluation. On the left, there is a "Назва вакансії" section with a "Вакансії" dropdown menu, a "Пул критеріїв" text area, and a "Створення шаблону" button. On the right, there is a table with columns for "Критерій1" through "Критерій5" and rows for "Критерій1" through "Критерій5". The table contains the following data:

	Критерій1	Критерій2	Критерій3	Критерій4	Критерій5
Критерій1	1				
Критерій2		1			
Критерій3			1		
Критерій4				1	
Критерій5					1

At the bottom right, there are two buttons: "Зберегти шаблон" (green) and "Видалити" (red).

Рисунок 2.18 – Головний екран підсистеми експертної оцінки критеріїв.

Заповнення МПП.

Розроблено вікно для визначення найкращого кандидата на посаду (рис.2.18). У вікні виводиться список претендентів, оціночні вектори та матриці, які використовуються алгоритмом МАІ, а також результати роботи алгоритму у текстовому та графічному видах.

The screenshot displays an HR system interface with the following components:

- Header:** "HR-система" and a browser address bar showing "http://hr.com/hr".
- Navigation:** A "Вихід" (Exit) button in the top right corner.
- Form:** "Виберіть вакансію:" (Select a vacancy) with a dropdown menu.
- Candidate List:** A table with columns: Ім'я (Name), Телефон (Phone), and Про кандидата (About candidate).

Ім'я	Телефон	Про кандидата
Ada Lovelace	December 10, 1815	Резюме
Grace Hopper	December 9, 1906	Резюме
Margaret Hamilton	August 17, 1936	Резюме
Joan Clarke	June 24, 1917	Резюме
- Action:** A green button labeled "Підібрати кандидата" (Select candidate).
- Priority Vectors:**
 - Вектор пріоритетів критеріїв (Criteria Priority Vector):**

Критерій	Важкість
Критерій 1	0.333
Критерій 2	0.333
Критерій 3	0.333
 - Матриця пріоритетів кандидатів (Candidate Priority Matrix):**

	За критерієм 1	За критерієм 2	За критерієм 3
Ada Lovelace	0.33	0.33	0.33
Grace Hopper			
Margaret Hamilton			
Joan Clarke			
- Global Priority Vector Matrix:**

	Ada Lovelace	Grace Hopper	Margaret Hamilton	Joan Clarke
	0.33	0.33	0.33	0.33
- Graphical Representation of Results:** A hierarchical decision tree diagram.
 - Goal:** Choose a Leader (1.000)
 - Criteria:** Experience (.647), Education (.127), Charisma (.270), Age (.056)
 - Alternatives:** Tom (.358), Dick (.493), Harry (.149)
- Recommendation:** "На посаду рекомендується призначити 'Ім'я обраного алгоритмом кандидата'" (It is recommended to appoint the name of the candidate selected by the algorithm).
- Action:** A blue button labeled "Сформувати запит" (Formulate request).

Рисунок 2.19 – Екран підсистеми рекрутера. Вибір кращого кандидата

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАСТОСУНКУ ІАС РЕКРУТИНГУ

3.1 Опис програмного забезпечення використаного при розробці ІАС

Для розробки програмного застосунку було прийнято рішення використати наступні мовні та програмні засоби (рис.3.1):

- Клієнтська частина:
 - HTML5
 - JavaScript
 - CSS
 - Bootstrap
- Серверна частина:
 - Flask – Python framework
 - СУБД SQLite
- Реалізація модулю MAI
 - Мова програмування Python

Розробка відбувалася у інтегрованому середовищі розробки PyCharm.

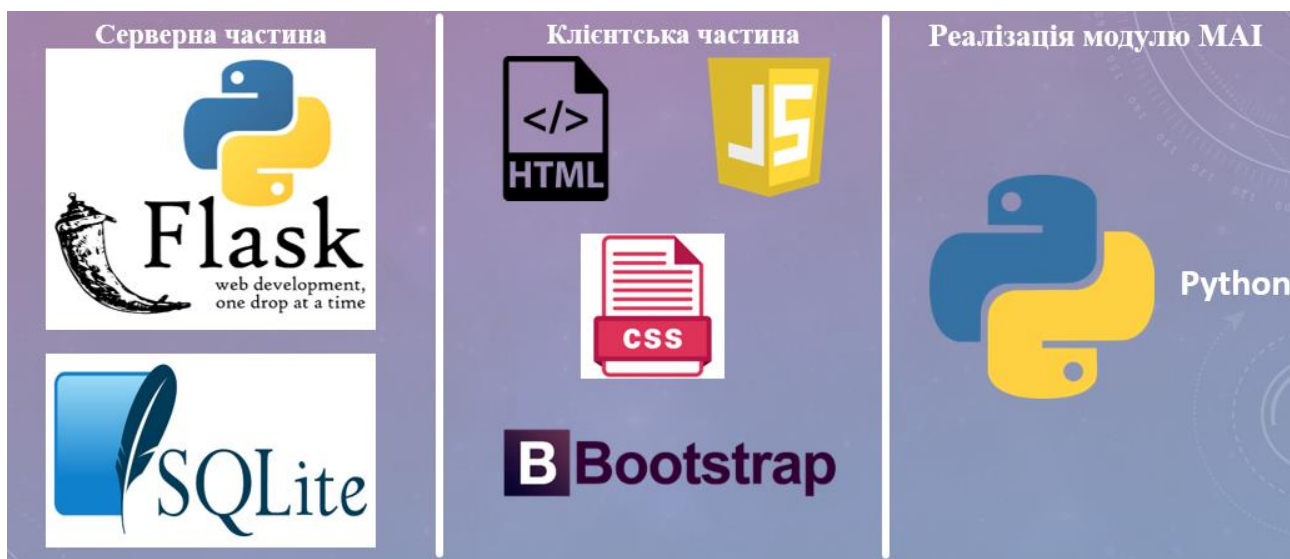


Рисунок 3.1 – Використане програмне забезпечення

PyCharm – це потужне середовище розробки для мови програмування Python, яке надає інструменти для аналізу коду, засоби для проведення юніт-тестів та дозволяє з легкістю встановлювати додаткові бібліотеки у Python-застосунок.

Можливості PyCharm:

- Аналіз коду;
- Керування файловою структурою проекту;
- Рефакторинг коду;
- Підтримка систем контролю версій;
- Вбудовані інструменти для WEB-розробки;
- Вбудований відладжувач. [19]

Python - інтерпретована високорівнева об'єктно-орієнтована мова програмування із суворою динамічною типізацією. За рахунок того, що цій мові не потрібен компілятор, виконання коду пришвидшується, що робить Python більш привабливим на фоні аналогів.[20] Також розробники люблять Python за широкий вибір бібліотек, які допоможуть вирішити задачу будь-якої складності.

HTML – це стандартизована мова розмітки, яка дозволяє браузеру інтерпретувати себе в інтерфейс на веб-сторінці. На сьогодні уявити веб-простір без HTML не можливо. [21]

CSS - (Cascading Style Sheets, в перекладі – каскадні таблиці стилів) - це спеціальна мова, за допомогою якої задають параметри того, як і де відобразити елементи веб-сторінки. Найчастіше CSS використовується поруч з такими мовами розмітки, як HTML, XHTML та XML. [22]

JavaScript – це мова програмування, що робить Web-сторінку інтерактивною. Скрипт, написаний цією мовою, виконується інтерпретатором, вбудованим в звичайний Web-браузер. Для виконання програми не потрібно навіть перезавантажувати Web-сторінку, бо у відповідь на будь-яку подію виконуються всі програми. [23]

Flask - мікрофреймворк для вебдодатків, створений з використанням Python. В його основу входять такі інструменти як Werkzeug та Jinja2. Flask

вважається мікрофреймворком, бо він не вимагає спеціальних засобів чи бібліотек. [24].

Властивості Flask:

- Вміщує в собі сервер для розробки та відлагоджувач;
- Підтримує юніт-тести;
- Підтримує безпечні cookie;
- Підтримує Unicode;
- Докладна документація

SQLite – полегшена реляційна система керування базами даних. Перевага саме даної СКБД полягає у її сумісності з великою кількістю інших СКБД, що робить застосунки, які написані на її основі, легкомасштабованими та кросплатформними.[25]

Обраний набір програмних засобів та технологій забезпечує усі потреби застосунку та має наступні переваги:

- широка сфера застосування, популярність, та велика спільнота розробників, які використовують інструменти з даного набору. Завдяки цьому немає проблем з пошуку спеціалістів для забезпечення підтримки програмного засобу.
- застосунок, створений за допомогою цих засобів, є web-застосунком, що в свою чергу відкидає потребу у інсталяції спеціалізованих додатків на робочу машину користувачів. Робота у програмі може відбуватися у будь-якому браузері, який підтримує JavaScript.
- виконання back-end частини застосунку відбуватиметься на сервері, а не на машині. За рахунок цього застосунок не прив'язаний до якоїсь конкретної операційної системи, і є кросплатформним.

3.2 Структура програмного забезпечення ІАС рекрутингу

Розробка інструментально-аналітичної системи рекрутингу відбувалася під процедурною парадигмою програмування. Процедурне програмування –

стиль програмування, в якому задача вирішується алгоритмом, частини якого огорнуті у функції (процедури). Перевагою такого підходу є можливість виклику будь-якої процедури з будь-якої точки програми, що дозволяє використати один і той самий код в різних місцях без дублювання цього коду. Такий код легший для розуміння порівняно з об'єктно-орієнтованим програмуванням, а також підтримує модульність та структурність. [26, с.110]

У результаті розробки, програма налічує в собі 5 модулів, кожен з яких містить по декілька процедур (рис. 3.2):

1. Модуль авторизації:
 - registr – процедура, яка забезпечує реєстрацію кандидата у системі;
 - login – процедура, яка дозволяє різним типам користувачів авторизуватися в системі;
 - logout – процедура, яка надає можливість користувачу деавторизуватися з системі.
2. Модуль робочого місця кандидата:
 - cndt – процедура, яка виводить на екран вакансії, на які кандидат надіслав резюме;
 - vacansy – процедура, яка надає можливість кандидату надіслати резюме або відредагувати уже існуюче;
 - open_vacansy – процедура, яка виводить на екран вакансії, на які наразі проводиться набір персоналу;
 - cand_vac_del – процедура, яка дозволяє кандидату видалити своє резюме з системи.
3. Модуль робочого місця експерта:
 - expert_vacancies – процедура, яка виводить на екран усі вакансії, які містяться в системі;
 - mpr – процедура, яка дозволяє заповнити матрицю попарних порівнянь критеріїв вакансії;

- mpp_del – процедура, яка видаляє матрицю попарних порівнянь критеріїв вакансії;
 - new_vacancy – процедура, що додає нову вакансію в систему;
 - new_criteria – процедура, що додає нові критерії в критеріальну базу системи;
 - del_criteria – процедура, що видаляє обрані критерії з критеріальної бази системи;
 - open_close_vac – процедура, яка відкриває або закриває набір персоналу на вакансію;
 - delete_vac – процедура, яка видаляє вакансію та її дані з системи.
4. Модуль робочого місця рекрутера:
- hr_vac – процедура, яка виводить на екран усі вакансії, які містяться в системі;
 - hr – процедура, що приводить в дію модуль методу аналізу ієрархій, та виводить на екран результати роботи даного методу, а також дозволяє зберегти результати у PDF файл.
5. Модуль методу аналізу ієрархій (MAI):
- Saati – процедура, яка зводить анкетні дані кандидатів до шкали Сааті;
 - crit – процедура, що дістає з БД МПП критеріїв по обраній вакансії;
 - vector – процедура, що будує вектор пріоритетів з МПП критеріїв або кандидатів;
 - find_sand – процедура, яка дістає з БД інформацію про кандидатів на обрану вакансію;
 - MAI – процедура, яка приводить в дію алгоритм методу аналізу ієрархій, використовуючи вище наведені процедури.

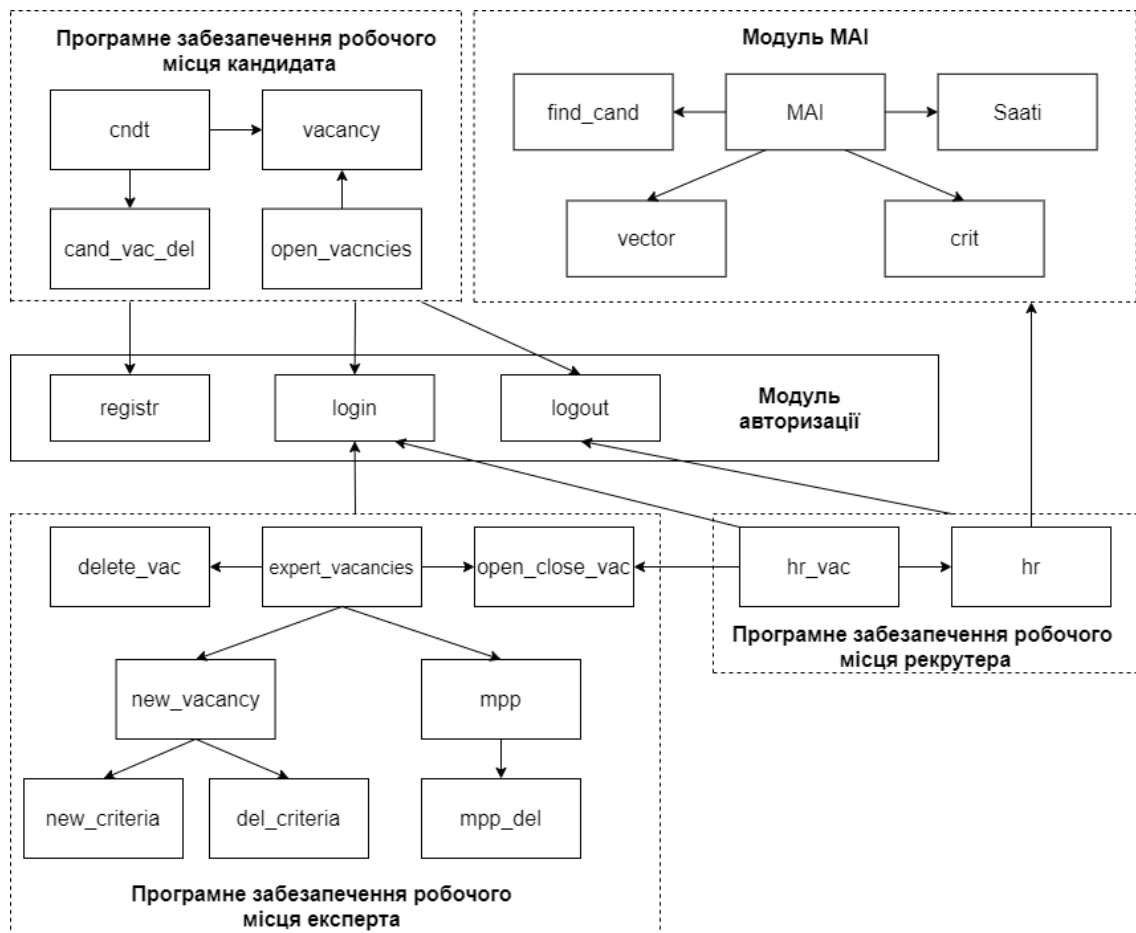


Рисунок 3.2 – Структурна схема розробленого програмного забезпечення ІАС рекрутингу

3.3 Тестування коректності роботи застосунку

Тестування програмного забезпечення – це процес у якому програмний застосунок піддається дослідженню, пошуку помилок або визначення рівня якості.

Тестувальник, під час своєї роботи проводить оцінювання таких аспектів:

- На скільки ПЗ відповідає функціональним і нефункціональним вимогам і чи відповідає ним взагалі;
- Коректність вихідних даних на різних наборах вхідних даних;
- Прийнятність терміну виконання функцій ПЗ;
- Сумісність з обладнанням, операційними системами та іншим програмним забезпеченням;

- Перевірка досяжності результату, якого очікують зацікавлені сторони. [27]

Дослідження коректності роботи елементів ІАС рекрутера описано в таблицях 3.1 – 3.3 та на рисунках 3.3 – 3.21.

Таблиця 3.1 – Перевірка коректності розрахунків МАІ

Мета тесту	Перевірити коректність розрахунків МАІ реалізованого в ІАС рекрутера
Початковий стан	Відкрита сторінка зі списком вакансій у підсистемі рекрутера
Вхідні дані	Ідентифікатор вакансії
Схема проведення тесту	Натискаємо на кнопку «Підібрати працівника» для вакансії «Інструктор»
Очікуваний результат	Результати збігаються з ручними розрахунками у MS Excel
Фактичний результат	Результати збігаються з ручними розрахунками у MS Excel

Критерій	Стаж водіння	Тривалість володіння посвідченням	К-ть відзнак	Ср.Геом	W	Критерій	Стаж водіння	Тривалість володіння посвідченням	К-ть відзнак	Ср.Геом	W
Микола Панасюк	1	6	4	2,8844991	0,69066581	Микола Панасюк	1	2	0,22	0,76059049	0,16578405
Василь Баранчук	0,16666667	1	0,33	0,3802952	0,09105807	Василь Баранчук	0,5	1	0,11111	0,38157014	0,08316991
Оксана Зарічна	0,25	3,03030303	1	0,9116092	0,21827612	Оксана Зарічна	4,54545455	9,00009	1	3,44567827	0,75104605
Сума:					4,1764036	Сума:					4,5878389
Микола Панасюк	1	1,5714	0,1746	0,64979594	0,13580002	Микола Панасюк	1	1,5714	0,1746	0,64979594	0,13580002
Василь Баранчук	0,63637521	1	0,11111	0,41351265	0,08641948	Василь Баранчук	0,63637521	1	0,11111	0,41351265	0,08641948
Оксана Зарічна	5,72737686	9,00009	1	3,72163866	0,7777805	Оксана Зарічна	5,72737686	9,00009	1	3,72163866	0,7777805
Сума:					4,78494725	Сума:					4,78494725
Микола Панасюк	1	9	9	4,32674871	0,81818182	Микола Панасюк	1	9	9	4,32674871	0,81818182
Василь Баранчук	0,11111111	1	1	0,48074986	0,09090909	Василь Баранчук	0,11111111	1	1	0,48074986	0,09090909
Оксана Зарічна	0,11111111	1	1	0,48074986	0,09090909	Оксана Зарічна	0,11111111	1	1	0,48074986	0,09090909
Сума:					5,28824842	Сума:					5,28824842

Рисунок 3.3 – Ручні розрахунки МАІ у MS Excel

Матриця пріоритетів критеріїв

Стаж водіння	0.691
Тривалість володіння водійським посвідченням	0.091
К-ть відзнак	0.218

Матриця пріоритетів кандидатів

	Стаж водіння	Тривалість володіння водійським посвідченням	К-ть відзнак
Микола Панасюк	0.167	0.136	0.818
Василь Баранчук	0.083	0.086	0.091
Оксана Зарічна	0.75	0.778	0.091

Вектор глобальних пріоритетів

Микола Панасюк	Василь Баранчук	Оксана Зарічна
0.306	0.085	0.609

На посаду рекомендується призначити кандидата: #4 - Оксана Зарічна

Рисунок 3.4 – Результат роботи МАІ у ІАС рекрутера

Результати ручних обчислень (рис 3.3) і результати обчислень в ІАС рекрутера (рис 3.4) однакові. Отже, тест пройдено успішно.

Таблиця 3.2 – Перевірка коректності розрахунків МАІ

Мета тесту	Перевірити захист від несанкціонованого доступу
Початковий стан	Відкрита сторінка авторизації
Вхідні дані	Номер телефону та пароль кандидата
Схема проведення тесту	Вводимо дані кандидата та намагаємося авторизуватися як експерт
Очікуваний результат	Невдалий вхід. Система запропонує звернутися до адміністратора.
Фактичний результат	Невдалий вхід. Система запропонувала звернутися до адміністратора.



Рисунок 3.5 – Спроба авторизації клієнта в підсистемі експерта

Система не впустила користувача у систему експерта з авторизаційними даними кандидата та було виведено інформаційне вікно. Отже, тест пройдено успішно.

Таблиця 3.3 – Перевірка коректності розрахунків МАІ

Мета тесту	Перевірити адаптивність інтерфейсу застосунку
Початковий стан	Система відкрита на телефоні
Вхідні дані	-
Схема проведення тесту	Заходимо в систему з мобільного пристрою
Очікуваний результат	Елементи на сторінках відображаються коректно
Фактичний результат	Елементи на сторінках відображаються коректно



Рисунок 3.6 – Вигляд сторінок сайту на мобільному пристрої

Елементи веб-сторінки відображаються на мобільному пристрої коректно, гармонійно та читабельно. Отже, тест пройдено успішно.

3.4 Опис роботи застосунку

Як уже було згадано раніше, ІАС рекрутера поділяється на 3 підсистеми:

- Підсистема кандидата;
- Підсистема експерта;
- Підсистема рекрутера.

Кожна з цих систем має свої функції і варта окремого розгляду.

3.4.1 Опис роботи підсистеми кандидата

У підсистемі кандидата ІАС рекрутера передбачені наступні функції:

- Перегляд поданих заявок на працевлаштування
Для цього необхідно зайти на сторінку «Мої заявки» (рис.3.7).

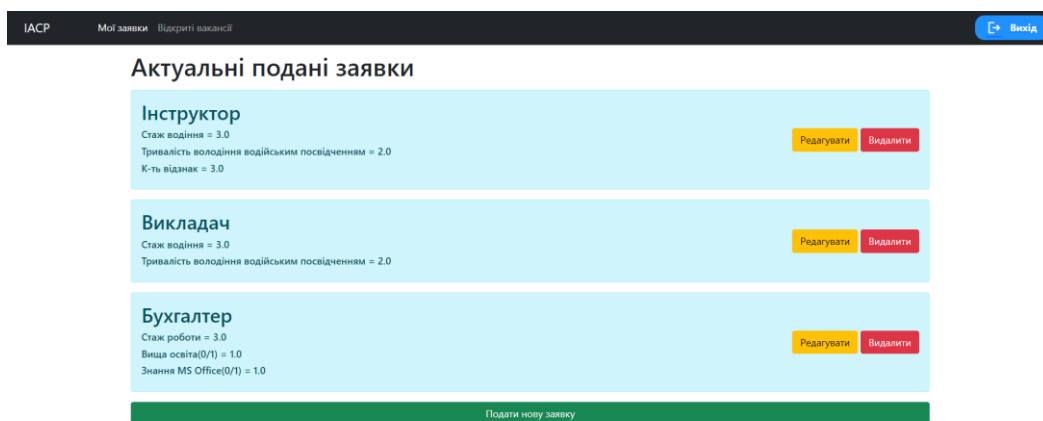


Рисунок 3.7 – Сторінка «Мої заявка» підсистеми кандидата

- Перегляд відкритих на даний момент заявок
Для цього необхідно зайти на сторінку «Відкриті вакансії» (рис.3.8).

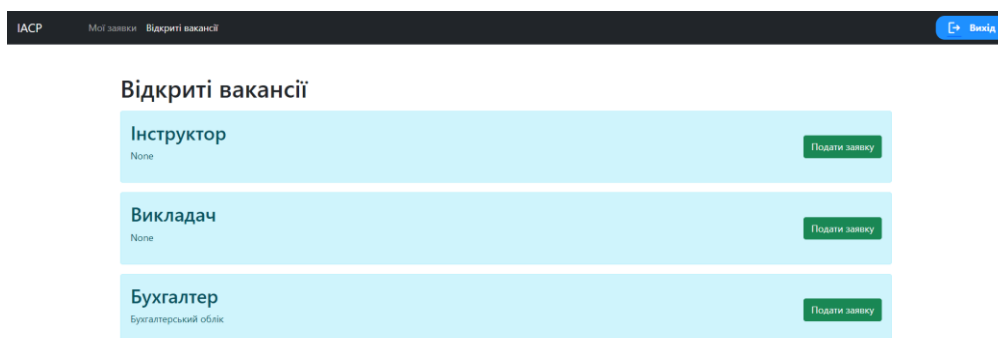


Рисунок 3.8 – Сторінка «Відкриті вакансії» підсистеми кандидата

- Функція надсилання нової заявки на вакансію або її редагування
Для її застосування необхідно пройти за шляхом «Мої заявки» - «Подати нову заявку» - ввести свої анкетні дані – «Подати заявку»; або за

шляхом «Відкриті вакансії» - «Подати заявку» - ввести свої анкетні дані – «Подати заявку». Для редагування можна скористатися шляхом «Мої заявки» - «Редагувати» - ввести свої анкетні дані – «Подати заявку» (рис.3.9).

Рисунок 3.9 – Сторінка оформлення заявки на вакансію у підсистемі кандидата

- Видалення надісланої заявки

Щоб видалити заявку потрібно пройти за шляхом «Мої заявки» - натиснути «Видалити» поруч з необхідною заявкою – «Ок» (рис.3.10).

Рисунок 3.10 – Видалення надісланої заявки

3.4.2 Опис роботи підсистеми експерта

У підсистемі експерта ІАС рекрутера передбачені наступні функції:

- Перегляд вакансій, які додані в систему

Для цього необхідно зайти на сторінку «Вакансії» (рис.3.11).

Рисунок 3.11 – Сторінка «Вакансії» підсистеми експерта

- Відкриття/закриття набору на вакансію

Щоб відкрити або закрити набір персоналу на вакансію потрібно перейти за шляхом «Вакансії» – натиснути «Закрити набір»/«Відкрити набір» поруч з необхідною вакансією – «Ок» (рис.3.12).

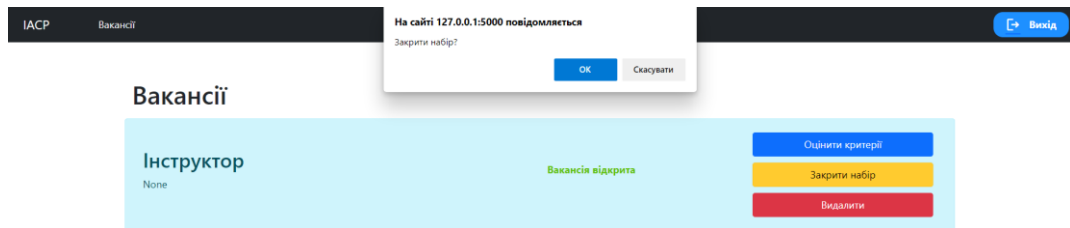


Рисунок 3.12 - Процес закриття вакансії у підсистемі експерта

- Додавання нової вакансії

Для додавання нової вакансії застосування необхідно пройти за шляхом «Вакансії» - «Нова вакансія» - ввести назву вакансії та її опис – обрати критерії – «Створити» (рис.3.13).

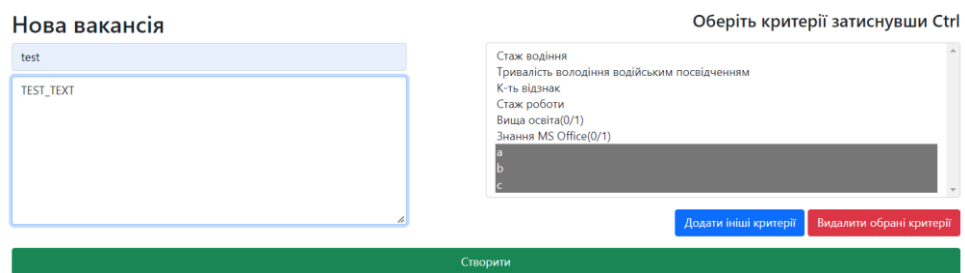


Рисунок 3.13 – Створення нової вакансії

- Редагування критеріальної бази

Щоб додати до бази нові критерії, необхідно на сторінці створення нової вакансії натиснути кнопку «Додати інші критерії» і у вікні, яке відкрилося, ввести через кому нові критерії та підтвердити введення клавішею «Зберегти» (рис.3.14).

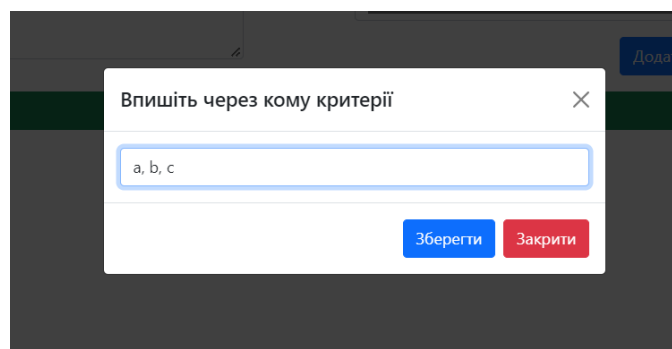


Рисунок 3.14 – Додавання критеріїв

Щоб видалити критерії з критеріальної бази необхідно обрати потрібні критерії у вікні та натиснути «Видалити обрані критерії» (рис.3.15).

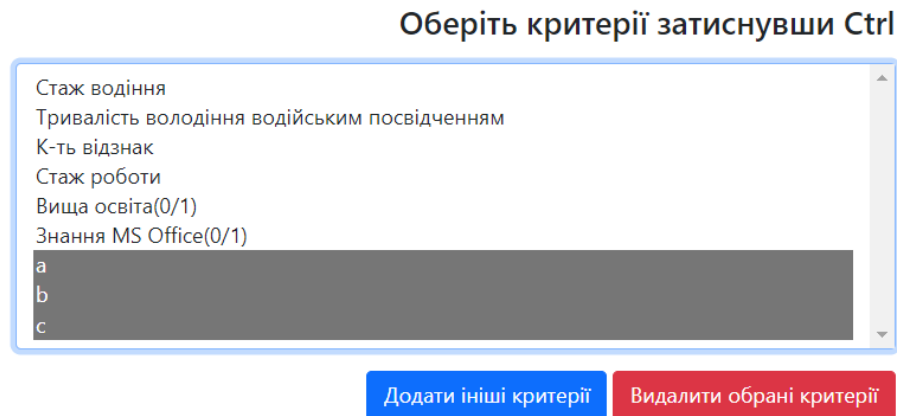


Рисунок 3.15 – Видалення критеріїв

- Створення шаблонів оцінки для вакансії (заповнення МПП)

Шаблон, за яким в подальшому буде проводитися ранжування кандидатів можна заповнити за шляхом «Вакансії» – «Оцінити критерії» – ввести значення, визначені експертом, у верхню трикутну матрицю – «Зберегти» (рис.3.16).

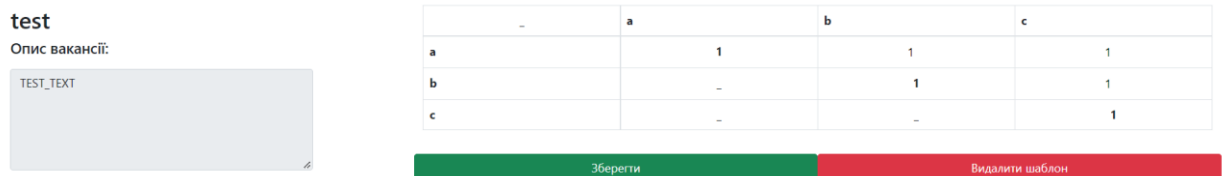


Рисунок 3.16 – Заповнення МПП

- Видалення вакансії

Щоб видалити вакансію потрібно пройти за шляхом «Вакансії» - натиснути «Видалити» поруч з необхідною Вакансією – «Ок» (рис.3.17).

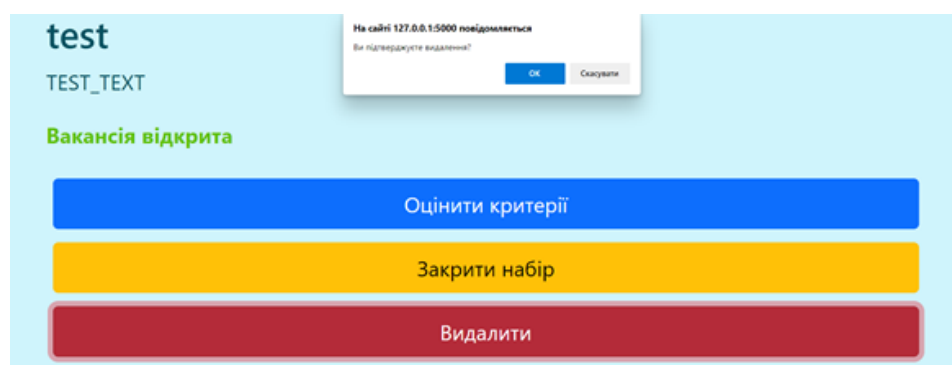


Рисунок 3.17 – Видалення вакансії експертом

3.4.3 Опис роботи підсистеми рекрутера

У підсистемі рекрутера ІАС рекрутера передбачені наступні функції:

- Перегляд вакансій, які додані в систему
Для цього необхідно зайти на сторінку «Усі вакансії» (рис.3.18).

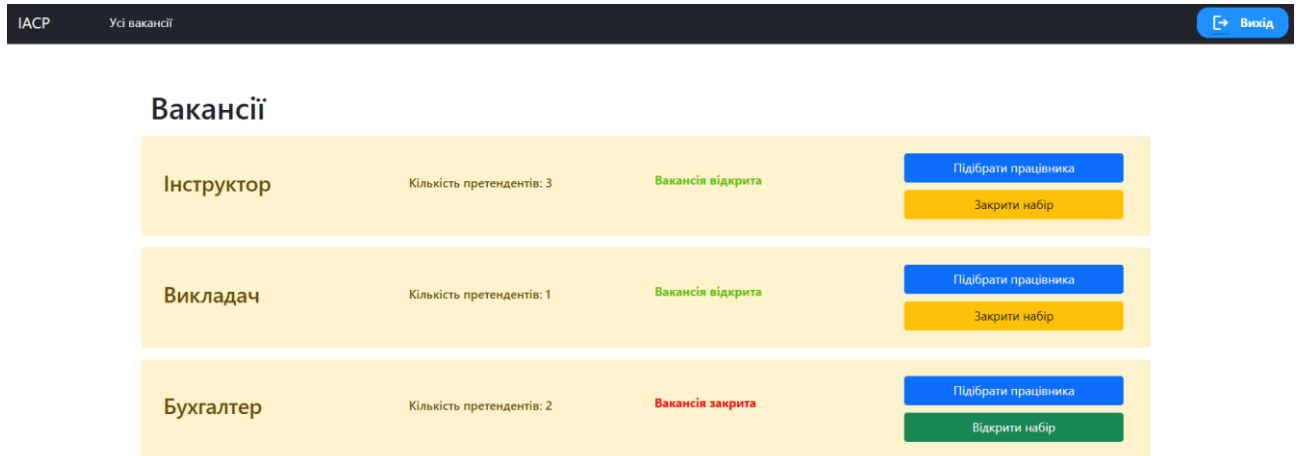


Рисунок 3.18 – Відображення списку вакансій у системі рекрутера

- Відкриття/закриття набору на вакансію
Щоб відкрити або закрити набір персоналу на вакансію потрібно пройти за шляхом «Усі вакансії» - натиснути «Закрити набір»/«Відкрити набір» поруч з необхідною вакансією – «Ок» (рис.3.19).

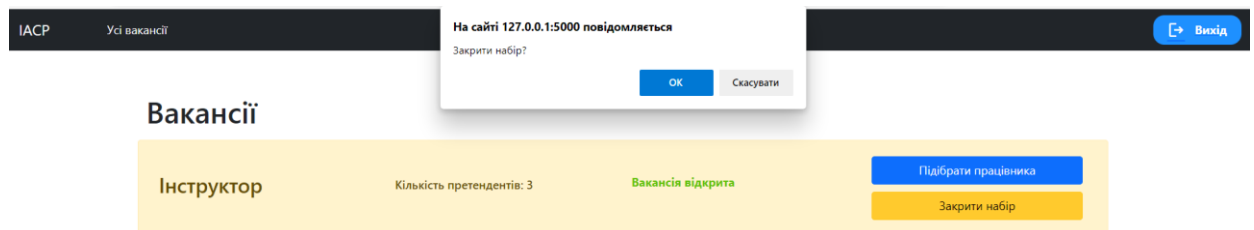


Рисунок 3.19 – Процес закриття вакансії у підсистемі рекрутера

- Відбір персоналу за допомогою МАІ та виведення результатів на екран
Щоб підібрати найкращого кандидата на посаду, на сторінці «Усі вакансії» натисніть «Підібрати працівника» поруч з бажаною вакансією.

Після чого відкриється сторінка з розрахунками методу та рекомендацією що до найкращого кандидата (рис.3.20).

ІАСР Усі вакансії Вхід

Інструктор

Ім'я	Телефон	Про кандидата
Микола Панасюк	+380689992535	Резюме
Василь Баранчук	+380960000000	Резюме
Оксана Зарічна	+380961992027	Резюме

Матриця пріоритетів критеріїв

Стаж водія	0.718
Тривалість володіння водійським посвідченням	0.106
К-ть відзнак	0.177

Матриця пріоритетів кандидатів

	Стаж водія	Тривалість володіння водійським посвідченням	К-ть відзнак
Микола Панасюк	0.167	0.136	0.818
Василь Баранчук	0.083	0.086	0.091
Оксана Зарічна	0.75	0.778	0.091

Вектор глобальних пріоритетів

Микола Панасюк	Василь Баранчук	Оксана Зарічна
0.279	0.085	0.637

На посаду рекомендується призначити кандидата: #4 - Оксана Зарічна

Сформувати звіт

Рисунок 3.20 – Сторінка з результатами роботи алгоритму

- Вивантаження результатів в файл
Передбачена можливість вивантаження результатів у файл доступна на сторінці результатів за кнопкою «Сформувати звіт». Результат роботи даної функції наведено на рисунку 3.21



Рисунок 3.21 – PDF документ, у який було вивантажено результати роботи алгоритму

ВИСНОВКИ

У підсумку даної кваліфікаційної роботи бакалавра було проаналізовано предметну область рекрутингу, визначено її основні аспекти та поставлено функціональні та нефункціональні вимоги до майбутнього застосунку. На основі отриманих даних була спроектована й успішно реалізована ІАС (інструментально-аналітична система) рекрутингу, яка вирішує питання спрощення відбору персоналу на ту чи іншу посаду.

Перед розробкою були досліджені та проаналізовані технології та інструменти, які допомогли вирішити задачу рекрутингу та реалізувати програмний засіб. Вибір впав на розробку web-застосунку, серверна частина якого побудована на фреймворкові Flask та мові програмування Python, а у якості СУБД було використано SQLite. Даний набір інструментів був обраний за їх кросплатформність. Адаптивність web-застосунку забезпечена за допомогою таблиць стилів Bootstrap.

Усі, поставлені на початку розробки, вимоги були виконані та протестовані. Після контрольного тестування різних модулів системи та зв'язків між цими модулями, проблем виявлено не було – результати тестування збігаються з очікуваними.

Дане програмне забезпечення буде в нагоді рекрутерам, зекономить їх час, час компанії та забезпечить прозору, чесну конкуренцію між кандидатами, виключивши людський фактор при порівнянні кандидатів між собою. Отже, проведена робота над даною ІАС є раціональною та виправданою.

Список використаних джерел

1. Свінцицька О.М., Богоявленська Ю.В., Ліханова В.О. Сучасні технології в управлінні розвитком персоналу та їх вплив на ефективність бізнесу. Інфраструктура ринку. 2017. Вип. 14. С. 183–188. URL : http://www.market-infr.od.ua/journals/2017/14_2017_ukr/34.pdf.
2. Dr. Jon Krohn. The Evolution Of The Recruitment Industry. URL: <https://www.gqrgm.com/part-1-the-evolution-of-the-recruitment-industry/>
3. Лисак В.Ю., Семендяк В.М. Суть та значення рекрутингу в системі управління персоналом підприємства. Приазовський економічний вісник. 2019. Вип. 2(13). С. 130–134. URL : http://pev.kpu.zp.ua/journals/2019/2_13_uk/23.pdf
4. Gherasymova V. O., Chyrva V. S. (2018) Innovacijni metody rekrutynghu v suchasnykh umovakh [Innovative recruiting methods in modern conditions]. Prychornomorsjki ekonomichni studiji [Black Sea Economic Studies], vol. 27, pp. 82–86.
5. *Maria Gold*. Empire Resume. External Recruitment. URL: <https://empireresume.com/internal-vs-external-recruiting/>
6. Підбір, відбір і найм персоналу — 5 ефективних методів набору співробітників у компанію + реальні приклади пошуку працівників. Путівник у світі фінансів і бізнесу, банківські і фінансові послуги. URL: <http://bigenergy.com.ua/fnansi/bznes-dlya-pdpri/1140-pidbir-vidbir-i-najm-personalu--5-metodiv-poshuku-spivrobotnikiv.html>.
7. Li R. Y. M., Chau K. W., Zeng F. F. Ranking of Risks for Existing and New Building Works. *Sustainability*. 2019. Vol. 11, no. 10. P. 2863. URL: <https://doi.org/10.3390/su11102863> (date of access: 17.06.2022).
8. Saaty, Thomas L. (June 2008). Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors – The Analytic Hierarchy/Network

- Process. Review of the Royal Academy of Exact, Physical and Natural Sciences, Series A: Mathematics (RACSAM) 102 (2): 251–318.
9. Williamson O. E. Markets and Hierarchies : Analysis and Antitrust Implications. Free Press, 1983. 286 p.
 10. Метод аналітичних мереж - Система підтримки прийняття рішень ІТ-компанії "Тріумф ІТ". Система підтримки прийняття рішень ІТ-компанії "Тріумф ІТ". URL: <https://dss.tg.ck.ua/anp-help>.
 11. Критерії прийняття рішень в умовах повної невизначеності. *Pidru4niki*. URL: https://pidru4niki.com/15461213/ekonomika/kriteriyi_priynyattya_rishe_n_umovah_povnoyi_neviznachenosti.
 12. DecisionGrid Accounting & Bookkeeping. *DecisionGrid Accounting & Bookkeeping*. URL: <http://www.thedecisiongrid.com>.
 13. Welcome to Business Performance Management. *BPMSG – Business Performance Management Singapore*. URL: <http://bpmsg.com>.
 14. Decision Lens | Portfolio and Budget Planning. *Decision Lens | Portfolio and Budget Planning*. URL: <http://www.decisionlens.com>.
 15. Strategic Planning Software for Smarter Decisions. *Strategic Planning Software for Smarter Decisions*. URL: <http://expertchoice.com>.
 16. Super Decisions. *Super Decisions*. URL: <https://superdecisions.com>.
 17. Дворівнева клієнт-серверна архітектура. *ni.biz.ua*. URL: http://ni.biz.ua/7/7_9/7_95326_dvuhurovnevaya-klient-servernaya-arhitektura.html.
 18. Begg C., Connolly T. Database Systems: A Practical Approach to Design, Implementation, and Management (3rd Edition). Addison Wesley, 2001. 1236 p.
 19. PyCharm, потужна IDE для створення програм за допомогою Python. Linux Adictos. URL: <https://www.linuxadictos.com/uk/pycharm-un-potente-ide-para-crear-programas-con-python.html>.
 20. Rossum G. V. Python Reference Manual. Iuniverse Inc, 2000. 108 p.

21. А що таке HTML? - Українське програмування. *Українське програмування - Українське програмування*. URL: <http://programming.in.ua/web-design/html/73-html-introduction.html>.
22. Що таке CSS. *Український веб-довідник*. URL: https://css.in.ua/article/shcho-take-html_10.
23. Мова JavaScript та її можливості - Web технології та web дизайн. Google Sites: Sign-in. URL: <https://sites.google.com/site/webtehnologiietawebdizajn/mova-javascript-ta-ieie-mozlivosti>.
24. Foreword – flask documentation (0.10). Wayback Machine. URL: <https://web.archive.org/web/20151110013607/http://flask.pocoo.org/docs/0.10/foreword/#what-does-micro-mean#what-does-micro-mean>.
25. About SQLite. SQLite Home Page. URL: <https://www.sqlite.org/about.html>.
26. Althoff C. The self-taught programmer. 2017. 283 p.
27. Kaner C. Testing computer software. 2nd ed. London : International Thomson Computer Press, 1993. 480 p.

ДОДАТОК 1

Лістинг програми

```

from flask import Flask, render_template, request, redirect, flash
import sqlite3
import os
from main import MAI

currentdirectory = os.path.dirname(os.path.abspath(__file__))
app = Flask(__name__)
app.config['SECRET_KEY'] = 'dfgsgsa4vag'
id_user = 0
id_expert = 0
id_hr = 0

# Вхід в систему
@app.route('/', methods=['POST', 'GET'])
def index():
    global id_user
    global id_expert
    global id_hr
    if id_user != 0:
        return redirect('/open_vacancies')
    elif id_expert != 0:
        return redirect('/expert/vacancies')
    elif id_hr != 0:
        return redirect('/hr/hr_vac')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    if request.method == "POST":
        user = request.form['user']
        phone = str(request.form['phone'])

        password = str(request.form['password'])
        match user:
            case 'candidate':
                q1 = "SELECT * FROM candidate WHERE candidate.phone_can = ?;"
                person = cursor.execute(q1, (phone,)).fetchone()
                if person:
                    if password == person[3]:
                        id_user = person[0]
                        connection.close()
                        return redirect('/candidate')
                    else:
                        flash('Невірний пароль')
            else:
                connection.close()
                return redirect('/registr')
        case 'expert':
            q1 = "SELECT * FROM experts WHERE experts.phone = ?;"
            person = cursor.execute(q1, (phone,)).fetchone()
            if person:
                if password == person[3]:
                    id_expert = person[0]
                    connection.close()
                    return redirect('/expert/vacancies')
                else:
                    flash('Невірний пароль')
            else:
                flash('Зверніться до адміністратора, щоб додав вас у систему!')
        case 'hr':

```

```

        if request.form['phone'] == "admin" and request.form['password'] == "admin":
            connection.close()
            id_hr = 1
            return redirect('/hr/hr_vac')
        else:
            flash('Невірні дані')
    connection.close()
    return render_template("index.html")

# Реєстрація
@app.route('/registr', methods=['POST', 'GET'])
def registr():
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    if request.method == "POST":
        phone = str(request.form['phone'])
        user_name = request.form['user_name']
        about_me = request.form['about_me']
        password = str(request.form['password'])
        cpassword = str(request.form['confirm_password'])

        q1 = "SELECT * FROM candidate WHERE candidate.phone_can = ?;"
        person = cursor.execute(q1, (phone,)).fetchone()
        print(person)
        if person:
            flash("Користувач уже зареєстрований")
        else:
            if password != cpassword:
                flash("Паролі не збігаються!")
            else:
                sql = "INSERT INTO candidate (name_can, phone_can, password, about_me)
VALUES(?,?,?,?)"
                cursor.execute(sql, (user_name, phone, password, about_me))
                connection.commit()
                global id_user
                id_user = cursor.execute("SELECT id_can FROM candidate ORDER BY id_can DESC LIMIT
1").fetchone()[0]
                connection.close()
                return redirect('/open_vacancies')
    connection.close()
    return redirect("/")

# Виведення на екран вакансії на які кандидат надіслав резюме
@app.route('/candidate', methods=['POST', 'GET'])
def cndt():
    global id_user
    if id_user == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql = "SELECT id_vac, name_vac, v_c.id_cv, name_crit, value FROM (SELECT name_crit, id_cv,
vacancies.* FROM criteria INNER JOIN crt_vac ON criteria.id_crit = crt_vac.id_c INNER JOIN (
SELECT id_vac, name_vac FROM vacancies WHERE status = 0 )as vacancies ON crt_vac.id_v =
vacancies.id_vac ORDER BY id_vac, id_cv) as crt_vac INNER JOIN (SELECT id_cv, value FROM
value_candidate WHERE id_can = ?) as v_c ON crt_vac.id_cv = v_c.id_cv"
    my_anc1 = cursor.execute(sql, (id_user,)).fetchall()
    my_anc = [[]]
    if len(my_anc1) != 0:
        id_vac = my_anc1[0][0]
        for m in my_anc1:
            if m[0] != id_vac:
                id_vac = m[0]
                my_anc.append([])
            my_anc[len(my_anc) - 1].append(m)
        del my_anc1
    connection.close()

```

```

return render_template("cndt/candidate.html", my_anc=my_anc)

# Виведення відкритих вакансій на екран у підсистемі кандидата
@app.route('/open_vacancies', methods=['POST', 'GET'])
def open_vacancies():
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    open_vac = cursor.execute("SELECT * FROM vacancies WHERE status = FALSE").fetchall()

    connection.close()
    return render_template("cndt/open_vacancies.html", open_vac=open_vac)

# Посилання/редагування резюме кандидата по вакансії в БД
@app.route('/open_vacancies/vacancy=<int:id_vac>', methods=['POST', 'GET'])
def vacancy(id_vac):
    global id_user
    if id_user == 0:
        return redirect('/')
    print(id_user)
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql = "SELECT id_cv, name_crit FROM crt_vac INNER JOIN criteria on crt_vac.id_c =
criteria.id_crit WHERE crt_vac.id_v IN(SELECT id_vac FROM vacancies WHERE status = FALSE AND
id_vac = ?);"
    anc_crit = cursor.execute(sql, (id_vac,)).fetchall()

    sql = "SELECT vacancies.name_vac, vacancies.about FROM vacancies WHERE vacancies.id_vac = ?;"
    vac_info = cursor.execute(sql, (id_vac,)).fetchone()
    if request.method == "POST":

        for el in anc_crit:
            row = (request.form["crit" + str(el[0])], el[0], id_user)
            sql = "SELECT EXISTS(SELECT * FROM value_candidate WHERE id_cv= ? and id_can = ?)"

            if cursor.execute(sql, (el[0], id_user,)).fetchone()[0] != 0:
                sql = "UPDATE value_candidate SET value = ? WHERE id_cv= ? and id_can = ?"
                cursor.execute(sql, row)

            else:
                sql = "INSERT INTO value_candidate (value, id_cv, id_can) VALUES(?,?,?)"
                cursor.execute(sql, row)
        # sql = "DELETE FROM value_candidate WHERE id_can = 0"
        # cursor.execute(sql)
        connection.commit()
        connection.close()
        return redirect('/candidate')
    connection.close()
    return render_template("cndt/vacancy.html", anc_crit=anc_crit, vac_info=vac_info)

# Видалення резюме кандидата по вакансії з БД
@app.route('/open_vacancies/vacancy=<int:id_vac>/delete')
def cand_vac_del(id_vac):
    global id_user
    if id_user == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql = 'DELETE FROM value_candidate WHERE value_candidate.id_cv in (SELECT id_cv FROM crt_vac
WHERE crt_vac.id_v = ?) and value_candidate.id_can = ?;'
    cursor.execute(sql, (id_vac, id_user,))
    connection.commit()
    connection.close()
    return redirect('/candidate')

```

```

# Виведення усіх вакансій на екран у підсистемі експерта
@app.route('/expert/vacancies')
def expert_vacancies():
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    global id_expert
    if id_expert == 0:
        return redirect('/')
    vac = cursor.execute("SELECT * FROM vacancies").fetchall()

    connection.close()
    return render_template("exprt/vacancies.html", vac=vac)

# Заповнення на запис в БД МПП критеріїв вакансії
@app.route('/expert/vacancy=<int:id_vac>/mpp', methods=['POST', 'GET'])
def mpp(id_vac):
    global id_expert
    if id_expert == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql = "SELECT id_cv, name_crit FROM crt_vac INNER JOIN criteria on crt_vac.id_c =
criteria.id_crit WHERE id_v = ? ORDER BY id_cv"
    a = cursor.execute(sql, (id_vac,)).fetchall()
    mas = [e[1] for e in a]
    a = [e[0] for e in a]
    print(a)
    if request.method == "POST":
        value_crit = request.form.getlist('value_cr')
        k = 0
        for i in range(0, len(a)):
            for j in range(i + 1, len(a)):
                sql = "SELECT EXISTS(SELECT * FROM opinion_crit WHERE id_exp = ? and id_cv1 = ?
and id_cv2 = ?)"

                if cursor.execute(sql, (id_expert, a[i], a[j],)).fetchone()[0] == 0:
                    sql = "INSERT INTO opinion_crit (id_exp, id_cv1, id_cv2, value)
VALUES(?,?,?,?)"
                    cursor.execute(sql, (id_expert, a[i], a[j], value_crit[k],))
                else:
                    sql = "UPDATE opinion_crit SET value = ? WHERE id_exp = ? and id_cv1 = ? and
id_cv2 = ?"
                    cursor.execute(sql, (value_crit[k], id_expert, a[i], a[j],))
                k += 1
            connection.commit()
        return redirect('/expert/vacancies')
    sql = "SELECT name_vac, about FROM vacancies WHERE id_vac = ?"
    v = cursor.execute(sql, (id_vac,)).fetchone()
    sql = "SELECT opinion_crit.value FROM opinion_crit WHERE opinion_crit.id_exp = ? and id_cv1
in (SELECT id_cv FROM crt_vac WHERE crt_vac.id_v = ?)"
    val = cursor.execute(sql, (id_expert, id_vac,)).fetchall()
    return render_template("exprt/mpp.html", v=v, id_vac=id_vac, mas=mas, val=val)

# Видалення матриці попарних порівнянь критеріїв вакансії
@app.route('/expert/vacancy=<int:id_vac>/mpp/del', methods=['POST', 'GET'])
def mpp_del(id_vac):
    global id_expert
    if id_expert == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql="DELETE FROM opinion_crit WHERE id_exp = ? and id_cv1 in (SELECT id_cv FROM crt_vac WHERE
id_v = ?)"
    cursor.execute(sql, (id_expert, id_vac,))
    connection.commit()
    connection.close()

```

```

return redirect('/expert/vacancy='+str(id_vac)+'/mpp')

# Створити нову вакансію
@app.route('/expert/new_vacancy', methods=['POST', 'GET'])
def new_vacancy():
    global id_expert
    if id_expert == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql = "SELECT * FROM criteria"
    crit = cursor.execute(sql).fetchall()
    if request.method == "POST":
        name_vacancy = request.form['name_vacancy']
        about = request.form['about']
        ids_crit = request.form.getlist('criteria')
        sql = 'SELECT EXISTS(SELECT * FROM vacancies WHERE name_vac = ?)'

        print(cursor.execute(sql, (name_vacancy,)).fetchone()[0])
        if cursor.execute(sql, (name_vacancy,)).fetchone()[0] == 0:
            sql = "INSERT INTO vacancies (name_vac, about) VALUES(?,?)"
            cursor.execute(sql, (name_vacancy, about))
            connection.commit()
            sql = 'SELECT id_vac FROM vacancies ORDER BY id_vac DESC'
            id_vac = cursor.execute(sql).fetchone()[0]
            sql = "INSERT INTO crt_vac (id_c, id_v) VALUES(?,?)"
            for id_crit in ids_crit:
                cursor.execute(sql, (id_crit, id_vac))
            connection.commit()
            return redirect('/expert/vacancy=' + str(id_vac) + '/mpp')
        else:
            flash("Вакансія уже існує")

    return render_template("exprt/new_vacancy.html", crit=crit)

# Додати нові критерії
@app.route('/expert/new_crit', methods=['POST', 'GET'])
def new_criteria():
    global id_expert
    if id_expert == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()

    if request.method == "POST":
        crit = request.form['new_crit'].split(", ")
        for c in crit:
            cursor.execute("INSERT INTO criteria (name_crit) VALUES(?)", (c,))
        connection.commit()
    connection.close()
    return redirect('/expert/new_vacancy')

# Видалити критерії з критеріальної бази
@app.route('/expert/del_crit', methods=['POST', 'GET'])
def del_criteria():
    global id_expert
    if id_expert == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()

    if request.method == "POST":
        ids_crit = request.form.getlist('criteria')
        print(ids_crit)
        sql = "DELETE FROM criteria WHERE id_crit = ?"

```

```

        for id_crit in ids_crit:
            cursor.execute(sql, (id_crit,))

        connection.commit()
    connection.close()
    return redirect('/expert/new_vacancy')

# Відкрити/закрити набір на вакансію
@app.route('/expert/vacancy=<int:id_vac>/set_status=<int:status>/close')
def open_close_vac(id_vac, status):
    global id_expert
    global id_hr
    if id_expert == 0 and id_hr == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql = "UPDATE vacancies SET status = ? WHERE id_vac = ?"
    cursor.execute(sql, (status, id_vac))
    connection.commit()
    connection.close()
    return '<script>document.location.href = document.referrer</script>'

# Видалення вакансії з БД
@app.route('/expert/vacancy=<int:id_vac>/delete')
def delete_vac(id_vac):
    global id_expert
    if id_expert == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    connection.execute("PRAGMA foreign_keys = 1")
    cursor = connection.cursor()
    sql = "DELETE FROM vacancies WHERE id_vac = ?"
    cursor.execute(sql, (id_vac,))
    connection.commit()
    connection.close()
    return redirect('/expert/vacancies')

# Виведення результатів роботи алгоритму MAI
@app.route('/hr/vacancy=<int:id_vac>')
def hr(id_vac):
    global id_hr
    if id_hr == 0:
        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()

    sql = "SELECT name_crit FROM criteria WHERE id_crit IN (SELECT id_c FROM crt_vac WHERE id_v
    =?)"
    name_crit = cursor.execute(sql, (id_vac,)).fetchall()
    sql = "SELECT name_can, phone_can, id_can, about_me FROM candidate WHERE candidate.id_can IN
    (SELECT value_candidate.id_can FROM value_candidate WHERE value_candidate.id_cv IN (SELECT
    crt_vac.id_cv FROM crt_vac WHERE crt_vac.id_v = ?))"
    name_cand = cursor.execute(sql, (id_vac,)).fetchall()
    sql = "SELECT name_vac FROM vacancies WHERE id_vac = ?"
    name_vac = cursor.execute(sql, (id_vac,)).fetchone()
    mai, W, candW, GW = MAI(id_vac, len(name_crit))
    return render_template("hr/hr.html", mai=mai, name_crit=name_crit, W=W, name_cand=name_cand,
    candW=candW, GW=GW, name_vac=name_vac[0])

# Виведення усіх вакансій на екран у підсистемі рекрутера
@app.route('/hr/hr_vac')
def hr_vac():
    global id_hr
    if id_hr == 0:

```

```

        return redirect('/')
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    vac = cursor.execute("SELECT * FROM vacancies").fetchall()
    sql = "SELECT id_v, COUNT(id_can) FROM (SELECT crt_vac.id_v, candidate.id_can FROM candidate
INNER JOIN value_candidate ON candidate.id_can = value_candidate.id_can INNER JOIN crt_vac WHERE
value_candidate.id_cv = crt_vac.id_cv GROUP BY crt_vac.id_v, candidate.id_can) GROUP BY id_v"
    count = cursor.execute(sql).fetchall()
    sql = "SELECT DISTINCT id_v FROM crt_vac WHERE crt_vac.id_cv IN (SELECT opinion_crit.id_cv1
FROM opinion_crit)"
    ov = cursor.execute(sql).fetchall()
    ov = [list(e)[0] for e in ov]
    print(ov)

    connection.close()
    return render_template("hr/hr_vac.html", vac=vac, count=count, ov=ov)

# Розлогінення
@app.route('/logout')
def logout():
    global id_user
    id_user = 0
    global id_expert
    id_expert = 0
    global id_hr
    id_hr = 0
    return redirect('/')

if __name__ == '__main__':
    app.run(debug=True)

```

Реалізація алгоритму MAI:

```

import copy
import math
import numpy as np
from flask import Flask, render_template, request
import sqlite3
import os
from scipy.stats.mstats import gmean

currentdirectory = os.path.dirname(os.path.abspath(__file__))

# Вибірка кандидатів на вакансію
def find_cand(id_vac):
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    q1 = "SELECT DISTINCT candidate.* FROM candidate INNER JOIN value_candidate on
candidate.id_can = value_candidate.id_can INNER JOIN (SELECT crt_vac.id_cv FROM vacancies INNER
JOIN crt_vac on vacancies.id_vac = crt_vac.id_v WHERE vacancies.id_vac = ?) as cvid on
value_candidate.id_cv = cvid.id_cv ORDER BY candidate.id_can"

    return cursor.execute(q1, str(id_vac)).fetchall()

# Запуск алгоритму MAI
def MAI(id_vac, lenW):
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()
    sql = "SELECT DISTINCT id_exp FROM opinion_crit WHERE opinion_crit.id_cv1 IN(SELECT
crt_vac.id_cv FROM crt_vac WHERE crt_vac.id_v=?)"

```

```

ids_exp = cursor.execute(sql, (id_vac,)).fetchall()
ids_exp = [list(e)[0] for e in ids_exp]
W = np.zeros(lenW)
i = 0
for id_exp in ids_exp:
    i += 1
    cr = crit(id_vac, id_exp)
    W += np.array(vector(cr))
W /= i
W = np.around(W, 3)

cand = np.array(Saati(id_vac))

candW = []
for e1 in cand:
    candW.append(vector(e1))
candW = np.around(np.array(candW), 3)

GW = np.around(W.dot(candW), 3)

return find_cand(id_vac)[np.argmax(GW)], W, candW, GW

# Побудова вектору пріоритетів
def vector(matrix):
    W = []
    S = 0
    for i in range(0, len(matrix)):
        a = round(gmean(matrix[i]), 5)
        S += a
        W.append(a)

    return [num / S for num in W]

# Зведення оцінок кандидатів до шкали Саати
def Saati(id_vac):
    connection = sqlite3.connect(currentdirectory + "\\HR.db")
    cursor = connection.cursor()
    q = "SELECT value_candidate.* FROM value_candidate INNER JOIN (SELECT crt_vac.id_cv FROM crt_vac WHERE crt_vac.id_v = ?) as cv on value_candidate.id_cv = cv.id_cv ORDER BY id_cv, id_can;"
    m = cursor.execute(q, str(id_vac)).fetchall()
    arr = []
    i = 0
    while len(m) != 0:
        arr.append([])
        e = copy.deepcopy(m[0])
        while e[1] == m[0][1]:
            arr[i].append(m[0])
            m.pop(0)
            if len(m) == 0:
                break
        i = i + 1
    del m
    del e

    for i in range(0, len(arr)):
        arr[i] = np.array(arr[i]).T.tolist()

        Max = max(arr[i][2])
        Min = min(arr[i][2])
        for j in range(0, len(arr[i][2])):
            if Max != Min:
                arr[i][2][j] = (arr[i][2][j] - Min) * 8 / (Max - Min) + 1
            else:
                arr[i][2][j] = 1
        arr[i] = np.array(arr[i]).T.tolist()

```

```

b = []

for k in range(0, len(arr)):
    a = np.zeros((len(arr[0]), len(arr[0])))
    for i in range(0, len(arr[k])):
        for j in range(i, len(arr[k])):
            a[i][j] = arr[k][i][2] / arr[k][j][2]
            a[j][i] = 1 / a[i][j]

    b.append(a.tolist())

connection.close()
return b

# Вибірка МПП критеріїв по вакансії з БД
def crit(id_vac, id_exp):
    connection = sqlite3.connect(currentdirectory + "\HR.db")
    cursor = connection.cursor()

    sql = "SELECT * FROM opinion_crit WHERE opinion_crit.id_cv1 IN(SELECT crt_vac.id_cv FROM
crt_vac WHERE crt_vac.id_v=?) and opinion_crit.id_exp = ? ORDER BY id_cv1, id_cv2"
    op_crit = cursor.execute(sql, (id_vac, id_exp,)).fetchall()
    x = int((1 + math.sqrt(1 + 8*len(op_crit)))/2)
    cr = np.ones((x, x))
    for i in range(0, len(cr)):
        for j in range(i+1, len(cr)):
            cr[i][j] = op_crit[0][3]
            cr[j][i] = 1/cr[i][j]
            op_crit.pop(0)
    connection.close()
    return cr

```