

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Інформаційна система підтримки роботи фахівця по роботі з персоналом»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ.м-01/2 Маркова Анастасія Леонідівна

**Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою**

«___» грудня 2021 р.

Науковий керівник

(підпис)

к.т.н., доц., Марченко А. В.

Голова комісії

(підпис)

Шифрін Д. М.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«__» _____ 2021 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Маркова Анастасія Леонідівна

(прізвище, ім'я, по батькові)

1 Тема проекту Інформаційна система підтримки роботи фахівця по роботі з персоналом.

затверджена наказом по університету від « 29 » 10 2021 р. № 0787-IV

2 Термін здачі студентом закінченого проекту « 10 » _____ грудня _____ 2021 р.

3 Вхідні дані до проекту _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) _____

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

Дата видачі завдання _____.

Керівник _____
(підпис)Завдання прийняв до виконання _____
(підпис)**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1.	Оформлення календарного плану робіт	01.09.2021 – 10.09.2021	1.
2.	Визначення мети проекту	11.09.2021 – 15.09.2021	2.
3.	Аналіз додатків аналогів	15.09.2021 – 20.09.2021	3.
4.	Визначення функціональних вимог до інформаційної сиситеми	21.09.2021 – 30.09.2021	4.
5.	Визначення засобів реалізації	01.10. 2021 – 10.10.2021	5.
6.	Моделювання та проектування інформаційної сиситеми	11.10.2021 – 25.10.2021	6.
7.	Реалізація інформаційної сиситеми	26.10.2021 – 25.11.2021	7.
8.	Тестування розробником та незалежною особою	26.11.2021 – 30.11.2021	8.
9.	Виправлення виявлених помилок	01.12.2021 – 05.12.2021	9.
10.	Оформлення пояснювальної записки	05.12.2021 – 12.12.2021	10.
11.	Здача інформаційної сиситеми для підтримки діяльності комп'ютерного сервісу	До 15.12.2021	11.

Магістрант _____

Маркова А. Л.

Керівник роботи _____

к.т.н., доц. Марченко А. В.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Інформаційна система підтримки роботи фахівця по роботі з персоналом».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 30 найменувань, додатків. Загальний обсяг роботи – 63 сторінки, у тому числі 49 сторінок основного тексту, 3 сторінки списку використаних джерел, 12 сторінок додатків.

Кваліфікаційну роботу магістра присвячено розробці підтримки роботи фахівця по роботі з персоналом.

В роботі проведено аналіз останніх досліджень та публікацій, проведено аналіз сучасного стану застосування інформаційної системи для підтримки персоналу. На основі отриманих даних було обрентовано актуальність розроблюваної роботи.

В роботі проведено аналіз аналогів розроблюваної системи, моделювання та проектування системи.

Як результат роботи було розроблено програмне забезпечення у вигляді телеграм-боту та веб-сторінки на прикладі задачі офісного менеджменту, а саме: пошук нових кандидатів на відкриту роль у компанії, обробка резюме, комунікація з кандидатом, планування співбесід на основі результатів аналізу резюме кандидатів.

Практичне значення роботи полягає у автоматизації основних бізнес процесів рекрутингових задач: збір, обробка та зберігання відфільтрованої інформації про потенційного кандидата та більш зручною комунікацією з кандидатом на ту чи іншу посаду.

Ключові слова: телеграм бот, hr, рекрутинг, рекрутер, резюме, посада, вакансія, інформаційна сиситема, Python, Django, telegram bot.

ЗМІСТ

ВСТУП.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1. Дослідження актуальності проблеми	8
1.2. Аналіз продуктів-аналогів	10
1.3. Мета дослідження.....	15
2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	16
2.1 Постановка задачі	16
2.2 Вибір засобів реалізації.....	17
3. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	20
3.1 Структурно-функціональне моделювання проекту	20
3.2 Моделювання діаграми варіантів використання	23
3.3 Проектування моделі бази даних	25
4. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	29
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК А.....	54

ВСТУП

Технології активно впливають на всі аспекти нашого життя. Це торкнулося всіх сфер суспільного життя, в тому числі й бізнесу. Спосіб життя людей, спілкування, навчання та мислення трансформується з появою інновацій. Усе це так чи інакше змінює погляд на стиль життя і якість життя. Сучасні тенденції показують, що суспільство чекає ще більший вплив технологій на приватний та діловий сектор, як наслідок – прогнозується збільшення можливостей для подальшого розвитку бізнесу. Завдяки технологіям суспільство отримало можливість робити ті речі, які раніше вважалися неможливими. Це призвело до зниження потреб у співробітниках, однак підвищило ефективність роботи фахівців.

Нині будь-яка організація зацікавлена у пошуку та утриманні висококваліфікованих кадрів. Саме тому процес найму та відбору персоналу на вакантні посади є одним із найважливіших питань та ключовим завданням у системі управління персоналом. Залучення в організацію потенційних кандидатів є однією з важливих функцій системи управління людськими ресурсами організації, оскільки це безпосередньо впливає на майбутнє компанії та її імідж.

Бізнес-процеси у сфері Human Resources на сьогоднішній день слабо автоматизовані в більшості компаній. На відміну від кадрового діловодства, де все регламентовано нормами законодавчих актів та трудового кодексу, сфера управління людськими ресурсами найчастіше спирається на імпровізацію та неформалізовані процеси, що склалися з часом.

Автоматизована Human resource management (HRM) система, дозволить оперативно знайти інформацію щодо відповідальних осіб та дані для зв'язку, тобто взаємодіяти з кандидатами чи співробітниками в режимі реального часу.

Актуальність даної роботи полягає в автоматизації процесу управління персоналом надаючи спеціалістам по персоналу достатньо часу для роботи над важливими видами діяльності, такими як інтерактивні послуги співробітників,

розвиток кар'єри та організаційне проектування замість того, щоб витратити час на рутинні завдання.

На основі аналізу додатків аналогів було сформовано мету роботи - створити інформаційну систему збору та обробки інформації, необхідної для підбору кандидата на вакантну посаду, що, в свою чергу, полегшить HR-спеціалісту або рекрутеру перегляд інформації, призначення співбесіди на ту чи іншу посаду в компанії та комунікацію з потенційним кандидатом.

Для досягнення поставленої мети необхідно вирішити певний перелік задач:

- проаналізувати предметну область управління персоналом;
- виділити основні вимоги інформаційної системи, що розробляється;
- вибрати засоби проектування та розробки;
- спроектувати інформаційну систему;
- розробити інформаційну систему;
- провести тестування інформаційної системи.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Дослідження актуальності проблеми

Пошук кандидата на вакантну посаду ускладнюється використанням численних цифрових технологій. Останні, з одного боку, автоматизують частину бізнес-процесів, а з іншого боку навантажують HR-фахівця додатковими задачами перегляду онлайн-дошок.

В сучасному інфрпросторі України задачі підбору персоналу вивчають створення стратегій діяльності рекрутингових компаній, як практичну проблему з пристосуванням технологій моделей діяльності іноземних агенцій по підборі персоналу, а не, як теоретичні дослідження самотійного пошуку спеціалістів самим підприємством [1].

У ході створення українського ринку праці, на ньому відбуваються суттєві зміни: якщо раніше проблемою був пошук роботи, то сьогодні – пошук «кращого» кандидата [2]. Пошук та аналіз резюме потрібних спеціалістів, первинний відбір та співбесіда – все це займає багато часу та більш того – багато зусиль, які, при неправильному підході підбору персоналу, можуть просто не досягати поставлених результатів.

Автоматизація HR-функцій є найважчою з технологічної точки зору, тому що в даному випадку автоматизована система повинна працювати з параметрами, які важко формалізуються або взагалі неформалізуються, частина з яких взагалі має пряме відношення до психології. У «профілі компетенцій співробітника» крім традиційних для обліку параметрів (рівень освіти, спеціальність, досвід роботи) враховуються такі показники, як адаптивність, лідерські якості, здатність працювати в колективі, відкритість мислення, «порахувати» які просто неможливо через суб'єктивний характер їх трактування [3-4]. Але з швидким розвитком інформаційних технологій ця задача стає нам посильною.

Необхідність полегшення роботи менеджерів з пошуку потрібних спеціалістів на вакантну посаду, збору, зберігання, переробки та аналізу великого обсягу отриманих даних, надання її користувачам у зручній формі ставить завдання створення відповідної інформаційної системи.

Програмне забезпечення для відстеження кандидатів пропонує ряд переваг та корисних функцій протягом усього процесу найму.

По-перше, інформаційна система підтримки роботи фахівця з підбору персоналу пришвидшує весь цикл набору персоналу в цілому.

По-друге, користувач, незалежно від того, найнятий він чи ні, отримає вражаючий досвід роботи з автоматизованою системою відстеження кандидатів. Більш ефективний процес прийому на роботу та подання заявок – ключ до успіху будь-якого бізнесу.

Ці ж кандидати можуть сказати позитивні слова про компанію, що допоможе їй зайняти сильні позиції на ринку [5].

Переважно, люди подають заяву про прийом на роботу наосліп, не звертаючи уваги на вимоги до роботи та свою відповідність ним. У таких випадках HR відчуває сильне розчарування, оскільки йому доводиться фільтрувати сотні додатків, а потім відфільтровувати ті, які стосуються претендентів вакансій, а потім відбирати з них найбільш відповідні для цього завдання.

Тут можна побачити третю перевагу - інформаційна система дозволяє уникнути аналізу непотрібних заяв про прийом на роботу.

Крім того, розроблювана інформаційна система дозволить спростити розробку розкладу співбесід як для кандидата на вакантну посаду, так і для HR-спеціалістів [6-7].

Інформаційна система може надсилати автоматично згенеровані електронні листи відібраним кандидатам, які можуть інформувати їх про співбесіду, тим самим заощаджуючи багато часу та перешкод для відділу кадрі [8].

Таким чином, розробка інформаційної системи для підтримки роботи фахівця по роботі з персоналом дозволить скоротити часові витрати на пошук

на підбір кандидатів, підвищить якість результатів цього процесу, сприятиме покращенню іміджу компанії, зміцнить її становище на ринку.

1.2. Аналіз продуктів-аналогів

Сьогодні на ринку існує багато програм, так чи інакше пов'язаних із підбором персоналу та управлінням співробітників. Програмні рішення різняться за функціоналом, структурою.

Розглянемо кілька варіантів, проаналізуємо переваги та недоліки.

Програма «Workbright»

WorkBright пропонує повністю віддалене підключення для будь-якої організації. Додаток дозволяє керувати документами про новий найм та файлами співробітників з будь-якого місця, а також відправляти автоматичні нагадування протягом всього процесу адаптації після того, як співробітник прийняв вашу пропозицію (рис 1.1 – 1.2).



Рисунок 1.1 – Головна сторінка інформаційної системи «Workbright»

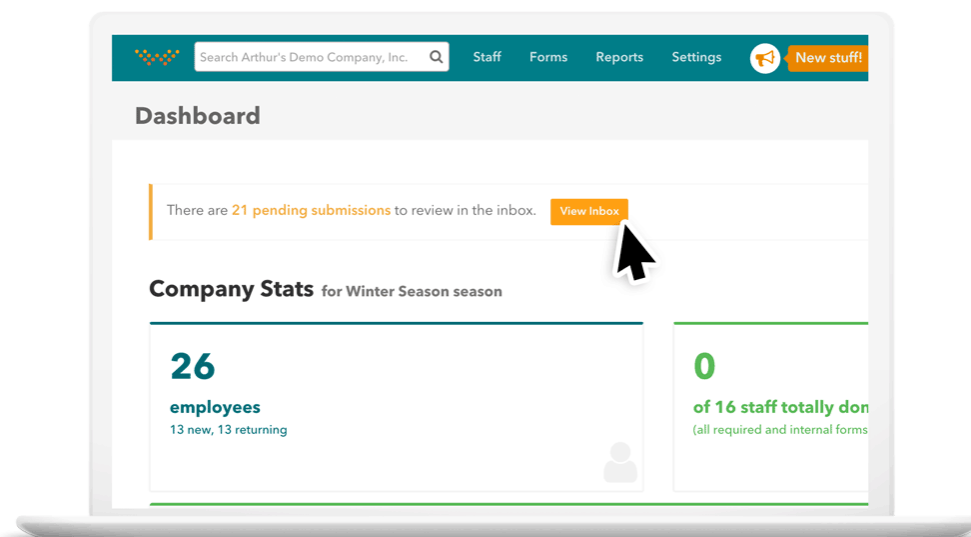


Рисунок 1.2 – Сторінка дошок системи «Workbright»

Ціни на WorkBright починаються зі 158 доларів на місяць і збільшуються в залежності від розміру компанії. Вони пропонують індивідуальні розцінки, щоб якнайкраще задовольнити потреби вашого бізнесу.

Програма «Резюмакс»

Основна ціль інформаційно системи «Резюмакс» - управління даними та відстеження інформації про організацію, професійний досвід та навички співробітників (рис. 1.3).

Додаток автоматизує підбір персоналу в корпорації або кадровому агентстві, яке має бажання покращити комунікацію між рекрутерами та начальниками відділів.

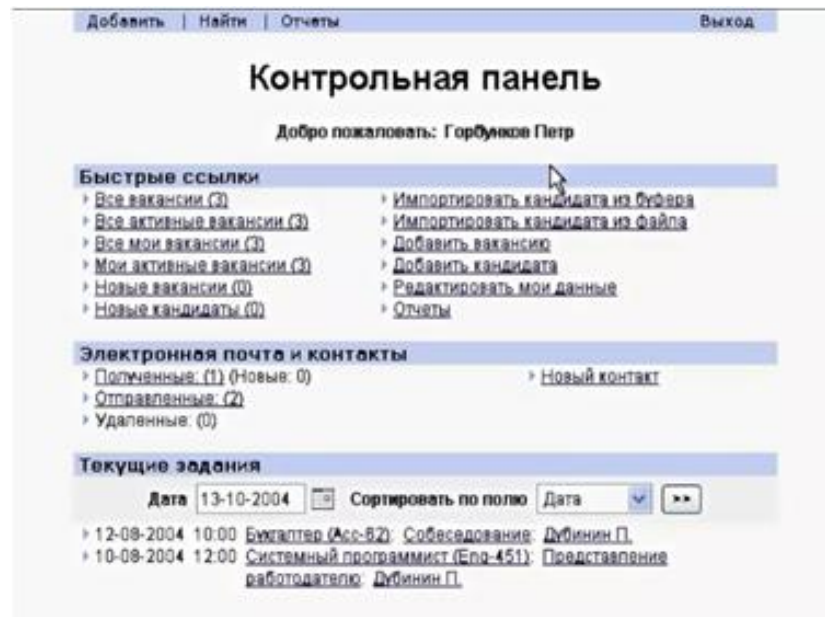


Рисунок 1.3 – Контрольна панель інформаційної системи «Резюмакс»

Інформаційна система «Резюмакс» надає можливість у повному обсязі здійснювати облік кандидатів та наявних вакансій. Крім того, «Резюмакс» має можливість повідомляти кандидата та HR-спеціаліста про майбутні призначені співбесіди.

Дана інформаційна система має складний для розуміння пересічним користувачем інтерфейс, а також високу вартість та відсутність супровідної документації.

Програма «E-StaffРекрутер»

Основні функції даної інформаційної системи – автоматизоване розміщення вакансій, пошук резюме на професійних порталах. Програму можна використовувати як комплексну програму для співробітника кадрового відділу, і універсальний інструмент великих кадрових агентств (рис. 1.4).

The screenshot displays the E-StaffRecruiter web application. The main content area shows a table of vacancies with columns for 'Наименование', 'Структурное подразделение', 'Кол во позиц.', 'Кол во откр. позиц.', 'Дней в работе', 'Текущий статус', 'Дата', and 'Финальный к'. The table lists various roles such as 'Водитель', 'Главный бухгалтер', 'Делопроизводитель', 'Инженер по закупкам', 'Инженер по тестированию', 'Инженер проектирования', 'Инженер-технолог', 'Кассир', and 'Кассир'. The 'Делопроизводитель' role is highlighted, and a detailed view of this vacancy is shown below, including a table of candidate statuses: 'Рассматриваются' (2), 'Отклонены после интервью с заказчиком' (1), 'Резерв' (1), and 'Отклонены' (1). Below this, a table shows candidate details with columns for 'ФИО', 'Статус', 'Дата', 'Последнее место работы', and 'Рекрутер'. Candidates listed include 'Кизикова Виктория' (Принят на работу, 05.07.2018) and 'Плотникова Ангелина Петровна' (Отклонены после интервью с заказчиком, 05.07.2018). The interface also features a left sidebar with navigation options like 'Орг. структура', 'Вакансии', 'Кандидаты', and 'События', and a top navigation bar with user profiles and search filters.

Рисунок 1.4 – Сторінка переліку вакансій інформаційної системи «E-StaffРекрутер»

Переглянувши інформаційну систему «E-StaffРекрутер» слід відзначити, що дана система має зручний інтерфейс, база даних зберігається на персональному комп'ютері, інформація про кандидатів зберігається у вигляді *.xml файлів. Добре розроблена безпека даних.

Проте в даній інформаційній системі відсутня можливість виключати опрацьованих у попередніх запитах кандидатів із результатів нового пошуку.

Після проведення аналізу додатків-аналогів була розроблена порівняльна характеристика аналогів та розробленої інформаційної системи за кожним критерієм, що представлена у табл. 1.1.

Таблиця 1.1. – Порівняльна характеристика аналогів інформаційних систем

Критерії	«Workbright»	«Резюмакс»	«E-Staff Рекрутер»	ІС підтримки роботи фахівця по роботі з персоналом
Скринінг-система резюме	-	-	-	+
Зворотній зв'язок з клієнтом	+	+	-	+
Робота над вакансіями	+	+	+	+
Формування звітності	-	+	-	-
Безкоштовний	-	-	-	+
Зручний та інтуїтивно зрозумілий інтерфейс та дизайн	+	-	+	+

1.3. Мета дослідження

Визначивши актуальність та розглянувши аналоги інформаційних систем, було сформульовано мету дипломного проекту, яка полягає у створенні інформаційної системи підтримки роботи фахівця по роботі з персоналом. Основною метою даної системи є автоматизація роботи HR-спеціаліста, яка дозволить скоротити часові витрати на пошук та підбір кандидатів, підвищить якість результатів цього процесу, сприятиме покращенню іміджу компанії, зміцнить її становище на ринку.

Дана інформаційна система надасть ряд переваг:

- швидкий пошук кандидатів;
- економія часу та підвищення ефективності;
- оптимізація витрат та ресурсів;
- наочність подання інформації.

2. ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Постановка задачі

Основна мета кваліфікаційної роботи магістра полягає у розробленні інформаційної системи підтримки роботи фахівця по роботі з персоналом для автоматизації основних бізнес процесів, збирання, обробки та зберігання відфільтрованої інформації про потенційного кандидата та більш зручної комунікації з кандидатом на ту чи іншу посаду.

Створена інформаційна система матиме дві інтегровані між собою частини: телеграм бот та веб-сервіс. Інформаційна система повинна мати зручний та інтуїтивно зрозумілий інтерфейс, надавати можливість кандидату відправляти резюме, отримувати необхідну інформацію у разі, якщо резюме не пройшло необхідну перевірку вбудованою скринінг-системою, надавати можливість HR-спеціалісту переглядати резюме кандидатів у відфільтрованому вигляді по доступним вакансіям та підтримувати зв'язок з конкретним кандидатом за допомогою інтеграції телеграм-боту та веб-сервісу.

Було визначено наступний перелік задач для досягнення поставленої мети проекту:

- проаналізувати предметну область;
- проаналізувати створенні продукти-аналоги;
- обрати засоби реалізації проекту;
- розробити структуру інформаційної системи;
- реалізувати дану систему у вигляді інтегрованого телеграм-боту та веб-сервісу;
- розробити функціонал інформаційної системи;
- оформити супровідну документацію.

Розроблювана інформаційна система повинна відповідати наступним функціональним вимогам:

- перегляд інформації про вакансії;
- перегляд інформації про резюме;
- можливість завантаження файлу резюме на ПК;
- можливість обробки та фільтрації резюме;
- зворотній зв'язок з клієнтом.

2.2 Вибір засобів реалізації

Для розробки інформаційної системи буде враховано сучасні тенденції та технології при проектуванні програмного забезпечення.

Для початку визначимо мову програмування для розробки інформаційної системи управління персоналом. Тут важливо виділити наступні критерії оцінювання:

- кросплатформна мова, яка дозволяє працювати розробленій системі в різних операційних системах;
- легкість та швидкість освоєння;
- безкоштовний або умовно безкоштовний;
- доступність літератури;
- об'єктно-орієнтований;
- дозволяє легко створювати парсери для збирання інформації із файлів;
- надійність.

Розглядалися найпопулярніші мови програмування такі, як java, c++, c#, php, python за даними <https://habr.com> (рис 2.1) [9].

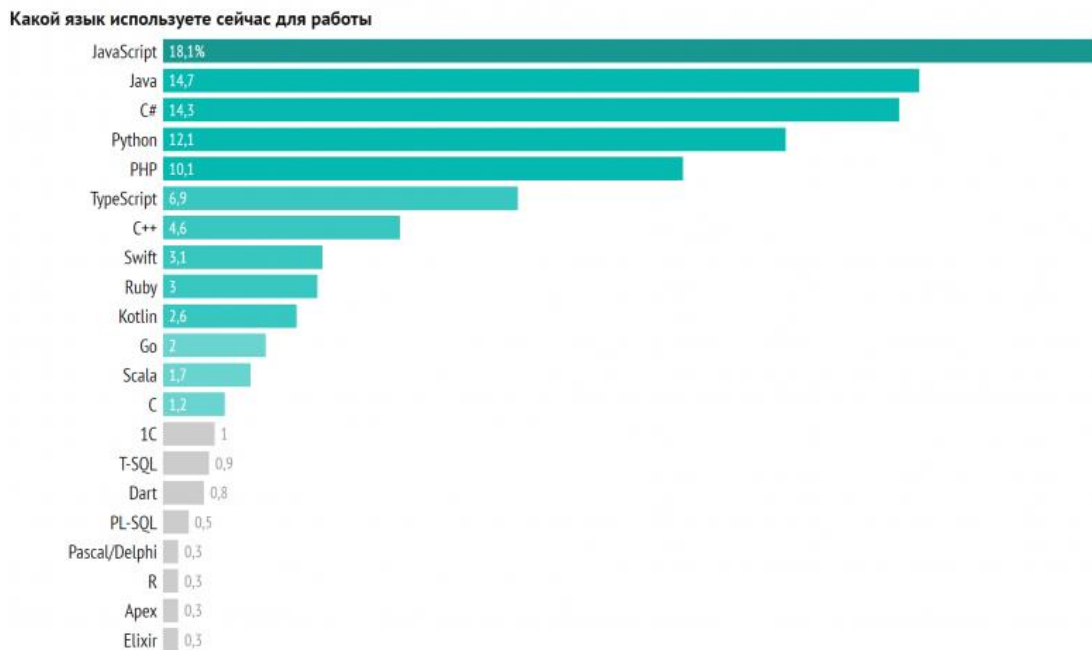


Рисунок 2.1 - Рейтинг мов програмування 2021.

Коли мова заходить про створення парсерів, всі згадують PHP і Python – саме ці мови працюють на стороні сервера. Незважаючи на те, що безліч програм все ще працює на PHP, зараз набагато вигідніше користуватися Пітоном [10-11].

Python є високорівневою мовою. Він не має конкретного призначення, тому підходить для вирішення будь-яких завдань, у тому числі для написання парсерів. Синтаксис Пітона мінімальний, причому мова підтримує ООП, функціональні, імперативні та інші типи програмування. Завдяки цьому парсинг на Python набагато зручніший.

Python вважається більш універсальним, ніж php. Його лаконічність приносить йому помітну затребуваність у таких областях, як:

- розробка веб-додатків з використанням фреймворків Django/Flask (BackEnd напрямок);
- завдання системного адміністрування;
- DevOps інженерія (написання скриптів);
- автоматизоване тестування (Automation QA);
- робота з великими обсягами даних,
- машинне та глибоке навчання, нейронні мережі (Big Data, Data Science) [12].

PHP може використовуватися для інших цілей, але він призначений саме для створення веб-сторінок, і це те, що він робить найкраще. Це складна мова програмування, призначена для створення складних веб-програм.

Python має величезну кількість бібліотек з відкритим кодом та документацією. У Python є бібліотека Telebot, за допомогою якої розробнику з легкістю можливо створити чат-бота в месенджері Telegram, використовуючи готові класи та функції [13].

Оскільки головна задача розроблюваної інформаційної системи - збирання та обробка інформації, необхідної для підбору кандидатів на вакантну посаду з використанням телеграм боту, в якості мови програмування було обрано Python.

Також на його користь було ухвалено рішення через його популярність серед фахівців з розробки, існує безліч доступної літератури та відеопідкастів для його вивчення і він є одним із найбільш затребуваних на ринку праці.

В якості фреймворка для веб-розробки на Python було використано Django фреймворк.

Архітектура Django схожа на «Модель-Уявлення-Контролер» (MVC). Контролер класичної моделі MVC приблизно відповідає рівню, який Django називається Представлення (View), а презентаційна логіка Представлення реалізується в Django рівнем Шаблонів (Templates). Через це рівневу архітектуру Django часто називають "Модель-Шаблон-Представлення" (MTV) [14].

Django підтримує бази даних PostgreSQL, MySQL, SQLite та Oracle. Так, як стандартна бібліотека Python вже містить модуль sqlite3, в розроблюваній роботі буде використовуватися СУБД SQLite. Крім того, SQLite є найпростішою базою даних, до якої можна підключитися за допомогою Python, оскільки для цього не потрібно встановлювати зовнішні модулі. Більш того, SQLite база даних не вимагає сервера і є самодостатньою, тобто просто читає та записує дані у файл [15-16].

Таким чином, для розробки інформаційної системи було обрано мову програмування Python, систему управління базами даних SQLite, середовище розробки – PyCharm [17].

3. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Структурно-функціональне моделювання проекту

IDEF0 – методологія функціонального моделювання і графічного описання процесів. Завдяки методології функціонального моделювання, систему, яку будуть розроблювати можна побачити у вигляді набору взаємопов'язаних функцій (функціональних блоків). Побудова моделі IDEF0 постає перед розробником спочатку з уявлення системи як єдиного цілого – одного функціонального блоку з інтерфейсними дугами, що тягнуться за межі області. Така діаграма з одним функціональним блоком називається контекстною діаграмою і позначається ідентифікатором «А-0». [18].

Тому процес проектування web-системи необхідно розпочинати з розробки контекстної діаграми IDEF0. До контекстної діаграми входять:

- Вхід – ресурси, що використовуються для виконання функції.
- Вихід – результати виконання функції.
- Управління – керуючі впливи, які визначають, як і скільки потрібно зробити результатів.
- Механізми – ресурси, які виконують роботу (наприклад, співробітники, обладнання, пристрої тощо).

На рис.2.1 зображено діаграму IDEF0, на рис.2.2 зображена декомпозиція діаграми.

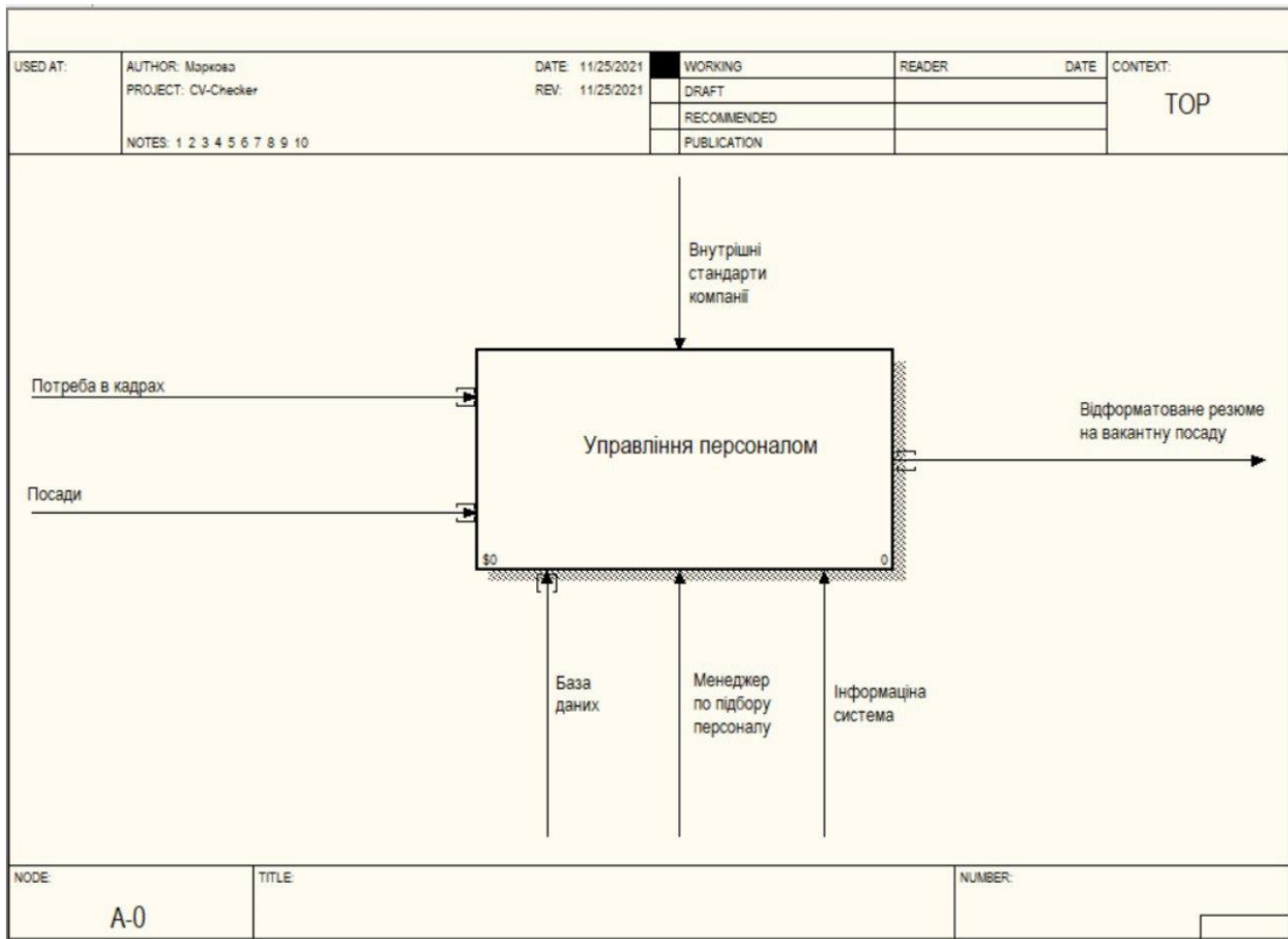


Рисунок 2.1 – Функціональне моделювання інформаційної системи в IDEF0

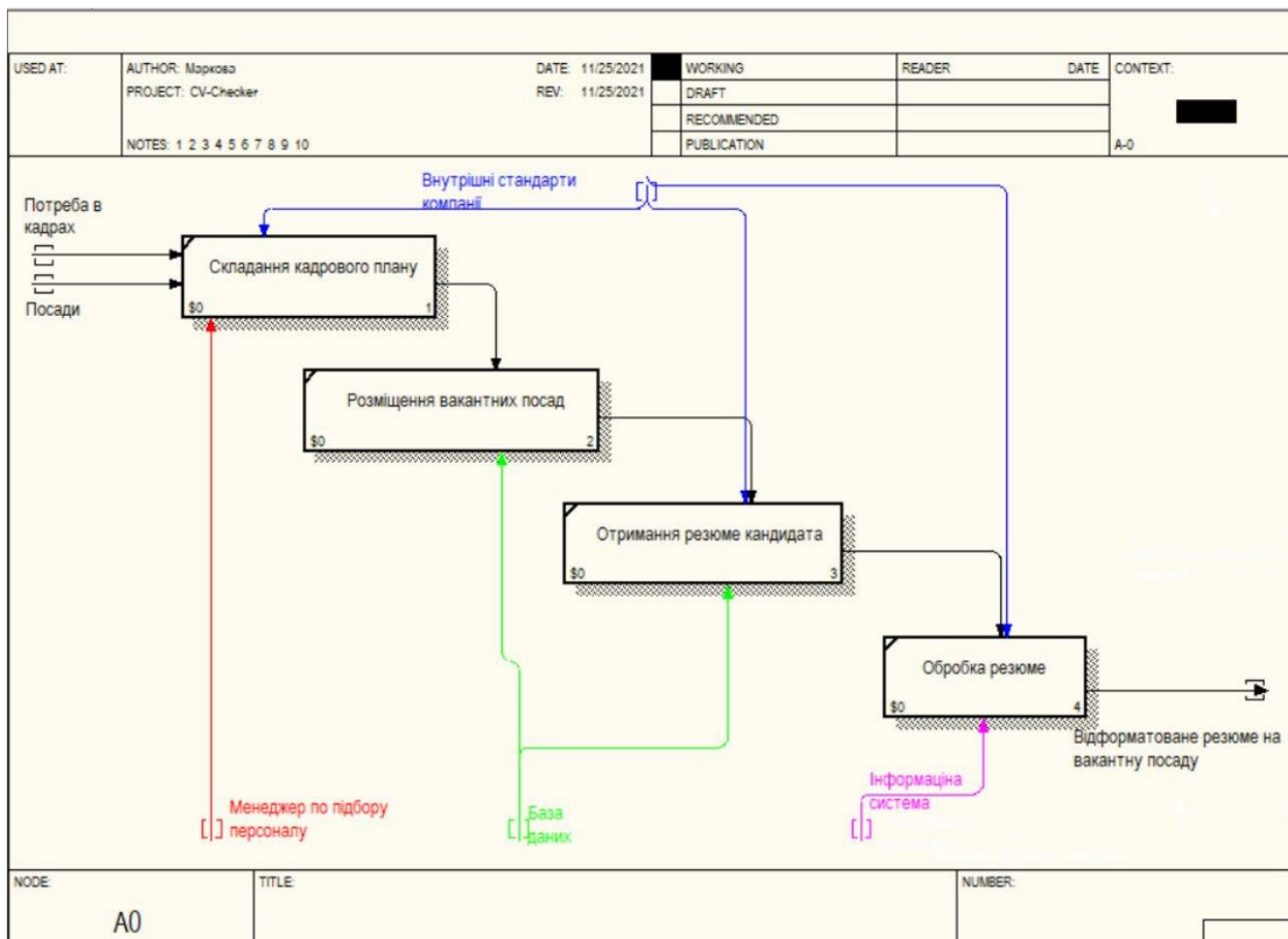


Рисунок 2.2 – Декомпозиція діаграми IDEF0

Моделювання діаграми варіантів використання

Після розробки контекстних діаграм, необхідно розробити діаграму варіантів використання. Діаграма варіантів використання – це діаграма, що описує, який функціонал програмної системи, що розробляється, доступний кожній групі користувачів. [19].

Було визначено наступних акторів для розроблюваної інформаційної системи:

- HR – має доступ до функціоналу інформаційної системи та може вносити зміни до контенту;
- База Даних – зберігає інформацію;
- Кандидат на вакантну посаду – має доступ до частини функціоналу інформаційної системи.

Далі було проідентифіковано варіанти використання (процесів, які можуть виконувати користувачі) для даної інформаційної системи, а саме:

- ВВ «Авторизація» - ВВ дає можливість користувачу авторизуватися у системі;
- ВВ «Додавання нових вакантних посад» - ВВ дає можливість менеджеру додавати актуальні вакантні посади до інформаційної системи;
- ВВ «Перегляд всіх резюме» - ВВ дає можливість менеджеру здійснювати перегляд надісланих кандидатами резюме;
- ВВ «Перегляд конкретного резюме» - ВВ дає можливість менеджеру відкривати конкретне резюме кандидата;
- ВВ «Зворотній зв'язок» - ВВ дозволяє менеджеру надсилати повідомлення кандидату;

- ВВ «Відправити резюме» - ВВ дає можливість кандидату відправити резюме на обрану вакантну посаду.

Діаграма варіантів використання зображена на рис.2.3

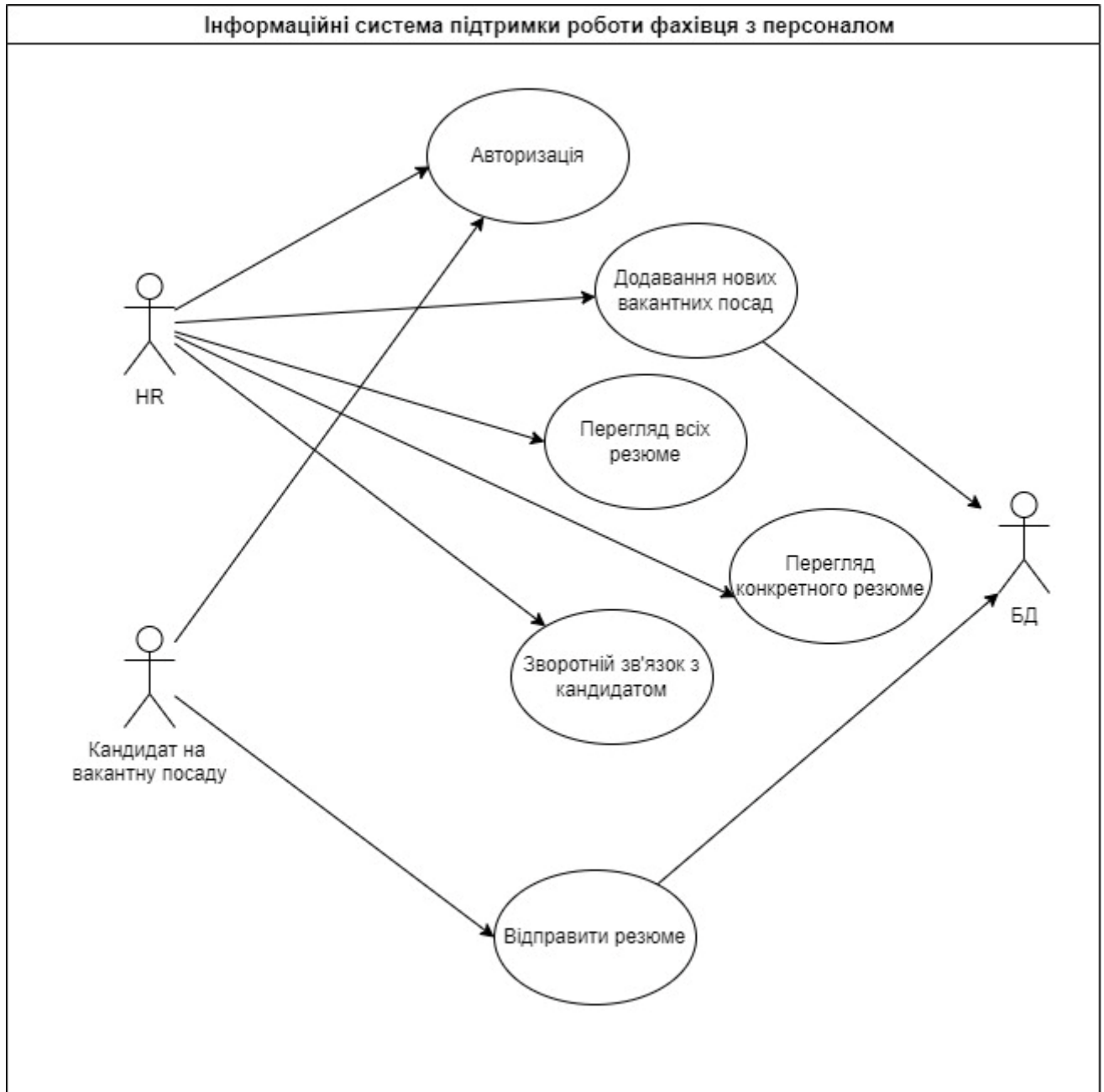


Рисунок 2.3 – Діаграма варіантів використання

Проектування моделі бази даних

Інформаційна база системи була реалізована з використанням СУБД SQLite через незначний обсяг даних, який буде зберігатись у базі даних сайту. SQLite — компактна реляційна база даних, що вбудовується. Вихідний код бібліотеки передано до суспільного надбання.

Слово "вбудований" означає, що SQLite не використовує архітектурну модель клієнт-сервер. Тобто двигун SQLite не є окремо працюючим процесом, з яким взаємодіє програма, а надає бібліотеку, з якою програма компонується і двигун стає складовою програми. Такий підхід зменшує накладні витрати, час відгуку та спрощує програму. [20-21].

Для бази даних були виділені наступні сутності:

- users – зберігає інформацію про користувачів;
- candidates – зберігає інформацію про кандидатів;
- resumes – зберігає інформацію про резюме;
- vacancies – зберігає інформацію про вакантні посади;

У табл. 2.1. наведена інформація про атрибути до відповідних сутностей.

Таблиця 2.1. – Інформація про атрибути та сутності

Назва сутності	Назва атрибуту	Тип атрибуту	Ключі	Опис атрибуту
users	id_user	INT	PK	Унікальний ідентифікатор користувача
	username	VARCHAR(50)	-	Логін користувача
	email address	VARCHAR(50)	-	Пошта користувача
	first name	VARCHAR(50)		Ім'я користувача
	last name	VARCHAR(50)		Прізвище користувача
	staff status	ENUM(«true», «false»)		Наявність HR статусу
vacancies	vacancy	INT	PK	Унікальний ідентифікатор вакансії
	name	VARCHAR(50)		Назва вакансії
	description	VARCHAR(250)		Опис вакансії

	status	ENUM(«Open», «Close»)		Статус вакансії
	timestamp	DATE()		Дата додавання
candidates	candidate	INT	PK	Унікальний ідентифікатор кандидату
	name	INT		Ім'я кандидата
	surname	VARCHAR(250)	-	Прізвище кандидата
	email address	VARCHAR(250)	-	Пошта кандидата
resumes	candidate	INT	FK1	Унікальний ідентифікатор кандидату
	vacancy	INT	FK2	Унікальний ідентифікатор вакансії
	file	VARCHAR(250)	-	Файл резюме
	cover letter	VARCHAR(250)	-	Короткі відомості про кандидата
	extracted text	VARCHAR(250)	-	Основні відомості з файлу резюме
skills	skill	INT	PK	Унікальний ідентифікатор навички
	candidate	INT	FK1	Унікальний ідентифікатор кандидату
additional_info	id_info	INT	PK	Унікальний ідентифікатор кандидату
	candidate	INT	FK1	Унікальний ідентифікатор кандидату
	name	VARCHAR(50)		Назва інформації
	value	VARCHAR(50)		Значення інформації
required_skills	required_skills	INT	PK	Унікальний ідентифікатор необхідної навички
	vacancy	INT	FK2	Унікальний ідентифікатор вакансії
	value	VARCHAR(250)	-	Значення необхідної навички

Визначивши сутності та атрибути інформаційної системи була побудована ER-діаграма, яка зображена на рис. 2.4.

ER-діаграма являє собою графічне представлення сутностей та їх взаємозв'язків між собою.

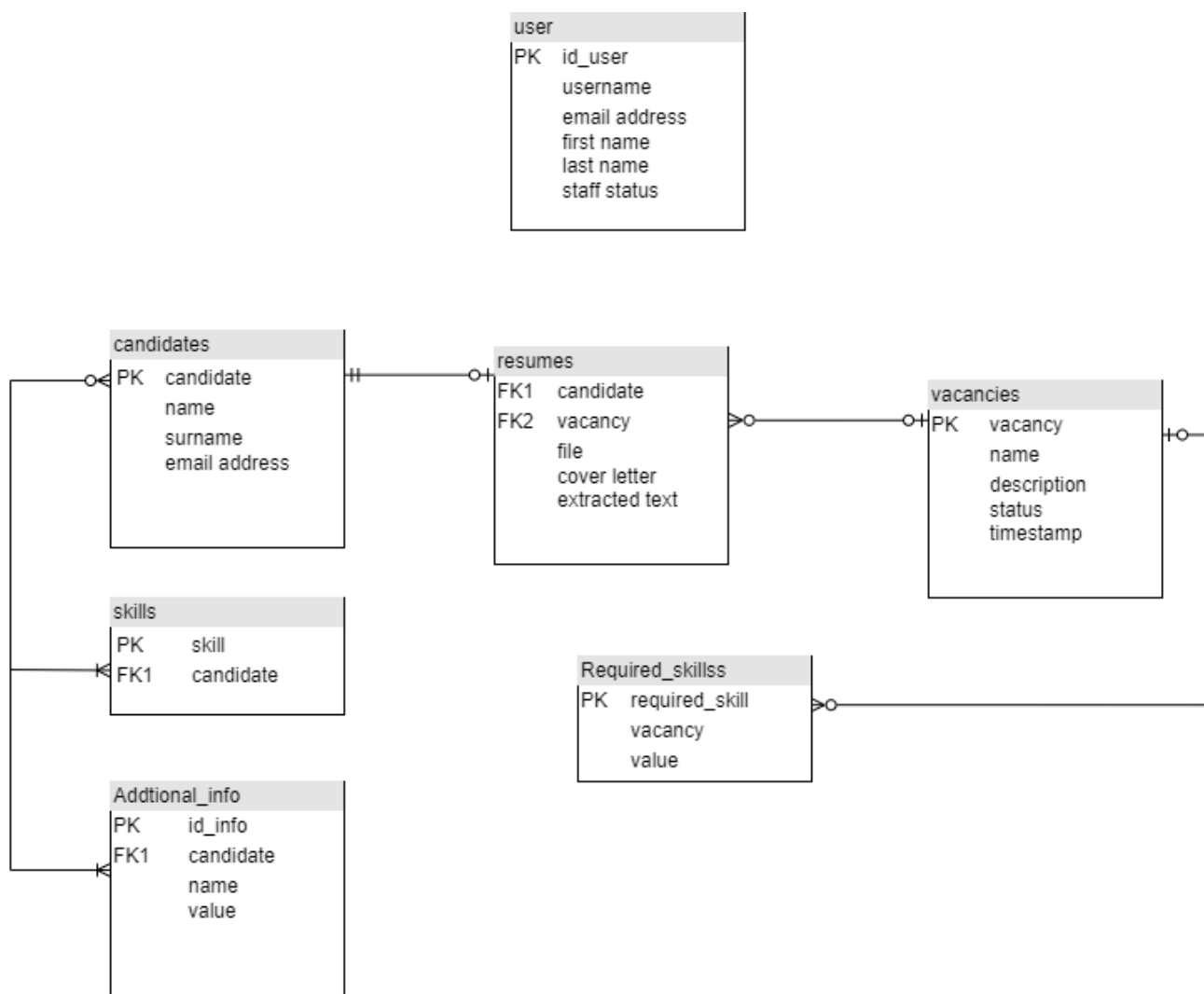


Рисунок 2.4 – ER діаграма

За допомогою даної діаграми можна побачити, як сутності зв'язані між собою в інформаційній системі:

- candidates та resumes мають зв'язок «один до одного», один кандидат може відправити одне резюме, та одне резюме повинно відноситися до одного кандидата;
- vacancies та resumes мають зв'язок «один до багатьох», одне резюме може відноситися до однієї вакансії, одна вакансія може мати багато резюме;
- vacancies та required skills мають зв'язок «один до багатьох», одна навичка може відноситися до однієї вакансії, одна вакансія може мати багато навичок;

- candidates та skills мають зв'язок «багато до багатьох», одна навичка може відноситися до багатьох кандидатів, один кандидат повинен мати хоча б одну навичку;
- candidates та additional info мають зв'язок «багато до багатьох», один пункт може відноситися до багатьох кандидатів, один кандидат повинен мати хоча б один параметр.

4. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Архітектура програмного додатку

За основу архітектури інформаційної системи розглядилась "клієнт-сервер" архітектура та "файл-сервер" архітектура.

Файл-серверні програми — це виділений комп'ютер у мережі, призначений для зберігання файлів. До нього організований спільний доступ користувачів, які можуть завантажувати, завантажувати, змінювати та видаляти файли.

Основний недолік роботи с файловою системою це те, що дані зберігаються одному місці, а обробляються іншому. Фактично це значить, що дані потрібно передавати по мережі, що, в свою чергу, призводить до високих навантажень мережі і, як наслідок, продуктивність програми зменшується зі збільшенням кількості клієнтів, що одночасно працюють. Другим недоліком такої архітектури є те, що вона не може забезпечити необхідний рівень надійності та достовірності інформації. Така архітектура не підтримує цілісність баз даних і не мають механізмів управління транзакціями, що ускладнює забезпечення логічної несуперечності інформації при збоях обладнання та програм. [22].

Головна відмінність архітектури клієнт-сервер від архітектури файл-сервер є абстрагування від внутрішнього представлення даних (фізичної схеми даних).

Наразі клієнтські програми маніпулюють даними на рівні логічної схеми. Як наслідок, використання архітектури клієнт-сервер підтримує максимальний рівень надійності зберігання, актуальність та достовірність програм, що розраховані на багато користувачів ІС з централізованою базою даних, незалежні від апаратної частини сервера БД, які підтримують графічний інтерфейс користувача на клієнтських станціях, пов'язаних локальною мережею. Крім того, вона дозволяє поєднати різні клієнти. Використовувати ресурси одного сервера

часто можуть клієнти з різними апаратними платформами, операційними системами тощо. [23].

Отже для даної інформаційної системи було обрано клієнт-серверну концепцію.

Model-view-controller (MVC) - шаблон архітектури програмного забезпечення, який має на увазі поділ програми на 3 слабозв'язані компоненти, кожен з яких відповідає за свою сферу діяльності. Основна мета застосування MVC полягає у розподілі даних та бізнес-логіки від візуалізації. За рахунок такого розподілу підвищується можливість повторного використання програмного коду. Також спрощується і супровід програмного коду: внесення змін до зовнішнього вигляду, наприклад, не відбивається на бізнес-логіці, а зміни бізнес-логіки не торкаються візуалізації.

Концепція MVC дозволяє розділити дані, представлення та обробку дій користувача на три окремі компоненти.

Більш того, фреймворк за допомогою якого розроблюється інформаційна система (Django) слідує концепції MVC досить суворо, щоб його можна було називати "MVC фреймворк". Ось короткий опис того, як M, V та C розділені в середовищі Django:

Модель (англ. Model). Сюди відноситься все, що пов'язано з даними - як отримати до них доступ, як перевірити їх, яке у них поведінка та ставлення один до одного.

Подання, вид (англ. View). Відповідає за відображення інформації (візуалізацію). Одні й самі дані можуть представлятися різними способами й у різних форматах.

Контролер (англ. Controller). Забезпечує зв'язок між користувачем та системою: контролює введення даних користувачем та використовує модель та уявлення для реалізації необхідної реакції.

Як уявлення, і контролер залежить від моделі. В свою чергу модель не залежить від контролера та уявлення. Відповідно до цього, реалізується призначення такого поділу: воно дозволяє будувати модель незалежно від

візуального уявлення, а також створювати кілька різних уявлень для однієї моделі. На рисунку 4.1 представлено схему MVC.

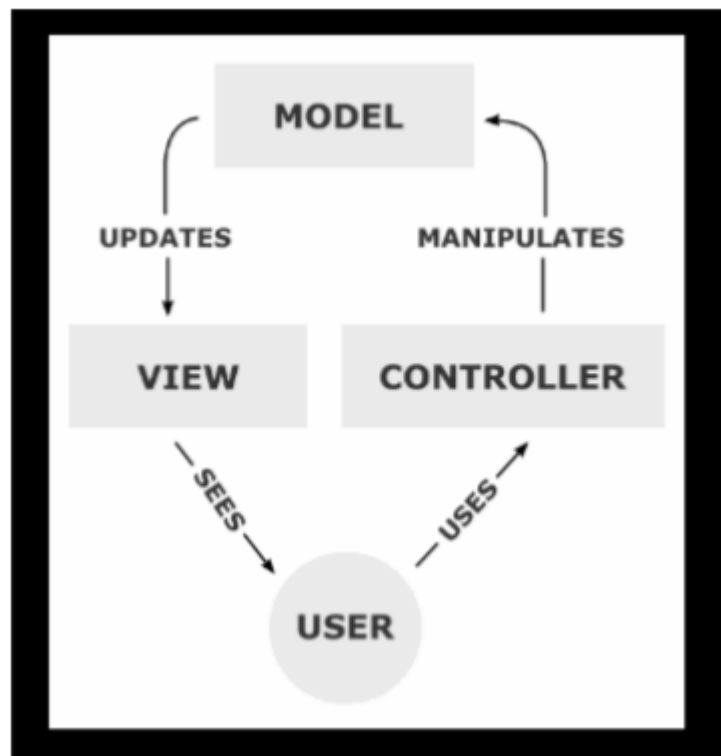


Рисунок 4.1 – Модель MVC

Така ж структура буде характерна для структури інформаційної системи, що розробляється.

Клієнт-серверна архітектура даної інформаційної системи може бути описана наступним чином.

Користувач потрапляє до інформаційної системи через веб-браузер. Далі користувач переходить до особистого кабінету, в той момент, запит передається до веб-серверу, веб-сервер, в свою чергу, передає користувачу дані, які він може побачити на веб сторінці. Якщо клієнт хоче відправити резюме, а потім відправити кандидату повідомлення, з боку клієнтської частини дані проходять валідацію і відправляються на веб-сервер. На веб-сервері дані проходять валідацію, якщо все коректно – відбувається запит до серверу СУБД, куди відбувається запис про кандидата. Веб-сервер передає користувачу запрошені користувачем дані з БД: інформація про кандидата та його резюме.

4.2 Реалізація бази даних

Під час створення і редагування бази даних інформаційної системи використовується Django administration. Одна з найсильніших сторін Django – це автоматичний інтерфейс адміністратора. Він використовує мета-дані моделі, щоб надати багатофункціональний, готовий до використання інтерфейс для роботи з вмістом сайту. SQLite - це легка та проста в налаштуванні система управління реляційними базами даних. SQLite не вимагає сервера, що є його найбільшою перевагою. На відміну від інших СУБД, таких як MySQL або PostgreSQL, сервер не потрібний для запуску бази даних. Таким чином, нам не потрібне встановлення [24].

Бази даних SQLite зберігаються локально разом із файлами, що зберігаються на диску. Це значно прискорює доступ до даних у базі даних та управління ними.

Усі транзакції SQLite відповідають вимогам ACID.

Для створення бази даних SQLite на Python був використаний вбудований модуль sqlite3. Модуль sqlite3 надає API, за допомогою якого можна створювати базу даних. Він сумісний з базою даних Python API і не вимагає ніяких зовнішніх бібліотек.

База даних розроблюваної системи має назву «cv-checker». Фізична реалізація представлена на рис. 4.2.

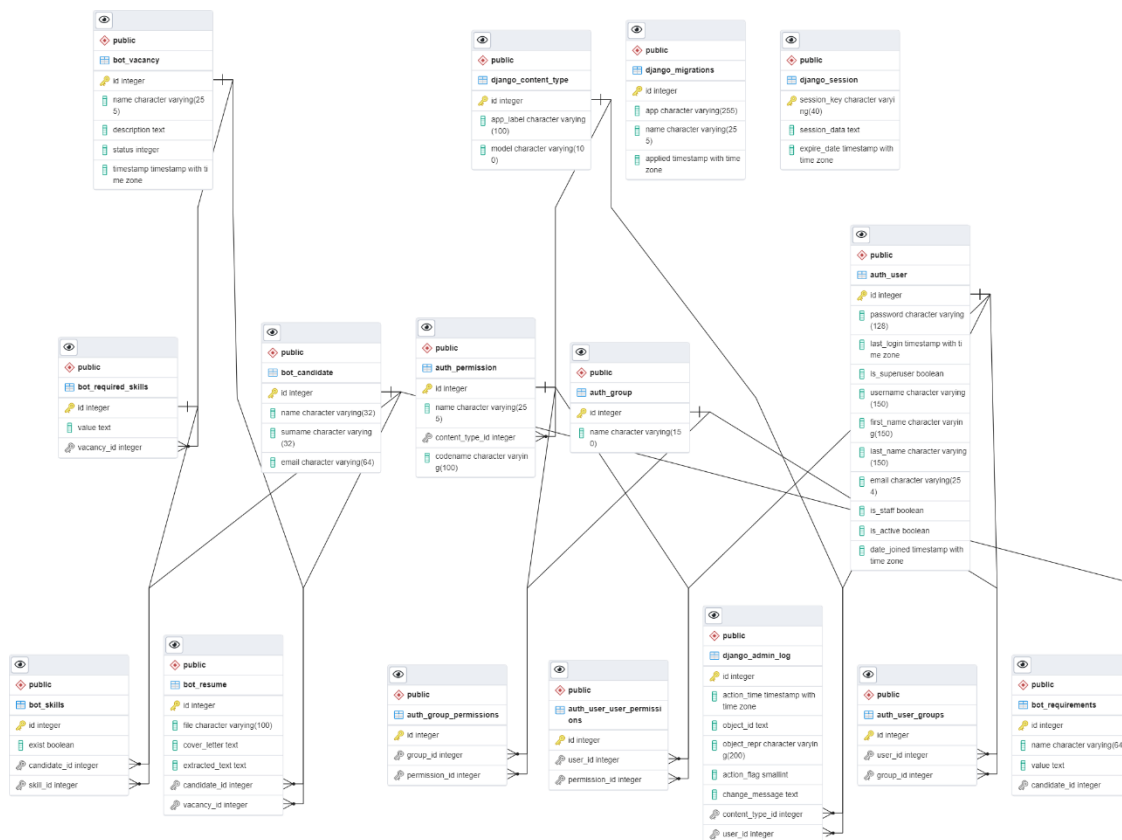


Рисунок 4.2 – Фізична реалізація бази даних інформаційної системи

4.3 Програмна реалізація

Інформаційна система підтримки роботи фахівця по роботі з персоналом складається з наступних модулів: Telegram-бот; веб сторінка; база даних.

Для реалізації інформаційної системи використовувалися наступні технологічні інструменти:

- Python (а саме бібліотека Django) – використовувався для реалізації серверної частини інформаційної системи;
- JavaScript – використовувалась для реалізації інтерактивних елементів клієнтської частини;
- HTML/CSS – використовувалися для реалізації статичних сторінок і шаблоні.

Реалізацію інформаційної системи можна поділити на три етапи.

До першого етапу слід віднести телеграм-бота. Телеграм-бот був реалізований завдяки Telegram API та серверної мови програмування Python [25].

До другого етапу слід віднести адмін-панель HR-спеціаліста. Вона була реалізована за допомогою HTML/CSS та Java Script, саме тому вона має простий та інтуїтивно зрозумілий дизайн.

Третій етап це панель адміністратора, оскільки вона була виконана за допомогою шаблону Django.

Адмін-панель

Адмін-панель призначена для контролю за роботою усієї системи. Вона реалізована, як веб-сторінка за допомогою, якої користувач має змогу додавати, змінювати та видаляти дані про кандидатів та вакантні посади (рис.4.3).

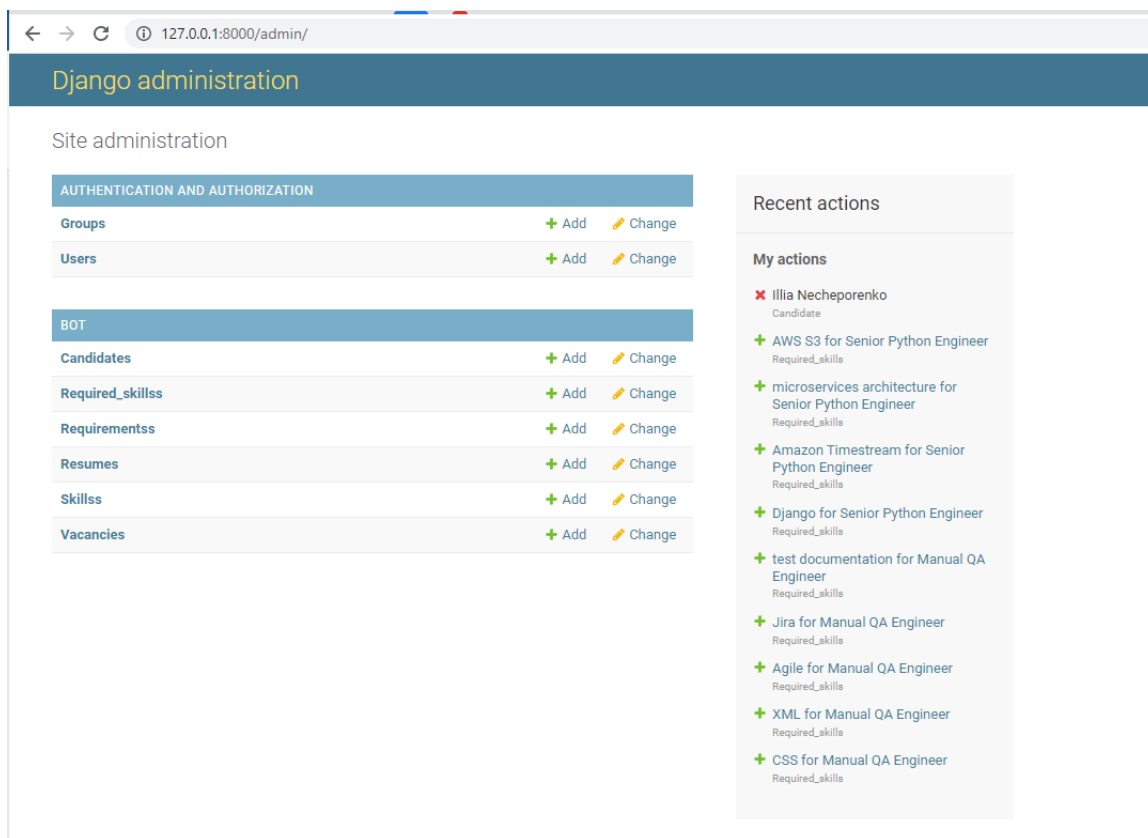


Рисунок 4.3 — Початковий екран адмін-панелі

Використовуючи адмін-панель HR-спеціаліст має можливість створювати, редагувати та видаляти актуальну інформацію про вакантні посади (рис 4.4 – 4.5).

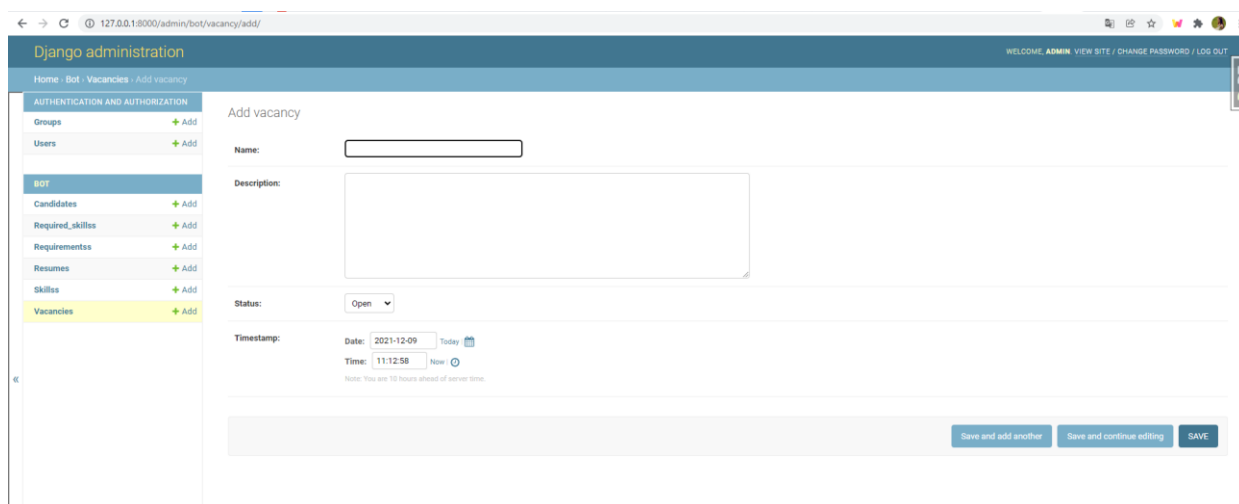


Рисунок 4.4 — Екран додавання нової вакантної посади

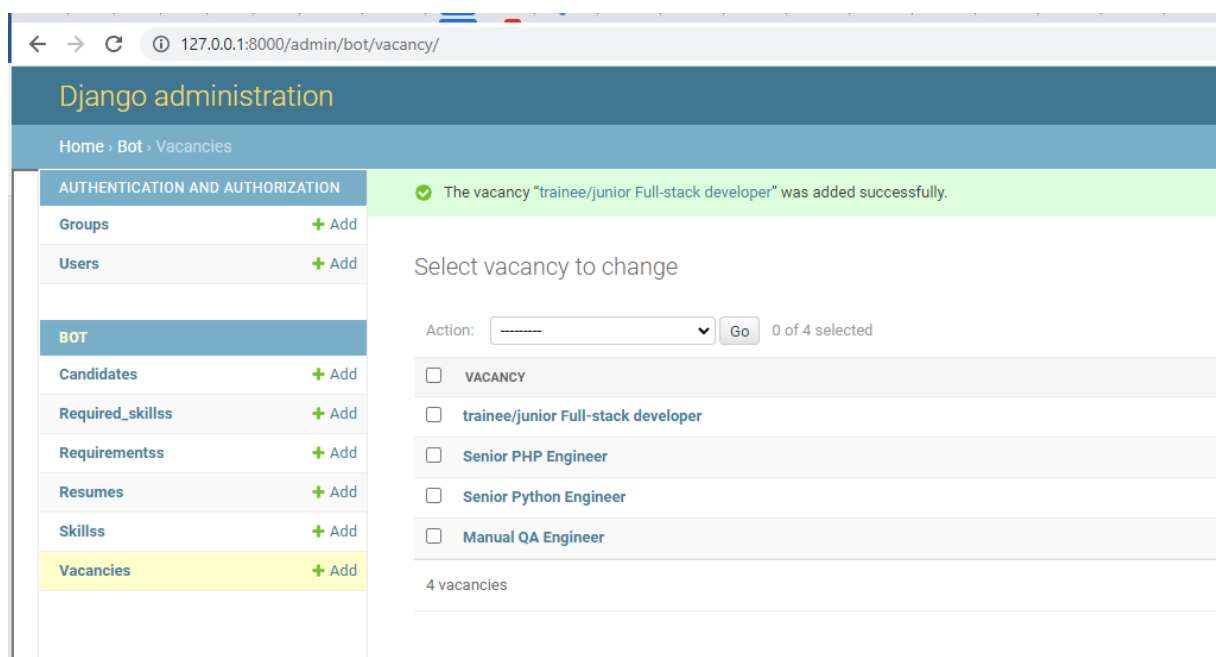


Рисунок 4.5 — Экран перегляду вакантних посад

Також HR-спеціаліст має змогу додати основні необхідні навички, це зроблено для того, щоб система мала змогу валідувати відправлені резюме на наявність необхідних навичок (рис.4.6).

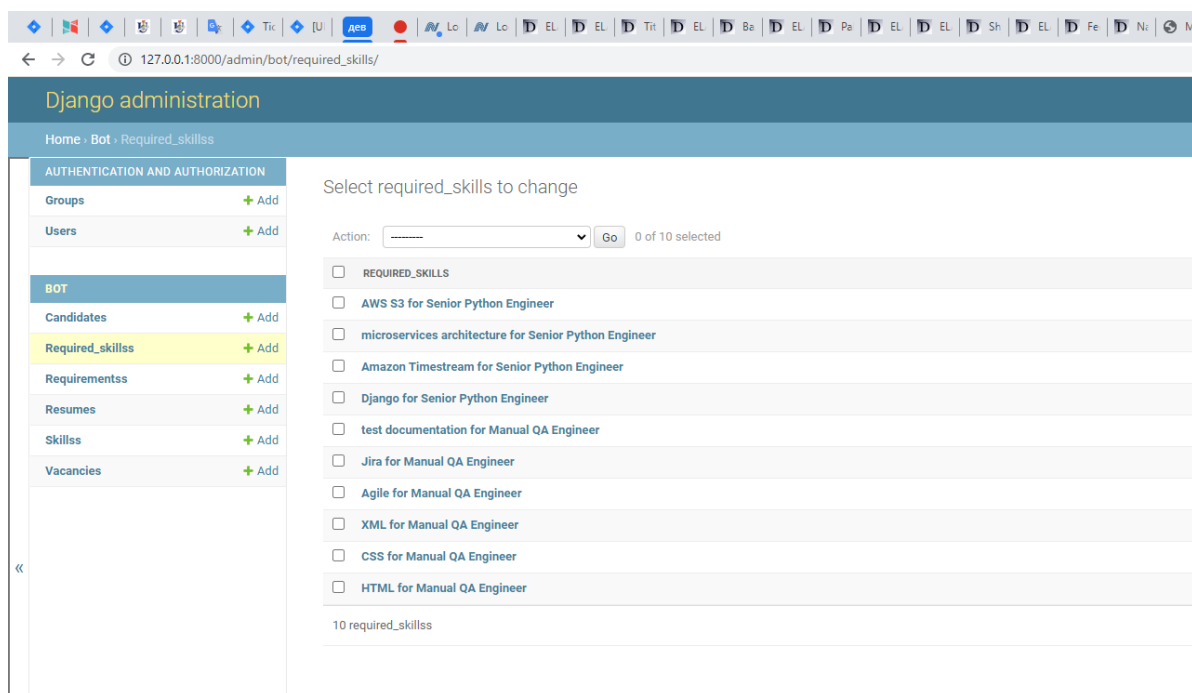


Рисунок 4.6 — Экран перегляду необхідних навичок

Всі інші таблиці система заповнює автоматично після того, як кандидат скористається телеграм-ботом.

При запуску програми з'являється початкове вікно, наведене на рисунку 4.7.

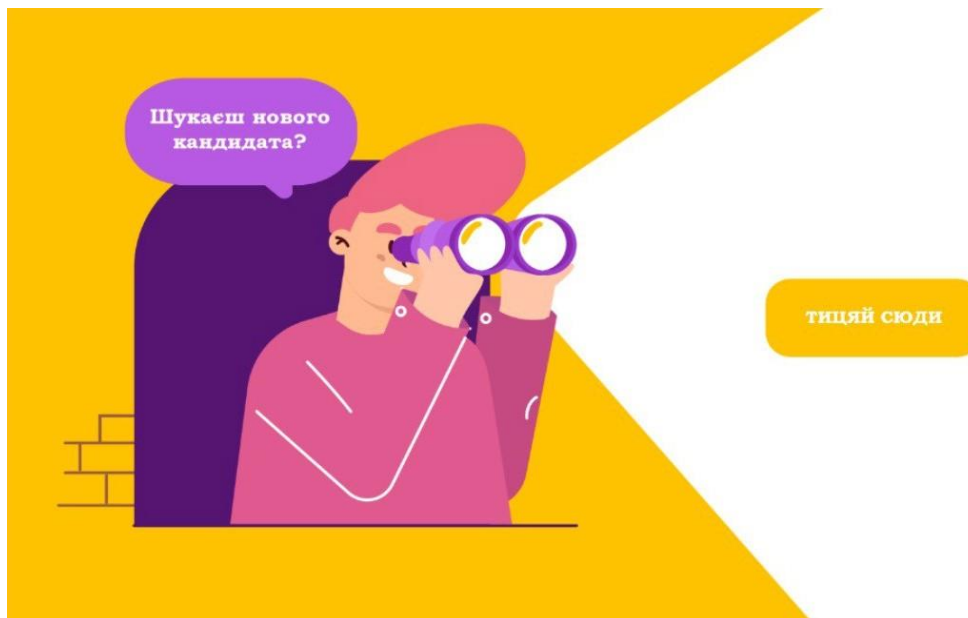


Рисунок 4.7 — Екран перегляду вітальної сторінки

Натиснувши на кнопку «Тицяй сюди» відкриється вікно авторизації, наведене на малюнку 4.8.

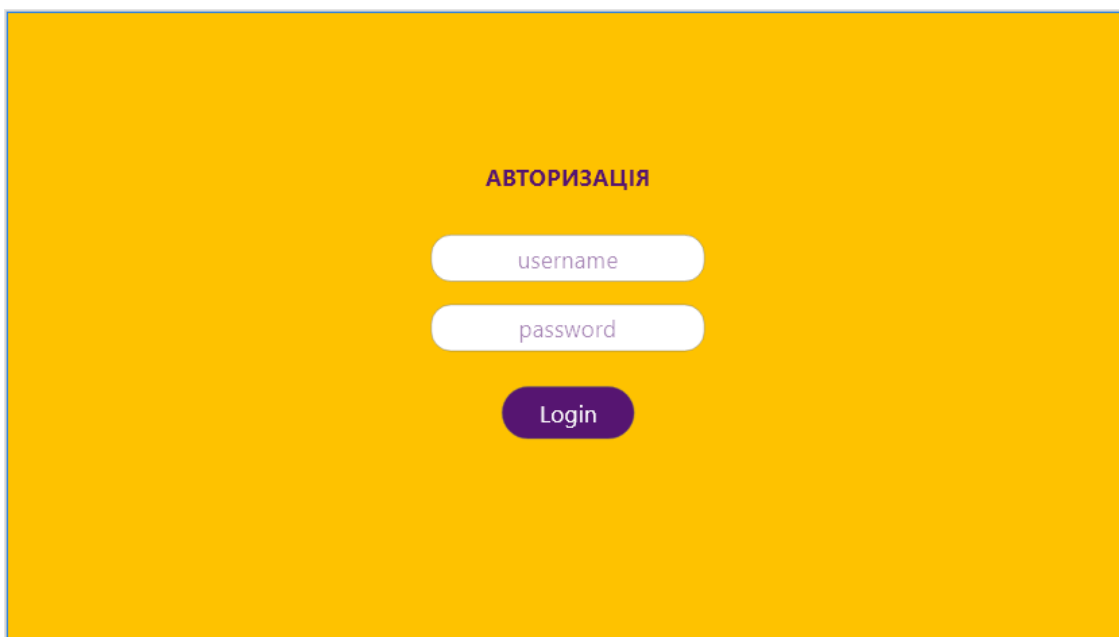


Рисунок 4.8 — Екран авторизації користувача

Після авторизації HR-спеціаліст потрапляє на головну сторінки системи, що надає змогу:

- переглядати список кандидатів, які відправили резюме до компанії;
- переглядати резюме конкретного кандидата;
- фільтрувати таблицю за вакантною посадою;
- переглядати по яким необхідним навичкам відбулося співпадіння з кандидатом;
- проводити розсилку повідомлень.

HR-спеціаліст може переглядати перелік кандидатів, які відправили своє резюме до компанії (рис. 4.9).

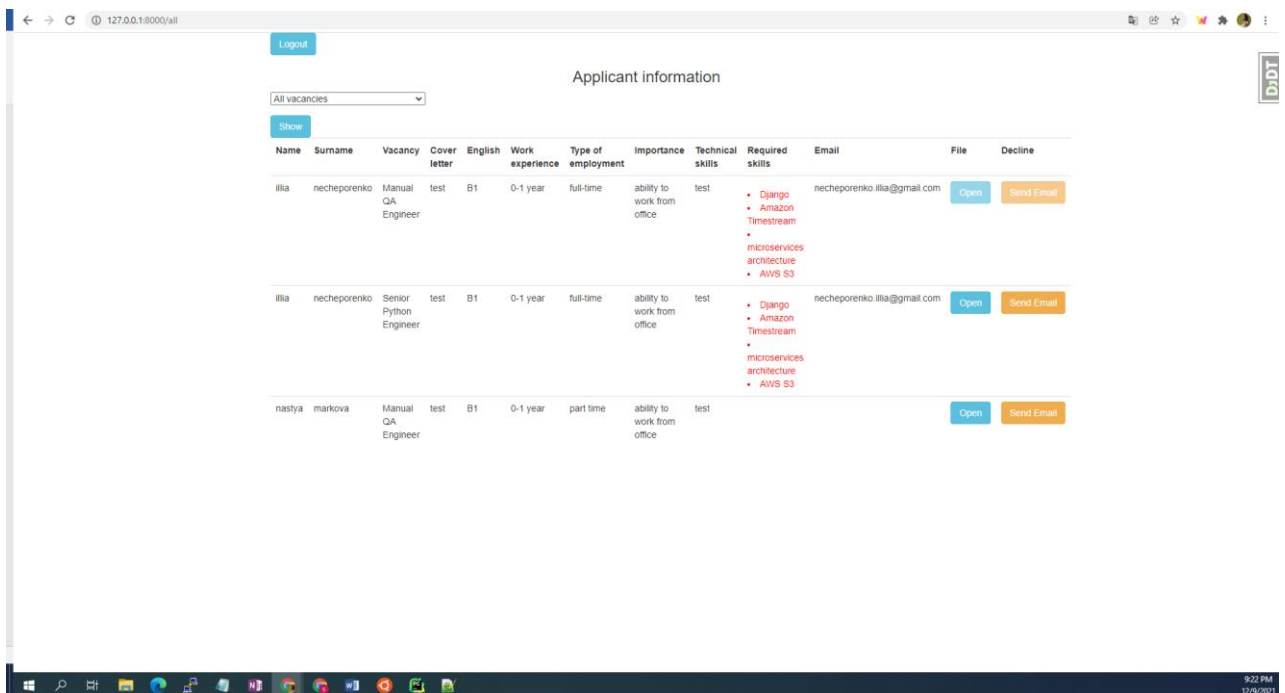


Рисунок 4.9 — Екран перегляду списку кандидатів

Також, HR-спеціалісту дається можливість проводити розсилку конкретному кандидату про відмову в подальшій співпраці з компанією, чи запрошення на співбесіду та пропозицію про подальшу співпрацю, натиснувши кнопку Send Email (рис. 4.10).

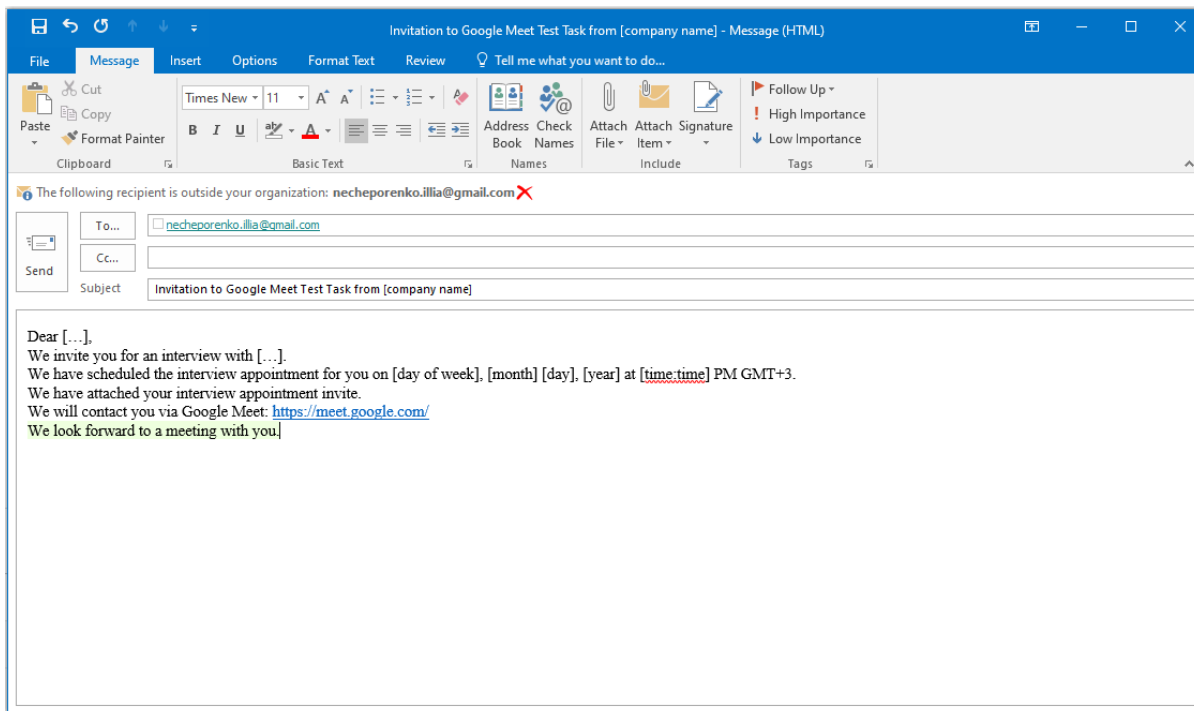


Рисунок 4.10 — Экран розсилки повідомлень

Користувач системи може відфільтрувати таблицю за конкретною позицією, обравши потрібну роль у полі над таблицею (рис. 4.11-4.12).

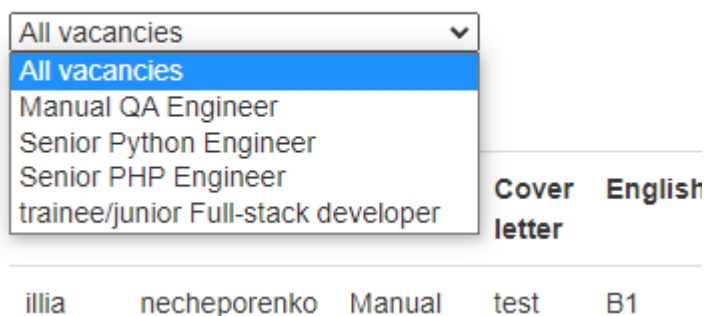


Рисунок 4.11 — Список наявних вакантних ролей

Applicant information

All vacancies ▼

Show

Name	Surname	Vacancy	Cover letter	English	Work experience	Type of employment	Importance	Technical skills	Required skills	Email	File	Decline
illia	necheporenko	Manual QA Engineer	test	B1	0-1 year	full-time	ability to work from office	test	<ul style="list-style-type: none"> • Django • Amazon Timestream • microservices architecture • AWS S3 	necheporenko.illia@gmail.com	Open	Send Email
nastya	markova	Manual QA Engineer	test	B1	0-1 year	part time	ability to work from office	test			Open	Send Email

Рисунок 4.12 — Відфільтрована таблиця по вакантній ролі «Manual QA Engineer»

Система порівнює необхідні навички до обраної кандидатом вакансії з навичками описаними кандидатом в резюме, і відображає в таблиці зеленим кольором ті навички, які співпали у кандидата з необхідними навичками, і червоним, яких навичок немає у кандидата (рис. 4.13).

Applicant information

All vacancies ▼

Show

Name	Surname	Vacancy	Cover letter	English	Work experience	Type of employment	Importance	Technical skills	Required skills	Email	File	Decline
illia	necheporenko	Manual QA Engineer	test	B1	0-1 year	full-time	ability to work from office	test	<ul style="list-style-type: none"> • Django • Amazon Timestream • microservices architecture • AWS S3 	necheporenko.illia@gmail.com	Open	Send Email
illia	necheporenko	Senior Python Engineer	test	B1	0-1 year	full-time	ability to work from office	test	<ul style="list-style-type: none"> • Django • Amazon Timestream • microservices architecture • AWS S3 	necheporenko.illia@gmail.com	Open	Send Email
Anastasia	Markova	Manual QA Engineer	test	A2	0-1 year	part time	ability to work from office	It's a perfect company for me. I think the company has a lot of benefits.	<ul style="list-style-type: none"> • HTML • CSS • XML • Agile • Jira • test documentation 	markovan97@gmail.com	Open	Send Email

Рисунок 4.13 — Навички кандидата, які співпали з необхідними навичками до вакансії, і які не співпали

Натиснувши на кнопку «Відкрити» система відкриває у сусідньому вікні PDF-файл резюме кандидата (рис.4.14).

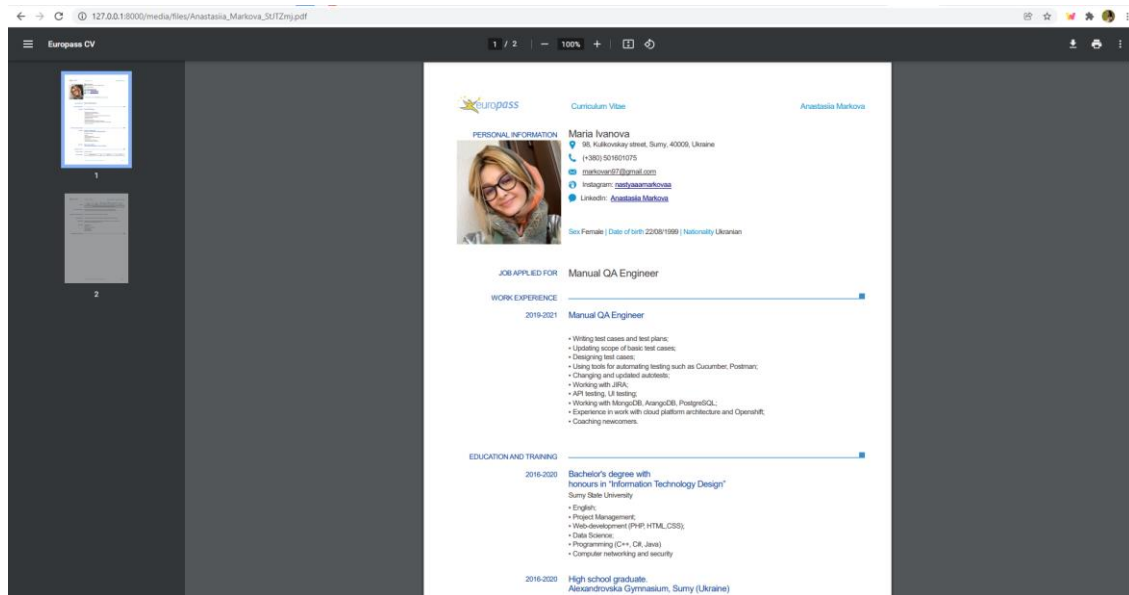


Рисунок 4.14 — Резюме кандидата

Далі розглянемо роботу телеграм-боту зі сторони кандидата.

На початку роботи з ботом кандидат натискає на кнопку Start і починає роботу з ботом. В першу чергу бот пропонує кандидату ввести своє ім'я та прізвище (рис.4.15).



Рисунок 4.15 — Початок роботи з ботом

Далі бот пропонує кандидату обрати потрібну йому вакансію, після того, як користувач обирає цікаву йому вакансію, бот відправляє опис обраної вакансії (рис.4.16).

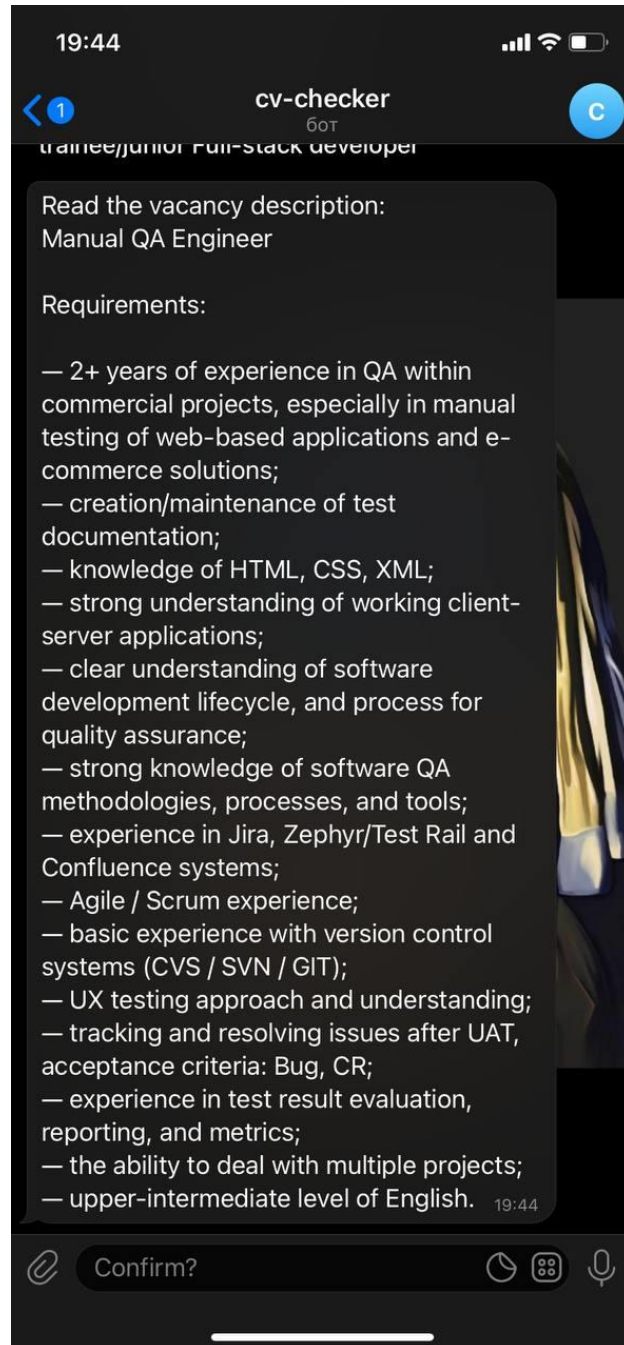


Рисунок 4.16 — Опис обраної вакансії, який надсилає бот

Після того, як користувач ознайомився з описом обраної вакансії, він може обрати нову вакансію, якщо він помилився і йому не підходить та, яку він обрав раніше, або продовжити роботу далі з обраною вакансією.

Після того, як користувач натиснув кнопку Confirm, користувачу треба пройти невелике опитування, яке виглядає для всіх кандидатів однаково (рис. 4.17).

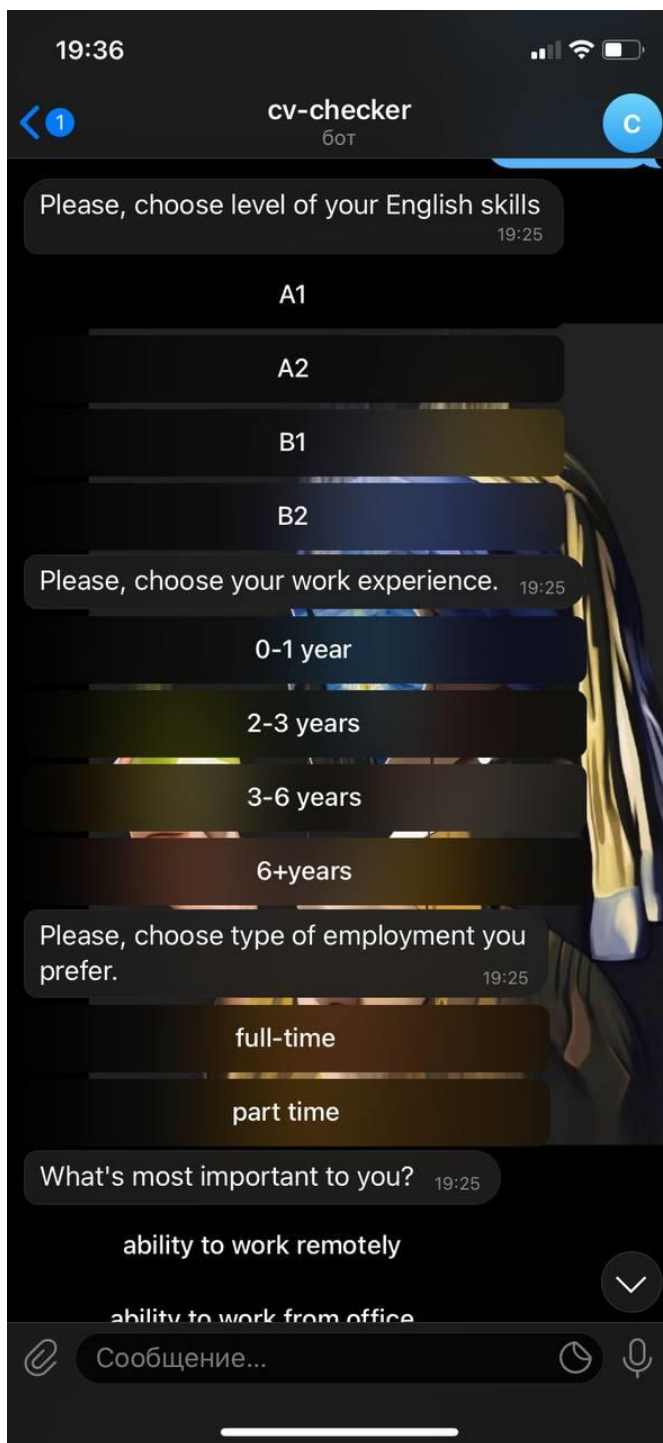


Рисунок 4.17 — Екран з опитуванням.

Після того, як кандидат пройшов опитування, система зберігає в базу відповіді, які відправлені кандидатом, і потім відображає їх в таблиці.

Далі кандидату пропонується відправити своє резюме.

Система має декілька етапів перевірки резюме на валідність.

Якщо кандидат відправляє резюме не з тим ім'ям та прізвищем, що було вказано на початку роботи з ботом, тоді бот повідомляє про це користувача (рис. 4.18).

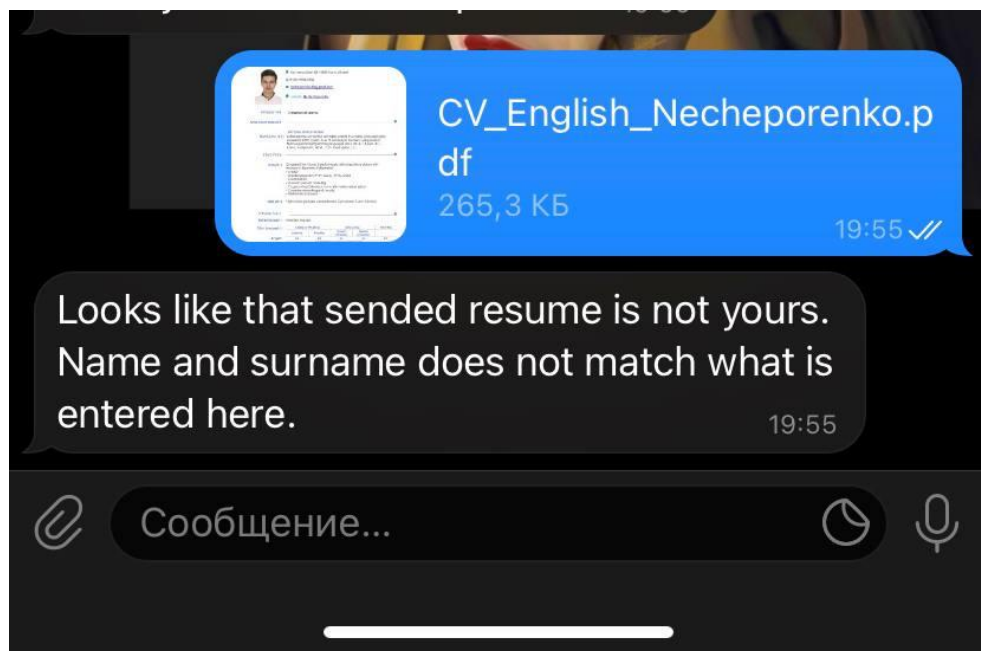


Рисунок 4.18 — Екран з повідомленням про некоректне ім'я та прізвище

Якщо кандидат відправляє резюме з неправильною адресою пошти, бот також повідомляє про це користувача (рис. 4.19).

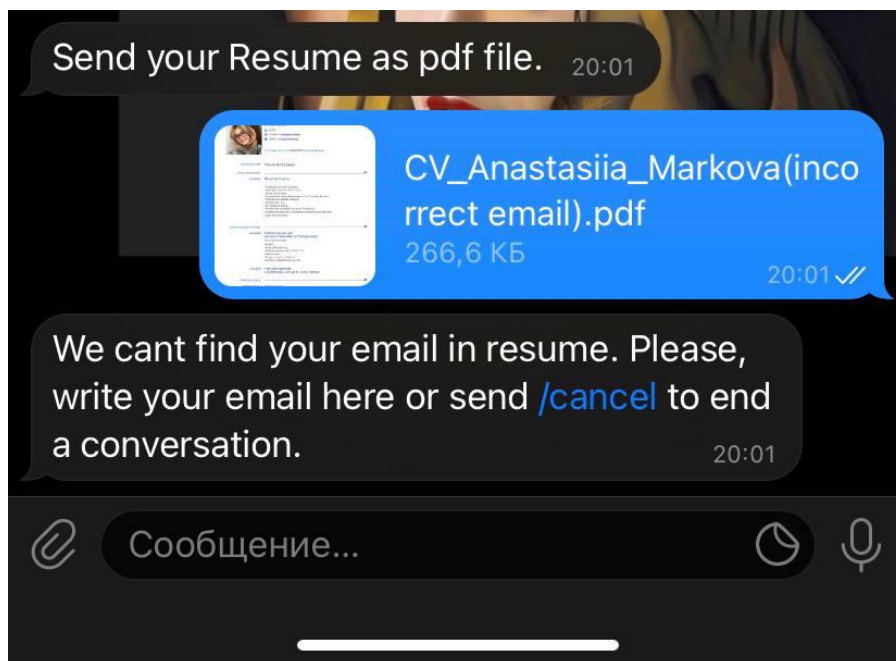


Рисунок 4.19 — Экран з повідомленням про некоректну адресу пошти

Якщо система не знаходить ніяких помилок у резюме кандидата, воно відправляє наступне повідомлення кандидату (рис 4.20).

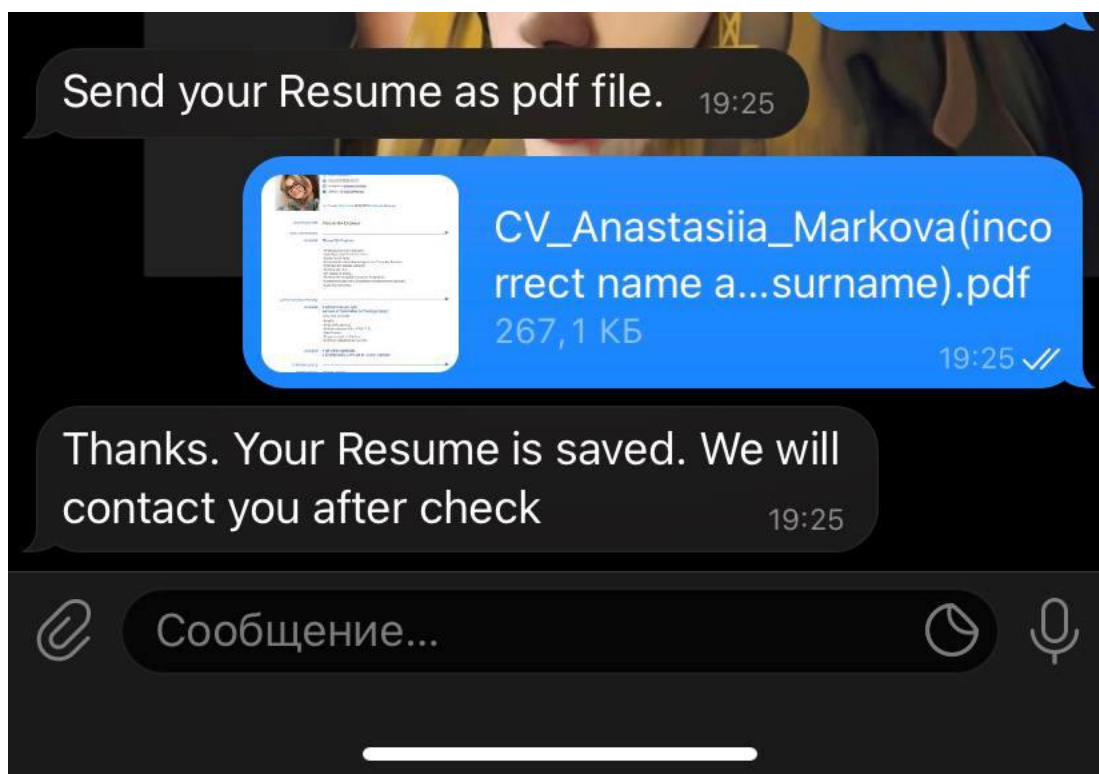


Рисунок 4.20 — Экран з повідомленням про успішну відправку резюме.

Також система перевіряє чи вже існує в системі відправлене резюме та надсилає повідомлення користувачу (рис. 4.21).

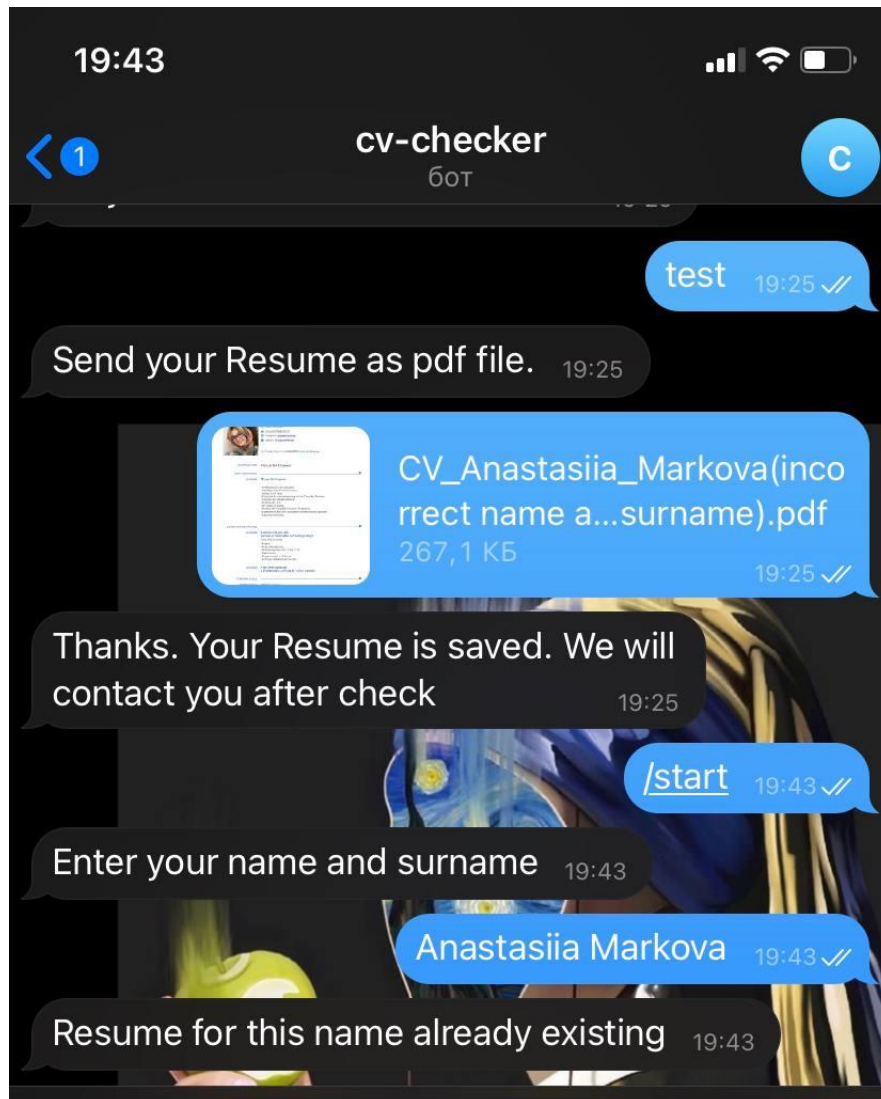


Рисунок 4.21 — Екран з повідомленням про те, що резюме с таким ім'я вже існує в базі

Раніше дані з резюме HR-фахівці вносили до картки кандидата вручну. На заповнення полів витрачалося багато часу. Хотілося автоматизувати цей процес.

Основна мета інформаційної системи була розробити парсер файли резюме. Для досягнення цілі були використані бібліотеки `pdfminer`, яка працює з PDF.

У цій роботі використовувався парсинг файлів резюме на регулярних виразах за допомогою сторонньої бібліотеки `pdfminer` [26].

Після завантаження кандидатами у HR-систему резюме у форматі PDF бібліотеки перетворюють ці файли на звичайний текст. Обробник через регулярні висловлювання витягує з тексту дані, розкидає їх у відповідні поля картки кандидата та записує до бази даних (рис. 4.22).

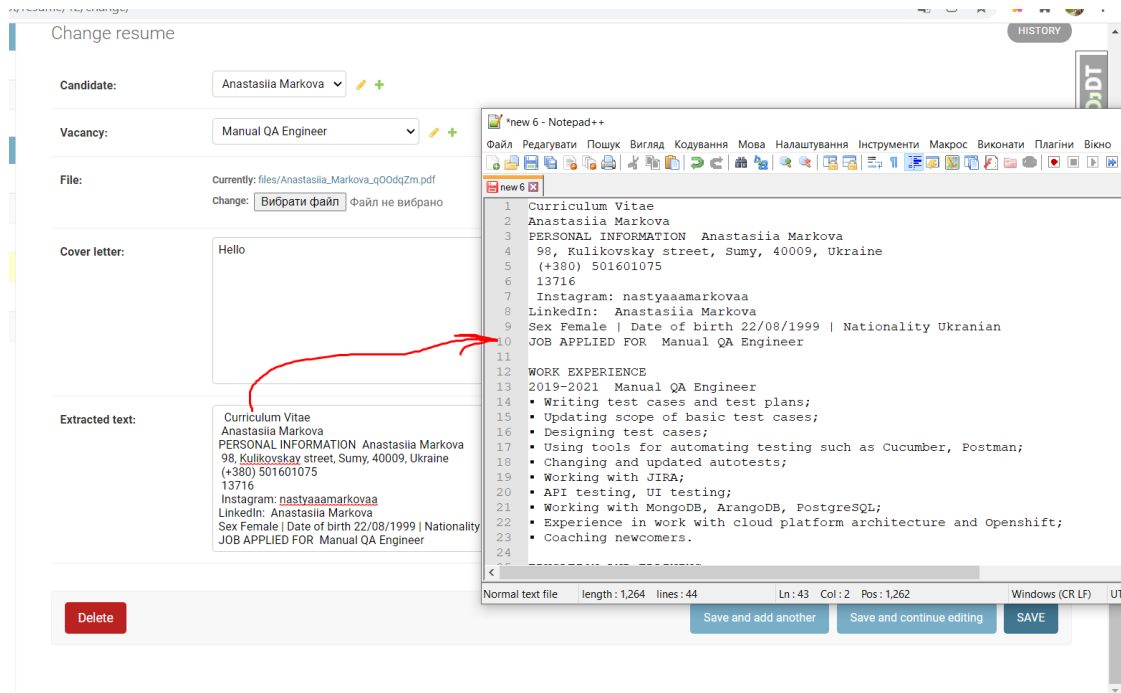


Рисунок 4.22 — Екран з розпарсеним резюме кандидата у базі даних

ВИСНОВКИ

Під час розробки магістерської роботи було досліджено та проаналізовано всі головні аспекти інформаційних потоків, які функціонують у сфері управління персоналом. Також, був проведений аналіз предметної області на основі літературних джерел, зазначених у роботі, визначено актуальність інформаційної системи, яка розроблялась. На основі добутих даних, було прийняте рішення про розробку інформаційної системи, через високу актуальність обраної теми на сьогоднішній день у вигляді телеграм боту та веб-сторінки.

Розроблена інформаційна задовольняє всі функціональні вимоги, які описані у завданні:

- перегляд інформації про вакансії;
- перегляд інформації про резюме;
- можливість завантаження файлу резюме на ПК;
- можливість обробки та фільтрації резюме;
- зворотній зв'язок з клієнтом;

Крім того, було проведено аналіз існуючих рішень, які схожі за функціональними вимогами до розроблюваної інформаційної системи. Після чого було проведено аналіз існуючих інструментів та технологій та обґрунтовано вибір найбільш оптимального рішення.

Для написання інформаційної системи були використані сучасні інструменти: мова програмування Python, веб-фреймворк Django. Базою даних у цьому сервісі обрано SQLite, що дозволило суттєво скоротити процес розробки та створити гнучку та зручну інформаційну систему управління персоналом, яка дозволяє менеджеру скоротити час на обробку даних та введення первинної інформації.

Як результат роботи було розроблено програмне забезпечення у вигляді телеграм-боту та веб-сторінки на прикладі задачі офісного менеджменту, а саме: пошук нових кандидатів на відкриту роль у компанії, обробка резюме, комунікація

з кандидатом, планування співбесід на основі результатів аналізу резюме кандидатів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Жуковська В. М. Цифрові технології в управлінні персоналом: сутність, тенденції, розвиток. Науковий вісник Міжнародного гуманітарного університету. 2017. № 2. С. 13–17
2. Кравчук О. І. Цифрова компетентність менеджера з персоналу. Соціально-трудові відносини: теорія та практика. 2018. № 1. С. 172–191
3. ДІДЖІТАЛІЗАЦІЯ HR-ПРОЦЕСІВ У СУЧАСНИХ РЕАЛІЯХ [Електронний ресурс] / Н.Данилевич, С. Рудакова, Л. Щетініна, Я. Касяненко // Галицький економічний вісник. – 2020. – Режим доступу до ресурсу: <https://galicianvisnyk.tntu.edu.ua/index.php?lang=eng>
4. Савченко И.П., Гранадская О.В. Инновации в системе управления персоналом компании // Научный журнал «АРМОМ. Серия: Гуманитарные науки», 2015. № 2. С. 35-41.
5. Кисляк Н. В. УПРАВЛЕНИЕ ПЕРСОНАЛОМ (С ПРИМЕНЕНИЕМ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ) / Н. В. Кисляк. – Екатеринбург, 2007. – 91 с.
6. Василів В. Б. ІНФОРМАЦІЙНІ СИСТЕМИ В УПРАВЛІННІ ПЕРСОНАЛОМ / В. Б. Василів. – Рівне: НУВГП, 2014. – 148 с.
7. Тарутин А.И., Поворина Е.В. Использование информационных технологий в управлении персоналом // Новая наука: Опыт, традиции, инновации, 2016. № 7 (111). С. 20-24.
8. Гвоздева В. А. Основы построения автоматизированных информационных систем [Текст] / В. А. Гвоздева — Москва: Форум, Инфра-М, 2016. — 320с.
9. 15 самых популярных языков программирования по версии GitHub [Електронний ресурс]. — Режим доступу до ресурсу: <https://habrahabr.ru/post/310262/>
10. Лобанович Б. Простой Python. Современный стиль программирования / Б. Лобанович. – Санкт-Петербург: Питер, 2016. – 480 с.

11. Дронов В. А. Django 2.1. Практика создания веб-сайтов на Python / В. А. Дронов. – Санкт-Петербург: БХВ-Петербург, 2019. – 673 с
12. Лутц М. Программирование на Python / М. Лутц. — СПб.: СимволПлюс, 2011. — 992 с.
13. Lundh F. Python Standard Library / F. Lundh. — O'Reilly & Associates, 2001. — 304 с.
14. Django documentation [Электронный ресурс]. – 2021. – Режим доступа до ресурсу: <https://docs.djangoproject.com/en/4.0/>.
15. Маккинли У. Python и анализ данных / У. Маккинли. — ДМК Пресс, 2015. — 482 с
16. Э. Фримен, Э. Фримен. Изучаем HTML, XHTML и CSS = Head First HTML with CSS & XHTML.—П.: «Питер», 2010.—656с.
17. PyCharm [Электронный ресурс] – Режим доступа до ресурсу: <https://uk.wikipedia.org/wiki/PyCharm>
18. Методология IDEF0 [Электронный ресурс]. – Режим доступа: <http://sewiki.ru/IDEF0> – 23.04.2020р. – Методология IDEF0.
19. Мінухін С. В. Методи і моделі проектування на основі сучасних CASE-засобів. Навчальний посібник / С.В. Мінухін, О.М. Беседовський, С. В. Знахур. – Харків: Вид. ХНЕУ, 2008. – 272 с. (укр. мов .)
20. Роб П. Системы баз данных: проектирование, реализация и управление / П. Роб, К. Коронел. – СПб.: БХВ-Петербург, 2004. – 1040 с.
21. Файли К. SQL: Руководство по изучению языка. / К. Файли. — М.: Peachpit Press, 2003. — 456 с
22. Рогозов Ю. И. Архитектура информационных систем / Ю. И. Рогозов, А. С. Свиридов, С. А. Кучеров. – Ростов-на-Дону: Южного федерального университета, 2014. – 122 с
23. Брауде Э. Дж. Технология разработки программного обеспечения. – СПб: Питер, 2004. – 655 с
24. Wesley J. Core Python Programming / J. Wesley. — Prentice Hall PTR, 2000. — 1136 с.

25. Telegram Bot API [Электронный ресурс] – Режим доступа: <https://core.telegram.org/bots/api>
26. PDF parser and analyzer [Электронный ресурс] // Python Software Foundatio. – 2019. – Режим доступа до ресурсу: <https://pypi.org/project/pdfminer/>. (

ДОДАТОК А

ПЛАНУВАННЯ РОБІТ

1.1 Ідентифікація мети проекту

В останні десятиліття інформацію розглядають як один з основних ресурсів розвитку суспільства, а інформаційні системи та технології – як засіб підвищення продуктивності праці та ефективності роботи персоналу.

Найбільш широко інформаційні системи та технології використовуються у виробничій, управлінській і фінансовій діяльності. Використання інформаційних систем і технологій у процесі управління підприємством робить його більш конкурентоспроможним, зокрема за рахунок об'єднання усіх структурних підрозділів в єдиний інформаційний простір.

Основною метою даної системи є автоматизація роботи HR-спеціаліста, яка дозволить скоротити часові витрати на пошук та підбір кандидатів, підвищить якість результатів цього процесу, сприятиме покращенню іміджу компанії, зміцнить її становище на ринку.

Дана інформаційна система надасть ряд переваг:

- швидкий пошук кандидатів;
- економія часу та підвищення ефективності;
- оптимізація витрат та ресурсів;
- наочність подання інформації.

1.2 Деталізація мети методом SMART

Метою проекту є створення інформаційної системи підтримки роботи фахівця по роботі з персоналом. Головна задача даної роботи полягає у розробленні інформаційної системи підтримки роботи фахівця по роботі з персоналом для автоматизації основних бізнес процесів, збору, обробки та зберігання відфільтрованої інформації про потенційного кандидата та більш зручною комунікацією з кандидатом на ту чи іншу посаду.

Результати деталізації мети проекту методом SMART розміщені у табл. 1.1.

Таблиця 1.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробити інформаційну систему підтримки роботи фахівця по роботі з персоналом.
Measurable (вимірювання)	Результатом роботи інформаційної системи підтримки роботи фахівця по роботі з персоналом є оцінка замовника.
Achievable (досяжна, узгоджена)	Ціль даного проекту вважається досяжною, оскільки розробник володіє необхідними навичками для створенні веб-сторінок засобами html, css, javascript. А також має достатньо досвіду з повнофункціональним серверним веб-фреймворком - django, написаним на python. Мету було узгоджено з вимогами та потребами замовника.
Relevant (реалістична)	Мети реально досягнути, так як розробка інформаційної системи за допомогою сучасних веб-технологій, не вимагає надзвичайно складних дій або велику кількість ресурсів.
Time-framed (обмежена в часі)	Обмеженість в часі зумовлена рішенням замовника, щоб отримати програмне забезпечення як можна швидше.

1.3 Планування змісту структури робіт ІТ-проекту

WBS – це графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов'язані з продуктом проекту. На верхньому першому рівні WBS фіксується продукт проекту – інформаційна система підтримки фахівця по роботі з персоналом. Наступний рівень відповідає основним діям проекту. Далі проводиться розбиття цих дій до елементарних робіт, тобто тих, які мають один чіткий результат з одним відповідальним та на яку можна обчислити витрати праці та тривалість виконання. Виконаємо декомпозицію робіт для даного проекту.

WBS-структура для даного проекту представлена на рис. А.1

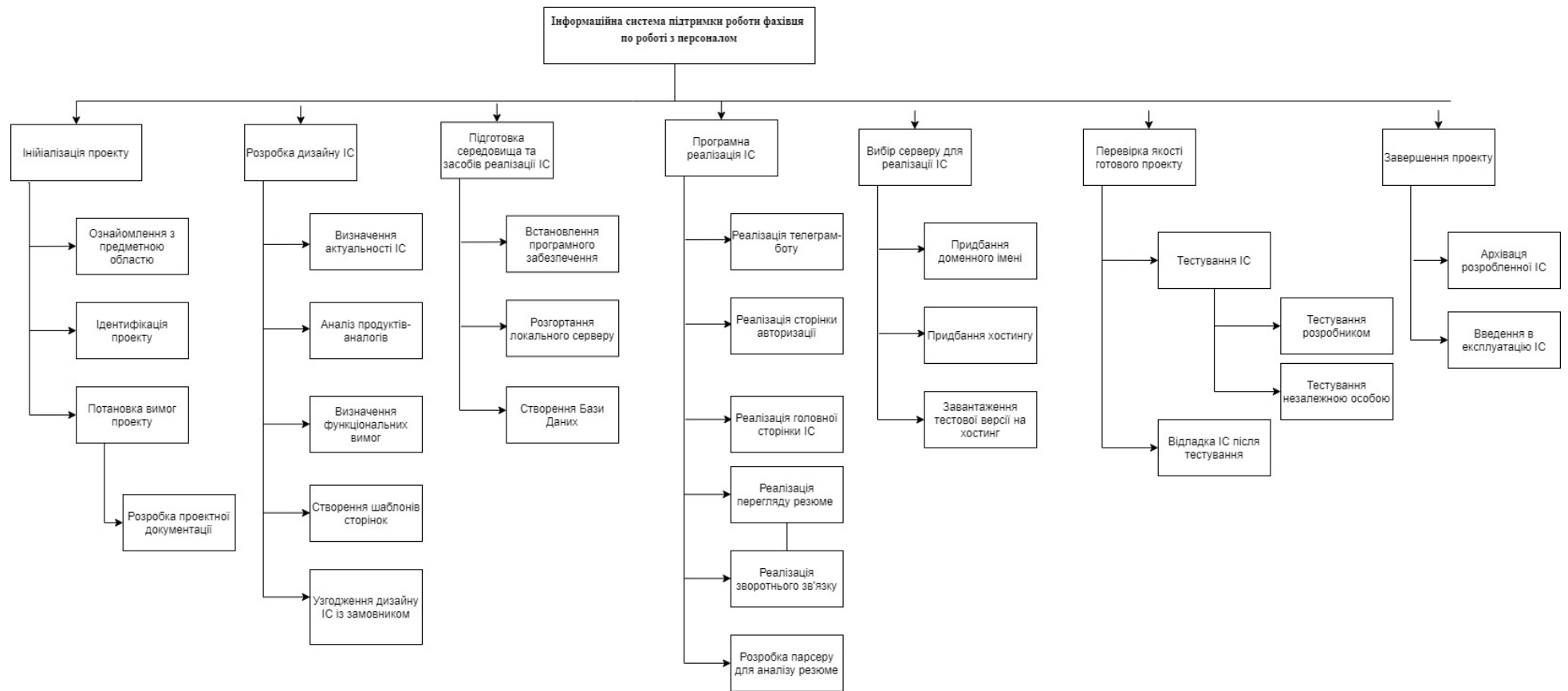


Рисунок А.1 – WBS-структура інформаційної системи підтримки роботи фахівця по роботі з персоналом.

Наступним кроком розробки структури проекту є визначення організаційної структури (OBS) проекту.

Організаційна структура проекту (OBS) – це графічне відображення учасників проекту, які відповідають та які відносяться до певної частини робіт. Визначається з елементарних робіт (робіт найнижчого рівня) кожної з гілок WBS. На верхньому рівні знаходиться керівник та команда управління проектом, на наступних – відповідальні особи.

OBS-структура для даного проекту представлена на рис. А.2.

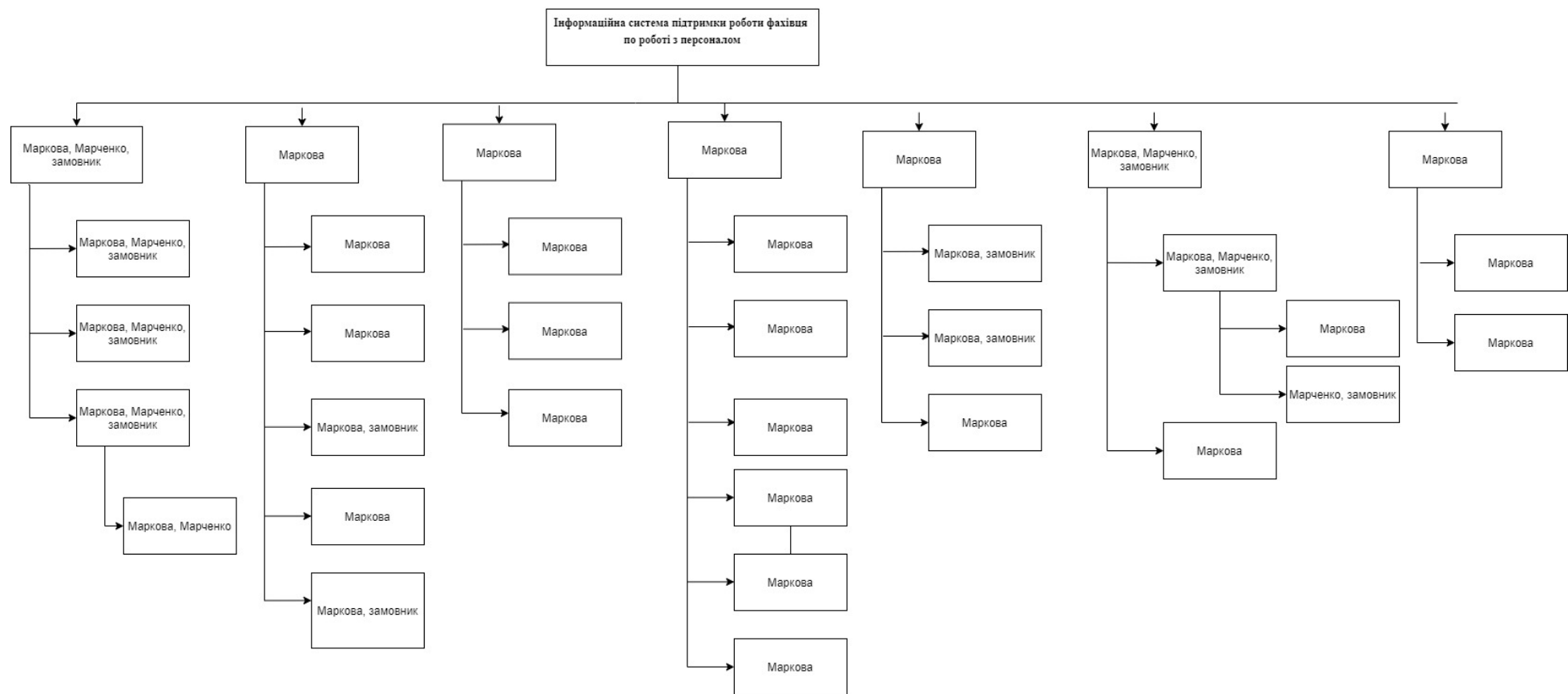


Рисунок А.2 – Організаційна структура виконавців

1.4 Побудова календарного графіку ІТ-проекту

Наступним кроком проектування розроблюваної ІС є побудова календарного графіку виконання ІТ – проекту (включаючи побудову часткових мережевих моделей у вигляді діаграм Ганта).

Діаграма Ганта – лінійна діаграма, яка показує задачі проекту, що представляються часовими відрізками, які характеризуються датами початку та закінчення робіт. Діаграма дозволяє відслідковувати відсоток робіт виконаних по кожному завданню. Цей графік представлено за допомогою програмного засобу MS Project.

Графік виконання дипломного проекту представлено у вигляді Діаграми Ганта на рис. А.3.

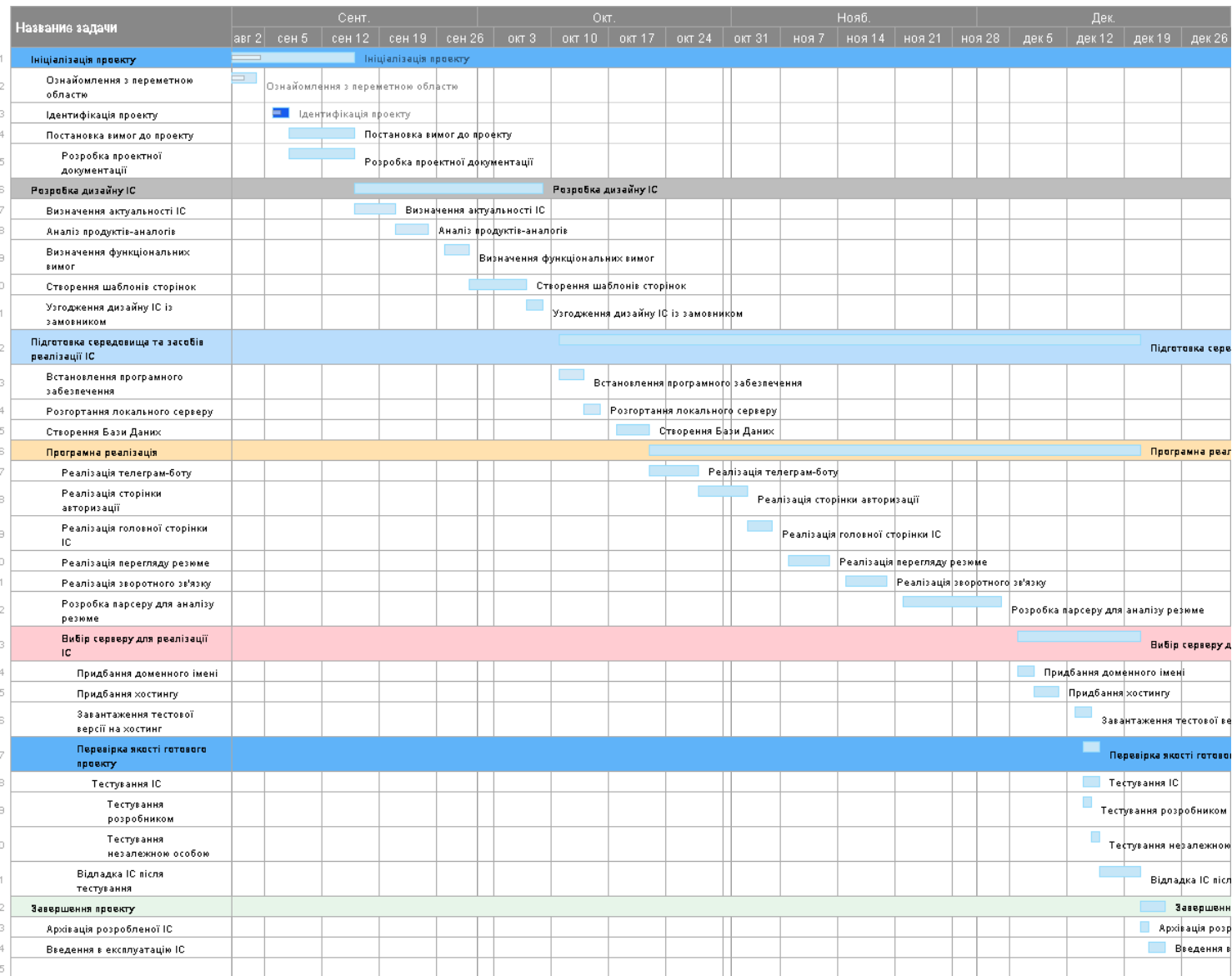


Рисунок А.3 – Діаграма Ганта

1.5 Планування ризиків проекту

Після визначення всіх робіт проекту, людей, які їх будуть виконувати та календарних планів проекту, необхідно продумати можливі ризики, які можуть виникнути в ході реалізації проекту.

Було виділено наступні ризики у даному проекті:

- R1 – зміна ТЗ на етапі розробки;
- R2 – недотримання календарного плану;
- R3 – некоректна робота програмного забезпечення;
- R4 – виникнення незапланованих робіт по проекту;
- R5 – хвороба розробника;
- R6 – неоптимізована робота БД;
- R7 – пошкодження файлу.

Далі шляхом експертної оцінки було визначено ймовірність виникнення усіх ризиків і побудовано відповідну таблицю (табл. 1.2).

Таблиця 1.2 – Шкала оцінювання ймовірності виникнення ризиків

Ймовірність виникнення	R1	R2	R3	R4	R5	R6	R7
Слабко ймовірний							
Малоймовірний							
Ймовірний							
Дуже ймовірний							
Майже можливий							

Далі було побудовано таблицю можливих втрат при виникненні ризиків, що наведені в табл. 1.3.

Таблиця 1.3 – Втрати при виникненні ризиків

Значимість втрат	R1	R2	R3	R4	R5	R6	R7
Мінімальна							
Низька							
Середня							
Висока							
Максимальна							

На основі цих даних була побудовано матрицю ймовірності втрат для оцінки ризиків даного проекту, що наведена в табл. 1.4.

Таблиця 1.4 – Матриця ймовірності-втрат

Ймовірність					
				R3	R7
				R4	
	R1	R5	R6		
			R2		
	Втрати				

Після виконання прогнозування виникнення ризиків та їх ступінь впливу на результат реалізації проекту, були розроблені варіанти запобігання та реакції на кожний із них.

Результати роботи представлені в таблиці 1.5.

Таблиця 1.5 – Оцінка рівня ризику

Ризики проекту	План запобігання ризику	План реакції на ризик
Зміна ТЗ на етапі розробки	Замовник повинен скласти детальне ТЗ, дотримуючись затвердженого плану: словник термінів, мета проекту, усі види вимог, терміни. Замовник та розробник повинні обговорити та затвердити його.	Уважно та чітко окреслити те, що треба змінити (після розмови із замовником) та зробити правки.
Недотримання календарного плану	1.Створення плану реалізації проекту на основі ретельного аналізу списку всіх робіт. 2.Затвердження зазначених термінів із замовником	Домовитися про умови зміни термінів із замовником. Якщо це недопустимо, тоді переорганізувати роботи таким чином, щоб в результаті терміни виконувалися.
Некоректна робота програмного забезпечення	1.Встановлення ліцензійного програмного забезпечення з перевірених джерел перед початком роботи. 2.Забезпечити наявність антивірусного програмного забезпечення.	Перезапуск або переустановлення програми, яка дала збій, відкат до останньої робочої версії.
Виникнення незапланованих робіт по проекту	При плануванні термінів залишити декілька резервних днів для таких випадків.	Передати повноваження робітнику робота якого менш пріоритетна.
Хвороба розробника	Виконувати певну частину роботи в команді для того, щоб члени проекту змогли замінити один одного при необхідності.	Передати повноваження робітника іншому члену команди, якщо цього вимагають терміни виконання