

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

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА
НА ТЕМУ**

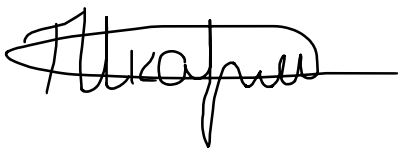
Система підтримки прийняття рішень дослідження
маніпулювання вибором у задачах багатокритеріальної
оптимізації

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Аналітика даних»**

Освітній рівень: бакалавр



Виконав: студент 4 курсу, групи АнД- 41

Кармелюк Т.І.

(прізвище та ініціали)

Керівник Гнатієнко Г. М.

(прізвище та ініціали)

К.Т.Н.

(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:

Зав. кафедри інтелектуальних технологій

Гларіонов О.Є.
(звання, прізвище та ініціали)

_____ (підпис)

«__»_____2022р.

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Кармелюку Тимофію Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема роботи

«СППР дослідження маніпулювання вибором у задачах багатокритеріальної оптимізації»
затверджена протоколом засідання кафедри від «23.12» 2021р. № 4

2. Термін здачі студентом закінченого роботи

31 травня 2022 року

3. Вихідні дані до роботи

Система підтримки прийняття рішень створюється для вибору кандидатів на посаду

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)

Вибір та аналіз предметної області, дослідження видів наукових подій та їх характеристик, формулювання та вибір задач, розробка архітектури додатку, візуалізація результатів, тестування роботи системи

5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)

Об'єкт, предмет, мета та завдання дослідження (1 слайд), проектування інформаційної системи (3 слайди), інформаційне забезпечення (1 слайд), структура програмного забезпечення (1 слайд), огляд процесу тестування (1 слайд), висновок (1 слайд)

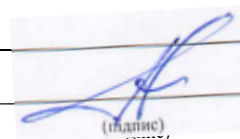
6. Консультанти з випускної кваліфікаційної роботи із зазначенням її розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання

15 лютого 2022 року

Керівник _____

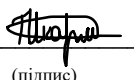


(підпис)

/ Гнатієнко Г.М. /

(ініціали та прізвище)

Завдання прийняв до виконання _____



(підпис)

/ Кармелюк Т.І. /

(ініціали та прізвище)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1.	Аналіз літературних джерел, аналіз існуючих методів, аналіз основних процесів предметного середовища, постановка задачі	15.02.2022 – 01.03.2022	Виконано
2.	Проектування web-додатку для визначення пріоритетності кандидатів	02.03.2022 – 05.04.2022	Виконано
3.	Розробка та тестування web-додатку для визначення пріоритетності кандидатів	06.04.2022 – 28.04.2022	Виконано
4.	Оформлення пояснювальної записки, підготовка презентації	29.04.2022 – 10.05.2022	Виконано

Студент _____

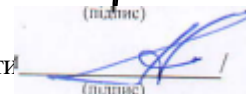


(підпис)

/ Кармелюк Т.І. /

(ініціали та прізвище)

Керівник випускної кваліфікаційної роботи _____



(підпис)

/ Гнатієнко Г.М. /

(ініціали та прізвище)

Анотація

Робота включає базові основи теорії прийняття рішень, зокрема моделі та методи багатокритеріального прийняття рішень, прийняття рішень в умовах невизначеності та конфлікту, прийняття рішень в умовах нечіткої інформації. Розглянуто проблеми маніпулювання вибором варіантів рішень у ситуаціях експертного оцінювання. Здійснено ознайомлення з основними результатами, теоретичними положеннями та основними застосуваннями моделей та методів прийняття рішень в умовах визначеності, невизначеності та ризику, в умовах конфлікту та нечіткої інформації. Наведено описання задачі експертного оцінювання у вигляді кортежу, на основі якого досліджено можливості маніпулювання вибором. Особливу увагу приділено задачам багатокритеріальної оптимізації (ЗБКО). Описано постановку ЗБКО з акцентом на її складові та евристики, які можуть бути використані для маніпулювання. Запропоновано класифікацію задач маніпулювання вибором в ситуаціях експертного оцінювання, які формалізуються у класі багатокритеріальних оптимізаційних задач.

Summary

The work includes the basic foundations of decision theory, in particular models and methods of multi-criteria decision-making, decision-making in conditions of uncertainty and conflict, decision-making in conditions of fuzzy information. The problems of manipulating the choice of solutions in situations of expert evaluation are considered. The main results, theoretical positions and main applications of models and methods of decision-making in the conditions of certainty, uncertainty and risk, in the conditions of conflict and vague information are acquainted. The description of the problem of expert evaluation in the form of a tuple is given, on the basis of which the possibilities of manipulating the choice are investigated. Special attention is paid to the problems of multicriteria optimization

(MIC). The statement of ZBKO is described with an emphasis on its components and heuristics that can be used for manipulation. The classification of problems of manipulation of choice in situations of expert estimation which are formalized in a class of multicriteria optimization problems is offered.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ФОРМУЛЮВАННЯ ТА СУТНІСТЬ ЗАДАЧ.....	10
1.1 Задачі експертного оцінювання.....	10
1.2 Постановка та особливості визначення розв’язків задачі багатокритеріальної оптимізації.....	13
1.3 Класифікація задач маніпулювання вибором при розв’язанні задач багатокритеріальної оптимізації.....	17
1.3.1 Задачі маніпулювання множиною альтернатив.....	17
1.4 Маніпулювання з обмеженнями ЗБКО та шкалами вимірювання.....	18
1.5 Маніпулювання шляхом впливу на експертну групу.....	20
1.6 Маніпулювання множиною цілей, що задекларовані при розв’язанні задачі.....	21
1.7 Маніпулювання системою переваг експертів.....	21
1.8 Висновки до першого розділу.....	22
РОЗДІЛ 2 РОЗРОБКА АРХІТЕКТУРИ ДОДАТКУ З ВИЗНАЧЕННЯ ПРІОРИТЕТНОСТІ КАНДИДАТА.....	23
2.1 Розробка архітектури.....	23
2.1.1 Функціональний аналіз.....	23
2.1.2 IDEF0 процесу видачі рекомендацій за допомогою інформаційної системи.....	27
2.1.3 Архітектура інформаційної системи.....	29
2.2 Інформаційне забезпечення рекомендаційної.....	29
2.2.1 Розробка інформаційного забезпечення.....	29
2.3 Висновки до другого розділу.....	33
РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ WEB – ДОДАТКУ ДЛЯ ВИЗНАЧЕННЯ ПРІОРИТЕТНОСТІ КАНДИДАТА.....	34
3.1 Обґрунтування вибору програмних засобів.....	34
3.2 Структура програмного забезпечення.....	34
3.3 Керівництво користувача.....	36
3.4 Огляд процесу тестування.....	48
3.5 Висновки до третього розділу.....	49
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	51
ДОДАТКИ.....	53

ВСТУП

У сучасному світі проблема маніпулювання вибором стає все більш актуальною. Адже задачі вибору виникають у найрізноманітніших сферах, а можливість вибору є основою демократії. В той же час відомо, що маніпулювання – це вид прихованого управління. Тому дослідження можливостей маніпулювання для різних моделей прийняття рішень на сьогодні є актуальним і перспективним напрямком досліджень. Вивчення проблем маніпулювання є тим більш важливим тому, що нерідко застосування шкал при вимірюванні деяких явищ перетворюється на тривіальне присвоєння цифр, а не на задеклароване дослідниками вимірювання суб'єктивних складових задачі прийняття рішень.

Сучасне виробництво являє собою складні комплексні підприємства, які обладнані потужною технікою. Планування їх діяльності й управління ними вимагають від керівника будь-якого рангу вміння швидко і правильно приймати різні рішення.

При цьому необхідно враховувати умови зростання обсягів виробництва, подальшого розвитку техніки, підвищення вимог до максимального використання надр і охорони довкілля, соціальні, економічні та інші фактори. У цих умовах рішення, прийняті на основі особистого досвіду та інженерної інтуїції, можуть виявитися малоефективними, оскільки в них складно врахувати цілий ряд протидійних умов.

Крім того, сучасне виробництво, як правило, потребує значних капітальних затрат, що різко підвищує збитки від помилок у прогнозуванні й управлінні. Така ситуація зумовлює застосування технології прийняття рішень, котра базується на кількісних оцінках варіантів, виключає або зменшує значення суб'єктивних факторів і при цьому враховує вплив різних неточно або невизначено описаних параметрів.

На практиці доволі часто постають багатоаспектні проблеми, якість розв'язання яких можна відобразити певною множиною критеріїв або ж пошук розв'язків яких покладено на групу експертів, котрі мають власні уявлення та фахові знання з конкретної проблемної області. Крім того, у деяких ситуаціях інтереси сторін, які мають синтезувати рішення, неузгоджені й, можливо, у чомусь або загалом протилежні, що ще більше ускладнює пошук задовільного рішення. Теорія прийняття рішень має теоретично формально обґрунтовувати й описувати такі ситуації, виявляти наявні парадокси та пропонувати можливі способи пошуку прийнятних рішень.

Маніпулювання в задачах вибору добре досліджене і для таких задач розроблена теорія агенди [1] – маніпулювання «порядком денним», послідовністю застосування процедур вибору. Відомо також [2], що правило вибору є захищеним від маніпулювання тоді і лише тоді, коли воно є диктаторським.

У цій роботі будемо розглядати задачі маніпулювання вибором, які виникають при формалізації практичних проблем у класі задач багатокритеріальної оптимізації.

Критеріальний підхід є розповсюдженим підходом при розв'язанні задач прийняття рішень. Цей підхід до описання задач прийняття рішень пов'язаний з припущенням, що кожен об'єкт можна оцінити конкретним числом, яке є значенням критерія, тому порівняння об'єктів зводиться до порівняння відповідних їм чисел.

У багатьох ситуаціях експертного оцінювання є доцільним задавати не одну скалярну оцінку якості об'єктів, а сукупність показників якості і розглядати задачу у векторній постановці. При цьому виникає задача багатокритеріальної оптимізації. Часто багатокритеріальність є способом збільшення адекватності описання цілі. В той же час, як буде показано нижче, багатокритеріальність відкриває широкі можливості для маніпулювання вибором.

Проблеми прийняття рішень і особи, які їх приймають, останнім часом заслуговують на все більшу увагу. Це зумовлено зростаючим динамізмом навколишнього середовища, збільшенням взаємозалежності багатьох рішень, стрімким темпом розвитку науково-технічного прогресу. Керівники, приймаючи рішення, стикаються зі складним вибором, з необхідністю розгляду множини альтернативних варіантів. Для оцінювання варіантів використовуються знання фахівців, складні аналітичні розрахунки, наукові дослідження, засоби сучасних інформаційних технологій.

РОЗДІЛ 1 ФОРМУЛЮВАННЯ ТА СУТНІСТЬ ЗАДАЧ

1.1 Задачі експертного оцінювання

Залежно від мети експертного оцінювання й методу обрахунку експертних оцінок виникають наступні основні задачі: побудова узагальненої оцінки понять й об'єктів на основі індивідуальних оцінок експертів, побудова узагальненої оцінки на основі парного порівняння об'єктів кожним з експертів, визначення відносних ваг взаємозв'язку об'єктів, визначення залежностей між ран жировками, визначення узгодженості думок експертів, оцінка надійності обробки результатів.

При вирішенні багатьох задач недостатньо упорядкування об'єктів по одному або по групі показників. Необхідно мати числові значення для кожного об'єкта, що визначають його перевагу перед іншими об'єктами. Наявність таких оцінок дозволить визначити узагальнену оцінку для всієї групи експертів. Визначення узгодженості думок експертів провадиться шляхом обчислення числової міри, що характеризує ступінь близькості індивідуальних думок. Аналіз значення міри узгодженості сприяє отриманню правильного судження про загальний рівень знань з розв'язуваної проблеми й виявленню групувань думок експертів. Обробка експертних оцінок дозволяє розкрити зв'язані показники порівняння й здійснити групування по ступеню зв'язку.

Для описання задач експертного оцінювання використовується її символічне представлення у вигляді кортежу [3] $\langle A, S, R, E, C, P \rangle$, де A – множина об'єктів (варіантів, альтернатив, наборів параметрів), S – множина обмежень, R – множина критеріїв, шкал вимірювання по критеріях, згорток критеріїв, відображення множини допустимих альтернатив у множину критеріальних оцінок, E – множина формальних характеристик експертів, C – множина цілей, що стоять перед дослідниками, P – система переваг вирішуючого елемента: особи, що приймає рішення (ОПР), або групи експертів.

Множина A , що складається з n об'єктів, які порівнюються між собою:

$$a_i \in A, i \in \{1, \dots, n\} = I.$$

Об'єкти описуються множиною m параметрів, тобто кожен об'єкт є точкою деякого параметричного простору Ω^m

$$a_i = (a_i^1, \dots, a_i^m), a_i \in A, i \in I, A \subset \Omega^m,$$

де $a_i^j, i \in I, j \in \{1, \dots, m\} = J$, – значення j -го параметра i -го об'єкта.

З множини A ОПР вибирається найкращий за встановленими критеріями об'єкт та обґрунтовується його вибір. На етапі обґрунтування підходів до вибору та власне самої процедури вибору виникають можливості для маніпулювання результатами розв'язання задачі.

Для опису методів та алгоритмів розв'язання ЗЕО наведемо основні поняття, які будуть використовуватися у подальшому викладенні. Під ЗЕО розуміють пару $\langle C, A \rangle$, де C – принцип оптимальності, A – задана множина об'єктів. Розв'язком ЗЕО $\langle C, A \rangle$ є підмножина об'єктів $A_0 \subset A$, одержана з використанням принципу оптимальності: $A_0 = C(A)$. В теорії дослідження операцій [4] пара $\langle C, A \rangle$ називається:

- задачею прийняття рішень, якщо C та A не є апріорно заданими і можуть варіюватися;
- задачею вибору, якщо множина об'єктів A є заданою, а принцип оптимальності C – таким, що варіюється;
- загальною задачею оптимізації, коли C та A є заданими.

Відповідно, можливість для маніпулювання виникає саме у тих ситуаціях, де допускаються варіації принципу оптимальності або множини об'єктів.

Варіаційне числення являє собою розділ математичного аналізу, пов'язаний з проблемами оптимізації при умовах більш загального характеру, ніж ті, які розглядаються в звичайній теорії знаходження максимумів і мінімумів деяких функцій. Воно має справу з максимізацією або мінімізацією

функціоналів, коли потрібно визначити не окремі значення функції, а всю функцію загалом.

Але зосередимося на останній ситуації – задачах оптимізації, досліджуючи проблематику задач багатокритеріальної оптимізації.

Для рішення задачі оптимізації насамперед необхідно визначити цільову або вартісну функцію процесу, що оптимізується. При цьому потрібно дати відповідне формулювання задачі в фізичній формі і здійснити переклад фізичного опису на мову математики.

Для здійснення ефективного управління процесом необхідно знати його поточний стан. Будемо називати це - задачею оцінки стану. Крім того, необхідно охарактеризувати процес за допомогою адекватної моделі, що залежить від різних чинників. Це називається ідентифікацією системи.

При умові знання функції вартості, стану і параметрів системи можна потім визначити найкраще управління, що мінімізує (або максимізує) функцію вартості. Це може бути задача із замкненим або розімкненим контуром в залежності від того, чи є управління функцією стану системи. Одна з фундаментальних задач оптимального управління виникає в тому випадку, коли потрібно отримати замкнений контур управління для лінійної стохастичної системи з спотвореними шумом спостереженнями станів системи.

1.2 Постановка та особливості визначення розв'язків задачі багатокритеріальної оптимізації

Задача багатокритеріальної оптимізації полягає у пошуку вектора цільових змінних, який задовільняє накладеним обмеженням та оптимізує векторну функцію, елементи якої відповідають цільовим функціям. Ці функції утворюють математичне описання критерію задовільності та, зазвичай, взаємно конфліктують.

Задача багатокритеріальної оптимізації формалізується у такій постановці [3, 5]:

$$y_i = f_i(x) \rightarrow \max, \quad i \in L_1, \quad (1)$$

$$y_i = f_i(x) \rightarrow \min, \quad i \in L_2, \quad (2)$$

$$x \in A, \quad A \subseteq E^m, \quad (3)$$

де A – множина об'єктів, які характеризуються m параметрами, тобто належать простору E^m ; $y(x) = (f_1(x), \dots, f_k(x))$ – вектор оцінок об'єктів або критеріїв, який задається відображенням $f: A \rightarrow E^k$, $L = \{1, \dots, k\}$ – множина індексів критеріїв, $L_1 = \{1, \dots, k_1\}$, $L_2 = \{k_1 + 1, \dots, k\}$ – множини індексів критеріальних функцій, які, відповідно, максимізуються та мінімізуються, $L_1 \cup L_2 = L$. У деяких задачах множина об'єктів виділяється із ширшої множини $A \subseteq D \subset E^m$ за допомогою обмежень, які найчастіше задаються системою нерівностей.

Вектори $y = (y_1, \dots, y_k)$ та $y' = (y'_1, \dots, y'_k)$, $y \neq y'$, знаходяться у відношенні " $>$ ", якщо $y_i \geq y'_i$, $\forall i \in L$, і хоча б одна нерівність є строгою.

Оцінка об'єкта $y^0 = f(a_0)$, $y^0 = (y_1^0, \dots, y_k^0)$, називається ефективною (оптимальною за Парето, Парето-оптимальною, непокрещуваною, мажорантною, недомінованою), за відношенням " $>$ ", якщо не існує іншої оцінки y , яка б строго переважала y^0 , тобто виконувалося б відношення $y > y^0$.

Відомо [5], що два ефективних об'єкти є або еквівалентними, або непорівняними між собою за множиною критеріїв.

Проблема визначення області Парето є строго об'єктивною і вирішується без застосування будь-яких евристик [5]. Але область компромісів є множиною точок, з яких у більшості випадків треба вибрати одну. Звуження області ефективних об'єктів і, тим більше, вибір єдиного з них принципово потребує застосування додаткової інформації від експертів, оскільки ефективні набори параметрів не можна порівняти один з одним формально.

Оцінки об'єктів мають різну розмірність, оскільки характеризують різні фізичні властивості об'єктів, тому доцільно розглядати не абсолютні значення оцінок об'єктів, а їх безрозмірні значення. Монотонні перетворення, які переводять оцінки об'єктів до безрозмірного виду, наведено у першому розділі, в якому вони представлені формулами...

Для описання одного із способів розв'язання ЗБКО, вводяться евристики.

Евристика E1. Вибір виду монотонного перетворення $\omega_i(a)$, $i \in L$, $a \in A$, для переведення значень параметрів об'єктів до безрозмірного виду здійснюється експертом за формулами, які мають задовольняти таким вимогам:

- враховувати необхідність мінімізації відхилень від оптимальних значень по кожній цільовій функції (ЦФ);
- мати загальний початок відліку і однаковий порядок зміни значень на усій множині об'єктів;
- зберігати відношення переваги на множині об'єктів, які порівнюються, за сукупністю ЦФ, і, таким чином, не змінювати множину ефективних об'єктів.

Розв'язок ЗБКО може виявитися не оптимальним для жодної ЦФ виду (1), (2), але одночасно бути у певному розумінні найкращим розв'язком для усіх критеріальних функцій одночасно.

Евристика E2. Найкращим об'єктом при розв'язанні ЗБКО слід вважати такий об'єкт, для якого відхилення від найкращих значень по кожній оцінці є мінімальними.

Якщо найменші значення відхилень по кожному критерію не досягаються одночасно на жодному об'єкті, то виникає необхідність порівнювати ці відхилення між собою, що пов'язано з необхідністю залучення додаткових евристик від експертів. Для конструктивного визначення найкращого об'єкта наведемо ще одну теорему.

Відомо, що для кожного об'єкта $a \in A$, такого, що у просторі перетворених значень ЦФ справедливі нерівності $0 < \omega_i(a) < 1, \forall i \in L$, існує вектор $\rho = (\rho_i, i \in L)$, який задовольняє співвідношенням нормованості:

$$\rho_i > 0, i \in L, \quad (4)$$

$$\sum_{i \in L} \rho_i = 1, \quad (5)$$

і число k_0 , такі, що об'єкт $a \in A$, задовольняє одночасно k рівностям

$$\rho_i \omega_i(a) = k_0, \forall i \in L. \quad (6)$$

Введемо ще три евристики.

Евристика E3. Вектор вагових коефіцієнтів ЦФ $\rho = (\rho_i, i \in L)$, який задовольняє співвідношенням (4), (5), будемо інтерпретувати як відношення переваги у кількісному вигляді на множині ЦФ.

Евристика E4. Під розв'язком ЗБКО для заданого вектора вагових коефіцієнтів $\rho = (\rho_i, i \in L)$, будемо розуміти такий компромісний об'єкт, який належить множині ефективних об'єктів і знаходиться на напрямку, який визначається вектором $\rho = (\rho_i, i \in L)$.

Евристика E5. Компромісний розв'язок $a_0 \in A$ ЗБКО має забезпечувати однакові мінімальні зважені відносні відхилення $\rho_i \omega_i(a_0), i \in L$, за всіма критеріями одночасно.

Доведено [5], що якщо $a_0 \in A$ є ефективним об'єктом для вектора $\rho = (\rho_i, i \in L)$, то цьому об'єкту відповідає найменше значення параметра k_0 , при якому система (6) виконується одночасно для усіх ЦФ.

Відомо також [5], що для того, щоб об'єкт $a^* \in A$, такий, що $\omega_i(a^*) > 0, \forall i \in L$, був ефективним при заданому векторі вагових коефіцієнтів $\rho = (\rho_i, i \in L)$, достатньо, щоб $a^* \in A$ був єдиним розв'язком системи нерівностей

$$\rho_i \omega_i(a) \leq k_0, \quad \forall i \in L, \quad (7)$$

для мінімального значення параметра k_0^* , при якому ця система є сумісною.

Таким чином, компромісний розв'язок ЗБКО, визначений евристикою Е5, може бути знайдений як єдиний розв'язок системи нерівностей виду (7) для мінімального значення параметра k_0 , при якому ця система ще сумісна. У просторі відносних значень параметрів об'єктів компромісному об'єкту відповідає точка перетину променя, направляючі косинуси якого визначаються заданим вектором відносної важливості ЦФ $\rho = (\rho_i, i \in L)$, з множиною ефективних об'єктів. Якщо такої точки не існує, тобто на промені, що визначається вектором $\rho = (\rho_i, i \in L)$, не лежить жодна точка, яка відповідає ефективному об'єкту, то компромісним об'єктом вважається той, для якого виконується система нерівностей (7) і цьому об'єкту відповідає точка, найближча до заданого променя. Якщо компромісний об'єкт не єдиний, тобто існує деяка підмножина ефективних об'єктів, еквівалентних з точністю до деякого достатньо малого числа за значенням параметра k_0 , то вибір компромісного об'єкта здійснюється з допомогою іншого критерію.

Зазначимо, що альтернативами в ЗБКО (1)-(3) можуть бути, зокрема, ранжування [3] або об'єкти чи явища іншої природи.

1.3 Класифікація задач маніпулювання вибором при розв'язанні задач багатокритеріальної оптимізації

Відповідно до загального опису задач експертного оцінювання та описаного вище підходу до розв'язання ЗБКО, наведемо розроблену автором класифікацію задач маніпулювання вибором у задачах багатокритеріальної оптимізації.

1.3.1 Задачі маніпулювання множиною альтернатив

Множина альтернатив може бути змінена різними способами:

- волюнтаристським підходом ОПР;
- імітацією демократичних процедур, які в результаті призводять до зміни початкової множини A ;
- введенням додаткових обмежень та умов участі альтернатив у подальшому розв'язанні багатокритеріальної задачі;
- свідомим ігноруванням деяких обмежень з метою розширення початкової множини альтернатив;
- іншими підходами, які призводять до зміни початкової множини альтернатив з метою маніпулювання та «обґрунтованого» вибору потрібної для ОПР альтернативи.

1. Доповнення початкової множини альтернатив A деякою підмножиною A^* , $A^* \cap A = \emptyset$, яка містить альтернативу $a^* \in A^*$, що має бути вибраною як розв'язок у результаті розв'язання ЗБКО виду (1)-(3). Тоді $a^* \in A^1 = A \cup A^*$.

2. Вилучення з початкової множини альтернатив A деякої підмножини A^- , $A \setminus A^- = A^1$, яка містить альтернативу $a^* \in A^-$, яка не повинна бути вибраною як розв'язок у результаті розв'язання задачі БКО (...).

3. Комбінація підходів 4.1.1 та 4.1.2: одночасне доповнення початкової множини допустимих альтернатив A деякою підмножиною A^* і вилучення небажаної для вибору підмножини A^- . Тобто $A^1 = A \cup A^* \setminus A^-$.

4. Зміна вагомості параметрів, які характеризують об'єкти початкової множини A . Для дослідження цього підходу до маніпулювання вибором автором розроблено групу методів адаптивного визначення вагових коефіцієнтів параметрів об'єктів [3].

Крім цілеспрямованої зміни вагомості параметрів об'єктів можуть бути застосовані також більш радикальні варіації цього підходу.

5. Доповнення початкової множини параметрів альтернатив A деякою підмножиною A^* , $A^* \cap A = \emptyset$, яка містить альтернативу $a^* \in A^*$, що має бути вибраною як розв'язок у результаті розв'язання ЗБКО виду (1)-(3). Тоді $a^* \in A^1 = A \cup A^*$.

6. Вилучення з початкової множини альтернатив A деякої підмножини A^- , $A \setminus A^- = A^1$, яка містить альтернативу $a^- \in A^-$, яка не повинна бути вибраною як розв'язок у результаті розв'язання задачі БКО (1)-(3).

7. Комбінація підходів 5 та 6: одночасне доповнення початкової множини допустимих альтернатив A деякою підмножиною A^* і вилучення небажаної для вибору підмножини A^- . Тобто $A^1 = A \cup A^* \setminus A^-$.

1.4 Маніпулювання з обмеженнями ЗБКО та шкалами вимірювання

1. Доповнення початкової множини обмежень S деякою підмножиною S^* , $S^* \cap S = \emptyset$.

2. Вилучення з початкової множини обмежень S деякої підмножини S^- , $S \setminus S^- = S^1$.

3. Комбінація підходів 4.2.1. та 4.2.2: одночасне доповнення обмежень початкової множини обмежень та вилучення небажаної підмножини обмежень.

4. Маніпулювання шкалами при вимірюванні параметрів альтернатив: заміна шкал вимірювання, застосування неприпустимих операцій при операціях над результатами вимірювання.

5. Маніпулювання з шкалами вимірювання при визначенні вагових коефіцієнтів параметрів, відносної ваги критеріїв, коефіцієнтів компетентності експертів (вагомості джерел інформації [6]).

5.1. При вимірюванні в ординальних шкалах організатори експертизи можуть примусити експертів визначати свої переваги у просторі строгих відношень переваги, позбавивши таким чином учасників експертної групи можливості виражати ситуацію рівноцінності або байдужості.

5.2. Можуть бути встановлені обмеження в кардинальних шкалах: наприклад, в діапазоні від 1/9 до 9, що суттєво порушує транзитивність вже на етапі початкової експертизи та апріорно не дозволяє досягти ситуації надтранзитивності відношень між параметрами, критеріями чи компетентністю експертів.

5.3. Встановлення вимоги фіксованого задання значень параметрів чи відношень переваги між об'єктами, їх параметрами, критеріями чи компетентністю експертів. Це нерідко призводить до суттєвого порушення адекватності моделювання предметної області та навмисної втрати інформації про явище, яке моделюється.

Для досягнення мети маніпулювання можна майже непомітно для невідготовленого спостерігача вибрати з початкової множини допустимих альтернатив A практично будь-яку альтернативу, бажану для ОПР.

1. При розв'язанні задачі в ординальних шкалах існує кілька найпопулярніших метрик – Хемінга, Кука, Евкліда тощо [3]. Як правило, застосування кожної з метрик дає результати, що не збігаються з розв'язками, одержаними в інших метриках. Тому ОПР достатньо лише вибрати, розв'язок за якою метрикою її найбільше влаштовує.

2. При застосуванні формул для переходу до безрозмірного виду критеріїв суттєвими є границі зміни критеріальних функцій. Вибираючи гіпотетично ці границі, можна підібрати потрібні для маніпулювання розв'язком значення і суттєво вплинути на остаточне рішення.

3. На етапі підготовки до розв'язання задачі багатокритеріальної оптимізації на попередніх етапах можна застосувати один або кілька турів процедур послідовного аналізу варіантів. Тобто попередньо встановити такі критерії, які здійснять відсів небажаних для ОПР альтернатив з початкової множини A , як заздалегідь безперспективних з огляду на «легітимно» встановлені критерії.

4. Зміна класу задач, в якому слід формалізувати початкову модель багатокритеріальної оптимізації. Це автоматично тягне за собою обґрунтоване та непомітне для фахівця застосування потрібних для досягнення мети маніпулювання методів розв'язання задачі та застосування відповідних видів згортки критеріїв задачі.

5. Доповнення початкової множини критеріїв деякою новою підмножиною або одним, навіть несуттєвим критерієм. Таким чином можна зробити ефективною альтернативу, яка у попередній моделі була домінованою. Така маніпуляція може суттєво вплинути на множину ефективних рішень початкової ЗБКО. А відтак – і на визначення компромісного розв'язку задачі (1)-(3).

6. Вилучення з початкової множини критеріїв R деякої підмножини критеріїв або навіть одного з них. Таке втручання у процес може суттєво вплинути та побудовану багатокритеріальну модель досліджуваного явища.

7. Комбінація підходів 5 та 6 може спричинити синергетичний ефект у порівнянні з автономним застосуванням кожного з цих підходів. Зокрема, можна вивести з вибору не вигідну для ОПР домінуючу альтернативу під деяким слушним приводом.

1.5 Маніпулювання шляхом впливу на експертну групу

1. Заміна (відвід) неугодного експерта у разі, коли призначена одноосібна експертиза.

2. Доповнення початкової множини експертів деякою підмножиною експертів, на яких має вплив ОПР або інші учасники прийняття рішення, зацікавлені у маніпулюванні – для підготовки та обґрунтування потрібного рішення.

3. Вилучення з початкової множини експертів тих, які є незручними для ОПР та мають свою особисту думку.

4. Комбінація підходів 2 та 3 – одночасне доповнення групи експертів новими членами групи та відведення з групи деяких її членів.

1.6 Маніпулювання множиною цілей, що задекларовані при розв'язанні задачі

1. Збільшення підмножини вибору в ході розв'язання задачі, що дозволить ввести до підмножини переможців бажані для ОПР альтернативи.

2. Досягнення бажаного для ОПР рішення: з допомогою потужного інструменту, оригінального математичного забезпечення можна досягти як узгодження рішень різних рівнів, так і надати широкі і непомітні можливості для недоброчесного маніпулювання.

3. У випадку, коли підготовлене та обґрунтоване рішення не задовольняє ОПР, можна досягти визнання вибору недійсним з різних мотивів.

1.7 Маніпулювання системою переваг експертів

1. Мінімальними змінами у деяких відношеннях переваги, заданих експертами, можна досягти зміни початкового компромісного рішення.

2. Для досягнення мети маніпулювання може слугувати заміна лише деяких індивідуальних переваг експерта з метою згладжування його переваг або узгодження групового експертного рішення.

3. Маніпулювання ваговими коефіцієнтами компетентності експертів (вагомості джерел інформації [6]) можна досягнути бажаного для ОПР рішення або, щонайменше, блокування того рішення, яке є найбільш неприйнятним.

4. Виключення деяких елементів матриці парних порівнянь з розгляду – тобто переведення її у категорію неповних матриць. Таким чином змінюється не тільки інформація, але й методи розв’язання задачі експертного оцінювання. Це може потягнути за собою обчислення потрібного компромісного рішення.

5. Заміна елемента матриці парних порівнянь, який найсуттєвішим чином впливає на результат – спосіб непрямого впливу на остаточний розв’язок ЗБКО

1.8 Висновки до першого розділу

Досліджено різні аспекти та можливості маніпулювання вибором у задачах експертного оцінювання у випадку формалізації їх у вигляді моделі багатокритеральної оптимізації.

У загальному випадку розглядаються різні аспекти та джерела можливого маніпулювання у задачах експертного оцінювання:

- виборцем;
- організатором процедури;
- ОПР.

Вибір може бути обґрунтованим або волюнтаристським. При добре організованій маніпуляції ОПР можна підвести до потрібного для маніпулятора вибору. Тому в задачах експертного оцінювання особливо важливим є чітке представлення евристик, на які спираються дослідники при визначенні компромісного розв’язку.

РОЗДІЛ 2 РОЗРОБКА АРХІТЕКТУРИ ДОДАТКУ З ВИЗНАЧЕННЯ ПРІОРИТЕТНОСТІ КАНДИДАТА

2.1 Розробка архітектури

2.1.1 Функціональний аналіз

Бізнес-процеси допомагають отримати повну картину початку розробки системи. Модуль визначеності пріоритету кандидата вершина дерева функцій. Основні модулі функціонування сайту діляться на два модулі: клієнтський модуль та адміністративний модуль.

До клієнтського модулю функцій належать основні три модулі: модуль ведення акаунту який включає в собі можливість редагувати особистий акаунт користувача, модуль реєстрації для нових користувачів які не мають акаунту на сайті та головний модуль перегляду кандидатів, головна функція отримання рекомендацій. Модуль рекомендацій вміщує в собі модуль сортування готовими модулями сортування: сортування за прізвищем, очікуваною зарплатою та інші варіанти сортування. Модуль отримання рекомендацій експертів дозволяє отримувати оцінку та довіритись живій людині замість алгоритму.

До адміністративних функцій належить модуль роботи з кандидатами та можливістю додаванням нових кандидатів до бази даних сайту.



Рисунок 1 – Дерево функцій

Функціонування сайту визначення пріоритету кандидата починається з відкриттям сторінки в браузері за посиланням. Менеджер авторизується на сайті або реєструється - якщо новий користувач та не має особистого кабінету. Після авторизації менеджер переходить до сторінки кандидатів та має можливість переглянути зі списку основних кандидатів які є на сайті. Користувач відвідує основні функції сайту, а саме можливість сортування через експертів за їх оцінками, користувач отримує відсортований список від експертів за даними кандидатів на даний момент часу.

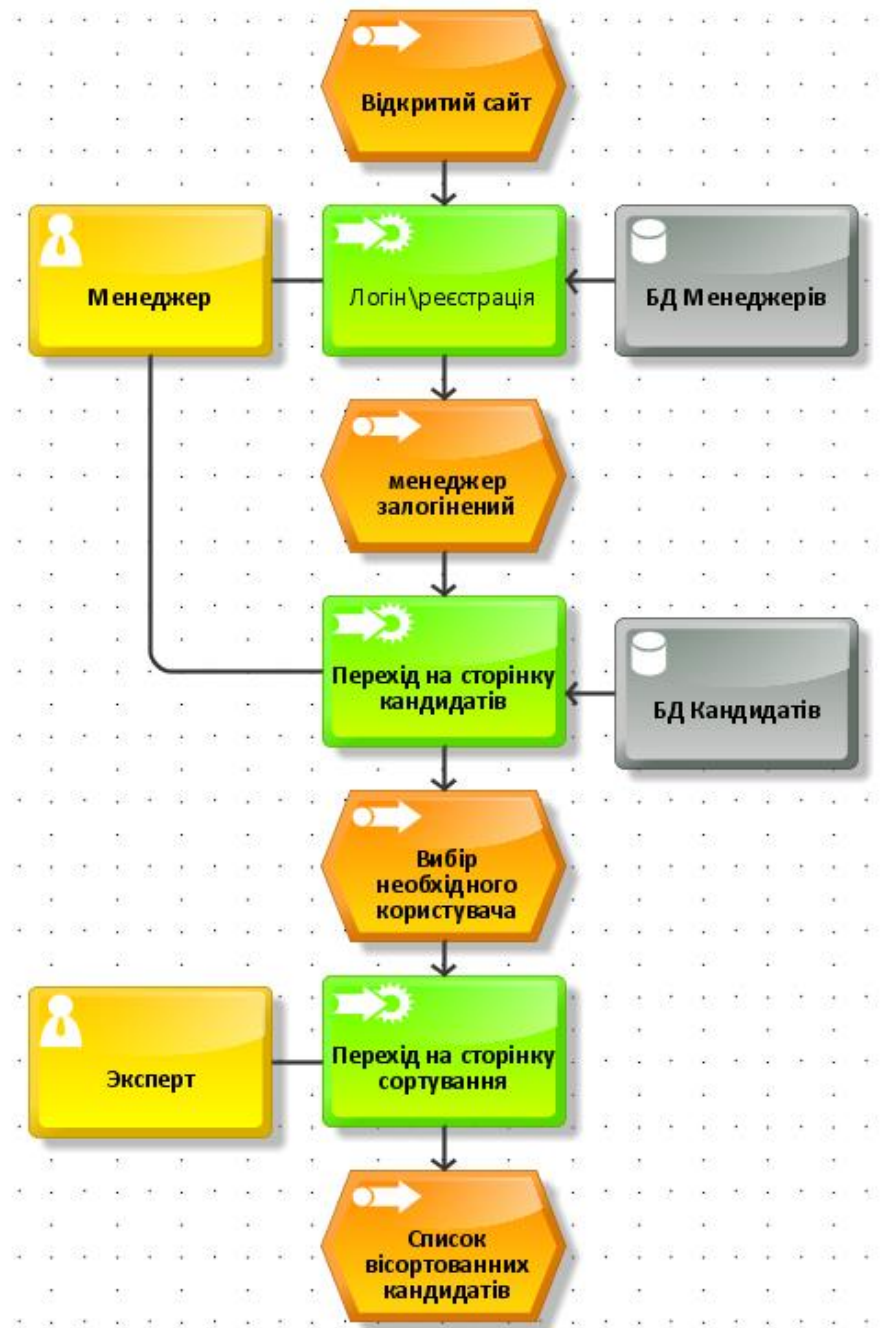


Рисунок 2 – Діаграма Event-Driven Process Chain для функціонування сайту

Наступний етап проектування є створення дерева ланцюжків для відображення що відбувається під час функціонування сайту. На вершині знаходиться головний процес “Робота з сортуванням кандидатів” який вміщує два основних підпроцеси: “Реєстрація в додатку визначення пріоритетності кандидатів на посаду” та підпроцес “Користування додатком визначення пріоритетності кандидатів на посаду”.

До процесу “Реєстрація в додатку визначення пріоритетності кандидатів на посаду” входять два головних підпроцеси: “Заповнення основних даних” який включає в собі підпроцеси заповнення основних полей для реєстрації: “Заповнення логіну”, “Заповнення паролю” та “Заповнення ПІБ”. Другим підпроцесом процесу “Реєстрація в додатку визначення пріоритетності кандидатів на посаду” є “Підтвердження реєстрації” яка здійснюється через електронну пошту.

До процесу “Користування додатком визначення пріоритетності кандидатів на посаду” входить головний підпроцес сайту “Вибір кандидата” який вміщує в собі чотири види процесу сортування: “Сортування за прізвищем”, “Сортування за очікуваною зарплатою”, “Сортування за віком”, та “Сортування експертів”.

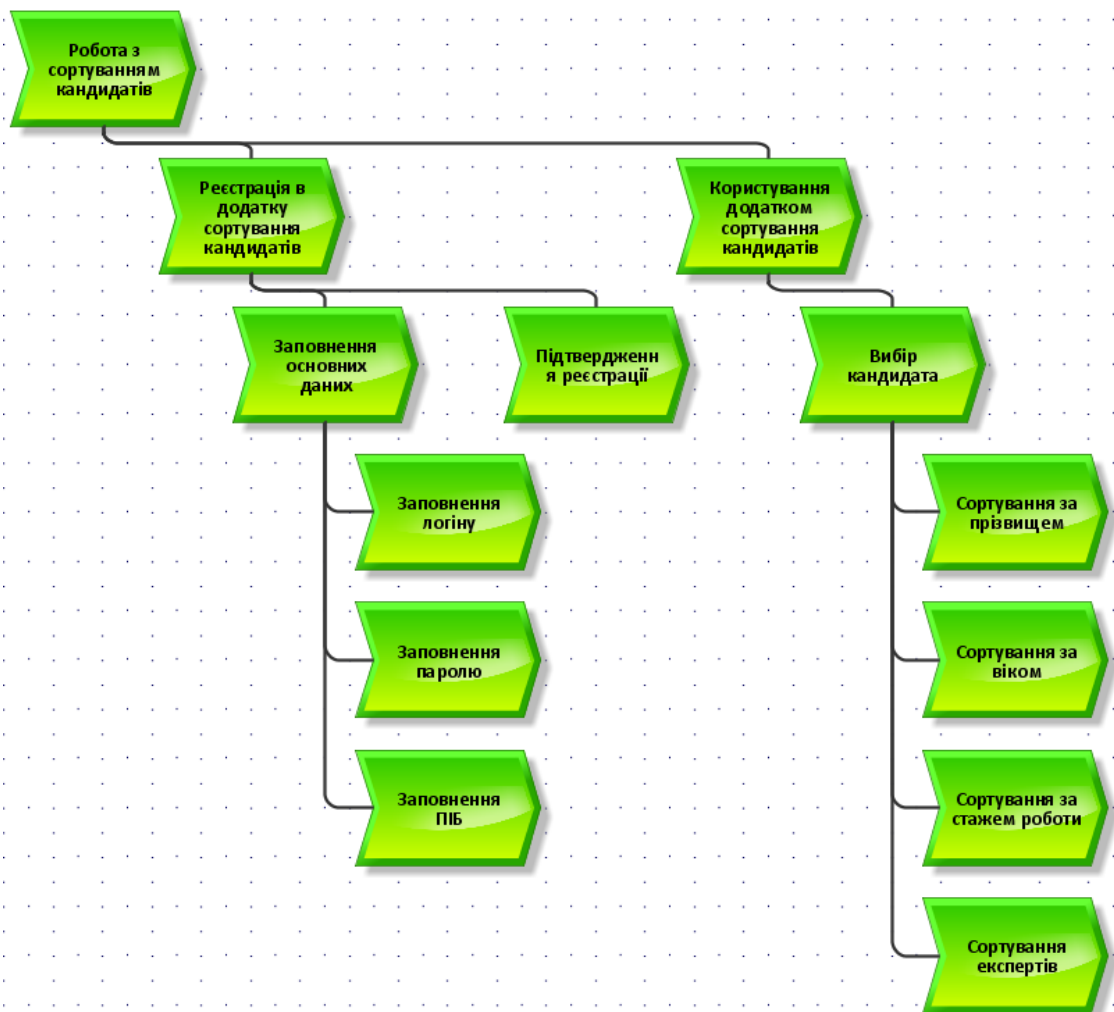


Рисунок 3 – Діаграма Value Added Chain Diagram

2.1.2 IDEF0 процесу видачі рекомендацій за допомогою інформаційної системи

Представимо виконувані процеси у предметному середовищі за допомогою діаграм «ЯК БУДЕ» у нотації IDEF0, тобто вже з використанням сайту визначення пріоритету кандидата.

Першим етапом запуску процесу є отримання інформації користувача на вході до системи. Система рекомендацій кандидатів отримує дані які вводить менеджер та підтверджує вибір необхідного кандидата якого переглядає експерт та видає необхідний список кандидатів.

Елементом керування є закон України про «Захист персональних даних», тому що на кожному етапі зберігаються особисті дані користувача який має юридичну силу зберігання особистих даних користувача.

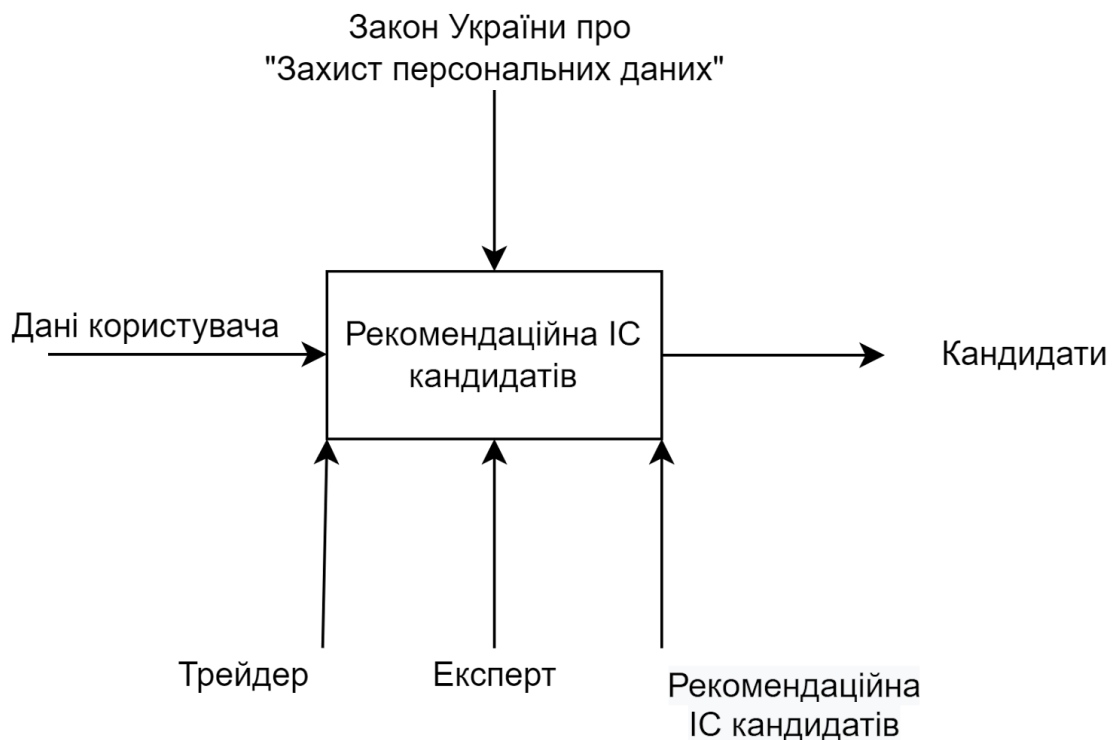


Рисунок 4 - Контекстна діаграма ЯК БУДЕ

На наступному етапі проведемо декомпозицію основних процесів діаграми ЯК БУДЕ.

Перший процес отримання системою даних користувача для авторизації та перехід до сторінки вибору необхідного кандидата для наступного шагу та передачі даних кандидата процесу аналізу даних який переглядає експерт завдяки сортуванню системи та видає свої оцінки для створення списку кандидатів для користувача та самого рекомендаційного списку. На виході системи користувач отримує необхідну інформацію про кандидата та оцінку експертів.

Елементом керування є закон України про «Захист персональних даних», тому що на кожному етапі зберігаються особисті дані користувача який має юридичну силу зберігання особистих даних користувача.

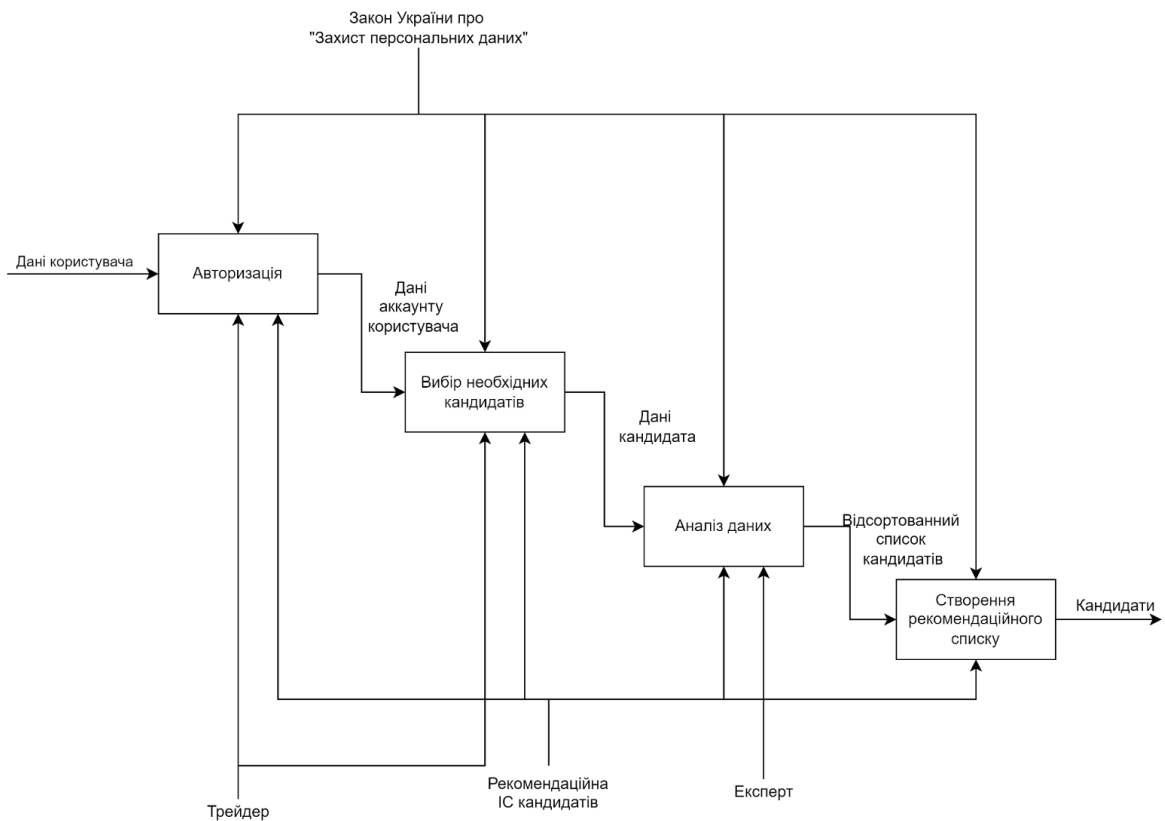


Рисунок 5 – Декомпозиція основних процесів діаграми ЯК БУДЕ

2.1.3 Архітектура інформаційної системи

Новий етап - побудова архітектури розробленої інформаційної системи. Дана схема відображає модель, структуру, виконувані функції та взаємозв'язок компонентів. На вершині знаходиться модуль перегляду кандидатів який вміщує в свою чергу два основних модулі: клієнтський модуль та адміністративний модуль.

До клієнтського модулю входять три модулі: модуль роботи з профілем який дозволяє переглядати основні поля та інформацію користувача. Модуль реєстрації для нових користувачів які не відвідували сайт раніше та головний модуль всіх користувачів - модуль визначення пріоритету кандидата. Модуль визначення пріоритету кандидата дозволяє користувачам користуватися модулями сортування за прізвищем, модулем сортування за віком, модулем сортування за бажаннями користувача зарплатної плати та головний модуль - сортування з експертами кандидатів.

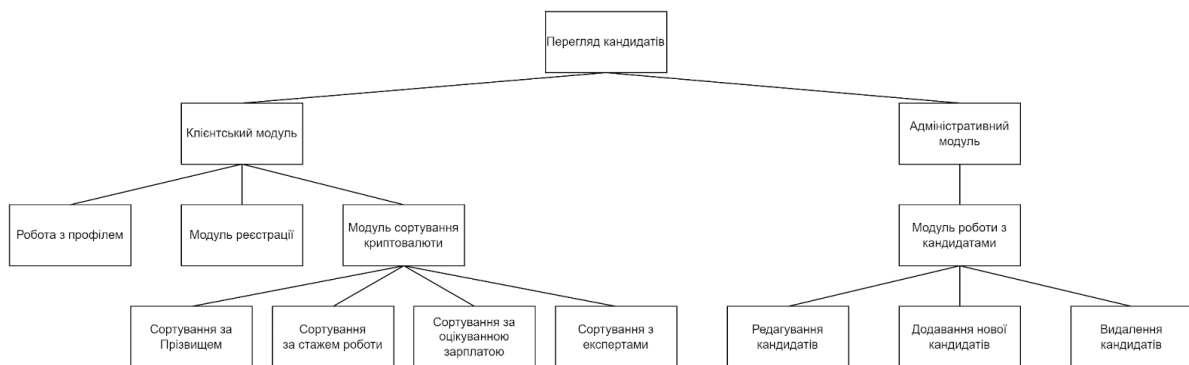


Рисунок 6 – Архітектура системи

2.2 Інформаційне забезпечення рекомендаційної

2.2.1 Розробка інформаційного забезпечення

Як вже було зазначено раніше, дана інформаційна система для правильної роботи використовує певні дані. Для збереження цих даних буде розроблена база даних, яка зберігатиме та видаватиме потрібну інформацію.

Спочатку розробимо концептуальну модель бази даних (рис.). На ній для заданої предметної області відображено наступні сутності:

- Менеджер
- Кандидат

Кожна з них має свої атрибути, тобто майбутні поля у таблицях. Наприклад, «Менеджер» має логін, пароль та ім'я, які він вводить при реєстрації. Та «Кандидат» яка вміщує в собі поля ціни назви та інші.

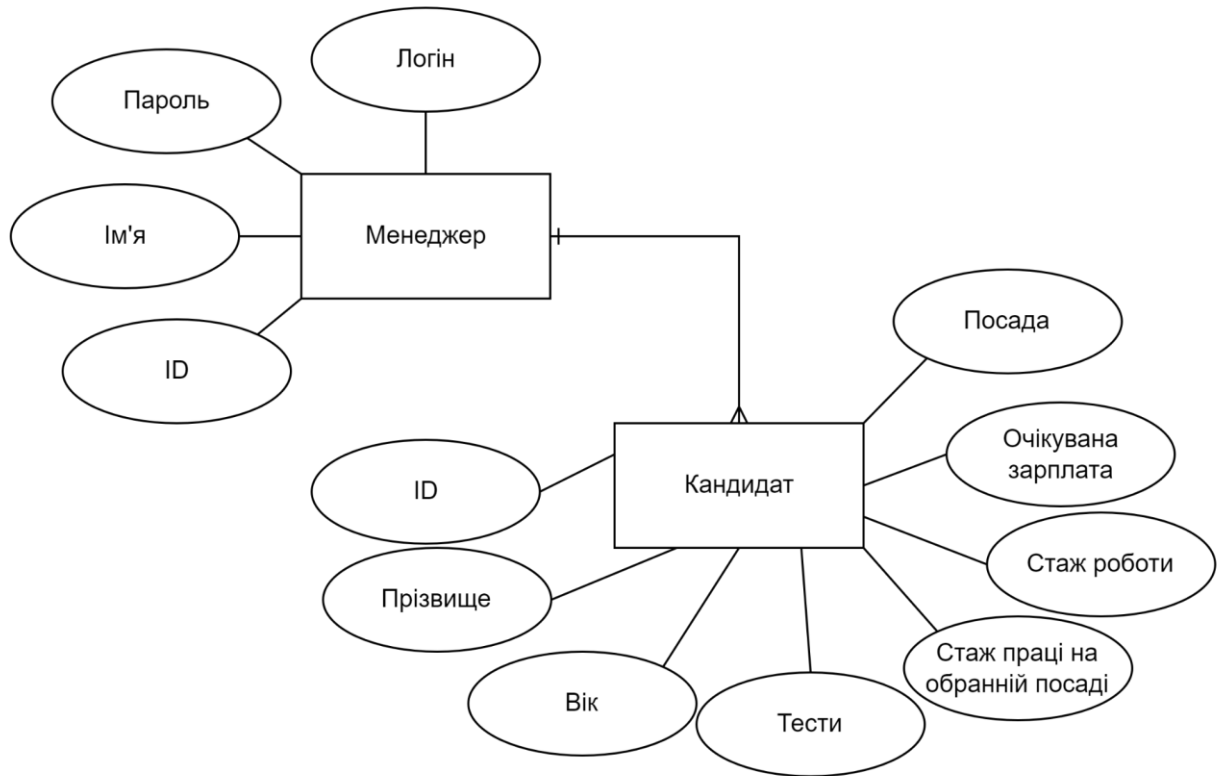


Рисунок 7 – Концептуальна модель бази даних

Таблиця 1

Зв'язки між сутностями

№	Сутності, що утворюють зв'язок	Тип зв'язку	Пояснення
1	Менеджер - Кандидат	Один до багатьох	Один менеджер має декілька кандидатів. Інформація про кандидатів отримує один менеджер.

Наступною є логічна модель (рис.). Вона також, як і концептуальна, описує дані на досить абстрактному рівні, але все ж більше наближеному до реальних баз даних.

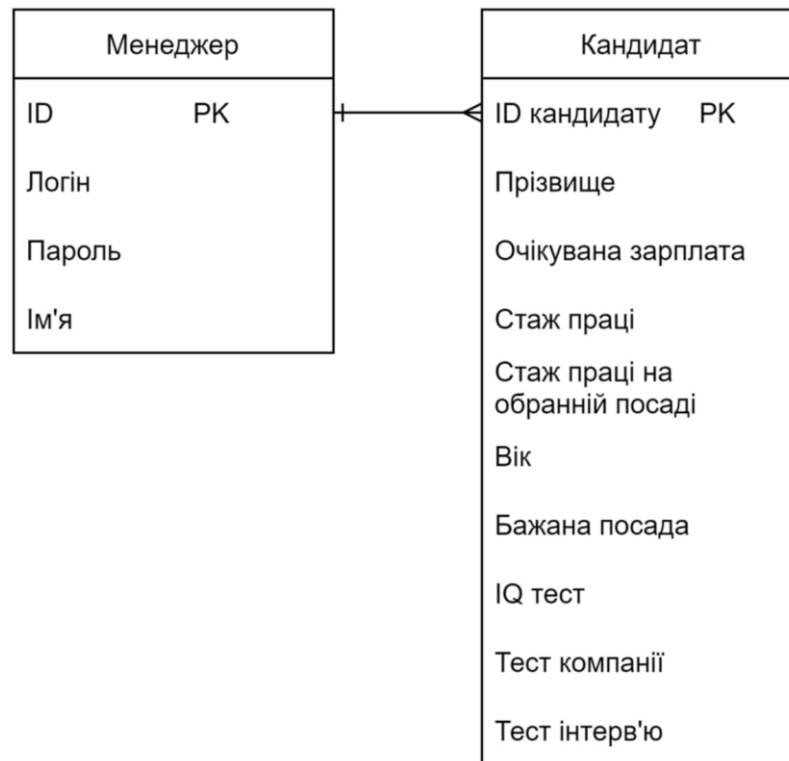


Рисунок 8 – Логічна модель бази даних

Останньою є фізична модель бази даних, розроблена на основі даталогічної. Вона вже відображає, як дані насправді зберігаються в базі даних. На ній вже можна побачити детальніші дані про атрибути, а саме – тип даних та обмеження для деяких з них. Опишемо їх у таблиці.

Таблиця 2

Склад та характеристика атрибутів таблиць

№ п/п	Назва елемента даних	Тип елемента даних	Обов'язкове значення	Обмеження	Ключ
Таблиця «Менеджер»					
1	ІД менеджера	Ціле число	так	30	ПК
2	Пароль	Символьний	так	–	–
3	Логін	Символьний	так	–	–

Продовження таблиці 2

4	Ім'я	Символьний	так	20	–
Таблиця «Кандидат»					
1	ID кандидата	Ціле число	так	30	ПК
2	Прізвище	Символьний	так	–	–
3	Очікувана зарплата	Ціле число	так	–	–
4	Стаж праці	Ціле число	так	–	–
5	Вік	Ціле число	так	–	–
6	Стаж роботи на очікуваній посаді	Ціле число	–	–	–
7	Очікувана посада	Символьний	–	–	–
8	IQ тест	Ціле число	–	–	–
9	Тест компанії	Ціле число	–	–	–
10	Тест інтерв'ю	Ціле число	–	–	–

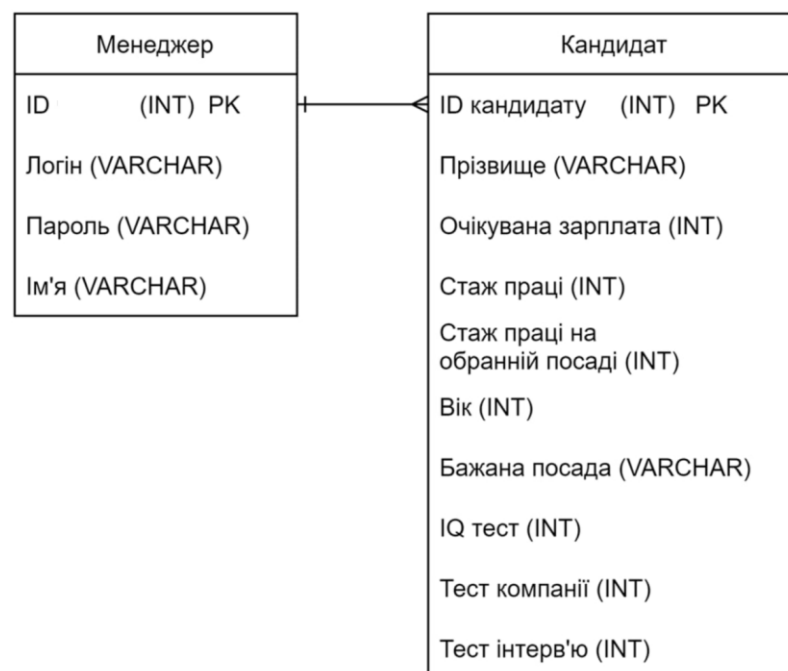


Рисунок 9 – Фізична модель бази даних

2.3 Висновки до другого розділу

В результаті розробки архітектури сайту рекомендації кандидатів було спроектовано систему для подальшої її реалізації.

Було проведено функціональний аналіз та побудовано дерево функцій, яке відображає бізнес-процеси інформаційної системи. Також було розроблено діаграми Event-Driven Process Chain та Value Added Chain Diagram для функціонування web-додатку. На даних схемах було зображено та описано логіку роботи програми. Створена база даних додатку визначення пріоритетності кандидатів на посаду, фізична модель та логічна модель БД.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ WEB – ДОДАТКУ ДЛЯ ВИЗНАЧЕННЯ ПРІОРИТЕТНОСТІ КАНДИДАТА

3.1 Обґрунтування вибору програмних засобів

Для розробки застосунку було використано ряд інструментів та методів, за допомогою яких створено інформаційну систему з відповідною базою даних та користувацьким інтерфейсом.

Основна мова програмування JavaScript ES6 з використанням HTML5 та CSS3.

Було використані головні можливості JavaScript ES6 ECMAScript ³/₅, а саме використання особистого конструктора завдяки DOM елементам які будуються відповідно до бази даних та відсортованих списків, було використовувані функції запуску та відображення можливостей сортування та можливостей зберігання особистих кандидатів.

3.2 Структура програмного забезпечення

Сайт будує себе самостійно в залежності від бази даних завдяки DOM елементам які описані в файлах скрипту які відображаються на основній сторінці. Існує є лише основна та головна сторінка .html “Головна сторінка” яка відображає всі необхідні елементи.

Таблиця 3

Специфікація програмних модулів

Модуль	Опис
Головна сторінка	<p>Початковий модуль, який відображає блок авторизації та на якому знаходиться на початку вікно авторизації та після успішної авторизації відображення списку кандидатів та можливих модулів сортування</p> <p>-</p> <p>Вхідна інформація: Логін та пароль користувача.</p> <p>-</p> <p>Вихідна інформація: Список кандидатів та можливі функції сортування.</p>

Продовження таблиці 3

script.js	Головний файл який зберігає основні функції відображення та сортування інформації на сайті - Вихідна інформація: Список відсортованої кандидатів.
function login(user)	Головна функція авторизації користувача - Вхідна інформація: Логін та пароль користувача. Вихідна інформація: Відображення авторизованих користувачів.
function expertsList()	Функція сортування завдяки інформації існуючих кандидатів та даних на даний момент по кожній та дані від експертів. - Вихідна інформація: Відсортований список кандидатів.
function buildCoinList()	Функція відображення кандидатів на початку та відсортованих. - Вихідна інформація: Відображення списку криптовалюти

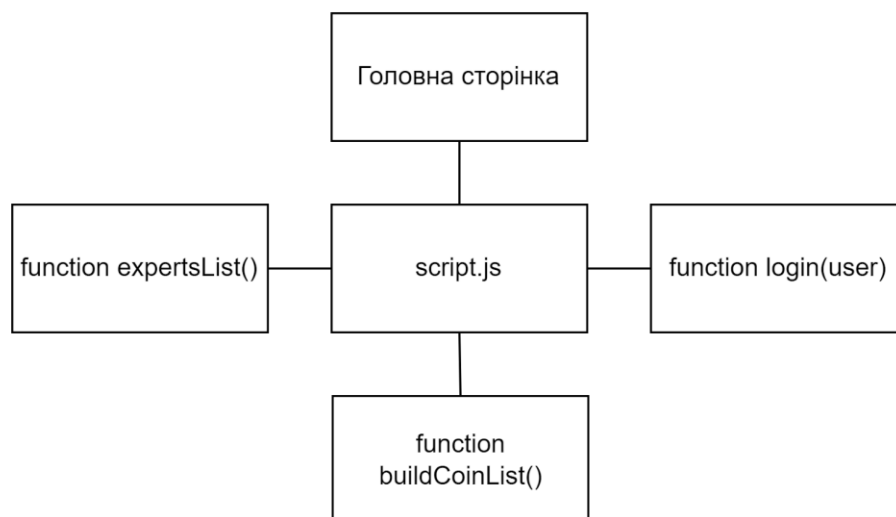
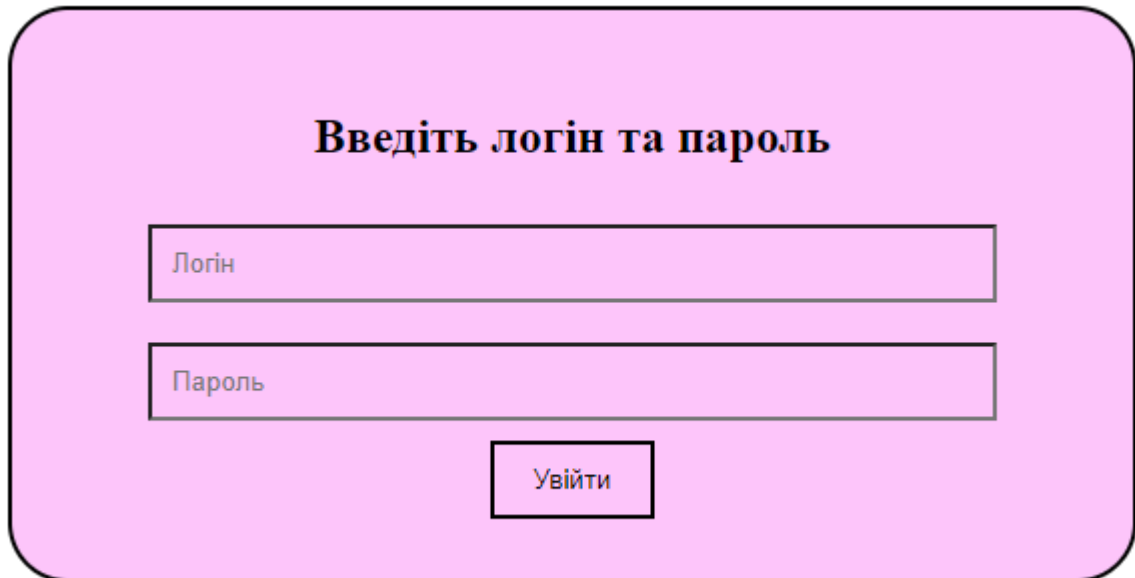


Рисунок 10 – Структурна схема програмних модулів

3.3 Керівництво користувача

Початок користувача на сайті визначення пріоритету кандидата починається з авторизації, в полях якої користувач вводить особисті дані для авторизації.



The image shows a login form with a light purple background and rounded corners. At the top, the text "Введіть логін та пароль" is centered. Below it are two horizontal input fields. The first field is labeled "Логін" and the second is labeled "Пароль". Below the second field is a button labeled "Увійти".

Рисунок 11 – Відображення полів для вводу особистих даних для авторизації

User1

Відсортувати за:

Рисунок 12 – Відображення авторизованого користувача

При успішній авторизації користувачу відображується головна сторінка сайту сортування андидатів. На якій користувач відразу бачить весь список кандидатів та їх інформацію в лівий частині екрану на якій списком відображено кожену людину яка бажає влаштуватись до компанії, її прізвище та іншу інформацію. Також можливість додати кандидата до особистого списку вподобань.








Всі кандидати										
1.	 Бандера С. Чоловік	Очікувана зарплата: 500\$	Вік: 28 років	Стаж праці: 5	Стаж праці бажаної посади: 3	Бажана посада: Програміст	Iq тест: 110	Тест компанії: 90%	Оцінка інтерв'ю: 90%	Додати в окремий список
2.	 Шевченко І. Чоловік	Очікувана зарплата: 2000\$	Вік: 45 років	Стаж праці: 10	Стаж праці бажаної посади: 9	Бажана посада: Керівник	Iq тест: 100	Тест компанії: 40%	Оцінка інтерв'ю: 70%	Додати в окремий список
3.	 Карпенко А. Жінка	Очікувана зарплата: 1500\$	Вік: 75 років	Стаж праці: 30	Стаж праці бажаної посади: 15	Бажана посада: Менеджер	Iq тест: 90	Тест компанії: 30%	Оцінка інтерв'ю: 60%	Додати в окремий список
4.	 Гамонька С. Жінка	Очікувана зарплата: 2800\$	Вік: 40 років	Стаж праці: 20	Стаж праці бажаної посади: 9	Бажана посада: Програміст	Iq тест: 70	Тест компанії: 60%	Оцінка інтерв'ю: 77%	Додати в окремий список
5.	 Стус В. Чоловік	Очікувана зарплата: 300\$	Вік: 21 років	Стаж праці: 0	Стаж праці бажаної посади: 0	Бажана посада: Менеджер	Iq тест: 100	Тест компанії: 100%	Оцінка інтерв'ю: 50%	Додати в окремий список
6.	 Шевченко В. Чоловік	Очікувана зарплата: 1200\$	Вік: 25 років	Стаж праці: 5	Стаж праці бажаної посади: 1	Бажана посада: Менеджер	Iq тест: 99	Тест компанії: 60%	Оцінка інтерв'ю: 0%	Додати в окремий список
7.	 Карпенко А. Чоловік	Очікувана зарплата: 10\$	Вік: 15 років	Стаж праці: 0	Стаж праці бажаної посади: 0	Бажана посада: Керівник	Iq тест: 87	Тест компанії: 0%	Оцінка інтерв'ю: 0%	Додати в окремий список

Рисунок 13 – Відображення списку кандидатів

Верхня частину сайту відображає основні функції сортування списку кандидатів. Існує чотири основні функції швидкого сортування: “За прізвищем”, “Очікувана зарплата”, “Вік” та “Стаж праці” та дві особливі функції “Мої улюблені” та “Експерти”

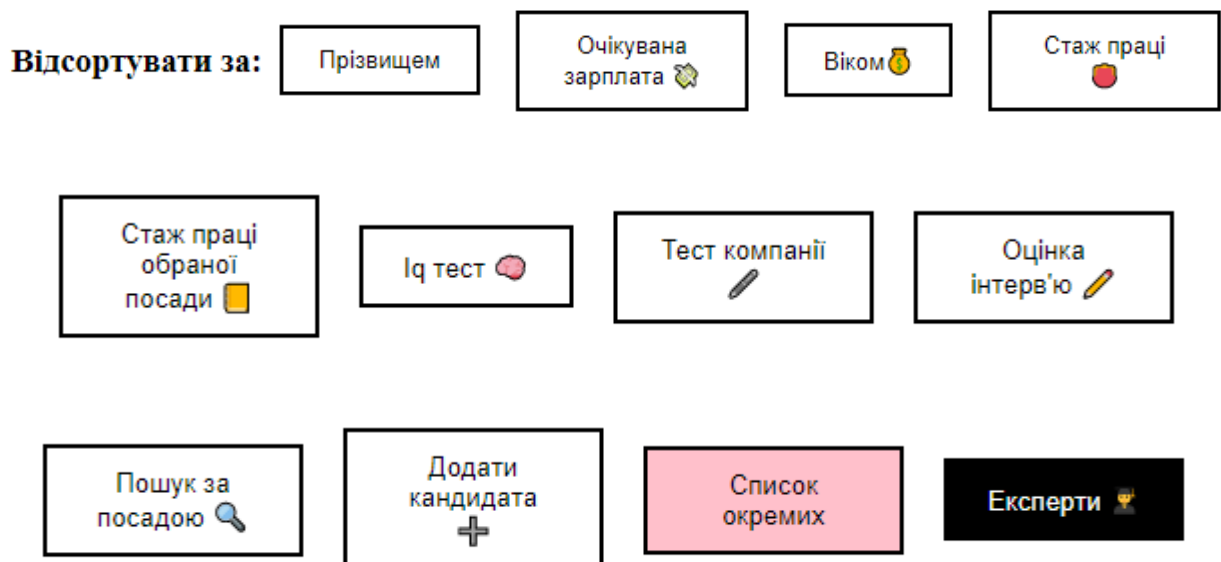


Рисунок 14 – Відображення списку можливих сортувань

Коли користувача натискає кнопку “Прізвище” відображається відсортований список кандидатів за Прізвищем.

ідсортувати за: Прізвищем


1.		Бандера С.	
		Чоловік	
2.		Гамоцька С.	Очікувана зарплата 28
		Жінка	
3.		Гонгало Ф.	Очікувана зарплата 40
		Жінка	
4.		Карпенко А.	

Рисунок 15 – Відображення списку людей за прізвищем

Коли користувача натискає кнопку “Очікувана зарплата” відображається відсортований список людей за їх очікуваною заробітною платою.

Відсортувати за:

Прізвищем

Очікувана
зарплата 

1.		Гонгало Ф. Жінка	Очікувана зарплата: 4000\$	Вік рок
2.		Гамоцька С. Жінка	Очікувана зарплата: 2800\$	Вік рог
3.		Шевченко І. Чоловік	Очікувана зарплата: 2000\$	Вік: рок
4.		Токарь А. Чоловік	Очікувана зарплата: 1999\$	Вік рог

Рисунок 16 – Відображення списку людей за очікуваною зарплатою

Коли користувача натискає кнопку “Віком” відображається відсортований список людей за віком.

Відсортувати за:





1.		Карпенко А. Жінка	Очікувана зарплата: 1500\$	Вік: 75 років	Стаж праці: 30
2.		Шевченко І. Чоловік	Очікувана зарплата: 2000\$	Вік: 45 років	Стаж праці: 10
3.		Гонгало Ф. Жінка	Очікувана зарплата: 4000\$	Вік: 45 років	Стаж праці: 27
4.		Гамопька С.	Очікувана зарплата: 2800\$	Вік: 40 років	Стаж праці: 20

Рисунок 17 – Відображення списку людей за віком

Коли користувача натискає кнопку “Стаж праці” відображається відсортований список кандидатів за стажем праці.

Відсортувати за:




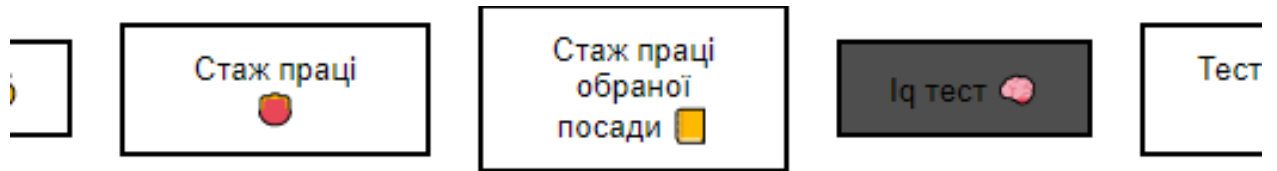
1.		Карпенко А. Жінка	Очікувана зарплата: 1500\$	Вік: 75 років	Стаж праці: 30	Стаж праці бажаної: 15
2.		Гонгало Ф. Жінка	Очікувана зарплата: 4000\$	Вік: 45 років	Стаж праці: 27	Стаж праці бажаної: 20
3.		Гамопька С. Жінка	Очікувана зарплата: 2800\$	Вік: 40 років	Стаж праці: 20	Стаж праці бажаної: посади:

Рисунок 18 – Відображення списку людей з найбільшим стажем праці



Всі кандидати

Стаж праці: 5	Стаж праці бажаної посади: 3	Бажана посада: Программіст	Iq тест: 110
Стаж праці: 10	Стаж праці бажаної посади: 9	Бажана посада: Керівник	Iq тест: 100
Стаж праці: 0	Стаж праці бажаної посади: 0	Бажана посада: Менеджер	Iq тест: 100
Стаж праці: 5	Стаж праці бажаної посади: 1	Бажана посада: Менеджер	Iq тест: 99

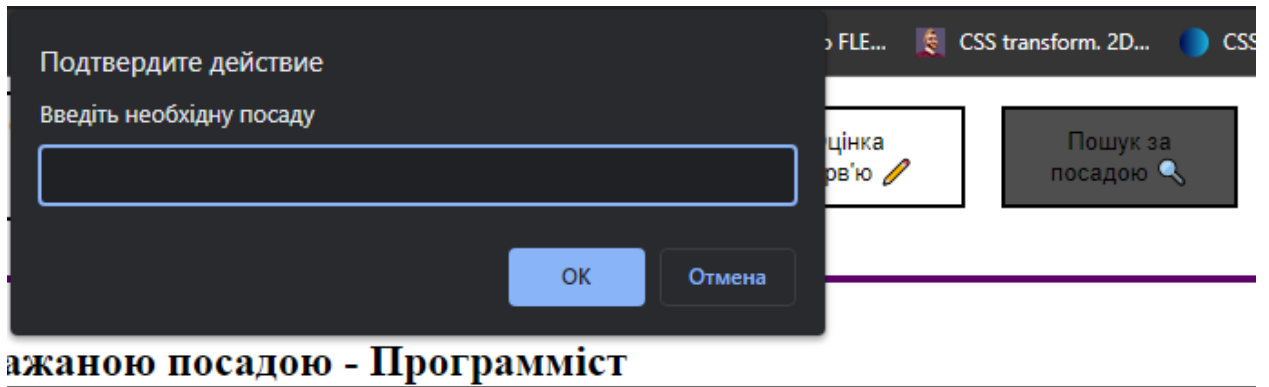
Рисунок 19 – Відображення списку людей з найбільшим IQ тестом



ІДЕНТИ

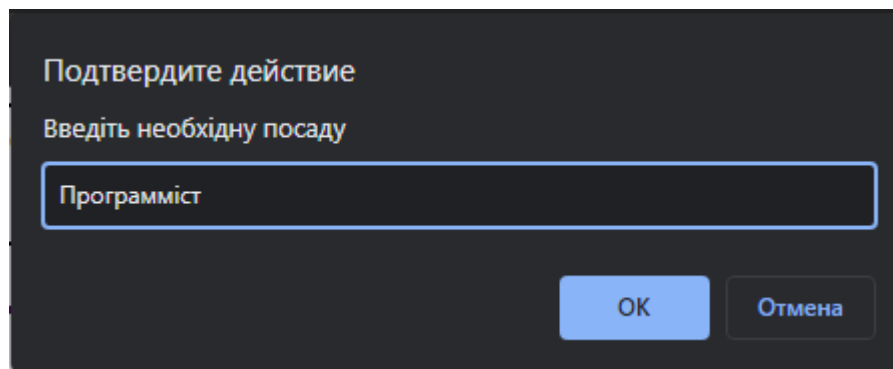
Імена посади: Менеджер	Iq тест: 100	Тест компанії: 100%	Оцінка інтерв'ю: 50%	Додати окремий список
Імена посади: Інженер	Iq тест: 92	Тест компанії: 100%	Оцінка інтерв'ю: 50%	Додати окремий список
Імена посади: Програміст	Iq тест: 110	Тест компанії: 90%	Оцінка інтерв'ю: 90%	Додати окремий список
Імена посади: Програміст	Iq тест: 99	Тест компанії: 90%	Оцінка інтерв'ю: 77%	Додати окремий список
Імена посади: Програміст	Iq тест: 97	Тест компанії: 77%	Оцінка інтерв'ю: 80%	Додати окремий список

Рисунок 20 – Відображення списку людей з найбільшим результатом за тест компанії



Інтерв'ю

Рисунок 21 – Відображення списку людей з найбільшим результатом за тест інтерв'ю



Всі кандидати з бажаною посадою - Программіст




1.	 Бандера С. Чоловік	Очікувана зарплата: 500\$	Вік: 28 років	Стаж праці: 5	Стаж праці бажаної посади: 3	Бажана посада: Программіст	Iq те 11
2.	 Прохова А. Жінка	Очікувана зарплата: 1002\$	Вік: 22 років	Стаж праці: 4	Стаж праці бажаної посади: 0	Бажана посада: Программіст	Iq те 99
3.	 Токарь А. Чоловік	Очікувана зарплата: 1999\$	Вік: 25 років	Стаж праці: 7	Стаж праці бажаної посади: 5	Бажана посада: Программіст	Iq те 97
4.	 Гамопька С. Жінка	Очікувана зарплата: 2800\$	Вік: 40 років	Стаж праці: 20	Стаж праці бажаної посади: 9	Бажана посада: Программіст	Iq те 70

Рисунок 22 – Відображення списку людей за пошуком посади

Також менеджер має можливість додати користувача до списку завдяки кнопці “Додати кандидата”. Відкривається поле для вводу інформації для нового користувача.

Рисунок 23 – Відображення окремого списку кандидатів

Як додавати так і можливо видаляти назавжди непотрібного кандидата якого ми вже не будемо оцінювати.


1.	 Бандера С. Чоловік	Очікувана зарплата: 500\$	Вік: 28 років	Стаж праці: 5	Стаж праці бажаної посади: 3	Бажана посада: Програміст	Іq тест: 110	Тест компанії: 90%	Оцінка інтерв'ю: 90%	<input type="button" value="Додати в окремий список"/>	<input type="button" value="Видалити кандидата"/>
----	--	------------------------------	------------------	------------------	---------------------------------	------------------------------	-----------------	-----------------------	-------------------------	--	---

Рисунок 24 – Видалення інформації про окремого кандидата

При використанні навпроти кандидата функції “Додати в окремий список” користувач додав до окремого списку кандидата який відображається в кладці “Окремий список”.


Окремий список кандидатів											
1.		Карпенко А. Чоловік	Очікувана зарплата: 10\$	Вік: 15 років	Стаж праці: 0	Стаж праці бажаної посади: 0	Бажана посада: Керівник	Iq тест: 87	Тест компанії: 0%	Оцінка інтерв'ю: 0%	Видалити зі списку ✕

Рисунок 25 – Відображення окремого списку кандидатів







Додано до списку 

Рисунок 26 – Повідомлення про успішне додавання в окремий список

У вкладці “Окремий список” відображається список людей та можливість видалити інформацію окремого кандидата з окремого списку. Якщо видалити інформацію про окрему людину – вона перестала відображатись в окремому списку.

Відсортувати за:

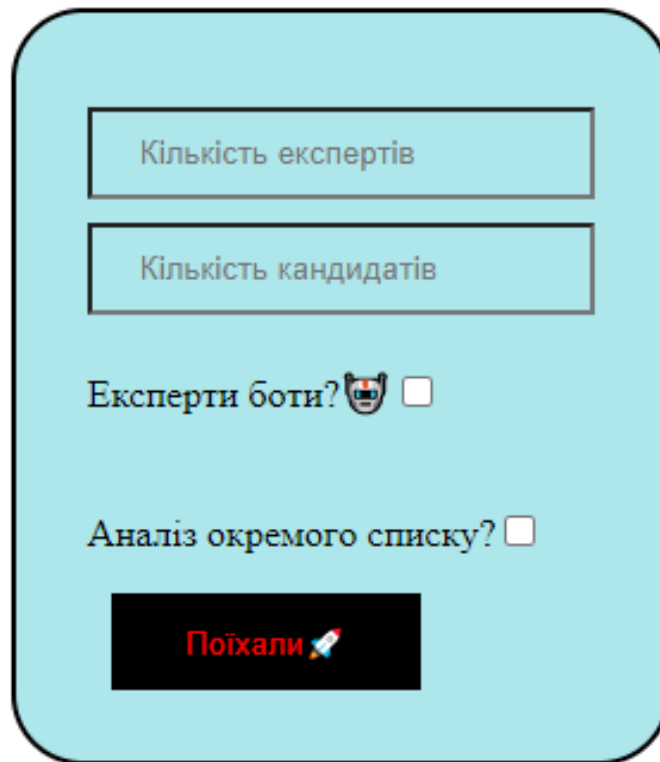
- Прізвищем
- Очікувана зарплата 
- Віком 
- Стаж праці 
- Стаж праці обраної посади 
- Iq тест 

Ви не вибрали жодного кандидата!

Окремий список кандидатів

Рисунок 27 – Повідомлення про пустий список улюблених кандидатів та повідомлення про видалення кандидата

При використанні кнопки “Експерти” користувачу надається доступ до вікна в якому відображаються поля для вводу кількості експертів які будуть приймати участь в оцінюванні кандидата, кількість кандидатів – вписуємо число кількості людей які будуть отримувати користувач. Пункт “Експерти боти?” надає можливість системі сортування пропустити момент вписання оцінок від людей та замінює їх голоси підбором який залежить від удачі. Пункт “Аналіз мого списку?” допомагає сортувати з особистого списку кандидатів.



Кількість експертів

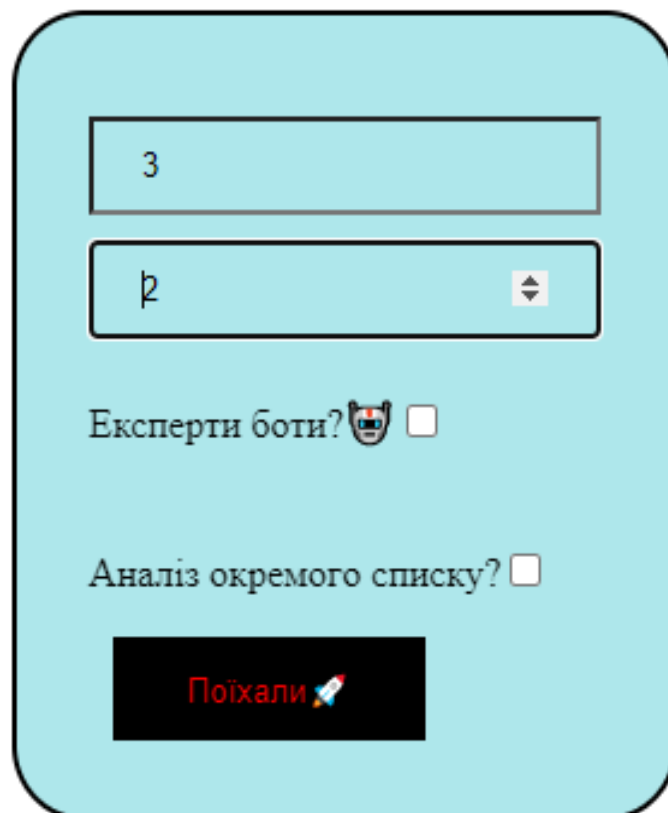
Кількість кандидатів

Експерти боти? 🤖

Аналіз окремого списку?

Поїхали 🚀

Рисунок 28 – Меню для вибору характеристик сортування



3

2

Експерти боти? 🤖

Аналіз окремого списку?

Поїхали 🚀

Рисунок 29 – Інформація від користувача з необхідною кількістю експертів та людей


Введіть результати експертів										
Сортувати										
1.	 Прохова А. Жінка	Очікувана зарплата: 1002\$	Вік: 22 млрд \$	Стаж праці: 4 млрд \$	Стаж праці бажаної посади: 0	Бажана посада: Програміст	Iq тест: 99	Тест компанії: 90%	Оцінка інтерв'ю: 77%	1 експерт: <input type="text"/> 2 експерт: <input type="text"/>
2.	 Бандера С. Чоловік	Очікувана зарплата: 500\$	Вік: 28 млрд \$	Стаж праці: 5 млрд \$	Стаж праці бажаної посади: 3	Бажана посада: Програміст	Iq тест: 110	Тест компанії: 90%	Оцінка інтерв'ю: 90%	1 експерт: <input type="text"/> 2 експерт: <input type="text"/>
3.	 Токарь А. Чоловік	Очікувана зарплата: 1999\$	Вік: 25 млрд \$	Стаж праці: 7 млрд \$	Стаж праці бажаної посади: 5	Бажана посада: Програміст	Iq тест: 97	Тест компанії: 77%	Оцінка інтерв'ю: 80%	1 експерт: <input type="text"/> 2 експерт: <input type="text"/>

Рисунок 30 – Відображення поля для кожного кандидата та кількість експертів та їх оцінок

При заповненні всіх полів для оцінок від експертів користувачу необхідно натиснути кнопку сортувати. Якщо один з експертів на якомусь кандидаті не вписав особисту оцінку, замість нього система оцінює його голос як 0.




Введіть результати експертів										
Сортувати										
1.	 Прохова А. Жінка	Очікувана зарплата: 1002\$	Вік: 22 млрд \$	Стаж праці: 4 млрд \$	Стаж праці бажаної посади: 0	Бажана посада: Програміст	Iq тест: 99	Тест компанії: 90%	Оцінка інтерв'ю: 77%	1 експерт: <input type="text" value="8"/> 2 експерт: <input type="text" value="2"/>
2.	 Бандера С. Чоловік	Очікувана зарплата: 500\$	Вік: 28 млрд \$	Стаж праці: 5 млрд \$	Стаж праці бажаної посади: 3	Бажана посада: Програміст	Iq тест: 110	Тест компанії: 90%	Оцінка інтерв'ю: 90%	1 експерт: <input type="text" value="8"/> 2 експерт: <input type="text" value="3"/>
3.	 Токарь А. Чоловік	Очікувана зарплата: 1999\$	Вік: 25 млрд \$	Стаж праці: 7 млрд \$	Стаж праці бажаної посади: 5	Бажана посада: Програміст	Iq тест: 97	Тест компанії: 77%	Оцінка інтерв'ю: 80%	1 експерт: <input type="text" value="12"/> 2 експерт: <input type="text"/>

Рисунок 31 – Заповненні поля балів експертів для кожного користувача

Після сортування користувачу відкривається вікно вже з відсортованими кандидатами та можливість завантажити список на свій ПК або додати з цього списку в свій особистий кабінет улюбленого кандидата.


Кандидати експертів												
Скачати список												
1.	 Токарь А. Чоловік	Очікувана зарплата: 1999\$	Вік: 25 років	Стаж праці: 7	Стаж праці бажаної посади: 5	Бажана посада: Програміст	Iq тест: 97	Тест компанії: 77%	Оцінка інтерв'ю: 80%	Середня оцінка експертів: 6	1 експерт: 12 2 експерт: 0	Додати в окремий список
2.	 Прохова А. Жінка	Очікувана зарплата: 1002\$	Вік: 22 років	Стаж праці: 4	Стаж праці бажаної посади: 0	Бажана посада: Програміст	Iq тест: 99	Тест компанії: 90%	Оцінка інтерв'ю: 77%	Середня оцінка експертів: 2.5	1 експерт: 3 2 експерт: 2	Додати в окремий список
3.	 Бандера С. Чоловік	Очікувана зарплата: 500\$	Вік: 28 років	Стаж праці: 5	Стаж праці бажаної посади: 3	Бажана посада: Програміст	Iq тест: 110	Тест компанії: 90%	Оцінка інтерв'ю: 90%	Середня оцінка експертів: 1.5	1 експерт: 0 2 експерт: 3	Додати в окремий список

Рисунок 32 – Відсортований список за результатами експертів

Після повного оцінювання експертів є можливість виявити найкращого кандидата по критеріям: Оцінка експертів, оцінка тесту компанії, оцінка тесту інтерв'ю, оцінка IQ тесту та стаж праці на обрану посаду.

Кандидати експертів														
Сховати список		Знайти найкращого												
1.	 Байдера С. Чоловик	Очікувана зарплата: 500\$	Вік: 28 років	ГОЛОВНА ОЦІНКА ПО ВСІМ КРИТЕРІЯМ: 68.5	Стаж праці: 5	Стаж праці бажаної посади: 3	Бажана посада: Програміст	Iq тест: 110	Тест компанії: 90%	Оцінка інтерв'ю: 90%	Середня оцінка експертів: 57	1 експерт: 61 2 експерт: 53	Додати в окремий список	Видалити кандидата взагалі ✖
2.	 Шевченко І. Чоловик	Очікувана зарплата: 2000\$	Вік: 45 років	ГОЛОВНА ОЦІНКА ПО ВСІМ КРИТЕРІЯМ: 50.3	Стаж праці: 10	Стаж праці бажаної посади: 9	Бажана посада: Керівник	Iq тест: 100	Тест компанії: 40%	Оцінка інтерв'ю: 70%	Середня оцінка експертів: 44	1 експерт: 62 2 експерт: 26	Додати в окремий список	Видалити кандидата взагалі ✖
3.	 Карпенко А. Жінка	Очікувана зарплата: 1500\$	Вік: 75 років	ГОЛОВНА ОЦІНКА ПО ВСІМ КРИТЕРІЯМ: 47.9	Стаж праці: 30	Стаж праці бажаної посади: 15	Бажана посада: Менеджер	Iq тест: 90	Тест компанії: 30%	Оцінка інтерв'ю: 60%	Середня оцінка експертів: 45.5	1 експерт: 60 2 експерт: 31	Додати в окремий список	Видалити кандидата взагалі ✖

Рисунок 33 – Відсортований список за всіма критеріями

3.4 Огляд процесу тестування

Останнім кроком оглянемо процес тестування системи визначення пріоритету кандидата. Для цього напишемо тести у вигляді так званих тест кейсів.

Таблиця 4

id	Пріоритет	Модуль	Кроки	Очікувані результати
1	найвищий	function login(user)	Можливість увійти в свій акаунт Кроки: 1. Ввести правильний логін 2. Ввести правильний пароль 3. Натиснути кнопку «Увійти»	1. Логін відображено у полі логіну 2. Пароль відображено як крапки у полі паролю 3. Користувач переходить на сторінку зі списками криптовалют 4. Відображає інформацію про користувача
2	високий	function login(user)	Перевірка невірно заповненого паролю для входу Кроки: 1. Ввести правильний логін 2. Ввести неправильний пароль 3. Натиснути кнопку «Увійти»	1. Логін відображено у полі логіну 2. Пароль відображено як крапки у полі паролю 3. Користувач залишається на початковій сторінці

Продовження таблиці 4

3	високий	function expertsList()	Можливість сортування особистого списку кандидатів Кроки: 1. У користувача присутні кандидати для сортування в особистому кабінеті 2. Вибір функції “Експерти” 3. Заповнення необхідних полів та вибір пункту “Аналіз мого списку?” 4. Натиснути кнопку “Поїхали”	1. Користувач отримує повідомлення в виді обраних людей та їх кількість та можливість записувати оцінки експертів 2. Користувач натискає сортувати 3. Користувачу видає відсортований список від експертів та їх середній бал по людині
4	високий	function expertsList()	Перевірка сортування пустого особистого списку людей Кроки: 1. У користувача відсутні люди для сортування в особистому кабінеті 2. Вибір функції “Експерти” 3. Заповнення необхідних полів та вибір пункту “Аналіз мого списку?” 4. Натиснути кнопку “Поїхали”	1. Користувач отримує повідомлення в виді пустого повідомлення без жодного кандидата

3.5 Висновки до третього розділу

Для безпосередньої реалізації додатку визначення пріоритету кандидата на роботу було спроектовано додаток на базі минулих спроектованих архітектур та бази даних.

Початок шляху вибір мови програмування та застосунків реалізації матеріалу.

Було визначено структуру програмного застосунку, побудовано її у вигляді схеми та описано специфікацію програмних модулів у вигляді таблиці. Також було представлено. Заключним етапом був огляд процесу тестування створення додатку для сортування списку кандидатів на роботу завдяки

оцінкам експертів, а саме – було створено ряд тест кейсів для перевірки функціоналу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Лезина З.М. “Манипулирование выбором вариантов (теория агенды)”, Автомат. и телемех., 1985, № 4, С.5–22.
2. Волошин О.Ф., Машенко С.О. Теорія прийняття рішень: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2006. – 304 с.
3. Гнатієнко Г.М., Снитюк В.Є. Експертні технології прийняття рішень: Монографія. – К.: ТОВ «Маклаут», 2008. – 444с.
4. Макаров И.М., Виноградская Т.М., Рубчинский А.А. и др. Теория выбора и принятия решений: Учебное пособие. – М.:Наука, 1982. – 328 с.
5. Михалевич В.С., Волкович В.Л. Вычислительные методы исследования и проектирования сложных систем. – М.: Наука, 1982. – 286 с.
6. Додонов А.Г., Ландэ Д.В., Цыганок В.В., Андрейчук О.В., Каденко С.В, Грайворонская А.Н. – К.: ООО «Инжиниринг», 2017. – 282 с.
7. https://uk.wikipedia.org/wiki/Система_підтримки_рішень
8. Ларичев О. И., Петровский А. В. Системы поддержки принятия решений. Современное состояние и перспективы их развития. // Итоги науки и техники. Сер. Техническая кибернетика. — Т.21. М.: ВИНТИ, 1987.
9. Сараев А. Д., Щербина О. А. Системный анализ и современные информационные технологии //Труды Крымской Академии наук. — Симферополь: СОНАТ, 2006. — С. 47-59.
10. Alter S. L. Decision support systems: current practice and continuing challenges. Reading, Mass.: Addison-Wesley Pub., 1980.
11. Bonczek R.H., Holsapple C., Whinston A.B. Foundations of Decision Support Systems.- New York: Academic Press, , 1981.

12. Davis G. Management Information Systems: Conceptual Foundations, Structure, and Development. — New York: McGraw-Hill, 1974.
13. Druzdel M. J., Flynn R. R. Decision Support Systems. Encyclopedia of Library and Information Science. — A. Kent, Marcel Dekker, Inc., 1999.
14. Edwards J.S. Expert Systems in Management and Administration — Are they really different from Decision Support Systems? // European Journal of Operational Research, 1992. — Vol. 61. — pp. 114—121.
15. Eom H., Lee S. Decision Support Systems Applications Research: A Bibliography (1971—1988) // European Journal of Operational Research, 1990. — N 46. — pp. 333—342.

ДОДАТКИ

```
var user = [  
  {  
    name: "User0",  
    login: "0",  
    password: "0",  
  },  
  {  
    name: "User1",  
    login: "1",  
    password: "1",  
  },  
  {  
    name: "User2",  
    login: "2",  
    password: "2",  
  }  
]  
  
var coin = [  
  {  
    id: 0,  
    fullName: "Бандера С.",  
    abbreviation: "Чоловік",  
    img: "img/img1.png",  
    price: 500,  
    marketCap: 28,  
    vol: 5,  
    yearJob: 3,  
    jobTitle: "Программіст",  
    iqTest: 110,  
    test: 90,  
    interview: 90,  
  },  
  {  
    id: 1,  
    fullName: "Шевченко І.",  
    abbreviation: "Чоловік",  
    img: "img/img2.png",  
    price: 2000,  
  }  
]
```

```
marketCap: 45,  
vol: 10,  
yearJob: 9,  
jobTitle: "Керівник",  
iqTest: 100,  
test: 40,  
interview: 70,  
},  
{  
id: 2,  
fullName: "Карпенко А.",  
abbreviation: "Жінка",  
img: "img/img3.png",  
price: 1500,  
marketCap: 75,  
vol: 30,  
yearJob: 15,  
jobTitle: "Менеджер",  
iqTest: 90,  
test: 30,  
interview: 60,  
},  
{  
id: 3,  
fullName: "Гамоцька С.",  
abbreviation: "Жінка",  
img: "img/img4.png",  
price: 2800,  
marketCap: 40,  
vol: 20,  
yearJob: 9,  
jobTitle: "Программіст",  
iqTest: 70,  
test: 60,  
interview: 77,  
},  
{  
id: 4,  
fullName: "Стус В.",  
abbreviation: "Чоловік",  
img: "img/img5.png",
```

```
price: 300,  
marketCap: 21,  
vol: 0,  
yearJob: 0,  
jobTitle: "Менеджер",  
iqTest: 100,  
test: 100,  
interview: 50,  
},  
{  
  
id: 5,  
fullName: "Шевченко В.",  
abbreviation: "Чоловік",  
img: "img/img6.png",  
price: 1200,  
marketCap: 25,  
vol: 5,  
yearJob: 1,  
jobTitle: "Менеджер",  
iqTest: 99,  
test: 60,  
interview: 0,  
},  
{  
  
id: 6,  
fullName: "Карпенко А.",  
abbreviation: "Чоловік",  
img: "img/img7.png",  
price: 10,  
marketCap: 15,  
vol: 0,  
yearJob: 0,  
jobTitle: "Керівник",  
iqTest: 87,  
test: 0,  
interview: 0,  
},  
{  
  
id: 7,  
fullName: "Пугачева А.",  
abbreviation: "Жінка",
```

```
    img: "img/img8.png",
    price: 1200,
    marketCap: 18,
    vol: 0,
    yearJob: 0,
    jobTitle: "Логіст",
    iqTest: 86,
    test: 50,
    interview: 10,
  },
  {
    id: 8,
    fullName: "Гонгало Ф.",
    abbreviation: "Жінка",
    img: "img/img9.png",
    price: 4000,
    marketCap: 45,
    vol: 27,
    yearJob: 20,
    jobTitle: "Керівник",
    iqTest: 92,
    test: 100,
    interview: 50,
  },
  {
    id: 9,
    fullName: "Коваленко О.",
    abbreviation: "Чоловік",
    img: "img/img10.png",
    price: 500,
    marketCap: 21,
    vol: 3,
    yearJob: 1,
    jobTitle: "Керівник",
    iqTest: 96,
    test: 67,
    interview: 100,
  },
  {
    id: 10,
    fullName: "Токарь А.",
```

```
        abbreviation: "Чоловік",
        img: "img/img11.png",
        price: 1999,
        marketCap: 25,
        vol: 7,
        yearJob: 5,
        jobTitle: "Программіст",
        iqTest: 97,
        test: 77,
        interview: 80,
    },
    {
        id: 11,
        fullName: "Прохова А.",
        abbreviation: "Жінка",
        img: "img/img12.png",
        price: 1002,
        marketCap: 22,
        vol: 4,
        yearJob: 0,
        jobTitle: "Программіст",
        iqTest: 99,
        test: 90,
        interview: 77,
    },
]

var root = document.querySelector("#AllList")

function loginRender() {
    const divLogin = document.createElement("div")
    divLogin.className = "divLogin"

    const title = document.createElement("h2")
    title.innerHTML = "Введіть логін та пароль"

    const login = document.createElement("input")
    login.className = "login"
    login.setAttribute("placeholder", "Логін")
```

```

const password = document.createElement("input")
password.className = "password"
password.setAttribute("placeholder", "Пароль")
password.setAttribute("type", "password")

const button = document.createElement("button")
button.innerHTML = "Увійти"
button.setAttribute("onclick", "login(user)")

divLogin.append(title)
divLogin.append(login)
divLogin.append(password)
divLogin.append(button)

root.append(divLogin)
}

function login(user) {
const login = document.querySelector(".login").value
const password = document.querySelector(".password").value

for (let i = 0; i < user.length; i++) {
    if (user[i].login == login && user[i].password == password) {
        const divLogin = document.querySelector(".divLogin")

        const headerUser = document.createElement("h1")
        headerUser.innerHTML = user[i].name
        root.append(headerUser)

        divLogin.remove()
        buttonsFunc()
        buildCoinList(coin)
    }
}
}

var myLoveCoin = [

]

function buildCoinList(coinList, type, job) {

```

```

const searchCoinList = document.querySelector(".containerCoinList")
if (searchCoinList) searchCoinList.remove()

const containerCoinList = document.createElement("div")
containerCoinList.className = "containerCoinList"

var titileFirstCoin = document.createElement("h2")

if (type == "mylist") {

    titileFirstCoin.innerHTML = "Окремий список кандидатів"
    titileFirstCoin.style.color = "green"

    if (myLoveCoin.length == 0) {
        {
            const titleNoCoin = document.createElement("h3")
            titleNoCoin.innerHTML = "Ви не вибрали жодного кандидата!"
            titleNoCoin.style.color = "red"
            containerCoinList.append(titleNoCoin)
        }
    }
}
else if (type == "search") {
    titileFirstCoin.innerHTML = `Всі кандидати з бажаною посадою - ${job}`
}
else {
    titileFirstCoin.innerHTML = "Всі кандидати"
}
containerCoinList.append(titileFirstCoin)

root.append(containerCoinList)
for (let i = 0; i < coinList.length; i++) {
    const coinDiv = document.createElement("div")
    coinDiv.className = "coinDiv"

    const leftDiv = document.createElement("div")
    leftDiv.className = "leftDiv"

```

```
const img = document.createElement("img")
img.className = "img_coin"
img.setAttribute("src", coinList[i].img)
img.setAttribute("alt", "Фото монети")

const paragraphNo = document.createElement("h3")
paragraphNo.innerHTML = `${i + 1}`

const fullNameAndAbbreviation = document.createElement("div")
fullNameAndAbbreviation.className = "fullNameAndAbbreviation"

const name = document.createElement("h3")
name.innerHTML = coinList[i].fullName

const abbreviation = document.createElement("p")
abbreviation.innerHTML = coinList[i].abbreviation

fullNameAndAbbreviation.append(name)
fullNameAndAbbreviation.append(abbreviation)

const price = document.createElement("p")
price.innerHTML = `Очікувана зарплата: <b>${coinList[i].price}</b>`

const marketCap = document.createElement("p")
marketCap.innerHTML = `Вік: <b>${coinList[i].marketCap}</b> років`

const vol = document.createElement("p")
vol.innerHTML = `Стаж праці: <b>${coinList[i].vol}</b>`

const yearJob = document.createElement("p")
yearJob.innerHTML = `Стаж праці бажаної посади: <b>${coinList[i].yearJob}</b>`

const jobTitle = document.createElement("p")
jobTitle.innerHTML = `Бажана посада: <b>${coinList[i].jobTitle}</b>`

const iqTest = document.createElement("p")
iqTest.innerHTML = `Іq тест: <b>${coinList[i].iqTest}</b>`

const test = document.createElement("p")
```

```

test.innerHTML = `Тест компанії: <b>${coinList[i].test}%</b>`

const interview = document.createElement("p")
interview.innerHTML = `Оцінка інтерв'ю: <b>${coinList[i].interview}%</b>`

const buttonLove = document.createElement("button")

if (type === "mylist") {
    buttonLove.innerHTML = "Видалити зі списку ✕ "
    buttonLove.style.background = "black"
    buttonLove.style.color = "white"
    buttonLove.setAttribute("onclick", `delCoin(${coinList[i].id})`)
}
else {
    buttonLove.innerHTML = "Додати в окремий список"

    buttonLove.setAttribute("onclick", `addCoin(${coinList[i].id})`)

    var buttonDel = document.createElement("button")
    buttonDel.innerHTML = "Видалити кандидата ✕ "
    buttonDel.setAttribute("onclick", `deletUser(${coinList[i].id})`)
}

leftDiv.append(paragrafNo)
leftDiv.append(img)
leftDiv.append(fullNameAndAbbreviation)
leftDiv.append(price)
leftDiv.append(marketCap)
leftDiv.append(vol)
leftDiv.append(yearJob)
leftDiv.append(jobTitle)
leftDiv.append(iqTest)
leftDiv.append(test)
leftDiv.append(interview)

```

```

    leftDiv.append(buttonLove)
    if (!type) leftDiv.append(buttonDel)

    coinDiv.append(leftDiv)
    containerCoinList.append(coinDiv)
  }
}
function buttonsFunc() {
  const divButtons = document.createElement("div")
  divButtons.className = "divButtons"

  const titleButton = document.createElement("h3")
  titleButton.innerHTML = "Відсортувати за:"

  const buttonName = document.createElement("button")
  buttonName.innerHTML = "Прізвищем"
  buttonName.setAttribute("onclick", "nameCoin(coin)")

  const buttonPrise = document.createElement("button")
  buttonPrise.innerHTML = "Очікувана зарплата 📊"
  buttonPrise.setAttribute("onclick", "price(coin)")

  const buttonPriseCap = document.createElement("button")
  buttonPriseCap.innerHTML = "Віком 📅"
  buttonPriseCap.setAttribute("onclick", "priseCap(coin)")

  const buttonVol = document.createElement("button")
  buttonVol.innerHTML = "Стаж праці 📅"
  buttonVol.setAttribute("onclick", "volCoin(coin)")

  const buttonyearJob = document.createElement("button")
  buttonyearJob.innerHTML = "Стаж праці обраної посади 📅"
  buttonyearJob.setAttribute("onclick", "yearJob(coin)")

  const buttoniqTest = document.createElement("button")
  buttoniqTest.innerHTML = "Іq тест 📊"
  buttoniqTest.setAttribute("onclick", "iqTest(coin)")

  const buttontest = document.createElement("button")

```

```

buttontest.innerHTML = "Тест компанії ✎"
buttontest.setAttribute("onclick", "test(coin)")

const buttoninterview = document.createElement("button")
buttoninterview.innerHTML = "Оцінка інтерв'ю ✎"
buttoninterview.setAttribute("onclick", "interview(coin)")

const buttonJtitle = document.createElement("button")
buttonJtitle.innerHTML = "Пошук за посадою 🔍"
buttonJtitle.setAttribute("onclick", "Jtitle(coin)")

const buttonAdd = document.createElement("button")
buttonAdd.innerHTML = "Додати кандидата ➕"
buttonAdd.setAttribute("onclick", "addDiv()")

const buttonMyList = document.createElement("button")
buttonMyList.innerHTML = "Список окремих"
buttonMyList.style.background = "pink"

buttonMyList.setAttribute("onclick", "buildCoinList(myLoveCoin, 'mylist')")

const buttonPro = document.createElement("button")
buttonPro.innerHTML = "Експерти 🧑‍💻"
buttonPro.style.background = "black"
buttonPro.style.color = "white"
buttonPro.setAttribute("onclick", "expertsDiv()")

divButtons.append(titleButton)
divButtons.append(buttonName)
divButtons.append(buttonPrise)
divButtons.append(buttonPriseCap)
divButtons.append(buttonVol)
divButtons.append(buttonyearJob)
divButtons.append(buttoniqTest)
divButtons.append(buttontest)
divButtons.append(buttoninterview)
divButtons.append(buttonJtitle)
divButtons.append(buttonAdd)

```

```

divButtons.append(buttonMyList)
divButtons.append(buttonPro)

root.append(divButtons)
}
var pr = 0
function price(coin) {

if (pr == 0) {
    const arr = coin
    buildCoinList(arr.sort((a, b) => b.price - a.price))
    pr++
}
else {
    const arr = coin;
    buildCoinList(arr.sort((a, b) => a.price - b.price))
    pr--
}
}
var cp = 0
function priseCap(coin) {

if (cp == 0) {
    const arr = coin;
    buildCoinList(arr.sort((a, b) => b.marketCap - a.marketCap))
    cp++
}
else {
    const arr = coin;
    buildCoinList(arr.sort((a, b) => a.marketCap - b.marketCap))
    cp--
}
}
var na = 0
function nameCoin(coin) {
if (na == 0) {
    const arr = coin.sort((a, b) => a.fullName > b.fullName ? 1 : -1)
    buildCoinList(arr)
    na++
}
}

```

```

else {
    const arr = coin.sort((a, b) => a.fullName < b.fullName ? 1 : -1)
    buildCoinList(arr)
    na--
}
}

```

```

var yj = 0
function yearJob(coin) {
    if (yj == 0) {
        const arr = coin.sort((a, b) => a.yearJob > b.yearJob ? 1 : -1)
        buildCoinList(arr)
        yj++
    }
    else {
        const arr = coin.sort((a, b) => a.yearJob < b.yearJob ? 1 : -1)
        buildCoinList(arr)
        yj--
    }
}
}

```

```

var iT = 0
function iqTest(coin) {
    if (iT == 0) {
        const arr = coin.sort((a, b) => a.iqTest > b.iqTest ? 1 : -1)
        buildCoinList(arr)
        iT++
    }
    else {
        const arr = coin.sort((a, b) => a.iqTest < b.iqTest ? 1 : -1)
        buildCoinList(arr)
        iT--
    }
}
}

```

```

var t = 0
function test(coin) {
    if (t == 0) {
        const arr = coin.sort((a, b) => a.test > b.test ? 1 : -1)
        buildCoinList(arr)
    }
}

```

```

        t++
    }
    else {
        const arr = coin.sort((a, b) => a.test < b.test ? 1 : -1)
        buildCoinList(arr)
        t--
    }
}

var inter = 0
function interview(coin) {
    if (inter == 0) {
        const arr = coin.sort((a, b) => a.interview > b.interview ? 1 : -1)
        buildCoinList(arr)
        inter++
    }
    else {
        const arr = coin.sort((a, b) => a.interview < b.interview ? 1 : -1)
        buildCoinList(arr)
        inter--
    }
}

```

```

function Jtitle(coin) {

    const listJob = []

    const job = prompt("Введіть необхідну посаду")

    if (job) {

        for (let i = 0; i < coin.length; i++) {
            if (coin[i].jobTitle == job) {
                listJob[listJob.length] = coin[i]
            }
        }
    }
}

```

```
        else if (i == coin.length - 1 && listJob.length == 0) {
            alert("Така посада відсутня")
        }
        buildCoinList(listJob, "search", job)
    }
}

}
```

```
function addDiv() {
```

```
    addDivRemove()
```

```
    const divAdd = document.createElement("div")
    divAdd.className = "divAdd"
```

```
    const inputName = document.createElement("input")
    inputName.className = "fullName"
    inputName.setAttribute("placeholder", "ПІБ")
```

```
    const inputabbreviation = document.createElement("input")
    inputabbreviation.className = "abbreviation"
    inputabbreviation.setAttribute("placeholder", "Стать")
```

```
    const inpuprice = document.createElement("input")
    inpuprice.className = "price"
    inpuprice.setAttribute("placeholder", "Очікувана ЗП")
    inpuprice.setAttribute("type", "number")
```

```
    const inputAge = document.createElement("input")
    inputAge.className = "marketCap"
    inputAge.setAttribute("placeholder", "Вік")
    inputAge.setAttribute("type", "number")
```

```
    const inputAgeJ = document.createElement("input")
```

```
inputAgeJ.className = "vol"
inputAgeJ.setAttribute("placeholder", "Стаж праці")
inputAgeJ.setAttribute("type", "number")

const inputAgeJs = document.createElement("input")
inputAgeJs.className = "yearJob"
inputAgeJs.setAttribute("placeholder", "Стаж праці обраної посади")
inputAgeJs.setAttribute("type", "number")

const inputjobTitle = document.createElement("input")
inputjobTitle.className = "jobTitle"
inputjobTitle.setAttribute("placeholder", "Посада")

const inputiqTest = document.createElement("input")
inputiqTest.className = "iqTest"
inputiqTest.setAttribute("placeholder", "Результат Іq тесту")
inputiqTest.setAttribute("type", "number")

const inputTestT = document.createElement("input")
inputTestT.className = "test"
inputTestT.setAttribute("placeholder", "Результат тесту компанії")
inputTestT.setAttribute("type", "number")

const inputinterview = document.createElement("input")
inputinterview.className = "interview "
inputinterview.setAttribute("placeholder", "Результат тесту інтерв'ю")
inputinterview.setAttribute("type", "number")

divAdd.append(inputName)
divAdd.append(inputabbreviation)
divAdd.append(inpuprice)
divAdd.append(inputAge)
divAdd.append(inputAgeJ)
divAdd.append(inputjobTitle)
divAdd.append(inputAgeJs)
divAdd.append(inputiqTest)
divAdd.append(inputTestT)
divAdd.append(inputinterview)

const buttonAdd = document.createElement("button")
buttonAdd.innerHTML = "Додати кандидата"
```

```

buttonAdd.setAttribute("onclick", "add()")

const buttonClose = document.createElement("button")
buttonClose.innerHTML = "Закрити вікно"
buttonClose.setAttribute("onclick", "addDivRemove()")

divAdd.append(buttonAdd)
divAdd.append(buttonClose)

root.append(divAdd)

}

function addDivRemove() {
divAddSearch = document.querySelector(".divAdd")
if (divAddSearch) divAddSearch.remove()
}

function add() {

const fullName = document.querySelector(".fullName").value
const abbreviation = document.querySelector(".abbreviation").value
const price = document.querySelector(".price").value
const marketCap = document.querySelector(".marketCap").value
const vol = document.querySelector(".vol").value
const yearJob = document.querySelector(".yearJob").value
const jobTitle = document.querySelector(".jobTitle").value
const iqTest = document.querySelector(".iqTest").value
const test = document.querySelector(".test").value
const interview = document.querySelector(".interview").value

if (fullName && abbreviation && price && marketCap && vol && yearJob && jobTitle && iqTest &&
test && interview) {

    coin[coin.length] = {
        id: coin.length,
        fullName: fullName,
        abbreviation: abbreviation,
        img: "img/img1.png",
        price: price,
        marketCap: marketCap,

```

```

        vol: vol,
        yearJob: yearJob,
        jobTitle: jobTitle,
        iqTest: iqTest,
        test: test,
        interview: interview,
    }

    buildCoinList(coin)
    addDivRemove()
}
else alert("Введіть всі поля")
}

function deleteUser(id) {
    for (let i = 0; i < coin.length; i++) {
        if (coin[i].id === id) {

            coin = coin.filter((n) => { return n.id !== id });

        }
    }
    delCoin(id)
    buildCoinList(coin)
}

var vol = 0
function volCoin(coin) {
    if (!vol === 0) {
        const arr = coin.sort((a, b) => a.vol > b.vol ? 1 : -1)
        buildCoinList(arr)
        vol++
    }
    else {
        const arr = coin.sort((a, b) => a.vol < b.vol ? 1 : -1)
        buildCoinList(arr)
        vol--
    }
}
}

function addCoin(id) {

```

```

for (let i = 0; i < coin.length; i++) {
    if (coin[i].id === id) myLoveCoin[myLoveCoin.length] = coin[i]
}


```

```

myLoveCoin = myLoveCoin.reduce((o, i) => {
    if (!o.find(v => v.id === i.id)) {
        o.push(i);
    }
    return o;
}, []);

```

```

const love = document.createElement("p")
love.className = "love"
love.innerHTML = "Додано до списку  "
root.append(love)

```

```

setTimeout(() => {
    const love = document.querySelector(".love")
    love.remove()
}, 1500);

```

```

}
function delCoin(id) {


```

```

    var idToDelete = id
    myLoveCoin.forEach(function (el, i) {
        if (el.id === idToDelete) myLoveCoin.splice(i, 1)
    })

```

```

    buildCoinList(myLoveCoin, 'mylist')
    const del = document.createElement("p")
    del.className = "del"
    del.innerHTML = "Видалено  "
    root.append(del)

```

```

setTimeout(() => {
    const del = document.querySelector(".del")
    del.remove()

}, 1500);

}

var ex = 0
function expertsDiv() {

if (ex === 1) {
    const searchExpertDiv = document.querySelector(".expertsDiv")
    searchExpertDiv.remove()
    ex--
} else {
    const expertsDiv = document.createElement("div")
    expertsDiv.className = "expertsDiv"

    const inputExperts = document.createElement("input")
    inputExperts.className = "inputExperts"
    inputExperts.setAttribute("placeholder", "Кількість експертів")
    inputExperts.setAttribute("type", "number")
    inputExperts.setAttribute("min", "1")

    const paragraphBor = document.createElement("p")
    paragraphBor.innerHTML = "Експерти боти? 

    const inputBot = document.createElement("input")
    inputBot.className = "inputBot"
    inputBot.setAttribute("type", "checkbox")
    paragraphBor.append(inputBot)

    const paragraphMyList = document.createElement("p")
    paragraphMyList.innerHTML = "Аналіз окремого списку?"
    const inputMylist = document.createElement("input")
    inputMylist.className = "inputMylist"

    inputMylist.setAttribute("type", "checkbox")
    paragraphMyList.append(inputMylist)

```

```

const inputNoCoin = document.createElement("input")
inputNoCoin.className = "inputNoCoin"
inputNoCoin.setAttribute("placeholder", "Кількість кандидатів")
inputNoCoin.setAttribute("type", "number")
inputNoCoin.setAttribute("min", "2")

const buttonStart = document.createElement("button")
buttonStart.innerHTML = "Поїхали 🚀"
buttonStart.style.background = "black"
buttonStart.style.color = "red"
buttonStart.setAttribute("onclick", "expertsList()")

expertsDiv.append(inputExperts)
expertsDiv.append(inputNoCoin)
expertsDiv.append(paragrafBor)
expertsDiv.append(paragrafMyList)

expertsDiv.append(buttonStart)

root.append(expertsDiv)
ex++
}
}

function expertsList() {

const expertsAmount = Number(document.querySelector(".inputExperts").value)
const coinAmount = Number(document.querySelector(".inputNoCoin").value)
const inputBot = document.querySelector(".inputBot")
const myList = document.querySelector(".inputMylist")

var newArrCoins = []

if (inputBot.checked) {

    if (!myList.checked) {

```

```

for (let i = 0; i < coinAmount; i++) {
    newArrCoins[i] = coin[i]
    newArrCoins[i].gredeExpert = []
    for (let j = 0; j < expertsAmount; j++) {

        const ball = getRandomInt(100)
        newArrCoins[i].gredeExpert[j] = ball

    }

}

for (let i = 0; i < newArrCoins.length; i++) {

    const average = averageCost(newArrCoins[i].gredeExpert)
    newArrCoins[i].average = Number(average.toFixed(1))

}

newArrCoins.sort((a, b) => b.average - a.average)
buildNewCoinList(newArrCoins, "bot")
expertsDiv()
}
else {
for (let i = 0; i < coinAmount; i++) {
    newArrCoins[i] = myLoveCoin[i]
    newArrCoins[i].gredeExpert = []
    for (let j = 0; j < expertsAmount; j++) {

        const ball = getRandomInt(100)
        newArrCoins[i].gredeExpert[j] = ball

    }

}

for (let i = 0; i < newArrCoins.length; i++) {

    const average = averageCost(newArrCoins[i].gredeExpert)
    newArrCoins[i].average = Number(average.toFixed(1))

}

newArrCoins.sort((a, b) => b.average - a.average)
buildNewCoinList(newArrCoins, "bot")

```

```

        expertsDiv()
    }
}
else {

    if (!myList.checked) for (let i = 0; i < coinAmount; i++) newArrCoins[i] = coin[i]
    else for (let i = 0; i < coinAmount; i++) newArrCoins[i] = myLoveCoin[i]

    buildInnerList(newArrCoins, expertsAmount)
    expertsDiv()
}
}

function getRandomInt(max) {
return Math.floor(Math.random() * max);
}
function averageCost(arr) {
return arr.reduce((partial_sum, a) => partial_sum + a, 0) / arr.length;
}
var dowCoin
function buildNewCoinList(coinList, type) {
dowCoin = coinList

const ExpertList = document.querySelector("#ExpertList")

const searchCoinList = document.querySelector(".containerCoinListExperts")
if (searchCoinList) searchCoinList.remove()

const containerCoinList = document.createElement("div")
containerCoinList.className = "containerCoinListExperts"

const buttonDownload = document.createElement("button")
buttonDownload.innerHTML = "Скачати список"
buttonDownload.setAttribute("onclick", "downloadList('Експерти',dowCoin)")

const buttonBest = document.createElement("button")
buttonBest.innerHTML = "Знайти найращого"
buttonBest.setAttribute("onclick", "theBest(dowCoin)")

```

```

var titileFirstCoin = document.createElement("h2")
if (type == "bot") titileFirstCoin.innerHTML = "Кандидати експертів(БОТИ) 🤖"
else titileFirstCoin.innerHTML = "Кандидати експертів"

containerCoinList.append(titileFirstCoin)
containerCoinList.append(buttonDonwload)
containerCoinList.append(buttonBest)
ExpertList.append(containerCoinList)
for (let i = 0; i < coinList.length; i++) {
    const coinDiv = document.createElement("div")
    coinDiv.className = "coinDiv"

    const leftDiv = document.createElement("div")
    leftDiv.className = "leftDiv"

    const img = document.createElement("img")
    img.className = "img_coin"
    img.setAttribute("src", coinList[i].img)
    img.setAttribute("alt", "Фото монети")

    const paragrafNo = document.createElement("h3")
    paragrafNo.innerHTML = `${i + 1}.`

    const fullNameAndAbbreviation = document.createElement("div")
    fullNameAndAbbreviation.className = "fullNameAndAbbreviation"

    const name = document.createElement("h3")
    name.innerHTML = coinList[i].fullName

    const abbreviation = document.createElement("p")
    abbreviation.innerHTML = coinList[i].abbreviation

    fullNameAndAbbreviation.append(name)
    fullNameAndAbbreviation.append(abbreviation)

    const price = document.createElement("p")
    price.innerHTML = `Очікувана зарплата: <b>${coinList[i].price}</b>`

```

```

if (type == "best") {
    var bestStore = document.createElement("p")
    bestStore.style.color = "red"
    bestStore.innerHTML = `ГОЛОВНА ОЦІНКА ПО ВСІМ КРИТЕРІЯМ:
<b>${coinList[i].bestStore}</b>`
}

const marketCap = document.createElement("p")
marketCap.innerHTML = `Вік: <b>${coinList[i].marketCap}</b> років</b>`

const vol = document.createElement("p")
vol.innerHTML = `Стаж праці: <b>${coinList[i].vol}</b>`

const yearJob = document.createElement("p")
yearJob.innerHTML = `Стаж праці бажаної посади: <b>${coinList[i].yearJob}</b>`

const jobTitle = document.createElement("p")
jobTitle.innerHTML = `Бажана посада: <b>${coinList[i].jobTitle}</b>`

const iqTest = document.createElement("p")
iqTest.innerHTML = `Іq тест: <b>${coinList[i].iqTest}</b>`

const test = document.createElement("p")
test.innerHTML = `Тест компанії: <b>${coinList[i].test}%</b>`

const interview = document.createElement("p")
interview.innerHTML = `Оцінка інтерв'ю: <b>${coinList[i].interview}%</b>`

const average = document.createElement("p")
average.innerHTML = `Середня оцінка експертів: <b>${coinList[i].average}</b>`

const listexp = document.createElement("div")
listexp.className = "listexp"
for (let j = 0; j < coinList[i].gredeExpert.length; j++) {
    const averageEx = document.createElement("p")
    averageEx.innerHTML = `${j + 1} експерт: <b>${coinList[i].gredeExpert[j]}</b>`
    listexp.append(averageEx)
}

const buttonLove = document.createElement("button")

```

```

buttonLove.innerHTML = "Додати в улюблені"
buttonLove.innerHTML = "Додати в окремий список"
buttonLove.style.background = "pink"
buttonLove.setAttribute("onclick", `addCoin(${coinList[i].id})`)

const buttonDel = document.createElement("button")
buttonDel.innerHTML = "Видалити кандидата взагалі ✕"
buttonDel.setAttribute("onclick", `deletUser(${coinList[i].id})`)

leftDiv.append(paragrafNo)
leftDiv.append(img)
leftDiv.append(fullNameAndAbbreviation)
leftDiv.append(price)
leftDiv.append(marketCap)
if (type == "best") leftDiv.append(bestStore)
leftDiv.append(vol)
leftDiv.append(yearJob)
leftDiv.append(jobTitle)
leftDiv.append(iqTest)
leftDiv.append(test)
leftDiv.append(interview)
leftDiv.append(average)
leftDiv.append(listexp)
leftDiv.append(buttonLove)
leftDiv.append(buttonDel)

coinDiv.append(leftDiv)
containerCoinList.append(coinDiv)
}
}

function theBest(dowCoin) {

const newBestList = []

for (let i = 0; i < dowCoin.length; i++) {
    const valueList = rounded(dowCoin[i].average * 0.4 + dowCoin[i].yearJob * 0.3 + dowCoin[i].test
* 0.2 + dowCoin[i].interview * 0.2 + dowCoin[i].iqTest * 0.08)

```

```

newBestList[newBestList.length] = { bestStore: valueList, ...dowCoin[i] }

newBestList.sort((a, b) => b.bestStore - a.bestStore)
}

buildNewCoinList(newBestList, "best")

}

var rounded = function (number) {
return +number.toFixed(2);
}

var newCoinListWar
var NoEx
function buildInnerList(coinList, experts) {

const ExpertList = document.querySelector("#ExpertList")

newCoinListWar = coinList
NoEx = experts
const searchCoinList = document.querySelector(".containerCoinListExperts")
if (searchCoinList) searchCoinList.remove()

const containerCoinList = document.createElement("div")
containerCoinList.className = "containerCoinListExperts"

const buttonNext = document.createElement("button")
buttonNext.className = "buttonNext"
buttonNext.innerHTML = "Сортувати"
buttonNext.setAttribute("onclick", `valueExp(newCoinListWar, NoEx)`)

var titileFirstCoin = document.createElement("h2")
titileFirstCoin.innerHTML = "Введіть результати експертів"

containerCoinList.append(titileFirstCoin)

```

```
containerCoinList.append(buttonNext)
```

```
ExpertList.append(containerCoinList)
```

```
for (let i = 0; i < coinList.length; i++) {
```

```
    const coinDiv = document.createElement("div")
```

```
    coinDiv.className = "coinDiv"
```

```
    const leftDiv = document.createElement("div")
```

```
    leftDiv.className = "leftDiv"
```

```
    const img = document.createElement("img")
```

```
    img.className = "img_coin"
```

```
    img.setAttribute("src", coinList[i].img)
```

```
    img.setAttribute("alt", "Фото монети")
```

```
    const paragrafNo = document.createElement("h3")
```

```
    paragrafNo.innerHTML = `${i + 1}`
```

```
    const fullNameAndAbbreviation = document.createElement("div")
```

```
    fullNameAndAbbreviation.className = "fullNameAndAbbreviation"
```

```
    const name = document.createElement("h3")
```

```
    name.innerHTML = coinList[i].fullName
```

```
    const abbreviation = document.createElement("p")
```

```
    abbreviation.innerHTML = coinList[i].abbreviation
```

```
    fullNameAndAbbreviation.append(name)
```

```
    fullNameAndAbbreviation.append(abbreviation)
```

```
    const price = document.createElement("p")
```

```
    price.innerHTML = `Очікувана зарплата: <b>${coinList[i].price}</b>`
```

```
    const marketCap = document.createElement("p")
```

```
    marketCap.innerHTML = `Вік: <b>${coinList[i].marketCap}</b> млрд $</b>`
```

```
    const vol = document.createElement("p")
```

```
    vol.innerHTML = `Стаж праці: <b>${coinList[i].vol}</b> млрд $</b>`
```

```

const yearJob = document.createElement("p")
yearJob.innerHTML = `Стаж праці бажаної посади: <b>${coinList[i].yearJob}</b>`

const jobTitle = document.createElement("p")
jobTitle.innerHTML = `Бажана посада: <b>${coinList[i].jobTitle}</b>`

const iqTest = document.createElement("p")
iqTest.innerHTML = `Іq тест: <b>${coinList[i].iqTest}</b>`

const test = document.createElement("p")
test.innerHTML = `Тест компанії: <b>${coinList[i].test}%</b>`

const interview = document.createElement("p")
interview.innerHTML = `Оцінка інтерв'ю: <b>${coinList[i].interview}%</b>`

const listexp = document.createElement("div")
listexp.className = "listexp"
for (let j = 0; j < experts; j++) {
    const averageEx = document.createElement("p")

    const innerEx = document.createElement("input")
    innerEx.className = "innerEx"
    innerEx.setAttribute("id", `ex${i}`)
    innerEx.setAttribute("type", "number")
    innerEx.setAttribute("min", "1")
    innerEx.setAttribute("max", "100")
    averageEx.innerHTML = `${j + 1} експерт:`

    listexp.append(averageEx)
    listexp.append(innerEx)
}

leftDiv.append(paragrafNo)
leftDiv.append(img)

```

```

    leftDiv.append(fullNameAndAbbreviation)
    leftDiv.append(price)
    leftDiv.append(marketCap)
    leftDiv.append(vol)
    leftDiv.append(yearJob)
    leftDiv.append(jobTitle)
    leftDiv.append(iqTest)
    leftDiv.append(test)
    leftDiv.append(interview)

    leftDiv.append(listexp)

    coinDiv.append(leftDiv)

    containerCoinList.append(coinDiv)
}

}

function valueExp(coinlist, NoExp) {
  for (let i = 0; i < coinlist.length; i++) {
    const expValue = document.querySelectorAll(`#ex${i}`)
    coinlist[i].gredeExpert = []
    for (let j = 0; j < expValue.length; j++) {
      coinlist[i].gredeExpert[j] = Number(expValue[j].value)
    }
  }
}

for (let i = 0; i < coinlist.length; i++) {
  const average = averageCost(coinlist[i].gredeExpert)
  coinlist[i].average = Number(average.toFixed(1))
}
coinlist.sort((a, b) => b.average - a.average)
buildNewCoinList(coinlist)
}

```

```
function downloadList(filename, data) {  
  var blob = new Blob([JSON.stringify(data)], { type: "text/javascript" });  
  var link = document.createElement("a");  
  link.setAttribute("href", URL.createObjectURL(blob));  
  link.setAttribute("download", `${filename}.txt`);  
  link.click();  
}
```

```
// loginRender()
```

```
buttonsFunc()
```

```
buildCoinList(coin)
```