

Міністерство освіти і науки України
Криворізький національний університет
Факультет інформаційних технологій
Кафедра автоматизації, комп'ютерних наук і технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти – бакалавр
за освітньо-професійною програмою
«Комп'ютерні науки»
зі спеціальності
122 – Комп'ютерні науки

тема роботи:

***«Розробка веб-орієнтованої системи підтримки прийняття рішень,
щодо вибору постачальників послуг на основі нечітких моделей
оцінки репутації»***

Виконав студент гр. КН-22-1 _____	Скакуненко Б.І.
Керівник _____	Савицький О.І.
Нормоконтроль _____	Маринич І. А.
Завідувач кафедри _____	Рубан С. А.

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет: інформаційних технологійКафедра: автоматизації, комп'ютерних наук і технологійСтупінь вищої освіти: БакалаврСпеціальність: 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Зав. кафедрою: к.т.н. Рубан С.А.« 29 » січня 2026 р.

ЗАВДАННЯ

на кваліфікаційну роботу бакалавра

студентові групи КН-22-1 Скакуненко Б.І.

1. Тема кваліфікаційної роботи: Розробка веб-орієнтованої системи підтримки прийняття рішень, щодо вибору постачальників послуг на основі нечітких моделей оцінки репутації. затверджено наказом по університету № 173с від 30.03.2026 р.

2. Термін здачі кваліфікаційної роботи: 15.06.2026 р.

3. Склад кваліфікаційної роботи: Пояснювальна записка обсягом 66 с., презентація у Microsoft PowerPoint (30 слайдів) в електронному та друкованому вигляді

4. Консультанти кваліфікаційної роботи:

Розділ 1-2доц. Савицький О.І.Нормоконтрольдоц. Маринич І. А.

5. Календарний план:

№	Етапи роботи	Термін виконання
1	<i>Вступ</i>	<i>18.02.26</i>
2	<i>Розділ 1</i>	<i>01.05.26</i>
3	<i>Розділ 2</i>	<i>01.06.26</i>
4	<i>Висновки</i>	<i>05.06.26</i>
5	<i>Оформлення кваліфікаційної роботи</i>	<i>08.06.26</i>
6	<i>Підготовка презентації та графічного матеріалу</i>	<i>10.06.26</i>
7	<i>Підготовка доповіді до захисту</i>	<i>15.06.26</i>

6. Дата видачі завдання: 28.01.2026р.

Керівник _____ Савицький О.І.

7. Запевнення: Я, Скакуненко Б.І., запевняю, що ця кваліфікаційна робота виконана самостійно, не містить академічного плагіату, фабрикації, фальсифікації. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Із чинним Положенням про академічну доброчесність Криворізького національного університету ознайомлений.

Чітко усвідомлюю, що в разі виявлення у кваліфікаційній роботі умисних порушень робота не допускається до захисту або оцінюється незадовільно.

Студент _____ Скакуненко Б.І.

АНОТАЦІЯ

Скакуненко Б.І. «Розробка веб-орієнтованої системи підтримки прийняття рішень, щодо вибору постачальників послуг на основі нечітких моделей оцінки репутації»: кваліфікаційна робота бакалавра: 122 - Комп'ютерні науки. Кривий Ріг. Криворізький національний університет, 2026, 66 с.

Актуальність теми полягає у застосуванні систем підтримки прийняття рішень (СППР), які враховують декілька критеріїв оцінювання одночасно та дозволяють формувати комплексну репутаційну оцінку виконавців.

Метою кваліфікаційної роботи є розробка веб-орієнтованої системи ППР щодо вибору постачальників послуг на основі нечітких моделей оцінки репутації, яка забезпечує підвищення об'єктивності ранжування виконавців та автоматизацію процесу вибору оптимального постачальника послуг.

У роботі проведено аналіз сучасних онлайн-платформ пошуку послуг, обґрунтовано доцільність використання нечітких моделей оцінки репутації та багатокритеріального підходу до ППР під час вибору постачальників послуг.

У результаті виконання роботи спроектовано та реалізовано веб-орієнтовану інформаційну систему SmartChoice, яка забезпечує реєстрацію користувачів, керування профілями клієнтів і виконавців, створення та пошук послуг, оформлення замовлень, обмін повідомленнями, формування відгуків та накопичення статистичних даних щодо діяльності виконавців.

Ключовим елементом системи є репутаційний модуль, концептуально побудований на принципах нечіткого оцінювання, який використовує сукупність показників довіри клієнтів, професійної надійності та активності виконавця для формування інтегрального індексу репутації.

Проведене тестування підтвердило працездатність основних функціональних модулів системи та коректність реалізації алгоритмів репутаційного оцінювання. Отримані результати свідчать про можливість практичного використання розробленої системи для ППР під час вибору виконавців послуг.

КЛЮЧОВІ СЛОВА: ВЕБ-ОРІЄНТОВАНА СИСТЕМА, СППР, РЕПУТАЦІЙНЕ ОЦІНЮВАННЯ, РЕЙТИНГ ВИКОНАВЦІВ, НЕЧІТКА ЛОГІКА.

ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА	
ЗАДАЧІ ДОСЛІДЖЕННЯ.....	7
1.1 Тенденції розвитку цифрового ринку послуг в Україні та актуальність впровадження інтелектуальних систем підтримки прийняття рішень.....	7
1.2 Аналіз існуючих платформ та систем пошуку постачальників послуг	9
1.3 Методи оцінювання репутації постачальників послуг та підтримки прийняття рішень	14
1.3.1 Репутація як критерій оцінювання постачальника послуг	14
1.3.2 Математичний аналіз класичних підходів до формування рейтингів та їхні обмеження.....	16
1.3.3 Методи багатокритеріального вибору в системах підтримки прийняття рішень	20
1.3.4 Використання нечіткої логіки для оцінювання репутації постачальників послуг	22
1.4 Обґрунтування вибору технологій для реалізації веб-орієнтованої системи підтримки прийняття рішень	24
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ	
ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, ЩОДО ВИБОРУ ПОСТАЧАЛЬНИКІВ	
ПОСЛУГ НА ОСНОВІ НЕЧІТКИХ МОДЕЛЕЙ ОЦІНКИ РЕПУТАЦІЇ	28
2.1 Формалізація вимог до системи та моделювання бізнес-процесів взаємодії	28
2.1.1 Аналіз ролей користувачів та функціональних сценаріїв роботи системи	28
2.1.2 Формалізація функціональних та нефункціональних вимог	29
2.1.3 Моделювання бізнес-процесу взаємодії клієнта та виконавця	31
2.1.4 Формування інформаційних потоків для інтелектуального оцінювання	33
2.2 Програмне забезпечення системи та структура бази даних веб-платформи	34
2.2.1 Характеристика технологічного стеку та архітектури додатка	34
2.2.2 Структура та схема реляційної бази даних	36
2.3 Опис алгоритмів та функціональних модулів інформаційної системи...	40
2.4 Тестування та демонстрація роботи веб-системи «SmartChoice».....	51
ВИСНОВКИ.....	63
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	65

ВСТУП

Сучасний розвиток цифрових технологій сприяв активному поширенню онлайн-платформ пошуку та замовлення послуг. Водночас збільшення кількості виконавців ускладнює процес вибору постачальника послуг, а традиційні механізми оцінювання, що базуються переважно на рейтингах і відгуках користувачів, не завжди забезпечують об'єктивну оцінку якості послуг. Тому актуальним є застосування систем підтримки прийняття рішень, які враховують декілька критеріїв оцінювання одночасно та дозволяють формувати комплексну репутаційну оцінку виконавців.

Об'єктом дослідження є процес вибору постачальників послуг в електронних сервісах.

Предметом дослідження є методи та програмні засоби побудови веб-орієнтованих систем підтримки прийняття рішень на основі нечітких моделей оцінки репутації.

Метою кваліфікаційної роботи є розробка веб-орієнтованої системи підтримки прийняття рішень щодо вибору постачальників послуг на основі нечітких моделей оцінки репутації, яка забезпечує підвищення об'єктивності ранжування виконавців та автоматизацію процесу вибору оптимального постачальника послуг.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Провести аналіз сучасних платформ пошуку послуг, підходів до оцінювання репутації виконавців та методів підтримки прийняття рішень.
2. Дослідити можливості використання нечітких моделей та розробити математичну модель оцінки репутації постачальників послуг.
3. Спроекувати архітектуру та структуру веб-орієнтованої системи підтримки прийняття рішень.
4. Реалізувати програмне забезпечення системи, провести його тестування та оцінити результати роботи.

Практичне значення роботи полягає у створенні веб-орієнтованої інформаційної системи, яка забезпечує автоматизацію процесу пошуку та оцінювання виконавців послуг і підтримує прийняття рішень під час вибору постачальника послуг.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Тенденції розвитку цифрового ринку послуг в Україні та актуальність впровадження інтелектуальних систем підтримки прийняття рішень

Сучасний етап розвитку глобальної та національної економіки характеризується масштабною цифровою трансформацією, що охоплює не лише сектор промислового виробництва та торгівлі матеріальними благами, а й сферу послуг. Стрімка інтеграція інформаційно-комунікаційних технологій (ІКТ) зумовила появу та еволюцію парадигми електронної комерції (e-commerce), яка трансформувалася у специфічний підсектор – електронний ринок послуг (e-services market). Ринок побутових, ремонтних, освітніх та цифрових послуг в Україні демонструє стійку тенденцію до децентралізації та переходу у веб-орієнтоване середовище, де взаємодія між кінцевим споживачем (замовником) та постачальником (виконавцем) опосередковується спеціалізованими програмними платформами [1].

Зростання попиту на послуги індивідуальних майстрів (електриків, сантехніків, репетиторів, ІТ-спеціалістів) висуває нові вимоги до швидкості, гнучкості та найголовніше, надійності процесів їхнього підбору. Проте специфіка сфери послуг, на відміну від товарного ринку, полягає у високому рівні асиметрії інформації та невизначеності щодо якості кінцевого продукту. Це зумовлено фундаментальними економіко-технічними властивостями самої послуги:

- **нематеріальність (невідчутність):** послугу неможливо продемонструвати, оцінити чи протестувати до моменту її безпосереднього отримання, що створює високі ризики для замовника;
- **невіддільність від джерела:** процес надання послуги нерозривно пов'язаний з особистістю виконавця, його професійними та психологічними характеристиками;

- **мінливість якості:** кінцевий результат залежить від кваліфікації майстра, його поточного психоемоційного стану, часу виконання та зовнішніх чинників;
- **нездатність до зберігання:** невикористана вчасно послуга не може бути накопичена на складі для подальшої реалізації, що вимагає динамічного керування попитом і пропозицією у реальному часі [2].

У таких умовах ключовим чинником, що визначає вибір споживача та економічну стабільність взаємодії, стає репутація постачальника послуг. Аналіз динаміки українського сегменту електронних послуг свідчить, що традиційні підходи до агрегації оголошень та організації транзакцій вичерпують свій потенціал через системні недоліки архітектури сучасних веб-платформ [3].

Узагальнення наявних підходів дозволяє констатувати наявність системного науково-технічного протиріччя між стрімким кількісним зростанням національного ринку електронних послуг та обмеженими можливостями наявного програмного інструментарію для його якісного аналізу. Традиційні транзакційні веб-системи (OLTP-платформи), що функціонують за принципом реляційних баз даних та жорсткої лінійної фільтрації, здатні ефективно обробляти лише детерміновані дані (ціна, місто, категорія) [6]. Проте процес вибору виконавця людиною завжди є багатокритеріальним і спирається на слабоструктуровану інформацію, яка містить якісні, лінгвістичні та суб'єктивні оцінки [4].

Впровадження інтелектуальних систем підтримки прийняття рішень (ІСППР) у дану предметну область та перехід від концепції звичайного «каталогу оголошень» до розумного вибору є високоактуальним з огляду на такі фундаментальні чинники.

Необхідність формалізації нечітких знань. Репутація, надійність та компетентність майстра є динамічними параметрами, які не мають чітких числових меж. Використання інтелектуального ядра дозволяє перевести природно-мовні експертні оцінки користувачів у чіткий математичний формат за допомогою теорії нечітких множин (Fuzzy Logic), що є неможливим у класичних реляційних моделях [6].

Критичне зростання обсягів неструктурованих даних. Велика кількість відгуків, оцінок та текстових описів, які генеруються користувачами, створює надмірне когнітивне навантаження на споживача. СППР бере на себе виконання рутинних розрахунків, агрегуючи цей багатовимірний простір ознак у єдиний інтегральний показник – рекомендаційний бал [7].

Необхідність мінімізації часу на пошук та прийняття рішень. В умовах воєнного стану та критичних інфраструктурних викликів в Україні (аварійні ремонти, перебої з енергопостачанням) швидкість підбору оптимального виконавця та врахування його географічної доступності й оперативної готовності виїхати на об'єкт набувають стратегічного значення [1, 4].

Автоматизація управління ризиками та динамічність параметрів: інтелектуальний алгоритм здатний у реальному часі реагувати на зміну поведінкових характеристик виконавця (наприклад, збільшення часу відповіді на запити, падіння відсотка успішно завершених замовлень чи появу скарг), оперативно коригуючи його позицію в системі та захищаючи замовника від неякісного сервісу [7].

З огляду на це, розробка архітектури та комп'ютерне моделювання веб-орієнтованої СППР є актуальним науково-практичним завданням у галузі системного аналізу, теорії прийняття рішень та інтелектуальних інформаційних технологій. Створення такої системи дозволяє реалізувати на практиці повний життєвий цикл комп'ютеризованого управління репутаційними ризиками: від формалізації нечітких експертних знань до побудови архітектури розподіленої інформаційної системи та її інтелектуального аналітичного ядра [5].

1.2 Аналіз існуючих платформ та систем пошуку постачальників послуг

Стрімкий розвиток цифрової економіки сприяв появі великої кількості веб-платформ, призначених для пошуку виконавців та замовлення різноманітних послуг. Такі платформи виконують функції посередника між замовником і

виконавцем, забезпечуючи розміщення оголошень, пошук послуг, комунікацію між сторонами та систему відгуків.

На українському ринку найбільш поширеними є два типи електронних платформ. До першої групи належать універсальні дошки оголошень, серед яких найбільш відомою є платформа OLX. До другої групи відносяться спеціалізовані маркетплейси послуг, такі як Kabanchik.ua та Freelance.ua. Незважаючи на популярність і широке використання, більшість таких систем реалізують лише базові механізми пошуку та не забезпечують інтелектуальної підтримки користувача під час прийняття рішення щодо вибору виконавця [9].

Платформа OLX (рисунк 1.1) орієнтована переважно на розміщення та перегляд оголошень. Пошук виконавців здійснюється за допомогою фільтрації за категорією, місцем розташування та окремими параметрами. Основною перевагою такого підходу є простота використання та велика кількість пропозицій. Водночас відсутність механізмів оцінювання репутації, автоматичного ранжування та аналізу якості виконавців значно ускладнює процес вибору для користувача [8].

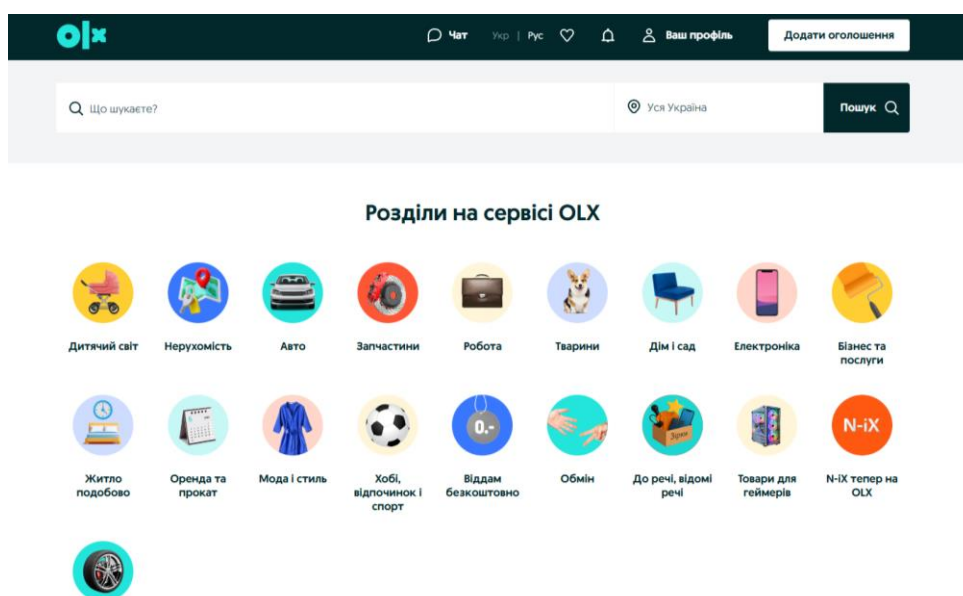


Рисунок 1.1 - Платформа OLX

Сервіс Kabanchik.ua (рисунк 1.2) реалізує більш розвинений підхід до організації взаємодії між замовниками та виконавцями. Платформа підтримує систему рейтингів, відгуків, безпечних платежів та категоризацію послуг. Однак

алгоритми підбору виконавців залишаються переважно лінійними та базуються на простому сортуванні за рейтингом або кількістю виконаних замовлень. У результаті користувач змушений самостійно аналізувати значну кількість параметрів та приймати остаточне рішення [10].

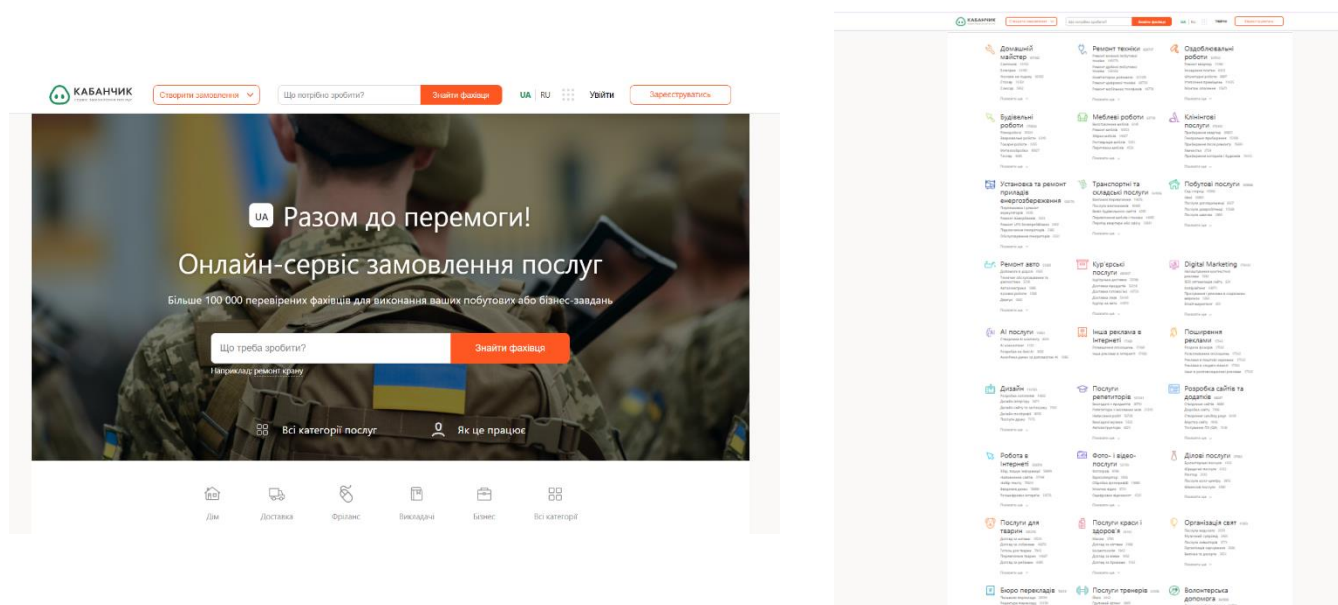


Рисунок 1.2 - Сервіс Kabanchik.ua

Freelance.ua орієнтована насамперед на дистанційне виконання проєктів у сфері інформаційних технологій, дизайну та маркетингу. Платформа використовує систему рейтингів та портфолію, однак також не враховує комплексно різноманітні критерії оцінювання виконавців і не надає механізмів автоматизованої підтримки прийняття рішень [13].

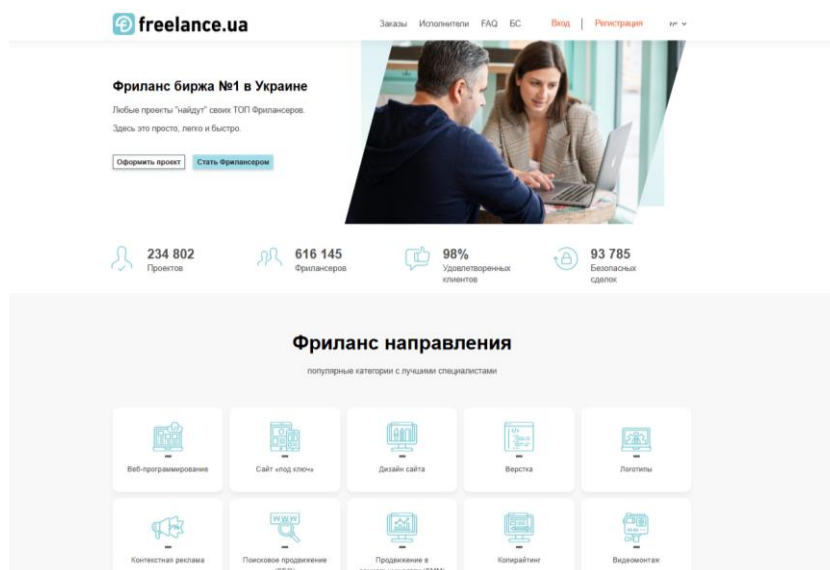


Рисунок 1.3 – Платформа Freelance.ua

Проведений аналіз дозволяє виділити низку спільних недоліків, характерних для більшості сучасних платформ пошуку послуг.

По-перше, переважна більшість систем використовує механізм лінійного усереднення оцінок. Рейтинг виконавця визначається як середнє арифметичне значення отриманих відгуків, що не враховує статистичну достовірність вибірки. У результаті виконавець із кількома позитивними оцінками може отримувати вищу позицію у видачі порівняно з досвідченим фахівцем, який має сотні виконаних замовлень та незначно нижчий середній рейтинг [11].

По-друге, існуючі системи значною мірою залежать від суб'єктивних оцінок користувачів. Відгуки можуть формуватися під впливом емоційних факторів, особистих уподобань або зовнішніх обставин, що не завжди об'єктивно характеризують якість виконаної роботи. Крім того, існує ризик маніпуляції рейтингами шляхом створення фіктивних відгуків або навмисного заниження оцінок конкурентами [12].

По-третє, суттєвою проблемою є ефект «холодного старту» (Cold Start Problem). Нові виконавці, які не мають історії замовлень та накопиченої репутації, практично не отримують можливості конкурувати з більш відомими спеціалістами. Це негативно впливає як на розвиток самої платформи, так і на рівень конкуренції на ринку послуг [13].

По-четверте, більшість платформ використовує єдиний рейтинг для всього профілю виконавця. Такий підхід є некоректним у випадках, коли одна особа надає послуги в декількох професійних сферах. Наприклад, високий рейтинг виконавця як електрика не гарантує аналогічного рівня компетентності у сфері репетиторства чи інформаційних технологій. Використання спільного рейтингу призводить до викривлення результатів пошуку та знижує достовірність оцінювання [14].

По-п'яте, сучасні системи практично не враховують поточну активність виконавців. Майстер може мати високий історичний рейтинг, однак тривалий час не користуватися платформою, повільно реагувати на запити або не приймати

нові замовлення. У результаті користувач отримує рекомендації спеціалістів, які фактично недоступні для співпраці [12].

По-шосте, пошук виконавців здійснюється переважно шляхом жорсткої фільтрації за окремими параметрами. Такий підхід не дозволяє враховувати взаємозв'язок між критеріями та знаходити компромісні рішення. Наприклад, замовник може бути зацікавлений у виконавцеві з дещо вищою вартістю послуг, але значно кращою репутацією та швидшим часом реагування. Класичні механізми фільтрації не здатні ефективно вирішувати подібні багатокритеріальні задачі [15].

Узагальнюючи результати проведеного аналізу (таблиця 1.1), можна зробити висновок, що більшість існуючих платформ орієнтована на забезпечення інформаційної взаємодії між замовником та виконавцем, але не на підтримку процесу прийняття рішення. Відсутність механізмів комплексного оцінювання репутації, багатокритеріального аналізу та персоналізованих рекомендацій знижує ефективність вибору постачальника послуг.

Таблиця 1.1 – Результати проведеного аналізу досліджень

КРИТЕРІЙ	OLX	Kabanchik.ua	Freelance.ua	Запропонована система
Пошук за категоріями	+	+	+	+
Пошук за містом	+	+	+	+
Система відгуків	Частково	+	+	+
Рейтинг виконавців	—	+	+	+
Врахування кількості відгуків	—	Частково	Частково	+
Верифікація виконавців	—	+	Частково	+
Оцінка активності виконавця	—	—	—	+
Персоналізований підбір	—	—	—	+
Багатокритеріальний аналіз	—	—	—	+
Нечітка модель оцінки репутації	—	—	—	+
СППР (Decision Support System)	—	—	—	+
Окремий рейтинг за різними послугами	—	—	—	+

Для усунення зазначених недоліків доцільним є використання веб-орієнтованої системи підтримки прийняття рішень, яка поєднує механізми оцінювання репутації виконавців на основі нечітких моделей із персоналізованим багатокритеріальним підбором послуг. Такий підхід дозволяє формалізувати невизначеність, враховувати як об'єктивні характеристики виконавця, так і індивідуальні потреби користувача, а також підвищити якість та обґрунтованість прийнятих рішень.

1.3 Методи оцінювання репутації постачальників послуг та підтримки прийняття рішень

1.3.1 Репутація як критерій оцінювання постачальника послуг

У сучасних електронних сервісах та цифрових маркетплейсах репутація постачальника послуг виступає одним із ключових чинників, що впливають на вибір користувача. На відміну від традиційного товарного ринку, де споживач може оцінити фізичні характеристики продукції до моменту придбання, у сфері послуг остаточна якість результату стає відомою лише після завершення взаємодії між замовником і виконавцем. Саме тому репутаційні механізми виконують роль інструменту зниження інформаційної невизначеності та формування довіри між сторонами електронної взаємодії [14].

У науковій літературі репутація розглядається як інтегральна характеристика суб'єкта, що формується на основі накопиченого досвіду взаємодії, оцінок користувачів, історії виконання зобов'язань та інших показників поведінки в інформаційному середовищі. Згідно з підходом, запропонованим А. Йосангом, репутація є агрегованим відображенням колективної думки учасників системи щодо надійності певного об'єкта або суб'єкта та використовується для прогнозування його майбутньої поведінки [14].

Особливого значення репутаційні механізми набули в умовах розвитку цифрових платформ, що функціонують за моделлю посередництва між виконавцями та замовниками. У таких системах потенційний клієнт не має

можливості безпосередньо перевірити кваліфікацію майстра, його професійні навички або якість майбутнього результату. У зв'язку з цим рішення про вибір виконавця значною мірою базується на непрямих характеристиках, серед яких центральне місце займає репутація [12].

Практика функціонування сучасних маркетплейсів послуг свідчить, що репутація формується під впливом багатьох факторів. До основних показників, які використовуються для її оцінювання, належать:

- середня оцінка, отримана від попередніх клієнтів;
- кількість відгуків;
- досвід роботи виконавця;
- кількість успішно виконаних замовлень;
- швидкість реагування на звернення користувачів;
- відсоток завершених замовлень;
- рівень верифікації профілю;
- активність користувача в системі [12, 14].

Водночас використання лише одного окремого показника не дозволяє об'єктивно оцінити надійність постачальника послуг. Наприклад, висока середня оцінка не завжди свідчить про значний професійний досвід, а велика кількість виконаних замовлень не гарантує стабільно високої якості роботи. Таким чином, репутація є багатофакторним поняттям, що потребує комплексного аналізу декількох взаємопов'язаних характеристик [16].

Додатковою проблемою є суб'єктивний характер більшості репутаційних показників. Оцінки користувачів залежать від індивідуальних очікувань, емоційного стану та особистого досвіду взаємодії з виконавцем. У результаті навіть за однакової якості наданої послуги різні клієнти можуть виставляти суттєво відмінні оцінки. Це створює невизначеність, яку складно врахувати за допомогою класичних математичних моделей [14].

Отже, репутація постачальника послуг є складною багатокритеріальною характеристикою, що формується під впливом великої кількості кількісних та якісних параметрів. Для її адекватного оцінювання необхідно використовувати

підходи, здатні враховувати невизначеність, суб'єктивність та неповноту вхідних даних. Саме тому при побудові сучасних систем підтримки прийняття рішень дедалі ширше застосовуються інтелектуальні методи аналізу даних та моделі нечіткої логіки [18].

1.3.2 Математичний аналіз класичних підходів до формування рейтингів та їхні обмеження

У більшості сучасних веб-платформ оцінювання постачальників послуг здійснюється за допомогою рейтингових механізмів, які дозволяють ранжувати виконавців відповідно до рівня довіри та якості їхньої роботи. Основною метою рейтингу є спрощення процесу вибору виконавця шляхом перетворення великої кількості різнорідної інформації у єдиний числовий показник, який може бути використаний для сортування результатів пошуку та формування рекомендацій [12].

Для формалізації репутації в сучасних інформаційних системах використовуються різні математичні підходи, кожен з яких має власні переваги та обмеження в умовах неповноти, невизначеності та слабкої структурованості даних.

Середнє арифметичне (Mean Rating)

Найпростішим та найбільш поширеним способом оцінювання репутації є використання середнього арифметичного значення оцінок користувачів:

$$R_{mean} = \frac{\sum_{i=1}^n r_i}{n} \quad (1.1)$$

де:

R_{mean} - середній рейтинг виконавця;

r_i - оцінка, отримана за (i)-те замовлення;

n - загальна кількість відгуків.

Популярність даного підходу пояснюється простотою реалізації та зрозумілістю для користувачів. Проте його використання має низку суттєвих недоліків. Насамперед середнє арифметичне не враховує обсяг статистичної

вибірки. У результаті виконавець із двома позитивними відгуками може отримати такий самий рейтинг, як і спеціаліст із сотнями успішно виконаних замовлень. Для ілюстрації зазначеної проблеми розглянемо приклад.

Таблиця 1.2 - Приклад порівняння виконавців за середнім рейтингом

Показник	Майстер А	Майстер Б
Середня оцінка	5,0	4,9
Кількість відгуків	2	200
Виконані замовлення	2	200

Відповідно до класичної схеми сортування перший виконавець буде розташований вище у пошуковій видачі, незважаючи на те, що другий має значно більший практичний досвід та статистично більш надійні результати. Така ситуація є прикладом викривлення результатів ранжування та зниження достовірності рекомендацій [11].

Зважене середнє (Weighted Mean)

Для часткового усунення недоліків середнього арифметичного використовується модель зваженого середнього, яка дозволяє враховувати різну значущість окремих оцінок:

$$R_{weighted} = \frac{\sum_{i=1}^n w_i * r_i}{\sum_{i=1}^n w_i} \quad (1.2)$$

де:

w_i - ваговий коефіцієнт оцінки;

r_i - оцінка користувача;

$R_{weighted}$ - підсумковий рейтинг.

У практичних системах вагові коефіцієнти можуть визначатися залежно від статусу користувача, вартості виконаного замовлення, тривалості співпраці або інших характеристик. Такий підхід дозволяє підвищити вплив більш значущих транзакцій на загальний рейтинг виконавця.

Однак навіть використання вагових коефіцієнтів не усуває проблему недостатньої кількості відгуків та не вирішує проблему так званого «холодного старту» для нових виконавців [13].

Баєсівський рейтинг (Bayesian Rating)

Більш досконалим підходом є використання баєсівського рейтингу, який враховує не лише власні оцінки виконавця, але й загальну статистику платформи:

$$R_{bayesian} = \frac{v \cdot m + n \cdot R}{v + n} \quad (1.3)$$

де:

R - середній рейтинг конкретного виконавця;

n - кількість отриманих відгуків;

m - середній рейтинг по всій платформі;

v - мінімальна кількість відгуків, необхідна для стабілізації оцінки.

Баєсівський підхід дозволяє зменшити вплив випадкових оцінок та згладити рейтинги користувачів із малою кількістю відгуків. Фактично нові виконавці отримують рейтинг, близький до середнього по системі, а зі збільшенням кількості відгуків їхній рейтинг поступово наближається до реального значення [20].

Незважаючи на статистичну обґрунтованість такого підходу, він вимагає постійного обчислення глобальних параметрів системи та не враховує особливості різних категорій послуг, що обмежує його застосування в багатопрофільних маркетплейсах [12, 20].

Інтервал згоди Вілсона (Wilson Score Interval)

Для вирішення проблеми малих вибірок у рейтингових системах також використовується інтервал згоди Вілсона (Wilson Score Interval). Метод базується на статистичному оцінюванні довірчого інтервалу для частки позитивних оцінок та дозволяє ранжувати користувачів з урахуванням рівня невизначеності.

Нижня межа інтервалу Вілсона визначається за формулою [21]:

$$S_{wilson} = \frac{\hat{p} + \frac{z_{\alpha/2}^2}{2n} - z_{\alpha/2} \sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z_{\alpha/2}^2}{4n^2}}}{1 + \frac{z_{\alpha/2}^2}{n}} \quad (1.4)$$

де:

$\hat{p}$ - частка позитивних відгуків;

n - загальна кількість оцінок;

$Z_{\alpha/2}$ - квантиль нормального розподілу для обраного рівня довіри α .

Особливістю даного методу є те, що профілі з малою кількістю оцінок автоматично отримують нижчі значення рейтингу через високий рівень статистичної невизначеності. Таким чином система уникає ситуацій, коли декілька випадкових позитивних відгуків штучно піднімають нового виконавця на верхні позиції рейтингу.

Разом із тим використання інтервалу Вілсона суттєво ускладнює обчислювальний процес, особливо при роботі з багатобальними шкалами оцінювання (наприклад, від 1 до 5 зірок), а отримані результати часто є недостатньо зрозумілими для кінцевих користувачів [12, 21].

Отже, проведений аналіз показує, що класичні рейтингові моделі дозволяють частково розв'язувати задачу оцінювання репутації виконавців, однак кожна з них має певні обмеження. Середнє та зважене середнє не враховують статистичну достовірність даних, баєсівські моделі потребують додаткових глобальних параметрів системи, а метод Вілсона орієнтований переважно на статистичну оцінку позитивних і негативних відгуків.

Найважливішим недоліком усіх розглянутих підходів є те, що вони працюють переважно з однорідними числовими показниками та не враховують якісні характеристики виконавця, такі як професійний досвід, швидкість реагування на звернення клієнтів, рівень верифікації профілю, активність у системі чи стабільність виконання замовлень. Крім того, зазначені моделі не дозволяють ефективно поєднувати різнорідні критерії в єдину інтегральну оцінку без втрати змістовної інтерпретації результатів. Це обумовлює доцільність використання методів багатокритеріального аналізу та нечіткої логіки, які розглядаються в наступних підрозділах.

1.3.3 Методи багатокритеріального вибору в системах підтримки прийняття рішень

У реальних умовах процес прийняття рішень рідко ґрунтується на одному критерії. Під час вибору товару, послуги, постачальника або виконавця користувач одночасно враховує декілька взаємопов'язаних характеристик, які можуть мати різну важливість та навіть суперечити одна одній. Саме тому в теорії прийняття рішень широкого поширення набули методи багатокритеріального аналізу (Multi-Criteria Decision Making, MCDM), призначені для формалізації процесу вибору альтернатив за сукупністю критеріїв [15].

Особливістю багатокритеріальних задач є наявність множини альтернатив:

$$A = \{a_1, a_2, \dots, a_n\}$$

та множини критеріїв оцінювання:

$$K = \{k_1, k_2, \dots, k_m\}$$

Кожна альтернатива характеризується набором значень критеріїв, які можуть мати різну природу: кількісну, якісну, лінгвістичну або експертну. Завдання системи підтримки прийняття рішень полягає у визначенні найкращої альтернативи на основі узагальнення цих характеристик [18].

У контексті вибору постачальника послуг альтернативами виступають виконавці, а критеріями можуть бути:

- рейтинг користувачів (k_1): згладжений або трансформований показник задоволеності клієнтів;
- кількість відгуків (k_2): міра статистичної надійності профілю, що відображає досвід взаємодії системи з даним суб'єктом;
- професійний досвід (k_3): часовий період активної діяльності майстра у вказаній категорії;
- активність на платформі (k_4): частота оновлення статусу, швидкість реакції на нові заявки;
- вартість послуг (k_5): відповідність вартості послуг середньоринковим очікуванням замовника;

- географічна доступність / відстань (k_6): просторова віддаленість виконавця від місця виклику, що критично для екстрених сервісів.
- рівень верифікації профілю (k_7).

В найпростішому випадку інтегральна оцінка альтернативи може визначатися методом зваженої суми:

$$S = \sum_{i=1}^m w_i * x_i \quad (1.5)$$

де:

S - інтегральна оцінка альтернативи;

w_i - ваговий коефіцієнт i -го критерію;

x_i - значення відповідного критерію.

При цьому виконується умова нормалізації ваг:

$$\sum_{i=1}^m w_i = 1 \quad (1.6)$$

Метод зваженої суми є простим та ефективним для задач із повністю формалізованими критеріями. Однак його використання суттєво ускладнюється у випадках, коли частина параметрів описується не точними числовими значеннями, а якісними характеристиками типу «високий рейтинг», «достатній досвід», «швидка відповідь» або «низька вартість» [17].

Одним із найвідоміших підходів багатокритеріального аналізу є метод аналізу ієрархій (Analytic Hierarchy Process, АНР), запропонований Томасом Сааті. Метод передбачає побудову ієрархічної структури критеріїв та виконання попарного порівняння їх важливості [17].

Для матриці попарних порівнянь:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \dots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad (1.7)$$

визначаються власні вектори, що формують систему ваг критеріїв.

Незважаючи на високу точність, застосування АНР в онлайн-системах реального часу є обмеженим через необхідність виконання великої кількості попарних порівнянь та залучення експертів до формування матриць оцінювання.

Аналіз сучасних веб-платформ показує, що процес вибору виконавця послуг належить до класу слабоструктурованих багатокритеріальних задач. Користувач одночасно оцінює об'єктивні показники діяльності виконавця та власні суб'єктивні вимоги щодо бюджету, терміновості чи місця виконання робіт. У таких умовах традиційні методи багатокритеріальної оптимізації потребують доповнення інструментами, здатними працювати з невизначеністю та нечіткими знаннями [11].

Саме тому перспективним напрямом розвитку систем підтримки прийняття рішень є поєднання багатокритеріального аналізу з апаратом нечіткої логіки, який дозволяє формалізувати експертні оцінки та працювати з лінгвістичними змінними.

1.3.4 Використання нечіткої логіки для оцінювання репутації постачальників послуг

Однією з фундаментальних проблем оцінювання репутації постачальників послуг є високий рівень невизначеності та суб'єктивності вихідних даних. Такі характеристики, як професіоналізм, надійність, оперативність або довіра клієнтів, не можуть бути описані виключно точними числовими значеннями. У більшості випадків користувачі формулюють власні оцінки природною мовою, використовуючи поняття «високий рейтинг», «великий досвід», «швидка відповідь» або «надійний виконавець» [19].

Для формалізації подібних понять у 1965 році Лотфі Заде запропонував теорію нечітких множин (Fuzzy Sets Theory), яка стала основою сучасних систем нечіткого виведення та інтелектуальних СППР [19].

На відміну від класичної логіки, де елемент або належить множині, або не належить їй, нечітка логіка дозволяє визначати ступінь належності елемента до множини за допомогою функції належності:

$$\mu_A(x): X \rightarrow [0; 1] \quad (1.8)$$

де:

X - універсальна множина;

A - нечітка множина;

$\mu_A(x)$ - ступінь належності елемента x до множини A .

Наприклад, досвід роботи виконавця може бути представлений трьома лінгвістичними термами:

Experience = {Початківець, Надійний, Експерт}

Для кожного терма будується власна функція належності, яка визначає ступінь відповідності конкретного значення досвіду певній категорії.

У системах підтримки прийняття рішень найбільш поширеними є трикутні та трапецієподібні функції належності через простоту їх реалізації та інтерпретації [18].

Трикутна функція належності задається виразом:

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \geq c \end{cases} \quad (1.9)$$

де:

a, b, c — параметри нечіткої множини.

Процес нечіткого виведення складається з декількох послідовних етапів:

1. фазифікація вхідних параметрів;
2. застосування бази нечітких правил;
3. агрегування результатів;
4. дефазифікація та отримання числового значення.

Загальна структура правила нечіткого виведення має вигляд:

if $X_1 = A_1$ *and* $X_2 = A_2$ *and* $X_3 = A_3$ *then* $Y = B$

де:

- X_1, X_2, X_3 - вхідні параметри;
- A_1, A_2, A_3 - нечіткі терми;
- Y - результат оцінювання репутації;
- B - вихідна лінгвістична оцінка.

Для предметної області вибору постачальників послуг доцільним є використання трьох інтегральних параметрів:

X_1 - довіра клієнтів,

X_2 - професійна надійність,

X_3 - активність та доступність.

На основі цих параметрів система формує інтегральний індекс репутації виконавця:

$$Y = f(X_1, X_2, X_3) \quad (1.10)$$

де, функція f реалізується через базу нечітких правил.

Наприклад:

if Довіра клієнтів = Висока and Професійна надійність = Експерт and Активність = Висока then Репутація = Відмінна.

Перевагою такого підходу є можливість одночасного врахування як кількісних, так і якісних характеристик виконавця без втрати змістової інтерпретації результатів. Крім того, нечітка логіка дозволяє моделювати людський процес прийняття рішень, що особливо важливо для веб-платформ пошуку послуг, де остаточний вибір значною мірою залежить від суб'єктивного сприйняття користувача [14].

Таким чином, використання апарату нечіткої логіки створює теоретичне підґрунтя для побудови інтелектуальної системи підтримки прийняття рішень, яка дозволяє оцінювати репутацію виконавців за сукупністю різномірних критеріїв та формувати персоналізовані рекомендації для клієнтів.

1.4 Обґрунтування вибору технологій для реалізації веб-орієнтованої системи підтримки прийняття рішень

Розробка веб-орієнтованої системи підтримки прийняття рішень щодо вибору постачальників послуг потребує комплексного підходу до вибору програмних засобів та технологій реалізації. На відміну від звичайних

інформаційних веб-сайтів, запропонована система, яка буде називатися «SmartChoice» повинна одночасно забезпечувати зберігання та обробку даних користувачів, підтримувати механізми бронювання послуг, виконувати розрахунки репутаційних показників у режимі реального часу та формувати персоналізовані рекомендації на основі математичної моделі нечіткого оцінювання.

Особливістю предметної області є необхідність одночасної роботи з декількома категоріями користувачів, серед яких клієнти, виконавці та адміністратори системи. Кожна з груп має власні сценарії взаємодії із програмним середовищем, що зумовлює необхідність реалізації гнучкої архітектури програмного забезпечення, здатної забезпечити масштабованість та подальший розвиток функціональних можливостей системи без суттєвої зміни її структури [22].

З огляду на зазначені вимоги, доцільним є використання веб-орієнтованої архітектури, побудованої за принципом клієнт-серверної взаємодії. Такий підхід дозволяє забезпечити доступ до системи незалежно від типу пристрою користувача, операційної системи або місця його перебування. Доступ до функціональних можливостей здійснюється через стандартний веб-браузер, що усуває необхідність встановлення спеціалізованого програмного забезпечення та спрощує супровід програмного продукту [23].

Для реалізації клієнтської частини системи доцільно використовувати компонентний підхід до побудови інтерфейсу користувача. Основною перевагою такого підходу є можливість створення незалежних програмних компонентів, які можуть багаторазово використовуватися в різних частинах системи. З урахуванням значної кількості однотипних елементів інтерфейсу (картки послуг, профілі виконавців, форми реєстрації, модулі бронювання та рейтингового оцінювання) використання компонентної архітектури дозволяє зменшити обсяг програмного коду та спростити його подальшу підтримку [24].

Для безпосередньої реалізації інтерфейсу користувача обрано бібліотеку React, яка забезпечує ефективне керування станом веб-додатка та підтримує

побудову складних односторінкових застосунків (Single Page Application). Використання React дозволяє мінімізувати кількість безпосередніх звернень до сервера під час навігації між сторінками та забезпечує швидке оновлення елементів інтерфейсу без повного перезавантаження веб-сторінки. Такий підхід є особливо важливим для систем, у яких користувачі активно взаємодіють із пошуковими фільтрами, формами динамічних замовлень та інтерактивними персоналізованими рекомендаціями [24]. Для реалізації миттєвого відображення оновлень у системі (наприклад, відкриття вікна чату після бронювання, передача текстових повідомлень у реальному часі або миттєва зміна статусу замовлення) у клієнт-серверній архітектурі додатково передбачено використання протоколу двостороннього зв'язку WebSockets.

Для забезпечення масштабованості, безпеки та ефективної обробки даних на серверній стороні обґрунтовано використання сучасної хмарної BaaS-платформи Supabase, яка інтегрує в собі засоби автентифікації користувачів, реляційну систему керування базами даних PostgreSQL, механізми збереження статичних файлів та хмарні безсерверні обчислення для гнучкої реалізації бізнес-логіки системи [26].

Особливістю розроблюваної системи є наявність інтелектуального модуля підтримки прийняття рішень, який виконує розрахунок інтегрального індексу репутації виконавців та формування персоналізованих рекомендацій на основі вхідних параметрів довіри клієнтів, професійної надійності та поточної активності провайдера. З метою забезпечення безпеки алгоритмів, захисту бази правил від несанкціонованих маніпуляцій та суттєвого зменшення обчислювального навантаження на клієнтський застосунок, обчислення математичної нечіткої моделі Мамдані виконується строго на серверній стороні із використанням Supabase Edge Functions на базі ізольованого середовища виконання Deno. Реалізація цих безсерверних функцій засобами мови TypeScript забезпечує уніфікований підхід до розробки як Frontend, так і Backend компонентів, а також спрощує підтримку всього програмного коду [25, 26].

Застосування серверної обробки дозволяє надійно приховати математичні алгоритми вагових коефіцієнтів від кінцевого користувача, централізувати процес обчислення репутаційних показників та забезпечити автоматичний асинхронний перерахунок рейтингу за допомогою серверних тригерів безпосередньо після надходження нових відгуків або зміни статистичних показників діяльності виконавця.

Для збереження, структурування та агрегування інформації в екосистемі Supabase використовується потужна реляційна СКБД PostgreSQL, яка гарантує підтримку принципів ACID і забезпечує роботу зі складними SQL-запитами, тригерами, представленнями (Views) та процедурами обробки даних. З огляду на специфіку предметної області, структура бази даних проектується за схемою суворої декомпозиції відношень типу «один до багатьох» (1:N):

Provider(1) → (N) Service(1) → (N) Reviews

Така архітектурна організація зв'язків дозволяє ізолювати оцінки та відгуки в межах конкретної послуги майстра, а не всього його профілю. Використання PostgreSQL дозволяє здійснювати попередню статистичну агрегацію та згладжування первинних числових показників (кількості відгуків, оцінок) безпосередньо на рівні бази даних за допомогою вбудованих функцій і представлень, після чого очищені детерміновані метрики передаються в інтелектуальний модуль Supabase Edge Functions для подальшого нечіткого логічного аналізу [27].

Обраний технологічний стек повністю відповідає сучасним тенденціям розробки розподілених веб-орієнтованих інформаційних систем і забезпечує можливість надійної інтеграції інтелектуальних алгоритмів підтримки прийняття рішень без необхідності розгортання та адміністрування додаткової громіздкої серверної інфраструктури. Використання хмарної архітектури також спрощує подальше горизонтальне масштабування системи SmartChoice та її швидке впровадження в реальних умовах експлуатації.

РОЗДІЛ 2.

ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ, ЩОДО ВИБОРУ ПОСТАЧАЛЬНИКІВ ПОСЛУГ НА ОСНОВІ НЕЧІТКИХ МОДЕЛЕЙ ОЦІНКИ РЕПУТАЦІЇ

2.1 Формалізація вимог до системи та моделювання бізнес-процесів взаємодії

2.1.1 Аналіз ролей користувачів та функціональних сценаріїв роботи системи

Проектування веб-орієнтованої системи підтримки прийняття рішень SmartChoice базується на принципах рольового розмежування доступу та сценарного підходу до моделювання поведінки користувачів. Такий підхід дозволяє забезпечити цілісність інформаційних потоків, підвищити безпеку системи та спростити подальше масштабування програмного продукту.

У межах предметної області було визначено три основні категорії користувачів: клієнт, виконавець та адміністратор системи. Кожна роль має власний набір функцій та взаємодіє із системою відповідно до визначених бізнес-процесів.

Клієнт є кінцевим споживачем послуг та виступає ініціатором більшості транзакцій у системі. Основним завданням клієнта є пошук виконавця, який найбільш повно відповідає встановленим критеріям. Після проходження реєстрації користувач отримує доступ до персонального кабінету, де може редагувати власні контактні дані, переглядати історію замовлень, здійснювати пошук послуг та формувати запити до виконавців.

Процес взаємодії клієнта із системою починається з вибору категорії послуг та налаштування параметрів пошуку. Після аналізу доступних пропозицій клієнт обирає конкретну послугу, надсилає запит виконавцю та очікує на його відповідь. У разі підтвердження замовлення система відкриває канал комунікації між сторонами, що забезпечує можливість уточнення деталей виконання робіт. Після

завершення замовлення клієнт отримує можливість оцінити якість наданої послуги та залишити текстовий відгук.

Виконавець є постачальником послуг та основним джерелом формування контенту платформи. Після створення облікового запису користувач отримує доступ до персонального кабінету, де може створювати та редагувати власні послуги, керувати портфоліо виконаних робіт, переглядати вхідні запити клієнтів та контролювати поточні замовлення.

На відміну від традиційних платформ оголошень, у системі SmartChoice кожна послуга розглядається як окрема сутність із власними характеристиками, параметрами оцінювання та репутаційними показниками. Такий підхід дозволяє уникнути ситуації, коли загальний рейтинг виконавця поширюється на всі напрями його діяльності незалежно від фактичного досвіду.

Після надходження запиту виконавець може прийняти або відхилити його. У випадку прийняття формується активне замовлення, в межах якого здійснюється подальша взаємодія із клієнтом. Після завершення робіт виконавець позначає замовлення як виконане, що відкриває можливість для формування зворотного зв'язку та оновлення репутаційних показників.

Адміністратор системи виконує функції контролю та підтримки працездатності платформи. До його завдань належать перевірка профілів виконавців, модерація контенту, управління категоріями послуг, контроль коректності функціонування системи та вирішення спірних ситуацій між користувачами.

Таким чином, визначені ролі формують замкнений цикл взаємодії, який забезпечує безперервне накопичення інформації про якість наданих послуг та створює основу для роботи інтелектуального механізму підтримки прийняття рішень.

2.1.2 Формалізація функціональних та нефункціональних вимог

Після визначення основних ролей користувачів наступним етапом проєктування є формалізація вимог до майбутньої системи. Вимоги визначають

функціональні можливості програмного забезпечення, а також обмеження та характеристики якості, яким повинна відповідати розроблена інформаційна система.

До функціональних вимог SmartChoice належать можливості реєстрації та автентифікації користувачів, створення персональних профілів, формування та редагування послуг, виконання пошуку за встановленими критеріями, надсилання запитів виконавцям, ведення переписки між учасниками замовлення, управління статусами замовлень та формування системи відгуків.

Система повинна забезпечувати підтримку різних типів користувачів та надавати їм індивідуальні інтерфейси відповідно до їхніх ролей. Клієнт повинен мати можливість швидко знаходити потрібні послуги, переглядати інформацію про виконавців та здійснювати замовлення. Виконавець повинен отримати інструменти для управління власними послугами, портфолію та взаємодії із клієнтами.

Окремою функціональною вимогою є реалізація механізму оцінювання якості послуг та формування репутаційних показників. Система повинна забезпечувати накопичення статистичних даних про діяльність виконавця, які в подальшому використовуються для побудови рекомендаційних моделей.

Крім функціональних характеристик, важливе значення мають нефункціональні вимоги, що визначають якість роботи системи.

Однією з ключових вимог є продуктивність. Інтерфейс користувача повинен забезпечувати швидку реакцію на дії користувача та комфортну навігацію між сторінками. Пошук послуг, завантаження профілів та обробка замовлень мають виконуватися без помітних затримок.

Не менш важливою є вимога масштабованості. Архітектура системи повинна передбачати можливість збільшення кількості користувачів, послуг та транзакцій без суттєвого зниження продуктивності. Для цього використовуються сучасні веб-технології та хмарна інфраструктура з можливістю горизонтального розширення.

Особливу увагу приділено безпеці інформації. Система повинна забезпечувати захист персональних даних користувачів, контроль доступу до функціональних модулів та запобігання несанкціонованим змінам інформації. Механізми автентифікації та авторизації мають гарантувати розмежування прав доступу між різними категоріями користувачів.

Важливою вимогою також є доступність. Користувачі повинні мати можливість працювати із системою незалежно від типу пристрою чи операційної системи. Інтерфейс має коректно адаптуватися до різних розмірів екранів та забезпечувати зручність використання як на персональних комп'ютерах, так і на мобільних пристроях.

Сукупність наведених функціональних та нефункціональних вимог формує основу для подальшого проектування архітектури системи та реалізації її програмних компонентів.

2.1.3 Моделювання бізнес-процесу взаємодії клієнта та виконавця

Ефективність роботи системи підтримки прийняття рішень значною мірою залежить від правильного проектування бізнес-процесів взаємодії між учасниками платформи. Саме тому на етапі аналізу було сформовано узагальнену модель життєвого циклу замовлення, яка відображає послідовність дій користувачів від моменту реєстрації до формування репутаційної оцінки виконавця.

Початковим етапом бізнес-процесу є реєстрація користувача в системі. Після створення облікового запису виконавець отримує можливість створювати власні послуги, визначати їх характеристики та публікувати їх у каталозі платформи.

Наступним етапом є пошук послуг клієнтом. На основі обраної категорії, міста та інших параметрів користувач отримує перелік доступних пропозицій та обирає виконавця, який найбільше відповідає його потребам.

Після вибору послуги формується запит на виконання робіт. Запит надходить до кабінету виконавця, де може бути прийнятий або відхилений. У разі

підтвердження запиту створюється активне замовлення та відкривається можливість обміну повідомленнями між сторонами.

Подальший етап передбачає виконання робіт та фіксацію результату. Після завершення замовлення виконавець змінює його статус на виконаний, що активує механізм збору зворотного зв'язку.

Завершальним етапом є оцінювання якості послуги клієнтом. Отримані оцінки та відгуки використовуються для оновлення показників репутації та формування рекомендацій для наступних користувачів.

Узагальнена схема бізнес-процесу має такий вигляд:

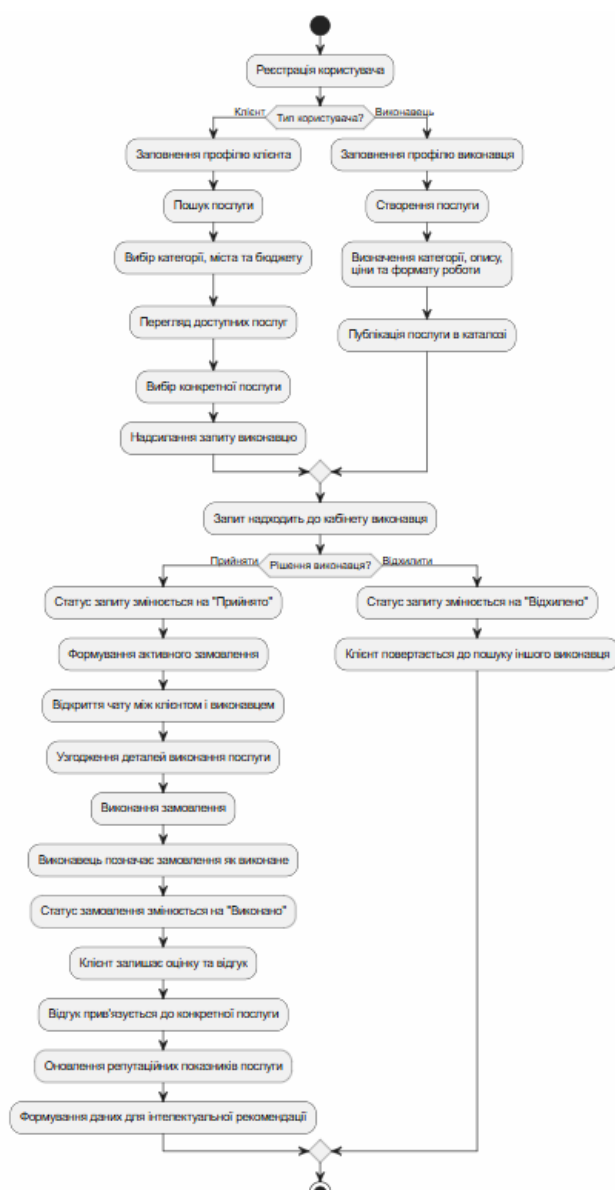


Рисунок 2.1 — UML-діаграма активності бізнес-процесу взаємодії клієнта та виконавця в системі SmartChoice.

Запропонована модель забезпечує безперервний цикл накопичення даних та створює основу для роботи інтелектуальної системи підтримки прийняття рішень.

2.1.4 Формування інформаційних потоків для інтелектуального оцінювання

Особливістю розроблюваної системи є використання репутаційної моделі, яка базується не лише на суб'єктивних оцінках користувачів, а й на об'єктивних характеристиках діяльності виконавця. Для забезпечення роботи такого механізму необхідно визначити джерела формування вхідних даних для подальшого інтелектуального аналізу. Архітектуру бази даних проєктуємо таким чином, щоб автоматично акумулювати метрики на кожному етапі життєвого циклу замовлення, описаного в підрозділі 2.1.3.

Показник довіри клієнтів (вхідна змінна X_1) формується на основі інформації, отриманої після переведення замовлень у фінальний статус «Виконано». Джерелом цих даних є детерміновані оцінки та текстові відгуки, які залишають клієнти. При цьому в базі даних запис відгуку жорстко пов'язується із унікальним ідентифікатором послуги. Крім середнього арифметичного значення оцінок, обчислювальне ядро системи враховує загальну кількість накопичених відгуків, що дозволяє реалізувати алгоритми статистичного згладжування та підвищити достовірність результатів оцінювання, захищаючи систему від штучних накруток.

Показник професійної надійності (вхідна змінна X_2) формується на основі аналізу досвіду роботи виконавця за конкретним напрямком (поле стажу роботи в описі послуги), а також загального аналізу історії транзакцій. Інформаційним потоком для цієї змінної виступає відношення кількості успішно завершених замовлень (статус «Виконано») до загальної кількості колись прийнятих заявок, а також бінарний статус адміністративної верифікації профілю майстра за завантаженими документами.

Показник активності та доступності (вхідна змінна X_3) базується на поведінкових факторах взаємодії виконавця з платформою. Джерелом даних тут

виступає ступінь заповненості профілю виконавця, наявність контактної інформації, портфоліо робіт та швидкість реагування на вхідні запити.

Усі зазначені показники формуються та записуються в базу даних автоматично в процесі природної взаємодії користувачів із платформою. Завдяки цьому виключається необхідність ручного введення додаткових параметрів, мінімізується людський фактор та забезпечується об'єктивність оцінювання.

Сформовані інформаційні потоки виступають основою для подальшого застосування математичного апарату нечіткої логіки та розрахунку *інтегрального показника репутації* (вихідна змінна Y), що використовується системою для підтримки прийняття рішень під час вибору постачальника послуг.

На відміну від традиційних платформ пошуку послуг, де основним критерієм вибору є середня оцінка користувачів, у проєктованій системі використовується багатокритеріальний підхід до формування репутації. Це дозволяє враховувати як суб'єктивні оцінки клієнтів, так і об'єктивні характеристики діяльності виконавця, що підвищує достовірність рекомендацій та забезпечує більш обґрунтоване прийняття рішень.

2.2 Програмне забезпечення системи та структура бази даних веб-платформи

2.2.1 Характеристика технологічного стеку та архітектури додатка

Розробка веб-орієнтованої інформаційної системи підтримки прийняття рішень «SmartChoice» виконана з використанням сучасної клієнт-серверної архітектури та компонентного підходу до побудови користувацького інтерфейсу. Основною метою архітектурного рішення є забезпечення розділення функціональних модулів системи, підтримка ролівої моделі доступу, збереження даних у реляційній структурі та можливість розрахунку репутаційних показників для кожної окремої послуги виконавця.

Клієнтська частина системи реалізована засобами бібліотеки React та мови програмування TypeScript. Використання TypeScript забезпечує сувору типізацію

об'єктів, зменшує ймовірність помилок під час роботи з даними та дозволяє формалізувати структуру основних сутностей системи. Для організації маршрутизації використовується TanStack Router, що забезпечує поділ системи на окремі сторінки: головну сторінку, сторінку пошуку послуг, сторінку «Розумний вибір», детальну сторінку послуги, сторінку авторизації, а також захищені кабінети клієнта та виконавця.

Структура маршрутів системи передбачає наявність як публічних сторінок, доступних усім користувачам, так і захищених сторінок, доступ до яких відкривається лише після авторизації. Зокрема, до публічної частини належать головна сторінка, пошук послуг, сторінка конкретної послуги та модуль розумного вибору. До захищеної частини належать кабінет клієнта та кабінет виконавця.

Для побудови інтерфейсу використано компонентний підхід. Візуальні елементи інтерфейсу формуються з окремих багаторазових компонентів: кнопок, діалогових вікон, форм введення, слайдерів, вкладок та карток послуг. Така організація дає змогу підтримувати єдиний стиль системи та спрощує подальше розширення функціональності.

Для зберігання даних використовується хмарна реляційна база даних PostgreSQL. Взаємодія клієнтської частини з базою даних здійснюється через API-клієнт, який забезпечує виконання операцій читання, створення та оновлення записів. Типи таблиць, зв'язків та перелічуваних значень описані в окремому файлі типізації, що дозволяє узгодити програмний код із фактичною структурою бази даних.

У системі реалізовано такі основні функціональні модулі:

- ✓ модуль автентифікації та керування сесією користувача;
- ✓ модуль профілів клієнтів і виконавців;
- ✓ модуль створення та перегляду послуг;
- ✓ модуль пошуку та фільтрації послуг;
- ✓ модуль оформлення замовлень;
- ✓ модуль обміну повідомленнями;

- ✓ модуль відгуків;
- ✓ репутаційний модуль підтримки прийняття рішень.

Таким чином, архітектура системи побудована за модульним принципом і забезпечує можливість розділення відповідальності між окремими функціональними частинами програмного продукту.

2.2.2 Структура та схема реляційної бази даних

Зберігання інформації, необхідної для функціонування системи SmartChoice та розрахунку репутаційних показників виконавців, реалізовано за допомогою реляційної бази даних. Основними сутностями системи є профілі користувачів, послуги, портфоліо, замовлення, повідомлення та відгуки.

Для забезпечення цілісності даних у базі реалізовано перелічувані типи даних. Тип `user_role` визначає роль користувача в системі та може набувати значень `customer`, `provider`, `admin`. Тип `order_status` визначає стан замовлення та містить значення `pending`, `accepted`, `rejected`, `completed`.

Фрагмент створення перелічуваних типів має такий вигляд:

```
CREATE TYPE public.user_role AS ENUM ('customer','provider','admin');
CREATE TYPE public.order_status AS ENUM ('pending','accepted','rejected','completed');
```

Таблиця `profiles` використовується для зберігання профілів користувачів. Вона містить ідентифікатор користувача, роль, ім'я, електронну пошту, телефон, місто, аватар, контакт у месенджері та статус верифікації.

Таблиця 2.1 - Сутність `profiles` (Профілі користувачів)

Назва поля	Тип даних	Обмеження	Опис атрибута
<code>id</code>	UUID	PRIMARY KEY, FK	Унікальний ідентифікатор (зв'язок з <code>auth.users</code>)
<code>role</code>	<code>user_role</code>	NOT NULL, DEFAULT 'customer'	Рольова модель доступу користувача (<code>customer</code> , <code>provider</code> , <code>admin</code>)
<code>name</code>	TEXT	NOT NULL, DEFAULT	Повне ім'я або назва компанії / виконавця

email	TEXT	NOT NULL, DEFAULT	Електронна пошта для сповіщень та авторизації
phone	TEXT	NULL	Контактний номер телефону користувача
city	TEXT	NULL	Місто проживання / надання послуг
avatar	TEXT	NULL	Графічне зображення профілю (аватар)
is_verified	BOOLEAN	NOT NULL, DEFAULT false	Статус офіційної верифікації документів майстра

Таблиця `services` є однією з ключових у системі, оскільки саме послуга є основним об'єктом оцінювання. Репутація формується не для профілю виконавця загалом, а для кожної окремої послуги. Такий підхід дозволяє коректно оцінювати виконавця в різних напрямках діяльності.

Таблиця 2.2 - Сутність `services` (Послуги виконавців)

Назва поля	Тип даних	Обмеження	Опис атрибута
id	UUID	PRIMARY KEY, DEFAULT uuid()	Унікальний ідентифікатор оголошення послуги
provider_id	UUID	NOT NULL, FOREIGN KEY	Посилання на автора послуги (<code>profiles.id</code>)
title	TEXT	NOT NULL	Назва послуги (наприклад, «Ремонт / заміна бойлера»)
category	TEXT	NOT NULL	Назва категорії послуги (для групування та фільтрації)
experience_years	INT	NULL	Задекларований професійний стаж майстра (у роках)
base_price	NUMERIC	NOT NULL, DEFAULT 0	Початкова вартість виконання робіт (базовий тариф)
city	TEXT	NULL	Локація/місто надання даної послуги
work_format	TEXT	NULL	Формат взаємодії (на виїзд до клієнта, у майстерні)

Таблиця `service_portfolio` призначена для збереження посилань на зображення виконаних робіт, що належать до конкретної послуги.

```
CREATE TABLE public.service_portfolio (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  service_id UUID NOT NULL REFERENCES public.services(id) ON DELETE CASCADE,
  image_url TEXT NOT NULL,
  created_at TIMESTAMPTZ NOT NULL DEFAULT now()
);
```

Таблиця `orders` фіксує життєвий цикл взаємодії між клієнтом та виконавцем. Кожне замовлення пов'язане з конкретною послугою, виконавцем і клієнтом. Поле `status` визначає поточний стан замовлення.

Таблиця 2.3 - Сутність `orders` (Життєвий цикл замовлень)

Назва поля	Тип даних	Обмеження	Опис атрибута
<code>id</code>	UUID	PRIMARY KEY, DEFAULT <code>uuid()</code>	Унікальний номер замовлення (транзакції)
<code>service_id</code>	UUID	NOT NULL, FOREIGN KEY	Зв'язок із конкретною послугою (<code>services.id</code>)
<code>provider_id</code>	UUID	NOT NULL, FOREIGN KEY	Виконавець замовлення (майстер, <code>profiles.id</code>)
<code>customer_id</code>	UUID	NOT NULL, FOREIGN KEY	Клієнт-ініціатор замовлення (<code>profiles.id</code>)
<code>status</code>	<code>order_status</code>	NOT NULL, DEFAULT 'pending'	Поточний стан виконання (pending, accepted, rejected, completed)
<code>comment</code>	TEXT	NULL	Супровідний опис технічного завдання від клієнта
<code>completed_at</code>	TIMESTAMP TZ	NULL	Фіксація точного часу успішного закриття замовлення

Таблиця `order_messages` забезпечує збереження повідомлень між клієнтом і виконавцем у межах конкретного замовлення. Повідомлення можуть створюватися лише учасниками відповідного замовлення.

```
CREATE TABLE public.order_messages (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  order_id UUID NOT NULL REFERENCES public.orders(id) ON DELETE CASCADE,
  sender_id UUID NOT NULL REFERENCES public.profiles(id) ON DELETE CASCADE,
  sender_role public.user_role NOT NULL,
  text TEXT NOT NULL,
  created_at TIMESTAMPTZ NOT NULL DEFAULT now()
);
```

Таблиця `reviews` використовується для збереження оцінок та текстових відгуків клієнтів. Важливою особливістю є те, що відгук пов'язаний не лише з виконавцем, а й із конкретною послугою та конкретним завершеним замовленням. Це забезпечує достовірність репутаційних показників.

Таблиця 2.4 - Сутність `reviews` (Оцінки та відгуки)

Назва поля	Тип даних	Обмеження	Опис атрибута
<code>id</code>	UUID	PRIMARY KEY, DEFAULT <code>uuid()</code>	Унікальний номер відгуку в системі
<code>order_id</code>	UUID	NOT NULL, UNIQUE, FK	Зв'язок із транзакцією замовлення (<code>orders.id</code>)
<code>service_id</code>	UUID	NOT NULL, FOREIGN KEY	Конкретний об'єкт оцінювання (<code>services.id</code>)
<code>rating</code>	INT	CHECK (<code>rating BETWEEN 1 AND 5</code>)	Оцінка якості робіт за 5-бальною шкалою
<code>text</code>	TEXT	NULL	Текстова рецензія/коментар клієнта

З метою захисту даних у системі реалізовано політики доступу на рівні рядків. Наприклад, профілі доступні для перегляду всім користувачам, але редагувати профіль може лише його власник. Послуги доступні для перегляду всім користувачам, однак керувати ними може тільки виконавець, який їх створив. Замовлення доступні лише сторонам взаємодії: клієнту та виконавцю.

Приклад політики доступу до послуг:

```
CREATE POLICY "Services viewable by everyone"
ON public.services
FOR SELECT
USING (true);
CREATE POLICY "Providers manage own services"
ON public.services
FOR ALL USING (auth.uid() = provider_id)
WITH CHECK (auth.uid() = provider_id);
```

Таким чином, структура бази даних забезпечує збереження всіх необхідних даних для функціонування системи, підтримує рольову модель доступу, дозволяє

відстежувати життєвий цикл замовлення та формує інформаційну основу для розрахунку репутаційних показників.

2.3 Опис алгоритмів та функціональних модулів інформаційної системи

Модуль автентифікації та керування користувачами

Для забезпечення персоналізованого доступу до функцій системи реалізовано *модуль автентифікації користувачів*. Після успішного входу система створює користувацьку сесію, завантажує дані профілю та визначає роль користувача.

У системі передбачено три ролі користувачів:

- клієнт (customer);
- виконавець (provider);
- адміністратор (admin).

Після авторизації система автоматично виконує завантаження профілю користувача з бази даних та надає доступ до відповідного функціоналу. Для клієнта відкривається кабінет клієнта, а для виконавця – кабінет виконавця.

Модуль також забезпечує підтримку активної користувацької сесії, оновлення даних профілю та завершення роботи користувача шляхом виходу із системи. Реалізація рольової моделі доступу дозволяє обмежити доступ до функціональних можливостей відповідно до типу користувача та забезпечує коректну роботу всіх функціональних модулів системи. Профіль виконується за ідентифікатором користувача:

Модуль створення та відображення послуг

Модуль послуг забезпечує створення виконавцем власних пропозицій та їх подальше відображення в системі. Кожна послуга є окремою сутністю з власними характеристиками, що дозволяє формувати незалежні репутаційні показники для різних напрямів діяльності одного виконавця.

До основних параметрів послуги належать назва, категорія, спеціалізація, стаж роботи, базова ціна, місто, формат роботи, опис, посилання на соціальні мережі та відеоматеріали.

Формування об'єкта послуги в програмному коді здійснюється шляхом об'єднання основних характеристик, введених виконавцем через форму створення послуги:

```
const serviceData = {
  title,
  category,
  specialization,
  experience_years,
  base_price,
  city,
  work_format,
  description,
  social_url,
  youtube_url
};
```

Після заповнення форми введені дані проходять перевірку коректності та зберігаються в таблиці `services`. Надалі створена послуга стає доступною для відображення в особистому кабінеті виконавця, у результатах пошуку та в рекомендаційному модулі системи.

Модуль пошуку послуг

Модуль пошуку дозволяє клієнту знайти послугу за категорією, містом, ключовими словами та бюджетом. Для цього система формує запит до таблиці `services` та отримує також пов'язані дані про виконавця, відгуки та замовлення.

```
let qb = supabase
  .from("services")
  .select(
    "id,title,category,city,base_price,experience_years, provider:
    profiles!services_provider_id_fkey(name), reviews(rating), orders(status)" )
  .gte("base_price", budget[0])
  .lte("base_price", budget[1])
  .order("created_at", { ascending: false });
```

За необхідності до запиту додаються фільтри:

```
if (category) qb = qb.eq("category", category);
if (city) qb = qb.ilike("city", `%${city}%`);
if (q) qb = qb.ilike("title", `%${q}%`);
```

Після отримання даних система розраховує середню оцінку, кількість відгуків, кількість завершених замовлень та передає ці дані до репутаційного модуля.

```
const reviews = s.reviews ?? [];
const orders = s.orders ?? [];

const reviewsCount = reviews.length;
const avgNum = reviewsCount
? reviews.reduce((a, b) => a + b.rating, 0) / reviewsCount : 0;
const completed = orders.filter((o) => o.status === "completed").length;
const rep = calcReputation({
  averageRating: avgNum,
  reviewsCount,
  experienceYears: s.experience_years ?? 0,
  completedOrders: completed,
  receivedRequests: orders.length,
});
```

У результатах пошуку клієнт бачить не внутрішні змінні моделі, а зрозумілий підсумковий показник: індекс репутації та рівень виконавця.

```
<div className="mt-3 flex items-center justify-between gap-2 rounded-lg bg-secondary/40 px-3 py-2 text-sm">
  <span className="text-muted-foreground">Репутація</span>
  <span className="text-right">
    <span className="font-semibold">{rep.y}/100</span>
    {" · "}
    <span className="text-muted-foreground">{rep.level}</span>
  </span>
</div>
```

Таким чином, модуль пошуку виконує не лише функцію фільтрації, а й забезпечує первинну підтримку прийняття рішень клієнтом.

Модуль сторінки окремої послуги та створення замовлення

Детальна сторінка послуги відображає повну інформацію про конкретну пропозицію виконавця: категорію, назву, місто, формат роботи, спеціалізацію, стаж, опис, портфоліо, відгуки та репутаційний індекс.

Під час завантаження сторінки система отримує дані про послугу, її портфоліо, відгуки та пов'язані замовлення.

```

const { data: s } = await supabase
  .from("services")
  .select("*", provider:profiles!services_provider_id_fkey(id,name,city,is_verified,avatar_url))
  .eq("id", id)
  .maybeSingle();

setService(s);

const { data: p } = await supabase
  .from("service_portfolio")
  .select("*")
  .eq("service_id", id);

setPortfolio(p ?? []);

const { data: r } = await supabase
  .from("reviews")
  .select("*", customer:profiles!reviews_customer_id_fkey(name))
  .eq("service_id", id)
  .order("created_at", { ascending: false });

setReviews(r ?? []);

```

Для формування репутаційного показника система обчислює середню оцінку, кількість відгуків і кількість завершених замовлень:

```

const avgNum = reviews.length
  ? reviews.reduce((a, b) => a + b.rating, 0) / reviews.length : 0;
const completedOrders = orders.filter((o) => o.status === "completed").length;

const reputation = calcReputation({
  averageRating: avgNum,
  reviewsCount: reviews.length,
  experienceYears: service.experience_years ?? 0,
  completedOrders,
  receivedRequests: orders.length,
});

```

Клієнту на сторінці послуги відображаються середня оцінка, кількість відгуків, індекс репутації та рівень виконавця.

```

<div className="flex items-center justify-between gap-3">
  <span className="text-muted-foreground">Індекс репутації</span>
  <span className="font-semibold">{reputation.y}/100</span>
</div>
<div className="flex items-center justify-between gap-3">
  <span className="text-muted-foreground">Рівень репутації</span>
  <span className="font-medium">{reputation.level}</span>
</div>

```

Для створення замовлення клієнт натискає кнопку «Замовити послугу», після чого система створює запис у таблиці `orders`.

```

const order = async () => {
  if (!user) {
    navigate({ to: "/auth", search: { mode: "login" } });
    return;
  }

  if (profile?.role !== "customer") {
    toast.error("Замовляти можуть лише клієнти");
    return;
  }

  const { error } = await supabase.from("orders").insert({
    service_id: id,
    provider_id: service.provider_id,
    customer_id: user.id,
    comment,
  });

  if (error) {
    toast.error(error.message);
    return;
  }

  toast.success("Запит надіслано виконавцю");
  navigate({ to: "/customer-dashboard" });
};

```

Цей фрагмент демонструє рольове обмеження: створювати замовлення можуть лише користувачі з роллю клієнта.

Модуль «Розумний вибір»

Модуль «Розумний вибір» реалізовано як інтерфейсний та алгоритмічний рівень персоналізованого підбору послуг у системі SmartChoice. Його призначення полягає у розширенні стандартного пошуку шляхом урахування індивідуальних пріоритетів клієнта під час вибору виконавця.

На відміну від звичайного пошуку, який базується переважно на фільтрації за категорією, містом і бюджетом, модуль «Розумний вибір» дозволяє враховувати кілька критеріїв одночасно: репутацію виконавця, вартість послуги, близькість до клієнта та очікувану швидкість виконання. Користувач задає важливість кожного критерію за допомогою інтерактивних елементів інтерфейсу, після чого система формує персоналізовану логіку добору альтернатив.

Загальна модель оцінювання відповідності послуги потребам клієнта може бути подана у вигляді зваженої згортки:

$$S = w_1 * R + w_2 * P + w_3 * D + w_4 * T \quad (2.1)$$

де

S — персоналізований показник відповідності послуги запиту клієнта,

R — репутаційна оцінка послуги,

P — відповідність вартості бажаному бюджету,

D — територіальна близькість виконавця,

T — відповідність очікуваному терміну виконання,

w_1, w_2, w_3, w_4 — вагові коефіцієнти, що задаються користувачем.

Наприклад:

Наприклад:

Репутація важлива = 50%

$w_1 = 0.5$

Ціна важлива = 20%

$w_2 = 0.2$

Близькість = 20%

$w_3 = 0.2$

Швидкість = 10%

$w_4 = 0.1$

Фрагмент програмної логіки розрахунку персоналізованого показника має такий вигляд:

```
const priceScore = anyBudget
  ? 1
  : Math.max(0, 1 - s.base_price / Math.max(budget, 1));

const repScore =
  (avg / 5) * 0.7 + Math.min(1, s.reviews.length / 10) * 0.3;

const speedScore = Math.min(1, completed / 10);

const score =
  repScore * w.reputation +
  priceScore * w.price +
  speedScore * w.speed +
  0.5 * w.proximity;
```

У цьому фрагменті priceScore характеризує відповідність ціни заданому бюджету, repScore враховує оцінки та кількість відгуків, speedScore відображає наявність завершених замовлень, а загальний показник score формується з урахуванням вагових коефіцієнтів, визначених користувачем.

Після розрахунку показника відповідності для кожної послуги список альтернатив сортується за спаданням значення score:

```
list.sort((a, b) => b._score - a._score);
```

Таким чином, модуль «Розумний вибір» забезпечує персоналізоване ранжування доступних послуг відповідно до індивідуальних пріоритетів клієнта. Його використання підвищує зручність вибору виконавця та доповнює

репутаційний модуль системи, який формує базову оцінку надійності кожної послуги.

Важливим джерелом даних для роботи модуля «Розумний вибір» є репутаційний модуль системи. Під час формування рекомендацій використовується інтегральний індекс репутації Y , який обчислюється на основі показників довіри клієнтів, професійної надійності та активності виконавця. Таким чином, репутаційний модуль виступає базовим механізмом оцінювання якості послуг, а модуль «Розумний вибір» використовує отримані результати для персоналізованого ранжування альтернатив відповідно до індивідуальних пріоритетів користувача.

Модуль відгуків

Модуль відгуків забезпечує накопичення оцінок клієнтів після завершення замовлення. Відгук прив'язується до конкретного замовлення, послуги, виконавця та клієнта. Така структура дозволяє уникнути загального усереднення по всьому профілю виконавця та забезпечує формування репутації окремо за кожною послугою.

У базі даних для відгуку передбачено обмеження: оцінка має бути в діапазоні від 1 до 5, а один відгук може відповідати лише одному замовленню. Крім того, створення відгуку дозволяється лише для завершеного замовлення.

```
rating INT NOT NULL CHECK (rating BETWEEN 1 AND 5),
order_id UUID NOT NULL REFERENCES public.orders(id) ON DELETE CASCADE UNIQUE
```

Бізнес-правило створення відгуку реалізовано через перевірку статусу замовлення:

```
CREATE POLICY "Customer creates review for own completed order"
ON public.reviews
FOR INSERT
WITH CHECK (
    auth.uid() = customer_id
    AND EXISTS (
        SELECT 1
        FROM public.orders o
        WHERE o.id = order_id
            AND o.customer_id = auth.uid()
            AND o.status = 'completed'
    )
);
```

Це підвищує достовірність репутаційних даних, оскільки клієнт може залишити відгук лише після фактичного завершення взаємодії.

Репутаційний модуль підтримки прийняття рішень

Ключовим елементом інформаційної системи SmartChoice є репутаційний модуль, що забезпечує автоматичний розрахунок інтегрального індексу репутації для кожної окремої послуги. Модуль реалізовано у файлі `src/lib/reputation.ts`.

Вхідними даними для алгоритму є набір первинних статистичних показників:

```
[
I = \langle averageRating, reviewsCount, experienceYears, completedOrders, receivedRequests \rangle
]
```

У програмному коді ця структура описана так:

```
export type ReputationInput = {
  averageRating: number;
  reviewsCount: number;
  experienceYears: number;
  completedOrders: number;
  receivedRequests: number;
};

export type Reputation = {
  x1: number;
  x2: number;
  x3: number;
  y: number;
  level: string;
};
```

Для інтерпретації підсумкового індексу реалізовано функцію класифікації:

```
export function reputationLevel(y: number): string {
  if (y <= 20) return "Початковий рівень";
  if (y <= 40) return "Базовий виконавець";
  if (y <= 60) return "Надійний виконавець";
  if (y <= 80) return "Досвідчений виконавець";
  return "Експерт";
}
```

Основний алгоритм розрахунку виконується у функції `calcReputation`.

```

export function calcReputation(i: ReputationInput): Reputation {
  const averageRating = i.averageRating ?? 0;
  const reviewsCount = i.reviewsCount ?? 0;
  const experienceYears = i.experienceYears ?? 0;
  const completedOrders = i.completedOrders ?? 0;
  const receivedRequests = i.receivedRequests ?? 0;

  const normalizedAverageRating = reviewsCount
    ? (averageRating / 5) * 100
    : 0;

  const normalizedReviewsCount = Math.min((reviewsCount / 10) * 100, 100);
  const normalizedExperience = Math.min((experienceYears / 10) * 100, 100);
  const normalizedCompletedOrders = Math.min((completedOrders / 10) * 100, 100);
  const normalizedRequests = Math.min((receivedRequests / 10) * 100, 100);

  const x1 = Math.round(
    0.7 * normalizedAverageRating +
    0.3 * normalizedReviewsCount
  );

  const x2 = Math.round(
    0.3 * normalizedExperience +
    0.7 * normalizedCompletedOrders
  );

  const x3 = Math.round(
    0.5 * normalizedRequests +
    0.5 * normalizedCompletedOrders
  );

  const y = Math.round(
    0.5 * x1 +
    0.3 * x2 +
    0.2 * x3
  );

  return { x1, x2, x3, y, level: reputationLevel(y) };
}

```

Перед обчисленням усі первинні показники нормалізуються до шкали від 0 до 100. Для кількості відгуків, завершених замовлень, отриманих запитів і стажу роботи використано поріг насичення 10 одиниць. Це означає, що значення 10 і більше відповідає 100 балам за відповідним критерієм.

Під час реалізації репутаційного модуля для нормалізації показників кількості відгуків, завершених замовлень, отриманих запитів та стажу роботи було використано поріг насичення, рівний 10 умовним одиницям. Такий підхід дозволяє забезпечити достатню чутливість моделі на початкових етапах функціонування системи та уникнути ситуації, коли нові виконавці тривалий час отримують надто низькі репутаційні оцінки через недостатню кількість накопичених даних.

Обране значення не є фіксованим параметром моделі та може бути адаптоване відповідно до обсягу статистичних даних, накопичених у системі. У разі збільшення кількості користувачів, послуг, замовлень та відгуків поріг нормалізації може бути переглянутий або замінений більш складними механізмами масштабування, що дозволить забезпечити вищу точність репутаційного оцінювання в умовах великої кількості активних виконавців.

Проміжний показник X_1 характеризує довіру клієнтів і формується на основі середньої оцінки та кількості відгуків:

$$X_1 = 0.7 * R_{avg} + 0.3R_{count} \quad (2.2)$$

де:

R_{avg} - нормалізована середня оцінка клієнтів;

R_{count} - нормалізована кількість відгуків.

Показник X_2 характеризує професійну надійність виконавця:

$$X_2 = 0.3 * E + 0.7 * C \quad (2.3)$$

де:

E — нормалізований стаж роботи;

C — нормалізована кількість завершених замовлень.

Більша вага завершених замовлень пояснюється тим, що реальні виконані роботи всередині системи є більш об'єктивним показником надійності, ніж лише задекларований стаж.

Показник X_3 описує активність та доступність виконавця:

$$X_3 = 0.5 * Q + 0.5 * C \quad (2.4)$$

де:

Q — нормалізована кількість отриманих запитів;

C — нормалізована кількість завершених замовлень.

Фінальний інтегральний індекс репутації визначається за формулою:

$$Y = 0.5 * X_1 + 0.3 * X_2 + 0.2 * X_3 \quad (2.5)$$

де:

X_1 - довіра клієнтів;

X_2 - професійна надійність;

X_3 - активність та доступність;

Y - інтегральний індекс репутації послуги.

Така структура вагових коефіцієнтів відображає пріоритетність довіри клієнтів, оскільки саме реальні оцінки та відгуки є найбільш вагомим фактором вибору виконавця. Водночас професійна надійність та активність також впливають на підсумкову оцінку, але не домінують над клієнтським досвідом.

Результат роботи модуля використовується у двох формах. Для клієнта відображається лише підсумковий індекс репутації та текстовий рівень виконавця. Для виконавця у кабінеті доступна деталізація за всіма проміжними показниками X_1, X_2, X_3 та фінальним індексом Y . Такий підхід забезпечує прозорість оцінювання для виконавця та зручність сприйняття для клієнта.

Кабінет виконавця як аналітичний інтерфейс

Всередині захищеної панелі керування виконавця (`provider-dashboard.tsx`) реалізовано повний аналітичний екран рисунок 2.1. На відміну від клієнтського інтерфейсу, майстер отримує доступ до деталізованої декомпозиції власної репутації, розбитої по кожній його послуді окремо.

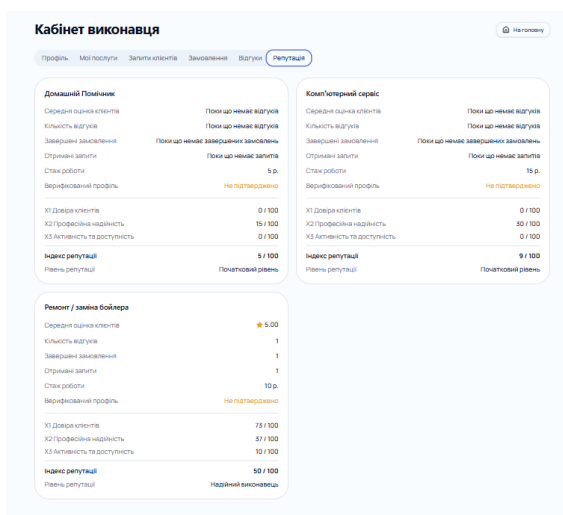


Рисунок 2.2 – Модуль «Репутація» у профілі виконавця

Інтерфейс кабінету виконавця візуалізує:

1. Первинні лічильники: Середня оцінка, Кількість відгуків, Кількість завершених замовлень, Отримані запити та Стаж роботи майстра за конкретним напрямком.

2. Внутрішні субіндекси (проміжне відображення):

X_1 — Довіра клієнтів (співвідношення оцінки та активності відгуків).

X_2 — Професійна надійність (де 70% ваги припадає саме на практичний досвід виконання замовлень всередині платформи, а 30% — на загальний стаж).

X_3 — Активність та доступність на платформі (конверсія запитів у виконанні завдання).

3. Результуючий статус: Підсумковий індекс репутації Y та його словесний еквівалент (наприклад, для послуги «Ремонт / заміна бойлера» зафіксовано реальні значення: оцінка — 5.0, відгуків — 1, замовлень — 1, стаж — 10 років, що дало підсумковий результат *50/100* із присвоєнням статусу «Надійний виконавець»).

Завдяки такій програмній реалізації виконавець може чітко бачити, яка саме слабка сторона (наприклад, низька швидкість обробки нових запитів чи брак відгуків) погіршує його позиції у загальному рейтингу системи, що стимулює підвищення якості надання послуг.

2.4 Тестування та демонстрація роботи веб-системи «SmartChoice»

У даному підрозділі представлено результати практичного розгортання та перевірки взаємодії ключових модулів веб-системи «SmartChoice». Головна увага під час тестування приділялася верифікації інтеграційних зв'язків між візуальними компонентами користувацького інтерфейсу та реляційними структурами хмарного сховища даних. Моделювання робочих процесів платформи виконувалося шляхом послідовного проходження базового користувацького сценарію за участю двох тестових профілів із різними рівнями доступу.

Отримані в процесі демонстрації результати та відповідні їм графічні інтерфейси (скріншоти екранних форм) детально описують логіку роботи веб-додатка на кожному етапі користувацького шляху.

Демонстрація роботи системи починається з головної сторінки системи. Вона містить основну навігацію, короткий опис призначення платформи, кнопки переходу до пошуку послуг, модуля розумного вибору та реєстрації виконавця. Головна сторінка виконує роль початкової точки входу користувача в систему та демонструє загальну ідею SmartChoice як інтелектуальної платформи вибору постачальників послуг.

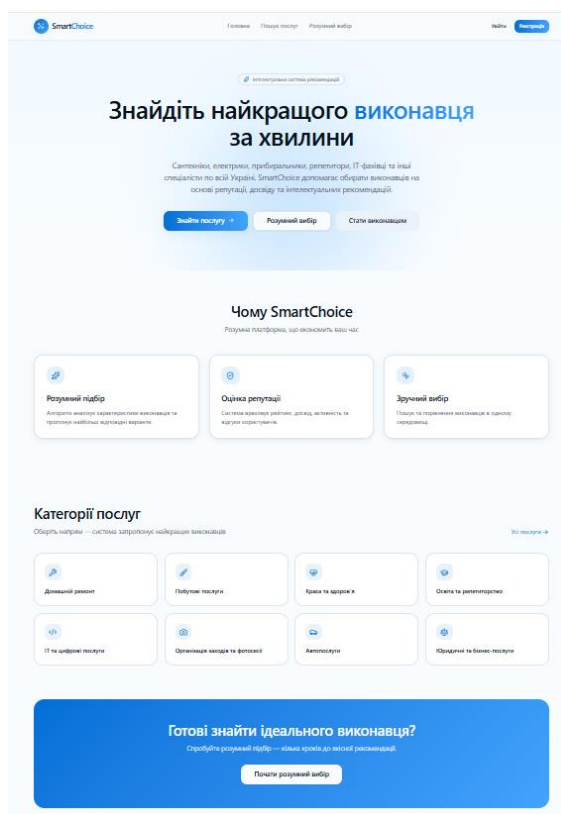


Рисунок 2.3 - Головна сторінка сайту

Наступним етапом протестуємо механізм реєстрації користувачів. Система передбачає вибір ролі користувача: клієнт або виконавець. Для обох ролей передбачено введення базових даних: імені, електронної пошти, пароля, телефону та міста. Після успішної реєстрації створюється профіль користувача, а подальший доступ до системи здійснюється через механізм авторизації.

The screenshot shows the 'SmartChoice' website's registration page. At the top, there's a navigation bar with 'Головна', 'Пошук послуг', 'Як це працює', 'Увійти', and 'Реєстрація'. The main heading is 'Реєстрація' with the subtitle 'Оберіть тип акаунту, щоб продовжити'. Below this are two large buttons: 'Клієнт' (Client) with a person icon and 'Виконавець' (Executor) with a briefcase icon. A link 'Вже маєте акаунт? Увійти' is at the bottom. To the right, a 'Вхід' (Login) modal is open, asking to 'Увійдіть, щоб користуватися рекомендаціями'. It has fields for 'Email' (you@example.com) and 'Пароль' (password), a 'Увійти' button, and a 'Увійти через Google' button. A link 'Немає акаунту? Зареєструватися' is at the bottom.

Рисунок 2.4 - Форма вибору ролі під час реєстрації

Two screenshots of the registration forms. The left one is for 'Реєстрація клієнта' (Client registration) and the right one is for 'Реєстрація виконавця' (Executor registration). Both forms have a '← Змінити роль' link at the top left. The client form has fields for 'Ім'я' (Name), 'Email', 'Пароль' (Password), 'Телефон' (Phone), and 'Місто' (City), with a 'Створити акаунт' button. The executor form has fields for 'Ім'я', 'Email', 'Пароль', 'Місто', 'Телефон', 'Telegram / Viber', and a 'Створити профіль' button.

Рисунок 2.5 - Форми реєстрації клієнта та виконавця

Після входу в систему клієнт отримує доступ до персонального кабінету, у якому відображаються основні розділи: профіль користувача, замовлення та відгуки. Профіль клієнта містить контактні дані, що використовуються для подальшої взаємодії із виконавцями. Наявність окремого кабінету дозволяє централізовано керувати історією звернень та результатами взаємодії із сервісом.

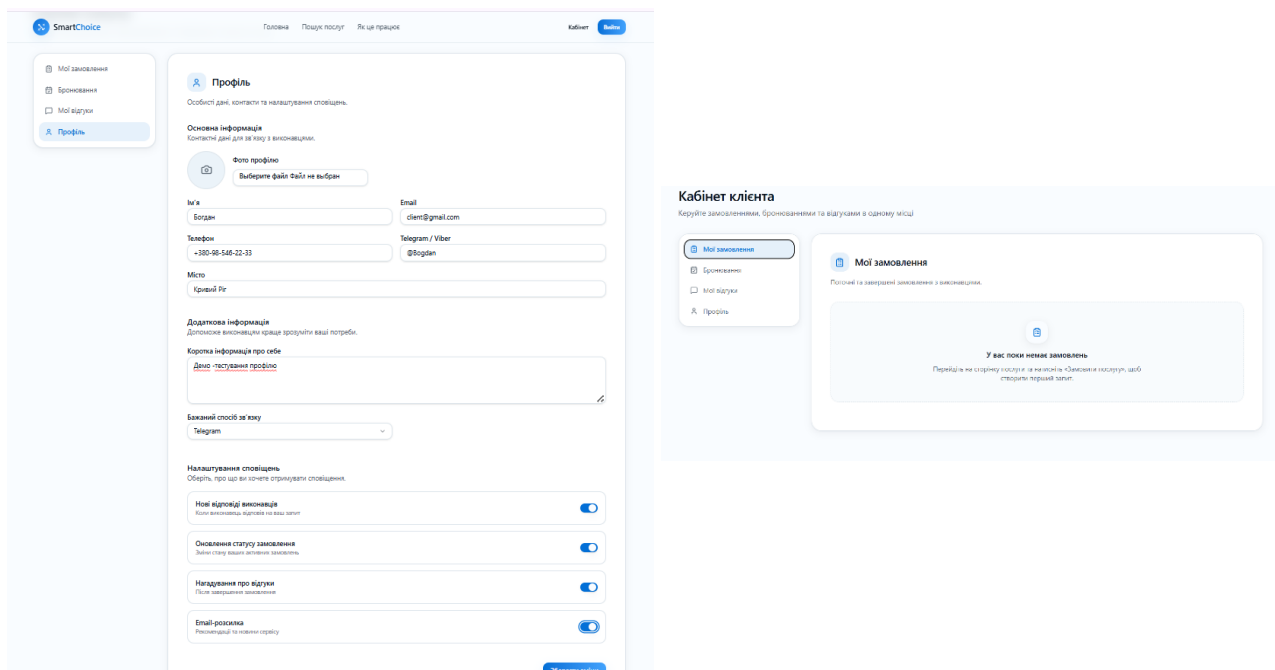


Рисунок 2.6 - Профіль клієнта

Для виконавця система формує окремий кабінет, що містить розділи керування профілем, послугами, запитами клієнтів, замовленнями, відгуками та репутацією. Така структура відповідає логіці функціонального розмежування ролей і забезпечує виконавцю можливість керувати власними послугами в межах однієї платформи.

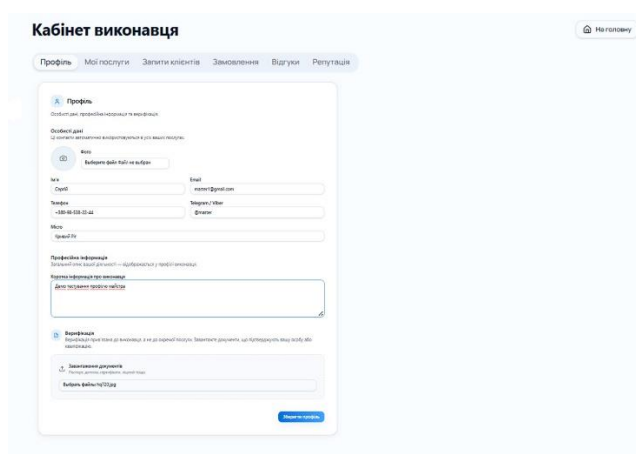


Рисунок 2.7 - Профіль виконавця

Одним із ключових етапів тестування є перевірка створення послуги виконавцем. У відповідній формі користувач зазначає назву послуги, категорію, спеціалізацію, стаж роботи, базову ціну, місто, формат роботи та опис. Такий

підхід дозволяє одному виконавцю створювати декілька незалежних послуг у різних категоріях. Це є важливою архітектурною особливістю системи, оскільки репутаційні показники формуються не для профілю виконавця загалом, а для кожної конкретної послуги окремо.

The image shows two parts of a web interface for creating a new service. The left part is a form titled 'Нова послуга' (New Service) with the instruction 'Заповніть інформацію про послугу. Виконавець може мати кілька послуг у різних категоріях.' (Fill in information about the service. The executor can have several services in different categories). The form fields include: 'Назва послуги' (Service Name) with an example 'Наприклад: встановлення кондиціонера' (Example: installation of an air conditioner); 'Категорія' (Category) with a dropdown 'Оберіть категорію'; 'Спеціалізація' (Specialization) with an example 'Наприклад: монтаж сантехніки'; 'Стаж роботи (років)' (Years of experience) with a value of 0; 'Базова ціна (€)' (Basic price (€)) with a value of 0; 'Формат роботи' (Work format) with a dropdown 'Оберіть формат'; and 'Опис послуги' (Service description) with a text area 'Опишіть, що входить у послугу'. Below the form is a 'Portfolio - фото робіт' section with a button 'Вибрати файли' and a message 'Файл не вибран'. The right part shows the 'Portfolio - фото робіт' section with a button 'Вибрати файли' and a message 'Файл не вибран'. It also includes a 'Посилання на соцмережі' (Social media links) field with a placeholder 'https://...', a 'Посилання на YouTube' (YouTube link) field with a placeholder 'https://youtube.com/...', and a checkbox 'Контактні дані (телефон, месенджери, фото) автоматично беруться з вашого профілю. Верифікація налаштується у розділі «Профіль».' (Contact data (phone, messengers, photo) are automatically taken from your profile. Verification will be set up in the 'Profile' section.). At the bottom are buttons 'Скасувати' (Cancel) and 'Зберегти послугу' (Save service).

Рисунок 2.8 - Форма додавання послуги

The image shows the 'Кабінет виконавця' (Executor's Cabinet) interface. It has a navigation bar with tabs: 'Профіль' (Profile), 'Мії послуги' (My services), 'Запити клієнтів' (Client requests), 'Завдання' (Tasks), 'Відгуки' (Reviews), and 'Репутація' (Reputation). There is a '+ Додати послугу' (Add service) button. The main area displays a list of services created by the executor. Each service card includes the service name, location, description, and price. For example, one service is 'Домашній Помічник' (Home Assistant) in 'Кривий Ріг' (Kriviy Rih) for 700 грн. Another is 'Ремонт / заміна бойлера' (Repair / replacement of boiler) in 'Кривий Ріг' for 1500 грн. A third is 'Комп'ютерний сервіс' (Computer service) in 'Кривий Ріг' for 1000 грн.

Рисунок 2.9 - Список створених послуг у кабінеті виконавця

Після створення послуги вона відображається в кабінеті у виконавця та у модулі пошуку всієї системи. Клієнт може здійснювати пошук за категорією, містом, бюджетом та іншими параметрами. Результати пошуку відображаються у вигляді карток послуг, де подано основну інформацію про послугу, виконавця, ціну та репутаційні характеристики. Така організація інтерфейсу дозволяє користувачу швидко порівнювати доступні варіанти.

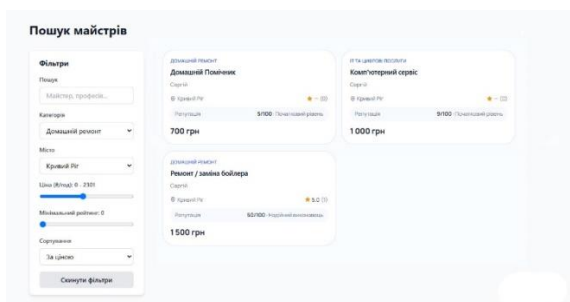


Рисунок 2.10 - Сторінка пошуку послуг

Окремим елементом системи є модуль «Розумний вибір», який призначений для підтримки прийняття рішень клієнтом. У цьому модулі користувач задає параметри запиту: категорію послуги, місто, бюджет, терміновість та пріоритетні напрями. На основі цих даних система формує рекомендаційну видачу, орієнтовану на потреби конкретного користувача.

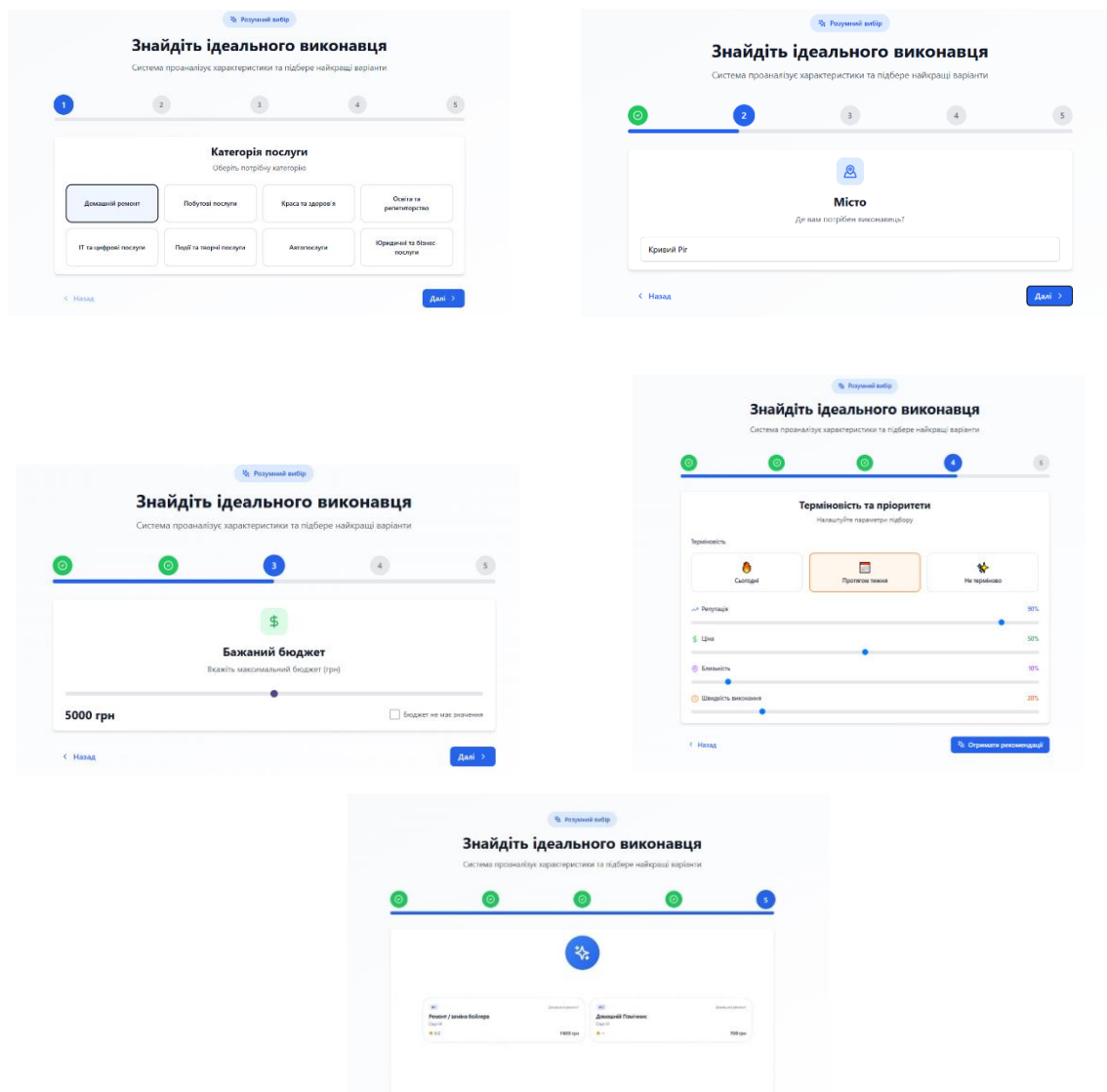


Рисунок 2.11 - Сторінка «Розумний вибір»

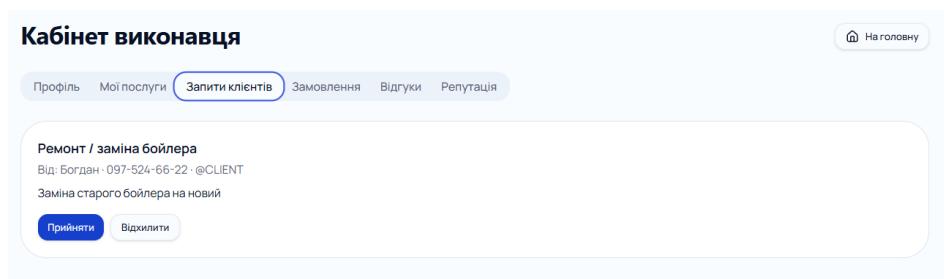


Рисунок 2.13 - Запит клієнта в кабінеті виконавця

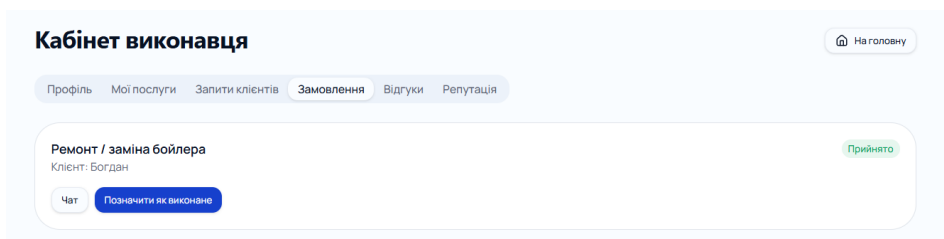


Рисунок 2.14 - Прийнята в опрацювання послуга в кабінеті виконавця

Після прийняття замовлення система відкриває можливість комунікації між клієнтом і виконавцем через чат. Чат використовується для демонстрації базової логіки обміну повідомленнями, що дозволяє сторонам узгоджувати деталі виконання робіт.

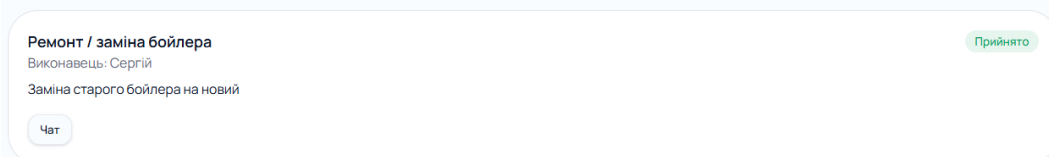


Рисунок 2.13 - Підтверджена і прийнята на опрацювання послуга в кабінеті клієнта

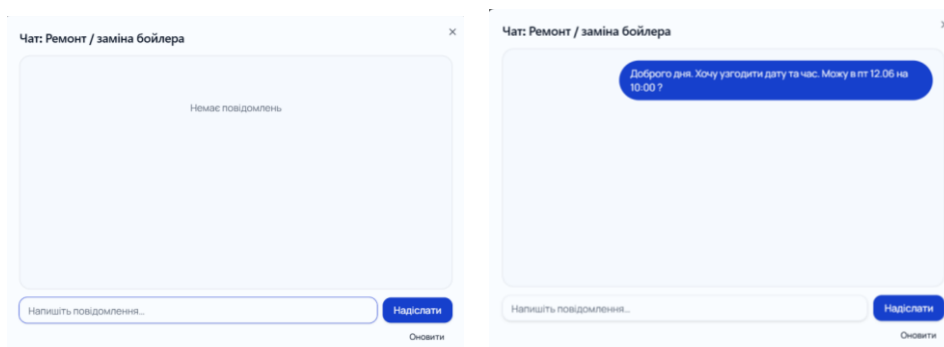


Рисунок 2.15 - Комунікації між клієнтом і виконавцем через чат

Після завершення виконання робіт виконавець змінює статус замовлення на «Виконано». Саме цей статус є підставою для відкриття клієнту можливості залишити оцінку та текстовий відгук. Такий підхід запобігає створенню відгуків без фактичної взаємодії між сторонами та підвищує достовірність репутаційних даних.

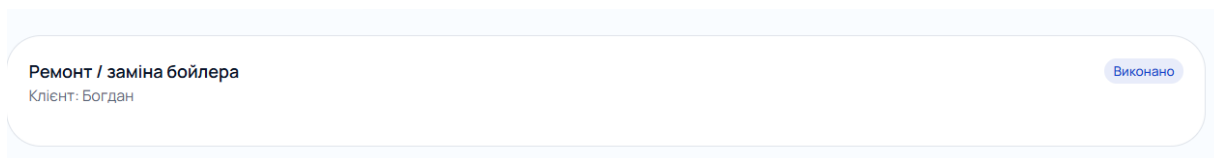


Рисунок 2.16 - Підтвердження виконання послуги у кабінеті виконавця

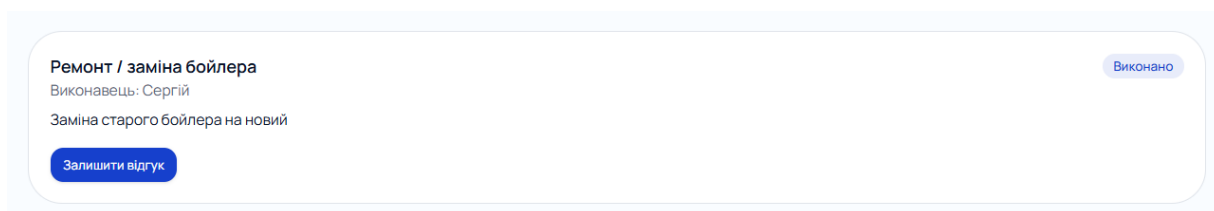


Рисунок 2.17 - Завершене замовлення у кабінеті клієнта

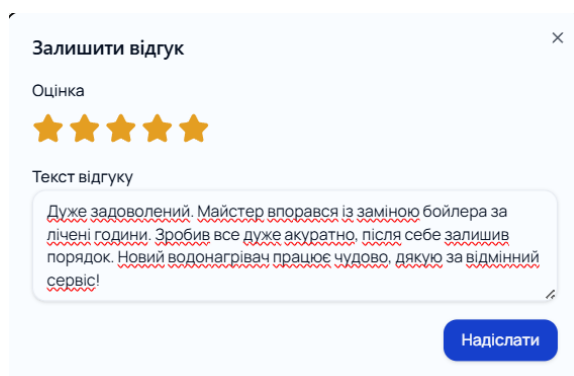


Рисунок 2.18 - Форма створення відгуку у кабінеті клієнта

Після залишення відгуку відповідна інформація прив'язується до конкретної послуги, а не до профілю виконавця загалом. Це дозволяє коректно оцінювати якість роботи виконавця в межах окремих напрямів діяльності. Наприклад, один і той самий користувач може надавати послуги у різних категоріях, однак репутаційні показники для кожної з них мають формуватися незалежно.

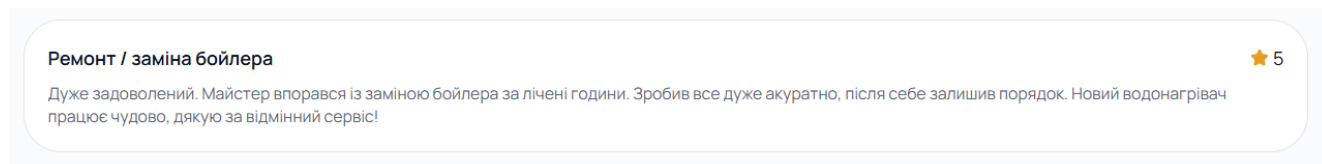


Рисунок 2.19 – Мої відгуки в профілі клієнта

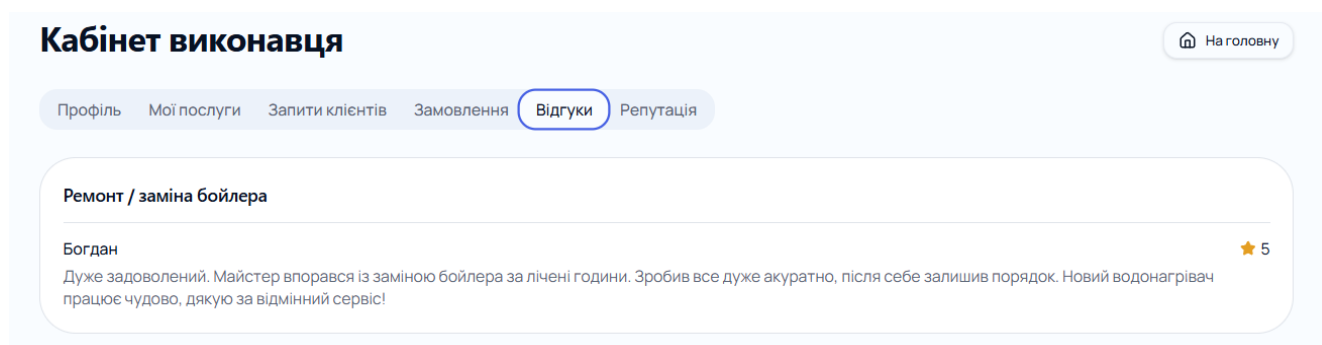


Рисунок 2.20 - Відгуки в профілі виконавця

У розділі «Репутація» кабінету виконавця відображаються агреговані показники за кожною послугою: середня оцінка клієнтів, кількість відгуків, кількість завершених замовлень, кількість отриманих запитів, стаж роботи та статус верифікації профілю. На основі цих даних система автоматично розраховує проміжні показники X_1 (довіра клієнтів), X_2 (професійна надійність) та X_3 (активність та доступність), які використовуються для формування інтегрального індексу репутації виконавця.

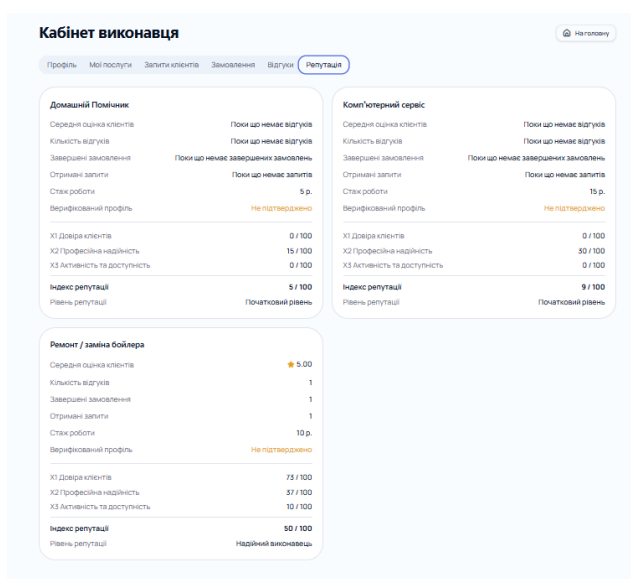


Рисунок 2.21 - Вкладка «Репутація» кабінету виконавця

На рисунку 2.22 представлено роботу репутаційного модуля інформаційної системи SmartChoice. Для кожної послуги автоматично розраховуються проміжні показники X_1 , X_2 та X_3 , після чого формується інтегральний індекс репутації Y та відповідний рівень виконавця. Залежно від отриманого значення система автоматично відносить виконавця до одного з рівнів: «Початковий рівень», «Базовий виконавець», «Надійний виконавець», «Досвідчений виконавець» або «Експерт». Отриманий індекс використовується під час ранжування послуг та підтримки процесу вибору виконавця клієнтом.

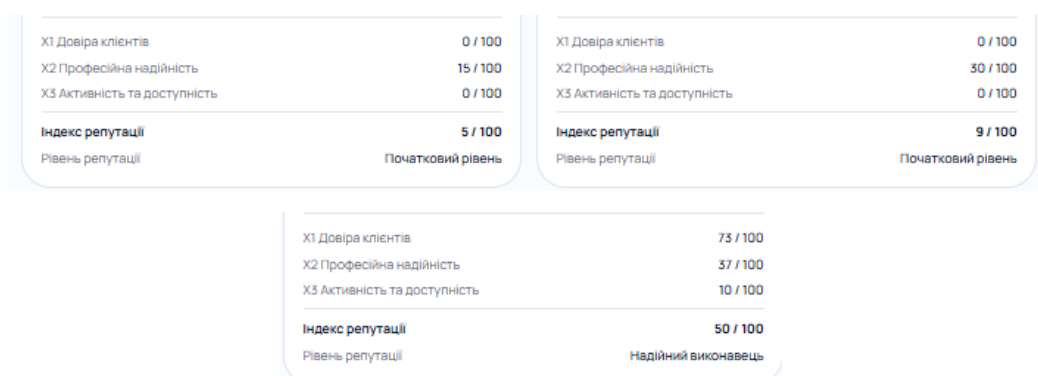


Рисунок 2.22 - Репутаційний модуль оцінювання виконавців

У результаті тестування було підтверджено, що система реалізує повний життєвий цикл взаємодії між клієнтом та виконавцем: від реєстрації користувача і створення послуги до формування замовлення, виконання робіт, створення відгуку та автоматичного розрахунку репутаційних показників. Проведене тестування підтвердило коректність роботи основних функціональних модулів, а також можливість накопичення та аналізу даних для підтримки прийняття рішень під час вибору виконавця послуг.

Таким чином, тестування підтвердило відповідність розробленої веб-орієнтованої інформаційної системи SmartChoice основним функціональним вимогам, сформульованим у попередніх підрозділах. Отримані результати демонструють практичну можливість використання системи для пошуку, оцінювання та вибору виконавців послуг, а також створюють основу для подальшого розвитку інтелектуальних механізмів рекомендацій та репутаційного оцінювання.

Перспективним напрямом подальшого розвитку інформаційної системи SmartChoice є розширення функціональних можливостей репутаційного модуля за рахунок накопичення більшого обсягу статистичних даних про взаємодію між клієнтами та виконавцями. Зі збільшенням кількості користувачів, послуг, замовлень та відгуків система зможе формувати більш точні та обґрунтовані репутаційні оцінки, що підвищить якість рекомендацій та ефективність підтримки прийняття рішень. Крім того, подальше вдосконалення може бути пов'язане з покращенням користувацького інтерфейсу, розширенням механізмів верифікації виконавців, впровадженням додаткових критеріїв оцінювання якості послуг та використанням більш складних методів інтелектуального аналізу даних для персоналізованого ранжування виконавців.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досліджено особливості функціонування сучасних онлайн-платформ пошуку послуг, проаналізовано існуючі підходи до оцінювання виконавців та визначено основні проблеми, пов'язані з вибором надійного постачальника послуг. Проведений аналіз показав, що більшість існуючих платформ використовують спрощені механізми оцінювання, які переважно базуються на середньому рейтингу та текстових відгуках користувачів, що не завжди забезпечує об'єктивне уявлення про якість роботи виконавця.

Метою роботи було розроблення веб-орієнтованої системи підтримки прийняття рішень щодо вибору постачальників послуг на основі нечітких моделей оцінки репутації. Для досягнення поставленої мети було виконано аналіз предметної області, сформовано вимоги до системи, спроектовано архітектуру програмного забезпечення, структуру бази даних та реалізовано основні функціональні модулі веб-системи SmartChoice.

У процесі розробки було створено інформаційну систему, яка забезпечує реєстрацію користувачів, керування профілями клієнтів і виконавців, створення та публікацію послуг, пошук і перегляд пропозицій, оформлення замовлень, обмін повідомленнями між учасниками взаємодії, формування відгуків та накопичення статистичних даних щодо діяльності виконавців.

Особливу увагу приділено реалізації репутаційного модуля, який є ключовим елементом системи підтримки прийняття рішень. Запропонований підхід передбачає формування інтегрального індексу репутації на основі трьох груп показників: довіри клієнтів, професійної надійності та активності виконавця. Для кожної послуги окремо розраховуються проміжні показники X_1 , X_2 та X_3 , які надалі використовуються для формування підсумкового індексу репутації Y . Такий підхід дозволяє здійснювати більш об'єктивне оцінювання виконавців порівняно з традиційними рейтинговими системами.

У межах роботи було сформовано основу для використання нечітких моделей оцінювання репутації виконавців. Запропонована система враховує декілька взаємопов'язаних критеріїв та дозволяє перейти від спрощеного рейтингового оцінювання до комплексного аналізу діяльності постачальників послуг. Реалізована модель може бути використана як база для подальшого впровадження повноцінного механізму нечіткого логічного виведення та адаптивного налаштування вагових коефіцієнтів.

Проведене тестування підтвердило працездатність основних функціональних модулів системи, коректність взаємодії між клієнтами та виконавцями, а також правильність розрахунку репутаційних показників на основі накопичених даних. Отримані результати свідчать про можливість практичного використання розробленої системи для підтримки процесу вибору постачальників послуг у різних сферах діяльності.

Практична цінність роботи полягає у створенні веб-орієнтованої інформаційної системи, яка дозволяє автоматизувати процес пошуку та оцінювання виконавців, накопичувати статистичну інформацію про якість наданих послуг і формувати обґрунтовані рекомендації для користувачів.

Перспективами подальшого розвитку системи є розширення репутаційної моделі за рахунок збільшення кількості критеріїв оцінювання, удосконалення алгоритмів персоналізованого ранжування, впровадження повноцінних механізмів нечіткої логіки для підтримки прийняття рішень, автоматичне налаштування вагових коефіцієнтів на основі накопичених даних, а також подальше вдосконалення користувацького інтерфейсу та функціональних можливостей системи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Плескач В. Л., Захарченко Ю. В. Електронна комерція : підручник. Київ : Знання, 2017. 511 с.
2. Братійчук М. В. та ін. Теорія прийняття рішень : підручник / за ред. М. В. Братійчука. Київ : Академвидав, 2015. 432 с.
3. Андрієнко В. М. Моделювання процесів прийняття рішень в управлінні соціально-економічними системами. *Економічний вісник Донбасу*. 2021. № 2 (64). С. 112–119.
4. Ситник В. Ф. Системи підтримки прийняття рішень : навч. посіб. Київ : КНЕУ, 2009. 614 с.
5. Turban E., King D., Lee J. K., Liang T. P., Turban D. C. *Electronic Commerce 2018: A Managerial and Social Networks Perspective*. Springer, 2018. 732 p.
6. Зайченко Ю. П. Нечіткі моделі і методи в інтелектуальних системах підтримки прийняття рішень : навч. посіб. Київ : Слово, 2019. 344 с.
7. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень : навч. посіб. Запоріжжя : ЗНТУ, 2008. 341 с.
8. OLX Україна : офіційний вебсайт. URL: <https://www.olx.ua> (дата звернення: 30.05.2026).
9. Turban E., Pollard C., Wood G. *Information Technology for Management: Driving Digital Transformation to Increase Local and Global Performance, Growth and Sustainability*. 12th ed. Hoboken : John Wiley & Sons, 2021. 560 p.
10. Kabanchik.ua : офіційний вебсайт сервісу пошуку виконавців послуг. URL: <https://kabanchik.ua> (дата звернення: 30.05.2026).
11. Power D. J. *Decision Support, Analytics, and Business Intelligence*. New York : Business Expert Press, 2019. 284 p.
12. Ricci F., Rokach L., Shapira B. *Recommender Systems Handbook*. 3rd ed. Cham : Springer, 2022. 1548 p.
13. Freelance.ua : офіційний вебсайт біржі фрилансу. URL: <https://freelance.ua> (дата звернення: 30.05.2026).

14. Jøsang A. Subjective Logic: A Formalism for Reasoning Under Uncertainty. Cham : Springer, 2018. 344 p.
15. Saaty T. L., Peniwati K. Group Decision Making: Drawing Out and Reconciling Differences. Pittsburgh : RWS Publications, 2017. 385 p.
16. Resnick P., Zeckhauser R. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System // *The Economics of the Internet and E-commerce*. Amsterdam : Elsevier, 2019. P. 127–157.
17. Saaty T. L. Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World. Pittsburgh : RWS Publications, 2016. 323 p.
18. Turban E., Sharda R., Delen D. Decision Support and Business Intelligence Systems. 11th ed. Boston : Pearson, 2021. 720 p.
19. Zadeh L. A. Fuzzy Sets Information and Control. 1965. Vol. 8. No. 3. P. 338–353.
20. Bayes T. An Essay towards Solving a Problem in the Doctrine of Chances // *Philosophical Transactions of the Royal Society of London*. 1763. Vol. 53. P. 370–418.
21. Wilson E. B. Probable Inference, the Law of Succession, and Statistical Inference // *Journal of the American Statistical Association*. 1927. Vol. 22. No. 158. P. 209–212.
22. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 880 p.
23. Kurose J. F., Ross K. W. Computer Networking: A Top-Down Approach. 8th ed. New York : Pearson, 2021. 864 p.
24. Banks A., Porcello E. Learning React: Modern Patterns for Developing React Apps. 2nd ed. Sebastopol : O'Reilly Media, 2020. 310 p.
25. Cherny B. Programming TypeScript: Making Your JavaScript Applications Scale. Sebastopol : O'Reilly Media, 2019. 324 p.
26. Supabase : Official Documentation. URL: <https://supabase.com/docs> (дата звернення: 31.05.2026).
27. PostgreSQL Global Development Group. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs> (дата звернення: 31.05.2026).
28. Marcotte E. Responsive Web Design. New York : A Book Apart, 2014. 150 p.