

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ
МЕДИЦИНИ ТА БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГЖИЦЬКОГО
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему:

«Розробка платформи для організації та проведення онлайн-аукціонів»

Виконав: студент 4 курсу
спеціальності 126 «Інформаційні
системи та технології»

_____ Дмитров І.І
(прізвище та ініціали)

Керівник: _____ Татомир А. В.
(прізвище та ініціали)

ДУБЛЯНИ 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Рівень вищої освіти перший (бакалаврський)
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

д.т.н., професор, Тригуба А. М.

(вч. звання, прізвище, ініціали)

“ ” _____ 202 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка платформи для організації та проведення онлайн-аукціонів»

керівник роботи к. т. н., доцент., Татомир А.В.

(наук.ступінь, вч. звання, прізвище, ініціали)

затверджені наказом Львівського НАУ _____

2. Строк подання студентом роботи _____

3. Вихідні дані: аналітичні дані роботи та характеристика об'єкту дослідження, опис бібліотек мов програмування, науково-технічна і довідкова література.

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

Вступ

1. Аналіз предметної області

2. Постановка задачі

3. Проектування платформи онлайн-аукціонів

4. Охорона праці

Висновки

Бібліографічний список

5. Перелік графічного матеріалу

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3	<i>Татомир А.В., к.т.н., доцент кафедри інформаційних технологій</i>			
4	<i>Городецький І.М., к.т.н., доцент кафедри інженерної механіки</i>			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Відмітка про виконання
1	<i>Отримання завдання. Вивчення рекомендованої літератури по темі роботи. Написання аналітичного огляду предметної області.</i>	<i>01.11.2024 – 15.12.2024</i>	
2	<i>Проектування та опис технічного завдання, функціональних вимог та технічної сторони реалізації проекту (написання проектної частини).</i>	<i>16.12.2024 – 05.02.2025</i>	
3	<i>Програмна реалізація проекту (вибір мови програмування, розробка сайту, тестування його роботи, розрахунок ефективності проекту)</i>	<i>06.02.2025 – 31.03.2025</i>	
4	<i>Розгляд питань з охорони праці та безпеки у надзвичайних ситуаціях</i>	<i>01.04.2025 – 30.04.2025</i>	
5	<i>Завершення оформлення основної частини, написання висновків та підготовка презентаційного матеріалу</i>	<i>01.05.2025 – 26.05.2025</i>	
6	<i>Завершення роботи в цілому. Підготовка до захисту кваліфікаційної роботи</i>	<i>27.05.2025 – 10.06.2025</i>	

Студент

(підпис)*Дмитров І.І.*
(прізвище та ініціали)

Керівник роботи

(підпис)*Татомир А.В.*
(прізвище та ініціали)

УДК 004.738.5:339.17

Кваліфікаційна робота: 53 сторінок текстової частини, 1 таблиця, 10 рисунків, 20 джерел літератури, 5 додатків.

«Розробка платформи для організації та проведення онлайн-аукціонів»
Дмитров І.І. – Кваліфікаційна робота. Кафедра інформаційних технологій.
Дубляни, Львівський національний університет ветеринарної медицини та біотехнологій імені С.З. Гжицького, 2025 р.

Подано аналіз сучасного стану розвитку онлайн-аукціонів та електронної комерції у сфері колекціонування. Проаналізовано предметну область, існуючі аналоги аукціонних платформ та визначено функціональні вимоги до веб-платформи. Здійснено проектування архітектури системи та бази даних у відповідності до принципів об'єктно-орієнтованого програмування та сучасних методик розробки.

Запропоновано використання технології WebSocket для забезпечення інтерактивної взаємодії в реальному часі під час проведення аукціонів. Розроблено алгоритм проведення торгів з урахуванням обробки одночасних ставок та виняткових ситуацій.

Здійснено аналіз травматичних ситуацій при виконанні різних робіт у сфері розробки програмного забезпечення з використанням комп'ютерної техніки, викладено питання охорони праці та безпеки в надзвичайних ситуаціях.

Ключові слова: веб-платформа, онлайн-аукціони, Nest.js, WebSocket, PostgreSQL, ООП, електронна комерція

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Дослідження сучасних тенденцій у сфері онлайн-аукціонів	7
1.2 Аналіз існуючих рішень та виявлення їх переваг і недоліків	9
1.3 Обґрунтування доцільності розробки нової системи.....	13
РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ	16
2.1 Мета та завдання розробки	16
2.2 Визначення цільової аудиторії та сценаріїв використання платформи	18
2.3 Вибір технологій розробки	21
РОЗДІЛ 3 ПРОЕКТУВАННЯ ПЛАТФОРМИ ОНЛАЙН-АУКЦІОНІВ	25
3.1 Моделювання взаємодії користувачів з платформою.....	25
3.2 Об'єктно-орієнтоване проектування аукціонної платформи.....	32
3.3 Проектування моделі бази даних	37
3.4 Розробка алгоритму проведення аукціону	28
3.5 Проектування користувацького інтерфейсу	41
РОЗДІЛ 4 ОХОРОНА ПРАЦІ.....	46
4.1 Аналіз стану виробничої санітарії і гігієни праці	46
4.2 Обґрунтування організаційно-технічних рекомендацій з охорони праці.....	48
4.3 Пожежна безпека	48
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	52
ДОДАТКИ.....	54

ВСТУП

В умовах стрімкого розвитку інформаційних технологій та електронної комерції особливого значення набувають системи, що забезпечують проведення торгівельних операцій у віртуальному просторі. Онлайн-аукціони наразі перетворилися на потужний економічний інструмент, який створює нові можливості для продавців і покупців, дозволяючи здійснювати транзакції незалежно від географічного розташування учасників. Актуальність розробки сучасних платформ для проведення аукціонів зумовлена постійно зростаючим попитом на електронні торгівельні майданчики та необхідністю впровадження технологічних інновацій у цій сфері.

Незважаючи на існування низки аукціонних платформ, більшість із них має суттєві обмеження щодо масштабованості, безпеки транзакцій, зручності користувацького інтерфейсу та інтерактивності. Часто такі системи не забезпечують належної прозорості процесу торгів або мають недостатню швидкодію при високих навантаженнях. Ці недоліки створюють підґрунтя для розробки вдосконаленої платформи, що враховуватиме сучасні технологічні можливості та потреби користувачів.

Метою даної кваліфікаційної роботи є розробка повнофункціональної платформи для організації та проведення онлайн-аукціонів із застосуванням сучасних технологій, що забезпечить високу продуктивність, інтерактивність та надійність сервісу.

Для досягнення поставленої мети необхідно вирішити такі завдання: провести аналіз існуючих рішень у сфері онлайн-аукціонів та виявити їхні переваги й недоліки; сформулювати функціональні та нефункціональні вимоги до платформи; обґрунтувати вибір технологічного стеку для реалізації проекту; спроектувати архітектуру системи та модель бази даних; розробити механізм проведення аукціонів у реальному часі; імплементувати функціональність платформи; провести тестування та оптимізацію розробленого рішення.

Об'єктом дослідження виступають процеси організації та проведення електронних аукціонів у мережі Інтернет.

Предметом дослідження є методи та засоби розробки інтерактивних веб-платформ для проведення онлайн-аукціонів із використанням технологій Nest.js, WebSocket, PostgreSQL та TypeORM.

Інформаційною базою дослідження слугують наукові праці вітчизняних і закордонних авторів, технічна документація використовуваних технологій, статистичні дані щодо функціонування існуючих аукціонних платформ, а також нормативно-правові акти, що регулюють електронну комерцію.

Практична цінність отриманих результатів полягає у створенні платформи, що дозволить підвищити ефективність проведення онлайн-аукціонів за рахунок впровадження технологій реального часу, забезпечить надійність зберігання даних та високу інтерактивність взаємодії користувачів із системою. Запропонована платформа може бути використана для проведення комерційних аукціонів різного спрямування, благодійних торгів, реалізації конфіскованого майна тощо.

Відмінністю пропонованого рішення від існуючих аналогів є використання сучасного стеку технологій (Nest.js, WebSocket, PostgreSQL, TypeORM), що забезпечує високу продуктивність системи та можливість роботи з великою кількістю одночасних користувачів. Архітектурні рішення, що впроваджуються у розроблюваній платформі, дозволяють досягти вищого рівня безпеки транзакцій та захисту даних користувачів порівняно з існуючими системами.

Розроблена платформа демонструє можливість ефективного поєднання технологій серверної обробки даних із механізмами взаємодії в реальному часі, що створює підґрунтя для подальшого розвитку не лише аукціонних систем, але й інших інтерактивних веб-платформ.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження сучасних тенденцій у сфері онлайн-аукціонів

Онлайн-аукціони стали невід'ємною частиною не лише світового ринку колекційних цінностей, але й сучасної системи управління державними та муніципальними активами. Вони забезпечують прозору платформу для взаємодії між різними учасниками ринку незалежно від їхнього географічного розташування. На відміну від традиційних форм торгів, віртуальні аукціонні майданчики пропонують значно ширші можливості для участі, суттєво нижчі вхідні бар'єри та цілодобовий доступ до процесу торгів [1].

Особливе значення онлайн-аукціони набули у сфері управління земельними ресурсами та державним майном. Органи місцевого самоврядування активно використовують електронні аукціонні платформи для прозорого розпорядження комунальною власністю, оренди земельних ділянок комунального значення та продажу муніципального майна. Такий підхід забезпечує максимальну відкритість процедур, мінімізує корупційні ризики та дозволяє отримати справедливую ринкову вартість за об'єкти торгів.

Аграрні підприємства знайшли в онлайн-аукціонах ефективний інструмент для розширення земельного банку та оптимізації структури землекористування. Можливість брати участь у торгах за оренду державних та комунальних земель сільськогосподарського призначення без необхідності фізичної присутності значно спростила процедури для сільгоспвиробників. Це особливо актуально для великих агрохолдингів, які ведуть діяльність у різних регіонах країни та потребують оперативного доступу до земельних ресурсів.

Державні земельні аукціони через електронні платформи забезпечили революційні зміни у сфері земельних відносин. Прозорість процедур, можливість онлайн-участі для учасників з різних регіонів та автоматизація процесу визначення переможця суттєво підвищили ефективність розподілу

земельних ресурсів. Аграрні підприємства отримали рівні можливості для участі у торгах, що сприяло формуванню справедливих ринкових цін на оренду сільськогосподарських угідь.

Аналіз сучасного стану розвитку онлайн-аукціонів дозволяє виявити низку важливих тенденцій. Насамперед спостерігається активне впровадження технологій реального часу, які дозволяють учасникам відстежувати перебіг торгів без затримок. Використання WebSocket забезпечує миттєве оновлення інформації про поточні ставки без необхідності оновлення сторінки, що створює ефект присутності та підвищує динамічність процесу торгів за земельні ділянки.

Сучасні онлайн-аукціони приділяють значну увагу вдосконаленню системи безпеки та верифікації користувачів. Використання біометричних даних для ідентифікації, впровадження багатофакторної автентифікації та залучення незалежних експертів для підтвердження автентичності лотів стали невід'ємними елементами розвинутих аукціонних платформ [2].

Паралельно відбувається формування спеціалізованих аукціонних майданчиків, орієнтованих на конкретні сегменти ринку. Такі платформи забезпечують більш точну оцінку об'єктів торгів, адекватну експертизу земельних ділянок та формують професійні спільноти учасників за галузевими інтересами. Це сприяє підвищенню довіри до системи та створює комфортне середовище для взаємодії фахівців у сфері земельних відносин.

Зростання популярності мобільних пристроїв стимулювало розробку спеціалізованих додатків для участі в аукціонах. Можливість слідкувати за торгами та робити ставки з будь-якого місця суттєво розширила аудиторію потенційних учасників, що особливо важливо для аграрних підприємств, керівництво яких часто перебуває у польових умовах або віддалених регіонах.

Сучасні онлайн-аукціони приділяють значну увагу вдосконаленню системи безпеки та верифікації користувачів. Використання електронних цифрових підписів, впровадження багатофакторної автентифікації та інтеграція з державними реєстрами стали невід'ємними елементами

розвинутих аукціонних платформ, що особливо актуально для торгів державним та комунальним майном.

Інтеграція з державними інформаційними системами дозволила створити нові можливості для автоматизації процедур. Автоматична перевірка правоздатності учасників через реєстри юридичних осіб, валідація фінансових гарантій через банківські системи та електронне оформлення результатів торгів значно підвищили ефективність аукціонних процедур для органів місцевого самоврядування.

Також спостерігається тенденція до розвитку додаткових сервісів навколо основної функціональності аукціонів. Консультації щодо оформлення документів, юридичний супровід угод, технічна експертиза земельних ділянок та фінансові послуги формують комплексну екосистему, що суттєво підвищує зручність користування сервісом для аграрних підприємств та органів місцевого самоврядування.

Таким чином, сучасний ринок онлайн-аукціонів характеризується динамічним розвитком та активним впровадженням інноваційних технологій у сфері управління державними та комунальними ресурсами. Розробка нової платформи для організації та проведення онлайн-аукціонів потребує врахування цих тенденцій для створення універсального продукту. Особлива увага має приділятися впровадженню технологій реального часу, забезпеченню надійного захисту даних та розробці інтуїтивно зрозумілого інтерфейсу.

1.2 Аналіз існуючих рішень та виявлення їх переваг і недоліків

В процесі розробки платформи для організації та проведення онлайн-аукціонів необхідно ретельно проаналізувати існуючі рішення на ринку. Такий аналіз дозволяє виявити сильні та слабкі сторони конкурентів, визначити незадоволені потреби користувачів та сформулювати конкурентні переваги

нового продукту. Розглянемо найбільш впливові платформи, що працюють у сфері онлайн-аукціонів, включаючи як комерційні майданчики для колекціонерів, так і державні системи для торгівлі земельними ділянками та майном.

Система "Прозорро.Продажі" представляє собою ключову державну платформу України для проведення електронних аукціонів з продажу державного та комунального майна. Заснована у 2018 році, ця система функціонує за дворівневою архітектурою (рис. 1.1). На першому рівні розташована центральна база даних (ЦБД), у якій проходять аукціони та зберігається вся інформація про торги. На другому рівні діють незалежні електронні майданчики, які забезпечують інтерфейс для користувачів та взаємодіють з центральною базою даних.



Рисунок 1.1 – Схема, як працює система в Prozorro.Sale

Ключовими перевагами системи "Прозорро.Продажі" є максимальна прозорість процедур, інтеграція з державними реєстрами для автоматичної перевірки учасників, використання електронних цифрових підписів для забезпечення безпеки та можливість оскарження результатів торгів. Система забезпечує рівні умови для всіх учасників та мінімізує корупційні ризики при розпорядженні державним майном.

Завдяки дворівневій структурі ЕТС, заведений організатором лот автоматично з'являється на сайтах всіх електронних майданчиків одночасно і стає доступним кожному покупцю. Відтак система не взаємодіє з учасниками

та організаторами торгів, а відповідає виключно за якісну та безперебійну роботу ЦБД, у якій відбуваються аукціони [16].

Проте система має певні обмеження, пов'язані з складністю процедур для нових користувачів, необхідністю отримання електронного цифрового підпису та жорсткими вимогами до документообігу, що може відлякувати потенційних учасників торгів.

Платформа UA.Land є спеціалізованим майданчиком для торгівлі земельними ділянками та нерухомістю в Україні. Ця платформа орієнтована переважно на аграрний сектор та забезпечує проведення аукціонів з оренди державних земель сільськогосподарського призначення. UA.Land пропонує зручний інтерфейс для пошуку земельних ділянок за географічним розташуванням, площею та цільовим призначенням.

Основними перевагами UA.Land є спеціалізація на земельних ресурсах, інтеграція з кадастровими картами, можливість перегляду детальної інформації про земельні ділянки та їх історію використання (рис. 1.2). Платформа також надає аналітичні інструменти для оцінки ринкової вартості земель та прогнозування тенденцій ринку.

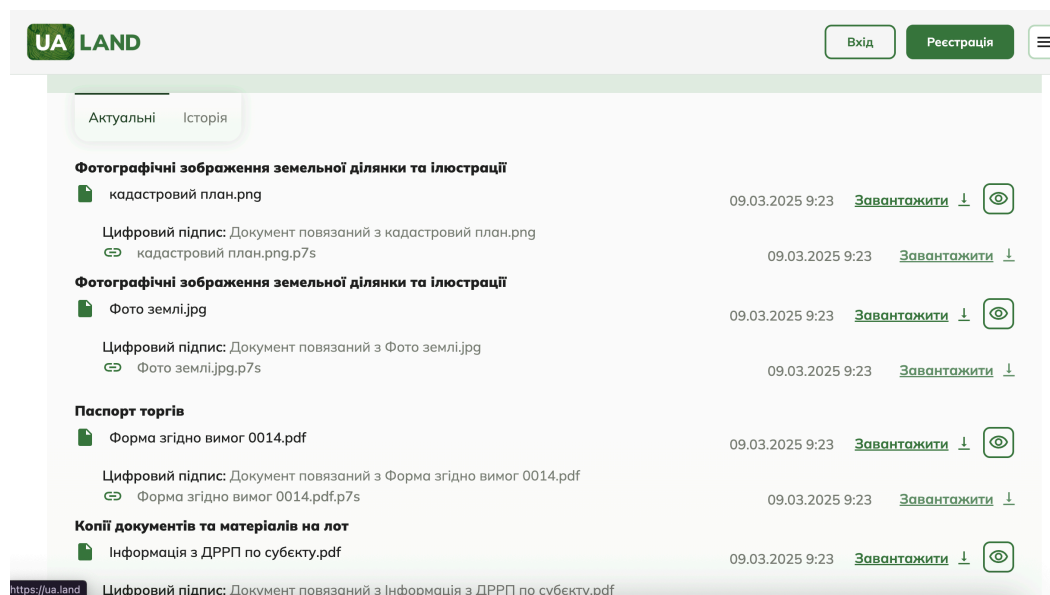


Рисунок 1.2 – Перегляд документів на землю на UA.Land

До недоліків можна віднести обмежену географію роботи, фокус виключно на українському ринку та відсутність функціональності для інших типів майна крім земельних ділянок.

Майданчик Zakupivli.pro функціонує як комплексна платформа для електронних торгів, що включає не лише аукціони, але й тендерні процедури [17]. Платформа обслуговує як державний, так і приватний сектори, забезпечуючи проведення різноманітних типів торгів від продажу майна до закупівель товарів та послуг (рис. 1.3).

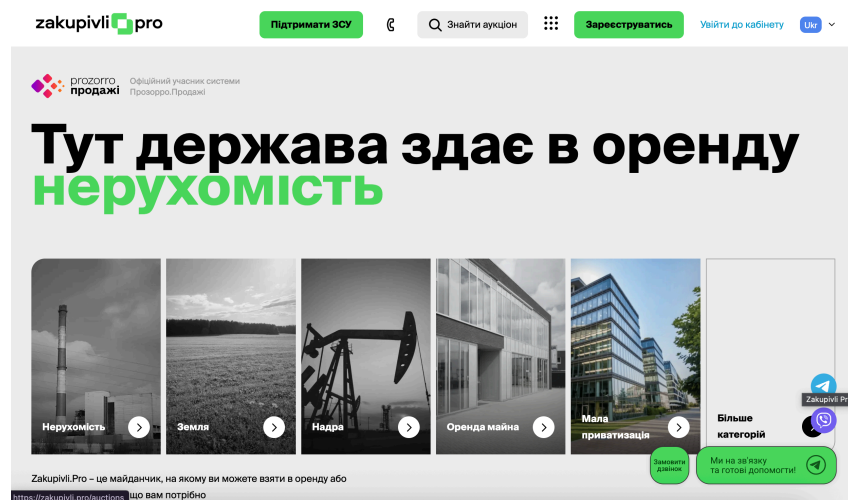


Рисунок 1.3 – Категорії аукціонів на Zakupivli.pro

Перевагами Zakupivli.pro є універсальність платформи, підтримка різних типів процедур, інтеграція з банківськими системами для автоматичного внесення гарантійних внесків та розвинена система аналітики торгів. Платформа також пропонує консультаційні послуги для учасників торгів.

Недоліками є складність інтерфейсу через намагання охопити всі можливі типи торгів, що може ускладнювати використання для спеціалізованих потреб, та відносно високі комісійні збори за користування платформою.

1.3 Обґрунтування доцільності розробки нової системи

Як показує аналіз альтернатив у п.1.2, існуючі аукціонні платформи мають суттєві обмеження, що створюють передумови для розробки нової системи. Основні напрямки вдосконалення включають технологічні інновації, покращення користувацького досвіду, гнучку автентифікацію, універсальність, сучасну архітектуру, соціальні функції та мобільність.

Впровадження технологій реального часу (WebSocket) замість застарілих підходів з періодичним оновленням сторінки забезпечить безперервну взаємодію користувачів з аукціонним процесом та підвищить їх емоційну залученість [4].

Розробка інтуїтивно зрозумілого інтерфейсу на основі принципів UX-дизайну зменшить бар'єр входу для нових користувачів та спростить навігацію порівняно з перевантаженими системами на кшталт Zakupivli.pro.

Впровадження багатофакторної автентифікації з можливістю вибору рівня безпеки залежно від типу торгів дозволить поєднати доступність приватних платформ з надійністю державних систем без обов'язкових вимог до ЕЦП.

На відміну від вузькоспеціалізованих рішень (UA.Land) або перевантажених універсальних платформ, нова система забезпечить гнучкі налаштування для різних типів аукціонів при збереженні простоти використання.

Використання стеку Nest.js, TypeORM, PostgreSQL та WebSocket забезпечить високу продуктивність, масштабованість та можливості для майбутнього розширення функціоналу без кардинальних змін у базовій структурі [5,6].

Інтеграція елементів соціальної взаємодії створить унікальне середовище для формування спільнот та обміну досвідом, що відсутнє в усіх аналізованих платформах.

Проектування з підходом "mobile-first" забезпечить оптимальний досвід користувачів на всіх типах пристроїв, враховуючи зростання частки мобільного трафіку.

Поєднання прозорості державних систем, спеціалізації галузевих платформ та зручності сучасних технологій дозволить створити універсальне рішення, що задовольнить потреби різних категорій користувачів та забезпечить якісно новий рівень взаємодії в сфері онлайн-аукціонів.

У таблиці 1.1 наведена порівняльна характеристика існуючих аукціонних платформ та планованої системи за основними функціональними та технічними критеріями.

Таблиця 1.1 - Порівняльна характеристика аукціонних платформ

Критерій	Prozorro.Sale	UA.Land	Zakupivli.pro	Розроблювана платформа
Основна аудиторія	Державні органи	Аграрні підприємства	Універсальна	Універсальна
Складність інтерфейсу	Висока	Середня	Висока	Низька
Підтримка реального часу	Ні	Ні	Обмежено	Так
Вимоги до реєстрації	ЕЦП обов'язковий	Стандартна	Складна	Стандартна
Спеціалізація	Державне майно	Земельні ділянки	Універсальна	Універсальна
Мобільна версія	Обмежена	Частково	Ні	Планується

Продовження таблиці 1.1

Соціальні функції	Відсутні	Відсутні	Відсутні	Інтегровані
Гнучкість налаштувань	Низька	Низька	Середня	Висока
Підтримка різних типів торгів	Обмежена	Тільки земля	Широка	Широка

Як видно з таблиці 1.1, розробка нової платформи для організації та проведення онлайн-аукціонів є доцільною та перспективною з огляду на виявлені обмеження існуючих рішень.

РОЗДІЛ 2

ПОСТАНОВКА ЗАДАЧІ

2.1 Мета та завдання розробки

Головною метою розробки платформи для організації та проведення онлайн-аукціонів є створення сучасного, високопродуктивного та інтуїтивно зрозумілого веб-сервісу, що забезпечить функціональні можливості для проведення аукціонів у режимі реального часу, формування активної спільноти та максимальної безпеки транзакцій між учасниками.

Відповідно до сформульованої мети та проведеного аналізу предметної області визначено комплекс ключових завдань розробки. Першочерговим завданням є проектування та реалізація архітектури системи з використанням сучасного технологічного стеку, що включає NestJS для серверної частини, WebSocket для комунікації в реальному часі та PostgreSQL з TypeORM для управління даними. Архітектура має забезпечувати високу продуктивність, масштабованість та надійність при зростанні навантаження.

Важливим аспектом проекту є розробка гнучкої та розширюваної моделі даних, що дозволить ефективно зберігати та обробляти інформацію про користувачів, аукціони, ставки, категорії товарів та транзакції. Модель даних має передбачати можливість подальшого додавання нових сутностей без суттєвої реструктуризації системи, що забезпечить гнучкість та адаптивність платформи до змінних вимог ринку та потреб користувачів [7,8].

Особливу увагу слід приділити створенню інтуїтивно зрозумілого інтерфейсу користувача, що забезпечить зручну навігацію та взаємодію з платформою як для новачків, так і для досвідчених користувачів. Інтерфейс має враховувати принципи адаптивного дизайну для оптимального відображення на різних пристроях, що є критично важливим в умовах зростання частки мобільного трафіку.

Ключовою технологічною перевагою розроблюваної платформи стане впровадження механізму проведення аукціонів у реальному часі з використанням технології WebSocket. Це дозволить забезпечити миттєве оновлення інформації про поточні ставки, залишок часу до завершення аукціону та інших параметрів торгів без необхідності оновлення сторінки, створюючи ефект присутності та підвищуючи емоційну залученість учасників.

Безпека користувачів та їхніх даних є пріоритетним завданням при розробці системи. Необхідно створити надійну систему автентифікації та авторизації з багатофакторною перевіркою, що забезпечить високий рівень захисту облікових записів та попередить несанкціонований доступ до персональних даних. Також важливим аспектом безпеки є розробка механізму рейтингування користувачів на основі їхньої активності, успішних транзакцій та зворотного зв'язку від інших учасників, що підвищить рівень довіри між користувачами та мінімізує ризики шахрайства.

Для зручності пошуку та навігації необхідно впровадити розвинену систему категоризації та пошуку товарів з можливістю фільтрації за різними параметрами. Це спростить користувачам процес знаходження потрібних предметів відповідно до їхніх колекційних інтересів та сприятиме підвищенню активності на платформі.

Соціальний аспект платформи реалізується через функціонал для управління персональними колекціями користувачів, включаючи можливість каталогізації власних предметів, відстеження бажаних лотів та аналізу цінової динаміки на подібні товари. Також необхідно розробити систему сповіщень для інформування користувачів про важливі події, такі як поява нових лотів у бажаних категоріях, зміни статусу аукціонів, в яких вони беруть участь, перевищення їхніх ставок іншими учасниками.

Фінансова складова проекту передбачає інтеграцію платіжної системи для безпечного проведення транзакцій між користувачами, з підтримкою різноманітних способів оплати та захистом від шахрайських дій. Це є

критично важливим для формування довіри до платформи та забезпечення її комерційної успішності.

Для постійного вдосконалення платформи необхідно впровадити аналітичні інструменти для відстеження активності користувачів, популярності різних категорій товарів та ефективності проведення аукціонів. Також важливим завданням є розробка адміністративної панелі для модерації контенту, управління користувачами, категоріями та вирішення спірних ситуацій, що забезпечить підтримку здорової атмосфери на платформі.

Завершальним етапом розробки стане проведення комплексного тестування системи, включаючи функціональне, інтеграційне та навантажувальне тестування, для виявлення та усунення потенційних проблем перед запуском платформи. Також необхідно забезпечити документування коду та створення технічної документації для подальшої підтримки та розвитку системи.

Успішне виконання цих завдань дозволить створити конкурентоспроможну платформу для організації та проведення онлайн-аукціонів, що відповідатиме сучасним технологічним стандартам та задовольнятиме потреби цільової аудиторії.

2.2 Визначення цільової аудиторії та сценаріїв використання платформи

Ефективна розробка та впровадження платформи для організації та проведення онлайн-аукціонів вимагає чіткого розуміння потенційних користувачів та їхніх потреб. Детальний аналіз цільової аудиторії дозволяє сформулювати функціональні вимоги до системи, оптимізувати користувацький досвід та забезпечити максимальну відповідність платформи очікуванням різних категорій учасників ринку.

Цільова аудиторія розроблюваної платформи є досить різноманітною та включає як традиційних учасників аукціонного ринку, так і представників

державного та приватного секторів (рис. 2.1). Органи місцевого самоврядування становлять важливий сегмент користувачів, які потребують прозорих та ефективних інструментів для розпорядження комунальним майном. Їхньою основною мотивацією є забезпечення максимальної прозорості процедур, отримання справедливої ринкової вартості за об'єкти торгів та мінімізація корупційних ризиків.

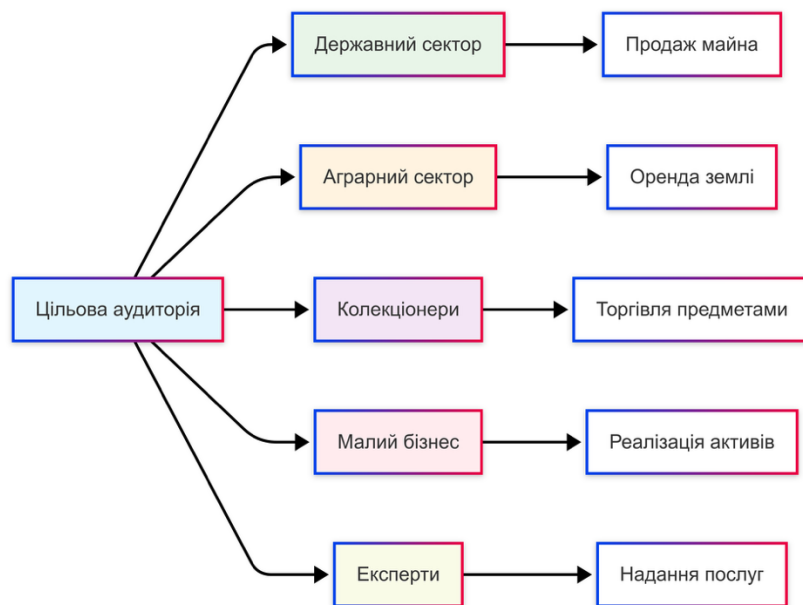


Рисунок 2.1 – Можлива цільова аудиторія та її потреби

Аграрні підприємства та сільгоспвиробники представляють ключовий комерційний сегмент, зацікавлений у доступі до земельних ресурсів через аукціонні механізми. Для них критично важливими є можливість участі в торгах за оренду державних та комунальних земель, простота процедур реєстрації та участі, а також доступ до аналітичної інформації про земельний ринок.

Колекціонери різноманітних предметів залишаються традиційним ядром користувацької бази аукціонних платформ. Це збирачі антикваріату, нумізмати, філателісти, колекціонери мистецтва та інших цінностей. Їхньою основною мотивацією є можливість знаходження рідкісних предметів,

відстеження ринкових тенденцій та взаємодія з однодумцями в професійному середовищі.

Державні установи та організації використовують аукціонні платформи для реалізації конфіскованого майна, продажу непрофільних активів та проведення інших процедур відчуження державної власності. Для них ключовими вимогами є дотримання законодавчих норм, забезпечення прозорості процедур та інтеграція з існуючими державними інформаційними системами.

Ділери та професійні продавці представляють важливий комерційний сегмент, що потребує доступу до широкої аудиторії покупців, інструментів для аналізу ринку та можливості проведення аукціонів з оптимальними умовами. Цей сегмент особливо чутливий до розміру комісійних зборів та функціональності платформи.

Приватні особи та малий бізнес, які час від часу потребують продажу майна або придбання специфічних товарів, становлять сегмент з підвищеними вимогами до простоти використання платформи та мінімальних бар'єрів входу.

На основі визначеної цільової аудиторії можна сформулювати ключові сценарії використання платформи. Сценарій державних торгів передбачає, що представник органу місцевого самоврядування створює аукціон з продажу комунального майна, завантажує необхідну документацію, встановлює параметри торгів відповідно до законодавчих вимог, контролює процес торгів та оформлює результати відповідно до встановленого порядку.

Сценарій земельного аукціону включає створення торгів за право оренди земельної ділянки, інтеграцію з кадастровими системами для отримання технічних характеристик ділянки, участь аграрних підприємств у торгах та оформлення договорів оренди з переможцями. Особливістю цього сценарію є необхідність роботи з великими обсягами картографічної інформації та технічними характеристиками земельних ділянок.

Сценарій колекційного аукціону зберігає традиційні елементи: користувач-колекціонер створює аукціон антикварного предмету, надає детальний опис та фотографії, встановлює параметри торгів, взаємодіє з потенційними покупцями через систему запитань та коментарів, завершує угоду з переможцем та отримує відгуки про якість сервісу.

Сценарій експертної підтримки передбачає залучення незалежних експертів для оцінки складних об'єктів торгів, проведення експертиз автентичності колекційних предметів, консультування щодо ринкової вартості та надання висновків для учасників торгів.

Розуміння різноманітних сценаріїв використання платформи дозволяє структурувати її функціонал таким чином, щоб забезпечити максимальну зручність та ефективність для всіх категорій користувачів. Особливу увагу слід приділити гнучкості налаштувань для різних типів торгів, інтеграції з зовнішніми системами, забезпеченню прозорості процедур та створенню універсального інструменту, що задовольнятиме потреби як державного, так і приватного секторів.

Таким чином, визначення цільової аудиторії та сценаріїв використання платформи становить фундаментальну основу для подальшого проектування універсальної системи, що забезпечуватиме ефективну роботу різних категорій користувачів та сприятиме розвитку прозорого та конкурентного аукціонного ринку в Україні.

2.3 Вибір технологій розробки

Вибір технологій для розробки платформи онлайн-аукціонів є стратегічно важливим рішенням, що визначає архітектурні можливості системи, продуктивність, масштабованість та перспективи подальшого розвитку проекту. Після ретельного аналізу сучасних інструментів та

фреймворків було сформовано технологічний стек, що найкраще відповідає функціональним та нефункціональним вимогам до платформи.

Для розробки серверної частини було обрано фреймворк Nest.js, який побудований на основі Node.js та впроваджує принципи об'єктно-орієнтованого програмування, функціонального програмування та функціонально-реактивного програмування. Nest.js забезпечує структуровану архітектуру на основі модулів, що значно полегшує підтримку та розширення кодової бази. Фреймворк також надає широкі можливості для створення REST API, WebSocket серверів та мікросервісів, що є критично важливим для реалізації інтерактивних аукціонних функцій.

Особливою перевагою Nest.js є вбудована підтримка TypeScript, що дозволяє використовувати статичну типізацію, декоратори та інші передові мовні можливості. Це сприяє підвищенню якості коду, зменшенню кількості помилок під час виконання та покращенню процесу розробки через автодоповнення та статичний аналіз коду. Крім того, Nest.js має потужну систему внутрішніх залежностей (dependency injection), що спрощує тестування та дотримання принципів SOLID.

Для забезпечення двосторонньої комунікації в реальному часі між сервером та клієнтом було обрано технологію WebSocket. На відміну від традиційного HTTP-підходу, WebSocket підтримує постійне з'єднання, що дозволяє серверу миттєво надсилати оновлення клієнтам без необхідності постійних запитів з їхнього боку. Ця технологія є ідеальною для реалізації аукціонної системи, де важлива швидкість передачі інформації про нові ставки, зміну статусу аукціону та інші події, що відбуваються в реальному часі.

Nest.js надає вбудовану підтримку WebSocket через інтеграцію з бібліотекою Socket.io, що забезпечує надійне з'єднання з автоматичним відновленням у разі розриву, підтримкою різних транспортних протоколів та можливістю групової розсилки повідомлень. Це дозволяє створити ефективну

систему сповіщень та оновлень для учасників аукціонів, що значно покращує користувацький досвід.

Для зберігання даних обрано реляційну систему управління базами даних PostgreSQL, яка відзначається високою надійністю, відповідністю стандартам SQL та потужними можливостями для роботи з різними типами даних. PostgreSQL підтримує транзакції, складні запити, індексацію та забезпечує цілісність даних, що є критично важливим для фінансових операцій на аукціонній платформі. Крім того, PostgreSQL має розширену підтримку JSON, що дозволяє гнучко зберігати напівструктуровані дані, коли це необхідно.

Для взаємодії з базою даних було обрано ORM-фреймворк TypeORM, який спеціально розроблений для TypeScript та забезпечує типізований доступ до бази даних. TypeORM підтримує шаблон проектування Active Record та Data Mapper, що дозволяє гнучко структурувати код відповідно до потреб проекту. Використання TypeORM суттєво спрощує операції з базою даних, автоматизує міграції, забезпечує зручний API для створення запитів та відносин між сутностями, що прискорює процес розробки та зменшує ймовірність помилок.

Для забезпечення безпеки системи планується використання JWT (JSON Web Tokens) для автентифікації та авторизації користувачів. JWT дозволяє передавати інформацію про користувача між клієнтом та сервером у вигляді підписаного токена, що значно спрощує процес аутентифікації та дозволяє реалізувати безстатевий (stateless) підхід до управління сесіями [2,9,10].

Для розробки клієнтської частини буде використано сучасний JavaScript-фреймворк, який забезпечить створення інтерактивного та адаптивного інтерфейсу. Фреймворк має підтримувати TypeScript, компонентний підхід до розробки та ефективну взаємодію з WebSocket, що дозволить забезпечити плавне та реактивне оновлення інтерфейсу під час аукціонних процесів.

Також планується використання Docker для контейнеризації додатку, що дозволить створити ізольоване середовище для розробки, тестування та розгортання. Це значно спростить процес розгортання системи на різних середовищах та забезпечить стабільність роботи незалежно від конфігурації сервера.

Для забезпечення якості коду та процесу розробки будуть використовуватися інструменти статичного аналізу коду (ESLint, Prettier), системи неперервної інтеграції та розгортання (CI/CD), а також фреймворки для тестування (Vitest, Supertest).

Обраний технологічний стек забезпечує оптимальний баланс між продуктивністю, надійністю, масштабованістю та швидкістю розробки. Використання сучасних технологій дозволить створити високоякісну платформу для проведення онлайн-аукціонів, що відповідатиме високим вимогам користувачів та матиме потенціал для подальшого розвитку та вдосконалення.

РОЗДІЛ 3

ПРОЕКТУВАННЯ ПЛАТФОРМИ ОНЛАЙН-АУКЦІОНІВ

3.1 Моделювання взаємодії користувачів з платформою

Моделювання взаємодії є критично важливим етапом проектування платформи онлайн-аукціонів, оскільки дозволяє детально описати часову послідовність дій користувачів та відповідних реакцій системи. Використання діаграм послідовності у нотації UML забезпечує наочне представлення комунікаційних потоків між різними компонентами системи та дозволяє виявити потенційні проблеми взаємодії на етапі проектування (рис. 3.1).

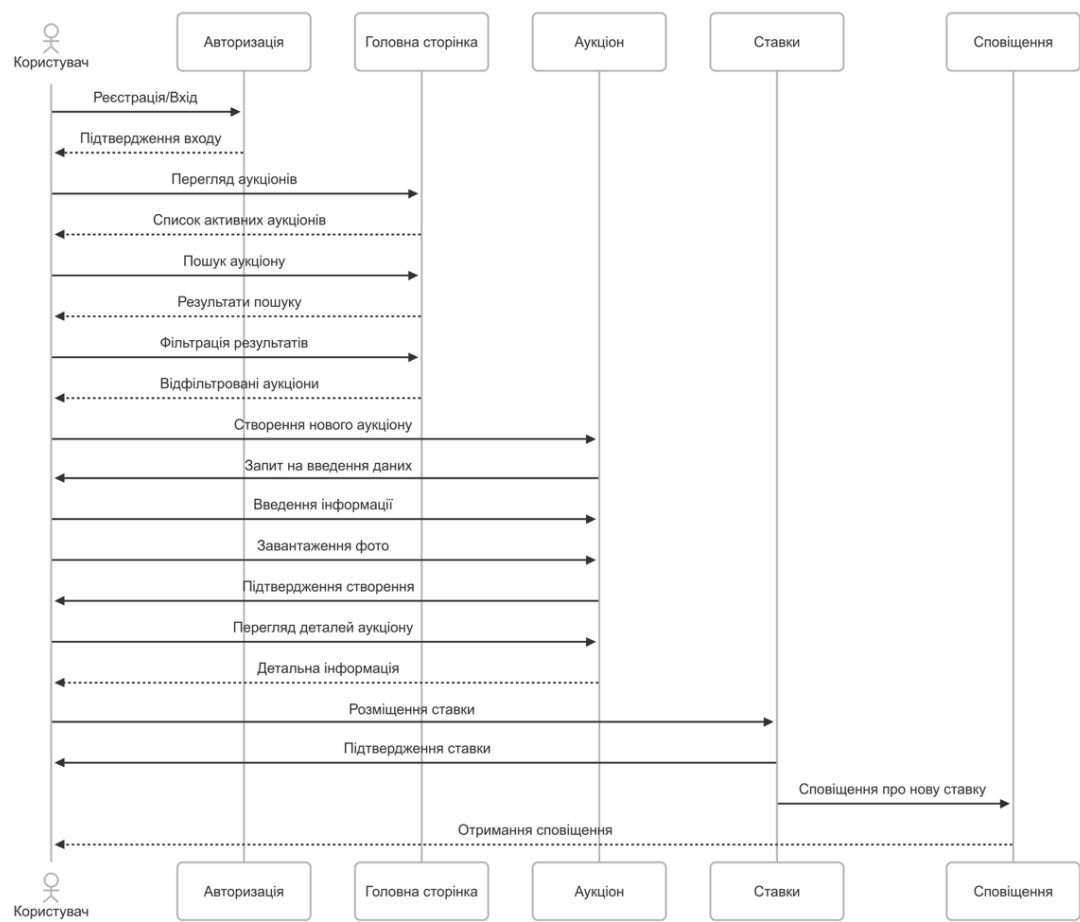


Рисунок 3.1 - UML діаграма послідовності користування платформою

Процес автентифікації та авторизації є початковим етапом взаємодії користувача з платформою. Коли користувач ініціює процес реєстрації або входу в систему, він відправляє відповідний запит до компонента авторизації. Система обробляє надані облікові дані, проводить валідацію інформації у базі даних, генерує JWT токен у випадку успішної автентифікації та повертає підтвердження входу користувачу. Цей процес забезпечує безпеку платформи та створює персоналізований досвід для кожного користувача.

Навігація по головній сторінці представляє основний сценарій взаємодії, коли авторизований користувач переглядає доступні аукціони. Запит користувача на перегляд аукціонів ініціює звернення до головної сторінки, яка у свою чергу взаємодіє з базою даних для отримання актуальної інформації про активні торги. Система формує список аукціонів з урахуванням їх статусу, часових рамок та релевантності, після чого повертає структуровані дані користувачу для відображення в інтерфейсі.

Функціональність пошуку та фільтрації реалізується через послідовність взаємодій. Коли користувач вводить пошуковий запит, система передає його до компонента головної сторінки, який обробляє критерії пошуку та формує відповідний SQL-запит до бази даних. Результати пошуку повертаються користувачу у вигляді релевантних аукціонів. Подальша фільтрація результатів ініціює додаткові запити з уточненими параметрами, що дозволяє користувачу поступово звужувати область пошуку до найбільш цікавих лотів.

Створення нового аукціону є багатоетапним процесом, що вимагає послідовної взаємодії між користувачем та системою. Ініціація створення аукціону користувачем призводить до активації відповідного компонента, який запитує необхідну інформацію про лот. Користувач поетапно вводить дані про предмет, встановлює параметри торгів, завантажує фотографії та додаткові матеріали. Кожен етап передбачає валідацію введених даних системою та надання зворотного зв'язку користувачу. Після завершення всіх необхідних кроків система підтверджує створення аукціону та робить його доступним для інших користувачів.

Процес участі в аукціоні демонструє найбільш інтерактивну частину платформи. Коли користувач обирає конкретний аукціон для детального перегляду, компонент аукціону надає повну інформацію про лот, включаючи поточну найвищу ставку, історію торгів та часові рамки. Рішення користувача зробити ставку ініціює взаємодію з компонентом ставок, який валідує суму відповідно до правил аукціону та поточного стану торгів.

Система ставок забезпечує критично важливу функціональність платформи через складну послідовність операцій. При отриманні нової ставки від користувача система перевіряє її валідність, порівнюючи з поточною найвищою ставкою та мінімальним кроком торгів. У випадку успішної валідації ставка фіксується в базі даних, оновлюється поточний стан аукціону та генерується підтвердження для користувача, який зробив ставку.

Система сповіщень активується автоматично при прийнятті нової ставки та забезпечує інформування всіх зацікавлених учасників торгів. Компонент ставок передає інформацію про нову ставку до системи сповіщень, яка визначає список користувачів, що потребують інформування (попередній лідер торгів, підписники аукціону, продавець), формує персоналізовані повідомлення та доставляє їх через різні канали комунікації.

Інтеграція WebSocket технології забезпечує реальночасове оновлення інформації для всіх активних учасників аукціону. Коли користувач переглядає детальну інформацію про аукціон, встановлюється постійне з'єднання через WebSocket, що дозволяє миттєво отримувати оновлення про нові ставки без необхідності оновлення сторінки. Це створює динамічне та захоплююче середовище для торгів.

Обробка виняткових ситуацій також моделюється через послідовність взаємодій. Якщо користувач намагається зробити некоректну ставку (нижчу за поточну або таку, що не відповідає мініальному кроку), система відхиляє запит та надає детальне пояснення причин відмови. Аналогічно обробляються ситуації з технічними проблемами, недостатньою авторизацією або спробами доступу до неіснуючих аукціонів.

Завершення аукціону запускає декілька автоматичних операцій. Система визначає переможця на основі найвищої ставки, оновлює статус аукціону, генерує сповіщення для продавця та переможця, ініціює процес фінансової транзакції та створює записи для подальшого відстеження виконання угоди.

Асинхронні процеси також враховуються при моделюванні послідовності. Багато операцій, таких як обробка зображень, відправка електронних листів або генерація звітів, виконуються асинхронно для забезпечення швидкої реакції інтерфейсу на дії користувача.

Моделювання послідовності взаємодії дозволяє забезпечити узгодженість поведінки системи, виявити потенційні вузькі місця в продуктивності та сформулювати чіткі вимоги до реалізації кожного компонента платформи. Такий підхід створює основу для розробки інтуїтивно зрозумілого та надійного користувацького досвіду на всіх етапах взаємодії з аукціонною платформою.

3.2 Розробка алгоритму проведення аукціону

Ефективний алгоритм проведення аукціону є критично важливим компонентом платформи, оскільки він безпосередньо впливає на прозорість торгів, задоволеність користувачів та загальну надійність системи. При проектуванні даного алгоритму необхідно врахувати різноманітні сценарії та граничні випадки, що можуть виникнути під час проведення аукціону, а також забезпечити оптимальну продуктивність системи при одночасній участі багатьох користувачів [14].

Ключовою особливістю розроблюваної платформи є використання технології WebSocket для забезпечення комунікації в реальному часі. Це дозволяє реалізувати динамічний процес торгів з миттєвим оновленням

інформації для всіх учасників, створюючи ефект присутності та підвищуючи залученість користувачів [6].

Ініціалізація аукціону відбувається, коли продавець створює новий аукціон через інтерфейс платформи. На цьому етапі встановлюються базові параметри, такі як опис лота, стартова ціна, мінімальний крок ставки, дата та час початку і завершення аукціону, а також опціональні параметри, наприклад, резервна ціна (мінімальна прийнятна ціна для продавця) або функціонал автоматичного продовження. Система валідує всі введені дані, генерує унікальний ідентифікатор аукціону та зберігає інформацію в базі даних. Аукціон отримує статус "очікує початку".

Активация аукціону відбувається автоматично при настанні запланованої дати та часу початку. Система змінює статус аукціону на "активний" та надсилає сповіщення зацікавленим користувачам (тим, хто додав аукціон до списку спостереження або підписаний на відповідну категорію). На цьому етапі також створюється окрема кімната WebSocket для даного аукціону, до якої приєднуються всі активні учасники.

Процес торгів є центральним елементом алгоритму. Коли користувач робить ставку, клієнтська частина надсилає запит через WebSocket з'єднання, включаючи ідентифікатор користувача, ідентифікатор аукціону та суму ставки. Серверна частина виконує ряд перевірок, а саме авторизацію користувача, перевірку статусу аукціону, перевірку суми ставки (вона має перевищувати поточну найвищу ставку щонайменше на розмір мінімального кроку), перевірка на конфлікти (чи не зроблена вже інша ставка в момент обробки поточної).

Якщо всі перевірки пройдені успішно, система створює новий запис про ставку в базі даних, оновлює поточну найвищу ставку аукціону та надсилає оновлення всім учасникам через WebSocket. Користувач, який зробив попередню найвищу ставку, отримує спеціальне сповіщення про перебиття його ставки.

Для забезпечення точності та справедливості обробки одночасних ставок реалізується черга обробки запитів на стороні сервера. Це дозволяє послідовно обробляти ставки, навіть якщо вони надійшли майже одночасно, та уникнути гонки даних (race condition). У випадку, коли надходять абсолютно одночасні ставки з однаковою сумою, пріоритет надається тій, що першою потрапила до черги (рис. 3.2).

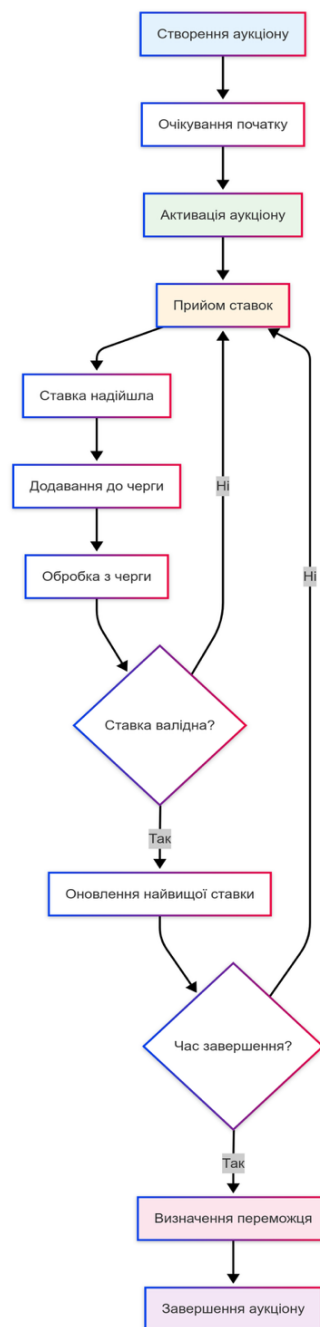


Рисунок 3.2 – Спрощена діаграма роботи алгоритму

Автоматичне продовження аукціону (опціональна функція) активується, якщо нова ставка зроблена в останні хвилини перед запланованим завершенням аукціону. Це запобігає тактиці "снайпінгу", коли учасники намагаються зробити ставку в останні секунди, не залишаючи іншим часу на реакцію. При активації цієї функції час завершення аукціону автоматично подовжується на певний період (наприклад, 5 хвилин).

Завершення аукціону відбувається автоматично при настанні запланованого часу закінчення (з урахуванням можливих автоматичних продовжень). Система змінює статус аукціону на "завершений" та визначає переможця – користувача з найвищою ставкою, якщо така ставка перевищує або дорівнює резервній ціні (якщо вона була встановлена). Якщо жодна ставка не перевищила резервну ціну, аукціон завершується без переможця.

Система надсилає сповіщення всім учасникам про завершення аукціону, окремо інформуючи переможця та продавця про результати та наступні кроки. Також оновлюється статус лота та створюються записи для відстеження подальшої транзакції.

Після визначення переможця ініціюється процес фіналізації, який включає обмін контактною інформацією між продавцем та покупцем (якщо це передбачено політикою платформи), створення платіжного доручення та інструкцій щодо доставки. Система може автоматично стягувати комісійні збори з суми продажу відповідно до умов використання платформи.

Закриття транзакції відбувається після підтвердження отримання оплати продавцем та отримання товару покупцем. На цьому етапі користувачі можуть залишити відгуки один про одного, що впливає на їхні рейтинги довіри. Система оновлює статус транзакції та завершує життєвий цикл аукціону.

Важливим аспектом алгоритму є обробка виняткових ситуацій, таких як технічні проблеми у користувачів, відмова переможця від покупки, непередбачене зняття лота продавцем тощо. Для кожного такого випадку розробляються спеціальні процедури, що дозволяють мінімізувати негативний вплив на користувачів та підтримувати цілісність системи.

Для забезпечення стійкості системи до високих навантажень реалізуються механізми балансування навантаження та кешування даних. Зокрема, інформація про поточний стан активних аукціонів може зберігатися в оперативній пам'яті для швидкого доступу, з періодичною синхронізацією з базою даних для забезпечення надійності.

Алгоритм проведення аукціону також включає механізми моніторингу та виявлення підозрілої активності, такої як штучне підняття ціни, змова між учасниками або використання ботів для автоматичних ставок. Це дозволяє підтримувати чесність та прозорість торгів, що є критично важливим для репутації платформи.

Таким чином, розроблений алгоритм проведення аукціону забезпечує повний життєвий цикл торгів від створення до завершення транзакції, враховуючи різноманітні сценарії та виняткові ситуації. Використання технології WebSocket для комунікації в реальному часі створює динамічне та захопливе середовище для учасників, підвищуючи їхнє залучення та задоволеність від використання платформи.

3.3 Об'єктно-орієнтоване проектування аукціонної платформи

Розробка платформи для організації та проведення онлайн-аукціонів базується на принципах об'єктно-орієнтованого програмування (ООП), що дозволяє створити модульну, масштабовану та підтримувану архітектуру системи. Використання парадигми ООП забезпечує ефективну організацію коду через інкапсуляцію, успадкування, поліморфізм та абстракцію, що є особливо важливим для складних бізнес-систем.

Архітектура платформи побудована на основі модульного підходу, де кожен модуль відповідає за конкретну функціональність та взаємодіє з іншими компонентами через чітко визначені інтерфейси. Такий підхід відповідає

принципу єдиної відповідальності (Single Responsibility Principle) та дозволяє незалежно розвивати окремі частини системи.

Основні сутності системи представлені класами-ентіті, що відображають бізнес-логіку аукціонної платформи (рис. 3.3). Клас `User` інкапсулює всю інформацію про користувача системи, включаючи особисті дані, автентифікаційну інформацію та зв'язки з іншими сутностями. Атрибути класу (`id`, `email`, `password`, `firstName`, `lastName`, `role`, `createdAt`, `updatedAt`) забезпечують повне представлення користувача в системі, а відносини з аукціонами, ставками та сповіщеннями створюють основу для складної бізнес-логіки платформи [11]. Фрагмент коду в якому реалізовано клас `User` наведено в додатку А.

Клас `Auction` є центральною сутністю системи, що моделює аукціон з усіма його характеристиками. Інкапсуляція даних про назву, опис, стартову та поточну ціни, часові рамки та статус дозволяє ефективно управляти життєвим циклом аукціону. Зв'язки з продавцем, переможцем, ставками та категорією реалізують складні бізнес-правила через асоціативні відносини між об'єктами.

Клас `Bid` представляє ставку в аукціоні та демонструє принцип композиції, оскільки існує лише в контексті конкретного аукціону та користувача. Мінімалістичний набір атрибутів (`id`, `amount`, `auctionId`, `userId`, `createdAt`) забезпечує ефективне зберігання та обробку великої кількості ставок без надлишкових даних.

Модульна структура системи реалізована через створення спеціалізованих модулів, кожен з яких відповідає за конкретну предметну область. `Auth Module` інкапсулює всю логіку автентифікації та авторизації, використовуючи класи `AuthService`, `AuthController` та `JwtStrategy`. Цей модуль демонструє застосування патерну `Strategy` для реалізації різних методів автентифікації, що дозволяє легко розширювати систему новими способами входу користувачів. Фрагмент коду в якому реалізовано клас `AuthService` наведено в додатку Г.

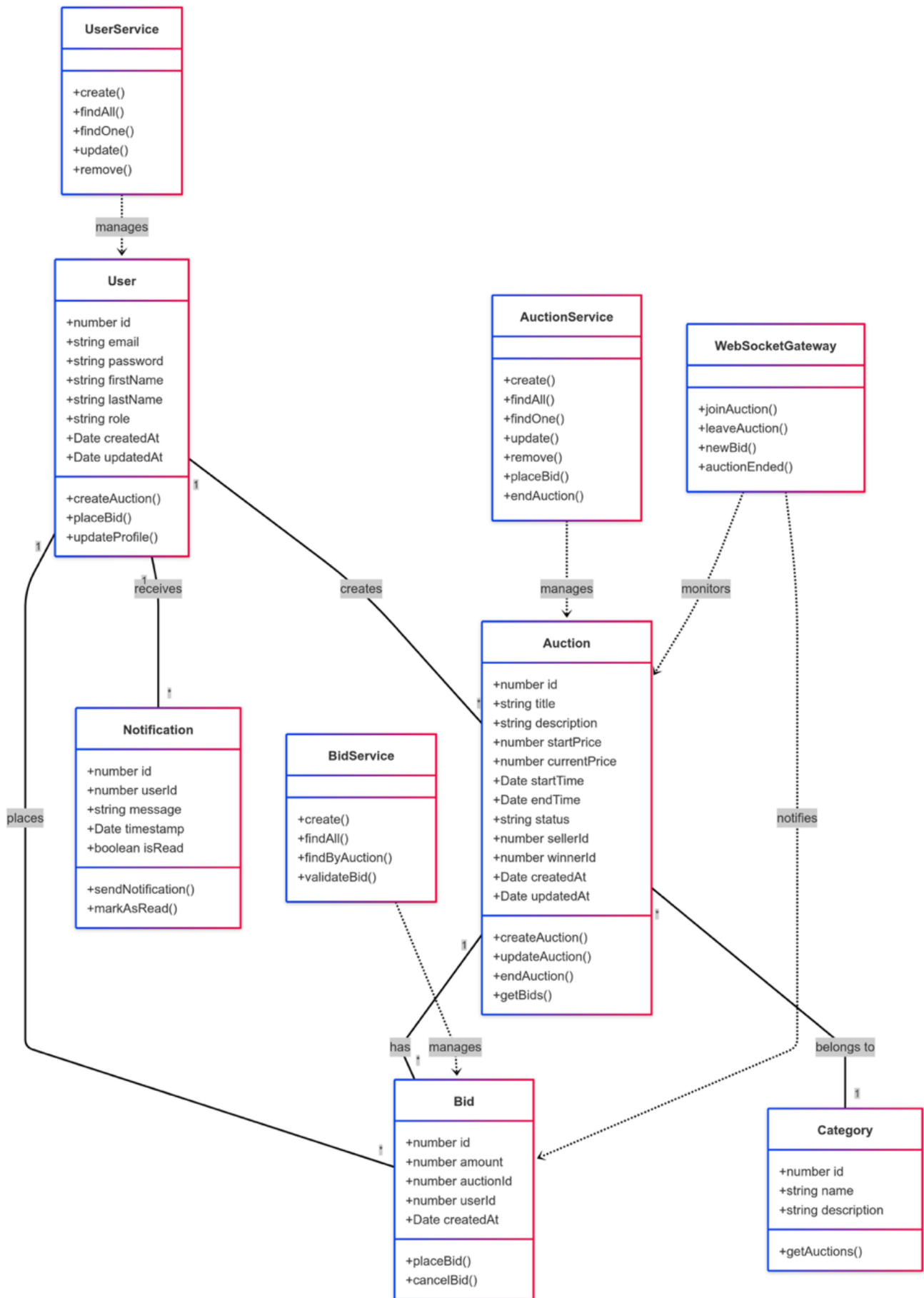


Рисунок 3.3 – Взаємозв'язок класів аукціон платформи

Auction Module є найскладнішим компонентом системи, що керує всіма аспектами функціонування аукціонів. Використання класів AuctionService, AuctionController та AuctionEntity демонструє розділення відповідальності між рівнями презентації, бізнес-логіки та доступу до даних. Такий підхід дозволяє незалежно тестувати та модифікувати кожен рівень без впливу на інші компоненти. Фрагмент коду в якому реалізовано клас AuctionService наведено в додатку Б.

Bid Module реалізує функціональність управління ставками через BidService, BidController та BidEntity. Особливістю цього модуля є тісна інтеграція з WebSocket функціональністю для забезпечення оновлень в реальному часі, що демонструє ефективне поєднання різних архітектурних патернів. Фрагмент коду в якому реалізовано клас BidService наведено в додатку В.

WebSocket Module забезпечує інтерактивність платформи через WebSocketGateway та WebSocketService. Цей модуль демонструє застосування патерну Observer, де зміни в стані аукціону автоматично транслюються всім зацікавленим клієнтам без необхідності активного опитування сервера.

Сервісний шар системи реалізує основну бізнес-логіку через спеціалізовані сервіси. AuctionService інкапсулює всі операції з аукціонами, від створення до завершення торгів. Методи create, findAll, findOne, update, remove, placeBid, endAuction забезпечують повний життєвий цикл аукціону та демонструють принцип інкапсуляції бізнес-правил в межах відповідного класу.

BidService спеціалізується на операціях зі ставками, включаючи їх створення, пошук та валідацію. Метод validateBid особливо важливий, оскільки забезпечує дотримання бізнес-правил аукціону та попереджає некоректні операції.

UserService керує всіма аспектами роботи з користувачами, від реєстрації до управління профілями. Інкапсуляція операцій користувача в окремому сервісі дозволяє централізовано управляти бізнес-логікою, пов'язаною з обліковими записами.

Контролери представляють рівень презентації та забезпечують взаємодію з зовнішніми клієнтами через RESTful API. AuctionController надає повний набір ендпоінтів для управління аукціонами, дотримуючись REST-принципів та забезпечуючи інтуїтивно зрозумілий інтерфейс для клієнтських додатків.

Функціональність WebSocket реалізована через AuctionGateway, що обробляє події joinAuction, leaveAuction, newBid, auctionEnded. Цей компонент демонструє застосування подійно-орієнтованої архітектури та забезпечує реактивну взаємодію користувачів з платформою.

Взаємодія між компонентами побудована на основі чітко визначених зв'язків. Відношення "один-до-багатьох" між User та Auction дозволяє користувачу створювати множину аукціонів, зберігаючи при цьому цілісність даних через зовнішні ключі. Аналогічно, зв'язки між Auction та Bid, а також User та Bid забезпечують правильне моделювання бізнес-процесів аукціонної платформи.

Система безпеки та валідації реалізована через Guards (JwtAuthGuard, RolesGuard), що демонструють застосування патерну Decorator для додавання функціональності безпеки до контролерів. Data Transfer Objects (DTOs) як CreateAuctionDto, UpdateAuctionDto, CreateBidDto забезпечують типізовану валідацію вхідних даних та інкапсулюють правила перевірки в межах відповідних класів.

Використання Validation Pipes демонструє застосування патерну Pipeline для послідовної обробки та валідації запитів, що забезпечує надійність та безпеку системи на рівні архітектури.

Така об'єктно-орієнтована архітектура забезпечує високу модульність, тестованість та розширюваність платформи, дозволяючи ефективно

реалізувати складну бізнес-логіку аукціонної системи з дотриманням сучасних принципів проектування програмного забезпечення.

3.4 Проектування моделі бази даних

Проектування моделі бази даних для платформи онлайн-аукціонів базується на реляційному підході та передбачає створення нормалізованої структури, що забезпечує цілісність даних, оптимальну продуктивність запитів та можливість ефективного масштабування системи і відображає об'єктно-орієнтовану сутність структур даних. Структура та взаємозв'язки між таблицями наведені в рисунку 3.4. Використання PostgreSQL у поєднанні з TypeORM надає потужні інструменти для реалізації складної моделі даних з різноманітними зв'язками між сутностями та забезпечує типізований доступ до даних на рівні додатку.

Центральною сутністю системи є таблиця `users`, що містить інформацію про всіх користувачів платформи. Структура таблиці включає первинний ключ `id` типу `integer`, унікальну електронну адресу `email` для автентифікації, хешований пароль `password`, персональні дані користувача (`first_name`, `last_name`), роль користувача в системі `role` та службові поля для відстеження часу створення та оновлення записів (`created_at`, `updated_at`). Поле `role` дозволяє диференціювати права доступу між звичайними користувачами, адміністраторами та експертами, створюючи гнучку систему авторизації [12]. Фрагмент коду в якому реалізовано таблицю `User` наведено в додатку А.

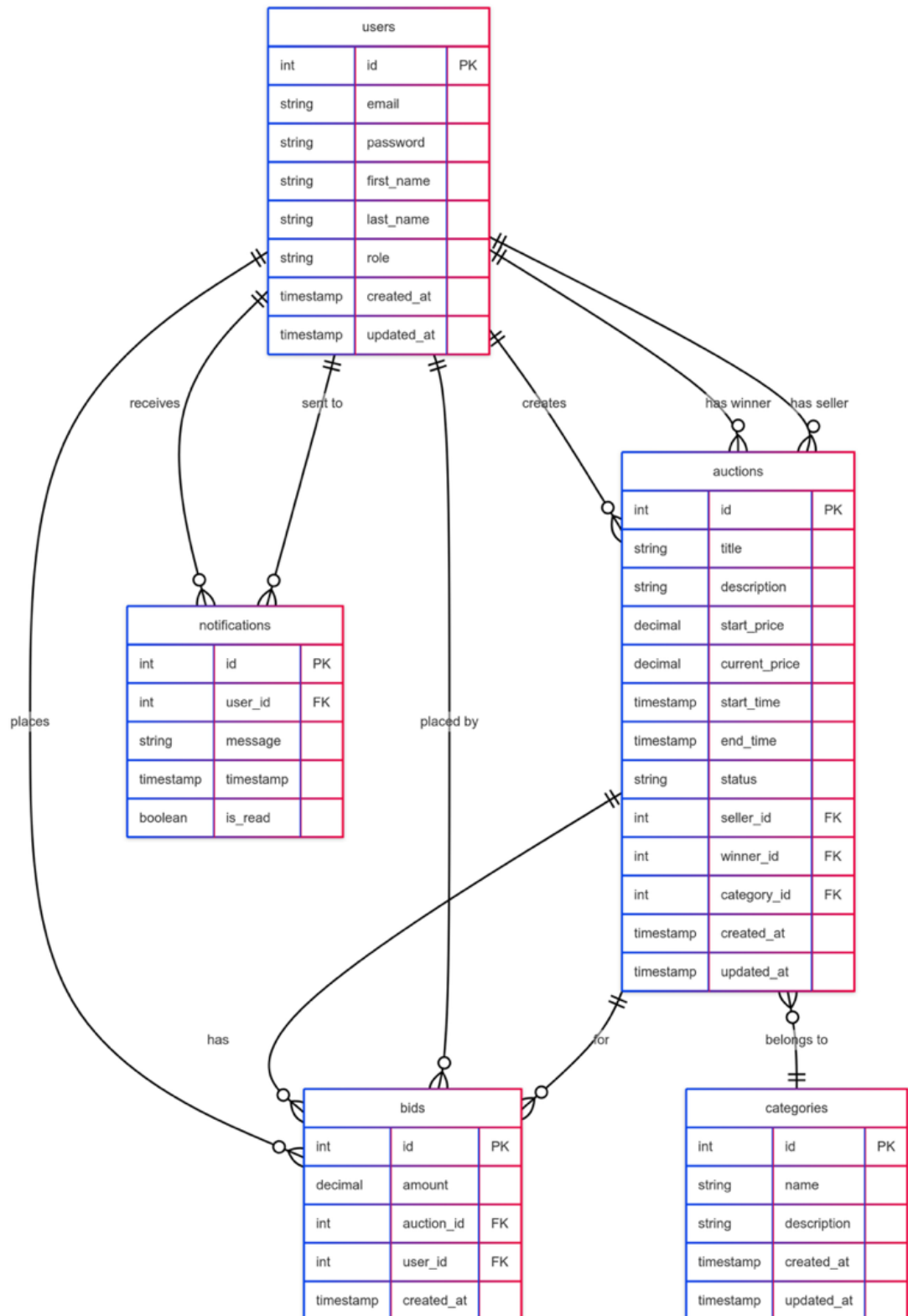


Рисунок 3.4 – Взаємозв'язки між таблицями в БД

Таблиця **auctions** представляє основну бізнес-сутність системи та містить всю необхідну інформацію для проведення аукціонів. Первинний ключ **id** забезпечує унікальну ідентифікацію кожного аукціону, поля **title** та

description містять описову інформацію для користувачів. Фінансові параметри представлені полями start_price та current_price типу decimal, що забезпечує точність розрахунків при роботі з грошовими сумами. Часові рамки аукціону визначаються полями start_time та end_time типу timestamp, що дозволяє автоматично управляти життєвим циклом торгів [13].

Особливу увагу приділено зовнішнім ключам у таблиці auctions. Поле seller_id посилається на таблицю users та ідентифікує користувача-продавця, який створив аукціон. Поле winner_id також посилається на users, але може бути NULL до завершення аукціону, після чого воно заповнюється ідентифікатором переможця торгів. Поле category_id створює зв'язок з таблицею categories, дозволяючи класифікувати аукціони за тематичними розділами. Поле status відображає поточний стан аукціону (очікує початку, активний, завершений, скасований).

Таблиця bids фіксує всі ставки, зроблені користувачами під час аукціонів. Мінімалістична структура включає первинний ключ id, суму ставки amount типу decimal, зовнішні ключі auction_id та user_id для зв'язку з відповідними аукціоном та користувачем, а також часову мітку created_at для фіксації моменту створення ставки. Ця таблиця є критично важливою для продуктивності системи, оскільки може швидко зростати за обсягом, тому потребує ретельного індексування.

Таблиця categories забезпечує систему класифікації аукціонів та включає первинний ключ id, назву категорії name, детальний опис description та службові поля часових міток. Ієрархічна структура категорій може бути реалізована через додавання поля parent_id, що дозволить створювати багаторівневу систему класифікації колекційних предметів.

Таблиця notifications реалізує систему сповіщень користувачів про важливі події на платформі. Структура включає первинний ключ id, зовнішній ключ user_id для ідентифікації отримувача, текст повідомлення message, часову мітку timestamp та булеве поле is_read для відстеження статусу прочитання сповіщення.

Зв'язки між таблицями реалізують складну бізнес-логіку аукціонної платформи. Відношення "один до багатьох" між users та auctions дозволяє одному користувачу створювати множину аукціонів, при цьому кожен аукціон має єдиного продавця. Аналогічне відношення існує між users та bids, оскільки користувач може робити багато ставок, але кожна ставка належить конкретному користувачу.

Зв'язок між auctions та bids також має тип "один до багатьох", оскільки в одному аукціоні може бути багато ставок, але кожна ставка стосується лише одного аукціону. Цей зв'язок є основою для реалізації механізму торгів та визначення переможця аукціону.

Особливий зв'язок існує між auctions та users через поле winner_id, що реалізує відношення "багато до одного" та дозволяє визначити переможця конкретного аукціону. Цей зв'язок може бути NULL до завершення торгів та заповнюється системою автоматично при визначенні переможця.

Зв'язок між auctions та categories типу "багато до одного" дозволяє кожному аукціону належати до конкретної категорії, при цьому одна категорія може містити багато аукціонів. Це забезпечує ефективну систему пошуку та фільтрації лотів за тематичними напрямками.

Індексування та оптимізація є критично важливими аспектами проектування бази даних. Окрім автоматично створюваних індексів для первинних ключів, необхідно створити додаткові індекси для полів, що часто використовуються у запитах. Зокрема, індекси для полів email у таблиці users, start_time та end_time у таблиці auctions, auction_id та user_id у таблиці bids значно покращать продуктивність системи.

Для забезпечення цілісності даних використовуються зовнішні ключі з відповідними обмеженнями CASCADE та RESTRICT. Наприклад, при видаленні користувача система може запобігти операції, якщо у нього є активні аукціони, або автоматично анулювати його ставки, залежно від бізнес-логіки.

Типи даних обрані з урахуванням специфіки аукціонної платформи. Використання `decimal` для грошових сум забезпечує точність фінансових розрахунків, `timestamp` для часових міток дозволяє точно відстежувати події у системі, а `string` типи оптимізовані за довжиною відповідно до очікуваного контенту.

Масштабованість бази даних забезпечується через правильне проектування індексів, партиціонування великих таблиць (особливо `bids`) за часовими інтервалами та можливість горизонтального масштабування через реплікацію `read-only` копій для операцій читання.

Така архітектура бази даних забезпечує надійне підґрунтя для функціонування аукціонної платформи, дозволяє ефективно реалізувати всі необхідні бізнес-процеси та має потенціал для подальшого розширення функціональності без кардинальної реструктуризації системи.

3.5 Проектування користувацького інтерфейсу

Проектування користувацького інтерфейсу для аукціонної платформи є критично важливим аспектом, що визначає успішність взаємодії користувачів з системою. Враховуючи різноманітність цільової аудиторії, яка включає органи місцевого самоврядування, аграрні підприємства та колекціонерів, необхідно створити універсальний інтерфейс, що задовольнятиме потреби всіх категорій користувачів при збереженні простоти та інтуїтивності використання (рис. 3.5).

The screenshot shows a web interface for creating a new auction. At the top, there is a blue navigation bar with the text 'Auction Site' on the left and 'AUCTIONS CREATE AUCTION PROFILE LOGOUT' on the right. Below this, the main content area features a form titled 'Create New Auction'. The form contains several input fields: 'Title *' with the text 'Продаж земельної ділянки приватної власності', 'Description *' with the text 'Продаж земельної ділянки приватної власності Для ведення особистого селянського господарства.Кадастровий номер:0722883400:02:000:1846. Площа 2,5414га.', 'Starting Price *' with the value '9000000', and 'End Date *' with the date '06/30/2025 12:00 AM'. Below these fields is a 'Choose File' button and a file name 'image.jpg'. At the bottom of the form, there is a preview image of a map or land plot.

Рисунок 3.5 – Мініmalістичний інтерфейс розроблювальної платформи

Загальні принципи проектування інтерфейсу базуються на сучасних стандартах UX/UI дизайну з урахуванням специфіки аукціонних процесів. Основою архітектури інтерфейсу є компонентний підхід, реалізований за допомогою бібліотеки React, що забезпечує високу модульність та можливість повторного використання компонентів для різних типів торгів.

Ключові принципи проектування включають модульність, консистентність та зворотний зв'язок. Модульність реалізується через незалежні компоненти інтерфейсу, адаптовані для різних типів аукціонів. Консистентність забезпечує єдину стилістику та поведінку елементів для всіх сегментів користувачів. Зворотний зв'язок надає миттєве інформування про результати дій, критично важливе для швидкості реакції в торгах.

Архітектура компонентів базується на ієрархічній структурі React-компонентів з чітко визначеною відповідальністю та можливістю повторного використання.

Система сповіщень є одним з ключових елементів користувацького інтерфейсу, що забезпечує миттєвий зворотний зв'язок з користувачем про важливі події в системі. Система реалізована за допомогою React Context API

та складається з трьох основних компонентів: ToastContext для управління станом сповіщень, ToastProvider для забезпечення глобального доступу до функцій сповіщень, та ToastNotification для відображення окремих сповіщень.

Контекстна архітектура забезпечує глобальний доступ до системи сповіщень з будь-якого компонента додатку без необхідності передачі пропсів через всю ієрархію компонентів. Автоматичне управління життєвим циклом сповіщень реалізується через механізм setTimeout, де сповіщення автоматично видаляються через заданий час (за замовчуванням 4 секунди), що зменшує когнітивне навантаження на користувача.

Система підтримує чотири типи сповіщень з унікальними візуальними стилями: Success (зелений колір) для підтвердження успішних операцій, Error (червоний колір) для повідомлення про помилки, Warning (жовтий колір) для важливих застережень та Info (синій колір) для загальної інформації.

Інтеграція з WebSocket забезпечує тісну взаємодію системи сповіщень з технологією реального часу для миттєвого інформування про оновлення аукціонів (рис. 3.6).

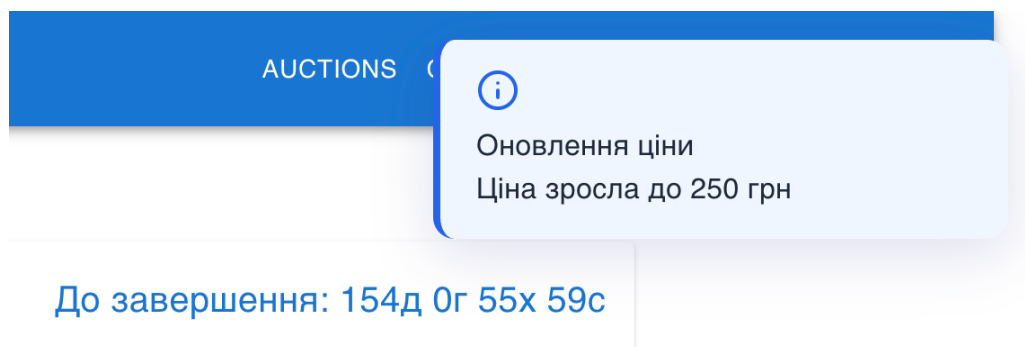


Рисунок 3.6 – Сповіщення про оновлення ціни в реальному часі

Для державних торгів система автоматично генерує сповіщення про зміни у документації, появу нових учасників та завершення періодів подачі заявок. Аграрні підприємства отримують персоналізовані сповіщення про появу нових земельних ділянок у цікавих регіонах, зміни умов оренди та результати торгів. Колекціонери мають доступ до розширеної системи

сповіщень про появу предметів у цікавих категоріях та активність улюблених продавців.

Таким чином, розроблена система користувацького інтерфейсу забезпечує сучасний, інтуїтивний та доступний досвід взаємодії з аукціонною платформою для всіх категорій користувачів. Використання компонентної архітектури React, системи контекстів та сучасних бібліотек для анімацій створює міцну основу для подальшого розвитку та масштабування додатку з особливою увагою до системи real-time сповіщень, критично важливої для ефективного функціонування аукціонної платформи. Повний код додатку можна переглянути за посиланням: <https://github.com/igordmitrow/auction>

3.6 Тестування додатку

Для забезпечення якості та надійності аукціонної платформи було впроваджено комплексну систему тестування з використанням фреймворку Vitest. Вибір Vitest обумовлений його високою швидкістю виконання, нативною підтримкою TypeScript та відмінною інтеграцією з екосистемою Vite, що забезпечує ефективне тестування як серверної, так і клієнтської частин додатку.

Тестування охоплює всі критично важливі компоненти системи, включаючи бізнес-логіку аукціонів, валідацію ставок, роботу Entity класів та компоненти користувацького інтерфейсу. Особливу увагу приділено тестуванню функціональності real-time оновлень через WebSocket, що є критично важливою для аукціонних процесів. Фрагмент коду такого тесту можна побачити в додатку Г.

Модульне тестування покриває всі сервіси та контролери серверної частини, включаючи перевірку коректності обробки ставок, визначення переможців аукціонів та валідацію даних.

Інтеграційне тестування перевіряє взаємодію між різними шарами системи, включаючи контролери, сервіси та базу даних. Особлива увага приділяється тестуванню сценаріїв, що моделюють реальні умови використання платформи різними категоріями користувачів - органами місцевого самоврядування, аграрними підприємствами та колекціонерами.

Покриття тестами включає всі критично важливі функції платформи з фокусом на бізнес-логіку аукціонів, безпеку транзакцій та коректність відображення даних в реальному часі. Автоматизоване генерування звітів про покриття дозволяє контролювати якість тестування та виявляти неохоплені ділянки коду, що забезпечує високу надійність системи при роботі з фінансовими операціями та державними торгами.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ

4.1 Аналіз стану виробничої санітарії і гігієни праці

При розробці платформи для організації та проведення онлайн-аукціонів важливо провести детальний аналіз потенційних травмонебезпечних ситуацій, які можуть виникнути в процесі виконання робіт. Хоча розробка програмного забезпечення зазвичай не асоціюється з високим ризиком фізичних травм, проте специфіка тривалої роботи за комп'ютером та особливості робочого середовища створюють певні ризики для здоров'я працівників.

Тривале статичне навантаження на опорно-руховий апарат, що може призвести до розвитку захворювань кістково-м'язової системи. Особливо вразливими є шийний та поперековий відділи хребта, а також суглоби кистей рук. Неправильна ергономіка робочого місця, недостатня кількість перерв та відсутність фізичної активності значно підвищують ризик виникнення таких проблем як синдром зап'ястного каналу, тендиніт, остеохондроз та інші патології.

Зорове перенапруження є типовим ризиком для програмістів та інших фахівців, які працюють з екранними пристроями. Тривале зосередження на моніторі, особливо при невідповідному освітленні, може призвести до розвитку астенопії (зорової втоми), симптомами якої є печіння в очах, відчуття піску під повіками, головний біль та тимчасове погіршення зору. У довгостроковій перспективі це може сприяти розвитку більш серйозних офтальмологічних проблем.

Ризики, пов'язані з електробезпекою, становлять важливу категорію потенційних небезпек. Використання несправного обладнання, пошкоджені кабелі, перевантаження електричних мереж можуть призвести до ураження електричним струмом та електротравм різного ступеня тяжкості. Особливу

увагу слід звернути на стан електропроводки та заземлення електрообладнання.

Психоемоційне перенапруження, хоча і не є прямою причиною фізичних травм, проте може значно підвищувати ризик їх виникнення через зниження концентрації уваги та погіршення загального функціонального стану організму. Напружений графік роботи, необхідність дотримання жорстких дедлайнів, складність вирішуваних завдань – все це створює значне психологічне навантаження.

Ризики падіння та травмування через недостатньо оптимальну організацію робочого простору. Загромадження проходів, нестійкі меблі, слизька підлога, непередбачене розташування обладнання можуть призвести до спотикань, падінь та отримання травм різного ступеня тяжкості.

Мікрокліматичні фактори також можуть створювати травмонебезпечні ситуації. Надмірно низька вологість повітря може призводити до пересихання слизових оболонок та збільшення електростатичного заряду на поверхнях, що підвищує ризик ураження статичною електрикою [18].

Для мінімізації зазначених ризиків необхідно впроваджувати комплексну систему заходів з охорони праці, яка включає організацію ергономічних робочих місць, регулярні перерви в роботі, проведення інструктажів з безпеки, контроль за станом електрообладнання та мікрокліматичними параметрами робочого середовища. Особливу увагу слід приділяти профілактиці професійних захворювань шляхом впровадження виробничої гімнастики та створення умов для підтримки фізичної активності працівників.

Важливим аспектом попередження травмонебезпечних ситуацій є також систематичне навчання працівників правилам безпечної роботи та формування культури безпеки праці в колективі. Усвідомлення ризиків та відповідальне ставлення до їх попередження з боку кожного працівника значно підвищує загальний рівень безпеки робочого середовища.

4.2 Обґрунтування організаційно-технічних рекомендацій з охорони праці

На основі аналізу травмонебезпечних ситуацій при розробці платформи онлайн-аукціонів пропонується комплекс рекомендацій для забезпечення безпечних умов праці. Динамічна організація робочого процесу з регулярними перервами кожні 60 хвилин та виконанням спеціалізованих вправ знизить фізичне навантаження на організм програмістів.

Ергономічне обладнання робочих місць (столи з регульованою висотою, крісла з підтримкою поперекового відділу) дозволить змінювати положення тіла та знизить ризик розвитку професійних захворювань.

Система освітлення з автоматичним регулюванням яскравості, монітори з антибліковим покриттям та правильне їх розташування зменшать зорове напруження.

Комплексна перевірка електрообладнання, надійне заземлення та якісні мережеві фільтри мінімізують ризики, пов'язані з електробезпекою. Оптимальні мікрокліматичні параметри забезпечуватимуться системами кондиціонування та зволожувачами повітря.

Регулярні тренінги з ергономіки, наочні матеріали та система управління стресом (техніки релаксації, зони відпочинку, гнучкий графік) сприятимуть психоемоційній стабільності працівників.

Економічна ефективність цих заходів обґрунтовується підвищенням продуктивності, зниженням кількості помилок та зменшенням витрат на лікування, що позитивно вплине на якість і швидкість реалізації проекту.

4.3 Пожежна безпека

Розробка платформи онлайн-аукціонів передбачає необхідність забезпечення безпеки персоналу під час виникнення надзвичайних ситуацій.

Основними потенційними загрозами для офісних приміщень, де здійснюється розробка, є пожежі, відключення електроенергії, природні катаклізми та терористичні загрози.

Для забезпечення безпеки необхідно розробити план евакуації з чітким позначенням основних та альтернативних шляхів виходу, який розміщується на видному місці кожного поверху будівлі. Всі співробітники повинні пройти інструктаж щодо дій у надзвичайних ситуаціях та брати участь у регулярних тренувальних евакуаціях не рідше двох разів на рік [19]. В офісі необхідно обладнати систему оповіщення, що активується автоматично при спрацюванні датчиків диму чи тепла. Враховуючи специфіку роботи з електронним обладнанням, приміщення повинні бути оснащені вуглекислотними вогнегасниками, які ефективні для гасіння пожеж класу Е та не пошкоджують електронну техніку.

В цілях захисту даних, що розробляються, необхідно налагодити систему регулярного резервного копіювання з дублюванням на віддалених серверах. Для мінімізації наслідків можливих техногенних аварій необхідно маркувати та захищати всі комунікації, інженерні мережі та обладнання згідно з нормативними вимогами [15].

Ефективність запропонованих заходів забезпечується шляхом призначення відповідальних осіб за кожен аспект безпеки та проведення регулярних перевірок систем безпеки. Економічна доцільність впровадження цих заходів обґрунтовується зменшенням потенційних збитків від втрати обладнання, даних та простою у роботі, а також уникненням можливих штрафів за порушення норм безпеки.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи нами було розроблено інноваційну веб-платформу для організації та проведення онлайн-аукціонів, що відповідає актуальним тенденціям розвитку електронної комерції та задовольняє потреби сучасних користувачів.

1. Аналіз предметної області, зокрема альтернативних рішень, виявив низку недоліків існуючих систем, що стало основою для обґрунтування доцільності розробки нової платформи.

2. В процесі постановки задачі було визначено цільову аудиторію та сценаріїв використання платформи, що дало змогу встановити ключові сегменти потенційних користувачів: органи місцевого самоврядування, аграрні підприємства, професійні продавці, колекціонери та випадкові покупці. Кожна з перелічених груп має свої специфічні потреби та сценарії взаємодії з системою.

3. Застосування обраного технологічного стеку дало змогу забезпечити високий рівень продуктивності та підтримованості коду.

4. Розроблена з урахуванням принципів об'єктно-орієнтованого програмування структурно-функціональна модель процесу проведення аукціонів дозволяє виділити основні функціональні блоки системи та зв'язки між ними. Особлива увага приділена розробці алгоритму проведення аукціону, який забезпечує прозорість торгів, обробку одночасних ставок та виняткових ситуацій, що підвищує загальний рівень безпеки при проведенні аукціонів. Також було здійснено моделювання варіантів використання платформи, що дозволило структурувати взаємодію різних типів користувачів з системою.

5. Запропоновані в роботі заходи з охорони праці уможливають проведення аукціонів з урахуванням чинних вимог до безпеки праці.

6. Спроектована повнофункціональна платформа для організації та проведення онлайн-аукціонів, що відрізняється інтуїтивно зрозумілим

інтерфейсом, надійною системою безпеки транзакцій та розвиненими соціальними функціями.

7. Практична цінність розробленої платформи полягає у створенні ефективного інструменту для проведення онлайн-аукціонів різного спрямування, що може бути використаний для комерційних аукціонів, благодійних торгів, реалізації конфіскованого майна тощо.

8. Перспективами подальшого розвитку проекту є розширення функціоналу для роботи з цифровими активами, включаючи невзаємозамінні токени (NFT), інтеграція з додатковими платіжними системами, впровадження елементів штучного інтелекту для персоналізації рекомендацій та автоматизації оцінки лотів, а також розробка мобільних додатків для розширення доступності платформи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Пінчук Н. С. Інформаційні системи і технології в маркетингу: навч.метод. посібник для самостійного вивчення дисципліни / Н. С. Пінчук, Г. П. Галузинський, Н. С. Орленко // Міністерство освіти і науки України, КНЕУ. – Київ: КНЕУ, 2010. – 352 с.
2. Аутентифікація і авторизація: що це і в чому відмінність. – QualityAssuranceGroup. – URL: <https://qagroup.com.ua/publications/autentyfikacii-i-avtoryzatsii/> (дата звернення: 07.12.2024).
3. Денисенко А. Фреймворки у веб-розробці – що це, які існують і для чого потрібні. – Highload. – URL: <https://highload.today/uk/frejmworki-u-veb-rozrobtsi-shho-tse-yaki-isnuyut-i-dlya-chogo-potribni/> (дата звернення: 10.12.2024).
4. Іванов В. Г. Сучасні інформаційні системи і технології: конспект лекцій / В. Г. Іванов, С. М. Іванов, В. В. Карасюк та ін.; за заг. ред. В. Г. Іванова, В. В. Карасюка. – Х.: Нац. юрид. ун-т ім. Ярослава Мудрого, 2014. – 347 с.
5. Архітектура та проектування програмного забезпечення : лекції. – Державний університет "Житомирська політехніка" - Освітній портал. – URL: <https://learn.ztu.edu.ua/mod/book/tool/print/index.php?id=278> (дата звернення: 14.12.2024).
6. Херрон Д. Node.js. Розробка серверних веб-додатків в JavaScript : / Пер. з англ. Слінкіна А.А: ДМК Пресс, 2012. – 144 с.
7. Демиденко М. А. Введення в сучасні бази даних : навч. посібник / М.А. Демиденко // НТУ «Дніпровська політехніка». – Д. : 2020. – 38 с.
8. Каштан В. Ю. Конспект лекцій з дисципліни “Бази даних в інформаційних системах” / В. Ю. Каштан, Д. В. Іванов. – Дніпро.: НТУ «ДП», 2020. – 58 с.
9. Як використовувати JSON Web Tokens (JWT) для автентифікації. – Codeguida.com, 2019. – URL: <https://codeguida.com/post/1567>

10. Introduction to JSON Web Tokens. – JWT.io. – URL: <https://jwt.io/introduction> (дата звернення: 15.03.2025).
11. Фаулер М. UML. Основы. 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с
12. Основні відомості про бази даних. – Microsoft.com. – URL: <https://support.microsoft.com/ukua/office/основні-відомості-про-бази-даних-a849ac16-07c7-4a31-9948-3c8c94a7c204> (дата звернення: 21.03.2025).
13. Сидоренко В. В. Організація баз даних: навч. посібник / В. В. Сидоренко, Л. В. Константинова, С. А. Смірнов // Видавець. Лисенко В. Ф, Кропивницький: ЦНТУ, 2018. – 274 с.
14. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» / Укл. Л. М. Дегтярьова. – Полтава: ПолтНТУ, 2017. – 218 с.
15. Табунщик Г. В. Проектування інформаційної інфраструктури медичних та телемедичних систем / Г. В. Табунщик, Т. І. Каплієнко, О. А. Петрова, О. В. Шитікова // Вид. ПП "Євро-Волинь", Житомир, 2021. – 198 с.
16. Як працює система? – Prozorro.продажі – URL: <https://prozorro.sale/platforms/> (дата звернення: 05.05.2025).
17. Лагодна Д. Зміни у функціоналі оголошення закупівель на майданчику Zakupivli.Pro – Радник, 2025. – URL: https://radnuk.com.ua/praktyka_zakupivel/nadporogi/zminy-u-funktsionalioholoshennia-zakupivel-na-majdanchyku-zakupivli-pro/ (дата звернення: 06.05.2025).
18. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень. URL: <https://zakon.rada.gov.ua/rada/show/va042282-99#Text> (дата звернення : 22.04.2025).
19. Правила пожежної безпеки в Україні (ред. 22.01.2022 р.). URL: <https://zakon.rada.gov.ua/laws/show/z0252-15#Text> (дата звернення: 22.04.2025).

ДОДАТКИ

Entity клас сутності User

```
@Entity('users')
export class User {
  @PrimaryGeneratedColumn('uuid')
  id: string;

  @Column({ unique: true })
  username: string;

  @Column({ unique: true })
  email: string;

  @Column()
  password: string;

  @Column({ default: false })
  role: string;

  @Column({ default: false })
  firstName: string;

  @Column({ default: false })
  lastName: string;

  @CreateDateColumn({ type: 'timestampz', name: 'created_at' })
  createdAt: Date;

  @UpdateDateColumn({ type: 'timestampz', name: 'updated_at' })
  updatedAt: Date;
}
```


Сервіс для роботи з аукціонами

```
@Injectable()
export class AuctionService {
  constructor(
    @InjectRepository(Auction)
    private auctionRepository: Repository<Auction>,
    private websocketGateway: WebsocketGateway,
  ) {}

  async create(createAuctionDto: CreateAuctionDto): Promise<Auction> {
    console.log(createAuctionDto);
    const auction = this.auctionRepository.create({
      ...createAuctionDto,
      currentPrice: createAuctionDto.startPrice,
      startingPrice: createAuctionDto.startPrice,
      startDate: new Date(),
    });

    const savedAuction = await this.auctionRepository.save(auction);

    // Notify all connected clients about the new auction
    this.websocketGateway.notifyNewAuction(savedAuction);

    return savedAuction;
  }

  async findAll(): Promise<Auction[]> {
    return await this.auctionRepository.find({
      where: { isEnded: false },
      relations: ['bids'],
      order: { endDate: 'ASC' },
    });
  }

  async findOne(id: string): Promise<Auction> {
    const auction = await this.auctionRepository.findOne({
      where: { id },
      relations: ['bids'],
    });

    if (!auction) {
      throw new NotFoundException(`Auction with ID ${id} not found`);
    }

    return auction;
  }

  async update(
    id: string,
```

```

        updateAuctionDto: UpdateAuctionDto,
    ): Promise<Auction> {
        const auction = await this.findOne(id);

        this.auctionRepository.merge(auction, updateAuctionDto);
        const updatedAuction = await this.auctionRepository.save(auction);

        // Notify all connected clients about the auction update
        this.websocketGateway.notifyAuctionUpdate(updatedAuction);

        return updatedAuction;
    }

    async remove(id: string): Promise<void> {
        const auction = await this.findOne(id);
        await this.auctionRepository.remove(auction);

        // Notify all connected clients about the auction removal
        this.websocketGateway.notifyAuctionRemoval(id);
    }

    async endAuction(id: string): Promise<Auction> {
        const auction = await this.findOne(id);

        if (auction.isEnded) {
            throw new Error('Auction is already ended');
        }

        auction.isEnded = true;

        // If there are bids, determine the winner
        if (auction.bids && auction.bids.length > 0) {
            const highestBid = auction.bids.sort(
                (a, b) => parseFloat(String(b.amount)) - parseFloat(String(a.amount)),
            )[0];

            auction.winnerId = highestBid.userId;
        }

        const endedAuction = await this.auctionRepository.save(auction);

        // Notify all connected clients about the auction ending
        this.websocketGateway.notifyAuctionEnded(endedAuction);

        return endedAuction;
    }
}

```

Сервіс для роботи з ставками

```
@Injectable()
export class BidService {
  constructor(
    @InjectRepository(Bid)
    private bidRepository: Repository<Bid>,
    private auctionService: AuctionService,
    private websocketGateway: WebsocketGateway,
  ) {}

  async create(createBidDto: CreateBidDto): Promise<Bid> {
    const auction = await this.auctionService.findOne(createBidDto.auctionId);

    // Check if auction is still active
    if (auction.isEnded) {
      throw new BadRequestException('Cannot place bid on an ended auction');
    }

    // Check if auction has started
    if (new Date() < auction.startDate) {
      throw new BadRequestException('Auction has not started yet');
    }

    // Check if bid amount is higher than current price
    if (Number(createBidDto.amount) <= Number(auction.currentPrice)) {
      throw new BadRequestException(
        'Bid amount must be higher than current price',
      );
    }

    // Create the bid
    const bid = this.bidRepository.create({
      ...createBidDto,
      auctionId: auction.id,
    });

    const savedBid = await this.bidRepository.save(bid);

    // Update the auction's current price
    await this.auctionService.update(auction.id, {
      currentPrice: createBidDto.amount,
    });

    // Notify all connected clients about the new bid
    this.websocketGateway.notifyNewBid(savedBid);

    return savedBid;
  }
}
```

```
async findByAuction(auctionId: string): Promise<Bid[]> {  
  // Get the auction to make sure it exists  
  await this.auctionService.findOne(auctionId);  
  
  return await this.bidRepository.find({  
    where: { auctionId },  
    order: { amount: 'DESC' },  
  });  
}  
  
async findOne(id: string): Promise<Bid> {  
  const bid = await this.bidRepository.findOne({  
    where: { id },  
    relations: ['auction'],  
  });  
  
  if (!bid) {  
    throw new NotFoundException(`Bid with ID ${id} not found`);  
  }  
  
  return bid;  
}
```

Сервіс для авторизації користувача

```
@Injectable()
export class AuthService {
  constructor(
    private jwtService: JwtService,
    private userService: UserService,
  ) {}

  async login(loginDto: LoginDto) {
    const user = await this.validateUser(loginDto);
    const payload = { username: user.username, sub: user.id };

    return {
      access_token: this.jwtService.sign(payload),
      user: {
        id: user.id,
        username: user.username,
        email: user.email,
      },
    };
  }

  async register(registerDto: RegisterDto) {
    // Check if username or email already exists
    const existingUser = await this.userService.findByUsernameOrEmail(
      registerDto.username,
      registerDto.email,
    );

    if (existingUser) {
      throw new UnauthorizedException('Username or email already exists');
    }

    // Hash the password
    const hashedPassword = await bcrypt.hash(registerDto.password, 10);

    // Create the user
    const newUser = await this.userService.create({
      username: registerDto.username,
      email: registerDto.email,
      password: hashedPassword,
    });

    // Generate JWT token
    const payload = { username: newUser.username, sub: newUser.id };

    return {
      access_token: this.jwtService.sign(payload),
      user: {
```

```
        id: newUser.id,  
        username: newUser.username,  
        email: newUser.email,  
      },  
    };  
  }  
  
  async validateUser(loginDto: LoginDto): Promise<any> {  
    const user = await this.userService.findByEmail(loginDto.email);  
  
    if (!user) {  
      throw new UnauthorizedException('Invalid credentials');  
    }  
  
    const isPasswordValid = await bcrypt.compare(  
      loginDto.password,  
      user.password,  
    );  
  
    if (!isPasswordValid) {  
      throw new UnauthorizedException('Invalid credentials');  
    }  
  
    return user;  
  }  
}
```

Unit тест для сервісу аукціонів (AuctionService)

```
describe('AuctionService', () => {
  let auctionService: AuctionService;

  beforeEach(() => {
    auctionService = new AuctionService();
  });

  it('should validate bid amount correctly', () => {
    const auction = { currentPrice: 100, minBidStep: 10 };
    const validBid = auctionService.validateBid(auction, 115);
    const invalidBid = auctionService.validateBid(auction, 105);

    expect(validBid).toBe(true);
    expect(invalidBid).toBe(false);
  });

  it('should determine auction winner', () => {
    const bids = [
      { userId: 'user1', amount: 100 },
      { userId: 'user2', amount: 150 },
      { userId: 'user3', amount: 120 }
    ];

    const winner = auctionService.determineWinner(bids);
    expect(winner.userId).toBe('user2');
    expect(winner.amount).toBe(150);
  });
});
```