

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Вебдодаток аукціону речей»

Виконав студент групи ПЗ-20-3
ГРУНИЦЬКИЙ Дмитро Сергійович

Керівник роботи:
ГОРДІЄНКО Ірина Валеріївна

Консультант:
ЧИЖМОТРЯ Олена Геннадіївна

Рецензент:
ВАКАЛЮК Тетяна Анатоліївна

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Завідувач кафедри інженерії
програмного забезпечення

Тетяна ВАКАЛЮК

«26» лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ГРУНИЦЬКИЙ Дмитро Сергійович**

Керівник роботи: **ГОРДІЄНКО Ірина Валеріївна**

Тема роботи: **«Вебдодаток аукціону речей»,**

затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р., №74/с**

Вихідні дані для роботи: **розробка вебдодатку аукціону речей, який охоплює в собі всі аспекти онлайн-торгівлі різноманітними предметами через аукціонну платформу.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Гордієнко І. В., Чижмотря О. Г.	27.02.2024р.	27.02.2024р.
2	Гордієнко І. В., Чижмотря О. Г.	27.02.2024р.	27.02.2024р.
3	Гордієнко І. В., Чижмотря О. Г.	27.02.2024р.	27.02.2024р.

РЕФЕРАТ

Пояснювальна записка з кваліфікаційної роботи на тему «Вебдодаток аукціону речей» містить текстову частину на 76 сторінках друкованого тексту, 8 сторінок додатків, 45 рисунків та 15 таблиць. Список використаних джерел включає 30 найменувань і займає 3 сторінки. Загальний обсяг роботи становить 84 сторінок.

Основною метою є розробка вебдодатку аукціону речей, що спрямований на вдосконалення процесів торгів, забезпечення зручності та ефективності для покупців і продавців, а також усунення недоліків інших аукціонних платформ.

Було визначено основні етапи і завдання розробки, включаючи визначення вимог користувачів, створення прототипу системи, аналіз аналогічних програмних рішень, вибір засобів реалізації, розробку структури бази даних та алгоритмів функціонування системи. Обрано відповідну архітектуру та розроблено модульну структуру системи з об'єктно-орієнтованими компонентами. Особлива увага була приділена детальному опису програмної реалізації сайту, включаючи розробку інтерфейсу користувача, інтеграцію функціональних модулів та оптимізацію продуктивності. Описано процеси створення та редагування лотів, проведення торгів, управління ставками, а також методи, які забезпечують зручність використання та інтуїтивність інтерфейсу для кінцевих користувачів.

КЛЮЧОВІ СЛОВА: АУКЦІОН, ВЕБДОДАТОК, ТОРГИ, КОРИСТУВАЧ, СИСТЕМА.

					ІПЗ.КР.Б – 121 – 24 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Д. С. Груницький				Вебдодаток аукціону речей	Літ.	Арк.	Аркушів
Керівник	І. В. Гордієнко						3	84
						Житомирська політехніка, група ІПЗ-20-3		
Рецензент	Т. А. Вакалюк							
Зав. каф.	Т. А. Вакалюк							

ABSTRACT

The explanatory note on the qualification work on the topic «Web application of the auction of things» contains a text part of 76 pages of printed text, 8 pages of applications, 45 figures and 15 tables. The list of references includes 30 titles and occupies 3 pages. The total volume of the work is 84 pages.

The main objective is to develop a web-based auction application for items aimed at improving the bidding process, ensuring convenience and efficiency for buyers and sellers, and eliminating the shortcomings of other auction platforms.

The main stages and tasks of the development were identified, including defining user requirements, creating a system prototype, analysing similar software solutions, selecting implementation tools, developing the database structure and algorithms for the system. An appropriate architecture was selected and a modular system structure with object-oriented components was developed. Particular attention was paid to a detailed description of the software implementation of the website, including the development of the user interface, integration of functional modules and performance optimisation. The article describes the processes of creating and editing lots, bidding, managing bids, as well as methods that ensure ease of use and intuitiveness of the interface for end users.

KEYWORDS: AUCTION, WEB APPLICATION, BIDDING, USER, SYSTEM.

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБДОДАТКУ АУКЦІОНУ РЕЧЕЙ	9
1.1. Постановка задачі	9
1.2. Аналіз аналогів програмного продукту	10
1.3. Вибір архітектури веборієнтованої системи аукціону речей	16
1.4. Обґрунтування вибору інструментальних засобів та вимоги апаратного забезпечення	20
Висновки до першого розділу	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБОРІЄНТОВАНОЇ СИСТЕМИ АУКЦІОНУ РЕЧЕЙ	24
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи	24
2.2. Розробка бази даних системи	34
2.3. Проєктування та реалізація алгоритмів роботи системи	39
2.4. Реалізація додатку веборієнтованої системи управління аукціону речей	44
Висновки до другого розділу	49
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБДОДАТКОМ АУКЦІОНУ РЕЧЕЙ	50
3.1. Порядок встановлення та налаштування параметрів системи	50
3.2. Структура інтерфейсу вебдодатку аукціону речей	52
3.3. Тестування роботи програмного комплексу	70
Висновки до третього розділу	72
ВИСНОВКИ	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	74
ДОДАТКИ	77

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

API - Application Programming Interface

СУБД – Система управління базами даних

ПЗ – Програмне забезпечення

ПК – Персональний комп'ютер

ОС – Операційна система

UI/UX – User Interface/User Experience, дизайн і враження від користування вебдодатком

CRUD – Create, Read, Update, Delete, базові операції з даними

JSON – JavaScript Object Notation, текстовий формат обміну даними

JWT – JSON Web Token, механізм автентифікації

ORM – Object-Relational Mapping, техніка перетворення даних між системами об'єктної моделі та реляційною базою даних

HTTP – HyperText Transfer Protocol

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Сучасні аукціони з речами – це жвавий та перспективний розділ торгівлі, де можна знайти різноманітні предмети для покупки. Як свідчать дані, аукціони речей стають все популярнішими, привертаючи увагу широкого кола покупців. У цьому контексті особливо актуальною стає тема аукціонів, спрямованих на продажі цінних або звичайних предметів.

В умовах сучасного суспільства, технологічний прогрес стає невід'ємною частиною нашого життя, аукціони речей переходять в онлайн-простір, забезпечуючи зручність та доступність для широкого кола учасників. Проте, не дивлячись на позитивні тенденції, ринок аукціонів речей має свої переваги та недоліки.

Метою роботи є розробка аукціону речей, спрямованого на аспекти аукціонів речей, які становлять виклики та можливості для подальшого розвитку цього унікального ринку. Розглядаючи потреби та очікування покупців і продавців, планується створити систему, яка зробить участь в аукціонах речей простішою та ефективнішою.

Завданнями дослідження для досягнення поставленої мети є наступні:

- Проведення аналізу існуючих систем, що використовуються для проведення аукціонів з продажу речей. Це включає дослідження їх функціональності, ефективності та інші ключові аспекти.
- Визначення основних вимог до системи. Це означає ідентифікацію потреб користувачів, включаючи як покупців, так і продавців, а також урахування технічних обмежень.
- Обрання оптимальної архітектури та розробка функціональних можливостей системи. Це включає вибір структури системи, а також розробку основних функцій, які задовольняють визначені вимоги.
- Вибір стеку технологій для розробки. Це включає вибір потрібних мов програмування, баз даних, фреймворків та інших інструментів, які найкраще відповідають потребам проєкту.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Об'єктом дослідження є вебдодаток аукціону речей, який охоплює всі аспекти розробки, впровадження та функціонування цієї платформи в інтернеті.

Предметом дослідження є розробка вебдодатку аукціону речей з використанням сучасних веборієнтованих технологій, спрямованих на створення зручного та ефективного інтерфейсу для учасників аукціонів. Дослідження охоплює всі аспекти створення та впровадження вебдодатку, включаючи розробку функціональності для додавання, перегляду та купівлі товарів, а також забезпечення безпеки та надійності платформи. Предмет дослідження також включає вивчення користувацького досвіду та взаємодії з платформою з метою оптимізації процесу участі у аукціонах.

Розроблений вебдодаток аукціону речей може стати корисним інструментом для покупців та продавців, які бажають здійснювати обмін різними предметами через інтернет. Ця система дозволить користувачам легко додавати свої товари на аукціон, переглядати доступні лоти, робити ставки та здійснювати покупки. Окрім цього, це може зробити процес обміну речами цікавішим і захоплюючим, а також забезпечити безпечні та надійні угоди між користувачами.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ ВЕБДОДАТКУ АУКЦІОНУ РЕЧЕЙ

1.1. Постановка задачі

Сучасний світ означений постійним стрімким розвитком та розширенням можливостей для здійснення унікальних та захоплюючих покупок. Один із інноваційних і ефективних методів, які вривається на даний момент – це аукціони речей. Вони стають центральним елементом споживчого інтересу, надаючи можливість придбання унікальних предметів, які можуть вражати своєю унікальністю та цінністю або просто речей, які можуть знадобитися в повсякденному житті.

Питання проведення аукціонів речей здобуває все більше популярності через свою різноманітність та можливість отримати предмети, недоступні в звичайних магазинах. Це робить аукціони захоплюючим методом для задоволення індивідуальних потреб та побажань користувачів.

Мета даного проєкту полягає в розробці інноваційного вебдодатка, який сприятиме ефективному прогресу аукціонних процесів, а також спростить комунікацію між учасниками. Розробка буде спрямована на створення інструменту, який не лише дозволить учасникам придбати унікальні або прості речі, а й спростить весь цей процес, зекономивши їхній час та забезпечивши зручний доступ до аукціонів.

Основні етапи розробки вебдодатку для аукціону речей включатиме в себе:

1. Визначення вимог та функціоналу додатка:
 - а. Аналіз потреб користувачів та визначення ключових функціональних можливостей.
2. Проєктування дизайну додатка:
 - а. Розробка інтуїтивно зрозумілого та привабливого інтерфейсу для основної та адміністративної частини.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- b. Адаптація дизайну під різні розміри екранів для максимальної зручності користувачів.
- 3. Проектування та розробка архітектури додатка:
 - a. Створення серверної та клієнтської архітектури для забезпечення ефективності та безпеки додатка.
- 4. Розробка компонентів, сторінок та функціоналу:
 - a. Компонент авторизації та реєстрації учасників.
 - b. Зберігання інформації про лоти на аукціоні.
 - c. Обробка та збереження лотів.
 - d. Виведення списку аукціонів та функціонал замовлення лотів.
 - e. Взаємодія з користувачами через спосіб відправки електронних листів.
 - f. Модуль початкової ініціалізації користувачів на сайті.
 - g. Компонент фільтрації даних для зручного пошуку.
- 5. Тестування роботи додатку:
 - a. Використання тестових сценаріїв для перевірки правильності та надійності роботи додатка.

Результатом цього проєкту буде вебдодаток для аукціонів речей, який спростить та зробить більш доступним процес обміну інформацією між учасниками, створюючи зручні умови для успішних та захоплюючих аукціонних торгів.

1.2. Аналіз аналогів програмного продукту

Було проведено аналіз багатьох аналогів програмного продукту серед наявних вебдодатків, що спрямовані на організацію аукціонів з продажу речей. Дослідження показало, що кожен з розглянутих вебдодатків має свої власні переваги та обмеження, визначаючи їхній функціонал та можливості загалом.

У контексті цього дослідження, вебдодаток має можливість виокремитися шляхом поєднання найкращих аспектів різних платформ та

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

надавати унікальний підхід, спрямований на покращення ефективності та широкий вибір як вживаних, так і нових речей.

Один з варіантів, який є аналогом майбутнього розробленого додатку, є **eBay** (рис. 1.1) [9]. eBay – це один з найкращих вебсайтів онлайн-аукціону, оскільки він безкоштовний для покупців, як продавець, так і покупець мають захист від шахрайства, а також, користувач не обмежений однією країною. Можливість купувати та продавати на eBay у більшості країн.

Переваги:

- Надійний захист: Забезпечення захисту як продавця, так і покупця.
- Широкий асортимент товарів: Дана платформа пропонує різноманітні товари, від електроніки до унікальних колекційних предметів, що дає змогу розширювати цільову аудиторію.
- Система рейтингів та відгуків: eBay надає можливість користувачам залишати відгуки та оцінювати продавців, що дає змогу сприяти підвищенню довіри між сторонами.
- Простота: Платформа має інтуїтивний і легко зрозумілий інтерфейс для обох сторін – продавців і покупців.
- Історія торгів: Аукціон зберігає історію торгів для кожного лоту, що дозволяє відстежувати активність та робити свідомі ставки.

Недоліки:

- Обмежені опції фільтрації: Обмежена кількість фільтрів може ускладнити точне визначення параметрів товару, що важливо для покупців.
- Неповна інформація в описах товарів: Деякі продавці не надають достатньої інформації про товар, що ускладнює прийняти чітке рішення зі сторони покупця.
- Обмеження ставки: Обмеження на максимальну ставку.
- Реклама: Відображення реклами третіх сторін і списків від інших продавців.

- Висока залежність від системи відгуків: Система відгуків сприяє довірі, що може стати об'єктом зловживань, де певні випадки можуть вплинути на репутацію продавця.

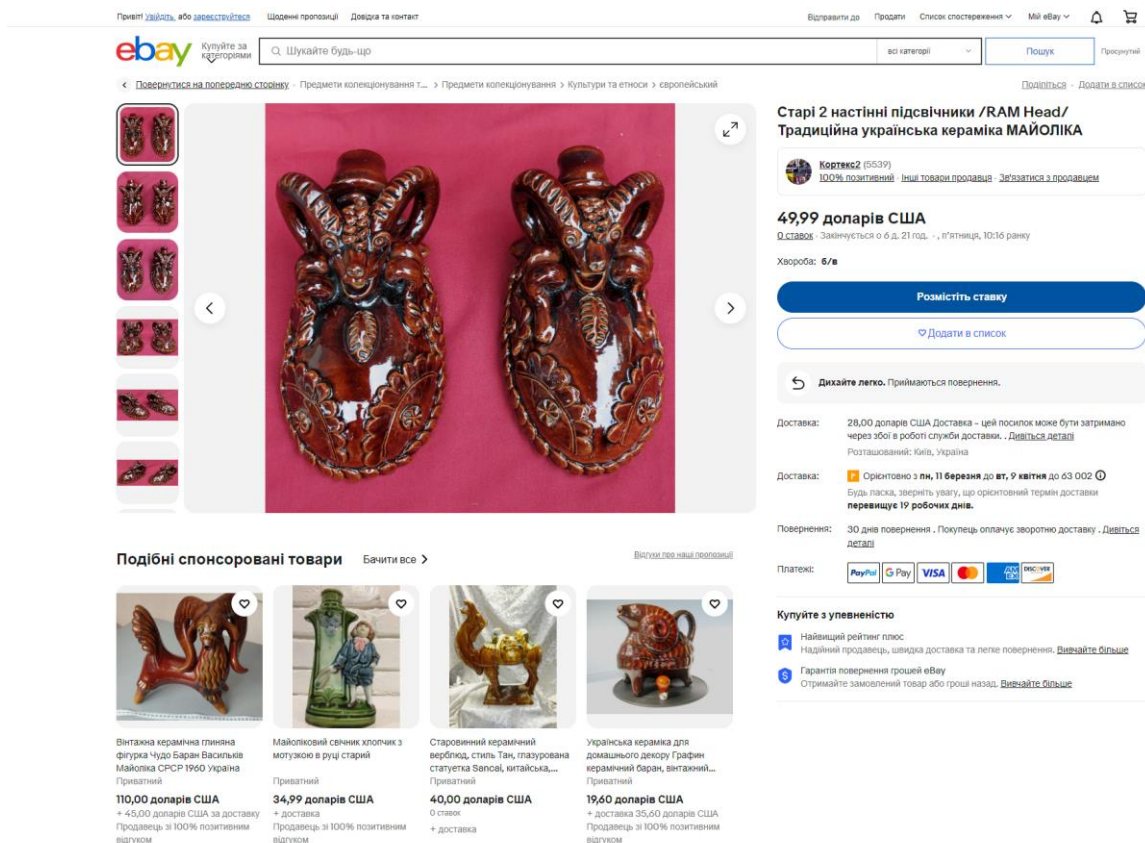


Рисунок 1.1 – eBay

Наступний варіант – це **AuctionNinja** (рис. 1.2) [10]. Це велика онлайн-платформа для проведення аукціонів з продажу речей, яка спеціалізується на спеціальних категоріях, таких як ювелірні вироби, мистецтво, антикваріат і предмети колекціонування. Низькі комісії та хороша підтримка продавців роблять його ідеальним для окремих осіб і невеликих продавців. Менша аудиторія означає нижчі ціни на товари.

Переваги:

- Широкий вибір: AuctionNinja пропонує широкий спектр лотів, доступних для торгів, що робить його непоганим ресурсом для пошуку унікальних та цікавих предметів.
- Зручність: Можливість легко робити ставки, відстежувати лоти та отримувати повідомлення про нові аукціони.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Kashalot (рис. 1.3) [11] – це український онлайн-аукціон, де можна знайти широкий спектр товарів, як нових, так і вживаних. Він пропонує зручну платформу для купівлі та продажу різноманітних предметів, від одягу та електроніки до антикваріату та нерухомості. Його інтерфейс відзначається простотою та зручністю в користуванні, що робить його привабливим для широкого кола користувачів.

Переваги:

- Великий вибір товарів: На Kashalot представлена велика кількість лотів, що робить його чудовим місцем для пошуку унікальних та важкодоступних предметів.
- Простота: Інтерфейс сайту простий та зрозумілий, що робить процес купівлі та продажу товарів максимально зручним.
- Безпечна платформа: Аукціон використовує сучасні методи захисту даних, щоб гарантувати безпеку своїх користувачів.
- Система відгуків: Kashalot надає можливість користувачам залишати відгуки.
- Історія торгів: Платформа зберігає історію торгів для кожного лоту, що дозволяє відстежувати активність та робити свідомі ставки.

Недоліки:

- Незручний інтерфейс: Інтерфейс може здатися складним та заплутаним для нових користувачів та навігація по сайту не завжди інтуїтивно зрозуміла.
- Відсутність деяких функцій: Відсутні деякі функції, які є стандартними для інших онлайн-аукціонів.
- Обмежені опції фільтрації: Обмежена кількість фільтрів може ускладнити точне визначення параметрів товару, що важливо для покупців.
- Неповна інформація в описах товарів: Деякі продавці не надають достатньої інформації про товар, що ускладнює прийняти чітке рішення зі сторони покупця.

- Необхідність знати правила аукціону: Для того, щоб успішно торгуватися, важливо знати та розуміти правила аукціону.

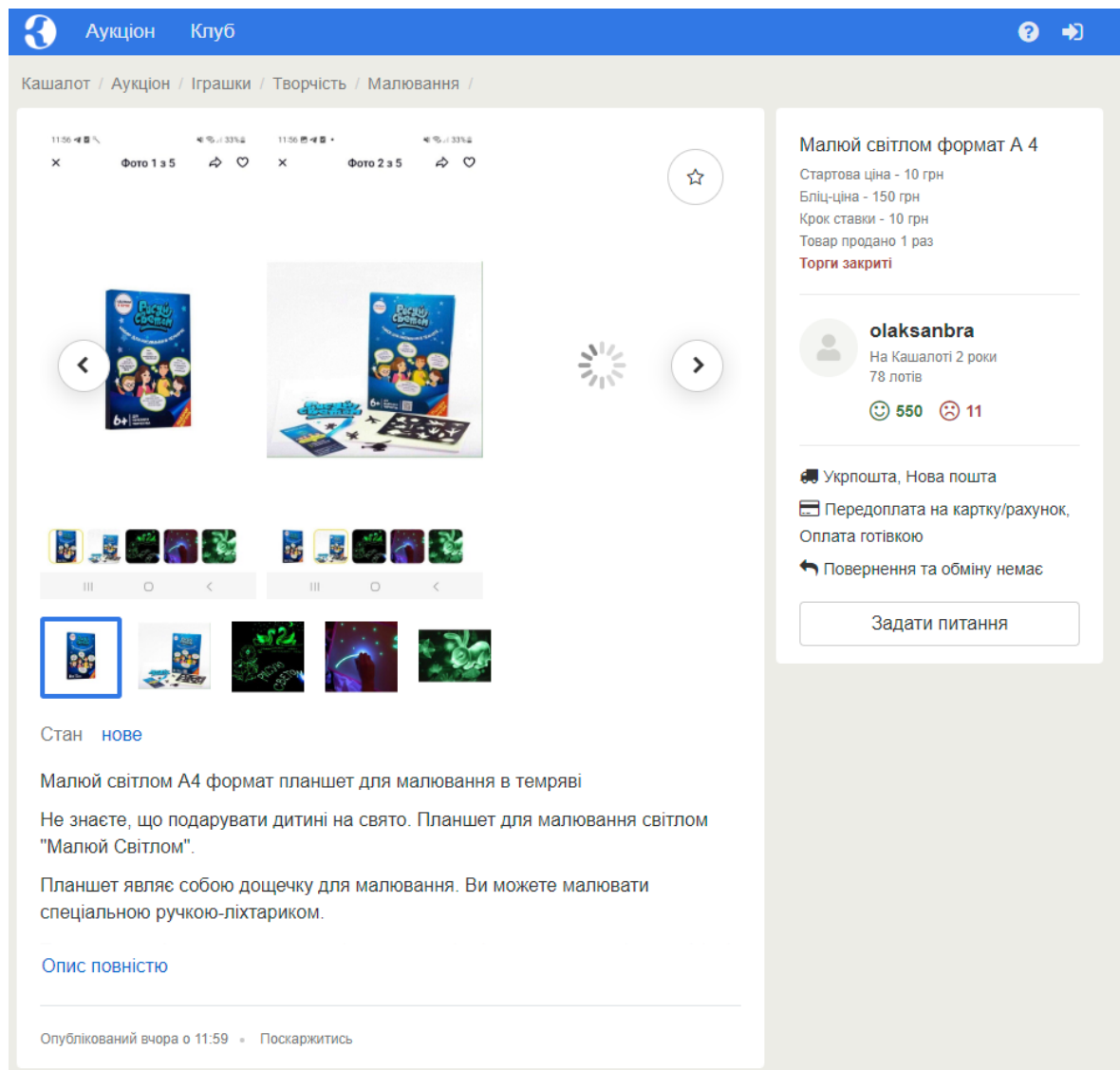


Рисунок 1.3 – Kashalot (Кашалот)

eBay, AuctionNinja та Kashalot, як аукціонні платформи, пропонують різноманіття товарів та послуг, з урахуванням своїх особливостей. Однак, і вони мають свої недоліки, такі як обмеженість фільтрації, пошуку та інших певних функцій. Врахувавши ці аспекти, вебдодаток включатиме ефективну систему пошуку, розширені опції фільтрації, а також вдосконалені функції для комфортної роботи з аукціоном. Забезпечення зручного та безпечного користувацького інтерфейсу, а також доступність, що може стати ключовою перевагою, роблячи додаток конкурентоспроможним та привабливим для різних користувачів у майбутньому.

1.3. Вибір архітектури веборієнтованої системи аукціону речей

Розробка веборієнтованої системи є складним завданням, вимагаючим уважного підходу до вибору архітектурної моделі [1]. Вибір правильної архітектури є важливим етапом, що визначатиме ефективність, масштабованість та зручність використання системи для користувачів. У даному контексті, буде розглянуто ключові аспекти вибору архітектури та визначено оптимальний напрямок для проєкту.

Вебдодаток – програма, з якою користувачі взаємодіють за допомогою браузера. Вона працює на віддалених серверах та надає різні функції або послуги користувачам через Інтернет. Як правило, користувачі отримують доступ так само як і до сайтів через URL-адресу [2].

У вебдодатках є серверна та клієнтська складова (клієнт-серверна модель) [7]. Користувачі взаємодіють із клієнтською частиною через інтерфейс, представлений у браузерах. За запитом від користувача дані передаються на сервер, де інформація обробляється відповідно до бізнес-логіки додатка, повертаючи відповідь [3].

Існують типи вебдодатків, які класифікують залежно від зовнішнього виду та функцій:

- **SPA** або односторінкова програма

Це односторінкова вебпрограма, яка завантажується на одну HTML-сторінку. Завдяки динамічному оновленню за допомогою JavaScript, під час використання не потрібно перезавантажувати або довантажувати додаткові сторінки. З приводу переваг, це висока швидкість, тобто всі ресурси завантажуються за одну сесію, а під час дій на сторінці дані просто змінюються, що дуже економить час. Гнучкість і чуйність користувальницького інтерфейсу за рахунок того, що вебсторінка всього одна, простіше побудувати насичений інтерфейс, зберігати відомості про сеанс, управляти станами уявлень і анімацією.

- **MPA** або багатосторінкова програма

Це багатосторінкові додатки, які працюють за традиційною схемою. Це означає, що при кожній незначній зміні даних або завантаженні нової інформації сторінка оновлюється. Такі додатки важче, ніж односторінкові, тому їх використання доцільно тільки в тих випадках, коли потрібно відобразити велику кількість контенту. Щодо переваг, це те, що можна оптимізувати кожну зі сторінок програми під потрібні ключові запити. Зручність, за рахунок простого інтерфейсу і класичної навігації. З недоліків, це тісний зв'язок між бекендом і фронтом, тому їх не виходить розвивати паралельно. Також, складна розробка, яка вимагає використання фреймворків як на стороні клієнта, так і на стороні сервера.

- **PWA** або прогресивний додаток

Це вебдодаток, створений з використанням певних технологій для досягнення цільових показників. Ця технологія використовує HTML, CSS та Javascript із можливостями WebAssembly, Web Storage, Service Workers та Manifests. Щодо переваг, PWA відрізняються високою швидкістю відгуку на дії користувача, можуть працювати без Інтернету (офлайн), надсилати push-сповіщення та завжди використовують HTTPS для захисту інформації. Серед недоліків, в основному пов'язують проблеми з функціями Bluetooth.

Коротко про API:

API [5] – це посередник між програмами, який задає правила «спілкування». Тому в його назві закладено поняття «інтерфейс» – межа між об'єктами. Одній програмі не важливо, як працює інша.

Існує три основні типи архітектур API: RPC, REST та SOAP. Їх можна назвати «форматами», кожен з яких має унікальні характеристики та використовується для різних цілей.

- **RPC**

Це простий формат для надсилання кількох параметрів та отримання результатів. API RPC викликають дії чи процеси, тоді як інші API переважно обмінюються даними чи ресурсами. RPC може використовувати дві мови для кодування – JSON та XML.

- SOAP

Використовує лише мову кодування XML. Цей тип роботи з API вважається найскладнішим: SOAP дуже структурований та строго контрольований формат [4].

- REST

REST вважається більш простою альтернативою SOAP. Кожна одиниця інформації для цього виду API унікальна URL-адреса, яку можна запросити. Використовують для швидкого обміну простими параметрами, у тому числі тими, з яких складаються бази даних. Тому REST API добре підходить для взаємодії великих баз даних [4].

Як було сказано раніше, вибір правильної архітектури є важливим етапом, тому, в даному випадку, порівнюємо три архітектурні моделі:

1) Мікросервісна архітектура (Microservice architecture)

Це підхід, коли єдиний додаток будується як сукупність невеликих, самодостатніх, незалежних, не тісно зв'язаних сервісів, що спілкуються між собою за допомогою легких механізмів як то HTTP, gRPC, AMQP. Ці сервіси побудовані навколо бізнес-потреб (кожен відповідальний за конкретний процес) та розгортаються незалежно з використанням повністю автоматизованого середовища. Мікросервіси допомагають доставляти зміни швидше, безпечніше і з більш високою якістю, тобто зберігати швидкість розвитку продукту, навіть коли той стає неосяжних розмірів. Адже не тісно зв'язані сервіси дають можливість проводити зміни з більшою частотою ітерацій мінімізуючи вплив змін на решту частин системи. З недоліків, такий підхід додає додаткову складність проєкту в цілому. При роботі з мікросервісами доводиться більше розгортати, ускладнюється система моніторингу, сильно розростається кількість можливих збоїв.

2) Монолітна архітектура (Monolithic Architecture)

Це підхід до розробки програмного забезпечення, при якому весь додаток розробляється як одна єдина технологічна система. Всі компоненти додатку взаємодіють один з одним і розгортання відбувається на одному

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

сервері або групі серверів. Монолітна архітектура має декілька переваг. Вона зазвичай простіша в розробці та тестуванні, оскільки всі компоненти додатку взаємодіють безпосередньо один з одним, що дозволяє швидко та легко вирішувати проблеми. Крім того, монолітні додатки можуть бути менш складними в управлінні, оскільки все програмне забезпечення працює на одній технологічній платформі. Однак, монолітні додатки можуть стати проблемою при масштабуванні, оскільки розширення системи може бути обмеженим через те, що всі компоненти додатку залежать один від одного.

3) Клієнт-серверна архітектура

Клієнт-серверна архітектура [7] набула своєї популярності завдяки динамічному розвитку мережі Інтернет та зосередження значної частини інформації в базах даних на серверах. Дану архітектуру можна означити, як концепцію інформаційної мережі, в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів. Така архітектура визначає такі типи компонентів:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Правила взаємодії між клієнтом і сервером називаються протоколом обміну (протоколом взаємодії). Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Щодо переваг, це економія ресурсів, що дозволяє забезпечити хорошу продуктивність програми. Відсутність дублювання коду. З недоліків, це те, що якщо виникне збій у роботі однієї з ключових компонентів даної архітектури, то тоді перестане працювати система.

Отже, було обрано клієнт-серверну архітектуру. Вона виявилася найбільш оптимальною серед розглянутих архітектурних підходів. Застосування цієї архітектури надає велику гнучкість та чітку взаємодію між

клієнтом та сервером, що дозволяє ефективно розробляти та управляти проєктом. Принцип розділення відповідальностей між клієнтом та сервером сприяє простоті управління та розширенню функціональності, а також полегшує процес впровадження нових можливостей у додаток.

У даному випадку, також було обрано тип API як REST, що є стандартом для взаємодії між клієнтом та сервером у веброзробці. REST дозволяє простий та ефективний обмін даними, забезпечуючи стабільність та легкість взаємодії між компонентами системи. Цей підхід допомагає забезпечити послідовність в структурі та взаємодії.

1.4. Обґрунтування вибору інструментальних засобів та вимоги апаратного забезпечення

Вибір інструментальних засобів для розробки вебдодатка є важливим кроком, який потребує ретельного врахування специфіки проєкту та вимог до його функціональності.

Для фронтенду було обрано React.js [15] завдяки його високій популярності, гнучкості та швидкості розробки інтерфейсу. Ця бібліотека дозволяє легко впроваджувати компонентний підхід, роблячи код більш структурованим та з легкістю підтримуючи його масштабованість.

Додатковою бібліотекою для стилізації інтерфейсу було обрано Material UI [19]. Це рішення базується на дизайн-мові Google Material Design і пропонує готові компоненти та стилі, що спрощує процес розробки та гарантує сучасний та стилізований вигляд додатку.

Для керування станом додатку було обрано Redux [18], що є потужним інструментом для управління станом застосунку в React-додатках. Redux дозволяє ефективно організовувати та управляти станом додатку, що є критично важливим для додатків зі складною логікою та багатьма взаємозв'язаними компонентами. Використання Redux спрощує відлагодження та підтримку коду, забезпечуючи однозначність управління станом та покращуючи його масштабованість.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						20
Змн.	Арк.	№ док.ум.	Підпис	Дата		

На стороні бекенда використовується Node.js [12] та Express.js [13]. Платформа Node.js забезпечує високу швидкість виконання завдяки використанню JavaScript на сервері, а Express.js є легковаговим та потужним фреймворком для роботи з HTTP-запитами. Цей тандем дозволяє швидко створювати надійні та ефективні серверні застосунки.

Для роботи з відсиленням повідомлень на пошту використовується Nodemailer [14]. Ця бібліотека забезпечує простий та надійний спосіб відправки електронних листів, що дозволяє ефективно взаємодіяти з користувачами щодо їхніх дій на аукціоні.

Для роботи з базою даних було обрано PostgreSQL [27] через його надійність, розширені можливості та підтримку транзакцій. PostgreSQL є потужною реляційною системою управління базами даних, що надає надійний та ефективний механізм зберігання даних для аукціонного вебдодатку.

Середовищем розробки є Visual Studio Code, який володіє інтегрованою підтримкою для JavaScript, що робить його ідеальним вибором для розробки. Також, за його легкості використання, багатий функціонал та широку підтримку розширень, що дуже полегшує розробку.



Рисунок 1.4 – Стек технологій

Використовується Git для контролю версій коду та спільної роботи над проєктом. Git дозволяє зберігати історію змін та відслідковувати розвиток проєкту з часом. Його використання забезпечує стабільну та організовану розробку програмного забезпечення.

Загалом, вибір цих технологій та інструментів (як показано на рис. 1.4) здійснювався з урахуванням того, що вони широко використовуються у світі програмування та відповідають всім потребам вебдодатка. Було обрано найпоширеніші та найефективніші засоби для розробки, щоб забезпечити високу якість та продуктивність проєкту. Враховано всі вимоги проєкту, щоб зробити його більш зручним та відповідним для користувачів.

Висновки до першого розділу

У першому підрозділі визначено мету створення вебдодатку аукціону речей та здійснено огляд існуючих аукціонних платформ, таких як eBay, AuctionNinja та Kashalot. Хоча ці платформи пропонують широкий вибір товарів та послуг, кожна з них має свої обмеження і недоліки, зокрема, обмеженість функцій фільтрації. На відміну від існуючих платформ, вебдодаток планує забезпечити розширені опції фільтрації, щоб користувачі могли знаходити потрібні їм товари швидко та ефективно. Окрім цього, буде створено зручний та безпечний користувацький інтерфейс, що стане ключовою перевагою додатку і зробить його простішим у використанні.

У другому підрозділі розглянуто архітектуру вебдодатку, вибрані технології та розроблення компонентів, які сприятимуть досягненню цілей та забезпечать успішну реалізацію концепції аукціону речей.

У третьому підрозділі було ретельно розглянуто архітектуру вебдодатку. Обрано клієнт-серверну архітектуру для забезпечення ефективної взаємодії між фронтендом та бекендом. Це дозволить досягти високої продуктивності та масштабованості системи, а також забезпечить зручність у розробці та підтримці.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

У четвертому підрозділі детально описано вибрані технології та стек, які будуть використані у розробці вебдодатку. Для фронтенду обрано React.js та Material UI для швидкої та зручної розробки користувацького інтерфейсу. На бекенді використовуватимуться Node.js та Express.js для створення надійного та ефективного серверного середовища. Для керування станом додатку використовуватиметься Redux. Базою даних обрано PostgreSQL, а для взаємодії з користувачами через електронну пошту буде використано Nodemailer. Середовищем розробки обрано Visual Studio Code, який забезпечує зручне та продуктивне рішення для розробки вебдодатків.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБОРІЄНТОВАНОЇ СИСТЕМИ АУКЦІОНУ РЕЧЕЙ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Для забезпечення безперервного функціонування та відповідності установленим стандартам і вимогам, було сформульовано високорівневі вимоги до вебдодатку аукціону речей. Ці вимоги спрямовані на забезпечення надійності та ефективності роботи системи, а також виконання всіх необхідних стандартів і вимог.

Бізнес вимоги

1. Ціль проєкту

Основною ціллю розробки вебдодатку аукціону речей є створення зручної та безпечної платформи для купівлі і продажу товарів через аукціон. Додаток повинен мати інтуїтивно зрозумілий інтерфейс, вдосконалені функції пошуку та фільтрації, простий процес реєстрації, включаючи перевірку особи та надійний механізм оплати, щоб бути конкурентоспроможним і привабливим для користувачів.

2. Представлення проєкту

Проєкт вебдодатку аукціону речей передбачає створення сучасної платформи для проведення онлайн-аукціонів. Додаток буде представлений як універсальний інструмент для купівлі та продажу товарів, що дозволяє користувачам брати участь в аукціонах, робити ставки, додавати товари до списку відстеження та здійснювати покупки. Платформа повинна бути доступною з будь-якого пристрою, що має доступ до Інтернету, і відповідати сучасним стандартам безпеки та зручності користування. Інтерфейс буде розроблений з використанням передових технологій для забезпечення продуктивності та привабливого користувацького досвіду.

Вимоги користувачів

1. Зручний інтерфейс

2. Реєстрація та авторизація

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Управління лотами
4. Пошук та фільтрація
5. Сортування
6. Система ставок
7. Список відстеження
8. Купівля
9. Взаємодія з користувачами
10. Адміністрування
11. Статистика

Адміністратор

Функціональні вимоги

- Мати доступ до адмін-панелі для управління всіма аспектами системи.
- Можливість переглядати, створювати, редагувати та видаляти лоти.
- Змінювати статус лотів (активний, завершений, видалений тощо).
- Управління користувачами (блокування, розблокування).
- Управління категоріями товарів (створення, редагування, видалення).
- Перегляд та управління історією лотів і ставками.
- Управління списком відстеження, замовленнями та платежами.
- Надання підтримки користувачам, відповідь на запити через систему повідомлень.
- Генерація та перегляд аналітичних звітів (статистика продажів, активність користувачів).
- Відправлення системних сповіщень та повідомлень користувачам.

Нефункціональні вимоги

- Високий рівень доступу до системи з надійною автентифікацією.
- Забезпечення конфіденційності та безпеки даних.
- Інтуїтивно зрозумілий інтерфейс адмін-панелі.
- Висока доступність системи з мінімальним часом простою для забезпечення безперервної роботи.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

- Швидкий відгук інтерфейсу для забезпечення ефективної роботи адміністратора.

Продавець

Функціональні вимоги

- Можливість реєстрації та авторизації в системі.
- Створення нових лотів з детальною інформацією (опис, фотографії, початкова ціна, категорії).
- Редагування своїх лотів.
- Перегляд історії ставок та статусу своїх лотів.
- Перегляд та управління своїм списком відстеження.
- Редагування особистої інформації в профілі (ім'я, контактна інформація, пароль тощо).

Нефункціональні вимоги

- Зручний та інтуїтивний інтерфейс для управління лотами.
- Безпечне зберігання особистих даних та інформації про лоти.
- Надійний механізм автентифікації.
- Масштабованість системи для підтримки збільшення кількості лотів та користувачів.

Покупець

Функціональні вимоги

- Можливість реєстрації та авторизації в системі.
- Перегляд всіх доступних лотів з можливістю сортування та фільтрації.
- Використання пошуку для знаходження конкретних лотів.
- Участь в аукціонах шляхом розміщення ставок.
- Додавання лотів до списку відстеження для подальшого спостереження.
- Здійснення покупки лотів.
- Взаємодія з продавцями через систему повідомлень.
- Редагування особистої інформації в профілі (ім'я, контактна інформація, пароль тощо).

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Нефункціональні вимоги

- Зручний та інтуїтивний інтерфейс для перегляду та взаємодії з лотами.
- Надійний механізм автентифікації.
- Безпечне зберігання особистих даних.
- Інтуїтивно зрозумілий процес реєстрації та участі в аукціонах.

Аналіз потреб користувачів та визначення ключових ролей дозволили створити діаграми використання вебдодатку аукціону речей для покупців (рис. 2.1), продавців (рис. 2.2) та адміністраторів (рис. 2.3).

На основі розроблених варіантів використання системи були створені ключові прецеденти, що описують конкретні сценарії або взаємодії користувачів із платформою. Нижче наведено основні та найважливіші прецеденти для вебдодатку аукціону речей, що ілюструють взаємодію користувачів з різними функціями системи.



Рисунок 2.1 – Варіанти використання додатку аукціону речей із точки зору ролі Користувача



Рисунок 2.2 – Варіанти використання додатку аукціону речей із точки зору ролі Продавця



Рисунок 2.3 – Варіанти використання додатку аукціону речей із точки зору ролі Адміністратор

Прецедент: Реєстрація

Актори: Користувач

Короткий опис: Користувач може зареєструвати новий обліковий запис у системі для доступу до лотів (участь в ставках).

Передумови: Користувач неавторизований в системі.

Асоціація: Користувач обирає роль (покупець або продавець), потім вводить особисті дані.

Включення: Після успішної реєстрації, користувач автоматично авторизується у системі.

Основний потік:

1. Користувач обирає опцію «Реєстрація».
2. Користувач обирає роль (покупець або продавець).
3. Система відображає форму реєстрації.
4. Користувач вводить особисті дані.
5. Користувач надсилає форму реєстрації.
6. Система відправляє пін-код для перевірки особи на вказану електронну пошту користувача.
7. Користувач вводить отриманий пін-код у відповідне поле на сторінці введення пін-коду.
8. Система перевіряє правильність введеного пін-коду.
 - Якщо пін-код правильний:
 - Система створює обліковий запис користувача та авторизує його.
 - Якщо пін-код неправильний:
 - Система відображає повідомлення про помилку і просить користувача ввести пін-код ще раз.

Прецедент: Авторизація

Актори: Користувач

Короткий опис: Користувач може увійти в систему, використовуючи свій обліковий запис та пароль.

Передумови: Користувач неавторизований в системі.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Асоціація: Користувач вводить свій логін та пароль.

Включення: Після успішної авторизації, користувач отримує доступ до особистого облікового запису.

Основний потік:

1. Користувач вводить свій логін та пароль.
2. Система перевіряє вірність даних.
3. Система відправляє пін-код для перевірки особи на вказану електронну пошту користувача.
4. Користувач вводить отриманий пін-код у відповідне поле на сторінці введення пін-коду.
5. Система перевіряє правильність введеного пін-коду.
 - Якщо пін-код правильний:
 - Система відкриває особистий обліковий запис користувача.
 - Якщо пін-код неправильний:
 - Система відображає повідомлення про помилку і просить користувача ввести пін-код ще раз.

Прецедент: Перегляд списку доступних лотів

Актори: Користувач

Короткий опис: Користувач може переглядати список активних лотів та вибирати ті, які його цікавлять.

Передумови: Користувач авторизований в системі.

Асоціація: Користувач обирає сторінку «Лоти».

Включення: Користувач може вибрати конкретний лот для подальшого перегляду та участі в ньому.

Основний потік:

1. Користувач переходить на сторінку «Лоти».
2. Система відображає список аукціонів з інформацією про них.

Прецедент: Вибір лоту

Актори: Користувач

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Короткий опис: Користувач може вибрати конкретний лот для подальшого перегляду та участі в ньому.

Передумови: Користувач авторизований в системі та переглядає список лотів.

Асоціація: Користувач обирає лот зі списку відкритих лотів.

Включення: Після вибору лоту, користувач може переглядати інформацію про обраний лот.

Основний потік:

1. Користувач обирає лот зі списку відкритих лотів.
2. Система відображає інформацію про обраний лот.

Прецедент: Участь у торгах

Актори: Користувач

Короткий опис: Користувач може взяти участь у торгах на обрану річ на сторінці даного лоту.

Передумови: Користувач авторизований в системі та переглядає інформацію про лот.

Асоціація: Користувач робить ставку (-и) на річ на сторінці про даний лот.

Включення: Після участі у торгах користувач отримує повідомлення про результати.

Основний потік:

1. Користувач робить ставку (-и) на обрану річ на сторінці про даний лот.
2. Система відображає ставки і повідомляє про результати торгів.

Прецедент: Створення нового лоту

Актори: Продавець

Короткий опис: Продавець може створити новий лот для продажу річі.

Передумови: Продавець авторизований в системі.

Асоціація: Продавець вводить інформацію про річ і параметри лоту.

Включення: Після створення лоту, система відображає його в списку відкритих лотів.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Основний потік:

1. Продавець обирає опцію «Створити лот».
2. Продавець вводить інформацію про річ і параметри лоту.
3. Система створює новий лот та відображає його в списку лотів.

Прецедент: Модерація аукціонів (адміністратор)

Актори: Адміністратор

Короткий опис: Адміністратор системи може модерувати лоти, виявляти порушення правил і втручатися у випадку нечесної гри.

Передумови: Адміністратор авторизований в системі та переглядає лоти.

Асоціація: Адміністратор переглядає список лотів та може вжити заходи, такі як завершення аукціону або блокування користувачів.

Основний потік:

1. Адміністратор переглядає список аукціонів.
 - У разі виявлення порушень правил, адміністратор вживає заходи, такі як закриття лоту або блокування користувачів.

На основі зібраної та опрацьованої інформації під час аналізу вимог та сценаріїв використання програмного продукту, було створено базову діаграму класів. Нижче, також продемонстровано головні класи системи:

1. **User** – представляє користувача системи та містить інформацію про користувача, таку як ім'я, прізвище, по батькові, електронна пошта, пароль, роль (покупець, продавець або адміністратор), телефонний номер, баланс та інші деталі. Користувач може взаємодіяти з лотами, робити ставки, проводити платежі та керувати своїм списком відстеження.
2. **Lot** – представляє лот на аукціоні та містить інформацію про нього, таку як заголовок, опис, стартова ціна, поточна ціна, кінцева дата, статус (відкритий, закритий, очікує), категорія, ціна «Купити зараз», мінімальний крок ставки, резервна ціна, кількість ставок та інші деталі.
3. **Bid** – представляє ставку, зроблену користувачем на лот, та містить інформацію про ставку, таку як сума ставки, ідентифікатор

користувача, що зробив ставку, та ідентифікатор лоту, на який зроблена ставка.

4. **AuctionHistory** – представляє історію змін цін на лот та містить інформацію про лот, попередню та нову ціну, ідентифікатор ставки, яка спричинила зміну ціни.
5. **Category** – представляє категорію, до якої належать лоти, та містить інформацію про категорію, таку як назва, опис та зображення.
6. **Watchlist** – представляє список відстеження користувача та містить інформацію про ідентифікатори користувача та лоту, які користувач відстежує.
7. **Payment** – представляє платіж, зроблений користувачем, та містить інформацію про суму платежу, ідентифікатор користувача, ідентифікатор лоту, статус платежу (очікує, завершено, не вдалося, повернено), тип платежу (депозит, покупка, зняття, продаж), комісію та ідентифікатор сесії.
8. **Order** – представляє замовлення на лот та містить інформацію про ідентифікатори користувача та лоту, суму замовлення та статус замовлення (завершено, обробляється, скасовано).

Зв'язки між класами:

- Клас User має зв'язок "один до багатьох" з класом Lot (користувач створює багато лотів).
- Клас User має зв'язок "один до багатьох" з класом Bid (користувач робить багато ставок).
- Клас User має зв'язок "один до багатьох" з класом Payment (користувач здійснює багато платежів).
- Клас User має зв'язок "один до багатьох" з класом Order (користувач має багато замовлень).
- Клас Lot має зв'язок "один до багатьох" з класом Bid (лот отримує багато ставок).

- Клас Lot має зв'язок "один до багатьох" з класом AuctionHistory (лот має багато записів історії аукціону).
- Клас Lot має зв'язок "один до багатьох" з класом Payment (лот включає багато платежів).
- Клас Lot має зв'язок "один до багатьох" з класом Order (лот включає багато замовлень).
- Клас Category має зв'язок "один до багатьох" з класом Lot (категорія включає багато лотів).
- Клас Bid має зв'язок "один до одного" з класом AuctionHistory (кожна ставка може бути пов'язана з записом в історії аукціону).

2.2. Розробка бази даних системи

Розгляд бази даних системи – це невід'ємна складова вебдодатку аукціону речей. Вона включає в себе дев'ять таблиць:

- Користувачі;
- Лоти;
- Ставки;
- Історії лотів;
- Категорії;
- Списки відстеження;
- Платежі;
- Замовлення;
- Міграції Sequelize.

Кожна таблиця охоплює різні аспекти системи. Перша та одночасно ключова таблиця – «Користувачі» (табл. 2.1).

Таблиця 2.1

Опис таблиці «Users»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор користувача
firstName	STRING	-	-	Ім'я користувача
lastName	STRING	-	-	Прізвище користувача

patronymic	STRING	-	-	По батькові користувача
email	STRING	-	-	Електронна адреса користувача
password	STRING	-	-	Пароль користувача
role	ENUM	-	-	Роль користувача
phoneNumber	STRING	-	-	Номер телефону користувача
PinCode	STRING	-	-	Пін-код користувача
PinCodeExpirationDate	DATE	-	-	Час, до якого доступний пін-код
pinCodeAttempts	INTEGER	-	-	Кількість разів введення пін-коду
pinCodeSendAttempts	INTEGER	-	-	Кількість разів надсилення пін-коду
pinCodeSendAttemptsResetTime	DATE	-	-	Час, до якого відбудеться блокування можливості надсилення пін-коду
resetPasswordToken	STRING	-	-	Токен для можливості скидання паролю
resetPasswordExpiration	DATE	-	-	Час, до якого доступний токен
passwordResetAttempts	INTEGER	-	-	Максимальна кількість разів скидання паролю
passwordResetAttemptsExpiration	DATE	-	-	Час, до якого відбудеться блокування можливості скидання паролю
companyName	STRING	-	-	Назва компанії
companySite	STRING	-	-	Сайт компанії
position	STRING	-	-	Позиція користувача в компанії
balance	DECIMAL	-	-	Баланс користувача
isBlocked	BOOLEAN	-	-	Блокування користувача
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Наступна таблиця – це «Лоти» (табл. 2.2), вона є важливим елементом бази даних вебдодатку аукціону речей, оскільки служить основним механізмом для управління та взаємодії з лотами, які представляють різноманітні товари та послуги на аукціоні.

Таблиця 2.2

Опис таблиці «Lots»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор лоту
userId	INTEGER	-	+	Ідентифікатор користувача
title	STRING	-	-	Заголовок лоту
description	TEXT	-	-	Опис до лоту
startingPrice	DECIMAL	-	-	Початкова ціна лоту
currentPrice	DECIMAL	-	-	Поточна ціна лоту
endDate	DATE	-	-	Дата та час завершення лоту
imageUrls	ARRAY	-	-	Масив зображень до лоту
status	ENUM	-	-	Статус лоту
categoryId	INTEGER	-	+	Ідентифікатор категорії
buyNowPrice	DECIMAL	-	-	Ціна купівлі лоту відразу
bidIncrement	DECIMAL	-	-	Крок ставки
reservePrice	DECIMAL	-	-	Резервна ціна
bidCount	INTEGER	-	-	Кількість ставок
winnerId	INTEGER	-	-	Ідентифікатор переможця
originaEndDate	DATE	-	-	Початкова дата завершення лоту
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Таблиця «Ставки» (табл. 2.3) відіграє ключову роль у системі аукціону речей, забезпечуючи можливість користувачам робити ставки на лоти, які є відкритими.

Таблиця 2.3

Опис таблиці «Bids»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор ставки
amount	DECIMAL	-	-	Сума ставки
userId	INTEGER	-	+	Ідентифікатор користувача
lotId	INTEGER	-	+	Ідентифікатор лоту
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Таблиця «Історії лотів» (табл. 2.4) зберігає інформацію про всі події, що стосуються конкретних лотів на аукціоні речей.

Таблиця 2.4

Опис таблиці «AuctionHistories»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор історії лоту
lotId	INTEGER	-	+	Ідентифікатор лоту
bidId	INTEGER	-	+	Ідентифікатор ставки
oldPrice	DECIMAL	-	-	Початкова ціна лоту
newPrice	DECIMAL	-	-	Поточна ціна лоту
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Таблиця «Категорії» (табл. 2.5) відображає всі доступні категорії товарів, які можуть бути виставлені на аукціон.

Таблиця 2.5

Опис таблиці «Categories»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор категорії
image	STRING	-	-	Зображення до категорії
name	STRING	-	-	Назва лоту
description	TEXT	-	-	Опис до лоту
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Таблиця «Списки відстеження» (табл. 2.6) містить інформацію про товари, які користувачі додали до своїх списків відстеження.

Таблиця 2.6

Опис таблиці «Watchlists»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор списку відстеження
userId	INTEGER	-	+	Ідентифікатор користувача
lotId	INTEGER	-	+	Ідентифікатор лоту
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Таблиця «Платежі» (табл. 2.7) зберігає дані про фінансові транзакції, пов'язані з аукціонними операціями, такі як оплата за придбання лотів, зняття коштів з рахунку або поповнення тощо.

Опис таблиці «Payments»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор платежу
amount	DECIMAL	-	-	Сума
userId	INTEGER	-	+	Ідентифікатор користувача
lotId	INTEGER	-	+	Ідентифікатор лоту
status	ENUM	-	-	Статус платежу
type	ENUM	-	-	Тип платежу
commission	DECIMAL	-	-	Комісія
sessionId	STRING	-	-	Сесія при оформленні покупки
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Таблиця «Замовлення» (табл. 2.8) відображає інформацію про замовлення, що створюються користувачами на аукціоні. Кожне замовлення містить дані про лот, який був виграний або придбаний, деталі оплати, а також іншу важливу інформацію, пов'язану з процесом замовлення.

Таблиця 2.8

Опис таблиці «Orders»

Поле	Тип даних	ПК	ЗК	Призначення
id	INTEGER	+	-	Ідентифікатор замовлення
userId	INTEGER	-	+	Ідентифікатор користувача
lotId	INTEGER	-	+	Ідентифікатор лоту
amount	DECIMAL	-	-	Сума
status	ENUM	-	-	Статус замовлення
createdAt	DATE	-	-	Датата час створення
updatedAt	DATE	-	-	Дата та час щоразового оновлення

Таблиця «Міграції Sequelize» (табл. 2.9) призначена для зберігання даних про міграції в механізмі міграцій Sequelize [21].

Таблиця 2.9

Опис таблиці «SequelizeMeta»

Поле	Тип даних	ПК	ЗК	Призначення
name	STRING	+	-	Ідентифікатор міграції

Структура бази даних вебдодатку аукціону речей охоплює основні компоненти, які включають профілі користувачів, інформацію про лоти, дані про ставки та інші важливі елементи, необхідні для оптимального зберігання

та обробки інформації. Це забезпечує ефективну роботу системи та забезпечує безперервне функціонування. Нижче наведено ER-діаграму бази даних (рис. 2.4), яка ілюструє структуру та взаємозв'язки між таблицями в системі аукціону речей.

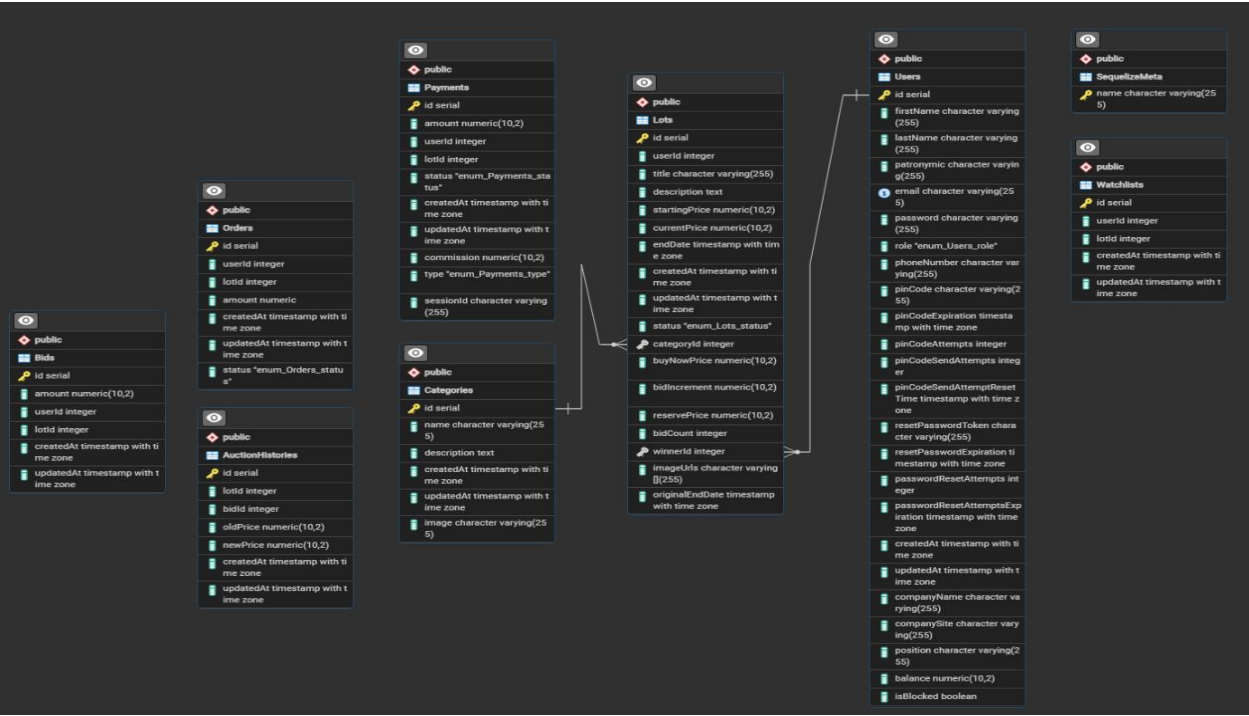


Рисунок 2.4 – ER-діаграма зв'язків бази даних

2.3. Проєктування та реалізація алгоритмів роботи системи

Потрапивши на сайт, користувач бачить головну сторінку, де він має можливість зареєструватися або увійти в систему, а також переглянути доступні лоти. Один з ключових процесів вебдодатку – це авторизація користувача, яка дозволяє йому брати участь в лотах або щось виставляти на продаж. Після успішної авторизації користувач може насолоджуватися повним спектром можливостей вебдодатку, включаючи участь у лотах, виставлення товарів на продаж, перегляд свого профілю, редагування персональної інформації, відстеження активних лотів і ставок, а також управління своїми лотами. Авторизація є критичним елементом безпеки вебдодатку, що забезпечує захист персональних даних користувачів. Розглянемо діаграму діяльності для авторизації (рис. 2.5).

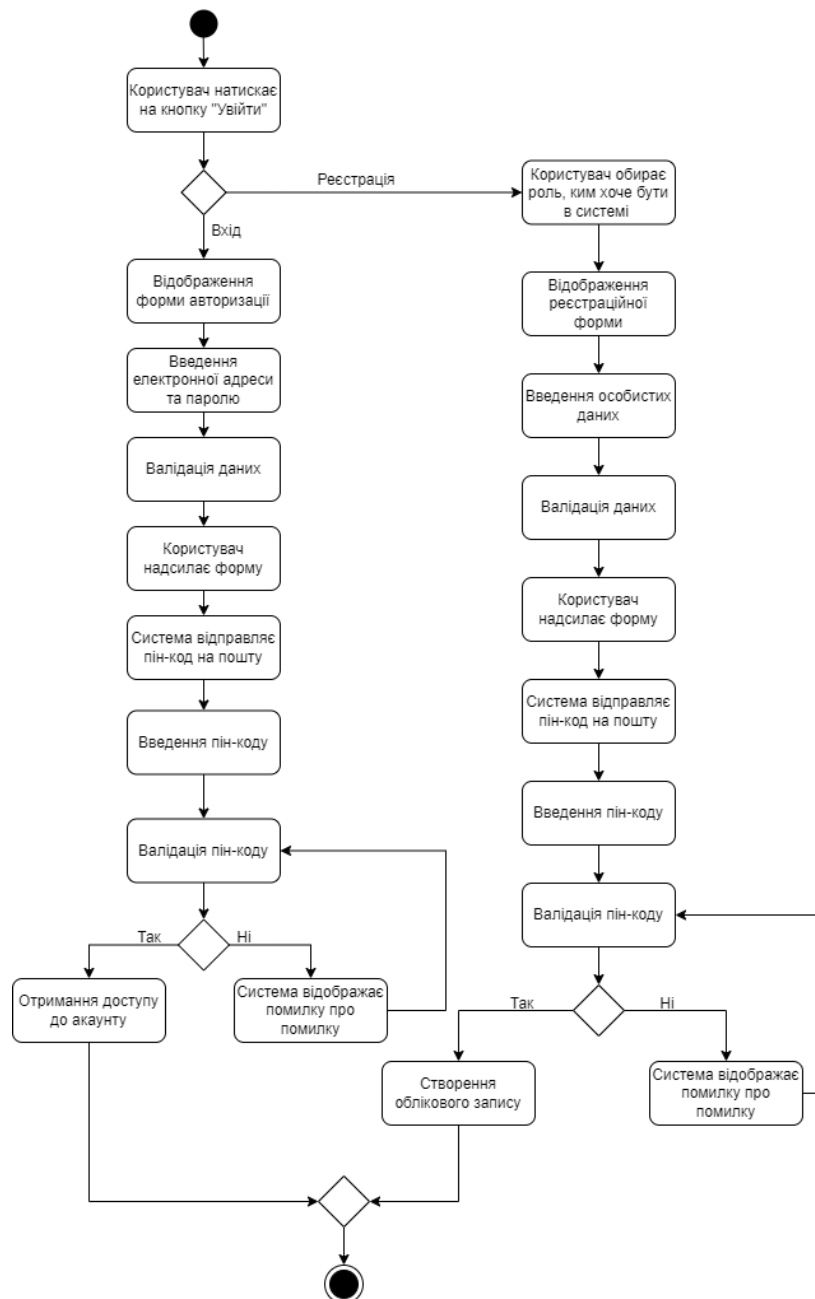


Рисунок 2.5 – Діаграма діяльності Авторизації

Процес авторизації починається з натискання кнопки «Увійти» на головній сторінці. Користувач вводить свою електронну адресу та пароль у відповідні поля форми авторизації. Після цього система перевіряє правильність введених даних. Якщо дані вірні, система відправляє на електронну пошту користувача пін-код для підтвердження особистості. Користувач вводить отриманий пін-код у відповідне поле на сайті. Якщо пін-код вірний, користувач успішно отримує доступ до облікового запису. У разі невірних даних система виводить повідомлення про помилку, і користувачеві

пропонується спробувати знову або скористатися функцією відновлення пароля.

Далі, одним з найважливіших алгоритмів у системі аукціону – це процес закриття лоту (рис. 2.6). Коли наближається завершення лоту, система починає ретельно перевіряти всі зроблені ставки. Спершу визначається найбільша ставка серед усіх учасників. Якщо така ставка є і вона перевищує резервну ціну, встановлену для лоту, система ідентифікує користувача, який зробив цю ставку. Цей користувач оголошується переможцем лоту.

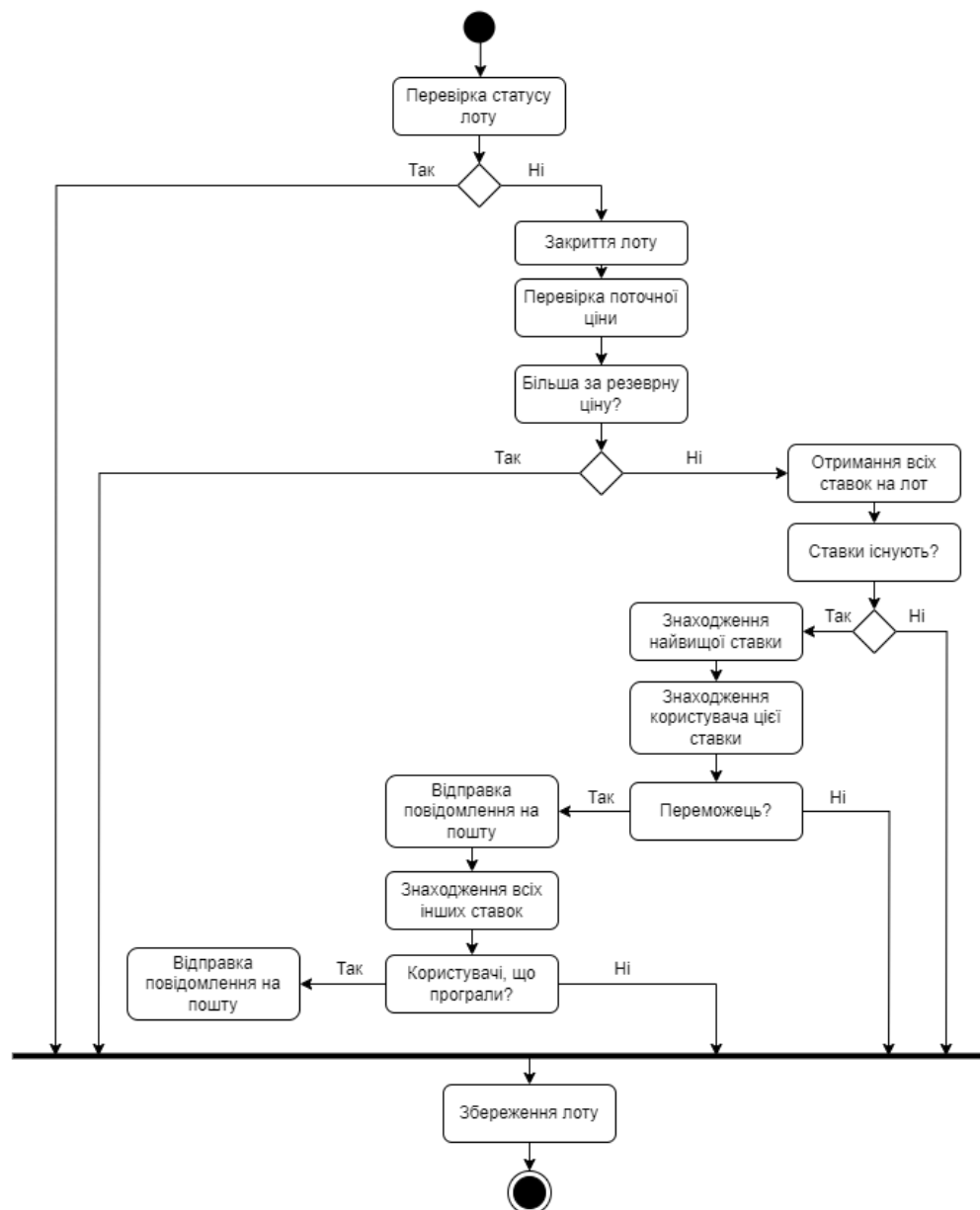


Рисунок 2.6 – Діаграма діяльності Процесу закриття лоту

Наступний етап включає відправлення переможцю повідомлення на електронну пошту з інформацією про його перемогу. Це важливо для

забезпечення прозорості процесу та інформування користувача про подальші кроки. Одночасно система аналізує всі інші ставки, визначає користувачів, які брали участь у торгах за цей лот, і надсилає їм повідомлення про те, що вони не виграли. Цей комплексний процес включає декілька кроків, кожен з яких важливий для забезпечення коректного функціонування системи і задоволення всіх учасників аукціону.

Найбільш складним і важливим алгоритмом системи є процес підтвердження покупки лоту (рис. 2.7). Цей алгоритм включає кілька ключових етапів і забезпечує, щоб кожна покупка була успішно завершена та підтверджена. Процес підтвердження покупки лоту починається з того моменту, коли користувач вирішує придбати лот, який він виграв на аукціоні. Розглянемо докладніше, як відбувається цей процес.

1. **Початок процесу покупки:** Після завершення аукціону і оголошення переможця, користувач, який виграв лот, отримує сповіщення на електронну пошту з посиланням для переходу до сторінки оформлення покупки. Це повідомлення містить основні деталі про виграний лот, такі як назва лоту, остаточна ціна, та інструкції щодо подальших дій.
2. **Перехід до сторінки оформлення:** Користувач переходить за посиланням в електронному листі на сторінку оформлення покупки на вебсайті. На цій сторінці користувачеві надається можливість переглянути деталі виграного лоту, остаточну ціну, а також інструкції щодо оплати.
3. **Спосіб оплати:** На сторінці підтвердження покупки користувач обирає спосіб оплати. Можливі варіанти включають кредитні картки та інші електронні методи оплати. Користувач вводить необхідні платіжні дані у відповідні поля.
4. **Обробка платежу:** Після введення платіжних даних система обробляє платіж за допомогою інтегрованих платіжних шлюзів. Цей процес включає перевірку платіжних даних та авторизацію платежу.

5. **Підтвердження оплати:** Якщо платіж успішно оброблений, система підтверджує покупку і відправляє користувачеві повідомлення на пошту про успішне завершення. Користувача також перекидає на сторінку підтвердження, що його покупка була успішно завершена.
6. **Оновлення статусу лоту:** Після успішного підтвердження оплати система автоматично оновлює статус лоту на «Закритий». Це забезпечує, що лот більше не буде відкритий для інших користувачів, і власник лоту отримує на свій баланс кошти за проданий лот.

Цей алгоритм забезпечує надійність і безпеку процесу підтвердження покупки. Процес включає кілька перевірок та підтверджень, щоб гарантувати, що всі сторони задоволені результатом.

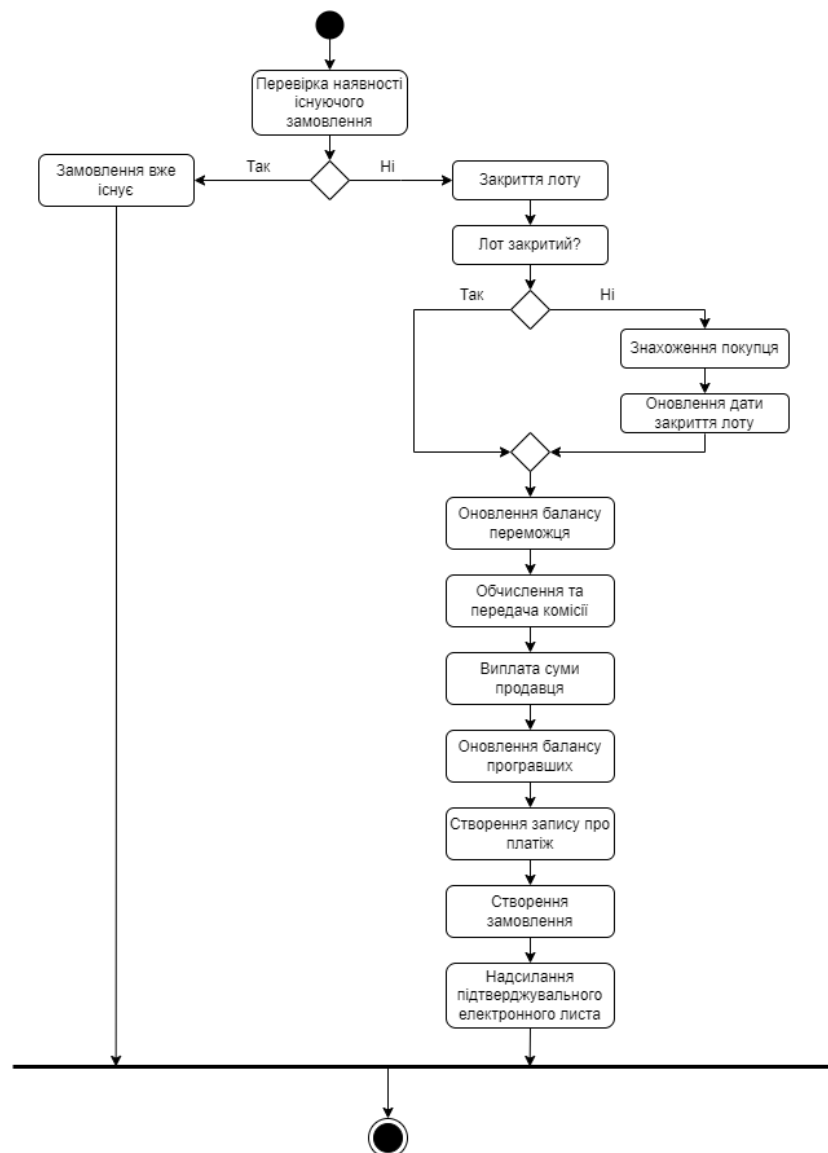


Рисунок 2.7 – Діаграма діяльності Підтвердження покупки лоту

2.4. Реалізація додатку веборієнтованої системи управління аукціону речей

Програмна система складається з двох основних компонентів: клієнтського додатка і серверної частини. Спочатку було розпочато створення основного інтерфейсу системи, з метою забезпечення інтуїтивно зрозумілої та зручної взаємодії користувачів із додатком. Проте, велика увага приділялася розробці серверної частини, яка включає в себе інтеграцію з базою даних, обробку запитів та логіку бізнес-процесів.

Для ефективного сповіщення користувачів про важливі події, такі як підтвердження реєстрації, перемога в аукціоні, завершення торгів та інші, було розроблено спеціальний сервіс для відправки електронних листів. Цей сервіс автоматизує процес відправлення повідомлень на електронну пошту, що забезпечує своєчасне інформування користувачів і підвищує рівень обслуговування та зворотного зв'язку в системі. Для цього було налаштовано сервіс:

```
async sendEmail(to, subject, text) {

  const transporter = nodemailer.createTransport({
    service: "gmail",
    auth: {
      user: process.env.EMAIL,
      pass: process.env.EMAIL_PASSWORD,
    },
    debug: true,
  });

  const mailOptions = {
    from: process.env.EMAIL,
    to,
    subject,
    html: htmlTemplate,
  };

  try {
    await transporter.sendMail(mailOptions);
  } catch (error) {
    console.log("Не вдалося надіслати електронного листа:", error);
  }
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Перш за все, було налаштовано генерацію пін-кодів для підтвердження особистості в системі. Це включало в себе вибірка випадкового набору чосел або чисел заданої довжини, який ставав унікальним ідентифікатором для кожного користувача або події. Після генерації пін-коду, він зберігався в базі даних разом з іншою відповідною інформацією такою як дата існування.

```
static async generatePinCode() {

    const pinCode = Math.floor(100000 + Math.random() * 900000).toString();
    const expiration = new Date();
    expiration.setHours(expiration.getHours() + 1);

    return { pinCode, expiration };

}
```

Далі, було реалізовано механізм відправки згенерованого пін-коду користувачеві поштою. Цей процес включав в себе взаємодію з електронною поштовою системою, де згенерований пін-код був відправлений на вказану користувачем електронну адресу.

```
static async sendPinCode(userId, email) {

    const user = await User.findByPk(userId);

    if (!user) {
        throw new Error("Користувача не знайдено.");
    }

    const pinCodeData = await this.generatePinCode();
    await this.savePinCode(userId, pinCodeData);

    await emailService.sendEmail(
        email,
        "Ваш пін-код",
        `Ваш пін-код: ${pinCodeData.pinCode}`
    );

    user.pinCodeSendAttempts += 1;
    await user.save();

}
```

Крім того, було реалізовано механізм перевірки пін-коду при підтвердженні певних операцій. Користувач вводив отриманий пін-код у відповідну форму на вебсайті, після чого система перевіряла його

відповідність збереженому значенню в базі даних. У разі співпадіння пін-коду, користувачеві надавалось дозвіл на виконання певної дії або операції.

```
static async checkPinCode(userId, enteredPinCode) {  
  
    const user = await User.findByPk(userId);  
    if (!user.pinCode) {  
        return false;  
    }  
    const isMatch = await bcrypt.compare(enteredPinCode, user.pinCode);  
    if (!isMatch) {  
        user.pinCodeAttempts += 1;  
        if (user.pinCodeAttempts >= 3) {  
            user.pinCode = null;  
            user.pinCodeExpiration = null;  
            user.pinCodeAttempts = 0;  
            await user.save();  
            return false;  
        }  
        await user.save();  
  
        return false;  
    }  
}
```

Зараз буде розглянуто більш детально процес закриття лоту в системі. Цей процес включає в себе кілька етапів, що забезпечують коректну завершеність лоту та розподіл переможців.

Для початку, після завершення терміну відкриття лоту або досягнення максимальної цінової пропозиції, система перевіряє статус лоту. Якщо лот вже закритий, процес автоматично завершується, щоб уникнути повторних дій.

```
if (  
    lot.status === "CLOSED" ||  
    new Date(lot.endDate).getTime() > new Date().getTime()  
) {  
    return;  
}  
  
lot.status = "CLOSED";
```

Після цього відбувається перевірка наявності ставок на лоті. Якщо ставки відсутні, лот закривається без визначення переможця, і власник лоту залишає його незмінним.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const bids = lot.Bids;
if (bids.length > 0) {
  const highestBid = bids.reduce(
    (max, bid) => (bid.amount > max.amount ? bid : max),
    bids[0]
  );
  lot.winnerId = highestBid.userId;
}
await lot.save();

```

У випадку наявності ставок, система визначає переможця аукціону. Для цього визначається найвища ставка серед усіх поданих.

Після визначення переможця, система відправляє йому повідомлення на електронну пошту, щоб повідомити про успішну перемогу. Також іншим учасникам аукціону, які не стали переможцями, надсилається повідомлення про їхню участь і результати. Це гарантує, що подальша участь у торгах для даного лоту буде припинена і жоден інший користувач не зможе робити нові ставки.

```

if (lot.winnerId) {

  const winner = await User.findById(lot.winnerId);
  await emailService.sendEmail(
    winner.email,
    "Вітаємо!",
    `Ви виграли лот: ${lot.title}. Перейдіть за посиланням для оформлення замовлення: http://localhost:3000/payment?lotId=${lot.id}.`
  );

}

```

Наступний крок – це оновлення балансів користувачів. Після закриття лоту, всім програвшим повертаються суми, які були використанні для ставок.

```

const losingBids = bids.filter((bid) => bid.userId !== lot.winnerId);
const userEmails = losingBids.map((bid) => bid.User.email);
if (userEmails.length) {

  await emailService.sendEmail(
    userEmails,
    "Дякуємо за участь.",
    `На жаль, ви не виграли лот: ${lot.title}. Ми цінуємо Вашу участь і сподіваємося, що Ви знайдете щось цікаве в наших наступних лотах.`
  );

}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Для здійснення покупок і оформлення платежів використовується платіжна система Stripe [29]. Перш ніж користувач зможе придбати лот на аукціоні, потрібно згенерувати сесію, яка дозволить здійснити платіж через Stripe.

За допомогою Stripe API створюється сесія платежу, яка пов'язується з конкретним лотом і користувачем. Ця сесія містить всю необхідну інформацію про покупку, таку як сума платежу, ідентифікатор лоту та інші деталі. Після успішного створення сесії, користувач перенаправляється на сторінку оплати, де він може ввести свої платіжні дані та завершити операцію.

Цей підхід дозволяє забезпечити безпеку та надійність платежів, а також зручність для користувачів, оскільки вони можуть швидко та безпечно здійснювати покупки прямо на платформі аукціону.

Цей процес також дозволяє використовувати різноманітні методи оплати, включаючи кредитні та інші популярні платіжні методи, що забезпечує максимальний комфорт для користувачів і робить процес покупок на платформі аукціону максимально доступним і зручним.

```
async createCheckoutSession(  
    amount,  
    user,  
    description,  
    successUrl,  
    cancelUrl  
) {  
    const session = await stripe.checkout.sessions.create({  
        payment_method_types: ["card"],  
        shipping_address_collection: {  
            allowed_countries: ["UA"],  
        },  
        line_items: [  
            {  
                price_data: {  
                    currency: "uah",  
                    product_data: {  
                        name: description,  
                    },  
                    unit_amount: amount * 100,  
                },  
            },  
        ],  
    });  
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        quantity: 1,
    },
],
billing_address_collection: "auto",
mode: "payment",
customer_email: user.email,
success_url: successUrl,
cancel_url: cancelUrl,
});

return session;
}

```

Висновки до другого розділу

У розділі дослідження було проведено детальний аналіз функціональних та нефункціональних вимог до програмного комплексу аукціону речей. Створено діаграми використання, визначено основні можливості системи та взаємодію користувачів з нею. На основі цих діаграм створені ролі користувачів та встановлено їх функціональні обов'язки.

У другому підрозділі було докладно проаналізовано структуру бази даних програмного комплексу. Надано опис до кожної таблиці, її зв'язки з іншими елементами системи. Це дозволило краще зрозуміти організацію даних у системі та їхній взаємозв'язок.

У третьому підрозділі розглянуто найтяжчі алгоритми роботи системи, такі як авторизація, закриття лоту та підтвердження покупки. Шляхом детального аналізу та опису цих алгоритмів, було з'ясовано, як вони впливають на функціонування системи та взаємодію з користувачами.

У четвертому підрозділі показано важливі моменти в реалізації програмного коду, описавши основні алгоритми та використані технології. Це дозволило краще зрозуміти роботу системи з точки зору програмування та підвищити її ефективність та надійність.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБДОДАТКОМ АУКЦІОНУ РЕЧЕЙ

3.1. Порядок встановлення та налаштування параметрів системи

Для повноцінного розгортання системи необхідно належним чином встановити та налаштувати всі її компоненти. Система складається з трьох основних частин, кожна з яких відповідає за власний функціонал та забезпечує певний аспект її роботи. На клієнтській стороні використовується технологія React [15] у поєднанні з бібліотекою Material-UI [19], що дозволяє створювати ефективні та зручні для взаємодії інтерфейси. Серверна частина системи реалізована на базі технологій Node.js [12] та Express.js [13], що забезпечує високу продуктивність та надійність обробки запитів. База даних спирається на реляційну систему керування базами даних PostgreSQL [27], що гарантує ефективне та оптимальне зберігання і обробку даних. Такий розподіл компонентів (рис. 3.1) дозволяє забезпечити якісну та надійну роботу системи, забезпечуючи оптимальну продуктивність та зручність користування для кінцевих користувачів.

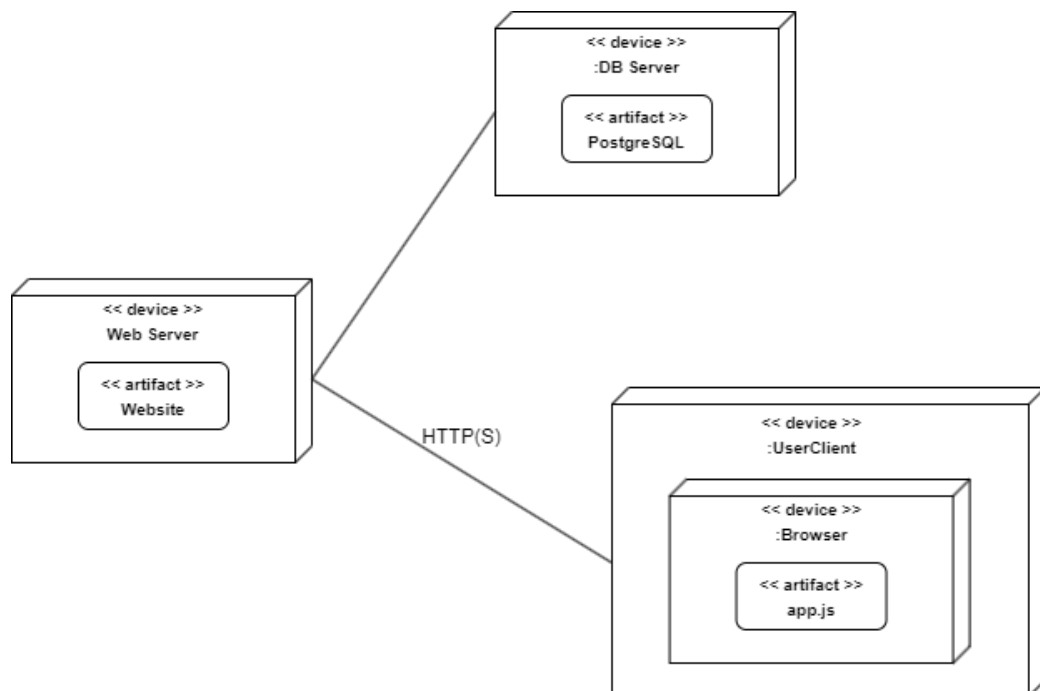


Рисунок 3.1 – Діаграма розгортання системи

Перед початком роботи важливо підготувати робоче оточення для запуску компонентів системи. На робочій машині повинні бути встановлені останні версії наступних компонентів і інструментів:

- Node.js (версія 20.14.0) для виконання JavaScript на сервері.
- Express.js (версія 4.19.2) – вебфреймворк для створення вебдодатків на Node.js.
- Redux – бібліотека керування станом для JavaScript додатків.
- React (версія 18.2.0) – бібліотека для створення інтерфейсів користувача.
- PostgreSQL (версія 16.3) – реляційна система керування базами даних для зберігання та обробки даних.

Для конфігурації параметрів запуску серверного додатку потрібне підключення до бази даних та обробки запиту від клієнтського додатку. Це вимагає внесення налаштувань у конфігураційні файли з назвами db.js та config.json.

db.js

```
const { Sequelize } = require("sequelize");

module.exports = new Sequelize(
  process.env.DB_NAME,
  process.env.DB_USER,
  process.env.DB_PASSWORD,
  {
    dialect: "postgres",
    host: process.env.DB_HOST,
    port: process.env.DB_PORT,
  }
);
```

config.json

```
"development": {
  "username": "postgres",
  "password": "12345",
  "database": "auction_things",
  "host": "localhost",
  "dialect": "postgres",
  "port": "5432"
},
```

					ІПЗ.КР.Б – 121 – 24 – ІЗ	Арк.
						51
Змн.	Арк.	№ док.ум.	Підпис	Дата		

3.2. Структура інтерфейс вебдодату аукціону речей

Інтерфейс аукціону речей є критичним компонентом програмного продукту, що забезпечує користувачам зручний доступ до всіх функцій та можливостей системи. У цьому розділі буде розглянуто основні елементи структури інтерфейсу системи, їх функціональність та взаємодію між ними.

Перш за все, буде розглянуто головну сторінку (рис. 3.2), так як це стартова точка для всіх користувачів системи. Вона включає такі елементи:

- **Категорії:** панель, що містить основні категорії лотів, такі як "Антикваріат", "Дитячі іграшки", "Образотворче мистецтво" та інші. Кожна категорія має короткий опис.
- **Рекомендовані лоти:** секція з найбільш популярними або новими лотами.

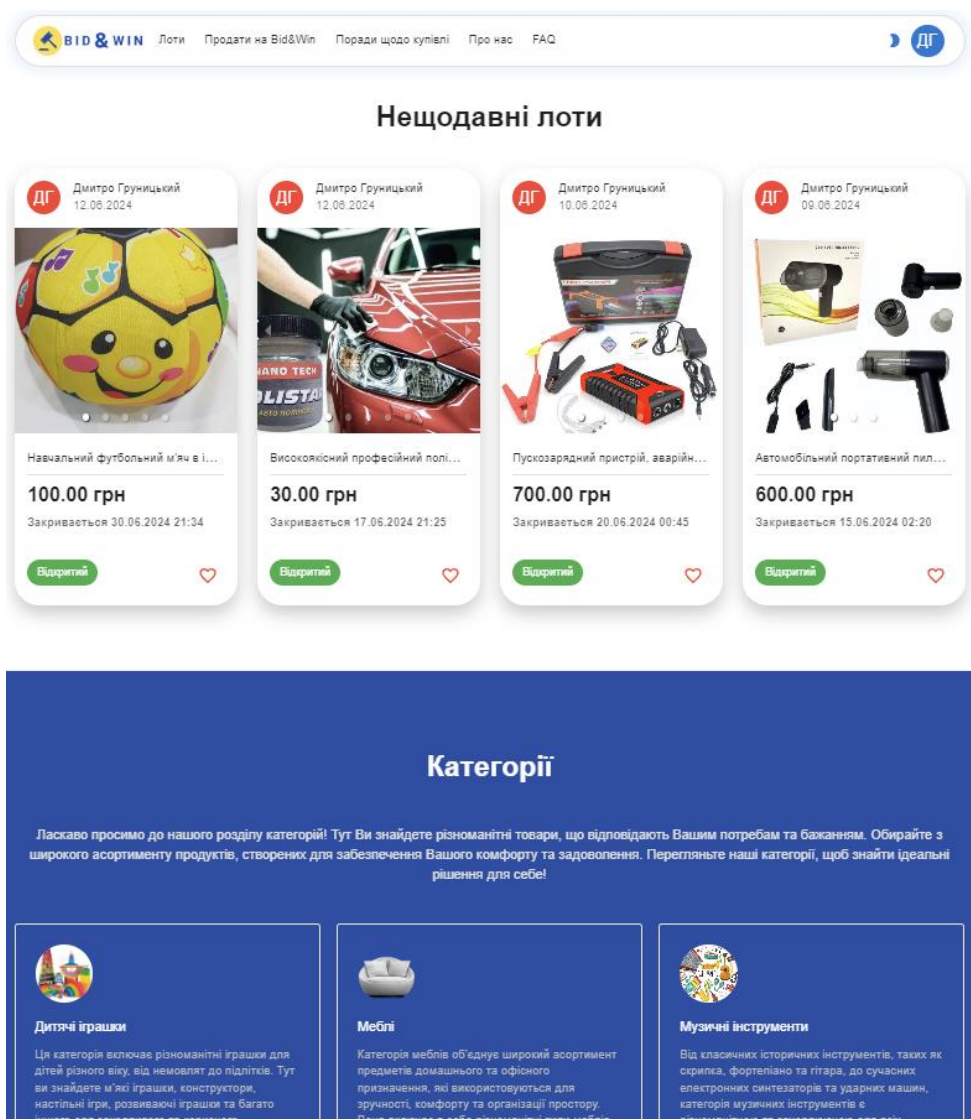


Рисунок 3.2 – Головна стрінка

Авторизація є важливим елементом функціональності аукціону речей, оскільки вона забезпечує безпеку та персоналізований досвід для користувачів. Процес авторизації дозволяє системі ідентифікувати користувача, надаючи доступ до персоналізованих налаштувань, історії покупок, балансу та інших індивідуальних даних. Це дозволяє користувачам брати участь в аукціонах, робити ставки та виставляти свої лоти.

На головній сторінці аукціону речей розміщена кнопка "Увійти", яка є видимою та легко доступною для всіх відвідувачів сайту. Натиснувши на цю кнопку, користувач потрапляє на окрему сторінку авторизації (рис. 3.3).



Рисунок 3.3 – Сторінка авторизації

Після введення коректних облікових даних користувач отримує електронний лист з одноразовим пін-кодом для підтвердження своєї особи (рис. 3.4). Цей пін-код слугує додатковим рівнем безпеки, запобігаючи несанкціонованому доступу до облікового запису користувача.

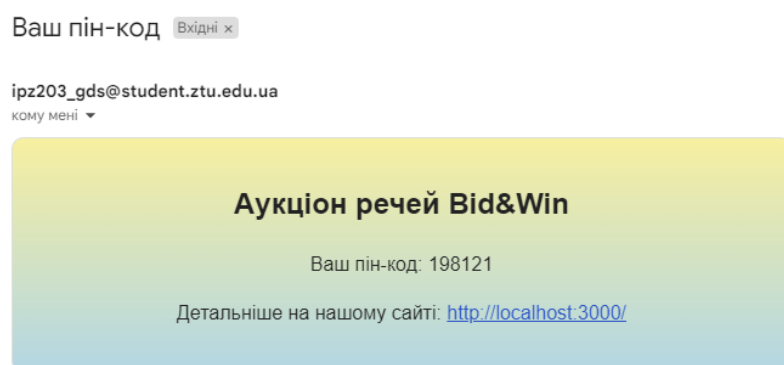


Рисунок 3.4 – Електронний лист з одноразовим пін-кодом

Після натискання на кнопку "Увійти" користувача автоматично перенаправляють на сторінку введення пін-коду (рис. 3.5). На цій сторінці користувач бачить поле введення пін-коду. Він повинен ввести отриманий код у це поле. Після введення пін-коду та натискання кнопки "Продовжити" код надсилається на сервер для перевірки, що забезпечує додатковий рівень безпеки перед наданням доступу до облікового запису та дає доступ до облікового запису.

ВАМ БУЛО НАДІСЛАНО ПІН-КОД НА ЕЛЕКТРОННУ АДРЕСУ.

Введіть пін-код для пошти:

victor.pavitsky@gmail.com

 |

ПРОДОВЖИТИ

Не отримали пін-код [Повторна відправка](#)

Рисунок 3.5 – Сторінка для введення пін-коду

Після натискання кнопки "Зареєструватися", користувача автоматично перенаправляє на сторінку вибору ролі в системі (рис. 3.6). На цій сторінці користувач бачить список доступних ролей, таких як "Покупець" або "Продавець".

Виберіть роль

Виберіть, ким ви хочете бути на нашому аукціоні речей: покупцем або продавцем.

☐ Покупець

☐ Продавець

ПРОДОВЖИТИ

Рисунок 3.6 – Сторінка вибору ролі в системі

Він вибирає відповідну роль і переходить до наступного етапу реєстрації, в якому він буде запрошений ввести додаткові дані, необхідні для

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

створення облікового запису обраної ролі (рис. 3.7). Цей процес реєстрації забезпечує зручний і персоналізований досвід для кожного користувача, дозволяючи йому налаштувати свій обліковий запис відповідно до своїх потреб і вимог.

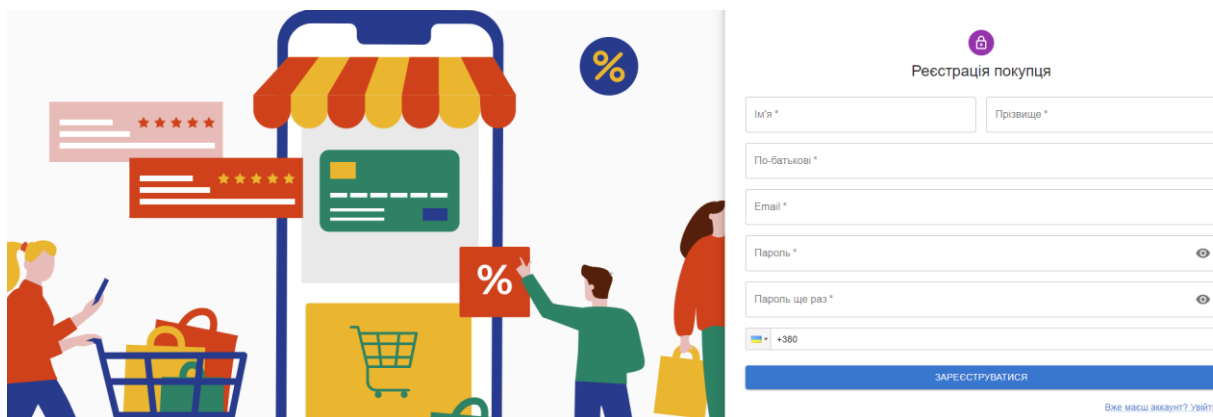


Рисунок 3.7 – Сторінка реєстрації для продавця

Після вибору ролі в системі та введення додаткових реєстраційних даних користувача, відбудеться автоматичний перехід на сторінку введення пін-коду, аналогічно до того, як це було показано раніше під час процесу авторизації. На цій сторінці користувач отримає можливість ввести пін-код, який був надісланий на його електронну адресу для підтвердження реєстрації облікового запису. Після успішного введення пін-коду відбувається створення облікового запису з обраною роллю та можливістю використовувати всі функції системи відповідно до своїх потреб і прав доступу.

На аукціоні речей доступна окрема сторінка, присвячена лотам (рис. 3.8), яка дозволяє користувачам переглядати інформацію про доступні лоти та брати в них участь. Для переходу на цю сторінку, користувачу потрібно натиснути на кнопку "Лоти", яка розміщена на основному меню. Після натискання на цю кнопку користувач автоматично перенаправляється на сторінку зі списком доступних лотів, де він може переглядати інформацію про кожен лот, таку як заголовок, опис, початкова ціна, поточна ціна, стан аукціону та інші важливі дані. Крім того, на цій сторінці користувач може здійснювати пошук та фільтрацію лотів за різними параметрами, щоб знайти ті, які відповідають його потребам та інтересам.

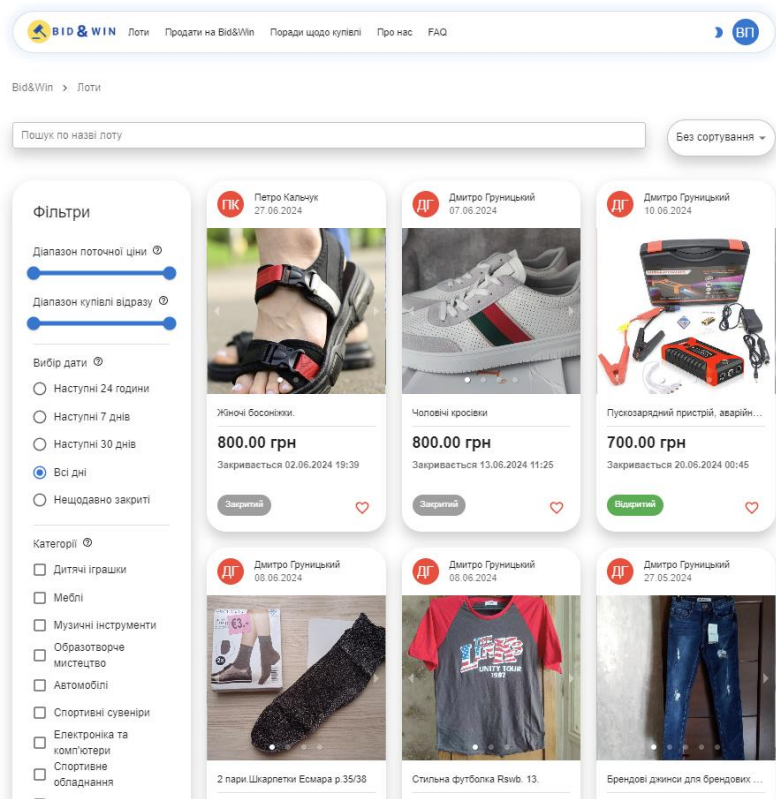


Рисунок 3.8 – Сторінка лотів

Користувач може перейти на сторінку будь-якого лоту, для цього йому потрібно навести курсор миші на назву лоту, після чого натиснути на неї. Після натискання на назву лоту користувач автоматично перенаправляється на окрему сторінку цього конкретного лоту (рис. 3.9). На цій сторінці користувач може детально ознайомитися з інформацією про обраний лот.

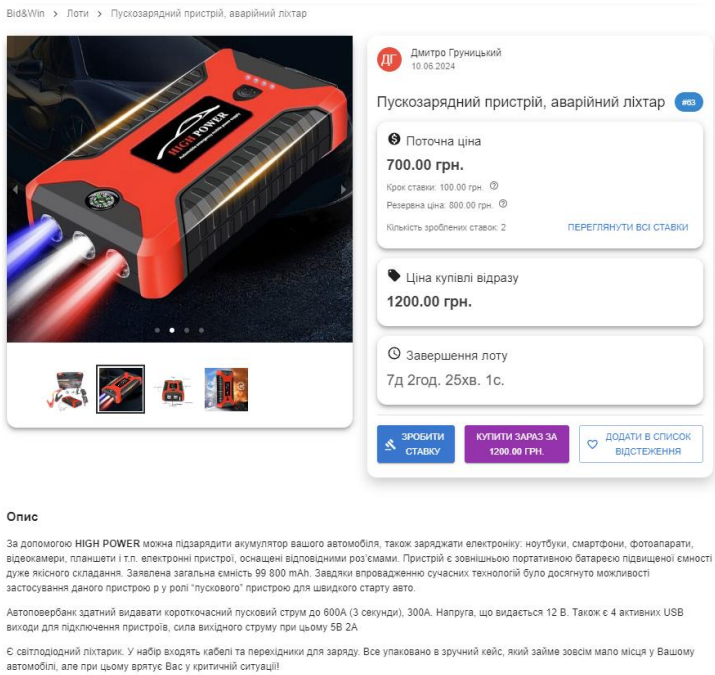


Рисунок 3.9 – Сторінка про лот

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

На сторінці лоту, користувач має можливість додати даний лот до списку відстеження (рис. 3.10). Для цього він може скористатися кнопкою "Додати в список відстеження", яка розташована поруч з іншими діями. Після додавання лоту до списку відстеження, користувач може легко переглянути, просто перейшовши в основне меню в розділ "Список відстеження". Ця функція дозволяє користувачеві швидко знаходити та відстежувати лоти, які його зацікавили, і в будь-який момент переглянути їх, не шукаючи серед інших.

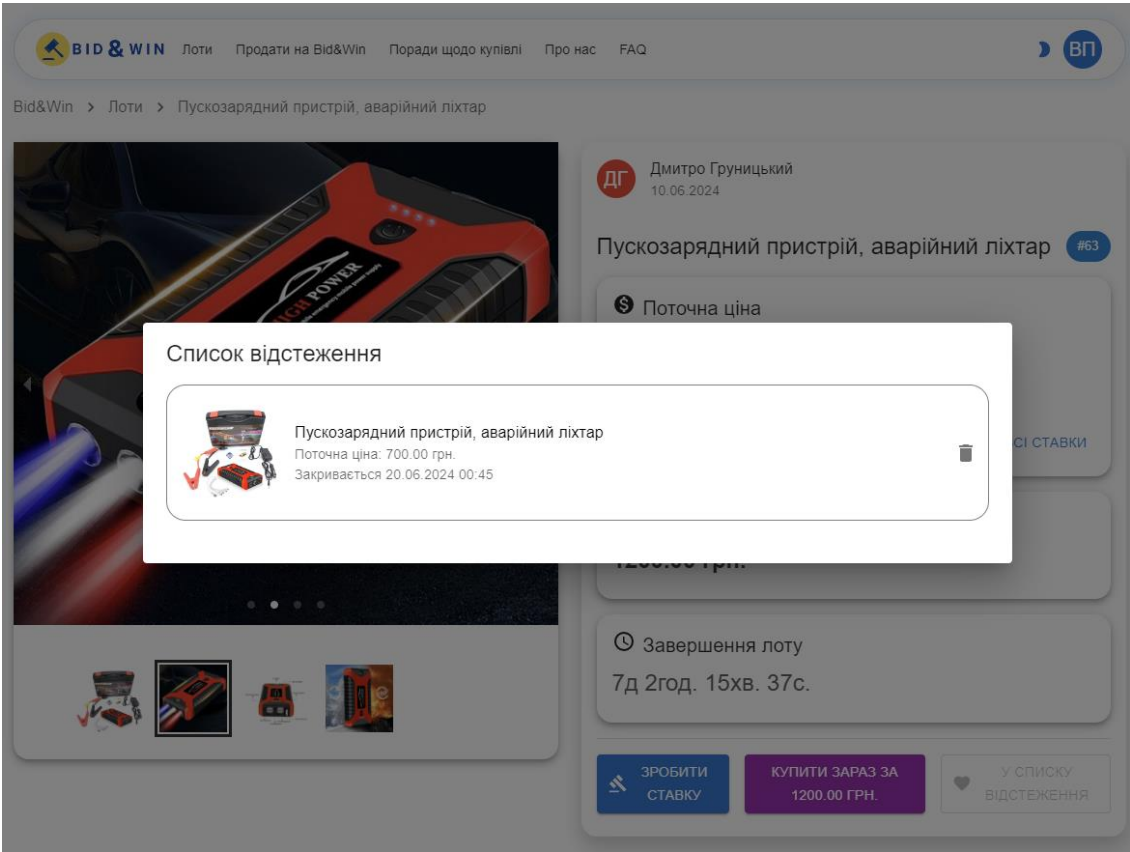


Рисунок 3.10 – Доданий лот до списку відстеження

Користувач має можливість переглянути всі зроблені ставки (рис. 3.11). Для цього йому потрібно натиснути на кнопку "Переглянути всі ставки", яка розміщена в інформації про лот. Після натискання на цю кнопку, відкривається модальне вікно, в якому користувач може переглянути всі зроблені ставки на даний лот. Там можна переглянути деталі кожної ставки, такі як сума ставки, лот, хто зробив ставку, дата та час розміщення ставки та інші важливі дані. Це дає користувачеві можливість легко відстежувати всі активні ставки.

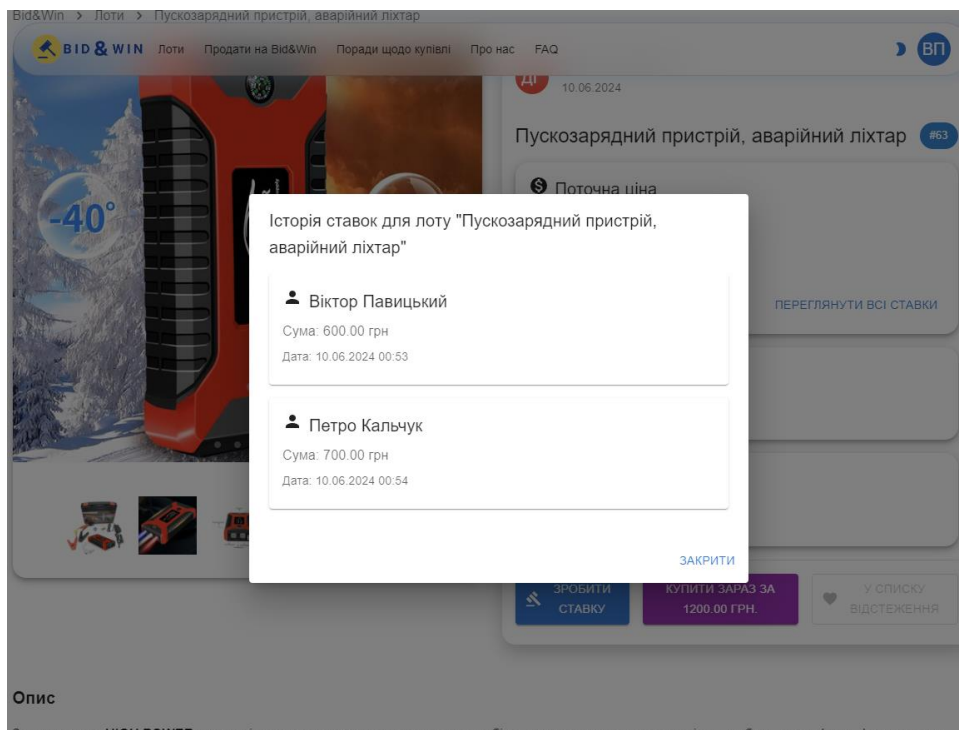


Рисунок 3.11 – Перегляд всіх зроблених ставок

Користувач може зробити ставку, якщо він авторизований і має достатню суму на балансі, щоб перевищити поточну ціну лоту. Для цього йому потрібно натиснути на кнопку "Зробити ставку", яка розташована поруч з іншими діями. Після натискання на цю кнопку користувачу відображається модальне вікно, в якому він може ввести суму своєї ставки (рис. 3.12). Після введення суми користувач натискає кнопку "Зробити ставку", і його ставка надсилається на сервер для обробки.

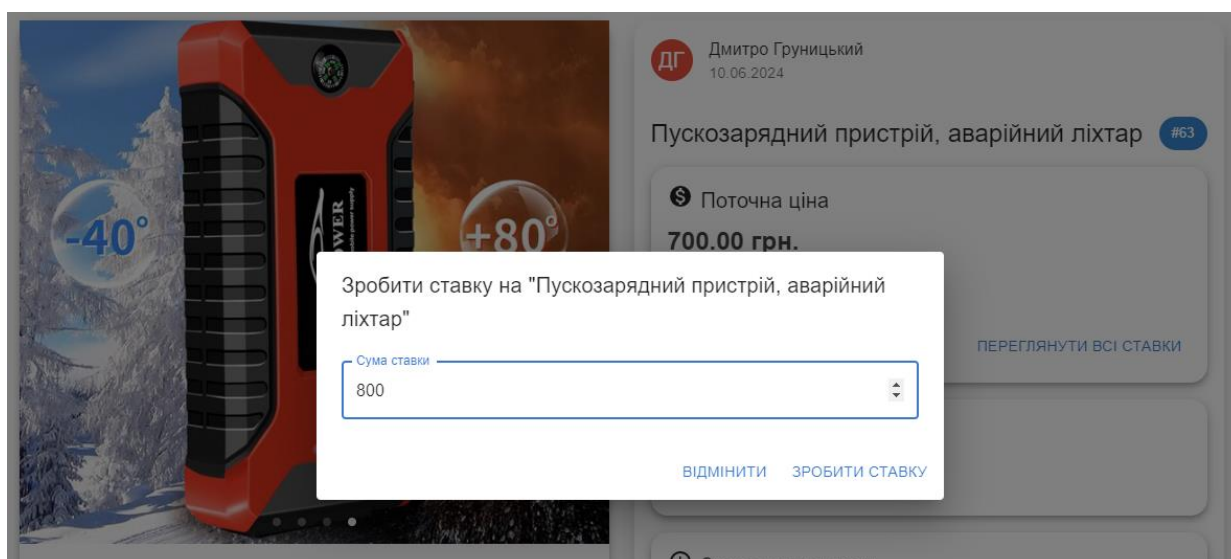


Рисунок 3.12 – Можливість зробити ставку

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо сума ставки перевищує поточну ціну лоту і користувач має достатню суму на балансі, ставка буде успішно зареєстрована (рис. 3.13), і нова ціна лоту буде встановлена як введена користувачем сума (рис. 3.13). Після цього ставка з'явиться в списку ставок користувача, і він зможе відстежувати її стан.

У випадку, якщо сума ставки не відповідає умовам (наприклад, менша або рівна поточній ціні лоту або недостатньо коштів на балансі), користувач отримає відповідне повідомлення про помилку і не зможе зробити ставку.

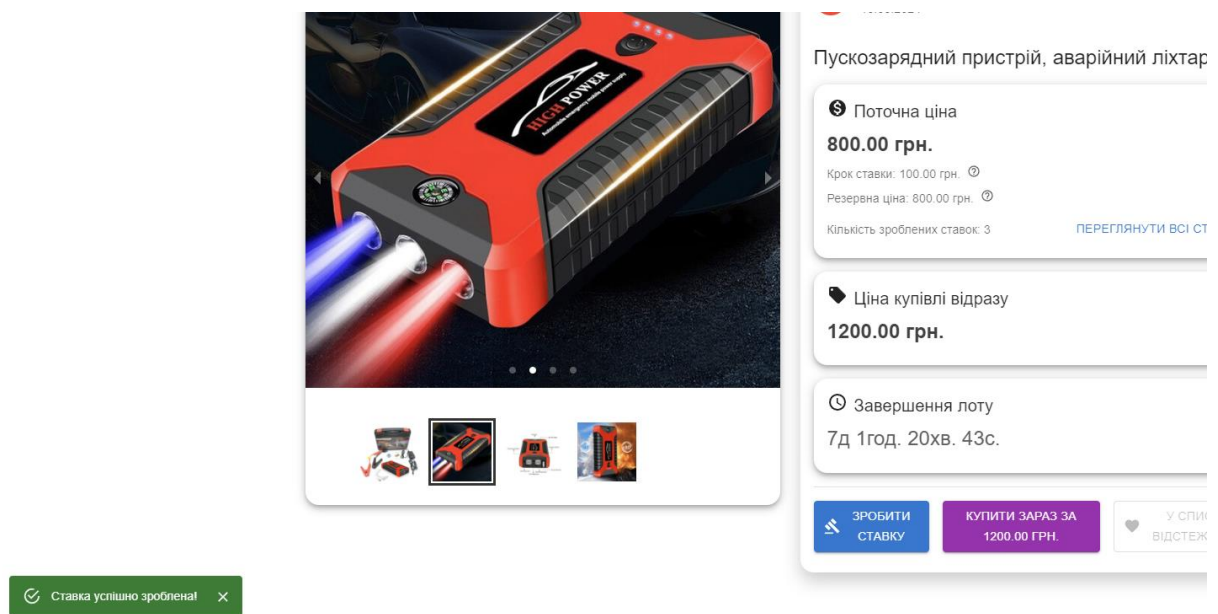


Рисунок 3.13 – Підтвердження успішної ставки

Після завершення лоту, система автоматично відправляє повідомлення учасникам аукціону, повідомляючи про завершення торгів з цим конкретним лотом (рис. 3.14). Це повідомлення надсилається до електронних адрес, які були вказані учасниками під час реєстрації на аукціон або під час зроблення ставки на даний лот. Учасники отримують сповіщення на свої електронні адреси та переконуються, що вони не пропустять важливу інформацію щодо завершення торгів (рис. 3.15 – 3.16). Це сповіщення від системи є важливою частиною процесу, оскільки воно дозволяє учасникам вчасно отримати інформацію про результати торгів, дізнатися чи виграли лот, чи, навпаки, програли та подальші дії, які можуть бути необхідними для завершення угоди.



ДГ Дмитро Груницький
10.06.2024

Пускозарядний пристрій,

Поточна ціна

800.00 грн.

Крок ставки: 100.00 грн.

Резервна ціна: 800.00 грн.

Кількість зроблених ставок: 3

Ціна купівлі відразу

1200.00 грн.

Завершення лоту

Лот завершено.

Лот завершився. Листи з результатами розіслані на вашу поштову скриньку. Будь ласка, перевірте свою електронну пошту для отримання додаткової інформації.

Рисунок 3.14 – Завершення лоту

Вітаємо! Вхідні x

ipz203_gds@student.ztu.edu.ua

кому мені ▼

Аукціон речей Bid&Win

Ви виграли лот: Пускозарядний пристрій, аварійний ліхтар. Перейдіть за посиланням для оформлення замовлення: <http://localhost:3000/payment?lotId=63>.

Детальніше на нашому сайті: <http://localhost:3000/>

Рисунок 3.15 – Результат завершення лоту (переможець)

Дякуємо за участь. Вхідні x

ipz203_gds@student.ztu.edu.ua

кому мені ▼

Аукціон речей Bid&Win

На жаль, ви не виграли лот: Пускозарядний пристрій, аварійний ліхтар. Ми цінуємо Вашу участь і сподіваємося, що Ви знайдете щось цікаве в наших наступних лотах.

Детальніше на нашому сайті: <http://localhost:3000/>

Рисунок 3.16 – Результат завершення лоту (той, хто програв)

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Після отримання повідомлення про перемогу на аукціоні, переможець може перейти за посиланням, яке надається у повідомленні, для оформлення покупки. Після переходу за цим посиланням переможець потрапляє на сторінку, де йому пропонується ввести всі необхідні дані для оформлення покупки (рис. 3.17). На цій сторінці переможець матиме можливість ввести свої персональні дані, такі як ім'я, прізвище, адреса та інші контактні дані.

Цей процес дозволяє переможцю швидко та зручно оформити покупку, надаючи всю необхідну інформацію для виконання доставки та оплати. Після введення всіх даних переможець може підтвердити своє замовлення, і тоді його покупка буде оформлена.

Рисунок 3.17 – Сторінка оформлення замовлення

Після успішного підтвердження оформлення покупки переможець буде перенаправлений на сторінку результату (рис. 3.18). На цій сторінці він побачить повідомлення, в якому буде виражена вдячність за покупку та

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

підтвердження успішного оформлення замовлення. Після цього користувач може переглянути додаткові деталі про своє замовлення.

Додатково, на електронну пошту під час реєстрації, буде відправлено повідомлення з підтвердженням замовлення (рис. 3.19). У цьому повідомленні буде вказано номер замовлення, що дозволить користувачу легко ідентифікувати своє замовлення. Після завершення перегляду результатів, користувач може повернутися на головну сторінку аукціону, використовуючи відповідну кнопку для цього.

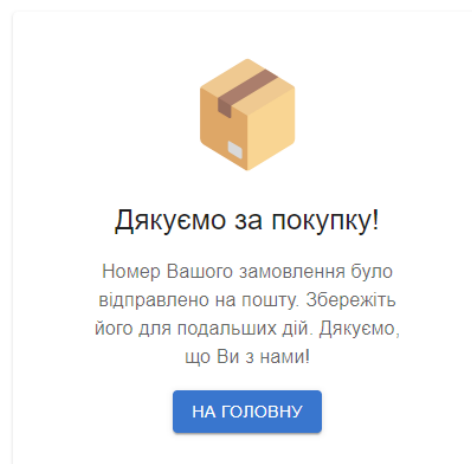


Рисунок 3.18 – Сторінка результату покупки

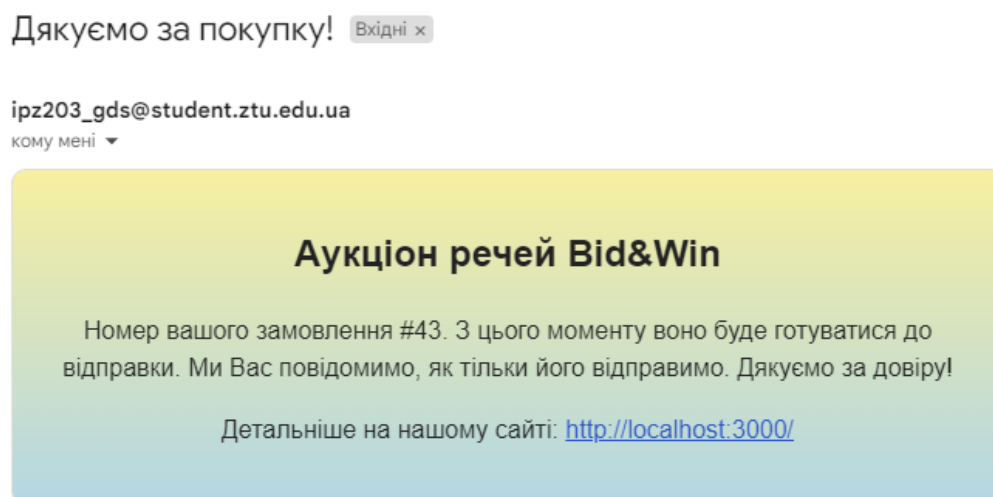


Рисунок 3.19 – Повідомлення з підтвердженням замовлення

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

Якщо користувач хоче купити відразу лот без участі в аукціоні, його перенаправляє на сторінку оформлення замовлення. На цій сторінці він може ввести необхідні дані для оформлення покупки, такі як ім'я, прізвище, адреса та інші контактні дані. Після введення всіх необхідних даних він може підтвердити своє замовлення, і тоді його покупка буде оформлена. Так само, як і в разі переможця аукціону, після успішного оформлення покупки, користувач буде перенаправлений на сторінку результату, де він побачить повідомлення про успішне оформлення замовлення та інші деталі. Після завершення перегляду результатів, користувач може повернутися на головну сторінку аукціону, використовуючи відповідну кнопку для цього.

У кожного користувача на аукціоні є особистий баланс, на якому відображена сума коштів, доступних для використання. Цей баланс можна поповнити або зняти кошти залежно від потреб користувача. Для зручності, інформація про баланс доступна в випадаючому меню при наведенні на профіль (рис. 3.20).

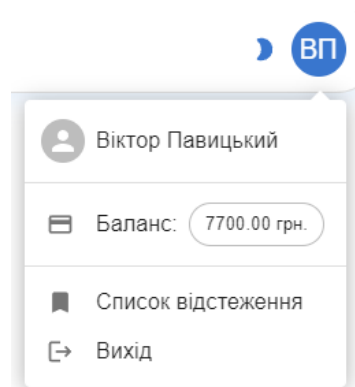


Рисунок 3.20 – Перегляд балансу

Якщо користувачу потрібно поповнити або вивести кошти з свого балансу, він може натиснути на відображену суму балансу в відкриваючому списку. Після цього він буде перенаправлений на сторінку, де буде доступний вибір дії, яку він хоче здійснити щодо свого балансу (рис. 3.21). На цій сторінці користувач може вибрати одну з наступних дій: поповнення балансу або виведення коштів. Після вибору відповідної опції користувачу буде запропоновано заповнити необхідні дані та виконати відповідні дії для завершення операції.

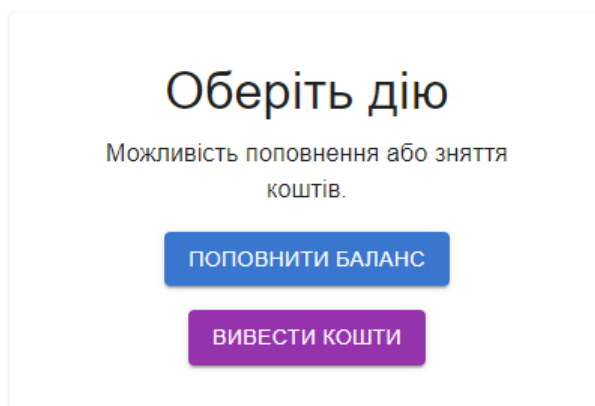


Рисунок 3.21 – Обрання дії

Якщо користувач бажає поповнити або вивести кошти з свого балансу, процес виконання цих дій майже однаковий. Спочатку користувач вводить суму, яку він бажає поповнити або вивести, у відповідне поле на сторінці керування балансом. Після цього він надає необхідні дані про свою платіжну картку (рис. 3.22 – 3.23).

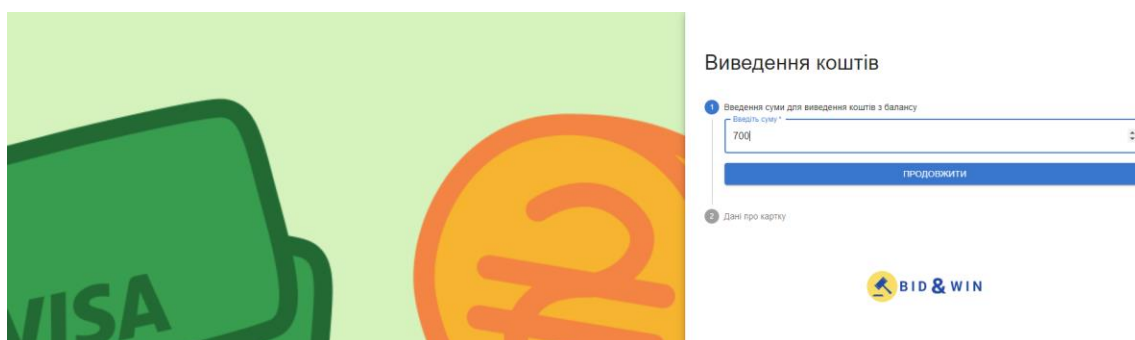


Рисунок 3.22 – Введення суми

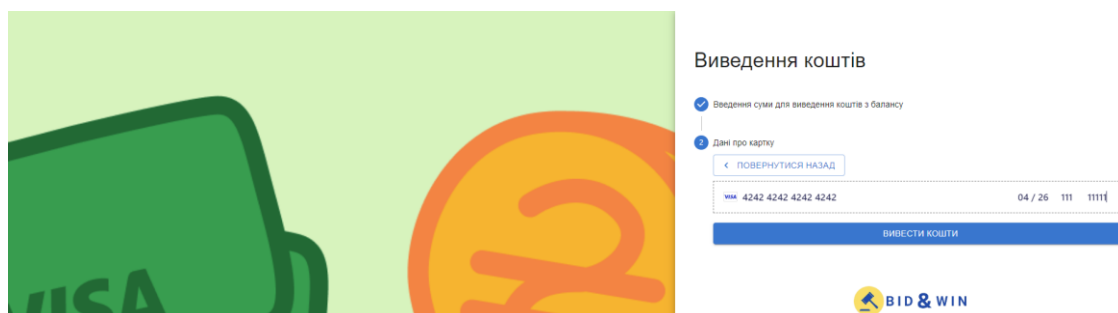


Рисунок 3.23 – Введення даних про картку

Після того, як користувач виконає операцію поповнення або виведення коштів, він побачить повідомлення від системи як результат своєї дії. Це повідомлення може містити інформацію про успішне або невдале виконання операції, а також інші деталі щодо статусу його балансу (рис. 3.24).

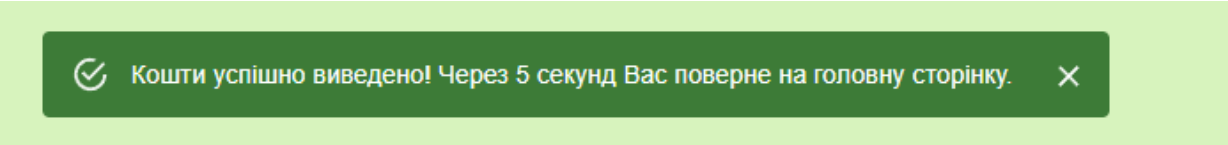


Рисунок 3.24 – Повідомлення від системи

У системі присутня адмін-панель, яка надає адміністратору розширені можливості управління та моніторингу аукціону (рис. 3.25). Через адмін-панель адміністратор може переглядати статистику, аналізувати дані та керувати різними аспектами системи.

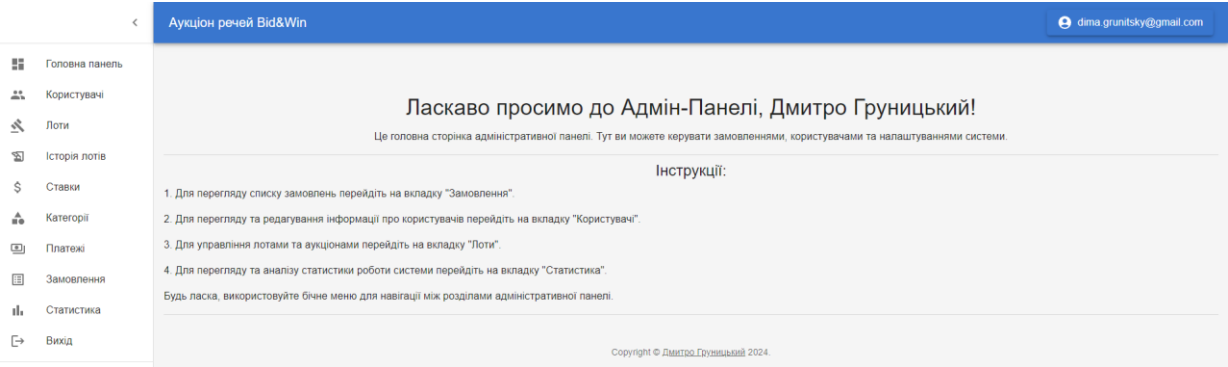


Рисунок 3.25 – Головна сторінка Адмін-панелі

У адмін-панелі доступні такі можливості:

- 1. **Перегляд статистики:** адміністратор може переглядати різні статистичні дані, такі як кількість зареєстрованих користувачів, кількість завершених лотів та інші показники (рис. 3.26).

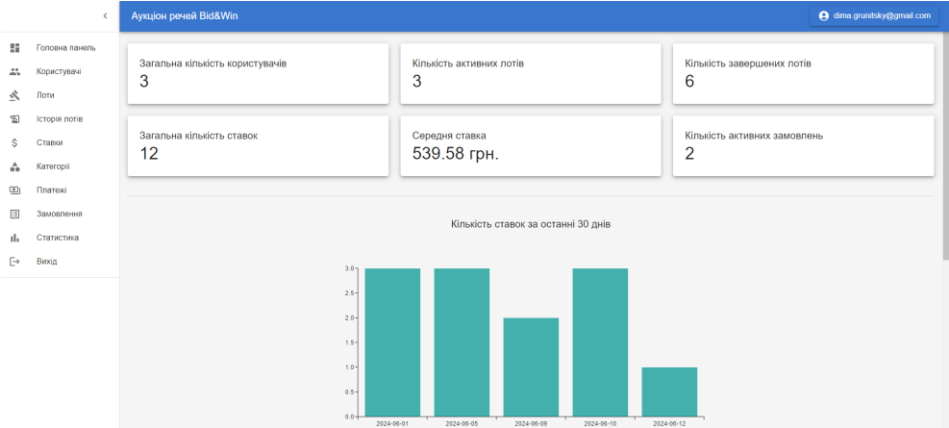


Рисунок 3.26 – Статистика в Адмін-панелі

2. **Керування користувачами:** адміністратор може переглядати список користувачів, блокувати або видаляти облікові записи та виконувати інші дії з акаунтами (рис. 3.27).

ID	Ім'я	Прізвище	По-батькові	Електронна адреса	Роль в системі	Номер телефону	Дії
1	Дмитро	Груничай	Сергійович	dima.grunitsky@gmail.com	ADMIN	+380 98 765 4321	[View] [Edit] [Delete]
78	Петро	Калычук	Віталійович	dmytro.hnulytskyi@gmail.com	SELLER	+380 98 764 7333	[View] [Edit] [Delete]
79	Віктор	Павлицький	Іванович	victor.pavitsky@gmail.com	BUYER	+380 98 765 4123	[View] [Edit] [Delete]

Рисунок 3.27 – Керування користувачами в Адмін-панелі

3. **Керування лотами:** адміністратор може переглядати список лотів, редагувати їх параметри, видаляти лоти або додавати нові (рис. 3.28).

№	Початкова ціна	Дата закінчення	Статус	Ціна	Підвищення ставки	Резервна ціна	Фотографії	Дії
Лонжми жіночі в спортивному ...	120.00 грн.	02.06.2024 19:39	CLOSED	700.00 грн.	50.00 грн.	700.00 грн.	[Photos]	[View] [Edit] [ВІДКРИТИ]
Ври Шкарпетки Есмара р 35/...	60.00 грн.	09.06.2024 00:15	CLOSED	80.00 грн.	5.00 грн.	70.00 грн.	[Photos]	[View] [Edit] [ВІДКРИТИ]
м секонд-хенд Розмір S, M Б...	30.00 грн.	08.06.2024 18:06	CLOSED	70.00 грн.	5.00 грн.	35.00 грн.	[Photos]	[View] [Edit] [ВІДКРИТИ]
Стан новий Розмір 41, 42, 4...	100.00 грн.	13.06.2024 11:25	CLOSED	650.00 грн.	100.00 грн.	600.00 грн.	[Photos]	[View] [Edit] [ВІДКРИТИ]
За допомогою «strong» HIGH...	500.00 грн.	12.06.2024 23:58	CLOSED	1200.00 грн.	100.00 грн.	800.00 грн.	[Photos]	[View] [Edit] [ВІДКРИТИ]
«i» «strong» Автомобільний л...	200.00 грн.	15.06.2024 02:20	OPEN	500.00 грн.	100.00 грн.	500.00 грн.	[Photos]	[View] [Edit] [ЗАКРИТИ]
«i» Стан відмінний. Усе прав...	100.00 грн.	30.06.2024 21:34	OPEN	500.00 грн.	50.00 грн.	450.00 грн.	[Photos]	[View] [Edit] [ЗАКРИТИ]
«strong» Брендів нові котон...	550.00 грн.	11.06.2024 21:50	CLOSED	870.00 грн.	100.00 грн.	700.00 грн.	[Photos]	[View] [Edit] [ВІДКРИТИ]
«i» Стан нове «i» «strong» Вис...	30.00 грн.	17.06.2024 21:25	OPEN	400.00 грн.	30.00 грн.	350.00 грн.	[Photos]	[View] [Edit] [ЗАКРИТИ]

Рисунок 3.28 – Керування лотами в Адмін-панелі

4. **Інші функції:** до інших можливостей адмін-панелі входять керування категоріями, перегляд замовлень, платежів та інші адміністративні операції.

Користувач має можливість переглянути та редагувати свій особистий профіль в системі (рис. 3.29). На своєму профілі він може переглянути основну інформацію про себе, таку як ім'я, прізвище, адресу електронної пошти та інші контактні дані. У разі потреби користувач може відредагувати свою особисту інформацію, внесену у свій профіль. Наприклад, він може змінити своє ім'я, адресу електронної пошти, номер телефону або навіть пароль. Це дозволяє користувачеві оновлювати свої контактні дані та іншу інформацію для забезпечення актуальності та правильності даних у системі.

Ім'я:

Віктор

Прізвище:

Павицький

По-батькові:

Іванович

Email:

victor.pavitsky@gmail.com

✉

Номер телефону:

380987654123

Роль:

Почепець

Дата реєстрації:

28.05.2024

✎

🔒

Рисунок 3.29 – Профіль користувача

Додаток є адаптивним, що означає, що він оптимізований для використання на різних пристроях і розмірах екранів (рис. 3.30 – 3.31). Незалежно від того, де він відкривається. Можливість користуватися зручним інтерфейсом. Адаптивність дозволяє отримати однаково зручний досвід використання незалежно від пристрою, що використовується.

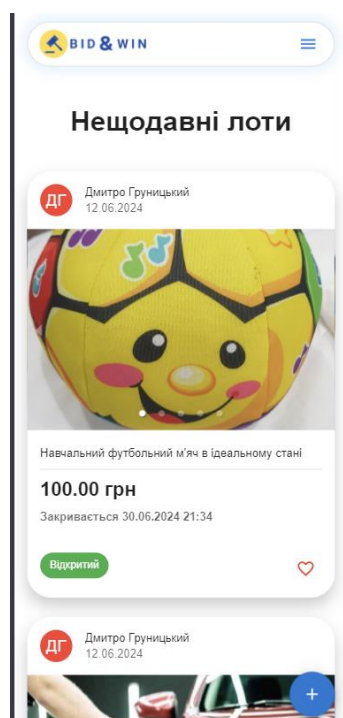


Рисунок 3.30 – Адаптивність (1)



Рисунок 3.31 – Адаптивність (2)

У додатку також присутня сторінка продавця, де продавець може керувати своїми лотами (рис. 3.32).

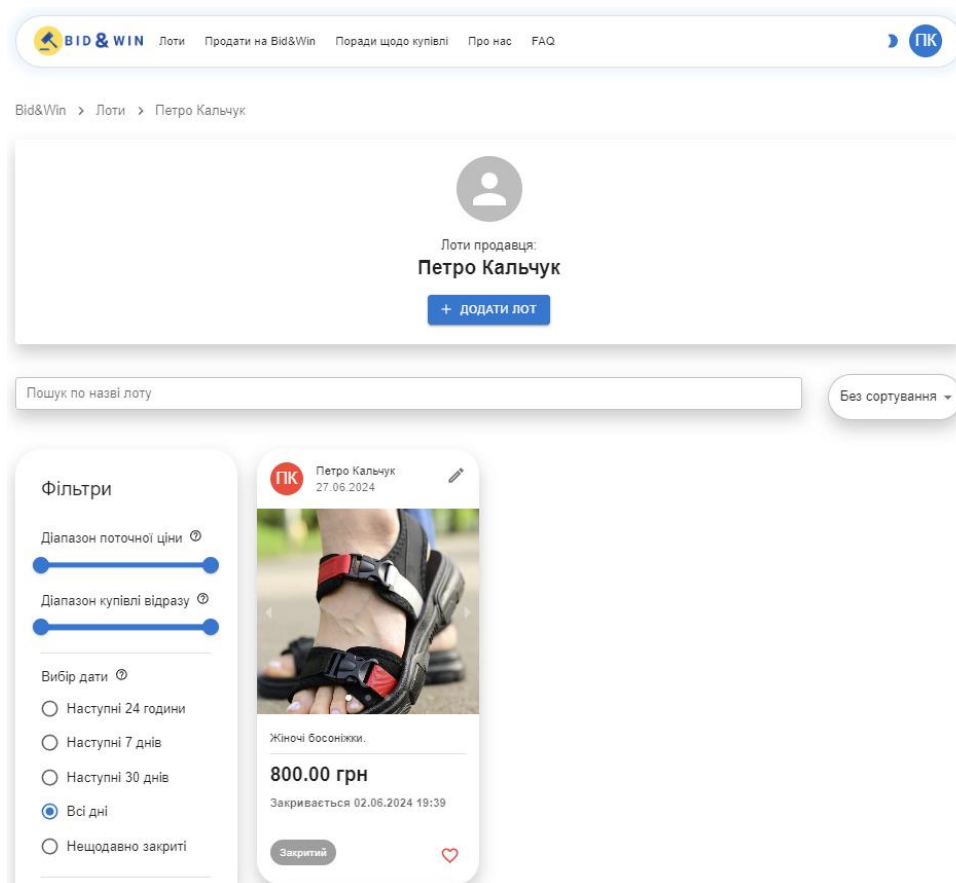


Рисунок 3.32 – Сторінка лотів продавця

На цій сторінці він може переглянути список своїх лотів, а також мати доступ до можливості додавання нових лотів або відредагування існуючих (рис. 3.33 – 3.34). Можливість легко додавати нові лоти, вказуючи всю необхідну інформацію про товар, його опис, початкову ціну, категорію тощо. Крім того, він може редагувати параметри своїх існуючих лотів, наприклад, змінювати ціну або оновлювати опис. Ця функція надає продавцям повний контроль над їхніми лотами та дозволяє їм ефективно керувати своїм асортиментом на аукціоні. Крім того, це забезпечує зручність та простоту у веденні бізнесу на аукціоні.


[Лоти](#)
[Продати на Bid&Win](#)
[Поради щодо купівлі](#)
[Про нас](#)
[FAQ](#)



При редагуванні, зображення змінювати не можна. Намагайтеся зображення встановлювати на початку.

Max file size: 28.00 MB, Files 0/5, 

Перетягніть файли сюди або натисніть, щоб вибрати

Підтримуються типи: png, jpeg, jpg

Назва *

↶ ↷

Параграф ▾

B

I







▾





▾

: =

: =

= =

= =

Тут повинен бути опис до лоту (про товар)...

Початкова ціна *

€ 0

Дата закінчення *

дд. мм. рррр --:--



Категорія

▾

Ціна купівлі зараз *

€ 0

Крок ставки *

€ 0

Резервна ціна *

€ 0

СТВОРИТИ ЛОТ

Рисунок 3.33 – Форма для створення нового лоту

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

Лоти
Продати на Bid&Win
Поради щодо купівлі
Про нас
FAQ

Лоти > Жіночі босоніжки.

Назва *

Жіночі босоніжки.

↶ ↷

Параграф ▾

B

I

▾

“ ”

▾

: =

≡

≡

Босоніжки жіночі в спортивному стилі. Країна виробник Китай.

Початкова ціна *

€ 120,00

Дата закінчення *

02.06.2024 16:39

Категорія

▾

Ціна купівлі зараз

€ 700,00

Крок ставки

€ 50,00

Резервна ціна

€ 700,00

Оновити лот

Рисунок 3.34 – Форма редагування лоту

3.3. Тестування роботи програмного комплексу

Тестування роботи програмного комплексу є критично важливим етапом розробки будь-якого вебдодатку, такого як аукціон речей. Від його результатів залежить якість та надійність програми, а також задоволення користувачів її функціоналом. Тестування допомагає виявити та виправити можливі помилки та проблеми, забезпечуючи стабільну роботу програми.

Нижче наведений чекліст для тестування вебдодатку аукціону речей:

Таблиця 3.1

Чекліст перевірки Авторизації та реєстрації

Авторизація та реєстрація	
Перевірка можливості успішної авторизації користувача введення правильного логіну та паролю.	Пройдено

Перевірка можливості видачі помилки при невірних або неповних даних авторизації.	Пройдено
Перевірка можливості успішної реєстрації нового користувача з правильно введеними даними.	Пройдено
Перевірка можливості видачі помилки при неповних або невірних даних реєстрації.	Пройдено

Таблиця 3.2

Чекліст перевірки Профілю користувача

Профіль користувача	
Перевірка можливості перегляду особистого профілю користувача з правильною інформацією.	Пройдено
Перевірка можливості редагування особистих даних користувача.	Пройдено

Таблиця 3.3

Чекліст перевірки Створення та редагування лотів

Створення та редагування лотів	
Перевірка можливості додавання нового лоту з правильно введеними даними.	Пройдено
Перевірка можливості редагування параметрів існуючого лоту.	Пройдено
Перевірка можливості видалення лоту з аукціону.	Провалено

Таблиця 3.4

Чекліст перевірки Роботи ставок

Ставки	
Перевірка можливості зроблення ставки на відкритий лот.	Пройдено
Перевірка можливості видалення ставки.	Провалено
Перевірка правильності визначення переможця лоту.	Пройдено

Таблиця 3.5

Чекліст перевірки Адміністративної панелі

Адміністративна панель	
Перевірка можливості доступу до адміністративної панелі з правильними обліковими даними.	Пройдено
Перевірка можливості перегляду та аналізу статистичних даних аукціону.	Пройдено

Перевірка можливості керування користувачами та лотами.	Пройдено
---	----------

Таблиця 3.6

Чекліст перевірки Адаптивності та сумісності

Адаптивність та сумісність	
Перевірка коректності відображення додатку на різних пристроях і розмірах екрану.	Пройдено
Перевірка сумісності з різними веббраузерами (Google Chrome, Mozilla Firefox, Safari, Microsoft).	Пройдено

Висновки до третього розділу

У першому підрозділі було докладно розглянуто налаштування, функціонування та тестування програмного комплексу аукціону речей. В першому підрозділі було показано процес налаштування конфігурації бази даних та використання необхідних технологій для побудови системи та їх версії.

У другому підрозділі було описано програмний продукт, включаючи деталі про процеси виграшу та програшу лотів, а також інші аспекти функціонування аукціону речей. Було розглянуто ключові функціональні можливості, такі як реєстрація та авторизація користувачів, створення та редагування лотів, система ставок, а також інтерфейс користувача.

В третьому підрозділі було описано процес тестування програмного комплексу. Проведено ручне тестування для перевірки коректності роботи всіх функцій додатку. Було складено чекліст, що включає тестування авторизації та реєстрації, роботи профілю користувача, створення та редагування лотів, системи ставок, адміністративної панелі, адаптивності та сумісності з різними браузерами. Проведене тестування допомогло виявити та виправити можливі помилки, що забезпечило стабільну та ефективну роботу системи.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Детально розглянуто процес створення вебдодатку аукціону речей, починаючи від визначення мети та аналізу існуючих рішень до розробки, налаштування та тестування програмного комплексу.

В першому розділі було визначено мету створення вебдодатку, проведено огляд існуючих аукціонних платформ та висвітлено їх недоліки. Було підкреслено важливість забезпечення розширених опцій фільтрації та створення зручного та безпечного інтерфейсу для користувачів. Також було розглянуто архітектуру вебдодатку, обрані технології та компоненти, які забезпечують успішну реалізацію концепції аукціону речей.

Другий розділ був присвячений аналізу функціональних та нефункціональних вимог до системи, створенню діаграм використання та опису структури бази даних. Було визначено основні алгоритми роботи системи, такі як авторизація, закриття лоту та підтвердження покупки, а також детально описано реалізацію програмного коду, що підвищило ефективність та надійність системи.

У третьому розділі було розглянуто процес налаштування конфігурації бази даних та використання необхідних технологій для побудови системи. Описано програмний продукт з точки зору функціонування, включаючи ключові можливості та інтерфейс користувача. Проведено ручне тестування функцій додатку, що допомогло виявити та виправити можливі помилки, забезпечуючи стабільну та ефективну роботу системи.

Таким чином, проведене дослідження продемонструвало комплексний підхід до розробки та тестування вебдодатку аукціону речей. Виконана робота забезпечила якість та надійність додатку, що відповідає поставленим цілям та вимогам, роблячи його зручним та ефективним інструментом для користувачів.

					ІПЗ.КР.Б – 121 – 24 – ІЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архітектура вебдодатків [Електронний ресурс] – Режим доступу до ресурсу:
<https://medium.com/@IvanZmerzlyi/%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%D0%B2%D0%B5%D0%B1%D0%B4%D0%BE%D0%B4%D0%B0%D1%82%D0%BA%D1%96%D0%B2-ca4c82f75bcf>.
2. Asabix /Що таке вебдодатки та як вони функціонують? [Електронний ресурс] – Режим доступу до ресурсу: <https://asabix.com.ua/what-are-web-applications-and-how-do-they-function/>.
3. Itrobes / What are the Advantages and Disadvantages of Web Applications? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.itrobes.com/what-are-the-advantages-and-disadvantages-of-web-applications/>.
4. SoapUI / SOAP vs REST: Understand The Differences [Електронний ресурс] – Режим доступу до ресурсу: <https://www.soapui.org/learn/api/soap-vs-rest-api/>.
5. HOSTIQ / API: простими словами про складне [Електронний ресурс] – Режим доступу до ресурсу: <https://hostiq.ua/blog/ukr/what-is-api/>.
6. hillenblog / Як обрати архітектуру для вебдодатку [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.ithillel.ua/articles/web-application-architecture>.
7. Клієнт-серверна архітектура та ролі серверів. [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@IvanZmerzlyi/%D0%BA%D0%BB%D1%96%D1%94%D0%BD%D1%82%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0%D0%B0%D1%80%D1%85%D1%96%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0%D1%82%D0%B0%D1%80%D0%BE%D0%BB%D1%96%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D1%96%D0%B2-9893d8048229>.

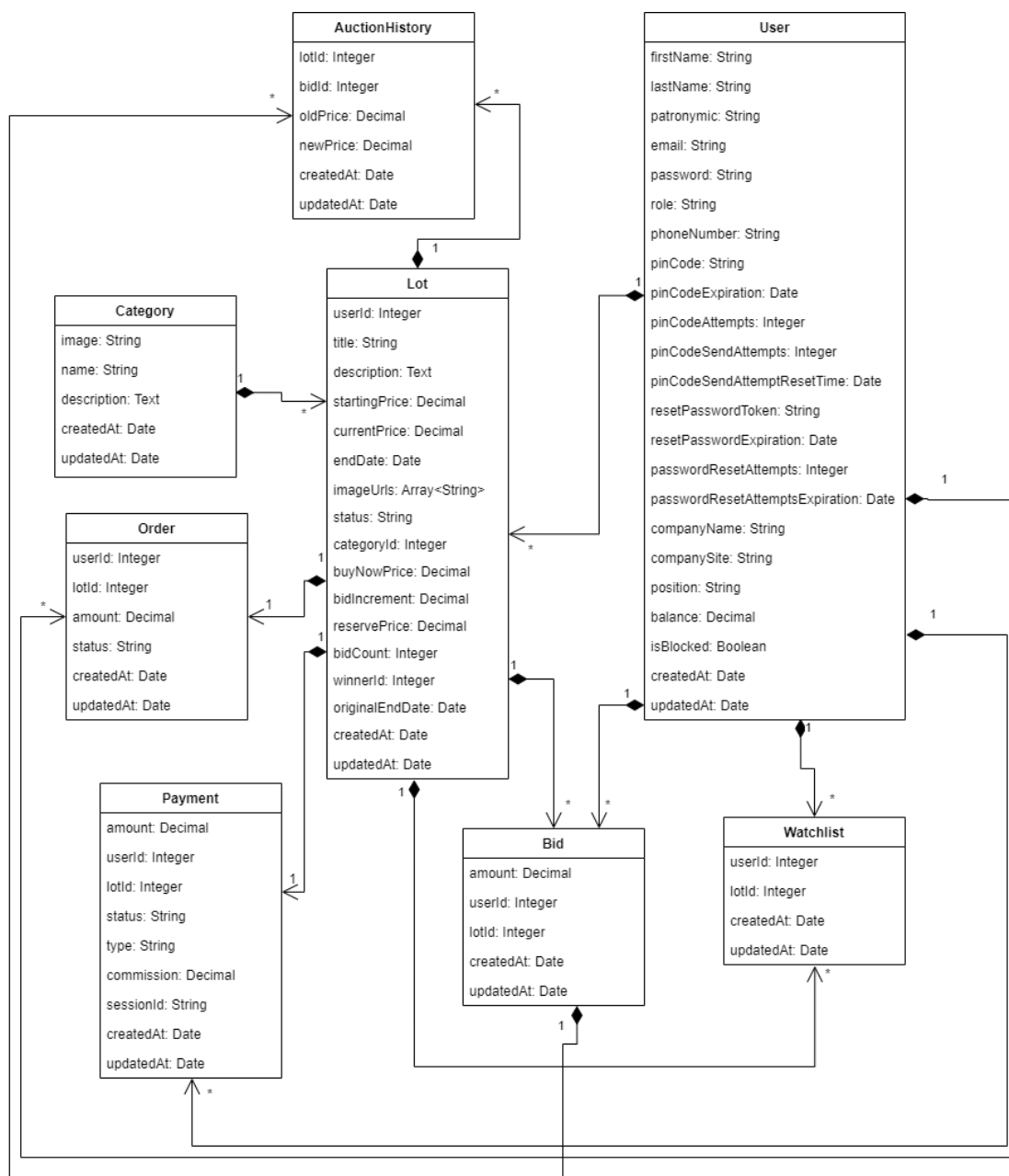
8. Огляд Serverless архітектури [Електронний ресурс] – Режим доступу до ресурсу: <https://devzone.org.ua/post/ogliad-serverless-arxitekturi>.
9. eBAy / Аукціони [Електронний ресурс] – Режим доступу до ресурсу: https://www.ebay.com/b/Ukrainian/4001/bn_55193513?LH_Auction=1&rt=nc.
10. AuctionNinja [Електронний ресурс] – Режим доступу до ресурсу: <https://www.auctionninja.com/>.
11. Kashalot / Кашалот [Електронний ресурс] – Режим доступу до ресурсу: <https://kashalot.com/>.
12. Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
13. Express.js [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/uk/guide/routing.html>
14. Nodemailer [Електронний ресурс] – Режим доступу до ресурсу: <https://nodemailer.com/>
15. React [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.legacy.reactjs.org/docs/hooks-state.html>.
16. express-validator [Електронний ресурс] – Режим доступу до ресурсу: <https://express-validator.github.io/docs/>.
17. React Router [Електронний ресурс] – Режим доступу до ресурсу: <https://reactrouter.com/en/main/hooks/use-action-data>.
18. Redux FAQ: Store Setup [Електронний ресурс] – Режим доступу до ресурсу: <https://redux.js.org/faq/store-setup>.
19. Material UI components [Електронний ресурс] – Режим доступу до ресурсу: <https://mui.com/material-ui/all-components/>.
20. Files UI [Електронний ресурс] – Режим доступу до ресурсу: <https://www.files-ui.com/>.
21. Sequelize/Model Basics [Електронний ресурс] – Режим доступу до ресурсу: <https://sequelize.org/docs/v6/core-concepts/model-basics/>.
22. Introduction to JSON Web Tokens [Електронний ресурс] – Режим доступу до ресурсу: <https://jwt.io/introduction>.

- 23.passport-google-oauth20 [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.passportjs.org/packages/passport-google-oauth20/>.
- 24.decimal.js [Електронний ресурс] – Режим доступу до ресурсу:
<https://mikemcl.github.io/decimal.js/#lte>.
- 25.expressjs/multer [Електронний ресурс] – Режим доступу до ресурсу:
<https://github.com/expressjs/multer?tab=readme-ov-file>.
- 26.Node/Postgres [Електронний ресурс] – Режим доступу до ресурсу:
<https://node-postgres.com/>.
- 27.PostgreSQL Documentation [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.postgresql.org/docs/>.
- 28.React rich text editor component [Електронний ресурс] – Режим доступу до ресурсу:
<https://ckeditor.com/docs/ckeditor5/latest/installation/integrations/react/react.html>.
- 29.Custom Checkout (React) [Електронний ресурс] – Режим доступу до ресурсу:
https://docs.stripe.com/js/custom_checkout/react.
- 30.Formik docs/Validation [Електронний ресурс] – Режим доступу до ресурсу:
<https://formik.org/docs/guides/validation>.

ДОДАТКИ

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						77
Змн.	Арк.	№ докум.	Підпис	Дата		

Діаграма класів:



Лістинг сервісу email-service:

```

const nodemailer = require("nodemailer");

class EmailService {
  async sendEmail(to, subject, text) {
    const transporter = nodemailer.createTransport({
      service: "gmail",
      auth: {
        user: process.env.EMAIL,
        pass: process.env.EMAIL_PASSWORD,
      },
      debug: true,
    });

    const htmlTemplate = `
    <!DOCTYPE html>
    <html lang="en">
    <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Аукцион речей Bid&Win</title>

      <style>
        body {
          background: #e8e8e8;
          font-family: Arial, sans-serif;
          margin: 0;
          padding: 0;
          display: flex;
          justify-content: center;
          align-items: center;
          height: 100vh;
        }
        .card {
          width: 80%;
          max-width: 600px;
          background: linear-gradient(to bottom, #faf09b, lightblue);
          border-radius: 10px;
          padding: 20px;
          text-align: center;
          border: 0.5px solid #dee0dc;
        }
        .heading {
          font-size: 24px;
          font-weight: bold;
          margin-bottom: 20px;
        }
        .content {
          font-size: 16px;

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        line-height: 1.5;
    }
</style>
</head>
<body>
    <div class="card">
        <h1 class="heading">Аукцион речей Bid&Win</h1>

        <div class="content">
            ${text}
            <p>Детальніше на нашому сайті: http://localhost:3000/</p>
        </div>
    </div>
</body>
</html>`;

const mailOptions = {
  from: process.env.EMAIL,
  to,
  subject,
  html: htmlTemplate,
};

try {
  await transporter.sendMail(mailOptions);
} catch (error) {
  console.log("Не вдалося надіслати електронного листа:", error);
}
}

module.exports = new EmailService();

```

Лістинг сервісу lot-service:

```

const { User } = require("../models");
const emailService = require("../services/email-service");

class LotService {
  async closeLot(lot) {
    if (
      lot.status === "CLOSED" ||
      new Date(lot.endDate).getTime() > new Date().getTime()
    ) {
      return;
    }

    lot.status = "CLOSED";

    if (lot.currentPrice < lot.reservePrice) {
      await lot.save();
    }
  }
}

```

					ІПЗ.КР.Б – 121 – 24 – ІЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    return;
  }

  const bids = lot.Bids;
  if (bids.length > 0) {
    const highestBid = bids.reduce(
      (max, bid) => (bid.amount > max.amount ? bid : max),
      bids[0]
    );
    lot.winnerId = highestBid.userId;
  }

  await lot.save();

  if (lot.winnerId) {
    const winner = await User.findByPk(lot.winnerId);
    await emailService.sendEmail(
      winner.email,
      "Вітаємо!",
      `Ви виграли лот: ${lot.title}. Перейдіть за посиланням для оформлення замовлення: http://localhost:3000/payment?lotId=${lot.id}.`
    );
  }

  const losingBids = bids.filter((bid) => bid.userId !== lot.winnerId);
  const userEmails = losingBids.map((bid) => bid.User.email);
  if (userEmails.length) {
    await emailService.sendEmail(
      userEmails,
      "Дякуємо за участь.",
      `На жаль, ви не виграли лот: ${lot.title}. Ми цінуємо Вашу участь і сподіваємося, що Ви знайдете щось цікаве в наших наступних лотах.`
    );
  }
}

async closeLotArray(lots) {
  for (const lot of lots) {
    await this.closeLot(lot);
  }
}
}

module.exports = new LotService();

```

Лістинг сервісу payment-service:

```

const { User, Payment, Order } = require("../models");
const emailService = require("../services/email-service");
const Decimal = require("decimal.js");

```

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						81
Змн.	Арк.	№ док.ум.	Підпис	Дата		

```

class PaymentService {
  async confirmPayment(lot, user, price) {
    const existingOrder = await Order.findOne({
      where: {
        userId: user.id,
        lotId: lot.id,
      },
    });

    if (existingOrder) {
      console.log("Замовлення вже існує. Повторне підтвердження неможливе.");
      return;
    }

    if (lot.status !== "CLOSED") {
      lot.status = "CLOSED";
      lot.winnerId = user.id;
      lot.endDate = new Date();
      await lot.save();
    }

    const bidsWinnerUser = await lot.getBids();
    const userBids = bidsWinnerUser.filter((bid) => bid.userId === user.id);
    if (userBids.length > 0) {
      for (const bid of userBids) {
        const winnerUser = await User.findByPk(bid.userId);
        const amountDecimal = new Decimal(bid.amount);
        winnerUser.balance = new Decimal(winnerUser.balance);

        winnerUser.balance = winnerUser.balance.add(amountDecimal);
        await winnerUser.save();
      }
    }

    price = new Decimal(price);
    const commission = price.times(0.05);
    const amountToSeller = price.minus(commission);

    const seller = await User.findByPk(lot.userId);
    seller.balance = new Decimal(seller.balance);
    seller.balance = seller.balance.add(amountToSeller);
    await seller.save();

    const bids = await lot.getBids();
    const losingBids = bids.filter((bid) => bid.userId !== user.id);
    for (const bid of losingBids) {
      const losingUser = await User.findByPk(bid.userId);
      const amountDecimal = new Decimal(bid.amount);
      losingUser.balance = new Decimal(losingUser.balance);

      losingUser.balance = losingUser.balance.add(amountDecimal);
    }
  }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        await losingUser.save();
    }
    await Payment.create({
        amount: price,
        userId: user.id,
        lotId: lot.id,
        status: "COMPLETED",
        type: "SALE",
        commission: commission,
    });

    const order = await Order.create({
        userId: user.id,
        lotId: lot.id,
        amount: price,
        status: "PROCESSING",
    });

    await emailService.sendEmail(
        user.email,
        "Дякуємо за покупку!",
        `Номер вашого замовлення #${order.id}. З цього моменту воно буде готуватися до відправки. Ми Вас повідомимо, як тільки його відправимо. Дякуємо за довіру!`
    );
}
}

module.exports = new PaymentService();

```

Лістинг сервісу stripe-service:

```

const stripe = require("stripe")(process.env.STRIPE_SECRET_KEY);

class StripeService {
    async createCheckoutSession(
        amount,
        user,
        description,
        successUrl,
        cancelUrl
    ) {
        const session = await stripe.checkout.sessions.create({
            payment_method_types: ["card"],
            shipping_address_collection: {
                allowed_countries: ["UA"],
            },
            line_items: [
                {
                    price_data: {
                        currency: "uah",
                        product_data: {

```

					ПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        name: description,
      },
      unit_amount: amount * 100,
    },
    quantity: 1,
  },
],
billing_address_collection: "auto",
mode: "payment",
customer_email: user.email,
success_url: successUrl,
cancel_url: cancelUrl,
});

return session;
}

async createPaymentIntent(amount, payment_method_id, customer, email) {
  const paymentIntent = await stripe.paymentIntents.create({
    amount: amount * 100,
    currency: "uah",
    payment_method: payment_method_id,
    customer: customer,
    receipt_email: email,
    confirm: true,
    return_url: process.env.CLIENT_URL,
  });

  return paymentIntent;
}

async createCustomer(fullName, email) {
  const customer = await stripe.customers.create({
    name: fullName,
    email: email,
  });

  return customer;
}

module.exports = new StripeService();

```