

Реферат

Інформаційна система автоматизації обліку товарообігу. Дипломний проєкт. Саган Олександр Анатолійович. Галицький коледж імені В'ячеслава Чорновола, відділення комп'ютерних та видавничих технологій. Спеціальність 122 «Комп'ютерні науки». ГК, 2021. Сторінок – 100, рисунків – 50, додатків – 4.

Об'єкт дослідження – методи та засоби автоматизації бізнес-процесів, інструменти для розробки та впровадження інформаційних систем, клієнт-серверна архітектура, методи та інструменти розробки адаптивних веб-інтерфейсів для взаємодії з базою даних.

Метою проєкту є реалізація інформаційної системи, яка забезпечить автоматизацію процесів обліку товарообігу та дасть можливість для подальшого розвитку в напрямку автоматизації інших процесів, які базуються на результатах попередніх.

Система повинна бути реалізована у вигляді бази даних, яка зберігатиме певні записи, буде подавати інформацію у структурованому вигляді та адаптивного веб-інтерфейсу для взаємодії з нею, який забезпечить високу доступність системи.

Інформаційна система повинна забезпечити цілісність та доступність даних та коректно виконувати запроєктовані функції, які забезпечують автоматизацію основних бізнес-процесів.

Для реалізації даної системи було використано інструменти, які надає мова програмування PHP, бібліотека jQuery-UI та СУБД MySQL.

Результатом розробки стала завершена система, яка виконує всі запроєктовані функції та готова до впровадження.

ІНФОРМАЦІЙНА СИСТЕМА, АВТОМАТИЗАЦІЯ БІЗНЕС-ПРОЦЕСІВ, PHP, APCACHE, JQUERY-UI, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, БАЗИ ДАНИ, MySQL.

Abstract

Information system for automation of trade accounting. Diploma project. Sahan Oleksandr Anatoliyovych. Galytsky College named after Viacheslav Chornovil, Department of Computer and Publishing Technologies. Speciality 122 "Computer Science". GC, 2021. Pages – 100, figures – 50, appendixes – 4.

The object of the research is to study and automatize business processes; to develop and implement information systems; to study client-server architecture; to develop adaptive web interfaces for interaction with the database.

The purpose of the project is to implement information system to automatize the processes of recording turnover and to create conditions for automatizing other processes based on the results of previous processes.

The system has to be implemented in the form of a database that will store certain records, provide information in a structured form and an adaptive web interface for interaction with it, which will ensure high availability of the system.

The information system must ensure the integrity and availability of data and correctly perform the designed functions that provide automatizing basic business processes.

For implementing the system, the tools provided by the PHP programming language, as well as the jQuery-UI library and MySQL database were used.

The result of the development was a complete system that performs all the designed functions and is ready for implementation.

INFORMATION SYSTEM, BUSINESS PROCESS AUTOMATION, PHP, APACHE, JQUERY-UI, CLIENT-SERVER ARCHITECTURE, DATABASES, MySQL.

ЗМІСТ

Вступ.....	7
1 Аналіз існуючих рішень та постановка завдання	8
1.1 Дослідження об'єкта інформатизації	8
1.2 Аналіз існуючих рішень	9
1.3 Обґрунтування доцільності створення системи	15
1.4 Постановка завдання.....	17
1.5 Дослідження технологій реалізації	19
2 Проєктування інформаційної системи.....	22
2.1 Формалізація вимог.....	22
2.2 Побудова дерева функцій та проєктування структури системи	24
2.3 Проєктування бази даних	25
2.4 Проєктування користувацького інтерфейсу.....	32
3 Реалізація та тестування інформаційної системи	34
3.1 Опис технологій та засобів реалізації	34
3.2 Реалізація основних функцій інформаційної системи	39
3.3 Реалізація користувацького інтерфейсу системи	44
3.4. Тестування інформаційної системи	49
4 Техніко-економічне обґрунтування	54
4.1 Аналіз ринку	54
4.2 Розрахунок витрат на проєктування інформаційної системи.....	55
4.3 Обґрунтування необхідності розробки	55
Висновки	56
Перелік джерел посилання	57
Додатки.....	59

					ДП.КН 21.449.12.000 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Саган О.А.				Інформаційна система автоматизації обліку товарообігу	Літ.	Арк.	Аркушів
Перевір.	Кульчинська Н.З.						5	100
Реценз.	Кузик В.М.					ГК. КВТ. К-47		
Н.контр.	Гавришків Н.Г.							
Зав. відділ.	Чубей О.О.							

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

СУБД – система управління базами даних.

ІСАОТ – інформаційна система автоматизації обліку товарообігу.

БД – база даних.

ІС – інформаційна система.

ЗП – зберезувальна процедура.

HTML – hyper text markup language.

PHP – Hypertext Preprocessor.

SQL – structured query language.

API – Application Programming Interface.

UI – user interface.

JS – JavaScript.

					<i>ДП. КН 21.449.12.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		6

ВСТУП

Розвиток інформаційних технологій надає можливості досягнути підвищення якості процесів та їх ефективності. Прикладом може слугувати автоматизація бізнес-процесів, яка підвищує продуктивності та знижує ресурсозатратність всіх процесів. Досягнення такого ефекту передбачає впровадження інноваційних рішень, або зміну існуючих. Предметом дослідження даного проєкту є бізнес-процеси підприємства, яке займається реалізацією різних товарів та впровадження інформаційної системи, яка б автоматизувала процеси ведення обліку товарообігу та обчислення показників, які на ньому ґрунтуються.

Враховуючи потреби замовника проєкту було прийнято рішення створити систему на основі клієнт-серверної архітектури, оскільки, вона знижує вартість розробки та спрощує процеси адміністрування і підтримки проєкту. Використання клієнт-серверної архітектури передбачає використання мережі для передачі даних. Для підвищення масштабованості проєкту в якості такої мережі використовується Internet. Використання глобальної мережі забезпечує ще одну важливу характеристику доступність даних. З аналогічних міркувань було прийнято рішення про розробку веб-інтерфейсу для забезпечення взаємодії користувача з базою даних та повноцінного функціонування інформаційної системи.

Отже, головною метою даного проєкту є дослідження основних бізнес-процесів підприємства та їх автоматизація шляхом впровадження інформаційної системи, яка буде виконувати облікові функції в сфері товарообігу та організаційних питаннях.

Актуальність теми дипломного проєкту підтверджується аналізом ринку, який надає широкий вибір програмного забезпечення, що має певні недоліки, через які на замовлення приватного підприємця було реалізовано новий проєкт.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дат		

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1 Дослідження об'єкта інформатизації

Компанія «UNICE multibrand» постачає косметичні та побутові засоби. Магазин займається реалізацією даної продукції у м.Тернопіль та за його межами, у випадку наявності інтернет-замовлень. Клієнт може придбати продукти двома способами: безпосередньо в магазині та через інтернет-магазин. В першому випадку потрібно з'явитися в офіс, обрати необхідний товар. При цьому враховуються різні знижки, які надаються зареєстрованим користувачам. Також розмір знижки залежить від суми покупки. Оплата покупки здійснюється у будь-який зручний час, але забрати можна тільки оплачене замовлення. Факт оплати фіксується в базі даних.

Процес оплати в магазині може відбуватися двома способами: оплата готівкою і на карту. Оплата інтернет-замовлень теж має два варіанти оплати: на карту, або поштовий переказ.

Розмір знижки для замовлень зареєстрованих клієнтів залежить від суми і може становити 25% для покупок до 900 грн., 27% від 900 грн. до 1800 грн. та 30% на суму від 3000 грн.

Керівник магазину повинен постійно вести облік товарообігу, який включає закупку продукції, її збут, нарахування балів консультантам і облік покупок незареєстрованих клієнтів. Крім цього потрібно аналізувати доходи за певні періоди часу: день, тиждень, місяць, квартал і т.д.

При зверненні клієнта в магазин продавець заповнює спеціальну форму, в яку вносяться дані про клієнта, товари та дату покупки. Алгоритм передбачає внесення інформації про товари у базу даних і таким чином дає можливість в майбутньому автоматизувати процеси підрахунку вартості, знижок та перевірки наявності реєстрації. Якщо клієнт зареєстрований, то продавець переходить до наступних кроків, у протилежному випадку пропонується зареєструватись, якщо клієнт відмовляється, то інформація про його покупку просто вноситься в БД без вказування інформації про покупця. При цьому,

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		8

обов'язкове внесення лише артикулів товарів та їх вартості. Покупки зареєстрованих клієнтів теж фіксуються, але крім артикулів і ціни обов'язковою є фіксація інформації про клієнта. Такі дані дають можливість в подальшому відстежувати інформацію про покупки для кожного клієнта.

1.2 Аналіз існуючих рішень

В якості аналогів було розглянуто 3 інтернет-сервіси: система керування підприємством «Мій склад», система керування складом «Simple WMS» та програма для обліку продажів «РемОнлайн».

Система керування підприємством «Мій склад».

Для того, щоб розпочати роботу із сервісом «Мій склад» потрібно заповнити форму реєстрації як на рисунку 1.1.

Рисунок 1.1 – Форма реєстрації у сервісі «Мій склад»

Після успішної реєстрації користувач потрапляє у свій кабінет, де йому пропонується пройти навчання, для кращого розуміння своїх можливостей в рамках сервісу.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дат		

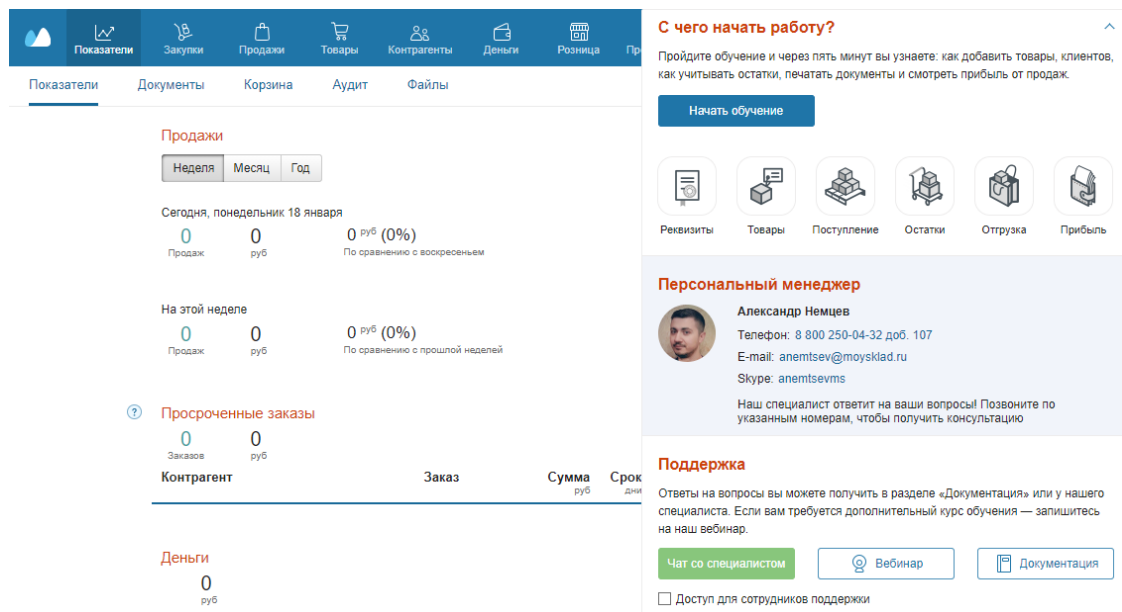


Рисунок 1.2 – Кабінет користувача

Процес навчання включає в себе набір динамічних підказок. Завдяки цими підказками користувач поступово освоює функціонал наданий сервісом.

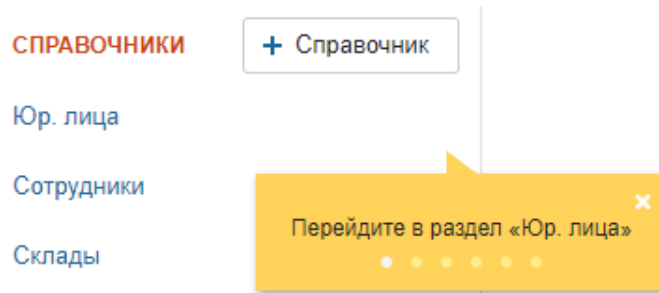


Рисунок 1.3 – Приклад підказки

В процесі навчання було заповнено профіль даними необхідними для роботи, опановано функції додавання товарів, їх закупок, продажів. Також було розглянуто можливість додавання контрагентів та керування ними. Крім того сервіс надає можливість постановки задач для співробітників та дозволяє зручно відстежувати товарообіг.

Перевагами цього сервісу є зручний та зрозумілий користувацький інтерфейс, велика кількість функцій, які можуть бути корисними, наявність мобільного та комп'ютерного додатків, гнучкі налаштування формату даних,

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		10

які зберігатимуться в базі даних. Крім того система вимагає мінімум персональних даних.

Недоліками є відсутність можливостей зміни мови інтерфейсу та додавання певних обчислень на основі інформації про продажі. Сайт коректно відображається лише на комп'ютері у повноекранному режимі. В безкоштовній версії система надає інструменти для роботи лише одному користувачу без можливості залучення співробітників.

Система керування складом «simple wms».

Для початку роботи у «Simple WMS» потрібно зареєструватися. Форму реєстрації подано на рисунку 1.4.

Рисунок 1.4 – Форма реєстрації у сервісі «Simple WMS»

Сервіс надає можливість автоматизувати складські процеси та весь ланцюжок поставок: приймання, розміщення, рух товару на складі, обробка замовлень та контроль роботи співробітників.

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		11

Крім того передбачений функціонал для обліку прибуття та від'їзду транспорту, моніторингу основних показників підприємства, система моніторингу руху товару та логістичний консалтинг.

Перевагами даного сервісу є адаптивний дизайн веб-інтерфейсу, велика кількість функцій, які можуть допомогти в автоматизації роботи підприємства, можливість вибору мови (українська, російська).

Недоліками є необхідність введення великої кількості персональних даних для реєстрації, відсутність мобільного або комп'ютерного додатку, надто складний для нових користувачів інтерфейс.

Більшість функцій потребують додаткових фінансових витрат і при цьому не передбачено алгоритму повернення коштів. Програма для обліку продажів «РемОнлайн». Для початку роботи з сервісом «РемОнлайн» потрібно зареєструватись в системі. Форму реєстрації подано на рисунку 1.5.

Рисунок 1.5 – Форма реєстрації в сервісі «РемОнлайн»

Після реєстрації логіну та пароллю користувачу пропонується заповнити профіль даними. Форму подано на рисунку 1.6.

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		12

**Ваш акаунт майже готовий!
Давайте разом його
налаштуємо**

Ім'я * **Прізвище**

Назва компанії *

Назву компанії завжди можна змінити в налаштуваннях

Країна *

Місто *

Телефон *

Ми не розсилаємо спам. Номер телефону потрібний для відновлення доступу до акаунту.

Налаштувати РемОнлайн під вид діяльності *

Рисунок 1.6 – Форма для заповнення профілю

У формі на рисунку 1.6 є можливість обрати предметну область, до якої відноситься діяльність користувача. Список можливих варіантів подано на рисунку 1.7.

- Не задано -
- Автосервіс / СТО
- Автомийка / Шиномонтаж
- Оренда і прокат
- Ательє з пошиття / ремонту одягу**
- Хімчистки, пральні, клінінг
- Ремонт вело / спортивного спорядження
- Ремонт взуття
- Ремонт мобільних / смартфонів
- Ремонт комп'ютерів / цифрової техніки
- Ремонт техніки Apple
- Ремонт побутової техніки
- Ремонт бензо / електроінструменту
- Ремонт промислового устаткування
- Роздрібні і оптові продажі
- Салони краси та SPA
- Стоматологія
- Медичні центри
- Кондиціонування, вентиляція і опалення
- Вікна та балкони

Рисунок 1.7 – Список предметних областей об'єкта автоматизації

Далі буде запропоновано опцію додавання необхідної кількості співробітників. Форму подано на рисунку 1.8.

					<i>ДП. КН 21.449.12.000 ПЗ</i>	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дат		

Запросіть колег для спільної роботи

Разом ви набагато швидше освоїтеся і оцініте можливості РемОнлайн

Email	Роль
<input type="text"/>	Виконавець ▾
<input type="text"/>	Виконавець ▾

[+ Додати співробітника](#)

Кожний співробітник отримає запрошення на вказаний E-mail. Змінити присвоєну роль або відредагувати її налаштування можна буде в будь-який момент після його реєстрації

Відправити запрошення і почати роботу

[Я зроблю це пізніше](#)

Рисунок 1.8 – Форма додавання співробітників

Після виконання всіх попередніх кроків користувачу надається інтерфейс для роботи відповідно до обраної предметної області (див. рисунок 1.9).

The screenshot shows a user interface for a service. On the left is a dark sidebar with a logo and navigation links: Задачі, Звернення, Замовлення, Продажі, Фінанси, Склад, Клієнти, Аналітика, Звіти, Дзвінки, and Налаштування. The main content area has a top orange banner with an email verification message and a green 'Надіслати ще раз' button. Below it is a grey banner with an SMS notification and a green 'Купити SMS' button. The central part features a 'Замовлення' section with a search bar, a '+ Замовлення' button, a 'Фільтр' button, and a table with columns: Замовлен..., Строк замовле..., Статус, Строк статусу, Створено, Виріб, Несправність, and Клієнт. Below the table, it says 'У вас ще немає замовлень'.

Рисунок 1.9 – Кабінет користувача

Сервіс надає великий вибір однотипних інструментів для роботи зі зверненнями, завданнями, замовленнями, продажами, фінансами, складом та клієнтами. Усі дані зручно підсумуються у розділі «Аналітика» (рис. 1.10).

					ДП. КН 21.449.12.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дат		

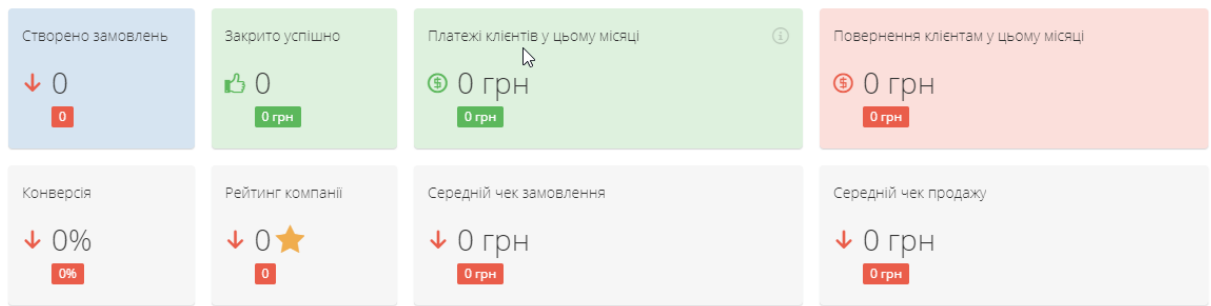


Рисунок 1.10 – Аналітика

Перевагами цього сервісу є зручний та зрозумілий користувацький інтерфейс виконаний в сучасному дизайні, наявність великої кількості розділів та предметних областей, які значно розширюють можливості системи та її користувачів. Система не вимагає надто багато персональних даних і реєстрація не займає багато часу. Також вагомими перевагами є адаптивність веб-інтерфейсу, можливість зміни мови та додавання співробітників.

Недоліками є відсутність можливості додавання певних обчислень на основі інформації про продажі, в безкоштовній версії функціонал надається лише на пробний період.

1.3 Обґрунтування доцільності створення системи

На сучасному етапі розвитку суспільства та інформаційних технологій важливою складовою успішного розвитку підприємства є автоматизація бухгалтерського обліку та доступність інформації.

Потужності обчислювальної техніки надають необхідні інструменти для досягнення певного ступеня автоматизації таких процесів. Вони надають ряд переваг:

- ефективна документація операцій;
- точна інвентаризація;
- наочність товарообігу;
- формування певних підсумків.

Обчислювальна техніка суттєво покращує та спрощує обробку облікової інформації. Це досягається шляхом зменшення кількості ручних операцій та зниженням впливу людського фактору.

Прикладом таких інструментів можуть слугувати розглянуті аналоги.

В ході огляду наявних рішень було виявлено ряд недоліків, які були притаманні першій з розглянутих систем:

- одномовність інтерфейсу;
- неможливість обчислень не передбачених розробниками;
- коректне відображення веб-інтерфейсу лише у повноекранному режимі на комп'ютерах та ноутбуках;
- відсутність можливості взаємодії з користувачами на безкоштовній основі.

Крім того потрібно уникнути недоліків другої системи:

- використання великої кількості персональних даних;
- складний користувацький інтерфейс;
- потреба у додаткових фінансових витратах;
- відсутність алгоритму повернення коштів.

І уникнути недоліків третьої системи:

- безкоштовна лише у пробний термін.

На основі аналізу аналогів було зроблено висновки про доцільність створення нової системи. Ця система повинна володіти такими перевагами:

- зрозумілий користувацький інтерфейс;
- адаптивність інтерфейсу;
- широкий вибір функцій для різноманітних обчислень;
- можливість вибору мов інтерфейсу;
- використання мінімуму персональних даних;
- навчальні підказки для нових користувачів;
- можливість додавання користувацьких формул.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дат		

Таким чином запланована система забезпечить схожий функціонал з рядом додаткових можливостей у місцях, які цього потребують. Також вона забезпечить наступні функції:

- інформаційна;
- комунікаційна;
- контрольна;
- прогнозна.

Суть інформаційної функції полягає у забезпеченні керівників підприємства інформацією необхідною для прийняття подальших рішень. Комунікаційна функція повинна забезпечити зв'язок між рівнями управління та структурними підрозділами. Контрольна функція відповідає за оцінку результатів діяльності підприємства в цілому та на шляху до певної мети. Прогнозна функція повинна забезпечити перспективне планування і координування майбутніх рішень та дій.

Щоб підсумувати все вище сказане потрібно відзначити, що автоматизація значно прискорює процеси обробки інформації, прийняття рішень на основі аналізу даних та підвищує ефективність роботи в цілому. Саме тому впровадження автоматизованої системи обліку товарообігу є доцільним.

1.4 Постановка завдання

Автоматизація обліку товарообігу підприємства значно підвищує ефективність бізнес-процесів, забезпечує швидку комунікацію між адміністрацією, працівниками та клієнтами, забезпечує високу якість надання послуг, і, як наслідок, дозволяє збільшити прибутковість діяльності підприємства.

В процесі аналізу діяльності компанії «UNICE multibrand» було детально досліджено бізнес-процеси, які відбуваються на підприємстві. Результати дослідження подано з використанням діаграми потоків даних, яку зображено

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		17

на рисунку 1.11. Дана діаграма наочно демонструє, які процеси необхідно автоматизувати для підвищення ефективності діяльності компанії.

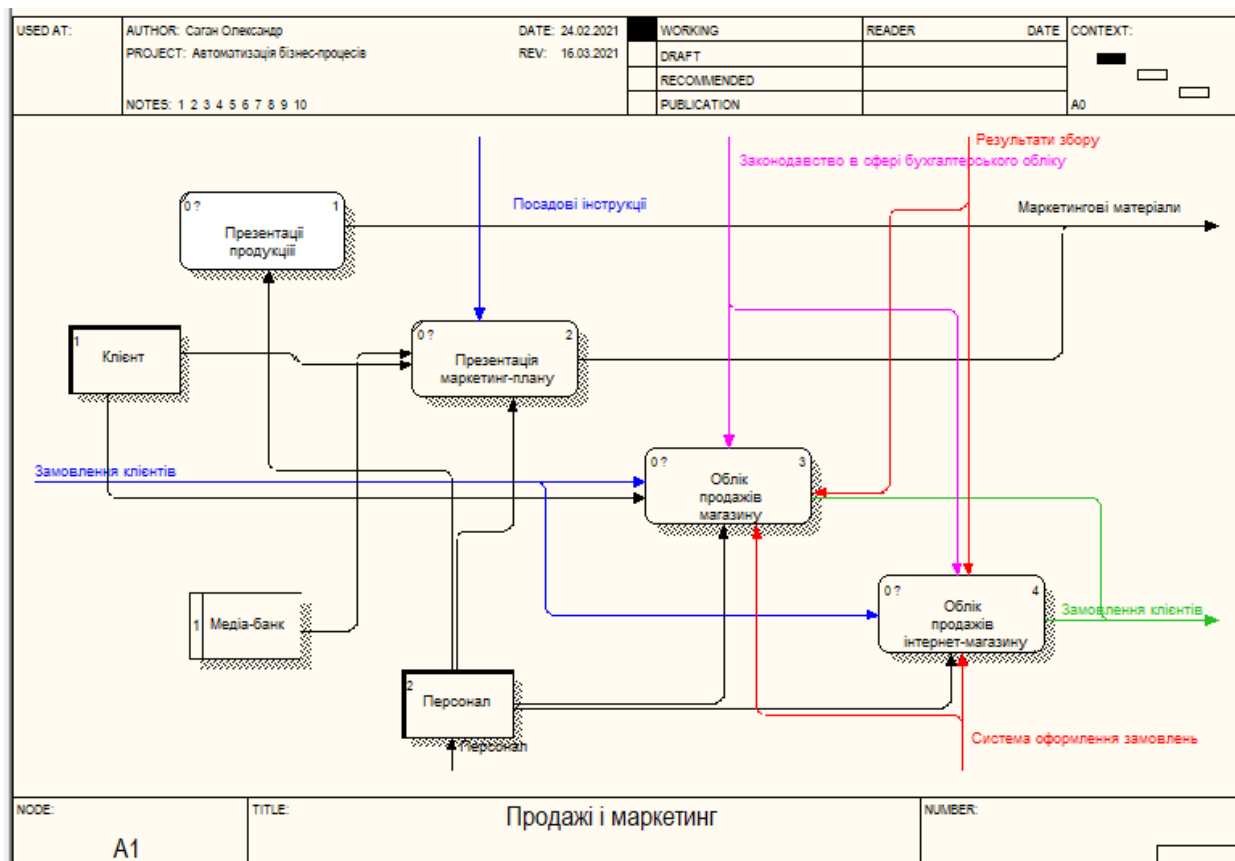


Рисунок 1.11 – Діаграма потоків даних підприємства

Завданням дипломного проектування є розробка інформаційної системи для автоматизації процесів обліку продажів, які включають:

- відстеження наявності товарів;
- збереження інформації про онлайн та оффлайн покупки;
- формування списків продукції на дозамовлення;
- перегляд покупок певних товарів.

Це реалізується шляхом створення бази даних, завданнями якої буде:

- збереження даних у структурованому вигляді;
- цілісність інформації;
- обчислення певних показників;
- доступність даних.

Однією з характеристик доступності даних є можливість їх отримання з будь-якого місця. Для цього необхідно забезпечити доступ до інформації через мережу Internet. Важливою складовою системи стане адаптивний веб-інтерфейс, який забезпечить:

- додавання записів;
- редагування записів;
- видалення записів;
- доступ з різних пристроїв;
- коректне відображення на різних пристроях.

Метою проекту є створення ІС для обліку товарообігу магазину із забезпеченням доступу до даних з використанням сучасного web-інтерфейсу, який забезпечить крім можливості автоматизації облікових функцій також доступ до інформаційної системи через мережу Інтернет.

1.5 Дослідження технологій реалізації

Запланована система має зберігати інформацію у структурованому вигляді, надавати можливість її редагувати і здійснювати різні операції з даними. Саме тому дані повинні зберігатися у базі даних. Це надає актуальності питанню вибору СУБД, які виконують одні і ті ж функції, але суттєво відрізняються алгоритмами роботи. Є багато різних варіантів, які володіють певними перевагами та недоліками. Було розглянуто 6 найпопулярніших СУБД, які використовуються в побудові веб-додатків.

Oracle 12c – продукт компанії Oracle, яка дуже добре зарекомендувала себе на ринку.

Дана СУБД націлена на хмарні сховища та може бути розміщена на кількох серверах. Це дозволяє оперативно керувати базами даних великих обсягів.

Деякі функції найновішої версії Oracle включають в себе grid framework та можливість використання як фізичних, так і логічних структур.

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		19

Серед переваг варто відзначити:

- інноваційність;
- велика функціональність;
- висока надійність.

До недоліків відносяться:

- висока вартість;
- вимагає значних ресурсів.

Далі було розглянуто СУБД MySQL, одну з найпопулярніших в даній галузі розробки. MySQL – це безкоштовний пакет програм від компанії Oracle, який постійно розвивається і удосконалює свій функціонал та безпеку. Існують платні версії даного продукту, але безкоштовна версія здатна виконати переважну більшість завдань, основна увага приділена швидкодії та надійності.

До переваг відноситься:

- можливість використання різних движків для зберігання даних;
- гнучкість;
- безкоштовне поширення;
- хороша документація.

Недоліки:

- відсутня вбудована підтримка XML/OLAP;
- для безкоштовної версії доступна лише платна підтримка.

Ще одним популярним представником СУБД є Microsoft SQL Server. Це система керування базами даних, движок якої працює на хмарних та локальних серверах. При чому типи серверів можна комбінувати.

Переваги:

- легкий у використанні;
- поточна версія працює швидко та стабільно;
- налаштування дозволяють знизити ресурсозатратність;
- можливість візуалізації на мобільних пристроях;
- хороша взаємодія з іншими продуктами Microsoft.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дат		

Недоліки:

- висока вартість;
- навіть при детальних налаштуваннях може використовувати всі ресурси;
- трапляються збої в роботі служби інтеграції для імпорту файлів.

Останім з розглянутих варіантів стала СУБД MongoDB. Це ще одна безкоштовна система, яка має комерційну версію. Вона застосовується у випадках, які передбачають використання як структурованих, так і неструктурованих даних. Ядро СУБД добре розвинуте і має велику кількість драйверів для роботи з багатьма мовами програмування.

Переваги: швидкість, простота у використанні; підтримка json та інших NoSQL-документів, швидке читання та збереження даних.

Недоліки: SQL не використовується в якості мови запитів, встановлення та налаштування займає багато часу [1].

В процесі роботи запланованої системи потрібно буде оперувати різними даними. Для цього потрібний зручний інтерфейс. Інтерфейс наданий СУБД не зможе надати потрібних можливостей користувачам без спеціальної підготовки. Саме тому було прийнято рішення розробити веб-інтерфейс.

Для створення веб-інтерфейсу потрібно скористатися мовою гіпертекстової розмітки, каскадними таблицями стилів CSS та мовою JavaScript або PHP. Зручним поєднанням цих інструментів є фреймворк Bootstrap. Він надає широкий вибір вже готових функцій та стилів.

Вибір між JavaScript та PHP не може однозначно виділити кращу мову, оскільки області застосування багато в чому суттєво відрізняються. І хоч вони можуть повністю виключати один одного, проте найбільш оптимальним варіантом є поєднання цих двох інструментів. Це пов'язане з тим, що обидві мови стрімко розвиваються і вже на даний момент є дуже популярними. Тобто можна зробити висновок про подальшу їх актуальність ще протягом досить тривалого часу.

					<i>ДП. КН 21.449.12.000 ПЗ</i>	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дат		

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Формалізація вимог

В результаті постановки завдання проєкту, було визначено необхідність проєктування та розробки інформаційної системи, яка дозволить автоматизувати функції додавання клієнтів, облік поставок, облік продажів, нарахування знижок, отримання інформації за прізвищем клієнта, перегляд інформації про покупки за певний період.

Користувачами даної системи повинні бути працівники магазину, його власник, клієнти.

На рисунках 2.1-2.2 подано бізнес-процеси, які будуть автоматизовані.

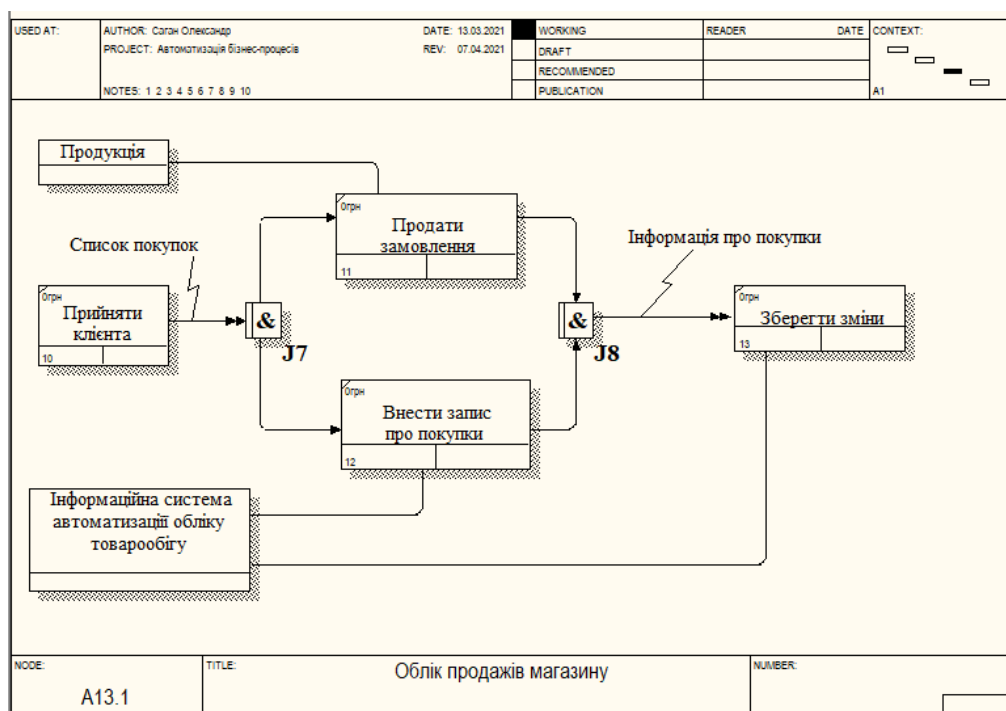


Рисунок 2.1 – Діаграма бізнес-процесу офлайн покупки

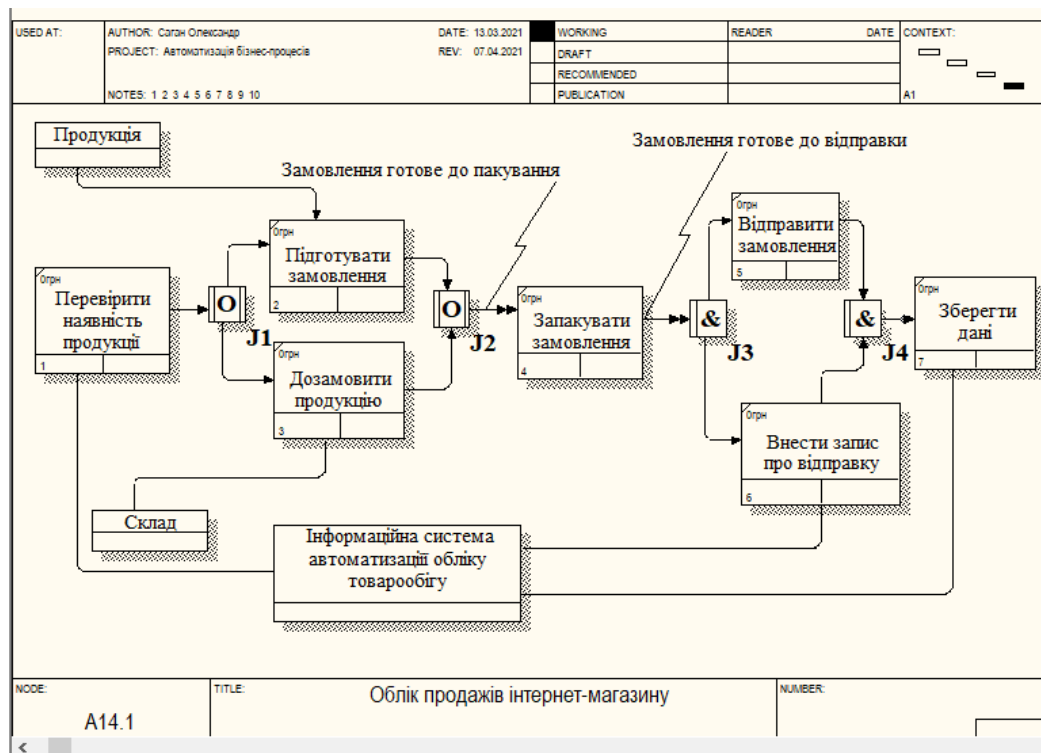


Рисунок 2.2 – Діаграма бізнес-процесу онлайн покупки

Основні елементи узагальненої умовно замкненої моделі подано на рисунку 2.3.

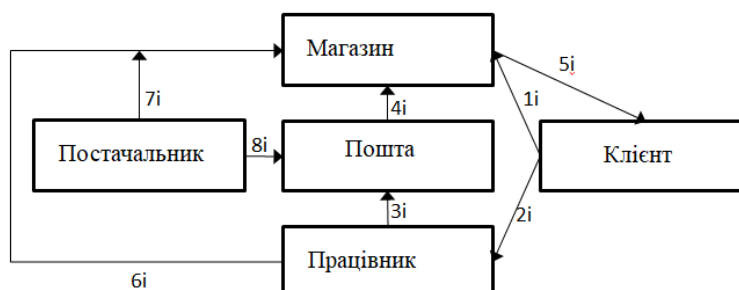


Рисунок 2.3 – Узагальнена умовно замкнута модель інформаційної системи

Зміст інформаційних управляючих зв'язків схеми (рис. 2.3):

- 1i – внесення персональних даних клієнта;
- 2i – клієнт звернувся до працівника магазину з метою придбання товарів;
- 3i – працівник забрав замовлення з пошти;
- 4i – товари з пошти прибули в магазин;
- 5i – магазин надав замовлені товари;

6і – за наявності продукції в магазині працівнику не потрібно звертатись до пошти;

7і – можлива адресна доставка від постачальника в магазин;

8і – замовлення прийшло від постачальника на пошту.

2.2 Побудова дерева функцій та проектування структури системи

Основним завданням магазину є оперативне оновлення інформації про товари і швидке обслуговування клієнтів. Для цього працівник фірми повинен постійно вести облік продукції, клієнтів, закупок тощо. Крім цього потрібно аналізувати числові показники.

Дерево функцій системи подано на рисунку 2.4.

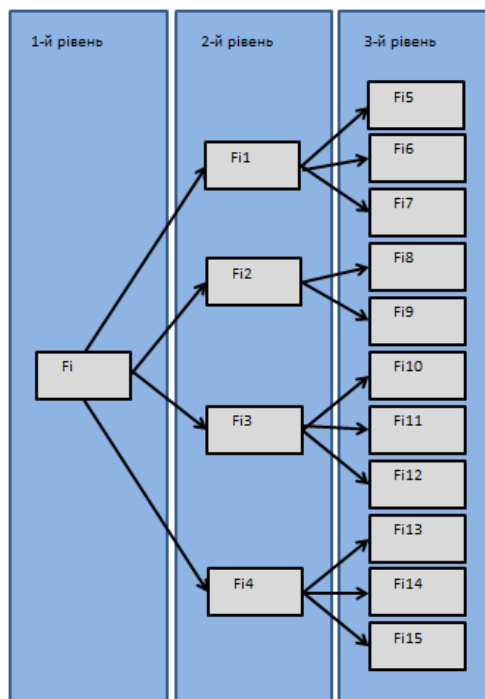


Рисунок 2.4 – Дерево функцій інформаційної системи

Fi1 – облік товарів:

Fi5 – облік надходжень;

Fi6 – облік наявності;

Fi7 – облік продажів.

Fi2 – інформація про доходи:

Fi8 – сумарна вартість замовлених товарів;

					ДП. КН 21.449.12.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дат		

Fi9 – доходи за певний період.

Fi3 – облік клієнтів:

Fi10 – нові клієнти (ще не зареєстровані);

Fi11 – постійні клієнти (зареєстровані);

Fi12 – отримані клієнтами замовлення.

Fi4 – облік товарообігу:

Fi13 – замовлені (ще в дорозі);

Fi14 – на складі (очікують покупця);

Fi15 – продано.

Відповідно до визначеного функціоналу було спроектовано структуру інформаційної системи, яка подана на рисунку 2.5.

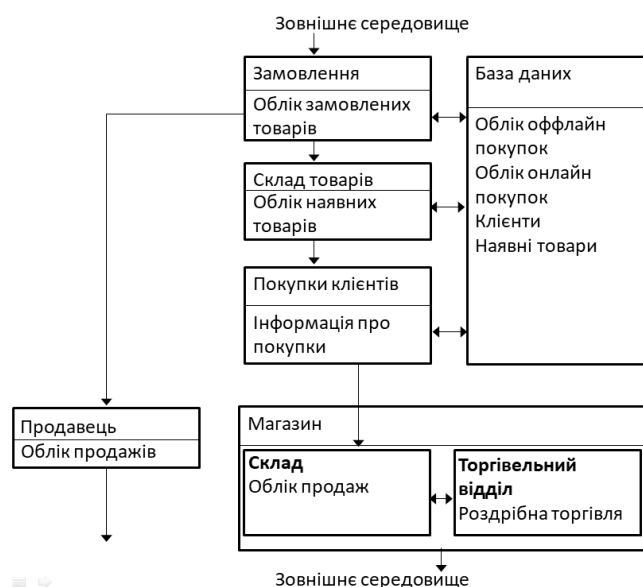


Рисунок 2.5 – Схема компонентів системи

2.3 Проектування бази даних

В результаті аналізу предметної області було виділено такі інформаційні сутності: «Керівник», «Продавець», «Покупець», «Замовлення», «Товари», «Звіт про продані товари», «Магазин», «Рядок замовлення». Далі було повторно проаналізовано предметну область для встановлення зв'язків між сутностями і проведено подальші дії процесу проектування [2].

В результаті аналізу між сутностями «Керівник» і «Магазин» було встановлено зв'язок типу «один до багатьох». В рамках поставленої задачі цей

зв'язок не має великого значення, оскільки впровадження системи передбачене для одного магазину, але в майбутньому він дає можливість додавання нових магазинів у випадку їх відкриття. Таким чином виділення керівника в окрему сутність значно підвищує масштабованість проєкту в цілому.

У магазині повинен працювати продавець. Наявність відповідної сутності дає можливість додавання кількох продавців, якщо у магазині працюватиме більше одного. Тому між сутностями «Магазин» і «Продавець» було встановлено зв'язок аналогічного типу, а саме: «один до багатьох».

Суть проєкту полягає в автоматизації основних бізнес-процесів, одним з яких є облік товарообігу. Саме тому важливо виокремити сутність «Наявні товари». Факт наявності означає їх знаходження у магазині, тому було встановлено аналогічний попереднім зв'язок «Наявні товари»-«Магазин».

Далі було встановлено зв'язок «багато до багатьох» між сутностями «Клієнт» і «Наявні товари». Він реалізується через проміжну сутність «Рядок замовлення». Було використано неідентифікуючі зв'язки, бо система має надавати можливість переглядати покупки певних покупців і кількість покупок певного товару.

Однією з функцій запланованої системи повинне бути формування звітів. Саме для цього було додано сутність «Звіти про продажі», яка зв'язується зв'язком типу «один до багатьох» із «Рядком замовлення» як це зображено у концептуальній моделі (рис. 2.6).



Рисунок 2.6 – Концептуальна модель бази даних

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		26

Після виділення сутностей було визначено необхідні ключі, які б давали можливість однозначно ідентифікувати кожну з них. Для кожної сутності було визначено унікальний числовий ідентифікатор, який не повторюється у представників однієї сутності і не пов'язаний з іншими. Далі було визначено первинні ключі та здійснено їх міграцію для реалізації запроєктованих зв'язків, як на рисунку 2.7.

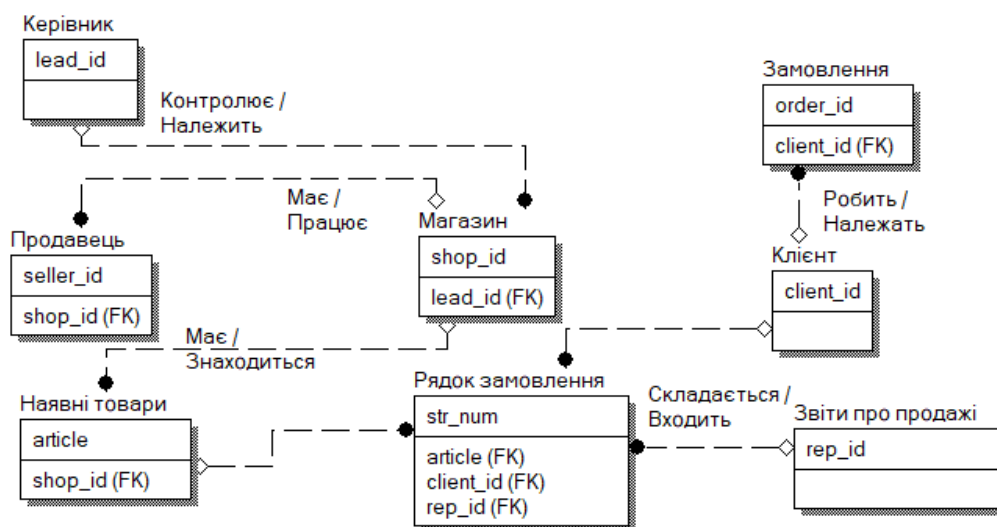


Рисунок 2.7 – Міграція ключів

Після виконання міграції ключів було визначено набір атрибутів для кожної сутності, які забезпечать збереження всіх необхідних даних для обліку товарообігу. Набори атрибутів подано у таблиці 2.1.

Таблиця 2.1 – Аналіз інформаційних об'єктів

Складний інформаційний об'єкт	Атрибути інформаційного об'єкту	Характеристики об'єкту
Керівник	lead_id lead_name lead_phone lead_email	Дано про керівника магазину (ім'я, телефон, ел. Адреса).

Продовження таблиці 2.1

Клієнт	client_id	Дано про клієнта (ім'я, телефон, інформація про реєстрацію).
	client_name	
	clientr_phone	
	client_reg	
Магазин	shop_id shop_address sold_sum bought_sum rent_coast	Дані про магазин (адреса, вартість проданої та закупленої продукції, вартість оренди).
Наявні товари	article gds_name gds_price gds_count	Дані про наявні товари (артикул, назва, ціна, кількість).
Замовлення	order_id client_id order_date article	Дані про замовлення (клієнт, дата, артикул).
Рядок замовлення	str_num article client_id gcount rep_id	Дані про покупки (ідентифікатор рядка, артикул, клієнт, кількість певного товару, ідентифікатор звіту).
Звіти про замовлення	order_id client_id order_date article	Дані про покупки конкретного клієнта (ідентифікатор звіту, ідентифікатор клієнта, дата, артикул).

Після міграції ключів та визначення наборів основних атрибутів було здійснено глибинний аналіз сутностей та їх атрибутів як у таблиці 2.2.

Таблиця 2.2 – Глибинний аналіз об'єктів-сутностей, та їх атрибутів

Інформаційний об'єкт сутності	Атрибут сутності	Ідентифікатор сутності	Ключі
Керівник	lead_id	lead_id	Р-ключ
	lead_name	lead_name	-
	lead_phone	lead_phone	-
	lead_email	lead_email	-

Продовження таблиці 2.2

Продавець	seller_id	seller_id	Р-ключ
	seller_name	seller_name	-
	seller_phone	seller_phone	-
	shop_id	shop_id	Р-ключ
	seller_email	seller_email	-
Клієнт	client_id	client_id	Р-ключ
	client_name	client_name	-
	client_phone	client_phone	-
	client_reg	client_reg	-
Магазин	shop_id	shop_id	Р-ключ
	shop_address	shop_address	-
	sold_sum	sold_sum	-
	bought_sum	bought_sum	-
	lead_id	lead_id	Р-ключ
	rent_coast	rent_coast	-
Наявні товари	article	article	Р-ключ
	shop_id	shop_id	Р-ключ
	gds_name	gds_name	-
	gds_price	gds_price	-
	gds_count	gds_count	-
Замовлення	order_id	order_id	Р-ключ
	client_id	client_id	Р-ключ
	order_date	order_date	-
	article	article	-
Рядок замовлення	str_num	str_num	Р-ключ
	article	article	Р-ключ
	client_id	client_id	Р-ключ
	gcount	gcount	-
	rep_id	rep_id	Р-ключ
Звіти про замовлення	order_id	rep_id	Р-ключ
	client_id	client_id	-
	order_date	order_date	-
	article	article	Р-ключ

Таблиця 2.3 – Аналіз та опис зв'язків між сутностями

Зв'язки між сутностями	Назва зв'язку	Тип зв'язку	Зміст зв'язку
Керівник-магазин	Контролює / Належить	1:Б	Керівник може контролювати декілька магазинів.

Продовження таблиці 2.3

Магазин-продавець	Має / Працює	1:Б	В одному магазині може працювати кілька продавців.покупець може мати кілька замовлень.
Магазин-Наявні товари	Має / Знаходиться	1:Б	Багато звернень можуть бути прийняті одним працівником;
Клієнт-Замовлення	Робить / Належить	1:Б	Один клієнт може зробити кілька замовлень.
Клієнт-Наявні товари	Купує / Належать	Б:Б	Багато клієнтів можуть купити багато товарів.

Наступним етапом проєктування є побудова логічної моделі реляційної бази даних.

Таблиця 2.4 lead

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	Р
_name	Текстовий	-	-	-	-	-
_phone	Текстовий	-	-	-	-	-
_email	Текстовий	-	-	-	-	-

Таблиця 2.5 client

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	Р
_name	Текстовий	-	-	-	-	-
_phone	Текстовий	-	-	-	-	-
_email	Текстовий	-	-	-	-	-

Таблиця 2.6 seller

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	P
_shop	Числовий	-	-	-	-	F
_name	Текстовий	-	-	-	-	-
_phone	Текстовий	-	-	-	-	-
_email	Текстовий	-	-	-	-	-

Таблиця 2.7 shop

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	P
_lead	Числовий	-	-	-	-	F
_address	Текстовий	-	-	-	-	-
_sold	Грошовий	-	-	-	-	-
_bought	Грошовий	-	-	-	-	-
_rent	Грошовий	-	-	-	-	-

Таблиця 2.8 goods

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_article	Текстовий	-	-	-	-	P
_shop	Числовий	-	-	-	-	F
_name	Текстовий	-	-	-	-	-
_price	Грошовий	-	-	-	-	-
_count	Числовий	-	-	-	-	-

Таблиця 2.9 order_str

Атрибути сутності	Обмеження атрибутів					Ключ
	Тип даних	Межі або допустимі значення	Структура (формат)	Умова	Значення за замовчуванням	
_id	Лічильник	-	-	-	-	P
_article	Числовий	-	-	-	-	F
_client	Текстовий	-	-	-	-	F
_count	Числовий	-	-	-	-	-
_date	Дата і час	-	-	-	-	-

2.4 Проєктування користувацького інтерфейсу

Отже, необхідно розділити функціонал на декілька ролей, як зображено на рисунку 2.8.

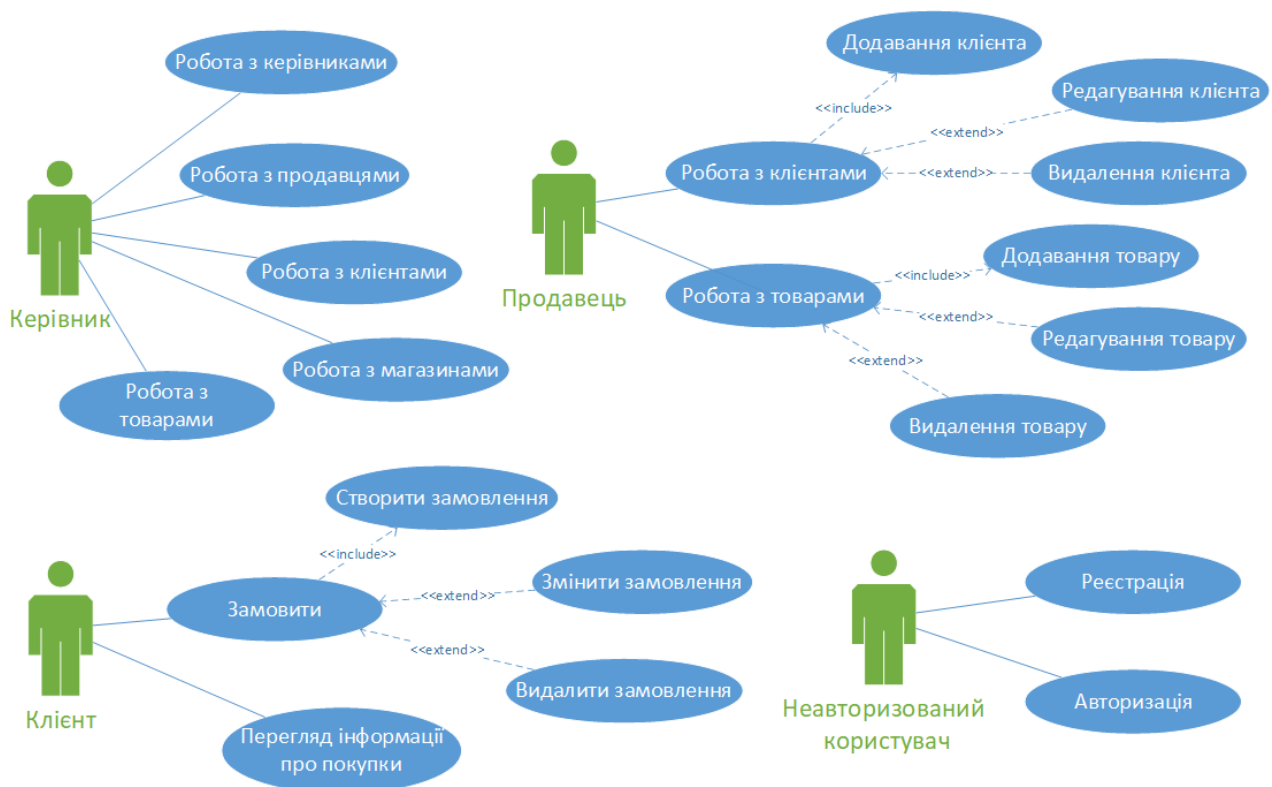


Рисунок 2.8 Діаграма прецедентів

Функції керівника:

- робота з новими керівниками (додавання, редагування, видалення);
- робота з продавцями (додавання, редагування, видалення);

- робота з клієнтами (додавання, редагування, видалення);
- робота з магазинами (додавання, редагування, видалення);
- робота з товарами(додавання, редагування, видалення).

Функції продавця:

- робота з клієнтами (додавання, редагування, видалення);
- робота з товарами(додавання, редагування, видалення).

Функції зареєстрованого користувача:

- додавання замовлень;
- перегляд інформації про свої покупки.

Функції неавторизованого користувача:

- реєстрація;
- авторизація.

Для реалізації даного функціоналу було спроектовано ряд форм, який включає: «Керівник», «Продавець», «Клієнт», «Товар», «Магазин», «Покупки», «Замовлення», «Реєстрація», «Авторизація».

Форма «Керівник» повинна надавати можливість додавати, видаляти і оновлювати інформацію про керівників, а саме: ім'я, телефон, електронна, пошта.

Форми «Продавець», «Клієнт», «Магазин» повинні надавати аналогічний функціонал для роботи з інформацією про відповідні сутності.

Форма «Товар», крім можливості додавання, редагування і видалення одного екземпляру повинна забезпечити внесення групи елементів, в той час як попередні форми такої можливості не потребують.

Форма «Покупки» теж буде повинна забезпечувати можливість внесення інформації, але відрізняється тим, що не повинна надавати можливості перегляду, редагування та видалення.

Форма «Реєстрація» повинна надавати можливість реєстрації в системі, а форма «Авторизація» забезпечує вхід в систему зареєстрованих користувачів.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дат		

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Опис технологій та засобів реалізації

Для конкретно цієї задачі було обрано СУБД phpMyAdmin. Ця СУБД має певні недоліки:

- громіздке відображення таблиць;
- помилки у генерації зв'язків.

Але компенсує їх великою кількістю переваг, а саме:

- автоматична генерація схеми;
- відображення ключів;
- зручне розташування схеми даних;
- низькі системні вимоги;
- безкоштовне поширення;
- веб-орієнтованість;
- постійний розвиток та розширення функціоналу;
- висока швидкість та надійність;
- можливість вибору ядра СУБД;
- велика популярність;
- хороша документація;
- може працювати з іншими БД [3].

Відповідно до завдань проєкту найбільш оптимальним варіантом інструментів розробки стала мова програмування PHP. Вона забезпечить можливість реєстрації, авторизації та надання користувацького інтерфейсу для роботи з базою даних. Система має бути максимально доступною, тому було прийнято рішення створити інтерфейс, який зможе підлаштовуватись під пристрій користувача. Хорошим інструментом для реалізації адаптивних веб-інтерфейсів є фреймворк Bootstrap [4].

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		34

Розробка інформаційної системи буде відбуватись на локальному сервері «Open Server» [6]. Для його інсталяції було завантажено виконуваний файл з офіційного сайту, який показаний на рисунку 3.1.



Рисунок 3.1 – Офіційний сайт open server

Після завершення завантаження було виконано наступні кроки: вибір мови, прийняття ліцензійних умов, вибір папки для встановлення, вибір компонентів та ін. Після вибору всіх налаштувань було виконано інсталяцію, як зображено на рисунку 3.2.

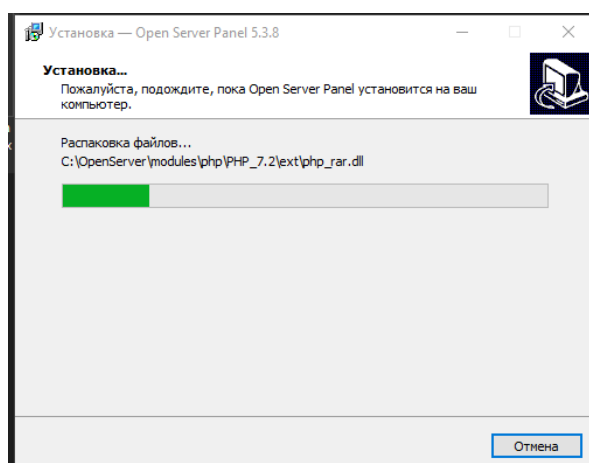


Рисунок 3.2 – Інсталяція локального сервера

Після успішного завершення було перезавантажено комп'ютер і встановлено налаштування версій, як показано на рисунку 3.3.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дат		

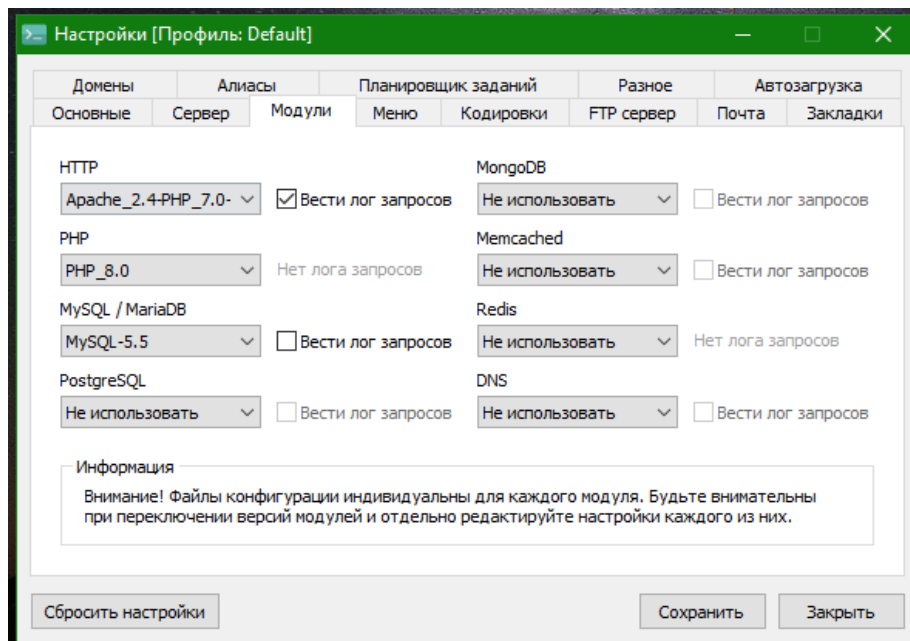


Рисунок 3.3 – Налаштування локального сервера

В якості IDE було обрано PhpStorm і завантажено з офіційного сайту, показаного на рисунку 3.4 [7].

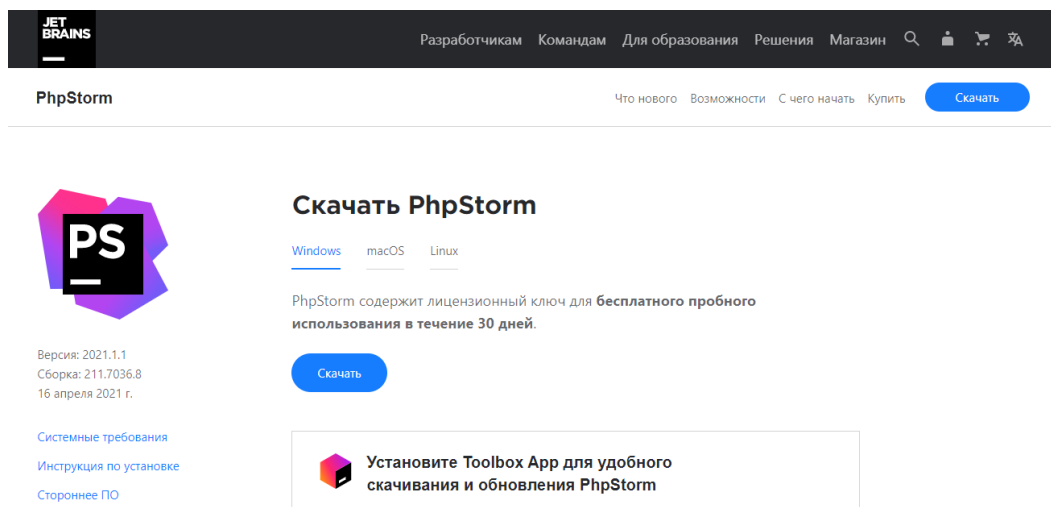


Рисунок 3.4 – Офіційний сайт JetBrains

В процесі інсталяції було обрано шлях для встановлення, і компоненти, які зображено на рисунку 3.5.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дат		

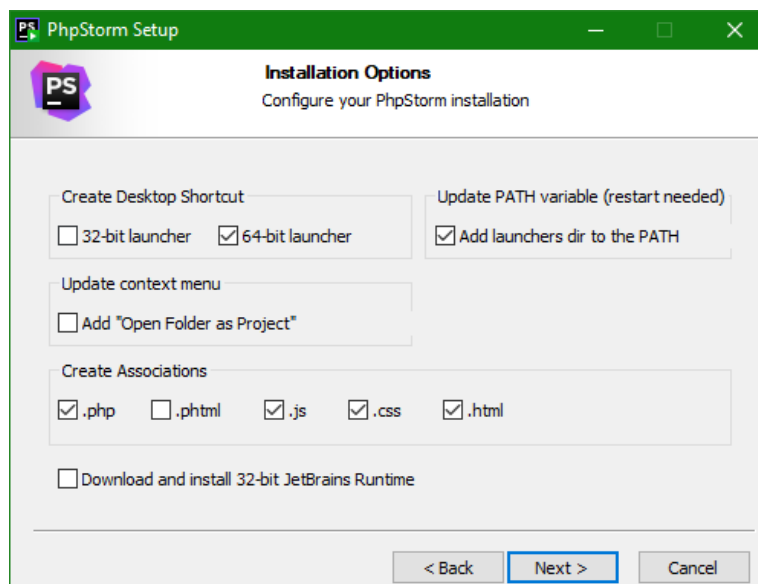


Рисунок 3.5 – Компоненти для встановлення

Далі було завантажено менеджер пакетів з офіційного сайту Composer, як на рисунку 3.6 [8].



A Dependency Manager for PHP

Latest: 2.0.12 ([changelog](#))

Composer 2.0 is now available! [Read our announcement!](#)

Рисунок 3.6 – Офіційний сайт Composer

При запуску виконуваного файлу запуситься інсталятор, який зображено на рисунку 3.7.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дат		

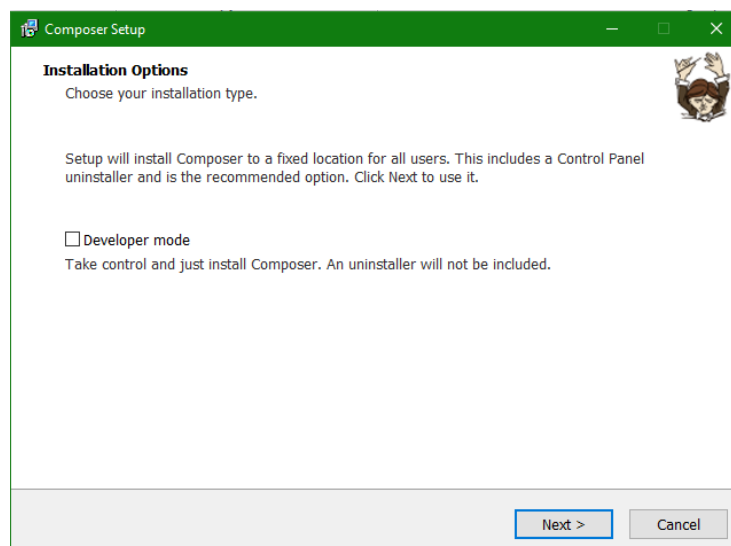


Рисунок 3.7 – Інсталятор Composer

Далі було обрано шлях до інтерпретатора php, як на рисунку 3.8.

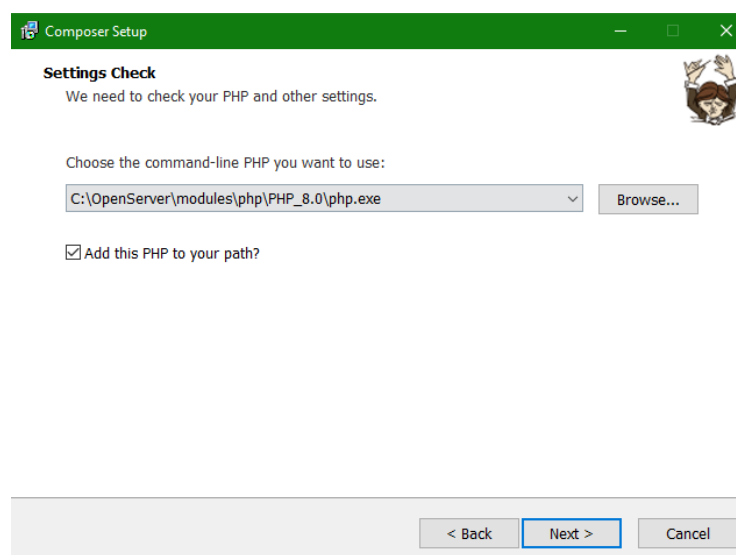


Рисунок 3.8 – Шлях до інтерпретатора php

Після завершення інсталяції було отримано відповідне повідомлення, яке зображено на рисунку 3.9.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дат		

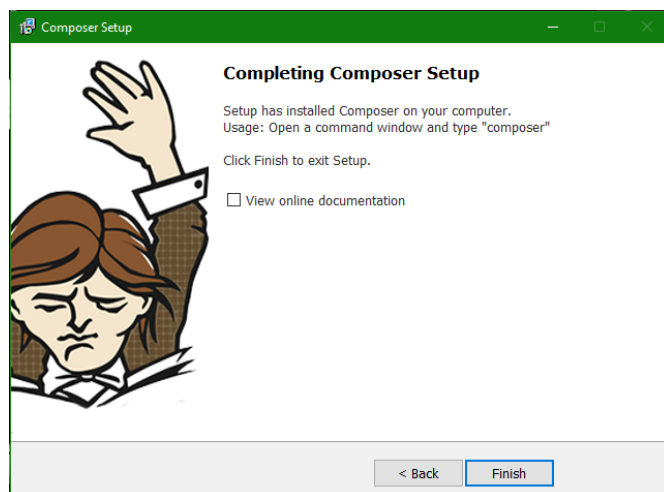


Рисунок 3.9 – Завершення інсталяції Composer

3.2 Реалізація основних функцій інформаційної СИСТЕМИ

Для реалізації функціоналу було розроблено базу даних, яка відповідає фізичній моделі, зображеній на рисунку 3.11

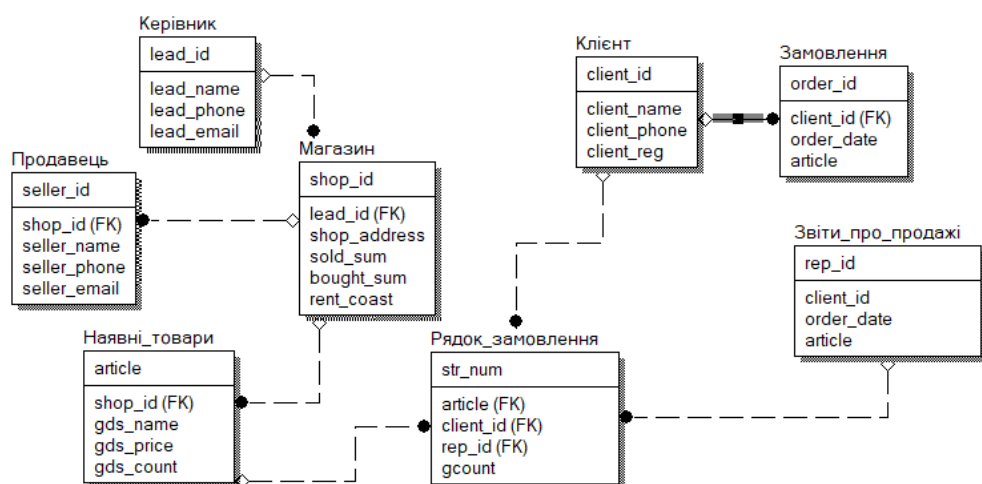


Рисунок 3.11 – Фізична модель бази даних

Відповідно до фізичної моделі було створено запити на створення таблиць (лістинг A1, додаток А), унікальних полів, індексів, ключів (лістинг A2, додаток А), та зберезувальні процедури, які дозволяють додавати і змінювати та видаляти дані виконуючи додаткові маніпуляції з вхідними параметрами [9].

Варто відзначити, що змінні екрануються, серверу передається прототип запиту та значення параметрів. Після чого виконується безпечний запит захищений від SQL-ін'єкцій [10].

Для сутності «Керівник» було створено 3 збережувальні процедури (лістинг A3, додаток A):

- create_lead;
- update_lead;
- delete_lead.

Збережувальна процедура create_lead приймає 3 параметри: ім'я керівника, електронну пошту і телефон. Далі перевіряється наявність керівника, у якого хоч 1 з полів буде співпадати з новим. Якщо збіг знайдено, то запис не буде додано. Якщо ж збігів не виявлено, то буде підготовано прототип запиту, передано його на сервер і окремо передано екрановані параметри, після чого запит буде виконано.

Збережувальна процедура update_lead приймає 4 параметри: ідентифікатор запису, ім'я користувача, телефон та електронну пошту. Після цього перевіряється можливість зміни даних керівника, на нові, тобто шукаються збіги з іншими записами. Якщо збіг знайдено, то запис не буде оновлено, в іншому випадку на серверу окремо передаються прототип запиту та параметри. Після екранування параметрів вони підставляються в запит і виконується оновлення записів.

Збережувальна процедура delete_lead приймає 1 параметр: ідентифікатор запису і виконує запит на видалення запису з відповідним ідентифікатором у таблиці керівників.

Для сутності магазин було реалізовано 3 збережувальних процедури (лістинг A4, додаток A):

- create_shop;
- update_shop;
- delete_shop.

Збережувальна процедура `create_shop` приймає 5 параметрів: ідентифікатор керівника, адреса магазину, сумарну вартість проданих і закуплених товарів на момент додавання запису про конкретний магазин та вартість оренди. Далі перевіряється наявність збігів між адресам наявних магазинів і нового. Якщо збігів не знайдено, то запис буде додано, у протилежному випадку виконання запиту не відбудеться.

Збережувальна процедура `update_shop` приймає ті ж параметри, що й `create_shop` і додатково ідентифікатор запису, який потрібно змінити. Далі йде перевірка можливості внесення змін, тобто порівнюються значення атрибутів наявних записів із значеннями, які передаються процедурі оновлення інформації. Якщо збігів не знайдено, то запис, ідентифікатор якого передано у збережувальну процедуру, оновлюється.

Збережувальна процедура `delete_shop` приймає 1 параметр: ідентифікатор запису. Оскільки, вхідний параметр приводиться до цілочисельного значення, екранування не виконується, через відсутність потреби. Перевірка наявності запису з переданим ідентифікатором, перед виконанням запиту на видалення, додана з метою подальшого нарощування функціоналу.

Для сутності «Продавець» було реалізовано 3 збережувальні (лістинг А5, додаток А) процедури, а саме:

- `create_seller`;
- `update_seller`;
- `delete_seller`.

Збережувальна процедура `create_seller` приймає 4 параметри: ідентифікатор магазину, за яким закріплений певний продавець, ім'я продавця, телефон, електронна пошта. Після цього перевіряється наявність записів, у яких хоч 1 з полів буде співпадати із полями нового запису. Якщо збіг знайдено запис не буде додано, в протилежному випадку буде додано запис про нового продавця.

Збережувальна процедура `update_seller` приймає 5 параметрів: ідентифікатор запису, який потрібно змінити, ідентифікатор магазину, за яким

					ДП. КН 21.449.12.000 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дат		

закріплений продавець, оновлене ім'я продавця та його контакти: номер телефону та електронна пошта. Якщо модифікація наявного запису не спричинить появи збігів полів ім'я, електронна пошта, номер телефону, то запис буде оновлено, в іншому випадку оновлення не відбудеться.

Збережувальна процедура `delete_seller` приймає 1 параметр: ідентифікатор запису про продавця, який потрібно видалити і виконує запит на видалення з відповідним параметром.

Для сутності «Клієнт» було розроблено 3 збережувальних процедури (лістинг А6, додаток А):

- `create_client`;
- `update_client`;
- `delete_client`.

Збережувальна процедура `create_client` приймає 3 параметри: ім'я клієнта, номер телефону та електронну адресу. Вставлення запису відбудеться лише за умови відсутності збігів в значеннях полів записів, після екранування параметрів та передачі прототипу запиту.

Збережувальна процедура `update_client` приймає 4 параметри: ідентифікатор клієнта, ім'я, телефон та електронну пошту, після чого відбувається перевірка наявності збігів, аналогічна попереднім і екранування параметрів з подальшим виконанням запиту.

Збережувальна процедура `delete_client` приймає один параметр: ідентифікатор клієнта, після чого виконується перевірки наявності даного і запису і запит на його видалення, якщо такий знайдено.

Для сутності «Товар» було реалізовано 3 збережувальні процедури (лістинг А7, додаток А):

- `create_goods`;
- `update_goods`;
- `delete_goods`.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дат		

Збережувальна процедура `create_goods` приймає 5 параметрів: артикул товару, ідентифікатор магазину, в якому він знаходиться, назва, ціна і кількість. Після цього виконується перевірка наявності записів з артикулом нового товару.

Якщо збіг знайдено, то до поля, в якому зберігається кількість наявних продуктів буде додано кількість товарів з параметрів процедури. Якщо збіг не знайдено, то в таблицю буде додано новий запис.

Збережувальна процедура `update_goods` приймає 4 параметри: артикул товару, назву, ціну та кількість, екранує параметри і виконує підготований запит на оновлення даних.

Збережувальна процедура `delete_goods` приймає 1 параметр: артикул товару і виконує запит на його видалення.

Для реалізації реєстрації та авторизації користувачів в системі було додано таблицю `users`. Для того щоб забезпечити можливість взаємодії з нею без загрози SQL-ін'єкцій було реалізовано 3 збережувальні процедури (лістинг А8, додаток А):

- `create_user`;
- `change_user_pass`;
- `delete_user`.

Збережувальна процедура `create_user` приймає 4 вхідних параметри: ім'я користувача, номер телефону, електронна пошта, пароль. Далі йде перевірка наявності запису із полями, які б збігались зі значеннями нового запису. Якщо збігів не знайдено, то запис додається в таблицю після екранування параметрів.

Збережувальна процедура `change_user_pass` приймає 2 параметри: електронну пошту та новий пароль і оновлює запис після перевірки коректності даних.

Збережувальна процедура `delete_user` приймає 1 вхідний параметр: ідентифікатор користувача і виконує запит на його видалення.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дат		

Для фіксації покупок користувачів було створено зберезувальні процедури, які забезпечують введення та модифікацію даних у таблиці order_str (лістинг А9, додаток А):

- create_order_str;
- update_order_str;
- delete_order_str.

Зберезувальна процедура create_order_str приймає 5 параметрів: артикул товару, ідентифікатор клієнта, ціну за одиницю товару з урахуванням знижки, кількість товарів кожного виду, дату покупки. Далі виконується запит на додавання запису без додаткових перевірок, тому що всі змінні приведені до конкретних типів, які не дозволяють виконати SQL-ін'єкцію.

Зберезувальна процедура update_order_str приймає 6 параметрів: ідентифікатор запису про певну одиницю товару, артикул товару, ідентифікатор клієнта, ціна товару, кількість та дата. Запит виконується без додаткових перевірок з причин аналогічних попередньому випадку.

Зберезувальна процедура delete_order_str приймає 1 параметр: ідентифікатор запису і видаляє його.

3.3 Реалізація користувацького інтерфейсу системи

Для розмежування прав доступу користувачів було реалізовано реєстрацію, яка потребує мінімум персональної інформації від користувача. Сам процес реєстрації передбачає заповнення форми (лістинг Б1, додаток Б), яка складається з набору полів (рис. 3.12).

Форма реєстрації

Новий користувач

прізвище, ім'я

Номер телефону

номер телефону

Email-адреса

name@example.com

Пароль

Реєстрація

Рисунок 3.12 – Форма реєстрації

					ДП. КН 21.449.12.000 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дат		

Після реєстрації користувачу потрібно ввести у форму авторизації (лістинг Б2, додаток Б) електронну пошту і пароль вказані при реєстрації (рис. 3.13), після чого виконається запит до БД з метою перевірки наявності відповідного облікового запису.



Рисунок 3.13 – Форма авторизації

Після успішної авторизації користувачу надається доступ до функцій системи, які передбачає роботу з керівниками, магазинами, продавцями, клієнтами, товарами та підсумковими даними, які вичисляються на основі значень таблиць бази даних.

Віджет роботи з керівниками включає в себе 4 вкладки (лістинг Б2, додаток Б): «Додати», «Змінити», «Видалити» і «Таблиця». Кожна вкладка надає форму для виконання дії з базою даних.

Вкладка «Додати» надає можливість додавати нових керівників, для цього потрібно заповнити форму (лістинг Б4, додаток Б), яка зображена на рисунку 3.14 і надіслати дані натисканням кнопки «Додати запис».

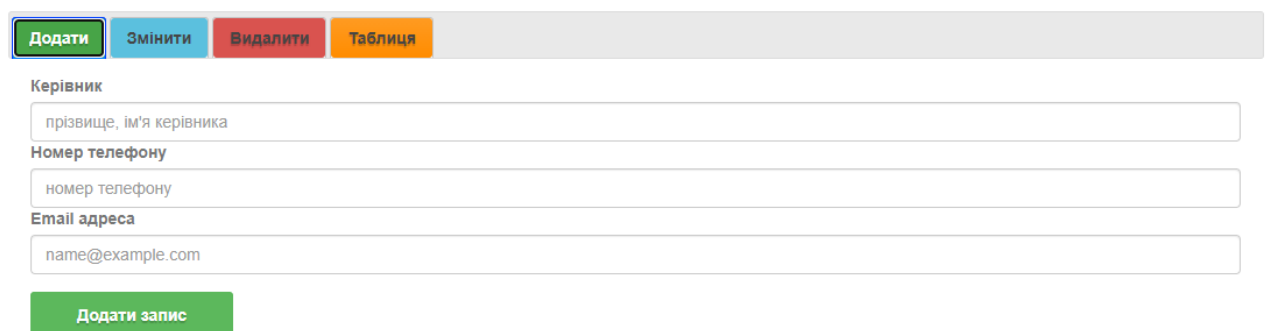


Рисунок 3.14 – Форма додавання керівників

Після натискання кнопки «Додати запис» дані буде надіслано скрипту (лістинг Б5, додаток Б), який обробляє значення полів форми і вносить їх у БД. При відправці даних оновлення сторінки не відбудеться завдяки

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		45

використанню технології AJAX при надсиланні форми (лістинг Б6, додаток Б), активною вкладкою автоматично стане «Таблиця» (лістинг Б7, додаток Б), де користувач зразу може переглянути внесені дані з поміткою «Новий» як на рисунку 3.15 [11].

Додати	Змінити	Видалити	Таблиця
id	Керівник	Телефон	Email
НОВИЙ	lead1	1111	post1@email

Рисунок 3.15 – Табличне подання записів БД

Після оновлення сторінки помітка «Новий» зникає і з’являються кнопки «Змінити» і «Видалити». При натисканні кнопки «Змінити» (лістинг Б8, додаток Б) біля запису таблиці як на рисунку 3.16 автоматично буде змінено активну вкладку на «Змінити».

Додати	Змінити	Видалити	Таблиця
id	Керівник	Телефон	Email
9	lead1	1111	post1@email
			Змінити Видалити

Рисунок 3.16 – Кнопки зміни і видалення записів

Форму «Змінити» (лістинг Б9, додаток Б) буде заповнено даними відповідного запису як показано на рисунку 3.17.

Додати	Змінити	Видалити	Таблиця
Ідентифікатор			
9			
Керівник			
lead1			
Номер телефону			
1111			
Email адреса			
post1@email			
Оновити запис			

Рисунок 3.17 - Форма зміни даних керівника

Після зміни даних і натискання кнопки «Оновити запис» виконується асинхронний обробник події надсилення форми (лістинг Б10, додаток Б) значення полів передаються скрипту (лістинг Б11, додаток Б), що їх обробляє і викликає відповідну зберезувальну процедуру. Оновлення сторінки теж не відбувається і дані в таблиці оновлюються автоматично. Для візуального виділення рядку таблиці задається інший фон як на рисунку 3.18, поки користувач не наведе курсор, після чого фон буде змінено на попередній.

Додати	Змінити	Видалити	Таблиця	
id	Керівник	Телефон	Email	
9	lead_edited	1111	post1@email	Змінити Видалити

Рисунок 3.18 – Результат зміни запису

При редагуванні записів неможливо змінювати значення поля «Ідентифікаторі», оскільки воно виступає ключем для пошуку відповідного запису і БД забороняє зміну первинних ключів.

Для видалення записів потрібно натиснути на кнопку «Видалити», після чого виконається обробник відповідної події (лістинг Б12, додаток Б) активною стане вкладка «Видалити» і форма заповниться автоматично як на рисунку 3.19.

Додати	Змінити	Видалити	Таблиця
Ідентифікатор			
9			
Керівник			
lead_edited			
Номер телефону			
1111			
Email адреса			
post1@email			
Видалити запис			

Рисунок 3.19 – Форма видалення записів керівників

Після натискання кнопки «Видалити запис» виконається асинхронний обробник події надсилення форми (лістинг Б13, додаток Б) і відповідному

					ДП. КН 21.449.12.000 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дат		

скрипту (лістинг Б14, додаток Б) буде передано ідентифікатор запису, який потрібно видалити і він викличе збережувальну процедуру, яка видалить запис. При надсиланні форми оновлення сторінки не відбудеться, а відповідний запис з таблиці буде виділено протягом 1 секунди і видалено як це показано на рисунку 3.20.

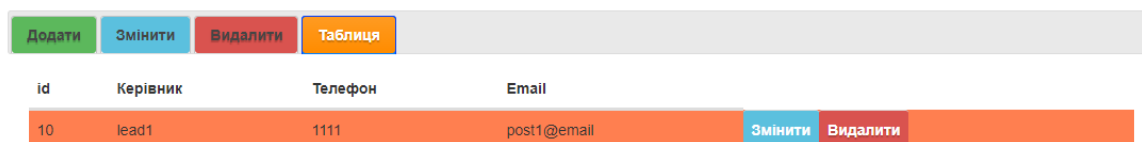


Рисунок 3.20 – Виділення запису при видаленні

Інтерфейси для роботи з іншими таблицями працюють аналогічним чином, але відрізняються наборами полів та логікою збережувальних процедур.

Віджет для роботи з магазинами складається з вкладок «Додати» (лістинг Б15, додаток Б), «Змінити» (лістинг Б16, додаток Б), «Видалити» (лістинг Б17, додаток Б) і «Таблиця» (лістинг Б18, додаток Б). Обробники подій натискання кнопок, додавання, зміни, видалення записів, відправки форм та виклику збережувальних процедур подано в лістингу Б19 додатку Б.

Віджет для роботи з продавцями включає такі вкладки: «Додати» (лістинг Б20, додаток Б), «Змінити» (лістинг Б21, додаток Б), «Видалити» (лістинг Б22, додаток Б) і «Таблиця» (лістинг Б23, додаток Б). Процедури обробки передбачених подій та виклику відповідних збережувальних процедур подано в лістингу Б24 додатку Б.

Віджет для роботи з клієнтами і товарами в свою структуру включають вкладки аналогічні вкладкам попередніх віджетів, розмітку та процедури обробки подій подано в лістингах Б25-29 додатку Б для першого і Б30-34 додатку Б для другого віджетів.

Віджет для здійснення покупок відрізняється своїм функціоналом, оскільки дає можливість додавати групу записів. Це було реалізовано за допомогою накопичення екземплярів сутності «Рядок замовлення». Сам процес накопичення відбувається при заповненні форми «Додати» і натисканні кнопки

					ДП. КН 21.449.12.000 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дат		

«Додати запис». При цьому всі одиниці товару та їх сумарну вартість буде відображено в таблиці як на рисунку 3.21.

Рисунок 3.21 Віджет для оформлення покупок

Додавання групи записів у БД відбувається після натискання кнопки «Купити», яке викликає AJAX-процедуру, що вносить нові записи у таблицю і передає масив об'єктів скрипту (лістинг 35, додаток Б). Всі віджети «вкладки» згруповано у віджет «акордеон» [12].

Для коректного відображення всіх елементів було використано каскадні таблиці стилів, а для його адаптації під різні розміри екранів було додано медіа-запити.

3.4. Тестування інформаційної системи

Для тестування збержувальних процедур було розроблено ряд запитів, які подано в лістингу В1 додатку В. Для перевірки коректності роботи ЗП create_lead її було викликано з передачею необхідних параметрів, як на рисунку 3.21 [13-15].

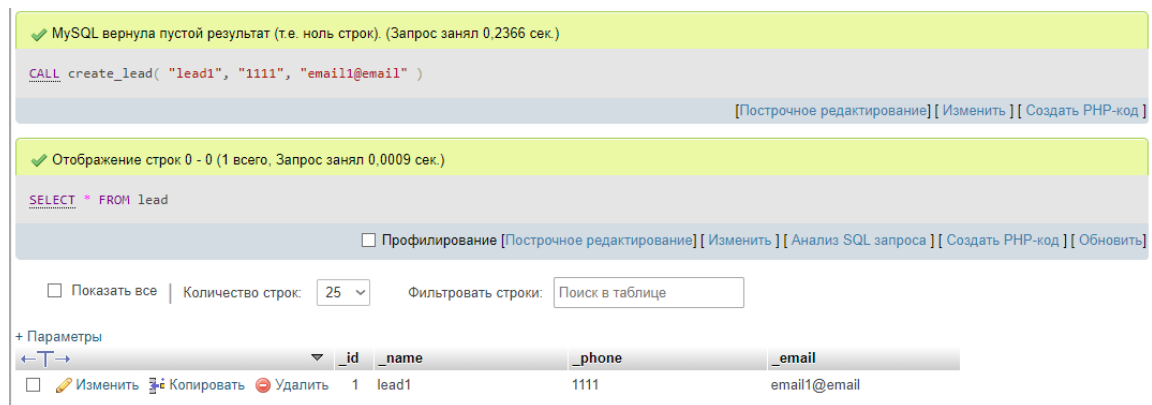


Рисунок 3.21 – Результат виконання ЗП create_lead

ЗП create_lead виконала перевірку та додавання нового запису. Далі було протестовано ЗП update_lead для оновлення існуючого запису (рис. 3.22).

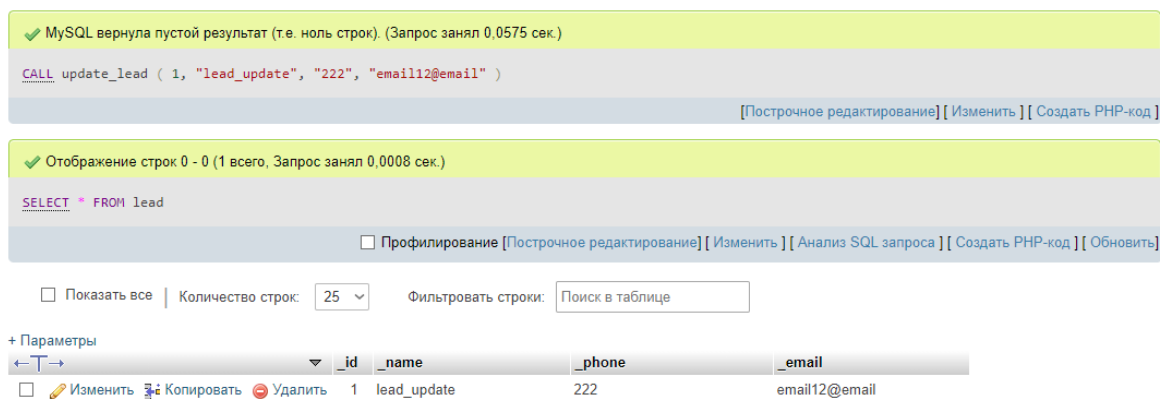


Рисунок 3.22 – Результат виконання ЗП update_lead

Запис було успішно оновлено після чого викликано ЗП для додавання запису про магазин з використанням ідентифікатора існуючого керівника, як на рисунку 3.23.

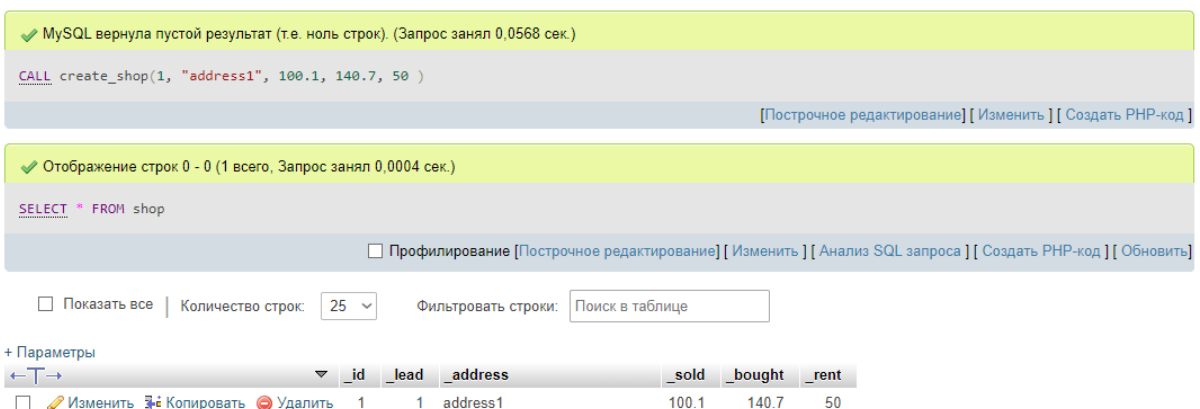


Рисунок 3.23 – Результат виконання ЗП create_shop

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дат		

Після успішного додавання запису про магазин було викликано ЗП для його оновлення з передачею параметрів поданих на рисунку 3.24.

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0524 сек.)

CALL update_shop(1, 1, "address_edited", 101.6, 130.9, 70)

Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

SELECT * FROM shop

id	lead	address	_sold	_bought	_rent
1	1	address_edited	101.6	130.9	70

Рисунок 3.24 – Результат виконання ЗП update_shop

Після оновлення запису про магазин було додано запис про продавця з використання ідентифікатора існуючого магазину та параметрами як на рисунку 3.25.

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,1523 сек.)

CALL create_seller (1, "seller1", "1111", "email1@gmail.com")

Отображение строк 0 - 0 (1 всего, Запрос занял 0,0008 сек.)

SELECT * FROM seller

id	_shop	_name	_phone	_email
1	1	seller1	1111	email1@gmail.com

Рисунок 3.25 – Результат виконання ЗП create_seller

Після додавання запису про продавця його було оновлено за допомогою ЗП update_seller за використанням параметрів поданих на рисунку 3.26.

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0495 сек.)

CALL update_seller (1, 1, "seller_edited", "2222", "email_edited@email")

Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

SELECT * FROM seller

id	_shop	_name	_phone	_email
1	1	seller_edited	2222	email_edited@email

Рисунок 3.26 – Результат виконання ЗП update_seller

Після оновлення запису про продавця було протестовано ЗП create_client з використання параметрів як на рисунку 3.27.

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0558 сек.)

```
CALL create_client ( "client1", "1111", "email@email" )
```

Отображение строк 0 - 0 (1 всего, Запрос занял 0,0008 сек.)

```
SELECT * FROM client
```

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

	id	name	phone	email
1	client1	1111	email@email	

Рисунок 3.27 – Результат виконання ЗП create_client

Після успішного додавання запису про клієнта було протестовано ЗП update_client з використанням параметрів як на рисунку 3.28.

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0615 сек.)

```
CALL update_client ( 1, "client_edited", "2222", "email_edited@email" )
```

Отображение строк 0 - 0 (1 всего, Запрос занял 0,0004 сек.)

```
SELECT * FROM client
```

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

	id	name	phone	email
1	client_edited	2222	email_edited@email	

Рисунок 3.28 – Результат виконання ЗП update_client

Далі було перевірено коректність роботи ЗП create_goods з передачею параметрів як на рисунку 3.29.

MySQL вернула пустой результат (т.е. ноль строк). (Запрос занял 0,0602 сек.)

```
CALL create_goods ( "000001", 1, "goods1", 10.5, 5 )
```

Отображение строк 0 - 0 (1 всего, Запрос занял 0,0005 сек.)

```
SELECT * FROM goods
```

Показать все | Количество строк: 25 | Фильтровать строки: Поиск в таблице

	article	shop	name	price	count
000001	1	goods1	10.5	5	

Рисунок 3.29 – Результат виконання ЗП create_goods

Після успішного додавання запису про товар було протестовано ЗП update_goods з параметрами, для оновлення запису як на рисунку 3.30.

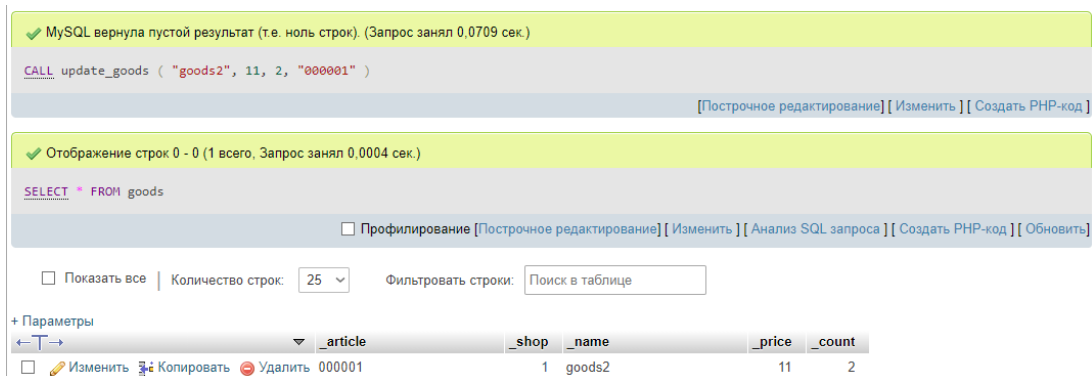


Рисунок 3.30 – Результат виконання ЗП update_goods

Далі було протестовано ЗП create_user з параметрами як на рисунку 3.31. Варто відзначити, що у випадку реєстрації користувача через відповідну форму в базу даних буде внесено захешований пароль і адміністратор бази даних не зможе його переглядати, або дешифрувати.

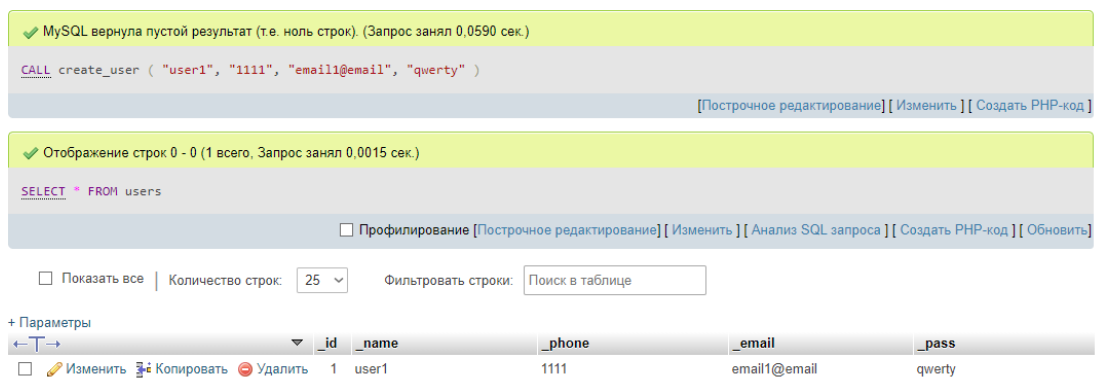


Рисунок 3.31 – Результат виконання ЗП create_user

В результаті тестування не було виявлено недоліків, які б могли завадити коректній роботі інформаційної системи.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ

4.1 Аналіз ринку

Програмне забезпечення інформаційної системи автоматизації обліку товарообігу (ІСАОТ) є сукупністю програмних засобів, які реалізують можливість структурованого зберігання даних, їх модифікацію, а також ведення обліку та обчислення певних показників підприємства. Вона побудована на принципах доступності та зручності користування. Розвиток інформаційних технологій дозволяє не лише створювати інформаційні системи, а й заробляти на цьому. Важливим моментом є легкість впровадження системи. В даному випадку всі необхідні дії зводяться до реєстрації в сервісі і уточнення предметної області. При цьому, ринком збуту є підприємства зацікавлені в автоматизації обліку та аналітики бізнес-процесів, які мають змогу використовувати мережу Internet, що надає можливість доступу до системи користувачам з різних куточків світу.

Конкуренція досить висока через велику актуальність проєктів даної тематики. Але серед основних недоліків є жорсткі рамки предметної області. В кращому випадку це вирішується списком доступних тематик, але все одно не вирішує проблему, яка виникає при потребі додавання специфічних обчислень. Крім того варто відзначити, що функціонал наявних аналогів доступний лише протягом короткого пробного періоду, а потім вимагає постійних грошових витрат. Саме ці причини забезпечують високий попит на нові інформаційні системи з більш гнучким функціоналом. При цьому обсяги продажу виробу обмежуються лише абстрактною кількістю користувачів, яких сервіс в даний час може обслуговувати та надавати підтримку.

Продукція конкурентів являє собою різноманітні програмні продукти, які можуть поширюватися у формі веб-ресурсів, або програмного забезпечення. Їх дизайн та функціонал відповідають вимогам користувача, але сильно обмежують їх жорсткістю визначення свого функціоналу.

					<i>ДП. КН 21.449.12.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		54

Життєвий цикл таких продуктів досить довготривалий і не прив'язується до часу існування конкретного підприємства, але може завершитись, якщо функціонал довго не буде розвиватись і перестане відповідати вимогам часу.

4.2 Розрахунок витрат на проєктування інформаційної системи

Сумарні затрати фінансів на соціальні потреби, прямі, накладні витрати, планові накопичення, кошторисну вартість, податок на додану вартість та договірну ціну подано в таблиці Г1 додатку Г. Зарплата проєктувальників було обчислено виходячи з їх кількості та тривалості роботи. В розробці даного проєкту брали участь особи, які займали посади керівника ДП, фронтенд-розробника, бекенд-розробника, тестувальника, DevOps-інженера та тімліда проєкту. Розрахунок заробітної плати проєктувальників та відрахувань було виконано відповідно до таблиці Г2 додатку Г, результати подано в таблицях Г3 – Г7 додатку Г.

4.3 Обґунтування необхідності розробки

Інформаційна система автоматизації обліку товарообігу розробляється для магазину косметики. Вона забезпечить автоматизацію основних бізнес-процесів пов'язаних з інвентаризацією та веденням статистики по певних показниках. Головний ефект полягає у зменшенні ресурсних затрат працівниками, а зокрема – часу. Це дасть можливість більш ефективно виконувати ту ж роботу за коротші терміни, що у свою чергу забезпечить хороші передумови для збільшення кількісних показників підприємства і не потребуватиме додаткових витрат. Зокрема, щомісячна економія сягатиме розмірів рівноцінних оплаті аналогічних продуктів від інших виробників і сприятиме оптимальнішому розподілу часових ресурсів, які неможливо виразити у грошовому еквіваленті без додаткових статистичних даних. Отже, основним ефектом є автоматизація рутинних процесів та підвищення загального рівня ефективності.

					ДП. КН 21.449.12.000 ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дат		

ВИСНОВКИ

В результаті дипломного проєктування було реалізовано інформаційну систему автоматизації обліку товарообігу, проведено аналіз ринку аналогів, який довів актуальність створення продукту, засобів реалізації. Також було детально досліджено предметну область, в результаті чого було виділено інформаційні об'єкти-сутності, які стали основою для подальшої розробки проєкту. Також було розглянуто системи управління базами даних і засоби реалізації.

За допомогою інструментів. PHP, JQuery-UI, Bootstrap та MySQL було реалізовано базу даних та адаптивний веб-інтерфейс, які входять до складу інформаційної системи. Розроблений проєкт містить набір інструментів, який забезпечує виконання запроєктованих функцій і дає можливість в майбутньому реалізувати розширення функціоналу, а саме:

- розробка дискретного інтерфейсу для взаємодії з БД MySQL;
- збір статистики про використання функцій сайту;
- систему бронювання товарів;
- підключення API нової пошти для відстеження відправок та отримань товарів;
- система навчання для нових користувачів тощо.

Отже, результатом дипломного проєктування стала повноцінна масштабована інформаційна система, яка відповідає поставленим вимогам, виконує запроєктовані функції та дає можливість для подальшої розробки та розширення функціоналу.

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		56

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Топ 10 СУБД. *Proglib*: веб-сайт. URL: <https://proglib.io/p/databases2019> (дата звернення: 02.03.2021).
2. Проєктування бази даних – Інформаційні системи та технології в банках – Навчальні матеріали онлайн. *Навчальні матеріали онлайн*: веб-сайт. URL: https://pidru4niki.com/11570718/bankivska_sprava/proektuvannya_baz_danih (дата звернення: 09.03.2021).
3. Возможности phpMyAdmin. *phpMyAdmin*: веб-сайт. URL: <https://php-myadmin.ru/doc/ability/> (дата звернення 17.03.2021).
4. Делаем сам: адаптивный сайт. *КОД*: веб-сайт. URL: <https://thecode.media/bootstrap/> (дата звернення: 01.04.2021).
5. Для чего хорош JavaScript. *Tproger*: веб-сайт. URL: <https://tproger.ru/articles/what-javascript-is-good-for/> (дата звернення: 14.04.2021).
6. 8 лучших локальных серверов. *КОМЬЮНИТИ*: веб-сайт. URL: <https://timeweb.com/ru/community/articles/luchshie-lokalnye-servery> (дата звернення: 25.04.2021).
7. Продуктивное использование PHPStorm. *Хабр*: веб-сайт. URL: <https://habr.com/ru/post/157409/> (дата звернення: 07.05.2021).
8. Зачем нужен PHP. *Хабр*: веб-сайт. URL: <https://qna.habr.com/q/468462> (дата звернення 10.00.2021).
9. Введение в хранимые процедуры. *RUSELLER.COM*: веб-сайт. URL: <https://ruseller.com/lessons.php?id=1189> (дата звернення 21.05.2021).
10. Саган О.А. Особливості розробки та адміністрування програмного забезпечення на базі клієнт-серверної архітектури. *Збірник наукових тез*: за матеріалами наукових читань. Тернопіль: Навчально-ропктична майстерня редакційно-видавничих технологій Галицького коледжу імені В'ячеслава чорновола, 2021р., С. 197-201.

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		57

11. AJAX – Руководство Web-разработчика | MDN. *MDN Web Docs*: веб-сайт. URL: <https://developer.mozilla.org/ru/docs/Web/Guide/AJAX> (дата звернення 29.05.2021).

12. jQuery-UI – Учебник. *jQuery-UI*: веб-сайт. URL: <http://www.wisdomweb.ru/JQ/ui.php> (дата звернення 16.06.2021).

13. CSS3 Tutorial. *W3school*: веб-сайт. URL: <https://www.w3schools.com/css/> (дата звернення 00.00.2021).

14. Медиа-запросы. *htmlbook*: веб-сайт. URL: <http://htmlbook.ru/css/value/media> (дата звернення 16.06.2021).

15. Тестирование базы данных: что нужно проверить в первую очередь. *TestMaticK*: веб-сайт. URL: <https://testmatick.com/ru/testirovanie-bazy-dannyh-cto-nuzhno-proverit-v-pervuyu-ochered/> (дата звернення 16.06.2021).

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дат		

ДОДАТКИ

Додаток А

Запити на створення об'єктів бази даних

Лістинг А1 – Запит на створення таблиць

```
DROP DATABASE IF EXISTS unice;  
CREATE DATABASE unice;  
USE unice;
```

```
CREATE TABLE client  
(  
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  _name VARCHAR(50) NOT NULL,  
  _phone VARCHAR(10) NULL,  
  _email VARCHAR(30) NULL  
)ENGINE=InnoDB;
```

```
CREATE TABLE lead  
(  
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  _name VARCHAR(50) NULL,  
  _phone VARCHAR(10) NULL,  
  _email VARCHAR(50) NULL  
)ENGINE=InnoDB;
```

```
CREATE TABLE shop  
(  
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  _lead INT NULL,  
  _address VARCHAR(50) NOT NULL,  
  _sold FLOAT NULL,  
  _bought FLOAT NULL,  
  _rent FLOAT NULL  
)ENGINE=InnoDB;
```

```
CREATE TABLE seller  
(  
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
  _shop INT NOT NULL,  
  _name VARCHAR(50) NOT NULL,  
  _phone VARCHAR(10) NULL,  
  _email VARCHAR(50) NULL  
)ENGINE=InnoDB;
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу A1

```
CREATE TABLE goods
(
  _article VARCHAR(6) NOT NULL PRIMARY KEY,
  _shop INT NULL,
  _name VARCHAR(50) NOT NULL,
  _price FLOAT NULL,
  _count INT NULL
)ENGINE=InnoDB;

CREATE TABLE order_str
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _article VARCHAR(6) NOT NULL,
  _client INT NULL,
  _price FLOAT NULL,
  _count INT NOT NULL,
Продовження лістингу A1
  _date DATE NULL
)ENGINE=InnoDB;

CREATE TABLE users
(
  _id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  _name VARCHAR(50) NOT NULL,
  _phone VARCHAR(10) NULL,
  _email VARCHAR(30) NULL,
  _pass VARCHAR(32) NULL
)ENGINE=InnoDB;
```

Лістинг A2 – Запит на створення ключів, індексів та унікальних полів

```
ALTER TABLE shop ADD UNIQUE (_id);
ALTER TABLE seller ADD UNIQUE (_id);
ALTER TABLE goods ADD UNIQUE (_article);
ALTER TABLE lead ADD UNIQUE (_id);
ALTER TABLE client ADD UNIQUE (_id);

ALTER TABLE shop ADD INDEX (_lead);
ALTER TABLE seller ADD INDEX (_shop);
ALTER TABLE goods ADD INDEX (_shop);
ALTER TABLE order_str ADD INDEX (_client);

ALTER TABLE shop
ADD FOREIGN KEY (_lead)
REFERENCES lead (_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу А2

```
ALTER TABLE seller
ADD FOREIGN KEY (_shop)
REFERENCES shop (_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;

ALTER TABLE goods
ADD FOREIGN KEY (_shop)
REFERENCES shop (_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;

ALTER TABLE order_str
ADD FOREIGN KEY (_client)
REFERENCES client(_id)
ON DELETE CASCADE
ON UPDATE CASCADE ;
```

Лістинг А3 – Запит на створення зберезувальних процедур lead

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_lead//
CREATE PROCEDURE create_lead ( IN name VARCHAR(50), IN phone
VARCHAR(10), IN email VARCHAR(50) )
BEGIN
    IF NOT EXISTS(SELECT 1 FROM lead WHERE lead._name = name) AND
        NOT EXISTS(SELECT 1 FROM lead WHERE lead._phone =
phone) AND
        NOT EXISTS(SELECT 1 FROM lead WHERE lead._email =
email) THEN
        SET @s = 'INSERT INTO lead (lead._name, lead._phone,
lead._email) VALUES(?, ?, ?)';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING name, phone, email;
    END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_lead//
CREATE PROCEDURE update_lead ( IN id INT, IN name
VARCHAR(50), IN phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
    IF NOT EXISTS(SELECT 1 FROM lead WHERE lead._name = name) OR
        NOT EXISTS(SELECT 1 FROM lead WHERE lead._phone =
phone) OR NOT EXISTS(SELECT 1 FROM lead WHERE lead._email = email)
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу А3

```
AND EXISTS(SELECT 1 FROM lead WHERE lead._id = id) THEN
  SET @s = 'UPDATE lead SET _name=?, _phone=?, _email=? WHERE
lead._id=?';
      PREPARE stmt FROM @s;
EXECUTE stmt USING name, phone, email, id;
  END IF;
END //
```

```
DELIMITER //
DROP PROCEDURE IF EXISTS delete_lead//
CREATE PROCEDURE delete_lead (IN id INT)
BEGIN
  IF EXISTS(SELECT 1 FROM lead WHERE lead._id = id) THEN
    DELETE FROM lead WHERE lead._id = id;
  END IF;
END //
```

Лістинг А4 – запит на створення зберезувальних процедур shop

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_shop//
CREATE PROCEDURE create_shop ( IN lead INT, IN address
VARCHAR(50), IN sold FLOAT, IN bought FLOAT, IN rent FLOAT )
BEGIN
  IF NOT EXISTS(SELECT 1 FROM shop WHERE shop._address =
address) THEN
    SET @s = 'INSERT INTO shop (shop._lead, shop._address,
shop._sold, shop._bought, shop._rent) VALUES(?, ?, ?, ?, ?)';
    PREPARE stmt FROM @s;
EXECUTE stmt USING lead, address, sold, bought, rent;
  END IF;
END //
```

```
DELIMITER //
DROP PROCEDURE IF EXISTS update_shop//
CREATE PROCEDURE update_shop ( IN id INT, IN lead INT, IN
address VARCHAR(50), IN sold FLOAT, IN bought FLOAT, IN rent
FLOAT )
BEGIN
  IF NOT EXISTS(SELECT 1 FROM shop WHERE shop._address =
address) AND
    EXISTS(SELECT 1 FROM shop WHERE shop._id = id) THEN
    SET @s = 'UPDATE shop SET _lead=?, _address=?, _sold=?,
_bought=?, _rent=? WHERE _id=?';
    PREPARE stmt FROM @s;
EXECUTE stmt USING lead, address, sold, bought, rent, id;
  END IF;
END //
```

```
DELIMITER //
DROP PROCEDURE IF EXISTS delete_shop//
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу А4

```
CREATE PROCEDURE delete_shop (IN id INT)
BEGIN
    IF EXISTS(SELECT 1 FROM shop WHERE shop._id = id) THEN
        DELETE FROM shop WHERE shop._id = id;
    END IF;
END //
```

Лістинг А5 – запит на створення зберезувальних процедур seller

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_seller//
CREATE PROCEDURE create_seller ( IN shop INT, IN name
VARCHAR(50), IN phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
    IF NOT EXISTS(SELECT 1 FROM seller WHERE seller._name = name)
AND
        NOT EXISTS(SELECT 1 FROM seller WHERE seller._phone =
phone) AND
        NOT EXISTS(SELECT 1 FROM seller WHERE seller._email =
email) THEN
        SET @s = 'INSERT INTO seller (seller._shop, seller._name,
seller._phone, seller._email) VALUES(?, ?, ?, ?)';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING shop, name, phone, email;
    END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_seller//
CREATE PROCEDURE update_seller ( IN id INT, IN shop INT, IN
name VARCHAR(50), IN phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
    IF NOT EXISTS(SELECT 1 FROM seller WHERE seller._shop = shop)
OR
        NOT EXISTS(SELECT 1 FROM seller WHERE seller._name =
name) OR
        NOT EXISTS(SELECT 1 FROM seller WHERE seller._phone
= phone) OR
        NOT EXISTS(SELECT 1 FROM seller WHERE seller._email
= email) AND
        EXISTS(SELECT 1 FROM seller WHERE seller._id = id)
THEN
        SET @s = 'UPDATE seller SET _shop=?, _name=?, _phone=?,
_email=? WHERE seller._id=?';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING shop, name, phone, email, id;
    END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS delete_seller//
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу А5

```
CREATE PROCEDURE delete_seller (IN id INT)
BEGIN
    IF EXISTS(SELECT 1 FROM seller WHERE seller._id = id)
THEN
        DELETE FROM seller WHERE seller._id = id;
    END IF;
END //
```

Лістинг А6 – запит на створення зберезувальних процедур client

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_client//
CREATE PROCEDURE create_client ( IN name VARCHAR(50), IN
phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
    IF NOT EXISTS(SELECT 1 FROM client WHERE client._name = name)
AND
        NOT EXISTS(SELECT 1 FROM client WHERE client._phone =
phone) AND
        NOT EXISTS(SELECT 1 FROM client WHERE client._email =
email) THEN
        SET @s = 'INSERT INTO client ( client._name, client._phone,
client._email) VALUES(?, ?, ?)';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING name, phone, email;
    END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_client//
CREATE PROCEDURE update_client ( IN id INT, IN name
VARCHAR(50), IN phone VARCHAR(10), IN email VARCHAR(50) )
BEGIN
    IF NOT EXISTS(SELECT 1 FROM client WHERE client._name = name)
OR
        NOT EXISTS(SELECT 1 FROM client WHERE client._phone =
phone) OR
        NOT EXISTS(SELECT 1 FROM client WHERE client._email =
email) AND
        EXISTS(SELECT 1 FROM client WHERE client._id = id)
THEN
        SET @s = 'UPDATE client SET _name=?, _phone=?, _email=?
WHERE client._id=?';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING name, phone, email, id;
    END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS delete_client//
CREATE PROCEDURE delete_client (IN id INT)
```

					ДП. КН 21.449.12.000 ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу А6

```
BEGIN
    IF EXISTS(SELECT 1 FROM client WHERE client._id = id)
THEN
    DELETE FROM client WHERE client._id = id;
    END IF;
END //
```

Лістинг А7 – запит на створення зберезувальних процедур goods

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_goods//
CREATE PROCEDURE create_goods ( IN article VARCHAR(6), IN
shop INT, IN name VARCHAR(50), IN price FLOAT, IN counter INT )
BEGIN
    IF NOT EXISTS(SELECT 1 FROM goods WHERE goods._article =
article) THEN
        SET @s = 'INSERT INTO goods (goods._article, goods._shop,
goods._name, goods._price, goods._count) VALUES(?, ?, ?, ?, ?)';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING article, shop, name, price, counter;

    ELSEIF EXISTS(SELECT 1 FROM goods WHERE goods._article =
article) THEN
        UPDATE goods SET _count = _count + counter;
    END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_goods//
CREATE PROCEDURE update_goods (IN name VARCHAR(50), IN price
FLOAT, IN counter INT, IN article VARCHAR(6) )
BEGIN
    IF EXISTS(SELECT 1 FROM goods WHERE goods._article =
article) THEN
        SET @s = 'UPDATE goods SET goods._name=?, goods._price=?,
goods._count=? WHERE goods._article = ?';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING name, price, counter, article;
    END IF;
END //

DELIMITER //
DROP PROCEDURE IF EXISTS delete_goods//
CREATE PROCEDURE delete_goods (IN article VARCHAR(6))
BEGIN
    IF EXISTS(SELECT 1 FROM goods WHERE goods._article =
article) THEN
        DELETE FROM goods WHERE goods._article = article;
    END IF;
END //
```

Лістинг А8 – запит на створення зберезувальних процедур users

```
DELIMITER //
DROP PROCEDURE IF EXISTS create_user//
CREATE PROCEDURE create_user ( IN name VARCHAR(50), IN phone
VARCHAR(10), IN email VARCHAR(50), IN pass VARCHAR(32) )

BEGIN

    IF NOT EXISTS(SELECT 1 FROM users WHERE users._name = name)
AND
        NOT EXISTS(SELECT 1 FROM users WHERE users._phone =
phone) AND
        NOT EXISTS(SELECT 1 FROM users WHERE users._email =
email) THEN
        SET @s = 'INSERT INTO users (users._name, users._phone,
users._email, users._pass) VALUES(?, ?, ?, ?)';
        PREPARE stmt FROM @s;

        EXECUTE stmt USING name, phone, email, pass;
        END IF;
    END //

DELIMITER //
DROP PROCEDURE IF EXISTS change_user_pass//
CREATE PROCEDURE change_user_pass (IN email VARCHAR(50), IN
pass VARCHAR(32) )

BEGIN
    IF EXISTS(SELECT 1 FROM users WHERE users._email = email)
THEN

        SET @s = 'UPDATE users SET _pass = ? WHERE users._email=?';
        PREPARE stmt FROM @s;
        EXECUTE stmt USING pass, email;
        END IF;
    END //

DELIMITER //

DROP PROCEDURE IF EXISTS delete_user//

CREATE PROCEDURE delete_user (IN id INT)

BEGIN
    IF EXISTS(SELECT 1 FROM users WHERE users._id = id) THEN
        DELETE FROM users WHERE users._id = id;
    END IF;
END //
```


Лістинг А9 – запит на створення зберезувальних процедур order_str

```

DELIMITER //
DROP PROCEDURE IF EXISTS create_order_str//
CREATE PROCEDURE create_order_str ( IN article VARCHAR(6), IN
client INT, IN price FLOAT, IN counter INT, IN date DATE )

BEGIN

INSERT INTO order_str (order_str._article, order_str._client,
order_str._price, order_str._count, order_str._date ) VALUES(
article, client, price, counter, date );

END //

DELIMITER //
DROP PROCEDURE IF EXISTS update_order_str//

CREATE PROCEDURE update_order_str (IN id INT, IN article
VARCHAR(6), IN client INT, IN price FLOAT, IN counter VARCHAR(50),
IN date DATE )

BEGIN

UPDATE order_str SET order_str._article=article,
order_str._client=client, order_str._price=price,
order_str._count=counter, order_str._date=date WHERE order_str._id
= id;
END //

DELIMITER //

DROP PROCEDURE IF EXISTS delete_order_str//
CREATE PROCEDURE delete_order_str (IN id INT)

BEGIN

DELETE FROM order_str WHERE order_str._id = id;

END //

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		67

Додаток Б

Розмітка та реалізація функціоналу

Лістинг Б – розмітка форми реєстрації

```
<form action="forms/user/create_user.php" method="POST"
id="create_user_form">
  <div class="form-group">
    <h4>Форма реєстрації</h4>
    <label>Новий користувач</label>
    <input type="text" class="form-control"
name="user_name_create" id="user_name_create"
placeholder="прізвище, ім'я" required>
    <label>Номер телефону</label>
    <input type="phone" class="form-control"
name="user_phone_create" id="user_phone_create" placeholder="номер
телефону" required>
    <label>Email-адреса</label>
    <input type="email" class="form-control"
name="user_email_create" id="user_email_create"
placeholder="name@example.com" required>
    <label>Пароль</label>
    <input type="password" class="form-control"
name="user_pass_create" id="user_pass_create" required>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-success mb-2" name="btn-
user-create" id="btn-user-create">Реєстрація</button>
  </div>
</form>
```

Лістинг Б2 – розмітка форми авторизації

```
<div class="col">
  <form action="forms/user/auth.php" method="POST"
id="auth_user_form">
    <div class="form-group">
      <h4>Форма авторизації</h4>
      <label>Email-адреса</label>
      <input type="email" class="form-control"
name="user_email_auth" id="user_email_auth"
placeholder="email@example.com">
      <label>Пароль</label>
      <input type="password" class="form-control"
name="user_pass_auth" id="user_pass_auth" placeholder="введіть
пароль">
    </div>
    <div class="form-group">
      <button type="submit" class="btn btn-success mb-2" name="btn-
user-auth" id="btn-user-auth">Вхід</button>
    </div>
```

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		68

```
</form>
</div>
```

Лістинг Б3 – розмітка віджета роботи з керівниками

```
<div class="tabs" id="tabs_lead" style="display: table;
width: 100%;">
  <ul>
    <li><a href="#tabs-1" class="btn btn-success">Додати</a></li>
    <li><a href="#tabs-2" class="btn btn-info">Змінити</a></li>
    <li><a href="#tabs-3" class="btn btn-
danger">Видалити</a></li>
    <li><a href="#tabs-4" class="btn btn-
primary">Таблиця</a></li>
  </ul>
  <div id="tabs-1">
    <?php require_once('forms/lead/create.php'); ?>
  </div>
  <div id="tabs-2">
    <?php require_once('forms/lead/update.php'); ?>
  </div>
  <div id="tabs-3">
    <?php require_once('forms/lead/delete.php'); ?>
  </div>
  <div id="tabs-4">
    <?php require_once('forms/lead/table_lead.php'); ?>
  </div>
</div>
```

Лістинг Б4 – розмітка форми додавання запису про керівника

```
<form method="POST" id="create_lead_form">
  <div class="form-group">
    <label>Керівник</label>
    <input type="text" class="form-control"
name="lead_name_create" id="lead_name_create"
placeholder="прізвище, ім'я керівника" required>
    <label>Номер телефону</label>
    <input type="phone" class="form-control"
name="lead_phone_create" id="lead_phone_create" placeholder="номер
телефону" required>
    <label>Email адреса</label>
    <input type="email" class="form-control"
name="lead_email_create" id="lead_email_create"
placeholder="name@example.com" required>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-success mb-2"
name="btn-lead-create" id="btn-lead-create">Додати запис</button>
  </div>
</form>
```

Лістинг Б5 – код скрипта для створення запису про керівника

```
<?php
    require_once("../../database/db_lead.php");
    $name = trim($_POST["lead_name_create"]);
    $phone = trim($_POST["lead_phone_create"]);
    $email = trim($_POST["lead_email_create"]);
    $insert = "CALL create_lead('$name', '$phone',
'$email')";
    $run = mysqli_query($conn,$insert);
?>
```

Лістинг Б6 – асинхронна функція додавання запису про керівника

```
$("#create_lead_form").submit(function() {
$( "#tabs_lead" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/lead/create_lead.php",
    data: $(this).serialize()
}).done(function() {

    let lead_name = $("#lead_name_create").val();
    let lead_phone = $("#lead_phone_create").val();
    let lead_email = $("#lead_email_create").val();
    $('#lead_table').append('<tr><td class="btn btn-
success">НОВИЙ</td><td>'+lead_name+'</td><td>'+lead_phone+'</td><t
d>'+lead_email+'</td></tr>');
});
return false;
});
```

Лістинг Б7 – скрипт для табличного подання записів про керівників

```
<table id="lead_table" class="table container justify-
content-center">
    <thead>
        <tr>
            <th>id</th>
            <th>Керівник</th>
            <th>Телефон</th>
            <th>Email</th>
        </tr>
    </thead>
<?php
    $select_lead = "SELECT * FROM lead";
    $run = mysqli_query($conn,$select_lead);
    while ($row = mysqli_fetch_array($run)){
?>
        <tr id="<?php echo $row['_id'].'lead_tr'; ?>" class="out-
tr">
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б7

```
<td id="<?php echo $row['_id'].'lead_id'; ?>"><?php echo  
$row['_id']; ?></td>  
    <td id="<?php echo $row['_id'].'lead_name'; ?>"><?php  
echo $row['_name']; ?></td>  
    <td id="<?php echo $row['_id'].'lead_phone';  
?>"><?php echo $row['_phone']; ?></td>  
    <td id="<?php echo $row['_id'].'lead_email';  
?>"><?php echo $row['_email']; ?></td>  
    <td id="<?php echo $row['_id'].'lead_e'; ?>"  
class="btn lead_editor btn-info">Змінити</td>  
    <td id="<?php echo $row['_id'].'lead_d'; ?>"  
class="btn lead_deleter btn-danger">Видалити</td>  
</tr>  
<?php } ?>  
</table>
```

Лістинг Б8 – обробник події натискання кнопки оновлення запису

```
$(".lead_editor").on('click', function () {  
    $( "#tabs_lead" ).tabs({  
        active: 1  
    });  
    let lead_id = parseInt ( $(this).attr("id") );  
    let lead_name = $("#"+lead_id+"lead_name").html();  
    let lead_phone = $("#"+lead_id+"lead_phone").html();  
    let lead_email = $("#"+lead_id+"lead_email").html();  
    $("#lead_id_update").val(lead_id);  
    $("#lead_name_update").val(lead_name);  
    $("#lead_phone_update").val(lead_phone);  
    $("#lead_email_update").val(lead_email);  
});
```

Лістинг Б9 – розмітка форми оновлення запису про керівника

```
<form method="POST" id="update_lead_form">  
    <div class="form-group">  
        <label>Ідентифікатор</label>  
        <input type="text" class="form-control"  
name="lead_id_update" id="lead_id_update"  
placeholder="ідентифікатор керівника" required readonly>  
        <label>Керівник</label>  
        <input type="text" class="form-control"  
name="lead_name_update" id="lead_name_update"  
placeholder="прізвище, ім'я керівника" required>  
        <label>Номер телефону</label>  
        <input type="phone" class="form-control"  
name="lead_phone_update" id="lead_phone_update" placeholder="номер  
телефону" required>  
        <label>Email адреса</label>
```

					ДП. КН 21.449.12.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б9

```
<input type="email" class="form-control"
name="lead_email_update" id="lead_email_update"
placeholder="name@example.com" required>
</div>
<div class="form-group">
    <button type="submit" class="btn btn-info mb-2"
name="btn-lead-update" id="btn-lead-update">Оновити запис</button>
</div>
</form>
```

Лістинг – Б10 асинхронна функція оновлення запису про керівника

```
$("#update_lead_form").submit(function() {
$( "#tabs_lead" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/lead/update_lead.php",
    data: $(this).serialize()
}).done(function() {
    let lead_id = $("#lead_id_update").val();
    let lead_name = $("#lead_name_update").val();
    let lead_phone = $("#lead_phone_update").val();
    let lead_email = $("#lead_email_update").val();
    $("#"+lead_id+"lead_tr").attr("bgcolor", "FAF0BE");
    $("#"+lead_id+"lead_name").html(lead_name);
    $("#"+lead_id+"lead_phone").html(lead_phone);
    $("#"+lead_id+"lead_email").html(lead_email);

    //$('#lead_table').append('<tr><td>'+lead_id+'</td><td>'+lead
_name+'</td><td>'+lead_phone+'</td><td>'+lead_email+'</td><td
class="btn btn-info">ЗМІНИТИ</td></tr>');
});
return false;
});
```

Лістинг Б11 – скрипт оновлення запису про керівника

```
<?php
require_once("../../database/db_lead.php");
$id = $_POST['lead_id_update'];
$name = $_POST['lead_name_update'];
$phone = $_POST['lead_phone_update'];
$email = $_POST['lead_email_update'];
$update = "CALL update_lead('$id', '$name', '$phone',
'$email')";
$run = mysqli_query($conn,$update);
?>
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дат		

Лістинг Б12 – обробник події натискання кнопки видалення запису

```
$( ".lead_deleter" ).on( 'click', function () {
$( "#tabs_lead" ).tabs({
    active: 2
});
let lead_id = parseInt ( $(this).attr("id") );
let lead_name = $( "#"+lead_id+"lead_name" ).html();
let lead_phone = $( "#"+lead_id+"lead_phone" ).html();
let lead_email = $( "#"+lead_id+"lead_email" ).html();
$( "#lead_id_delete" ).val(lead_id);
$( "#lead_name_delete" ).val(lead_name);
$( "#lead_phone_delete" ).val(lead_phone);
$( "#lead_email_delete" ).val(lead_email);
});
```

Лістинг Б13 – асинхронна функція видалення запису про керівника

```
$( "#delete_lead_form" ).submit(function() {
$( "#tabs_lead" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/lead/delete_lead.php",
    data: $(this).serialize()
}).done( function () {
    let lead_id = $( "#lead_id_delete" ).val();
    $( "#"+lead_id+"lead_tr" ).attr("bgcolor", "FF7F50");
    setTimeout(function() {
        $( "#"+lead_id+"lead_tr" ).remove();
    }, 1000);
});
return false;
});
```

Лістинг Б14 – скрипт видалення запису про керівника

```
<?php
require_once("../../database/db_lead.php");
$id = $_POST['lead_id_delete'];
$delete = "CALL delete_lead('$id')";
$run = mysqli_query($conn,$delete); //if run true
?>
```

Лістинг Б15 – розмітка форми додавання запису про магазин

```
<form method="POST" id="create_shop_form">
    <div class="form-group">
        <label>Керівник</label>
        <input type="text" class="form-control"
name="shop_lead_create" id="shop_lead_create"
placeholder="ідентифікатор керівника" required>
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б15

```

<label>Адреса</label>
  <input type="text" class="form-control"
name="shop_address_create" id="shop_address_create"
placeholder="адреса магазину" required>
  <label>Продано</label>
  <input type="text" class="form-control"
name="shop_sold_create" id="shop_sold_create" placeholder="сумарна
вартість проданих товарів">
  <label>Закуплено</label>
  <input type="text" class="form-control"
name="shop_bought_create" id="shop_bought_create"
placeholder="сумарна вартість закуплених товарів">
  <label>Оренда</label>
  <input type="text" class="form-control"
name="shop_rent_create" id="shop_rent_create"
placeholder="вартість оренди">
</div>
<div class="form-group">
  <button type="submit" class="btn btn-success mb-2"
name="btn-shop-create" id="btn-shop-create">Додати запис</button>
</div>
</form>

```

Лістинг Б16 розмітка форми оновлення запису про магазин

```

<form method="POST" id="update_shop_form">
  <div class="form-group">
    <label>Ідентифікатор</label>
    <input type="text" class="form-control"
name="shop_id_update" id="shop_id_update"
placeholder="ідентифікатор магазину" required readonly>
    <label>Керівник</label>
    <input type="text" class="form-control"
name="shop_lead_update" id="shop_lead_update"
placeholder="ідентифікатор керівника" required readonly>
    <label>Адреса</label>
    <input type="text" class="form-control"
name="shop_address_update" id="shop_address_update"
placeholder="адреса магазину" required>
    <label>Продано</label>
    <input type="text" class="form-control"
name="shop_sold_update" id="shop_sold_update" placeholder="сумарна
вартість проданих товарів">
    <label>Закуплено</label>
    <input type="text" class="form-control"
name="shop_bought_update" id="shop_bought_update"
placeholder="сумарна вартість закуплених товарів">
    <label>Оренда</label>
    <input type="text" class="form-control"
name="shop_rent_update" id="shop_rent_update"
placeholder="вартість оренди">

```

					ДП. КН 21.449.12.000 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б15

```
</div>
    <div class="form-group">
        <button type="submit" class="btn btn-info mb-2"
name="btn-shop-update" id="btn-shop-update">Оновити запис</button>
    </div>
</form>
```

Лістинг Б17 розмітка форми видалення запису про магазин

```
<form method="POST" id="delete_shop_form">
    <div class="form-group">
        <label>Ідентифікатор</label>
        <input type="text" class="form-control"
name="shop_id_delete" id="shop_id_delete"
placeholder="ідентифікатор магазину" readonly>
        <label>Керівник</label>
        <input type="text" class="form-control"
name="shop_lead_delete" id="shop_lead_delete"
placeholder="ідентифікатор керівника" readonly>
        <label>Адреса</label>
        <input type="text" class="form-control"
name="shop_address_delete" id="shop_address_delete"
placeholder="адреса магазину" readonly>
        <label>Продано</label>
        <input type="text" class="form-control"
name="shop_sold_delete" id="shop_sold_delete" placeholder="сумарна
вартість проданих товарів" readonly>
        <label>Закуплено</label>
        <input type="text" class="form-control"
name="shop_bought_delete" id="shop_bought_delete"
placeholder="сумарна вартість закуплених товарів" readonly>
        <label>Оренда</label>
        <input type="text" class="form-control"
name="shop_rent_delete" id="shop_rent_delete"
placeholder="вартість оренди" readonly>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-danger mb-2"
name="btn-shop-delete" id="btn-shop-delete">Видалити
запис</button>
    </div>
</form>
```

Лістинг Б18 скрипт для табличного подання записів про магазини

```
<table id="shop_table" class="table container justify-
content-center">
    <thead>
        <tr>
            <th>id</th>
            <th>Керівник</th>
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		75

Продовження лістингу Б18

```

        <th>Адреса</th>
        <th>Продано</th>
        <th>Закуплено</th>
        <th>Оренда</th>
    </tr>
</thead>
<?php
    $select_shop = "SELECT * FROM shop";
    $run = mysqli_query($conn,$select_shop);
    while ($row = mysqli_fetch_array($run)){
        ?>
        <tr id="<?php echo $row['_id'].'shop_tr'; ?>"
class="out-tr">
            <td id="<?php echo $row['_id'].'shop_id'; ?>"><?php
echo $row['_id']; ?></td>
            <td id="<?php echo $row['_id'].'shop_lead';
?>"><?php echo $row['_lead']; ?></td>
            <td id="<?php echo $row['_id'].'shop_address';
?>"><?php echo $row['_address']; ?></td>
            <td id="<?php echo $row['_id'].'shop_sold';
?>"><?php echo $row['_sold']; ?></td>
            <td id="<?php echo $row['_id'].'shop_bought';
?>"><?php echo $row['_bought']; ?></td>
            <td id="<?php echo $row['_id'].'shop_rent';
?>"><?php echo $row['_rent']; ?></td>
            <td id="<?php echo $row['_id'].'shop_e'; ?>"
class="btn shop_editor btn-info">Змінити</td>
            <td id="<?php echo $row['_id'].'shop_d'; ?>"
class="btn shop_deleter btn-danger">Видалити</td>
        </tr>
    <?php } ?>
</table>

```

Лістинг Б19 обробники подій та скрипти для додавання, оновлення і видалення записів про магазин

```

$("#create_shop_form").submit(function() {
$( "#tabs_shop" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/shop/create_shop.php",
    data: $(this).serialize()
}).done(function() {

    let shop_lead = $("#shop_lead_create").val();
    let shop_address = $("#shop_address_create").val();
    let shop_sold = $("#shop_sold_create").val();
    let shop_bought = $("#shop_bought_create").val();

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		76

Продовження лістингу Б19

```

let shop_rent = $("#shop_rent_create").val();

    $('#shop_table').append('<tr><td class="btn btn-
success">НОВИЙ</td><td>'+shop_lead+'</td><td>'+shop_address+'</td>
<td>'+shop_sold+'</td><td>'+shop_bought+'</td><td>'+shop_rent+'</t
d></tr>');
    });
    return false;
    });

    $("#update_shop_form").submit(function() {
    $( "#tabs_shop" ).tabs({
        active: 3
    });
    $.ajax({
        type: "POST",
        url: "forms/shop/update_shop.php",
        data: $(this).serialize()
    }).done(function() {
        let shop_id = $("#shop_id_update").val();
        let shop_lead = $("#shop_lead_update").val();
        let shop_address = $("#shop_address_update").val();
        let shop_sold = $("#shop_sold_update").val();
        let shop_bought = $("#shop_bought_update").val();
        let shop_rent = $("#shop_rent_update").val();
        $("#"+shop_id+"shop_tr").attr("bgcolor", "FAF0BE");
        $("#"+shop_id+"shop_lead").html(shop_lead);
        $("#"+shop_id+"shop_address").html(shop_address);
        $("#"+shop_id+"shop_sold").html(shop_sold);
        $("#"+shop_id+"shop_bought").html(shop_bought);
        $("#"+shop_id+"shop_rent").html(shop_rent);

        //$('#lead_table').append('<tr><td>'+lead_id+'</td><td>'+lead
_name+'</td><td>'+lead_phone+'</td><td>'+lead_email+'</td><td
class="btn btn-info">ЗМІНЕНО</td></tr>');
    });
    return false;
    });

    $("#delete_shop_form").submit(function() {
    $( "#tabs_shop" ).tabs({
        active: 3
    });
    $.ajax({
        type: "POST",
        url: "forms/shop/delete_shop.php",
        data: $(this).serialize()
    }).done( function () {
        let shop_id = $("#shop_id_delete").val();
        $("#"+shop_id+"shop_tr").attr("bgcolor", "FF7F50");
        setTimeout(function() {

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		77

Продовження лістингу Б19

```

        $("#"+shop_id+"shop_tr").remove();
    }, 1000);
    });
return false;
});

$(".shop_editor").on('click', function () {
    $("#tabs_shop").tabs({
        active: 1
    });
    let shop_id = parseInt ( $(this).attr("id") );
    let shop_lead = $("#"+shop_id+"shop_lead").html();
    let shop_address = $("#"+shop_id+"shop_address").html();
    let shop_sold = $("#"+shop_id+"shop_sold").html();
    let shop_bought = $("#"+shop_id+"shop_bought").html();
    let shop_rent = $("#"+shop_id+"shop_rent").html();

    $("#shop_id_update").val(shop_id);
    $("#shop_lead_update").val(shop_lead);
    $("#shop_address_update").val(shop_address);
    $("#shop_sold_update").val(shop_sold);
    $("#shop_bought_update").val(shop_bought);
    $("#shop_rent_update").val(shop_rent);
});

$(".shop_deleter").on('click', function () {
    $("#tabs_shop").tabs({
        active: 2
    });
    let shop_id = parseInt ( $(this).attr("id") );
    let shop_lead = $("#"+shop_id+"shop_lead").html();
    let shop_address = $("#"+shop_id+"shop_address").html();
    let shop_sold = $("#"+shop_id+"shop_sold").html();
    let shop_bought = $("#"+shop_id+"shop_bought").html();
    let shop_rent = $("#"+shop_id+"shop_rent").html();

    $("#shop_id_delete").val(shop_id);
    $("#shop_lead_delete").val(shop_lead);
    $("#shop_address_delete").val(shop_address);
    $("#shop_sold_delete").val(shop_sold);
    $("#shop_bought_delete").val(shop_bought);
    $("#shop_rent_delete").val(shop_rent);
});

<?php
    require_once("../..../database/db_lead.php");
    $lead = trim($_POST["shop_lead_create"]);
    $address = trim($_POST["shop_address_create"]);
    $sold = trim($_POST["shop_sold_create"]);
    $bought = trim($_POST["shop_bought_create"]);
    $rent = trim($_POST["shop_rent_create"]);

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		78

Продовження лістингу Б19

```

        $insert = "CALL create_shop('$lead', '$address', '$sold',
'$bought', '$rent')";
        $run = mysqli_query($conn,$insert);
    ?>

<?php
    require_once("../../database/db_lead.php");
    $id = $_POST['shop_id_update'];
    $shop_lead = $_POST['shop_lead_update'];
    $shop_address = $_POST['shop_address_update'];
    $shop_sold = $_POST['shop_sold_update'];
    $shop_bought = $_POST['shop_bought_update'];
    $shop_rent = $_POST['shop_rent_update'];
    $update = "CALL update_shop('$id', '$shop_lead',
'$shop_address', '$shop_sold', '$shop_bought', '$shop_rent')";
    $run = mysqli_query($conn,$update);
    ?>

<?php
    require_once("../../database/db_lead.php");
    $id = $_POST['shop_id_delete'];
    $delete = "CALL delete_shop('$id')";
    $run = mysqli_query($conn,$delete); //if run true
    ?>

```

Лістинг Б20 розмітка форми додавання запису про продавця

```

<form method="POST" id="create_seller_form">
    <div class="form-group">
        <label>Магазин</label>
        <input type="text" class="form-control"
name="seller_shop_create" id="seller_shop_create"
placeholder="магазин" required>
        <label>Продавець</label>
        <input type="text" class="form-control"
name="seller_name_create" id="seller_name_create"
placeholder="прізвище, ім'я продавця" required>
        <label>Номер телефону</label>
        <input type="phone" class="form-control"
name="seller_phone_create" id="seller_phone_create"
placeholder="номер телефону" required>
        <label>Email адреса</label>
        <input type="email" class="form-control"
name="seller_email_create" id="seller_email_create"
placeholder="name@example.com" required>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-success mb-2"
name="btn-seller-create" id="btn-seller-create">Додати
запис</button>

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дат		

```

    </div>
</form>

```

Лістинг Б21 розмітка форми оновлення запису про продавця

```

<form method="POST" id="update_seller_form">
  <div class="form-group">
    <label>Ідентифікатор</label>
    <input type="text" class="form-control"
name="seller_id_update" id="seller_id_update"
placeholder="ідентифікатор продавця" required readonly>
    <label>Магазин</label>
    <input type="text" class="form-control"
name="seller_shop_update" id="seller_shop_update"
placeholder="магазин" required>
    <label>Керівник</label>
    <input type="text" class="form-control"
name="seller_name_update" id="seller_name_update"
placeholder="прізвище, ім'я керівника" required>
    <label>Номер телефону</label>
    <input type="phone" class="form-control"
name="seller_phone_update" id="seller_phone_update"
placeholder="номер телефону" required>
    <label>Email адреса</label>
    <input type="email" class="form-control"
name="seller_email_update" id="seller_email_update"
placeholder="name@example.com" required>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-info mb-2"
name="btn-seller-update" id="btn-seller-update">Оновити
запис</button>
  </div>
</form>

```

Лістинг Б22 розмітка форми видалення запису про продавця

```

<form method="POST" id="delete_seller_form">
  <div class="form-group">
    <label>Ідентифікатор</label>
    <input type="text" class="form-control"
name="seller_id_delete" id="seller_id_delete"
placeholder="ідентифікатор продавця" readonly>
    <label>Магазин</label>
    <input type="text" class="form-control"
name="seller_shop_delete" id="seller_shop_delete"
placeholder="магазин" required>
    <label>Продавець</label>
    <input type="text" class="form-control"
name="seller_name_delete" id="seller_name_delete"
placeholder="прізвище, ім'я керівника" readonly>
    <label>Номер телефону</label>

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		80

Продовження лістингу Б22

```
<input type="phone" class="form-control"
name="seller_phone_delete" id="seller_phone_delete"
placeholder="номер телефону" readonly>
    <label>Email адреса</label>
    <input type="email" class="form-control"
name="seller_email_delete" id="seller_email_delete"
placeholder="name@example.com" readonly>
</div>
<div class="form-group">
    <button type="submit" class="btn btn-danger mb-2"
name="btn-seller-delete" id="btn-seller-delete">Видалити
запис</button>
</div>
</form>
```

Лістинг Б23 скрипт для табличного подання записів про продавця

```
<table id="seller_table" class="table container justify-
content-center">
    <thead>
        <tr>
            <th>id</th>
            <th>Магазин</th>
            <th>Продавець</th>
            <th>Телефон</th>
            <th>Email</th>
        </tr>
    </thead>
    <?php
$select_seller = "SELECT * FROM seller";
$run = mysqli_query($conn,$select_seller);
while ($row = mysqli_fetch_array($run)){
    ?>
    <tr id="<?php echo $row['_id'].'seller_tr'; ?>" class="out-
tr">
        <td id="<?php echo $row['_id'].'seller_id'; ?>"><?php
echo $row['_id']; ?></td>
        <td id="<?php echo $row['_id'].'seller_shop'; ?>"><?php
echo $row['_shop']; ?></td>
        <td id="<?php echo $row['_id'].'seller_name'; ?>"><?php
echo $row['_name']; ?></td>
        <td id="<?php echo $row['_id'].'seller_phone';
?>"><?php echo $row['_phone']; ?></td>
        <td id="<?php echo $row['_id'].'seller_email';
?>"><?php echo $row['_email']; ?></td>
        <td id="<?php echo $row['_id'].'seller_e'; ?>"
class="btn seller_editor btn-info">Змінити</td>
        <td id="<?php echo $row['_id'].'seller_d'; ?>"
class="btn seller_deleter btn-danger">Видалити</td>
    </tr> <?php } ?></table>
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		81

Лістинг Б24 обробники подій та скрипти для додавання, оновлення і

видалення записів про продавця

```

$("#create_seller_form").submit(function() {
$( "#tabs_seller" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/seller/create_seller.php",
    data: $(this).serialize()
}).done(function() {
    let seller_shop = $("#seller_shop_create").val();
    let seller_name = $("#seller_name_create").val();
    let seller_phone = $("#seller_phone_create").val();
    let seller_email = $("#seller_email_create").val();
    $('#seller_table').append('<tr><td class="btn btn-success">НОВИЙ</td><td>'+seller_shop+'</td><td>'+seller_name+'</td><td>'+seller_phone+'</td><td>'+seller_email+'</td></tr>');
});
return false;
});

$("#update_seller_form").submit(function() {
$( "#tabs_seller" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/seller/update_seller.php",
    data: $(this).serialize()
}).done(function() {
    let seller_id = $("#seller_id_update").val();
    let seller_shop = $("#seller_shop_update").val();
    let seller_name = $("#seller_name_update").val();
    let seller_phone = $("#seller_phone_update").val();
    let seller_email = $("#seller_email_update").val();
    $("#"+seller_id+"seller_tr").attr("bgcolor", "FAF0BE");
    $("#"+seller_id+"seller_shop").html(seller_shop);
    $("#"+seller_id+"seller_name").html(seller_name);
    $("#"+seller_id+"seller_phone").html(seller_phone);
    $("#"+seller_id+"seller_email").html(seller_email);
});
return false;
});

$("#delete_seller_form").submit(function() {
$( "#tabs_seller" ).tabs({
    active: 3
});
$.ajax({

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		82

Продовження лістингу Б24

```

        type: "POST",
        url: "forms/seller/delete_seller.php",
        data: $(this).serialize()
    }).done( function () {
        let seller_id = $("#seller_id_delete").val();
        $("#"+seller_id+"seller_tr").attr("bgcolor",
"FF7F50");

        setTimeout(function() {
            $("#"+seller_id+"seller_tr").remove();
        }, 1000);
    });
return false;
});

$(".seller_editor").on('click', function () {
    $( "#tabs_seller" ).tabs({
        active: 1
    });
    let seller_id = parseInt ( $(this).attr("id") );
    let seller_shop = $("#"+seller_id+"seller_shop").html();
    let seller_name = $("#"+seller_id+"seller_name").html();
    let seller_phone =
$("#"+seller_id+"seller_phone").html();
    let seller_email =
$("#"+seller_id+"seller_email").html();
    $("#seller_id_update").val(seller_id);
    $("#seller_shop_update").val(seller_shop);
    $("#seller_name_update").val(seller_name);
    $("#seller_phone_update").val(seller_phone);
    $("#seller_email_update").val(seller_email);
});

$(".seller_deleter").on('click', function () {
    $( "#tabs_seller" ).tabs({
        active: 2
    });
    let seller_id = parseInt ( $(this).attr("id") );
    let seller_shop = $("#"+seller_id+"seller_shop").html();
    let seller_name = $("#"+seller_id+"seller_name").html();
    let seller_phone = $("#"+seller_id+"seller_phone").html();
    let seller_email = $("#"+seller_id+"seller_email").html();
    $("#seller_id_delete").val(seller_id);
    $("#seller_shop_delete").val(seller_shop);
    $("#seller_name_delete").val(seller_name);
    $("#seller_phone_delete").val(seller_phone);
    $("#seller_email_delete").val(seller_email);
});

<?php
require_once("../../database/db_lead.php");

```

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		83

Продовження лістингу Б24

```
$shop = trim($_POST["seller_shop_create"]);
$name = trim($_POST["seller_name_create"]);
$phone = trim($_POST["seller_phone_create"]);
$email = trim($_POST["seller_email_create"]);
$insert = "CALL create_seller('$shop', '$name', '$phone',
'$email')";
$run = mysqli_query($conn,$insert);
?>

<?php
require_once("../../database/db_lead.php");
$id = $_POST['seller_id_update'];
$shop = $_POST['seller_shop_update'];
$name = $_POST['seller_name_update'];
$phone = $_POST['seller_phone_update'];
$email = $_POST['seller_email_update'];
$update = "CALL update_seller('$id', '$shop', '$name',
'$phone', '$email')";
$run = mysqli_query($conn,$update);
?>

<?php
require_once("../../database/db_lead.php");
$id = $_POST['seller_id_delete'];
$delete = "CALL delete_seller('$id')";
$run = mysqli_query($conn,$delete); //if run true
?>
```

Лістинг Б25 розмітка форми додавання записів про клієнта

```
<form method="POST" id="create_client_form">
  <div class="form-group">
    <label>Клієнт</label>
    <input type="text" class="form-control"
name="client_name_create" id="client_name_create"
placeholder="прізвище, ім'я клієнта" required>
    <label>Номер телефону</label>
    <input type="phone" class="form-control"
name="client_phone_create" id="client_phone_create"
placeholder="номер телефону" required>
    <label>Email адреса</label>
    <input type="email" class="form-control"
name="client_email_create" id="client_email_create"
placeholder="name@example.com" required>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-success mb-2"
name="btn-client-create" id="btn-client-create">Додати
запис</button>
  </div>
</form>
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						84
Змн.	Арк.	№ докум.	Підпис	Дат		

Лістинг Б26 розмітка форми оновлення записів про клієнта

```
<form method="POST" id="update_client_form">
  <div class="form-group">
    <label>Ідентифікатор</label>
    <input type="text" class="form-control"
name="client_id_update" id="client_id_update"
placeholder="ідентифікатор клієнта" required readonly>
    <label>Клієнт</label>
    <input type="text" class="form-control"
name="client_name_update" id="client_name_update"
placeholder="прізвище, ім'я клієнта" required>
    <label>Номер телефону</label>
    <input type="phone" class="form-control"
name="client_phone_update" id="client_phone_update"
placeholder="номер телефону" required>
    <label>Email адреса</label>
    <input type="email" class="form-control"
name="client_email_update" id="client_email_update"
placeholder="name@example.com" required>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-info mb-2"
name="btn-client-update" id="btn-client-update">Оновити
запис</button>
  </div>
</form>
```

Лістинг Б27 розмітка форми видалення запису про клієнта

```
<form method="POST" id="delete_client_form">
  <div class="form-group">
    <label>Ідентифікатор</label>
    <input type="text" class="form-control"
name="client_id_delete" id="client_id_delete"
placeholder="ідентифікатор клієнта" readonly>
    <label>Клієнт</label>
    <input type="text" class="form-control"
name="client_name_delete" id="client_name_delete"
placeholder="прізвище, ім'я клієнта" readonly>
    <label>Номер телефону</label>
    <input type="phone" class="form-control"
name="client_phone_delete" id="client_phone_delete"
placeholder="номер телефону" readonly>
    <label>Email адреса</label>
    <input type="email" class="form-control"
name="client_email_delete" id="client_email_delete"
placeholder="name@example.com" readonly>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-danger mb-2"
name="btn-client-delete" id="btn-client-delete">Видалити
запис</button></div></form>
```

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		85

Лістинг Б28 скрипт для табличного подання записів про клієнтів

```
<table id="client_table" class="table container justify-content-center">
  <thead>
    <tr>
      <th>id</th>
      <th>Клієнт</th>
      <th>Телефон</th>
      <th>Email</th>
    </tr>
  </thead>
  <?php
    $select_client = "SELECT * FROM client";
    $run = mysqli_query($conn,$select_client);
    while ($row = mysqli_fetch_array($run)){
      ?>
      <tr id="<?php echo $row['_id'].'client_tr'; ?>"
class="out-tr">
        <td id="<?php echo $row['_id'].'client_id';
?>"><?php echo $row['_id']; ?></td>
        <td id="<?php echo $row['_id'].'client_name';
?>"><?php echo $row['_name']; ?></td>
        <td id="<?php echo $row['_id'].'client_phone';
?>"><?php echo $row['_phone']; ?></td>
        <td id="<?php echo $row['_id'].'client_email';
?>"><?php echo $row['_email']; ?></td>
        <td id="<?php echo $row['_id'].'client_e'; ?>"
class="btn client_editor btn-info">Змінити</td>
        <td id="<?php echo $row['_id'].'client_d'; ?>"
class="btn client_deleter btn-danger">Видалити</td>
      </tr>
    <?php } ?>
</table>
```

Лістинг Б29 обробники подій та скрипти для додавання, оновлення і видалення записів про клієнта

```
$("#create_client_form").submit(function() {
$( "#tabs_client" ).tabs({
  active: 3
});
$.ajax({
  type: "POST",
  url: "forms/client/create_client.php",
  data: $(this).serialize()
}).done(function() {

  let client_name = $("#client_name_create").val();
  let client_phone = $("#client_phone_create").val();
  let client_email = $("#client_email_create").val();
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		86

Продовження лістингу Б29

```
$('#client_table').append('<tr><td class="btn btn-  
success">НОВИЙ</td><td>'+client_name+'</td><td>'+client_phone+'</t  
d><td>'+client_email+'</td></tr>');  
});  
return false;  
});  
  
$("#update_client_form").submit(function() {  
$( "#tabs_client" ).tabs({  
    active: 3  
    });  
$.ajax({  
    type: "POST",  
    url: "forms/client/update_client.php",  
    data: $(this).serialize()  
}).done(function() {  
    let client_id = $("#client_id_update").val();  
    let client_name = $("#client_name_update").val();  
    let client_phone = $("#client_phone_update").val();  
    let client_email = $("#client_email_update").val();  
    $("#"+client_id+"client_tr").attr("bgcolor", "FAF0BE");  
    $("#"+client_id+"client_name").html(client_name);  
    $("#"+client_id+"client_phone").html(client_phone);  
    $("#"+client_id+"client_email").html(client_email);  
});  
return false;  
});  
  
$("#delete_client_form").submit(function() {  
$( "#tabs_client" ).tabs({  
    active: 3  
    });  
$.ajax({  
    type: "POST",  
    url: "forms/client/delete_client.php",  
    data: $(this).serialize()  
}).done( function () {  
    let client_id = $("#client_id_delete").val();  
    $("#"+client_id+"client_tr").attr("bgcolor",  
"FF7F50");  
    setTimeout(function() {  
        $("#"+client_id+"client_tr").remove();  
    }, 1000);  
});  
return false;  
});  
  
$(".client_editor").on('click', function () {  
$( "#tabs_client" ).tabs({  
    active: 1  
    });  
});
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б29

```

        let client_id = parseInt ( $(this).attr("id") );
        let client_name = $("#"+client_id+"client_name").html();
        let client_phone =
$("#"+client_id+"client_phone").html();
        let client_email =
$("#"+client_id+"client_email").html();
        $("#client_id_update").val(client_id);
        $("#client_name_update").val(client_name);
        $("#client_phone_update").val(client_phone);
        $("#client_email_update").val(client_email);
    });

$(".client_deleter").on('click', function () {
    $("#tabs_client").tabs({
        active: 2
    });
    let client_id = parseInt ( $(this).attr("id") );
    let client_name = $("#"+client_id+"client_name").html();
    let client_phone = $("#"+client_id+"client_phone").html();
    let client_email = $("#"+client_id+"client_email").html();
    $("#client_id_delete").val(client_id);
    $("#client_name_delete").val(client_name);
    $("#client_phone_delete").val(client_phone);
    $("#client_email_delete").val(client_email);
});

<?php
    require_once("../../database/db_lead.php");
    $name = trim($_POST["client_name_create"]);
    $phone = trim($_POST["client_phone_create"]);
    $email = trim($_POST["client_email_create"]);
    $insert = "CALL create_client('$name', '$phone',
'$email')";
    $run = mysqli_query($conn,$insert);
?>

<?php
    require_once("../../database/db_lead.php");
    $id = $_POST['client_id_update'];
    $name = $_POST['client_name_update'];
    $phone = $_POST['client_phone_update'];
    $email = $_POST['client_email_update'];
    $update = "CALL update_client('$id', '$name', '$phone',
'$email')";
    $run = mysqli_query($conn,$update);
?><?php
    require_once("../../database/db_lead.php");
    $id = $_POST['client_id_delete'];
    $delete = "CALL delete_client('$id')";    $run =
mysqli_query($conn,$delete); //if run true?>

```

									Арк.
									88
Змн.	Арк.	№ докум.	Підпис	Дат	ДП. КН 21.449.12.000 ПЗ				

Лістинг Б30 розмітка форми додавання запису про товар

```
<form method="POST" id="create_goods_form">
  <div class="form-group">
    <label>Артикул</label>
    <input type="text" class="form-control"
name="goods_id_create" id="goods_id_create" placeholder="артикул"
required>
    <label>Магазин</label>
    <input type="text" class="form-control"
name="goods_shop_create" id="goods_shop_create"
placeholder="магазин" required>
    <label>Найменування товару</label>
    <input type="text" class="form-control"
name="goods_name_create" id="goods_name_create"
placeholder="найменування товару" required>
    <label>Ціна</label>
    <input type="text" class="form-control"
name="goods_price_create" id="goods_price_create"
placeholder="ціна" required>
    <label>Кількість</label>
    <input type="text" class="form-control"
name="goods_counter_create" id="goods_counter_create"
placeholder="кількість" required>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-success mb-2"
name="btn-goods-create" id="btn-goods-create">Додати
запис</button>
  </div>
</form>
```

Лістинг Б31 розмітка форми оновлення запису про товар

```
<form method="POST" id="update_goods_form">
  <div class="form-group">
    <label>Артикул</label>
    <input type="text" class="form-control"
name="goods_id_update" id="goods_id_update" placeholder="артикул"
required readonly>
    <label>Магазин</label>
    <input type="text" class="form-control"
name="goods_shop_update" id="goods_shop_update"
placeholder="магазин" required readonly>
    <label>Найменування товару</label>
    <input type="text" class="form-control"
name="goods_name_update" id="goods_name_update"
placeholder="найменування товару" required>
    <label>Ціна</label>
    <input type="price" class="form-control"
name="goods_price_update" id="goods_price_update"
placeholder="ціна" required>
    <label>Кількість</label>
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		89

Продовження лістингу БЗ1

```
        <input type="number" class="form-control"
name="goods_counter_update" id="goods_counter_update"
placeholder="кількість" required>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-info mb-2"
name="btn-goods-update" id="btn-goods-update">Оновити
запис</button>
    </div>
</form>
```

Лістинг БЗ2 розмітка форми видалення запису про товар

```
<form method="POST" id="delete_goods_form">
    <div class="form-group">
        <label>Артикул</label>
        <input type="text" class="form-control"
name="goods_id_delete" id="goods_id_delete" placeholder="артикул"
required readonly>
        <label>Магазин</label>
        <input type="text" class="form-control"
name="goods_shop_delete" id="goods_shop_delete"
placeholder="магазин" required readonly>
        <label>Найменування товару</label>
        <input type="text" class="form-control"
name="goods_name_delete" id="goods_name_delete"
placeholder="найменування товару" required readonly>
        <label>Ціна</label>
        <input type="price" class="form-control"
name="goods_price_delete" id="goods_price_delete"
placeholder="ціна" required readonly>
        <label>Кількість</label>
        <input type="number" class="form-control"
name="goods_counter_delete" id="goods_counter_delete"
placeholder="кількість" required readonly>
    </div>
    <div class="form-group">
        <button type="submit" class="btn btn-danger mb-2"
name="btn-goods-delete" id="btn-goods-delete">Видалити
запис</button>
    </div>
</form>
```

Лістинг БЗ3 скрипт для табличного подання записів про товари

```
<table id="goods_table" class="table container justify-
content-center">
    <thead>
        <tr>
            <th>Артикул</th>
```

					ДП. КН 21.449.12.000 ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б33

```

        <th>Магазин</th>
        <th>Найменування товару</th>
        <th>Ціна</th>
        <th>Кількість</th>
    </tr>
</thead>
<?php
    $select_goods = "SELECT * FROM goods";
    $run = mysqli_query($conn,$select_goods);
    while ($row = mysqli_fetch_array($run)){
        ?>
        <tr id="<?php echo $row['_article'].'goods_tr'; ?>"
class="out-tr">
            <td id="<?php echo $row['_article'].'goods_id';
?>"><?php echo $row['_article']; ?></td>
            <td id="<?php echo $row['_article'].'goods_shop';
?>"><?php echo $row['_shop']; ?></td>
            <td id="<?php echo $row['_article'].'goods_name';
?>"><?php echo $row['_name']; ?></td>
            <td id="<?php echo $row['_article'].'goods_price';
?>"><?php echo $row['_price']; ?></td>
            <td id="<?php echo
$row['_article'].'goods_counter'; ?>"><?php echo $row['_count'];
?></td>
            <td id="<?php echo $row['_article'].'goods_e'; ?>"
class="btn goods_editor btn-info">Змінити</td>
            <td id="<?php echo $row['_article'].'goods_d'; ?>"
class="btn goods_deleter btn-danger">Видалити</td>
        </tr>
        <?php } ?>
    </table>

```

Лістинг Б34 обробники подій та скрипти для додавання, оновлення і видалення записів про товар

```

$("#create_goods_form").submit(function() {
    $( "#tabs_goods" ).tabs({
        active: 3
    });
    $.ajax({
        type: "POST",
        url: "forms/goods/create_goods.php",
        data: $(this).serialize()
    }).done(function() {
        let goods_id = $("#goods_id_create").val();
        let goods_shop = $("#goods_shop_create").val();
        let goods_name = $("#goods_name_create").val();
        let goods_price = $("#goods_price_create").val();
        let goods_counter = $("#goods_counter_create").val();
    });

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		91

Продовження лістингу Б34

```
$('#goods_table').append('<tr><td>'+goods_id+'</td><td>'+goods_shop+'</td><td>'+goods_name+'</td><td>'+goods_price+'</td><td>'+goods_counter+'</td></tr>');
});
return false;
});

$("#update_goods_form").submit(function() {
$( "#tabs_goods" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/goods/update_goods.php",
    data: $(this).serialize()
}).done(function() {
    let goods_id = $("#goods_id_update").val();
    let goods_shop = $("#goods_shop_update").val();
    let goods_name = $("#goods_name_update").val();
    let goods_price = $("#goods_price_update").val();
    let goods_counter = $("#goods_counter_update").val();
    $("#"+goods_id+"goods_tr").attr("bgcolor", "FAF0BE");
    $("#"+goods_id+"goods_shop").html(goods_shop);
    $("#"+goods_id+"goods_name").html(goods_name);
    $("#"+goods_id+"goods_price").html(goods_price);
    $("#"+goods_id+"goods_counter").html(goods_counter);
});
return false;
});

$("#delete_goods_form").submit(function() {
$( "#tabs_goods" ).tabs({
    active: 3
});
$.ajax({
    type: "POST",
    url: "forms/goods/delete_goods.php",
    data: $(this).serialize()
}).done( function () {
    alert($("#goods_id_delete").val());
    let goods_id = $("#goods_id_delete").val();
    $("#"+goods_id+"goods_tr").attr("bgcolor",
"FF7F50");
    setTimeout(function(){
        $("#"+goods_id+"goods_tr").remove();
    }, 1000);
});
return false;
});
$(".goods_editor").on('click', function () {
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						92
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б34

```

$( "#tabs_goods" ).tabs({
    active: 1
});
let goods_id = ($(this).attr("id")).slice(0, 6);
let goods_shop = $("#"+goods_id+"goods_shop").html();
let goods_name = $("#"+goods_id+"goods_name").html();
let goods_price = $("#"+goods_id+"goods_price").html();
let goods_counter =
$("#"+goods_id+"goods_counter").html();
$("#goods_id_update").val(goods_id);
$("#goods_shop_update").val(goods_shop);
$("#goods_name_update").val(goods_name);
$("#goods_price_update").val(goods_price);
$("#goods_counter_update").val(goods_counter);
});

$(".goods_deleter").on('click', function () {
$( "#tabs_goods" ).tabs({
    active: 2
});
let goods_id = ($(this).attr("id")).slice(0, 6);
let goods_shop = $("#"+goods_id+"goods_shop").html();
let goods_name = $("#"+goods_id+"goods_name").html();
let goods_price = $("#"+goods_id+"goods_price").html();
let goods_counter = $("#"+goods_id+"goods_counter").html();
$("#goods_id_delete").val(goods_id);
$("#goods_shop_delete").val(goods_shop);
$("#goods_name_delete").val(goods_name);
$("#goods_price_delete").val(goods_price);
$("#goods_counter_delete").val(goods_counter);
});

<?php
    require_once("../../database/db_lead.php");
    $id = trim($_POST["goods_id_create"]);
    $shop = trim($_POST["goods_shop_create"]);
    $name = trim($_POST["goods_name_create"]);
    $price = trim($_POST["goods_price_create"]);
    $counter = trim($_POST["goods_counter_create"]);
    $insert = "CALL create_goods('$id', '$shop', '$name',
'$price', '$counter')";
    $run = mysqli_query($conn,$insert);
?>

<?php
    require_once("../../database/db_lead.php");
    $id = trim($_POST["goods_id_update"]);
    $name = trim($_POST["goods_name_update"]);
    $price = trim($_POST["goods_price_update"]);
    $counter = trim($_POST["goods_counter_update"]);

```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		93

Продовження лістингу Б34

```
$update = "CALL update_goods('$name', '$price',
'$counter', '$id')";
$run = mysqli_query($conn,$update);
?>

<?php
require_once("../../database/db_lead.php");
$id = $_POST['goods_id_delete'];
$delete = "CALL delete_goods('$id')";
$run = mysqli_query($conn,$delete); //if run true
?>
```

Лістинг Б35 обробники подій та скрипти для додавання записів про сутність «Рядок замовлення»

```
function makeObjArr( globalObjArr ){
let currentObject = [];
let order_str_article = $("#order_str_article_create").val();
let order_str_client = $("#order_str_client_create").val();

let order_str_price = $("#order_str_price_create").val();
let order_sum = $("#order_sum").text();
$("#order_sum").text(Number(order_str_price)+Number(order_sum
));
console.log(order_sum);

let order_str_counter = $("#order_str_counter_create").val();
let order_str_date = $("#order_str_date_create").val();
$('#current_order').append('<tr><td>'+order_str_article+'</td>
<td>'+order_str_client+'</td><td>'+order_str_price+'</td><td>'+or
der_str_counter+'</td><td>'+order_str_date+'</td></tr>');
currentObject = [order_str_article, order_str_client,
order_str_price, order_str_counter, order_str_date];
globalObjArr.push(currentObject);
currentObject = [];}

function buyHandler(globalObjArr){

console.log(globalObjArr);
$.ajax({
type: "POST",
url: "forms/order_str/create_order_str.php",
data: {objArr: globalObjArr},
success:function(data) {console.log(data);}}
).done(function() {

globalObjArr.forEach(element => {
let order_str_article = element[0];
let order_str_client = element[1];
let order_str_price = element[2];
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу Б35

```

        let order_str_counter = element[3];
        let order_str_date = element[4];
        $('#order_str_table').append('<tr><td class="btn btn-
success">НОВИЙ</td><td>'+order_str_article+'</td><td>'+order_str_c
lient+'</td><td>'+order_str_price+'</td><td>'+order_str_counter+'<
/td><td>'+order_str_date+'</td></tr>');
    });
});
return false;
}
<?php
    require_once("../../database/db_lead.php");
    $obj_array = $_POST["objArr"];

    foreach ($obj_array as $item) {
        $article = $item[0];
        $client = $item[1];
        $price = $item[2];
        $counter = $item[3];
        $date = $item[4];
        $insert = "CALL create_order_str('$article',
'$client', '$price', '$counter', '$date')";
        $run = mysqli_query($conn,$insert);
    }
?>

```

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		95

Додаток В

Тестування збержувальних процедур

Лістинг В1 запити для тестування ЗП

```
SELECT * FROM lead;
CALL create_lead( "lead1", "1111", "email1@email" );
SELECT * FROM lead;
CALL update_lead ( 1, "lead_update", "222", "email12@email"
);
SELECT * FROM lead;

SELECT * FROM shop;
CALL create_shop(1, "address1", 100.1, 140.7, 50 );
SELECT * FROM shop;
CALL update_shop( 1, 1, "address_edited", 101.6, 130.9, 70
);
SELECT * FROM shop;

SELECT * FROM seller;
CALL create_seller ( 1, "seller1", "1111", "email1@gmail.com"
);
SELECT * FROM seller;
CALL update_seller ( 1, 1, "seller_edited", "2222",
"email_edited@email" );
SELECT * FROM seller;

SELECT * FROM client;
CALL create_client ( "client1", "1111", "email@email" );
SELECT * FROM client;
CALL update_client ( 1, "client_edited", "2222",
"email_edited@email" );
SELECT * FROM client;

SELECT * FROM goods;
CALL create_goods ( "000001", 1, "goods1", 10.5, 5 );
SELECT * FROM goods;
CALL update_goods ( "goods2", 11, 2, "000001" );
SELECT * FROM goods;

SELECT * FROM users;
CALL create_user ( "user1", "1111", "email1@email", "qwerty"
);
SELECT * FROM users;
CALL change_user_pass ("email1@email", "no-qwerty" );
SELECT * FROM users;
CALL delete_user ( 1 );
SELECT * FROM users;
```

					ДП. КН 21.449.12.000 ПЗ	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дат		

Продовження лістингу В1

```
SELECT * FROM order_str;
CALL create_order_str( "000001", 1, 150, 2, '2021-06-04' );
SELECT * FROM order_str;
CALL update_order_str( 1, "000001", 1, 160, 2, '2021-06-04'
);

SELECT * FROM order_str;
CALL delete_order_str( 1 );
SELECT * FROM order_str;

CALL delete_goods ( 1 );
SELECT * FROM goods;
CALL delete_client ( 1 );
SELECT * FROM client;
CALL delete_seller ( 1 );
SELECT * FROM seller;
CALL delete_shop ( 1 );
SELECT * FROM shop;
CALL delete_lead ( 1 );
SELECT * FROM lead;
```

					ЛП. КН 21.449.12.000 ПЗ	Арк.
						97
Змн.	Арк.	№ докум.	Підпис	Дат		

Додаток Г

Таблиці до економічного розділу

Таблиця Г1 – Кошторис витрат на проектування

Найменування статей витрат	Сума, грн	Обґрунтування
1 Зарплата проектувальників.	246500	Сумарна вартість оплати праці усіх фахівців до створення проекту.
2. Відрахування на соціальні потреби.	54230	Сума соціальних відрахувань із коштів виділених на оплату праці (становить 22% від суми заробітних плат).
3. Контрагентські роботи і послуги.	0	Проект не потребує контрагентських робіт та послуг.
4. Витрати на відрядження.	0	Витрати на відрядження не пербачені через відсутність їх потреби.
5. Інші прямі витрати.	419	Вартість домену та хостингу.
6. Усього прямих витрат.	301149	Прямі витрати проекту.
7. Накладні витрати.	90344,7	Витрати на забезпечення проведення робіт, амортизація.
8. Планові накопичення.	78298,74	Витрати на розвиток проекту.
9. Усього, кошторисна вартість проекту.	469792,44	Сума прямих, накладних витрати і планових накопичень.
10. Податок на додану вартість.	93958,488	Розраховується як 20% від кошторисної вартості.
11. Загалом, договірна ціна розробки Зп.	563750,928	Витрати замовника на проект.

Таблиця Г2 – Розрахунок заробітної плати проектувальників

Посада	Оклад,	Відрахування	Кількість		Сума
виконавця	грн/міс	грн/міс	чол.	місяців	з/п, грн.
Керівник ДП	7000	1365	1	4	28000
Back-end розробник	30000	5850	1	1	30000
Front-end розробник	25000	4875	1	1	25000
Тестувальник	12500	2437,5	1	1	12500
DevOps	11000	2145	1	1	11000
Team Lead	35000	6825	1	4	140000
Усього зарплати:					246500

Таблиця Г3 – Розрахунок податків на доходи фізичних осіб

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник ДП	$7000 \times 18\% = 1260$ грн
Бекенд-розробник	$30000 \times 18\% = 5400$ грн
Фронтенд-розробник	$25000 \times 18\% = 4500$ грн
Тестувальник	$12500 \times 18\% = 2250$ грн
DevOps	$12500 \times 18\% = 2250$ грн
тімлід	$35000 \times 18\% = 6300$ грн

Таблиця Г4 – Розрахунок військового збору

Керівник ДП	$7000 \times 1,5\% = 105$ грн
Бекенд-розробник	$30000 \times 1,5\% = 450$ грн
Фронтенд-розробник	$25000 \times 1,5\% = 375$ грн
Тестувальник	$12500 \times 1,5\% = 187.5$ грн
DevOps	$12500 \times 1,5\% = 187.5$ грн
тімлід	$35000 \times 1,5\% = 525$ грн

Таблиця Г5 – Розрахунок єдиного соціального внеску

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник ДП	$7000 \times 22\% = 1540$ грн
Бекенд-розробник	$30000 \times 22\% = 6600$ грн
Фронтенд-розробник	$25000 \times 22\% = 5500$ грн
Тестувальник	$12500 \times 22\% = 2750$ грн
DevOps	$12500 \times 22\% = 2750$ грн
тімлід	$35000 \times 22\% = 7700$ грн

					ДП. КН 21.449.12.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дат		99

Таблиця Г6 – Розрахунок вартості утримання

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник ДП	1365 грн (1260 грн+105 грн)
Бекенд-розробник	5850 грн (5400 грн+450 грн)
Фронтенд-розробник	4875 грн (4500 грн+375 грн)
Тестувальник	2937.5 грн (2750 грн+187.5 грн)
DevOps	2937.5 грн (2750 грн+187.5 грн)
тімлід	6825 грн (6300 грн+525 грн)

Таблиця Г7 – Розрахунок суми виплати працівнику

Посада	Обчислення згідно чинного законодавства в сфері податкування та середньої ринкової заробітної плати
Керівник ДП	5635 грн.(7000 грн-1260 грн-105 грн)
Бекенд-розробник	34150 грн.(30000 грн-5400 грн-450 грн)
Фронтенд-розробник	20125 грн.(25000 грн-4500 грн-375 грн)
Тестувальник	9562.5 грн.(12500 грн-2937.5 грн)
DevOps	9562.5 грн.(12500 грн-2937.5 грн)
тімлід	28175 грн.(35000 грн-6828 грн)