

Київський національний торговельно-економічний університет

Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

*Розробка програмного забезпечення інтернет-магазину продажу
автозапчастин*

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення

підпис студента

Горбуліна Ігоря
Олександровича

Науковий керівник
кандидат технічних наук,
професор

підпис керівника

Пашорін Валерій
Іванович

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Цензура Микола
Олександрович

КИЇВ – 2020

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

Спеціалізація / освітня програма 121 «Інженерія програмного забезпечення»

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки
Криворучко О. В.
"7" листопада 2019 р.

Завдання на випускн кваліфікаційну роботу студентові

Горбуліну Ігорю Олександровичу

1. Тема випускної кваліфікаційної роботи: Розробка програмного забезпечення інтернет-магазину продажу автозапчастин

Затверджена наказом ректора від «02» грудня 2019 р. № 4112

2. Строк здачі студентом закінченої роботи _____

3. Цільова установка та вихідні дані до роботи:

Мета роботи: аналіз наявних програм для ведення обліку та продажу автозапчастин, а також вивчення й аналіз питань розробки програм для створення продукту «Магазин автозапчастин».

Об'єкт дослідження: діяльність магазину з продажу автозапчастин.

Предмет дослідження: розробка інформаційної системи обліку та продажу автозапчастин.

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ТЕХНІЧНЕ ЗАВДАННЯ ТА МАРКЕТИНГОВЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

1.1. Загальні відомості

1.1.1. Назва системи

1.1.2. Терміни роботи

1.1.3. Порядок оформлення і пред'явлення результатів робіт

1.1.4. Потенційні користувачі системи

1.2. Мета, призначення та передумови створення системи

1.2.1. Призначення системи

1.3. Вимоги до системи

1.3.1. Вимоги до системи в цілому

1.3.2. Функціональні вимоги

1.3.4. Вимоги до захисту інформації від несанкціонованого доступу

1.4. Вимоги до функцій (задач), що виконуються системою

1.4.1. Вимоги до програмного забезпечення

1.4.2. Вимоги до технічного забезпечення

1.4.3. Вимоги до організаційного забезпечення

1.5. Огляд існуючих рішень для ведення обліку в магазинах з продажу автозапчастин

1.6. Аналіз мов програмування

1.6.1. Мова програмування C++

1.6.2. Мова програмування Java

1.6.3. Мова програмування C#

1.6.4. SQL

1.7. Системи керування базами даних

1.8. Висновки до розділу 1

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА БАЗИ ДАНИХ

2.1. Постановка задач

2.2. Проектування бази даних

2.2.1. Проектування логічної моделі

2.2.2. Проектування фізичної моделі

2.3. Створення бази даних в додатку MS SQL Server Management Studio

2.4. Висновки до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМИ

3.1. Створення проекту та підключення бази даних

3.2. Створення та програмування форми «Авторизація»

3.3. Створення головної форми додатку

3.4. Створення другорядних форм

3.5. Створення форми додавання нового запису головної форми

3.6. Створення форми додавання нового запису до другорядної форми

3.7. Створення форми детальної інформації про товар

3.8. Створення форми видалення даних

3.9. Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>		
2.	<i>Вступ та перелік літературних джерел</i>		
3.	<i>Розділ 1. Технічне завдання та маркетингове дослідження програми</i>		
4.	<i>Розділ 2. Проектування та розробка бази даних</i>		
5.	<i>Розділ 3. Реалізація програми</i>		
6.	<i>Висновки</i>		
7.	<i>Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)</i>		
8.	<i>Підготовка автореферату та презентації доповіді</i>		
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>		
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>		
11.	<i>Здача прожитої випускної кваліфікаційної роботи на кафедрі</i>		
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання « » 20 р.

8. Науковий керівник випускної кваліфікаційної роботи

Пашорін В. І.

9. Гарант освітньої програми

Цензура М. О.

10. Завдання прийняв до виконання студент

Горбулін І. О.

АНОТАЦІЯ

Дипломна робота на тему «Розробка програмного забезпечення інтернет-магазину продажу автозапчастин» містить 55 сторінок, 25 рисунків та 1 додаток.

Мета дослідження: аналіз наявних програм для ведення обліку та продажу автозапчастин, а також вивчення й аналіз питань розробки програм для створення продукту «Магазин автозапчастин».

Об'єкт дослідження: діяльність магазину з продажу автозапчастин.

Предметом дослідження є розробка інформаційної системи обліку та продажу автозапчастин.

У першому розділі складено технічне завдання, проведено маркетингове дослідження; створено вимоги до розробки програми.

У другому розділі було спроектовано та створено базу даних програми.

У третьому розділі описано процес реалізації програми.

ANNOTATION

The thesis on the topic "Software development of an online store selling auto parts" contains 55 pages, 25 drawings and 1 appendix.

The purpose of the study: analysis of existing programs for accounting and sales of auto parts, as well as the study and analysis of software development for the product "Auto Parts Store".

Research subject: activity of a shop selling auto parts.

The study's subject is the development of an information system for accounting and sales of auto parts.

In the first section the technical task is made, the marketing research is carried out; created requirements for program development.

In the second section, the program database was designed and created.

The third section describes the process of program implementation.

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ТЕХНІЧНЕ ЗАВДАННЯ ТА МАРКЕТИНГОВЕ ДОСЛІДЖЕННЯ ПРОГРАМИ.....	6
1.1. Загальні відомості	6
1.1.1. Назва системи	6
1.1.2. Терміни роботи	6
1.1.3. Порядок оформлення і пред’явлення результатів робіт.....	6
1.1.4. Потенційні користувачі системи	7
1.2. Мета, призначення та передумови створення системи.....	7
1.2.1. Призначення системи.....	7
1.3. Вимоги до системи	7
1.3.1. Вимоги до системи в цілому	7
1.3.2. Функціональні вимоги.....	8
1.3.4. Вимоги до захисту інформації від несанкціонованого доступу	8
1.4. Вимоги до функцій (задач), що виконуються системою.....	9
1.4.1. Вимоги до програмного забезпечення	9
1.4.2. Вимоги до технічного забезпечення	9
1.4.3. Вимоги до організаційного забезпечення.....	9
1.5. Огляд існуючих рішень для ведення обліку в магазинах з продажу автозапчастин	10
1.6. Аналіз мов програмування	11
1.6.1. Мова програмування C++	11

					<i>КНТЕУ 121 07-04.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>			<i>Розробка програмного забезпечення інтернет-магазину продажу автозапчастин</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Пашорін В.О.</i>				<i>Зміст</i>	<i>2</i>	<i>48</i>
<i>Гарант</i>		<i>Цензура М. О.</i>				<i>Факультет інформаційних технологій, 4 курс, 7 група</i>		
<i>Розроб.</i>		<i>Горбулін І.О.</i>						
					<i>Зміст</i>			

1.6.2. Мова програмування Java.....	12
1.6.3. Мова програмування C#.....	12
1.6.4. SQL.....	12
1.7. Системи керування базами даних	13
1.8. Висновки до розділу 1.....	16
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА БАЗИ ДАНИХ	18
2.1. Постановка задач	18
2.2. Проектування бази даних.....	19
2.2.1. Проектування логічної моделі	19
2.2.2. Проектування фізичної моделі.....	20
2.3. Створення бази даних в додатку MS SQLServer Management Studio.....	21
2.4. Висновки до розділу 2.....	27
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМИ	28
3.1. Створення проекту та підключення бази даних.....	28
3.2. Створення та програмування форми «Авторизація».....	29
3.3. Створення головної форми додатку.....	30
3.4. Створення другорядних форм	33
3.5. Створення форми додавання нового запису головної форми.....	34
3.6. Створення форми додавання нового запису до другорядної форми	37
3.7. Створення форми детальної інформації про товар.....	39
3.8. Створення форми видалення даних.....	42
3.9. Висновки до розділу 3.....	44
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	45
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	47
ДОДАТКИ.....	49

					<i>КНТЕУ 121 07-04.БР</i>	<i>Аркуш</i>
						3
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

ВСТУП

Характерними рисами сучасного суспільства є постійний розвиток та активне впровадження процесів інформатизації у більшість сфер діяльності людини. При споживанні великої кількості інформації виникає потреба у застосуванні нових інформаційних технологій. Інформаційні технології спрямовані на доставку інформації споживачам – менеджерам, конструкторам, керівникам, конкретним покупцям. Завдяки ним можна здійснити автоматизацію рутинних операцій керування, отримання інформації для прийняття рішень в управлінні підприємством.

Інформаційна система створюється для конкретного підприємства відповідно до поставленої мети. Ефективна інформаційна дає змогу кожному рівню управління отримувати тільки ту інформації, що йому необхідна для ефективної реалізації функцій управління. Впровадження інформаційних систем застосовується з метою підвищення ефективності виробничо-господарської діяльності фірми за рахунок не тільки опрацювання і збереження інформації, автоматизації робіт, але і за рахунок принципово нових методів управління та прийняття рішень.

Таким чином, можна дійти висновку, що впровадження ефективних інформаційних систем є важливим етапом розвитку успішного підприємства, необхідним для автоматизації діяльності та покращення умов управління підприємством.

					<i>КНТЕУ 121 07-04.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка програмного забезпечення інтернет-магазину продажу автозапчастин</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>				<i>В</i>	<i>4</i>	<i>48</i>
<i>Керівник</i>		<i>Пашорін В.І.</i>				<i>Вступ</i>		
<i>Гарант</i>		<i>Цензура М. О.</i>						
<i>Розроб.</i>		<i>Горбулін І.О.</i>						

Актуальність. Розробка інформаційної системи обліку та продажу автозапчастин є актуальною, оскільки, зі зростанням кількості автомобілів, зростає потреба в автозапчастинах. В наслідок цього, з'являються нові постачальники та магазини, яким потрібне якісне програмне забезпечення для ведення торгівлі.

Мета дослідження випускної роботи: аналіз наявних програм для ведення обліку та продажу автозапчастин, а також вивчення й аналіз питань розробки програм для створення продукту «Магазин автозапчастин».

Об'єкт дослідження: діяльність магазину з продажу автозапчастин.

Предметом дослідження є розробка інформаційної системи обліку та продажу автозапчастин.

Завдання дослідження:

- створити платформу для обліку товарів та продажів автозапчастин;
- використати нові платформи для розробки програмного забезпечення;
- провести аналіз аудиторії користувачів програми;
- провести підготовку програми до її реалізації;
- створити базу даних магазину.

Методи та технології розробки. Програмування виконувалося за допомогою таких платформ, як MS SQLServer, Microsoft Visual Studio 19, мова програмування – C#.

Результат роботи – розроблена легка в користуванні, конкурентоспроможна програма, для зберігання інформації про товари магазину та ведення обліку їх продажів та поставок.

					КНТЕУ 121 07-04.БР	Аркуш
						5
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 1.

ТЕХНІЧНЕ ЗАВДАННЯ ТА МАРКЕТИНГОВЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

1.1. Загальні відомості

1.1.1. Назва системи

Повне найменування системи: Інформаційна система обліку та продажу автозапчастин.

Коротке найменування системи: Магазин автозапчастин.

1.1.2. Терміни роботи

Початок роботи: жовтень 2019 р.

Закінчення роботи: травень 2020 р.

1.1.3. Порядок оформлення і пред'явлення результатів робіт

Програмне забезпечення створюється у вигляді готового програмного продукту з повним доступом до ресурсів програми, з можливістю їх подальшого використання та зміни. Роботи зі створення програми «Магазин автозапчастин» виконуються поетапно в наступній послідовності: етап, зміст роботи і результат:

- проведення маркетингового дослідження;
- створення технічного завдання;
- вибір платформи для роботи над проектом;
- проектування інтерфейсів та розробка графічного дизайну всіх модулів;
- створення прототипу (погодження функціоналу та дизайну програми);

Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 07-04.БР</i>			
Зав. кафедри		Криворучко О.В.			<i>Розробка програмного забезпечення інтернет-магазину продажу автозапчастин</i>	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.				Р1	6	48
Гарант		Цензура М. О.				Факультет інформаційних технологій, 4 курс, 7 група		
Розроб.		Горбулін І.О.						

- розробка програмної частини модулів системи, інтеграція шаблонів, налаштування інтерфейсів;
- проектування та створення бази даних системи, схеми даних, таблиць для збереження даних та зв'язків між ними;
- тестування системи на контрольних прикладах.

Після реалізації цих пунктів, результат проведених робіт пред'являється за допомогою презентації.

1.1.4. Потенційні користувачі системи

Цільовою аудиторією системи є фізичні та юридичні особи, які займаються роздрібною торгівлею автомобільних запчастин та яким потрібна зручна програма для ведення обліку товарів на складі, постачання, та продажів.

1.2. Мета, призначення та передумови створення системи

1.2.1. Призначення системи

Програмний продукт призначений автоматизувати та пришвидшити роботу диспетчера каси магазину автозапчастин. Програма дозволить без будь-яких труднощів оперувати товарами, постачаннями та замовленнями. Тобто додавати, редагувати та видаляти всі дані збережені в базі даних програми.

1.2.2. Мета створення системи

Проектування, розробка та введення в експлуатацію інформаційної системи ведення обліку продажів магазину автозапчастин.

1.3. Вимоги до системи

1.3.1. Вимоги до системи в цілому

					<i>КНТЕУ 121 07-04.БР</i>	<i>Аркуш</i>
						7
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

Програмний комплекс повинен забезпечувати зберігання та обробку даних з використанням бази даних, доступ до якої здійснюється в межах локальної обчислювальної мережі.

У склад програмного комплексу повинні входити програмні модулі:

- база даних;
- пошук і контроль товарів;
- ведення обліку замовлень.

1.3.2. Функціональні вимоги

Програмний продукт повинен володіти наступними функціональними характеристиками:

- зручний користувацький інтерфейс;
- наявність бази даних;
- можливість збереження та редагування даних компанії;
- пошук товарів за різними критеріями.

1.3.3. Вимоги до надійності

Надійне функціонування програмного комплексу має бути забезпечене шляхом:

- контролю коректності та повноти введених даних;
- відновлення після відмови;
- при виникненні системного збою програма повинна поновити роботу з останнього зафіксованого робочого стану.

1.3.4. Вимоги до захисту інформації від несанкціонованого доступу

Захист інформації від несанкціонованого доступу повинен забезпечуватися

					<i>КНТЕУ 121 07-04.БР</i>	Аркуш
						8
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

формою авторизації користувача.

1.4. Вимоги до функцій (задач), що виконуються системою

1.4.1. Вимоги до програмного забезпечення

Програмний комплекс повинен бути побудований на клієнт-серверній архітектурі з використанням технологій, що не вимагають додаткового ліцензування. Серверна частина програмного комплексу має базуватися на системі керування базами даних MS SQLServer. Базовою мовою програмування пропонується обрати C#. Програма повинна працювати в режимі офф-лайн. При розробці дозволяється застосовувати фреймворки. Клієнтська частина повинна бути побудована на стандартних технологіях, що не вимагають встановлення додаткових компонентів на комп'ютер.

1.4.2. Вимоги до технічного забезпечення

Програмний комплекс розрахований на функціонування при такому наборі технічних засобів:

- процесор з тактовою частотою від 1.4 ГГц;
- оперативна пам'ять об'ємом не менше 1 Гб;
- відеокарта та монітор з роздільною здатністю не менше 1024x768;
- клавіатура;
- мишка;
- вільний дисковий простір не менш ніж 500 Мб.

1.4.3. Вимоги до організаційного забезпечення

Програмне забезпечення повинно бути компактним і не займати багато місця(можливо, потрібна буде архівація програми). Вся навігація по змісту програми повинна бути описана в окремому документі Word.

					<i>КНТЕУ 121 07-04.БР</i>	<i>Аркуш</i>
						9
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

1.5. Огляд існуючих рішень для ведення обліку в магазинах з продажу автозапчастин

На програмному ринку у великій кількості представлені програми ведення обліку в магазинах. Більшість програм мають відносно невеликі функціональні можливості та забезпечують оперування лише з СУБД простої ієрархії, тому відносяться до локальних систем. Основною задачею таких систем є інформаційне забезпечення та автоматизація роботи відділу продажів. Також часто зустрічаються програми, розроблені на замовлення магазинів під їх особисті потреби.

«Продаж автозапчастин 3.0»

Компанія-розробник “Coding” має багаторічний досвід в розробці ПЗ для обліку та підбору автозапчастин. Додаток дозволяє вести облік надходження товарів, залишків товарних запасів, замовлення покупців та оформлення продажу.

Також, є можливість формування та автоматичної реєстрації нових посилок в Новій Пошті та Ін-Тайм.

«Альтаір-Сервіс»

Програма розроблена з урахуванням рекомендації компаній працюючих в сфері продажу автозапчастин. Додаток дозволяє вести облік товару який надійшов на склад, відстежувати його переміщення та наявність з урахуванням місця зберігання. Допомагає сформувати пакет документів для клієнтів з можливістю вказати автомобіль.

					<i>КНТЕУ 121 07-04.БР</i>	<i>Аркуш</i>
						10
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

«CarStore»

Продукт призначений в першу чергу для автосервісів, автомагазинів та складів з продажу запчастин для автомобілів та спецтехніки. Основна перевага програми CarStore - її пристосованість до специфіки роботи підприємств, що займаються торгівлею автозапчастинами. Головними можливостями додатку є: складський облік, товарний облік, фінансовий облік, планування замовлень автосервісу, інтеграція з різними базами даних.

«AutoІнтелект»

Програма розроблена спеціально для сучасних підприємств, які займаються продажами автозапчастин. Об'єднує в собі всі особливості торгівлі автозапчастинами: облік, підбір, багатофункціональний пошук, замовлення та продаж автозапчастин.

1.6. Аналіз мов програмування

1.6.1. Мова програмування C++

C++ – мова програмування високого рівня з підтримкою кількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм, наприклад, відеоігор. При створенні C++ прагнули зберегти сумісність з мовою C. Більшість програм на C справно працюватимуть і з компілятором C++. C++ має синтаксис, заснований на синтаксисі C.

						КНТЕУ 121 07-04.БР	Аркуш
							11
Зм.	Аркуш	№ докум	Підпис	Дата			

1.6.2. Мова програмування Java

Java – об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems у 1995 році як основний компонент платформи Java. Синтаксис мови багато в чому походить від C та C++. Java програми виконуються у середовищі віртуальної машини Java. Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною для конкретної платформи. В 2009 році Sun Microsystems придбала компанія Oracle, яка продовжує розвивати Java.

1.6.3. Мова програмування C#

C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Мова, розроблена для повної сумісності з ініціативою Microsoft .NET, використовуючи переваги, отримані від C, C++ та Java. C# є дуже близьким родичем мови програмування Java. Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів. Якщо програми на різних мовах виконуються на платформі .NET, .NET бере на себе клопіт щодо сумісності програм (тобто типів даних, за кінцевим рахунком).

1.6.4. SQL

SQL (англ. Structured Query Language - Мова структурованих запитів) - це комп'ютерна мова, призначена для зберігання, маніпулювання та запиту даних, що зберігаються в реляційних базах даних. SQL використовується для запиту, вставки, оновлення та модифікації даних. Більшість реляційних баз даних підтримують SQL, що є додатковою перевагою для адміністраторів баз даних

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

КНТЕУ 121 07-04.БР

(DBA), оскільки вони часто вимагають підтримки баз даних на декількох різних платформах.

Мова SQL поділяється на вирази, предикати, запити, інструкції. Вирази можуть генерувати скалярні значення, або таблиці з рядками та стовпчиками, запити отримують дані на основі заданих критеріїв, інструкції чинять дії на дані або контролюють потік виконання програми.

Команди SQL поділяються на декілька різних типів, серед них - мови маніпулювання даними (DML) і оператори мови визначення даних (DDL), управління транзакціями та заходи безпеки. Словник DML використовується для отримання та маніпулювання даними, тоді як DDL - виписки призначені для визначення та зміни структур бази даних. Контроль транзакцій допомагає керувати обробкою транзакцій, гарантуючи, що транзакції завершуються або відновлюються, якщо виникають помилки або проблеми. Записи безпеки використовуються для управління доступом до бази даних, а також для створення ролей і дозволів користувача.

1.7. Системи керування базами даних

Системи керування базами даних (СКБД) - системне програмне забезпечення для створення та управління базами даних. СКБД надає користувачам і програмістам систематичний спосіб створення, отримання, оновлення та керування даними.

СУБД дозволяє кінцевим користувачам створювати, читати, оновлювати і видаляти дані в базі даних. СУБД, по суті, служить інтерфейсом між базою даних і кінцевими користувачами або прикладними програмами, забезпечуючи послідовне організування даних і залишається легкодоступним.

					КНТЕУ 121 07-04.БР	Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата		

СУБД керує трьома важливими речами: даними, механізмом бази даних, що дозволяє отримувати доступ до даних, блокувати і змінювати, і схему баз даних, яка визначає логічну структуру бази даних. Ці три основні елементи допомагають забезпечити паралельність, безпеку, цілісність даних і єдині адміністративні процедури. Типові завдання адміністрування баз даних, що підтримуються СУБД, включають управління змінами, моніторинг, налаштування продуктивності, резервне копіювання та відновлення. Багато систем управління базами даних також несуть відповідальність за автоматичні відкати, перезавантаження та відновлення, а також за реєстрацію та аудит діяльності. СУБД є, мабуть, найбільш корисним для забезпечення централізованого перегляду даних, які можуть бути доступні декільком користувачам з декількох місць керування.

Найбільш популярними СКБД є Microsoft SQL Server, MySQL, PostgreSQL, Oracle, та Microsoft Access.

Microsoft SQL Server

Microsoft SQL Server - це система управління реляційними базами даних, або RDBMS, яка підтримує широкий спектр обробки транзакцій, бізнес-аналітики та аналітичних додатків у корпоративних IT-середовищах. Це одна з трьох провідних технологій баз даних, разом з Oracle Database та IBM DB2.

Переваги:

- Швидка і стабільна
- Надає доступ до візуалізації на мобільних пристроях

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

КНТЕУ 121 07-04.БР

- Добре взаємодіє з іншими продуктами Microsoft

Недоліки:

- Для малих підприємств ціна занадто висока і не відповідає бюджету
- Вимоглива до параметрів комп'ютера
- використовує основну мову, яка відрізняється від інших СКБД, таких як MySQL і Oracle.

MySQL

MySQL – найпопулярніша система управління базами даних SQL з відкритим кодом, розробляється, розповсюджується і підтримується корпорацією Oracle. Сервер MySQL надає систему управління базами даних з можливостями запитів і підключення, а також здатність мати відмінну структуру даних і інтеграцію з багатьма різними платформами. Вона може обробляти великі бази даних надійно і швидко у високопродуктивних виробничих середовищах.

Переваги:

- проста у використанні
- є найбільш безпечною і надійною СКБД
- Існує безліч користувацьких інтерфейсів, які можуть бути реалізовані
- Доступна безкоштовно
- Працює на багатьох ОС

Недоліки:

- Немає вбудованої підтримки XML або OLAP.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		15

КНТЕУ 121 07-04.БР

- Не рекомендується для високої швидкості передачі даних
- Використовує багато пам'яті

Microsoft Access

Microsoft Access – це система управління базами даних (СУБД) від Microsoft, яка поєднує в собі Microsoft Jet Database Engine з графічним інтерфейсом користувача та інструментами розробки програмного забезпечення. Він є членом пакету програм Microsoft Office, включених до професійних та вищих редакцій.

Переваги:

- Проста в установці та використанні
- добре працює з багатьма програмами, що базуються на Windows
- Багатокористувацька підтримка
- Імпортування даних

Недоліки:

- не настільки надійний, як MS SQL Server або Oracle
- Всі дані зберігаються в одному файлі
- Важко включити Мультимедійні дані в Microsoft Access

1.8. Висновки до розділу 1

У першому розділі було створено технічне завдання до системи та проведено маркетингове дослідження, визначено мету розробки системи та основні вимоги до неї.

Також, були проаналізовані існуючі програми для ведення обліку

					<i>КНТЕУ 121 07-04.БР</i>	<i>Аркуш</i>
						16
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

продажів магазину автозапчастин. Було виявлено, що розробка даного програмного продукту є досить актуальною, оскільки на ринку представлено досить багато програм з різними можливостями та недоліками. Головними недоліком притаманним майже кожній такій програмі є – ціна.

Крім того, були проаналізовані популярні мови програмування, їх переваги та недоліки. Їх вибір є цілком індивідуальним. Оскільки програмний продукт буде створюватися для операційної системи Windows, було вирішено використовувати мову програмування C# та середовище розробки Visual Studio 2017. У якості СУБД було обрано Microsoft SQL Server, оскільки вона легко інтегрується в середовище розробки .Net Framework, та повністю відповідає потребам даного програмного продукту

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	17

РОЗДІЛ 2.

ПРОЕКТУВАННЯ ТА РОЗРОБКА БАЗИ ДАНИХ

2.1. Постановка задач

Для реалізації програми необхідно спроектувати та розробити базу даних для магазину з продажу автозапчастин. Проектована база даних повинна зберігати та обробляти інформацію, введену на основі вхідних документів, виводити результати обробки користувачеві. На старті програми є список замовлень, які необхідно виконати, а також список доступних продуктів (деталей).

База даних повинна забезпечувати наступні можливості:

- Облік реалізації деталей;
- Облік товарів для продажу ;
- Перелік типів автозапчастин;
- Перелік автомобілів, деталі для яких продаються в магазині;
- Перелік країн виробників деталей, представлених для продажу;
- Таблиця клієнтів та їх особистих даних ;
- Можливість вносити зміни в дані та поповнення новими даними;
- Здійснювати функцію пошуку по замовленнях та товарах за заданими характеристиками;
- Містити необхідні запити до інформації ;

Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 07-04.БР</i>			
Зав. кафедри		Криворучко О.В.			<i>Розробка програмного забезпечення інтернет-магазину продажу автозапчастин</i>	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.				P2	18	48
Гарант		Цензура М. О.				Факультет інформаційних технологій, 4 курс, 7 група		
Розроб.		Горбулін І.О.				Проектування та розробка бази даних		

На даному етапі головним завданням є впровадження основних можливостей функціоналу. Більш складні можливості будуть реалізовані на наступних етапах розробки.

2.2. Проектування бази даних

2.2.1. Проектування логічної моделі

Наступним кроком для побудови бази даних є створення її логічної моделі. Перед проектуванням логічної моделі необхідно визначити які таблиці повинна містити база даних.

По-перше, база даних додатку має містити дві основні таблиці: таблицю “Товари”, яка зберігає список товарів, доступних для покупки, та інформацією про товар, а також, таблицю “Замовлення”, що зберігає інформацію про товари, які були продані.

По-друге, необхідні таблиці “Клієнти” та “Працівники”, які будуть містити особисті дані клієнтів та працівників магазину.

Також, знадобляться таблиці, що містять список типів деталей, автомобілів, для яких є деталі, країн виробників автозапчастин та самих автомобілів, а також, таблиці з даними про постачальників та постачання запчастин.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		19

КНТЕУ 121 07-04.БР

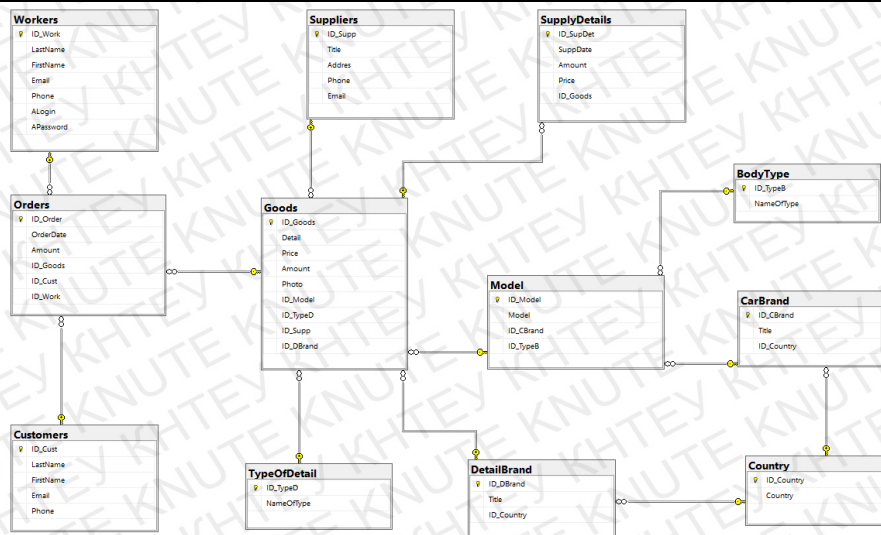


Рис. 2.1. Логічна модель бази даних

2.2.2. Проектування фізичної моделі

Поля, які містять первинні та зовнішні ключі мають тип даних INT та унікальні значення. Поле «Кількість» також має тип даних INT.

Поля «Дата замовлення» та «Дата поставки» має тип даних DATE, оскільки зберігають інформацію про дату замовлення та поставок.

Решта полів мають строковий тип даних VARCHAR. Оскільки поле «Телефон» може містити деякі символи, необхідні для позначення формату номера, то це поле також матиме строковий тип VARCHAR.

						Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	

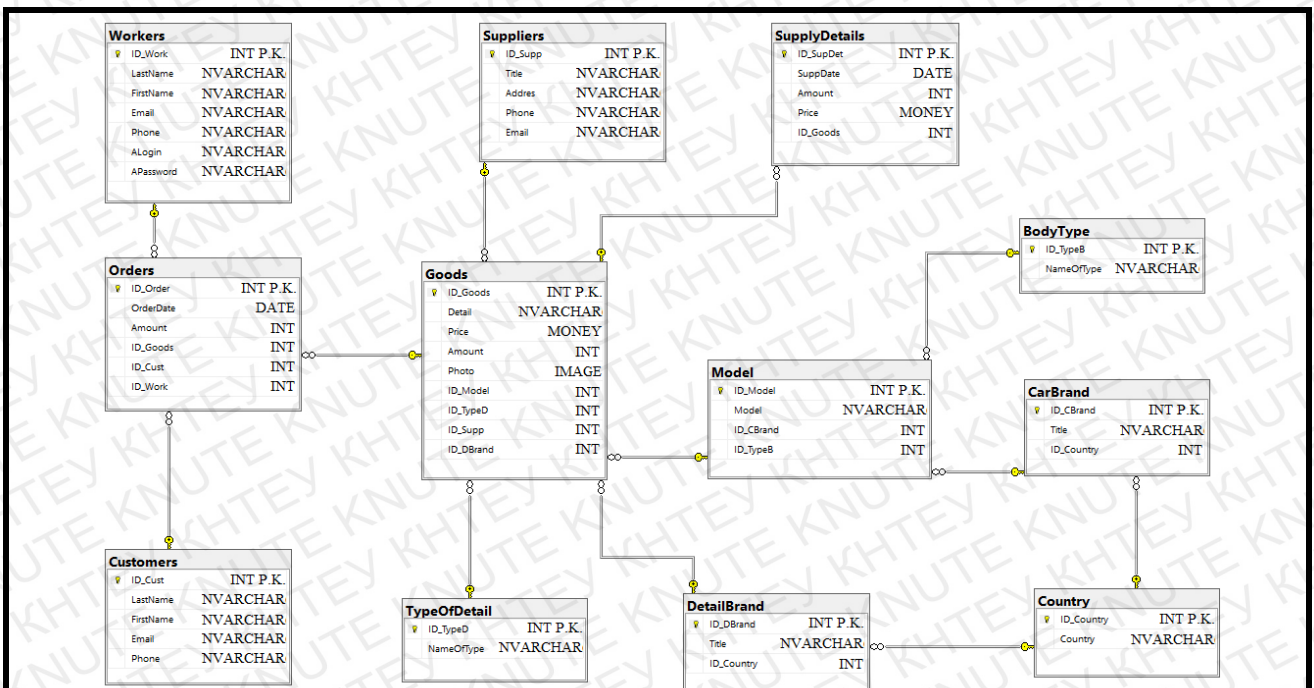


Рис. 2.2. Фізична модель бази даних

2.3. Створення бази даних в додатку MS SQLServer Management Studio

Перше, що потрібно зробити для створення бази даних, це запуснути середовище SQL Server Management Studio та підключитися до SQL серверу.

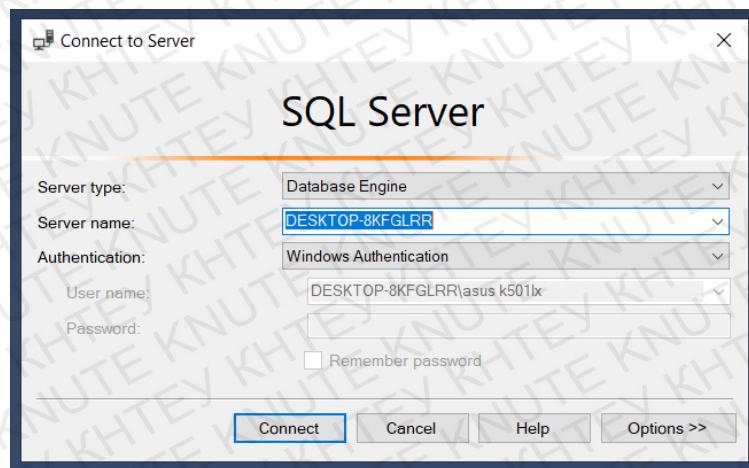


Рис. 2.3. Підключення до SQL серверу

					Аркуш
					КНТЕУ 121 07-04.БР
Зм.	Аркуш	№ докум	Підпис	Дата	21

Далі необхідно натиснути на призначення сервера у вікні Object Explorer і в меню виберемо пункт New Query.

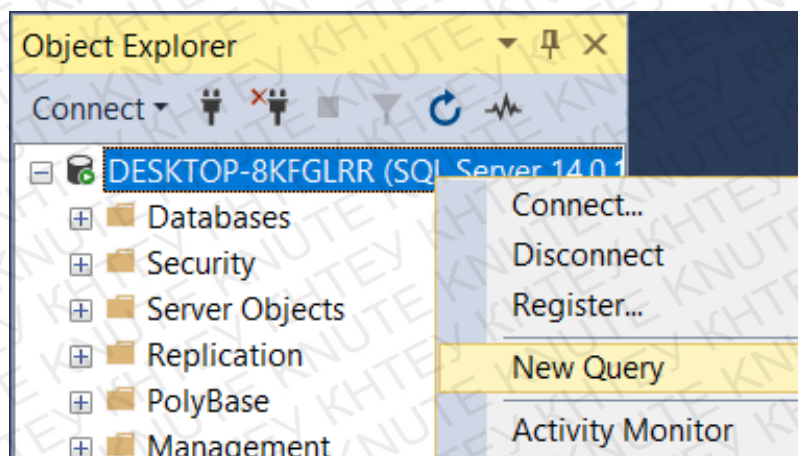


Рис. 2.4. Створення нового запиту

У центральне поле для введення виразів sql введемо наступний код:

```
CREATE DATABASE AutoParts;
```

Для виконання команди натиснемо на панелі інструментів на кнопку Execute або на клавішу F5. І на сервері з'явиться нова база даних.

Після створення бази даних, ми можемо встановити її в якості поточної за допомогою команди USE AutoParts;

Після цього створюємо всі таблиці за допомогою команди

```
CREATE TABLE;
```

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КХТЕУ 121 07-04.БР	22


```

CREATE TABLE Workers
(
    ID_Work INT PRIMARY KEY IDENTITY,
    LastName NVARCHAR(100),
    FirstName NVARCHAR(100),
    Email NVARCHAR(100),
    Phone NVARCHAR(100),
    ALogin NVARCHAR(100),
    APassword NVARCHAR(100)
)

```

```

CREATE TABLE Customers
(
    ID_Cust INT PRIMARY KEY IDENTITY,
    LastName NVARCHAR(100),
    FirstName NVARCHAR(100),
    Email NVARCHAR(100),
    Phone NVARCHAR(100)
)

```

```

CREATE TABLE Suppliers
(
    ID_Supp INT PRIMARY KEY IDENTITY,
    Title NVARCHAR(100),
    Address NVARCHAR(100),
    Phone NVARCHAR(100),
    Email NVARCHAR(100)
)

```

```

CREATE TABLE SupplyDetail
(
    ID_SupDet INT PRIMARY KEY IDENTITY,
    SuppDate DATE,
    Amount INT,
    Price MONEY,
    ID_Goods INT
)

```

```

CREATE TABLE BodyType
(
    ID_TypeB INT PRIMARY KEY IDENTITY,
    NameOfType NVARCHAR(100)
)

```

						Арқуш
Зм.	Арқуш	№ докум	Підпис	Дата		23

KHTEY 121 07-04.БР


```
CREATE TABLE CarBrand
(
  ID_CBrand INT PRIMARY KEY IDENTITY,
  Title NVARCHAR(100),
  ID_Country INT
)
```

```
CREATE TABLE Model
(
  ID_Model INT PRIMARY KEY IDENTITY,
  Model NVARCHAR(100),
  ID_CBrand INT,
  ID_TypeB INT
)
```

```
CREATE TABLE Country
(
  ID_Country INT PRIMARY KEY IDENTITY,
  Country NVARCHAR(100)
)
```

```
CREATE TABLE DetailBrand
(
  ID_DBrand INT PRIMARY KEY IDENTITY,
  Title NVARCHAR(100),
  ID_Country INT
)
```

```
CREATE TABLE Goods
(
  ID_Goods INT PRIMARY KEY IDENTITY,
  Detail NVARCHAR(100),
  Price MONEY,
  Amount INT,
  Photo IMAGE,
  ID_Model INT,
  ID_TypeD INT,
  ID_Supp INT,
  ID_DBrand INT
)
```

```
CREATE TABLE TypeOfDetail
(
  ID_TypeD INT PRIMARY KEY IDENTITY,
  NameOfType NVARCHAR(100)
)
```

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		24

KHTEY 121 07-04.БР

```

CREATE TABLE Orders
(
    ID_Order INT PRIMARY KEY IDENTITY,
    OrderDate DATE,
    Amount INT,
    ID_Goods INT,
    ID_Cust INT,
    ID_Work INT
)

```

Рис. 2.5. Створення таблиць

Для виконання команди натиснемо на панелі інструментів на кнопку Execute або на клавішу F5. І в базі даних з'являться наші таблиці.

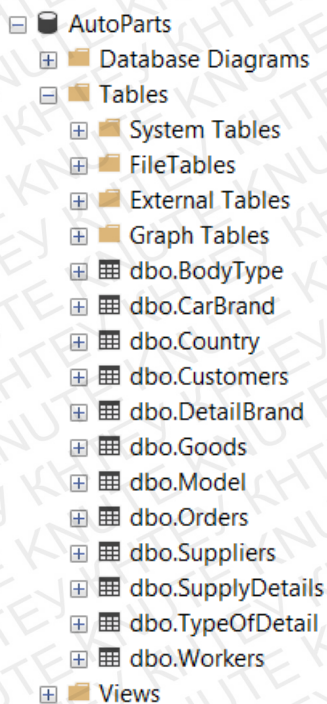


Рис. 2.6. Готові таблиці

Щоб створити зв'язки між таблицями, потрібно створити діаграму бази даних за допомогою вкладки «Database Diagrams». Після створення діаграми та завантаження на неї всіх таблиць бази даних потрібно з'єднати поля таблиць з

						Аркуш
						25
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	

однаковими назвами, а у вікні зв'язку вибрати поле, яке є первинним ключем, а яке – зовнішнім.

The screenshot shows a dialog box titled "Tables and Columns". It contains the following fields:

- Relationship name:
- Primary key table: (with a dropdown arrow)
- Foreign key table:
- Primary key field:
- Foreign key field:

At the bottom right, there are two buttons: "OK" and "Cancel".

Рис. 2.7. Встановлення зв'язків між таблицями

У результаті цього буде встановлено зв'язок між двома таблицями на основі відношення між полями таблиць з однаковою назвою.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

КНТЕУ 121 07-04.БР

2.4. Висновки до розділу 2

У другому розділі була виконана постановка задачі на побудову бази даних. Були визначені головні вимоги до функцій бази даних, кількості таблиць та полів необхідних для майбутньої програмами. Після цього були спроектовані логічна та фізична модель бази даних. На основі розроблених фізичної моделі, була створена нова база, яка містить 12 таблиць та всі необхідні поля для даних. Також, було становлено зв'язок між таблицями та основі первинних та вторинних ключів.

У результаті ми отримали готову базу даних зі встановленими зв'язками між таблицями, яка повністю відповідає вимогам даного проекту.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	27

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМИ

3.1. Створення проекту та підключення бази даних

Для створення нового проекту, після запуску Visual Studio, зліва під написом «Пуск» натискаємо «Створити проект...». У наступному вікні обрати тип проекту – Windows Forms Application, встановити шаблон Visual C#, та зберегти проект у деяку папку на комп'ютері. У результаті цього буде створену стандартну форму додатку типу «Windows Forms Application».

Оскільки база даних є важливою складовою даного проекту, наступним кроком стане підключення БД до програми.

Щоб підключити базу даних потрібно викликати команду «Додати нове джерело даних» з меню «Джерела даних». У вікні вибору джерела бази даних обираємо «База даних», далі – «Набір даних». У вікні вибору підключення до бази даних обираємо «Створити підключення...», обираємо джерелом даних «Файл бази даних Microsoft SQL Server» та вказуємо шлях до файлу. У наступному вікні відмічаємо усі таблиці та натискаємо «Готово». Після цього база даних підключена до нашого проекту, але, для того щоб програмно оперувати таблицями, потрібно підключити бібліотеку System.Data.SqlClient. Для цього в коді основної форми потрібно прописати рядок – «using System.Data.SqlClient;». Також потрібно написати властивість яка містить інформацію про розміщення файлу бази даних на диску. У нашому випадку це:

Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 07-04.БР</i>			
Зав. кафедри		Криворучко О.В.			<i>Розробка програмного забезпечення інтернет-магазину продажу автозапчастин</i>	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.				РЗ	28	48
Гарант		Цензура М. О.				Факультет інформаційних технологій, 4 курс, 7 група		
Розроб.		Горбулін І.О.						
					<i>Реалізація програми</i>			

```
public string _connection = @"Data Source=asusk501;
Initial Catalog = AutoParts;
Integrated Security = True";
```

Рис. 3.1. Підключення бази даних до проекту

Після цього можна переходити до розробки форми додатку.

3.2. Створення та програмування форми «Авторизація»

Гарантування конфіденційності та збереження цілісності даних є важливою складовою будь-якого сучасного проекту. Тому, було вирішено створити форму авторизації, яка б дозволила користуватися програмою тільки тим особам, які мають логін та пароль доступу.

Рис. 3.2. Форма авторизації

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29

КНТЕУ 121 07-04.БР

Форма має містити два поля для вводу та кнопку входу. У разі, якщо користувач введе правильний логін та пароль, та натисне кнопку «Увійти», форма «Авторизація» закриється та відкриється головна форма програми. Якщо введено невірний логін або пароль, з'явиться попередження про невірність вводу даних.

Для реалізації цього потрібно запрограмувати метод обробки події кліку по кнопці «Вхід» та записати до методу наступні рядки:

```
string CommandText =
"select * from Workers
where ALogin = '" + login.Text.Trim() + "' "+
"and APassword = '" + pass.Text.Trim() + "'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, _connection);
DataTable table = new DataTable();
dataAdapter.Fill(table);

if (table.Rows.Count > 0)
{
    this.Hide();
    logProgress lp = new logProgress();
    lp.ShowDialog();
    Parts P = new Parts();
    P.ShowDialog();
}
else
    MessageBox.Show("No");
```

Рис. 3.3. Лістинг форми авторизації

У результаті цього, при завантаженні програми, спочатку відкриється вікно авторизації, і тільки після введення правильного логіну на паролю відкриється основне робоче вікно.

3.3. Створення головної форми додатку

Головна форма програми зверху містить панель керування на якій

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		30

КНТЕУ 121 07-04.БР

розташовані кнопки додавання, видалення даних, а також кнопки перемикання відображення різних таблиць підключеної бази даних.

Усі іншу робочу область займає елемент «dataGridView». Елемент потрібний для відображення загальних даних з таблиці «Замовлення» та «Товари», в залежності від того, який пункт меню вибрав користувача.

The screenshot shows a web application window titled 'Автомагазин' (Car Store) with the 'AutoParts' logo. Below the header, there are navigation tabs for 'Замовлення' (Orders) and 'Деталі' (Parts). The main content area displays a table with the following data:

Виробник	Деталь	Тип деталі	Авто	Ціна, грн.	Кількість, шт.	Постачальник
HOLA - Голандія	Амортизатор задній	Ходова частина	Toyota Camry - Седан	900,0000	8	Автодімсервіс
BOSCH - Німеччина	Амортизатор передній	Ходова частина	BMW X5 - Кросовер	1200,0000	8	Авто-Оптiмал
Brembo - Італія	Колодки гальмівні	Гальмівна система	Land Rover Range Rover	400,0000	6	Авто-Оптiмал
BOSCH - Німеччина	Набір запобіжників	Електроніка	Volkswagen Passat B5 -	190,0000	9	Автолідер
Meyle - Німеччина	Датчик положення	Деталі двигуна	BMW 535i - Седан	4300,0000	9	Аверс
SKF - Швеція	Свічка запалювання	Система запалювання	Skoda Octavia A7 - ліфтбек	260,0000	6	Автодімсервіс
Patron - Китай	Форсунка паливна	Система запалювання	Renault Duster - Кросовер	1200,0000	6	Автопрайд
Ferodo - Велика	Диск зчеплення	Деталі трансмісії	Mazda 6 - Седан	800,0000	10	Автодімсервіс
Patron - Китай	Амортизатор задній	Ходова частина	Volkswagen Golf IV -	600,0000	8	Автолідер
Delphi - США	Амортизатор задній	Ходова частина	Honda CR-V - Кросовер	800,0000	8	Авто-Оптiмал

Рис. 3.4. Головне вікно програми

Дані відображені в елементі «dataGridView» отримуються з різних таблиць підключеної бази даних. Для того щоб вивести дані, з таблиці «Товари», в елемент «dataGridView1», необхідно створити метод, який містить запит повернення даних з таблиці бази даних згідно мови SQL:

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		31

КНТЕУ 121 07-04.БР


```

string CommandText =
"Select [Goods].[ID_Goods]," +
"[DetailBrand].[Title] + ' - ' + [Country].[Country] AS [Виробник]," +
"[Goods].[Detail] AS [Деталь]," +
"[TypeOfDetail].[NameOfType] AS [Тип деталі]," +
"[CarBrand].[Title] + ' ' + [Model].[Model] + ' - ' + [BodyType].[NameOfType] AS [Авто]," +
"[Goods].[Price] AS [Ціна, грн.]," +
"[Goods].[Amount] AS [Кількість, шт.]," +
"[Suppliers].[Title] AS [Постачальник]" +
"From Goods " +
"Join [Model] ON [Model].[ID_Model] = [Goods].[ID_Model]" +
"Join [CarBrand] ON [CarBrand].[ID_CBrand] = [Model].[ID_CBrand]" +
"Join [BodyType] ON [BodyType].[ID_TypeB] = [Model].[ID_TypeB]" +
"Join [TypeOfDetail] ON [TypeOfDetail].[ID_TypeD] = [Goods].[ID_TypeD]" +
"Join [Suppliers] ON [Suppliers].[ID_Supp] = [Goods].[ID_Supp]" +
"Join [DetailBrand] ON [DetailBrand].[ID_DBrand] = [Goods].[ID_DBrand]" +
"Join [Country] ON [Country].[ID_Country] = [DetailBrand].[ID_Country]";

SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, _connection);
DataSet ds = new DataSet();
dataAdapter.Fill(ds, "Goods");
dataGridView1.DataSource = ds.Tables[0].DefaultView;
dataGridView1.Columns[0].Visible = false;

```

Рис. 3.5. Лістинг методу виводу даних на форму

У цьому запиті змінна «CommandText» відповідає за рядок SQL-запиту який виводить усі дані з таблиці «Товари». Після цього створюється адаптер даних «SqlDataAdapter», який пересилає набори даних з викликаючого процесу і назад. Команда «dataAdapter.Fill(ds, "Goods");» - це заповнення набору даних значенням записів бази даних, а для відображення записів на формі використовується запит «dataGridView1.DataSource = ds.Tables[0].DefaultView;».

Для відображення даних з таблиці «Замовлення» в елементі «dataGridView» потрібно створити другий метод з запитом на виведення даних з бази даних в елемент на формі. Даний запит має лістинг аналогічний попередньому методові, крім змінної «CommandText». Вміст змінної слід змінити на команду виводу даних з іншої таблиці бази даних.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32

КНТЕУ 121 07-04.БР

3.4. Створення другорядних форм

Другорядні форми мають схожу, з головною формою, структуру, але основним завданням другорядних форм є збереження та відображення даних, необхідних при створенні нових записів до основних таблиць – «Товари» та «Замовлення».

Розглянемо процес створення другорядної форми на прикладі форми «Авто». Як і головна форма, вона має панель з елементами керування зверху та елемент «dataGridView» для виводу даних.

Title	Model	NameOfType	Country
BMW	535i	Седан	Німеччина
BMW	X5	Кросовер	Німеччина
BMW	760Li	Седан	Німеччина
Volkswagen	Passat B5	Універсал	Німеччина
Volkswagen	Golf IV	Хетчбек	Німеччина
Audi	A6	Універсал	Німеччина
Audi	Q7	Кросовер	Німеччина
Honda	CR-V	Кросовер	Японія
Land Rover	Range Rover Sport	Кросовер	Велика Британія
Volkswagen	Arteon	Ліфтбек	Німеччина

OK Назад

Рис. 3.5. Форма «Авто»

Для заповнення елемента «dataGridView» даними необхідно створити метод, який містить запит мовою SQL.

```
string CommandText =
"SELECT [Model].[ID_Model]," +
"[CarBrand].[Title]," +
"[Model].[Model]," +
"[BodyType].[NameOfType]," +
"[Country].[Country]" +
"FROM [Model]" +
"JOIN [CarBrand] ON [CarBrand].[ID_CBrand] = [Model].[ID_CBrand]" +
"JOIN [BodyType] ON [BodyType].[ID_TypeB] = [Model].[ID_TypeB]" +
"JOIN [Country] ON [Country].[ID_Country] = [CarBrand].[ID_Country]";

SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, _connection);
DataSet ds = new DataSet();
dataAdapter.Fill(ds, "Model");
dataGridView1.DataSource = ds.Tables[0].DefaultView;
dataGridView1.Columns[0].Visible = false;
```

Рис. 3.6. Лістинг методу виводу даних на форму «Авто»

Усі інші другорядні форми створюються та програмуються за прикладом форми «Авто».

3.5. Створення форми додавання нового запису головної форми

Для додавання нового запису головної форми використовується кнопка «Додати» з панелі керування. В залежності від того який пункт меню вибрав користувач, кнопка може відкривати форму додавання даних до таблиці «Замовлення» або таблиці «Товари» підключеної бази даних. Розглянемо процес додавання ного запису на прикладі форми «Додати деталь».

Для додавання нової деталі в таблицю «Товари» потрібно створити нову форму та її інтерфейс. Форма має містити кнопки вибору необхідних даних з другорядних форм, таких як: «Тип деталі», «Авто», «Постачальники»,

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		34

«Виробники». Необхідно кілька елементів «textBox» для введення назви товару та ціни за одиницю, а також, кнопка додавання зображення товару.

Додати деталь

Назва:

Ціна :

Кількість :



Авто : 

Тип деталі: 

Постачальник: 

Виробник: 



Додати зображення :  

ОК

Назад

Рис. 3.7. Форма додавання нової деталі

Для відображення даних в елементах типу dataGridView потрібно створити додатковий метод Get_Data(), який отримує 2 параметри: table_name – задає назву таблиці в базі даних, num_dG – задає номер елемента dataGridView.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		35

КНТЕУ 121 07-04.БР


```

private void Get_Data(string table_name, int num_dG)
{
    Orders f1 = new Goods();

    string CommandText = "SELECT * FROM ";

    CommandText += table_name;
    SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, f1._connection);
    DataSet ds = new DataSet();
    dataAdapter.Fill(ds, table_name);

    if (num_dG == 1) dataGridView1.DataSource = ds.Tables[table_name].DefaultView;
    if (num_dG == 2) dataGridView2.DataSource = ds.Tables[table_name].DefaultView;
    if (num_dG == 3) dataGridView3.DataSource = ds.Tables[table_name].DefaultView;
    if (num_dG == 4) dataGridView3.DataSource = ds.Tables[table_name].DefaultView;
}

```

Рис. 3.8. Лістинг виводу вибраних даних на форму

Далі, програмуємо подію Form2_Load(), яка виконуватиме створені методи при завантаженні форми.

```

private void Form2_Load(object sender, EventArgs e)
{
    Get_Data("Cars", 1);
    Get_Data("Suppliers", 2);
    Get_Data("TypeOfDetail", 3);
    Get_Data("DetailBrand", 4);
}

```

Рис. 3.9. Лістинг методу «FormLoad»

У методі AddDetail_Load() по черзі викликається метод Get_Data(), який заповнює усі елементи dataGridView.

Після цього потрібно повернутися до програмування головної форми, оскільки додавати новий товар ми будемо саме для неї.

Для виклику AddDetail з основної форми, потрібно запрограмувати кнопку «Додати» з групи на виконання запиту на виклик нової форми та додавання нового рядка в елемент «dataGridView» для головної форми.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		36

КНТЕУ 121 07-04.БР

Код для кнопки «Додати» виглядає так:

```
AddDetail AD = new AddDetail();
if (AD.ShowDialog() == DialogResult.OK)
{
    byte[] image = null;
    FileStream stream = new FileStream(AD.imgLocation, FileMode.Open, FileAccess.Read);
    BinaryReader brs = new BinaryReader(stream);
    image = brs.ReadBytes((int)stream.Length);

    string CommandText =
    "INSERT INTO Goods (Detail,Price,Amount,Photo,ID_Model,ID_TypeD,ID_Supp,ID_DBrand)+
    VALUES (" +
    "'" + AD.textBox1.Text + "'," + AD.textBox2.Text + "'," + AD.textBox3.Text + "',+
    @image,'" + AD.ID_CModel + "','" + AD.ID_TypeD + "','" + AD.ID_Supp + "','" +
    AD.ID_DBrand + "')";

    SqlConnection sql_conn = new SqlConnection(_connection);
    SqlCommand sql_comm = new SqlCommand(CommandText, sql_conn);
    sql_comm.Parameters.Add(new SqlParameter("@image", image));

    sql_conn.Open();
    sql_comm.ExecuteNonQuery();
    sql_conn.Close();
}
else
{
    dataGridView1.Visible = false;
    return;
}
```

Рис. 3.10. Лістинг кнопки «Додати»

Додавання нового запису до таблиці «Замовлення» відбувається за прикладом таблиці «Товари».

3.6. Створення форми додавання нового запису до другорядної форми

Розглянемо процес створення форми додавання нового запису до другорядної таблиці на прикладі форми «Додати постачальника».

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

КНТЕУ 121 07-04.БР

Для додавання нового постачальника до таблиці «Постачальники» бази даних, потрібно створити нову форму. Форма повинна містити 4 елементи textBox:

- textBox1 – «Бренд»
- textBox2 – «Адреса»
- textBox3 – «Телефон»
- textBox4 – «Email»

та 2 кнопки, які будуть відповідати за додавання введених даних та відміну операції.

Додати постачальника

Бренд : _____

Адреса : _____

Телефон : _____

Email : _____

ОК

Назад

Рис. 3.11. Форма «Додати постачальника»

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	38

Оскільки поля класу форми додавання нового білету будуть використовуватися в іншій формі, потрібно змінити права доступу до деяких елементів форми. Для цього у файлі .Designer форми встановлюємо доступ public для усіх елементів типу textBox.

Після цього переходимо на форму додавання нового товару та створюємо метод Add_Supplier() для додавання нових даних в таблицю «Постачальники».

```
AddSupply AS = new AddSupply();
if (AS.ShowDialog() == DialogResult.OK)
{
    string CommandText = "INSERT INTO Suppliers VALUES (" +
        "\"" + AS.textBox1.Text + "\", '" + AS.textBox2.Text + "', +
        '" + AS.textBox3.Text + "', +
        '" + AS.textBox4.Text + "')";

    SqlConnection sql_conn = new SqlConnection(_connection);
    SqlCommand sql_comm = new SqlCommand(CommandText, sql_conn);

    sql_conn.Open();
    sql_comm.ExecuteNonQuery();
    sql_conn.Close();
}
else
{
    return;
}
```

Рис. 3.12. Лістинг методу «AddSupply»

Форма додавання нового запису для інших другорядних форм створюється та програмується за зразком форми «Додати постачальника».

3.7. Створення форми детальної інформації про товар

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		39

Форма детальної інформації про товар повинна містити кілька полів в які буде записуватись інформація про деталь з бази даних, а саме: назва, тип деталі, інформація про виробника, авто, ціну, а також постачальника. Також потрібно створити елемент «PictureBox», в який, з бази даних, буде завантажуватись зображення товару.

Інформація про деталь

Деталь : Колодки гальмівні

Тип деталі : Гальмівна система

Виробник : Brembo

Авто : Land Rover Range Rover Sport

Ціна : 400.0000

Постачальник : Авто-Оптімал



Рис. 3.13. Форма «Інформація про деталь»

Для того щоб отримати доступ до елементів форми, які будуть заповнюватися інформацією, з іншої форми необхідно змінити права доступу на публічні в дизайнері форми.

						Аркуш
						40
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	

Після цього, на головній формі програми, необхідно запрограмувати подію «подвійний клік» по рядку елемента «dataGridView». Для цього, в налаштуваннях елемента, необхідно перейти на вкладку «Події» та два рази натиснути на рядок «CellDoubleClick», після чого автоматично створиться метод-обробник подвійного кліку по рядку.

У створений метод необхідно записати команду завантаження інформації про товар з бази даних у створені раніше поля.

```
int row = this.dataGridView1.CurrentRow.Index;
string ID_God = Convert.ToString(dataGridView1[0, row].Value);

GoodsInfo GI = new GoodsInfo();
string cmd =
"Select [Goods].[Detail]," +
"[Goods].[Price]," +
"[Goods].[Photo]," +
"[CarBrand].[Title] + ' ' + [Model].[Model]," +
"[TypeOfDetail].[NameOfType]," +
"[Suppliers].[Title]," +
"[DetailBrand].[Title]" +
"From Goods " +
"Join [Model] ON [Model].[ID_Model] = [Goods].[ID_Model]" +
"Join [CarBrand] ON [CarBrand].[ID_CBrand] = [Model].[ID_CBrand]" +
"Join [TypeOfDetail] ON [TypeOfDetail].[ID_TypeD] = [Goods].[ID_TypeD]" +
"Join [Suppliers] ON [Suppliers].[ID_Supp] = [Goods].[ID_Supp]" +
"Join [DetailBrand] ON [DetailBrand].[ID_DBrand] = [Goods].[ID_DBrand]" +
"Where [Goods].[ID_Goods] = '" + ID_God + "'";

SqlConnection sql_conn = new SqlConnection(_connection);
SqlCommand sql_comm = new SqlCommand(cmd, sql_conn);

sql_conn.Open();
sql_comm.ExecuteNonQuery();
```

Рис. 3.14. Лістинг методу «CellDoubleClick»

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		41

КНТЕУ 121 07-04.БР

Для завантаження зображення товару з бази даних в елемент «PictureBox» необхідно до методу «CellDoubleClick» дописати команду:

```
SqlDataReader dataReader = sql_comm.ExecuteReader();
dataReader.Read();

if (dataReader.HasRows)
{
    GI.label1.Text = dataReader[0].ToString();
    GI.label3.Text = dataReader[4].ToString();
    GI.label5.Text = dataReader[6].ToString();
    GI.label8.Text = dataReader[3].ToString();
    GI.label11.Text = dataReader[1].ToString();
    GI.label13.Text = dataReader[5].ToString();

    byte[] img = (byte[])dataReader[2];
    if (img == null)
    {
        GI.pictureBox1.Image = null;
    }
    else
    {
        MemoryStream msstream = new MemoryStream(img);
        GI.pictureBox1.Image = Image.FromStream(msstream);
    }
}
sql_conn.Close();
GI.ShowDialog();
```

Рис. 3.15. Лістинг методу «CellDoubleClick»

3.8. Створення форми видалення даних

У разі, якщо необхідно видалити певний запис з таблиці бази даних, потрібно запрограмувати метод обробки події кліку по кнопці «Видалити» на виконання запиту на видалення даних з таблиці. Але спочатку потрібно створити відповідну форму.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		42

КНТЕУ 121 07-04.БР

Видалення

Ви дійсно бажаєте видалити запис?

Так

Ні

Рис. 3.16. Форма видалення даних

Після натискання користувачем на кнопку «Видалити», з'явиться форма «Видалення» із запитанням чи насправді користувач має намір видалити запис. У разі позитивної відповіді, виконається запит на видалення даних з таблиці, записаний у коді головної форми, і запис буде видалено.

Запит на видалення має такий синтаксис:

```
int index;  
string id;  
  
index = dataGridView1.CurrentRow.Index;  
id = Convert.ToString(dataGridView1[0, index].Value);  
  
string CommandText = "Delete From Orders Where [Orders].[ID_Order] = '" + id + "'";  
  
SqlConnection sql_conn = new SqlConnection(_connection);  
SqlCommand sql_comm = new SqlCommand(CommandText, sql_conn);  
  
sql_conn.Open();  
sql_comm.ExecuteNonQuery();  
sql_conn.Close();
```

Рис. 3.17. Лістинг форми видалення

						Аркуш
					КНТЕУ 121 07-04.БР	43
Зм.	Аркуш	№ докум	Підпис	Дата		

3.9 . Висновки до розділу 3

У даному розділі було створено новий проект в середовищі розробки Visual Studio та підключено базу даних до нього. Також, були розроблені та створені форми необхідні для взаємодії з таблицями бази даних . У результаті було створено просту у використанні, а головне функціональну програму згідно технічного завдання та поставленої мети, а головне було засвоєно важливі аспекти програмування додатків мовою С# та програмної взаємодії з базою даних, та основними SQL-командами для обробки запитів до таблиць бази даних.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	44

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У ході роботи було проаналізовано вже існуючі програми для магазинів автозапчастин та виявлено, що розробці такого роду програмам приділяється багато уваги.

Отож, було вирішено створити проект «Інформаційна система обліку та продажу автозапчастин», який дозволить легко керувати усіма даними компанії, пов'язаними з продажами товарів, та матиме зручний інтерфейс, розібратися з яким буде не складно будь-кому.

Оскільки програмний продукт вирішено створювати для операційної системи Windows, після аналізу популярних мов програмування, було вирішено обрати мову програмування C# та середовище розробки Visual Studio, як найкращий варіант для даного проекту. Також було використано MS Sql Server, як спосіб управління базою даних, оскільки він повністю відповідає вимогам проекту та ідеально інтегрується в обране середовище розробки.

Було проведено системний аналіз, визначено цілі даного продукту, була виконана постановка завдання та розроблено алгоритм створення програми.

База даних є важливою складовою даної програми, тож було спроектовано та розроблено базу даних MS Sql Server. Після чого було розроблено користувацький інтерфейс та створено, в середовищі розробки Visual Studio, проект, який містить методи взаємодії між програмою та базою даних.

Оскільки розробка даного програмного продукту є досить актуальною, розробник має пропозицію розробити веб-версію даної системи та почати реалізовувати її для інтернет-магазинів автозапчастин. У цій версії планується

					<i>КНТЕУ 121 07-04.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>	
<i>Зав. кафедри</i>		<i>Криворучко О.В.</i>			<i>Розробка програмного забезпечення інтернет-магазину продажу автозапчастин</i>	<i>ВП</i>	<i>45</i>	<i>48</i>
<i>Керівник</i>		<i>Пашорін В.І.</i>						
<i>Гарант</i>		<i>Цензура М. О.</i>						
<i>Розроб.</i>		<i>Горбулін І.О.</i>						
					<i>Висновки та пропозиції</i>	<i>Факультет інформаційних технологій, 4 курс, 7 група</i>		

додати автоматичну обробку замовлень, зворотній зв'язок з клієнтами та реалізувати розсилку повідомлень на електронну адресу клієнтів з інформацією про новий асортимент товарів чи особливі пропозиції.

Розробка програма була доволі успішною. Було реалізовано значну частину ідей та вимог. Розробка проводилася поетапно: створено технічне завдання, проведено маркетингове дослідження, вибрано мови програмування та платформи для реалізації, розроблено дизайну та, власне, реалізовано задум продукту.

Результатом роботи стала повністю функціонуюча програма «Інформаційна система обліку та продажу автозапчастин», яка отримала досить привабливий інтерфейс та вражаючий функціонал.

						Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 07-04.БР	46

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Основний

1. Веймаер Р., Сотел Р. Опануй самостійно Microsoft SQL Server за 21 день, 2016. – 464 с.
2. Молінаро Е. SQL. Збірник рецептів, 2015. – 672 с.
3. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних. Книга 2. Система управління базами даних та знань / А. Ю. Берко, В. В. Пасічник, О. М. Верес, 2013. – 680 с.
4. Бьюлі А. Вивчаємо SQL / Алан Бьюлі, 2007. – 312 с.
5. Документація Stack Overflow. Microsoft SQL Server, 2018 – 286 с.
6. Коноваленко І. В. Програмування мовою С# 6.0 / Ігор Коноваленко, 2016. – 227 с.
7. Шилдта Г. Повне керування С# / Герберта Шилдта, 2011. – 1056 с.

Інтернет-ресурси

8. Database (DB). [Електронний ресурс] – Режим доступу: <https://searchsqlserver.techtarget.com/definition/database>
9. Java Language basics. [Електронний ресурс] – Режим доступу: <https://developer.ibm.com/tutorials/j-introtojava1/>
10. Learn C++ Programming. [Електронний ресурс] – Режим доступу: <https://www.programiz.com/cpp-programming>
11. MySQL. [Електронний ресурс] – Режим доступу: <https://dev.mysql.com/doc/>

					КНТЕУ 121 07-04.БР			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. кафедри		Криворучко О.В.			Розробка програмного забезпечення інтернет-магазину продажу автозапчастин	Стадія	Аркуш	Аркушів
Керівник		Пашорін В.І.				СД	47	48
Гарант		Цензура М. О.				Факультет інформаційних технологій, 4 курс, 7 група		
Розроб.		Горбулін І.О.						
					Список використаних джерел			

12. SQL Tutorial. [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/sql/>

13. SQL. Introduction for beginners. [Електронний ресурс] – Режим доступу: <https://learntocodewith.me/posts/sql-guide/>

14. SQL-програмування. [Електронний ресурс] – Режим доступу: <https://www.quora.com/What-is-SQL-programming>

15. Бази даних. [Електронний ресурс] – Режим доступу: <http://lib.mdpu.org.ua/e-book/vstup/L5.htm>

16. Мова програмування C#. [Електронний ресурс] – Режим доступу: https://uk.wikibooks.org/wiki/C_Sharp

17. Мова програмування C#. Огляд мови C#. [Електронний ресурс] – Режим доступу: <http://ci-sharp.ru/ukr/Teaching/mova-programmirovaniya-s-obzor-yazika-s.html>

18. Огляд мови програмування C#. [Електронний ресурс] – Режим доступу: http://stud.com.ua/97448/informatika/oglyad_movi_programuvannya

19. Середовище розробки Microsoft Visual Studio. [Електронний ресурс] – Режим доступу: http://informatics.in.ua/programming_csharp/part_01.php

20. СУБД Oracle. [Електронний ресурс] – Режим доступу: <https://uk.wikipedia.org/wiki/Oracle>

					Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	48

КНТЕУ 121 07-04.БР

ДОДАТКИ

Додаток А

```
CREATE TABLE Workers
```

```
(  
    ID_Work INT PRIMARY KEY IDENTITY,  
    LastName NVARCHAR(100),  
    FirstName NVARCHAR(100),  
    Email NVARCHAR(100),  
    Phone NVARCHAR(100),  
    Alogin NVARCHAR(100),  
    APassword NVARCHAR(100)  
)
```

```
CREATE TABLE Customers
```

```
(  
    ID_Cust INT PRIMARY KEY IDENTITY,  
    LastName NVARCHAR(100),  
    FirstName NVARCHAR(100),  
    Email NVARCHAR(100),  
    Phone NVARCHAR(100)  
)
```

```
CREATE TABLE Suppliers
```

```
(  
    ID_Supp INT PRIMARY KEY IDENTITY,  
    Title NVARCHAR(100),  
    Address NVARCHAR(100),  
    Phone NVARCHAR(100),  
    Email NVARCHAR(100)  
)
```

```
CREATE TABLE SupplyDetail
```

```
(  
    ID_SupDet INT PRIMARY KEY IDENTITY,  
    SuppDate DATE,  
    Amount INT,  
    Price MONEY,  
    ID_Goods INT  
)
```

```
CREATE TABLE BodyType
```

```
(  
    ID_TypeB INT PRIMARY KEY IDENTITY,  
    NameOfType NVARCHAR(100)  
)
```



```
CREATE TABLE CarBrand
(
    ID_CBrand INT PRIMARY KEY IDENTITY,
    Title NVARCHAR(100),
    ID_Country INT
)
```

```
CREATE TABLE Model
(
    ID_Model INT PRIMARY KEY IDENTITY,
    Model NVARCHAR(100),
    ID_CBrand INT,
    ID_TypeB INT
)
```

```
CREATE TABLE Country
(
    ID_Country INT PRIMARY KEY IDENTITY,
    Country NVARCHAR(100)
)
```

```
CREATE TABLE DetailBrand
(
    ID_DBrand INT PRIMARY KEY IDENTITY,
    Title NVARCHAR(100),
    ID_Country INT
)
```

```
CREATE TABLE Goods
(
    ID_Goods INT PRIMARY KEY IDENTITY,
    Detail NVARCHAR(100),
    Price MONEY,
    Amount INT,
    Photo IMAGE,
    ID_Model INT,
    ID_TypeD INT,
    ID_Supp INT,
    ID_DBrand INT
)
```

```
CREATE TABLE TypeOfDetail
(
    ID_TypeD INT PRIMARY KEY IDENTITY,
    NameOfType NVARCHAR(100)
)
```



```

CREATE TABLE Orders
(
    ID_Order INT PRIMARY KEY IDENTITY,
    OrderDate DATE,
    Amount INT,
    ID_Goods INT,
    ID_Cust INT,
    ID_Work INT
)

```

Рис. А.1. Створення таблиць бази даних

```

public string _connection = @"Data Source=asusk501;
Initial Catalog = AutoParts;
Integrated Security = True";

```

Рис. А.2. Підключення бази даних до проекту

```

string CommandText =
"select * from Workers
"where ALogin = '" + login.Text.Trim() + "' "+
"and APassword = '" + pass.Text.Trim() + "'";

SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, _connection);
DataTable table = new DataTable();
dataAdapter.Fill(table);

if (table.Rows.Count > 0)
{
    this.Hide();
    logProgress lp = new logProgress();
    lp.ShowDialog();
    Parts P = new Parts();
    P.ShowDialog();
}
else
    MessageBox.Show("No");

```

Рис. А.3. Лістинг форми авторизації

```

string CommandText =
"Select [Goods].[ID_Goods]," +
"[DetailBrand].[Title] + ' - ' + [Country].[Country] AS [Виробник]," +
"[Goods].[Detail] AS [Деталь]," +
"[TypeOfDetail].[NameOfType] AS [Тип деталі]," +
"[CarBrand].[Title] + ' ' + [Model].[Model] + ' - ' + [BodyType].[NameOfType] AS [Авто]," +
"[Goods].[Price] AS [Ціна, грн.]," +
"[Goods].[Amount] AS [Кількість, шт.]," +
"[Suppliers].[Title] AS [Постачальник]" +
"From Goods " +
"Join [Model] ON [Model].[ID_Model] = [Goods].[ID_Model]" +
"Join [CarBrand] ON [CarBrand].[ID_CBrand] = [Model].[ID_CBrand]" +
"Join [BodyType] ON [BodyType].[ID_TypeB] = [Model].[ID_TypeB]" +
"Join [TypeOfDetail] ON [TypeOfDetail].[ID_TypeD] = [Goods].[ID_TypeD]" +
"Join [Suppliers] ON [Suppliers].[ID_Supp] = [Goods].[ID_Supp]" +
"Join [DetailBrand] ON [DetailBrand].[ID_DBrand] = [Goods].[ID_DBrand]" +
"Join [Country] ON [Country].[ID_Country] = [DetailBrand].[ID_Country]";

SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, _connection);
DataSet ds = new DataSet();
dataAdapter.Fill(ds, "Goods");
dataGridView1.DataSource = ds.Tables[0].DefaultView;
dataGridView1.Columns[0].Visible = false;

```

Рис. А.4. Код виводу даних на форму

```

string CommandText =
"SELECT [Model].[ID_Model]," +
"[CarBrand].[Title]," +
"[Model].[Model]," +
"[BodyType].[NameOfType]," +
"[Country].[Country]" +
"FROM [Model]" +
"JOIN [CarBrand] ON [CarBrand].[ID_CBrand] = [Model].[ID_CBrand]" +
"JOIN [BodyType] ON [BodyType].[ID_TypeB] = [Model].[ID_TypeB]" +
"JOIN [Country] ON [Country].[ID_Country] = [CarBrand].[ID_Country]";

SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, _connection);
DataSet ds = new DataSet();
dataAdapter.Fill(ds, "Model");
dataGridView1.DataSource = ds.Tables[0].DefaultView;
dataGridView1.Columns[0].Visible = false;

```

Рис. А.5. Лістинг методу виводу даних на форму «Авто»


```

private void Get_Data(string table_name, int num_dG)
{
    Orders f1 = new Goods();

    string CommandText = "SELECT * FROM ";

    CommandText += table_name;
    SqlDataAdapter dataAdapter = new SqlDataAdapter(CommandText, f1._connection);
    DataSet ds = new DataSet();
    dataAdapter.Fill(ds, table_name);

    if (num_dG == 1) dataGridView1.DataSource = ds.Tables[table_name].DefaultView;
    if (num_dG == 2) dataGridView2.DataSource = ds.Tables[table_name].DefaultView;
    if (num_dG == 3) dataGridView3.DataSource = ds.Tables[table_name].DefaultView;
    if (num_dG == 4) dataGridView3.DataSource = ds.Tables[table_name].DefaultView;
}

```

Рис. А.6. Лістинг виводу вибраних даних на форму

```

private void Form2_Load(object sender, EventArgs e)
{
    Get_Data("Cars", 1);
    Get_Data("Suppliers", 2);
    Get_Data("TypeOfDetail", 3);
    Get_Data("DetailBrand", 4);
}

```

Рис. А.7. Лістинг методу «FormLoad»

```

int index;
string id;

index = dataGridView1.CurrentRow.Index;
id = Convert.ToString(dataGridView1[0, index].Value);

string CommandText = "Delete From Orders Where [Orders].[ID_Order] = '" + id + "'";

SqlConnection sql_conn = new SqlConnection(_connection);
SqlCommand sql_comm = new SqlCommand(CommandText, sql_conn);

sql_conn.Open();
sql_comm.ExecuteNonQuery();
sql_conn.Close();

```

Рис. А.8. Лістинг форми видалення


```

AddDetail AD = new AddDetail();
if (AD.ShowDialog() == DialogResult.OK)
{
    byte[] image = null;
    FileStream stream = new FileStream(AD.imgLocation, FileMode.Open, FileAccess.Read);
    BinaryReader brs = new BinaryReader(stream);
    image = brs.ReadBytes((int)stream.Length);

    string CommandText =
    "INSERT INTO Goods (Detail,Price,Amount,Photo,ID_Model,ID_TypeD,ID_Supp,ID_DBrand)+
    VALUES (" +
    "\"" + AD.textBox1.Text + "\",'" + AD.textBox2.Text + "','" + AD.textBox3.Text + "',+
    @image,'" + AD.ID_CModel + "','" + AD.ID_TypeD + "','" + AD.ID_Supp + "','" +
    AD.ID_DBrand + "')";

    SqlConnection sql_conn = new SqlConnection(_connection);
    SqlCommand sql_comm = new SqlCommand(CommandText, sql_conn);
    sql_comm.Parameters.Add(new SqlParameter("@image", image));

    sql_conn.Open();
    sql_comm.ExecuteNonQuery();
    sql_conn.Close();
}
else
{
    dataGridView1.Visible = false;
    return;
}

```

Рис. А.9. Лістинг кнопки «Додати»

```

AddSupply AS = new AddSupply();
if (AS.ShowDialog() == DialogResult.OK)
{
    string CommandText = "INSERT INTO Suppliers VALUES (" +
    "\"" + AS.textBox1.Text + "\",'" + AS.textBox2.Text + "','" +
    AS.textBox3.Text + "','" +
    AS.textBox4.Text + "')";

    SqlConnection sql_conn = new SqlConnection(_connection);
    SqlCommand sql_comm = new SqlCommand(CommandText, sql_conn);

    sql_conn.Open();
    sql_comm.ExecuteNonQuery();
    sql_conn.Close();
}
else
{
    return;
}

```

Рис. А.10. Лістинг методу «AddSupply»

```

int row = this.dataGridView1.CurrentRow.RowIndex;
string ID_God = Convert.ToString(dataGridView1[0, row].Value);

GoodsInfo GI = new GoodsInfo();
string cmd =
"Select [Goods].[Detail]," +
"[Goods].[Price]," +
"[Goods].[Photo]," +
"[CarBrand].[Title] + ' ' + [Model].[Model]," +
"[TypeOfDetail].[NameOfType]," +
"[Suppliers].[Title]," +
"[DetailBrand].[Title]" +
"From Goods " +
"Join [Model] ON [Model].[ID_Model] = [Goods].[ID_Model]" +
"Join [CarBrand] ON [CarBrand].[ID_CBrand] = [Model].[ID_CBrand]" +
"Join [TypeOfDetail] ON [TypeOfDetail].[ID_TypeD] = [Goods].[ID_TypeD]" +
"Join [Suppliers] ON [Suppliers].[ID_Supp] = [Goods].[ID_Supp]" +
"Join [DetailBrand] ON [DetailBrand].[ID_DBrand] = [Goods].[ID_DBrand]" +
"Where [Goods].[ID_Goods] = '" + ID_God + "'";

SqlConnection sql_conn = new SqlConnection(_connection);
SqlCommand sql_comm = new SqlCommand(cmd, sql_conn);

sql_conn.Open();
sql_comm.ExecuteNonQuery();

```

Рис. А.10. Лістинг методу «CellDoubleClick»