

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти на тему:

«Розробка інформаційної системи для магазину запчастин
сільськогосподарської техніки»

Виконав: студент 4 курсу групи Іт-41

Спеціальності 126 «Інформаційні системи та
технології»

(шифр і назва)

Сулятицький М.І.

(Прізвище та ініціали)

Керівник: Запорожцев С.Ю.

(Прізвище та ініціали)

Рецензент: Стукалець І.Г.

(Прізвище та ініціали)

ДУБЛЯНИ-2023

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

Освітній ступінь «Бакалавр» за спеціальністю –
126 – „Інформаційні системи та технології”

“ЗАТВЕРДЖУЮ”

Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 202_ р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Сулятицький Максим Іванович

1. Тема роботи: «Розробка інформаційної системи магазину запчастин для сільськогосподарської техніки»

Керівник роботи Запорожцев Сергій Юрійович, к.т.н., доцент.

Затверджені наказом по університету від _____ лютого 2023 року № _____ /к-с.

2. Строк подання студентом роботи 2.06.2023 р.

3. Початкові дані до роботи: 1. Технологічна карта процесу роботи магазину запчастин для сільськогосподарської техніки 2. ДСТУ, СНіПи.

4. Зміст розрахунково-пояснювальної записки:

Вступ

1. Аналіз предметної області.

1.1. Сучасний рівень механізації та автоматизації в сільському господарстві.

1.2. Особливості застосування техніки в сільськогосподарських технологічних процесах.

1.3. Класифікація сільськогосподарської техніки.

2. Вибір базового програмного забезпечення.

2.1. Основи розробки баз даних.

2.2. Сучасний стан технологій баз даних.

2.3. Клієнт-серверні технології.

2.4. Особливості обраної СУБД

3. Розробка інформаційної системи магазину запчастин.

3.1. Формування інфологічної моделі.

3.2. Створення даталогічної моделі.

3.3. Побудова логічної моделі.

3.4. Розробка схеми даних.

3.5. Побудова таблиць бази даних

3.6. Форми введення даних

3.7. Побудова запитів до бази даних

4. Охорона праці та безпека в надзвичайних ситуаціяхВисновки та пропозиції.Бібліографічний список.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): Аналіз предметної області. СУБД та середовище розробки. Інфологічна модель системи. Схема даних. Таблиці інформаційної системи. Приклади форм. Запити і звіти.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3	<i>Запорожцев С.Ю., доцент кафедри інформаційних технологій</i>		
4	<i>Городецький І.М., доцент кафедри управління проектами та безпеки виробництва</i>		

7. Дата видачі завдання 19 лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Написання першого розділу та означення головних завдань роботи</i>	19.02 - 21.03.23	
2	<i>Виконання другого розділу та формування початкових даних</i>	22.03 - 11.04.23	
3.	<i>Виконання третього розділу та узагальнення отриманих результатів роботи</i>	12.04 - 11.05.23	
4.	<i>Написання розділу: «Охорона праці»</i>	12.05 - 17.05.23	
5.	<i>Завершення роботи в цілому</i>	24.05 - 02.06.23	

Студент _____ Сулятицький М.І.
(підпис)

Керівник роботи _____ Запорожцев С.Ю.
(підпис)

УДК 004.658.2

Розробка інформаційної системи для магазину запчастин сільськогосподарської техніки. Сулятицький М.І. – Кваліфікаційна робота бакалавра. Кафедра інформаційних технологій – Дубляни, ЛНУП, 2023.

52 сторінки текстової частини, 20 рисунків, 9 таблиць та 15 джерел інформації.

Текстова частина включає вступ, чотири розділи, висновки, список використаних джерел.

У вступі висвітлено важливість інформаційних систем в спеціалізованих магазинах. Підкреслено, яку роль відіграють зараз інформаційні системи на ринку та в сільськогосподарській галузі.

У першому розділі подано аналіз предметної області, де описується технологічний прогрес в сільськогосподарській галузі. Описано сучасний стан механізації та автоматизації в галузі. Представлено класифікацію сільськогосподарської техніки та особливості її застосування в технічних процесах.

У другому розділі описано бази даних, їх різноманітність та сучасний стан технологій. Обґрунтовано вибір системи керування базою даних та клієнт-серверних інструментів.

У третьому розділі розроблено інфологічну, даталогічну та логічну моделі. Створено таблиці та зв'язки між ними. Розроблено схему даних. Описано форми введення та запитів, показано їх результат роботи.

У четвертому розділі розроблені заходи з охорони праці і навколишнього середовища.

На підставі виконаної роботи зроблено відповідні висновки.

Ключові слова: інформаційна система, сільськогосподарська галузь, база даних, магазин запчастин, таблиця, модель даних.

Keywords, information system, agricultural industry, database, spare parts store, table, data model.

ЗМІСТ

Вступ.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Сучасний рівень механізації та автоматизації в сільському господарстві.....	9
1.2 Особливості застосування техніки в сільськогосподарських технологічних процесах.....	10
1.3 Класифікація сільськогосподарської техніки.....	11
2. ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	14
2.1 Основи розробки баз даних.....	14
2.2 Сучасний стан технологій баз даних.....	18
2.3 Клієнт-серверні технології.....	20
2.4 Особливості обраної СУБД.....	22
3. РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ МАГАЗИНУ ЗАПЧАСТИН.....	24
3.1 Формування інфологічної моделі.....	25
3.2 Створення даталогічної моделі.....	28
3.3 Побудова логічної моделі.....	30
3.4 Розробка схеми даних.....	32
3.5 Побудова таблиць бази даних.....	34
3.6 Форми введення даних.....	40
3.7 Побудова запитів до бази даних.....	43
4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	45
4.1 Аналіз стану і заходи поліпшення виробничої санітарії і гігієни праці.....	45
4.2 Обґрунтування організаційно-технічних рекомендацій з охорони праці.....	47
4.3 Пожежна безпека.....	48
Висновки та пропозиції.....	50
Бібліографічний список.....	51

ВСТУП

У сучасному світі, де розвиток сільськогосподарської галузі є надзвичайно важливим і прогресуючим процесом, ефективне управління сільськогосподарською технікою є необхідністю. Магазины запчастин для сільськогосподарської техніки відіграють критичну роль у підтримці функціонування та ремонту с техніки. Однак, багато з цих магазинів стикаються з проблемами управління запасами, замовленнями та обліку, що може призвести до непотрібних затрат, затримок у постачанні запчастин та погіршення задоволення клієнтів.

У період сучасних комп'ютерних технологій велика кількість галузей праці потребують інформаційного вдосконалення. Інформаційні системи зараз відіграють значну роль в управлінні, аналізі та функціонуванні підприємств, фабрик, магазинів тощо.

Мета даної дипломної роботи полягає в розробці функціональної інформаційної системи для магазину запчастин сільськогосподарської техніки.

Завданням дипломної роботи буде оцінка існуючих інформаційних систем та розробка подібної з усіма її компонентами. Важливо розробити ефективну систему для задоволення потреб потенційного користувача.

Об'єктом дослідження в цій дипломній роботі виступає процес замовлення користувачем запчастини у інтернет-магазині.

Предмет дослідження являє собою вирішення проблеми чи декількох проблем на ринку, що пов'язані із швидкістю виконання завдань, зручності та ефективності даного типу інформаційних систем.

За допомогою розробленої інформаційної системи, магазин запчастин матиме можливість ефективно контролювати свої запаси, а клієнти зможуть замовляти необхідні деталі вчасно. Для розробки інформаційної системи будуть використані сучасні методи та інструменти програмування, бази даних та інформаційні технології.

Очікується, що розроблена інформаційна система значно покращить ефективність управління запасами, спростить процес замовлення та постачання запчастин, а також забезпечить точний облік та аналіз даних. Внаслідок цього, магазин запчастин сільськогосподарської техніки зможе знизити витрати, збільшити швидкість обробки замовлень, поліпшити якість обслуговування та забезпечити високий рівень задоволення клієнтів.

Метою дипломної роботи буде розробка функціональної інформаційної системи для даного типу магазину. Для досягнення поставленої мети і розв'язуваних в зв'язку з нею завдань, робота розділена на 4 розділи.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Аналіз предметної області для магазину запчастин сільськогосподарської техніки передбачає детальне дослідження та оцінку ринку сільськогосподарської техніки, ідентифікацію потенційних клієнтів та їх потреб, а також аналіз конкурентного середовища.

Галузь сільського господарства досить сильно розвинулась за останнє століття. Все більше і більше людей, пов'язаних з цією діяльністю впроваджують в процес праці нові досягнення технічного процесу. Тепер замість декілька десятків робітників задля обробки певного сектору угідь, потрібно лише декількох, котрі будуть керувати та обслуговувати трактор, комбайн та іншу техніку. Ручна праця дедалі і дедалі відходить на задній план. І якщо такий спосіб ще популярний серед людей, з приватними полями, які використовуються для задоволення власних потреб харчування, то великі компанії не можливо уявити без механічного втручання. Інформаційний простір зараз є дуже важливим аспектом в сільськогосподарській галузі і з кожним роком набуває все більшої популярності.

Інформаційні системи активно використовують в сільськогосподарській галузі не лише великі та популярні компанії, а й менш розвинуті і навіть приватні підприємства. Насамперед інформаційні системи потрібні для контролювання процесів щодо успішного функціонування, управління та обліку в підприємстві. В даній дипломній роботі буде розроблена інформаційна система для магазину запчастин сільськогосподарської техніки. Дана система буде спеціалізуватись на замовленні конкретного типу запчастини для конкретного типу техніки.

1.1 Сучасний рівень механізації та автоматизації в сільському господарстві

Розвиток сільського господарства вже є досить успішним і активним в порівнянні минулих декілька декад. Він і надалі буде лише набирати обороти і процес праці на полі буде зручніший, більш вдосконаленим та менш затратним в плані часу. Велику роль в розвитку с/г галузі відіграв технологічний процес. З появою перших земле-обробних машин та техніки, старі методи роботи на полі почали відходити на задній план. Коней замінили перші трактори, вози з продукцією почали замінювати першими вантажівками.

В сьогоднішній день рівень механізації набув такого розвитку, що без нього неможливо уявити звичайний процес роботи у сільськогосподарській галузі. Трактори та комбайни стали буденністю, а автомобілі тим паче. Механізація являє собою якраз таки використання тих самих комбайнів, тракторів, поливальної та земле-обробної техніки, сівалок, розпилювачів та інших машин. Це допомагає збільшити продуктивність праці, скоротити час виконання завдань і знизити фізичне навантаження на сільськогосподарських робітників. Наприклад, використання тракторів у полі впливає на швидкість та точність обробки ґрунту, а комбайни дозволяють збирати врожай швидко та ефективно.

Однак, на сьогоднішній день автоматизація стає все більш важливою складовою сучасного сільського господарства. Автоматизовані системи та технології, такі як системи збору та аналізу даних, GPS-навігація, дрони, сільськогосподарські роботи та інші, дозволяють збільшити точність та ефективність сільськогосподарських процесів. Наприклад, системи GPS-навігації уніфікують та автоматизують рух тракторів на полі, дозволяючи досягти більшої точності при виконанні різних операцій, таких як засівання або розпилення. Для прикладу спеціально для потреб агробізнесу EOS Data Analytics планує запуск власного супутникового угруповання EOS SAT, перший супутник якого, EOS SAT-1, вже успішно функціонує на орбіті Землі. Таким чином користувачі платформи отримують доступ до ще точніших та детальніших даних про стан своїх полів, зокрема невеликих. Окрім цього, автоматизація - це вищий

етап розвитку виробництва, на якому працівники вивільняються не тільки від фізичної праці, а й, частково, від функцій контролю за машинами, обладнанням, виробничими процесами та операціями з управління ними.

1.2 Особливості застосування техніки в сільськогосподарських технологічних процесах.

Техніка в галузі сільського господарства відіграє величезну роль. З приходом нових технологій, ручна праця стала менш затребуваною, а майже вся фізична праця робиться машинами. Це дозволяє пришвидшити процес роботи, зробити його ефективнішим, що в подальшому дозволяє розширювати спектр робіт.

Застосування техніки в сільськогосподарських технологічних процесах має свої особливості, оскільки вимагає врахування специфіки даного типу виробництва. Основні особливості застосування техніки включають:

- **Різноманітність задач:** Сільськогосподарські технологічні процеси охоплюють широкий спектр діяльності, від підготовки ґрунту та посіву до збирання врожаю та обробки продукції. Тому техніка використовується для різних завдань, включаючи обробку ґрунту, полив, добрива, обробку пестицидами, збирання та перевезення врожаю.
- **Врахування природних умов:** природні фактори, такі як клімат, ґрунтові умови та рослинні культури впливають на робочі здібності. Техніки повинна бути пристосована до цих умов і забезпечувати ефективну роботу незалежно від них. Наприклад, трактори та комбайни повинні мати достатню прохідність у важкому ґрунті або міцність для роботи в умовах зміни клімату.
- **Економічні обмеження:** Виробництво сільськогосподарської продукції має економічні обмеження, оскільки сільські господарства мають

балансувати між витратами на техніку та очікуваним доходом. Тому вибір техніки повинен бути обґрунтованим з точки зору економічної ефективності, враховуючи вартість техніки, її енергоефективність та потенційну вигоду для сільськогосподарського виробника.

- Технічне обслуговування: Сільськогосподарська техніка потребує регулярного обслуговування та утримання, особливо під час інтенсивного сезону робіт. Забезпечення належного технічного обслуговування є ключовим аспектом ефективного функціонування техніки і підтримання її надійності та тривалості служби.
- Загальна сумісність: У сільському господарстві часто використовуються різні види техніки в різних етапах виробництва. Важливо забезпечити сумісність техніки між собою, щоб забезпечити безперебійну роботу і ефективне використання ресурсів.
- Навчання та навички: Застосування сучасної техніки вимагає наявності належних знань, навичок та навчання персоналу. Оператори техніки повинні мати достатні знання про роботу з конкретними машинами, їх налаштування та управління, а також знати основні принципи обслуговування та діагностики.

Урахування цих особливостей є важливим елементом успішного впровадження техніки в сільськогосподарські технологічні процеси. Враховуючи специфіку галузі, правильний вибір, належне утримання та ефективне використання техніки сприяють покращенню продуктивності, збільшенню якості продукції та сталому розвитку сільського господарства.

1.3 Класифікація сільськогосподарської техніки

Сільськогосподарська техніка – це широке розмаїття машин, технічних засобів і агрегатів, які призначені для підвищення продуктивності праці в сфері с/г за допомогою автоматизації та механізації окремих технологічних процесів і

операцій. На сьогоднішній існують найрізноманітніші види техніки, які можна класифікувати в залежності від шуканих критеріїв.

Найчастіше сільськогосподарську техніку поділяють за видом робіт, що даною технікою виконуються. Для прикладу: збиральна техніка, посівна техніка, обприскувачі, ґрунтообробна техніка, кормозаготівельні машини і так далі. Відштовхуючись від цих класифікацій, кожне з цих понять розділяють на інші підвиди. Наприклад, збиральна техніка являє собою в більшості комбайни, котрі в свою чергу поділяються на зернові, бурякові, картопляні тощо, в залежності від виду культур. В такому випадку вибудовується певна ієрархія с/г техніки.

Окрім виду робіт, техніку також поділяють і за іншими критеріями. Такими як розмір, типом ходової частини, типом кістяка, типом палива тощо. Також потрібно не забувати про додаткові механізми, котрі також мають свою класифікацію. Такими механізмами виступають різного виду луцильники, плуги, обприскувачі, борони і так далі. Ці механізми майже завжди використовуються в парі з тракторами, або іншою рушійною технікою. Їхнє призначення є досить вузьким і використовується лише для певного виду робіт, такі як, культивуація ґрунту, внесення добрив в ґрунт, обприскування врожаю тощо. На відміну від цих механізмів, ті ж самі трактори, що виступають рушійною силою для них, є досить універсальною технікою.

Як було сказано раніше в якості основної «рушійної» сили в с/г виступають трактори (колісні або гусеничні), різних модифікацій і потужностей, які здатні працювати з широким спектром інших типів сільськогосподарської техніки, а саме:

- посівною та посадковою технікою;
- технікою для передпосівної обробки ґрунту;
- технікою для зрошення та поливу;
- технікою, призначеною для догляду за посівами;
- технікою для удобрення.

Для попередньої обробки полів перед їх засівом використовують такі агрегати як:

- плуг;
- борони;
- культиватор;
- луцильник;
- каток.

До посівної та посадкової техніки відносяться:

- сівалки, які передбачають найрізноманітніші варіанти висіву насіння;
- розсадосадильні машини;
- картоплесадильні машини.

Для догляду за посівами використовують:

- культиватори;
- окучувальники;
- полольники;
- машини для підрізання кущів;
- обприскувачі.

Для поливу і зрошення використовують:

- дощувальні машини (далекоструйні та двоконсольні);
- поливні машини на автомобільних шасі;
- багатоопорні поливні пересувні машини;
- стаціонарні машини для поливу

Добрива в ґрунт вносяться за допомогою:

- розкидачів (твердих і рідких мінеральних добрив, органічних добрив);
- машин для внесення рідких органічних добрив (поверхневого і внутріґрунтового внесення);
- машин для внесення напіврідкого гною і хіммеліорантів.

Коли справа доходить до збору врожаю, застосовується збиральна техніка (комбайни, косарки, жнивварки) і техніка для післязбиральної обробки (зерноочисні машини, зерноавантажувачі і зернокидачі).

РОЗДІЛ 2

ВИБІР БАЗОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для розробки інформаційної системи основним елементом виступає база даних MySQL, як власне місце зберігання даних та системи. А також декілька додаткових елементів, для управління і створення інформаційної системи. Такі як, phpMyAdmin – вебдодаток для адміністрування бази даних, а також XAMPP Control Panel для запуску веб-серверів.

2.1 Основи розробки баз даних

База даних (БД) – це структурована сукупність даних, які відображають стан об'єктів певної предметної області та зв'язки між ними. Так як бази даних це не різновид програмного забезпечення, але, скажемо так, лише документи, з якими проводять певні операції. То логічно припустити, що існують прикладні програми, які вже оперують базами даних. Ці програми називають системами керування (або управління) базою даних (СКБД).

Система керування базою даних (СКБД) – це програма, котра дозволяє створювати базу даних та проводити різноманітні операції з даними. Загалом, СКБД виступає такою собі зовнішньою оболонкою, яка встановлює певні правила для користувача або клієнта. Окрім цього, СКБД може працювати з іншими прикладними програмами. Важливо зазначити, що дані при такому підході, коли використовується СКБД, є більш захищені, а структура БД з меншою вірогідністю може втратити адекватний читабельний зміст. Для прикладу візьмемо клієнта, котрий вирішив ввести довільні значення в текстове поле, або електронну таблицю. СКБД таких даних «не пропустить», так як вони не будуть відповідати правилам введення даних або іншим умовам, що будуть закладені розробником бази даних. Приклад схеми роботи СКБД показано на рисунку 2.1.

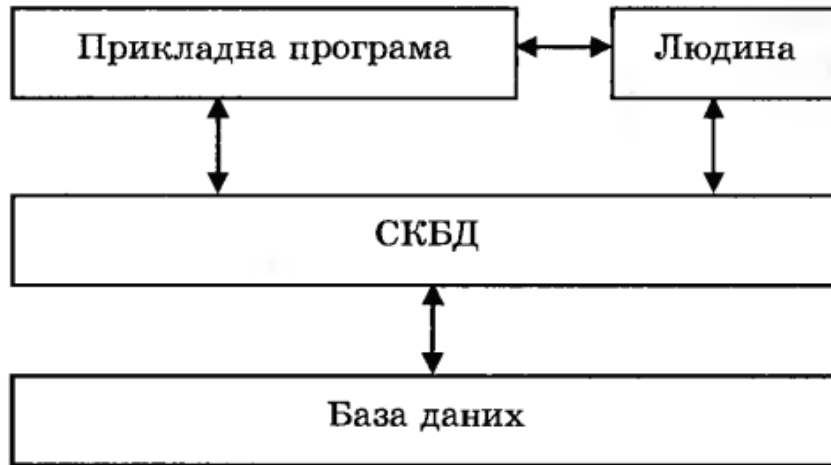


Рисунок 2.1 - Схема роботи системи керування базою даних

Правила і умови введення даних допомагають зберігати цілісність даних, що також є не мало важливою річчю в базах даних. Окрім цього, існують і інші переваги СКБД над прикладними програмами. Такі як: незалежність даних від програм, централізоване зберігання даних та забезпечення спільного доступу для декількох клієнтів або користувачів.

Як було сказано вище, існують правила побудови БД, котрі допомагають дотримуватись цілісності даних. Ці правила називають моделлю даних. Модель даних – це система правил, згідно з якими створюють структури даних, здійснюють доступ до них та змінюють їх. Моделей даних на сьогоднішній день є декілька видів. Інформаційний прогрес не стоїть на місці, тому деякі з них вже поступово виходять з моди, а деякі не використовуються. Для прикладу можна навести ієрархічну та мережеву модель даних, котрих замінила реляційна, яка є досить популярною на сьогоднішній день. Протягом останніх 15-20 років активно розвиваються постреляційні моделі, найбільш перспективною з яких є об'єктно-орієнтована. Проте, її розвиток є досить повільним, тому багато хто вважає, що реляційна модель даних ще не скоро втратить свою популярність.

Створення бази даних відбувається у різних контекстах та способах, в залежності від предметної області та об'єкта предметної області розроблюваної бази даних. Також є залежність від галузі, роду діяльності та інших потреб розробника. Хоч і всі ці перелічені аспекти допомагають і пояснюють

різноманітність баз даних, проте є декілька загальних пунктів, котрі допомагають створити базу даних. Ось декілька з них:

1. Визначення цілі бази даних: База даних, що створюється має мати певну мету, для чого її створюють. Також потрібно розуміти в якій області ця БД буде використовуватись. Наприклад БД для зберігання і керування інформацією про клієнтів, продукти, замовлення, студентів тощо. Важливо чітко зрозуміти, яку інформацію ви хочете зберігати і як вона буде організована.
2. Вибір типу бази даних: Існують різні типи баз даних, такі як реляційні, об'єктно-орієнтовані, графові тощо. Також це називають вибором моделі даних. Цей вибір є надважливим, тому потрібно розуміти, яка з моделей даних найкраще відповідає вашим потребам і вимогам проекту.
3. Проектування структури бази даних: На цьому етапі потрібно розробити схему бази даних, яка визначає таблиці, поля і зв'язки між ними. Це можна порівняти зі створенням скелету БД, на котрий вже буде наноситись м'ясо – значення, дані, текст. Наприклад, якщо ви створюєте реляційну базу даних, вам потрібно визначити таблиці, стовпці (поля) кожної таблиці і зв'язки між ними.
4. Використання запитів: SQL дозволяє створювати таблиці, вставляти, оновлювати, видаляти дані, а також виконувати складні запити для отримання потрібної інформації.
5. Вибір системи керування базами даних (СКБД): Виберіть СКБД, яка підтримує вибраний вами тип бази даних. Наприклад, якщо ви використовуєте реляційну базу даних, популярні СКБД включають MySQL, PostgreSQL та Microsoft SQL Server. Оберіть СКБД, яка найкраще підходить для вашого проекту за орієнтованістю на продуктивність, масштабованість, доступність та іншими факторами.
6. Створити базу даних: Використовуючи обрану СКБД, створіть фізичну базу даних. Це означає створення таблиць з необхідними стовпцями, визначеннями обмежень цілісності даних та інших налаштувань.

7. Розробка програмного забезпечення для доступу до бази даних: Якщо ви плануєте розробити власне програмне забезпечення для доступу до бази даних, вам потрібно створити програми або додатки, які забезпечують зчитування, запис та оновлення даних у базі даних. Це дуже зручний спосіб для роботи з базою даних. Замість того, щоб вносити якісь зміни чи нові дані вручну, програма буде робити більшість маніпуляцій за вас.
8. Забезпечити захист: Важливо приділити увагу захисту бази даних. Застосуйте відповідні заходи безпеки, такі як аутентифікація та авторизація користувачів, шифрування даних, резервне копіювання та відновлення, щоб захистити вашу базу даних від несанкціонованого доступу та втрати даних. Це не обов'язковий етап, але деякі проекти цього потребують.
9. Тестування та оптимізація: Перевірте правильність роботи бази даних, виконуючи різні операції з даними та переконайтеся, що все працює належним чином. Виконайте оптимізацію бази даних для забезпечення ефективності та швидкодії, таку як індексація таблиць, оптимізація запитів тощо.
10. Підтримка та розвиток: Після створення бази даних вона потребуватиме постійної підтримки. Це включає резервне копіювання даних, зміна застарілої інформації на нову, нововведення, та, навіть, можливий перенос БД на іншу структуру чи СКБД, з виходом нових версій програм і тощо.

Як було сказано вище – це загальні етапи створення БД, але це не означає, що саме за таким алгоритмом потрібно створювати власну базу даних. Проте в даній дипломній роботі було використано саме такий порядок дій.

2.2 Сучасний стан технологій баз даних

Цей розділ присвячений сучасному стану технології баз даних, опису ключових розробок та тенденцій, які впливають на спосіб зберігання, обробки та керування даними. Також тут буде інформація про актуальні інновації в області баз даних і, які можливості вони надають для ефективного управління даними.

Реляційні бази даних є одним з основних типів баз даних, які використовуються в сучасних системах. Вони забезпечують стандартизований підхід до організації даних за допомогою таблиць, взаємозв'язків та мови структурованих запитів SQL. Сучасні реляційні системи управління базами даних (RDBMS) надають широкий спектр можливостей, включаючи, оптимізацію запитів, транзакційну надійність та розширені функції для аналітики даних.

Зі зростанням обсягів даних та потребою в гнучкості з'явилися нереляційні бази даних або NoSQL (Not Only SQL). За допомогою них можна використовувати інші моделі зберігання даних, включаючи документи, ключ-значення, стовпці та графи. NoSQL бази даних підходять для великих обсягів даних, розподіленого зберігання, гнучкості схеми та високошвидкісного доступу до даних.

Хмарні технології стали все більш популярними для зберігання та обробки даних. Хмарні бази даних дозволяють користувачам зберігати свої дані на серверах хмарних провайдерів і отримувати доступ до них з будь-якого місця з підключенням до Інтернету. Це забезпечує гнучкість, масштабованість та високу доступність даних. Окрім цього, це дозволяє зберігати місце на пристрої.

Один з ключових аспектів сучасних технологій баз даних - це можливість проводити аналіз даних в реальному часі. Завдяки розробці потужних систем аналітики, компанії можуть отримувати негайну інформацію та реагувати на зміни в режимі реального часу. Це особливо важливо для таких сфер, як фінанси, маркетинг, логістика, де швидка реакція на дані може мати велике стратегічне значення.

Розвиток Інтернету речей призвів до значного збільшення обсягу даних, що збираються та обробляються з різних пристроїв. Базы даних грають ключову роль у зберіганні, обробці та аналізі цих даних. Вони дозволяють збирати, зберігати та аналізувати великі обсяги даних, зібраних від різних датчиків та пристроїв Інтернету речей.

Глибоке навчання є галуззю машинного навчання, що використовує нейронні мережі для розпізнавання та аналізу даних. Сучасні бази даних можуть підтримувати зберігання та обробку великих обсягів даних, які використовуються для тренування та виконання глибоких нейронних мереж. Це надає можливості для розробки складних систем аналітики та прогнозування на основі даних.

Захист даних є надзвичайно важливою темою в галузі баз даних. З поглибленням цифрової трансформації і збільшенням кількості кібератак, безпека даних стала пріоритетом для багатьох організацій. Сучасні технології баз даних включають механізми шифрування, контролю доступу, моніторингу та виявлення вторгнень, щоб забезпечити безпеку даних та запобігти несанкціонованому доступу.

У цьому розділі були розглянуті основні аспекти сучасного стану технологій баз даних. Від реляційних баз даних до нереляційних систем, від хмарних баз даних до аналітики в реальному часі, технології баз даних продовжують еволюціонувати та надавати нові можливості для зберігання, обробки та аналізу даних. Розуміння цих тенденцій є важливим для прогресу у сфері баз даних та використання їх потенціалу для ефективного управління даними.

2.3 Клієнт-серверні технології.

Клієнт-серверні технології є основою багатьох сучасних систем та додатків, які працюють через мережу. Ця архітектурна модель передбачає розподілення функціональності між двома основними компонентами: клієнтом і сервером. Клієнт є кінцевим користувачем або додатком, який ініціює запити та отримує результати від сервера. Сервер, в свою чергу, обробляє ці запити, забезпечує доступ до ресурсів та надає необхідні послуги клієнтам.

У клієнт-серверних технологіях існує кілька основних типів, в залежності від розподілу функціональності та масштабу системи. Основними типами можна представити:

- Клієнт-серверна архітектура з товарами-серверами: У цій архітектурі серверний компонент надає конкретні послуги або функціональність, відомі як "товари". Клієнти використовують ці послуги, взаємодіючи з сервером через мережу. Цей тип архітектури часто використовується в таких сферах, як електронна комерція, соціальні мережі та хмарні сервіси.
- Клієнт-серверна архітектура з базами даних: У цьому типі архітектури серверний компонент відповідає за збереження та управління базою даних, а клієнти отримують доступ до даних через запити до сервера. Цей підхід широко застосовується в системах керування базами даних (СКБД) та додатках, що працюють з великими обсягами даних.
- Клієнт-серверна архітектура з додатками на боці клієнта: У цьому варіанті архітектури клієнтський компонент відповідає за обробку та відображення даних, а серверний компонент забезпечує виконання запитів та надання необхідних ресурсів. Цей тип архітектури часто застосовується в мобільних додатках та веб-додатках.

Клієнт-серверні технології мають ряд переваг, таких як:

- Розподіл функціональності: Клієнтські та серверні компоненти можуть бути розгорнуті на різних платформах та машинних конфігураціях, що дозволяє оптимізувати використання ресурсів та забезпечувати більшу масштабованість.
- Легкість обслуговування: Зміни та поновлення можуть бути здійснені окремо для клієнтських та серверних компонентів, що спрощує процес обслуговування та впровадження змін.
- Забезпечення безпеки: Клієнтські компоненти можуть бути забезпечені захистом даних та аутентифікацією, що дозволяє контролювати доступ до системи та захищати конфіденційну інформацію.

Однак, реалізація клієнт-серверних технологій також має і недоліки:

- Мережева залежність: Клієнти та сервери повинні мати доступ до мережі для взаємодії, що може бути викликом у випадку недоступності мережі або низької швидкості з'єднання.
- Навантаження на сервер: Велика кількість клієнтів може призвести до значного навантаження на сервер, що вимагає масштабування та оптимізації серверних ресурсів.

Клієнт-серверні технології є ключовим елементом сучасних додатків та систем. Вони забезпечують розподілення функціональності між клієнтськими та серверними компонентами, що приносить переваги в ефективності, масштабованості та зручності для користувачів. Однак, їх успішна реалізація вимагає уваги до мережевих аспектів, безпеки та оптимізації ресурсів. Розуміння та використання клієнт-серверних технологій можуть покращити розробку та функціональність додатків, а також забезпечити задоволення потреб користувачів.

2.4 Особливості обраної СУБД.

Системи керування базами даних (СКБД) – це програми або набір взаємопов'язаних даних, котрі мають доступ до баз даних. СКБД надає такі можливості як: створення, збереження, редагування, оновлення, пошуку інформації у базах даних. СКБД були розроблені задля легшого управління базами даних і більш зручнішого проведення операцій з ними.

У цій дипломній роботі було використано досить поширену систему керування базами даних, а саме MySQL. MySQL – це вільна реляційна система керування базами даних, яка працює як сервер для забезпечення багатокористувацького доступу до великої кількості баз даних. Проекти, засновані на вільному програмному забезпеченні, які потребують повнофункціональної системи керування базами даних, часто використовують MySQL. До таких проектів належать, наприклад, WordPress, phpBB, Drupal та інше програмне забезпечення, побудоване на продуктовому стеку LAMP (Linux, Apache, MySQL, PHP/Perl/Python). Окрім цього MySQL використовується в таких популярних сервісах та проектах як: Вікіпедія, Google (для AdWords), Facebook, YouTube, Flickr, Yahoo. Ще одним плюсом даної СКБД є те, що вона безкоштовна і знаходиться у відкритому просторі для користування. Можливості сервера виглядають наступним чином:

- Простота встановлення та використання;
- Необмежена кількість користувачів, які одночасно працюють з базою даних;
- Кількість рядків таблиці може досягати 50 мільйонів;
- Висока швидкість виконання команд;
- Доступ на проста та ефективна система безпеки.

MySQL була розроблена в 1994 році компанією Hughes Technologies. І майже одразу ця СКБД здобула популярність у розробників, заповнивши собою прірву, що існувала між СКБД типу Microsoft Access і комерційними СКБД. Згодом, у 2009 році Oracle Corporation поглинула компанію, до якої входили розробники MySQL, після чого ця СКБД стала власністю Oracle.

Окрім MySQL було використано ще декілька інструментів та програм для розробки інформаційної системи. Одним з таких є XAMPP Control Panel, що дозволяє запустити повноцінний веб-сервер. XAMPP – безкоштовна багатоплатформова збірка веб-сервера з відкритим початковим кодом, що містить HTTP-сервер Apache, базу даних MariaDB, MySQL й інтерпретатори скриптів для мов програмування PHP та Perl.

PhpMyAdmin – це веб-додаток з відкритим вихідним кодом, написаний на PHP з графічним веб-інтерфейсом для керування базами даних MySQL або MariaDB. PhpMyAdmin дозволяє керувати вашим сервером MySQL з браузера, виконувати SQL запити, а також переглядати і редагувати вміст таблиць бази даних. Програма дуже зручна у використанні.

РОЗДІЛ 3

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ МАГАЗИНУ ЗАПЧАСТИН

Інформаційна система – це сукупність організаційних і технічних засобів для зберігання та обробки інформації з метою задоволення інформаційних потреб користувачів. Інформаційні система в сьогодення дуже поширені і використовуються не лише в ІТ-сфері. Технологічний прогрес доріс до того рівня, коли інформаційні системи все дедалі більше використовуються у найрізноманітніших сферах людської діяльності. Очевидно, що сфера сільського господарства не уникнула такої участі. Світові компанії вже давно використовують інформаційні системи різних типів у своїй роботі. Вони використовуються при організації певної роботи чи процесу, в управлінні господарством або обліку та інших робочих моментах. Розробка цієї інформаційної системи має на меті створити і показати процес замовлення запчастини або декількох запчастин для сільськогосподарської техніки у інтернет-магазині. Для кращої розробки, потрібно поділити весь процес на декілька менших за об'ємом та важкістю завдань. Крім цього, потрібно чітко усвідомити, які завдання потрібна виконувати система, та який результат має вийти на виході.

Виходячи з ситуації, у нас існує потенційний покупець, котрий хоче купити запчастину для певного типу с/г техніки. Тобто, вхідними даними для цієї системи буде бажання покупця отримати запчастину, а вихідними – виконане замовлення, отримане покупцем.

В основі розроблюваної інформаційної системи буде лежати база даних, що буде вмщати в собі таблиці. Колонки таблиць будуть містити в собі всю потрібну інформацію про техніку, запчастини, клієнта, постачальника тощо. Для того, щоб інформаційна система працювала належним чином, будуть створені різні типи зв'язків між таблицями.

3.1 Формування інфологічної моделі

Інфологічна модель є одним з ключових компонентів процесу проектування бази даних. Інфологічна модель визначає структуру та взаємозв'язки об'єктів даних, що зберігаються в базі даних. Також ця модель описує сутності, їх атрибути та зв'язки, незалежно від конкретної технології або системи управління базами даних. Для коректної роботи потрібно навести сутності, що зберігають інформацію для СУБД. А саме:

- Про запчастини;
- Про техніку;
- Про замовлення;
- Про клієнта (покупця);
- Про постачальника.

Кожна з цих сутностей має свої власні дані. Сутність «Запчастини» має наступні елементи:

- Код запчастини;
- Код постачальника;
- Ціна запчастини;
- Тип запчастини;
- Виробник.

Сутність «Техніка» містить:

- Код техніки;
- Тип техніки;
- Тип палива;
- Марка.

Сутність «Замовлення» включає в собі:

- Код замовлення;
- Код запчастини;
- Код клієнта;
- Код постачальника;
- Дата та час розміщення замовлення.

У середині сутності «Клієнт (покупець)» містяться наступні дані:

- Код клієнта;
- Прізвище, ім'я, по батькові;
- Електронна пошта;
- Мобільний телефон.

І наостанок сутність «Постачальник» містить в собі:

- Код постачальника;
- Прізвище, ім'я, по батькові або назва фірми, підприємства, фабрики;
- Електронна пошта;
- Мобільний телефон.

За допомогою даних, що наведено вище можна створити інфологічну модель, котра буде показувати всі сутності та дані, що вони включають. Розроблена інфологічна модель зображена на рисунку 3.1.

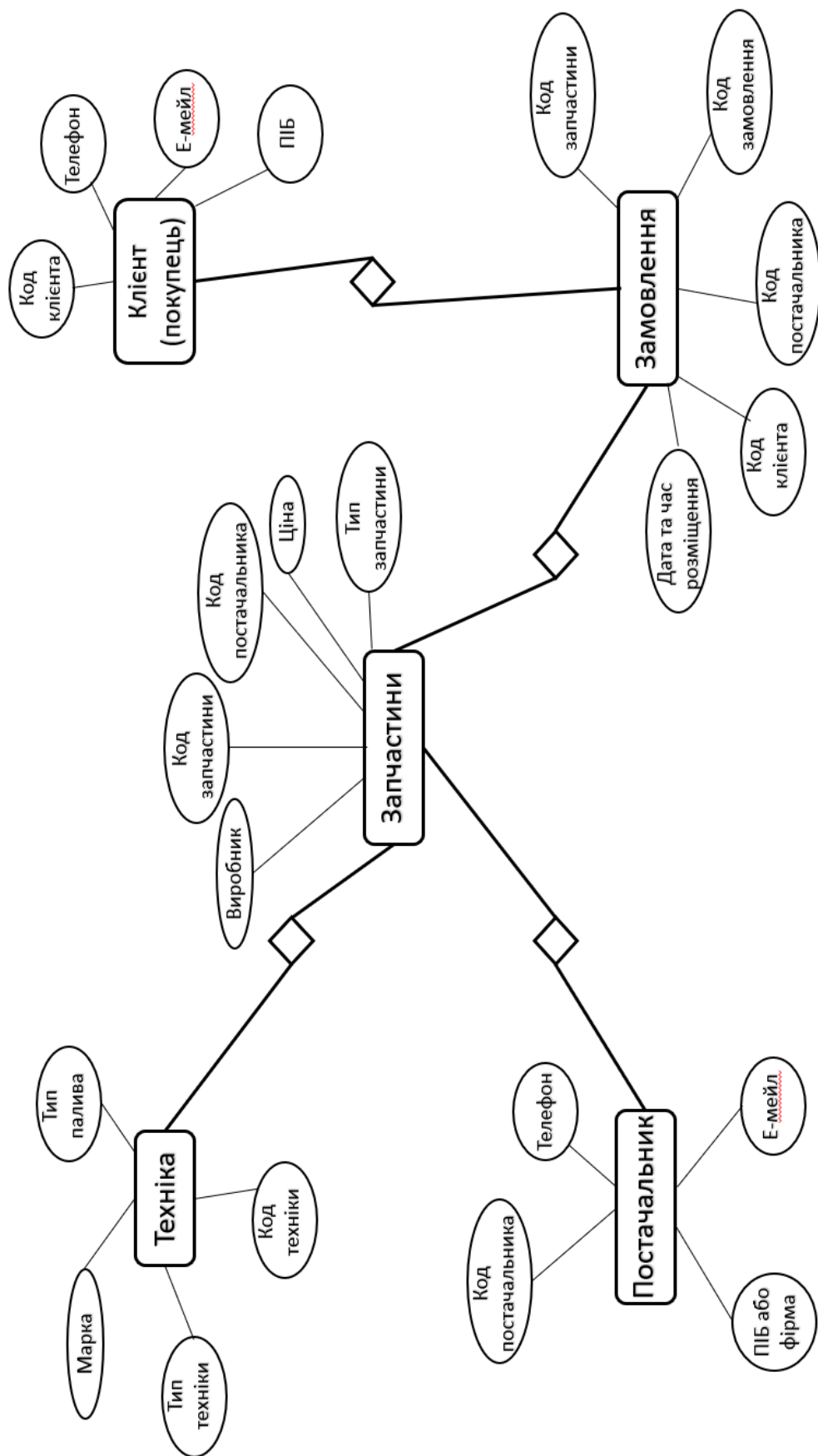


Рисунок 3.1 – Інфологічна модель

3.2 Створення даталогічної моделі

Даталогічна модель даних – це наступний етап проектування бази даних після інфологічної моделі. Даталогічна модель даних визначає структуру даних, використовуючи конкретні типи і структури даних, що надаються обраною системою управління базами даних. Вона визначає, як дані будуть зберігатися та організовуватися в базі даних. Коротше кажучи, інфологічна модель потрібна для наочного розуміння системи та її елементів і сутностей. Для читання її людиною. Даталогічна модель потрібна, щоб сутності та елементи були зрозумілі для СКБД. Це вже не картинка, а таблиці, з певними ключовими полями і зв'язками, з котрими працює СКБД.

У цій дипломній роботі було використано вид реляційної даталогічної моделі. Найбільшою перевагою реляційної моделі від інших є те, що інформація може надаватися у вигляді прямокутних таблиць, де в кожній з яких є унікальне ім'я, рядки та стовпці. Це набагато простіше ніж, для прикладу, мережеві чи ієрархічні моделі. Окрім цього їхні взаємозв'язки також не цілком підходять для розробки такого типу інформаційної системи. Так як реляційна модель бази даних добре працює з типами зв'язків «один-до багатьох» і «багато-до багатьох», то вибір є очевидним. Даталогічна модель представлена в таблицях 3.1 – 3.7.

Таблиця 3.1 - Дані про техніку

Ім'я поля	Тип даних	Є ключовим полем
Код техніки	Лічильник	Так
Марка	Текстовий	Ні
Тип техніки	Числовий	Ні
Тип палива	Числовий	Ні

Таблиця 3.2 – Дані про замовлення

Ім'я поля	Тип даних	Є ключовим полем
Код замовлення	Лічильник	Так
Код постачальника	Числовий	Ні
Код клієнта	Числовий	Ні
Код запчастини	Числовий	Ні
Дата та час розміщення	Дата/час	Ні

Таблиця 3.3 – Дані про запчастини

Ім'я поля	Тип даних	Є ключовим полем
Код типу запчастини	Лічильник	Так
Тип запчастини	Текстовий	Ні
Ціна	Числовий	Ні
Код постачальника	Числовий	Ні
Виробник	Текстовий	Ні

Таблиця 3.4 – Дані про постачальника

Ім'я поля	Тип даних	Є ключовим полем
Код постачальника	Лічильник	Так
ПІБ або фірма	Текстовий	Ні
Телефон	Числовий	Ні
Емейл	Текстовий	Ні

Таблиця 3.5 – Дані про клієнта

Ім'я поля	Тип даних	Є ключовим полем
Код клієнта	Лічильник	Так
ПІБ	Текстовий	Ні
Телефон	Числовий	Ні
Емейл	Текстовий	Ні

Таблиця 3.6 – Дані про тип техніки

Ім'я поля	Тип даних	Є ключовим полем
Код типу техніки	Лічильник	Так
Назва	Текстовий	Ні
Рік випуску	Дата/час	Ні
Url-посилання	Текстовий	Ні
Тип спеціалізації	Числовий	Ні

Таблиця 3.7 – Дані про тип запчастини

Ім'я поля	Тип даних	Є ключовим полем
Код типу запчастини	Лічильник	Так
Назва запчастини	Текстовий	Ні

Таблиця 3.8 – Дані про тип спеціалізації

Ім'я поля	Тип даних	Є ключовим полем
Код типу спеціалізації	Лічильник	Так
Назва спеціалізації	Текстовий	Ні

3.3 Побудова логічної моделі

Логічна модель бази даних є проміжним етапом у процесі проектування бази даних і передує фізичній реалізації. Вона описує структуру даних та їх взаємозв'язки без прив'язки до конкретної системи управління базами даних або фізичної організації даних.

Логічну модель неможливо правильно побудувати при не виконаних попередній етапах. Так як ця модель являється майбутнім прототипом розроблюваної бази даних, то тут вже повинні відобразитись дані нашої

предметної області та взаємозв'язки між ними. Для того щоб успішно розробити логічну модель потрібно використовувати інформацію та дані з минулих моделей. Інфологічна модель надає нам основні сутності та її атрибути (внутрішні дані сутностей), а даталогічна – конкретно розписує властивості атрибутів та їх значення. Окрім основних сутностей, котрі були описані і показані вище, деякі з їх внутрішніх атрибутів потребують додаткових даних та розділення на свої власні сутності. Таким чином, для прикладу, всередині сутності «Запчастини» атрибут «Тип запчастини» має свої власні внутрішні дані. В минулих моделях таке детальне розділення не було показано, проте логічна модель потребує опису абсолютно всіх елементів, значень в них та взаємозв'язків між ними. Також варто зауважити, що на даному етапі роботи були розроблені спеціальні таблиці, котрі можуть забезпечити зв'язок «багато-до-багатьох». Такі таблиці несуть виключно зв'язкову роль між іншими таблицями. Для прикладу – це «Частини техніки», що зв'язує таблиці «Техніка» і «Запчастини».

Отже, за допомогою даних з минулих моделей та розділів зробленої роботи логічна модель набуває наступного виду.

Запчастини (Код запчастини, Тип запчастини, Ціна, Код постачальника, Виробник, Назва запчастини). Або – spare_types (key_spare_part, type_spare_part, price, provider, name_spare_part).

Техніка (Код техніки, Марка, Тип техніки, Тип палива). Або – machinery (key_machinery, brand, type_machinery, type_gas).

Клієнт (Код клієнта, Ім'я, Мобільний телефон, Електронна пошта). Або – client (key_client, name, phone_number, e_mail).

Замовлення (Код замовлення, Код постачальника, Код клієнта, Код запчастини, Дата розміщення). Або – order (key_order, provider, client, spare_part, date).

Постачальник (Код постачальника, Назва фірми або ім'я, Мобільний телефон, Електронна пошта). Або – provider (key_provider, name_or_company, phone_number, e_mail).

Частина техніки (Код частини техніки, Код техніки, Код запчастини). Або – parts_machine (key_part_machine, key_machinery, key_spare_parts).

Частина замовлень (Код частини замовлень, Код замовлення, Код запчастини). Або – part_orders (key_parts_orders, key_orders, key_parts).

Тип техніки (Код типу техніки, Назва, Рік випуску, Url-посилання, Тип спеціалізації техніки) Або – types_machinery (key_types_machinery, name, age, url_link, type_specialization).

Тип спеціалізації (культури) (Код типу спеціалізації, Назва спеціалізації). Або – types_specialization (key_types_specialization, name_specialization).

Тип запчастини (Код типу запчастини, Назва типу запчастини). Або – types_spare_parts (key_type_spare_parts, name_type_spare_part).

3.4 Розробка схеми даних

Щоб СУБД працювала належним чином, на додаток до моделей і зв'язків, сформованих всередині них, необхідно побудувати схему даних. За допомогою PhpMyAdmin та вкладки «Designer» було створено схему даних, що представлена на рисунку 3.2. Процес створення не є важким, всі таблиці добре видно, а в боковому меню зручно створювати або редагувати зв'язки. Зв'язки представлені як лінії, що пов'язують потрібні колонки в таблицях. Це досить зручно, і навіть, якщо будь-хто інший вперше погляне на схему, зможе зрозуміти які таблиці та процеси між собою пов'язані.

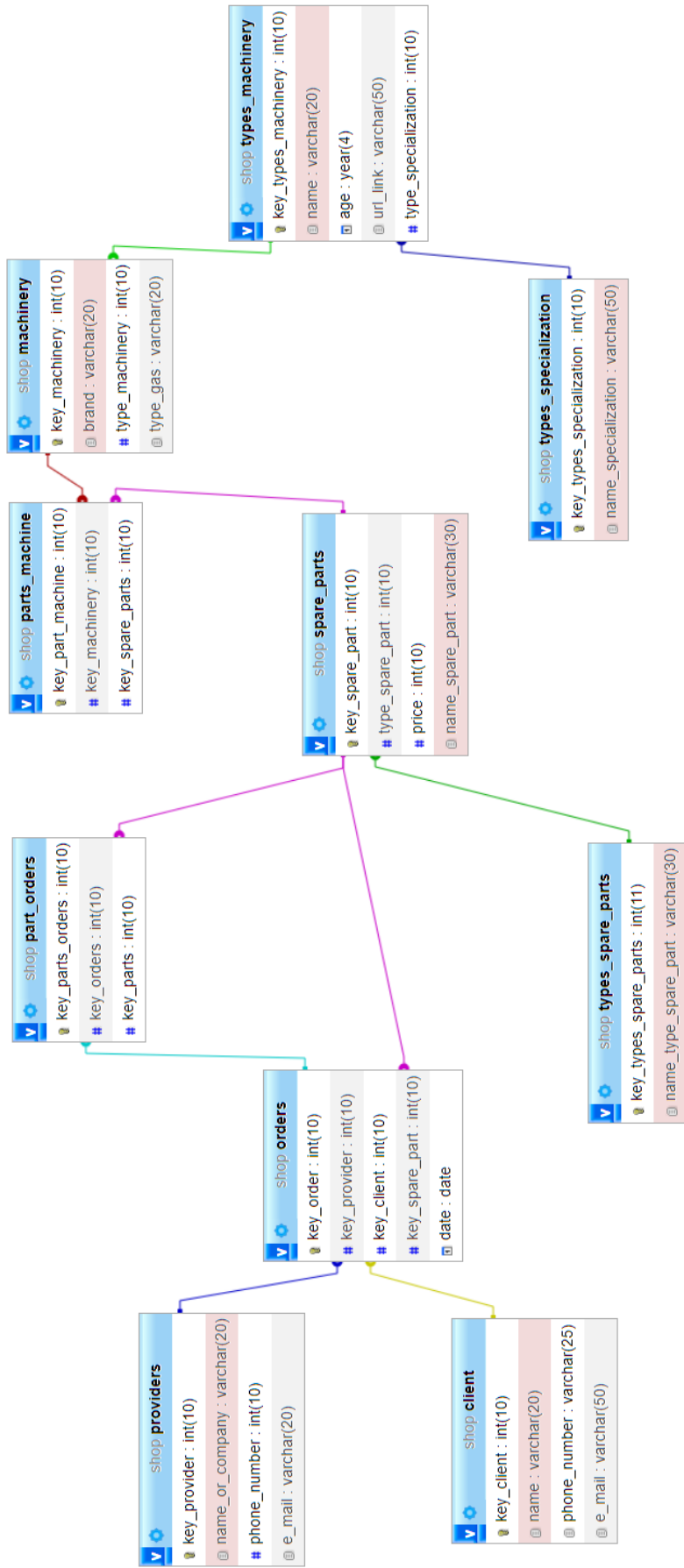


Рисунок 3.2 – Схема данных

3.5 Побудова таблиць бази даних

Для побудови таблиць використовується веб-додаток phpMyAdmin, на сторінці якого створюються, редагуються, оновлюються та видаляються таблиці. Проте, щоб запустити локальний сервер, спочатку потрібно запустити XAMPP Control Panel та в меню цієї програми активувати два додатки – Apache (серверна частина) і MySQL (база даних). Це зробити досить легко, потрібно лише натиснути кнопку «Start» навпроти цих модулів і чекати запуску. Запущений XAMPP Control Panel із активними Apache і MySQL представлено на рисунку 3.3.

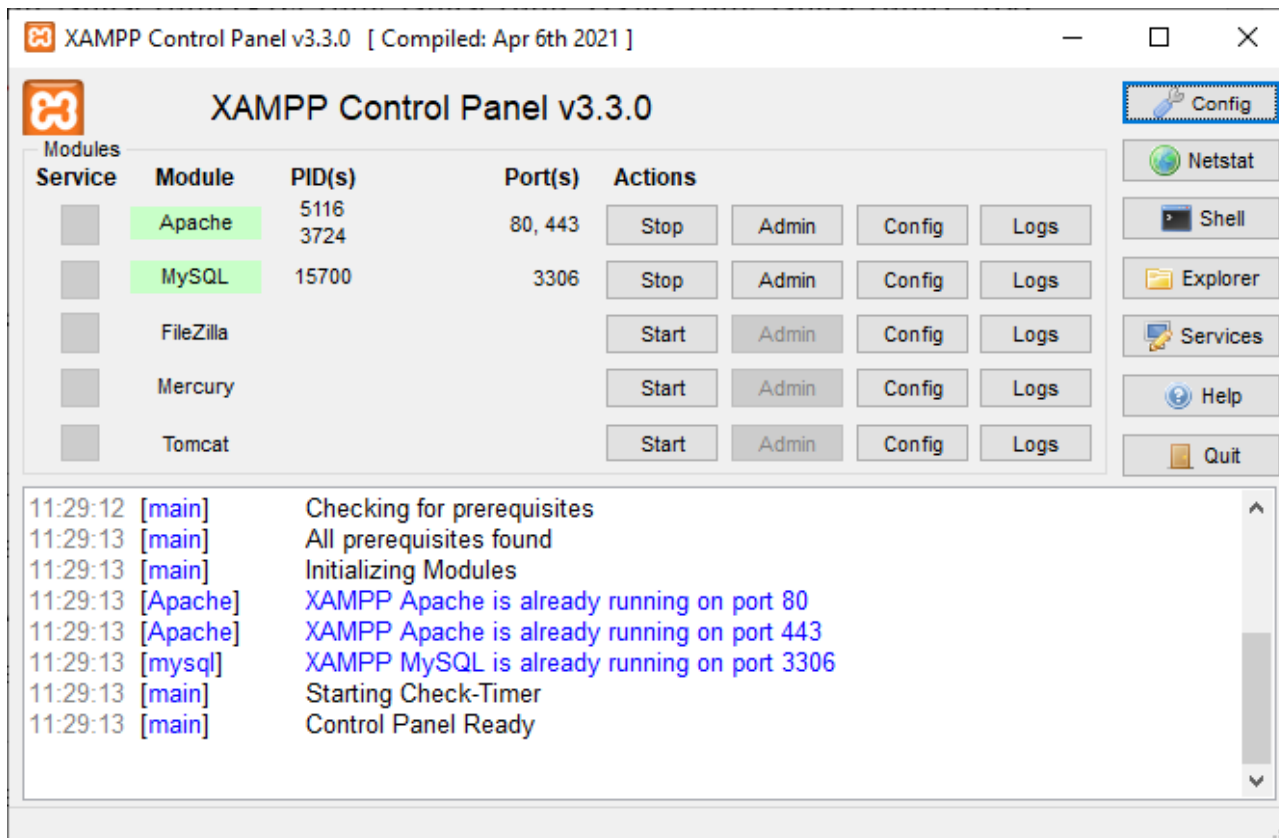


Рисунок 3.3 - XAMPP Control Panel

Наступним кроком потрібно відкрити браузер, в мене це Google Chrome, і вписати в адресний рядок наступне посилання на локальний сервер з базою даних – <http://127.0.0.1/phpmyadmin/index.php>. Після чого відкриється головна

сторінка phpMyAdmin, де будуть налаштування сервера, мови та теми. Головна сторінка phpMyAdmin показана на рисунку 3.4.

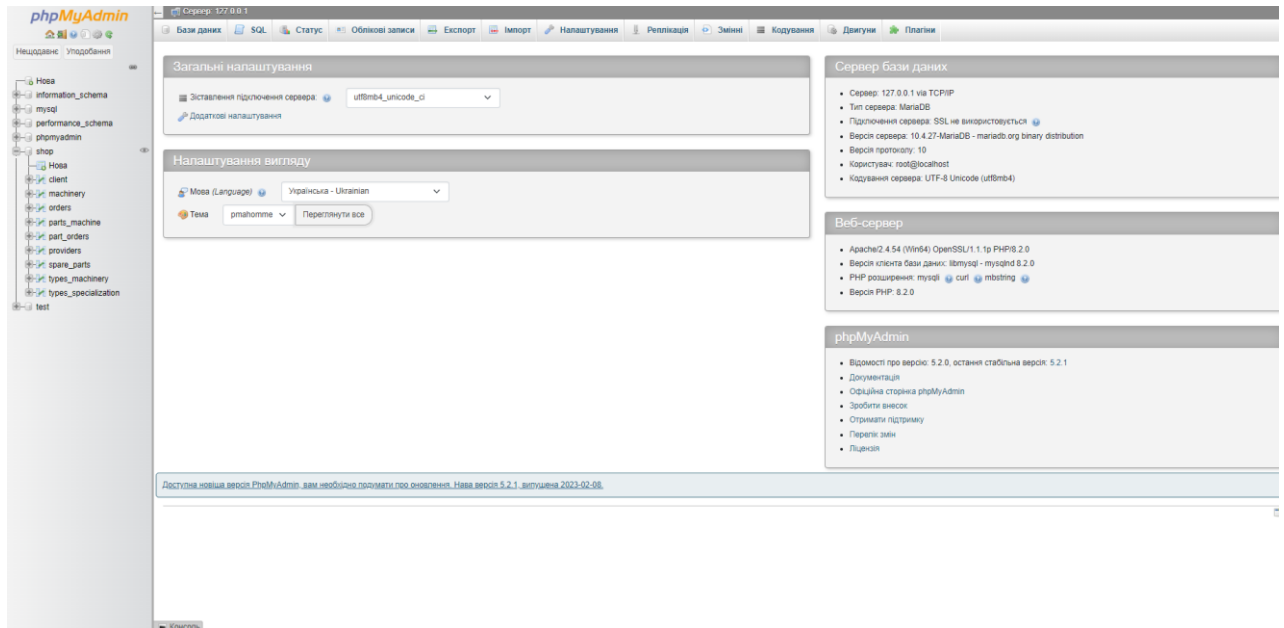


Рисунок 3.4 – Головна сторінка веб-додатку phpMyAdmin

Тепер можна вже перейти до безпосереднього створення бази даних. В меню зліва потрібно натиснути на «Нова» і відкриється сторінка для створення нової бази даних. В полі «Ім'я бази даних» вводимо «shop», так як база даних в нас для магазину запчастин. Вікно створення представлено на рисунку 3.5.

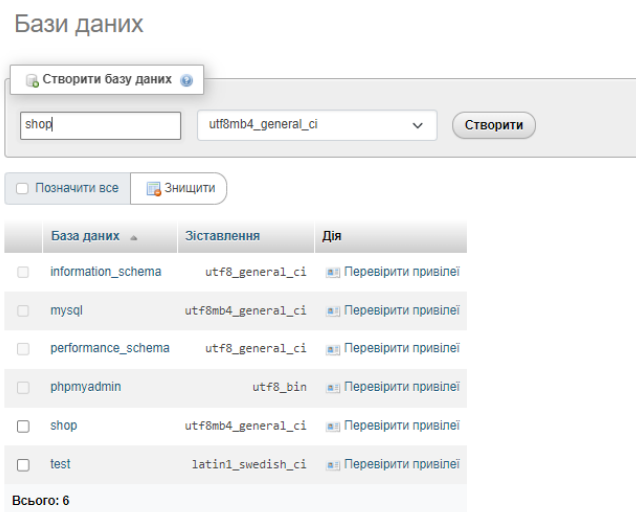


Рисунок 3.5 – Вікно створення бази даних

Тепер в лівому меню з'явилась нова база даних, проте, поки що, вона є пустою. Тому потрібно створити таблиці всередині неї. Для цього в тому ж самому лівому меню, за допомогою знаку плюс, що знаходиться навпроти нашої бази даних, розкриваємо її вміст. Після чого натискаємо на таку ж саму кнопку «Нова» і нам відкриється вікно для створення нової таблиці. Першою таблицею, яку ми створимо будуть власне запчастини. Отож, в вікні створення, в полі «Ім'я таблиці» пишемо «spare_parts». Назви таблиць можна писати і українською мовою, проте для коректної роботи SQL запитів рекомендується називати таблиці англійською. Після створення таблиця, відкривається наступне вікно, де представлені чотири рядки, котрі відповідають за колонки в таблиці. Звичайно ж, їх може бути більше ніж чотири, для цього існує кнопка «Додати стовпець(і)». Кожен стовпець відповідає нашим атрибутам, котрі були розроблені в моделях раніше. Тому для створення стовпців будемо використовувати даталогічну модель, так як зв'язки між ними нам поки що не потрібні. Отже, перший стовпець, як правило створюють з первинним ключем, котрий є у всіх таблицях, і може існувати лише в одному екземплярі. Для цього в полі «Ім'я» вписуємо «key_spare_parts». Поле «Довжина» відповідає за кількість символів у рядку даного стовпця. Так як для цього стовпця не потрібно якогось великого опису, то вписуємо значення «10». І тепер найголовніше, для того щоб наш стовпець набув значення первинного, потрібно активувати прапорець в полі «A/I», а в полі «Індекс» обрати з переліченого «PRIMARY». Наступні поля заповнюємо відповідно до даних в моделях, але без останній двох дій. Результати створення таблиці «Запчастини» показані на рисунку 3.6, а структуру на рисунку 3.7.

Ім'я таблиці: spare_parts Додати 1 стовпець(і) Виконати

Структура

Ім'я	Тип	Довжина/Значення	За замовчуванням	Зіставлення	Атрибути	Нуль	Індекс	A_I
key_spare_part <small>Вибрати з основних стовців</small>	INT	10	Немає			<input type="checkbox"/>	PRIMARY	<input checked="" type="checkbox"/>
type_spare_part <small>Вибрати з основних стовців</small>	INT	10	Немає			<input type="checkbox"/>	--	<input type="checkbox"/>
price <small>Вибрати з основних стовців</small>	INT	10	Немає			<input type="checkbox"/>	--	<input type="checkbox"/>
provider <small>Вибрати з основних стовців</small>	INT	10	Немає			<input type="checkbox"/>	--	<input type="checkbox"/>
name_spare_part <small>Вибрати з основних стовців</small>	VARCHAR	50	Немає			<input type="checkbox"/>	--	<input type="checkbox"/>

Рисунок 3.6 – Створення таблиці «Запчастини»

Сервер: 127.0.0.1 > База даних: shop > Таблиця: spare_parts

Переглянути Структура SQL Пошук Вставити Експорт Імпорт Привілеї Операції Відстеження Тригери

Структура таблиці Вид відносин

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/> 1	key_spare_part	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
<input type="checkbox"/> 2	type_spare_part	int(10)			Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/> 3	price	int(10)			Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/> 4	provider	int(10)			Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/> 5	name_spare_part	varchar(30)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 3.7 – Структура таблиці «Запчастини»

Наступні таблиці будуть створені по такому ж самому прикладі, що і минула. Тому далі буде показано лише результати створення всіх інших таблиць: «Техніка», «Клієнт», «Постачальник», «Замовлення», «Тип техніки», «Тип запчастини», «Тип спеціалізації». Та дві зв'язкові таблиці – «Частина замовлення» і «Частина техніки». Всі процеси створення даних таблиць та їх структура представлено на наступних рисунках 3.8-3.16.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_machinery	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	brand	varchar(20)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
3	type_machinery	int(10)			Ні	Немає			Змінити Знищити Більше
4	type_gas	varchar(20)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 3.8 – Структура таблиці «Техніка»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_client	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	name	varchar(20)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
3	phone_number	int(10)			Ні	Немає			Змінити Знищити Більше
4	e_mail	varchar(20)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 3.9 – Структура таблиці «Клієнт»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_provider	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	name_or_company	varchar(20)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
3	phone_number	int(10)			Ні	Немає			Змінити Знищити Більше
4	e_mail	varchar(20)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 3.10 – Структура таблиці «Постачальник»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_order	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	key_provider	int(10)			Ні	Немає			Змінити Знищити Більше
3	key_client	int(10)			Ні	Немає			Змінити Знищити Більше
4	key_spare_part	int(10)			Ні	Немає			Змінити Знищити Більше
5	date	date			Ні	Немає			Змінити Знищити Більше

Рисунок 3.11 – Структура таблиці «Замовлення»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_types_machinery	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	name	varchar(20)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
3	age	year(4)			Ні	Немає			Змінити Знищити Більше
4	uri_link	varchar(50)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше
5	type_specialization	int(10)			Ні	Немає			Змінити Знищити Більше

Рисунок 3.12 – Структура таблиці «Тип техніки»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_types_spare_parts	int(11)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	name_type_spare_part	varchar(30)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 3.13 – Структура таблиці «Тип запчастини»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_types_specialization	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	name_specialization	varchar(50)	utf8mb4_general_ci		Ні	Немає			Змінити Знищити Більше

Рисунок 3.14 – Структура таблиці «Тип спеціалізації»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	key_part_machine	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
2	key_machinery	int(10)			Ні	Немає			Змінити Знищити Більше
3	key_spare_parts	int(10)			Ні	Немає			Змінити Знищити Більше

Рисунок 3.15 - Структура таблиці «Частина техніки»

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 key_parts_orders	int(10)			Ні	Немає		AUTO_INCREMENT	Змінити Знищити Більше
<input type="checkbox"/>	2 key_orders	int(10)			Ні	Немає			Змінити Знищити Більше
<input type="checkbox"/>	3 key_parts	int(10)			Ні	Немає			Змінити Знищити Більше

Рисунок 3.16 – Структура таблиці «Частина замовлення»

3.6 Форми введення даних

Одним з основних способів збору інформації в електронному середовищі є форми введення даних. Вони широко застосовуються на веб-сторінках, в додатках та інших інтерактивних середовищах, і, в тому числі, в базах даних для збору даних від користувачів.

Форма введення даних складається з різних елементів, які дозволяють користувачам вводити і надсилати інформацію. Основні елементи форми включають поля введення, прапорці, кнопки, випадаючі списки та текстові поля. Ці форми дозволяють набагато швидше і зручніше вводити дані в таблиці.

Набагато простіше відкрити програму, вписати певні поля, а в деяких просто вибрати варіант із перелічених, ніж вносити ці дані вручну. Для цього вже давно існують програми, що містять в собі ці самі форми введення даних. Дані програми найчастіше розробляються під певну базу даних, так як є не складними в розробці. Головною ціллю цих програм має бути зрозумілий GUI (General User Interface), щоб користувач зміг без проблем і легко ввести всі потрібні дані.

Інтернет-магазини використовують форми на сайтах, щоб покупець міг зразу ж обрати товар, замовити доставку тощо. Проте, в даній дипломній роботі було розроблено власну програму для введення даних в таблицю бази даних.

Програма для форми введення даних створювалась за допомогою мови програмування Python в середовищі Anaconda з використанням графічної бібліотеки Tkinter.

Anaconda – це вільно та відкрито розповсюджуваний (freemium) дистрибутив різних програмних продуктів, зокрема, мов програмування Python та R.

Tkinter – багатоплатформна графічна бібліотека інтерфейсів на основі засобів Tk.

Вікно «Замовлення», що представлено на рисунку 3.17 є одним із найосновніших. Дана форма введення надає наступні можливості для користувача:

- введення нових та перегляд існуючих даних;
- зміна даних;
- друк вибраного переліку даних.

MySQL – це серверна система керування базами даних. Вона може містити декілька баз даних на одному сервері. Для взаємодії з базою даних необхідно встановити з'єднання з сервером. Після чого користувач може проводити різні операції, такі як, виведення, введення, оновлення або видалення даних. Інструментом підключення виступає драйвер MySQL Connector/Python, який встановлюється на комп'ютер за допомогою менеджера пакетів «pip». Робочий процес для програми на Python, яка взаємодіє з базою даних MySQL, зазвичай виглядає наступним чином:

- Підключення до сервера MySQL.
- Створення нової бази даних (якщо потрібно).
- Підключення до бази даних.
- Виконання SQL-запитів і збір результатів.
- Повідомлення бази даних про будь-які зміни в таблицях.
- Завершення з'єднання з сервером MySQL.

Щоб встановити з'єднання, використовуємо функцію `connect()` із модуля `mysql.connector`. Дана функція може приймати параметри `host`, `user` і `password` і повертає об'єкт `MySQLConnection`.

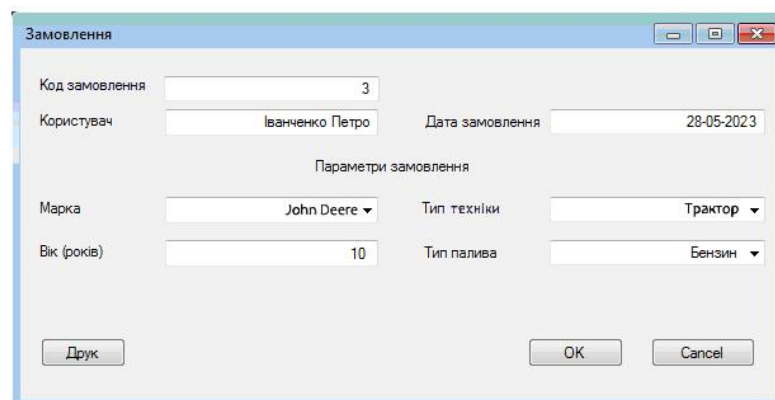
Об'єкт `MySQLConnection` зберігається в змінній `connection`, яку ми будемо використовувати для доступу до сервера `MySQL`. Таким чином, щоб підключитись до нашої бази даних, у функції `connect` знаходимо змінну `database` і прописуємо туди атрибут `shop`, тобто назву нашої БД.

Для вставлення даних у таблицю потрібно прописати: `INSERT INTO «назва таблиці» («дані»)`. А в кінці використати `connection.execute()`, щоб дані були внесені.

Для виведення потрібних даних також використовується `connection.execute()`, проте тепер перед цим потрібно прописати оператор `SELECT`.

За допомогою цих інструментів, бібліотек та методів було розроблено програму з формами введення та виведення даних з БД.

У відкритому вікні «Замовлення» наведено приклад введення даних користувачем. При введенні даних користувач використовує певні поля, де вводить своє прізвище та ім'я та вносить інформацію про вік техніки. Проте інші поля представлені у вигляді вибору із перелічених варіантів. Наприклад, це тип техніки, марка та тип палива.



Код замовлення	<input type="text" value="3"/>	Дата замовлення	<input type="text" value="28-05-2023"/>
Користувач	<input type="text" value="Іванченко Петро"/>		
Параметри замовлення			
Марка	<input type="text" value="John Deere"/>	Тип техніки	<input type="text" value="Трактор"/>
Вік (років)	<input type="text" value="10"/>	Тип палива	<input type="text" value="Бензин"/>

Рисунок 3.17 - Приклад введення даних у базу даних

3.7 Побудова запитів до бази даних

Запит до бази даних – це механізм для отримання, оновлення чи видалення даних із бази даних. Вони дозволяють користувачам взаємодіяти з даними, виконуючи різні операції, такі як вибір (вилучення) певних записів, оновлення значень полів або видалення даних із таблиці.

Як було сказано раніше SQL є стандартною мовою для роботи з реляційними базами даних. Крім того, Structured Query Language дослівно перекладається як структурована мова запитів.

Головним із завдань запитів є генерування нової таблиці, в котрій буде потрібна інформація для користувача. Окрім цього запити спроможні до більш складних операцій, таких як робити обчислення над полями БД, групувати записи і знаходити для полів підсумкові значення за допомогою різних статистичних функцій.

У базах даних існує кілька типів запитів, які можуть бути використані для роботи з даними. Основні типи запитів у базах даних включають:

1. SELECT: Запит SELECT використовується для вибірки (витягування) даних з бази даних. Цей тип запиту дозволяє вибрати певні колонки або всі колонки з таблиці, а також застосувати умови фільтрації даних для отримання конкретного результату.
2. INSERT: Запит INSERT використовується для вставки нових записів у таблицю бази даних. Зазвичай, він містить список полів та відповідних значень, які треба вставити у таблицю.
3. UPDATE: Запит UPDATE використовується для оновлення існуючих записів у таблиці. Він дозволяє змінити значення вибраних полів для вказаних записів у таблиці.

Форми також використовуються в запитах, проте це тепер форми виведення. Вони дозволяють вибрати та вивести певну потрібну інформацію для користувача. В даному випадку, для виведення звітів у вигляді html-сторінки

було використано оператор SELECT. Нижче на рисунку 3.18 показано приклад запиту, котрий виводить дані про замовлення в певний період часу.

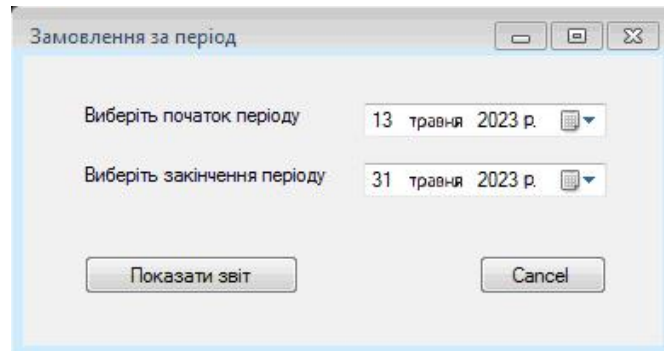


Рисунок 3.18 – Приклад введення запиту

Після чого формується список, в котрий виводиться результат даного запиту. Такий список представлено на рисунку 3.19

Дані про замовлення за період від 13.03.2023 до 31.03.2023:

1. Оформлено замовлень: **3**
2. Користувачі: **3**
3. Доставлено запчастин: **5**

Рисунок 3.19 – Згенерований список замовлень відповідно до запиту

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Охорона праці в секторі інформаційних технологій є актуальним питанням у сучасному суспільстві. Стрімкий розвиток технологій та збільшення використання комп'ютерів та іншого обладнання створили нові виклики та ризики для тих, хто працює в цій галузі. Цей розділ дипломної роботи присвячений детальному дослідженню проблем охорони праці в ІТ-секторі, виявленню їх причин та наслідків, а також пошуку ефективних заходів для покращення безпеки та охорони здоров'я працівників.

4.1 Аналіз стану і заходи поліпшення виробничої санітарії і гігієни праці

ІТ-сфера – це галузь праці, яка являє собою роботу з комп'ютерними пристроями, такими як персональний комп'ютер, ноутбук, смартфон, планшет тощо. То ж в першу чергу потрібно розглянути охорону праці з боку використання даних гаджетів. Працівник майже весь увесь свій робочий час проводить перед монітором чи екраном, сидячи на робочому місці в приміщенні. Такий принцип роботи викликає певні шкідливі чинники, що впливають на фізичне та ментальне здоров'я людини. Найбільш поширеними чинниками, що призводять до погіршення стану здоров'я працівника є:

- Погіршення зору;
- Порушення постави;
- Сидячий метод роботи;
- Довготривалі розумові навантаження;
- Стресові ситуації.

Якщо не запобігти цим чинникам, то в працівників можуть з'явитись професійні хвороби. Серед недугів, які айтішники вважають професійними

хворобами, чимало діагнозів. Зокрема: остеохондроз, ожиріння, біль у шиї та спині, сколіоз, міжхребцеві грижі, геморої та закрепи, зниження зору, проблеми із органами травлення та виразка шлунку, безсоння, мігрень, гіпертонія, тривожні розлади і депресія, невроз тощо.

На сьогоднішній день є чимало досліджень на тему проблем зі здоров'ям працівників в ІТ. Для запобігання будь-яких з перелічених синдромів чи захворювань існують певні комісії чи відповідальні люди, що допомагають передувати недугам на підприємствах чи компаніях. Процес охорони здоров'я відрізняється в залежності від компаній, проте існують загальні стандарти і правила. Наприклад, інструктажі по охороні праці, рекомендації щодо роботи за комп'ютером та планові перевірки персоналу.

Дані процеси позитивно впливають на робочий колектив і в реальному часі впроваджуються все новіші розробки і вдосконалення для збереження здоров'я. Наприклад, останнім часом набувають популярності спеціальні комп'ютерні окуляри, в яких лінза захищає око від синього спектру екрану. Це не виключає повноцінного псування зору, проте дозволяє очам менше напружуватись під час роботи перед монітором.

Окрім цього, потрібно знати правила поводження з електроприладами, та як діяти під час не штатної ситуації. Перед включенням електроприладу необхідно візуально перевірити електрошнур на наявність механічних порушень. Електроприлади повинні бути надійно заземлені згідно з правилами улаштування приладу. Забороняється працювати з електроприладами вологими руками. Не залишати електроприлад без нагляду на довгий час, після закінчення роботи перевірити, чи всі прилади вимкнені. При виявленні або виникненні несправності в електроприладі негайно викликати електрика, що обслуговує прилад. Категорично заборонено виконувати будь-які ремонтні роботи самостійно.

4.2. Обґрунтування організаційно-технічних рекомендацій з охорони праці

Другий аспект з яким стикається працівник в сфері інформаційних технологій – це його робоче місце та приміщення, в якому він працює. Безпека під час роботи є надважливою складовою, тому існують наступні норми та правила:

- Освітлення. Середня величина освітленості приміщення повинна коливатися в межах близько 800-100 люксів. Досягти даної норми можна як і за природного освітлення вдень, так і зі штучним – у вечірню пору. Рекомендується використовувати LED-лампи з денним світлом в якості штучного освітлення;
- Заземлення. Робоче приміщення повинно мати один або декілька заземлювачів, що гарантують вивід струму під час короткого замикання чи інших ситуацій, при яких з'являються різниці струми;
- Блискавкозахист. Важлива сукупність технічних заходів і засобів для зменшення збитку при ударі блискавки в приміщення;
- Первинні засоби гасіння пожежі. Наявність засобів для гасіння пожеж, таких як різного виду вогнегасники. Кнопки для виклику пожежної бригади та увімкнення тривоги у приміщенні. Датчики задимлення, що сповіщають про задимлення у приміщенні;
- Вентиляція. Мінімальна кількість свіжого повітря становить приблизно 60 метрів кубічних на одну людину. Відштовхуючись від цього, в приміщенні повинні встановленні штучні вентиляційні системи, або відкритий доступ до вікон, які забезпечують природну вентиляцію приміщення.

4.3 Пожежна безпека

Пожежна безпека є надважливою складовою в організації робочого процесу в сфері ІТ. Так як ця галузь активно пов'язана з використанням електроприладів в роботі, то ризик пожежної небезпеки є досить великим. Найбільш поширеними приміщеннями в ІТ-сфері є офіси, тому даний розділ потрібно розглядати згідно правил пожежної безпеки в офісних приміщеннях.

Управління з питань надзвичайних ситуацій та цивільного захисту населення Львівської міської ради рекомендує для ознайомлення та виконання всіма працівниками правил пожежної безпеки, які перебувають у службових приміщеннях, а також обслуговуючим персоналом:

- меблі та обладнання необхідно розміщувати таким чином, щоб забезпечувалася вільний евакуаційний прохід до дверей виходу з приміщення (завширшки не менше 1 м). Евакуаційні шляхи та виходи необхідно постійно утримувати вільними, нічим не зашарашувати;

- електромережі, електроприлади і апаратуру експлуатувати тільки у справному стані з урахуванням вказівок та рекомендацій підприємств-виготовлювачів. У разі виявлення пошкоджень електромереж, вимикачів, розеток та інших електровиробів слід негайно вимкнути їх та взяти необхідних заходів щодо приведення в пожежобезпечний стан;

- документи, папір та інші горючі матеріали слід зберігати на відстані не менше 1 м від електрощитів; 0,5 м від електросвітильників; 0,6 м від сповіщувачів автоматичної пожежної сигналізації та 0,15 м від приладів центрального водяного опалення.

- засоби протипожежного захисту слід утримувати у справному стані.

Усі працівники повинні вміти користуватись наявними вогнегасниками, іншими первинними засобами пожежогасіння, знати місце їх знаходження.

Відстань від найбільш віддаленого місця приміщення до місця розташування вогнегасника не повинна перевищувати 20 м.

У службових приміщеннях не допускати:

- влаштування тимчасових електромереж;
- прокладання електричних проводів безпосередньо по горючій основі;
- експлуатація електроприладів, які мають механічні пошкодження;
- захаращування підступів до засобів пожежогасіння;
- куріння, використання легкозаймистих рідин;
- проведення вогневих, зварювальних та інших робіт без спеціального дозволу;
- вмикання електронагрівальних приладів (чайників, кип'ятильників тощо) без негорючих підставок та в тих місцях, де їх використання не передбачено (або заборонено);
- захаращування шляхів евакуації та евакуаційних виходів.

Адміністрація повинна зобов'язати відповідального за протипожежний стан службових приміщень після закінчення роботи:

- оглядати приміщення, переконуватись у відсутності порушень, що можуть призвести до пожежі;
- перевіряти, щоб скрізь було вимкнене освітлення, електроживлення приладів та обладнання (за винятком електрообладнання, яке за вимогами технології повинно працювати цілодобово).

У разі, якщо пожежі не вдалось уникнути, необхідно:

- терміново повідомити пожежну охорону по телефону 101, вказати при цьому адресу, кількість поверхів, місце виникнення пожежі, наявність людей, своє прізвище;
- організувати евакуацію людей та матеріальних цінностей;
- повідомити про виникнення пожежі адміністрацію та чергового (за його наявності);
- вимкнути, у разі необхідності, струмоприймачі та вентиляцію;
- розпочати гасіння пожежі наявними первинними засобами пожежогасіння;
- організувати зустріч підрозділів пожежної охорони й надати їм консультаційну та іншу допомогу в процесі гасіння пожежі.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В даній дипломній роботі було проведено оцінку вже існуючих інформаційних систем, їхній сучасний стан та оцінку ринку.

Було розглянуто класифікацію сільськогосподарської техніки та галузі в цілому. Описано про нововведення та прогрес еволюції сільськогосподарської галузі.

Виконано інфологічне, даталогічне і логічне моделювання. Проаналізовані сучасні методи розробки та керування базами даних на базі SQL та мові програмування Python.

Було розроблено інформаційну систему на основі сьогоденних трендів та методів розробки подібних систем. Наведено інформацію про сучасні бази даних, їхню розробку та системи керування.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Agro Center Spare Parts For Agricultural Machinery. URL: <https://www.agro-center.com.ua/about-us.html> (дата звернення 14.05.2023).
2. EOS Data Analytics. URL: <https://eos.com/uk/blog/drony-ta-suputnyky> (дата звернення 14.05.2023).
3. УкрЗахідІнформ. URL: <http://uzinform.com.ua/news/2016/02/29/128882.html> (дата звернення 16.05.2023).
4. І.О. Завадський Основи баз даних / навч. посібник; рецензенти – П.П. Кулябко, Д.І. Кожем'яка. Київ 2011, 192с.
5. Портал знань. URL: <http://www.znannya.org/?view=mysql-intro> (дата звернення 19.05.2023).
6. DB-Engines. URL: <https://dbengines.com/en/system/MySQL%3BOracle%3BPostgreSQL> (дата звернення 20.05.2023).
7. Apache Friends. URL: <https://www.apachefriends.org/index.html> (дата звернення 20.05.2023).
8. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. Х.: ХНАМГ, 2010. 222 с.
9. Методи та засоби мультимедійних інформаційних систем: навч. посіб. / Т. М. Басюк, П. І. Жежнич; Нац. ун-т «Львів. політехніка». – Львів: Вид-во Львів. політехніки, 2015. 426 с.
10. Anaconda Documentation. URL: <https://web.archive.org/web/20190327135143/http://docs.anaconda.com/anaconda-cloud/faq/#what-is-anaconda-inc> (дата звернення 22.05.2023).
11. Wiki Python. URL: <https://wiki.python.org/moin/TkInter> (дата звернення 22.05.2023).
12. Dev.ua. URL: <https://dev.ua/news/aitishnyky-i-khvoroby-1661439760> (дата звернення 23.05.2023).

13. Львівська міська рада. URL: <https://city-adm.lviv.ua/lmr/socialni-iniciativi/2097-zakhody-pozhezhnoi-bezpeky-u-sluzhbovykh-prymishchenniakh-ofisakh> (дата звернення 23.05.2023).

14. Тимочко В.О., Городецький І.М., Березовецький А.П., Мазур І.Б. та ін. Безпека життєдіяльності та охорона праці. Навч. посібник. Львів: Сполом. 2022. 376 с.

15. Пістун І. П., Березовецький А. П., Тимочко В. О., Городецький І. М. Охорона праці (гігієна праці та виробнича санітарія): навч. посіб. / за ред. І.П.Пістуна. Львів: Тріада плюс, 2017. Ч. I. 620 с.