

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»

Навчально-науковий інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії

(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

кваліфікаційної роботи ступеня магістра

(бакалавра, спеціаліста, магістра)

студента Сидорова Владислава Івановича

академічної групи 123М-21-1

(ПБ)

(шифр)

спеціальності 123 «Комп'ютерна інженерія»

(код і назва спеціальності)

за освітньо-професійною програмою «Комп'ютерна інженерія»

(офіційна назва)

на тему Обґрунтування структури та параметрів комп'ютерної системи цеху збирання  
ТОВ «Усі мотоблоки»

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Бешта Д.О.			
розділів:				
теоретичний розділ	проф. Цвіркун Л.І.			
синтез системи	доц. Бешта Д.О.			
розроблення програмного забезпечення	ас. Панферова Я.В.			

Рецензент				
-----------	--	--	--	--

Нормоконтролер	проф. Цвіркун Л.І.			
----------------	--------------------	--	--	--

Дніпро  
2022

**ЗАТВЕРДЖЕНО:**  
завідувач кафедри  
інформаційних технологій  
та комп'ютерної інженерії  
(повна назва)

\_\_\_\_\_ Гнатушенко В.В.  
(підпис) (прізвище, ініціали)  
«\_\_\_» \_\_\_\_\_ 2022 року

**ЗАВДАННЯ на кваліфікаційну роботу**

ступеня магістра  
(бакалавра, спеціаліста, магістра)

студенту Сидоров В.І. академічної групи 123М-21-1  
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньою-професійною програмою 123 «Комп'ютерна інженерія»  
(офіційна назва)

на тему Обґрунтування структури та параметрів комп'ютерної системи цеху збирання  
ТОВ «Усі мотоблоки», затверджену наказом ректора НТУ «Дніпровська політехніка»  
від 31 жовтня 2022 р. №1200

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел сформулювати наукове завдання, конкретизувати предмет та мету досліджень	10.10.2022
Теоретичний	Обґрунтувати теоретичну базу розв'язання наукового завдання, якому присвячено роботу	24.10.2022
Синтез системи	Розробка комп'ютерної системи	14.11.2022
Розроблення програмного забезпечення	Розробка програмного забезпечення	28.11.2022
Експериментальний розділ	Проведення і обробка результатів експериментів	05.12.2022
Графічна частина	Графічні результати роботи подати у вигляді рисунків схем таблиць на 10 арк. формату А4.	10.12.2022

Завдання видано \_\_\_\_\_  
(підпис керівника)

доц. Бешта Д.О.  
(прізвище, ініціали)

Дата видачі 10 жовтня 2022 р.

Дата подання до екзаменаційної комісії

15.12.2022 р.

Прийнято до виконання \_\_\_\_\_  
(підпис студента)

Сидоров В.І.  
(прізвище, ініціали)

## РЕФЕРАТ

Об'єкт проектування: комп'ютерна система товариства з обмеженою відповідальністю «Усі мотоблоки».

Мета – вдосконалення комп'ютерної мережі підприємства. Визначте вузли мережі, на які найбільше впливає перевантаження мережевою інформацією.

Визначити умови та параметри властивостей мережевих технічних пристроїв, за яких можлива втрата продуктивності.

Розробити рекомендації щодо модернізації мережі для підвищення її стійкості до перевантажень.

Визначення найбільш слабких вузлів комп'ютерної мережі, які у разі перевантаження призводять до втрати продуктивності мережі та надання рекомендацій щодо вирішення цієї проблеми.

Практичні результати: виявлення недоліків та шляхів удосконалення комп'ютерної мережі досягаються шляхом використання наукового підходу до вирішення поставлених завдань. Прикладні наукові досягнення в моделюванні комп'ютерних мереж показали свою надійність на багатьох прикладах їх застосування.

Результати перевірки у вигляді таблиць і графіків описуються та наводяться в записці з поясненнями та додатками.МОТОБЛОК, ЦЕХ ЗБОРУ, МЕРЕЖА, ТОВАРИСТВО, МАГАЗИН

## ЗМІСТ

Вступ	6
1 Статус питання та постановка завдання дослідження	7
1.1 Загальна характеристика об'єкта управління	7
1.1.1 Характеристика галузі	7
1.1.2 Організаційна структура підприємства та особливості управління ТОВ «Усі мотоблоки»	8
1.2 Аналіз об'єкта впровадження	12
1.3 Аналіз існуючих проблем	13
1.4 Способи обробки	14
1.5 Висновки по розділу	14
2 ТЕОРЕТИЧНИЙ РОЗДІЛ	15
2.1 Концепція баз даних	15
2.2 Аналіз методів і засобів доступу до інформації сервера	19
2.3 Система управління бази даних	20
2.4 Можливості СУБД	23
2.5 Визначення файлу даних: запис, структура даних, поле запиту та його реквізити (ім'я, тип, розмір)	24
2.6 Висновки по розділу	25
3 РОЗДІЛ СИНТЕЗУ СИСТЕМИ	26
3.1 Аналіз вимог до БД	26
3.2 Визначення напрямків рішення поставлених задач	26
3.3 Розробка схеми функціональної структури	31
3.1.1 Вибір елементів бази даних системи	31
3.1.2 Налаштування HTTP та DNS	32
3.1.3 Розробка структури бази	35
3.1.4 Розробка мови ситуативного керування даними	36
3.4 Висновки по розділу	37

4. РОЗДІЛ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	38
4.1 Призначення і область застосування програми	38
4.2 Обґрунтування технічних характеристик програми	38
4.3 Опис і обґрунтування вибору складу технічних і програмних засобів	40
4.4 Структура БД	45
4.5 Вхідні і вихідні дані	50
4.6 Висновки по розділу	50
5. Експериментальний розділ	51
5.1 Формулювання вимог до експерименту	51
5.2 Опис експерименту	51
5.3 Контрольний приклад для проведення випробувань	51
5.3.1 Підготовка до роботи та запуску серверу	51
5.3.2 Дослідження достатності опису і понять	52
5.4 Висновки по розділу	55
ВИСНОВИКИ	56
ПЕРЕЛІК ПОСИЛАНЬ	57
Код програми Додаток А	58

## ВСТУП

В умовах високої конкуренції роздрібних продавців на сучасному ринку величезну роль грають торгово-технологічні процеси організації роздрібною торгівлі. Насичений ринок товарів і послуг створює необхідність не тільки конкурентних переваг за ціною, але і створення оптимальних умов і максимального комфорту при здійсненні покупок.

Компанія займається онлайн та офлайн продажами сільськогосподарської техніки. Засновниками підприємства являються юридичні фірми та фізичні особи. Вони є дилером провідних заводів тракторної техніки. Реалізація продукції відбувається як в Україні так і за кордон.

В даний час електронні бази та облік товару в цифровому форматі дозволяє зберігати інформацію більш надійно та компактно, та працювати з нею в різних куточках країни. Таким чином інформацію краще зберігати в електронній формі, так як це не гонитва за трендами а сучасна потреба.

Роль БД за умов інформаційної складової життя людини важко переоцінити.

Таким чином, мета кваліфікаційної роботи обґрунтувати та розробити систему доступу до бази даних.

Об'єкт дослідження – база даних та інформаційна складова серверу.

Параметри дослідження - склад, параметри, методи, оцінювання серверного обладнання та софту.

Ідея дослідження – облік та розподіл даних бази на різні відділи підприємства за рахунок впровадження нових рішень до серверної частини.

# 1 СТАН ПИТАННЯ І ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

## 1.1 Загальна характеристика об'єкта управління

### 1.1.1 Характеристика галузі

Компанія займається онлайн та офлайн продажами сільськогосподарської техніки. Засновниками підприємства являються юридичні фірми та фізичні особи. Вони є дилером провідних заводів тракторної техніки. Реалізація продукції відбувається як в Україні так і за кордон.

Однією зі сфер діяльності компанії є продаж міні-тракторів. Мінітрактор - це нове покоління потужної, багатофункціональної і економічної техніки, яка знайшла широке застосування в сільському господарстві, а також у будівельній, комунальній та інших сферах народного господарства. В продажі є мінітрактор білоруського (Мінітрактор "Білорус") і китайського (міні-трактори Xingtai, Jinma) виробництва.

Якщо когось з тих чи інших причин не влаштовує трактор, то він може придбати мотоблок. Мотоблок є відмінним варіантом для приватників, комунальних служб і озеленювачів, які мають територію невеликих розмірів. Також як і трактори, мотоблоки представляють собою поєднання садової, комунальної та збиральної техніки, машини обробки ґрунту і косіння трави, створення газону та інших завдань.

Білоруські мотоблоки вже не одне десятиліття користуються попитом. Мотоблок "Білорус" укомплектований потужним - двигуном "Хонда" японського виробництва. Завдяки цьому мотоблок поставляється на міжнародний ринок. Сьогодні він експортується до Болгарії, Туреччини, Німеччини.

В магазині можна придбати навісне обладнання для міні-тракторів, призначене для сільськогосподарських робіт різного рівня складності. Мінітрактор з навісним обладнанням може використовуватися в якості

збиральної техніки, машини для обробки ґрунту, косіння трави, створення газону, подрібнення деревини та інших операцій. Також пропонується навісне обладнання для мотоблоків.

Підприємство розташоване за адресою: Дніпропетровська область, смт Слобожанське, вул. Нова 45.

Технологічний процес товариства та включає наступні операції відвантаження, транспортування, прийом їх по кількості та якості, збір, зберігання, підготовка до продажу та сам продаж.

Вся продукція сертифікована і має гарантійний термін 1 рік.

### **1.1.2 Організаційна структура підприємства та особливості управління ТОВ «Усі мотоблоки»**

Організаційна структура являє собою – адміністрацію та підпорядковані їй цехи. До складу входять:

- керуючий товариством;
- заступник відділу продажу;
- заступник офісу закупівель;
- завідуючий складу;
- головний інженер цеху збору комплектуючих;
- менеджер.

Керівник виконує адміністративні функції та вирішує всі справи компанії.

Він організовує всю роботу магазину і всіх підпорядкованих підрозділів і несе повну відповідальність за його виробничо-господарську діяльність.

Відділ закупівель охоплює всі процеси, безпосередньо пов'язані з дослідженням і закупівлею обладнання.

Він займає центральне місце у всій виробничій діяльності



підприємства.

Від цього в основному залежить виконання виробничих завдань і поліпшення техніко-економічних показників підприємства. Надходження товарів складається з процесу матеріально-технічного забезпечення закупівель.

У цей офіс входить прийом товару з наявністю підтверджуючих документів: накладна, рахунок-фактура, сертифікат відповідності та сертифікат якості або паспорт обладнання.

Складальний цех - це високотехнічне виробництво, де перевіряються всі пристрої. Фахівці офісу збирають комплектуючі будь-якого ступеня складності.

Вся транспортна документація на обладнання проходить відвантаження. При надходженні, наприклад, мототрактора перевіряється правильність оформлення супровідних документів. Кожен технічний засіб слід ретельно вивчати. Комірник підбирає замовлення відповідно до заявки на відвантаження, контролює якість завантаження товару. Технічні засоби повинні бути належним чином закриті та опломбовані. Відбитки штампів повинні бути включені в товаросупровідну документацію.

Обладнання має надходити в магазин повністю готовим до продажу. Продаж товарів (онлайн або офлайн) є завершальним етапом торгово-технологічного процесу в магазині. Операції, які виконуються на цьому етапі, є найбільш відповідальними, оскільки безпосередньо пов'язані з підтримкою клієнтів.

Важливі функції в управлінні підприємством виконує бухгалтерія. Бухгалтерія веде бухгалтерський облік і звітність, контролює збереження і цільове використання коштів, що знаходяться в її розпорядженні, враховує витрати і визначає фактичну собівартість продукції, нараховує заробітну плату працівникам, визначає і аналізує результати господарської діяльності, складає бухгалтерський баланс. лист.

Відділ кадрів несе досить важливу відповідальність за співробітників. Тобто перевіряє і контролює процес працевлаштування, тобто момент – відповідність їх освіти, досвіду роботи професії, якою вони займаються.

Виробництво та склад продукції компанії постійно оновлюється.

Такі зміни відбуваються через те, що використання нового обладнання на підприємстві потребує залучення більш кваліфікованих працівників.

У штат входять працівники складу, відділу купівлі-продажу, водії автобусів, а також менеджери магазинів і продавці-консультанти.

Цей магазин один з небагатьох, який працює з виробниками напряму, а не через посередників.

Це дозволяє клієнтам пропонувати повний асортимент оригінальної продукції за низькими цінами.



Рисунок 1.1 - Організаційна структура робітників ТОВ «Усі мотоблоки»

## 1.2 Аналіз об'єкта впровадження

Відділи територіально розділені та знаходяться в різних точках міста тому передача даних для підприємства являється одним з необхідніших елементів праці тому було створено веб сервера для передачу даних між ними.

В якості об'єкту виступе цех збирання та офіс продаж с .

Цех збору включатиме в себе 20 робочих місць, офіс закупівель 10 місць та в цьому ж приміщенні розташований магазин, офіс прийому товарів – 5 робочих місць та відвантаження – 2 місця.

Схема розміщення робочих місць всіх відділів промислово-виробничого персоналу показані на рисунках

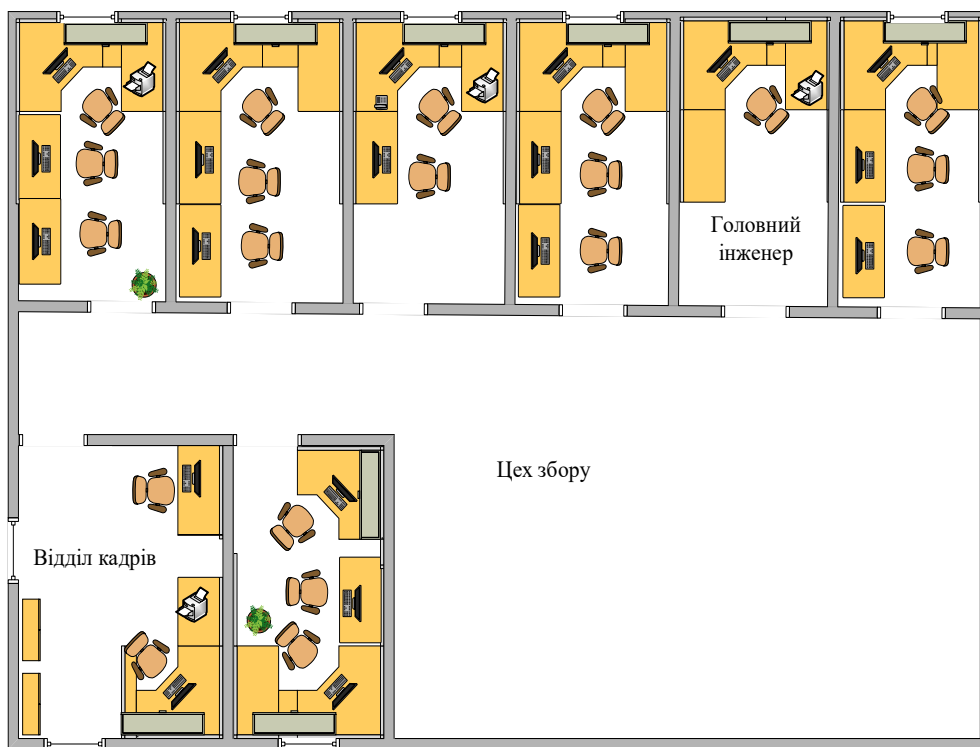


Рисунок 1.2 - План розміщення робочих місць цеху збору ТОВ «Усі

МОТОБЛОКИ»

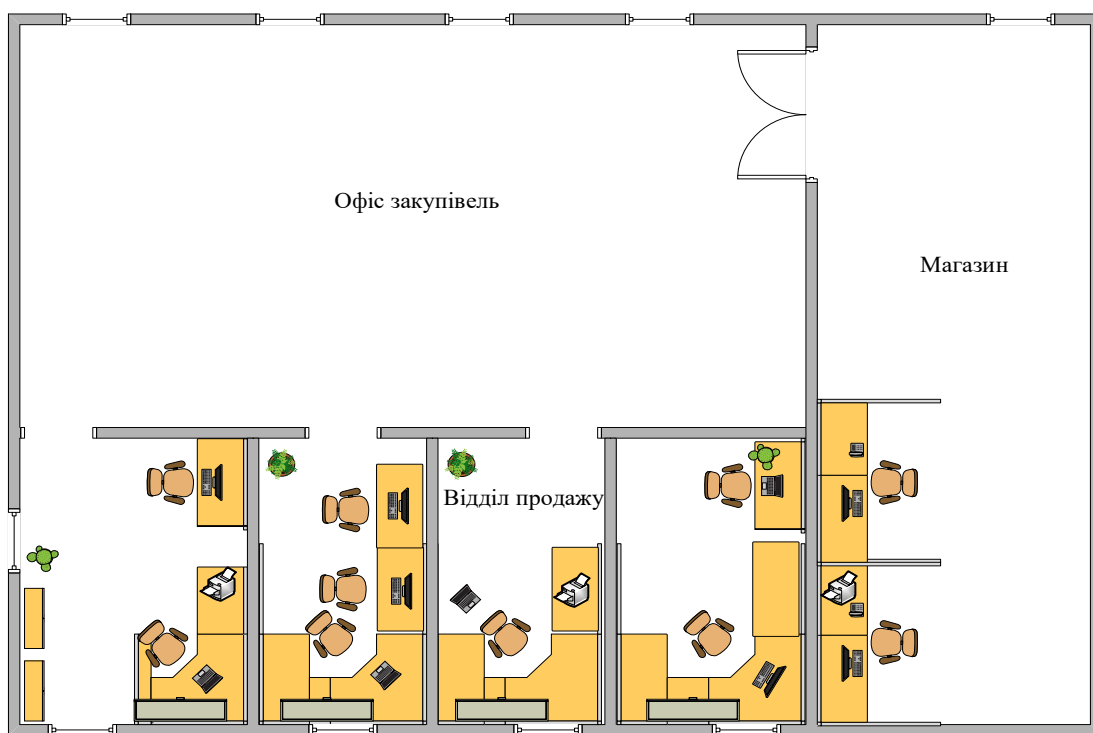


Рисунок 1.3 - План розміщення робочих місць офісу закупівель ТОВ  
«Усі мотоблоки»

### 1.3 Аналіз існуючих проблем

На даний момент облік товару ведеться через гугл таблиці, обробка даних займає чи не малий час.

Для кожного товару є своя ниша та індивідуальний опис і т.д.

Коли приходить товар його треба розмістити в таблиці занести його дані для подальшої обробки, що буде займати не мало часу.

На даний момент є декілька проблем:

- швидкість додання нового товару в БД;
- йде багато часу для додавання товару
- велика можливість заплутатись в даних та допустити помилку

В даний момент всі дії практично виконуються в ручну з значними витратами в часі. Тому ці задачі можливо автоматизувати.

#### **1.4 Способи обробки інформації і передачі інформації, принципи побудови досліджуваного об'єкта.**

Як було вище приведено компанія займається продажем сільгосп мото-технікою, раніше облік товару вівся в ручну дані заносилися в Гугл таблиці або в паперовий вид.

При виконанні щорічної інвентаризації складалася звіти які передавалися до відділу продаж та інтеграцій, де вказувались нові позиції які потрібно завозити, остачу не реалізованих позицій.

Тому треба вирішити наступні проблеми:

– інвентаризація та просто облік товару є складною роботою для комірника тому треба максимально модернізувати обладнання тому переносимо всі табличні та дані з паперових таблиць в базу даних;

– під час карантинних заходів деякі робітники не могли приходити до офісу працювати, тому треба організувати доступ до бази даних через крос-платформний веб-інтерфейс.

#### **1.5 Мета і завдання дослідження**

Метою представленої роботи є забезпечення ефективного функціонування вузлів системи бази даних для оперативної обробки даних менеджерами відділу продаж.

Завдання дослідження: забезпечити максимальну швидкодію серверної частини.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- обґрунтування критеріїв ефективності розподілу даних
- синтез системи підтримки дистанційного доступу до бази даних.
- модернізація бази даних за допомогою використання нових методів та технічних засобів.
- впровадження веб-додатку для покращення взаємодії з базою даних.

## 2 ТЕОРЕТИЧНИЙ РОЗДІЛ

### 2.1 Концепція бази даних

База даних (БД) — це сукупність даних, які взаємопов'язані та зберігаються разом.

Основні особливості бази даних:

- дані логічно пов'язані між собою і містять відповідну інформацію;
- що структура бази даних загалом відповідає конкретному набору даних, які вона містить;
- допускається мінімальний обсяг надлишкових даних, що сприяє їх оптимальному використанню в одній або кількох програмах;
- незалежність програмних даних;
- використовуються звичайні механізми пошуку та зміни даних;
- база даних, як правило, містить методи підтримки її цілісності та захисту від несанкціонованого доступу.

Системи керування базами даних — це програмні засоби, які дозволяють створювати, оновлювати та працювати з базами даних.

У світі існує багато різних систем керування базами даних. Багато з них насправді не є готовими до використання продуктами, а скоріше спеціалізованими мовами програмування, які дозволяють кожному, хто вивчає мову, створювати бажані структури та впроваджувати бажані елементи керування.

Ці мови включають Clipper, Paradox, FoxPro та інші.

Системи керування базами даних (СУБД) — це набір програмних засобів, необхідних для створення, використання та підтримки баз даних.

Система керування базами даних (СУБД) об'єднує інформацію з різних джерел у реляційну базу даних. Створені форми, програми та звіти дозволяють швидко та ефективно оновлювати дані, відповідати на запитання, шукати потрібні дані, аналізувати дані, друкувати звіти, діаграми та інструкції.

Організувати одну базу даних можна було лише шляхом створення спеціальних програмних продуктів – систем керування базами даних (СУБД).

Основне призначення СУБД — створити та підтримувати актуальну базу даних і зв'язати її з економічними програмами усунення несправностей (додатками користувача). База даних — це комп'ютерний термін, який використовується для позначення колекції інформації на певну тему або інформації, пов'язаної з корисним завданням.

Зберігаючи інформацію у формі бази даних, легко отримати доступ до необхідних фрагментів, здійснювати їх пошук і вилучення.

База даних може бути збережена на магнітному диску у вигляді одного файлу (БД MS Access, Informik тощо) або у вигляді файлової папки (БД Interbase, Paradok та ін.).

Системи керування базами даних (СУБД) виникли ще до персональних комп'ютерів.

Характеристики мейнфреймів, на яких він створювався в 70-х роках минулого століття, багато в чому визначили характеристики цих СУБД: вони дозволяли фахівцеві робити багато речей, від обробки транзакцій до передачі даних і програм в інші операційні системи. . інші системи та комп'ютери (платформи). .

Але ці СУБД (Oracle, INGRES тощо) не є нормою для персональних комп'ютерів, оскільки вони занадто вимогливі до характеристик використовуваного обладнання комп'ютера та якості користувача. У 1980-х роках було розроблено велику кількість СУБД спеціально для персональних комп'ютерів.

У нашій країні найпопулярнішими базами даних були FoxBASE, dBASE III plus, R: Base, Paradok, а в кінці 80-х став популярним пакет Clipper. Зверніть увагу, що FoxBASE, dBASE III plus і Clipper використовували однакові принципи організації інформації та були сумісні



на рівні файлів баз даних, тому іноді всі ці системи розглядали як модифікації dBASE III plus.

Ashton-Tate розробили програмну систему dBASE III plus на основі попередніх СУБД: dBASE II та dBASE III. У dBASE III plus основна увага була зосереджена на вдосконаленні інтерфейсу користувача (режим ASSIST), що значно спростило процес створення та модифікації баз даних, сортування та індексування баз даних.

З'явилася можливість створювати і використовувати складні структури баз даних безпосередньо з режиму ASSIST без перекладу прикладних програм на мову dBASE, що зробило цю СУБД доступною для широкого кола користувачів.

Це забезпечило широку популярність dBASE III plus, і до кінця 1980-х ця СУБД була фактичним стандартом для реляційних баз даних, незважаючи на деякі фундаментальні недоліки цієї системи.

Однак на початку 1990-х років ситуація змінилася. FoxBASE (FoxPro) і Paradox є найпопулярнішими системами баз даних для комп'ютерів. СУБД FoxPro була розроблена компанією Fox Software Inc. спеціально для створення додатків користувача.

Ця СУБД мала дуже потужні програмні засоби та полегшувала написання прикладних програм, хоча вона могла не мати зручного для користувача інтерфейсу dBASE III plus.

Borland International розробила СУБД Paradox, а також зосередилася на створенні додатків на основі повністю інтегрованої мови програмування PASCAL. Ця СУБД використовувала новий спосіб організації інформації на основі метафори таблиці, що дозволило легко реалізувати систему запитів, наприклад (Query ві Ekample), і з іншого боку, забезпечити дуже високу швидкість при пошуку інформації. Одним із недоліків СУБД FoxPro, dBASE III plus, Paradox була неможливість використовувати їх для створення файлів .EXE, які запускаються незалежно під DOS.

Таким чином, пакет Nantucket's Clipper, спочатку призначений для компіляції прикладних програм (принаймні тут), був розширений.

Clipper працював з файлами .DBF і був досить швидким.

Водночас це була відкрита система, яка дозволяла розширити можливості мови за допомогою додатків, написаних на інших мовах програмування: Assembler. У 1991 році, коли Ashton-Tate купила Borland International, була створена Fox Software Inc. - MicroSoft і компанія Nantucket - Computer Associates офіційно оголошують новий етап у розвитку СУБД для комп'ютера, хоча основні ідеї цього етапу були розроблені набагато раніше.

Ми можемо виділити три основні аспекти нової фази: розподілені бази даних, графічний інтерфейс користувача та архітектура клієнт-сервер. У 1993 році з'явилися перші СУБД цього нового покоління: Microsoft Access, dBase IV та ін. з'явився на ринку і отримав широке поширення серед користувачів ПК. В даний час СУБД Microsoft Access є стандартом де-факто для систем управління базами даних для персональних комп'ютерів.

Microsoft Access для Windows 95 — потужний інструмент адміністрування баз даних, який підтримує реляційну модель даних і дозволяє створювати складні програми на спеціальній мові Visual BASIC (VBA).

Microsoft Access забезпечує пошук і обробку всіх типів даних і підготовку звітних документів. Інтерфейс користувача досить простий і пропонує користувачеві зручні варіанти роботи з базою даних, тому освоєння пакету зазвичай не викликає проблем.

У зв'язку зі швидким розвитком Інтернету, який представляє собою величезну розподілену базу даних, зріс інтерес до СУБД типу Oracle. В даний час ця система управління базами даних встановлена на багатьох мережевих серверах. Система керування базами даних Oracle від Oracle є одним із лідерів ринку кросплатформних СУБД.

Він може працювати на більш ніж двох сотнях типів комп'ютерів, включаючи IBM PC і Apple Macintosh. Це програмне забезпечення СУБД містить одну з найповніших реалізацій мови структурованих запитів SQL, а також генератори для меню, звітів та інших відображень.

Крім того, програмне забезпечення дозволяє створювати понад 50 типів таблиць і графіків на основі інформації, що зберігається в СУБД. Oracle містить дуже надійну систему цілісності та узгодженості даних.

## **2.2 Аналіз методів і засобів доступу до інформації серверу.**

Основною задачею роботи сервера є її база, від складу якого залежить успіх роботи цеху збирання. Тому що саме сформованість БД визначає зміст, повноту та якість наповненості складу.

Чим краще сформована база, тим якісніше і краще магазин може задовольнити запит покупців.

Саме головне для нашої БД – визначати які товари відносяться до основної категорії, а які – ні, адже основна частина таблиці підлягає обліку та інвентаризації.

У разі вибуття товару з БД вказується причина вибування з нього: припинення випуску, товар закінчився на складі.

Головним завданням інвентаризації товару та звірка з БД є перевірка фактичної наявності товарів, та виявлення відхилень та з'ясування їх причин.

## **2.3 Система управління базами даних**

До популярних СУБД належать MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sibase, Interbase, Firebird і IBM Db2.

За моделлю даних, яка використовується в СУБД, вони поділяються на ієрархічні, мережеві, реляційні, об'єктно-реляційні та ін.

Прикладом СУБД є Microsoft Office Access. Він працює з об'єктами (таблицями, запитами, формами, звітами тощо), які зазвичай зберігаються в одному файлі з розширенням .accdb.

ACCESS : вперше продуктивна СУБД органічно поєднується з простотою використання для користувачів Microsoft Windows. Оскільки обидва продукти є продуктами Microsoft, вони ідеально працюють один з одним. Система Access працює під керуванням операційної системи Windows 2000 або Windows NT, тому під час роботи з нею користувач має доступ до всіх переваг системи Windows. Ви можете вирізати, копіювати та вставляти дані з будь-якої програми Windows у Access і навпаки; Ви можете створити проект форми в Access і перемістити його в конструктор форм.

Використовуючи Object Linking and Embedding (OLE) у компонентах Windows 2000 і Microsoft Office 2000 (Excel, Word, PowerPoint і Outlook), ви можете перетворити Access на справжній робочий простір бази даних.

За допомогою нових веб-розширень ви можете створювати форми, які безпосередньо взаємодіють із даними з Всесвітньої павутини, і переводити їх у презентації HTML, які працюють із такими продуктами, як Internet Explorer і Netscape Navigator. Зважаючи на це, Access — це не просто СУБД.

Як реляційна СУБД Access забезпечує доступ до всіх типів даних і дозволяє одночасне використання кількох таблиць бази даних. При цьому ми можемо значно спростити структуру даних і тим самим полегшити виконання завдань.

Таблицю Access можна пов'язати з даними, що зберігаються на комп'ютері чи сервері. Ви також можете використовувати таблиці, створені в Paradox або dBASE. Отримані результати можна швидко і легко зв'язати і об'єднати з даними з таблиць Excel. Працюючи в середовищі Microsoft Office 2000, користувач отримує Access, а також Word, Excel і PowerPoint, які повністю сумісні між собою.

MySQL - як і будь-яка інша СУБД, є серверною програмою, яка знаходиться в пам'яті комп'ютера і обслуговує порт TCP.

У випадку MySQL номер порту буде 3306. Клієнтська програма, будь то програма Perl CGI або програма C, підключається до СУБД через цей порт і надсилає їй рядки SQL.

Останній, у свою чергу, інтерпретує їх, виконує необхідні дії та відправляє результати запиту назад клієнту. Таким чином сервер бази даних спілкується з клієнтськими програмами.

Щоб запустити сервер MySQL, потрібно запустити файл `mysqld.exe`. Сервер працює як безвіконний фоновий процес. При цьому він залишається в пам'яті і обробляє запити від клієнтських додатків.

Щоб зупинити сервер, виконайте команду:

```
mysqladmin -u корінь вимкнення.
```

Якщо сервер вимкнено неправильно, під час наступного запуску до файлу `mysql.err` буде додано неправильний запис про завершення роботи.

Щоб зберегти всі дані в кеші MySQL, потрібне належне завершення роботи сервера.

MySQL має вдосконалену систему доступу до бази даних.

Користувач бази даних може отримати доступ до всієї бази даних, окремих таблиць і окремих стовпців таблиці. Існує різниця в діях, які користувач може виконувати над записами.

Для організації такої складної (на перший погляд) структури доступу використовується кілька таблиць в окремій базі даних.

Політики доступу створюються на основі значень у цих таблицях.

Основні команди для управління MySQL

Клієнтська програма MySQL.

Запуск: `miskl [SETTINGS] data base`

ОПИС: Клієнтська сторона MySQL називається MySQL. Надає інтерфейс командного рядка для СУБД MySQL і можливість неінтерактивної пакетної обробки.

`mysqlaccess`

Перевірка прав доступу користувача.

СИНТАКСИС:

`misklaccess [host] [user] [db] OPTIONS`

ОПИС: Сценарій `misklaccess` використовується для перерахування привілеїв користувача для певної бази даних. Це надає інформацію, яка може бути корисною для діагностики проблем із доступом користувачів до певної бази даних.

`miskldump`

Викидання вмісту бази даних.

СИНТАКСИС:

`miskldump [SETTINGA] [data base [table [field]]]`

ОПИС:

Програма `miskldump` використовується для створення дампу вмісту бази даних MySQL. Записує оператори SQL у стандартний вихід. Ці оператори SQL можна перепризначити файлу. Можна створити резервну копію бази даних MySQL за допомогою `miskldump`, але ви повинні переконатися, що база даних наразі не виконує жодних інших операцій.

`mysqlshow`

Відображення інформації про сервер, базу даних або таблицю.

СИНТАКСИС:

```
misklshov [OPTIONS] [data base [table [field]]]
```

ОПИС:

Програм мисклсхов се може користити да покаже са којим базама података ради МиСКЛ, које табеле садржи дата база података и која поља су присутна у табели те базе података.

## **2.4 Можливості СУБД**

– дозволяє створювати бази даних (створені за допомогою мови визначення даних (DDL)).

– дозволяє додавати, оновлювати, видаляти та читати інформацію з бази даних (за допомогою мови обробки даних DML, яку часто називають мовою запитів)

– можете надати контрольований доступ до бази даних:

Системи безпеки, що запобігають несанкціонованому доступу до бази даних;

Паралельні системи керування роботою прикладних програм, що контролюють процеси доступу до загальної бази даних;

Відновлення системи – дозволяє відновити базу даних до попереднього неконкуруючого стану, на який вплинула апаратна чи програмна помилка.

## **2.5 Визначення файлу даних: запис, структура запису, поле запису та його реквізити (ім'я, тип, розмірність)**

Файли - це області пам'яті на зовнішньому носії (зазвичай магнітному диску), призначені для:

– збереження даних;

– довгострокове зберігання даних (зберігаються при вимкненні машини).

– цим файли відрізняються від звичайних рядків

– може змінювати свій розмір;

– доступ до елементів цих масивів здійснюється не через операцію індексування, а через системні виклики та спеціальні функції;

– доступ до елементів файлу відбувається під час так званого «завдання читання/запису», яке виконується автоматично під час читання/запису, тобто файл сканується послідовно. Однак існують функції для довільної зміни цієї позиції.

Розглянемо структуру файлу даних (тип .DBF).

Файл бази даних складається із записів заголовків і записів даних. Запис заголовка визначає структуру бази даних і містить всю іншу інформацію, що стосується бази даних.

Він починається з нульової позиції у файлі.

Записи даних (\*) йдуть після заголовка (міри впорядковані послідовно) і містять фактичний вміст поля.

Довжина запису (в байтах) визначається додаванням заданої довжини всіх полів. Числа в цьому файлі розташовані в зворотному порядку.

## **2.6 Висновки по розділу**

В цьому розділі було розібрані методи та технології з допомогою яких буде виконуватися розробка додатку, розглянуті як плюси так і мінуси роботи з ним



## **3 СИНТЕЗ СИСТЕМИ**

### **3.1 Аналіз вимог до БД**

База даних має складатися з декількох таблиць.

В першу чергу таблицю товарів, таблицю збиральників.

Сховище даних повинно забезпечувати збереження інформації на протязі 1 року.

Підтримувати можливість відзеркалення накопичувача для відказостійкості.

### **3.2 Визначення напрямків рішення поставлених завдань**

Одною з основних завдань це було створення Веб-інтерфейсу для крос-платформного користування (користуванням на різних пристроях) для роботи з базою даних.

Для того щоб це реалізувати треба було обрано декілька варіантів як безпечно та надійно буде розвернути сервер.

Декілька рокі тому вже намагались реалізувати таку систему але вона була не надійною та незрозумілою для користування.

В цей раз ми будемо використовувати інші засоби.

На цей раз буде використовуватись MySQL, хоч вона і не сучасна але вона має добрий функціональний запас таке як повний віддалений доступ до бази даних.

Також в даному проєкті використовується система замовлення товарів, за допомогою чого збиральник буде знати через скільки треба буде відвантажити замовлення.

Система повинна мати можливість дистанційно, через мережу Інтернет, виконувати наступні функції:

- підтримка доступу до всіх вузлів;
- віддалене налаштування та обслуговування;

– швидке управління системою та швидкий доступ до відповідних вузлів;

– аналіз продуктивності серверного обладнання та відео обладнання.

Для вирішення поставлених задач потрібно організувати таку систему, яка зможе безперебійно обробляти сотні запитів за хвилини та тисячі запитів що години.

Вище було сказано що при створенні БД буде використовуватися MySQL.

На сьогоднішній день СУБД MySQL є однією з найвідоміших, надійних і найшвидших у всьому існуючому сімействі СУБД.

Однією з причин є правила розповсюдження: оплата не потрібна, і вона поширюється з вихідними текстами, інша причина: MySQL є відносно швидкою СУБД.

PostgreSQL, наприклад, також поширюється під \*GNU GPL, але не так широко використовується. Однією з причин є затримка, яка спостерігається.

Дві головні причини популярності MySQL – ціна та продуктивність. MySQL написаний для багатьох операційних систем. До них належать FreeBSD, OpenBSD, MacOS, OS/2, SunOS, Win9x/00/NT і Linux. Сьогодні MySQL особливо популярний на платформах Linux і Windows. Крім того, останній зустрічається набагато рідше.

Принцип роботи СУБД MySQL подібний до принципу роботи будь-якої СУБД, яка використовує SQL (Structured Query Language) як мову команд для створення/скидання баз даних, таблиць, заповнення таблиць даними та виконання пошуку даних. .

Метою цієї статті є опис конфігурації, адміністрування та програмування СУБД MySQL і створення програми, яка демонструє можливості MySQL.

Веб-інтерфейс буде створений за допомогою JavaScript і PHP, надалі при використанні програми потрібно буде зберігати дані.

MySQL не підходить, але при використанні можливостей PHP файли будуть зберігатися не в базі даних, а на диску в масиві Raid, що гарантуватиме цілісність даних у разі збою частини системи.

javascript

Це мова програмування, яка дозволяє створювати динамічно оновлюваний вміст на сторінці.

Наприклад, це може бути управління мультимедіа, анімація зображення тощо.

Знання JavaScript вважається ключовим навиком у контексті веб-розробки. Програми, написані цією мовою, називаються скриптами.

Ви можете вставити їх у свій HTML, і вони виконуватимуться автоматично під час завантаження веб-сторінки.

Скрипти розповсюджуються та виконуються у вигляді звичайного тексту. Для його виконання не потрібна спеціальна підготовка чи переклад.

JavaScript є «безпечною» мовою програмування. Він не дозволяє низькорівневий доступ до процесора чи пам'яті, оскільки спочатку був розроблений для браузерів, яким це не потрібно.

У браузері JavaScript є все, щоб маніпулювати веб-сторінками, взаємодіяти з користувачем і веб-сервером.

У браузері JavaScript може, наприклад:

- додавати новий HTML-код на сторінку, змінювати наявний вміст, змінювати стилі;
- реагування на дії користувача, клацання мишею, переміщення вказівника, натискання клавіш;
- надсилати мережеві запити на віддалені сервери, завантажувати та передавати файли;
- отримувати та встановлювати файли cookie, ставити запитання відвідувачу, відображати повідомлення;
- резервне копіювання даних для клієнта («локальне сховище»).

Перевага JS полягає в тому, що це єдина технологія браузера, яка виконує три основні функції:

- повна інтеграція з HTML/CSS;
- що прості речі робляться просто;
- підтримується всіма основними браузерами та ввімкнено за замовчуванням.

Ще цікавішим є функціонал, створений на основі основної мови JavaScript.

Інтерфейси прикладного програмування (API) надають вам додаткові функції, які ви можете використовувати у своєму коді JavaScript.

API — це готові до використання набори блоків коду, які дозволяють розробнику впроваджувати програми, які інакше було б важко або неможливо реалізувати.

Перевага JS полягає в тому, що це єдина технологія браузера, яка виконує три основні функції:

- повна інтеграція з HTML/CSS;
- що прості речі робляться просто;
- підтримується всіма основними браузерами та ввімкнено за замовчуванням.

## PHP

Це гіпертекстовий препроцесор і широко використовувана мова сценаріїв, розроблена для потреб веб-розробки для створення динамічних веб-сторінок.

З цією ж метою PHP-код буде вбудовано у вихідний HTML-документ. Веб-сервер разом із модулем обробки, який генерує веб-сторінку, інтерпретує цей документ. Знання того, як запускати PHP, може бути корисним для людей, які використовують мову сценаріїв.

### З переваг PHP

– відкрите джерело

PHP — це мова програмування з відкритим вихідним кодом, яка допомагає розробникам швидко встановити її, щоб швидко почати використовувати.

– незалежність

Мова програмування PHP підтримується всіма операційними системами.

– простота

Мова програмування легка та проста у вивченні та використанні. Команди легко зрозуміти та вивчити.

– база даних

Без проблем підключається до бази даних і забезпечує надійне підключення до баз даних.

– стабільність

Мова програмування PHP порівняно з іншими мовами програмування, оскільки PHP існує вже давно.

– надійність

Для захисту від підробки даних і впровадження SQL для розробки веб-додатків використовуються фреймворки PHP.

Для створення та формування сторінок веб-інтерфейсу використовуватимуться HTML та CSS, веб-сервер буде розроблено на базі Apache HTTP Server 2.4. Для розробки веб-сервера буде використовуватися мова програмування PHP, яка добре працює з базами даних MySQL.

### 3.3 Розробка схеми функціональної структури

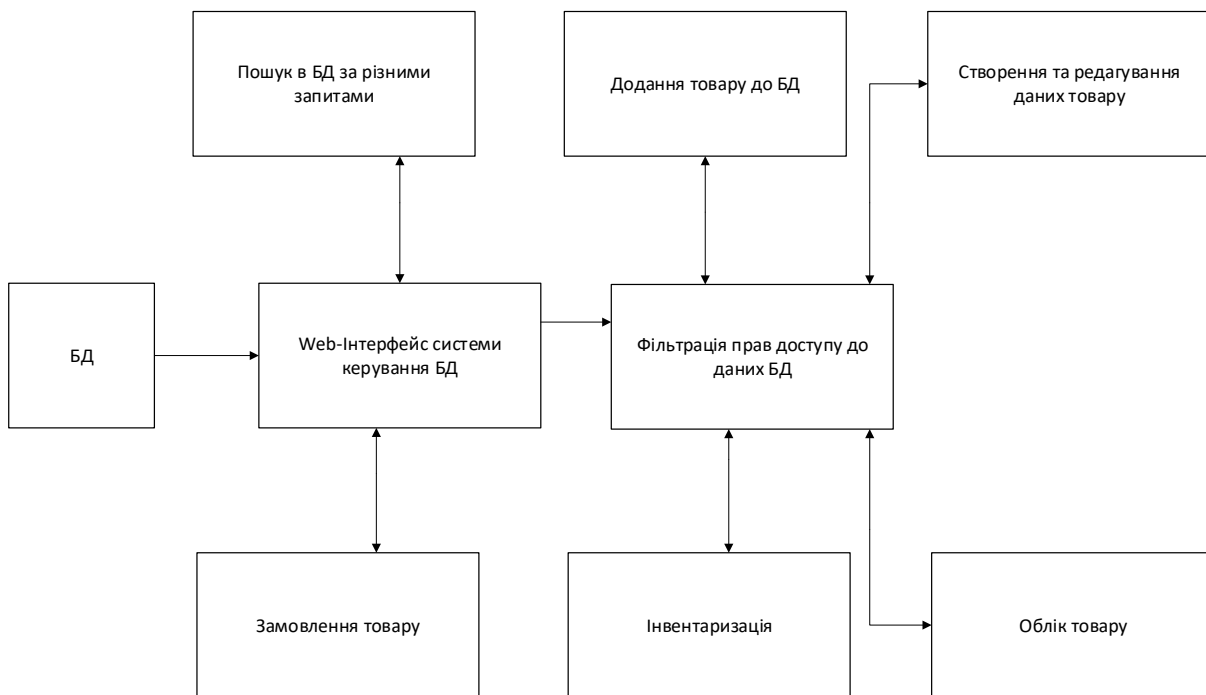


Рисунок 3.4 - Функціональна структура

Система реалізована на сервері. Керування сервером буде використовуватися віддалено на базі веб серверу.

Сервер було встановлено в цеху збирання та був підключений до мережі офісу.

Для зручності доступу користувачам до Web-серверу необхідно забезпечити DNS ім'я.

Приклад моделі мережі.

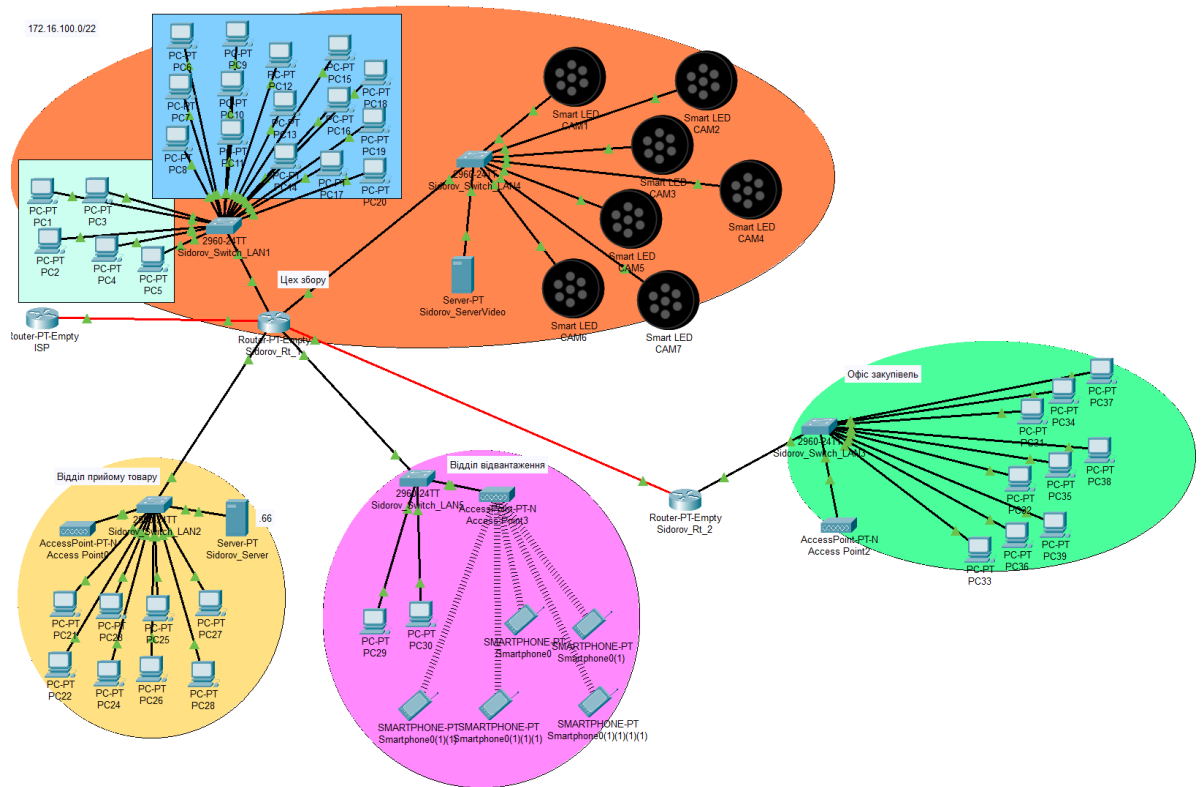


Рисунок – Модель комп’ютерної моделі ТОВ «Усі мотоблоки»

### 3.1.2 Вибір елементів бази системи

При виборі серверного обладнання треба урахувати наступні критерії:

- підтримка Raid
- модуль TPM 2.0

Також система повинна мати достатню потужність щоб могла опрацьовувати велику кількість запитів відділу продаж.

Декілька років тому для зручності роботи з товарами було встановлено сервер. Для успішності та меншої затратності буде використовуватися наявний сервер.

Таблиця 3.1 – Таблиця конфігурація сервера

Комплектуючі	Характеристики
Процесор	Чотири ядерний Intel Xeon E-2224G (3.5 — 4.7 ГГц)
Обсяг ОЗУ	16Гб
Контролери SAS/SATA	Intel Rapid Storage Technology Enterprise (RSTe)
Рівні RAID	0/1/5/10
Жорсткий диск	2 x 1 ТБ
Роз'єми	Передня панель: 1 x USB 3.1 Type-C 1 x USB 3.1 2 x USB 2.0 Задня панель: 1 x PS/2 порт для клавіатури 1 x PS/2 порт для миші 2 x USB 2.0 4 x USB 3.1 2 x DisplayPort 1 x послідовний порт 1 x аудіороз'єм Слоти розширення: 1 x PCI-E 3.0 x16 2 x PCI-E 3.0 x4 1 x PCI-E

Потужності такої конфігурації вистачить для опрацювання запитів всього відділу продаж та цеху збирання.

### 3.1.3 Налаштування HTTP та DNS

HTTP (протокол передачі гіпертексту) — це набір правил, яких повинен дотримуватися сервер, коли йдеться про передачу файлів (зображень, відео, аудіо та інших форматів файлів) через Всесвітню павутину (WWW). Коли користувач відкриває браузер, він уже використовує HTTP. По суті, це прикладний протокол, який працює поверх набору протоколів TCP/IP.

DNS-сервер — це сервер системи доменних імен (DNS), який



відповідає за зіставлення доменних імен Інтернету з IP-адресами комп'ютерів, на яких фізично розташовані ці домени. DNS-сервери дозволяють користувачам вводити звичайні адреси сайтів у браузер і позбавляють від необхідності запам'ятовувати IP-адреси.

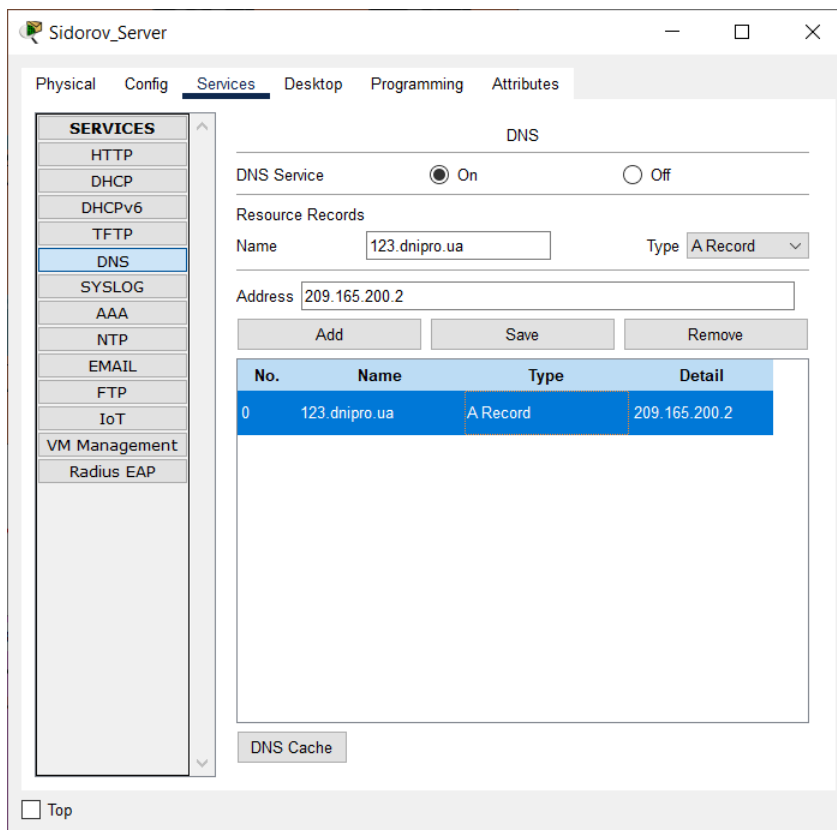


Рисунок 3.5 - Налаштування DNS-серверу

Після цього буде налаштовано NAT.

NAT (Network Address Translation) — це технологія для перетворення приватних IP-адрес на зовнішні адреси в IPv4.

Трансляцію адрес за допомогою методу NAT може виконувати практично будь-який мережевий пристрій - маршрутизатор, сервер доступу, брандмауер.

Трансляцію адрес за допомогою методу NAT може виконувати практично будь-який мережевий пристрій - маршрутизатор, сервер доступу, брандмауер.

### 3.1.4 Розробка структури бази

Таблиця 3.2 – Дані про товар

<b>Дані, що треба зберігати в таблиці</b>	<b>Тип даних</b>	<b>Обов'язковість заповнення</b>
Ідентифікатор товару	Число	Так
SKU (Складська одиниця)	Число	Так
Назва товару	Текст	Так
Бренд	Текст	Так
Категорія товару	Текст	Так
Опис товару	Текст	Ні
Рік випуску	Число	Ні
Кубатура товару	Число	Так
Дата прийняття	Число	Так
Посилання на товар	Текст	Ні
Помітка (пошкодження тощо)	Текст	Ні

### 3.1.5 Розробка мови ситуаційного керування пошуком

Для розробки ситуаційної мови керування необхідно встановити поняття, імена та відношення, що будуть описувати пошук по базі даних.

Таблиця 3.1 – Поняття що описують

<b>Позначення</b>	<b>Опис</b>
U	Пошук за полем “UID”
S	Пошук за полем “SKU”
T	Пошук за полем “Title”
B	Пошук за полем “Brand”
P	Пошук за полем “Price”
L	Доступність товару до продажу

Таблиця 3.2 – Таблиця наявності товару

<b>Позначення</b>	<b>Опис</b>
Y1	Товар доступний
Y2	Товару нема в наявності

В таблиці поняттю U можуть присвоюватися будь-які значення яке буде складатися з 12 символів що вело за собою велику кількість проблем з обліком товаром за часту оператор переплутував значення та надавав невірні дані в таблицю.

Також таку проблему можна побачити в полях S, B, T.

Тому було вирішено реалізувати систему гарячих клавiш для швидкого користування бази даних.

Для того щоб користувач швидко та безпомилково знайшов той чи інший товар, йому потрібно всього натиснути кнопку яка відповідає пошуку.



Рисунок 3.6 - Вигляд пошуку по базі

З такою системою позначень всім користувачам БД буде легко нею користуватися та така модель користування буде мати високу достатність так як не всі поняття будуть використовуватися.

Для отримання списку товарів достатньо задати один з параметрів пошуку за понятійною базою U, S, T, B, L при цьому також необхідна перевірка наявності товару на складі відповідно понятійної бази Y.

### **3.4 Висновки по розділу**

В даному розділу було організовано структуру бази даних, розроблені поняття ситуаційної мови пошуку для прискорення та редагування та роботи даних.

## **4 . Розділ розроблення програмного забезпечення**

### **4.1 Призначення та область застосування**

Програмне забезпечення призначено для керуванням базою даних складу.

Призначення програмного забезпечення – виконувати всі поставленні задачі які буди описані вище, а саме:

- запис даних про товар в БД
- редагування даних про товар в БД
- створення та редагування даних про товар
- облік товару
- щомісячна інвентаризація
- пошук в БД за різними запитами

### **4.2 Обґрунтування технічних характеристик програми**

Програма представляє з себе кросплатформенний веб-інтерфейс який написаний на мові програмування PHP: 8.1.0 та деякі функції вимальовки були виконані на мові JavaScript, HTML.

Код розробленої таблиці знаходиться в додатку А.

Веб-інтерфейс призначений для прискорення занесення та редагування даних в базі, він спроектований так щоб користувачу було з самого початку як взаємодіяти з інтерфейсом, для швидкого пошуку зроблені гарячі клавіші для швидкого редагування даних.

На головну сторінку винесені:

- пошук по базі даних
- статистичні панелі необхідні для роботи (огляд замовлень, відвідувачів, останнє замовлення і т.п.)

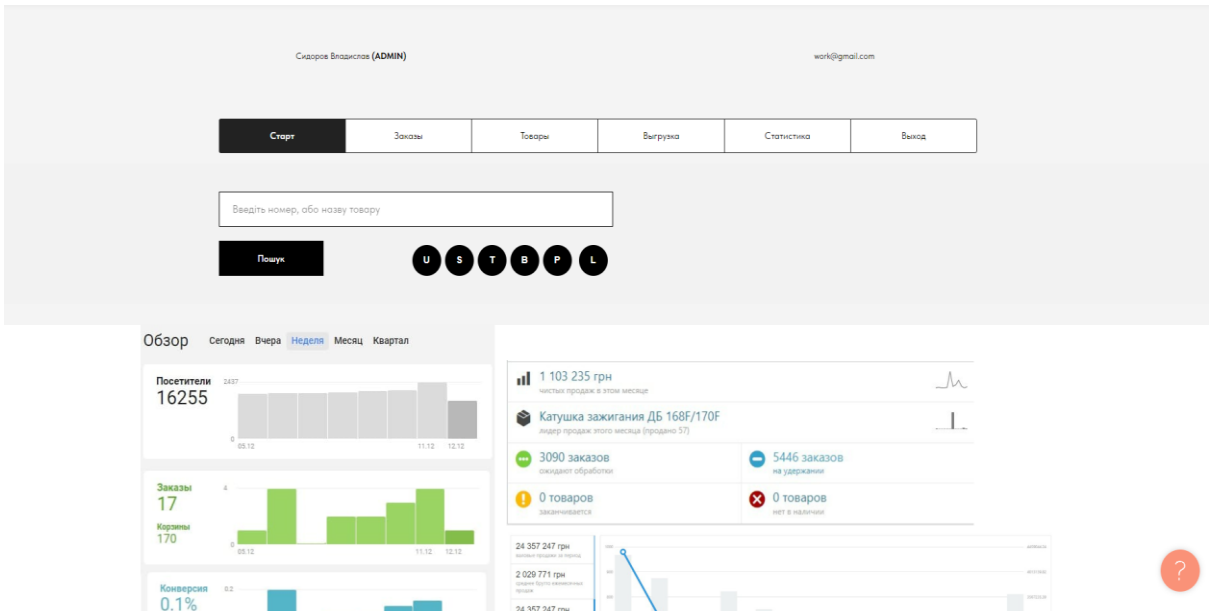


Рисунок 4.1 - Головна сторінка сайту

Також було розроблена сторінка для системного адміністратора для керування та аналізом швидкості передачі даних.

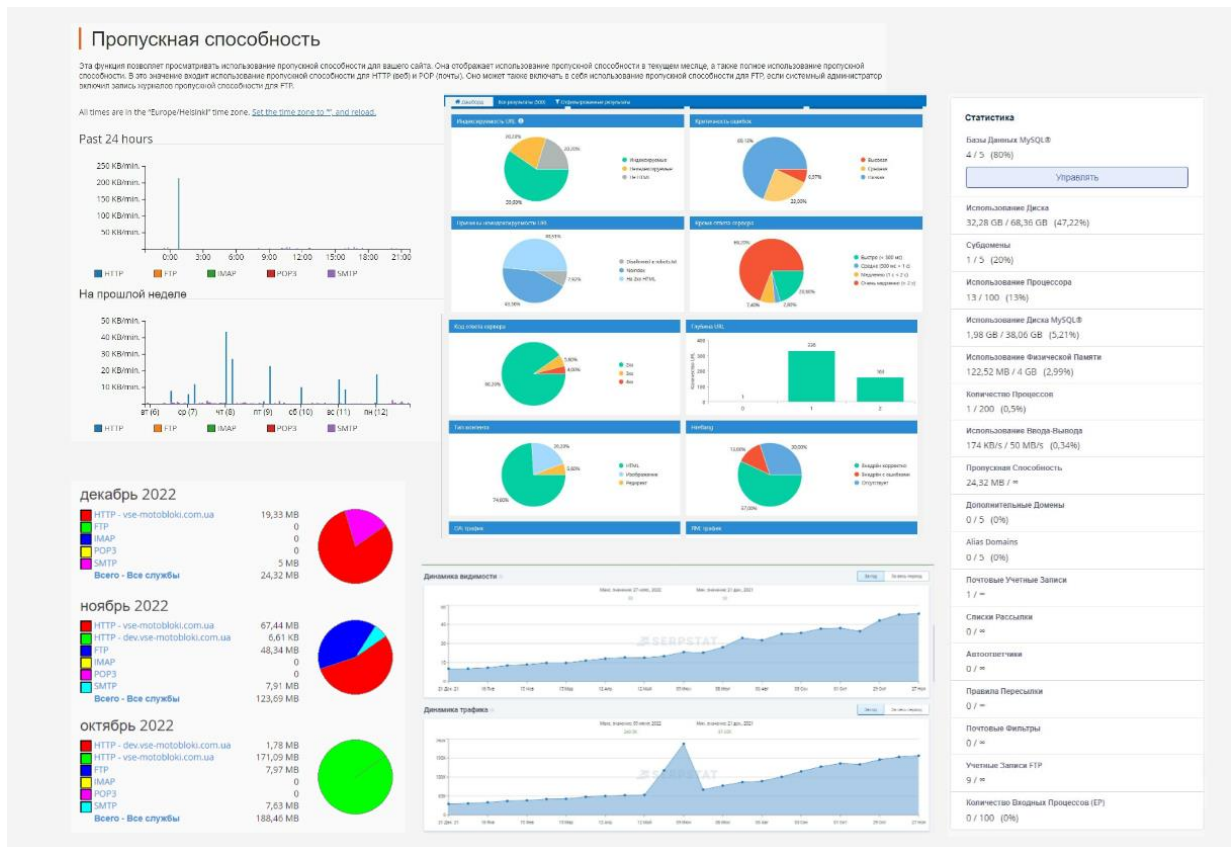


Рисунок 4.2 - Панель адміністратора

### **4.3 Опис і обґрунтування вибору складу технічних і програмних засобів.**

При розробці веб-інтерфейсу використовувались наступні програмні засоби:

Visual Studio Code

Visual Studio Code — це редактор вихідного коду, який сумісний з багатьма мовами програмування, включаючи C#, Java, JavaScript, Go, Node.js, Python, C++, C.

Він базується на структурі Electron, яка використовується для розробки веб-додатків Node.js, які працюють на механізмі компонування Blink.

Visual Studio Code використовує той самий компонент редактора (під кодовою назвою «Моносо»), який використовується в Azure DevOps (раніше називався Visual Studio Online і Visual Studio Team Services).

Замість системи проектів вона дозволяє користувачам відкривати один або кілька каталогів, які потім можна зберегти в робочих областях для подальшого використання.

Visual Studio Code підтримує багато мов програмування та набір функцій, який відрізняється для кожної мови. Небажані файли та папки можна виключити з проекту за допомогою налаштувань.

Код Visual Studio можна розширити за допомогою розширень, доступних через центральне сховище. Це включає доповнення до редактора та підтримку мови. Однією із функцій є можливість створювати розширення, які додають підтримку нових мов, тем, налагоджувачів, і т.д.

Керування вихідним кодом є вбудованою функцією Visual Studio Code. Він має спеціальну вкладку в рядок меню, де користувачі можуть отримати доступ до налаштувань керування версіями та переглянути зміни, внесені до поточного проекту.

Щоб використовувати цю функцію, Visual Studio Code має бути пов'язано з будь-якою підтримуваною системою керування версіями (Git,

Apache Subversion, Perforce тощо).

Це дозволяє користувачам створювати репозиторії та робити запити push і pull безпосередньо з Visual Studio Code.

Visual Studio Code включає кілька розширень для FTP, що дозволяє використовувати програмне забезпечення як безкоштовну альтернативу для веб-розробки. Код можна синхронізувати між редактором і сервером без завантаження додаткового програмного забезпечення.

Visual Studio Code дозволяє користувачам налаштувати кодову сторінку, на якій буде збережено активний документ, символ нового рядка та мову програмування активного документа. Це дозволяє використовувати його на будь-якій платформі, будь-де та для будь-якої заданої мови програмування.

Visual Studio Code збирає дані про використання та надсилає їх до Microsoft, хоча це можна вимкнути. Після відкриття програми код телеметрії стає доступним для громадськості, яка може бачити, що саме записується.

VSCodium надає двійкові версії коду VS без брендингу/телеметрії/ліцензій MS. VSCodium — це безкоштовний ліцензований двійковий дистрибутив редактора Microsoft VS Code без власного непрозорого коду Microsoft і телеметрії. Ці двійкові файли ліцензовані згідно з ліцензією MIT.

З коробки Visual Studio Code містить базову підтримку найпоширеніших мов програмування. Ця базова підтримка включає підсвічування синтаксису, зіставлення дужок, згортання коду та спеціальні фрагменти. Visual Studio Code також включає IntelliSense для JavaScript, TypeScript, JSON, CSS і HTML і підтримує налагодження Node.js. Безкоштовні розширення на VS Code Marketplace можуть забезпечити підтримку додаткових мов.

## PHP

PHP (рекурсивна аббревіатура PHP: Hypertext Preprocessor) — це широко використовувана мова сценаріїв загального призначення з відкритим кодом, яка особливо підходить для веб-розробки та може бути вбудована в HTML.



Замість ряду команд для генерування HTML (як це видно в С або Perl), сторінки PHP містять HTML із вбудованим кодом, який робить «щось» (у цьому випадку виводить «Привіт, я сценарій PHP!»). PHP-код міститься в спеціальних початкових і кінцевих директивах `<?php і ?>`, які дозволяють входити в режим PHP і виходити з нього.

Що відрізняє PHP від чогось на кшталт клієнтського JavaScript, це те, що код виконується на сервері та генерує HTML, який потім надсилається клієнту. Клієнт отримував би результати виконання цього сценарію, але не знав би базовий код. Ви навіть можете налаштувати свій веб-сервер на обробку всіх ваших файлів HTML за допомогою PHP, і тоді користувачі не дізнаються, що у вас в рукаві.

Найкраще у використанні PHP те, що він надзвичайно простий для початківців, але пропонує багато розширених функцій для професійних розробників. Не бійтеся читати довгий список функцій PHP. Ви можете миттєво розпочати роботу та почати писати прості сценарії всього за кілька годин.

Хоча розробка PHP зосереджена на написанні сценаріїв на стороні сервера, ви можете зробити з ним набагато більше. Читайте далі та дивіться більше на сторінці [Що може PHP?](#) або перейдіть відразу до вступного посібника, якщо вас цікавить лише веб-програмування.

## СУБД MySQL

На сьогоднішній день СУБД MySQL є однією з найвідоміших, надійних і найшвидших у всьому існуючому сімействі СУБД.

Однією з причин є правила розповсюдження: оплата не потрібна, і вона поширюється з вихідними текстами, інша причина: MySQL є відносно швидкою СУБД.

## HTML

HTML або HyperText Markup Language — це мова розмітки, яка використовується для створення зовнішньої частини веб-сайту. Структура веб-сторінки записується та відображається веб-браузерами на веб-сайтах,

які ми бачимо в Інтернеті.

HTML, зазвичай у поєднанні з CSS або каскадними таблицями стилів, дозволяє веб-сайтам виглядати так, як ми хочемо, щоб вони виглядали в Інтернеті.

HTML записується у вигляді блоків елементів і тегів. Ці теги показують структуру сторінки та вказують на тип тексту, який бачить глядач. Вони можуть означати що завгодно, незалежно від того, чи хочете ви, щоб текст відображався як заголовок, чи зображення на сторінці.

## JS

JavaScript — це мова програмування, яка дозволяє створювати динамічно оновлюваний вміст на сторінці.

Наприклад, це може бути управління мультимедіа, анімація зображень і так далі.

Знання JavaScript вважається ключовим навиком у контексті веб-розробки. Програми, написані цією мовою, називаються скриптами.

Вони можуть бути вбудовані в HTML і виконуватися автоматично під час завантаження веб-сторінки. Скрипти розповсюджуються та виконуються як звичайний текст. Для їх запуску не потрібна спеціальна підготовка чи компіляція.

JavaScript є «безпечною» мовою програмування. Він не забезпечує низькорівневий доступ до пам'яті чи процесора, оскільки спочатку був створений для браузерів, яким він не потрібен. Браузер JavaScript має все, щоб маніпулювати веб-сторінками, взаємодіяти з користувачем і веб-сервером.

## HTML

HTML використовує теги для відображення інформації різних типів, такої як: текст, мультимедійний вміст тощо.

Основною особливістю цієї мови є програмування з використанням

тегів, що складаються з імен елементів, оточених символами «<» і «>» для закриваючого тегу.

Цей тег включатиме спеціальні елементи, такі як <head>, <title>, <body>, <article>, <section>, <p>, <img>, <audio>, <canvas>.

Де основна частина нашого елемента:

- Відкриття тега. Він складається з імені елемента, укладеного у відкриваючі та закриваючі кутові дужки.

- Кінцевий тег. Це те саме, що початковий тег, за винятком того, що він містить скісну риску.

- Контент. Вміст елемента.

- Елемент. Відкриття, закриття та вміст разом складають елемент.

Елементи також можуть мати власні атрибути, які мають додатковий елемент для розмови або у фактичному вмісті.

## CSS

CSS (каскадні таблиці стилів) – використовується для стилізації та компонування веб-сторінок – наприклад, для налаштування зовнішнього вигляду шрифтів, розміру, кольору тощо.

CSS — це мова програмування на основі правил.

Правила відкриваються за допомогою селекторів.

Селектор вибирає HTML - елемент, який ми будемо стилізувати.

У цьому випадку ми будемо використовувати заголовки першого рівня.

Тоді ми маємо набір фігурних дужок {}. Всередині них буде одна або кілька декларацій, які мають форму пари: властивість і її значення.

## 4.4 Структура БД

Для встановлення зав'язків між таблицями використовуватись наступні позначення:

– UK – (Unique Key) – Основний ключ який матиме тільки унікальні значення.

– EK – (External Key) – зовнішній ключ як буде мати набір атрибутів потенціальним ключем для іншого відношення.

В таблицях приведено опис таблиць БД.

Таблиця 4.1 – Таблиця опису товару в Pit-Мор (товари)

Призначення	Назва поля	Тип	Довжина	Ключ
Ідентифікатор товару	UID	INT	12	UK
SKU (Складська одиниця)	SKU	VARCHAR	7	UK
Бренд	Brand	VARCHAR	30	
Категорія товару	Category	VARCHAR	22	
Опис товару	Desc	VARCHAR	500	
Рік випуску	Yer	YEAR	4	
Кубатура товару	Char:Cub	INT	3	
Кількість товару	Count	INT	400	
Ім'я збиральника	Name	VARCHAR	11	
Дата отримання товару	DataInsert	DATE		
Посилання на товар	URL	VARCHAR	450	
Помітка (пошкодження тощо)	Descp	VARCHAR		

Таблиця опису таблиці Els (опис)

<b>Призначення поля</b>	<b>Назва поля</b>	<b>Тип</b>	<b>Довжина</b>	<b>Ключ</b>
Ідентифікатор	Id-uid	INT	12	UK
SKU	SK	VARCHAR	6	UK
Статут	ST	VARCHAR	9	EK
Кількість	Cout	INT	1	
Дата збирання	Conf	DATE		
Стара ціна	PriceOld	INT		
Категорія	Category	VARCHAR	22	
Опис	Descript	VARCHAR	500	
Фото	Photo	VARCHAR	220	

Таблиця 4.2 – Данні які повинні зберігатися в таблиці

<b>Дані, що треба зберігати в таблиці</b>	<b>Тип даних</b>	<b>Обов'язковість заповнення</b>
1	2	3
Ідентифікатор товару	Число	Так

Продовження таблиці № 4.2

SKU (Складська одиниця)	Число	Так
Назва товару	Текст	Так
Бренд	Текст	Так
Категорія товару	Текст	Так
Опис товару	Текст	Ні
Рік випуску	Число	Ні
Кубатура товару	Число	Так
Кількість товару	Число	Так
Ім'я збиральника	Текст	Так
Дата отримання товару	Число	Так
Посилання на товар	Текст	Ні
Помітка (пошкодження тощо)	Текст	Ні

На основі цих даних було прийнято рішення будувати базу даних.

Для реалізації встановлюємо та налаштовуємо Apache 2.4.51 з PHP,  
MySQL.

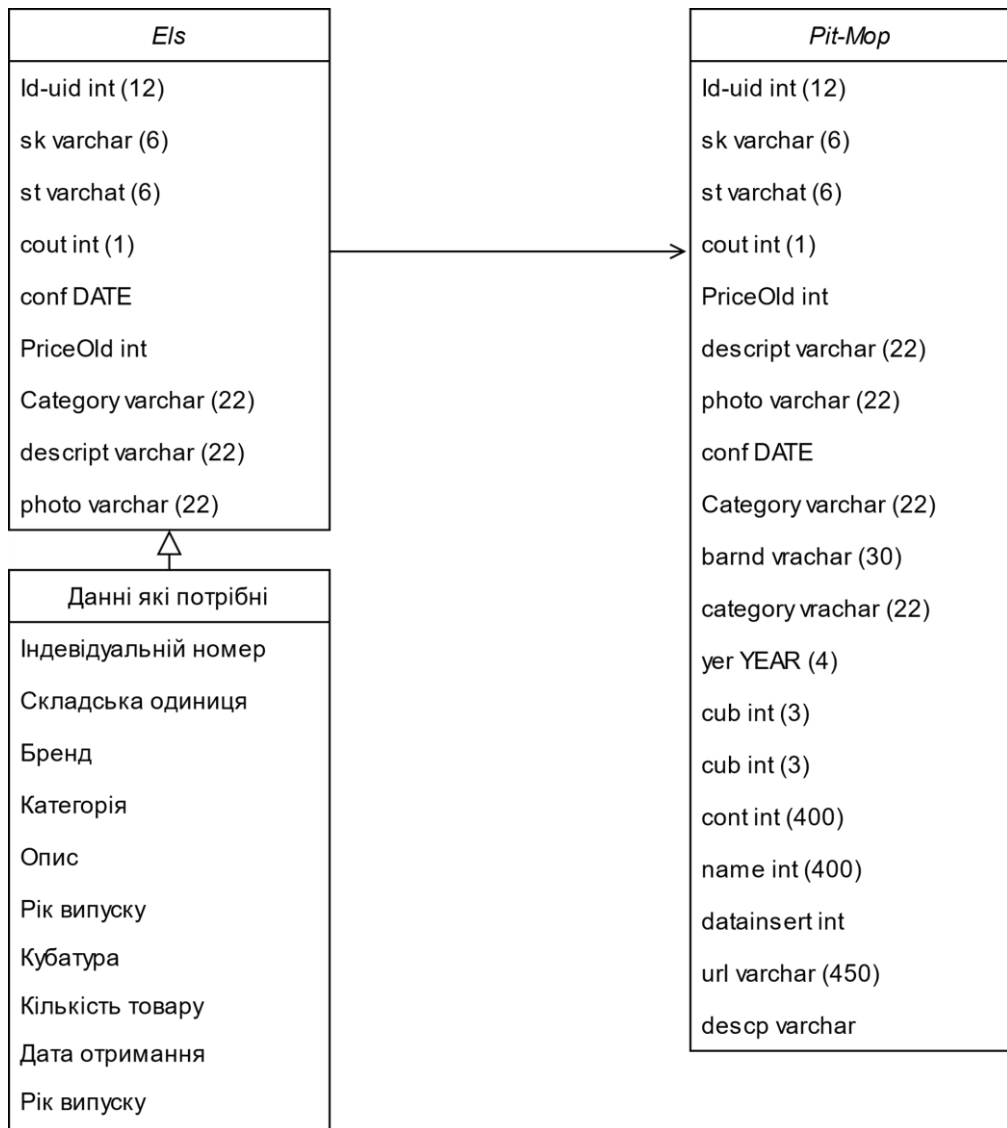


Рисунок 4.3 - Структура БД зі зв'язками

Керування базою даних на етапі проектування виконувалося за допомогою СУБД MySQL Workbench 8.0.27 та phpMyAdmin 5.1.1, яка встановлена на цьому ж веб-сервері.

Надалі все керування здійснюватиметься через розроблений веб-інтерфейс.

Версія PHP: 8.1.0.

Деякі функції вимальовки були виконані на мові JavaScript.

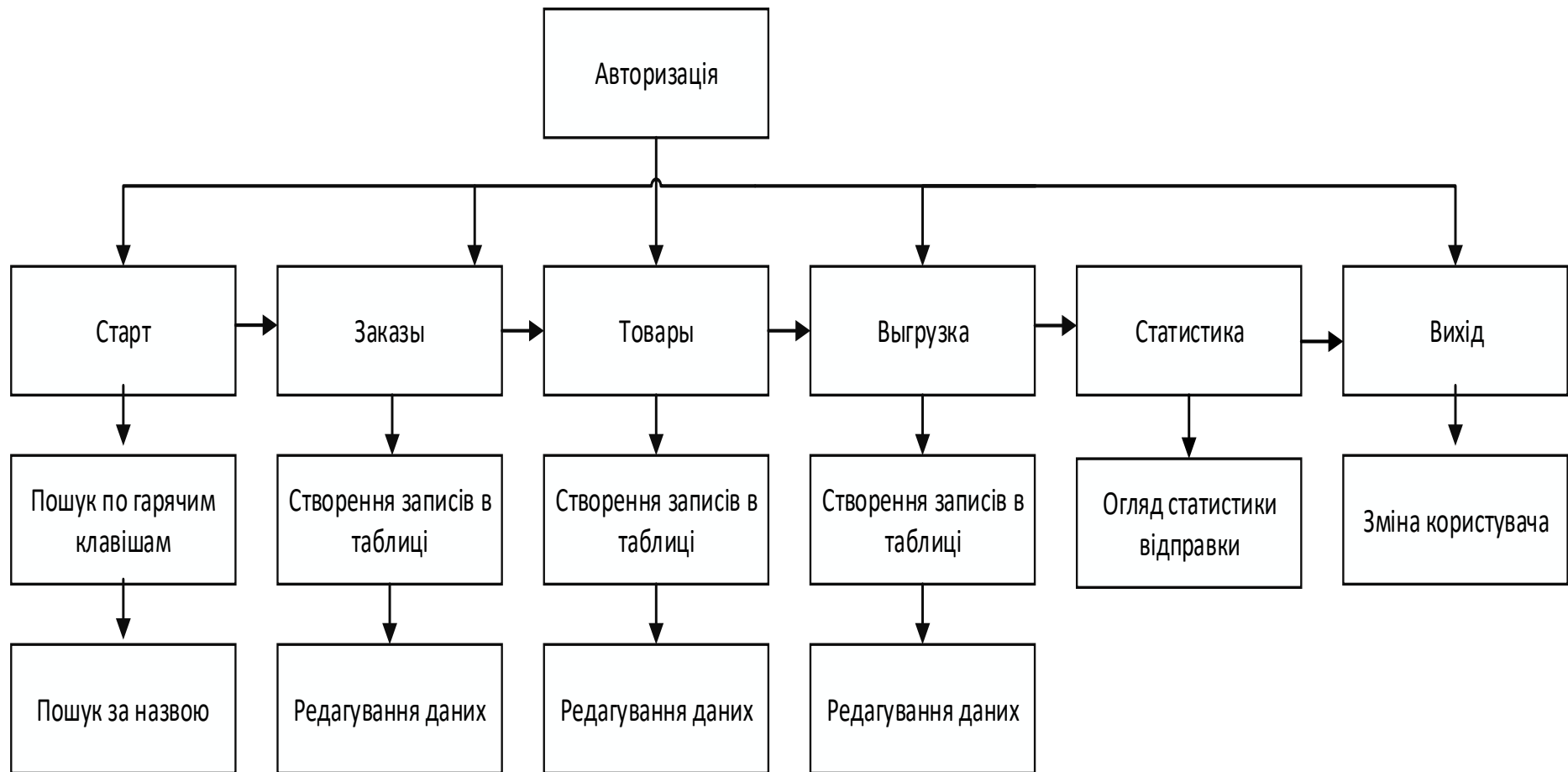


Рисунок 4.4 - Структура сайту



#### **4.5 Вхідні та вихідні дані.**

Вхідні данні будуть схожими з даними які вже зберігаються в БД, але вони не можуть використовуватися всі одночасно. Вихідними даними будуть масиви які повторюють всі поля таблиці БД.

#### **4.6 Висновки по розділ**

В даному розділі були розібрані методи та технології з допомогою яких виконувалась розробка додатку, представленні структури таблиць та зв'язки між ними, також були предаставлені скріншоти сторінок веб-додатку.

## **5 Експериментальний розділ**

### **5.1 Формування вимог до експерименту**

Експериментальним шляхом перевірити вимоги щодо достатності для цього треба виконати всього один запит для отримання коректного результату.

Для проведення треба провести тестування розробленого програмного забезпечення в різних умовах використання.

Мета експерименту – зробити перевірку на адекватність розробленої системи керування базою даних.

### **5.2 Опис експерименту**

Для експерименту будемо використовувати вже розроблене програмне забезпечення яке представлено в 4 розділі.

Роботу веб інтерфейсу можливо користуватися як на телефоні так і на комп'ютері.

### **5.3 Контрольний приклад для проведення випробувань**

#### **5.1.1 Підготовка до запуску та запуск серверу**

Після запуску серверів, переходимо в будь-який браузер для перевірки працездатності системи.

### 5.3.2 Дослідження достатності опису понять

Відкриваємо у вікні браузера розроблений веб-інтерфейс бази даних.

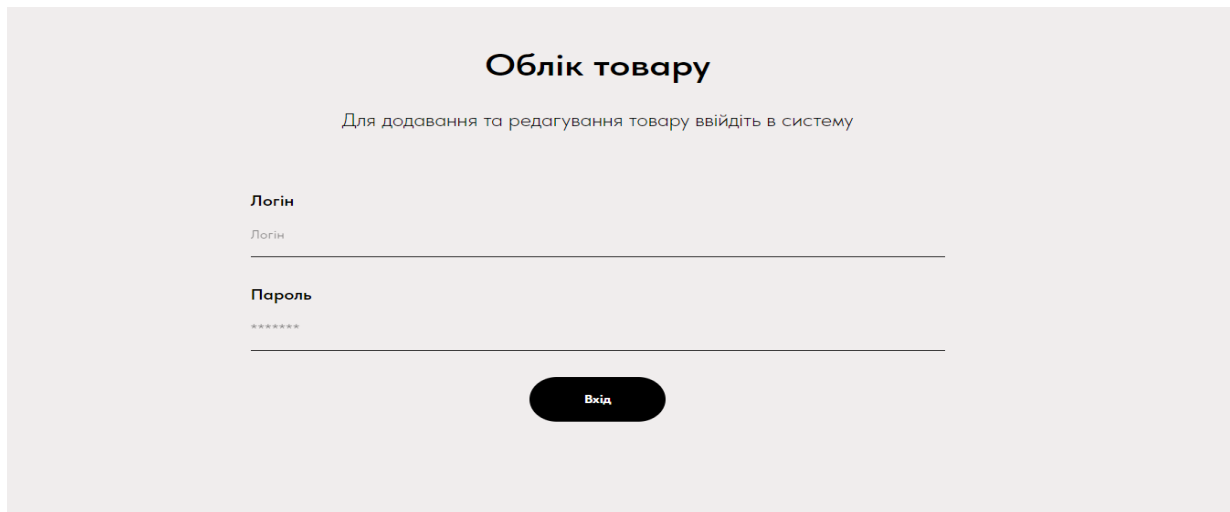


Рисунок 5.1 - Сторінка авторизації

Після успішного входу до системи ми потрапляємо на головну сторінку веб-сайту.

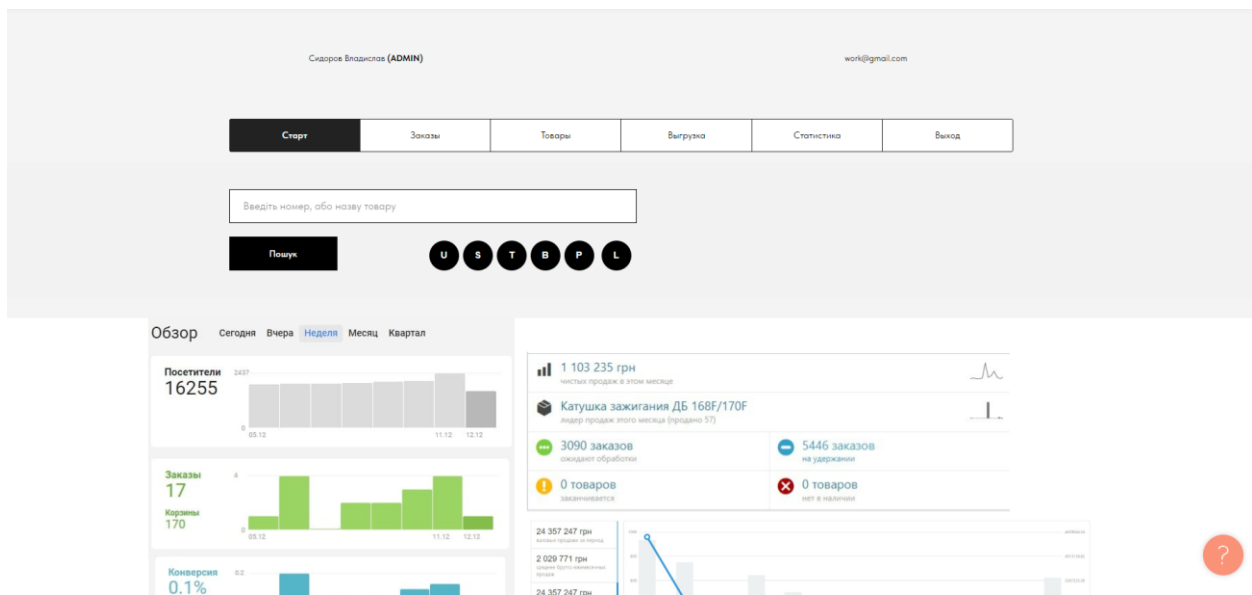


Рисунок 5.2 - Головна сторінка сайту

Для пошуку товару нам треба скористатися базою для його пошуку, тому скористаємося вкладкою База.



Важай

Пошук

U S T B P L

UID	SKU	Title	Brand	Price	Доступність	Редагувати	Видалити
749492162931	MOT-017	BAJAJ AVANGER CRUISE	BAJAJ	22999	Так		X
706359627621	MOT-018	BAJAJ AVANGER STREET	BAJAJ	19999	Так		X
276099529681	MOT-019	BAJAJ BOXER	BAJAJ	26999	Так		X
160829478541	MOT-020	BBAJAJ BOXER BM 150X	BAJAJ	22999	Так		X
772765342891	MOT-021	BAJAJ DOMINAR 400	BAJAJ	22999	Так		X
413881940861	MOT-022	BAJAJ PULSAR 190 NEON	BAJAJ	19999	Так		X
664012155871	MOT-023	BAJAJ PULSAR 200	BAJAJ	19999	Так		X
973140646911	MOT-024	BAJAJ BM 150X DISK	BAJAJ	19999	Так		X
330951891891	MOT-025	BAJAJ BM 150X DISK UG	BAJAJ	19999	Так		X
243109027361	MOT-026	BAJAJ DOMINAR 500 UG	BAJAJ	24999	Так		X
305038048231	MOT-027	BAJAJ PULSAR 150 DTS-I	BAJAJ	19999	Так		X
269659795781	MOT-028	BAJAJ PULSAR RS 200	BAJAJ	19999	Так		X
966529253071	MOT-029	BAJAJ V15	BAJAJ	19999	Так		X

Рисунок 5.5 - Пошук за запитом «ВАЈАЈ»

Отримуємо у відповідь запити 13 позицій які відповідають пошуковому запити.

Наступний крок нашого тестування пошук за індивідуальним номером.

Пошук за індивідуальним номером за часту використовується коли треба щось змінити у існуючому товарі.

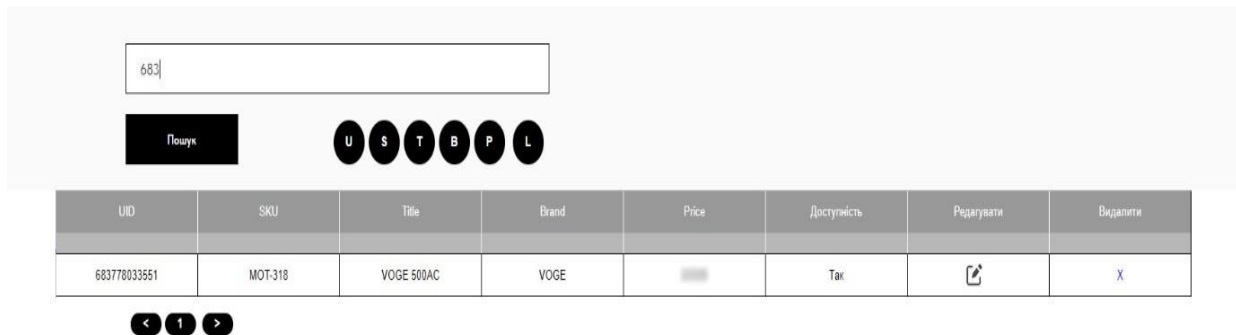


Рисунок 5.6 - Пошук за індивідуальним номером

В даному запиті знайдено лише одна позиція, як і потрібно.

#### 5.4 Висновки по розділу

Враховуючи результати всіх виконаних тестових запитів до бази даних, можу зробити висновок що створена модель відмінно працює та задовольняє умові достатності.

За будь-якою назвою або за індивідуальним номером пошук знаходить всі відповідності.

## ВИСНОВКИ

Кваліфікаційна робота магістра магістра вважається завершеною в ній вирішено задачу – розробка системи доступу до бази даних ТОВ «Усі мотоблоки».

Основні висновки і результати роботи:

– проведено аналіз існуючих проблем в галузі н прикладі цього об'єкта.

– розроблений веб-сервер, який виконує пошук.

– розроблені гарячі клавіші для прискорення обробки даних.

– проведено експериментальні дослідження веб-серверу.

Таким чином, моя задачою яка стояла у розробці системи керування базою даних вирішена в повній мірі.

Мета магістерської роботи - Обґрунтування структури та параметрів комп'ютерної системи цеху збирання ТОВ “Усі мотоблоки” – досягнута.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Інтернет ресурс - <https://studfile.net/preview/5607354/page:3/>  
«Основні переваги СУБД MySQL», дата звертання 24.10.22
2. Поспелов Д.А. Ситуационное управление: теория и практика. – М., 1986. – 288 с.
3. Буров Є. Комп'ютерні мережі. – 2-ге вид., оновлене і доповнене. – Львів. – 2003.- 584 с.
4. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник. - К.: Каравелла, 2004.- 464 с.
5. Клименко О.Ф., Головка Н.Р., Шарапов О.Д. Інформатика та комп'ютерна техніка: Навчально-методичний посібник / За заг. ред. О.Д. Шарапова. – К.: КНЕУ, 2002. – 534 с. 6.
6. Ладичук Д.О., Пічура В.І. Бази даних геоінформаційних систем – Херсон: Вид-во ХДУ, 2006. – с. 68 – с. 47-54.
7. Маслов В.П. Інформаційні системи і технології в економіці: Навчальний посібник.- К.: "Слово", 2005.- 264 с.
8. Ситник В.Ф. Основи інформаційних систем: Навч. посібник. – Вид. 2-ге, перероб. і доп./В.Ф. Ситник, Т.А. Писаревська, Н.В. Єрьоміна, О.С. Краєва; За ред.В.Ф. Ситника. – К.: КНЕУ, 2001.- 420 с.
9. Руденко В.Д., Макарчук О.Н. Комп'ютер та його програмне забезпечення. Курс інформатики (частина I)/За ред. Мадзігона В.М., Бикова В.Ю. – К.: Фенікс, 2001. - 370 с.



## Додаток А

Текст програми

Програмного забезпечення системи керування базою даних

бібліотеки

ТОВ «Усі мотоблоки»

Національний технічний університет  
“Дніпровська політехніка”

Програмне забезпечення  
Системи керування базою даних ТОВ «Усі мотоблоки»

Текст програми

345689078.220

Листів 14

## **АНОТАЦІЯ**

Даний документ містить ПО СУБД ТОВ «Усі мотоблоки»

Текст програми реалізовано на мовах HTML, CSS, PHP, JS  
Середовище розробки та налагодження - Apache 2.4.51 + PHP 8.1,  
MySQL 5.7 в середовищі ОС Windows 10.

## **ЗМІСТ**

1. Текст програми для створення БД

2. Вміст файлу index.php
3. Вміст main.php

### **1 Текст програми для створення БД**

-- MySQL dump 10.13 Distrib 8.0.27, for Win64 (x86\_64)

```

--
-- Host: localhost Database: sklad
-----
-- Server version 5.7.36-log
/*!40101
SET
  @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_C
CLIENT
  */;
/*!40101
SET
  @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_
RESU
  LTS */;
/*!40101
SET
  @OLD_COLLATION_CONNECTION=@@COLLATION_CONNE
CTION
  */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET
@OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
  UNIQUE_CHECKS=0 */;
/*!40014
  @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECK
S,
  FOREIGN_KEY_CHECKS=0 */;

```

```
/*!40101 SET @OLD_SQL_MODE=@ @SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;  
/*!40111 SET @OLD_SQL_NOTES=@ @SQL_NOTES,  
SQL_NOTES=0  
*/;
```

```
--  
-- Table structure for table `mop-pit`  
--  
DROP TABLE IF EXISTS `mop-pit`;  
/*!40101 SET @saved_cs_client = @@character_set_client */;  
/*!50503 SET character_set_client = utf8mb4 */  
CREATE TABLE `els` (  
  `UID` int(11) NOT NULL AUTO_INCREMENT,  
  `sku` varchar(45) COLLATE utf8_unicode_ci NOT NULL,  
  `title` varchar(20) DEFAULT NULL,  
  `brand` varchar(30) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NOT NULL,  
  `Category` varchar(22) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NOT NULL,  
  `Yer` year(4) NOT NULL,  
  `Desc` varchar(500) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci DEFAULT NULL,  
  `cub` int(3) DEFAULT NULL,  
  `fond` varchar(45) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci DEFAULT NULL,
```

```

`Count` int(400) DEFAULT NULL
`price` float DEFAULT NULL,
`dateinsert` date DEFAULT NULL,
`datedelete` date DEFAULT NULL,
`URL` varchar(300) COLLATE utf8_unicode_ci DEFAULT NULL,
EXTERNAL KEY (`mop-pit`),
Unique KEY (`mop-pit`),
UNIQUE KEY `mop-pit_UNIQUE` (`mop-pit`)
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT
CHARSET=utf8
COLLATE=utf8_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
--
-- Table structure for table `els`
--
DROP TABLE IF EXISTS `els`
/*!40101 SET @saved_cs_client = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `els` (
`sk` varchar(6) NOT NULL AUTO_INCREMENT,
`st` int(9) NOT NULL,
`cout` int(11) NOT NULL,
`conf` date DEFAULT NULL,
`PriceOld` int NOT NULL,
`category` varchar(22) DEFAULT NULL,
`Descript` varchar(500) DEFAULT NULL,
`descp` varchar(3) COLLATE utf8_unicode_ci NOT NULL DEFAULT
'Hi',
`note` varchar(45) CHARACTER SET utf8mb4 COLLATE

```

```
utf8mb4_unicode_ci DEFAULT NULL,  
UNIQUE KEY (`mot-pit`)  
KEY `theuser_idx` (`els`),  
KEY `thebook_idx` (`mop-pit`),  
CONSTRAINT `mot-pit` EXTERNAL KEY (`mop-pit`) REFERENCES  
`UID` (`mop-pit`) ON DELETE NO ACTION ON UPDATE NO  
ACTION,  
CONSTRAINT `theuser` FOREIGN KEY (`idusers`) REFERENCES  
`users` (`idusers`) ON DELETE NO ACTION ON UPDATE NO  
ACTION  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT  
CHARSET=utf8  
COLLATE=utf8_unicode_ci;  
/*!40101 SET character_set_client = @saved_cs_client */;
```

## 2 Вміст файлу index.php

```
<?php  
session_start()
```



```

require_once 'config/connect.php';
//var_dump($_SESSION);
if (!empty($_SESSION['user']))
    header("location: /testschoollib/main.php");
if(isset($_GET['link'])){
if ($_GET['link'] === 'logout') {
    unset($_SESSION['user']);
    session_destroy();
    header("location: /testschoollib/index.php");
}}
if (mysqli_connect_errno()) echo mysqli_connect_error();
if (!empty($_POST)) {
    $login = $_POST['login'];
    $password = $_POST['password'];
}
if (isset($_POST['do-registr'])) {
    //var_dump($_POST);
    $login = $_POST['mail'];
    $count = mysqli_fetch_row(mysqli_query($connect, "SELECT
count(*) FROM users WHERE mail = '$login'"));
    //var_dump($count[0]);
    if ($count[0] == 0 && !empty($password)) {
        `status`, `firstname`, `surname`, `mail`, `password`)
        VALUES (NULL, 'new', '$firstname', '$surname', '$login',
'$password')");
        header("location: /testschoollib/index.php?link=auth"); } } if
(isset($_POST['do-auth']))
    }
}

```

```
if (isset($_POST['do-auth'])) {  
    $query_auth = mysqli_query($connect, "SELECT  
`idusers`,`mail`,`password` FROM users WHERE mail = '$login' AND  
password = '$password' ");  
    if (mysqli_num_rows($query_auth) == 1) {  
        $row_auth = mysqli_fetch_assoc($query_auth);  
        $_SESSION['user'] = array($row_auth['idusers'],  
$row_auth['mail'], $row_auth['password']);  
        header("location: /testschoollib/main.php");  
    }  
}  
?>
```

```
<!doctype html>  
<html>  
<head>  
    <meta charset="utf-8">  
    <link rel="stylesheet" href="css/style.css">  
    <title>Обліковий запис</title>
```

```
#message {  
    display: none;  
    background: #f1f1f1;  
    color: #000;  
    position: absolute;  
    left: 0px;  
    top: 0px;  
    padding: 10px;  
    margin-top: 1px;
```

```
width: 100%;
line-height: 0.25%;
}
#message p {
padding: 10px 35px;
font-size: 18px;
}
/* Add a green text color and a checkmark when the requirements
are right */
valid {
color: grey;
}
.valid:before {
position: relative;
left: -35px;
content: "✓";
}
/* Add a red text color and an "x" when the requirements are
wrong */
.invalid {
color: red;
}
.invalid:before {
position: relative;
left: -35px;
content: "✗";
}
</style>
```

```

<body>
    <div style="text-align:center">
        <h3 style="font-size:30px">Вітаємо на веб-сервері обліку
товару</h3 >
    </div>
    <div style="text-align:center; font-
size:200%"><a
href="/testschoollib/index.php?link=auth">Авторизація

<?php
    if (isset($_GET['link'])) {
        $link = $_GET['link'];
        if ($link === 'registr') require_once("vendor/reg.php");
        elseif ($link === 'auth')
            require_once("vendor/auth.php");
    }
?>
</body>
<script>
var myInput = document.getElementById("psw");
var letter = document.getElementById("letter");
var capital = document.getElementById("capital");
var number = document.getElementById("number");
var length = document.getElementById("length");
// When the user clicks on the password field, show the message
box
myInput.onfocus = function() {
    document.getElementById("message").style.display = "block";
}

```

```
// When the user clicks outside of the password field, hide the
message box
myInput.onblur = function() {
document.getElementById("message").style.display = "none";
}
// When the user starts to type something inside the password field
myInput.onkeyup = function() {
// Validate lowercase letters
var lowerCaseLetters = /[a-z]/g;
if (myInput.value.match(lowerCaseLetters)) {
letter.classList.remove("invalid");
letter.classList.add("valid");
} else {
letter.classList.remove("valid");
letter.classList.add("invalid");
}
// Validate capital letters
var upperCaseLetters = /[A-Z]/g;
if (myInput.value.match(upperCaseLetters)) {
capital.classList.remove("invalid");
capital.classList.add("valid");
} else {
capital.classList.remove("valid");
capital.classList.add("invalid");
}
// Validate numbers
var numbers = /[0-9]/g;
if (myInput.value.match(numbers)) {
number.classList.remove("invalid");
```

```
number.classList.add("valid");
} else {
number.classList.remove("valid");
number.classList.add("invalid")
}
// Validate length
if (myInput.value.length >= 6) {
length.classList.remove("invalid");
length.classList.add("valid");
} else {
length.classList.remove("valid");
length.classList.add("invalid");
}
}
</script>
</html>
```

### **3 Вміст файлу main.php**

```
<?php
header("Content-Type: text/html; charset=UTF-8");
```

```
session_start();
if (empty($_SESSION['user'])) header("location:
/testschoollib/index.php?link=");
require_once 'config/connect.php';
// $books = mysqli_fetch_all(mysqli_query($connect, 'SELECT * FROM
books'));
#print_r($books);
$login = $_SESSION['user'][1];
$password = $_SESSION['user'][2];
$main_user = mysqli_fetch_assoc(mysqli_query($connect, "SELECT *
FROM users WHERE mail = '$login' AND password = '$password' "));
#print_r($main_user);
?>
<!DOCTYPE html>
<html lang="ua">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>Товары</title>
</head>
<style>
  .sidenav {
height: 100%;
width: 0;
position: fixed;
z-index: 1;
top: 0;
right: 0;
background-color: #111;
overflow-x: hidden;
transition: 0.5s;
padding-top: 60px;
}
  .sidenav a {
padding: 8px 8px 8px 32px;
text-decoration: none;
font-size: 25px;
```

```
color: #818181;
display: block;
transition: 0.3s;
}
.sidenav a:hover {
color: #f1f1f1;
}
.sidenav .closebtn {
position: absolute;
top: 0;
right: 25px;
font-size: 36px;
margin-left: 50px;
}
.exit {
margin-top: 30px;
}
</style>
<body>
<div id="top" class="topnav">
<a href="main.php" class="active">Старт</a>
<a href="bils.php">Закази</a>
<a href="tov.php">Товари</a>
<a href="vigr.php">Вигрузка</a>
<a href="statis.php">Статистика</a>
<a href="exit.php">Вихід</a>
<div class="topnav-right">
<a style="cursor:pointer" onclick="openNav()">Користувач</a>
</div>
</div>
<div>
<?php include 'userinfo.php' ?>
</div>
<div style="text-align:center">
<h3>Вітаємо на веб-сервері обліку товару</h3>
</div>
<div class="t1">
<form method="post">
<input type="text" placeholder="Введіть номер, або назву товару .."
name="search" value="<?php
```



```

if(isset($_POST['search'])) echo $_POST['search'];?>"><input
type="submit" name="submit" value="Пошук"></button>
</form>
</div>
<div>
<?php
if (isset($_POST['submit'])) {
$search = explode(" ", $_POST['search']);
$count = count($search);
$array = array();
$i = 0;
foreach ($search as $key) {
$i++;
$pol = "name,author,publication,class,data";
if ($i < $count) $array[] = "CONCAT (" . $pol . ") LIKE '%" . $key . "%'
AND (count != 0 OR filepath != ") OR ";
else $array[] = "CONCAT (" . $pol . ") LIKE '%" . $key . "%' AND
(count != 0 OR filepath != ")";
}
$sql = "SELECT
idbooks,name,class,author,data,publication,fond,count,filepath FROM
schoollib.books WHERE " . implode("", $array);
//var_dump($sql);
$query = mysqli_fetch_all(mysqli_query($connect, $sql));
}
?>
<div style="overflow-x:auto;" style="background-color:#aaa;">
<table id="myTable2">
<tr>
<th onclick="sortTable(0)" class="click">ID </th>
<th onclick="sortTable(1)" class="click">Назва</th>
<th onclick="sortTable(2)" class="click">Категорія</th>
<th onclick="sortTable(3)" class="click">Збиральник</th>
<th onclick="sortTable(4)" class="click">Рік випуску</th>
<th onclick="sortTable(5)" class="click">Товар</th>
<th onclick="sortTable(6)" class="click">Опис</th>
<th onclick="sortTable(7)" class="click">Доступно</th>
<th>Замовити</th>
<th>Файл</th>
</tr>

```

```
<?php
if (isset($query)) foreach ($query as $item) {
?>
<tr>
<td><?= $item[0] ?></td>
<td><?= $item[1] ?></td>
<td><?= $item[2] ?></td>
<td><?= $item[3] ?></td>
<td><?= $item[4] ?></td>
<td><?= $item[5] ?></td>
<td><?= $item[6] ?></td>
<td><?= $item[7] ?></td>
<td><a href="addoutput.php?idbooks=<?= $item[0]
?>&idusers=<?= $main_user['idusers'] ?>">Замовити</td>
<td><a href="<?= $item[8] ?>">&#128190;</a>
</td>
</tr>
<?php
}
?>
</table>
</div>
</div>
</body>
```