МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра комп’ютерних наук

**КУРСОВА РОБОТА**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни Бази даних

на тему:

**«База даних для інтернет-магазину біжутерії»**

студента ІІ курсу групи ІСТ-21-1

спеціальності 126 «Інфомраційні системи та технології»

Дешкова Максіма Юрійовича

(прізвище, ім’я та по-батькові)

Керівник Сугоняк І.І.

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 2023 р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ І.І. Сугоняк \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Коротун \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Кравченко \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Г. Чижмотря \_

(підпис) (прізвище та ініціали)

Житомир – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 122 «Комп’ютерні науки»

«ЗАТВЕРДЖУЮ»

Зав. кафедри

\_\_\_\_\_\_\_\_\_\_\_\_ Марина ГРАФ

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_2023 р.

ЗАВДАННЯ

НА КУРСОВУ РОБОТУ СТУДЕНТУ

Дешкову Максіму Юрійовичу

1. Тема роботи: База даних для інтернет-магазину біжутерії,

керівник роботи: Коротун Ольга Володимирівна

1. Строк подання студентом: “\_05\_” червня 2023р.
2. Вихідні дані до роботи: Розробити базу даних для Інтернет-магазину біжутерії.
3. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми

4. Опис роботи програми

5. Програмне дослідження

1. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація PowerPoint

2.Посилання на репозиторій:

https://gitlab.com/ist211\_dmyu/dmyu\_db\_coruseproject\_2023

1. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
| 1,2,3,4 | Сугоняк І.І. | 06.02.2023 | 05.06.2023 |
| 1,2,3,4 | Коротун О.В. | 06.02.2023 | 05.06.2023 |

1. Дата видачі завдання “ 06 ” лютого 2023 р.

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсової роботи | Строк виконання етапів проекту | Примітки |
| 1 | Постановка задачі | 06.02.2023 – 20.03.2023 | Виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 21.03.2023- 02.04.2023 | Виконано |
| 3 | Формулювання технічного завдання | 03.04.2023 – 10.04.2023 | Виконано |
| 4 | Опрацювання літературних джерел | 11.04.2023 – 16.04.2023 | Виконано |
| 5 | Проектування структури | 17.04.2023 – 27.04.2023 | Виконано |
| 6 | Написання програмного коду | 26.04.2023 -24.05.2023 | Виконано |
| 7 | Відлагодження | 26.05.2023 -28.05.2023 | Виконано |
| 8 | Написання пояснювальної записки | 29.05.2023 - 02.06.2023 | Виконано |
| 9 | Захист | 05.06.2023 | Виконано |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дешков М. Ю.

(підпис) (прізвище та ініціали)

**Керівник роботи** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Коротун О. В.

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до курсової роботи на тему сайт «База даних для інтернет магазину біжутерії» складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 40 сторінки друкованого тексту.

Пояснювальна записка має 7 сторінок додатків. Список використаних джерел містить 10 найменувань і займає 1 сторінку. В роботі наведено 24 рисунків. Загальний обсяг роботи – 47 сторінок.

У першому розділі було проведено теоретичний аналіз інтернет магазинів.

У другому розділі проведено проектування і розробка програмного забезпечення.

У третьому розділі проведено тестування програмного продукту.

Висновок містить в собі результати виконаної роботи при створенні бази даних для сайту на тему «Інтернет магазин біжутерії».

У додатку представлений лістинг розробленого програмного продукту.

Ключові слова: PHP, MYSQL, WEB-САЙТ, MVC, СЕРВЕР, СУБД, CMS-СИСТЕМА, АВТОРИЗАЦІЯ, МАГАЗИН, ІНТЕРНЕТ, ДАНІ, АДМІН.

Зміст

[ВСТУП 7](#_Toc136600589)

[РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ 8](#_Toc136600590)

[1.1 Аналіз інформаційних потреб та визначення предметної області дослідження 8](#_Toc136600591)

[1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи 11](#_Toc136600592)

[Висновки з першого розділу 14](#_Toc136600593)

[РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 15](#_Toc136600594)

[2.1 Загальний алгоритм роботи веб-додатку 15](#_Toc136600595)

[2.2 Розробка функціональних алгоритмів роботи програми 17](#_Toc136600596)

[2.3 Розробка програмного забезпечення 18](#_Toc136600597)

[2.4 Проектування структури бази даних за напрямком курсової роботи 20](#_Toc136600598)

[Висновки з другого розділу 25](#_Toc136600599)

[РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ 26](#_Toc136600600)

[3.1 Опис роботи з додатком 26](#_Toc136600601)

[3.2 Реалізація операцій обробки даних в БД 26](#_Toc136600602)

[Висновоки з третього розділу 31](#_Toc136600603)

[РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ 33](#_Toc136600604)

[4.1 Розробка заходів захисту інформації в БД 33](#_Toc136600605)

[4.2 Налаштування параметрів роботи з MySQL 35](#_Toc136600606)

[Висновки з четвертого розділу 36](#_Toc136600607)

[ВИСНОВКИ 36](#_Toc136600608)

[ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 38](#_Toc136600609)

[ДОДАТКИ 39](#_Toc136600610)

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База даних

КР – курсова робота

MVC – Модель–вигляд–контролер (або Модель–представлення–контролер, Model-view-controller, MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

ПЗ – Програмне забезпеченя

# ВСТУП

В данній курсовій роботі наведено процес створення програмного забезпечення (ПЗ), сайту, що стане корисним для користувачів, які шукають біжутерію, використовуючи сучасні технології, методи та підходи адміністрування сайту та використання і зберігання даних в БД.

Актуальність роботи полягає в застосуванні сучасних технологій об'єктно-орієнтованого підходу в програмуванні. Дана курсова робота показує на прикладі роботу ООП. Проект зберігає дані в БД та може надавати аналіз даних, що зберігаються. Сам сайт представляє собою інтернет магазин.

Мета роботи полягає в розробці веб сайту - розробка програмного комплексу з використанням об'єктно-орієнтованого підходу і користувальницьких класів. Для досягнення мети необхідно провести об'єктно-орієнтований аналіз, об'єктно-орієнтоване проектування, об'єктно-орієнтоване програмування і зібрати статистичні результати.

Об'єкт дослідження курсової роботи - процес обробки інформації наданої користувачами для здійснення статистичного зрізу чи надання інформації відповідно за запитом користувача.

Для вирішення поставлених завдань були використані наступні методи дослідження: аналіз, синтез, систематизація, зіставлення, що дозволило більш глибоко вивчити тему, яка вивчається.

## РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

### 1.1 Аналіз інформаційних потреб та визначення предметної області дослідження

Темою даної КР є розробка інтернет магазину біжутерії. Тобто, потрібно розробити веб-сайт, який надаватиме можливості реєстрації, авторизації, пошуку товару, отримання детальної інформації про товар, купівлі товару за допомогою взаємодії з БД.

При розробці даного сайту будемо використовувати об’єктно-орієнтований підхід до проектування ПЗ, адже він має безліч переваг над іншими методами програмування.

Об'єктно-орієнтоване програмування — це метод програмування, заснований на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування. Таким чином, програма являє собою набір об'єктів, що мають стан та поведінку. Об'єкти взаємодіють використовуючи повідомлення. Будується ієрархія об'єктів: програма в цілому — це об'єкт, для виконання своїх функцій вона звертається до об'єктів що містяться у ньому, які у свою чергу виконують запит шляхом звернення до інших об'єктів програми.

Важлива часина даного проекту є зберігання та обробка інформації.

Найзручніший спосіб збереження інформації є база даних.

База даних – це інструмент, використовуючи який, можна збирати й упорядковувати інформацію. У базах даних можна зберігати відомості про людей, продукти, замовлення тощо.

Комп’ютеризована БД – це контейнер об’єктів. Одна БД може містити кілька таблиць.

Тому, в даній програмі буде використано базу даних MуSQL Server.

Для роботи з даним програмним продуктом потрібно знати та використовувати мову SQL.Для роботи з даним програмним продуктом потрібно знати та використовувати мову SQL.

Мова SQL - мова програмування, призначена для управління даними в системах керування базами даних. Дана мова містить ключові слова, які слугають для визначення дій над даними. Наприклад, слово «select» використовується для вибірки даних з таблиць, «insert» - для додавання даних до таблиць, «update» - оновлення, «delete» видалення.

Моде́ль да́них (Data model) — абстрактне представлення реального світу, що відображає тільки ті об'єкти, що безпосередньо стосуються програми. Це, як правило, визначає специфічну групу об'єктів, їх атрибутивне значення і відношення між ними. Використовується механізм представлення і організації просторової моделі даних, або растрової моделі даних. Вона не залежить від комп'ютерної системи і пов'язана тільки зі структурою даних.

Для зручного, гнучкого та інтерактивного написання модулів до сайту використовується MVC паттерн.

Модель–вигляд–контролер (або Модель–представлення–контролер, Model-view-controller, MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування.

**Обґрунтування вибору засобів реалізації**

Система керування базами даних (СУБД) — це програмне забезпечення, призначене для зберігання, отримання, визначення та керування даними в базі даних.

Найважливіший фактор при виборі СУБД – реляційна вона (SQL) чи ні (NoSQL). В обох є свої переваги та недоліки, які варто враховувати при виборі.

Отже, для вирішення яку СУБД використовувати в курсовому проекті було проведено порівняльну характеристику декількох розповсюджених з них.

Таблиця 1.1

Порівняння характеристик СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| Характеристика | MySQL | Oracle | MongoDB |
| Модель даних | Реляційна | Реляційна | Документно-орієнтована |
| Адміністративне керування | Відмінно | Відмінно | Добре |
| Графічні інструменти | Добре | Добре | Добре |
| Простота обслуговування | Відмінно | Відмінно | Відмінно |
| Робота з декількома ЦП | Добре | Відмінно | Добре |
| Одночасний доступ декількох користувачів | Відмінно | Відмінно | Добре |
| Функції з’єднання і вибір індексів | Відмінно | Відмінно | Добре |
| Мова SQL | Відмінно | Добре | Відсутня (нереляційна СУБД) |
| Інтеграція з іншими СУБД | Добре | Добре | Добре |
| Побудова СУБД | Відмінно | Відмінно | Добре |
| Вбудована мова програмування | Добре | Задовільно | Задовільно |
| Підтримка об’єктно-орієнтованих парадигм | Добре | Відмінно | Задовільно |

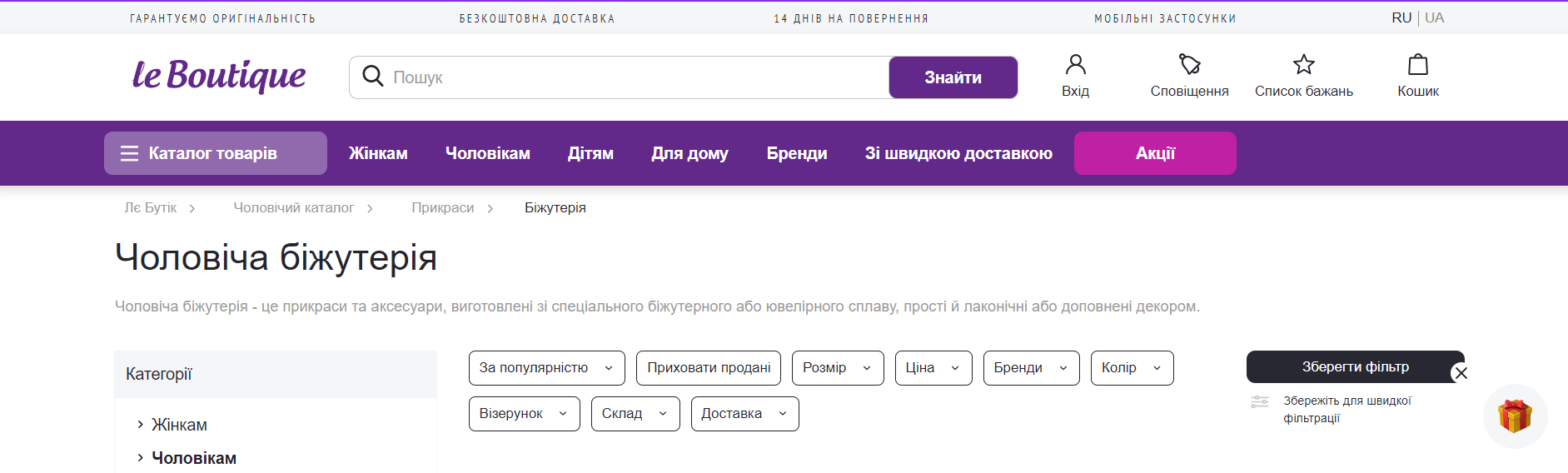
Отже було обрано MySQL, загалом є потужною та надійною СУБД, яка надає широкий функціонал, простоту використання та хорошу продуктивність. З його допомогою можна легко створювати, управляти та оптимізувати бази даних за допомогою мови запитів SQL.

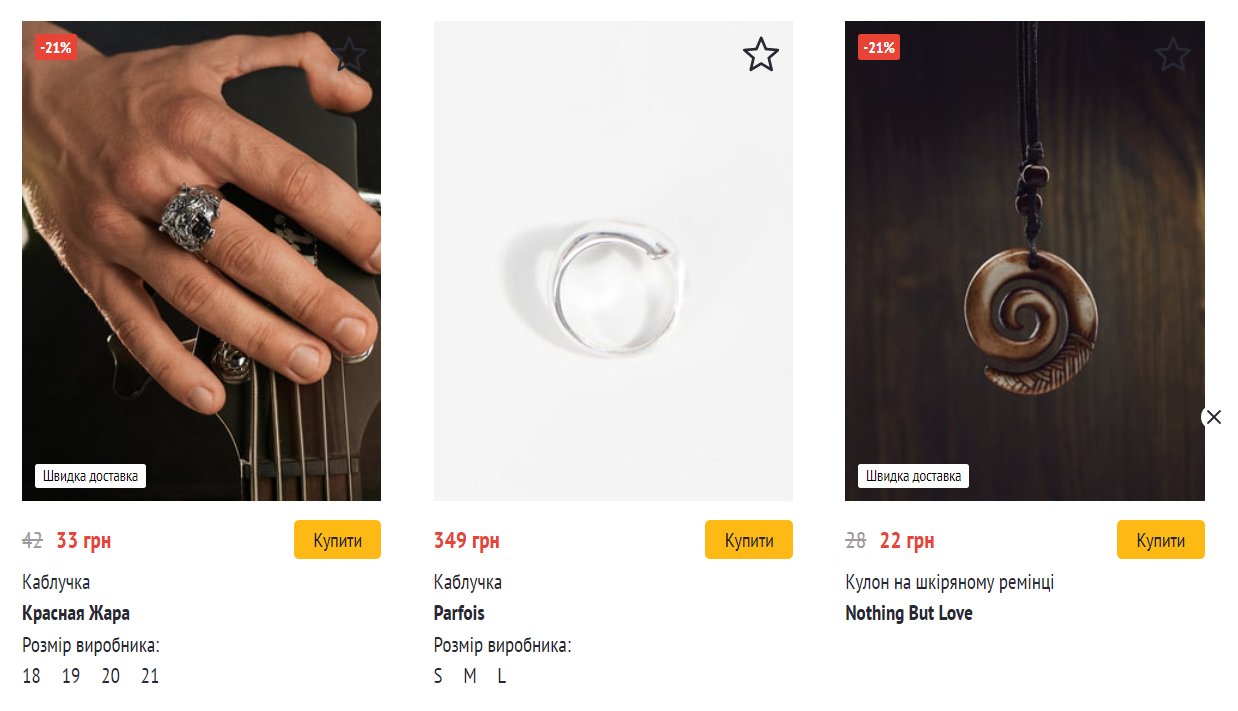
### 1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи

У ході пошуку було знайдено такі Інтернет-магазини:

leBotique

Зовнішній вигляд:



  
Рис. 1.1, 1.2 Зовнішній вигляд інтернет-магазину leBotique

Переваги:

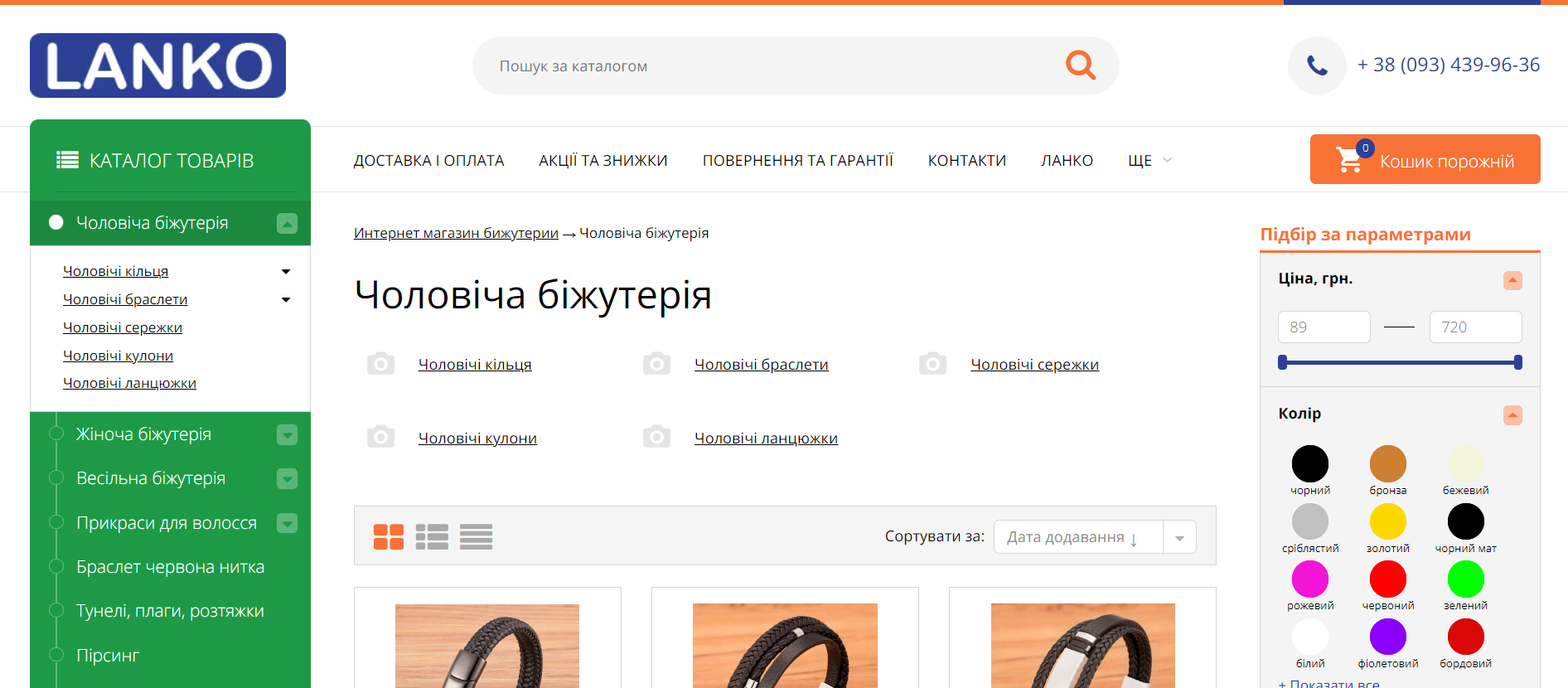
Має гарний структурований дизайн

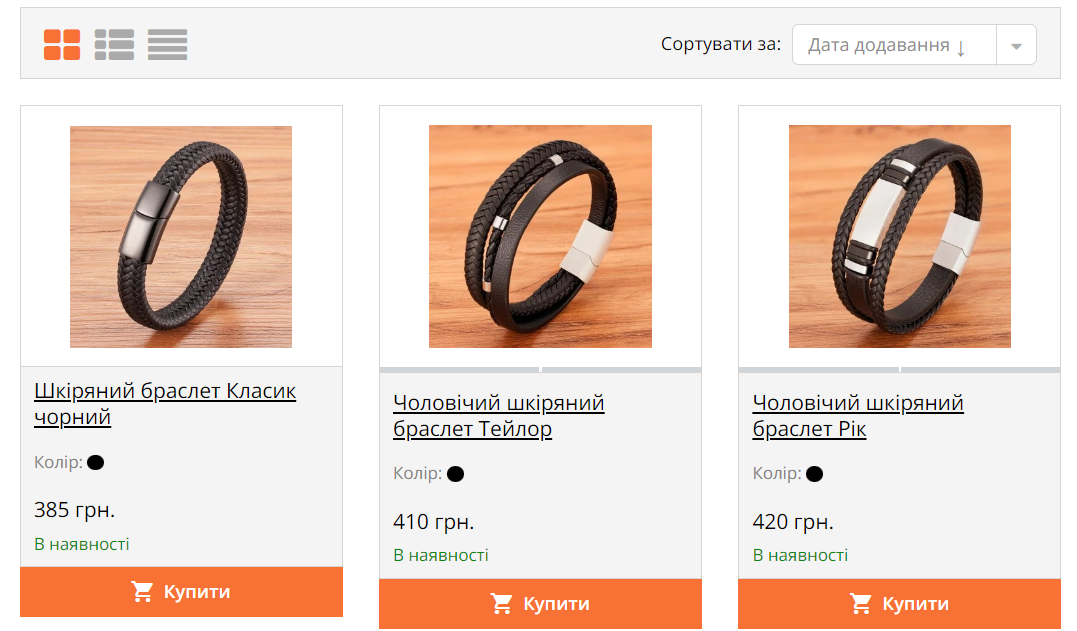
Характеристика продуктів добре складена, дозволяє обрати все необхідне, наявний пошук, коррзина, особистий акаунт – що значно покращує взаємодію з базою та процес покупки товару.

Недоліки: Відсутні.

Lanko

Зовнішній вигляд:



  
Рис. 1.3, 1.4 Зовнішній вигляд інтернет-магазину Lanko

Переваги:

Має простий дизайн.

Характеристика продуктів добре складена і дозволяє обрати все необхідне

Великий функціонал. Наявна корзина, контакти продавців.

Недоліки:

Відсутній особистий кабінет.

Повільна загрузка сторінок. Місцями до хвилини.

Silvers

Зовнішній вигляд:

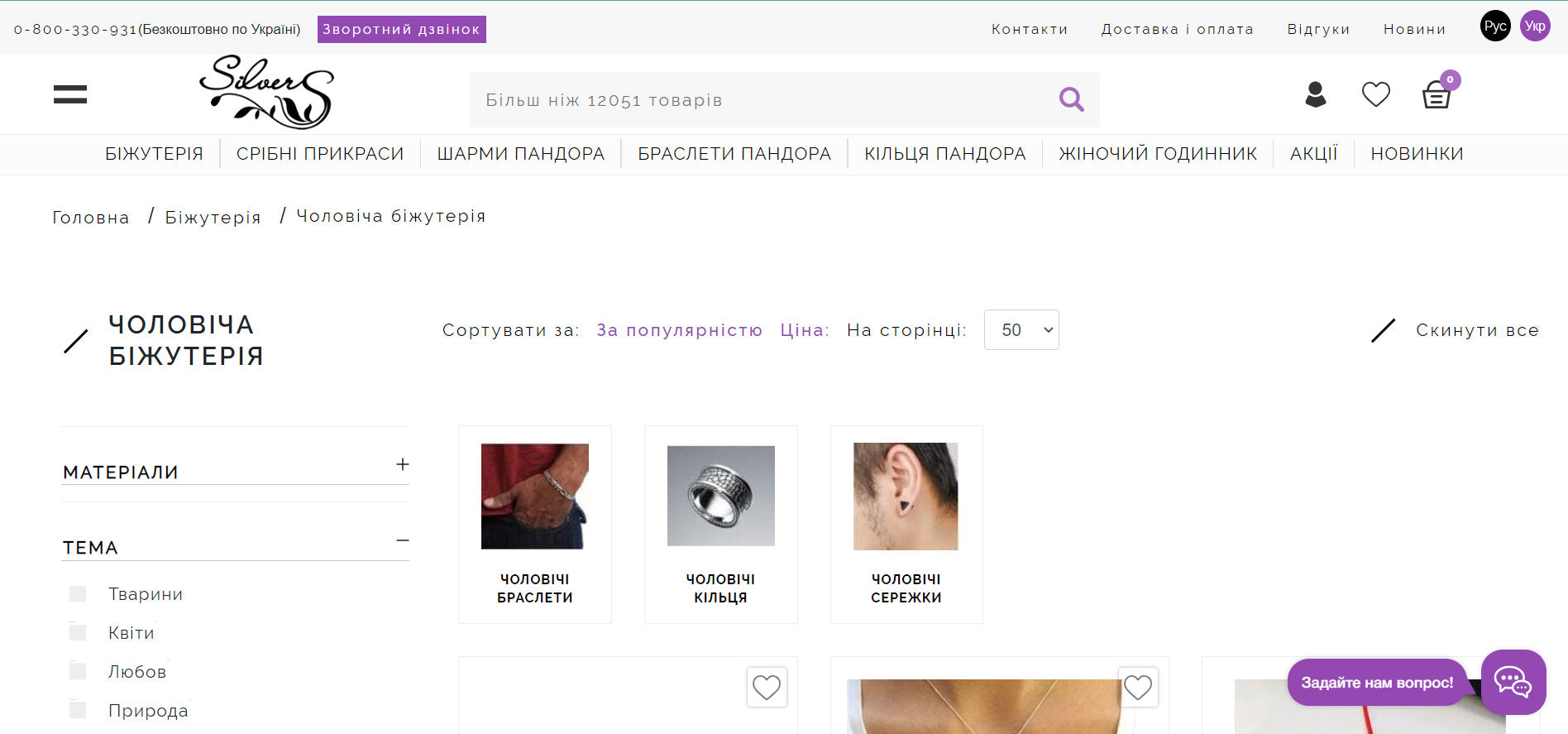
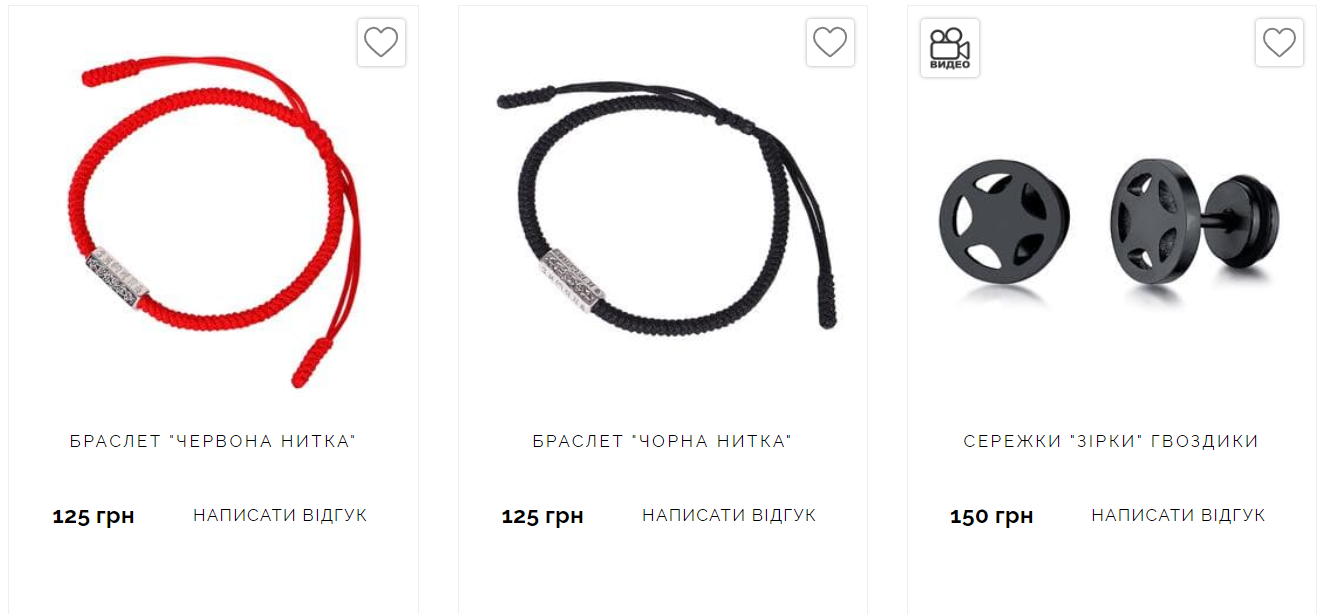
  


Рис. 1.5, 1.6 Зовнішній вигляд інтернет-магазину Silvers

Переваги: Має простий та приємний дизайн, великий функціонал. Наявний особистий кабінет, корзина, контактна інформація.

Недоліки:

Повільна загрузка сторінок, не якісно описана продукція.

### Висновки з першого розділу

В ході виконання першого розділу було праналізовано ПЗ схожого функціоналу та повністю сформовано уявлення про сайт, що розробляється. Отримано теоретичний фундамент для подальшого написання програмного коду, використовуючи ООП, базу даних, MVC паттерн.

Далі було визначено необхідні інструменти для реалізації сайту, а також інтерфейсу та функціоналу сайту, що розробляється.

## РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Загальний алгоритм роботи веб-додатку

На даному етапі КР чітко сформовано задачі, які повинний виконувати розроблений веб-сайт. Тобто визначено інтерфейс та задачі які має виконувати додаток. Розроблено методи вирішення поставлених задач та інтерфейсна частина. Основою для цього буде слугувати попередній розділ. На підставі отриманих даних можна сформувати алгоритм роботи програми, що допоможе реалізувати функціонал та методи виконня поставленої задачі. На рисунку 2.1 показано загальний алгоритм роботи програми.

Так як використовується pattern MVC, загальний алгоритм веб-додатку простий. Він буде використовуватися при кожній дії користувача. Схема алгоритму представлена нижче.



Рисунок 2.1 – Загальний алгоритм роботи веб додатку

Загальний алгоритм роботи додатку:

1. Користувач вводить адресу або натискає на посилання
2. Браузер реагує на ці дії і звертається до відповідного контроллера
3. Далі викликається відповідний метод контроллера
4. Метод контроллера звертається до відповідної моделі
5. Потім відбувається доступ до бази даних для отримання необхідної інформації
6. Далі данні потрапляють назад в модель
7. Через модель вони потрапляють в метод контроллера
8. Метод контроллера передає дані в представлення
9. Представлення з відповідними даними присилається користувачеві як відповідь на його запит

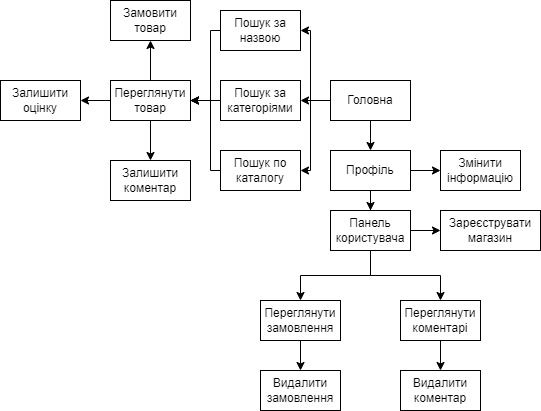


Рисунок 2.2 – Алгоритм дій на сайті в ролі користувача



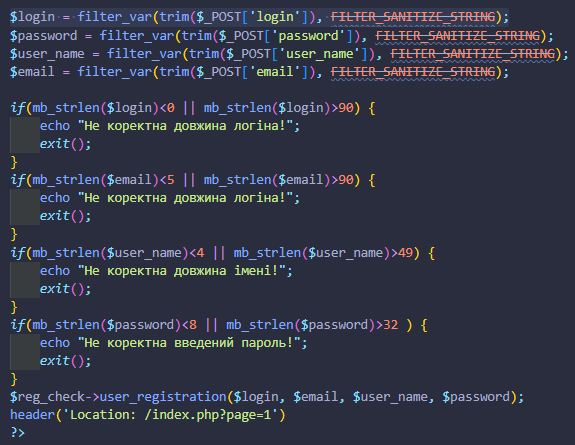
Рисунок 2.3 – Алгоритм дій на сайті в ролі продавця

### 2.2 Розробка функціональних алгоритмів роботи програми

Маючи загальний алгоритм веб-додатку, що ми отримали з попередніх підрозділів, можемо почати безпосередню роботу над проектом.

Найцікавіше в данній КР – це авторизація користувачів. Для авторизації та роботи з акаунтами було створено клас User в якому написані основні функції роботи з користувачами: реєстрація, авторизація, редагування профіля та перевірка коректності введених даних.

Приклад реалізації реєстрації:



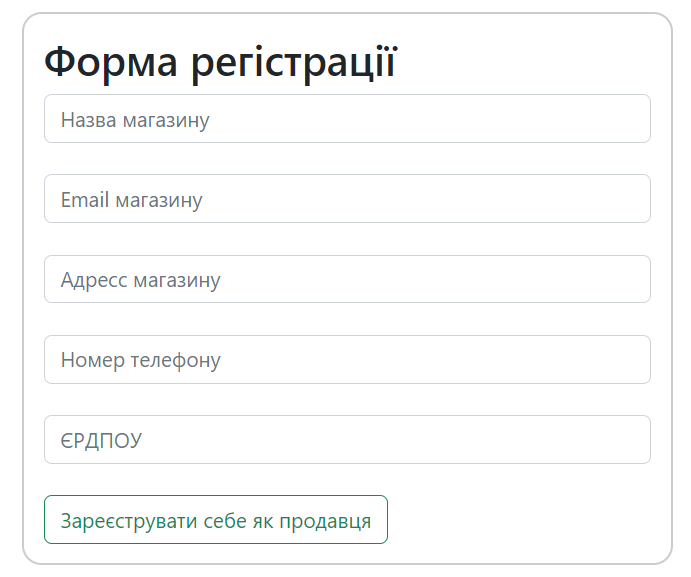


Рисунок 2.4 – Реєстрація користувача

Надалі заходячи у власний кабінет він має можливіть змінювати власні дані. Користувач може бути адміном та просто користувачем. Адмін має змогу додавати, редагувати та видаляти інформацію, звичайні користувачі можуть тільки переглядати товари та робити замовлення а також редагувати свою інформацію.



Рисунок 2.5 – Алгоритм дій користувача/адміна

### 2.3 Розробка програмного забезпечення

На даному етапі розробки курсової роботи відбуваєтсья перехід до безпосередньої розробки проектованого ПЗ. На даний момент вирішені всі організаційні та технічні питання з приводу розроблюваного додатку. Чітко сфомовано задачі, які повинний виконувати веб-додаток, вирішені питання по їх реалізації. Тому, притримуючись технічного завдання та вище оговорених положень можна переходити до розробки інтерфейсу додатку.

На даному етапі безпосередньо переходимо до розробки веб-сайту.

Надалі ми притримуємось MVC – паттерну для моделювання схеми нашого проекту.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

У рамках архітектурного шаблону модель–вигляд–контролер (MVC) програма поділяється на три окремі, але взаємопов'язані частини з розподілом функцій між компонентами. Модель (Model) відповідає за зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача (зміна положення курсора миші, натискання кнопки, ввід даних в текстове поле) і передає дані у модель.

Модель є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель стосується прямого керування даними, логікою та правилами застосунку.

Вигляд може являти собою будь-яке представлення інформації, одержуване на виході, наприклад графік чи діаграму. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації.

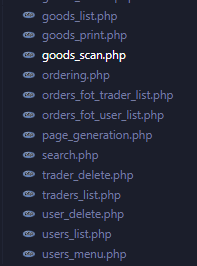
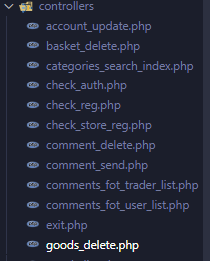
Контролер одержує вхідні дані й перетворює їх на команди для моделі чи вигляду.

Модель інкапсулює ядро даних і основний функціонал їхньої обробки і не залежить від процесу вводу чи виводу даних.

Вигляд може мати декілька взаємопов'язаних областей, наприклад різні таблиці і поля форм, в яких відображаються дані.

У функції контролера входить відстеження визначених подій, що виникають в результаті дій користувача. Контролер дозволяє структурувати код шляхом групування пов'язаних дій в окремий клас.

Таким чином ми створюємо шаблон для зовнішнього вигляду сайту, підставляючи з середину частину, що стосується кожної сторінки окремо.



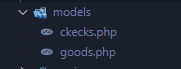


Рисунок 2.6 – Файлова структура сайту(model, controller)

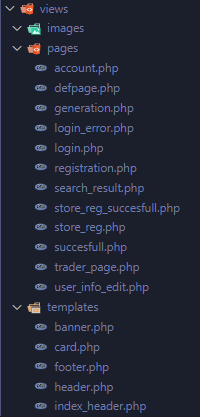


Рисунок 2.7 – Файлова структура сайту (pages, templates)

### 2.4 Проектування структури бази даних за напрямком курсової роботи

Щоб без проблем працював сайт та зберігалися дані, повинна бути створена БД. Для цього ми використаємо БД MySQL та зберігаємо дані в PhpMyAdmin.

Всього для зручного функціонування сайту було створено 10 таблиць:

- basket;

- comments\_to\_goods;

- goods;

- stores;

- users;

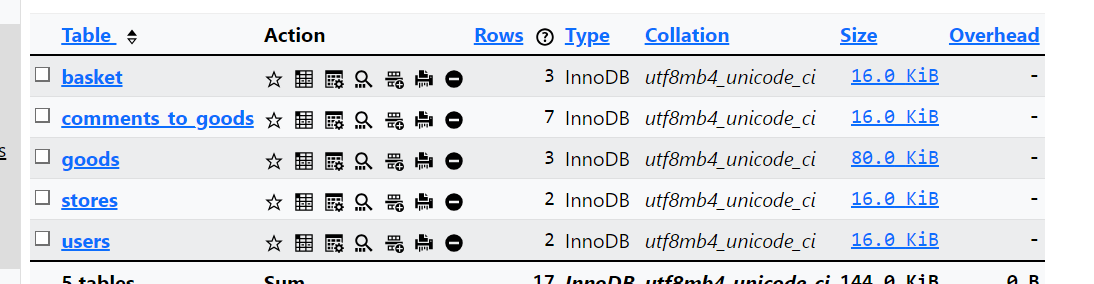


Рисунок 2.8 – Вигляд бази даних в phpMyAdmin

Розберемо структуру кожної таблиці.

Таблиця «basket»

Таблиця 2.1

Структура таблиці

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Призначення** |
| Id\_backet | int | Ідентифікаційний код товару |
| id\_user | int | Ідентифікаційний кот замовника |
| Id\_goods | int | Ідентифікаційний код продавця |
| count | int | Кількість прекрас |

Дана таблиця відбовідає за вкладку «Замовлення». В ній зберігаються всі дійсні замовлення.

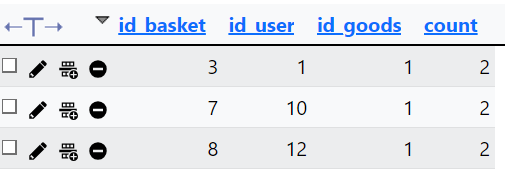


Рисунок 2.9 – Таблиця з замовленнями

Таблиця «comments\_to\_goods»

Таблиця 2.2

Структура таблиці

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Призначення** |
| Id\_comment | int | Ідентифікаційний код коментаря |
| Id\_user | int | Ідентифікаційний код автора |
| user\_name | varchar | Імя автора |
| Id\_goods | int | Ідентифікаційний код прикраси |
| comment | varchar | Вміст коментарю |
| rating | int | Оцінка 1-10 |

В даній таблиці зберігається інформація про коментарі, оцінку, їх вміст.



Рисунок 2.10 – Таблиця з коментарями.

Таблиця «goods»

Таблиця 2.3

Структура таблиці

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Призначення** |
| Id\_goods | int | Ідентифікаційний код прикраси |
| name | varchar | Назва |
| description | varchar | Опис |
| cost | int | Ціна |
| material | varchar | Матеріал |
| type | varchar | Тип прикраси |
| size | varchar | Розмір |
| made\_by | varchar | Ким виготовлено |
| Img1,img2,img3 | longblob | Зображення прикраси |
| Id\_store | int | Ідентифікаційний код магазину |

В цій табліці зберігаються всі товари добавлені продавцями.



Рисунок 2.11 – Таблиця з прикрасами

Таблиця «stores»

Таблиця 2.4

Структура таблиці

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Призначення** |
| Id\_store | int | Ідентифікаційний код магазину |
| name | varchar | Назва |
| adress | varchar | Адрес |
| email | varchar | Електронний адрес |
| Id\_user | int | Ідентифікаційний код власника |
| phone\_number | int | Номер телефону |
| erdpou | varchar | ЄРДПОУ |

В цій таблиці зберігаються всі магазини зареєстровані користувачами.

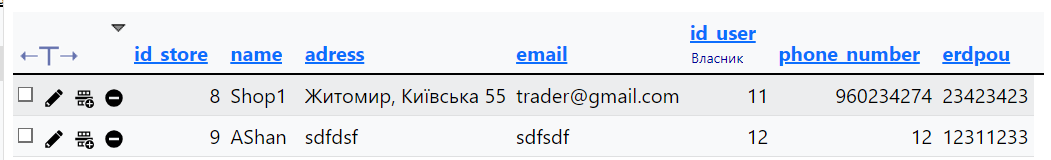


Рисунок 2.12 – Таблиця з магазинами.

Таблиця «users»

Таблиця 2.5

Структура таблиці

|  |  |  |
| --- | --- | --- |
| **Назва** | **Тип поля** | **Призначення** |
| Id\_user | int | Ідентифікаційний код |
| login | varchar | Логін |
| email | varchar | Електронний адрес |
| name | varchar | Ім’я |
| password | varchar | Пароль |
| adress | varchar | Адрес |
| Phone\_number | varchar | Номер телефону |
| trader | tinyint | Чи являється користувач продавцем? true/false |
| admin | tinyint | Чи являється користувач адміністратором? true/false |

Дана таблиця містить в собі всіх зареєстрованних користувачів.

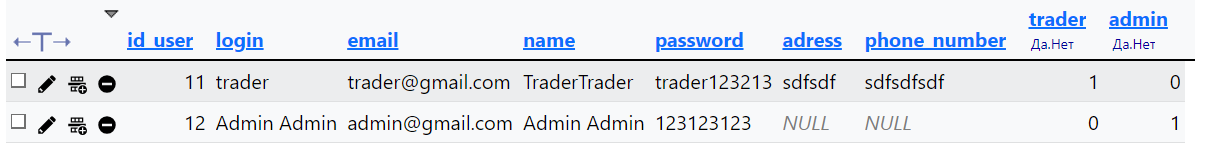


Рисунок 2.13 – Таблиця з користувачами

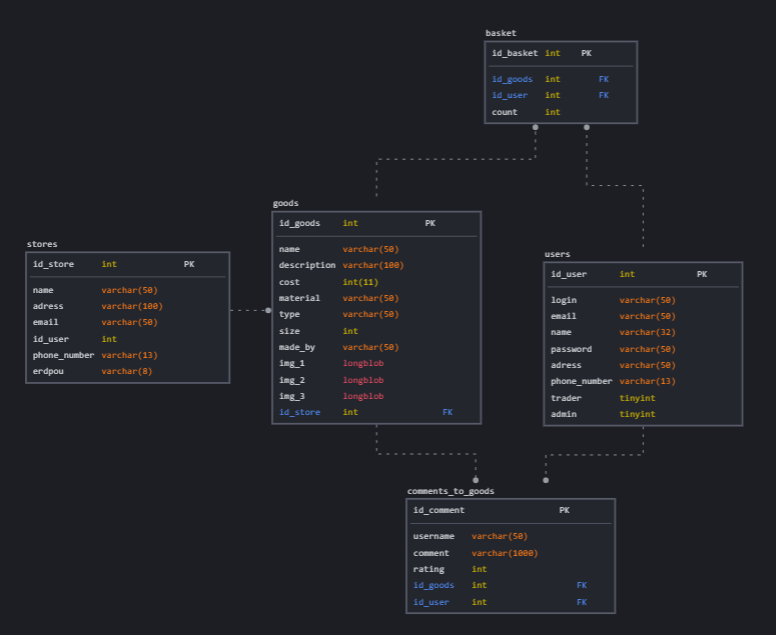


Рисунок 2.14 – Діаграма бази даних та таблиць

### Висновки з другого розділу

В даному розділі курсової роботи було проведено розробку загального і функціонального алгоритмів сайту, бази даних та описано принцип виконання основних методів роботи.

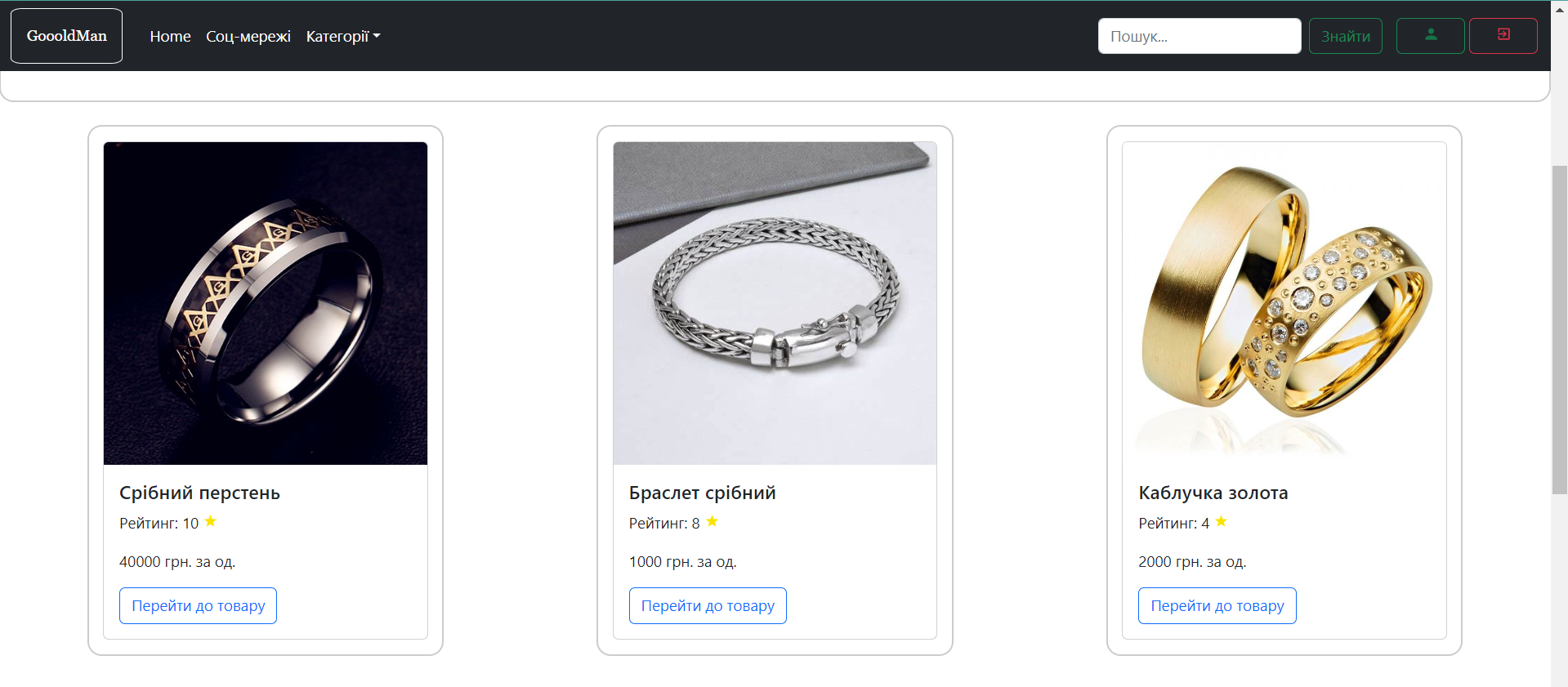
Розроблено загальний алгоритм функціонування роботи програми. Показані та проаналізовані таблиці бази даних, продемонстрована діаграма бази даних.

В результаті було отримано готовий програмний продукт з функціональністю, який виконує всі поставлені задачі та вимоги.

## РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

### 3.1 Опис роботи з додатком

Коли користувач підключається до сайту він бачить головну сторінку:

Рис. 3.1 Головна сторінка

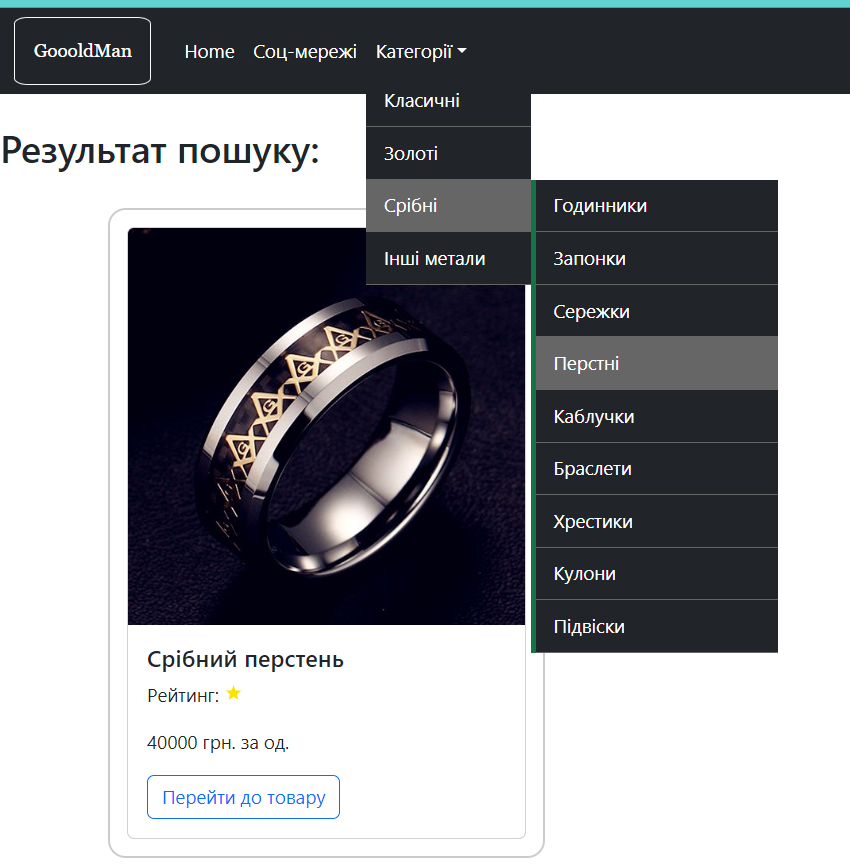
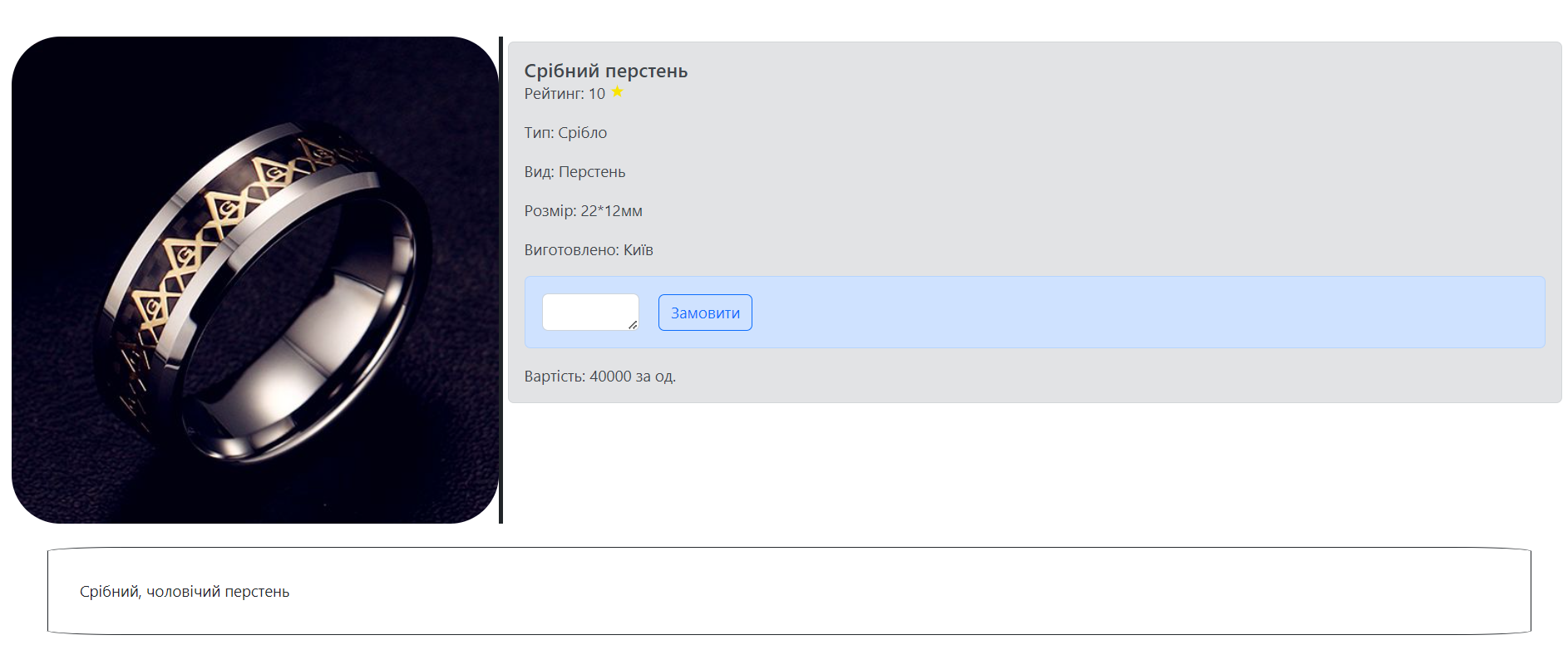
Полиставши трішки користувач може ознайомитись з ассортиментом, також він може використати категорії або пошук щоб прискорити пошук необхідних йому речей.  


Рис. 3.2 Пошук за категорією

Далі користувач може переглянути інформацію про продукт, натиснувши на карту товару, але не може його замовити та залишити коментар якщо не зареєстрований:

  
Рис. 3.3 Картка товару

Перейшовши на посилання «зареєструватись» або «авторизуватись» , користувач потрапляє на таку сторінку та вводить там свої коректні данні. Якщо все коректно, то він отримує повідомлення:

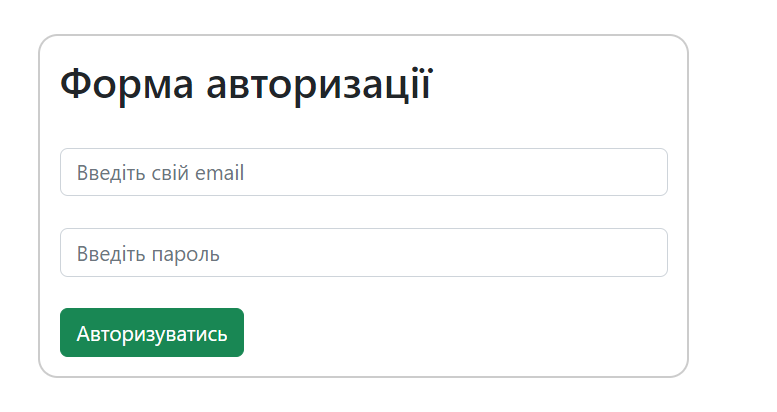
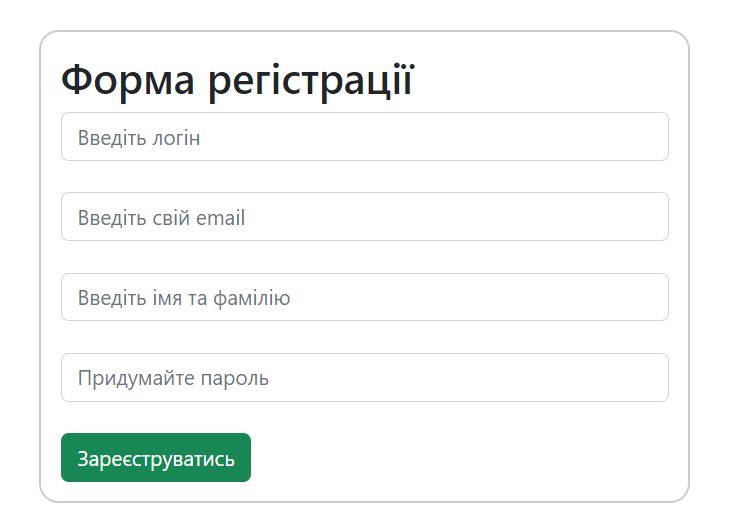
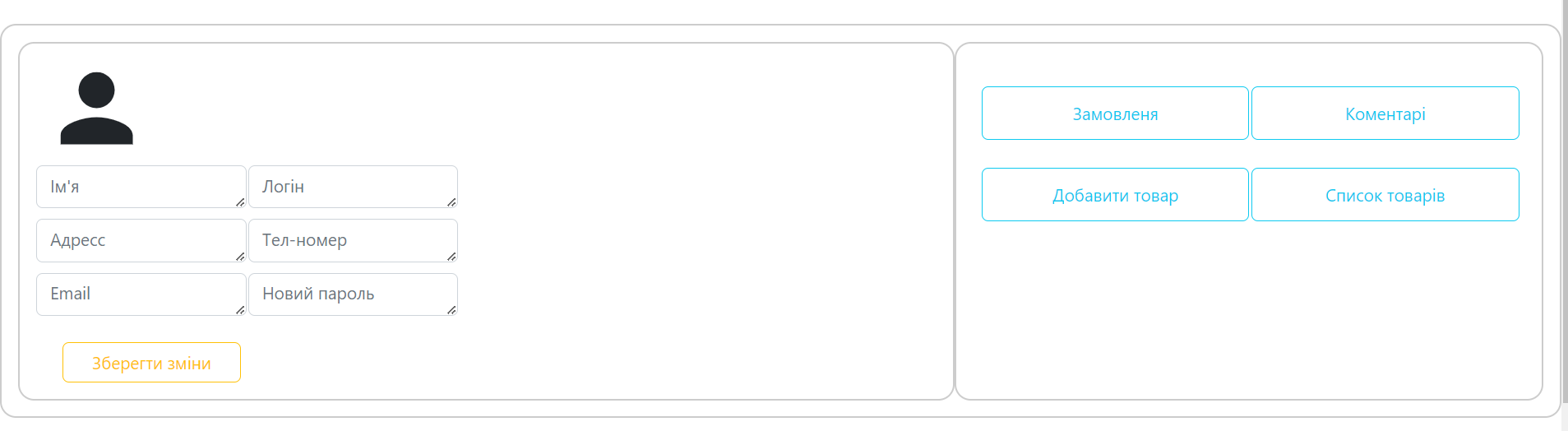
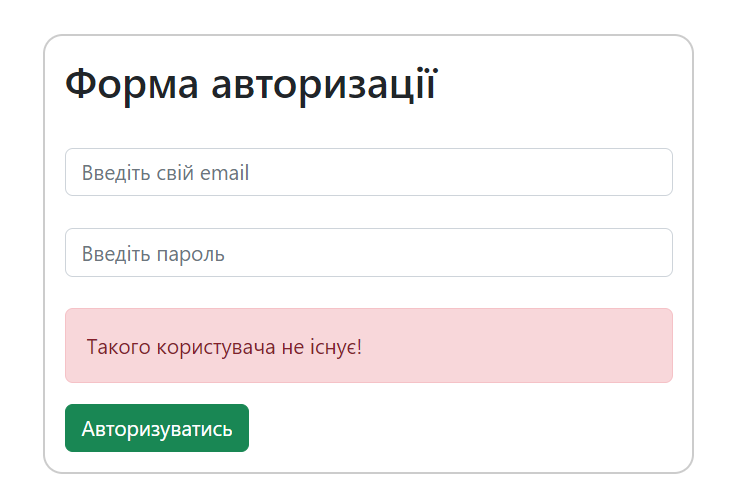


Рис. 3.4, 3.5 Авторизація та реєстрація.

В зареєстрованому вигляді користувач може замовляти, та переходити в свій профіль, де в нього з ’ являться нові можливості(редагування акаунту та ін.):

  
Рис. 3.6 Функціонал профілю.

Наведемо приклади деяких помилок.

  
Рис. 3.6 Помилка авторизації.

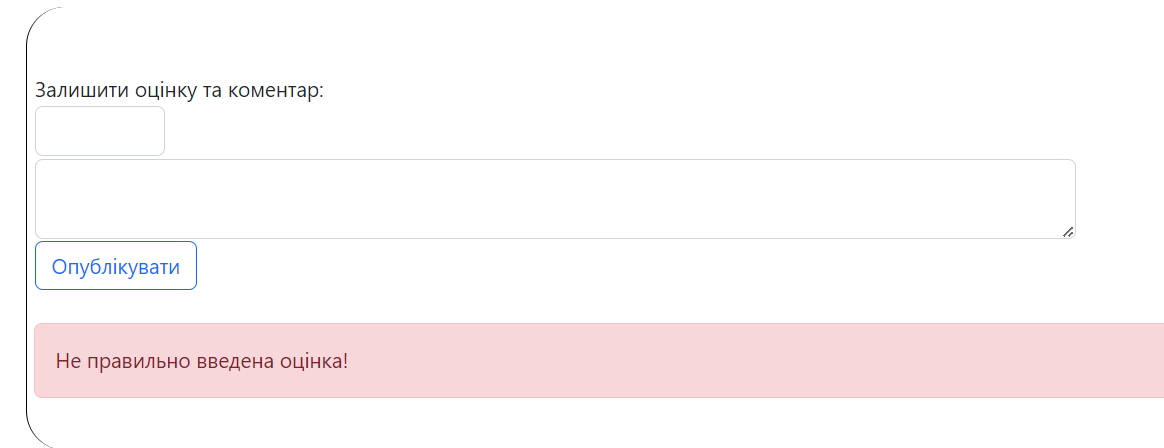
  
Рис. 3.7 Помилка коментаря.

  
Рис. 3.8 Помилка яка виникає коли користувач виконує дії не авторизовано.

В більшості ситуаціях перевірки зроблені в контроллер частині, наприклад перевірки на коректність даних, на довжину і так далі, це зроблено для того щоб не передавати в ‘model’ криві данні, і лишній раз не навантажувати систему.

### 3.2 Реалізація операцій обробки даних в БД

Операції над товарами об`єднані в один клас goods, та містять в собі різного поду операції над товарами. Додавання, видалення, редагування, тощо. Ось декілька прикладів:

Виведення всіх товарів:

public function goods\_mass()

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `goods`");

          $mysql->close();

          return $result;

     }

Редагування товару:

 public function update\_data($name, $cost, $size, $madeby, $material, $type, $description, $id\_store, $id\_goods){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("UPDATE `goods`  SET `name` = '$name', `cost` = '$cost', `size` = '$size', `made\_by` = '$madeby', `material` = '$material', `type` = '$type', `description` = '$description', `id\_store` = '$id\_store' WHERE `id\_goods` = '$id\_goods'");

          $mysql->close();

     }

Витадення товару:

   public function goods\_delete($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("DELETE FROM `goods` WHERE `goods`.`id\_goods` = '$id'");

          $mysql->close();

     }

Пошуку за різними параметрами:

  public function search\_by\_material\_and\_type($material, $type){

          $material = '%'.$material.'%';

          $type = '%'.$type.'%';

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `goods` WHERE `material` like '$material' and `type` like '$type'");

          $mysql->close();

          return $result;

          }

public function search\_by\_material($material){

               $material = '%'.$material.'%';

               $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

               $result = $mysql->query("SELECT \* FROM `goods` WHERE `material` like '$material'");

               $mysql->close();

               return $result;

          }

Також, наявний клас перевірик, який вміщує собі контроль доступу, перевірки паролів, логінів, шифрування, тощо. Ось декілька функцій з цього класу:

Авторизація:

public function user\_authorize($pass, $email){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT\*FROM `users` WHERE `email` = '$email' AND `password` = '$pass'");

          $user = $result->fetch\_assoc();

          $mysql->close();

          return $user['id\_user'];

     }

Реєстрація:

public function user\_registration($login, $email, $name, $pass){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query(

          "INSERT INTO `users`(`login`,`email`, `password`, `name`, `trader`)

          VALUES('$login','$email', '$pass', '$name', '0')");

          $mysql->close();

     }

Перевірки привілегій користувача :

public function this\_user\_trader($id)

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT\*FROM `users` WHERE `id\_user` = '$id'");

          $result\_fetch = $result->fetch\_assoc();

          if($result\_fetch['trader']==1){

               return true;

          }

          else{

               return false;

          }

          $mysql->close();

     }

     public function this\_user\_admin($id)

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT\*FROM `users` WHERE `id\_user` = '$id'");

          $result\_fetch = $result->fetch\_assoc();

          if($result\_fetch['admin']==1){

               return true;

          }

          else{

               return false;

          }

          $mysql->close();

     }

### Висновоки з третього розділу

В даному розділі курсвої роботи було описано методологію користування стовреним сайтом, та проведено тестування створеного додатку.

Опис роботи з сайтом включає в себе текстову та ілюстраційну чатину, які пояснюють принципи роботи з сайтом та опис елементів інтерфейсу сайту. При описі було роз’яснено призначення кожного елемента інтерфейсу у всіх вікнах.

При проведені тестування додаток був запущений на виконная у реальних умовах, при яких показав стабільну і коректну роботу.

**РОЗДІЛ 4. АДМІНІСТРУВАННЯ БАЗ ДАНИХ**

**4.1 Розробка заходів захисту інформації в БД**

Функція backup:

 function createDatabaseBackup(){

    $backupDir = ROOT.'/';

    date\_default\_timezone\_set('Europe/Kiev');

    $currentTime = time();

    $backupTime = strtotime(date('Y-m-d 23:00:00'));

    if ($currentTime > $backupTime) {

        $filename = 'backup\_' . date('Y-m-d') . '.sql';

        $filePath = $backupDir . $filename;

        if (!file\_exists($filePath)) {

            $command = "mysqldump --user=root --password=root --host=localhost coursework > {$filePath}";

            exec($command);

        }

    }

}

Для доступу до бази даних було створено 4 види користувачів з різними типами доступу, а саме: незареєстрований користувач, зареєстрований користувач, продавець, адміністратор. Тепер детальніше розгялнемо кожну з них.

1) Незареєстрований користувач: можливість перегляду каталогу товару, інформацію про товари, перегляд інформації про бренд салону, про бренди з якими співпрацює салон, контактну інформацію. Не має особистого кабінету та не зареєстрований на сайті. Його інформація не наявна в базі даних.

2) Зареєстрований користувач: можливість перегляду каталогу товару, інформацію про товари, перегляд інформації про бренд салону, про бренди з якими співпрацює салон, контактну інформацію. Можна здійснити купівлю товаару, додати його в кошик, оформити замовлення. Користувач має свій кабінет, його данні наявні в базі даних та він може редагувати свої особисті дані та переглядати свої замовлення.

3) Продавець: можливість перегляду каталогу товару, інформацію про товари, перегляд інформації про бренд салону, про бренди з якими співпрацює салон, контактну інформацію. Можна здійснити купівлю товару, додати його в кошик, оформити замовлення. Продавец має свій кабінет, його данні наявні в базі даних та він може редагувати свої особисті дані та переглядати свої замовлення. Також він має сторінку продавця, на котрій може здійснювати додавання товару на сайт, редагувати та видаляти свій товар.

4) Адміністратор: можливість перегляду каталогу товару, інформацію про товари, перегляд інформації про бренд салону, про бренди з якими співпрацює салон, контактну інформацію. Можна здійснити купівлю товаару, додати його в кошик, оформити замовлення. Адміністратор має свій кабінет, його данні наявні в базі даних та він може редагувати свої особисті дані та переглядати свої замовлення. Роль в базі даних зазначеня, як admin. Також адміністратору пресмтавлена адмінпанель, де він може переглядати, редагувати, додавати, видаляти дані з таблиць бази даних. Також може переглядати статистику продажів за сьогоднішній день, за місяць та за весь час.

Щоб краще відобразити доступ до різних ділянок сайту створемо матрицю доступу яка віалузуватими вище сказане. На перетені колонок введемо умовні позначення доступу а саме:

0 – немає доступу 1 – читання 2 – вставка 3 – редагування 4 – видалення 5 – повний доступ

Таблиця 4.1

Матриця доступу ролі

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиці / Ролі |  | Матриця доступу | | |
| Незареєстрований | Користувач | Продавець | Адміністратор |
| basket | 0 | 1, 2, 3,4 | 1, 2, 3,4 | 5 |
| Comments\_to\_goods | 1 | 1, 2, 3,4 | 1, 2, 3,4 | 5 |
| goods | 1 | 1 | 1, 2, 3, 4 | 5 |
| stores | 0 | 0, 2 | 3, 4 | 5 |
| users | 2 | 1, 3 | 1, 3 | 5 |

**4.2 Налаштування параметрів роботи з MySQL**

Пріцювати з базою даних будемо через phpMyAdmin, який є вбудованим в локальний сервер Open Server. Для запуска сервера використовуємо права адміністратора.

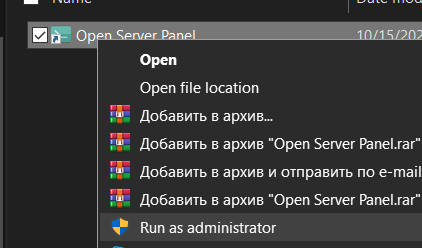
****

Рис.4.1 – Запуск сервера від імені адміністратора

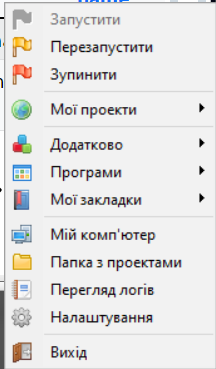


Рис.4.2 – Запуск сервера

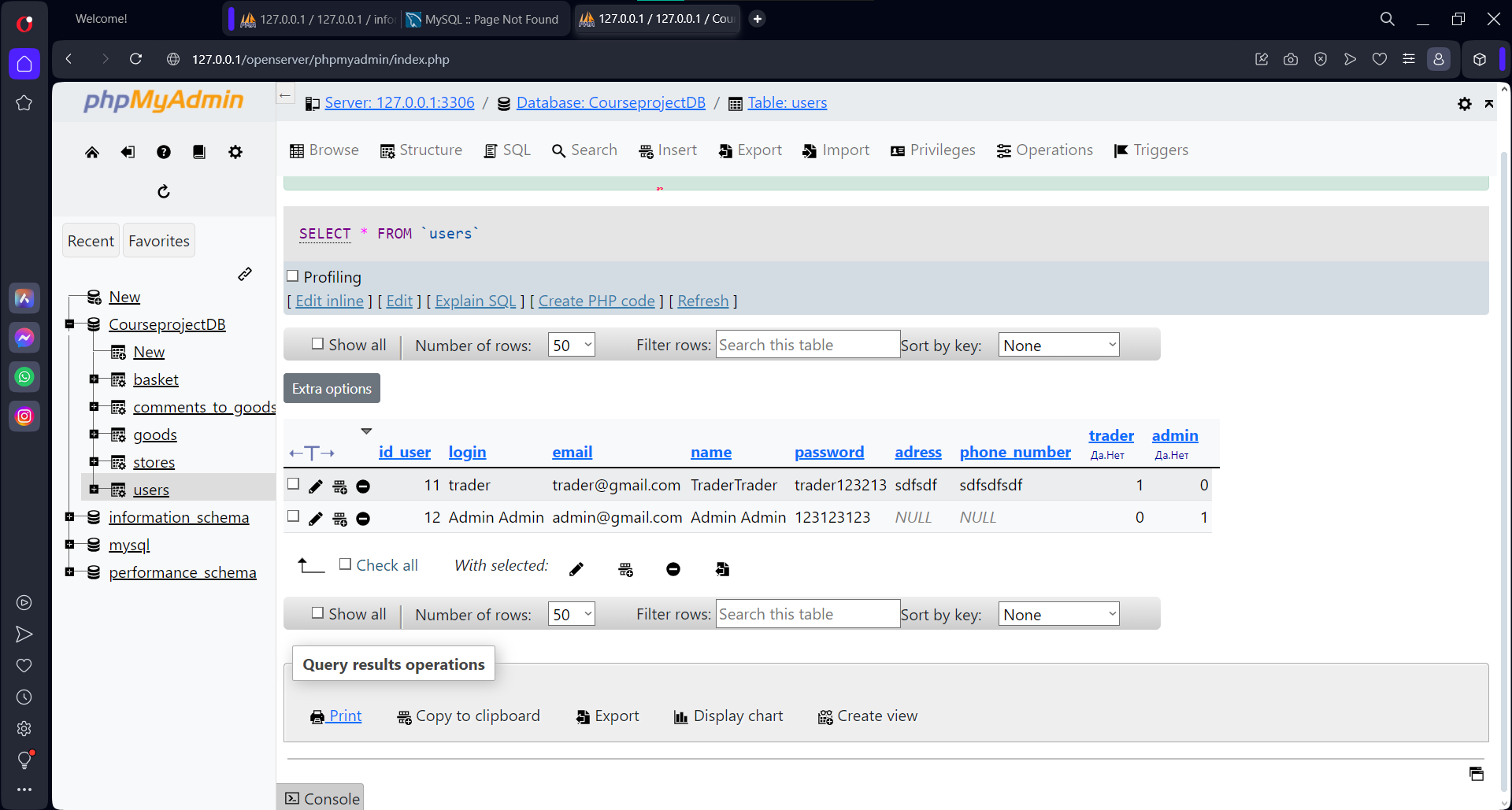


Рис.4.3 – Вікно PhpMyAdmin

**Висновки з четвертого розділу**

У четвертому розділі були розглянуті ролі та представлено матрицю доступу. Було представлено налаштування БД за допомогою PhpMyAdmin від OpenServer.

# **ВИСНОВКИ**

У першому розділі було проаналізовано сайти схожого функціоналу та повністю сформовано уявлення про сайт, що розробляється. Отримано теоретичний фундамент для подальшого написання програмного коду, використовуючи ООП та базу даних.

Далі було визначено необхідні інструменти для реалізації програмного забезпечення, а також інтерфейсу та функціоналу програми, що розробляється.

У другому розділі було проведено розробку загального і функціонального алгоритмів програм, розроблених на основі сформованого уявлення про задачі, які повиннен вирішувати даний сайт та функціонал, який він повинен містити.

Розроблено загальний алгоритм функціонування роботи сайту. Була розроблена інтерфейсна частина, змодельовані вкладки сайту, та спроектовано програмний код сайту. Також було розгялнуто структуру бази даних та таблиці.

В результаті було отримано готовий програмний продукт з повною функціональністю, який виконує всі поставлені задачі та вимоги.

У третьому розділі курсової роботи було описано методологію користування створеним сайтом, та проведено тестування створеного сайту.

При проведені тестування сайт був запущений на виконная у реальних умовах, при яких показав стабільну і коректну роботу.

У четвертому розділі розглянуто ролі програмного продукту, представлено матрицю доступу та розглянуто налаштування параметрів роботи з базою даних.

В результаті виконання курсової роботи було розроблено сайт та базу даних, що повністю відповідає поставленим вимогам. Сайт є простим у користуванні, зі зрозумілим інтерфейсом.

Спроектований сайт та база даних повністю відповідає поставленим вимогам, та не вимагає великих ресурсів комп’ютера.

# ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. А.Зандстра, Ф.Матт. PHP Objects, Patterns, and Practice, 2013. Режим доступу: <https://www.apress.com/la/book/9781430260325>
2. Дмитрий Кирсанов. “Веб-дизайн” [електронний ресурс] - 2006. Режим доступу: http://web-diz.com.ua/skachat/veb-dizayn--dmitriy-kirsanov-skachat-besplatno/
3. Тим Кедлек. “Адаптивный дизайн. Делаем сайты для любых устройств” [електронний ресурс] - 2013. Режим доступу: <https://www.ozon.ru/context/detail/id/21098804/>
4. Документація бібліотеки JQuery [Електронний ресурс] // JQuery incorp. – 2016. – Режим доступ: <https://jquery-docs.ru>
5. Выразительный JavaScript [електронний ресурс] - 2017. Режим доступу: https://legacy.gitbook.com/book/karmazzin/eloquentjavascript\_ru/details
6. Основы наследования [електронний ресурс] - 2015. Режим доступу: [https://professorweb.ru/my/ level7/7\_1.php](https://professorweb.ru/my/csharp/charp_theory/level7/7_1.php)
7. Плюси і мінуси обєктно-орієнтованого програмування [електронний ресурс] - 2015. Режим доступу: <http://damp.biz/plyusi-i-minusi-obyektno-oriyentovanogo-programuvannya/>
8. SQL [електронний ресурс] - 2016. Режим доступу: <https://uk.wikibooks.org/wiki/SQL>
9. Создание SQL запросов [електронний ресурс] - 2016. Режим доступу: <https://www.politerm.com/zuludoc/spatial_query_sql.htm#_sql_overview>
10. Все операции с БД в графическом приложении [Електронний ресурс] Режим доступу до ресурсу: https://metanit.com/sharp/adonet/3.5.php

# ДОДАТКИ

Додаток А

**goods.php**

<?php

namespace app\model\goods;

use mysqli;

class Goods\_class{

     public function goods\_mass()

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `goods`");

          $mysql->close();

          return $result;

     }

     public function goods\_all\_count()

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT count(\*) FROM `goods`");

          $mysql->close();

          return $result;

     }

     public function goods\_serch\_by\_name($name)

     {

          $name = '%'.$name.'%';

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `goods` WHERE `name` like '$name'");

          $mysql->close();

          return $result;

     }

     public function goods\_serch\_by\_id($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `goods` WHERE `id\_goods` like '$id'");

          $mysql->close();

          return $result;

     }

     public function goods\_searh\_name\_count($name){

          $name = '%'.$name.'%';

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT count(\*) FROM `goods` WHERE `name` like '$name'");

          $mysql->close();

          return $result;

     }

     public function comments\_array\_by\_id($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `comments\_to\_goods` WHERE `id\_goods` like '$id'");

          $mysql->close();

          return $result;

     }

     public function user\_by\_id($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `users` WHERE `id\_user` like '$id'");

          $mysql->close();

          return $result;

     }

     public function set\_comment($rating, $content, $id\_user, $id\_goods, $user\_name){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query(

          "INSERT INTO `comments\_to\_goods`(`id\_user`,`id\_goods`, `user\_name`, `rating`, `comment`)

          VALUES('$id\_user','$id\_goods', '$user\_name', '$rating', '$content');");

          $mysql->close();

     }

     public function add\_to\_baket($id\_goods, $id\_user, $count){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("INSERT INTO `basket` (`id\_basket`, `id\_user`, `id\_goods`, `count`) VALUES (NULL, '$id\_user', '$id\_goods', '$count');");

          $mysql->close();

     }

     public function rating\_by\_id($id)

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `comments\_to\_goods` WHERE `id\_goods` like '$id'");

          $fetch = mysqli\_fetch\_assoc($result);

          $rating\_sum=0;

          $i = 0;

          if($fetch!=false){

               for($i = 0; $fetch!=false; $i++){

                    $rating\_sum += $fetch['rating'];

                    $fetch = mysqli\_fetch\_assoc($result);

               }

              $goods\_rating = $rating\_sum/$i;

          }else $goods\_rating = 0;

          return $goods\_rating;

     }

     public function search\_by\_material($material){

               $material = '%'.$material.'%';

               $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

               $result = $mysql->query("SELECT \* FROM `goods` WHERE `material` like '$material'");

               $mysql->close();

               return $result;

          }

     public function search\_by\_material\_and\_type($material, $type){

          $material = '%'.$material.'%';

          $type = '%'.$type.'%';

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `goods` WHERE `material` like '$material' and `type` like '$type'");

          $mysql->close();

          return $result;

          }

     public function user\_orders($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `basket` WHERE `id\_user` like '$id'");

          $mysql->close();

          return $result;

     }

     public function trader\_orders($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("

          SELECT \* FROM `basket`

          INNER JOIN `goods` ON `goods`.`id\_goods` = `basket`.`id\_goods`

          WHERE `id\_store` = '$id'");

          $mysql->close();

          return $result;

     }

     public function basket\_delete($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("DELETE FROM `basket` WHERE `id\_basket` = '$id'");

          $mysql->close();

     }

     public function user\_comments($id\_user){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `comments\_to\_goods` WHERE `id\_user` like '$id\_user'");

          $mysql->close();

          return $result;

     }

     public function trader\_comments($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("

          SELECT \* FROM `comments\_to\_goods`

          INNER JOIN `goods` ON `goods`.`id\_goods` = `comments\_to\_goods`.`id\_goods`

          WHERE `id\_store` = '$id'");

          $mysql->close();

          return $result;

     }

     public function comment\_delete($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("DELETE FROM `comments\_to\_goods` WHERE `id\_comment` = '$id'");

          $mysql->close();

     }

     public function set\_data\_to\_server($name, $cost, $size, $madeby, $material, $type, $description, $id\_store, $img1, $img2, $img3){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("INSERT INTO `goods` (`name`, `cost`, `size`, `made\_by`, `material`, `type`, `description`, `id\_store`, `img1`, `img2`, `img3`) VALUES ('$name', '$cost', '$size', '$madeby', '$material','$type', '$description', '$id\_store', '$img1', '$img2', '$img3');");

          $mysql->close();

          //INSERT INTO `goods` (`id\_goods`, `name`, `description`, `cost`, `material`, `type`, `size`, `made\_by`, `img1`, `img2`, `img3`, `id\_store`) VALUES (NULL, '1', '1', '1', '1', '1', '1', '1', NULL, NULL, NULL, '1');

     }

     public function store\_by\_user\_id($id\_user){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `stores` WHERE `id\_user` like '$id\_user'");

          $mysql->close();

          return $result;

     }

     public function trader\_goods\_list($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("

          SELECT \* FROM `stores`

          INNER JOIN `goods` ON `goods`.`id\_store` = `stores`.`id\_store`

          WHERE `stores`.`id\_user` = '$id'");

          $mysql->close();

          return $result;

     }

     public function goods\_delete($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("DELETE FROM `goods` WHERE `goods`.`id\_goods` = '$id'");

          $mysql->close();

     }

     public function print\_all\_users(){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `users` WHERE `trader` = '0'");

          $mysql->close();

          return $result;

     }

     public function print\_all\_traders(){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT \* FROM `users`

          INNER JOIN `stores` ON `stores`.`id\_user` = `users`.`id\_user`

          WHERE `users`.`trader` = 1");

          $mysql->close();

          return $result;

     }

     public function user\_delete($id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("DELETE FROM `users` WHERE `id\_user` = '$id'");

          $mysql->close();

     }

     public function store\_delete($id\_store){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("DELETE FROM `stores` WHERE `id\_store` = '$id\_store'");

          $mysql->close();

     }

     public function trader\_delete($id\_user){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("UPDATE `users` SET `trader` = '0' WHERE `id\_user` = '$id\_user'");

          $mysql->close();

     }

     public function search\_by\_name\_and\_descrip($name, $description){

          $name = '%'.$name.'%';

          $description = '%'.$description.'%';

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT\*FROM `goods` WHERE `name` like '$name' and `description` like '$description'");

          $mysql->close();

          return $result;

     }

     public function update\_data($name, $cost, $size, $madeby, $material, $type, $description, $id\_store, $id\_goods){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("UPDATE `goods`  SET `name` = '$name', `cost` = '$cost', `size` = '$size', `made\_by` = '$madeby', `material` = '$material', `type` = '$type', `description` = '$description', `id\_store` = '$id\_store' WHERE `id\_goods` = '$id\_goods'");

          $mysql->close();

     }

     public function img\_upload($img, $id\_goods, $img\_num){

          $img\_is = 'img'.$img\_num;

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("UPDATE `goods` SET `$img\_is` = '$img' WHERE `id\_goods` = '$id\_goods'");

          $mysql->close();

     }

}

checks.php

<?php

namespace app\model\check;

use mysqli;

class Checks{

     public function user\_authorize($pass, $email){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT\*FROM `users` WHERE `email` = '$email' AND `password` = '$pass'");

          $user = $result->fetch\_assoc();

          $mysql->close();

          return $user['id\_user'];

     }

     public function user\_registration($login, $email, $name, $pass){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query(

          "INSERT INTO `users`(`login`,`email`, `password`, `name`, `trader`)

          VALUES('$login','$email', '$pass', '$name', '0')");

          $mysql->close();

     }

     public function this\_user\_trader($id)

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT\*FROM `users` WHERE `id\_user` = '$id'");

          $result\_fetch = $result->fetch\_assoc();

          if($result\_fetch['trader']==1){

               return true;

          }

          else{

               return false;

          }

          $mysql->close();

     }

     public function this\_user\_admin($id)

     {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $result = $mysql->query("SELECT\*FROM `users` WHERE `id\_user` = '$id'");

          $result\_fetch = $result->fetch\_assoc();

          if($result\_fetch['admin']==1){

               return true;

          }

          else{

               return false;

          }

          $mysql->close();

     }

     public function store\_registration($name, $email, $adress, $number, $erdpou, $id){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("INSERT INTO `stores`(`name`,`email`, `adress`, `phone\_number`, `erdpou`, `id\_user`)VALUES('$name','$email', '$adress', '$number', '$erdpou', '$id')");

          $mysql->query("UPDATE`users` SET `trader` = '1' WHERE id\_user like '$id'");

          $mysql->close();

     }

     public function account\_update($id ,$name, $email, $adress, $number, $login) {

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("UPDATE `users`  SET `name` = '$name', `email` = '$email', `adress` = '$adress', `phone\_number` = '$number', `login` = '$login'  WHERE `id\_user` = '$id'");

          $mysql->close();

     }

     public function password\_update($id, $password){

          $mysql = new mysqli('localhost', 'root', 'root', 'courseprojectbd');

          $mysql->query("UPDATE `users` SET `password` = '$password' WHERE `id\_user` = '$id';");

          $mysql->close();

     }

     function createDatabaseBackup(){

    $backupDir = ROOT.'/';

    date\_default\_timezone\_set('Europe/Kiev');

    $currentTime = time();

    $backupTime = strtotime(date('Y-m-d 23:00:00'));

    if ($currentTime > $backupTime) {

        $filename = 'backup\_' . date('Y-m-d') . '.sql';

        $filePath = $backupDir . $filename;

        if (!file\_exists($filePath)) {

            $command = "mysqldump --user=root --password=root --host=localhost coursework > {$filePath}";

            exec($command);

        }

    }

}

}