**Форма № Н-6.01**

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

факультет: Автоматики і комп'ютеризованих технологій

Кафедра: Системотехніки

**КУРСОВА РОБОТА**

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

Тема: Розробка елементів інформаційної системи «Інтернет-магазин» на прикладі

інтернет-магазину роздрібної торгівлі електронними пристроями

(Тема роботи)

з дисципліни: Системи управління базами даних

(Назва дисципліни)

Керівник: доцент Колесник Л.В.

(Підпис, дата, посада, прізвище, ініціали)

Студент: гр. АКТСIу-17-2 Черкашин В.А.

(Група, підпис, дата, прізвище, ініціали)

Робота захищена з оцінкою «\_\_\_\_\_\_\_\_\_»

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р

Комісія:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Підпис, посада, прізвище, ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Підпис, посада, прізвище, ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Підпис, посада, прізвище, ініціали)

Харків 2018

Харківський національний університет радіоелектроніки

Факультет: Автоматики і комп'ютеризованих технологій

Кафедра: Системотехніки

Дисципліна: Система управління базами даних

спеціальність: Системна інженерія

Курс: 1 група: АКТСIу-17-2 семестр: 2

**ЗАВДАННЯ**

на курсову роботу

Студенту: Черкашину Володимиру Андрійовичу

(прізвище ім'я по батькові)

1. Тема роботи: Розробка елементів інформаційної системи «Інтернет магазин»

2. Термін здачі студентом закінченої роботи: 20.06.18

3. Вхідні дані до проекту: функція: розробка інформаційного забезпечення «Інтернет магазин» на прикладі інтернет-магазину електронних пристроїв; Перелік використаних програмних засобів: ОС Microsoft Windows 7 і вище, PHP 7.2, MySQL Workbench, MySQL Server; Технічне забезпечення: IBM-сумісний ПК з МП Pentium II і вище.

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): 4.1 Вступ. 4.2 Аналіз предметної області. 4.3 Проектування бази даних. 4.3.1 ER – моделювання. 4.3.1.1 Опис сутностей. 4.3.1.2 Опис зв'язків. 4.3.2 Розробка фізичної схеми бази даних. 4.3.3 Нормалізація бази даних. 4.4 Вибір СУБД. 4.5 Висновки. 4.6 Перелік посилань.

5. Дата видачі завдання: 17.02.18

Керівник роботи: доцент Колесник Л.В.

(Підпис) (прізвище ім'я по батькові)

студент: Черкашин В.А.

(Підпис) (прізвище ім'я по батькові)

КАЛЕНДАРНИЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| №  з / п | Назва етапів курсової роботи | Термін виконання етапів роботи | Примітка |
| 1 | Отримання завдання на курсову роботу | 28.02.18 |  |
| 2 | Аналіз предметної області | 08.03.18 |  |
| 3 | Проектування бази даних | 10.03.18 |  |
| 4 | ER - моделювання | 24.03.18 |  |
| 5 | Опис сутностей | 31.03.18 |  |
| 6 | Опис зв'язків | 07.04.18 |  |
| 7 | Розробка фізичної схеми бази даних | 21.04.18 |  |
| 8 | Вибір СУБД | 05.05.18 |  |
| 9 | Вибір конкретної СУБД | 19.05.18 |  |
| 10 | Розробка елементів програмного забезпечення | 26.05.18 |  |
| 11 | Розробка «Інструкції» | 09.06.18 |  |
| 12 | Оформлення пояснювальної записки і графічних матеріалів | 16.06.18 |  |
| 13 | Захист курсової роботи | 20.06.18 |  |

Керівник роботи: доцент Колесник Л.В.

(Підпис) (прізвище ім'я по батькові)

студент: Черкашин В.А.

(Підпис) (прізвище ім'я по батькові)

«\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_\_ р.

Реферат

Пояснювальна записка до курсової роботи: 56 с., 20 рис., 3 табл., 2 додатки, 5 диаграм.

Об'єкт дослідження – база даних «Інтернет магазин» магазин» електронних пристроїв.

Мета роботи – розробка бази даних «Інтернет магазин» електронних пристроїв, розробка програмного засобу для управління базою даних.

Методи дослідження – вивчення літератури, пов'язаної із завданням, розробка ER – моделі та фізичної схеми бази даних; розробка програмного забезпечення та написання програми.

Результати курсової роботи – система управління базою даних «Інтернет магазин» у вигляді прикладної програми.

Область даних – інформація інтернет магазин роздрібної торгівлі електронними пристроями, зокрема мобільними телефонами та планшетами, також інформація про продаж різних гаджетів.

ЗМІСТ

ВСТУП. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.1 ER - моделювання. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.1.1 Опис сутностей. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.1.2 Опис зв'язків. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . . . . . . . . . 9

2.2 Розробка фізичної схеми бази даних. . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.3 Нормалізація бази даних . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3 ВИБІР СУБД ТА МОВИ ПРОГРАМУВАННЯ. . . . . . . . . . . . . . . . . . . . . . . . 15

ВИСНОВКИ. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

ПЕРЕЛІК ПОСИЛАНЬ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .20

Додаток А. Керівництва користувача. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

Додаток Б. Текст програми. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

ВСТУП

База даних  – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки. Основним засобом реалізації централізованого управління даними, збереженими в базі, доступу до них і підтримання їх в стані, відповідному стану предметної області, стали системи управління базами даних (СУБД). Технологія управління базами даних сформувалася на основі досвіду розробки і застосування сотень типів СУБД, багатьох мов визначення даних і маніпулювання даними. Одночасно створювалася теорія баз даних, в якій були розвинені такі розділи, як теорія моделей даних і методів їх еквівалентних перетворень, теорія функціональної залежності в базах даних, теорія баз даних з неповною інформацією, методи забезпечення цілісності баз даних в умовах потоку конкурентних запитів,

Для зберігання даних про інтернет магазині під час виконання курсового проекту була використана реляційна модель БД, розроблена в GUI для MySQL - phpMyAdmin.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Збільшення обсягу та структурної складності збережених даних, розширення кола користувачів інформаційних систем призвело до виникнення необхідності створення зручних засобів інтеграції даних, що зберігаються, а також управління ними.

У якості предметної області в роботі була розглянута інформація про різні інтернет магазина, а також інформація про продаж гаджетів. В процесі виконання роботи була розроблена база даних, яка вирішує наступні завдання:

– зберігання інформації о продуктах;

– зберігання інформації про користувачів з поділом на покупців і менеджера, який обробляє замовлення;

– зберігання інформації о поточних замовленнях;

– зберігання інформації про історію замовлень;

– функція для розрахунку рейтингу товарів за кількістю куплених продуктів;

– процедура для виведення поточної версії MySQL і угруповання деяких запитів;

– тригери на хешування пароля покупця і менеджера, а також контролю за клькістю товарів на складі.

Для вирішення цих завдань було проаналізовано предметна область, складена і нормалізована БД.

2 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

2.1 ER–моделювання

Робота з базою даних починається з побудови моделі. Найбільш поширеною є ER–модель – модель "Сутність–зв'язок".

Моделювання даних – це перший крок на шляху проектування БД, це перехід від об'єктів реального світу до комп'ютерної моделі БД.

ER–модель служить для об'єднання різних уявлень даних на концептуальному рівні. На основі ER–моделі будуються ER–діаграми, на яких відображаються три основних компоненти ER–моделі: сутність, атрибути, зв'язки.

2.1.1 Вибір та опис сутностей

У спроектованій базі даних можна виділити наступні сутності та атрибути, які представлені у таблиці 2.1.

Таблиця 2.1 - Опис сутностей

| Сутність | Атрибути |
| --- | --- |
| 1 | 2 |
| Users | Login (PK)  Password  Email  isManager  Name  LastName  MiddleName |

Продовження таблиці 2.1

|  |  |
| --- | --- |
| 1 | 2 |
| Categories | Id\_Cat (PK)  title |
| Orders | Id (PK)  TimeOrder  quntity  login (FK)  Product (FK)  Addr (Fk) |
| logOrders | Id (PK)  TimeOrder  quntity  login (FK)  Product (FK)  Price (FK)  Addr (FK) |
| Card | Id (PK)  Cart  Login (FK) |

2.1.2 Опис зв'язків

Сутності «Products» і «Categories» з'єднані зв'язком М:1. До однієї категорії може належати багато продуктів, але конкретний продукт належить тільки до однієї категорії.

Сутності «Users» і «card» з'єднані зв'язком 1: М. У одного користувача може бути багато карток (різних або одного і того ж банку), але конкретна карта належить, тільки одному користувачеві.

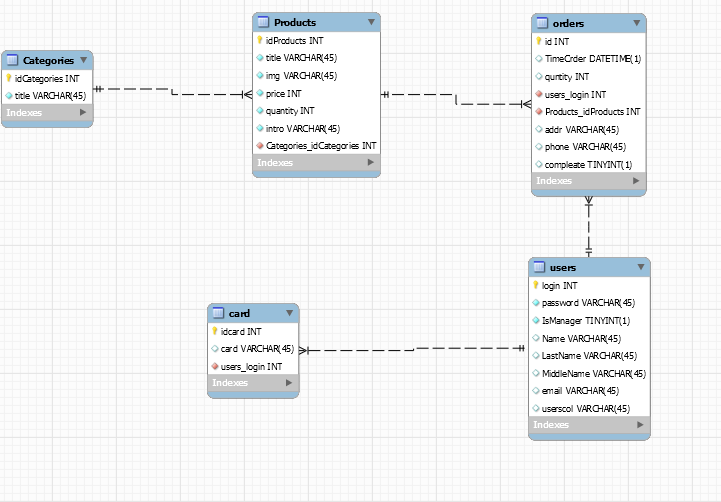
Сутності «Users» і «orders» з'єднані зв'язком 1: М. У одного користувача може бути багато замовлень, але конкретне замовлення належить тільки одному користувачеві.

Сутності «Orders» і «products» з'єднані зв'язком М: 1. Один продукт можуть замовляти багато разів, але одне замовлення єдиноразово оформлюється тільки на один продукт. Опис зв'язків наведено у Таблиці 2.2.

Таблиця 2.2 - Опис зв'язків

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Сутність 1 | Атрибут 1 | Наіменування | Крат-ність | Сутність 2 | Атрибут 2 |
| Products | idProducts | формують | М: 1 | Categories | Categories\_idCategories |
| Users | login | формують | 1: М | cart | Users\_login |
| Users | login | формують | 1: М | orders | Users\_login |
| Products | idProducts | формують | 1: М | orders | Product\_idProducts |

На основі таблиці 2.2 була спроектована ER – модель. Вона представлена на малюнку 2.1.



Малюнок 2.1 - ER-модель

2.2 Розробка фізичної моделі бази даних

В ході виконання курсового проекту для реалізації була обрана реляційна модель на основі моделі «сутність-зв'язок». Тобто, така модель даних, де текстова чи числова інформація зображається за допомогою таблиць.

Перевагами реляційної моделі є те, що вона спрямована на опрацювання документа в цілому, а не окремих його частин. Вона не потребує знань програмування від користувачів. Для використання реляційної моделі достатньо навичок роботи на ПК та знань основ інформаційних технологій. Тому реляційна модель є простою у розумінні та використанні.

Недоліками реляційної моделі є відносно низька швидкість доступу до даних та використання великого обсягу пам’яті на носіях інформації.

Реляційна модель спроектованої БД складається з сутностей, представлених в таблиці 2.3.

Таблиця 2.3 - Опис сутностей бази даних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця | Атрибут | Тип атрибута | Ключове поле | NULL / not NULL |
| 1 | 2 | 3 | 4 | 5 |
| Categories | idCategories | INT | + | NOT NULL |
| title | VARCHAR (45) | - | NULL |
| Products | idProducts | INT | + | NOT NULL |
|  | title | VARCHAR (45) | - | NULL |
| img | VARCHAR (45) | - | NULL |
| price | INT | - | NULL |
| quantity | INT | - | NULL |
| intro | TEXT | - | NULL |
| Categories\_IdCategories | INT | + (FK) | NULL |
| Users | Login | VARCHAR (45) | + | NOT NULL |
| password | VARCHAR (45) | - | NOT NULL |
| email | VARCHAR (45) | - | NOT NULL |
| IsManager | TINYINT (1) | - | NULL |
| Name | VARCHAR (45) | - | NULL |
| LastName | VARCHAR (45) | - | NULL |
| MiddleName | VARCHAR (45) | - | NULL |
| card | idCart | INT | + | NOT NULL |
| cart | VARCHAR (45) | - | NULL |
| Users\_login | VARCHAR (45) | + (FK) | NOT NULL |
| orders | id | VARCHAR (45) | + | NOT NULL |
| TimeOrder | DATATIME () | - | NULL |
| quantity | INT | - | NULL |

Продовження таблиці 2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблиця | Атрибут | Тип атрибута | Ключове поле | NULL / not NULL |
| 1 | 2 | 3 | 4 | 5 |
|  | Users\_login | VARCHAR (45) | + (FK) | NOT NULL |
| Products\_idProducts | INT | + (FK) | NOT NULL |
| Addr | VARCHAR (45) | - | NULL |
| phone | VARCHAR (10) | - | NULL |
| compleate | TINYINT (1) | - | NULL |

2.3 Нормалізація

Нормалізація – це розбиття таблиці на дві або більше, що володіють кращими властивостями включення, зміну або видалення даних, остаточна мета нормалізації зводиться до отримання такого проекту БД в якому кожен факт з'являється лише в одному місці, тобто виключена надмірність інформації.

Приведення таблиць до 1ї нормальної форми: таблиці, представлені в схемі бази даних, приведені до 1ї нормальної форми, тому в них визначені всі ключові атрибути, все значення атомарні, всі атрибути залежать від первинного ключа.

Приведення таблиць до 2ї нормальної форми: таблиці, представлені в схемі бази даних, наведені до 2ї нормальної форми, тому вони приведені до 1й нормальній формі і в них відсутні часткові залежності.

Приведення таблиць до 3ї нормальної форми: таблиці, представлені в схемі бази даних, приведені до 3ї нормальної форми, тому вони наведені до 2ї нормальної форми і відсутні транзитивні залежності. На діаграмах, приведених нижче, зображені діаграми залежностей для кожної із сутностей.

|  |  |
| --- | --- |
| idCategories | title |

Діаграма 2.1 - Діаграма залежностей для сутності «Categories»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| IdProducts | title | img | price | quantity | intro | idCat |

Діаграма 2.2 - Діаграма залежностей для сутності «Products»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| login | password | email | IsManager | Name | LastName | MName |

Діаграма 2.3 - Діаграма залежностей для сутності «Users»

|  |  |  |
| --- | --- | --- |
| IdCart | cart | Users\_login |

Діаграма 2.4 - Діаграма залежностей для сутності «Products»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | TimeOrder | quantity | UserLogin | PoroductId | IdAddr |

Діаграма 2.5 - Діаграма залежностей для сутності «Users»

Таким чином, база даних була приведена до 3ї нормальної форми

3 ВИБІР СУБД ТА МОВИ ПРОГРАМУВАННЯ

Вибір системи управління бази даних є одним з важливих етапів при розробці додатків баз даних. Обраний програмний продукт повинен задовольняти не тільки поточним, а й майбутнім потребам підприємства, при цьому треба враховувати фінансові витрати на придбання необхідного обладнання, програмного забезпечення, розробку необхідного програмного забезпечення на її основі, а також навчання персоналу. Крім того, необхідно переконатися, що нова СУБД здатна принести підприємству реальну вигоду.

В даному розділі за результатами аналізу доступних джерел робиться спроба сформулювати критерії при виборі СУБД, наводиться класифікація вимог / критеріїв. Найбільш простий підхід при виборі СУБД заснований на оцінці того, якою мірою існуючі системи задовольняють основним вимогам створюваного проекту інформаційної системи. Більш дорогим і складним варіантом є створення випробувального проекту на основі кількох СУБД і наступний вибір найбільш підходящої. Але і в цьому випадку необхідно обмежити коло можливих систем, спираючись на якісь критерії відбору. Вимоги до СУБД, використовуваних при аналізі тієї чи іншої інформаційної системи, може змінюватися в залежності від конкретної задачі.

Можна виділити кілька груп критеріїв:

* + - 1. моделювання даних;
      2. особливості архітектури програмного продукту;
      3. особливості розробки програми;
      4. продуктивність;
      5. надійність;
      6. вимога до ресурсів ПК.

Були проаналізовані існуючі БД такі, як: MySQL Server, MySQL Workbench. Беручи до уваги вимоги до СУБД, зупинимо свій вибір на MySQL Workbench.

В якості мови програмування була вибрана PHP, а також такі технолгіх як: html, css, js, jqury. Цей стек технологій є стандартним для написання сайтів. PHP є однією з найпоширеніших мов, що використовуються у сфері веб–розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг–провайдерів. PHP — проект відкритого програмного забезпечення. PHP інтерпретується веб–сервером у HTML–код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP–коду, бо браузер отримує готовий html–код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript–кодів, які виконуються вже на стороні клієнта.

PHP — мова, у код якої можна вбудовувати безпосередньо html–код сторінок, які, у свою чергу, коректно оброблюватимуться PHP–інтерпретатором. Обробник PHP просто починає виконувати код після відкриваючого тегу (<?php) і продовжує виконання до того моменту, поки не зустріне закриваючий тег (?>).

Велика різноманітність функцій PHP дає можливість уникати написання багаторядкових функцій, призначених для користувача, як це відбувається в C або Pascal.

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з С, Perl. Код РНР дуже схожий на той, який зустрічається в типових програмах мовами С або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP — мова, що поєднує переваги Perl та С і спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед web–програмістів, що в наш час є найпопулярнішою мовою для створення веб–застосунків (скриптів). Наведемо деякі переваги PHP, що призвели до вибору саме цієї мови при написанні програмного продукту.

Мова РНР була створена спеціально для генерації динамічних сторінок, тому сценарії на PHP вирішують цю задачу набагато простіше і швидше, ніж сценарії в інших програмуваннях. З іншого боку PHP представляє собою розширення HTML. CSS і JavaScript істотно розширили можливості HTML, проте PHP тільки дозволив створити повноцінні сайти на базі стандарту HTML.

Сценарії в мові PHP можуть бути виконані під всі основні операційні системи, в тому числі під Microsoft Windows і Unix-подібними операційними системами (Linux, OpenBSD, Solaris, HP-UX), Mac RISC OS, OS X і деякіми іншими. Большинство современных веб-серверов поддерживают PHP:

Для обробки тексту PHP підтримує використання регулярних виразів Perl, POSIX Extended і розбір XML-документів на основі стандартів DOM, SAX та розширення XSLT.

Для організації електронної комерції на сайті з допомогою PHP можуть використовуватися функції PHP CCVS, CyberMUT, Cybercash, Payflow Pro, VeriSign. Пошукова форма на сайті може бути організована за допомогою функції mnoGoSearch. PHP також підтримує функції календарей, переведення та архівування. PHP дозволяє взаємодіяти з сторонніми службами через протокол HTTP, POP3, COM (Windows), SNMP, NNTP, IMAP, LDAP і багато інших.

Одна з найсильніших переваг PHP перед іншими програмами, що використовуються для розробки WEB-приложений, – підтримка баз даних різних різних типів.

При цьому для PHP характерний простота генерації сторінки, що працює з базою даних. PHP підтримує роботу з наступними базами даних: MySQL, ODBC, dBase, Oracle, PostgreSQL, Adabas D, InterBase, Solid, Sybase, Informix, Hyperwave, Empress, Ingres, FrontBase, Ovrimos, Direct MS-SQL, Velocis, FilePro, mSQL, IBM DB2, Unix dbm. Язык сценариев PHP підтримує стандарт ODBC (Open Database Connection), який дозволяє працювати з будь-якою базою даних, яка підтримує цей міжнародний стандарт. А абстрактне розширення DBX дозволяє працювати з будь-якою базою даних, яка підтримується розширенням DBX.

За допомогою можливостей мови запитів SQL також були реалізовані наступні засоби:

* вибор та сортування даних об існуючих категоріях товару:

'SELECT \* FROM categories ORDER BY title;

* вибор товарі по заданій категорії:

SELECT \* FROM products Where CatName LIKE 'Телефоны';

* визов функції rate та round мови SQL:

SELECT round(rate('{$product['title']}'),2) AS `rate`;

* вибир продуктів по id:

SELECT \* FROM products WHERE id IN ( {$id\_sql} );

* видалення з таблиці:

DELETE FROM `orders` WHERE `orders`.`id` ='{$\_POST['manager']}';

* вибір за таблиці:

SELECT \* FROM `orders`

SELECT \* FROM products WHERE id IN ( {$id\_sql} );

* вибір кілкості товарів по id:

SELECT `quantity` FROM `products` WHERE `title`='{$product['title']}';

* вставка данних у таблицю:

INSERT INTO `orders` (`id`, `user`, `title`, `DateTime`, `price`,`quantity`);

VALUES (NULL, '{$\_SESSION['login']}', '{$product['title']}', NOW(), '{$product['price']}', '{$data[$i]}')

* вибір id з таблиці:

SELECT id FROM users WHERE login='$login';

* вставка данних у таблицю:

INSERT INTO users (login,password, email) VALUES('$login','$password', '$email');

* пошук like у таблиці продуктів:

"SELECT \* FROM products Where title LIKE '%$search%';

* вибор по категорії:

SELECT \* FROM products Where CatName='{$\_SESSION['catname'];

* оновлення даних:

UPDATE `userinfo` SET `Name`= '{$data['Name']}', `LastName`='{$data['LastName']}', `MiddleName`= '{$data['MiddleName']}', `cart`= '{$data['cart']}', `phone`= '{$data['phone']}', `addr`= '{$data['addr']}' WHERE `id`=(SELECT `idInfo` from `users` WHERE `login` = '{$\_SESSION['login']}';

* функція рейтингу

CREATE DEFINER=`root`@`localhost` FUNCTION `rate`(`r` VARCHAR(25)) RETURNS FLOAT NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER BEGIN

DECLARE rate FLOAT DEFAULT 0; SELECT (SUM(quantity)/(SELECT SUM(quantity) FROM logorders))\*100 into rate

FROM logorders WHERE title=r; RETURN rate;

END;

* триггер на хешування:

CREATE DEFINER=`root`@`localhost` TRIGGER `PasToMd5` BEFORE INSERT ON `users` FOR EACH ROW SET NEW.password = md5;

* триггер на кошик:

CREATE DEFINER=`root`@`localhost` TRIGGER `log` AFTER INSERT ON `orders` FOR EACH ROW BEGIN DECLARE var1 int(11) DEFAULT 0; SELECT quantity into var1 FROM products WHERE title=NEW.title;

IF var1>=NEW.quantity THEN INSERT INTO logorders SELECT \* FROM orders WHERE id=NEW.id;

UPDATE products SET products.quantity=products.quantity-NEW.quantity WHERE title=NEW.title; END IF; END

* процедура на визначення версії:

CREATE DEFINER=`root`@`localhost` PROCEDURE `myversion`() NOT DETERMINISTIC CONTAINS SQL SQL SECURITY DEFINER

* BEGIN SELECT VERSION(); END.

ВИСНОВКИ

У процесі виконання курсової роботи було розроблено програмний засіб для предметної області «Інтернет магазин Гаджетів».

Була спроектована база даних для обраної предметної області, яка була нормалізована до 3ї форми, що виключає появу аномалій даних. База даних містить 8 таблиць. У розробленому програмному засобі реалізовані наступні можливості:

– додавання в базу даних;

– видалення з бази даних;

– можливість оновлювати данні;

– можливість скласти рейтинг товарів;

– можливість робити вибірку с БД та виконувати сортування за критеріем.

Також були реализовані такі запити, як:

– запити, що містять некоррелірований підзапит;

– збережена функція для розрахунку рейтингу товарів;

– запити insert, delete, update, select, а також запити із сортуванням order by;

* запити із пошуком за критерієм;
* тригер на хешування пароля;
* тригер на заповнення (insert) в logorders;
* тригер на видалення товарів при заказі.

Даний засіб може використовуватися будь–якою організацією або приватною особою для продажу своїх товарів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Замулин А. В. Системи програмування баз даних і знань / А. В. Замулин. – Новосибірськ: Наука. Сиб. Від., 2002. – 352 с.
2. Мартін Дж. Організація баз даних в обчислювальних системах. / Дж. Мартін. – М .: Світ, 1980. – 662 с.
3. Романов Б. Л. dBase IV. Призначення, функції, застосування. / А. С. Кушніренко, Б. Л. Романов. – М .: Радіо і баз даних, 1991. – 388 с.
4. Ульман Дж. Основи систем баз даних. / Дж. Ульман. – М .: Фінанси і статистика, 1983. – 331 с.
5. MSSQL http://ru.wikipedia.org/wiki/MSSQL
6. MySQL http://ru.wikipedia.org/wiki/MYSQL

|  |
| --- |
| Міністерство освіти і науки України  затверджую  Керівник курсової роботи,  доц. кафедри системотехніки  \_\_\_\_\_\_\_\_\_\_\_\_ Л.В. Колесник  (Підпис, дата)  ІНФОРМАЦІЙНА СИСТЕМА «інтернет магізн»  Інструкція користувача  ЛИСТ затвердження  ГЮІК.508100.111 ІЗ – ЛУ  Студент групи \_\_\_\_\_\_\_\_\_\_\_\_  (назва групи)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Підпис, дата, прізвище, ім'я, по батькові)  2018 |

|  |
| --- |
| Міністерство науки і освіти України  ЗАТВЕРДЖУЮ  ГЮІК.508100.406 ІЗ – ЛУ  ІНФОРМАЦІЙНА СИСТЕМА «Інтернет магазин»  Інструкція користувача  ГЮІК.508100.406 ІЗ – ЛУ  листів 11  2018 |

А. ВСТУП

Дане програмне забезпечення застосовується в області продажу товаров через інтернет, на прикладі продажу електронної техніки. У ньому реалізовано наступні необхідні для будь–якого інтернет магазину функції такі як: корзина покупця, реєстрація, авторизація, вибір категорій, оформлення замовлення, розрахунок рейтингу сбуту товарів, можливість змінювати свої дані. Для менеджера реалізовані наступні функції: перегляд історій замовлень, можливість виконання замовлення, перегляд даних про покупців.

Всі вищеописані функції виконуються динамічно, тобто не потрібне перезавантаження сторінки, Що забезпечується за допомогою технології AJAX. Всі форми валідуються як на стороні сервера, так і на стороні клієнта, для цього використовується мова PHP і його стандартні методи для валідації – на стороні сервера і JS – на стороні клієнта.

А.1 ПРИЗНАЧЕННЯ І УМОВИ ЗАСТОСУВАННЯ

Розроблена база даних «Інтернет магазин» призначена для зберігання інформаціх о товарах, користувачів і т.д. Даний програмний продукт створений для продажу товарів через інтернет. Для використання даного програмного продукту необхідна наявність комп'ютера з мінімальними системними вимогами, зазначеними в наступному розділі.

А.2 ПІДГОТОВКА ДО РОБОТИ

Необхідні ресурси для роботи програми:

– процесор Pentium IV;

– мінімальний обсяг оперативної пам'яті – 1 Гбайт;

– жорсткий диск об'ємом вільного простору не менше 1 Гбайт;

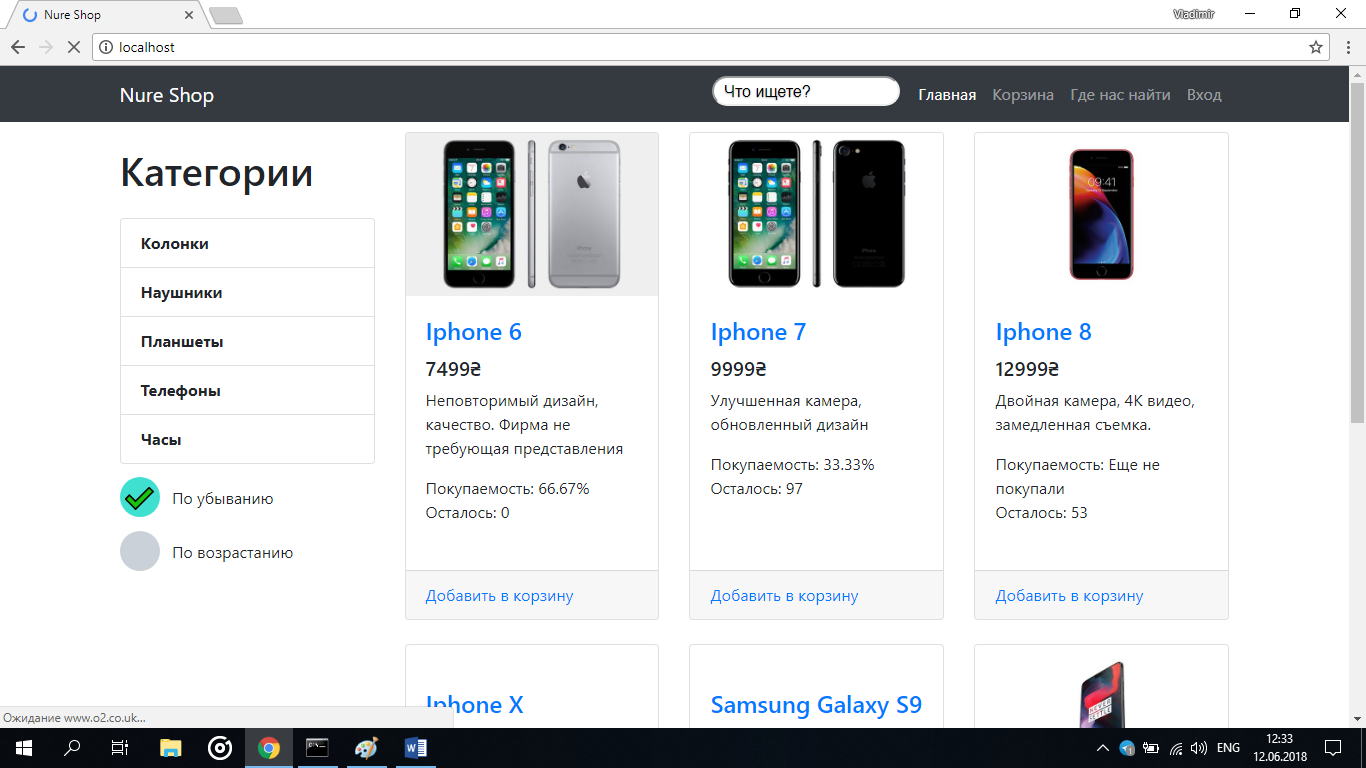
– рекомендується монітор типу VGA;

– клавіатура, миша;

– необхідно так само, щоб на ПК було встановлено ПО: ОС сімейства Windows, починаючи c Windows 7, Net Framework 4, MySQL.Connector.

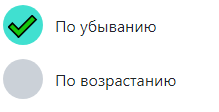
А.3 ОПИС ОПЕРАЦІЙ

При заході на сайт, з'являється екранна заставка запуску, який зображений на малюнку А.3.1.



Малюнок А.3.1 – Початковий екран програми.

При натисканні на кнопки сортування, відбувається сортування товарів, за ціною (малюнок А.3.2).



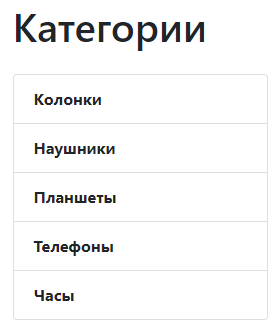
Малюнок А.3.2 – Кнопки сортування.

У полі зображеному на малюноку А.3.3 користувач може ввести назву товару, який він хоче знайти, якщо товар знайдений, то він виводиться у вікно продуктів, якщо не знайдений, то виведеться порожнє вікно, так як продукти не знайдені.



Малюнок А.3.3 – Input пошуку товарів по БД

При натисканні на відповідну категорію в вікні продуктів (малюнок А.3.4) виведеться продукти з цієї категорії.



Малюнок А.3.4 – Меню вибору товарів

При натисканні на «Главная» перейде на головну сторінку сайту, тобто на index.php. При натисканні на «Корзина» – перейде в кошик. При натисканні на «Где нас найти» – завантажиться Google maps c адресою нашого університету. При натисканні на «Вхід» перейде на форму входу / авторизації. Усі ці пункти зображені на малюнку А.3.5



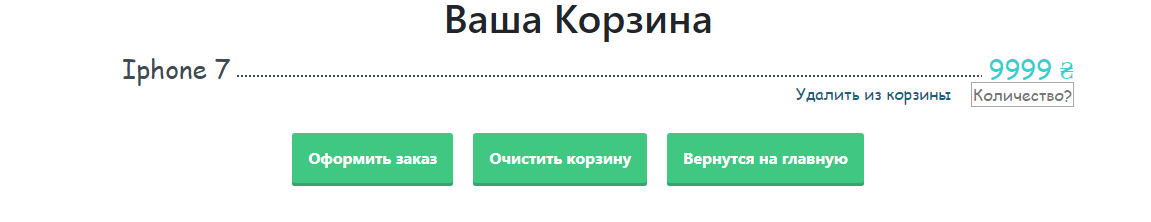
Малюнок А.3.5 – Меню сайту

Під кожним продуктом є кнопка додати в кошик, при натисканні на неї даний продукт додається в кошик. Приклад роботи із кошиком наведено на малюнках А.3.6 – А.3.8



Малюнок А.3.6 – Меню "Корзина".

При натисканні на кнопку видалити з кошика, продукт видаляється з кошика. Поле кількість – туди вводимо кількість товарів або залишаємо незаповненим, якщо вводиться кількість, якого немає на складі або від'ємне значення, то виводяться помилки (малюнки А.3.9 – А.3.11).



Малюнок А.3.7 – Вид кошика



Малюнок А.3.8 – Вид кошика



Малюнок А.3.9 – Введення від'ємних значень

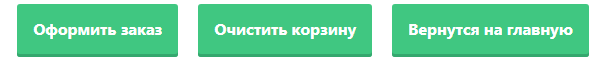


Малюнок А.3.10 – Введення кількості більшого, ніж на складі



Малюнок А.3.11 – Повідомлення при введенні правильних значеннь

При натисканні кнопки оформити замовлення (малюнок А.3.12) – оформляється замовлення, тобто в таблицю orders заносяться значення, через те, що на таблиці orders є тригер, ці значення заносяться в logorders.



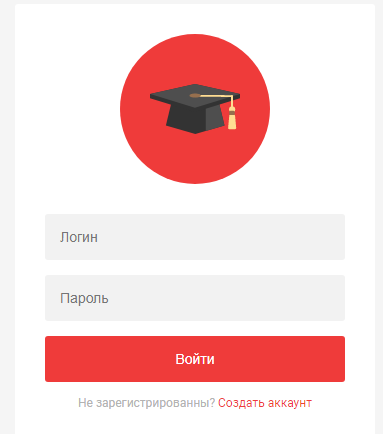
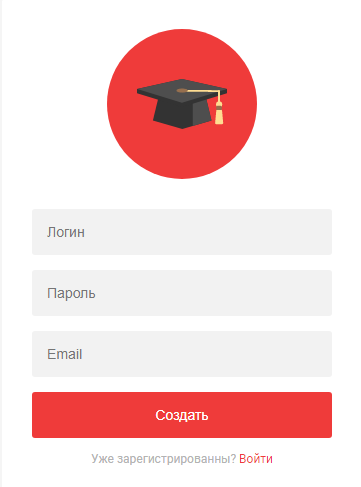
Малюнок А.3.12 – Меню кошика

У формі авторизації/входу користувач може зареєструватися або авторизуватися. Якщо користувач зарєестрований, то потрібно війти у систему, тому що якщо користувач не авторизованний, то тоді неможливо купити товар. Якщо користувач не зарєестрований, то необхидно зарєеструватися, а потім увійти у систему. Ці форми зображені на малюнках А.3.13 – А.3.14. При авторизації вводимо логін і пароль, який користувач вводив при реєстрації, якщо такий логін і пароль правильні і існують в БД, то при на натисканні на кнопку «Увійти» відбувається вхід і перехід на головну сторінку.

При реєстрації потрібно ввести бажаний логін, пароль і валідний email і натиснути на кнопку «створити», якщо все дані правильні, то користувач буде зареєстрований; щоб увійти, потрібно натиснути на кнопку нижче «Войти» (малюнок А.3.13).

При авторизації вводимо логін і пароль, який користувач вводив при реєстрації, якщо такий логін і пароль правильні і існують в БД, то при на натисканні на кнопку «Увійти» відбувається вхід і перехід на головну сторінку (малюнок А.3.14).

При успішній авторизації, користувач може здійснювати замовлення. І додається новий пункт меню Здравстуйте «Імя\_указанное\_при\_регістраціі» (малюнок А.3.15 ).

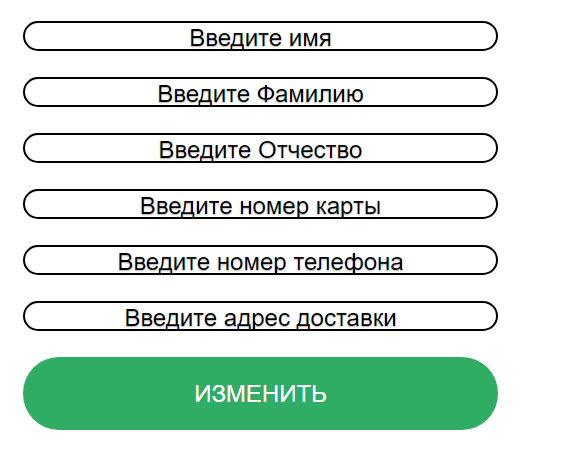
 

Малюнок А.3.13 – Форма авторизації та реєстрації



Малюнок А.3.14 – Меню сайту при успішної авторизації

При натисканні на кнопку Здрастуйте «Імя\_указанное\_при\_регістраціі», користувач переходить на вищенаведену форму. На цій формі він може за своїм бажанням залишити свої контактні дані. Однак ці дані можуть бути порожні, тоді менеджер буде зв'язується з користувачем поштою, яку він вказував при реєстрації. При введенні даних і натисканні на клавішу «Змінити» дані в базі даних оновлюються для авторизованого користувача.

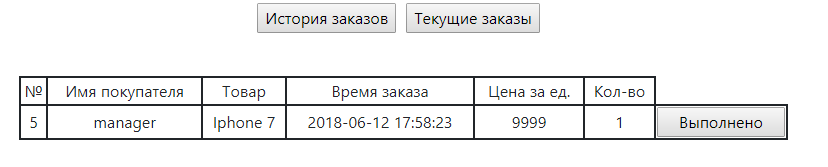


Малюнок А.3.15 – Форма для внесення даних про користувача

При вході від імені менеджера, в головне меню додається пункт «Заказы» (Малюнок А.3.16)



Малюнок А.3.16 – Меню при авторизації менеджера



Малюнок А.3.17 – Таблиця замовлень

Менеджер може натиснути на кнопку «Замовлення» в меню і подивитися поточні замовлення, а так само історію замовлень. Поточні замовлення він може виконувати, для цього він натискає на кнопку «виконано». При натисканні на неї даний запис видаляється з замовлень, проте вона збережена в історії замовлень.



Малюнок А.3.18 – Карта товару

У карті товару користувач може побачити наступну інформацію:

– найменування;

– ціну;

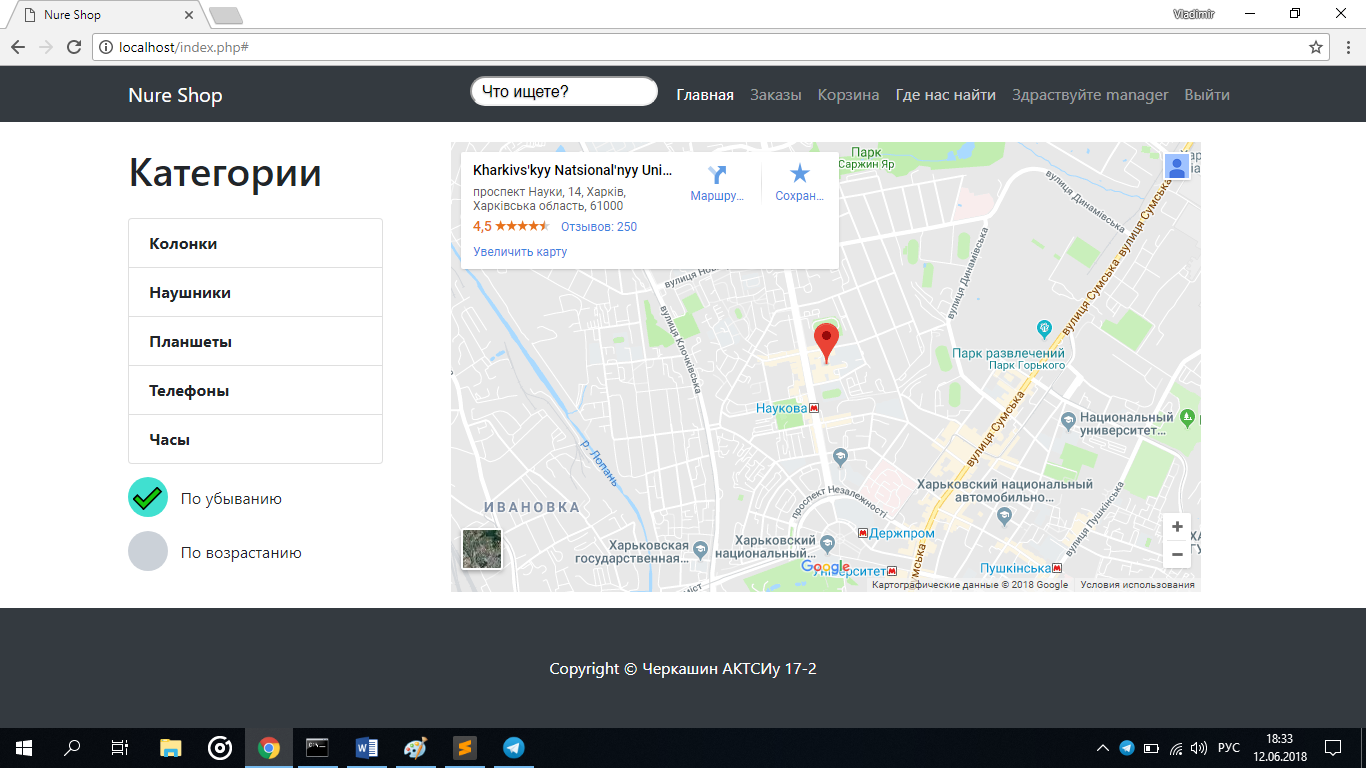
– текст товару;

– середня оцінка покупцями;

– кількість на сладе.

Середній рейтинг від покупців обчислюється за формулою кількість куплених конкретного товару поділений на загальну кількість куплених товарів. Якщо товарів даного виду не купували, тоді буде виведено – «Товар буде купувався».

У останньому пукнті «Где нас найти» будет загружена карта goole–maps з адресою нашого університета (Малюнок А.3.19).



Малюнок А.3.19 – Де нас знайти

|  |
| --- |
| Міністерство освіти і науки України  затверджую  Керівник курсової роботи,  доц. кафедри системотехніки  \_\_\_\_\_\_\_\_\_\_\_\_ Л.В. Колесник  (Підпис, дата)  ІНФОРМАЦІЙНА СИСТЕМА «Інтернет магазин»  текст програми  ЛИСТ затвердження  ГЮІК.508100.111 – 01 12 01 – ЛУ  Студент групи \_\_\_\_\_\_\_\_\_\_\_\_  (назва групи)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (Підпис, дата, прізвище, ім'я, по батькові)  2018 |

|  |
| --- |
| Міністерство науки і освіти України  ЗАТВЕРДЖУЮ  ГЮІК.508100.111 – 01 12 01 – ЛУ  ІНФОРМАЦІЙНА СИСТЕМА «Інтернет магазин»  текст програми  ГЮІК.508100.111 – 01 12 01 – ЛУ  листів 20  2018 |

Б.1 ТЕКСТ ПРОГРАМИ

– Файл index.php

<? Php

session\_start ();

include ( "db.php");

$ Categories = array ();

if ($ result = $ mysqli–> query ( 'SELECT \* FROM categories ORDER BY title')) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Categories [] = $ tmp;

}

$ Result–> close ();

}

$ Products = array ();

if ($ result = $ mysqli–> query ( "SELECT \* FROM products Where CatName LIKE 'Телефони'")) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Products [] = $ tmp;

}

$ Result–> close ();

}

?>

<! DOCTYPE html>

<Html lang = "ru">

<Head>

<Meta charset = "utf–8">

<Meta name = "viewport" content = "width = device–width, initial–scale = 1, shrink–to–fit = no">

<Title> Nure Shop </ title>

<Link href = "vendor / bootstrap / css / bootstrap.min.css" rel = "stylesheet">

<Link href = "css / shop–homepage.css" rel = "stylesheet">

</ Head>

<Body>

<! – Navigation –>

<Nav class = "navbar navbar–expand–lg navbar–dark bg–dark fixed–top">

<Div class = "container">

<a class="navbar–brand" href="#"> Nure Shop </a>

<Button class = "navbar–toggler" type = "button" data–toggle = "collapse" data–target = "# navbarResponsive" aria–controls = "navbarResponsive" aria–expanded = "false" aria–label = "Toggle navigation ">

<Span class = "navbar–toggler–icon"> </ span>

</ Button>

<Div class = "collapse navbar–collapse" id = "navbarResponsive">

<Ul class = "navbar–nav ml–auto">

<Li class = "nav–item active">

<Input id = "search" type = "text" placeholder = "Що шукаєте?">

</ Li>

<Li class = "nav–item active">

<a class="nav–link" href="index.php"> Головна

<Span class = "sr–only"> (current) </ span>

</a>

</ Li>

<? Php if (isset ($ \_ SESSION [ 'IsManager'])) {

if ($ \_SESSION [ 'IsManager']) {

?>

<Li class = "nav–item">

<a id="manager" class="nav–link" href="#"> Замовлення </a>

</ Li>

<? Php}}?>

<Li class = "nav–item">

<a class="nav–link" href="cart.php"> Кошик </a>

</ Li>

<Li class = "nav–item">

<a id="map" class="nav–link" href="#"> Де нас знайти </a>

</ Li>

<Li class = "nav–item">

<? Php if (isset ($ \_ SESSION [ 'login'])) {

echo "<a id='settings' class='nav–link' href='#'> Здраствуйте {$ \_SESSION [ 'login']} </a>";

}

else {

echo "<a class='nav–link' href='reg.php'> Вхід </a>";

}?>

</ Li>

<Li class = "nav–item">

<? Php if (isset ($ \_ SESSION [ 'login'])) {?>

<a class="nav–link" href="Exit.php"> Вийти </a>

<? Php}?>

</ Li>

</ Ul>

</ Div>

</ Div>

</ Nav>

<! – Page Content –>

<Div class = "container">

<Div class = "row">

<Div class = "col–lg–3">

<H1 class = "my–4"> Категорії </ h1>

<Div id = "cat" class = "list–group">

<? Php

foreach ($ categories AS $ category) {

echo '<a class="list–group–item">'.

$ Category [ 'title']. '</a>';

}

?>

</ Div>

<Div>

<Label>

<Input type = "radio" class = "option–input radio" name = "sort" checked />

За зменшенням

</ Label>

<Label>

<Input type = "radio" class = "option–input radio" name = "sort" />

За зростанням

</ Label>

</ Div>

</ Div>

<! – /.col–lg–3 –>

<Div class = "col–lg–9 fixed">

<Div id = "product" class = "row">

<? Php foreach ($ products AS $ product) {?>

<Div class = "col–lg–4 col–md–6 mb–4">

<Div class = "card h–100">

<a href="#"> <img class = "card–img–top" src = "<? php echo $ product [ 'img'];?>" alt = ""> </a>

<Div class = "card–body">

<H4 class = "card–title">

<a href="#"> <? php echo $ product [ 'title'];?> </a>

</ H4>

<H5> <? Php echo $ product [ 'price'];?> ₴ </ h5>

<P class = "card–text"> <? Php echo $ product [ 'intro'];?> </ P>

<P class = "rate"> купуватися: <? Php $ result = $ mysqli–> query ( "SELECT round (rate ( '{$ product [' title ']}'), 2) AS` rate` ");

$ Temp = $ result–> fetch\_assoc ();

if ($ temp [ 'rate']! = "") {

echo $ temp [ 'rate']. "% </ P>";

}

else {

echo "Ще не купували";

}

?>

<P> Залишилося: <? Php echo $ product [ 'quantity']. "</ P>"; ?>

</ Div>

<Div class = "card–footer">

<a href='#' id='add–<?php echo $product['id'];?> '> Додати в кошик </a>

</ Div>

</ Div>

</ Div>

<? Php}?>

</ Div>

<! – /.row –>

</ Div>

<! – /.col–lg–9 –>

</ Div>

<! – /.row –>

</ Div>

<! – /.container –>

<! – Footer –>

<Footer class = "py–5 bg–dark">

<Div class = "container">

<P class = "m–0 text–center text–white"> Copyright & copy; Черкашин АКТС 17–2 </ p>

</ Div>

</ Footer>

<Script src = "vendor / jquery / jquery.min.js"> </ script>

<Script src = "vendor / bootstrap / js / bootstrap.bundle.min.js"> </ script>

<Script src = "js / main.js"> </ script>

</ Body>

</ Html>

файл cart.class.php

<? Php

class Cart

{

private $ products;

function \_\_construct ()

{

$ This–> products = Cookie :: get ( 'Products') == null?

array ()

:

unserialize (Cookie :: get ( 'Products'));

}

public function getProducts ($ for\_sql = false)

{

if ($ for\_sql) {

return implode ( ',', $ this–> products);

}

return $ this–> products;

}

public function addProduct ($ id)

{

$ Id = (int) $ id;

if (! in\_array ($ id, $ this–> products)) {

array\_push ($ this–> products, $ id);

}

Cookie :: set ( 'Products', serialize ($ this–> products));

}

public function deleteProduct ($ id)

{

$ Id = (int) $ id;

$ Key = array\_search ($ id, $ this–> products);

if ($ key! == false) {

unset ($ this–> products [$ key]);

}

Cookie :: set ( 'Products', serialize ($ this–> products));

}

public function clear ()

{

Cookie :: delete ( 'Products');

}

public function isEmpty ()

{

return! $ this–> products;

}

}

файл cart.php

<? Php

session\_start ();

include ( "db.php");

require\_once ( 'cart.class.php');

require\_once ( 'cookie.class.php');

$ Cart = new Cart ();

$ Action = isset ($ \_ GET [ 'action'])? $ \_GET [ 'action']: 'list';

if ($ action == 'add') {

$ Id = $ \_GET [ 'id'];

$ Cart–> addProduct ($ id);

}

elseif ($ action == 'delete') {

$ Id = $ \_GET [ 'id'];

$ Cart–> deleteProduct ($ id);

}

elseif ($ action == 'clear') {

$ Cart–> clear ();

header ( 'Location: cart.php');

}

$ Products = array ();

$ Id\_sql = $ cart–> getProducts (true);

$ Sql ​​= "SELECT \* FROM products WHERE id IN ({$ id\_sql})";

$ Result = $ mysqli–> query ($ sql);

if ($ result) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Products [] = $ tmp;

}

}

?>

<! DOCTYPE html>

<Html lang = "en">

<Head>

<meta charset = "UTF–8">

<title> Кошик </ title>

<link rel = "stylesheet" href = "vendor / bootstrap / css / bootstrap.min.css">

<link rel = "stylesheet" href = "css / cart.css">

</ Head>

<Body>

<H1> Ваша Кошик </ h1>

<Form>

<Ul class = "dotted">

<? Php

$i = 0;

if ($ result) {

foreach ($ products AS $ product) {

echo "<li> <span> {$ product [ 'title']} </ span> <span> {$ product [ 'price']} ₴ </ span> </ li>

<Div class = 'but'>

<A id = 'deletes' href = 'cart.php? action = delete & id = {$ product [' id ']}' '> Видалити з кошика </a>

<Input class = 'quantity' name ='Order {$ i}' type = 'text' placeholder = 'Кількість?'>

</ Div> ";

$i ++;

}

}

?>

</ Ul>

</ Form>

<Div id = "result"> </ div>

<Div id = "buttonWrap">

<a id="getOrder" class="buttons" href="#"> Оформити замовлення </a>

<a class="buttons" href="cart.php?action=clear&id=<?php echo $product['id'];?> "> Очистити кошик </a>

<a class="buttons" href="index.php"> Повернутись на головну </a>

</ Div>

<Script src = "vendor / jquery / jquery.min.js"> </ script>

<Script src = "js / cart.js"> </ script>

</ Body>

</ Html>

файл cookies.class.php

<? Php

abstract class Cookie

{

public static function set ($ key, $ value, $ time = 31536000)

{

setcookie ($ key, $ value, time () + $ time, '/');

}

public static function get ($ key)

{

if (isset ($ \_ COOKIE [$ key])) {

return $ \_COOKIE [$ key];

}

return null;

}

public static function delete ($ key)

{

if (isset ($ \_ COOKIE [$ key])) {

self :: set ($ key, '', –3600);

unset ($ \_ COOKIE [$ key]);

}

}

}

файл db.php

<? Php

$ Mysqli = mysqli\_init ();

if (! $ mysqli) {

die ( 'mysqli\_init завершилася провалом');

}

if (! $ mysqli–> options (MYSQLI\_INIT\_COMMAND, 'SET AUTOCOMMIT = 0')) {

die ( 'Установка MYSQLI\_INIT\_COMMAND завершилася провалом');

}

if (! $ mysqli–> options (MYSQLI\_OPT\_CONNECT\_TIMEOUT, 5)) {

die ( 'Установка MYSQLI\_OPT\_CONNECT\_TIMEOUT завершилася провалом');

}

if (! $ mysqli–> real\_connect ( 'localhost', 'root', '', 'trainstation')) {

die ( 'Помилка підключення ('. mysqli\_connect\_errno (). ')'. mysqli\_connect\_error ());

}

$ Mysqli–> query ( "SET NAMES 'utf8'");

?>

файл Exit.php

<? Php

session\_start ();

$ \_SESSION [ 'login'] = NULL;

$ \_SESSION [ 'IsManager'] = NULL;

header ( 'Location: index.php');

файл manager.php

<? Php

include ( "db.php");

if (isset ($ \_ POST [ "manager"])) {

$ Mysqli–> query ( "DELETE FROM` orders` WHERE `orders`.`id` = '{$ \_ POST [' manager ']}'");

$ Mysqli–> commit ();

}

$ Orders = array ();

if ($ result = $ mysqli–> query ( "SELECT \* FROM` orders` ")) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Orders [] = $ tmp;

}

$ Result–> close ();

}

?>

<Div class = "order">

<Button id = "historyOrder"> Історія замовлень </ button>

<Button id = "currentOrder"> Поточні замовлення </ button>

</ Div>

<? Php

echo "<table id = 'table1'>"; ?>

<Tr>

<Td align = 'center'> № </ td>

<Td align = 'center'> Ім'я покупця </ td>

<Td align = 'center'> Товар </ td>

<Td align = 'center'> Час замовлення </ td>

<Td align = 'center'> Ціна за од. </ Td>

<Td align = 'center'> Кількість </ td>

</ Tr>

<? Php

/// початок таблиці

foreach ($ orders as $ value) {

echo "<tr>"; // починаємо рядок

foreach ($ value as $ key => $ val) {

echo "<td align = 'center'>"; // виводимо початок осередки

echo $ val;

echo "</ td>"; // виводимо кінець осередки

}

echo "<td> <button class = 'compleate' id = 'complete – {$ value [' id ']}'> Виконано </ button> </ td>";

echo "</ tr>"; // закінчуємо рядок

}

echo "</ table>"; // кінець таблиці

?>

файл manager.php

<? Php

include ( "db.php");

if (isset ($ \_ POST [ "manager"])) {

$ Mysqli–> query ( "DELETE FROM` orders` WHERE `orders`.`id` = '{$ \_ POST [' manager ']}'");

$ Mysqli–> commit ();

}

$ Orders = array ();

if ($ result = $ mysqli–> query ( "SELECT \* FROM` logorders` ")) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Orders [] = $ tmp;

}

$ Result–> close ();

}

?>

<Div class = "order">

<Button id = "historyOrder"> Історія замовлень </ button>

<Button id = "currentOrder"> Поточні замовлення </ button>

</ Div>

<? Php

echo "<table id = 'table1'>"; ?>

<Tr>

<Td align = 'center'> № </ td>

<Td align = 'center'> Ім'я покупця </ td>

<Td align = 'center'> Товар </ td>

<Td align = 'center'> Час замовлення </ td>

<Td align = 'center'> Ціна за од. </ Td>

<Td align = 'center'> Кількість </ td>

</ Tr>

<? Php

/// початок таблиці

foreach ($ orders as $ value) {

echo "<tr>"; // починаємо рядок

foreach ($ value as $ key => $ val) {

echo "<td align = 'center'>"; // виводимо початок осередки

echo $ val;

echo "</ td>"; // виводимо кінець осередки

}

echo "</ tr>"; // закінчуємо рядок

}

echo "</ table>"; // кінець таблиці

?>

файл map.php

<Div id = "MapWrap">

<Iframe src = "https://www.google.com/maps/embed?pb=!1m14!1m8!1m3!1d10255.176938719804!2d36.2280338!3d50.0152118!3m2!1i1024!2i768!4f13.1! 3m3! 1m2! 1s0x0% 3A0x6daed05394f03251! 2sKharkivs% CA% B9kyy + Natsional% CA% B9nyy + Universytet + Radioelektroniky! 5e0! 3m2! 1sru! 2sua! 4v1528491036964 "width =" 750 "margin =" 0 auto "height =" 450 " frameborder = "0" style = "border: 0" allowfullscreen> </ iframe>

</ Div>

<Style>

#MapWrap {

margin: 0 auto;

margin–top: 20px;

}

</ Style>

файл order.php

<? Php

session\_start ();

include ( "db.php");

require\_once ( 'cart.class.php');

require\_once ( 'cookie.class.php');

$ Cart = new Cart ();

if ($ cart–> getProducts (TRUE) == "") {

echo 'Ні продуктів для замовлення !!!';

die ();

}

$ Data = $ \_POST [ "data"];

$ Data = json\_decode ($ data);

$ Products = array ();

$ Id\_sql = $ cart–> getProducts (true);

$ Sql ​​= "SELECT \* FROM products WHERE id IN ({$ id\_sql})";

$ Result = $ mysqli–> query ($ sql);

if ($ result) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Products [] = $ tmp;

}

}

$ Error = FALSE;

$i = 0;

for ($ j = 0; $ j <count ($ data); $ j ++) {

if ($ data [$ j] <0) {

echo "Заборонений вводити негативні значення";

die ();

}

}

foreach ($ products as $ product) {

$tmp = $ mysqli–> query ( "SELECT` quantity` FROM `products` WHERE` title` = '{$ product [' title ']}' ");

$tmp = $ tmp–> fetch\_assoc ();

if ($ data [$ i] <= $ tmp [ 'quantity']) {

$mysqli–> query ( "INSERT INTO` orders` ( ​​`id`,` user`, `title`,` DateTime`, `price`,` quantity`)

VALUES (NULL, '{$ \_SESSION [ 'login']} ',' {$ product [ 'title']} ', NOW (),' {$ product [ 'price']} ',' {$ data [$ i]} ') ") ;

$mysqli–> commit ();

$ Cart–>deleteProduct ($ product [ 'id']);

}

else {

echo "Такої кількості {$ product [ 'title']} немає на складі";

$ Error = TRUE;

}

$i ++;

}

if ($ error == FALSE) {

echo "Ваше замовлення оброблений, очікуйте листа менеджера <a href='index.php'> Повернутись на головну </a>";

}

?>

файл reg.php

<! DOCTYPE html>

<Html lang = "ru">

<Head>

<Meta charset = "UTF–8">

<Title> Nure Login Form </ title>

<Link rel = "stylesheet" href = "https://cdnjs.cloudflare.com/ajax/libs/meyer–reset/2.0/reset.min.css">

<Link rel = 'stylesheet prefetch' href = 'https: //fonts.googleapis.com/css? Family = Roboto: 400,100,300,500,700,900'>

<Link rel = 'stylesheet prefetch' href = 'https: //fonts.googleapis.com/css? Family = Montserrat: 400,700'>

<Link rel = 'stylesheet prefetch' href = 'https: //maxcdn.bootstrapcdn.com/font–awesome/4.3.0/css/font–awesome.min.css'>

<Link rel = "stylesheet" href = "css / reg.css">

</ Head>

<Body>

<Div class = "container">

</ Div>

<Div class = "form">

<Div class = "thumbnail"> <img src = "https://s3–us–west–2.amazonaws.com/s.cdpn.io/169963/hat.svg" /> </ div>

<Form class = "register–form" action = "SaveUser.php" method = "POST">

<Input name = "login" type = "text" placeholder = "Логін" />

<Input name = "password" type = "password" placeholder = "Пароль" />

<Input name = "email" type = "text" placeholder = "Email" />

<Button> Створити </ button>

<P class = "message"> Уже зареєстровані? <a href="#"> Ввійти </a> </ p>

</ Form>

<Form class = "login–form" action = "SingUp.php" method = "POST">

<Input name = "login" type = "text" placeholder = "Логін" />

<Input name = "password" type = "password" placeholder = "Пароль" />

<Button> Вхід </ button>

<P class = "message"> Чи не зареєстровані? <a href="#"> Створити рахунок </a> </ p>

</ Form>

</ Div>

<Script src = 'vendor / jquery / jquery.min.js'> </ script>

<Script src = "js / reg.js"> </ script>

</ Body>

</ Html>

файл saveUser.php

<? Php

if (isset ($ \_ POST [ 'login'])) {$ login = $ \_POST [ 'login']; if ($ login == '') {unset ($ login);}}

if (isset ($ \_ POST [ 'password'])) {$ password = $ \_ POST [ 'password']; if ($ password == '') {unset ($ password);}}

if (isset ($ \_ POST [ 'email'])) {$ email = $ \_POST [ 'email']; if ($ email == '') {unset ($ email);}}

if (empty ($ login) or empty ($ password) or empty ($ email))

{

exit ( "Ви ввели не всю інформацію");

}

$ Login = stripslashes ($ login);

$ Login = htmlspecialchars ($ login);

$ Password = stripslashes ($ password);

$ Password = htmlspecialchars ($ password);

$ Email = stripslashes ($ email);

$ Email = htmlspecialchars ($ email);

$ Login = trim ($ login);

$ Password = trim ($ password);

$ Email = trim ($ email);

$ Db = mysqli\_connect ( "localhost", "root", "", "trainstation");

$ Result = mysqli\_query ($ db, "SELECT id FROM users WHERE login = '$ login'");

$ Myrow = mysqli\_fetch\_array ($ result);

if (! empty ($ myrow [ 'id'])) {

exit ( "Введіть інший логін.");

}

$ Result2 = mysqli\_query ($ db, "INSERT INTO users (login, password, email) VALUES ( '$ login', '$ password', '$ email')");

if ($ result2 == 'TRUE')

{

echo "Ви успішно зареєстровані! <a href='index.php'> Головна сторінка </a>";

}

else {

echo "Помилка! Ви не зареєстровані.";

}

?>

файл search.php

<? Php

include ( "db.php");

$ Search = $ \_POST [ "search"];

$ Products = array ();

if ($ result = $ mysqli–> query ( "SELECT \* FROM products Where title LIKE '% $ search%'")) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Products [] = $ tmp;

}

$ Result–> close ();

}

foreach ($ products AS $ product) {?>

<Div class = "col–lg–4 col–md–6 mb–4">

<Div class = "card h–100">

<a href="#"> <img class = "card–img–top" src = "<? php echo $ product [ 'img'];?>" alt = ""> </a>

<Div class = "card–body">

<H4 class = "card–title">

<a href="#"> <? php echo $ product [ 'title'];?> </a>

</ H4>

<H5> <? Php echo $ product [ 'price'];?> ₴ </ h5>

<P class = "card–text"> <? Php echo $ product [ 'intro'];?> </ P>

<P class = "rate"> купуватися: <? Php $ result = $ mysqli–> query ( "SELECT round (rate ( '{$ product [' title ']}')) AS` rate` ");

$ Temp = $ result–> fetch\_assoc ();

if ($ temp [ 'rate']! = "") {

echo $ temp [ 'rate']. "% </ P>";

}

else {

echo "Ще не купували </ p>";

}

?>

<P> Залишилося: <? Php echo $ product [ 'quantity']. "</ P>"; ?>

</ Div>

<Div class = "card–footer">

<a href='#' id='add–<?php echo $product['id'];?> '> Додати в кошик </a>

</ Div>

</ Div>

</ Div>

<? Php}?>

файл SelectCategories.php

<? Php

session\_start ();

include ( "db.php");

if (isset ($ \_ POST [ "name"])) {

$ Name = $ \_POST [ "name"];

}

if (isset ($ \_ POST [ "sort"])) {

$ Sort = $ \_POST [ "sort"];

}

else {

$ Sort = 'DESC';

}

if (isset ($ \_ POST [ "name"])) {

$ \_SESSION [ 'catname'] = $ name;

}

$ Products = array ();

if ($ result = $ mysqli–> query ( "SELECT \* FROM products Where CatName = '{$ \_ SESSION [' catname ']}'

ORDER BY `price` $ sort")) {

while ($ tmp = $ result–> fetch\_assoc ()) {

$ Products [] = $ tmp;

}

$ Result–> close ();

}

foreach ($ products AS $ product) {?>

<Div class = "col–lg–4 col–md–6 mb–4">

<Div class = "card h–100">

<a href="#"> <img class = "card–img–top" src = "<? php echo $ product [ 'img'];?>" alt = ""> </a>

<Div class = "card–body">

<H4 class = "card–title">

<a href="#"> <? php echo $ product [ 'title'];?> </a>

</ H4>

<H5> <? Php echo $ product [ 'price'];?> ₴ </ h5>

<P class = "card–text"> <? Php echo $ product [ 'intro'];?> </ P>

<P class = "rate"> купуватися: <? Php $ result = $ mysqli–> query ( "SELECT round (rate ( '{$ product [' title ']}')) AS` rate` ");

$ Temp = $ result–> fetch\_assoc ();

if ($ temp [ 'rate']! = "") {

echo $ temp [ 'rate']. "% </ P>";

}

else {

echo "Ще не купували </ p>";

}

?>

<P> Залишилося: <? Php echo $ product [ 'quantity']. "</ P>"; ?>

</ Div>

<Div class = "card–footer">

<a href='#' id='add–<?php echo $product['id'];?> '> Додати в кошик </a>

</ Div>

</ Div>

</ Div>

<? Php}?>

файл settings.php

<? Php

session\_start ();

include ( "db.php");

if (isset ($ \_ POST [ "data"])) {

$ Data = $ \_POST [ "data"];

$ Data = json\_decode ($ data, true);

$ Mysqli–> query ( "UPDATE` userinfo` SET `Name` = '{$ data [' Name ']}',` LastName` = '{$ data [' LastName ']}',

`MiddleName` = '{$ data [' MiddleName ']}', `cart` = '{$ data [' cart ']}',` phone` = '{$ data [' phone ']}',

`addr` = '{$ data [' addr ']}' WHERE `id` = (SELECT` idInfo` from `users` WHERE` login` = '{$ \_SESSION [' login ']}')

");

$ Mysqli–> commit ();

}

$ Name = $ mysqli–> query ( "SELECT \* FROM` userinfo` WHERE `id` = (SELECT` idInfo` from `users` WHERE` login` = '{$ \_SESSION [' login ']}') ");

$ Info = $ name–> fetch\_assoc ();

?>

<Form id = "application">

<Input name = "name" id = "applicationName" maxlength = "20" placeholder = "<? Php if ($ info [ 'Name']) {echo $ info [ 'Name'];} else {echo 'Введіть ім'я ';}?> "required />

<Input name = "lastName" id = "applicationLastName" maxlength = "20" placeholder = "<? Php if ($ info [ 'LastName']) {echo $ info [ 'LastName'];} else {echo 'Введіть Прізвище ';}?> "" required />

<Input name = "MiddleName" id = "applicationMiddleName" maxlength = "20" placeholder = "<? Php if ($ info [ 'MiddleName']) {echo $ info [ 'MiddleName'];} else {echo 'Введіть батькові ';}?> "" required />

<Input name = "cart" type = "text" id = "applicationCart" maxlength = "20" placeholder = "<? Php if ($ info [ 'cart']) {echo $ info [ 'cart'];} else {echo 'Введіть номер карти';}?> "required />

<Input name = "telephone" type = "Tel" id = "applicationTelephone" maxlength = "20" placeholder = "<? Php if ($ info [ 'phone']) {echo $ info [ 'phone'];} else {echo 'Введіть номер телефону';}?> "required />

<Input name = "name" id = "applicationAddr" maxlength = "20" placeholder = "<? Php if ($ info [ 'addr']) {echo $ info [ 'addr'];} else {echo 'Введіть адресу доставки ';}?> "required />

<Button id = "change" class = "applicationButton" type = "submit" form = "application"> Змінити </ button>

</ Form>

файл SingUp.php

<? Php

session\_start ();

include ( "db.php");

if (isset ($ \_ POST [ 'login'])) {$ login = $ \_POST [ 'login']; if ($ login == '') {unset ($ login);}}

if (isset ($ \_ POST [ 'password'])) {$ password = $ \_ POST [ 'password']; if ($ password == '') {unset ($ password);}

}

if (empty ($ login) or empty ($ password))

{

exit ( "Ви ввели не всю інформацію, поверніться назад і заповніть всі поля!");

}

$ Login = stripslashes ($ login);

$ Login = htmlspecialchars ($ login);

$ Password = stripslashes ($ password);

$ Password = htmlspecialchars ($ password);

$ Login = trim ($ login);

$ Password = trim ($ password);

$ Result = $ mysqli–> query ( "SELECT \* FROM users WHERE login = '$ login'");

$ Myrow = mysqli\_fetch\_array ($ result);

if (empty ($ myrow [ 'password']))

{

exit ( "Вибачте, введений вами login або пароль невірний.");

}

else {

if ($ myrow [ 'password'] == md5 ($ password)) {

$ \_SESSION [ 'IsManager'] = $ myrow [ 'IsManager'];

$ \_SESSION [ 'login'] = $ myrow [ 'login'];

echo "Ви успішно увійшли на сайт! <a href='index.php'> Головна сторінка </a>";

}

else {

exit ( "Вибачте, введений вами login або пароль невірний.");

}

}

файл main.js

$ (Document) .ready (function () {

let cat = $ ( '# cat'). children ();

for (let i = 0; i <cat.length; i ++) {

cat [i] .onclick = SelCat;

}

let name = "";

function SelCat () {

$ .ajax ({

type: 'POST',

url: '../ SelectCatagoty.php',

data: 'name =' + this.innerText,

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

}

$ ( "Body"). On ( 'click', "a [id | = 'add']", function () {

id = $ (this) .attr ( 'id'). split ( '–');

id = id [1];

$ .get ( 'cart.php? Action = add & id =' + id, alert ( 'Product' + id + 'added'));

});

$ ( "Body"). On ( 'click', "# currentOrder", function () {

$ .ajax ({

type: 'POST',

url: '../ manager.php',

data: '',

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

});

$ ( "Body"). On ( 'click', "# historyOrder", function () {

$ .ajax ({

type: 'POST',

url: '../ managerH.php',

data: '',

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

});

$ ( "Body"). On ( 'click', "[id ^ = complete]", function () {

let id = $ (this) .attr ( 'id'). split ( '–');

id = id [1];

$ .ajax ({

type: 'POST',

url: '../ manager.php',

data: {manager: id},

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

});

$ ( "Body"). On ( 'click', "[id = change]", function () {

let data = {};

data.Name = $ ( '# applicationName') [0] .value;

data.LastName = $ ( '# applicationLastName') [0] .value;

data.MiddleName = $ ( '# applicationMiddleName') [0] .value;

data.cart = $ ( '# applicationCart') [0] .value;

data.phone = $ ( '# applicationTelephone') [0] .value;

data.addr = $ ( '# applicationAddr') [0] .value;

data = JSON.stringify (data);

console.log (data);

$ .ajax ({

type: 'POST',

url: '../ settings.php',

data: {data: data},

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

});

$ ( '# Search'). On ( "keyup", function () {

if (event.keyCode == 13) {

$ .ajax ({

type: 'POST',

url: '../ search.php',

data: 'search =' + this.value,

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

}

});

$ ( '# Map') [0] .onclick = function () {

$ .ajax ({

type: 'POST',

url: '../ map.php',

data: '',

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

};

$ ( "Body"). On ( 'click', "# settings", function () {

$ .ajax ({

type: 'POST',

url: '../ settings.php',

data: '',

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

});

$ ( 'Input [name = sort') [0] .onclick = function () {

$ .ajax ({

type: 'POST',

url: '../ SelectCatagoty.php',

data: 'sort = DESC',

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

};

$ ( 'Input [name = sort') [1] .onclick = function () {

$ .ajax ({

type: 'POST',

url: '../ SelectCatagoty.php',

data: 'sort = ASC',

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

};

$ ( "Body"). On ( 'click', "# manager", function () {

$ .ajax ({

type: 'POST',

url: '../ manager.php',

data: '',

success: function (data) {

$ ( '# Product') [0] .innerHTML = data;

}});

});

});

файл reg.js

$ ( '. Message a'). Click (function () {

$ ( 'Form'). Animate ({height: "toggle", opacity: "toggle"}, "slow");

});

файл cart.js

$ (Document) .ready (function () {

$ ( '# GetOrder'). On ( 'click', function () {

let data = [];

for (let i = 0; i <$ ( 'input [name ^ = order'). length; i ++) {

if ($ ( 'input [name ^ = order') [i] .value == "") {

data.push ( '1');

}

else {

data.push ($ ( 'input [name ^ = order') [i] .value);

}

}

data = JSON.stringify (data);

$ .ajax ({

type: 'POST',

url: '../ order.php',

data: {data: data},

success: function (data) {

$ ( '# Result') [0] .innerHTML = data;

}});

})

});

Б.2 SQL КОД

CREATE FUNCTION rate (r varchar (15)) RETURNS FLOAT NOT DETERMINISTIC

BEGIN

DECLARE rate FLOAT DEFAULT 0;

SELECT (SUM (quantity) / (SELECT SUM (quantity) FROM logorders)) \* 100 into rate FROM logorders WHERE title = r;

RETURN rate;

END //

CREATE PROCEDURE myversion ()

BEGIN

SELECT VERSION ();

END //

CREATE DEFINER = `root` @` localhost` TRIGGER `log` AFTER INSERT ON` orders` FOR EACH ROW BEGIN DECLARE var1 int (11) DEFAULT 0; SELECT quantity into var1 FROMproducts WHERE title = NEW.title; IF var1> = NEW.quantity THEN INSERT INTO logorders SELECT \* FROM orders WHERE id = NEW.id; UPDATE products SET products.quantity = products.quantity–NEW.quantity WHERE title = NEW.title; END IF; END