

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА
ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій

Кафедра комп'ютерних наук та технологій

ДИПЛОМНА БАКАЛАВРСЬКА РОБОТА

на тему

Розробка бази даних інтернет магазину з продажу комп'ютерної техніки

Виконав: студент групи БІТск-19
спеціальності
122 Комп'ютерні науки

Нестеров Владислав Ігоревич

Науковий керівник Демківська Т. І.

Рецензент Чупринка В.І.

Київ 2021

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ТЕХНОЛОГІЙ ТА ДИЗАЙНУ

Факультет мехатроніки та комп'ютерних технологій
Кафедра комп'ютерних наук та технологій
Спеціальність 122 Комп'ютерні науки
Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри
комп'ютерних наук та технологій
_____ В.Ю.Щербань
(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ

НА ДИПЛОМНУ БАКАЛАВРСЬКУ РОБОТУ

студенту
Нестерову Владиславу

Розробка бази даних інтернет магазину з продажу комп'ютерної техніки

Науковий керівник роботи Демківська Тетяна Іванівна

затверджені наказом КНУТД від “15” березня 2021 року № 75-уч

2. Строк подання студентом дипломної роботи 14.06.2021 р.
3. Вихідні дані до дипломної роботи (проекту) Розробки кафедри комп'ютерних наук та технологій, рекомендована література
4. Зміст дипломної бакалаврської роботи: 1) загальний огляд інтернет магазинів, 2) принципи побудови та роботи інтернет магазинів, 3) розробка інтернет магазину та його програмна реалізація
5. Дата видачі завдання 15.03.2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної бакалаврської роботи	Терміни виконання етапів	
1	Вступ	06.05.2021	
2	Розділ 1 Загальний огляд інтернет магазинів	14.05.2021	
3	Розділ 2 Принципи побудови та роботи інтернет магазинів	22.05.2021	
4	Розділ 3 Розробка інтернет магазину та його програмна реалізація	30.05.2021	
5	Висновки	02.06.2021	
6	Оформлення дипломної бакалаврської роботи (чистовий варіант)	03.06.2021	
7	Здача дипломної бакалаврської роботи на кафедрі для рецензування (за 14 днів до захисту)	04.06.2021	
8	Перевірка дипломної бакалаврської роботи на наявність ознак плагіату (за 10 днів до захисту)	08.06.2021	
9	Подання дипломної бакалаврської роботи на затвердження завідувачу кафедри (за 7 днів до захисту)	08.06.2021	

Студент _____ Нестеров Владислав
(підпис)

Науковий керівник _____ Т. І. Демківська
роботи (підпис)

Рецензент _____ В.І. Чупринка
(підпис)

АНОТАЦІЯ

Нестеров Владислав Ігоревич. Тема Розробка бази даних інтернет магазину з продажу комп'ютерної техніки

Дипломна робота складається з 96 сторінок з яких 63 стр., основний текст, 23 рис., 1 таблиць, 1 додаток на 27 сторінках.

Об'єкт дослідження: Інтернет-магазин з продажу комп'ютерної техніки

Мета роботи: розробити web-сайт магазину з продажу комп'ютерної техніки та базу даних «Інтернет-магазину», для швидкого і комфортного користування та збереження інформації в базі даних.

Методи розробки: мова програмування PHP, середовище програмування Microsoft Visual Studio, база даних MySQL.

У першому розділі був проведений аналіз предметної області, огляд переваг та недоліків.

У другому розділі визначено принцип роботи та структура побудови завдання.

У третьому розділі описана розробка програмної реалізації та її тестування.

У додатках наведено: текст програми та її опис.

КЛЮЧОВІ СЛОВА: WEB-САЙТ, ІНТЕРНЕТ МАГАЗИН, БАЗА ДАНИХ.

SUMMARY

Nesterov Vladislav Igorevich. Topic Development of an online store database for the sale of computer equipment.

Thesis consists of 96 pages, of which 63 pages, main text, 23 figures, 1 tables, 1 appendix on 27 pages.

Object of research: Online store selling computer equipment Purpose: to develop a web-site of a store selling computer equipment and a database of «Online Store».

Purpose: to develop a web-site of a store selling computer equipment and a database of «Online Store», for quick and comfortable use and storage of information in the database. Development methods: PHP programming language, Microsoft Visual Studio programming environment, MySQL database.

In the first section, an analysis of the subject area, an overview of the advantages and disadvantages.

The second section defines the principle of operation and structure of the task.

The third section describes the development of software implementation and its testing.

The appendices contain: the text of the program and its description.

KEY WORDS: WEB SITE, ONLINE STORE, DATABASE.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 ЗАГАЛЬНИЙ ОГЛЯД ІНТЕРНЕТ-МАГАЗИНІВ.....	8
1.1.Інтернет-магазини та їх особливості	8
1.2. Класифікація інтернет-магазинів.....	11
1.3 Переваги та недоліки Інтернет-магазинів.....	13
1.4 Постановка задачі.....	18
РОЗДІЛ 2. ПРИНЦИПИ ПОБУДОВИ ТА РОБОТИ ІНТЕРНЕТ МАГАЗИНІВ.....	19
2.1 Структура побудови Інтернет-магазину.....	19
2.2 Аналіз платіжних систем.....	21
3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ ТА ЙОГО ПРОГРАМНА РЕАЛІЗАЦІЯ.....	27
3.1 Огляд використаної мови програмування PHP.....	27
3.2 Огляд використаної бази даних MySQL.....	36
3.3 Організація діяльності Інтернет-магазину.....	46
3.4 Опис роботи Web-сайту «Інтернет-магазин».....	52
ВИСНОВКИ.....	58
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ.....	59
ДОДАТОК ЛІСТИНГ ПРОГРАМИ.....	61

ВСТУП

Актуальність роботи. Розвиток покупок в Інтернеті та їх використання є актуальною проблемою сьогодні, адже багата кількість людей щодня купують товари в електронних магазинах, при цьому не виходячи з дому. У світі, і зокрема в Україні, кількість користувачів Інтернету, а отже, і кількості «електронних» користувачів зростає.

Інтернет зближує людей, реакція на будь-яку подію настає майже відразу, і відстань зникає. Інтернет-магазини значно зменшують витрати виробників, економить на утриманні звичайного магазину, розширюють торгові точки та розширюють можливості покупця - купуючи будь-який товар у будь-який час, у будь-якій країні, у будь-якому місті та у будь-який час доби. Це дає Інтернет-магазинам перевагу перед звичайними магазинами. Цей момент важливий під час переходу виробників від «звичайної» до «електронної» торгівлі[1].

Інтернет-магазини, які пережили кризу, зміцнили свої позиції на ринку, купуючи менш успішних конкурентів. Офлайн-бізнес-гіганти, котрі в 1999 році були вдсятеро дешевшими за своїх онлайн-колег, також мали можливість вийти на ринок Інтернету після кризи, пропонуючи послуги вищої якості та відоме ім'я.

Традиційно в Україні компанії, що надають послуги доступу до Інтернету, працюють у сфері електронної комерції: Інтернет-портали, пошукові системи, послуги електронної пошти, нові Інтернет-агенції. Немає офлайн-лідерів ринку чи представників великих міжнародних інтернет-компаній [8]. Традиційний ринок електронної комерції розвивався як ринок роздрібною торгівлі. Однак поступово рішення спрямовуються не до кінцевих споживачів, а до організацій, т. Зв Ринок B2b, тобто бізнес від бізнесу. Спочатку ринок B2b пропонував товари, пов'язані з організацією роздрібною торгівлі (готові інтернет-магазини, рекламні послуги, введення покупок в Інтернеті до традиційних бізнес-процесів тощо), проте поступово неінтернет-рішення (торгове програмне забезпечення, послуги з автоматизації, оптові, брокерські

послуги тощо) почали продаватися через Інтернет.

Вищевизначене свідчить про правомірність дослідження Інтернет-магазину та його розвитку із застосуванням сучасних Інтернет-технологій. В рамках магістерської роботи ми маємо завдання розробити веб-сайт «Інтернет-магазин комп'ютерного обладнання».

РОЗДІЛ 1 ЗАГАЛЬНИЙ ОГЛЯД ІНТЕРНЕТ-МАГАЗИНІВ

1.1 Інтернет-магазини та їх особливості

Інтернет-магазин - це програмне забезпечення, яке продає товари чи послуги через Інтернет та автоматизує управління бізнес-процесами. Інтернет-магазини поєднують елементи прямого маркетингу та традиційної торгівлі. Основні відмінності між інтернет-магазином та традиційним - це інтерактивність, велика кількість інформації та асортименту та індивідуальний підхід до кожного відвідувача. Найбільшим недоліком електронних магазинів є неможливість торкнутися товару та візуально оцінити його. Однак цей недолік успішно компенсується великою кількістю інформації, яку продавець не зможе надати в традиційному магазині.

Таблиця 1.1.

Порівняльна характеристика традиційної та електронної комерції

Звичайний магазин	Інтернет - магазин
Місце для торгівлі	Інтернет - магазин
Відвідання покупця торгового залу і вибір товарів на полицях	Перегляд покупцем сторінок сервера
Прямий контакт покупця з продавцем	Консультація у продавця (якщо треба) по в мережах або телефоном
Вибір покупцем товару	Вибір покупцем товару
Покупка товару	Замовлення товару через інтернет
Видача чеку на сплату	Пересилка рахунку на оплату
Оплата покупцем рахунку на товар в касі магазину готівкою або банківською картою	Оплата покупцем рахунку по якій-небудь системі електронних платежів (банківська карта, електронний чек, цифрові гроші, електронні гроші)

Основні функції електронного магазину включають [21]:

- надання максимально повної інформації про представлені товари та послуги;
- прийняття та виконання замовлень;
- персоналізація відвідувачів;
- здійснення платежів (за умови підключення до платіжної системи);
- збір та аналіз статистичної інформації.

Інтернет-магазин - це веб-сайт, обладнаний веб-сайтом Інтернет-магазину, який дозволяє приймати замовлення не тільки по телефону, а й через веб-сайт електронними засобами.

Інтернет-магазин дозволяє здійснювати платежі клієнтам як у готівковій, так і в безготівковій формі, з можливістю роздрукувати рахунок-фактуру та вибрати опції з « ПДВ» та без нього.

Інтернет-магазин також включає всі функції рішення корпоративного веб-сайту, що дозволяє поєднувати повноцінний корпоративний веб-сайт із роздрібним або оптовим інтернет-магазином.

Інтернет-магазин підходить для тих, хто хоче почати ефективно, швидко та дешево продавати через веб-сайт інтернет-магазину з функцією онлайн-замовлення.

1.2. Класифікація інтернет - магазинів

Критерії магазинів

1. Класифікація за бізнес-моделлю:

- Інтернет-магазин: поєднання офлайн та онлайн-активності (коли в Інтернеті магазин був побудований на основі фактичної комерційної структури, яка вже існувала).

існували).

2. Що стосується постачальників:

- мати власний склад (наявність фактичних запасів)

- вони працюють на підставі договорів з постачальниками (не суттєві власний запас).

3. Класифікація за асортиментом - книги, аудіо, відеокасети, CD, DVD, комп'ютери, побутова техніка, мобільні телефони.

Методи роздрібної торгівлі товарами в Інтернеті включають:

- Інтернет-магазини (автоматичні магазини);

- веб-сайти;

- Торговельні автомати.

Веб-сайт інтернет-магазину - це більше рекламний сервер. На веб-сайті відображається інформація про товари, які постійно оновлюються. Вартість його створення та адміністрування може бути досить низькою, і практичні вигоди від цього очевидні. Але це не торгівля. Потенційний покупець, відвідавши сайт, повинен зателефонувати компанії, оплатити товар, домовитись про доставку. Тому веб-сайт виправданий у тих випадках, коли покупцеві потрібно ознайомитися зі складними товарами, вивчення яких у торговому залі займе багато часу.

Інтернет-вітрину можна розмістити де завгодно - на власному сервері, на сервері провайдера, на сервері, що забезпечує безкоштовні веб-сайти. Все, що потрібно для роботи з вітриною - це підключення до телефонної лінії та мінімальне знання HTML.

Система може виконати не тільки функції веб-сайту, але й приймати замовлення та пересилати їх менеджеру, тобто розміщувати замовлення та виставляти рахунки-фактури для оплати без присутності покупця. Торговий автомат насправді торгує, і співвідношення витрат і вигод є найбільш переважним для тестових проєктів з низьким потоком. Торговий автомат, як і веб-сайт, можна розмістити як на власному сервері, так і на сервері провайдера. Однак для його створення та адміністрування потрібні навички та певна кваліфікація.

Автоматичний магазин – це комплексне рішення в комерційному бізнесі. Він не тільки виставляє рахунки, але й відстежує замовлення, приймає електронні платежі та створює замовлення на доставку товарів споживачам. Тут робота менеджера - контролювати роботу системи, яку важко підтримувати..

Оскільки автоматизований магазин повинен мати постійне з'єднання з ІТ-системою компанії, краще розміщувати його або на сервері компанії в локальній мережі, або на віддаленому сервері з постійним каналом зв'язку. Інтернет-магазини можуть торгувати чим завгодно.

1.3 Переваги та недоліки Інтернет-магазинів

Останніми роками покупка будь-якого товару в інтернет-магазинах стала майже поширеною. Це дуже серйозний крок до прогресу, насамперед тому, що він дає людям можливість уникнути необхідності їхати до торгової точки з продажу цегли та розчину. Звичайно, прості магазини як існували раніше, так і існують зараз, і вони завжди будуть там, доки є люди, які люблять «робити покупки». Однак не у всіх є така можливість. Наприклад, більшість людей настільки зайняті роботою та вирішенням деяких побутових проблем, що просто немає вільного часу для відвідування салонів та магазинів. Але що, якщо все-таки потрібно зробити серйозну покупку, як комп'ютер? Скористайтеся послугою Інтернет-магазину. Перш за все, це гарантія отримання сертифікованого продукту дійсно високої якості.

По-друге, це шанс уникнути виснажливих пошуків. По-третє, ви можете заощадити, адже вартість товарів в інтернет-магазинах часто трохи нижча за стандартну. По-четверте, ви можете знайти все, що завгодно, адже в Інтернеті ви можете придбати майже все. Не можна недооцінювати можливість безготівкового розрахунку, оскільки це дуже зручно, але, на жаль, не всі стаціонарні торгові точки дають своїм клієнтам такий шанс. Інтернет-покупки - це дуже необхідна і важлива річ.

Тому, якщо є попит, буде і пропозиція. Інтернет-магазинів стає все більше і більше. Це дуже корисна можливість розширити свій бізнес за рахунок збільшення продажів.

Структура інтернет-магазину мало чим відрізняється від традиційного, і лише, на відміну від останнього, він повинен працювати в більш складних умовах, оскільки в ньому відсутні переваги звичайного магазину у вигляді постійної аудиторії. Постійні відвідувачі в Інтернет-магазинах дуже важливі, оскільки половину покупок роблять «повторні» клієнти..

Тому, щоб утримати цих клієнтів та залучити нових, Інтернет-магазин повинен орієнтуватися на ті зручності для покупця, які можливі завдяки Інтернет-магазинам

[1,20].

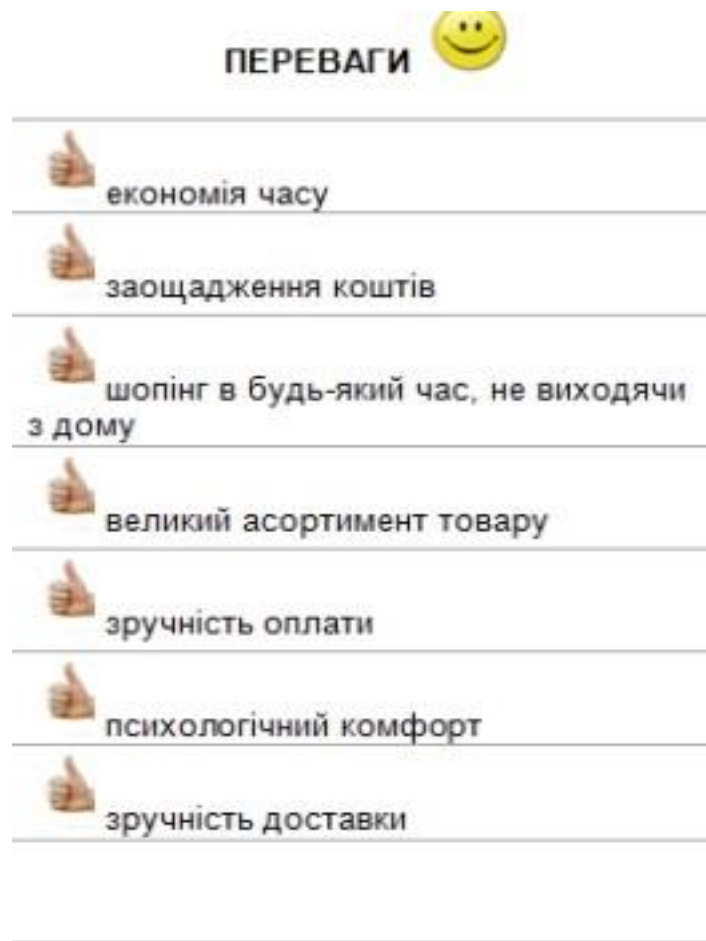


Рис. 1.1. Переваги Інтернет-магазинів

Чому люди все більше купують віддалено? Розглянемо ряд причин.

Економія свого часу. В Інтернеті можна вибрати і порівняти характеристики товарів з декількох магазинів, і процедура займає кілька хвилин, на відміну від нудної поїздки в традиційні магазини. Купуючи потрібні вам товари в Інтернеті, ви економите стільки дорогого часу. Інтернет-магазин доступний цілодобово, 7 днів на тиждень, 365 днів на рік. Працює без свят та вихідних, без черг. Покупець може в будь-який час відмовитись від вибору товару, якщо на даний момент часу немає, і повернутися до вибору навіть через кілька днів (всі дані про вибраний

товар залишаються в обліковому записі користувача). Вибір можна розширити на дозвіллі - на роботі чи вдома, ввечері або вночі..

Досить низькі ціни на товар. Більшість товарів в Інтернеті можна придбати за нижчою ціною, ніж у звичайному супермаркеті. У чому причина цього? По-перше, немає необхідності орендувати комерційні площі в Інтернет-магазині, по-друге, відсутні витрати на охорону та обслуговування торгових центрів, не потрібно витратити гроші на прийняття на роботу та навчання консультантів з продажу, по-третє, не потрібно платити за посередники.

Безкоштовна доставка (часто). Після оформлення замовлення та передачі його кур'єру кур'єр доставляє покупку до вашого дому чи офісу безкоштовно, в найкращий час. Покупці в інтернет-магазинах «не болять» від отримання замовлення і не обтяжені однією лише думкою вивезти важкі сумки додому з магазину.

Вільний вибір. Люди, які відвідують Інтернет-магазини, мають повну необмежену свободу вибору. Сформована заявка надсилається не на склад звичайного роздрібного магазину, а на найбільший склад, де є величезний вибір товарів. В Інтернет-магазині ви можете прочитати керівництво користувача та переглянути фотографії товару, не залишаючи улюбленого крісла перед монітором комп'ютера. Звичайно, ви не можете пізнати товар «наживо», відчутти його руками, але є можливість зайти в звичайний магазин, подивитися улюблений товар.

Знову зважте всі «за» і «проти» і значно заощадите значну суму грошей, купуючи інтернет-магазин.

Комфорт. Відвідувачі інтернет-магазинів не вписуються в натовп інших покупців, з нетерпінням чекаючи прибуття безкоштовного менеджера. Там ніхто не заважає настирливим порадам і не змушує купувати непотрібні речі. Тиску продавця немає. Консультації з кожного питання проводяться через онлайн-консультацію.

Зручна сплата за покупку. Люди обирають швидкий спосіб оплати за товар: накладеним платежем, поштовим, банківським переказом, оплатою кредитною картою або електронними грошима. Вони можуть скасувати покупку, не

заплативши за відправку.

Анонімність. Ніхто не буде знати про покупки людей, які відвідують інтернет-магазини, якщо вони самі не хочуть ділитися цією інформацією. Тільки в Інтернеті - магазин гарантує повну анонімність покупця, адже Ви можете зареєструватися під будь-яким ім'ям..

Недоліки інтернет-магазинів полягають у відсутності процесу продажу товарів, який складається з:



Рис 1.2. Недоліки Інтернет – магазинів

Які саме недоліка, розглянемо: недотримання строку доставки товару;

Часто саме швидкість доставки є вирішальним фактором при виборі товару в інтернет-магазині. Якщо вам терміново потрібні товари, будь ласка, виберіть магазин, який пропонує швидку доставку.

- дорога доставка товару;

Ефективність служби доставки багато в чому визначає «обличчя» магазину. Але поки, на думку співробітників інтернет-магазинів, доставка товарів є одним із найслабших місць компаній.

Незручна платіжна система. Виявилось, що вони часто відмовлялись робити

покупки лише тому, що в Інтернет-магазині не пропонували зручної форми оплати.

Найпоширенішим способом оплати є оплата готівкою кур'єром. Це вважається найбезпечнішим способом.

Найекономічнішою формою оплати є передплата банківським переказом. Широка система замовлення. Розгалужена система замовлення - це довгий і складний процес закупівлі. Покупцеві доводиться заповнювати довгий і часто незрозумілий бланк, в результаті чого він проводить багато часу в Інтернеті, що не завжди дешево.

Ви також можете помітити, що деякі магазини завантажуються повільно, що може призвести до виходу з магазину.

Нестабільний асортимент. Тому магазин, який пропонує користувачам весь необхідний асортимент, зможе отримати перевагу над конкурентами.

Крім того, нестабільний асортимент деяких магазинів, тобто коли після вибору товару та здійснення покупки виявилось, що бажаного товару немає на складі або що магазин просто не зв'язався з покупцем.

Необхідність реєстрації. Деякі магазини змушують покупця пройти дуже тривалий покроковий процес реєстрації. Це стримує потенційних покупців від покупок у цьому магазині, особливо коли покупець оцінює витрачений час.

Проблема віртуальності. Неможливість доторкнутися і потримати придбаний товар (це чудово виглядає на фото, а насправді це зовсім інше).

Неможливість вибрати замовлення. Може заплутати колір, модель, зовнішній вигляд тощо.

Втрата замовлення. Така ситуація буває рідко, але ж буває, коли замовлення повністю втрачаються.

1.4 Постановка задачі

Мета роботи: розробити web-сайт магазину з продажу комп'ютерної техніки та базу даних «Інтернет-магазину».

Об'єкт дослідження: Інтернет-магазин з продажу комп'ютерної техніки

Предмет дослідження: База даних для магазину комп'ютерної техніки

Методи дослідження: Використання сучасних Інтернет-технологій, а саме мови програмування PHP для створення веб-сайту та системи управління базами даних MySQL для створення бази даних Інтернет-магазину, мова JavaScript.

Завдання дослідження:

1. Ознайомитись із особливостями інтернет-магазину, його позитивними та негативними рисами та визначити їх класифікацію;
2. Дослідити існуючі Інтернет-магазини та вивчити принципи їх побудови та функціонування;
3. Розглянути структуру інтернет-магазинів;
4. Провести маркетингові дослідження та розглянути економічні детермінанти створення Інтернет-магазину;
5. Проаналізувати сучасні засоби розробки для створення Інтернет-магазинів;
6. Розробити веб-сайт та базу даних Інтернет-магазину комп'ютерної техніки.

РОЗДІЛ 2 ПРИНЦИПИ ПОБУДОВИ ТА РОБОТИ ІНТЕРНЕТ МАГАЗИНІВ

2.1 Структура побудови Інтернет-магазину

Успішний веб-сайт є надзвичайно ефективним маркетинговим інструментом - він може привернути увагу вашої аудиторії. Як і будь-який інший інструмент маркетингу прямої реакції, він повинен спочатку заінтригувати відвідувача, а потім підштовхнути його до певних дій. Однак багато людей ігнорують цю особливість домашньої сторінки, тому відвідувачі часто не затримуються на ній довго і залишають її відразу після входу на неї. Ці сторінки, іноді з безліччю корисних порад та статей, майже ніколи не досягають очікуваного рівня відвідуваності, не кажучи вже про продажі. [6].

Загалом веб-сайт може стати більш надійним та ефективним інструментом. Сайт, повинен привернути увагу та викликати інтерес, важливо щоб він спонукав не лише переглядати товари та робити покупки, а й рекомендувати його іншим людям

Стандартний інтерфейс складає [9]:

- каталог;
- пошук;
- кошик;
- реєстрацію;
- форма відправлення замовлення.

Каталог - це багаторівнева структура даних, яка повинна бути простим і способом упорядкування товарів. Найпростіший спосіб представити такий каталог - це Дерево об'єктів, верхній рівень якого складається зі списку розділів.

Розділи можуть містити підрозділи або посилання на конкретний товар тощо. Такий макет просто необхідний для зручного та швидкого пошуку та замовлення товарів. Пошукова система. Пошукова система є обов'язковим елементом динамічного каталогу і реалізована на стороні сервера. Хоча каталог забезпечує впорядкування та групування даних, пошукова система дозволяє користувачеві швидко знаходити інформацію, що особливо важливо, коли каталог являє собою досить розгалужену структуру даних з великою кількістю розділів, підрозділів та

продуктів, користувач не знає, в якому розділі це може бути ... товар, який його цікавить, і чи є він взагалі в каталозі. У деяких випадках пошукова машина може значно зменшити кількість перетворень сторінок каталогів в інформацію.

Особливістю пошуку є те, що він відбирає всі записи, які відповідають умовам нашого запиту. У випадку великої вибірки даних результати пошуку видаються на сторінках, тому відвідувачам не потрібно довго чекати завантаження всієї вибірки, яка може складати сотні або тисячі. Зазвичай відвідувачі переглядають не всі зразки сторінок, обмежуючись двома-трьома. Тому ця пошукова система дуже повільна та неефективна у багатьох випадках. Однак це дозволяє вибрати однакові товари різних постачальників, порівняти їх параметри між собою і вибрати найкращий варіант.

Кошик. Кошик - це сукупність даних, яка використовується для зберігання продуктів, замовлених користувачем..

Реєстраційний формуляр. Він використовується для введення персональних даних користувачів. Надалі ця інформація використовується для їх ідентифікації між сесіями з Інтернет-магазином. Ця інформація може зберігатися як на стороні сервера, так і на стороні клієнта. Форма відправки замовлення. Форма відправки замовлення служить для введення контактної інформації замовника і відправки її і замовлення на електронний ящик організації.

Спочатку ми бачимо перелік товарів на складі. Оскільки Інтернет-лічильник, як правило, підключений до будь-якої з систем автоматизації компанії, цей список включає ті самі товари, що продаються в звичайних магазинах.

Зміст представляється у вигляді ієрархічної деревної структури з основними елементами груп об'єктів. Коли ви клацаєте на групі, вона розширюється, відкриваючи список підгруп або конкретних продуктів цього типу. Іноді покупець може побачити фотографію товару та його характеристики та додати його до кошика.

Після заповнення кошика клієнт видає команду «Завершити замовлення» та обирає зручний спосіб оплати. Якщо він робить покупки вперше, його зазвичай просять надати якусь інформацію про себе - ім'я, номер телефону, адресу тощо.

Корпоративний покупець вказує назву компанії, номер рахунку, повне ім'я та номер контактної особи. На цьому етапі покупцеві присвоюється спеціальний ідентифікаційний код. Це робиться для того, щоб наступного разу, коли він піде до магазину, всю вищезазначену інформацію можна пропустити - просто введіть свій код. Потім розрахунки і пряма доставка товару замовнику.

Виконати замовлення можна такими способами: кур'єром, звичайною поштою або авіапоштою через Україну, близьке та далеке зарубіжжя, вивіз автомобілів та інші способи доставки.

2.2 Аналіз платіжних систем

Звичайні способи сплати, включаючи готівку, банківські перекази, чеки та пластикові картки, були винайдені задовго до електронної комерції. Тобто такі методи не повністю задовольняють її потреби. Звичайно, купуючи звичайні «фізичні» товари, гроші можна стягнути з покупця при доставці. Але якщо ви отримуєте «цифровий» товар або інформацію, повинен бути спосіб оплати покупок безпосередньо на веб-сайті продавця. Тому в електронній торгівлі надзвичайно важливо розробити універсальний спосіб оплати покупок в Інтернеті, який би дозволив вам здійснювати дешеві та безпечні платежі в режимі реального часу. Мобільні платежі - це використання портативних мобільних пристроїв для зв'язку, розваг, отримання та передачі інформації, транзакцій через державні та приватні мережі.

В даний час на ринку існує величезна можливість оплачувати послуги або товари різних компаній за допомогою мобільного телефону. Ви можете робити покупки за адресою

Інтернет-магазини, оплачують рахунки за комунальні послуги і навіть перераховують гроші на банківські реквізити. Деякі мобільні оператори пропонують цю послугу безкоштовно як частину пакету для звичайних користувачів мобільних телефонів.

Гроші, які будуть використані власником стільникового телефону для придбання товарів або послуг, можуть бути зняті з рахунку власника телефону.

Якщо сума платежу перевищує залишок на картці, треба активувати іншу картку. Тоді невикористані залишки готівки з попередніх карток додаються до активованої картки.

Сплата за допомогою скретч-карт. Скретч-картки - це передплачені картки, оскільки, купуючи їх, ми робимо передоплату за послугу чи товар, який ми маємо намір замовити. Свою назву вони отримали від англійського слова «scratch». Саме наявність спеціального захисного шару, який приховує буквено-цифровий набір, відрізняє передплачені картки. Дуже важливо пам'ятати, що кожна передплачена картка має свій унікальний серійний номер, який буде використовуватися при оплаті послуг.

Вони активно використовуються компаніями мобільного зв'язку, постачальниками Інтернету та інше.

Сплата електронними коштами. Основою системи онлайн-платежів є цифрова валюта. Їх можна розрізнити за типом організації їх діяльності, рівнем безпеки, а також методом розрахунку.

Ви можете уявити цифрову валюту як файл, що символізує обмін готівкою. Продавці та покупці можуть вільно обмінювати ці «монети» на товари та послуги в Інтернеті. Для цього учасники системи встановлюють на свої комп'ютери спеціальну програму - «електронний гаманець», що дозволяє бронювати та передавати токени та перевіряти їх справжність. Цифрові гроші можуть циркулювати в мережі необмежено довго, але їх також можна в будь-який час конвертувати у реальні гроші організаторами системи або банками-учасниками.

Різні електронні валюти - це цифрові касові апарати, засновані на використанні сучасних карткових технологій. Смарт-карта - це невеликий комп'ютер із власним процесором, пам'яттю, програмним забезпеченням та системою вводу-виводу. У випадку з цим типом карток, окрім електронних грошей, зберігаються у тому сенсі, що описаний вище.

На сьогоднішній день смарт-картка використовується як стандартна банківська картка (так званий електронний гаманець), яка використовується для запису лише готівки або інформації про клієнта. Цифрові гроші, засновані на смарт-картці, можуть не тільки забезпечити необхідний рівень конфіденційності та конфіденційності, але й не вимагають підключення об'єкта для перевірки оплати, на відміну від подібних систем на базі персонального комп'ютера. Проблема в тому, що лише електронна картка для переказу грошей у комп'ютері, або ж назад до кардрідера вам потрібен спеціальний кардрідер, підключений до комп'ютера. Платіжні картки. Каарка и амаахда - зареєстрований платіжно-депозитний документ у формі дебетової картки, що надається банком людям, які вносять безготівкові кошти за товари та послуги в роздрібну мережу, оснащену комп'ютерними пристроями, що надсилають банківські платіжні запити.

Кредитна картка гарантує наявність у власника картки дійсного банківського рахунку. Кредитні картки американських кредиторів та фінансових груп, таких як Visa, Mastercard, American Express тощо, використовуються в різних країнах.

Кредитні картки можна використовувати, коли продукти та послуги, що надаються через Інтернет, так само, як і в банкоматі, щоб отримати готівковий журнал: гроші, які потрібно сплатити, клієнтам доведеться вказати номер картки та PIN-код.

Карти дій. Оскільки використовувати стандартну кредитну картку для прямих онлайн-платежів ризиковано, такі віртуальні картки, як VISA E-c @ rd та Virtual Mastercard, виготовляються лише для прямих платежів. VISA E-c @ rd базується на міжнародній технології платіжних систем VISA Int. Virtual Mastercard базується на міжнародній технології платіжних систем Mastercard Int.

Сьогодні різні банки вчасно випускають такі картки, оцінюючи їх переваги та недоліки. VISA Ec @ rd та Virtual Mastercard використовуються виключно для онлайн-оплати товарів та послуг в інтернет-магазинах по всьому світу, включаючи Україну, а також для послуг мобільних компаній, інтернет-провайдерів, туристичних агентств та готелів.

Покупки, а не електроніка, включають:

- видача грошового переказу;

- Попередня оплата безготівкових розрахунків;

Оплата кур'єрові готівкою. Один з найпростіших способів оплати - це оплата поштою. Все, що вам потрібно зробити, - це передати паперові гроші та підписати квитанцію. На жаль, цей спосіб не скрізь можливий.

Після вибору товару в Інтернет-магазині покупець робить замовлення, вказуючи адресу доставки (будинок, офіс). Лист доставляє товар за вказаною адресою та бере плату.

Переваги цього методу можна підсумувати як:

- Гарантований доступ до товарів (покупець) та грошей (продавець). Звичайно, тут можна обдурити, але можливості не більші за звичайний магазин.

- можливість швидко забезпечити повернення товару (та його завершення), не відмовляючи у покупці.

- подивіться в очі живій людині, попросіть її розповісти вам товар, показати роботу.

- отримати всі необхідні документи: гарантійний талон, підтвердження продажу. Недоліки також очевидні:

- не завжди приємно бачити у себе вдома (або в офісі) незнайомої людини (кур'єра), спілкуватися з ним.

- часто очікування кур'єра перетворюється на торттури, і відбиває всяке бажання що-небудь купувати.

Надання банківського рахунку без оплати в організації тепер можливо лише в Україні. Під час замовлення вкажіть назву організації. Ви зможете заповнити або роздрукувати рахунок-фактуру.

Ви можете оплатити замовлення в найближчому відділенні Ощадбанку (або в чомусь іншому). Для цього вам потрібно роздрукувати форму електронною поштою, інакше вона буде видана вам відразу після процесу замовлення (цю форму теж можна легко використовувати, наприклад, для заповнення банківської форми). Оплата через Ощадбанк коштуватиме 3% від ціни замовлення, що набагато дешевше і швидше, ніж переказ поштою чи телефоном. Як тільки гроші надходять на рахунок, магазини надсилають замовлення.

Оплата післяплатою. Оплачена покупка, отримана квитанцією поштового відділення, називається післяплатою. Коли ви платите поштою, пошта стягуватиме з вас 8% від ціни замовлення. Платіжна система така:

- замовлення відправляється за вказаною адресою, або заповнюється спеціальна форма оплати, надана Інтернет-магазином.

- ваше повне ім'я, повна адреса з індексом, номер контактної телефону та електронною поштою.

- ларкии хело дигнинта поштове відділення, поштове відділення перенесено витрати та витрати на замовлення реквізитів зазначеного банку.

- віддати квитанцію в банк від працівників пошти, замовлення видано.

РОЗДІЛ 3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ ТА ЙОГО ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Огляд використаної мови програмування PHP

PHP (PHP: Hypertext Preprocessor), раніше: Personal Home Tools - мова програмування, була розроблена для створення HTML-сторінок на стороні сервера. PHP може бути використаний як середній інструмент управління веб-сайтом (наприклад, Java, .NET, Perl, Python, Ruby). PHP підтримується більшістю хостинг-провайдерів. Проект, створений для PHP, є проектом з відкритим кодом [19,21].

PHP перекладає HTML-код веб-сервера, який передається клієнту. На відміну від текстових мов, таких як JavaScript, користувач не має доступу до коду PHP, що є перевагою безпеки, але значно послаблює взаємодію зі сторінкою. Але ніщо не зупиняє використання PHP для створення та коду JavaScript, вже реалізовані на стороні клієнта.

PHP - це мова, яка може бути вбудована безпосередньо в html-код сторінок, знову ж таки, що буде точно обробляти інтерпретатор PHP. Механізм PHP просто починає виконувати код після першої серії екранів (<?) і продовжує виконувати його, поки не зустрінє послідовність екранів (?>)

Різні типи функцій PHP дозволяють уникнути написання багатобуквених завдань на C або Pascal

Доступ до декількох баз даних:

- PHP створив бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, DBM, Hyperware, Informix, InterBase, Sybase.

- за допомогою стандарту Open Data Connection Standard (ODBC) ви можете підключитися до всіх баз даних, доступних драйверу.

Культура

Мова PHP, схоже, знайома з програмами, які працюють у різних сферах. Багато мовних будівель запозичені у C, Perl. PHP-код дуже схожий на код, який можна знайти в стандартних програмах на мові C або Pascal. Це значно зменшує початкові зусилля дослідження PHP. PHP може бути використаний як джерело інформації для зайнятих кількох днів і C, а також такий і як джерело інформації для талагу в Інтернеті, який може бути використаний як джерело інформації. Хоча PHP є молодою мовою, вона стала популярною серед веб-програм, які зараз є чи не найпопулярнішою мовою для створення веб-додатків (скриптів).

Наявність вихідного коду та безкоштовність

Стратегія з відкритим кодом та розповсюдження оригінальних сценаріїв для публічних програм, безсумнівно, мали благотворний вплив на багато проектів, особливо на Linux, хоча успіх проекту Apache значно зміцнив позиції шанувальників з відкритим кодом. Це також стосується історії створення PHP, оскільки підтримка користувачів у всьому світі була основним фактором розвитку проекту PHP. Прийняття стратегії з відкритим кодом та безкоштовне розповсюдження оригінальних PHP-скриптів забезпечили дуже цінну послугу для користувачів. Крім того, спільнота використовує відповіді PHP за умови надання свого роду послуг колективної підтримки, популярних електронних конференцій та надання відповідей навіть на найскладніші питання.

Ефективність є дуже важливим фактором програмування для багатьох середовищ користувачів, включаючи мережу. Головною перевагою PHP є те, що ця мова належить перекладачам. Це дозволяє запускати сценарії на дуже високій швидкості. За деякими оцінками, стільки скриптів PHP (особливо найскромніших) looga працює швидше, ніж аналогічні програми, написані за напружену пару днів. Однак, щоб розробники PHP цього не робили, зібрана реалізація працюватиме швидше - десятки, а іноді і сотні разів. Але продуктивності PHP достатньо для створення більш серйозних Інтернет-додатків.

Більш детальний опис мови. Що стосується системи друку, PHP - це мова програмування з активною машинкою. Немає необхідності чітко визначати тип змінних, хоча такі можливості існують. Посилаючись на змінну, інтерпретатор PHP інтерпретує її тип відповідно до контексту. За необхідності можна змінити змінну на інший тип за допомогою відповідної мовної структури. Це може знадобитися, якщо ви вважаєте, що значення змінної може інтерпретуватися по-різному залежно від типу. Також можна визначити тип змінної, що відповідає певному етапу написання. Зміна імен є делікатним.

До основних типів належать булеві дані, змішані числа та числа з плаваючою комою, а також рядки. Булеві дані показують істинність ціни. Комбінації можуть бути представлені вісімковою, десятковою та шістнадцятковою. Розмір інтегранку

може змінюватися залежно від сайту, часто розмір невеликої фракції становить 32. PHP не підтримує непідписані номери. Числа з плаваючою комою можуть бути задані десятковими або квадратними цифрами.

Робота з рядками. Стрічки поділяються на два типи - оглядові та неаналітичні. У першому класі перекладач перевіряє наявність посилань на інші змінні і, якщо такі є, значення значень змінюються, де це доречно. Рядковий символ може мати одне значення 256 значень, але можна працювати з багатобайтовими символами. Рядкові символи можна отримати, використовуючи подібний підхід до матеріалу підготовки. PHP надає широкий спектр функцій для пошуку та модифікації текстових каналів. Для цього використовують традиційний метод і конкретний метод, заснований на використанні стандартного виразу.

У той же час мова підтримує два типи безперервного вираження - Perl-walk та POSIX-walk, що відрізняються синтаксисом та особливостями.

Змішані типи даних. До змішаних типів належать фігури та елементи. Лінгвістична підготовка - це сукупність змінних, розділених на одну змінну. Вимога щодо відповідності заповнення форм не встановлена. Технічно організатори - це карти на вимогу, які відображають ключові значення в даних, що змінюються. Вміст значення, зазначеного ключем, може бути будь-яким, що може бути представлене як змінна. Інших обмежень немає, крім обсягу пам'яті, накладеного на кількість ключів у черзі.

Лінгвістичні відмінності - це відмова ділитися спільною формою площинних ключів. Також були впроваджені моделі різних комбінацій, не обмежуючи глибину свого гнізда. Корисною особливістю PHP є можливість підключення форм та дзвінків. Ці завдання дозволяють виконувати дії в одному або декількох пакетних форматах.

Також існує два типи конкретних даних - ресурсні та NULL. Ресурси - унікальна змінна, що складається із зовнішніх мережевих з'єднань. Ресурси створюються і використовуються для виконання конкретних завдань. Оскільки тип містить конкретні показники для відкриття файлів, посилань тощо, будь-яка дія зі значеннями ресурсів не має сенсу.

Сфера дії змінної - це визначене середовище. Існують місцеві та міжнародні змінні. Як правило, всі змінні мають локальну дію. Існує певний тип змінної - нерухома, яка дозволяє отримати змінну з певної частини коду, а змінна збереже отримане раніше значення. Існує також концепція суперглобальних змінних, які служать сховищем локальних даних або зовнішніх даних. Підтримується концепція динамічних змінних та функцій.

Константи в PHP — це простий ідентифікатор значення. Можна визначити постійний, після оголошення його буде неможливо змінити його значення або скасувати. Константа може мати лише значення ваги. Підтримується можливість отримання постійного значення для активного імені. Постійний розмір буде універсальним для тексту та всіх пов'язаних компонентів. Крім того, багато регулярних систем визначаються ядром мови.

Конструкції мови. Лінгвістичні оператори дозволяють виконати відповідні дії з одним або кількома операндами. Існує три типи операторів - одинарний, двійковий та потрійний. Оператори, як і інші мови, характеризуються не тільки діями, але також узгодженістю та пріоритетністю. Різниця між булевими операціями порівняння полягає у розділенні двох класів з огляду на тип і без урахування типу спаду відповідного типу, що відбувається. Цикли завжди невеликі. Мова реалізує конкретні робочі класи - впровадження, управління помилками та перевірка, пов'язана з класом.

Діяльність мовного перекладу базується на кодї, і можна призначити завдання та інші класи. Це ґрунтується на можливості умовного визначення роботи. У цьому випадку необхідно заявити про названу продуктивність, яка не потрібна в інших випадках. Неможливо перевизначити або припинити роботу. Результат повернення на роботу може бути будь-яким.

Мова реалізує взаємопов'язані функціональні можливості. Ви можете створити скільки завгодно імен, посилаючись на частину даних. Коли ви звільняєте один із заголовків, розділ даних залишається в пам'яті, поки текст не буде завершено або всі посилання звільнені.

Для завдань у PHP замість принципу надміру прийняття багатомовних функцій, що дозволяють змінювати певні параметри продуктивності залежно від типу та кількості переданих меж, використовується активний метод налагодження. Це дозволяє не тільки визначати кількість обмежень активності під час звітування, але і працювати з аргументами, отриманими під час виклику завдання.

В тілі роботи можна знайти кількість поданих аргументів та провести відповідні маніпуляції. Оголошуючи результативність стандартного методу, можна уточнити значення стандартного аргументу. Завдання можуть повертати лише одне значення, але обмеження можна перевищити, використовуючи не лише форми, але й посилання. Посилання не можуть подаватися під час виконання та оголошення роботи.

Після виконання нотаток збирач сміття очищає простір пам'яті, в якому вони знаходяться. Однак при необхідності можна виконати очищення пам'яті непотрібних компонентів даних під час виконання тексту. Користуватися послугами очищення пам'яті нечесно, хоча це можливо. Для побудови програмних пакетів можна використовувати структурований формат, виконуючи різне кодування. При необхідності можна підключити необхідні компоненти, а операція виконання може бути умовною. Файли, вкладені в текст, можуть відновити значення.

Додаткові можливості. Мова явно підтримує файли cookie HTTP відповідно до визначення Netscape. Це дозволяє встановлювати та читати невеликі фрагменти даних на стороні замовника. .PP надає можливість організувати взаємодію користувачів під час сеансів. Ви можете зберігати різноманітні дані, включаючи об'єкти, сеанси.

PHP має назву The PHP License version 3.0, ЧаП по цій ліцензії та її текст можна знайти на офіційному сайті The PHP Group, а також файл із ліцензією містить дистрибутив PHP (звичайно license.txt).

Історія PHP сягає 1995 року, коли Расмус Лердорф розробив простий додаток Perl, який аналізує відвідування користувачами резюме веб-сайту. Потім, коли цією програмою вже користувалося кілька людей, і кількість людей, які бажали

отримати до неї доступ, постійно збільшувалася, Лерддорф назвав свою персональну побутову техніку типу 1 та зробив її доступною безкоштовно. З тих пір він розпочав безпрецедентний підйом популярності PHP.

Як завжди, існує нагальна потреба в доопрацюванні та нових доповненнях. Для його реалізації Rasmus створює нову версію пакету, на даний момент написану мовою C. Отриманий пристрій отримує робочу назву PHP / FI (Домашня сторінка / Форми інтерпретації - Особиста сторінка / Персоналізована форма перекладача), після чого вона також буде називатися PHP 2 Цей тип вже схожий на сучасний PHP. Він мав конфігурацію та іменування змінних у мовному форматі Perl, можливість вставляти оператори PHP у html-код на сторінці, автоматичну інтерпретацію форм, взаємодію з інтеграцією баз даних. При цьому все працювало швидше, оскільки PHP збирався на веб-сервері Apache. Зез 1997 г. PHP є натоварен з логу істикмалай 50 (на (не більше 1% усіх веб-серверів).

В 1997 році Зеєв Сураскі та Енді Гутманс приєдналися до проекту PHP. Будучи студентом ізраїльського університету, вони намагалися використовувати PHP / FI як один з комерційних університетських проектів. У той же час вони зіткнулися з багатьма проблемами та обмеженнями з цією технологією. Вивчаючи вихідний код PHP 2, Ziv та Andy прийшли до висновку, що необхідно створити, а точніше, оптимальну роботу журналу PHP, особливо стосовно мови дизайну. Протягом кількох місяців вони добре адаптувались до цього завдання.

Коли вони закінчили роботу, Зів та Енді домовились з Расмусом співпрацювати над розвитком та розвитком мови. З тих пір з'явилася команда PHP - група однодумців, яка працює над розробкою технології PHP. Отриманий продукт був випущений в 1998 році під назвою PHP 3.

Головною особливістю PHP 3 була можливість розширити ядро, що залучило його до роботи з багатьма сторонніми розробниками PHP, що створюють власні модулі. Їх наявність дала PHP можливість працювати з великою кількістю баз даних, протоколів та підтримувати велику кількість API. Зез 1998 р., РНогато PHP наброював 100 000, PHP нараховується з 10% серверних послуг в Інтернеті.

У той же час публікація електронної преси та публікація книг про Водночас друк електронної преси та публікація наукових книг PHP сприяли широкому використанню цієї мови.

Відразу після виходу PHP 3 Енді Гутманс та Зів Сураскі почали працювати над ядром PHP. По-перше, потрібно було вирішити проблему зростання продуктивності праці. Новий продукт під назвою «Zend Engine» (представляє творців: Зеєва та Анді) успішно адаптував завдання і був реалізований в 1999 році. Основними ідеями, що реалізовувались, була здатність складати текст із примусовим виконанням, завдяки продуктивності якого можна здійснюється за великим замовленням.

PHP 4 і робоче ядро, випущений в 2000 році. Окрім підвищення продуктивності, PHP 4 вони зможуть підтримувати нову сесію, виробництво продуктів, систем та безпечну обробку даних, введений користувачем журналу, та нові форми мови . З виходом версії 4 PHP почали використовувати більш ніж на 20% веб-сайтів.

У період між 2000 і 2004 роками продовжувалася активна робота з удосконалення типу 4, але незабаром група PHP почала розглядати можливості нової версії. По-перше, було вирішено посилити можливості продукту мови, що дозволило використовувати його для великих проектів. Робота типу 5 зайняла багато часу, включаючи низку експертів із звукозаписів, серед яких Стерлінг Хьюз та Маркус Бургер.

Офіційний реліз PHP 5 вийде в липні 2004 року. По-перше, як і планувалось, весь процес роботи з об'єктами був перероблений. І якщо попередні версії цільових програм PHP були певною мірою можливими, саме тому вони не часто використовуються на практиці, то PHP 5 має чудовий потенціал для реалізації цільових програм. Крім того, PHP збагачений низкою дорогих розширень для роботи з XML, різними джерелами даних, вилученням зображень та багатьма іншими.

Всі основні бібліотеки, що працюють з XML, запозичені з PHP 4, були детально розглянуті. Популярні розширення, такі як SAX, DOM та XSLT, тепер використовують libxml2, що робить їх ще більш ефективними.

PHP 5 також включає два нові компоненти для інтеграції протоколів - SimpleXML і SOAP. SimpleXML значно спрощує роботу з XML-даними, відображаючи вміст XML-документа як PHP-об'єкт. Розширення SOAP дозволяє створювати PHP-сценарії, перетворюючи інформацію в інші програми, використовуючи XML-повідомлення про існуючі протоколи, оскільки модуль HTTP SOAP у PHP 5 надає розробникам інструменти для швидкого створення ефективних та ефективних клієнтів SOAP та SOAP сервера.

Нова система PHP 5 MySQLi (модифікована MySQL) розроблена для роботи з MySQL-сервером версії 4.1. 2 і вище, реалізуючи не тільки систему, а й об'єкт MySQL. Додаткові функції цього розділу включають - SSL, контроль транзакцій транзакцій, повторювану підтримку тощо. Очевидно, на PHP історія на цьому не закінчується. Вам слід очікувати наступних типів мови з розширеними функціями.

Базовий синтаксис. Фрагменти коду PHP також відображаються в HTML

Усі тексти впорядковано в числові блоки. Ці блоки можуть бути встановлені як HTML-код, але розділені відповідними обмеженнями. PHP-код у HTML повинен містити між оригінальними символами `<?php` та кінцевим `?>` (або між `<script language=«php»` та `</script>`)

Кращим варіантом розподілу коду PHP є `<? Php?>`, Оскільки ці початки та кінці дозволяють використовувати PHP-код у документах відповідно до правил XML. Ви також можете використовувати скорочений ввід: `<? >` Однак формат ASP не рекомендується, і, як очікується, він відсутній у PHP6.

Відокремлення інструкцій. Інструкції PHP сортуються; Не потрібно ставити крапку з комою перед кінцем тексту (перед`?>` Позначкою).

Коментарі в PHP можуть бути як однорядкові так і багаторядкові. Однорядкові коментарі починаються із `//` або `#`. Багаторядкові коментарі

починаються символами /* і закінчуються */. Слід зауважити, що однорядкові коментарі йдуть або до кінця текучого рядка або до закриваючого тега ?>

Змінні. Змінні є основою будь-якої програми. У PHP є три типи змінних:

- Коротка форма: в більшості випадків використовується змінна \$. Зміни, створені програмним кодом, такі. Цей метод також можна використовувати для створення змінної looga у формі diiwaanka_globals be kartiray файл конфігурації php.ini.

- Середній формат: \$ _POST ['variable'], \$ _GET ['variable'] ', \$ _REQUEST ['variable '] Використовується для отримання даних із форм FORM. Залежно від типу передачі даних в обох випадках встановлюється POST або GET або APPLICATION.

- Довгий формат:

\$ HTTP_POST_VARS ['змінна'], \$ HTTP_GET_VARS ['змінна'] Найповніший метод запису, принаймні, що використовується. У майбутньому він вважатиметься закінченим і тому непотрібним.

Визначені користувацькі змінні: \$ variable = 0.00;, \$ variable1 = 'привіт PHP-Master!' [19,21,24-26].

3.2 Огляд використаної бази даних MySQL

MySQL - безкоштовна система управління збором даних (СУБД). MySQL належить Sun Microsystems, яка розробляє та підтримує додаток. Розподілена загальна ліцензія GNU та її ліцензія на бізнес, необов'язкові. Крім того, MySQL АВ покращує функціональність ліцензованих користувачів, і завдяки такому типу команд технологія відтворення з'явилася майже в перших моделях [19,21].

MySQL - це рішення для малих та середніх програм. Введіть LAMB. MySQL часто використовується як сервер, доступний локальним або віддаленим клієнтам, але дистрибутив включає внутрішню бібліотеку, яка дозволяє додавати MySQL до стандартних програм.

Гнучкість баз даних MySQL підтримується великою кількістю типів таблиць: користувачі можуть вибрати як таблиці MyISAM, які підтримують повнотекстовий

пошук, так і таблиці InnoDB, які підтримують окремі транзакції на рівні реєстру. Крім того, дані MySQL постачаються з певним типом таблиць ПРИКЛАД, які показують принципи створення нових типів таблиць. Dhismahaагодарение на архітектуру та GPL-ліцензування відкрито, нові типи таблиць регулярно відображаються в СУБД MySQL.

Про походження MySQL. MySQL виникла як спроба застосувати mSQL до власної розробки компанії: таблиці, що використовуються для ISAM - системи низького рівня. В результаті було розроблено новий інтерфейс SQL-SQL, але інтерфейс API-API успадкований від mSQL. Походження назви «MySQL» невідоме.

Ліцензування. MySQL має дві ліцензії. MySQL можна розповсюджувати відповідно до вимог GPL. Але відповідно до вимог GPL, якщо будь-яка програма вимагає MySQL, вона також повинна розповсюджуватися за ліцензією GPL [6]. Однак це може відрізнитися від планів розробки, які не хочуть відкривати вихідний код для своїх програм. У таких випадках видається ділова ліцензія від MySQL AB, яка також забезпечує якісну сервісну підтримку.

Платформи. MySQL інтегрована з великою кількістю різних програм: AIX, BSDi, FreeBSD, HP-UX, GNU / Linux, Mac OS X, NetBSD, OpenBSD, OS / 2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare , Tru64, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Server 2003 та Windows Vista. У OpenVMS також є порт MySQL. Важливо зазначити, що MySQL AB забезпечує безкоштовне завантаження не тільки вихідного коду, але й компіляції та експлуатації спеціальних операційних систем, готових до роботи із завантажуваними компонентами, представленими на офіційному веб-сайті.

Мови програмування. MySQL може використовувати API на C, C ++, Eiffel, Java, Lisp, Perl, PHP, Python, Ruby, Smalltalk та Tcl ODBC MyODBC Driver.

Застарілий тип

Хоча версія 4.0 застаріла, вона все ще широко поширена. Основні особливості цього типу:

- буукінта бууксда в ANSI SQL-99, оо плюс розширення;
- структурна адаптація;
- Незалежні системи зберігання даних (MyISAM для швидкого читання, InnoDB для транзакцій та почесне посилення);
- підтримка SSL;
- запити на риболовлю;
- Повторення: один клієнт під одним менеджером, багато підлеглих під одним керівником;
- Повнотекстова індексація та пошук методом MyISAM;
- Реалізована база даних;
- дникод (UTF-8);
- таблиці InnoDB забезпечують відповідність вимогам ACID;
- На сервері, що дозволяє самостійно додавати програми MySQL.

Рекомендована версія 2005 року - MySQL 4.1, випущена 27 жовтня 2004 року, містить нові функції:Плутані запитання та дублікати таблиць;

Нова система запису та сортування;

Гнучкий та простий у використанні протокол клієнт-сервер з підтримкою підготовлених додатків, що забезпечує оптимальну продуктивність;

Нова програма установки й настроювання для Microsoft Windows і GNU/Linux;

Захищені через OpenSSL з'єднання клієнт-сервер;

Високо-оптимізована бібліотека, що може бути використана в сторонніх програмах;

- Повноцінна підтримка Юнікода (UTF-8 і UCS2);
- Стандартні просторові типи даних GIS, для зберігання географічної інформації;

- Поліпшений повнотекстовий пошук і система допомоги.

MySQL5.0

Функціональність MySQL 5.0 робить MySQL подібним до комерційних СУБД значно розширеним. Якщо СУБД MySQL спочатку звинувачували в недостатній підтримці стандарту SQL, то з появою п'ятої частини цієї популярної бази даних була майже повна підтримка на рівні SQL. MySQL 5.0 включає наступні результати:

- збережені системи та функції;
- детектори помилок;
- прокляття;
- ; ісіала;
- толко бандитичен;
- інформаційний план (так званий системний словник).

MySQL5.1

MySQL версії 5.1 продовжує шлях до рівня SQL2003. MySQL 5.1 містить нові функції:

- Графік роботи;
- кала кайбсананта;
- змінено поведінку ряду операторів, щоб забезпечити більшу відповідність стандарту SQL2003;
- суттєві зміни в роботі команди MySQL, такі як, наприклад, можливість зберігати дані на робочих столах;
- Виробництво на основі послідовності;
- Впровадження повнотекстового формату підручника.

MySQL6.0

Версія MySQL 6.0 все ще проходить альфа-тестування. Спочатку було вирішено створити версію 5.2, але незабаром цю версію було змінено на 6.0.

Одним із головних нововведень версії 6.0 є новий формат зберігання Falcon, розроблений MySQL AB як заміна InnoDB InnoDB, придбаний Oracle. Повний перелік змін можна переглянути на офіційному веб-сайті.

Локалізація. Починаючи з версії 4.1, у дані MySQL була введена нова система сортування та сортування. Якщо ви використовуєте програму cp1251, перед виконанням оператора SQL потрібно налаштувати посилання на сценарій, використовуючи такі оператори:

```
ВСТАНОВИТИ character_set_client = 'cp1251';
```

```
ВСТАНОВИТИ character_set_results = 'cp1251';
```

```
ВСТАНОВИТИ character_set_connection = 'cp1251';
```

Ці три оператори еквівалентні виклику одного оператора:

```
ВСТАНОВИТИ ІМЕНІ 'cp1251'
```

The_client_part_print змінює кодування коду даних, надісланих від клієнта, змінна поведінка_set_results встановлює кодування даних, що надсилаються атрибутами клієнта, сервером. При використанні Unicode UTF-8 це твердження виглядає так: ВСТАНОВИТИ ІМЕНІ 'utf8'

Основні можливості MySQL

Далі наведено перелік найважливіших функцій програм баз даних MySQL:

- Написано на C та C ++.

- Перевірено на широкому діапазоні компіляторів

- Працює на багатьох різних платформах.
- Доступні API для C, C ++, Eiffel, Java, Perl, PHP, Python, Ruby та Tel.
- Абсолютно більшість зараз використовують кабелі ядра. Може працювати на більш ефективних системах.
- Надає орієнтовані на клієнта методи та методи зберігання.
- Використовує швидкісні таблиці дисків (MyISAM) з двійковою навігацією на основі дерева (B-Trees).
- Порівняно легко додати до інших методів зберігання. Це корисно, якщо ви хочете додати інтерфейс SQL до своїх даних.
- Дуже швидка система розподілу пам'яті на основі потоку.
- Дуже швидкі з'єднання за допомогою декількох інтегрованих з'єднань.
- Таблиці, що використовуються для тимчасових таблиць.
- Функціонал SQL реалізований за допомогою високошвидкісної бібліотеки класу. Як правило, після запиту не призначаються нагадування.
- Код MySQL був протестований із використанням засобів пам'яті, комерційних засобів та засобів відкритого коду.

Сервер доступний як окрема програма для використання в середовищі підключення клієнт-сервер. Крім того, він також постачається з бібліотекою, яка може розміщувати окремі програми. Такі програми можна використовувати в ізольованому середовищі або в мережі без доступу до мережі.

- Різновиди стовпців
- Багато типів даних для стовпців таблиць: знакові/беззнакові цілі довжиною в 1, 2, 3, 4 і 8 байт; типи FLOAT, DOUBLE, CHAR, VARCHAR, TEXT,

BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM і просторові типи OpenGIS.

- Запис фіксованої та змінної довжини.

- Оператори й функції.

- Повна підтримка операцій і функцій у конструкціях SELECT і WHERE запитів, наприклад: `mysql> SELECT CONCAT(first_name, ' ', last_name) -> FROM citizen -> WHERE income/dependents > 10000 AND age > 30;`

- Повна підтримка конструкцій GROUP BY і ORDER BY. Підтримка групових функцій (COUNT (), COUNT (DISTINCT . . .), AVG(), STD(), SUM(), MAX(), MIN() I GROUP_CONCAT()).

- Підтримка LEFT OUTER JOIN і RIGHT OUTER JOIN як із синтаксисом SQL, так і із синтаксисом ODBC.

- Підтримка псевдонімів для таблиць і стовпців, як того вимагає стандарт SQL.

- Оператори DELETE, insert, REPLACE і UPDATE повертають кількість рядків, які були змінені. Замість цього можна задати повернення кількості рядків, що відповідають запиту, для чого буде потрібно встановити відповідний прапор при підключенні до сервера.

- Специфічна для MySQL команда SHOW може бути використана для добування інформації про бази даних, таблицях і індексах. Команда EXPLAIN дозволяє переглянути, як оптимізатор виконує запит.

- Імена функцій не конфліктують з іменами таблиць і стовпців. Наприклад, ABS — абсолютно коректне ім'я стовпця. Єдине обмеження, що накладається на виклик функцій, - це те, що між ім'ям функції й наступної за ним відкриваючою дужкою.

- Можна змішувати таблиці з різних баз даних в одному запиті (як в MySQL 3.22).

- Безпека.

- Система, заснована на пароліях і привілеях, є винятково гнучкою і безпечною й дозволяє організувати верифікацію засобами хоста. Паролі захищені, оскільки весь трафік паролів під час з'єднання із сервером шифрується.

- Масштабованість і обмеження.

- Підтримує роботу баз даних величезних обсягів. Наприклад, компанія MySQL AB застосовує сервер MySQL для обслуговування бази даних, утримуючих 50 мільйонів записів. Відома також організація, що використовує сервер MySQL для обслуговування бази даних з 60 000 таблиць, що зберігає близько 5 мільярдів записів.

Дозволяється мати до 64 індексів на таблицю (у версіях, що передують MySQL 4.1.2, допускалося до 32 індексів).

Кожний індекс може містити від 1 до 16 стовпців або частин стовпців. Максимальна ширина індексу становить 1000 байт (500 байт у версіях, що передують MySQL 4.1.2). Для індексу може застосовуватися префікс стовпців з типами CHAR, VARCHAR, BLOB і TEXT.

Мережна зв'язність. Клієнти можуть підключатися до сервера MySQL за допомогою портів TCP / IP на будь-якій платформі. У системах Windows NT (NT, 2000 або XP) клієнти можуть підключатися за допомогою призначених каналів. У системах на базі UNIX клієнти можуть підключатися до базових файлів домену UNIX. Інтерфейс ODBC дозволяє MySQL підтримувати клієнтські програми, що використовують ODBC-Connection. Наприклад, ви можете використовувати MS Access для підключення до сервера MySQL. Клієнтське програмне забезпечення може працювати в Windows або UNIX. Код з'єднання / ODBC доступний. Всі функції ODBC 2.5 підтримуються, як і багато інших.

Інтерфейс JDBC дозволяє MySQL взаємодіяти з клієнтськими програмами Java за допомогою JDBC Connections. Клієнтське програмне забезпечення може працювати в Windows або UNIX. Код посилання / JDBC доступний.

Локалізація. Сервер може надсилати повідомлення про помилки клієнтам різними мовами. Повністю підтримується кілька кодеків, включаючи latin1 (ISO-8859-1), німецьку, big5, ujis та інші. Наприклад, скандинавські символи, такі як «а»,

«a» і «b», можуть використовуватися для імен таблиць і стовпців. Починаючи з MySQL версії 4.1, також надається підтримка Unicode. Усі дані зберігаються у вибраній конфігурації. Усі порівняння звичайних рядків є чутливими. Сортування здійснюється відповідно до обраного методу (як метод використовується шведська) Ви можете змінити це під час запуску сервера MySQL. Наприклад, якщо класифікація проведена в достатній мірі, рекомендується звернути увагу на номер класифікації в чеській мові. Сервер MySQL підтримує безліч різних додатків і може вказувати як час збору, так і час роботи.

Клієнти та обладнання. MySQL Server має підтримку структури операторів SQL для перевірки, редагування та отримання таблиць. Ці оператори можна виконати у форматі командного рядка за допомогою клієнтської програми `mysqlcheck`. MySQL також включає `myisamchk`, дуже швидко командну команду для виконання тих самих операцій на робочому столі MyISAM. Всіма програмами MySQL можна керувати за допомогою `-help`.

Плюси. Незалежність від збору приватних даних. Хоча в тексті є діалекти та варіації, більшість текстових SQL-запитів, включаючи файли `ddl` та `DML`, можуть легко шукати базу даних і передавати її іншій. Існують системи, розроблені до того, як їх змусили використовувати мінімальну кількість даних (наприклад, система електронних документів Documentum може працювати як з Oracle Database, так і з Microsoft SQL Server та IBM DB2).

Наявність стандартів Наявність стандартів та типів тестів для визначення сумісності та послідовності конкретної реалізації SQL до загальновизнаного рівня лише сприяє «стабілізації» мови.

Сортування За допомогою SQL програма описує лише дані, які потрібно витягти або змінити. Як це зробити, безпосередньо визначається базою даних при обробці запиту SQL.

Недоліки

Несумісність формату контактних даних. Model Relational Creator, Едгар Кодд, Крістофер Дат та їхні прихильники зазначають, що SQL не є справжньою мовою. Зокрема, вони вказують на такі проблеми з SQL:

- Невстановлене значення (нулі)
- Чітке відображення команди зліва направо
- Стовпи, які не мають однакових назв та назв черг
- Домашня допомога "="
- Використовуючи інструкції
- Висока пильність

Опубліковані Крістофером Датою та Х'ю Дарвенем, вони встановлюють принципи для наступного покоління СУБД та надають мову підручника D, яка дійсно актуальна. [19,21,24-26].

3.3 Організація діяльності Інтернет-магазину

Створення Інтернет-магазину пов'язане з багатьма проблемами [20]:

По-перше, необхідно розробити цей проект. По-друге, необхідно просувати пошукові системи, що займає багато часу, але в основному це важливо, адже вам потрібно бути скрізь на перших сторінках, інакше потенційні клієнти просто не зможуть вас знайти..

Крім того, слід мати на увазі, що тут, як і в реальному житті, існує дуже складна конкуренція, тобто вам потрібно ретельно продумати свою стратегію роботи, щоб виграти цей конкурс.

Вам потрібно швидко підготуватися до доставки замовлення покупця, що зробити непросто, особливо працюючи в містах тощо.

Як бачите, проблем вистачає. Але нічого у цьому світі не трапляється так, і щоб бути успішним підприємцем, потрібно багато працювати.

У цей стрімкий вік комп'ютерних технологій комп'ютери стали важливою особливістю майже кожного будинку та, звичайно, офісів, тому цей тип бізнесу з самого початку приречений на успіх, однак зусилля повинні змінити

кожен. Крім того, в цій галузі відбуваються постійні нововведення, що комп'ютерні деталі оновлюються та вдосконалюються, коли ці люди в потрібний час у цій ситуації і відкрили власний бізнес, пов'язаний з торгівлею такою продукцією, не помиляються.

Але завжди потрібно цінувати щось більше, досягти нової висоти, єдиним шансом такої унікальної ситуації є створення Інтернет-магазину. Це відкриває велику надію для підприємців у всіх сферах діяльності та надає можливість для розширення бізнесу та кар'єрного зростання.

Спробуємо визначити основні елементи, які нам потрібно створити в Інтернет-магазині:

1. Розробка функціонуючої структури Інтернет-магазину (визначення основних принципів роботи)
2. Розробка структури сайту і схем сторінок
3. Створення дизайн-макета
4. Підготовку прав на різні типи операцій на сайті для груп користувачів
5. Створення необхідної кількості розділів і інформаційних блоків
6. Розробку структури товарного каталога
7. Налаштування експорту і імпорту даних
8. Узгодження необхідних характеристик товарів для порівняння і виводу
9. Налаштування форм замовлення і зворотного зв'язку
10. Обробка зображень, наповнення сайту.

Велику увагу при створенні інтернет-магазину слід приділити розвитку інтеграції. Згуртованість - у широкому розумінні - це спільна межа між незалежними об'єктами. Зв'язування визначає межі, закономірності та характеристики взаємодії об'єктів.

Взаємодія з користувачем - об'єкти та компоненти програми, здатні впливати на взаємодію користувачів у програмі. Зокрема:

- засоби відображення інформації, інформація, що відображається, формати і коди;
- командні режими, мова користувач-інтерфейс;

- пристрої і технології введення даних;
- діалоги, взаємодія і транзакції між користувачем і комп'ютером;
- зворотний зв'язок з користувачем;
- підтримка ухвалення рішень в конкретній наочній області;
порядок використання програми і документація на неї.

Є декілька простих правил, що дозволяють інтерфейсу магазину бути зрозумілим клієнтові [18].

Чим простіше, тим краще. Це не означає, що є лише текст і контактна інформація, але на сайті не повинно бути марної інформації, шрифт повинен бути зручним для читання. Зміст діаграми має бути чітким, виразним і швидко протікати. Неприпустимо, що анімація та звук використовуються на веб-сайті магазину, що займає тривалий час для завантаження та відволікання покупця.

Людське око шукає сторінки зверху вниз. Фокус зосереджений на верхньому лівому куті сторінки. Тому вгорі сторінки, як правило, знаходиться найважливіша інформація: назва компанії, логотип, назва сайту тощо. Настільки простим, як і верх сторінки, легко запам'ятати назву сайту та саму компанію.

Загальною помилкою багатьох сторінок є збирання на поверхні авангардної сторінки та композиція багатьох зображень, які включають назву компанії, яку неможливо помітити видно, та назву сайту. Як результат, людське око, скануючи цю частину веб-сторінки, не встигає визначити та запам'ятати необхідну інформацію. Відбувається перенасичення. Замість збільшення пам'яті таких сторінок зменшується пам'ять найважливіших текстів (картинок), назви компанії та назви сайту.

Простота пошуку потрібного товару. Покупець повинен мати легкий доступ до товару, який його цікавить, і мати можливість отримати повну інформацію про нього (опис у друкованому вигляді та кілька фотографій). На вітрині бажано розміщувати дані про солідну активність продавця.

Поширення групової продукції. Предмети слід розділити на групи. Необхідно мати можливість шукати товар за назвою та описом.

Високоякісне замовлення. Замовляючи замовлення, покупець повинен ввести контактну інформацію, після перевірки даних клієнт повинен отримати підтвердження замовлення.

Платежі Розрахунок ціни та ціни емісії повинні здійснюватися в національній валюті. Клієнт повинен вибрати спосіб отримання замовлення самостійно.

У електронному магазині можуть бути і інформаційні розділи:

- з даними про магазин (сфера діяльності, адреса, контактні телефони і так далі);

- новини магазину;

- статті (системи управління новинами і статтями надають можливість використовувати Інтернет-магазин як справжній інформаційний портал);

- інша корисна інформація.

Збільшенню потоку людей Інтернет-магазину та кількості постійних клієнтів сприяла також присутність менеджера по роботі з клієнтами. Важливо, щоб менеджери реагували на електронні листи якомога швидше.

Натомість, невидима для покупця, сторона Інтернет-магазину - це система управління. Вхід в систему адміністрування виконується лише після входу та пароля адміністратора. Менеджер має можливість повністю управляти вмістом інтернет-магазину:

- додавати або видаляти товари, описи і фотографії до них, змінювати їх вартість, умови доставки товарів і рівень знижок;

- редагувати розділи магазину;

- редагувати контактну інформацію Інтернет-магазину;

- редагувати зміст заголовків і текстів листів, що відправляються покупцеві при реєстрації і покупці товару;

- проглядати історію замовлень і статистику покупців та інше.

Керуючись даними принципами розробки інтерфейсу, було вирішено зробити ставку на простоту та інформативність, що б користувач, потрапляючи на сайт, міг отримувати чітку інформацію про товар.

Так само про те, як він зможе сплатити замовлення, які умови і терміни доставки і так далі.

Отже функціонуючу структуру Інтернет-магазину представимо наступним чином (Рис.4.1):

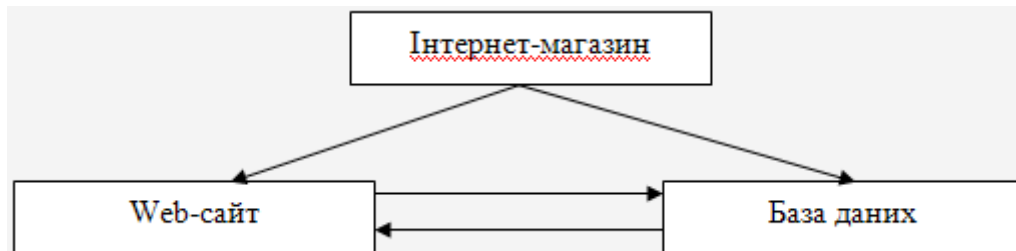


Рис.4.1. Функціонуюча структура Інтернет-магазину

Загальна інформація про Інтернет-магазини, описана в розділах 1 та 2, описана вище програма програмного забезпечення цього магазину. Фрагмент коду, див. Додаток



Рис.4.2. Структура бази даних

База даних «Інтернет-магазину» складається з 11 таблиць (структуру бази даних можна переглянути на Рис.4.2):

- Товари;
- Групи товарів;
- Групи компаній;
- Замовлення;

- Замовлення товару;
- Замовник;
- Спосіб оплати;
- Міста;
- Виконавець;
- Малюнок товару;
- Користувачі.

З наведеної структури бази даних (Рис.4.2) видно, що всі таблиці пов'язані зв'язками один-до багатьох. Дана схема зв'язків забезпечує цілісність бази даних.

Структуру Web-сайту Інтернет-магазину представимо наступним чином:

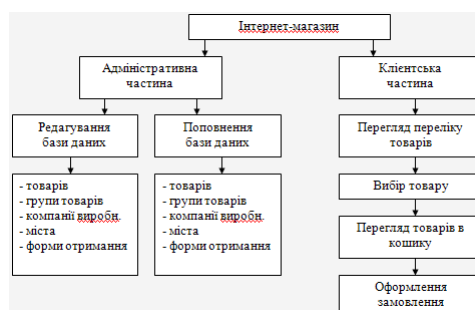


Рис.4.3. Структура Web-сайту Інтернет-магазину

Із структури видно (Рис.4.3), що в Інтернет-магазині пропонується 2 робочих частини: адміністративна та клієнтська.

Адміністративна частина дозволяє адміністратору Інтернет-магазину (після входження в систему адміністрування) здійснювати редагування бази даних та внесення додаткових елементів (товарів) в базу даних. Клієнтська частина даної можливості немає, тобто доступ до ресурсів Інтернет-магазину закритий для загального огляду.

Клієнтська частина доступна всім користувачам і відвідувачам Інтернет-магазину. Дозволяє клієнтам здійснювати перегляд, пошук, вибір потрібного товару, який розміщений в різних групах. Це значно полегшує роботу користувача при пошуку певної інформації. Клієнт має змогу оформити замовлення товару, вибрати тип оплати та доставки товару.

3.4 Опис роботи Web-сайту «Інтернет-магазин»

1. Задля функціонування Інтернет-магазину, треба звернутись до компаній, що надають послуги зареєстрації Інтернет-магазину.

При завантаженні Web-сайту перед нами відкривається головна сторінка (Рис.4.4):



Рис.4.4 Головна сторінка Web-сайту «Інтернет-магазин»

На цій сторінці є основні реквізити Інтернет-магазину, меню користувача (Рис.4.5), категорії (Рис.4.6), та кількість обраних товарів в кошику (Рис.4.7).

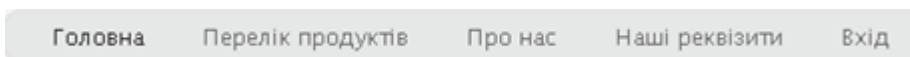


Рис.4.5. Меню користувача

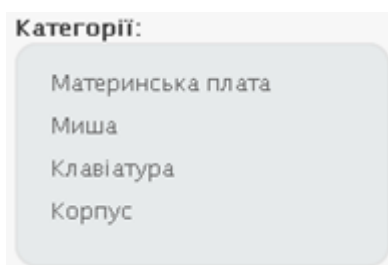


Рис.4.6. Категорії товарів

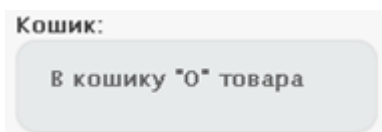


Рис.4.7. Кошик користувача

Вхід в розділ адміністрування сторінки здійснюється натисканням пункту GALKA в меню користувача. Адміністратора просять перевірити особисті дані (логін та пароль) (рис. 4.8). Якщо ви правильно вставили дані в мікшер і пароль, ви можете увійти у вікно журналу з наступними характеристиками, відкриваючи керівник Інтернет-магазину:

- перегляд товарів в базі (Рис.4.9);
- додавання нового товару (Рис. 4.10);
- редагування товару - при виборі опції редагувати - в режимі перегляду вмісту товарів в базі (Рис.4.11);
- перегляд груп товарів та їх редагування (Рис.4.12);



Рис. 4.8. Вхід в систему



Рис.4.9. Перегляд вмісту товарів в базі



Рис.4.10. Додавання нового товару в базу



Рис.4.11. Редагування існуючого товару в базі



Рис.4.12. Перегляд груп товарів

- Додавання нової групи товарів в базу (Рис.4.13);



Рис.4.13. Додавання нової групи товарів в базу

- перегляд компаній-виробників та їх редагування (Рис.4.14);



Рис.4.14. Перегляд компаній-виробників

- Додавання нової компанії-виробника (Рис.4.15);



Рис.4.15. Додавання нової компанії-виробника

- Перегляд існуючих форм отримання замовлення т(Рис.4.16);



Рис.4.16. Перегляд існуючих форм отримання замовлення

- Додавання нової форми отримання замовлення (Рис.4.17);



Рис.4.17. Додавання нової форми отримання замовлення

Для виходу з адміністративної частини в клієнтську на меню користувача необхідно натиснути кнопку ВИХІД.

Вхід в розділ клієнта здійснюється автоматично при заповненні Інтернет-магазину. Користувач може шукати потрібний йому товар за КАТЕГОРІЯМИ, що значно пришвидшить його роботу. Після вибору потрібної категорії відкривається

вікно зі списком товарів та коротким описом товару та його ціною (ціна всіх товарів вказана в національній валюті) (рис.4.18).

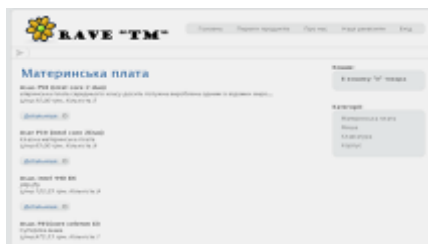


Рис.4.18. Перегляд відвідувачем переліку товарів в базі

Під кожним товаром знаходиться слово детальніше - , натиснувши на нього, ви зможете знайти більш повну інформацію про вибраний товар, також вставте цей товар в машину і поверніться до списку товарів, вибравши опцію НАЗАД (Рис.4.19).



Рис.4.19 Занесення вибраного товару в кошик

Після огляду користувач може подивитись обрані товари в кошику і відредагувати дані, якщо це потрібно (Рис.4.20).



Рис.4.20. Перегляд вибраних товарів в кошику

Для оформлення замовлення, користувачу необхідно вибрати опцію ЗАМОВИТИ, після чого йому надається невелика за розміром форма для заповнення основних даних (Рис.4.21).

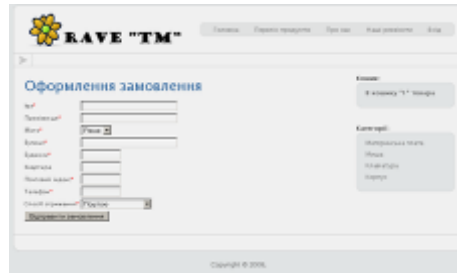
The image shows a web browser window displaying the 'Оформлення замовлення' (Order Form) page for 'RAVE "TM"'. The page has a light blue header with the company logo and navigation links. The main content area contains a form with the following fields: 'Ім'я' (Name), 'Прізвище' (Surname), 'Електронна пошта' (Email), 'Адреса' (Address), 'Телефон' (Phone), and 'Спосіб доставки' (Delivery method). A 'Зробити замовлення' (Place Order) button is located at the bottom of the form. On the right side, there is a sidebar with a 'Категорії' (Categories) section listing 'Аксесуари та інші', 'Музика', 'Підручники', and 'Іграшки'. The footer of the page contains the text 'Copyright © 2008'.

Рис.4.21. Оформлення замовлення

Після підтвердження замовлення в базі даних Інтернет-магазину відбувається автоматичне редагування даних, що відповідають вибраному клієнтом товару.

ВИСНОВКИ

В результаті виконання дипломної бакалаврської роботи було розроблено базу даних інтернет магазину з продажу комп'ютерної техніки. З його допомогою користувачі зможуть отримувати необхідну інформацію про товар, а також можливість купівлі товару.

Досліджено предметну область, а саме класифікацію, особливості, переваги та недоліки інтернет магазинів. Описано структуру розробки Інтернет-магазину, умови його функціонування та заходи, які слід виконати.

Визначає мову програмування та базу даних для розробки Інтернет-магазину. Також описується, як користуватися цим Інтернет-магазином.

Впроваджено програмне забезпечення в інтернет-магазині.

Правильний підбір усіх необхідних аспектів сучасних програмних продуктів був розроблений Інтернет-магазином, який повністю відповідає зазначеним критеріям та особливостям. Він має дружню взаємодію, можливість оплати та доставку користувача, невелику форму замовлення, пряму зміну даних магазину при замовленні товарів, сортування груп у групи, що пришвидшує пошук бажаного товару.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Разработка Web - приложений с помощью PHP и MySQL, 3-е издание. Люк Веллинг, Лора Томсон. Вільямс, 2005
2. Фролов А.В., Фролов Г.В. Глобальные сети компьютеров. Практическое введение в Internet, E-mail, FTP, WWW, и HTML, программирование для Windows Sockets. - Диалог - МИФИ, 1996. Ковалев А., Курдюмов И и др. Управление проектом по созданию интернет-сайта - М.: Альпина Паблишер, 2001. - 337 с.
3. Энди Харрис. PHP/MySQL для початківців. Кудиц-образ, 2005.
4. Internet, Сервіси, HTML і web дизайн. Глинський. Деол, 2000.
5. PHP 5.0 для начинающих, или как создать динамичный web-сайт. - 2-е изд., дополненное и исправленное. Леонтьев Б. Новый издательский дом, 2006.
6. WEB-дизайн Руководство пользователя. Под ред. Леонтьева. Познавательная книга, 2000.
7. WEB. Дизайн и коммерция. Конспект программиста. Под ред. Цеховой. Наука и техника, 2000.
8. Гизберт Дамашке PHP и MySQL; ИТ Пресс - М., 2016
9. Дакетт, Джон HTML и CSS. Розробка та дизайн веб-сайтов (+ CD-ROM) / Джон Дакетт. - М.: Эксмо, 2013
10. Зандстра аМ. PHP. Об'єкти, шаблони та методики програмування./М. Зандстра— М: Вильямс, 2011
11. Ловейн П. PHP 5. ООП. / П.Ловейн. – М.:ИТ Пресс, 2007
12. Дронов, Володимир PHP, MySQL и Dreamweaver MX 2004. Розробка інтерактивних Web-сайтів / Володимир Дронов, 2015
13. Віктор Гольцман, MySQL 5.0. Бібліотека програміста.

14. Бенкен Елена PHP, MySQL, XML. Программирование для Интернета; БХВ-Петербург - М., 2017.
15. Энсор, Д.; Стивенсон, Й. Проектирование баз данных; Киев: ВНУ, 2010. – 560.
16. Принципи побудови та сфери застосувань баз даних [Електронний ресурс] – Режим доступу до ресурсу: tdmuv.com/kafedra/internal/informatika/classes_
17. Принципи проектування програм [Електронний ресурс] – Режим доступу до ресурсу: https://allref.com.ua/uk/skachaty/Principi_proektuvannya_program.
18. Путівник мовою програмування Python [Електронний ресурс] / Олександр Мізюк. – 2020. – Режим доступу до ресурсу: <https://pythonguide.rozh2sch.org.ua/>.
19. Создание базы данных и таблиц [Електронний ресурс] – Режим доступу до ресурсу: <https://younglinux.info/sqlite/create>.
20. Як працює Веб [Електронний ресурс] // MDN. – 2016. – Режим доступу до ресурсу: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/How_the_Web_works

ДОДАТОК ЛІСТИНГ ПРОГРАМИ

```
index.php
<?php
//=====================================================
include(«php/sesion.php»);
include(«php/gui.php»);
include(«php/savepic.php»);
include(«php/edit_tovar.php»);
include(«php/guest.php»);
include(«php/zamov.php»);
include(«php/main.php»);
//=====================================================
//upload_pic(«index.php»,»mytov»);
start_d();
menu_top($_GET['main']);
if(isset($_GET['main']))
main_column(main_login());
if($_SESSION['_USER']==«guest»)
{
if(isset($_GET['zamov']))
main_column(main_zamov());
else
if(!isset($_GET['main']))
main_column(main_shov());
side_column(«<strong>Кošик:</strong>«.left_menu(return_menu_koshik()).
«<strong>Катеропii:</strong>«.
left_menu(return_menu_group_t()));
}
else
{
if($_SESSION['_USER']==«admin»)
{
main_column(main_tov());
side_column(left_menu(ret_menu_tovar()));
}
}
//side_column(left_menu($m_e11));
//side_column(left_menu($m_e11));
//side_column(«fdsjkhfksdhfkjdh jfsdgfjs gd dslhgfkshf kjsdhfk sd»);
end_d();
?>
guest.php
<?php
```

```

function return_menu_group_t()
{
    $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
    if(!empty($adb))
    {
        if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
        {
            $aSql=«select * from group_tovar»;
            $aRes=mysql_query($aSql,$adb);
            if($aRes==true)
            {
                $i=1;
                while($aRow=mysql_fetch_array($aRes))
                {
                    // $m_el1 [1][1]=«index.php?tovar=0»;
                    // $m_el1 [1][2]=«Товар»;
                    $ret[$i][1]=«index.php?show=«. $aRow[«id»];
                    $ret[$i][2]=$aRow[«name_group»];
                    $i++;
                }
                mysql_free_result($aRes);
            }
        }
        mysql_close($adb);
        return $ret;
    }
}

function show_guest_tovar($id_g)
{
    $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
    if(!empty($adb))
    {
        if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
        {
            $aSql=«select id,name_group from group_tovar where id=«. $id_g;
            $aRes=mysql_query($aSql,$adb);
            if($aRes==true)
            {
                if($aRow=mysql_fetch_array($aRes))
                {
                    $ret=$ret.»<table class='contentpaneopen'>

```

```

<tbody>
<tr>
<td class='contentheading' width='100%'>«.$aRow[«name_group»].»</td>
</tr></tbody>
</table>«;
}
}
mysql_free_result($aRes);
$aSql=«select
tovar.id as id,
tovar.name_tovar as name,
tovar.amount as amount,
tovar.count as count,
tovar.description as description,
group_compani.name_compani as compani_t
from tovar,group_compani where
tovar.id_group=«.$id_g.» and tovar.id_compani=group_compani.id»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
while($aRow=mysql_fetch_array($aRes))
{
if(strlen($aRow[«description»])>80)
$desc=substr($aRow[«description»],1,80).»...»;
else
$desc=$aRow[«description»];
$ret=$ret.»<div class='module'><div><div><div>«;
$ret=$ret.» <strong>«.$aRow[«compani_t»].» «.$aRow[«name»].»</strong>
<br>
«.$desc.»<br>
<i>Ціна:». $aRow[«amount»].» грн. «.» Кількість:». $aRow[«count»].»</i>
<a class='readon' href='index.php?detail=«. $aRow[«id»].
«'>Детальніше</a>
</div>
</div>
</div>
</div>«;
}
//$ret=$ret.»
mysql_free_result($aRes);
}
else
{

```

```

$ret=«<center>Помилка при роботі з базою даних</center>«;
}
}
}
mysql_close($adb);
return $ret;
}
function detail_tovar_show($id_t)
{
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«select
tovar.id as id,
tovar.name_tovar as name,
tovar.amount as amount,
tovar.count as count,
tovar.description as description,
group_tovar.name_group as group_t,
group_compani.name_compani as compani_t
from tovar,group_tovar,group_compani where
tovar.id_group=group_tovar.id and
tovar.id_compani=group_compani.id
and tovar.id=«.$id_t;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
if($aRow=mysql_fetch_array($aRes))
{
$ret=$ret.»<div class='module'>
<table class='contentpaneopen'><tbody>
<tr>
<td class='contentheading' width='100%'>
«.$aRow[«group_t】.» «.$aRow[«compani_t】.»</td>
</tr></tbody>
</table>«;
$ret=$ret.»<table class='contentpaneopen'>
<div class='module'>
<div>
<div>

```

```

<div><<
$ret=$ret.» <strong><<.$aRow[«compani_t»].» <<.$aRow[«name»].»</strong>
<br>
<<.$aRow[«description»].»<br>
<i>Ціна:>.$aRow[«amount»].»грн. <<» Кількість:>.$aRow[«count»].»</i><<;
if($aRow[«count»]>0)
$ret=$ret.»<a class='readon' href='index.php?add=<<.$aRow[«id»].»'>в
КОШИК</a><<;
$ret=$ret.»</div>
</div>
</div>
</div><<;
}
$ret=$ret.»</ul>
</td>
</tr>
</tbody></table></div><<;
mysql_free_result($aRes);
}
else
{
$ret=<<<center>Помилка при роботі з базою даних</center><<;
}
}
}
mysql_close($adb);
return $ret;
}
function return_menu_koshik()
{
$size_m=sizeof($_SESSION['_SEL_T']);
$ret[1][1]=<<index.php?kosh=0>>;
$ret[1][2]=<<<strong>В кошику \>><<.$size_m.>>» товари</strong><<;
return $ret;
}
function add_koshik($id_tov)
{
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{

```

```

$aSql=«select
tovar.id as id,
tovar.name_tovar as name,
tovar.amount as amount,
tovar.count as count,
tovar.description as description,
group_tovar.name_group as group_t,
group_compani.name_compani as compani_t
from tovar,group_tovar,group_compani where
tovar.id_group=group_tovar.id and
tovar.id_compani=group_compani.id
and tovar.id=«. $id_tov;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
if($aRow=mysql_fetch_array($aRes))
{
$i=sizeof($_SESSION['_SEL_T']);
$_SESSION['_SEL_T'][$i+1]=$id_tov;
$ret=$ret.»<div class='module'>«;
$ret=$ret.»<table class='contentpaneopen'>
<div class='module'>
<div>
<div>
<div>«;
$ret=$ret.» <img class='speakers' src='images/action/package.png'>
<strong>Товар доданий до кошика</strong><br><br>
«. $aRow[«group_t»].»
«. $aRow[«compani_t»].» «. $aRow[«name»].»<br>
<a class='readon' href='index.php?detail=«. $id_tov.»'>назад</a>
</div>
</div>
</div>
</div>«;
}
$ret=$ret.»</ul>
</td>
</tr>
</tbody></table></div>«;
mysql_free_result($aRes);
}
else
{

```

```

$ret=«<center>Помилка при роботі з базою даних</center>«;
}
}
}
mysql_close($adb);
return $ret;
}
function show_koshik()
{
$size_k=sizeof($_SESSION['_SEL_T']);
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb) && ($size_k>0))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$ret=$ret.»
<table class='contentpaneopen'>
<tbody>
<tr>
<td class='contentheading' width='100%'>Товари в кошику</td>
</tr></tbody>
</table>«;
$ret=$ret.
«<div class='module'>
<div><div><div>
<table class='contentpaneopen' border=1 BORDERCOLOR='#d7d7d7'
RULES='GROUPS'>
<tr class='small' BGCOLOR='#eff2f3'>
<td align=center><b>№</b></td>
<td align=center><b>Товар</b></td>
<td align=center><b>Ціна</b></td>
<td align=center><b>Видалити</b></td>
</tr>«;
$sum_amount=0;
for($i=1;$i<=$size_k;$i++)
{
$aSql=«select tovar.id as id,tovar.name_tovar as name,tovar.amount as amount,
tovar.count as count,tovar.description as description,
group_tovar.name_group as group_t,group_compani.name_compani as
compani_t
from tovar,group_tovar,group_compani where
tovar.id_group=group_tovar.id and tovar.id_compani=group_compani.id

```



```

and tovar.id=«.$_SESSION['_SEL_T'][$i];

$Res=mysql_query($Sql,$adb);
if($Res==true)
{
if($Row=mysql_fetch_array($Res))
{
$ret=$ret.» <tr class='small'>
<td>«.$i.»</td>
<td>«.$Row[«group_t»].»
«.$Row[«compani_t»].»
«.$Row[«name»].»
</td>
<td align=center>
«.$Row[«amount»].» грн.
</td>
<td align=center>
<a href='index.php?kosh=1&id=«.$i.»'>
<img src='images/action/tool_delete.png' width='16' height='16'>
</a>
</td>
</tr>«;
$sum_amount=$sum_amount+$Row[«amount»];
}
mysql_free_result($Res);
}
}
$ret=$ret.»
<tr class='small' align=right BGCOLOR='#eff2f3'>
<td colspan=4><b>Всього: «.$sum_amount.» грн.</b></td>
</tr>
</table></div></div></div></div>«;
$ret=$ret.» <div class='module'>
<div>
<div>
<div>
<a class='readon' href='index.php?zamov'>Замовити</a>
</div>
</div>
</div>
</div><«;
}
mysql_close($adb);

```

```

}
return $ret;
}
function remove_from_koshik($id)
{
    $index=1;
    for($i=1;$i <= sizeof($_SESSION['_SEL_T']);$i++)
    {
        if($i != $id)
        {
            $ret[$index]=$_SESSION['_SEL_T'][$i];
            $index++;
        }
    }
    $_SESSION['_SEL_T']=null;

    $_SESSION['_SEL_T']=$ret;
}
//=====
function main_shov()
{
    if(isset($_GET['show']))
        $ret=show_guest_tovar($_GET['show']);
    else
        if(isset($_GET['detail']))
            $ret=detail_tovar_show($_GET['detail']);
        else
            if(isset($_GET['add']))
                $ret=add_koshik($_GET['add']);
            else
                if(isset($_GET['kosh']))
                {
                    switch($_GET['kosh'])
                    {
                        case 0:
                            $ret=show_koshik();
                            break;
                        case 1:
                            remove_from_koshik($_GET['id']);
                            $ret=show_koshik();
                            break;
                    }
                }
            }
}
}

```

```

return $ret;
}
//=====================================================
?>
Session.php
<?php
session_start();
session_register('_DB');
session_register('_DB_NAME');
session_register('_DB_USER');
session_register('_DB_PASS');
session_register('_SEL_T');
session_register('_USER');
$_SESSION['_DB']='localhost';
$_SESSION['_DB_NAME']='rave_mdb';
$_SESSION['_DB_USER']='guest';
$_SESSION['_DB_PASS']='loh';
if($_SESSION['_USER']!='admin')
$_SESSION['_USER']='guest';
//$_SESSION['_USER']='admin';
?>
EditTovar.php
<?php
function show_tovar()
{
$ret='<<<';
$db=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($db))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$db))
{
$aSql='select
tovar.id as id,
tovar.name_tovar as name,
tovar.amount as amount,
tovar.count as count,
group_tovar.name_group as group_t,
group_compani.name_compani as compani_t
from tovar,group_tovar,group_compani where
tovar.id_group=group_tovar.id and tovar.id_compani=group_compani.id';
$aRes=mysql_query($aSql,$db);
if($aRes==true)

```

```

{
$ret=«





```

```

    $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESSION['_DB_PASS']);
    if(!empty($adb))
    {
    if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
    {
    $aSql=«select * from tovar where id=«. $id_tov;
    $aRes=mysql_query($aSql,$adb);
    if($aRes==true)
    {
    if($aRow=mysql_fetch_array($aRes))
    {
    $t_group_id=$aRow[«id_group»];
    $c_group_id=$aRow[«id_compani»];
    $ret=«<form action='index.php?tovar=2&id=«. $aRow[«id»].»' method='post'>
    <table class='contentpaneopen'>
    «;
    $ret=$ret.»<tr><td>Назва товару:</td><td><INPUT TYPE=text SIZE=20
    NAME=tovar_name VALUE='«. $aRow[«name_tovar»].»'></td></tr>
    <tr><td>Ціна товару:</td><td><INPUT TYPE=text SIZE=20
    NAME=tovar_amount VALUE='«. $aRow[«amount»].»'></td></tr>
    <tr><td>Кількість товару:</td><td><INPUT TYPE=text SIZE=20
    NAME=tovar_count VALUE='«. $aRow[«count»].»'></td></tr>
    <tr><td>Опис товару:</td><td><TEXTAREA NAME=tovar_description
    ROWS=5 COLS=50>«. $aRow[«description»].»</TEXTAREA></td></tr>«;
    }
    mysql_free_result($aRes);
    }
    $aSql=«select * from group_tovar»;
    $aRes=mysql_query($aSql,$adb);
    if($aRes==true)
    {
    $tov_group=«<tr><td>Група товару:</td>
    <td><select name='tov_group'>«;
    while($aRow=mysql_fetch_array($aRes))
    {
    if($aRow[«id»]==$t_group_id)
    $tov_group=$tov_group.»<option value='«. $aRow[«id»].»'
    selected>«. $aRow[«name_group»];
    else
    $tov_group=$tov_group.»<option
    value='«. $aRow[«id»].»'>«. $aRow[«name_group»];
    }
    }

```

```

$stov_group=$stov_group.»</select></td></tr><<
mysql_free_result($aRes);
}
$ret=$ret.$stov_group;

$aSql=«select * from group_compani»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$stov_compani=«<tr><td>Компанія:</td><td><select name='tov_compani'><<
while($aRow=mysql_fetch_array($aRes))
{
if($aRow[«id»]==$c_group_id)
$stov_compani=$stov_compani.»<option
value='«.$aRow[«id»].»'
selected><<.$aRow[«name_compani»];
else
$stov_compani=$stov_compani.»<option
value='«.$aRow[«id»].»'><<.$aRow[«name_compani»];
}
$stov_compani=$stov_compani.»</select></td></tr><<
mysql_free_result($aRes);
}
$ret=$ret.$stov_compani.»<tr><td
colspan=2><INPUT
TYPE=submit
name='Submit' VALUE='Зберегти'>
</td></tr></table></form><<
}
}
mysql_close($adb);
return $ret;
}
function add_tovar()
{
$ret=«««;
$aadb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESSION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$ret=«<form action='index.php?tovar=2' method='post'>
<table class='contentpaneopen'><<
$ret=$ret.»<tr><td>Назва товару:</td><td><INPUT
TYPE=text SIZE=50
NAME=tovar_name VALUE=«.$aRow[«name»].»></td></tr>

```

```

        <tr><td>Ціна товару:</td><td><INPUT TYPE=text SIZE=20
NAME=tovar_amount VALUE=«.$aRow[«amount»].»></td></tr>
        <tr><td>Кількість товару:</td><td><INPUT TYPE=text SIZE=20
NAME=tovar_count VALUE=«.$aRow[«count»].»></td></tr>
        <tr><td>Опис товару:</td><td><TEXTAREA NAME=tovar_description
ROWS=5 COLS=50>«.$aRow[«description»].»</TEXTAREA></td></tr>«;
        $aSql=«select * from group_tovar»;
        $aRes=mysql_query($aSql,$adb);
        if($aRes==true)
        {
            $tov_group=«<tr><td>Група товару:</td>
<td><select name='tov_group'>«;
            while($aRow=mysql_fetch_array($aRes))
            $tov_group=$tov_group.»<option
value='«.$aRow[«id»].»'>«.$aRow[«name_group»];
            $tov_group=$tov_group.»</select></td></tr>«;
            mysql_free_result($aRes);
        }
        $ret=$ret.$tov_group;
        $aSql=«select * from group_compani»;
        $aRes=mysql_query($aSql,$adb);
        if($aRes==true)
        {
            $tov_compani=«<tr><td>Компанія:</td><td><select name='tov_compani'>«;
            while($aRow=mysql_fetch_array($aRes))
            {
                if($aRow[«id»]==$c__group_id)
                $tov_compani=$tov_compani.»<option
selected>«.$aRow[«name_compani»];
                else
                $tov_compani=$tov_compani.»<option
value='«.$aRow[«id»].»'>«.$aRow[«name_compani»];
            }
            $tov_compani=$tov_compani.»</select></td></tr>«;
            mysql_free_result($aRes);
        }
        $ret=$ret.$tov_compani.»<tr><td colspan=2><INPUT TYPE=submit
name='Submit' VALUE='Зберегти'>
</td></tr></table></form>«;
    }
}
mysql_close($adb);
return $ret;

```

```

    }
    function save_tovar($id_tov,$key)
    {
        $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
        if(!empty($adb))
        {
            if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
            {
                if($key==1)
                $aSql=«update tovar set
                name_tovar=«$_POST['tovar_name'].»,
                amount=«$_POST['tovar_amount'].»,
                count=«$_POST['tovar_count'].» ,
                description='«$_POST['tovar_description'].»' ,
                id_group=«$_POST['tov_group'].» ,
                id_compani=«$_POST['tov_compani'].» where id=«.$id_tov;
                else
                $aSql=«insert
                tovar(name_tovar,amount,count,description,id_group,id_compani)
                values('«$_POST['tovar_name'].»,
                «$_POST['tovar_amount'].»,
                «$_POST['tovar_count'].» ,
                '«$_POST['tovar_description'].»' ,
                «$_POST['tov_group'].»,
                «$_POST['tov_compani'].»)»;
                $aRes=mysql_query($aSql,$adb);
                if($aRes==true)
                $ret=$ret.»<center>Дані успішно внесені в базу даних</center>«;
                else
                $ret=$ret.»<center>Помилка при внесені даних</center>«;
            }
        }
        mysql_close($adb);
        return $ret;
    }
    function show_group_t()
    {
        $ret=«««;
        $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
        if(!empty($adb))
        {

```



```

if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
    $aSql=«select * from group_tovar»;
    $aRes=mysql_query($aSql,$adb);
    if($aRes==true)
    {
        $ret=«<table class='contentpaneopen' border=1 BORDERCOLOR='#d7d7d7'
RULES='All' cellpadding='0' cellspacing='0'>
        <tr class='small' BGCOLOR='#eff2f3' align=center>
        <td>№</td>
        <td>Назва групи</td>
        <td>Ред.</td>
        </tr>«;
        while($aRow=mysql_fetch_array($aRes))
        {
            $ret=$ret.»<tr class='small'>«;
            $ret=$ret.»<td>«. $aRow[«id»].»</td>
            <td>«. $aRow[«name_group»].»</td>«;
            $ret=$ret.»<td align=center><a href='index.php?gt=1&id=«. $aRow[«id»].»'>
            <img src='images/action/char.png' width='16' height='16'>
            </a></td></tr>«;
        }
        $ret=$ret.»</table>«;
        mysql_free_result($aRes);
    }
    else
    {
        $ret=$ret.»<center>Помилка при роботі з базою даних</center>«;
    }
}
mysql_close($adb);
return $ret;
}
function edit_group_t($id_g)
{
    $ret=«««;
    $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
    if(!empty($adb))
    {
        if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
        {

```

```

$aSql=«select * from group_tovar where id=«. $id_g;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
if($aRow=mysql_fetch_array($aRes))
{
$ret=«<form action='index.php?gt=2&id=«. $aRow[«id»].»' method='post'>
<table class='contentpaneopen'>
«;
$ret=$ret.» <tr><td>Назва групи:</td>
<td><INPUT TYPE=text SIZE=50 NAME=group_name
VALUE='«. $aRow[«name_group»].»'></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form>«;
}
mysql_free_result($aRes);
}
}
mysql_close($adb);
return $ret;
}
function add_group_t()
{
$ret=«<form action='index.php?gt=2' method='post'>
<table class='contentpaneopen'>
<tr><td>Назва групи:</td>
<td><INPUT TYPE=text SIZE=50 NAME=group_name VALUE=></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form>«;
return $ret;
}
function save_group_t($id_g,$key)
{
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESSION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
if($key==1)

```

```

$aSql=«update group_tovar set
name_group='«.$_POST['group_name'].»'
where id=«.$id_g;
else
$aSql=«insert into group_tovar(name_group)
values('«.$_POST['group_name'].»')»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
$ret=$ret.»<center>Дані успішно внесені в базу даних</center>«;
else
$ret=$ret.»<center>Помилка при внесені даних</center>«;
}
}
mysql_close($adb);
return $ret;
}
function show_group_c()
{
$ret=«««;
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«select * from group_compani»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$ret=«<table class='contentpaneopen' border=1 BORDERCOLOR='#d7d7d7'
RULES='All' >
<tr class='small' BGCOLOR='#eff2f3' align=center>
<td>№</td>
<td>Назва компанії</td>
<td>Ред.</td>
</tr>«;
while($aRow=mysql_fetch_array($aRes))
{
$ret=$ret.»<tr class='small'>«;
$ret=$ret.»<td>«.$aRow[«id»].»</td>
<td>«.$aRow[«name_compani»].»</td>«;
$ret=$ret.»<td align=center><a href='index.php?gc=1&id=«.$aRow[«id»].»'>
<img src='images/action/char.png' width='16' height='16'>

```

```

</a></td></tr><<;
}
$ret=$ret.»</table><<;
mysql_free_result($aRes);
}
else
{
$ret=$ret.»<center>Помилка при роботі з базою даних</center><<;
}
}
}
mysql_close($adb);
return $ret;
}
function edit_group_c($id_g)
{
$ret=<<<<;
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=<<select * from group_compani where id=<<.$id_g;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
if($aRow=mysql_fetch_array($aRes))
{
$ret=<<<form action='index.php?gc=2&id=<<.$aRow[<<id>>].>>' method='post'>
<table class='contentpaneopen'>
<<;
$ret=$ret.» <tr><td>Назва компанії:</td>
<td><INPUT TYPE=text SIZE=50 NAME=compani_name
VALUE='<<.$aRow[<<name_compani>>].>>'></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form><<;
}
mysql_free_result($aRes);
}
}
}
}

```

```

mysql_close($adb);
return $ret;
}
function add_group_c()
{
$ret=«<form action='index.php?gc=2' method='post'>
<table class='contentpaneopen'>
<tr><td>Назва компанії:</td>
<td><INPUT TYPE=text SIZE=50 NAME='compani_name'></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form>«;
return $ret;
}
function save_group_c($id_g,$key)
{
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESSION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
if($key==1)
$aSql=«update group_compani set
name_compani='«.$_POST['compani_name'].»' where id=«.$id_g;
else
$aSql=«insert into group_compani(name_compani)
values('«.$_POST['compani_name'].»）」;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
$ret=$ret.»<center>Дані успішно внесені в базу даних</center>«;
else
$ret=$ret.»<center>Помилка при внесені даних</center>«;
}
}
mysql_close($adb);
return $ret;
}
//=====
function show_sity()
{
$ret=«««;

```

```

    $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
    if(!empty($adb))
    {
    if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
    {
    $aSql=«select * from sity»;
    $aRes=mysql_query($aSql,$adb);
    if($aRes==true)
    {
    $ret=«<table class='contentpaneopen' border=1 BORDERCOLOR='#d7d7d7'
RULES='All' >
    <tr class='small' BGCOLOR='#eff2f3' align=center>
    <td>№</td>
    <td>Назва міста</td>
    <td>Ред.</td>
    </tr>«;
    while($aRow=mysql_fetch_array($aRes))
    {
    $ret=$ret.»<tr class='small'>«;
    $ret=$ret.»<td>«.$aRow[«id»].»</td>
    <td>«.$aRow[«name»].»</td>«;
    $ret=$ret.»<td align=center><a href='index.php?sity=1&id=«.$aRow[«id»].»'>
    <img src='images/action/char.png' width='16' height='16'>
    </a></td></tr>«;
    }
    $ret=$ret.»</table>«;
    mysql_free_result($aRes);
    }
    else
    {
    $ret=$ret.»<center>Помилка при роботі з базою даних</center>«;
    }
    }
    }
    mysql_close($adb);
    return $ret;
    }
    function edit_sity($id_g)
    {
    $ret=«««;
    $adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);

```

```

if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$Sql=«select * from sity where id=«. $id_g;
$Res=mysql_query($Sql,$adb);
if($Res==true)
{
if($Row=mysql_fetch_array($Res))
{
$ret=«<form action='index.php?sity=2&id=«. $Row[«id»].»' method='post'>
<table class='contentpaneopen'>
«;
$ret=$ret.» <tr><td>Назва міста:</td>
<td><INPUT TYPE=text SIZE=50 NAME=sity_name
VALUE='«. $Row[«name»].»'></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form>«;
}
mysql_free_result($Res);
}
}
mysql_close($adb);
return $ret;
}
function add_sity()
{
$ret=«<form action='index.php?sity=2' method='post'>
<table class='contentpaneopen'>
<tr><td>Назва міста:</td>
<td><INPUT TYPE=text SIZE=50 NAME='sity_name'></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form>«;
return $ret;
}
function save_sity($id_g,$key)
{
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))

```

```

{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
if($key==1)
$aSql=«update sity set name='«.$_POST['sity_name'].»' where id=«.$id_g;
else
$aSql=«insert into sity(name) values('«.$_POST['sity_name'].»）」;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
$ret=$ret.»<center>Дані успішно внесені в базу даних</center>«;
else
$ret=$ret.»<center>Помилка при внесені даних</center>«;
}
}
mysql_close($adb);
return $ret;
}
//=====
//=====
=====
function show_oplata()
{
$ret=«««;
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESSION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«select * from type_oplaty»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$ret=«<table class='contentpaneopen' border=1 BORDERCOLOR='#d7d7d7'
RULES='All' >
<tr class='small' BGCOLOR='#eff2f3' align=center>
<td>№</td>
<td>Форма отримання</td>
<td>Ред.</td>
</tr>«;
while($aRow=mysql_fetch_array($aRes))
{
$ret=$ret.»<tr class='small'>«;

```



```

$ret=$ret.»<td>«. $aRow[«id»].»</td>
<td>«. $aRow[«name_oplaty»].»</td>«;
$ret=$ret.»<td align=center><a href='index.php?opl=1&id=«. $aRow[«id»].»'>
<img src='images/action/char.png' width='16' height='16'>
</a></td></tr>«;
}
$ret=$ret.»</table>«;
mysql_free_result($aRes);
}
else
{
$ret=$ret.»<center>Помилка при роботі з базою даних</center>«;
}
}
}
mysql_close($adb);
return $ret;
}
function edit_oplata($id_g)
{
$ret=«««;
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«select * from type_oplaty where id=«. $id_g;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
if($aRow=mysql_fetch_array($aRes))
{
$ret=«<form action='index.php?opl=2&id=«. $aRow[«id»].»' method='post'>
<table class='contentpaneopen'>
«;
$ret=$ret.» <tr><td>Форма отримання:</td>
<td><INPUT TYPE=text SIZE=50 NAME=oplaty_name
VALUE='«. $aRow[«name_oplaty»].»'></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form>«;
}
}
}
}
}

```

```

mysql_free_result($aRes);
}
}
}
mysql_close($adb);
return $ret;
}
function add_oplata()
{
$ret=«<form action='index.php?opl=2' method='post'>
<table class='contentpaneopen'>
<tr><td>Форма отримання:</td>
<td><INPUT TYPE=text SIZE=50 NAME='oplaty_name'></td></tr>
<tr><td colspan=2><INPUT TYPE=submit name='Submit'
VALUE='Зберегти'></td></tr>
</table></form>«;
return $ret;
}
function save_oplata($id_g,$key)
{
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
if($key==1)
$aSql=«update type_oplata set name_oplata=«.$_POST['oplaty_name'].»'
where id=«.$id_g;
else
$aSql=«insert into type_oplata(name_oplata)
values(«.$_POST['oplaty_name'].»)»»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
$ret=$ret.»<center>Дані успішно внесені в базу даних</center>«;
else
$ret=$ret.»<center>Помилка при внесені даних</center>«;
}
}
mysql_close($adb);
return $ret;
}

```

```

//=====
=====
function ret_menu_tovar()
{
    $m_el1 [1][1]=«index.php?tovar=0»;           $m_el1 [1][2]=«Товар»;
    $m_el1 [2][1]=«index.php?tovar=1»;           $m_el1 [2][2]=«:::НОВИЙ ТОВАР»;
    $m_el1 [3][1]=«index.php?gt=0»;               $m_el1 [3][2]=«Група
товарів»;
    $m_el1 [4][1]=«index.php?gt=1»;               $m_el1 [4][2]=«:::НОВА
група»;
    $m_el1 [5][1]=«index.php?gc=0»;               $m_el1 [5][2]=«Компанії»;
    $m_el1 [6][1]=«index.php?gc=1»;               $m_el1 [6][2]=«:::НОВА
компанія»;
    $m_el1 [7][1]=«index.php?sity=0»;             $m_el1 [7][2]=«Міста»;
    $m_el1 [8][1]=«index.php?sity=1»;             $m_el1 [8][2]=«:::НОВЕ МІСТО»;
    $m_el1 [9][1]=«index.php?opl=0»;              $m_el1 [9][2]=«Форми
отримання»;
    $m_el1 [10][1]=«index.php?opl=1»;             $m_el1 [10][2]=«:::НОВА ФОРМА»;
    return $m_el1;
}
//=====
function main_tov()
{
    if(isset($_GET['tovar']))
    {
        switch($_GET['tovar'])
        {
            case 0:
                $ret=show_tovar();
                break;
            case 1:
                if(isset($_GET['id']))
                $ret=edit_tovar($_GET['id']);
                else
                $ret=add_tovar();
                break;
            case 2:
                if(isset($_POST['Submit']))
                {
                    if(isset($_GET['id']))
                    $ret=save_tovar($_GET['id'],1);
                    else
                    $ret=save_tovar(«»,0);
                }
        }
    }
}

```

```
}
break;
}
}
else
{
if(isset($_GET['gt']))
{
switch($_GET['gt'])
{
case 0:
$ret=show_group_t();
break;
case 1:
if(isset($_GET['id']))
$ret=edit_group_t($_GET['id']);
else
$ret=add_group_t();
break;
case 2:
if(isset($_POST['Submit']))
{
if(isset($_GET['id']))
$ret=save_group_t($_GET['id'],1);
else
$ret=save_group_t(<<<,0);
}
break;
}
}
else
{
if(isset($_GET['gc']))
{
switch($_GET['gc'])
{
case 0:
$ret=show_group_c();
break;
case 1:
if(isset($_GET['id']))
$ret=edit_group_c($_GET['id']);
else
```

```

$ret=add_group_c();
break;
case 2:
if(isset($_POST['Submit']))
{
if(isset($_GET['id']))
$ret=save_group_c($_GET['id'],1);
else
$ret=save_group_c(<<<,0);
}
break;
}
}
else
{
if(isset($_GET['sity']))
{
switch($_GET['sity'])
{
case 0:
$ret=show_sity();
break;
case 1:
if(isset($_GET['id']))
$ret=edit_sity($_GET['id']);
else
$ret=add_sity();
break;
case 2:
if(isset($_POST['Submit']))
{
if(isset($_GET['id']))
$ret=save_sity($_GET['id'],1);
else
$ret=save_sity(<<<,0);
}
break;
}
}
else
{
if(isset($_GET['opl']))
{

```

```

switch($_GET['opl'])
{
case 0:
$ret=show_oplata();
break;
case 1:
if(isset($_GET['id']))
$ret=edit_oplata($_GET['id']);
else
$ret=add_oplata();
break;
case 2:
if(isset($_POST['Submit']))
{
if(isset($_GET['id']))
$ret=save_oplata($_GET['id'],1);
else
$ret=save_oplata(<<<,0);
}
break;
}
}
}
}
}
}
}
return $ret;
}
//=====================================================
?>
Zamov.php
<?php

```

```

function show_zamov()
{
$ret=$ret.» <table class='contentpaneopen'>
<tbody>
<tr>
<td class='contentheading' width='100%'>Оформлення замовлення</td>
</tr></tbody>
</table>
<div class='module'>
<div><div><div>

```

```

<form action='index.php?zamov=1' method='post'>
<table>
<tr class='small'>
<td>Імя<font color='#ff0000'>*</font></td>
<td>
<INPUT TYPE=text SIZE=30 NAME=user_name VALUE="">
</td>
</tr>
<tr class='small'>
<td>Привізвище<font color='#ff0000'>*</font></td>
<td>
<INPUT TYPE=text SIZE=30 NAME=user_secname VALUE="">
</td>
</tr>
<tr class='small'>
<td>Місто<font color='#ff0000'>*</font></td>
<td><<;
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«select id,name from sity»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$ret=$ret.»<SELECT NAME='sity_id'><<;
while($aRow=mysql_fetch_array($aRes))
{
$ret=$ret.»<OPTION VALUE='«.$aRow[«id»].»'><<.$aRow[«name»];
}
$ret=$ret.»</SELECT><<;
mysql_free_result($aRes);
}
}
}
mysql_close($adb);
$ret=$ret.»</td></tr>
<tr class='small'>
<td>Вулиця<font color='#ff0000'>*</font></td>
<td>
<INPUT TYPE=text SIZE=30 NAME=user_street VALUE="">

```

```

</td>
</tr>
<tr class='small'>
<td>Будинок<font color='#ff0000'>*</font></td>
<td>
<INPUT TYPE=text SIZE=10 NAME=user_home VALUE="">
</td>
</tr>
<tr class='small'>
<td>Квартира</td>
<td>
<INPUT TYPE=text SIZE=10 NAME=user_flat VALUE="">
</td>
</tr>
<tr class='small'>
<td>Почтовий індекс<font color='#ff0000'>*</font></td>
<td>
<INPUT TYPE=text SIZE=10 NAME=user_post VALUE="">
</td>
</tr>
<tr class='small'>
<td>Телефон<font color='#ff0000'>*</font></td>
<td>
<INPUT TYPE=text SIZE=10 NAME=user_phone VALUE="">
</td>
</tr>«;
$ret=$ret.»<tr class='small'>
<td>Спосіб отримання<font color='#ff0000'>*</font></td>
<td>«;
$db=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«select * from type_oplaty»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$ret=$ret.»<SELECT NAME='oplaty_id'>«;
while($aRow=mysql_fetch_array($aRes))
{
$ret=$ret.»<OPTION VALUE='«.$aRow[«id»].»'>«.$aRow[«name_oplaty»];

```



```

}
$ret=$ret.»</SELECT>«;
mysql_free_result($aRes);
}
}
}
mysql_close($adb);
$ret=$ret.»</td> </tr>«;
$ret=$ret.»<tr class='small'>
<td colspan=2>
<INPUT TYPE=submit name='Submit' VALUE='Відправити замовлення'>
</td>
</tr>
</table>
</form>
</div></div></div>
</div>«;
return $ret;
}
function save_zamov()
{
$adb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESS
ION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«insert                                     into
zamovnik(name,second_name,id_sity,street,home,flat,post_code,phone)
values(
'«.$_POST['user_name'].»',
'«.$_POST['user_secname'].»',
«.$_POST['sity_id'].»,
'«.$_POST['user_street'].»',
'«.$_POST['user_home'].»',
'«.$_POST['user_flat'].»',
'«.$_POST['user_post'].»',
'«.$_POST['user_phone'].»')»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$id_zamovnik=mysql_insert_id($adb);
//mysql_free_result($aRes);

```

```

$aSql=«insert into zamov(data_zamov,id_zamovnika,id_oplaty)
values('«.date(«Y-m-d»).'»,»,»,».$id_zamovnik.»»,».$_POST['oplaty_id'].»)»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$id_zamov=mysql_insert_id($adb);
$size_k=sizeof($_SESSION['_SEL_T']);
for($i=1;$i<=$size_k;$i++)
{
$aSql=«insert into zamov_tovar(id_zamov,id_tovar,count_tovar)
values(«.$id_zamov.»»,».$_SESSION['_SEL_T'][$i].»,1)»;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
$aSql=«update tovar set count=count-1 where id=«.$_SESSION['_SEL_T'][$i];
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
}
}
}
}
$ret=
«
<table class='contentpaneopen'>
<tbody>
<tr>
<td class='contentheading' width='100%'>Дякуємо</td>
</tr></tbody>
</table>
<div class='module'>
<div>
<div>
<div>
<img class='speakers' src='images/action/email.png'>
<strong>Ваше замовлення відправлене</strong>
</div>
</div>
</div>
</div>
«;
$_SESSION['_SEL_T']=null;
}
}

```

```

else
$ret=«<center>Помилка в ведених даних</center>»;
}
}
mysql_close($adb);
return $ret;
}
//=====
function main_zamov()
{
if(isset($_GET['zamov']))
{
switch ($_GET['zamov'])
{
case 1:
$ret=save_zamov();
break;
default:
$ret=show_zamov();
}
}
return $ret;
}
//=====
?>
main.php
<?php
function login_show()
{
$ret=
«
<table class='contentpaneopen'>
<tbody>
<tr>
<td class='contentheading' width='100%'>Вхід в систему</td>
</tr></tbody>
</table>
<div class='module'>
<div>
<div>
<div>
<img class='speakers' src='images/action/amsn.png'>
<form action='index.php?main=5' method='post'>

```

```

<table>
<tr class='small'>
<td>Login:</td>
<td><INPUT TYPE=text SIZE=20 NAME=login VALUE=""></td>
</tr>
<tr class='small'>
<td>Password:</td>
<td><INPUT TYPE=password SIZE=20 NAME=password
MAXLENGTH=10></td>
</tr>
<tr class='small'>
<td colspan=2><INPUT TYPE=submit name='Submit' VALUE=' Ok '></td>
</tr>
</table>
</form>
</div></div></div></div>
«;
return $ret;
}
function login_authorized($login,$pass)
{
$aadb=mysql_connect($_SESSION['_DB'],$_SESSION['_DB_USER'],$_SESSION['_DB_PASS']);
if(!empty($adb))
{
if(mysql_select_db($_SESSION['_DB_NAME'],$adb))
{
$aSql=«select * from users where login='«.$login.»' and pass='«.$pass.»'«;
$aRes=mysql_query($aSql,$adb);
if($aRes==true)
{
if($aRow=mysql_fetch_array($aRes))
{
$_SESSION['_USER']=«admin»;
}
mysql_free_result($aRes);
} } }
return $ret;
}
function login_close()
{
$_SESSION['_USER']=«guest»;
}

```

```
//=====
function main_login()
{
if(isset($_GET['main']))
{
switch($_GET['main'])
{
case 5:
if(isset($_POST['Submit']))
{
if(isset($_POST['login']))
login_authorized($_POST['login'],$_POST['password']);
}
else
$ret=login_show();
break;
case 6:
login_close();break;
} }
return $ret;
}
//=====
?>
```