

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

Аліна САВЧЕНКО

“ ” _____ 2021 р.

ДИПЛОМНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТРА”

ЗА ОСВІТНЬО-ПРОФЕСІЙНОЮ ПРОГРАМОЮ “ІНФОРМАЦІЙНІ
УПРАВЛЯЮЧІ СИСТЕМИ ТА ТЕХНОЛОГІЇ”

Тема: “Метод Web-розробки Internet-магазину для торгівлі з прямими поставками”

Виконавець: Лазоренко Максим Олександрович

Керівник: професор Віноградов Микола Анатолійович

Нормоконтролер: _____ Ігор РАЙЧЕВ

Київ - 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Галузь знань, спеціальність, освітньо-професійна програма: 12
“Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні
управляючі системи та технології”

ЗАТВЕРДЖУЮ

Завідувач кафедри

Аліна САВЧЕНКО

« ____ » _____ 2021р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Лазоренка Максима Олександровича

(прізвище, ім'я, по батькові)

- 1. Тема роботи:** «Метод Web-розробки Internet-магазину для торгівлі з
прямими поставками» затверджена наказом . ректора від 12.10.2021 за
№ 2228/ст.
- 2. Термін виконання роботи:** з 12.10.2021 по 31.12.2021.
- 3. Вихідні дані до роботи:** теоретичні відомості та основи проектування
інформаційних систем, множина відомих архітектур програмних компонентів
і додатків та програмних систем (ПС), множина відомих патернів
проектування .
- 4. Зміст пояснювальної записки:** вступ, огляд теоретичної бази, огляд
програмних компонентів, розробка проекту, висновки..
- 5. Перелік обов'язкового ілюстративного матеріалу:** слайди, презентація.

6. Календарний план-графік

п/п	Завдання	Термін виконання	Підпис керівника
1.	Отримання завдання на дипломну роботу та побудова плану-графіку виконання робіт.	12.10.2021– 15.10.2021	
2.	Огляд методів продажу товару. Формування списку перевах та недоліків Дропшипінгу.	16.10.2021– 19.10.2021	
3.	Огляд та аналіз відомих програмних архітектур і патернів	20.10.2021– 24.10.2021	
4.	Проектування програмної системи управління інтернет магазином.	25.10.2021– 31.10.2021	
5.	Написання Розділу 1 дипломної роботи.	01.11.2021– 07.11.2021	
6.	Розробка та реалізація програмного комплексу управління інтернет магазином. Написання Розділу 2 дипломної роботи.	08.11.2021– 17.11.2021	
7.	Написання Розділу 3 дипломної роботи. Завершення створення пояснювальної записки дипломної роботи.	18.11.2021– 01.12.2021	
8.	Оформлення та друк пояснювальної записки.	02.12.2021– 11.12.2021	
9.	Створення презентації, доповіді та підготовка до захисту дипломної роботи	12.12.2021– 20.12.2021	

7. Дата видачі завдання: 12.10.2021р.

Керівник дипломної роботи _____ Микола ВІНОГРАДОВ

(підпис керівника)

Завдання прийняв до виконання _____ Максим ЛАЗОРЕНКО

(підпис випускника)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Метод Web-розробки Internet-магазину для торгівлі з прямими поставками» викладена на 92 сторінках, містить 45 рисунків, 14 літературних джерел.

Мета проекту – розробка системи для керування процесами дропшипінгу.

Об'єкт дослідження – власноруч розроблена система управління інтернет-магазином.

Предмет дослідження – Web-сайт, інтернет-магазину що створений за допомогою BITRIX для продаж за системою DropShiping.

Ключові слова - САЙТ, СИСТЕМИ УПРАВЛІННЯ САЙТОМ, BITRIX, JAVASCRIPT, MYSQL, HTML, CSS, PHP, CMS, OPENSERVER, ВЕБ-СЕРВЕР, DROPSHIPING.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП.....	8
РОЗДІЛ 1. Вибір програмного забезпечення	9
1.1. Постановка задачі	9
1.2. Переваги дропшипінгу.....	10
1.3. Недоліки дропшипінгу.....	12
1.4. Вибір програмного забезпечення.....	15
1.5. Список систем управління.....	20
1.6. Вибір системи управління	25
1.7. Вибір бази даних.....	25
1.8. Обмін даними з постачальником	38
РОЗДІЛ 2. Створення сайту	48
2.1. Формат діяльності підприємства	48
2.2.Етапи створення сайту	49
2.3. Текстовий редактор	52
2.4. Налаштування локального серверу	53
2.5. Встановлення Вітріх.....	57
2.6. Створення карти сайту.....	67
2.7. Створення дизайну сайту.....	72
2.8. Розробка шаблону сайту	73
2.9. Розробка автоматичного завантаження каталогу з сервера постачальника.....	74
2.10. Обмін виконаними замовленнями	78
2.11. Опрацювання замовлення.....	78
2.12. Список користувачів	79

РОЗДІЛ 3. Відображення готового сайту	81
3.1. Опис інтерфейсу адміністратора сайту	81
3.2. Вигляд сайту для користувачів	84
3.3. Список замовлень	87
3.4. Проактивний захист	88
ВИСНОВКИ	90
СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

MVC - Model View Controller

PHP - personal home page

JS - JavaScript

HTTP - Hyper Text Transfer Protocol

API - application programming interface

FTP - File Transfer Protocol

СУБД - система управління бази даних

БД - база данных

URL - Uniform Resource Locator

UML - Unified Modeling Language

ВСТУП

В умовах ринкової економіки в наслідок розвитку ІТ сектору та збільшення попиту на товари, виникають нові, альтернативні форми продажу та обороту товарів. Яскравим прикладом такого є дропшипінг, що став новим явищем на ринку.

Суть дропшипінгу полягає в тому, що посередник за попередньою домовленістю з виробником/оптовиком пропонує на продаж свої товари з націнкою, покупець обирає товар, вносить передоплату дропшиперу; Дропшиппер, у свою чергу, оплачує вартість товару без націнки (націнка – це дохід дропшиппера) виробнику/оптовику, який після отримання оплати від імені дропшиппера відправляє товар покупцю. Дропшипінг працює на довірі, тому що, з одного боку, покупець переводить кошти без гарантій отримання товару, з іншого - дропшипер оплачує товар постачальнику без впевненості у доставці та якості товару, тому ризики drop shipping виникають через недовіру, неможливість контролювати всі процеси та якість товару. Тому необхідно визначити переваги та недоліки дропшипінгу.

В даній роботі розглянемо розробку системи для керування та автоматизації процесів дропшипінгу. Система складається публічної частини та адміністративної частини.

Публічна частина це сайт – інтернет магазин. Для нього розроблено каталог, детальну сторінку товару, кошик та особистий кабінет.

З боку адміністративної частини розроблено функціонал списку замовлень для їх менеджменту, та список зареєстрованих користувачів.

РОЗДІЛ 1

ВИБІР ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Постановка задачі

Дропшипінг — це спосіб виконання замовлень, при якому магазин не зберігає продані товари на складі. Натомість магазин купує товар у стороннього постачальника і відправляє його клієнту. Як наслідок, продавцю не потрібно безпосередньо обробляти товар.

Найбільша відмінність між дропшипінгом і стандартною моделлю роздрібною торгівлі полягає в тому, що продавець не має запасів товару на складі і не володіє ними. Замість цього продавець закупає товари за потребою у третьої сторони — зазвичай оптового продавця або виробника — для виконання замовлень.



Рис. 1.1. Схема роботи дропшипінгу

Кафедра КІТ (47)				НАУ 21 32 00 000 ПЗ			
Виконав	Лазоренко М.О.			Постановка задачі	літера	арк	аркушів
Керівник	Віноградов М.А.					9	39
Консульт					УС-212М 122		
Н. контрол	Райчев І.Е.						

1.2. Переваги дропшипінгу

Дропшипінг — це чудова бізнес-модель для підприємців-початківців, оскільки вона доступна. Завдяки дропшипінгу ви можете швидко тестувати різні бізнес-ідеї з обмеженими недоліками, що дає змогу дізнатися багато нового про те, як вибирати та продавати затребувані продукти. Ось кілька інших причин, чому дропшипінг є такою популярною моделлю.

1.2.1. Менший капітал

Ймовірно, найбільша перевага дропшипінгу полягає в тому, що можна запустити магазин електронної комерції без необхідності інвестувати тисячі доларів у запас. Традиційно роздрібним торговцям доводилося зв'язувати величезні обсяги запасів капіталу.

З моделлю дропшипінгу вам не потрібно купувати продукт, якщо ви вже продали товар і не заплатили клієнту. Без значних авансових інвестицій у запаси можна розпочати пошук продуктів і розпочати успішний бізнес дропшипінг з дуже невеликими коштами. А оскільки ви не збираєтеся продавати через будь-які запаси, придбані наперед, як у традиційному роздрібному бізнесі, є менше ризиків, пов'язаних із відкриттям магазину дропшипінг або тестуванням нової лінії продуктів.

1.2.2. Легко почати

Ведення бізнесу електронної комерції набагато легше, якщо вам не потрібно мати справу з фізичними продуктами. Завдяки дропшипінгу вам не потрібно турбуватися про:

- Управління або оплата складу.
- Пакування та відправка ваших замовлень.

- Відстеження запасів для обліку.
- Обробка повернення та вхідних відправок.
- Постійно замовляєте продукти та контролюйте рівень запасів.

1.2.3. Низькі накладні витрати

Оскільки вам не потрібно займатися закупівлею наявності товару або займатися управлінням складом, ваші накладні витрати досить низькі. Насправді, багато успішних магазинів дропшипінгу працюють як домашні підприємства, що вимагають лише ноутбука та кількох періодичних витрат для роботи. У міру вашого зростання ці витрати, ймовірно, зростатимуть, але все одно залишаться низькими в порівнянні з витратами на традиційний бізнес.

1.2.4. Незалежність від геолокації

Дропшипінг підприємством можна керувати практично будь-де, де є інтернет-з'єднання. Поки ви можете легко спілкуватися з постачальниками та клієнтами, ви можете керувати своїм бізнесом.

1.2.5. Широкий вибір товарів для продажу

Оскільки вам не потрібно попередньо купувати товари, які ви продаєте, ви можете запропонувати своїм потенційним клієнтам низку популярних продуктів. Якщо у постачальників є товар, ви можете виставити його на продаж у своєму інтернет-магазині без додаткових витрат.

1.2.6. Легше тестувати

Дропшипінг - це корисний метод роботи як для відкриття нового магазину, так і для власників бізнесу, які хочуть перевірити апетит клієнтів до додаткових категорій продуктів, наприклад, аксесуарів або абсолютно нових ліній продуктів. Головною перевагою дропшипінгу, знову ж таки, є можливість перераховувати та потенційно продавати продукти, перш ніж купувати велику кількість товарів.

1.2.7. Легке масштабування

У традиційному роздрібному бізнесі, якщо ви отримуєте втричі більше замовлень, вам, як правило, доведеться виконати втричі більше роботи. Використовуючи постачальників дропшипінгу, більшість роботи по обробці додаткових замовлень будуть нести постачальники, що дозволить вам розширюватися з меншою кількістю проблем із зростанням і меншою роботою.

Зростання продажів завжди приносить додаткову роботу, особливо пов'язану з підтримкою клієнтів, але підприємства, які використовують масштаби дропшипінгу, особливо добре порівняно з традиційними компаніями електронної комерції.

1.3. Недоліки Дропшипінгу

Усі переваги, згадані вище, роблять дропшипінг дуже привабливою моделлю для тих, хто починає працювати з інтернет-магазином, або для тих, хто хоче розширити свої існуючі пропозиції продуктів. Але, як і всі підходи, дропшипінг також має свої мінуси. Взагалі кажучи, зручність і гнучкість мають свою ціну.

1.3.1. Низька маржа

Низька маржа є найбільшим недоліком роботи в висококонкурентній ніші дропшипінгу. Оскільки розпочати роботу так легко, а накладні витрати настільки мінімальні, багато конкуруючих магазинів створюють магазини та продають товари за найнижчими цінами, намагаючись збільшити дохід. Оскільки вони так мало інвестували в початок бізнесу, вони можуть дозволити собі працювати з мінімальною прибутковістю.

Зазвичай такі продавці мають низькоякісні веб-сайти та погане (якщо таке є) обслуговування клієнтів, що можна використати, щоб виділити свій бізнес. Але це не завадить клієнтам порівнювати свої ціни з вашими. Це посилення жорсткої конкуренції швидко зашкодить потенційній прибутку в ніші. На щастя, можна зробити багато чого, щоб пом'якшити цю проблему, вибравши нішу, яка добре підходить для дропшипінгу.

1.3.2. Проблеми інвентаризації

У випадку власного складу і самостійного зберігання товару, досить просто відстежувати, які товари є на складі, а які відсутні. Але коли ви купуєте товари з кількох складів, які також виконують замовлення для інших продавців, інвентар може змінюватися щодня. На щастя, сьогодні є кілька програм, які дозволяють синхронізуватися з постачальниками. Таким чином, дропшиппери можуть «передавати» замовлення постачальнику одним або двома кліками, і вони повинні мати можливість в режимі реального часу бачити, скільки наявних товарів у постачальника.

1.3.3. Складності доставки

Якщо робота налагоджена з декількома постачальниками як це роблять більшість дропшиперів товари у інтернет-магазині будуть надходити через кілька різних постачальників. Це ускладнює витрати на доставку.

Скажімо, клієнт розміщує замовлення на три елементи, які доступні лише в окремих постачальників. За відправку кожного товару клієнту ви стягнете три окремі оплати за доставку, але, ймовірно, недоцільно передавати цю плату клієнту. І навіть якщо є сенс включити ці витрати клієнту, автоматизувати ці розрахунки може бути важко.

1.3.4. Помилки постачальника

Навіть найкращі постачальники дропшипінгу роблять помилки, виконуючи замовлення помилки, за які ви повинні нести відповідальність і вибачатися. А посередні та неякісні постачальники будуть викликати нескінченне розчарування через брак товарів, невдале відправлення та неякісне пакування, що може зіпсувати репутацію.

1.3.5. Обмежені налаштування та брендування

На відміну від товарів, виготовлених на замовлення, дропшипінг не дає вам великого контролю над самим товаром. Зазвичай, товар, що поставляється дропшиперами, розроблений і брендований постачальником.

Деякі постачальники популярного китайського маркетплейсу Alibaba можуть врахувати зміни для товарів, враховуючи вимоги вашого бізнесу, але навіть тоді постачальник має найбільший контроль над самим продуктом. Будь-які зміни або доповнення до самого продукту зазвичай вимагають мінімальної кількості замовлення, щоб зробити його життєздатним і

доступним для виробника, адже мало партійні товари, при високих затратах на початку виробництва, не можуть окупитися через їхню низьку маржинальність.

1.4. Вибір програмного забезпечення

Майже всі веб додатки використовують в своїй реалізації архітектуру клієнт-сервер. Дана архітектура є найпоширенішою в створенні мережових веб-додатків, і базується на обміні даними між клієнтом та сервером, через мережу інтернет, використовуючи протоколи шифрування та обміну даними.

Основою даної архітектури є:

- 1) Клієнти, які роблять запити до серверів. Такими клієнтами зазвичай є програмні або апаратні продукти, які встановлені на обчислювальних машинах користувачів. Це можуть бути браузер, додатки та ін.
- 2) Сервери, які приймають запити, обчислюють вхідні дані та надають відповіді клієнтам. Сервер це автономний комп'ютер або його імітація програмним забезпеченням на іншому комп'ютері, яке реагує на зовнішні події, відповідно до програмного забезпечення.
- 3) Мережа, яка надає зв'язок між сервером та клієнтом. Зачасту це мережа інтернет, яка є набором маршрутизаторів та роутерів з'єднаних між собою. Комунікація відбувається за протоколом TCP/IP.

Опишемо основні критерії якими ми будемо керуватися під час вибору програмного забезпечення:

- 1) Актуальність на момент розробки, та перспектива підтримки в майбутньому.
- 2) Популярність, мультиплатформеність.
- 3) Безпека програмного забезпечення, регулярні оновлення патчів безпеки.

- 4) Час розробки проєкту за допомогою обраного ПЗ. Як дане ПЗ прискорить розробку та мінімізує ресурсні затрати, такі як час, гроші та ін.
- 5) Простота керування контентом після розробки, адже після введення в експлуатацію, наповнення сайту, можливо буде створюватись людиною без технічної освіти чи досвіду роботи з нею.

1.4.1. Вибір клієнтських мов програмування

Якщо дивитись на клієнтські мови програмування то вони майже не мають аналогів та конкурентів, тому будь обрані програмні засоби описані нижче.

1.4.1.1. HTML

Найпоширенішим та майже єдиним базовим засобом для створення веб-сайту є мова програмування HTML. HTML — це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа. Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами, написаними з використанням кутових дужок.

Ми будемо використовувати найбільш актуальну версію HTML 5.

1.4.1.2. CSS

CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі. CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

В даному проекті буде використана найбільш нова версія CSS 3.

1.4.1.3. JavaScript

JavaScript (JS) - динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки.

Данна мова програмування дасть можливість створити веб-додаток який буде динамічним, тобто вміст сторінки буде змінюватись без її перезавантаження.

1.4.2. Серверні мови програмування

Серед серверних мов програмування найбільш поширеними є:

1.4.2.1. ASP.NET Core

ASP.NET Core - це фреймворк для створення серверної частини сучасних веб-сайтів, що дозволяє визначити комплексну архітектуру програми, що володіє цілим рядом унікальних можливостей і переваг в порівнянні з іншими технологіями подібного роду інших платформ.

В основі ASP.NET Core лежить мова програмування, яка встигла себе зарекомендувати з хорошого боку - C#. У розпорядженні ASP.NET-програміста будуть такі потужні інструменти, як LINQ, лямбда-вирази і безліч різного "синтаксичного цукру", значною мірою спрощує процес розробки - в основному за рахунок збільшення швидкості написання коду і менших витратах на підтримку. Також при бажанні розробник легко зможе компонувати інші мови .NET групи, а саме: Visual Basic і F#.

Приставка Core говорить про те, що фреймворк виконаний на базі найновішої платформи, реліз якої ознаменував перехід до нової ери розробки під егідою Microsoft. Йдеться про стандарт .NET Core.

.NET Core - це новий стандарт розробки, що прийшов на зміну .NET Framework. І хоча у .NET Core немає мети витіснити .NET Framework з ринку програмного забезпечення, багато фахівців відразу ж помітили в Core-версії великий потенціал і почали використовувати його у своїй роботі.

1.4.2.2. Node Js

Node Js – це програмний засіб для виконання JavaScript, на стороні сервера. асинхронна одно-нитева модель виконання запитів;

Node.js має наступні властивості:

- 1) Асинхронна одно-нитева модель виконання запитів;
- 2) Неблокуючий ввід/вивід;
- 3) система модулів CommonJS;
- 4) рушій JavaScript Google V8;

Для керування модулями використовується пакетний менеджер npm (node package manager).

1.4.2.3. PHP

PHP – це скриптова мова програмування, яка створена для роботи на сервері, генерації HTML – коду та роботи з базою даних. PHP входить в список мов програмування з динамічною типізацією, тобто тип змінною визначається не під час створення змінної а піз час її присвоєння. PHP - мова, що інтерпретується. Написані на ньому програми інтерпретуються в момент звернення за допомогою спеціальних програм. Інтерпретовані мови не залежать від платформи, але поступаються компільованим мовам в швидкості виконання. PHP відноситься до мов загального призначення. Проте ця мова перш за все застосовується для розробки веб-додатків. Це серверна мова, тобто він виконується на сервері. Написані на PHP програми отримують дані від користувачів сайту, обробляють їх, взаємодіють з базами даних, повертають на сайт оброблену інформацію.

На PHP написані популярні системи управління контентом (CMS), наприклад, WordPress, Joomla !, Drupal. На одному тільки WordPress працює близько третини від усіх сайтів в інтернеті. Це підтверджує актуальність і популярність PHP. Цією мовою написані і популярні фреймворки для створення сайтів, наприклад, Laravel, Yii2, Symfony. Нарешті, такі масштабні і популярні проекти, як «Facebook» і «ВКонтакте», теж створені на PHP. Більш як 75% всіх сайтів в інтернеті створені за допомогою PHP.

1.4.2.4. Вибір серверної мови програмування

Враховуючи всі, аргументи, які описані вище, мій вибір мова програмування PHP. Вона відповідає всім критеріям для даного проекту, і дозволяє виконати всі завдання, з найменшими затратами часу.

1.5. Список систем управління

Як відомо, для розробки сайту є декілька шляхів. Можна створювати сторінки з допомогою *HTML*, а можна застосувати мови програмування *.NET*, *Node Js* чи *PHP*, та базу даних. З їх допомогою можна створити сайт. Проте, є деякі нюанси, які слід враховувати при розробці сайту. Наприклад, час який потрібно буде затратити для написання коду сайту у випадку його розробки від нуля. Тому тут слід проаналізувати чи є сенс затрачати на таке ресурси, чи краще використати *CMS* - систему керування сайтом. Для вибору систем управління на основі яких можна створювати функціональні і легко керовані сайти без серйозних фінансових витрат.

1.5.1. Порівняння систем управління вмістом і вибір найбільш важливих аспектів

Для розробки сайту використовують спеціальне програмне забезпечення. Розглянемо декілька систем керування вмістом.

CMS (Content Management System) – це комп'ютерна програма або система, яку використовують для забезпечення і організації сумісного процесу створення, редагування і керування вмістом сайту (текстовими, графічними чи мультимедійними елементами). Вміст *CMS* розглядається як сукупність неструктурованих даних предметного завдання на противагу до структурованих даних сайтів, які знаходяться під керуванням системи управління базою даних (СУБД).

CMS має панель управління, яка є лише частиною системи, але достатньою для керування сайтом. Існують різноманітні системи управління сайтом, що створені за різними технологіями, серед яких є платні і безкоштовні.

Основні завдання *CMS* зібрати і об'єднати до єдиного цілого різнотипні джерела знань і інформації, які є доступними як всередині організації, так і за її межами.

Забезпечити взаємодію співробітників, робочих груп і проектів із створеними ними базами знань, інформацією і даними так, щоб їх легко можна було знайти, витягнути і повторно використати в звичний для користувача спосіб.

Більша частина сучасних систем управління вмістом реалізується у вигляді візуального редактора — програми, яка створює *HTML*-код із спеціальної спрощеної розмітки, що дозволяє користувачеві легко форматувати текст.

Центральним елементом будь-якої *CMS* є сховища інформації. В сучасних системах управління контентом - це реляційна база даних. Слово «реляційна» вказує на те, що база складається з таблиць, між якими уставлені відносини (*relation* — «відношення»).

Якщо в *CMS* необхідно зберегти певну інформацію, вона записує її в базу даних. Для кожної сутності в базі даних відведено окрему таблицю.

Наприклад, таблиця, яка зберігає вміст веб-сторінок. В ній, окрім тексту сторінки, зберігається назва матеріалу, дата створення і відомості про автора. Програмний рушій бази даних вибирається в залежності від платформи. Якщо використовується платформа *Windows*, то це *MSSQL*, якщо *UNIX* платформа, то *MYSQL*.

Після вибору бази даних варто визначитися, як краще запрограмувати роботу з нею в *CMS*. Кращим підходом є створення абстрактного проширення роботи з базою даних. Реалізувати його можна як у вигляді спеціального класу,

так і у вигляді набору функцій. В ідеалі основний код CMS має бути однаковим для обох баз даних, міняється лише код для бази даних.

Інформацію, яку необхідно відобразити (наприклад, текст статті), CMS отримує з бази даних. Для відображення інформації у форматі *HTML* використовується механізм шаблонів. Шаблон є файлом з дизайном сторінки, що створено засобами спеціальної мови. Зазвичай, це певним чином розмічений код *HTML*, в якому вказано, де треба вставляти назву сторінки, де — основний текст, де — меню чи інші елементи, які беруться з бази даних. Найпростішим варіантом буде створення шаблону мовою *PHP*.

1.5.1. Система керування сайтом WordPress

WordPress - вільне програмне забезпечення, яке можна використовувати для створення красивого сайту, блогу чи веб-додатку.

Випуск нових версій *WordPress* не реалізуються. Якщо виявлено будь-який недолік то розробники оперативно випускають оновлення.

У використанні *WordPress* для створення сайтів електронної комерції є свої переваги:

1) велика кількість готових тем. Для магазину можна підібрати тему із величезної кількості. При цьому, любую тему можна відредагувати.

2) зручні та доступні плагіни. Багато корисних плагінів для *WordPress* у вільному доступі. Можна використовувати для покращення функціональності системи та розширення можливостей.

Недоліки цієї системи керування:

1) велике навантаження на сервер. Якщо магазин проектується на велику кількість відвідувачів, то з обробкою замовлень можуть виникати проблеми. Від великої кількості діючих скриптів піднімається навантаження на сервер. В результаті чого можуть бути як збої так і «падіння».

2) для налаштування до потреб інтернет магазину, потрібно встановлювати плагіни та теми. При спробі оновлення *WordPress* з роботою

системи можуть виникнути складності. Справа в тому, що більшість плагінів і тем перестають нормально працювати після оновлення платформи.

Система *WordPress* створювалась в якості платформи для блогів. При розробці порталу для новин, чим більший функціонал потрібний, тим більше додаткових плагінів встановлюється. В чому виникає залежність, чим більше плагінів, тим більше навантаження на сервер та бази даних.

1.5.2. Система керування сайтом Magento

Magento – система керування сайтом з відкритим кодом, призначена для створення інтернет магазинів. Magento одна з найпопулярніших систем з відкритим кодом для організації електронної комерції на базі цієї платформи створено понад 100 тис. інтернет магазинів, та майже 2 тисячі розширень створили сторонні розробники, код платформи був навантажений більше 2 млн. разів, через данну платформу було продано товарів на більш ніж 25 млрд доларів. Перша версія Magento була випущена в 2007 році, платформа отримує оновлення до нових версій і досі. У березні 2010 року компанія eBay придбала 49% відсотків якцій компанії, чим прискорила розвиток данного проекту.

Але данна система, створена швидше для інтернет магазинів тому для цього проекту підходить не дуже.

1.5.3. Система керування сайтом Bitrix

Bitrix - система керування сайтом, яка була створена в 2003 році, це професійна система управління веб-проектами, універсальний програмний продукт для створення, підтримки та успішного розвитку. Продукт забезпечує високий рівень захищеності сайтів від злону. Якість реалізації комплексу «Проактивний захист»: проактивний фільтр, вбудований веб-антивірус, сканер безпеки, одноразові паролі (ОТР) та інші технології захисту надають впевненість не тільки в надійності ядра системи, а й в безпеки інтернет-рішення на її основі,

з урахуванням надбудов та доопрацювань. «Бітрікс» надає своїм клієнтам безкоштовну можливість автоматичного «хмарного» бекапу. Сервіс Прискорення сайту (CDN) дозволяє завантажувати весь статичний контент (картинки, файли стилів css, скрипти js) через мережу дистрибуції контенту (Content Delivery Network або Content Distribution Network, CDN).

Початкові вимоги:

- 1) Необхідний мінімум знань для вивчення курсу:
- 2) базові навички роботи з комп'ютером та ОС Windows;
- 3) базові знання WWW та організації доступу до веб-сервера;
- 4) знання системи в рамках курсу Content Manager, щоб не заплутатися в інтерфейсі.
- 5) Як плюс базові навички встановлення та адміністрування систем nginx.

Bitrix має такий функціонал:

- 1) Інтернет-магазини
- 2) Корпоративні сайти
- 3) Інформаційні портали
- 4) Сторінки спільнот, форуми
- 5) Рекламні лендінги

1.6. Вибір системи управління

Проаналізувавши недоліки та переваги описаних CMS систем, було обрано систему Vitrix, яка дозволяє швидко почати роботу має вбудовані модулі для новин, отримує регулярні оновлення та патчі безпеки.

1.7. Вибір бази даних

База даних (англ. Database) - сукупність даних, організованих відповідно до концепції, що описують характеристики даничів і взаємозв'язок між елементами; ця сукупність буде додана в одну із зон перевантаженості (відповідно до стандарту ISO/IEC 2382:2015 [1]). У загальному випадку база даних помсти схем, таблиць, подання, збережених процедур та інші об'єкти. Дані з бази організовані відповідно до моделі організації даних. З таким званням, основою данини, даних так щастить, щоб помститися їм, так само описують, щоб помститися їм за їх обробку.

1.7.1. Види баз даних

База даних (БД) — це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів. У технічному розумінні включно й система управління БД.

Система управління базами даних (СУБД) - це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації.

Баз даних та систем управління базами даних існує величезна кількість, понад п'ятдесят видів баз даних.

Наведемо основні класифікації.

Класифікація за моделлю даних:

- 1) Ієрархічні;
- 2) Мережеві;
- 3) Реляційні;
- 4) об'єктні;
- 5) об'єктно-орієнтовані;
- 6) об'єктно-реляційні;

Класифікація Баз Даних за технологією фізичного зберігання:

- 1) БД у вторинній пам'яті (традиційні);
- 2) БД в оперативній пам'яті (in-memory databases);
- 3) БД у третинній пам'яті (tertiary databases).

Класифікація БД за вмістом:

- 1) географічні.
- 2) історичні.
- 3) наукові.
- 4) мультимедійні.

Класифікація БД за ступенем розподіленості:

- 1) централізовані (зосереджені);
- 2) розподілені.

1.7.1.1. Структуровані та неструктуровані БД

Структуровані бази даних використовують структури даних, тобто структурований опис типу за допомогою схеми даних, такої як модель даних. Об'єкти та взаємовідносини описує модель даних. Існує декілька моделей (чи

типів) баз даних, основні: ієрархічна, мережна та реляційна. Приблизно з 2000 року більше половини БД використовують реляційну модель.

До *неструктурованих* БД відносяться повнотекстові бази даних, які містять неструктуровані тексти статей чи книг у формі, що дозволяє здійснювати швидкий пошук (як наприклад вікіпедія).

1.7.1.2. Ієрархічна база даних

Інформація в базах даних поділена на окремі записи, якими називають групу пов'язаних між собою елементів даних. Характер зв'язку між записами визначає два основних типи організації баз даних: ієрархічний і реляційний. В ієрархічній базі даних елементи впорядковуються в певну послідовність.

Ієрархічна база даних за своєю структурою відповідає структурі ієрархічної файлової системи. Такі системи представлені як дерево, з об'єктів різних рівнів. Верхній рівень займає один об'єкт, другий – об'єкти другого рівня, і так по порядку.

1.7.1.3. Реляційна база даних

Реляційна база даних являє собою двовимірну таблицю. Стовпці таблиці називаються полями: кожне поле характеризується своїм ім'ям і типом даних. Поле БД - це стовпець таблиці, що містить значення певної властивості. У реляційній БД використовуються чотири основних типів полів:

- 1) числовий, символічний (слова, тексти, коди і т.д.);
- 2) дата (календарні дати у формі «день / місяць / рік»);
- 3) логічний (приймає два значення: «так» - «ні»);

Така модель орієнтована на організацію даних у вигляді двовимірних таблиць. Кожна реляційна таблиця являє собою двовимірний масив і має наступні властивості:

- 1) кожен елемент таблиці - один елемент даних;
- 2) всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символний тощо);
- 3) кожен стовпчик має унікальне ім'я;
- 4) однакові рядки в таблиці відсутні;
- 5) порядок проходження рядків і стовпчиків може бути довільним.

1.7.1.4. Мережеві бази даних

Мережеві бази даних подібні до ієрархічних, за винятком того, що в них є покажчики в обох напрямках, які з'єднують споріднену інформацію.

До основних понять мережевої моделі бази даних відносяться: рівень, елемент (вузол), зв'язок.

Вузол - це сукупність атрибутів даних, що описують деякий об'єкт. На схемі ієрархічного дерева вузли представляються вершинами графа. У мережній структурі кожен елемент може бути пов'язаний з будь-яким іншим елементом.

Незважаючи на те, що ця модель вирішує деякі проблеми, пов'язані з ієрархічною моделлю, виконання простих запитів залишається досить складним процесом.

Також, оскільки логіка процедури вибірки даних залежить від фізичної організації цих даних, то ця модель не є повністю незалежною від програми. Іншими словами, якщо необхідно змінити структуру даних, то потрібно змінити і додаток.

1.7.1.5. Об'єктно-орієнтована база даних

Об'єктно-орієнтована база даних (ООБД) - база даних, в якій дані оформлені у вигляді моделей об'єктів, що включають прикладні програми, які управляються зовнішніми подіями. Результатом поєднання можливостей

(особливостей) баз даних і можливостей об'єктно-орієнтованих мов програмування є об'єктно-орієнтовані системи управління базами даних (ООСУБД). ООСУБД дозволяють працювати з об'єктами баз даних також, як з об'єктами у програмуванні в об'єктно-орієнтованих мовах програмування. ООСУБД розширює мови програмування, прозоро вводячи довготривалі дані, управління паралелізмом, відновлення даних, асоційовані запити й інші можливості.

Об'єктно-орієнтовані бази даних звичайно рекомендовані для тих випадків, коли потрібна високопродуктивна обробка даних, які мають складну структуру.

Система, яка забезпечує об'єктну інфраструктуру і набір реляційних розширювачів, називається "об'єктно-реляційною".

Об'єктно-реляційні системи поєднують переваги сучасних об'єктно-орієнтованих мов програмування з такими властивостями реляційних систем як множинні представлення даних і високорівневі непроцедурні мови запитів.

1.7.1.7. Централізовані бази даних

За технологією обробки даних бази даних поділяються на централізовані й розподілені.

Централізована база даних зберігається у пам'яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом мережі ЕОМ, можливий розподілений доступ до такої бази. Такий спосіб використання баз даних часто застосовують у локальних мережах ПК.

1.7.1.8. Розподілені бази даних

Розподілені бази даних (РБД) — це сукупність логічно пов'язаних між собою БД, які є розподіленими у комп'ютерній мережі.

Переваги Розподілених баз даних:

- 1) ця модель відображає інформацію в найбільш простій для користувача формі;
- 2) заснована на розвиненому математичному апараті, який дозволяє досить лаконічно описати основні операції з даними;
- 3) дозволяє створювати мови маніпулювання даними не процедурного типу;
- 4) маніпулювання даними на рівні вихідний БД і можливість зміни.

Недоліки Розподілених баз даних:

- 1) найповільніший доступ до даних;
- 2) трудомісткість розробки.

1.7.2. Види систем управління базами даних

Система управління базою даних — це програмні засоби, які виступають посередником між БД та її користувачами. Завдяки сукупності мовних та програмних засобів, системи управління базами даних сприяють створенню, веденню та спільного використання БД різними користувачами.

Сучасна програма система управління базою даних складаються з ядра, процесору мови БД, підсистеми підтримки часу виконання та сервісних програм, які надають додаткові можливості для обслуговування інформаційних систем.

Вибір правильної системи управління базою даних — це одне із ключових питань при створенні свого ресурсу. Рекомендується орієнтуватися

на завдання, які має виконувати сайт, а також враховувати характер та обсяги даних, які система повинна обробляти.

Коротко зупинимося на конкретних програмних продуктах, що відносяться до класу СУБД.

На найзагальнішому рівні усі СУБД можна розділити:

- 1) професійні, чи промислові;
- 2) персональні.

Професійні (промислові) СУБД представляють собою програму основу для автоматичних систем управління великими економічними даними. На їх основі використовувались великі управлінські комплекси інформації про великі підприємства, банки або навіть цілі галузі.

Першочерговими умовами, які задовольняють професійні СУБД, є:

- 1) можливість організації спільної та паралельної роботи масивної кількості користувачів;
- 2) масштабованість, вірогідність зростання системи пропорційно розширенню керованого підприємства;
- 3) стабільність по відношенню до відмов різного типу, включаючи наявність багаторівневої системи резервного копіювання збереженої інформації;
- 4) забезпечення безпеки збережених даних та добре розробленої структурованої системи доступу до них.

Промислові СУБД наразі мають досить багату історію розвитку. Зокрема, можна зазначити, що наприкінці 70-х - початку 80-х років СУБД Adabas активно використовувався в автоматизованих системах, побудованих на базі великих комп'ютерів. В даний час типовими представниками професійних СУБД є програмні продукти, такі як Oracle, DB2, Sybase, Informix, Ingres, Progress.

Системи управління персональними даними - це програмне забезпечення, яке фокусується на завданнях місцевого користувача або компактної групи користувачів і призначене для використання на середньостатистичному персональному комп'ютері. Це також пояснює їх другу назву – настільні, або для робочого столу. Визначальними характеристиками настільних систем є:

- 1) відносна простота використання, що дозволяє створювати працездатні програми на їх основі як "просунутим" користувачам, так і тим, у кого кваліфікація низька;
- 2) відносно обмежені вимоги до обладнання.

Історично першою серед персоналізованих СУБД, яка набула широкого поширення, була Dbase від Ashton-Tate (пізніше права на неї були передані Borland, а з 1999 року ця програма підтримується dBASE Inc.). Згодом ряд реляційних персональних СУБД поповнився такими продуктами, як FoxBase / FoxPRO (Fox Software, далі - Microsoft), Clipper (Nantucket, потім Computer Associates), R: base (Microrim), Paradox (Borland, в даний час володіє правами Corel), Access (Microsoft), Approach (Lotus).

Системи Dbase, FoxPRO і Clipper, які набули широкої популярності в СНГ, працювали з реляційними таблицями даних, розташованими у файлах, що мають розширення .dbf (термін dbf формат став загальноприйнятим). Згодом сімейство цих баз даних отримало інтегровану назву Xbase.

Незважаючи на неминучі відмінності внаслідок намірів розробників, усі ці системи в процесі своєї еволюції придбали ряд загальних конструктивних особливостей, серед яких, насамперед, можна назвати:

- 1) наявність візуального інтерфейсу, який автоматизує процес створення засобів маніпулювання даними - екранні форми, шаблони звітів, запити тощо;
- 2) наявність інструментів для створення об'єктів баз даних у режимі діалогу: Wizards в Access, Experts в Paradox, Assistants в Approach;

- 3) Подібність розроблених інструментів для створення розширень програмного забезпечення в одному середовищі СУБД: мова розробки додатків PAL у Paradox, VBA (Visual Basic для додатків) у Access, Lotus Script у Approach;
- 4) Вбудована підтримка універсальних мов управління даними, таких як SQL або QBE (Query By Example).

Системи управління базами даних, які, відносно кажучи, займають проміжне місце між настільними та промисловими системами, можна назвати SQLWindows / SQLBase від Centura (до 1996 року Gupta), InterBase (Borland), Microsoft SQL Server і нарешті MySQL.

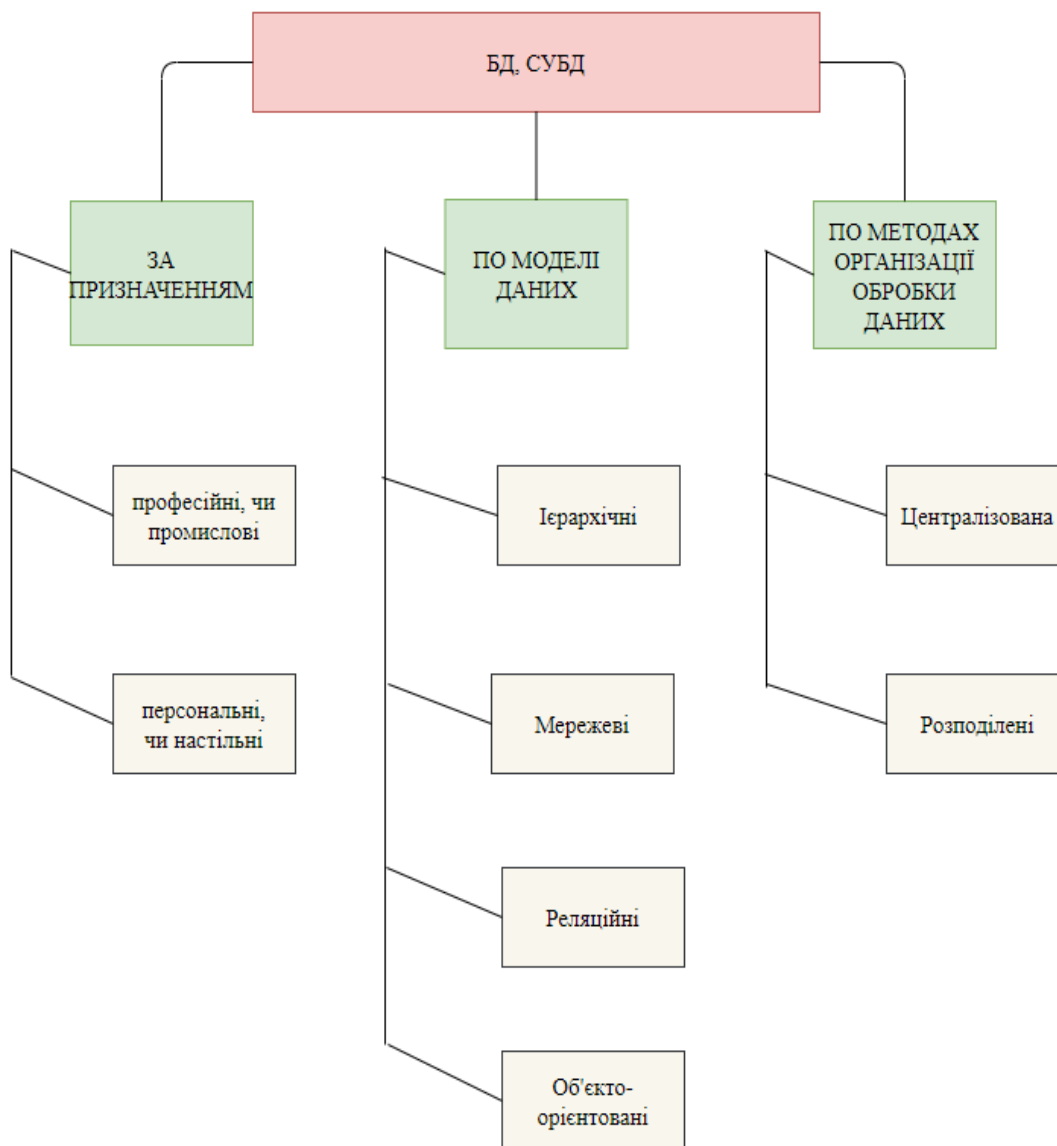


Рис. 1.2. Класифікація СУБД та БД

1.7.2.1. Система управління базами даних Oracle

Засновниками Oracle СУБД були групи американських розробників (Ларрі Еллісбн, Роберт Майнер та Едвард Оутс), які створили Relational Software Inc. понад двадцять років тому. і поставили перед собою завдання створити систему, яка на практиці реалізує ідеї, викладені в роботах Е. Ф. Кодда і К. Дж. Дані. Результатом їх діяльності стала впровадження

портативної системи управління реляційною базою даних з базовою мовою обробки SQL. У 1979 році клієнтам було представлено версію Oracle для міні-комп'ютерів PDP-11 від корпорації Digital Equipment Corporation для декількох операційних систем: RSX-11, IAS, RSTS та UNIX. Трохи пізніше Oracle був перенесений на комп'ютери VAX, на яких працює VAX VMS. Значна частина коду була написана в асемблері, і тому процес перенесення системи на нову платформу вимагав значних зусиль. Основна відмінність Oracle від наступної, третьої версії полягала в тому, що вона повністю написана на C. Це рішення забезпечило переносимість системи на багато нових платформ, зокрема, до різних клонів UNIX. Другою важливою особливістю нової (1983 р.) Версії була підтримка концепції транзакцій. Приблизно в той же час компанія отримала нову назву - Oracle Corporation - і зайняла лідируючу позицію на ринку для виробників СУБД. Четверта версія Oracle характеризувалася розширенням списку підтримуваних платформ і операційних систем. Oracle був перенесений як в мейнфрейми IBM (мейнфрейми), так і на персональні комп'ютери, на яких працює MS DOS. Саме в четвертій версії був зроблений важливий крок у розробці технології підтримки цілісності баз даних. Для багатокористувацьких систем запропоновано оригінальне рішення Oracle для послідовності читання. У п'ятій версії вперше була реалізована СУБД з архітектурою клієнт-сервер. Подальші версії СУБД Oracle були зосереджені на побудові масштабних систем обробки транзакцій, зміні методів впровадження вводу-виводу, буферизації, паралельного контролю доступу, підсистем резервного копіювання та відновлення. Також була реалізована підтримка симетричних багатопроцесорних архітектур.

1.7.2.2. Система управління базами даних Ingres

Дизайн та експериментальна версія бази даних Ingres була розроблена в університеті Берклі під керівництвом одного з найвідоміших вчених та експертів із бази даних Майкла Стоунбрейкера. З самого початку СУБД Ingres

розроблялася як мобільна система, що працює в середовищі ОС UNIX. Перша версія Ingres була розроблена для 16-бітних комп'ютерів і працювала переважно на машинах серії PDP. Це перша СУБД, яка розповсюджувалася безкоштовно для використання в університетах. Згодом команда Stonebreaker перенесла Інгрес у середовище ОС UNIX BSD, яке також було розроблено в університеті Берклі. Сімейство СУБД Інгреса з університету в Берклі зазвичай називають університетом Інгреса. На початку 80-х років було утворено RTI (Relational Technology Inc.), яке розробило і почало просувати комерційну версію Ingres. В даний час комерційний Ingres підтримується, розробляється та продається Computer Associates. Зараз це одна з найбільш розвинених комерційних реляційних СУБД. У той же час існує багато якісних публікацій про університет Інгрес. Більше того, університет Інгрес може бути перевірений на практиці і навіть побачити його вихідний код.

1.7.2.3. Система управління базами даних MySQL

MySQL - безкоштовна система управління табличними тобто реляційними базами даних. MySQL був розроблений «ТсХ» для збільшення швидкості обробки великих баз даних. Дана система управління базами даних з відкритим кодом (СУБД), була створена як альтернатива промисловим системам управління базами даних. MySQL був дуже схожий на mSQL з самого початку, але з часом він виріс, і зараз MySQL є однією з найпопулярніших систем управління базами даних. В основному використовується для створення динамічних веб-сторінок, оскільки має чудову підтримку з різних мов програмування.

Можливості сервера MySQL:

- 1) простота установки та використання;
- 2) підтримується безкінечна кількість юзерів та рограм, які одночасно працюють із базою даних;
- 3) необмежена кількість рядків у таблицях;
- 4) висока швидкість виконання запитів;
- 5) проста та ефективна система безпеки.

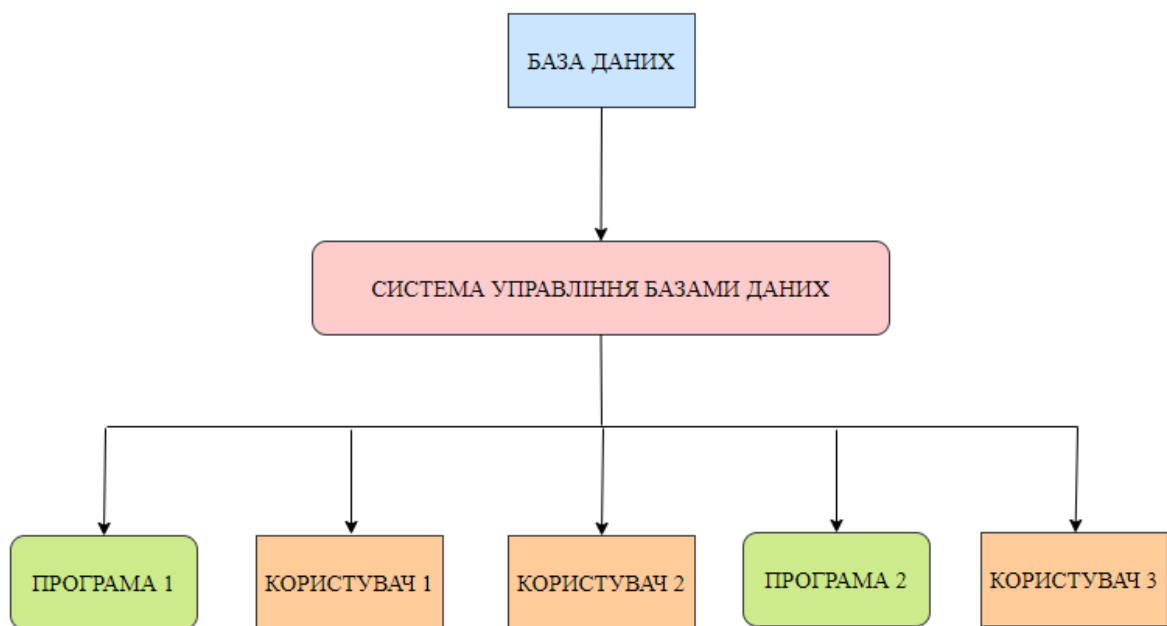


Рис. 1.3. Система управління базами даних

MySQL підтримує мову запитів SQL в стандарті ANSI 92, а також має багато розширень цього стандарту, яких немає в жодній іншій базі даних.

1.8. Обмін даними з постачальником

На даний момент найбільш популярними є два формати обміну даними, XML та JSON. Вибір адекватного формату обміну даними може мати серйозні наслідки для швидкості передачі та продуктивності. Ми опишемо мовні специфікації та відповідні їм налаштування використовувати. Потім проводиться тематичне дослідження для порівняння використання ресурсів та відносна продуктивність програми, які використовують формати обміну. Ми знайшли, що JSON значно швидше XML, і ми далі записувати інші показники, пов'язані з ресурсами, на нашу отримані результати.

Як JSON, так і XML є текстовими форматами, які можна читати і підтримують створення, читання та декодування в реальних програмах. Обидва є ієрархічними та незалежними від мови для обміну даними. Хоча вони схожі, вони відрізняються за багатьма аспектами, такими як типи даних, багатослівність, стек інструментів тощо. Хоча XML — це мова текстової розмітки, яка спеціалізується на ділових операціях у всесвітній мережі, JSON - це легкий відкритий стандартний формат для обміну даними, який розширюється за допомогою JavaScript. XML означає "розширена мова розмітки" і записується так само, як і в HTML, а json для "нотації об'єкта JavaScript", яка є підмножиною синтаксису мови JavaScript і повністю не залежить від мови.

1.8.1. XML

Якщо ви знайомі з HTML, ви дізнаєтесь XML – скорочення від «eXtensible Markup Language». У поєднанні з визначенням схеми XML (XSD), яке визначає, як має бути структурований XML-документ, дані, якими обмінюються дві системи, можуть перевірятися на узгодженість та

завершення на льоту, знімаючи цю відповідальність з коду, який оброблятиме ці дані.

XML (скорочення від Extensive Markup Language) — це формат текстових даних, отриманий від SGML (ISO 8879) і написаний подібним чином, за яким слідує HTML. Формат XML існує вже багато років і в першу чергу був розроблений для вирішення завдань великомасштабного електронного видавництва.

Те, що він просто робить, це аутсорсинг даних. Він зберігає дані у текстовому форматі, а не інтегрує їх у документ HTML, що робить його ідеальним для представлення ієрархічних даних, таких як документи, транзакції, транзакції, рахунки-фактури, книги тощо.

Це незалежний обмін даними у форматі, який кодує документи у форматі, який читається машиною та людиною. Це гнучкий спосіб створення інформаційних форматів та обміну структурованими даними у всесвітній мережі.

Насправді це підмножина мови `sgml` (стандартної узагальненої мови розмітки), подібної до HTML, яка містить розмітку символів для опису вмісту сторінки, що дозволяє користувачам визначати власні мови розмітки.

Основною перевагою XML є те, що він незалежний від платформи, що означає, що користувачі можуть отримувати дані з інших програм, таких як SQL, і конвертувати їх у XML, а потім обмінюватися даними з іншими платформами. Простіше кажучи, це документоорієнтована технологія, яка надає можливість зберігати та відображати дані як у машинозчитуваному, так і в зрозумілому людині форматі.

Він більше схожий на мета-мову без внутрішньої семантики, що робить його ідеальним форматом для створення спеціальних даних і документування форматів інформації.

```

<?xml version="1.0" encoding="iso-8859-8" standalone="yes" ?>
<CURRENCIES>
  <LAST_UPDATE>2004-07-29</LAST_UPDATE>
  <CURRENCY>
    <NAME>dollar</NAME>
    <UNIT>1</UNIT>
    <CURRENCYCODE>USD</CURRENCYCODE>
    <COUNTRY>USA</COUNTRY>
    <RATE>4.527</RATE>
    <CHANGE>0.044</CHANGE>
  </CURRENCY>
  <CURRENCY>
    <NAME>euro</NAME>
    <UNIT>1</UNIT>
    <CURRENCYCODE>EUR</CURRENCYCODE>
    <COUNTRY>European Monetary Union</COUNTRY>
    <RATE>5.4417</RATE>
    <CHANGE>-0.013</CHANGE>
  </CURRENCY>
</CURRENCIES>

```

Рис. 1.4. Приклад XML

1.8.2. JSON

JSON (скорочення від JavaScript Object Notation) — це інший текстовий формат обміну даними, який використовує текстові та числові типи даних для представлення об'єктів. Це відкритий стандартний формат, заснований на підмножині мови програмування JavaScript і повністю не залежить від мови. Це спосіб передачі об'єктів даних, що складаються з масивів типу даних і пар атрибут-значення, між сервером і веб-браузером. Він використовує формат, зрозумілий людині, для представлення простих структур даних у веб-додатках на основі коду.

Завдяки своїй гнучкості JSON краще підходить для обміну даними між веб-додатками та веб-сервісами. Як мова розмітки, XML лише додає додаткову інформацію до звичайного тексту, тоді як JSON, як випливає з назви, є способом представлення об'єктів даних.

Він також використовується як у настільних, так і в серверних середовищах програмування. На відміну від XML, JSON використовує простий підхід до представлення структурованих даних без складних математичних позначень і алгоритмів, і його легко навчитися, що робить його ідеальним способом створення більш інтерактивних сторінок.

Як кажуть, проблема одного – перевага іншого. Синтаксис XML не містить семантики, але він є багатослівним, що означає, що його складність ускладнює використання для кожної програми. XML був розроблений для читабельності, але не для ефективності. Синтаксис JSON набагато компактніший із встановленою семантикою, що робить його кращим форматом даних над XML.

1.8.3. Порівняння JSON та XML

XML – це спрощена версія SGML, яка використовується для зберігання та представлення структурованих даних у форматі, який є машиночитаним і читаним людиною. Він призначений для покращення читабельності, оскільки це мова розмітки, яка додає додаткову інформацію до звичайного тексту. JSON, з іншого боку, є полегшеним форматом обміну даними, який використовується для представлення ієрархічних даних і заснований на синтаксисі об'єкта JavaScript.

1.8.3.1. Значення Json і XML

XML це аббревіатура - "Extensive Markup Language" і є документоорієнтованою технологією, яка використовується для кодування даних у зручному для людини форматі. Це гнучкий формат файлу, придатний для використання в Інтернеті. JSON означає JavaScript Object Notation, і, як випливає з назви, він заснований на мові програмування JavaScript.

1.8.3.2. Ціль Json і XML

XML був розроблений Консорціумом World Wide Web як добре документований відкритий стандартний формат, що містить набір правил щодо того, як шифрувати документи у форматі, який читається людиною та машиною. JSON був розроблений Дугласом Крокфордом як простий, легкий формат файлу для обміну даними.

1.8.3.3. Синтаксис JSON і XML

JSON не має початкових або кінцевих тегів, а синтаксис легший за XML, оскільки він орієнтований на дані з меншою надлишковістю, що робить його ідеальною альтернативою для обміну даними через XML. XML, з іншого боку, має більше символів для представлення тих самих даних.

1.8.3.4. Типи даних у JSON і XML

JSON – формат підтримує текстові та числові типи даних, включаючи цілі числа та рядки. Структуровані дані можуть бути представлені за допомогою масивів та об'єктів. XML має пряму підтримку типу масиву, але він підтримує багато типів даних, таких як числа, текст, зображення, графіки, діаграми тощо.

1.9. Швидкодія сайту

Швидкодію web-сайту доцільно перевіряти використовуючи безкоштовний сервіс Google Page Speed Insights. Це – це комплексний інструмент для визначення фактичної роботи та вибору ефективних шляхів оптимізації сайту. Удобний для використання як на комп'ютері, так і на

мобільному пристрої. Не показує абсолютну швидкість сторінки, аналізує ефективність динаміки завантажень і переглядів у браузері клієнта. При цьому враховуються тільки не залежні від типу факторів інтернет-з'єднань: JavaScript, CSS, структура HTML, конфігураційний сервер, розмір зображень і т.п. д.

Щоб отримати детальний звіт про ефективність, вам потрібно перейти на офіційний веб-сайт Page Speed Insights і вставити посилання на досліджуваний ресурс у рядок «Введіть URL-адресу веб-сторінки» (Рис. 1.5).

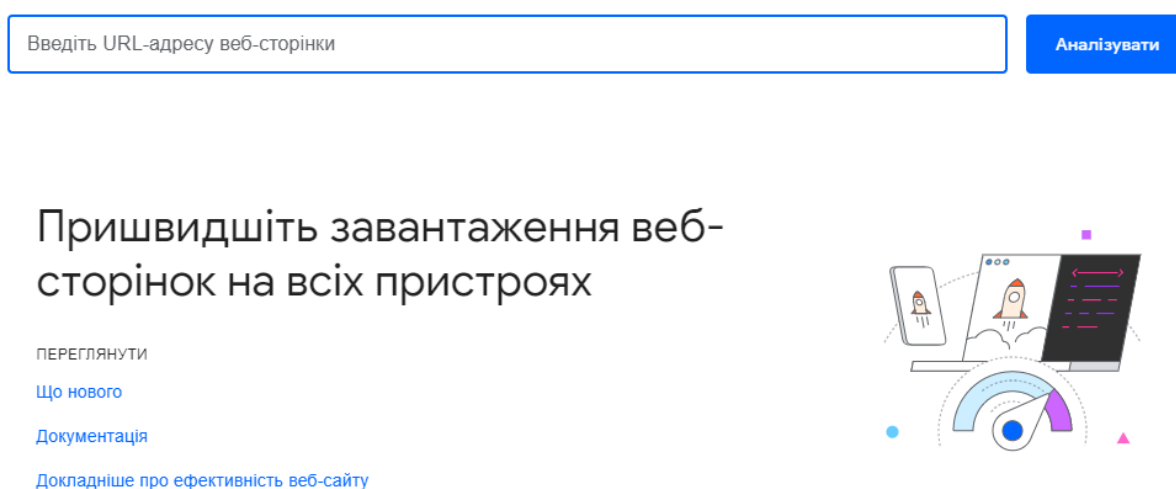


Рис 1.5. Page Speed Insights

Час підготовки та доставки результату займає менше хвилини. Користувач може відразу оцінити необхідність SEO оптимізації або переконатися в ефективності використаного алгоритму пошукової оптимізації сайту. Звіт про швидкість завантаження веб-сайту має графічні та цифрові компоненти, окремо для ПК та мобільних пристроїв.

Індикатор Page Speed (швидкість сторінки) демонструє динаміку в секундах від початку першої візуалізації вмісту (FCP) до моменту завантаження HTML і створення дерева DOM (DCL), тобто повного ініціалізація інтерфейсу для користувача. Чим менше цей час, тим швидше сайт буде відображатися в браузері. Щоб забезпечити рівноправність усіх

ресурсів, Google Page Speed Insights використовує середню швидкість завантаження в мережі як відправну точку, встановлюючи один із рейтингів:

- Повільний – повільний (червоний);
- Середній - нормальний (помаранчевий);
- Швидкий – швидкий (зелений).

Наявність графіка розподілу швидкості дозволяє легко визначити проблемні місця за швидкістю навантаження і вжити необхідних заходів. Чітке значення мають і кольорові графічні маркування.

Крім того, онлайн-сервіс Page Speed Insights використовує 100-бальну шкалу для оцінки веб-ресурсів за часом, необхідним для отримання результатів для пошукового запиту та повного завантаження сторінки. Всього існує три діапазони швидкості:

- Низький (низький) - 0-59 балів;
- Середній (середній) - 60-79;
- Добре (добре) - 80-100.

Низький рейтинг веб-ресурсу майже завжди свідчить про великі недоліки в організації процесу візуалізації, які необхідно усунути.

1.3.1. Оцінка швидкості сторінки та просування SEO

Google використовує близько 200 факторів для визначення рейтингу сайту, серед яких найважливішими є ключове слово на початку тегу заголовка, обсяг і відповідність вмісту пошуковим запитам, щільність і розташування ключових слів, а також абсолютне значення швидкості завантаження. Показник Google Page Speed Insights безпосередньо впливає на видачу 1–3 результатів із 100. При цьому в офіційних вимогах системи зазначено, що «сайт повинен працювати максимально швидко». Цей фактор позитивно впливає на історію поведінки веб-ресурсу та значно підвищує його рейтинг.

Ось чому ефективно просування сайту SEO вимагає звичайної оцінки швидкості сторінки, але не обов'язково добре та швидко. Яскравим підтвердженням цього факту є мегапопулярний ресурс Amazon.com із середнім рівнем оптимізації та високим рейтингом.

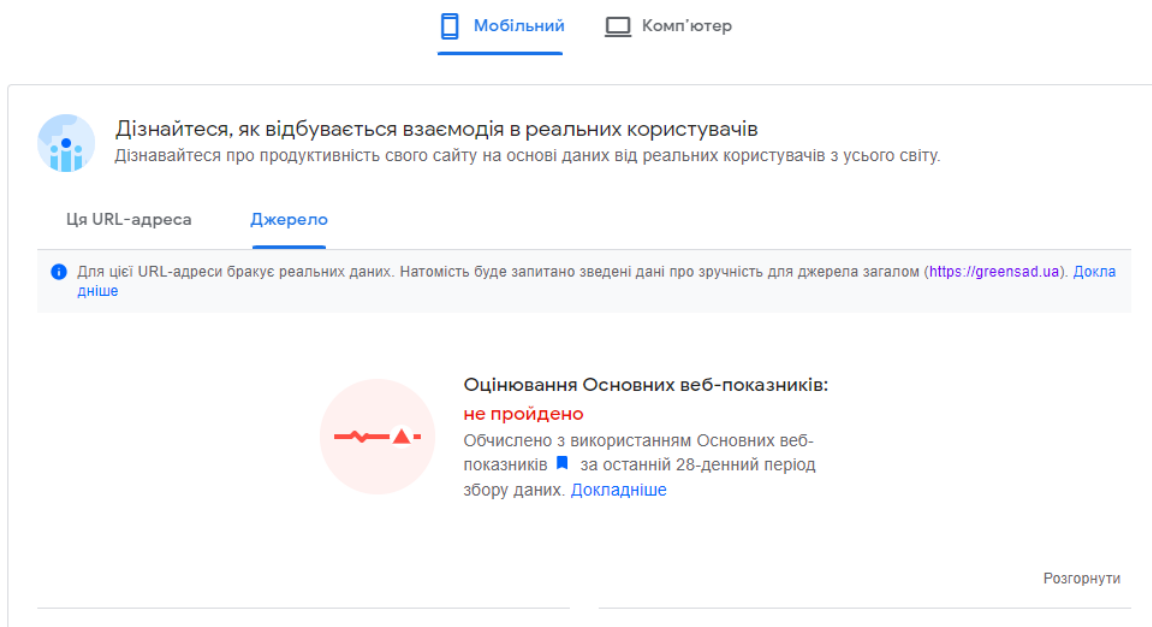


Рис. 1.6. Оцінка мобільної версії сайту

Крім детального звіту про роботу веб-ресурсу, сервіс також пропонує різні варіанти його покращення. Наприклад, Amazon.com отримав наступні пропозиції щодо оптимізації.

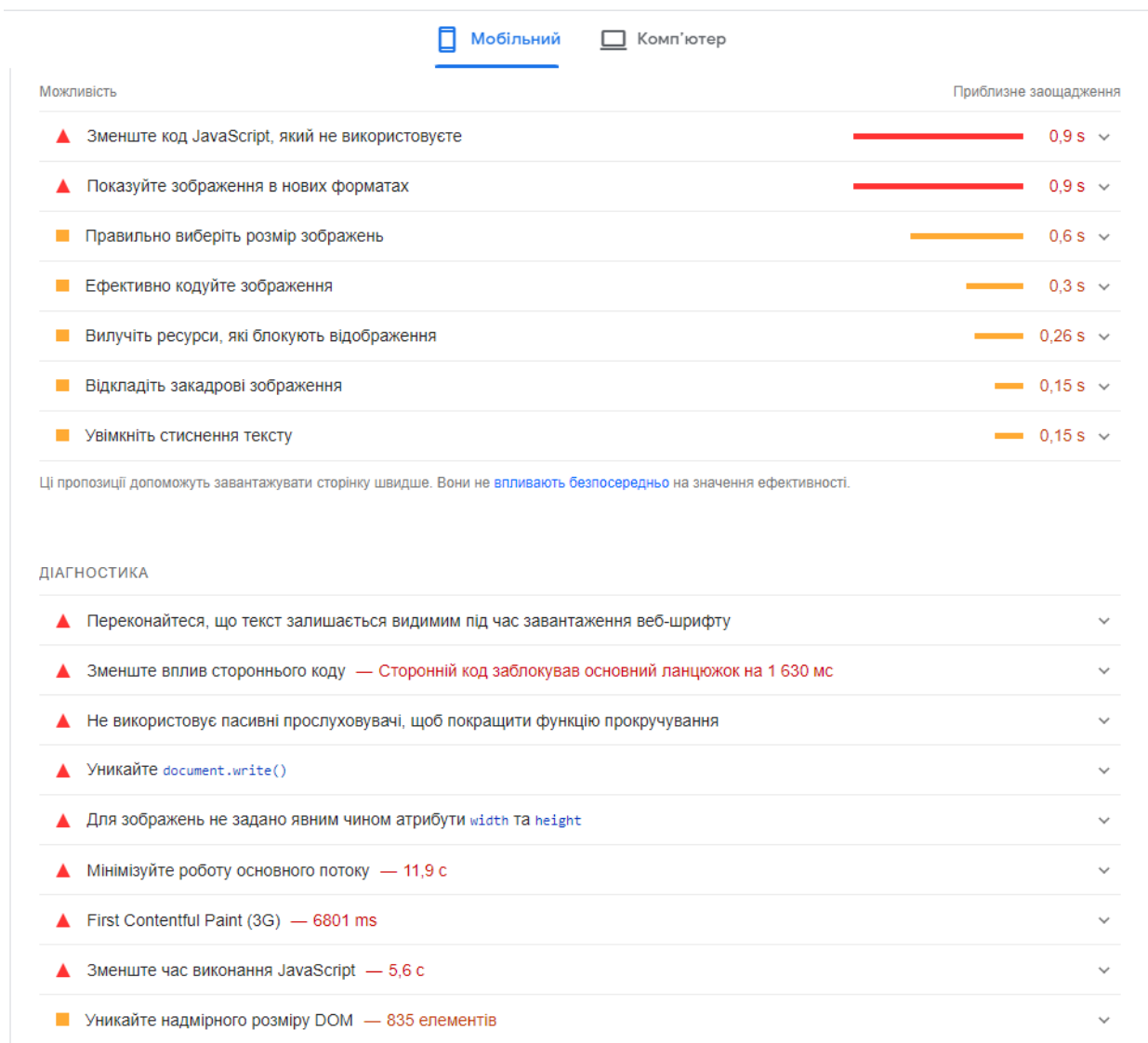


Рис. 1.7. Оцінка сайту Amazon та рекомендації до оптимізації

Використання кешу браузера. Практична та корисна порада щодо використання заголовків кешу для зберігання копій статичних файлів (не змінюйте один пошуковий запит на інший) у браузері користувача.

Можливо, в списку посилань є ресурс, хостинг якого реалізований на сторонньому сайті, що унеможлиблює виконання цієї рекомендації.

Видалення блокування відтворення JavaScript. Якщо можливо, організуйте виконання сценаріїв інтерактивних веб-сторінок у нижньому колонтитулі сайту або зробіть асинхронне завантаження заголовка та JavaScript. Також частину коду можна перенести в основну HTML-розмітку. Тут важлива участь професіонала, щоб не порушити коректну роботу

функціоналу сайту. Ця рекомендація не підходить для ресурсів, які використовують попередньо створений шаблон WordPress.

Оптимізація каскадної таблиці стилів. Є можливість розділення CSS-коду для пріоритетного вивантаження стилів оформлення першого екрану в заголовку сайту та додавання решти каскадних таблиць до нижнього колонтитула. Також потрібна участь верстальника, щоб уникнути проблем у роботі сайту в майбутньому.

Стиснення зображень. Ефективний спосіб збільшити швидкість завантаження веб-сайту – оптимізувати розмір зображень і фотографій. Для цього можна скористатися редактором Adobe Photoshop або спеціалізованими онлайн-сервісами: Compressor.io, TinyPNG, Optimizilla, Web Resizer. Рекомендована ширина зображення у форматі jpeg, gif або png 400–600 пікселів і вага 35–75 Кб.

Безкоштовний інструмент Google Page Speed Insights – швидкий, простий у використанні, детальний аналіз продуктивності сайту та практичні поради щодо підвищення швидкості завантаження. Використовуючи цей сервіс, ви зможете покращити конверсію свого веб-ресурсу та уникнути ситуації з відходом користувача через затримку рендерингу або тривалого завантаження зображень.

РОЗДІЛ 2 СТВОРЕННЯ САЙТУ

2.1. Формат діяльності підприємства

На першому етапі розробки сайту проаналізуємо предметну область інтернет магазину.

Предметна область - інтернет магазин. Це приватний веб-портал, для продажу рослин. Особливістю даного підприємства є відсутність складу для зберігання товарів, та жодного контакту з товаром. Дана система є тестовою та створена для демонстрації переваг дропшипінгу перед класичним методом закупки товарів під реалізацію через інтернет чи офлайн. Основною ідеєю є створення сайту, його SEO-просування тобто збільшення унікального трафіку, та збільшення продажів товару онлайн.

Дропшипінг - це спосіб виконання замовлень, при якому магазин не зберігає продані товари на складі. Натомість магазин купує товар у стороннього постачальника і відправляє його клієнту. Як наслідок, продавцю не потрібно безпосередньо обробляти товар.

Найбільша відмінність між дропшипінгом і стандартною моделлю роздрібної торгівлі полягає в тому, що продавець не має запасів і не володіє ними. Замість цього продавець закупає запаси у третьої сторони - зазвичай оптового продавця або виробника - для виконання замовлень.

Кафедра КІТ (47)				НАУ 21 32 00 000 ПЗ			
Виконав	Лазоренко М.О			Формат діяльності підприємства	Літера	арк	Аркушів
Керівник	Віноградов М.А					48	33
Консульт					УС-212М 122		
Н. Контрол	Райчев І.Е						

Онлайн замовлення передається постачальнику напряму, тобто продавець отримує замовлення, і одразу передає данні про клієнта та замовлений товар, до постачальника, який бере на себе всі проблеми пов'язані із зберіганням товару, логістикою чи передачею товару до служби доставки.

Ризик при данному підході мінімальний, так як він не вимагає реальної покупки чи крупної закупки товару.

2.2. Етапи створення сайту

Розробку сайту можна порівняти з побудовою складного механізму, послідовно мають бути виконані зазначені етапи: розробка проекту, дизайн, програмування, розміщення на хостингу, наповнення контентом та просування.

Етапи розробки як правило проводяться послідовно, тому важливо розуміти, що не продумані заздалегідь зміни можуть значно вплинути на функціонал та ефективність роботи, так як без їх правильної інтеграції вони будуть впливати на роботу та швидкодію вже створених компонентів.

2.2.1. Підготовка проекту

Розробка проекту - це перший із кроків створення сайту. Для чого необхідний проект? У процесі створення сайту залучають експертів у абсолютно різних галузях, залежно від проекту, кожен учасник процесу точно визначить обсяг та поставлені перед ним задачі. Процес написання проекту передбачає активну участь розробника, проектувальника та клієнта. Лише після розробки проекту можна приступити до роботи над створенням сайту.



Рис. 2.1. Стек Web-технологій

2.2.2 Розробка дизайну

Розробка дизайну, відрізняється від всіх інших напрямків візуального оформлення, яка поділяється на три підкатегорії:

- 1) *Дизайн-концепція сайту*. Перш за все ідея, та основна графічна концепція дизайну, головна та декілька другорядних сторінок, для прикладу. Схематичний макет шапки, підвалу, основних елементів таких як бокові меню, форма пошуку сайту.
- 2) *Технічний дизайн*. Необхідно визначити ці елементи вказати їх, розмістити в логічній та скоріше елементарній послідовності для того

щоб користувач який має досвід в користуванні іншими веб сторінками відчував оригінальність нашої, але так же щоб нічого не викликало у нього логічного дисонансу, не змушувало обдумувати, що і де знаходиться. Можна навести приклад супермаркету, ми завжди знаємо що молочний відділ в дальньому кутку від входу в приміщення, для того щоб завжди зберігати оптимальну температуру, так же ми знаємо що пошук завжди зверху в шапці сайту.

3) *Верстка*. Створення HTML шаблонів сторінок, накладання стилів за допомогою CSS на основі прийнятого дизайну. Перевірка на правильність написання коду, це потрібно для того, щоб не викликати помилок у вбудованому інтерпретаторі браузеру та для оптимальної пошукової оптимізації сайту (SEO).

2.2.3. Програмування серверної частини

Один з найбільших, трудомістких та часозатратних процесів при створенні сайту. Цей етап розробки вимагає особливого професіоналізму розробника, від якого буде залежати подальша робота сайту. Сюди входить: налаштування системи управління, програмування та кастомізація компонентів якщо необхідно, забезпечення безпеки та тестування.

Налаштування системи управління. Зваженим рішенням є використання CMS, як готового продукту, який відповідає всім важливим параметрам, особливо це актуально для сайтів із складною структурою та великим об'ємом даних. Першим кроком є стандартне налаштування системи, підготовка сервера та встановлення продукту.

Програмування та кастомізація. На цьому кроці відбувається доопрацювання функціоналу що не міститься в стандартній версії CMS.

Забезпечення безпеки та тестування. Після закінчення робіт необхідно протестувати сайт на наявність помилок, здійснити пошук слабких місць

проекту, наприклад: HTML-форми за допомогою яких можна зробити SQL-ін'єкцію в БД.

2.3. Текстовий редактор

Для якісної та швидкою розробки ПЗ необхідно встановити ІДЕ.

Інтегроване середовище розробки – це комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

В якості інтегрованого середовища розробки будемо використовувати PHPStorm, який має такі можливості:

- 1) Підтримка ключових фреймворків. PhpStorm ідеально підходить для роботи з Symfony, Drupal, WordPress, Zend Framework, Laravel, Magento, Joomla!, CakePHP, Yii та іншими структурами.
- 2) Підтримка фронтенд-технологій. PhpStorm глибоко аналізує структуру коду і дійсно розуміє ваш код, підтримуючи всі можливості мови PHP як в нових, так і в legacy-проектах. Редактор підтримує автодоповнення коду і рефакторинг, запобігає помилки на льоту.
- 3) У PhpStorm ви можете працювати з найсучаснішими технологіями: HTML 5, CSS, Sass, Less, Stylus, CoffeeScript, TypeScript, Emmet і JavaScript. При цьому будуть доступні рефакторинг, налагодження та юніт-тестування. Завдяки функції Live Edit всі зміни можна тут же подивитися в браузері.
- 4) Вбудовані інструменти для розробників. Одноманітні завдання зручно виконувати прямо в PhpStorm. IDE інтегрована з системами контролю версій, підтримує віддалене розгортання, бази даних і SQL,

інструменти командного рядка, Docker, Composer, REST-клієнт і багато інших інструментів.

- 5) PhpStorm = WebStorm + PHP + БД / SQL. PhpStorm включає в себе всю функціональність WebStorm, а також повноцінну підтримку PHP, баз даних і SQL.

Завантажимо та встановимо актуальну версію PhpStorm, з сайту розробника <https://www.jetbrains.com/ru-ru/phpstorm/> (Рис. 2.1).

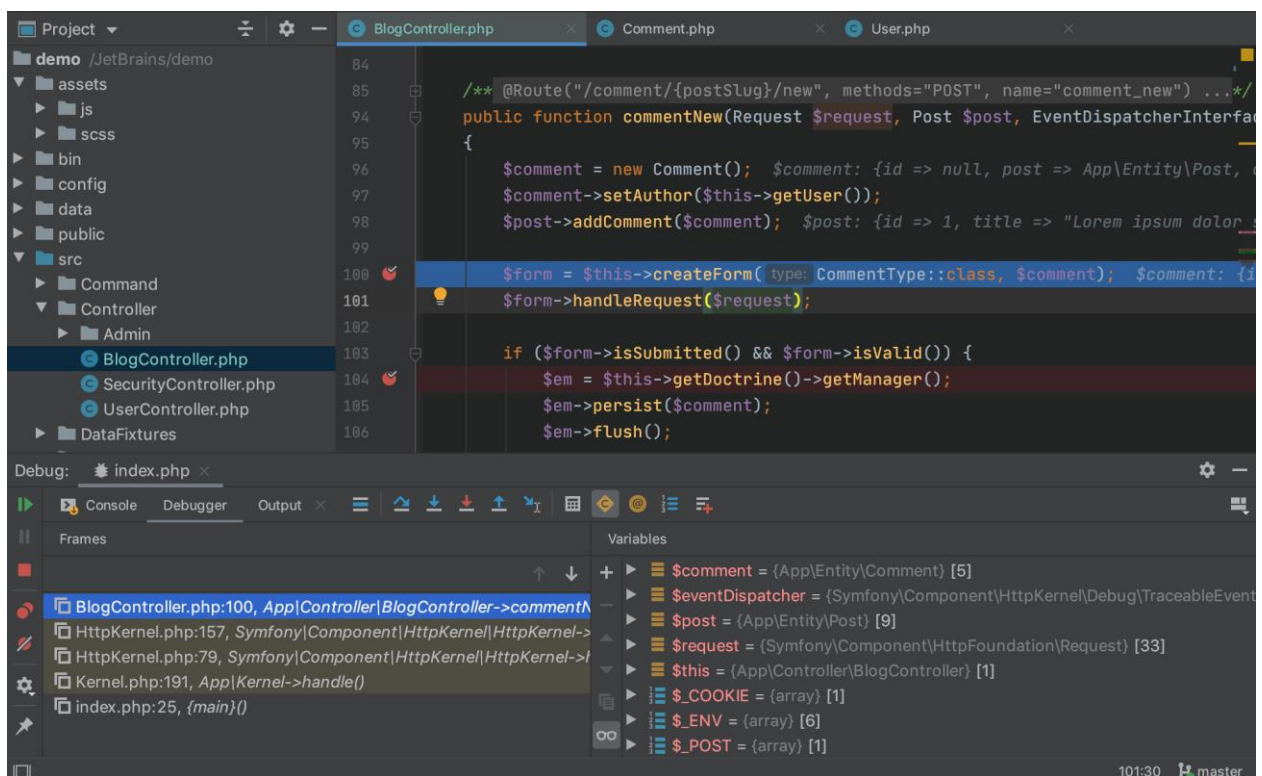


Рис. 2.1. Розробка в PhpStorm

2.4. Налаштування локального серверу

Для розробки проекту, використовують локальний сервер на робочій машині, тобто на ПК. Вже після того як проект можна вважати готовим його можна розмістити на хостингу, для цього використаємо Open Server.

Open Server- це портативна серверна платформа та програмне середовище, розроблені спеціально для веб-розробників на основі їх рекомендацій та побажань.

Програмний пакет має багатий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс, має потужні можливості адміністрування та налаштування компонентів. Платформа широко використовується для розробки, налагодження та тестування веб-проектів, а також для надання веб-сервісів у локальних мережах.

Хоча спочатку програмні продукти, що складаються з комплексу, не були розроблені спеціально для роботи один з одним, такий набір став дуже популярним серед користувачів Windows, насамперед тому, що вони отримали безкоштовний програмний пакет з надійністю на рівні серверів Linux.

Зручність та простота управління точно не залишать вас байдужими; За час свого існування Open Server зарекомендував себе як першокласний і надійний інструмент для кожного веб-майстра.

Даний програмний продукт надає нам такий програмний комплекс для розробки:

1) Основні компоненти:

- Apache **2.2.31 / 2.4.38 / 2.4.41 / 2.4.43;**
- Bind **9.16.1;**
- ConEmu **19.10.12;**
- FTP FileZilla **0.9.60;**
- Ghostscript **9.52;**
- HeidiSQL **11.0.0.5944;**
- Nginx **1.17.10;**
- NNCron Lite **1.17;**
- Opera **67.0.3575.137;**
- Sendmail **32;**

- Sublime **3.2.2.3211**;
- Wget **1.20.3**.

2) Системи управління базами даних:

- MariaDB **5.5.67 / 10.1.44 / 10.2.31 / 10.3.22 / 10.3.22 / 10.4.12**;
- Memcached **1.2.6 / 1.4.5 / 1.5.10**;
- MongoDB **2.4.14 / 2.6.12 / 3.0.15 / 3.2.22 / 3.4.24 / 3.6.17 / 4.0.17 / 4.2.5**;
- MySQL **5.1.73 / 5.5.62 / 5.6.47 / 5.7.29 / 8.0.19**;
- PostgreSQL **9.2.24 / 9.3.25 / 9.4.26 / 9.5.21 / 9.6.17 / 10.12 / 11.7 / 12.2**;
- Redis **2.8.2402 / 3.0.504 / 3.2.100 / 4.0.14.2 / 5.0.6-dev**.

3) PHP модулі:

- PHP **5.2.17** + розширення;
- PHP **5.3.29** + розширення;
- PHP **5.4.45** + розширення;
- PHP **5.5.38** + розширення;
- PHP **5.6.40** + розширення;
- PHP **7.0.33** + розширення;
- PHP **7.1.33** + розширення;
- PHP **7.2.29** + розширення;
- PHP **7.3.17** + розширення;
- PHP **7.4.05** + розширення.

4) Додаткове ПЗ:

- Adminer **4.7.6**;
- PHPMemcachedAdmin **1.3**;
- PHPMyAdmin **5.0.2**;

- PHPPgAdmin 7.12.1;
- PHPRedisAdmin 1.13.1.

За посиланням <https://ospanel.io/> з офіційного сайту розробника, завантажимо, актуальну версію продукту (Рис. 2.2).

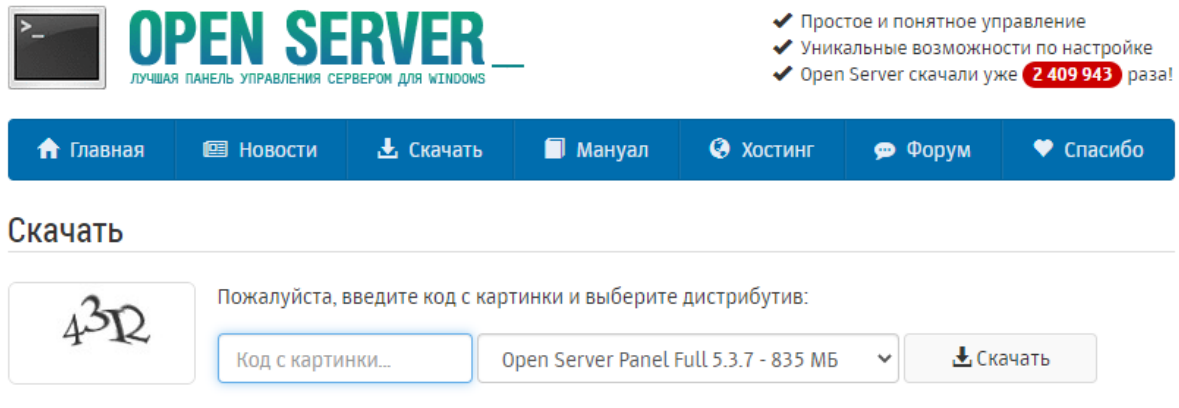


Рис. 2.2. Завантаження Open Server

Після встановлення ПЗ виглядає так (Рис. 2.3):

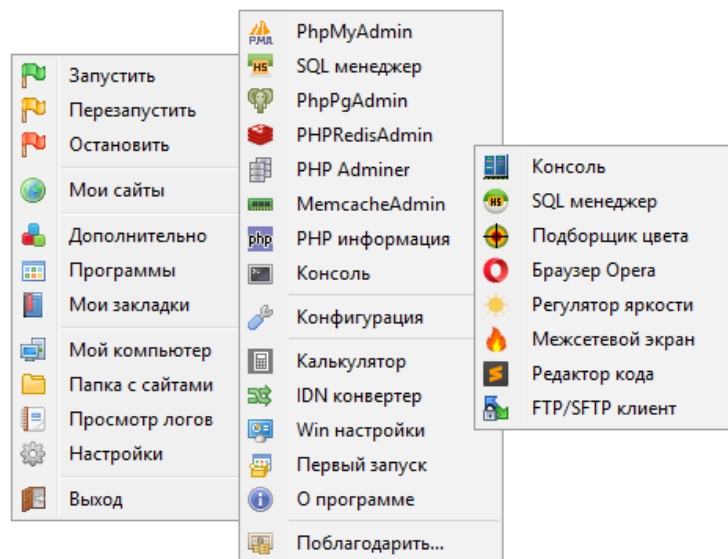


Рис. 2.3. Встановлений Open Server

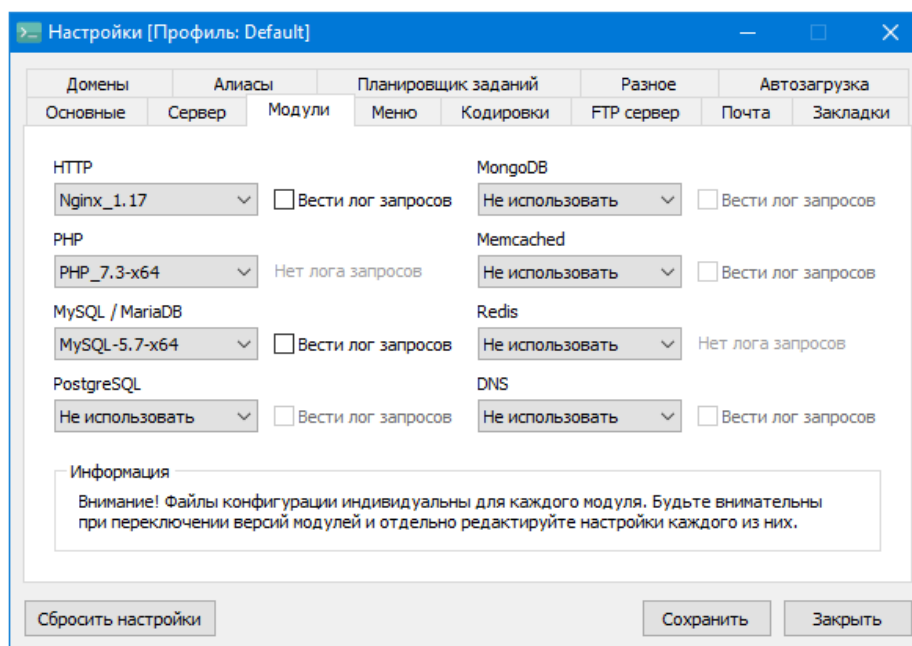


Рис 2.4. Налаштування модулів

Налаштуємо Apache 2.4 як локальний сервер, версію PHP 7.3 для операційних систем з кодування 64 біт та MySQL 5.7 також для операційних систем з кодування 64 біт (Рис. 2.4).

В директорії D:\OSpane\domains створимо директорію з назвою gresnad, та презавантажимо веб-сервер OpenServer. Таким чином отримуємо локальний домен за адресою <http://gresnad.local/> на ПК для розробки.

2.5. Встановлення Bitrix

Система управління Bitrix дуже потужна, і дозволяє створювати повнофункціональні сайти.

Для її встановлення необхідно завантажити актуальну версію, з сайту розробника <https://www.bitrix.ua/download/>. Після цього розпакувати архів на створеному домені, та виконати інсталяцію.

Після копіювання файлів з архіву в директорію бачимо таку файлову структуру (Рис. 2.5).

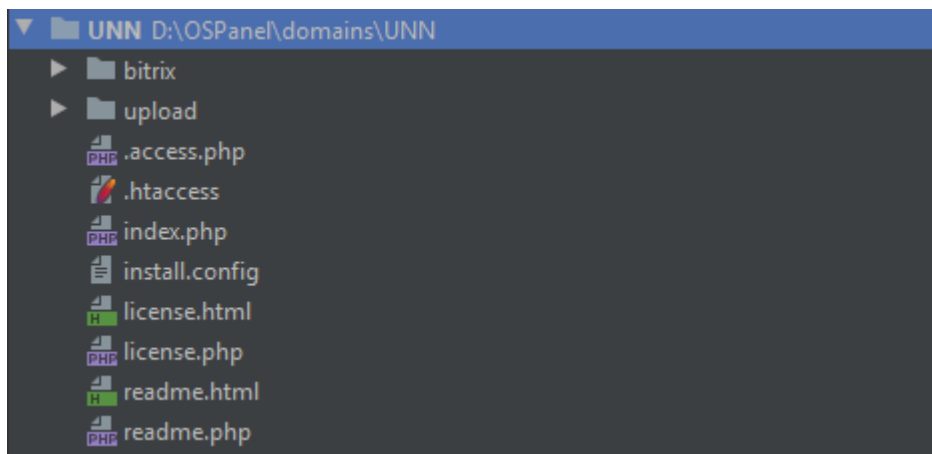


Рис 2.5. Файлова структура Вітріх до інсталяції

Перейшовши за посиланням домену <http://greensad.local/> . Бачимо стартову сторінку встановлення CMS. Покроково виконуємо усі вимоги (Рис. 2.6).

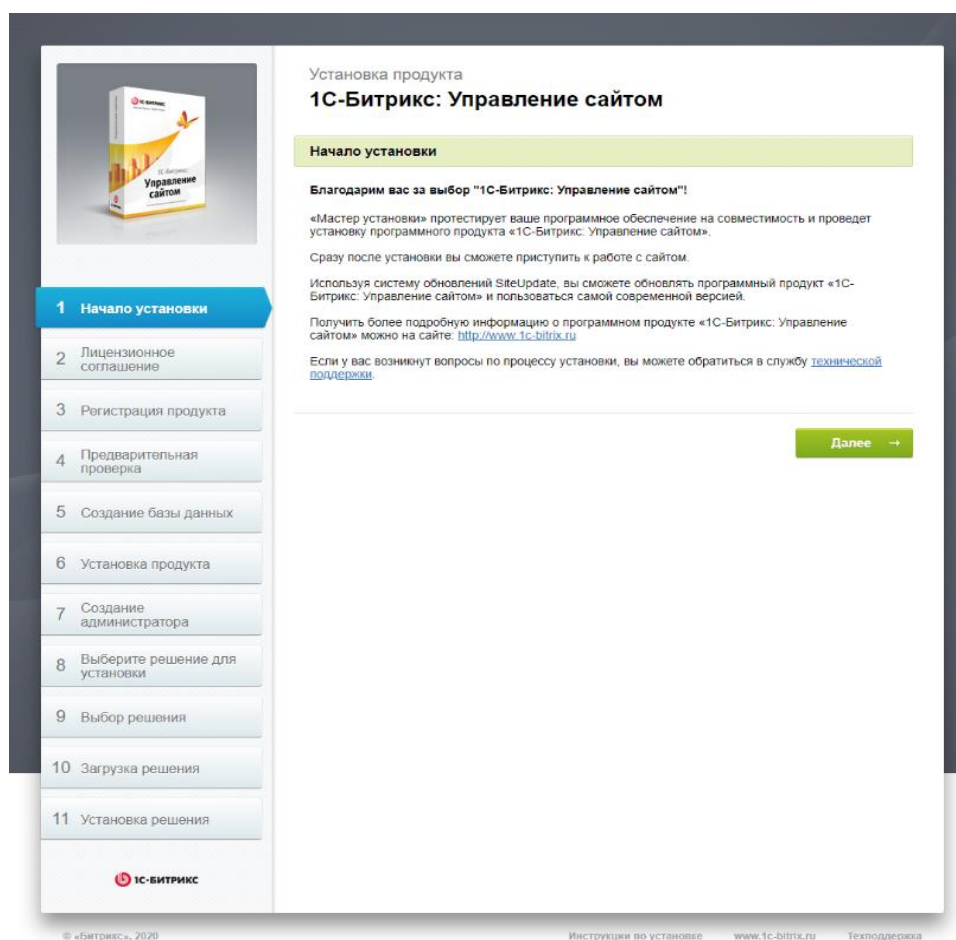


Рис. 2.6. Крок 1, стартова сторінка встановлення CMS

На 2 кроці читаємо та погоджуємося з ліцензійною угодою для користувачів (Рис. 2.7).

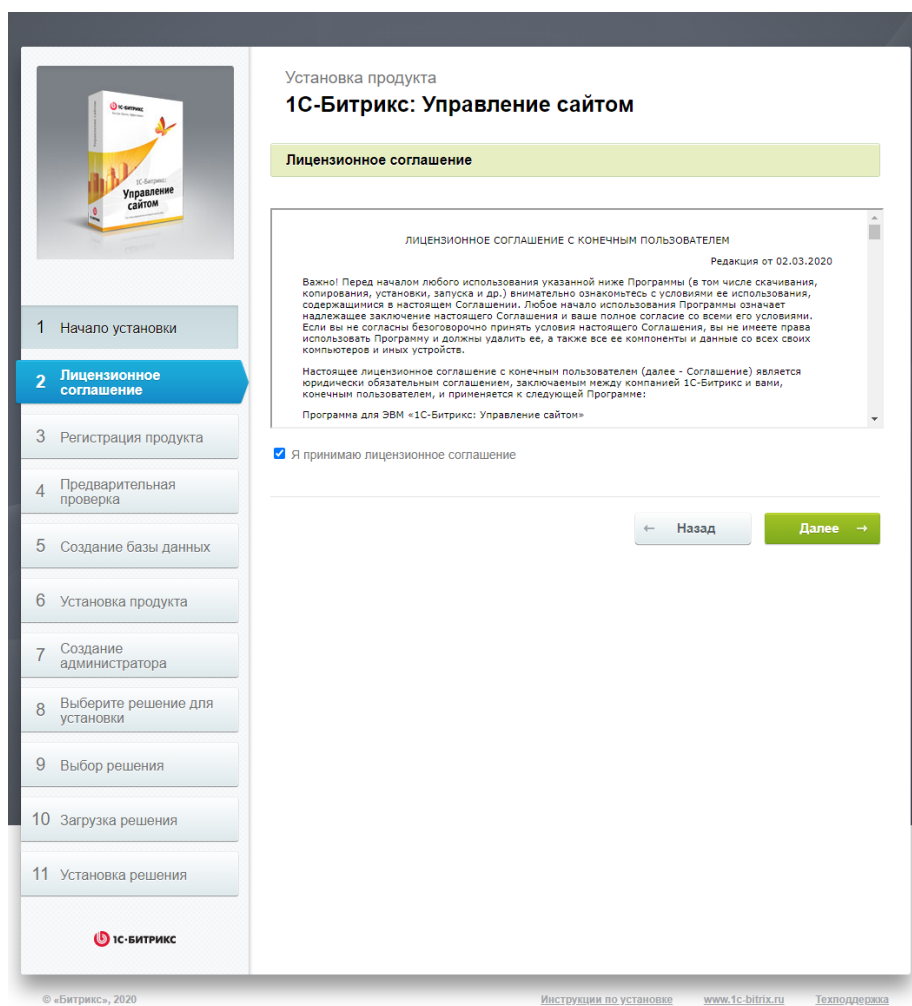



Рис. 2.7. Крок 2 ліцензійна угода

На 3 кроці реєструємо свою копію продукту за допомогою своєї електронної адреси, цей крок потрібен для отримання оновлень та готових рішень з Маркетплейсу Бітрікс, якщо дані функції не потрібні пропускаємо цей крок.

На кроці 4 ПЗ під назвою менеджер з установки продукту, виконує перевірку системи на сумісність. Система повинна відповідати всім параметрам, якщо хоч один з параметрів виділений червоним, то його

потрібно виправити та налаштувати, інакше CMS може бути не дієздатною після встановлення. В нашому випадку все добре (Рис. 2.8).



- 1 Начало установки
- 2 Лицензионное соглашение
- 3 Регистрация продукта
- 4 Предварительная проверка
- 5 Создание базы данных
- 6 Установка продукта
- 7 Создание администратора
- 8 Выберите решение для установки
- 9 Выбор решения
- 10 Загрузка решения
- 11 Установка решения

Установка продукта

1С-Битрикс: Управление сайтом

Предварительная проверка

Обязательные параметры системы

Ваша система должна соответствовать обязательным параметрам. Если какой-либо из этих параметров выделен красным цветом, то вам необходимо исправить его. В противном случае работоспособность сайта не гарантируется.

Параметр	Требуется	Текущее значение
Версия веб-сервера не выяснен		
Версия PHP	7.1.0 и выше	7.3.9
Установки PHP:		
safe mode	Выключен	Выключен
date.timezone	Установлен	Europe/Moscow
Требуемые модули PHP:		
функции MySQL	Установлен	Установлен
поддержка регулярных выражений (Perl-Compatible)	Установлен	Установлен
поддержка hash функций	Установлен	Установлен
поддержка функций JSON	Установлен	Установлен
OpenSSL	Установлен	Установлен
Для установки продукта в кодировке UTF-8		
Multibyte String	Установлен	Установлен
mbstring.func_overload	2	2
default.charset	UTF-8	UTF-8

Проверка доступа к диску

Скрипты сайта должны иметь доступ на запись к файлам сайта. Это необходимо для работы модуля управления структурой сайта, загрузки файлов, а так же для работы системы обновлений, которая обновляет ядро сайта до последней версии.

Параметр	Значение
Публичная часть D:/OSPanel/domains/UNN	Доступен для чтения и записи
Ядро сайта D:/OSPanel/domains/UNN/bitrix	Доступен для чтения и записи
Файл D:/OSPanel/domains/UNN/index.php	Доступен для чтения и записи
Каталог D:/OSPanel/domains/UNN/bitrix/modules	Доступен для чтения и записи

Рекомендуемые установки

Параметры, которые рекомендуется установить в соответствующие значения, чтобы обеспечить полную совместимость с требованиями продукта. Сайт будет работать даже в том случае, если некоторые установки не соответствуют рекомендациям. Но в этом случае часть функционала сайта может оказаться недоступной.

Параметр	Рекомендуется	Текущее значение
Обработка .htaccess	Включена	Включена
Доступное место на диске сайта	не менее 500 Mb	18705 Mb
Рекомендуемые установки PHP:		
доступная память	не менее 64 Mb, рекомендуется не менее 256 Mb	1536M
разрешить загрузку файлов (file_uploads)	Включено	Включено
показывать ошибки (display_errors)	Включено	Включено
квотирование Sybase (magic_quotes_sybase)	Выключено	Выключено
Рекомендуемые модули PHP:		
Zlib Compression	Установлен	Установлен
Библиотека GD (функции для работы с графикой)	Установлен	Установлен
Free Type Library	Установлен	Установлен

Зеленым цветом выделены значения, которые удовлетворяют требованиям системы.
Красным цветом выделены значения, которые **не** удовлетворяют требованиям системы.

← Назад
Далее →

Рис. 2.8. Перевірка параметрів системи

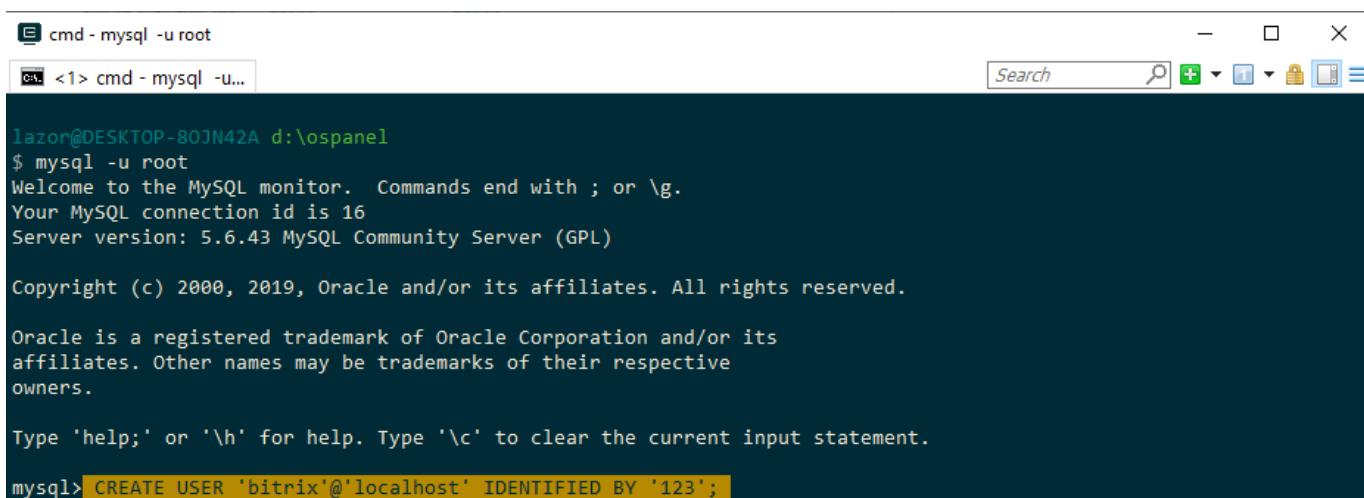
Кроком 5 є налаштування бази даних, для початку треба створити користувача для CMS. За допомогою терміналу авторизуємося в базі даних MySQL під акаунтом супер користувача root, пишемо команду:

```
mysql -u root
```

Після цього за допомогою команди:

```
CREATE USER 'bitrix'@'localhost' IDENTIFIED BY '123';
```

Створюємо користувача bitrix за паролем 123 (Рис. 2.9).

A screenshot of a Windows command prompt window titled "cmd - mysql -u root". The window shows the execution of the command "mysql -u root". The output displays the MySQL command-line interface, including the MySQL version (5.6.43) and the user's connection ID (16). The command "CREATE USER 'bitrix'@'localhost' IDENTIFIED BY '123';" is entered and highlighted in yellow. The terminal background is dark green with white text.

```
lazor@DESKTOP-80JN42A d:\ospanel
$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.6.43 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'bitrix'@'localhost' IDENTIFIED BY '123';
```

Рис. 2.9. Створення користувача MySQL

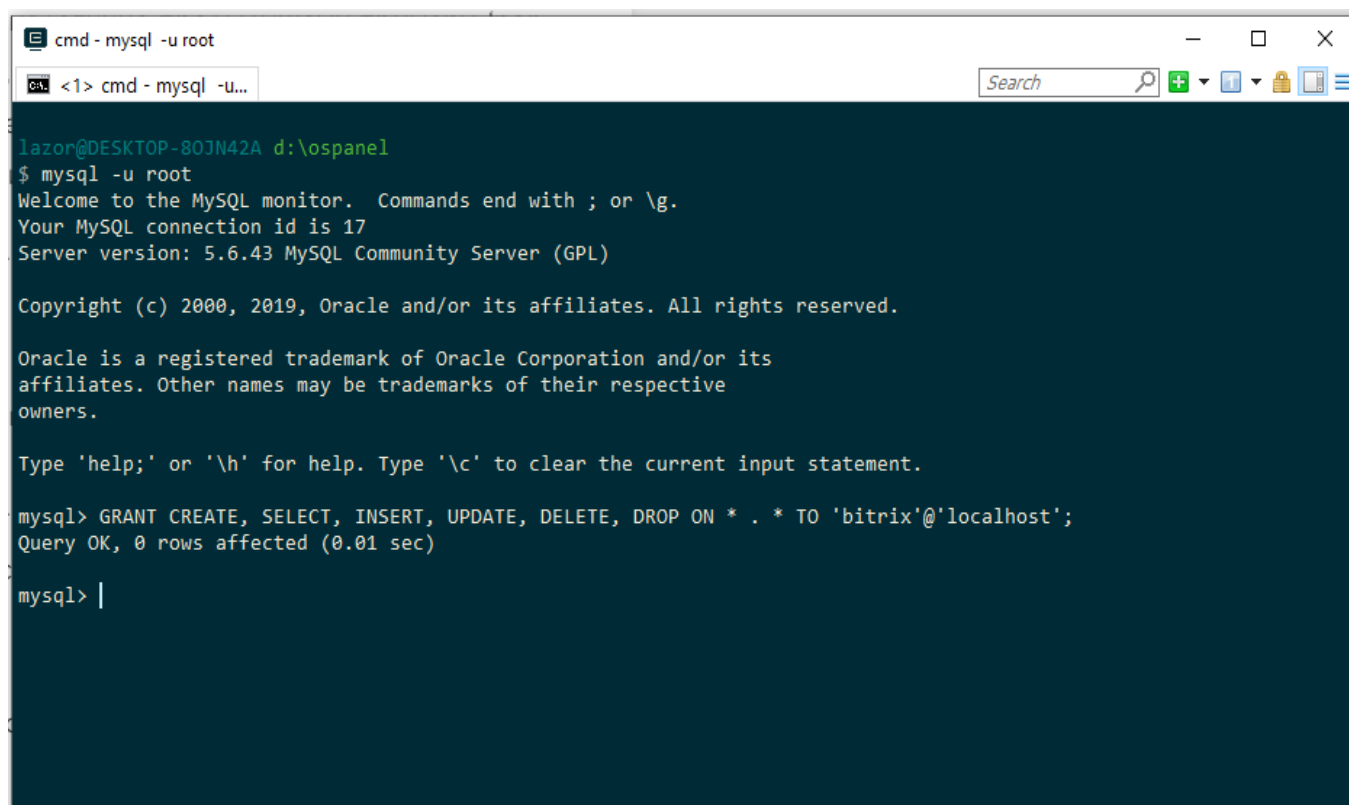
Надамо права на створення та зміни таблиць, додавання, змінну та видалення даних в БД.

Список прав:

- 1) CREATE - Дозволяє користувачам створювати бази даних / таблиці
- 2) SELECT - Дозволяє користувачам робити вибірку даних
- 3) INSERT - Дозволяє користувачам додавати нові записи в таблиці
- 4) UPDATE - Дозволяє користувачам змінювати існуючі записи в таблицях
- 5) DELETE - Дозволяє користувачам видаляти записи з таблиць
- 6) DROP - Дозволяє користувачам видаляти записи в базі даних / таблицях
- 7)

Виконаємо команду (Рис. 2.10):

```
GRANT [тип прав] ON [ім'я бази даних].[ім'я таблиці] TO  
bitrix'@'localhost';
```

A screenshot of a Windows command prompt window titled "cmd - mysql -u root". The window shows a user named "lazon" at a desktop with IP "80JN42A" in the directory "d:\ospanel". The user has executed the command "mysql -u root", which has opened the MySQL command-line interface. The MySQL prompt shows the user is logged in as "root" and the server version is "5.6.43 MySQL Community Server (GPL)". The user has entered the command "GRANT CREATE, SELECT, INSERT, UPDATE, DELETE, DROP ON * . * TO 'bitrix'@'localhost';". The MySQL prompt shows the command was executed successfully, returning "Query OK, 0 rows affected (0.01 sec)". The prompt is now ready for the next command.

```
cmd - mysql -u root  
lazon@DESKTOP-80JN42A d:\ospanel  
$ mysql -u root  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 17  
Server version: 5.6.43 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> GRANT CREATE, SELECT, INSERT, UPDATE, DELETE, DROP ON * . * TO 'bitrix'@'localhost';  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> |
```

Рис. 2.10. Надання прав користувачеві БД

Повертаємось до налаштування CMS, заповнюємо наші дані в відповідні поля (Рис. 2.11).

Установка продукта
1С-Битрикс: Управление сайтом

Создание базы данных

Параметры базы данных

* Сервер: localhost
Сервер, на котором работает система управления базами данных

Пользователь базы данных: Существующий
 Новый

* Имя пользователя: bitrix
Логин для соединения с базой данных

Пароль: ...
Пароль для соединения с базой данных

База данных: Существующая
 Новая

*Имя базы данных: UKRAINE_NEWS_NETWORK
База данных, в которую будет установлен продукт

Тип таблиц базы данных: стандартный

Дополнительные параметры

Права на доступ к файлам сайта: 0644
Права должны быть достаточными для доступа веб-сервера на запись, уточните необходимое значение у хостера

Права на доступ к папкам сайта: 0755
Права должны быть достаточными для доступа веб-сервера на запись, уточните необходимое значение у хостера

← Назад Далее →

Рис 2.11. Налаштування CMS, до створеної БД

На кроці 6 чекаємо встановлення системи, йде розпакування архівів які залишились та заповнення БД системними таблицями та даним (Рис. 2.12).

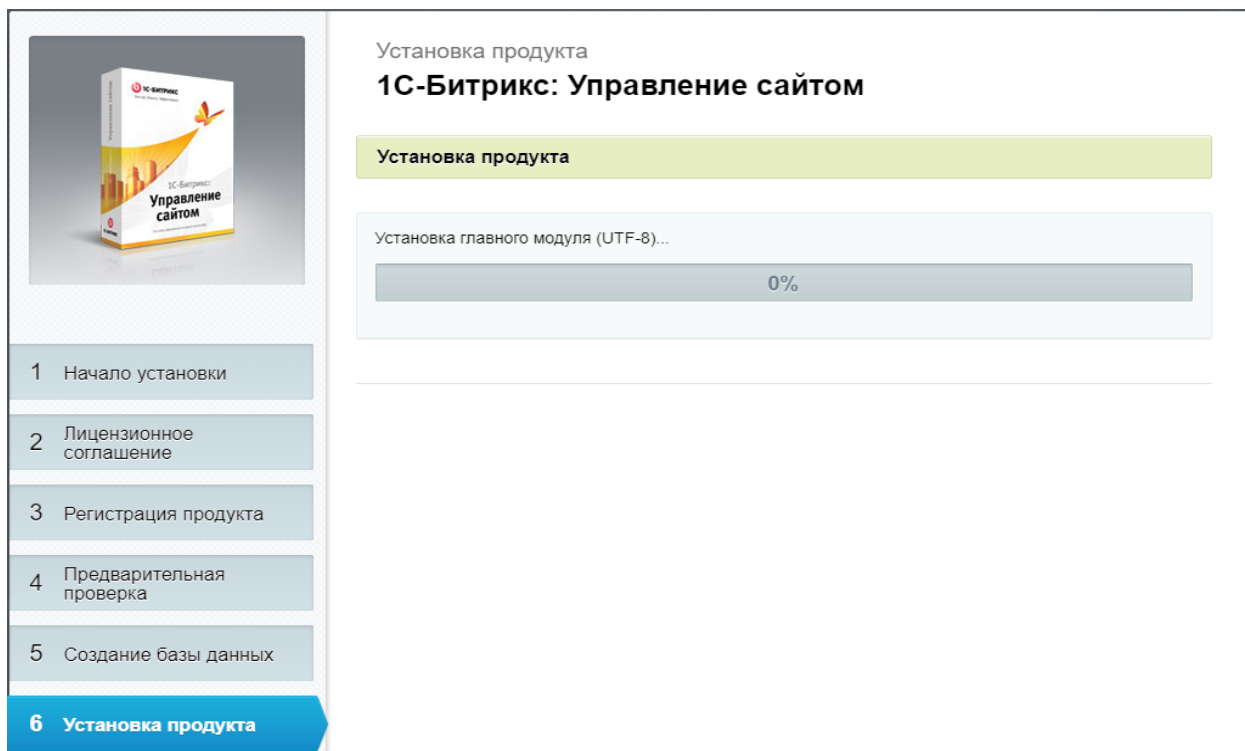


Рис. 2.12. Встановлення системи

Крок 7 творюємо адміністратора для адмін-панелі, запишемо логін та пароль у відповідні поля (Рис. 2.13).

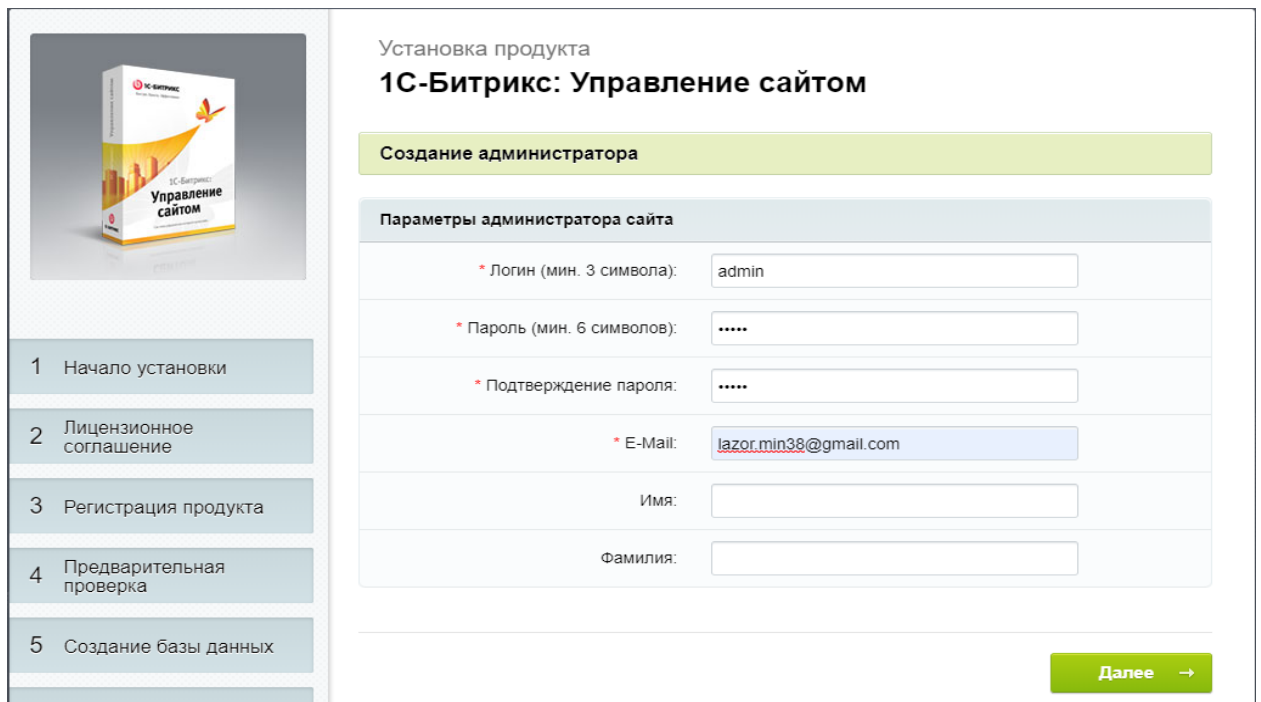


Рис. 2.13. Створення адміністратора

Після цього бачимо, сторінку з рядком завантаження, чекаємо, і нас перенаправляє в адмін-панель (Рис. 2.14).

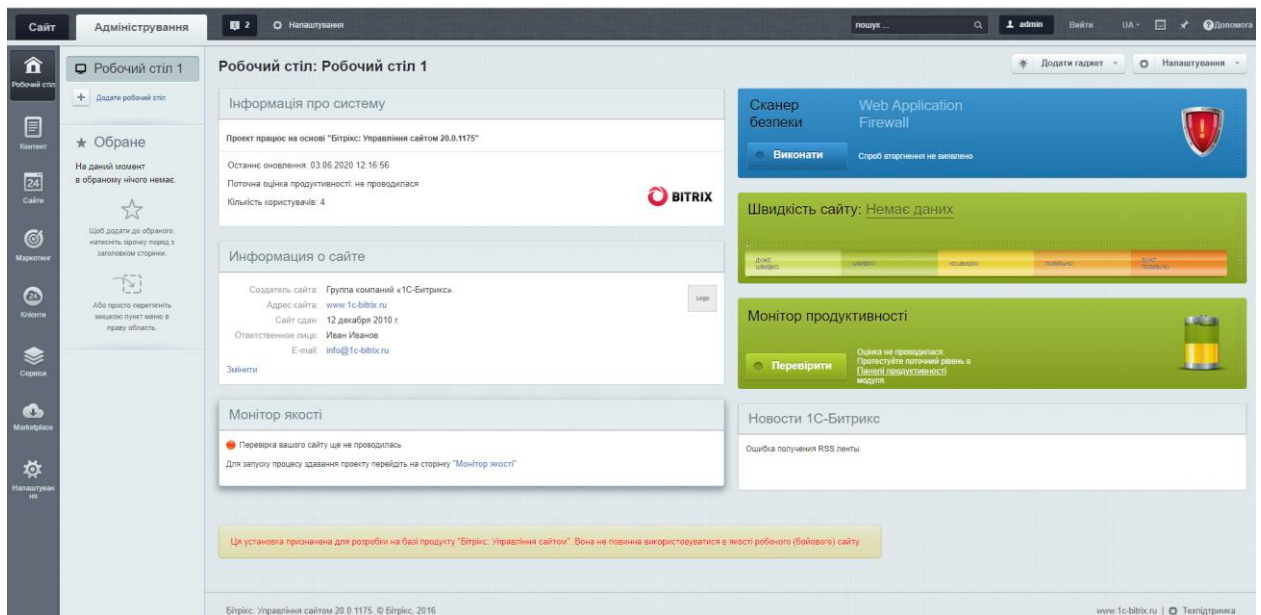


Рис. 2.14. Адмін-панель

Якщо ми подивимось на файлову структуру після інсталяції, то помітимо що вона дуже сильно розширилась (Рис. 2.15).

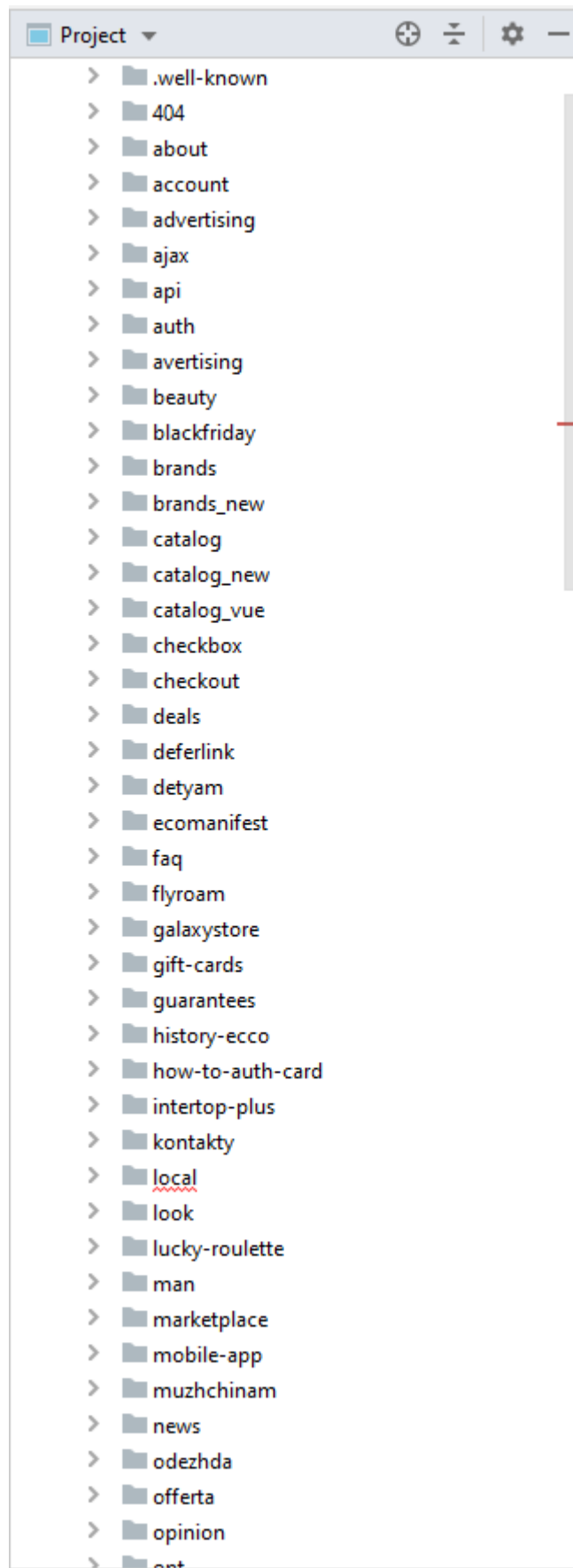


Рис. 2.15. Файлова структура після інсталяції

2.6. Створення карти сайту

Карта сайту не список сторінок сайту, які представлені в графічному або письмовому вигляді. Карти сайту використовуються для того щоб допомогти відвідувачам чи пошуковим роботам знайти потрібну інформацію чи сторінку.

Карти сайтів бувають трьох видів:

- 1) Карти сайтів які використовуються при розробці.
- 2) Карти, які показують користувачам структуру сайту.
- 3) Карти у форматі, наприклад, XML для пошукових роботів

Карти сайтів у текстовому форматі, використовуються для кращого, для звукових систем, для індексації сайту. Створення карти сайту - важливий процес, в якому необхідно чітко її читати, оскільки сайт потребує індексації і бути красивішим за індексацію. Також питання про те, який тип картини може знайти сайт, легко дізнатися, як створити карту для сайту. Як відкрити html-карту для сайту, обговорювати окремо сенсеу немає. Легко зрозуміти, як створити карту з файлу xml. Існує кілька основних способів передачі зображення на сайт, але вони будуть пояснені, якщо буде знайдена карта сайту і файл карти сайту буде відтворено звуковими системами. Як уже було написано вище, файл картинки на сайт поширюється в корені сайту. Поштові системи можуть самостійно відображати файл на сайті. Існує кілька способів надсилання прямих повідомлень у файл(и) на сайт(и) зображення для кращого виявлення за допомогою звукових систем. Найпростіший спосіб надати швидкий шлях до файлу карти сайту – це простий спосіб використовувати файл карти сайту, не переглядаючи файли карти сайту в інструментах для веб-майстрів від Яндекс та Google. Там ви також можете переглянути карту сайту, проаналізувати карту сайту на коректність і побачити, як посилання з карти сайту відомі звуковій системі та як вони знаходяться в індексі. Для того, щоб карта сайту була правильно проіндексована та адекватно надіслана на сайт,

змінити її збоку більше 100 разів не можна. Недостатньо мати багато місця збоку від усіх продуктів, які не помістилися в меню.

Карта сайту підрозділяється на 2 основних типи форматів: HTML-файл карти сайту та файл XML-файлу карти сайту. HTML карта для сайту - ціна за сайт, за який здійснюється перерахування посилання. Зателефонуйте нам за ціною посилання на сайт. HTML-карта для сайту більше призначена для людей, менше для роботів і для додаткової допомоги, які ширше використовуються в основних розділах сайту. Щодо зображення сайту, у вигляді html-боку, багато думається про одну сторону. Як тільки, з іншого боку, він занадто великий, щоб запитувати, не все це можна проіндексувати, тому що бік картинки на сайті можна ввімкнути на жарт для надсвіту, але не знаю, чи всередині .

Для того, щоб карта сайту була правильно проіндексована та адекватно надіслана на сайт, змінити її збоку більше 100 разів не можна. Недостатньо мати багато місця збоку від усіх продуктів, які не помістилися в меню.

Називаємо карту сайту у форматі HTML, але вона має деревоподібну структуру, позначається розгорнутими розділами і підрозділами. Перш за все, величезна кількість HTML-картинок для сайту, часто прикрашених графічними елементами, CSS-стилями та оновленими Java-скриптом. Однак HTML карти сайту не має великого значення для систем постпродакшн.

HTML карта сайту - не карта сайту. Як бути, як сотні, тисячі, десятки тисяч сторінок на сайті? Загалом, необхідно розповсюдити інформацію на всі сторони карти сайту у форматі xml.

Створимо карту сайту для пошукових роботів за допомогою стандартних інструментів Bitrix. В адмін-панелі перейдемо на вкладку маркетинг і перейдемо на сторінку «Налаштування Sitemap.xml» (Рис. 2.16).

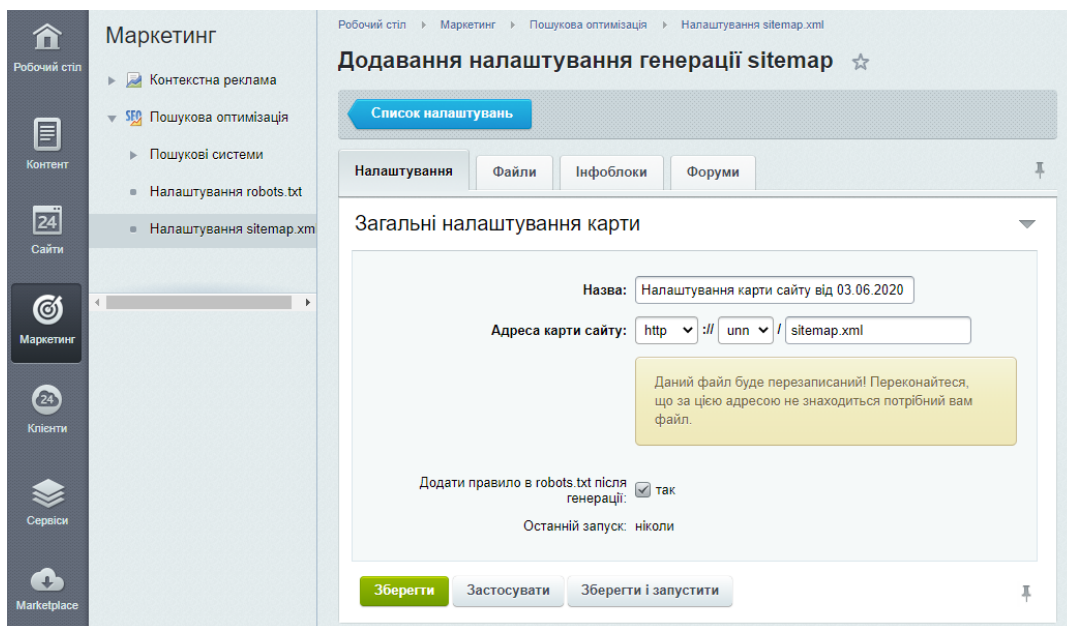


Рис. 2.16. Створення карти сайту в форматі XML

Після цього бачимо подібну структуру, в файлі sitemap.xml (Рис. 2.17)

```
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://mysite.ua/</loc>
    <lastmod>2014-09-18T18:54:13+04:00</lastmod>
    <changefreq>always</changefreq>
    <priority>1.0</priority>
  </url>
  <url>
    <loc>http://mysite.ua/category/</loc>
    <lastmod>2014-09-18T18:57:09+04:00</lastmod>
    <changefreq>hourly</changefreq>
    <priority>0.8</priority>
  </url>
  <url>
    <loc>http://mysite.ua/page/</loc>
    <lastmod>2014-09-18T18:59:37+04:00</lastmod>
    <changefreq>daily</changefreq>
    <priority>0.6</priority>
  </url>
</urlset>
```

Рис. 2.17. Карту сайту в форматі XML

Де використовуються такі обов'язкові теги:

<urlset> - батьківський тег, він містить усі URL-адреси;

<url> - тег, що містить інформацію про певну url-адресу;

<loc> - цей тег містить саму URL-адресу.

Крім того, наступні додаткові теги:

<lastmod> - цей тег містить дату останньої зміни сторінки;

<changefreq> - тег використовується для визначення частоти змін сторінки: завжди, щогодини, щодня, щотижня, щомісяця, щороку, ніколи;

<пріоритет> - вказує пріоритет певної сторінки, відносно інших сторінок сайту, від 0,1 - низький пріоритет, до 1 - високий пріоритет.

Файл карти сайту у форматі xml повинен містити вказівку на простір імен XML:

```
xmlns = "http://www.sitemaps.org/schemas/sitemap/0.9"
```

Якщо файл карти сайту містить більше 50 тисяч посилань або розмір файлу sitemap.xml перевищує 10 МБ, рекомендується розділити карту сайту на кілька файлів. У цьому випадку в карті сайту необхідно вказати кілька посилань на різні файли карти.

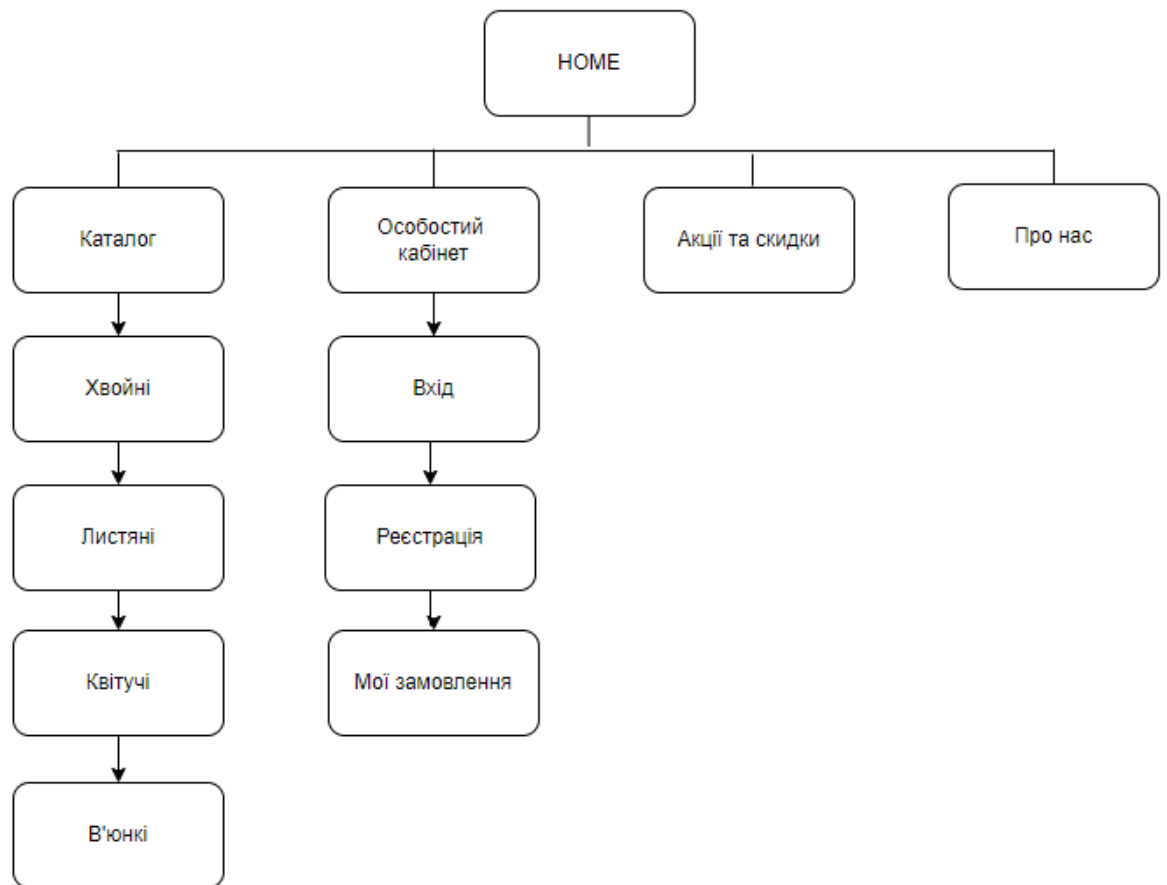


Рис. 2.18. Графічна карта сайту

Будемо використовувати цю карту при розробці, та заповненні інформації сайту (Рис. 2.18).

2.7. Створення дизайну сайту

Перед створенням дизайну сайту, необхідно створити макет сайту, це своєрідний план сайту. Макет не повинен бути дуже докладним, але важливою частиною має включати всі елементи які ми хочемо бачити на сторінці. Пізніше можна додати всі елементи які не входять в макет, пізніше макет послугує візуальним орієнтиром, коли дизайн буде робитись по справжньому (Рис. 2.19).

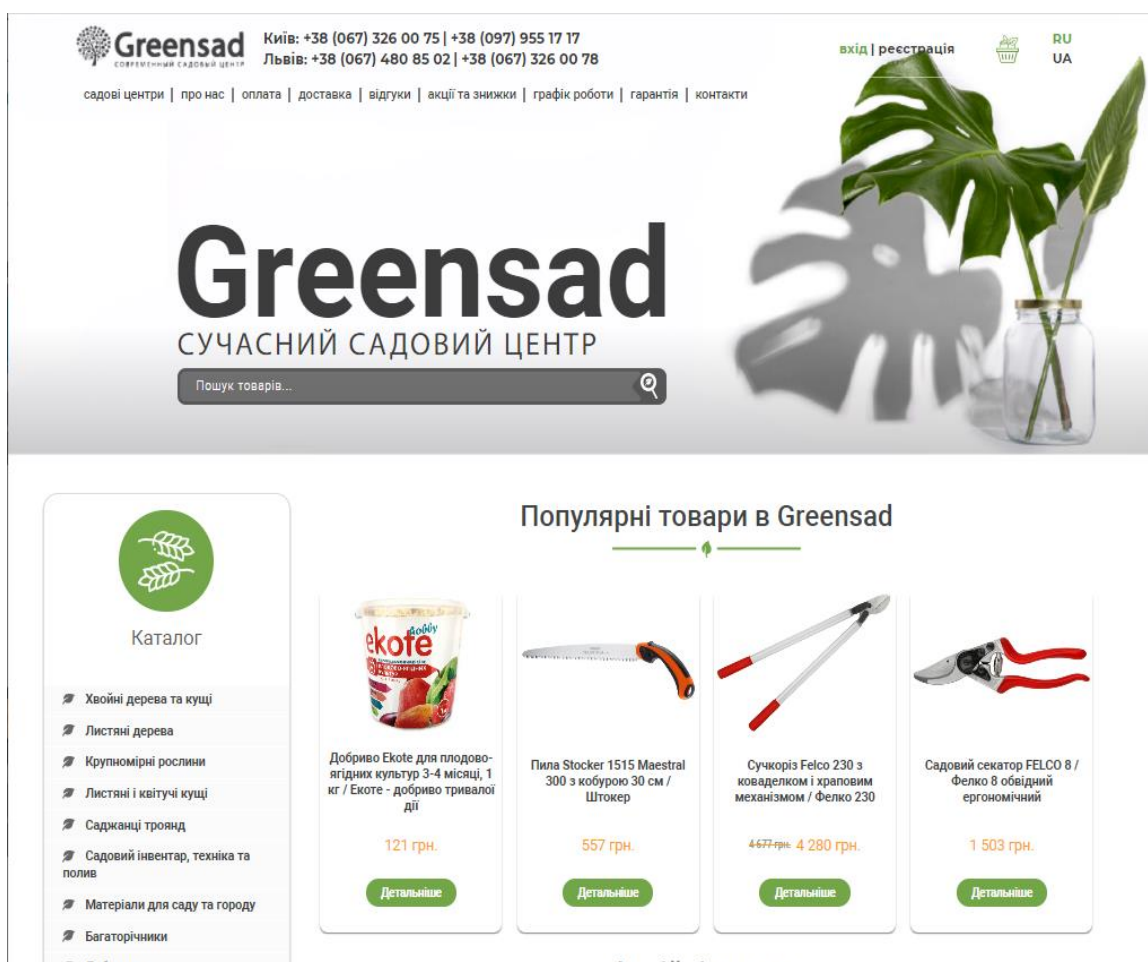


Рис. 2.19. Макет сайту

2.8. Розробка шаблону сайту

За допомогою описаних вище HTML та CSS. Створюємо шаблон нашого сайту.

Почнемо з шапки сайту, керуючись нашим макетом. Шапка сайту знаходиться у верхній частині і займає всю ширину сторінки, вона є на всіх сторінках, тобто є майже основним елементом дизайну, в шапці сайту знаходиться логотип, посилання на реєстрацію та авторизацію, меню з розділами порталу, та форма пошуку. Як логотип використаємо звичайний текст. Для меню з розділами використаємо список елемент HTML ``. В кінцевому етапі це буде мати такий вигляд (Рис. 2.20).

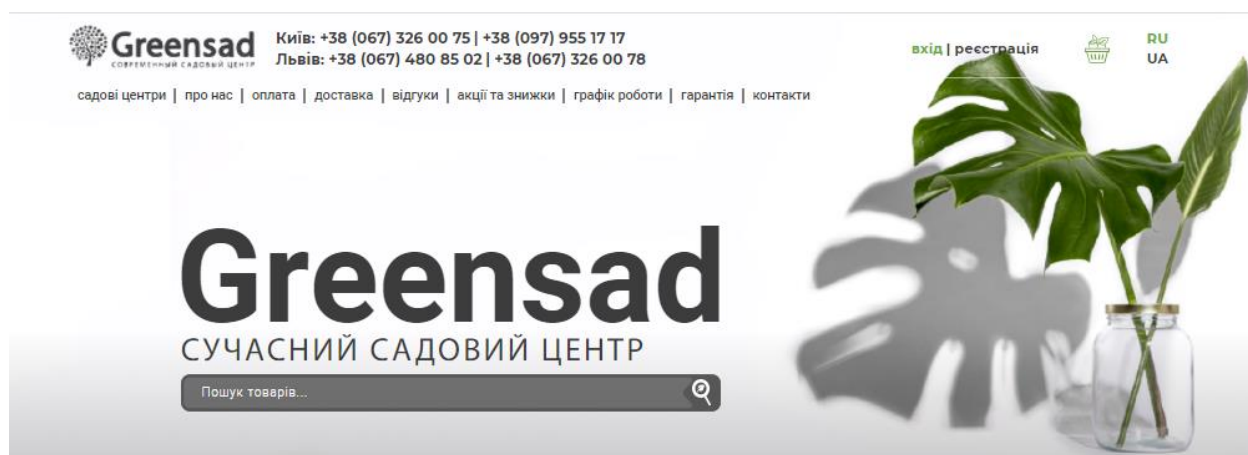


Рис. 2.20. Шапка сайту

Для форми пошуку можна елемент HTML `<input>` та компонент `bitrix.search.form` на серверній частині, даний компонент буде виконувати пошук по новинам, і відправляти для клієнта варіанти пошуку, які ми будемо обробляти за допомогою JavaScript і показувати користувачу (Рис. 2.21).



Рис. 2.21. Форма пошуку



Рис. 2.22. Детальна сторінка товару

2.9. Розробка автоматичного завантаження каталогу з сервера постачальника

Постачальник формує XML файл і розміщує його у себе на сервері. Наша задача в автоматичному режимі звернутися до сервера за протоколом HTTPS, отримати файл обробити його і зберегти товари до БД.

Для цього була написана функція “importCatalogXml” так вона виглядає:

```
function importCatalogXml () {
    \Bitrix\Main\Loader::includeModule('iblock');
    \Bitrix\Main\Loader::includeModule('catalog');

    $arElements = [];

    $elementsQuery = CIBlockElement::GetList(
        [],
        [
            'IBLOCK_ID' => '33',
            'ACTIVE' => 'Y',
        ]
    );
}
```

```

    ],
    false,
    false,
    [
        'ID',
        'CODE',
        'PROPERTY_OLD_CLIMALAND_ID'
    ]
);

while($element = $elementsQuery->fetch()){
    if(!empty($element['PROPERTY_OLD_CLIMALAND_ID_VALUE']))
        $arElements[$element['PROPERTY_OLD_CLIMALAND_ID_VALUE']] = $element;
    else
        $arElements[$element['CODE']] = $element;
}
$xmlFile =
simplexml_load_file('https://klimatlend.ua/bitrix/catalog_export/hotline.php');
$offersArray = (array) $xmlFile->shop->offers;
$offersArray = $offersArray['offer'];

foreach ($offersArray as $offer){
    $arOffer = (array) $offer;
    $urlSting = $arOffer['url'];
    $id = $arOffer['@attributes']['id'];
    $code = next(explode('klimatlend.ua/catalog/' , $urlSting ));
    $code = current(explode('.html' , $code ));
    );
    $CCatalogProductEntity->Add(
        [
            'ID' => $newId,
            'TYPE' => \Bitrix\Catalog\ProductTable::TYPE_PRODUCT,
            'QUANTITY' => 1000,
        ]
    );
    $resultPrice = CPrice::GetList(
        array(),
        array(
            'PRODUCT_ID' => $newId,

```

```
'CATALOG_GROUP_ID' => 1
    )
);
if ($arr = $resultPrice->Fetch()) {
    CPrice::Update($arr['ID'], $arFields);
} else {
    CPrice::Add($arFields);
}
unset($arElements[$id]);
}
}
return 'importCatalogXml()';
}
```

Дана функція встановлена на Cron, і запускається 4 рази щодобу. Cron це класична програма в системі Unix, використовується для періодичного виконання зачач в заданий час. Релярні команди описуються інструкціями в файлі crontab.

Команда виглядає так:

```
1 */* * * * php72 -f /var/www/grensad/local/checkCatalog.php
```

Шаблон завдання cron виглядає так:

Хвилини (0-59) Години (0-24) День (1-31) Місяць (1-12) День тижня (0-6) Команда

Зірочки позначають певні проміжки часу. Щоб відобразити вміст crontab поточного користувача, скористайтеся командою:

```
$ crontab -l
```

Для редагування завдань користувача є команда:

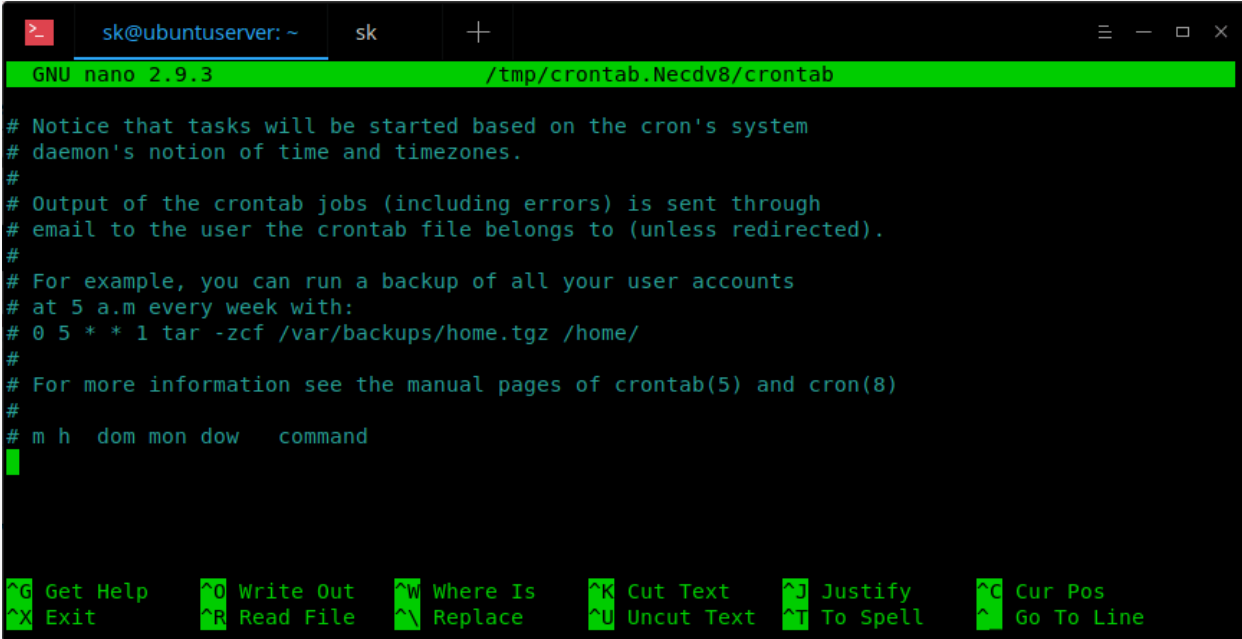
```
$ crontab -e
```

Якщо ви вперше запускаєте цю команду, вам буде запропоновано вибрати редактор для Cron:

```
no crontab for sk - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano <---- простейший
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]:
```



```
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command

```

Рис. 2.23. Вигляд Cron файлу в редакторі Vim

У цьому файлі вам просто потрібно перерахувати всі команди одну за одною.

1) Щоб виконувати команду щохвилини, завдання має бути таким:

```
* * * * * <Команда для виконання >
```

2) Аналогічне завдання, тільки команда буде викликатися кожні п'ять хвилин:

```
*/5 * * * * <Команда для виконання >
```

3) Виклик команди 4 рази на годину (кожні 15 хвилин):

```
*/15 * * * * * <Команда для виконання >
```

4) Щоб виконувати команду кожну годину за 30 хвилин, ми пишемо:

```
30 * * * * * <Команда для виконання >
```

5) Також є готові завдання:

- @reboot - виконання однієї команди під час завантаження;
- @yearly - раз на рік;
- @annually - також раз на рік;
- @monthly - раз на місяць;
- @weekly - раз на тиждень;
- @daily - один раз на день;
- @midnight - також раз на день;
- @hourly - раз на годину.

2.10. Обмін виконаними замовленнями

Для оперативної передачі замовлень розроблено, інтерфейс з списком замовлень. Що дозволяє ефективно и оперативно керувати замовленнями, та їхніми статусами. Менеджер має змогу відмінити замовлення в разі якщо воно не є коректним.

2.11. Опрацювання замовлення

Після того як ми на своєму сайті отримуємо замовлення від клієнта менеджеру в адмін-панелі сайту приходить сповіщення. В списку замовлень він бачить данні клієнта, його ім'я, адресу, номер телефону та данні про оплату. Менеджер сайту в ручному режимі передає данні про клієнта

постачальнику, який в свою чергу відправляє замовлення клієнту. Таким чином не маючи товару ми виступаємо посередником в продажі.

	Дата создания	ID	Покупатель	Статус	Оплачен	Отменен	Отгружен	Проблема с заказом	Сумма	Позиции
	28.05.2021 17:21:56	№92	Дмитрий	Принят, ожидается оплата	Нет	Нет	[184] Нет	Нет	710.00 грн.	[21439] Терморегулятор Electrolux Thermotronic ETB-16 (Basic) (1.0000 шт)
	28.05.2021 14:37:36	№91	Вадим Майстренко	Принят, ожидается оплата	Нет	Нет	[182] Нет	Нет	2 565.00 грн.	[23308] Радиатор Royal Thermo Vittoria 350 - 10 секц (1.0000 шт)

Рис. 2.23. Список замовлень

Замовлення передаються постачальнику кожні 2 хвилини, виконання цього завдання також встановлене на Cron. Інструкція для Cron виглядає так:

```
*/2 * * * * php72 -f /var/www/grensad/local/checkCatalog.php
```

2.12. Список користувачів

Для керування користувачами на сайті створено функціонал «Користувачі». Даний модуль дає можливість адміністратору сайту, бачити та керувати інформацією про користувачів. В модулі доступні такі можливості:

- 1) Авторизація під аккаунтом користувача;
- 2) Реєстрація нового користувача;
- 3) Видалення існуючого користувача;
- 4) Зміна Пошти, Логіну та Паролю користувача;
- 5) Прапор активності користувача;
- 6) Дата останньої авторизації користувача;
- 7) Повідомлення для користувача по пошті, про зміну його особистих даних;
- 8) Зміна аватара користувача.

Зовнішній вигляд, модуля «Користувачі» наведено на Рис 2.23.

<input type="checkbox"/>	⚙	ID ▾	ОСТАННЯ АВТОРИЗАЦІЯ	ЛОГІН	E-MAIL	АКТИВНІСТЬ	ДАТА ЗМІНЕННЯ	ІМ'Я
<input type="checkbox"/>	≡	46820	02.12.2021 21:32:52	gonchar.petro	gonchar.petro@gmail.com	Так		Петро
<input type="checkbox"/>	≡	46819	01.12.2021 16:50:41	tanger73	tanger73@ukr.net	Так	01.12.2021 16:59:44	татьяна
<input type="checkbox"/>	≡	46818	01.12.2021 12:29:58	vitass2warwar	vitass2warwar@gmail.com	Так		Віталій
<input type="checkbox"/>	≡	46817	30.11.2021 23:39:27	Stevetup	rezaasadi6780@gmail.com	Так		Stevetup
<input type="checkbox"/>	≡	46816	29.11.2021 22:46:57	oleksandr0408.zo	oleksandr0408.zo@gmail.com	Так		Олександр
<input type="checkbox"/>	≡	46815		380970301633	380970301633@greensad.ua	Так		380970301633
<input type="checkbox"/>	≡	46814	28.11.2021 13:34:23	JamesFlith	gioppamercstruale@gmail.com	Так		JamesFlith
<input type="checkbox"/>	≡	46813	27.11.2021 23:05:31	Jerrytab	kvartira38.com@gmail.com	Так		Jerrytab
<input type="checkbox"/>	≡	46812	26.11.2021 12:17:02	elena.vobiachenko@gmail.com	elena.vobiachenko@gmail.com	Так	26.11.2021 12:17:10	Елена
<input type="checkbox"/>	≡	46811	23.11.2021 14:02:27	no7anvalentina	no7anvalentina@ukr.net	Так		Валентина
<input type="checkbox"/>	≡	46810		380505912304	380505912304@greensad.ua	Так		380505912304
<input type="checkbox"/>	≡	46809		380980444611	380980444611@greensad.ua	Так		380980444611
<input type="checkbox"/>	≡	46808	22.11.2021 20:45:15	shirshova.dina64	shirshova.dina64@gmail.com	Так		Дина

Рис. 2.23. Модуль Користувачі

РОЗДІЛ 3 ВІДОБРАЖЕННЯ ГОТОВОГО САЙТУ

3.1. Опис інтерфейсу адміністратора сайту

В дипломній роботі проведено розробку сайту, інтернет магазину GreenSad.

В панелі адміністратора доступні оступні опції: додавання, редагування товарі, замовлень, новин та їх видалення. Для цього потрібно авторизуватися, перейшовши за посиланням `"/bitrix/admin/"`, ми побачимо форму авторизації (Рис. 3.1).

Рис. 3.1. Форма авторизації в адмін-панелі

Кафедра КІТ (47)				НАУ 21 32 00 000 ПЗ			
Виконав	Лазоренко М.О			Опис інтерфейсу адміністратора сайту	Літера	арк	аркушів
Керівник	Віноградов М.А					81	9
Консулт.					УС-212М 122		
Н. Контрол	Райчев І.Е						

Перейшовши на вкладку «Контент», інфоблок «Товари», натиснувши на редагування товару, ми бачимо форми вводу для їхніх значень. В інтерфейсі редагування товару є режим правки в текстовому редакторі.

Це дозволяє адміністратору редагувати текст опис товару, як наприклад в програмі «Microsoft Word», це показано на скріншоті (Рис. 3.2)

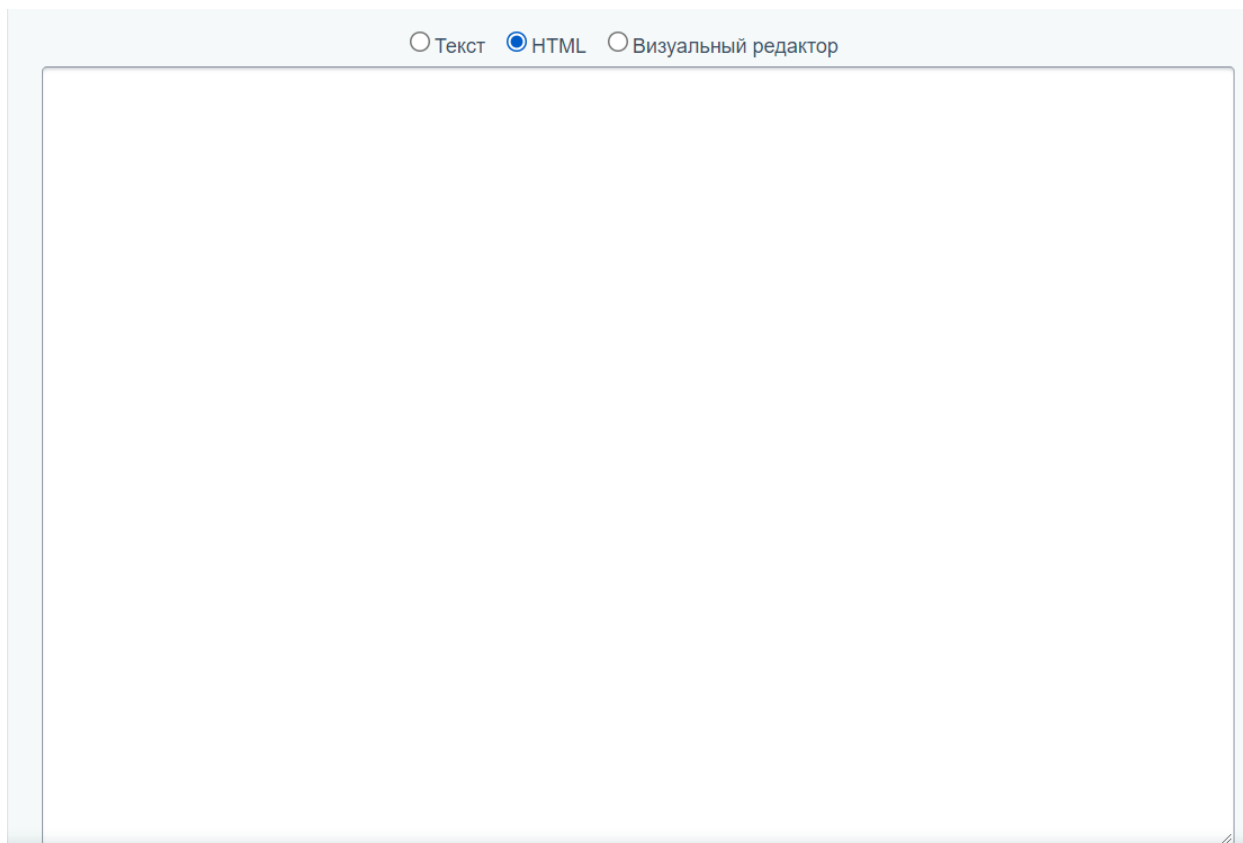
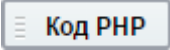


Рис. 3.2. Візуальний редактор

При достатній підкованості користувача, можна перейти в HTML-режим правки тексту. Це дозволяє більш обширно та точно правити текст, додавати до нього стилі за допомогою каскадної таблиці стилів CSS (Рис. 3.3).

У HTML-редакторі можна розмістити довільний php-код таким чином: перейшовши в режим редагування вихідного коду (або змішаний режим) і написавши необхідний код. Потім у режимі візуального редагування (або у відповідному полі змішаного режиму) з'явиться ярлик  **Код PHP** вміст php-коду якого можна редагувати в режимі роботи з вихідним кодом і в області.

Атрибути, які обробляються для посилань, зображень та аудіо:

для атрибутів тегу <a>: href, title, class, style

для атрибутів тегу : src, alt, class, style

для атрибутів тегу <audio>: src, autoplay, control, loop, preload

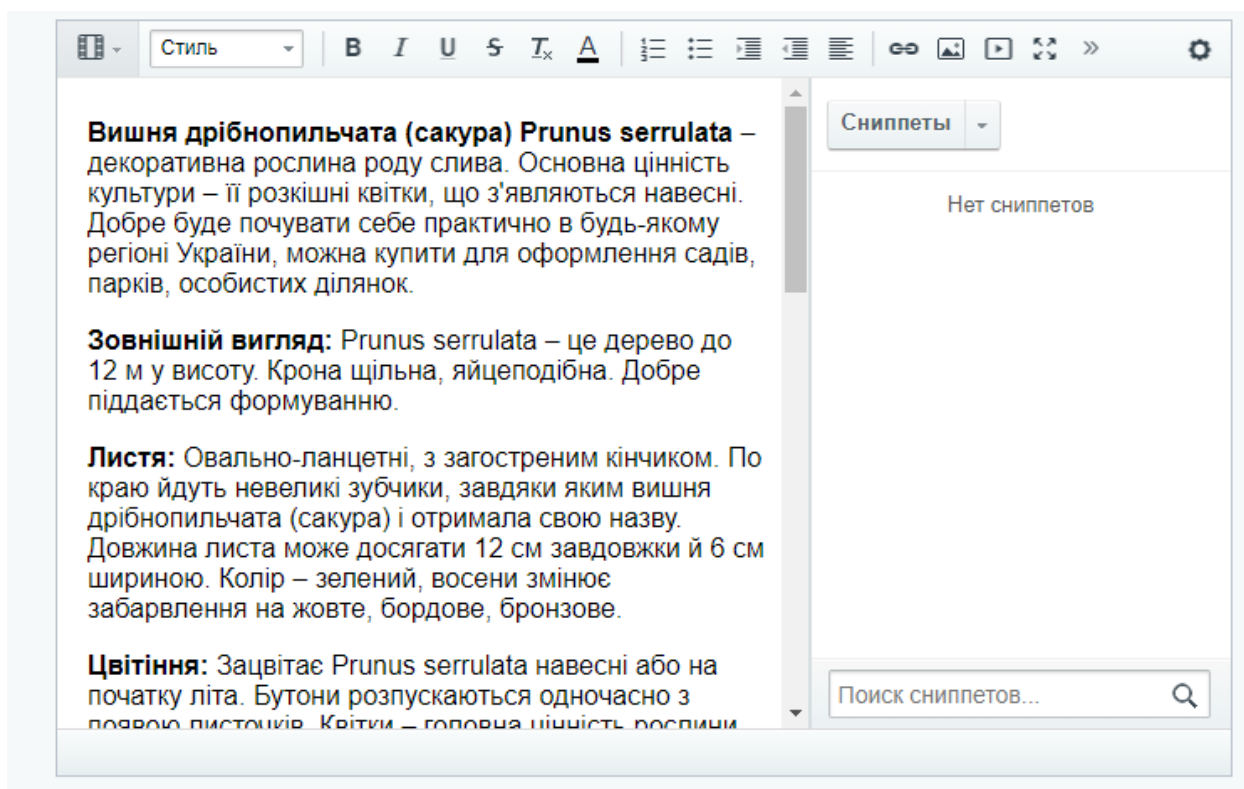


Рис. 3.3. Візуальний HTML редактор

Дуже хорошим плюсом цього редактору є те що якщо користувач в HTML редакторі зробить помилку, наприклад, не закриє тег, чи зробить помилку, то візуальний редактор спробує її виправити, привівши текст до такого вигляду що не викличе помилки у браузері.

3.2. Вигляд сайту для користувачів

При вході на сайт, користувач потрапляє на головну сторінку (Рис. 3.7)

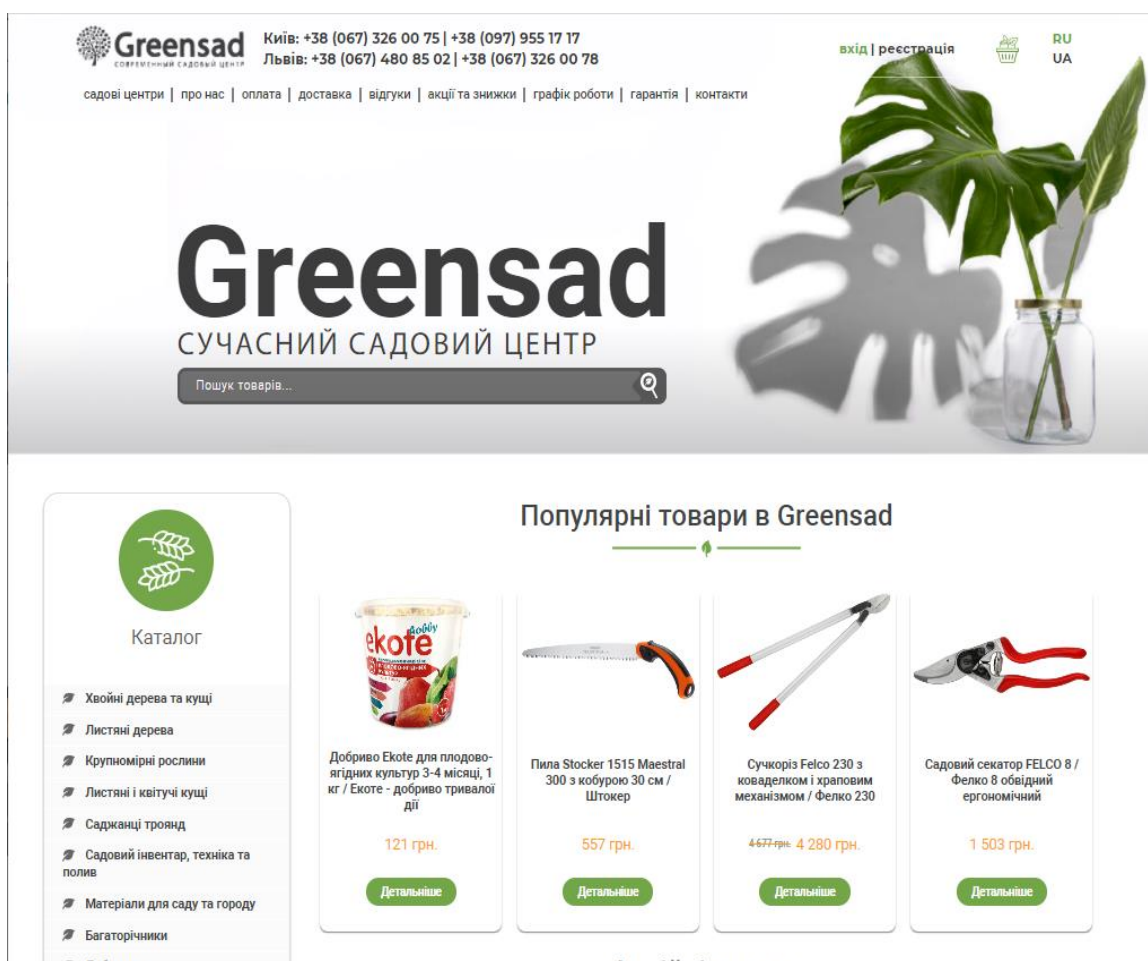


Рис. 3.7. Головна сторінка

На головній сторінці ми бачимо, шапку сайту, форму пошуку, меню з розділами каталогу, список популярних товарів.

Перейшовши на сторінку детального перегляду товару ми бачимо картинку товару, короткий опис, інформацію про наявність товару, та кнопку для покупки товару, або по телефону або за допомогою стандартного функціоналу оформлення замовлення. (Рис 3.8).



Рис 3.8. Детальний перегляд товару

Після того як варвар доданий до кошика (Рис 3.9) можна перейти до оформлення замовлення (Рис. 3.10)

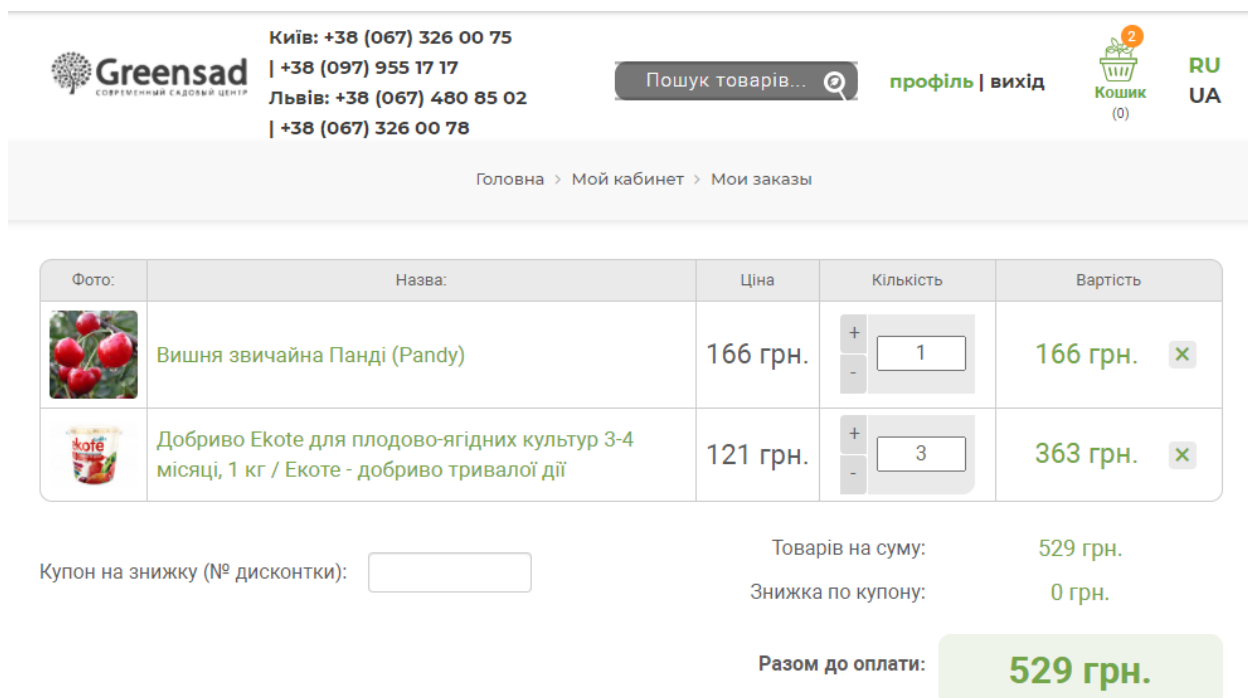


Рис. 3.9. Кошик

Швидке оформлення
Повне оформлення

Контактна інформація

Фамилия: *

Имя: *

Телефон: *

Е-Mail: *

Коментарі до замовлення

Спосіб доставки

Доставка по Киеву и области	Адрес:
Интайм	<input type="text" value="Город"/>
Самовывоз	<input type="text" value="Улица, дом, квартира"/>
Новая почта	
Мист Експрес	

Спосіб оплати

Наличные
 Безналичный расчет
 Предоплата на карту Privatbank

Підтвердити замовлення

Рис. 3.10. Оформлення замовлення

Після оформлення замовлення, менеджер побачить замовлення в адмін панелі і почне його обробку та передачу постачальнику. Передача може бути без дзвінка до клієнта або з дзвінком, залежить він домовленості з постачальником. Можливий варіант коли постачальник буде обробляти замовлення як звичайне замовлення до його системи. Якщо користувач обере наложенний платіж, тобто при отриманні товару на пошті, то постачальник буде зобов'язаний повернути нашу націнку на товар. Якщо клієнт оплатить замовлення у нас на сайті, тоді ми утримаємо нашу націнку, а постачальнику відправимо суму яка буде рівна ціні товару у нього в каталозі.

3.3. Список замовлень

Для менеджменту, замовлень в адмін панелі розроблено функціонал «Список замовлень». Дана сторінка розроблена для відображення замовлень, та їхнього коригування в разі необхідності, перед тим як відправити постачальнику.

	ID	Дата створення	Статус	Покупець	Сума	E-mail покупця
☰	№162612	01.12.2021 17:02:17	В обробці - Україна	татьяна Дергач	466.00 грн.	tanger73@ukr.net
☰	№162611	01.12.2021 16:50:42	В обробці - Україна	татьяна Дергач	1 650.00 грн.	tanger73@ukr.net
☰	№162610	01.12.2021 12:29:59	В обробці - Україна	Виталий	382.00 грн.	vas2warwar@gmail.com
☰	№162609	29.11.2021 22:46:58	Надіслане	Олександр	461.00 грн.	oleksandr0408.zo@gmail.com
☰	№162608	28.11.2021 23:07:58	Скасовано	380970301633	3 465.00 грн.	380970301633@greensad.ua
☰	№162607	28.11.2021 09:14:37	Скасовано	plants2020	1 401.50 грн.	site@plants-club.ua
☰	№162606	23.11.2021 14:02:27	Надіслане	Валентина	1 172.00 грн.	no7anvalentina@ukr.net
☰	№162605	23.11.2021 10:46:56	Скасовано	380505912304	4 950.00 грн.	380505912304@greensad.ua
☰	№162604	22.11.2021 21:21:14	Скасовано	380980444611	1 980.00 грн.	380980444611@greensad.ua
☰	№162603	22.11.2021 20:45:16	Надіслане	Дина	776.00 грн.	shirshova.dina64@gmail.com
☰	№162602	22.11.2021 12:34:25	Надіслане	Якименко Людмила	1 951.00 грн.	380668463835@example.com
☰	№162601	21.11.2021 20:03:01	Скасовано	Андрій	2 475.00 грн.	380632318297@greensad.ua
☰	№162600	20.11.2021 17:03:40	Скасовано	Екатерина	2 640.00 грн.	380930891736@greensad.ua
☰	№162599	19.11.2021 21:55:05	Надіслане	Олександр	805.00 грн.	wynder@ukr.net
☰	№162598	19.11.2021 08:44:44	Надіслане	Татьяна	345.00 грн.	plokhotnyukv@yahoo.com
☰	№162597	18.11.2021 11:41:41	Надіслане	380668835602	1 815.00 грн.	380668835602@greensad.ua
☰	№162596	17.11.2021 12:59:50	Надіслане	380954861112	73.00 грн.	380954861112@greensad.ua
☰	№162595	14.11.2021 14:43:49	Скасовано	380677481890	5 115.00 грн.	380677481890@greensad.ua
☰	№162594	14.11.2021 00:05:34	Надіслане	Евгений	2 231.00 грн.	patriotukraine@ukr.net
☰	№162593	13.11.2021 12:44:07	Скасовано	Ирина Убогая	272.00 грн.	anatoliy_ubogiy@bigmir.net
☰	№162592	12.11.2021 23:41:36	Надіслане	380958303315	2 228.00 грн.	380958303315@greensad.ua
☰	№162591	12.11.2021 13:25:52	Надіслане	380992880022	600.00 грн.	380992880022@greensad.ua
☰	№162590	11.11.2021 00:25:45	Надіслане	Вячеслав	1 828.00 грн.	slavko.batyr@gmail.com
☰	№162589	08.11.2021 15:13:48	Скасовано	дмитро	1 459.00 грн.	medvedmetro@outlook.com

Рис. 3.11. Список замовлень

Даний модуль дозволяє налаштувати список замовлень індивідуально для кожного адміністратора!

Можна додати в список замовлень наступні дані:

- 1) Спосіб доставки.
- 2) Спосіб оплати.
- 3) Кількість куплених товарів і самі товари.
- 4) Електронну пошту покупця.
- 5) Список промоакцій в замовленні.
- 6) Місто покупця.
- 7) Коментарі покупця та адміністратора.
- 8) Дата відвантаження.
- 9) Номер відправлення.

Також можна задати свій порядок сортування, в списку замовлень.

3.4. Проактивний захист

Дуже часто трапляється так що сайт. Може бути об'єктом кібер атак. Для цього передбачений модуль проактивного захисту.

Проактивний захист — це сукупність технологій і методів, що використовуються в антивірусному програмному забезпеченні, основна мета яких, на відміну від реактивних (сигнатурних) технологій, — запобігання зараженню системи користувача, а не пошук у системі вже відомих шкідливих програм. У той же час проактивний захист намагається заблокувати потенційно небезпечну активність програми, лише якщо ця активність становить реальну загрозу. Серйозним недоліком проактивного захисту є блокування легітимних програм (помилкові спрацьовування).

Сьогодні проактивні технології є невід'ємною і невід'ємною складовою антивірусного програмного забезпечення. Антивірусне програмне забезпечення, як правило, використовує комбінацію відразу кількох

технологій проактивного захисту, наприклад, евристичний аналіз і емуляція коду вдало поєднуються з аналізом поведінки, що дає можливість багаторазово підвищити ефективність сучасних антивірусних продуктів проти нових, більш більш складні шкідливі програми.

Технології проактивного захисту:

- 1) Евристичний аналіз.
- 2) Емуляція.
- 3) Аналіз поведінки.
- 4) Пісочниця.
- 5) Віртуалізація робочого простору.

ВИСНОВКИ

В даній роботі спроектовано систему, яка є посередником між постачальником та клієнтом та формує продаж товару за допомогою методу прямої поставки – Dropshipping, коли продавець товару є посередником між постачальником та клієнтом. Створено функціонал для автоматизації процесів в формуванні каталогу та передачі замовлень постачальнику.

Вирішено такі завдання:

- 1) Аналіз методу розробки інтернет магазину для торгівлі, базуючись на бізнес моделі Дропшипінг.
- 2) Аналіз систем управління контентом, які зараз актуальні на ринку;
- 3) Проведено аналіз потреб користувачів та клієнтів веб-ресурсів;
- 4) Виділено моделі якості функціонування сайтів;
- 5) Зібрано актуальну інформацію про використані засоби для розробки;
- 6) Проведено дослідження основних функцій CMS Bitrix.
- 7) Створений функціонал для автоматичного завантаження каталогу товарів від постачальника.
- 8) Створено середовище для обробки замовлень.

СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ

1. Федоричак В. Дропшипінг: переваги та недоліки такої моделі бізнесу [Електронний ресурс] – Режим доступу до ресурсу: <https://webexpert.com.ua/ua/dropshippingperevagi-ta-nedoliki>
2. Діаграма компонентів [Електронний ресурс] // wikipedia – Режим доступу до ресурсу: <https://uk.wikipedia.org>.
3. Документація [Електронний ресурс] // 1С Битрікс – Режим доступу до ресурсу: <https://dev.1c-bitrix.ru/docs/>.
4. Барысов Р. Структура файлів [Електронний ресурс] / Роберт Барысов // Bitrix. – 2010. – Режим доступу до ресурсу: <https://dev.1c-bitrix.ru/learning/course/>.
5. Farhad M. Energy Efficiency in Data Centers and Clouds [Електронний ресурс] / Mehdipour Farhad – Режим доступу до ресурсу: <https://www.sciencedirect.com/topics/computer-science/big-data-processing>.
6. Amazon Web Services [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Amazon_Web_Services.
7. <https://habr.com/ru/company/otus/blog/484048/> [Електронний ресурс]. – 2016.
8. PHP.SU - MySQL для починаючих [Онлайновий] // PHP.SU. - <http://www.php.su/articles/>.
9. Myers, Glenford J. The art of software testing / Glenford J. Myers, Corey Sandler, Tom Badgett. — 3rd ed. - John Wiley & Sons, Inc., Hoboken, New Jersey, 2012. - 240 p.
10. Bird C. The Art and Science of Analyzing Software Data / Christian Bird, Tim Menzies, Thomas Zimmermann - Morgan Kaufmann, Waltham, MA 02451, 2015. - 660 p.
11. Roger Lee (Ed.). Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. / Springer Nature Switzerland AG 2020. - 262 p.

12. Ajay Kumar Jena, Himansu Das, Durga Prasad Mohapatra (Editors) - Automated Software Testing: Foundations, Applications and Challenges. / Springer Nature Singapore Pte Ltd. 2020. - 165 p.

13. А.В. Тирінов. Сутність та особливості обліку операцій дропшипінгу в Україні – Режим доступу до ресурсу: <http://ej.journal.kspu.edu/index.php/ej/article/view/76/73>.

14. В.Н. Ходыревская, И.В. Припадчева, Е.Н. Заикин. Повышение эффективности управления сбытом готовой продукции на сельскохозяйственном предприятии [Электронный ресурс] / Вестник Курской государственной сельскохозяйственной академии. - Режим доступу до ресурсу: <https://cyberleninka.ru/article/n/povyshenieeffektivnosti-upravleniya-sbytom-gotovoy-produktsii-naselskohozyaystvennom-predpriyatii>.